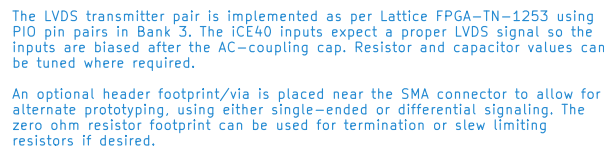
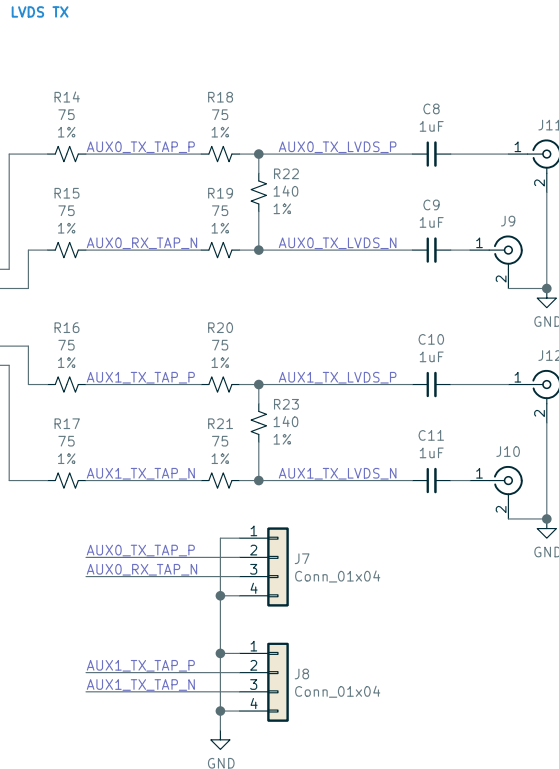


LVDS RX



LVDS TX



Fault pin straps allow setting fixed fault values. The switch attached to ROT_FLT provides a convenient way to cause fault interrupts during integration tests. The header allows reuse of these pins for alternative prototyping.

The diagram shows a connector J14 with pins 1, 3, 5, 7, 9, and 11 connected to a V3P3 supply. Pins 2, 4, 6, 8, 10, and 12 are labeled ID0, ID1, ID2, ID3, ID4, and ID5 respectively. Each ID pin is connected to a common ground line through a 10K resistor (R30 to R35). A text box at the bottom states: "ID straps allow setting the desired value. The header allows reuse of these pins for alternative prototyping".

Diagram showing the connection for Pin 1:

- Pin 1: PWR_EN
- Pin 2: CMD1
- Pin 3: CMD2
- Pin 4: GND

Command bits are exposed on a header for easy test probing and alternative prototyping.

