OEE Toolkit User Manual

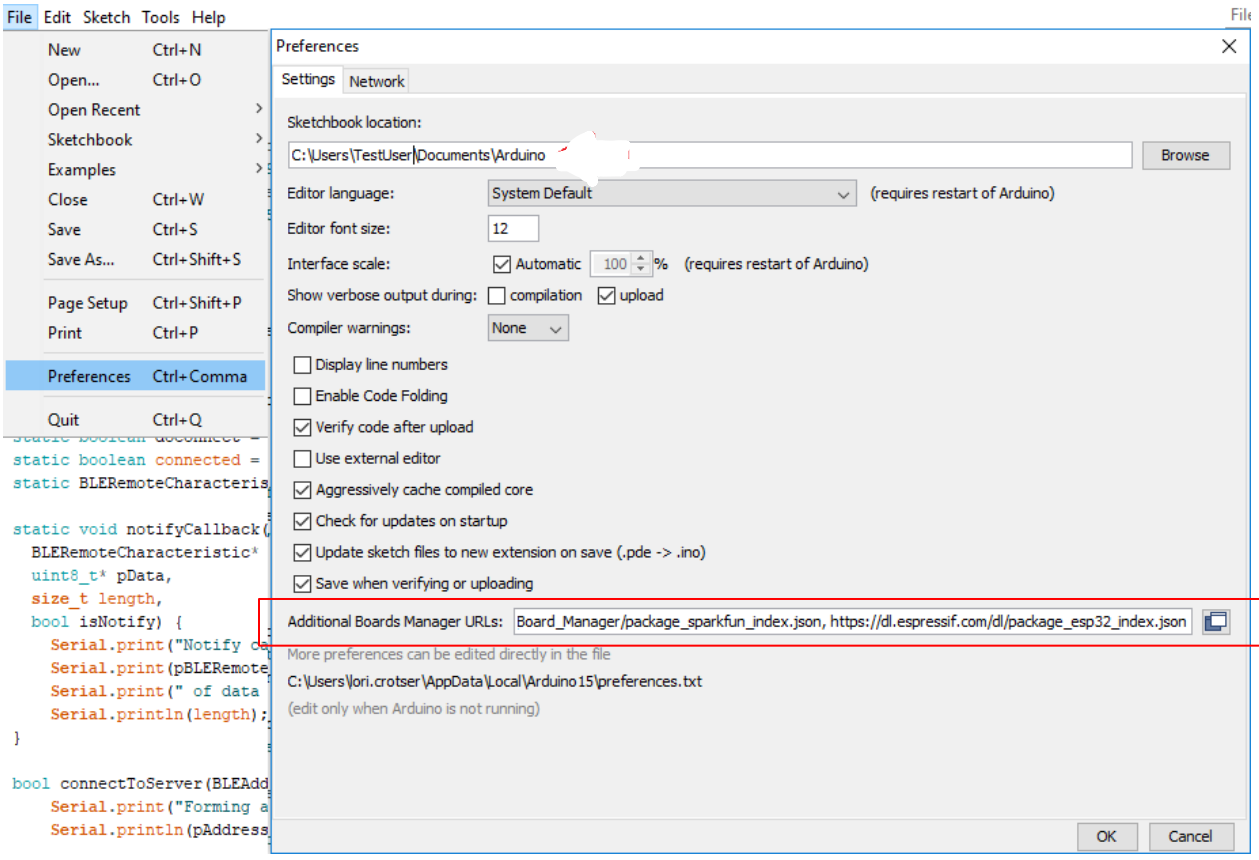
Arduino IDE Configuration Settings

1. Download and Install Arduino IDE from
   https://www.arduino.cc/en/software

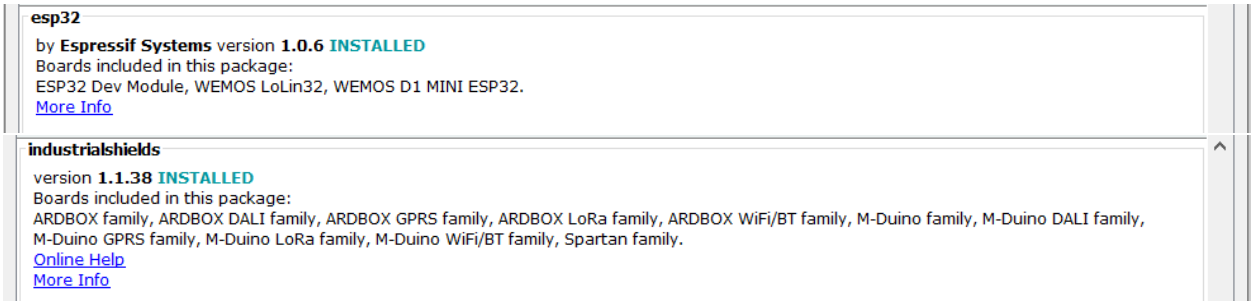
2. Download and Install CP2104 USB to UART Bridge VCP Driver from
   https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads

3. Paste the following preferences in the Additional Boards Manager URLs.
   https://dl.espressif.com/dl/package_esp32_index.json,
   http://apps.industrialshields.com/main/arduino/boards/package_industrialshields_index.json

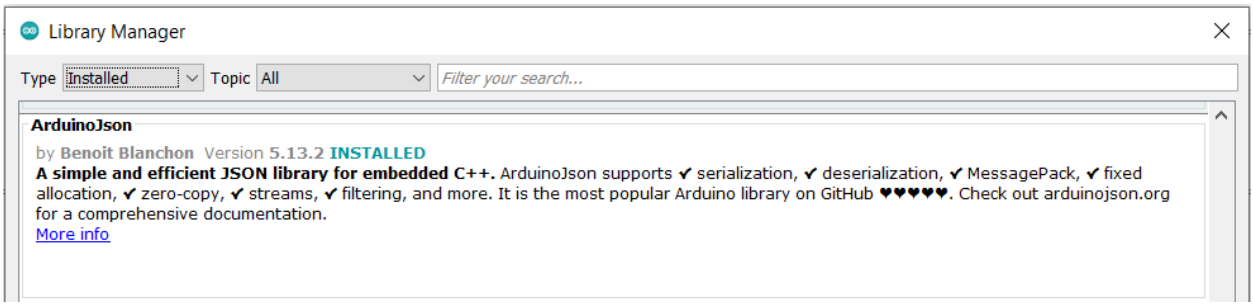
   This can be located by looking at your Sketchbook location under File > Preferences.



