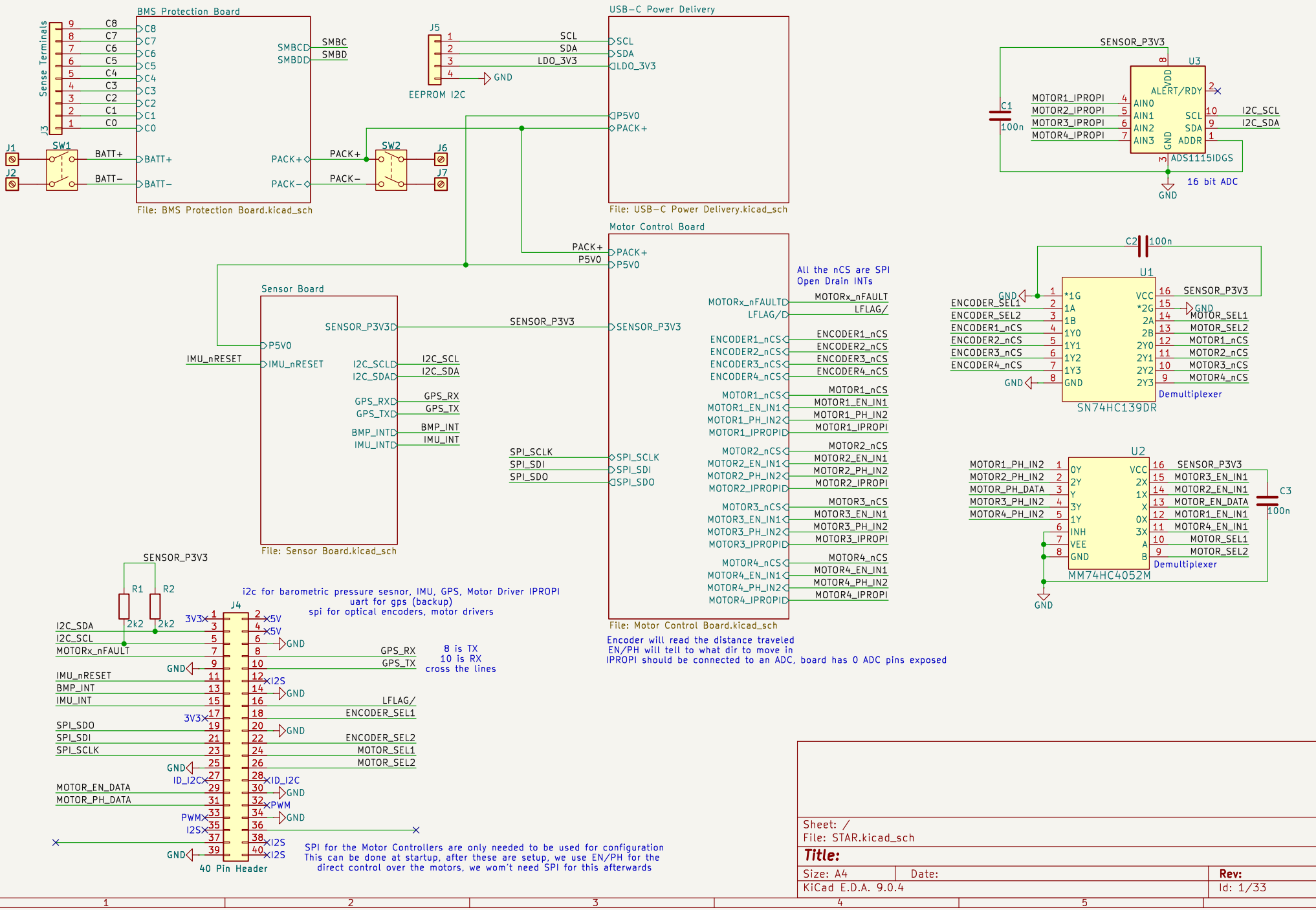
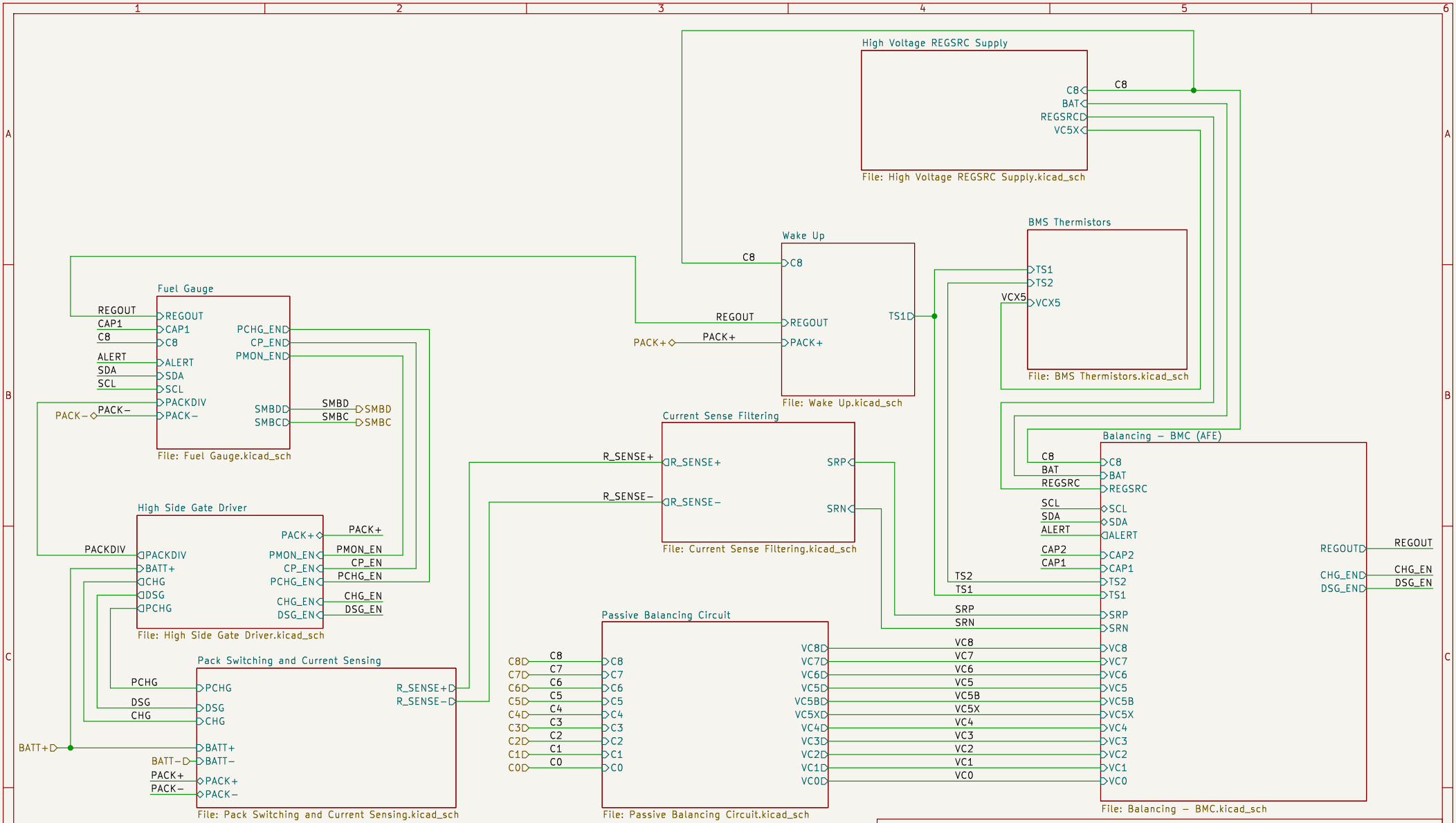

BMS Protection Board
Sense Terminals
J3
C8
C7
C6
C5
C4
C3
C2
C1
C0
SMBC
SMBD
J1
J2
SW1
BATT+
BATT−
PACK+
PACK−
SW2
J6
J7
File: BMS Protection Board.kicad_sch
J5
EEPROM I2C
SCL
SDA
LDO_3V3
GND
USB−C Power Delivery
P5V0
PACK+
File: USB−C Power Delivery.kicad_sch
SENSOR_P3V3
U3
VDD
ALERT/RDY
AIN0
AIN1
AIN2
AIN3
GND
SCL
SDA
ADDR
ADS1115IDGS
C1
100n
MOTOR1_IPROPI
MOTOR2_IPROPI
MOTOR3_IPROPI
MOTOR4_IPROPI
I2C_SCL
I2C_SDA
16 bit ADC
Motor Control Board
All the nCS are SPI
Open Drain INTs
MOTORx_nFAULT
LFLAG/
ENCODER1_nCS
ENCODER2_nCS
ENCODER3_nCS
ENCODER4_nCS
MOTOR1_nCS
MOTOR1_EN_IN1
MOTOR1_PH_IN2
MOTOR1_IPROPI
MOTOR2_nCS
MOTOR2_EN_IN1
MOTOR2_PH_IN2
MOTOR2_IPROPI
MOTOR3_nCS
MOTOR3_EN_IN1
MOTOR3_PH_IN2
MOTOR3_IPROPI
MOTOR4_nCS
MOTOR4_EN_IN1
MOTOR4_PH_IN2
MOTOR4_IPROPI
SPI_SCLK
SPI_SDI
SPI_SDO
File: Motor Control Board.kicad_sch
Encoder will read the distance traveled
EN/PH will tell to what dir to move in
IPROPI should be connected to an ADC, board has 0 ADC pins exposed
Sensor Board
IMU_nRESET
GPS_RX
GPS_TX
BMP_INT
IMU_INT
File: Sensor Board.kicad_sch
C2
100n
U1
*1G
*2G
1A
1B
1Y0
1Y1
1Y2
1Y3
2A
2B
2Y0
2Y1
2Y2
2Y3
ENCODER_SEL1
ENCODER_SEL2
ENCODER1_nCS
ENCODER2_nCS
ENCODER3_nCS
ENCODER4_nCS
MOTOR_SEL1
MOTOR_SEL2
MOTOR1_nCS
MOTOR2_nCS
MOTOR3_nCS
MOTOR4_nCS
SN74HC139DR
Demultiplexer
U2
0Y
2Y
Y
3Y
1Y
INH
VEE
0X
2X
1X
X
3X
A
B
MOTOR1_PH_IN2
MOTOR2_PH_IN2
MOTOR_PH_DATA
MOTOR3_PH_IN2
MOTOR4_PH_IN2
MOTOR3_EN_IN1
MOTOR2_EN_IN1
MOTOR_EN_DATA
MOTOR1_EN_IN1
MOTOR4_EN_IN1
MOTOR_SEL1
MOTOR_SEL2
MM74HC4052M
C3
100n
SENSOR_P3V3
R1
R2
2k2
2k2
J4
i2c for barometric pressure sesnor, IMU, GPS, Motor Driver IPROPI
uart for gps (backup)
spi for optical encoders, motor drivers
8 is TX
10 is RX
cross the lines
3V3
5V
GND
I2S
ID_I2C
PWM
LFLAG/
ENCODER_SEL1
ENCODER_SEL2
MOTOR_SEL1
MOTOR_SEL2
40 Pin Header
SPI for the Motor Controllers are only needed to be used for configuration
This can be done at startup, after these are setup, we use EN/PH for the
direct control over the motors, we wom't need SPI for this afterwards
Sheet: /
File: STAR.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 9.0.4
Id: 1/33



