Chapter 4

Connector Standards

"The nice thing about standards is that you have so many to choose from." - Andrew Tanenbaum

Venerable connector standards like USB (e.g. Types A, C, and micro-B), HDMI (e.g. standard and micro, aka Types A and D) and Ethernet (RJ45) are both relatively compact and ubiquitous in modern laptop computers and SBCs; their use thus requires no further discussion here. Also well standardized are 3.5mm TRRS A/V jacks (albeit, with certain incompatibilities), and MIPI DSI display ports and CSI camera ports (though the MIPI specs are not themselves publicly released). However, as addressed in this chapter, other commonly-needed connectors for wiring together host and client devices are currently much less standardized, sometimes leading to device incompatibilities and often requiring fragile (and, easily misrouted) custom wire harnesses to address.

Contents

4.1 Existing I2C, SPI, and UART connector standards

Many attempts have been made to standardize powered wire harnesses for attaching various sensors and other "client" devices to "hosts" (e.g., SBCs) using various short-range comm protocols (see §3.2), prescribing both the connectors and the corresponding pin order to be used, including:

- standard I2C, with data and clock lines {SDA, SCL},
- extended I2C, adding {INT/SMBA, RES/SMBS} lines to standard I2C,
- SPI, with {MOSI, MISO, SCK, SS}, often with multiple SS (slave select) lines to support multiple devices,

- extended SPI, adding {INT, RES} lines to standard SPI,
- simplex or half duplex UART, using Tx or Rx only, or a single combined Tx/Rx line,
- full duplex UART, with separate transmit and receive lines {Tx, Rx},
- UART with Hardware Flow Control (HFC), adding {CTS, RTS} lines to avoid channel contention,
- (synchronous) USART, adding a clock line SCK to UART to synchronize the receiver and transmitter, and
- other analog or digital signals, such as GPIOs, PWMs, encoder signals, clocks, etc.

Several manufacturers have proposed standardized powered[1] connection protocols to address this need, and marketed a variety of host and client devices mounted on small PCBs using the proposed protocols, including:

- PMOD, a standard by Digilent for 6- and 12-wire harnesses mated with 0.1” pin headers, including:
  - for I2C channels, a 1x6 connector with pin order[2,3] {INT, RESET, SCL, SDA, GND, Vcc},
  - for UART channels, a 1x6 connector with pin order {CTS, Tx, Rx, RTS, GND, Vcc},
  - for SPI channels, a 1x6 connector with pin order {SS, MOSI, MISO, SCK, GND, Vcc},
  - etc. (a handful of other 1x6 and 2x6 connectors are also defined; see the PMOD spec for details);
- Grove, a standard by Seeed for 4-wire harnesses with 2 mm pitch proprietary connectors (see here) with:
  - for I2C connections, a pin order[4] of {SCL, SDA, Vcc, GND},
  - for UART connections, a pin order of {Rx, Tx, Vcc, GND},
  - for other digital and PWM-driven devices, a pin order denoted {D0, D1, Vcc, GND}, and
  - for analog devices, a pin order denoted {A0, A1, Vcc, GND};
- STEMMA, a standard[5] by Adafruit for 3- and 4-wire harnesses with 2 mm pitch JST-PH connectors, with:
  - 4-pin connectors, designed for I2C only, with pin order {SCL, SDA, Vcc, GND}, and
  - 3-pin connectors, designed for analog, digital, and PWM-driven devices, with pin order {GND, Vcc, Signal};
- Gravity, a standard by DFRobot for 3- and 4-wire harnesses with 2 mm pitch JST-PH connectors, with:
  - for I2C or UART channels, a 4-pin connector with pin order[6] {Vcc, GND, SCL, SDA} or {Vcc, GND, Rx, Tx},
  - for analog, digital, and PWM-driven devices, a 3-pin connector with pin order {GND, Vcc, Signal};
- Qwiic, a standard by SparkFun for 4-wire harnesses[7] with 1 mm pitch JST-SH connectors, with:
  - 4-pin connectors, designed for I2C only, with pin order {GND, Vcc, SDA, SCL};
- STEMMA-QT, a standard by Adafruit for 4-wire I2C harnesses with JST-SH connectors compatible with Qwiic.

Also noteworthy with regard to connector standardization (or, the glaring lack thereof...) in the industry are:

- the several 1 mm pitch JST-SH connectors used by the Beaglebone Blue, including:
  - for I2C channels, a 4-pin connector with pin order {GND, 3.3V, SCL, SDA},
  - for UART channels, a 4-pin connector with pin order {GND, 3.3V, Rx, Tx},
  - for SPI channels, a 6-pin connector with pin order {GND, 3.3V, MOSI, MISO, SCK, SS},
  - etc. (a handful of other JST-SH connectors are also incorporated; see, e.g., here for pin order), and
- the 3-pin 1.5 mm pitch JST-ZH connector[8], with pin order {3.3V, GND, Rx}, used by the DSM radio receivers.

[1] Most devices that implement these standards require 3.3V PWR, some use 5V, and some can use either. Check the specs!
[2] Many PMOD I2C connectors on clients are 2x6, with identical columns, facilitating easy daisy-chain wiring of I2C devices.
[3] An older PMOD spec excluded the {INT, RESET} lines from I2C wire harnesses, using 1x4 (or, by footnote 2, 2x4) connectors.
[4] Note that Seeed sells branch cables to facilitate multiple I2C devices hooked to a single I2C channel.
[5] 4-pin STEMMA and Grove I2C devices are interoperable, and (if powered by 3.3V) STEMMA-QT and Qwiic are interoperable.
[6] Despite published claims to the contrary (1, 2), 4-pin Gravity (with Vcc on pin 1) is not pin compatible with Grove or STEMMA.
[7] SparkFun’s Qwiic modules each incorporate a pair of 4-pin JST-SH connectors to facilitate easy daisy-chain wiring of I2C devices.
[8] This 3-pin JST-ZH connector is also included on the BeagleBone Blue, as well as several flight control boards meant for UAVs.

