P A R K   R O A D   D E S I G N S

PRD-GP-STM

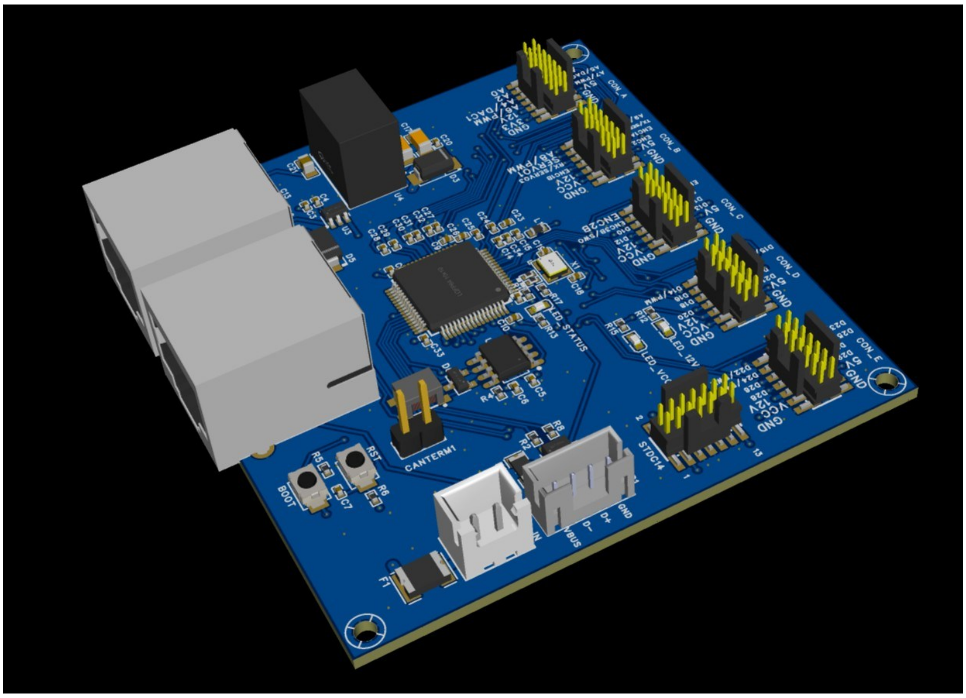
General Purpose Processing Board

User Guide

Version 1.0

STM32F072RBT6 · CAN Bus · USB HID · CAN-USB Bridge

www.parkroaddesigns.com



PRD-GP-STM General Purpose Processing Board — 3D render

1. Safety Notices

Electrical Hazard

This board operates with voltages up to 12V DC. While not dangerous to human health under normal circumstances, incorrect wiring can permanently damage the board, connected peripherals, or the host computer. Always disconnect all power before making wiring changes.

Not for Safety-Critical Applications

The PRD-GP-STM is designed for use in flight simulation and hobbyist applications only. It must not be used in any safety-critical, life-critical, or mission-critical application. Park Road Designs accepts no liability for damage, injury, or loss arising from use of this product.

RJ45 Connectors are NOT Ethernet

The RJ45 connectors on this board carry CAN bus signals and 12V DC power. They are NOT compatible with Ethernet networks or equipment. Connecting this board to an Ethernet switch, router, or network port may damage the board, the network equipment, or both. Always label CAN bus cables clearly.

ESD Handling

This board contains ESD-sensitive components. Handle using standard ESD precautions -- use an anti-static wrist strap when handling the bare board and store in anti-static packaging when not in use.

CN1 Power Connector and Polyfuse F1

The CN1 external 12V power connector and F1 polyfuse (1812L200/30GR, 2A hold / 4A trip) are fitted on all boards. CN1 allows any board to act as the 12V power injector for the CAN bus. See Section 7 for details.

2. Introduction

The PRD-GP-STM General Purpose Processing Board is a compact, versatile microcontroller board designed primarily for use in flight simulator cockpit peripherals. It provides a rich set of analog inputs, digital GPIO, hardware quadrature encoders, PWM servo outputs, CAN bus connectivity, and full-speed USB -- all in a single universal PCB that can be programmed for any peripheral role.

The board is built around the STM32F072RBT6 microcontroller from ST Microelectronics, a 48MHz ARM Cortex-M0 processor with native USB support, CAN bus controller, 12-bit ADC, and extensive timer peripherals. Every board in a rig runs identical firmware -- the board determines its own role automatically based on what is physically connected to it at startup.

A key feature of the PRD-GP-STM architecture is the CAN-USB bridge capability. Whichever board in the rig has a USB cable connected to a PC automatically becomes the CAN-USB bridge for the entire rig, forwarding CAN frames from all other boards to the PC as a USB HID joystick device and SLCAN virtual COM port simultaneously. No dedicated bridge hardware is required.

2.1 Key Features

- Universal design -- one board serves any flight sim peripheral role
- STM32F072RBT6 ARM Cortex-M0 at 48MHz, 128KB flash, 16KB RAM
- 10 x 12-bit ADC inputs for potentiometers and hall effect sensors
- 2 x 12-bit DAC outputs for analog meter drives or position indicators
- 3 x hardware quadrature encoder inputs (32-bit and 16-bit counters)
- Up to 7 simultaneous PWM servo outputs
- 30 digital GPIO pins across 5 signal connectors
- CAN 2.0B at 500kbps via TJA1051T/3 transceiver
- USB Full Speed 2.0 -- HID joystick + SLCAN CDC bridge
- Automatic CAN-USB bridge role via leader election protocol
- 12V CAN bus power distribution via RJ45 daisy-chain
- STDC14 debug connector (SWD + VCP + SWO trace)
- 4-layer PCB with solid GND plane for clean analog performance
- No dedicated bridge hardware needed -- any board can be the bridge

2.2 What's in the Box

The PRD-GP-STM is supplied as a fully assembled and tested PCB. The following items are included:

- 1 x PRD-GP-STM assembled PCB
- This user guide

The following items are required but not included and must be sourced separately:

- 12V DC power supply (see Section 7)
- Standard Ethernet patch cables for CAN bus connections (see Section 8)
- IDC ribbon cable assemblies for signal connectors CON-A through CON-E
- Peripheral hardware (potentiometers, encoders, switches) appropriate to your build
- ST-LINK V3SET or compatible SWD programmer for firmware development (USB DFU programming requires no additional hardware)

Firmware and Resources
Firmware examples, documentation updates, and application notes are available at www.parkroaddesigns.com

3. Key Specifications

Parameter	Specification
Microcontroller	STM32F072RBT6 ARM Cortex-M0 48MHz
Flash memory	128KB
RAM	16KB SRAM
ADC	12-bit, 10 channels (A0-A9)
DAC	12-bit, 2 channels (A4, A5)
Hardware encoders	3 x quadrature (TIM2 32-bit, TIM3 16-bit, TIM4 16-bit)
PWM servo outputs	Up to 7 simultaneous (TIM1, TIM15, TIM3)
Digital GPIO	30 pins across CON-A to CON-E
USB	Full Speed 2.0, composite HID + CDC
CAN bus	CAN 2.0B, 500kbps, via TJA1051T/3
Power input -- CAN bus	12V nominal via RJ45 (SMBJ15A TVS + SS34 reverse polarity protection)
Power input -- USB	5V via USB connector (500mA polyfuse)
Power input -- EXT (CN1)	12V via CN1 (fitted on all boards, 2A/4A polyfuse F1)
Operating voltage	3.3V core (regulated by TPRT9013-33GB LDO from 5V rail)
5V rail voltage	~4.7V (post Schottky diode, within spec for all components)
Signal connectors	5 x 14-pin 1.27mm SMD IDC (CON-A to CON-E)
Debug connector	STDC14 -- SWD + VCP + SWO
CAN connectors	2 x RJ45 (daisy-chain in/out)
Crystal	16MHz X322516MLB4SI with 12pF load caps
Operating temperature	-40 to +85 degrees C
PCB	4-layer FR4

4. Board Overview

The PRD-GP-STM is a rectangular 4-layer PCB. The following sections describe the key areas and components visible on the board.

