Fortress

Fortress - An E1T Timepiece

Six Miniature CRTs In Action

Copyright Grahame Marsh/Nick Stock 2019

Fortress[1]

What is the Fortress? It's the culmination of an awful lot of work by two like-minded individuals who like to tinker with vintage display technologies and turn them into functional and unusual time-pieces.[2] This clock is somewhat unique in many respects. It is, to our knowledge, the only 'commercial' E1T clock deliberately offered for sale <u>*ever*</u>. Others have offered PCBs etc, but this is the *whole enchilada* and as such is a rarity to be treasured and enjoyed.[3] Fortress is a combination of old-skool technology married with a very modern micro-controller and 21st century case construction and aesthetics. It offers lots of customisation options, such as different 'faces'[4], light-guided LED colours, 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 Fortress E1T clock, replete with case of your choosing[5], 6 tested E1T tubes[6] and either a GPS or WiFi module (both come with external antennae, depending on your order).
- A pre-built 5A power supply and IEC mains cable.[7]
- A custom programmer used to update firmware and act as a serial console connection.
- An Infrared remote.

What is an E1T Tube?

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 Systems[8]), neon gas based displays such as Nixie tubes[9] and Dekatrons[10], vacuum fluorescent displays[11] (VFDs), light-emitting diodes[12] (LEDs) and liquid crystal displays[13] (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*[14]. What makes the E1T (or S10S1, the East German Clone) stand out from these technologies is that it represents the adaptation of

[1] Throughout this document, you will note the use of hyperlinks which both lead you to external references as well as internal bookmarks.
[2] Stay tuned for more!
[3] However, you can purchase *just the PCBs* if you wish….see, how's that for being flexible! 😀
[4] And a whole programming language (Nuggle) so you can design your own colourful displays!
[5] Currently, we can offer the case in any colour of acrylic, but the standard is a clear laser cut case, or for a little extra, a glass effect finish with a professional welded case top. At the time of writing a premium, special edition brushed-anodised aluminium version is in preparation.
[6] 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 rare as hens-teeth, but occasionally appear on eBay and the like.
[7] If readily available to us for your part of the world……
[8] https://stevenjohnson.com/nls/index.htm
[9] https://en.wikipedia.org/wiki/Nixie_tube
[10] https://en.wikipedia.org/wiki/Dekatron
[11] https://en.wikipedia.org/wiki/Vacuum_fluorescent_display
[12] https://en.wikipedia.org/wiki/Light-emitting_diode
[13] https://en.wikipedia.org/wiki/Liquid-crystal_display
[14] Beauty is in the eye of the beholder….

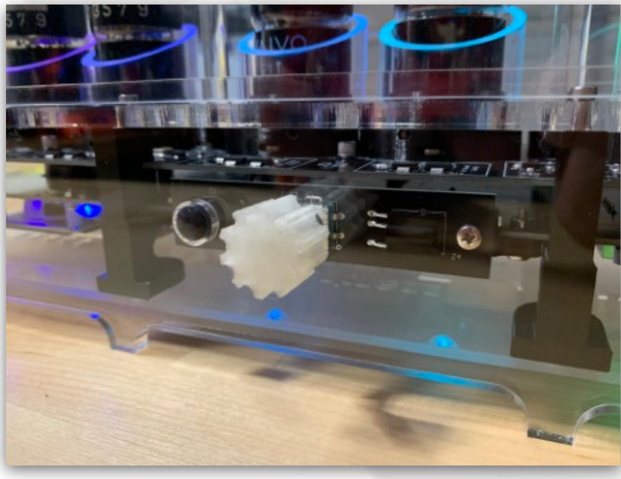
Copyright Grahame Marsh/Nick Stock 2019

another vintage display technology, the *cathode ray gun*[15] into a discrete counting device. The E1T uses a cathode ray and some neat tricks to display a green line in 10 positions, each cunningly numbered from 0 to 9. These were used in early counters (such as the one pictured on the right). A far more detailed explanation of how to electrically drive an E1T can be found later on in this document. There are also some great resources on the web for history and data about the E1T.[16]

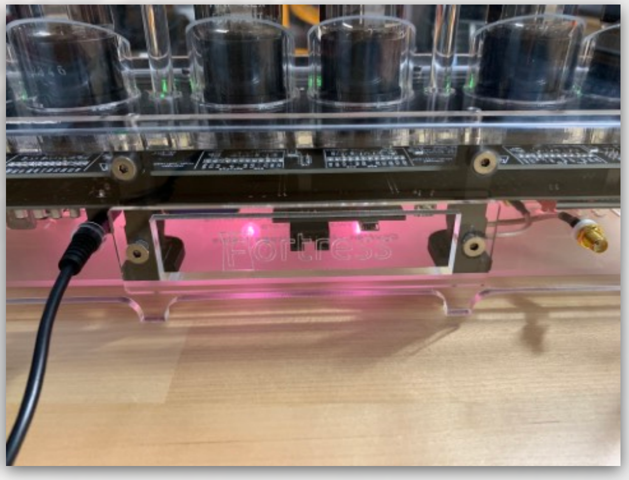
Anatomy of a Fortress

The clock consists of the following components;

- Six E1T tubes encased with raked cast acrylic light guides.[17]
- Two acrylic light guide rod colons.
- PIR[18] - 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.[19]
- Two high quality downward firing speakers - the Fortress is capable of using .WAV files to play chimes and audio.[20]
- 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 5A output).[21]
- A case top/cover for the clock.



Left: PIR, **Middle:** Rotary encoder, **Right:** InfraRed receiver



Left: PSU input, **Middle:** Access plate for SD and CPU socket, **Right:** GPS/WiFi antenna connector

---

[15] https://en.wikipedia.org/wiki/Cathode-ray_tube

[16] http://www.tube-tester.com/sites/nixie/different/e1t-tubes/E1T_philips/e1t-phil.htm; https://www.dos4ever.com/E1T/E1T.html

[17] As interesting as the E1T is as a piece of vintage CRT technology, we thought we'd spruce things up with a little extra bling….which you can turn off if you so desire!

[18] PIR - Passive InfraRed sensor.

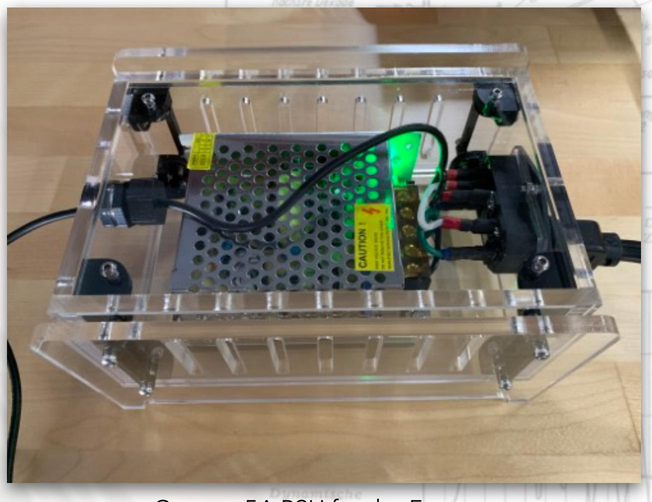
[19] For clear acrylics, these are IR transparent, but opaque materials and aluminium cases have a cut-out for the IR receiver.

[20] Some of it *quite amusing*.

[21] The clock doesn't require a continuous 5A input, but upon startup the heaters require quite a bit of *juice* to warm up. This is mitigated a lot by the slow startup implementation in the clock hardware, but current spikes of approximately 3A are not uncommon from a cold start.

Copyright Grahame Marsh/Nick Stock 2019

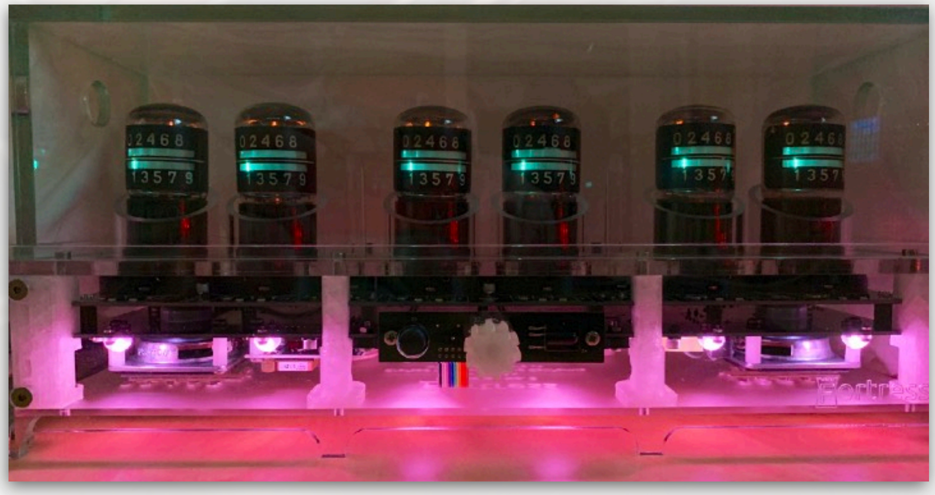
One of two speakers on the *underside* of the Fortress



Custom 5A PSU for the Fortress

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 E1T displays and the LED highlighting. The clock does have an auto adjustment for the LED intensity based on the ambient light in the room.[22] 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.[23] You will also want to attach either the GPS/WiFi antenna to the back of the clock. If you're using the WiFi option you will need to enter your network credentials (SSID[24] 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 minute, the E1T tubes will illuminate and countdown to 0.[25] You will then be greeted with the first of many available 'faces' of Fortress. To change from one to the next, either use the rotary encoder at the front of the clock or the remote control (see page 67 for a full breakdown of the available remote control operations). To change the volume of any chime



Powering up the Fortress - E1T tubes counting down on startup

[22] This can be changed, like most other presets in the clock, by altering the *system.ini* file on the SD card. See Page 60.
[23] 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.
[24] SSID - Service Set Identifier - The given name to your local WiFi network.
[25] Why does it take this long you ask? Well, there's a lot going on behind the scenes....and the slow start of the E1T heaters hopefully prolongs their working life. If you want to see what checks etc. go on during startup, then see Page 47 and use the serial console.

 Copyright Grahame Marsh/Nick Stock 2019

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 E1T tubes). You can also use the IR remote for the same purpose.

Clock Customisation

One of the core principles embedded in an Open Source device such as Fortress is the ability to customise it by *understanding* how it ticks[26]. Faces for the clock are written in a hybrid coding language called 'Nuggle' and exemplary examples are provided in a separate manual called 'The Nuggle Cookbook' available on the Dropbox. As supplied, the Fortress 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 chimes. 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 13.

Clock Care and Troubleshooting

Fortress 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 E1T tubes are very valuable and (alas) a non-renewable resource, so please only operate the clock with the case top in place to protect them from accidental contact and stray flying objects.[27] As with most complicated electronic devices, sometimes things don't go according to plan.[28] 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 47) to see where things are *hanging up*. 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 E1T tubes as these are obviously out of our control and have a finite lifespan. We will, however, warranty any issue in the Fortress 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.[29] 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.

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 Fortress or any other device offered by us, please contact us at either;

web.sgitheach@googlemail.com or **stocksclocks@gmail.com**

---

[26] See what I did there? Ha ha ha....

[27] This may seem a little OTT, but as a favourite author of mine once wrote, '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.' - Mort, Sir Terry Pratchett.

[28] This should be a *rare* event.

[29] Common sense should prevail here.

 Copyright Grahame Marsh/Nick Stock 2019

Table of Contents

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

web.sgitheach@googlemail.com or **stocksclocks@gmail.com**

Copyright Grahame Marsh/Nick Stock 2019

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.[30] 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).[31]

iv) Plug the clock in (12V DC, minimum 5A, centre positive) - Clock will take about 1 minute 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 (time - page 37) and 961 (date - page 38).

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

web.sgitheach@googlemail.com or **stocksclocks@gmail.com**

[30] Ignore this if you've already asked us to take care of this at purchase.
[31] You may have not opted for either, which is OK, but you'll have to set the correct time yourself.

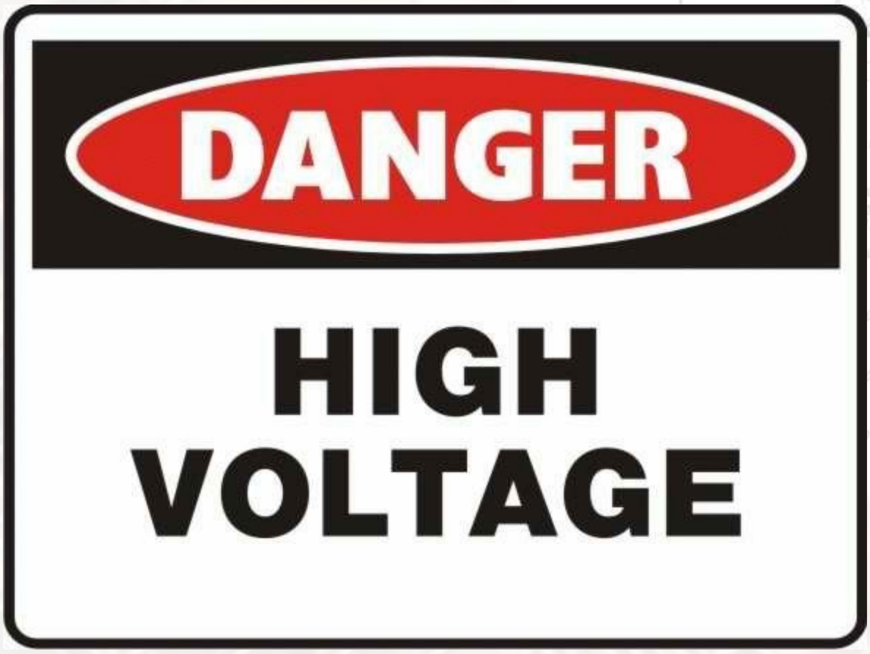
Copyright Grahame Marsh/Nick Stock 2019

SAFETY AND LEGAL[32]

SAFETY, LEGAL & LICENSE CONSIDERATIONS

[32] You know, the fun stuff….

 Copyright Grahame Marsh/Nick Stock 2019



Annex - Safety and Legal Statements

Safety Statement

Like all scope clocks, The Fortress uses high voltages in order to operate the cathode ray tube (CRT) inside the E1T tube. You need to respect this and other hazards inherent in these circuits.

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

Caution! The Fortress generates high voltages in the region of 300V during operation. These voltages are present on the power supply board and base of the E1T 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![33]

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.[34]

Caution! The E1T tube envelope is made of glass and may be broken if the clock is dropped or inadvertently struck.[35]

[33] It wouldn't taste very good…

[34] Why you would given the rather exquisite case the clock is supplied in is beyond the author's imagination…

[35] Keep this out of the way of stray pets…

 Copyright Grahame Marsh/Nick Stock 2019

Legal Statement

The Fortress 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;

i) You may share, copy and redistribute the material in any medium or format,
ii) You may remix, transform and build upon the material presented herein,
iii) You MUST give appropriate credit, provide a link to the license and *indicate if changes have been made*.
iv) This license is for NON-COMMERCIAL use only, you may not use the material for commercial gain.
v) 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 Fortress 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, the presence of high vacuum within the cathode ray tube 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 Fortress 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.

Copyright Grahame Marsh/Nick Stock 2019

Atmel Licence

The Fortress firmware makes extensive use of the Atmel[36] libraries. The Atmel License (also given in every Atmel library file) is;

Copyright (c) 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:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. 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.
3. The name of Atmel may not be used to endorse or promote products derived from this software without specific prior written permission.
4. 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 Fortress uses a custom SAM3X8C core board and as such makes use of a programmer that uses LUFA.

LUFA Library
Copyright (C) Dean Camera, 2010.

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

[36] Now owned by MicroChip

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 author 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, sub-license, 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[37] 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 Fortress clock may be hazardous if not assembled and operated by suitably knowledgable 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[38] technology employed in the Fortress clock (primarily the E1T 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.

[37] Please, no requests months *after the fact*…I'm reasonable but *not that reasonable*.
[38] But charming!

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 the insurance claim.

No refunds on partially or fully constructed kits are possible.

 Copyright Grahame Marsh/Nick Stock 2019

Fortress - Clock Distribution Faces[39]

The Standard Face Feature Set

[39] For the latest Fortress Faces distribution, see the Fortress Face Supplemental on the DropBox

Copyright Grahame Marsh/Nick Stock 2019

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!![40]

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 Fortress E1T 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 file[41] 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.

---

[40] We look forward to what you come up with…..and this is Open Source *after all*.
[41] In the */system* folder on the SD card

Copyright Grahame Marsh/Nick Stock 2019

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 E1T tubes off and bling set to black	400	*blank.nug*
2	Show the time with a spinner but no excessive bling or chimes. A very simple display.	106	*time11.nug*
3	Time with bling and chimes	201	*nick02.nug*
4	Time with date shown each minute. Bling but no chimes.	121	*dattim02.nug*
5	Time with date shown each minute. Bling and Derby chimes.	123	*dattim04.nug*
6	Original Star Trek themed clock	152	*ost01.nug*
7	Sequential display of four other clock faces	7 calling 251 252 253 254	*callsq10* *seq10a.nug* *seq10b.nug* *seq10.cnug* *seq10d.nug*
8	Sun rise time	301	*sunrise.nug*
9	Sun set time	302	*sunset.nug*

However, 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 satisfactory[42] - 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 case 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	Star Trek Next Generation themed clock	154	*tng01.nug*
11	"PIR Game"	350	*pir.nug*
12	Larson bling scanner in red	404	*larson.nug*

[42] .. 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.

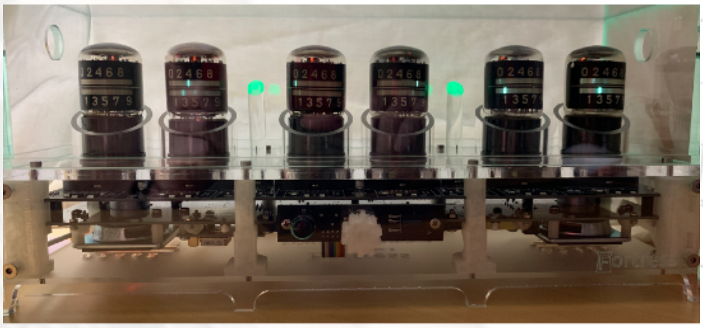
 Copyright Grahame Marsh/Nick Stock 2019

Distribution Clock Faces in Groups

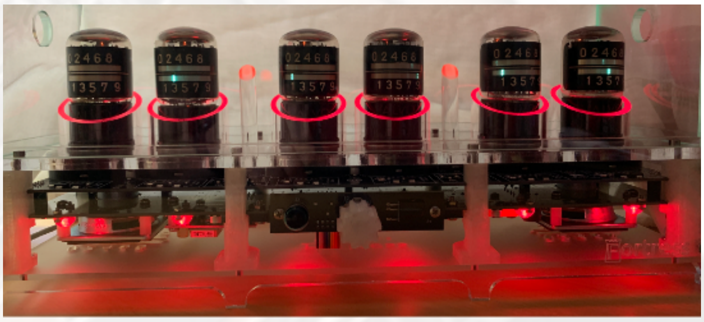
100 - Illustrating faces that just show the time with minimal bling

All these faces respect the user settings for 12 or 24 hour displays and as such the 10's hours E1T should be blanked when displaying the time from 0 to 9.

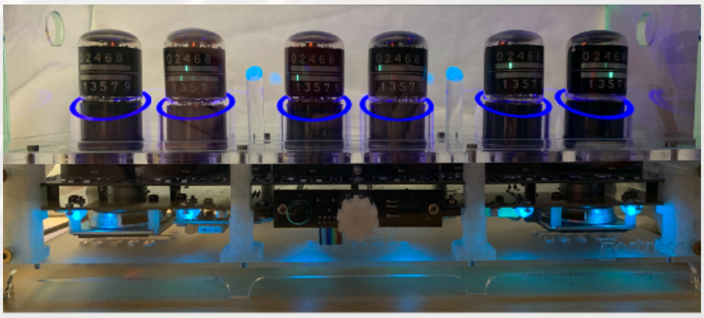
Face number	**100**
Nuggle file	*time01.nug*
Description	Simple time display, respect user format setting
Colons	Flash green
Bling	None
Chimes	None



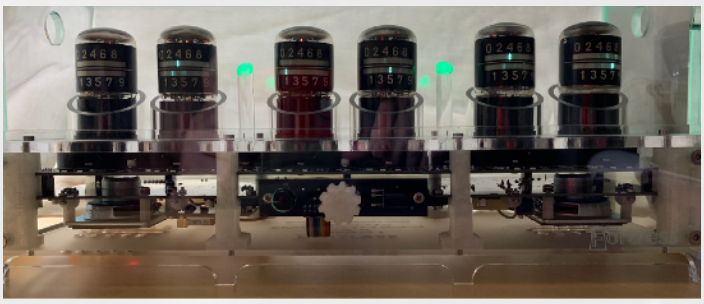
Face number	**101**
Nuggle file	*time02.nug*
Description	Simple time display, respect user format setting
Colons	Flash red
Bling	Dim red
Chimes	None



Face number	**102**
Nuggle file	*time03.nug*
Description	Simple time display, respect user format setting
Colons	Flash cyan
Bling	Dim blue and cyan
Chimes	None

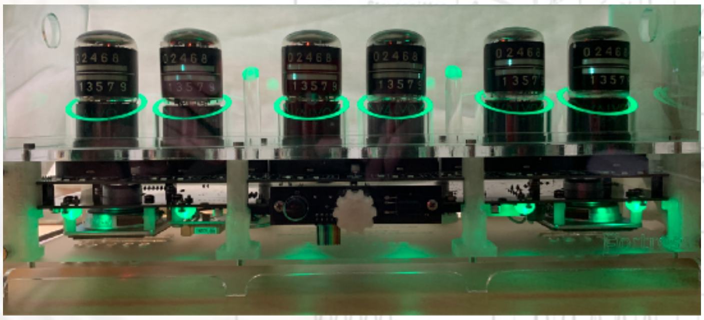


Face number	**105**
Nuggle file	*time10.nug*
Description	Simple time display, respect user format setting. Spin the seconds each second then all tubes on each minute roll over
Colons	Flash green
Bling	None
Chimes	None



Copyright Grahame Marsh/Nick Stock 2019

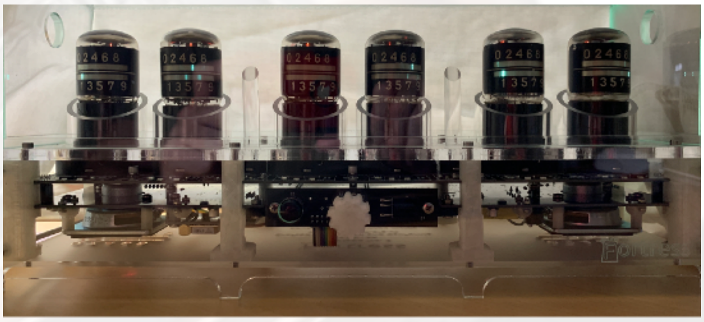
Face number	106
Nuggle file	*time11.nug*
Description	Simple time display, respect user format setting. Spin the seconds each second then all tubes on each minute roll over
Colons	Flash green
Bling	Dim green
Chimes	None



110 - Illustrating faces that just show the date with minimal bling

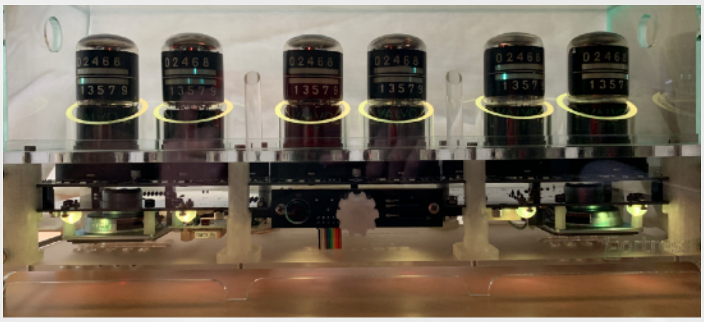
All of these faces respect the user settings for year, month and day leading zeros and the display order.

Face number	110
Nuggle file	*date01.nug*
Description	Simple date display, respect user format settings
Colons	None
Bling	None
Chimes	None



Face number	111
Nuggle file	*date02.nug*
Description	Simple date display, respect user format settings
Colons	None
Bling	White theme
Chimes	None

Face number	112
Nuggle file	*date03.nug*
Description	Simple date display, respect user format settings
Colons	None
Bling	Yellow theme
Chimes	None

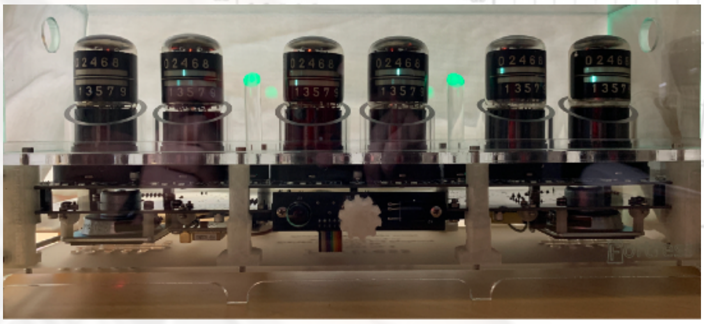


Copyright Grahame Marsh/Nick Stock 2019

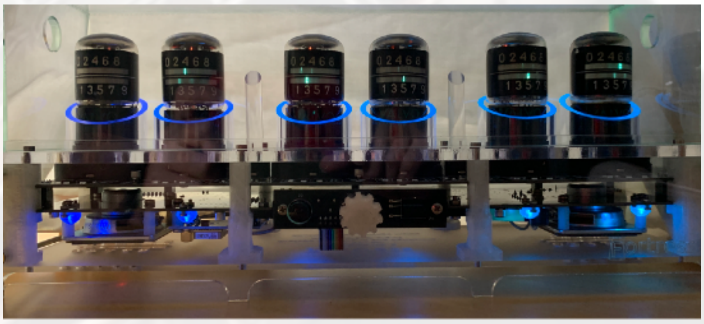
120 - Faces that show the time and date sequentially during each minute, either no or rainbow bling

All these faces respect the user settings for year, month and day leading zeros and the display order when showing the date. They also respect the user settings for 12 or 24 hour displays and if the 10's hours E1T should be blanked when displaying the time.

Face number	**120**
Nuggle file	*dattim01.nug*
Description	Time is shown except during seconds 40 to 49 when the date is shown
Colons	Showing time - flash green Showing date - fixed green
Bling	None
Chimes	None

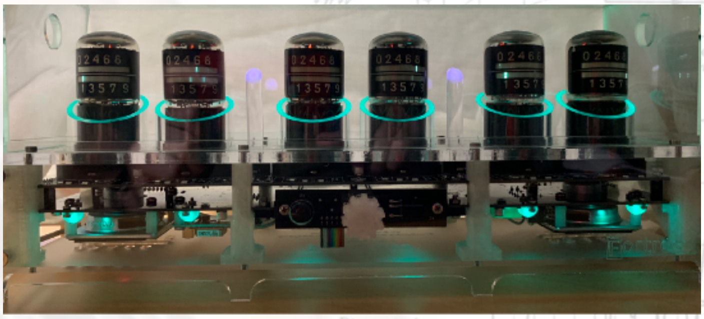


Face number	**121**
Nuggle file	*dattim02.nug*
Description	Time is shown except during seconds 40 to 49 when the date is shown
Colons	Showing time - flash white showing date - fixed white
Bling	Rotating rainbow of colours, colour progress round the colour wheel and right to left across the tube bling at a constant rate
Chimes	None



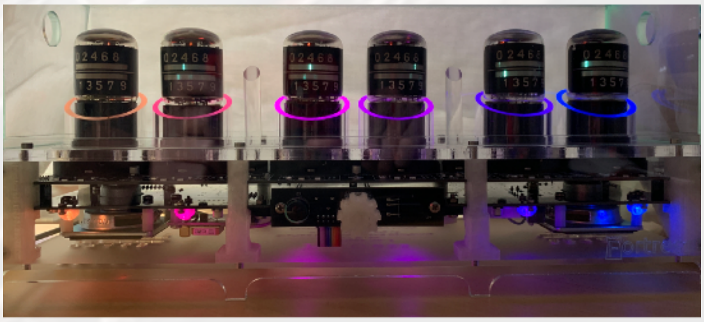
Copyright Grahame Marsh/Nick Stock 2019

Face number	122
Nuggle file	*dattim03.nug*
Description	Time is shown except during seconds 40 to 49 when the date is shown
Colons	Showing time - flash white Showing date - fixed white
Bling	All tube and downward bling show the same HSV colour. The IR handset***** can be used to switch to a hue using keys 0 to 7. Keys 8 and 9 are ignored but the rest of the keys operate as normal.
Chimes	none



***** Because the IR handset key's 0 to 9 default operation are overridden by the Nuggle code the use of keys 0 to 9 to dial in a new face number won't work.

Face number	123
Nuggle file	*dattime04.nug*
Description	As with 121 *dattim02.nug* except bling has a wider hue range and chimes are added
Colons	Showing time - flash white Showing date - fixed white
Bling	Rotating rainbow of colours, colour progress round the colour wheel and right to left across the tube bling at a constant rate
Chimes	derby00.wav, derby15.wav, derby30.wav, derby45.wav, derbyt.wav, tick01.wav



 Copyright Grahame Marsh/Nick Stock 2019

130 - Westminster Quarters Set

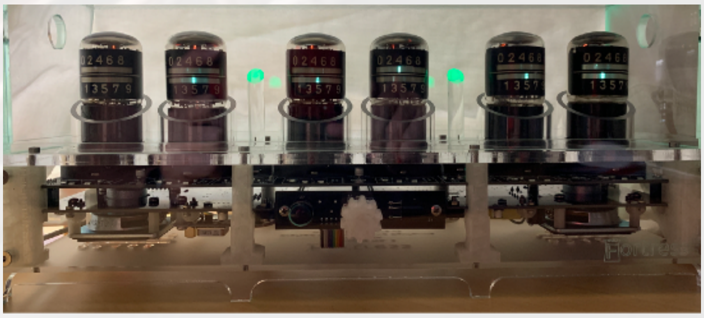
The Fortress E1T clock has three sets of Westminster Quarters[43],[44] WAV files including compatible tick-tock WAV files. I am reasonably sure that none of these files are the actual sound of the bells used at the palace of Westminster[45]. Like Marmite[46] you will either hate or love these chimes.

A point of note please: Big Ben is the nickname for the Great Bell of the clock at the north end of the Palace of Westminster in London that tolls the hours. The official name of the tower in which Big Ben is located was originally the Clock Tower, but it was renamed Elizabeth Tower in 2012 to mark the Diamond Jubilee of Elizabeth II.

These first sets of clock faces concentrate on the chimes and use no bling.

