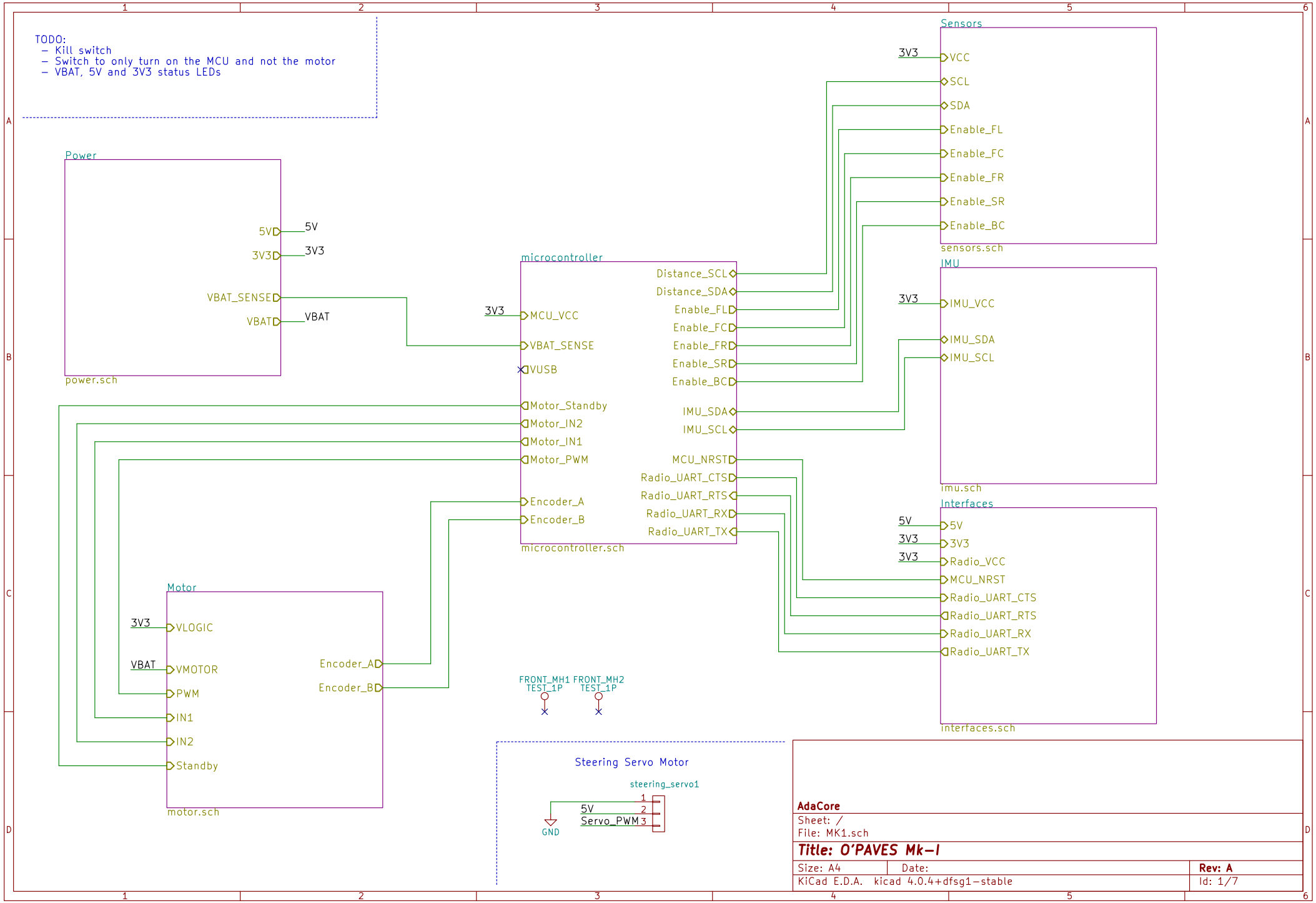
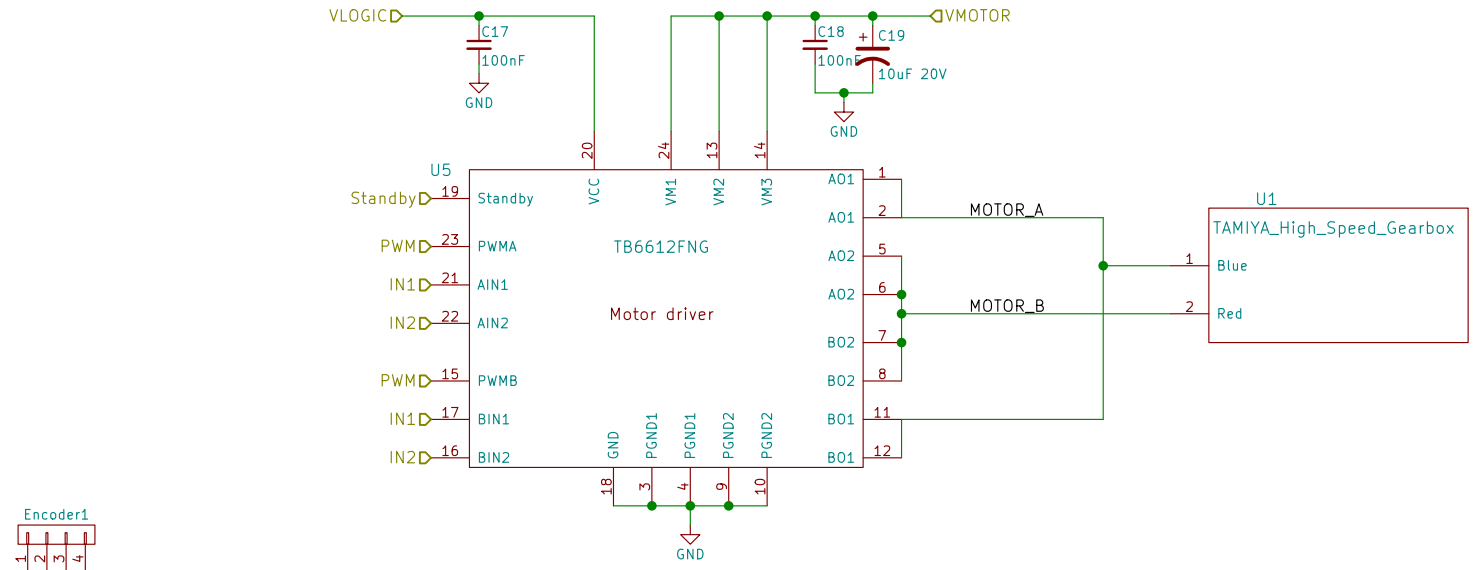
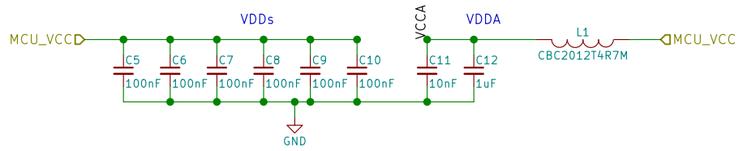
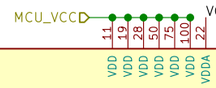

