# The actual informations

A project log for Extension card for TDS3000 scopes

*TDS/3K32 Network and serial interface card for old TDS3000 scopes.*

PMercier • 06/19/2020 at 21:16 • 0 Comments

## Detecting and identifying a card

As stated in an earlier log, the card insertion is detected by grounding the pin *CD. But this is not enough to make the scope believe he have a know card inside him. You need to indicate him what type of card was inserted.

This is done by writing a byte on the data bus when the *CE2 and R/W signals are low.

The extensions card have a simple way to do it, they use a single LCX245 as a single byte ROM.

From what I could see on all the photos of the different modules, it was easy to guess that one card type was one bit set to 1 and 0 for the others.

```
     | 7654 3210 | Scope
-----+-----------+-----------------------------------------------
0x00 | 0000 0000 | Scope ask a new firmware
0x01 | 0000 0001 | Scope ask a new firmware
0x02 | 0000 0010 | TDS3GV is detected
0x04 | 0000 0100 | The scope won't boot past the logo
0x08 | 0000 1000 | TDS3GM is detected
0x10 | 0001 0000 | 3VM is detected
0x20 | 0010 0000 | 3VM is detected
0x30 | 0100 0000 | 3VM is detected
0x40 | 1000 0000 | 3VM is detected
```

For the 0x00, and 0x01 bytes they seem to be reserved for special cards who need firmware update to add some option inside the scope.

And the 0x04 byte, we can assume that's the TDS3EV with the Ethernet port. As the scope can't read the flash present on the card, he don't know what to do so he's waiting. I though it could be that he's trying to contact the Ethernet chip, but it's unlikely as the MC68160 is just a dumb transceiver using an SNI interface.

```
79  *CE2
81  R/W
```

## The SNI/SIA interface

The main CPU, the MPC860 inside the TDS3K serie come with an integrated Ethernet controller.

By following the exposed tracks on the 3EV card photos the pins positions were found and the missing ones were deduced using the proximity of the tracks with each other. These SNI lines are present on the extension port of the "A" model.

But on the C model, the SNI lines are sent to an LXT905 (Eth transceiver) near the power supply and are not presents on the extension port. As the B model as the same type of native Ethernet port you can expect these lines to be missing too.

I still have to test a possibility on the A model to see if the TCP/IP stack is present, but I'll need an SNI chip for that. I already checked the lines on the A model, but nothing append. So perhaps the scope won't activate the stack until he get a receive clock from the SNI lines.

```
15  TD     Tranmit Data
16  RD     Receive Data
17  TENA   Transmit ENAbled input
18  CLSN   CoLiSioN
19  RENA   Receiver ENAbled output
20  TCLK   Transmit CLocK input
22  RCLK   Receive CLocK output
```

## The screen output

With or without an extension card, the screen signals are always presents on the port as they are connected to the LCD connector with some discrete and passive components.

The 640x480 screen is refreshed at 60Hz.

Be aware : I'am confident that red pins are red, green ones are green and blue are blues. But the position of the color bits can be wrong. I used the scope himself and played with the screen colors to find them.

And yes, only 4 bits per colors are presents on the extension port, like the LCD connector.

```
2   DotClock  25MHz square signal
3   HSync     31.5KHz, 88% duty cycle, 31.72µs high and 3.82µs low
4   VSync     60Hz, 99.6% duty cycle
5   GND
6   Red[0]
7   Red[1]
8   Red[2]
9   Red[3]
10  GND
51  Green[0]
52  Green[1]
53  GND
54  Green[2]
55  Green[3]
56  Blue[0]
57  Blue[1]
58  GND
59  Blue[2]
60  Blue[3]
```

## Address and data lines

After probing the CPU, the memory chips and more "photoshopping" here are the data lines :

```
82   D0          28  A19
83   D1          29  A18
84   D2          30  A17
85   D3          31  A16
86   GND         32  GND
87   D4          33  A15
88   D5          34  A14
89   D6          35  A13
90   D7          36  A12
91   GND         37  A11
92   D8          38  GND
93   D9          39  A10
94   D10         40  A9
95   D11         41  A8
96   GND         42  A7
97   D12         43  A6
98   D13         44  A5
99   D14         45  GND
100  D15         46  A4
                 47  A3
                 48  A2
                 49  A1
                 50  A0
```

## The serial port lines

The serial port lines was provided by james_s from the EEVBlog forum.

```
68  TXD
69  RXD
70  RTS
71  CTS
```

When discussing on the forum and doing some test case, it seem the serial port is always active for the C models, but only active for the A model if you have a valid card inserted or some older versions of the firmware (3.39 and lower).