Face number	130
Nuggle file	*awest.nug*
Description	Shows the time as hh:mm:ss using flashing colons, no bling, tick-tock and Westminster files set A.
Colons	Colons flash green each sec.
Bling	None
Chimes	a-west00.wav, a-west15.wav, a-west30.wav, a-west45.wav, a-westt.wav, tick2a.wav, tick2b.wav

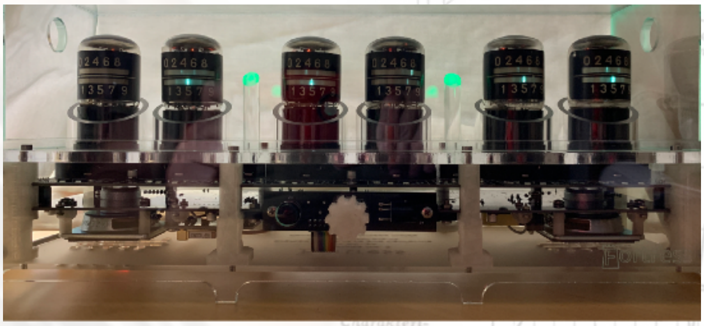
Face number	131
Nuggle file	*bwest.nug*
Description	Shows the time as hh:mm:ss using flashing colons, no bling and Westminster Quarters file set B.
Colons	Colons flash green each sec.
Bling	None
Chimes	b-west00.wav, b-west15.wav, b-west30.wav, b-west45.wav, b-westt.wav



43 https://en.wikipedia.org/wiki/Westminster_Quarters
44 Otherwise known as the Cambridge Chimes
45 https://en.wikipedia.org/wiki/Big_Ben
46 https://www.marmite.co.uk/

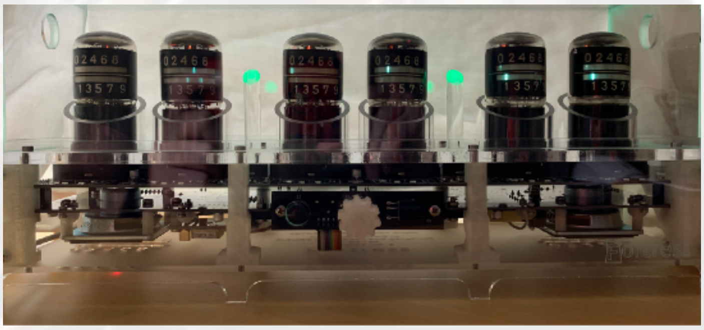
Copyright Grahame Marsh/Nick Stock 2019

Face number	132
Nuggle file	*cwest.nug*
Description	Shows the time as hh:mm:ss using flashing colons, no bling and Westminster Quarters file set C
Colons	Colons flash green each sec.
Bling	None
Chimes	c-west00.wav, c-west15.wav, c-west30.wav, c-west45.wav, c-wt1.wav, c-wt2.wav, c-wt3.wav, c-wt4.wav, c-wt5.wav, c-wt6.wav, c-wt7.wav, c-wt8.wav, c-wt9.wav, c-wt10.wav, c-wt11.wav, c-wt12.wav

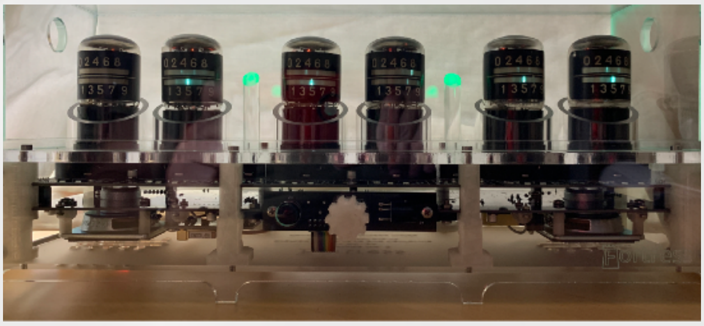


135 - Other Chimes

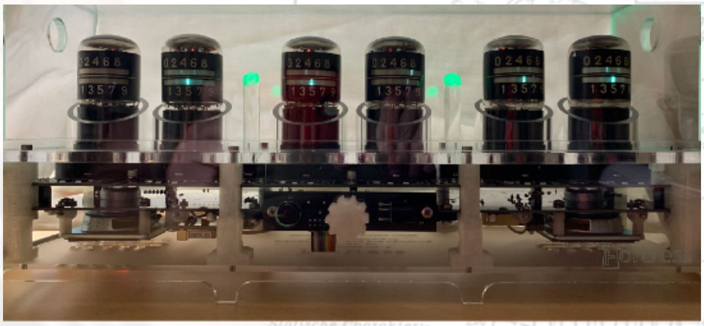
Face number	135
Nuggle file	*amaria.nug*
Description	Shows the time as hh:mm:ss Ave Maria Chimes, flashing colons, no bling
Colons	Colons flash green each sec.
Bling	None
Chimes	amaria00.wav, amaria15.wav, amaria30.wav, amaria45.wav, amariat.wav



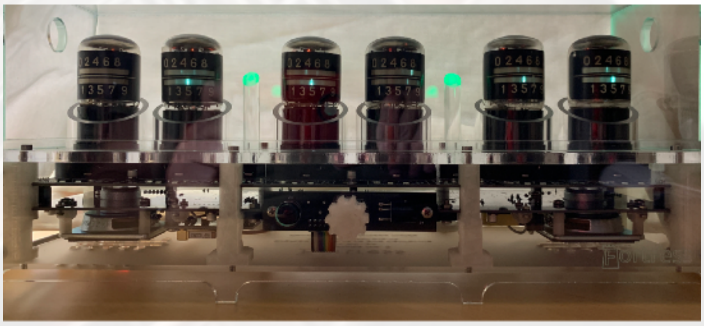
Face number	136
Nuggle file	*whit.nug*
Description	Shows the time as hh:mm:ss Whittington Chimes, ticking, flashing colons, no bling
Colons	Colons flash green each sec.
Bling	None
Chimes	whit00.wav, whit15.wav, whit30.wav, whit45.wav, whitt.wav, tick01.wav



Copyright Grahame Marsh/Nick Stock 2019

Face number	137	
Nuggle file	*steph.nug*	
Description	Shows the time as hh:mm:ss St. Stephen Chimes, ticking, flashing colons, no bling	
Colons	Colons flash green each second	
Bling	none	
Chimes	steph00.wav, steph15.wav, steph30.wav, steph45.wav, stepht.wav, tick01.wav	

140 - Ship's bells

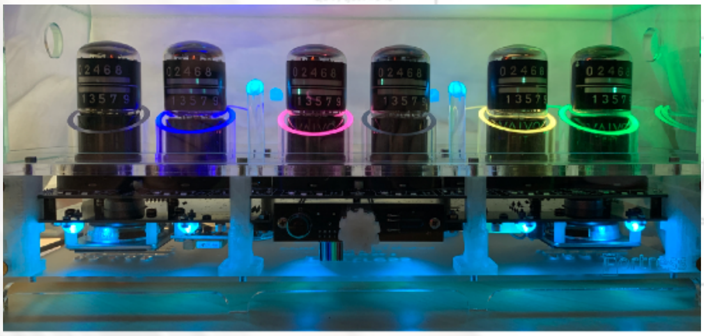
Face number	140	
Nuggle file	*ships.nug*	
Description	Shows the time as hh:mm:ss ships bells, flashing colons, no bling. The bell is from HMS Chepstow. Classic British bells.	
Colons	Colons flash green each sec.	
Bling	None	
Chimes	ship1.wav, ship2.wav, ship21.wav, ship22.wav, ship221.wav, ship222.wav, ship2221.wav, ship2222.wav, ship16.wav	

Copyright Grahame Marsh/Nick Stock 2019

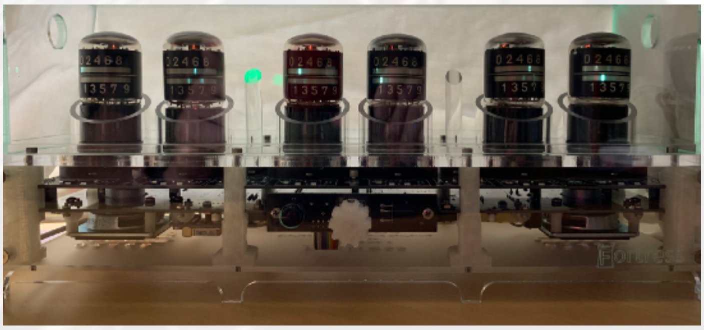
150 - Miscellaneous Faces

Here are some … strange … clock faces

Face number	**150**
Nuggle file	*resistor.nug*
Description	Shows the time as hh:mm:ss.
Colons	Colons flash cyan each second
Bling	The bling around each tube lights with the corresponding colour to the value shown using the resistor colour code numbers. Black = 0, brown = 1 and so on. The downward bling are lit cyan.
Chimes	none

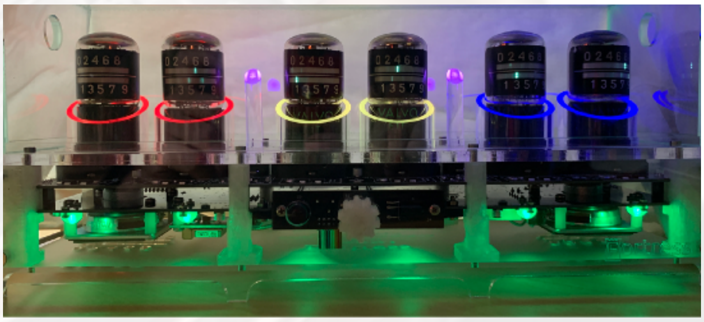


Face number	**151**
Nuggle file	*decimal.nug*
Description	Shows the time as HH.hhhh where HH is the hours 00 to 23 and hhhh is the minutes and seconds expressed as a fraction of an hour. So 07:30:00 would be displayed as 07.5000
Colons	The left colon flashes green every second. Right colon is unused
Bling	None
Chimes	None



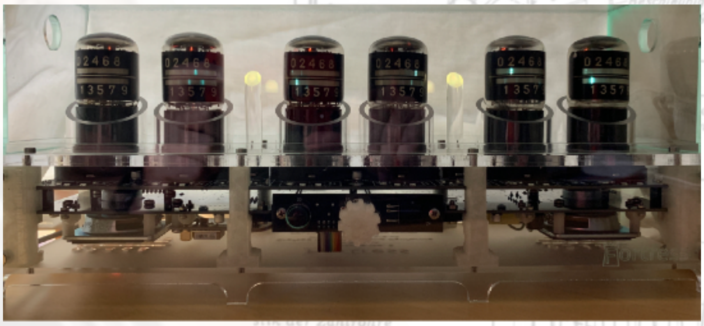
 Copyright Grahame Marsh/Nick Stock 2019

Face number	152
Nuggle file	*ost01.nug*
Description	Original Star Trek themed clock. Clock shows hh:mm:ss
Colons	Static purple - the colour of the sky.
Bling	Two red shirts (security in this case- https://en.wikipedia.org/wiki/Redshirt_(stock_character)), two yellow shirts (command, helm), two blue shirts (medical, science) will accompany you to the planet. Occasionally the Redshirts might be shot dead with a phaser.. don't worry after a few moments a couple more will be beamed down (to await the same fate). Downward bling are a planetary green. Note that the bling fades in during the face start up sequence.
Chimes	ost_open.wav, ost_end.wav, ost_cc.wav, laser0.wav, ost_trn1.wav, ost_trn2.wav, ost_chss.wav, ost_gq3.wav, ost_myst.wav, ost_sus1.wav, ost_sus2.wav, ost_sus3.wav, ost_sus4.wav, ostupyr.wav, ost_cm.wav, ost_door.wav, ost_fas.wav, ost_intc.wav, ost_irrt.wav, ost_aye.wav, ost_c1.wav, ost_c2.wav, ost_c3.wav, ost_c4.wav, ost_c5.wav, ost_llp.wav, ost_log.wav, ost_nc.wav, ost_phas.wav, ost_pt.wav, ost_red.wav, ost_trb1.wav, ost trb2.wav,



Copyright Grahame Marsh/Nick Stock 2019

Face number	153
Nuggle file	*elvis.nug*
Description	Elvis (https://en.wikipedia.org/wiki/Elvis_Presley) themed clock (quarters and toll). Shows the time as hh:mm:ss.
Colons	Colons flash gold each second
Bling	None
Chimes	elvis00.wav, elvis15.wav, elvis30.wav, elvis45.wav, elvist.wav



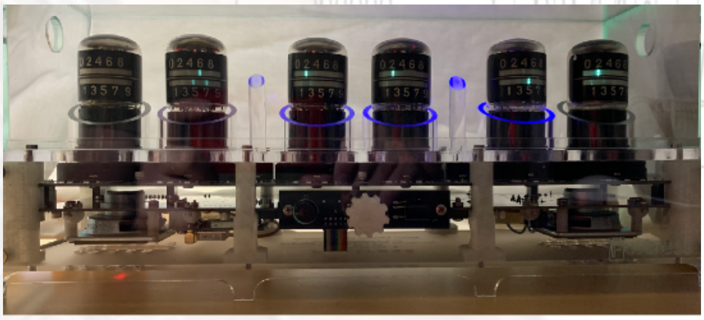
Face number	154
Nuggle file	*tng01.nug*
Description	Star Trek The Next Generation themed clock
Colons	Colons flash white each second
Bling	None
Chimes	tng_bell.wav, tng_2hrs.wav, tng_acc.wav, tng_aff.wav, tng_eng.wav, tng_hail.wav, tmg_holo.wav, tng_make.wav, tng_mess.wav, tng_over.wav, tng_stat.wav, tng_tran.wav, tng_work.wav

Copyright Grahame Marsh/Nick Stock 2019

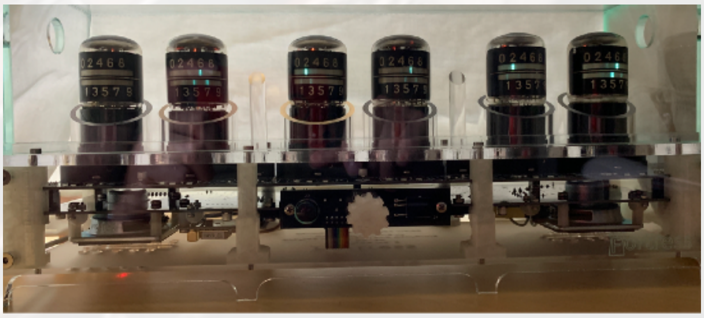
200 - Complex Faces

These faces are where all the previous themes and displays are brought together. So these faces combine use of the E1T tubes, complex bling patterns and colours, and chimes together into single faces. Perhaps not for the faint-hearted.

Face number	**200**
Nuggle file	*nick01.nug*
Description	Show the time using spinner (as with 105 time10.nug) with bling and chimes
Colons	None - used as bling
Bling	A fast rotation of blue bling from left to right across the top bling modules and then right to left along the downward modules.
Chimes	tintan00.wav, tintan15.wav, tintan30.wav, tintan45.wav, tintant.wav



Face number	**201**
Nuggle file	*nick02.nug*
Description	As with nick01.nug
Colons	None - used as bling
Bling	As with nick01.nug except the colour is a rotating hue
Chimes	palm00.wav, palm15.wav, palm30.wav, palm45.wav, palmt.wav



  Copyright Grahame Marsh/Nick Stock 2019

250 - Sequential Faces[47]

Face number	**250**
Nuggle file	*seq10.nug*
Description	This is the sequential face calling file. It displays nothing itself but runs a series of other Nuggle face files in sequence. It makes use of NVR variables to pass information to the called Nuggle programs. See Annex A for their allocation. Changes sequenced face every hour.
Colons	None
Bling	None
Chimes	None

Face number	**251**
Nuggle file	*seq10a.nug*
Description	As in 201 nick02.nug but arranged for sequential calling and return

Face number	**252**
Nuggle file	*seq10b.nug*
Description	As in 140 ship.nug but arranged for sequential calling and return

Face number	**253**
Nuggle file	*seq10c.nug*
Description	As in 200 nick01.nug but arranged for sequential calling and return

Face number	**254**
Nuggle file	*seq10d.nug*
Description	As in 152 ost01.nug but arranged for sequential calling and return

[47] Top of the food chain?

 Copyright Grahame Marsh/Nick Stock 2019

300 - Sun Information

The Fortress E1T clock firmware contains calculations of the time of sunrise and sunset for the geo-location set by the user in the *locale.ini* file[48]. The calculation is reasonably accurate but since it depends on some simplifications, the results are only displayed to hours and minutes. Displaying seconds would be misleading.

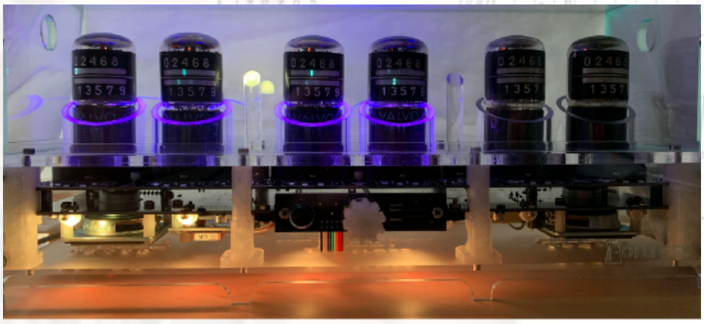
The firmware can also calculate the instantaneous position of the sun in the sky from the clock's position. This is given as the angle around the clock's horizontal plane - the azimuth and the angle from the horizon - the altitude. The azimuth can vary from 0 to 359 degrees and the altitude from -90 to +90 degrees; a negative altitude indicating the angle below the horizon.

Face number	**300**
Nuggle file	*sunpos.nug*
Description	Show the instantaneous position of the sun. The three left most tubes (tubes 0 - 2) show the azimuth in degrees. The right most tubes (4 - 5) show the altitude in degrees (see Colons below for the sign of the azimuth). Tube 3 is not used.
Colons	The right colon shows the sign of the altitude - red indicates a positive value and blue a negative value. The left colon is not used.
Bling	Azimuth tubes - green Altitude tubes - magenta Downwards - yellow
Chimes	300

[48] *locale.ini* is located in the */system* folder on the SD card

Copyright Grahame Marsh/Nick Stock 2019

Face number	301
Nuggle file	*sunrise.nug*
Description	Shows the sunrise today as hh:mm using tubes 0 to 3. Tubes 4 and 5 remain blank. There are conditions where the sun might not rise and so all the tubes will remain blank and the bling (see below) is used to indicate them.
Colons	The bling colours below will light the left colon. The right colon will remain off.
Bling	dark blue - the sun does not rise today, or yellow - the sun does not set today, or orange - downward bling - the sun does rise (the tubes will show the time) and purple/blue - tube bling
Chimes	None

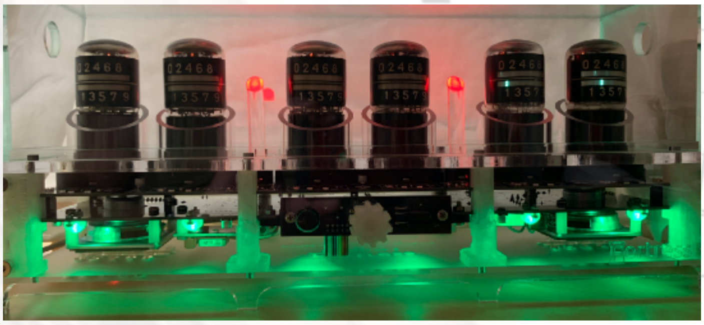


Face number	302
Nuggle file	*sunset.nug*
Description	Shows the sunset today as hh:mm using tubes 0 to 3. Tubes 4 and 5 remain blank. There are conditions where the sun might not set and so all the tubes will remain blank and the bling (see below) is used to indicate them.
Colons	The bling colours below will light the left colon. The right colon will remain off.
Bling	Dark blue - the sun does not rise today, or yellow - the sun does not set today, or blue - downward bling - the sun does rise (the tubes will show the time) and orange/purple - tube bling
Chimes	None

Copyright Grahame Marsh/Nick Stock 2019

350 - Hardware games

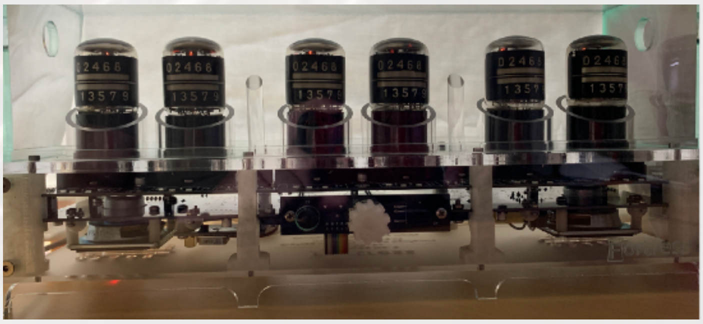
Face number	350
Nuggle file	*pir.nug*
Description	This is a game. You select the face away from the clock and then walk towards it. Each time the PIR sensor sees you it will increment the count on the tubes, flash the bling and make a noise...
Colons	Green - the PIR does not sense motion. Red - the PIR has sensed motion
Bling	Green - downward pointing bling
Chimes	uhoh.wav is played when the PIR senses motion



400 - Screen Savers

During the going to sleep process a screen saver state can be reached. Such savers are aimed at extending the life of the E1T tubes by preventing phosphor burns if the same number is shown for a long period of time. Such screen savers are no more than Nuggle programs and can be selected by a Nuggle program or by using the face key in the [sleep] section of *system.ini*.[49]

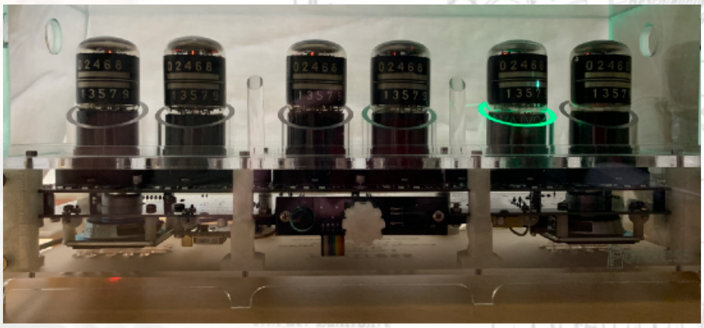
Face number	400
Nuggle file	*blank.nug*
Description	A screen saver Nuggle program - completely blank
Colons	None
Bling	None
Chimes	None



[49] *system.ini* is located in the root directory of the SD card

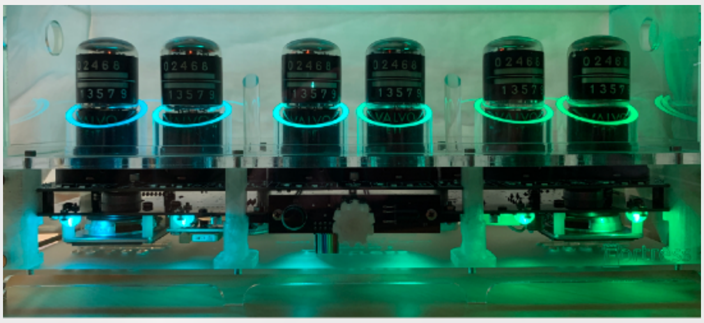
 Copyright Grahame Marsh/Nick Stock 2019

Face number	401
Nuggle file	*runner.nug*
Description	The leftmost tube counts 0 to 9 with the remain tubes blanked, then the next tubes counts 0 to 9 and so on until 9 is reached on the rightmost tube. The cycle then repeats.
Colons	None
Bling	A green flash follows the lit tube from left to right.
Chimes	None



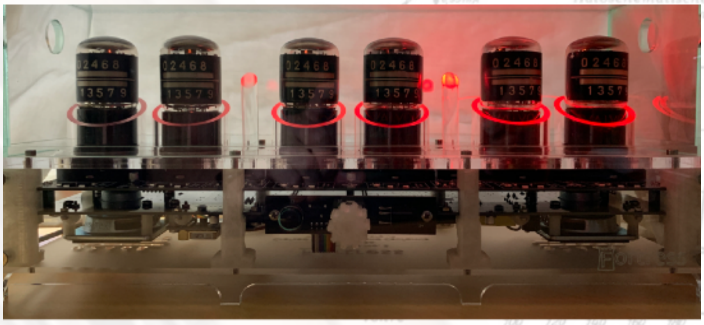
Face number	402
Nuggle file	*picker.nug*
Description	A random tube is selected and a random number is displayed. All the other tubes are off.
Colons	None
Bling	The random tube that has been picked to display a random number is given a random bling hue. The other tubes bling hues are unchanged. The downward bling is dim white.
Chimes	None

Face number	403
Nuggle file	*rainbow.nug*
Description	A random tube is selected and random number is displayed. All the other tubes are off.
Colons	None
Bling	Rotating rainbow of colours, colour progress round the colour wheel and right to left across the tube bling at a constant rate. The downward bling follow the tube bling above.
Chimes	None



Copyright Grahame Marsh/Nick Stock 2019

Face number	404
Nuggle file	*larson.nug*
Description	A bling Larson (https://en.wikipedia.org/wiki/Glen_A._Larson) scanner from a Cylon (https://en.wikipedia.org/wiki/Cylon_(Battlestar_Galactica)) or Kitt (https://en.wikipedia.org/wiki/KITT). The E1T tubes are not used and remain off.
Colons	Used as part of the bling cycle.
Bling	The bling is pulsed with a red fading light sweeping from side to side.
Chimes	None



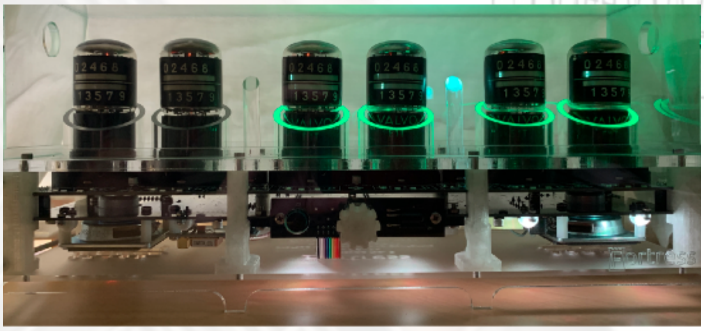
Face number	405
Nuggle file	*larsonr.nug*
Description	As face number 404 above but more colourful.
Colons	Used as part of the bling cycle.
Bling	As in face number 404 above except red is replaced by a colour hue that rotates round the colour wheel.
Chimes	None

Face number	406
Nuggle file	*wiggle.nug*
Description	Bling is lit sequentially lighting each LED individually and fading all to black.
Colons	Used as part of the bling cycle.
Bling	One half of a Larson scanner with the return stroke on the downward LEDs. Rotating colour wheel hues.
Chimes	None

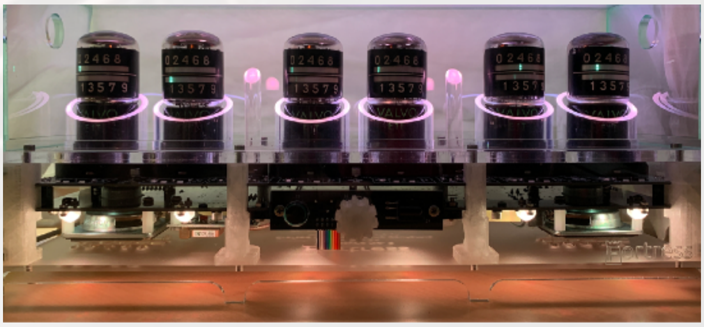
Copyright Grahame Marsh/Nick Stock 2019

These faces show various internal settings for the clock. They are not really intended as day-to-day displays but can be used to check a limited number of values without plugging in the USB console connection.

Face number	**900**
Nuggle file	*hardware.nug*
Description	Shows the detected hardware version using tubes 2 to 5 in the form MM.mm where MM is the major hardware version number and mm is the minor hardware version number. Tubes 0 and 1 remain blank.
Colons	The right colon is lit green between the major and minor hardware version numbers.
Bling	Bling on the used tubes and below the used tubes is lit green. The bling around and under unused tubes remains dark.
Chimes	None

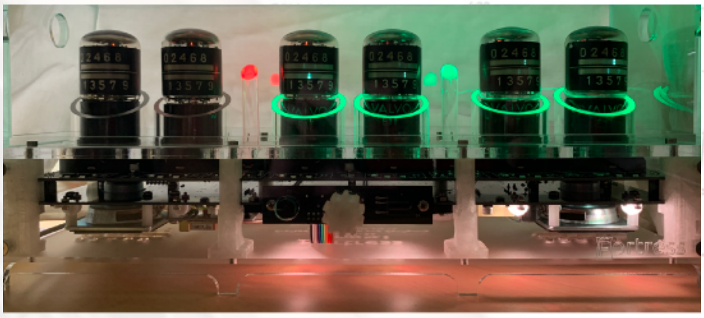


Face number	**901**
Nuggle file	*firmware.nug*
Description	Shows the firmware version in the form MM.mm.BB where MM is the major hardware version number, mm is the minor hardware version number, and BB is the build number.
Colons	The colons are lit to separate the version number elements.
Bling	A variety of light brown tones are used.
Chimes	None

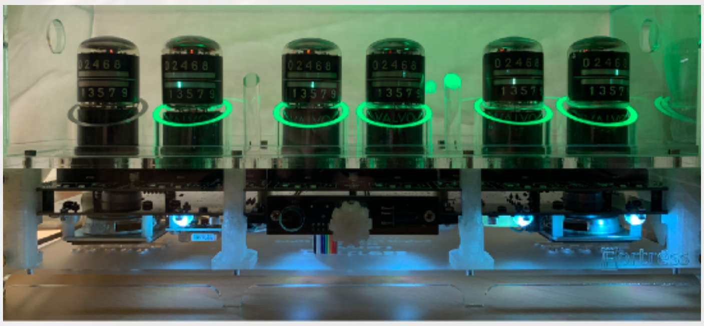


 Copyright Grahame Marsh/Nick Stock 2019

Face number	910
Nuggle file	*latitude.nug*
Description	Shows the clock latitude in the form DD.dd where DD is the latitude in degrees as a positive number (the sign will be shown using the left colon) and dd is the fraction. This value is read from *locale.ini*.
Colons	The right colon is lit green to separate the degrees and fraction degrees. The left colon is lit red to indicate a positive latitude, i.e. degrees north of the equator and is lit blue to indicate a negative latitude, i.e. degrees south of the equator.
Bling	Bling is lit around and under the active tubes.
Chimes	None

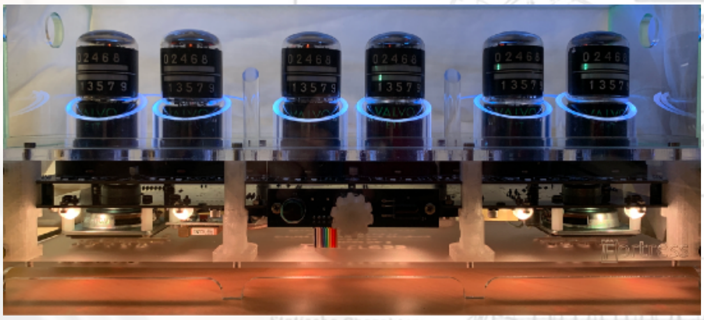


Face number	911
Nuggle file	*longtude.nug*
Description	Shows the clock longitude east of Greenwich in the form DDD.dd where DDD is the longitude in degrees always as a positive number and dd is the fraction. This value is read from locale.ini. Conventionally longitudes west of Greenwich are shown as negative numbers, in this case they are all shown as east of Greenwich and therefore vary 0 - 360 degrees.
Colons	The right colon is lit to separate the degrees and fraction degrees.
Bling	Bling is lit around and under the active tubes.
Chimes	None



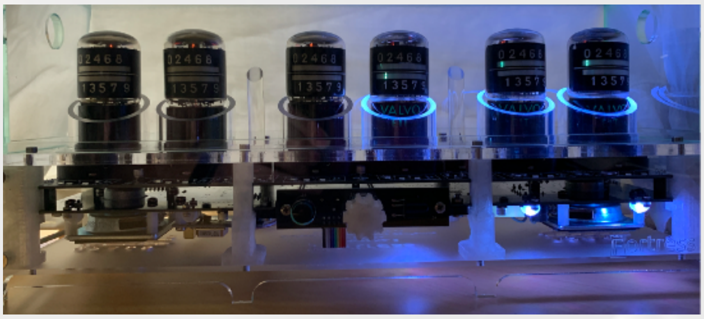
Copyright Grahame Marsh/Nick Stock 2019

Face number	912
Nuggle file	*elevatn.nug*
Description	Shows the clock elevation using the user preferred units (metres or feet). The value and preferred units are read from locale.ini.
Colons	None
Bling	Is lit around and under all tubes.
Chimes	None



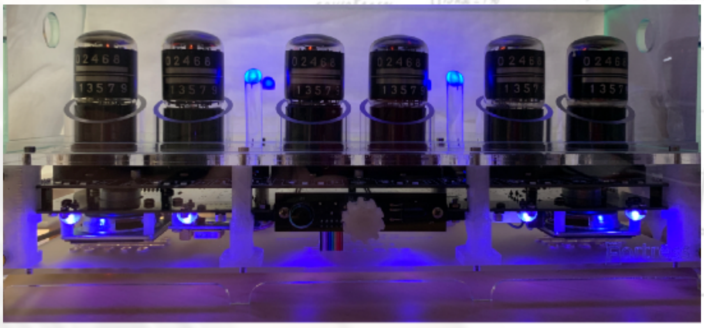
Face number	920
Nuggle file	*chiptemp.nug*
Description	Show the internal SAM3X8C chip temperature in the form TT where TT is the temperature in degrees Celsius. There is no fraction shown. Whilst there is no user preference setting for display units, the Nuggle program includes the necessary conversion to Fahrenheit which can be activated by changing the value of a constant.
Colons	None
Bling	Is lit around and under all tubes. The hue is temperature dependent.
Chimes	None

Face number	921
Nuggle file	*ldr.nug*
Description	Show the LDR brightness value 0 - 100 % using tubes 3 to 5. Tubes 0 to 2 are unused
Colons	None
Bling	Bling is lit around and under the active tubes.
Chimes	None

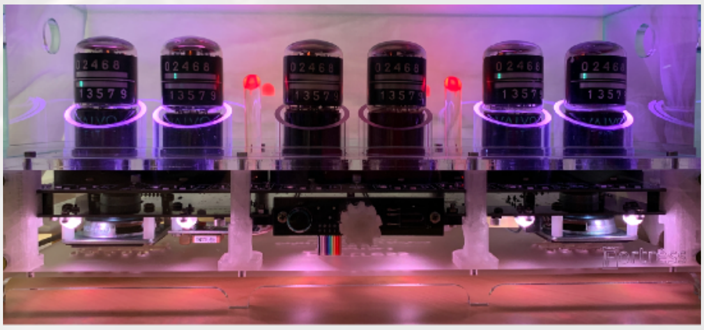


Copyright Grahame Marsh/Nick Stock 2019

Face number	**930**
Nuggle file	*wifi.nug*
Description	Shows the status of the WiFi connection using the bling and colons. The E1T tubes are not used.
Colons	No WiFi module installed - both colons are blue. WiFi installed - the left colon shows the WiFi connection status: red not connected, green connected. The right colon shows the SNTP status: red not invalid, green valid
Bling	No WiFi module installed - downward bling is blue. WiFi installed - the downward bling is light brown.
Chimes	None

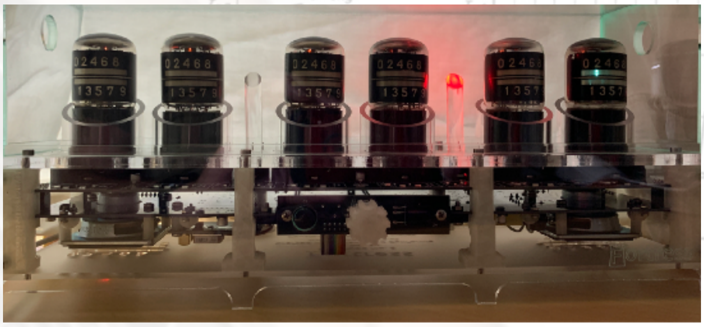


Face number	**940**
Nuggle file	*gps.nug*
Description	Shows the status of the GPS receiver using the bling, colons and tubes. With no GPS receiver installed the tubes all remain off. With the GPS receiver installed then tubes 0 and 1 show the number of satellites in view and tubes 4 and 5 show the number of satellites in use. Tubes 2 and 3 are unused.
Colons	No GPS module installed - both colons are blue. GPS installed - if the GPS data is valid both colons are green, if the data is invalid then the colons are red.
Bling	No GPS module installed - both colons are blue. GPS installed - the downward bling is light brown and the four used tubes are violet.
Chimes	none

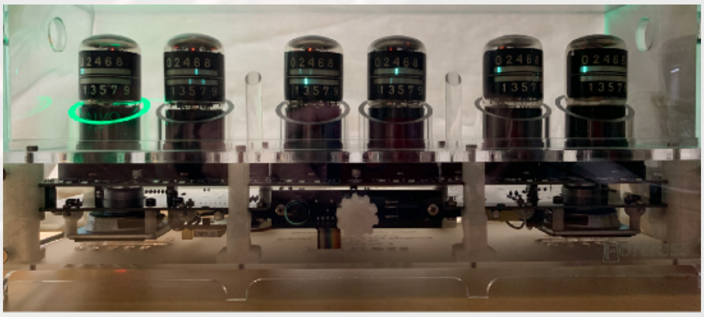


Copyright Grahame Marsh/Nick Stock 2019

Face number	950
Nuggle file	*paired.nug*
Description	Shows the IR Handset pairing status. If the pairing is disabled or forced to be enabled then the tubes are not used. If pairing is enabled then the pairing number is shown. The status is only shown when the face is first displayed. If you change the status then the face must be re-shown.
Colons	The right colon shows the pairing status: disabled shows blue, always forced to be enabled shows white, enabled and paired shows green, enabled but not paired shows red
Bling	None
Chimes	None



Face number	960
Nuggle file	*timeset.nug*
Description	Allow the RTC UTC time setting to be edited using the IR handset. A full description follows this table.
Colons	None
Bling	The tube position that can currently be changed is lit with green bling. All other tube bling remains dark.
Chimes	uhoh.wav, out.wave, notify.wav



Internally the clock works entirely in UTC[50] and not local time. Therefore, if you are setting the clock time, for example you do not have the WiFi or GPS modules to do this for you, then you must be prepared to enter the time in UTC by taking into account your time zone and daylight saving.

[50] https://en.wikipedia.org/wiki/Coordinated_Universal_Time

 Copyright Grahame Marsh/Nick Stock 2019

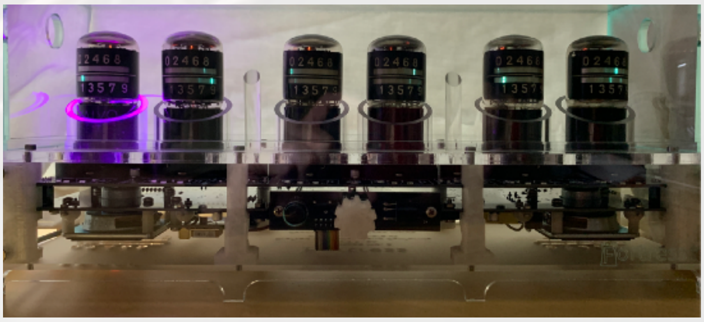
For example: You live in San Diego, CA. It is summer time. You want to set the clock to 12:35:00 local time. San Diego uses daylight saving so the local time without daylight saving is 11:35:00. San Diego is Pacific Time which is UTC-8 hours, therefore the current time in UTC is +8 hours so is 19:35:00 and this value should be entered into the clock. *Timeset.nug* takes over most of the IR handset keys. When the Nuggle program is run the tubes will display the current UTC hours and minutes from the RTC with the seconds set to zero. The bling around the leftmost tube, tens of hours, will light showing that tube is selected and its value can be changed. The IR handset keys are used as follows:

- '0' .. '9' key value is applied to the selected tube. The selected tube moves one to the right. The units seconds tube rolls over to the tens of hours tubes. Using these numerical keys you can dial in a new UTC time.
- '|<<' or Prev key will move the selected tube one position to the left. The leftmost tube rolls under to the rightmost tube.
- '>>|" or Next key will move the selected tube one position to the right. The rightmost tube rolls over to the leftmost tube.
- 'EQ' key accepts the UTC time as shown and writes the time to the RTC that will immediately start using it. The clock will then change to show the face that was shown before this face was selected.
- '>||' or Play key acts as a cancel or escape key, the time shown is abandoned and the clock will then change to show the face that was shown before this face was selected.

If an illegal entry is attempted, for example pressing '4' when the tens of hours tube is selected, then a warning chime will sound and the value shown on the tube will not change.

All keys not listed here will retain their default function, for example, pressing the '+' key will still increase the chime volume. Additionally, moving the rotary encoder will change the face selected and so effectively cancel the set time function.

Face number	**961**
Nuggle file	*dateset.nug*
Description	Allow the RTC UTC date setting to be edited using the IR handset. A full description follows this table.
Colons	None
Bling	The tube position that can currently be changed is lit with red bling. All other tube bling remains dark.
Chimes	uhoh.wav, out.wave, notify.wav



Internally the clock works entirely in UTC and not the local date. Therefore, if you are setting the clock date, for example you do not have the WiFi or GPS modules to do this for you, then you must be prepared to enter the date in UTC by taking into account your time zone and daylight savings.

