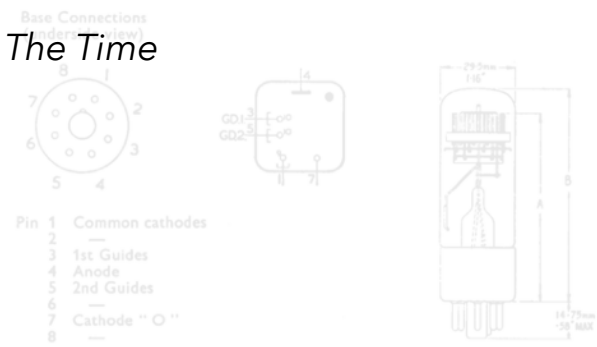


Harwell - A Dekatron Timepiece

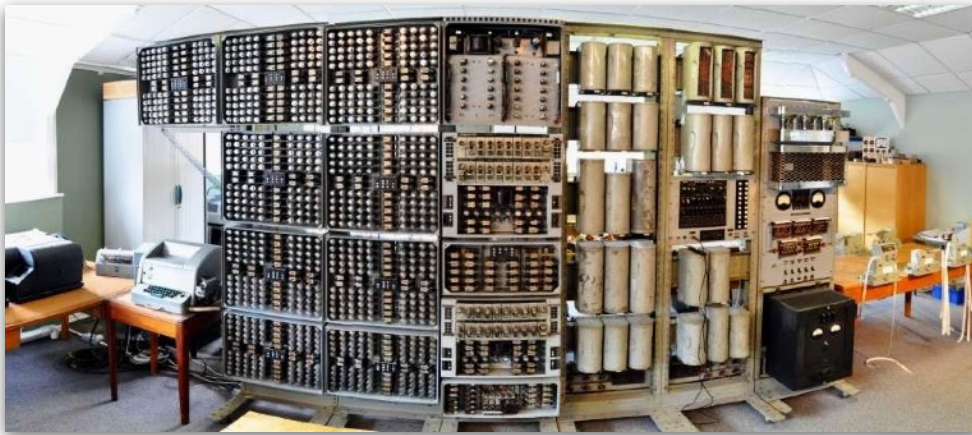
Seven Glow Transfer Tubes Tell The Time



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	72	72	72	72	72
B	85	85	85	85	85

Harwell¹

What is the Harwell? It's a rather special timepiece that uses seven vintage neon counting tubes to tell the time among other *interesting* things. After completing the Fortress E1T clock,² our attention turned to a relic of the early computational age, the Dekatron tube. The clock is named after one of the early relay based computers of the 1950's that made heavy use of these neon filled counting tubes.³



Combining a modern micro-controller with some more cool 'old-skool' technology, we think we've constructed a pretty interesting homage to the computing/counting days of yore. It is also, to our knowledge, the only 7 tube 'commercial' Dekatron clock

deliberately offered for sale ever. It offers lots of customisation options, such as different 'faces',⁴ light-guided LED collars, 'chimes', GPS or WiFi disciplined timekeeping and is also *Open Design and Open Source* in its publication philosophy to boot!

What's Included in the Box?

If you purchased a complete clock, then you'll be receiving the following in the box:

- A Harwell Dekatron clock, 6 tested GC10B Dekatrons, 1 tested GC12/4B Dekatron⁵ and either a GPS or WiFi module (both come with external antennae, depending on your order)
- A pre-built 2A power supply and IEC mains cable.⁶
- A custom programmer used to update firmware and act as a serial console connection.
- An Infrared remote.



¹ Throughout this document, you will note the use of hyperlinks which both lead you to external references as well as internal bookmarks.

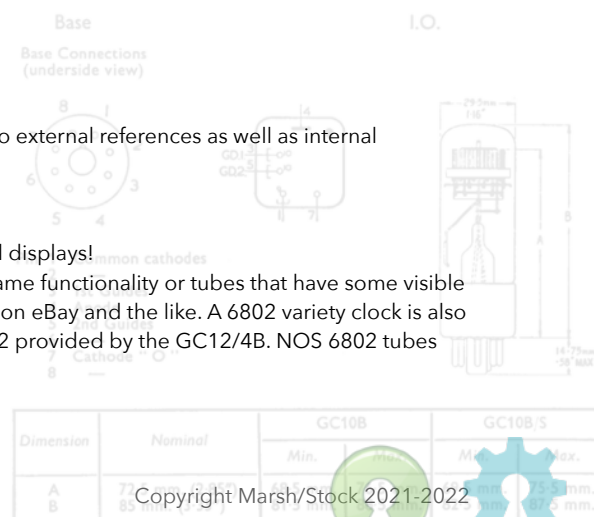
² <http://www.sgitheach.org.uk/fortress.html>

³ https://en.wikipedia.org/wiki/Harwell_computer

⁴ And a whole programming language (Nuggle) so you can design your own colourful displays!

⁵ We can currently offer NOS tubes, tubes of unknown usage but look NOS with the same functionality or tubes that have some visible signs of wear but still function well. These are now quite rare, but occasionally appear on eBay and the like. A 6802 variety clock is also available, the faceplate differs slightly due to the 10 'stations' with this tube over the 12 provided by the GC12/4B. NOS 6802 tubes are also available.

⁶ If readily available to us for your part of the world.....

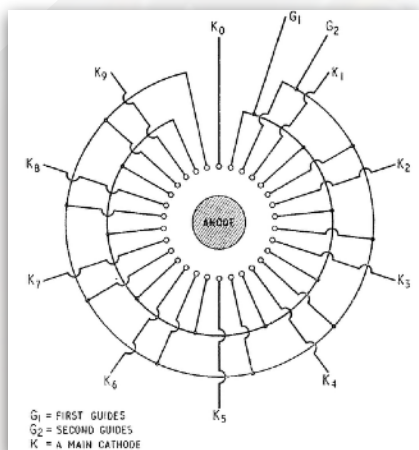


If you purchased a clock kit, then you'll be receiving the following in the box (or boxes):

- A complete kit to build the clock - all the PCBs with SMD parts fitted, all the through hole components to solder, an IR remote and the hardware needed to assemble the boards.
- 6 tested GC10B (or variants) and 1 tested GC12/4B (7 tested GC10B if you prefer). The "quality" of the tubes from apparently NOS down to *well used* is yours to decide (and pay for)!
- An optional kit for either a GPS or WiFi module.
- A kit to build a dedicated 12V 2A power supply.
- A kit to build a custom programmer which will be used to update firmware and act as a serial console connection.
- Optional case kits for the clock and power supply.

What is a Dekatron?

The plethora of methods developed during the late 1940's to the 1970's for displaying/counting numerical information were numerous and varied. These included incandescent displays (e.g. edge lit displays such as those from Non-Linear Systems⁸), neon gas based displays such as Nixie tubes⁹ and Dekatrons, vacuum fluorescent displays¹⁰ (VFDs), light-emitting diodes¹¹ (LEDs) and liquid crystal displays¹² (LCDs). Some of these possess a *certain charm*, such as nixie tubes and Dekatrons with their warm **orange**, neon derived, glow whilst others *not so much*¹³.



A Dekatron is a glow transfer tube or sometimes known as a cold-cathode counting tube. They operate by passing an ionized glow around a ring of cathodes by means of an electrical pulse to intermediate guide cathodes. They came in three basic forms, counting tubes, selectors and register tubes. As such, they were very useful in their day for counting electrical inputs such as in a frequency counter, or radiation counter. For a detailed description of their operation, see the 'Dekatron Tube - How It Works' section taken from the book *Electronic Counting Circuits* by J. B. Dance.



⁷ <https://en.wikipedia.org/wiki/Dekatron>

⁸ <https://stevenjohnson.com/nls/index.htm>

⁹ https://en.wikipedia.org/wiki/Nixie_tube

¹⁰ https://en.wikipedia.org/wiki/Vacuum_fluorescent_display

¹¹ https://en.wikipedia.org/wiki/Light-emitting_diode

¹² https://en.wikipedia.org/wiki/Liquid-crystal_display

¹³ Beauty is in the eye of the beholder....

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72				
B	85				

Anatomy of a Harwell

The clock consists of the following components;

- Seven Dekatron tubes encased in acrylic light guides.¹⁴
- Two MTX90 thyatrons as colons.
- PIR¹⁵ - so the clock can go to 'sleep' if no-one is in the room to bask in its beauty.
- Rotary encoder - to select the different 'faces' to the clock and also to adjust the volume.
- Infrared remote sensor and handset which can also select clock faces and adjust the volume. It can also be used to control other clock functions.
- Two high quality speakers - the Harwell is capable of using .WAV files to play chimes and audio.¹⁶
- SD Card - this contains all the face file descriptors (Nuggle programs), audio files, chimes, system files and *system.ini* file.
- A removable plate at the rear to access both the SD card and the programming port on the CPU board.
- A GPS/WiFi antenna connector.
- PSU Input hole (12V DC rated at a *minimum* of 2A output).

Initial Clock Setup and Operation

First, you'll need to find a good spot to place your clock, the best options being somewhere out of direct sunlight as this will washout the displays and the LED highlighting. The clock does have an auto adjustment for the LED intensity based on the ambient light in the room.¹⁷ You will need to adjust a few things on the SD card so it *knows* which part of the world you're in. These details are kept in a file on the SD card called *locale.ini* in the System folder. You can edit this file using any simple text editor (i.e. Programmers Notepad, Text Edit, Windows Notepad, Text Wrangler etc...). The main thing you will want to change is in which time zone you're located and what DST (Daylight Saving Time) rules to use. Simply change the file accordingly and save your changes. If you're so inclined, you should probably also change your latitude and longitude.¹⁸ You will also want to attach either the GPS or WiFi antenna to the back of the clock. If you're using the WiFi option you will need to enter your network credentials (SSID¹⁹ and Password) in the *wifi.ini* file, again on the SD card in the System folder.

Once that's all complete, plug in the PSU cable to the back of the clock and turn on the power. You will notice that the underside LEDs will all light up and after about a 15 seconds, the Dekatron tubes will illuminate and do a little dance. You will then be greeted with the first of many available 'faces' of Harwell. To change from one to the next, either use the rotary encoder at the front of the clock or the remote control (see page 88 for a full breakdown of the available remote control operations). To change the volume of any chime audio on a particular clock face simply press the encoder in and turn clockwise to increase and anti-clockwise to decrease (the volume level will be displayed on the two rightmost Dekatron tubes). You can also use the IR remote for the same purpose.

¹⁴ As interesting as the Dekatron is as a piece of vintage neon technology, we thought we'd spruce things up with a little extra bling....which you can turn off if you so desire!

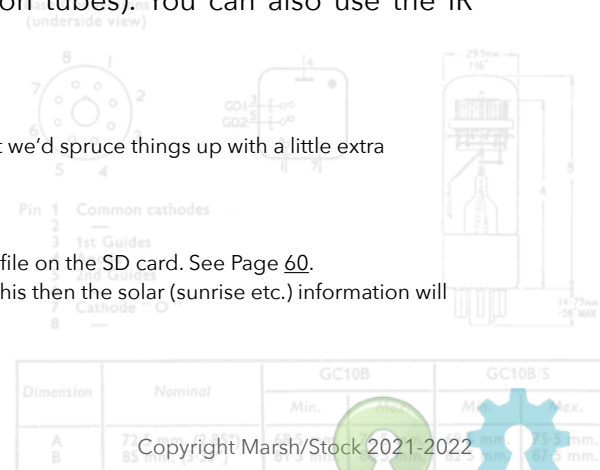
¹⁵ PIR - Passive InfraRed sensor.

¹⁶ Some of it *quite amusing*.

¹⁷ This can be changed, like most other presets in the clock, by altering the *system.ini* file on the SD card. See Page 60.

¹⁸ If you ask us nicely, we can set this all up for you beforehand....if you don't change this then the solar (sunrise etc.) information will be incorrect.

¹⁹ SSID - Service Set Identifier - The given name to your local WiFi network.



Clock Customisation

One of the core principles embedded in an Open Source device such as Harwell is the ability to customise it by *understanding* how it ticks.²⁰ Faces for the clock are written in a hybrid coding language called 'Nuggle'. As supplied, the Harwell comes with a pretty comprehensive set of 'faces' which combine both timekeeping, LED lighting schemes, solar information, screen savers, and the occasional light-hearted audio chime set. All of these are provided on the SD card and can be viewed and edited as needed. For a detailed description of the standard Face Distribution set start at page [15](#).

Clock Care and Troubleshooting

Harwell is a pretty special clock and as such should be suitably cared for to enable a long life and enjoyment for years to come. Please only clean the acrylic with a lint-free cloth (scratches aren't cool), and keep it away from prying fingers. There are high voltages employed within its confines and even though it is well protected, accidents can happen. The Dekatron tubes are very valuable and (alas) a non-renewable resource, so please only operate the clock out of the way of *stray flying objects*...😬. As with most complicated electronic devices, sometimes things don't go according to plan.²¹ If the clock, for any reason, doesn't operate correctly, the first thing to try is a hard reset by turning the power off, waiting 5 seconds then turning power back on. Pulling the PSU socket out and plugging it back in isn't ideal as this can also lead to hanging the CPU, so please only cycle power using the switch on the power supply. If *that* doesn't fix things, then please try using the serial console (see Page [54](#)) to see where things are *hanging up*. There is also a FAQ section in this manual for assistance and if you built the clock from a kit then look at the troubleshooting guide in the clock kit construction manual. If *that* doesn't pinpoint the issue, then please contact us and we will help you to get things up and running again. This doesn't extend to failure of the Dekatron tubes as these are obviously out of our control and have a finite lifespan. We will, however, warranty any issue in the Harwell electronics due to premature component failure or faulty construction for 1 year after date of purchase. This does not extend to accidental events such as spilling drinks on the clock or dropping it on a hard floor etc.²² This warranty also **only applies** to pre-built clocks purchased directly from us. If you are having problems with kit construction we are here to help but some costs may be incurred for such assistance depending on whether shipping the clock is involved etc.

About the Harwell Clock and UTC

Throughout this manual and engraved on the clock itself is the the acronym UTC²³. Strictly speaking the clock does not use UTC but uses an approximation (a *very close* approximation) to it. When using a GPS plug-in module to set the time it will be GPS Time as converted by the GPS module, when using a WiFi module it will be time from a SNTP server and when using neither it will be the time that you set as the user. Each of these time values, GPS, WiFi or User will include some delay from what is actually received and used by the clock. For example, when the clock "ticks" one second, it takes about 40mS before the Dekatron glow positions start to be updated. This may not seem much but for time accuracy nerds it is a huge amount of time. Therefore, when you read "UTC" think that it is the clock's approximation to actual UTC. One important matter is that the clock does use this approximation to UTC internally and any changes to your local time

²⁰ See what I did there? Ha ha ha....

²¹ This should be a rare event.....

²² Common sense should prevail here.

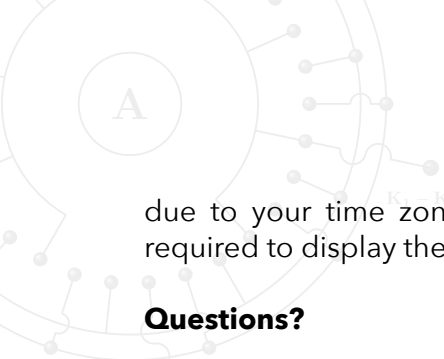
²³ https://en.wikipedia.org/wiki/Coordinated_Universal_Time



Pin 1	Common cathodes
2	—
3	1st Guides
4	Anode
5	2nd Guides
6	—
7	Cathode "O"
8	—

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	72				
B	85				

Copyright Marsh/Stock 2021-2022



due to your time zone and daylight saving adjustments are applied by the clock when it is required to display the local time.

Questions?

We love old display technologies. We have a passion for bringing them back to life in unique and distinctive timepieces and putting them back on display. We also like hearing from people who share our interests in vintage electronics, so if you have any questions regarding Harwell or any other device offered by us, please contact us at either;

sgitheach.org.uk or stocksclocks.com

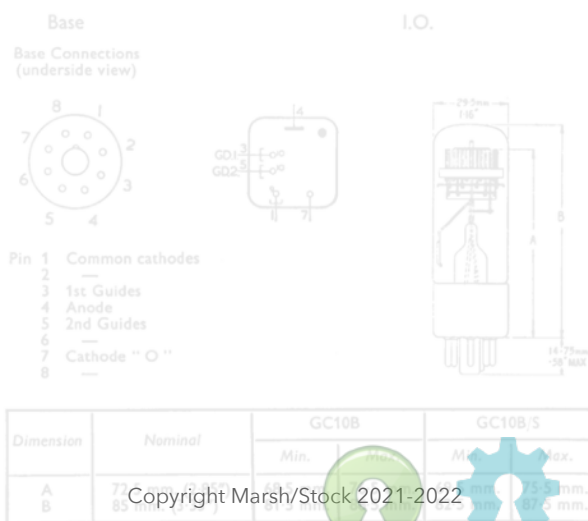




Table of Contents

Quick Start Guide

For the impatient in all of us.

Frequently Asked Questions

Quick answers to burning questions.

Safety Information and Legal

Information on how to use the clock safely, warranty information and the Open Source philosophy involved.

Harwell Clock Distribution Faces

A list of all the clock 'faces' supplied on the SD card and descriptions of their function.

Harwell Clock Operation Manual

Technical aspects of the Harwell Clock design and its operation. Covers the basics on how to modify aspects of the clock to suit your tastes and also troubleshooting.

How The Clock Works

Technical description of all the subsystems and circuitry that makes the clock tick along with schematics.

How a Dekatron Works

An Extract From Dance's "Electronic Counting Circuits" describing how these wonderful vintage devices operate.

The Harwell Dekatron Computer

An excerpt by Kevin Murrell of the Computer Conservation Society and The National Museum of Computing, UK on restoration of the original Harwell Dekatron Computer

Nuggle Supplemental For The Harwell V1.3 and Below

Nuggle specific code additions for the clock for firmware versions up to 1.3.0.

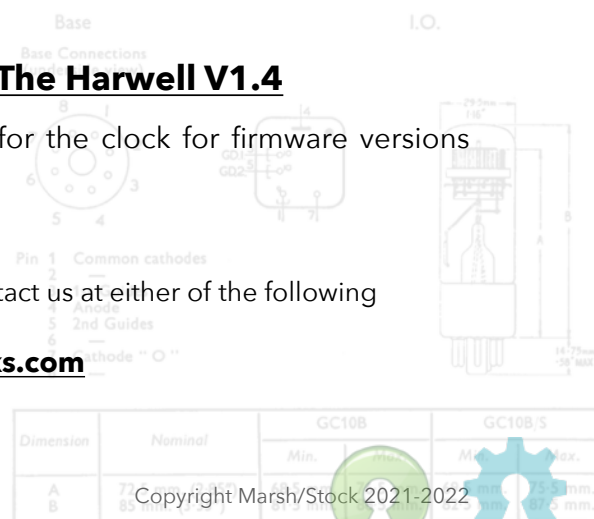
Nuggle Supplemental For The Harwell V1.4

Nuggle specific code additions for the clock for firmware versions 1.4.0 and above.

Annexes

Any questions or concerns, then please don't hesitate to contact us at either of the following

sgitheach.org.uk or stocksclocks.com



Page 8

Pages 9 - 10

Pages 11 - 16

Pages 17 - 52

Pages 53 - 90

Pages 91 - 118

Pages 119 - 133

Pages 134 - 139

Pages 140 - 155

Pages 156 - 171

Pages 172 - 173

Quick Start Guide

Impatient huh? Want to just plug this thing in and get on with your life? Here's the *minimum* you should consider knowing before enjoying the show.

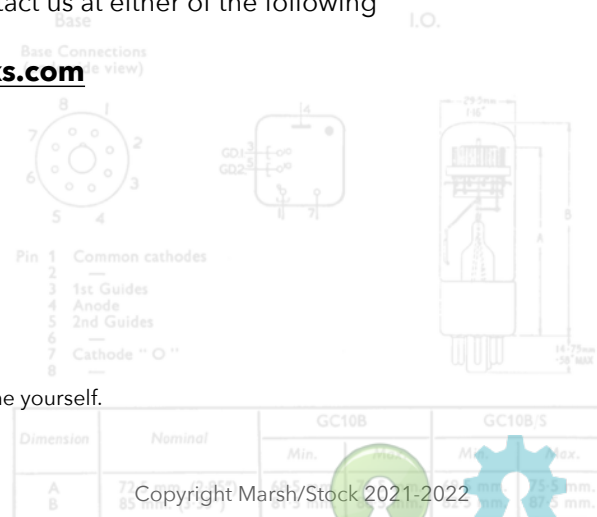
- i) Read the Safety and Legal chapter. Note the do's and don'ts please.
- ii) Change your location details on the SD card.²⁴ Also your WiFi SSID and password if you're using the WiFi option. Make a back up copy of the SD card if you can.
- iii) Attach the antenna to the back of the clock (either the GPS or WiFi version, depending upon your order).²⁵
- iv) Plug the clock in (12V DC, minimum 2A, centre positive) - Clock will take about 20 seconds to 'boot up'. Make sure to operate the power with the switch on the PSU *only*. The time will most certainly be wrong until either the GPS gets its location data or the WiFi unit has got its IP and fetched the time from an NTP server. If you need to manually set the time, go to Face numbers 960 and 961 (page 43).
- v) Change the clock face display with either the rotary encoder knob on the front of the clock or the remote control.
- vi) Change the volume of any chimes by pressing in the knob and turning clockwise (volume up) or anti-clockwise (volume down).
- vii) Sit back and enjoy.
- viii) Read the rest of the manual. 😊 You can learn the following.
 1. Creating your own clock face
 2. How the clock works
 3. Editing *system.ini*
 4. Connecting the console/programmer

Any questions or concerns, then please don't hesitate to contact us at either of the following

sgitheach.org.uk or stocksclocks.com

²⁴ Ignore this if you've already asked us to take care of this at purchase.

²⁵ You may have not opted for either, which is OK, but you'll have to set the correct time yourself.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72.5mm				
B	85mm				

Copyright Marsh/Stock 2021-2022

Frequently Asked Questions

Date and Time

What if I don't have the GPS or WiFi options? How do I change the date and time? You can set the time and date using the clock "faces" 960 and 961 respectively. Or if you plug in the console you can use the menu command **TIMS** and **DATS** respectively. Note that the time you enter must be UTC (Zulu Time) and not your local time.

The time is UK time and not the time here in San Diego? You need to look up your local time zone code in the file `/system/timezone.csv` on the SD card choosing the standard time name and not the daylight saving name. In your case you are in Pacific Standard Time and the code is PST. This code is put in the index key in the [timezone] section of `/system/locale.ini` on the SD card.

How do I check the daylight saving time rule is correct for where I live? You can check the setting `/system/locale.ini` on the SD card by looking at the rule key in the [timezone] section.

The sun rise and sun set times displayed by faces 300 and 301 are not right. Have you put your latitude, longitude and elevation in the [location] section of `/system/locale.ini` on the SD card? These values are need to calculate the position of the sun in the sky in relation to the observer. Will the clock display leap-seconds? No. When the clock is next synchronised by WiFi or SNTP the seconds correction will appear.

How to I display the date in the American MM-DD-YY format? Consider joining WOW and use DD-MM-YY or, even better, the ISO YY-MM-DD format! However, if you want to use the American format then change the endian key to endian-middle in the [datefmt] section of `/system/locale.ini` on the SD card.

Does the clock actually use UTC? Strictly speaking no - it uses GPS time or SNTP fetched time or a time manually set by the user. All should be approximately UTC but not accurately as there are various delays and errors in all three time setting methods as well as (small) delays introduced by the clock itself. Therefore, take the "UTC" label with a small pinch of salt.

Display Units

The clock displays my elevation in metres, I prefer the American units of feet. You can change the displayed units to feet by selecting the units key in the [location] section of `/system/locale.ini` on the SD card.

Chimes and Beeps

How do I turn off all the chimes and beeps all the time? They are driving me/the partner/the dog crazy. The simplest way is to fit Option Jumper 7 on the main board. This will mute all chimes and beeps. When removing or fitting the jumper power cycle the clock for the new setting to be used.

Controls and Communications

What is the USB connector on the SAM3X8C module for? This allows you to access the SD card on the clock as a removable drive on your PC/Mac when plugging it in to your computer.

What does the rotary encoder do? Rotating the encoder changes the face number displayed (clockwise increases the face number); pressing and rotating the encoder changes the chime volume (but not the beep volume) (clockwise increase the volume). Do not spin the controller rapidly but take measured steps.

How do I use the IR handset? The handset can be used to change the operation of many clock functions. The interpretation of some key strokes is changed from default by some clock faces. A description of its default use is in Section 10 of this manual.

The IR handset keystrokes change all of the Sgitheach clocks that I own? The same handset is used with Scope Clock Due, the Fortress E1T Clock and The Harwell Dekatron Clock - read the text on "pairing" in section 10 of this manual. You'll need to make changes to the pairing and pair keys in the [IR] section of *system.ini* on the SD card on each clock and assign a unique number to each clock.

Other IR handsets that I have change clock settings or the IR handset interferes with the operation of other appliances I own. What to do? This is perhaps unfortunate but will happen whenever you bring a new IR handset into your home. So far we only have one case of another appliance (an A/C unit) that responded to the handset and no case of another handset changing the clock operation. If you can live without the IR control of your other appliance then don't use it. You can disable the Harwell IR receiver by changing the enable key to off in the [IR] section of *system.ini* on the SD card. Also, read the section on "pairing" text in section 10 of this manual that is used to prevent the handset from changing more than one Sgitheach clock at a time, this might help with your appliance interferences.

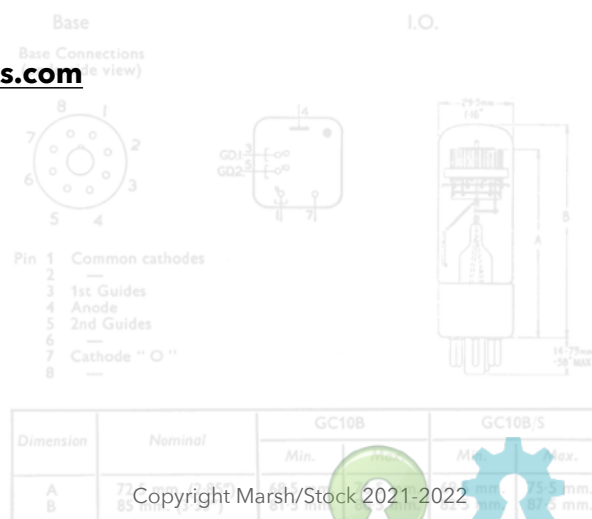
The IR handset no longer works? Err... has the battery gone flat? It should see many years of use. Is pairing in use? Read the text on pairing in section 10 of this manual. If you have pairing on then you need to use the correct keystroke sequence to "pair" the handset to the clock. When you have the IR handset working (or use the encoder) select clock face 950 to see the pairing status.

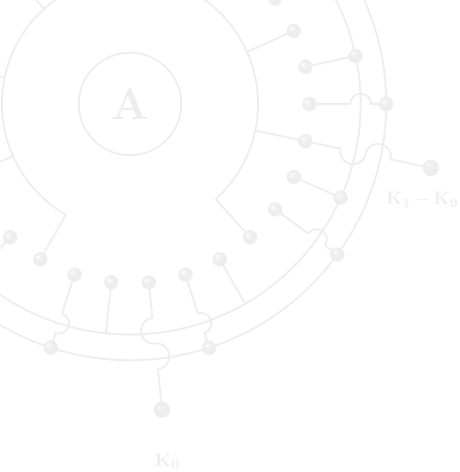
How does the serial console work? How do I use it? The programmer/console box plugs into the SAM3X8C plug in module and is connected to your PC using a USB cable. You must discover the serial COM number assigned by Windows using the Device Manager. Then use your favourite serial console program (I use Tera Term) to connect at 115200 baud, 8 data bits, 1 stop bit, no parity, no flow control. See page XX in this manual.

Contact

How do I contact you? You should know our email addresses from purchase correspondence but if you have lost them then use the email addresses that you find on these web pages ...

sgitheach.org.uk or stocksclocks.com

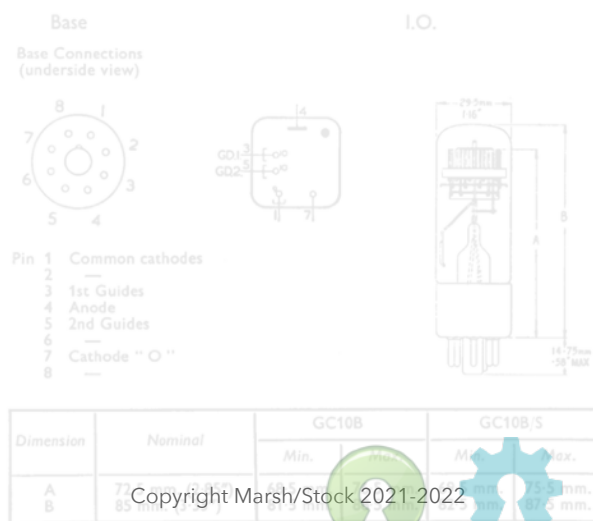




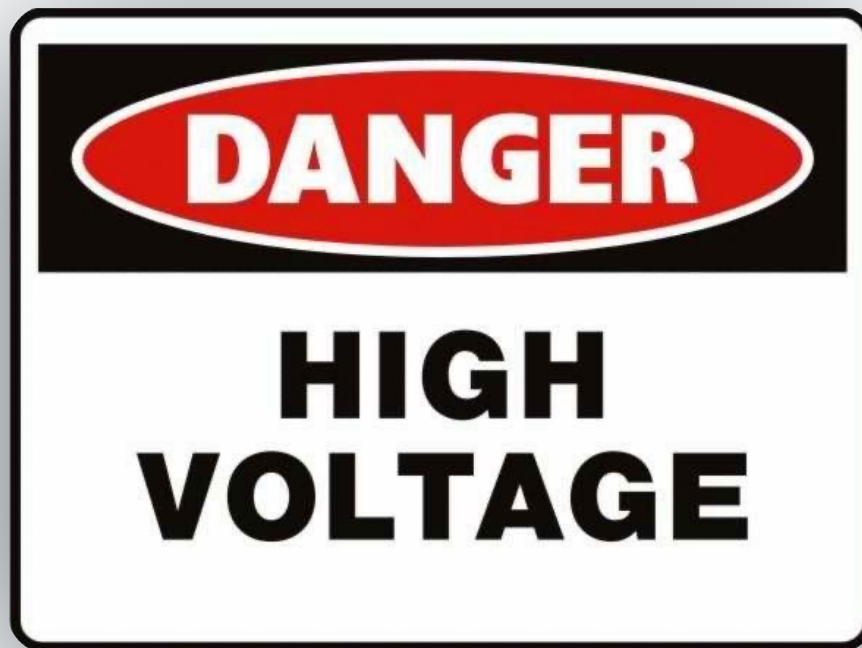
SAFETY AND LEGAL²⁶

SAFETY, LEGAL & LICENSE CONSIDERATIONS

²⁶ You know, the fun stuff....



Important Safety Information



Safety Statement

Like all Dekatron clocks,²⁷ The Harwell uses high voltages in order to operate the tubes. You need to respect this and other hazards inherent in these circuits.

Caution! The Harwell clock must be correctly earthed (grounded) using only a 3-core mains cable to a correctly earthed mains outlet.

Caution! The Harwell generates high voltages in the region of 500V during operation. These voltages are present on the power supply board and base of the Dekatron tubes; These voltages may be maintained for a period of time after input power is removed.

Caution! Do not touch the electronics while the clock is in use or has been recently operated. Treat the clock with the same level of care and common sense as any mains-powered electrical device – do not expose to wet environments, keep out of the reach of children, animals etc. Do not eat!

Caution! Some components may be warm to the touch during use. This is a perfectly normal consequence of their operation, but you should remember it when handling the board or considering alternative clock enclosures.²⁸

Caution! The Dekatron tube envelope is made of glass and may be broken if the clock is dropped or inadvertently struck.²⁹

Caution! If you have built the clock into your own custom metal case then ensure that the case is earthed/grounded by connecting it to the screw connector on the PSU board provided for *exactly* that purpose.

²⁷ Rare beasts they are...

²⁸ We're pretty proud of the way the supplied case came out, but you just can't please some people....

²⁹ We have replacements, but *they'll* cost you....

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72				
B	85				

Legal Statement

The Harwell clock is built and documented with an Open Source philosophy in mind. All the source files including circuit diagrams, Eagle board, software and design files are provided under a Creative Commons ShareAlike 4.0 International license.

More specifically;

- You may share, copy and redistribute the material in any medium or format,
- You may remix, transform and build upon the material presented herein,
- You MUST give appropriate credit, provide a link to the license and indicate if changes have been made.
- This license is for NON-COMMERCIAL use only, you may not use the material for commercial gain.
- If you remix, transform, 'improve', modify or build upon the material presented herein, you must distribute your contributions under the same license as the original.

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

For further information, please see the following URL: <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

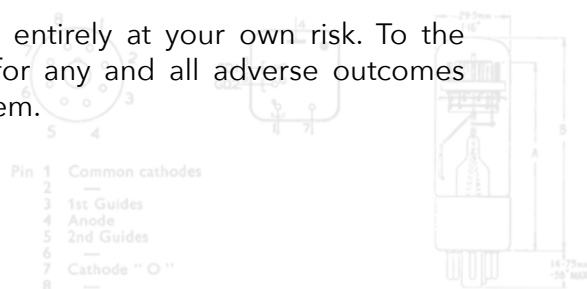
In addition you should note the following (in the event that there is any conflict between these notes and the License given above, then the License shall take priority).

The Harwell Clock may be hazardous if not assembled and operated by suitably knowledgeable and practised persons or if abused. It is your responsibility to carefully review the documentation, the design and the kit contents, and to assure yourself that you have the necessary expertise to construct and/or operate the clock safely. In particular, it is also your responsibility to ensure that the completed clock meets any necessary safety and other regulations or guidelines for such items in your jurisdiction. In that respect, any supplied enclosure is intended as a basis for you to customise the final clock to meet such regulations. It is possible that in some jurisdictions, a completely different type or construction of enclosure may be required to obtain regulatory compliance. Assembly instructions in the kit documentation are intended as a starting point, to be amended or not according to the judgement and expertise of a suitably qualified constructor.

The hazards of this kit include, but are not limited to, high voltages, the generation of heat during operation, the presence of toxic substances within the components of the kit and the presence of sharp and/or fragile glass and metal items. Not all components within this kit comply with the Restriction of Hazardous Substances regulations (RoHS), though compliant components have been selected in most cases.

In summary, you own, construct and use the Harwell Clock entirely at your own risk. To the maximum extent permitted by law, we disclaim all liability for any and all adverse outcomes associated with your ownership, construction and use of this item.

Base Connections
(underside view)



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	73				
B	85				

Copyright Marsh/Stock 2021-2022

Atmel Licence

The Harwell firmware makes extensive use of the Atmel³⁰ libraries. The Atmel License (also given in every Atmel library file) is

Copyright © 2012-2015 Atmel Corporation. All rights reserved.

License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of Atmel may not be used to endorse or promote products derived from this software without specific prior written permission.
- This software may only be redistributed and used in connection with an Atmel micro-controller product.

THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

My distribution of the Atmel libraries as part of the open source firmware is in compliance with this license.

For further information visit <http://www.microchip.com>

LUFA Licence

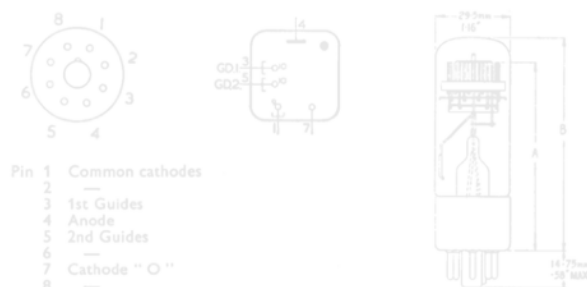
The Harwell uses a custom SAM3X8C core board and as such makes use of a programmer that uses LUFA.

LUFA Library
Copyright © Dean Camera, 2010.

dean [at] fourwalledcubicle [dot] com
www.fourwalledcubicle.com

³⁰ Now owned by MicroChip

Base Connections
(underside view)



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	7.5				
B	8.5				

Copyright Marsh/Stock 2021-2022

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the author not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The authors disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

For further information see <http://fourwalledcubicle.com/LUFA.php>

JSMN minimalist JSON parser - <https://zserge.com/jsmn/> - MIT License

Copyright © 2010 Serge Zaitsev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

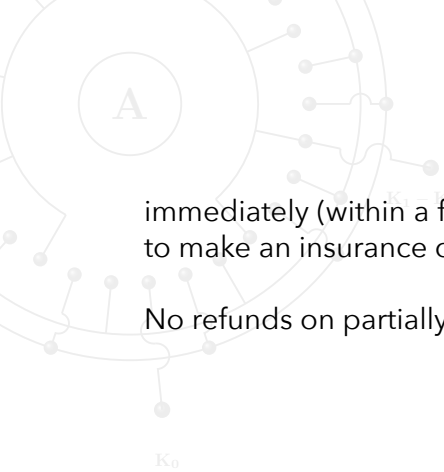
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Warranty Information

Upon receipt of the kit of parts, any missing or broken pieces will be replaced. It is incumbent upon the recipient to check the contents in a prompt manner against the supplied parts lists found within the construction manuals. As a kit of parts, no warranty can be provided pertaining to the quality of construction and operation of the final product as this is the duty of the purchaser and is dependent upon their skill. The Harwell clock may be hazardous if not assembled and operated by suitably knowledgeable persons and it is the owners responsibility to carefully review all the supplied documentation. The authors have made their best attempts to explain and detail the construction and hazards associated with operation of the clock within the supplied manuals. Due to the nature of the obsolete technology employed in the Harwell clock (primarily the Dekatron tubes themselves), certain hazards are present, namely high voltages and fragile glass under vacuum and due care and attention should be paid when handling said items.

If you have purchased a complete operational clock, then a limited warranty is provided in a separate document supplied with your documentation. If the clock kit or complete clock has arrived in a damaged state such that an insurance claim is likely to be made, then please notify us

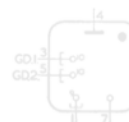


immediately (within a few days of receipt). It is likely that photographic evidence will be asked for to make an insurance claim.

No refunds on partially or fully constructed kits are possible.



Base
Base Connections
(underside view)

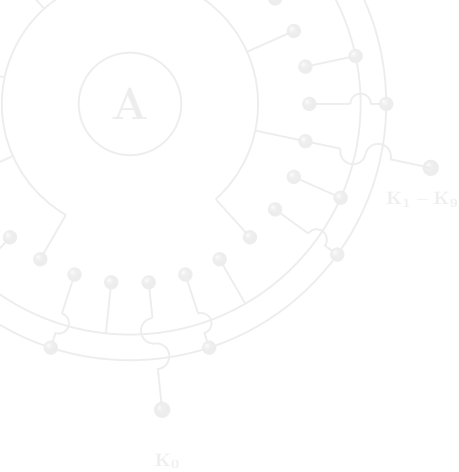


I.O.



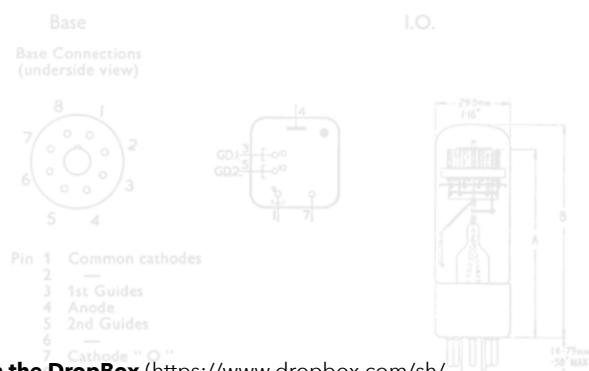
Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72 mm (2.83")	72 mm (2.83")	72 mm (2.83")	72 mm (2.83")	72 mm (2.83")
B	85 mm (3.35")	85 mm (3.35")	85 mm (3.35")	85 mm (3.35")	85 mm (3.35")





Harwell - Clock Distribution Faces³¹

The Standard Face Feature Set



³¹ For the latest Harwell Faces distribution, see the Harwell Face Supplemental on the DropBox (<https://www.dropbox.com/sh/vbxwpxwj4dzgiuu/AAADKYDZXZ3gdRCGZ0uGjtjda?dl=0>)

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72.5 mm 2.85 in.				
B	85 mm 3.35 in.				
C	75 mm 2.95 in.				

Copyright Marsh/Stock 2021-2022



Introduction

This manual section describes the clock “faces” that you will find on the supplied SD card (or the standard set you download should you be building the clock yourself). Whilst you can just use these distribution faces and their numbering and arrangement, you are encouraged to rearrange the faces and pick the ones to commonly use yourself. You are also encouraged to edit the face files, change the sound files and reconfigure the displayed clock faces to your own amusement. If you take the plunge and decide to start coding your own faces, then please share with us!!³²

Clock Face Numbering

There is only one rule: clock faces must be numbered between 1 and 999999.

The rest of this description should not be taken as hard and fast rules - this is just how the faces are arranged on the SD card at distribution - all can be changed at your whim.

Clock faces are normally selected on the Harwell Clock by:

- Turning the rotary encoder when not pressed
- Pressing the CH- or CH+ keys on the handset
- By “dialling” in number using the 0 to 9 keys and pressing EQ.

To select quickly and easily, I arrange my favourite faces to be in the number range 1 to 9 and other good clock faces that I might occasionally want to use in the range 10 to 99. This minimises the ‘travel’ distance using the encoder and key presses when using the IR handset. I use numbers above 100 to display all the clock faces (even repeating the favourite ones) in logical groups so that the full range is available for use and for demonstration.

Clock Face Filenames and Number Assignment

Clock faces are Nuggle program files living in the `/nuggle` folder on the SD card with `.nug` file extensions. The [faces] section in the `nuggle.ini` file³³ is used to link a required clock face number to a Nuggle file.

For example:

[faces]

4 = awest

will make the `awest.nug` program execute when clock face number 4 is selected.

Note that whilst face numbers can only be used once, the Nuggle filenames can appear more than once. Therefore in addition to the assignment of `awest.nug` to face number 4, you could have:

152 = awest

further down the list. This enables you to build clusters of popular face numbers in the list.

³² We look forward to what you come up with..... this is Open Source *after all*.

³³ In the `/system` folder on the SD card

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75	75	75	75.5	75.5
B	85	85	85	87.5	87.5

My Favourite Clock Face Selection

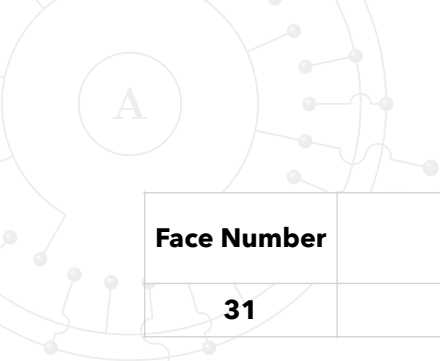
Whilst you can choose yourself how to use the face numbers, I use faces 1 to 9 for my favourites, as the effort to select one of them is therefore minimal. In all cases these faces are duplicates of main grouped clock faces with numbers 100 and above. You will need to look there for the detail of each clock face.

Face Number	Description	Grouped Face Number	Filename
1	Blanked clock - all dekatrons are off and bling set to black	400	<i>blank.nug</i>
2	Simple time with spinner, no bling, no chimes	105	<i>datetime.nug</i>
3	Date and time with lots of bling and TinTan chimes	200	<i>datetime.nug</i>
4	Date and time, bling, no chimes	121	<i>datetime.nug</i>
5	Date and time, bling and Derby chimes	123	<i>datetime.nug</i>
6	Original Star Trek Themed	160	<i>startrek.nug</i>
7	Holy Grail Themed	192	<i>holyclck.nug</i>
8	Date and time with daily changing chimes	124	<i>datetime.nug</i>
9	Star Wars Themed	180	<i>starwars.nug</i>

Don't forget these are *my favourites only*. If you find that the bling is not quite the right shade or you want different timing - the face is not entirely satisfactory³⁴ - then please look at editing the faces selected list or the Nuggle files themselves. However, I do recommend keeping clock face 1 as the blank clock. In some cases an error condition in a Nuggle program needs a clock face to move to. I use face 1 for this purpose.

Face Number	Description	Grouped face number	Filename
10	Holy Grail Story	190	<i>holyclck.nug</i>
11	Black Knight Fight	191	<i>black.nug</i>
12	Next Generation Star Trek (incomplete)	161	<i>startrek.nug</i>
13	PIR Game	350	<i>pir.nug</i>
14	Larsen Scanner	430	<i>larsen.nug</i>
15	Cuckoo Clock	125	<i>datetime.nug</i>
20	Sun Rise	300	<i>sun.nug</i>
21	Sun Set	301	<i>sun.nug</i>
22	Sun Position	302	<i>sun.nug</i>
30	RTC Sync	962	<i>syncrtc.nug</i>

³⁴ .. for these extremely rich merchants life eventually became rather dull and it seemed that none of the worlds they settled on was entirely satisfactory: either the climate wasn't quite right in the later part of the afternoon, or the day was half an hour too long, or the sea was just the wrong shade of pink. And thus were created the conditions for a staggering new form of industry: custom-made luxury planet building. - H2G2 Douglas Adams.



Face Number	Description	Grouped face number	Filename
31	Tick Adjustment	963	rtcfine.nug

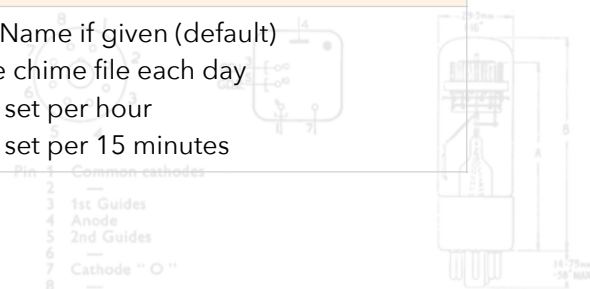
Distribution Clock Faces in Groups

Standard Date and Time Faces 100 -

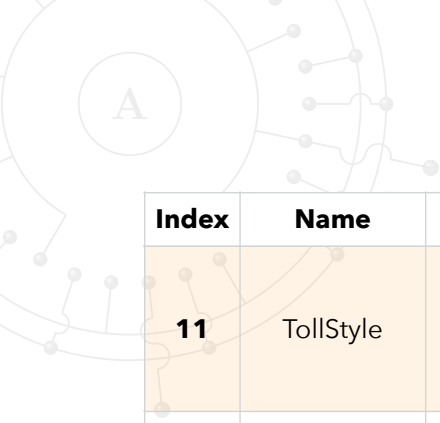
About datetime.nug:

This nuggle file is used for faces that show the time and/or date. It takes a lot of arguments to configure its use (text in *italics* are either system constants or constants define in *constants.ini*):

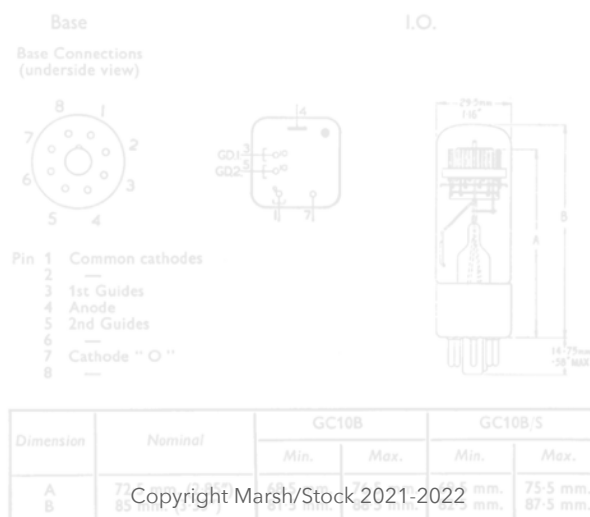
Index	Name	Type	Details
1	FaceTitle	string	This will appear on the console when the face has run
2	ShowTime	integer	0 <i>false</i> time not shown (default) 1 <i>true</i> time shown 2 <i>decimal</i> (special) shows time in decimal hh.frac
3	ShowDate	boolean	<i>True</i> will show the date, default is <i>false</i>
			If both ShowTime and ShowDate are true then the date is show between 40 and 50 seconds during each 1 minute cycle
4	ColonStyle	integer	0 <i>cs_none</i> show neither colons (default) 1 <i>cs_left</i> show the left colon 2 <i>cs_pair</i> show both colons 3 <i>cs_right</i> show the right colon
5	ColonTime	integer	The PWM value (0 - 100) when the time is shown
6	ColonDate	integer	The PWM value (0 - 100) when the date is shown
7	SpinStyle	boolean	<i>True</i> will spin the Dekatrons as seconds and minutes advance
8	TickStyle	integer	0 <i>ts_none</i> none (default) 0 <i>ts_tick</i> one chime file used 0 <i>ts_ticktock</i> two chime files used on alternate seconds
9	ChimeName	string	The base name of the chimes to use. The clock will add to the name to derive the quarterly and toll chime filenames
10	ChimeRotate	integer	0 <i>cr_none</i> use ChimeName if given (default) 1 <i>cr_daily</i> change the chime file each day 2 <i>cr_hourly</i> one chime set per hour 3 <i>cr_quarterly</i> one chime set per 15 minutes




Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm





Index	Name	Type	Details		
11	TollStyle	integer	0	<i>os_none</i>	no tolls (default)
			1	<i>os_chime</i>	tolls based on chime name
			10	<i>os_westc</i>	special case for west-c bells
			20	<i>os_ships</i>	ship's bells
12	BlingStyle	integer	0	<i>bs_none</i>	none (default)
			1	<i>bs_single</i>	single colour simple bling, all and down in BlingColour1
			2	<i>bs_double</i>	colour all in BlingColour1 down bling in BlingColour3
			3	<i>bs_down</i>	down bling only BlingColour1
			4	<i>bs_resistor</i>	resistor colour codes
			5	<i>bs_rainbow</i>	rotating rainbow bling
			6	<i>bs_chase</i>	fast rotating bling single colour
			7	<i>bs_swirl</i>	fast rotating HSV rainbow
13	BlingColour1	integer	Colour values which can be hex a decimal in the form 0xRRGGBB where RR is the red intensity 00 to ff, and so on for GG green and BB blue. Standard colour constants can be used, e.g. <i>dim red</i>		
14	BlingColour2	integer			
15	BlingColour3	integer			
16	BlingColour4	integer			
17	BlingColour5	integer			

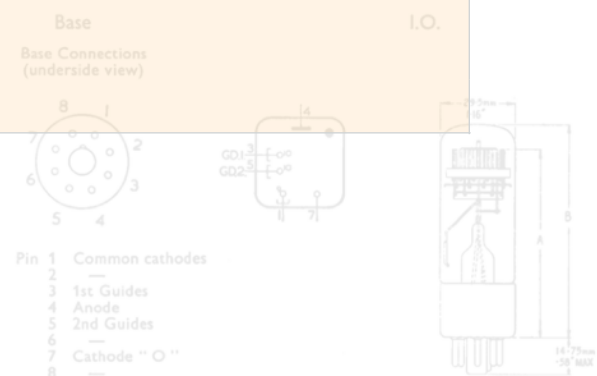


Time Only Faces 100

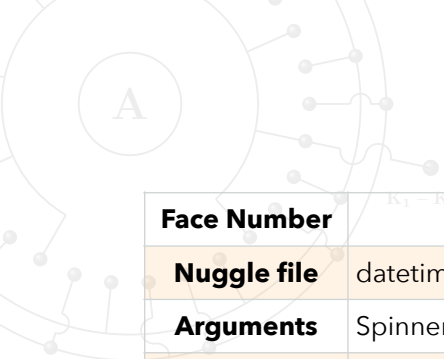
Face Number	100	
Nuggle file	datetime.nug	
Arguments	Time Only, true	
Description	Show the time only with no bling or colons	
Colons	None	
Bling	None	
Chimes	None	


Face Number	101	
Nuggle file	datetime.nug	
Arguments	Time Only- Red Bling, true, , cs_pair, 10, , , , , bs_double, smolder, dimred	
Description	<ul style="list-style-type: none"> Shows the time only, dim red bling, colons flash dimly, no chimes V1 to V6 - show the time V7 - indicate Time 	
Colons	Dim flash	
Bling	Dim red colours	
Chimes	None	


Face Number	102	
Nuggle file	datetime.nug	
Arguments	Time Only- Blue/Cyan Bling, true, , cs_pair, 100, , , , , bs_double, deepblue, dimblue	
Description	<ul style="list-style-type: none"> Shows time only, dim blueish colours, colons flash bright, no chimes V1 to V6 - show the time V7 - indicate Time 	
Colons	Bright flashing	
Bling	Blueish colours	
Chimes	None	

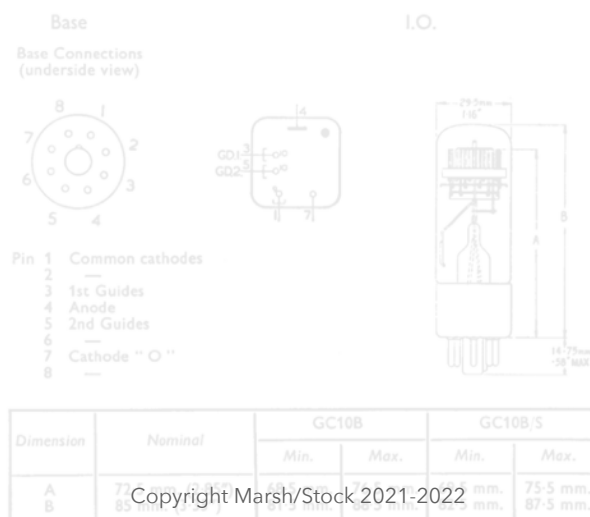


Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	72 mm	72 mm	72 mm	72 mm	72 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm



Face Number	105	
Nuggle file	datetime.nug	
Arguments	Spinner & Time Only, true, , , , , true	
Description	<ul style="list-style-type: none">Time only - spinning dekatrons - no bling, colons or chimesV1 to V6 - show the timeV7 - indicate Time	
Colons	None	
Bling	None	
Chimes	None	


Face Number	106	
Nuggle file	datetime.nug	
Arguments	Spinner & Time Green Bling, true, , cs_pair, 100, , true, , , , , bs_single, darkgreen	
Description	<ul style="list-style-type: none">Time only - green bling with spinning dekatronsV1 to V6 - show the timeV7 - indicate Time	
Colons	Flashing	
Bling	All green bling	
Chimes	None	




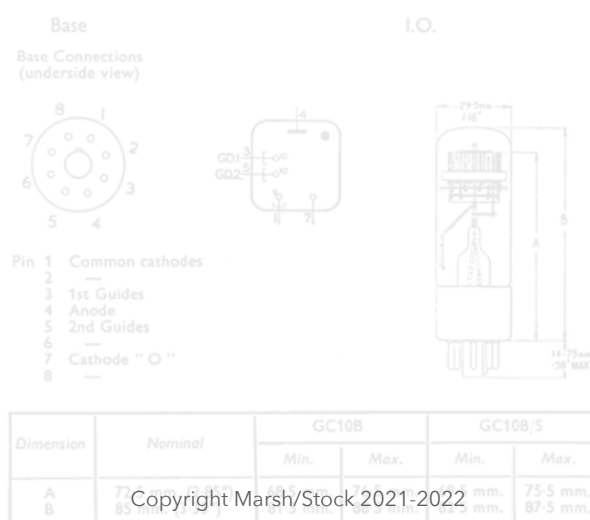


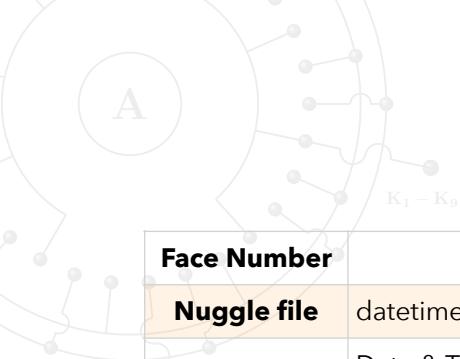
Dimension	Nominal	GC10B		GC10B.5	
		Min.	Max.	Min.	Max.
A B	72.5 mm 85 mm	69.5 mm 82 mm	75.5 mm 88 mm	68.5 mm 81 mm	75.5 mm 87.5 mm


Date and Time Faces

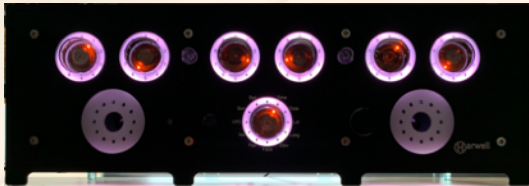
Face Number	121	
Nuggle file	datetime.nug	
Arguments	Date & Time - Rainbow Bling, true, true, cs_pair, 100, 25, , , , , bs_rainbow	
Description	<ul style="list-style-type: none"> Show the date and the time, use rotating rainbow hues bling, colons and no chimes V1 to V6 - date and time V7 - indicate Date or Time as appropriate 	
Colons	Flashing and bright for time, steady and dim for date	
Bling	Rotating rainbow hues	
Chimes	None	

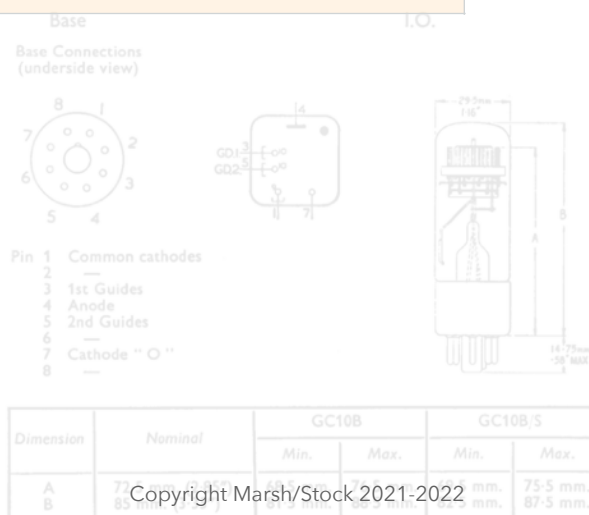
Face Number	123	
Nuggle file	datetime.nug	
Arguments	Date & Time - Rainbow Bling, true, true, cs_pair, 100, 25, , ts_tick, derby, , os_chime, bs_rainbow	
Description	<ul style="list-style-type: none"> Show the date and time, use rotating rainbow hues bling, colons and Derby chimes V1 to V6 - date and time V7 - indicate Date or Time as appropriate 	
Colons	Flashing and bright for time, steady and dim for date	
Bling	Rotating rainbow hues	
Chimes	derby*.wav	






Face Number	124	
Nuggle file	datetime.nug	
Arguments	Date & Time - Rainbow Bling & Rotating Chimes, true, true, cs_pair, 100, 25, , ts_tick, a-west, cr_daily, os_chime, bs_rainbow	
Description	<ul style="list-style-type: none"> Show the date and the time, use rotating rainbow hues bling, colons and Westminster chimes initially then change the chimes each day. V1 to V6 - date and time V7 - indicate Date or Time as appropriate 	
Colons	Flashing and bright for time, steady and dim for date	
Bling	Rotating rainbow hues	
Chimes	west-a*.wav initially then any of the clock chime file sets	


Face Number	125	
Nuggle file	datetime.nug	
Arguments	Date & Time Cuckoo Clock, true, true, , , , , ts_ticktock, cuckoo, , os_chime, bs_single, wheat	
Description	<ul style="list-style-type: none"> Show the date and the time, use single colour bling, no colons and Cuckoo toll noises. V1 to V6 - date and time V7 - indicate Date or Time as appropriate 	
Colons	Flashing and bright for time, steady and dim for date	
Bling	wheat	
Chimes	cuckoo*.wav	




Time Only Faces with Clock Chimes and no Bling

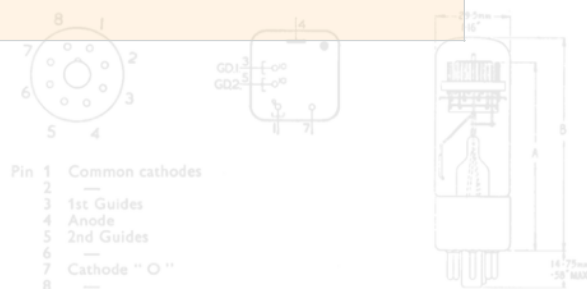
Not all of the available chimes are demonstrated here.

