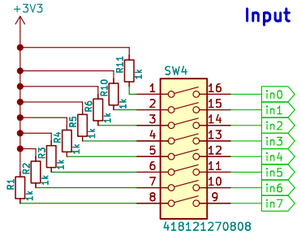
Tiny Tapeout 4/5 Demo Board

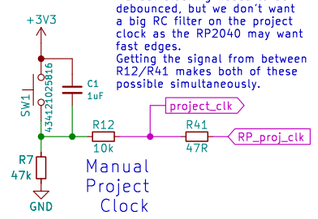
User Input + Config

Input DIP

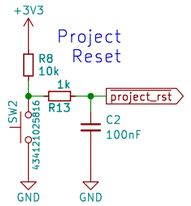


Alt: 76SB08ST

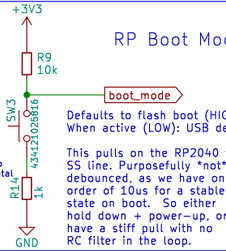
Momentary Switches (debounced)



Manual clocking needs to be debounced, but we don't want a big RC filter on the project clock as the RP2040 may want fast edges.
Getting the signal from between R12/R41 makes both of these possible simultaneously.



RP Boot Mode



Pull-down just to deal with accidental contention on cs.

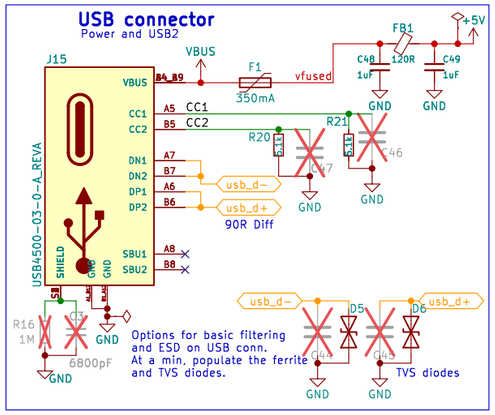
Defaults to flash boot (HIGH)
When active (LOW): USB device

This pulls on the RP2040 flash SS line. Purposefully *not* debounced, as we have on the order of 10us for a stable state on boot. So either hold down + power-up, or have a stiff pull with no RC filter in the loop.

Power

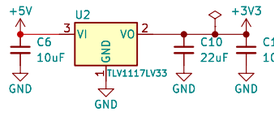
USB connector

Power and USB2



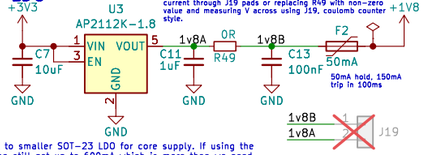
Options for basic filtering and ESD on USB conn.
At a min, populate the ferrite and TVS diodes.

3V3 LDO



Overpowered LDO allows for external projects/PMODs to be supplied from demo board without issue

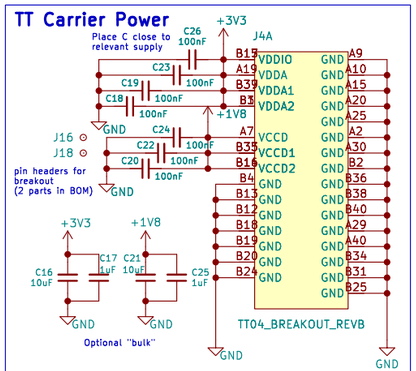
1.8V LDO



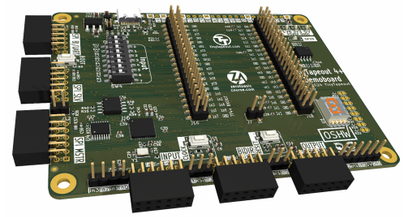
Core power profiling possible by either removing R49 and metering current through J19 pads or replacing R49 with non-zero value and measuring V across using J19. coulomb counter style.

Changed to smaller SOT-23 LDO for core supply. If using the 2112, can still get up to 600mA which is more than we need and should be safe for external draw.

TT Carrier Power



Note: All this decoupling should be amply handled by the carrier. Would rather DNP than regret.

