BTLE Protocol Specification

Version 1.1

Last Revised: April 28, 2017

Change Log

1.0	Initial Release
1.1	Update Scope State Characteristic to move 0x00 byte from byte 19 to byte 0. Update FPGA register map trigger ctrl register.

This document will explain the protocol used to control and communicate with the Aeroscope Wireless Oscilloscope. Aeroscope uses Bluetooth 4.0 with the Generic Attribute Profile (GATT) to communicate with a central device (such as a phone or tablet). The central can identify available Aeroscopes to connect to using its service's 128 bit long Universally Unique Identifier (UUID).

Aeroscope's Service UUID: F9541234-91B3-BD9A-F077-80F2A6E57D00

Once the central has connected to Aeroscope it will use its service's four characteristics to communicate. Each characteristic is 20 bytes long, even if the command being sent only requires one character the other 19 bytes must be zero padded. The sections below will detail the available controls for each characteristic and how these can be used to communicate with Aeroscope.

Scope State Characteristic (0x1237) (Write)

The Aeroscope system consists of an Analog Front End (AFE), a field programmable gate array (FPGA), and an ARM processor with an integrated BTLE radio. After the measured signal has been filtered and attenuated/amplified by the AFE, it is digitized and the digitized data is fed into the FPGA. The FPGA provides most of the oscilloscope functionality. It includes a sampler to modify the equivalent sample rate, the trigger system to detect trigger events, the memory system to store and readback the measured data, a controller for the AFE and the DAC, among other things.

The scope state characteristic is used to write to the FPGA's writeable registers. Each byte maps to an FPGA register, with the least significant byte always being set to 0x00. See Table 1 for a map of this register and Table 5 for a complete list of all available FPGA registers.

Scope In Characteristic (0x1236) (Write)

Similarly to how the state characteristic is used to control the FPGA, the In characteristic is used to control the ARM. The In characteristic consists of multiple messages used to generally control the state of the system. See Table 2 for a map of this register.

Run Mode: Sending an "R" to this characteristic will put the scope into run mode. Aeroscope will continuously acquire frames of data and send them to the central.

Run Mode

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
'R'																			

Stop Mode: Sending an “S” to this characteristic will put the scope into stop mode. Aeroscope will no longer send over frames of data. However, if a frame was already requested and Aeroscope hasn’t received a trigger yet, it will send over this last frame as soon as it is triggered. You can prevent this “extra frame” by using the cancel command shown below.

Stop Mode

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘S’																			

Single Frame: Sending “F” will put the scope into single shot mode. Aeroscope will send over the next frame it captures and then go into stop mode.

Single Shot

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘F’																			

Full Frame: Sending “L” will send over all data that has been previously written to memory and then go into stop mode. It will not acquire a new frame but just send over the data that was previously acquired. In normal operation, Aeroscope will write more memory than is read. For all but the slowest time scales, writing is very fast compared to reading. The latency experienced by the user is unchanged if we write 512 samples to memory versus 4096 samples to memory. However, the latency is noticeably worse if we read 4096 samples versus 512 samples. This full frame command can be used to selectively download all of the memory.

For example, you could send a single shot command and once the single shot frame is received, send a full frame command. The single shot frame will quickly send over 512 samples, and then the full frame will send over the rest of the memory (including the same 512 samples that were already sent). The user can then browse through much more captured data in stop mode and the latency isn’t noticeable since they are looking at a stopped waveform instead of video data.

Full Frame Download

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘L’																			

Cancel Frame: Sending an “X” will cancel any previously requested frames. For example, if the scope is put into stop mode after a frame was already requested, the scope will still send over the frame once it receives a trigger. The FPGA doesn’t know if the scope is in run mode or stop mode, it receives frame requests and then sends them over as soon as it has been triggered and written the appropriate amount of data to memory. This cancel command can be used to prevent any future frames from being sent. Keep in mind that sending this cancel frame command over after requesting a full frame will cancel the full frame request.