Face Number	130	
Nuggle file	datetime.nug	
Arguments	Time Only - No Bling - Westminster A, true, , cs_pair, 100, , , ts_ticktock, a-west, , os_chime	
Description	Westminster Chimes (alternative A)	
Colons	None	
Bling	None	
Chimes	west-a*.wav	

Face Number	131	
Nuggle file	datetime.nug	
Arguments	Time Only - No Bling - Westminster B, true, , cs_pair, 100, , , b-west, , os_chime	
Description	Westminster Chime (alternative B)	
Colons	None	
Bling	None	
Chimes	west-b*.wav	


Face Number	132	
Nuggle file	datetime.nug	
Arguments	Time Only - No Bling - Westminster C, true, , cs_pair, 100, , , c-west, , os_westc	
Description	Westminster Chime (alternative C)	
Colons	None	
Bling	None	
Chimes	west-c*.wav	


Base Connections
(underside view)





Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Copyright Marsh/Stock 2021-2022

Face Number	135	
Nuggle file	datetime.nug	
Arguments	Time Only - No Bling - Ave Maria, true, , cs_pair, 100, , , , amaria, , os_chime	
Description	Time only with no bling, Ave Maria chimes	
Colons	None	
Bling	None	
Chimes	amaria*.wav	


Face Number	136	
Nuggle file	datetime.nug	
Arguments	Time Only - No Bling - Whittington, true, , cs_pair, 100, , , , whit, , os_chime	
Description	Whittington Chimes	
Colons	None	
Bling	None	
Chimes	whit*.wav	


Face Number	137	
Nuggle file	datetime.nug	
Arguments	Time Only - No Bling - St Stephens, true, , cs_pair, 100, , , , steph, , os_chime	
Description	St. Stephen's Chimes	
Colons	None	
Bling	None	
Chimes	steph*.wav	

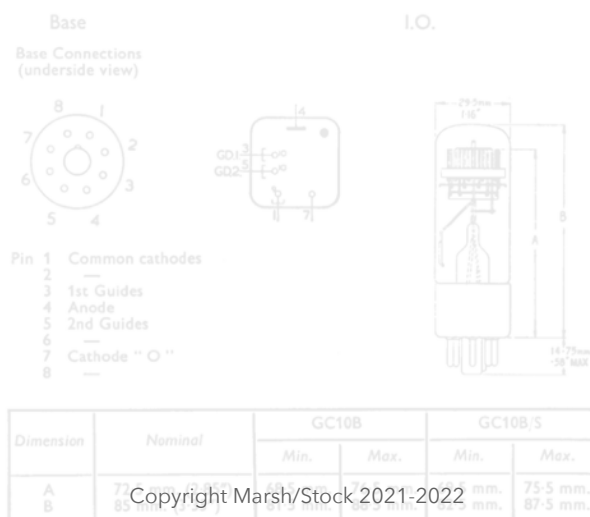
Face Number	140	
Nuggle file	datetime.nug	
Arguments	Time Only - No Bling - Ship's Bells, true, , cs_pair, 100, , , , ship, , os_ships	
Description	Ship's Bells	
Colons	None	
Bling	None	
Chimes	ship*.wav	

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75	75	75	75
B	85 mm	85	85	85	85

Unusual Clock Faces

Face Number	150	
Nuggle file	datetime.nug	
Arguments	Time Only- Resistor Bling, true, , cs_pair, 100, , , , , bs_resistor, 100	
Description	Show the time and light the bling on each dekatron using the resistor colour code.	
Colons	Flashing	
Bling	<ul style="list-style-type: none"> V1 - V6 - colour according to the value shown using resistor colour codes V7 - indicates time Downward is coloured dark blue 	
Chimes	None	

Face Number	151	
Nuggle file	datetime.nug	
Arguments	Decimal Time - minimal bling, decimal, , cs_left, 100, , , , , bs_single, 100	
Description	<ul style="list-style-type: none"> Show the time as hours and the minutes and seconds as a decimal fraction. So 10:30 would be shown as 10:5000. V1 V2 - hours V3 - V6 - fractional hours V7 - indicate Time 	
Colons	The left colon acts as a separator between hours and fractional hours. Flashes.	
Bling	All dark blue colour	
Chimes	None	





Fun Themed Faces³⁵

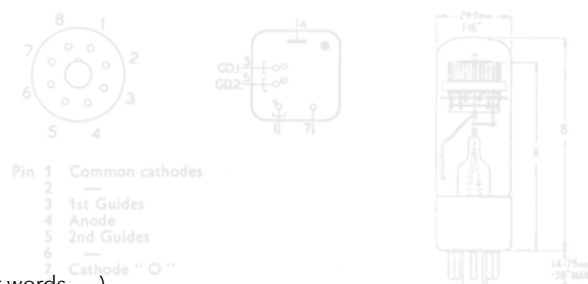
About startrek.nug:

This file takes only one argument: 0 for original Star Trek³⁶ or 1 for Next Generation Star Trek³⁷

Face Number	160	
Nuggle file	startrek.nug	
Arguments	0	
Description	<p><i>Original Star Trek tribute face</i></p> <ul style="list-style-type: none">• Two scientists (blue), two command officers (yellow) and two red-shirts beam down to a planet with purple a landscape and green skies.• Something bad on the planet, well, does what happens to red-shirts...• V1 to V6 show the time• V7 spins	
Colons	None	
Bling	<ul style="list-style-type: none">• V1 V2 - red (security)• V3 V4 - yellow (command officers)• V5 V6 - blue (scientists)• V7, speakers - green (the sky)• Downward - purple (the landscape)	
Chimes	<ul style="list-style-type: none">• ost_*.wav for chimes, credits, transporter noise and tolls• laser0.wav for a gun noise	
IR/Encoder Controls	> ' (play) do the job yourself	



Face Number	161	
Nuggle file	startrek.nug	
Arguments	1	
Description	Star Trek Next Generation tribute <i>Incomplete</i> (but has some functionality)	





³⁵ Well, we think they're fun....


³⁶ https://en.wikipedia.org/wiki/Star_Trek (if you have to look at this, then we're lost for words.....)

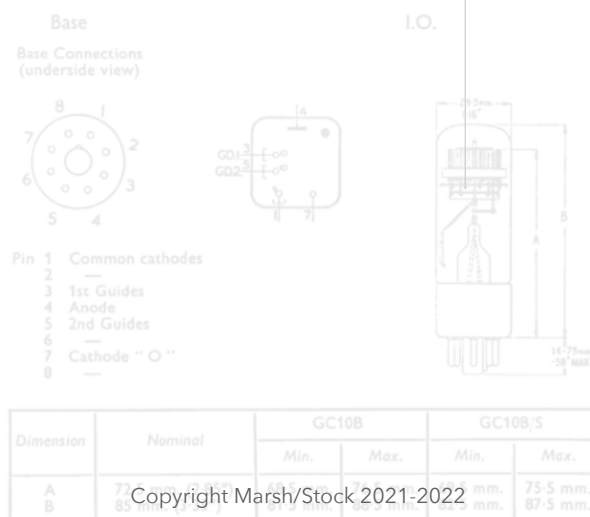
³⁷ https://en.wikipedia.org/wiki/Star_Trek:_The_Next_Generation

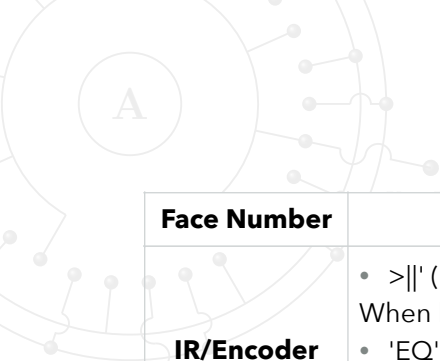
Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

Face Number	170	
Nuggle file	elvis.nug	
Arguments	None	
Description	Elvis inspired clock face Incomplete (but has some functionality)	


Face Number	180	
Nuggle file	starwars.nug	
Arguments	None	
Description	Star Wars tribute face.	
Colons	Flashing	
Bling	<ul style="list-style-type: none">• Normally rotating blue/green hues.• Blue/Green and Red during a duel with white flashes when the sabres collide!• Orange during face shutdown.	
Chimes	sw*.wav	
IR/Encoder Controls	> ' (play) will start the light sabre duel!	

Face Number	190	
Nuggle file	holygrl.nug	
Arguments	None	
Description	<ul style="list-style-type: none"> Tells the Holy Grail story. An excerpt is played each minute. The story can be started, stopped and restarted. Takes about 3 hours to tell the whole story. Story status is saved to NVR so the face will restart where it was left. V1 to V6 show the time. V7 orbits slowly. 	
Colons	Dim when the story is halted, bright and flashing when the story is being told.	
Bling	Slowly rotating hue varied bling across dekatrons V1 to V6 and downwards.	
Chimes	hg*.wav	

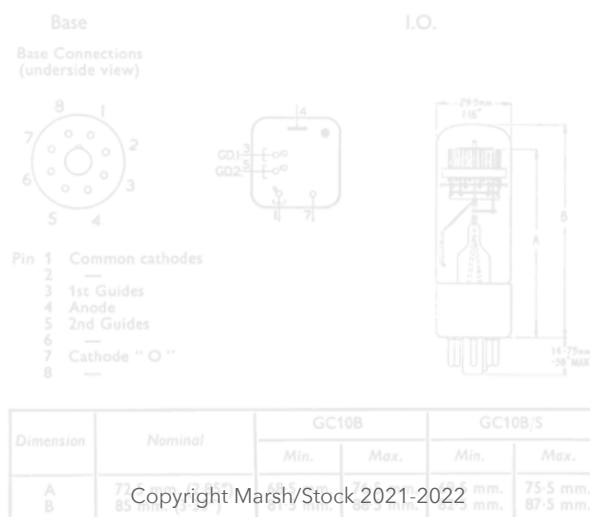




Face Number	190	
IR/Encoder Controls	<ul style="list-style-type: none"> • '> ' (play) will stop and start the story <p>When being told:</p> <ul style="list-style-type: none"> • 'EQ' will restart the story • ' <<' (prev) will replay the last excerpt. • '>> ' (next) will move onto the next excerpt. 	

Face Number	191	
Nuggle file	black.nug	
Arguments	None	
Description	<ul style="list-style-type: none"> • The Black Knight and King Arthur Fight! • <i>On the hour or on demand!</i> • V1 to V6 show the time normally and V7 indicates Time. During the fight the date is shown and V7 highlights Date. 	
Colons	Flashing	
Bling	Starting green and turning red during the fight. Things do get a bit messy.	
Chimes	<ul style="list-style-type: none"> • hg059.wav for the introduction. • hg*.wav for 15 minute "chimes" • clipclop.wav for the second tick noise 	
IR/Encoder Controls	<ul style="list-style-type: none"> • '> ' (play) will provoke the combatants! • ' <<' (prev) will silence the 'clip-clop' second tick • '>> ' (next) will play the 'clip-clop' second tick 	


Notes - The Black Knight fight clip-clop status is stored in NVR and will be restored when the face is shown again.

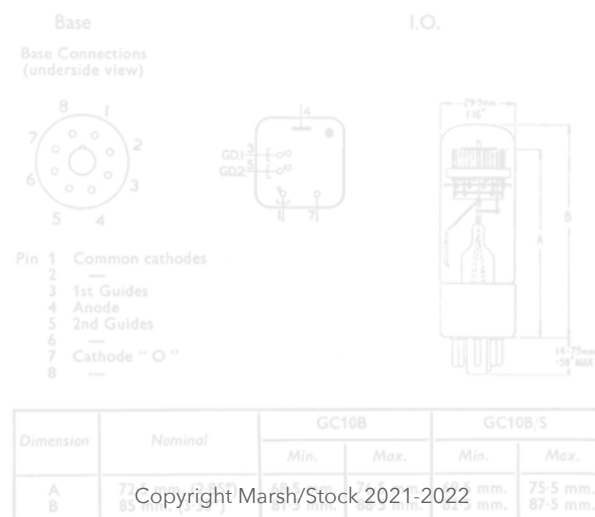


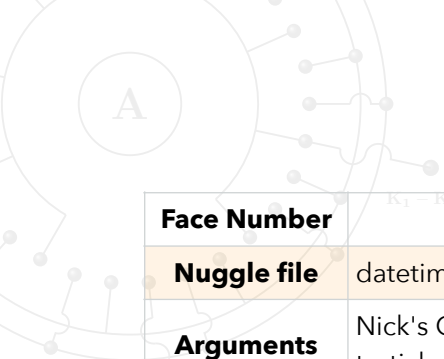
Face Number	192	
Nuggle file	holyclck.nug	
Arguments	None	
Description	<ul style="list-style-type: none"> This face uses Holy Grail excerpts for quarterly and hourly chimes and hour tolls. V1 to V6 show the time normally and V7 indicates Time. 	
Colons	Flashing	
Bling	Rotating rainbow hues	
Chimes	<ul style="list-style-type: none"> hgtitle.wav at the start hg*.wav for the 15 minute "chimes" hgt*.wav for randomly selected tolls clipclop.wav for the second tick noise 	
IR/Encoder Controls	<ul style="list-style-type: none"> <<" (prev) will silence the 'clip-clop' second tick '>> ' (next) will play the 'clip-clop' second tick 	


Notes - The Holy Grail Clock clip-clop status is stored in NVR and will be restored when the face is shown again.

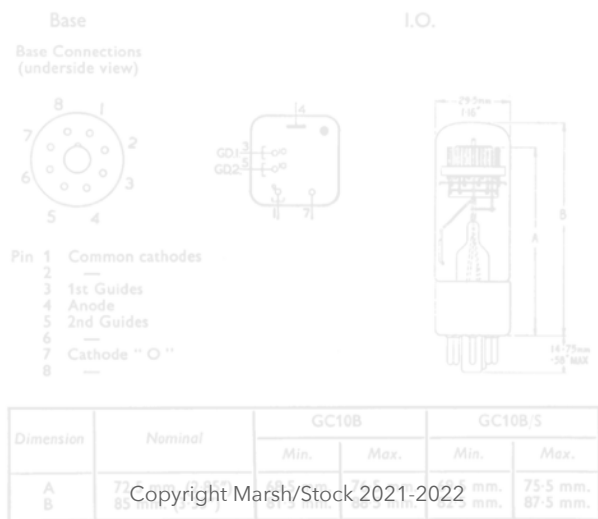
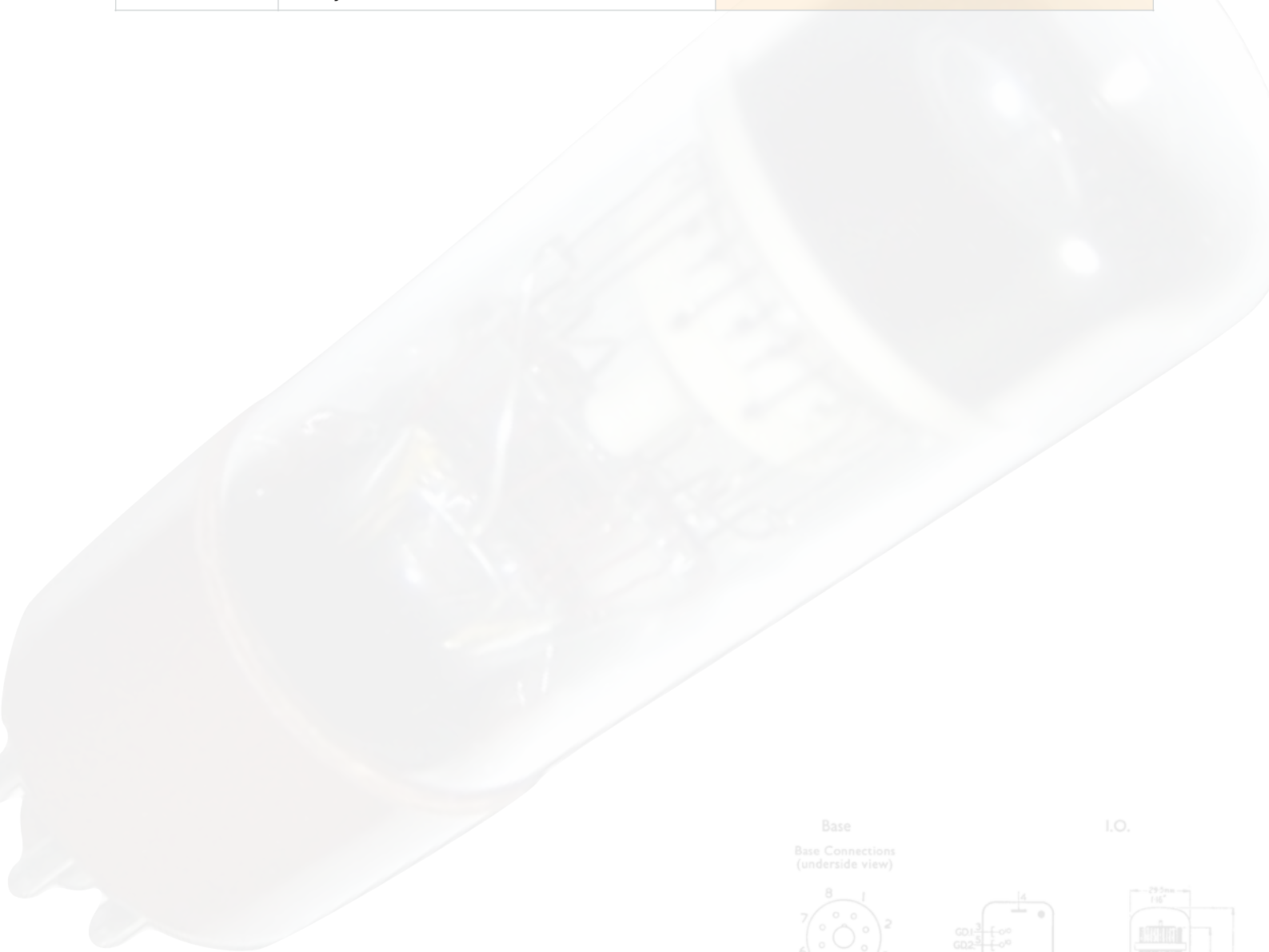
Clock Faces that are blingy-blinged to death (almost) 200 -

Face Number	200	
Nuggle file	datetime.nug	
Arguments	Nick's Clock 1 - Blinged to death, true, true, , , , true, ts_tick, tintan, , os_chime, bs_chase	
Description	Time, date, spinning dekatrons. swirling blue bling, ticks and chimes	
Colons	None	
Bling	Blue swirl	
Chimes	TinTan chimes	





Face Number	201	
Nuggle file	datetime.nug	
Arguments	Nick's Clock 2 - OMG, true, true, , , , true, ts_tick, derby, , os_chime, bs_swirl	
Description	Time, date, spinning dekatrons, swirling, hue changing bling, ticks and chimes	
Colons	None	
Bling	Hue varying swirl	
Chimes	Derby	





Astronomical Faces 300 -

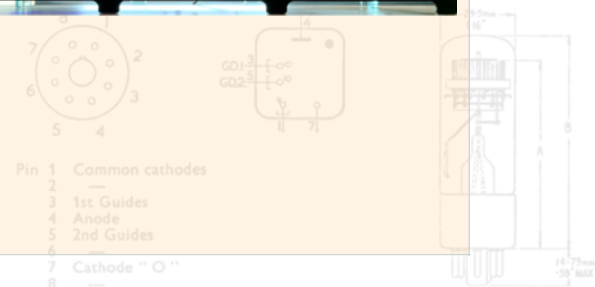
About sun.nug: sun.nug has only one argument:

1. Display sunrise time.
2. Display sunset time.
3. Current sun position, azimuth and altitude.

K₀

Face Number	300	
Nuggle file	sun.nug	
Arguments	1	
Description	<ul style="list-style-type: none"> • Sun rise time on dekatrons V1 - V4 as HH:MM. V5 and V6 show 00. • Bling colour is used to show if the sun does not rise. • Dekatron V7 indicates Time and Sun. 	
Colons	Left colon is lit if HH:MM is displayed	
Bling	<ul style="list-style-type: none"> • If the sun does rise this day then yellow, purple and blues are used to light the bling. • If the sun stays below the horizon all day then dark blue lights the bling. • If the sun stays above the horizon all day then yellow lights the bling 	
Chimes	None	

Face Number	301	
Nuggle file	sun.nug	
Arguments	2	
Description	<ul style="list-style-type: none"> • Sun set time on dekatrons V1 - V4 as HH:MM. V5 and V6 show 00. • Bling used to show if sun does not rise. • Dekatron V7 indicates Date and Sun. 	
Colons	Left colon is lit if HH:MM is displayed	
Bling	<ul style="list-style-type: none"> • If the sun does rise this day then yellow, purple and blues are used to light the bling. • If the sun stays below the horizon all day then dark blue lights the bling. • If the sun stays above the horizon all day then yellow lights the bling 	
Chimes	None	



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Face Number	302	
Nuggle file	sun.nug	
Arguments	3	
Description	<ul style="list-style-type: none"> Sun position in the sky as Azimuth (0 - 359 degrees) and Altitude (-90 to +90 degrees) from the observer's position. The display is updated once per minute. V1 to V3 show the azimuth. V4 shows the count towards the next update i.e. it orbits one per minute. V5 and V6 show the sun's altitude, the sign is shown using the bling. V7 indicates Long and Sun 	
Colons	None	
Bling	<ul style="list-style-type: none"> If the sun does rise this day then yellow, purple and blues are used to light the bling. If the sun stays below the horizon all day then dark blue lights the bling. If the sun stays above the horizon all day then yellow lights the bling 	
Chimes	None	



Games 350 -

Face Number	350	
Nuggle file	pir.nug	
Arguments	None	
Description	<ul style="list-style-type: none"> Start the game and the walk towards the clock trying not to let the PIR see you if it does then the colons will light and the number of times you have been seen will be counted on the tubes. Bling will react and a chime will sound. V1 - V6 count the number of hits. V7 spins slowly and reverses direction each time you are seen. 	
Colons	Dimly lit normally, bright when the PIR sees you.	
Bling	Normally green but V7 and the speaker bling lights red when the PIR sees you.	
Chimes	uhoh.wav when the PIR sees you	




Pin 1 Common cathodes
 2 —
 3 1st Guides
 4 Anode
 5 2nd Guides
 6 —
 7 Cathode "O"
 8 —



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

"Screen Saver" Faces 400 -


Face Number	400	
Nuggle file	blank.nug	
Arguments	None	
Description	The flyback is switched off, bling is all off	
Colons	None	
Bling	All bling is black (unlit)	
Chimes	None	

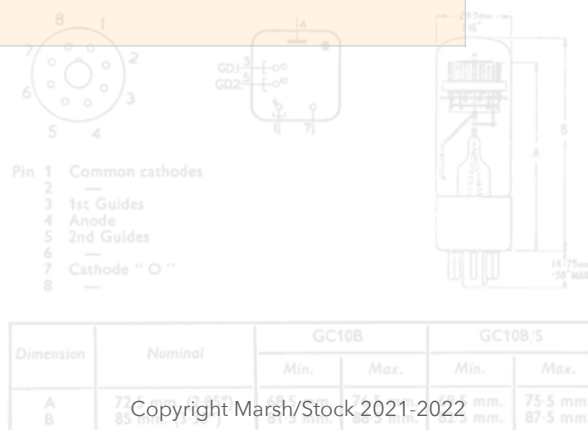
About runner.nug:


These face sets are "screen savers" as they ensure all the cathodes are exercised on all of the Dekatrons. The nuggle file takes two arguments:


Runner spins each tube V1 to V6 and lights their bling which then fades. V7 just spins slowly in time with the spinning moving from tube to tube.


Index	Name	Type	Details
1	Speed	integer	The step rate in mS (default = 1000)
2	Colour	integer	<ul style="list-style-type: none"> Colour values which can be hexadecimal in the form 0xRRGGBB where RR is the red intensity 00 to ff, and so on for GG green and BB blue. Standard colour constants can be used, e.g. yellow (Default is limegreen)

Face Number	401	
Nuggle file	runner.nug	
Arguments	None	
Description	Spin each tube V1 to V6 in turn taking one second to spin each tube. Blinged.	
Colons	None	
Bling	Lime green pulse and fade on V1 - V6, none on V7, downward bling lit	
Chimes	None	



Face Number	402	
Nuggle file	runner.nug	
Arguments	150	
Description	Identical to 401 but much faster	
Colons	None	
Bling	Lime green pulse and fade on V1 - V6, none on V7, downward bling lit	
Chimes	None	

Face Number	403	
Nuggle file	runner.nug	
Arguments	2000, yellow	
Description	Identical to 401 but slower and yellow	
Colons	None	
Bling	Yellow pulse and fade on V1 - V6, none on V7, downward bling lit	
Chimes	None	


Face Number	410	
Nuggle file	picker.nug	
Arguments	None	
Description	Randomly pick a tube and cathode position, randomly light its bling	
Colons	None	
Bling	Randomly selected	
Chimes	None	


About rainbow.nug


A random tube is given a random cathode position. The bling lights in a rotating hue.

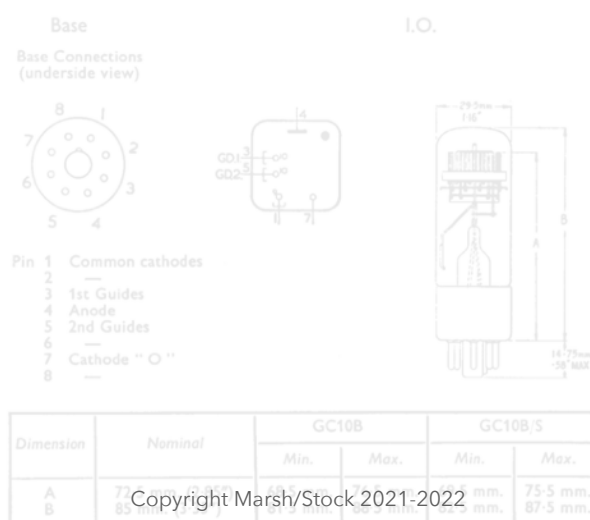
Index	Name	Type	Details	Default
1	Speed	integer	The random change period in mS	1000
2	Inc	integer	Hue change per step	1
3	Span	integer	Hue span across the bling	128
4	Sat	integer	HSV Saturation	255
5	Val	integer	HSV Value	128

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Face Number	420	
Nuggle file	rainbow.nug	
Arguments	None	
Description	Default case	
Colons	None	
Bling	Rainbow hues across the bling	
Chimes	None	

Face Number	421	
Nuggle file	rainbow.nug	
Arguments	, 10	
Description	As 420 but with more rapid hue changes	
Colons	None	
Bling	Rainbow hues across the bling	
Chimes	None	


Face Number	422	
Nuggle file	rainbow.nug	
Arguments	10, , 64	
Description	Fast cathode changes	
Colons	None	
Bling	Slow changing hue and a smaller span	
Chimes	None	



About larson.nug

A "Larson Scanner" ³⁸ is traditionally red and sweeps back and forth. ^{39,40,41}

Index	Name	Type	Details	Default
1	Speed	integer	The step rate in mS	100
2	Mode	integer	0 - Larson fixed colour 1 - Larson rotating hue 2 - Wiggle fixed colour 3 - Wiggle rotating hue	0
3	Colour	integer	<ul style="list-style-type: none"> Fixed mode (0, 2) colour Colour values which can be hex a decimal in the form 0xRRGGBB where RR is the red intensity 00 to ff, and so on for GG green and BB blue. Standard colour constants can be used, e.g. yellow 	Red
4	Fade	integer	Fade rate factor	B0
5	Hue	integer	Starting HSV Hue	Random
6	dHue	integer	Rate at which hue changes per step	10
7	Sat	integer	HSV Saturation	255
8	Val	integer	HSV Value	240

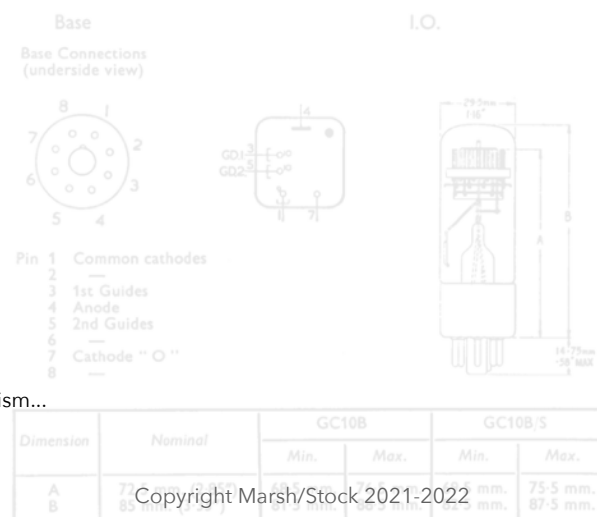
Face Number	430	
Nuggle file	larson.nug	
Arguments	None	
Description	Traditional Larson Scanner	
Colons	None	
Bling	Red	
Chimes	None	


³⁸ https://en.wikipedia.org/wiki/Glen_A._Larson


³⁹ [https://en.wikipedia.org/wiki/Cylon_\(Battlestar_Galactica\)#Original_Cylons](https://en.wikipedia.org/wiki/Cylon_(Battlestar_Galactica)#Original_Cylons)


⁴⁰ <https://en.wikipedia.org/wiki/KITT>


⁴¹ Note the large number of credits - Glen A Larson was accused and fined for plagiarism...



Face Number	431	
Nuggle file	larson.nug	
Arguments	, 1	
Description	Rotating Hue Scanner	
Colons	None	
Bling	Changing hue	
Chimes	None	

Face Number	432	
Nuggle file	larson.nug	
Arguments	25, , yellow, 0xC0	
Description	Fast yellow scanner	
Colons	None	
Bling	Yellow	
Chimes	None	

Face Number	433	
Nuggle file	larson.nug	
Arguments	, 2, lightgoldenrodyellow	
Description	Wiggle mode bright single colour	
Colons	None	
Bling	Lightgoldenrodyellow	
Chimes	None	

Face Number	434	
Nuggle file	larson.nug	
Arguments	, 3, , , 1	
Description	Wiggle Rotating Hue Scanner	
Colons	None	
Bling	Slow changing hue	
Chimes	None	


Pin 1 Common cathodes
 2 —
 3 1st Guides
 4 Anode
 5 2nd Guides
 6 —
 7 Cathode "O"
 8 —





Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75	75	75	75
B	85 mm	85	85	85	85

About technicl.nug:


Technicl.nug takes one argument which is the number of the technical aspects of the clock to display or control. Chimes, the IR handset and the rotary controller are not used.


Face Number	900	
Nuggle file	technicl.nug	
Arguments	1	
Description	<ul style="list-style-type: none"> Clock hardware version number MM.mm is shown on V3 - V6 V7 lights Sys and Time marks 	
Colons	RH colon is lit	
Bling	V3 - V6 lit in green, downward bling lit in olivedrab	
Chimes	None	

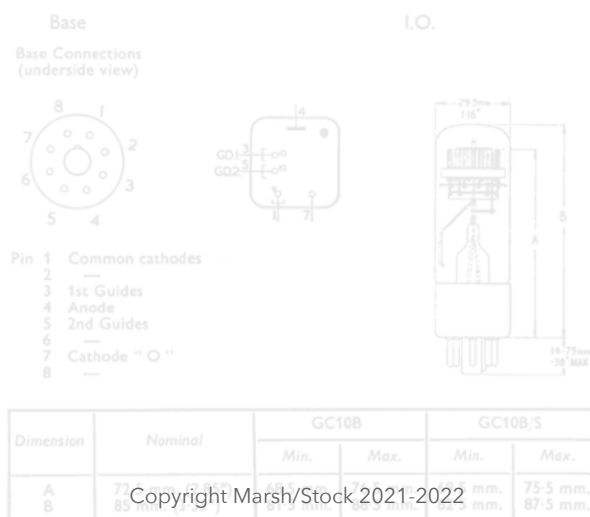
Face Number	901	
Nuggle file	technicl.nug	
Arguments	2	
Description	<ul style="list-style-type: none"> Clock firmware version MM.mm.bb is shown on V1 - V6 V7 lights with Sys and Date 	
Colons	Both colons light	
Bling	All lit using wheat, chocolate and goldenrod	
Chimes	None	


Face Number	910	
Nuggle file	technicl.nug	
Arguments	3	
Description	<ul style="list-style-type: none"> Latitude (from locale.ini) is shown on V3 - V6 (range 0 - 90 degrees) V7 lights with Lat 	
Colons	RH colon is lit	
Bling	<ul style="list-style-type: none"> V3 - V6 are lit in green V7 is lit in red if the location is north of the equator and blue if south Downward bling are goldenrod 	
Chimes	None	


Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.


Face Number	911	
Nuggle file	technicl.nug	
Arguments	4	
Description	<ul style="list-style-type: none"> Longitude (from locale.ini) is shown on V2 - V6 (range 0 -180 degrees) V7 lights with Long 	
Colons	RH colon is lit	
Bling	<ul style="list-style-type: none"> V2 - V6 are lit in green V7 is lit in red if the location is east of Greenwich and blue if west Downward bling are goldenrod 	
Chimes	None	

Face Number	912	
Nuggle file	technicl.nug	
Arguments	5	
Description	<ul style="list-style-type: none"> Elevation (from locale.ini) is shown on V1 - V6 (range 0 -999999m , in theory!) V7 lights with Elev The value shown will be in feet if the user preferred units are set to feet (in locale.ini) otherwise in metres 	
Colons	<ul style="list-style-type: none"> V1 - V7 are lit in greenish blue Downward bling are goldenrod 	
Bling	<ul style="list-style-type: none"> V2 - V6 are lit in green V7 is lit in red if the location is east of Greenwich and blue if west Downward bling are goldenrod 	
Chimes	None	



Face Number	920	
Nuggle file	technicl.nug	
Arguments	6	
Description	<ul style="list-style-type: none"> Chip temperature, the value shown will be °F (range 32 - 212) if the user set preferred units are set to °F (in locale.ini) or otherwise °C (range 0 - 100) V7 shows Sys and Long The display is updated every second. N.B. the results are very odd - on my bug list to review 	
Colons	None	
Bling	<ul style="list-style-type: none"> The chip temperature is used as the hue value for V5, V6 (°C) and V4 - V6 (°F) Downward bling are goldenrod. 	
Chimes	None	


Face Number	921	
Nuggle file	technicl.nug	
Arguments	7	
Description	<ul style="list-style-type: none"> Room brightness read by the LDR (range 0 - 100%) on V4 - V6 V7 shows Sys and Lat 	
Colons	None	
Bling	<ul style="list-style-type: none"> V4 - V6 bling are steel blue Downward bling are goldenrod. 	
Chimes	None	


Face Number	930	
Nuggle file	technicl.nug	
Arguments	8	
Description	<ul style="list-style-type: none"> WiFi module presence, connection and SNTP time validity. V7 shows Sys and Elev 	
Colons	<ul style="list-style-type: none"> Left colon shows if the WiFi module is connected. Right colon shows if the clock's last attempt at a SNTP fetch produced a valid time 	

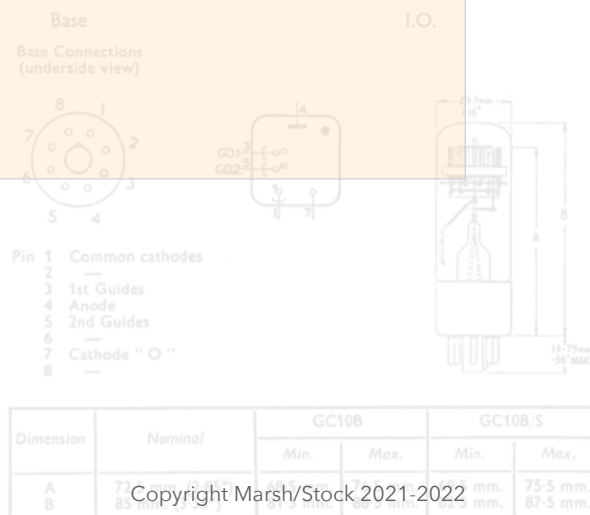
Pin 1 Common cathodes
 2 —
 3 1st Guides
 4 Anode
 5 2nd Guides
 6 —
 7 Cathode "O"
 8 —

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

Face Number	930	
Bling	If there is no WiFi module detected then downward bling are dark blue, if a module has been detected then the downward bling are goldenrod	
Chimes	None	

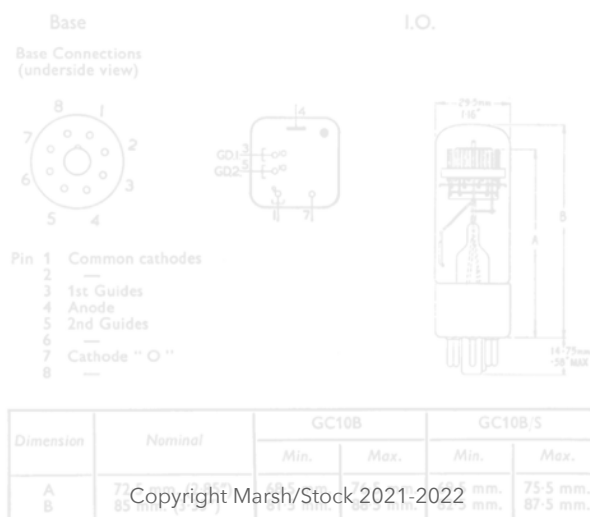
Face Number	940	
Nuggle file	technicl.nug	
Arguments	9	
Description	<ul style="list-style-type: none"> GPS module presence and SNTP time validity. V1, V2 show the number of satellites in view and V5, V6 show the number in use. V7 shows Sys and Elev 	
Colons	Both lit if the GPS data is valid	
Bling	If there is no GPS module detected then downward bling are dark blue, if a module has been detected then the downward bling are goldenrod	
Chimes	None	

Face Number	950	
Nuggle file	technicl.nug	
Arguments	10	
Description	<ul style="list-style-type: none"> Shows the pairing status. V7 shows Pair. V1 - V6 will show the pair number. 	
Colons	None	
Bling	<p>The V7 bling will light:</p> <ul style="list-style-type: none"> Dark blue is pairing is disabled White if enabled but not active Red if enabled but not paired Green if enabled and paired <p>Downward bling are goldenrod.</p>	
Chimes	None	



Date and Time Related

Face Number	960	
Nuggle file	timeset.nug	
Arguments	None	
Description	<ul style="list-style-type: none"> This face can be used to set the RTC UTC time using the IR handset. When run the dekatrons V1 - V6 show the current time but with seconds set to 00. V7 indicates Time. The IR handset is used to change the numbers shown and then accept or cancel the shown time. 	
Colons	None	
Bling	The dekatron currently being edited (V1 - V6) is lit in dark green, the rest are unlit.	
Chimes	<ul style="list-style-type: none"> notify.wav - when accepting the time shown out.wav - when making changes uhoh.wav - on error - trying to set an impossible time 	
IR/Encoder Controls	<ul style="list-style-type: none"> 0' - '9' - use this number on the selected digit and step onto the next, roll over 'T' - the 200+ key cancels the edit and goes to clock face 1 (the blank face) 'P' - the '> ' is also a cancel key '<' - step to the previous dekatron, roll under '>' - step to the next dekatron, roll over '=' - accept the currently displayed time and go to clock face 1 (the blank face) 	



Face Number	961	
Nuggle file	dateset.nug	
Arguments	None	
Description	<ul style="list-style-type: none"> This face can be used to set the RTC UTC date using the IR handset. When run the dekatrons V1 - V6 show the current date. The IR handset is used to change the numbers shown and then accept or cancel the shown date. 	
Colons	None	
Bling	The dekatron currently being edited (V1 - V6) is lit in violet, the rest are unlit.	
Chimes	<ul style="list-style-type: none"> notify.wav - when accepting the date shown out.wav - when making changes uhoh.wav - on error - trying to set an impossible date 	
IR/Encoder Controls	<ul style="list-style-type: none"> 0' - '9' - use this number on the selected digit and step onto the next, roll over 'T' - the 200+ key cancels the edit and goes to clock face 1 (the blank face) 'P' - the '> ' is also a cancel key '<' - step to the previous dekatron, roll under '>' - step to the next dekatron, roll over '=' - accept the currently displayed date and go to clock face 1 (the blank face) 	



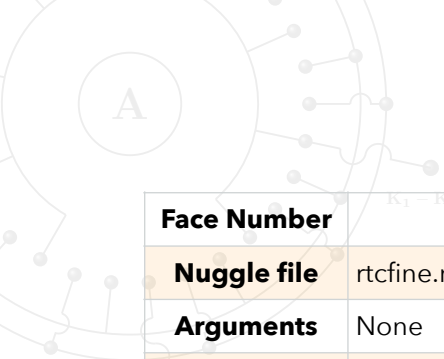
Face Number	962	
Nuggle file	syncrtc.nug	
Arguments	None	
Description	<ul style="list-style-type: none"> Try to sync the clock with the GPS or WiFi module. V7 will indicate Time and Date The clock will then go to face 1 (blank). 	
Colons	None	
Bling	The bling will briefly flash green on success or red on failure	
Chimes	None	
IR/Encoder Controls	None	



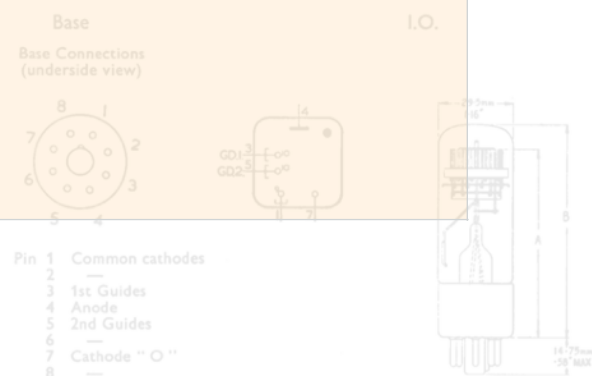
Pin 1 Common cathodes
 2 —
 3 1st Guides
 4 Anode
 5 2nd Guides
 6 —
 7 Cathode
 8 —

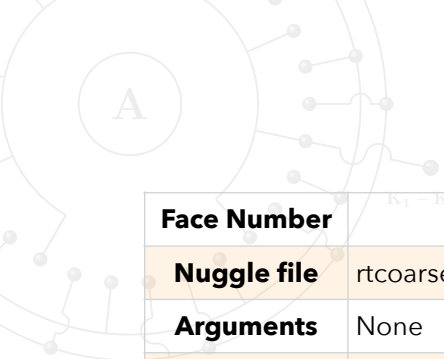


Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

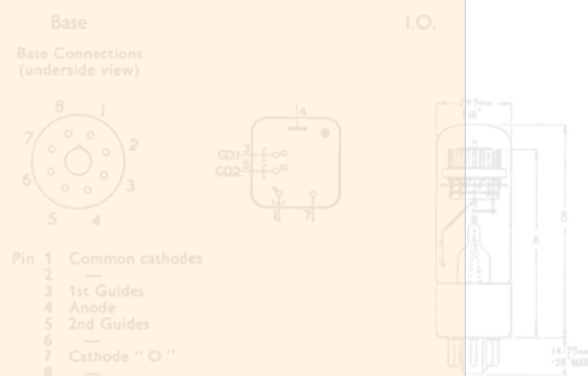


Face Number	963	
Nuggle file	rtcfine.nug	
Arguments	None	
Description	<ul style="list-style-type: none">• This is an experimental face - it is used to make small adjustments to the RTC tick repetition so that when the clock chimes a tick sound it is in time with another "reference clock" tick. The face can delay between 0 and 990mS in 10mS steps.• Operate the face by listening to the two clocks tick and adjust the delay so they tick in unison.• V1 - V4 - current delay in mS• V5 - V6 - seconds count• V7 indicates Sys and Time• Set the tick volume before running this face as the encoder and volume buttons ('+' '-') on the encoder are overridden.• Stop the operation by moving to another face and the last delay value will be preserved in NVR.• Use in conjunction with face 964.	
Colons	None	
Bling	<ul style="list-style-type: none">• V1 - V4 - lit in dark red• V5 - V6 - dark yellow• V7 - dark magenta	
Chimes	tick1.wav is played for audible synchronisation	
IR/Encoder Controls	<p>Encoder -</p> <ul style="list-style-type: none">• rotate right adds 100mS• rotate left deducts 100mS• pushed in and rotate right adds 10mS• pushed in and rotate left deducts 10mS. <p>IR handset -</p> <ul style="list-style-type: none">• '>> ' add 10mS,• ' <<' deducts 10mS• '+' adds 100mS• '-' deducts 100mS• '> ' restore starting delay value	





Face Number		964
Nuggle file	rtcoarse.nug	
Arguments	None	
Description	<ul style="list-style-type: none">This is an experimental face - it is used to make coarse adjustments to the time fetched by the WiFi (using SNTP) of GPS (by parsing NMEA statements). The adjustment can be used to coarsely adjust for different timings and latencies between the clock and a "reference clock". The adjustment can be from -10 seconds to +10 seconds in 1 second steps.Operate the face by watching the seconds dekatrons (V5, V6) and the second hand of your reference clock. Adjust the timings so the Harwell and the reference clock are reasonably close when showing the seconds e.g. when one clock shows 00 so does the other.V1 - V4 - current adjustment in secondsV5 - V6 - seconds countV7 indicates Sys and TimeSet the tick volume before running this face as the encoder and volume buttons ('+' '-') on the encoder are overridden.Stop the operation by moving to another face and the last adjustment value will be preserved in NVR.Use in conjunction with face 963.	
Colons	None	
Bling	<ul style="list-style-type: none">V1 - V4 shows dark red for positive adjustments, black (unlit) for zero adjustment and dark blue for negative adjustmentV5 - V6 show dark yellowV7 shows dark magenta	
Chimes	tick1.wav is played for audible synchronisation	
IR/Encoder Controls	<p>Encoder -</p> <ul style="list-style-type: none">rotate right adds 1Srotate left deducts 1S <p>IR handset -</p> <ul style="list-style-type: none">'+' adds 1S'-' deducts 1S'> ' restore starting delay value	



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75	75	75	75	75
B	85	85	85	85	85



And Finally....

Face Number	999	
Nuggle file	reset.nug	
Arguments	None	
Description	Clock will be given a hardware reset after a 10 second count down	
Colons	None	
Bling	Orange turning black	
Chimes	bark.wav during last 4 seconds	
IR/Encoder Controls	None	

Tests, Aids, Examples 1000 -

Nuggle Tests

This basket of tests perform simple tests on the Nuggle lexer, parser and expression evaluator. They send results to the console only and do not use the Dekatrons. Some tests use the bling and colons. For Dekatron tests and examples see below. These Nuggle tests are all made using test1.nug which takes just one argument which is the number of the test to perform.

Face number	Argument	Purpose
1001	1	Writes "Hello World" to the console in magenta
1002	2	Tests If, Else, IfEnd, Incl, Decl, Succ(), Pred()
1003	3	Test For .. To/DownTo, ForEnd, Break, Continue
1004	4	Test Call, Return , including Call with arguments
1005	5	Test Repeat ... Until, Break, Continue
1006	6	Test While ... WhileEnd, Break, Continue
1007	7	Test Loop ... LoopEnd, Break, Continue
1008	8	Test Beep
1009	9	Test NVR, Push, Face This face runs (or chains to) face 1100 (t_dest.nug) which then uses NVR() and Pop() to retrieve and use the data.
1010	10	Test number functions Abs, Sqr, Sqrt, Min, Max, Pred, Succ, Odd, Even, Encode, High, MidHigh, MidLow, Low, LoCase, UpCase, Random, VerifyDate, VerifyTime, GPIO, NVR, FacePred, FaceSucc, IndexOf
1011	11	Test stack procedures and functions Push, Pop(), Peek(), Empty(), Flush
1012	12	Test Timer, Timer()

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

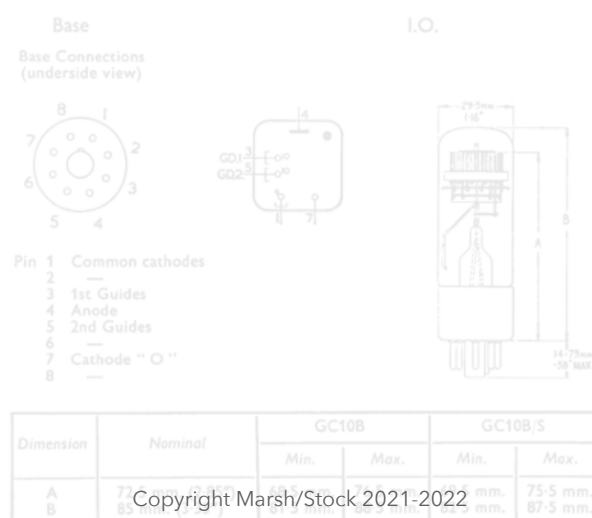
These Nuggle tests are all made using test2.nug which normally takes just one argument which is the number of the test to perform.

Face number	Argument	Purpose
1013	2	Test Jump
1014	3	Test expression evaluator unary+, unary-, ! ~ * / + - << >> & && ^ == != > >= < <==, test precedence
1015	4	Test TimeZone, Restore
1016	5	Test Reset (see note 1)
1017	6	Bling Bling Test Fill, GradientRGB, GradientHSV, Fade, Transmit
1018	7	Test Colon
1019	8	Test Switch, Case, CaseEnd, SwitchEnd
1020	9	Test CallOn, JumpOn
1021	10	String Tests StrCopy, StrWrite, StrLeft, StrRight, StrMiddle, StrCat, StrLen, IndexOf, StrEval, StrPos
1022	11	Argument Test FaceArg (see note 2)
1023	12	Text 8 bit colour palette
1024	13	Text TextColour, TextAttribute
1025	14	Square root Sqrt(-1) of a negative number error test
1026	15	Divide by zero 1 / 0 error test
1030	none	Event testing Event_Start, Event_encoder, Event_IR, Event_PIR, Event_Tick, Event_Fast, Event_Sleep, Event_Leave


Notes:

1. Reset is inhibited in the Nuggle file to prevent accidental triggering. It needs to be enabled before it can be tested. Instructions are in the file.
2. The argument list (in *nuggle.ini*) for these tests is:

11, apple pie, yellow, false, 0x12345689, -42



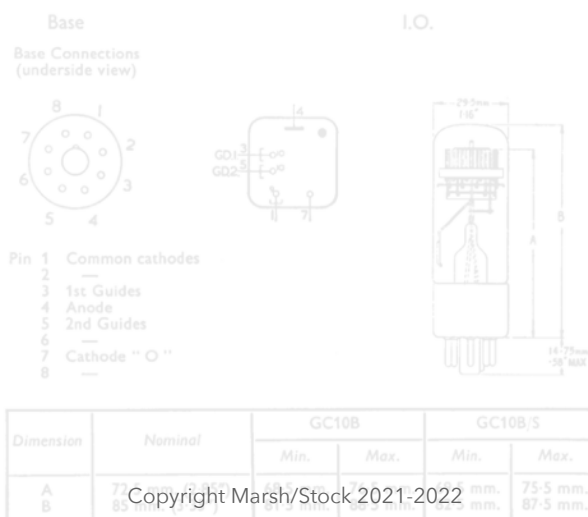
Alignment Aid

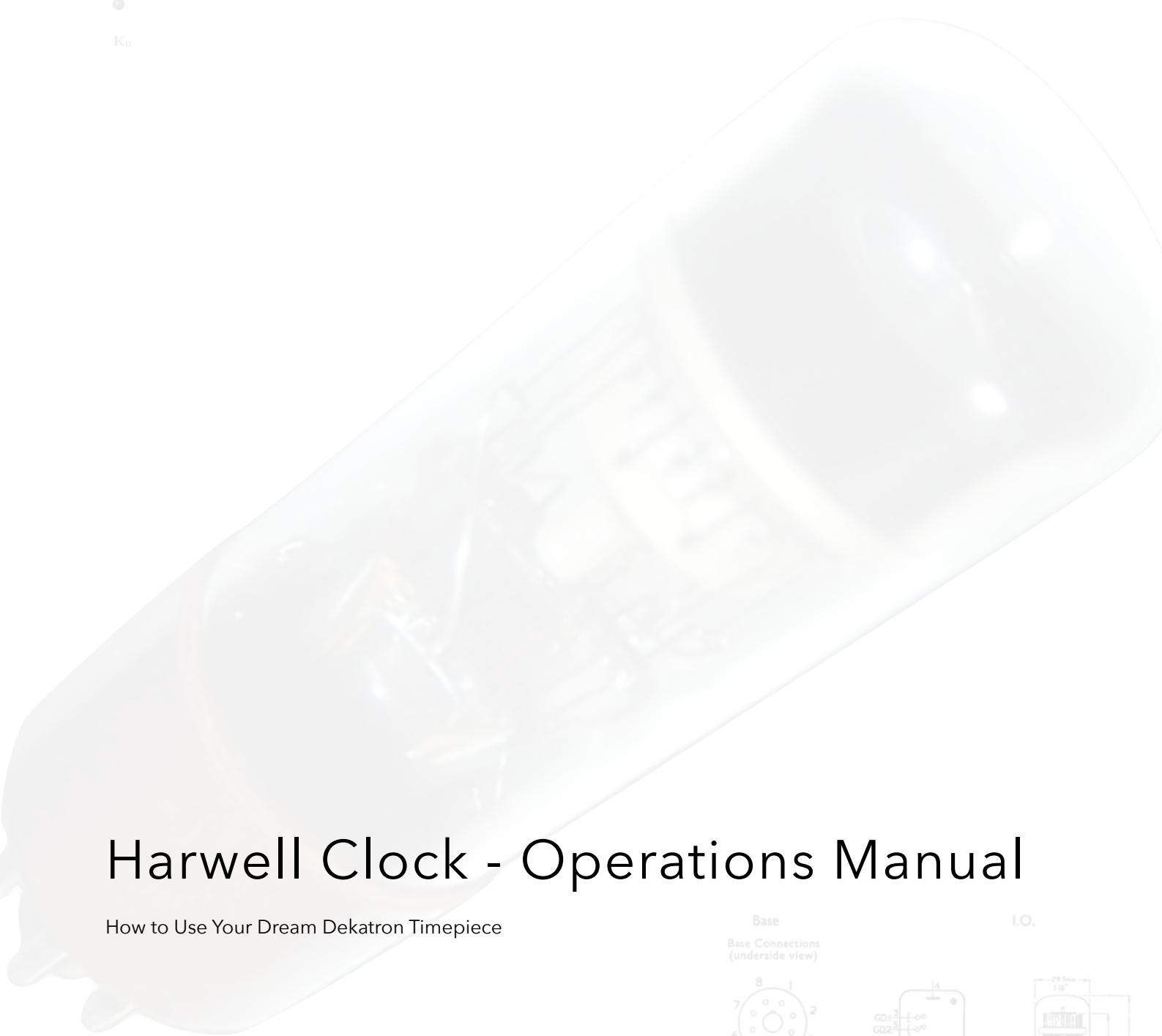
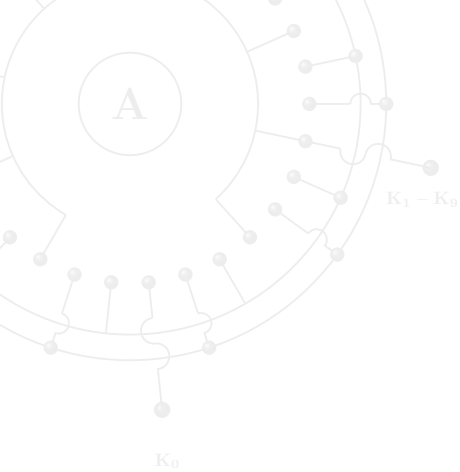
Face Number	1200	
Nuggle file	align.nug	
Arguments	27	
^{K₀} Description	All dots are sent to cathode '0' to assist in tube rotation and clamping so that cathode '0' is at the 12 o'clock position.	
Colons	None	
Bling	None	
Chimes	None	
IR/Encoder Controls	None	

Dekatron Display Test/Example Faces

These all use dekatest.nug using a single argument to select the test to perform.

Face number	Argument	Purpose
1300	1	Zero all the dekatrons
1301	2	Move V1 to position 0, V2 to position 1 etc.
1302	3	Move V1 to cathode 0, V2 to cathode 1 etc.
1303	4	Spinner examples
1304	5	Spinner examples
1305	6	Spinner examples
1306	7	Orbit a random tube once per second
1307	8	Highlight points
1308	9	Show two points on each tube
1309	10	Pendulum examples





Harwell Clock - Operations Manual

How to Use Your Dream Dekatron Timepiece

Base
Base Connections
(underside view)



- Pin 1 Common cathodes
- 2 —
- 3 1st Guides
- 4 Anode
- 5 2nd Guides
- 6 —
- 7 Cathode "O"
- 8 —



I.O.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72 mm (2 7/8")	72 mm (2 7/8")	72 mm (2 7/8")	72 mm (2 7/8")	72 mm (2 7/8")
B	85 mm (3 3/8")	85 mm (3 3/8")	85 mm (3 3/8")	85 mm (3 3/8")	85 mm (3 3/8")

Copyright Marsh/Stock 2021-2022



Harwell Operations⁴²

1. Introduction

This section describes most of the technical aspects of the Harwell Dekatron Clock design and its operation. It also covers the basics on how to modify aspects of the clock to suit your tastes.