TODO:
– Kill switch
– Switch to only turn on the MCU and not the motor
– VBAT, 5V and 3V3 status LEDs
Power
5V
3V3
VBAT_SENSE
VBAT
power.sch
microcontroller
3V3
MCU_VCC
VBAT_SENSE
VUSB
Motor_Standby
Motor_IN2
Motor_IN1
Motor_PWM
Encoder_A
Encoder_B
Distance_SCL
Distance_SDA
Enable_FL
Enable_FC
Enable_FR
Enable_SR
Enable_BC
IMU_SDA
IMU_SCL
MCU_NRST
Radio_UART_CTS
Radio_UART_RTS
Radio_UART_RX
Radio_UART_TX
microcontroller.sch
Sensors
3V3
VCC
SCL
SDA
Enable_FL
Enable_FC
Enable_FR
Enable_SR
Enable_BC
sensors.sch
IMU
3V3
IMU_VCC
IMU_SDA
IMU_SCL
imu.sch
Interfaces
5V
3V3
3V3
5V
3V3
Radio_VCC
MCU_NRST
Radio_UART_CTS
Radio_UART_RTS
Radio_UART_RX
Radio_UART_TX
interfaces.sch
Motor
3V3
VBAT
VLOGIC
VMOTOR
PWM
IN1
IN2
Standby
Encoder_A
Encoder_B
motor.sch
FRONT_MH1
TEST_1P
FRONT_MH2
TEST_1P
Steering Servo Motor
steering_servo1
5V
Servo_PWM
GND
1
2
3
AdaCore
Sheet: /
File: MK1.sch
Title: O'PAVES Mk-I
Size: A4
Date:
Rev: A
KiCad E.D.A. kicad 4.0.4+dfsg1-stable
Id: 1/7

