

ATARI ST REALTIME CLOCK

THE STORY SO FAR

ABSTRACT

Having become involved with the Atari ST again, when I added a 1040 STF and built an STE from a mainboard and casework purchased from Ebay, I started to become involved in modding my STE, with a number of items, including a . 1.44Mb FDD adapter, 32Mhz booster and TOS switcher from Chris Swinson (AKA exxos) and an IDE interface from Alan Hourihane.

At some point along the way, there were also discussions centred around a realtime clock, which Thibaut Guillet had been talking to Chris about, based on schematic from Atari Bible, and I agreed, along with Thibaut, to trial a revised version of the ForgetMeClock II cartridge, but for fitting internally as a simple PCB to solder in.

Having received my circuit board from Chris, I assembled the parts and installed, but none of the software would see the clock, so we began to think about what was next.

The purpose of this document is to gather thoughts/knowledge about the experiences around the realtime clock are, so we have our knowledge in one place, and can be expanded on as we move forward.

When you're reading this, remember that I am not an electronic engineer, so a lot of what's in here was based in logic and deduction, and experience as an engineer in the IT world.

This is not intended to be a horribly technical document, but it will give you enough in the form of schematics and information to allow you to build a working RTC for an Atari ST or STE.

I have this working in a 4160 STE at both 8Mhz and 32Mhz with TOS 1.62 and 2.06, and a 520 STFM running TOS 1.04.

I may yet also attempt to add this to my oldest machine, a 1040STF, which I think also has TOS 1.04.

I also hope it may give some historical context for those that follow.

Hope this is of use to those of you who get to read this, and thanks to Chris Swinson, Thibaut Guillet and Troed Sangberg for all of their time, patience and input.

I'd also like to thank Techie Alison posthumously, she sadly died in a road accident in 2011, but left a lasting Atari legacy, as it is her version of the ForgetMeClock II software, corrected to allow time setting beyond 1999, which I have ultimately used for this project.

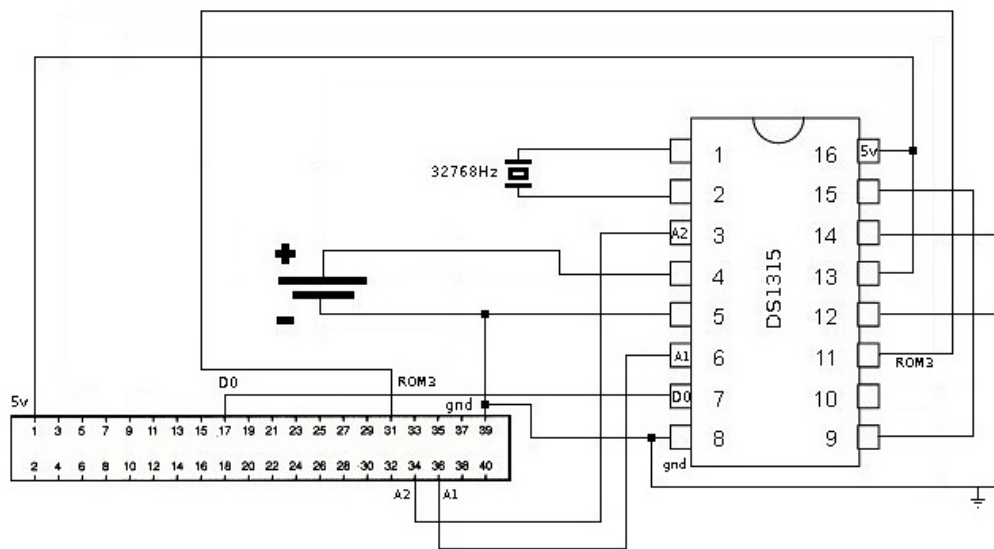
Jon Whiteside

October 2016

IN THE BEGINNING

I think the whole discussion was started by Chris Swinson, who was looking to put together an RTC for both the Falcon and the ST/STE, and he'd been discussing the whole thing with Thibaut Guillet.

The original design came from the Atari Bible, a French site (see the links section), and had the following design/schematic:



Chris had designed some PCB's around this, and asked if I would be interested in being a beta tester along with Thibaut. How could I resist!

The board duly arrived and was assembled, and I installed into my STE, which is the main object of my modding.

Though the original article also shows pick off points on the mainboard, I opted to go for soldering to the cartridge port internally to try and replicate the ForgetMeClock 2, around which this is based.