Subjects covered:

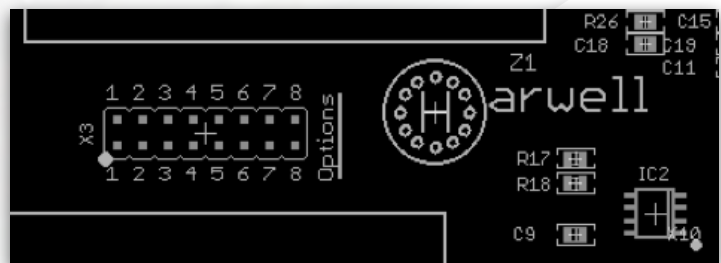
- The main board option jumpers
- Console connection
- Console menus
- Programming the microcontrollers
- Accessing the clock SD card, directories and files
- Adding or changing chime WAV files
- Adding or changing Nuggle files
- Compiling the SAM3X8C firmware
- Infrared remote
- Dekatron 7 version (Annex A)

Stand-alone manuals describe the Nuggle programming language, clock construction from a kit of parts, programming the WiFi module (if used) and others. These can be found on the Dropbox links provided here...

<http://www.sgitheach.org.uk/harwell.html#documentation>

2. Main Board Option Jumpers

There are eight jumpers on the main board (component X3 near the centre of the board near the clock logo) which can select operating options that need to be decided as the clock is initially starting up. Jumpers⁴³ must only be fitted or removed when the clock is switched off.



The options jumpers are numbered 1 to 8 and control the following:

Option jumpers 1 and 2 - These work together giving 4 operational conditions:

Neither jumper fitted: Clock enters the "normal" operation - this will be the common setting.

Jumper 1 only fitted: Start in Test Mode - when fitted the clock starts into "test mode" rather than "normal mode". In test mode most of the high level clock automation is stopped. All of the clock sub-systems like the SD card, the LDR etc. all operate but the sleep system, automatic clock start up etc. are held off. This enables the clock low level systems to be tested

⁴² Everything you wanted to know but dare not ask....

⁴³ Header shorts for those on the other side of the pond....

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

using the console n

Jumper 2 only fitt
operation so new f
the TWI. This is one
made to enter boo

Jumper 2 only fitted: Start the Dekatron card bootloader - the clock will start bootloader operation so new firmware can be uploaded to the MEGA324PA on the Dekatron card using the TWI. This is one of the methods to force bootloader operation, normally the clock can be made to enter bootloader mode using menu commands.

Both jumpers 1 and 2 fitted: Plug-in module pass-through mode - all of the clock's normal operations are stopped except the console is connected directly to the serial port on the plug-in module, i.e. the WiFi module or GPS module. In the case of the WiFi module it enables commands to be typed and sent to the ESP32-BIT and its responses seen. In the case of the GPS module the console shows all the NMEA text data from the module.

Option 3: Permit automatic PSU start up - the 500V/170V flyback HT PSU will normally start up automatically when required. Fitting this jumper allows the PSU to start automatically. This option (of not fitting the jumper) is primarily for safety. The flyback PSU can still be started manually using the console. See the OUTS command.

Option 4: Console colour output: When the jumper is not fitted the text output will use escape sequences to produce a colour highlighted output. If your console program does not support the escape sequences used or if you prefer a B/W display then fit the jumper.

Option 5: SD Card USB remote drive only operation: If you fit this jumper the clock will start up only as a USB remote drive for the SD Card. You can use the clock in this mode without any interference from any clock operation that want to compete for access to the SD card. It is significantly faster with this jumper fitted. Use this mode if you don't own a SD card USB reader.

Option 6: Non-volatile RAM clear: The clock has a small memory chip that is “non-volatile” in that its contents are not lost when the clock is switched off. The chip can be cleared to the default settings by fitting this jumper and starting the clock. Once the clock has started up, the clock can be switched off and the jumper removed.

Option 7: Forced audio mute: Fitting this jumper will force the clock to be entirely quiet. It will not beep or play chimes. The setting cannot be overridden using the IR handset, the rotary encoder or using the console commands. The clock will remain entirely quiet!

Option 8: Verbose scripts: When the clock is starting up and operating normally detailed progress messages will be sent to the console. These messages are different to the Nuggle debug settings in the lexer, parser and expression evaluator.

3. Console Connection

The console is a simple text-based user interface that allows many clock settings to be adjusted and tested. It is an imported debug tool when writing Nuggle programs. The console hardware is also the SAM3X8C programmer.

The SAM3X8C programmer

The programmer is plugged into the Harwell processor plug-in board using a 6-way ribbon cable and IDC connector. Plug the USB mini connector into a computer USB port using a suitable cable. The tests here that rely on the computer assume you are using a PC with Windows 10 installed. I cannot cover other operating systems as I have neither a Linux or MAC system.

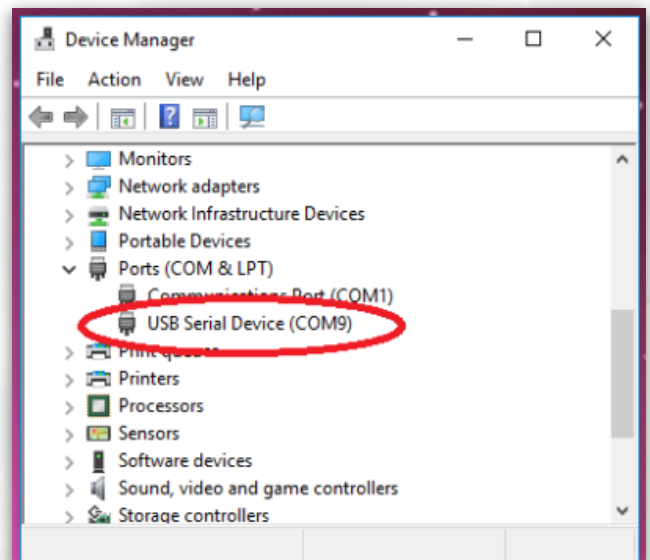
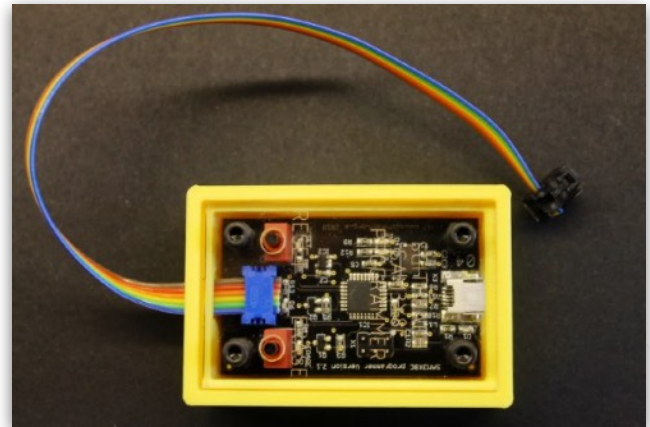
The PC should enumerate the board and a standard serial device interface installed. Look in the Device Manager and under the Ports (COM & LPT) a new USB Serial Device (COMxx) should have appeared. Where COMxx will be replaced by the COM port now assigned. Note the number as it will be needed later. In my case the port number was COM9. If you are uncertain which USB Serial Device has just appeared (you may have others) then disconnect and reconnect the board and look for the device that disappears and then reappears.

Warning! When using the programmer as a console connection be careful not to accidentally press the "Erase" button or else you will erase the flash contents of the SAM3X8C. Not the end of the world but probably a nuisance as you will need to re-flash the SAM.

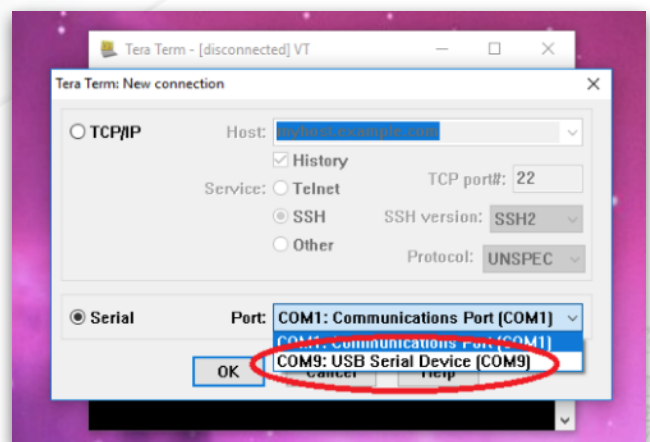
You should also see the power LED D1 next to the USB min connector glowing. If you look carefully, then you will see the SAM connection LED D6 next to the ribbon cable connector glowing as some power bleeds through to it.

Run a serial console program - I often use Tera Term⁴⁴. Set the serial port to the COM number you noted in step 1 above. Configure the serial console to something like 115200, 8, N, 1 with no flow control.

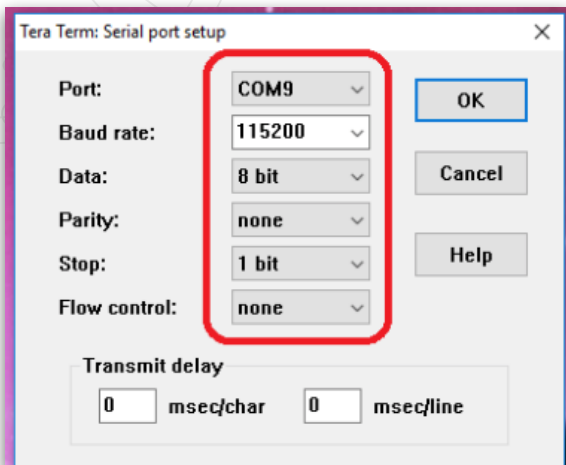
⁴⁴ <http://ttssh2.osdn.jp/>



Device Manager Showing the USB Serial Device (COM9)



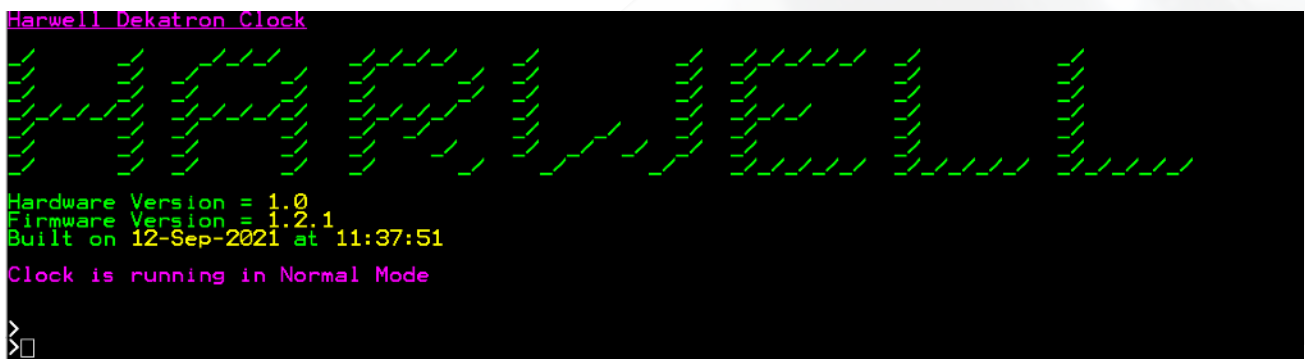
Selecting COM9 Serial Port in Tera Term



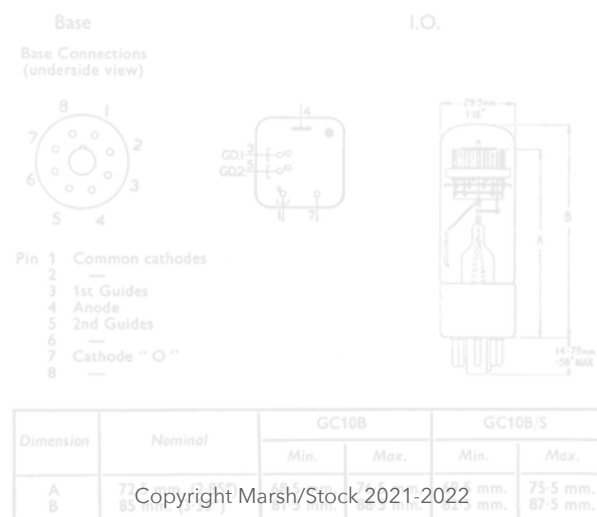
Setting up the Serial Port in Tera Term

When power is applied to the Harwell and the reset button on the clock pressed briefly you should see a simple welcome message⁴⁵. Something like:

Clock Start-Up Welcome Message



⁴⁵ Or a much longer one if the “verbose” jumper 8 is fitted.



4. Console Menus

The console menus are a simple, text-based system to allow commands to be entered that can run Nuggle programs, alter clock settings, interrogate system settings etc.

All commands are four letter keywords like DATG and SYSR. The first three letters are the name (or noun) of the part of the clock that the command acts on and the last letter is the action (or verb) that is to take place.

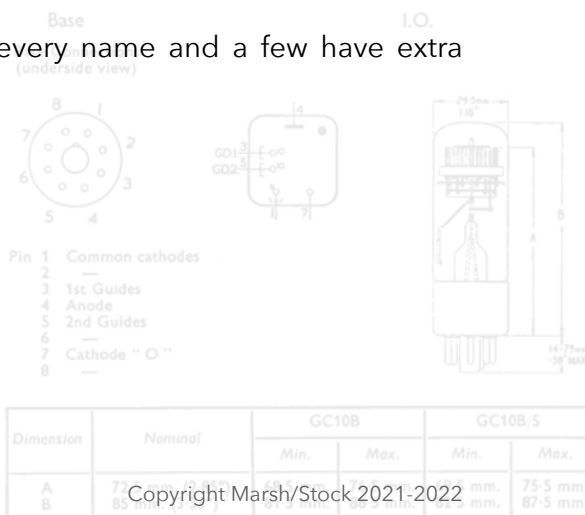
All text can be upper or lower case.

The three letter names/noun are:

ADC	ADC readings and interpretation
BEP	BEeP tests
BOO	BOOtloader operation
CHM	CHiMe system
COL	COLon commands
CON	CONstants interrogation
DAT	clock DATE
DEK	DEKatron tube control
EXP	EXPreSion evaluator
FAC	FACe command help message
GPS	Global Positioning System (if fitted)
IRH	InfraRed Handset
LDR	Light Dependent Resistor (measures ambient light)
LED	LED bling controls
LOC	LOCale reports
NUG	NUGgle commands
OPT	OPTion settings
OUT	digital OUTputs
PIR	PIR controls
ROT	ROTary encoder controls
RTC	Real Time Clock commands
STR	SToRage information
SYS	SYStem information
TIM	clock TIME
VAR	system VARiable value interrogation
VOL	left and right amplifier VOLume controls
WIF	WIFI controls (if fitted)

The single letter actions/verb - not all actions operate with every name and a few have extra action letters:

?	name help
C	Continuous or pass-through mode
G	Get current setting
H	action Help
L	Load data again
K	Kill, turn function off
R	Report





S
U

Set a value
Use the named device (in some way)

For example:

DATG
SYSU
LDR?
LEDK

"get" the current date
"use" the system - a parameter sets how
"help", LDR help information
"kill", turn all bling LEDs off

Some commands have mandatory parameters, for example:

TIMS 12:00:00
OUTS 500,0

"set" the TIME to midday
"set" digital OUTput for the 500V HT (500) supply off (0)

In all cases the 3 letter name with a ? will list the actions for it, together with any mandatory or optional parameters. Typing the command HELP will list the available commands

Command entry

Commands are entered at a '>' text prompt. If the prompt is not there press return key and a '>' should appear. The '>' text prompt changes depending on clock mode. In normal mode it is '>', in test mode '?', in bootloader mode it is '^', and '!' when the Dekatron board is running in demonstration mode (the board "test" jumper is fitted).

The entry editor is very simple. With only the following control keys having a use:

- Backspace (BS) - deletes the last character typed or is ignored at the start of the line
- Escape (Esc) - deletes the whole entry
- Return - enters the command for use

ADC readings and interpretation

The Harwell clock uses four ADC channels to sense the main board hardware version, sense the plug-in board, measure the ambient light level and to report the SAM3X8C core temperature.

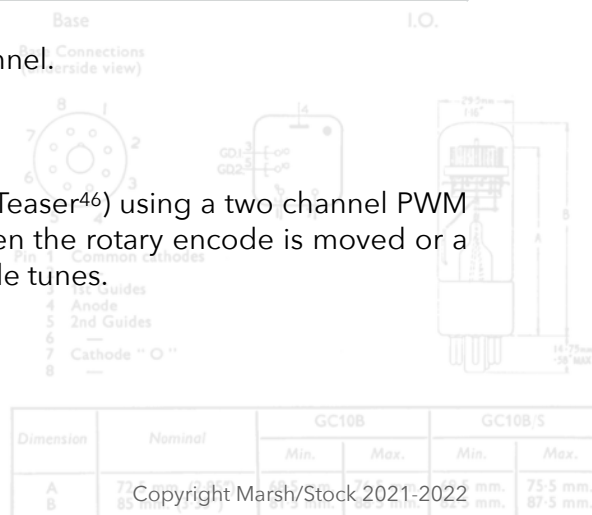
ADC?	ADC command help message
ADCR	Report the current ADC values and offer interpretation of the hardware channels and convert the SAM3X8C core temperature into °C

See also the LDR commands for more information on that channel.

Beep tester

The clock can make beep-beep noises (but the clock is not a Teaser⁴⁶) using a two channel PWM generator. It is used for warnings, making irritating noises when the rotary encode is moved or a key is pressed on the IR handset and can be used to play simple tunes.

⁴⁶ <https://h2g2.com/entry/A100503>



BEP?	BEeP command help message
BEPS <i>c, f, d</i>	Play a beep c = channel 1 = right 2 = left 3 = both f = frequency 30 .. 5000 Hz d = duration 10 .. 5000 mS
BEPU <i>n</i>	Play a pre-defined beep sequence 0 .. 4

See also the VOLume commands.

BOOtloader commands

These commands are used to upload new firmware to the MEGA324PA on the Dekatron board using the TWI.

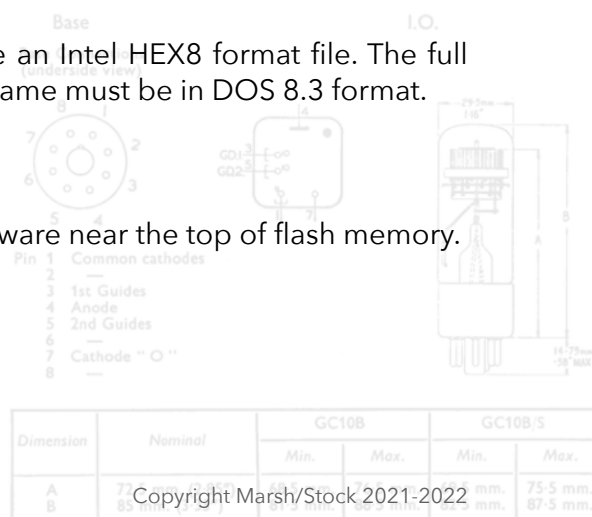
BOO?	BOOtloader command Help message
BOOA <i>file</i>	From normal mode, BOOtloader All will switch to bootloader mode, then test the files for errors, erase the MEGA324PA, then write the file to it, then verify that the written data is correct and then finally switch to normal mode.
BOOD <i>file</i>	Dump the BOOtloader file contents to the console
BOOE	Erase the MEGA 324PA flash memory using the BOOtloader
BOOG <i>addr</i>	Get a single page (128 bytes) using the BOOtloader from flash memory
BOOR	BOOtloader Report
BOOV <i>file</i>	BOOtloader Verify that the firmware is the same as the file
BOOW <i>file</i>	Write the BOOtloader file
BOOY	From normal clock mode switch to the BOOtloader mode
BOOZ <i>file</i>	Check that the gives file an Intel HEX8 file.

All the commands except **BOO?**, **BOOD**, **BOOY** and **BOOZ** require the bootloader to be running. The command **BOOY** does not operate in bootloader mode. Use the **DEKY** command (Dekatron reset) to go from bootloader mode back to normal mode.

The *file* must be in the root folder on the SD card. It must be an Intel HEX8 format file. The full filename must be given including the .hex extension. The file name must be in DOS 8.3 format.

The *addr* must be a decimal number, for example:

BOOG 28672 will list the first 128 bytes of the bootloader firmware near the top of flash memory.



ChiMe system tests

CHM?	CHiMe command help message
CHMK	Kill, stop the current playing chime
CHMR	Will list the available chimes to play with information on each chime
CHMU <i>name</i>	Play the chime by giving its name, quotes and file extension are not used example: chmu strk12

See also the VOLume commands.

COLon commands

COL?	COLon command Help message
COLG <i>n</i>	Get COLon <i>n</i> settings, left colon = 1, right colon = 2
COLK	Kill, turn off, both COLons
COLR	Report all COLon settings
COLS <i>n, b</i>	Set COLon <i>n</i> to brightness <i>b</i> = 0..100

CONstants interrogation

Clock has a number of standard constants and clock specific standard constants held in the *constants.ini* file on the SD card. These can be used in Nuggle programs and in other ini files instead of numbers. The **CON** commands allows you to check these constants.

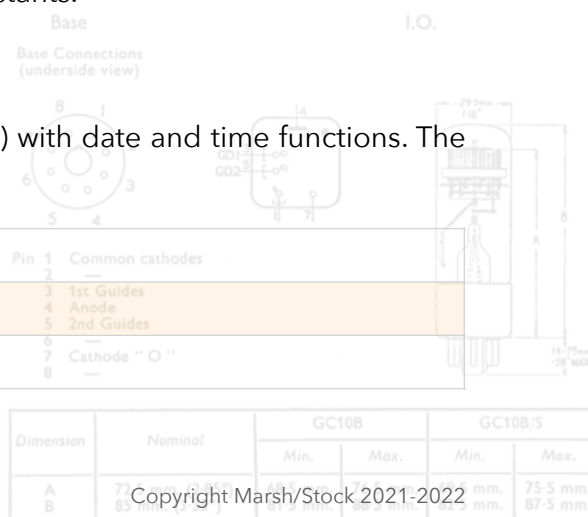
CON?	CONstants command help message
CONG	Will return the value in decimal and hexadecimal of a known constant example: cong red will reply: red = 16711680 0xff0000
CONL	Reload constants file
CONR	Will report the number of constants known and the memory used (heap) to hold them

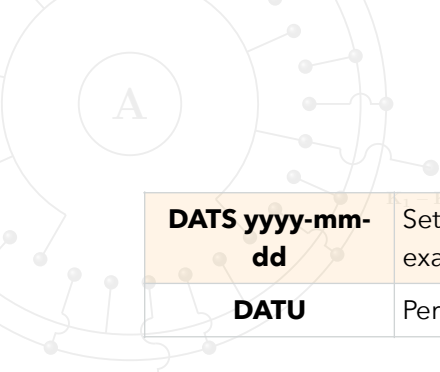
See the Nuggle manual section 2 for more information on constants.

DATE from RTC

The SAM3X8C microcontroller contains a real time clock (RTC) with date and time functions. The **DATE** command allows you to get and set the date.

DAT?	DATe command help message
DATG	Get the current UTC date
DATR	Report the date for UTC, ISO and local.





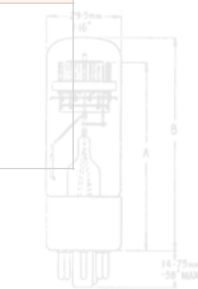
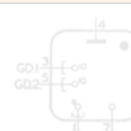
DATS yyyy-mm-dd	Set the UTC date example: datS 2018-1-14
DATU	Perform a synchronisation with the plug-in WIFI or GPS module

See also the **TIME** command.

Dekatron commands

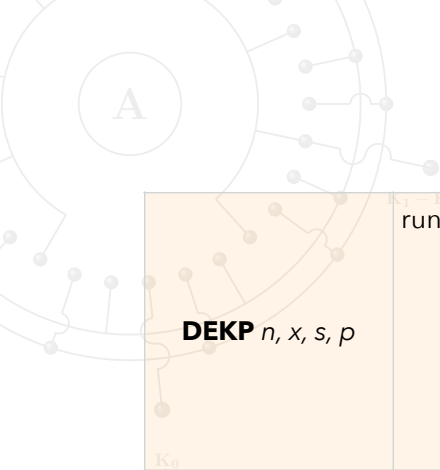
DEK?	Dekatron command Help message
DEKC <i>n, c, d, p</i>	move Dekatron n = tube 0..6 c = cathode 0..9 (0..11 for V7) d = direction 0..3 (see notes) p = pace mS per dot moved
DEKC <i>n, c, d, p</i>	move Dekatron n = tube 0..6 c = cathode 0..9 (0..11 for V7) d = direction 0..3 (see notes) p = pace mS per dot moved
DEKG <i>n</i>	Get Dekatron tube position n = tube 0..6
DEKI <i>n</i>	Idle a Dekatron tube n = tube 0..6
DEKJ	idle all Dekatron tubes
DEKK	Kill, set all DEKatrns to zero quickly
DEKL	reload the driver settings from Dekatron.ini file
DEKM <i>n, x, d, p</i>	Move dot to a new position n = tube 0..6 x = position 0..29 (0..35 for V7) d = direction 0..3 (see notes) p = pace mS per dot moved
DEKN <i>n, x, d, p</i>	spiN a Dekatron tube n = tube 0..6 x = starting dot position 0..29 (0..35 for V7) d = direction 0..1 p = pace mS per dot moved
DEKO <i>n, d, p</i>	spin a Dekatron Once tube n = tube 0..6 d = direction 0..1 p = pace mS per dot moved

Base
Base Connections
(underside view)



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	73 mm	73 mm	73 mm	73 mm	73 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm





DEKP n, x, s, p	run Dekatron Pendulum n = tube 0..6 x = starting dot position 0..29 (0..35 for V7) s = style 10 = linear 20 = gravity p = pace - mS per swing (10..10000)
DEKQ n	move a Dekatron glow to the nearest cathode n = tube 0..6
DEKR	Dekatron Slave Report
DEKS n, x, d, p	Show a value on a Dekatron n = tube 0..6 x = starting dot position 0..29 (0..35 for V7) d = direction 0..1 p = pace mS per dot moved
DEKW $n, c1, c2, p$	show tWo cathodes on a Dekatron n = tube 0..6 c1 = cathode 0..9 (0..11 for V7) c2 = cathode 0..9 (0..11 for V7) p = pause period mS
DEKY	reset the Dekatron driver TWI slave
DEKZ n	Zero a Dekatron n = tube 0..6

Direction codes are

- 0 = forwards, clockwise
- 1 = backwards, anticlockwise
- 2 = shortest route
- 3 = longest route

EXPression evaluator

The Nuggle parser (the part that actually runs the Nuggle program) contains an expression evaluator that will evaluate numerical and boolean expressions. It will use available Nuggle variables if a Nuggle program is running.

EXP?	EXPression evaluator help message	Base	I.O.
EXPG	<p>Enter an expression for evaluation</p> <p>example: expg red >> 10 returns:decimal = 16320 hex = 0x3fc0</p> <p>example:expg red > blue returns:decimal = 1 hex = 0x1</p> <p>Since numerically constant red has a larger value than the constant blue so the value 1, boolean true, is returned</p>		

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

See the Nuggle manual and the Nuggle Harwell Supplemental for more information on Nuggle expressions and the numbers, operators, functions, constants, variables etc. that the clock uses.

FACe Command Help Message

FACG	Return the index number of the current displayed face
FACL	Reload the face index table from [faces] section in <i>nuggle.ini</i>
FACR	Clock face report
FACS <i>n</i>	Change the displayed face to face index number <i>n</i>

GPS plug-in module tests

These command require the GPS module to be installed (obviously).

GPS?	GPS help message
GPSC	Put the console into an operation where all text received from the GPS module is continuously played. Press the Esc key to stop.
GPSR <i>n</i>	GPS report where <i>n</i> = 0 - report data validity, geolocation etc. <i>n</i> = 1 - report on satellite data
GPSU	Hardware reset and restart the GPS Module

IR Handset tests

Operation of the IR handset can be tested.

IRH?	IR Handset help message
IRHC	Put the console into an operation where all IR Handset keypresses are played to the console. Press the Esc key to stop.
IRHR	IR Handset settings report.

LDR sensor controls

Values returned by the LDR can be observed. Useful to test under varying light conditions to set the brightness range parameters.

LDR?	LDR sensor help message
LDRC	Put the console into an operation where the LDR reading is continually reported to the console. Press Esc to stop.
LDRG	Get the current LDR threshold values
LD RR	LDR settings and reading report.



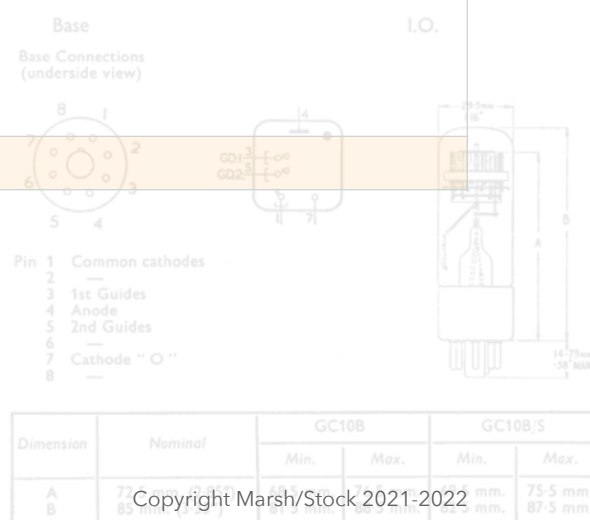
LDRS *l, d*

Change the LDR light and dark threshold values. The values are not retained if the clock is switched of. For a permanent change the [ldr] section in system.ini on the SD card should be edited.

LED bling-bling controls

Try different bling LED module brightness and colour controls. Brightness can be static or dynamic. Colours can be entered as RGB or HSV formats.

LED?	LED bling-bling help message
LEDA <i>fxd</i>	set the bling brightness setting to static level fxd = 1 .. 31 static level not > 10 is recommended
LEDB <i>min,max</i>	set the bling brightness setting to dynamic level using the ambient light level measured by the LDR min = level corresponding to 0% LDR brightness max = level corresponding to 100% LDR brightness
LEDK	kill, turn off all LED modules
LEDR	LED module report
LEDS <i>0, grp, rgb</i>	Set a group of LEDs to the specified RGB colour. Groups and RGB colour values are explained in the Nuggle manual. example: leds 0, 0x3f, 0xff will light the downward pointing LEDs in blue
LEDS <i>1, num, rgb</i>	Set a single LED to the specified RGB colour. example: leds 1, 23, 0xff0000 will light the right-hand colon LED in red
LEDU	Restore default brightness settings.
LEDV <i>0, grp, hsv</i>	Set a group of LEDs to the specified HSV colour. Groups and HSV colour values are explained in the Nuggle manual. example: ledv 0, 0x3f, 0x80ffff will light the downward pointing LEDs in aqua
LEDV <i>1, num, hsv</i>	Set a single LED to the specified HSV colour



LOCaLe reports

Report locale information such as DST, time zone, geolocation and user preferred formats.

LOC?	LOCaLe report help message
LOCR n	Locale reports: $n = 0$ - DST rules showing the selected rule if any $n = 1$ - time zones showing the selected time zone $n = 2$ - geolocation information read from the SD card $n = 3$ - user preferred date and time formatting settings

NUGgle programs

Run, list, stop etc. Nuggle programs.

NUG?	NUGgle program help message
NUGG	List the running Nuggle program from the info file
NUGR	Report on the running Nuggle program
NUGS $name$	Run the Nuggle program name
NUGU	Dump the running Nuggle info and data as a hexadecimal table

OPTion jumpers

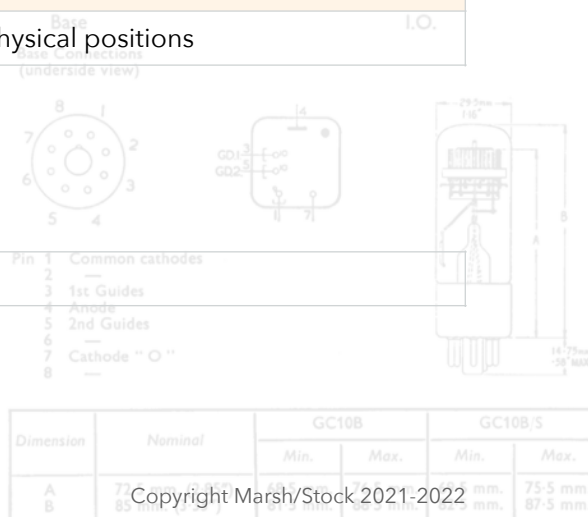
Get, set and report the main board option jumpers. In particular note that the set command can override the physical jumper position so tests can be made without fitting or removing a jumper.

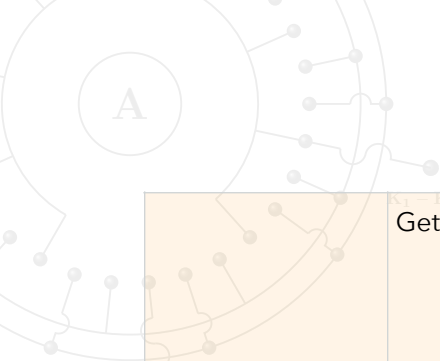
OPT?	OPTion jumper help message
OPTG n	Get the current setting of option jumper n $n = 1 \dots 8$
OPTR	Report position of all jumpers
OPTS n, v	Set the position of option jumper n to v $n = 1 \dots 8$ $v = 0 \mid 1$ this is a firmware setting that overrides the hardware/physical jumpers
OPTU	Restore the firmware settings to the hardware/physical positions

digital OUTput controls

Get, set and report on a limited range digital IO pins.

OUT?	Digital OUTput help message
-------------	-----------------------------





OUTG n	Get the current setting of digital output n $n = 5$ Port A Pin 5 on board back $n = 7$ Port A Pin 7 on board back $n = 10$ Port A Pin 11 on X8 pin 3 $n = 11$ Port A Pin 11 on X8 pin 2 $n = 25$ Port B Pin 25 on board back $n = 28$ Port B Pin 28 on board back $n = 29$ Port B Pin 29 on board back $n = 31$ Port B Pin 31 on board back $n = 500$ Flyback 500V enable
OUTR	Report all settings
OUTS n, v	Set the digital output n to value v $n =$ as above $v = 0 \mid 1$

PIR sensor

Operation of the PIR can be tested.

PIR?	PIR sensor help message
PIRC	Put the console into an operation where all PIR hits are played to the console by printing an exclamation mark. Press the Esc key to stop.
PIRR	PIR sensor report

Real Time Clock commands

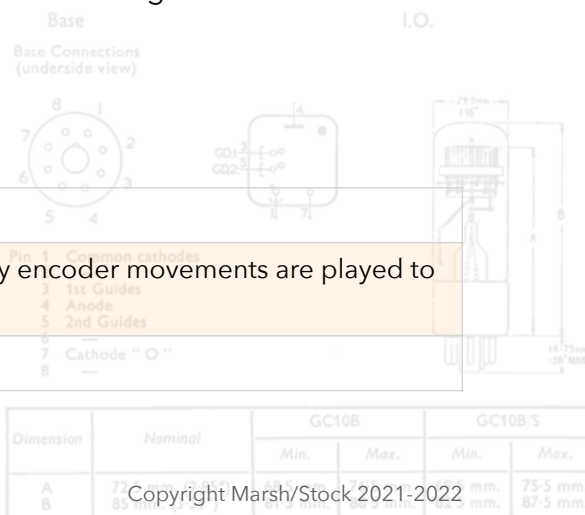
RTC?	RTC command Help message
RTCJ n	Set RTC fine offset
RTCR	RTC Report
RTCU	Use the plug-in (WiFi or GPS) to Sync the RTC
RTCZ	Compare the RTC to the plug-in

The RTC fine adjustment, **RTCJ**, is provided to allow the RTC to be deliberately delayed by up to a second to synchronise the clock to another clock. The process for doing this is in Section ABC.

ROTary encoder tests

Operation of the rotary encoder can be tested.

ROT?	ROTary encoder help message
ROTC	Put the console into an operation where all rotary encoder movements are played to the console. Press the Esc key to stop.
ROTR	Rotary encoder report



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	73 mm				
B	85 mm				



SToRage reports

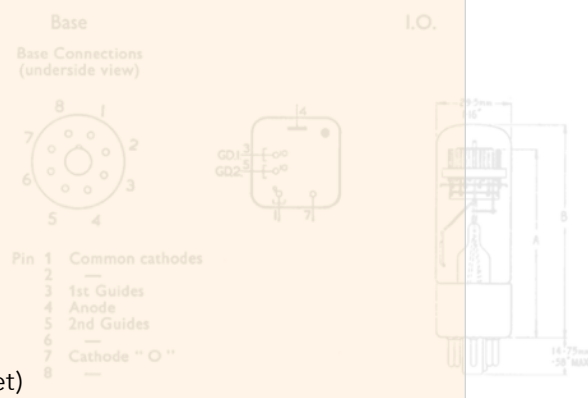
Report on the various memory/storage media in the clock.

STR?	SToRage reports help message
STRR <i>n</i>	Storage report <i>n</i> <i>n</i> = 0 - file system <i>n</i> = 1 - SD Card <i>n</i> = 2 - NVRAM <i>n</i> = 3 - memory - see SYSR 0

SYStem reports and actions

A variety of system reports and actions to invoke.

SYS?	SYStem help message
SYSC <i>n</i>	Continuous ADC readings <i>n</i> = 0 - chip temperature <i>n</i> = 1 - direct LDR reading <i>n</i> = 2 - repeat plug-in reading <i>n</i> = 3 - repeat board reading
SYSR <i>n</i>	System report <i>n</i> <i>n</i> = 0 - memory use <i>n</i> = 1 - SAM3X8C chip identification <i>n</i> = 2 - SAM3X8C core temperature °C <i>n</i> = 3 - task system status <i>n</i> = 4 - demonstrate floating point errors <i>n</i> = 5 - demonstrate random number generation <i>n</i> = 6 - version and build information <i>n</i> = 7 - sleep report <i>n</i> = 8 - profiler report <i>n</i> = 9 - usage log report <i>n</i> = 10 - SYSL command line <i>n</i> = 11 - heap data <i>n</i> = 12 - 256 colour palette <i>n</i> = 13 - text attributes
SYSU <i>n</i>	System actions: <i>n</i> = 0 - invoke sleep <i>n</i> = 1 - wake up <i>n</i> = 2 - shut down the clock (requires reset) <i>n</i> = 3 - reset the clock <i>n</i> = 4 - reload the help file index <i>n</i> = 5 - reload the DST data <i>n</i> = 6 - reload the time zone data <i>n</i> = 7 - reload your geolocation <i>n</i> = 8 - reload formatting preferences <i>n</i> = 63 - fatal error display test (requires reset)



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

TIME from RTC

The SAM3X8C micro-controller contains a real time clock (RTC) with date and time functions. The **TIME** command allows you to get and set the date.

TIM?	TIME help message
TIMG	Get the current UTC time
TIMR	Report the date for UTC, ISO and local.
TIMS <i>hh:mm:ss</i>	Set the UTC time example: tims 12:30:00
TIMU	Perform a synchronisation with the plug-in WIFI or GPS module

See also **DAT** commands.

system VARIABLE report and interrogation

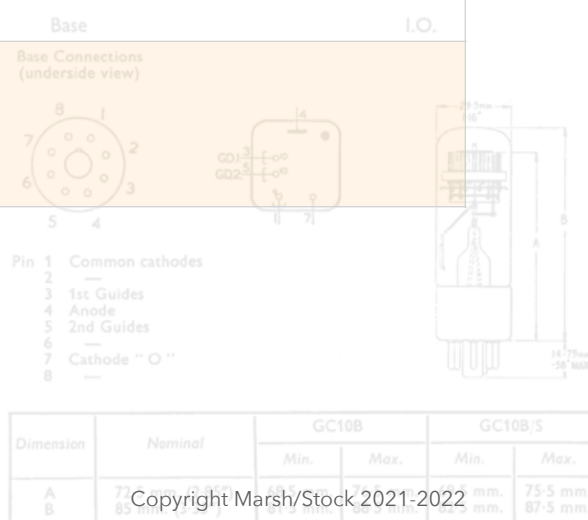
Nuggle programs can access system information using so-called system variables. These values are variable in that the system will change them but the user cannot (at least within a Nuggle program). They are therefore considered read-only.

VAR?	System VARIABLE help message
VARG <i>name</i>	Get the current value of a system variable, in both decimal and hexadecimal example: varg latitude returns: latitude = 57662 0xe13e
VARR	Report system variable usage

amplifier VOLUME control

Get and set the beep and chimes volume control settings. Report on the audio amplifiers.

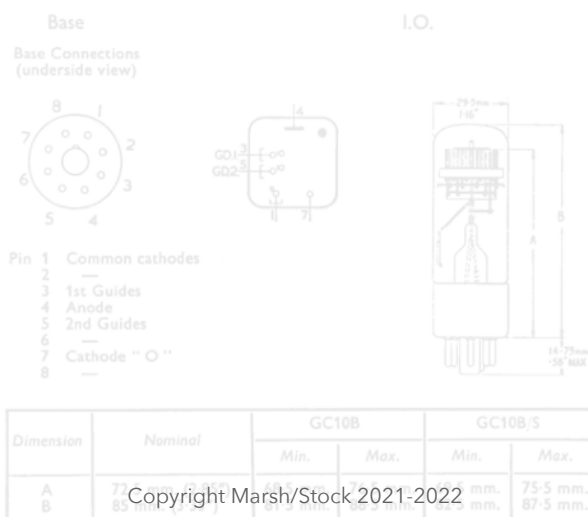
VOL?	amplifier VOLUME control help message
VOLG	get the beep and chimes volume settings
VOLR	report
VOLS <i>c, v</i>	Set the volume for beep or chimes, where n = 0 - beep n = 1 - chimes v - volume setting 0 .. 31



WiFi tests and interrogation

These command require the WIFI module to be installed (obviously). They should be read and used in conjunction with the ESP32-BIT documentation.

WIF?	WiFi help message
WIFG	Get the date and time using SNTP
WIFI command	A send an AT command to the WIFI module.
WIFR	Report on current WIFI settings.
WIFU	Reset the WIFI module.



5. Programming the microcontrollers

5.1 SAM3X8C

This is a simplified and shortened guide to programming the SAM3X8C. For a complete description see the *Sgitheach SAM3X8C Programmer/Serial Console Board* manual. As a programmer the programmer/serial console board uses the BOSSA⁴⁷ programming utility to talk to the boot-loader in the SAM3X8C. There has been some uncertainty over which versions of BOSSA work with which micro-controllers. Therefore it is recommended that you do not download the latest version of BOSSA but use the version you will find on the dropbox as this is known to work with the SAM3X8C. The steps to set up and use the BOSSA programming tool are as follows:

1. Download

Download the SAM3X8C programming files from the dropbox into a suitable directory on your PC (or whatever). You will find a version of BOSSA called bossac.exe and a batch file called samprog.bat. Note the directory name where you have put these files.

2. Using BOSSA directly from a command prompt

From a Command Prompt window you can run the batch file using:

```
\pathto\samprog comxx \pathto\xxxxx.bin
```

where comxx is the COM number that the board is using⁴⁸ and xxxxx.bin is the binary image of the firmware you want to program into the flash.

example:

```
d:\bossa\samprog COM9 d:\harwell\firmware\harwell.1.0.0.bin
```

When the batch file is run you will see the erase LED flash, followed by the reset LED followed by the RxD and TxD LEDs as the binary data is transferred. Note that as the same COMxx connection is used for both the serial connection and the programmer, the programmer will not run while the serial port is connected. Additionally, unplug the SD card USB connection if in use.

3. Update the SD card

Transfer any new files from the SD card image on the dropbox to the clock's SD card. The release notes will tell you which. In a particularly large update a whole new image may be needed which will again require your locale data to be edited. It is recommended that you keep the clock's SD card image backed-up on your PC (or Mac...).

4. Restart the clock

Switch off the Harwell clock and wait a few moments and then power the clock again. The "cold boot" is a requirement of the SAM3X8C.

⁴⁷ <https://github.com/shumatech/BOSSA>

⁴⁸ using the method in section 3 above



The new firmware should now run.

5.2 Programming the MEGA324PA

The Dekatron board of the Harwell clock uses an ATMEGA324PA to control the Dekatrons and provide the large number of I/O lines. It connects to the SAM3X8C microcontroller using a TWI.⁴⁹ The ATMEGA324PA needs to be programmed and there are a couple of methods to accomplish this.

It should be remembered that the kit can come with it pre-programmed with the current firmware. But you may need to be able to program it again to take advantage of any clock firmware upgrades.⁵⁰

A built and fully tested Harwell clock comes with the latest firmware already installed of course.

Quick Start

If you prefer not to read this somewhat long and turgid description of using the bootloader to update the firmware in the ATMEGA324PA then follow these simpler instructions:⁵¹

1. Plug the clock USB port into your PC and the SD card will appear as a removable drive.
2. Copy the new firmware to the root directory of the SD Card. The filename must be in 8.3 format for example *m324.hex*.
3. Plug the console into the clock and a USB cable and check that you can give menu commands.
4. Enter the command **BOOA m324.hex** (using your filename if different) and the clock will update the firmware, a process that takes about 30 seconds.

If no error messages appear then that is job done.

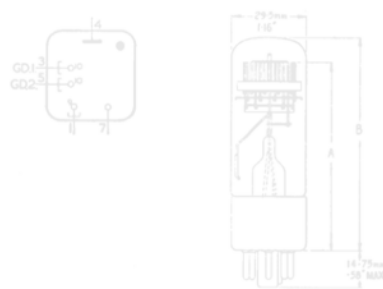
An in depth look at programming the ATMEGA324PA

Method 1: Using an AVR ISP - bootloader not used

If you own an AVR ISP (and know how to use it) then you can use it to burn the ATMEGA324PA firmware and fuses.

The AVR ISP uses the standard 3 x 2 header connection. It is plugged into X3 between V2 and V3 on the Dekatron board.

Base Connections
(underside view)



⁴⁹ Two Wire Interface - AVR speak for I2C.

⁵⁰ Bug fixes etc..

⁵¹ This assumes that the clock has been preloaded with the bootloader.

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm



5.2.1 Set the fuses

Using your favourite AVR ISP software set the fuses to:⁵²

Features

Full Swing Oscillator; Start-up time: 16K CK + 65 ms; Crystal Osc.; slowly rising power; [CKSEL=0111 SUT=11]

☐ Clock output on PORTB1; [CKOUT=0]

☐ Divide clock by 8 internally; [CKDIV8=0]

☐ Boot Reset vector Enabled (default address=\$0000); [BOOTRST=0]

Boot Flash section size=2048 words Boot start address=\$3800; [BOOTSZ=00] ; default value

☐ Preserve EEPROM memory through the Chip Erase cycle; [EESAVE=0]

☐ Watchdog timer always on; [WDTON=0]

☒ Serial program downloading (SPI) enabled; [SPIEN=0]

☐ JTAG Interface Enabled; [JTAGEN=0]

☐ On-Chip Debug Enabled; [OCDEN=0]

Brown-out detection level at VCC=2.7 V; [BODLEVEL=101]

This gives:

Low Fuse - 0xF7
High Fuse - 0xD9
Extended Fuse - 0xFD

For example, if you were using AVRDUDE with a USBTiny⁵³ then you would type the following:

```
avrdude -c USBTiny -P usb -b 19200 -B 8 -p m324p -U lfuse:w:0xf7:m -U hfuse:w:0xd9:m -U efuse:w:0xfd:m
```

5.2.2 Burn the firmware

Then using your favourite AVR ISP software burn the firmware. The file from the distribution dropbox will have a name like "Dekatron.1.2.1.hex", the numbers identifying the version, currently 1.2.1. Of course, the board must be powered when programming.

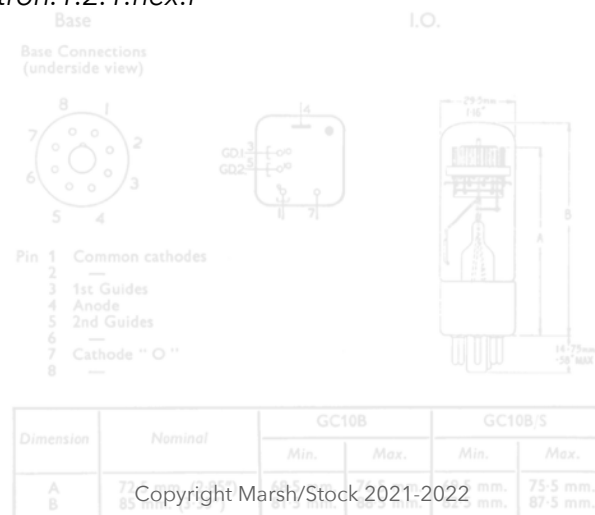
For example:

```
avrdude -c USBTiny -P usb -b 19200 -p m324p -U flash:w:Dekatron.1.2.1.hex:i
```

This completes the programming.

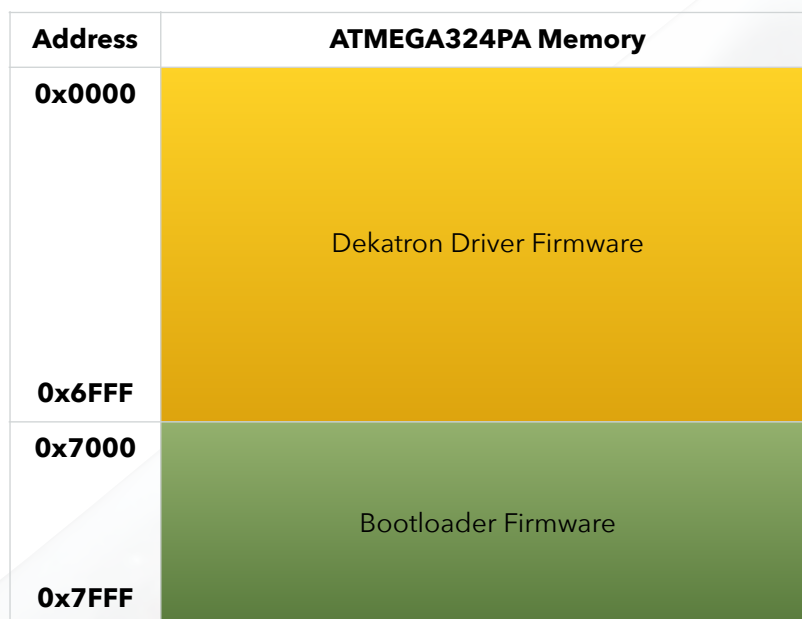
⁵² Image taken from <https://www.engbedded.com/fusecalc/>

⁵³ This is a pretty common arrangement...



Method 2: Using the TWI Bootloader

The TWI bootloader allows the programming or re-programming of the ATMEGA324PA microcontroller using the TWI with the SAM3X8C instead of a traditional programmer used in method 1. Once the bootloader is programmed into the microcontroller, it remains until the chip is erased. The bootloader allows a simple extension to the main clock firmware to download an Intel hex file into the microcontroller. The bootloader firmware sits in the top 4K of the flash memory leaving the remaining 28K for the Dekatron driver firmware:



Under normal conditions both the driver and bootloader firmware sections are present in the flash memory with this method. When the clock is started (or reset) the bootloader firmware in the ATMEGA328PA is first run. It determines if it is desired and safe for the driver firmware to run. If that is the case then execution is switched to the driver firmware and the bootloader has no further role in clock operation.

Why might it not be desired to run the driver firmware? The commonest case is when the driver firmware is to be updated. Why might it not be safe? When the part is brand new the driver firmware section is completely erased (all memory locations are set to 0xFF). Therefore, it would not be good to try and execute a program that is not there!

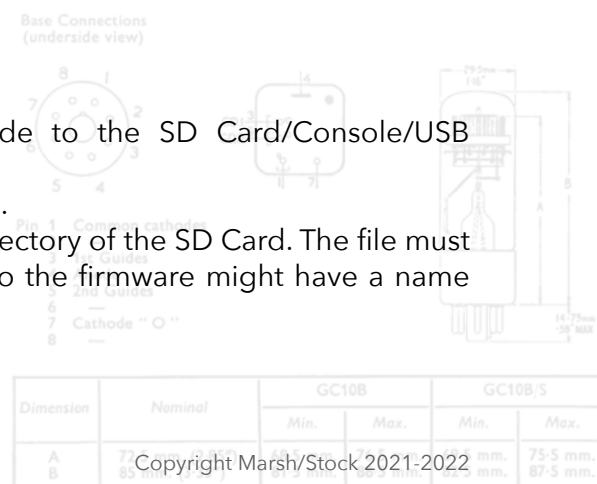
5.3. Using the Bootloader to update the driver firmware

This will be the commonest case so let's discuss it first. Other cases are variations on this one.

The pre-requisites are:

- Functioning Harwell Clock powered and access made to the SD Card/Console/USB connectors.
- Console connected so menu commands can be entered.
- The new driver firmware has been copied to the root directory of the SD Card. The file must be an Intel hex⁵⁴ format file saved with an 8.3 name. So the firmware might have a name

⁵⁴ Strictly an I8HEX format file. https://en.wikipedia.org/wiki/Intel_HEX



such as m324.hex but is just the distribution file, for example, Dekatron.1.2.1.hex rename and copied to the SD card.⁵⁵

The SD card root directory will look something like this:

Name	Date modified	Type	Size
CHIMES	03/10/2020 18:28	File folder	
include	23/05/2020 16:59	File folder	
nuggle	23/05/2020 16:59	File folder	
RUNTIME	31/12/2006 23:00	File folder	
system	23/05/2020 16:59	File folder	
wav	23/05/2020 16:58	File folder	
m324.hex	28/07/2021 10:34	HEX File	25 KB
system.ini	29/11/2020 15:33	INI File	15 KB

The bootloader console menu commands all start with **BOO**.⁵⁶ Many of the commands will only be accepted when the bootloader is running. Two commands will operate in normal clock mode and in bootloader mode:

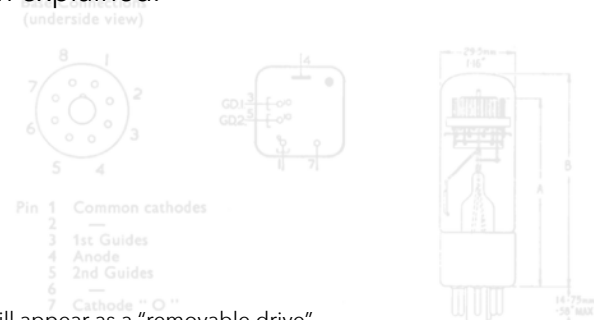
BOO?	Simple help message
BOOA filename	All function executed: <ul style="list-style-type: none"> • switch to bootloader mode (BOOY) • check file (BOOZ filename) • erase flash (BOOE) • write file to flash (BOOW filename) • verify flash and file match (BOOV filename) • switch to normal mode (DEKY)
BOOD filename	Display the contents of the hex file
BOOZ filename	Read and test the given hex filename. The file is tested against the expected I8HEX format

In these instructions, it is assumed that the filename is m324.hex.

BOOA is a wrapper command that executes a series of individual commands in sequence to use the bootloader function. This is the way the bootloader is normally expected to be used. It is easier to understand once the individual commands have been explained.

⁵⁵ You can copy the file using a USB cable connected to your PC where the SD Card will appear as a "removable drive".

⁵⁶ Don't be scared by this....Boo!



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

Using the other three commands on the console gives:

```
COM31:115200baud - Tera Term VT
File Edit Setup Control Window Help
>boo?
Bootloader commands
  BOO?      this help message
  BOOA file test, erase, write and verify in order
  BOOD file display hex file contents
  BOOE      erase flash
  BOOG addr read a page (128 bytes) of memory
  BOOR      bootloader report
  BOOV file verify flash against file
  BOOW file write file to flash
  BOOY      enter bootloader mode
  BOOZ file test hex file formatting
All commands except BOO?, BOOD, BOOY & BOOZ require the bootloader to be running
Command BOOY does not operate in bootloader mode. Use DEKY (dekatron reset)
to enter normal mode.

>booz m324.hex
Intel Hex file formatting is Ok

>bood m324.hex
0000 22 c1 00 00 3d c1 00 00 3b c1 00 00 39 c1 00 00 " . . . = . . . : . . . 9 . . .
0010 37 c1 00 00 35 c1 00 00 33 c1 00 00 31 c1 00 00 7 . . . 5 . . . 3 . . . 1 . . .
0020 2f c1 00 00 2d c1 00 00 2b c1 00 00 29 c1 00 00 / . . . - . . . + . . . ) . . .
0030 27 c1 00 00 25 c1 00 00 23 c1 00 00 21 c1 00 00 ' . . . % . . . # . . . ! . . .
0040 0c 94 4d 0b 1d c1 00 00 1b c1 00 00 19 c1 00 00 . M . . . . . . . . . . . . . . .
0050 17 c1 00 00 15 c1 00 00 13 c1 00 00 11 c1 00 00 . . . . . . . . . . . . . . .
0060 0f c1 00 00 0d c1 00 00 0c 94 3b 0c 09 c1 00 00 . . . . . . . . . . . . . . .
0070 07 c1 00 00 05 c1 00 00 03 c1 00 00 e6 02 9e 04 . . . . . . . . . . . . . . .
0080 f0 02 33 03 5a 03 cf 03 0a 04 9e 04 58 04 9e 04 . 3 . Z . . . . . . . . . . .
0090 e6 02 0e 07 10 07 1f 07 21 07 31 07 1f 07 21 07 C . ! . o . ! . 1 . . . . .
00a0 43 07 21 07 6f 07 95 07 21 07 ca 07 f0 07 21 07 . 6 . R . P . C . . . . . g .
00b0 2c 08 36 08 52 08 70 08 7b 08 86 08 5f 0e 67 0e . . . . . . . . . . . . . . .
00c0 98 0e a9 0e 94 0e 9f 0e 80 0e b1 0e b9 0e ca 0e . . . . . . . . . . . . . . .
00d0 df 0e f4 0e 01 0f 13 0f 21 0f 3d 0f 74 0e 58 0f . . . . . . . . . . . . . . .
```

Note that the full filename, including the .hex extension is entered. The **BOOD** output can be very long so is truncated in this screenshot.

To activate the bootloader so it can be used to write new firmware to the flash you use the **BOOY** command.

BOOY	Enter bootloader mode
-------------	-----------------------

Note that the command prompt has changed from a greater than⁵⁷ symbol '>' to a caret '^'. The '^' is used to indicate the console is now in bootloader mode.