4.2 Recon: an extensible JST-ZH powered connector standard

Digilent's 0.1" pitch PMOD, Seeed's 2 mm pitch Grove, DFRobot's 2 mm pitch Gravity, Adafruit's 2 mm pitch STEMMA and 1 mm pitch STEMMA-QT, and SparkFun's 1 mm pitch Qwiic standards all have their pros and cons. The substantial benefit that they share is the large catalog (from each respective manufacturer) of ready-to-use devices, preassembled on PCBs incorporating the necessary passives, and sold with suitable wiring harnesses and connectors. However 0.1" (2.54 mm) pitch pin headers and 2 mm pitch JST-PH (and similar) connectors are unnecessarily large when considering the current requirements of most devices in these catalogs (connector size becomes an essential limiting factor when designing space-constrained logic boards), whereas 1 mm pitch JST-SH, 1.25 mm pitch JST-GH, and similar connectors are only available as SMD, which are fragile (these connectors often rip off a host PCB if used extensively). Further, the general lack of flexibility in existing standards, in terms of optional additional pins, presents a significant downside for many applications.

The smallest broadly-available, low-cost, 1A-rated connector standard with durable PTH shrouded headers for mounting on a PCB is the 1.5 mm pitch JST-ZH standard, connectors for which are nonreversible (as with all JST standards, but not with bare 0.1" pitch pin headers), as the pins are displaced from the centerline of the connector shroud. Conveniently, JST-ZH wire housings with M pins can also fit into JST-ZH shrouded headers (on a PCB) with N pins so long as $M \leq N$. This leads to the possibility of creating a uniquely *extensible* standard using this type of connector that, for each comm protocol, picks up Vcc and GND on the first 2 pins, then all essential pins for a given comm protocol, followed by a flexible number of optional pins for that comm protocol, all in a predefined order. Following this approach, hosts may be used to drive clients directly using a given comm protocol following this new standard (without incorporating custom wire harnesses that reorder the pins) so long as the shrouded header on the host incorporates at least as many optional pins as the wire harness from the client. Such hosts may include multiple connectors of a given comm protocol (like UART), some with fewer optional pins and some with more, to more efficiently support a rich variety of auxiliary devices.

The new extensible JST-ZH based open connector standard proposed here is dubbed **Recon**, and comes in four main types (underlined pins are required, non-underlined are optional):

$$
\begin{array}{lcccccccccc}
\text{pin \#} \rightarrow & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \ldots \\
\textbf{Recon Basic} & [\,\underline{\text{Vcc}}, & \underline{\text{GND}}, & \text{S0}, & \text{S1}, & \text{S2}, & \text{S3}, & \text{S4}, & \text{S5}, & \text{S6}, & \ldots] \\
\textbf{Recon I2C} & [\,\underline{\text{Vcc}}, & \underline{\text{GND}}, & \underline{\text{SDA}}, & \underline{\text{SCL}}, & \text{INT/G2}, & \text{RES/G3}, & \text{G4}, & \text{G5}, & \text{G6}, & \ldots] \\
\textbf{Recon SPI} & [\,\underline{\text{Vcc}}, & \underline{\text{GND}}, & \underline{\text{MOSI}}, & \underline{\text{MISO}}, & \underline{\text{SCK}}, & \underline{\text{SSa}}, & \text{INT/SSb}, & \text{RES/SSc}, & \text{SSd}, & \ldots] \\
\textbf{Recon UART-T} & [\,\underline{\text{Vcc}}, & \underline{\text{GND}}, & \underline{\text{Tx}}, & \text{Rx/G1}, & \text{SCK/Vbat/G2}, & \text{CTS/G3}, & \text{RTS/G4}, & \text{G5}, & \text{G6}, & \ldots] \\
\textbf{Recon UART-R} & [\,\underline{\text{Vcc}}, & \underline{\text{GND}}, & \underline{\text{Rx}}, & \text{Tx/G1}, & \text{SCK/Vbat/G2}, & \text{RTS/G3}, & \text{CTS/G4}, & \text{G5}, & \text{G6}, & \ldots]
\end{array}
\tag{4.1}
$$

To accelerate the adoption of the Recon standard, ready-made wire harnesses that convert directly from Recon hosts (like the Berets discussed in §5) to PMOD, Grove/STEMMA, Gravity, and Qwiic/STEMMA-QT clients are available, thus enabling such hosts to connect directly (without requiring user-made custom wire harnesses) to *all* of the large catalogs of available client devices incorporating these current competing standards.

The most common voltage used by currently-available client devices is 3.3V; many 5V devices, and an increasing number of 1.8V devices, are also available. The Recon standard thus requires that hosts provide Vcc = 3.3V, and use 3.3V TTL logic, by default on all I2C, UART, and SPI connectors. Other voltages (5V and 1.8V in particular) may be selectable on individual connectors on the host, in order to support an even larger range of client devices. If Vcc = 5V is selectable on a given connector, its (3.3V TTL) digital pins must simply be 5V tolerant. If, Vcc = 1.8V is selectable on a given connector, on the other hand, all of its digital signals must be level shifted (on the host) to the value of Vcc selected (using, e.g., a TI TxB0108 level shifter).

