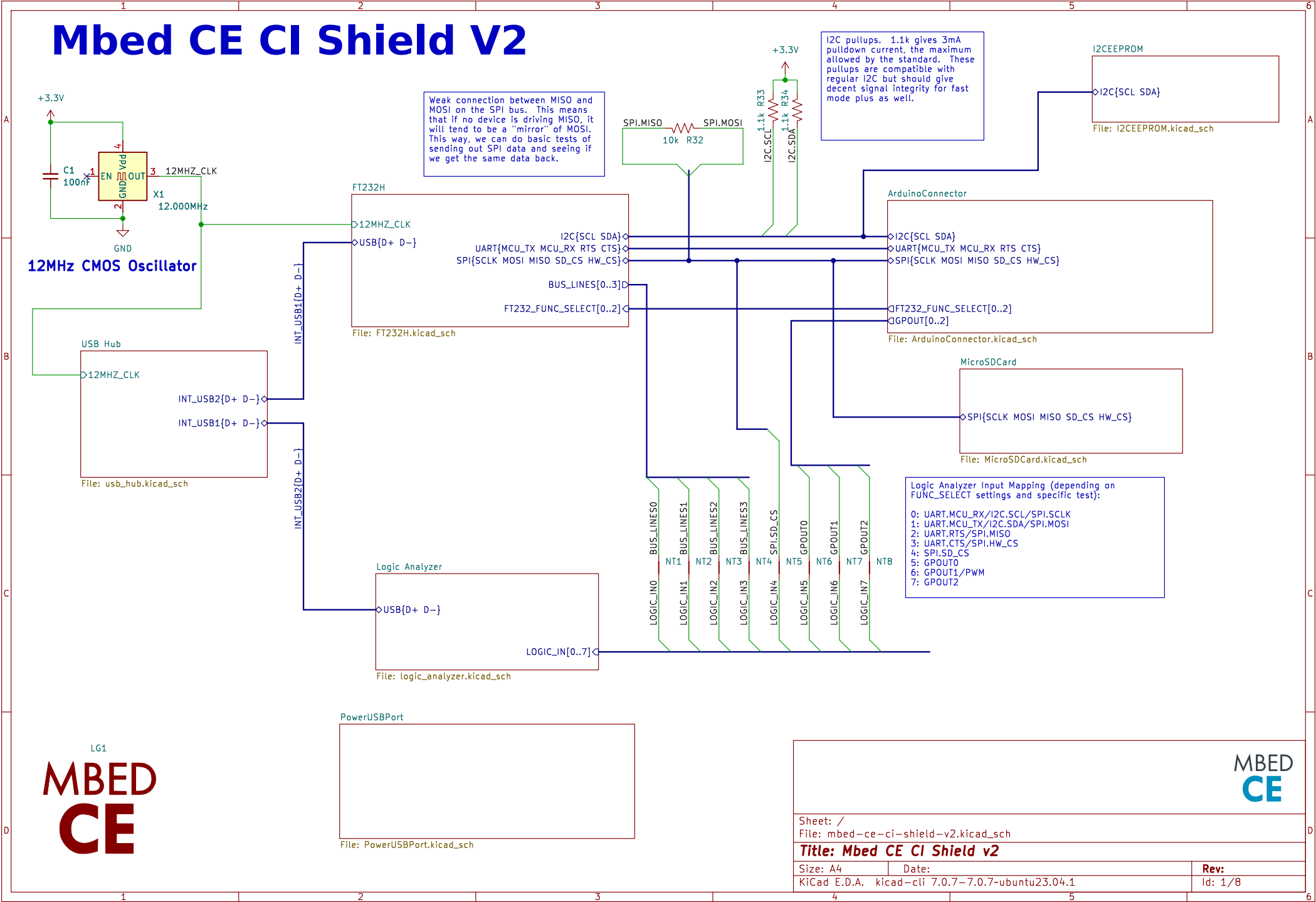
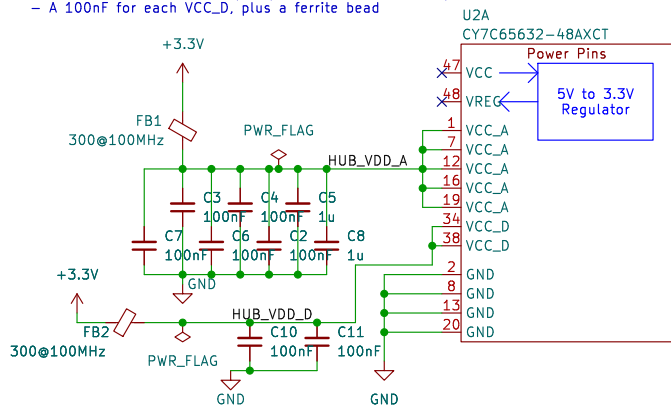