Commands that now operate in addition⁵⁸ to **BOO?**, **BOOD** and **BOOZ** are:

```
COM31:115200baud - Tera Term VT
File Edit Setup Control Window Help
>booy
^
```

BOOE	Erase Dekatron driver flash memory area
BOOG <i>addr</i>	Display 128 bytes of flash memory starting at the given (decimal) address
BOOR	Provide a bootloader report
BOOV <i>filename</i>	Verify the contents of the flash memory matches the contents of the Intel hex file

⁵⁷ A '>' is shown here but if the clock is operating in a different mode then the prompt symbol may be different.

⁵⁸ **BOOY** no longer operates as the clock is now in bootloader mode.

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

BOOW filename Write the Intel hex file to flash memory

You can now use **BOOE** to erase the driver flash area, **BOOW** to write a new Intel hex file and then **BOOV** to verify the flash contents against the hex file:

When erase, write and verify operations are underway the LED on the board will flash⁵⁹ and then return to a steady on state when the commands are finished.

Two other commands are **BOOG** and **BOOR**: **BOOG** displays a single 128 byte (a "programming page") from flash memory, for example:⁶⁰

```
COM31:115200baud - Tera Term VT
File Edit Setup Control Window Help

^booz m324.hex
Intel Hex file formatting is Ok

^booe
Erasing...
Erase took 958 mS

^boow m324.hex
Writing...
Write took 15846 mS

^boov m324.hex
Verifying...
Verify took 9361 mS

^
```

```
COM31:115200baud - Tera Term VT
File Edit Setup Control Window Help

^boog 0
0000 22 c1 00 00 3d c1 00 00 3b c1 00 00 39 c1 00 00 " . . . = . . . : . . . 9 . . .
0010 37 c1 00 00 35 c1 00 00 33 c1 00 00 31 c1 00 00 7 . . . 5 . . . 3 . . . 1 . . .
0020 2f c1 00 00 2d c1 00 00 2b c1 00 00 29 c1 00 00 / . . . - . . . + . . . ) . . .
0030 27 c1 00 00 25 c1 00 00 23 c1 00 00 21 c1 00 00 ' . . . % . . . # . . . ! . . .
0040 0c 94 4d 0b 1d c1 00 00 1b c1 00 00 19 c1 00 00 . . . M . . . . . . . . . . . . . . .
0050 17 c1 00 00 15 c1 00 00 13 c1 00 00 11 c1 00 00 . . . . . . . . . . . . . . .
0060 0f c1 00 00 0d c1 00 00 0c 94 3b 0c 09 c1 00 00 . . . . . . . . . . . . . . .
0070 07 c1 00 00 05 c1 00 00 03 c1 00 00 e6 02 9e 04 . . . . . . . . . . . . . . .

^
```

BOOR gives a report on the bootloader and current settings:

Version and *Compiled* are obvious, *LED* and *Test state* are the onboard LED and the test jumper (X4) states, *Jumpers* is the interpretation of the three tube selection jumpers J0, J1 and J2.

```
COM31:115200baud - Tera Term VT
File Edit Setup Control Window Help

^boor
Dekatron Board Bootloader Report
Version: 1.2.0
Compiled: 28 Jul 2021 at 11:17:35
LED state: off
Test state: open
Jumpers: British GC10 and GC12
EEPROM CRC: 0x80
Flash CRC: 0x80
Reason: Magic number found
Signature: 0x1e9511
Fuses: Low:0xf7 High:0xd8 Extended:0xfd

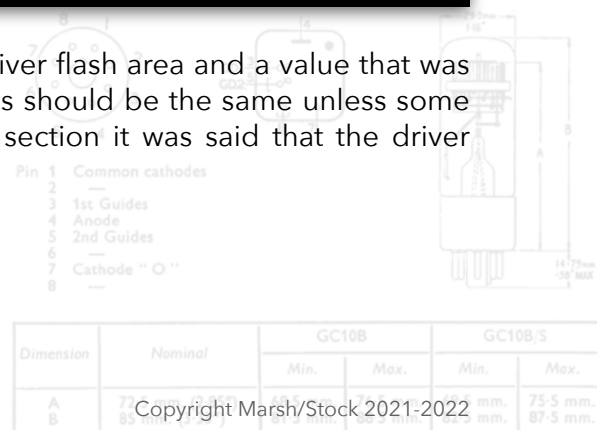
^
```

The two CRC⁶¹ values are a computed CRC for the current driver flash area and a value that was computed when the flash is verified (using **BOOV**). The values should be the same unless some corruption has occurred. In the opening description of this section it was said that the driver

⁵⁹ To try to convince you that something is going on...

⁶⁰ Me thinks the author was enjoying some of these acronyms.... Ed.

⁶¹ https://en.wikipedia.org/wiki/Cyclic_redundancy_check



firmware would be run if it was safe to do so. These two CRC values are, in part, used to determine that safety.

Finally, the signature and fuse are read from the microcontroller. The signature should always be 0x1e9511⁶² and the fuses should be those needed for correct bootloader operation of the ATMEGA324PA (see section 4 below).

This report should confirm what you know already!

Returning to the **BOOA** command. This command wraps all the individual commands in sequence to write the new file to the ATMEGA324PA together with all the test and verify steps. For example:

5.4. What could possibly go wrong?

With something arcane as a bootloader there must be some chance⁶³ of things going wrong. The most likely problems will occur if the *write* phase of the bootloader is interrupted, for example, loss of power to the clock or a problem reading the SD Card.

During bootloader development attempts were made to upset various processes to ensure that clock would recover, albeit with some intervention.



```
COM31:115200baud - Tera Term VT
File Edit Setup Control Window Help

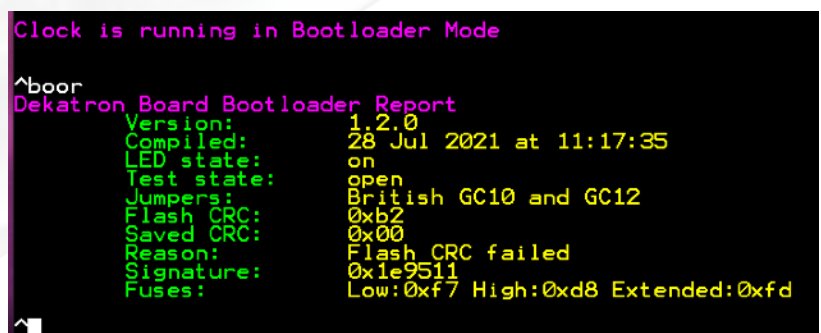
>booa m324.hex
Intel Hex file formatting is Ok
Now in Bootloader mode
Erasing...
Erase took 958 mS
Writing...
Write took 16036 mS
Verifying...
Verify took 9289 mS
Now in Normal mode
>■
```

These are not experiments that you are expected to repeat!

Example: Loss of power.

Tested by switching the clock off during the **BOOA** write phase⁶⁴... On power on again, the clock detected there is a problem with the Dekatron firmware and entered bootloader mode by itself:

You can see here that the reason for entering bootloader mode was that the Flash CRC failed. So it was doing its job correctly.



```
Clock is running in Bootloader Mode

^boor
Dekatron Board Bootloader Report
Version: 1.2.0
Compiled: 28 Jul 2021 at 11:17:35
LED state: on
Test state: open
Jumpers: British GC10 and GC12
Flash CRC: 0xb2
Saved CRC: 0x00
Reason: Flash CRC failed
Signature: 0x1e9511
Fuses: Low:0xf7 High:0xd8 Extended:0xfd
^■
```

Example: Other problems

I ran a version of the bootloader that made it look like that the Dekatron driver flash is corrupted but the stored CRC made it look like the flash is Ok. The "faulty" Dekatron driver made the clock hang.

⁶² The signature of an ATMEGA324PA.

⁶³ "Scientists have calculated that the chances of something so patently absurd actually existing are millions to one. But magicians have calculated that million-to-one chances crop up nine times out of ten." - Terry Pratchett, Mort

⁶⁴ I also tried yanking out the SD card, removing the TWI cable from the main board and anything else I could think of...



There are three ways out of this hole.

1. Set jumper and reboot the clock

It might be possible for the firmware to sort itself out. To try this, remove Option Jumper 1 on the main board and fit Option Jumper 2. Power the clock and if successful clock will restart in bootloader mode and allow you to erase and write the flash again. This method is simple and fairly non-invasive but may not always work.

2. Remove the tube selection jumpers on the Dekatron card

On the Dekatron card the three jumpers J0, J1 and J2 are normally used to select the type of dekatrons in use (British, American, Soviet), however, there is a special setting of all jumpers open (none fitted/closed) that the bootloader recognises as telling it to run.

It is necessary to partly dismantle the clock to remove the jumper(s) using a soldering iron but it is a pretty reliable way to restore a clock from major flash /eeprom corruption.

Having removed all the links (only link J0 in my case as I am using British tubes) the clock restarted into bootloader mode and the **BOOR** command showed:

Here you can see that the CRC values are the same so did not force the clock into bootloader mode. But the reason for the bootloader mode start was *Jumper settings* showing that the bootloader correctly detected that all the jumper J0, J1 and J2 were open. After writing a new bootloader firmware (**BOOW**), verifying the write (**BOOV**), then the required links are reinstated.

This is a pretty drastic process but it should work against some pretty weird failures⁶⁵ or accidents that might occur during the write or verify phases.

```
COM31:115200baud - Tera Term VT
File Edit Setup Control Window Help
^boor
Dekatron Board Bootloader Report
Version: 1.2.0
Compiled: 28 Jul 2021 at 11:17:35
LED state: on
Test state: open
Jumpers: Force Bootloader
Flash CRC: 0x32
Saved CRC: 0x32
Reason: Jumper settings
Signature: 0x1e9511
Fuses: Low:0xf7 High:0xd8 Extended:0xfd
```

3. Use an AVR ISP

However, if you have an AVR ISP but are using the bootloader to avoid the hassle of using the AVR ISP then you will be able to recover from any bootloader corruption by using it to erase the whole of the flash (Dekatron driver and bootloader firmware) and then reprogram the bootloader firmware. See the following section. This will avoid de-soldering the links.

5.5 Programming the bootloader firmware

The £46,172.48 question⁶⁶, if you have spotted it, is how does the bootloader firmware get into the top of the flash memory in the ATMEGA324PA? To do this you must use an AVR ISP. Don't forget to order the kit with the bootloader installed if you don't own an AVR ISP! However, if you

⁶⁵ They are most likely to occur during code development that with finished, tested code. It was not easy at all to get the clock to misbehave such that this measure was required.

⁶⁶ \$64000 at today's exchange rate...

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm			75.5 mm	
B	85 mm			87.5 mm	

do need to install the bootloader firmware (for example you have modified the code or you haven't bought a kit) then the steps are.

5.5.1 Set the fuses

Using your favourite AVR ISP software set the fuses to:

Features

Full Swing Oscillator; Start-up time: 16K CK + 65 ms; Crystal Osc.; slowly rising power; [CKSEL=0111 SUT=11]

☐ Clock output on PORTB1; [CKOUT=0]

☐ Divide clock by 8 internally; [CKDIV8=0]

☒ Boot Reset vector Enabled (default address=\$0000); [BOOTRST=0]

Boot Flash section size=2048 words Boot start address=\$3800; [BOOTSZ=00] ; default value

☐ Preserve EEPROM memory through the Chip Erase cycle; [EESAVE=0]

☐ Watchdog timer always on; [WDTON=0]

☒ Serial program downloading (SPI) enabled; [SPIEN=0]

☐ JTAG Interface Enabled; [JTAGEN=0]

☐ On-Chip Debug Enabled; [OCDEN=0]

Brown-out detection level at VCC=2.7 V; [BODLEVEL=101]

This gives:

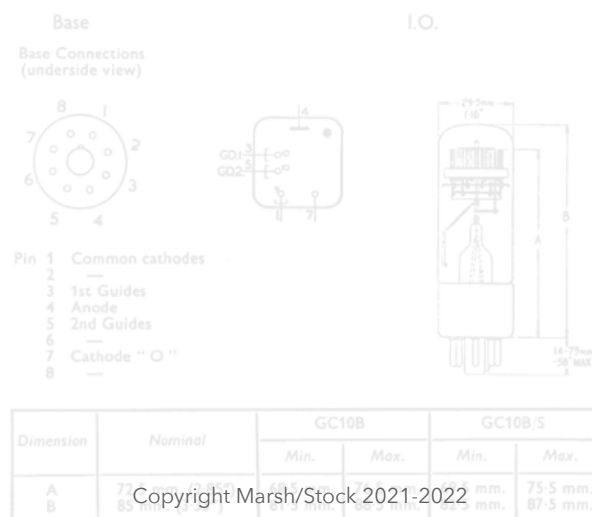
Low Fuse - 0xF7
High Fuse - 0xD8
Extended Fuse - 0xFD

These are almost the same as in 1.1 above, just one bit has changed to activate the boot reset vector.

5.5.2 Burn the firmware

Then using your favourite AVR ISP software burn the firmware. The file from the distribution dropbox will have a name like "bootloader.1.2.1.hex", the numbers identifying the version, currently 1.2.1. The board must be powered when programming.

This completes the programming.



6. SD Card Access, Folders and Files

To inspect or change the SD card contents it can either be removed and plugged into a reader or a USB cable can be plugged into the SAM3X8C board and your PC, then the SD card will appear as a "USB Drive"⁶⁷ with a drive letter assigned.

The operating system uses the classic DOS 8.3 file name format and not long filename formats. The filenames are however case insensitive so lowercase filenames and extensions are permitted. Like any file system you should occasionally back up the contents of the SD card. I just keep the contents mirrored to a folder on my PC. Whilst the clock does write files to the SD card for its own purposes, all of them can be regenerated by the clock and so there is normally no need to back up the SD card until you have made changes by editing a file, adding new WAV or NUG files etc. or perhaps when a new firmware version has been installed.

Root Folder

The root folder contains the following:

File Name	Description
system.ini	Paths to files, controls for the amplifier volume, PIR, IR handset, encoder, LDR, bling-bling and sleep.
chimes	Clock use only - stores prepared chime files
nuggle	Nuggle source code files
runtime	Clock use only - stores runtime Nuggle
system	See below
wav	Chime WAV files

System.ini is an important file for you to read through and edit. The file is self documented and contains many options for you to select, such as your preferred chime muting times, the clock IR remote pair number, the parameters used for various clock beeps etc. It also contains a number of system allocation factors (e.g. file path names). These won't need changing unless you start to seriously delve into the clock's firmware.

System Folder

This folder contains several files that you will need to edit and data files that you will need to reference.

File	Contents
astro.ini	Contains astronomical calculation controls and factors (minimal contents for the Harwell Clock).
constant.ini	Constant name to value look up table contents. The user can add additional values for use in their own Nuggle programs.

⁶⁷ In the Device Manager it appears as an "ATMEL SD/MMC Card Slot USB Device"

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

File	Contents
dst.ini	Daylight saving time rules for the various countries that use DST. Should not require editing. If a rule needs to be changed then expect a new version of the file from me.
help.txt	The help messages used by the menu console. So when you enter ADC? for example, the help text is read from this file. Should not require editing.
locale.ini	An important file for you to edit. See below.
nuggle.ini	An important file for you to edit. See below.
timezone.csv	World time zone location, names, short name, and offsets. Countries the use DST will have two entries. Examine this file for your locale information. Should not require editing. If an entry needs to be changed then expect a new version of the file from me.
wifi.ini	An important file for you to edit. See below.

Note:

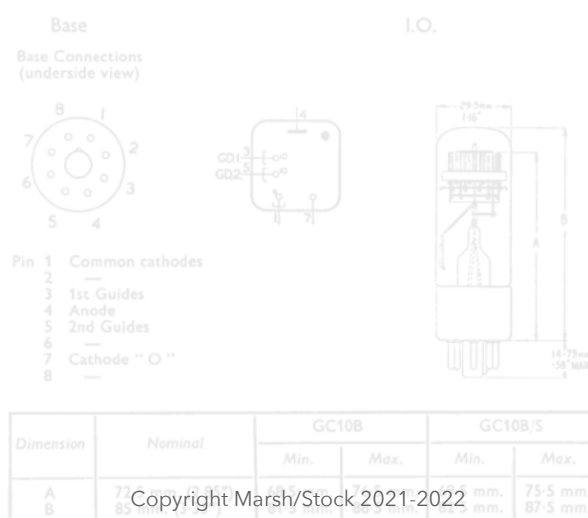
In the system folder there is a hidden file "CACHE.DAT". This file contains indices for locating chime files quickly and is unique to the SD card. It is important that this file is not copied and transferred to another SD card as the chime system will not operate correctly.

locale.ini is used to store your location information. You should read through this file and edit the entries as directed in the documentation within the file.

- The [location] information is used to calculate your local sunrise and sunset times.
- [datefmt], [timefmt] Your date and time format preferences
- [timezone] Unless you are in the UK you will need to change the time zone name and you may need to change the DST rule.

nuggle.ini holds, *inter alia*, the look up table from a Nuggle program filename on the SD card to a numerical index value that the clock uses to select a clock face. This is an important function of this file as it allows you to make a list of the clock faces that you like best and place them close together so you can use the rotary encoder or IR handset to select them. It is also used to control the Nuggle parser, lexer and expression evaluator. You probably won't need to change these settings unless you delve deeply in to Nuggle.

wifi.ini is used if you are using the WIFI plug in module. The only editing you must do to this file is to enter your WIFI log in credentials. You can read through the remaining sections in the file and make other changes as you see fit. They allow you to change the DHCP, SNTP and DNS settings.⁶⁸



⁶⁸ If you don't know what these are then just leave alone and use the default settings.



7. Adding or changing Chime WAV files

Introduction

The Harwell Clock uses WAV files to sound chimes and make other noises. The clock hardware uses the two 12 bit Digital to Analogue Converter (DAC) outputs from the SAM3X8C each passed through Low Pass Filter (LPF) to an audio amplifier feeding a small speaker.

Input WAV file requirements

The input file requirements are:

- WAV file format⁶⁹
- Sample rate not higher than 44100Hz
- Uncompressed
- Up to 32 bit format (not 64 bit)
- DOS 8.3 file naming i.e. no long filenames

The input file can be:

- 8 bit unsigned, or
- 16 bit signed
- mono or stereo

Converting other formats

As long as another format file, such as a MP3, can be converted to a WAV format file then it should be possible to use it. For file conversion I recommend the free Audacity⁷⁰ software. Using this software you can export the file as a WAV format readable by the Harwell. It also enable you to play and edit the file. You can blend several files together and add effects like echo and reverb. Have fun!

Copying the WAV file to the SD card

All WAV files must be copied to the WAV folder in the root directory of the SD Card. You can access the SD card using the methods described in section 6 of this chapter.

Pre-processing the WAV files for use

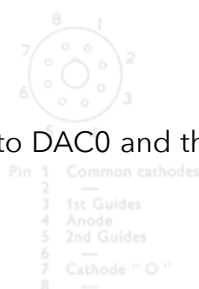
The Harwell Clock does not use the WAV file directly while playing a chime or sound effect. Instead a preprocessor first converts the WAV file into a format that is directly compatible with the DAC requirements:

- Unsigned 8 bit data is converted to signed 12 bit data
- Signed 16 bit data is converted to signed 12 bit data
- Mono data is saved as two identical stereo channels
- Channel tags are added that direct the left data stream to DAC0 and the right data stream to DAC1

⁶⁹ <https://en.wikipedia.org/wiki/WAV>

⁷⁰ <https://www.audacityteam.org/>

Base Connections
(underside view)



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

Copyright Marsh/Stock 2021-2022



The converted data is saved to a directory called CHIMES in the root directory of the SD card. For a given WAV file, two files with the same name are created with extensions CHM containing the DAC ready data and HDR containing set up data such as the sample rate and information about the original data, such if it was mono or stereo.

The console menu command **CHMC** is used to run the WAV file pre-processor. To connect the console see section 3 above and to use the menu commands, section 4.

Before giving the CHMC command ensure that the RTC contains the correct data and time as the file timestamps are used to determine which WAV files need to be updated.

When the command is given all clock functions are stopped and, if connected, the USB drive is disconnected by the firmware. The clock firmware then:

- Creates the CHIMES directory if it does not exists
- Scans the WAV directory for WAV files
 - If the corresponding CHM and HDR files don't exist in the CHIMES directory then it creates them and proceeds with the conversion
 - If the CHM and HDR files exist then it replaces them if the WAV file is newer
- When all the conversions have been completed the Harwell Clock resets itself and the clock restarts.

N.B. The file pre-processor is quite a slow operation. Conversion of a file can take several seconds to many minutes. So if you converting several files then go and make a cup of coffee and have a piece of cake...

Other useful menu commands

CHMK	Chime Kill	Stop the chime currently playing
CHMR	Chime Report	List all the preprocessed files available with some key data
CHMU <i>name</i>	Chime Use	Play the given named chime

Un-fragmented File Requirement

For speed, the chime CHM files are read directly by the low-level disc functions and not using the higher level FAT32 functions. To play correctly, this means that a CHM file must not be fragmented. If a chime "breaks up" when playing then it is an indication the file is fragmented. The best process to defragment an SD card is to copy the files contents to a directory on your PC. Delete all of the SD card contents. Then copy the saved copy back again. I do not recommend using a defrag tool on the SD card. You can use the **CHMT** to determine if any files are fragmented.

You'll need to keep a copy of the SD card contents as a back up in any case.

Using new chime files

The chimes are played by statements in the Nuggle file that displays a given clock face. If you have replaced a file with the same name then the Nuggle will use the new file. Otherwise you will need to edit the Nuggle text - this is not at all difficult - see the next section.



8. Adding or changing Nuggle programs

Nuggle is a simple programming language used by Sgitheach clocks. It describes how the display devices such as VFD, CRT, Dekatron or E1T tubes, bling LEDs, chime system will be used. The Harwell clock comes with a large number of Nuggle files available so you can review these to see actual Nuggle source text examples.

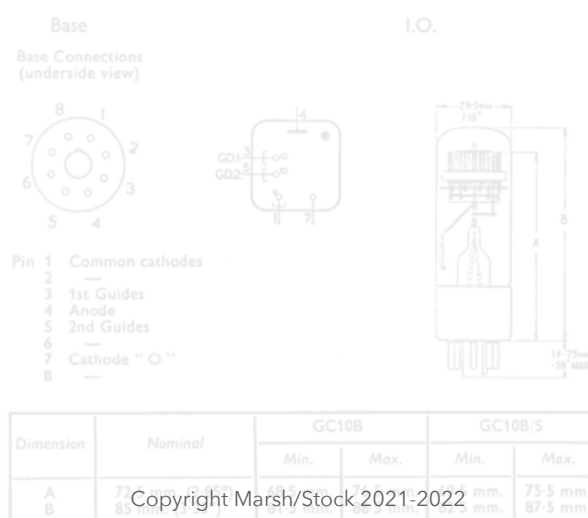
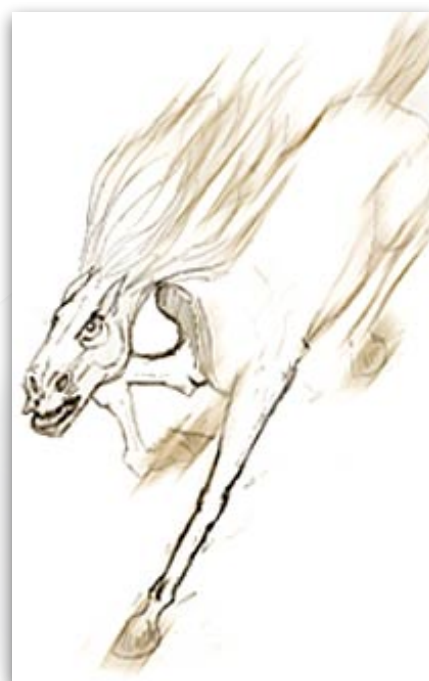
All Nuggle programs reside in the **nuggle** folder and always have the extension **.nug**⁷¹. They are simple text files.

If you edit and save an existing Nuggle file then, when that face is shown again, the clock firmware will see that the date stamp on the file is newer than before so it will process the file and display the modified face.

If you create a file with a new filename (for example by editing an existing file and then saving the modified file into a new filename) then you now need to assign an index number to the file as Nuggle files are selected to run by a number and not the filename directly. To do this, you edit **nuggle.ini** and in the [faces] section you create a new entry with an unused index and your filename (the instructions are with **nuggle.ini**). Next, you force the clock to reload the indices by using the **NUGL** menu command (or power cycling the clock or pressing the reset button). The newly indexed file should be selectable using the IR handset or the rotary encoder.

Whilst this looks like a complex process at first sight, remember that you have created or modified a clock face without having to resort to modifying the clock source code, compiling the C and burning the SAM3X8C.

Have fun with Nuggle! To help you, there is a generalised, clock independent Nuggle manual, a manual specific for the Harwell Clock and a Nuggle cookbook.⁷²



⁷¹ Could it be anything else?

⁷² In preparation....

9. Compiling the Firmware for the Harwell Clock

9.1 Compiling the SAM3X8C Firmware

Introduction

These notes are to help you install the final release 1 firmware source code and how to compile it. The output of this process are binary files that are identical to the binary files available on the dropbox. These notes are not a tutorial on using Atmel Studio and assume you have a good working knowledge of Atmel Studio 7.0 and where everything is.

At this time I am using:

- Atmel Studio 7.0.1931
- Atmel Framework version 3.35.1
- Windows 10 - NT6.2.92000.0
- ARM/CNU C Compiler 6.3.1

Distribution file download and installation

The firmware source code files for the SAM3X8C are distributed as Atmel Studio Template files. You copy the *Sgitheach 1.2.1.zip*⁷³ from the Dropbox to the Studio Template directory. On my PC this is:

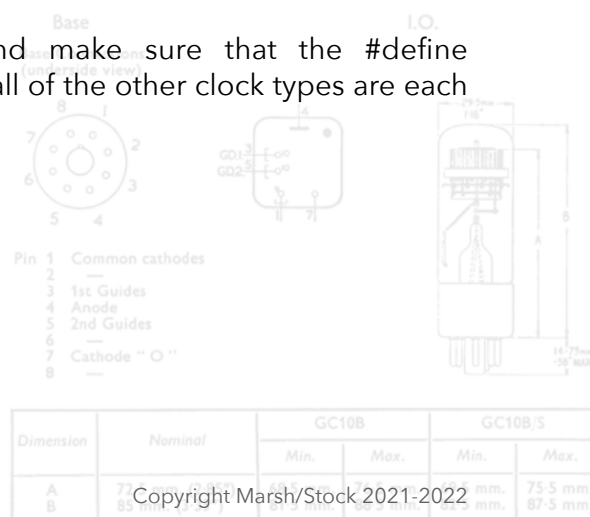
`c:\users\grahame\documents\visual studio 2015\my exported templates`

so it will definitely be in another location on your system!

Creating a project and compiling the SAM3X8C firmware

- Run Atmel Studio and start a new project.
- The new project dialogue box should now show an installed C/C++ template called "Sgitheach Clocks 3X8X.1.2.1". Select this file and set the name to "Harwell01" or whatever you want and the location to where you want the project files to be placed.
- Remove/delete the *Device_Setup* folder and all of its contents.
- Remove/delete the *main.c* file at the bottom of the explorer window (leaving the *main.c* that is inside the *src* folder).
- Open the *conf_version.h* file in the *config* folder and make sure that the `#define HARWELL_DEKATRON_CLOCK` is not commented out but all of the other clock types are each commented out.
- Change the Solution Configuration to **Release**.
- Open the project properties dialogue and change:

⁷³ Or the latest and greatest on the [Dropbox](#)...



- Compiler General - untick the default char type and default bit field type checkboxes if ticked.
- Compiler Optimisation - change the level to **Optimise size (-Os)**.
- Compiler Optimisation - add, if missing, **-fdata-sections -fstack-protector-all**.
- Compiler Symbols - if present, delete the **printf=iprintf** and **scanf=iscanf** symbols.
- Compiler Symbols - add, if missing, **BOARD=USER_BOARD** and delete any other **BOARD=** symbols.
- Compiler Symbols - add, if missing, **STACK_CHK_GUARD=0xa242b7f0**.
- Linker Miscellaneous - add, if missing, Other objects - **-Wl,-defsym=HEAP_SIZE=0x10000**.
- Build the solution, the output should show no errors, warnings or messages.
- The output tab should show (exactly):

```

Program Memory Usage      :      335056 bytes      63.9 % Full
Data Memory Usage         :      24112 bytes       24.5 % Full

```

You have now compiled a version identical to the *Sgitheach 1.2.1.bin* file on the documentation links provided at <http://www.sgitheach.org.uk/harwell.html#documentation>. The source code is now yours to modify⁷⁴!

9.2 Compiling the MEGA324PA Firmware

You only need to do this if you wish to modify or explore the source code. This section assumes you have installed Atmel Studio 7 and know how to use it.

The firmware source code file for the MEGA324PA bootloader is distributed as an Atmel Studio Template file. You should copy the file Dekatron.1.2.1.zip from the Dropbox to a suitable directory on your system. Then

- Run Atmel Studio.
- Select File - Import - Project Template...
- Select the file Dekatron.1.2.1.zip from the directory where you saved it and press OK.
- Atmel Studio should say that a project template has been successfully imported.
- Select File - New - Project...
- The new project dialogue box should now show an installed C/C++ template called Dekatron.1.2.1. Select this file and set the name to mydeka.1.2.1 or whatever you want and the location to where you want the project files to be placed.
- Open the project properties dialogue and in the Toolchain tab:
 - In the AVR/GNU C Compiler General tab:
 - Untick the default char type and default bit field type checkboxes.
- Change the Solution Configuration to Release.

⁷⁴ If you dare! :-)

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

- In the Solution Explorer window delete the main.c file at the end of the tree (not the main.c file under the src indent).
- Build the solution, the output should show no errors, warnings or other messages!

The output tab should show (exactly):

```
Program Memory Usage : 9338 bytes    28.5 % Full
Data Memory Usage :   1479 bytes    72.2 % Full
```

You have now compiled a version identical to the Dekatron.1.2.1.hex file on the [Dropbox](#)⁷⁵. The source code is now yours to modify!

9.3 Compiling the MEGA324PA Bootloader

You only need to do this if you wish to modify or explore the source code. This section assumes you have installed Atmel Studio 7 and know how to use it.

The firmware source code file for the MEGA324PA bootloader is distributed as an Atmel Studio Template file. You should copy the file bootloader.1.2.1.zip from the [Dropbox](#) to a suitable directory on your system. Then

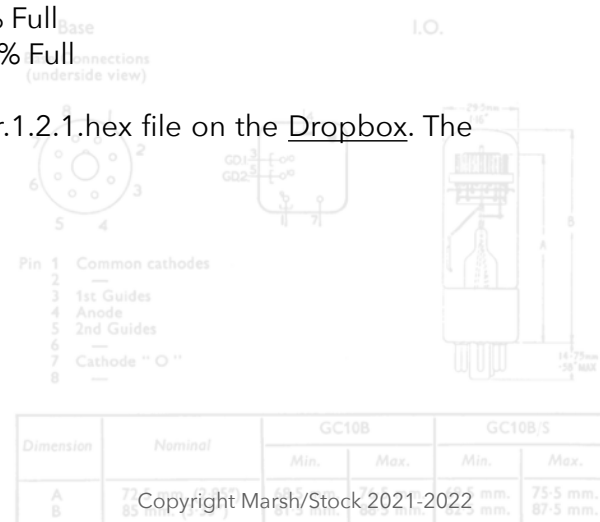
- Run Atmel Studio.
- Select File - Import - Project Template...
- Select the file bootloader.1.2.1.zip from the directory where you saved it and press OK.
- Atmel Studio should say that a project template has been successfully imported.
- Select File - New - Project...
- The new project dialogue box should now show an installed C/C++ template called bootloader.1.2.1. Select this file and set the name to myboot.1.2.1 or whatever you want and the location to where you want the project files to be placed.
- Open the project properties dialogue and in the Toolchain tab:
 - In the AVR/GNU C Compiler General tab:
 - Untick the default char type and default bit field type checkboxes.
 - And in the Linker Memory Settings tab add this entry to the FLASH segment if not present:
 - .text=0x3800
- Change the Solution Configuration to Release.
- In the Solution Explorer window delete the main.c file at the end of the tree (not the main.c file under the src indent).
- Build the solution, the output should show no errors, warnings or other messages!

The output tab should show (exactly):

```
Program Memory Usage : 2580 bytes    7.9 % Full
Data Memory Usage :   439 bytes     21.4 % Full
```

You have now compiled a version identical to the bootloader.1.2.1.hex file on the [Dropbox](#). The source code is now yours to modify!

⁷⁵ Or whatever the latest version is.....



10. InfraRed Remote Control

The IR handset is used to remotely and conveniently change the way the Harwell Dekatron clock is operating. The keystrokes on the IR handset can be overridden by certain Nuggle face scripts that are running. The description here is the default use of the IR keys. You should check with the Harwell Clock Distribution Faces section to see if a particular face overrides the key use described here (not many actually do).

By default, the IR handset will automatically start to repeat a transmission if the key is pressed and held down.

The default key functions described here assume a starting condition of the clock displaying a clock face. By default the clock beeps on each keystroke to give some audio feedback that the clock has received the key press.

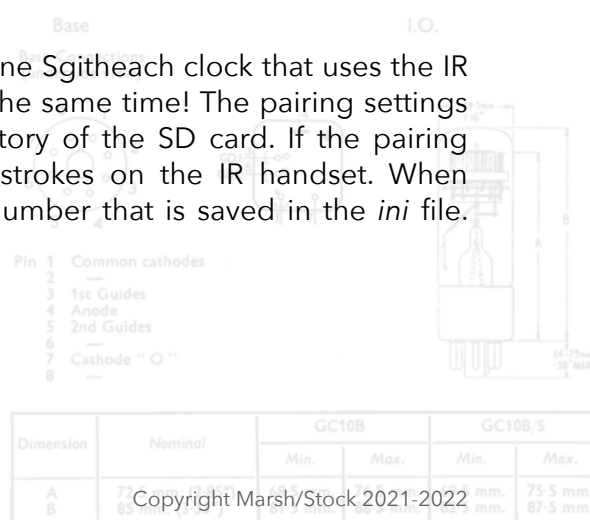


Key	Default Action
CH-	Go to the clock face before the current face in the clock face list.
CH	Show the current clock face number for a few seconds before returning to the face
CH+	Go to the clock face after the current face in the clock face list.
 <<	Decrease the beep volume. The clock will beep and the tubes will display the current volume level (0 - 31).
>> 	Increase the beep volume. The tubes will display the current volume level (0 - 31).
> 	Start the clock "pairing" keystroke sequence - see below.
-	Decrease the chime volume level. The tubes will display the current volume level (0 - 31).
+	Increase the chime volume level. The tubes will display the current volume level (0 - 31).
EQ	Unused - however the 'EQ' key is used by many key stroke sequences described below.
0 ... 9	Start the entry of a new clock face number keystroke sequence - see below.
100+	Unused - however the '100+' key is used by many key stroke sequences described below
200+	Unused - however the '200+' key is used by many key stroke sequences described below

Pairing Keystroke Sequence

The "Pairing" function is only needed if you have more than one Sgitheach clock that uses the IR handset so that both clocks don't respond to key presses at the same time! The pairing settings are located in the [ir] section of *system.ini* in the root directory of the SD card. If the pairing function is disabled then the clock will respond to all keystrokes on the IR handset. When enabled, the clock must be "paired" by entering the clock number that is saved in the *ini* file. Recommended numbers are (so far):

1. Scope Clock Due
2. Fortress E1T Clock
3. The Harwell Dekatron Clock



However, you can use any number you wish in the range 1 to 999999. Obviously if you have two Sgitheach clocks of the same type then you'll need to give them different numbers.

To pair the clock, you start by pressing the '>||' key. All Sgitheach clocks respond to this key press and enter the pair number entry mode. By default the left colon will light blue to show the clock has entered pairing mode and the rest of the bling will be black.

Then enter the pairing number by using:

- The numbers '0' to '9' to enter a number
- The '100+' key acts as a "backspace" key and deletes the last number entered
- The '200+' key acts as an "escape" key and the entry of the pairing number is cancelled
- Press the 'EQ' key when you have entered the correct number

By default, the pairing number you enter is saved to NVR so it will be restored the next time the clock is restarted.

When the 'EQ' key is pressed, only the Sgitheach clock with the corresponding pairing number will respond to other key strokes.

Hint:

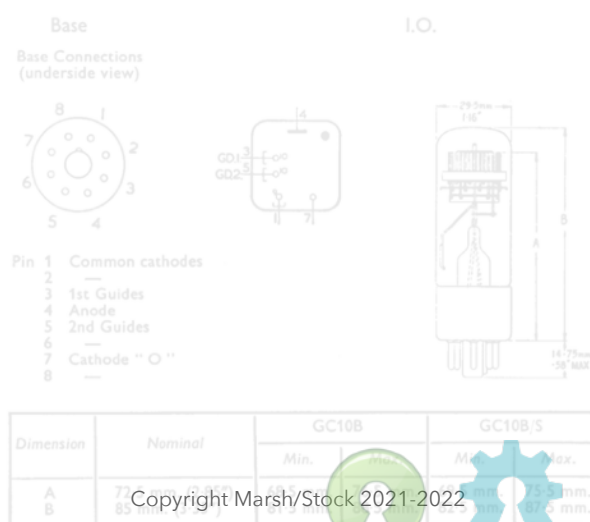
Clock face 950 and console menu command **IRHR** will show the pairing number and status.

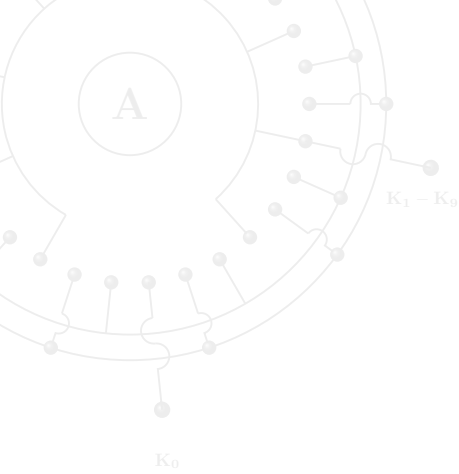
New Face Keystroke Sequence

A face number can be dialled in using the number keys. When the first number key is present then it will be displayed on the tubes. By default the left colon will light green to show the clock has entered face number entry mode and the rest of the bling will be black.

- Use the numbers '0' to '9' to enter the rest of the number
- The '100+' key acts as a "backspace" key and deletes the last number entered
- The '200+' key acts as an "escape" key and the entry of the pairing number is cancelled
- Press the 'EQ' key when you have entered the correct number

If you enter a number with no corresponding face then face 1, blank by default, will be displayed.

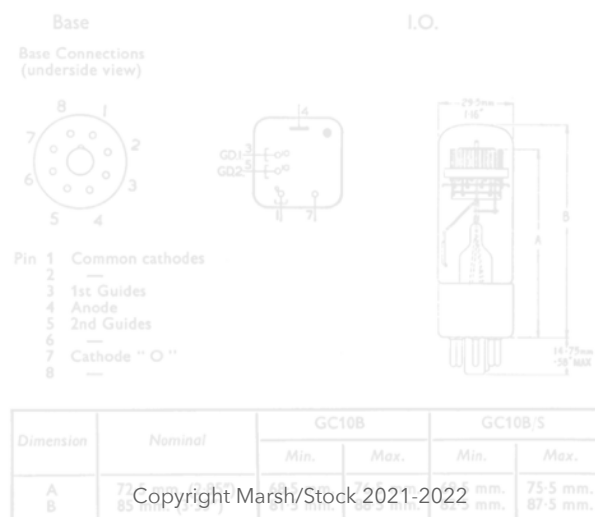




Harwell Clock - How It Works

What Makes The Clock Tick⁷⁶

⁷⁶ You'd think that joke would get old....but...



Harwell's Inner Workings...

Introduction

The Harwell Clock schematic diagrams are spread across over a dozen pages and many of the individual pages have several subsystems on them. This Harwell manual section is a walk through of the schematic drawings.

Subsystems

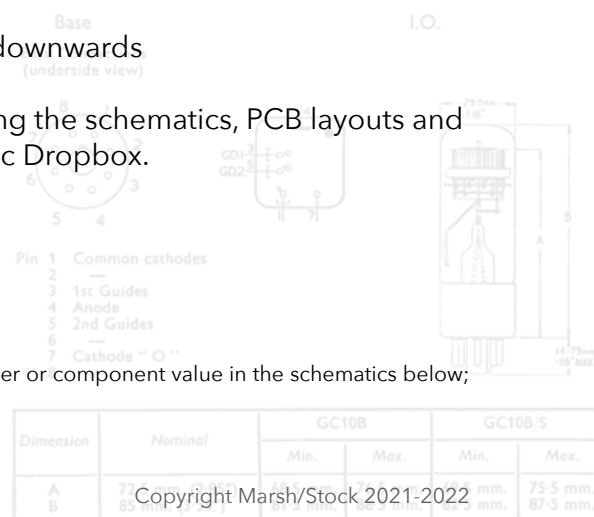
The clock consist of the following subsystems:

- SAM3X8C micro-controller - this is mounted on its own PCB that plugs into the clock's main PCB
- +5V power supervisor
- SD Card
- NVRAM (non-volatile random access memory)
- Super Cap backup power supply for the RTC
- User selected option jumpers
- Hardware version detection
- Stereo audio amplifiers
- A front panel sub-board carrying:
 - PIR (passive infrared) motion detector
 - IR (infrared) receiver
 - Rotary encoder
 - Speakers
 - LDR (Light Dependent Resistor) to measure ambient room light level
 - Two neon filled MTX90 tubes for colons
 - Tricolour (red, green, blue) LED (light emitting diode) modules to bling⁷⁷ the clock
- +500V, +170V, +100V and +50V 10mA power supply to run the Dekatrons and MTX90 tubes
- +5V and +3.3V power supplies to run all the analogue and digital electronics
- An optional WiFi module
- An optional GPS receiver
- Seven tube display drivers mounted on a sub-board
- The seven display drivers are controlled by a MEGA324PA TWI slave
- Each tube driver consists of:
 - A circuit to cause the first guide electrode to glow
 - A circuit to cause the second guide electrode to glow
 - Detection that cathode '0' is glowing
- Further Tricolour LED modules that are arranged to fire downwards

Now let's look at these in more detail. The Eagle files⁷⁸ including the schematics, PCB layouts and Gerbers are all open design and available on the project public Dropbox.

⁷⁷ Or pimp if you prefer.....hey, we won't judge...

⁷⁸ Only the part numbers in the Eagle files are up to date. Do not rely on a part number or component value in the schematics below; they are to illustrate function and operation only.



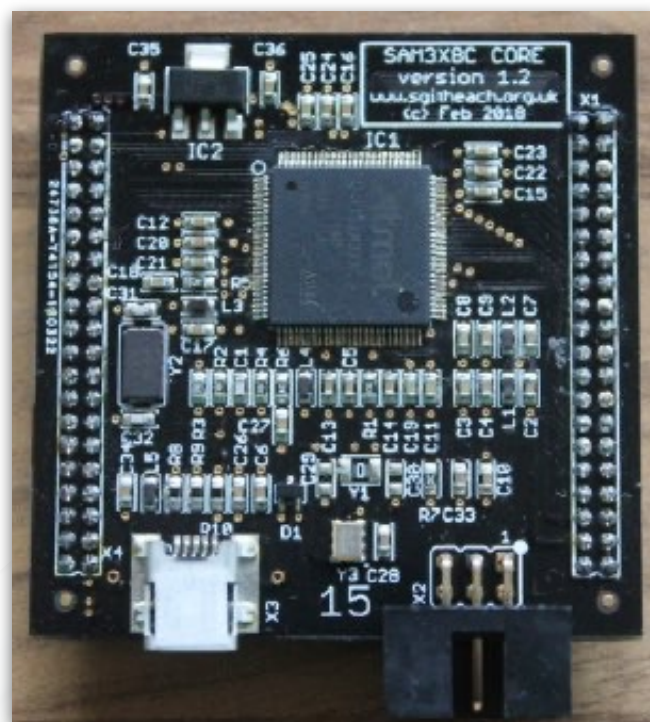
1.0 SAM3X8C Microcontroller Plug-in

Arduino fans may recognise this number in relation to the Arduino Due⁷⁹ as it uses the SAM3X8E micro-controller. The SAM3X8C⁸⁰ is a smaller sibling in a 100 pin package and has less I/O pins but internally it is more or less the same, has the same flash memory (512K) and runs at 84MHz clock speed. The Arduino Due implementation is somewhat restricted as only a selection of the available I/O pins and facilities are available externally. The Arduino Due is limited in shape to conform with other Arduino types (not that this is a bad thing).

The SAM3X8C plug is a four layer PCB measuring 52mm x 52mm:

It carries:

- Native USB port
- Separate 3.3V regulator
- 6 pin programming port⁸¹
- Filtering and decoupling according to the Atmel data sheet
- 12MHz crystal for the system oscillator
- Either a 32768Hz crystal or a 32768Hz TCXO (temperature compensated crystal oscillator) for the RTC
- All I/O pins, power, reset etc. lines are brought out to two 2 x 20 way headers
- External power requirements are 5V and 3V battery for RTC backup
- JTAG support is not implemented
- PA8 and PA9 I/O pins are available on the headers but are used by the programmer



Base
Base Connections
(underside view)



I.O.



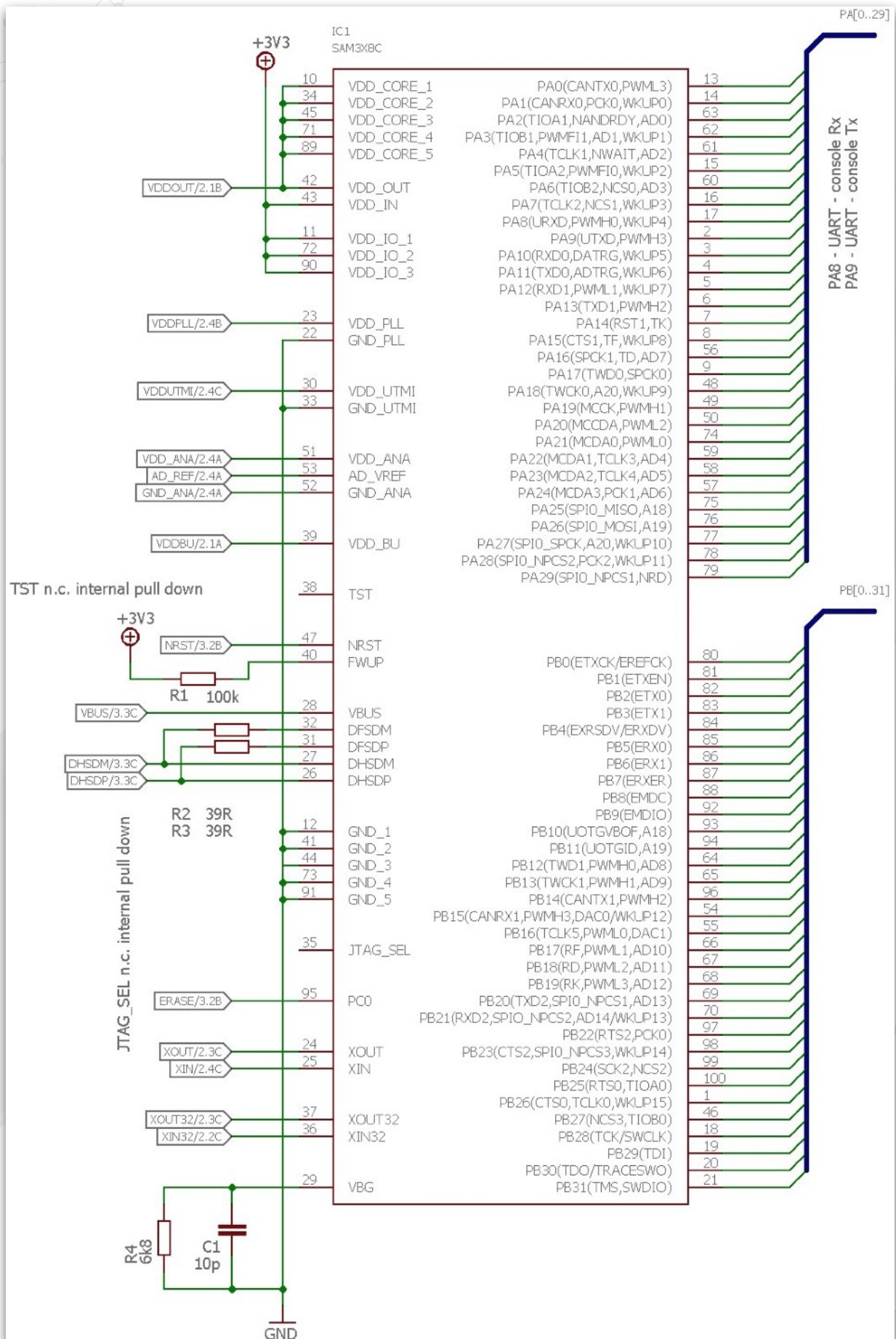
⁷⁹ <https://store.arduino.cc/arduino-due>

⁸⁰ <https://www.microchip.com/wwwproducts/en/ATSAM3X8C>

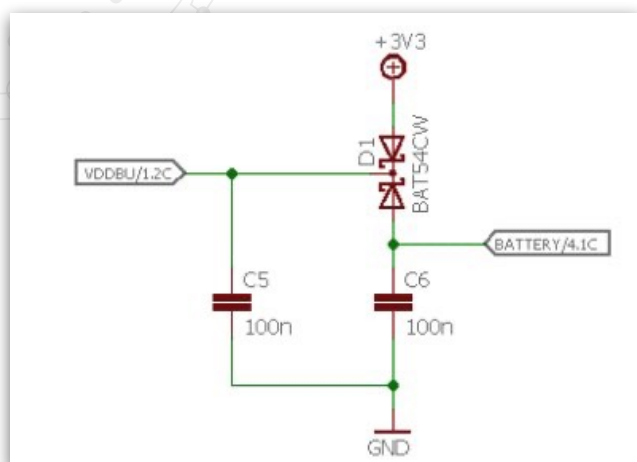
⁸¹ For use with the Sgitheach SAM programmer ONLY and not any old Atmel product

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72.5 mm			75.5 mm	
B	85 mm			87.5 mm	

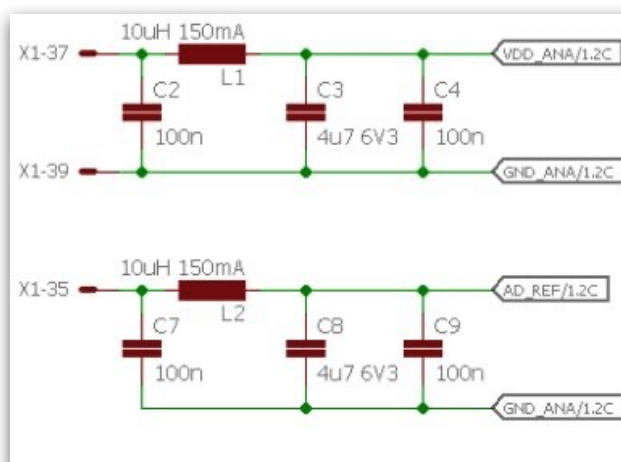
1.1 SAM3X8C schematic



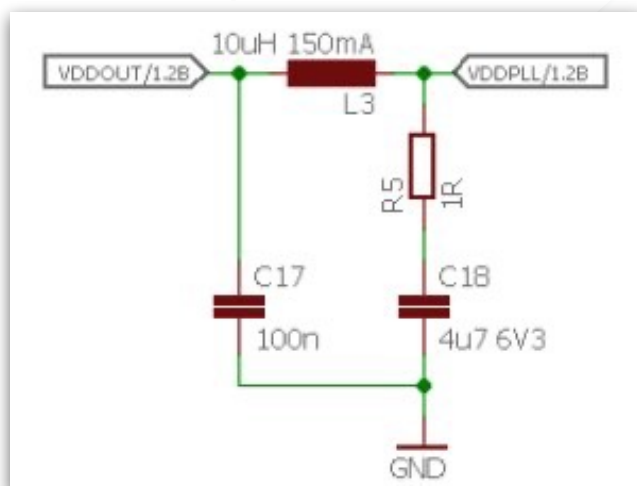
1.2 Battery Backup



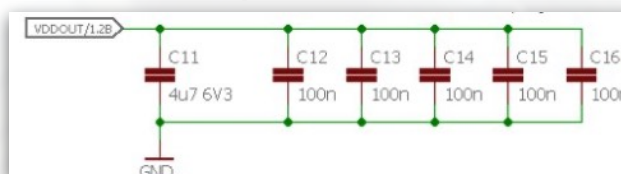
1.3 Analogue Decoupling



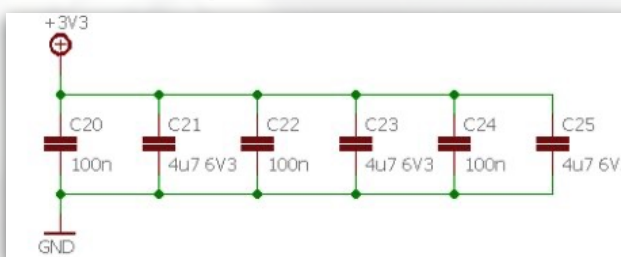
1.4 PLL Decoupling



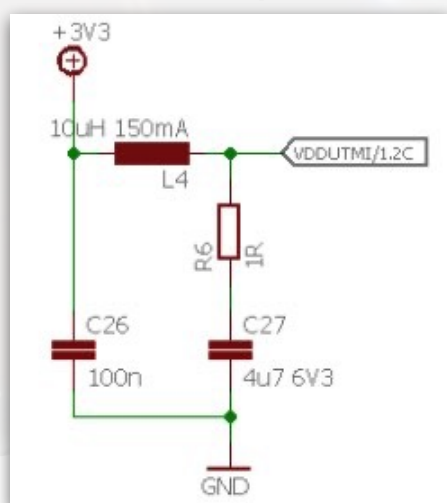
1.5 Core Decoupling



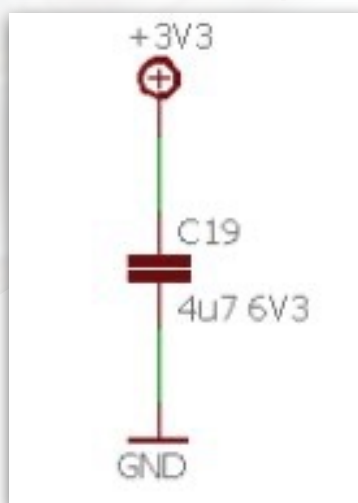
1.6 IO Decoupling



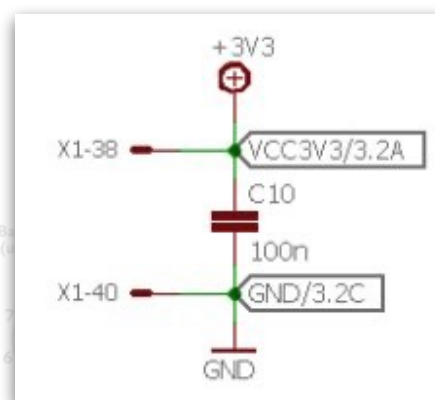
1.7 UTMI Decoupling



1.8 3.3V IN Decoupling



1.9 3.3V OUT

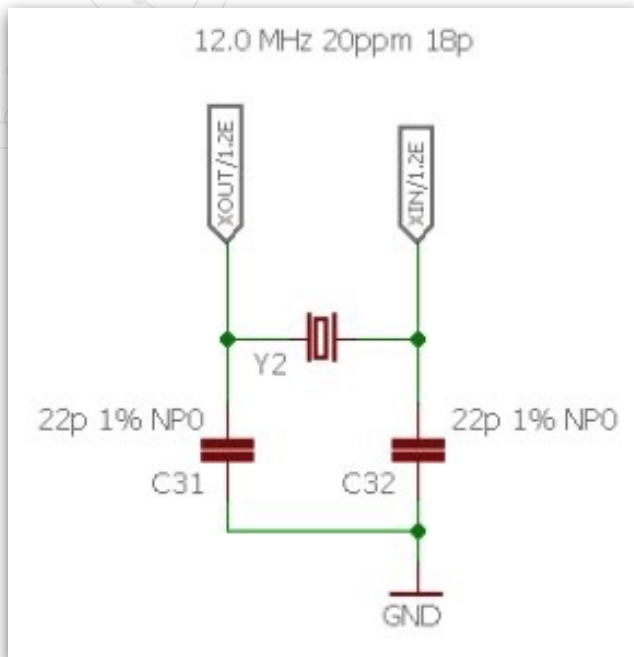


- Pin 1 Common cathodes
2 —
3 1st Guides
4 Anode
5 2nd Guides
6 —
7 Cathode "O"
8 —

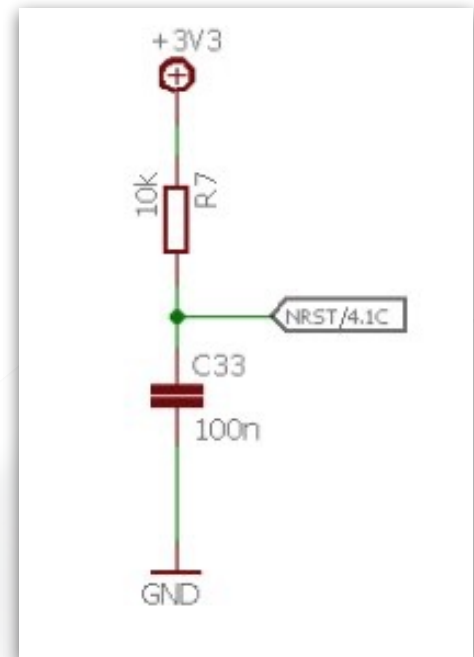


Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

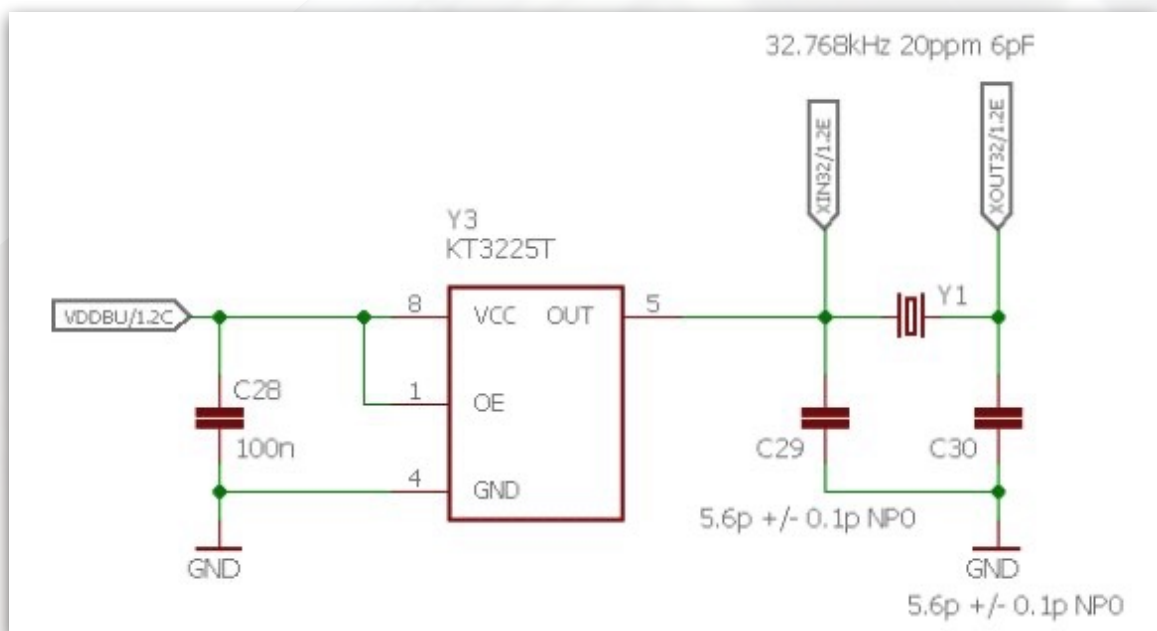
1.10 System Oscillator 12MHz Crystal



1.12 Reset

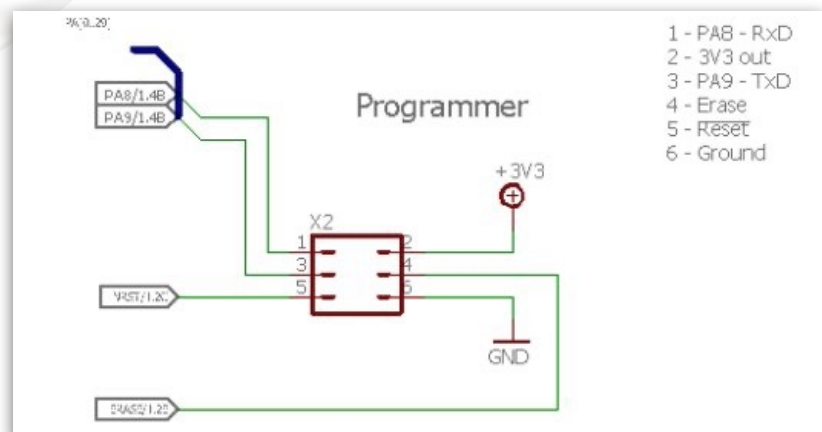


1.11 RTC Oscillator Options



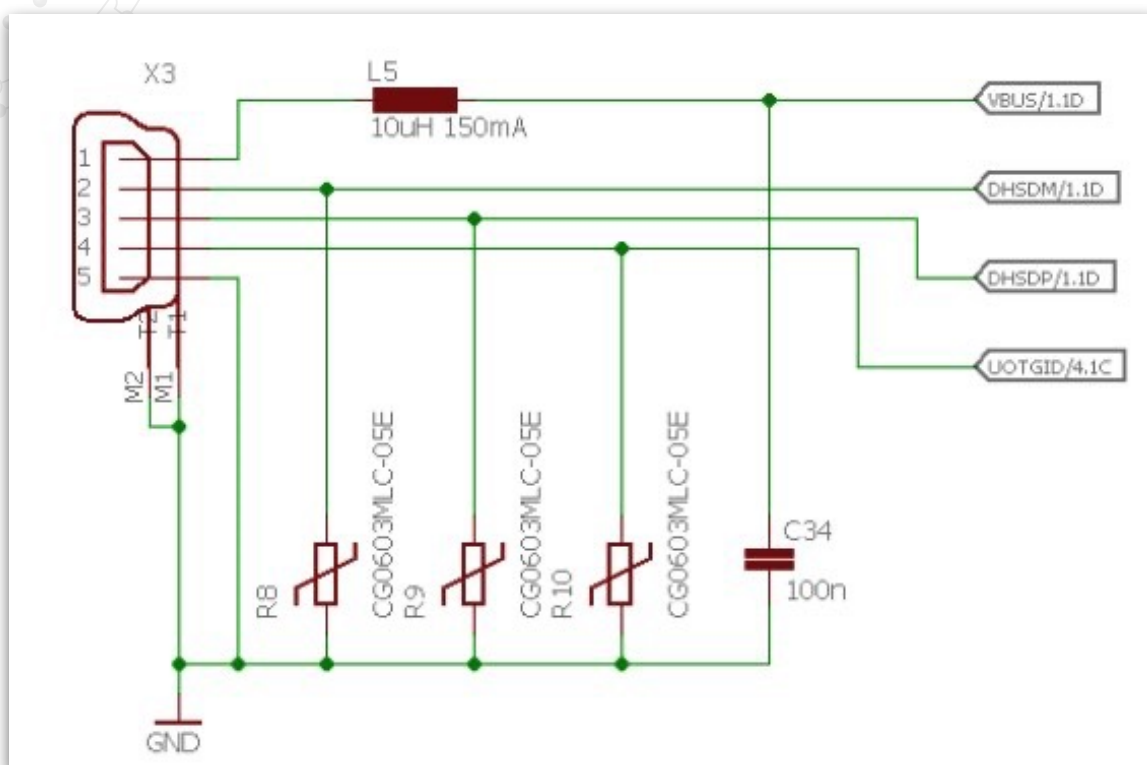
1.13 Programmer Socket

The connections made here are for the Sgitheach SAM3X8C programmer only. No other programmer can be plugged in directly.