As I mentioned earlier, this didn't work, so I opted to try and connect via the cartridge port as per the original, and as below:

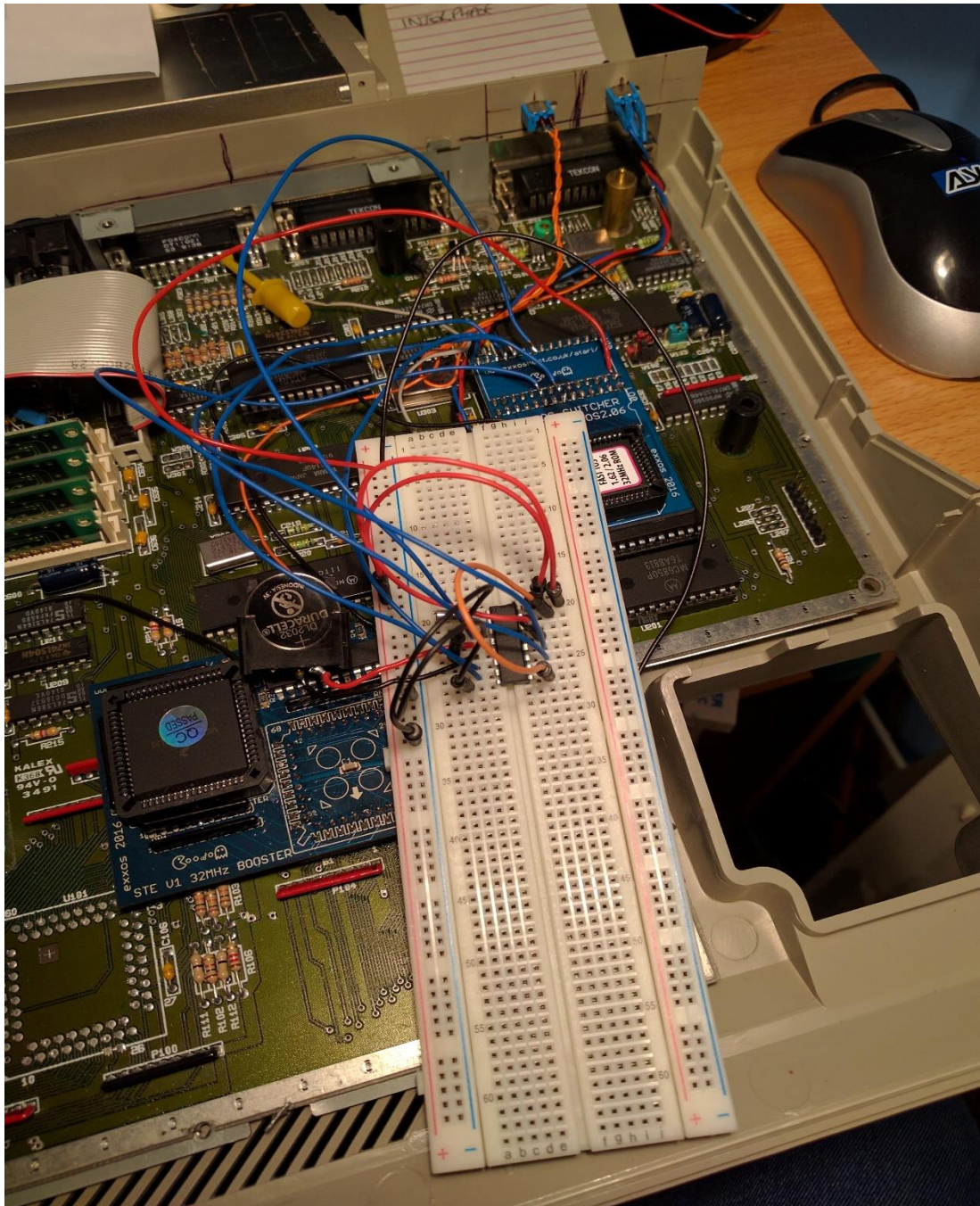


Again, this didn't work, but for reasons that would become clearer later on after some digging through datasheets and close examination of the images of the ForgetMeClock II PCB.