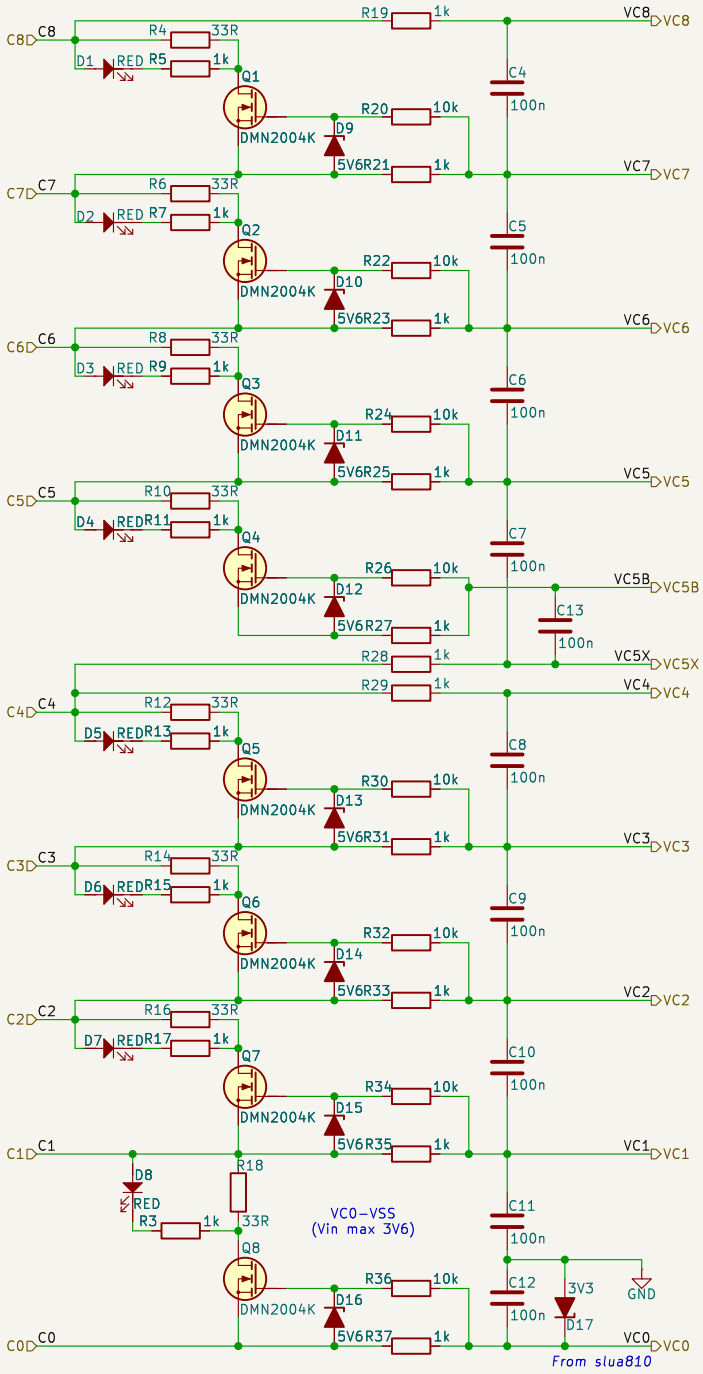
BMS Protection Board for 8S Battery Pack
In/Out: min voltage: 25.6V, nominal voltage: 29.6V, max voltage: 33.6V
Output: 50A continuous discharge current
Min Each Cell set to 3.2V, Max Each Cell set to 4.2V

Sheet: /BMS Protection Board/
File: BMS Protection Board.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 2/33

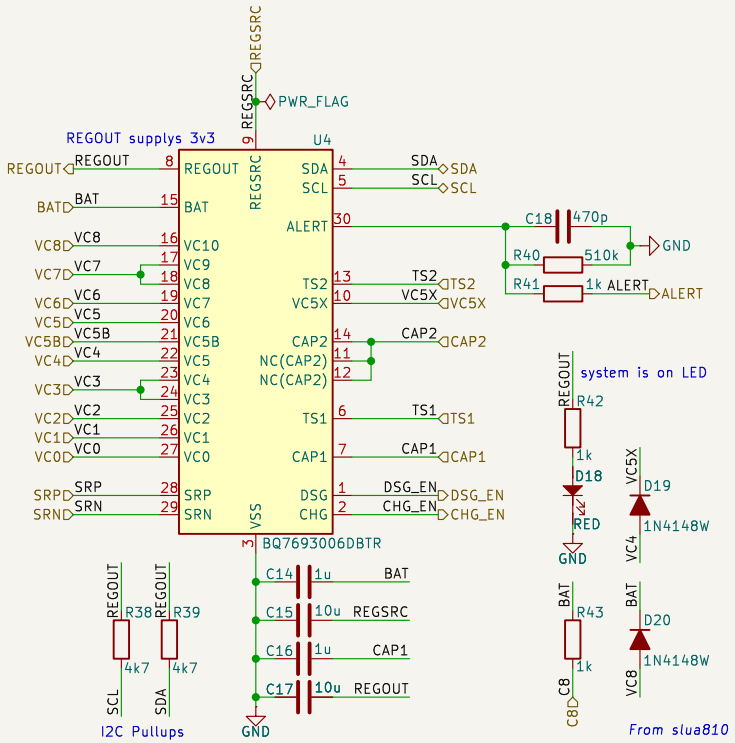
Passive Balancing Circuit



Sheet: /BMS Protection Board/Passive Balancing Circuit/
File: Passive Balancing Circuit.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 3/33



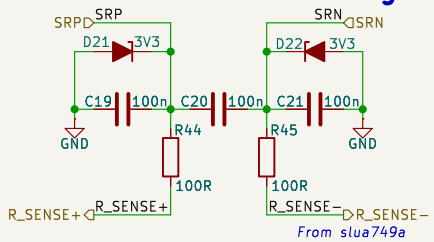
Balancing & BMC (AFE)

Sheet: /BMS Protection Board/Balancing – BMC (AFE)/
File: Balancing – BMC.kicad_sch

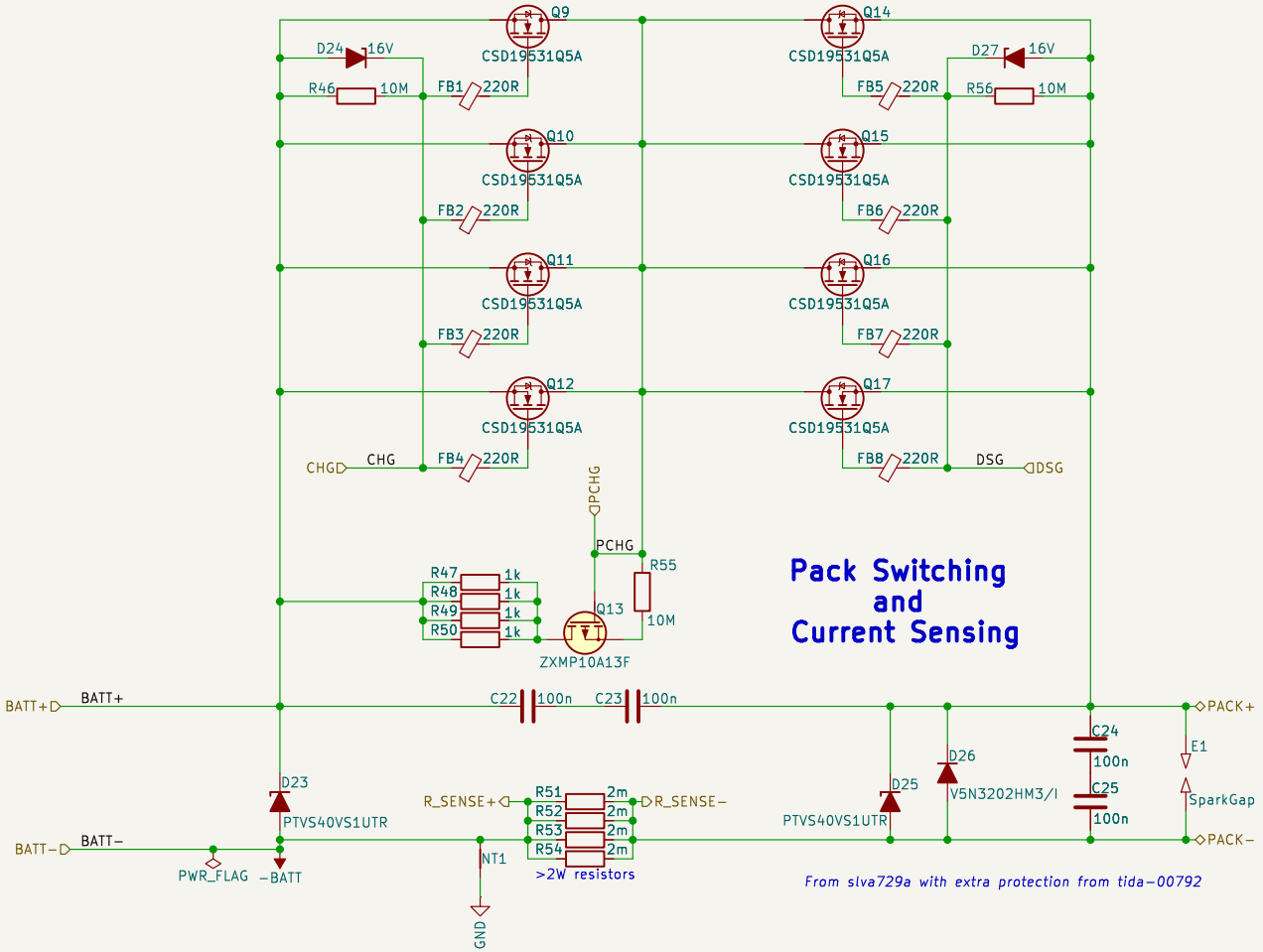
Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 4/33

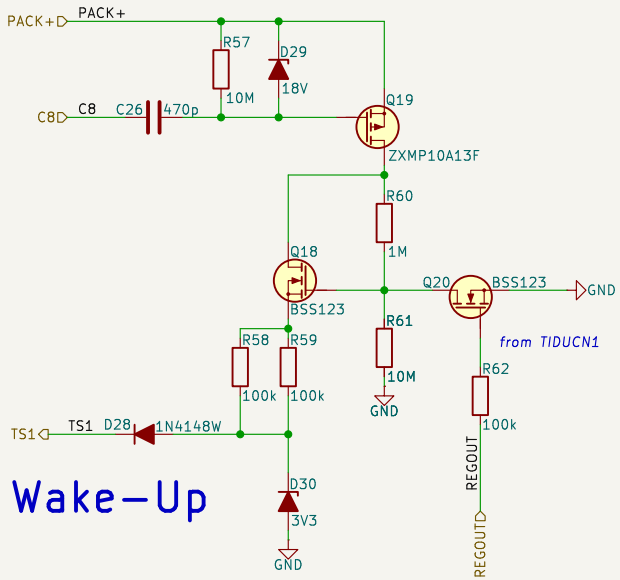
Current Sense Filtering



Sheet: /BMS Protection Board/Current Sense Filtering/		
File: Current Sense Filtering.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 5/33

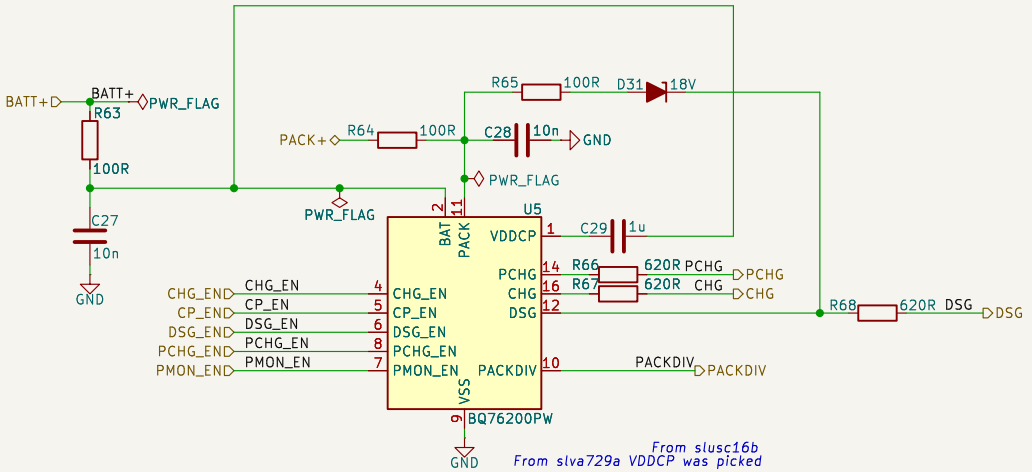


Sheet: /BMS Protection Board/Pack Switching and Current Sensing/		
File: Pack Switching and Current Sensing.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 6/33



Sheet: /BMS Protection Board/Wake Up/		
File: Wake Up.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 7/33

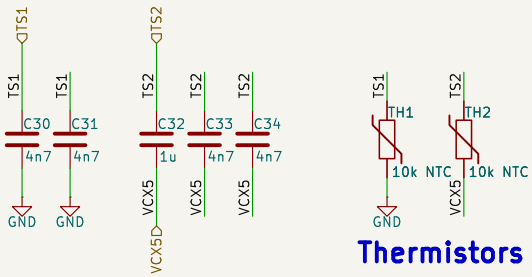
High-Side Gate Driver



Sheet: /BMS Protection Board/High Side Gate Driver/
File: High Side Gate Driver.kicad_sch