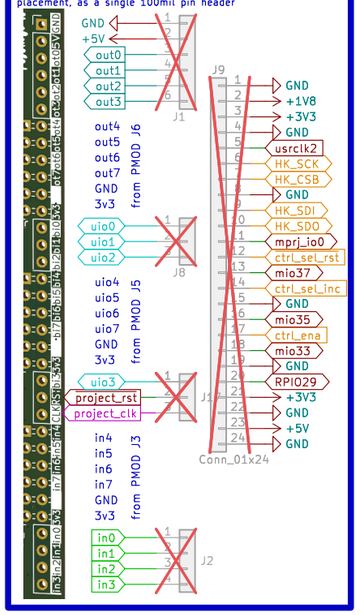


Extensive labelling, designed for TT4+ with new MUX, RP2040 on board, accessed via USB.
Project clock from RP2040, external or manual, DIP switches for inputs, 7-segment display (remappable with jumpers) on outputs, full access to 8 in, out and bidirectionals via PMODs, all pins broken out in headers.

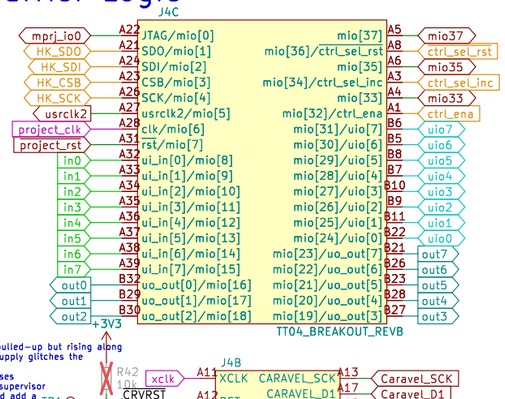
Power via +5V USB, or 5V breakout pin. On-board regulation to 3v3 and 1v8. VDDIO is 3v3, including on PMODs. https://github.com/TinyTapeout/tt-demo-pcb

Board edge connectors (pinheaders)

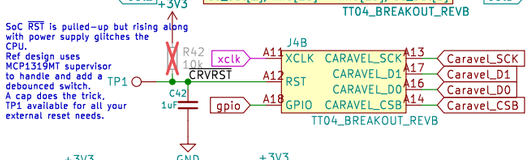
I/O is available through standard PMODs or, with judicious placement, as a single 100mil pin header

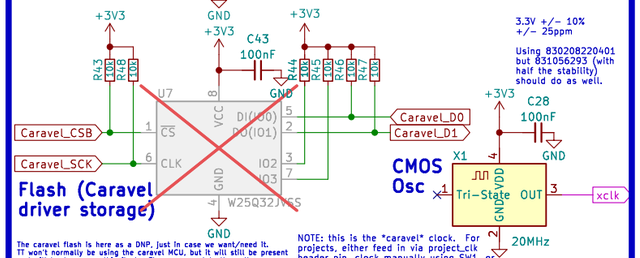


TT Carrier Logic



SoC $\overline{RST}$ is pulled-up but rising along with power supply glitches the CPU.
Ref design uses MCP1319MT supervisor to handle and add a debounced switch.
A cap does the trick.
TP1 available for all your external reset needs.





