

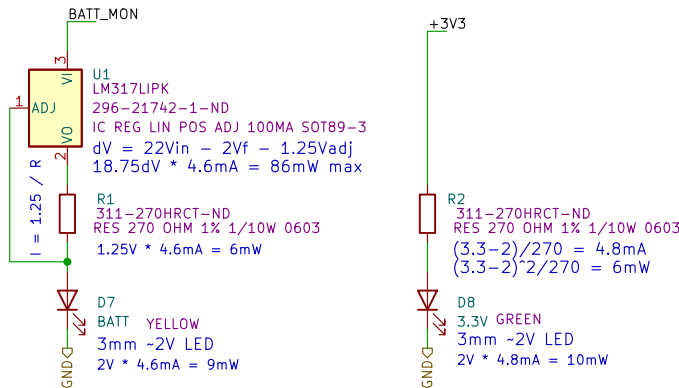
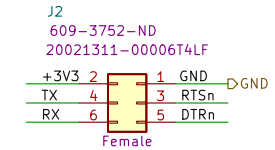
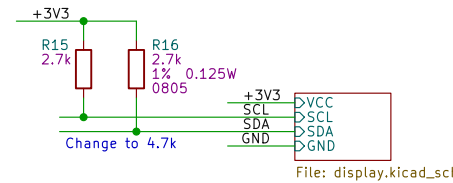
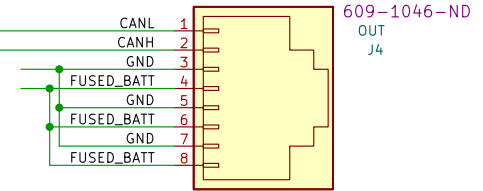
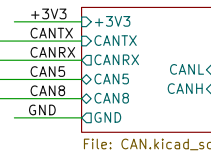
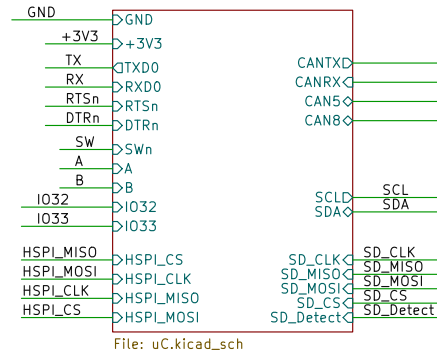
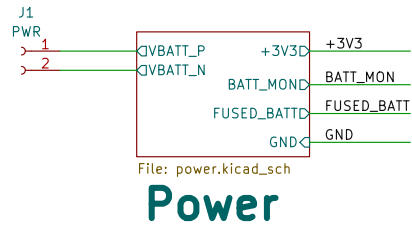
WiFi CAN & ESP32 Breakout

CAN / Display / Knob / uSD Card / UART

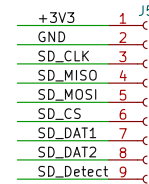
Changes 2023APR04:
 - Power good LED for BATT_MON using LM317 to limit current over wide input range.
 - Increased R10 to Q3B base to ensure IO0 is sufficiently pulled up during boot.
 - R15 & R16 reduced to 2.7k to allow faster I2C/longer wires

7~22VDC

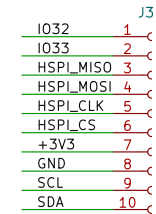
- Vehicle 12V Lead-Acid
- Ryobi 18V Battery
- 9V battery



Pin alignment for:
<https://www.adafruit.com/product/4682>
 Micro SD SPI or SDIO Card Breakout.



Quadrature Knob w/ SW



SDCARD

QENC

AUX

- H1 MountingHole-Mechanical
- H2 MountingHole-Mechanical
- H3 MountingHole-Mechanical
- H4 MountingHole-Mechanical