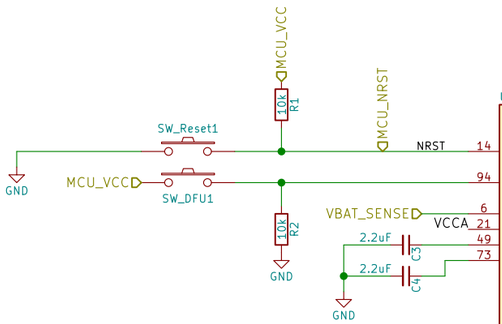


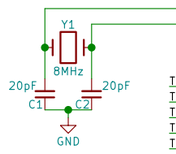
AdaCore	
Sheet: /Motor/	
File: motor.sch	
Title: O'PAVES Mk-I	
Size: A4	Date:
KiCad E.D.A. kicad 4.0.4+dfsg1-stable	Rev: A
	Id: 2/7

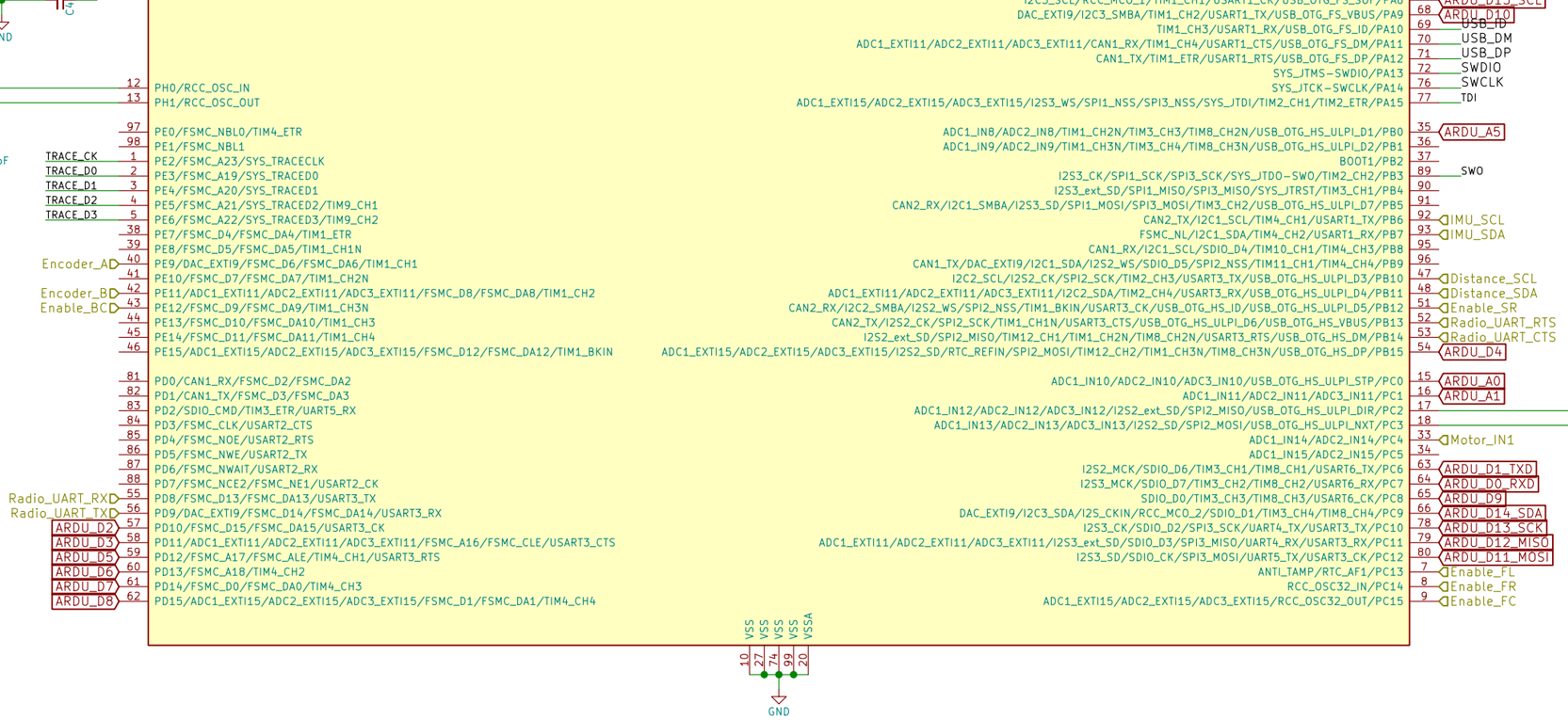
Decoupling



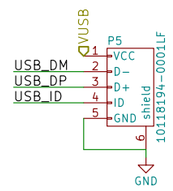






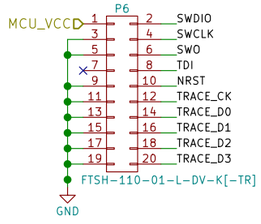


Micro USB

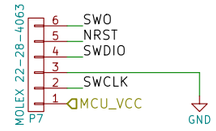


Debug

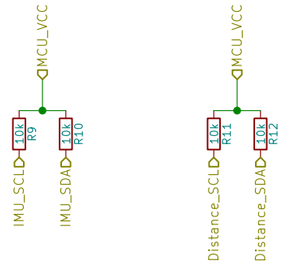
Trace connector



ST-LINK Connector



Pull-UPs



AdaCore

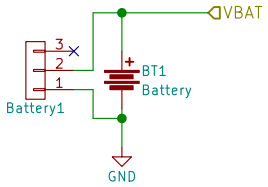
Sheet: /microcontroller/
File: microcontroller.sch

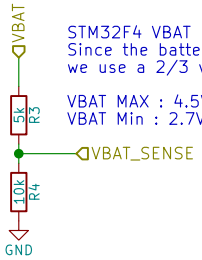
Title: O'PAVES Mk-I

Size: A3	Date:	**Rev: A**
KiCad E.D.A. kicad 4.0.4+dfsg1-stable		Id: 3/7

Battery

3 Cell NiMH:
- Max: 4.5V
- Typical: 3.6V
- Low: 2.7V



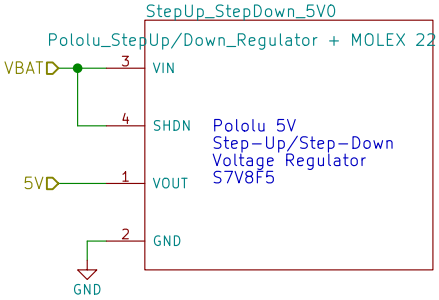


STM32F4 VBAT must be from 1.65V to 3.6V.
Since the battery voltage is outside this range,
we use a 2/3 voltage divider to adjust it.