Copyright Grahame Marsh/Nick Stock 2019

Only the two least significant digits of the year can be changed. The most significant digits are always 20. Thus the date can only be changed within the range 2000 to 2099.[51]

Datetime.nug takes over most of the IR handset keys. When the Nuggle program is run the tubes will display the current UTC day, month and year from the RTC. The bling around the leftmost tube, tens of years, will light showing that tube is selected and its value can be changed. The IR handset keys are used as follows:

- '0' .. '9' key value is applied to the selected tube. The selected tube moves one to the right. The units seconds tube rolls over to the tens of years tubes. Using these numerical keys you can dial in a new UTC date.
- '|<<' or Prev key will move the selected tube one position to the left. The leftmost tube rolls under to the rightmost tube.
- '>>|" or Next key will move the selected tube one position to the right. The rightmost tube rolls over to the leftmost tube.
- 'EQ' key accepts the UTC date as shown and writes it to the date to the RTC that will immediately start using it. The clock will then change to show the face that was shown before this face was selected.
- '>||' or Play key acts as a cancel or escape key, the date shown is abandoned and the clock will then change to show the face that was shown before this face was selected.

If an illegal entry is tried, for example pressing '4' when the tens of months tube is selected, then a warning chime will sound and the value shown on the tube will not change.

All keys not listed here will retain their default function, for example, pressing the '+' key will still increase the chime volume. Additionally, moving the rotary encoder will change the face selected and so effectively cancel the set date function.

Face number	**962**
Nuggle file	*syncrtc.nug*
Description	Force the RTC to sync with the GPS or WiFi (which ever installed). The clock will revert to the predecessor face.
Colons	Not used
Bling	The tube bling LEDs will flash green briefly on a successful synchronisation.
Chimes	None

[51] Outside that range a constant within the C firmware must be changed and the SAM burned. In practice, a date range up to the year 2099 seems adequate…;-)

Copyright Grahame Marsh/Nick Stock 2019

Face number	999
Nuggle file	*reset.nug*
Description	After a count down period of ten seconds the clock will attempt to hard reset. The E1T tubes show the count down. The bling shows the last six seconds and the chimes the last four seconds.
Colons	Not used
Bling	Initially the tube bling is all orange. It goes black one tube at a time during the last six seconds.
Chimes	bark.wav

Copyright Grahame Marsh/Nick Stock 2019

1000 - Nuggle Tests

These test aspects of the Nuggle firmware in C. It is expected that these are run using the console as many send reports. These tests are simple and are not considered comprehensive or stressful.

Face Number	Filename	Description
1000	*t_hello.nug*	Hello World! using WriteLn
1001	*t_if.nug*	Tests If, Else, IfEnd, Succ(), Pred(), Incl, Decl
1002	*t_event.nug*	All Event_??? sections. Default, NoDefault procedures
1003	*t_for.nug*	Tests For, To, DownTo, ForEnd, Break and Continue
1004	*t_call.nug*	Tests Call and Return
1005	*t_repeat.nug*	Tests Repeat, Until, Break and Continue
1006	*t_while.nug*	Tests While, WhileEnd, Break and Continue
1007	*t_loop.nug*	Tests Loop, LoopEnd, Break and Continue
1008	*t_beep.nug*	Beep procedure; Random function
1009	*t_face.nug*	Face procedure; use of NVR and Push (simple stack) to transfer data to the new face (t_dest.nug)
1010	*t_dest.nug*	This is the destination face for t_face.nug so is not run directly
1011	*t_func.nug*	Tests Abs(), Sqrt(), Max(), Min(), Odd(), Even(), Encode(), High(), MidHigh(), MidLow(), Low(), LoCase(), UpCase(), Random(), VerifyDate(), VerifyTime(), GPIO(), NVR(), FacePred(), FaceSucc(), IndexOf()
1012	*t_stack.nug*	Tests Empty(), Pop(), Peek(), Push and Flush. Also can be used to test stack data permanence.
1013	*t_timer.nug*	Tests Timer and Timer()
1014	*t_jump.nug*	Tests Jump and labels
1015	*t_exp.nug*	Expression evaluation - all operators, use of parentheses, limited precedence checking
1016	*t_tz.nug*	Tests TimeZone changing clock mode, DST rule, and time zones to various locations around the world.
1017	*t_reset.nug*	Reset procedure. To prevent accidentally resetting the clock the text will need to be edited before use
1018	*t_bling.nug*	Colour, Transmit, GradientRGB, GradientHSV, Rainbow, Fade procedures; FormHSV function
1019	*t_colour.nug*	Colour function and procedure; FormRGB, FormHSV, RGB2HSV, HSV2RGB functions; Clear procedure

Many procedures, declaration elements etc. are used in many of the Nuggle test programs but these commonly used elements are not documented above.

 Copyright Grahame Marsh/Nick Stock 2019

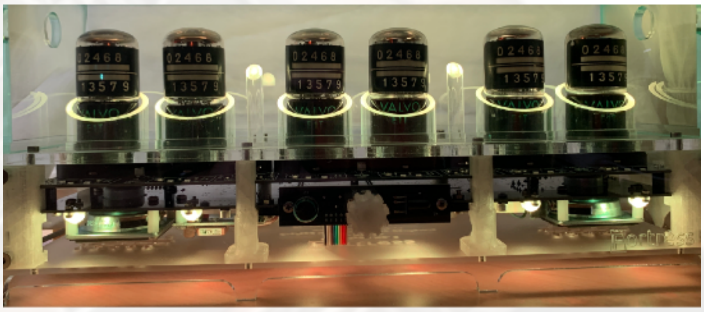
2000 - Nuggle Cookbook Examples

The **Nuggle Cookbook** is in preparation at this time so this section is far from complete.[52]

2100 - Illustrating Sequentially Addressed Faces

Clock faces can be displayed for a period of time and then replaced with an alternative. A chain of faces can be built and displayed sequentially. Therefore you might show one type of bling for a few hours before changing to a second. The Nuggle code presented here as an illustration is aimed at making changes over a matter of minutes, hours or even days. If a change is required in a period less than one minute then a single Nuggle file is recommended (see face number 120 which shows the date for a ten second period during each minute).

Face number	**2100**
Nuggle file	*seq01.nug*
Description	This is the sequential face calling file. It displays nothing itself but runs a series of other Nuggle face files in sequence. It makes use of NVR variable to pass information to the called Nuggle programs. See Annex A for their allocation. Changes sequenced face every 10 seconds.
Colons	none
Bling	none
Chimes	none



Face number	**2101 - 2104**
Nuggle file	*seq01a.nug, seq01b.nug, seq01c.nug, seq01d.nug,*
Description	Four Nuggle files called in sequence repeatedly. The only difference between the files is the colour of the bling. These files are to illustrate a sequential technique and not offer a mind blowing clock experience...
Colons	None
Bling	Colour changes between the files
Chimes	None

52 Still…

 Copyright Grahame Marsh/Nick Stock 2019

Annex A - NVR Variable Usage

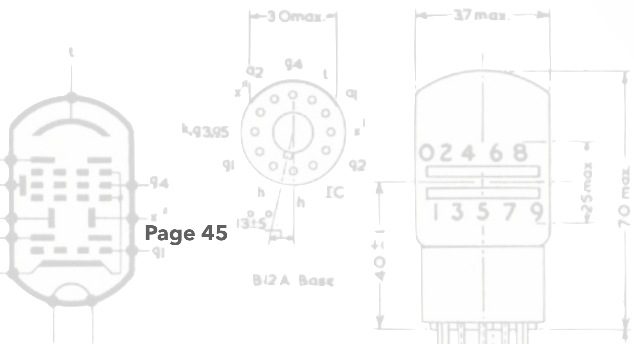
Values can be stored in NVR to pass information between Nuggle programs and to preserve data when the clock is reset or restarted.

NVR Index	Use
0	Face counter, incremented each time a new face is shown by the sequential counter
1	Sequential calling Nuggle program to return to
2	Period in seconds to show the sequentially called file for
10	Used in t_face.nug to test the Nuggle **Face** procedure

Copyright Grahame Marsh/Nick Stock 2019

Fortress E1T Operations Guide

Everything you wanted to know but daren't ask….



 Copyright Grahame Marsh/Nick Stock 2019

1. Introduction

This section describes most of the technical aspects of the Fortress E1T 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 SAM3X8C micro-controller
- 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

Standalone 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.[53]

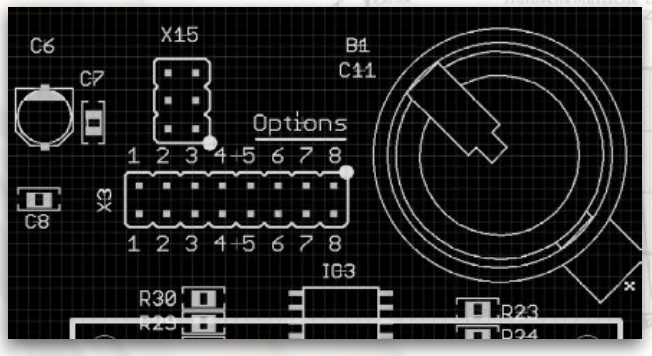
[53] Two separate Dropbox folders exist - One for the Fortress here and another covering the common modules used here.

 Copyright Grahame Marsh/Nick Stock 2019

2. Main Board Option Jumpers

There are eight jumpers on the main board (component X3 near to the backup battery/super-cap) which can select operating options that need to be decided as the clock is initially starting up. Jumpers **must** only be fitted or removed when the clock is switched off.

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



1. 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 using the console menus without interference from the high level clock automation.

2. Plug-in module pass-through mode - if the clock is in "test mode" then all of the clock's normal operation is 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 enable 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. If the clock is not in "test mode" then the setting of this jumper is ignored.

3. Permit automatic PSU start up - the 6.3V heater PSU and 300V flyback HT PSU will normally start up automatically when required. Fitting this jumper allows these PSUs to start automatically. This option is primarily for safety. The two PSUs can still be started manually using the console. See the OUTS command.

4. Not currently used.

5. Not currently used.

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.

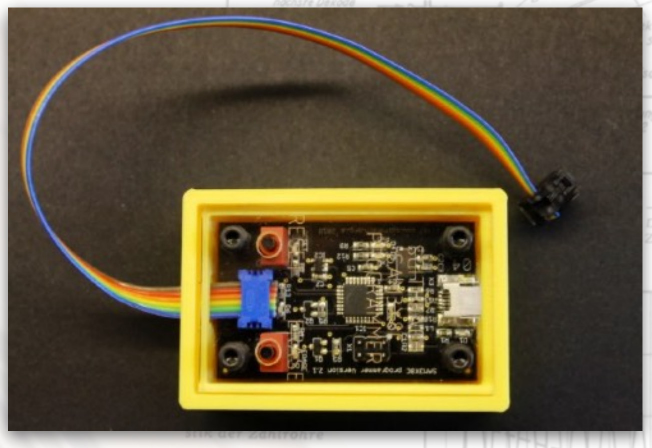
7. Forced audio mute - fitting this jumper will force the clock to be entirely quiet. It will not beep or play chimes. The setting cannot be overridden using the IR handset, the rotary encoder or using the console commands. The clock will remain entirely quiet!

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.

The option jumper positions are normally only read only once when the clock is started up but see the **OPTS** and **OPTU** menu commands.

 Copyright Grahame Marsh/Nick Stock 2019

3. Console Connection

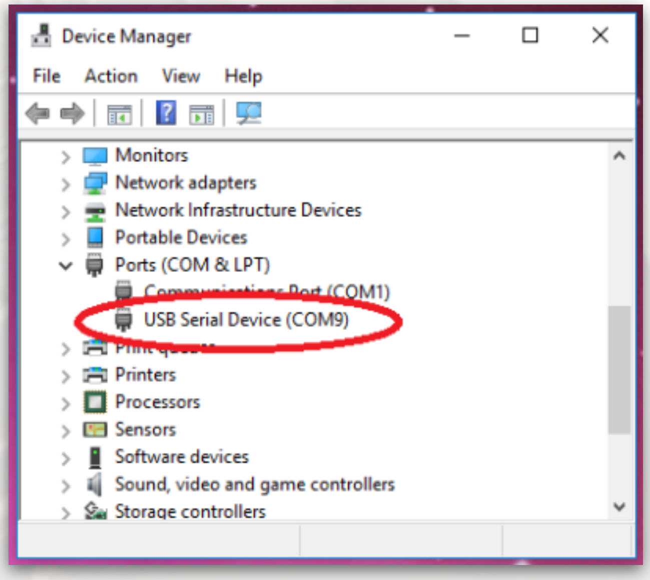


The console is a simple text-based user interface that allows many clock settings to be adjusted and tested. It is an imported debug tool when writing Nuggle programs. The console hardware is also the SAM3X8C programmer.

The SAM3X8C programmer

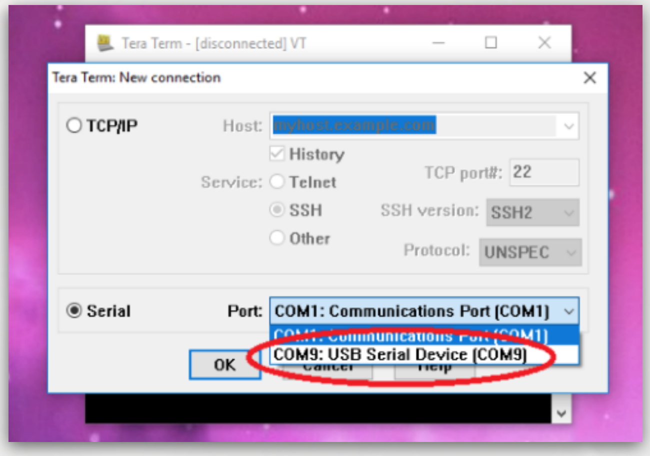
The programmer is plugged into the Fortress E1T clock processor plug-in board using a 6-way ribbon cable and IDC connector. Plug the USB mini connector into a computer USB port using a suitable cable. The tests here that rely on the computer assume you are using a PC with Windows 10 installed. I cannot 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.



Device Manager Showing the USB Serial Device (COM9)

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.



Selecting COM9 Serial Port in Tera Term

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.

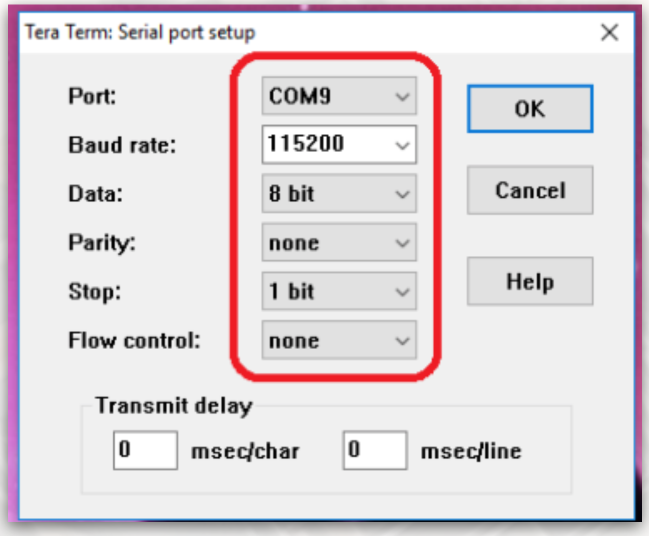
Run a serial console program - I often use Tera Term[54]. Set the serial port to the COM number you noted in step 1 above. Configure the serial

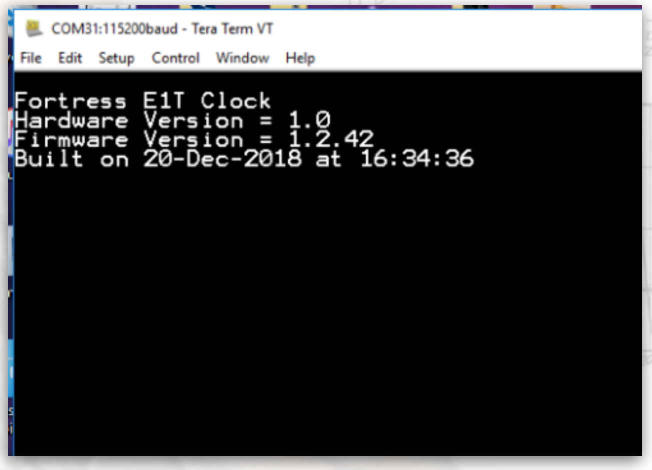
[54] http://ttssh2.osdn.jp/

 Copyright Grahame Marsh/Nick Stock 2019

console to something like 115200, 8, N, 1 with no flow control.

When power is applied to the Fortress E1T clock and the reset button on the clock pressed briefly you should see a simple welcome message[55]. Something like:





Setting up the Serial Port in Tera Term

[55] Or a much longer one if the "verbose" jumper 8 is fitted.

Copyright Grahame Marsh/Nick Stock 2019

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
CHM	CHiMe system
CON	CONstants interrogation
DAT	clock DATe
E1T	E1T tube control
EXP	EXPression 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
STR	Storage information
SYS	SYStem information
TIM	clock TIMe
VAR	system VARiable value interrogation
VOL	left and right amplifier VOLume controls
WIF	WiFi controls

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

 Copyright Grahame Marsh/Nick Stock 2019

For example:

DATG “get” the current date
VERR “report” hardware/firmware version and build information
SYSU "use" the system - a parameter sets how
LDR? “help”, LDR help information
LEDK “kill”, turn all bling LEDs off

Some commands have mandatory parameters, for example:

TIMS 12:00:00 “set” the TIMe to midday
OUTS 6,1 “set” digital OUTput for the 6.3V heater (6) supply on (1)
OUTS 300,0 “set” digital OUTput for the 300V HT (300) 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 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 Fortress E1T 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 Teaser)[56] using a two channel PWM generator. It is used for warnings, making irritating noises when the rotary encoder is moved or a key is pressed on the IR handset and can be used to play simple tunes.

[56] https://hitchhikers.fandom.com/wiki/Teaser

 Copyright Grahame Marsh/Nick Stock 2019

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

See also the VOLume commands.

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 name	Play the chime by giving its name, quotes and file extension are not used example: **chmu strk12**

See also the VOLume commands.

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

Copyright Grahame Marsh/Nick Stock 2019

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.

E1T display tests

The six E1T tubes can be addressed.

E1T?	E1T command help message
E1TC n	Continuously spin tube n. Press *Esc* key to stop
E1TG n	Get the position of tube n where the leftmost tube is 0 and the rightmost is 5
E1TK	Switch all E1T to the off position
E1TR	Report all E1T positions
E1TS n, p	Set the position of tube n to position p n = leftmost tube is 0 and the rightmost is 5 p = 0 .. 9 the tube glows in that position, or p = 10 the tube is turned off

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
EXPG	Enter an expression for evaluation example: **expg red >> 10** returns: decimal = 16320 hex = 0x3fc0 example: **expg red > blue** returns: decimal = 1 hex = 0x1 Since numerically constant red has a larger value than the constant blue so the value 1, boolean true, is returned

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

Copyright Grahame Marsh/Nick Stock 2019

FACe Command Help Message

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

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 n	GPS report where n = 0 - report data validity, geolocation etc. n = 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
LDRR	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.

Copyright Grahame Marsh/Nick Stock 2019

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. See also the **TheMe** commands

LED?	LED bling-bling help message
LEDA fxd	set the bling brightness setting to static level fxd = 1 .. 31 static level not > 10 is recommended
LEDB min,max	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 0, grp, rgb	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 1, num, rgb	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 0, grp, hsv	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 1, num, hsv	Set a single LED to the specified HSV colour

Copyright Grahame Marsh/Nick Stock 2019

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 .. 8
OPTR	Report position of all jumpers
OPTS n, v	Set the position of option jumper n to v n = 1 .. 8 v = 0 | 1 this is a firmware setting that overrides the hardware/physical jumpers
OPTU	Restore the firmware settings to the hardware/physical positions

Copyright Grahame Marsh/Nick Stock 2019

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 = 2 - pin 2 on header X1 n = 3 - pin 3 on header X1 n = 6 - 6.3V heater enable n = 300 - Flyback 300V enable
OUTR	Report all settings
OUTS n, v	Set the digital output n to value v n = as above v = 0 | 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

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

SToRage reports

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

STR?	StoRage reports help message
STRR n	Storage report n n = 0 - file system n = 1 - SD Card n = 2 - NVRAM n = 3 - memory - see SYSR 0

Copyright Grahame Marsh/Nick Stock 2019

SYStem reports and actions

A variety of system reports and actions to invoke.

SYS?	SYStem help message
SYSR n	System report *n* n = 0 - memory use n = 1 - SAM3X8C chip identification n = 2 - SAM3X8C core temperature °C n = 3 - task system status n = 4 - demonstrate floating point errors n = 5 - demonstrate random number generation n = 6 - version and build information
SYSU n	System actions: n = 0 - invoke sleep n = 1 - wake up n = 2 - shut down the clock (requires reset) n = 3 - reset the clock n = 4 - cancel any **C**ontinuous commands (same as pressing *esc*) n = 63 - fatal error display test (requires reset)

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 hh:mm:ss	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.

Copyright Grahame Marsh/Nick Stock 2019

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 name	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 c, v	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	Send an *AT* command to the WiFi module.
WIFR	Report on current WiFi settings.
WIFU	Reset the WiFi module.

Copyright Grahame Marsh/Nick Stock 2019

5. Programming the 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 BOSSA[57] 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 using[58] and xxxxx.bin is the binary image of the firmware you want to program into the flash.

example:

d:\bossa\samprog COM9 d:\fortress\firmware\fortress.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 Fortress E1T clock and wait a few moments and then power the clock again. The "cold boot" is a requirement of the SAM3X8C.

The new firmware should now run.

[57] https://github.com/shumatech/BOSSA
[58] using the method in section 3 above

 Copyright Grahame Marsh/Nick Stock 2019

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"[59] 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 Fortress Clock).
constant.ini	Constant name to value look up table contents. The user can add additional values for use in their own Nuggle programs.

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

Copyright Grahame Marsh/Nick Stock 2019

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.

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.[60]

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

Copyright Grahame Marsh/Nick Stock 2019

7. Adding or changing Chime WAV files

Introduction

The Fortress E1T Clock uses WAV files to sound chimes and make other noises. The clock hardware uses the two 12 bit Digital to Analogue Convertor (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 format[61]
- 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 Audacity[62] software. Using this software you can export the file as a WAV format readable by the Fortress E1T Clock. 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 Fortress E1T 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

61 https://en.wikipedia.org/wiki/WAV
62 https://www.audacityteam.org/

Copyright Grahame Marsh/Nick Stock 2019

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 Fortress E1T Clock resets itself and the clock restarts.

N.B. The file pre-processor is quiet 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 *name*	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 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. The copy the saved copy back again. I do not recommend using a defrag tool on the SD card.

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 give 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.

 Copyright Grahame Marsh/Nick Stock 2019

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 Fortress E1T 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**[63]. 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 Fortress E1T Clock (see Page 110) and a Nuggle cookbook.[64]

[63] Could it be anything else?
[64] In preparation….still…

 Copyright Grahame Marsh/Nick Stock 2019

9. Compiling the Sgitheach SAM3X8C Clock Firmware for the Fortress E1T Clock

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 *Fortress 1.0.2.zip*[65] 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 “Fortress 1.0.2”. Select this file and set the name to “Fortress01” 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 FORTRESS_E1T_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:

  - Compiler General - untick the default char type and default bit field type checkboxes if ticked

[65] Or the latest and greatest on the Dropbox…

Copyright Grahame Marsh/Nick Stock 2019

- Compiler Optimisation - change the level to **Optimise most (-O3)**
- 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
- Complier 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 : 298632 bytes 57.0 % Full
Data Memory Usage : 23600 bytes 24.0 % Full

You have now compiled a version identical to the Fortress.1.0.2.bin file on the Dropbox. The source code is now yours to modify[66]!

66 If you dare! :-)

Copyright Grahame Marsh/Nick Stock 2019

10. InfraRed Remote Control

The IR handset is used to remotely and conveniently change the way the Fortress E1T 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 Fortress 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

 Copyright Grahame Marsh/Nick Stock 2019

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.

Copyright Grahame Marsh/Nick Stock 2019

11. Troubleshooting Guide

Introduction

Given the complexity of the Fortress E1T Clock it is difficult to give a completely comprehensive guide to the faults that might occur and how to find them. I would expect 99.9% of first time faults to be soldering problems, either a poor solder joint or a bridge between adjacent conductors.

Main Board testing

For main board troubleshooting it is easiest if the main board is not plugged into the display board and is laid with the components uppermost.

12V, 5V and 3.3V rails

The initial check must be to ensure that the 12V, 5V and 3.3V power supplies are all correct.

SAM3X8C plug in module

Once it is asserted that the 5V power supply is OK then the SAM3X8C plug in module can be tested. The first stage is to plug in the SAM3X8C programmer and use it as a serial console to check that a start up script is produced and the menu system is operating. The process is described in *Section 3*.

Given that the SAM3X8C plug in module has been tested before dispatch then, any problem found must be due to the SAM3X8C programmer. The build instructions contain detailed test processes that should have demonstrated that the programmer is working correctly.

Remember that accidentally[67] pressing the "Erase" button on the programmer when connected and powered will erase the flash memory in the SAM3X8C. The chip will then require flashing again.

Main board in more detail

Once the console is operating and you can issue menu commands, you can move on to testing individual sections on the main board. To accomplish this it is probably best if you use the option jumpers to set the clock into verbose mode (fit jumper 8) and into test mode (fit jumper 1). For safety you can omit jumper 3 so that the +300V HT flyback and 6.3V heater power supply will not start.

To assist this process there are four sections of this manual which will help you:

- *Section 2* of details the use of the option jumpers.
- *Section 4* list all of the menu commands.
- The *Main Board Assembly* section gives an example of the start up script sent to the console. This script may contain error reports.
- The *How It Works* is a breakdown of the schematics by function.

[67] …or deliberately! ;-)

 Copyright Grahame Marsh/Nick Stock 2019

Additionally, the project Eagle[68] files are contained on the project Dropbox. These files can be opened using the free Eagle version.

The main board will operate happily without the display board plugged in. Therefore, you can detach the display board and flip the main board over so you have easy access to its component side.

Obviously, given the number of subsystems on the main board it is not possible to go any further with advice on trouble shooting a problem on the main board. However, if you have a problem then please feel free to contact us and we will try to help. As a last resort you can return the board (to Scotland) and it will be fixed. Depending on the nature of the fault there may be a fee and maybe a return postage cost.

Display Board testing

Because the E1T tubes and display board drivers for them are slightly unusual, a more detailed explanation is given here on what you might expect to find.

When probing points on the board, take extreme care that you do not accidentally short adjacent connections. There are a wide range of digital and power traces on the board. Shorting digital to power will almost certainly damage the SAM3X8C and possibly other components. Shorting power to ground will most likely damage the corresponding convertor or regulator.

Due to component tolerances the 300V, 6.3V and the four regulated voltages on the display voltages will vary. These are what I measured on the prototype clock:

Desired Voltage	Measured Voltage
300V	300.5V
6.3V	6.23V
156V	158.1V
30V	30.6V
15V	16.2V
12V	13.1V

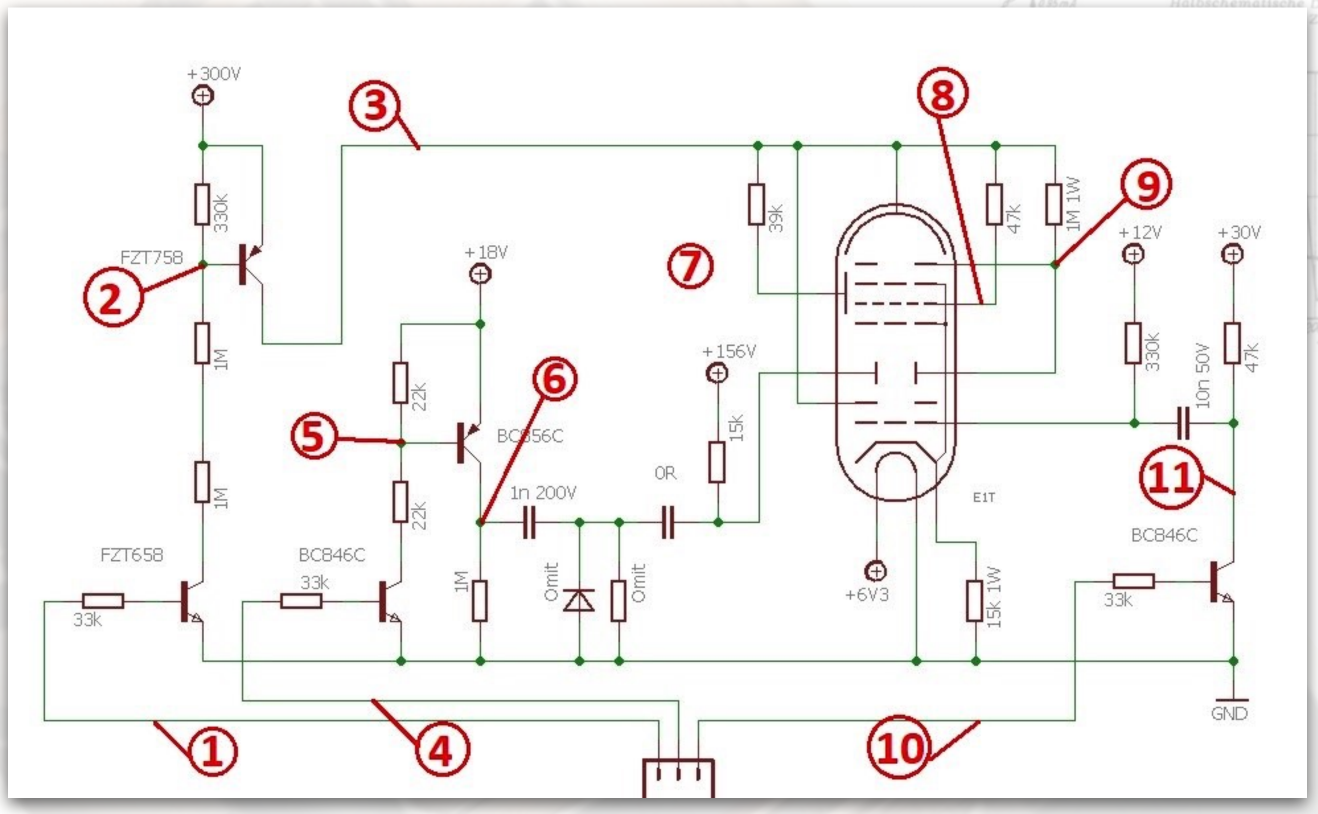
I had no problems with the variability you see here. Widely differing voltages from those listed indicates a problem of course.

[68] https://www.autodesk.com/products/eagle/free-download

 Copyright Grahame Marsh/Nick Stock 2019

E1T Static Voltages

These voltages were measured[69] with no Nuggle program running and an E1T glowing in position '0'. Under these conditions the HT switch transistors are conducting, the step pulse transistor is off and the reset transistor is conducting.



Point	Reference	Measured
1	Ground	3.28V
2	+300V	570mV
3	+300V	50mV
4	Ground	7mV
5	Ground	16.2V
6	Ground	0V
7	+300V	300.5V
8	Ground	260V
9	Ground	83V
10	Ground	3.28V
11	Ground	15mV

1. It is very difficult to measure the static voltage on point 5 without pulsing the tube so make sure you reset the tube to the '0' position when measuring other voltages.
2. It is very difficult to measure the voltage on point 9 without the tube stepping to position '9'

[69] See the NUGK and E1TS console commands

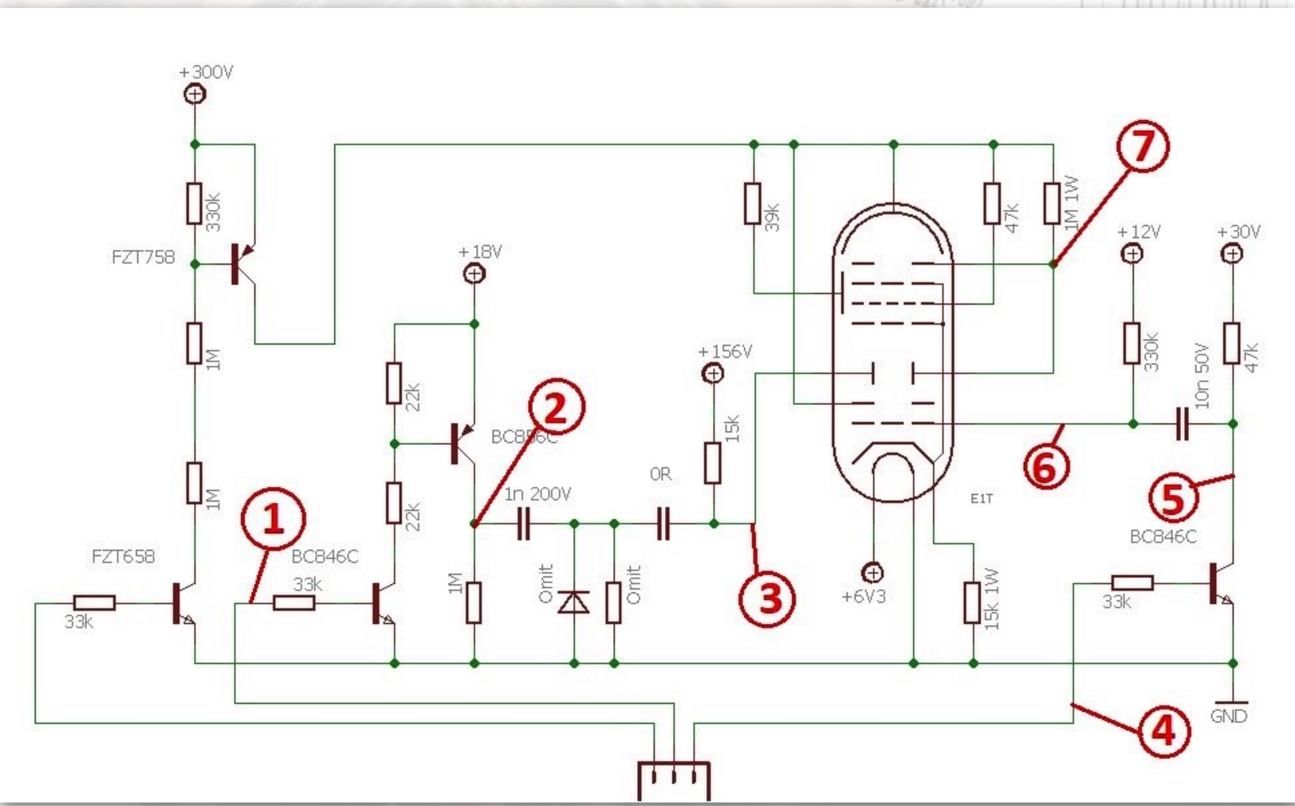
Copyright Grahame Marsh/Nick Stock 2019

E1T Circuit Waveforms

The following oscilloscope images were captured using a Rigol DS1052E. Not a particularly high specification oscilloscope but good enough for this task. An ordinary x1 probe was used with DC coupling unless otherwise stated. In all cases the E1T was spinning at 100mS between step pulses using the console **E1TC** command.