(c) Joshua Alden 2023

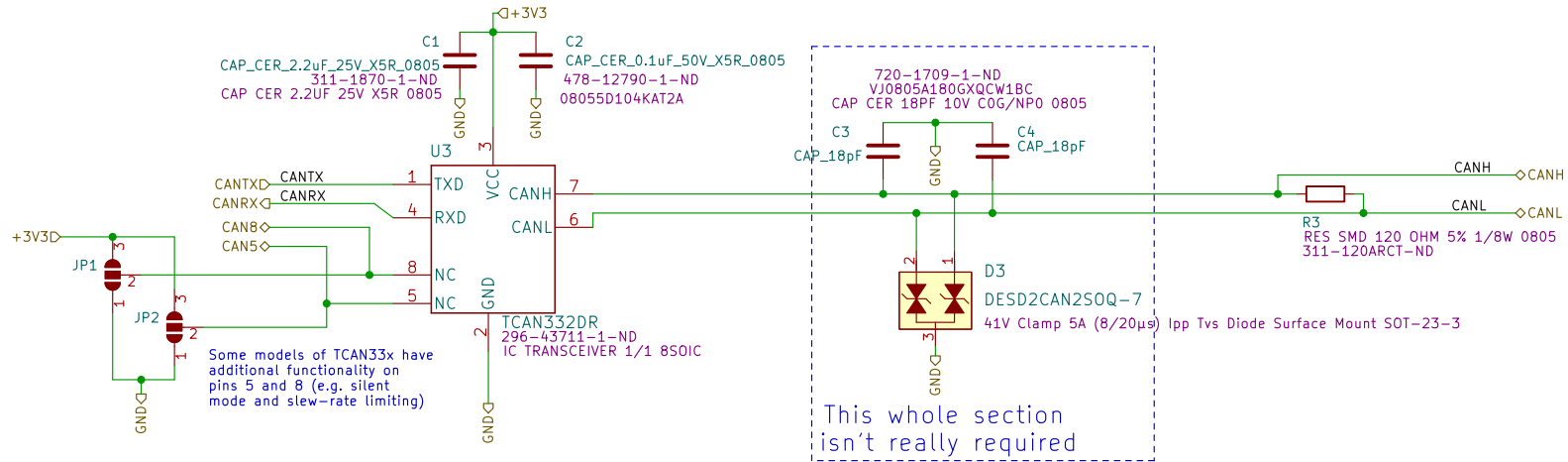
Sheet: /
 File: Esp32CAN.kicad_sch

Title: WiFi CAN

Size: A4 Date: 2023-04-04
 KiCad E.D.A. kicad 6.0.4-6f826c9f35-116-ubuntu18.04.1

Rev: 0.1
 Id: 1/6

CAN Transceiver



(c) Joshua Alden 2023

Sheet: /CAN/
 File: CAN.kicad_sch

Title: CAN Transceiver

Size: A4
 KiCad E.D.A. kicad 6.0.4-6f826c9f35-116-ubuntu18.04.1

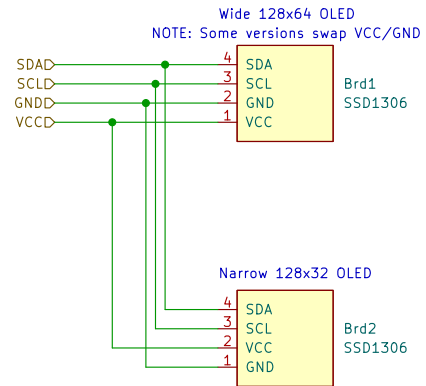
Date: 2023-04-04

Rev: 0.1

Id: 2/6

I2C Displays

These are often sold as cheap display modules online. They seem to occasionally swap VCC and GND, so keep options open to allow for swapping these.



