



National Coding Week Activity

How to create a working LoRa Tracker board using code:



Purchasing the equipment:

Heltec ESP32 LoRa Meshtastic Asset Tracker SX1262 Wi-Fi Bluetooth GPS Tracker for vehicle bike kids pet car fitness tracker – *this should take 7 working days to arrive.*

Link:

https://www.aliexpress.com/item/1005005681504175.html?spm=a2g0n.order_detail.order_detail_item.3.4ddff19cHu4NiX

Please note: Make sure you choose the 433MHz version, you can do this by selecting the correct image when placing the order – see below:

Color: 433MHz



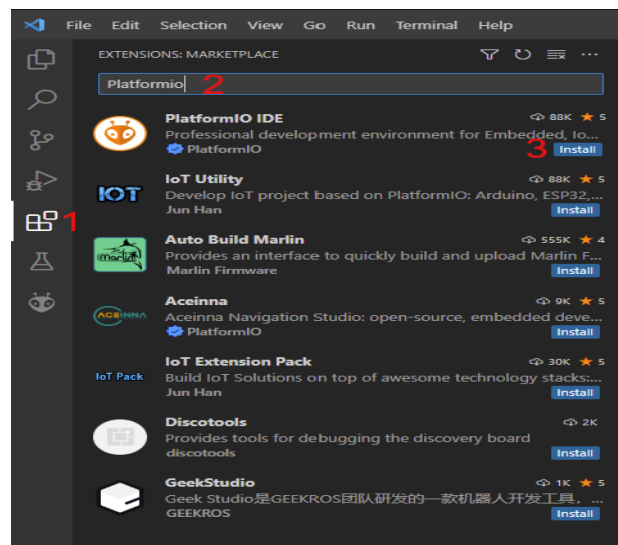
You will also need a USB to USB-C lead



The activity:

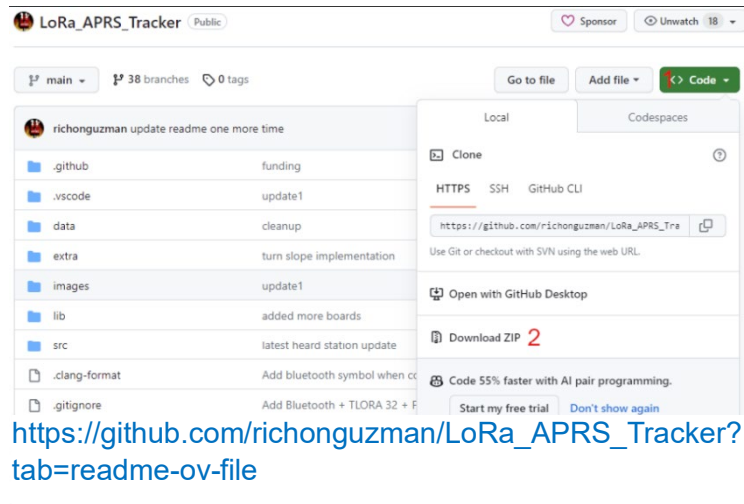
Step 1.1: Download VSCODE: <https://code.visualstudio.com/download> and install it on your computer.

Step 1.2: Open VSCODE and look at the left vertical menu for Extensions. There look for “Platformio” and install it.