VBAT MAX : 4.5V -> 3V
VBAT Min : 2.7V -> 1.8V

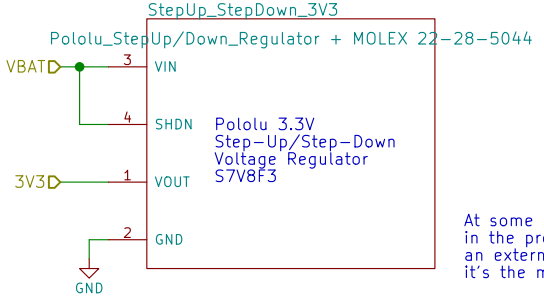
TODO: Reverse polarity protection (maybe)

5V step up / step down regulator



At some point this should be directly in the project rather than relying on an external board. For the moment it's the more convenient solution.

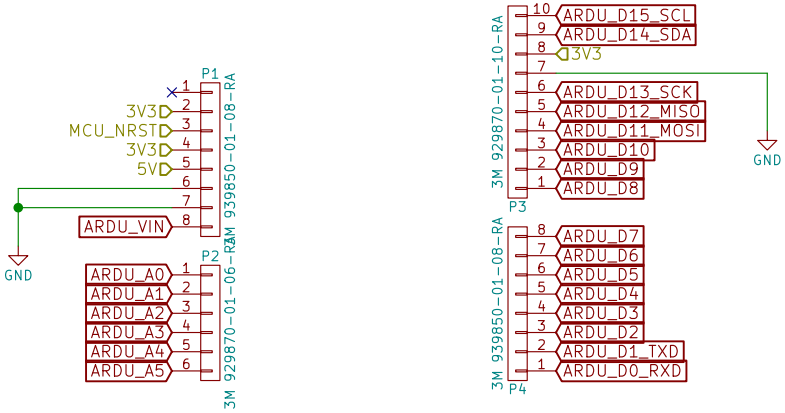
3.3V step up / step down regulator



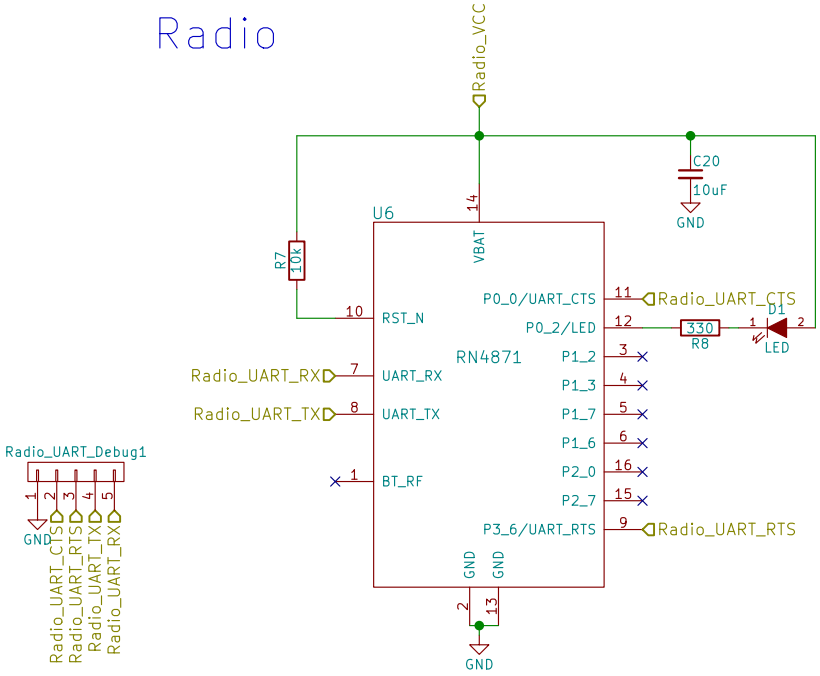
At some point this should be directly in the project rather than relying on an external board. For the moment it's the more convenient solution.