We now discuss some details related to each of the four main types of Recon connectors.

4.2.1 Recon Basic and its variants

The simplest Recon connector is a 2-pin power connector, $\{\text{Vcc}, \text{GND}\}$. From this starting point, the Recon Basic standard shares power and a set of one or more generic numbered signals, denoted $\{S0, S1, S2, S3, \dots\}$, which is useful for signals that do not follow one of the three main short-range digital comm protocols {I2C, UART, SPI} discussed in the following three subsections. For signals that are intended for more specific purposes, different one- or two-character identifiers, plus a sequencing number, may be used; for example, instead of using the name **Basic** and the generic signal names $\{S0, S1, S2, S3, \dots\}$, one may substitute as follows:

- **GPIO**s may be denoted $\{G0, G1, G2, G3, \dots\}$,
- **PWM** based signals (usually, as outputs from the host) may be denoted $\{P0, P1, P2, P3, \dots\}$,
- **Encoder** signals (usually, as inputs to the host) may be denoted[9] $\{E0a, E0b, E1a, E1b, \dots\}$,
- **Clock** signals for general-purpose applications may be denoted $\{CK0, CK1, CK2, \dots\}$,
- **Analog** signals may be denoted $\{A0, A1, A2, A3, \dots\}$, and
- **Digital** comm signals not following the I2C, UART, or SPI standards may be denoted $\{D0, D1, D2, \dots\}$,

thus defining the **Recon GPIO**, **Recon PWM**, **Recon Enccoder**, **Recon Clock**, **Recon Analog**, and **Recon Digital**[10] variants of the **Recon Basic** standard. Other specific signal names and enumerations may also be proposed and used when necessary, if appropriately documented in the corresponding device datasheet; the Recon Basic standard itself is meant to be extensible and flexible[11].

GPIOs may be appended to the Recon Basic variants described above, and to the Recon I2C, SPI, and UART standards discussed below; e.g., the connector [Vcc, GND, CK0, CK1, CK2, G3] fully conforms to the Recon Clock spec. A logical numbering for these optional GPIOs, consistent with the Recon Basic spec, is proposed in (4.1); other short/descriptive names for these GPIOs, appropriately documented, should instead be used on clients to identify the functions of the GPIOs that they include, noting that the Recon spec calls for all required and (if included) optional signals listed in (4.1) to remain in the order specified for the corresponding connector.

4.2.2 Recon I2C

Perhaps the most widely adopted standard for low-speed short-distance serial communication between a host (aka "master") and multiple clients (aka "slaves") is I2C. The I2C standard facilitates half-duplex[12] communication rates up to 400 kbps, with extensions to 1 Mbps and, via additional logic and clock stretching, to 3.4 Mbps. Standard I2C requires just two digital signals, data and clock $\{\text{SDA}, \text{SCL}\}$ (pins 3 and 4 of the Recon I2C standard). An I2C master may communicate individually with up to 112 slaves at addresses x08 to x77 using simple 7-bit device addresses (or up to 1024 slaves, at addresses x000 to x3FF, using 10-bit device addresses). Traditionally, communication via standard I2C requires all transmissions to be initiated by a single master; however, some newer I2C devices implement a multimaster protocol in which different devices (each of which implement the multimaster protocol) can take over the master role on a single I2C bus at different times. Common extensions of the I2C standard add the following (optional pins 5 and 6 of the Recon I2C standard):

- INT, an active low open drain output from the slaves(s) meant to alert the master of new data to report, and

[9] Most modern encoders are quadrature encoders, the outputs of which are attached to the host a pair at a time in order to discern both the speed and direction of rotation of the shaft to which they are attached. For clarity, such pins should thus be enumerated a pair at a time; this modified enumeration of the signals of the Recon Basic standard should not present any confusion.

[10] Other digital comm approaches are typically bit-banged from the MCU, which requires the host CPU to manage; the advantage of the I2C, UART, and SPI standards, and their common extensions, is that they may typically be handled by dedicated subunits on modern MCUs, offloading the computational burden of controlling these channels from the available CPU core(s) on the host.

[11] Note that, e.g., H-bridge outputs for driving brushed DC motors and steppers are generally not considered to be part of the Recon standard, as they are not "powered" connectors with $\{\text{Vcc}, \text{GND}\}$.

[12] Half duplex means that communication in one direction at a time only is allowed.

- RES, an active low "reset" or "suspend" output from the master meant both to drive the slaves(s) into a low-power "sleep" state if available, and to re-initialize certain settings on the slaves(s) once released.

The SMBus and related PMBus standards are based closely on I2C; all three types of devices may generally be mixed on a single bus. Amongst other refinements, SMBus standardizes the behavior of the optional INT (aka SMBALERT#) and RES (aka SMBSUS#) pins in a useful way; if these standardized behaviors on the optional INT and/or RES pins are available on a given host or client, they are (for brevity) to be denoted in the Recon I2C standard as SMBA and SMBS, respectively, on the corresponding device.

The newer I3C standard might well reshape how hosts communicate with multiple low-power clients in the coming decade. I3C is also based on the I2C standard, and is compatible with older I2C devices, while allowing much faster comm with other I3C devices over the same {SDA, SCL} pins. The I3C standard facilitates communication rates up to 12.5 Mbps, with extensions to 33 Mbps. One of the new features of I3C is in-band interrupts, which provide an efficient way for slaves to alert the host of new data to report without using a separate INT/SMBA pin, or swapping out which device plays the role of master (the logic of which can get complicated). The Recon I2C connector standard is, of course, compatible with I3C; if/when I3C becomes widely adopted, the name of the Recon I2C standard might well need to be updated to reflect this compatibility.

