Golf Mat Sensor Information

Communication

The sensor controller uses a dual-mode Bluetooth module that advertises both a Bluetooth Classic connection and a Bluetooth 4.2 Low Energy connection. The Bluetooth Classic connection can stream data at a much higher rate but requires the traditional pairing process and is not available on iOS without licensing from Apple. In the current firmware, none of the power-saving features of Bluetooth LE are implemented since the primary purpose is to stream the sensor readings in real-time.

The **Bluetooth classic** mode implements the **Serial Port Profile (SPP)**. SPP is a BT classic profile that is designed to make the wireless BT link behave as if it were a wired serial cable link, appearing as a COM port to most systems and transferring the data in ASCII format.

The **Bluetooth LE** mode makes use of a vendor specific “transparent UART” protocol from Microchip. Microchip’s **Transparent UART service UUID** is **49535343-FE7D-4AE5-8FA9-9FAFD205E455** and it is included in the BLE advertising packet.
The characteristic **UUID to read** the sensor data is **49535343-1E4D-4BD9-BA61-23C647249616** , and notifications can be enabled for this read UUID to receive the data as the sensor sends it.

Viewing Sensor Data

Mobile Devices

If you use Android, a quick way to view the data is to use a terminal emulator app that supports BT connections – such as:
https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal - my favorite because it supports Bluetooth LE (works without pairing to phone but slower) AND Bluetooth Classic
or
https://play.google.com/store/apps/details?id=com.timersnsavers.bluetoothusbwifiterminal&hl=en_US

on iOS,
https://itunes.apple.com/us/app/hm10-bluetooth-serial-lite/id1030454675 (not sure if this supports the Microchip vendor specific transparent UART UUID’s – iOS does not allow connection to Bluetooth Classic SPP without MFi licensing)

Windows Computer

Pair the sensor with the computer using the Bluetooth Classic SPP mode, perform the following steps:

1) turn on the sensor
2) go to the Bluetooth settings options menu and choose to add a new Bluetooth device
3) select the Bluetooth device called "**SsYoga-1000**" – make sure to select the device with the headphones icon (Windows) and the full name. The shortened name with the different icon is the Low Energy option.

This will pair the sensor to the computer such that on that computer moving forwards, the sensor will be recognized.

Bluetooth Terminal App:

Download “Bluetooth Serial Terminal” app from Microsoft Store
https://www.microsoft.com/store/productId/9WZDNCRDFST8

Open app, select device from dropdown menu, and click Connect. Make sure to select “New Line” from the “Transmit Line Ending” box on the right side, and leave the Transmit and Receive Format boxes as String.

Generic Terminal Emulator:

In Windows, the sensor will appear as a COM port after it is paired with the computer, so any terminal emulator program can be used to view the raw data streaming from the sensor. Some terminal emulator programs include PuTTY, Tera Term, and Realterm. The serial monitor for Arduino is also an easy tool if you happen to have the Arduino ide installed since it has a line to type a command string and a send button that will automatically add an NL at the end of the string.

Sensor Data Structure:

The data is transferred as ASCII characters where each frame of data (one scan of the sensor) is line separated (DOS style CRLF – carriage return /r, line feed /n). The data format is comma-separated JSON-ish objects, where the X and Y values denote the horizontal and vertical sensor location, and the Z value is the raw 12-bit ADC count where the higher the count, the higher the force. The X axis of the mat is the longer length dimension, and the Y axis is the narrower width dimension. The X=0, Y=0 corner of the mat contains the microcontroller and battery. These 3 demo golf sensor mats have X values ranging from X=0 to X=23, and Y values ranging from Y=0 to Y=11 (i.e. an X-Y matrix of 24 by 12). The standard production golf sensor mats will contain a denser sensor matrix than these 3 demo mats.

