# Hardware Setup:

1. Connect pin 32 and 33 of ESP32 to SCL and SDA of I/O card respectively.
2. Connect +5V power source to I/O card.
3. Toggle Reset pin of all MCP23017 LOW then HIGH.

# ESP32 Setup:

1. Install latest version of Arduino IDE from <https://www.arduino.cc/en/main/software>
2. Open ‘Preferences’ from ‘File’ drop down menu.
3. In ‘Additional Boards Manager URLs’ enter <https://dl.espressif.com/dl/package_esp32_index.json>
4. Open ‘Board Manager’. Tools > Board > Boards Manager.
5. Search for ESP32 and press install button for the “ESP32 by Espressif Systems“
6. Files download for ESP32 support will begin automatically. (Let the download complete).

More detailed information at: <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

# Host Setup:

Host is setup using Arduino IDE

1. Setup host IP address as:

IPAddress local\_IP(192, 168, 1, 107);

IPAddress gateway(192, 168, 1, 1);

IPAddress subnet(255, 255, 255, 0);

1. Setup UDP port to listen to as:

unsigned int localUdpPort = 4210;

# Client Setup:

Python sockets programming is used to setup UDP/IP connect between clients with host.

1. Enter host details to send data to as:

server\_address = ('192.168.1.107', 4210)

1. Use ‘pack(datatype, data)’ to create a packet to send to host.

(Datatype B: Byte)

|  |  |  |  |
| --- | --- | --- | --- |
| (Byte 0) Header 1: 0x7D | | | |
| (Byte 1) Header 2: 0xD7 | | | |
| (Byte 2) Function | Configure: 0x01 | Input: 0x02 | Output: 0x03// need to consider data again (no data sent) |
| (Byte 3)  Sub-Function | Configure all pins: 0x01 | Read all digital input pins: 0x02 | Write all output pins: 0x06 |
| Read specific digital input pin: 0x03 | Write specific output pin: 0x07 |
| Configure 7-segment port: 0x08 | Read all analog input pins:  0x04 | Write data to specific 7-segment: 0x10 |
| Configure Stepper motor port and sensor pins: 0x09 | Read specific analog input pin: 0x05 | Write data to all 7-segment: 0x11 |
| (Onwards Byte 4) Data | Configure all pins: Data: 4-19 bytes  (config for 128 pins) | Read all digital input pins: 0x90 | Write all output pins LOW: 0x90  Write all output pins HIGH: 0x91 |
| Read specific digital input pin: Input pin number (decimal value: 0 – 127) |
| 7-Segment port: 2-Bytes data | Read all analog input pins: 0x91 | Byte 4: Write specific output pin: Output pin number  Byte 5: Pin status  0x01 : HIGH  0x00 : LOW |
| Stepper motor port; Sensor Pin: 3-Bytes data | Read specific digital input pin: Input pin number (decimal value: 0 – 127) |
| Byte 4: 7-segment port (0-15)  Byte 5-9: Digits to be displayed |
|  |
| (2nd last Byte) Footer1: 0xFF | | | |
| (Last Byte) Footer2: 0xFA | | | |

# Packet Table

|  |  |  |
| --- | --- | --- |
| **Configuration**  **(0x00)** | Configure all I/O Ports  (0x01) | Configure all ports Data: 16 Bytes (4-19 Bytes)  (16 ports for a single card) |
| Configure 7-Segment  (0x02) | 7-Segment port Data: 2 Bytes (4-5 Bytes) |
| Reset Configuration (0x03) | Single Byte (0x00) |
|  | Read all digital input ports (0x10) | Single Byte (0x90) [Any number greater than 0x7F can be set] |
| **Input**  **(0x01)** | Read specific digital input port (0x11) | Port: Single Byte between 0x00 – 0x0F |
| Read all analog inputs (0x12) | Single Byte (0x91) [Any number greater than 0x7F can be set] |
| Read specific analog input (0x13) | Pin: Single Byte between 0x00 – 0x0F |
| **Output**  **(0x02)** | Write all digital output ports (0x20) | Single Byte (0x90) |
| Write specific digital output port (0x21) | Port: Single Byte between 0x00 – 0x0F |
| Write specific 7-Segment display (0x22) | 6 Bytes (4-9 Bytes): 1st Byte Port of 7-segment, next 5 Bytes digits to be displayed |
| Write all 7-Segment displays (0x23) | Multiple of 6 (4th Byte onward) |
|  | | |