Cancel Frame

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘X’																			

Cal: Sending a “CI” will put Aeroscope into a calibration routine that will determine the offset error for each voltage channel. For more information regarding the calibration, see the scope out characteristic section. Sending a “CX” will clear prior calibration parameters.

Start Calibration

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘C’	‘I’																		

Clear Calibration

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘C’	‘X’																		

Sleep: Sending “ZZ” will put Aeroscope into its lowest power state. In this state, Aeroscope will not advertise and can only be “woken up” by pressing the button or plugging in a charger. Sending “ZR” will reset the MCU on Aeroscope, and power cycle the FPGA.

Deep Sleep

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘Z’	‘Z’																		

Reset

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘Z’	‘R’																		

Scope Name: Sending "N" followed by 19 characters will change Aeroscope's advertised name. This 19 character name will be written to FLASH so that Aeroscope will retain its name even if it runs out of battery or is reset.

Change scope name

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
'N'	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x = any character
Null terminate string to indicate end of name

Power State: Sending "PF" will fully turn on Aeroscope, "PO" will turn off the FPGA. When the FPGA is off Aeroscope will not be able to send frame data to the central. The power state will automatically be set to fully on when Aeroscope connects to a central, but there is some delay before this happens due to the FPGA configuration time. The central should not request frames until the power state is fully on.

Query: The "Q" commands will query data from Aeroscope. The data will be returned using the scope out characteristic. For more information on the returned data format, see the scope out characteristic section. The following queries are supported:

"QTI" – Queries telemetry data. Aeroscope will respond with charging information, battery voltage, and current temperature

"QVR" – Queries version data. Aeroscope will respond with the ARM firmware revision, the FPGA firmware revision, and the device serial number

"QE" – Queries the error log. Aeroscope keeps a log of the last 19 errors that have occurred.

"QC" – Queries calibration parameters.

"QP" – Queries the current power state.

Query Telemetry

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
'Q'	'T'	'I'																	

Query Version

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
'Q'	'V'	'R'																	

Query Power State

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘Q’	‘P’																		

Query Calibration Parameters

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘Q’	‘C’																		

Clear Error Log: Sending “EX” will clear the error log.

Clear Error Log

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘E’	‘X’																		

Scope Data Characteristic (0x1235) (Read/Notify)

The scope data characteristic is used to receive measured frame data from Aeroscope. The central must subscribe to receive notifications from this characteristic. Frame data is sent over in packets of 20 bytes. See Table 3 for a map of this register.

The first packet in a frame is referred to as a start of frame packet (SOF). It contains a frame size code in the first byte to indicate how big the frame is (i.e. how many bytes the central should expect to complete the frame). The second byte of the first packet indicates the subtrigger value, subtrigger is explained below. The remaining 18 bytes in the first packet are measurement data.

The frame size codes are:

0x01 = 16 sample frame
0x06 = 512 sample frame
0x09 = 4096 sample frame

All subsequent packets after the SOF packet will have a first byte equal to 0x00. This indicates that they are continuation of frame packets, the remaining 19 bytes in the packet are measurement data. There may be zero padding characters at the end of the last packet.

The subtrigger is used to correct for phase uncertainty between the ADC sample timing and the waveform being measured. This phase uncertainty can cause the displayed waveform to appear jittery since the sample points are lining up slightly differently on the waveform for each frame. This jitter appears worse as the number of sample points on screen are reduced. When displaying 512 sample points on screen this jitter is barely noticeable, but if we reduce the displayed sample points to 256 or 128 the jitter becomes very noticeable.

