UNI-T 210E Hacking Info : Part 1

Found online at EEVBlog, Kerry Wong's Website etc.

In regards to programming the EEPROM in this device. Everything is provided, there is a place for the installation of the connector, pitch is 2mm. The pinouts: GND, VCC, TEST, SCL, SDA. You can directly connect TEST to GND. VCC taken directly from the programmer . The only thing you need to remember to switch UT210E selector to any position except OFF.

Following information is about modifying features of you meter. Take backup of eeprom's content if you are going to modify it's content. You are doing possible modifications at your own risk!

There has been some talk about reordering the modes, for example to make DCA come before ACA. Unless I've missed a post or few, there hasn't been English information about what those bytes actually mean and how to possibly make other similar changes.
Knowing what bytes need to be swapped to swap places of DC and AC modes is nice, but wouldn't it be great to have **information how to reorder the modes the way YOU want (within some limitations)**?

For quick reference, check out first attachment image to see where modes of each selector switch are being set and second attachment to see what the bytes on those locations mean. Second attachment also mentions some jumpers... You can switch places of modes only if they have same set of jumpers mentioned.

The longer explanation:
Chinese datasheet of DTM0660L (Here's a link to that datasheet in PDF format. Link was shared by user Puomjw on the thread of Russian forum that has been discussed earlier.) explains the EEPROM content at 0x80 - 0xBF, which is where order of the modes is set. **Least significant nibble of address on that area represents the selector switch position and most significant nibble represents the mode selectable by Select-button**.
For example bytes in locations 0x8F, 0x9F, 0xAF and 0xBF set all four modes of one specific selector switch position. 0x8F contains default mode, pressing the Select-button cycles between that and the modes set in 0x9F, 0xAF and 0xBF. See the first attached image for clarification.

UT210E's eeprom has 0x07, 0x09, 0x0A and 0x0B in those four locations that were mentioned as an example. The datasheet has a chart (Section 11.3 in the datasheet I linked in this post, also in second attachment of this post) which tells what modes each of the bytes represents.
Comparing those bytes from eeprom to the chart on datasheet: 0x07 = Resistance, 0x09 = Continuity, 0x0A = Diode and 0x0B = Capacitance. Looking at the UT210E, there surely is a selector switch position which has resistance, continuity, diode and capacitance modes. So, now we know which four byte section sets modes of that selector switch position. Doing similar comparisons to rest of the four byte sections on that 0x80-0xBF area of UT210E's eeprom results this:

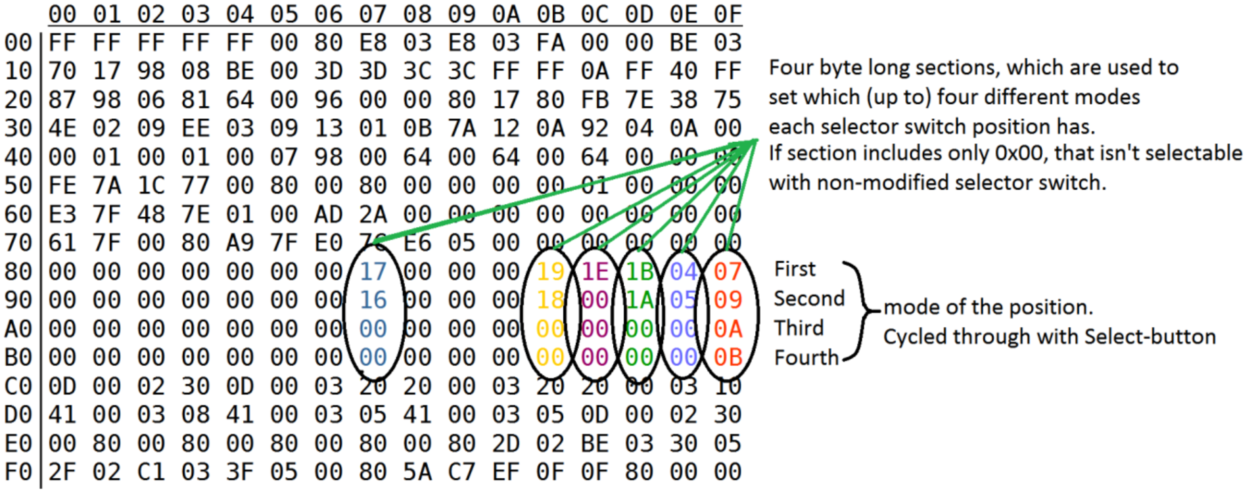
Locations	**Selector switch position**
0x87, 0x97, 0xA7 and 0xB7	2A
0x8B, 0x9B, 0xAB and 0xBB	20A
0x8C, 0x9C, 0xAC and 0xBC	NCV
0x8D, 0x9D, 0xAD and 0xBD	100A
0x8E, 0x9E, 0xAE and 0xBE	Volts
0x8F, 0x9F, 0xAF and 0xBF	Resistance/Continuity/Diode/Capacitance