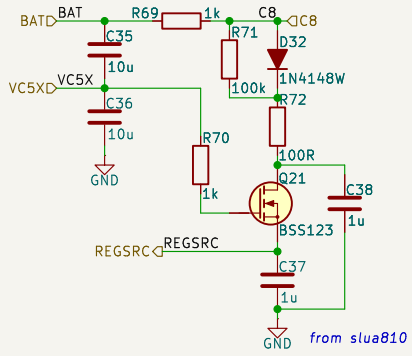
Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 8/33



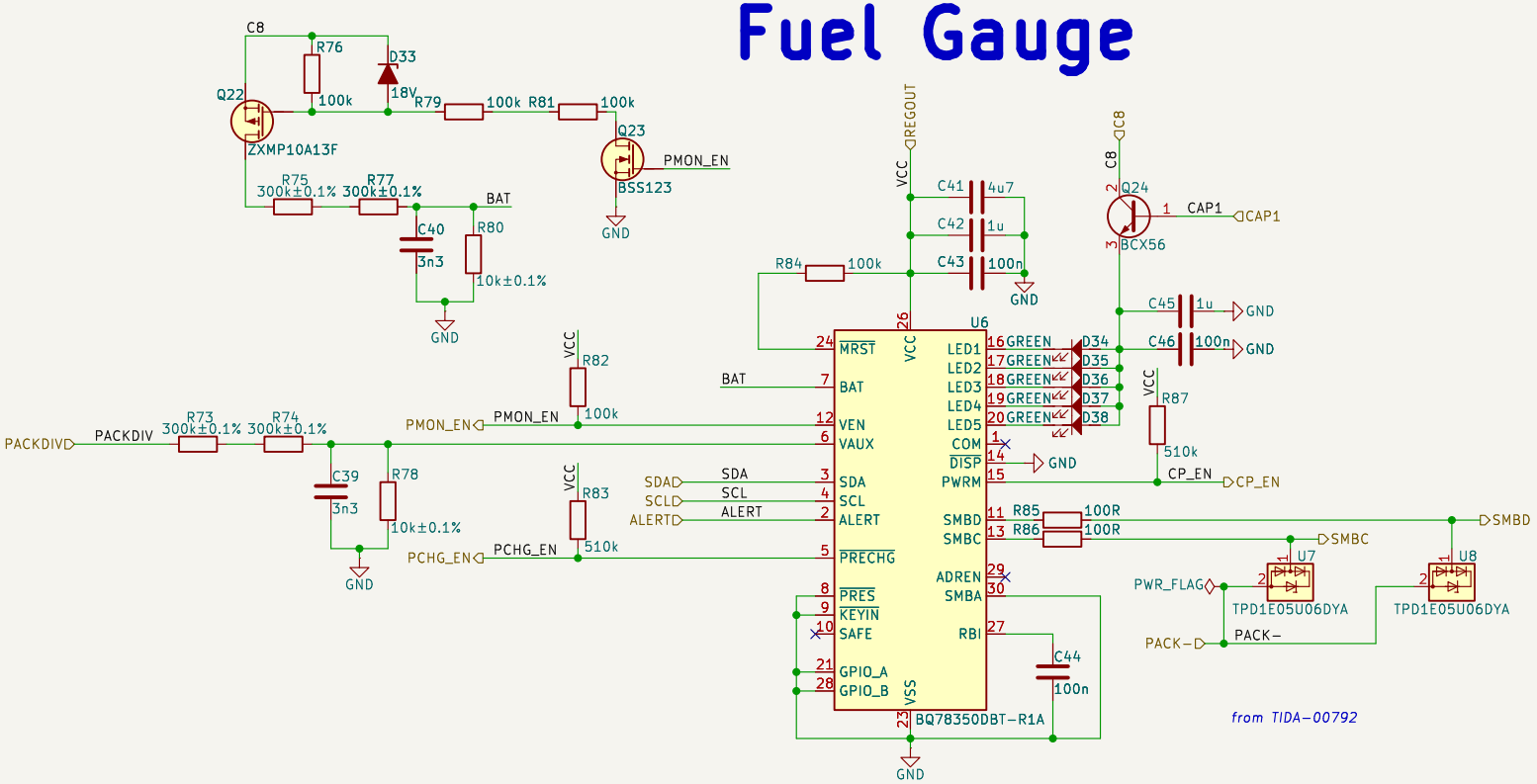
Sheet: /BMS Protection Board/BMS Thermistors/		
File: BMS Thermistors.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 9/33

High Voltage REGSRC Supply



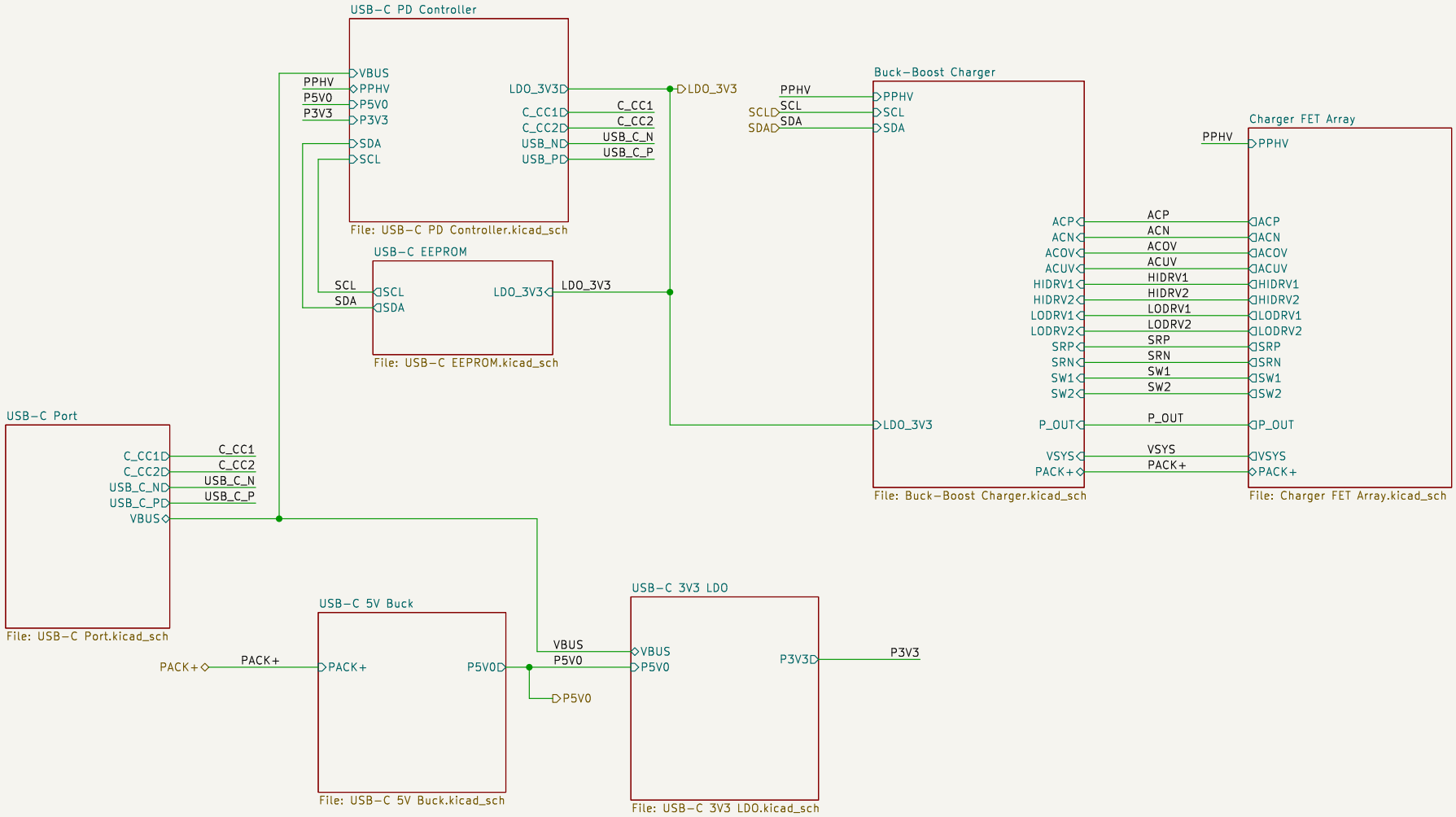
Sheet: /BMS Protection Board/High Voltage REGSRC Supply/		
File: High Voltage REGSRC Supply.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 10/33

Fuel Gauge



Flash this with TI's software, will need to connec to computer with SMB to USB adapter, could use the expensive one from TI, but $$$

Sheet: /BMS Protection Board/Fuel Gauge/		
File: Fuel Gauge.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 11/33



Design is based off of PMP41062

The Correct Flashing Process
Here is the correct workflow:

Generate Your Configuration: Use the TPS2575x Application Customization Tool from Texas Instruments to create the binary file (.bin) that contains all your desired settings.

Program the EEPROM Separately: Connect the CAT24C512 EEPROM to your dedicated flasher or an Arduino. Do this before connecting it to the TPS25751.

Connect power (VCC), ground (GND), SDA, and SCL.

Use the flasher's software to write the .bin file you generated onto the EEPROM.

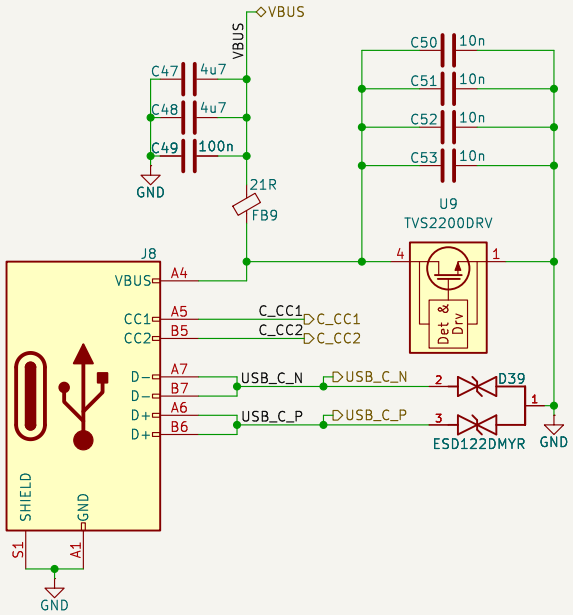
Assemble the Final Circuit: Once the EEPROM is successfully flashed, connect it to the I2Cc (the controller port) of the TPS25751.

Now, when you power on your circuit, the TPS25751 will use its I2Cc port to automatically read the settings you loaded onto the EEPROM and will configure itself correctly.