4.2.3 Recon SPI

Another common protocol for short-distance serial communication between one master and multiple slaves is SPI. The SPI approach, which does not have any formal standard, facilitates fast full-duplex[13] synchronous[14] communication, often at rates exceeding 10 Mbps. Typical SPI implementations (aka "4-wire SPI") use four signals: master-out-slave-in, master-in-slave-out, serial clock, and slave select, denoted {MOSI, MISO, SCK, SS} (all four signals are required on SPI hosts by the Recon SPI standard), where SS is active low. When multiple slaves are attached to a single SPI master, a different slave select signal {SSa, SSb, SSc, . . . } is connected to each attached device; custom wire harnesses are thus generally required.

A few common simplifications of standard SPI (full-duplex, with {MOSI, MISO, SCK, SS} signals) exist:

- In simplex[15] mode, either the MOSI or the MISO wire is simply dropped.

- In half-duplex (aka "3-wire SPI") mode, a single SDIO signal is used for both input and output at different times[16]. On some MCUs, 3-wire SPI mode can simply be selected in software when needed (i.e., to communicate with a 3-wire SPI slave), making SDIO available directly on the host's SPI MOSI pin. Selecting this feature in software on a host allows both 3-wire SPI comm to certain slaves at some times, and 4-wire SPI comm to other slaves at other times. On other MCUs, to facilitate 3-wire SPI, the MOSI pin on the host must be connected via a resistor (on the PCB, likely as a DNP[17]) to the MISO pin, and the modified MISO line subsequently connected to the SDIO pin of the slave. Unfortunately, this hard-wired approach to generating combining MOSI and MISO on a host might interfere with the communications of other 4-wire SPI slaves on the same SPI channel.

Note that, even if there is only one slave device driven by a given SPI channel, the corresponding SS pin on the slave can usually not simply be tied off to GND and the SS signal eliminated, as state transitions on the SS pin are often (but not always) used by the slave to detect the beginning and end of each data transmission.

[13] Full duplex means that simultaneous communication in both directions is possible.

[14] Synchronous means that there is a shared serial clock signal from the master, denoted SCK, upon which both the transmit and receive signals at both ends of the comm channel are coordinated.

[15] Simplex means that communication in one direction only is possible.

[16] Warning: in 3-wire SPI, a resistor is generally needed somewhere along the communication path between the master and the slave, to prevent a possible (though, temporary) direct connection between a driven pin on the master and a driven pin on the slave at the opposite logic state, as the master and slave nodes are not generally synchronized.

[17] DNP means Do Not Populate, or Do Not Place, a given component during the board assembly process, but instead leave an open solder pad at this location, for a component to be added later by the user if desired.

As with I2C, communication via standard SPI requires all transmissions to be initiated by a single master[18]. Thus, common extensions of the SPI protocol add (software-controlled) INT and RES signals (optional pins 7 and 8 of the Recon SPI standard), the functionality of which is defined as for I2C channels (see §4.2.2).

4.2.4 Recon UART

A UART is a ubiquitous MCU subunit for asynchronous full-duplex short-distance serial communication, nominally point-to-point (between two devices). UART communication speed is configurable (and, measured on the fly at the opposite end of each wire, rather than being synced via a shared clock), with rates up to 5 Mbps realistically achievable, and 20 Mbps possible under ideal conditions, though many devices top out at 115.2 kbps or less. Unlike I2C and SPI, there is no concept of master or slave in UART; either device can initiate a transmission. Like I2C, standard UART requires just two digital signals, transmit and receive {Tx, Rx} (pins 3 and 4 of the Recon UART standard). Unlike I2C and SPI, the connection of these two signals need to be crossed between one end of the wire harness and the other (i.e., Tx connects to Rx, and Rx connects to Tx); this is accomplished in the Recon UART standard (4.1) by defining a UART-T pin order, usually implemented on hosts, and a UART-R pin order, usually implemented on clients, thus obviating the need for crossing wires within harnesses that connect UART-T connectors to UART-R connectors.

Notable extensions to the UART standard include the following.

- SCK (optional pin 5 of the Recon UART standard) is a serial clock signal used, in a modern yet still somewhat uncommon extension of UART dubbed USART, to synchronize the transmit and receive signals at both ends of the comm channel and thereby facilitate faster communication rates, as done in SPI (see §4.2.3). In fact, some flexible USART subunits on MCUs can also support an SPI (master or slave) mode of operation; the Recon UART pin order is designed specifically to support this.

- Vbat (optional use of pin 5 in the Recon UART spec) is a secondary (low-current) standby voltage source, which is required by some UART clients (e.g., GPS modules) for efficient operation[19]. If implemented, it is anticipated that Vbat would usually be made available on pin 5 of a Recon UART connector via a PCB solder jumper.

- {CTS, RTS} (optional pins 6 and 7 of the Recon UART spec) are used for Hardware Flow Control (HFC), which is today also somewhat uncommon in clients. The names and functions of these signals are derived from the (once-ubiquitous, but now mostly legacy) full RS232 standard[20]; note that {CTS, RTS} are crossed between the client and the host, like {Tx, Rx}, as again facilitated by the distinct UART-T and UART-R pin orders.