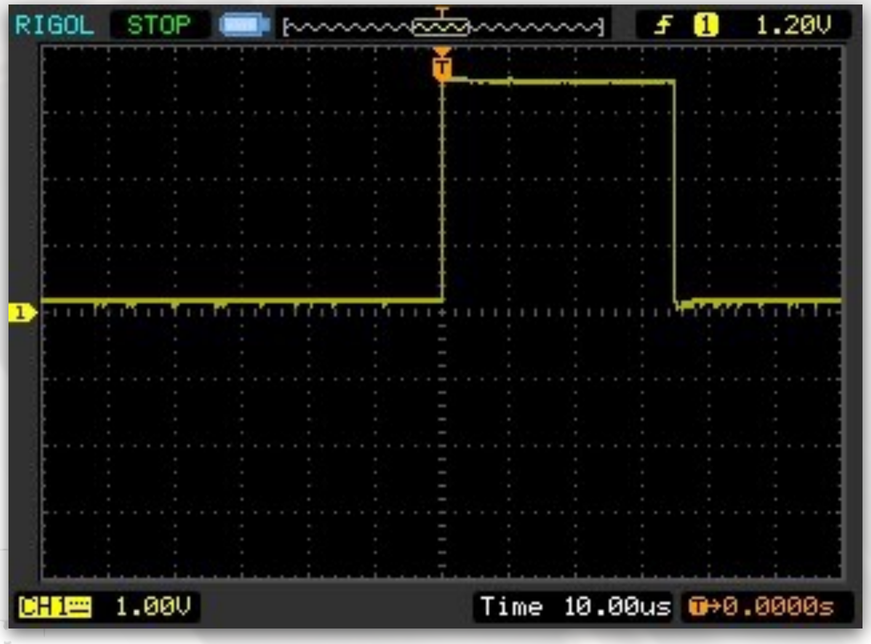
<u>Schematic</u>

The following waveforms were captured. The step pulse shaper used is by Ron Dekker.



1. *Step pulse input.*

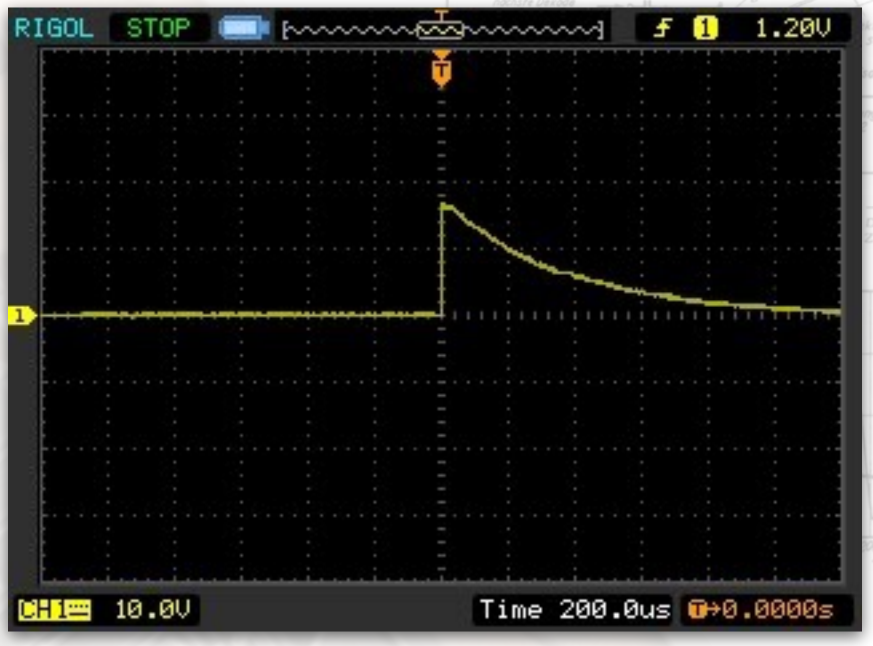
This is a connection from the SAM3X8C so shows a pulse 3.3V high and 33µS long.



Copyright Grahame Marsh/Nick Stock 2019

2. Step pulse after the drive transistors.

This is the voltage on the collector of the PNP step driver transistor. It is about 18V in amplitude and has about a 1mS decay period.



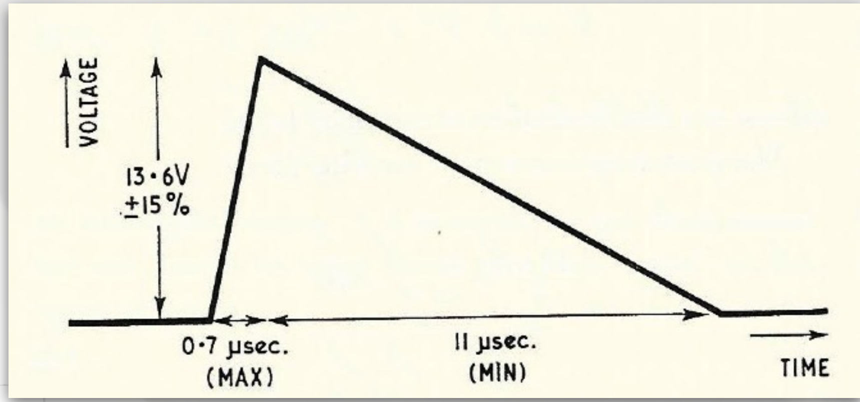
3. Step deflection pulse.

A x100 probe was used with DC coupling.

This is the pulse applied to the deflection plate x' in the E1T[70]. The pulse rises quickly to 15.2V and then decays over a period of about 20 µS.

The ideal waveform.[71]

The measured waveform meets the requirements for the maximum rise time (< 0.7µS), the minimum decay time (> 11µS) and the pulse amplitude (actually 12% above 13.6V).



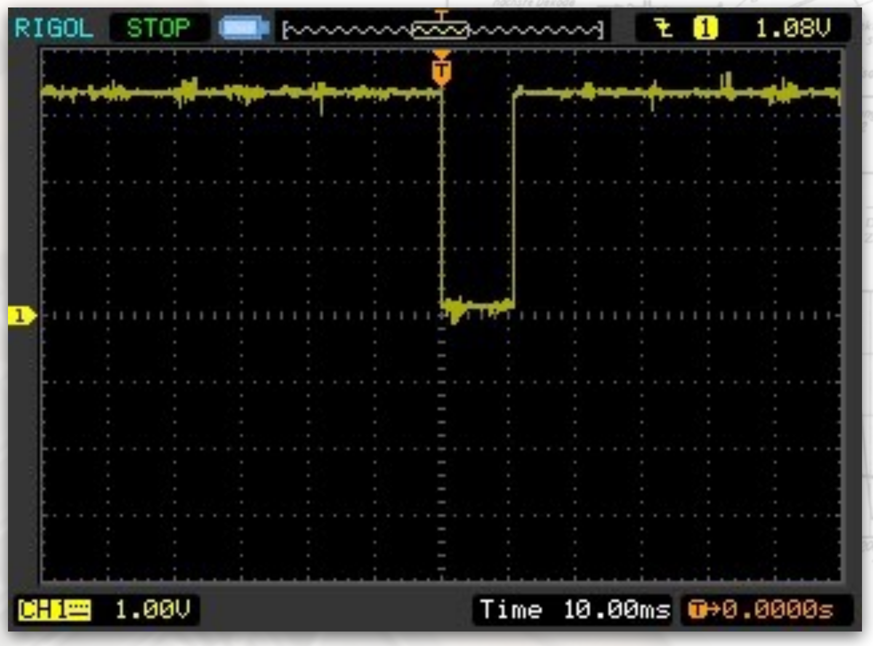
[70] J. B. Dance - E1T Decade Counting Circuits

[71] Dance - Fig 5.9

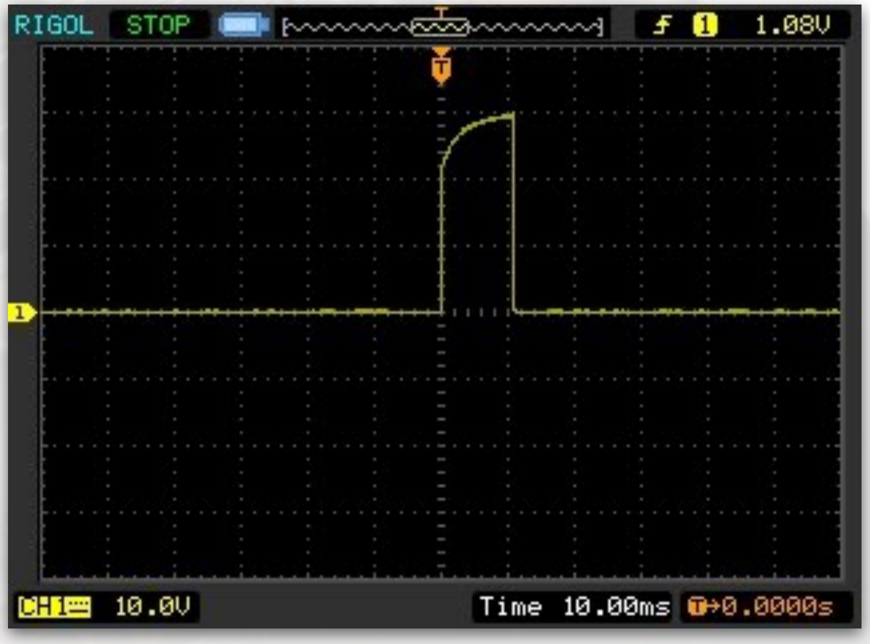
Copyright Grahame Marsh/Nick Stock 2019

4. Reset pulse.

This is a connection from the SAM3X8C, a pulse 3.3V low and 10mS long.



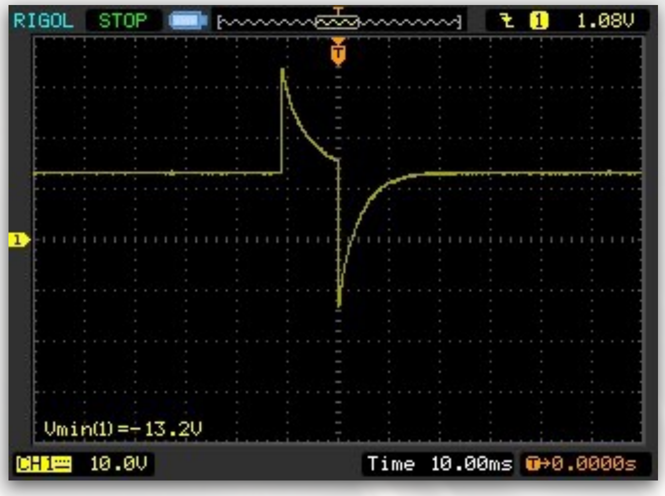
5. Reset pulse transistor collector.



6. E1T Grid.

This is the voltage on the E1T grid. The strong negative pulse to -13V cuts off the electron flow through the tube causing it to reset to position 0 when the current flow is re-established.

Normally the cathode potential is about 15V above ground and the grid is at 12V above ground. The -13V pulse means that the grid falls to about -28V with respect to the cathode. The tube requires -24V for cut off.



  Copyright Grahame Marsh/Nick Stock 2019

7. E1T Deflection Plate x'' and anode 2.

A x100 probe was used with DC coupling. In this image the cyan trace (channel 2) shows the position of the ground (0V).

A few observations:

The high pulse at the start of the cycle (below the T symbol) reaches nearly 300V or the +HT supply. This occurs when the negative pulse to the grid has cut off the electron flow. This will mean that when the tube is turned on again at the end of the reset pulse the beam will be in position '0' as required.

The voltage of the '0' step is about 228V and the lowest, just after '9', is 94V. Therefore the voltage per step is (228 - 94)/10 = 13.4V. This just about corresponds to the step voltage of 13.6V for counting.

Understanding E1T "Faults"

I call these "faults" but they are part of the charm of the E1T. From the Fortress clock point of view they are not faults of the Fortress clock but faults in the E1T tubes themselves. Before blaming the Fortress clock you might take some simple steps of swapping the tubes around to see if the fault moves or not. If the fault doesn't move then you can begin to suspect the clock hardware and not the tube.

Most E1T tubes do not light up perfectly from positions '0' to '9' in identical fashion. Manufacturing differences mean that the glow or glows have different intensities.[72] Other faults are real - the tube doesn't count or has physical defects.

Spurious lines

The most common 'faults' seen on an E1T, even perhaps most E1T tubes, are additional fainter lines: Here's a tube in position '3' with a fainter '2' lit. The explanation for this is asymmetrical deflection. Deflection plate x' operates at a fixed voltage but deflection plate x'' (as shown above using an oscilloscope) varies. This asymmetrical deflection causes the electron beam to widen as it moves towards the left (as seen from the electron gun) which is the higher numbered positions on the E1T.

72 These tubes were made in the 1950's after all….

 Copyright Grahame Marsh/Nick Stock 2019

Dim Glows

Another occasional 'fault' is that the selected position is not very bright. This tube has a remarkably dim '9'. It is still visible and brighter than the '8' and '7' additional lines. Therefore I use this tube in one of the 10's positions, for example 10's of hours, so the '9' is not used much. I do have the odd tube where the wanted line and the spurious lines are more or less the same brightness. I would use this tube in a units of seconds position so it changes every second, then the eye sees it steps through the sequence '3' … '3' & '4' … '5' and you can then read the '4' position correctly.

Burns

Look at position '6' - there is a black line showing the phosphor has been burned by the tube. This indicates that the tube has been left glowing in that position for a long time. This tube was sold as NOS but it is most clearly not!

Stuck Tube

The occasional tube is just plain stuck and won't budge *whatsoever*. Here's an E1T that just displays multiple lines with no movement. *Who knows why*. Clearly not of much use in a *timepiece*..

Copyright Grahame Marsh/Nick Stock 2019

Miss-Counting

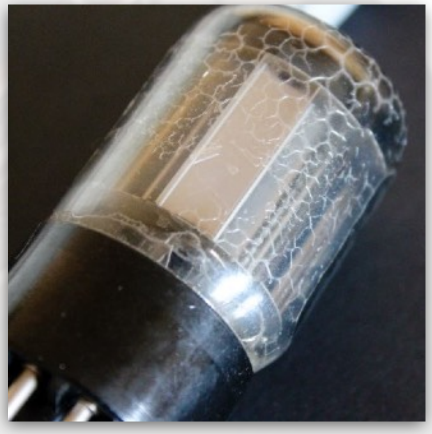
I have one tube that counts 0,2,3,4,5,6,7,8,9,9,0,2 … Looking at the x' and anode 2 voltage you see:

You can see the step from the '0' position to the '2' position is twice the height of the other steps. Double stepping like this does not normally occur until the step pulse voltage is over 20V. Experimentally I tried reducing the step voltage but all that happened was that tube stopped stepping entirely at about 10V.

My conclusion is that this a a faulty tube and no tweak to the clock electronics would get the tube working.

The tube was given to a collector.

Crazy Lacquering



Some E1T tubes appear to have been lacquered. We know not why. With time the lacquer can be damaged or in extreme cases become crazed. It is not hard to remove the lacquer with a mild solvent such as isopropyl alcohol (IPA). If you decide to do so be careful not to damage the decal.

Damaged Decals

It is not unusual to buy second hand tubes that have scratched decals. Here the decal on the left is badly damaged. The decal on the right has a fairly small hole in the bottom left corner which would not really notice in use in a Fortress E1T Clock. (We hope to have replacement decals available at some stage, so all will not be lost.)

Copyright Grahame Marsh/Nick Stock 2019

Offset Decals

Occasionally, the decal has not been applied exactly in alignment with the glow positions. Here the glow position is a few mm to the right of the centre of the figure '2'. Perhaps I'm being a bit picky...[73]

An Indicator of Tube Age?

Apart from obvious signs of age such a dirt, grime or a damaged decal, there is little to indicate how well used a tube is. A physical observation I have made are NIB tubes have a top glass dome silvering that is is bright and uniform. On tubes which are clearly used and old, the dome silvering is also usually not uniform or sometimes significantly worn away. The tube on the left is (what I believe to be) a new, unused tube. The silvering just reflects the ceiling and camera. However, the tube on the right shows a sort of "horseshoe" marking centred above the top of the heater/cathode. (This marking is inside the tube not on the outside of the glass). I have no proof that these markings occur with age as yet. I intend to document the visible changes to an E1T tube when I have a clock running in the house over a period of years. This said, I would be suspicious of an E1T sold as new/NIB/NOS that showed such markings.

[73] Yes you are….*Ed.*

Copyright Grahame Marsh/Nick Stock 2019

Fortress - How It Works

Board Schematics and Function

Copyright Grahame Marsh/Nick Stock 2019

1. Introduction

The Fortress E1T Clock schematic diagrams are spread across over a dozen pages and many of the individual pages have several subsystems on them. This Fortress 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).
- Battery backup power supply for the RTC.
- User selected option jumpers.
- Hardware version detection.
- Stereo audio amplifiers and speakers.
- A front panel sub-board carrying:
  - PIR (passive infrared) motion detector.
  - IR (infrared) receiver.
  - Rotary encoder.
- Thirty two tricolour (red, green, blue) LED (light emitting diode) modules to bling[74] the clock.
- +300V 10mA power supply to run the E1T tubes.
- +6.3V 2A power supply to run the E1T tube heaters.
- +5V and +3.3V power supplies to run all the analogue and digital electronics.
- An optional WiFi module.
- An optional GPS receiver.
- Six E1T tube display drivers. Each E1T tube driver consists of:
  - +300V switch so an individual tube can be blanked.
  - Reset circuit that forces an individual tube back to the '0' position.
  - Pulse circuit that causes and individual tube to step forward one position, say, '0' to '1', then on the next pulse '1' to '2'.

Now let's look at these in more detail. The Eagle files[75] including the schematics, PCB layouts and Gerbers are all open design and available on the project public Dropbox.

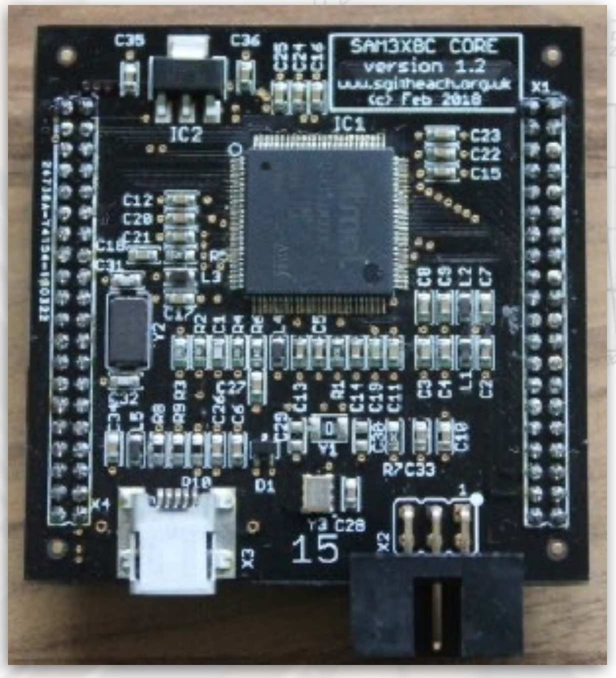
[74] Or pimp if you prefer…..hey, we won't judge…

[75] 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.

Copyright Grahame Marsh/Nick Stock 2019

1.0 SAM3X8C Microcontroller Plug-in

Arduino fans may recognise this number in relation to the Arduino Due[76] as it uses the SAM3X8E micro-controller. The SAM3X8C[77] 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:

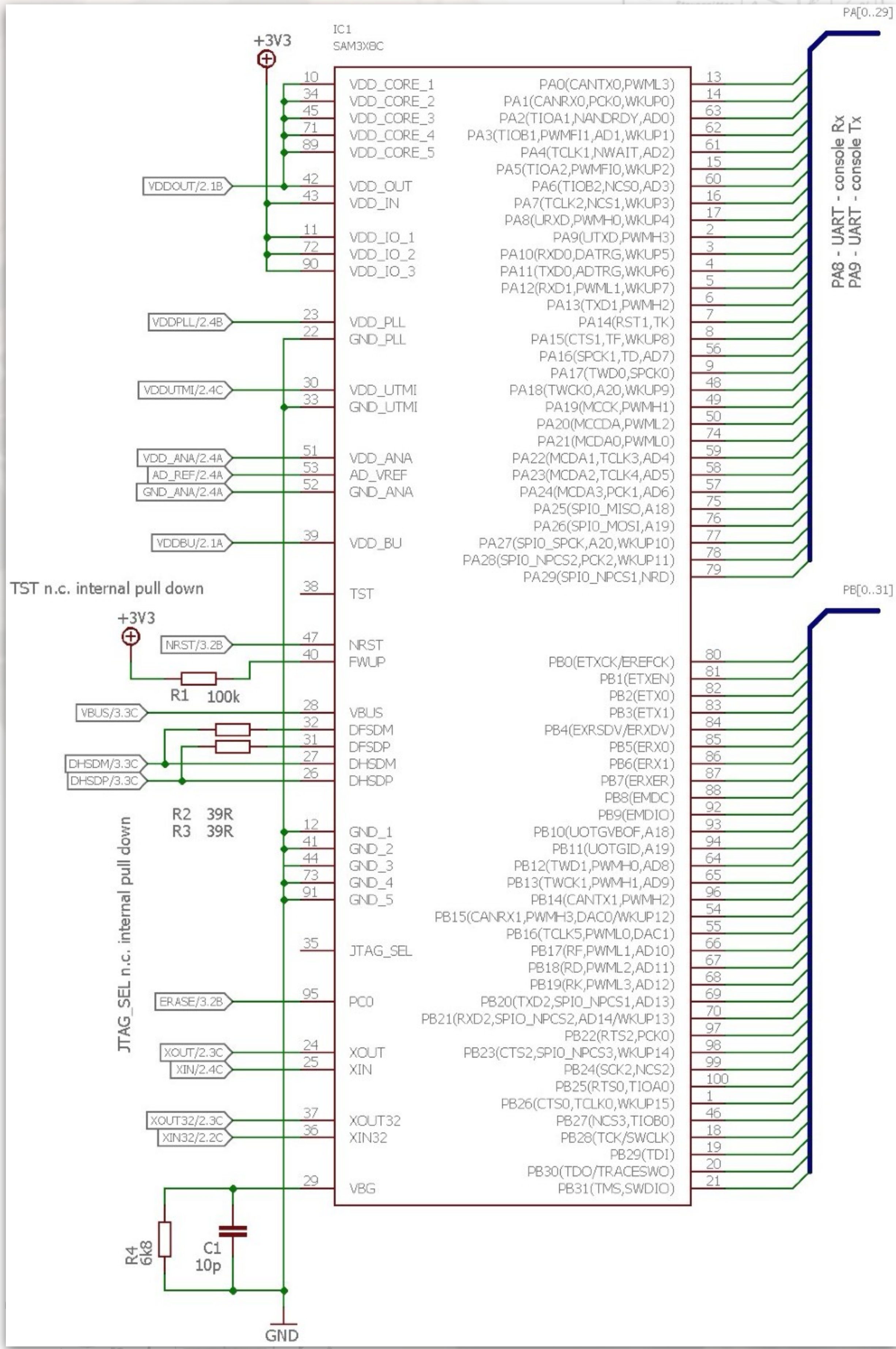
It carries:

- Native USB port.
- Separate 3.3V regulator .
- 6 pin programming port.[78]
- 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.

[76] https://store.arduino.cc/arduino-due
[77] https://www.microchip.com/wwwproducts/en/ATSAM3X8C
[78] For use with the Sgitheach SAM programmer ONLY and not any old Atmel product