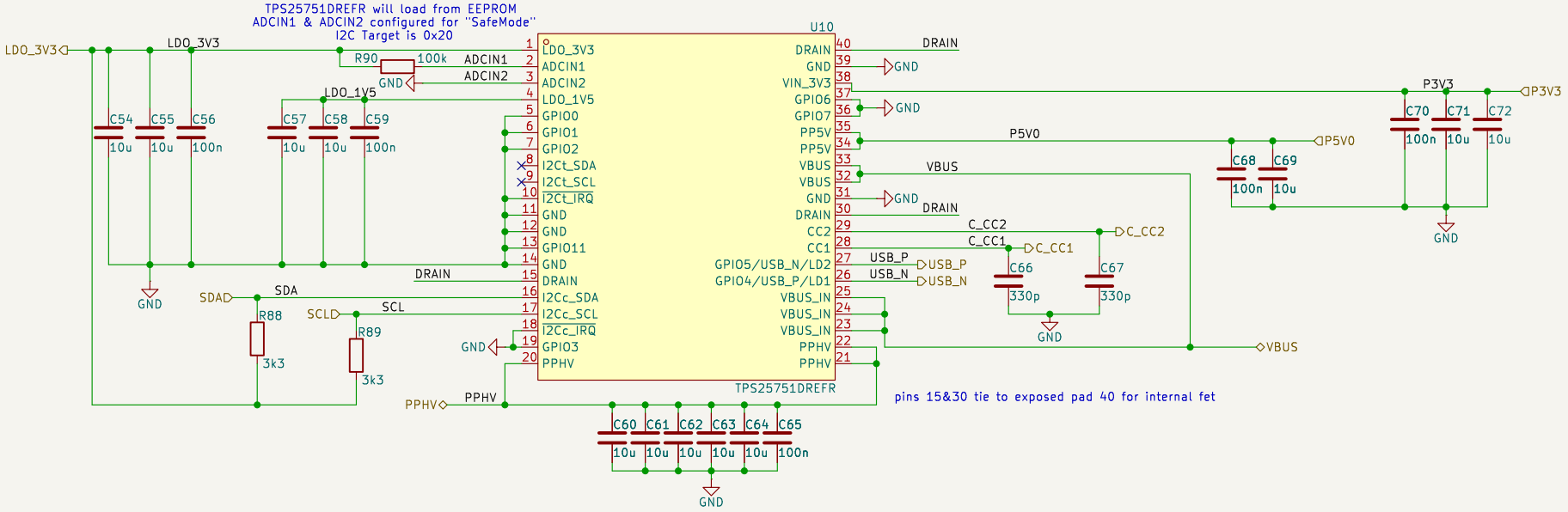
Sheet: /USB-C Power Delivery/
File: USB-C Power Delivery.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 12/33



Sheet: /USB-C Power Delivery/USB-C Port/		
File: USB-C Port.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 13/33



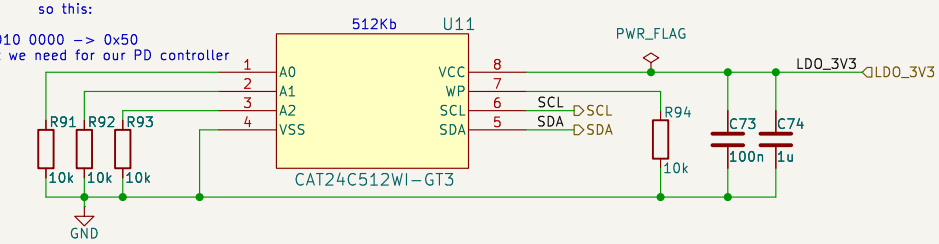
Sheet: /USB-C Power Delivery/USB-C PD Controller/		
File: USB-C PD Controller.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 14/33

1010 ????
GND=0

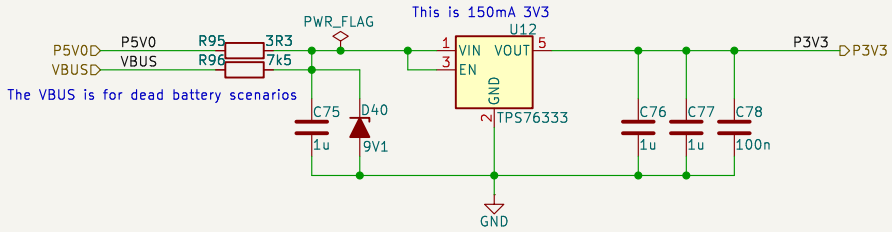
so this:

1010 0000 -> 0x50
which is what we need for our PD controller

Flash this with the bin taken from TI's website. Use I2C to USB adapter.
Flash this when the system is off (not powered), only flash/power this.



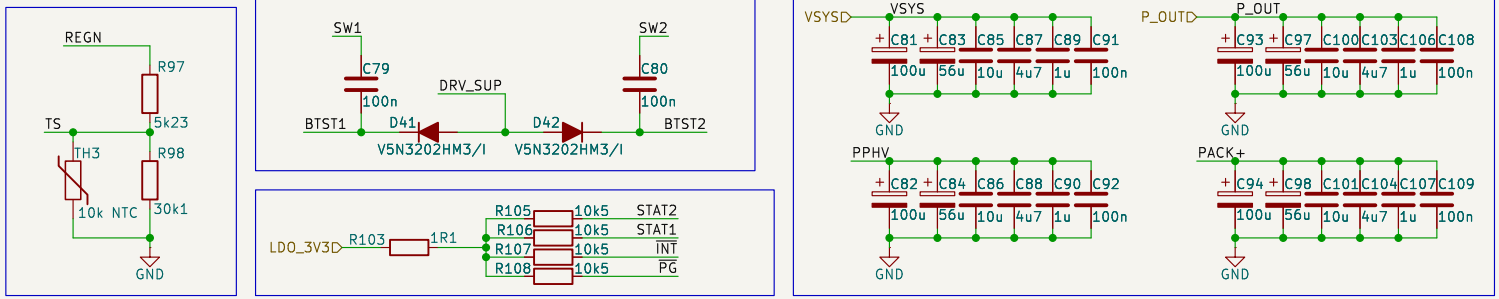
Sheet: /USB-C Power Delivery/USB-C EEPROM/		
File: USB-C EEPROM.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 15/33

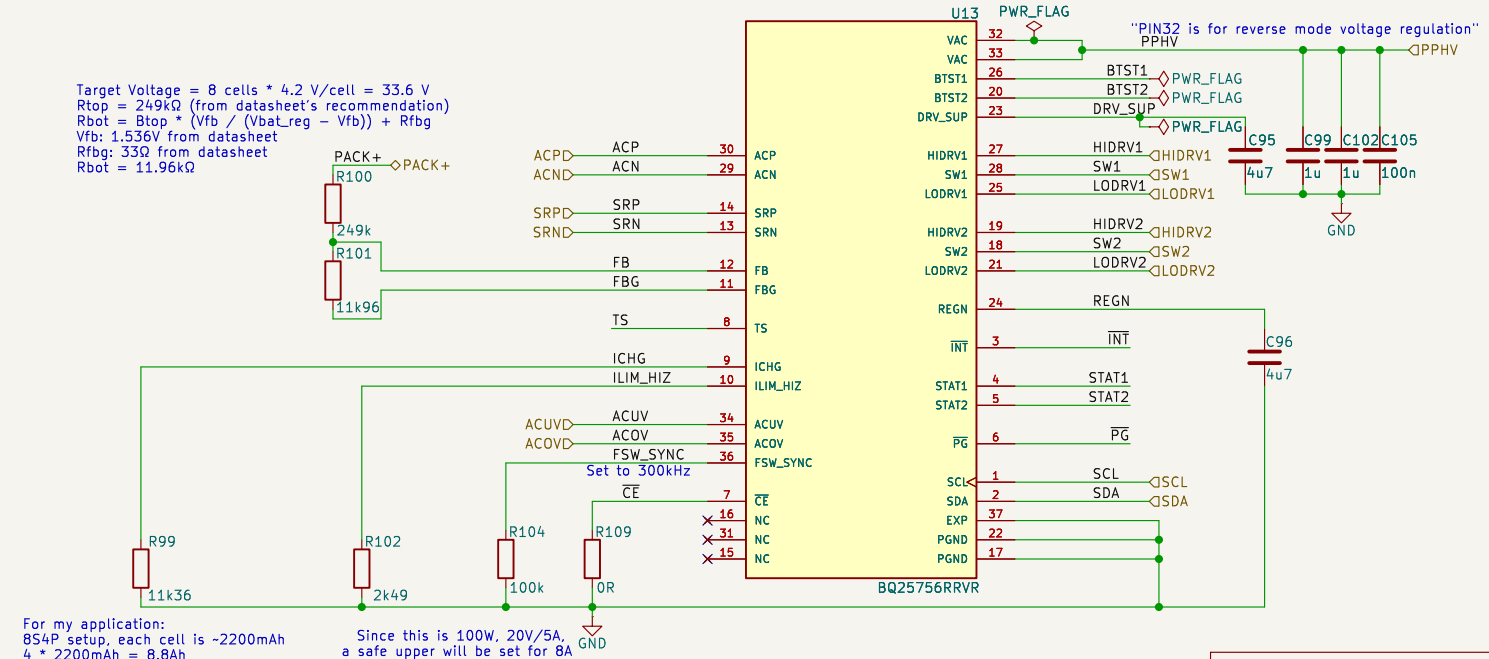


Sheet: /USB-C Power Delivery/USB-C 3V3 LDO/
File: USB-C 3V3 LDO.kicad_sch

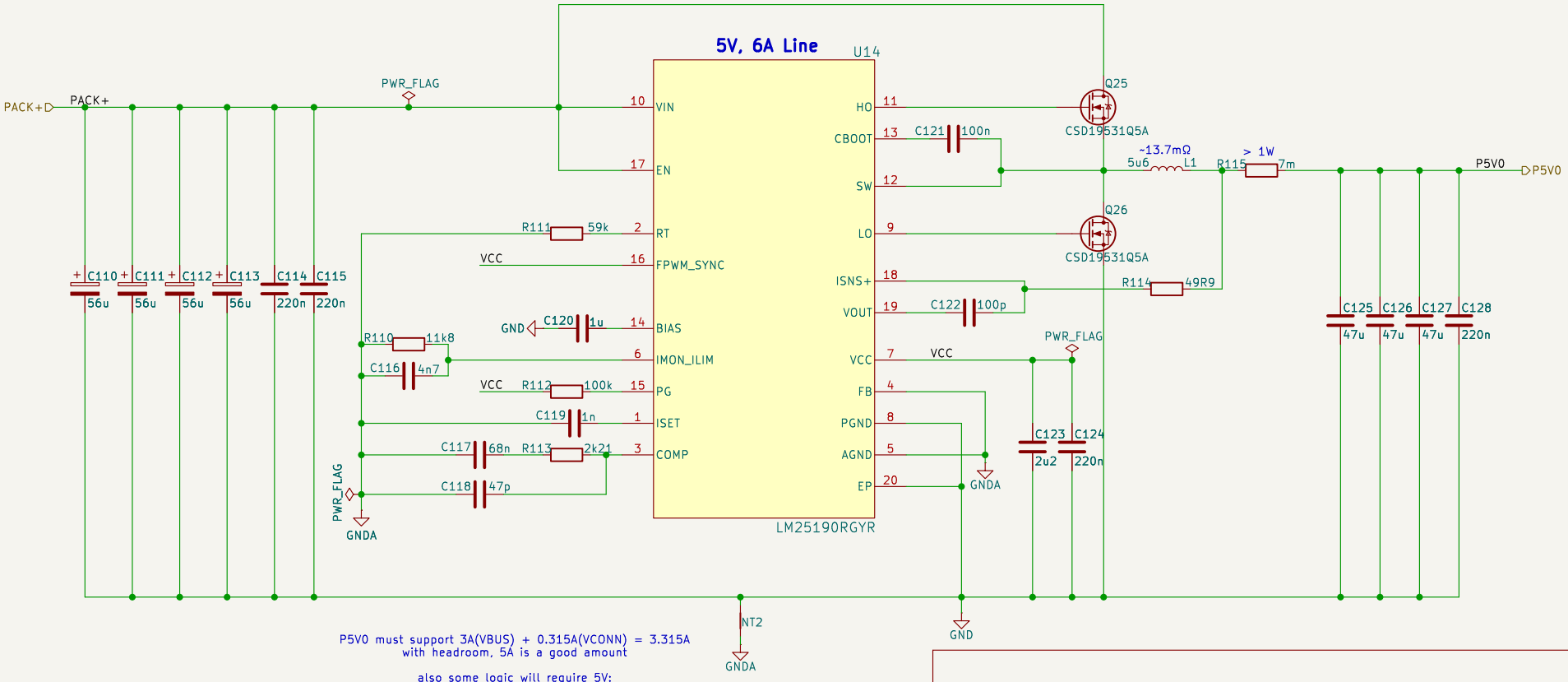
Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 16/33