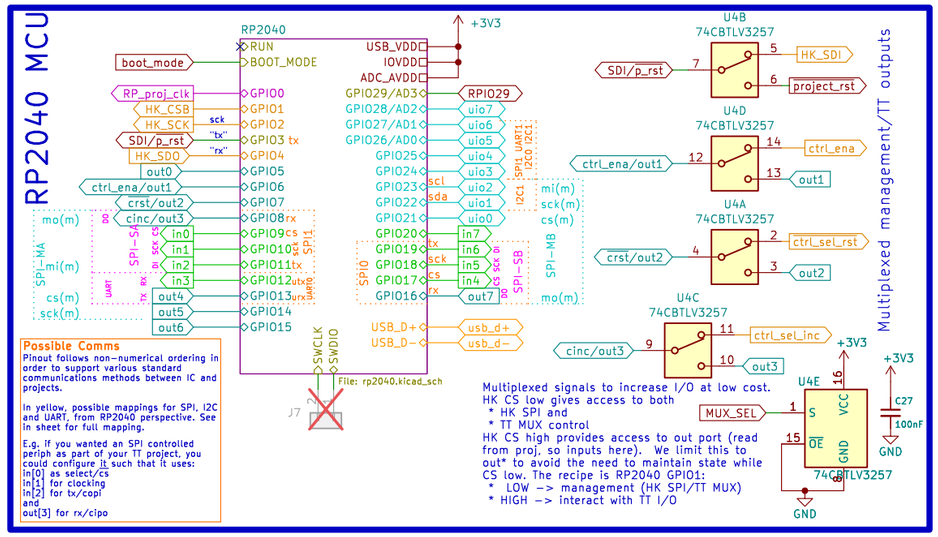
3.3V +/- 10% +/- 25ppm

Using 830208220401 but 831056293 (with half the stability) should do as well.

The caravel flash is here as a DNP, just in case we want/need it. TT won't normally be using the caravel MCU, but it will still be present and will try to access this flash. Thus, we populate the pull-ups regardless of flash stuff status, and have some stable values on the lines at all times.

NOTE: this is the *caravel* clock. For projects, either feed in via project_clk header pin, clock manually using SW1, or use the RP2040 RP_proj_clk pin.

RP2040 MCU



Possible Comms

Pinout follows non-numerical ordering in order to support various standard communications methods between IC and projects.

In yellow, possible mappings for SPI, I2C and UART, from RP2040 perspective. See in sheet for full mapping.

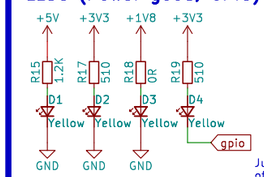
E.g. if you wanted an SPI controlled periph as part of your TT project, you could configure it such that it uses:
in[0] as select/cs
in[1] for clocking
in[2] for tx/copi
and
out[3] for rx/cipo

Multiplexed management/TT outputs

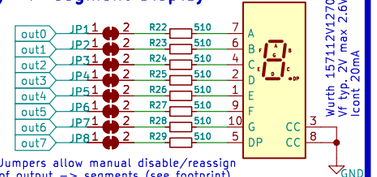
Multiplexed signals to increase I/O at low cost.
HK CS low gives access to both
* HK SPI and
* TT MUX control
HK CS high provides access to out port (read from proj. so inputs here). We limit this to out* to avoid the need to maintain state while CS low. The recipe is RP2040 GPIO1:
* LOW -> management (HK SPI/TT MUX)
* HIGH -> interact with TT I/O

Indication

LEDs (Power good, GPIO)



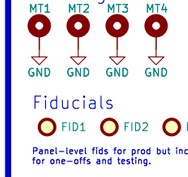
7-segment Display



Jumpers allow manual disable/reassign of output -> segments (see footprint)

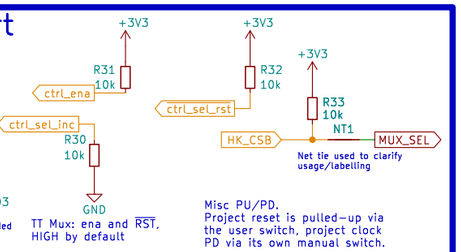
Misc Support

Mounting holes



Fiducials

Panel-level fids for prod but included for one-offs and testing.

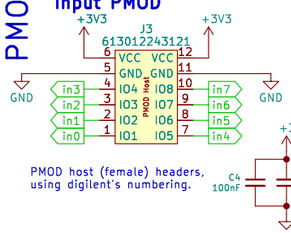


TT Mux: ena and $\overline{RST}$, HIGH by default

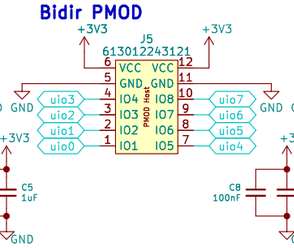
Misc PU/PD.
Project reset is pulled-up via the user switch, project clock PD via its own manual switch.

