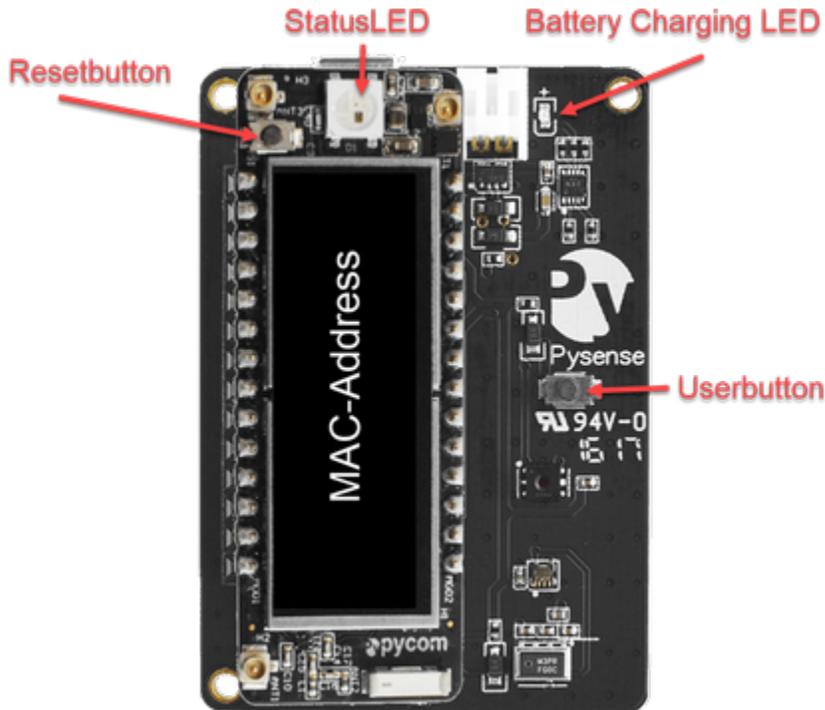


Field Tester Documentation

- Overview
- Event-Cycles
- Downlink configuration
- LED Behaviour
- Decoder

Overview



StatusLED : The status LED shows the various states of the field tester and allows user interaction. The details of this are documented in a table below

User button : User button allows the user to wake up the device from deep sleep and perform certain user actions like (factory reset, rejoin, deepsleep) as described further in the document

Reset button : Reset button allows power cycling of the field tester

Battery Charging LED : Indicated charging of battery (red color)

Event-Cycles

1. The device will boot and enter the user mode
2. Then depending on the user action, the device can rejoin, go to deep-sleep or get factory reset (described in detail below)
3. If no action is taken by the user then the normal up-link procedure is followed and the LED will flash rapidly with BLUE colour
4. The device will now go into deep-sleep according to the given delay period
5. To wake the device up from deep-sleep mode the USER button needs to be pressed.

Downlink configuration

The device can be set via downlink. Use default port 1 to send downlink.

Time Interval	GPS	CDU/UDU	LED
Delay time interval (in Seconds)	GPS ON 1 / OFF 0 (Only Pytrack)	CDU 1 / UDU 0	LED ON 1 / OFF 0
First 7 bits specify the time delay or the time interval between the data up-links. This is defined in seconds Eg. 1 minute will be 0000060	The second parameter sets the GPS value (on Pytrack Only) .. '1' will set the GPS ON and '0' will set GPS OFF	The third parameter is for Confirmed or UN-confirmed packet up. '1' will setup a CDU and '0' will setup a 'UDU'	The fourth parameter is for the LED behaviour. In order to save battery after the Device has been setup in the Field the LED behaviour can be turned OFF or ON by modifying this bit. '1' will set the LED's to be ON and '0' will set the LED's to be OFF

(There is an internal Delay of ~18 seconds so the minimum time period of delay should be greater than 19 Seconds - or by default it will be set to 19 Seconds (There is a slight variation for boards)

Example:

Time Interval	GPS	CDU/UDU	LED
0000060	0	0	0
60 Second time interval	GPS off	UDU	LED off

Put the downlink on port 1 with code: 00006000

LED Behaviour

When THE LED blinks in **Green** Colour the device goes in user mode. While the LED is blinking in green you have to press the Userbutton. The green LED will finish the blinking cycle and then enters in the

1. Deep Sleep activation is possible by pressing the USER_BUTTON for 5 of 9 blinks of the **BLUE LED**. To bring the device back from deep sleep USER BUTTON has to be pressed again
2. Rejoining the network is possible by pressing USER BUTTON for 3 of 9 blinks of the **BLUE LED**.
3. Factory Reset to default settings is possible by pressing the USER_BUTTON for 11 blinks of **BLUE LED**.

The LED behaviour acts and the user interface to control / understand the device when it is detached from the console.

Below are listed few LED functions :

Colour	Details
GREEN	When the device is booted and its waiting for the user_button action as mentioned above [1]
VIOLET	When the LED blinks violet 3 times the device will enter deep sleep mode
BLUE	When the Blue LED Blinks it acts as an acknowledgement of User Button being pressed. Blue LED also blinks fast when the device sends an up-link
SKY-BLUE	Sky blue light blinks when the Device is trying to join a network
YELLOW	When the device joins a network a bright yellow light will blink twice
RED	On error with joining or sending RED LED will Flash fast couple of times to indicate an error. Factory reset will also flash the RED LED 3 times.

Decoder

FieldTester Decoder

```
// New Pycom FieldTester Decoder
function fieldtesterDecoder(bytes) {
    var obj = {};
    var hexStr = "";
    for (i = 0; i < bytes.length; i++)
    {
        hexStr = hexStr.concat( ("0" +
(Number(bytes[i]).toString(16))).slice(-2).toUpperCase())
    }
    var k = parseInt("0x"+hexStr.substring(0,2));
    var l = parseInt("0x"+hexStr.substring(2,4));
    var m = parseInt("0x"+hexStr.substring(4,6));
    var n = parseInt("0x"+hexStr.substring(6,8));

    var o = ( (k>>2) + (l<<6) ) << 2 ;
    var p = ( (m>>2) + (n<<6) ) << 2 ;

    obj.BatteryVolt= o/100 ;
    obj.LastRSSI= p ;
    return obj;
}

// SmartMakers
function decode(byte_payload) {
    return fieldtesterDecoder(byte_payload)
}
```