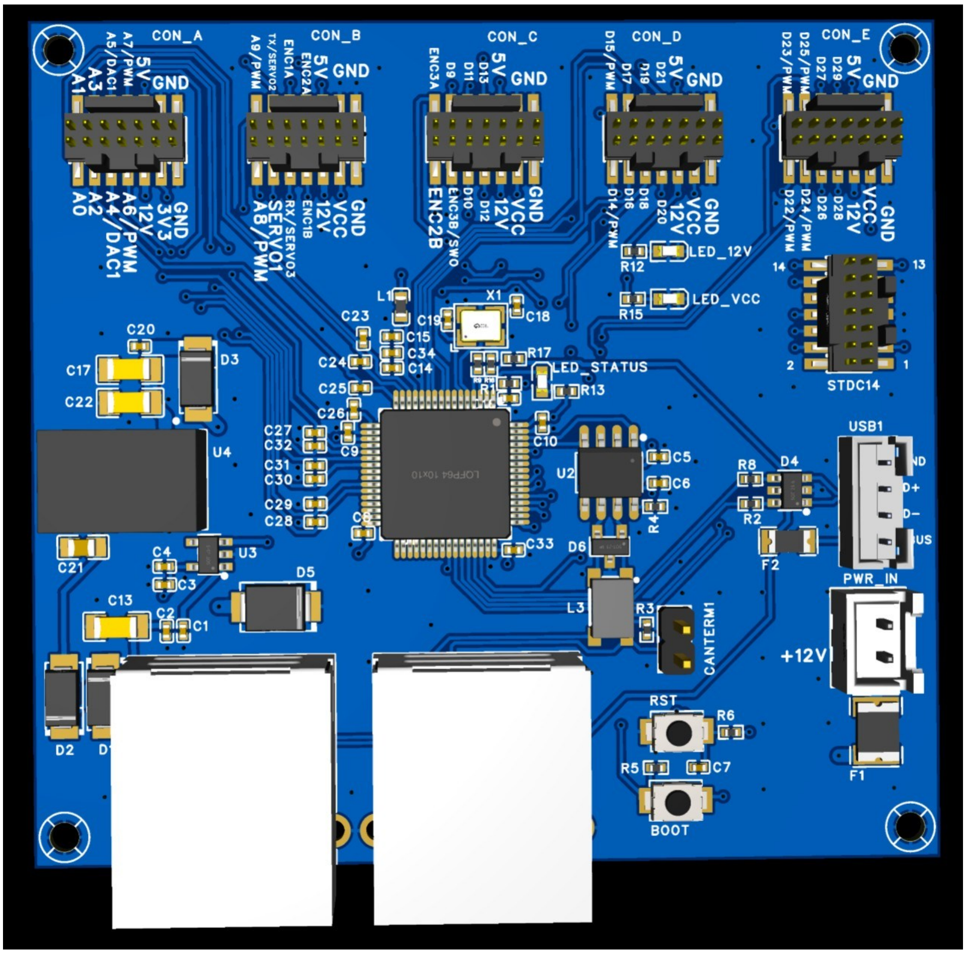


Figure 1 -- PRD-GP-STM top view showing connectors CON-A through CON-E (top), RJ45 CAN bus connectors (bottom), and key components

4.1 Top Edge -- Signal Connectors

Five 14-pin 1.27mm SMD IDC connectors (CON-A through CON-E) run across the top edge of the board. These provide all signal connectivity to peripheral devices. Pin 1 of each connector is marked with a dot on the silkscreen. Each connector carries 8 signal pins (pins 1-8) and 6 power pins (pins 9-14).

> **WARNING: Pin 1 orientation is critical. A reversed IDC cable will apply 12V to signal pins and will permanently damage the MCU. Always verify pin 1 alignment before applying power.**

4.2 Bottom Edge -- CAN Bus Connectors

Two RJ45 connectors (RJ1 and RJ2) on the bottom edge provide CAN bus connectivity. They are wired identically and in parallel -- the bus daisy-chains through both connectors, allowing boards to be connected in a chain with standard Ethernet patch cables.

4.3 Right Edge -- Debug, USB, and Power

Three connectors are grouped on the right edge:

- STDC14 -- 14-pin 1.27mm SMD debug connector for SWD programming, virtual COM port, and SWO trace output. Connect to an ST-LINK V3SET for firmware development.
- USB1 -- PH2.0-4P USB connector for USB HID and CAN-USB bridge operation. This is also used for DFU firmware upload on new boards.
- CN1 -- 2-pin 1.27mm SMD external 12V power input. Fitted on all boards. Allows any board to act as the 12V power injector for the CAN bus. Protected by F1 polyfuse (2A hold, 4A trip).

4.4 Bottom-Right -- Buttons

Two tactile push buttons:

- SW1 (RST) -- Reset button. Press to restart the MCU.
- SW2 (BOOT) -- Bootloader entry button. Hold SW2 while pressing and releasing SW1 to enter USB DFU bootloader mode for firmware upload without an ST-LINK.

4.5 LED Indicators

LED	Label	Colour	Meaning
LED_STATUS	STATUS	Red	Firmware-controlled. Lit when firmware is running and operational.
LED_VCC	VCC	Red	3.3V power rail present. Lit whenever the board is powered from any source.
LED_12V	12V	Red	12V supply present. Lit when 12V is available from the CAN bus RJ45 or CN1.

4.6 Power Section (Bottom-Left)

The power regulation components occupy the bottom-left area of the board, physically separated from the analog signal section. Key components visible here include U4 (K7805 DC-DC converter, 12V to 5V), U3 (TPRT9013 LDO, 5V to 3.3V), D1/D2/D3 (SS34 Schottky diodes for supply OR-ing), and D5 (SMBJ15A TVS transient suppressor).

5. Board Roles and Automatic Role Detection

Every PRD-GP-STM board runs identical firmware. The board determines its own operating role at startup by checking the state of PC14, which senses whether a USB cable is connected (VBUS present). No configuration switches, jumpers, or separate firmware images are needed.

