5V

Vcc

GND

GND

A4

SDA

A5

SCL

**Schematic Diagrams**

**Temp. Sensor ( LM75A )**

**Arduino UNO**

**Temp. Sensor ( LM75A )**

**Arduino Mega**

3.3V

Vcc

Arduino Mega

Vcc

Vcc

SCL 22

SDA 21

**Esp32**

GND

SCL

SDA

GND

GND

**Temp. Sensor ( LM75A )**

SCL

SDA

GND

SCL 21

SDA 20

**Temperature Sensor (LM75A)**

**NAME OF THE LIBRARY:**  
#include<Wire.h>

**LIST OF FUNCTIONS IN LIBRARY:**

1. **Wire.begin()** - When this function is called, it performs the following actions:

Initializes the I2C hardware on the Arduino board, setting up the appropriate registers and configurations.

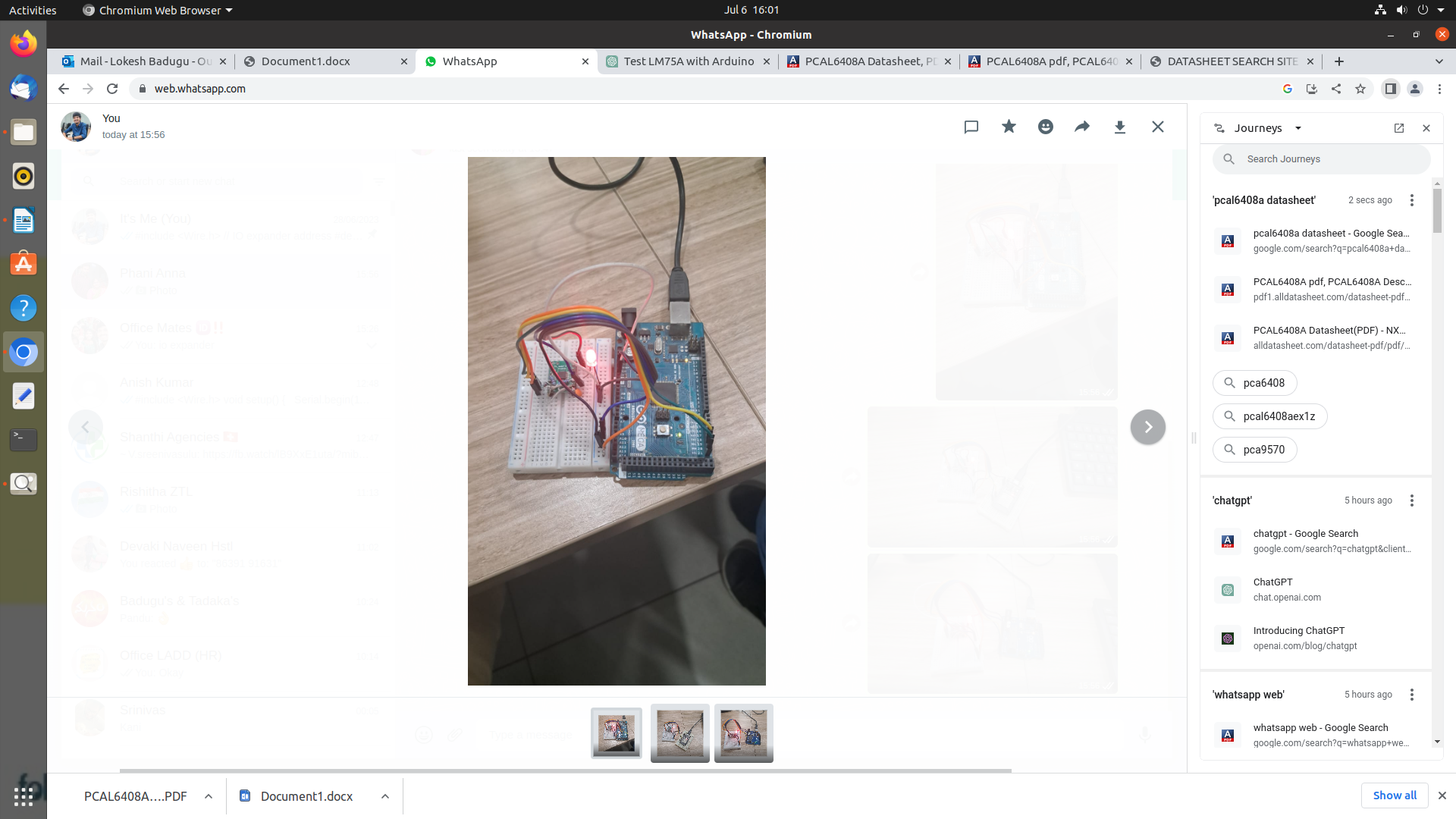
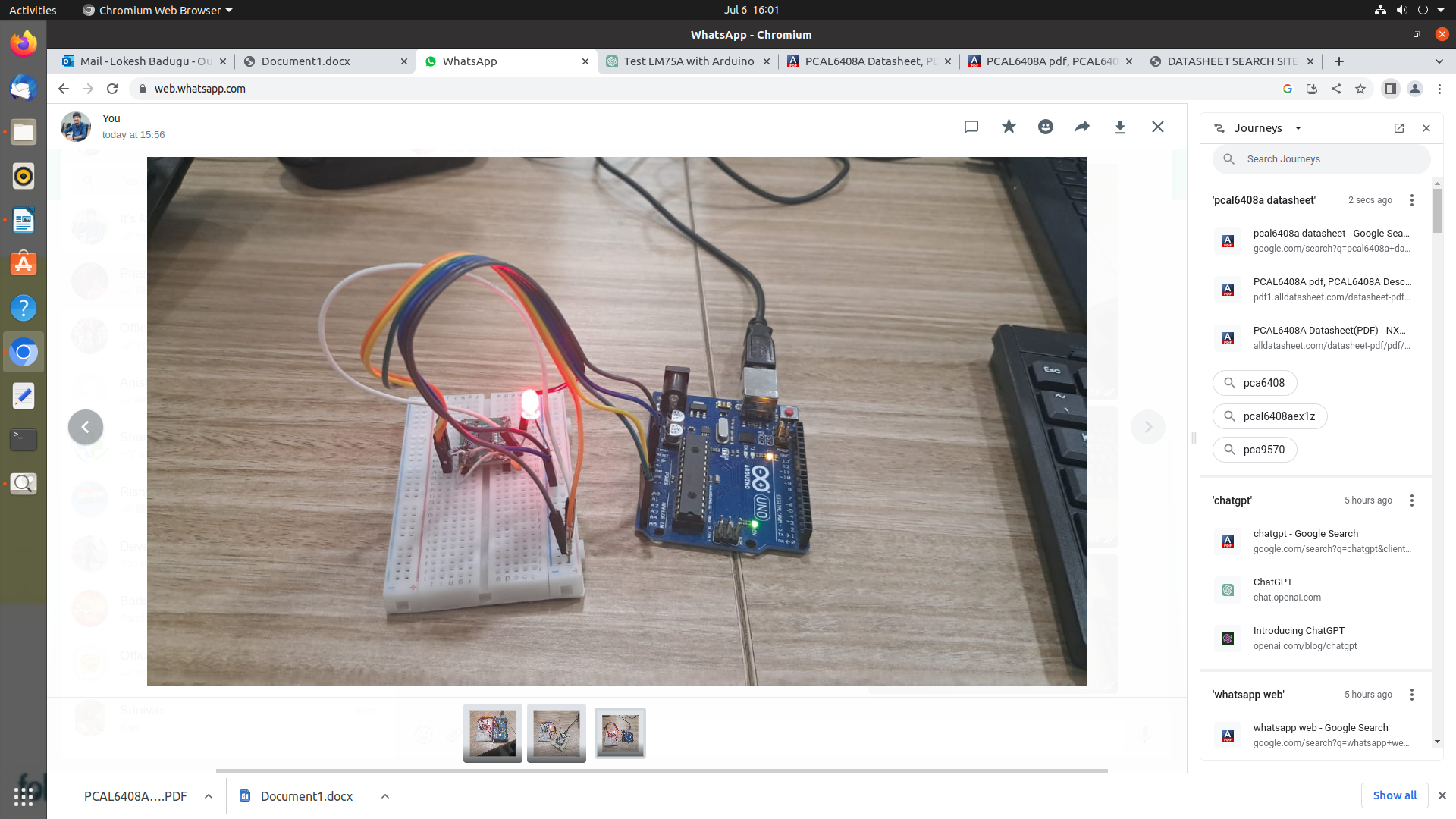
 Resets the I2C bus, ensuring a clean starting state for communication. Configures the Arduino board to act as the I2C master device.

2. **Wire.beginTransmission(Address)** - The Address parameter is the 7-bit address of the I2C slave device to which you want to send data. This address is typically provided in hexadecimal format (e.g., 0x68). When this function called, it starts the process of transmitting data to the specified I2C slave device. The function internally sets the I2C bus into transmission mode and sends the start condition (a start bit) to initiate communication. The address provided in the address parameter is sent on the I2C bus, indicating the target slave device for the upcoming data transmission.

