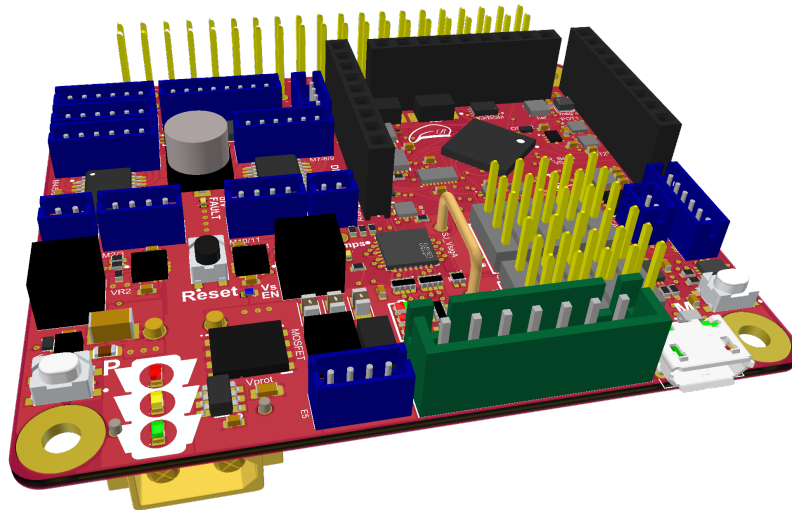


Chapter 5

Berets



Bespoke
Emerging
Robotics
Electronic
Technologies



The **Raspberry**, **Red**, **Black**, **White**, **Green** and **Blue** Berets, from RenaissanceRobotics.com, are a family of 6 bespoke motor control boards that are remarkably compact, powerful, efficient, and extensible. Berets are designed to be operated as daughterboards for essentially *all* of the leading SBCs, as reviewed in Table 1.7, or for standalone operation. Their modern hardware and modular software, described in detail in this chapter¹, are open source, providing a reference platform that can be reduced or extended for a host of practical applications in mobile robotics, industrial automation, precision agriculture, pharmaceutical development, food preparation, remote inspection, smart-grid HVAC, elder care, toys, etc. The motivation for developing and adopting the Beret family of boards is to simplify and accelerate the deployment of bespoke feedback controllers for complex mechatronic systems, empowering robotics students (in high school, college, and beyond), and expediting and streamlining the workflow from lab prototype to commercial product.

Berets are cross-platform², open-design demonstrators of emerging technologies for motor control, power regulation, and coordinated feedback in robotics applications. Most of the ICs selected for the Berets, by **TI** (motor drivers, voltage regulation, power protection, opamps, switches/multiplexers, digital pots, CAN and RS485 transceivers, level shifters, LED driver), **ST** (MCU, magnetometer, barometer), **NXP** (GPIO expander), and **TDK** (IMU), were announced shortly before the Berets were designed (as a Covid lockdown project, in 2020-2021). Being open design, the hardware and software used by the Berets facilitate the development of bespoke derivative boards custom fit to the user's application. To realize this vision, we present below various implementation details and salient features of the several subsystems of the Beret family.

¹In fact, this chapter serves as the datasheet for the Beret family of boards. As a developer of feedback control solutions for mobile robots and cyber-physical systems, the reader must become adept at reading datasheets carefully. This chapter serves as a case study for such datasheets, and should be informative even if different components are ultimately needed in the user's application.

²That is, Berets facilitate (a) the easy porting of real-time multithreaded control code from one linux SBC family to another (see Table 1.7) as algorithmic demands (e.g., vision processing) on the SBC increase, or (b) the elimination of the SBC altogether (in favor of a multithreaded ARM-Cortex M implementation) to substantially reduce system cost for high-volume production (e.g., toys).

We begin with a chapter-level table of contents to help navigate this (rather long) chapter.

Contents

5.1 Overview	5-3
5.2 Power subsystem	5-15
5.2.1 Main power source (Vin)	5-15
5.2.2 Supplemental power sources	5-15
5.2.3 Reverse-voltage, over-current/short-circuit, and ESD protections	5-17
5.2.4 Vin->Vs1 switching regulator	5-17
5.2.5 Vin->Vmb switching regulator	5-18
5.2.6 Vin->3.3V switching regulator	5-18
5.2.7 Vmb->3.3V switching regulator	5-18
5.2.8 Vs2 power opamp	5-19
5.2.9 Switching default power on various connectors and sensors	5-19
5.2.10 Charging and voltage monitoring of Vcoin	5-19
5.3 Control of brushed DC motors & steppers with the DRV8912-Q1	5-21
5.3.1 Parallel mode	5-22
5.3.2 Sequential mode	5-22
5.4 IMU, magnetometer, and barometer	5-23
5.4.1 Data-ready and sensor-driven interrupts	5-24
5.5 STM32G474 microcontroller features, pinouts, and GPIOs	5-28
5.5.1 Real-time clock (RTC), and scheduled/commanded wakeup from VBAT mode	5-29
5.5.2 Customization with Quad-SPI Flash	5-29
5.6 Connectivity and i/o	5-30
5.6.1 Quadrature encoder counters and connectors	5-30
5.6.2 PWM-based servo and ESC controllers and the Signal Headers	5-30
5.6.3 Encoding for IR communication	5-31
5.6.4 CAN FD and RS485 transceivers and connectors	5-31
5.6.5 USART, UART, and LPUART modules and connectors	5-31
5.6.6 I2C modules and the I2C Header	5-32
5.6.7 SPI modules and the SPI Header	5-33
5.6.8 LEDs, Buttons, and Displays	5-33
5.6.9 USB Micro-B connector and Device Firmware Upgrades	5-34
5.7 Analog subsystem	5-35
5.7.1 Generation of Vs2 = 1.2V to 2.1V, and two high-power DAC outputs	5-35
5.7.2 Tunable filtering/gain of two unipolar, bipolar, or differential ADC inputs	5-36
5.7.3 Analog Header and user-developed analog filters	5-37
5.7.4 Voltage monitoring of Vin, Vs1, Vs2, and the individual battery cells	5-38
5.8 Beret Shields	5-39
5.9 MB headers, MB header breakout SHIMs, and ID EEPROMs	5-41
5.9.1 RPi-compatible motherboards (MBs)	5-41
5.9.2 96B-compatible MBs	5-44
5.9.3 BB-compatible MBs	5-45
5.10 Layout	5-47
5.10.1 Overall organization and power flow	5-47
5.10.2 Layer stackup, signal routing, and high-density integration (HDI)	5-48
5.10.3 EMI and signal-integrity considerations	5-49
5.11 Bill Of Materials (BOM)	5-51
5.12 Schematics	5-53

Beret	MB header	PCB dimensions	balance	analog	peripherals	bolt pattern
Raspberry	RPi	full size 65 × 56	✓	✓	3.3V, 5V	RPi
Red	RPi	full size 65 × 56	✓	×	3.3V, 5V	RPi
Black	96B	full size 85 × 54	✓	✓	3.3V, 5V	96B
White	BB	full size 3.4" × 2.15"	✓	✓	3.3V, 5V	BB
Green , Blue	(none)	half size 39 × 61.5	×	✓	3.3V only	(custom)

Beret	MB voltage	Drivers:	motors	encoders,	servos/ESCs	Remote Busses	Optional
Raspberry	5V MB	24 HB	8	7	10	CAN/RS485	Coin, Flash
Red	5V MB	12 HB	4	5	5	(local only)	Coin, Flash
Black	12V MB	24 HB	8	7	10	CAN/RS485	Coin, Flash
White	5V MB	24 HB	8	7	10	CAN/RS485	Coin, Flash
Green	(none)	0 HB	0	5	10	CAN/RS485	Coin, Flash
Blue	(none)	12 HB	4	5	5	(local only)	Coin, Flash

Beret	PCB edges	mounting hole centers	IMU center	Analog pin 9
Ras , Red	(± 32.5, ± 28)	(± 29, ± 24.5)	(25,0)	(0,?)
Black	(± 42.5, ± 27)	(± 38.5, -8.5), (± 38.5, 23)	(?,?)	(0,?)
White	(± 1.7", ± 1.075")	(-1.475", ± 0.825"), (1.125", ± 0.95")	(?,?)	(0,?)
Green , Blue	(± 19.5, ± 30.75)	(± 16, 27.25), (0, -27.25)	(12.5,0)	(-12.5,0)

Table 5.1: Essential features of the six initial Beret variants. All distances in mm except on the **White**, for which distances are given in inches. The center of each board is taken as the reference point when defining the PCB edges, the mounting hole centers, and the center of the IMU coordinate system. On all Berets, M2.5 hardware may be used; on the **White** + BB, #4-40 hardware fits a bit better in its 0.125" holes.

5.1 Overview

The essential features of the six initial Beret variants are outlined in Table 5.1. In particular:

- the **Raspberry** (aka **Ras**) Beret may be operated as a daughterboard for recent versions of the RaspberryPi,
- the entry-level (lower-cost) **Red** Beret is a partially-populated **Raspberry** Beret³ with reduced specs,
- the **Black** Beret may be operated as a daughterboard for SBCs following the 96Boards CE specification,
- the **White** Beret may be operated as a daughterboard for the BeagleBone Black and AI, and
- the **Green** Beret is designed for standalone (or, wired remote) operation with 3.3V peripherals only.
- the **Blue** Beret is designed for standalone mobile applications with 3.3V peripherals only.

Note that, in fact, all six Beret variants may be operated standalone when properly programmed.

In this section, we briefly summarize the subsystems used in this family of boards; the balance of §5 explains these subsystems in greater detail. Note that the modern switching regulators, motor drivers, MCU, MOSFET, op amps, and other power components discussed below are best-in-class in terms of their efficiency. This, of course, means that the system can run a bit longer on a single battery charge than it could otherwise. Perhaps even more important, however, is that a reduced amount of waste heat is generated by these components, thus significantly improving the capability of Berets to handle high-current operating conditions.

³Because red is a less-fancy way of saying raspberry; Prince wrote a [song](#) about the latter, not the former...

POWER

The **Raspberry**, **Red**, and **White** Berets, dubbed **5V MB** Berets, target 5V motherboards. They are powered over an **XT30** connector by a **Vin = 6.2V – 28V** input at up to 15A continuous / 20A peak, thus accommodating a **2S – 6S LiPo** (3.1 – 4.2V per cell) or LiHV (3.2 – 4.35V per cell), 3S – 7S LiFe (2.5 – 3.65V per cell), 7S – 18S NiMH (1 – 1.5V per cell), or a **7V – 28V wall adapter**, which is down-regulated as follows:

- the **Vin->Vs1 switching regulator** provides **Vs1 = 4.8V to min(12V, 0.8*Vin)** at up to **6A** for servos & ESCs,
- the **Vin->Vmb switching regulator** provides **Vmb = 5.1V** at up to **6A** for an RPi or BB compatible MB,
- the **Vmb->3.3V switching regulator** provides **3.3V** at up to **3A** for logic circuits and connected sensors, and
- the **Vs2 power opamp** provides **Vs2 = 1.2V to 2.1V** at up to **± 400 mA**.

The **12V MB Black** Beret targets 12V motherboards. It is powered over an XT30 connector by a **Vin = 12V – 28V** input at up to 15A continuous / 20A peak, accommodating a **4S – 6S LiPo** or LiHV, 5S – 7S LiFe, 12S – 18S NiMH, or a **12V – 28V wall adapter**, which is down-regulated slightly differently:

- the **Vin->Vs1 switching regulator** provides **Vs1 = 4.8V to min(12V, 0.8*Vin)** at up to **6A** for servos & ESCs,
- the **Vin->Vmb switching regulator** provides **Vmb = min(12V, 0.8*Vin)** at up to **6A** for a 96B compatible MB,
- the **Vmb->3.3V switching regulator** provides **3.3V** at up to **3A** for logic circuits and connected sensors, and
- the **Vs2 power opamp** provides **Vs2 = 1.2V to 2.1V** at up to **± 400 mA**;

in addition, per the **96Boards (96B) CE specification**, the 96B motherboard, if one is attached, down-regulates the Vmb line and passes back (via the low-speed header) **5V** at up to **1A**. The 12V MB Beret thus does not itself have a 5V regulator (and, thus, the 5V subsystem on this Beret is not functional unless a 96B compatible motherboard is attached).

The **Green** and **Blue** Berets are built for compact standalone applications with 3.3V peripherals only, bypassing the generation of both Vmb and 5V (saving both board area and cost). They are powered over an XT30 connector by a **Vin = 5V – 28V** input at up to 15A continuous / 20A peak, accommodating a **2S – 6S LiPo** or LiHV, 2S – 7S LiFe, 5S – 18S NiMH, or a **5V – 28V wall adapter**, which is down-regulated as follows:

- the **Vin->Vs1 switching regulator** provides **Vs1 = 4.8V to min(12V, 0.8*Vin)** at up to **6A** for servos & ESCs,
- the **Vin->3.3V switching regulator** provides **3.3V** at up to **3A** for logic circuits and connected sensors, and
- the **Vs2 power opamp** provides **Vs2 = 1.2V to 2.1V** at up to **± 400 mA**.

In general, on all Berets:

- **Vin** powers the motor drivers directly,
- **Vs1** powers the signal headers (for servos & ESCs),
- **Vmb** powers the attached motherboard,
- the **3.3V** and **5V** lines power the logic circuits on the Beret, the JSTs, and the digital and analog headers, and
- all digital outputs operate at **3.3V TTL**, and all digital inputs are **5V tolerant**.

Again, note that the Vmb and 5V lines are absent on the **Green** and **Blue** Berets.

MICROCONTROLLER (MCU)

For real-time control of brushed (BDC) motors, stepper motors, servo motors, electronic speed controllers (ESCs), and brushless (BLDC) motors, coordinating motor commands with a wide variety of sensor inputs, Berets incorporate⁴ a 100-pin 170 MHz (213 DMIPS) **STM32G474VE** with an **ARM Cortex-M4 core**, 512 KB flash⁴, 128 KB SRAM, and integrated DSP, FPU, and **CORDIC** (transcendental) & **FMAC** (filter math) accelerators.

Berets run **FreeRTOS** and leverage the efficient, portable, open-source **Robot Control library** (written in **C**) for driving all hardware. A **ROS** interface to these subroutines is under development.

⁴Note that the entry-level **Red** Beret uses a **STM32G474VB** instead of a **STM32G474VE**, with 128 KB flash instead of 512 KB, but is otherwise identical in terms of the specs described here.

Significantly, Berets break out many of the [STM32G474's dedicated subsystems](#), each of which operate without loading the main ARM core on the STM itself. An additional 24 GPIOs are provided by a dedicated [GPIO expander](#). Note that computationally-heavy tasks (e.g., for vision-based situational awareness) should be deferred to the attached linux-based motherboard.

MOTORS, ENCODERS, SERVOS & ESCs

The **Raspberry**, **Black**, and **White** Berets, dubbed **24 HB** Berets, have **24 half bridges**, with drivers for simultaneous independent bidirectional control of **8 brushed DC motors** operating at V_{in} at up to 12A total; the connectors to these half bridges may be [ganged together](#) in various ways to control: 2 motors at 6A, 4 motors at 3A, 4 motors at 2A + 4 motors at 1A, etc., or attached in a unique [sequential mode](#) for independent bidirectional control of up to 24 1A motors at reduced duty cycles. The 24 HB Berets also provide dedicated hardware support for **7 quadrature encoders** and **10 servos or ESCs** of a wide variety of sizes and types.

The **12 HB Red** and **Blue** Berets have **12 half bridges**, with drivers for control of **4 brushed DC motors** operating at V_{in} at up to 6A total; the connectors to these half bridges may be ganged together to control: 1 motor at 6A, 2 motors at 3A, 2 motors at 2A + 2 motors at 1A, etc., or attached in sequential mode for independent bidirectional control of up to 12 1A motors at reduced duty cycles. The **Red** and **Blue** Berets also provide dedicated hardware support and pinouts for **5 quadrature encoders** and **5 servos or ESCs**.

The **0 HB Green** Beret actually has no half bridges itself (which, admittedly, at first seems peculiar for a motor control board!). This Beret leverages daughterboards, dubbed [Beret Shields](#), for implementing the specific motor drivers that may be required in any given application. The **Green** Beret provides dedicated hardware support and pinouts for **5 quadrature encoders** and **10 servos or ESCs**.

BATTERY MONITORING & CHARGING

The **Raspberry**, **Red**, **Black**, and **White** Berets, dubbed **full size** Berets, monitor [individual battery cell voltages](#) when running. The **half size Green** and **Blue** Berets, on the other hand, forgo individual battery cell voltage monitoring; when running on a battery, they only monitors the [overall battery charge](#).

Note that, on all Berets, [battery charging](#) must be done via a (high-quality) off-board battery charger.

SENSORS & CONNECTORS

All six Berets include a sensitive 6-axis [IMU](#) (3 accels + 3 gyros), 3-axis [magnetometer](#), and [barometer](#), in addition to discrete connectors for driving a host of [I2C](#), [SPI](#), and [UART](#) sensors and other devices (such as [GPS/GNSS units](#) and [DSM radio receivers](#)), a [USB Micro-B](#) input for programming, and numerous channels configurable as [GPIOs](#). In addition, the **Raspberry**, **Black**, **White**, and **Green** Berets, dubbed the **CAN/RS485** Berets, include high-speed [CAN-FD](#) and full duplex [RS485](#) transceivers and connectors.

Servos and ESCs are supported on the Berets by industry-standard (triple row, 0.1" pitch, 3A per pin, PTH) [Signal Headers](#), arranged in one or two easy-to-use 3x5 cluster(s) of pins. All six Berets also have an Arduino-style (1x9, 0.1" pitch, 3A per pin, PTH) [SPI Header](#) and [I2C Header](#).

ANALOG SUBSYSTEM

The **Raspberry**, **Black**, **White**, **Green** and **Blue** Berets also have a 1x9, 0.1" pitch [Analog Header](#), limited to 0V to 3.3V operation in a unipolar or bipolar setting, with reference GND at 0V or $V_{s2} \approx 1.65V$, and:

- two 16-bit **ADCs** with tunable gain (x1 to x4096) & tunable second-order low-pass filtering ($f_c = 34$ to 3400 Hz),
- two 12-bit **DACs** with ± 400 mA outputs, and
- a ± 400 mA **opamp** pinout (V_+ , V_- , V_o).

The entry-level **Red** Beret forgoes the entire analog subsystem, including the Analog Header.

CONNECTOR STANDARDS: RECON AND YUKON

Sturdy [JST-ZH connectors](#) (1.5 mm pitch, 1A per pin, PTH) are used for motors, encoders, U(S)ART, CAN, and RS485. The pin order on all JST-ZH connectors and single-row headers follow the standards defined in §4:

- E1-2 (I2Cd), E3-4 (UARTt), E5 (UARTr), E6-7, USART (UARTb/SPIb) follow the Recon Basic, Recon I2C, Recon UART-T, Recon UART-R, and Recon SPI pin order standards defined in §4.2,
- CAN follows the Yukon CAN standard defined in §4.4.3, while RS485 follows the Yukon RS485-Y (host) standard on the **Raspberry**, **Black**, and **White** Berets, and the RS485-A (client) standard on the **Green** Beret (and, when acting as UARTa, the Recon UART-T standard on all four of these Berets). On the **Blue** Beret, the corresponding connector is UART-T only.
- the SPI and IC2 Headers follow the Stackable Extended Recon standards defined in §4.2.8, and
- the Analog Header follows the Yukon Basic standard defined in §4.4.1.

EXPANSION BOARDS: BERET SHIELDS

The (1x9) SPI Header, I2C Header, and (if included) Analog Header, in addition to the (3x5) Signal Header A, are all aligned on a 0.1" pitch grid, facilitating the easy and secure mounting of stackable COTS and user-designed [Beret Shields](#) with additional analog and digital circuitry.

MOTHERBOARD (MB) HEADERS

Berets communicate with motherboards using SPI. To make this connection easier,

- the **Raspberry** and **Red** Berets, dubbed the **RPi** Berets, have a 2x20, 0.1" pitch stackable [RPi header](#),
- the **Black** Beret, dubbed the **96B** Beret, has a 2x20, 2 mm pitch stackable [96B header](#), and
- the **White** Beret, dubbed the **BB** Beret, has a 2x23, 0.1" pitch stackable [BB header](#).

The **Green** and **Blue** Beret do not have any board-specific MB headers, though they may be connected locally to virtually any MB (e.g., over SPI, CAN or RS485, etc), or operated standalone.

On the **full size** Berets, the MB Header may be broken out with additional compact PCBs (a.k.a. [SHIMS](#)), and standardized [EEPROMs](#) are included to identify the Berets appropriately to connected MBs.

OTHER FEATURES

The STM's dedicated hardware timers (for, e.g., encoder counting and PWM generation) are highly reconfigurable, with additional (unidirectional) encoder counters easily configurable on the Signal Headers, or additional PWM outputs (for more servos and ESCs) easily configurable on the encoder connectors, if necessary.

The battery state of charge is indicated with a [three bicolor LED power gauge](#). Three [buttons](#) (reset/shut-down, pause, mode), three user-programmable [stoplight LEDs](#), and various [status LEDs](#) are also included.

An (optional) rechargeable **Vcoin = 2.6V to 3.05V coin cell** may be installed on Berets to keep the [real-time clock \(RTC\)](#) current, thus facilitating scheduled system wakeups.

Berets are also easily upgraded with an (optional) low-cost [4 MB to 512 MB flash IC](#).

Note: **Ras** and **Red** are 36.4 cm², **Black** is 45.9 cm², **White** is 47.2 cm², **Green** and **Blue** are 24 cm².

FEATURE SET SUMMARY, PINOUTS, LAYOUT, AND FUNCTIONAL DIAGRAMS

The distribution of the above-described features over the six different Beret versions is summarized in Table 5.1. Layout and functional representations of each is given in Figures 5.1-5.6.

The pinouts of the several connectors on the Berets are listed in Table 5.2. The rest of §5 is devoted to explaining these subsystems further. In particular, discussion of how the specific channels on the STM32 are hooked up to these several connectors is summarized in §5.5; see in particular Figure 5.9 and Table 5.4.

Connector ^A	Pins	Signal
<i>MB Header</i> ^D	all	(see §5.9)
USB Micro-B	1 2,3,4	USB_5V DM, DP, GND
XT30 (Power In)	1 2	Vin (up to 28V/20A) GND
<i>Balance</i> ^{E,F}	1 2 3 4 5 6 7	B1 (cell 1 low) B2 (cell 1 high / 2 low) B3 (cell 2 high / 3 low) B4 (cell 3 high / 4 low) B5 (cell 4 high / 5 low) B6 (cell 5 high / 6 low) B7 (cell 6 high)
<i>M1</i>	1,2	M1a , M1b
<i>M2-3</i> (<i>M4-5-6</i> , <i>M7-8-9</i> , <i>M10-11</i> , and <i>M12</i> are similar)	1-4	M2a , M2b , M3a , M3b
E1-2 ^G (I2Cd)	1,2 3 4 5 6	[3.3V/5V/Vs1/off] ^B , GND E1b /SDA/G0 E2a /SCL/G1 E2b /SMBA/G2 E1a /RES/G3
E3-4 (UARTt)	1,2 3,4 5,6	[3.3V/5V] ^B , GND E3a /Tx/G0, E3b /G1 E4a /G2, E4b /G3
E5 (UARTr)	1,2 3,4	[3.3V/5V] ^B , GND E7a /Rx/G0, E7b /G1
<i>E6-7</i>	1,2 3,4 5,6	[3.3V/5V/Vs1/off] ^B , GND E5a /G0, E5b /G1 E6a /G2, E6b /G3
Connector ^A	Pins	Signal
USART (UARTb/SPIb)	1,2 3,4 5,6 7 8	[3.3V/5V] ^B , GND Tx/MOSI/G0, Rx/MISO/G1 SCK/G2/ Vcoin , CTS/SSa/G3 RTS/INT/SSb/G4 RES/SSc/G5
<i>CAN</i> ^C	1,2	CANH, CANL
<i>RS485</i> ^C (<i>UARTa</i>)	1 2 3 4	Y or A or [3.3V/5V] ^B Z or B or GND A or Y or Tx/G0 B or Z or Rx/G1
Signal Header A ^H (S1-S5/I2Cb/I2Cc) (top row)	1 2 3 4 5	S1 /I2Cb_SDA/I2S_MCK S2 /I2Cb_SCL/I2Cc_SMBA S3 /I2Cb_INT S4 /I2Cc_SDA S5 /I2Cc_SCL
(middle row)	1-5	[Vs1/Vin] ^B
(bottom row)	1-5	GND (<i>Signal Header B</i> is similar, with S6 - S10 in top row, and restricted to Vs1 in the middle row)
SPI Header (SPIa)	1,2,3 4,5,6 4,5,6 7,8,9	5V , 3.3V , GND MOSI /SD/G0, MISO /G1 SCK /G2, SSa /WS/G3 SSb /G4, SSa /IR_OUT/G5
I2C Header (I2Ca)	1-5 6,7 8,9	Vcoin , Vs2 , Vs1 , 3.3V , GND SDA/G0, SCL /BOOT0/G1 G2, G3
<i>Analog Header</i> ^{A,I}	1,2 3,4,5 6,7,8 9	DAC1buf , DAC2buf V+ , V- , Vo Vref , ADC1 , ADC2 ADC2filt

Table 5.2: Pinouts (primary role: **output**, **input**, i/o, **power/ground**) of the connectors on all six Berets, with some of the connectors in *italics* dropped on the **12 HB** and **0 HB** Berets (for details, see Table 5.1).

USAGE NOTES

A. Connector pins, except on the MB Header, are numbered W (left) to E (right), or N (top) to S (bottom) [see Figs 5.1-5.6]. All digital signals on the encoder and USART JSTs, and on the Signal, SPI, and I2C Headers, may be configured as GPIOs in software. All connectors follow the Recon and Yukon standards of §4. Digital outputs all operate 0 to 3.3V TTL. Digital inputs are all 5V tolerant, however: **warning**: all pins on the Analog Header are limited to 0 to 3.3V operation.

B. **Warning**: the power supplied to this pin (default is **bold**) may be changed by the user, using a multiplexer, a backside solder jumper, or a shunt connector (see §5.2.9).

C. **Warning:** the differential CAN and RS485 transceivers operate 0 to 3.3V; RS485 may be changed to 0 to 5V via a backside solder jumper. Using a 4PDT switch, if the Beret GPIO RS485_SEL=1 (see §5.5), the RS485 connector functions as a full-duplex RS485-Y (host) connector on **Raspberry**, **Black**, and **White** Berets, and as a RS485-A (client) connector on **Green** Berets; if RS485_SEL=0, it functions as a UART-T connector. On the **Blue** Beret, the corresponding connector is UART-T only.

D. The wiring of the Beret's SPI3 channel, its user-defined {MB_G0, MB_G1, MB_G2} GPIOs, and (optionally) its three sensor interrupt channels to the motherboard (MB) header varies somewhat amongst the different Beret versions, as described in §5.9. All JSTs and single-row pin headers on the Berets are connected to the STM and its associated ICs on the Beret, not to the attached MB. Compact breakout-boards (aka SHIMs) are available separately to conveniently break out the functionality on the corresponding MB Headers.

E. The Balance connector is JST-XH, and all other JSTs on the Beret are JST-ZH. As opposed to, e.g., the fragile SMD JST-SHs on the Beaglebone Blue, all JSTs and headers on the Beret are PTH, for durability (still, be gentle!).

F. The custom JST-XH connector on the Beret is modified in such a way as to connect securely to the (3-pin to 7-pin) JST-XH balance connectors on 2S-6S batteries. Included with this custom connector is a set of small/snug "dummy plugs" that may be used to cover the 1 to 4 unused pins on this connector when using 5S to 2S batteries, which helps to provide a more secure fit, and also prevents the balance connector from being plugged into the wrong pins when swapping such batteries to recharge; always make sure that one end of the JST-XH balance connector lines up with white triangle printed on the PCB. **Warning:** these dummy plugs can be pried out with a small screwdriver when necessary, but do so only when the board is not connected to a power source.

G. As specifically permitted in the flexibility of the Recon Basic standard in §4.2.1, note that the order of the {E1a, E1b, E2a, E2b} signals on the E1-2 connector are permuted in such a way that, if used for I2Cd, this connector is in standard Recon I2C pin order using the hardware I2C4 channel on the STM.

H. The signal headers, designed to drive up to⁵ 10 servos (motors packaged with simple control logic designed to turn a shaft to a desired position and hold it there) or ESCs (motors packaged with control logic designed to accelerate/decelerate a shaft to a desired angular velocity and keep it there), accept all modern 0.1" pitch 3x1 servo connectors^{6,7}, including both Fubata J and JR/Universal/Hitec S/Airtronic Z. Low-power servos and ESCs typically power both their control logic and their motor using the GND and Vs1 pins on the signal header. High-power servos and ESCs, on the other hand, typically power their control logic using the GND and Vs1 pins on the signal header, but power their motor directly from the power supply (i.e., not via the Beret!). The Beret provides Vs1 = 4.8V to 12V (adjustable in software), at up to 3A on each individual servo connector (6A total), to support a broad range of small servos, ESCs, or other peripherals. Y Adapters (available separately) may be inserted between high-capacity LiPos and the Beret to break out power separately for higher-current servos and ESCs.

I. The (buffered) DAC outputs operate between GND and 3.3V at up to 400mA. The ADC inputs may be configured for software-tunable amplification (x1 to x4096) and 2nd-order low-pass filtering (with $f_c = 34$ to 3400 Hz) of unipolar or bipolar analog signals, or for differential comparison of two analog signals (see §5.7).

⁵The 12 HB Berets have one bank of 0.1" pitch 3x5 pin headers, for driving up to 5 servos or ESCs. The 24 HB Berets have two such banks of 0.1" pitch 3x5 pin headers, for driving up to 10 servos or ESCs; the signal headers are broken out into separate 3x5 banks like this because many servo connectors are, unfortunately, about 0.1 mm wider than the standard 2.54 mm pin spacing near their tips, and 0.2 mm wider away from their tips, which gradually adds up; putting more than N=5 such connectors next to each other on standard 0.1" 3xN pin headers ultimately puts undue stress on the header pins.

⁶Note: modern servo connectors, all of which have power (the red wire) attached to the central pin, may easily be accidentally plugged into the 3x5 pin headers backwards. This is completely safe; it will result in the corresponding servo not functioning correctly until the plug is reversed, but it will not damage either the Beret or the servo.

⁷**Warning:** do not use old servos with Airtronic T connectors with the Beret; damage to the Beret and/or servo will result. This obsolete configuration can be recognized easily by the fact that, on it, power is connected to pin 1 or 3, not to the central pin.

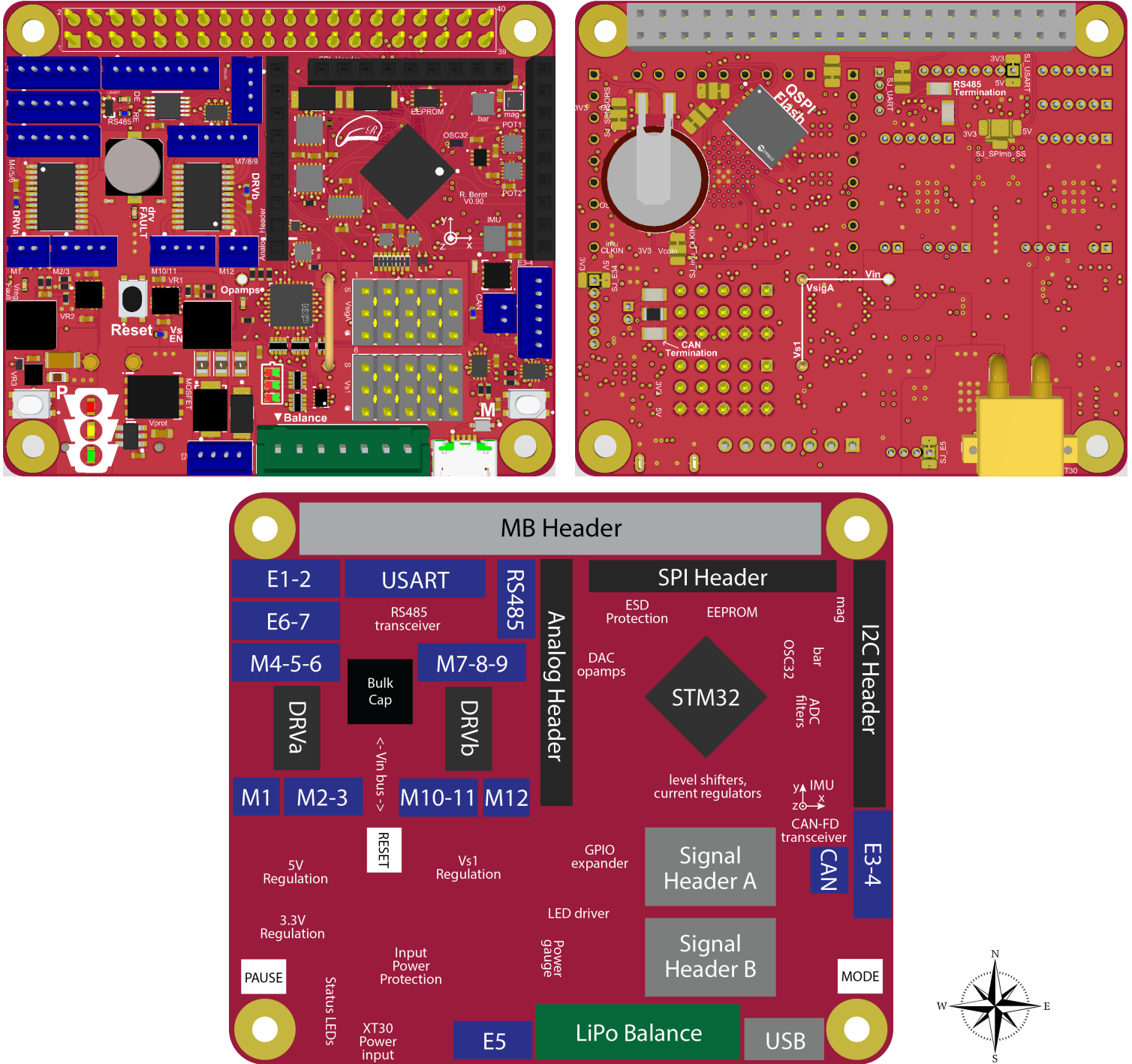


Figure 5.1: (top) Layout of the (left) front and (right) back sides, and (bottom) a functional cartoon, of the **Raspberry Beret** (**RPi** Header, **full size**, **5V MB**, **24 HB**, **CAN/RS485** busses). In the cartoon: blue denotes JST-ZH connectors of various sizes; gray is used for the MB Header, the 3x5 Signal Headers, and the USB Micro-B port; green is used for the JST-XH Balance connector. Beret Shields (§5.8) may be placed atop the Analog Header, the SPI Header, the I2C Header, and (optionally) the first row of the 3x5 Signal Header A (SigA), with the Beret’s main logic ICs lying on the PCB beneath. The XT30 main power input and all solder jumpers are situated on the back side of the board; the user may (optionally, also on the back side) install a 6mm x 8mm Flash IC, a rechargeable VL-1220/FCN coin cell, and/or passives for RS485 or CAN termination.