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	72.5 mm	72.5 mm	72.5 mm	72.5 mm	72.5 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

1.14 USB Socket

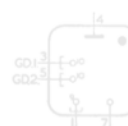


The USB socket cannot be used to power the plugin board. The board can only be powered using connections to the headers.

Base
Base Connections
(underside view)



- Pin 1 Common cathodes
2 —
3 1st Guides
4 Anode
5 2nd Guides
6 —
7 Cathode "O"
8 —

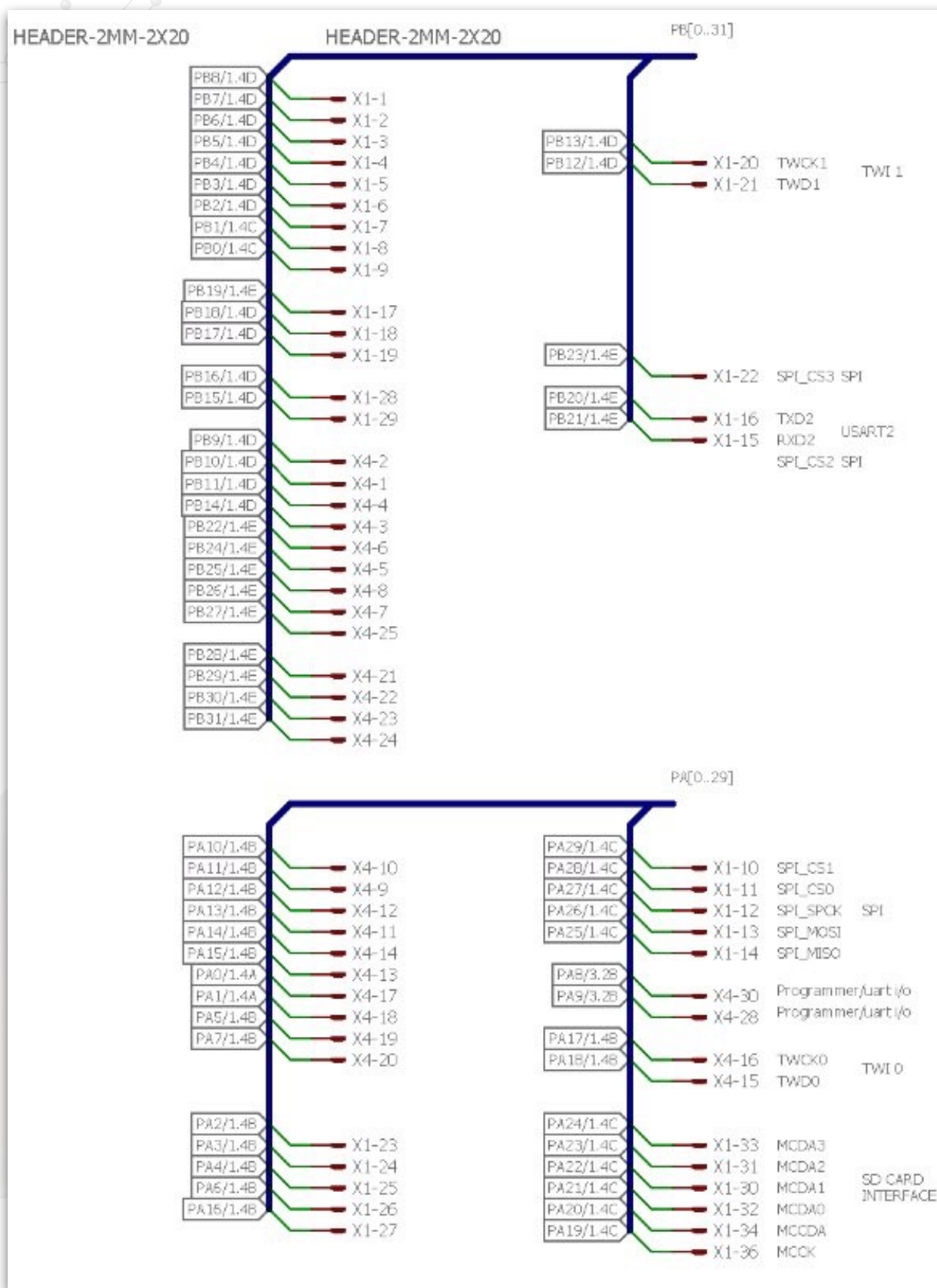


I.O.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72 mm	72 mm	72 mm	75.5 mm	75.5 mm
B	85 mm	85 mm	85 mm	87.5 mm	87.5 mm

1.15 Header Connectors



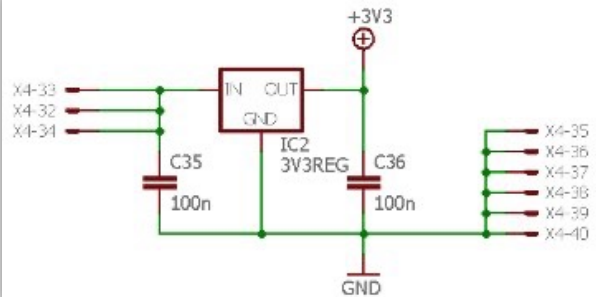
To conserve space 2mm spaced headers are used instead of the more usual 2.54mm spaced headers.

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

1.16 Miscellaneous Connectors



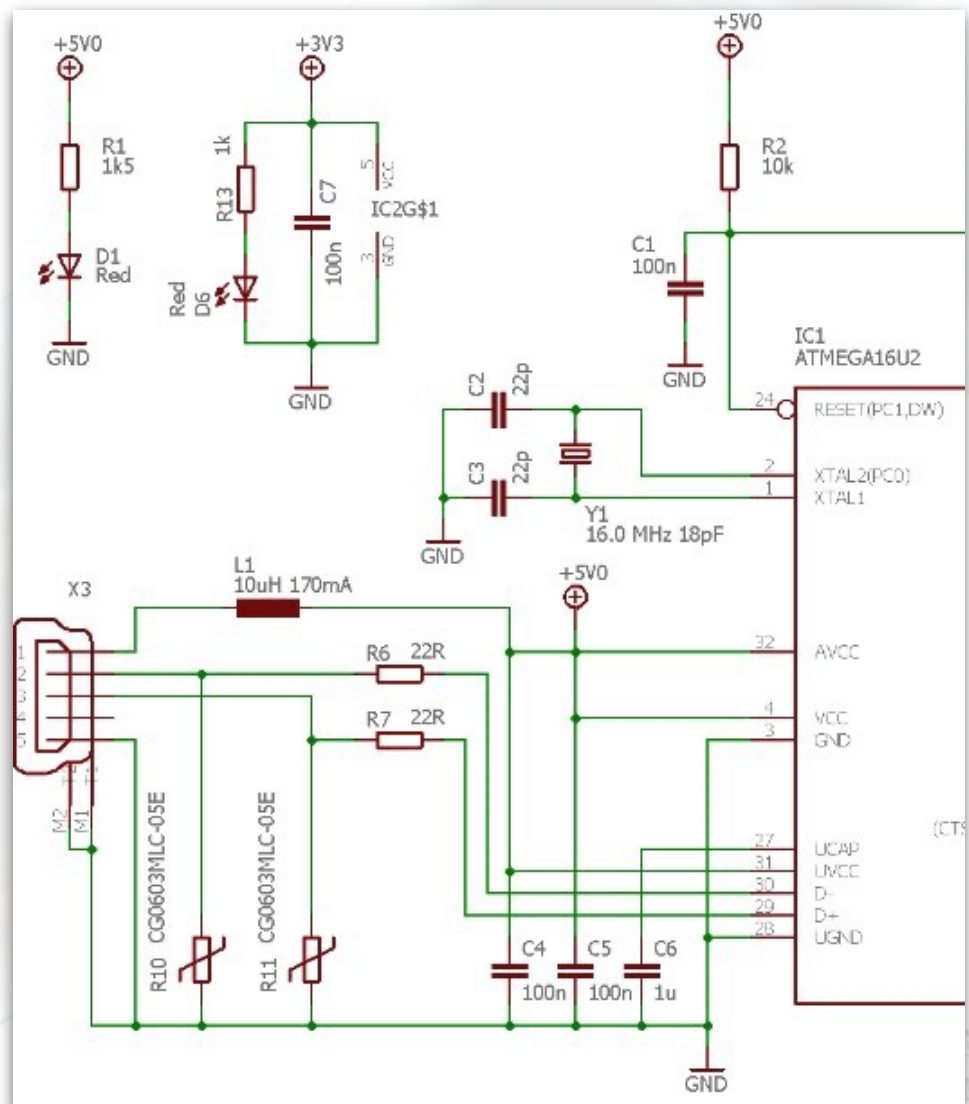
1.17 3.3V PSU



This regulator can be used to supply power but care must be taken not to overload it.

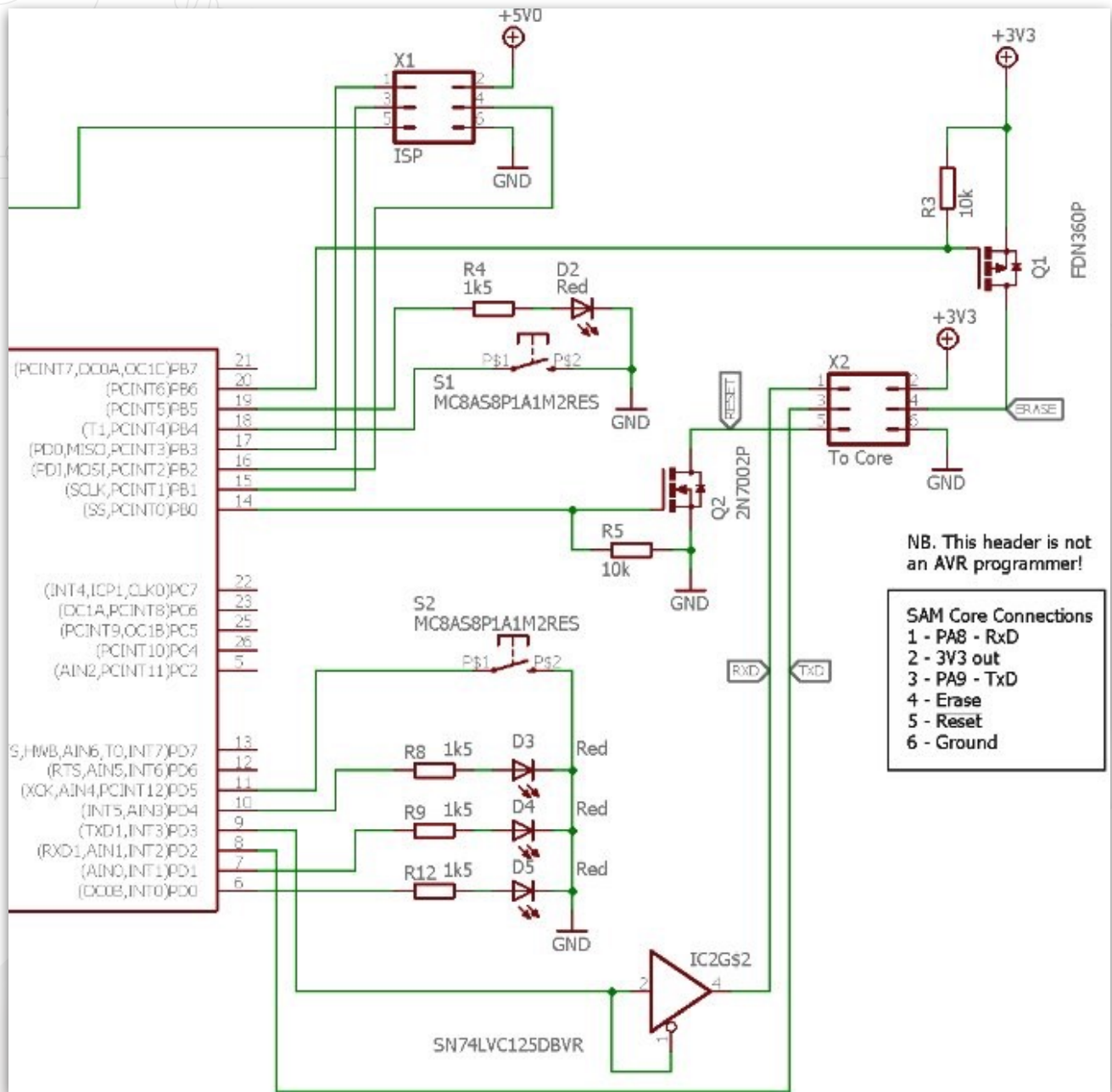
2.0 SAM3X8C Programmer

- 5V power indicator
- 3.3V power indicator
- Reset circuit
- 16MHz clock oscillator
- USB socket and interface
- Decoupling



Pin 1 Common cathodes
2 —
3 1st Guides
4 Anode
5 2nd Guides
6 —
7 Cathode "O"
8 —

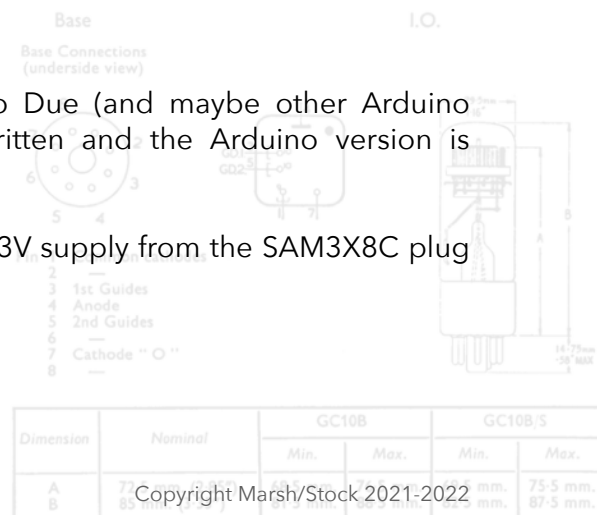
Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm



- MEGA16U2 ISP socket
- Connection to SAM
- Indicator LEDs for Reset, Erase, data Tx and data Rx
- Level shifter for Tx data
- Drivers for SAM Reset and Erase lines
- Optional control switches

This is more or less the programmer used on the Arduino Due (and maybe other Arduino versions as well). The firmware has been completely rewritten and the Arduino version is incompatible with this hardware.

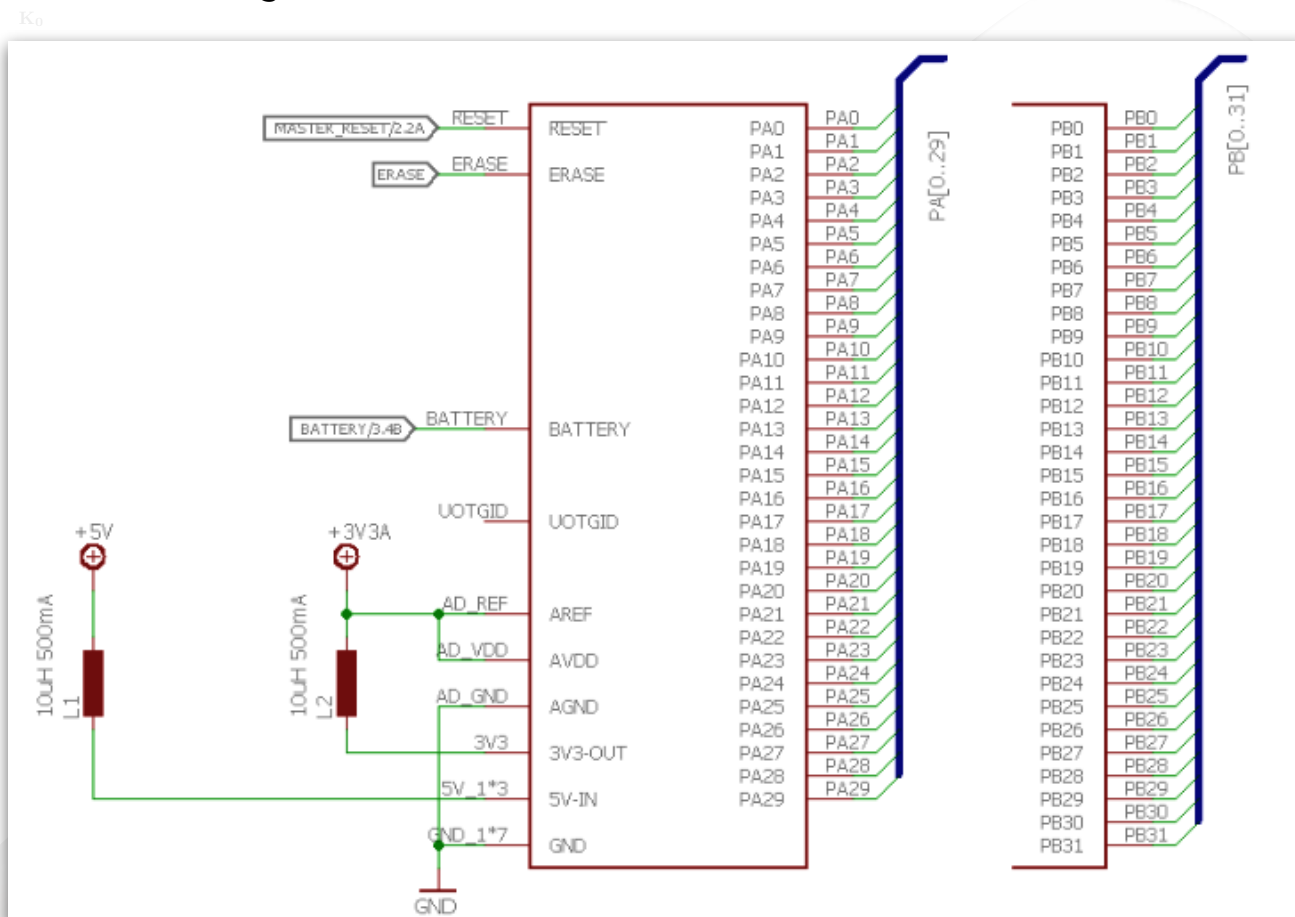
The programmer requires both the USB 5V supply and the 3.3V supply from the SAM3X8C plug in board to function.



3.0 Harwell Main Board

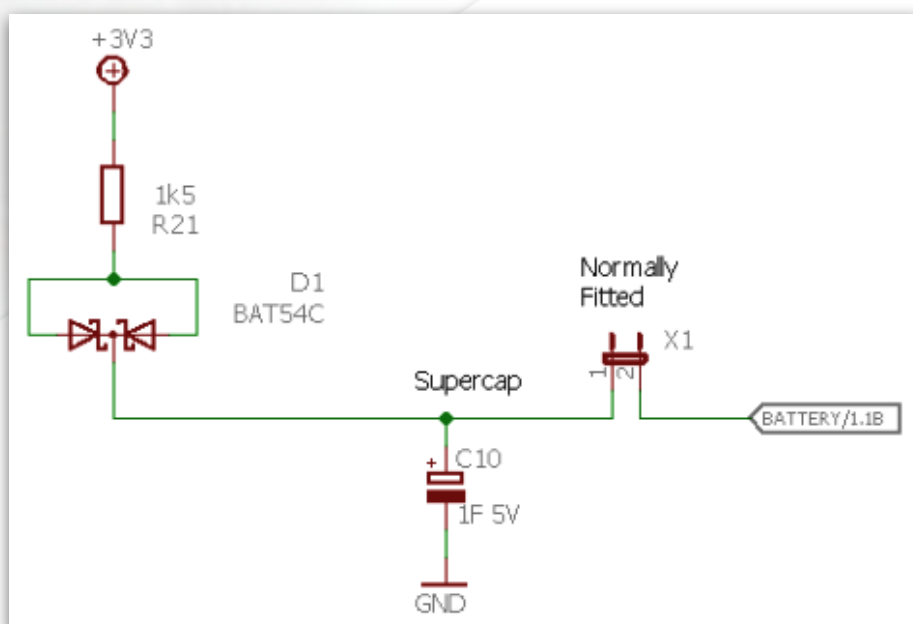
The main board carries the bulk of the digital and analogue electronics. The board is designed to mount horizontally in the case so that its various external connectors etc. are made available through the case rear panel.

3.1 SAM3X8C Plug-in



3.2 RTC Back up power

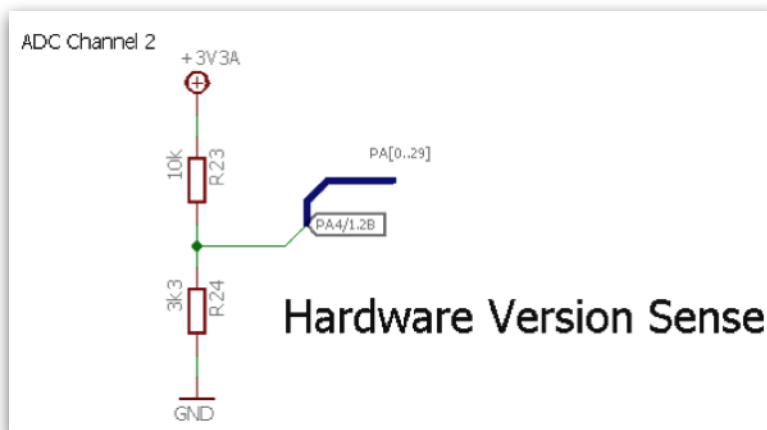
X1 allows the back up power supply PSU to be disconnected from the SAM3X8C so all of the registers in the chip can be reset to factory default values.



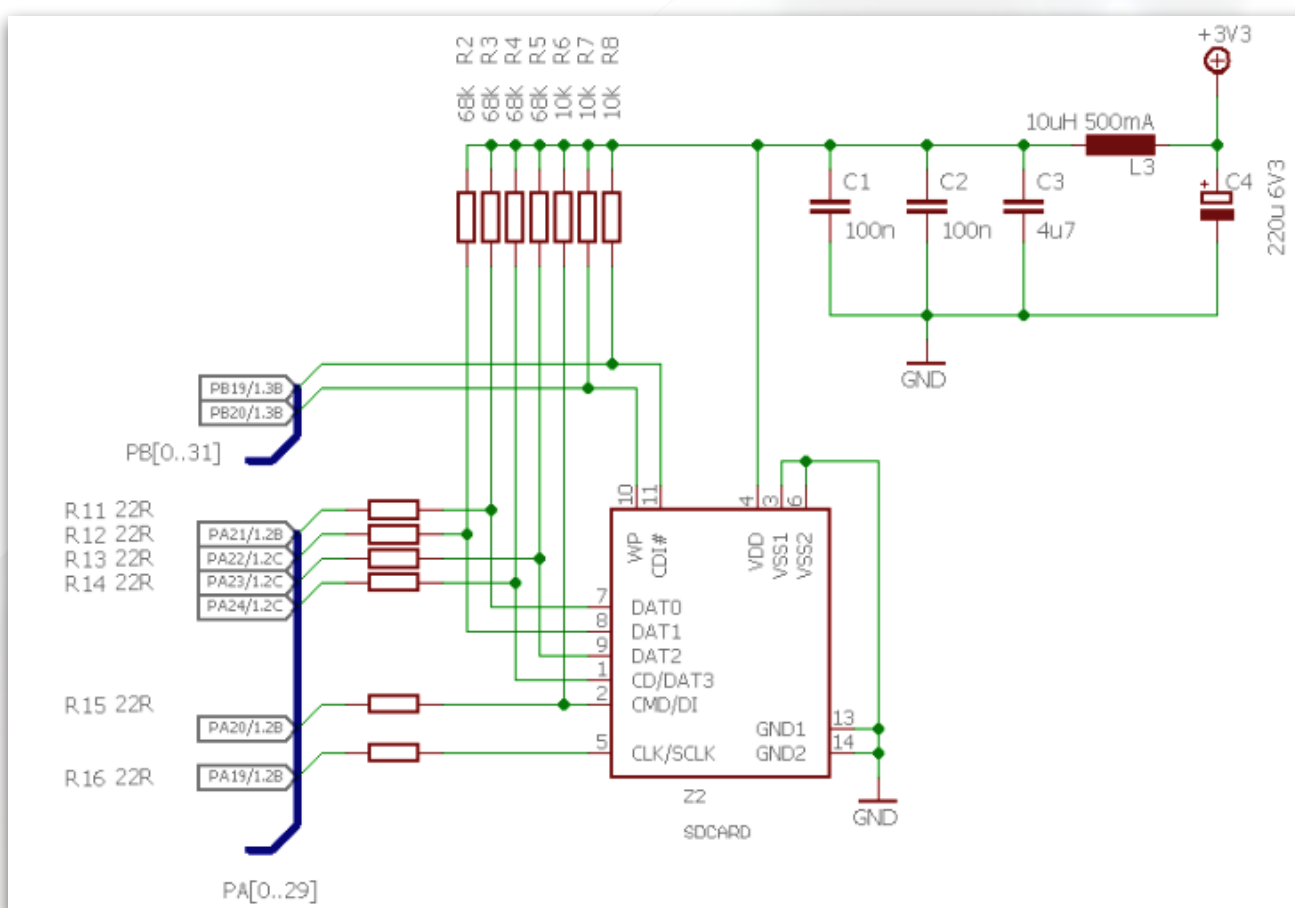
Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm				
B	85 mm				

3.3 Hardware Version Sense

A unique resistor potential divider allows the firmware to detect the hardware version.

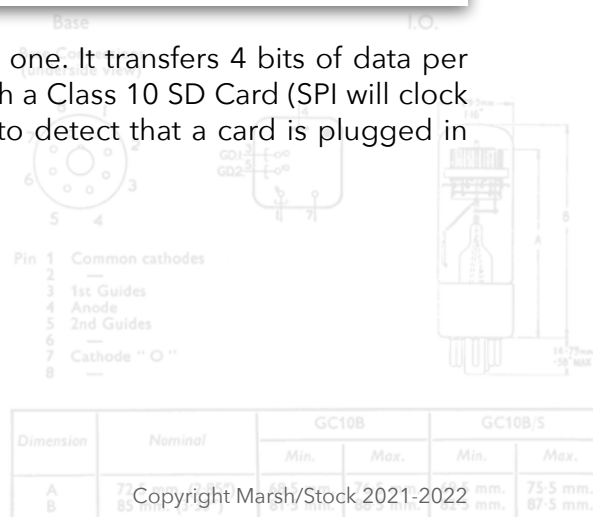


3.4 Storage - SD Card

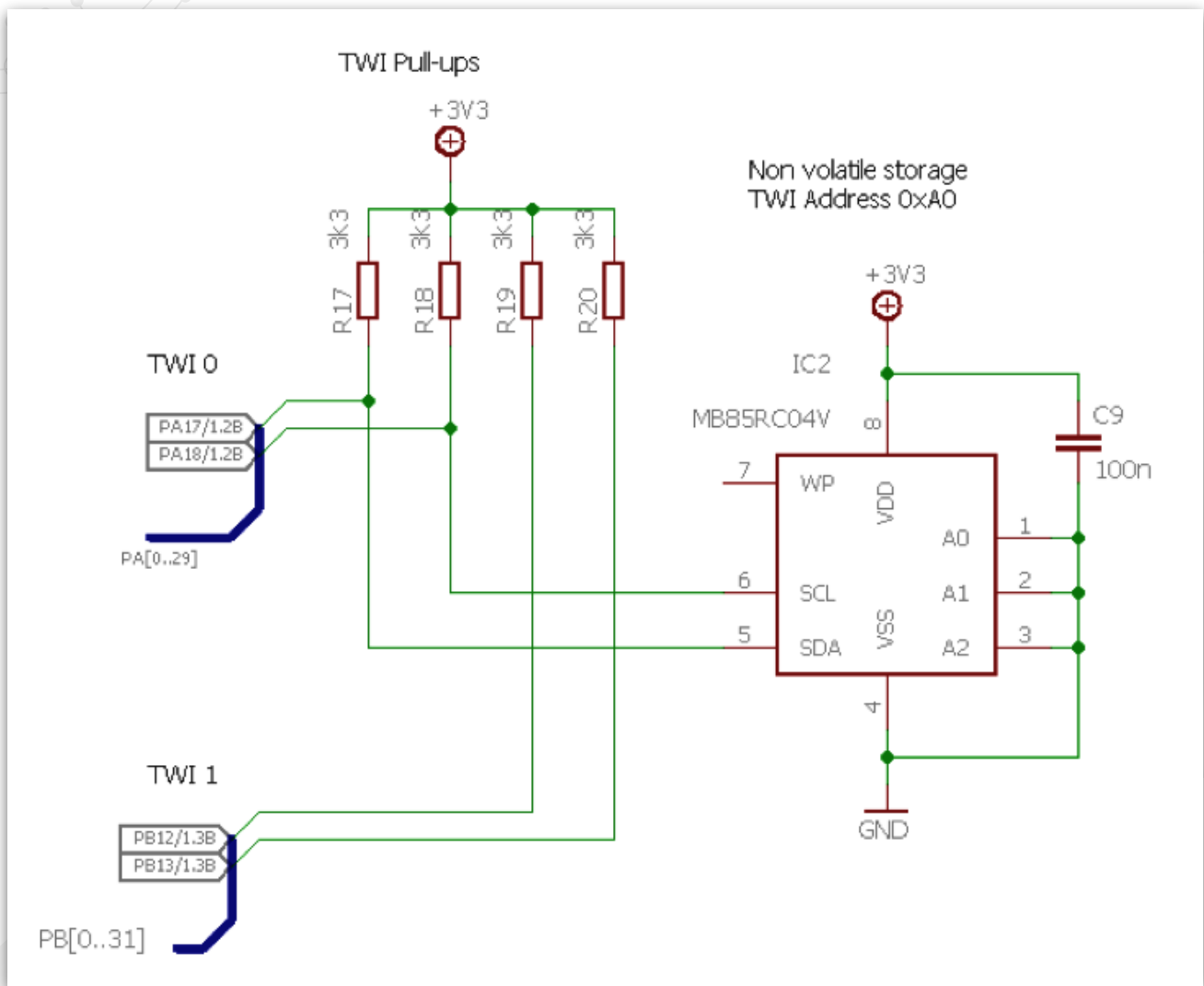


This is a native SD Card interface⁸² and not a slow SPI based one. It transfers 4 bits of data per clock cycle (the SPI transfers 1 bit) and will clock at 50MHz with a Class 10 SD Card (SPI will clock at 10MHz). This is roughly 20 times faster. I/O pins are used to detect that a card is plugged in and whether it is write protected or not.

⁸² SAM3X8C data sheet - Chapter 37

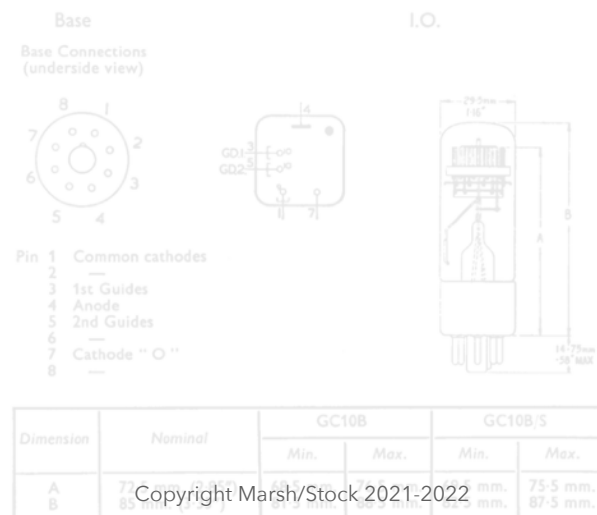


3.5 Storage - Non-Volatile RAM (NVR)

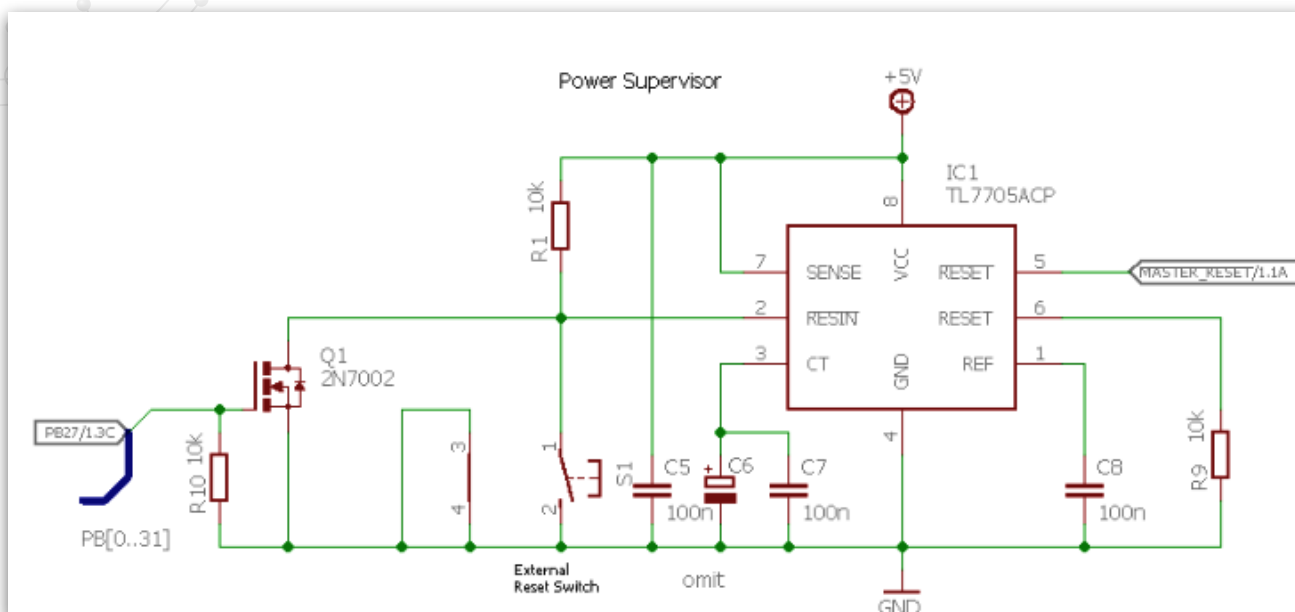


The SAM3X8C has no internal EEPROM so some TWI (I2C) NVR⁸³ is provided to hold data when the clock is switched off and can also be used to pass information between clock faces. The required TWI pull up resistors are here for both TWI ports.

⁸³ <https://www.fujitsu.com/uk/Images/MB85RC04V.pdf>



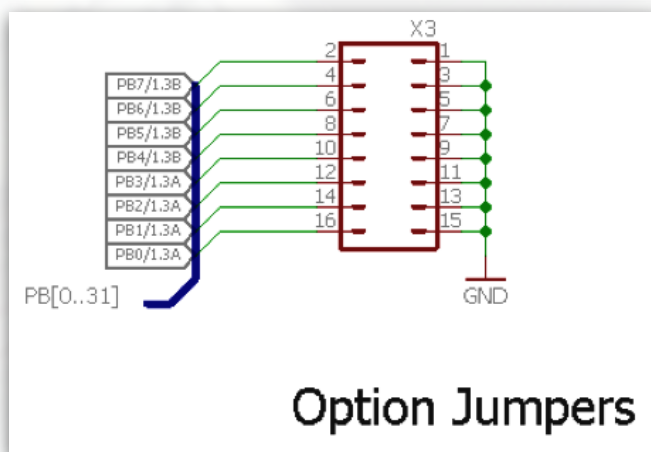
3.6 Reset and Power Supervisor



This handles low 5V detection causing the SAM3X8C to reset. Reset can also be accomplished by the switch or the SAM by pulling the gate of Q1 high.

3.7 Option Jumpers

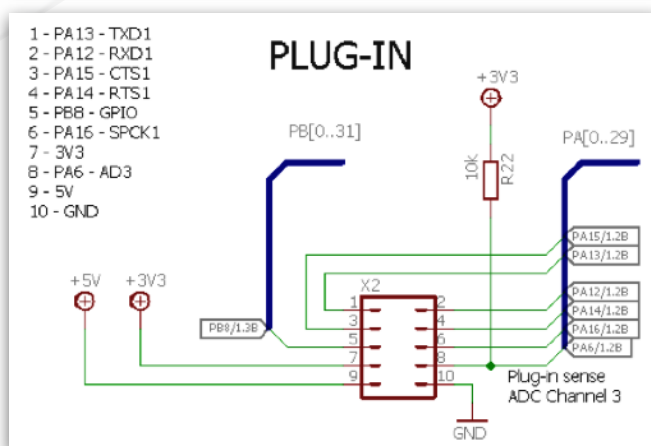
The firmware uses these jumper to configure the clock without having to use any more complex hardware like the SD card. The settings are described in the Operations Guide section.



Option Jumpers

3.8 Plug-in Module Select

Optional add-ons include a GPS receiver or a WiFi adapter to allow the clock to automatically set the date and time. See below.

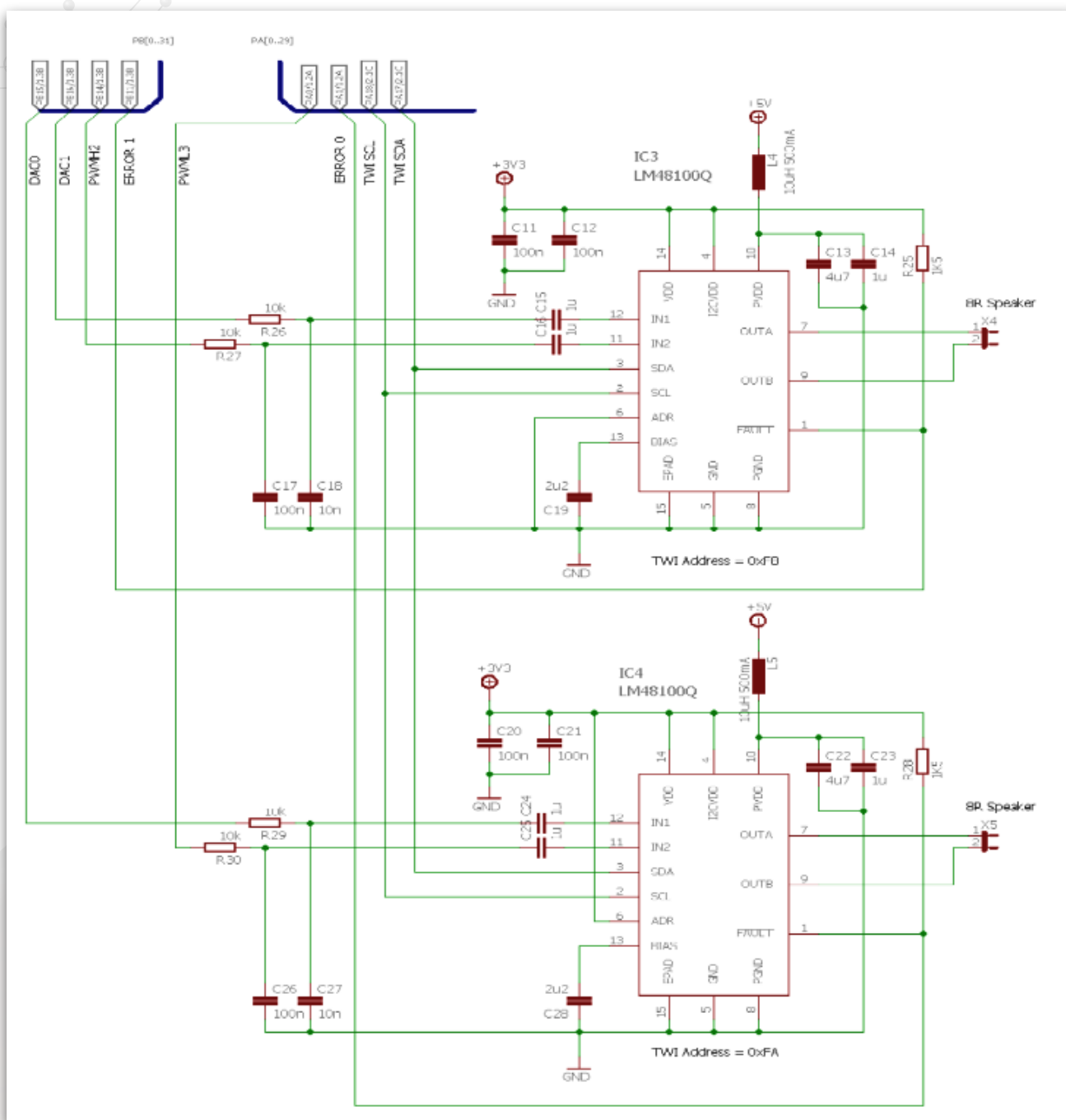


PLUG-IN

- 1 - PA13 - TXD1
- 2 - PA12 - RXD1
- 3 - PA15 - CTS1
- 4 - PA14 - RTS1
- 5 - PB8 - GPIO
- 6 - PA16 - SPCK1
- 7 - 3V3
- 8 - PA6 - AD3
- 9 - 5V
- 10 - GND

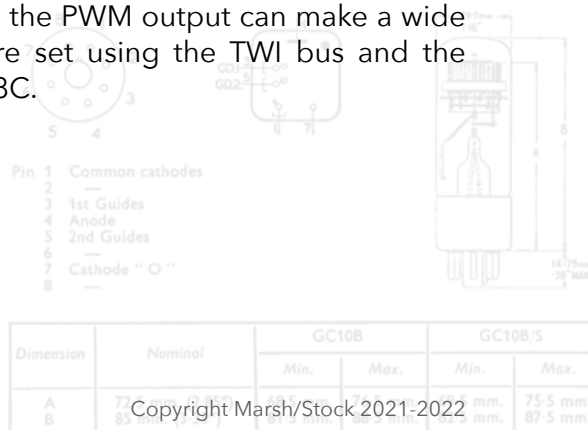
Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	73 mm	73 mm	73 mm	73 mm	73 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

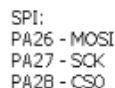
3.9 Audio Amplifiers



The stereo audio amplifier⁸⁴ each take signals from the SAM3X8C digital to analogue (DAC) convertor outputs and from pulse width modulated (PWM) I/O pins. Four signals in total. The DAC outputs are used to play WAV files from the SD card and the PWM output can make a wide variety of simple beep-beep noises. The volume controls are set using the TWI bus and the amplifier chips can report error conditions back to the SAM3X8C.

⁸⁴ www.ti.com/lit/ds/symlink/lm48100q-q1.pdf

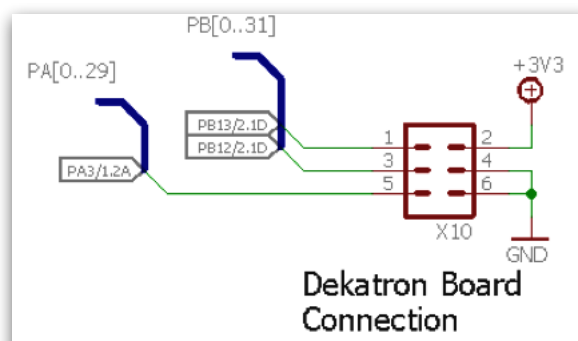




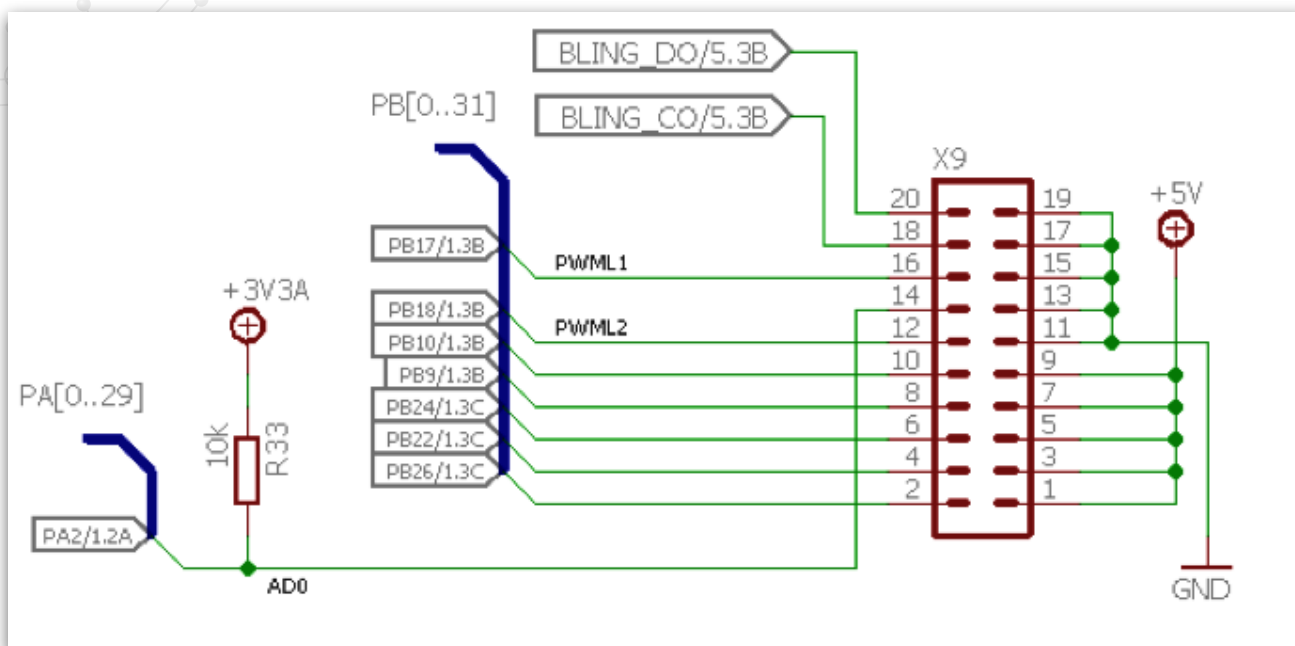
Many of the subsystems on the main board require +3.3V to operate and a local regulator is provided.



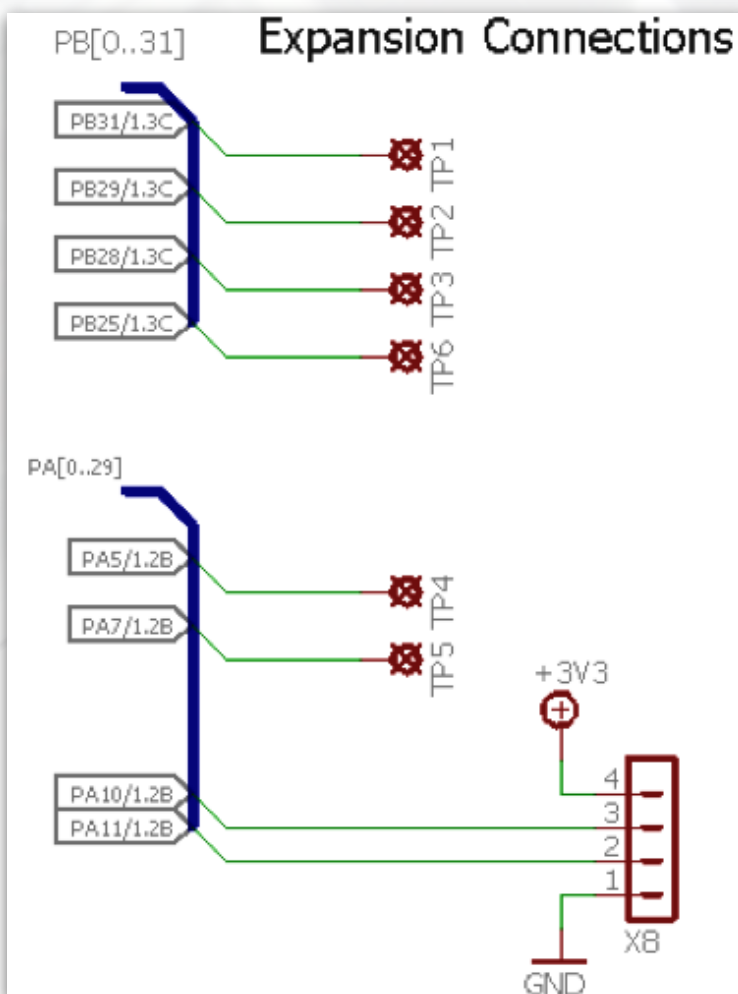
The diagram illustrates the connection for the PSU Board. A blue line labeled `PB[0..31]` is connected to a green line labeled `PB30/1.3C`. This green line is connected to the +5V pin of a 10-pin connector labeled `X6`. The connector has pins numbered 1 through 10. Pin 1 is connected to +5V, and pin 10 is connected to GND. The text `Enable Flyback Convertor` is written near the green line.



3.14 Front Panel Board Connector



3.15 Expansion Connectors (otherwise unused pins)



4 Anode
5 2nd Guides
6 —
7 Cathode "O"
8 —



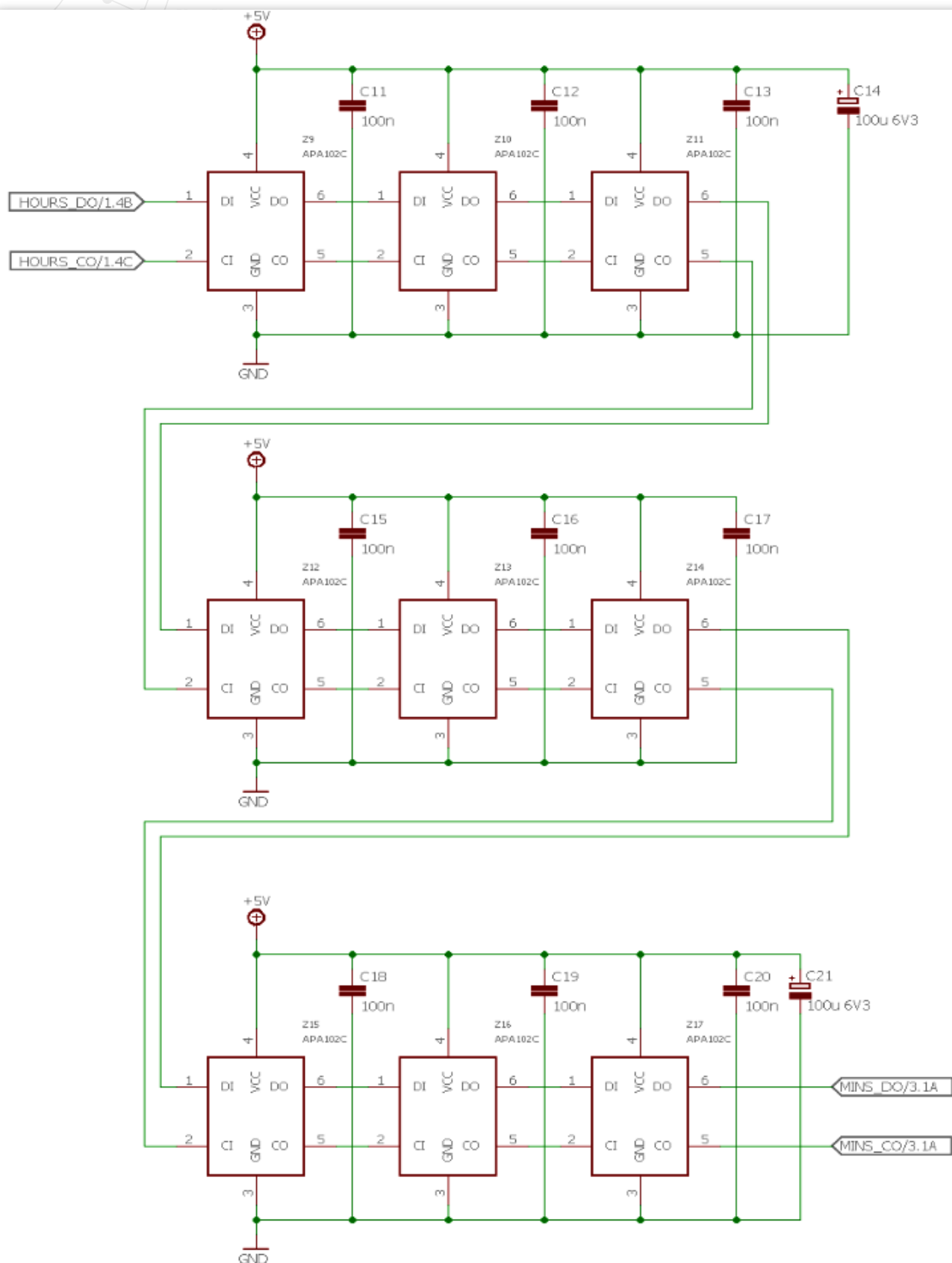
Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75.5 mm	75.5 mm	75.5 mm	75.5 mm	75.5 mm
B	85.5 mm	85.5 mm	85.5 mm	85.5 mm	85.5 mm

Copyright Marsh/Stock 2021-2022



4.1 Bling

Dimension	Nominal	GC108		GC108,5	
		Min.	Max.	Min.	Max.
A	72,5 mm 75 mm	69 mm	75 mm	68,5 mm	75,5 mm
B	85 mm	82 mm	85 mm	82 mm	87,5 mm



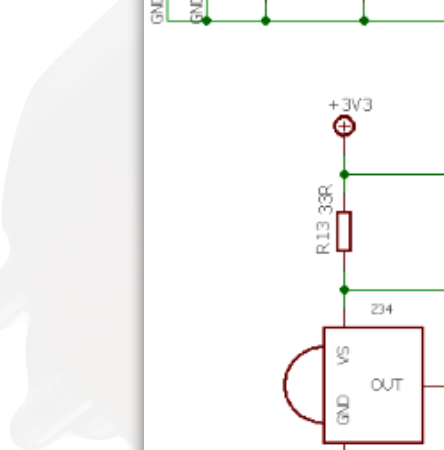
Pin 1 Common cathodes
 2 —
 3 1st Guides
 4 Anode
 5 2nd Guides
 6 —
 7 Cathode "O"
 8 —



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	7.5 mm	7.5 mm	7.5 mm	7.5 mm	7.5 mm
B	8.5 mm	8.5 mm	8.5 mm	8.5 mm	8.5 mm

Copyright Marsh/Stock 2021-2022

4.3 PIR, IR, Encoder, I



the

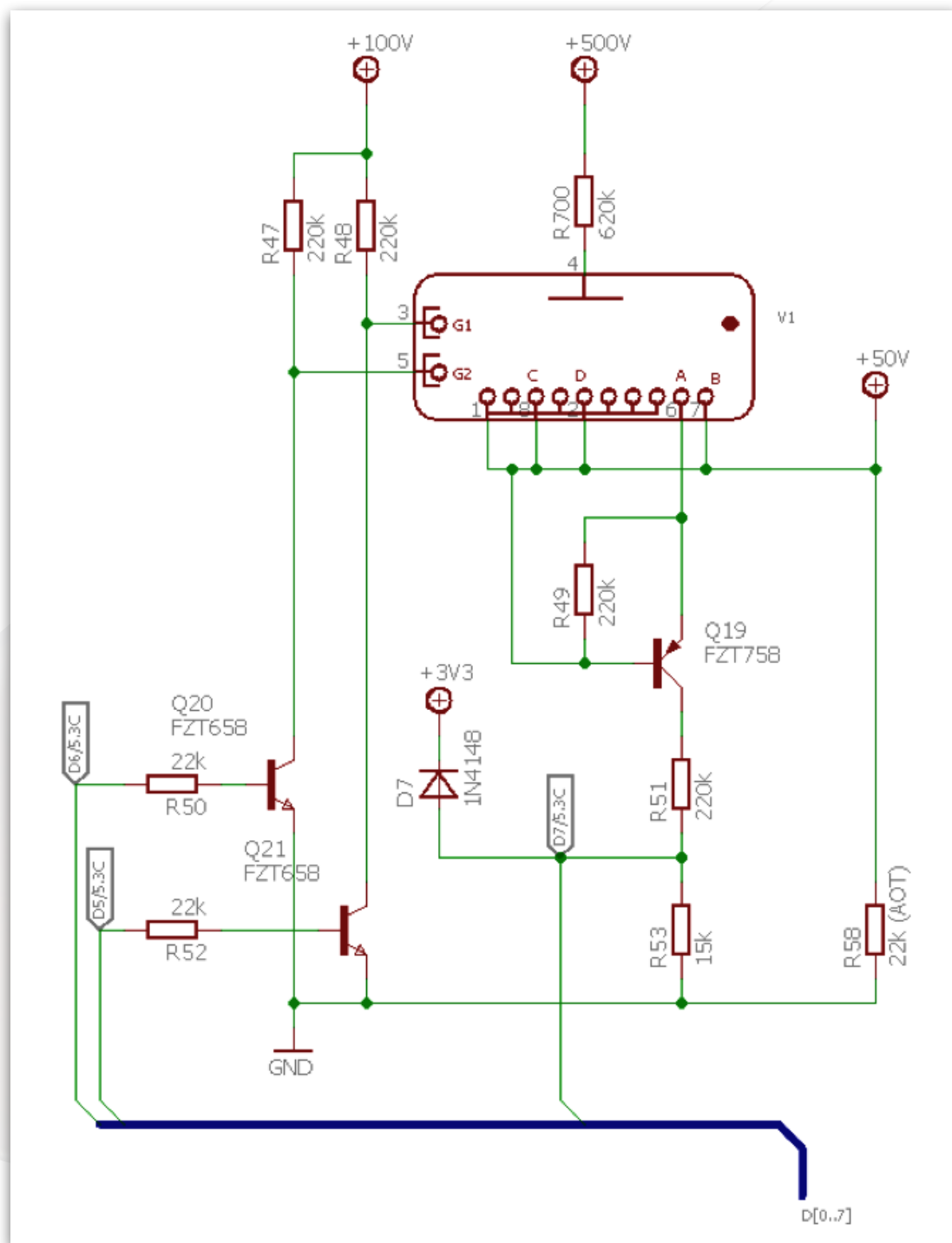
Dimension	Nominal	GC108		GC108,5	
		Min.	Max.	Min.	Max.
A	72,5 mm, 73,95 mm	49,5 mm	75 mm	48,5 mm,	75,5 mm,
B	85 mm, 86,95 mm	62 mm	87 mm	61 mm,	87,5 mm.