A few simplifications of standard UART (full-duplex, with {Tx, Rx} signals) are also quite common (and thus permitted on both hosts and clients by the Recon UART-T and UART-R connector standards):

- In simplex mode, either the Tx or the Rx wire from the host is simply dropped. Notable common examples

[18]Though rarely used, some hosts do implement a multimaster SPI protocol, though multimaster SPI is restricted to operate between two compatible devices only; unfortunately, this approach does not readily extend to SPI channels with additional slaves on it.

[19]To provide such standby power from the host to SPI or I2C clients, or to USART clients which make use of the SCK pin, while maintaining maximum flexibility and extensibility according to the Recon spec, it is recommended to use a separate 2-pin secondary power connector of the Recon Basic type (see §4.2.1).

[20]There are 6 control pins on the common DE9 connector used in this once-ubiquitous standard: {CD, CTS, RTS, DSR, DTR, RI}, standing for Carrier Detect, Clear To Send, Request To Send, Data Set Ready, Data Terminal Ready, Ring Indicator. Of these, only {CTS, RTS} are still in significant use today. If the need arises to support all 6 of the control signals on DE9 connectors (primarily, to support legacy equipment), the Recon UART-T and UART-R standards may be augmented as follows (with typical outputs, inputs specified, noting that DTE originally stood for Data Terminal Equipment, and DCE stood for Data Circuit-terminating Equipment):

$$\begin{array}{llr} \textbf{Recon RS232-T} & [\,\underline{\mathrm{Vcc}},\ \underline{\mathrm{GND}},\ \underline{\mathrm{Tx}},\ \mathrm{Rx},\ \mathrm{CD},\ \mathrm{CTS},\ \mathrm{RTS},\ \mathrm{DSR},\ \mathrm{DTR},\ \mathrm{RI},\ \ldots\,] & \leftarrow \text{host (aka DTE)} \\ \textbf{Recon RS232-R} & [\,\underline{\mathrm{Vcc}},\ \underline{\mathrm{GND}},\ \underline{\mathrm{Rx}},\ \mathrm{Tx},\ \mathrm{CD},\ \mathrm{RTS},\ \mathrm{CTS},\ \mathrm{DTR},\ \mathrm{DSR},\ \mathrm{RI},\ \ldots\,] & \leftarrow \text{client (aka DCE)} \end{array} \quad (4.2)$$

include seven segment display drivers, which are transmit only from the host, and DSM receivers, which are receive only at the host (e.g., a mobile robot or drone).

- In half-duplex (aka "single-wire" or "1-Wire") mode, a single signal is again used for both input and output. On some MCUs, this mode (on the Tx line) can simply be selected in software when needed. Usually, a pull-up resistor is needed somewhere along this single wire; to facilitate this mode, it is thus suggested that DNP pads be left for such a pull-up resistor on the host. If a hardware single-wire mode is not available in the UART subunit on the host MCU, and the UART transmit module is (or can be configured as) open drain, the Tx and Rx pins may simply be connected to enable single-wire functionality. Unfortunately, most UART transmit modules are push/pull, thus requiring extra circuitry to convert them to open drain behavior before connecting the Tx and Rx lines to enable single-wire functionality, as discussed further here.

Creative switching strategies and nonstandard ring connections are occasionally proposed to interconnect multiple UART devices. This gets complicated and inefficient (requiring substantial intervention by the CPU) in a hurry; if multiple devices need to be interconnected, the authors thus recommend instead using standard I2C (§4.2.2) or SPI (§4.2.3) for short-range connections, or CAN or RS485 (§4.4.3) for longer-range connections.

4.2.5 Reasoning for the Recon pin order

The logic for the pin order adopted across the entire Recon standard is as follows:

- Vcc and GND, which by definition are required on all powered connectors, come first. Following the uniquely extensible Recon standard, JST-ZH wire housings with M pins will often be fit into JST-ZH shrouded headers with N pins, where M<N. It is thus important that Vcc be located on the very first pin, as this prevents Vcc from accidentally being sent directly to any other pin on the client if the connector is inserted incorrectly (not engaging the first pin), thus minimizing the possibility of damaging the client device.

- Data lines come next, with priority given to output from the master in cases that data is carried over 2 wires,

- Clock comes next, followed by slave select(s).

- Optional coordinating signals come last, after all of the required signals, and ordered by frequency of use; these optional signals notably include INT/RES, CTS/RTS, extra SS lines, and extra GPIOs.

It is hoped that, following the logic presented here, new host and client devices following the space-efficient (1.5 mm pitch), secure, durable, extensible, and inexpensive JST-ZH based standard outlined in (4.1) will be developed by various manufacturers. Several conversations advocating for this new standard are already underway; if interested, please contact the author.

4.2.6 Recon compatibility, and incompatibility, with pin muxing on current devices

The Recon pin order is compatible with some essential pin muxing design decisions already made for a number of currently-available market-leading host and client devices, including the following:

1. Pin multiplexing {SDA, SCL} on I2C lines with, respectively, {Tx, Rx} on UART lines, as suggested by the UART-T standard, is consistent with the approach taken on several host MCUs, including the Broadcomm BCM2711 in the RPi4 (Table 5.9), and the TI C2000 and MSP432. Recon I2C connectors on such hosts can be converted directly into Recon UART-T connectors (at least, on these primary 2 signals) via a switch in software.

2. Alternate pin functions of STM32 USART modules between UART and SPI modes are consistent with the Recon standard; that is, USART modules on STM32-based hosts can be converted from Recon UART-T connectors (including {SCK, CTS, RTS}) to Recon SPI connectors (including {SCK, SSa, SSb}) via a switch in software. Further, on certain (host) STM32 ICs, {MOSI,MISO} of at least some dedicated SPI modules align with {Tx,Rx} of other dedicated UART modules, consistent with the Recon UART-T standard (at least, on these 2 signals).

