



Hacking Tektronix feature modules

Post Reply Search this topic...

6 posts • Page 1 of 1

Daniel Wee
Site Admin



Hacking Tektronix feature modules

Wed 01 Sep 01 2010 5:35 pm

- <http://herzogmuehlweg.de/TDS3UAM/TDS3UAM.html>
- [http://hackaday.com/2010/03/10/50mhz-to ... onversion/](http://hackaday.com/2010/03/10/50mhz-to...onversion/)
- [http://translate.google.com/translate?j ... S3UAM.html](http://translate.google.com/translate?j...S3UAM.html)
- [http://translate.google.com/translate?j ... F86697%2F2](http://translate.google.com/translate?j...F86697%2F2)

So we now have the re-cycled SIM card socket to be made into a universal feature module. This is mounted on a piece of breadboard which will also hold the 24C02 I2C EEPROM part. I'm a little concerned if the SOIC part profile may be too high.

Daniel



Daniel Wee
Site Admin



Re: Hacking Tektronix feature modules

Thu 02 Sep 02 2010 6:05 pm

Physical mounting issues aside - each of the slots (up to four possible) have the A0 and A1 addresses hard-coded. Now, the 24C02 has 256-bytes so the A0 and A1 bits are pure chip-select pins. With the 24C08 they actually form part of the addressing (A8 and A9). So when the master sends the device select code on the I2C bus, it will check against the settings. In the case of the 24C08, it becomes part of addressing the memory itself. This may be what we really want to do here. The single byte addressing rules out the possibility of using the 24FC512, for example.

If we understand this correctly, the scope will poll for the device on all combinations of the A0/A1 (E0/E1) bits, which represents the four slots. These would call up the corresponding 256-byte page. At the bottom of each page, the lowest 16-bytes identifies the module as follows:-

```
0xFF 0xFF 0xFF 0xFF 'D' 'P' 'O' '2' 'C' 'O' 'M' 'P' 0x00 0xFF 0xFF 0xFF
0xFF 0xFF 0xFF 0xFF 'D' 'P' 'O' '2' 'E' 'M' 'B' 'D' 0x00 0xFF 0xFF 0xFF
0xFF 0xFF 0xFF 0xFF 'D' 'P' 'O' '2' 'A' 'U' 'T' 'O' 0x00 0xFF 0xFF 0xFF
```

or (respectively):-

```
0xFF 0xFF 0xFF 0xFF 0x44 0x50 0x4F 0x32 0x43 0x4F 0x4D 0x50 0x00 0xFF 0xFF 0xFF
0xFF 0xFF 0xFF 0xFF 0x44 0x50 0x4F 0x32 0x45 0x4D 0x42 0x44 0x00 0xFF 0xFF 0xFF
0xFF 0xFF 0xFF 0xFF 0x44 0x50 0x4F 0x32 0x41 0x55 0x54 0x4F 0x00 0xFF 0xFF 0xFF
```

This would be placed in locations:-

```
0x00 0x00
```

0x01 0x00
0x02 0x00

Where the first byte represents what goes into A0/A1. The corresponding select codes are:-

0xA1 (read) - 0xA0 (write)
0xA3 (read) - 0xA2 (write)
0xA5 (read) - 0xA4 (write)

Keep in mind that the A2/E2 line needs to be grounded, whereas the A0/A1 line can be left floating in the 24C08 application since these will be ignored anyway. The write-control (_WC) line needs to be driven low as well, to enable writes to be made to the chip.

Daniel

Attachments

[ST-24C02-16 EEPROM.pdf](#)

(483.15 KiB) Downloaded 2320 times



Daniel Wee

Site Admin



Re: Hacking Tektronix feature modules

📅 Thu 02 Sep 02 2010 11:33 pm

Okay, we built the dongle and programmed the 2408 part but it seems that the scope voltage was too high for it. I guess I should order a proper 24C08 for the job.