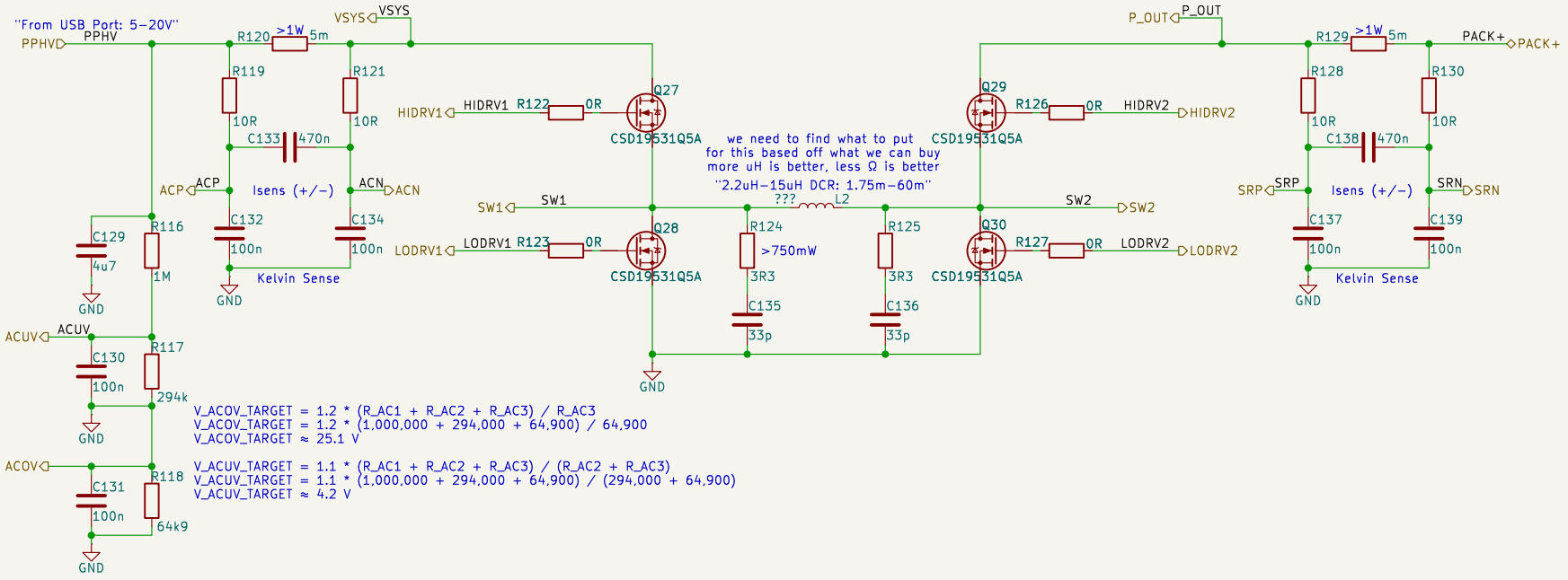
Sheet: /USB-C Power Delivery/Buck-Boost Charger/		
File: Buck-Boost Charger.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 17/33



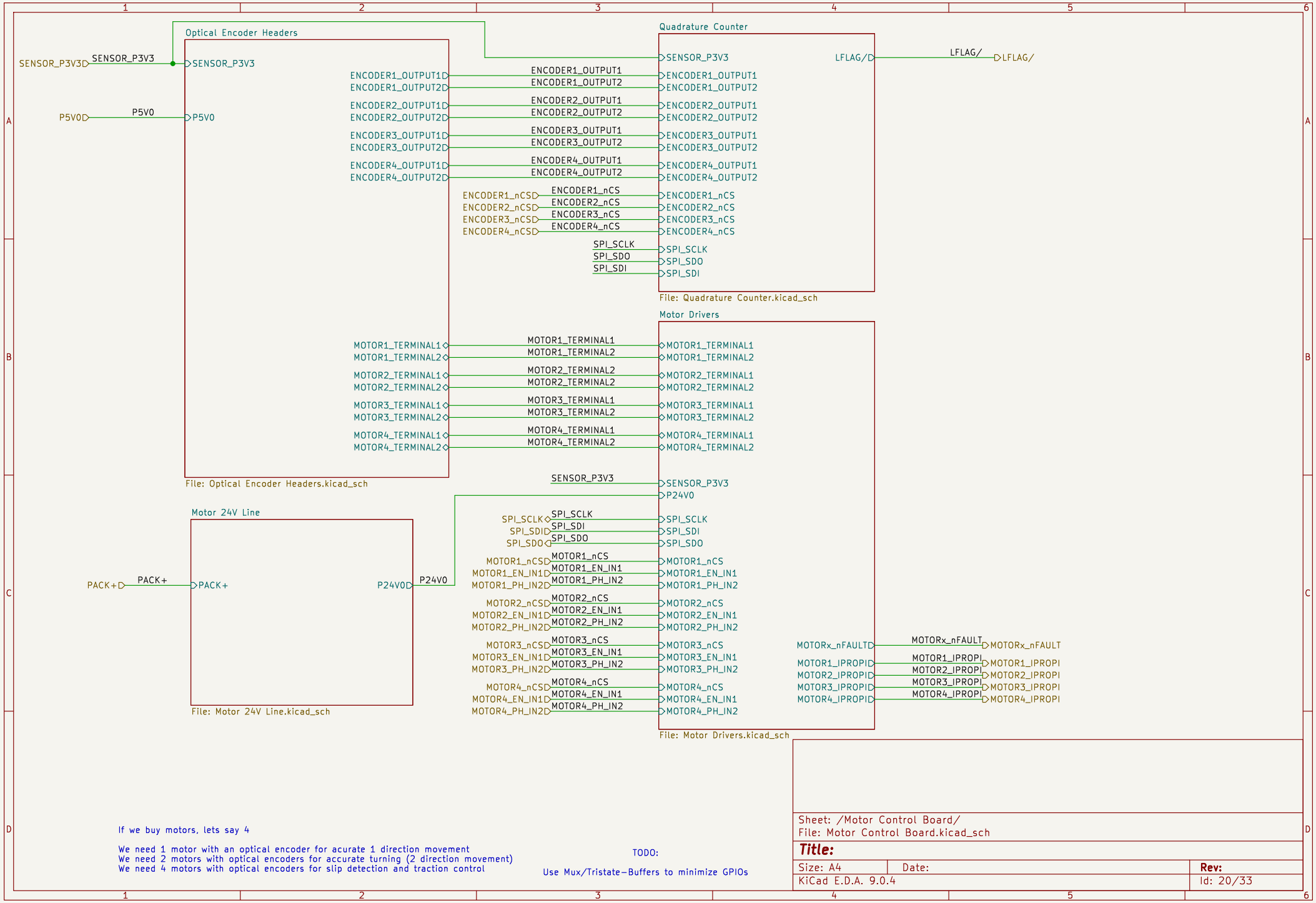
Sheet: /USB-C Power Delivery/USB-C 5V Buck/
File: USB-C 5V Buck.kicad_sch

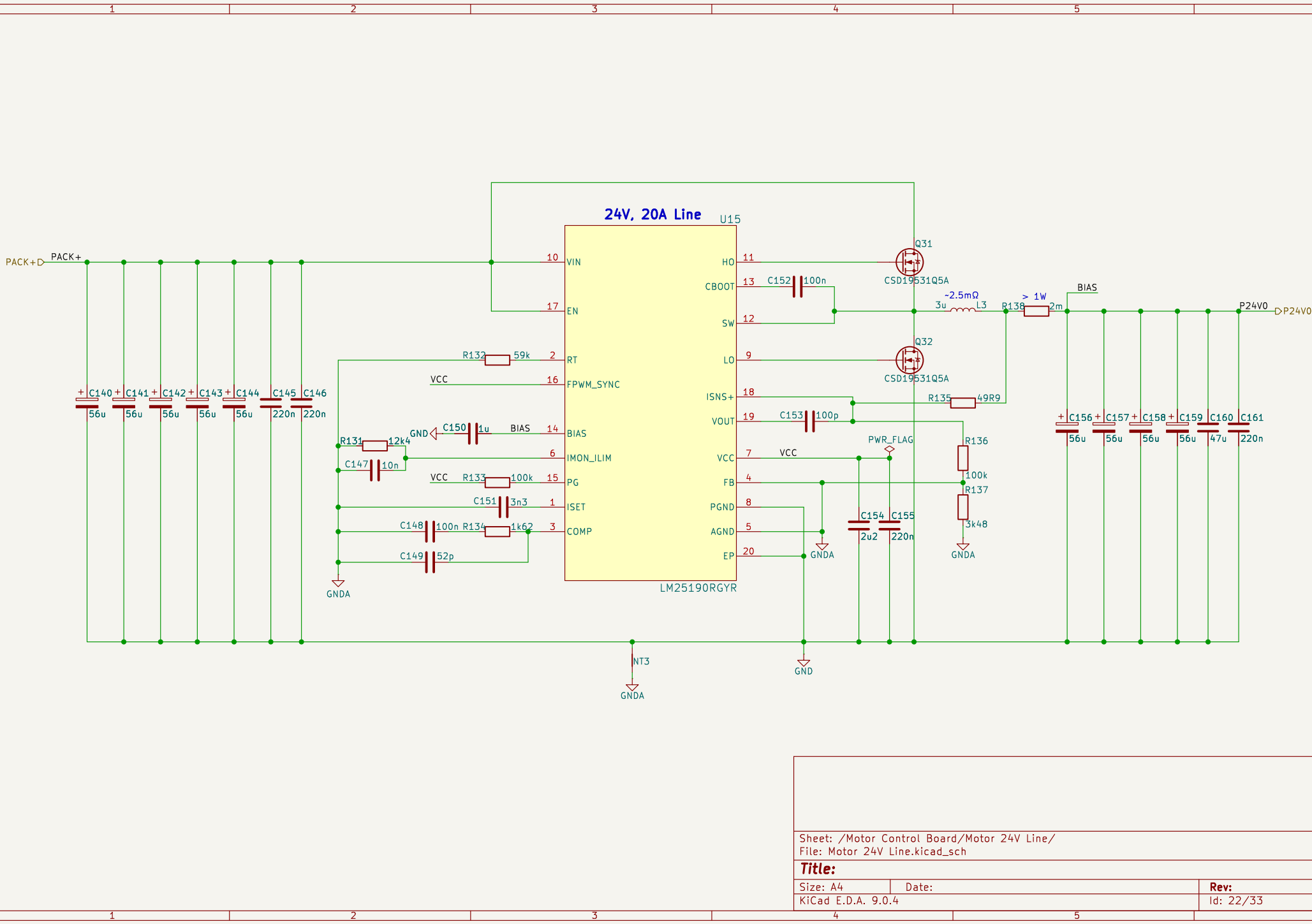
Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 18/33



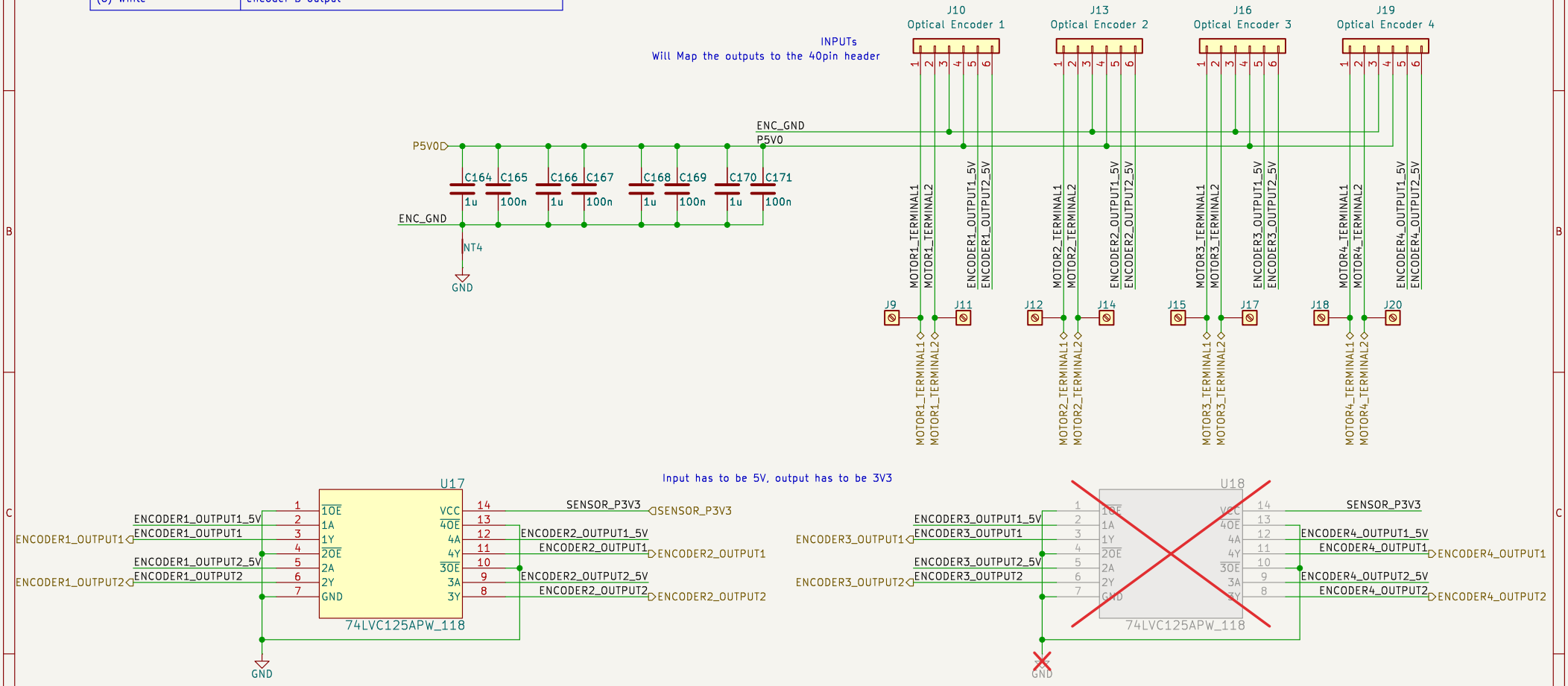
Sheet: /USB-C Power Delivery/Charger FET Array/		
File: Charger FET Array.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 19/33