3. DSM2/DSMX receivers, which happen to be available already with JST-ZH connectors, are compatible with the Recon UART-R standard. Further, STM32 USART, UART, and LPUART modules, when operating in half-duplex mode and not transmitting, can perform Rx functions on the Tx pin. Thus, the DSM receiver pinout is also compatible with the Recon UART-T standard when using an STM32 host operating in half-duplex mode.

Unfortunately, many pin multiplexing decisions made for currently-available host and client devices do *not* allow for simple software conversion between different comm protocols with consistent pin orders on a given connector (especially on the optional additional pins); indeed, some of these pin multiplexing decisions seem to have been made almost at random. For example:

A. On the STM32, the pin multiplexing between {SDA, SCL} and {Tx, Rx} matches the Recon UART-T order on some channels, but the Recon UART-R order on other channels.

B. On the RPi4, the pin multiplexing between SPI and UART with (optional) HFC, as shown in Table 5.9, does not follow any easily discernible reasoning [cf. §4.2.5].

It is suggested that broadly adopting a logical pin muxing standard, consistent with (4.1), might help both IC and PCB manufacturers, of both host and client devices, to market more capable and interoperable products with fewer pins. This may be made possible by deploying *reconfigurable* comm ports that may easily be switched between different comm protocols (e.g., via solder jumpers on clients, or via software on hosts), without having to change the wiring between the host and client devices, thus reducing both IC package size and board and wiring complexity, ultimately reducing manufacturing costs. If the idea of standardizing [to (4.1)] both pin multiplexing and (on connectors) pin order becomes well adopted, such ports could be named as, e.g., a **Recon I2C/UART-R** port (on a client), or a **Recon I2C/SPI/UART-T** port (on a host), thereby indicating both the various comm protocols available on those ports as well as the standardized Recon pin order used.

4.2.7 Extended Recon

It is at times useful to provide multiple regulated voltages over a connector. Perhaps the most common need for this is to provide a (low-current) standby voltage to GPS modules to facilitate warm starts; this need is addressed with the optional Vbat pin function in the Recon UART standard, as discussed in §4.2.4. However, other situations are anticipated which might also call for multiple regulated voltages to be provided over a single connector. To facilitate this, if appropriately documented (and, if possible, clearly called out on the silkscreen on the PCB itself), **Extended Recon** connector standards may be proposed, implementing one or more extra (optional) regulated voltages provided on pins placed *before* (to the left of) the Vcc = 3.3V pin appearing in the Recon standard given in (4.1). **Warning**: accidentally plugging into these extra voltage pins with a standard Recon connector will likely damage or destroy the host and/or client device; to reduce the likelihood of such a consequential mistake, small dummy plugs should be used to block these pins, thus safely reducing an Extended Recon connector to a standard Recon connector as defined in (4.1).

4.2.8 Stackable Recon

The Recon standard is designed to compactly, securely, and extensibly connect PCB hosts to nearby client devices elsewhere on the same mobile robot or electromechanical machine. As motivated in the first two paragraphs of §4.2, the Recon standard calls for JST-ZH connectors to be used.

At times when building a mobile robot or electromechanical machine, however, the SBC controlling the machine (and/or its COTS motor control board, such as those described in §5) does not quite have all of the necessary control or filter electronics implemented, and some addition custom circuits are required. In such situations, it is necessary for the user to design and use a custom daughterboard, to connect this custom daughterboard to one or more of the analog or digital comm (I2C, SPI, UART) channels on the SBC or the COTS motor control board, and to securely mount this custom electronics somewhere nearby.

For this task, the use of multiple single-row 0.1” pitch female headers laid out on a 0.1” grid, as popularized by Arduino, is quite convenient. Such an arrangement provides both electrical connectivity to the necessary channels as well as secure physical mounting of the custom daughterboard itself. Also, with such an arrangement and the use of stackable headers, two or more custom daughterboards may be stacked.

Stackable Recon and **Stackable Extended Recon** standards are thus defined that follow the same pin order as the Recon and Extended Recon standards defined above, but using single-row 0.1” pitch female headers (with 0.025” square pins) instead of JST-ZH connectors. The SPI and I2C Headers defined in Table 5.2 are examples of Stackable Extended Recon SPI and Stackable Extended Recon I2C connectors.

Note that small **servos** and **ESCs** ubiquitously come with 1x3 female jacks which mate with 0.1” pitch male header pins on the host. The (non-Recon) order of pins in modern servo connectors is {PWM signal, Vcc, GND}, respectively, with Vcc in the range of +4.8V to +12V. The reasoning for this order for servo connectors is that the 1x3 female jack may easily be plugged into the male header pins backwards; with this ordering (only), this is safe: it will result in the corresponding servo not functioning correctly until the plug is reversed, but it will not damage either the host or the servo. This ordering is well motivated and should not be changed.

4.3 Yukon: unpowered connectors

By removing the shared Vcc connection, to interconnect devices that are otherwise already powered, Recon connectors of the five types defined in (4.1) reduce to what we dub Yukon[21] connectors (again, leveraging the fact that durable [PTH] JST-ZH shrouded headers with N pins can accept JST-ZH wire housings with M pins when $M \le N$, thus creating a signal-extensible connector standard) as follows:

Yukon Basic	[GND,	S0,	S1,	S2,	S3,	S4,	S5,	S6,	...]	
Yukon I2C	[GND,	SDA,	SCL,	INT/G2,	RES/G3,	G4,	G5,	G6,	...]	
Yukon SPI	[GND,	MOSI,	MISO,	SCK,	SSa,	INT/SSb,	RES/SSc,	SSd,	...]	(4.3)
Yukon UART-T	[GND,	Tx,	Rx/G1,	SCK/Vbat/G2,	CTS/G3,	RTS/G4,	G5,	G6,	...]	
Yukon UART-R	[GND,	Rx,	Tx/G1,	SCK/Vbat/G2,	RTS/G3,	CTS/G4,	G5,	G6,	...]	