To solve this phase uncertainty issue, Aeroscope measures the delta between the trigger threshold and the two adjacent samples. Using these deltas, we can determine an amount to shift the waveform in time to compensate for the phase uncertainty. The subtrigger value is a 6-bit number that indicates how to shift the waveform on screen. Divide the subtrigger value by 64 and shift the waveform by the corresponding fraction of a sample interval. A larger value indicates that the sample after the trigger event is further away from the trigger threshold and the sample just prior to the trigger event is closer to the trigger threshold. A mid-scale value, i.e. 31, indicates that the trigger event occurred just between the two adjacent sample points so the waveform would be shifted by half a sample interval. A value of 0 indicates that no shift is necessary.

Scope Out Characteristic (0x1239) (Read/Notify)

The scope out characteristic is used to read data other than measured frame data from Aeroscope. The central must subscribe to receive notifications from this characteristic. The data sent over can be in response to a query (see scope in characteristic section) or can be unsolicited data sent over automatically. For example, telemetry data is sent over every thirty seconds when Aeroscope is connected to a central. Aeroscope will also send over its power state after a connection, once the FPGA has configured. See Table 4 for a map of this register.

Telemetry Data: A packet received from this characteristic with a first byte of "T" is a telemetry packet. The second byte will be battery charging information, the third byte will be battery voltage level and the fourth and fifth bytes will be temperature data.

A. Charging Byte – The MSBit indicates whether a charger is currently plugged in to Aeroscope, and the second MSBit indicates whether Aeroscope is currently charging. For example, a value of 0xC0 would indicate that both a charger is connected and it is charging. A value of 0x80 would indicate that a charger is connected but Aeroscope is no longer charging (i.e. fully charged)
B. Battery Level Byte – This number represents the current battery charge level. Values above 238 indicate a fully charged battery, values from 226 – 238 are a partially charged battery, and values from 220 – 226 indicate the battery is in a low charge state.
C. Temperature Bytes – The temperature is represented as a 16-bit number with the fourth packet word being the MSByte and the fifth packet word being the LSByte. This number is ten times the current temperature in degrees Celsius. For example, a value of 0x00FB would correspond to a current temperature of 25.1 C.

Telemetry Response Packet

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
'T'																			

	Charger Status
	Battery Voltage
	Temperature

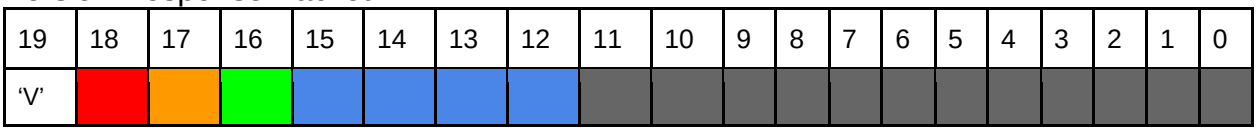
Charger Status Bits

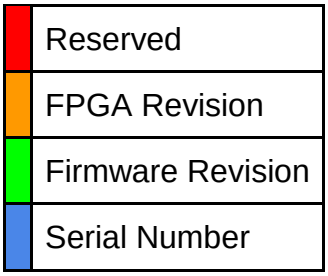
7	6	5	4	3	2	1	0
Charger Connected	Charging						

Version Data: A packet received with a first byte of "V" is a version packet. The second byte is reserved for internal use, the third byte is FPGA firmware revision, the fourth byte is MCU firmware revision, and bytes five through eight are device serial number.

A. FPGA Firmware Revision – The version number for the FPGA firmware.
B. MCU Firmware Revision – The version number for the MCU firmware.
C. Device Serial Number – Each device is assigned a 32 bit serial number.

Version Response Packet





Error Log: A packet that begins with a first byte of "E" is a record of the last 19 errors that Aeroscope has recorded. Each error is indicated by a one byte error code.

Critical Error: If Aeroscope experiences a serious error that it can't recover from it will immediately send up a critical error notification. This will be a packet that begins with "EC" followed by the error code for the critical error.

Calibration Parameters: Aeroscope has a built-in calibration routine to determine the offset error for each voltage setting. The results of this routine are used to correct for any offset error in the AFE. Once the calibration routine has finished it will send a packet beginning with “CB” followed by the 16-bit offset values for each voltage setting. These offset values should be added to the DAC setting for each voltage setting.