4. Now go in Tools > Board > Board Manager and download the following boards

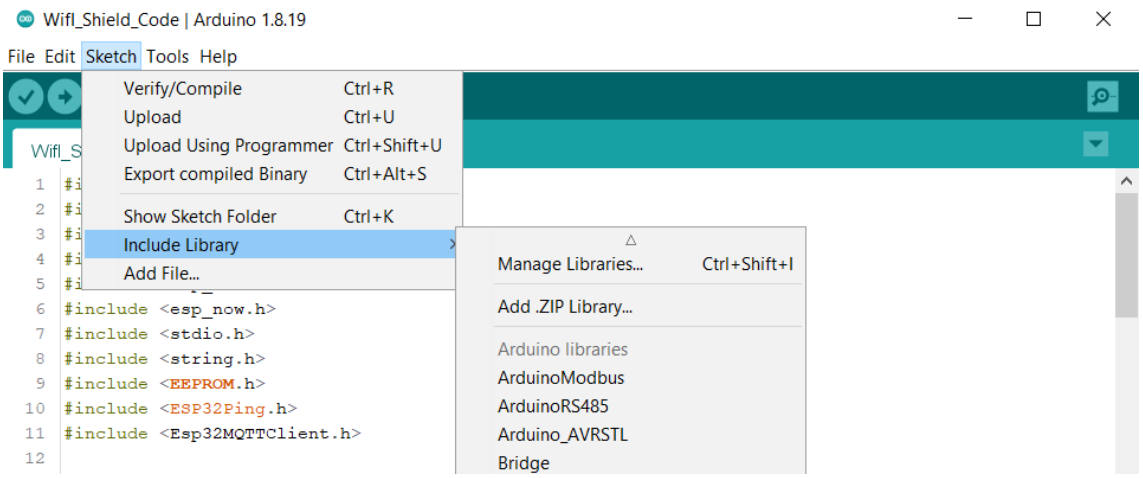


5. Go in Sketch > Include Library > Manage Libraries and install the following library with below 6.0 version



6. Download Esp32 Ping library Zip file from https://github.com/marian-craciunescu/ESP32Ping

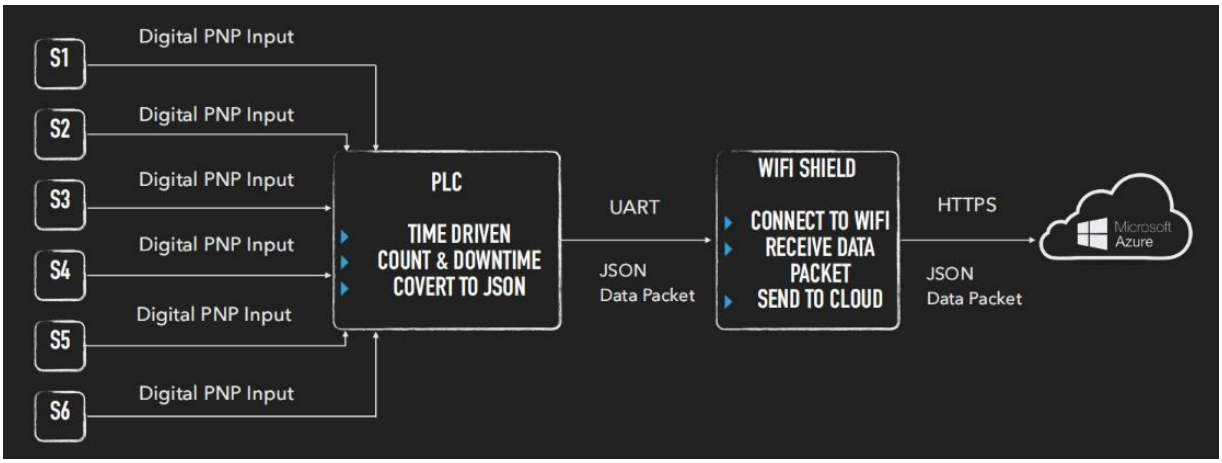
And include it in the Arduino IDE Libraries from Sketch > Include Library > Add .Zip Library



Congratulation! Now you can use PLC and Wi-Fi Shield Codes.

System's Architecture and Toolkit's Configuration

End to End Flow of Data (Device to Cloud)



Follow these articles for understanding of code:

Spark fun's Official Guide
https://learn.sparkfun.com/tutorials/esp32-thing-plus-hookup-guide/introduction

ESP32 to Azure IoT Hub via MQTT
https://automation.baldacchino.net/?p=952

Connect ESP32 to Azure IoT Hub via Rest
https://thingpulse.com/how-to-connect-your-esp32-to-the-azure-iot-cloud-over-rest/

System has 3 architectures running in parallel:

Time Driven Architecture

It will generate packet every 5 minute.

Event Driven Architecture

It will make sure to keep dashboard status in sync by generating data packet whenever status of the line changes.

Disconnection Recovery Architecture

This will store the data when the is a disconnection in the network.

Disconnection Data Recovery

1. Disconnection data holding capacity depends upon the number of sensors. For 1 sensor it is 6 hours, for 2 it is 4 hours, for 4 sensors it is 3 hours, and for 6 sensors it is 2 hours.

2. Data recovery is only possible if the toolkit does not loss its power while in disconnect state because it will make the system to restart from the scratch when it gets power again.

Code Uploading Activity :

1. When uploading the code, main breaker should be down i.e. Power must be OFF.

2. Make sure the no. of sensors connected on respective pins in hardware are the same as in the code.

3. Confirm the proper serial communication as:
   > TX-RX of Wi-Fi Shield and PLC must be connected via proper Wiring.

4. Upload the Code to PLC as:
   > Select board Industrial Shields > M-Duino family > M-Duino 21+
   > Select proper COM Port

5. Upload the Code to Wi-Fi Shield as:
   > Select board ESP32 > Spark fun ESP32 Thing Plus
   > Select proper COM Port

6. Now upload the Wi-Fi credentials and Azure IoT Hub Connection String in Wi-Fi Shield via BLE App as:
   > First send connection string then send Wi-Fi credentials e.g. ssid,pass
   > Make sure there is no space in between, or in the end.

7. Restart the toolkit and notice that it will send first packet after 10 sec (PTS : 10000). After this it will set data sending frequency to 5 minutes.

8. Wi-Fi Shield keep rebooting itself every 1 minute to stay connected with azure IoT Hub, and to keep the dashboard sync with the line status whether it is Up or Down via implementation of Event Driven Architecture.

Note : Verify the Credentials, Azure Connectivity and Data Receiving/Sending from Serial Monitor of Wi-Fi Shield and Azure IoT Hub’s Transaction Search.