Sheet: Power

Power

File: ignitionlet-power.sch

Sheet: Config

Programming & Configuration

File: ignitionlet-config.sch

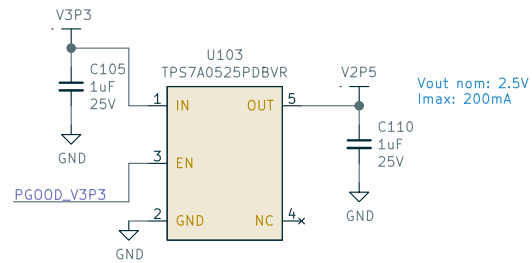
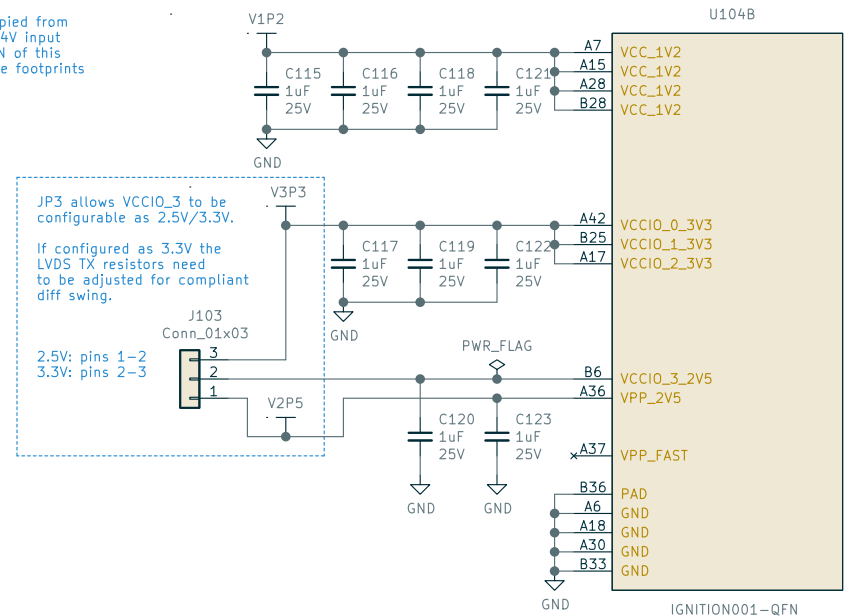
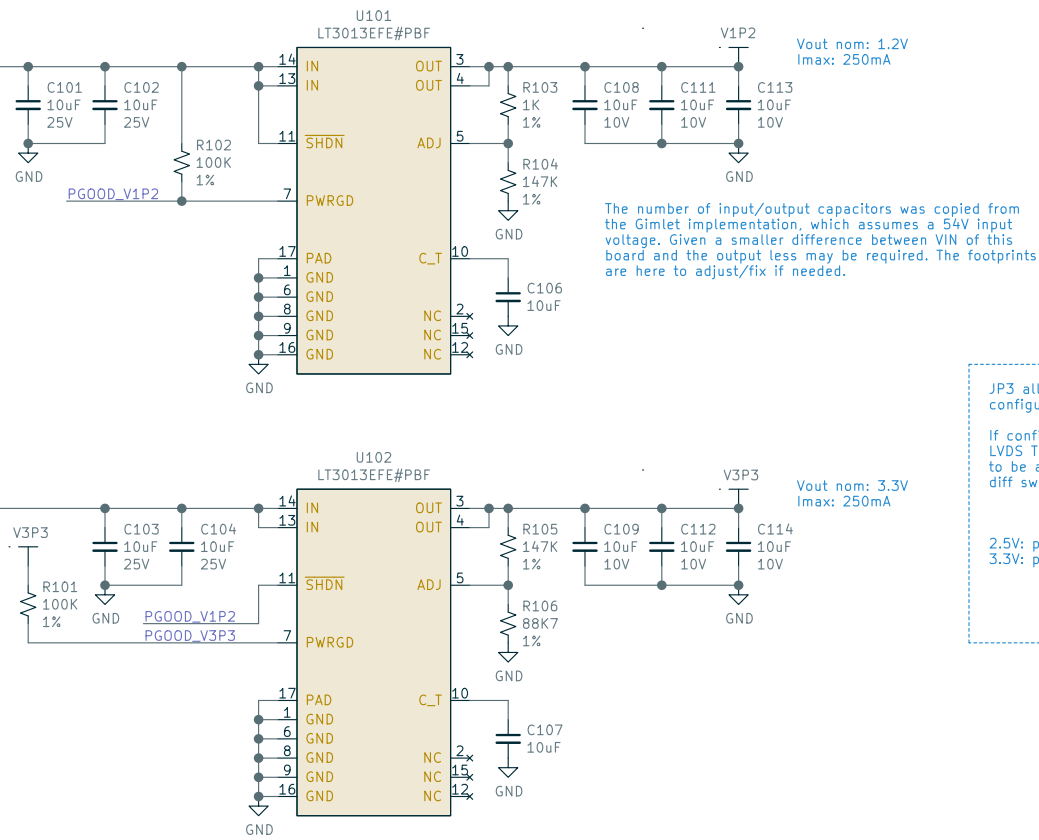
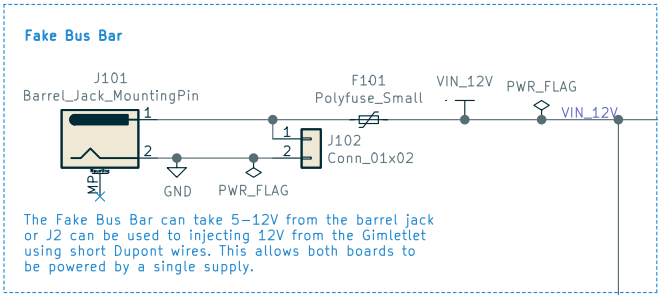
Sheet: /
File: ignitionlet.sch

Title: Ignition Target

Size: A3	Date: 2021-06-18
KiCad E.D.A. kicad (5.1.10-1-10_14)	

Size: A3	Date: 2021-06-18
KiCad E.D.A. kicad (5.1.10-1-10_14)	

Rev: 1
Id: 1/3



Sheet: /Power/
File: ignitionlet-power.sch

Title: Power

Size: A3 Date: 2021-06-18
KiCad E.D.A. kicad (5.1.10-1-10_14)

Rev: 1
Id: 2/3

Operating Modes

- FPGA as SPI master (default)

The default operating mode for this board is with the FPGA acting as SPI master. Without anything driving SPL_SS this signal is pulled high. On init (after PoR or asserting CRESET) the FPGA will sample this pin. With the pin pulled high it will resume its init sequence as SPI master. Consequently it will then assert SPL_SS and drive SPL_SCK, allowing it to read a bitstream from SPI flash and enter the user application.

- Program the SPI flash from IGNITION_TARGET_HDR

The second mode is to program the SPI flash via the IGNITION_TARGET_HDR using an FTDI USB programmer. The programmer will assert both SPL_RESET and SPL_SS, causing the FPGA to go/stay in reset while selecting the SPI flash as peripheral. The programmer then writes to SPI flash as normal.