Rest of the four byte sections include only 0x00 and aren't used (not selectable with selector switch).

Limitations

There are some things that limit the possible modifications you can do to order of the modes: In second attachment there is column with jumper names: J1A, J5 etc. Selector switch also shorts or opens these jumpers, modifying the way input signal travels inside the meter.
You can only replace a mode with another one, if that new mode has exact same set of jumpers mentioned, as the old one. For an example: Looking at second attachment, you can move continuity before resistance, but you cannot replace NCV with continuity.

There aren't too many things one can do, but I hope this helps!



```
   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00|FF FF FF FF FF 00 80 E8 03 E8 03 FA 00 00 BE 03
10|70 17 98 08 BE 00 3D 3D 3C 3C FF FF 0A FF 40 FF
20|87 98 06 81 64 00 96 00 00 80 17 80 FB 7E 38 75
30|4E 02 09 EE 03 09 13 01 0B 7A 12 0A 92 04 0A 00
40|00 01 00 01 00 07 98 00 64 00 64 00 64 00 00 00
50|FE 7A 1C 77 00 80 00 80 00 00 00 00 01 00 00 00
60|E3 7F 48 7E 01 00 AD 2A 00 00 00 00 00 00 00 00
70|61 7F 00 80 A9 7F E0 7C E6 05 00 00 00 00 00 00
80|00 00 00 00 00 00 00 17 00 00 00 19 1E 1B 04 07
90|00 00 00 00 00 00 00 16 00 00 00 18 00 1A 05 09
A0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0A
B0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B
C0|0D 00 02 30 0D 00 03 20 20 00 03 20 20 00 03 10
D0|41 00 03 08 41 00 03 05 41 00 03 05 0D 00 02 30
E0|00 80 00 80 00 80 00 80 00 80 2D 02 BE 03 30 05
F0|2F 02 C1 03 3F 05 00 80 5A C7 EF 0F 0F 80 00 00
```