Role	Condition	Behaviour
Peripheral only	No USB connected, 12V from CAN bus	Reports sensor data (axes, buttons) over CAN bus to whichever board is acting as bridge.
Peripheral + Bridge	USB connected, 12V from CAN bus or CN1	Reports its own sensor data over USB as an HID joystick AND bridges all CAN traffic to/from the PC via SLCAN CDC.
Bridge only	USB connected, CN1 12V supply, no other nodes	Pure CAN-USB bridge with 12V power injection to the bus.
USB peripheral only	USB connected, no 12V available	USB HID device only. CAN bus inactive (TJA1051 requires 5V from 12V supply chain).

5.1 Bridge Leader Election

If multiple boards in the rig have USB connected simultaneously, a CAN-based leader election protocol prevents multiple simultaneous bridges. When a board detects USB connection, it does not immediately become the bridge -- instead it broadcasts an election message on the CAN bus containing its unique 96-bit device ID. The board with the lowest device ID wins the election and becomes the bridge. All other USB-connected boards suppress their CDC interface and operate as HID-only USB devices.

Reserved CAN IDs for bridge management (do not use for application data):

CAN ID	Message	Direction
0x7FF	BRIDGE_ELECTION	Broadcast by any board wanting to become bridge
0x7FE	BRIDGE_ACTIVE / BRIDGE_HEARTBEAT	Broadcast by current bridge every 1000ms
0x7FD	BRIDGE_YIELD	Broadcast by election loser
0x7FC	BRIDGE_RESIGN	Broadcast by bridge when USB disconnected

Hot-swap USB
The bridge role transfers cleanly when the USB cable is moved to a different board. The original bridge broadcasts BRIDGE_RESIGN and de-initialises USB. The new board initiates election and takes over after 50ms. All other peripheral boards continue operating without interruption throughout this process.

6. First Use -- Getting Started

6.1 Before You Power On

1. Inspect the board for any visible assembly defects -- check for solder bridges on the MCU (LQFP-64, 0.5mm pin pitch).
2. Confirm SW1 and SW2 buttons are fitted and not stuck.
3. Confirm the CANTERM1 header jumper is in the correct position for this board's position in the CAN bus chain -- jumper fitted for end nodes, jumper removed for middle nodes.
4. Confirm F1 polyfuse is correctly oriented. Confirm CN1 connector is correctly oriented (pin 1 = positive 12V).

6.2 Installing Firmware for the First Time

New boards ship without user firmware. The STM32F072 contains a factory ROM bootloader that allows firmware to be installed via USB without any additional programmer.

Step 1 -- Install STM32CubeProgrammer on your PC. Download from st.com/stm32cubeprog. Available for Windows, Mac, and Linux.

Step 2 -- Enter DFU bootloader mode:

- Hold SW2 (BOOT button)
- While holding SW2, press and release SW1 (RST button)
- Release SW2
- Connect USB cable to the board and to your PC

Step 3 -- Upload firmware:

- Open STM32CubeProgrammer
- Select USB as the connection type
- Click Connect -- the board should be detected as STM32 BOOTLOADER
- Click Open File and select your firmware .hex or .bin file
- Click Download to flash the firmware
- Press SW1 (RST) to start the firmware

Firmware Examples
Reference firmware and examples for common peripheral configurations are available at www.parkroaddesigns.com

6.3 Verifying Basic Operation

After firmware upload, press SW1 (RST). LED_VCC should illuminate immediately confirming 3.3V power. LED_STATUS should illuminate or flash once firmware starts. If USB is connected, the PC should enumerate the board as a new HID joystick device and a new COM port within a few seconds.

7. Power Supply

7.1 Power Sources

The PRD-GP-STM accepts power from three sources, combined via a Schottky diode OR circuit. Only one source is needed for the board to operate.

Source	Connector	Voltage	When to use
CAN bus 12V	RJ45 (RJ1 or RJ2)	12V	Normal operation with multiple boards. 12V distributed across all boards in the rig via RJ45 cables.
USB 5V	USB1 (PH2.0-4P)	5V VBUS	Single board use, or firmware development. CAN bus is inactive without 12V.
External 12V (CN1)	CN1 (2-pin 1.27mm SMD)	12V	On the designated power injector board. Feeds 12V onto the CAN bus cable for all other boards.

7.2 12V Power Supply Requirements

For a complete rig with multiple boards, a 12V DC regulated power supply connected to CN1 on the bridge board provides power to all boards via the CAN bus cables.

Power budget per board at typical load (potentiometers, switches, LEDs):

- MCU and logic: ~50mA at 3.3V from 12V via K7805 and LDO
- Typical per-board 12V draw: ~80-120mA at moderate load
- Maximum per-board 12V draw: ~230mA at full load

Number of boards	Typical 12V current	Recommended supply
1 board	~100mA	USB power alone is sufficient -- no 12V supply needed
2-4 boards	200-480mA	12V 1A regulated supply minimum
5-8 boards	500mA - 1A	12V 2A regulated supply recommended
9-12 boards	720mA - 1.4A	12V 3A regulated supply recommended

> **WARNING: Only ONE 12V supply should be connected to the rig at any time. Multiple 12V supplies connected simultaneously via CN1 on different boards will fight each other on the shared 12V bus rail. The supply with higher output voltage will back-feed into the lower, potentially tripping polyfuses or causing instability.**

7.3 Fitting CN1 and F1

CN1 is a 2-pin 1.27mm SMD header connector fitted on all boards. F1 is a 1812L200/30GR resettable polyfuse (2A hold, 4A trip, 30V rated) also fitted on all boards. CN1 allows any board to inject 12V power onto the CAN bus cable -- typically only one board in the rig will have a 12V supply connected to CN1 at any time, but the connector and protection components are present on every board for flexibility.

CN1 Pin	Connection
Pin 1	Positive 12V input (via F1 polyfuse to 12V_IN net)

CN1 Pin	Connection
Pin 2	GND

The polyfuse F1 (1812L200/30GR) protects the supply against overload. It will trip if the total rig current exceeds approximately 4A and reset automatically once the overload is removed and the board cools.

7.4 Power Rail Voltages on Signal Connectors

Power rails are available on pins 9-14 of every signal connector (CON-A through CON-E):

Pin	Net	Voltage	Max current per connector	Notes
9	+12V	~12V	Limited by F1 polyfuse (2A total rig)	Only present when 12V supply connected
10	+5V	~4.7V	500mA total across all connectors	Available from 12V or USB supply
11	3V3 (VCC)	3.3V	200mA total across all connectors	Always regulated to 3.3V
12	GND	0V	--	Ground
13	GND	0V	--	Ground
14	GND	0V	--	Ground

> *NOTE: The +5V rail is approximately 4.7V rather than 5.0V due to the Schottky diode forward voltage drop in the OR circuit. This is within specification for all components on the board and for standard peripheral devices. External devices powered from the +5V pin should tolerate 4.5-5.0V.*

8. CAN Bus Network

8.1 Overview

The CAN bus connects all PRD-GP-STM boards in the rig into a single network, carrying sensor data from peripheral boards to the bridge board and distributing 12V power to all boards simultaneously. The CAN bus operates at 500kbps using the CAN 2.0B standard.