(c) Joshua Alden 2023

Sheet: /display/
File: display.kicad_sch

Title: Display

Size: A4 Date: 2023-04-04
KiCad E.D.A. kicad 6.0.4-6f826c9f35-116-ubuntu18.04.1

Rev: 0.1
Id: 3/6

Microcontroller And Pinout

Module Only

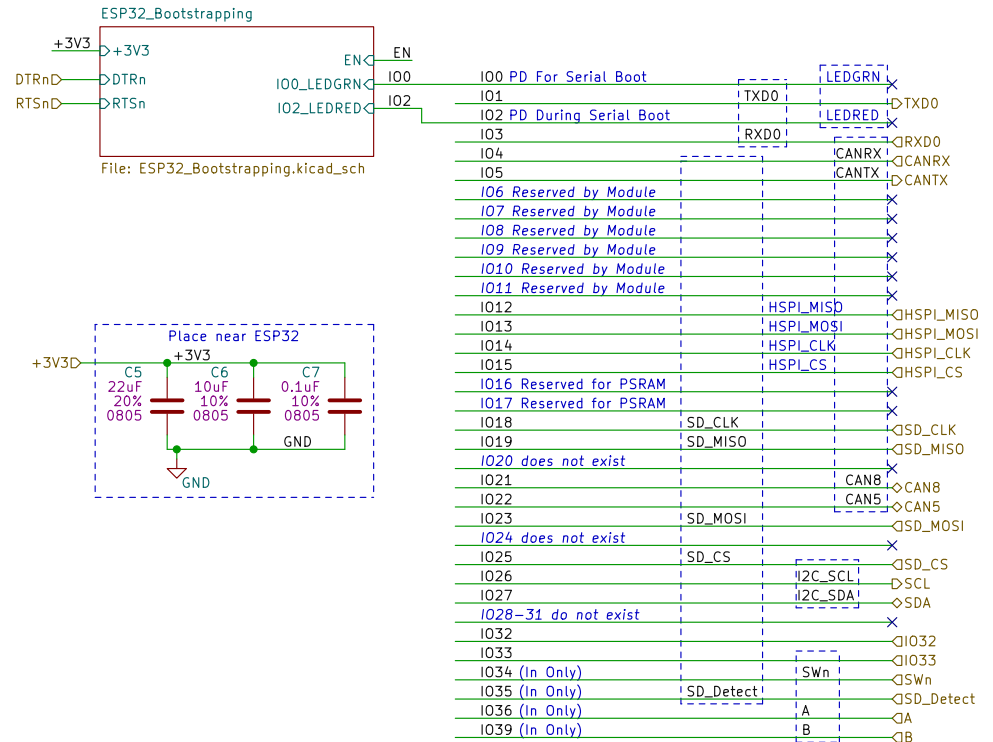
U2A1 ESP32-WROVER-E		GND_THERMAL		GND	
GND	1	GND	39	GND	
+3V3	2	3V3	38	GND	
EN	3	EN/CHIP_PU	37	I023	
I036	4	I036/SENSOR_VP/ADC1_CH0	36	I022	
I039	5	I039/SENSOR_VN/ADC1_CH3	35	I01	
I034	6	I034/ADC1_CH6	34	I03	
I035	7	I035/ADC1_CH7	33	I021	
I032	8	I032/32K_XP/ADC1_CH4	32	NC	
I033	9	I033/32K_XN/ADC1_CH5	31	I019	
I025	10	I025/DAC_1/ADC2_CH8	30	I018	
I026	11	I026/DAC_2/ADC2_CH9	29	I05	
I027	12	I027/ADC2_CH7	28	I017	
I014	13	I014/MTMS/ADC2_CH6	27	I016	
I012	14	I012/MTDI/ADC2_CH5	26	I04	
GND	15	GND	25	I00	
I013	16	I013/MTCK/ADC2_CH4	24	I02	
NC	109	I09_NC(1)	23	I015	
NC	I010	I010_NC(1)	22	I08	NC
NC	I011	I011_NC(1)	21	I07	NC
			20	I06	NC

*I012: Controls PSRAM LDO; *MUST* Float or PD at bootup for WROVER-(i)E
 *I00: PD for boot mode; float or PU for normal mode
 *I02: Float or PD if I00 is low, else undocumented test modes
 *I015: Float/PU for U0TXD Debug Info; PD for silent boot
 *I05: PU default; mostly doesn't matter unless using SDIO slave-boot: Might PWM at boot
 NC(1): GPIO6 to GPIO11 are connected to the SPI flash integrated on the WROVER-(i)E module
 NC(2): GPIO16 and GPIO17 are used for PSRAM integrated on the WROVER-(i)E module
 ADC2 pins do not operate properly as ADC with WIFI active
 ADC readings are non-linear and weird near rails
 RTC Wakeup IO: 0,2,4,12-15,25-27,32-36,39
 2x I2C; Any pin; Arduino defaults to I021/SDA1 & I022/SCL1
 VSPi Default: I023/MOSI, I019/MISO, I018/CLK, I05/CS
 HSPi Default: I013/MOSI, I012/MISO, I014/CLK, I015/CS
 IO Output High/PWM During Boot: 1, 3, 5, 6-11, 14, 15