Mbed CE CI Shield V2

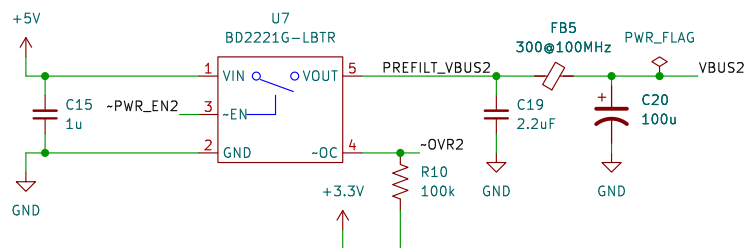
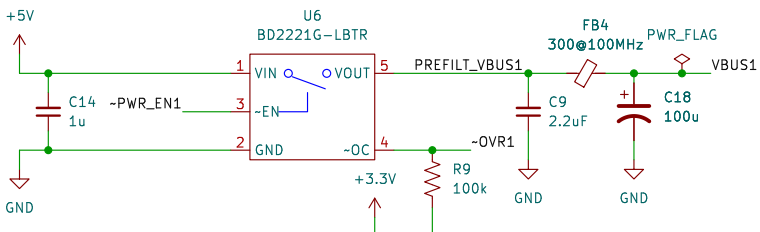


The reference design seems to recommend:

- 1x 10uF and 1x 1uF on VREG (which I assume is needed only if the internal regulator is used)
- A 100nF for each VCC_A pin, plus 2x 1uF on VCC_A, plus a ferrite bead
- A 100nF for each VCC_D, plus a ferrite bead

