USB Control SPEC V 2.00

Monday, March 01, 2010
3:59 PM

Contact:[redacted] with questions

Changes:
states new requirements for abiding to SWARMS standard
Easier to read.

Endpoint Types

Motor control	Controls Motor
Power Management	Used to get battery status etc.
Logic board	Hardware IO.

Software Implementation

BASICS:

BUS	LibUSB Driver
Command Timeouts	~ 1 second no activity

All commands listed as numbers are sent as decimal bytes unless otherwise marked
^ indicates a command that is given an example

The basic visual representation of sending a command frame:
To the coordinator from processor:

Command ID (decimal byte)	Payload length (decimal byte)	Payload data

Usually responding a good response:
From the coordinator:

ACK	0 (no payload)

Usually responding a bad response:
From the coordinator:

Abort	1 byte long	(the same) command ID

Main Default Commands

Command Name	Decimal (byte)	Payload Length(byte)	Payload data	Return	Synopsis
Ack -Afirm	1	0	NA	1	Acknowledge
Abort (stop)	2	1	Command active	1 or 0 (no command)	Aborts an active command
Status	3	1	Command to poll	0 (not active) or string	If command is active sys returns a representative string (depending on the command)
Reset	4	1 or 0	Endpoint ID	1	Req to reset all systems (0 payload) or specifics (endpoint)
^LED Wink	5	1	LED to blink	1 = ack	Flashes a built-in led --- to set leds on or off use digital outputs
Validate Command	6	1	Command ID	1 -- available 2 -- Unknown 3 -- Not available	Validates a command to a controller to see its status or availability
SYS Version	7	0	0	Double as string w/ version	Returns a string representative of vers # (EX: V 1.01 returns "101")

Standard command system (Std) =

1 Ack	returned when command is success -- ('1'ack, 1(payload length) ,commandID)
2 BadAck	returned when command fails --('2'bad-ack, 1(payload length) ,commandID + special codes)
3 Not Available	Returned when command is not available -- ('3'not-available, 1(payload length) ,commandID + special codes)

Statement	Explanation
Returns set value when payload is 0? Zero Value Fetch (ZVF)	When command is called with 0 payload length, the systems returns (command ID, payload length, representation equivalent to 'Payload data bytes')

Motor Command	Byte	Payload Length	Payload data bytes	Return	Synopsis -- main purpose	Returns set value when payload is 0?
Base Radius	10	# byes in payload	# cms as string	1 - Std ack	Set radius of wheel to wheel dist.	Yes
Wheel Radius	11	#BTS PYLD	# cms as string	1 - Std ack	Set BASIC wheel radius	Yes
^ Move Robot CM distance	12	#BTS PYLD	# cms as string	1 - Std ack	Set wheels to move forward or backward *	Yes ,CM remaining
Zero-Point turn	13	#BTS PYLD	Degrees -string Like: -120 for *-120	1 - Std ack	Zero point turn based on wheel diameter and base diameter	NO-- returns abort
Ratio-drive distance	14	#BTS PYLD	#cms as string (?)	1 - Std ack	Using the locked, predefined ratio, drive this many cms	Yes -- cms remaining
Ratio-drive right segment	15	#BTS PYLD	The string-represented #	1 - Std ack	Set the right part of the wheel ratio L:R	Yes
Ratio-drive left segment	16	#BTS PYLD	The string-represented #	1 - Std ack	Set the left part of the wheel ratio L:R	Yes
Ratio-mode cm/s speed	17	#BTS PYLD	# cm/s as string	1 - Std ack	Set/get the avg requested cm/s speed	Yes
Ratio-mode is set	18	#BTS PYLD	1 for on or 0 for	1 - Std ack	Enables or disables ratio mode subsystem	Yes (as decimal)
Set Abs Speed	19	#BTS PYLD	String of speed int	1 - Std ack	Sets absolute motor speed with int from +/- 100 *	Yes string int current speed
Set Drive DIR	20	#BTS PYLD	String of dir int	1 - Std ack	Sets absolute wheel turn with int from +/- 100 *Only* in V#1*	Yes -- when supported
Vehicle type	21	#BTS PYLD	Byte of vehicle type index	1 - Std ack	Sets vehicle type that affects supported actions	Yes -- returns same index

* Automatically disabled with ratio mode enabled turns not available

Sensors/Power/IO

Command	Byte	Length	Data	Return command	Ret data	Synopsis
Get analog input	40	# bytes	IDs of analog sensor wanted	40 unless invalid then Std	Sensor data double string	Gets & returns analog sensor values
Get digital input	41	# bytes	IDs of dig input	41 or Std	0 or 1	Get sens value unless sens is invalid or unavailable
digital out	42	# bytes or 0	Byte # 1 = output id bytes # 2 - inf. = output value	Std	STD ; if sent payload byte 2 =0 then current value returned (ZVF)	Set digi outs to values If only sensor and no value in payload then ret current value
Digital to analog	43	# bytes or 1	Byte # 1 = output id # 2 = output value	43	Std. if sent payload byte 2 =0 then ZVF	sets DAC value for specified sensors

Vehicle Types -- V#

Index V#	Name	Description	Special Limitations/capabilities
1	4 wheel front turn OBSC	Four wheel vehicle with front rotating wheels	Drive wheel direction
2	4 wheel square Differential	Square vehicle that has drive in corners	
3	Differential circle	Round vehicle that has wheels positioned so it	
4	Triple Omni drive	Three omni wheels that rotate to turn and drive	NOT implemented!

Upper and Lower Parameter Value Limits

Name	Lower Limit	Upper Limit	Guaranteed	Info
LED	1	255	2	At least two LEDs to wink
Set Drive DIR	"-100"	"100"	n.a.	These are strings including "0"
Digital Out	1	255	0	Up to 255 outs -- not counting ZDF
Analog Out	1	255	0	Up to 255 outs -- not counting ZDF
Analog In	1	255	0	Up to 255 inputs
Digital In	1	255	0	Up to 255 inputs

Examples

Where ' ' represent ascii bytes and all others are decimal

Move Robot CM distance
Move robot backward 12.2 cms
Processor Sends:

12	5	'-'	'1'	'2'	'.'	'2'

Coordinator Sends

1	0

LED Wink
Wink LED 2
Processor Sends:

5	1	2

Coordinator Replies:

1	0

Hardware Implementation

Requirements

Input/Outputs:

Should be labeled with following conventions

Common Name	Prefix	Numbering	Note
Digital Output	DO	1 - inf.	EX: DO1
Analog Output	AO	1 - inf.	EX: AO5
Digital Input	DI	1 - inf.	EX: DI1
Analog Input	AI	1 - inf.	EX: AI3

Should have a three wire connections for IO where

Black - GND (-)	White - Data (Output pin (AO/DO))	Red - Vcc (+)

Three wire setup is using .1" spaced pin header
Pinout labels should either:

Appear to the left of the GND Pin EX:

DO1 (* * *)
[b w r]

Appear above the (* * *) with GND Pin on the leftmost part of the setup EX:

DI4
(* * *)
[b w r]