AdaCore		
Sheet: /Power/ File: power.sch		
Title: O'PAVES Mk-I		
Size: A4	Date:	**Rev: A**
KiCad E.D.A. kicad 4.0.4+dfsg1-stable		Id: 4/7

Arduino UNO like headers

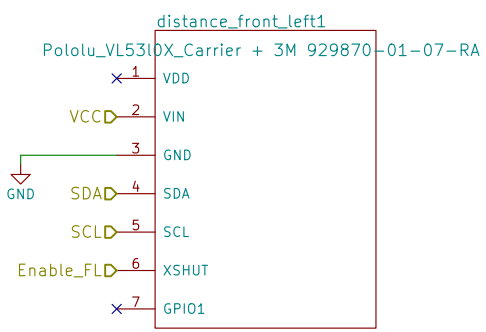


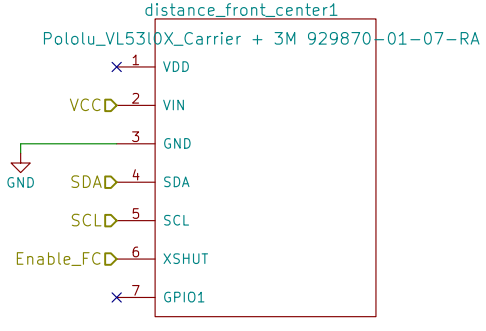
Radio

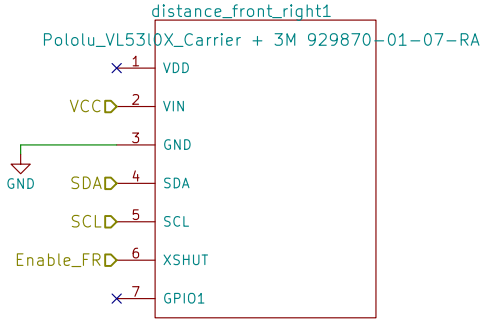


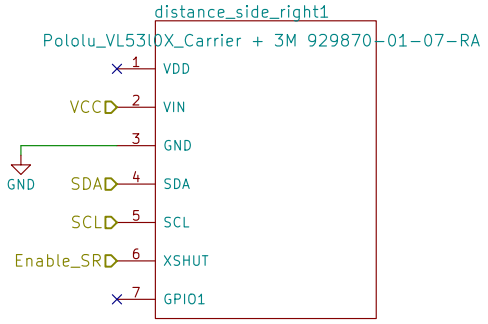
AdaCore		
Sheet: /Interfaces/		
File: interfaces.sch		
Title: O'PAVES Mk-I		
Size: A4	Date:	**Rev: A**
KiCad E.D.A. kicad 4.0.4+dfsg1-stable		Id: 5/7