For example, mid-scale on the 16-bit offset DAC should correspond to 0 V at the ADC and on screen. If there is any offset error, the DAC code to display 0 V needs to be shifted to compensate for this error. The numbers returned by the calibration routine will indicate how many DAC counts the DAC code needs to be offset.

Cal Response Packet

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘C’	‘B’																		

	10V Cal Offset
	5V Cal Offset
	2V Cal Offset
	1V Cal Offset
	500mV Cal Offset
	200mV Cal Offset
	100mV Cal Offset

Button Press: A packet beginning with “BP” indicates that the button has been pressed. The central should toggle the scope mode as a result of the button press, e.g. if the scope is in “Run” it should send a command to put the scope into “Stop”.

Button Pressed

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘B’	‘D’																		

Power State: A packet beginning with “P” is power state message. The second byte will be either an “F” to indicate that the power is fully on and the FPGA is configured or an “O” to indicate that the power isn’t yet fully on or the FPGA isn’t finished configuring. The power state will be sent up upon connection once the FPGA has configured. The power state can also be queried. The central should not put the scope into run mode or single mode until the power state is fully on.

Power On

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘P’	‘F’																		

Power Off

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
‘P’	‘O’																		

BTLE Characteristics Register Map

0x1237 (Write)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write FPGA registers	REG 0	REG 1	REG 2	REG 3	REG 4	REG 5	REG 6	REG 7	REG 8	REG 9	REG 10	REG 11	REG 12	REG 13	REG 14	REG 15	REG 16	REG 17	REG 18	0x00

Table 1. Scope state characteristic register map

0x1236 (Write)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Run Mode	"R"																			
Stop Mode	"S"																			
Run calibration routine	"C"	"I"																		
Clear calibration data	"C"	"X"																		
Get single frame	"F"																			
Get full frame	"L"																			
Cancel Frame	"X"																			
Enter factory ship mode	"Z"	"Z"																		
Reset	"Z"	"R"																		
Set Scope Name	"N"																			
Set power state to full on	"P"	"F"																		
Set power state to full off	"P"	"O"																		
Query telemetry	"Q"	"T"	"I"																	
Query version	"Q"	"V"	"R"																	
Query error log	"Q"	"E"																		
Query calibration parameters	"Q"	"C"																		
Query power state	"Q"	"P"																		
Clear error log	"E"	"X"																		

Table 2. Scope in characteristic register map

0x1235 (Read/Notify)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Scope Frame Data	Packet Header	SubTrig or Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data

Table 3. Scope data characteristic register map

0x1239 (Read/Notify)	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Telemetry	"T"	Charger	Battery	H Temp	L Temp															
Version	"V"	HW ID	FPGA Rev	Firmware Rev	[31: 24] SN	[23: 16] SN	[15: 8] SN	[7:0] SN												
Error Log	"E"																			
FPGA failed to config	"E"	"C"	0xC0																	
FPGA deconfigged	"E"	"C"	0xC1																	
Cal error	"E"	"C"	0xC6																	
Cal Params	"C"	"B"	H_10V	L_10V	H_5V	L_5V	H_2V	L_2V	H_1V	L_1V	H_500mv	L_500mv	H_200mv	L_200mv	H_100mv	L_100mv				
Button Press	"B"	"D"																		
Power	"P"	"F/O"																		