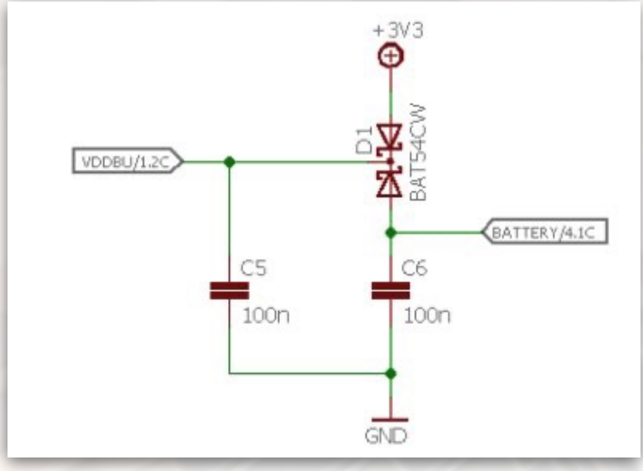
Copyright Grahame Marsh/Nick Stock 2019

1.1 SAM3X8C schematic

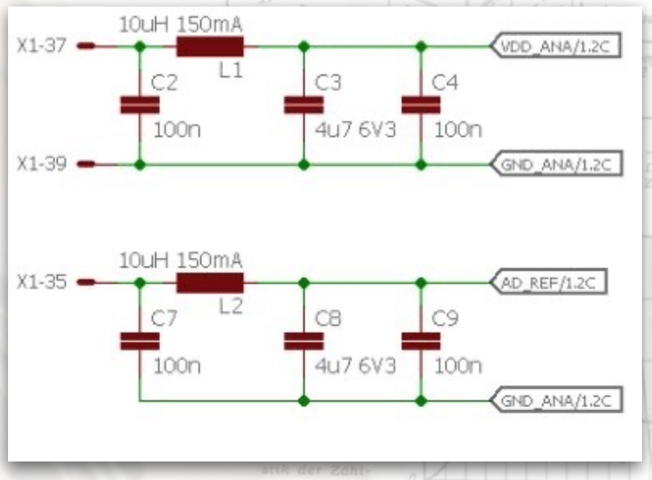


 Copyright Grahame Marsh/Nick Stock 2019

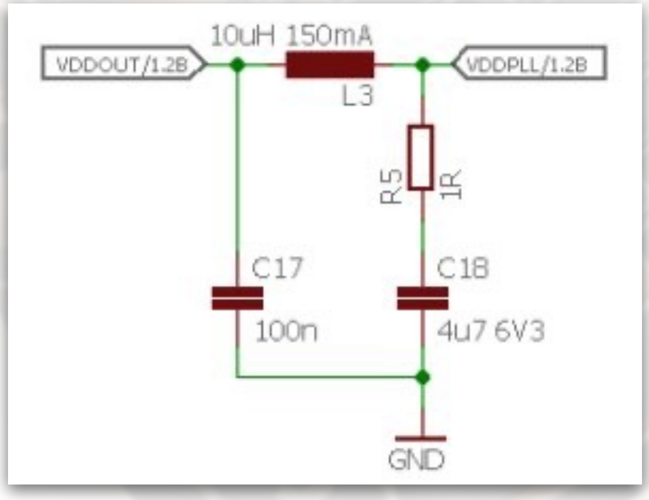
1.2 Battery Backup



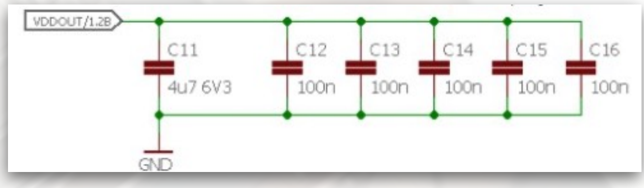
1.3 Analogue Decoupling



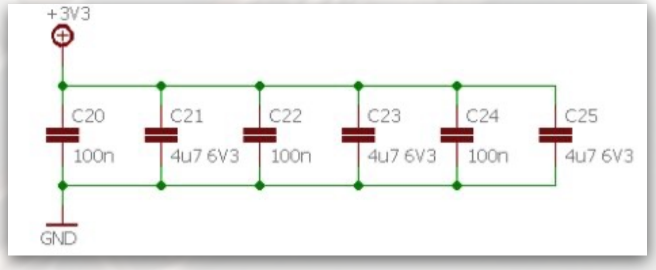
1.4 PLL Decoupling



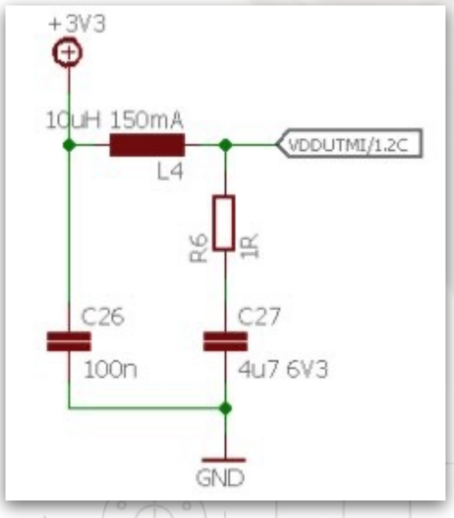
1.5 Core Decoupling



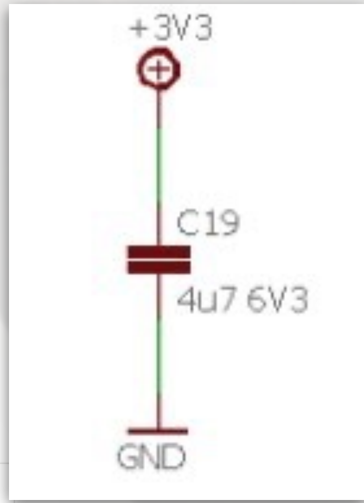
1.6 IO Decoupling



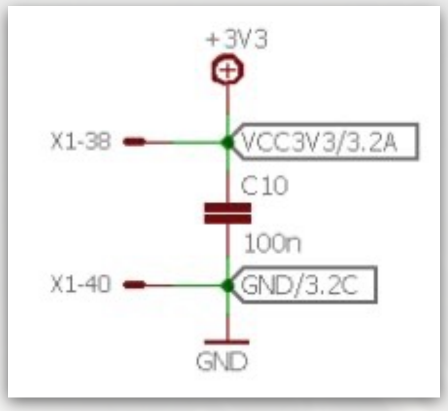
1.7 UTMI Decoupling



1.8 3.3V IN Decoupling

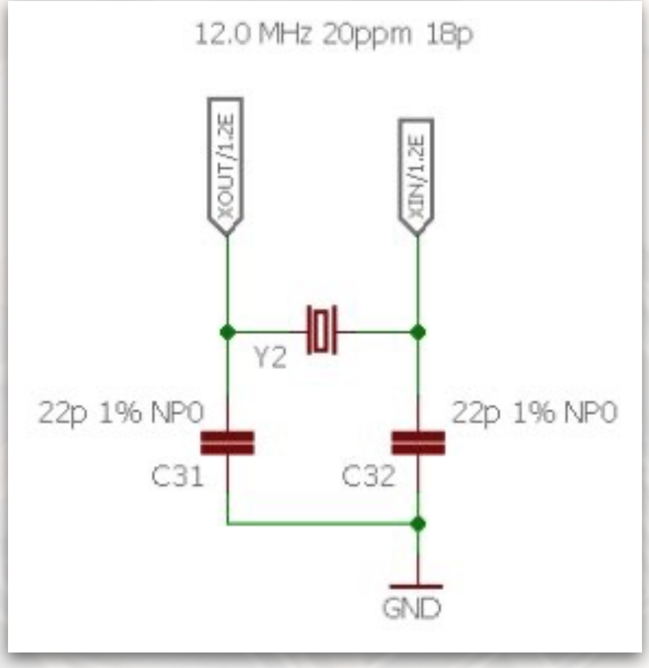


1.9 3.3V OUT

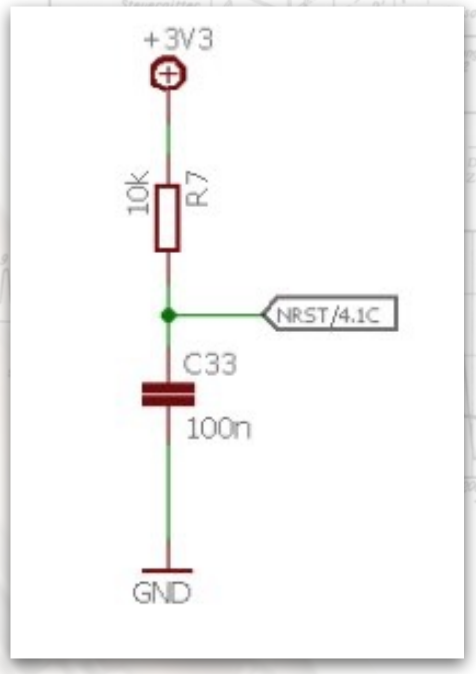


Copyright Grahame Marsh/Nick Stock 2019

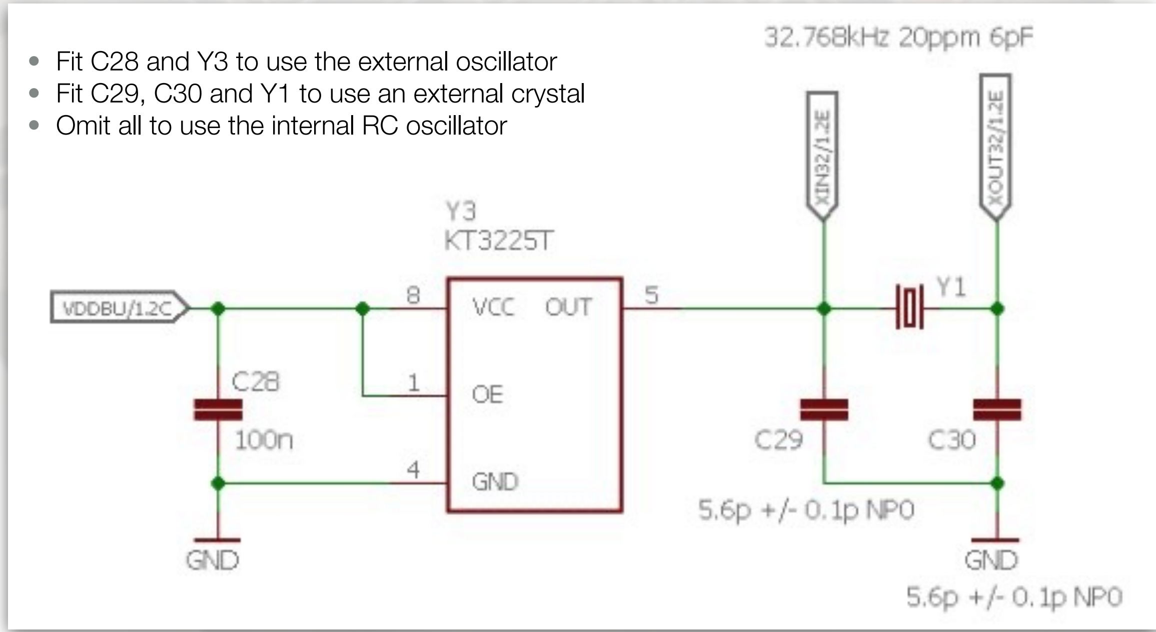
1.10 System Oscillator 12MHz Crystal



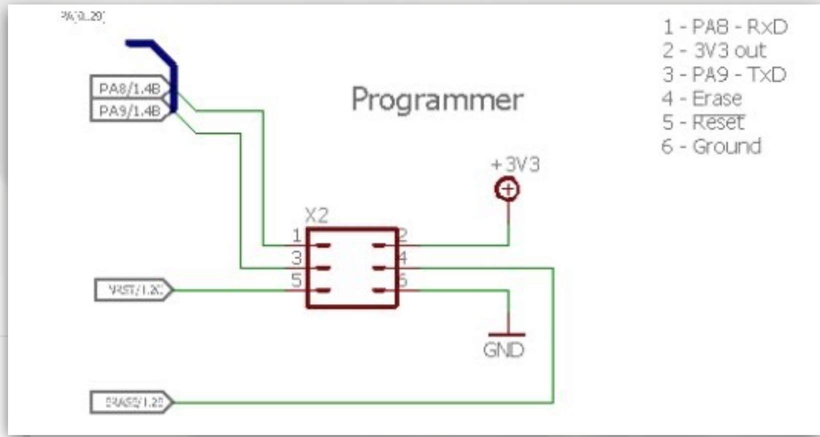
1.12 Reset



1.11 RTC Oscillator Options



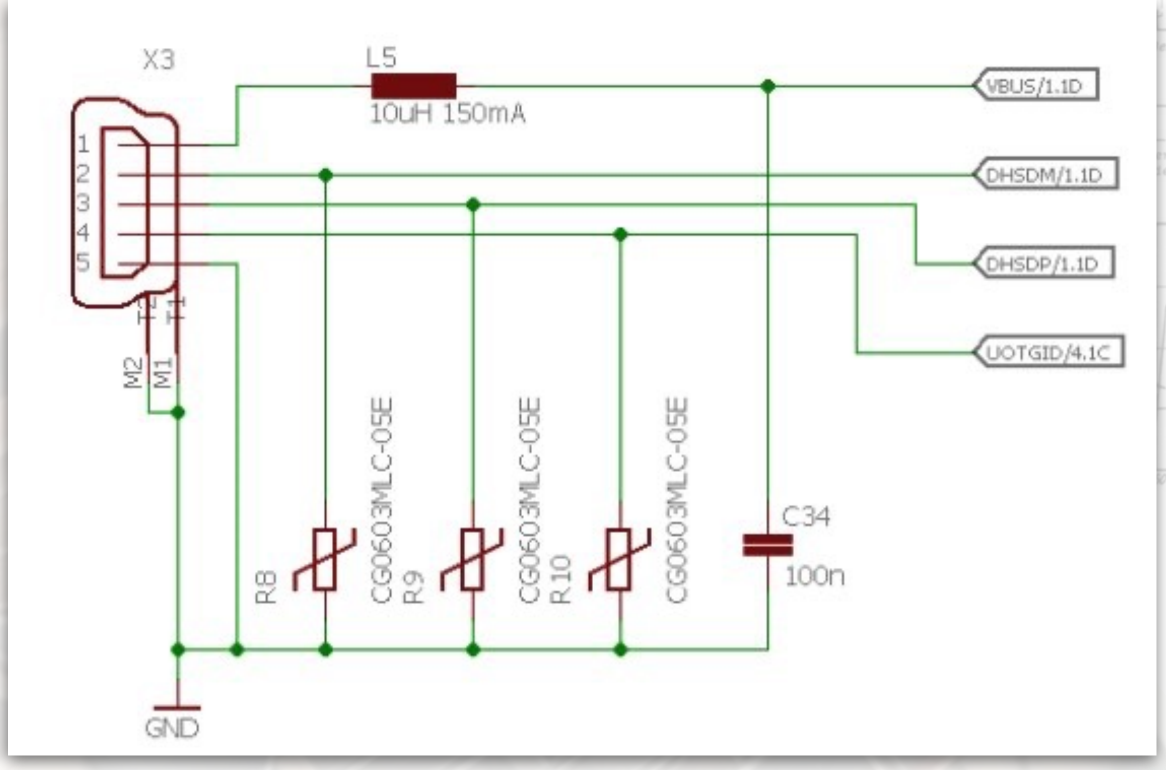
1.13 Programmer Socket



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

Copyright Grahame Marsh/Nick Stock 2019

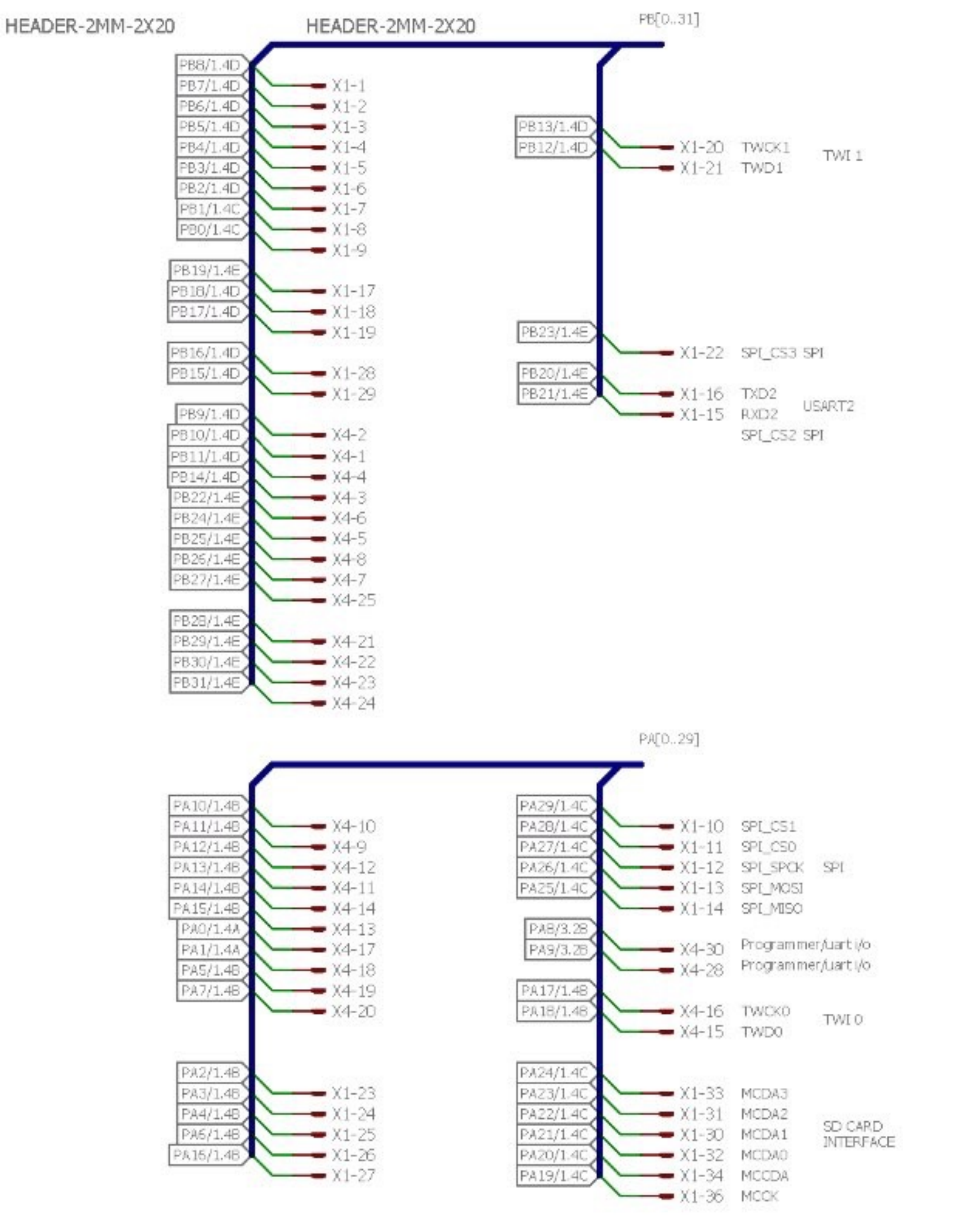
1.14 USB Socket



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

Copyright Grahame Marsh/Nick Stock 2019

1.15 Header Connectors

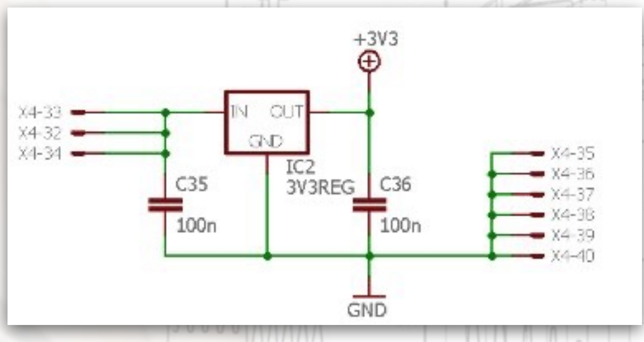


To conserve space 2mm spaced headers are used instead of the more usual 2.54mm spaced headers.

Copyright Grahame Marsh/Nick Stock 2019

1.16 Miscellaneous Connectors

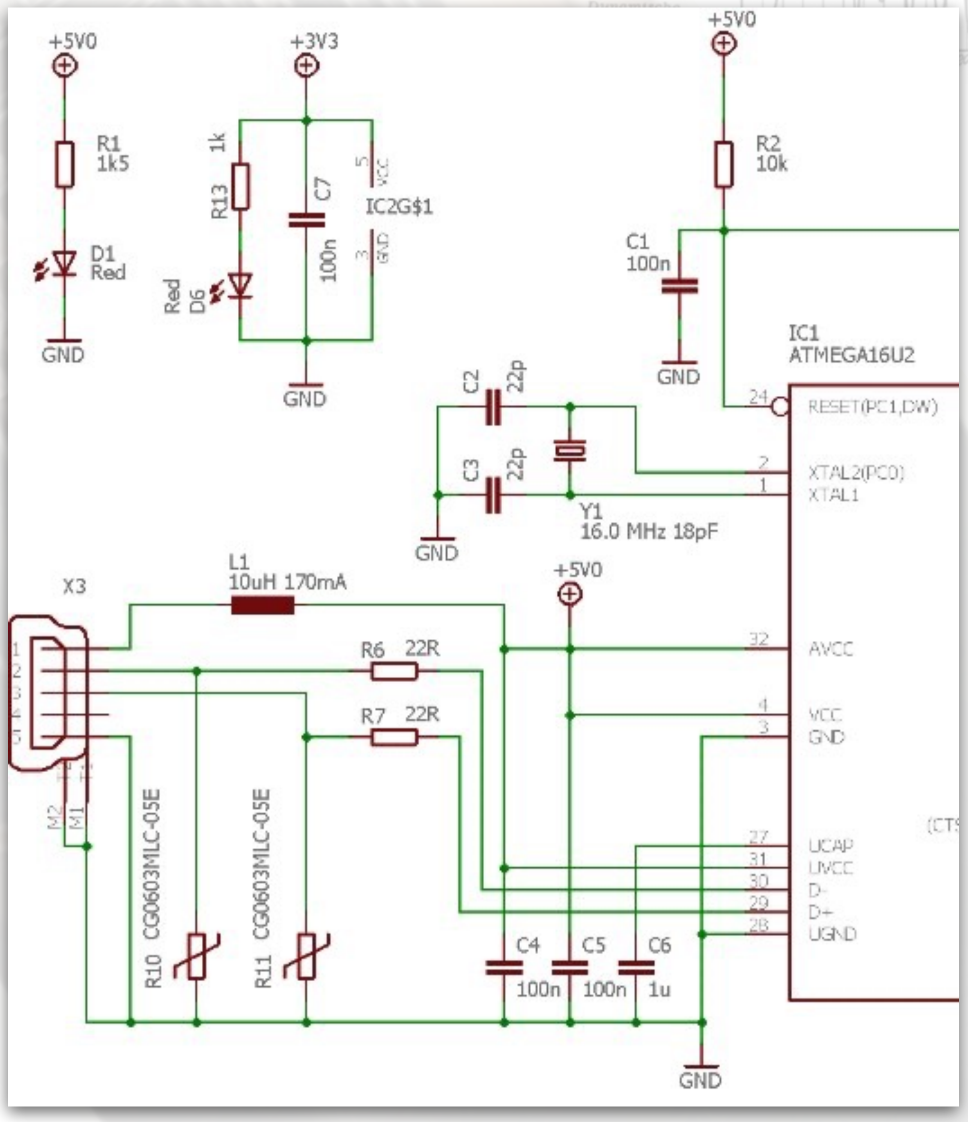
1.17 3.3V PSU



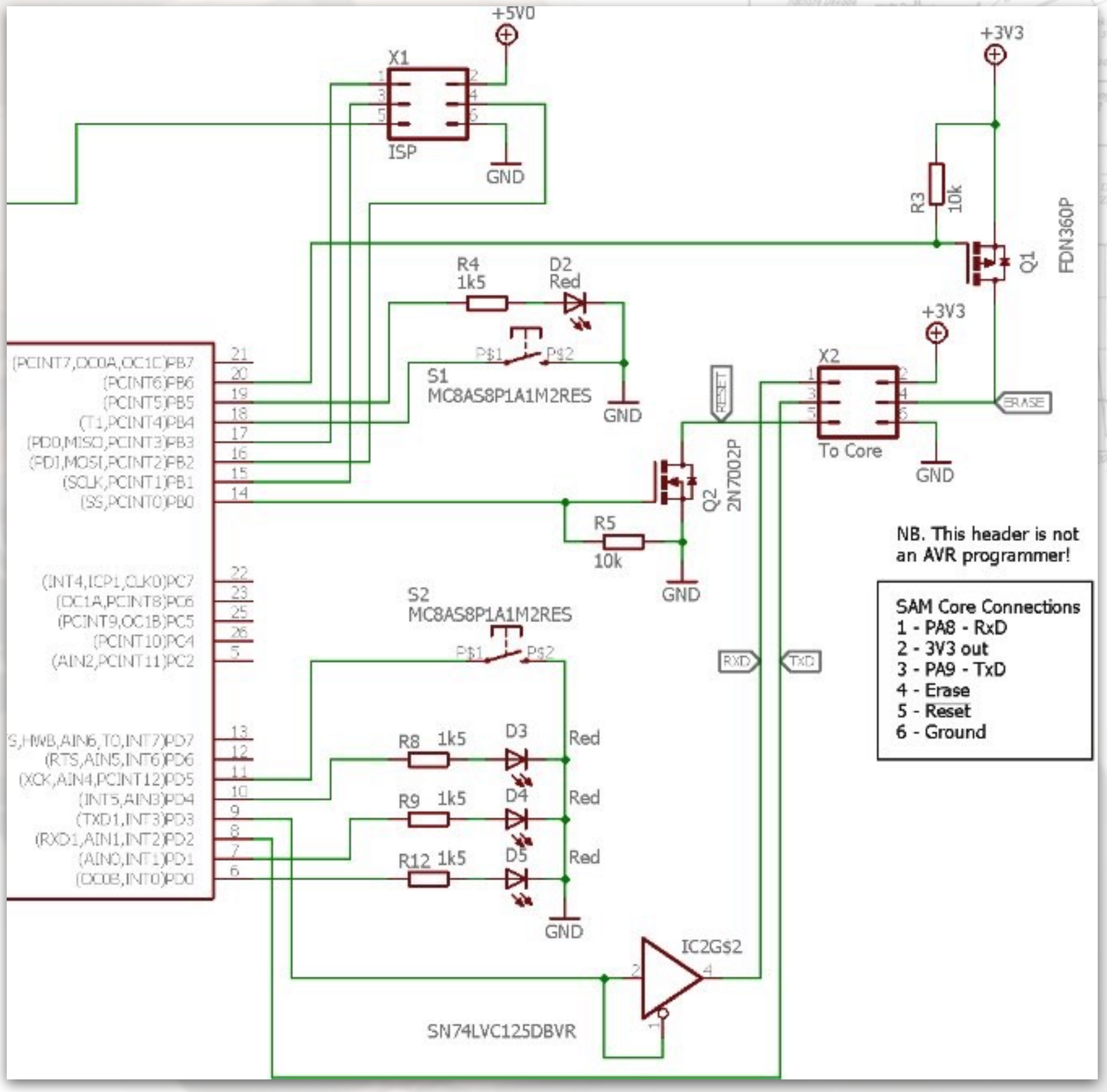
This regulator can be used to supply power but care must be taken not overload it.

2.0 SAM3X8C Programmer

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



  Copyright Grahame Marsh/Nick Stock 2019



- 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.

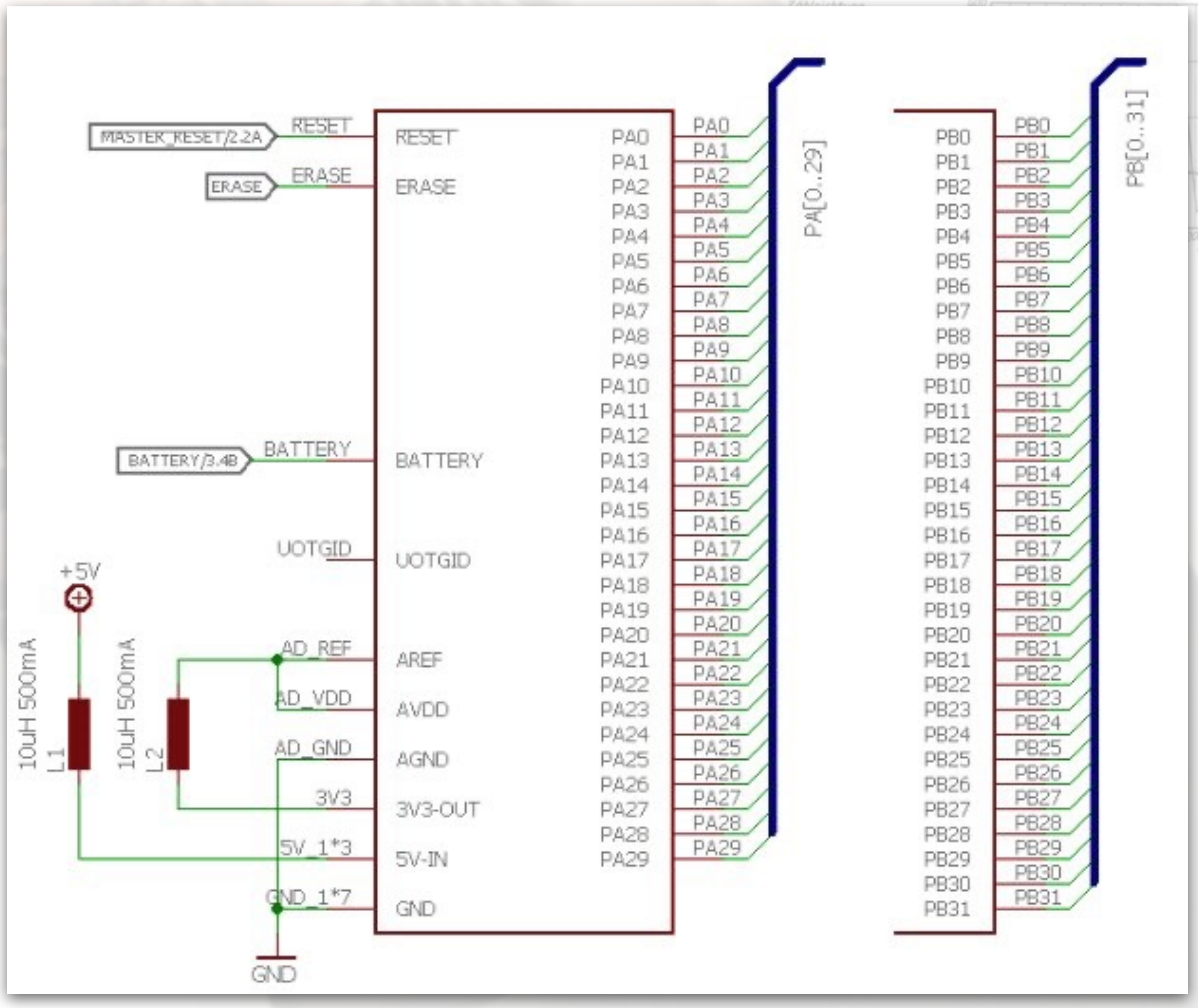
The programmer requires both the USB 5V supply and the 3.3V supply from the SAM3X8C plug in board to function.

Copyright Grahame Marsh/Nick Stock 2019

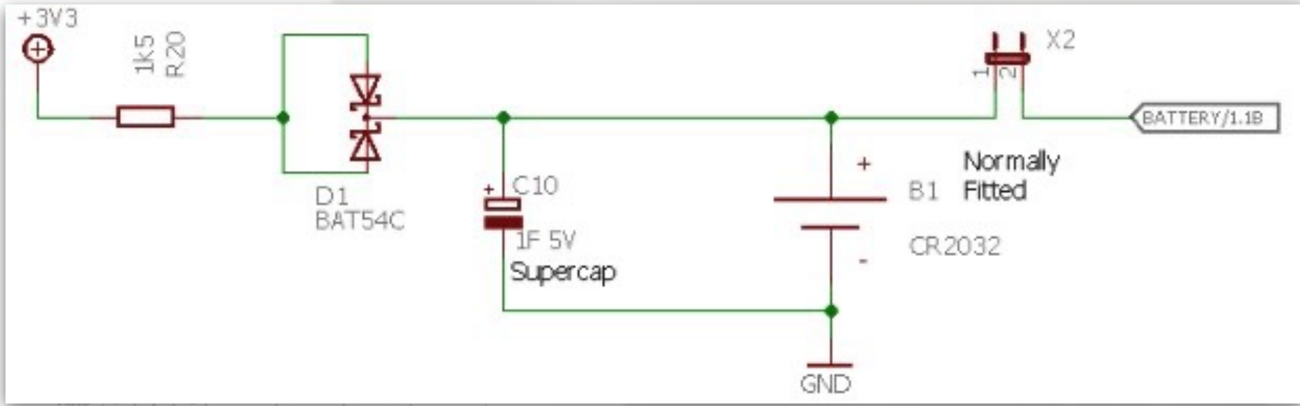
3.0 Fortress Main Board

The main board carries all of the clock electronics except the E1T tubes and their drivers. The E1T display board and main board are designed to mount back to back with inter-connectors to pass signals and power between the two boards.

3.1 SAM3X8C Plug In



3.2 Back up power (A super cap or a Li cell provide back up power; one or the other is fitted).

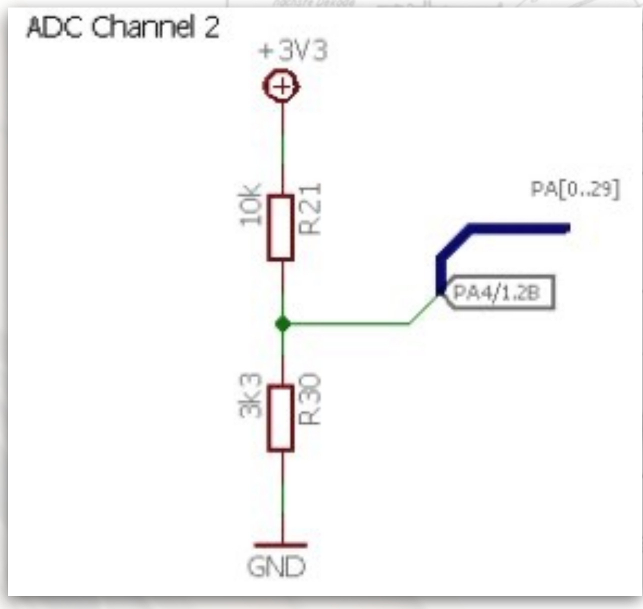


X2 allows the back up power supply PSU to be disconnected from the SAM3X8C so all of the registers in the chip can be reset to factory default values.

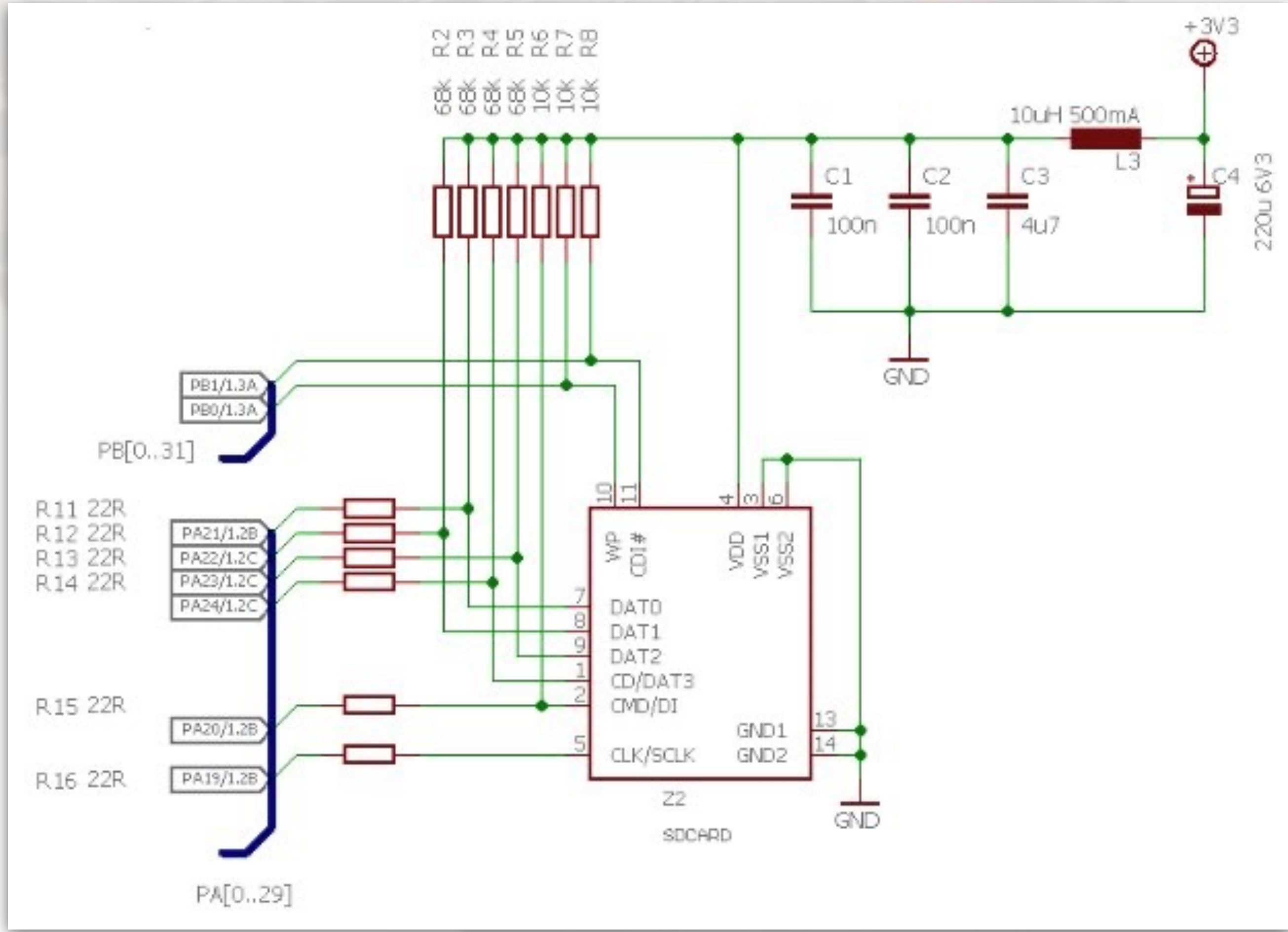
 Copyright Grahame Marsh/Nick Stock 2019

3.3 Hardware Version Sense

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



3.4 Storage - SD Card



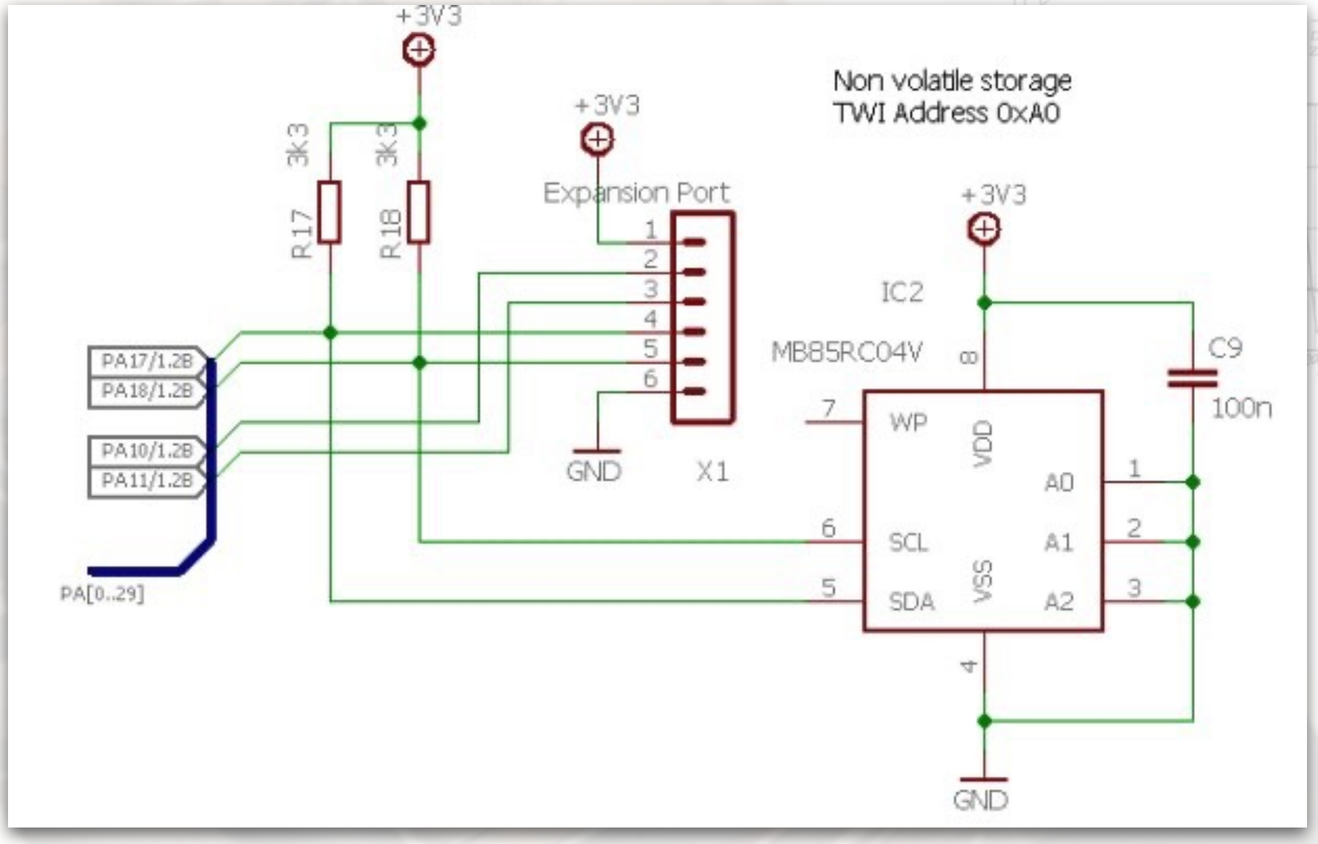
This is a native SD Card interface[79] and not a slow SPI based one. It transfers 4 bits of data per clock cycle (the SPI transfers 1 bit) and will clock at 50MHz with a Class 10 SD Card (SPI will clock

[79] Chapter 37 SAM3X8C data sheet

Copyright Grahame Marsh/Nick Stock 2019

at 10MHz). So 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.

3.5 Storage - Non-Volatile RAM (NVR)

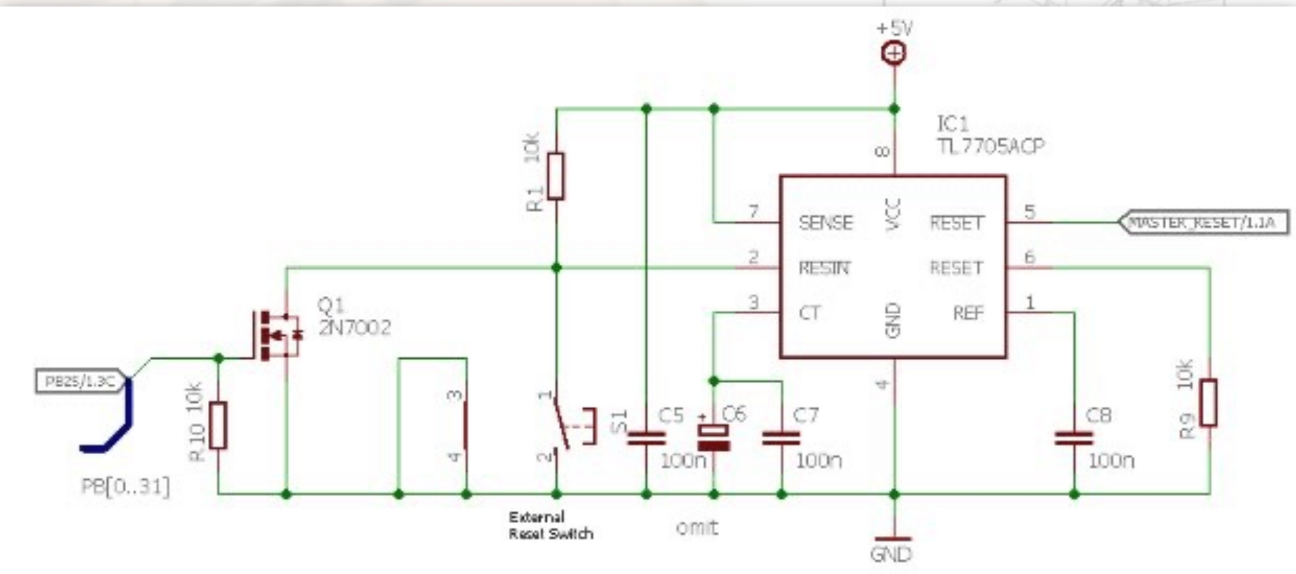


The SAM3X8C has no internal EEPROM so some TWI (I2C) NVR[80] is provided to hold data when the clock is switched off and can also be used to pass information between clock faces. The required TWI pull up resistors are here. Two, otherwise unused I/O pins, power and the TWI bus are brought out on to a header to provide a possible route for expansion. Other TWI users are the option jumpers and audio amplifiers, see below.

80 https://www.fujitsu.com/uk/Images/MB85RC04V.pdf

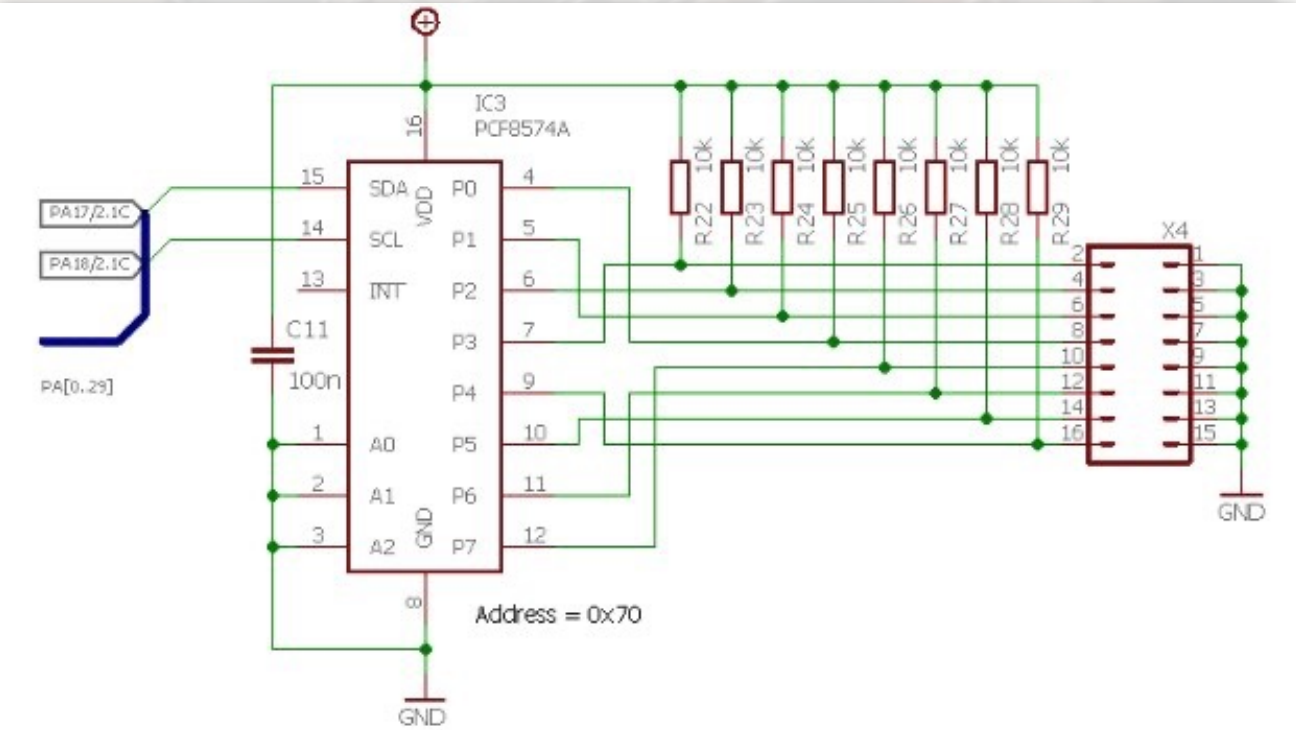
 Copyright Grahame Marsh/Nick Stock 2019

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.