8.2 Cabling

Standard Ethernet patch cables (straight-through, NOT crossover) are used for CAN bus connections. Any Category 5 or better patch cable will work. The maximum recommended total bus length is 100 metres at 500kbps, though in practice sim rig cable runs will be well under 5 metres.

> **WARNING: Do NOT use crossover Ethernet cables. Do NOT connect these cables to Ethernet switches, routers, or any network equipment. The RJ45 connectors carry CAN signals and 12V DC, not Ethernet signals. Label all CAN bus cables to prevent accidental connection to network infrastructure.**

8.3 RJ45 Connector Pinout

RJ45 Pin	Signal	Notes
1	CAN_H	CAN bus high
2	CAN_L	CAN bus low
3	GND	Ground
4	12V	12V power distribution
5	12V	12V power (duplicated for current capacity)
6	GND	Ground
7	GND	Ground
8	GND	Ground

8.4 Bus Topology

The CAN bus uses a linear daisy-chain topology. Each board has two RJ45 connectors (RJ1 and RJ2) wired in parallel, allowing cables to pass through each board in the chain.

Example 4-board rig:
[Bridge+Power] --cable-- [Throttle] --cable-- [Yoke] --cable-- [Rudder]

8.5 Bus Termination

The CAN bus requires 120 ohm termination resistors at both physical ends of the bus. Each PRD-GP-STM board has a CANTERM1 2-pin 2.54mm header. When a jumper cap is fitted on this header, it connects a 120 ohm resistor across CANH and CANL, terminating the bus at that point.

> **WARNING: Only the two boards at the physical ends of the daisy chain should have a jumper cap fitted on CANTERM1. All boards in the middle of the chain must have the CANTERM1 jumper cap removed. Incorrect termination causes CAN bus errors across all nodes.**

Board position	CANTERM1 state
First board in chain (one end)	JUMPER FITTED -- fit a 2.54mm jumper cap on CANTERM1 header
Middle boards	JUMPER REMOVED -- no jumper cap on CANTERM1 header
Last board in chain (other end)	JUMPER FITTED -- fit a 2.54mm jumper cap on CANTERM1 header

8.6 CAN Bus Protocol and Message IDs

Application firmware should use CAN IDs in the range 0x000-0x7FB for application data. IDs 0x7FC-0x7FF are reserved for bridge management messages and must not be used by application code.

Recommended ID allocation for a typical rig (adjust to suit your design):

CAN ID range	Suggested use
0x010-0x01F	Analog axis data (throttle, mixture, prop, rudder)
0x020-0x02F	Button/switch states
0x030-0x03F	Encoder values
0x040-0x04F	Servo commands (from PC to peripheral)
0x7FC-0x7FF	RESERVED -- bridge management (do not use)

9. USB Interface and CAN-USB Bridge

9.1 USB Connector

The USB connector (USB1) is a PH2.0-4P type on the right edge of the board. Connect to a PC using a USB cable with a PH2.0-4P plug on the board end and a standard USB-A or USB-C on the PC end.

9.2 USB Device Enumeration

When USB is connected and firmware is running, the board enumerates as a USB composite device presenting two interfaces to the PC:

Interface	USB Class	PC Device	Purpose
Interface 0	HID Joystick	Game controller	Reports this board's analog axes and button states for use by the simulator software
Interface 1	CDC Virtual COM Port	COM port (COMx on Windows, /dev/ttyACMx on Linux/Mac)	CAN bus frame forwarding using SLCAN protocol

9.3 SLCAN Protocol

The CDC virtual COM port uses SLCAN (Serial Line CAN), a widely supported ASCII protocol for CAN frame forwarding. The port can be opened at any baud rate -- the SLCAN protocol is independent of the COM port baud rate setting.

SLCAN Command	Format	Description
Open channel	O\r	Open CAN channel at configured rate (500kbps)
Close channel	C\r	Close CAN channel
Set baud rate	S5\r	Set 500kbps (S3=100k, S4=250k, S5=500k, S6=800k, S8=1M)
Transmit standard frame	tIIILDD...\r	t + 3-char hex ID + 1-char DLC + data bytes
Transmit extended frame	TIIIIIIIILDD...\r	T + 8-char hex ID + 1-char DLC + data bytes
Receive standard frame	tIIILDD...\r	Sent by board when standard frame received
Receive extended frame	TIIIIIIIILDD...\r	Sent by board when extended frame received

Example: transmit a standard CAN frame with ID 0x010, 2 data bytes 0xAB 0xCD:

```
t01020ABCD\r
```

SLCAN is supported natively by many tools including Wireshark (with plugin), python-can, CANalyzer, and many others. For python-can:

```
import can
bus = can.interface.Bus(interface='slcan', channel='COM5', bitrate=500000)
msg = can.Message(arbitration_id=0x010, data=[0xAB, 0xCD])
bus.send(msg)
```

9.4 First-Time Firmware Installation via USB

New boards have no user firmware installed but contain a factory ROM DFU bootloader. Firmware can be installed via USB without any additional programmer:

5. Hold SW2 (BOOT), press and release SW1 (RST), then release SW2
6. Connect USB cable -- PC detects STM32 BOOTLOADER device
7. Open STM32CubeProgrammer, select USB connection, connect
8. Load firmware .hex file and click Download
9. Press SW1 (RST) to start firmware

10. Signal Connector Reference

Five 14-pin 1.27mm SMD IDC headers (CON-A through CON-E) provide all signal connectivity. Pin 1 is marked with a dot on the PCB silkscreen. Pins 1-8 carry signals. Pins 9-14 carry power rails.

10.1 Power Pin Layout (identical on all connectors)

Pin	Signal	Voltage	Notes
9	+12V	~12V	CAN bus supply -- only present when 12V connected
10	+5V	~4.7V	Available from 12V or USB
11	3V3	3.3V	Regulated
12	GND	0V	Ground
13	GND	0V	Ground
14	GND	0V	Ground

10.2 CON-A -- Analog Inputs

All 8 signal pins are connected to the STM32 12-bit ADC. A 100nF ceramic filter capacitor to GND is fitted on-board at each ADC pin for noise suppression. Pins A4 and A5 additionally support 12-bit DAC output.