WHAT CAME NEXT: PART I

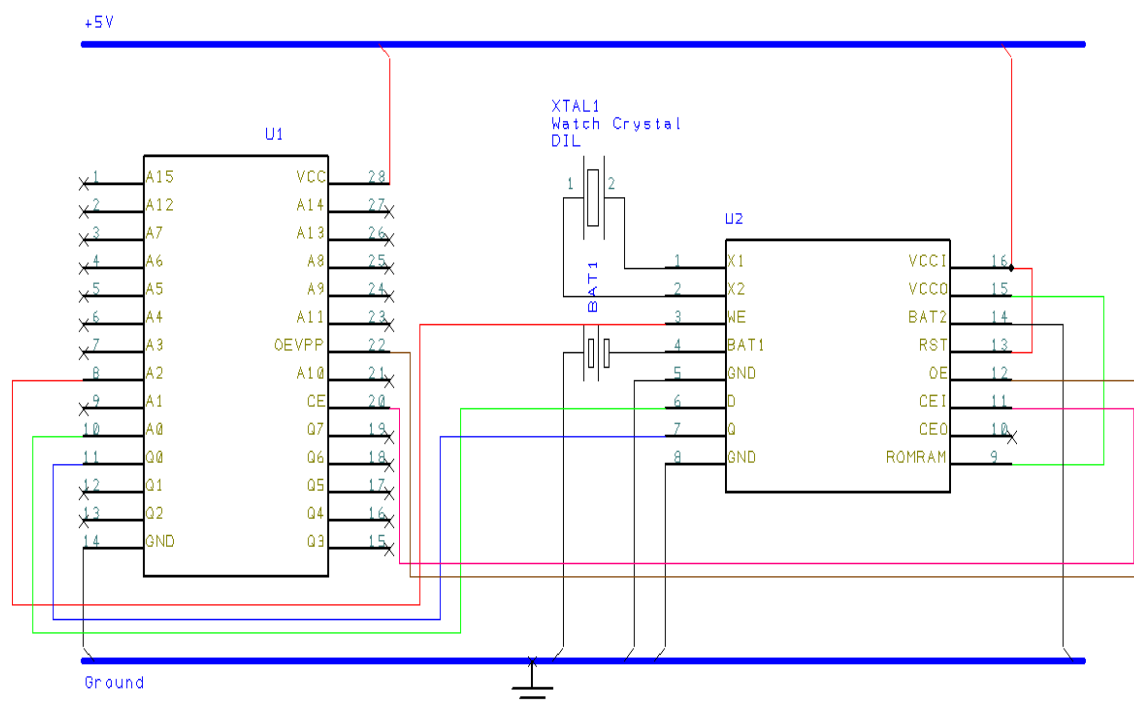
At this point, I was a little disappointed it wasn't working and having come across Troed Sangberg's site during my investigations, I saw that he was using a DS1216 clock chip, which is essentially the same clock, but in a 28 or 32 pin package that could be plugged into an **ATARI ROM** socket.

Having done a side by side comparison of the data sheets for both the DS1216 (No longer easily available) and the DS1315 (Easily available), it was apparent that the pinouts of both corresponded, so I breadboarded the components, and connected the DS1315 up to the top of my ROM switcher, matching the pins to those which would be present on the DS1216, as below:



Somewhat to my surprise and delight (I'm not an electronic engineer, remember), whilst the ForgetMeClock II software didn't work, the AReal software (See links below) that Troed had used, in combination with his Y2KFix for TOS did, and I was able to have a very stable clock with Y2K correction.

The schematic for this particular versions is shown below, **OPTION I**:



You all know that there is a but coming here, don't you.

Part of my reason for testing, was to see if it worked on my particular STE, which has Chris' 32Mhz booster installed, and whilst it was stable at 8Mhz, it was not picked up at all by the clock software at 32Mhz.

After discussing over Facebook with Chris, Thibaut, and Troed, it was thought that pin 10 of the DS1315, should probably be used as a pass through for CE.

So the next test involved be running a bypass from CE on the ROM socket, to pin 11 on the DS1315, and then having pin 10 connected to CE, or pin 22 on the ROM itself.

This would, in effect mimic the CE through on the DS1216.

Sadly, this didn't work either at 32Mhz, so I reported back my findings, and we all retired to think about what would come next.

I also tried this out later on my 520 STFM (Unboosted, so 8Mhz only), however during boot, when the clock software loaded, it caused the machine to reboot. If I did a clean boot with no floppy, it came up ok, and I could load the software and set the clock, however operation seemed unstable. As it happens, what came next involved some quality time with enlarged and printed out images of the ForgetMeClock II PCB, but we'll come onto that momentarily.

WHAT CAME NEXT: PART 2

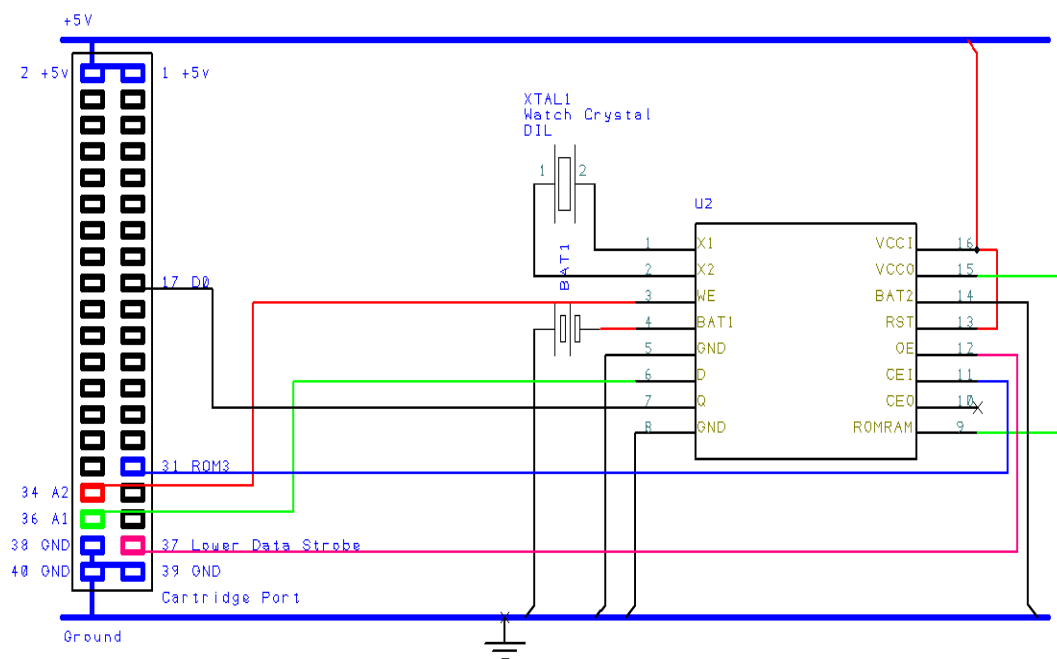
So the next stage was to take a deeper look at how the original ForgetMeClock II was wired, so I downloaded the pictures of both sides of the PCB from Atari Bible, printed them out large scale, and set aside some time to trace the tracks on the PCB.