U2B1
ESP32-DevKitC

+3V3	1	3V3	38	GND	
EN	2	CHIP_PU	37	I023	
I036	3	SENSOR_VP/GPIO36/ADC1_CH0	36	I022	
I039	4	SENSOR_VN/GPIO39/ADC1_CH3	35	I01	
I034	5	VDET_1/GPIO34/ADC1_CH6	34	I03	
I035	6	VDET_2/GPIO35/ADC1_CH7	33	I021	
I032	7	32K_XP/GPIO32/ADC1_CH4	32	GND	
I033	8	32K_XN/GPIO33/ADC1_CH5	31	I019	
I025	9	DAC_1/ADC2_CH8/GPIO25	30	I018	
I026	10	DAC_2/ADC2_CH9/GPIO26	29	I05	
I027	11	ADC2_CH7/GPIO27	28	I017	
I014	12	MTMS/GPIO14/ADC2_CH6	27	I016	
I012	13	*MTDI/GPIO12/ADC2_CH5	26	I04	
GND	14	GND	25	I00	
I013	15	MTCK/GPIO13/ADC2_CH4	24	I02	
I09	16	SD_DATA2/GPIO9	23	I015	
I010	17	SD_DATA3/GPIO10	22	I08	NC
NC	I011	CMD	21	I07	NC
+5V	19	5V	20	I06	NC

DevKitC - DNI



Probably Review ESP-WROVER-KIT_SCH-3

Seems to have some of the extras, like pulling I02 down to avoid undocumented test modes.
</home/junknerd/Electronics/datasheets/ESP32 Stuff/>

Review my old DevKitTestProject for one I used w/ a TFT display, I2C, and a few other things.

</home/junknerd/esp/MyProjects/DevkitTestProject/main/>

Review firework control code for CAN pin assignment.

