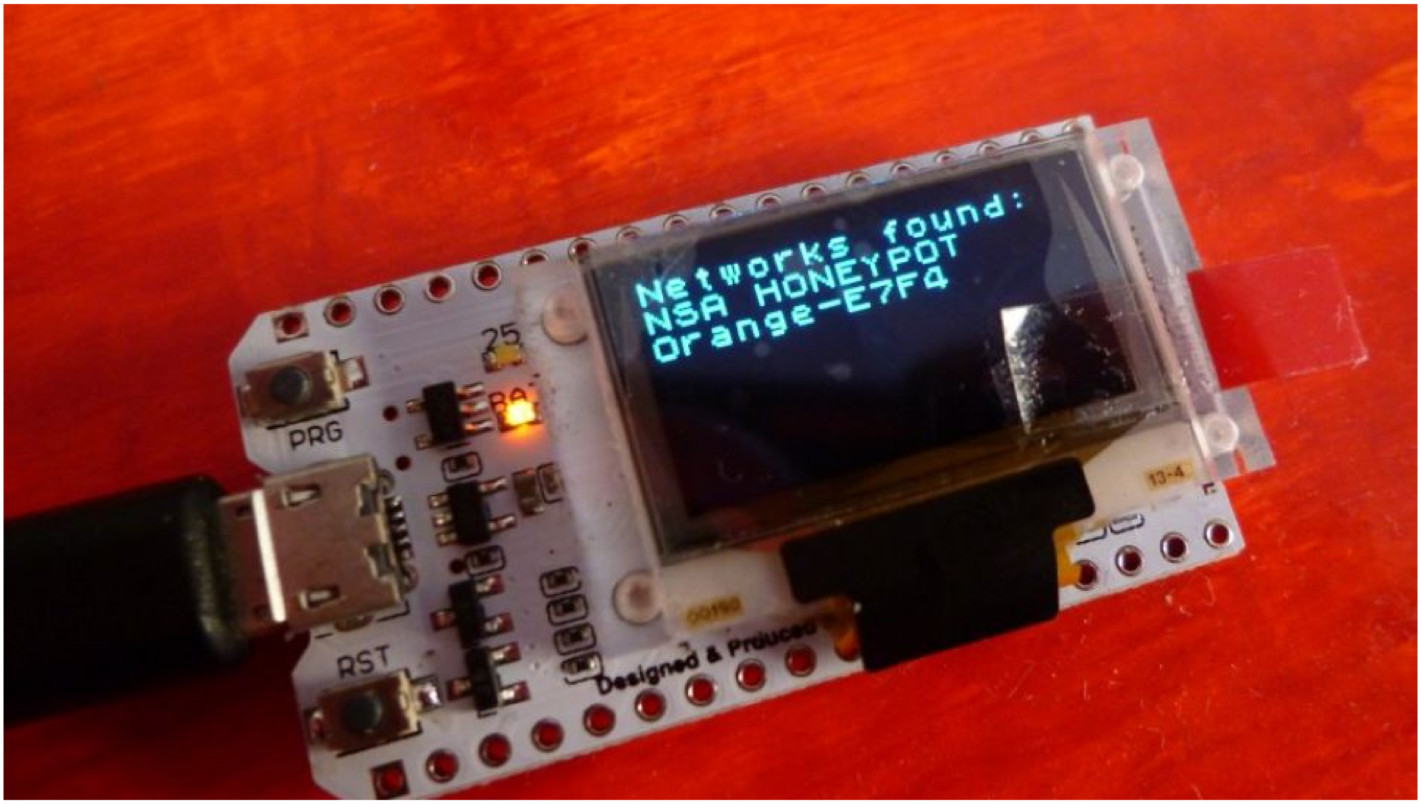
ESP32

ESP32 Built-in OLED – Heltec WiFi Kit 32

September 22, 2017 · Updated: March 27, 2019 17:17:59 · WordBot 76578 30



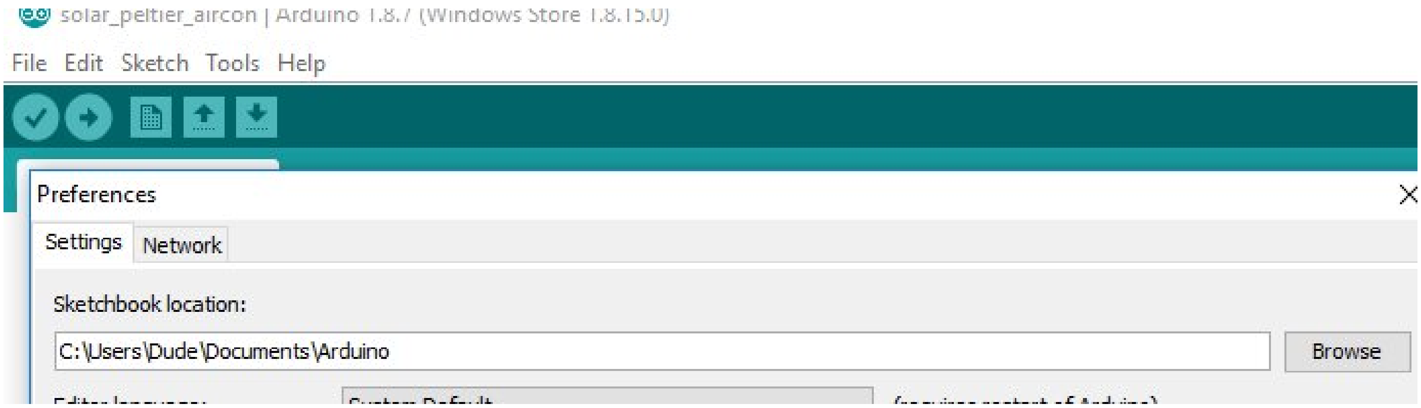
An introduction and set-up for the Heltec Automation WiFi Kit 32 development board with OLED display. Follow the steps below to have the example WiFiScan script show your local access points on the built-in display.

This board is based on the ESP32 chip and has onboard WiFi, Bluetooth, a 0.96 OLED display, lithium battery connector charging and a CP2102 USB to serial interface. It also works with the Arduino IDE. They are available from the Heltec Store on Aliexpress.

Setting Up the Arduino IDE for the ESP32 Range

New Easy Method

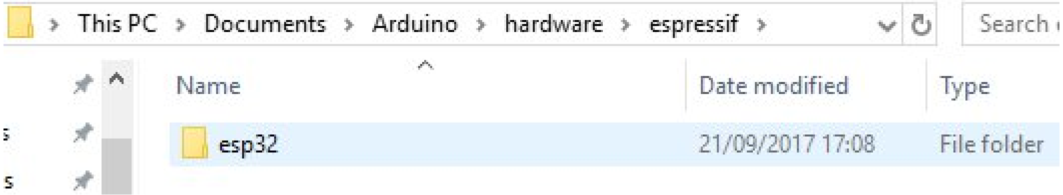