Some time equated to about 3 hours of poring over the printed images, and cross referring to my laptop, and zooming in to the images on there to try and get more detail, and drawing the traces on my printouts.

What became clear from this review, was that all of the connections, bar pins 10 and 11 (CEO/CEI) simply picked off connections as they went by the DS1215 (DS1315 is Y2K corrected version of this).

Pins 10 & 11 were much different, in that pin 31 from the cartridge port went to pin 11 of the clock chip, and pin 10 then connected through to the through port socket on the outside of the cartridge.

This led initially to the following schematic, **OPTION 2**:

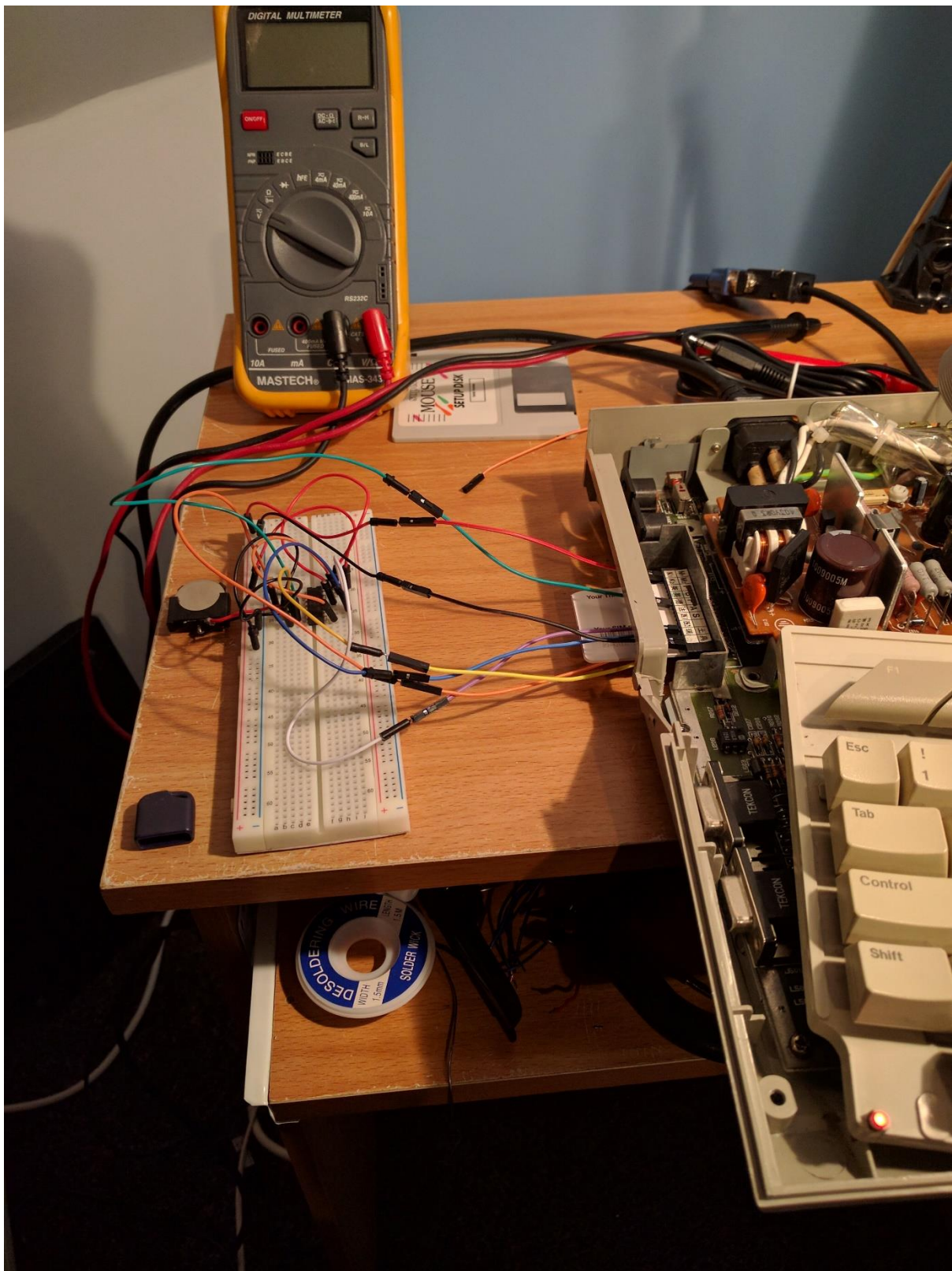


As can be seen, if comparing the above, to the original Atari Bible schematic, the only real main difference is that pin 12 on the DS is no longer connected to ground, but to pin 37, lower data strobe, on the cartridge port.

