Probe-Scope CDC Interface Spec v1.0

Overview

The Probe-Scope hardware (specifically the PIC32MZ EF microprocessor) communicates with the Python host application over USB as a Communications Device Class (CDC) device. This operates basically as a virtual COM port. In general, all serial control information will be in ASCII, but actual data will be in binary/raw format. The protocol will operate in two paradigms: sample data can be requested with one command that will result in a bulk return; control registers for all user-controllable aspects of the hardware can be accessed through read and write commands that operate on a Virtual Memory Space.

Presently, the hardware operates in a limited capacity to avoid the complexity of interfacing with the HyperRAM device. As such sample data can be sent in its entirety in one transmission. At such time as the HyperRAM is implemented and sample data can be much longer than the PIC can handle at once, this protocol will be amended to accommodate multi-block sample data.

General Format

The protocol will adopt ASCII Record Separator (RS, 0x1E) as the start of message symbol, ASCII End of Transmission (EOT, 0x04) as the end of message symbol, ASCII End of Transmission Block (ETB, 0x17) as a for-future-use symbol, and ASCII Substitute (SUB, 0x1A) as the escape character. These four symbols are reserved and must not appear in the serial data except to invoke their functionality. If the data includes one of these symbols, it must be escaped: sending the escape character (SUB, 0x1A) will cause the next character to be interpreted literally.

Escapement Example

Raw Index	Message Index	Byte	Description
47	47	0x1D	Literal 0x1D
48	N/A	0x1A	Escape
49	48	0x1E	Literal 0x1E (Would be start of message without escape)
50	49	0x1F	Literal 0x1F

Transmissions must always begin with a start of message symbol (RS, 0x1E) and end with an end of message symbol (EOT, 0x04). The second byte of every transmission will be “C” (0x43) to indicate a command, or “R” (0x52) to indicate a result. The third byte of every transmission will be the command ID (detailed below). The next bytes and the message length will be determined by the command itself. As USB CDC uses bulk transfers which use CRC and have guaranteed delivery according to the protocol, no data integrity checks will be included in this protocol.

Transmission Format

Index	Byte	Description
0	0x1E	Start of Message
1	0x43 or 0x52	Command or Result, respectively
2	Any non-reserved	Command ID
…	…	Command/result data
N	0x04	End of Message

Command Formats

Each command will have an ID which is any byte not on the reserved symbols list (and which should be a lowercase ASCII character). Each command will have a message length which is either fixed or which can be directly calculated from a fixed-length subset of its data at the beginning of the command. Commands will be either from Host to Hardware or vice versa, and are not valid commands in the opposite direction.

Request Sample Data (s, 0x73) (Host to Hardware)

This command will request the full sample memory, aligned with a trigger (the trigger will be in the exact center of the returned data). Naturally, the result will not be transmitted until the trigger has occurred and the second half of the sample buffer has been filled.

The command has a fixed message length of 4 bytes. The result will include the length of the sample data and its message length will be that value plus 10 bytes.

Command Format

Index	Byte	Description
0	0x1E, RS	Start of Message
1	0x43, C	Command Indicator
2	0x73, s	Command: Request Sample Data
3	0x04, EOT	End of Message

Result Format

Index	Byte	Description
0	0x1E, RS	Start of Message
1	0x52, R	Result Indicator
2	0x73, s	For Command: Request Sample Data
3	0x4C, L	Sample Data Length Field Indicator (32-bit, little-endian, unit: bytes)
4	…	Sample Data Length LSB
5	…	Sample Data Length …
6	…	Sample Data Length …
7	…	Sample Data Length MSB
8	0x44, D	Sample Data Field Indicator
…	…	Sample Data of length (in bytes) as Sample Data Length Field
N	0x04, EOT	End of Message

Triggered (t, 0x74) (Hardware to Host)

This command will indicate that the Hardware has been triggered. It will be transmitted by the Hardware automatically when the trigger condition has been met, but will not be transmitted if the device has sent the Request Sample Data command and is awaiting a response. This command should not be used by the Host for timing purposes as it will be delayed if another command is already in progress; it is for information purposes only.

The command has a fixed message length of 4 bytes. There is no result for this command.

Command Format

Index	Byte	Description
0	0x1E, RS	Start of Message
1	0x43, C	Command Indicator
2	0x74, t	Command: Triggered
3	0x04, EOT	End of Message

Write Registers (w, 0x77) (Host to Hardware)

This command will write data of a specified length to a specified address in the Virtual Memory Space. The result will contain the number of bytes actually written; if this differs from the number of bytes requested to write, that constitutes a failure of the command. In the case of a failure of the command, the Hardware may either write no data and return 0 bytes written, or it may write as many bytes as it can and return the actual number of bytes written.

The command will include the length of the data to write and its message length will be that value plus 14 bytes. The result will have a fixed message length of 9 bytes.