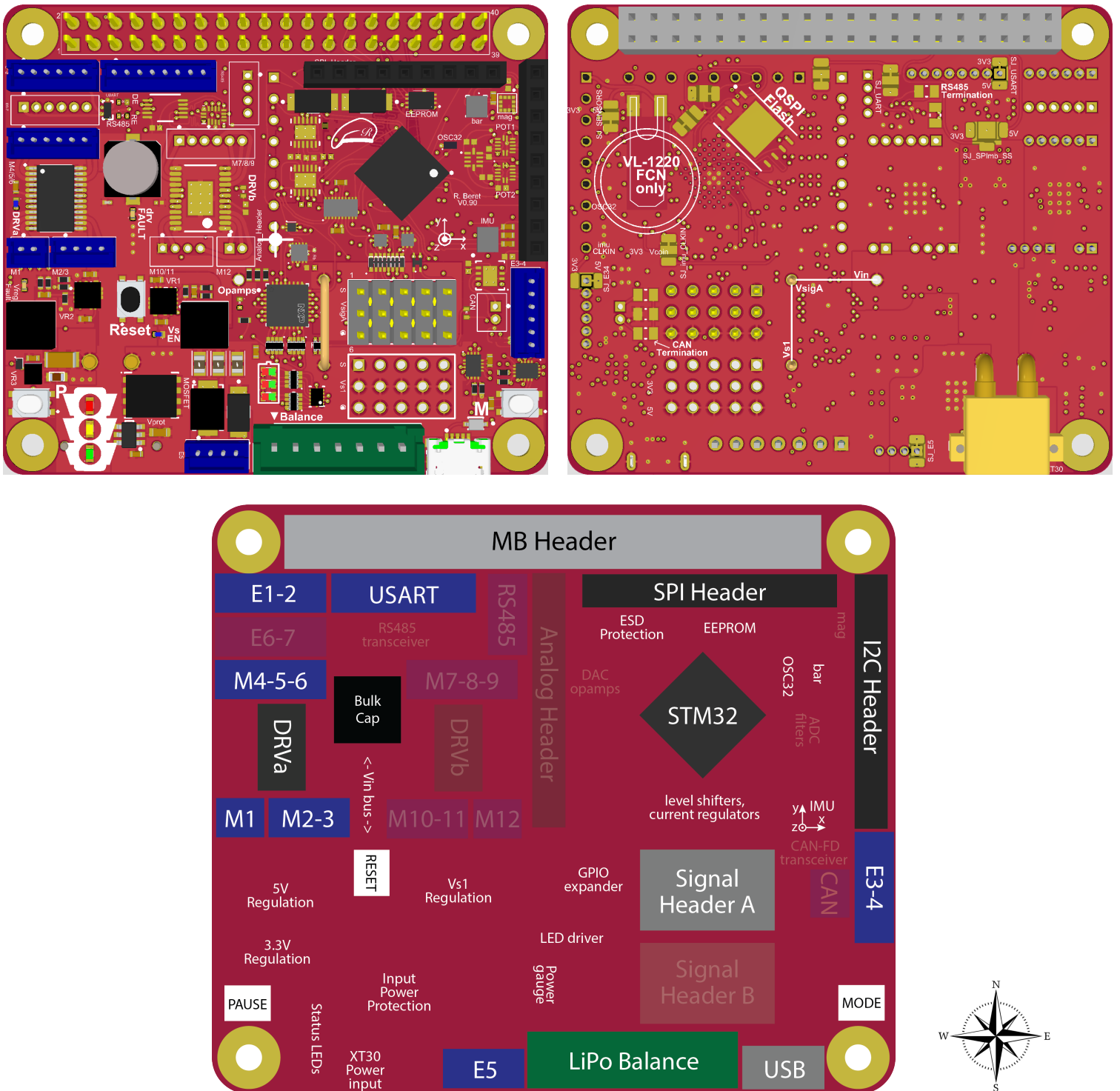


Figure 5.2: (top) Layout of the (left) front and (right) back sides, and (bottom) a functional cartoon, of the **Red Beret** (**RPi** Header, **full size**, **5V MB**, **12 HB**); for legend, see Fig. 5.1. Note that the entry-level **Red Beret** is simply a partially-populated **Raspberry Beret**, with the lower-cost STM32G4VB implemented (with 128KB flash instead of 512KB).

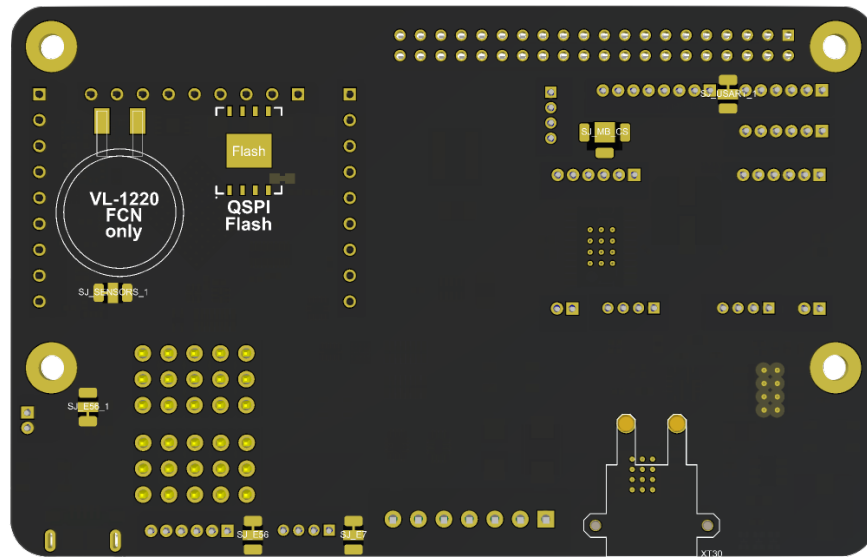
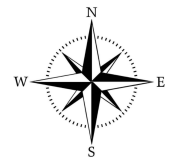
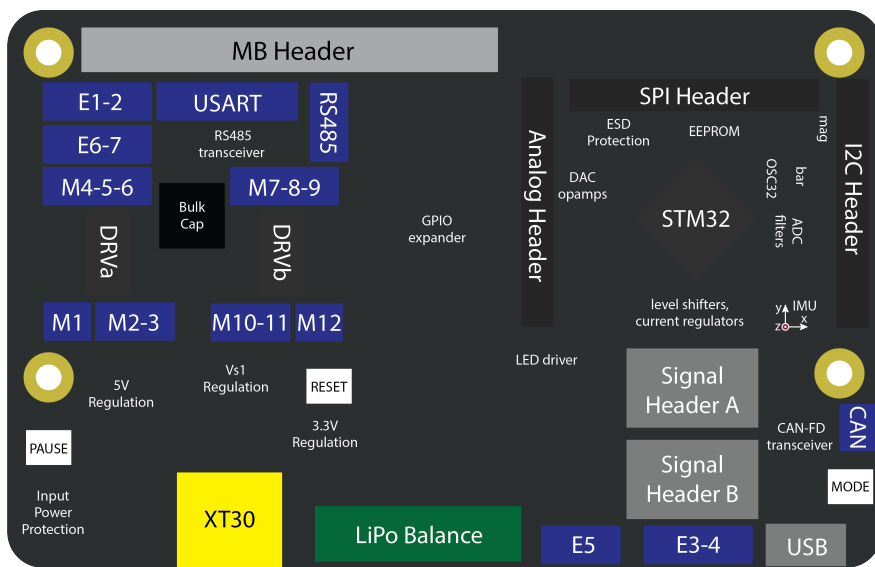
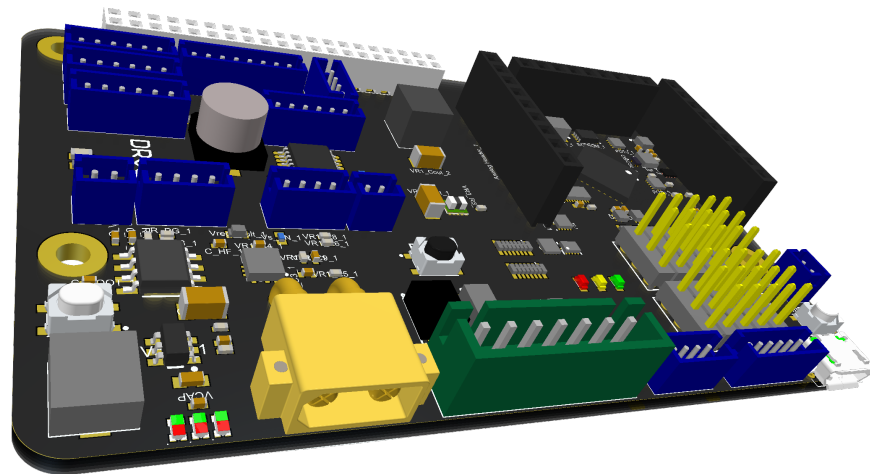


Figure 5.3: Layout of the (top) front and (bottom) back sides, and (center) a functional cartoon, of the **Black** Beret (**96B** Header, **full size**, **12V MB**, **24 HB**, **CAN/RS485** busses); for legend, see Fig. 5.1. **NOTE:** this board is still under development.

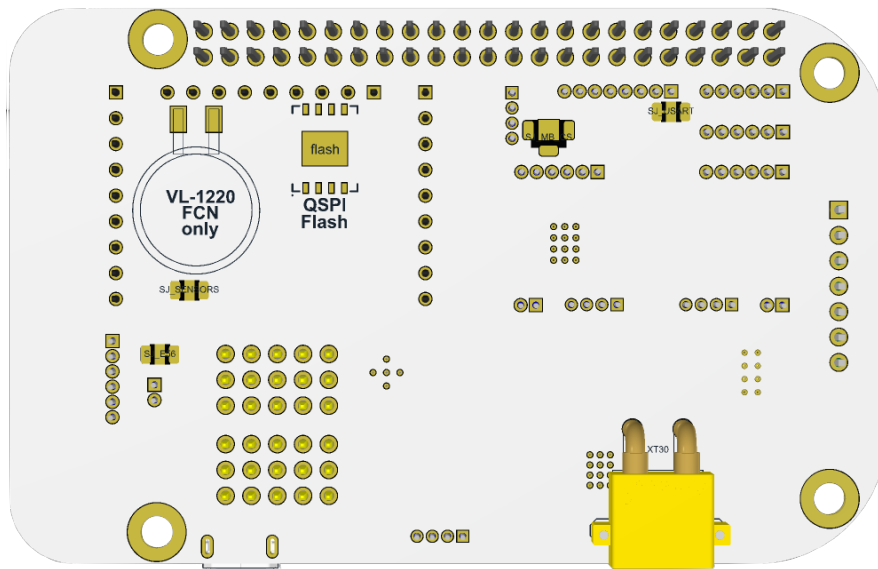
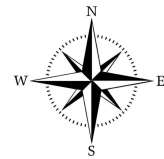
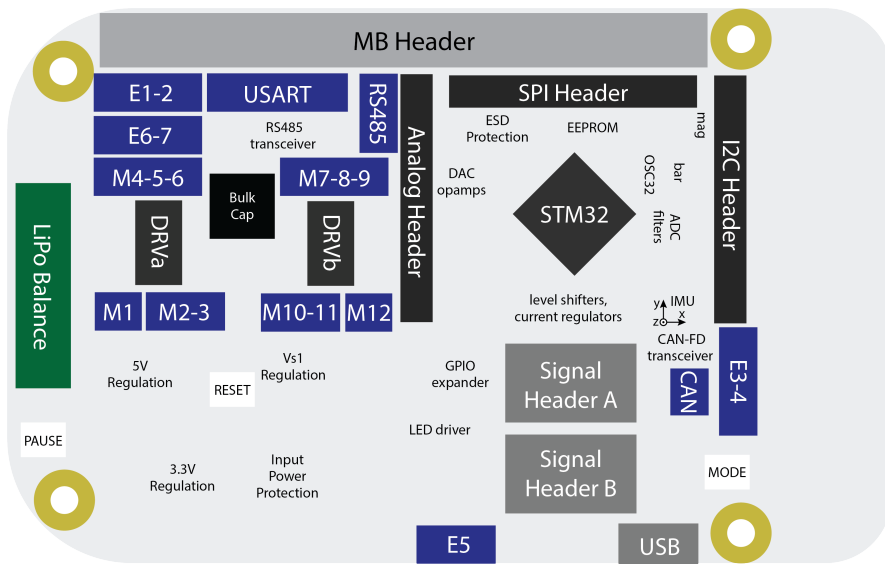
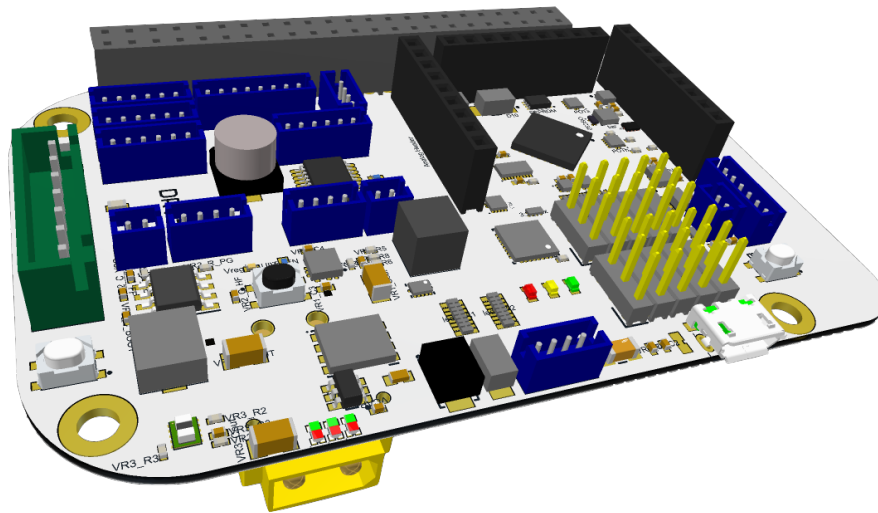


Figure 5.4: Layout of the (top) front and (bottom) back sides, and (center) a functional cartoon, of the **White** Beret (**BB** Header, **full size**, **5V MB**, **24 HB**, **CAN/RS485** busses); for legend, see Fig. 5.1. **NOTE:** this board is still under development.

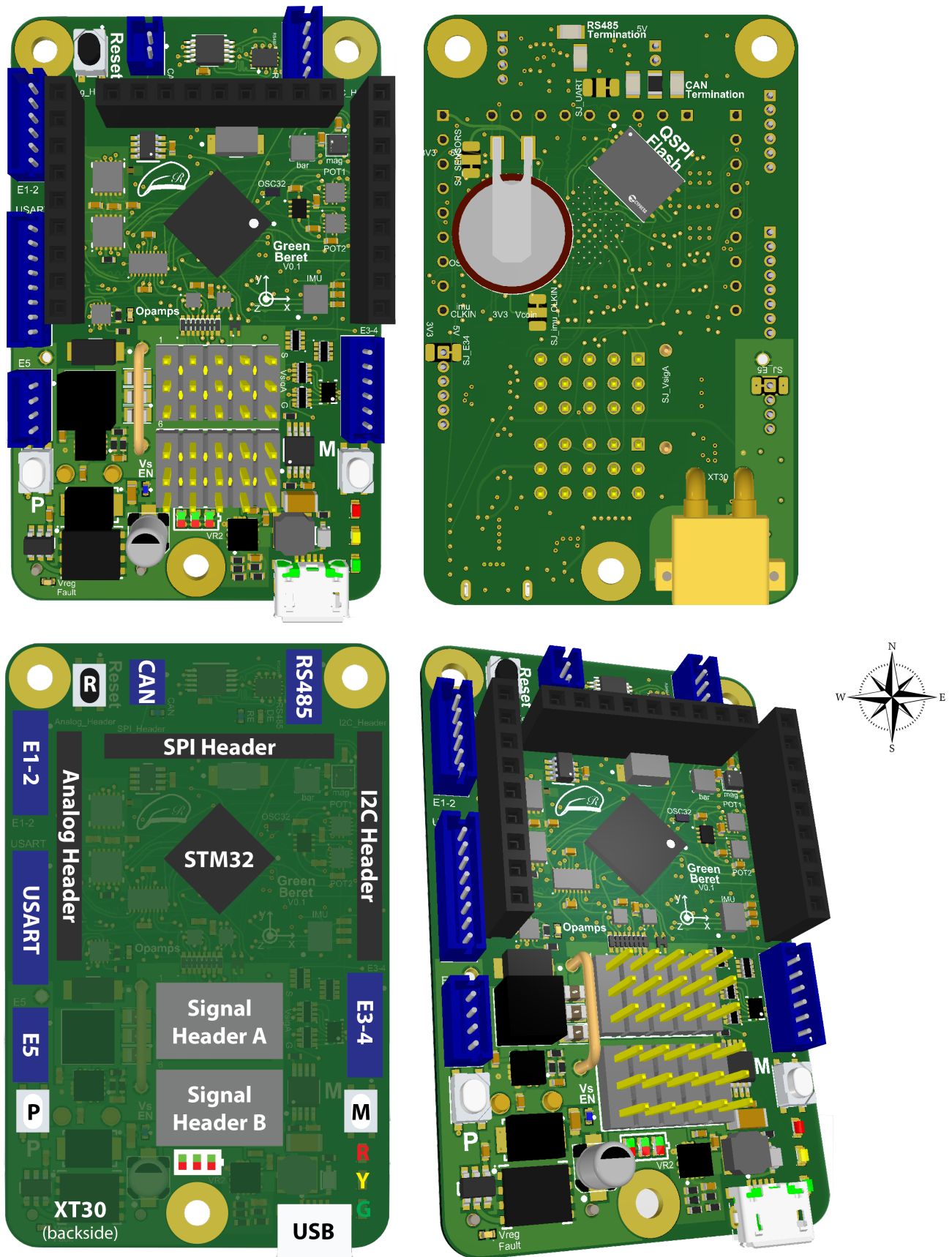


Figure 5.5: Layout of the (a) front and (b) back sides, (c) a functional cartoon, and (d) an oblique view, of the **Green Beret** (standalone, **half size**, **0 HB**, **CAN/RS485** busses); for legend, see Fig. 5.1.

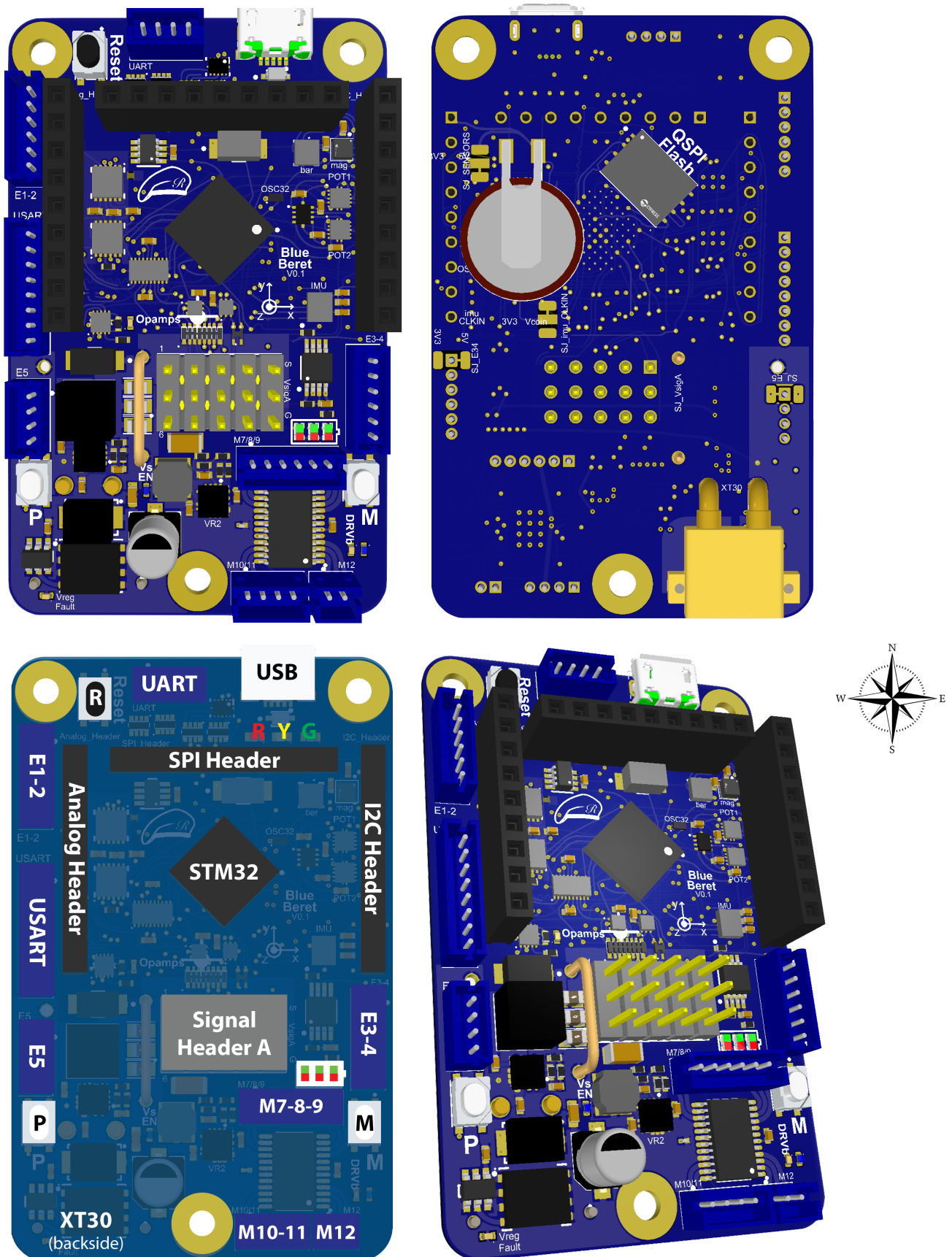


Figure 5.6: Layout of the (a) front and (b) back sides, (c) a functional cartoon, and (d) an oblique view, of the **Blue** Beret (standalone, **half size**, **12 HB**); for legend, see Fig. 5.1.

5.2 Power subsystem

5.2.1 Main power source (Vin)

As outlined in §5.1, the power input ($V_{in} = 6.2V - 28V$ on the **5V MB** Berets, and $V_{in} = 12V - 28V$ on the **12V MB** Beret, at up to 15A continuous / 20A peak) is brought onto the Berets via a backside XT30 connector. The XT30 jack provides the sole source of power for the motor drivers and servo headers on the Berets; power provided over the XT30 jack also drives all other circuits on the Beret and the attached MB, via the voltage regulators discussed in the sections that follow. Adapters are available separately to convert the connector on the main power source, if necessary, from XT60, Traxxas, EC3, and Deans (aka “T”) connectors, 5.5 mm x 2.1 mm barrel jacks⁸, and 4.75 mm x 1.65 mm barrel jacks⁹ to the XT30 standard¹⁰.

WARNINGS: when using high-power batteries like LiPos, it is essential that the user be aware of several delicate issues regarding their selection, care, use, charging, storage, and disposal. One of the more thorough online guides available on this important subject is available [here](#); key takeaways include:

- i. invest in a high-quality charger appropriate to the batteries you will be using,
- ii. though LiPo cells can operate from 3.0 V to 4.2 V, a range of 3.1 V to 4.1 V extends battery life significantly,
- iii. always balance while charging, as some cells will discharge faster than others (especially in older batteries),
- iv. never use [parallel charging boards](#),
- v. store batteries at about half charge ($\sim 3.8V$ per cell for LiPos),
- vi. don't tug on bare wires ([AB Clips](#) can help),
- vii. watch for swelling (dispose of swollen battery immediately), and
- viii. always dispose of batteries properly.

Keeping an [inexpensive tester](#) handy for checking the charges of all cells of a given battery is often useful.

DC motors and steppers connected to a Beret operate directly at V_{in} , after the TVS and reverse-voltage protections (see §5.2.3); this power is modulated by the H-bridges (§5.3) to control the motor speed and direction. Y adapters may be inserted between the power source and the Beret to drive high-current servos, ESCs, and BLDC motors directly off the power source, at V_{in} . The choice of the voltage V_{in} of the power source (e.g., a 2S to 6S LiPo) should thus be made according to the voltage required the motors and steppers (and, the high-current servos, ESCs, and BLDC motors) to be used. Note that the voltage of a LiPo battery reduces significantly as the battery is discharged; tuned feedback gains used to drive the motors (see §5.3) may be scaled inversely with the battery voltage to offset this effect over time. In contrast, the behavior of most servos and ESCs, which incorporate their own feedback, is relatively insensitive to the voltage supplied, as long as it remains within the recommended operating range; selecting V_{s1} (for low-current servos/ESCs) or V_{in} (for high-current servos/ESCs) near the upper end of this operating range generally provides increased maximum torque.

5.2.2 Supplemental power sources

As discussed in §5.2.1, if main power is provided to the Beret over the XT30 connector (which is, in fact, the only way that the Beret will actually power the motor drivers and signal headers), then this power also drives the STM and all other circuits on the Beret, as well as the attached MB, as detailed in the sections that follow. There are a few other places that supplemental power might be brought onto a Beret (or, to a Beret+MB system), however, that the user also needs to be well familiar with.

⁸Wall adapters with 5.5 mm barrel jacks, with positive 2.1 mm pins and negative sleeves, is an [emerging standard](#) in robotics applications that the user should stick with. **Warning:** this standard for 5.5 mm barrel jacks is not universal; some 5.5 mm barrel jacks have either negative pins with positive sleeves, and/or 2.5 mm pins. If you are using a wall adapter with such a barrel jack, which is not recommended, a custom adapter to the XT30 standard will be required; make absolutely certain you get the polarity right!

⁹Wall adapters with 4.75 mm barrel jacks, with positive 1.65 mm pins operating at 12V, is the power supply [standard](#) for 96Boards.

¹⁰A substantial power source, capable of delivering 4A or more, is recommended for driving motors, servos, etc. with a Beret.

In normal operation, power is provided to a Beret via ONE of the following three sources:

1. the main XT30 power input on the Beret (connected to a high-power battery or a wall wart), OR
2. via the MB Header, up from a MB that is powered directly, OR
3. via the USB Micro-B connector on the Beret itself (connected, e.g., to a laptop or desktop computer).

Modes 2 and 3, which are discussed further in the following few paragraphs, may be useful at times for programming the STM on the Beret, for reading the Beret's onboard sensors, and for testing various other devices connected to the Beret's JSTs. **WARNING:** Mode 1 and mode 2 must NOT be used at the same time, to eliminate the possibility of these two power sources interfering with each other. Note that mode 3 is protected with a diode, and may thus be wired in conjunction with mode 1 or mode 2 (or, used stand alone) without concern.

Power from the MB, up through the MB Header, to the Beret

As mentioned above, RPi, 96B, and BB (and compatible) motherboards can all be powered by a Beret over the corresponding MB Header. Power may INSTEAD be provided to such motherboards directly, e.g.:

- via USB C on an RPi-4B or BeagleBone AI,
- via a USB Micro B input attached to a substantial (2A to 3A) 5V wall wart on RPi-2 and RPi-3 models,
- via the DC input jack on the BeagleBone Black and 96B CE boards,

etc. If power is provided to such motherboards directly, in most cases this power is transmitted back up to the Beret via the same MB Header pins that are otherwise used to transmit power down to the MB from the Beret. When this happens, the Vmb circuit on the Beret is energized with sufficient power (again, in most cases) to drive the Vmb->3.3V switching regulator (see §5.2.7), thus booting the STM on the Beret, and powering up the Beret's onboard sensors (see §5.4) as well as any small devices connected to the Beret's JSTs.

WARNING: As mentioned above, one should NOT provide power directly to both the Beret (via the XT30) and an attached MB¹¹. Doing this can fry the Beret, the MB, and/or one or both of the power sources used.

5V power from the USB Micro-B port on the Beret

An unpowered, isolated (not connected to any MB) Beret may also, conveniently, be powered via the 5V line on the USB Micro B connector on the Beret, which may also be used for programming the (3.3V) STM on the Beret, for reading the Beret's onboard sensors, and for testing various other small devices connected to the Beret's JSTs. This programming mode is quite convenient, as it only requires a single standard USB cable ([Type A Male to Micro B Male](#)), in addition to the Beret and your laptop/desktop computer.

Unfortunately, the current capability of USB ports on most laptop/desktop computers is not sufficient to drive both the Beret and a connected MB, and attempting to do so is thus not advised.

Note that the USB specification prohibits a USB client from back-driving a USB host with 5V power on the USB voltage bus. Doing such can, in fact, cause major damage to a (potentially, expensive) USB host (i.e., your laptop/desktop computer). To protect against this possibility, a schottky diode (rated to [10V/3A](#) on the 5V MB Berets, and rated to [40V/3A](#) on the **Black**, **Green**, and **Blue** Berets), with a 0.35 to 0.55V voltage drop, is implemented between the 5V power bus of the USB Micro B connector on the Beret and the Vmb line on the Beret (or, the Vin line on the **Green** and **Blue** Berets). Thus, when powered by the USB Micro-B port alone, the Vmb (or, Vin) line will be held to about 4.5V, which is sufficient to generate 3.3V and fire up the STM.

When the Beret is otherwise powered (either via the XT30, or by the MB via the MB Header), the diode mentioned above will be under reverse bias, and no current will flow (either in to, or out of) the 5V power bus on the USB Micro B connector on the Beret; this configuration is thus considered to be **safe**.

¹¹In certain special cases, this situation is known to be ok. Specifically, on the **5V MB** Berets, the Vin->Vmb switching regulator (see §5.2.5) is operated in a mode that can handle ~5V being driven at its output, whether or not power is provided (from the XT30) at its input [thus, no protection diode (aka ZPD) is required to isolate the Vin->Vmb regulator on the Beret]; note also that the last few paragraphs of the [RPi HAT design guide](#) specifically state conditions in which ~5V may safely be provided over the MB Header to an already-powered RPi. Regardless, there truly appears to be no practical reason to push your luck by trying this. So, don't.

RTC backup battery (VL-1220/FCN rechargeable 3V coin cell)

All Berets come configured with backside solder pads for a [VL-1220/FCN](#) (only!) rechargeable 3V coin cell. **Warning:** special care is required careful when soldering the VL-1220/FCN battery bracket onto the Beret. Panasonic's [Lithium Handbook](#), which describes the VL-1220/FCN on pages 56-59, gives the following advice (on page 80): *Do not allow the soldering iron to make direct contact with the body of the battery. Proceed with the soldering quickly (within 5 seconds) while maintaining the iron tip temperature at about 350°C, and do not allow the temperature of the battery body to exceed 85°C.*

The VL-1220 keeps the real-time clock (RTC) of the STM powered up during replacements of the main battery (for recharging), and facilitates the scheduled wake-up of the Beret and the attached MB from a low-power sleep (aka VBAT mode), during which main power is turned off to both the Beret and the MB (see §5.5.1). The VL-1220 battery is [automatically recharged](#), when necessary, by the 3.3V line on the Beret when the STM is in run mode.

5.2.3 Reverse-voltage, over-current/short-circuit, and ESD protections

All Berets implement a [TVS diode](#) across the XT30 input, and [TI TPD6E004](#) TVS diode arrays on USB and S1-S10, for protection from [Electrostatic Static Discharge](#) (ESD, i.e.. voltage spikes). On the 5V MB Berets, the {Vs1, 5V, 3.3V} lines are protected from voltage spikes by {[13V](#), [5.6V](#), [3.6V](#)} 1.5W zener diodes; on the 12V MB Beret, the {Vs1, 12V, 5V, 3.3V} lines are protected using {[13V](#), [13V](#), [5.6V](#), [3.6V](#)} 1.5W zener diodes. 100 Ω resistor arrays are used on S1-S10 for over-current (short-circuit) protection. The modern voltage regulators (§5.2.4-5.2.7), motor drivers (§5.3), and CAN & RS485 transceivers (§5.6.4) on the Beret provide further over-current and ESD protections. A [TI CSD18510Q5B](#) MOSFET [with $R_{DS(on)} = 0.96 \text{ m}\Omega$ and $t_{d(off)} = 44 \text{ ns}$] is implemented at the XT30 to provide reverse voltage protection, and to turn off the board when necessary; the gate voltage of this MOSFET is optimally adjusted by a [TI LM74700-Q1](#) ideal diode controller, wired as suggested in figure 21 of its datasheet. The battery gauge LEDs (see §5.6.8) are illuminated whenever this main MOSFET is enabled [and, thus, the Vmb (5V or 12V) and 3.3V switching regulators (see §5.2.5-5.2.7) are powered up].

WARNINGS: Notwithstanding the modern power protections implemented, as outlined above, it is still quite possible to “release the magic smoke”¹² from a Beret; the user is thus strongly urged to carefully:

- ensure proper polarity at the XT30 ([red/positive wire to the right](#) when viewed from above; see Figures 5.1-5.6),
- keep the power input at the XT30 at or below [28V](#), preventing any voltage spikes beyond this value, and
- avoid any short circuits between any two pins (see Table 5.2), or draw current on any subsystem beyond the [maximum current](#) values highlighted in §5.2.4 - §5.2.8, and §5.3, of this datasheet.

In §5.2.4 through §5.2.8, the voltage regulation circuits on the Berets are described in detail.

5.2.4 Vin->Vs1 switching regulator

The [Vin->Vs1 switching regulator](#) implemented on all five of the Berets, for generating a software-adjustable [Vs1 = 4.8V to min\(12V, 0.8*Vin\)](#) at up to [6A](#) (for SigA, SigB), is the [TI TPS56637](#), coupled with a [5.6 μH inductor](#) and three [22 μF output capacitors](#). The TPS56637 converter is wired as suggested in figure 17 its datasheet, with EN and PG wired to Vs1_EN and Vreg_FAULT on the GPIO expander (see Table 5.6), taking¹³ $R7 = 6.49 \text{ k}\Omega$ and replacing R6 by a [28.7 kΩ](#) resistor in series with half of [POT3](#), a [TI TPL0102-100](#) dual digital pot, in Rheostat

¹²It is sometimes said that “magic smoke” makes an IC work, as whenever this smoke escapes, the IC ceases to function; some vendors even sell [replacement magic smoke](#). **Warning:** more seriously, the Beret can deliver very high currents indeed; severe injury or death may result from its misuse, so extreme caution and adherence to the warnings in this datasheet is absolutely required.