Chris had suggested this earlier in our Facebook discussions, so it was nice to confirm by tracing through on the PCB.

I put this together on a breadboard, and wired up to my STE cartridge port, where I was able to successfully use a combination of the Y2K software from Troed, and FGMC II software, where the clock ran fine for a number of days for testing.

The breadboarded solution attached to the STEs cartridge port:



From the breadboarded solution, I then built a version of the clock on protoboard, and connected internally to the cartridge port, and the clock has been running absolutely fine ever since, with just one minor modification, around the CEI/CEO pins, which I will come onto later.

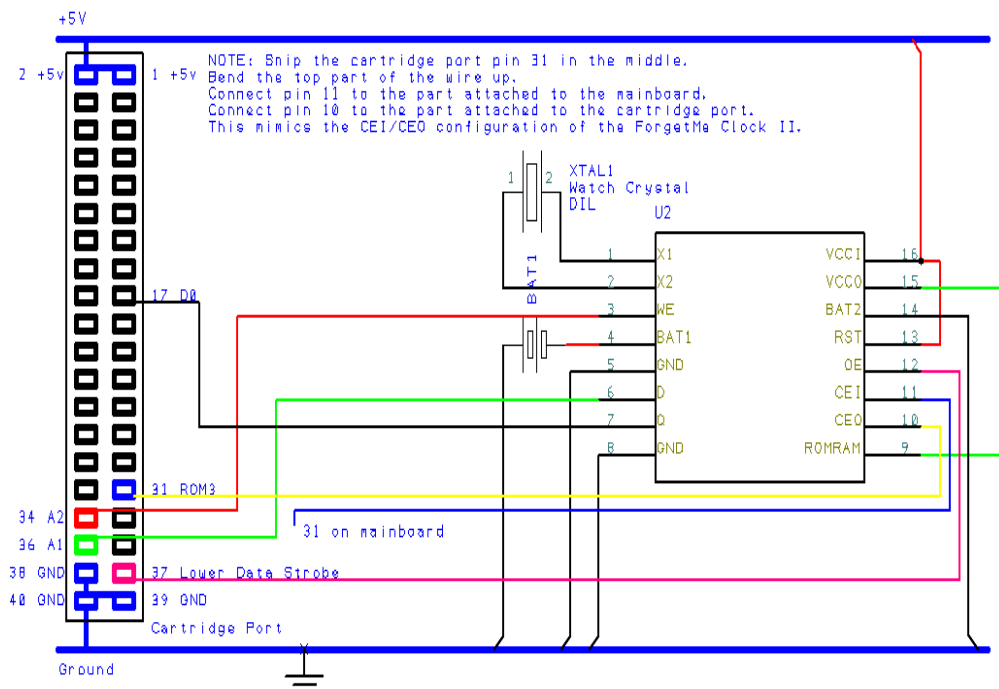
It is this version which I then came to install into my 520 STFM, which has been running successfully for some time now.

At this point it was not clear if we needed to simply pick off CEI from the cartridge port, or if that would not work if external cartridges were plugged in. Opinion seems to be that it should work, since the clock is only read during boot, and unless you have call to set it, that this configuration should work in all scenarios.

Not having any cartridges to test with has made this part difficult.

I suggested that perhaps the pin 31 connection on the cartridge port could be snipped, then we could perhaps mimic more closely the way the original FGMC II was set up.

I went ahead and tried on my STE, by snipping pin 31 on the cartridge port header in the middle. I bent up the top part so it was straight out, leaving the lower part sticking vertically out of the mainboard. I connected pin 11 of the DS1315 to the mainboard, and pin 10 of the DS1315 to the part of the wire left in the cartridge socket, mimicking the behaviour of the FGMC II as I described earlier. This is the schematic, **OPTION 3**:

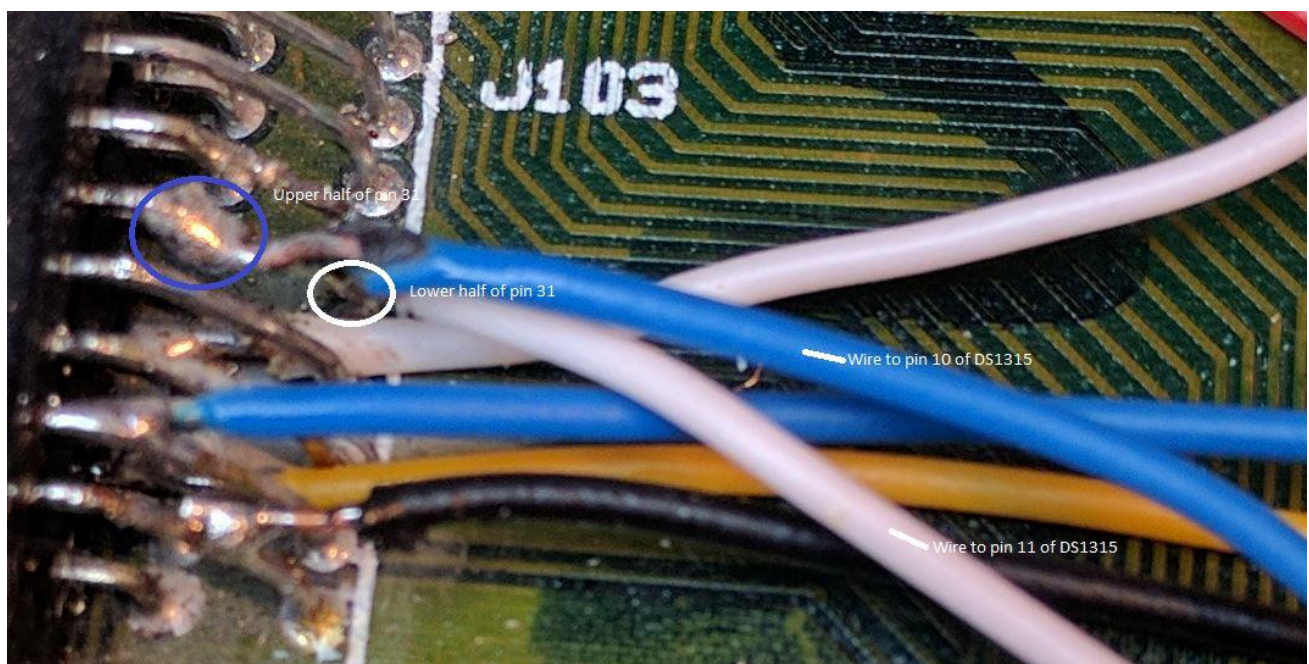
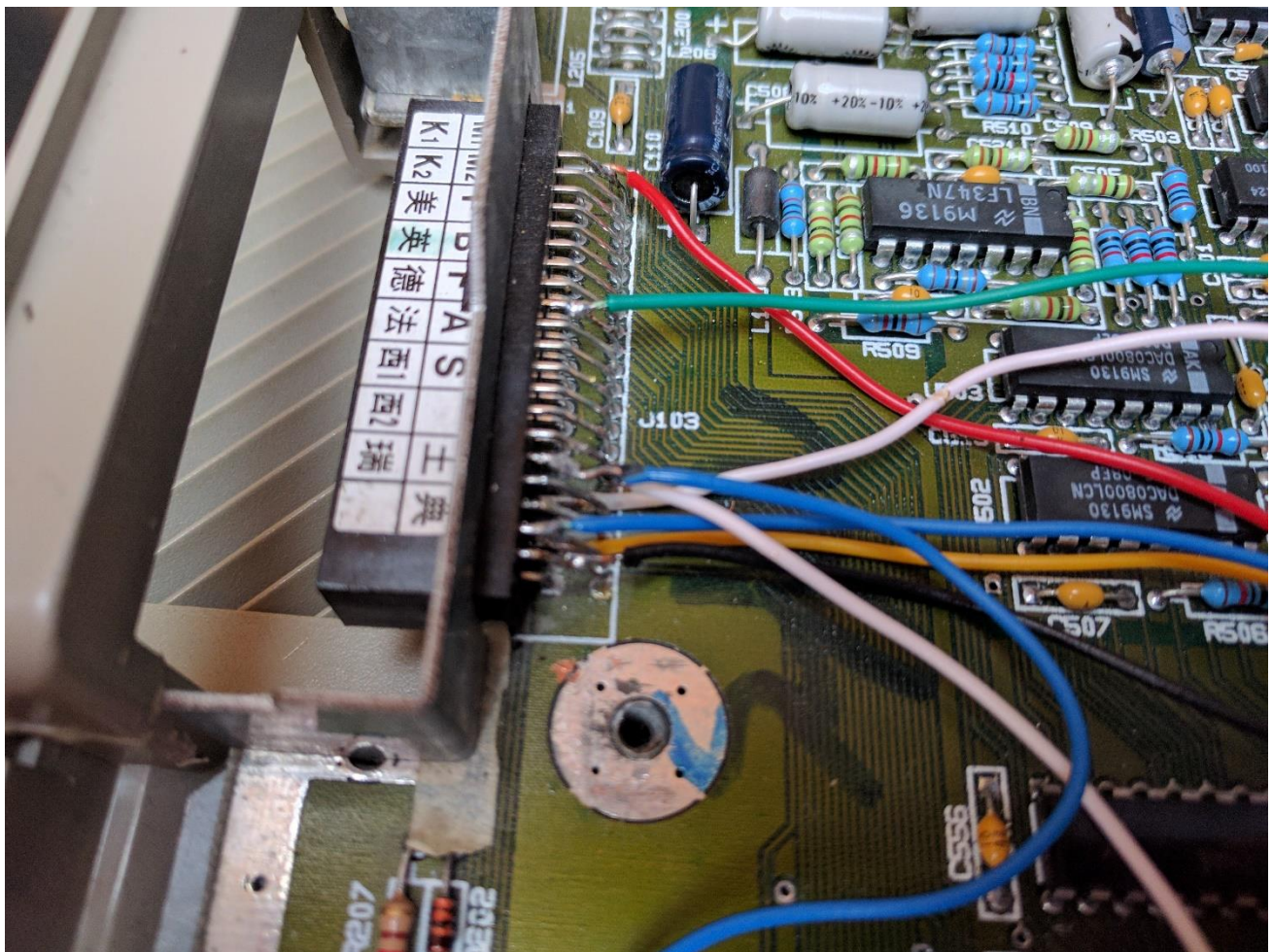