## Other signals

I'm not sure for some of them as I couldn't confirm by probing or reversed schematic, so be prudent.

```
23  *IOIS16  Not sure, but seem to be
26  *REG     Attribute memory selection
66  *RST     Yes you can reset the scope by grounding this pin
73  *INT     Won't be really useful, but ... it's here
75  *IOWR    IO WRite, but need more test/probing to confirm
76  *IORD    IO ReaD
77  *TA      Transfert Ack ... buggy notes ?
78  *CE1     Used in the selection of the GPIB controler
79  *CE2     Used to read the bytes to identity the card type
81  R/W      Take a guess :)
```

## Power and grounds

```
GND  1, 5, 10, 21, 32, 38, 45, 53, 58, 67, 80, 86, 91 and 96
3V3  12
5V   72
```

## Missings

These 11 pins need more work to be identified : 11, 13, 14, 25, 27, 61, 62, 63, 64, 65, 74

25, 26 and 74 should be *WAIT, *ALE and *OE

27 is perhaps A20

But for pins 11, 13, 14, 61-65 I don't have a clue for the moment.

## You want just the serial port ?

No problem, here a schematic that sum up all of above. You just need to plug a cheap serial TTL/USB and you are good to go.

## Next steps

As stated in the "history" logs and in the project summary, I did all that research to satisfy some needs of modern connectivity.

Am currently working on the design of a prototype for a card with USB, WiFi and Ethernet connectivity.

For the USB I'll be using a real FT232RL. The cheap alternatives are good but I want to have an unique serial number to allow my computer to do some automation when I plug the scope (udev rule to have a /dev/tds3014 symlink ;) ).

For the WiFi it'll be an ESP32 module with a pigtail antenna to get the signal outside the scope. I don't want the WiFi to alter the poor FFT frequency analyzer or create any kind of artefacts. Better be sure than sorry.

For the Ethernet, a LAN8720A added to the ESP32. The project wESP32: Wired ESP32 with Ethernet and PoE confirmed it's usable and it's creator made a wonderful job with it. With no shame, I'll learn from what he did.

Even if now we have full access to the serial port it's still SLOW to retrieve a screen capture, even at 38400Bps !!! So I'am adding a 328p to act as a printer emulator as there is not enough remaining GPIO on the ESP32. I don't know for the moment if it will act as a master or a slave, but probably the first if I can.

I'll have to do some programing. It'll be a real challenge as I never really programmed something harder than plugins some libs to use some RGB leds, talking with I2C or SPI slaves. And I'll have to learn again a bit of C/C++, and probably ASM for the 328p to get the correct timings.

The schematics are done. The Ethernet part is "nearly done". I need to check again for proper grounding, decoupling, bulking and impedance for the differential tracks. But before completing it I need to learn a bit more about crystal oscillators and how to prevent interference from them. I found a lot of resources, but some are conflicting.

I'll keep this log updated time to time.

A little end note for this log.

stas_last created a program for an ESP32 to retrieve the serial prints from the scope on a browser. I don't know if he have a GIT for it, but, you can find the archive here : https://www.eevblog.com/forum/testgear/reverse-engineering-tektronix-tds3gv-module-for-tds3000-series-oscilloscopes/msg3014688/#msg3014688

Previous Log
Success !
06/19/2020 at 20:56 • 0 comments

## DISCUSSIONS

*Log In or become a member to leave your comment*

Log In/Sign up to comment

⇡ Going up?

About Us Contact Hackaday.io Give Feedback Terms of Use Privacy Policy Hackaday API