freETarget - Interface Control Document

1. Executive Summary

This document describes the interface between the freETarget PC and Arduino.

1.1. Applicable Documents

Overview of JSON protocol: https://www.w3schools.com/js/js_json_objects.asp

2. Interface

The overall connection between the target Ardunino and display computer is illustrated in Figure 2-1.

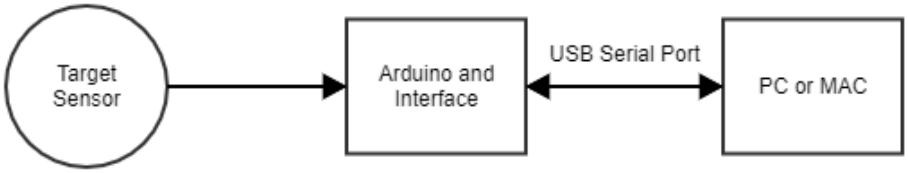

Figure 2-1: General Connection

The data is transferred between the Arduino and PC using a USB serial port. The baud rate shall be set to 115,200

All messages shall be transmitted as a JSON payload between the two computers. See the applicable documents section for an illustration

2.1. Arduino to PC

Target information, for example impact location is transmitted from the Arduino to the PC.

2.1.1. Valid Shot Information

Shots are recorded as

```
{
(required)
"shot": number,                          // Shot identifier
"miss": 0,                               // = =0 to indicate a valid shot
"name":  target name                     // Used to identify the target source
"time": shot time,                       // Shot recorded time (100 ms after reset)
"x": X position                          // X position in mm from centre
"y":Y position                           // Y position in mm from centre
"r": radius                              // Distance from centre in mm
"a": polar angle from centre             // Angle in degrees from 3 o'clock position

(optional)
"N": time                                // North timer value in counts
"E": time                                // East timer value in counts
"S": time                                // South timer value in counts
"W": time                                // West timer value in counts
"V_REF": voltage                         // Trip point value in Volts
"T": temperature                         // Air temperature in 'C
"VERSION": version                       // Firmware revision
}
```

Example
{"shot":1, "miss":0, "name":"TARGET", "time":97, "x":-0.29, "y":0.88, "r":0.92, "a":108.43, "N":0, "E":0, "S":0, "W":0, "V_REF":1.87, "T":20.50, "VERSION":"3.05.6 November 28, 2021" }

It is up to the display program in the PC to determine if the shot originates from a pistol or rifle, and hence the score associated with a given distance.

Optional items are provided for diagnostics purposes and not used by the PC. Additional optional items may be added if needed without modification by the PC program

2.1.2 Bad Trigger Response or Missed Shot

On occasion, one or two of the sensors will pick up the shot, but the remainder of the sensors will not. In this case, there is not enough information to compute a shot location. The Arduino will output a test message

```
{
"shot": number,                    // Shot identifier
"miss": 1,                         // = =1 to indicate a miss
"name":  target name               // Used to identify the target source
"time": shot time,                 // Shot recorded time (100 ms after reset)
"x": 0                             // Place holder
"y": 0                             // Place holder
}
```

Example

("shot":2, "miss":1, "name":"TARGET", "time":149, "x":0}

2.2. Setup and Diagnostics

The PC can send JSON messages to the Arduino for testing or configuration. Entering an unsupported command will result in the current command list being displayed

The messages are shown in Table 2.2-1

Table 2.2-1: PC to Arduino JSON commands

Command	Action	Use
Setup		
{“ANGLE”:angle in degrees}	Sets the sensor angle into the software	Needed for calculations, typically 45 degrees
{“CALIBRE_x10”:bullet calibre}	Sets the bullet calibration into the software	Future Use, typically 45 (for 4.5 pellet)
{“DIP”:value}	Set the DIP switch to a value and store in persistent storage.	Allows for remote configuration without the need to disassemble the target
{“LED_BRIGHT”:0-100}	Sets the LED brightness	Controls illumination 0=off, 100=on all the time
{“MFS”:0-9 0-9}	Sets the use of the multifunction switches	Determines the operation of the multifunction switches, Packed decimal number, ex 41
{“NAME_ID”:value}	Sets the name of the target	Differentiate targets in multi target setups
{“PAPER_ECO”:0/1}	Sets the value of the paper control	Set to 1 the paper does not advance if there is a shot outside of the black
{“PAPER_TIME”:time in 10 ms}	Sets the witness paper on time (10 ms)	Controls the amount of paper moved after a shot
{“POWER_SAVE”:time in minutes}	How long will the LEDs be on after the last shot	Used to power down the target after the last shot
{“SEND_MISS”:0/1}	Send shot information if there is a miss	Misses will be identified if SEND_MISS = 1

