

FlyWire

FireFly Communication Protocol for the Macro Scale



This document describes the commands available for the FireFly to communicate with a Macro Scale and how to use the additional factory, test and custom commands. See the FlyWire communication protocol document for additional information.

The following topics will be addressed in this document:

1. Principle of operation
2. Factory/test commands
3. Custom commands
4. EEPROM map

1. Principle of operation

The Quantified Macro Scale is weighing sensor designed to be used in conjunction with a Quantified FireFly. The Macro scale contains one S-type load cell in a rugged enclosure. Typical applications of the Macro Scale include tracking of biomass growth, or tracking soil moisture content as part of a lysimeter.

Beside the load cell, the Macro scale contains a microcontroller running the FireFly stack and a dedicated 24 bit ADC (Analog to Digital Converter) with integrated PGA (Programmable Gain Amplifier).

On the Macro Scale all standard and optional commands are implemented, except for Set Values. See the FlyWire protocol document for more details. These command suffice for regular usage in combination with the FireFly. In order to be able to calibrate the device for usage with different load cells, several factory commands and custom commands are implemented.

The microcontroller on the Macro Scale converts the raw ADC value to a weight in grams. This value can be obtained by sending a Request Sensor Values (0x02) command. The result is a 32 bit signed long in 2's complement format.

The Macro Scale will tare when a magnet is used to reset the FireFly. The additional tare value is volatile, so after a power-down or a Reboot command (0x03) this value is set to zero.

2. Factory/test commands

The following test command has been implemented:

Stay Awake:

0x5A | 0x04 | 0xE0 | CHK

This command will override the 1 second auto sleep feature of the Macro Scale. After issuing this command the Macro Scale will stay awake until a power cycle occurs, a Reboot or a Hibernate command. This command is useful during testing without having to send a break each time to wake-up the device.

The Macro Scale will reply with an ACK.

STF | LOF | NAK | CHK

In **Error! Reference source not found.** the Stay Awake command has been issued to prevent the Macro Scale from going to sleep. This allows the user to press the Get Raw Value button without having to send a break condition to wake up the device after a 1 second delay.