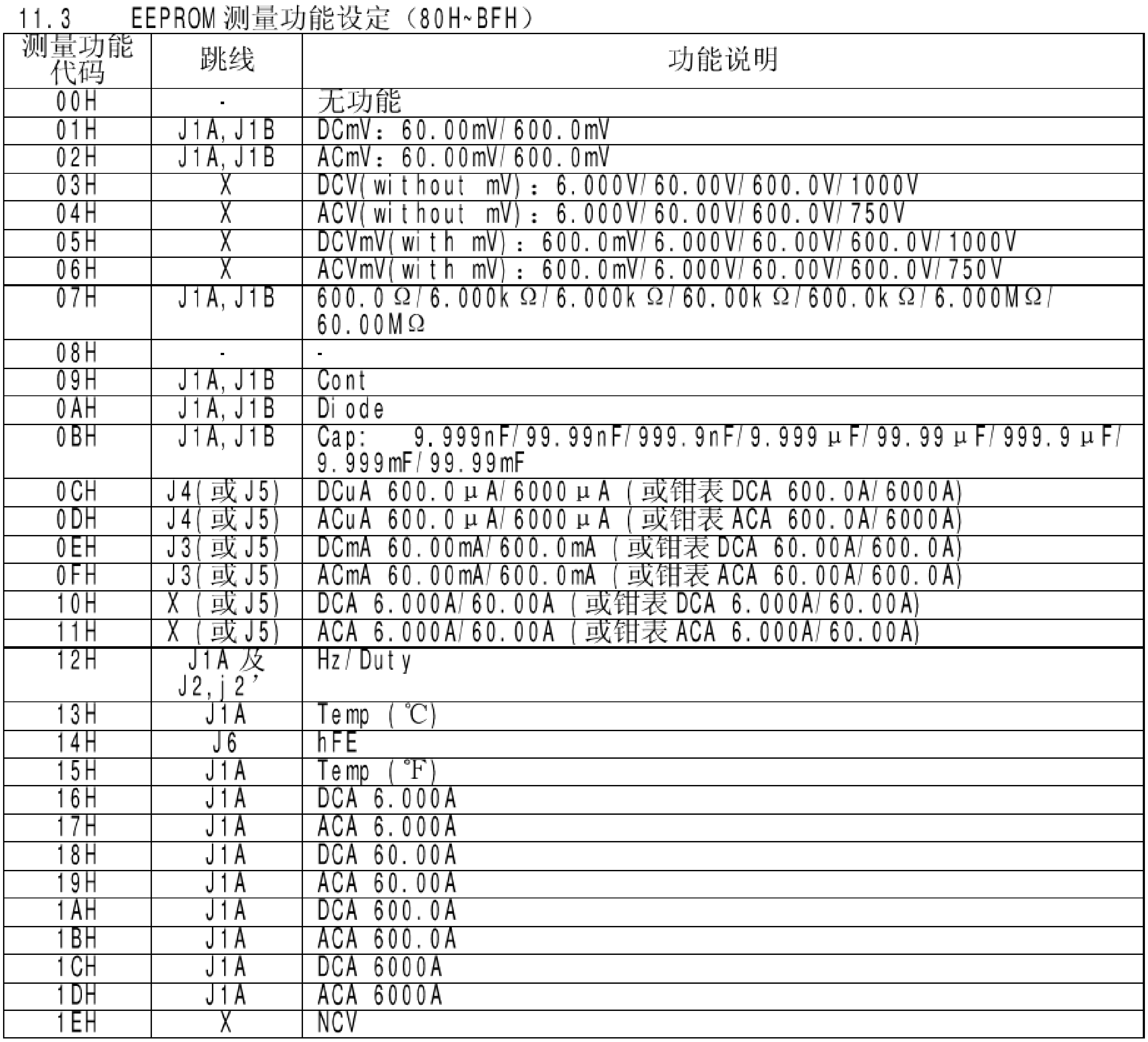
Four byte long sections, which are used to set which (up to) four different modes each selector switch position has.
If section includes only 0x00, that isn't selectable with non-modified selector switch.

First
Second
Third
Fourth

mode of the position.
Cycled through with Select-button

Example EEPROM content. Joeqsmith's Uni-T UT210E.

This dump is used for reference, DO NOT use this as "backup". Take backup of your meter's eeprom before modifying it's content!



11.3 EEPROM 测量功能设定（80H~BFH）

测量功能代码	跳线	功能说明
00H	-	无功能
01H	J1A, J1B	DCmV：60.00mV/600.0mV
02H	J1A, J1B	ACmV：60.00mV/600.0mV
03H	X	DCV(without mV)：6.000V/60.00V/600.0V/1000V
04H	X	ACV(without mV)：6.000V/60.00V/600.0V/750V
05H	X	DCVmV(with mV)：600.0mV/6.000V/60.00V/600.0V/1000V
06H	X	ACVmV(with mV)：600.0mV/6.000V/60.00V/600.0V/750V
07H	J1A, J1B	600.0Ω/6.000kΩ/6.000kΩ/60.00kΩ/600.0kΩ/6.000MΩ/60.00MΩ
08H	-	-
09H	J1A, J1B	Cont
0AH	J1A, J1B	Diode
0BH	J1A, J1B	Cap: 9.999nF/99.99nF/999.9nF/9.999μF/99.99μF/999.9μF/9.999mF/99.99mF
0CH	J4(或J5)	DCuA 600.0μA/6000μA (或钳表DCA 600.0A/6000A)
0DH	J4(或J5)	ACuA 600.0μA/6000μA (或钳表ACA 600.0A/6000A)
0EH	J3(或J5)	DCmA 60.00mA/600.0mA (或钳表DCA 60.00A/600.0A)
0FH	J3(或J5)	ACmA 60.00mA/600.0mA (或钳表ACA 60.00A/600.0A)
10H	X (或J5)	DCA 6.000A/60.00A (或钳表DCA 6.000A/60.00A)
11H	X (或J5)	ACA 6.000A/60.00A (或钳表ACA 6.000A/60.00A)
12H	J1A 及 J2,j2’	Hz/Duty
13H	J1A	Temp (℃)
14H	J6	hFE
15H	J1A	Temp (℉)
16H	J1A	DCA 6.000A
17H	J1A	ACA 6.000A
18H	J1A	DCA 60.00A
19H	J1A	ACA 60.00A
1AH	J1A	DCA 600.0A
1BH	J1A	ACA 600.0A
1CH	J1A	DCA 6000A
1DH	J1A	ACA 6000A
1EH	X	NCV