Copyright Marsh/Stock 2021-2022

5.0 Dekatron Board

This board carries the seven Dekatrons, their driver transistors, local 50V and 100V power supplies and a ATMEGA324PA microcontroller as a TWI slave that manages all the Dekatron drivers.

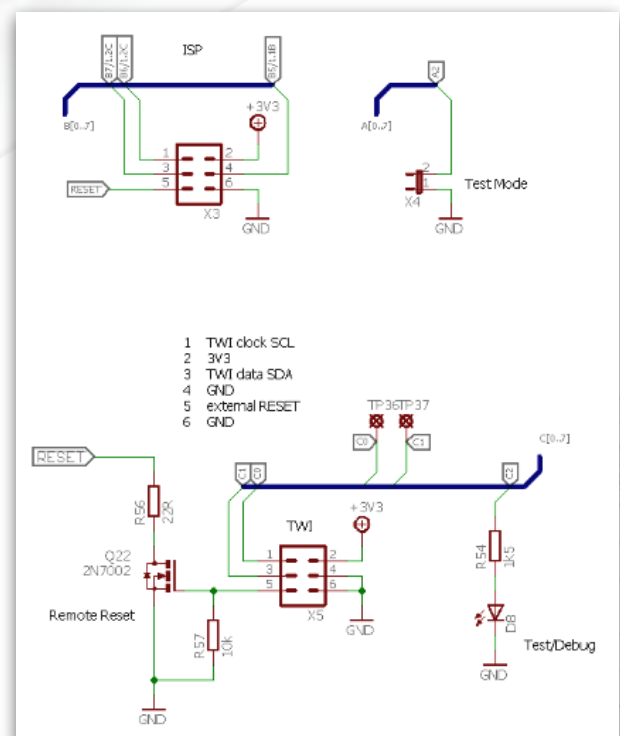
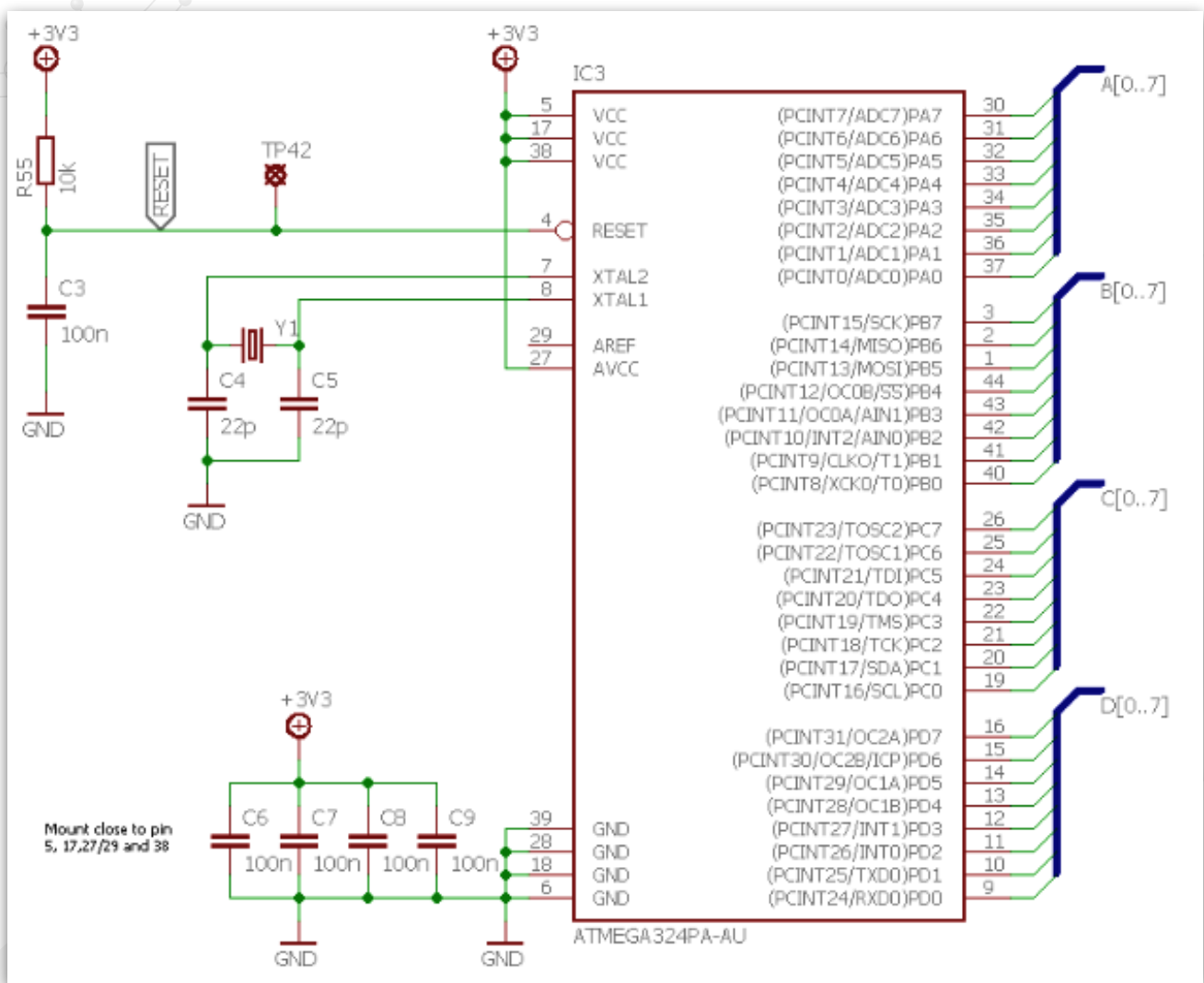
5.1 Dekatron Driver



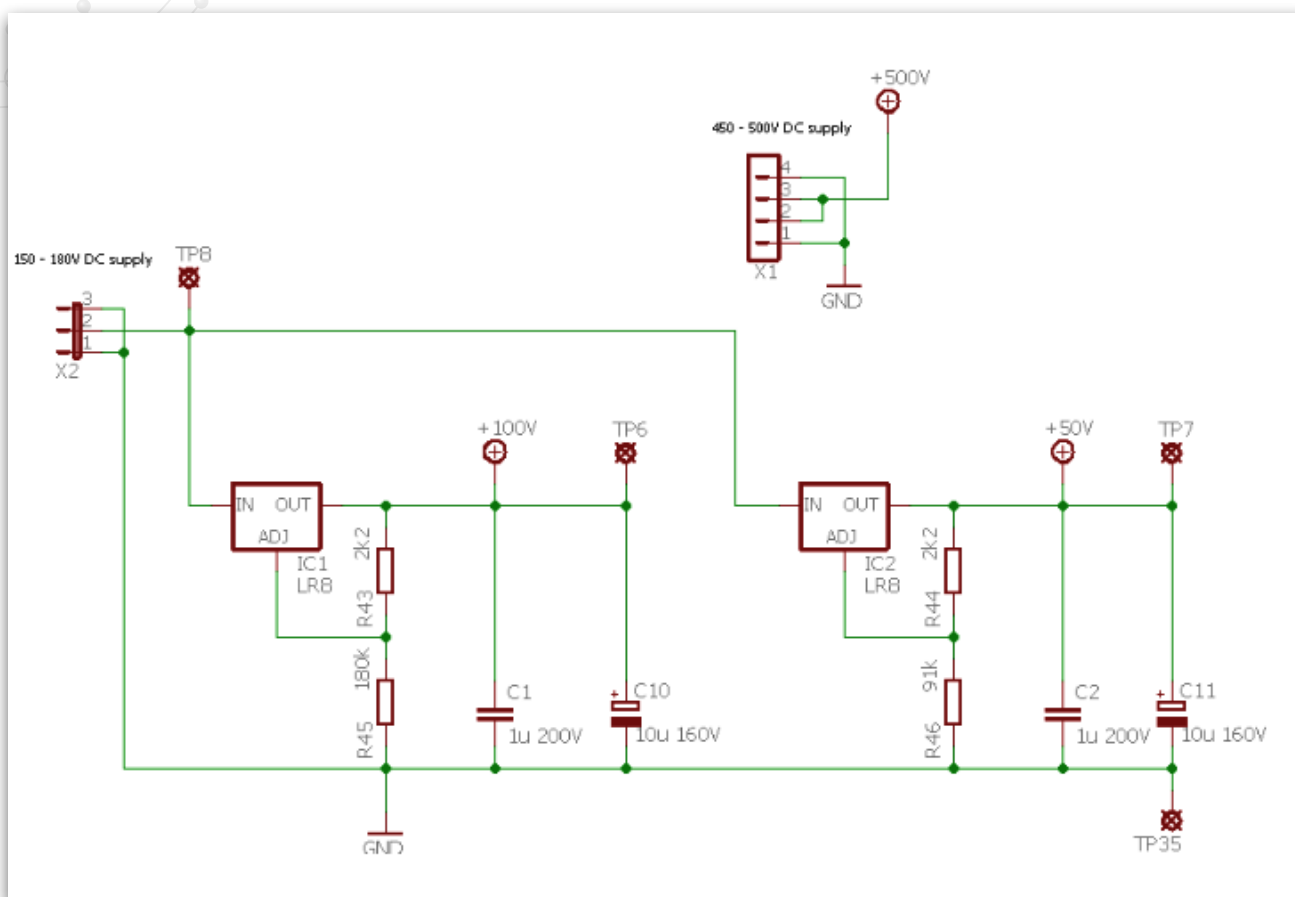
This is one of the seven identical drivers. Q20 and Q21 will pull their respective guide electrodes low cause them to glow. Q19 detects current flow through cathode '0' so is used to determine if that cathode is glowing. The anode resistor R700 is not mounted on the PCB.

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

5.2 Microcontroller



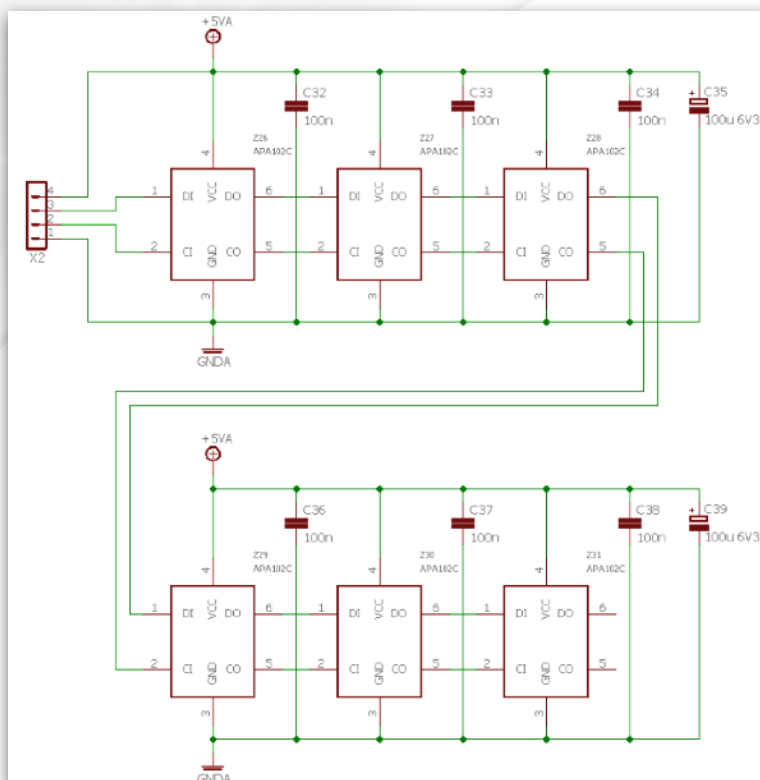
5.3 Power Supplies and Power Connectors



The board connects to the +500V and +70V outputs on the PSU board.

6.0 Downward Bling Board

X2 connects to the front display board.



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	72.5 mm	72.5 mm	72.5 mm	72.5 mm	72.5 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

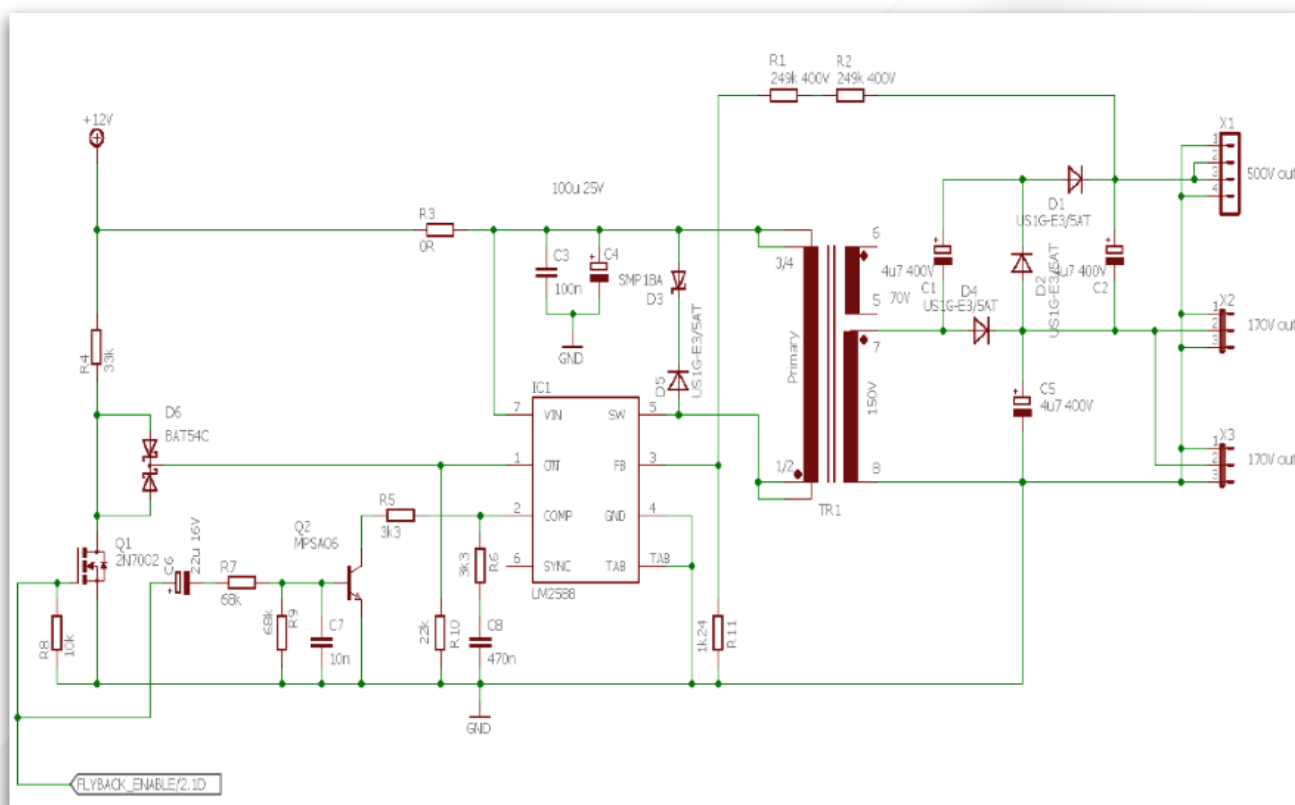
The power supply board carries the bulk of the power supply electronics. The board is designed to mount horizontally in the case so that its 12V DC supply jack is made available through the case rear panel.

The 5V Power supply is a simple step down convertor. X5 is a ground connector that must be used to connect any metal case to the clock ground.

7.2 +500V and +170V Power Supply

The Dekatron anodes require a +500V to operate, the guide drivers require +50V and +100V to operate (on the Dekatron board) so are supplied by the +170V power supply. The MTX90 colons require a +170V supply.

This power supply is a simple flyback convertor using a custom transformer. It is a very robust supply. Q1 acts as a level convertor and makes the logic positive: a 3.3V input is required to switch the convertor on. The circuit around Q2 provides a soft start to the convertor.



Base
Base Connections
(underside view)



I.O.



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm



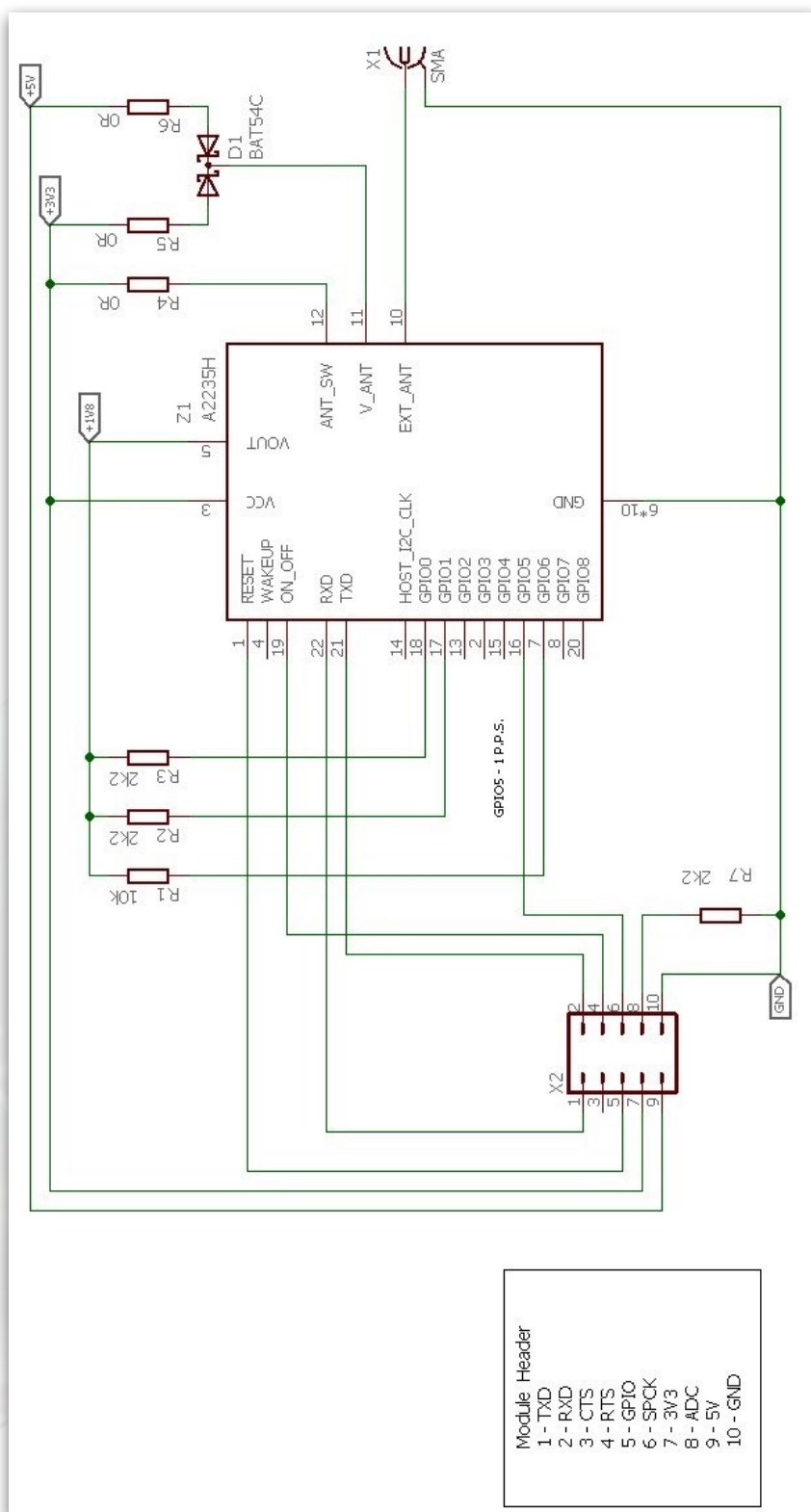
8.1 WiFi Plug In

This uses an ESP32-BIT module. Serial data is transferred to the SAM3X8C. The module can be reset using Q1. Programming the ESP32 is accomplished using the SAM programmer and fitting the program jumper.



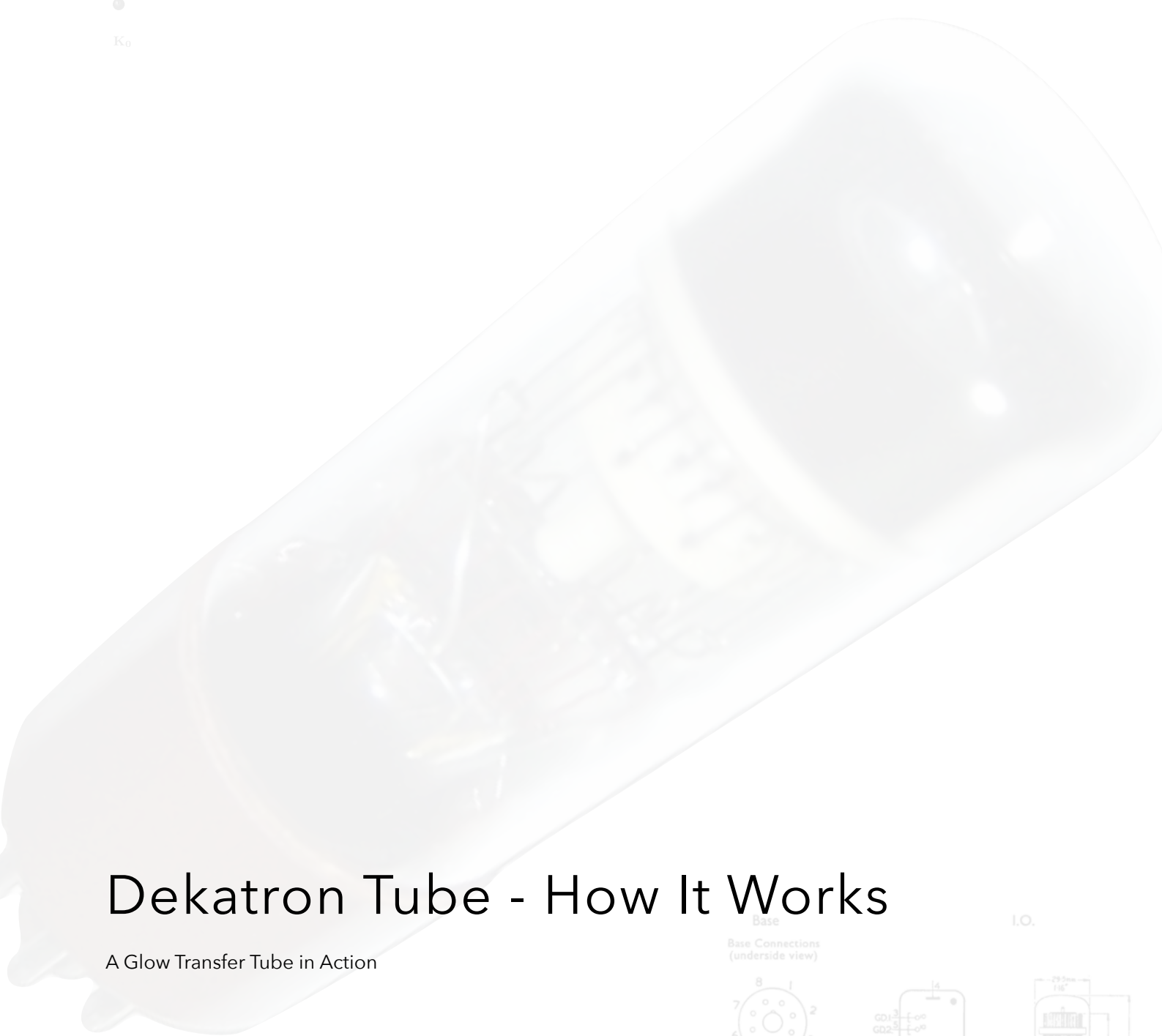
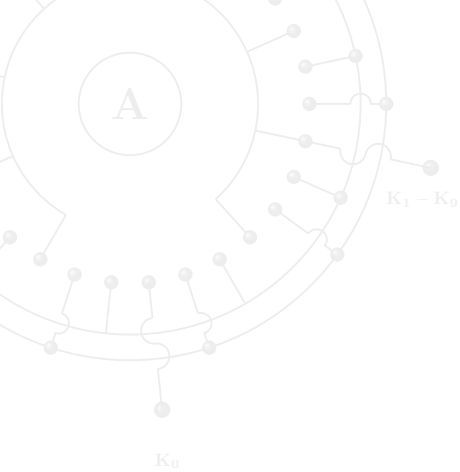
8.2 GPS Plug In

Serial data as NMEA statements are transferred from the GPS module to the SAM3X8C. The use of an external active antenna is selected using R4, R5 and R6. The schematic here shows the GPS plug-in using the A2235H GPS module. A second plug-in may be supplied which uses the A2200A GPS module - see the Sgitheach Commoners manual for complete details of both modules.



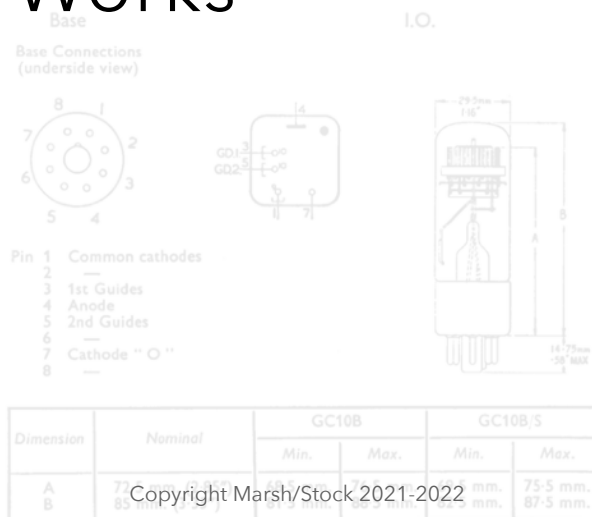
- Pin 1 Common cathodes
 2 —
 3 1st Guides
 4 Anode
 5 2nd Guides
 6 —
 7 Cathode "O"
 8 —

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	73 mm	73 mm	73 mm	73 mm	73 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm



Dekatron Tube - How It Works

A Glow Transfer Tube in Action



An Extract From Dance's "Electronic Counting Circuits"

4.1 INTRODUCTION

One of the disadvantages of trigger tube counting circuits is that ten of the tubes are required in each decade. Great efforts have, therefore, been made to design cold cathode tubes one of which can replace ten trigger tubes in counting circuits; the resulting polycathode tubes are one of the most commonly used devices in medium speed counting equipment. Most of them can count at frequencies up to four thousand pulses per second, some up to twenty thousand, whilst two types can count at up to one million pulses per second.

Most polycathode tubes require a greater H.T. supply voltage than trigger tube counting circuits, but they usually pass a smaller current than trigger tubes. In common with most other cold cathode tubes, the polycathode counting tubes have a very long life when they are correctly used. The number of components required per decade is usually less when a polycathode tube is being used than when trigger tubes are employed, but a more complicated input circuit may be necessary.

4.1.1 Construction

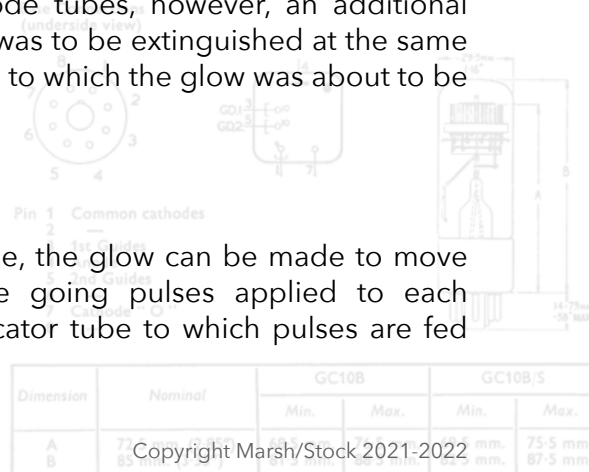
In practice, simple polycathode gas filled tubes for decade counting usually consist of twenty, thirty or forty cathodes placed around a single common anode. Groups of the cathodes are joined together internally so that the number of external connections is reasonably small. A discharge takes place between the anode and one of the cathodes when the tube is in operation and a 'negative' glow is formed. The position of the glow can be observed through the dome of the glass envelope and provides a visual indication of the state of the count. Most of the tubes are fitted into an escutcheon on which the digits are marked, but a few of the tubes do not require an escutcheon.

The tubes have the same basic properties as the simple cold cathode gas filled tubes described in Chapter 3. The striking voltage is normally much greater than the maintaining voltage. If a discharge is taking place to one of the cathodes, it can be transferred to an adjacent cathode by the application of a negative going pulse to the latter. The coupling between the two cathodes takes place automatically by means of the ions provided by the discharge at the first cathode. Cathodes adjacent to the discharge are much more strongly primed than any of the other cathodes.

The coupling components which are used in trigger tube circuits are not required when polycathode tubes are used. No priming electrodes are required, since a discharge is always taking place to one of the cathodes when the tube is in operation. After the transfer has taken place, the discharge to the cathode which was initially glowing is normally extinguished automatically by the fall of anode voltage. In some of the earlier polycathode tubes, however, an additional positive going pulse had to be applied to the cathode which was to be extinguished at the same time that the negative going pulse was applied to the cathode to which the glow was about to be transferred.

4.1.2. Types of Tubes

If a tube contains only ten cathodes around a common anode, the glow can be made to move from cathode to cathode by suitable successive negative going pulses applied to each succeeding cathode in turn. This is the principle of the indicator tube to which pulses are fed



from a counting circuit. Such a simple tube with only ten cathodes cannot actually perform the counting operation itself, since a circuit is said to be able to count only if it changes its state when successive input pulses are fed into the circuit along the same wire. Indicator tubes (as opposed to counting tubes) require pulses to be fed successively to each of the cathodes along different wires; they are described in Chapter 10. In counting tubes at least one additional cathode (known as a transfer or guide cathode) is placed between each two adjacent main cathodes. The discharge does not remain at any transfer cathode for more than a very short time. The main cathode at which the glow rests corresponds to a certain digit which can be read from the escutcheon which surrounds the domed end of most of the tubes. If only one transfer cathode is employed between each two main cathodes, all of the transfer cathodes are normally connected together. Negative going pulses to be counted are applied to all of the transfer cathodes, but the discharge can move only to a transfer cathode which is adjacent to the main cathode which is initially glowing, since no other transfer cathode is appreciably primed by the ions from the discharge. At the end of the pulse the discharge is made to travel from the transfer cathode to the next main cathode, usually by means of a positive bias which is applied to the transfer cathodes.

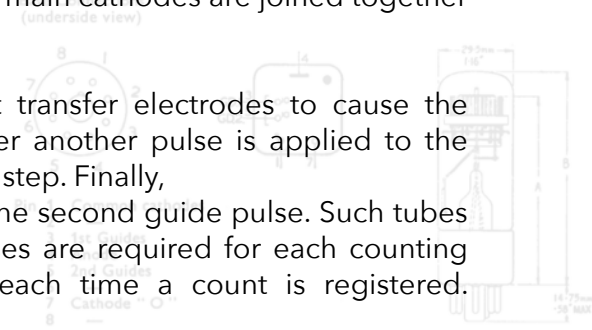
4.1.3 Asymmetrical Tubes

If only ten transfer electrodes are employed and they are all connected together, the electrode geometry must be asymmetrical so that the discharge can travel around the tube only in the desired direction. For example, the main cathodes may be so shaped that the discharge can pass only to the transfer electrode which is one step in a clockwise direction from the main cathode which is initially glowing. The discharge may be prevented from returning to the previous main cathode at the end of the pulse by the effect of a parallel capacitor and resistor from each main cathode to earth. Whilst the discharge remains at a main cathode the capacitor charges and subsequently holds this cathode at a positive potential whilst the discharge is at the succeeding transfer cathode. This technique employing directional main cathodes is used in the S.T.C. G10/241E 'Nomotron' counting tube. In other types of counting tubes both the main and transfer cathodes are directional. The counting speed is then not limited to such an extent by the effect of the cathode circuit time constant, since the latter does not control the direction of rotation of the discharge. The Elesta EZIOB tube employs this principle. Tubes which use directional asymmetrical electrode structures cannot be used for reverse counting (subtraction).

4.1.4 Symmetrical Tubes

In many other types of tube two transfer electrodes are employed between each two of the main cathodes. The tubes themselves are symmetrical in each direction and the direction of the rotation of the discharge is determined by the timing of the applied input pulses. All of the transfer cathodes which are on the clockwise side of the main cathodes adjacent to them are joined together and are known as the first guides (see Fig. 4.1). Similarly all of the transfer electrodes which are on the anticlockwise side of the adjacent main cathodes are joined together and are known as the second guides.

In operation a negative going pulse is applied to the first transfer electrodes to cause the discharge to move one step and a fraction of a second later another pulse is applied to the second guide electrodes so that the discharge moves another step. Finally, the discharge moves to the next main cathode at the end of the second guide pulse. Such tubes are known as double pulse tubes, since two successive pulses are required for each counting operation. Three distinct stepping operations take place each time a count is registered.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

Copyright Marsh/Stock 2021-2022

Examples of those tubes operating on this particular principle are the Mullard Z504S, the Ericsson GC10B and the Sylvania 6476.

Another type of symmetrical tube has ten main cathodes and thirty transfer cathodes surrounding a common anode. The Ericsson GC10D is a tube of this type and is known as a single pulse Dekatron, since only one pulse is fed to the circuit to operate this tube.

4.1.5 Counter and Selector Tubes

The 'zero' main cathode of almost all tubes is brought out to a separate base pin so that an output pulse can be taken from for triggering the next decade. In many counting tubes which are intended for use in simple straightforward counting circuits, the other nine main cathodes are brought out to a single common base pin. Such tubes are known as counter tubes.

Some tubes, such as the Mullard Z504S and the Ericsson GS10C/S, have each main cathode connected to a separate base pin so that an output pulse can be obtained from the circuit after any desired number of counts. This pulse may be used to initiate some external action. Such tubes are known as selector tubes and must be used when electrical readout from any cathode is required. They can perform any function which can be carried out by counter tubes.

Counter tubes can sometimes be manufactured rather more cheaply than the equivalent type of selector tube. In addition, the user of counter tubes has fewer soldered connections to make than the user of selector tubes.

Some tubes are also manufactured in which some but not all of the main cathodes are connected to separate base pins; they are useful in bidirectional counting circuits. A typical example is the Ericsson GC10/4B.

4.1.6 Scale of Twelve

Cold cathode tubes are available with ten main cathodes for decade counting and also with twelve main cathodes for counting on a scale of twelve. Other types of counting tubes, such as EIT tubes and Trochotrons, are not manufactured for scale of twelve operation, presumably because of a lack of demand.

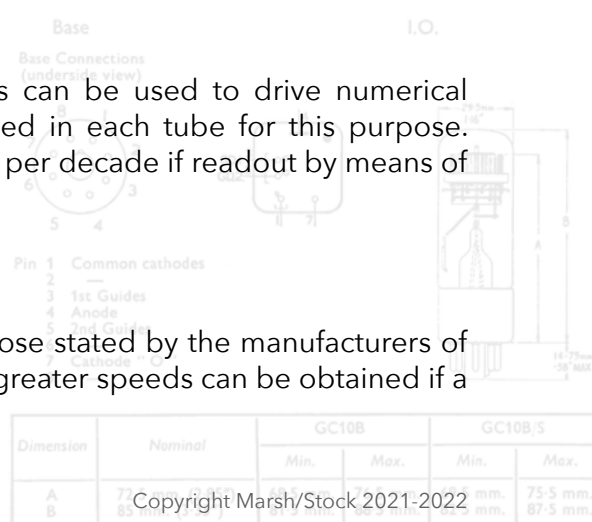
Scale of twelve tubes are useful when pence are to be counted, since one output pulse is provided for each twelve pence counted. Tubes with twelve main cathodes are also useful dividing the number of input pulses by three, four, six or twelve (for example, in conjunction with a decade tube for converting one pulse per second into one pulse per minute).

4.1.7 The Operation of Numerical Indicator Tubes

The Ericsson GCA10G counter and GSA10G selector tubes can be used to drive numerical indicator tubes directly, ten additional anodes being provided in each tube for this purpose. Other cold cathode decade tubes require ten amplifier stages per decade if readout by means of a numerical indicator tube is required.

4.1.8 Maximum Counting Speed

The maximum counting speeds quoted in this Chapter are those stated by the manufacturers of the tube concerned. In some cases operation at considerably greater speeds can be obtained if a



stabilised power supply and close tolerance components are used and if the input pulses are accurately controlled in amplitude and duration. Some manufacturers seem to be rather more conservative in their maximum frequency ratings than others. Generally, however, it is not wise to attempt to appreciably exceed the maximum counting speed quoted by the manufacturers of the tube if a reliable counting circuit is required.

Some indication of the safety margin which is available for any tube when it is operating at a certain speed may be obtained by varying the anode resistor (and hence the anode current) of the tube when it is operating at that speed in order to ascertain the anode current range for satisfactory operation. The actual readings of the anode current should be taken whilst the discharge is stationary in the tube. The anode current should not be allowed to exceed the maximum value for the tube concerned, although input pulse frequencies greater than the maximum recommended for the tube being used may be tried. The anode current range over which satisfactory operation is obtained normally becomes smaller as the operating frequency is increased in the region of the maximum published frequency for the tube. A stabilised H.T. supply is, therefore, normally desirable if the maximum possible operating frequency is required at the expense of circuit simplicity. Tests for reliability of counting using various input pulse amplitudes and/or durations are also very helpful in assessing whether or not a particular tube will operate satisfactorily at a certain counting speed.

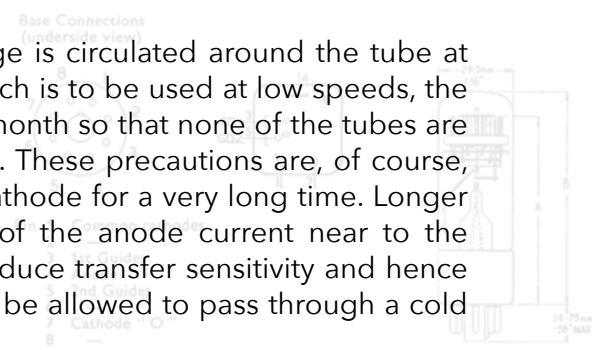
Most double pulse tubes have a maximum counting speed of about 4,000 pulses per second, but other types can operate at up to 10,000, 50,000 or 100,000 pulses per second. The single pulse GC10D and the S.T.C. Nomotron can operate at up to 20,000 pulses per second whilst the Elesta EZ10B and ECT100 can count up to one million pulses per second.

The maximum counting speed may be reduced at high temperatures (above about 60° C) owing to the increased pressure of the gas. Substances which can poison the electrode surfaces may also be given off from the glass envelope if the ambient temperature increases, but these substances will not settle on the cathodes if the discharge is circulated. The maximum speed of operation may also be reduced at temperatures below -15 °C.

4.1.9 Life

The life of most types of polycathode gas filled tube is normally many tens of thousands of hours (tens of years). If, however, the discharge is allowed to remain at one cathode and is never circulated, minute amounts of material may be sputtered from the electrodes and be deposited on the adjacent cathodes. The discharge characteristics are different for sputtered nickel and pure nickel and after the discharge has rested at one cathode for a very long time, longer input pulses may be needed for the operation of the tube. The tube may then be considered to be at the end of its useful life, although longer pulses may be used to drive it at low speeds. The sputtered material from the cathodes may also darken the inside surface of the glass envelope.

Longer life can be obtained from some tubes if the discharge is circulated around the tube at least once per week. Alternatively in a multidecade scaler which is to be used at low speeds, the positions of the tubes may be interchanged about once per month so that none of the tubes are used in the slowest decade for more than a month at a time. These precautions are, of course, required only when the discharge is likely to remain at one cathode for a very long time. Longer life may be obtained from some types of tube if a value of the anode current near to the minimum recommended value is used. This may, however, reduce transfer sensitivity and hence the maximum operating speed somewhat. No current should be allowed to pass through a cold



Base Connections (underside view)

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

Copyright Marsh/Stock 2021-2022

cathode tube in the reverse direction, since this will probably result in damage to the surfaces of the cathodes.

4.1.10 Design for Optimum Anode Current

The current flowing through the tube should be within the limits specified by the manufacturer. A current greater than the specified maximum value leads to excessive sputtering of material from the cathodes and to short life, whilst a current below the specified minimum value will result in reduced transfer sensitivity and hence to unreliable counting or reduced operating speed. High speed tubes generally require a greater anode current than the simple 4 kc/s double pulse tubes.

The potential across a cold cathode tube is almost independent of the current flowing through the tube provided that it is being operated in the 'normal' region of the characteristic. The current which flows through the tube can be calculated by applying Ohm's Law to the circuit :

$$V_b - V_m = I_a(R_a + R_h)$$

V_b is the H.T. supply voltage

V_m is the tube maintaining voltage

I_a is the tube anode current

R_a is the anode resistance

R_h is the cathode resistance

A minimum value for the H.T. supply voltage is quoted by the tube manufacturers. If a smaller supply voltage than this is used, the tube may not strike when the voltage is first applied. Since the tubes are not primed, there may be a delay of up to about a minute before they strike if a value of H.T. supply voltage near to the minimum recommended value is employed.

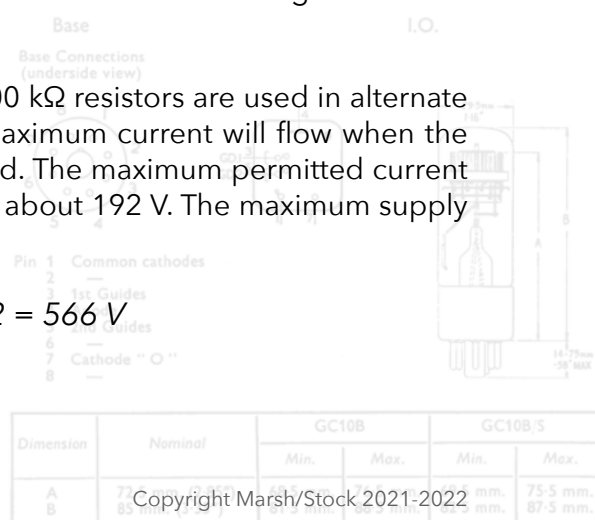
There is normally no upper limit to the H.T. supply voltage which can be used provided that the cathode current is kept within the rating of the tube by a suitable choice of resistor values. In most cases the H.T. supply need not be stabilised.

When the value of R_a has been chosen, the maximum value of the supply voltage may be calculated from the above equation by substituting for I_a the maximum permissible value of the tube current and for R_k the smallest value of resistor which is to be used in any of the cathode circuits of the tube. (The maximum current flows in that cathode circuit which has the smallest value of cathode resistor.)

Similarly the minimum value of the supply voltage may be calculated by using the same equation and substituting in it the minimum permitted value of the tube current and the largest value of cathode resistor which is to be used in any cathode circuit.

In the case of the GS10C/S, for example, let us assume that 100 k Ω resistors are used in alternate cathode circuits and that the anode resistor is 680 k Ω . The maximum current will flow when the discharge rests at one of the cathodes which is directly earthed. The maximum permitted current for the GS10C/S tube is 550 μ A and the maintaining voltage is about 192 V. The maximum supply voltage is, therefore, given by :

$$V_{b \text{ max}} = (0.00055 \times 680,000) + 192 = 566 \text{ V}$$



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

Copyright Marsh/Stock 2021-2022

The minimum recommended anode current is 250 μA . The anode current will have its smallest value when the discharge is resting at one of the cathodes connected to a 100 k Ω resistor. The minimum supply voltage is therefore given by

$$V_{b \min} = 0.00025(680,000 + 100,000) + 192 = 387 \text{ V}$$

This value is slightly below the minimum recommended value for reliable striking (400 V). The mean of the maximum and minimum values of H.T. supply voltage, that is 475 V, may be used as the H.T. supply voltage design value.

It might appear that this mean value could vary by +90 V or have a 64 V R.M.S. ripple. If variations of this magnitude did occur, however, the change in current flowing through the cathode resistors would cause a variation of about 110 V on the nominal output voltage of 30. This might well cause unsatisfactory operation of the circuits into which the output voltage is fed.

The circuits of some of the tubes are fairly critical and it is therefore usually advisable to employ a circuit which has been designed and thoroughly tested by the manufacturers of the tube concerned as the basis of any design for a piece of counting equipment.

4.2 4 kc/s DOUBLE PULSE TUBES

Some of the most commonly used cold cathode counter tubes function on the double pulse principle. The electrode structure of a double pulse selector tube is shown in Fig. 4.1. The anode is a circular metal disc placed near to the domed end of the tube (Plates 6 and 7). Thirty identical rods of small diameter are placed symmetrically around the anode; they are the main cathodes and the transfer cathodes. There are two transfer electrodes between each two main cathodes. All of the first guides are joined together and all of the second guides are joined together. In a counter tube (as opposed to a selector tube) the main cathodes K1 to 1(9 are also brought out to one common external connection.

The basic circuit for the normal operation of the tube is shown in Fig. 4.2. The tube may be represented as shown, the circular structure being illustrated as a linear one for convenience. The square brackets near each set of transfer electrodes indicate that there are more than one first guide and more than one second guide in the tube. The main cathodes are normally at earth potential whilst both sets of guide cathodes have a quiescent positive potential which is determined by the source of bias voltage. When an H.T. supply of over 400 V is connected to the circuit, one of the main cathodes will strike preferentially to any guide cathode, since the guide bias renders the anode to main cathode potential greater than the potential between the anode and any guide. As soon as ignition has taken place at any one main cathode, the potential between the anode and that main cathode will drop from the striking voltage to the

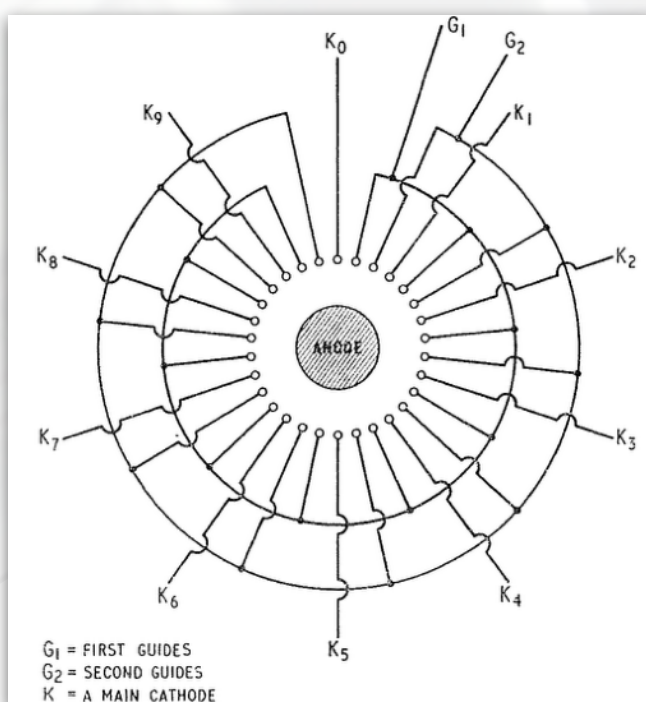


Fig. 4.1 The electrode structure of a double pulse selector tube

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

maintaining voltage owing to the fall of potential across the anode resistor, R_a . The anode potential is then below the striking voltage to any other cathode and, therefore, the discharge will occur at only one main cathode.

4.2.1 Double Pulse Counting

The counting operation is performed in three stages. A negative going pulse is first applied to all of the first guide electrodes so that they fall in potential, to a value which is appreciably below earth potential. In a typical double pulse tube with a maintaining voltage of 190 V, the priming effect of the ions from the discharge at the main cathode reduces the striking voltage at the two adjacent guide electrodes to approximately 200 V, whilst the striking voltage at the cathode three positions away is reduced to about 250 V. If the glowing cathode is earthed, the potential difference between the anode and the first guide will be 200 V when the first guide potential has fallen to -10 V. The first guide which is adjacent to the main cathode, therefore, commences to strike when it is at this potential. No other first guide is sufficiently primed for striking to occur.

As the guide potential falls further, the current to the guide increases so that the operating point moves to the flat portion of the anode voltage/anode current characteristic where the voltage between the anode and the first guides is almost constant and independent of the current flowing. The anode potential falls with the potential of the first guides so that the potential difference between these electrodes is constant and equal to the maintaining voltage for the tube concerned. This fall of anode voltage results in the voltage between the anode and the main cathode falling below the maintaining voltage of this gap and the discharge to the main cathode is, therefore, extinguished.

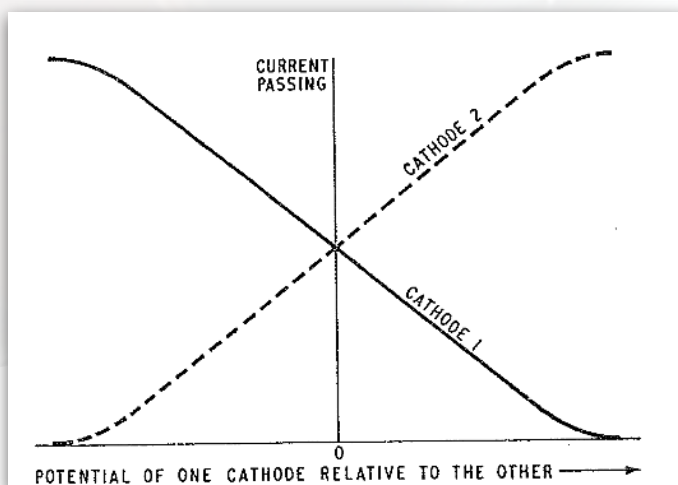


Fig. 4.3 The transfer characteristic for two similar cathodes in a cold cathode tube

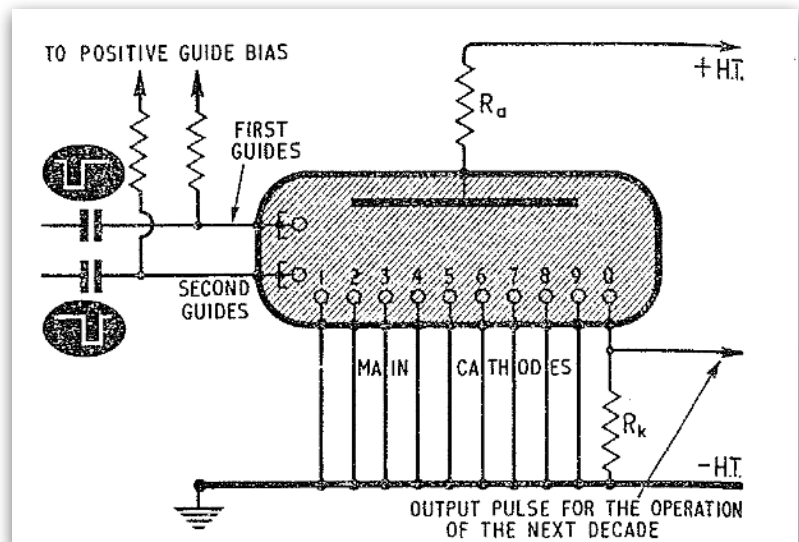


Fig. 4.2 The basic circuit for the operation of a double pulse tube

The transfer characteristic of the tube for any two adjacent cathodes is of the form shown in Fig. 4.3. It can be seen that as the current passing to the cathode which is about to glow increases, the current passing to the cathode which was initially glowing decreases and the total anode current remains more or less constant.

It should be noted that the guide electrode, which is one position in an anticlockwise direction from the discharge at the main cathode, is a second guide. The discharge shows no tendency to move in an anticlockwise direction to this electrode, since the second guide electrodes are still

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

receiving a positive bias.

The discharge has thus moved one step in a clockwise direction to the first guide and now primes the succeeding second guide. If the pulse now ceased, however, the discharge would return to its original position at the main cathode owing to the positive guide bias. When the discharge has been fully transferred to the first guide electrode, a negative going pulse is applied to the second guide electrodes so that their potential is reduced to a value which is appreciably below that of the main cathodes and which is approximately equal to that of the first guides which are still receiving a pulse. The pulse to the first guides terminates soon after the application of the second guide pulse and the first guide potential rises towards the bias voltage. The anode potential also rises so that the anode to first guide voltage is kept constant at the maintaining voltage of the tube. Soon the anode to second guide primed striking potential of about 200 V is reached. The second guide which is primed then strikes and the anode voltage falls until the anode to second guide potential is equal to the maintaining voltage of the tube. The anode to first guide potential is now below the maintaining voltage for this gap and the discharge at the first guide is extinguished. The discharge has now moved two positions clockwise.

Finally, when the second guide pulse terminates, the anode voltage again rises, since the anode to second guide potential tends to remain constant at the maintaining voltage. When the potential of the second guides reaches about 10 V above earth whilst returning to the quiescent bias potential, the discharge will move one further step in a clockwise direction to the next (primed) main cathode. There is obviously no tendency for the discharge to move in an anti-clockwise direction to the first guide, since this electrode is at a positive potential with respect to the main cathode and the anode to first guide striking potential is, therefore, not reached. One of the purposes of the guide bias is to cause transference of the discharge to the main cathode at the end of the second guide pulse. The three successive stepping operations have now been completed and one count has been registered.

The guides are used to determine the direction in which the discharge rotates in the tube. If the second guides receive a negative going pulse and subsequently the first guides receive a similar negative going pulse just before the termination of the second guide pulse, the discharge will move in an anti-clockwise or reverse direction. Circuits for addition or subtraction can, therefore, be constructed using double pulse tubes.

When the anode current flows to the zero cathode, K0, the voltage produced across the cathode resistor (see Fig. 4.2) can be used to trigger the next decade. The output pulse is not suitable for feeding decade. The output pulse is not suitable for feeding directly to the counting tube of the next decade, but must be fed into a coupling circuit which amplifies must be fed into a coupling circuit which amplifies it, changes its polarity and converts it into the required double pulse.