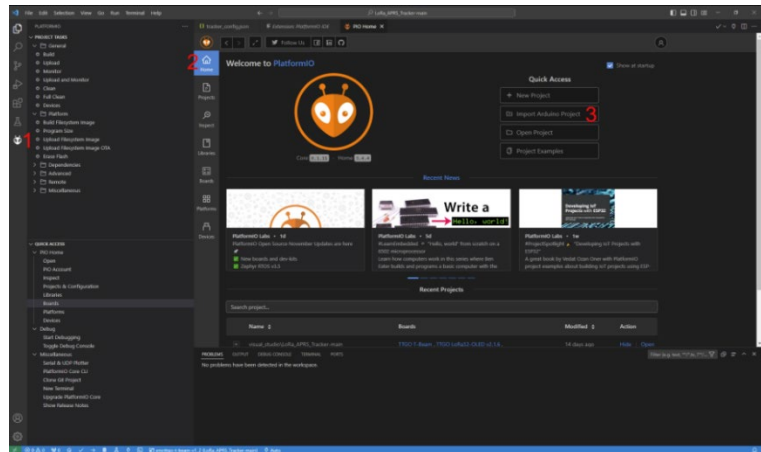




Step 1.3: Download the Tracker firmware zip file to your computer from this link and repository to a folder that will be the working project.



Step 1.4: Unzip and import as a new project into Platformio. (hint: on the bottom horizontal line of VSCODE is a house icon/symbol. Press it and Platformio home/page will open inside VSCODE). Press "Import Arduino Project" and a new menu will open.

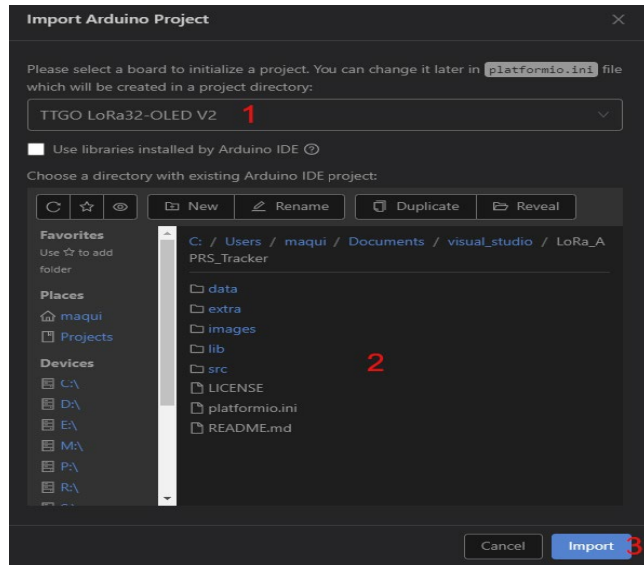




Select your board version: TTGO
LoRa32-OLED V1 / TTGO LoRa32-
OLED V2 / TTGO LoRa32-OLED
V2.1.6

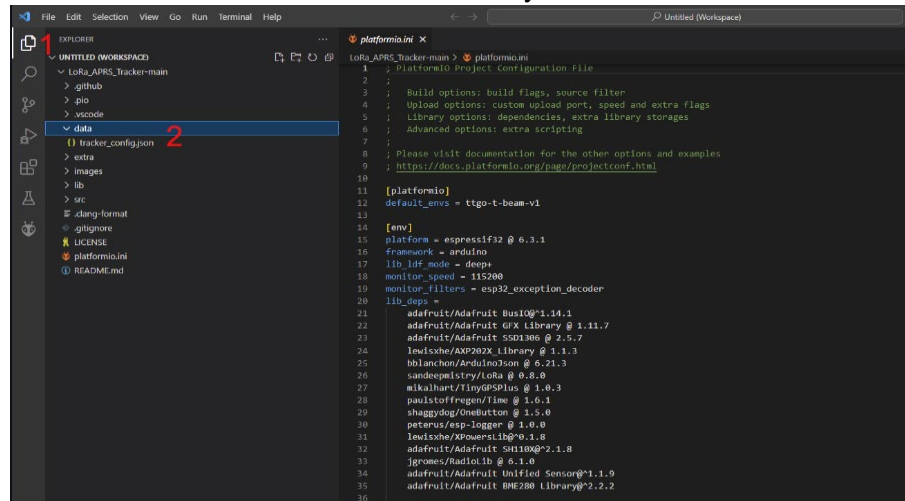
Select the folder where you have put
the unzipped firmware file you
downloaded.

Press "Import" button.



