

Interface Name	Interface Type	Specifics
USB_Connection	USB Port	Vmax = 5.2V Vmin = 4.2V Imax = 500mA Inom = 100mA
Motor_Control	Two Wires	V ~ 5-24+ V Imax ~ 4A Inom ~ 500ma
Encoder_Data	Four wires (Quadrature)	+ = 5V - = GND A = Square wave (period offset) signal B = Square wave (period offset) signal
Motor_Wires	Two wires	V = 12V Imax ~ 8.5A Inom ~ 500mA
Firmware_Math	Code	Function Call with floating point 'error' value. Expected input varies where error is target-actual. Returns expected change for system.
Firmware_Constant_Set	Code	Function Call with Floating point data. Expected input is a floating point constant. Could be any constant with no limit.
Firmware_Limit_Check	Code	Function call with floating point position data. Returns True if position is valid. Expected input is a floating point number.to be compared with the saved floating point limits.
Firmware_Limit_Save	Code	Function call with floating point position limits. Expected input is a floating point value to be compared with actual input to determine if the system is out of it's limit.

Quadrature_Encoder_Software_Link	Two Wires (4 with power and gnd) Software Link	V+ = 3.3V V- = 0V Software Counter that counts up when encoder move in positive direction and down when it moves in negative direction. Direction is also stored.
USB_Link	Two wires (3 with gnd; power not used) Software link	V+ = ~2.8V V- = ~0.3V Voltage values depend on differential signaling on D+ and D- Software link is an ST provided middleware module that handles setup, receive, transmit, and teardown. Send is synchronous function while receive uses a callback function that is triggered when data is available to read from the bus.
PWM_Output_for_Motor_Drive	Two wires: PWM and PWM_N (PWM complement) Software link	V+ = ~3.3V V- = 0V Integrated advanced timers in the STM32 are used to generate the PWM signals. The hardware is set up to generate a PWM signal and its complement. This saves the cost of a not gate in an h-bridge driver configuration. The duty cycle of the PWM is configured through the setting of registers in the microcontroller.
GUI_User_Inputs	Text input (code)	At least 3 text input fields for motion control parameters. Input must be float value and within range shown on GUI (where range is easily modified in code).
GUI_Encoder_Output	Text output (code)	At least 1 text output field for encoder data. Include description and float value for each piece of data from the

		encoder.
GUI_Input_To_Firmware	Code	Code-behind function to send user input to the system. Check that value is valid (eg. float and within range) and send to firmware. Send data via Serial.
Encoder_To_GUI	Code	Code-behind function to send encoder data from system to GUI. Receive data via Serial.