4.2.2 Cathode Resistor and Guide Bias Values

If the cathode resistor is small in value compared with the anode resistor, an increase in the value of the cathode resistor will not appreciably affect the magnitude of the current passing through the tube. The output voltage available at the cathode will therefore be proportional to the value of the cathode resistor if the latter is small. As the cathode voltage increases with increasing values of cathode resistor, however, it will approach the bias potential of the two sets of guides. Further increases in the value of the cathode resistor then merely cause more of the anode current to flow to the adjacent guides and less to the main cathode (see Fig. 4.3). Any increase in the output voltage as the value of the cathode resistor increases is then negligible. Another effect

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

occurs if the guides receive a large positive bias (say +100 V) in an attempt to prevent the above effect from limiting the output voltage. The maintaining voltage of the tube is virtually constant and as the cathode at which the discharge is occurring becomes more positive, the potential of the anode will increase by the same amount as that of the cathode. The non-glowing cathodes remain at earth potential, however, and therefore the potential between them and the anode has increased. The discharge may, therefore, spread somewhat to the adjacent cathode gaps and these may break down at an anode-cathode potential difference of about 250 V which corresponds to a cathode potential of less than +60 V. The glow is especially likely to jump back to the previous main cathode, however, if that cathode has not completely deionised.

The optimum value of the guide bias is normally a compromise between a high value which would result in limited tube life and a low value which would limit the output pulse amplitude. A guide bias of about +40 V is about the maximum which is recommended for 4 kc/s tubes ; if this value of bias is used, an output pulse of about 35 V across a 150 k Ω resistor can then be obtained. Under carefully controlled conditions output pulses of 65 V across 200 μ A cathode resistors have been obtained with a guide bias of +65 V, but these operating conditions are not recommended for general use.

4.2.3 Negative Cathode Bias

An increased output pulse amplitude can be obtained by returning the cathode load resistors to a source of negative voltage. For instance, if they are returned to a -20 V line, a 50 V output signal can be obtained if the cathodes rise to +30 V when passing a current. The anode potential will then rise to only +220 V (for a tube with a maintaining voltage of +190 V), so that there is no danger of an adjacent cathode striking. The bias potential of the output cathode may also be employed to bias the succeeding valve in the coupling stage to cut off, the cathode of the coupling valve being earthed.

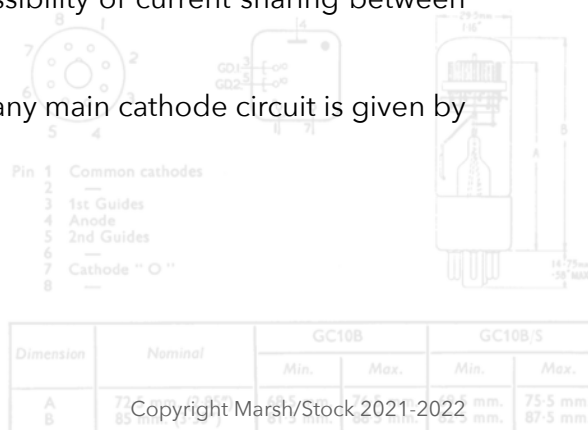
When a negative bias is applied to the output cathode(s), the minimum amplitude of the pulses applied to the first guides must be increased by an amount equal to the negative bias. This ensures that the guides fall in potential by an amount which is sufficient to cause reliable transfer of the discharge from the negatively biased cathodes to the first guides.

It is not wise to return the output cathode load resistor to a bias voltage which is more negative than -20 V, or the discharge may transfer correctly from the output cathode to the first and second guides and then suddenly jump back to it, as it will still be primed somewhat and is at a greater negative potential than the succeeding main cathode. Most of the circuits published by the Ericsson Company for their tubes have the output cathodes returned via the cathode resistor to a -20 V line. The Mullard/Philips circuits employ a -12 V line for the same purpose.

The potential of any cathode which is used to generate an output pulse should not be allowed to rise to within ten volts of the positive guide bias potential or the glow discharge may fail to transfer from it to the succeeding first guide owing to the possibility of current sharing between the main cathode and the preceding second guide electrode.

The maximum recommended value of the cathode resistor in any main cathode circuit is given by the equation:

$$R_k = \frac{(V_g + V_k - 10)R_a}{(V_b - V_m - V_g + 10)}$$



the output voltage for any value of R_k is given by:

$$V_{out} = \frac{(V_b - V_m + V_k)}{(R_a + R_k)}$$

where V_g is the positive guide bias and V_k is the output cathode negative bias. The other symbols are as defined earlier for equation (1) in Section 4.1.10.

If the values of $V_g = +40$ V and $V_k = -12$ V (as recommended in Mullard/Philips circuits) are used, the maximum value of R_h is found to be 140 k Ω . The preferred value of 120 k Ω is, therefore, recommended and output pulses of 30 V are obtained.

4.2.4 Output Pulse Shape

When the discharge is transferred to the main cathode, the current does not increase very suddenly, but depends on the instantaneous value of the potential difference between the main cathode and the guide from which the discharge is transferred (see Fig. 4.3). The rate of rise of the leading edge of the output pulse is approximately equal to the rate of decay of the trailing edge of the second guide pulse. The transfer from the second guide to the main cathode may not take place at the same potential difference at various positions in the tube and there can be jitter in the time at which the leading edge occurs. The trailing edge of the output signal is produced by the leading edge of the first guide pulse which is usually quite sharp; the trailing edge of the output signal is, therefore, more suitable for use when the pulse is to be employed as a form of time marker.

The duration of the output pulse is approximately equal to the time during which neither set of guides is receiving a pulse.

4.2.5 Input Pulse Requirements

It is essential that the pulses applied successively to the two sets of guides should be of a suitable amplitude and duration and that they should be correctly timed with respect to each other. Transfer can be effected by a number of types of waveform, but for maximum speed of operation the optimum waveforms are rectangular pulses which have a slight overlap in time as shown in Fig. 4.4. It is not usually practical to construct input circuits which convert a single input pulse into two almost perfectly shaped rectangular overlapping pulses for the guides, although a suitable circuit for this purpose has been described. In actual practice pulses similar to those shown in Fig. 4.5 are usually used. Although the second guide pulse is very different in shape from the ideal pulses of Fig. 4.4, the only disadvantage in the use of pulses of the shape shown in Fig. 4.5 is that the maximum operating speed of the tube is slightly reduced. The pulse for the second guides is normally obtained by integrating the first guide pulse by means of a simple resistance-capacitance circuit.

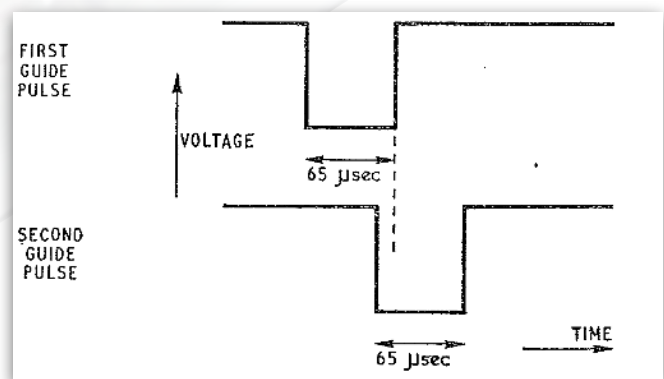


Fig. 4.4 Ideal rectangular negative going pulses for feeding to the first and second guides

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

If the pulse to the second guides is applied too soon after the pulse to the first guides, the discharge will not have been fully transferred to the first guides and the preceding second guide will still be primed to some extent. The discharge will be pulled forward to the first and second guides at the same time as it is being pulled backwards to the preceding second guide with the probable result that no transfer at all will take place. If the first pulse terminates appreciably before the beginning of the second pulse, the glow will transfer to the first guide, but during the interval between the two guide pulses it will return to the main cathode from which it came. When the second guide pulse is applied, it will move one further step in an anti-clockwise direction to the second guide preceding the main cathode at which the discharge initially rested. Finally at the end of the second guide pulse the discharge will return to the initial position at the main cathode.

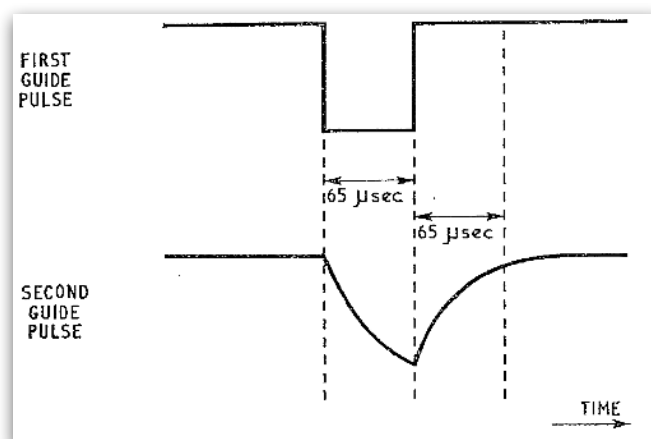


Fig. 4.5 Practical waveforms for 4 kc/s double pulse tubes. The second guides pulses are obtained by integrating the pulses applied to the first guides

New tubes may count correctly if a small gap is present between the two guide pulses, but a minimum overlap of one or two microseconds is essential if the tube characteristics have been affected by long stand-by periods.

4.2.6 Pulse Duration

The pulse applied to the first guides must be of sufficient duration for three successive processes to take place. First of all, the discharge must be established at the first guides and the anode to main cathode glow must be extinguished. Secondly, the priming of the succeeding second guide electrode must take place and, finally, the second guide preceding the main cathode which was initially glowing must have time to become deionised. These processes take a total time of about 65 μsec, but a nominal first guide pulse width of about 75 μsec is recommended so that an adequate allowance can be made for tolerances, etc.

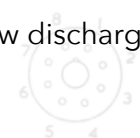
Similarly the pulse applied to the second guide electrodes must be of a sufficient duration for three similar processes to occur. The discharge must be formed at the second guide electrodes and extinguished at the first guides. Secondly, the succeeding main cathode must be primed. In addition the main cathode which was previously glowing must be deionised, but this last process can occur during the total time in which the pulses are applied to the first and second guide electrodes. The required pulse duration to the second guides is about the same as that to the first guides and a minimum of about 75 μsec is recommended.

If the guide bias is 40 V +10 %, it is recommended that the glow discharge should remain at each main cathode for at least 100 μsec.

4.2.7 Maximum Counting Speed

The total time occupied by the three separate steps is $75 + 75 + 100 = 250 \mu\text{sec}$. The maximum speed of operation is, therefore, about 4,000 pulses per second.

Base Connections
(underside view)



Pin 1 Common cathodes
Pin 2 2nd Guides
Pin 3 Cathode "O"
Pin 4 1st Guides
Pin 5
Pin 6
Pin 7
Pin 8



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Copyright Marsh/Stock 2021-2022

4.2.8 Pulse Amplitude

The use of guide pulses of fairly large amplitude generally results in the most reliable counting. The upper limit of the pulse amplitude is set by the breakdown of the main cathode to guide gap which occurs at a potential of about 140 V. The guide to main cathode voltage should always be appreciably less than this figure or the adjacent main cathode may act as an additional anode, in which case the surface of the electrode would be ruined. In addition, if the guide pulses are too large in amplitude, it is possible for an unprimed guide to strike.

The minimum pulse amplitude which must be applied to the guide electrodes to accomplish the transfer is a function of the pulse duration. It is also dependent on the guide bias ; the greater the positive guide bias, the greater the negative pulse voltage required to overcome this bias and to cause the transfer to occur. If the minimum permissible pulse duration of 65 μsec is employed, the potential difference required between a primed cathode and anode for transfer is about 231 V for Mullard/Philips tubes. The amplitude of the pulse which must be applied to the primed guide cathode is $(231 - V_m)$ where V_m is the maintaining voltage. This pulse amplitude is equal to approximately 35 V. If the pulse length is increased to 100 μsec or more, the required anode to cathode potential is reduced to 214 V so that the negative guide pulses need have an amplitude of only about 18 V. A further increase of pulse length will not reduce the required pulse amplitude any further. These figures apply to the least favourable tubes at the start of life.

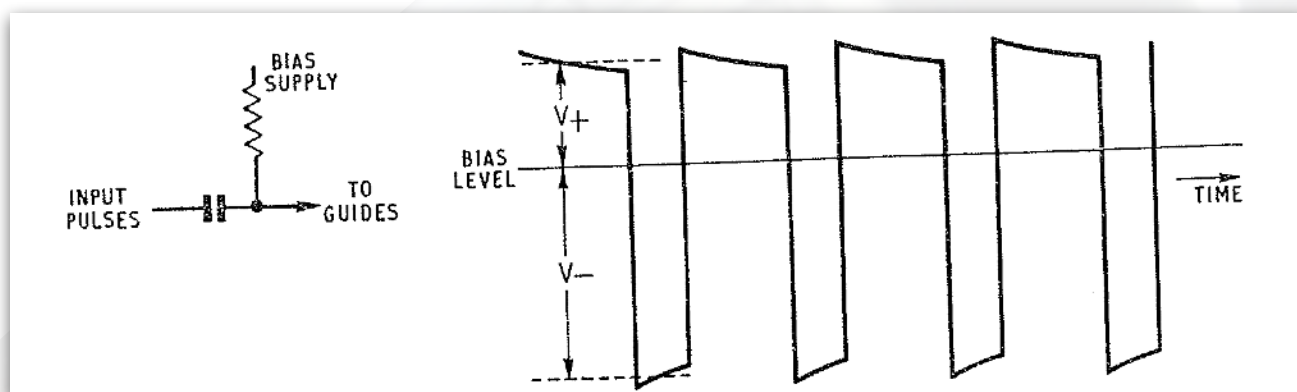


Fig 4.6 Instantaneous guide potential plotted against time

An optimum anode to guide voltage during the pulse of $V_m + 80 \pm 120$ V is recommended for Mullard/Philips 4 kc/s double pulse tubes. This is equivalent to applying a negative pulse to the first guide equal in amplitude to $V_g + 80 \pm 20$ V where V_g is the positive guide bias voltage. It is also recommended that the second guides should receive a pulse of about the same amplitude. A positive guide bias of $+40 \text{ V} \pm 10\%$ is recommended.

If high speeds are not required, 4 kc/s double pulse tubes may be operated with a guide bias of +8 V. Pulses of -15 V in amplitude will then drive the tube. Under these conditions the maximum counting speed is about 700 pulses per second and the output signal amplitude is about 1 V.

The leading edge of the guide pulses should have a rise time exceeding 1 μsec or otherwise a discharge may occur between two of the leads which connect the electrodes to the tube base.

4.2.9 Guide Bias Circuit

When the pulses are fed to the guides through a capacitor, the effective value of the guide bias is

Pin 1 Common cathodes
2 1st Guides
3 Anode
4 2nd Guides
5 Cathode
6
7
8

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			75.5 mm	
B	85 mm			87.5 mm	

Copyright Marsh/Stock 2021-2022

different from the applied bias voltage. When a train of rectangular pulses is passed through a capacitor to the guides (Fig. 4.6), the potential at the guide is such that the area of the waveform above the horizontal line representing the applied bias is equal to the area below this line. If the pulse applied to one set of guides is 60 μsec in duration with inter-pulse spacings of 190 μsec and the total voltage swing ($V+ + V-$) is 80 V, it can easily be shown that $V+$ will be 19.2 V. $V+$ is effectively added to the steady bias applied to the guides. If the repetition rate is halved, the interval between the pulses becomes 440 μsec and $V+$ falls to 10 V. Thus if either the pulse amplitude or the mark to space ratio can vary and the tube is to be operated at fairly high speeds, it is essential to use clamping diodes to ensure that the guide bias is kept constant.

The internal resistance of the bias supply for the guides (R in Fig. 4.7) also requires some consideration. Immediately after a pulse is applied to the guides, the coupling capacitor, C , begins to charge from V_g through the resistor R and also from the current passing through the tube; this charging current also passes through the internal resistance R' of the source of the guide pulses. The potential of the guide rises and if C is small it may rise so much that the discharge moves from the guide whilst the pulse is still being applied to it. R and C should, therefore, be large during the time that the pulse is applied so that C does not charge appreciably from V_g .

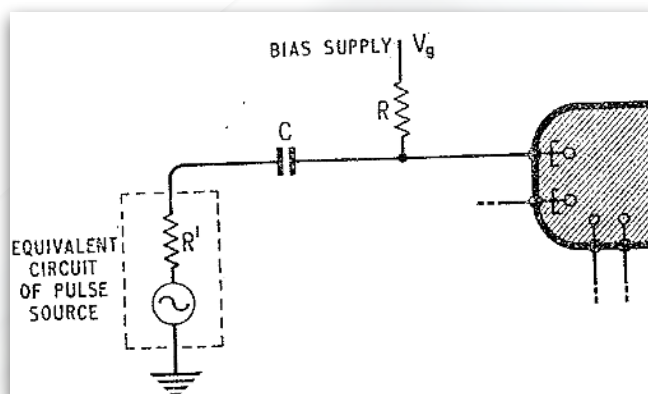


Fig. 4.7 Bias supply impedance

At the end of the guide pulse, however, it is desirable that R and C should be small so that C can be discharged through R and R' in series before the next pulse arrives.

In practice one satisfactory arrangement consists in the use of a fairly large value for C and a diode for R . The diode presents a large impedance during the pulse, but its resistance is very low as C discharges by sending a current through it in the forward direction. The use of a potentiometer to supply the guide bias for a number of double pulse tubes is only permissible when the potential divider resistance values are so low that the guide currents of all stages together cause a bias change of only a few volts. Adequate decoupling should be provided.

4.2.10 Basic Guide Integrator Circuit

Normally two suitable separate input pulses are not available for the operation of a double pulse tube and therefore the pulse required for the second guides must be obtained from that applied to the first guides. In practice the second guide pulse is almost always obtained by passing the first guide pulse through a simple integrating circuit (see Fig. 4.8). The pulse is delayed by the desired amount, but its amplitude may be reduced somewhat by the integrating circuit. The circuit design is fairly critical, since a compromise between the desired second guide pulse amplitude, width

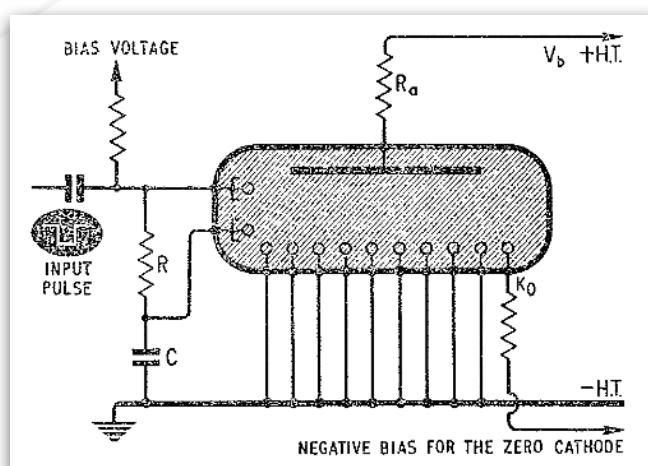


Fig. 4.8 The basic RC integrator circuit for obtaining

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

and delay must be made. Numerous pulse shapes are possible, but satisfactory results will normally be obtained if a voltage at least equal to the minimum recommended transfer voltage is maintained for at least the minimum recommended transfer time with a suitable overlap. The form of the integrated second guide pulse is shown in Fig. 4.5.

When the negative going input pulse of about 120 V in amplitude is applied to the input of the circuit of Fig. 4.8, the first guide potential falls and the capacitor C begins to charge from the negative the capacitor C begins to charge from the negative pulse. The time constant for this charging is determined by the values of R and C; by a suitable choice of these components the potential of the second guides can be made to reach the transfer potential after any desired time.

At the end of the input pulse the first guide potential rises immediately to the guide bias voltage, but the capacitor C takes time to discharge through the resistor R and the bias supply resistors. The second guide potential thus rises exponentially to the guide bias level as shown in Fig. 4.5 and the transfer is then complete.

4.2.11 Anode Capacity

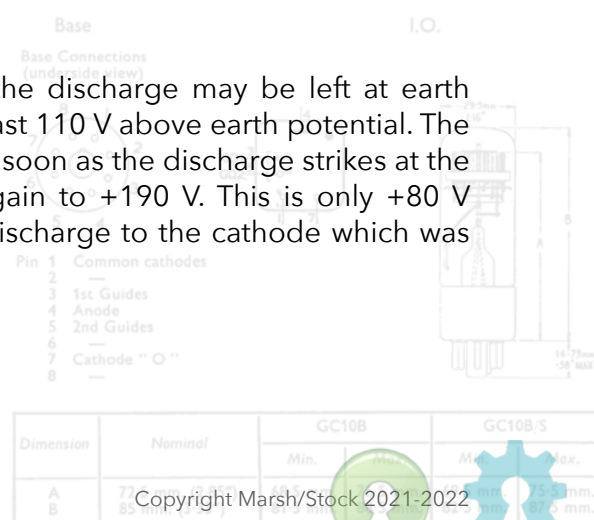
If the tube anode to cathode capacity is excessive, the anode potential may be prevented from rising rapidly as the guide potential rises at the end of the second guide pulse. The anode to second guide potential may then fall below the maintaining voltage of the tube so that the glow is extinguished. The anode voltage will rise and ignition may occur at any of the ten main cathodes. This difficulty is most likely to occur when the trailing edge of the second guide pulse is steep ; for this reason the slope of the trailing edge should not exceed 100 μ sec. Stray anode circuit capacitance should also be minimised by soldering the anode resistor directly to the tag of the tube base. The problems associated with the anode circuit capacity become much more acute when high speed cold cathode tubes are used.

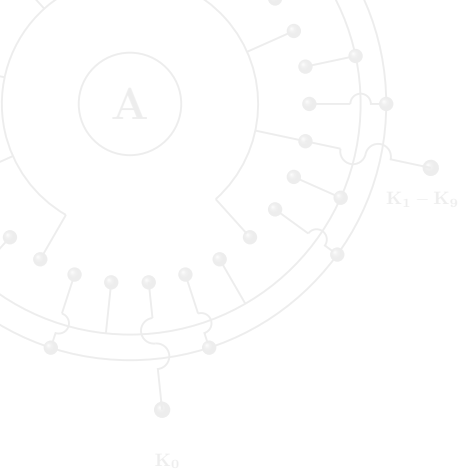
4.2.12 Reset

When a discharge is present at any place in a 4 kc/s double pulse tube, the striking voltage of any anode to cathode gap does not exceed about 300 V. The discharge may, therefore, be reset to zero (or, in the case of a selector tube, to any desired digit) by causing the above potential difference to be present for a short time across the gap to which it is desired transfer the discharge. There are two basic methods by which this may be accomplished.

If the glowing cathode is earthed, the anode potential will be about +190 V with respect to earth. If any other cathode receives a pulse which reduces its potential to at least 110 V below earth potential, this cathode will strike. The anode voltage will fall to +80 V so that the potential across the is tube is equal to the maintaining voltage. The cathode which was initially glowing will therefore be extinguished.

Alternatively the cathode to which it is desired to transfer the discharge may be left at earth potential and all of the other cathodes may be pulsed to at least 110 V above earth potential. The anode potential will commence to rise towards +300 V, put as soon as the discharge strikes at the desired anode-cathode gap, the anode potential will fall again to +190 V. This is only +80 V higher than the potential of all the other cathodes and the discharge to the cathode which was initially glowing will, therefore, be extinguished.





The Harwell Dekatron Computer

An Excerpt by Kevin Murrell - Computer Conservation Society and The National Museum of Computing, UK





The Harwell Dekatron Computer
Kevin Murrell

Computer Conservation Society and The National Museum of Computing, UK

Abstract. The Harwell Dekatron Computer is a very early digital computer designed and built by the British Atomic Energy Research Establishment in 1952. The computer used British Post Office relays for control and sequencing, and Dekatron counting tubes for storage. After several years' service, it was passed to a college where it was used to teach computer programming, before being lost to various storage centres. In 2008 the machine was re-discovered and the decision was made to restore the computer to working order. This paper describes the machine and the choices and decisions made during the restoration process.

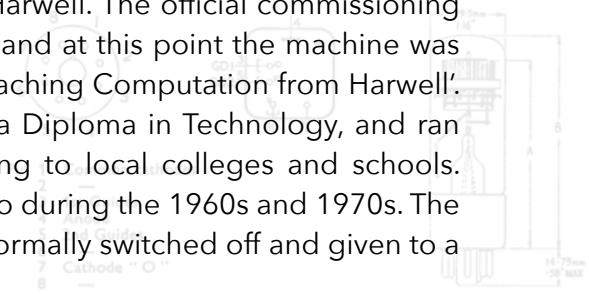
Keywords: Harwell Dekatron Computer, restoration, WITCH, National Museum of Computing, Dekatrons.

Introduction: Restoring the Harwell Dekatron Computer

Design of the computer began in 1949 using techniques that were generally well understood, and used components that were available to the Harwell engineers at the time. There seem to have been remarkably few problems with the design when it came to commissioning the machine and running the first few tests, and the computer was completed and first made to work properly in April 1951 using just two of the store units. This was sufficient to prove the machine was functioning correctly, and to allow simple programs to be written to test the machine more fully. By 1952, the team had added the remaining Dekatron store units and handed the computer over to the theoretical physics division for proper use.

The very conservative design of the computer was quite deliberate as it led to a reliable workhorse that could be left unattended to plough through a long series of calculations. Harwell kept precise records of the work done by the computer, and in a paper published in 1953 it was reported that in its first year with the division, from May 1952 to February 1953, the machine ran on average 92½ hours each week or for 55% of the total available time. Typically any lost time was when the machine had finished its allotted tasks and was waiting for its operators to arrive the next morning!

By 1957, the mathematicians at Harwell were making little use of the Dekatron Computer, but did not wish to see the machine broken up, so an imaginative competition was devised to make further use of the computer and give it a new home in education. On the 25th February 1957, Wolverhampton and Staffordshire College of Technology announced to the world they had won the competition and would soon receive the computer from Harwell. The official commissioning day in Wolverhampton was Wednesday, 4th December 1957, and at this point the machine was re-christened 'WITCH' - the 'Wolverhampton Instrument for Teaching Computation from Harwell'. The college quickly established both a full time degree and a Diploma in Technology, and ran regular courses, many in the evening, to introduce computing to local colleges and schools. Computing moves at a frenetic space, and this was especially so during the 1960s and 1970s. The college made full use of the computer until 1973 when it was formally switched off and given to a local museum.



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.

The computer was displayed in the museum for some time, but was eventually dismantled and put into long-term storage. In 2008, the author, by then the secretary of the Computer Conservation Society and a founder trustee of The National Museum of Computing in the UK, discovered a few parts of the machine and set about tracking down the remains of the computer. Quite quickly we discovered the bulk of the computer, albeit broken down into many pieces.

By now, there was a feeling of genuine excitement that this machine could be restored to working order and presented again for the public to see and appreciate. The work on collecting information about the machine continued in earnest, and again fate intervened when at an unrelated BCS (British Computer Society) meeting, we were given the contact details for Ted Cooke-Yarborough who was living near Abingdon – not far from Harwell. Ted was able to contact Dick Barnes, and very soon a meeting was arranged to tell them of our re-discovery of the machine and our plans for its restoration. The project team travelled to Abingdon in March 2009 to meet the two engineers to show them the results of our investigations and discuss our hopes to restore the machine to full working order. Barnes had kept a scrapbook of newspaper cuttings that proved very useful in understanding the history of the machine.

In April 2009 the project team wrote a formal restoration project plan to define what we hoped to achieve, what we needed to complete the project, and to detail any risks along the way. This plan was presented to the Computer Conservation Society (CCS) in May 2009 and was accepted as a new CCS Working Party. The CCS also kindly offered funds towards the initial costs of the project.

Some of the considerations we discussed were:

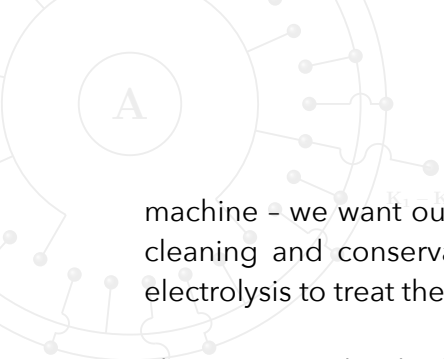
- Might we simply re-assemble the machine back to its original condition?
- Is enough of the original system available to consider its restoration to working order?
- Do we have the necessary tools and documentation?
- Do we have the sufficient spare parts that might be needed?
- How do we decide what might be replaced?
- Is original software available or accessible?
- Do we have the funds and the space to do this?

In September 2009, the machine was moved to its new home - the Large Systems Gallery at the National Museum of Computing. Even before any restoration work commenced, it was already attracting the curiosity of visitors, many of whom had never seen a computer looking anything like this machine with its rows of Dekatrons, valves and trigger tubes, and mysterious grey canisters filled with relays.

Our objective therefore was to restore the machine to a state where it could be reliably and safely operated for demonstration to the public and the many educational and corporate groups who visit the museum.

In common with all our restoration work, the philosophy is to change as little as possible, particularly external appearance. Although it may be tempting to re-spray a rusty old machine, as one might routinely do when restoring a classic car, we feel this alters the character of the

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.



machine - we want our historic exhibits to look their age! In general we confine our activities to cleaning and conservation to prevent further degradation. On the Harwell machine we used electrolysis to treat the areas worst affected by rust.

This computer has had a uniquely long working life, during which it was extensively modified and repaired. We decided early on that we would not attempt to return it to an 'original' condition, and indeed, as work progressed, it became clear that establishing the exact details of the original design would be rather difficult in some areas.

Particularly in the power supplies, it was necessary to replace failed elderly components that are no longer available, some of these being quite prominently visible. In those cases the original components have been left in place, with the much smaller modern replacement components hidden from view. This helps to ensure the machine retains its unique character.

Restoration work fell into three main categories: relays, mechanical, and electronics. The machine came with full circuit diagrams, and an excellent detailed technical description written by Dick Barnes at Harwell. There was also a reasonable quantity of spare components.

Once work had begun in earnest we quickly discovered that the machine deviated from the diagrams in a number of areas, and furthermore that modifications had been made for which documentation hasn't survived. This has made the work more challenging than it might otherwise have been, and efforts are still on-going to fully document the machine in its current state.

Relays

The machine was built using Post Office 3000 type relays, which were at the time a mature and well understood technology. They were designed to operate for decades in the harsh environment of a telephone exchange, and consequently were extremely well made. A number of relays were discovered with bent contacts, and one concern was they may have been deliberately set out of normal adjustment to compensate for some timing problem within the design of the machine. In the event we have discovered very few areas that depend critically on relative relay timings.

The long and tedious job of relay adjustment has been rewarded by relatively trouble free operation of the complex relay logic of the machine, which at the time of writing has been working for over two years. During this period the machine has been regularly demonstrated to museum visitors by means of a simple program that reads from two tape readers alternately using block search and transfer control instructions. We anticipate an on-going need for periodic cleaning and adjustment of the relay sets.

Mechanical

The output equipment consists of a Creed 7B printer, a Creed 75 tele-printer and a Creed 7TR re-perforator. All of these were specially modified at Harwell to operate with this machine. The printer and re-perforator date from the early 1940s, and so pre-date the rest of the machine by quite a few years.

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Copyright Marsh/Stock 2021-2022

The standard Creed models used serial I/O – a more primitive version of the RS-232 serial port which until recently could be found on every modern PC. By contrast this machine uses parallel 5-wire I/O. There are standard 5-bit character codes, but in common with many other early computers this machine used a custom character set. One reason for this was to simplify the logic required to assure correct operation – thus the numbers 0-9 that form the instruction code are all represented by combinations of two holes, and the block marking codes used to delineate program or data sections by combinations of three holes. This meant relatively simple logic could detect when an unexpected code was encountered and halt operation.

Restoration of these machines involved nothing more than a good clean and some adjustments – they really were built to last! Again we are fortunate to have experts in this technology at the museum who were able to carry out the work.

Electronics

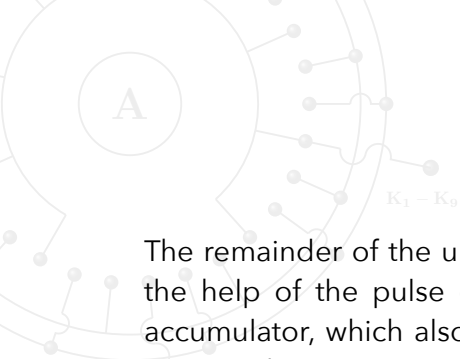
Sequence control is entirely relay based, but arithmetic is all electronic, built using valves, trigger tubes and Dekatrons. There are also two high voltage power supply units – the rectifier unit converts the AC mains into several high DC voltages, and the stabiliser unit generates about a dozen precise DC voltages for use in the various circuits.

The rectifier unit has been extensively modified during the operational life of the machine, and now sports a variety of rectification technologies dating from the 1950s (copper oxide) to the present day (silicon diodes). We had to replace some of the original copper oxide rectifiers with modern components, but others had already been replaced during the machine's time at Wolverhampton. The other major problem in this unit was a leaking oil-filled transformer, eventually solved by copious use of Araldite. All the transformers in the machine were tested to ensure the insulation was still in good condition.

Once the power supplies were operating reliably we turned our attention to the pulse generator, which is one of the most complex parts of the machine. Its main job is to generate various sequences of pulses to step the Dekatrons, but it also has a hand in starting arithmetic operations, deciding when they are complete, and controlling the transfer and carry units which perform the actual addition and subtraction operations. Some of the subtler (and undocumented) aspects of its operation only became clear when it was back on the racks and interacting with the rest of the machine.

The majority of the work was done with the pulse generator on the bench, with a few switches connected up to provide control inputs. We found around half a dozen failed components – mainly diodes and capacitors. The other major issue has proven to be the high-speed trigger tubes which start and stop the Dekatron used in the pulse generator to count out ten pulses. We had a large supply of spare tubes that came with the machine, but the majority turned out not to work. We were lucky to find a supplier who allowed us to test their entire stock of several hundred, but this only yielded a small number of working ones, and some of these no longer work at the time of writing. The most likely explanation is the gas has escaped around where the pins enter the base – this is surprising since the problem of making valves reliably gas tight had been largely solved by the early 1950s. Keeping the machine running long term will eventually require a modification of these circuits to use a more reliable type of trigger tube.

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75			mm.	75.5 mm.
B	85			mm.	87.5 mm.



The remainder of the units in the arithmetic rack were repaired one at a time on the bench with the help of the pulse generator and power supplies. These consist of the two halves of the accumulator, which also contain plug-in transfer and carry modules, and the translator unit that converts between 5-wire code and Dekatron compatible pulse trains. The translator is used both when reading numbers from tape into store and when printing out from store. No major issues were found with any of these units bar a handful of failed components.

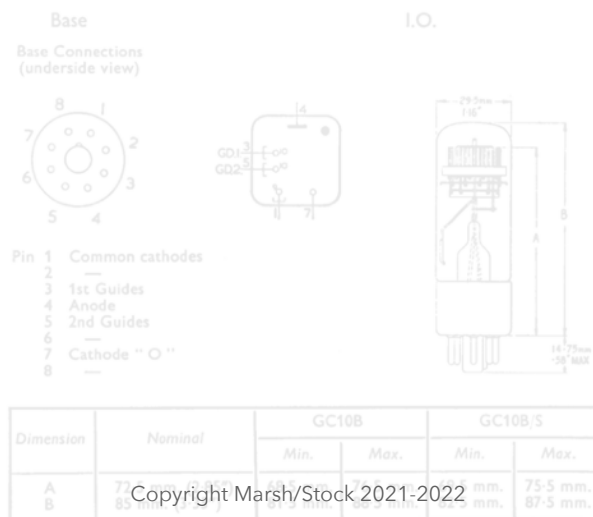
With the arithmetic rack completed we turned our attention to the stores. The only major problem that Dekatrons suffer from is contamination of adjacent electrodes due to the glow being left on one cathode for a long period of time. This results in 'sticky' operation - either the glow won't reliably step past a particular cathode, or jumps straight over it. Both of these problems cause an incorrect answer. Luckily the process can be reversed by simply allowing the Dekatron to 'spin' for a long enough period.

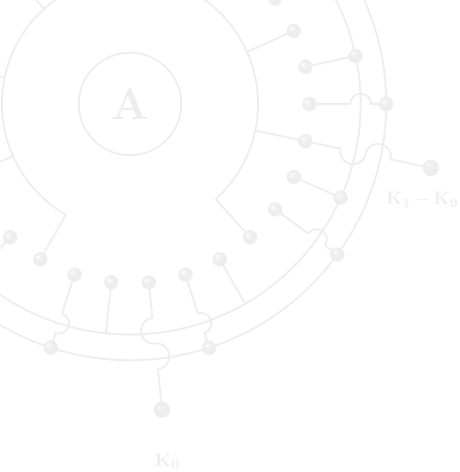
No detailed operational log of the machine survives, but it's likely some of these faults have been present for many years and were probably partly responsible for the unreliability which we know the machine developed in its later life. These circuits must only just have worked when the machine was new, and hence even small component value changes caused a problem.

The goal of the restoration was to have the machine fully working and as reliable as it was when new. At the time of writing this has largely been achieved, with only a few reliability issues left to solve. The machine is once again able to run the original programs written for it in the 1950s.

The museum plans to produce teachers and students information packs about the machine that can be downloaded in advance of their visit. The packs will include detailed instructions on how the machine is programmed, example programs and potential programming challenges. We also hope to include a software emulation of the computer so that students can prepare programs prior to their visit, and then on the day actually run them on the live machine.

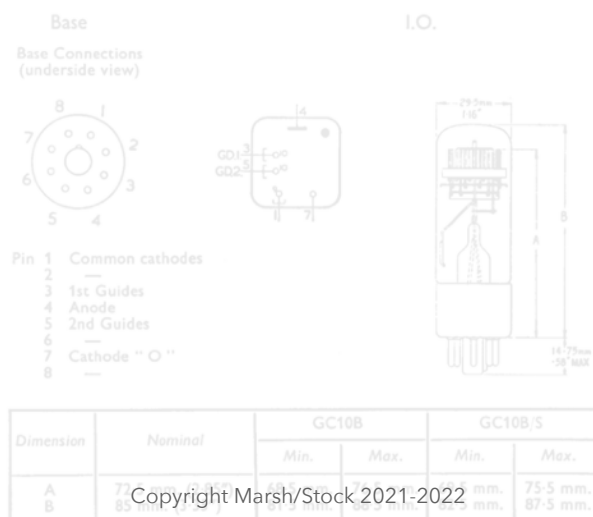
Very little remains of the history of computing in the UK from the late 1940s and early 1950s, and we are very lucky that this machine has led such a charmed life and managed to 'avoid the chop' on several occasions. Its important early role in pioneering nuclear technology, its long second life in education, and its third life as a museum exhibit enthusing young people, can now continue as an active machine enthusing and helping train a new generation of engineers and programmers.





Harwell - Nuggle for the Dekatron

Writing code for your Harwell Clock using firmware version up to 1.3.1



1. Introduction

Nuggle⁸⁵ supplementary documentation for the Harwell Dekatron Clock using clock firmware up to version 1.3.0.

2. Basic Language Elements

Reserved Words

Reserved words that are new or have a specific meaning for the Harwell version of Nuggle are:

Colon, Dekatron, Dekatron6, Dekatron7, DekatronReset, Effect, Flyback, GPIO, Saver, WriteDekatron

5. Program Heading and Program Block

System Defined Number Variables

There are no additional system defined number variables for the Harwell Clock.

Statement Part

Harwell Clock specific events are described in section 10.

7. Functions

Harwell Clock specific function designators are as follows:

Standard System Functions

Colon

Syntax: Colon (*side*)

Returns the intensity value from the colon indicator given by *side*. The returned value will be in the range 0 (off) to 100 (maximum brightness). The argument *side* can be 1 for the left colon and 2 for the right colon.

Dekatron

Syntax: Dekatron (*tube*)

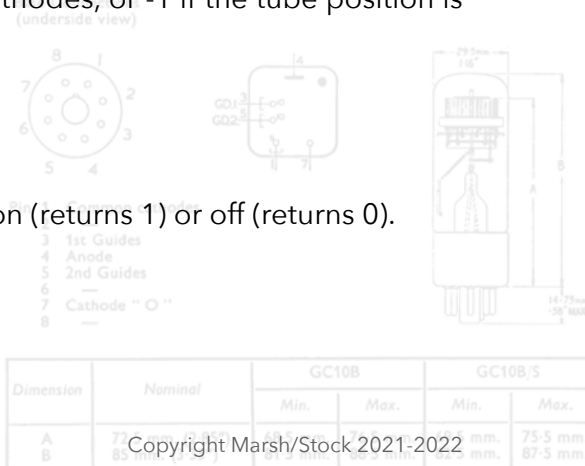
Returns the position of the Dekatron tube. The value will be from 0 to 9 if the tube is glowing and has 10 cathodes and 0 to 11 if it has 12 cathodes, or -1 if the tube position is undefined as the flyback supply is off.

Flyback

Syntax: Flyback ()

Returns the boolean value if the Dekatron HT supply is on (returns 1) or off (returns 0).

⁸⁵ See the Sgitheach Nuggle Manual for details



GPIO

Syntax: **GPIO** (*pin*)

Returns the boolean value of the GPIO *pin* as follows:

Pin	Location	SAM3X8C port and pin
5	Back of PCB	Port A Pin 5
7	Back of PCB	Port A Pin 7
10	Pin 3 Connector X8	Port A Pin 10
11	Pin 2 Connector X8	Port A Pin 11
25	Back of PCB	Port B Pin 25
28	Back of PCB	Port B Pin 28
29	Back of PCB	Port B Pin 29
31	Back of PCB	Port B Pin 31
500	500V flyback on/off	N/A

9. Procedures

Colon

Syntax: **Colon** *side, intensity*

Set the colon indicated by *side* to brightness *intensity*. The argument *side* can be 1 for the left colon and 2 for the right colon. The *intensity* argument takes a value from 0 (colons are dark) to 100 (colons are at maximum brightness).

Dekatron

Syntax: **Dekatron** *tube, value*

Move the glow on the tube to the position *value*. The tubes are number 0 (leftmost) to 5 (rightmost) on the top row and 6 (one, centred) on the bottom row. The Dekatron tubes electrodes are 0 (top) clockwise to 9 if the tube has 10 cathodes and 0 (top) clockwise to 11 if the tube has 12 cathodes.

Dekatron6

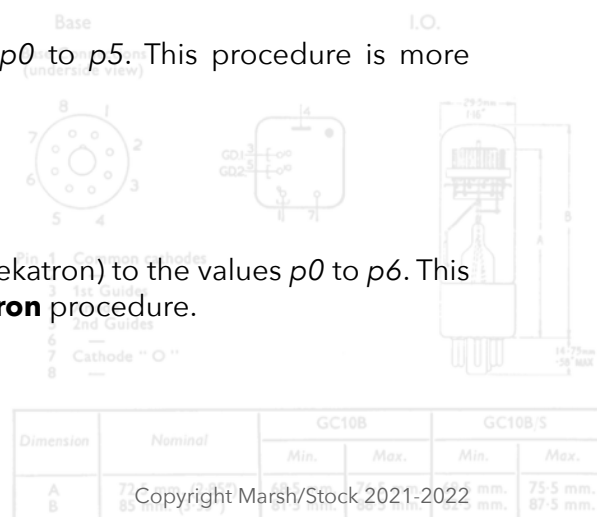
Syntax: **Dekatron6** *p0, p1, p2, p3, p4, p5*

Set dekatrons V1 to V5 (the top row) to the values *p0* to *p5*. This procedure is more efficient than using the **WriteDekatron** procedure.

Dekatron7

Syntax: **Dekatron7** *p0, p1, p2, p3, p4, p5, p6*

Set dekatrons V1 to V5 (the top row) and V6 (bottom Dekatron) to the values *p0* to *p6*. This procedure is more efficient than using the **WriteDekatron** procedure.



Effect

Syntax: **Effect** *type* [, ...]

The **Effect** procedure is a macro command that causes a single, or all the dekatrons, to behave in some way. The first argument is always the effect type and is followed by an argument list that varies according to the effect type.

The effect *type* can be any value from:

Value	Effect Action	Constant name
0	move to a specified cathode (see note 1)	effect_cathode
1	idle (stop any other effect operating) a specified Dekatron and leave the dot where it is	effect_idle
2	idle all dekatrons	effect_idle_all
3	move the dot and go idle	effect_move
4	spin a Dekatron once and go idle	effect_once
5	operate the Dekatron as a pendulum	effect_pendulum
6	show or highlight a position	effect_show
7	continuously spin a Dekatron	effect_spin
8	show or highlight two dot positions	effect_two
9	move a Dekatron to position 0	effect_zero

Notes: **1.** You can specify a glow position by a cathode value or a dot value, note that quite simply $dot = 3 * cathode$ but it is more efficient to use the *effect_cathode* as the multiplication takes place in C rather than Nuggle.

Some of the procedures take a *direction* argument which can be any value from:

Value	Effect Action	Constant name (2)
0	move forwards i.e. clockwise	dir_forward
1	move backwards i.e. anticlockwise	dir_backward
2	take the shortest route	dir_shortest
3	take the longest route	dir_longest

Notes: **2.** These constant names are also used by the bling GradientHSV procedure.

Procedures taking a *tube* number must use the value 0 for Dekatron V1 up to 6 for Dekatron V7. Constants V1, V2 ... V7 are defined for use. Procedures taking a *cathode* number must use a value from 0 to 9 for tubes V1 to V6 and a value from 0 to 11 for Dekatron V7. Procedures taking a *dot* number must use a value from 0 to 29 for tubes V1 to V6 and a value from 0 to 35 for Dekatron V7. A *pace* number is the rate at which the dot moves per mS e.g. a value of 1000 would move the dot between adjacent positions once per second.

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75	75	75	75	75
B	85	85	85	85	85

Syntax: Effect *effect_cathode, tube, cathode, direction, pace*

Move the glow position to the required cathode on tube in the given direction and pace.

Example:

Effect *effect_cathode, V7, 6, dir_shortest, 100*

Will move the glow on the bottom Dekatron, V7, to the cathode, taking the shortest route at the pace of one dot every 100mS

Syntax: Effect *effect_idle, tube*

Stop any other effect that Dekatron tube is operating and leave the dot where it is.

Syntax: Effect *effect_idle_all*

Stop effects on all dekatrons.

Syntax: Effect *effect_move, tube, dot, direction, pace*

Move the glow position to the required dot on tube in the given direction and pace.

Syntax: Effect *effect_once, tube, direction, pace*

Spin the Dekatron tube once using the *direction*⁸⁶ and pace given.

Example:

For *toob := V1 To V6*

Effect *effect_once, toob, dir_forward, 10*

ForEnd

Spin dekatrons V1 to V6 once rapidly in a clockwise direction

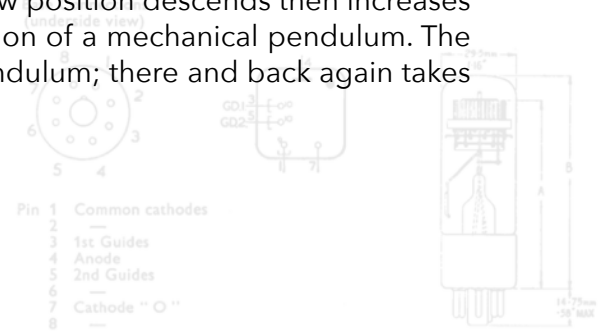
Syntax: Effect *effect_pendulum, tube, dot, style, period*

Make the Dekatron act in a the fashion of a pendulum. Dekatron *tube* is used. The starting position is given by *dot*. The glow position will "fall" towards the bottom dot the continue "up" to the corresponding dot position when it will reverse direction and fall again. This cycle is repeated. The rise and fall speed is set by two arguments. The *style can be linear*, where an equal amount of time is spent with each dot glowing, and, *gravity* where the time spent with each dot glowing decreases as the glow position descends then increases again as the dot rises. The *gravity style* gives a simulation of a mechanical pendulum. The *period* value is the time in mS for one swing of the pendulum; there and back again takes $2 * period$ mS.

Example:

Effect *effect_pendulum, V7, 12, linear, 500*

⁸⁶ Because the starting and ending glow positions are the same *dir_shortest* and *dir_longest* have no meaning and *dir_forward* will be used



Pin 1 Common cathodes
2 —
3 1st Guides
4 Anode
5 2nd Guides
6 —
7 Cathode "O"
8 —

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm				
B	85 mm				

Copyright Marsh/Stock 2021-2022

mm. mm. 75.5 mm. 87.5 mm.

Dekatron V7 swings back and forth each second in a linear pendulum fashion.

Syntax: Effect *effect_show, tube, dot, swing, pace*

Show or highlight, a given *dot* position by oscillating the glow position either side or it. The amplitude of the oscillation is given by *swing* and the rate of motion by *pace*.

Example:

Effect *effect_show, V3, minutes / 2, 3, 10*

Effect *effect_show, V4, minutes / 2, 3, 10*

Highlight the dot position on V3 and V4 based on the *minutes* system variable value.

Syntax: Effect *effect_spin, tube, dot, direction, pace*

Spin a Dekatron *tube*, continuously. The spinner will start from position *dot*, move in *direction*⁸⁷ at the given pace.

Example:

Effect *effect_spin, V3, 0, dir_forward, 100*

Effect *effect_spin, V4, 0, dir_backward, 100*

This will spin V3 and V4 in opposite directions. Both dekatrons will start at the top.

Syntax: Effect *effect_two, tube, cathode1, cathode2, pause*

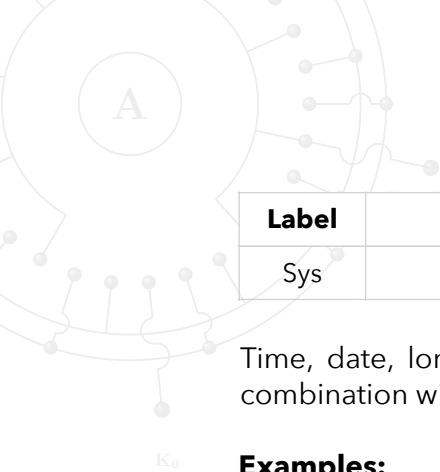
Highlight two cathodes on Dekatron tube by rapidly moving between them and pausing on each. The two positions are given by *cathode1* and *cathode2*. The pause time is in mS.

Dekatron V7 at the bottom of the clock has engraved labels so the Dekatron can be used to indicate what is displayed on the top six dekatrons. Each of the labels has a corresponding constant that can be used in this effect.

Label	Meaning	Constant	Cathode
Idle	the V7 indicator function is not used	dot_idle	0
Time	a time is displayed	dot_time	1
Date	date	dot_date	2
Long	longitude	dot_longitude	3
Lat	latitude	dot_latitude	4
Elev	elevation	dot_elevation	5
Face	face number is shown or being changed	dot_face	6
Pair	pair value is shown or being changed	dot_pair	7
UTC	date or time is UTC	dot_utc	8
Sun	time, longitude, latitude relate to the sun	dot_sun	9

⁸⁷ Because this is a spinner, only *dir_forward* and *dir_backward* have meaning

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75	75	75	75
B	85 mm	85	85	85	85



Label	Meaning	Constant	Cathode
Sys	system data	dot_system	10

Time, date, longitude, latitude, elevation, face and pair can be used in isolation or in combination with UTC, sun and system to give more meanings:

Examples:

- Time indicated alone means the local time is being shown
- Time in conjunction with UTC means that UTC time is being shown
- Time in conjunction with sun shows that a sun related time (such as sun rise or set time)
- Time in conjunction with system shows that the hardware version is being displayed

Obviously the last bullet example is not intuitive and you need to know the system+ meanings. The system dot can be joined with:

Second dot indicated	Meaning
Time	hardware version
Date	firmware version
Long	core temperature
Lat	LDR data
Elev	GPS data
Face	WiFi data

The effect_two is used primarily to display on Dekatron V7 what the other six are being used for.

Example:

WriteDekatron "%02d%02d%02d", hour, minute, second

Effect effect_cathode, V7, dot_time, dir_shortest, 1

will show the time on dekatrons V1 to V6 and indicate "time" on Dekatron V7.

Example:

WriteDekatron "%02d%02d%02d", utc_hour, utc_minute, utc_second

Effect effect_two, V7, dot_utc, dot_time, 1000

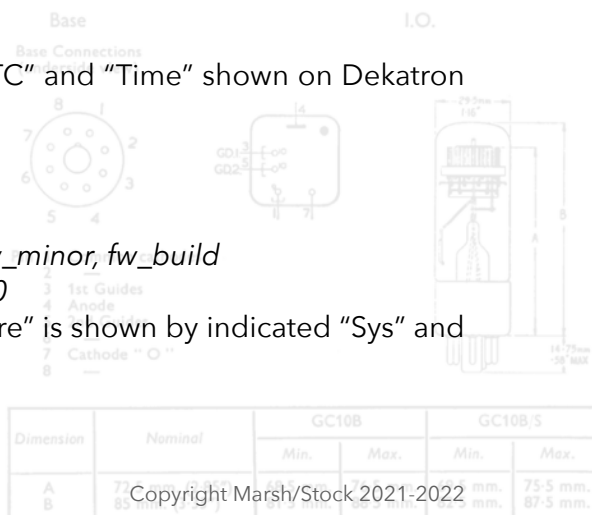
The UTC time is shown on dekatrons V1 to V6 and, "UTC" and "Time" shown on Dekatron V7.

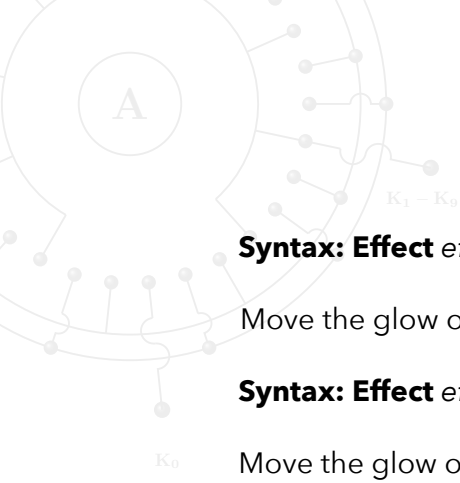
Example:

WriteDekatron "%02d%02d%02d", fw_major, fw_minor, fw_build

Effect effect_two, V7, dot_system, dot_date, 1000

The firmware version is shown on V1 to V6 and "firmware" is shown by indicated "Sys" and "Date" on Dekatron V7.





Syntax: Effect *effect_zero, tube, dir, pace*

Move the glow on Dekatron *tube* to cathode 0 in the given direction *dir* and *pace*.

Syntax: Effect *effect_zero_all*

Move the glow on all the dekatrons to cathode 0.

Flyback

Syntax: Flyback *state*

Turns the Dekatron HT supply to *state*. Where *state* is true (1) or off (0).

GPIO

Syntax: GPIO *pin, value*

This sets the state of a GPIO *pin* to logical 1 if *value* is true and logical 0 if *value* is false. The available values of *pin* and the GPIO port and pin controlled are as follows:

Pin	Location	SAM3X8C port and pin
5	Back of PCB	Port A Pin 5
7	Back of PCB	Port A Pin 7
10	Pin 3 Connector X8	Port A Pin 10
11	Pin 2 Connector X8	Port A Pin 11
25	Back of PCB	Port B Pin 25
28	Back of PCB	Port B Pin 28
29	Back of PCB	Port B Pin 29
31	Back of PCB	Port B Pin 31
500	500V flyback on/off	N/A

Saver

Syntax: Saver *value*

Specify the “screen saver” display to be used with the current clock face overriding the default saver. The argument value is the index of a Nuggle program defined in nuggle.ini.

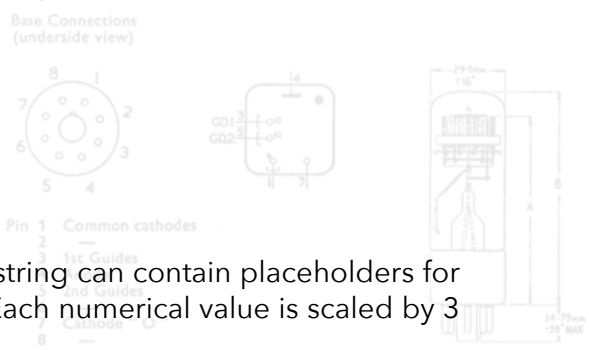
Example:

Saver 402

WriteDekatron

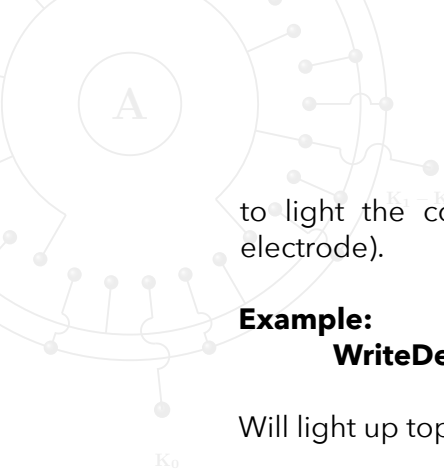
Syntax: WriteDekatron *Str, [, ...]*

Writes the string *Str* to the top six Dekatron tubes. The string can contain placeholders for the optional comma separated arguments that follow. Each numerical value is scaled by 3



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72 mm	68 mm	75 mm	75 mm	87 mm
B	85 mm	82 mm	90 mm	87 mm	95 mm





to light the correct cathode (unlike the Dekatron procedure that lights a specified electrode).

Example:

```
WriteDekatron "012317"
```

Will light up top row of six dekattrons using this test pattern.

Obviously the Dekatron tubes are limited to displaying the numbers '0' to '9' so any other character in the string *Str* will be ignored by the corresponding tube. It is recommended to use the space character. Only the first six characters of the string will be used and any other characters that follow are ignored. If the string is shorter than 6 characters then the remaining tubes are not changed.