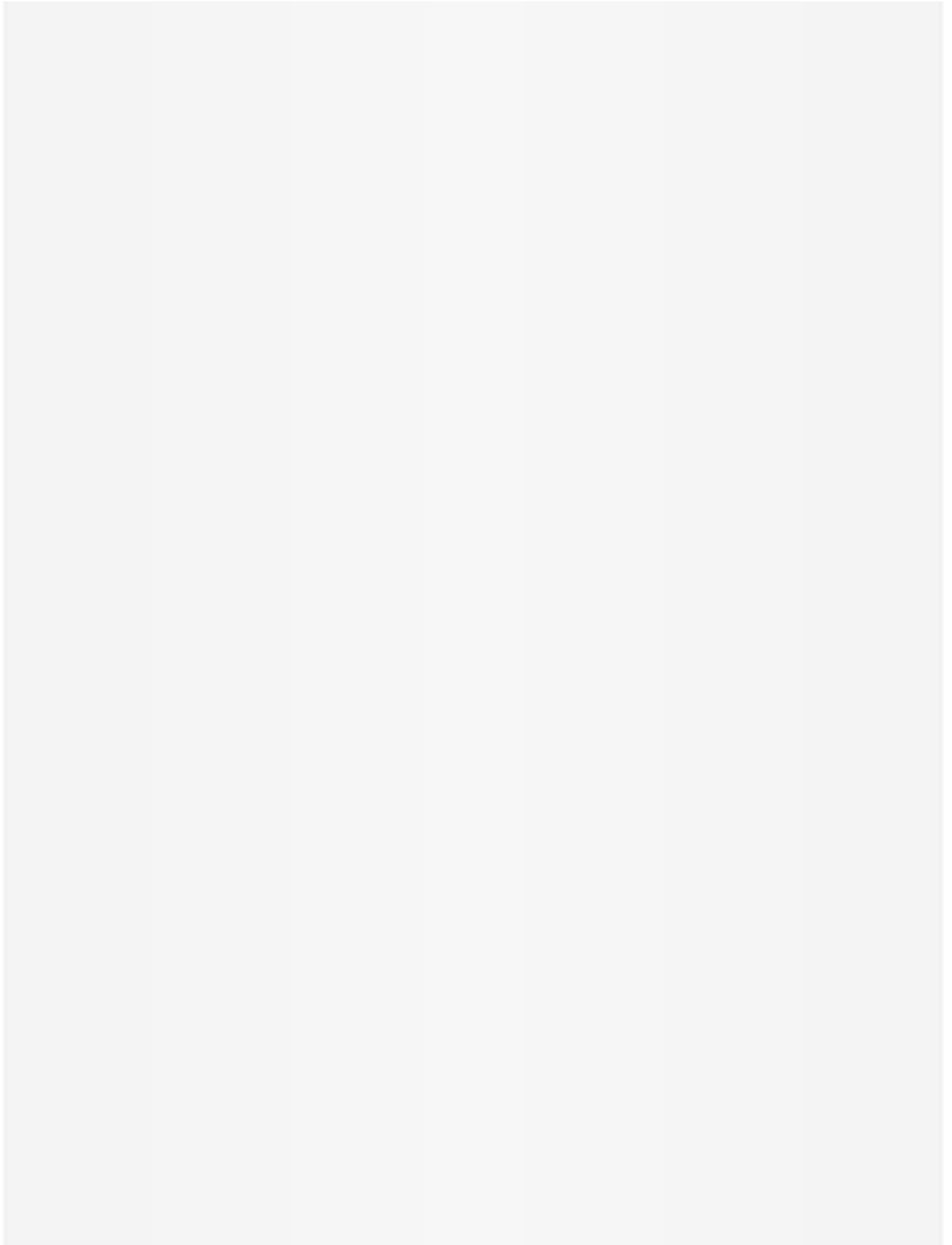
Daniel



Daniel Wee

Site Admin





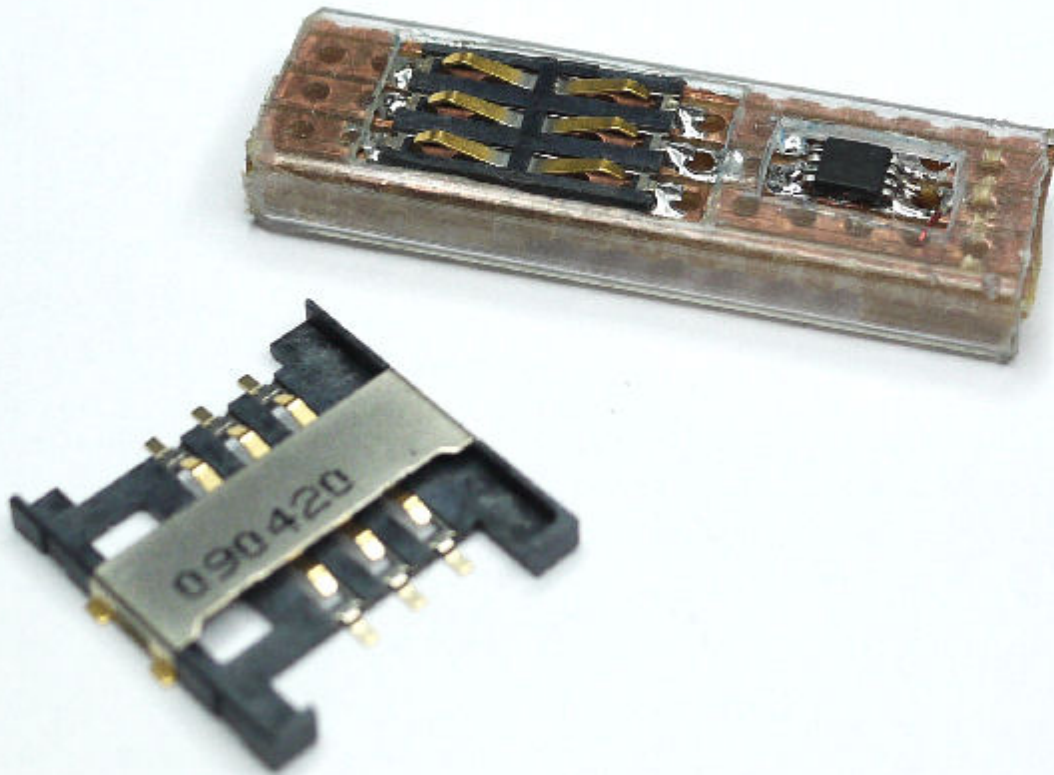
Re: Hacking Tektronix feature modules

■ Fri 03 Sep 03 2010 11:22 pm

Okay, I re-built using a 24C04 which was in a nicer MSOP8 package but turned out to have a quirk in that it ignores the A1/A2. So I wanted to try a 24C02 one, which is a real compromise but thought I'd give my original key a shot. As it turns out, it wasn't bad after all - it just needed a better connection to my programming interface (which is made of breadboard).

So I zero out the EEPROM, put in the four tags at the four 256-byte boundaries. The scope, however, only seems to recognize the first and the third blocks. Stuff in the second and fourth block are ignored. This is not surprising because the scope only has two slots but I was hoping that the code actually allowed for four. What is unusual is that it seemd to use the 00 and 10 combinations rather than the 00 and 01 combinations - meaning that it is the A1 bit that goes high while the A0 stays low. This would suggest that you could only pull this off with a 24C08 anyway, if you wanted more than one tag per chip - which, fortunately, is what we have done.