PMOD

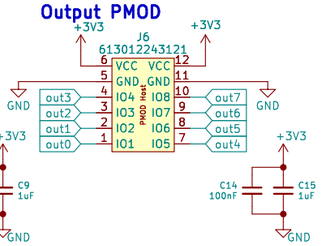
Input PMOD



Bidir PMOD



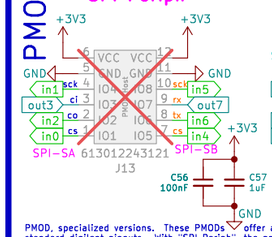
Output PMOD



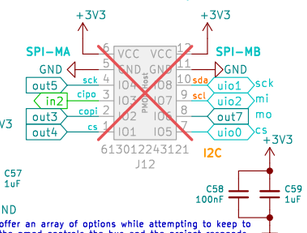
PMOD host (female) headers, using digilent's numbering.

PMOD

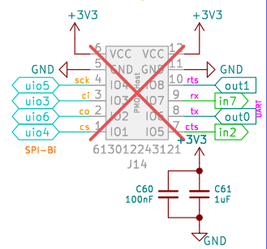
SPI Periph



SPI Control/I2C



SPI Bi/UART



PMOD, specialized versions. These PMODs offer an array of options while attempting to keep to standard digilent pinouts. With "SPI Periph", the pmod controls the bus and the project responds. On "SPI Control", the pmod is a peripheral device to the project. The bidir/uart header is flexible.

Psychogenic Technologies

Sheet: /
File: tinytapeout-demo.kicad_sch

Title: Tiny Tapeout 4/5 Demo Board

Size: A3	Date: 2024-04-12	**Rev: 1.2.2**
KiCad E.D.A. 8.0.8		Id: 1/2

RP2040 Basic Support

IOVDD

Logic supply, nominally 3v3.

BOOT_MODE — QSPI_SS

When held low on powerup, flash SS determines boot mode (HIGH == flash boot, LOW == USB device)

USB_VDD supplies USB PHY, nominal 3v3. If IOVDD is 3v3, can share supply.

In fact, in this and many applications, IOVDD, USB_VDD and ADC_AVDD are all powered directly from a single 3v3 supply, with the 1v1 digital core being handle by on-board regulator.

VREG_VOUT: Int core regulator, 1.1V
Can supply DVDD
Place 1uF in/out bypass near pin.

C32 1uF 1v1 — C37 1uF

R36 10k — RUN async reset

J10 Short to hold in reset

J11 — QSPI_SS, QSPI_CLK, QSPI_SD0, QSPI_SD1, +3V3

Flash program header
Note: should we replace 3v3 with RUN, to be able to reset/hold while updating flash?

WE 830108206909:
CFPX-180 model
10 ppm tol, 20ppm stab
CL 8pF

C29 6pF — Y1 12.0MHz — C30 6pF — R37 1k

Rule of thumb
C1, C2 = 2 * CL - 2 * Cstray
Using a stray cap of 5pF, gives
Cn = 6pF

Into:
CL = (C1 * C2) / (C1 + C2) + Cstray
These Cn = 6p give
CL = 8pF -- just what we need.