Pin	Label	GPIO	ADC	Alt function	Typical use
1	A0	PA0	CH0	TIM2_CH1	Throttle / rudder / pitch axis
2	A1	PA1	CH1	TIM2_CH2	Mixture / left brake / roll axis
3	A2	PA2	CH2	TIM2_CH3, TIM15_CH1	Prop pitch / right brake / POV horizontal
4	A3	PA3	CH3	TIM2_CH4, TIM15_CH2	POV vertical / spare analog
5	A4/DAC1	PA4	CH4	DAC1 output	Analog input or DAC output 1
6	A5/DAC2	PA5	CH5	DAC2 output	Analog input or DAC output 2
7	A6/PWM	PA6	CH6	TIM3_CH1, TIM16_CH1	Analog input or servo PWM output
8	A7/PWM	PA7	CH7	TIM3_CH2, TIM17_CH1	Analog input or servo PWM output

> *NOTE: When A4 or A5 are configured as DAC outputs, the corresponding ADC channel is unavailable. These pins are useful for driving analog panel meters or servo position displays.*

10.3 CON-B -- Analog (continued) + Servo + Encoders

Pin	Label	GPIO	Type	Alt function	Typical use
1	A8/PWM	PB0	ADC CH8	TIM3_CH3	Analog input or servo PWM output

Pin	Label	GPIO	Type	Alt function	Typical use
2	A9/PWM	PB1	ADC CH9	TIM3_CH4	Analog input or servo PWM output
3	SERVO1	PA8	PWM out	TIM1_CH1	Primary servo output (TIM1 advanced timer)
4	TX/SERVO2	PA9	UART TX	USART1_TX, TIM1_CH2	VCP TX to STDC14. Servo output when VCP not used.
5	RX/SERVO3	PA10	UART RX	USART1_RX, TIM1_CH3	VCP RX from STDC14. Servo output when VCP not used.
6	ENC1A	PB4	Encoder A	TIM3_CH1	Encoder 1 phase A -- hardware quadrature (TIM3)
7	ENC1B	PB5	Encoder B	TIM3_CH2	Encoder 1 phase B -- hardware quadrature (TIM3)
8	ENC2A	PB6	Encoder A	TIM4_CH1	Encoder 2 phase A -- hardware quadrature (TIM4)

10.4 CON-C -- Encoders (continued) + Digital GPIO Bank 1

Pin	Label	GPIO	Type	Alt function	Typical use
1	ENC2B	PB7	Encoder B	TIM4_CH2	Encoder 2 phase B -- hardware quadrature (TIM4)
2	ENC3A	PA15	Encoder A	TIM2_CH1	Encoder 3 phase A -- hardware quadrature (TIM2, 32-bit)
3	ENC3B/SWO	PB3	Encoder B / SWO	TIM2_CH2, TRACESWO	Encoder 3 phase B or SWO debug trace output
4	D9	PB2	GPIO	--	Digital input -- switches, buttons
5	D10	PB10	GPIO	TIM2_CH3, I2C2_SCL	Digital input
6	D11	PB11	GPIO	TIM2_CH4, I2C2_SDA	Digital input
7	D12	PB12	GPIO	TIM1_BKIN, SPI2_NSS	Digital input
8	D13	PB13	GPIO	TIM1_CH1N, SPI2_SCK	Digital input

10.5 CON-D -- Digital GPIO Bank 2

Pin	Label	GPIO	Type	Alt function	Typical use
1	D14/PWM	PB14	GPIO/PWM	TIM15_CH1, SPI2_MISO	Digital input or servo PWM output

Pin	Label	GPIO	Type	Alt function	Typical use
2	D15/PWM	PB15	GPIO/PWM	TIM15_CH2, SPI2_MOSI	Digital input or servo PWM output
3	D16	PC0	GPIO	--	Digital input
4	D17	PC1	GPIO	--	Digital input
5	D18	PC2	GPIO	SPI2_MISO	Digital input
6	D19	PC3	GPIO	SPI2_MOSI	Digital input
7	D20	PC4	GPIO	--	Digital input
8	D21	PC5	GPIO	--	Digital input

10.6 CON-E -- Digital GPIO Bank 3 (PWM-capable)

Pins D22-D25 support PWM output via TIM3 channel remapping. Note that TIM3 can only be remapped to one location -- if D22-D25 are used for PWM, then A6/A7/A8/A9 on CON-A/CON-B cannot simultaneously be used as TIM3 PWM outputs.

Pin	Label	GPIO	Type	Alt function	Typical use
1	D22/PWM	PC6	GPIO/PWM	TIM3_CH1 (remap)	Digital input or servo PWM
2	D23/PWM	PC7	GPIO/PWM	TIM3_CH2 (remap)	Digital input or servo PWM
3	D24/PWM	PC8	GPIO/PWM	TIM3_CH3 (remap)	Digital input or servo PWM
4	D25/PWM	PC9	GPIO/PWM	TIM3_CH4 (remap)	Digital input or servo PWM
5	D26	PC10	GPIO	USART3_TX	Digital input / spare
6	D27	PC11	GPIO	USART3_RX	Digital input / spare
7	D28	PC12	GPIO	USART5_TX	Digital input / spare
8	D29	PD2	GPIO	TIM3_ETR	Digital input / spare

11. STDC14 Debug Connector

The STDC14 connector provides SWD debugging, virtual COM port serial output, and SWO trace output via a single 14-pin 1.27mm SMD connector. Connect to an ST-LINK V3SET (or compatible) using the appropriate 14-pin ribbon cable.

Pin	Signal	GPIO	Notes
1-2	Reserved	--	No connection
3	T_VCC	VCC (3.3V)	Target voltage sense -- input to ST-LINK
4	SWDIO	PA13	SWD data line
5	GND	--	Ground
6	SWDCLK	PA14	SWD clock line
7	GND	--	Ground
8	SWO	PB3	Serial Wire Output trace -- ITM printf and variable plotting
9-10	NC	--	Not connected (JTAG only)
11	GNDDetect	--	GND -- pulled low to signal target connection
12	NRST	NRST	Reset
13	VCP_RX	PA9 (USART1_TX)	ST-LINK receives, MCU transmits (naming from ST-LINK perspective)
14	VCP_TX	PA10 (USART1_RX)	ST-LINK transmits, MCU receives

> *NOTE: SWO trace on pin 8 (PB3) enables non-blocking ITM printf output and live variable graphing in STM32CubeIDE's SWV ITM Data Console. This is the recommended debug output method for time-critical code -- it does not affect interrupt timing unlike UART printf.*

12. Hardware Quadrature Encoders

Three hardware quadrature encoder pairs are available, each assigned to a dedicated hardware timer. Counting is performed entirely in hardware with no CPU overhead, ensuring no pulses are missed even during intensive CAN and USB processing.

Encoder	Phase A	Phase B	Timer	Counter width	Connectors
Encoder 1	ENC1A / PB4	ENC1B / PB5	TIM3	16-bit (0-65535)	CON-B pins 6, 7
Encoder 2	ENC2A / PB6	ENC2B / PB7	TIM4	16-bit (0-65535)	CON-B pin 8, CON-C pin 1
Encoder 3	ENC3A / PA15	ENC3B / PB3	TIM2	32-bit (0-4294967295)	CON-C pins 2, 3

> *NOTE: Encoder 3 uses TIM2 which is a 32-bit timer -- ideal for high-resolution or fast-spinning encoders such as trim wheels. Encoders 1 and 2 use 16-bit timers. For applications requiring counting beyond 65535, handle overflow in firmware.*

12.1 Connecting Mechanical Rotary Encoders

Mechanical rotary encoders (as used in squawk boxes, trim wheels, and rotary selectors) require hardware debouncing to prevent phantom counts from contact bounce. Add the following circuit on the adapter board for each encoder phase:

- 1k ohm resistor in series between the IDC connector pin and the encoder terminal
- 10nF ceramic capacitor from the encoder terminal to GND

This creates a 10 microsecond low-pass filter, sufficient to suppress contact bounce at typical rotation speeds. Hall effect encoders (magnetic, no contact bounce) do not require debouncing.