</home/junknerd/esp/MyProjects/DevkitTestProject/main/>

(c) Joshua Alden 2023

Sheet: /uC/
 File: uC.kicad_sch

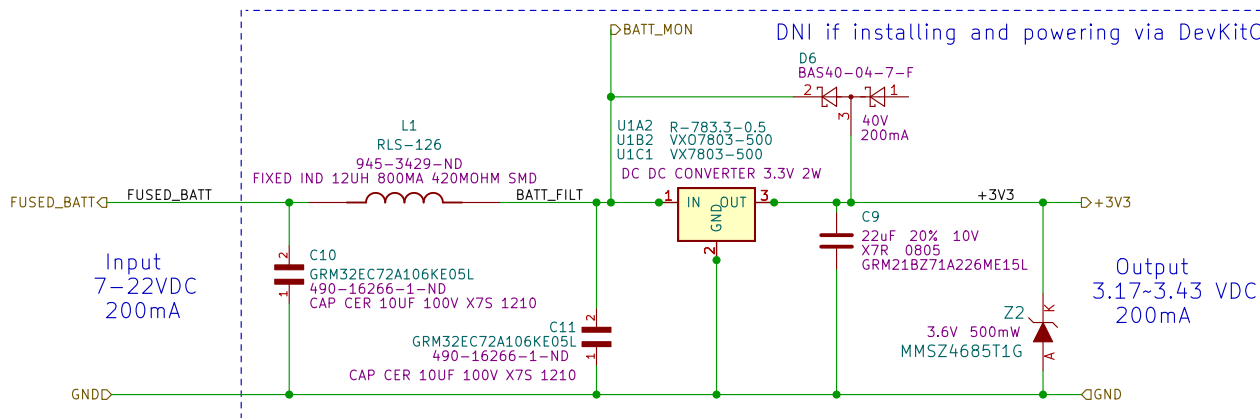
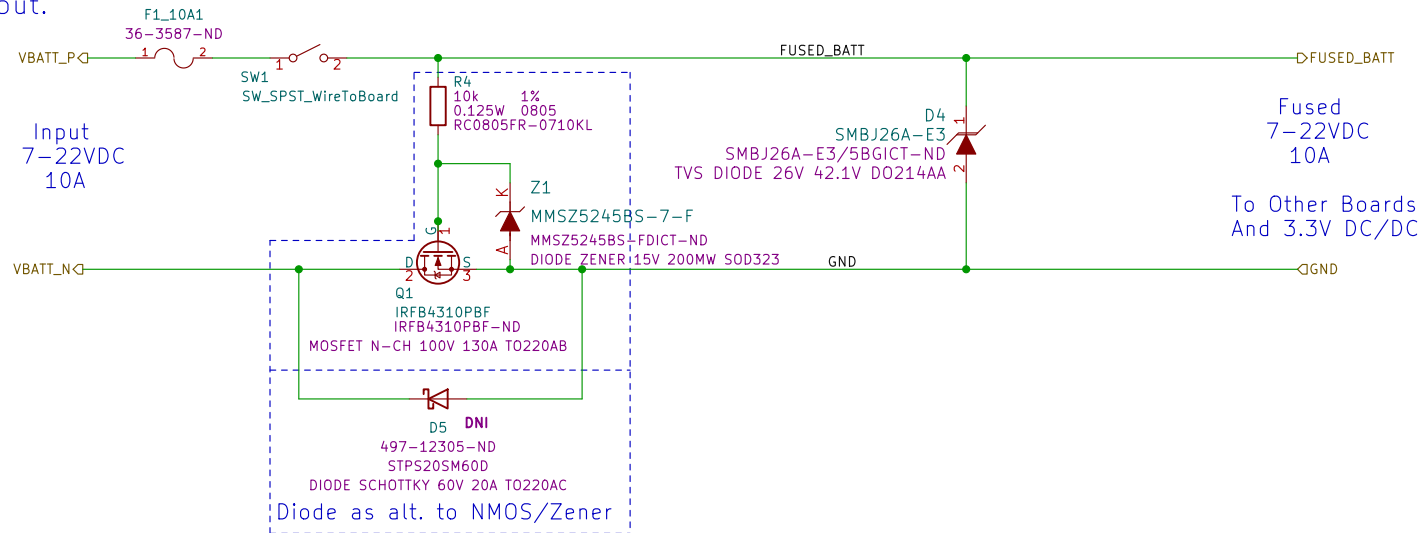
Title: Microcontroller

Size: A4 | Date: 2023-04-04
 KiCad E.D.A. kicad 6.0.4-6f826c9f35-116-ubuntu18.04.1