¹³These resistor values were selected by noting eq. 5 in the [TPS56637 datasheet](#), applying a 3% margin to the desired limits, solving the following simultaneous systems of equations in Matlab, and rounding to the nearest common resistor values:

$$\text{syms x y; S=solve(3.3/1.03==0.6*(1+y/x), 12*1.03==0.6*(1+(y+100)/x)); x=eval(S.x), y=eval(S.y)}$$

mode, that is adjustable electronically¹⁴ from 0 to 100 k Ω in 256 increments, providing a maximum peak-to-peak ripple current according to eq. 7 of the datasheet of up to 2.4 mA (at $V_{s1}=12V$). A blue status LED in this quadrant illuminates whenever the V_{s1} voltage regulator is enabled, and an amber status LED illuminates whenever any of the voltage regulators signal a fault condition (see §5.6.8).

5.2.5 Vin->Vmb switching regulator

The **Vin->Vmb switching regulator** implemented, for generating both **$V_{mb} = 5.1V$ at up to 6A** on the **5V MB** Berets and **$V_{mb} = \min(12V, 0.8 * V_{in})$ at up to 6A** on the **12V MB** Berets, is another **TI TPS56637** again coupled with a **5.6 μH inductor**, and a **68 μF output capacitor** on the **5V MB** Berets, or a **47 μF output capacitor** on the **12V MB** Berets. In both cases, the TPS56637 converter is again wired as suggested in figure 17 of its datasheet, with EN tied to V_{in} and PG to V_{reg_FAULT} , using values from its table 4 for 5V or 12V output as appropriate.

The V_{in} -> V_{s1} and V_{in} -> V_{mb} switching regulators, as well as the motor drivers (see §5.3), are placed close to each other, allowing them to share the same 100 μF **bulk capacitor** near their respective inputs.

As mentioned previously, on the **12V MB** Beret, the 96B motherboard, if one is attached, down-regulates from the $V_{mb}=12V$ line (and, passes back via the low-speed header) **5V at up to 1A**; the 12V MB Beret makes use of this 5V line directly, and thus does not itself have a 5V regulator.

5.2.6 Vin->3.3V switching regulator

The **Vin->3.3V switching regulator** implemented on the **Green** and **Blue** Berets, for generating **3.3V at up to 3A**, is another **TI TPS56637** with a **2.2 μH inductor**, and a **33 μF output capacitor**. The TPS56637 converter is again wired as suggested in figure 17 of its datasheet, with EN tied to V_{in} and PG to V_{reg_FAULT} . The V_{in} -> V_{s1} and V_{in} ->3.3V switching regulators are placed close to each other, allowing them to share the same 18 μF **bulk capacitor** near their respective inputs.

5.2.7 Vmb->3.3V switching regulator

The **Vmb->3.3V switching regulator** implemented on the **5V MB** Berets, for reducing $V_{mb} = 5.1V$ to **3.3V at up to 3A**, is the fixed voltage **TI TPS6208833**, coupled with a **220 nH inductor** and a **22 μF output capacitor**. This converter is wired as suggested in figure 6 of its datasheet, with EN tied to V_{mb} and PG to V_{reg_FAULT} .

On the other hand, the **Vmb->3.3V switching regulator** implemented on the **12V MB** Beret, for reducing $V_{mb} = 12V$ to **3.3V at up to 3A**, is the **TI TPS62913**, coupled with a **4.7 μH inductor** and a **47 μF output capacitor**. This converter is wired as suggested in figure 8.1 of its datasheet (with $R_{fbt}=15.4$ k Ω , $R_{fb}=4.87$ k Ω , C_{ff} open, and L_f removed), with EN tied to 5V and PG to V_{reg_fault} .

The V_{in} -> V_{mb} and V_{mb} ->3.3V switching regulators are placed close to each other (on both the 5V MB and 12V MB Berets), allowing the output capacitor of the former to serve as the input capacitor of the latter.

¹⁴Specific care is taken in the software driving this digital potentiometer in order to cycle the power to the switching regulator off completely before adjusting the value of the resistance of the digital potentiometer (leveraging a look-up table, which is calibrated on the fly, and a false-position search based on this look-up table) in order to, whenever the value of the digital potentiometer is changed, both (a) reset the (internal) compensation coefficient of the switching regulator IC, and (b) prevent Over Voltage Protection (OVP) from kicking in and triggering a soft restart of the switching regulator IC.

5.2.8 Vs2 power opamp

The **Vs2 power opamp** implemented, for generating a second adjustable voltage **Vs2 = 1.2V to 2.1V at up to ± 400 mA**, is provided by a **TI ALM2402-Q1** high-power opamp, wired as described in Figure 5.10a-b (and surrounding discussion). As opposed to the other regulated voltages mentioned in the above four subsections, Vs2 can source or sink up to 400 mA.

Note that, in addition to the Vs1, 12V, 5V, 3.3V, and Vs2 lines discussed above, DAC1 and DAC2 (see §5.7.1) can also each source (or, sink) 400 mA, and can be set as constant (or, time varying) voltage sources, if needed, in the 0V to 3.3V range.

Note that Vs2, DAC1, and DAC2 are not available on the **Red** Beret and, again, that Vmb and 5V are not available on the **Green** and **Blue** Berets.

5.2.9 Switching default power on various connectors and sensors

The power provided to {E1-2, E6-7} may be switched between {3.3V, 5V, Vs1, off} by a **TI TS5A3359** multiplexer, via the Venc_3.3V and Venc_5V Beret GPIOs (see §5.5).

The nominal 3.3V power provided to {E3-4, E5, USART, USART} may be switched to 5V via backside solder jumpers, as illustrated in Figures 5.1-5.6.

The nominal 3.3V power supplied to the IMU, magnetometer, and barometer may be switched to the coin cell (i.e., Vcoin) via a backside solder jumper, as discussed further in §5.4.1.

The nominal Vs1 power supplied to Signal Header A may be changed to Vin by reorienting the 12A shunt connector. **TODO: provide some photos of how to make these changes.**

5.2.10 Charging and voltage monitoring of Vcoin

When in run mode, the STM monitors the voltage, Vcoin, of the VL-1220/FCN coin cell (if installed) using the STM's internal ADC1_IN17 channel, and (if necessary) recharges this cell, which generally operates in the 2.6V to 3.05V range, using the 3.3V power bus. If it is ever found that $2.6\text{V} < V_{\text{coin}} < 2.8\text{V}$, software on the STM selects VBRS=1 in the PWR_CR4 register on the STM (see the STM datasheet), thereby charging the coin cell (until Vcoin = 3.0V, after which charging is turned off) through a 1.5 k Ω resistor internal to the STM, while limiting the charging current applied to

$$(3.3\text{V} - 2.6\text{V})/1.5\text{ k}\Omega \approx 0.47\text{ mA}$$

or less, as per the VL-1220 specification on page 57 of Panasonic's **Lithium Handbook**, which requires this charge current to be 0.5 mA or less. If it is ever found that $2.0\text{V} < V_{\text{coin}} < 2.6\text{V}$, software on the STM selects VBRS=0 in the PWR_CR4 register on the STM, thereby charging the coin cell (until Vcoin = 2.6V, after which VBRS is set to 1) through a 5 k Ω resistor internal to the STM, while limiting the charging current applied to

$$(3.3\text{V} - 2.0\text{V})/5\text{ k}\Omega \approx 0.26\text{ mA}.$$

or less. If ever the charge of the VL-1220 coin cell is found to be less than 2.0V, the battery is flagged as either dead or not installed, and no charging is attempted.

	DRVa						DRVb (24 HB Berets only)					
Outputs on DRVs	6,4	9,11	3,10	5,1	7,12	8,2	3,10	11,9	4,6	8,2	12,7	1,5
JST connector on Beret	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12

Table 5.3: Connections between DRVs and the motor JSTs M1 through M12.

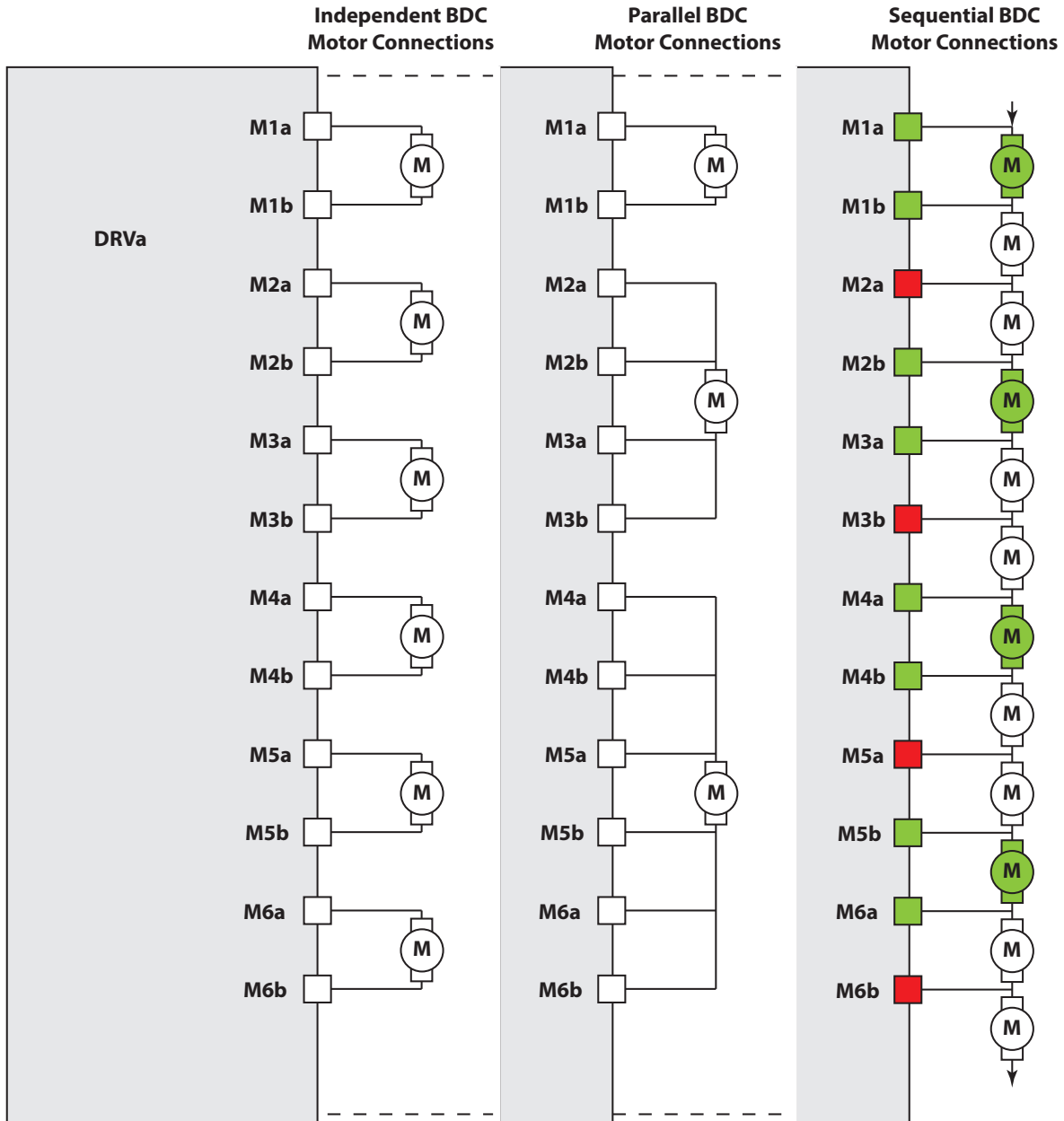


Figure 5.7: Illustration of the independent, parallel, and sequential modes of operation of the 12 outputs of DRVa. For convenience when operating in independent mode, these outputs (reordered and renamed as described in Table 5.3) are arranged on the Beret as six pairs of outputs on each DRV8912-Q1 (M1-M6 on DRVa and, if present, M7-M12 on DRVb).

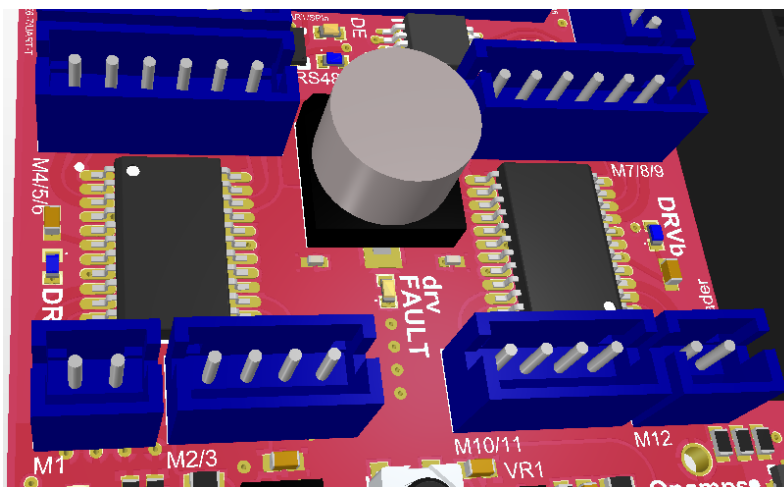


Figure 5.8: Closeup of the connections to the two DRV8912-Q1 motor drivers on a **Raspberry** Beret.

5.3 Control of brushed DC motors & steppers with the DRV8912-Q1

A central feature of the Berets is the [TI DRV8912-Q1](#) brushed motor driver(s) (hereafter, DRVa and/or DRVb) about which they are built¹⁵ (see Figure 5.8). This powerful subsystem draws a maximum of **6A per DRV**, or **1A max per channel**. Each of these remarkably versatile motor drivers incorporates **12 half bridges**, which may be configured in any of a myriad of ways, and **4 synchronized internal PWM generators** (operating at about **80 Hz**, **100 Hz**, **200 Hz**, or **2000 Hz**, selectable in software) to drive them. The outputs of the DRVs are broken out on the Beret one pair at a time, labelled M1 through M6 on DRVa (and, if present, M7 through M12 on DRVb). To simplify layout, the Beret reorders the outputs of the DRVs, as shown in Table 5.3, with a single combined drv_SLEEP channel and a single combined drv_FAULT channel tied to the GPIO expander (see Table 5.6). A blue status LED near the DRVs illuminate whenever they are enabled, and an amber status LED illuminates when the DRV signals a fault (see §5.6.8). Note that the outputs of the DRVs may also be used to drive high-power PWM-driven circuits like LEDs (again, max 6A total per DRV, or max 1A per output).

Note that the Beret’s SPIdrv channel (that is, the STM’s SPI4 channel), operating with a single SS line in daisy-chain mode, gives the STM a dedicated high-speed connection to DRVa and DRVb.

The user might at first just plug into the M1 – M12 discrete JSTs directly, and use the half-bridge outputs one pair at a time, in **independent mode**, for bidirectional control on each DRV of up to 4 “independent” motors running at up to 1A each, and up to 2 “slave” motors operating in brake, coast, full forward, or full reverse, or duplicating the PWM frequency and duty cycle of one of the “independent” motors. If 6 motors are wired to one DRV at the same time, which 4 are chosen to be the “independent” motors, and which 2 as the “slave” motors, may be redefined on the DRV whenever necessary.

Notably, by ganging the DRV outputs together in **parallel mode**, and/or stringing the DRV outputs along in **sequential mode** (see Figure 5.7), a wide array of different motor configurations also becomes possible, as discussed further in the following two subsections. Note, of course, that the independent, parallel, and/or sequential modes can actually be combined on each of the DRV(s), for a wide variety of possible configurations for developing **complex yet practical systems**.

¹⁵The DRV8912-Q1 drivers described here provide a compact general-purpose solution for the control of a wide range of small (up to 28V and 6A) brushed-DC (BDC) motors (in forward, reverse, coasting, and braking operational modes, and alternating between two such modes with PWM) and stepper motors. The other main type of motors used in mechatronic systems, **brushless-DC (BLDC) motors**, require more carefully tuned electronic coordination, with **BLDC motor drivers** fairly closely matched to the BLDC motors to be used. BLDC motor drivers are therefore located on Beret Shields (see §5.8) when using the Beret ecosystem.

5.3.1 Parallel mode

Parallel mode gangs the outputs of 2 to 6 pairs of half bridges on each DRV together, using external wire harnesses, to create full H-bridges for powering higher-current motors, thus driving (on each DRV):

- 1 motor at 1A, 1 motor at 2A, and 1 motor at 3A (as shown in Figure 5.7),
- 1 motor at 6A,
- 2 motors at 3A, etc.

Warnings: on the **24 HB** Berets, DRVa & DRVb operate independently, and thus asynchronously. Thus:

- a. any individual motor may be wired to DRVa or DRVb, but never to both;
- b. parallel connections must join together the same number of DRV outputs on the left and right sides of any motor using a custom wire harness, using a [sufficient wire gauge](#) to handle the resulting current; and
- c. parallel connections must synchronize, via appropriate programming, the wires that are ganged together (e.g., in the configuration shown in Figure 5.7, {M2a, M2b} and {M3a, M3b} must be synchronized to form the left and right sides of the 2A motor, and {M4a, M4b, M5a} and {M5b, M6a, M6b} must be synchronized to form the left and right sides of the 3A motor, see, e.g., §8.3.1.1.3 and §8.3.1.1.4 of the [DRV8912-Q1 datasheet](#)).

5.3.2 Sequential mode

Sequential mode, on the other hand, strings the output of one motor together with the input of the next, in a serial fashion, for bidirection control of a remarkable total of **up to 12 motors at up to 1 A on each DRV**, albeit at reduced duty cycles (i.e., not all at once). Note that interleaving half-bridges must to be turned off at any instant for sequential mode to work correctly, and the voltage used must be slightly above the stall torque of a single motor, so that two motors can not be driven by the supply voltage when applied in series. Thus, for example, if 12 motors are hooked up to a single DRV as shown in Figure 5.7, then:

- A. the 1st, 4th, 7th, & 10th motors are driven for a while (with M2a, M3b, M5a, & M6b off), then
- B. the 2nd, 5th, 8th, & 11th motors are driven for a while (with M1a, M2b, M4a, & M5b off), then
- C. the 3rd, 6th, 9th, & 12th motors are driven for a while (with M1b, M3a, M4b, & M6a off);

these three steps then repeat from the beginning. PWM with independently controllable duty cycles may be used on each channel (e.g., at 2000 Hz) to run these motors (4 at a time on each DRV) at partial power and in either direction, while the cycling between these three steps happens much more slowly (e.g., at 10 to 100 Hz). If, periodically, 4 to 8 of the motors hooked to each DRV are not used (e.g., in sequential assembly-line operations), then these motors may be grouped together as appropriate, and the corresponding step(s) in the above-described cycle can be skipped, improving the smoothness in the driving of the remaining motors (by reducing the time that they spend in a step that turns them off).

5.4 IMU, magnetometer, and barometer

All six Berets includes three built-in environmental sensors:

- a [TDK ICM-42688-P](#) 6-axis IMU (3-axis accel + 3-axis gyro + thermometer), connected via SPI,
- an [ST LIS3MDLTR](#) 3-axis magnetometer, connected via I2C, and
- an [ST LPS22HB](#) barometer + thermometer, connected via I2C.

The dedicated SPIimu connection between the STM and the IMU facilitates, if needed, remarkably fast update rates (up to 32 kHz) for IMU data, suitable for problems in which high-frequency mechanical vibrations need to be characterized. Data from the magnetometer and barometer is usually low-pass filtered (on the sensors themselves) to reduce noise. The magnetometer is generally operated at 80 Hz or less, and the barometer at 75 Hz or less; for such signals, communication over the shared I2Ca bus is thus sufficient.

Note that the extensive bus connectivity (UART, I2C, SPI, USB, CAN) provided by the Beret allows, of course, many additional sensors to be attached easily, as necessary for the user's particular application.

If the center of the top surface of each Beret is taken as the origin, then the centers of the mounting holes and the IMU coordinate system are situated as defined in Table 5.1. Note also that, on each Beret, the magnetometer is located in a corner of the logic quadrant that is as distant as possible from the board's power electronics, thus minimizing the contamination of its readings by stray EM fields.

As depicted by the axes printed next to the STM, the native (x, y, z) coordinates of both the IMU and the magnetometer on the Beret are (in the reference orientation depicted in Figs 5.1-5.6, where "up" is towards the viewer when looking at the top side of the board) aligned with the (East, North, Up) [a.k.a. ENU] directions, respectively. Positive rotations are given by right-hand rotations about each of these axes. For example, as in the ISO 8855 automotive standard, if the body-fixed (x, y, z) axes are taken as vectors out the (front, left, top) of a vehicle, respectively, then positive [a.k.a. 3-2-1 Tait-Bryan] rotations about the (z, y, x) coordinates may be referred to as (yaw, pitch, roll), and denoted (α, β, γ) , respectively, where each of these rotations being positive is given by, respectively, (nose to the left, nose down, right side down).

Other coordinate conventions are easily derived from the native (x, y, z) coordinates used by the Beret. For example, taking $(\hat{x}, \hat{y}, \hat{z})$ as (North, East, Down) [a.k.a. NED] on the board (again, in the reference orientation depicted in Figs 5.1-5.6), with positive (right-hand) rotations about each denoted $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$, it follows that $(\hat{x}, \hat{y}, \hat{z}) = (y, x, -z)$ and $(\hat{\alpha}, \hat{\beta}, \hat{\gamma}) = (\beta, \alpha, -\gamma)$; linear and angular velocities and accelerations in these two different coordinate conventions are related similarly. For example, as implemented broadly in the aerospace industry, as well as by the SAE J670 and J1594 automotive standards, if the body-fixed $(\hat{x}, \hat{y}, \hat{z})$ axes are taken as vectors out the (front, right, bottom) of a vehicle, respectively, then positive rotations $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$ about the $(\hat{z}, \hat{y}, \hat{x})$ coordinates may again be referred to as (yaw, pitch, roll), where each of these rotations being positive is now given by, respectively, (nose to the right, nose up, right side down).

To estimate the time evolution the 6DOF configuration [position plus orientation] of the system to which the Beret is attached, in addition to the 6DOF rate of change of this configuration [together referred to as the 12DOF state of the system], one must integrate the raw linear acceleration and angular velocity data from the IMU, and (optionally, on a slower time scale) fuse this information with the absolute position and orientation measurements that may be obtained from the magnetometer, barometer, and/or attached GPS/GNSS unit, as well as the linear and angular velocity measurements that may be obtained from laser rangefinders, optical flow processing, etc. This complex task, discussion of which is beyond the scope of the present document, must be solved on the STM (not on the IMU itself).

However, the IMU does features a flexible set of built-in programmable digital filters, and includes a proprietary on-chip motion processing engine designed for gesture recognition and activity classification, and can also function effectively as a pedometer (see also §5.4.1).

Note that not all IMUs, magnetometers, and barometers are created equal. Far from it, in fact (buyer beware!). The specs on the sensors selected for the Beret are, as of 2021, best in class:

- the accelerometer noise is about $70 \mu\text{g}/\sqrt{\text{Hz}}$, and the gyro noise about $2.8 \text{ mdps}/\sqrt{\text{Hz}}$,
- the sensitivity of the magnetometer is about $\pm 0.4 \text{ mT}$, and
- the relative accuracy of the barometer is about $\pm 1 \text{ Pa}$ (i.e., $\pm 8.8 \text{ cm}$ change in altitude at sea level¹⁶).

See the corresponding datasheets for further such characterizations, and compare such specs carefully when selecting IMUs, magnetometers, and barometers for your own board designs. Note in particular that many popular “9-axis” IMUs, which (conveniently) incorporate a 3-axis magnetometer as well as a motion processing engine to automatically estimate the system state (albeit, without incorporating inputs from auxiliary sensors such as barometers, GPS/GNSS units, laser rangefinders, etc), compromise on sensor sensitivity in order to fit more functionality onto a single IC; we have thus avoided using such a 9-axis IMU in the Berets.

Note also that the STM and IMU are slaved to the same external 32.7680 kHz MEMS oscillator (see §5.5.1) on the Berets, which keeps their clocks accurately in sync.

5.4.1 Data-ready and sensor-driven interrupts

The [IMU](#), [magnetometer](#), and [barometer](#) used on all six Berets are each capable of taking measurements at a pre-specified rate¹⁷, and alerting the STM (via the `imu_INT2_DRDY`, `mag_DRDY`, and `bar_INT_DRDY` channels, as shown in Tables 5.5-5.6) as soon as there is fresh `data_ready` (DRDY) to be read from the corresponding sensor, so that this sensed data may subsequently be used, with minimum latency, in feedback algorithms.

The IMU, magnetometer, and barometer are also capable of running in the background (without constant monitoring by the STM), and issuing interrupts when a variety of environmental conditions are detected. The IMU can be programmed to issue an interrupt to the STM, over the `imu_INT1_DRDY` and/or `imu_INT2` channels, when any of the following events are detected:

- a step or tap,
- a tilt beyond 35° for more than a certain (programmable) period of time,
- a raise-to-wake or lower-to-sleep gesture, or
- when net accelerations exceed a certain (programmable) magnitude.

Similarly, the magnetometer can be programmed to issue an interrupt, over the `mag_INT` channel, when

- the magnetic field exceeds a (programmable) magnitude in the x, y, or z direction (selectable),

and the barometer can be programmed to issue an interrupt, over the `bar_INT_DRDY` channel, when

- the atmospheric pressure attains a (programmable) maximum or minimum threshold.

The `{imu_INT1_DRDY, mag_INT, bar_INT_DRDY}` channels may also be made connected (via backside solder jumpers on the Beret that are initially open) to the MB (see §5.5 for details).

As also noted in §5.2.9, the nominal power supplied to the IMU, magnetometer, and barometer is 3.3V, but this may be switched via a backside solder jumper to the coin cell (i.e., `Vcoin`). Combining this feature with the programmable interrupts discussed above, any of the above noted events can be programmed to generate an interrupt that wakes both the Beret and the motherboard from a low power sleep mode, which operates solely on the coin cell.

¹⁶Note that atmospheric pressure [decreases](#) at a rate of about 11.3 Pa per meter increase in altitude at sea level.

¹⁷The IMU operates at data rates of 12.5 Hz to 32 kHz, the barometer operates at data rates of 1 Hz to 75 Hz, and the magnetometer nominally operates at data rates from 0.625 Hz to 80 Hz, though in fast mode the magnetometer can be operated from 155 Hz to 1 kHz. The IMU and barometer both incorporate convenient digital low-pass filters that may be enabled.

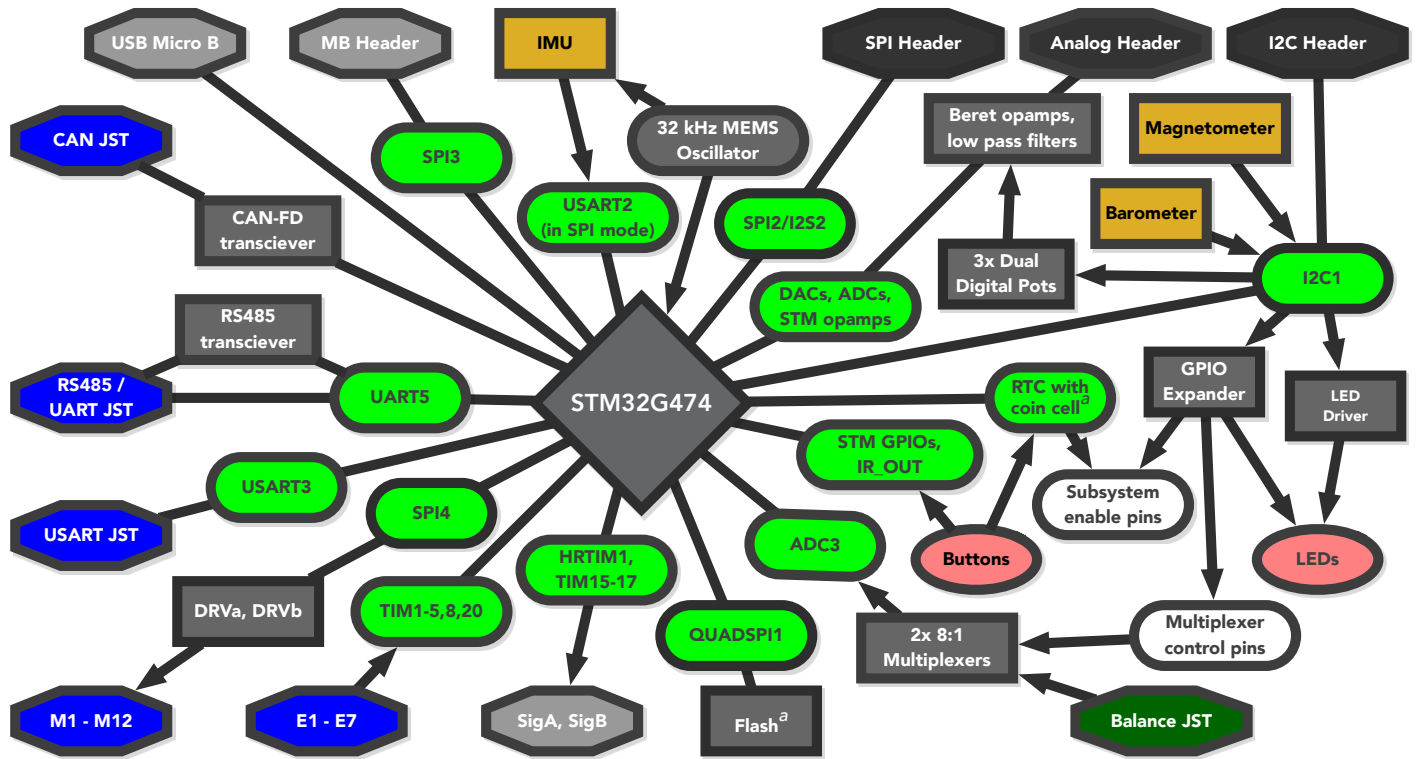


Figure 5.9: Connectivity of the primary STM buses on Berets, with rectangles denoting ICs, brown rectangles denoting environmental sensors, octagons denoting connectors, capsules denoting channels, ellipses denoting buttons & LEDs, and ()^a denoting optional backside components. See Table 5.4 for the correspondence between Beret channel names and the STM32 hardware channel names.