Kerry's Arduino sketch can be used to read original values from the EEPROM and also to write the new ones to the EEPROM. Without modifications it reads the content. Do that and save the output from serial monitor to a text file. Using the data from previous step, see what is the original value on the location you want to modify. Convert that HEX value to binary, change the bits based on the table found in the datasheet (http://www.kerrywong.com/2016/04/03/dtm0660-datasheet-translated/) and then convert that binary number back to a HEX value

Uncomment "writeByte(I2C_ADDR, 0xFA, (byte) 0xCE);" in the setup - function of Kerry's Arduino sketch and change 0xFA to be the location you want to modify and change 0xCE to be the value you want to store in that location.

That makes the Arduino modify the data inside the multimeter's EEPROM with the new value.

Offset Issues

I just received my UT139C so I'll try this thing out... anyway, I have a question for you - my UT139C shows some value on the lower uA~, mA~, A~ and mV~ ranges, even with nothing connected & leads shorted. Of course i can subtract that offset or use the REL function to offset this, but it's still annoying... any idea how to calibrate the zero offsets without modifying the EEPROM? If I decide to modify the EEPROM content, the "range noise to be subtracted" values should be the ones that I should modify, right? Thanks.

- Having a non-zero reading in mV mode with nothing connected is normal, this is because of very high input impedance. When the leads are shorted, they might become an antenna and pick up noise from environment, this could cause very small reading to appear... Again because of the very large input impedance. I tested with my UT139C and when I short the leads, I get 0.00mV DC or 0.03mV AC. By really trying to pick up noise from surrounding devices, I managed to get maximum reading of 0.1mV AC... No more than that. On uA and mA modes, on both AC and DC mode, I get 0 when leads are shorted and 0 when the leads aren't connected. Depending on how large values you are getting on those lower uA~ and mA~ ranges, the calibration data could be wrong, or maybe it is so small that it's caused by interference from the environment. Depending on where you bought the meter, I would suggest trying to get it replaced instead of fixing it. But I do get it: Why wait for replacement for weeks when the problem can be fixed in minutes, right? Unfortunately I don't have any more information about the calibration values (or calibration process) and you guess is as good as mine. If you proceed and try those "range noise to be subtracted" registers, be sure to take backup of the eeprom first!

Hi, so I successfully applied this hack (but I did a few things a tiny bit differently) and it works perfectly, thanks. Also managed to solve my offset problem (multimeter reading non-zero values on lower AC ranges with leads shorted). The "AC60mV range noise to be subtracted" register takes care of that. This may have resulted in some accuracy loss when measuring higher values though.

DON'T DO THIS.....
Press the blue and yellow button, and turned on to Reset calibration

I am hoping I don't have to pull the eeprom 😮 Confirming, you tried attaching external 5v to VCC and through 10k resistors to SCL/SDA and it didn't work?

Yes, I tried, using a bus pirate connected to the 5 pin header on the PCB (see photos earlier in the thread). The BP could snoop the bus and see valid transactions whenever the unit was turned on or function was changed. When trying to read or write, the BP always complained about the lines not going high. Unfortunately the microprocessor seems to drive those lines low when not actively talking to the EEPROM... It does not float them like it should. So I pulled and reinstalled the EEPROM (a few times). I used a TL866 and a SOIC test socket to read and write it.

I'm thinking of mounting my EEPROM at a slight angle so the pins 4 to 7 (Vcc, WP, SDA, SCK) are lifted about 1 mm off the board, and pins 1-4 remain securely on the board. Then I can bridge the gaps with solder for normal operation or wick the solder out to attach leads for programming.