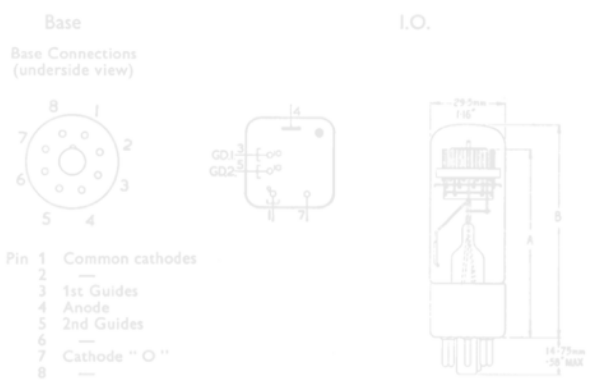
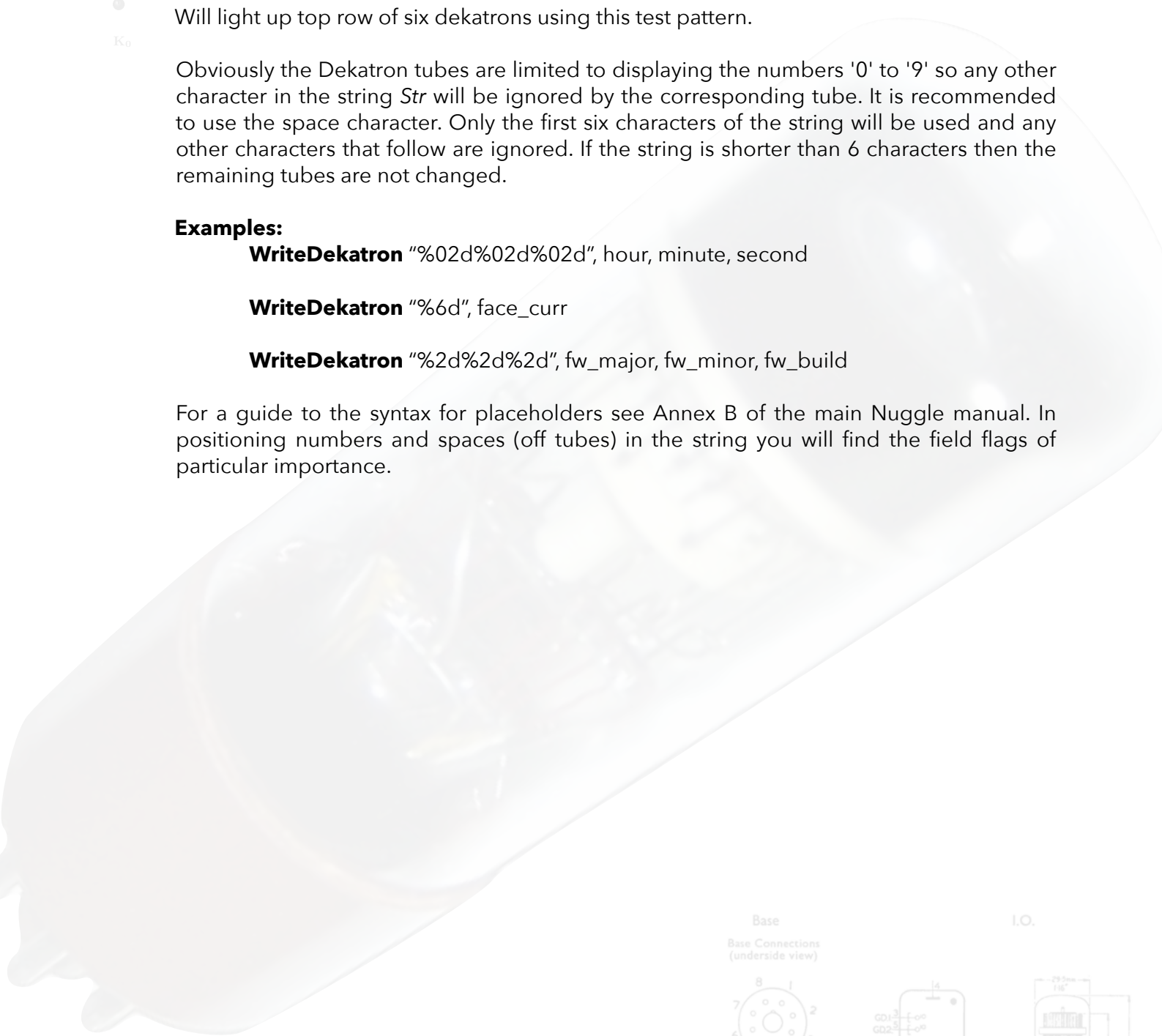
Examples:

```
WriteDekatron "%02d%02d%02d", hour, minute, second
```

```
WriteDekatron "%6d", face_curr
```

```
WriteDekatron "%2d%2d%2d", fw_major, fw_minor, fw_build
```

For a guide to the syntax for placeholders see Annex B of the main Nuggle manual. In positioning numbers and spaces (off tubes) in the string you will find the field flags of particular importance.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	73 mm				
B	85 mm				

10. Events

Event Happenings

There are no additional event happenings.

Event Default Actions

The Harwell Dekatron clock has the following event default actions:

Event Label:	Default Action:
event_encoder:	If the clock is asleep (any stage) it wakes up. Clock face or sound volume changed. See below.
event_ir:	If the clock is asleep (any stage) it wakes up. See below for other default functions.
event_pir:	If the clock is asleep (any stage) it wakes up.
event_sleep:	The event occurs as the clock wakes up from being asleep and the clock goes from being awake progressively to being asleep. See below.
event_enter:	None
event_leave:	When the Nuggle program is stopping.
event_tick:	None

Default Encoder Events

If the encoder is moved clockwise when displaying a clock face, then the next clock face is displayed, if anti-clockwise then the previous face is displayed.

If the encoder button is pressed and it is moved clockwise then the chime sound volume increases, if moved anti-clockwise, then the sound volume decreases.

In summary:

arg_encoder value	Happening	Default Actions
U'	Turned clockwise	Next clock face is shown
+'	Press and turn clockwise	Chime sound volume is increased show current setting on the Dekatron tubes
D'	Turn anticlockwise	Previous clock face is shown
-'	Press and turn anticlockwise	Chime sound volume is decreased show current setting on the Dekatron tubes

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Default IR Handset Events

The IR Handset has default actions for all of the buttons.

You can change the volume of the chimes and the beeps, set the pairing number, select a clock face directly or by scrolling through available faces and finally enquire what face number is currently displayed.

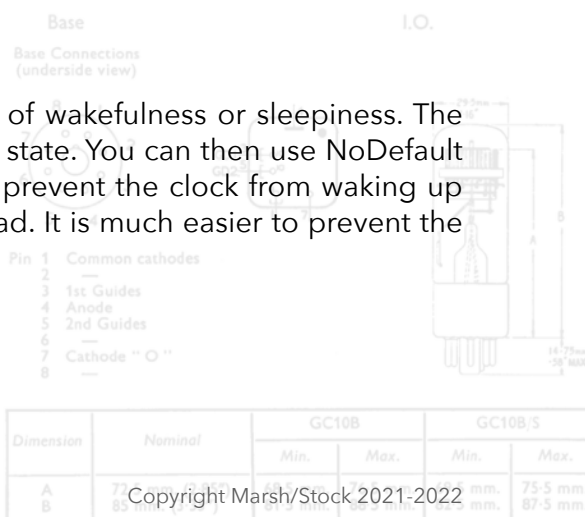
arg_ir value	Happening - button pressed	Default Actions
'U'	CH+	Next clock face is shown
'D'	CH-	Previous clock face is shown
'C'	CH	Display current face on Dekatron tubes
'+'	+	Chime sound volume is increased show current setting on the Dekatron tubes
'.'	-	Chime sound volume is decrease show current setting on the Dekatron tubes
'>'	<<	Beep sound volume is increased show current setting on the Dekatron tubes
'<'	>>	Beep sound volume is decrease show current setting on the Dekatron tubes
'P'	>	Start pair number entry
'0' ... '9'	0 ... 9	Start face number entry
'='	EQ	Pairing number or face number accepted
'H'	100+	In pairing or face entry mode delete last number
'T'	200+	Escape from pairing or face entry mode and discard the entry

Pairing number entry is started with the '>||' key and allows the entry of a number, if it matches the clock pair number then the other keys will operate.

Face number entry allows you to immediately go to a face, if you know its number, without using the CH- and CH+ keys repeatedly.

Default Sleep Events

Invoked when the clock wishes to either go to a higher state of wakefulness or sleepiness. The system variable arg_sleep contains a code to tell you the next state. You can then use NoDefault or Default to allow the state change or not. Obviously, if you prevent the clock from waking up then you will need to use a range of Nuggle commands instead. It is much easier to prevent the clock going to sleep again.



Wake up from being asleep

During the wake up cycle - fully asleep to fully awake - the following states are passed through and actions are taken. The timings of some stages are adjustable in the [sleep] section of system.ini on the SD card.

arg_sleep value	Constant	Actions	[Sleep] key
14	awake_flybackonreq	Request to turn on flyback PSU	
15	awake_flybackon	Flyback PSU turned on	flyback_on = 2 seconds delay
17	awake_complete	Flyback is turned on	
18	awake_release	Request to go to fully awake status	
3	awake_fully	The clock is fully awake	

Going to sleep

During the sleep cycle - going from fully awake to fully asleep - the following states are passed through and actions are taken. The timings of some stages are adjustable in the [sleep] section of system.ini on the SD card.

arg_sleep value	Constant	Actions	[Sleep] key
4	sleep_savercall	Request to invoke the screen saver	
5	sleep_saver	Screen saver displayed	screen_saver = 10 ; minutes delay
6	sleep_flybackoffreq	Request to turn off the flyback PSU	
7	sleep_flybackoff	Flyback PSU turned off	flyback_off = 10 ; minutes delay
9	sleep_release	The clock is about to be fully asleep	
2	sleep_fully	The clock is fully asleep (idle)	

If a wake from sleep trigger is received during these stages the going to sleep process is reversed and the value of *arg_sleep* will change to the relevant waking up state.

If you want to see these processes in action then use the sleep debug system at level 3 (set in system.ini) or write a Nuggle program to do so.

Sleep states, 0, 1, 10, 13 and 16 are used internally and are not sent to the event handler. Sleep states 8, 11 and 12 are not used and are never called.

If the **Default** action takes place at step 5 then the screen saver Nuggle program is run and your program loses control. Your program is then not run again until step 3 has been reached in the wake up cycle. The upshot of this is that your Nuggle program will only receive event steps 4 and 5.



Default Start Events

The default start events are to:

- Set all the Dekatrons to cathode '0'
- Clear all the LEDs to off i.e. black.
- Stop any chime that is playing
- Stop any beep that is sounding
- Set both colons to off, i.e. set the brightness PWM signal to 0 for both colons.

Using **NoDefault** will allow one Nuggle program to take over the display from another without any jitter.

Default Tick Events

The RTC ticks each second and generates an event. The value of the system variable *arg_tick* is set as follows:

arg_tick value	Constant	Highest tick period that has occurred
0	tick_second	1 second
1	tick_minute	1 minute
2	tick_hour	1 hour
3	tick_day	1 day
4	tick_month	1 month
5	tick_year	1 year

For example, a 1 day tick only occurs at 00:00:00 local time. And another example, the year tick only occurs at 00:00:00 on 1st January each year. Time is always local time and takes into account DST and the time zone. The clock UTC tick is not used to generate events.

Miscellaneous Procedures

Clear

Syntax: Clear

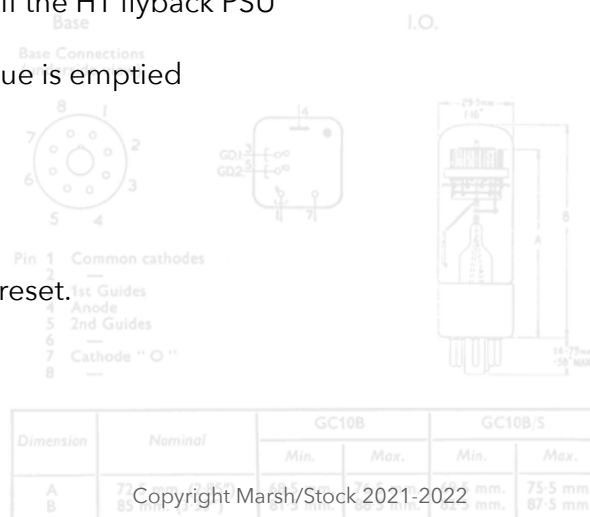
The following happens:

1. Dekatron and colon tubes are blanked by turning off the HT flyback PSU
2. All bling LED are set to *black*, i.e. off
3. Any beep in progress is stopped and the beep queue is emptied
4. Any chime in progress is stopped

DekatronReset

Syntax: DekatronReset

Causes the Dekatron driver slave microcontroller to be reset.





Annex A System Variables

Dekatron Information

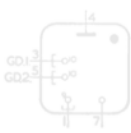
Variable Name	Function	Notes
K₀ type12	Is V7 a 12 cathode Dekatron?	boolean



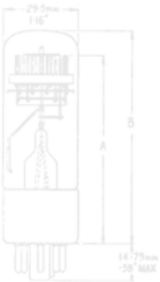
Base
Base Connections
(underside view)



- Pin 1 Common cathodes
2 —
3 1st Guides
4 Anode
5 2nd Guides
6 —
7 Cathode "O"
8 —

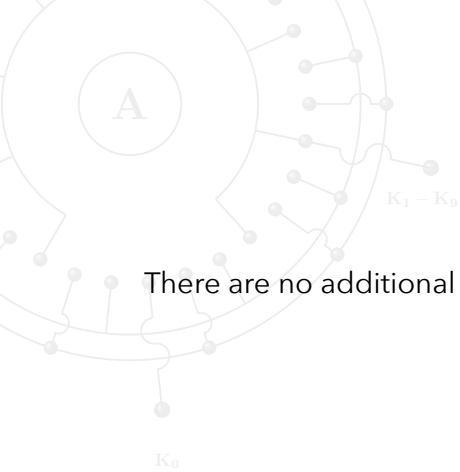


I.O.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72 mm				
B	85 mm				

Copyright Marsh/Stock 2021-2022



Annex C Fatal Error Messages

There are no additional fatal error messages.



Base
Base Connections
(underside view)



- Pin 1 Common cathodes
- 2 —
- 3 1st Guides
- 4 Anode
- 5 2nd Guides
- 6 —
- 7 Cathode "O"
- 8 —

I.O.



Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	72 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Copyright Marsh/Stock 2021-2022

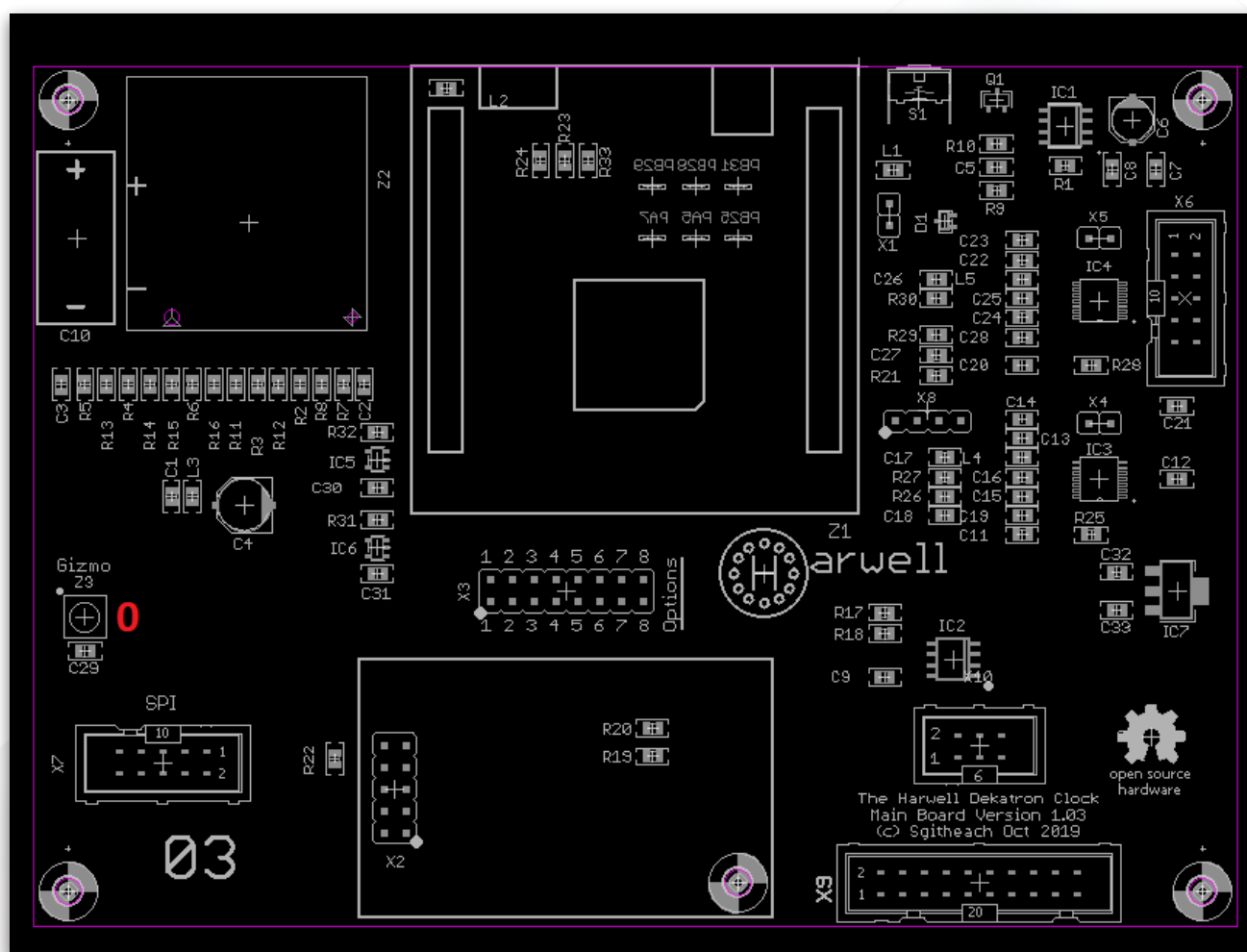
Annex D Bling LED Numbering

Example of use:

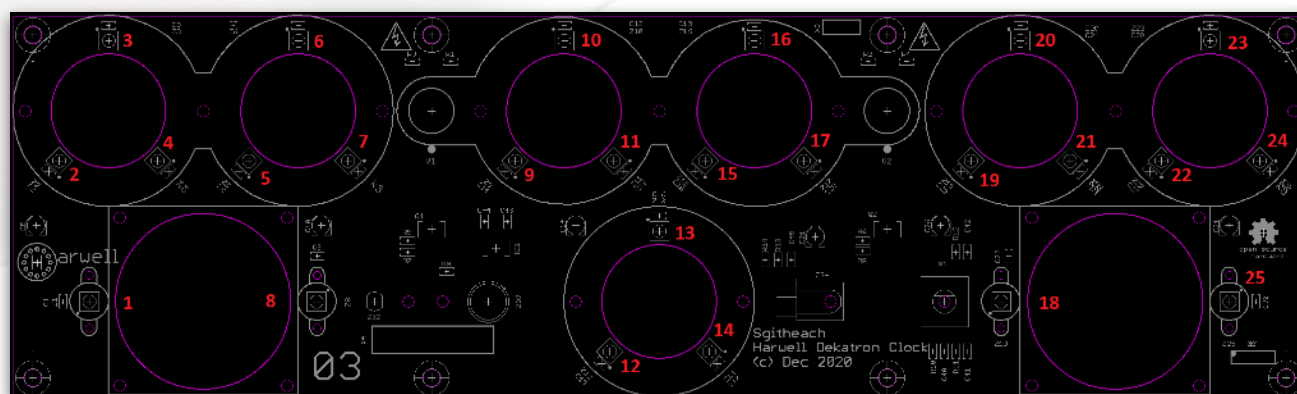
Colour red, 14

Colour red, 23

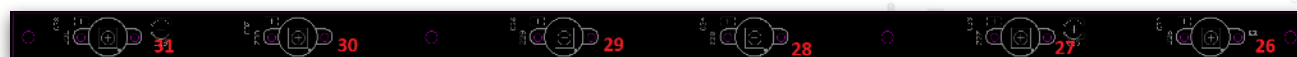
Position numbers of the bling LED modules as viewed from the silkscreen side of the PCB.



Only one bling LED on the Main Processor board - '0'

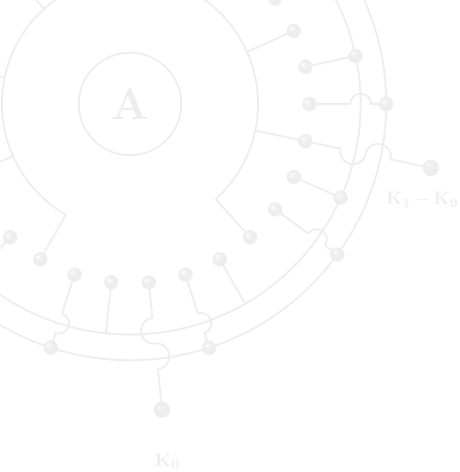


Bling LED numbering on the light guide board - 1 to 25



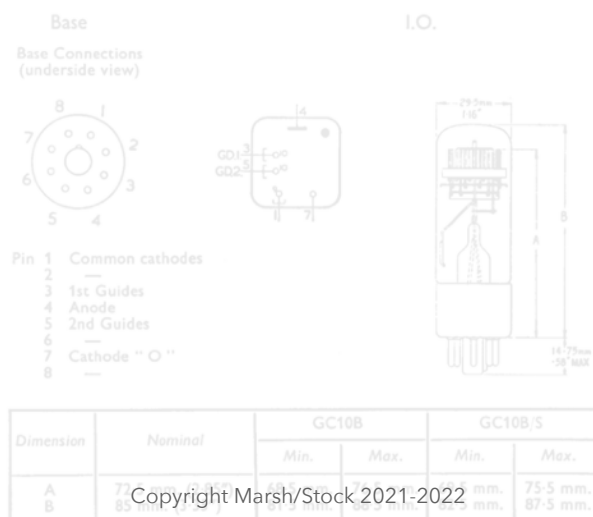
Bling LEDs 26 to 31 on the downward illumination board

		Nominal	Min.	Max.	Min.	Max.
A	B	75			mm.	75.5 mm.
		85			mm.	87.5 mm.



Harwell - Nuggle for the Dekatron

Writing code for your Harwell Clock using clock firmware V1.4.0 and above



1. Introduction

Nuggle⁸⁸ supplementary documentation for the Harwell Dekatron Clock using clock firmware from version 1.4.0

2. Basic Language Elements

Reserved Words

Reserved words that are new or have a specific meaning for the Harwell version of Nuggle are:

Colon, Dekatron, Flyback, GPIO, Saver

5. Program Heading and Program Block

System Defined Number Variables

There are no additional system defined number variables for the Harwell Clock.

Statement Part

Harwell Clock specific events are described in section 10.

7. Functions

Harwell Clock specific function designators are as follows:

Standard System Functions

Colon

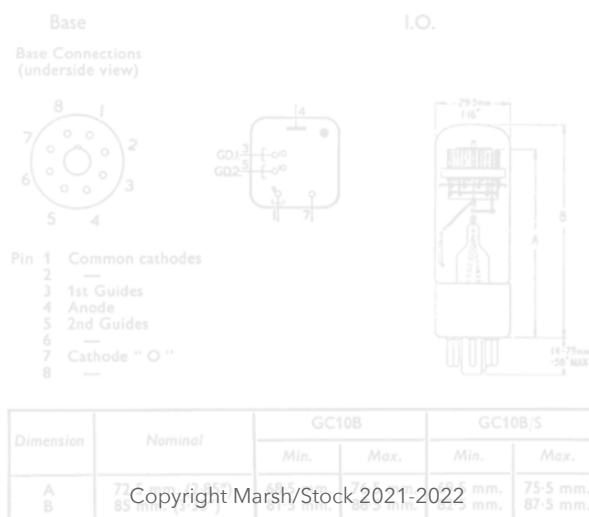
Syntax: Colon (*side*)

Returns the intensity value from the colon indicator given by *side*. The returned value will be in the range 0 (off) to 100 (maximum brightness). The argument *side* can be 1 for the left colon and 2 for the right colon.

Flyback

Syntax: Flyback ()

Returns the boolean value if the Dekatron HT supply is on (returns 1) or off (returns 0).



⁸⁸ See the Sgitheach Nuggle Manual for details

GPIO

Syntax: **GPIO** (*pin*)

Returns the boolean value of the GPIO *pin* as follows:

Pin	Location	SAM3X8C port and pin
5	Back of PCB	Port A Pin 5
7	Back of PCB	Port A Pin 7
10	Pin 3 Connector X8	Port A Pin 10
11	Pin 2 Connector X8	Port A Pin 11
25	Back of PCB	Port B Pin 25
28	Back of PCB	Port B Pin 28
29	Back of PCB	Port B Pin 29
31	Back of PCB	Port B Pin 31
500	500V flyback on/off	N/A

9. Procedures

Colon

Syntax: **Colon** *side, intensity*

Set the colon indicated by *side* to brightness *intensity*. The argument *side* can be 1 for the left colon and 2 for the right colon. The *intensity* argument takes a value from 0 (colons are dark) to 100 (colons are at maximum brightness).

Dekatron

Syntax: **Dekatron** *dekaAction [, arguments]*

All Nuggle **Dekatron** procedures are of the form:

Dekatron *dekaAction, arg0, arg1, arg2...*

Where **Dekatron** is a reserved word (command) and is the lead-in to all bling procedures. *dekaAction* is a constant value for the action you want the dekatron displays to perform such as change a dot position, read a dot position and so on; *arg0, arg1, arg2...* are the other arguments needed by the action execution. Because the *dekaAction* constants are all predeclared any error will be flagged if you try to reuse the identifier for another purpose.

Procedures taking a *tube* number must use the value 0 for Dekatron V1 up to 6 for Dekatron V7. Constants V1, V2 ... V7 are defined for use. Procedures taking a *cathode* number must use a value from 0 to 9 for tubes V1 to V6 and a value from 0 to 11 for Dekatron V7. Procedures taking a dot number must use a value from 0 to 29 for tubes V1 to V6 and a value from 0 to 35 for Dekatron V7.

Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

A pace number is the rate at which the dot moves per mS e.g. a value of 1000 would move the dot between adjacent positions once per second. If the pace is omitted then a default value of 10mS is used.

Procedures taking a *direction* can use these constants:

Value	Action	Constant name
0	move forwards i.e. clockwise	dir_forward
1	move backwards i.e. anticlockwise	dir_backward
2	take the shortest route	dir_shortest
3	take the longest route	dir_longest

The constant names are also used by the **Bling** *blGradientHSV* procedure. If *direction* is omitted then *dir_shortest* is used by default.

In the following syntax action list, arguments within square brackets [] can be omitted and default values will be used.

Dekatron *dekaCathode*

Syntax: **Dekatron** *dekaCathode*, *tube*, *position* [, *direction*, *pace*]

Move the dot position to the required cathode on the tube to position n the given direction and pace.

Example:

Dekatron *dekaCathode*, V7, 6 , , 1000

Move the dot on dekatron V7 to cathode 6 in the default direction (shortest) at the rate of 1 movement per second i.e. slowly!

Dekatron *dekaGet*

Syntax: **Dekatron** *dekaGet*, *numvar*, *tube*

Return the dot position of tube in the number variable *numvar*.

Example:

#Number *TubePos*

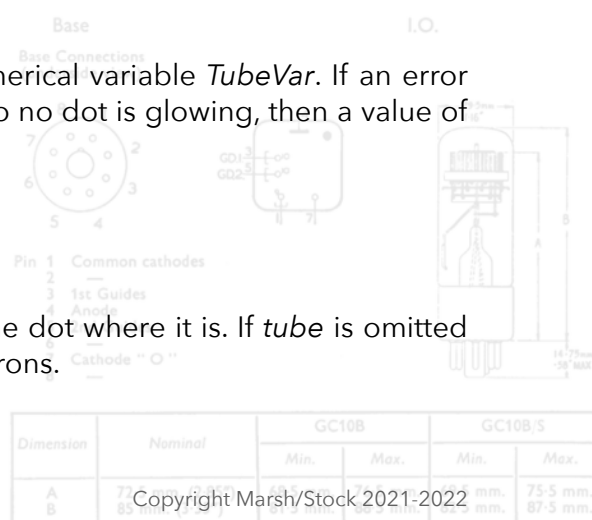
Dekatron *dekaGet*, *TubePos*, V1

This will return the position of dekatron V1 in the numerical variable *TubeVar*. If an error occurs, for example if the flyback power supply is off so no dot is glowing, then a value of 0 is returned.

Dekatron *dekaldle*

Syntax: **Dekatron** *dekaldle* [, *tube*]

Stop any action that the *tube* is operating and leave the dot where it is. If *tube* is omitted then the *dekaldle* action will be applied to all the dekatrons.



Dekatron *dekaOnce*

Syntax: Dekatron *dekaOnce*, tube [, direction, pace]

Spin the dekatron tube once in the given *direction* and *pace*. Because the starting dot and ending dot positions are the same, shortest and longest directions have no meaning and forward direction will be used. Forward is also the default direction.

Example:

#Number Toob

For Toob := V1 to V6

Dekatron *dekaOnce*, toob

ForEnd

Spin dekatrons V1 to V6 once in a clockwise direction.

Dekatron *dekaPendulum*

Syntax: Dekatron *dekaPendulum*, tube [, dot, style, period]

Make the dekatron act in the fashion of a pendulum. Dekatron *tube* is used. The starting position is given by *dot* (default 12). The glow position will "fall" towards the bottom dot the continue "up" to the corresponding dot position when it will reverse direction and fall again. This cycle is repeated. The rise and fall speed is set by two arguments. The style can be *linear*, where an equal amount of time is spent with each dot glowing, and, *gravity* where the time spent with each dot glowing decreases as the glow position descends then increases again as the dot rises (default *linear*). The gravity style gives a simulation of a mechanical pendulum. The period value is the time in mS for one swing of the pendulum; there and back again takes 2 * period mS (default 500mS giving one complete swing per second).

Example:

Dekatron *dekaPendulum*, V7, , , 5000

Dekatron V7 swings as a linear pendulum starting from dot 12 and taking 10S for a complete swing there and back.

Dekatron *dekaReset*

Syntax: Dekatron *dekaReset*

Causes the Dekatron driver slave microcontroller to be reset.

Dekatron *dekaSet*

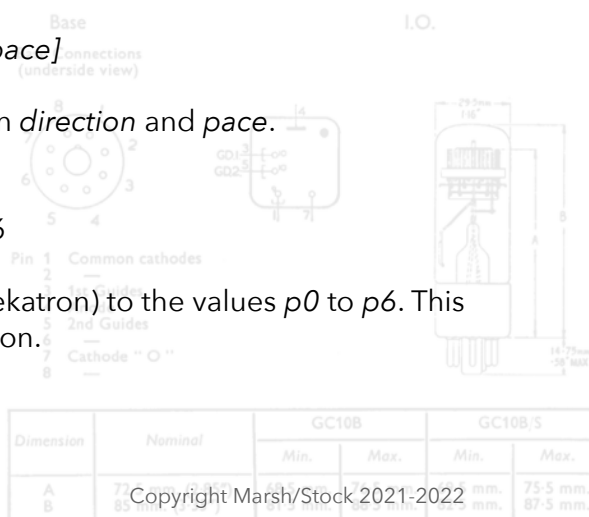
Syntax: Dekatron *dekaSet*, tube, position [, direction, pace]

Move the glow on dekatron tube to position at the given *direction* and *pace*.

Dekatron *dekaSeven*

Syntax: Dekatron *dekaSeven*, p0, p1, p2, p3, p4, p5, p6

Set dekatrons V1 to V6 (the top row) and V7 (bottom Dekatron) to the values *p0* to *p6*. This procedure is more efficient than using the dekaText action.



Dekatron dekaShow

Syntax: Dekatron dekaShow, tube, dot [, swing, pace]

Show or highlight, a given dot position by oscillating the glow position either side of it. The amplitude of the oscillation is given by swing (default value 3) and the rate of motion by pace (default value 10 mS).

Example:

Dekatron dekaShow, V3, minutes / 2

Dekatron dekaShow, V4, minutes / 2

Highlight the dot position on V3 and V4 based on the minutes system variable value.

Dekatron dekaSix

Syntax: Dekatron dekaSeven, p0, p1, p2, p3, p4, p5

Set dekatrons V1 to V6 (the top row) to the values p0 to p5. This procedure is more efficient than using the dekaText action.

Dekatron dekaSpin

Syntax: Dekatron dekaSpin, tube [, dot, direction, pace]

Continuously spin a Dekatron tube. The spinner will start from position dot (default value 0, i.e. at the 12 'o'clock position), move in direction (because this is a spinner, only dir_forward and dir_backward have meaning) at the given pace (default 33 mS).

Example:

Dekatron dekaSpin, V3, , , 100

Dekatron dekaSpin, V4, , dir_backward, 100

This will spin V3 and V4 in opposite directions. Both dekatrons will start at the top.

Dekatron dekaTwo

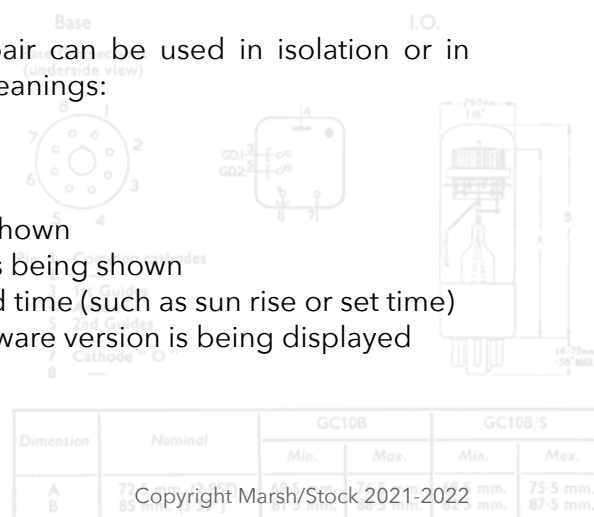
Syntax: Effect effect_two, tube, cathode1, cathode2 [, pause]

Highlight two cathodes on Dekatron tube by rapidly moving between them and pausing on each. The two positions are given by cathode1 and cathode2. The pause time is in mS (default value 1000 mS). Dekatron V7 at the bottom of the clock has engraved labels so the Dekatron can be used to indicate what is displayed on the top six dekatrons. Each of the labels has a corresponding constant that can be used in this action.

Time, date, longitude, latitude, elevation, face and pair can be used in isolation or in combination with UTC, sun and system to give more meanings:

Examples:

- Time indicated alone means the local time is being shown
- Time in conjunction with UTC means that UTC time is being shown
- Time in conjunction with sun shows that a sun related time (such as sun rise or set time)
- Time in conjunction with system shows that the hardware version is being displayed



Label	Meaning	Constant	Cathode
Idle	the V7 indicator function is not used	dot_idle	0
Time	a time is displayed	dot_time	1
Date	date	dot_date	2
Long	longitude	dot_longitude	3
Lat	latitude	dot_latitude	4
Elev	elevation	dot_elevation	5
Face	face number is shown or being changed	dot_face	6
Pair	pair value is shown or being changed	dot_pair	7
UTC	date or time is UTC	dot_utc	8
Sun	time, longitude, latitude relate to the sun	dot_sun	9
Sys	system data	dot_system	10

Obviously the last bullet example is not intuitive and you need to know the system+ meanings. The system dot can be joined with:

Second dot indicated	Meaning
Time	Hardware version
Date	Firmware version
Long	Core temperature
Lat	LDR data
Elev	GPS data
Face	WiFi data

The *dekaTwo* action is used primarily to display on Dekatron V7 what the other six are being used to display.

Example:

Dekatron *dekaWrite*, "%02d%02d%02d", hour, minute, second

Dekatron *dekaCathode*, V7, dot_time

Will show the time on dekatrons V1 to V6 and indicate "time" on Dekatron V7.

Example:

Dekatron *dekaWrite*, "%02d%02d%02d", utc_hour, utc_minute, utc_second

Dekatron *dekaTwo*, V7, dot_utc, dot_time

The UTC time is shown on dekatrons V1 to V6 and, "UTC" and "Time" shown on Dekatron V7.

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm	75 mm	75 mm	75 mm	75 mm
B	85 mm	85 mm	85 mm	85 mm	85 mm

Example:

Dekatron *dekaWrite*, "%02d%02d%02d", fw_major, fw_minor, fw_build
Dekatron *dekaTwo*, V7, dot_system, dot_date

The firmware version is shown on V1 to V6 and "firmware" is shown by indicated "Sys" and "Date" on Dekatron V7.

Dekatron *dekaWrite*

Syntax: **Dekatron** *dekaWrite*, *str* [, ...]

Write the string *str* to the top six dekatrons, V1 .. V6. The string can contain placeholders for the optional comma separated arguments that follow. Each numerical value lights the cathode corresponding to the character '0' ... '9' in the string.

Example:

Syntax: **Dekatron** *dekaWrite*, "012375"

Will light up the top row of dekatrons in this test pattern.

Obviously the Dekatron tubes are limited to displaying the numbers '0' to '9' so any other character in the string *str* will be ignored by the corresponding tube. It is recommended to use the space character. Only the first six characters of the string will be used and any other characters that follow are ignored. If the string is shorter than 6 characters then the remaining tubes are not changed.

Examples:

Dekatron *dekaWrite*, "%02d%02d%02d", hour, minute, second

Dekatron *dekaWrite*, "%6d", face_curr

Dekatron *dekaWrite*, "%2d%2d%2d", fw_major, fw_minor, fw_build

For a guide to the syntax for placeholders see Annex B of the main Nuggle manual. In positioning numbers and spaces (off tubes) in the string you will find the field flags of particular importance.

Dekatron *dekaZero*

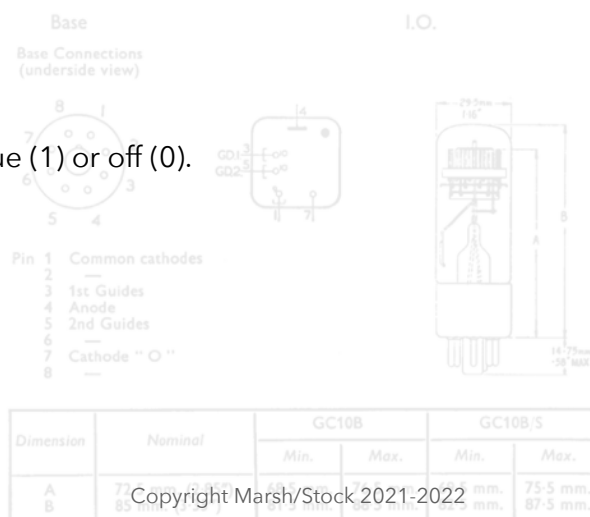
Syntax: **Dekatron** *dekaZero* [, tube, direction, pace]

Move the glow on dekatron *tube* position to cathode '0' (12 o'clock). If *tube* is omitted then all the dekatrons are zeroed and any direction and pace values are ignored. Zeroing all tubes simultaneously is meant to be a fast as possible action.

Flyback

Syntax: **Flyback** *state*

Turns the Dekatron HT supply to *state*. Where *state* is true (1) or off (0).



GPIO

Syntax: GPIO *pin*, *value*

This sets the state of a GPIO *pin* to logical 1 if *value* is true and logical 0 if *value* is false. The available values of *pin* and the GPIO port and pin controlled are as follows:

Pin	Location	SAM3X8C port and pin
5	Back of PCB	Port A Pin 5
7	Back of PCB	Port A Pin 7
10	Pin 3 Connector X8	Port A Pin 10
11	Pin 2 Connector X8	Port A Pin 11
25	Back of PCB	Port B Pin 25
28	Back of PCB	Port B Pin 28
29	Back of PCB	Port B Pin 29
31	Back of PCB	Port B Pin 31
500	500V flyback on/off	N/A

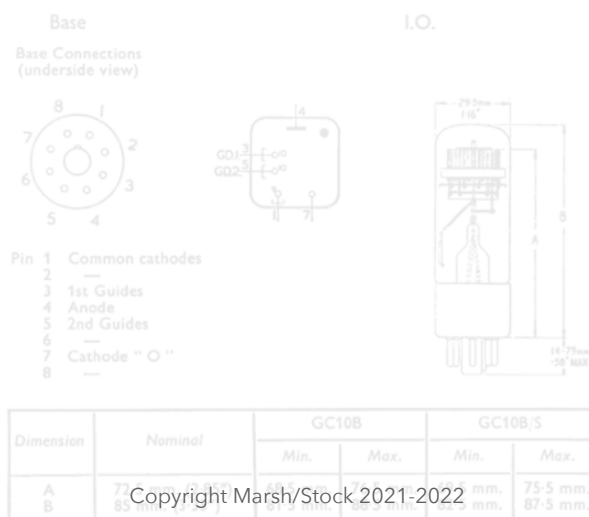
Saver

Syntax: Saver *value*

Specify the "screen saver" display to be used with the current clock face overriding the default saver. The argument value is the index of a Nuggle program defined in nuggle.ini.

Example:

Saver 402



10. Events

Event Happenings

There are no additional event happenings.

Event Default Actions

The Harwell Dekatron clock has the following event default actions:

Event Label:	Default Action:
event_encoder:	If the clock is asleep (any stage) it wakes up. Clock face or sound volume changed. See below.
event_ir:	If the clock is asleep (any stage) it wakes up. See below for other default functions.
event_pir:	If the clock is asleep (any stage) it wakes up.
event_sleep:	The event occurs as the clock wakes up from being asleep and the clock goes from being awake progressively to being asleep. See below.
event_enter:	None
event_leave:	When the Nuggle program is stopping.
event_tick:	None

Default Encoder Events

If the encoder is moved clockwise when displaying a clock face, then the next clock face is displayed, if anti-clockwise then the previous face is displayed.

If the encoder button is pressed and it is moved clockwise then the chime sound volume increases, if moved anti-clockwise, then the sound volume decreases.

In summary:

arg_encoder value	Happening	Default Actions
U'	Turned clockwise	Next clock face is shown
+'	Press and turn clockwise	Chime sound volume is increased show current setting on the Dekatron tubes
D'	Turn anticlockwise	Previous clock face is shown
-'	Press and turn anticlockwise	Chime sound volume is decreased show current setting on the Dekatron tubes

Dimension	Nominal	GC108		GC108/S	
		Min.	Max.	Min.	Max.
A	75 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.

Default IR Handset Events

The IR Handset has default actions for all of the buttons.

You can change the volume of the chimes and the beeps, set the pairing number, select a clock face directly or by scrolling through available faces and finally enquire what face number is currently displayed.

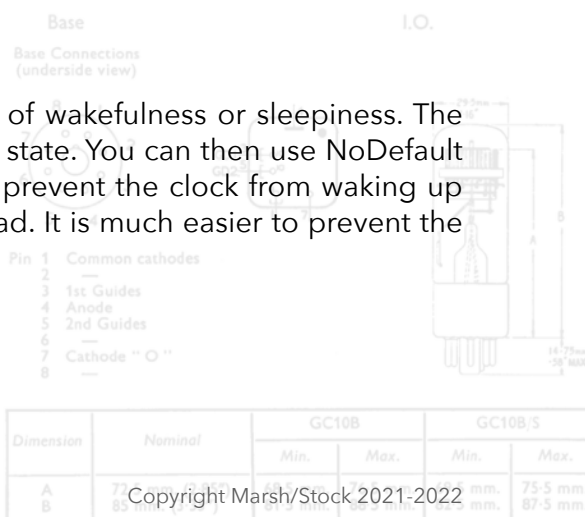
arg_ir value	Happening - button pressed	Default Actions
'U'	CH+	Next clock face is shown
'D'	CH-	Previous clock face is shown
'C'	CH	Display current face on Dekatron tubes
'+'	+	Chime sound volume is increased show current setting on the Dekatron tubes
'.'	-	Chime sound volume is decrease show current setting on the Dekatron tubes
'>'	<<	Beep sound volume is increased show current setting on the Dekatron tubes
'<'	>>	Beep sound volume is decrease show current setting on the Dekatron tubes
'P'	>	Start pair number entry
'0' ... '9'	0 ... 9	Start face number entry
'='	EQ	Pairing number or face number accepted
'H'	100+	In pairing or face entry mode delete last number
'T'	200+	Escape from pairing or face entry mode and discard the entry

Pairing number entry is started with the '>||' key and allows the entry of a number, if it matches the clock pair number then the other keys will operate.

Face number entry allows you to immediately go to a face, if you know its number, without using the CH- and CH+ keys repeatedly.

Default Sleep Events

Invoked when the clock wishes to either go to a higher state of wakefulness or sleepiness. The system variable arg_sleep contains a code to tell you the next state. You can then use NoDefault or Default to allow the state change or not. Obviously, if you prevent the clock from waking up then you will need to use a range of Nuggle commands instead. It is much easier to prevent the clock going to sleep again.



Wake up from being asleep

During the wake up cycle - fully asleep to fully awake - the following states are passed through and actions are taken. The timings of some stages are adjustable in the [sleep] section of system.ini on the SD card.

arg_sleep value	Constant	Actions	[Sleep] key
14	awake_flybackonreq	Request to turn on flyback PSU	
15	awake_flybackon	Flyback PSU turned on	flyback_on = 2 seconds delay
17	awake_complete	Flyback is turned on	
18	awake_release	Request to go to fully awake status	
3	awake_fully	The clock is fully awake	

Going to sleep

During the sleep cycle - going from fully awake to fully asleep - the following states are passed through and actions are taken. The timings of some stages are adjustable in the [sleep] section of system.ini on the SD card.

arg_sleep value	Constant	Actions	[Sleep] key
4	sleep_savercall	Request to invoke the screen saver	
5	sleep_saver	Screen saver displayed	screen_saver = 10 ; minutes delay
6	sleep_flybackoffreq	Request to turn off the flyback PSU	
7	sleep_flybackoff	Flyback PSU turned off	flyback_off = 10 ; minutes delay
9	sleep_release	The clock is about to be fully asleep	
2	sleep_fully	The clock is fully asleep (idle)	

If a wake from sleep trigger is received during these stages the going to sleep process is reversed and the value of *arg_sleep* will change to the relevant waking up state.

If you want to see these processes in action then use the sleep debug system at level 3 (set in system.ini) or write a Nuggle program to do so.

Sleep states, 0, 1, 10, 13 and 16 are used internally and are not sent to the event handler. Sleep states 8, 11 and 12 are not used and are never called.

If the **Default** action takes place at step 5 then the screen saver Nuggle program is run and your program loses control. Your program is then not run again until step 3 has been reached in the wake up cycle. The upshot of this is that your Nuggle program will only receive event steps 4 and 5.



Default Start Events

The default start events are to:

- Set all the Dekatrons to cathode '0'
- Clear all the LEDs to off i.e. black.
- Stop any chime that is playing
- Stop any beep that is sounding
- Set both colons to off, i.e. set the brightness PWM signal to 0 for both colons.

Using **NoDefault** will allow one Nuggle program to take over the display from another without any jitter.

Default Tick Events

The RTC ticks each second and generates an event. The value of the system variable *arg_tick* is set as follows:

arg_tick value	Constant	Highest tick period that has occurred
0	tick_second	1 second
1	tick_minute	1 minute
2	tick_hour	1 hour
3	tick_day	1 day
4	tick_month	1 month
5	tick_year	1 year

For example, a 1 day tick only occurs at 00:00:00 local time. And another example, the year tick only occurs at 00:00:00 on 1st January each year. Time is always local time and takes into account DST and the time zone. The clock UTC tick is not used to generate events.

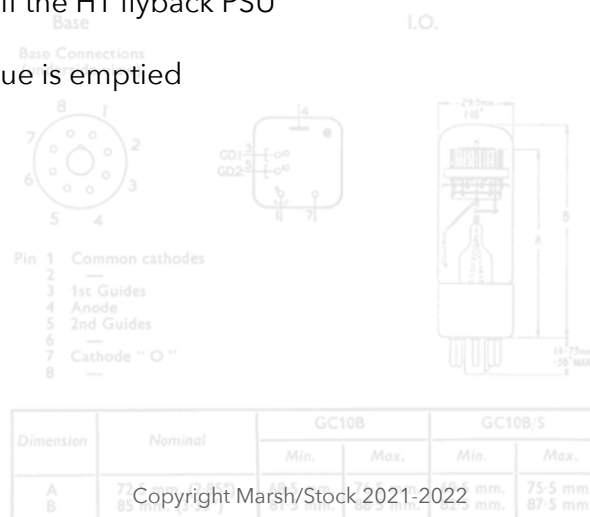
Miscellaneous Procedures

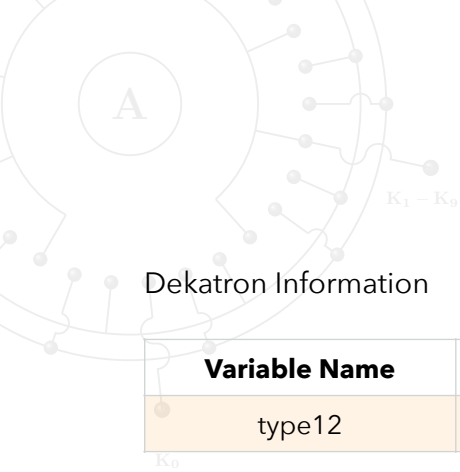
Clear

Syntax: Clear

The following happens:

1. Dekatron and colon tubes are blanked by turning off the HT flyback PSU
2. All bling LED are set to *black*, i.e. off
3. Any beep in progress is stopped and the beep queue is emptied
4. Any chime in progress is stopped





Annex A System Variables

Dekatron Information

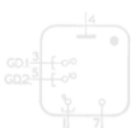
Variable Name	Function	Notes
type12	Is V7 a 12 cathode Dekatron?	boolean



Base
Base Connections
(underside view)



- Pin 1 Common cathodes
- 2 —
- 3 1st Guides
- 4 Anode
- 5 2nd Guides
- 6 —
- 7 Cathode "O"
- 8 —

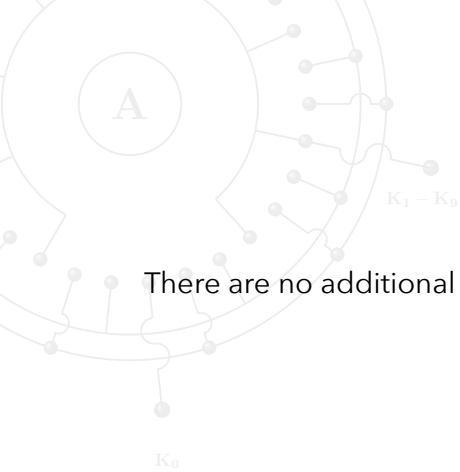


I.O.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72 mm				
B	85 mm				

Copyright Marsh/Stock 2021-2022



Annex C Fatal Error Messages

There are no additional fatal error messages.



Base
Base Connections
(underside view)



- Pin 1 Common cathodes
- 2 —
- 3 1st Guides
- 4 Anode
- 5 2nd Guides
- 6 —
- 7 Cathode "O"
- 8 —

I.O.



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	72 mm			mm.	75.5 mm.
B	85 mm			mm.	87.5 mm.



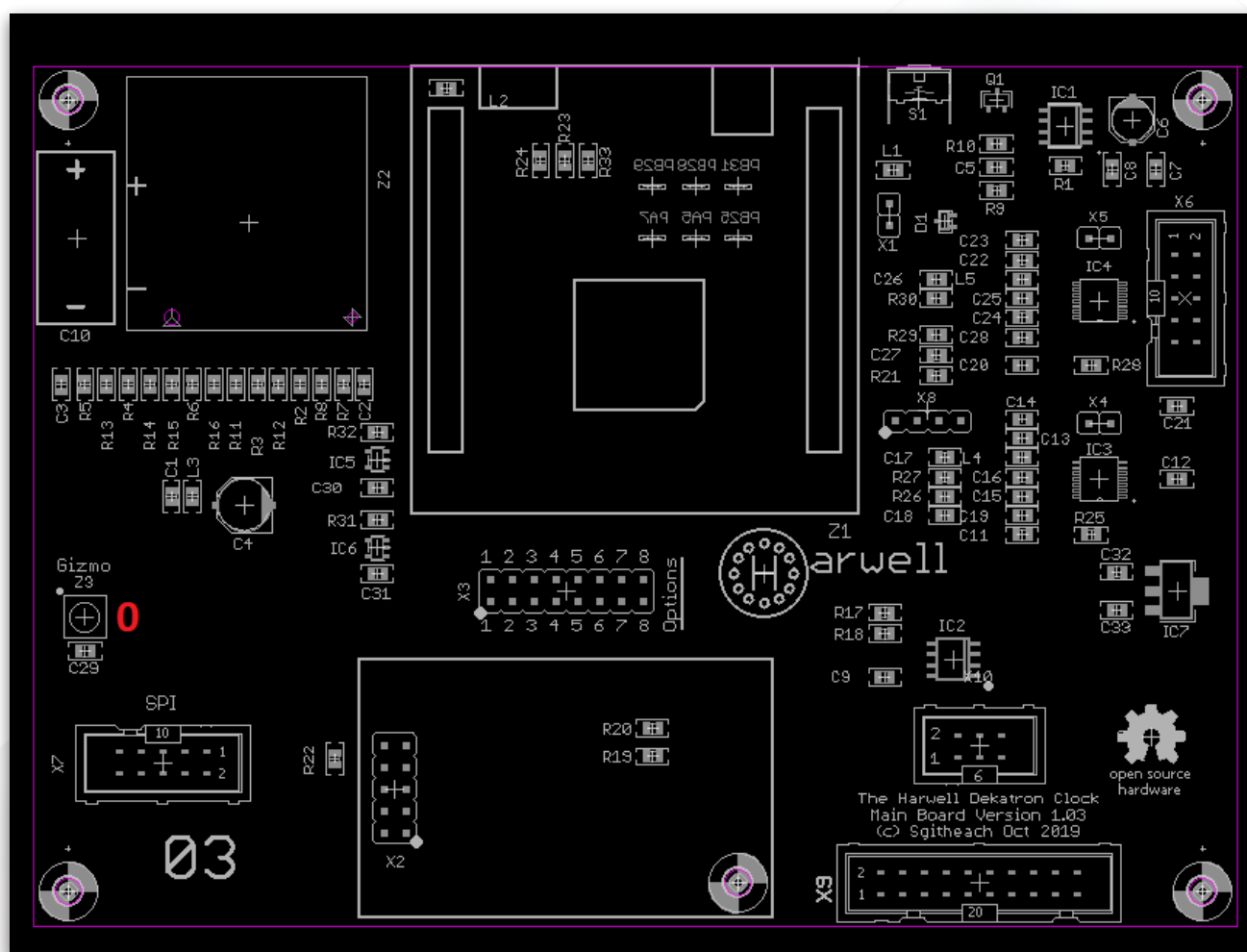
Annex D Bling LED Numbering

Example of use:

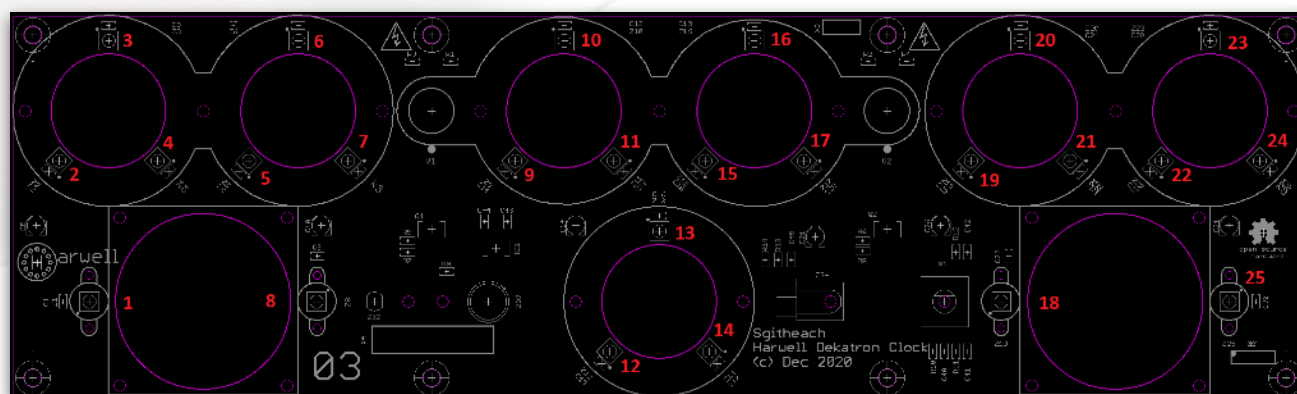
Colour red, 14

Colour red, 23

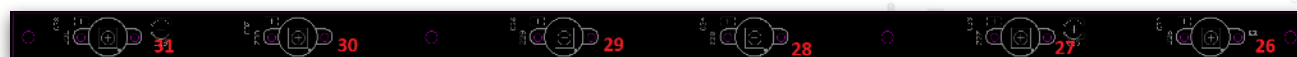
Position numbers of the bling LED modules as viewed from the silkscreen side of the PCB.



Only one bling LED on the Main Processor board - '0'



Bling LED numbering on the light guide board - 1 to 25



Bling LEDs 26 to 31 on the downward illumination board

		Nominal	Min.	Max.	Min.	Max.
A	B	75			mm.	75.5 mm.
		85			mm.	87.5 mm.

Annex E - Dekatron 7 Usage

The Harwell Dekatron Clock uses six 10 cathode Dekatrons in a row to show date, time and other information, and a seventh Dekatron (V7) below to indicate what is being displayed. V7 can also provide eye-candy by spinning, acting as a pendulum and showing other effects.

The standard case is engraved around V7 to indicate what is being shown on the six Dekatrons. V7 can be a 12 cathode Dekatron (GC12/4B) or use a more common 10 cathode Dekatron. The case engraving changes accordingly.

12 cathode function engravings (from the top clockwise) are

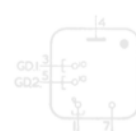
- (no engraving)
- Time
- Date
- Lat
- Long
- Elev
- Face
- Pair
- Vol
- UTC
- Sun
- Sys

and the 10 cathode engravings are

- Sys
- Time
- Date
- Lat
- Long
- Elev
- Face
- Pair
- Vol
- UTC

However, there are more things to display than there are available cathodes, so V7 sometimes uses a mode where two cathodes are illuminated alternately to highlight two functions. For example, when Time alone is indicated it means that the current local time (adjusted for DST and time zone) is displayed, but when Time and UTC are indicated alternately it means the UTC time is being shown.

(underside view)



Dimension	Nominal	GC10B		GC10B/S	
		Min.	Max.	Min.	Max.
A	75 mm			75.5 mm	
B	85 mm			87.5 mm	



The following table lists the displayed value, how V7 is used and how the 10 or 12 cathode variants are dealt with.

Displayed Value	12 cathode engraving	10 cathode engraving
Local Time	Time	
Local Date	Date	
UTC Time	UTC + Time	
UTC Date	UTC + Date	
Latitude (note 1)	Lat	
Longitude (note 2)	Long	
Elevation	Elevation	
Sunrise	Sun+Time	Pair+Time
Sunset	Sun+Date	Pair+Date
Sun's azimuth	Sun+Long	Pair+Long
Sun's elevation	Sun+Elev	Pair+Elev
Current face number or face number entry	Face	
Current Pair number or pair number entry	Pair	
Current chime or beep volume	Vol	
Hardware version	Sys+Time	
Firmware version	Sys+Date	
Chip temperature	Sys+Long	
LDR reading	Sys+Lat	
GPS information	Sys+Elev	
WiFi information	Sys+Face	
Idle	Idle (note 3)	Sys

Notes.

1. The bling LEDs are used to indicate north or south of the equator.
2. The bling LEDs are used to indicate east or west of Greenwich.
3. Idle is not engraved.