If you previously installed the hardware libraries for the ESP32 using the old method you need to delete them. Find the folder where your Arduino libraries are kept by opening File > Preferences in the Arduino IDE:


solar_peltier_aircon | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help
Preferences
Settings
Network
Browse

Sketchbook location: C:\Users\Dude\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Inside this folder open the hardware folder and find and delete either the esp32 folder or espressif folder.

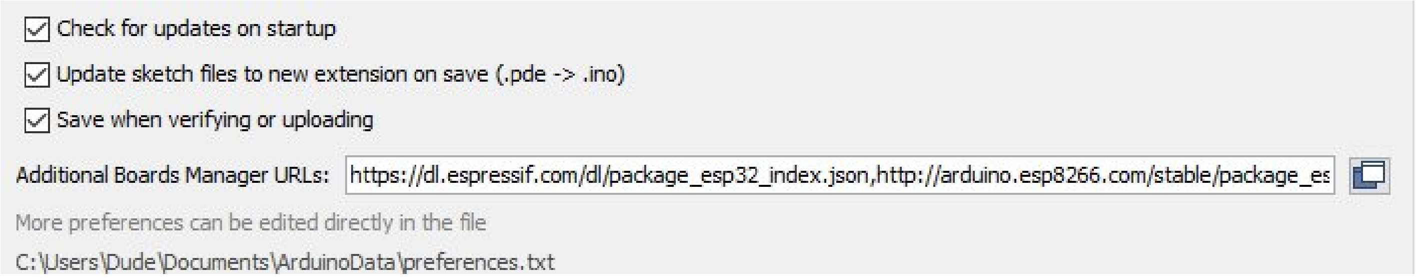


Name	Date modified	Type
esp32	21/09/2017 17:08	File folder

Now you can set up the ESP32 libraries the easy way. In the File > Preferences window in the IDE paste the following line into the Additional Boards Manager URL:

https://dl.espressif.com/dl/package_esp32_index.json

If you have entries in this field already then add the new line before them but separate them with a comma:



☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

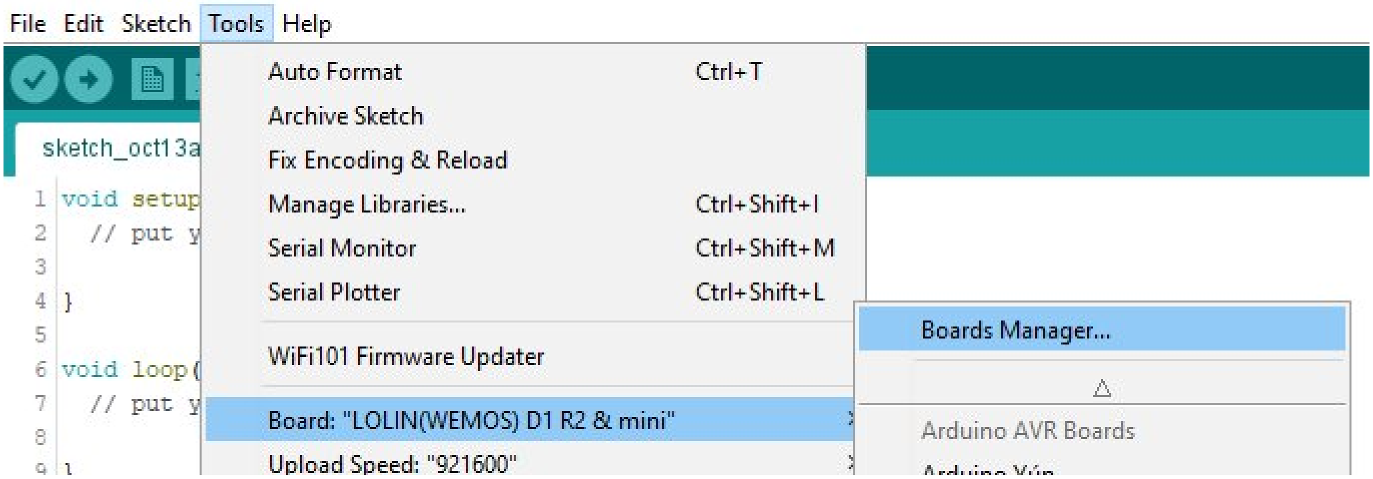
☑ Save when verifying or uploading

Additional Boards Manager URLs: https://dl.espressif.com/dl/package_esp32_index.json,http://arduino.esp8266.com/stable/package_es

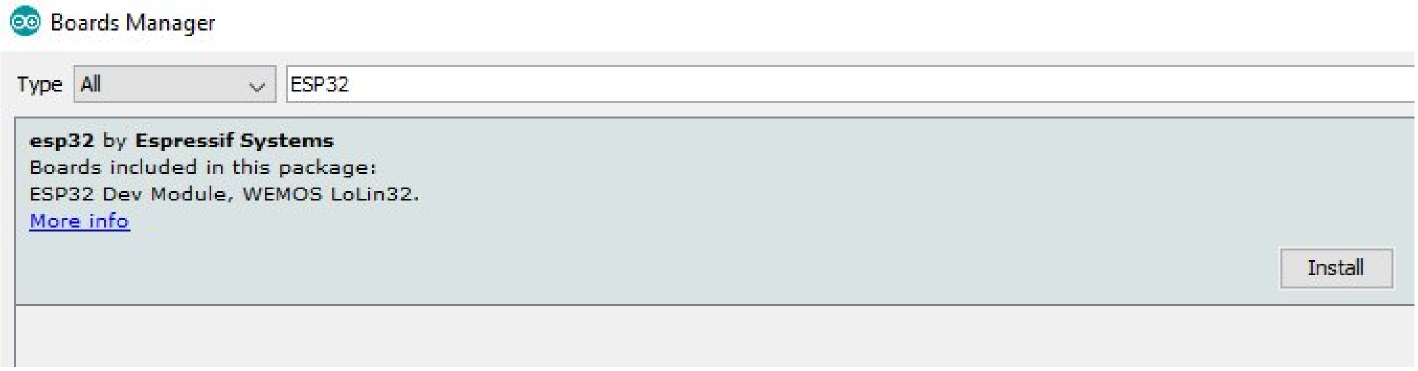
More preferences can be edited directly in the file

C:\Users\Dude\Documents\ArduinoData\preferences.txt

Then go to Tools > Board > Board Manager:

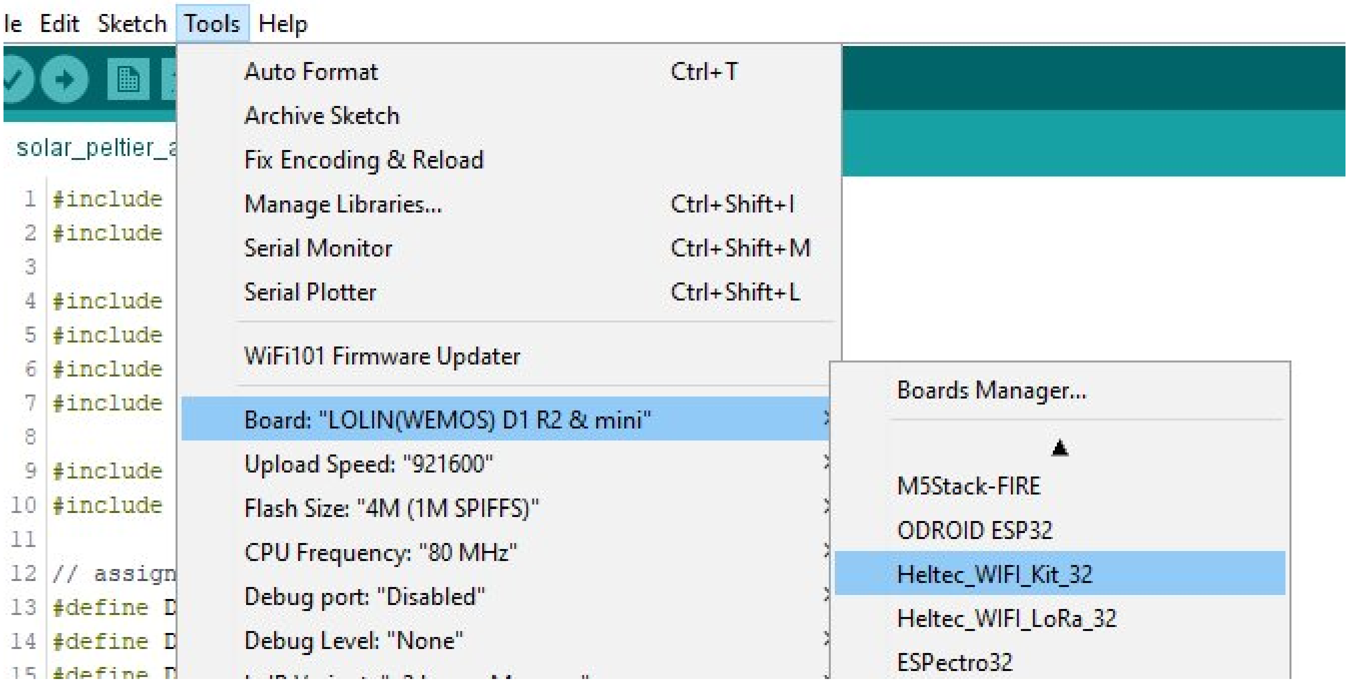


Type ESP32 and Install the new hardware libraries:



esp32 by **Espressif Systems**
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32.
More info

When the ESP32 library is installed you will be able to find all the ESP32 boards including the Heltec Wifi 32 in the IDE's board selector:



Previous Method

First unplug any boards you have connected and close the Arduino IDE if it's open.

For these boards to show in the Arduino IDE you have to install the hardware libraries locally using Git. Git is basically a way to keep local files synchronized with files on the internet. In this case it's used to download the files used by the IDE to work with the ESP32 boards that are available.

If you don't have Git installed then you need to download and install it from here: https://git-scm.com/download/win
After installation you should run Git GUI (should be in Programs in the Start menu). Click 'Clone existing repository' and...

- In the Source Location box enter: https://github.com/espressif/arduino-esp32.git
- In the Target Location box enter:
  C:/Users/[YOUR_USER_NAME]/Documents/Arduino/hardware/espressif/esp32
  replacing [YOUR_USER_NAME] with your login name. You can see this name on the Start menu by mousing over the grey circle icon.

Click Clone to start cloning the files to your PC. This might take a while.
When this has completed navigate to this directory:
C:/Users/[YOUR_USER_NAME]/Documents/Arduino/hardware/espressif/esp32/tools and double-click get.exe. Again this might take a while.

First Run

You can now plug in the Heltec board. Windows will attempt to install any necessary drivers. In my case I had to manually install the USB to UART driver from here:
https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

Everything should now be ready to test the board! So...

Start Arduino IDE, select your board in Tools > Board menu and select the COM port.

To test the board is basically working you can use the example WifiScan sketch here: File> Examples > Wifi > WifiScan If you open the Serial Monitor (Tools > Serial Monitor) you will be able to see any WiFi access points in range. Check the baud rate is the same as in the sketch – probably 115200.

Testing the Heltec ESP32 with the Onboard OLED

Once you have the ESP32 libraries installed and you've tested that the board can run the basic WifiScan sketch we can install the display libraries for the OLED.