As above, I lifted the front (pins 4-7) of the EEPROM a little, just enough to remove the solder bridging the pins to the board. Then I soldered 5 cm long wire-wrap wires (30 ga Kynar) to the leads, soldered those to a 5-pin socket. Then I use breadboard jumper wires from that socket to a 8-pin DIP machine pin socket placed in the TL866 ZIF. It's less complicated than it sounds. I reprogrammed the EEPROM with the Russian hacker's data deltas (as below) and ... boom! Well, no, no boom, just a 9999 count meter. The issue of 2000 counts on the 2 A range (only that one) when using ZERO remains unfortunately. Besides the 9999 counts, the other change is defaulting to DC for all volt and amp measurements.

Addr	OLD	NEW
10	70	10
11	17	27
12	98	0F
13	08	27
87	17	16
8B	19	18
8D	1B	1A
8E	04	03
97	16	17
9B	18	19
9D	1A	1B
9E	05	04
FC	0F	00
AE	00	05
BE	00	06

Note address FC is the backlight timeout, and 00 means do not time out (never turn off automatically). Otherwise, the value is in seconds. I set mine to 1E (30 seconds). Make sure to archive a copy of your unmodified EEPROM data.

> ~~I don't remember if there was a millivolt range in either DC or AC previously, but there is not now (0.000 V to 9.999 V is the lowest range). Can someone check?~~

edit: Added mods at offsets AE and BE to enable DC and AC mV ranges.

With selector in V range order is: V(DC) - V(AC) - V(DC mV) - V(AC mV)

AE: 00 05
BE: 00 06

I tried to enable the frequency counter and temperature measurement - but they do not work.

In conclusion, - these are the most mods that you can safely do to UT210E:

1) Selector on V, ranges: V (DC) - V (AC) - V (DC mV) - V (AC mV)
2) Selector on R, functions: resistance - connectivity- diode - capacity
3) Selector on 2A: A (DC) - A (AC)
4) Selector on 20A : A (DC) - A (AC)
5) Selector on 100A Switch: A (DC) - A (AC)
6) Change/Remove the Automatic backlight off
7) Extended range up to 10,000 counts. This means that up to 9,999 counts in any mode(including 2A) comes with three characters after the decimal point, then it switches range. The lower limit left as is, down to 1.90 will be two decimal, then switch range. Most of the time this allows more convenient to work with similar values, without changing the range.

JoeQSmith Post....

0x10 - 0x70
0x11 - 0x17

So 0x1770 or 6000 decimal

Below is the entire EPROM contents for mine in raw format:

```
FFFFFFFFFF0080E803E803FA0000BE03
70179808BE003D3D3C3CFFFF0AFF40FF
87980681640096000080 1780FB7E3875
4E0209EE030913010B7A120A92040A00
00010001000798006400640064000000
FE7A1C77008000800000000001000000
E37F487E0100AD2A0000000000000000
617F0080A97FE07CE605000000000000
0000000000000017000000191E1B0407
000000000000001600000018001A0509
0000000000000000000000000000000A
0000000000000000000000000000000B
0D0002300D0003202000032020000310
4100030841000305410003050D000230
008000800080008000802D02BE033005
2F02C1033F0500805AC7EF0F0F800000
```

MacBoy’s Comment

I spent a little time playing with modifying the EEPROM.

First, a comment about setting up 9999 counts. This absolutely works fine for DCV and resistance measurements. For ACV (or ACA), there is an issue. The problem is caused by the fact that this meter uses fast sampling and DSP to calculate the RMS value of an AC input, rather than using the traditional approach of an RMS-to-DC converter (e.g. AD637). The peak value of a sine wave is 1.414 the RMS value. So to get an RMS reading on the meter of say 9.999 VAC, the meter needs to sample instantaneous voltages up to 14.141 V. *It can't*.

The ADC tops out at roughly 12000 counts. This means that, for AC, the maximum displayed counts should be limited to around 8000 count. This allows enough headroom in the ADC to sample the peaks of sine waves (and clipped sine waves which you will likely see on AC mains). Still, any waveform with a higher crest factor may show a RMS reading below actual, when the reading is reaching the top of a range. A workaround might have been to manually range higher, but this meter has no range button, so you can't do that. **All this was to say that I recommend leaving the maximum counts to 8000 or less.**

I have also used what I learned here (big thanks to **hugatry**) to modify another DTM0660-based meter that I own. On that meter I added a mV range with high impedance input.