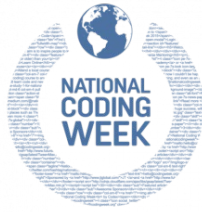
Tracker configuration process:

In Platformio (inside VSCODE), there should be a folder with the firmware you imported. Inside this folder you should find the "data" folder. Press it and you will find "Tracker_conf.json" file. Now it's time to edit this to configure your Tracker:



callsign

Change all callsigns from "NOCALL-7" to your Callsign+SSID. You could use three different Callsigns+SSID or the same callsigns with different SSIDs or even the same Callsign+SSID and only change symbols and smart beacon data. Once it's operational, with the buttons you change from one profile to another, so that it controls the beacon with different parameters depending on use, speeds....



symbol and overlay

Change them to your needs. Usually only changing symbol is needed (for the special event of wanting to use “\” as overlay you should use “\\” as this is the way the firmware would process it).

Mic-E

Add it if you want to send Mic-E GPS encoded aprs packets. By default, is empty "", but if you add the 3 Mic-E Message Type identification it will send in Mic-E encoding. For example, Mic-E encodings are "111" Off Duty, "110" en Route. For more information: <http://www.aprs.org/doc/APRS101.PDF> (page 45).

comment

Text that will be sent each time you send a beacon. If this isn't necessary, please leave it empty ("") as this will shorten the LoRa packet and increase success of being listened and uses less time in RF.

```
1 {
2   "beacons": [
3     {
4       "callsign": "NOCALL-7",
5       "symbol": "[",
6       "overlay": "/",
7       "comment": "",
8       "smart_beacon": {
9         "active": true,
10        "slowRate": 120,
11        "slowSpeed": 3,
12        "fastRate": 60,
13        "fastSpeed": 15,
14        "minTxDist": 50,
15        "minDeltaBeacon": 20,
16        "turnMinDeg": 12,
17        "turnSlope": 60
18      }
19    },
20    {
21      "callsign": "NOCALL-7",
22      "symbol": ">",
23      "overlay": "/",
24      "comment": "",
25      "smart_beacon": {
26        "active": true,
27        "slowRate": 120,
28        "slowSpeed": 10,
29        "fastRate": 60,
30        "fastSpeed": 70,
31        "minTxDist": 100,
32        "minDeltaBeacon": 12,
33        "turnMinDeg": 10,
34        "turnSlope": 80
35      }
36    }
37  ]
38 }
```

smart_beacon

Here you adjust the values for better smartBeacon results in your Tracker:

- “active”: true, means it calculates the time between beacons as a product of speed, course, time, distance and angle of your movement. When “false” is selected “nonSmartBeaconRate” will be used as a time between beacons
- “slowRate”: number of seconds as the longest time between beacons
- “slowSpeed”: speed in km/h at which the “slowRate” is used. Slower speed won't send longer time between beacons as “slowRate”
- “fastRate”: number of seconds as the fastest time between beacons
- “fastSpeed”: speed in km/h at which the “fastRate” is used. Faster speed won't send shorter time between beacons as “fastRate”
- “minDistTx”: distance in meters that the Tracker should move before sending another beacon. If this isn't achieved in certain time (“slowRate”) the Tracker will send only one beacon at 15min (“standingUpdateTime”)



- “minDeltaBeacon”: number of seconds between each beacon. No faster than this time will be sending packets even at high speed or changing many times from course angle
- “turnMinDeg” + “turnSlope”: are being used together as the angle and the change in angle corresponded to speed to calculate if the course has been changed enough to send a beacon (Excel table on “extra” folder is there to show how is being used)

LoRa

“frequency”, “spreadingFactor”, “signalBandwidth”, “codingRate4” and “power” should be left as they are unless your country needs to change this to use it according to their RF rules.