Optical Encoder Headers
SENSOR_P3V3
P5V0
ENCODER1_OUTPUT1
ENCODER1_OUTPUT2
ENCODER2_OUTPUT1
ENCODER2_OUTPUT2
ENCODER3_OUTPUT1
ENCODER3_OUTPUT2
ENCODER4_OUTPUT1
ENCODER4_OUTPUT2
ENCODER1_nCS
ENCODER2_nCS
ENCODER3_nCS
ENCODER4_nCS
SPI_SCLK
SPI_SDO
SPI_SDI
MOTOR1_TERMINAL1
MOTOR1_TERMINAL2
MOTOR2_TERMINAL1
MOTOR2_TERMINAL2
MOTOR3_TERMINAL1
MOTOR3_TERMINAL2
MOTOR4_TERMINAL1
MOTOR4_TERMINAL2
File: Optical Encoder Headers.kicad_sch
Quadrature Counter
LFLAG/
File: Quadrature Counter.kicad_sch
Motor Drivers
P24V0
MOTOR1_nCS
MOTOR1_EN_IN1
MOTOR1_PH_IN2
MOTOR2_nCS
MOTOR2_EN_IN1
MOTOR2_PH_IN2
MOTOR3_nCS
MOTOR3_EN_IN1
MOTOR3_PH_IN2
MOTOR4_nCS
MOTOR4_EN_IN1
MOTOR4_PH_IN2
MOTORx_nFAULT
MOTOR1_IPROPI
MOTOR2_IPROPI
MOTOR3_IPROPI
MOTOR4_IPROPI
File: Motor Drivers.kicad_sch
Motor 24V Line
PACK+
File: Motor 24V Line.kicad_sch
If we buy motors, lets say 4
We need 1 motor with an optical encoder for acurate 1 direction movement
We need 2 motors with optical encoders for accurate turning (2 direction movement)
We need 4 motors with optical encoders for slip detection and traction control
TODO:
Use Mux/Tristate-Buffers to minimize GPIOs
Sheet: /Motor Control Board/
File: Motor Control Board.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 9.0.4
Id: 20/33



Sheet: /Motor Control Board/Motor 24V Line/		
File: Motor 24V Line.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 22/33

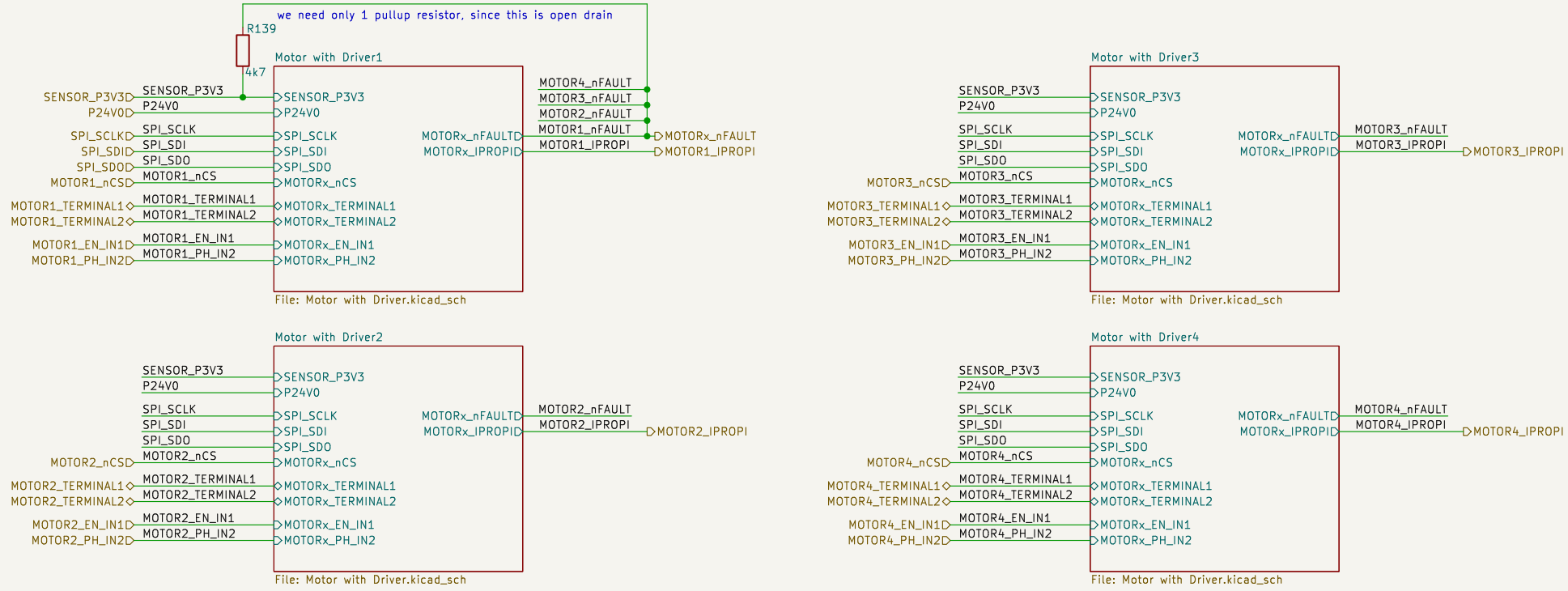
(1) Red	motor power (connects to one motor terminal)
(2) Black	motor power (connects to the other motor terminal)
(3) Green	encoder GND
(4) Blue	encoder Vcc (3.5 – 20 V)
(5) Yellow	encoder A output
(6) White	encoder B output



Sheet: /Motor Control Board/Optical Encoder Headers/
File: Optical Encoder Headers.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 24/33



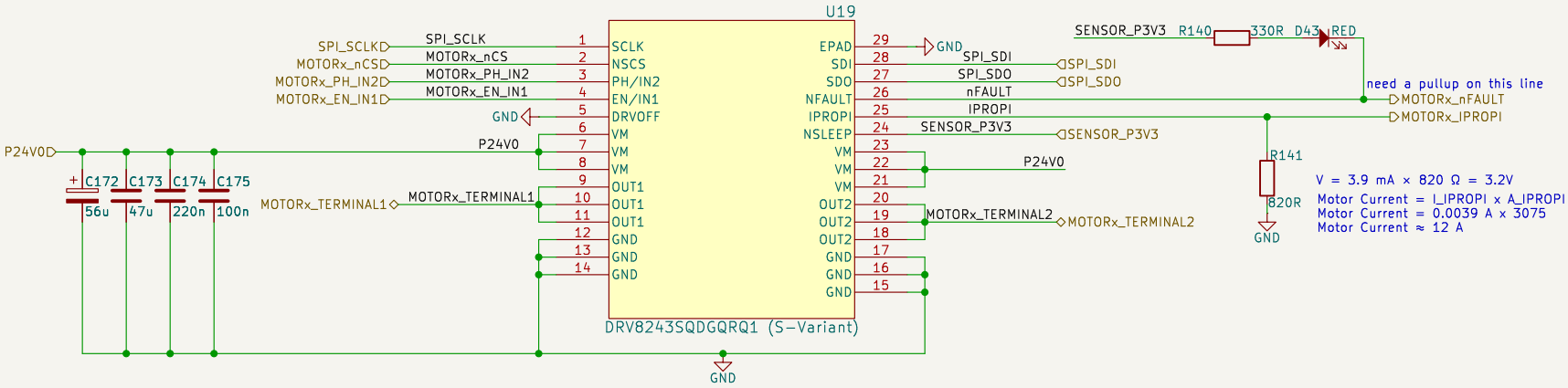
Sheet: /Motor Control Board/Motor Drivers/
File: Motor Drivers.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 25/33

For the SPI (P) variant, you can set the SCLK frequency up to 10 MHz.

For the SPI (S) variant, the maximum is 10 MHz, but you should reduce it to 8 MHz if you have a 20 pF load on the SDO line.



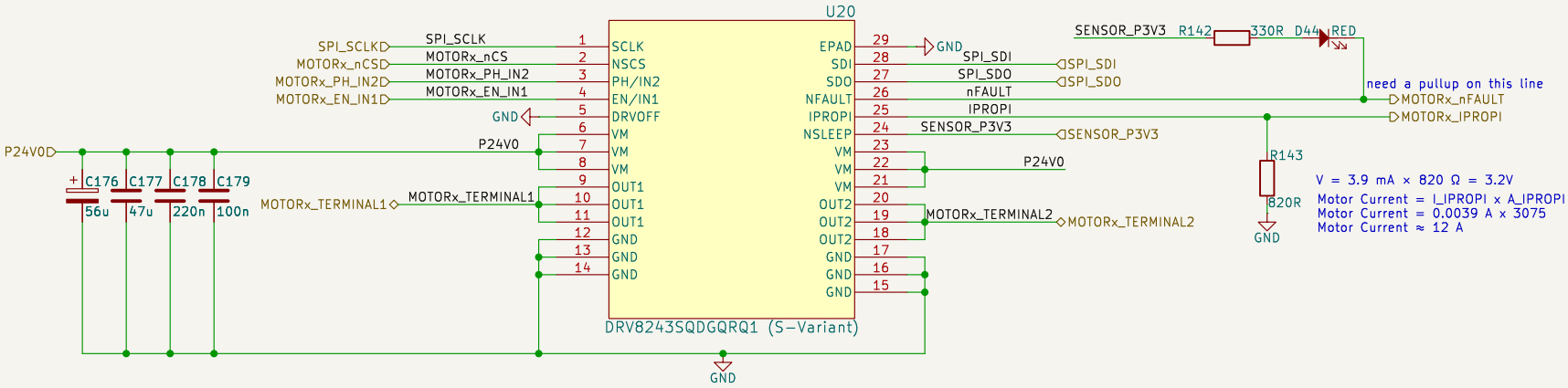
Sheet: /Motor Control Board/Motor Drivers/Motor with Driver1/
File: Motor with Driver.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 26/33

For the SPI (P) variant, you can set the SCLK frequency up to 10 MHz.

For the SPI (S) variant, the maximum is 10 MHz, but you should reduce it to 8 MHz if you have a 20 pF load on the SDO line.



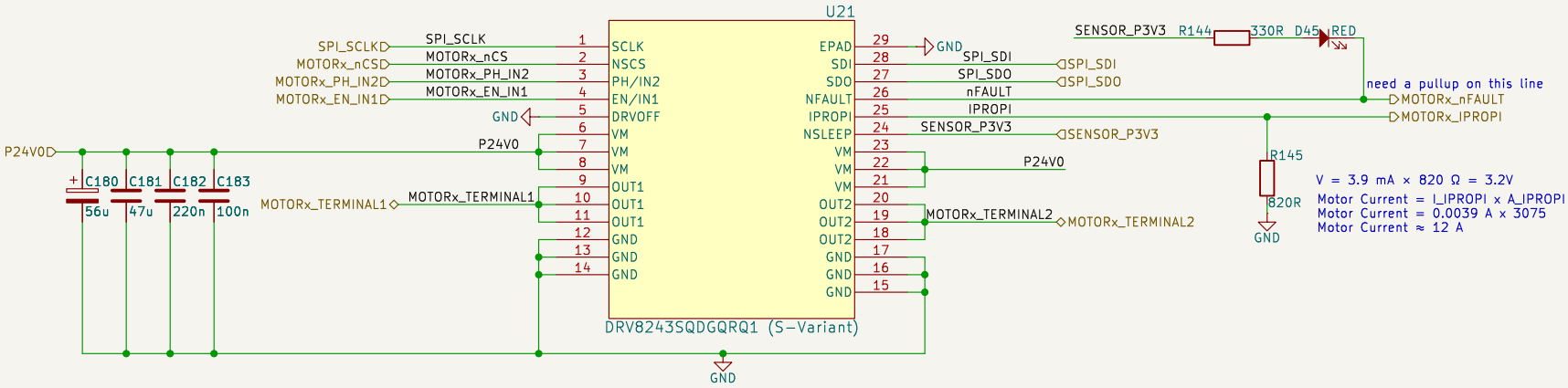
Sheet: /Motor Control Board/Motor Drivers/Motor with Driver2/
File: Motor with Driver.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 27/33

For the SPI (P) variant, you can set the SCLK frequency up to 10 MHz.