Beret location	Beret channel names	STM32 hardware channel names	§
USART JST	USARTb; UARTb; SPIb	USART3	5.6.5
RS485 JST	RS485; UARta	UART5 (+ RS485 transceiver)	5.6.4
CAN JST	CAN	FDCAN1 (+ CAN transceiver)	5.6.4
E1-2 JST	E1-2; I2Cd	TIM3_CH1/2, TIM8_CH1/2; I2C4	5.6.1
E3-4 JST	E3-4; UARtt	TIM4_CH1/2, TIM2_CH1/2; USART1_TX	5.6.1
E5 JST	E5; UARtr	TIM1_CH1/2; LPUART1_RX	5.6.1
E6-7 JST	E6-7	TIM5_CH1/2, TIM20_CH1/2	5.6.1
Signal Header A	S1-S5; I2Cb, I2Cc	HRTIM_CHA1/A2/E2/E1, TIM16_CH1; I2C2, I2C3	5.6.2
Signal Header B	S6-S10	TIM15_CH1/2, TIM17_CH1, HRTIMCHD1/C2	5.6.2
SPI Header	SPIa; I2S, IR	SPI2, I2S2, IR_OUT	5.6.7
I2C Header	I2Ca	I2C1	5.6.6
Analog Header	DAC1, DAC2	DAC1_OUT1, DAC1_OUT2	5.7.3
MB Header	SPImb	SPI3	5.9
comm to DRVs	SPIdrv	SPI4	5.3
comm to IMU	SPIimu	USART2	5.4
comm to flash	QSPIflash	QSPI1_BK2	5.5.2

Table 5.4: High-level correspondence between various Beeret channel names and the corresponding STM32 hardware channel names. See Table 5.2 for the assignments to each individual pin on the Beret connectors, and Table 5.5 for the assignments to each individual pin on the STM32. Note that, by default, UARtt is transmit-only or half duplex, and UARtr is receive only. See also Figure 1.4.

pin	STM name(s)	Beret name(s)
B5	PB5 SPI3_MOSI (GPIO)	SPImb_MOSI (I2C_G3)
C7	PC11 SPI3_MISO (GPIO)	SPImb_MISO (Vs1_EN)
A5	PB3 SPI3_SCK (GPIO)	SPImb_SCK (Vreg_FAULT)
H6	PE9 ADC3_IN2	Vmon1
K7	PE13 ADC3_IN3	Vmon2

C5	PB4 TIM3_CH1	E1a
F10	PC7 {TIM3_CH2 I2C4_SDA	{E1b I2Cd_SDA
F9	PC6 {TIM8_CH1 I2C4_SCL	{E2a I2Cd_SCL
B10	PA14 {TIM8_CH2 I2C4_SMBA	{E2b I2Cd_SMBA
A4	PB6 {TIM4_CH1 USART1_TX	{E3a UARTt_TX
G9	PD13 TIM4_CH2	E3b
B9	PA15 TIM2_CH1	E4a
A8	PD4 TIM2_CH2	E4b
F2	PC0 {TIM1_CH1 LPUART1_RX	{E5a UARTr_RX
J6	PE11 TIM1_CH2	E5b
J4*	PB2 [†] TIM5_CH1 (GPIO)	E6a_s (Vmon_A0)
G2*	PA1 [†] TIM5_CH2 (GPIO)	E6b_s (Vmon_EN)
C3	PE2 TIM20_CH1 (GPIO)	E7a (CAN_SHDN)
B2	PE3 TIM20_CH2 (GPIO)	E7b (CAN_STB)

C9	PA12 USB_DP	USB_DP
C10	PA11 USB_DM	USB_DM
B8	PD0 FDCAN1_RX	CAN_RX
A9	PD1 FDCAN1_TX	CAN_TX
B7	PD2 UART5_RX	UARTa_RX
A10	PC12 UART5_TX	UARTa_TX
H7	PE15 USART3_RX	USARTb_RX
C8	PC10 USART3_TX	USARTb_TX
G7	PD10 USART3_CK	USARTb_CK
D8	PA13 USART3_CTS	USARTb_CTS
J10*	PD12 [†] USART3_RTS_DE	USARTb_RTS_s

E10	PA8 {HRTIM1_CHA1 I2C2_SDA I2S2_MCK	{S1 I2Cb_SDA I2S_MCK
D10	PA9 {HRTIM1_CHA2 I2C2_SCL I2C3_SMBA	{S2 I2Cb_SCL I2Cc_SMBA
C4	PE0 TIM16_CH1	S3
E9	PC9 {HRTIM1_CHE2 I2C3_SDA	{S4 I2Cc_SDA
E8	PC8 {HRTIM1_CHE1 I2C3_SCL	{S5 I2Cc_SCL
E3	PF9 TIM15_CH1	S6
E4	PF10 TIM15_CH2	S7
B3	PE1 TIM17_CH1	S8
H9*	PB14 [†] HRTIM1_CHD1	S9_s
J9*	PB13 [†] HRTIM1_CHC2	S10_s

Table 5.5: Pinouts of the STM32G474 (VE or VB) on all six Berets. (*) denotes an STM pin that is not natively tolerant to 5V inputs; those marked (*) are also available at a JST or header on the Beret. The 8 digital i/o channels marked ([†]) are passed through an 8-channel TI TXB0108 level shifter to assure 5V tolerance. The 12 channels marked (GPIO) have different functions on the **Green** and **Blue** Berets, as indicated. (Note: table continues on next page.)

pin	STM name(s)	Beret name(s)
K1*	PA7 OPAMP1_VINP	ADC1
H3*	PA3 OPAMP1_VINM	Vref
H1	PA2 OPAMP1_VOUT	lpf1a
F8*	PD14 OPAMP2_VINP	lpf1c
J3*	PC5 OPAMP2_VINM	Vref
H4*	PB0 OPAMP3_VINP	ADC2
J8*	PB10 OPAMP3_VINM	Vref
K3*	PB1 OPAMP3_VOUT	lpf2a
H10*	PD11 OPAMP4_VINP	lpf2c
K10*	PD8 OPAMP4_VINM	Vref
H2*	PA4 DAC1_OUT1	DAC1
J1*	PA5 DAC1_OUT2	DAC2
B4	PB7 I2C1_SDA	I2Ca_SDA
A3	PB8 {I2C1_SCL BOOT0}	{I2Ca_SCL BOOT0}
A2	PB9 IR_OUT	IR_OUT
K9*	PB15† {SPI2_MOSI I2S2_SD}	{SPIa_MOSI_s I2S_SD_s}
D9	PA10 SPI2_MISO	SPIa_MISO
E2	PF1 {SPI2_SCK I2S2_CK}	{SPIa_SCK I2S_SCK}
E1	PF0 {SPI2_NSS I2S2_WS}	{SPIa_SS I2S_WS}
A7	PD5 USART2_TX	SPIimu_MOSI
A6	PD6 USART2_RX	SPIimu_MISO
B6	PD7 USART2_CK	SPIimu_SCK
G4*	PA0 USART2_CTS	SPIimu_SS
J7	PE14 SPI4_MOSI (GPIO)	SPIdrv_MOSI (pause)
B1	PE5 SPI4_MISO (GPIO)	SPIdrv_MISO (mode)
G6	PE12 SPI4_SCK (GPIO)	SPIdrv_SCK (RS485_RE)
A1	PE4 SPI4_NSS (GPIO)	SPIdrv_SS (RS485_DE)
C2	PE6 RTC_TAMP3	reset
F3	PG10 NRST	reset
D4	PC13 RTC_OUT1	power
D3	VBAT VBAT	Vcoin
C1	PC14 OSC32_IN	OSC32
F4*	PC1 QSPI1_BK2_IO0	QSPIflash_IO0
F1	PC2 QSPI1_BK2_IO1	QSPIflash_IO1
G1*	PC3 QSPI1_BK2_IO2	QSPIflash_IO2
K2	PC4 QSPI1_BK2_IO3	QSPIflash_IO3
K6	PE10 QSPI1_CLK	QSPIflash_SCK
C6	PD3 QSPI1_BK2_NCS	QSPIflash_SS
J2*	PB11 GPIO	gpio_INT
H8*	PA6 GPIO	mag_INT
K8*	PB12 GPIO (GPIO)	imu_INT2 (amp_OTF_SLEEP)
G5*	PE7† GPIO	bar_INT_DRDY
G8*	PD9† GPIO	imu_INT1_DRDY
D1	PC15 GPIO (GPIO)	SPImb_SS (RS485_SEL)
G3	PF2 GPIO	USART_G5
G10	PD15 GPIO	I2C_G2
H5	PE8 GPIO	SPI_G4

Table 5.5: Continued from previous page.

GPIO connections on the STM					Connections on the GPIO Expander				
pin	i/o	channel name	notes	§	pin	i/o	channel name	notes	§
G5 [†]	i	bar_INT_DRDY	OD, AL, PU	5.4	P0_4	o	Vs1_EN	PP, AH, B/G	5.2.4
G8 [†]	i	imu_INT1_DRDY	OD, AL, PU	5.4	P0_5	i	Vreg_FAULT	OD, AL, PU, A/3	5.2.4
K8*	i	imu_INT2	OD, AL, PU	5.4	P1_3	o	drva_SLEEP	PP, AL, B/G	5.3
H8*	i	mag_INT	PP, AL	5.4	P1_4	o	drvb_SLEEP	PP, AL, B/G	5.3
J2*	i	gpio_INT	OD, AL, PU	5.5	P1_5	i	drv_FAULT	OD, AL, PU, A/3	5.3
G3	i/o	USART_G5	configurable	5.6.5	P1_2	i	mag_DRDY	PP, AL	5.4
G10	i/o	I2C_G2	configurable	5.6.6	P1_6	o	Venc_3.3V	PP, AH	5.6.1
H5	i/o	SPI_G4	configurable	5.6.7	P1_7	o	Venc_5V	PP, AH	5.6.1
D1	i	SPImb_SS	PP, AL	5.9	P2_0	o	CAN_SHDN	PP, AH	5.6.4
					P2_1	o	CAN_STB	PP, AH, B/3	5.6.4
					P2_2	o	RS485_SEL	PP, AH	5.6.4
					P2_3	o	RS485_RE	PP, AL, B/3	5.6.4
					P2_4	o	RS485_DE	PP, AH, A/G	5.6.4
					P0_3	i/o	I2C_G3	OD, AL, PU	5.6.6
					P2_5	i	pause	OD, AL, PU	5.6.8
					P2_6	i	mode	OD, AL, PU	5.6.8
					P2_7	i/o	amp_OTF_SLEEP	OD, AL, A/G	5.7
					P0_6	o	Vmon_EN	PP, AH	5.7.4
					P0_7	o	Vmon_A0	PP	5.7.4
					P1_0	o	Vmon_A1	PP	5.7.4
					P1_1	o	Vmon_A2	PP	5.7.4
					P0_0	i/o	SPImb_G0	configurable	5.9
					P0_1	i/o	SPImb_G1	configurable	5.9
					P0_2	i/o	SPImb_G2	configurable	5.9

Connections on the LED Driver		
pin	LED name	§
P0	gauge_G1	5.6.8
P1	gauge_G2	5.6.8
P2	gauge_G3	5.6.8
P3	gauge_R	5.6.8
P4	LED_R	5.6.8
P5	LED_Y	5.6.8
P6	LED_G	5.6.8

Table 5.6: GPIOs on the **Raspberry**, **Black**, and **White** Berets located on (top/left) the STM and (right) the GPIO expander; note that the **Green** and **Blue** Berets, with fewer GPIOs, do not use a GPIO expander at all (they have all GPIOs moved to the STM, as indicated in Table 5.5). (bottom/left) The LED Driver connections on all six Berets. Inputs with a Pull Up resistor are labelled PU; internal pull up resistors on the STM are 40 k Ω , those on the GPIO expander are 55 k Ω . Outputs: OD = Open Drain, PP = Push/Pull. Logic: AL = Active Low, AH = Active High. {B/G, B/3, A/G, A/3} indicate the channel is associated with a Blue or Amber status LED, connected to Ground or 3.3V; all 9 LEDs tied to the LED driver connect to 3.3V. **TODO: update pin #s of GPIOs.**

5.5 STM32G474 microcontroller features, pinouts, and GPIOs

All six Berets are controlled by a 100-pin [STM32G474VE](#) (or, VB) microcontroller¹⁸, which includes a 170 MHz [ARM Cortex-M4 core](#) with [512 KB flash](#) and 128 KB SRAM, and several useful features for efficient implementation of difference equations for feedback control of electromechanical systems, such as STM's [Adaptive real-time \(ART\) memory accelerator](#) and a [Filter Math Accelerator \(FMAC\)](#), which implements low-level circular buffers for offloading the computation of FIR and IIR filters from the main ARM core (see §1.5.3.2).

The STM32G474 includes an extensive set of dedicated hardware subsystems for driving busses and peripherals without putting a computational load on the ARM core itself. As shown in Figure 5.9, the Beret makes considerable use of these hardware subsystems on the STM. In fact, after connecting up these many subsystems and connectors on the **Raspberry**, **Black**, and **White** Berets, only 9 pins on the STM were left over to use as dedicated [STM GPIOs](#); 24 additional GPIOs are thus obtained on these Berets using a [NXP PCAL6524HEHP](#) GPIO expander¹⁹. As stated in Note A of Table 5.2, any STM i/o channel available on a Beret JST that is not

¹⁸Key references for the STM32G474, which users of the Beret will need frequently, are its [datasheet](#) and [reference manual](#). Also available from ST is a comprehensive set of [training courses](#) (both videos and PDFs) specifically designed for the STM32G4 series.

¹⁹A change of state of an input to the GPIO expander is indicated by the gpio_INT channel on the STM (see Tables 5.5-5.6).

otherwise being used for its primary purpose can be reconfigured as an auxiliary STM GPIO.

The pinouts and GPIO channels²⁰ of the STM32G474VE/B (TFBGA100 variants of the STM32G474), the [LED Driver](#), and the [GPIO Expander](#) are specified in Tables 5.5-5.6. The {VSS, VSSA} pins {D2, D6, E5, E6, E7, F6, K4} are wired to **GND**, and the {VDD, VDDA, VREF+} pins {D5, D7, F5, F7, J5, K5} are wired to **3.3V**.

5.5.1 Real-time clock (RTC), and scheduled/commanded wakeup from VBAT mode

All six Berets include a modern **MEMS oscillator** (specifically, an [SiTime SIT1532AC-J5-DCC-32.768E](#), operating at 32.7680 kHz), rather than a quartz crystal resonator with load caps, to generate OSC32 to drive both the **STM real-time clock** (RTC) as well as the [ICM-42688-P](#) IMU (see §5.4).

When in run mode, pressing the reset button on the Beret drives the **STM NRST channel** on the PG10 pin low, resetting the STM. When in VBAT mode (low-power sleep, powering the STM solely by the VL-1220 coin cell, with the main power to the Beret turned off at the power MOSFET), pressing the reset button drives the RTC_TAMP3 channel on the PE6 pin²¹ low. The STM RTC module is used to wake the board from VBAT mode (either if RTC_TAMP3 is driven low, or if scheduled, or if a wake-up signal is received over the LPUART channel), leveraging the STM RTC_OUT1 channel on the PC13 pin, which is wired to the enable pin on the Beret's ideal diode controller, which in turn is wired to the main power MOSFET (see §5.2.3); the 5V and 3.3V Vregs, as well as the STM, quickly power back up automatically when this power MOSFET is turned back on. **[Check!]**

5.5.2 Customization with Quad-SPI Flash

The **full size** Berets are also preconfigured with a pinout of the **STM QuadSPI module** QSPI1_BK2 for easy addition of a standard-size (6 mm x 8 mm, 8-WSON) fast, low-cost, high-capacity **4 MB to 512 MB flash IC** for extending the (**NAND or NOR**) flash memory capacity of the system, for those applications that need it.

²⁰A GPIO may be either active high or active low, and either push-pull or open drain. **Active high** means the channel is “active” or “on” in the Logical 1 state (3.3V on the Beret), and “inactive” or “off” in the Logical 0 state (GND); **active low** means the opposite. **Push-Pull** means the connected device either drives the channel low (through an NPN BJT or n-channel MOSFET to GND), or drives the channel high (through a PNP BJT or p-channel MOSFET to 3.3V). **Open Drain** (a.k.a. **Open Collector**), in contrast, means the connected device drives or “asserts” the channel to the low state (through an NPN BJT or n-channel MOSFET to GND), but the channel is left floating by the connected device instead of driving it to the high state, thus requiring a **pull-up resistor** (connecting the channel to 3.3V via a resistor, often internal to the MCU) to achieve a definitive boolean state. [A **pull-down resistor** (connecting to GND) is occasionally needed in certain analogous situations, in which a device (like a button) might assert a channel to the high state, but otherwise leaves it floating.] An advantage of the Open Drain setting is that several devices can be tied to a single GPIO channel on the MCU; this channel then functions as a **wired OR** device for active low logic (or, as a **wired AND** device for active high logic).

²¹The PE6 pin and the PG10 pin, operated as inputs, are wired together; when in run mode, PE6 is ignored, and when in VBAT mode, PG10 is ignored.

5.6 Connectivity and i/o

5.6.1 Quadrature encoder counters and connectors

As indicated in Figure 5.9 and Tables 5.4-5.5, the encoder channels E5, E2, and E7 on the Berets connect to pairs of outputs on the [STM advanced control timers](#) TIM1, TIM8, and TIM20, and the encoder channels E4, E1, E3, and E6 on the Beret connect to pairs of outputs on the [STM general purpose timers](#) TIM2, TIM3, TIM4, and TIM5. All 7 of these hardware timers are nominally operated in the quadrature encoder counting mode, with up/down counting, without loading the main ARM core. The encoder connectors on the Beret also include connections to power. As discussed in §5.2.9, the (3.3V or 5V) power provided to the E3-4 and E5 connectors may be changed via backside solder jumpers, and the (3.3V, 5V, Vs1, or off) power provided to the E1-2 and E6-7 connectors is selected via a [multiplexer](#) using the Venc_3.3V and Venc_5V Beret GPIOs.

Further, the E1-2, E3-4, E5, and E6-7 connectors are wired, primarily, for standard two-channel (AB) quadrature encoders. However, if needed, the STM32 ETR pins for 5 of these 7 encoder channels are readily available on other various other available pins the Berets, specifically:

- PE7 is connected to I2C_G2 on the I2C Header, and can act as ETR for TIM1 (E5).
- PB12 is connected to SPI_G4 on the SPI Header, and can act as ETR for TIM5 (E6).
- PA8 is connected to S1 on Signal Header A, and can act as ETR for TIM4 (E3).
- PE0 is connected to I2C_G2 on the I2C header, and can act as ETR for TIM20 (E7).
- PD2 is connected to UARTa_RX on the RS485/UART JST, and can act as ETR for TIM3 (E1).

This facilitates the use of up to five three-channel (ABZ) encoders without requiring a Beret Shield.

As discussed in section 3.24 of the [STM32G474 datasheet](#), these timers are quite powerful, and thus the convenient E1-2, E3-4, E5, and E6-7 connectors may be used for a host of alternative functions, such as the generation of PWMs for driving up to 14 additional servos and ESCs (again, without loading the ARM core itself), which may be useful if the 10 discrete servo/ESC connectors discussed in §5.6.2 provide insufficient connectivity for a given application. Other functionality also available on the encoder connectors includes:

- I2C4 (including SMBA), available on E1-2 (see §5.6.6),
- a half-duplex UART channel (USART1_TX), available on E3-4 (see §5.6.5), and
- the LPUART1_RX channel (in receive only mode, as used by DSM receivers), available on E5 (see §5.6.5).

Warning: no specific Electrostatic Discharge (ESD) protection is provided on the encoder connectors.

5.6.2 PWM-based servo and ESC controllers and the Signal Headers

As indicated in Figure 5.9 and Table 5.5, signals {S1, S2, S4, S5, S9, S10} connect to the [STM high-resolution timer](#) HRTIM1, and signals {S6, S7, S3, S8} connect to the [STM general purpose timers](#) TIM15, TIM16, and TIM17. Signal Headers A and B expose signals S1 through S10, along with power and GND, in an industry-standard manner (see Note H of Table 5.2). Vs1 is provided on Signal Header B, and the user may select between Vs1 and Vin on Signal Header A (see §5.2.9). A blue status LED near the Vs1 voltage regulator illuminates whenever power (Vs1) to the signal headers is enabled (see §5.6.8).

Again, these timers are quite powerful, and thus signals S1 through S10 on Signal Headers A and B may be used for a variety of alternative functions, such as adding a few unidirectional encoder counters (using TIM15, TIM16, and TIM17, with up-counting only – and, again, without loading the ARM core itself), which may be useful if the 7 quadrature encoder counters (with up/down counting) discussed in §5.6.1 are insufficient for a given application. The following alternative functionality is also available on the signal headers:

- I2C2 is available on {S1,S2} (see §5.6.6), and
- I2C3 is available on {S4,S5}, with SMBA available on S2 (see §5.6.6).

Note also that signals S1 through S5, and thus the high-resolution timer to which they connect, are readily available for creative use on Extended Beret Shields, as discussed in §5.8.

ESD protection is provided on the S1 - S10 signal lines by two [TI TPD6E004s](#).

5.6.3 Encoding for IR communication

The output of the [STM IR communication module](#) may be routed (as IR_OUT) to pin 9 of the SPI Header. When the IR communication module is enabled, S3 and S8 on Signal Headers A and B are converted into GPIOs.

5.6.4 CAN FD and RS485 transceivers and connectors

On the [CAN/RS485](#) Berets, the [STM FD-CAN controller](#) of the STM32G4 is paired with a (3.3V) [TI TCAN334G CAN-FD transceiver](#), to generate the differential pair of signals, {CANH, CANL}, required for [CAN](#) communication, supporting the CAN FD (CAN with flexible data-rate) protocol at up to 5 Mbps. These two signals are made available on the CAN JST connector (see Table 5.2). Note that the STB and SHDN pins of the TCAN334G are wired to the Beret CAN_STB and CAN_SHDN GPIO channels (see §5.5), to put the transceiver in low-power standby and shutdown states when not in use (see the device datasheet for details).

On the [CAN/RS485](#) Berets, the STM's UART5 channel may be used to drive a (3.3V or 5V selectable) [TI THVD1452 RS485 transceiver](#), with built-in idle bus failsafe, to generate the two differential pairs of signals, {Y,Z,A,B}, required for full-duplex RS485 communication, supporting the RS485 protocol at up to 50 Mbps. This is accomplished by setting the Beret's RS485_SEL GPIO to 1, and using the GPIO channels RS485_RE and RS485_DE to enable its receiver and driver (i.e., transmitter) as necessary. Note that this transceiver presents a 1/8 Unit Load, allowing up to 256 receivers on a single bus.

On the backside of these Berets, the user may install (in the footprints provided) two 0804 resistors and (optionally) a 0804 capacitor for termination of the CAN bus, and a single resistor (on each differential pair) for termination of the (full-duplex) RS485 bus, as discussed in the [TI AN-1057](#) report.

Note that the CAN and RS485 transceivers provide certain levels of ESD protection, as specified in the corresponding device datasheets. For both, if necessary, additional TVS diodes may be placed on the bulkhead of the environmental housing (see §4.4.3), where the rugged field-serviceable connectors (see §4.4.2) attach to the short jumper cables that brings the differential CAN and/or RS485 signal pairs onto the Beret. This physical separation helps to prevent voltage transients, ESD, and noise from propagating onto the Beret itself.

5.6.5 USART, UART, and LPUART modules and connectors

Four independent [STM USART and UART modules](#), and the [STM LPUART module](#), are used on the Beret:

- the STM USART3 module (RX, TX, CK, CTS/NSS, & RTS/DE) is fully broken out on the USART JST,
- the STM UART5 module (RX & TX) is broken out on the RS485 (UARTa) JST if RS485_SEL=0 (see also 5.6.4),
- the STM USART1 module (TX only²², for use in half duplex mode) is available (as E3a) on the E3-4 JST,
- the STM LPUART1 module (RX only²³, for use in receive mode only) is available (as E5a) on the E5 JST, and
- the STM USART2 module (in SPI mode) gives the Beret a dedicated high-speed connection to the IMU.

²²Via a backside solder jumper, the 4th pin on the E3-4 JST can be switched over to STM pin B3, which can be set in software to function as the USART1_RX input. If this is done, E3-4 can then function as a full duplex Recon UART-T connector with two GPIOs. Note that doing this makes the S8 channel on Signal Header B unusable for other purposes.

²³Via a backside solder jumper, the 4th pin on the E5 JST can be switched over to STM pin H8, which can be set in software to function as the LPUART1_TX output. If this is done, E5 can then function as a full duplex Recon UART-R connector. Note that doing this makes the mag_INT channel (to the STM and the MB Header) unusable as a magnetometer interrupt. Note also that, though H8 is not 5V tolerant, it is set as an output when E5 is converted to a UART-R connector in this manner, so this connector can still be connected to 5V TTL UART devices.

The four JSTs mentioned above are useful for connecting (primarily point-to-point, with one device per channel²⁴) a variety of additional sensors and other devices to the Beret, such as GPS/GNSS units. Extra GPIOs, if needed by a particular device, can be picked up from unused pins on other JSTs, or from the I2C and SPI headers (see Note A of Table 5.2). The first three pins (power, GND, signal) of E5 is suitable for direct connection to 2.4Ghz DSMX/DSM2 radio receivers, such as the [Spektrum SPM4648](#) and [OrangeRx R110x](#).

Supported modalities on the STM32's flexible USART channel (available on the Beret's USART JST) include:

- full duplex (RX-to-TX and TX-to-RX) and single-wire half duplex (TX-to-TX) UART modes,
- UART (asynchronous, without CK) and USART (synchronous, with CK) modes,
- multiprocessor communication [e.g., direct from the STM on one Beret to the STM(s) on other Beret(s)],
- SPI master (TX-to-MOSI, RX-to-MISO) and SPI slave (TX-to-MISO, RX-to-MOSI, slave select on NSS) modes,
- Smartcard ISO7816 communication,
- IrDA serial infrared communication, and
- [RS232](#) and [RS485](#) hardware flow control.

Warning: no specific ESD protection is provided on the USART or UART JSTs, or other pins where UART functionality is available, as specified above.

5.6.6 I2C modules and the I2C Header

Four independent [STM I2C modules](#), at comm speeds up to 1 MHz, are broken out and used on the Beret:

- the STM I2C1 module (SDA & SCL) is broken out on the I2C Header,
- the STM I2C2 module (SDA & SCL) is available (as S1 & S2) on Signal Header A,
- the STM I2C3 module (SDA, SCL, & SMBA) is available (as S4, S5, & S2) on Signal Header A, and
- the STM I2C4 module (SDA, SCL, & SMBA) is available on the E1-2 JST.

Note that the I2C1 channel (on which the STM operates as master) also connects on the Beret to:

- three [TI TPL0102-100](#) dual digital pots²⁵:
 - POT1, at I2C address 101 0101b (0x55h), POT2, at I2C address 101 0110b (0x56h) [see §5.7.2], and
 - POT3, at I2C address 101 0111b (0x57h) [see §5.2.4 and §5.7.1],
- the [GPIO expander](#), at I2C address²⁶ 010 0010b (0x44h for write, 0x45h for read) [see §5.5],
- the [LED driver](#), at I2C address 100 0101b (0x8Ah for write, 0x8Bh for read) [see §5.6.8],
- the [barometer](#), at I2C address²⁷ 101 1100b (0xB8h for write, 0xB9h for read) [see §5.4], and
- the [magnetometer](#), at I2C address²⁸ 001 1100b (0x38h for write, 0x39h for read) [see §5.4].

Warning: when connecting other common I2C devices on the I2C Header, care must be taken to not conflict with the eleven underlined I2C hex addresses listed above, which are already used. Due to this existing traffic on the I2C1 channel, when attaching additional I2C devices, it is recommended that the user instead consider the use of the I2C2, I2C3, and I2C4 modules, if the various pins for them are available, as discussed above.

Warning: no specific ESD protection is provided on the I2C Header, or other pins where I2C functionality is available, as specified above.

²⁴Operating the USART3 module (wired to the USART JST) in SPI mode allows it to connect to multiple devices, with one Beret GPIO operating as a dedicated CS for each connected device.

²⁵POT1 is wired with {A2,A1,A0}={1,0,1}, POT2 is wired with {A2,A1,A0}={1,1,0}, POT3 is wired with {A2,A1,A0}={1,1,1}.

²⁶The GPIO expander is wired with {A2,A1,A0}={0,0,1}.

²⁷The barometer is wired with SA0=0.

²⁸The magnetometer is wired with SA1=0.

5.6.7 SPI modules and the SPI Header

Three²⁹ independent **STM SPI modules** are broken out and used on the Berets:

- the STM SPI2/I2S2 module (MOSI, MISO, SCK, NSS in SPI mode, or SD, WS/LRCLK, SCK/BCLK in **I2S mode**³⁰) is broken out on the SPI Header, providing unencumbered SPI (or I2S) functionality to the user,
- the STM SPI3 module gives the Beret a dedicated high-speed connection to the SPI channel on the MB (except, of course, on the **Green** and **Blue** Berets), using (on the STM) GPIOs connected to MB's SS channels (see Table 5.6) for software slave select, and
- the STM SPI4 module, operating with a single SS line in **daisy-chain** mode, gives the Beret a dedicated high-speed connection to DRVa and DRVb (see §5.3).

The USART JST (i.e., USART3, see §5.6.5), operating in SPI mode, provides further unencumbered SPI functionality to the user if needed.

NSS on the SPI header (i.e., SPI2) provides a hardware slave select line when the Beret is used as an SPI slave on this channel. When the Beret is used as the SPI master on this channel, on the other hand, multiple SPI slaves can be attached using STM GPIOs as separate slave select pins, recalling again that extra GPIOs, where needed, can be picked up from any unused pins on the SPI and I2C headers, or over on the JSTs.

Warning: no specific ESD protection is provided on the SPI or MB Headers.

5.6.8 LEDs, Buttons, and Displays

LEDs. A **TI TCA6507** LED driver is used to control the main LED circuits on the Beret, including three user LEDs (**LED_R**, **LED_Y**, **LED_G**, forming a “stoplight” of sorts) and three **bicolor LEDs** (controlled via the channels **gauge_G1**, **gauge_G2**, **gauge_G3**, **gauge_R**³¹), forming a “power gauge” next to the XT30 input, which are programmed to be illuminated (if a battery is detected at the Balance connector) based on the charge of the individual cell operating with the lowest charge as follows:

- Three solid green 85% to 100% charge
- Two solid + one flashing green 70% to 85% charge
- Two solid green 55% to 70% charge
- One solid + one flashing green 40% to 55% charge
- One solid green 25% to 40% charge
- One flashing green 10% to 25% charge
- Three solid red 1% to 10% charge, user should shutdown immediately
- Three (quickly) flashing red Vmin reached, automatic board shutdown imminent.

The voltage levels for these indicator transitions come pre-programmed for LiPo batteries, with $V_{min} = 3.2V$ and $V_{max} = 4.1V$, but may be adjusted further by the user. If no battery connection is detected on the Balance connector³², the center green LED is illuminated simply to indicate that the board is powered up and running normally (most likely from a wall adapter). Note that alternate red/green patterns slowly flashing in the power gauge, using a simple binary representation in the greens, may be used to indicate eight distinct user-programmable error codes.

In addition, there are **blue** and **amber** status LEDs on the Berets, each located in the respective quadrant of Figs 5.1-5.6, and attached directly to the corresponding GPIO channel (see Table 5.6):

²⁹In addition (see §5.6.5), recall that the STM USART2 module, operating in SPI mode, gives the Beret a dedicated high-speed connection to the IMU.

³⁰If the **I2S** MCK channel is needed, it is available on pin S1 of Signal Header A when using Extended Beret Shields (see §5.8).

³¹The three green gauge LEDs are controlled independently, whereas the three red gauge LEDs are controlled by a single channel.

³²**Warning:** this system is designed to be used with a balance connector whenever a battery is being used, and will not actively monitor the state of charge of the battery unless the balance connector is plugged in.

- the blue Vs1 LED illuminates whenever the Vs1 Vreg circuit is enabled,
- the blue DRVa LED illuminates whenever the DRVa motor driver is enabled,
- the blue DRVb LED illuminates whenever the DRVb motor driver is enabled,
- the blue CAN LED illuminates when the CAN transceiver is in run mode,
- the blue RS485_RE LED illuminates when the RS485 receiver is enabled,
- the amber RS485_DE LED illuminates whenever the RS485 driver (transmitter) is enabled,
- the amber Vreg_fault LED illuminates whenever any of the Vreg ICs signals a fault,
- the amber drv_FAULT LED illuminates whenever either of the DRVs signals a fault, and
- the amber opamps LED illuminates when the Beret opamps are enabled and no fault is triggered³³.

The red, yellow, amber, and bicolor LEDs on the Berets, including those in the bicolor LEDs, operate at about 2V and 6 mA, and the blue and green LEDs on the Berets operate at about 2.9V and 5 mA. As the LED circuits are all powered with 3.3V, $(3.3V-2V)/6\text{ mA} \approx 220\ \Omega$ resistor arrays are used to regulate the current to the red, amber, yellow, and bicolor LEDs, and $(3.3V-2.9V)/5\text{ mA} \approx 82\ \Omega$ resistor arrays are used to regulate the current to the blue and green LEDs.

The several red and amber LEDs on the Beret are manufactured using AlInGaP (Aluminium, Indium, Gallium and Phosphorous) technology, and the several green and blue LEDs on the Beret are manufactured using InGaN (Indium, Gallium and Nitrogen) technology. These modern choices provide maximum brightness while drawing the minimum amount of current.

Buttons. Two small user-programmable buttons (with white actuators) are included as inputs, with pull-up resistors on the GPIO expander; pressing these buttons connects these resistors to GND. The buttons can be used for any function in software, but are preassigned the names “pause” and “mode”, which is appropriate for typical use cases. A third button (with a black actuator), named “reset”, is used to reset the STM when in run mode, or to wake the entire board from low-power sleep mode when the board is powered down (see §5.5.1).

Displays. A Beret Shield (see §5.8) implementing COTS I2C 0.96” OLED Display module is planned. *An eInk Beret Shield would also be nice. Size?*

5.6.9 USB Micro-B connector and Device Firmware Upgrades

The STM USB module, available on pins USB_DP and USB_DM, is wired directly to a standard USB Micro B connector on all six Berets. This USB connector may be used for ordinary programming, in addition to performing Device Firmware Upgrades (DFUs) using the STM DeFuse software, as discussed further here, in conjunction with the BOOT0 pin available on pin 7 of the I2C header (see Table 5.2).

The USB connector can power a Beret for the purpose of STM programming only (see §5.2.2); the USB connector does not provide enough current to power motors or a connected MB, which should not be attempted.

ESD protection is provided on the USB data lines by a TI TPD6E004.

³³The amp_OTF_SLEEP i/o channel, when set as an output and driven low by the GPIO expander, puts the ALM2402-Q1 opamps (§5.7.1) and TLV9002S opamps (§5.7.2) into a low-power sleep mode. When set as an input on the GPIO expander, it is left floating by the opamps during normal operation, and is pulled up to logic high by an external 2.5 k Ω resistor to 5V while simultaneously illuminating a corresponding 2V amber opamps LED tied to ground, unless/until a fault is triggered, then it is driven low.

5.7 Analog subsystem

The **Raspberry**, **Black**, **White**, **Green**, and **Blue** Berets include a 0V – 3.3V analog subsystem with:

- two high-power (400 mA) DAC channels,
- (V+, V-, Vo) pinouts for a spare opamp that may be configured (on a [Beret Shield](#)) by the user, and
- two ADC channels with:
 - adjustable gain, x1 to x4096, and
 - adjustable second-order filtering, with tunable cutoff frequency $\omega_c = 2\pi f_c$ and damping $\zeta = 1/(2Q)$.

A amber status LED (see §5.6.8) next to the Analog Header illuminates whenever the analog subsystem, and the opamps that drive it, are enabled by the STM.

5.7.1 Generation of Vs2 = 1.2V to 2.1V, and two high-power DAC outputs

Figure 5.10a illustrates how the reference voltage $V_2 = 1.2V$ to $2.1V$ is generated, with $R_a = R_b = 133\text{ k}\Omega$ and R_v given by half of **POT3**, a TI TPL0102-100 dual digital pot (100 k Ω) operating in voltage divider mode.

Two **TI ALM2402-Q1** dual high-power opamps are included on the full size Berets, each with a pair of **IGBTs** arranged as **class AB amplifiers** with zero **crossover distortion**, negligible voltage offset, and robust current limiting. The [open-drain, active low] amp_OTF_sleep i/o channel, with an external 2.5 k Ω pull-up resistor to 5V (**CHECK for Blue**) as well as a 2V amber diode to GND, is connected to the GPIO expander via a 220 Ω resistor (see Table 5.6). When this channel is set on the GPIO expander as an input, it is used to monitor for an Over Temperature Flag on these opamps; when set as an output and driven low, this channel puts both ALM2402-Q1s into a low-power sleep mode.

Figure 5.10b shows how three of these high-power opamps are used to buffer $V \in \{V_2, \text{DAC1}, \text{DAC2}\}$, thereby generating the buffered outputs $\bar{V} \in \{V_{s2}, \text{DAC1buf}, \text{DAC2buf}\}$ made available on the Analog Header, each of which is capable of sourcing or sinking 400mA. [Note that the 3 terminals of the fourth high-power opamp are provided directly on the Analog Header, as discussed further in §5.7.3.] When set near 1.65V, Vs2 is useful as a bipolar offset for the analog subsystem.

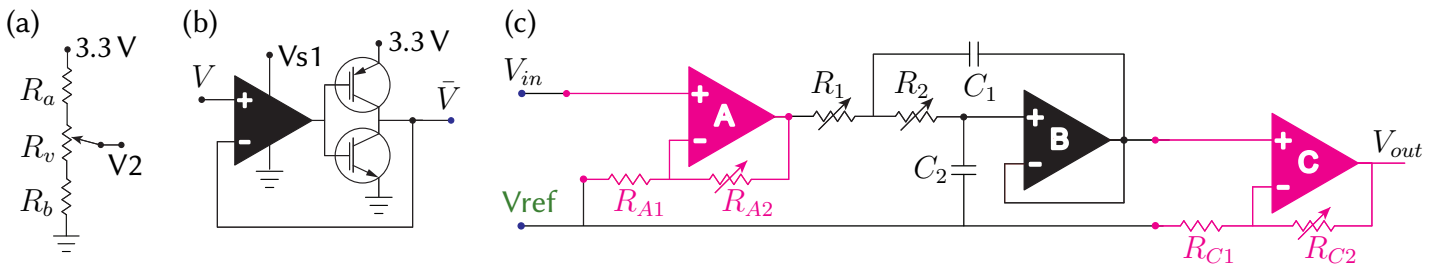


Figure 5.10: Circuits for (a) generation of $V_2 = 1.2\text{ V}$ to 2.1 V adjustable, (b) buffering of $V \in \{V_2, \text{DAC1}, \text{DAC2}\}$ (each buffered output, $\bar{V} \in \{Vs_2, \text{DAC1buf}, \text{DAC2buf}\}$, is capable of sourcing or sinking 400 mA), and (c) amplification and filtering of the $V_{in} \in \{\text{ADC1}, \text{ADC2}\}$ inputs, each of which are compared to the reference voltage $V_{ref} \in \{\text{GND}, Vs_2, \text{ADC3}\}$, with outputs V_{out} routed to ADC2_IN3 and ADC4_IN3 channels internally on the STM. Note that magenta components/traces are within the STM, and black traces/components/signals are on the Beret. See Table 5.2 for how these various input, output, and power pins appear on the Analog Header.