{“SENSOR”: sensor diameter in mm	Sets the distance between the sensor faces used in the impact calculations	Allows for the use of larger targets
{“STEP_COUNT”:value}	Number of steps issued to stepper motor	How many counts are used to advance the witness paper
{“STEP_TIME”:value in 10ms}	Duration of step pulse	How long the step pulse is applied
{“TABATA_CYCLES”:value}	Number of Tabata or rapid fire sycle	How many Tabata cycles before turning off
{“TABATA_ON”: time in 100ms}	Time LEDs is turned on	ON time in 100 ms increments
{“TABATA_REST”:time in seconds}	Time the LEDs are turned off	Used to control the rest time between shots
{“TARGET_TYPE”:value}	Overrides the location of shots	Used to change the target type or shot location, ex TARGET_TYPE = 1 -> 5 shot bull
{“Z_OFFSET”: distance in mm}	Distance from target to sensor plane	Used to adjust the calculations based on sensor location
In Development		
{“NORTH_X”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned North sensor
{“NORTH_Y”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned North sensor
{“EAST_X”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned East sensor
{“EAST_Y”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned East sensor
{“SOUTH_X”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned South sensor
{“SOUTH_Y”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned South sensor
{“WEST_X”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned West sensor
{“WEST_Y”: distance in mm}	Offset sensor position	Used to compensate for mis-aligned West sensor
{“DIP”:value}	Set the DIP switch to a value and store in persistent storage.	Allows for remote configuration without the need to disassemble the target
Diagnostics and Control		
{“BYE”:0}		Powers down unit
		{“GOOD_BYE”:0} Response from Arduino {“HELLO_WORLD”:0} If the Arduino wakes up on it’s own
{“ECHO”:value”}	Returns the value as a JSON string. Also returns the values of other settings in persistent storage	Used to verify the communications path, and display the current settings (see list above)
{“TEST”: value}	Starts a hardware self test	value = 0 Display installed tests
		value = 1 Digital I/O test.
		Value = 2 Counter test. Wait for external trigger from sensors
		Value = 3 Counter test Internally triggered
		Value = 4 Run the digital oscilloscope. Ends when five (5) serial characters are received
		Value = 5 Format the digital oscilloscope for display on the PC
		Value = 6 Advance the paper one position as programmed by {“PAPER”:value}
		Value = 7 Spiral Unit Test. Generates timer data to verify software operation
		Value = 8 Grid Unit Test. Generates a square grid to correlate timer values to shot position
		Value = 9 One Time Unit Test. Single shot to verify calculations
		Value = 10 Auxiliary port pass through
		Value = 11 Calibrate the trip point
		Value = 12 Cross unit transfer test (deprecated)
		Value = 13 Serial port test
		Value = 14 LED brightness test
		Value = 15 Face strike test
		Value = 16 WiFi test
		Value = 17 Dump Nonvol
		Value = 18 Send sample shot
{“TRACE”:0/1}	Enables and disables software trace	1 = enable trace

Information from Arduino, in response to ECHO command		
Configuration		Strings as above in setup
“IS_TRACE”:0/1	Shows current trace status	
“RUNNING_MINUTES”:time	Shows time since last reset in minutes	
“TEMPERATURE”:in C	Shows temperature read from sensor	
“SPEED_SOUND”:in mm/us	Shows the speed of sound based on current temperature	
“V_REF”:Volts	Sensor trip point,	Set by calibration
“TIMER_COUNT”:counts	Computed distance between furthest sensors in counts	Verify that the timers and sensors match for the target size
“DIP_HEX”: value	Value read from the configuration switch	
“WiFi”:0/1	ESP-01 status	1 = ESP-01 is present
“VERSION”:”version”	Current software version	
“BD_REV”:revision	Current board revision	Used to modify the hardware operation based on installed hardware

2.3 Power On and Connection

When the PC attaches to the Arduino, the PC expects to see a set string to indicate that the Target is connected and it’s configuration.

The version is used to identify that hardware and is displayed in the corner of the program

The configuration is copied into the setup page of the PC program and is used to change the operation of the device if needed.