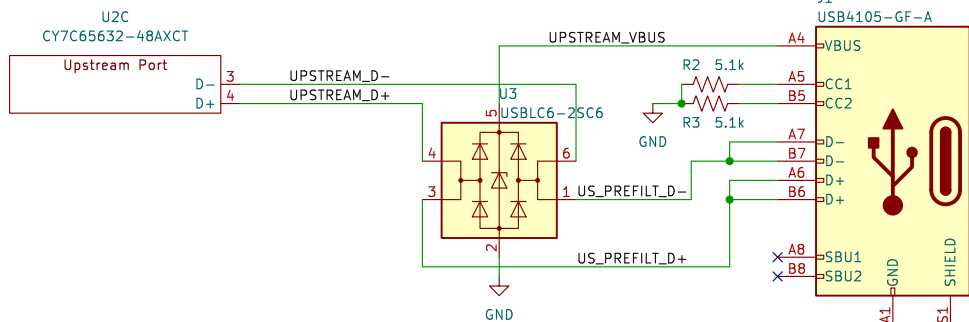


External port power switches

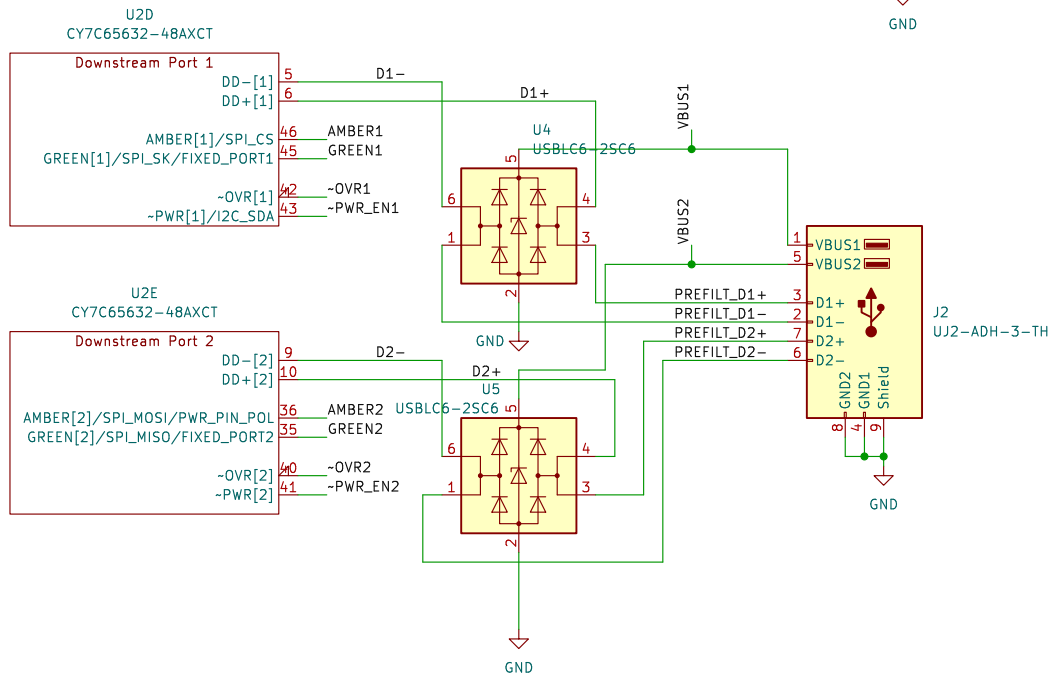


Upstream Port

Upstream port (USB-C, connects to host computer).
This uses the USB connector setup from here:
<https://hackaday.com/2022/12/06/usb-c-introduction-for-hackers/>
This requests 5V@1.5A for our board.



External Ports



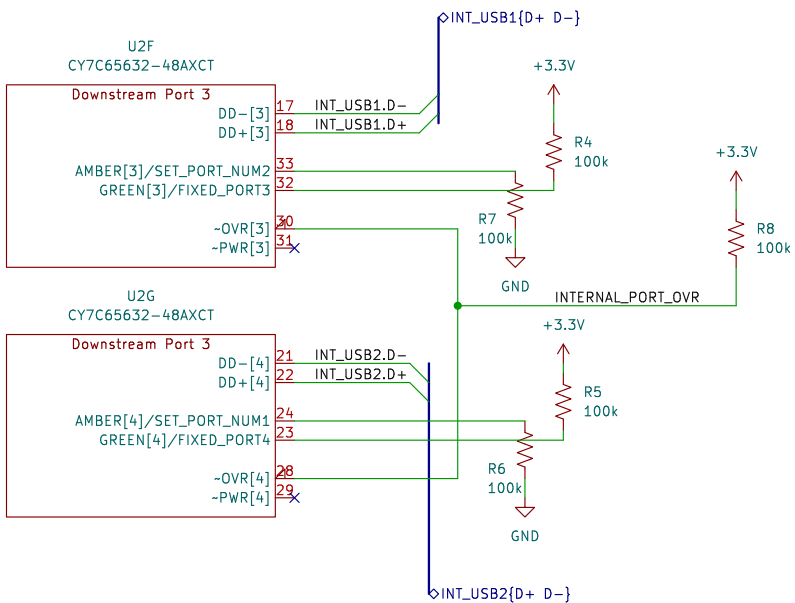
Internal Ports

Internal ports. These ports are used to provide USB to devices inside the CI shield. Due to this, they lack ESD protection or their own power supply.

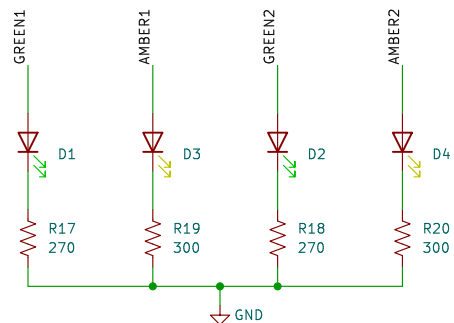
The FIXED_PORT pins are strapped to high to indicate to the hub that these devices are non-removable.

The SET_PORT_NUM[2:1] pins are strapped to 00 to indicate that all 4 ports are in use.

The OVR pins are tied to high to disable overcurrent detection.



Port Indicators



Note: For the GREEN pins, we rely on the internal pulldown resistor to strap them as low. This configures port 1 and port 2 as removable.

Also, for AMBER2, we allow this to be pulled low internally to strap the -PWR[] pins to active-low polarity.