Table 4. Scope out characteristic register map

FPGA Register Map

Address	Name	Register Size (bits)	Default Value	Description	Notes
0x00	Trigger Ctrl	8	0x03	[7] = NULL [6] = NULL [5] = Noise Reject Trigger Function En [4] = NULL [3] = NULL [2] = Falling Edge Trigger En [1] = Rising Edge Trigger En [0] = Auto Trigger En	Enables various trigger modes
0x01	Trigger Set Pt	8	0x80	ADC sample threshold to locate the trigger, 0 - 255	
0x02	PLL Ctrl	8	0xC5	Not Currently Used	
0x03	Front End Ctrl	8	0xE0	[7] = DC Couple En [6] = Hi Gain En [5] = Low Gain En [4] = NULL [3] = NULL [2] = Atten Mux S2 [1] = Atten Mux S1 [0] = Atten Mux S0	100mV/div = 0x60 200mV/div = 0x41 500mV/div = 0x20 1V/div = 0x22 2.5V/div = 0x03 5V/div = 0x04 10V/div = 0x05

0x04	Sampler Ctrl	8	0x00	Sets the divide ratio for the ADC sample clock. Uses a scientific notation type format. The value of the upper five bits is multiplied by the lower three bits raised to the power of 10. For example, 0x09 = 00001001 = 1 * 10^1 = 10. 0x51 = 01010001 = 10 * 10^1 = 100. The default sample rate (i.e. when the divide ratio is 1) is 100 MSPS. 500ns : 0x01 1us : 0x10 2us : 0x20 5us : 0x09 10us : 0x11 20us : 0x21 50us : 0x0A 100us : 0x12 200us : 0x22 500us : 0x0B 1ms : 0x13 2ms : 0x23 5ms : 0x0C 10ms : 0x14 20ms : 0x24 50ms : 0x0D 100ms : 0x15 200ms : 0x25 The 4 highest codes for this register are used to enable roll mode: 0xE7: 500ms/div, 10ms between samples 0xEF: 1s/div, 20ms between samples 0xF7: 2s/div, 40ms between samples 0xFF: 5s/div, 100ms between samples	support divide ratios of 2, 4, 10, 20, 40, 50, 100, 200, 400, 500, 1000, 2000, 4000, 5000, 10000, 20000, 40000, 50000, 100000, 200000, 400000 Any value that isn't recognized will result in a divide ratio of 1. For example, setting the register to 0 will result in a divide ratio of 1 = 100 MSPS sampling rate
0x05	Trigger X Pos[11:8]	4	0x8	Location of trigger in memory, total memory will be the size of write sample depth	
0x06	Trigger X Pos[7:0]	8	0x00	A setting of 0 will be all post trigger data, a setting of (write sample depth) will be all pre trigger data, a setting half the size of write sample depth will result in equal post and pre trigger data being captured	
0x07	Read Memory Start Address [11:8]	4	0x7	Memory location of the first sample to be read from memory. This and a number of the subsequent samples (set by the read sample depth register) will be read from memory.	

0x08	Read Memory Start Address [7:0]	8	0x00		
0x09	Write Sample Depth	4	0x9	1 => write record length = 16 6 => write record length = 512 9 => write record length = 4096	number of samples to write to memory, all samples written to memory will be read when Aeroscope enters stop mode
0x0A	Read Sample Depth	4	0x6	1 => read record length = 16 6 => read record length = 512 9 => read record length = 4096	number of samples to read from memory. All samples written will be read when Aeroscope enters stop mode
0x0B	DAC Control [15:8]	8		Sets the output voltage of the offset DAC. This 16 bit DAC controls the DC offset into the analog section. Its range is from -5 to +5 V which corresponds to an input referred range of +/- 50V (due to the front end attenuator with a gain of 0.1). The sensitivity of the DAC depends on gain settings as shown in Table 6 below.	
0x0C	DAC Control [7:0]	8			

Table 5. FPGA register map

Front End Register Setting	Voltage Resolution	Gain Setting	Attenuator Setting	DAC bits/ADC bit
0x60	100mV	9.375	1	2.048
0x41	200mV	5	0.9375	4.096
0x20	500mV	1.875	1	10.24
0x22	1V	1.875	0.5	20.48
0x03	2V	1	0.46875	40.96
0x04	5V	1	0.1875	102.4
0x05	10V	1	0.09375	204.8

Table 6. DAC counts to ADC counts conversion