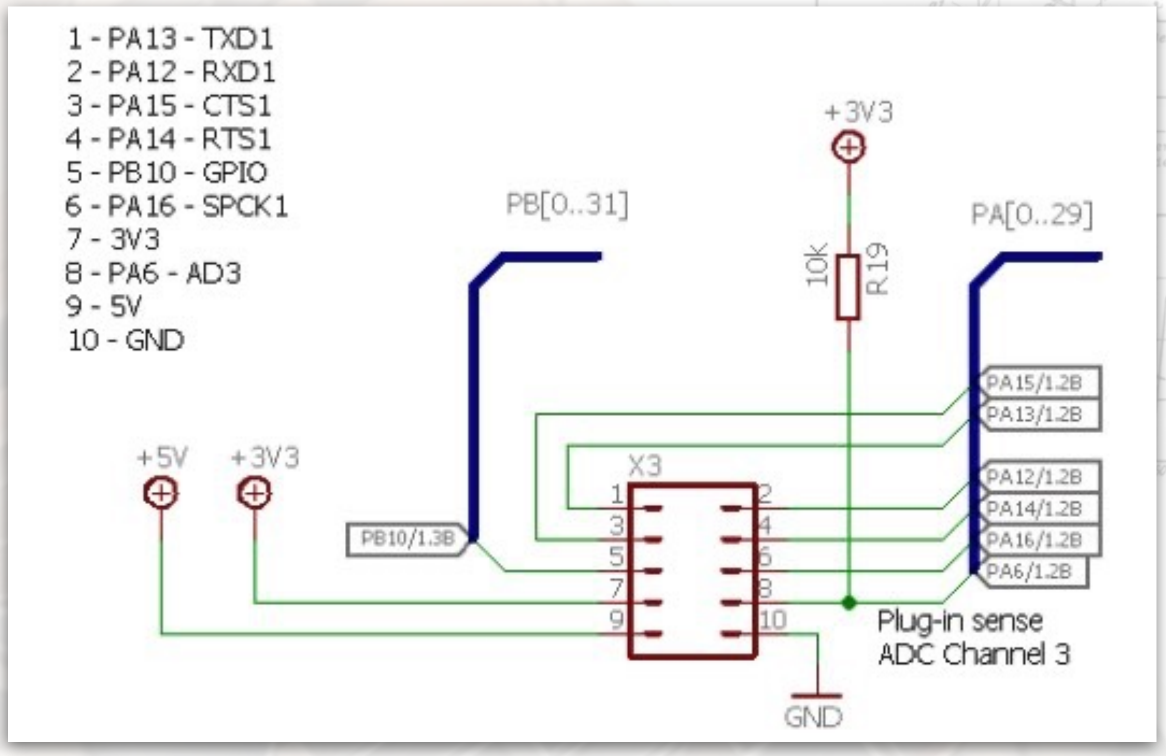
3.7 Option Jumpers



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.

  Copyright Grahame Marsh/Nick Stock 2019

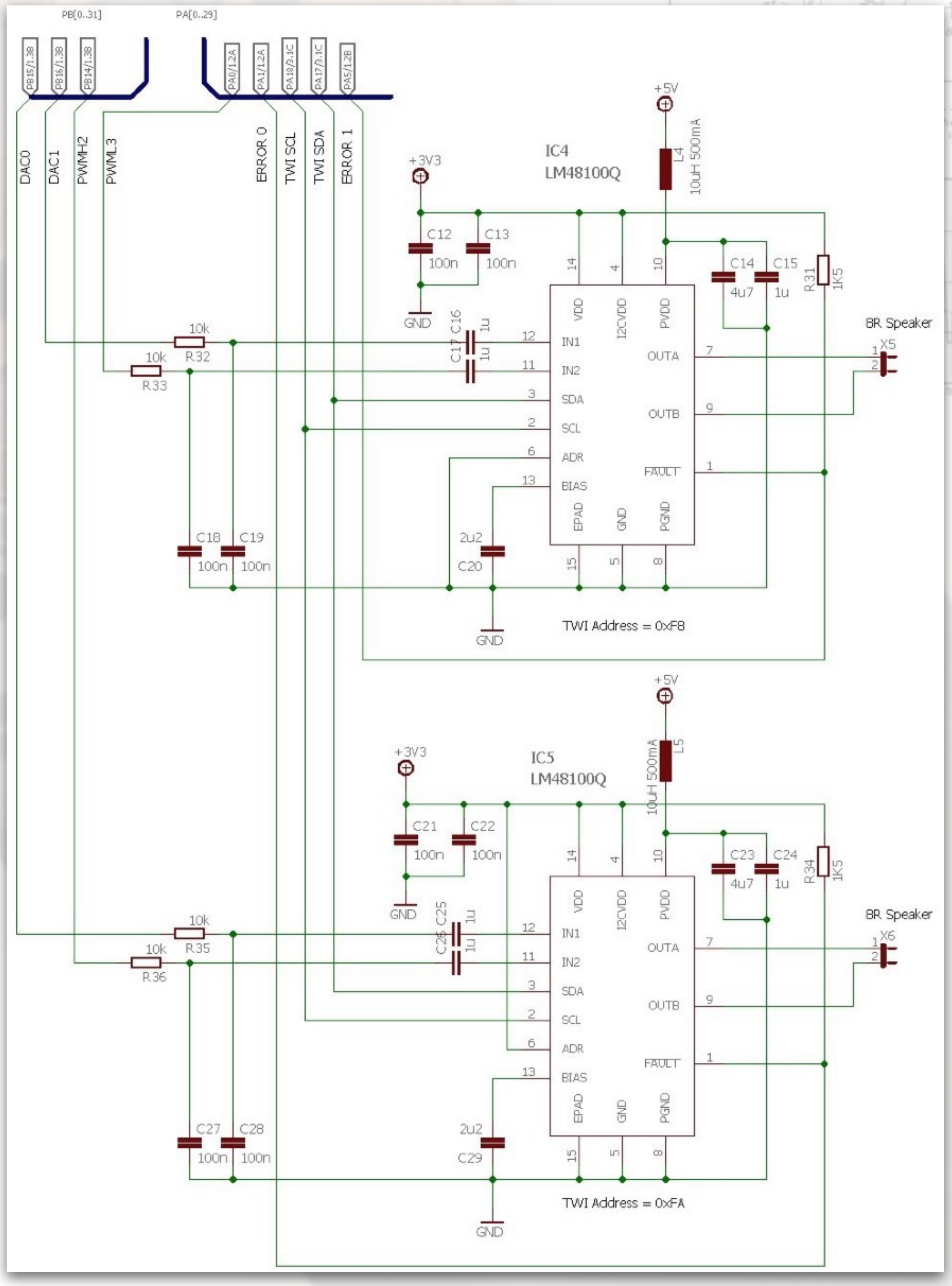
3.8 Plug In Module Socket



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

 Copyright Grahame Marsh/Nick Stock 2019

3.9 Audio Amplifiers



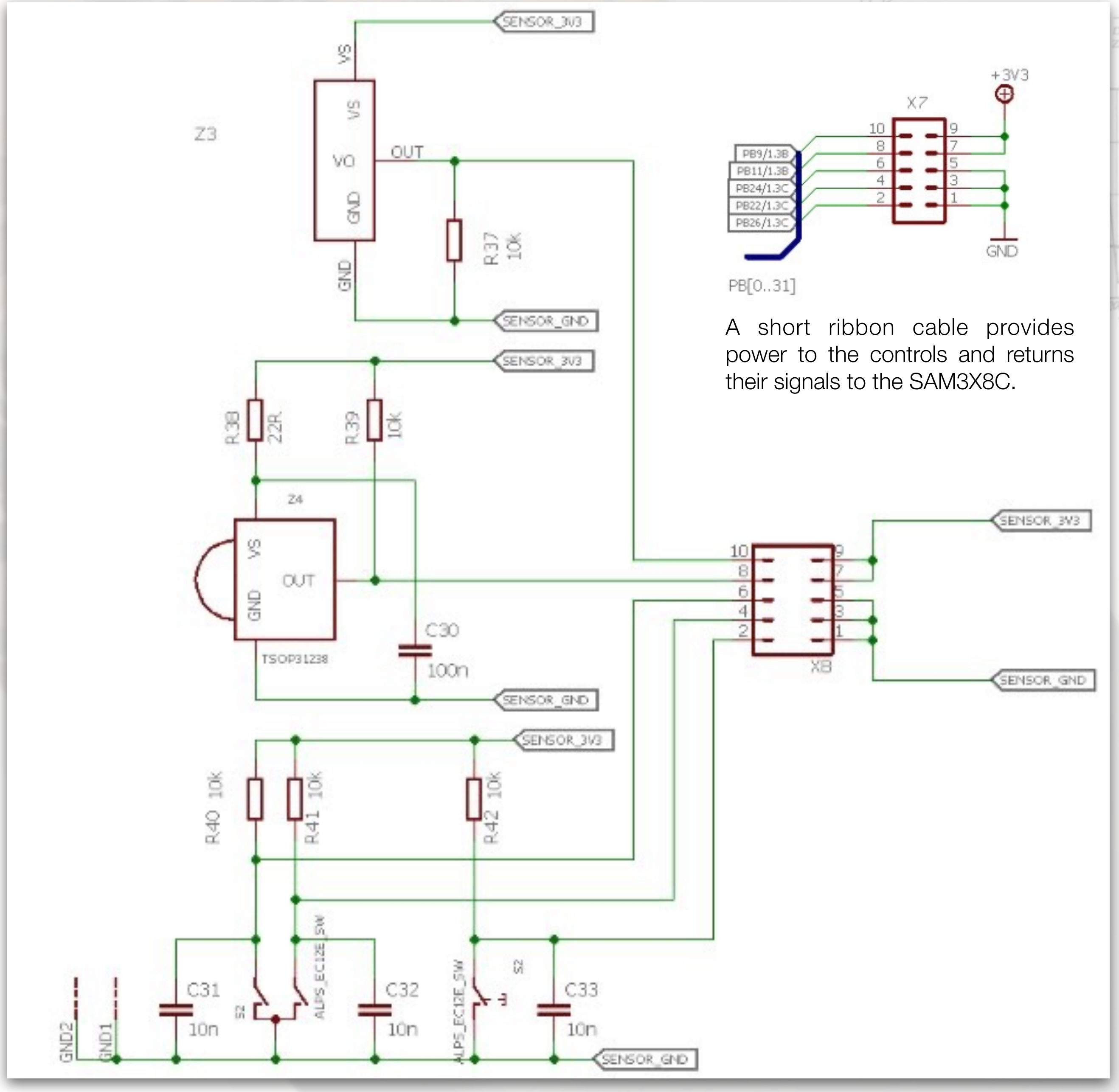
The stereo audio amplifier[81] each take signals from the SAM3X8C digital to analogue (DAC) convertor outputs and from pulse width modulated (PWM) I/O pins. Four signals in total. The DAC output are used to play WAV files from the SD card and the PWM output can make a wide

[81] www.ti.com/lit/ds/symlink/lm48100q-q1.pdf

  Copyright Grahame Marsh/Nick Stock 2019

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.

3.10 Front Panel Controls

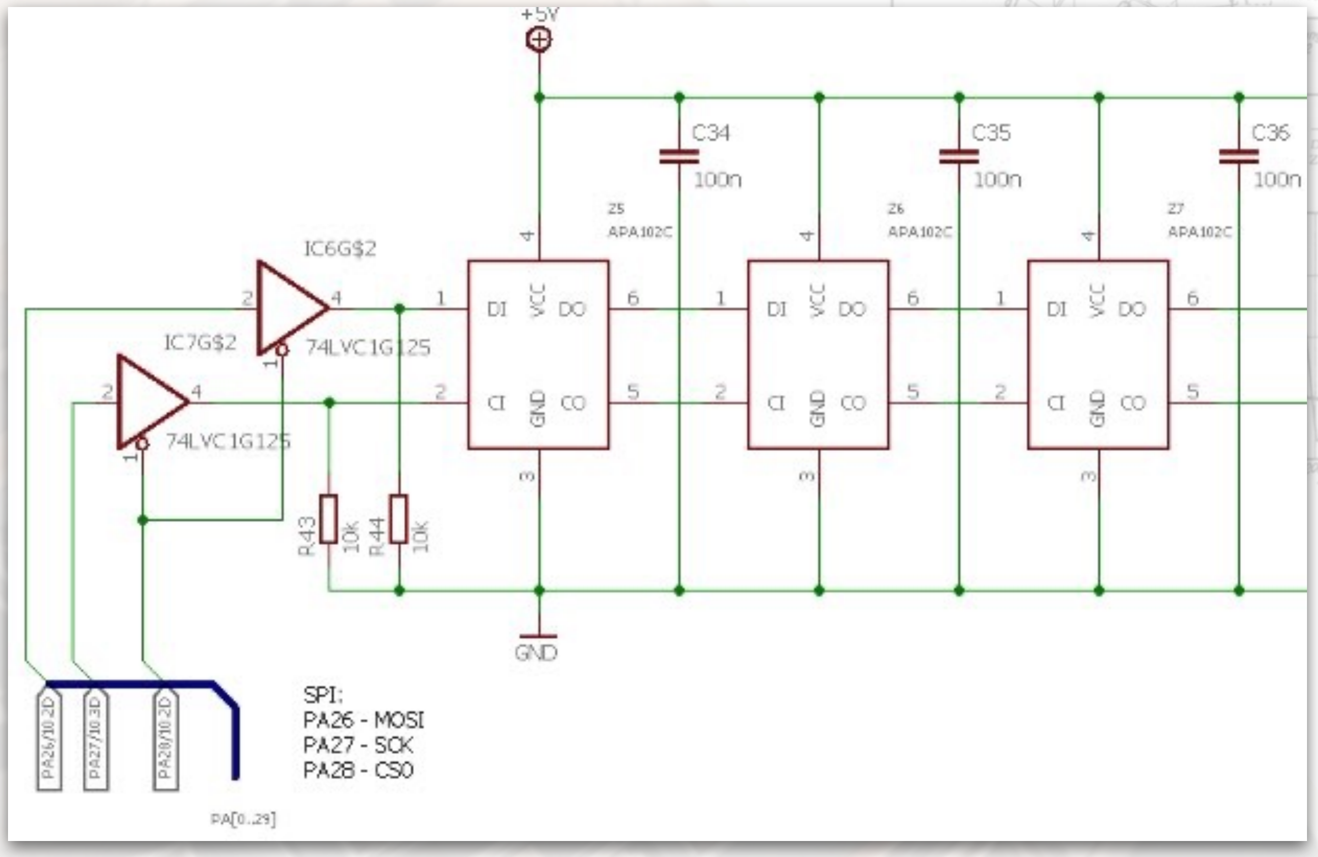


The Fortress clock has three front panel controls/devices. Z3 is a PIR sensor used to detect room occupancy. When the room has been unoccupied for a period of time the clock goes to sleep. Z4 is an IR receiver and the clock then uses a small IR transmitter handset which can be used to select the clock “face” to use, adjust the audio amplifier volume settings, etc. Finally, S2 is a rotary encoder that again can select the clock “face” and adjust the chime volume.

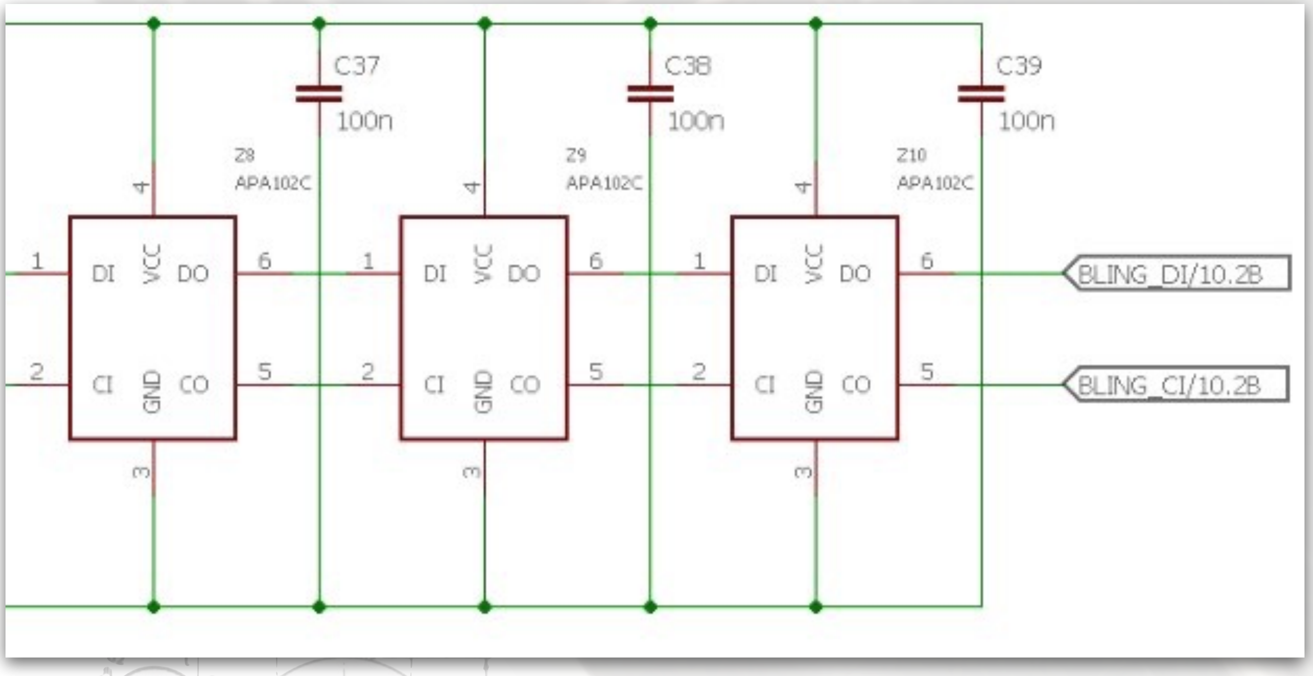
The components shown above are mounted on their own PCB which is arranged to be vertical so the controls face into the room.

 Copyright Grahame Marsh/Nick Stock 2019

3.11 Bling-Bling RGB LED Modules



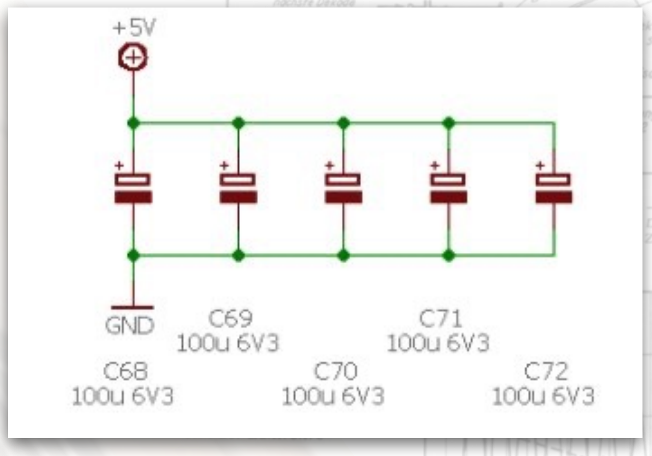
The Fortress clock has a total of thirty two RGB LED modules. These devices are connected to Channel 0 of the SPI bus. The devices used require a 5V supply and signals so two buffers are used to select the SPI channel and act as 3.3V to 5V level shifters. The thirty two modules are chained together and the first three are shown above.



 Copyright Grahame Marsh/Nick Stock 2019

The main board has six modules in total and these are arranged to fire downwards to illuminate underneath the case. The outputs from Z10, the 6th module, are taken to connectors to the display board.

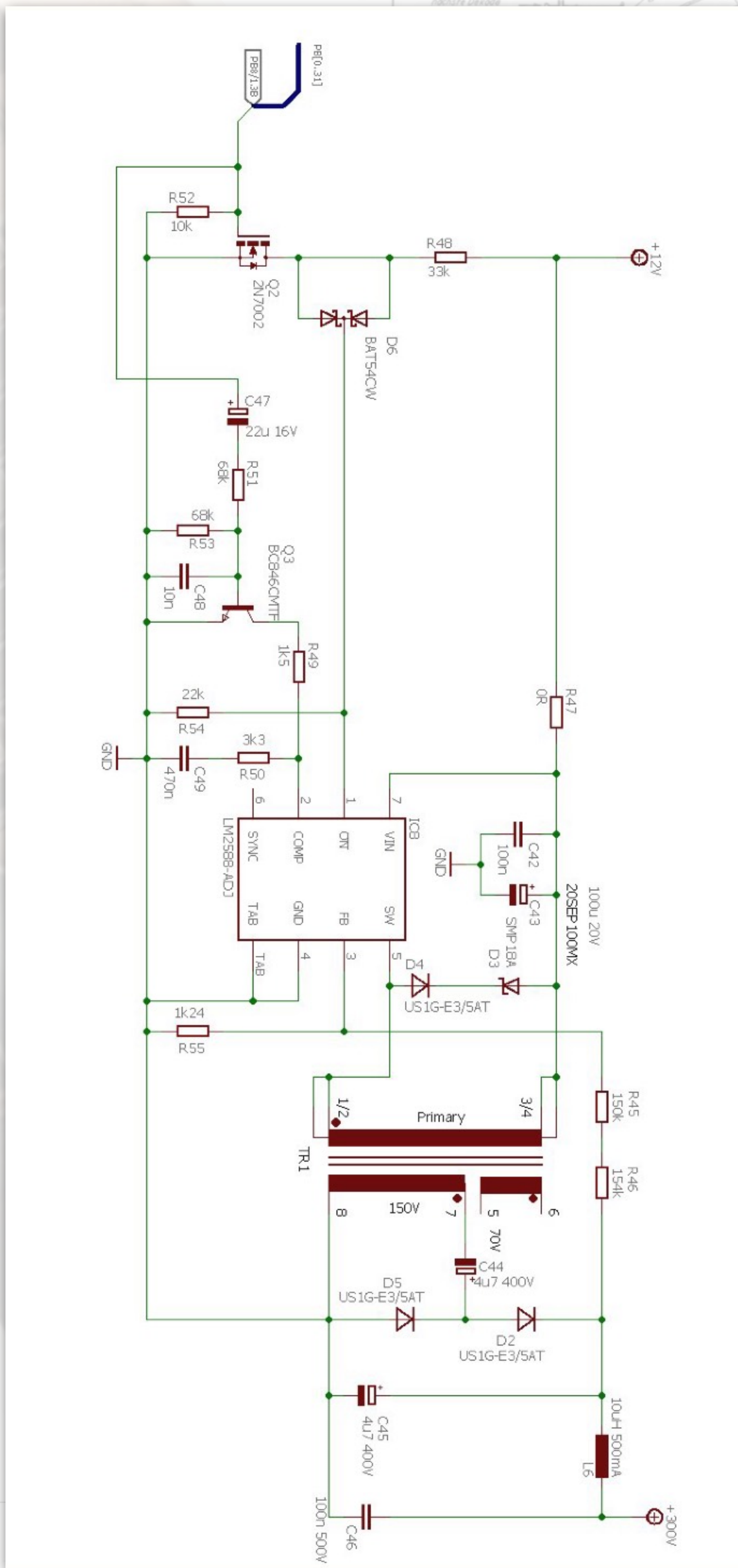
In addition to the decoupling capacitor across each module, some energy storage is allowed for along the 5V supply traces on the PCB.



Copyright Grahame Marsh/Nick Stock 2019

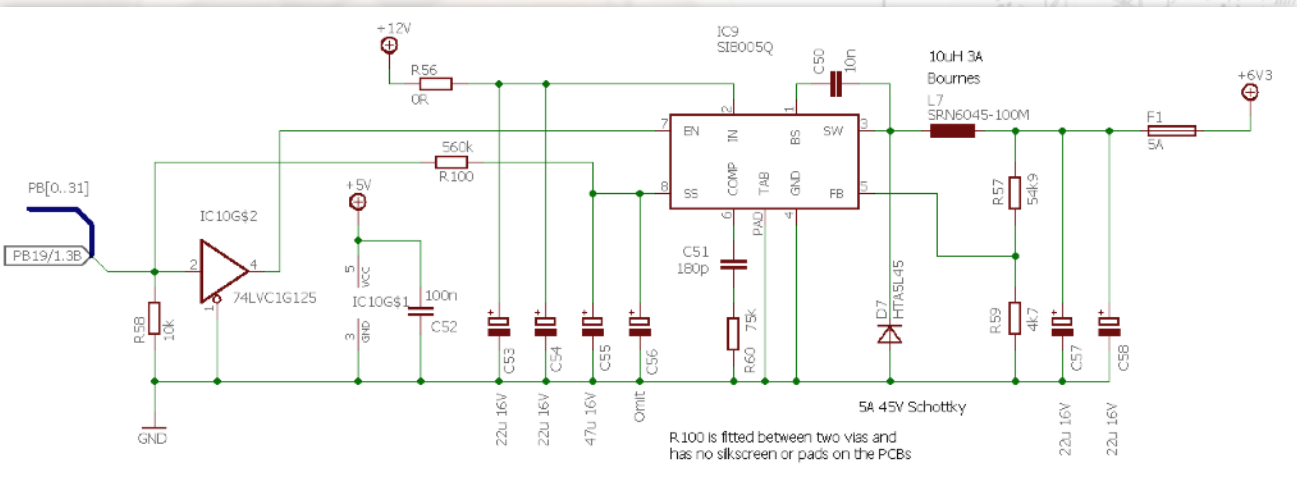
3.12 +300V HT PSU

The E1T tubes require +300V to operate. This power supply is a simple flyback convertor using a custom transformer. It is a very robust supply. Q2 acts as a level convertor and makes the logic positive: a 3.3V input is required to switch the convertor on. The circuit around Q3 provides a soft start to the convertor.



 Version 1.1 Copyright Grahame Marsh/Nick Stock 2019

3.13 6.3V Heater PSU



The E1T tube heaters operate at 6.3V 300mA, therefore this switched mode down convertor will provide about 2A. The heater PSU can be switched on by the SAM3X8C via the buffer/level convertor IC10. The convertor has a soft start feature which is configured to take roughly 10 seconds to increase the output voltage from 0V to 6.3V. This substantially reduces the inrush current to the cold heaters, hopefully prolonging their life.

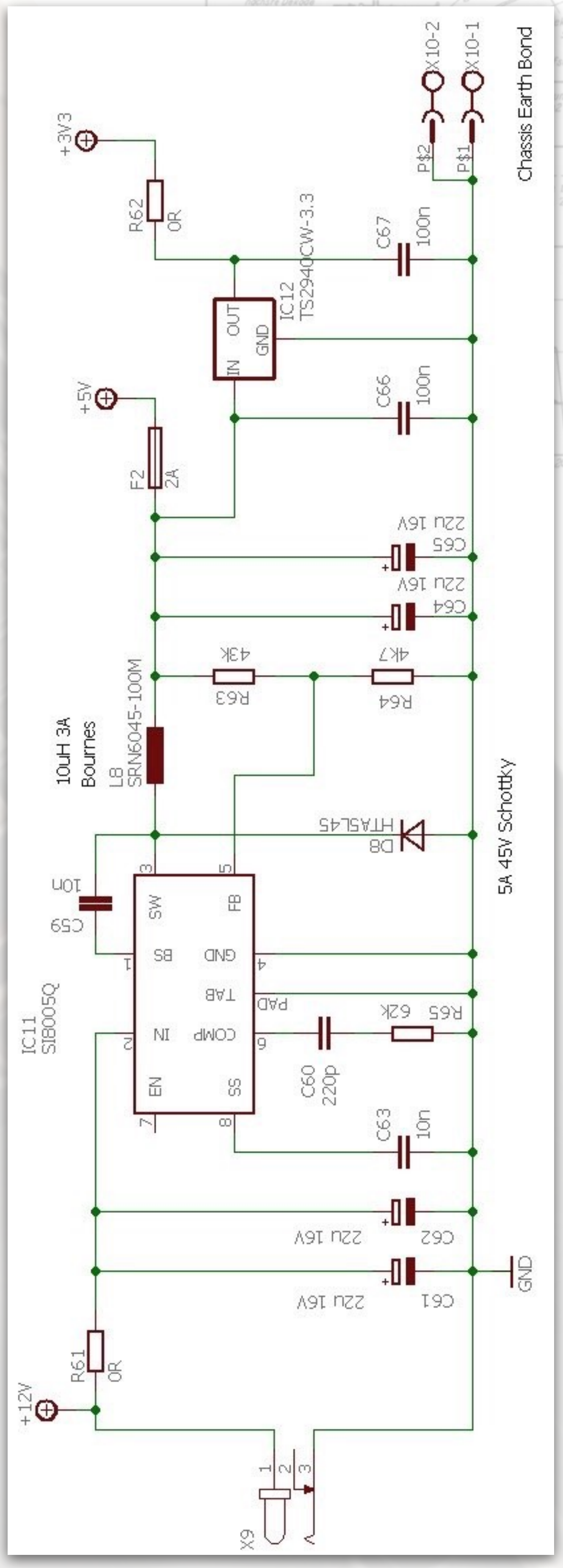
Copyright Grahame Marsh/Nick Stock 2019

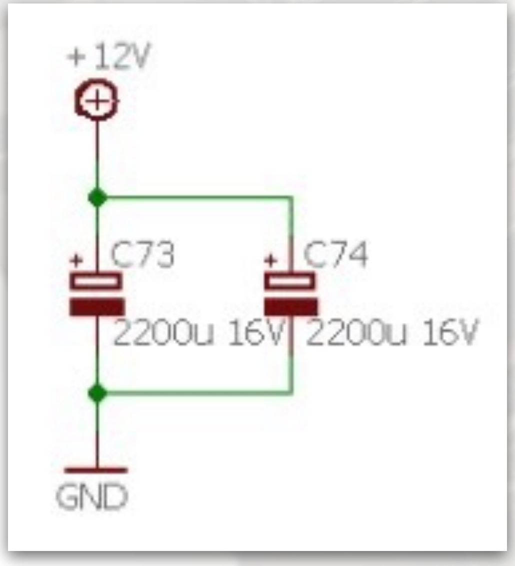
3.14 5V PSU and 3.3V PSU

This starts with the 12V DC input from the remote power supply. The switch mode down converter is almost the same as the 6.3V heater PSU. Differences are that it has no on/off function but remains on all the time. The feedback potential divider has different value of course and the compensation network is slightly different. Otherwise it uses the the same chip, inductor and diode.

The 5V output is passed to a 3.3V low dropout regulator.

Also note the negative input is connected to two terminals marked chassis earth bond. If the final case is metal then these connections are used to connect the chassis and the negative supply line must be earthed in the remote 12V PSU.

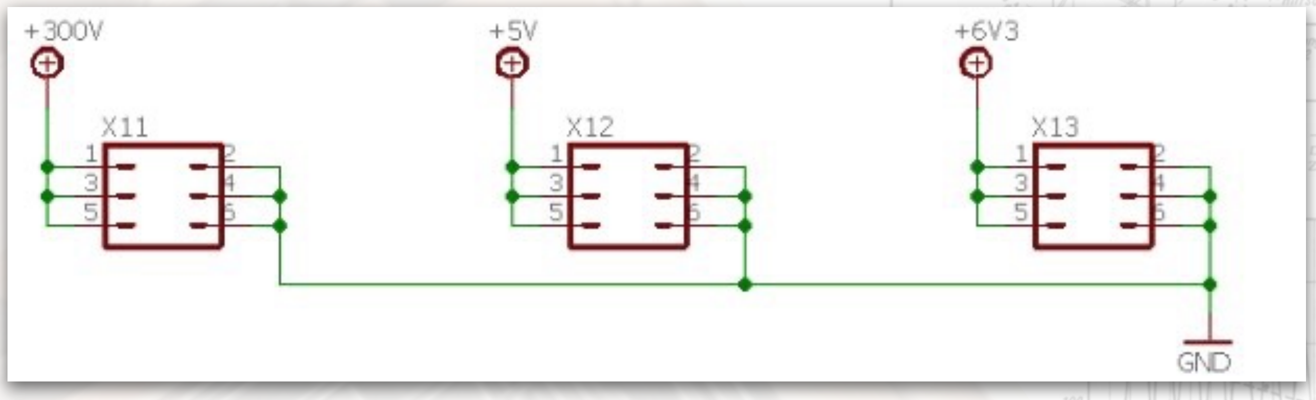




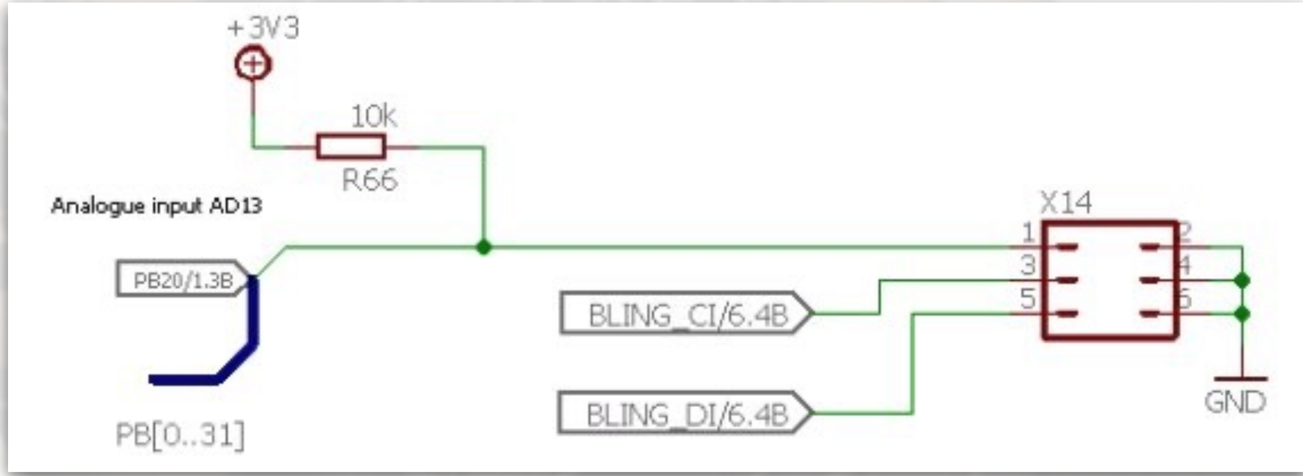
These capacitors are added for local energy storage for peak current demand, such as when the flyback convertor starts. In practice I have not found them to be critical.

Copyright Grahame Marsh/Nick Stock 2019

3.15 Inter-board Connectors

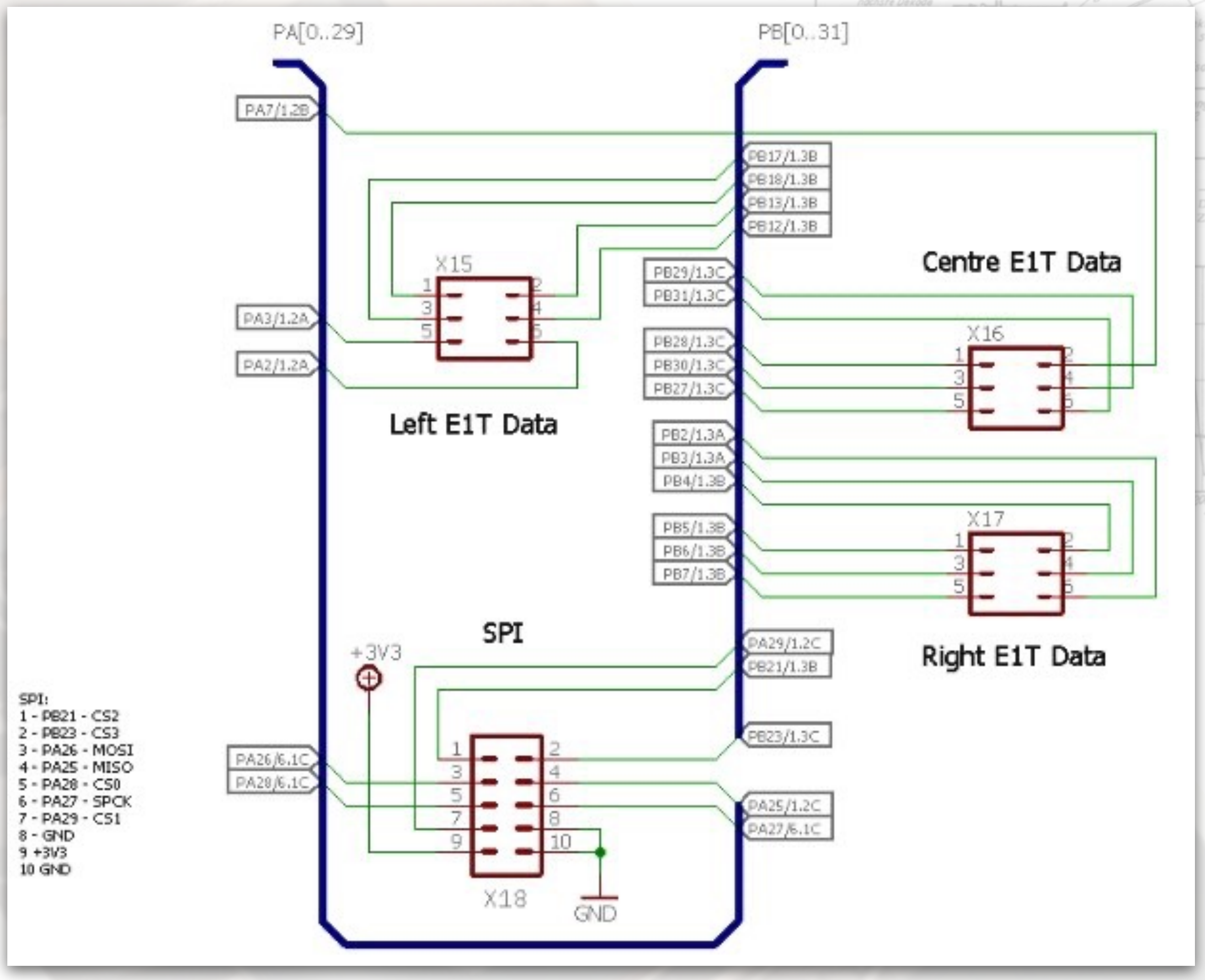


These transfer +300V, +5V and 6.3V to the display board.