5.7.2 Tunable filtering/gain of two unipolar, bipolar, or differential ADC inputs

Figure 5.10c illustrates the circuit used (leveraging 4 of the internal opamps on the STM, together with a TI TLV9002S dual low-cost opamp) to amplify and filter the $V_{in} \in \{\text{ADC1}, \text{ADC2}\}$ inputs on the Analog Header, before their corresponding amplified/filtered outputs V_{out} are routed to internal ADC units on the STM.

There are three natural choices for selecting the bias voltage V_{ref} about which these ADC inputs are compared and amplified (this choice is made, by the user, by the wiring on the Beret Shield where the analog circuit is developed; for further discussion, see §5.8):

- for **unipolar** analog signals, the user should wire V_{ref} to **GND** (again, on the Beret Shield);
- for **bipolar** analog signals, wire V_{ref} to **Vs2** (as a Bipolar Offset, tunable in the vicinity of 1.65 V);
- for **differential comparison** of analog signals, wire V_{ref} to a third (user-provided) analog signal **ADC3**.

Note that the **ADC1** and **ADC2** inputs on the Analog Header lead directly (without any intervening resistors or capacitors) to opamp input terminals in Figure 5.10c, so these filters perform predictably even for “weak” analog sources with low output impedance. Note further that:

- For $V_{in} = \text{ADC1}$, **A** is OPAMP1, **B** is half of the TLV9002S, **C** is OPAMP2, and $V_{out} = \text{ADC2_IN3}$;
- For $V_{in} = \text{ADC2}$, **A** is OPAMP3, **B** is half of the TLV9002S, **C** is OPAMP4, and $V_{out} = \text{ADC4_IN3}$.

In the (non-inverting) “PGA mode” shown for both opamps **A** and **C**, the internal resistors $\{R_{A1}, R_{A2}, R_{C1}, R_{C2}\}$ can be selected (in software) to achieve amplification ratios of $\times 2$ to $\times 64$; note that $\{R_{A2}, R_{C2}\}$ can also be bypassed, and the corresponding connections to V_{ref} (through $\{R_{A1}, R_{C1}\}$) cut (in software, by selecting “follower mode”, as shown here for opamp B) in order to achieve amplification ratios of $\times 1$ on all three opamps. Thus, the overall low-frequency amplification of this circuit can be varied, in software, from $\times 1$ to $\times 4096$.

The pairs of resistors $\{R_1, R_2\}$ and capacitors $\{C_1, C_2\}$ in Figure 5.10c, looping around opamp B, form a **Sallen-Key second-order low-pass filter** (LPF), leading (see, e.g., [here](#) or [here](#)) to the transfer function

$$\frac{V_{out}(s)}{V_{in}(s) - V_{ref}} = A_A A_C \frac{\omega_c^2}{s^2 + 2\zeta\omega_c s + \omega_c^2}$$

where $A_A = 1 + R_{A2}/R_{A1}$, $A_C = 1 + R_{C2}/R_{C1}$, and the cutoff frequency $\omega_c = 2\pi f_c$ and damping $\zeta = 1/(2Q)$ of the second-order low-pass filter are $\omega_c = 1/\sqrt{R_1 C_1 R_2 C_2}$ rad/s and $\zeta = C_2 (R_1 + R_2) \omega_c / 2$.

A target value of damping to use in such a filter is $\zeta \approx 0.7$. Common capacitor values of $C_1 = 68\text{ nF}$ and $C_2 = 33\text{ nF}$ have been selected for the Beret, together with **POT1** and **POT2**, two TI TPL0102-100 dual digital potentiometers that are adjustable electronically (over I2C) from 0 to $100\text{ k}\Omega$ in 256 increments, for R_1 and R_2 . Setting $R_1 = R_2$ and adjusting both (together) over a range from $100\text{ k}\Omega$ down to $1\text{ k}\Omega$ results in $\zeta \approx 0.7$, and

f_c ranging from 34 Hz to 3400 Hz, as appropriate for many small electromechanical systems; adjusting R_1 and R_2 separately also allows the damping ratio ζ to be tuned³⁴.

Warning: attempting the following is quite difficult, voids any sort of manufacturer's warranty expressed or implied, and should only be attempted by advanced users willing to possibly fry their board. Using a very fine-tipped soldering iron, advanced users might choose to attempt to replace C_1 and C_2 in these circuits with capacitors 1 or 2 orders of magnitude larger or smaller than those selected here in order to achieve different frequency ranges for the low-pass filter on the ADC channels. Alternatively, and more simply, removing C_2 altogether (i.e., taking $C_2 \rightarrow 0$ in the transfer function listed above and simplifying) reduces the circuit to a first-order filter with transfer function

$$\frac{V_{out}(s)}{V_{in}(s) - V_{ref}} = A_A A_C \frac{\omega_1}{s + \omega_1}$$

with cutoff frequency $\omega_1 = 1/[C_1(R_1 + R_2)]$; setting $R_1 = R_2$ and adjusting both over the range from 100 k Ω to 1 k Ω results in $f_1 = \omega_1/(2\pi)$ ranging from 12 Hz to 1200 Hz. Subsequently replacing C_1 with a capacitor 1 or 2 orders of magnitude larger or smaller can again be done to achieve different frequency ranges. Finally, removing both C_1 and C_2 altogether (and setting $R_1 = R_2 = 1$ k Ω) removes the low-pass filter entirely, allowing the user to take responsibility for any necessary analog low-pass filtering (on a Beret Shield) of an ADC input before it is sampled by the STM's ADC unit. **Warning:** please re-read the warning at the beginning of this paragraph. Ok, you've been warned, good luck. (Actually, if you have a specific/substantial need for low-pass filtering over a different frequency range, it's probably better to contact us and have us make a variant of the board with different capacitors installed...)

5.7.3 Analog Header and user-developed analog filters

As shown in Table 5.2, the Analog Header provides the following nine analog (0V – 3.3V) signals:

- two buffered (up to +/- 400 mA) outputs `DACbuf1`, `DACbuf2`,
- the positive and negative inputs, and (0V – 3.3V, ± 400 mA) output, of a power opamp, $\{V+, V-, Vo\}$,
- the voltage `Vref` (input to the Beret) about which the ADCs are compared and amplified (see §5.7.2),
- two inputs `ADC1`, `ADC2`, and
- the low-pass-filtered `ADC2filt` [i.e., the analog output of OPAMP B (see Figure 5.10c) in the ADC2 filter].

Again, low-pass filtering with tunable gain (x1 to x4096), tunable cutoff frequency (f_c ranging from 34 Hz to 3400 Hz) and tunable damping (nominally, $\zeta \approx 0.7$) is applied to the ADC inputs before sampling by the STM.

Note also that GND, 3.3V, and Vs2 are readily available on the nearby SPI and I2C Headers.

The functionality describe above facilitates the connection of a number of analog sensors and actuators, the experimental determination of MIMO Bode Plots of continuous-time electro-mechanical systems, as well as the easy implementation of other user-developed analog filters on [Beret Shields](#) (see, e.g., TI's [Op Amps For Everyone](#) for several filter ideas).

Warning: no specific ESD protection is provided on the pins of the Analog Header.

³⁴By sinusoidally exciting a DAC channel over a range of frequencies, and routing the output directly to an ADC channel, the Bode plot of the corresponding Sallen-Key second-order low-pass filter may be measured directly, and the values of R_1 and R_2 subsequently tuned in software to achieve the desired filter response.

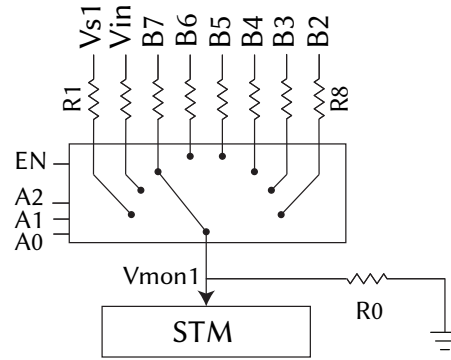


Figure 5.11: Vmon1 circuit with a TI MUX508 8:1 multiplexer, used to monitor {Vs1, Vin, B7, B6, B5, B4, B3, B2} on the **full size** Berets, with $\{R1, R2, R3, R4, R5, R6, R7, R8\} = \{27.4, 76.8, 73.2, 59.0, 45.3, 31.6, 17.4, 3.74\}$ k Ω and $R0 = 10$ k Ω [standard 1% resistor values in the E96 series; see Table 10.5]. Note that B1=GND. When EN=0, Vmon1=GND, and the current through all voltage dividers is zero. Note that Vmon2 is connected directly to Vs2, with no voltage divider. A TI TMUX6219 SPDT switch is used to monitor {Vs1, Vin} only on the **half size** Beret, using effectively same circuit and dropping {A2, A1}, with the same values of {R1, R2} and R0. The nominal scale factors relating the measured values at Vmon1 to the quantities of interest is laid out in Table 5.7.

control inputs	000	001	010	011	100	101	110	111
input voltage	Vs1	Vin	B7	B6	B5	B4	B3	B2
nominal scale factor	3.74 \times	8.68 \times	8.32 \times	6.90 \times	5.53 \times	4.16 \times	2.74 \times	1.374 \times

Table 5.7: Quantities measured by the circuit in Figure 5.11 for different inputs {A2, A1, A0} on the **full size** Berets, and their nominal associated scale factors. Only the first two columns apply on the **half size** Berets.

5.7.4 Voltage monitoring of Vin, Vs1, Vs2, and the individual battery cells

Berets periodically monitor Vin, Vs1, and Vs2 when running. Noting warnings ii and iii of §5.2.1, **full size** Berets also periodically monitor (using the custom JST-XH balance connector described in Note F of Table 5.2) the differential voltage over the individual battery cells of 2S – 6S batteries via their stock JST-XH balance connectors, to ensure that no individual cell drops below the minimum allowed voltage, Vmin, while the Beret is operating. Vmin is adjustable in software; a value in the range of 3.1V to 3.2V is appropriate for LiPos.

Voltage monitoring of Vin, Vs1, and the battery cells is done using the multiplexed voltage divider circuit in Figure 5.11, with EN and {A2, A1, A0} tied to the GPIO expander (see Table 5.6) to enable the system and select the active input. The STM's ADC3_IN3 channel, denoted Vmon1 on the Berets, periodically monitors the output of the 8:1 multiplexer in this circuit on the **full size** Berets, or of the SPDT switch on the **half size** Beret, while the STM's ADC3_IN3 channel, denoted Vmon2, periodically monitors Vs2 directly. As suggested by the analysis of TI Report SLVA450A, a target current of about 0.3 mA was chosen for the voltage dividers. For a target maximum value for Vmon1 of about 3.2V, this gives R0=10 k Ω . Given the relationship between the voltages at the top, middle, and bottom of a voltage divider (see Example 10.1), the resistor values Ri, for i=1 to 8, were then selected (see code) so that Vmon1 was about 3.2V for the maximum values of each input Vi.

1% tolerance resistors (E96 series, 0.01 W rating is sufficient) are implemented in the Vmon1 circuit on the Berets. This gives some inaccuracy in the calculated voltages, as quantified by equation (13) of SLVA450A. The nominal scale factors (to determine each input voltage Vi from the measured value of Vmon1) listed in Table 5.7 should thus be **calibrated** using otherwise-measured voltages {Vs1, Vin, B7, B6, B5, B4, B3, B2} in order to eliminate these inaccuracies, as the relationships between the Vi and Vmon1 are accurately linear. Note that higher-precision resistors (with the same nominal values, in the E192 series) could be used in this circuit, but doing such is expensive and unnecessary if the scale factors are to be calibrated after manufacturing.

5.8 Beret Shields

On all six Berets, convenient and stackable **Small** (1.3" x 0.9") and **Extended** (1.3" x 1.1", 1.3" x 1.3", or larger) **Beret Shields** (that is, small daughterboards) may be attached, in a manner similar to [Arduino Shields](#), atop

- (i) the (1x9) SPI/I2S Header,
- (ii) the (1x9) I2C Header,
- (iii) the (1x9) Analog Header (if present), and
- (iv) (on 1.3" x 1.1" or larger Beret Shields) the first row (signals S1 - S5) of Signal Header A, or
(on 1.3" x 1.3" or larger Beret Shields) all three rows (S1 - S5, Vs1/Vin, and GND) of Signal Header A,

thus enabling the user to build up quickly, and attach securely, any extra analog or digital circuitry that might be needed in a given application. The pins on these headers are aligned on a 0.1" grid, facilitating the use of:

Prototyping Beret Shields, with an array of predrilled holes on a 0.1" grid, which may be

- **plated**, for rapid development and testing of simple circuit designs, or
- **unplated**, providing a sturdy mechanical backing for COTS PCBs.

Prefabricated Beret Shields implementing commonly needed additional components, such as:

- 2x custom 128-pin solderless breadboards + 2x more 1x9 Headers (USART and GPIO) + LEDs/buttons,
- a 0.96" OLED display + 2x buttons,
- 2x more [BDC motor drivers](#) (24 half bridges), wired as described in §5.3,
- 6x sensorless [BLDC motor drivers](#) with integrated 28V/3A MOSFETs and [JST-PA](#) connectors,
- 2x continuous-time (CT, i.e., analog) notch filters (to eliminate the tonal "buzz" in 2 input signals),
- 2x CT lead/lag/PID feedback control circuits with digitally-adjustable poles, zeros, and gain,
- a u-blox [zed-f9p](#) dGPS/GNSS unit,
- a wifi/bluetooth module,
- an array of additional buttons, LEDs, and Recon UART and I2C connectors,
- arrays of connectors supporting other standards (PMOD, Grove, STEMMA, Qwiic, etc).

Custom Beret Shields compactly implementing your choice of components, layout, and connectivity.

Examples are shown in Figure 5.12. All three types of Beret Shields are low cost and easy to use. In particular:

- Prefabricated Beret Shields provide a fast and flexible way to extend the capability of the Beret ecosystem with a variety of commonly-needed additional components via open hardware designs.
- Custom Beret Shields facilitate the dense and secure arrangements of electronic components of the user's choosing for long-term use, and may easily be [designed](#) using free software, leveraging directly the open hardware circuit designs of the Prefabricated Beret Shields, and may be [fabricated](#) at remarkably low cost.

The (1.3" x 0.9") Small Beret Shields connect to the Beret using [three 1x9 male headers](#). The (1.3" x 1.1" or larger) Extended Beret Shields may also include a [1x5 female header](#) connecting to the first row of Signal Header A, whereas the (1.3" x 1.3" or larger) Extended Beret Shields may include a [3x5 female header](#) connecting to all three rows of Signal Header A, including high-current [Vs1](#) or [Vin](#), and [GND](#), on the second and third rows.

Use of a (1.3" x 0.9") Small Beret Shield leaves unobstructed all JSTs, buttons, and LEDs on the Berets, in addition to all 5 columns of Signal Header A and all 5 columns of Signal Header B, for easy attachment of 10 servo and/or ESC connectors. A (1.3" x 1.1" or 1.3" x 1.1") Extended Beret Shield connects directly to Signal Header A, but leaves unobstructed all 5 columns of Signal Header B.

Small and Extended Beret Shields are directly portable across the entire line of Berets. Note that the entry-level **Red** Beret does not have an Analog subsystem, and the corresponding Analog Header is absent; Beret Shields that do not use the analog subsystem are still fully compatible with this board.

We are also designing a stand-alone high-current brushless motor driver board, in the footprint of a Raspberry Pi, with an [STM32G474VE](#) (controllable over a Recon SPI connector), 6x [gate drivers](#), 18x [40V/50A dual MOSFETs](#), and high-current [XT60](#) (LiPo) and 6x [MR30](#) (BLDC motors) AMASS connectors.

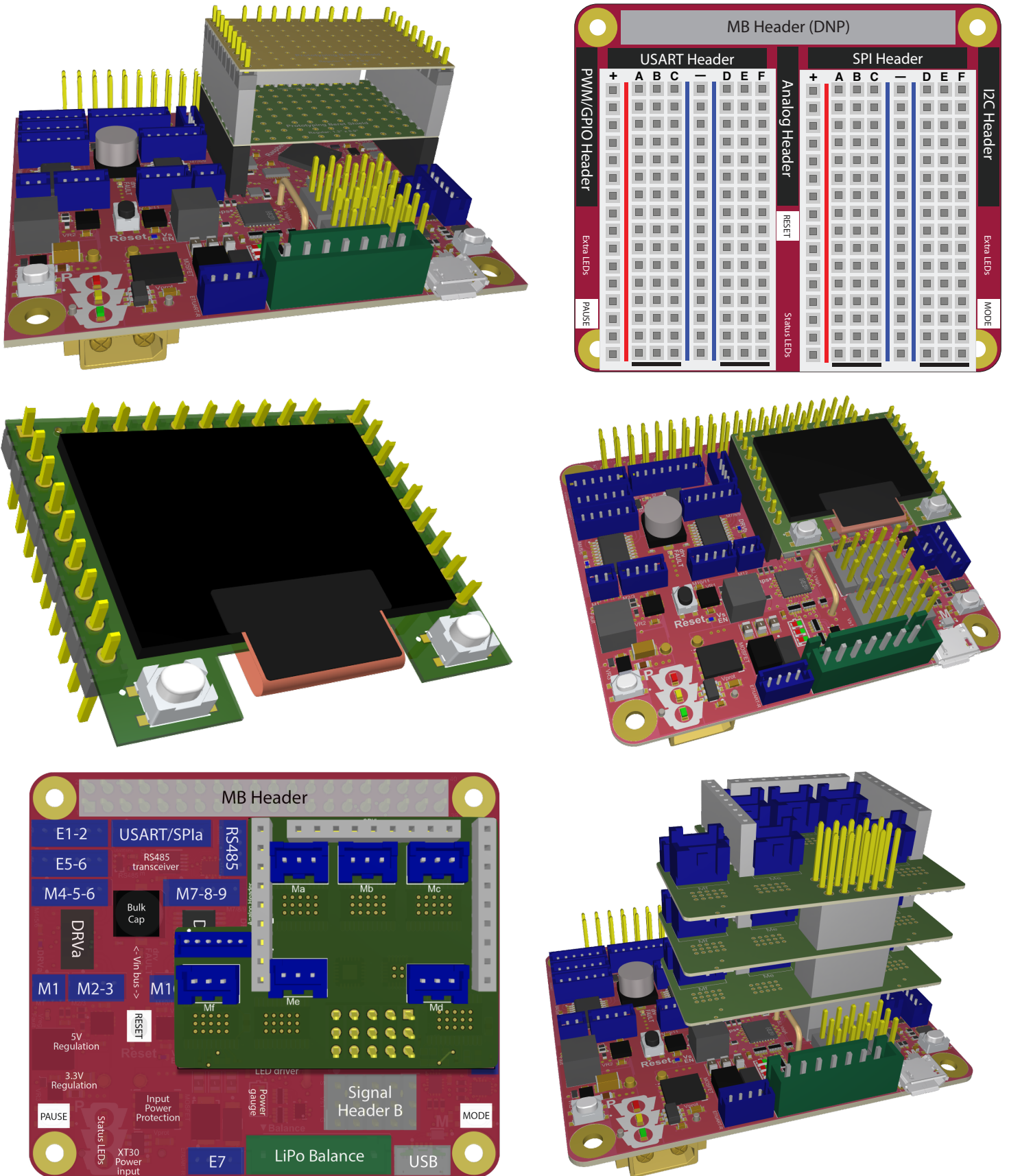


Figure 5.12: Layout of the (top) Prototyping (unplated and plated) and Breadboard, (middle) OLED, and (bottom) BLDC Beret Shields, including various views illustrating how they mount (and, stack) on a **Raspberry** Beret.

5.9 MB headers, MB header breakout SHIMs, and ID EEPROMs

5.9.1 RPi-compatible motherboards (MBs)

As shown in Table 5.8, the 2x20 header on the **RPi** Berets has 2 pins over which 5V/2A power may be provided to the RPi, 2 pins over which 3.3V power (regulated by the RPi) may be picked up by connected daughterboards, 8 GND pins, and 28 “BCM” pins, arranged in a peculiar order, connected to the RPi’s Broadcom MCU. BCM0 and BCM1 are used for an I2C connection to an EEPROM to identify attached daughterboards (a.k.a. **HATs**). BCM2 through BCM27 may be used as GPIOs, denoted GPIO2 through GPIO27; each of these channels may instead be switched over in software to provide various alternative functions. The “primary” such alternative functions, available on all RPis with 40-pin headers (and most RPi clones, as outlined in Table 1.7), are listed in Table 5.8.

A more comprehensive list of the useful alternative functions of GPIO2 through GPIO27 is given by Table 5.9, which also highlights many of the new alternative functions available with the RPi4. In this table:

- CTS and RTS are (optional) UART hardware flow control (**HFC**) channels,
- PCM (pulse-code modulation) is an advanced digital audio standard (used, e.g., by **hifiberry**),
- **SD0** is a proprietary Broadcom controller channel used to boot and communicate with the RPi eMMC,

Beret name	alt. function	PWR/BCM	pin	pin	PWR/BCM	alt. function	Beret name
	-	3.3V	1	2	5V	-	Vmb
	I2C1_SDA	GPIO2	3	4	5V	-	Vmb
	I2C1_SCL	GPIO3	5	6	GND	-	GND
mb_G0	GPCLK0	GPIO4	7	8	GPIO14	UART0_TX	
GND	-	GND	9	10	GPIO15	UART0_RX	
	SPI1_CE1	GPIO17	11	12	GPIO18	SPI1_CE0	
	SD0_DAT3	GPIO27	13	14	GND	-	GND
	SD0_CLK	GPIO22	15	16	GPIO23	SD0_CMD	
	-	3.3V	17	18	GPIO24	SD0_DAT0	
SPImb_MOSI	SPI0_MOSI	GPIO10	19	20	GND	-	GND
SPImb_MISO	SPI0_MISO	GPIO9	21	22	GPIO25	SD0_DAT1	
SPImb_SCK	SPI0_SCLK	GPIO11	23	24 [†]	GPIO8	SPI0_CE0	SPImb_SS[†]
GND	-	GND	25	26*	GPIO7*	SPI0_CE1	mag_INT*/SPImb_SS [†]
ID_SDA	I2C0_SDA	0 (reserved)	27	28	1 (reserved)	I2C0_SCL	ID_SCL
mb_G1	GPCLK1	GPIO5	29	30	GND	-	GND
mb_G2	GPCLK2	GPIO6	31	32*	GPIO12*	PWM0	bar_INT_DRDY*
imu_INT1_DRDY*	PWM1	GPIO13*	33*	34	GND	-	GND
	SPI1_MISO	GPIO19	35	36	GPIO16	SPI1_CE2	
	SD0_DAT2	GPIO26	37	38	GPIO20	SPI1_MOSI	
GND	-	GND	39	40	GPIO21	SPI1_SCLK	

Table 5.8: PWR function or BCM (Broadcom pin number, each associated with a GPIO) corresponding to each pin on the 2x20 header on the **RPi** Berets, along with the “primary” alternative function and corresponding Beret name (if any) of each, indicating **PWR**, **GPIO**, **UART**, **I2C**, **SPI**, and **SDIO** channels, as well as PWM and GPCLK functions. Table 5.9 lists the additional functions available on each. **Boldface** indicates channels that are connected by default to the Beret. (*) denotes 3 optional interrupt GPIOs, connected via backside solder jumpers; (†) denotes 2 possible SPI0 SS connections, 1 of which must be selected via a backside solder jumper (default is **SPI0_CE0**).

- **SD1** is a 50 MHz SD/SDIO standard channel for interfacing with secondary SD cards, eMMC, [wifi](#), etc,
- **ARM** is a universal 6-pin [JTAG debugging channel](#), with adaptive clocking via a Return TCK channel,
- ALT3 mode on BCM8 - BCM11 (on the RPi4) provides slave mode SPI or I2C, denoted here **SPIs** and **I2Cs**,
- PWM0/1 are hardware-generated (on the RPi) PWM channels, and GPCLK0/1/2 are [general purpose clocks](#).

Which function is selected on any given pin is [configured](#) by the corresponding {ALT0, ALT3, ALT4, ALT5} flag. [As explained at [linux.org](#), other features not shown (on ALT1 and ALT2) relate primarily to a secondary memory interface and a parallel display interface, neither of which can be used with the Raspberry Beret.]

BCM	pin	ALT0	ALT3	ALT4	ALT5	Beret name	SHIM	pin
0 (res)	27	I2C0_SDA	-	-	-	ID_SDA	I2C0	3
1 (res)	28	I2C0_SCL	-	-	-	ID_SCL	I2C0	4
GPIO2	3	I2C1_SDA	-	-	-		I2C1	3
GPIO3	5	I2C1_SCL	-	-	-		I2C1	4
GPIO4	7	GPCLK0	<i>SPI4_CE0</i>	<i>UART3_TX</i>	<i>I2C3_SDA</i>	mb_G0	GPIO	3
GPIO5	29	GPCLK1	<i>SPI4_MISO</i>	<i>UART3_RX</i>	<i>I2C3_SCL</i>	mb_G1	GPIO	4
GPIO6	31	GPCLK2	<i>SPI4_MOSI</i>	<i>UART3_CTS</i>	<i>I2C4_SDA</i>	mb_G2	GPIO	5
GPIO7*	26*	SPI0_CE1	<i>SPI4_SCLK</i>	<i>UART3_RTS</i>	<i>I2C4_SCL</i>	mag_INT*/ SPImb_SS[†]	GPIO	6*
GPIO12*	32*	PWM0	<i>SPI5_CE0</i>	<i>UART5_TX</i>	<i>I2C5_SDA</i>	bar_INT_DRDY*	GPIO	7*
GPIO13*	33*	PWM1	<i>SPI5_MISO</i>	<i>UART5_RX</i>	<i>I2C5_SCL</i>	imu_INT1_DRDY*	GPIO	8*
GPIO10	19	SPI0_MOSI	<i>SPIs_MOSI</i>	<i>UART4_CTS</i>	<i>I2C5_SDA</i>	SPImb_MOSI	SPI0	3
GPIO9	21	SPI0_MISO	<i>SPIs_MISO</i>	<i>UART4_RX</i>	<i>I2C4_SCL</i>	SPImb_MISO	SPI0	4
GPIO11	23	SPI0_SCLK	<i>SPIs_SCLK</i>	<i>UART4_RTS</i>	<i>I2C5_SCL</i>	SPImb_SCK	SPI0	5
GPIO8	24[†]	SPI0_CE0	<i>SPIs_CE</i>	<i>UART4_TX</i>	<i>I2C4_SDA</i>	SPImb_SS[†]	SPI0	6[†]
GPIO14	8	UART0_TX	<i>SPI5_MOSI</i>	<i>UART5_CTS</i>	UART1_TX		UART0	3
GPIO15	10	UART0_RX	<i>SPI5_SCLK</i>	<i>UART5_RTS</i>	UART1_RX		UART0	4
GPIO20	38	PCM_DIN	<i>SPI6_MOSI</i>	SPI1_MOSI	GPCLK0		SPI1	3
GPIO19	35	PCM_FS	<i>SPI6_MISO</i>	SPI1_MISO	PWM1		SPI1	4
GPIO21	40	PCM_DOUT	<i>SPI6_SCLK</i>	SPI1_SCLK	GPCLK1		SPI1	5
GPIO18	12	PCM_CLK	<i>SPI6_CE0</i>	SPI1_CE0	PWM0		SPI1	6
GPIO16	36	-	UART0_CTS	SPI1_CE1	UART1_CTS		SPI1	7
GPIO17	11	-	UART0_RTS	SPI1_CE2	UART1_RTS		SPI1	8
GPIO24	18	SD0_DAT0	SD1_DAT0	ARM_TDO	-		SDIO	3
GPIO25	22	SD0_DAT1	SD1_DAT1	ARM_TCK	<i>SPI4_CE1</i>		SDIO	4
GPIO26	37	SD0_DAT2	SD1_DAT2	ARM_TDI	<i>SPI5_CE1</i>		SDIO	5
GPIO27	13	SD0_DAT3	SD1_DAT3	ARM_TMS	<i>SPI6_CE1</i>		SDIO	6
GPIO22	15	SD0_CLK	SD1_CLK	ARM_TRST	<i>I2C6_SDA</i>		SDIO	7
GPIO23	16	SD0_CMD	SD1_CMD	ARM_RTCK	<i>I2C6_SCL</i>		SDIO	8

Table 5.9: Alternative functions of each of the 28 digital i/o pins on the RPi header, and the corresponding Beret names (if any). *Italics* indicate functions that are available on the RPi4 only. As in Table 5.8, indicated are **GPIO**, **UART**, **I2C**, **SPI**, **SDIO** channels, as well as PWM and GPCLK functions with, as before, **boldface** indicating default connections, and (*) and ([†]) indicating optional connections, between the Beret and the RPi. BCM pins 0 through 8 have default pull up resistors, and BCM pins 9 through 27 have default pull down resistors. Data from the [RPi v4 datasheet](#), [linux.org](#), and the [RPi forums](#). Note that {*SPIs_MOSI*, *SPIs_SCLK*} on the RPi4 (pins 19 and 23 in ALT3 mode) can also function as {*I2Cs_SDA*, *I2Cs_SCL*}.

As indicated by Table 5.9, the RPi4 (when NOT connected to the Beret) can simultaneously operate, e.g.:

- 4 **UART** channels: UART0/3/4 (w/ HFC) and UART5 (w/o HFC), or
- 4 SPI channels (w/ 8 total SS lines): SPI0 (w/ CE0), SPI1 (w/ CE0/1/2), and SPI4/5 (each w/ CE0/1), or
- 6 **I2C** channels: I2C1/3/4/5/6, plus (in slave mode) I2Cs [in addition to I2C0, reserved for the EEPROM], or
- mix A: UART1 (w/ HFC), UART4/5 (w/o HFC), SPI4/6 (each w/ CE0/1), and I2C1/5/6, or
- mix B: UART0 (w/o HFC), SPIs (in slave mode), SPI4 (w/ CE0), SPI1 (w/ CE0/1/2), I2C1, PWM0/1, SD1.

When an RPi2, RPi3, RPi4, or RPi Zero is fully connected via the RPi Header to a Beret, including the MB SPI channel, the MB ID I2C channel, the interrupt connections {imu_INT1_DRDY, bar_INT_DRDY, mag_INT}, and the GPIOs {mb_G0, mb_G1, mb_G2}, the RPi still has 16 unused BCM channels available on the RPi Header. These channels can be used as GPIOs or, alternatively, can simultaneously operate, e.g.:

- I2C1, UART0 (w/o HFC), SPI1 (w/ 3 available SS lines), and either SD1 or JTAG.

If using an RPi4, and/or not using one or more of the (optional) sensor interrupt (INT) connections mentioned above, various alternative channels and functions also become available if needed, as shown in Table 5.9.

In summary, even though the RPi Berets connect to up to 10 of the BCM2 to BCM27 channels on the RPi Header, in addition to the ID pins on BCM0/1, substantial connectivity options remain for connecting the RPi to other boards or devices. Further, using different SS pins on the SPI0 channel for each board, and programming the {mb_G0, mb_G1, mb_G2} channels appropriately, two RPi Berets can be directly attached to a single RPi using COTS RPi HAT stacking solutions (see, e.g., [here](#)).

As discussed further in the paragraph below, convenient MB header breakout SHIMs, which may be used even when one or more Beret(s) are attached to the MB, are available separately, and may be used to break out the additional functionality on the MB header discussed above onto standard Recon connectors.

MB Header Breakout SHIM. A small **SHIM** is under development to conveniently break out all 28 BCM pins of the RPi Header, one functional group at a time, in Recon order, in addition to incorporating an SD card holder and a multichannel LED display driver. **TODO: include further explanation and a pic of this SHIM, once available.** **Warning:** Though the MB Header breakout JSTs on this SHIM can provide 3.3V or 5V power, as selected via backside power jumpers, they operate at 5V TTL, not 3.3V TTL, so **any devices connected to the JSTs on this SHIM must be 5V tolerant.**

ID EEPROM. On the RPi Berets, the UDFN8 version of the (32 Kb) **CAT24C32** EEPROM is used for board identification, programmed as described in the **RPi HAT ID EEPROM spec**. The 7-bit address used for this EEPROM is 1010000b (0x50h) by default, as required by this spec; however, the last bit of this device's I2C address may be changed, via a backside solder jumper, to 3.3V or GND, thus enabling the use of either 0x50h or 0x51h as the ID EEPROM address on these Berets, and making the connection of two Berets to a single RPi straightforward (using an RPi header extension cable) while keeping both of the ID EEPROMs of the connected Berets individually readable. Note that the write protect (WP) pin on the EEPROM is by default connected to 3.3V (read only), but this may also be switched to GND (to enable write mode) via a backside solder jumper.

5.9.2 96B-compatible MBs

As shown in Table 5.10, the 2x20 header on 96B motherboards has 2 pins over which Vmb = 8V to 18V power may be provided to the 96B motherboard by connected daughterboards (aka **mezzanines**, e.g. the **Black** Beret), 1 pin over which 5V power (regulated by the 96B motherboard) may be picked up by connected mezzanines, 1 pin over which 1.8V power (regulated by the 96B motherboard) may be picked up by connected mezzanines, 4 GND pins, and 32 other pins, arranged in a rather well-structured order by their primary functions as defined by the MCU on the 96B motherboard. The pins on this header that are connected to the **Black** Beret are also indicated in Table 5.10. **Warning:** All digital pins on the 96B header operate at 1.8V TTL, and thus must usually be level shifted on attached mezzanines (e.g. to be used by 3.3V MCUs, as implemented on Berets).

MB Header Breakout SHIM. The pins on the 96B Header (Table 5.10) are logically ordered by their associated functions. A SHIM designed to level shift these functions to (5V tolerant) 3.3V TTL, and present these functions on JSTs in Recon order (with associated power/GND pins) will be developed if sufficient interest is expressed.