(Version String)

freETarget "3.05.6 November 28, 2021"

(Configuration)


```
{
"NAME":"TARGET",
"ANGLE": 45,
"CALIBREx10": 45,
"DIP": 0,
"LED_BRIGHT": 50,
"MFS": 0,
"NAME_ID": 0,
"PAPER_ECO": 0,
"PAPER_TIME": 0,
"POWER_SAVE": 30,
"SEND_MISS": 0,
"SENSOR": 230.00,
"SN": 101,
"STEP_COUNT": 0,
"STEP_TIME": 0,
"TABATA_CYCLES": 0,
"TABATA_REST": 0,
"TABATA_ON": 0,
"TARGET_TYPE": 0,
"Z_OFFSET": 0,
"NORTH_X": 0,
"NORTH_Y": 0,
"EAST_X": 0,
"EAST_Y": 0,
"SOUTH_X": 0,
"SOUTH_Y": 0,
"WEST_X": 0,
"WEST_Y": 0,
"IS_TRACE": 0,
"RUNNING_MINUTES": 0,
```

"TEMPERATURE": 24.00,
"SPEED_SOUND": 0.35,
"V_REF": 1.86,
"TIMER_COUNT":5568,
"DIP_HEX": 0x00,
"WiFi": 1,
"VERSION": "3.05.6 November 28, 2021",
"BD_REV": 3.10
}

The Arduino may be powered down by sending the command {“BYE”:0} which is acknowledged by {“GOOD_BYE”:0}. If the Arduino comes out of the power down state on it’s own, ex user presses a multifunction switch, it will send out {“HELLO_WORLD”:0}

2.4. Sensor Connector

The sensors are connected to the main board using a 12 pin IDC (0.100” spacing) connector. The pinning of the connector is illustrated in Figure 2-3. Each sensor, North, East, South, West is carried over a separate conductor, and the sensor module is responsible for managing the cable.

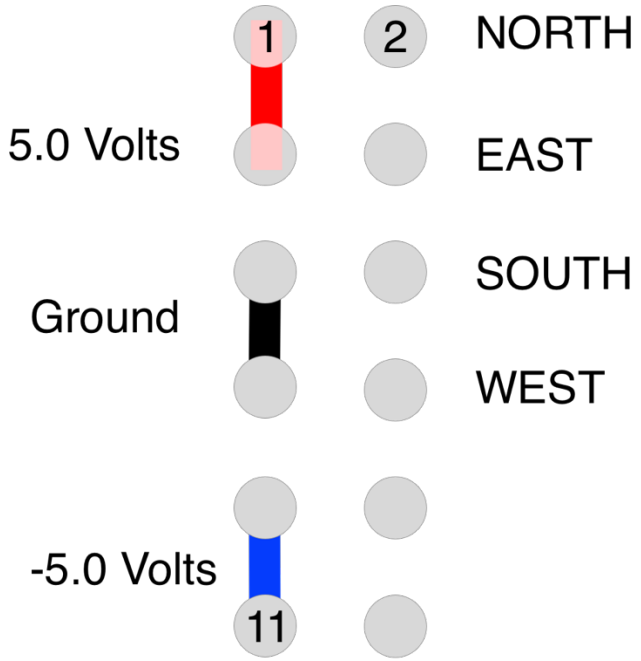

Figure 2-3: Sensor Connector. Note Pins 11 and 12 are not provided in Version 3.1 hardwar

2.5. Sensor Location

The sensors are located around the edge of the target at a distance of 150mm from each other. The location of the sensors is shown in Figure 2-4.

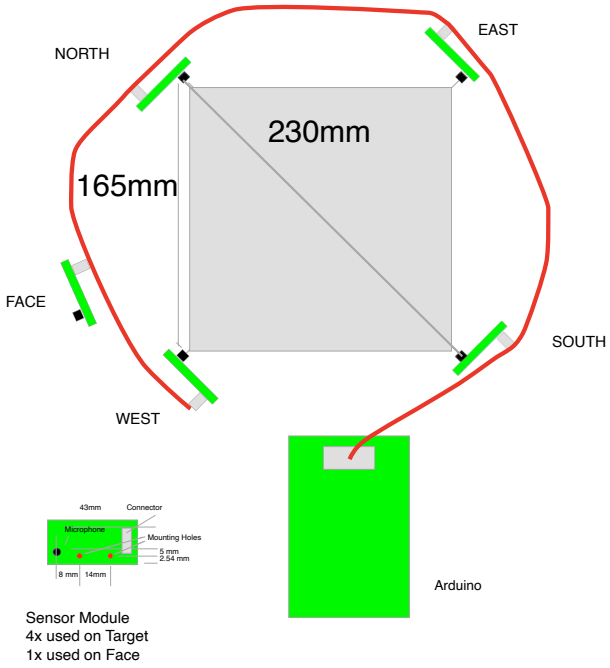

Figure 2-4: Sensor Mechanical Assembly

The critical dimensions to observe are:

- Edge-to-edge distance of the sensors (230 mm)
  - Errors in this dimension will shift the centre location
- Angular alignment of the sensors
  - Errors in this dimension will rotate the shot group.