Rev: 0.1
 Id: 4/6

Power Conditioning

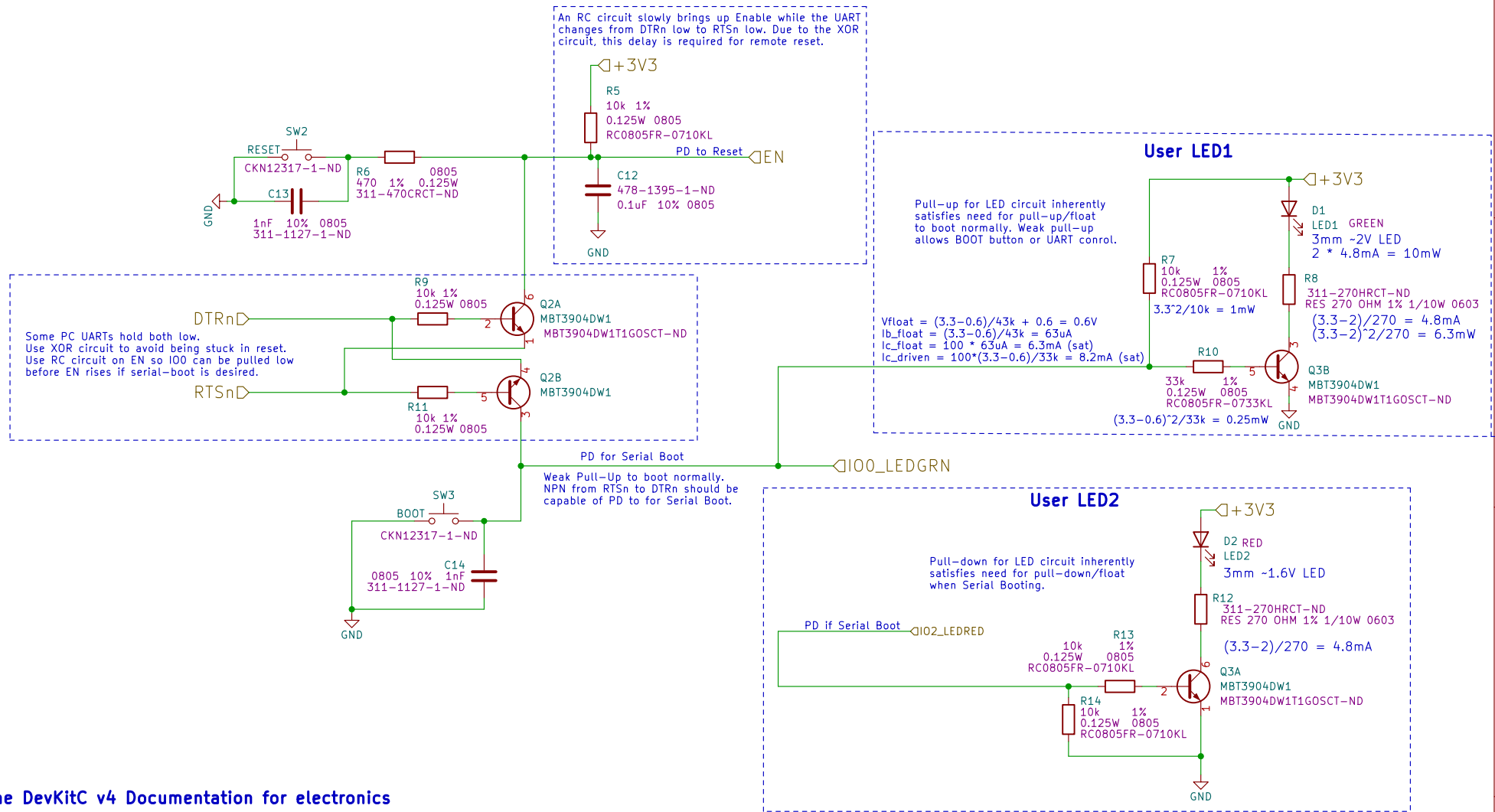
NOTE: Typical use expected to include 18V Lithium battery and 9V battery, so 7-22VDC input.



== DC/DC Selection ==
 R-783.3-0.5
 - Most expensive
 + Most efficient
 + Lowest noise & ripple
 VX(0)7803-500
 + Much cheaper
 + Wider input range
 - Less efficient
 - More noise

(c) Joshua Alden 2023	
Sheet: /Power/	
File: power.kicad_sch	
Title: Power Conditioning	
Size: A4	Date: 2023-04-04
KiCad E.D.A. kicad 6.0.4-6f826c9f35-116-ubuntu18.04.1	Rev: 0.1
	Id: 5/6

Bootstrapping of ESP32 and User LEDs



Review the DevKitC v4 Documentation for electronics

/Electronics/datasheets/ESP32 Stuff/USE THIS ONE_esp32-devkitc-v4-reference_design/ESP32-DevKitC-V4_Reference_Design/01_Schematic/ESP32-DevKitC-V4_SCH_20180607A.pdf

Things that exist onboard DevKitC-V4

- 3V3: Bypass with 22uF and 0.1uF to GND
- EN: RC to delay EN slightly
 - 10k PU resistor and 0.1uF cap to GND
 - Momentary btn to gnd w/ parallel 0.1uF to debounce
- IO0: 1:Normal Boot 0:Download New Code
 - Momentary btn to gnd w/ parallel 0.1uF to debounce
- IO0 and EN: Consult connection to DTR and RTS to allow USB/UART to automatically handle reset and boot-mode.

(c) Joshua Alden 2023

Sheet: /uC/ESP32_Bootstrapping/
File: ESP32_Bootstrapping.kicad_sch

Title: ESP32 Bootstrapping and LEDs

Size: A4 Date: 2023-04-04
KiCad E.D.A. kicad 6.0.4-6f826c9f35-116-ubuntu18.04.1

Rev: 0.1
Id: 6/6