U6 RP2040

	Function								
GPIO	F1	F2	F3	F4	F5	F6	F7	F8	F9
0	SPI0 RX	UART0 TX	I2C0 SDA	PWM0 A	SIO	PIO0	PIO1		USB OVCUR DET
1	SPI0 CSn	UART0 RX	I2C0 SCL	PWM0 B	SIO	PIO0	PIO1		USB VBUS DET
2	SPI0 SCK	UART0 CTS	I2C1 SDA	PWM1 A	SIO	PIO0	PIO1		USB VBUS EN
3	SPI0 TX	UART0 RTS	I2C1 SCL	PWM1 B	SIO	PIO0	PIO1		USB OVCUR DET
4	SPI0 RX	UART1 TX	I2C0 SDA	PWM2 A	SIO	PIO0	PIO1		USB VBUS DET
5	SPI0 CSn	UART1 RX	I2C0 SCL	PWM2 B	SIO	PIO0	PIO1		USB VBUS EN
6	SPI0 SCK	UART1 CTS	I2C1 SDA	PWM3 A	SIO	PIO0	PIO1		USB OVCUR DET
7	SPI0 TX	UART1 RTS	I2C1 SCL	PWM3 B	SIO	PIO0	PIO1		USB VBUS DET
8	SPI1 RX	UART1 TX	I2C0 SDA	PWM4 A	SIO	PIO0	PIO1		USB VBUS EN
9	SPI1 CSn	UART1 RX	I2C0 SCL	PWM4 B	SIO	PIO0	PIO1		USB OVCUR DET
10	SPI1 SCK	UART1 CTS	I2C1 SDA	PWM5 A	SIO	PIO0	PIO1		USB VBUS DET
11	SPI1 TX	UART1 RTS	I2C1 SCL	PWM5 B	SIO	PIO0	PIO1		USB VBUS EN
12	SPI1 RX	UART0 TX	I2C0 SDA	PWM6 A	SIO	PIO0	PIO1		USB OVCUR DET
13	SPI1 CSn	UART0 RX	I2C0 SCL	PWM6 B	SIO	PIO0	PIO1		USB VBUS DET
14	SPI1 SCK	UART0 CTS	I2C1 SDA	PWM7 A	SIO	PIO0	PIO1		USB VBUS EN
15	SPI1 TX	UART0 RTS	I2C1 SCL	PWM7 B	SIO	PIO0	PIO1		USB OVCUR DET
16	SPI0 RX	UART0 TX	I2C0 SDA	PWM0 A	SIO	PIO0	PIO1		USB VBUS DET
17	SPI0 CSn	UART0 RX	I2C0 SCL	PWM0 B	SIO	PIO0	PIO1		USB VBUS EN
18	SPI0 SCK	UART0 CTS	I2C1 SDA	PWM1 A	SIO	PIO0	PIO1		USB OVCUR DET
19	SPI0 TX	UART0 RTS	I2C1 SCL	PWM1 B	SIO	PIO0	PIO1		USB VBUS DET
20	SPI0 RX	UART1 TX	I2C0 SDA	PWM2 A	SIO	PIO0	PIO1	CLOCK GPIN0	USB VBUS EN
21	SPI0 CSn	UART1 RX	I2C0 SCL	PWM2 B	SIO	PIO0	PIO1	CLOCK GPOUT0	USB OVCUR DET
22	SPI0 SCK	UART1 CTS	I2C1 SDA	PWM3 A	SIO	PIO0	PIO1	CLOCK GPIN1	USB VBUS DET
23	SPI0 TX	UART1 RTS	I2C1 SCL	PWM3 B	SIO	PIO0	PIO1	CLOCK GPOUT1	USB VBUS EN
24	SPI1 RX	UART1 TX	I2C0 SDA	PWM4 A	SIO	PIO0	PIO1	CLOCK GPOUT2	USB OVCUR DET
25	SPI1 CSn	UART1 RX	I2C0 SCL	PWM4 B	SIO	PIO0	PIO1	CLOCK GPOUT3	USB VBUS DET
26	SPI1 SCK	UART1 CTS	I2C1 SDA	PWM5 A	SIO	PIO0	PIO1		USB VBUS EN
27	SPI1 TX	UART1 RTS	I2C1 SCL	PWM5 B	SIO	PIO0	PIO1		USB OVCUR DET
28	SPI1 RX	UART0 TX	I2C0 SDA	PWM6 A	SIO	PIO0	PIO1		USB VBUS DET
29	SPI1 CSn	UART0 RX	I2C0 SCL	PWM6 B	SIO	PIO0	PIO1		USB VBUS EN

in2 GPIO11, 1RTS
out0 GPIO 5, 1RX
in7 GPIO20, 1TX
out1 GPIO6, 1CTS

Flash

Note: SS pulled-up externally, from bootmode switch

R34 10k — C31 100nF — R35 10k, R38 10k, R39 10k, R40 10k

U5 W25Q32JVSS — QSPI_SS, QSPI_CLK, DI(IO0), DO(IO1), IO2, IO3 — QSPI_SD0, QSPI_SD1, QSPI_SD2, QSPI_SD3

Quad SPI requires QE bit in status register-2 to be set.
In this case, WP becomes IO2 and HOLD becomes IO3.

Supply bypass, place near 1, 10, 22, 33, 42, 49

C33, C35, C38, C34, C36, C39 100nF

Supply bypass, near 23, 50

C40, C41 100nF

Psychogenic Technologies

Sheet: /RP2040/
File: rp2040.kicad_sch

Title: RP2040 Basic Support

Size: A4 | Date: 2023-11-22 | **Rev: 1.0.3**
KiCad E.D.A. 8.0.8 | Id: 2/2