Other

Here are more custom configuration options:

- “simplifiedTrackerMode”: true was created to users who just want to send GPS packets and not use the menu of the Tracker. Default is false
- “showSymbolOnScreen”: true enables seeing the symbol next to the Callsign+SSID on the Oled Screen. Default is true. If you only see a letter of the symbol selected and not a drawn symbol, this is because it has not yet been drawn in the software
- “sendCommentAfterXBeacons”: the Tracker can send comment at each beacon but with this setting you can choose to send the comment text only after a number of beacons. Default is 10. This means that only after 10 beacons the “comment” text will be transmitted also
- “displayEcoMode”: false will let the screen stay always on (although you can choose to turn it off with displayEcoMode in the Tracker Menu later. true will force the Tracker to have the screen turned off right at the bootup to avoid being activated each time with the menu. Default is false
- “displayTimeOut”: seconds that the Tracker will wait to turn the screen off when “displayEcoMode” is activated (by configuration or by Menu). Default is 4
- “path”: WIDE1-1 is being used as the default path for sending beacons and messages
- “nonSmartBeaconState”: when “smart beacon” is not active this will be the time in minutes at which each beacon will be sent at a fixed rate. Default is 15
- “rememberStationTime”: time in minutes that the Tracker remembers a station being heard to show it on the Listening Menu of other Trackers around before being dropped from the list. Default is 30
- “maxDistanceToTracker”: distance in kilometres for a future function not implemented (leave this as is please)
- “standingUpdateTime”: time in minutes that the Tracker waits to send a beacon if the Tracker has not move outside the “minDistTx” to force a beacon transmission. Default is 15



- “sendAltitude”: if true the Tracker sends altitude+course in the encoded GPS data, if false the Tracker will send velocity+course. Remember that APRS calculates speed automatically even when sending altitude+course. Default is true
- “sendBatteryInfo”: true adds the battery voltage to the comment text in the beacon. Default is false
- “bluetoothType”: 1 selects the type of connection for using the Tracker as TNC. 1 is for Android and Aprsdroid. 0 is for iPhone and APRS.fi app
- “bluetoothActive”: true enables the Tracker to connect to BT devices like Phones and Apps over TNC2 or KISS TNC. Default is true
- “disableGPS”: true turns off the GPS of the board if the Tracker is used with the GPS of the phone over Bluetooth or when using a LoRa32 board which hasn't GPS module attached. Default is false

PTT trigger

(used for external power amps to boost the power in Tx)

- “active”: true enables the following configurations. Default is false
- “io_pin”: pin number at which the trigger is connected. Default is 4
- “preDelay”: number of milliseconds to wait before activating “io_pin” and transmitting. Default is 0
- “postDelay”: number of milliseconds to wait after transmitting to turn off “io_pin”. Default is 0
- “reverse”: some triggers need “io_pin” to start HIGH of LOW. Adjust to your needs

BME (Wx data)

- “active”: true enables the Wx data (Temperature, Humidity and Pressure) to be shown on the Oled Screen. Default is false
- “sendTelemetry”: true enables Wx data to be transmitted to APRS-IS servers and appears as telemetry. Default is false. To Send Telemetry you need both “active”:true and “sendTelemetry”:true. Please be aware that this forces 10min between Wx data Tx (without affecting the usual GPS data beacon Tx) and also uses a momentary different Symbol for APRS-IS servers to process the beacon packet as Wx data
- “heightCorrection”: 0 number of meters you want to change the reference of height for certain places when measuring altitude. Default is 0

Notification (led + buzzer)

(This is by adding YL44 buzzer Module and leds to the io_pins of the Tracker to add visual and sound notifications)

- “ledTx”: true flashes the Tx-Led when a LoRa packet is being transmitted. Default is false
- “ledTxPin”: io_pin where the Tx-Led (+) pin should be connected. Default is io_13. Use 470 ohms resistor in series with the led when using difuminaste Led or 10K when using ultra bright led