These transfer the LED module serial data and clock signals and return the LDR resistance to ground.

 Copyright Grahame Marsh/Nick Stock 2019

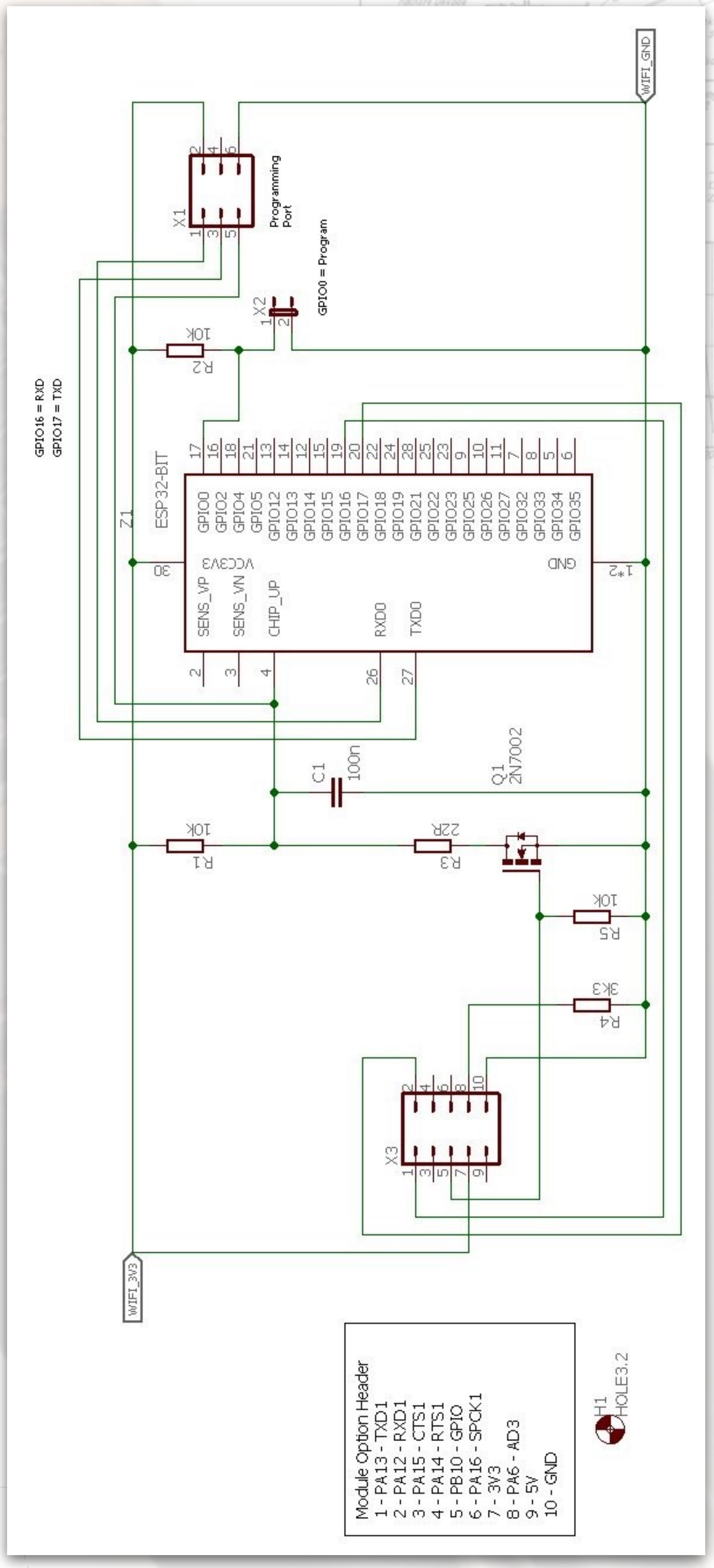


Each E1T requires three signal lines, one to switch the 300V supply, one to reset the tube to zero and one to step the tube by one. Therefore eighteen control lines are required which are transferred using the three 2x3 headers. Finally, the SPI bus is made available for expansion purposes.

 Copyright Grahame Marsh/Nick Stock 2019

3.16 WiFi Plug In

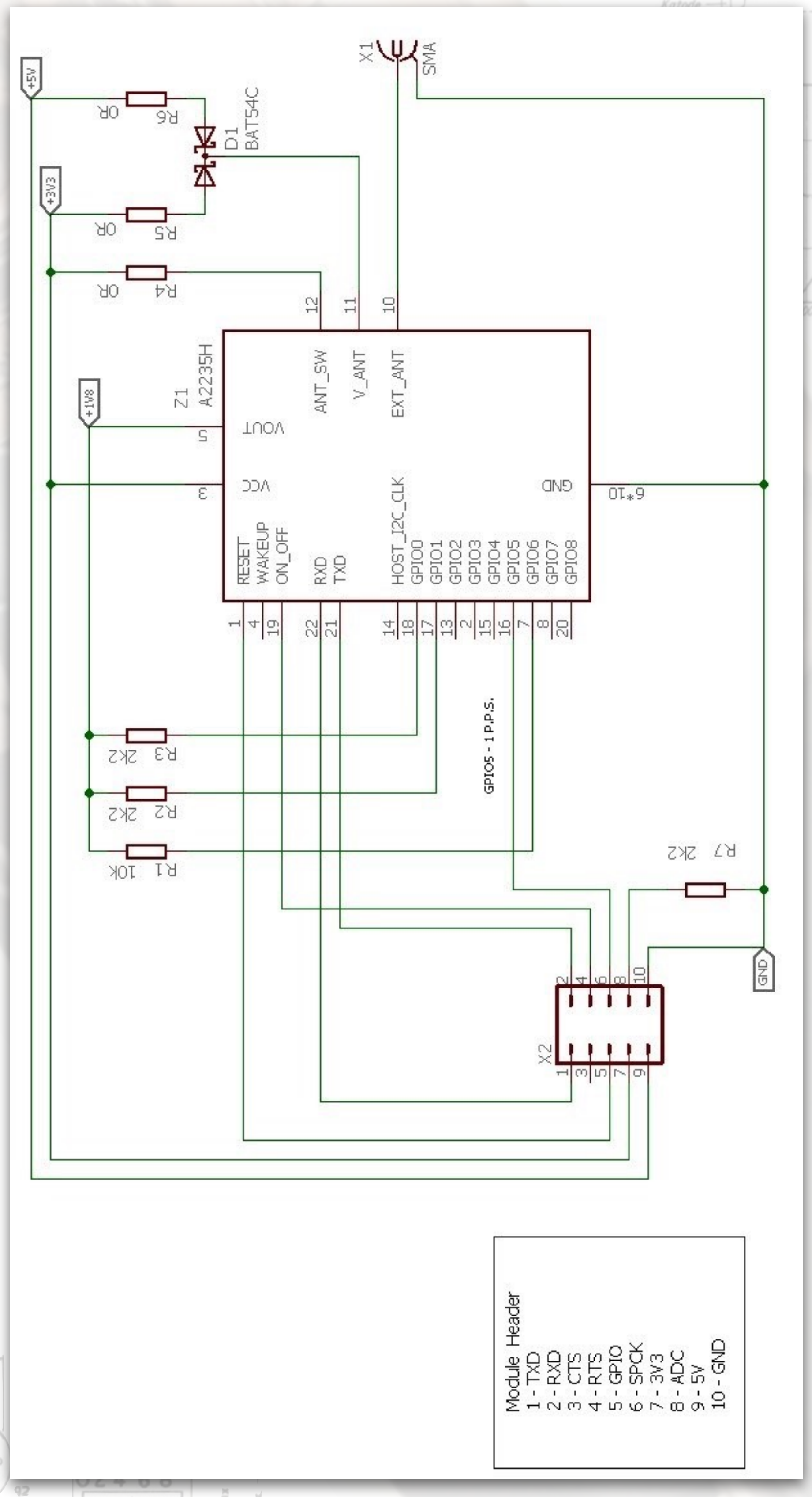
This uses an ESP32-BIT module. Serial data is transferred to the SAM3X8C. The module can be reset using Q1. Programming the ESP32 is accomplished using the SAM programmer and fitting the program jumper.



 Copyright Grahame Marsh/Nick Stock 2019

3.17 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.



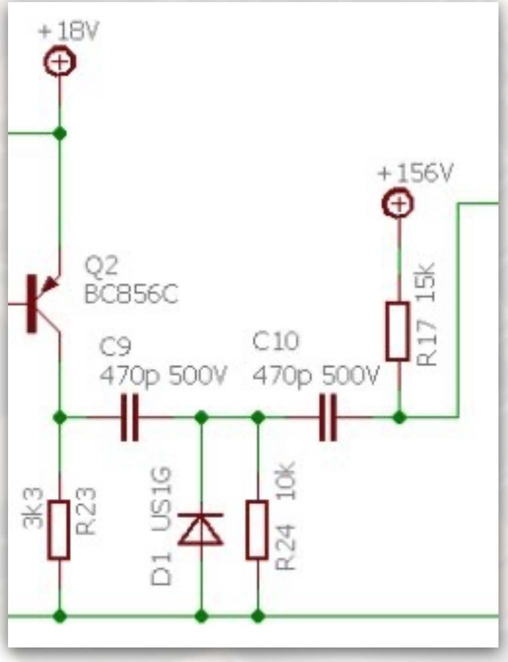
 Copyright Grahame Marsh/Nick Stock 2019

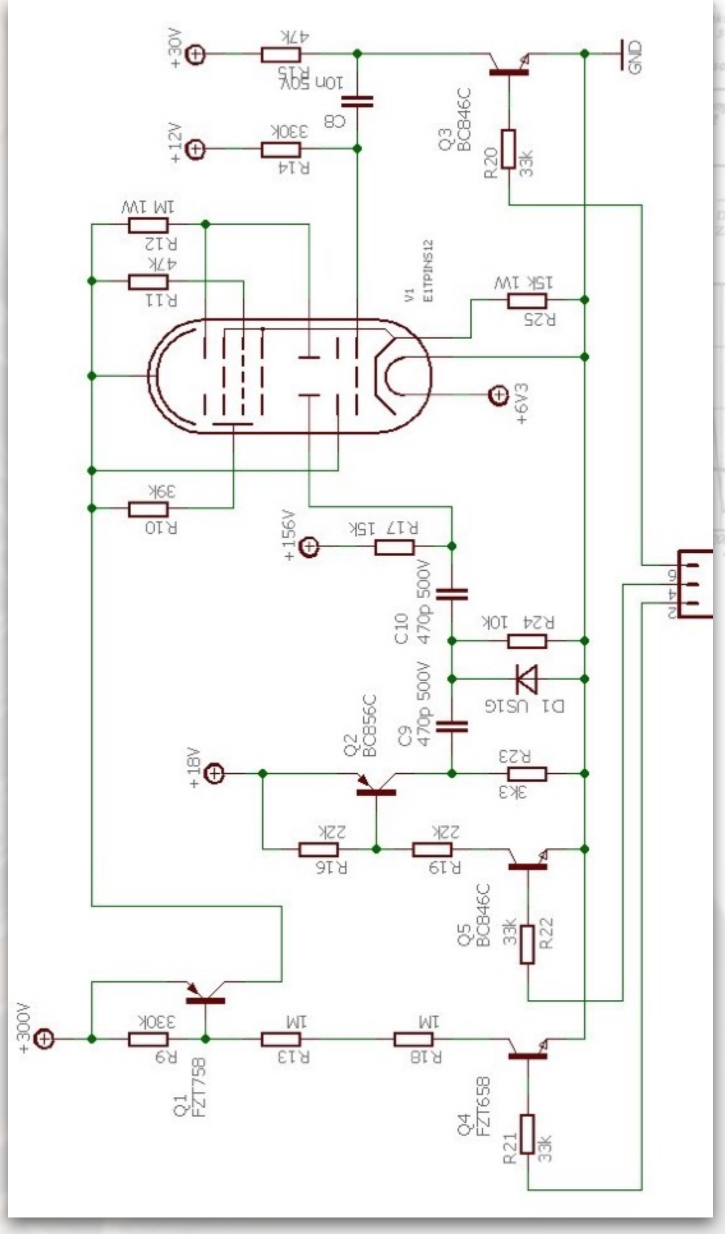
4.0 Display Board

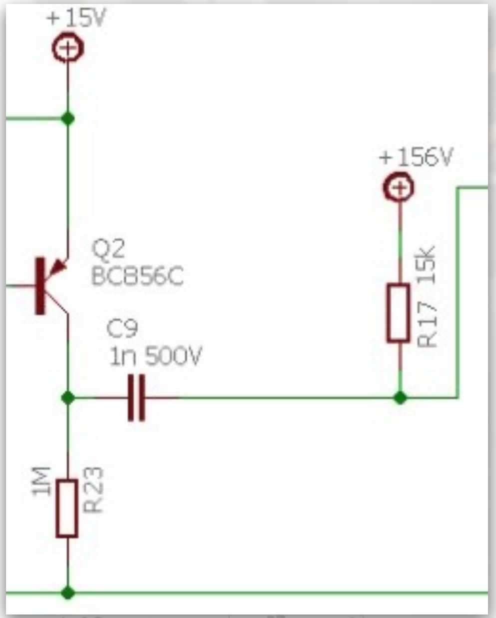
4.1 E1T Drivers

The six E1T tubes have identical drivers. There are three 2x3 headers that transfer the signals from the SAM3X8C.

The schematic below shows Dieter Wächter's step pulse circuit.[82]





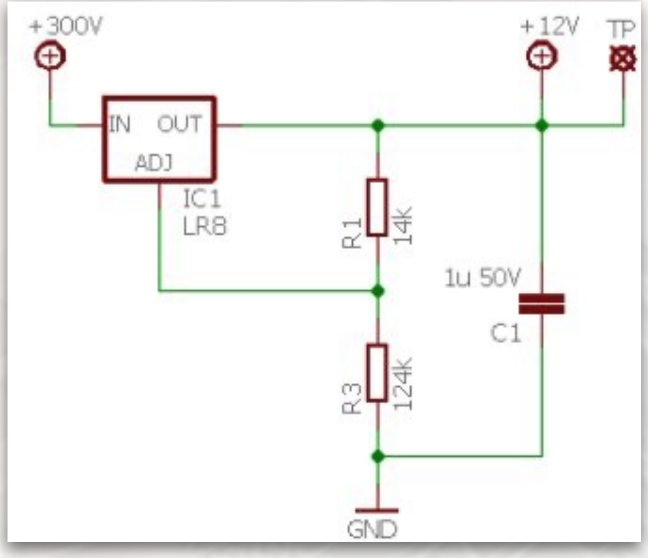


By omitting a few components, changing a few and decreasing the step voltage then the same PCB can use the step pulse circuit by Ron Dekker[83]. In practice a 0R resistor is substituted for C10.

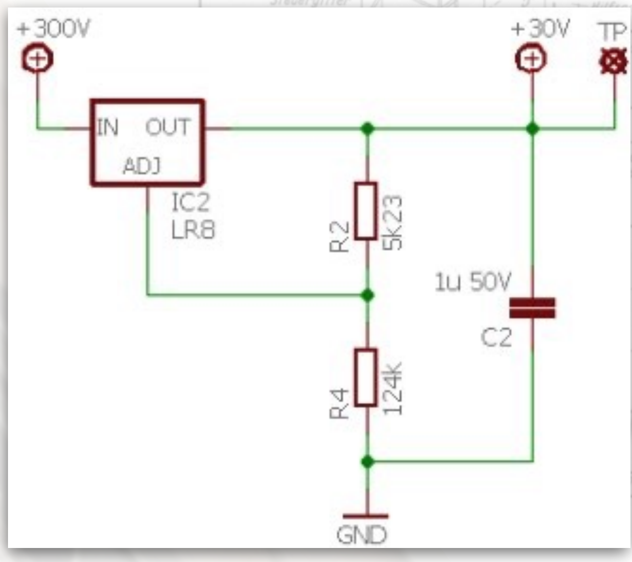
[82] http://www.tube-tester.com/sites/nixie/different/e1t-clock/e1t.htm
[83] http://www.dos4ever.com/E1T_clock/E1T.html

 Copyright Grahame Marsh/Nick Stock 2019

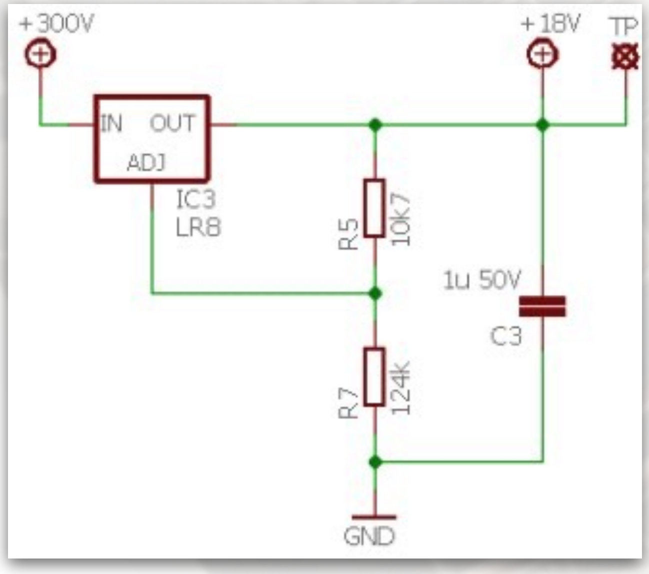
4.2 +12V Regulator



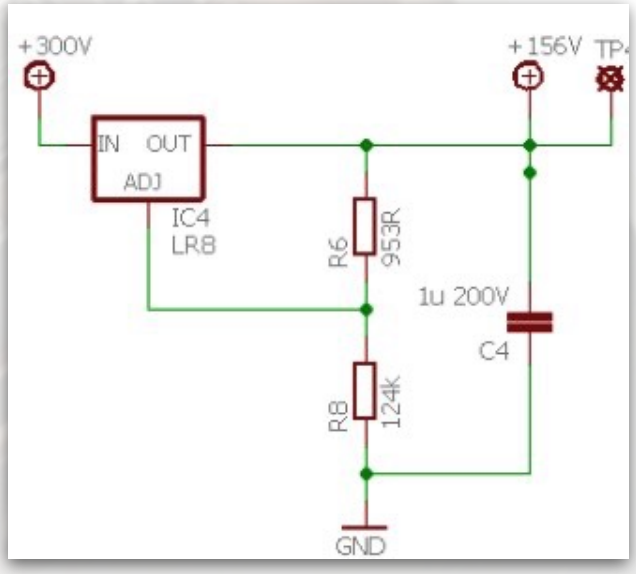
4.3 +30V Regulator



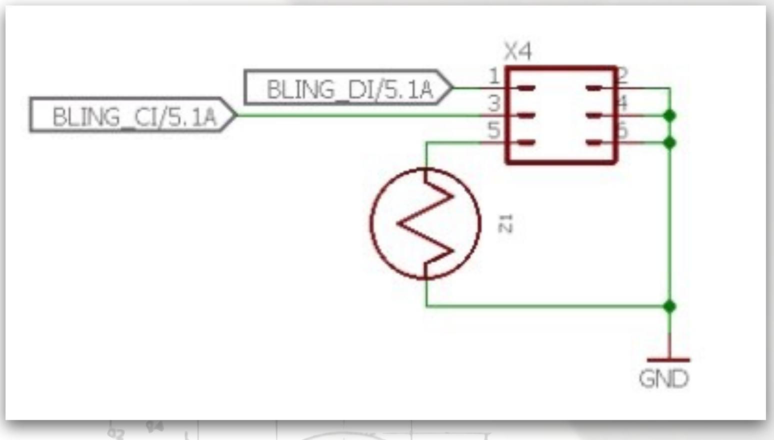
4.4 +15V/18V Regulator



4.5 +156V Regulator



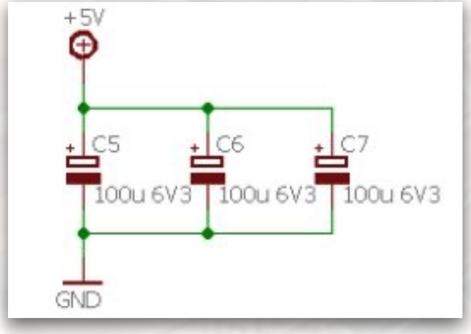
4.6 Bling Data and Clock, LDR Connector

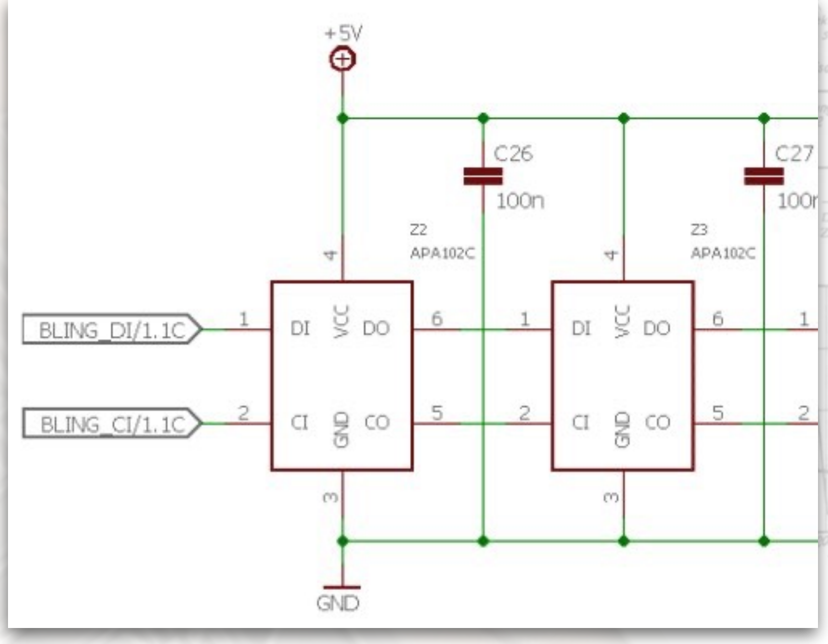


Copyright Grahame Marsh/Nick Stock 2019

4.7 Bling-Bling LED RGB Modules

A total of twenty six LED modules are used; just two are shown here. The E1T tubes are lit with four apiece and the remaining two are used as colon separators.





Additional 5V power storage is provided along the 5V PCB traces.

Copyright Grahame Marsh/Nick Stock 2019

E1T Tube - How 1T Works[84]

A Miniature CRT In Action

[84] Get it? Ha ha ha..

 Copyright Marsh/Stock 2019-2022

An Extract From Dance's "Electronic Counting Circuits"

Chapter 5 - E1T Decade Counting Circuits

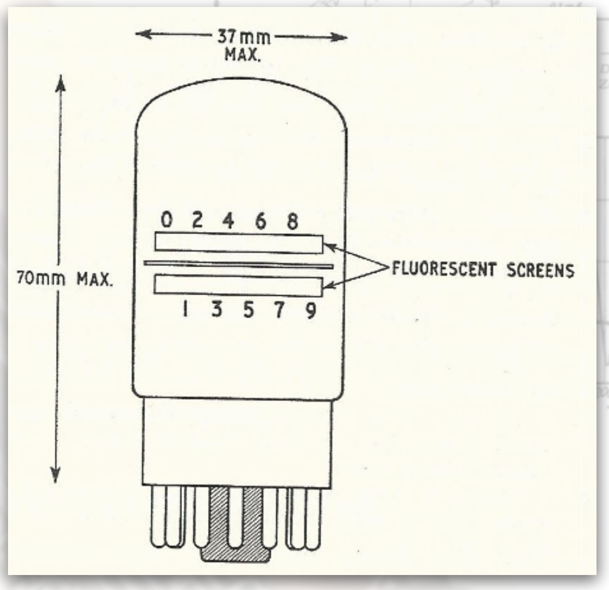


Fig 5.1 - The E1T Decade Counting Tube

The principle of operation of the Mullard/Philips E1T decade counter tube is fundamentally different from that of all other types of counting tube. The E1T is a high vacuum tube which has been especially designed for counting purposes; it has an indirectly heated cathode. The E1T is basically a small cathode ray tube of special design without any vertical deflecting plates and of about the same size as an octal based radio receiving tube. It has the advantage of being a self indicating device, but it cannot easily be used to control digital indicator tubes because the same electrodes are always being employed in the E1T whatever the state of the count. The method of readout is unique. An H.T. supply of 300V is adequate for most E1T circuits. The E1T is not a gas filled device and, therefore, its maximum operating speed is not limited by ionisation and de-ionisation times. All E1T tubes can operate at counting speeds up to at least 30,000 pulses per second, but about 75 % of all the tubes can be used in slightly more complicated circuits for counting at frequencies up to 100,000 pulses per second. Operation at frequencies of the order of one million pulses per second has been reported. The form and the dimensions of the E1T are shown in Fig 5.1.

5.1 - READOUT

The E1T tube employs a ribbon shaped electron beam of rectangular cross section which has ten stable positions in the tube. A small portion of the beam passes through one of the ten holes in the anode and strikes a fluorescent coating on the inside of the tube envelope so that a vertical green luminescent mark is formed in a position near to the digit which is to be indicated. The ten digits themselves are marked on a paper mask which is fixed to the outside of the tube. The beam advances at one step per input pulse until the digit 'nine' is reached, after which a further input pulse will reset the beam to zero. Even digits are indicated as a mark on the upper strip of fluorescent material and odd digits on the lower strip (see Fig. 5.1); this enables a clearer indication to be obtained than would be possible if only one fluorescent strip were used to indicate all ten digits. The beam itself is not deflected vertically in order to enable it to strike the appropriate fluorescent strip, but is merely deflected horizontally across the tube. There are holes placed alternately in the upper and lower parts of the anode; when the beam passes through one of the upper holes an even digit is indicated, but at the next step it will pass through one of the lower holes to indicate an odd digit. Only a small portion of the beam passes through a hole, the remainder of the beam being intercepted by the anode.

5.1.1 - The Electrodes of the E1T

In order to show that the tube has ten stable positions, the somewhat complicated electrode structure of the tube (shown in cross section in Fig. 5.2) must be studied. The conventional

 Copyright Marsh/Stock 2019-2022

symbol for the tube, as used in circuits, is shown in Fig. 5.3 with the connections to the B12A base.

Some of the less important electrodes which have no external connection are not shown in this symbol. Sometimes g_3 and g_5 are also omitted from the symbol. The electron beam is formed at the rectangular shaped cathode, k, the front of which is covered with an electron emissive oxide coating. The beam flows through the control grid, g_1, past the beam forming electrodes, b, and is then accelerated through the electrode g_2. These electrodes focus the beam and also give it the desired rectangular cross section which resembles a piece of thick ribbon placed in a vertical plane. The beam is then deflected by the deflector plates, x' and x", into one of the ten stable positions. The electrodes g_3 and g_5 are suppressor grids which are internally connected to the cathode to prevent any unwanted effects which might be caused by secondary electron emission from g_4 or from the anode, a_2. The electrode g_4 has slots of the shape shown in Fig. 5.4. As will be shown later, it can be arranged that the electron beam will be stable only when a certain fraction of it is passing through one of the vertical rectangular slots in g_4. The purpose of the horizontal slot will be discussed later. The beam then travels to the anode, a_2. A portion of it passes through the anode to the fluorescent target, t. This target is covered with a conductive coating which is connected to the positive H.T. supply line so as to prevent the accumulation of negative charge from the electron beam which might disturb the operation of the tube. The electrode a_l is the reset anode. When the tube is indicating the digit 'nine' and a further pulse is received, the beam is deflected by the plates x' and x" so that it strikes the reset anode; the mechanism by which the tube is reset is then initiated by the fall of the reset anode potential. The auxiliary anode, aux, is internally connected to the accelerating electrode, g_2, and is employed to capture undesired stray electrons. The screen s is internally connected to the cathode.

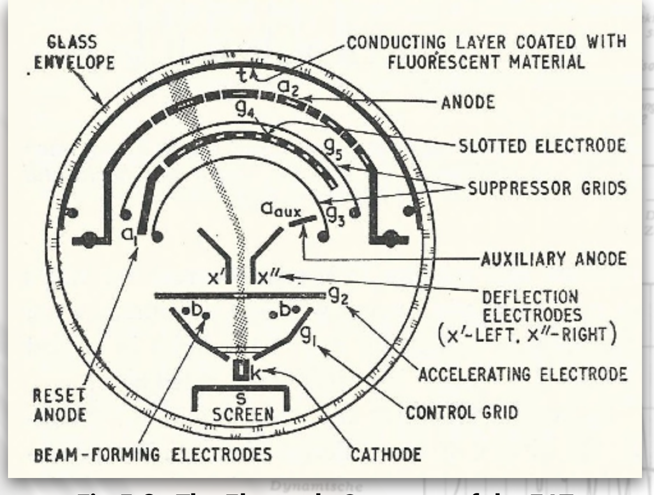


Fig 5.2 - The Electrode Structure of the E1T

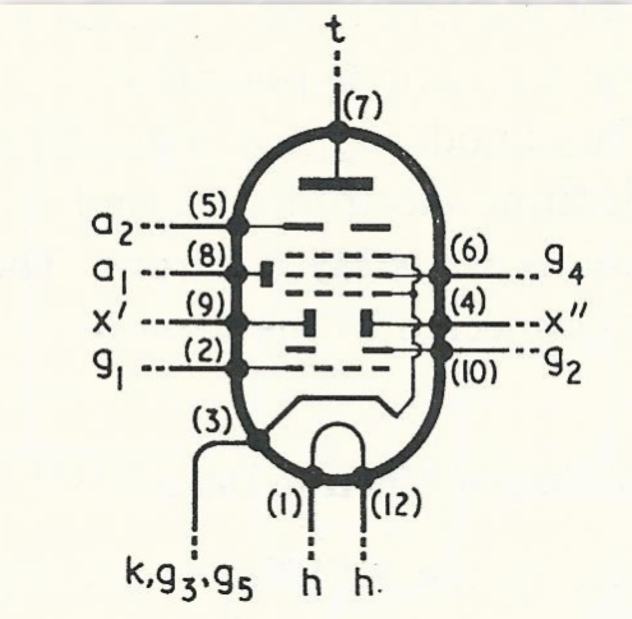


Fig 5.3 - The Symbol for the E1T; the Numbers Indicate the Base Connections

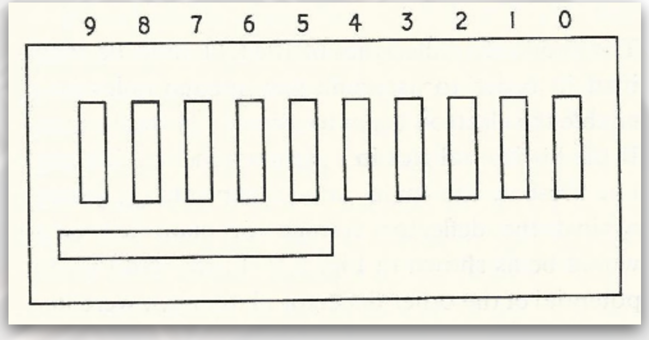


Fig 5.4 - The g4 Electrode Showing the One Horizontal and Ten Vertical Slots

Copyright Marsh/Stock 2019-2022

5.1.2 - Ribbon Shaped Electron Beams

In tubes such as the E1T in which the beam is deflected only in one plane, a ribbon shaped electron beam of relatively large cross sectional area can be used (since the resolution in one plane is unimportant), but in normal cathode ray tubes a very small[2] circular beam must be used to obtain good resolution in two dimensions. For a given charge density in the beam and a given applied voltage, a larger current will flow in a ribbon shaped beam than in a small circular beam owing to the larger cross sectional area of the former. A large current is desirable in the E1T so that the stray electrode capacitances can be quickly charged. The ribbon shaped beam enables the tube to operate from fairly small voltages. This favours high operating speeds because the change in the electrode potentials (and hence the change in the charge of the stray capacitances) is kept small. The use of a ribbon shaped beam also has the additional advantages that the dimensions of the tube (and hence the inter-electrode capacitances) can be small and that the alignment of the tube need be carried out accurately only in one dimension. In the E1T a beam current of about 1 mA is used at an applied potential of about 300V.

5.2 - ANODE CHARACTERISTICS

The anode characteristics of the E1T must be examined in order to ascertain why the ten holes in g_4 enable the electron beam to exist in ten stable states. If the horizontal slot in g_4 (shown in Fig. 5.4) were not present, the main anode current I_{a2} plotted against the deflector voltage of plate x" ($V_{x''}$) would be as shown in Fig. 5.5 (a) provided that the potential of the other deflector plate, $V_{x'}$ were kept constant. When the potential of x" is altered, the beam is deflected and passes through a series of maxima and minima as it passes across the holes in g_4. The anode current will be a maximum when the beam is centred on one of the holes in g_4 and will be zero when it is entirely intercepted by g_4. In normal operation the main anode, a_2, is connected directly to the deflector plate x". The potentials of x" and of a_2 are therefore identical and Fig. 5.5(a) is the dynamic anode current/anode voltage characteristic for this method of connection when the potential of x' is constant. The presence of the horizontal slot in Fig. 5.4 changes the anode characteristic from that shown in Fig. 5.5(a) to that shown in Fig. 5.5(b). When the beam is in a position to the left of the fifth vertical slot in Fig. 5.4, a constant current passes through the horizontal slot and this current is super-imposed on any current which may pass through one of the vertical slots. Hence the shape of the Fig. 5.5(b) characteristic. In practice the characteristic is further modified by the fact that the slots in g_4 are not of constant width. The actual E1T anode characteristic is shown in Fig. 5.6 for the case when the x' deflector electrode has a potential of 156 V. It may be noted that when both of the deflector electrodes have the same potential ($V_{x'} = V_{x''} = V_{a2} = 156V$), the beam is not deflected and the tube indicates a number in about the middle of the decade.