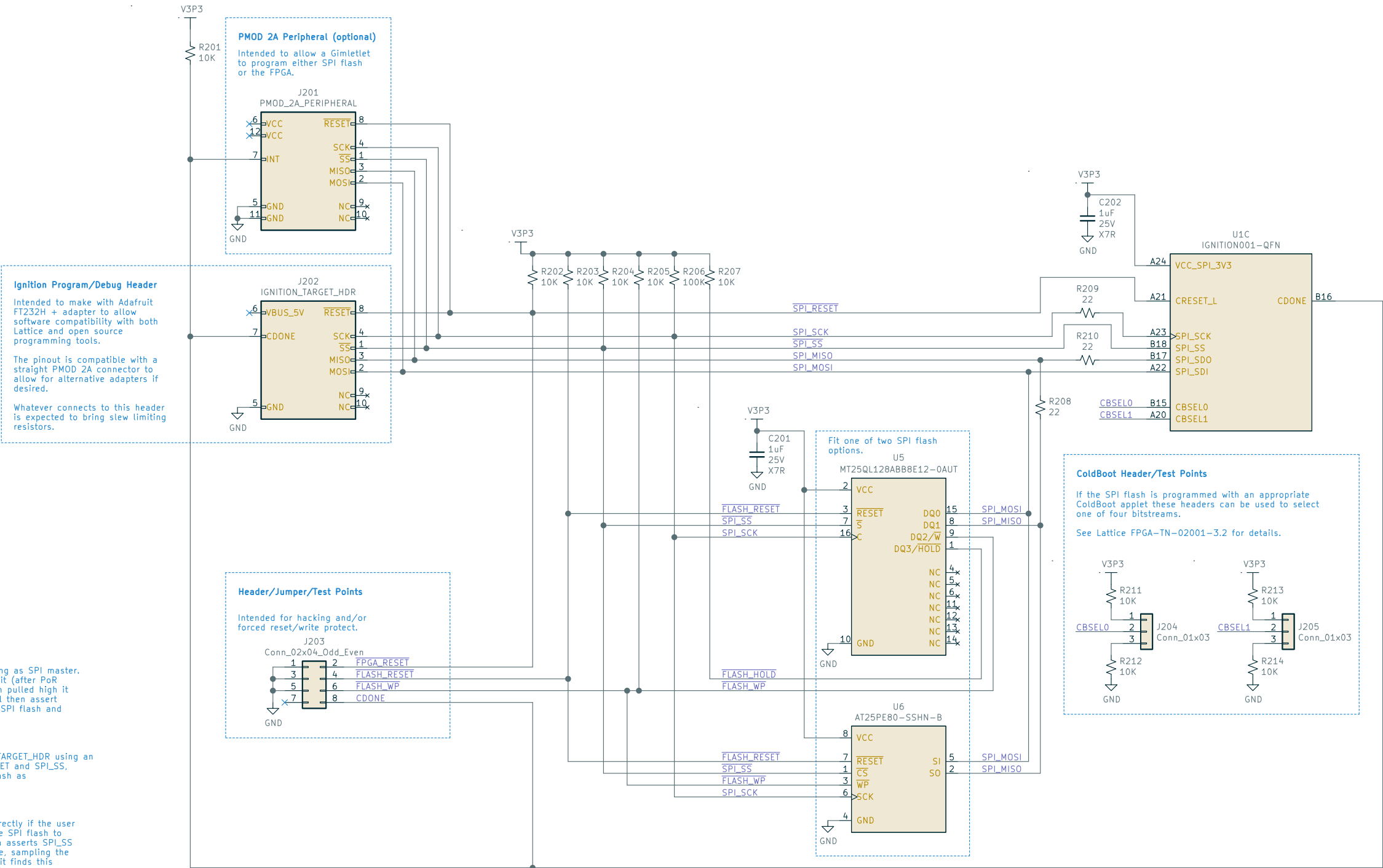
- Program the FPGA SRAM from IGNITION_TARGET_HDR

The FTDI programmer can program the FPGA SRAM (or NVCM) directly if the user installs a jumper on the FLASH_RESET signal. This will cause the SPI flash to remain in reset and ignore any SPI traffic. The programmer then asserts SPL_SS while toggling SPL_RESET. This causes the FPGA to (re-)initialize, sampling the SS pin, and initialize as SPI peripheral instead of master when it finds this signal low. The programmer then writes to the FPGA as per Lattice FPGA–TN–02001–3.2.

Holding the flash in reset can potentially be automated by connecting FLASH_RESET to an additional GPIO pin on the programming adapter and modifying the (open source) programming software to assert this pin during the programming cycle. This may already be supported in software, but is not tested.

- Program the FPGA SRAM from PMOD_2A

In the same way as stated above the PMOD header can be used to program the FPGA SRAM from an attached Gimletlet using software running on the SP. This mode assumes the SPI flash is disabled during programming, either using the described FLASH_RESET jumper, or by connecting the FLASH_RESET signal to an SP GPIO using a Dupont wire, allowing flash reset to happen under software control.



Sheet: /Config/
File: IgnitionLet-config.sch

Title: Programming & Configuration

Size: A3 Date: 2021–06–18
KiCad E.D.A. kicad (5.1.10–1–10_14)

Rev: 1
Id: 3/3