12.2 STM32CubeIDE Configuration

To configure a hardware quadrature encoder in STM32CubeMX:

10. Select the timer (TIM2, TIM3, or TIM4) in the pinout view
11. Set Combined Channels to Encoder Mode
12. Set Channel 1 and Channel 2 to Input Capture Direct Mode
13. Set Encoder Mode to TIM_ENCODERMODE_TI12 (count on both edges for maximum resolution)
14. Set GPIO mode for both pins to Alternate Function Pull-Up
15. In firmware, read the counter: __HAL_TIM_GET_COUNTER(&htimX)

13. PWM and Servo Outputs

Standard hobby servo PWM uses a 50Hz signal with pulse width between 1ms (full deflection one way) and 2ms (full deflection the other), with 1.5ms at centre. The PRD-GP-STM provides multiple timer channels capable of generating this signal.

Label	GPIO	Timer	Connector	Notes
SERVO1	PA8	TIM1_CH1	CON-B pin 3	Primary servo -- TIM1 advanced control timer
TX/SERVO2	PA9	TIM1_CH2	CON-B pin 4	Available when USART1 VCP not in use
RX/SERVO3	PA10	TIM1_CH3	CON-B pin 5	Available when USART1 VCP not in use
A6/PWM	PA6	TIM3_CH1	CON-A pin 7	Servo or ADC -- select in firmware
A7/PWM	PA7	TIM3_CH2	CON-A pin 8	Servo or ADC -- select in firmware
A8/PWM	PB0	TIM3_CH3	CON-B pin 1	Servo or ADC -- select in firmware
A9/PWM	PB1	TIM3_CH4	CON-B pin 2	Servo or ADC -- select in firmware
D14/PWM	PB14	TIM15_CH1	CON-D pin 1	Servo -- no ADC conflict
D15/PWM	PB15	TIM15_CH2	CON-D pin 2	Servo -- no ADC conflict
D22/PWM	PC6	TIM3_CH1 remap	CON-E pin 1	Servo -- mutually exclusive with A6/PWM
D23/PWM	PC7	TIM3_CH2 remap	CON-E pin 2	Servo -- mutually exclusive with A7/PWM
D24/PWM	PC8	TIM3_CH3 remap	CON-E pin 3	Servo -- mutually exclusive with A8/PWM
D25/PWM	PC9	TIM3_CH4 remap	CON-E pin 4	Servo -- mutually exclusive with A9/PWM

Maximum simultaneous servo outputs without ADC or VCP conflicts: SERVO1, D14/PWM, D15/PWM, D22/PWM-D25/PWM = 7 servo channels.

> **WARNING: TIM3 channels can be mapped to either CON-A/CON-B pins (A6, A7, A8, A9) OR CON-E pins (D22-D25) via alternate function remapping -- not both simultaneously. Choose one mapping per firmware build.**

14. ADC Inputs -- Potentiometers and Hall Effect Sensors

14.1 Connecting Potentiometers

Standard 10k ohm linear potentiometers are recommended for all axis inputs (throttle, mixture, prop, rudder, etc.). Connect as follows:

- One end terminal to GND (connector pin 12, 13, or 14)
- Other end terminal to 3V3 (connector pin 11)
- Wiper (centre terminal) to the ADC signal pin (connector pins 1-8 on CON-A or pins 1-2 on CON-B)

The ADC reads 0 to 4095 (12-bit) corresponding to 0V to 3.3V on the input pin. With a 10k pot connected between GND and 3V3, full travel gives the full 0-4095 range.

14.2 Connecting Hall Effect Sensors

Hall effect position sensors (linear type, e.g. A1302, SS495) provide a ratiometric analog output suitable for direct connection to the ADC pins. Connect:

- VCC pin of sensor to 3V3 (connector pin 11)
- GND pin of sensor to GND (connector pin 12-14)
- Signal (output) pin to ADC input (connector pins 1-8 CON-A or 1-2 CON-B)

14.3 ADC Firmware Configuration

- Resolution: 12-bit (values 0-4095)
- Reference voltage: VDDA = 3.3V
- Minimum sample time for 10k source impedance: 55.5 ADC clock cycles
- Use DMA for multi-channel scanning to avoid CPU polling
- Apply 8-sample rolling average in firmware to reduce reading jitter

14.4 DAC Outputs (A4 and A5)

Pins A4 (PA4) and A5 (PA5) can be configured as 12-bit DAC outputs. These are useful for:

- Driving analog panel meters (fuel, temperature gauges)
- Driving servo position indicators
- Generating analog reference voltages

> *NOTE: When configured as DAC outputs, A4 and A5 cannot simultaneously be used as ADC inputs. Configure in firmware as either ADC or DAC per your application.*

15. Digital GPIO -- Switches and Buttons

15.1 Connecting Switches and Buttons

Digital inputs D9 through D29 are suitable for connecting switches, push buttons, toggle switches, and any other two-state digital input. The recommended circuit is:

- Connect switch between the signal pin and GND
- Add a 10k ohm pull-up resistor from the signal pin to 3V3, OR configure the STM32 internal pull-up resistor in firmware (approximately 40k ohm)
- Add a 100nF capacitor from the signal pin to GND for hardware debouncing (recommended for mechanical contacts)

In firmware, configure the GPIO as Input with Pull-Up. The pin reads logic high (1) when the switch is open, and logic low (0) when the switch is closed (connects pin to GND).

15.2 Software Debouncing

For mechanical switches without hardware debouncing capacitors, implement software debouncing in firmware:

- Sample the pin state every 1ms using a timer interrupt
- Only register a state change after the input has been stable for 10-20 consecutive samples
- This provides 10-20ms debounce time, sufficient for all mechanical contacts

15.3 Adapter Board Recommendations

An adapter board between the 14-pin IDC connector and the physical switches is recommended. A simple adapter board provides:

- JST-XH 2-pin connectors per switch (pin 1 = GND, pin 2 = switch terminal)
- 10k pull-up resistors from each signal pin to 3V3
- 100nF decoupling capacitors per input

16. Firmware Development

16.1 Development Environment

- STM32CubeIDE (latest version) -- free download from st.com
- STM32CubeMX (integrated within STM32CubeIDE) for peripheral configuration
- STM32 HAL (Hardware Abstraction Layer) drivers
- Programmer: ST-LINK V3SET for SWD debugging, or USB DFU for programming only

16.2 GPIO Configuration Quick Reference

Use case	GPIO mode	Pull	Notes
Potentiometer / hall effect	Analog	None	Disables digital buffer to reduce ADC noise
Switch / button input	Input	Pull-Up	Switch wired pin to GND
Encoder phase A or B	Alternate Function Pull-Up	Pull-Up	Timer alternate function
Servo / PWM output	Alternate Function Push-Pull	None	Timer alternate function
SPI CLK / MOSI	Alternate Function Push-Pull	None	
I2C SDA / SCL	Alternate Function Open-Drain	Pull-Up	I2C requires open-drain
VBUS sense (PC14)	Input	None	External 100k pull-down fitted on board
Status LED (PC13)	Output Push-Pull	None	Max 3mA -- series resistor already fitted on board