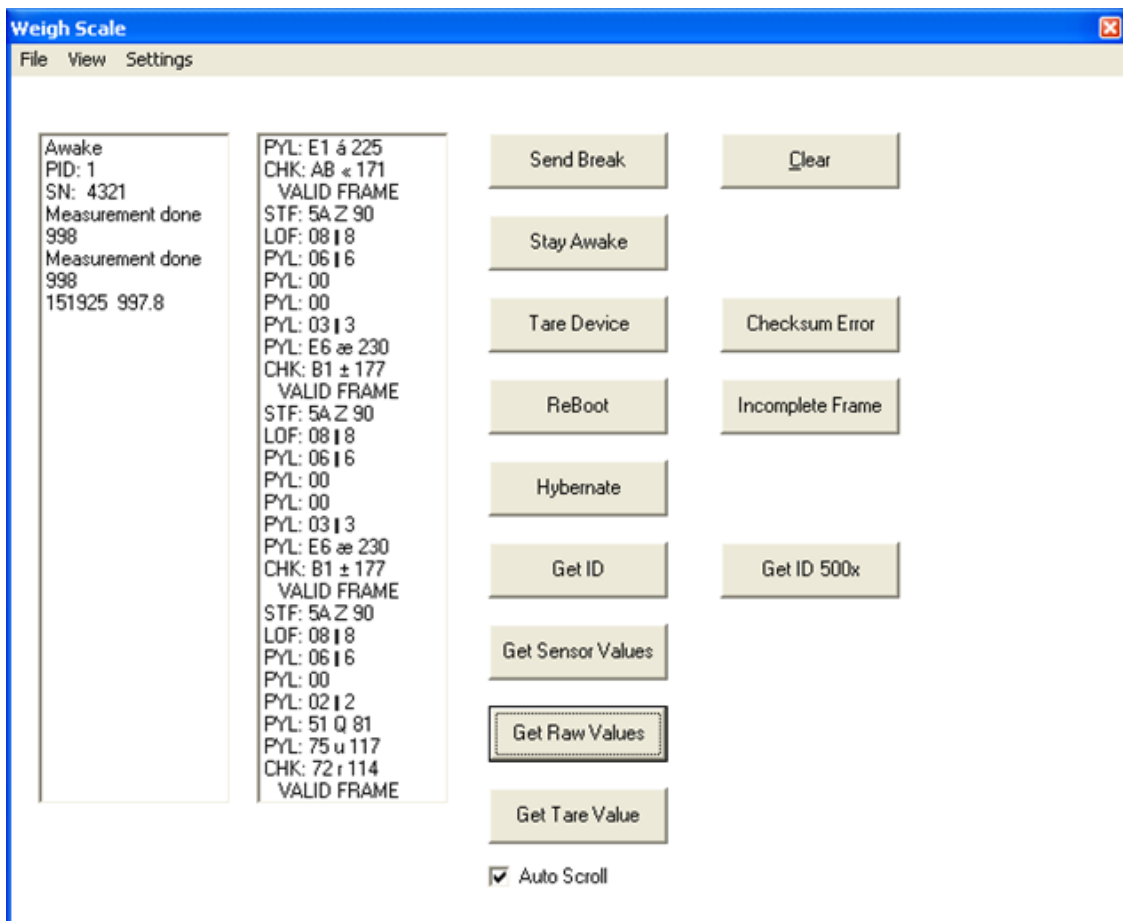


Figure 1: Macro Scale response to Stay Awake and Get Sensor Values commands.

3. Custom commands

Compliant with the FlyWire protocol one command has been implemented to read the raw ADC value. Internally the ADC is oversampled 16 times and averaged.

Get Raw Value:

0x5A | 0x04 | 0xF1 | CHK

This command will start the 16 times oversampled ADC conversions and returns the averaged value in 32 bit 2's complement format.

The Slave will respond with:

STF | 0x08 | 0x06 | MMSB | MSB | LSB | LLSB | CHK

MMSB is the most significant byte and LLSB is the least significant byte. As an example a negative value of 0xFF8A2365 will be returned as follows:

STF | 0x08 | 0x06 | 0xFF | 0x8A | 0x23 | 0x65 | CHK

4. EEPROM map

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x10	4D	65	67	61	53	63	61	6C	65	50	72	6F	74	6F	0	0
0x20	1	0	2	8	7	14	14	8	7	14	14	FF	FF	FF	FF	FF
0x30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x40	FF	FF	FF	FF	FF	FF	FF	FF	0	FF	FF	FF	FF	FF	FF	FF
0x50	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x60	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x70	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0x80	43	B	81	48	0	0	2E	E5	1	FF	FF	FF	FF	FF	FF	FF
0x90	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0xA0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0xB0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0xC0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0xD0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0xE0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0xF0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

EEPROM	Change	Details	Format
<input type="button" value="Read EEPROM"/> <input type="button" value="Write EEPROM"/> <input type="button" value="Erase EEPROM"/>	<input type="button" value="Name"/> <input type="button" value="Product ID"/> <input type="button" value="Serial Number"/> <input type="button" value="Production Date"/> <input type="button" value="Calibration Date"/>	Name MegaScaleProb Product ID 1 Serial Number 2 Production 8-7-2020 Calibration 8-7-2020	<input checked="" type="radio"/> Hexadecimal <input type="radio"/> Decimal <input type="radio"/> ASCII
			<input type="button" value="Load from Disk"/> <input type="button" value="Save to Disk"/> <input type="button" value="Close"/>

Figure 2: Example EEPROM memory snapshot.

0x00 – 0x0F Reserved

0x10 – 0x1F These locations hold the name of the device in ASCII. Unused locations at the end should be padded with 0x00. In the example above the EEPROM holds the name Macro Scale.

0x20 – 0x22 Product ID and Serial Number. The SN consists of two bytes forming an unsigned integer. High byte at 0x21, low byte at 0x22. In the example the Product ID is 1 for a Macro Scale (see list of supported devices in the FlyWire protocol document). The Serial Number is 0x0002 (2 in decimal)

- 0x23 – 0x26 Production date in DD/MM/YY/YY format. The year is split in two bytes. E.g. 2020 is stored as 20, 20 or 0x14, 0x14 in hex format. In the example both production date and calibration date are set to July 8th, 2020.
- 0x27 – 0x2A Calibration date in DD/MM/YY/YY format. In the example the calibration date is set to July 8th, 2020.
- 0x80 – 0xFF This area is reserved for calibration data.
- 0x88 Location 0x88 is used for debugging. If its value is non-zero, the on-board LED will be lit as long as the Solar Chimney is active, and switch off in hibernation. When set to zero, the LED will always be off to preserve power.