Command Format

Index	Byte	Description
0	0x1E, RS	Start of Message
1	0x43, C	Command Indicator
2	0x77, w	Command: Write Registers
3	0x41, A	Write Data Address Field Indicator (32-bit, little-endian, unit: bytes)
4	…	Write Data Address LSB
5	…	Write Data Address …
6	…	Write Data Address …
7	…	Write Data Address MSB
8	0x4C, L	Write Data Length Field Indicator (32-bit, little-endian, unit: bytes)
9	…	Write Data Length LSB
10	…	Write Data Length …
11	…	Write Data Length …
12	…	Write Data Length MSB
13	0x44, D	Write Data Field Indicator
…	…	Write Data of length (in bytes) as Write Data Length Field
N	0x04, EOT	End of Message

Result Format

Index	Byte	Description
0	0x1E, RS	Start of Message
1	0x52, R	Result Indicator
2	0x77, w	For Command: Write Registers
3	0x4C, L	Written Data Length Field Indicator (32-bit, little-endian, unit: bytes)
4	…	Written Data Length LSB
5	…	Written Data Length …
6	…	Written Data Length …
7	…	Written Data Length MSB
8	0x04, EOT	End of Message

Read Registers (r, 0x72) (Host to Hardware)

This command will read data of a specified length from a specified address in the Virtual Memory Space. The result will contain the number of bytes actually read; if this differs from the number of bytes requested to read, that constitutes a failure of the command. In the case of a failure of the command, the Hardware may either read no data and return 0 bytes read, or it may read as many bytes as it can and return the actual number of bytes read.

The command has a fixed message length of 14 bytes. The result will include the length of the read data and its message length will be that value plus 10 bytes.

Command Format

Index	Byte	Description
0	0x1E, RS	Start of Message
1	0x43, C	Command Indicator
2	0x72, r	Command: Read Registers
3	0x41, A	Read Data Address Field Indicator (32-bit, little-endian, unit: bytes)
4	…	Read Data Address LSB
5	…	Read Data Address …
6	…	Read Data Address …
7	…	Read Data Address MSB
8	0x4C, L	Read Data Length Field Indicator (32-bit, little-endian, unit: bytes)
9	…	Read Data Length LSB
10	…	Read Data Length …
11	…	Read Data Length …
12	…	Read Data Length MSB
13	0x04, EOT	End of Message

Result Format

Index	Byte	Description
0	0x1E, RS	Start of Message
1	0x52, R	Result Indicator
2	0x72, r	For Command: Read Registers
3	0x4C, L	True Read Data Length Field Indicator (32-bit, little-endian, unit: bytes)
4	…	True Read Data Length LSB
5	…	True Read Data Length …
6	…	True Read Data Length …
7	…	True Read Data Length MSB
8	0x44, D	Read Data Field Indicator
…	…	Read Data of length (in bytes) as True Read Data Length
N	0x04, EOT	End of Message

Virtual Memory Space

Registers from the PIC, PGA, DAC, and FPGA are all mapped into a common space for ease of access by the Host software. The following table describes that map. **Note that some or all read operations may be unimplemented. Note that limitations of the target hardware on reading or writing must be respected (i.e. word alignments), failure to do so constitutes undefined behavior.**

Virtual Memory Map

Start Byte	End Byte	Target Device	Usage
0x00000000	0x0000FFFF	Reserved	
0x00010000	0x00010007	PIC	Protocol Debugging
0x00010008	0x0001FFFF	PIC Reserved	
0x00020000	0x00020001	FPGA	Acquisition Settings
0x00020002	0x0002FFFF	FPGA Reserved	
0x00030000	0x00030001	PGA	Filter Settings
0x00030002	0x0003FFFF	PGA Reserved	
0x00040000	0x00040010	DAC	Vertical Offset and VGA
0x00040011	0x0004FFFF	DAC Reserved	
0x00050000	0xFFFFFFFF	Reserved	

PIC Register Map

Address	Type	Description
0x00020000	uint8_t[8]	Meaningless Buffer (for protocol debugging)

FPGA Register Map

Address	Type	Description
0x00020000	int8_t	Trigger Level
0x00020001	uint16_t	Decimation Factor

PGA Register Map

Address	Type	Description
0x00030000	Bitfield (15 bits)	Configuration (see ADRF6518 datasheet, Table 5)

DAC Register Map

Address	Type	Description
0x00040000	Bitfield (17 bits)	Channel A (VGN1) Configuration (see MCP4728 DS, Table 4-1)
0x00040002	Bitfield (17 bits)	Channel B (VGN2) Configuration
0x00040005	Bitfield (17 bits)	Channel C (VGN3) Configuration
0x00040008	Bitfield (17 bits)	Channel D (Vertical Offset) Configuration