16.3 VBUS Detection and Role Assignment

```
void board_init(void) {
  /* Configure PC14 as VBUS sense -- external 100k pull-down fitted */
  GPIO_InitTypeDef gpio = {0};
  gpio.Pin  = GPIO_PIN_14;
  gpio.Mode = GPIO_MODE_INPUT;
  gpio.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOC, &gpio);

  CAN_Init();           /* Always initialise CAN bus */
  Peripheral_Start();   /* Always start peripheral function */

  if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_14) == GPIO_PIN_SET) {
    Bridge_StartElection(); /* USB present -- try to become bridge */
  }
}
```

16.4 Encoder Setup Example

```
/* Configure TIM3 for Encoder 1 (PB4 = ENC1A, PB5 = ENC1B) */
```

```
TIM_Encoder_InitTypeDef enc = {0};
enc.EncoderMode = TIM_ENCODERMODE_TI12;  /* Count on both edges */
enc.IC1Polarity = TIM_ICPOLARITY_RISING;
enc.IC1Selection = TIM_ICSELECTION_DIRECTTI;
enc.IC2Polarity = TIM_ICPOLARITY_RISING;
enc.IC2Selection = TIM_ICSELECTION_DIRECTTI;
HAL_TIM_Encoder_Init(&htim3, &enc);
HAL_TIM_Encoder_Start(&htim3, TIM_CHANNEL_ALL);

/* Read encoder value */
uint16_t count = __HAL_TIM_GET_COUNTER(&htim3);
```

16.5 ADC DMA Scanning Example

```
/* Configure ADC1 with DMA for channels A0-A3 (PA0-PA3) */
/* Set up in CubeMX: ADC1, DMA continuous scan, 4 channels */
uint32_t adc_values[4];  /* Populated automatically by DMA */
HAL_ADC_Start_DMA(&hadc, adc_values, 4);

/* Apply rolling average (8 samples) */
static uint32_t avg_buf[4][8];
static uint8_t avg_idx = 0;
for (int i = 0; i < 4; i++) {
  avg_buf[i][avg_idx] = adc_values[i];
}
avg_idx = (avg_idx + 1) & 7;
uint32_t smoothed[4];
for (int i = 0; i < 4; i++) {
  uint32_t sum = 0;
  for (int j = 0; j < 8; j++) sum += avg_buf[i][j];
  smoothed[i] = sum >> 3;
}
```

Firmware Resources
Complete firmware examples for common peripheral configurations including throttle quadrant, yoke, rudder pedals, encoder inputs, and the CAN-USB bridge are available at www.parkroaddesigns.com

17. Device Configuration Examples

17.1 Throttle / Mixture / Prop Quadrant

Control	Connector	Pin	GPIO	Configuration
Throttle axis	CON-A	1	PA0 (A0)	ADC CH0 -- pot between GND/3V3, wiper to A0
Mixture axis	CON-A	2	PA1 (A1)	ADC CH1 -- pot between GND/3V3, wiper to A1
Prop pitch axis	CON-A	3	PA2 (A2)	ADC CH2 -- pot between GND/3V3, wiper to A2

17.2 Yoke (Pitch, Roll, Buttons, POV Hat)

Control	Connector	Pin	GPIO	Configuration
Pitch axis	CON-A	1	PA0 (A0)	ADC CH0
Roll axis	CON-A	2	PA1 (A1)	ADC CH1
POV hat horizontal	CON-A	3	PA2 (A2)	ADC CH2
POV hat vertical	CON-A	4	PA3 (A3)	ADC CH3
Button 1-6	CON-C	4-8 + CON-D pin 1	PB2-PB13, PB14	GPIO Input Pull-Up, switch to GND

17.3 Squawk Box (4 Encoder Digits + Mode Switch)

Control	Connector	Pins	GPIO	Configuration
Digit 1 (thousands)	CON-B	6, 7	PB4/PB5	TIM3 encoder mode
Digit 2 (hundreds)	CON-B pin 8, CON-C pin 1	8, 1	PB6/PB7	TIM4 encoder mode
Digit 3 (tens)	CON-C	2, 3	PA15/PB3	TIM2 encoder mode
Mode switch	CON-C	4	PB2 (D9)	GPIO Input Pull-Up
IDENT button	CON-C	5	PB10 (D10)	GPIO Input Pull-Up

> *NOTE: The squawk box requires all three hardware encoder pairs. Each encoder digit should use a 12 or 16 PPR mechanical encoder. Apply 1k + 10nF debouncing on each encoder phase on the adapter board.*

17.4 Trim Wheel

Control	Connector	Pins	GPIO	Configuration
Trim encoder	CON-B	6, 7	PB4/PB5	TIM3 encoder mode -- use 32-bit TIM2 for high-res trim
Trim indicator servo	CON-B	3	PA8 (SERVO1)	TIM1_CH1 PWM 50Hz

17.5 Flaps Lever

Control	Connector	Pin	GPIO	Configuration
Flaps detent switch 1	CON-B	6	PB4 (D3)	GPIO Input Pull-Up
Flaps detent switch 2	CON-B	7	PB5 (D4)	GPIO Input Pull-Up
Flaps detent switch 3	CON-B	8	PB6 (D5)	GPIO Input Pull-Up
Flaps detent switch 4	CON-C	1	PB7 (D6)	GPIO Input Pull-Up
Flaps position servo	CON-B	3	PA8 (SERVO1)	TIM1_CH1 PWM 50Hz

18. Board Bring-Up Procedure

Follow this sequence when commissioning a new board for the first time.

18.1 Pre-Power Visual Inspection

16. Inspect under magnification for solder bridges on the MCU (U1, LQFP-64, 0.5mm pitch)
17. Confirm SW1 and SW2 buttons are present and not mechanically stuck
18. Confirm CANTERM1 header jumper cap is correctly fitted (end nodes only) or removed (middle boards)
19. Confirm CN1 is NOT populated unless this is the power injector board
20. Confirm LED components are oriented correctly (check cathode/anode marking)

18.2 Resistance Checks (Before Applying Power)

Test points	Expected reading	If incorrect
VCC to GND	> 1M ohm (open circuit)	Short circuit -- check LDO and MCU for solder bridges
+5V to GND	> 1M ohm (open circuit)	Short circuit -- check K7805 and filter caps
12V_IN to GND	Open circuit (TVS reverse biased)	Short circuit -- check SMBJ15A TVS orientation
NRST to GND	~10k ohm (pull-up resistor)	Missing pull-up or solder bridge on NRST
BOOT0 to GND	~10k ohm (pull-down resistor)	Missing pull-down or solder bridge on BOOT0

18.3 First Power-On (USB Only)

21. Connect USB cable -- do NOT connect 12V at this stage
22. LED_VCC illuminates immediately -- confirms LDO is producing 3.3V
23. Measure VCC with multimeter: should read 3.28-3.36V
24. Measure +5V rail: should read 4.5-4.8V
25. If any reading is incorrect: disconnect USB immediately and investigate before proceeding