ID EEPROM. On the 96B Beret, the UDFN8 version of the (1 Mb) [CAT24M01WI-GT3](#) EEPROM is used for board identification, programmed (for the moment) as described in §5.9.1 (as for the RPi Berets); the programming of this (larger-capacity) EEPROM is subject to change during the next rev of the 96B [Mezzanine Design Guidelines](#).

Beret name	function	pin	pin	function	Beret name
GND	GND	1	2	GND	GND
	UART0_CTS	3	4	PWR_BTN_N	
	UART0_TxD	5	6	RST_BTN_N	
	UART0_RxD	7	8	SPI0_SCLK	SPImb_SCK
	UART0_RTS	9	10	SPI0_DIN	SPImb_MISO
	UART1_TxD	11	12	SPI0_CS	SPImb_CS0
	UART1_RxD	13	14	SPI0_DOUT	SPImb_MOSI
ID_SCL	I2C0_SCL	15	16	PCM_FS	
ID_SDA	I2C0_SDA	17	18	PCM_CLK	
	I2C1_SCL	19	20	PCM_DO	
	I2C1_SDA	21	22	PCM_DI	
mb_G0	GPIO-A	23	24	GPIO-B	
mb_G1	GPIO-C	25	26	GPIO-D	
mb_G2	GPIO-E	27	28	GPIO-F	
mag_INT*	GPIO-G	29*	30	GPIO-H	
bar_INT_DRDY*	GPIO-I	31*	32	GPIO-J	
imu_INT1_DRDY*	GPIO-K	33*	34	GPIO-L	
	1V8	35	36	SYS_DCIN	Vmb
5V	5V	37	38	SYS_DCIN	Vmb
GND	GND	39	40	GND	GND

Table 5.10: Primary functions corresponding to each pin on the 2x20 header of the 96B format, along with the corresponding net name (if any) on the **Black** Beret, indicating **PWR**, **GPIO**, **UART**, **I2C**, and **SPI**. **Boldface** indicates channels that are connected by default to the Beret. (*) denotes 3 optional interrupt GPIOs, attached to the header via backside solder jumpers.

5.9.3 BB-compatible MBs

As shown in Tables 5.11 and 5.12, the 2x23 header on BB motherboards has 2 pins over which a $V_{mb} \approx 5V$ power supply may be provided to the BB motherboard by connected daughterboards (aka capes, e.g. the **White** Beret), 2 pins over which 5V power (regulated by the 96B motherboard) may be picked up by connected capes, 2 pins over which 5V power (regulated by the BB motherboard) may be picked up by connected capes, 6 GND pins, and 34 other pins with various digital and analog functions. The pins on this header that are connected to the **White** Beret are also indicated in Tables 5.11 and 5.12.

MB Header Breakout SHIM. The pins on the BB Header (Tables 5.11 and 5.12) are generally clustered by their associated functions. A SHIM designed to present these pins on JSTs in Recon order (with associated power/ground pins) will be developed if sufficient interest is expressed.

ID EEPROM. On the BB Berets, the UDFN8 version of the (32 Kb) **CAT24C32** EEPROM (as also used on the RPi Berets) is used for board ID, programmed as described, e.g., in this [BB ID EEPROM programming tutorial](#).

Beret name	alt. function	PWR/GPIO	pin	pin	PWR/GPIO	alt. function	Beret name
GND	-	DGND	1	2	DGND	-	GND
	-	3.3V	3	4	3.3V	-	
Vmb	-	VDD_5V	5	6	VDD_5V	-	Vmb
	-	SYS_5V	7	8	SYS_5V	-	
	PWR_BTN	-	9	10	-	SYS_RESET	
	UART4_RX	GPIO_30	11	12*	GPIO_60*	-	mag_INT*
	UART4_TX	GPIO_31	13	14*	GPIO_40*	PWM1A	bar_INT_DRDY*
	-	GPIO_48	15	16*	GPIO_51*	PWM1B	imu_INT1_DRDY*
SPImb_SS	SPI0_CS0	GPIO_4	17	18	GPIO_5	SPI0_D1	SPImb_MISO
ID_SCL	I2C2_SCL	-	19	20	-	I2C2_SDA	ID_SCL
SPImb_MOSI	SPI0_D0	GPIO_3	21	22	GPIO_2	SPI0_SCLK	SPImb_SCK
mb_G0	-	GPIO_49	23	24	GPIO_15	UART1_TX	
mb_G1	-	GPIO_117	25	26	GPIO_14	UART1_RX	
mb_G2	-	GPIO_125	27	28	GPIO_123	SPI1_CS0	
	SPI1_D0	GPIO_111	29	30	GPIO_112	SPI1_D1	
	SPI1_SCLK	GPIO_110	31	32	VDD_ADC	-	
	ADC_IN4	-	33	34	GND_ADC	-	
	ADC_IN6	-	35	36	-	ADC_IN5	
	ADC_IN2	-	37	38	-	ADC_IN3	
	ADC_IN0	-	39	40	-	ADC_IN1	
	-	GPIO_20	41	42	GPIO_7	SPI1_CS1	
GND	-	DGND	43	44	DGND	-	GND
GND	-	DGND	45	46	DGND	-	GND

Table 5.11: PWR function or GPIO number, and the “primary” alternative function, corresponding to each pin on the 2x23 header on the **BB Black**, along with the corresponding net name (if any) on the **White** Beret, indicating **PWR**, **GPIO**, **UART**, **I2C**, and **SPI**, as well as PWM and ADC functions (cf. Table 5.12 for the corresponding numbering of the GPIO, UART, SPI, I2C, and PWM channels on the BB AI). **Boldface** indicates channels that are connected by default to the Beret. (*) denotes 3 optional interrupt GPIOs, attached to the header via backside solder jumpers.

Beret name	alt. function	PWR/GPIO	pin	pin	PWR/GPIO	alt. function	Beret name
GND	-	DGND	1	2	DGND	-	GND
	-	3.3V	3	4	3.3V	-	
Vmb	-	VDD_5V	5	6	VDD_5V	-	Vmb
	-	SYS_5V	7	8	SYS_5V	-	
	PWR_BTN	-	9	10	-	SYS_RESET	
	UART5_RX	GPIO_241	11	12*	GPIO_128*	-	mag_INT*
	UART5_TX	GPIO_172	13	14*	GPIO_121*	PWM3A	bar_INT_DRDY*
	-	GPIO_76	15	16*	GPIO_122*	PWM3B	imu_INT1_DRDY*
SPImb_SS	SPI2_CS0	GPIO_209	17	18	GPIO_208	SPI2_D0	SPImb_MISO
ID_SCL	I2C4_SCL	-	19	20	-	I2C4_SDA	ID_SCL
SPImb_MOSI	SPI2_D1	GPIO_67	21	22	GPIO_179	SPI2_SCLK	SPImb_SCK
mb_G0	SPI2_CS1	GPIO_203	23	24	GPIO_175	UART10_TX	
mb_G1	-	GPIO_177	25	26	GPIO_174	UART10_RX	
mb_G2	-	GPIO_111	27	28	GPIO_113	SPI3_CS0	
	SPI3_D1	GPIO_139	29	30	GPIO_140	SPI3_D0	
	SPI3_SCLK	GPIO_138	31	32	VDD_ADC	-	
	ADC_IN4	-	33	34	GND_ADC	-	
	ADC_IN6	-	35	36	-	ADC_IN5	
	ADC_IN2	-	37	38	-	ADC_IN3	
	ADC_IN0	-	39	40	-	ADC_IN1	
	-	GPIO_180	41	42	GPIO_114	SPI3_CS1	
GND	-	DGND	43	44	DGND	-	GND
GND	-	DGND	45	46	DGND	-	GND

Table 5.12: PWR function or GPIO number, and the “primary” alternative function, corresponding to each pin on the 2x23 header on the **BB AI**, along with the corresponding net name (if any) on the **White** Beret, indicating **PWR**, **GPIO**, **UART**, **I2C**, and **SPI**, as well as PWM and ADC functions (cf. Table 5.11 for the corresponding numbering of the GPIO, UART, SPI, I2C, and PWM channels on the BB Black). **Boldface** indicates channels that are connected by default to the MB Header on the Beret. (*) denotes 3 optional interrupt GPIOs, connected to the MB Header via backside solder jumpers on the Beret. Note in particular that, between the BB Black pinout depicted in Table 5.11, and the BB AI pinout depicted here, the D0 & D1 nets are swapped on pins 18 & 21, and on pins 29 & 30; these swaps are easily accounted for in software.

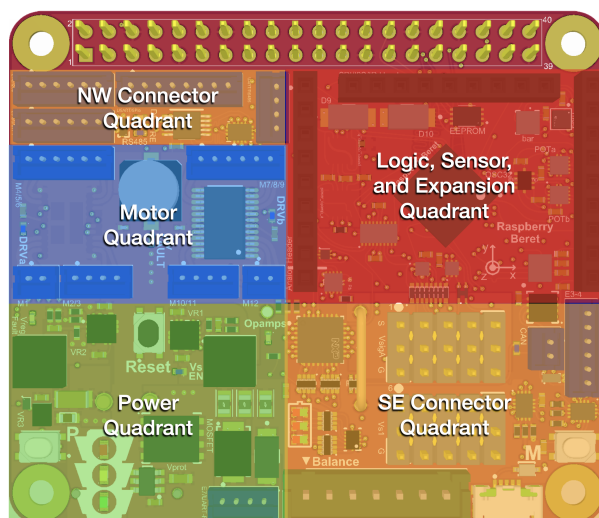


Figure 5.13: General layout of the Berets, as organized into quadrants.

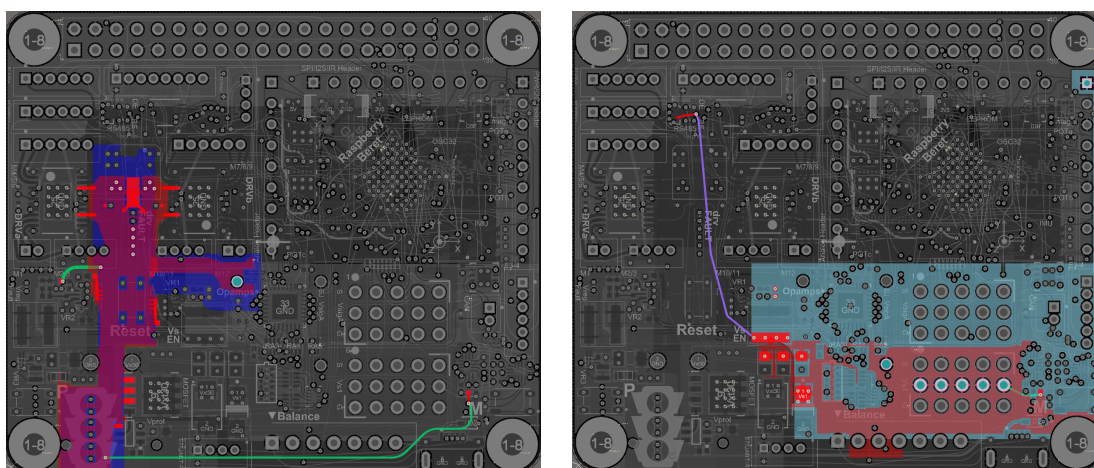


Figure 5.14: (left) Stitching of V_{in} traces on layers 1 and 8. (right) Stitching of V_{s1} traces on layers 1 and 4.

5.10 Layout

5.10.1 Overall organization and power flow

As indicated previously, it is convenient to refer to directions on the Berets in terms of directions on a compass. As illustrated in Figure 5.13, the layout of the Berets is generally organized into four main quadrants:

- the Logic, Sensor, and Expansion Quadrant, in the NE,
- the Connector Quadrant, split between the NW and SE,
- the Power Quadrant, in the SW, and
- the Motor Quadrant, located just N of Power Quadrant.

The bottom layer of the Power and Motor Quadrants is mostly large Power and GND pours, with many “thermal vias” connected to the (hot) undersides of the high-power components, for enhanced **thermal radiation**. Also, **short** and **wide** traces are used for all high-current pathways. **Stitching** of overlying vias is performed across multiple current-carrying layers for the highest-current pathways, including those taking V_{in} from the Power Quadrant to the Motor Quadrant, as shown in Figure 5.14a, and those taking V_{s1} from the Power Quadrant to the SE Connector Quadrant, as shown in Figure 5.14b.

Layer	Color	Logic & Connector Quadrants		Power & Motor Quadrants	copper oz/ft ²
		Function	Trace Directions	Function	
Top	■	Signal/ICs	N-S, E-W	Power/Signal/ICs	1.5
2	■	GND	(fills)	Power	2
3	■	Signal	E-W	Power/Signal	1
4	■	GND/5V/Vs1	(fills)	GND/5V/Vs1	2
5	■	GND/5V/Vs1	E-W	GND/5V/Vs1	1
6	■	Signal	N-S	Signal	1
7	■	3.3V	(fills)	3.3V	2
Bottom	■	Power/Signal	N-S	GND	1.5

Table 5.13: Eight-layer stackup used on the Berets, indicating the colors used in Figures 5.14, 5.16, and 5.17.

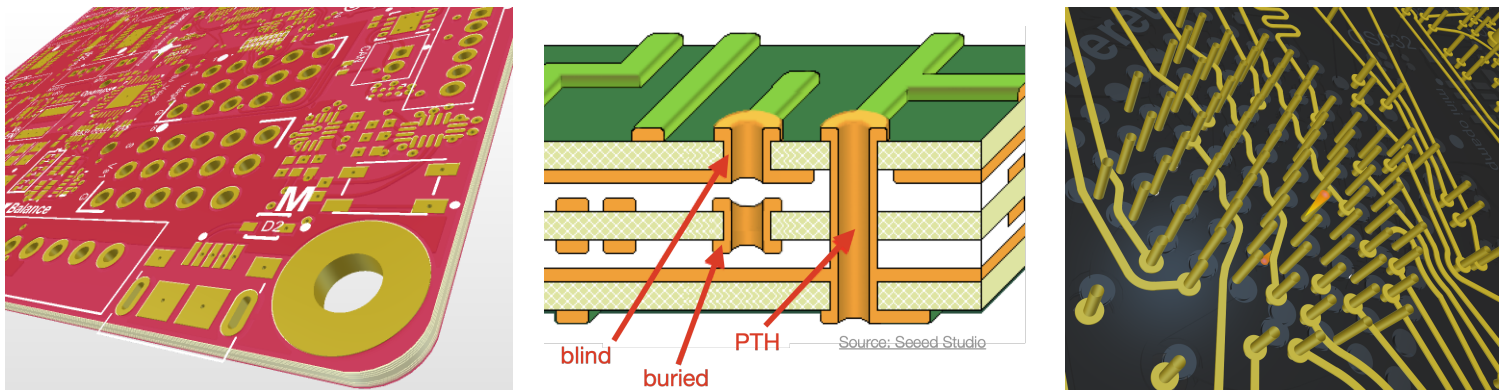


Figure 5.15: (left) Closeup of the SE corner of a Beret, illustrating the extensive use of via-in-pad techniques. (center) Comparison of blind, buried, and through-hole vias. (right) Copper traces on one of the layers under the BGA on a Beret, illustrating the removal of unnecessary annular rings, which gives substantially increased clearance for breaking out the 8mm pitch BGA using 6 mil traces, 7 mil spaces, and through-hole vias only.

5.10.2 Layer stackup, signal routing, and high-density integration (HDI)

With some careful design effort (see in particular the stackup plan in Table 5.13), Berets achieve a remarkable high-density integration (HDI) of components and functionality in a very small footprint for an 8-layer board. This is achieved, in part, by making extensive use of **via-in-pad** technology, which allows the placement of solder pads for various components directly over vias, as illustrated in Figure 5.15a. The (per board, not per instance) cost of implementing this modern (but by now fairly common) technology for HDI, which necessitates plugging the vias and plating them over, is *much* less than the cost of traditional **blind vias** (exposed on only one side of the PCB) and **buried vias** (not exposed on either side of the PCB), as illustrated in Figure 5.15b. Indeed, *all* vias on the Berets are in fact the (much lower-cost) **through-hole vias**.

Careful tradeoffs were involved in the stackup design (Table 5.13). Layer thicknesses had to be made:

- sufficiently thin to use 6 mil traces, 7 mil spaces, 6 mil diameter vias, and 18 mil diameter annular rings to break out the 8 mm (31.5 mil) pitch BGA (ball grid array) on the STM, and
- sufficiently thick to handle high current (up to 12A in places) where necessary.

A useful technique implemented to break out the BGA at this resolution was the removal of the (unnecessary) annular rings on the (through-hole) vias under the BGA on layers in which these vias did not actually connect to traces, as illustrated in Figure 5.15c. This resulted in increased clearance to route traces out from under the BGA between the closely-spaced vias with, on any given layer, most of these annular rings removed.

Layout of all 8 layers of the Raspberry Beret is illustrated in full in Figures 5.16-5.17. As indicated in Table 5.13, note that different thicknesses of copper are used on these different layers, based on their differing primary functions, which is helpful to better handle the high-current traces/pours. Though this is a tad unusual, most PCB fab facilities can accommodate this when fabricating PCBs at high volume.

To address the routing of the many crossed traces in the design, the following approach was followed:

- primarily N-S traces were isolated on layers 6 and 8 (see, e.g., the purple traces of Layer 6 in Figure 5.17), while
- primarily E-W traces were isolated on layers 3 and 5 (see, e.g., the green traces of Layer 5 in Figure 5.17).

Carefully selecting where such traces are joined (using vias) facilitated the “untangling” of the hundreds of nets involved in the design.

5.10.3 EMI and signal-integrity considerations

Four primary techniques were used to maintain **signal integrity** on high-speed communication channels (SPI, USART, I2C) and (simultaneously) to reduce the electromagnetic interference (**EMI**) generated by the board:

1. GND and/or Power planes were situated immediately next to each high-speed signal trace³⁵.
2. Curved traces with no sharp corners were used everywhere.
3. Matched-length traces were used, on each layer, for the parallel traces associated with clocked high-speed communication channels (e.g., MOSI, MISO, SCK), as illustrated, e.g., by the extra wiggles of the purple traces in the NE corner of Layer 5 in Figure 5.17.
4. Power GND, which inevitably fluctuates some due to the strongly time-varying loads placed on it (associated with the PWM generation used by the high-power components), was carefully isolated from Signal GND.

The use of modern ECAD software (Altium) was essential in order to implement techniques 2 and 3 above.

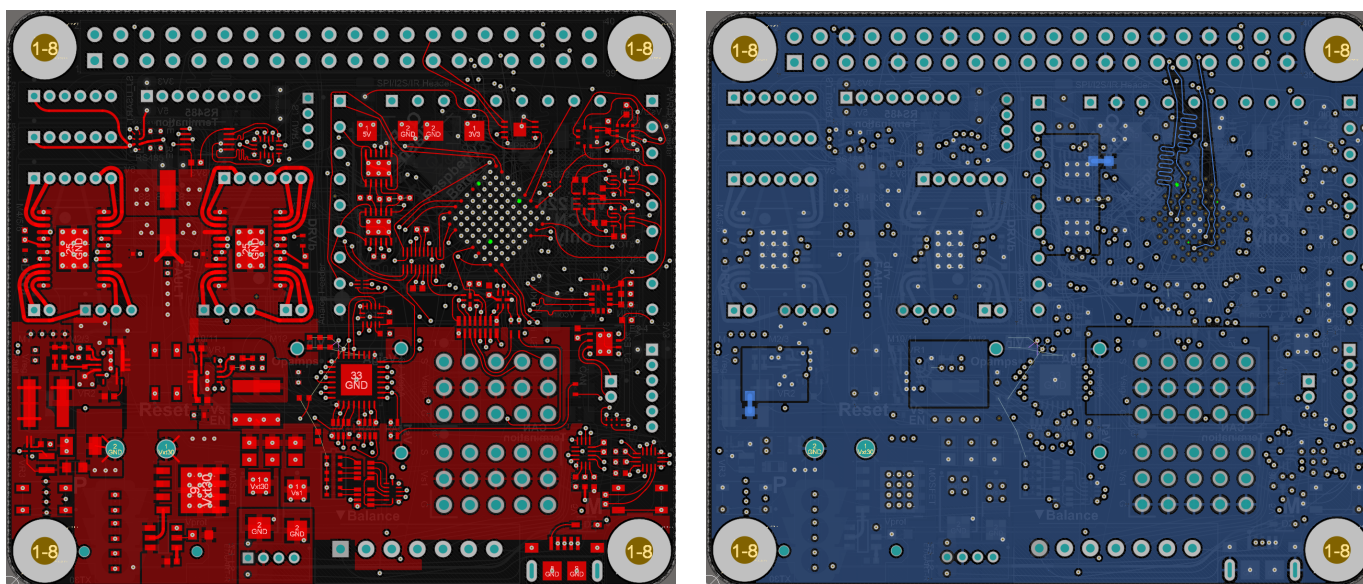


Figure 5.16: Layout of layers 1 and 2 of the **Raspberry** Beret (continued on next page).

³⁵A high-speed signal is actually mostly carried in the space *between* the trace and the corresponding reference layer, not along the trace itself, so the close proximity of such reference layers is essential in order to not turn your PCB into a radio unintentionally!

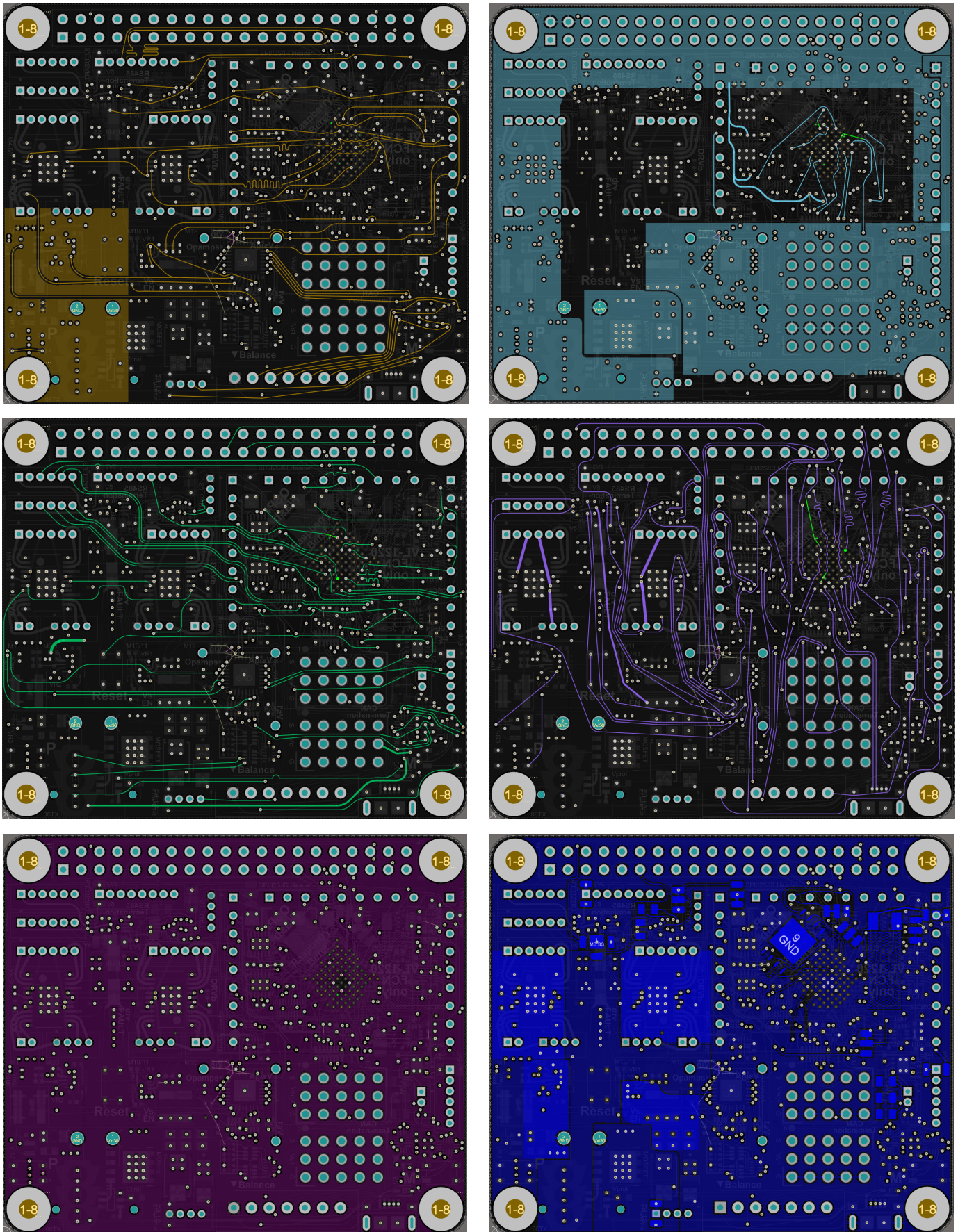


Figure 5.17: Layout of layers 3 through 8 of the **Raspberry** Beret (continued from previous page).

manufacturer	part	description	\$	R	r	K	W	G	B	§
TI	DRV8912-Q1	motor drivers	2.42	2	1	2	2	0	1	5.3
TI	CSD18510Q5B	power <u>MOSFET</u>	0.75	1	1	1	1	1	1	5.2.3
TI	LM74700-Q1	<u>ideal diode</u> controller	0.62	1	1	1	1	1	1	5.2.3
TI	TPS56637	<u>Vin->Vs1</u> , <u>Vin->Vmb</u> , <u>Vin->3.3V</u> Vregs	1.32	2	2	2	2	2	2	5.2.4 5.2.5 5.2.6
TI	TI TPS6208833	<u>Vmb->3.3V</u> switching Vreg	1.10	1	1	0	1	0	0	5.2.7
TI	TI TPS62913	<u>Vmb->3.3V</u> switching Vreg	1.10	0	0	1	0	0	0	5.2.7
TI	TS5A3359	<u>Venc mux</u> (3:1+off)	0.34	1	1	1	1	1	1	5.2.9
TI	TXB0108DQSR	8 channel <u>5V - 3.3V</u> level shifter	0.39	1	0	1	1	0	0	5.5
TI	TCA6507	<u>LED driver</u>	0.40	1	1	1	1	1	1	5.5
TI	THVD1452	<u>RS485 transceiver</u>	1.003	1	0	1	1	1	0	5.6.4
TI	TCAN334GDCNR	<u>CAN-FD transceiver</u>	1.114	1	0	1	1	1	0	5.6.4
TI	SN74CBTLV3257	<u>RS485-UART mux</u>	0.200	1	0	1	1	1	0	5.6.4
TI	ALM2402-Q1	<u>power opamps</u> (DAC1, DAC2, Vs2, user)	1.13	2	0	2	2	2	2	5.7.1 5.7.1 5.7.3
TI	TLV9002	dual <u>mini opamp</u> (low-pass filters)	0.21	1	0	1	1	1	1	5.7.2
TI	TPL0102-100	digital <u>pots</u> (Vs1, Vs2, $\omega_{c1a/b}$, $\omega_{c2a/b}$)	0.60	3	1	3	3	3	3	5.2.4 5.7.1 5.7.2
TI	TMUX1208	<u>Vmon mux</u> (8:1+off)	0.32	2	2	2	2	0	0	5.7.4
TI	TS5A23166	<u>Vmon SPDT switch</u>	0.24	1	1	1	1	0	0	5.7.4
TI	TMUX1204DQAR	<u>Vmon 4:1 switch</u>	0.09	0	0	0	0	1	1	5.7.4
ST	STM32G474VEH6	microprocessor with 512KB Flash	4.488	1	0	1	1	1	1	5.5
ST	STM32G474VBH6	microprocessor with 128KB Flash	3.521	0	1	0	0	0	0	5.5
ST	LIS3MDLTR	3-axis <u>magnetometer</u>	0.714	1	1	1	1	1	1	5.4
ST	LPS22HB	<u>barometer</u>	1.28	1	1	1	1	1	1	5.4
TDK	ICM-42688-P	6-axis <u>IMU</u>	2.997	1	1	1	1	1	1	5.4
NXP	PCAL6524HEHP	GPIO <u>expander</u>	0.825	1	1	1	1	0	0	5.5
SiTIME	SIT1532AC	32.7680 kHz <u>oscillator</u> (STM RTC & IMU)	0.60	1	1	1	1	1	1	5.5.1 5.4
ON	CAT24C32HU4I-GT3	32 Kb I2C EEPROM	0.276	1	1	0	1	0	0	5.9.1
ON	CAT24M01WI-GT3	1 Mb I2C EEPROM	0.523	0	0	1	0	0	0	5.9.2

Table 5.14: BOM Part A: primary components of the **Ras**, **Red**, **Black**, **White**, **Green**, **Blue** Berets (R, r, K, W, G, B). Identifying components by the underlined abbreviated names is convenient.

5.11 Bill Of Materials (BOM)

The Bill Of Materials of the Raspberry Beret is listed in Tables 5.14-5.15. A complete BOM, including all minor components (including, e.g., all of the small 0402 discrete resistors and capacitors) are listed in the Altium viewer for the project, available online at <http://dynamics.ucsd.edu/berets>. Some commonly-needed add-on components are listed in Table 5.16. **Random notes and questions:**

1. The custom 7-pin XH-compatible has a slot cut out of one side of a JST-XH, so 3-pin to 6-pin connectors can be plugged in as well. We need to find someone to manufacture that for us. See the [Revolectrix SPA Single Port Safe Parallel Adapter](#) and the [ISDT PC-4860 1S-6S Lipo Battery Charger](#) for examples.

2. Do we need **common-mode chokes** for Electro Magnetic Interference (EMI) filtering on USB, RS485, or CAN? This [post](#) says it degrades USB signal quality, but might be necessary to pass [FCC Title 47 CFR Part 15](#). This [post](#) discusses it for CAN. For RS485 and CAN, maybe we can leave places for it off of the Beret, out on the enclosure bulkhead (next to the spot for the optional TVS diodes; see §5.6.4).

manufacturer	part	description	\$	R	r	K	W	G	B	§
Littelfuse	SMBJ33CA	main TVS diode for Vin spikes	0.140	1	1	1	1	1	1	5.2.3
ON	1SMA5928BT3G	13V zener diode for 12V spikes	0.0924	1	1	2	1	1	1	5.2.3
ON	1SMA5919BT3G	5.6V zener diode for 5V spikes	0.0924	1	1	1	1	0	0	5.2.3
ON	1SMA5914BT3G	3.6V zener diode for 3.3V spikes	0.0924	1	1	1	1	1	1	5.2.3
ST	BAT60JFILM	10V Schottky for USB 5V protection	0.0363	1	1	1	1	1	1	5.2.3
TI	TPD6E004	ESD diode arrays for USB, S1-S10	0.16	2	2	2	2	2	2	5.2.3
Bourns	SRP5050FA-5R6M	5.6 μ H (7.2A) inductors on Vs1, Vmb	0.533	2	2	2	2	2	2	{ 5.2.4 5.2.5 }
Vishay Dale	IHHP0806ABERR22M01	220 nH (5.3A) inductor on 3.3V	0.148	1	1	0	1	1	1	5.2.7
Coincraft	XGL4030-472	4.7 μ H (3.2A) inductor on 3.3V	0.49	0	0	1	0	0	0	5.2.7
Chemi-Con	EMZR350ARA101MF61G	100 μ F bulk cap on Vin	0.220	1	1	1	1	1	1	5.2.4
Samsung	CL21A226MAYNNNE	22 μ F output caps on Vs1	0.0716	3	3	3	3	3	3	5.2.4
TDK	C3216X5R0J686M160AB	68 μ F output cap on Vmb	0.2871	1	1	0	1	0	0	5.2.5
Taiyo-Yuden	EMK316BBJ476ML-T	47 μ F output cap on Vmb	0.2210	0	0	1	0	0	0	5.2.5
Samsung	CL10A226MQ8NRNE	22 μ F output cap on 3.3V	0.0527	1	1	0	1	1	1	5.2.7
Murata	GRM188R60J476ME15D	47 μ F output cap on 3.3V	0.1493	0	0	1	0	0	0	5.2.7
Panasonic	EXB-28V820JX	<u>82 Ω 4RA</u> for {B,G} LEDs, S1-S10	0.0084	1	1	1	1	1	1	5.2.3
Panasonic	EXB-24V820JX	<u>82 Ω 2RA</u> for {B,G} LEDs, S1-S10	0.0105	1	1	1	1	1	1	5.2.3
Panasonic	EXB-28V221JX	<u>220 Ω 4RA</u> for {A,Y,R,R/G} LEDs	0.0084	1	1	1	1	1	1	5.2.3
Panasonic	EXB-24V221JX	<u>220 Ω 2RA</u> for {A,Y,R,R/G} LEDs	0.0105	1	1	1	1	1	1	5.2.3
Inolux	IN-S42BTR	<u>red</u> LED (stoplight)	0.0606	1	1	1	1	1	1	5.6.8
Inolux	IN-S42BT5Y	<u>yellow</u> LED (stoplight)	0.0707	1	1	1	1	1	1	5.6.8
Inolux	IN-S42BT5G	<u>green</u> LED (stoplight)	0.0818	1	1	1	1	1	1	5.6.8
Kingbright	APHB1608LZGKSURKC	<u>bicolor</u> LEDs (power gauge)	0.2282	3	3	3	3	3	3	5.6.8
Inolux	IN-S42BT5B	<u>blue</u> LEDs (enable status)	0.0873	6	5	6	6	2	2	5.6.8
Inolux	IN-S42BT5A	<u>amber</u> LEDs (fault status)	0.0707	3	3	3	3	2	2	5.6.8
C&K	PTS815-SJM-250-SMTR	white buttons (<u>pause</u> , <u>mode</u>)	0.111	2	2	2	2	2	2	5.6.8
C&K	PTS815-SJG-250-SMTR	black button (<u>reset</u>)	0.143	1	1	1	1	1	1	5.6.8
Amass	XT30PW-M	sideways <u>XT30</u> (main power in)	0.500	1	1	1	1	1	1	5.2.1
custom	custom	custom 7-pin XH (Balance)	<i>cost?</i>	1	1	1	1	0	0	5.7.4
4Ucon	01056	3x5 0.1" male (<u>SIGa</u> , <u>SIGb</u>)	0.0422	2	1	2	2	2	1	5.6.2
4Ucon	11071	<u>USB Micro-B</u> female	0.0795	1	1	1	1	1	1	5.6.9
4Ucon	00532	{ 1x9 0.1" female for <u>Analog</u> , <u>SPI</u> , <u>I2C</u> Headers	0.0625	3	2	3	3	3	3	5.8
JST	B8B-ZR-3.4(LF)(SN)	8-pin JST-ZH for <u>USART</u>	0.171	1	1	1	1	1	1	5.6.5
JST	B6B-ZR-3.4(LF)(SN)	{ 6-pin JST-ZH for <u>M4-5-6</u> , <u>M7-8-9</u> , <u>E1-2</u> , <u>E3-4</u> , <u>E6-7</u>	0.131	5	3	5	5	2	3	{ 5.3 5.6.1 }
JST	B4B-ZR-3.4(LF)(SN)	{ 4-pin JST-ZH for <u>M2-3</u> , <u>M10-11</u> , <u>E5 RS485/UART</u>	0.096	4	2	4	4	2	3	{ 5.3 5.6.1 5.6.4 }
JST	B2B-ZR-3.4(LF)(SN)	2-pin JST-ZH for <u>M1</u> , <u>M12</u> , <u>CAN</u>	0.072	3	1	3	3	1	1	5.6.5
4Ucon	20565	2x20 0.1" stackable <u>RPi</u> header	0.4139	1	1	0	0	0	0	5.9.1
4Ucon	00324 pins too short?	2x20 2mm <u>96B</u> header	0.4139	0	0	1	0	0	0	5.9.2
4Ucon	20582	2x23 0.1" stackable <u>BB</u> header	0.4139	0	0	0	1	0	0	5.9.3

Table 5.15: BOM Part B: secondary components for the **Ras**, **Red**, **Black**, **White**, **Green**, **Blue** Berets (R, r, K, W, G, B). This table still under construction.

manufacturer	part	description	\$	#	§	usage notes	
Panasonic	VL1220/FCN	rechargeable 3V coin cell	4.03	1	5.2	with SMT bracket (solder on backside)	
{ Winbond Winbond GigaDevice	W25Q64JVZEIQ	8 MB 133 MHz	1.053	}	1	5.5.2	QSPI Flash (solder on backside)
	W25N512GVEIG	64 MB 166 MHz	1.95				
	GD5F1GQ4UFYIGR	128 MB 120MHz	3.05				
4Ucon	00812	1x9 0.1" male	0.0207	3	5.6.2	mates with Analog, SPI, I2C headers	
4Ucon	00526	1x5 0.1" female	0.0370	1	5.8	mates with first row of SigA header	

Table 5.16: Commonly-needed add-on components for Berets, including the rechargeable coin cell, flash memory, and connectors for Beret Shields. Prices quoted are for single unit quantities as of Spring 2021, except for the 4Ucon connectors, which are quoted for quantities of 1000. The components used for CAN and RS485 termination (also an optional add-ons) are standard 0804 resistors and capacitor (see §5.6.4), solder footprints for which are provided on the back of the Berets. **This table still under construction.**

5.12 Schematics

A shared schematic arrangement is used to define the six Berets. The **Raspberry** Beret, which was designed first, uses eleven schematic sheets (included as the following eleven pages of this datasheet):

1. **Master_Raspberry** (the main datasheet that connects all others for the **Raspberry** Beret; see §5.1),
2. **Power** (defines the wiring of the various voltage regulators on the PCB; see §5.2),
3. **Motors** (defines the wiring of the DRV8912-Q1 motor drivers; see §5.3),
4. **Sensors** (defines the wiring of the IMU, magnetometer, and barometer; see §5.4),
5. **MCU** (defines the pinouts of the STM32, GPIO expander, Level Shifter, Flash, and OSC32; see §5.5),
6. **Connectors** (defines the wiring of most of the connectors on the PCB; see §5.6),
7. **Signal_Headers** (defines the wiring of the Signal Headers; see §5.6.2),
8. **UI** (i.e., User Interface, defines the wiring of the buttons and LEDs; see §5.6.8),
9. **Analog** (defines the wiring of the analog subsystem; see §5.7),
10. **Vmon** (defines the wiring of the voltage monitoring circuit; see §5.7.4), and
11. **Header_RPi** (defines the wiring to the RPi header; see §5.9.1).