Thus far, the clock has been working perfectly, though again, I am currently unable to test due to having no cartridges.

Option 2 Installation in my 520STFM:



Option 3 on my STE, as close as I can get:



You can see the blue wire going to the top portion of the snapped pin 31 on the cartridge port, and a white wire going to the bottom portion that is left attached to the mainboard.

CLOCK SETTING AND CHECKING

I have included a link below which has a pre prepped boot disk image, which you can use with a Gotek or similar if you have one, or use something like FloImg to write to a physical floppy for use with a standard floppy drive.

Before attempting to set the clock, it is imperative that you have a fully functional battery. The clock won't respond if there is no battery, and you will go mad trying to set it!!

In order to work correctly, DALLRTC.PRG must load before AUTOFMC.PRG. If you are making your own disk with auto folder, be sure to put DALLRTC.PRG in the auto folder first.

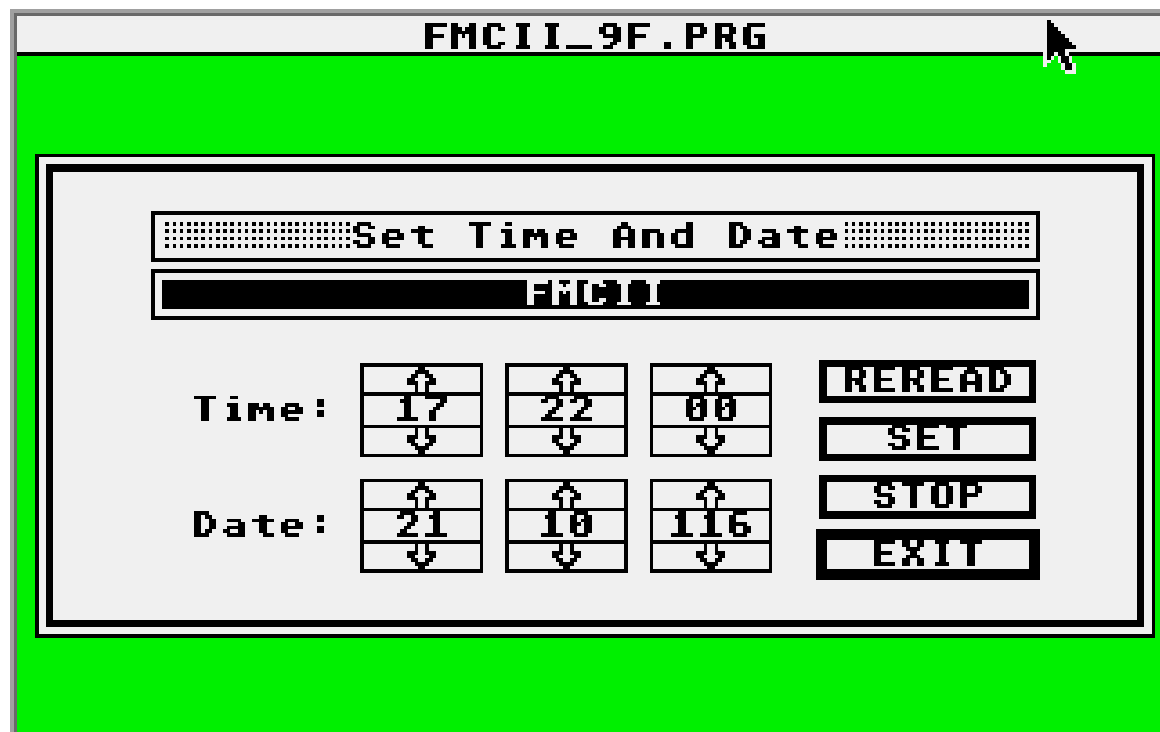
For initial setting, you DO NOT need to boot with anything in your auto folder.

To set the clock, after boot, run FMCII_9F.PRG. It may say there is no clock present, but if you have installed it correctly, it is just waiting to be set.

Set the time in 24 hour clock, HH:MM:SS, and the date, DD:MM:YYY.