- "ledMessage": true flashes Message-Led when a message is received. Default is false
- "ledMessagePin": io_pin where the Message-Led (+) pin should be connected. Default is io_2. Use 470 ohms resistor in series with the led when using difuminaste led or 10K when using ultra bright led
- "buzzerActive": true is to activate sound notifications using YL44 module. Default is false
- "buzzerPinTone": io_pin where the I/O pin of the YL44 module should be connected. Default is io_33
- "buzzerPinVcc": io_pin where the VCC pin of the YL44 module should be connected. Default is io_25
- "bootUpBeep": true enables buzzer notification at BootUp of the Tracker. Default is false
- "txBeep": true enables buzzer notification at LoRa packet Transmission. Default is false
- "messageRxBeep": true enables buzzer notification at LoRa Message received. Default is false
- "stationBeep": true enables buzzer notification when another LoRa Station GPS packet is received (Tracker or iGATE/Digi). Default is false
- "lowBatteryBeep": true enables buzzer notification when Battery is lower than 20%. Default is false. When lower than 5% it will buzz twice

Nearly the final step – uploading firmware and filesystem to your Tracker board:

Before you upload, make sure the correct board has been defined.

Click at the bottom of the screen – see below highlighted area:

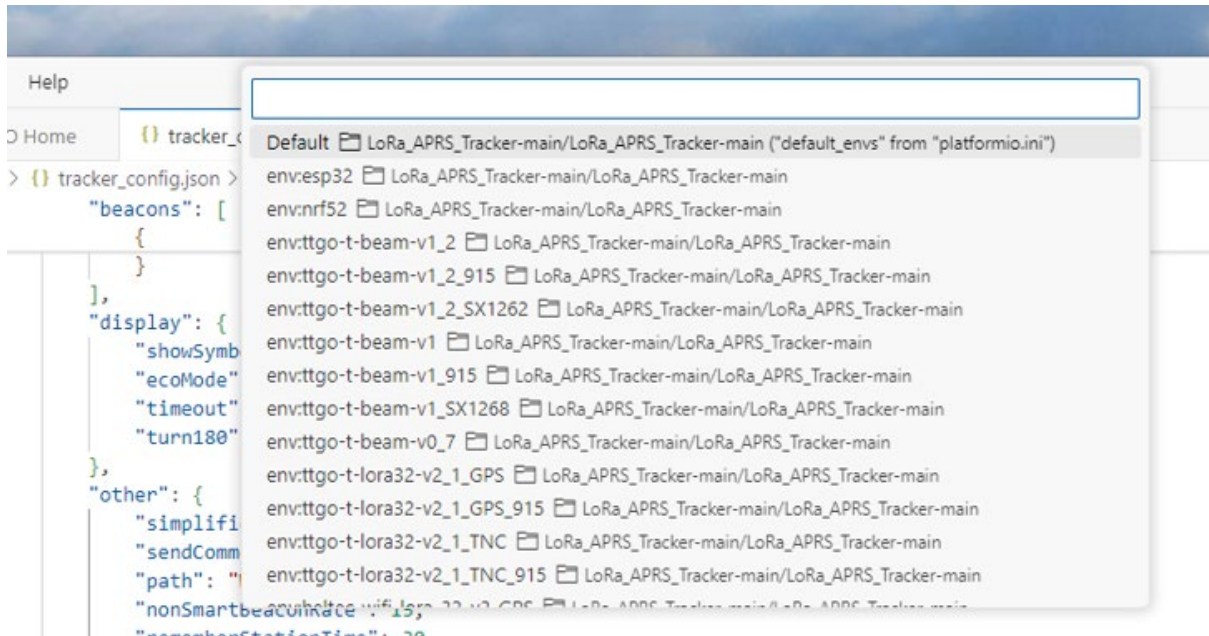
```
-----
heltec_wireless_tracker SUCCESS 00:00:38.267
===== 1 succeeded
* Terminal will be reused by tasks, press any key to close it.
|
```