For the SPI (S) variant, the maximum is 10 MHz, but you should reduce it to 8 MHz if you have a 20 pF load on the SDO line.



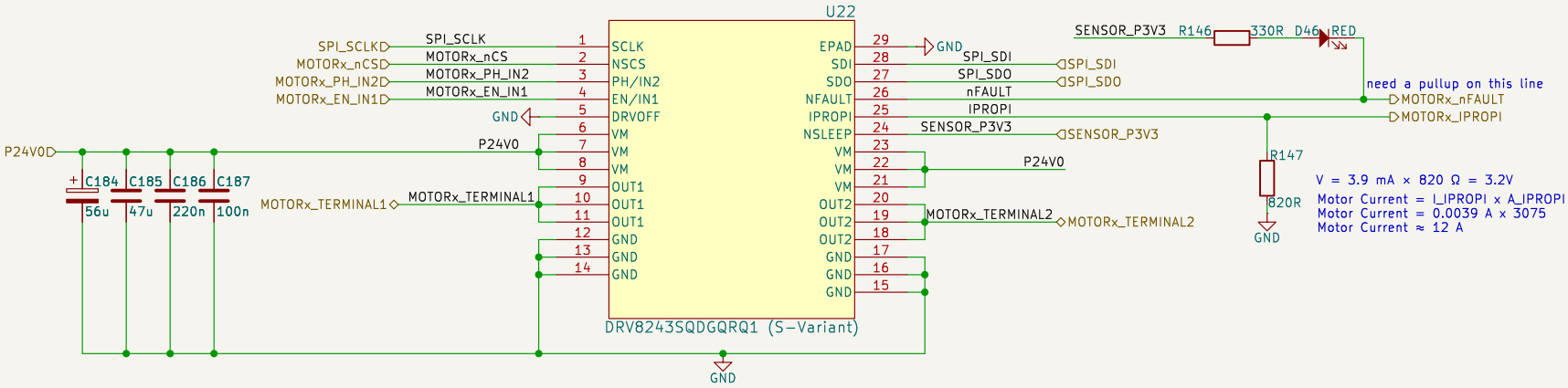
Sheet: /Motor Control Board/Motor Drivers/Motor with Driver3/
File: Motor with Driver.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 28/33

For the SPI (P) variant, you can set the SCLK frequency up to 10 MHz.

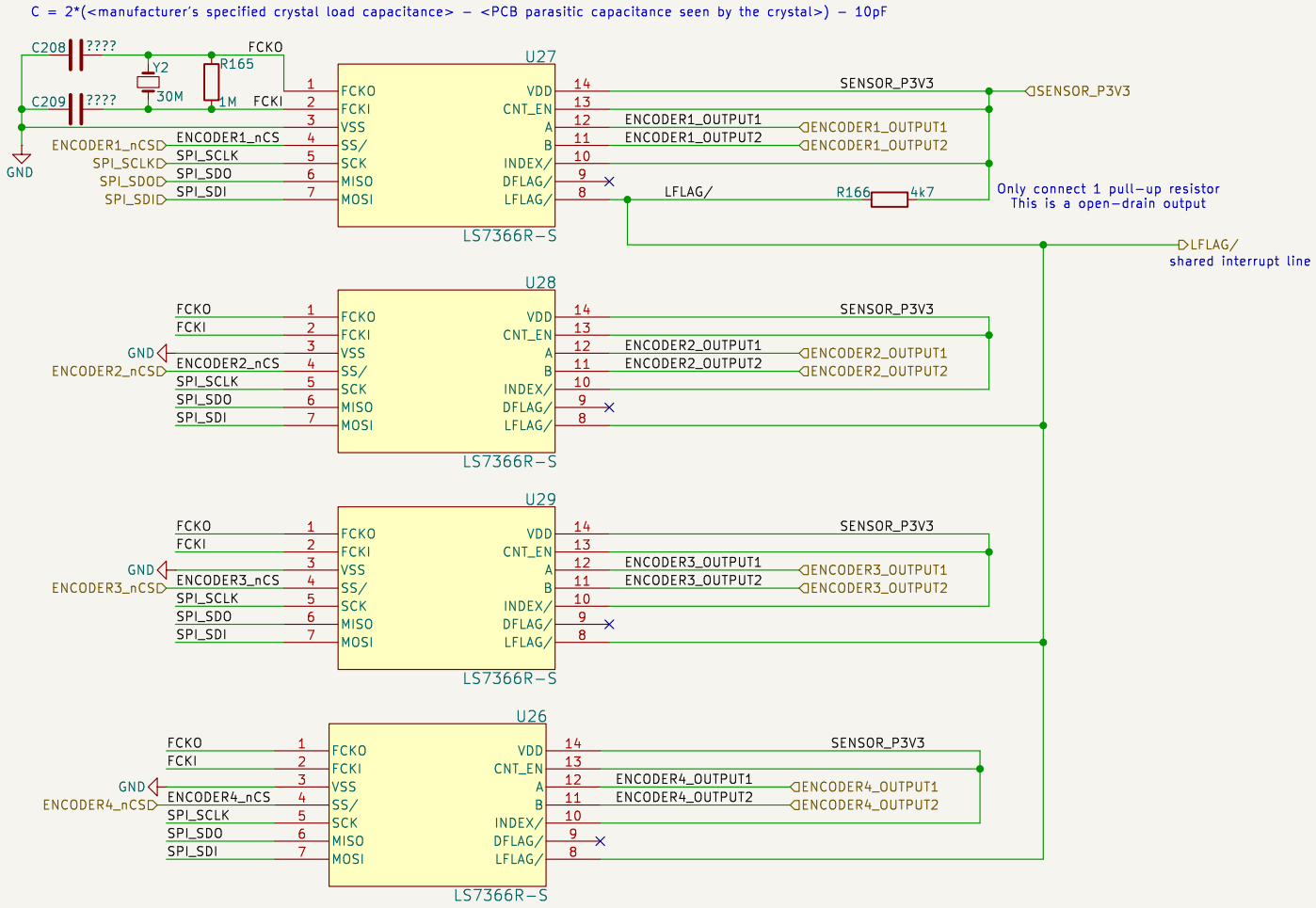
For the SPI (S) variant, the maximum is 10 MHz, but you should reduce it to 8 MHz if you have a 20 pF load on the SDO line.



Sheet: /Motor Control Board/Motor Drivers/Motor with Driver4/
File: Motor with Driver.kicad_sch

Title:

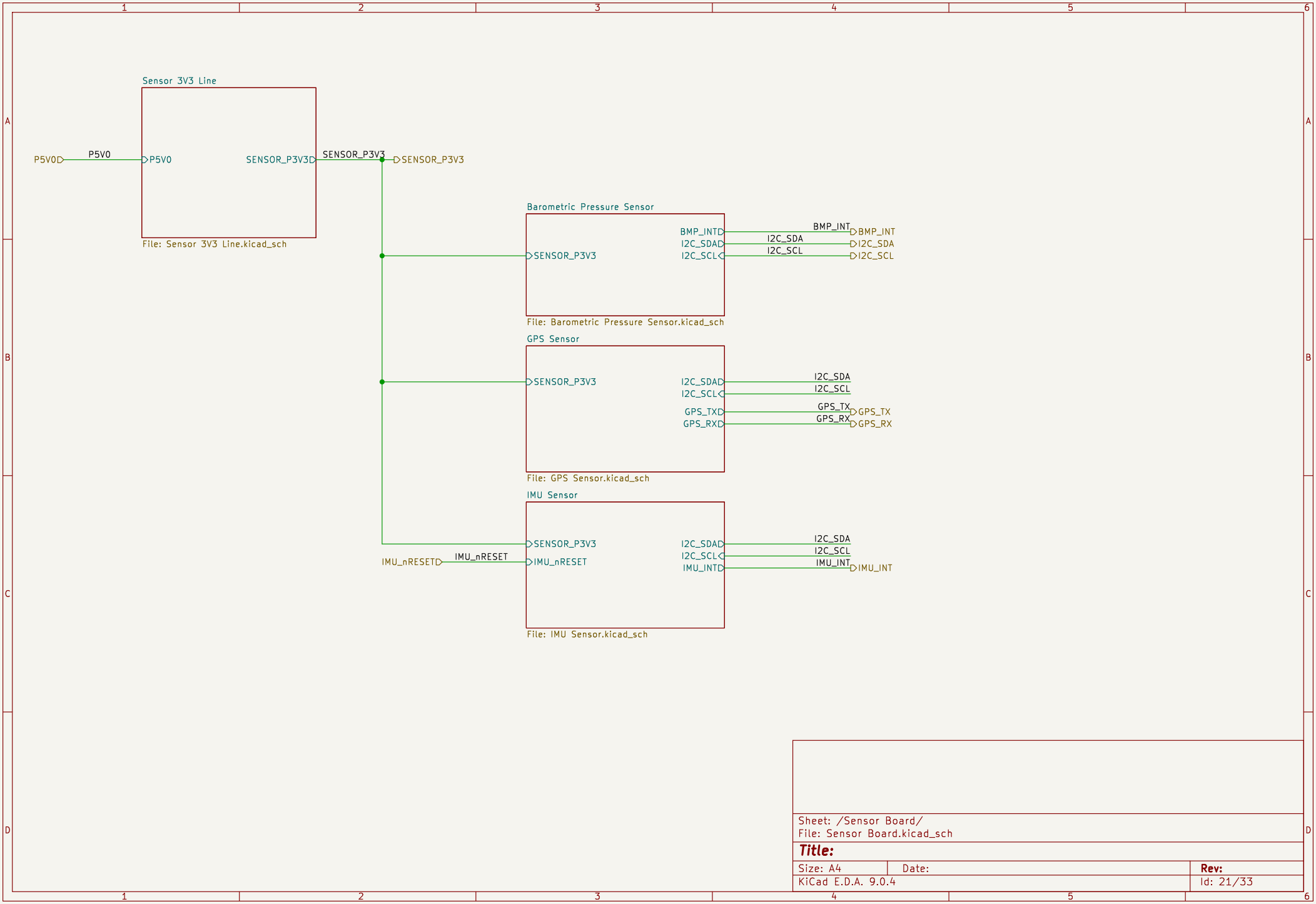
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 29/33

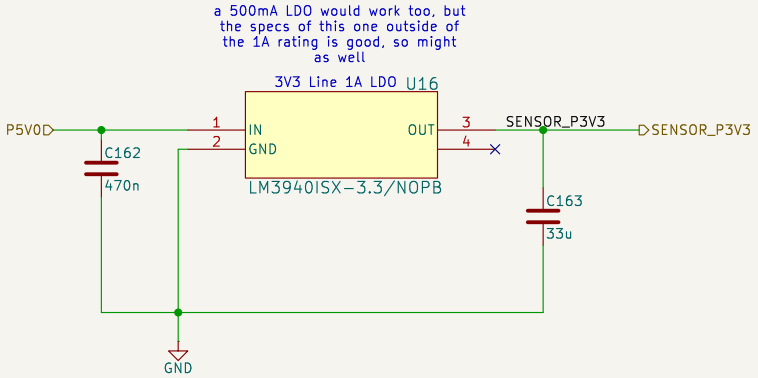


Sheet: /Motor Control Board/Quadrature Counter/
File: Quadrature Counter.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 33/33