18.4 First Firmware Load

26. Enter DFU mode: hold SW2, press/release SW1, release SW2
27. PC Device Manager should show STM32 BOOTLOADER under Universal Serial Bus controllers
28. Open STM32CubeProgrammer, connect via USB DFU
29. Load firmware .hex file and click Download
30. Press SW1 (RST) -- LED_STATUS illuminates confirming firmware is running

18.5 CAN Bus Test

31. Connect 12V via CN1 (if this is the power injector board) or RJ45 cable from another powered board
32. LED_12V illuminates confirming 12V supply is reaching the board

33. If ST-LINK is connected: verify CAN TX/RX activity via SWO trace in STM32CubeIDE
34. Measure CANH and CANL with oscilloscope: differential voltage during dominant bits should be approximately 2V

19. Troubleshooting

Symptom	Likely cause	Solution
LED_VCC not lit after USB connection	Power section fault	Check LDO U3 orientation. Check 5V rail for short circuit. Measure resistance VCC to GND before powering.
LED_12V not lit	12V not reaching board	Check RJ45 cable is straight-through (not crossover). Check CN1 is populated and 12V supply is on. Check SMBJ15A TVS and SS34 D3 orientation.
PC does not detect board as USB device	Firmware not running or USB fault	Try DFU mode (hold BOOT, press RST). Check USBLC6 ESD protection orientation. Verify USB cable is data cable not charge-only.
CAN bus errors on all nodes	Termination fault	Verify exactly two boards (the physical end nodes) have a jumper cap fitted on CANTERM1. All middle boards must have the jumper cap removed.
Phantom encoder counts	No debouncing on mechanical encoder	Add 1k series + 10nF to GND on each encoder phase on adapter board.
ADC readings noisy or unstable	Decoupling or layout issue	Confirm 100nF filter caps are fitted at MCU analog pins. Check pot wiring -- wiper to ADC pin, ends to GND and 3V3 only.
Multiple boards appear as separate joysticks	Multiple USB connections	Normal behaviour -- one USB connection per rig is recommended. Leader election protocol manages bridge role automatically.
Board not detected in DFU mode	BOOT0 circuit fault	Measure BOOT0 pin to GND with SW2 held -- should read ~3.3V. Check R5 (10k pull-down) and SW2 connections.
Servo jitter or incorrect position	PWM frequency or width error	Confirm TIM configuration gives 50Hz period with 1-2ms pulse width. Verify GPIO is in Alternate Function Push-Pull mode.
No SWO trace output in STM32CubeIDE	SWO not enabled or wrong clock	Enable SWV in debug configuration. Set SWO clock to 2000000 (2MHz). Confirm PB3 is configured as TRACESWO alternate function.

20. Compliance and Legal Notices

20.1 Copyright

20.2 Disclaimer

The PRD-GP-STM General Purpose Processing Board is supplied for use in flight simulation and hobbyist applications. Park Road Designs makes no warranty, express or implied, regarding the fitness of this product for any particular purpose. Park Road Designs shall not be liable for any damages, direct or indirect, arising from the use or misuse of this product.

This product is not suitable for, and must not be used in, any safety-critical, life-critical, automotive, medical, aviation, or military application.

20.3 Third-Party Acknowledgements

This product incorporates the STM32F072 microcontroller from STMicroelectronics and the TJA1051 CAN transceiver from NXP Semiconductors. All trademarks are the property of their respective owners.

20.4 Contact

Park Road Designs
www.parkroaddesigns.com
info@parkroaddesigns.com

21. Revision History

Version	Date	Changes
1.0	2025	Initial release

Appendix A -- Complete Pin Reference

Label	GPIO	Connector	Pin	Primary function	Key alternate function
A0	PA0	CON-A	1	ADC CH0	TIM2_CH1
A1	PA1	CON-A	2	ADC CH1	TIM2_CH2
A2	PA2	CON-A	3	ADC CH2	TIM2_CH3 / TIM15_CH1
A3	PA3	CON-A	4	ADC CH3	TIM2_CH4 / TIM15_CH2
A4/DAC1	PA4	CON-A	5	ADC CH4	DAC1 output
A5/DAC2	PA5	CON-A	6	ADC CH5	DAC2 output
A6/PWM	PA6	CON-A	7	ADC CH6	TIM3_CH1 / TIM16_CH1
A7/PWM	PA7	CON-A	8	ADC CH7	TIM3_CH2 / TIM17_CH1
A8/PWM	PB0	CON-B	1	ADC CH8	TIM3_CH3
A9/PWM	PB1	CON-B	2	ADC CH9	TIM3_CH4
SERVO1	PA8	CON-B	3	TIM1_CH1 servo	MCO / USART1_CK
TX/SERVO2	PA9	CON-B	4	USART1_TX / VCP	TIM1_CH2 servo
RX/SERVO3	PA10	CON-B	5	USART1_RX / VCP	TIM1_CH3 servo
ENC1A	PB4	CON-B	6	TIM3_CH1 encoder A	SPI1_MISO
ENC1B	PB5	CON-B	7	TIM3_CH2 encoder B	SPI1_MOSI
ENC2A	PB6	CON-B	8	TIM4_CH1 encoder A	I2C1_SCL
ENC2B	PB7	CON-C	1	TIM4_CH2 encoder B	I2C1_SDA
ENC3A	PA15	CON-C	2	TIM2_CH1 encoder A	SPI1_NSS
ENC3B/SWO	PB3	CON-C	3	TIM2_CH2 encoder B / SWO	SPI1_SCK
D9	PB2	CON-C	4	GPIO digital in	--
D10	PB10	CON-C	5	GPIO digital in	TIM2_CH3 / I2C2_SCL
D11	PB11	CON-C	6	GPIO digital in	TIM2_CH4 / I2C2_SDA
D12	PB12	CON-C	7	GPIO digital in	TIM1_BKIN / SPI2_NSS
D13	PB13	CON-C	8	GPIO digital in	TIM1_CH1N / SPI2_SCK
D14/PWM	PB14	CON-D	1	GPIO / TIM15_CH1	SPI2_MISO
D15/PWM	PB15	CON-D	2	GPIO / TIM15_CH2	SPI2_MOSI
D16	PC0	CON-D	3	GPIO digital in	--
D17	PC1	CON-D	4	GPIO digital in	--
D18	PC2	CON-D	5	GPIO digital in	SPI2_MISO
D19	PC3	CON-D	6	GPIO digital in	SPI2_MOSI
D20	PC4	CON-D	7	GPIO digital in	--
D21	PC5	CON-D	8	GPIO digital in	--

Label	GPIO	Connector	Pin	Primary function	Key alternate function
D22/PWM	PC6	CON-E	1	GPIO / TIM3_CH1 remap	USART6_TX
D23/PWM	PC7	CON-E	2	GPIO / TIM3_CH2 remap	USART6_RX
D24/PWM	PC8	CON-E	3	GPIO / TIM3_CH3 remap	--
D25/PWM	PC9	CON-E	4	GPIO / TIM3_CH4 remap	--
D26	PC10	CON-E	5	GPIO spare	USART3_TX
D27	PC11	CON-E	6	GPIO spare	USART3_RX
D28	PC12	CON-E	7	GPIO spare	USART5_TX
D29	PD2	CON-E	8	GPIO spare	TIM3_ETR