env:heltec_wireless_tracker (LoRa_APRS_Tracker-main) Auto

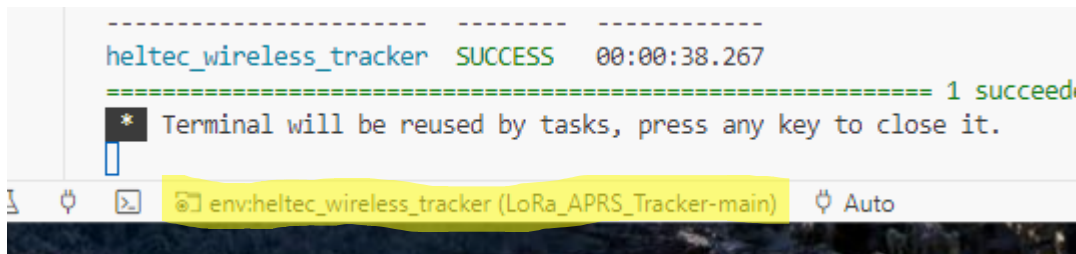
```
-----
heltec_wireless_tracker SUCCESS 00:00:38.267
===== 1 succeeded
* Terminal will be reused by tasks, press any key to close it.
|
```

env:heltec_wireless_tracker (LoRa_APRS_Tracker-main) Auto

This will open a window at the top of the screen – please enter:
env:heltec_wireless_Tracker into this box.



Choose the correct drop down, and you should see the bottom of the screen change to the below, this is correct.



After all the editing of "Tracker_conf.json" file now it's time to upload all:

I recommend uploading the firmware first, and then filesystem.

Upload firmware

On the bottom of the VSCODE screen you should find a "swoosh or tick" symbol/icon. Press it to validate all was imported and edited correctly.

If all is ok, connect the LoRa Tracker board with the (data) USB cable to your PC. You will need to purchase one of these if you don't have one (USB to USB-C). Find the "horizontal arrow pointing to the right" symbol/icon on the right of the tick symbol and press it. This will upload the firmware code to the Tracker board (you are halfway now).

It may take a few minutes, even on a reasonably fast PC.

Now the upload of firmware is successful, please make sure you press the following on your LoRa Tracker board to continue:



HOLD DOWN THE 'USER' SMALL BUTTON AND THEN PRESS THE 'RST' SMALL BUTTON ONCE – LOOSE GO OF THE 'USER' BUTTON NOW.

Doing this enables you to upload the filesystem.

Upload filesystem

The Tracker screen is not ready yet, so this part is also necessary.

On the left of VSCODE screen are a few symbol/icons. Find the Platformio icon (head of an ant) and press it. A new menu would appear.

Look for Project Task, this is a drop-down bar. In this dropdown, look for Platform and there you should find Upload Filesystem Image. Press it and it should upload the json file with all the data you changed in the Tracker Configuration part.

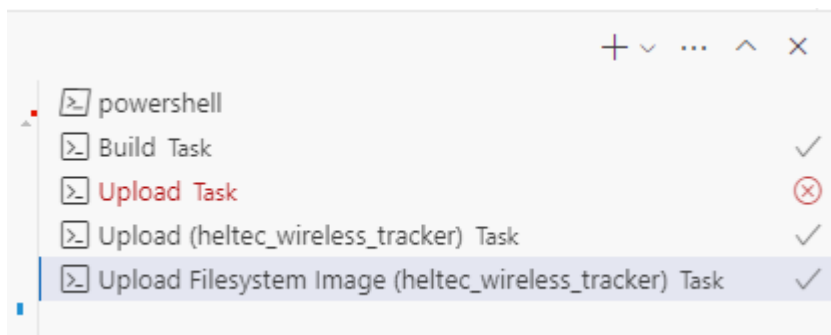
NOTE: Sometimes VSCODE has a bug here. If it doesn't work with the upload, close VSCODE, open VSCODE again and try this last part again (I had to do this).

If all went well, you should see the welcome message on the Tracker screen and then the Tracker menu with your callsign and all...

 Congrats!

⇒ Just a small note:

The software keeps track of what you have done at the bottom RHS of the screen.



You can see I failed at one spot.

I had the wrong board loaded.