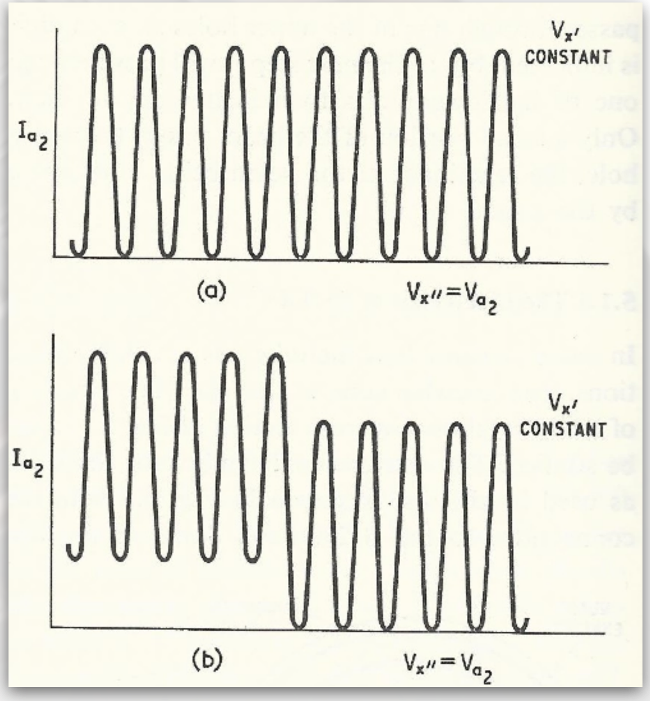


Fig 5.5 - Theoretical Characteristics of the EIT, (a) When g4 Has No Horizontal Slot and (b) When the Horizontal Slot is Present in g4

 Copyright Marsh/Stock 2019-2022

5.3 - BEAM STABILITY

The basic type of circuit used to supply voltages to the tube is shown in Fig. 5.7. The anode resistor, R_{a2} normally has a value of 1MΩ. The straight line in Fig. 5.6 is the load line for this value of resistor. If the beam is initially at the position a of Fig. 5.6 (indicating the digit zero) and the potential of the deflector electrode x' is increased relatively slowly (so slowly that the effect of the stray capacitance, C, shown in Fig. 5.7 is negligible), the beam will tend to be deflected towards the electrode x'. As the beam moves, however, it can be seen from the anode characteristic of Fig. 5.6 that it begins to pass out of the slot in g_4 and less of it strikes the anode. The resulting reduction in anode current leads to a reduction in the voltage dropped across the resistor R_a, and hence to an increase in the common potential of the anode and of the deflector electrode x". The slope of the E1T characteristic is very steep at the points where it is crossed by the load line shown and therefore this increase in the potential of x" is almost equal to the initial increase in the potential of x' which caused it. As both deflector electrodes are increased in potential by almost equal amounts, the amount by which the beam is deflected is virtually unchanged. Similarly if the beam is at a and x' becomes slowly more negative, the anode current is increased (see Fig. 5.6) and this in turn causes a reduction in the potential of x". Thus the position a in Fig. 5.6 is a very stable one. The intersections of the load line with the rising parts of the E1T characteristic are the ten stable beam positions which are required for storing the information about the state of the count. The anode a_2 and the x" deflector plate are connected in a feedback system. The slope of the E1T characteristic is very much greater than the slope of the 1 MS/ load line at the operating point and this results in the feedback factor - and hence the stability of the operating point - being very high. The positions a, c, e, g, i, etc. in Fig. 5.6 are all very stable.

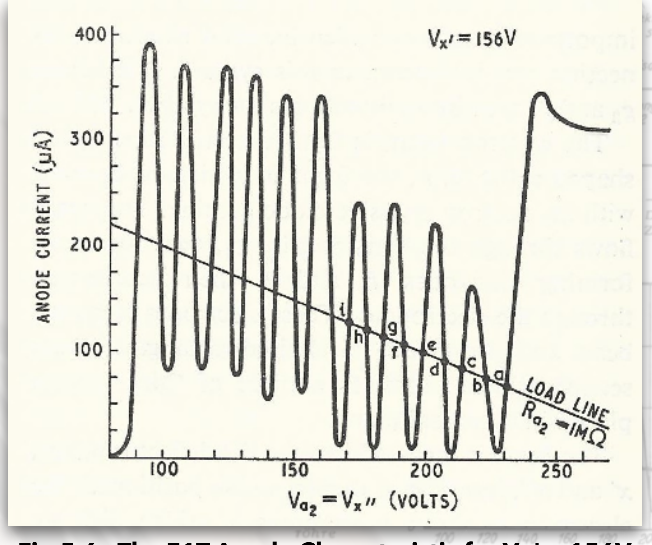


Fig 5.6 - The E1T Anode Characteristic for Vx' = 156V

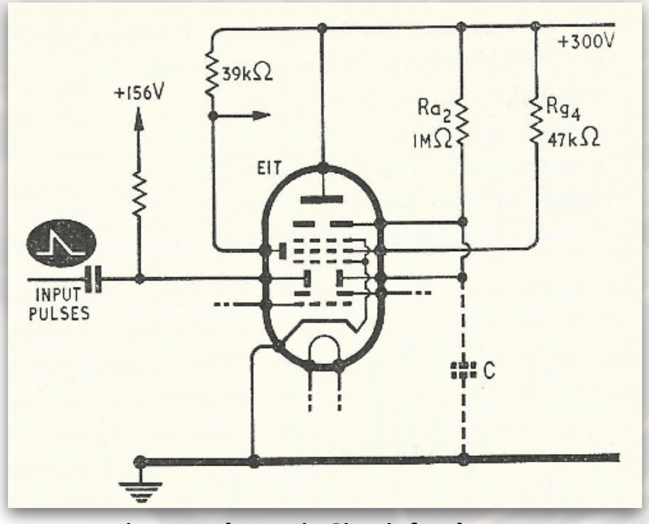


Fig 5.7 - The Basic Circuit for the E1T

If the beam is at any moment at b or d, any slight increase in the potential of x' will cause the beam to be deflected towards this electrode and it can be seen from Fig. 5.6 that the anode current will then increase as more of the beam passes through the slot in g_4. The potential of the anode and of x" therefore decreases causing the beam to swing farther away from the x" electrode. Eventually the beam will come to rest at one of the stable points c or e. Similarly if the beam is momentarily at b or d and the potential of x' is decreased slightly, the beam will move so that the voltage of x" becomes higher until a stable operating point is reached. The positions b, d, f, etc. are therefore unstable and the beam does not stay in a position represented by one of these points for more than a minute fraction of a second.

 Copyright Marsh/Stock 2019-2022

At these unstable points the anode current decreases with increasing anode voltage, thus giving a negative resistance effect over this portion of the curve. The criterion of stability for any operating point in Fig. 5.6 is that the anode current of the E1T must increase as the anode voltage increases. That is, the point at which the load line cuts the characteristic of the tube is stable if the characteristic at that point slopes upwards from left to right. If for any reason (such as a change in the supply voltage) the potential of x' alters fairly slowly, the anode current/anode voltage characteristic will maintain the same general form as shown in Fig. 5.6, but will be moved horizontally along the x axis (anode voltage axis) of the graph. This is because the stabilising effect discussed above alters the voltage of the anode and x" electrodes to maintain the beam deflection almost constant. Fig. 5.8 shows the anode characteristic for an EIT with a potential of 170V applied to the x' deflector electrode. It can be seen that the same system of stable and unstable operating points will be present and the general operation of the tube is unaffected by this voltage change.

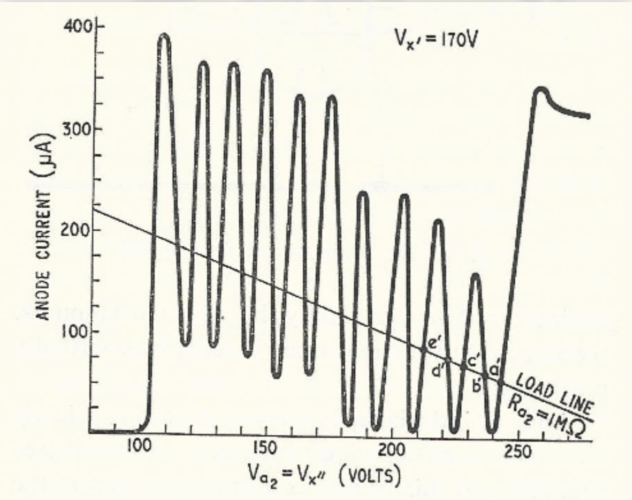


Fig 5.8 - The E1T Anode Characteristic for Vx' = 170V

5.4 THE COUNTING PROCESS

A very different process occurs when a positive going pulse with a very short rise time is fed to the x' electrode. The beam will be deflected to the left and the potential of the anode and x" electrode will again tend to rise by the process discussed previously. The capacitance C (shown dotted in Fig. 5.7) prevents any very rapid change in the potential of the anode and of the electrode x", as time is taken for C to charge through R_a. The capacitance C is merely the inter-electrode and stray wiring capacitance of the tube circuit. The beam is therefore deflected to the left before the voltage of x" has time to rise appreciably. If the pulse is rapid enough and of a suitable amplitude, the beam will therefore move to the next stable position to the left of the initial position in Fig. 5.6 and a count will have been registered. The stabilising mechanism of the tube circuit cannot work more quickly than is permitted by the anode resistance R_a and the unavoidable stray parallel capacitance, C. The pulse rise time and amplitude are quite critical. If the pulse is of too small an amplitude, the beam will not be deflected as far as the next stable position and no count will be registered, whilst if the amplitude is too large, the beam may pass through one stable position and register two counts for only one input to x'. The amplitude of the input pulse should be approximately equal to the difference of the tube anode voltage between two adjacent working points, e.g. a and c in Fig. 5.6. The geometry of the tube and the shape of the electrodes are carefully chosen so that the voltage difference between each of the stable working points (a to c, c to e, etc. in Fig. 5.6) is constant (about 13.6 V). The input voltage required to cause the tube to register one additional count is therefore independent of the digit being indicated. It is most important that the input pulse amplitude to the x' plate of the tube should be 13.6V ± 15% (that is, 11.5 to 15.5V). An additional requirement is that the trailing edge of the pulse must not be too sharp or it will deflect the electron beam back to its initial state and no count will be registered. If the slope of the trailing edge is not very great, the stabilising effect discussed previously will prevent the tube returning to its initial state when the trailing edge is applied to x'. If the time of fall of the pulse is too long, however, the maximum counting speed is reduced. It might be thought that if the stray capacitance, C, could be made very small, the maximum counting rate could be increased. In actual practice, however, the reset time is usually

 Copyright Marsh/Stock 2019-2022

longer than the counting process it-self and sets a limit to the maximum counting speed. A suitable pulse for feeding into the x' electrode of the E1T is shown in Fig. 5.9. The slope of the leading edge of the pulse should not be less than 2 x 10^7V/sec and that of the trailing edge should not be greater than 1.2 x 10^6V/sec. If the average amplitude of the pulse is to be 13.6V, the rise time should not therefore be greater than 0.7 μsec and the time of fall should not be less than 11 μsec.

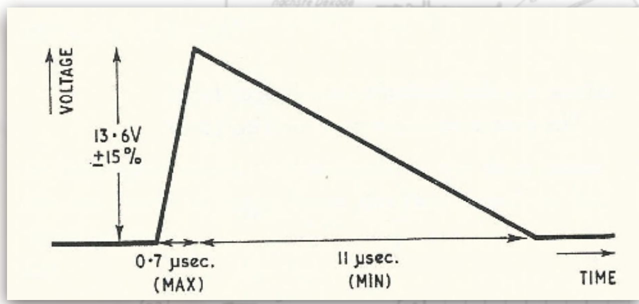


Fig 5.9 - An Input Pulse Suitable for the Operation of the E1T

The mechanism of the counting process can be considered to operate in the following way. If the operating point is at a in Fig. 5.6 corresponding to an indication of zero, the anode and x" potential is about 230V whilst the x' deflector electrode potential is about 156V. If a fast rising positive going pulse of 14V is applied to the x' electrode (raising its potential to 170V), the voltage of the x" electrode remains constant for a very small fraction of a second owing to the stabilising effect of the capacitance C. The operating point is therefore momentarily moved to the point c' on the characteristic of Fig. 5.8. The pulse then decays slowly so that the potentials of x' and x" decrease at about the same rate. Thus the operating point on the characteristic of Fig. 5.8 at c' is transformed relatively slowly into the characteristic of Fig. 5.6, but the operating point now has time to move along with the curve and finishes at c in Fig. 5.6.

It can be seen from Fig. 5.6 that the horizontal slot in g_4 (see Fig. 5.4) lifts up the low voltage part of the anode current/anode voltage characteristic so that the height of each peak above the load line is fairly constant. The rate at which the stray capacitance, C, can be discharged by the E1T anode current during counting operations is dependent on the height of each peak of the characteristic above the load line. A reasonable height for each peak is essential in high speed counting circuits. This subject is more fully discussed in the section of this chapter which deals with the design of an input circuit for 100 kc/s operation. The stabilising effect of C on the anode potential should not be confused with the stabilising effect that R_{a2} has on the position of the beam. Advantage is taken of the latter effect (which is suppressed during the steep front of the input pulse by the presence of C) for maintaining the beam at the correct position after it has been displaced.

5.5 - FLYBACK CIRCUITS

When the tenth input pulse is received, the E1T tube must be reset from 'nine' to 'zero'. Normally this resetting process is initiated by a pulse from the reset anode, al, which is connected to the H.T. positive line via a 39kΩ resistor (as in Fig. 5.7). If the tube is initially indicating the digit 'nine' and an additional input pulse is received, the beam will be deflected to strike the reset anode. The current passing to this anode will cause a voltage drop across the 39kΩ resistor and a negative pulse can therefore be obtained from the reset anode. The pulse may be used to trigger a monostable multi-vibrator which is designed to provide suitable pulses to reset the tube and also to trigger the next decade. Another method of obtaining a pulse to reset the E1T circuit does not depend on the use of a reset anode. When the beam is deflected from position 'nine' onto the reset anode, it leaves the g_4 electrode. This electrode is fed from the H.T. line via the resistor R_{g4} and its potential therefore rises as the current through the resistor falls. This rise in potential can be used to render a triode conducting and the triode in turn provides a pulse to cut off the E1T. The E1T itself may be reset by two basic methods. In the first method a negative pulse is applied to the control grid, g_1 or a positive pulse to the cathode, k. This pulse should have an

 Copyright Marsh/Stock 2019-2022

amplitude of at least 24V so that it is large enough to completely cut off the electron beam. The main anode current falls and therefore the main anode and x" electrode potential rises. The change of the x" electrode potential causes the beam to be deflected towards it so that 'zero' is indicated. This method of resetting the tube takes a comparatively long time and cannot therefore be used in high speed circuits. The circuitry required is, however, simpler than that used in the higher speed resetting circuits. An example of a practical circuit involving beam cut off will be given in the circuit of Figs. 5.10. In the second method of resetting the tube, a positive pulse is applied to the x' electrode and deflects the beam to the zero position. This method is suitable for high speed circuits operating at up to one million pulses per second.

5.5.1 - Reset Involving Beam Cut Off

When an E1T tube is cut off, its anode voltage will rise exponentially as the stray capacitance C (shown dotted in Fig. 5.7) charges through the resistor R_a. The time taken for this capacitance to charge limits the maximum frequency of operation of the tube. The minimum reset time may be estimated by the method discussed below. It is important to ensure that the duration of the cut off pulse fed to the tube is great enough (with an adequate safety margin) to allow the stray capacitance, C, to charge to a potential which is enough to cause the beam to return at least as far as the zero position. Otherwise the beam may come to rest at any intermediate position. If the cutoff time is too long, however, the reset time will be increased and the maximum counting rate will be reduced. If the beam is deflected too far, it will be in an unstable state and will quickly return to the zero position at the end of the cutoff pulse.

It can be estimated from Fig. 5.6 (allowing adequate safety margins for normal tolerances, etc.) that the maximum voltage swing of the anode a2 ever likely to occur in practice is from $V_{a2}(9)$ = 95V in position 'nine' to $V_{a2}(0)$ = 240V at the 'zero' position. The maximum stray capacitance, C, in parallel with R_{a2} can be taken as 16.5 pF. If a close tolerance 1 % high stability resistor is used for R_{a2}, the maximum possible value of this resistor will be 1.01MΩ. In addition a 10kΩ resistor is normally placed in series with R_{a2} for test purposes (as shown in Figs. 5.13 and 5.15). The maximum value of R_{a2} is therefore 1.02MΩ. The capacitance C charges from the H.T. supply voltage V_b from the initial anode voltage of $V_{a2}(9)$ volts to $V_{a2}(0)$ volts during the cut off pulse. It is shown in many elementary text books on electricity that if a capacitor C is charged from a source of voltage V_b via a resistor R, the voltage V across the capacitor after a time t is given by the relation:

$$V = V_b(1 - e^{\frac{-t}{RC}})$$

where e is the base of natural logarithms. The above equation may be altered to:

$$\frac{V_b - V}{V} = e^{\frac{-t}{RC}}$$

This equation applies only if V= 0 when t = 0. In the case of the stray capacitance C charging through the resistor R_{a2}, however, V= $V_{a2}(9)$ initially. If C had charged to a potential of $V_{a2}(9)$ from an initial potential of zero through Ice the time taken, t_1, would be given by:

$$\frac{V_b - V_{a2}(9)}{V} = e^{\frac{-t_1}{R_{a2}C}} \qquad (1)$$

Copyright Marsh/Stock 2019-2022

If t_2 is the total time taken for the potential across the capacitance C to reach the value $V_{a2}(0)$ from an initial value of zero,

$$\frac{V_b - V_{a2}(0)}{V} = e^{\frac{-t_2}{R_{a2}C}}$$

Dividing (2) by (1):

$$\frac{V_b - V_{a2}(0)}{V_b - V_{a2}(9)} = e^{\frac{-t_2 - t_1}{R_{a2}C}}$$

In this equation $(t_2 - t_1)$ is equal to the time taken for the beam to move from position 'nine' to the `zero' position. Let $(t_2 - t_1) = T$

$$\frac{V_b - V_{a2}(0)}{V_b - V_{a2}(9)} = e^{\frac{-T}{R_{a2}C}}$$

Putting the values quoted above into this equation:-

$$\frac{300 - 240}{300 - 95} = e^{\frac{-T}{1.02 \times 10^6 \times 16.5 \times 10^{12}}}$$

When this equation is solved for T, it is found to be about 20.68 µsec. This is the minimum possible resetting time. In actual practice the resetting pulse should be somewhat longer than this in order to allow an adequate margin of safety. If an allowance of 33 µsec is made for the resetting time, the maximum counting rate which can be attained is about 30,000 per second. It is found in actual practice that E1T circuits can operate reliably at up to 30,000 pulses per second when the resulting operation is carried out by cutting off the electron beam in the tube in the type of circuit shown in Fig. 5.10. In practical circuits the stray capacitance C should be kept as low as possible.

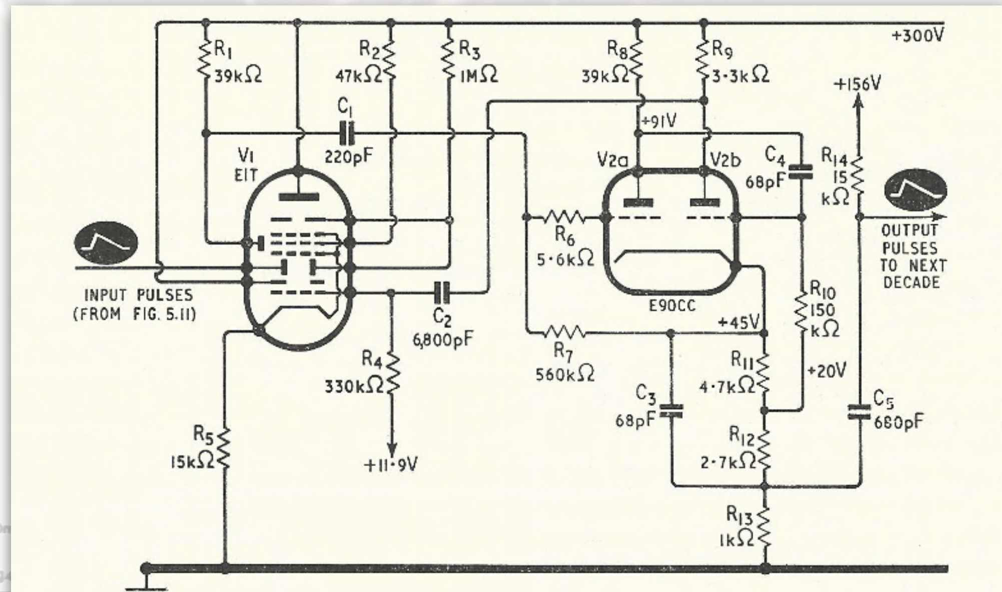


Fig 5.10 - An E1T Counting and Reset Circuit for Operation at Frequencies up to 30 kc/s

 Copyright Marsh/Stock 2019-2022

6370		E1T

S.Q. TUBE

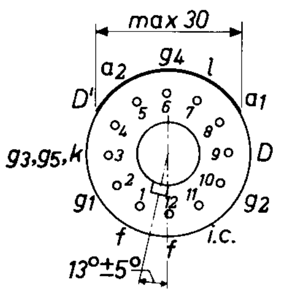
Special quality decade counter tube.

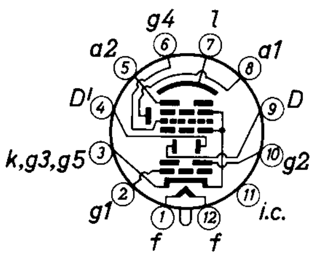
QUICK REFERENCE DATA			
Life test	10 000 hours		
Base	Duodecal (12 pins)		
Heating	Indirect A.C. or D.C.; Series or parallel supply		
Heater voltage	V_f	6.3	V
Heater current	I_f	300	mA

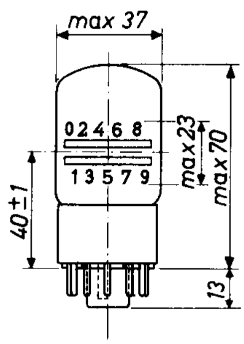
DIMENSIONS AND CONNECTIONS

Dimensions in mm

Base: Duodecal







APPLICATION DIRECTIONS

Mounting

Any mounting position, except horizontal with screen down, is permitted.

Sensitivity to magnetic fields

To prevent interference by magnetic fields the flux density of these fields should not exceed $2x10^{-4}$ Wb/m^2 (= 2 Gauss) in any direction.

December 1968		1

Copyright Marsh/Stock 2019-2022

APPLICATION DIRECTIONS

Ambient illumination

To obtain a clair reading the ambient illumination should range from 40-400 lux measured with an illumination-meter placed in vertical position. This illumination range incorporates the best compromise between the visibility of the figures of the mask and the luminescent picture.

CHARACTERISTICS

Heater voltage	V_f	6.3	V
Heater current	I_f	300	mA

CAPACITANCES

Anode No.2 to all other electrodes	$C_{a_2/R}$	10.5	pF
Deflection plate to all other electrodes	$C_{D/R}$	3.5	pF
Deflection plate to all other electrodes	$C_{D'/R}$	3.8	pF
Anode No.1 to all other electrodes	$C_{a_1/R}$	4.9	pF
Grid No.1 to all other electrodes	$C_{g_1/R}$	6.8	pF
Grid No.4 to all other electrodes	$C_{g_4/R}$	7.7	pF

OPERATING CHARACTERISTICS

Column I Nominal value
II Permitted values of spread and variation

		I	II	
Supply voltage	V_b	300		V
Grid No.1 supply voltage	V_{bg_1}	11.9	± 0.15	V
Grid No.2 supply voltage	V_{bg_2}	300		V
Deflection plate supply voltage	V_D	156	± 1.5	V
Luminescent screen voltage	V_ℓ	300		V
Cathode current	I_k	0.95		mA
Grid No.2 current	I_{g_2}	0.1		mA
Cathode resistor	R_k	15	± 1%	kΩ
Grid No.4 resistor	R_{g_4}	47	± 5%	kΩ
Anode No.1 resistor	R_{a_1}	39	±10%	kΩ
Anode No.2 resistor	R_{a_2}	1	± 1%	MΩ

Copyright Marsh/Stock 2019-2022

OPERATING CHARACTERISTICS (continued)

Note

The tube should be used in the circuit of fig.2.
Provided the ratio of the supply voltages V_{bg_1} and V_D is strictly maintained the supply voltage V_b is allowed to vary within the range of V_b nom. ±10%.
This condition can be realised by using a voltage divider R_1, R_2, R_3 with 1% precision resistors as indicated in the diagram fig.2.
A max. counting speed of 30 000 count/s can be obtained with this circuit.
The input pulse at D should have a positive value of 13.6 V ±15%. The slope of the leading edge should be at least 20 x 10^6 V/s. The slope of the trailing edge should not exceed 1.2 x 10^6 V/s.

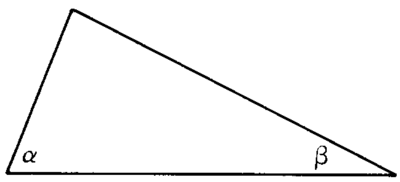
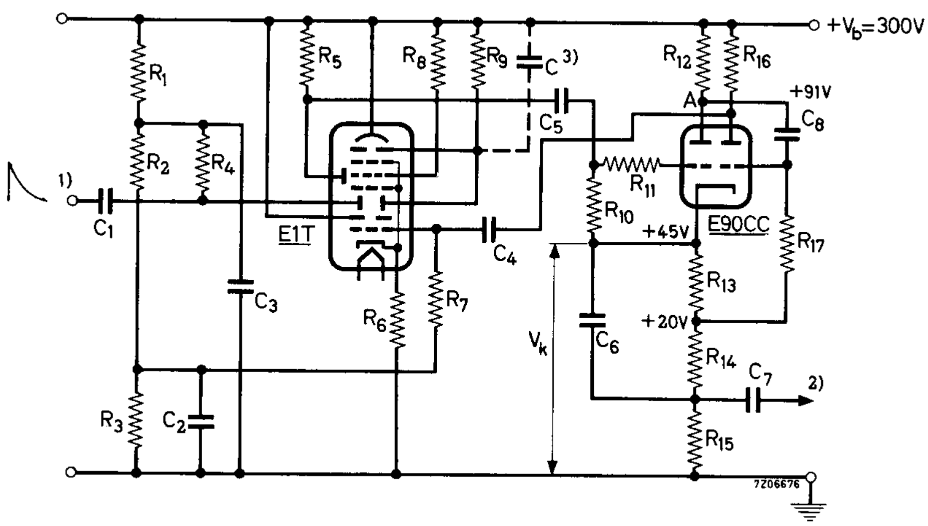


Fig.1

$\tan \alpha > 20 \times 10^6$ V/s
$\tan \beta < 1.2 \times 10.6$ V/s



Copyright Marsh/Stock 2019-2022

R_1	68 kΩ ± 1%	R_{10}	0.56 MΩ ±10%	C_1	1)
R_2	68 kΩ ± 1%	R_{11}	5.6 kΩ ±10%	C_2	0.39 μF ±20%
R_3	5.6 kΩ ± 1%	R_{12}	39 kΩ ± 2%	C_3	0.15 μF ±20%
R_4	15 kΩ ± 2%	R_{13}	4.7 kΩ ± 2%	C_4	6800 pF ±10%
R_5	39 kΩ ±10%	R_{14}	2.7 kΩ ± 2%	C_5	220 pF ±10%
R_6	15 kΩ ± 1%	R_{15}	1 kΩ ± 1%	C_6	68 pF ± 2%
R_7	0.33 MΩ ±10%	R_{16}	3.3 kΩ ± 2%	C_7	680 pF ± 5%
R_8	47 kΩ ± 5%	R_{17}	0.15 MΩ ± 2%	C_8	68 pF ± 2%
R_9	1 MΩ ± 1%				

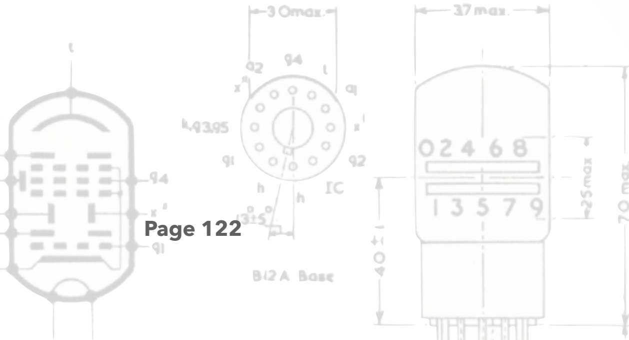
1. Connected to the preceeding E90CC pulse shaper (C_1 = 6800 pF ±10%) or the preceeding E90CC interstage pulse shaper (C_1 = 680 pF ±5%).
2. Connected to deflection plate D of next counter tube.
3. This parasitic capacitance should be reduced to the minimum by keeping the wiring as short as possible.

LIMITING VALUE of supply voltage V_b (See operating characteristics):
V_b = max. 400 V

Copyright Marsh/Stock 2019-2022

Fortress - Nuggle for the E1T

Writing code for your Fortress Clock



1. Introduction

Nuggle[85] supplementary documentation for the Fortress E1T Clock

2. Basic Language Elements

Reserved Words

Reserved words that are new or have a specific meaning for the Fortress E1T clock version of Nuggle are:

Clear, Display, E1T, Flyback, GPIO, Heater, Saver, WriteE1T, WriteDisplay

5. Program Heading and Program Block

System Defined Number Variables

There are no additional system defined number variables for the Fortress E1T Clock.

Statement Part

Fortress E1T Clock specific events are described in section 10.

7. Functions

Fortress E1T Clock specific function designators are as follows:

Standard System Functions

E1T

Syntax: **E1T** *(tube)*

Returns the position of E1T tube. The value will be from 0 to 9 if the tube is glowing and 10 if the tube position is undefined as the HT supply is off.

Heater

Syntax: **Heater**()

Returns the boolean value if the E1T heater supply is on (returns 1) or off (returns 0).

Flyback

Syntax: **Flyback**()

Returns the boolean value if the E1T HT supply is on (returns 1) or off (returns 0).

[85] See the Sgitheach Nuggle Manual for details.

Copyright Marsh/Stock 2019-2022

GPIO

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

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

Pin	Location	SAM3X8C port and pin
0	Pin 2 on connector X1	Port A Pin 10
1	Pin 3 on connector X1	Port A Pin 11

9. Procedures

Display

Syntax: **Display** *tube, value*

This is a synonym for the Fortress E1T procedure and function and can be used where possible to write display independent Nuggle.

E1T

Syntax: **E1T** *tube, value*

Move the glow on the *tube* to the position value. The E1T tubes are number 0 (leftmost) to 9 (rightmost). A value outside this range will turn the HT to the tube off and the tube position becomes undefined. A value of 10 is recommended for this function.

Flyback

Syntax: **Flyback** *state*

Turns the E1T 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
0	Pin 2 on connector X1	Port A Pin 10
1	Pin 3 on connector X1	Port A Pin 11

Heater

Syntax: **Heater** *state*

Turns the E1T Heater supply to state. Where state is on (1) or off (0).

 Copyright Marsh/Stock 2019-2022

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

WriteDisplay

Syntax: WriteDisplay *Str [, …]*

This is a synonym for the Fortress **WriteE1T** procedure and can be used where possible to write display independent Nuggle.

WriteE1T

Syntax: **WriteE1T** *Str, [, ...]*

Writes the string Str to the E1T tubes. The string can contain placeholders for the optional comma separated arguments that follow.

Example:

WriteE1T "012345"

Will light up the six E1T's using this test pattern.

Obviously the E1T tubes are limited to displaying the numbers '0' to '9' so any other character in the string *Str* will turn off 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 turned off.

Examples:

WriteE1T "%02d%02d%02d", hour, minute, second

WriteE1T "%6d", face_curr

WriteE1T "%2d%2d%2d", fw_major, fw_minor, fw_build

For 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.

10. Events

Event Happenings

There are no additional event happenings.

 Copyright Marsh/Stock 2019-2022

Event Default Actions

Fortress E1T 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 E1T tubes
'D'	Turn anticlockwise	Previous clock face is shown
'-'	Press and turn anticlockwise	Chime sound volume is decreased show current setting on the E1T tubes

Copyright Marsh/Stock 2019-2022

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 E1T tubes
'+'	+	Chime sound volume is increased show current setting on the E1T tubes
'-'	-	Chime sound volume is decrease show current setting on the E1T tubes
'>'	|<<	Beep sound volume is increased show current setting on the E1T tubes
'<'	>>|	Beep sound volume is decrease show current setting on the E1T 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.

Copyright Marsh/Stock 2019-2022

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
11	awake_heateronreq	Request to turn on heater PSU	
12	awake_heateron	Heater PSU turned on	heater_on = 15 seconds delay
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
8	sleep_heateroffreq	Request to turn off the heater PSU	
9	sleep_release	Heater PSU turned off 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.

Copyright Marsh/Stock 2019-2022

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.

If the **Default** action takes place at step 5 then the screen saver Nuggle program is run and your program looses 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:

- Clear the display E1T by turning the HT to each tube off.
- Clear all the LEDs to off i.e. black.
- Stop any chime that is playing.
- Stop any beep that is sounding.

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

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. E1T tubes are blanked by turning off each tube's HT supply.
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.

Copyright Marsh/Stock 2019-2022

Annex A System Variables

There are no additional system variables.

 Copyright Marsh/Stock 2019-2022

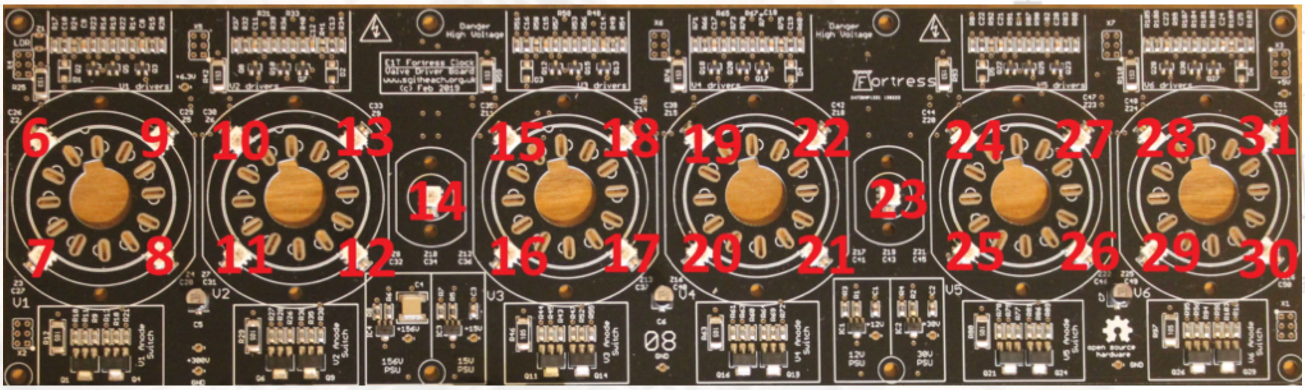
Annex C Fatal Error Messages

There are no additional fatal error messages.

 Copyright Marsh/Stock 2019-2022

Annex D Bling LED Numbering

These images show the position numbers of the bling LED modules as seen from the board silkscreen side. Remember that the main board is mounted upside down so the LED module numbers running 0 to 5 left to right are actually right to left when viewed. Therefore, for example, LED module 0 in underneath tube 6.

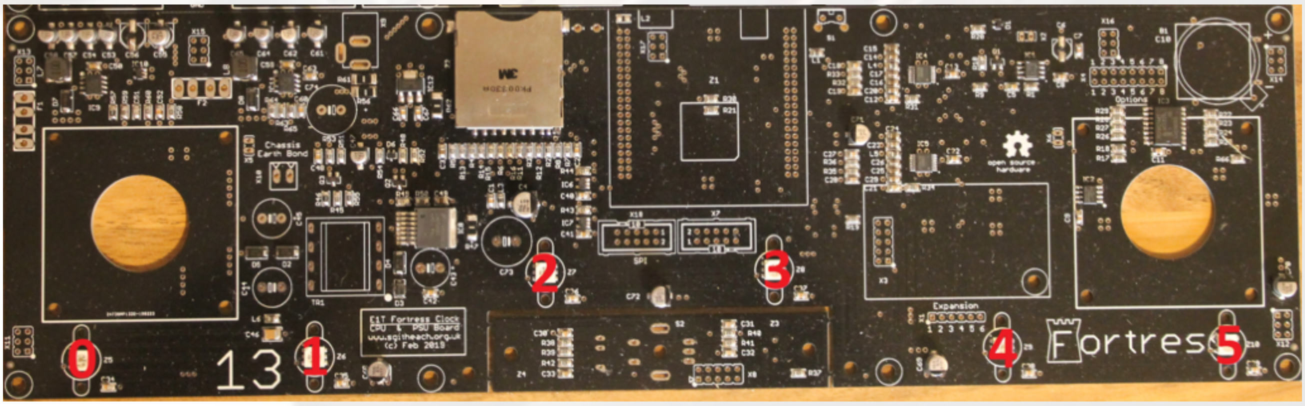


Numbering system for the E1T uplighting LED's and colons LED's

Example of use:

> **Colour** *red*, 14
> **Colour** *red*, 23

will light the two colon LED modules in the colour red.



Numbering system for downward illuminating LED's on the main board

 Copyright Marsh/Stock 2019-2022