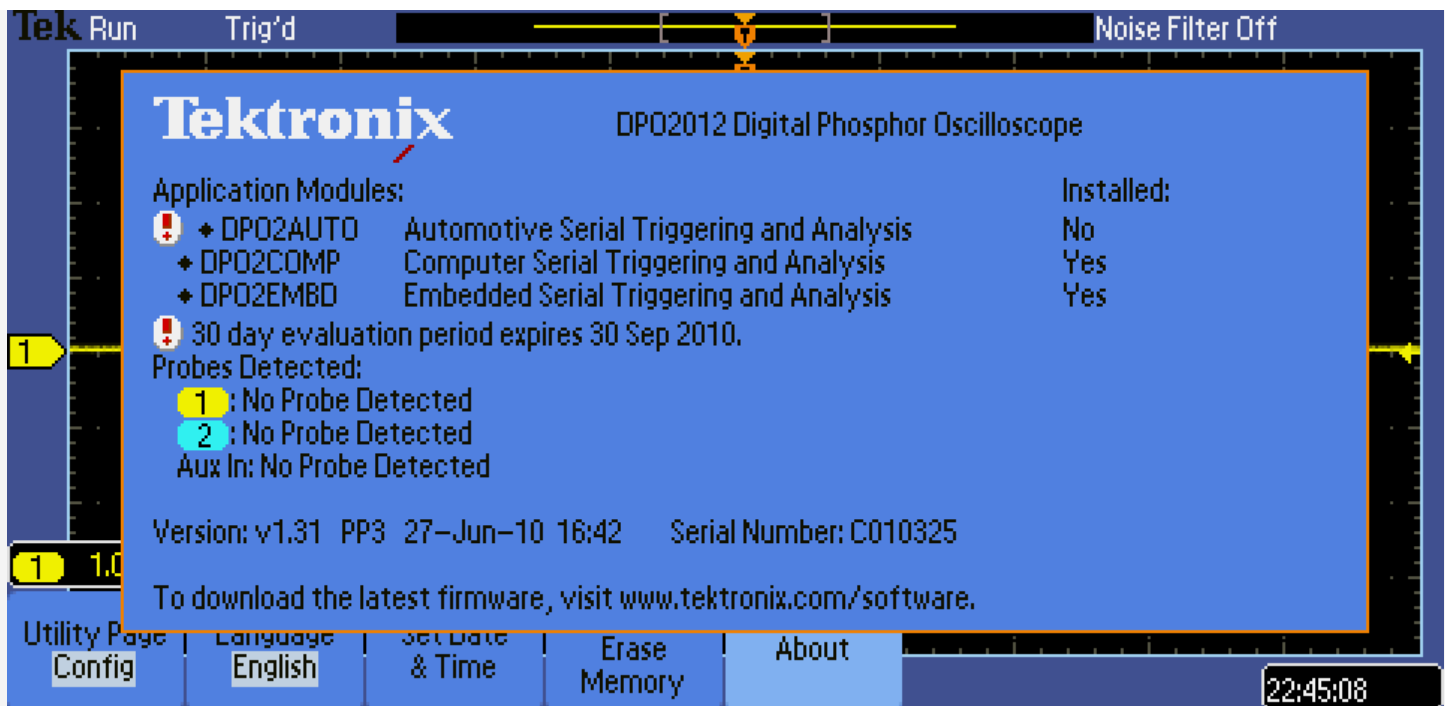
I took some measurements of the socket with my vernier calipers and found the dimensions to be approximately 36.5mm (length is not too critical) x 11.42mm (height) x 4.37mm (depth). There are two plastic catches that locks the module in place if you are using the original DPO2xxxx modules. The closest thing I could find was a plastic IC tube which just fit. So I cut that up and fashioned a board using SIM connectors and the TSSOP 24C08 that I pulled out from the handphone. This is what I made:-



TEKHACK 005a.JPG (54.46 KiB) Viewed 32634 times

I then hacked one of my control boards to program the I2C using the above mentioned clamp interface. Programming seemed to go well so I plugged the key into the scope slot and powered it up. For those of you who are familiar with the DPO/MSO2000 scopes, you will know that the start-up takes quite a long time (compared to my other OWON scope). When I originally had the module programmed wrongly, the scope would get stuck in startup and not come out for a very long time. This time round it did power up and still had the message that DPO2AUTO's free trial will expire soon. The first good sign is that there was no mention of the DPO2COMP and DPO2EMBD!

Hitting the Utility->About yielded this proof of success:-



TEKHACK 001a.PNG (10.67 KiB) Viewed 32634 times

So there you have it. Tek hacked! Before the trial ran out. So now I can go back to getting some real work done.

Daniel

Daniel Wee

Site Admin



Re: Hacking Tektronix feature modules

■ Sat 04 Sep 04 2010 2:30 am

Now, I was wondering why it only recognized two of the four possible modules. I found that I had not zeroed out the second and fourth block. It appears that it is actually possible for all four modules to be recognized. So I fixed the programming and made sure all the blocks were zeroed out except for where it mattered. This unfortunately didn't make the second and fourth blocks available even though I could read them using my own hardware. The EEPROM is working for sure.

I did find out that zeroing is not necessary if you write 0xFF's to all the locations. What this means is that the information on the net on the length of the string is somewhat inaccurate. The 0xFF's aren't so much padding as they are simply unwritten locations. The protocol, therefore, simply calls for a null terminated string that starts at address 0x04 of each 256-byte block.

update: Upon reviewing the information, it looks like no one has actually gotten all four modules to be recognized simultaneously by the DPO2000 series so I think it is quite safe to say that only two modules will be read at any given point in time. I'll just have to content myself with the DPO2COMP and the DPO2EMBD module functions.