My favourite display library for these OLEDs is the U8g2 (https://github.com/olikraus/u8g2) library. This is a lot easier to install as it can be found in the Arduino IDE library manager. Open Sketch > Include Library > Manage Libraries and search for and then install U8g2.

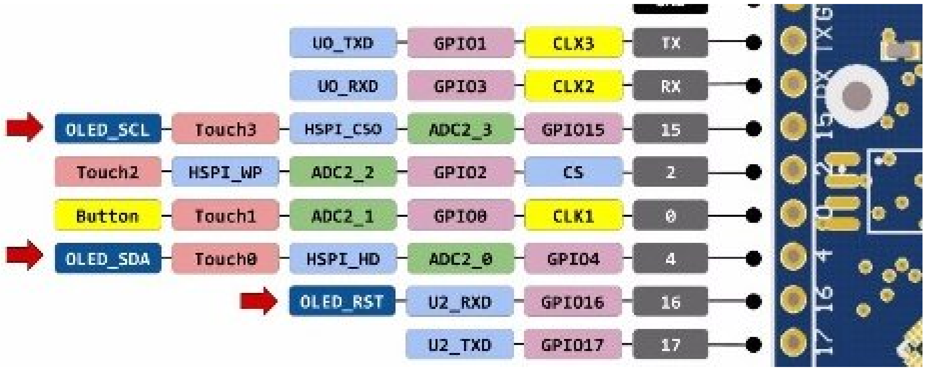
U8g2 has three different display methods. If you want to quickly test all three, the following examples show the correct constructor.

Full Buffer:
In the Arduino IDE: File> Examples > U8g2 > full_buffer > GraphicsTest
Paste: U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock=*/ 15, /* data=*/ 4, /* reset=*/ 16);
Above the line: // Please UNCOMMENT one of the contructor lines below
Upload the sketch

Page Buffer:
In the Arduino IDE: File> Examples > U8g2 > page_buffer > GraphicsTest
U8G2_SSD1306_128X64_NONAME_1_SW_I2C u8g2(U8G2_R0, /* clock=*/ 15, /* data=*/ 4, /* reset=*/ 16);
Above the line: // Please UNCOMMENT one of the contructor lines below
Upload the sketch

U8x8:
In the Arduino IDE: File> Examples > U8g2 > u8x8 > GraphicsTest
Paste: U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(/* clock=*/ 15, /* data=*/ 4, /* reset=*/ 16);
Above the line: // Please UNCOMMENT one of the contructor lines below
Upload the sketch

So, what are the numbers clock=*/ **15**, /* data=*/ **4**, /* reset=*/ **16** for in each of the examples above? These are pin numbers for the I2C controlled OLED on the board. This tells the display library which pins to use to communicate with the display.



U0_TXD – GPIO1 – CLX3 – TX
U0_RXD – GPIO3 – CLX2 – RX
OLED_SCL – Touch3 – HSPI_CS0 – ADC2_3 – GPIO15 – 15
Touch2 – HSPI_WP – ADC2_2 – GPIO2 – CS – 2
Button – Touch1 – ADC2_1 – GPIO0 – CLK1 – 0
OLED_SDA – Touch0 – HSPI_HD – ADC2_0 – GPIO4 – 4
OLED_RST – U2_RXD – GPIO16 – 16
U2_TXD – GPIO17 – 17

Show WifiScan Sketch Display on OLED

Below is a sketch that displays the results of the WiFi scan on the OLED using the U8x8 version of the display library

```
#include "WiFi.h"
#include <U8x8lib.h>

// the OLED used
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(/* clock=*/ 15, /* data=*/ 4, /* reset=*/ 16);


void setup()
{
  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  u8x8.begin();
  u8x8.setFont(u8x8_font_chroma48medium8_r);
}


static void doSomeWork()
{
  int n = WiFi.scanNetworks();

  if (n == 0) {
    u8x8.drawString(0, 0, "Searching networks.");
  } else {
    u8x8.drawString(0, 0, "Networks found: ");
    for (int i = 0; i < n; ++i) {
      // Print SSID for each network found
      char currentSSID[64];
      WiFi.SSID(i).toCharArray(currentSSID, 64);
      u8x8.drawString(0, i + 1, currentSSID);
    }
  }
  // Wait a bit before scanning again
  delay(5000);
}


void loop()
{
  doSomeWork();
}
```

Buy Me A Coffee

If I've saved you some time and you want to say 'Thanks!' you can buy me a coffee here...

🛒 (Paypal) ☕ €2

Or send me Bitcoin/Litecoin/Etherium 🙂

Post Views: 76,578