As with the Recon Basic standard discussed in §4.2.1, the Yukon Basic standard may be implemented in **Yukon GPIO**, **Yukon PWM**, **Yukon Enccoder**, **Yukon Clock**, **Yukon Analog**, and **Yukon Digital** variants.

Provided that significant care is exercised when plugging in the connector (in this case, NOT engaging the first pin), Recon connectors can actually be used as Yukon connectors. **Warning**: if this is done incorrectly, power on one side will be connected directly to GND on the other, likely damaging or destroying one or both devices; to reduce the likelihood of such a consequential mistake, while also making the connection of the wire housing even a bit more secure, a small dummy plug should be used to block the first pin of any Recon connector, thus safely reducing it into a corresponding Yukon connector.

As in §4.2.8, **Stackable Yukon** connectors are also defined, which follow exactly the same pin order as the Yukon standards defined above, but use single-row 0.1" pitch female headers instead of JST-ZH connectors. The Analog Header defined in Table 5.2 is an example of a Stackable Yukon Analog connector.

4.4 CAN and RS485 differential interfaces for remote connections

Though a variety of different effective distances are reported around the web under various conditions (spacing and characteristic impedance of the traces and wires used, electromagnetic interference, possible impedance mismatches at IC/trace and trace/wire junctions, etc), without a repeater and when operating at low comm speeds, I2C links are practically limited to somewhere around 5 m, SPI links are limited to around 10 m, and UART links are limited to around 15 m; at higher comm speeds these effective distance limits are all substantially reduced. To connect over longer distances, differential interfaces communicating over one or more twisted pairs of wires are needed. The two dominant standards today for such differential interfaces are[22] CAN and RS485[23]. A few useful comparisons of these two standards are available here, here, and here. There are a number of subtle issues, including interconnect topology, termination, biasing, grounding, etc., involved in making such systems work well; a succinct review is available here. Industrial RS485 data cables are typically 24 AWG with a characteristic impedence of 100 Ω to 120 Ω; automotive CAN data cables are typically 18 to 20 AWG with characteristic impedence of 110 Ω to 130 Ω. CAT5e or CAT6 cables are often-used inexpensive COTS substitutes for low-cost RS485 networks. Shielding is helpful for maintaining signal integrity, if it is available.

[21] In contrast to the Recon ("Renaissance Connector") standard for short-distance powered connections to sensors, the name of the Yukon standard, which itself evokes extreme physical distancing, is derived as a homophone of Ucon ("Unpowered Connector").

[22] That is, other than Ethernet, which itself may be a good choice for many long-distance local networks.

[23] RS485 is related to the older but still commonly used RS422 standard. Through a driver enable (DE) feature, RS485 systems can operate with multiple drivers (transmitters) on a single pair of wires, thus facilitating half-duplex (two-way communication) over a single twisted pair (RS422 is simplex only over each pair of wires). RS485 transmitters can handle the load of 32 to 256 receivers on a single twisted pair; some RS422 transmitters can only handle the load of 10 receivers. RS485 can also handle much larger ground potential differences between devices. For these reasons, RS485 is generally preferred over RS422 for new designs.

4.4.1 To ground, or not to ground?

The question of whether or not a GND connection should be shared between different devices when using a differential interface is particularly delicate. Notwithstanding advice to the contrary in the RS485 (aka TIA485-A) standard itself, which recommends simply using resistors between the ground wire on the interconnecting RS485 cable and the local GND on individual devices, as well as a lot of other misleading advice elsewhere on the web, careful modern guidance is somewhat more nuanced. In short, non-isolated GND should not be shared in situations for which the ground potential difference (GPD) of all devices to be connected will remain well within ± 7 V, and thus non-isolated CAN transceivers and RS485 transceivers may be used, whereas isolated GND should be shared in situations for which the GPD might exceed this range, and thus slightly more expensive/complex isolated CAN transceivers and RS485 transceivers should be used instead.

4.4.2 Field-serviceable, secure, and durable wiring solutions

As discussed in the definition of the Recon standard §4.2, for the wiring of single-board computers to nearby sensors and other devices in mobile robots and within individual space-constrained electromechanical machines, standardized connectors are called for that are:

(a) small (1.5mm pitch appears to be the sweet spot),
(b) secure (not disconnecting due to system vibrations),
(c) durable (able to withstand hundreds of connector insertion/removal cycles – generally this means PTH),
(d) extensible (able to incorporate additional signals if available/necessary), and
(e) inexpensive (leveraging COTS connectors wherever possible, especially for mass-market products).

In contrast, for long-distance twisted-pair wiring solutions (e.g., for automotive, industrial, and outdoor applications), wires and connectors are needed that are:

(A) field serviceable (allowing wires to be replaced with only simple tools, or no tools whatsoever),
(B) even more secure (not disconnecting due to accidental direct tugs on the wire), and
(C) even more durable (surviving heat, direct sunlight, vibration, water, dust, grease, cleaning solvents, etc).