MBED
CE

Sheet: /USB Hub/
File: usb_hub.kicad_sch

Title: USB Hub

Size: A3

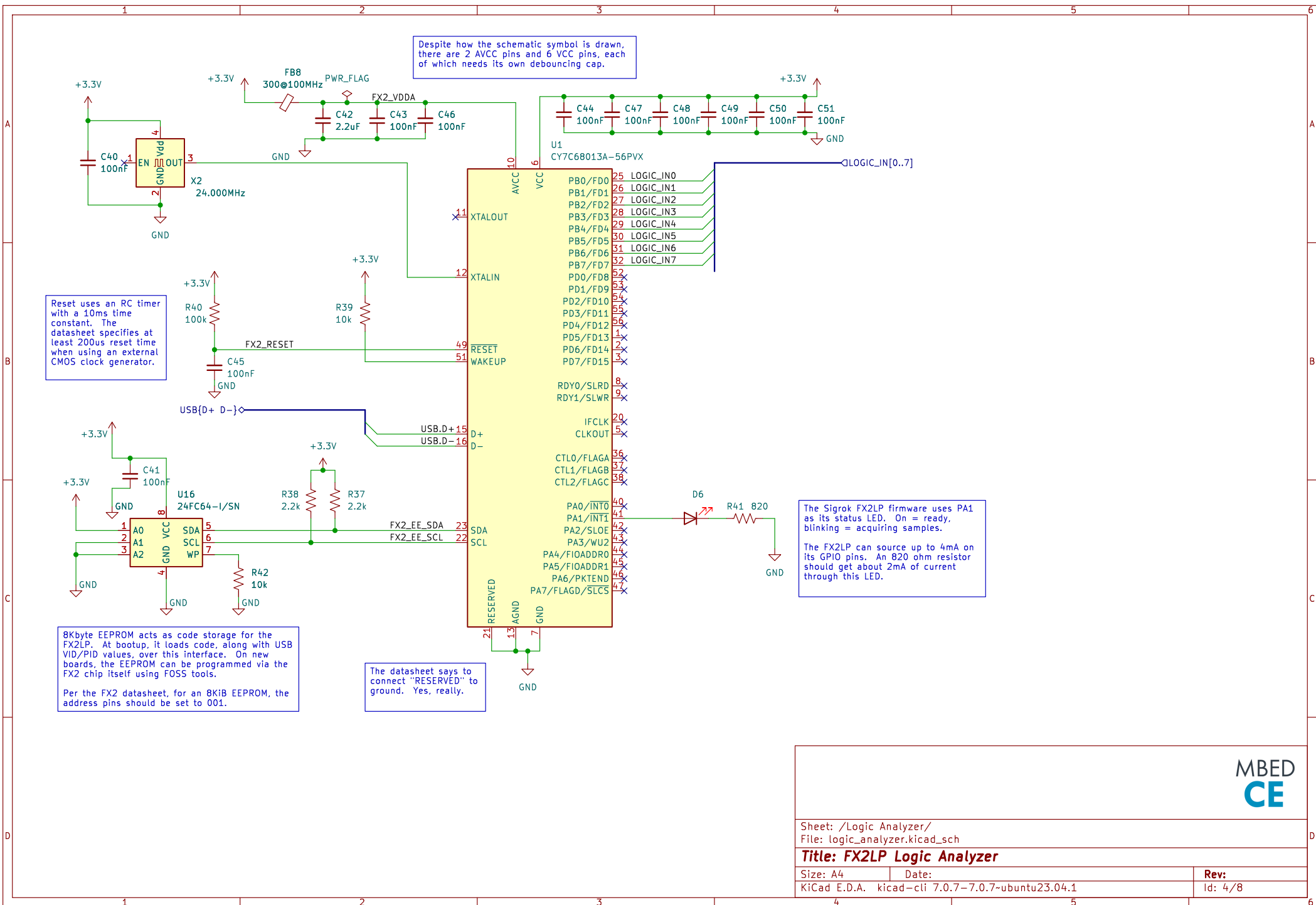
Date:

Rev:

KiCad E.D.A. kicad-cli 7.0.7-7.0.7-ubuntu23.04.1

Id: 2/8



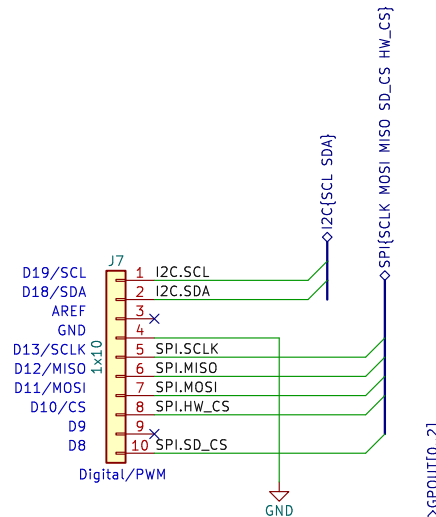
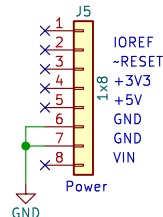
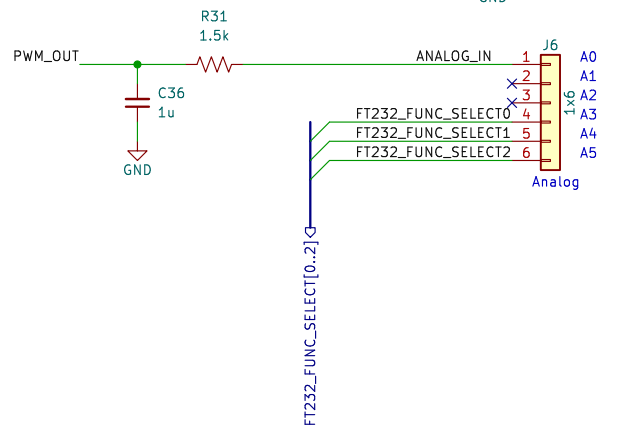


In order to generate analog voltages to test the ADC, we can use a PWM output from the MCU under test itself.

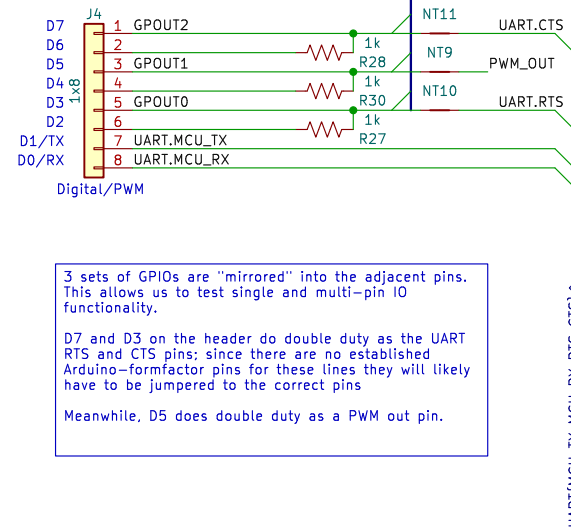
A PWM signal can be effectively averaged into a voltage by an RC filter with a cutoff frequency at 1/100 the PWM frequency.

This filter has a cutoff frequency of 100 Hz, so it can take a 10kHz PWM signal and average it into a constant voltage with a response time of roughly 10ms.

Source: <https://www.allaboutcircuits.com/technical-articles/low-pass-filter-a-pwm-signal-into-an-analog-voltage/>



SPI has 2 chip selects:
 * SD_CS goes to the SD card socket. It's monitored by the logic analyzer but not hooked to the FT232H. This can be any GPIO pin since the SD card driver does not work with HW_CS anyway
 * HW_CS is used for master-mode SPI operations to the logic analyzer, and also slave-mode SPI operations with the FT232H as the master. Since slave mode usually only works with a hardware CS pin, HW_CS should be connected to a hardware CS pin if available.



3 sets of GPIOs are "mirrored" into the adjacent pins. This allows us to test single and multi-pin IO functionality.

D7 and D3 on the header do double duty as the UART RTS and CTS pins; since there are no established Arduino-formfactor pins for these lines they will likely have to be jumpered to the correct pins

Meanwhile, D5 does double duty as a PWM out pin.