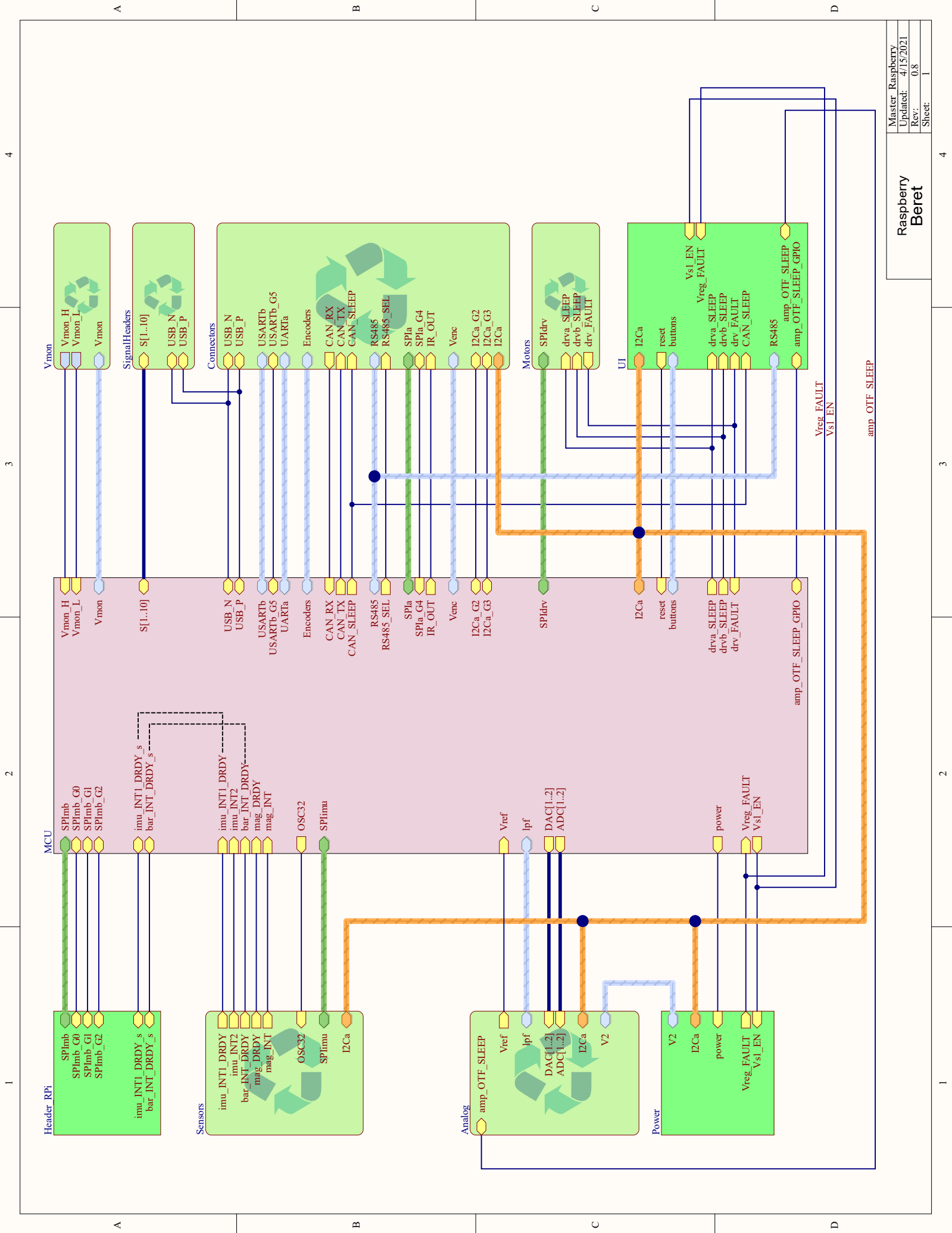
Note that the **Red** Beret is a **variant** of the **Raspberry** Beret, using the same PCB (and, thus, the same schematics), but with a lower-cost STM32G4, and several components flagged Do Not Populate (**DNP**). Eleven alternative schematic sheets are also defined, with different functionality implemented (generating Vmb = 12V instead of Vmb = 5V, etc), as listed here:

- | | | |
|---------------------------|-----------------------------------|-------------------------------|
| 1a. Master_Black , | 2b. Power_No_MB , | 10a. Vmon_No_Balance , |
| 1b. Master_White , | 5a. MCU_No_GPIO_Expander , | 11a. Header_96B , |
| 1c. Master_Green , | 6a. Connectors_Black , | 11b. Header_BB . |
| 1d. Master_Blue , | 6b. Connectors_Green , | |
| 2a. Power_12V_MB , | 6c. Connectors_Blue , | |

The other Berets are then defined using the following shared schematic sheet arrangement:

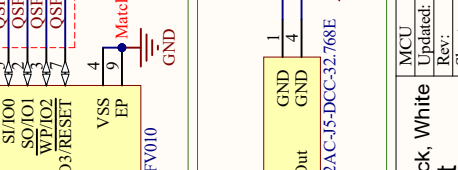
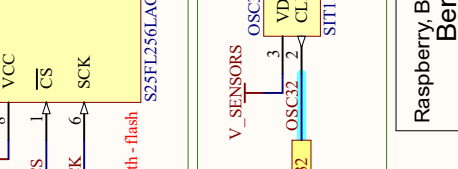
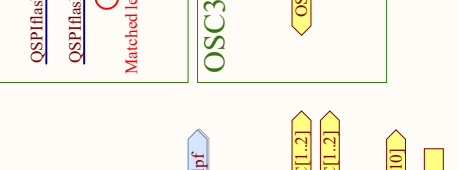
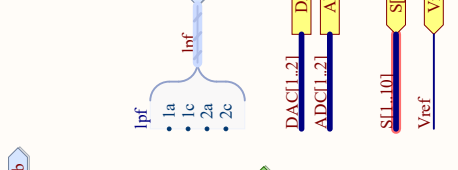
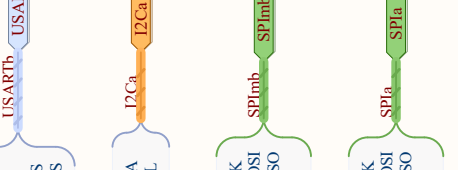
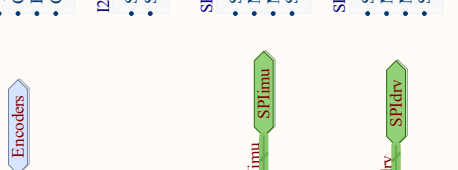
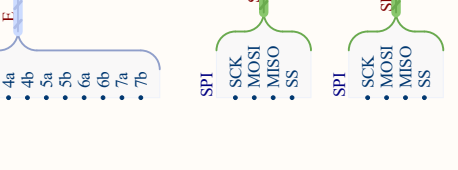
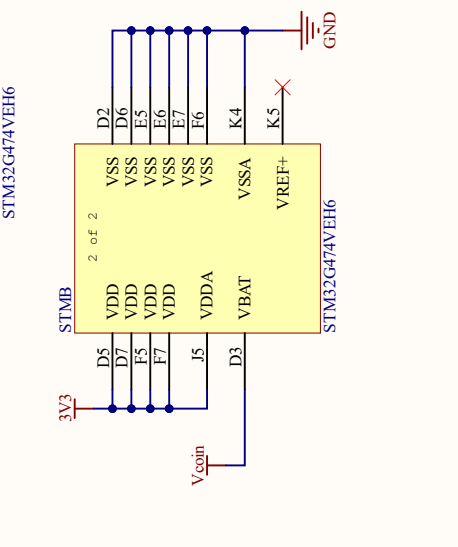
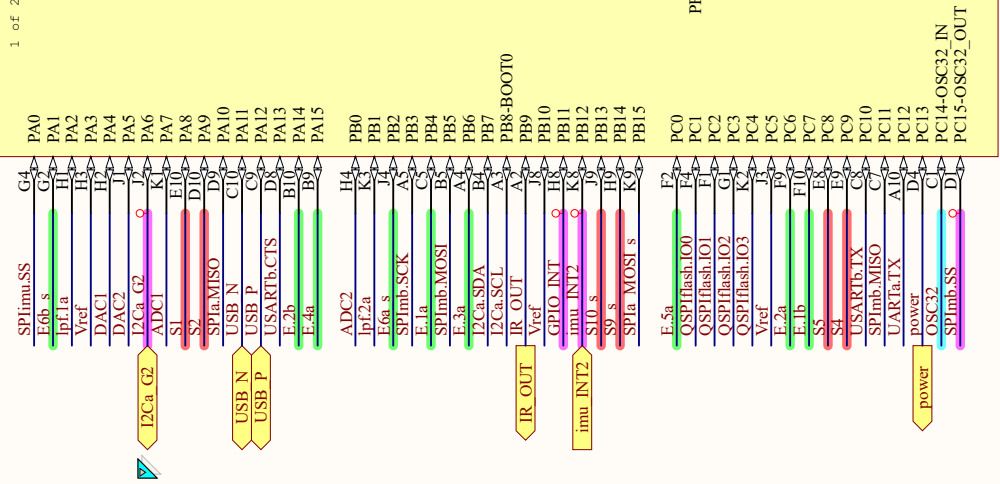
- Schematic sheets {1a, 2a, 3, 4, 5, 6a, 7, 8, 9, 10, 11a} define the **Black** Beret,
- Schematic sheets {1b, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11b} define the **White** Beret,
- Schematic sheets {1c, 2b, 4, 5a, 6b, 7, 8, 9, 10a} define the **Green** Beret.
- Schematic sheets {1d, 2b, 3, 4, 5a, 6c, 7, 8, 9, 10a} define the **Blue** Beret.

In this way, as the designs of the Berets are tweaked (changing resistor values in certain circuits, etc), the entire set of PCBs in the Beret family can more easily inherit all the updates made, and thus be kept in sync.

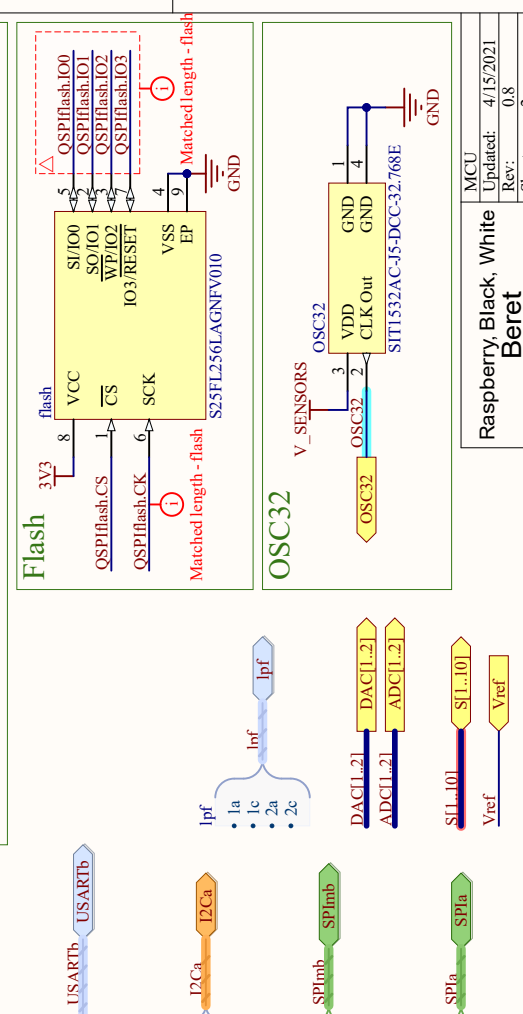
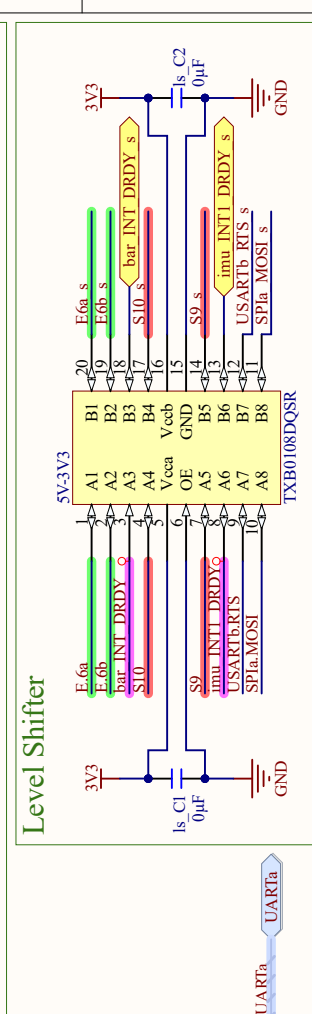
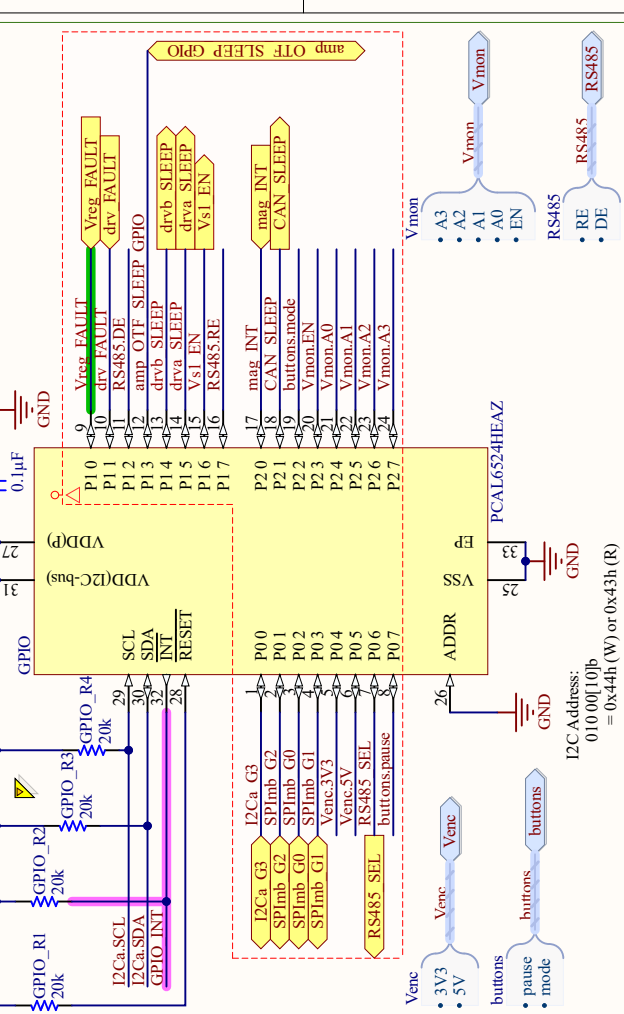


STM
STM32G474VEH6

STMA

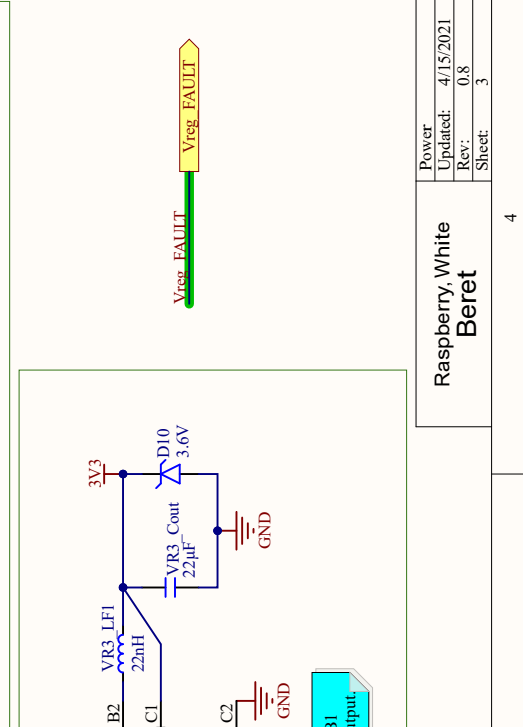
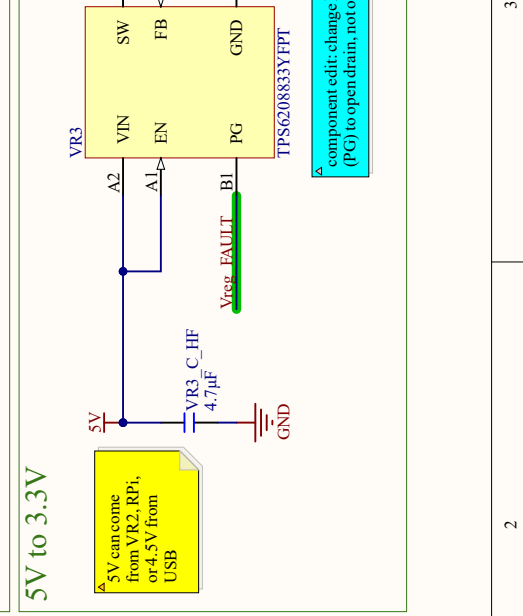
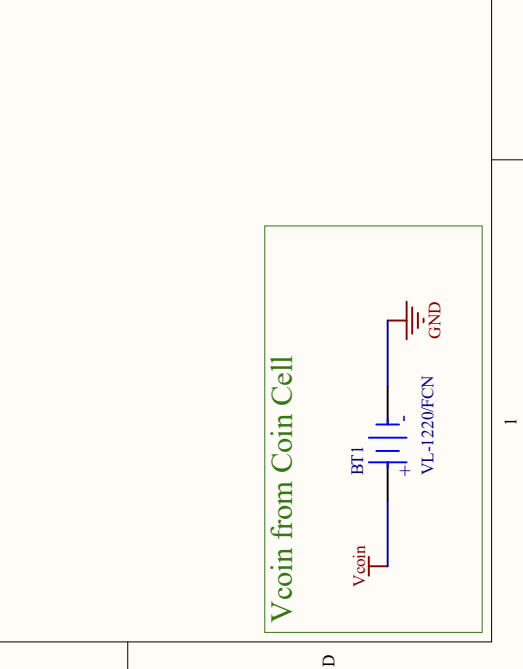
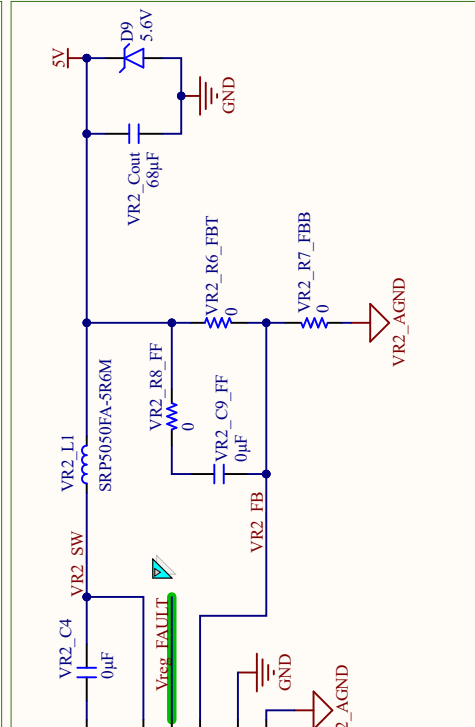
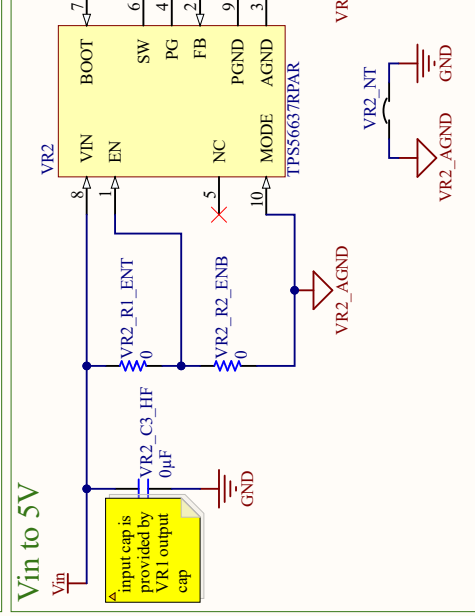
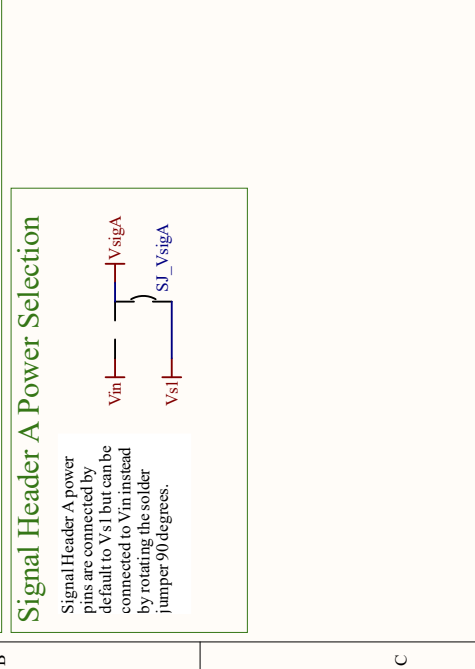
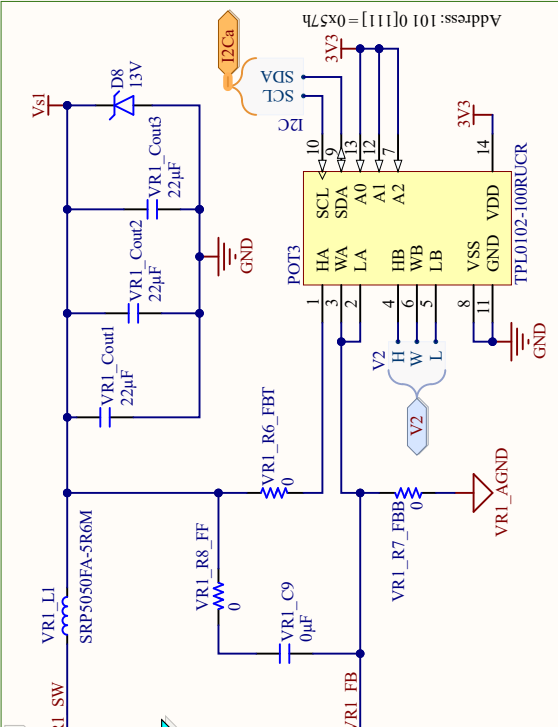
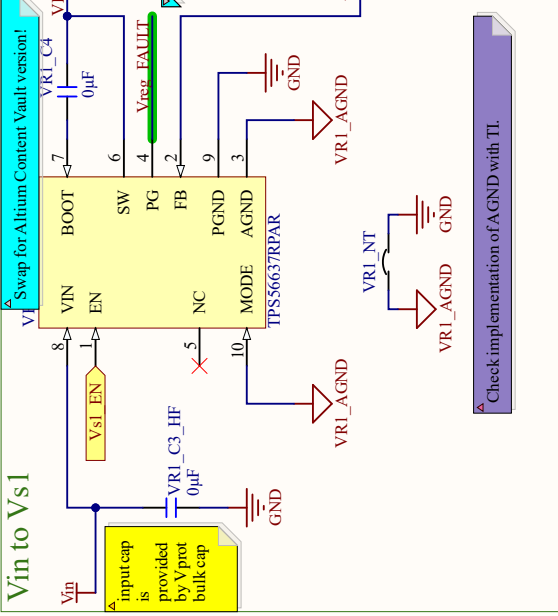
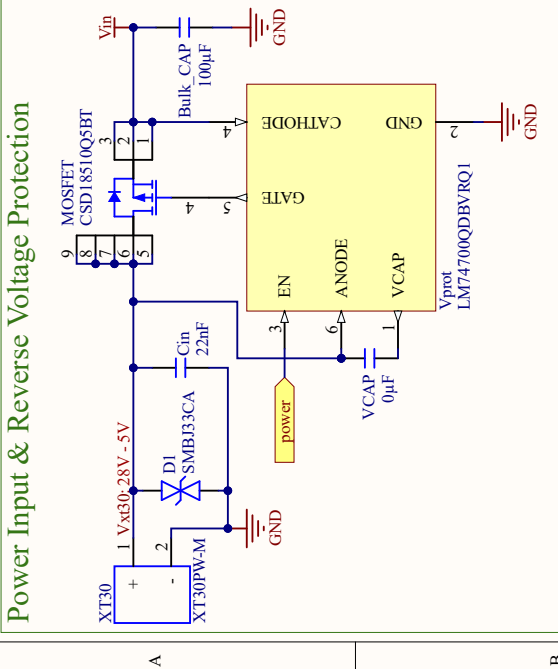


GPIO Expander

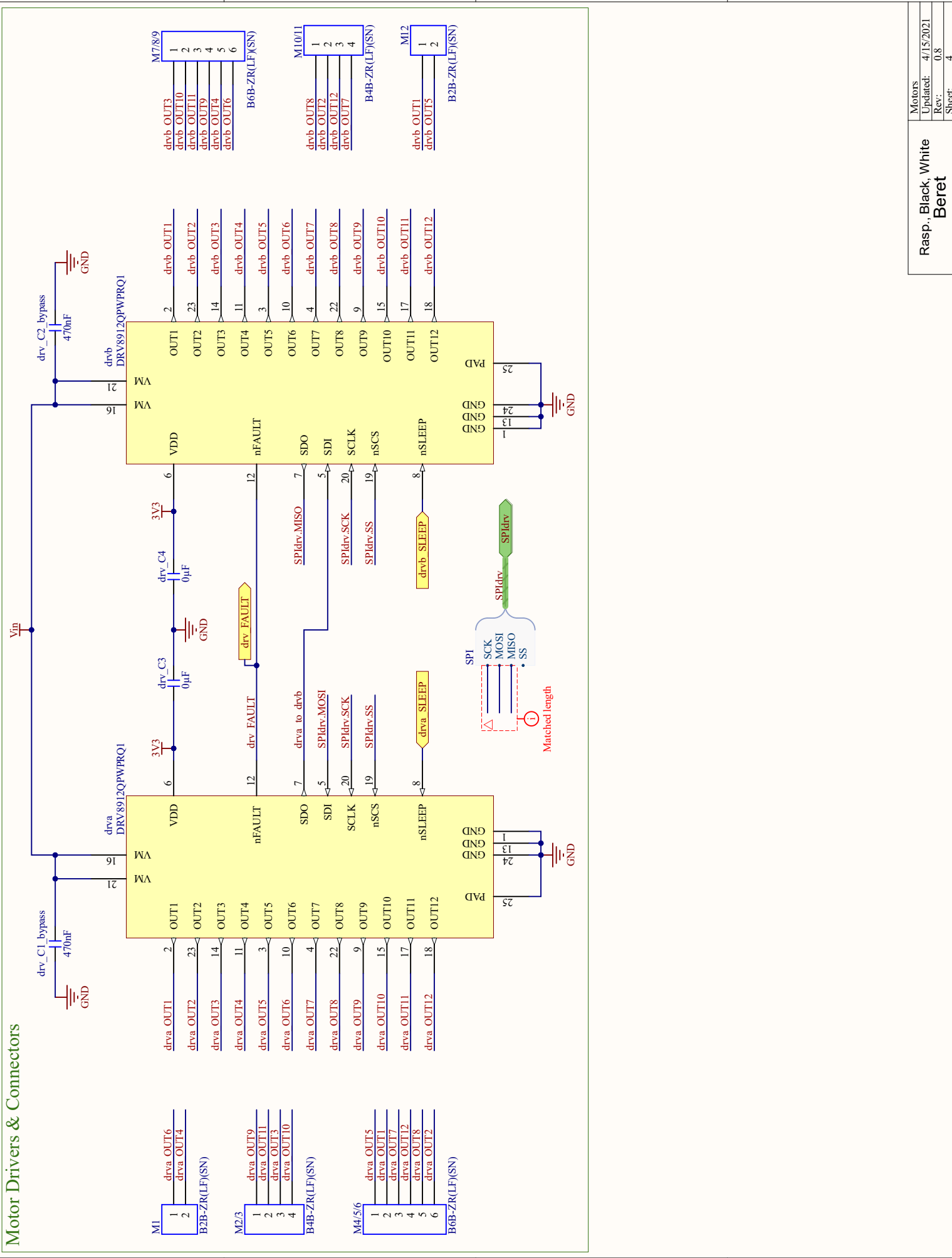


MCU
Updated: 4/15/2021
Rev.: 0.8
Sheet: 2

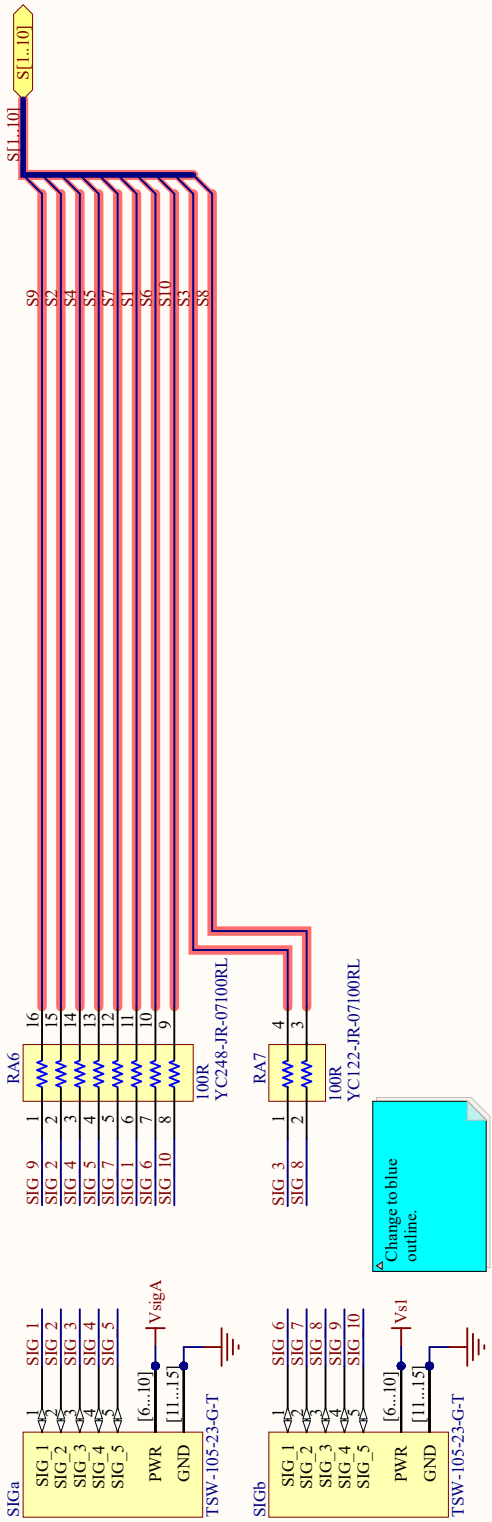
Raspberry, Black, White
Beret



Motor Drivers & Connectors

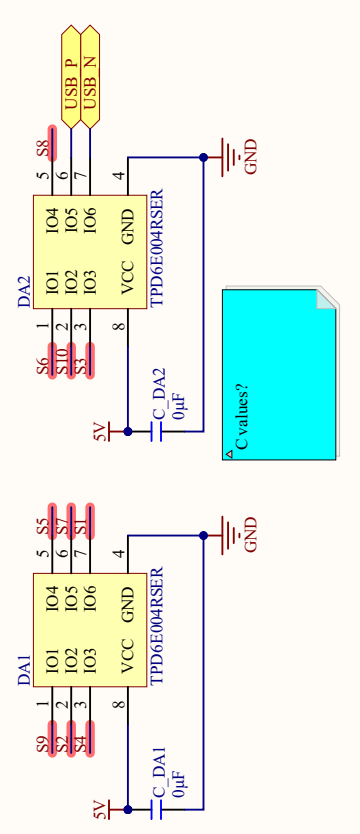


Signal Headers

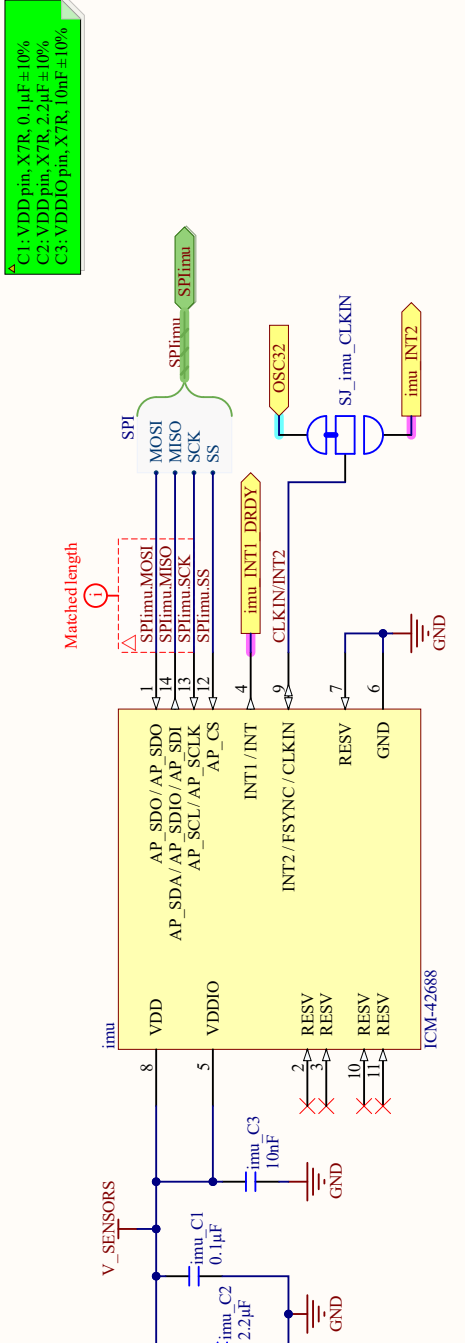


Change to blue outline.