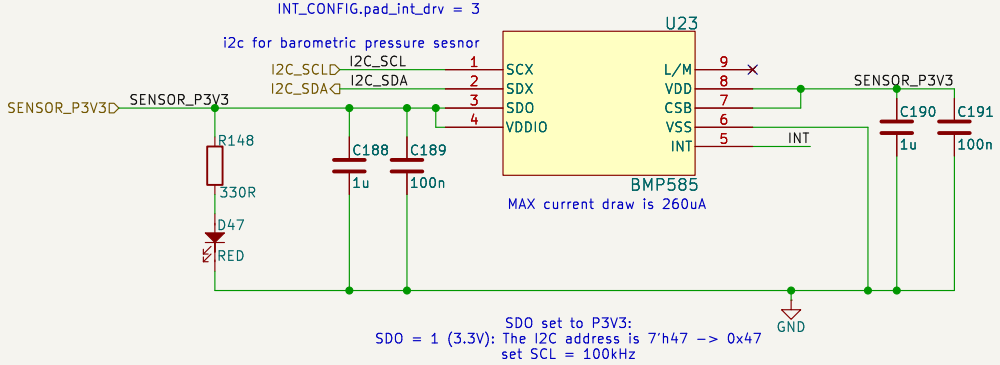

Sensor 3V3 Line
P5V0
P5V0
P5V0
SENSOR_P3V3
SENSOR_P3V3
SENSOR_P3V3
File: Sensor 3V3 Line.kicad_sch
Barometric Pressure Sensor
SENSOR_P3V3
BMP_INT
I2C_SDA
I2C_SCL
BMP_INT
I2C_SDA
I2C_SCL
BMP_INT
I2C_SDA
I2C_SCL
File: Barometric Pressure Sensor.kicad_sch
GPS Sensor
SENSOR_P3V3
I2C_SDA
I2C_SCL
GPS_TX
GPS_RX
I2C_SDA
I2C_SCL
GPS_TX
GPS_RX
GPS_TX
GPS_RX
File: GPS Sensor.kicad_sch
IMU Sensor
SENSOR_P3V3
IMU_nRESET
IMU_nRESET
IMU_nRESET
I2C_SDA
I2C_SCL
IMU_INT
I2C_SDA
I2C_SCL
IMU_INT
IMU_INT
File: IMU Sensor.kicad_sch
Sheet: /Sensor Board/
File: Sensor Board.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 9.0.4
Id: 21/33

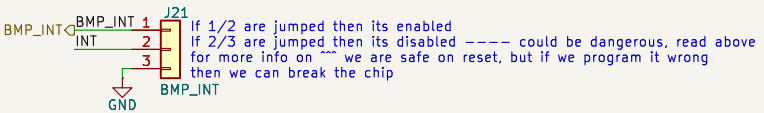


Sheet: /Sensor Board/Sensor 3V3 Line/		
File: Sensor 3V3 Line.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 23/33

Per the datasheet, the unused INT pin is tied to GND to prevent a floating input.
CRITICAL: The interrupt pin must be disabled in software to prevent a short circuit.
Ensure the 'int_en' bit in the INT_CONFIG register (0x14) remains disabled (set to 0).

Datasheet sets the IRQ to 0 on start up, so never turn it on
PAGE49: 0x14 on reset is set to 0x35
PAGE49: <int_en 2bits> <int_od 2bits> <int_pol 2bits> <int_mode 2bits LSB>
PAGE51: 8.5, setup for 0x14, <0> <1> <0> <1>
this means: int_mode = latched, int_pol = active low, int_od = open_drain, int_en = disabled
INT_CONFIG.int_en = 0
INT_CONFIG.od = 1
INT_CONFIG.pol = 0
INT_CONFIG.mode = 1
INT_CONFIG.pad_int_drv = 3

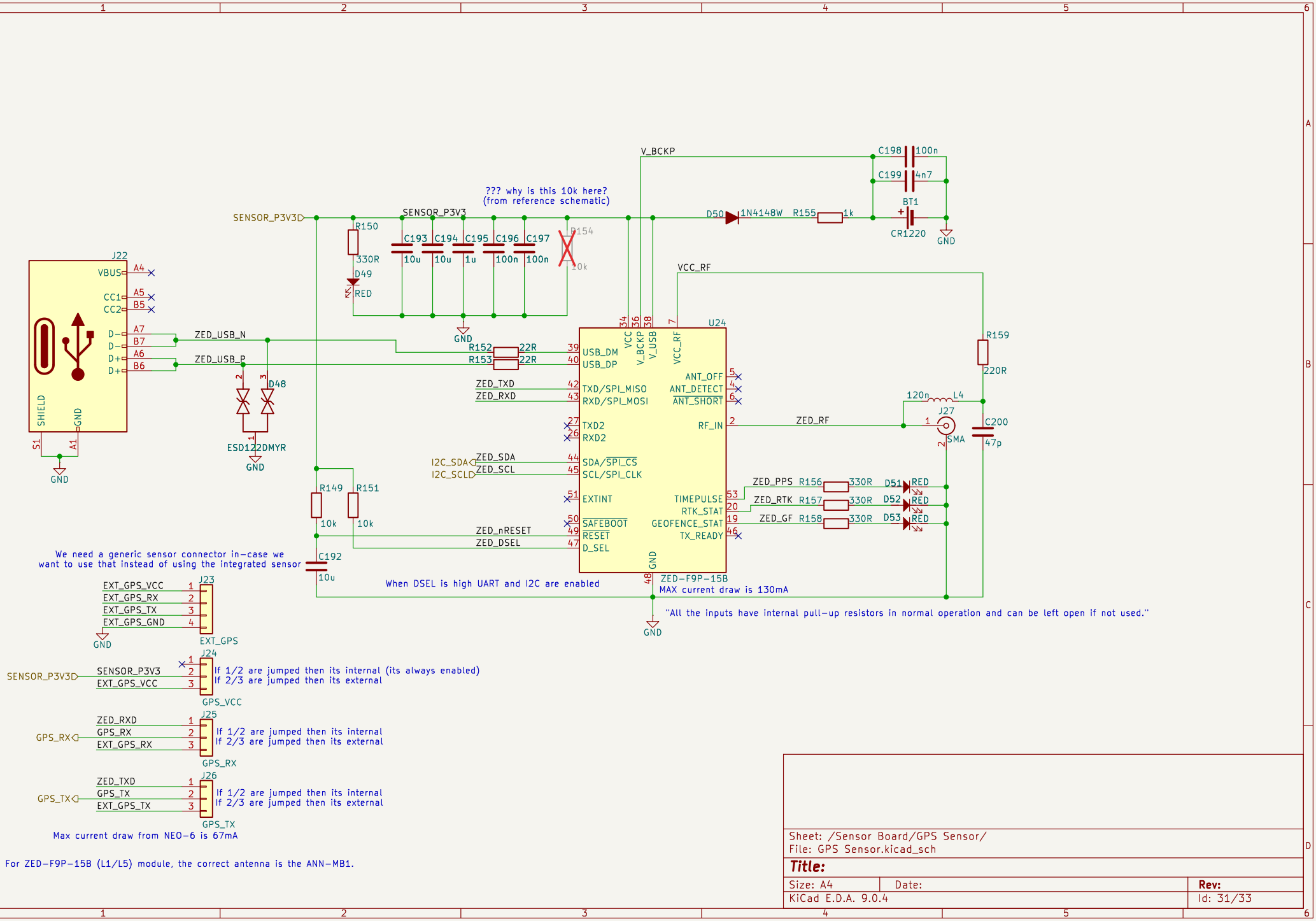


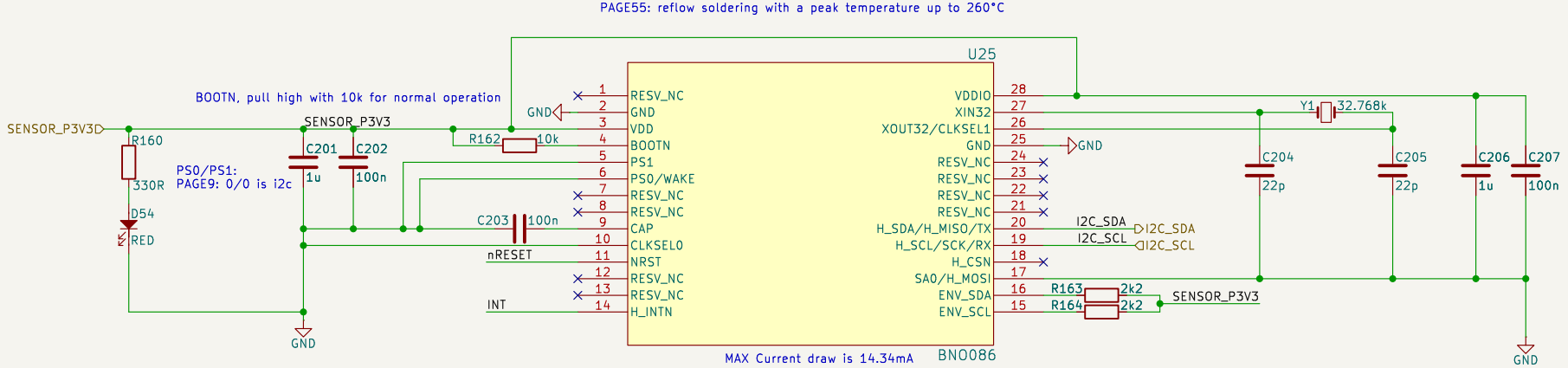


Sheet: /Sensor Board/Barometric Pressure Sensor/
File: Barometric Pressure Sensor.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 30/33


V_BCKP
C198 100n
C199 4n7
BT1
CR1220
GND
??? why is this 10k here?
(from reference schematic)
SENSOR_P3V3
D50 1N4148W R155 1k
R150 330R
D49 RED
C193 10u
C194 10u
C195 1u
C196 100n
C197 100n
R154 10k
VCC_RF
J22
VBUS A4
CC1 A5
CC2 B5
D- A7
D- B7
D+ A6
D+ B6
SHIELD
GND
S1
A1
ZED_USB_N
ZED_USB_P
D48
ESD122DMYR
R152 22R
R153 22R
U24
USB_DM 39
USB_DP 40
VCC 34
V_BCKP 36
V_USB 38
VCC_RF 7
ANT_OFF 5
ANT_DETECT 4
ANT_SHORT 6
ZED_TXD 42
ZED_RXD 43
TXD/SPI_MISO
RXD/SPI_MOSI
TXD2 27
RXD2 26
RF_IN 2
ZED_RF
R159 220R
120n L4
J27 SMA
C200 47p
I2C_SDA ZED_SDA 44
I2C_SCL ZED_SCL 45
SDA/SPI_CS
SCL/SPI_CLK
R149 10k
R151 10k
EXTINT 51
SAFEBOOT 50
RESET 49
D_SEL 47
TIMEPULSE 53
RTK_STAT 20
GEOFENCE_STAT 19
TX_READY 46
ZED_PPS R156 330R D51 RED
ZED_RTK R157 330R D52 RED
ZED_GF R158 330R D53 RED
ZED_nRESET
ZED_DSEL
C192 10u
GND 48
ZED-F9P-15B
MAX current draw is 130mA
When DSEL is high UART and I2C are enabled
"All the inputs have internal pull-up resistors in normal operation and can be left open if not used."
We need a generic sensor connector in-case we
want to use that instead of using the integrated sensor
J23
EXT_GPS_VCC 1
EXT_GPS_RX 2
EXT_GPS_TX 3
EXT_GPS_GND 4
EXT_GPS
J24
SENSOR_P3V3 1
EXT_GPS_VCC 2
3
If 1/2 are jumped then its internal (its always enabled)
If 2/3 are jumped then its external
GPS_VCC
J25
ZED_RXD 1
GPS_RX 2
EXT_GPS_RX 3
If 1/2 are jumped then its internal
If 2/3 are jumped then its external
GPS_RX
J26
ZED_TXD 1
GPS_TX 2
EXT_GPS_TX 3
If 1/2 are jumped then its internal
If 2/3 are jumped then its external
GPS_TX
Max current draw from NEO-6 is 67mA
For ZED-F9P-15B (L1/L5) module, the correct antenna is the ANN-MB1.
Sheet: /Sensor Board/GPS Sensor/
File: GPS Sensor.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. 9.0.4
Id: 31/33



Sheet: /Sensor Board/IMU Sensor/
File: IMU Sensor.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. 9.0.4		Id: 32/33