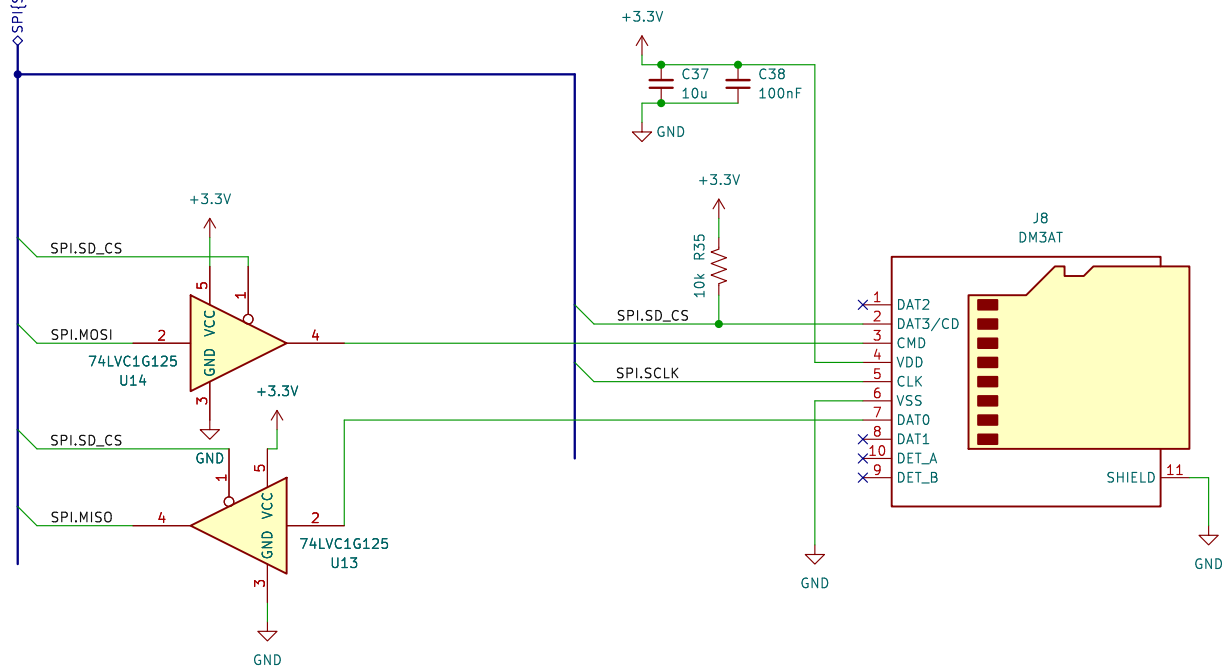
Sheet: /ArduinoConnector/
 File: ArduinoConnector.kicad_sch

Title: Arduino Connector

Size: A4 Date: KiCad E.D.A. kicad-cli 7.0.7-7.0.7-ubuntu23.04.1

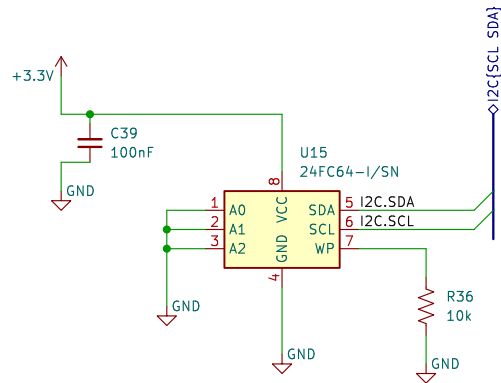
Rev: Id: 5/8

SPI[SCLK MOSI SD_CS HW_CS]



When it is in SD mode (instead of SPI mode), the MicroSD card might try to output on its CMD and DAT0 lines in response to the SPI clock going active, which would interfere with other users of the SPI bus. We guard against this by sending the MOSI and MISO lines through a tri-state buffer, so the card is electrically disconnected from the bus when the CS line is high.

Note: MicroSD card is used in SPI mode, not SD mode, so the pin labels on the symbol don't match up.



This I2C EEPROM provides something to test an MCU's I2C implementation against. There are huge numbers of I2C EEPROMs out there, but I chose this specific one because it supports all three bus speeds in common use: 100kHz, 400kHz, and 1MHz (Fast Mode Plus). Additionally, it's cheap, seems reasonably available, and worked on the previous revision of the board.

If this specific part becomes hard to find, there should be other pin-compatible options, as this form factor of EEPROM is sort of a de-facto standard

MBED
CE

Sheet: /I2CEEPROM/
File: I2CEEPROM.kicad_sch

Title: I2C EEPROM

Size: A5 Date:
KiCad E.D.A. kicad-cli 7.0.7-7.0.7-ubuntu23.04.1

Rev:
Id: 7/8

Power Input USB Port

This port supplies power to the board but not data. A separate port is used to avoid demanding too much power from the host, which might be a single-board computer without too much power to share with USB devices.

3.3V LDO