Daniel




Daniel Wee

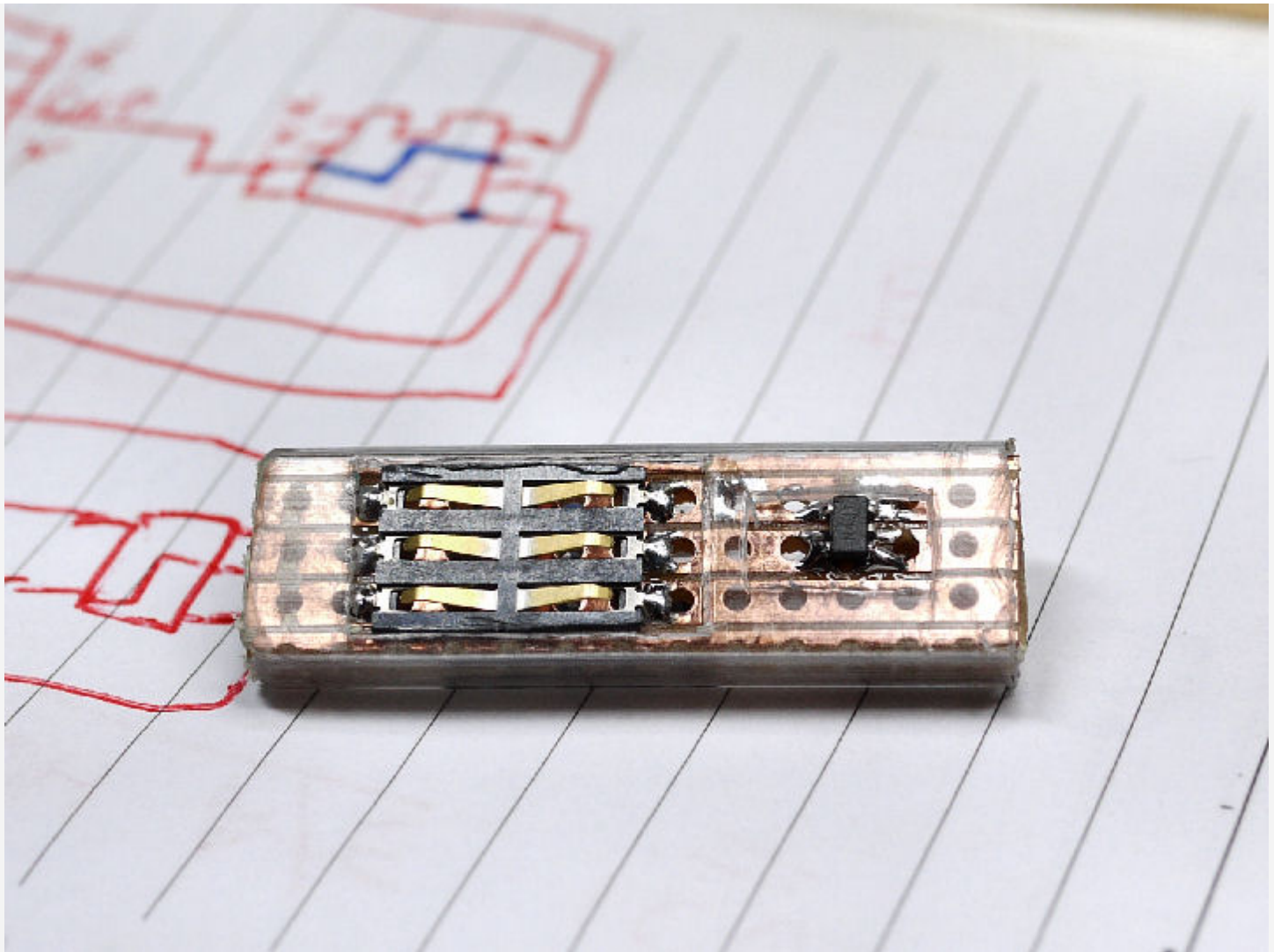
Site Admin



Re: Hacking Tektronix feature modules

 Mon 06 Sep 06 2010 4:09 pm

I decided to re-build the dongle using the 24LC08 (SOT23-5 packaging) instead for a cleaner assembly and better fit. It worked perfectly:-



TEKHACK 002a.JPG (76.8 KiB) Viewed 32619 times

Some super-glue at both ends keep the board in place. I think this is it, I'm done with this hack.

Daniel



[Post Reply](#) ↩



6 posts • Page 1 of 1

[< Return to "Projects"](#)

[Jump to](#) ▾

[🏠 Board index](#)

[✉ Contact us](#)

[🛡 The team](#)

[🗑 Delete all board cookies](#)

All times are UTC+08:00

Powered by [phpBB®](#) Forum Software © phpBB Limited
Style by [Arty](#) - Update phpBB 3.2 by MrGaby