For long-distance connections, the wires themselves are often the weakest links, and must often be replaced when damaged, or cut to a new lengths when the system is reconfigured. In such a setting, connector size and cost are often only minor secondary issues, and field serviceability is paramount. A variety of standardized connectors and T-junctions are available, with 2 to 9 poles, that are well suited in such settings, notably including:

- Pico (M8), Micro (M12), and Mini (7/8 in) connectors, many of which are IP67[24] rated and IDC[25] type,
- RJ45 connectors (8-pin, straight-through T568B, also available as IDC), as used widely for wired ethernet,
- D-sub 9-pin (DE9, aka DB9) connectors, a 0.108" pitch D-shaped shrouded standard that is broadly adopted,
- Micro-D 9-pin connectors, a 0.05" pitch miniaturized version of the DE9 (also available in powered variants),
- simple terminal blocks, which are easy to service by hand but not environmentally hardened, etc.

In each setting, it is essential to follow the corresponding industry or manufacturer's spec as much as possible for where to attach the primary (and if present, secondary) pair of signal wires. For example, if using RJ45 connectors [typically, with inexpensive commercial off-the-shelf (COTS) CAT5e or CAT6 cables], it is recommended that the standard Power over Ethernet pin order be followed:

- the primary twisted pair should be attached to pins 3 and 6,
- the secondary twisted pair (if any; e.g., in full-duplex RS485 mode) should be attached to pins 1 and 2,

[24] IP67 means that the component is dust-proof and capable of withstanding temporary immersion up to 1 m depth.

[25] Field-servicable Insulation Displacement Connection (IDC) type connectors may be installed with only simple tools, or in certain cases with no tools whatsoever.

- GND (if connected) should be carried on pins 7 and 8, and
- DC power (if connected, which is sometimes convenient for small remote sensors) should be on pins 4 and 5.

4.4.3 Recon/Yukon Differential Pairs

The various types of rugged (field serviceable, extra secure, extra durable) connectors discussed above are often mounted on a bulkhead (that is, on the boundary of an environmentally-hardened shell protecting the electronics), and are often too big to mount directly on the PCB itself. In this common setting, the rugged connector on the bulkhead needs to connect to a (small, extensible) connector on the PCB via a short jumper wire (twisted pair ribbon cables are often a good choice). For the connector on the PCB in this setting, a simple extension of the Recon/Yukon standard is recommended:

Recon/Yukon Differential Pairs: [Vcc, GND, A+, A-, B+, B-, C+, C-, D+, D-, ...] (4.4)

Only the first differential pair, denoted here {A+, A-}, are required by this spec; additional differential pairs may be added if available. If both Vcc and GND are included, it is referred to as a Recon Differential Pair connector, otherwise it is referred to as a Yukon Differential Pair connector. Further, on Yukon Differential Pair connectors, GND is also optional; recalling the discussion in §4.4.1, a GND connection should generally not be made between two or more connected devices unless isolated CAN or RS485 transceivers are used[26]. Useful definitions of the CAN and RS485 variants of the Recon/Yukon Differential Pairs spec follow[27]:

Recon/Yukon CAN:	[Vcc, GND, CANH, CANL]	
Recon/Yukon RS485-H:	[Vcc, GND, A, B]	for half duplex connections over the A/B channel,
Recon/Yukon RS485-Y:	[Vcc, GND, Y, Z, A, B]	full duplex, with the Y/Z (transmit) channel first,
Recon/Yukon RS485-A:	[Vcc, GND, A, B, Y, Z]	full duplex, with the A/B (receive) channel first.

(4.5)

In the full duplex case, akin to the Recon/Yukon UART-T and UART-R standards, note that:

- master devices transmit on the first pair and receive on the second using Recon/Yukon RS485-Y connectors,
- slave devices receive on the first pair and transmit on the second using Recon/Yukon RS485-A connectors,

thus facilitating full duplex communication between any master and any slave on the network. Warning: in either the full-duplex or half duplex case, care must be taken in software such that, on any given twisted pair, only one RS485 node has an active driver (transmitting) at an given time.

By electrically connecting the Y and A pins and the Z and B pins of a full-duplex RS485 transceiver at the JST-ZH connector [e.g., with (initially-open) solder jumpers on the PCB nearby], and attaching a data cable to the A and B (or, to the Y and Z) pins only, a (full-duplex) Yukon RS485-A or RS485-Y connector is reduced to (half-duplex) RS485-H functionality. This is a useful way to configure full-duplex RS485 transceivers+connectors for general use, if you don't know whether a full-duplex or half-duplex RS485 network will ultimately be deployed.

In the full duplex case, the receive and transmit functions are decoupled from each other at any given node. In the half-duplex case, it is common to turn the receiver off whenever transmitting, and vice-versa, simply by tying the (active high) DE (driver enable) pin to the (active low) $\overline{\text{RE}}$ (receiver enable) pin. This is not the only valid approach, however:

- by receiving all the time (tying $\overline{\text{RE}}$ low), a transmitting node also receives its own data as it is being sent (this is called a "loopback" or "echo" function, and can be used to verify the quality of each transmission), or
- by shutting off both the transmitter and the receiver (setting DE low and $\overline{\text{RE}}$ high), a node can save energy.

[26] To prevent making a mistake in this regard, Yukon connectors without ground pins should be used on PCBs with non-isolated transceivers, and Recon or Yukon connectors with ground pins may be used on PCBs with isolated transceivers; the latter should always be selected, and the corresponding isolated GND pins connected, if significant GPDs are expected.

[27] If using simple terminal blocks on the bulkhead, we recommend following the same order.