ESD Protection (Signal Headers & USB)



IMU

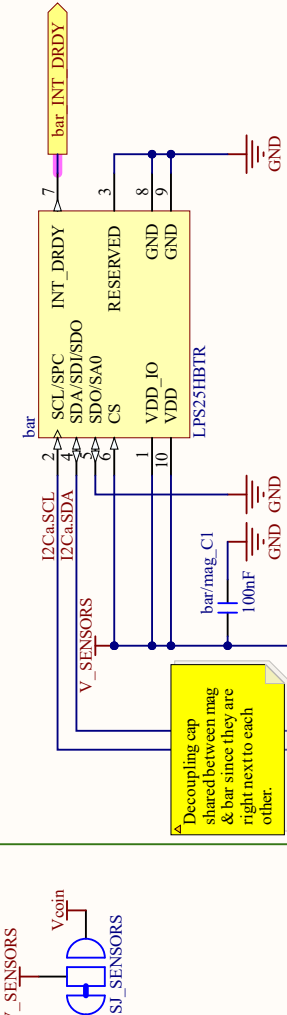


C1: 100nF power supply decoupling capacitor must be placed as near as possible to the supply pads (pin 10)

C2: 1µF low-frequency electrolytic capacitor Removed-accounted for in imu_C2 2.2µF cap

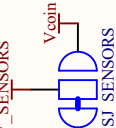
C3: 100nF - ceramic high-frequency decoupling capacitor as near as possible to the supply pin (pin 5) Removed-accounted for in bar/mag_C1

Barometer

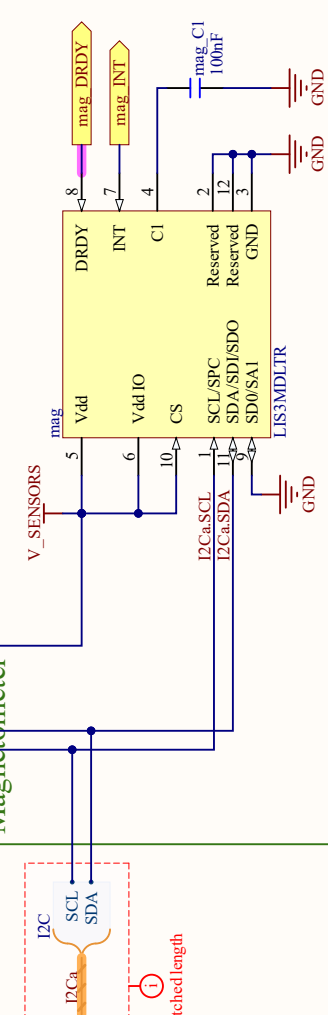


C1: 100nF power supply decoupling capacitor must be placed as near as possible to the supply pads (pin 10)

C2: 1µF low-frequency electrolytic capacitor Removed-accounted for in imu_C2 2.2µF cap



Magnetometer

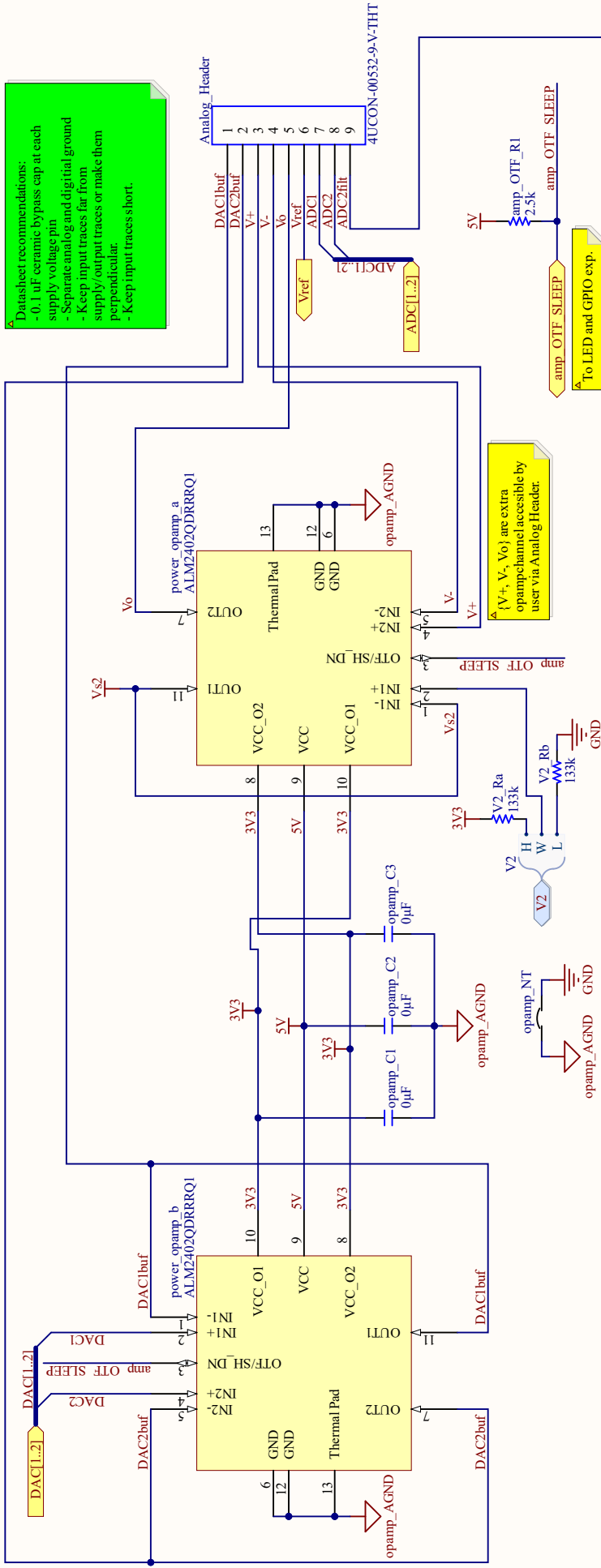


C1: 100nF power supply decoupling capacitor must be placed as near as possible to the supply pads (pin 10)

C2: 1µF low-frequency electrolytic capacitor Removed-accounted for in imu_C2 2.2µF cap

DAC Buffers

VS2 Buffer & Spare OpAmp

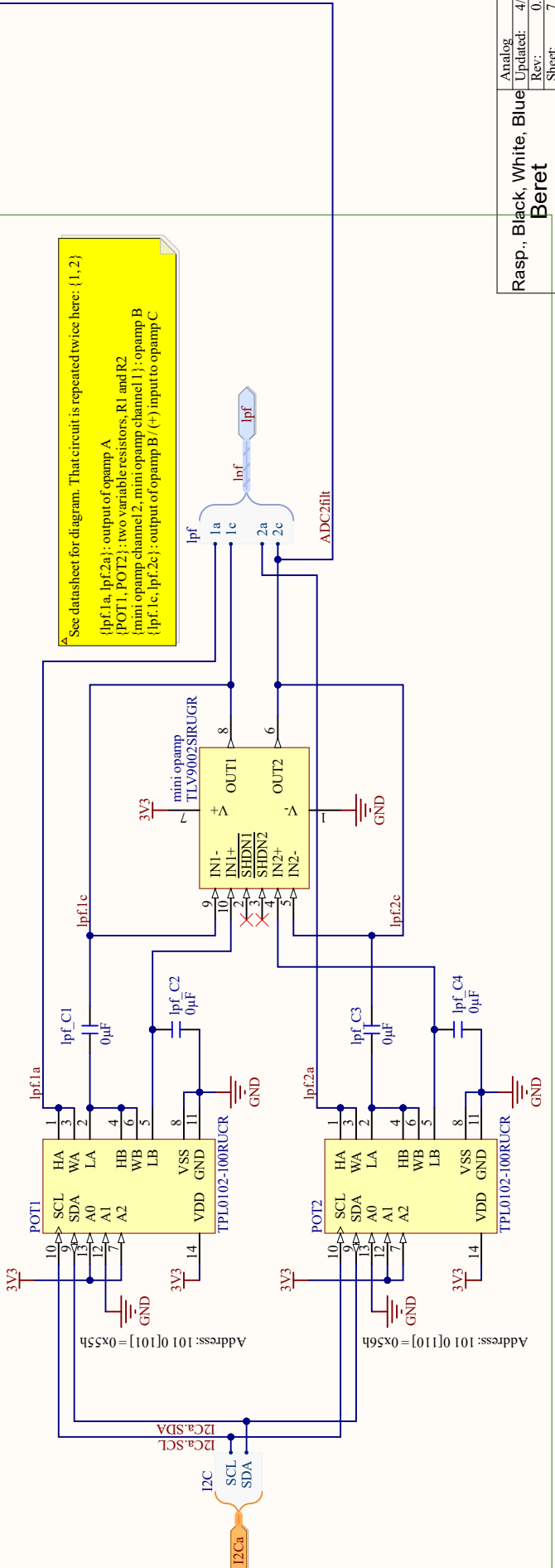


Datasheet recommendations:
 - 0.1 uF ceramic bypass cap at each supply voltage pin
 - Separate analog and digital ground
 - Keep input traces far from supply/output traces or make them perpendicular.
 - Keep input traces short.

V+, V-, Vo are extra opampchannel accessible by user via Analog Header.

To LED and GPIO exp.
 amp_OTF_SLEEP

Low Pass Filtering

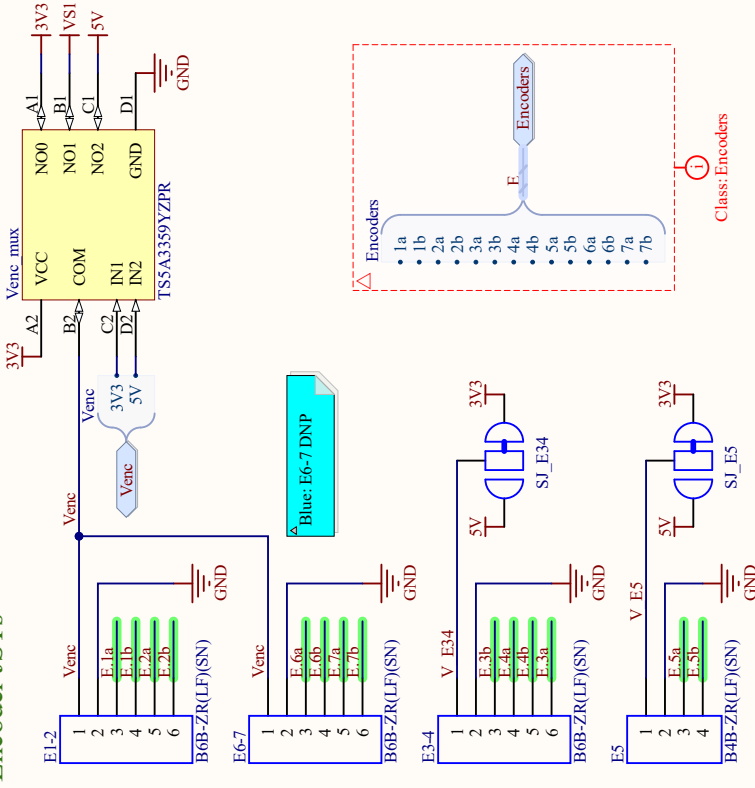


See datasheet for diagram. That circuit is repeated twice here: {1, 2}
 {lpf1a, lpf2a}; output of opamp A
 {POT1, POT2}; two variable resistors, R1 and R2
 {mini opamp channel 2, mini opamp channel 1}; opamp B
 {lpf1c, lpf2c}; output of opamp B / (+) input to opamp C

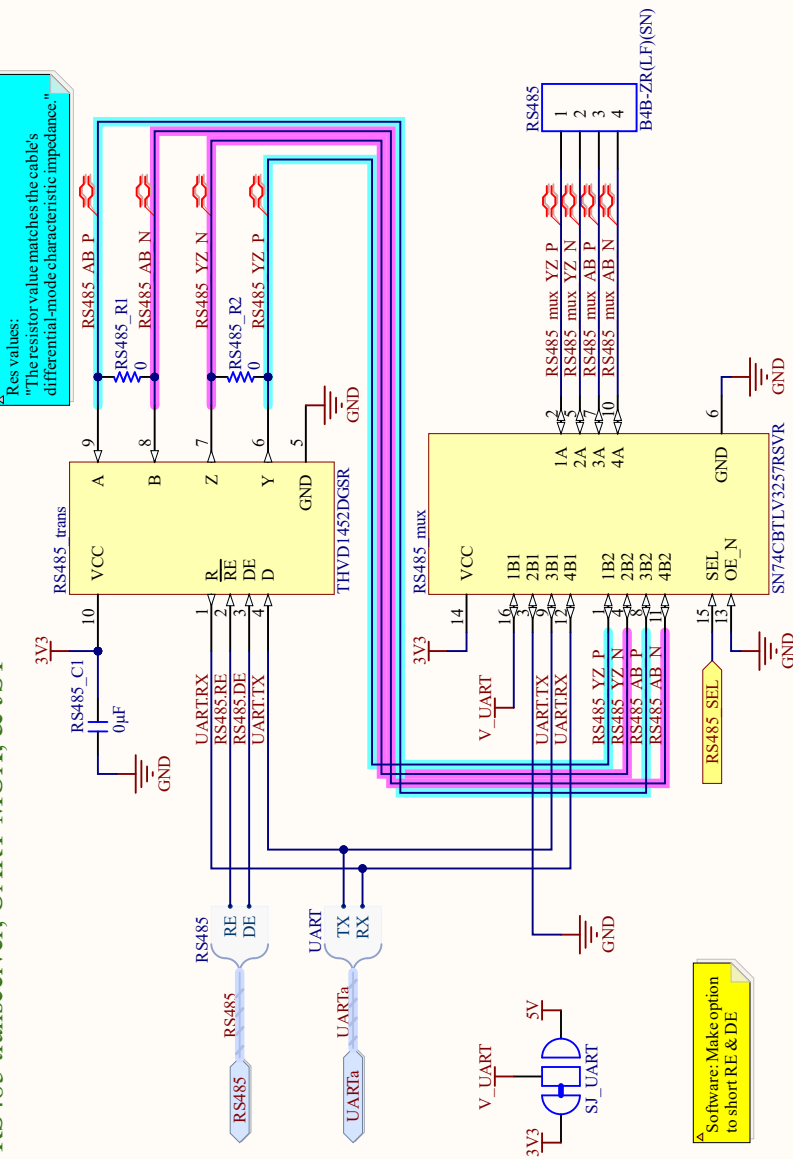
Address: 101 01101 = 0x55h

Address: 101 01101 = 0x56h

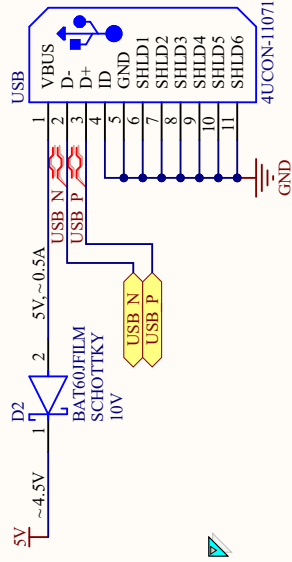
Encoder JSTs



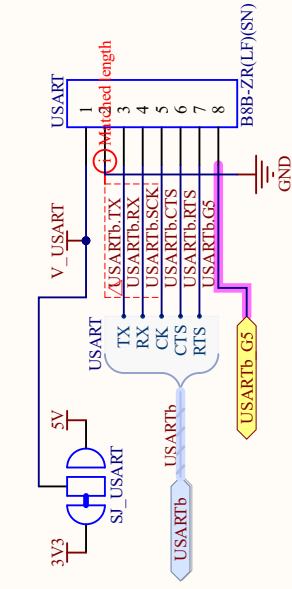
RS485 transceiver, UART MUX, & JST



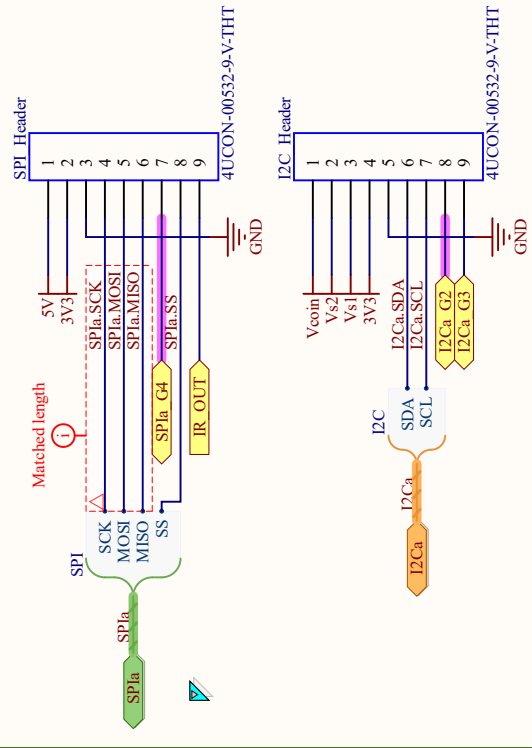
USB micro B



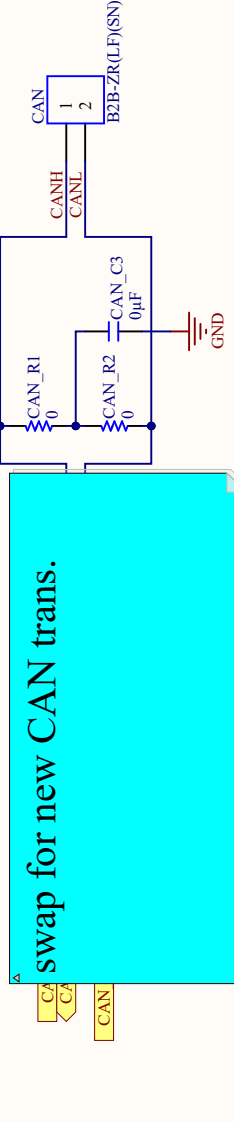
USART JST

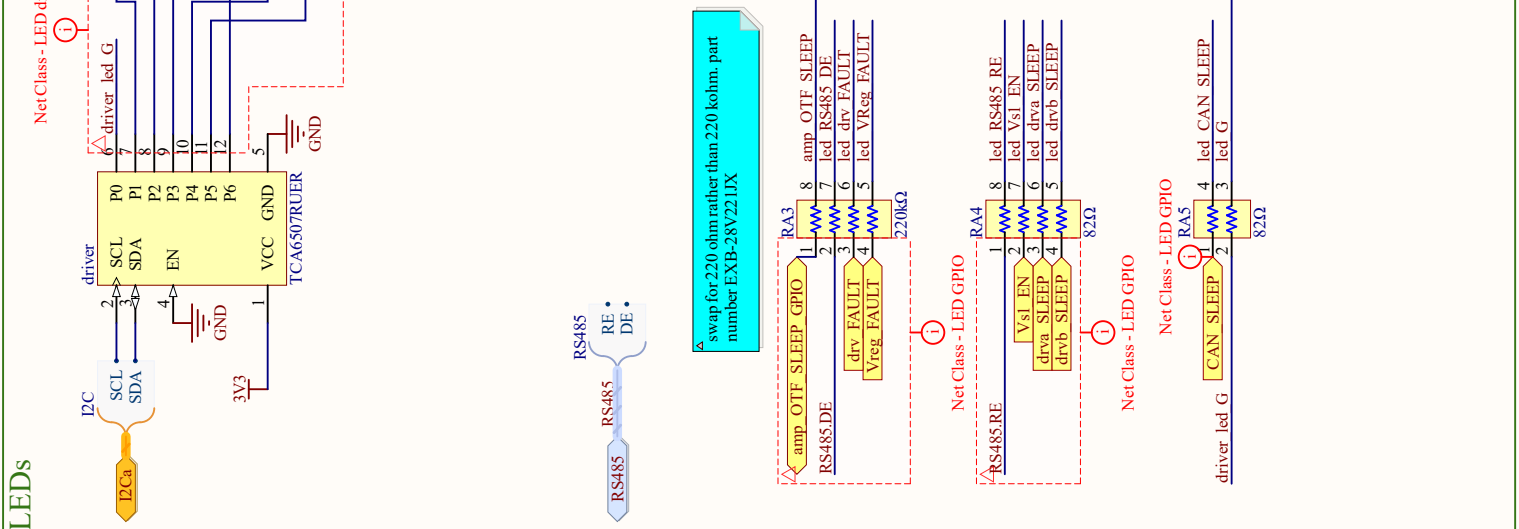
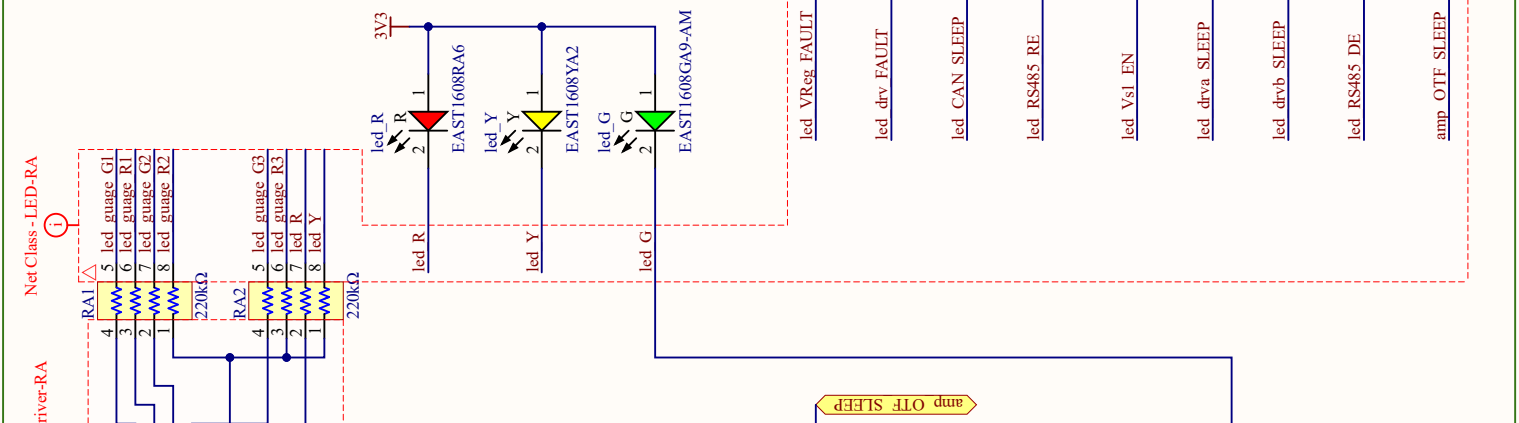
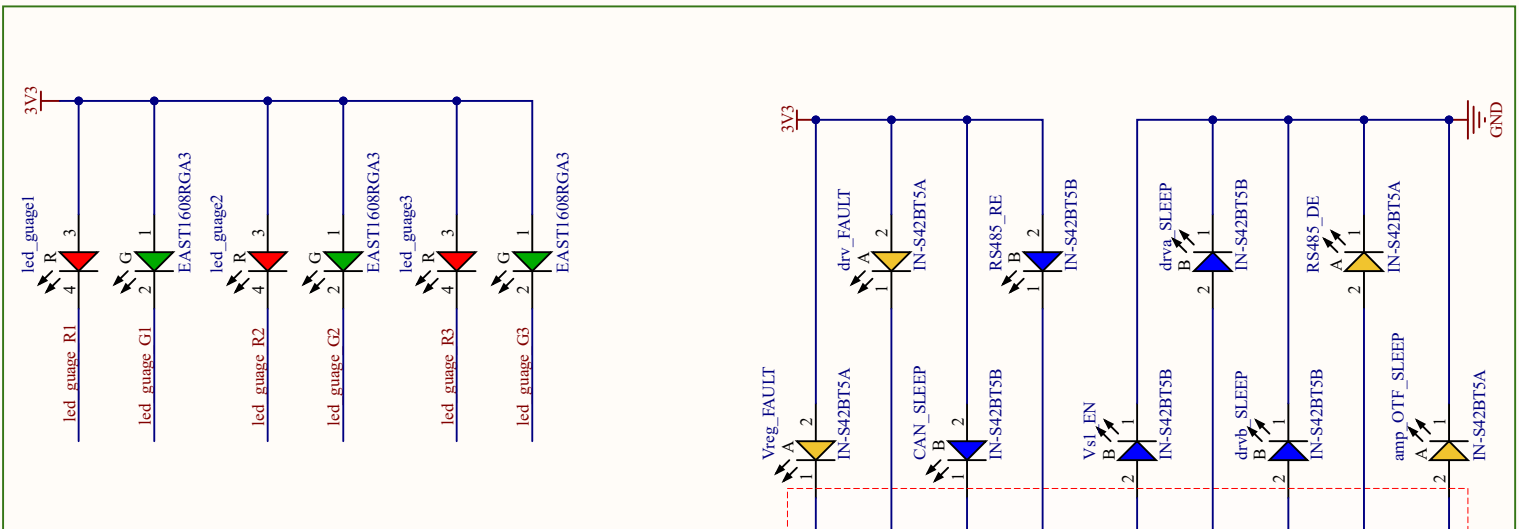
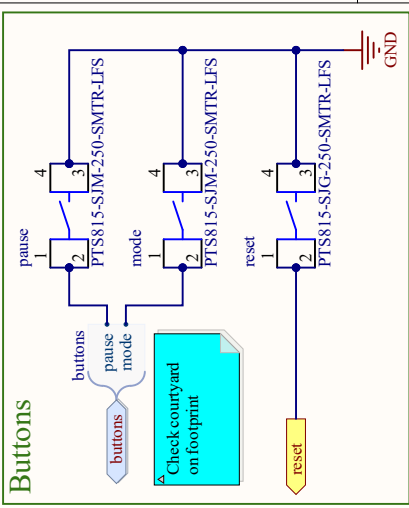


SPI & I2C Headers



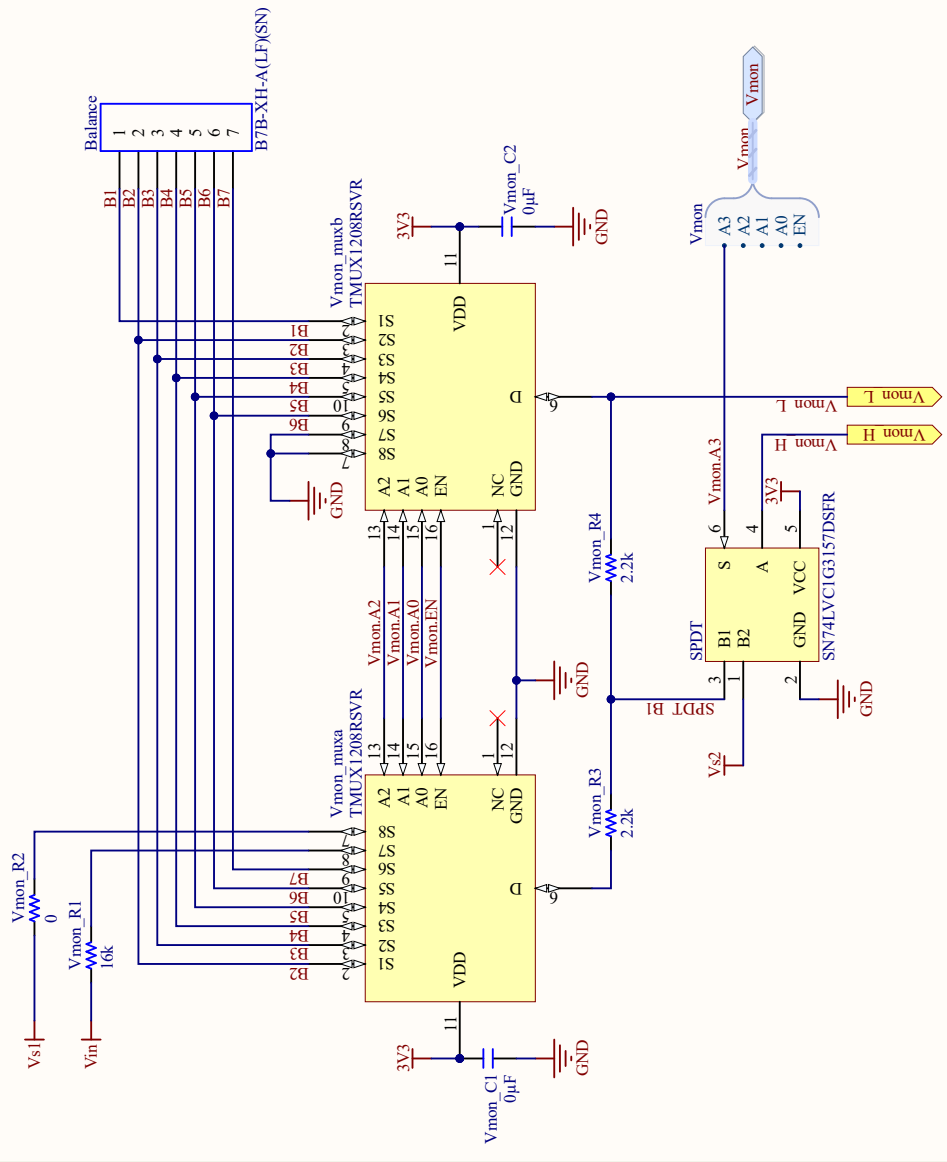
CAN transceiver & JST





⚠ Datasheet error: blue LED links to the 0603 but should link to 0402 at <https://www.digkey.com/en/products/detail/molux/IN-S42BT5B/0384754>

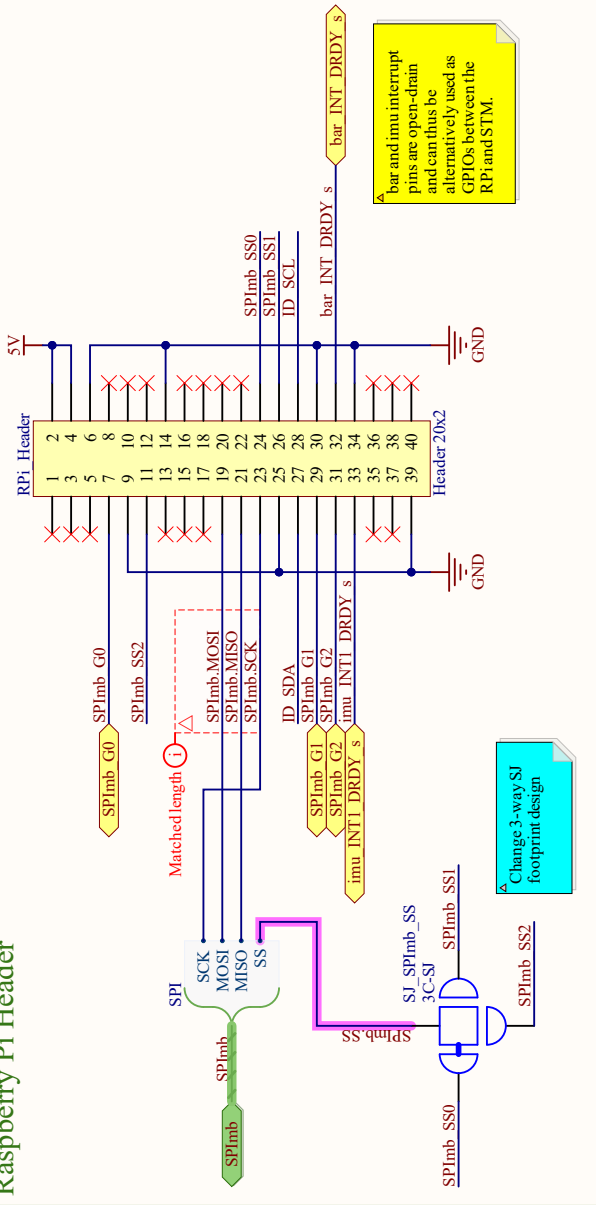
Voltage Monitoring (Vmon)



Voltage Monitoring	
Updated:	4/15/2021
Rev:	0.8
Sheet:	10

Rasp., Black, White
Beret

Raspberry Pi Header



EEPROM