The date is strange to account for Y2K, if you want to set for 2016, set YYY to be 116, 2017 would be 117 and so on.

It will look something like this:



Then hit the set button, and if you have connected everything correctly, it will blink and the clock is set.

If you click "REREAD", the software will go and retrieve the current time, and you will see the seconds increment.

This worked for TOS 1.04 and TOS 1.62.

When using TOS 2.06, I encountered some odd behaviour when setting the clock as above. I found it was first necessary to boot without loading DALLRTC.PRG, so you may as well boot with an empty AUTO folder.

It was then necessary to use FMCII.PRG to set to the correct time, but a date below Y2K, such as 99 (1999).

From there, it was possible to use FMCII_9F.PRG to set a date of 116 (2016) successfully.

After that you can happily boot with DALLRTC.PRG and AUTOFMC.PRG in your AUTO folder.

Don't forget, the DS1315 is Y2K compliant, so setting it isn't really the issue with the right combination of software, but reading the clock and passing the data through to TOS is where the Y2K fix is needed in order for everything to behave correctly.

TESTING/TROUBLESHOOTING

In order to verify clock operations, I carried out the following:

1. Created a folder on disk, and checked the date/time of the folder using the 'Show Information) option from the GEM menu.
2. Used SYSINFO.PRG to confirm the date/time in TOS.
3. Powered down the ST/STE for extended periods, with the power lead removed (About a week is the longest so far), then test clock operation by booting up with the disk provided below, and test using the methods in the previous 2 steps.

In reality, there is not a massive amount of troubleshooting you can do, but good areas to check are:

1. Is the battery good? It just won't behave well without.
2. Are your solder connections good? Soldering to A1 & A2 on the topside of the mainboard is extremely difficult, as the pins are obscured by the top pins of the cartridge port, so you may wish to pick these off on the underside of the cartridge port, or elsewhere on the mainboard to make life easier. Re-check all joints regardless, as a single dry joint will stop this working.
3. If FMCII.PRG or FMCII_9F.PRG say the clock is not running, or that you have the incorrect number of days for the month, it usually indicates the clock has never been set. This might be after a battery change. Set the date to something like 1/1/99, and the time to anything other than what is shown when the software boots. Hit the set button, and it should take this. Hit reread, and you should see the seconds increment. You can now set to the correct time.

SUMMARY

To summarise the results we've had here thus far, we have 3 options overall for an RTC.

All 3 options use the same common connections to a DS1315 clock chip. I have built 3 versions, one on a breadboard, and 2 on protoboard, and all used the same connections to the chip, just different connections to the computer, so if it doesn't work one way, just wire it up the other and hopefully you will see results.

OPTION 1

We can build a clock and attach directly to the ROM. In reality, whilst this worked for the STE, the results on a 520STFM were not good, so for me, this is the least likely option. It is also difficult to replicate the CEI/CEO passthrough if the ROMs are soldered to the mainboard. Requires the AReal software plus Troed Sangberg's Y2K fix for setting and reading in this configuration.

OPTION 2

We can build the same clock as option 1, and attach internally to the cartridge port, but ignore CEI/CEO passthrough.

This appears very reliable this far in my 520STFM, and uses the ForgetMeClock II software for setting and reading in this configuration.

As yet this is untested with cartridges.

OPTION 3

We can build the same clock as option 1, and attach internally to the cartridge port, but replicate CEI/CEO passthrough as in the ForgetMeClock II.

This is done by snipping the pin connecting the mainboard, to pin 31 of the cartridge port, and connecting pin 11 of the DS1315 to the end left in the mainboard and pin 10 of the DS1315 to the end left attached to the cartridge port.

This appears very reliable this far, and uses the ForgetMeClock II software for setting and reading in this configuration.

Again, this is, as yet, untested with cartridges.

I have both option 2 & 3 running happily in my 520 STFM and my 4160 STE.

If after reading this, you have questions, I can be contacted using: atarifanboi@deep-core.co.uk

WEB LINKS

Atari ST and Ste users on Facebook: <https://www.facebook.com/groups/133161394213/>

Chris' projects and web store: <http://www.exxoshost.co.uk/atari/last/index.htm>

Atari Bible:

<http://fr.meric.free.fr/Articles/articlesba/forgetmeclockinside/forgetmeclockinside.html>

Troed's blog: <https://blog.troed.se/projects/dallas-rtc-sync-tos-y2k-fix/>

DS1216 datasheet: <http://pdfserv.maximintegrated.com/en/ds/DS1216-DS1216H.pdf>

DS1315 datasheet: <http://pdfserv.maximintegrated.com/en/ds/DS1216-DS1216H.pdf>

ARes software: <http://storage.atari-source.org/atari/mirrors/kurobox.serveftp.net/utilities/clocks/>

Boot disk image ready to run, including FMCI, FMCI_9F, AUTOFMC and DALLRTC :
<https://drive.google.com/file/d/0B8c04HGf1tP3QklpdU1iSzR3ZFE/view?usp=sharing>