Example of Data Structure (without CoP):

[[{X,Y,Z,},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z}]]

[[{X,Y,Z,},{X,Y,Z},{X,Y,Z},{X,Y,Z}]]

[[{X,Y,Z,},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z}]]

For example,

[[{6,0,370},{7,0,621},{8,0,1090},{9,0,1095}]]

[[{7,0,488},{8,0,888},{9,0,1163}]]

[[{7,0,240},{8,0,602},{9,0,1127}]]

[[{8,0,319},{9,0,929}]]

Example of Data Structure (with CoP):

[[{X,Y,Z,},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z}]cop:{X,Y}]

[[{X,Y,Z,},{X,Y,Z},{X,Y,Z},{X,Y,Z}] cop:{X,Y}]

[[{X,Y,Z,},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z},{X,Y,Z}] cop:{X,Y}]

For example,

[[{6,0,370},{7,0,621},{8,0,1090},{9,0,1095}]cop:{7.916,0}]

[[{7,0,488},{8,0,888},{9,0,1163}] cop:{8.266,0}]

[[{7,0,240},{8,0,602},{9,0,1127}] cop:{8.45,0}]

[[{8,0,319},{9,0,929}] cop:{8.744,0}]

The Z threshold is currently set to 200, which means the sensor will only send data when it senses a force with a Z reading of 200 or higher – this prevents continuously sending unnecessary data. Therefore, to observe any streaming data, the mat must be stepped on.

The frame rate of the mat is currently set to 10 frames per second. The Z threshold, frame rate, and many other settings are stored in the mat controller's EEPROM memory that can be updated if desired.

If maximum data throughput is a concern, the sensor's data format can be changed to binary in the EEPROM settings. This transmits every single reading of the sensor matrix as two-byte binary data starting at x=0,y=0, then x=1,y=0, etc. increasing along the x axis first, then the y axis. Each frame of data is separated by a 4 byte frame separator 0xFF 0xFF. This can significantly increase the received data frame rate if you are not able to reach the desired frame rate with the standard data format settings. For more details on the binary data structure, please ask.

Adjusting Sensor Settings – Changing EEPROM settings:

Certain settings within the firmware can be adjusted with special commands. For the micro-controller to parse the command, it must be formatted as an ASCII character string followed by a new-line character "\n" (LF).

The micro-controller can be placed into "command mode" which will pause the sensor scanning and only listen for new commands. If a command is not recognized, nothing happens. If the command is recognized the command with the updated value will be echoed back over the serial port and the setting will be immediately applied to the firmware. The settings are stored to EEPROM when "command mode" is exited.

In order to change a setting and store it in persistent EEPROM memory, place the controller into "command mode", enter the command to change the variable(s) of interest, and then exit "command mode" to store the updated settings into the EEPROM.

To change a setting without saving it to EEPROM, do not enter command mode (the sensor will continue scanning) before entering the specific command. So long as the "exit" command is not issued, the changed settings will not be stored to EEPROM and will revert back to its previous value on the next power cycle.

The following is a short list of some of the available commands (all of these should be a character string terminated with a NL character):

"c" - places the micro controller into "command mode" which pauses the sensor scanning

"exit" - exits the "command mode" and stores all of the current values in EEPROM (stores any changes you've made) "evars" - short command to display the full list of EEPROM variables that can be changed as well as their current values

“t=###” – command to set the Z threshold that controls what readings are transmitted. This shows up as “*[3] – adcReadingThreshold = ###*”

“fps=##” – command to set the frame rate variable that controls the scanning rate of the sensor. This shows up as “*[4] – adc_average_count = ##*”

“z” – command to zero (or tare) the mat with a 3 sec average of the current readings. This does not show up in the list of EEPROM settings.

“zf” – command to zero (or tare) the mat with a single frame of data of the current readings. This does not show up in the list of EEPROM settings.

“y” or “apply_zero_matrix=1” – command to turn on zeroing (tare function) where a stored zero-value (tare-value) is subtracted from every matrix point’s raw ADC reading. This shows up as “*[27] – apply_zero_matrix=1”*

“x” or “apply_zero_matrix=0” – command to turn off zeroing (tare function) so the raw ADC reading is shown for every matrix point.. This shows up as “*[27] – apply_zero_matrix=0”*

“calcCoP=1” or “calcCoP=0” – command to enable or disable the calculation and addition to the data structure of the Center of Pressure reading for the sensor. This shows up as “*[40] – calcCoP = 1*” or “*[40] – calcCoP = 0*”

Example:
To change the Z threshold and frame rate, perform the following sequence (with LF termination):

"c" (to enter command mode)

"evars" (to show the current controller settings - note the value of [3] - adcReadingThreshold and [4] - adc_average_count)

"t=150" (to increase the Z threshold – ADC readings less than this value are ignored)

"fps=5 " (to change the frame rate to 5 frames per second)

"exit" (to exit command mode and store the changed setting)