Flywheelz’s Answer

What values I should change/use to get the 8000 count setting? I'm a "little" confused with those 10-13 location values...

Btw. I used a cheap CH341A programmer and test clips from eBay, programming worked just fine. Only needed to turn meter's mode switch to voltage position, didn't work in the off position.

Hi, I will try to add some info.

From datasheet:
Code: [Select]

```
10H, 11H 1770H Default full range 6000d (10H stores low byte 70H, 11H stores high byte 17H)
same low byte/high byte arrangement below
12H, 13H 1838H Range switch upper limit 6200d (for full range of 4000d the upper limit is 4200d)
```

First you need to convert 8000 decimal to hex. Using Windows Calculator in programmer mode, Make sure radio button is on Dec. Type 8000 then flip to Hex and you get 1F40.

In the Eeprom the data is reversed (Little Endian) so you want to write it as 401F.

```
Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
```

```
00000000   FF FF FF FF FF 00 80 E8  03 E8 03 FA 00 00 BE 03
00000010  40 1F 40 1F BE 00 3D 3D  3C 3C FF FF 0A FF 40 FF
```

Finally I cracked open my ut210e to mod the eeprom. I am using Arduino wired directly to the header next to the chip. At first the chip would not respond. I then disconnected the WP pin 7 from GND and now I can dump the eeprom. But as soon as WP is grounded the chip will not respond. I noticed when WP is grounded the SCL and SDA drop to ~2.5volts.

Any ideas?

here is data from my eeprom
Code: [Select]

```
FF FF FF FF FF 00 80 E8 03 E8 03 FA 00 00 BE 03
70 17 98 08 BE 00 3D 3D 3C 3C FF FF 0A FF 40 FF
7A 98 06 81 64 00 96 00 00 80 17 80 FB 7E 38 75
4E 02 09 5A 02 09 A0 FF 0A 3D 13 0A 5D 03 0A 00
00 01 00 01 00 07 98 00 64 00 64 00 64 00 00 00
83 75 EC 70 00 80 00 80 00 00 00 00 01 00 00 00
E3 7F 48 7E 01 00 BA 2A 00 00 00 00 00 00 00 00
C2 7F 00 80 0A 80 E0 7C 07 06 00 00 00 00 00 00
00 00 00 00 00 00 00 17 00 00 00 19 1E 1B 04 07
00 00 00 00 00 00 00 16 00 00 00 18 00 1A 05 09
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0A
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B
0D 00 02 30 0D 00 03 20 20 00 03 20 20 00 03 10
41 00 03 08 41 00 03 05 41 00 03 05 0D 00 02 30
```

```
00 80 00 80 00 80 00 80 00 80 49 02 E5 03 59 05
4A 02 EB 03 6E 05 00 80 5A C7 EF 0F 0F 80 00 00
```

I can read the eeprom just fine as long as WP is not grounded. The ut210e circuit design pulls WP high through 10k.

Trouble starts when I ground the WP, then I can't read or write. Maybe I need to remove the 10k on the pcb? But others did not need to do that.

I am reading Kerry's blog, http://www.kerrywong.com/2016/03/19/hacking-dtm0660l-based-multimeters/

With WP not grounded I get:

Code: [Select]

```
VCC=4.90V
WP=4.90V
SCL=4.70V
SDA=4.85V
```

With WP grounded I get:

Code: [Select]

```
VCC=4.84V
WP=0.01V
SCL=2.34V
SDA=2.40V
```

So I am why grounding WP also pulls down SCL and SDA to ~2.4volts.

I did move the Selector from Off.

Looks like the eeprom's WP pin (2nd from bottom) is connected on top side of pcb to the proccessor. I wonder why?

Update:

The WP is connected to DTM0660 [Pin 13] (PT2.3, TMCK1, LVDIN, Data input/output. RC clock input, LVD external input). It might not be happy to be grounded on this pin.