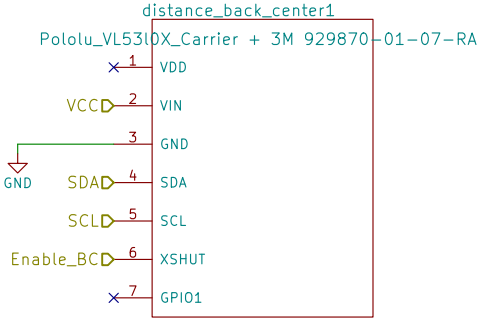
Distance sensors





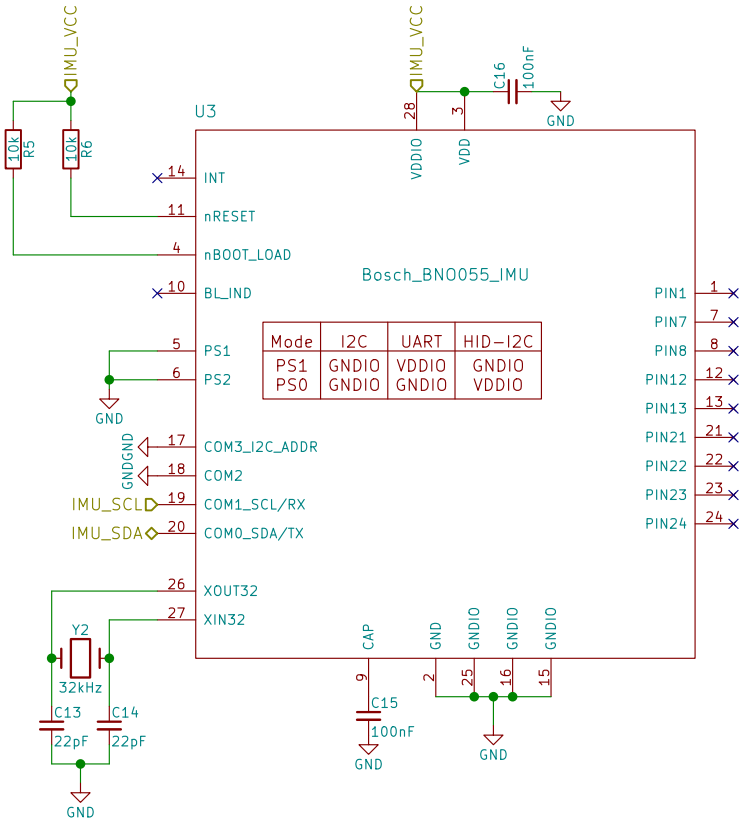






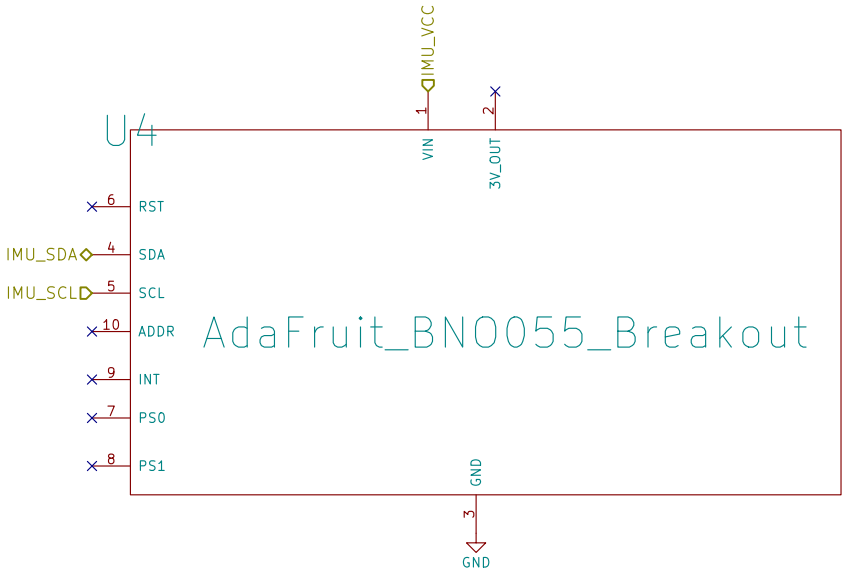
AdaCore		
Sheet: /Sensors/		
File: sensors.sch		
Title: O'PAVES Mk-I		
Size: A4	Date:	**Rev: A**
KiCad E.D.A. kicad 4.0.4+dfsg1-stable		Id: 6/7

IMU



Breakout backup

If for some reason it's not possible to use the on-board IMU (bad circuit, too difficult to solder, etc) we will use a breakout board from AdaFruit.



AdaCore	
Sheet: /IMU/	
File: imu.sch	
Title: O'PAVES Mk-I	
Size: A4	Date:
KiCad E.D.A. kicad 4.0.4+dfsg1-stable	Rev: A Id: 7/7