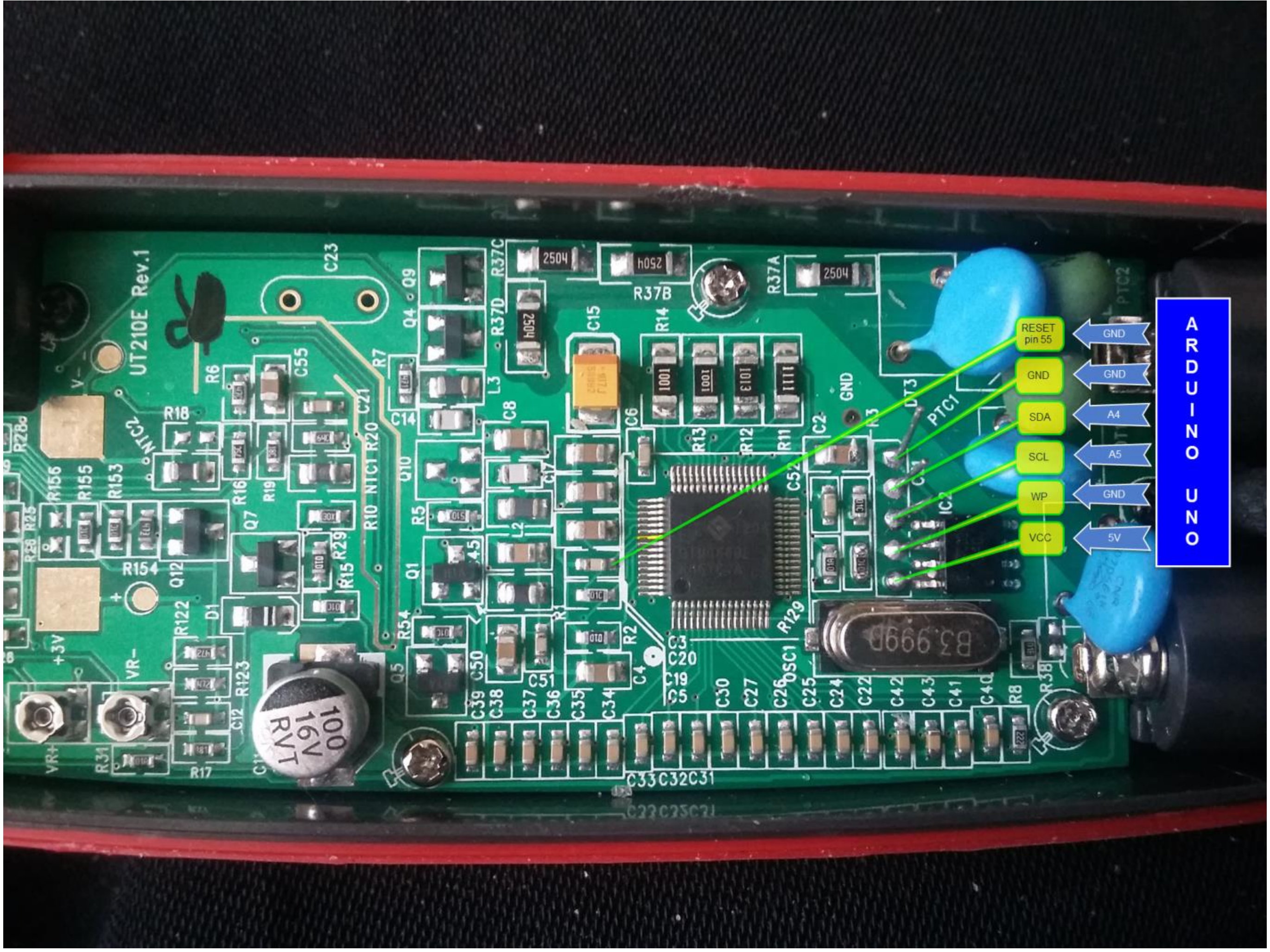
Update: Removed the WP pull-up 10k but the problem still there Also, I noticed with WP not connected the meter beeps twice but with WP grounded it makes two cracking sound, weird.

SUCCESS* Man, I like this UT210E so much more now with these mods. It defaults to DC(v)(a), 6000 count, 3min backlight, 30min Power-off, full 6000 count 2A Dotless mode and NCV extra readings, WoW

However, I could not find the RST pad on my UT210E Opened up the DTM0660 datasheet and found the RST is on pin 55. Looking at PCB I found a Cap I could easily attach the ground to to set RST low.

Here is how I did it with Arduino UNO. I've attached the sketch I used based on Kerry's D source with additions.

I just wired it up as shown in the diagram.



Had to ground WP and Reset in order to write to the EEPROM