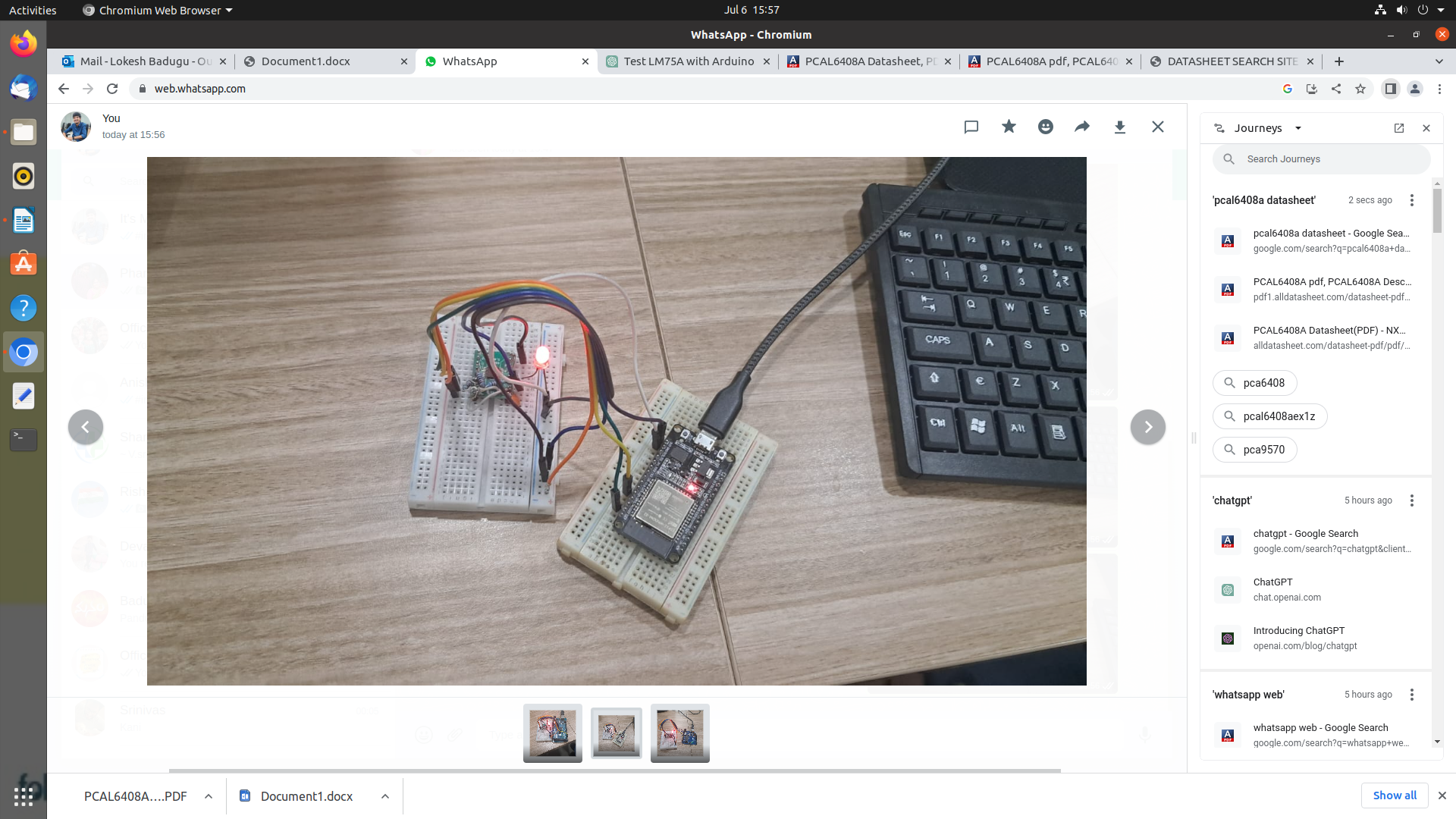
3. **Wire.write(Data)** - This function is called to send data over the I2C bus. The data to be sent is provided as an argument to the function. It can be a single byte or an array of bytes. The function internally sends the data over the I2C bus in the form of bytes.

4. **Wire.requestFrom(7-Bit Address, No. Of bytes)** - This function is called to initiate the process of requesting data from an I2C slave device. The function takes two arguments: the 7-bit address of the I2C slave device and the number of bytes to request from the device. The I2C bus is internally prepared for data reception, and a request is made to the specified I2C slave device. The I2C slave device is expected to respond by sending the requested number of bytes.  The Arduino board, acting as the I2C master, receives the response from the I2C slave device and stores the received data in an internal buffer.

5. **Wire.endTransmission()** function is called to complete the transmission of data over the I2C bus. It returns an integer value that indicates the status of the transmission. The return value can be used to determine if the transmission was successful or if an error occurred. The function internally sends the appropriate I2C stop condition, indicating the end of the transmission.



A5(SCL)



Vss-8

Vss

GND

SDA

A4(SDA)

SDA

A4 (SDA)

SCL

SCL

A5 (SCL)

RESET

**Arduino Mega**

**PCAL6408A**

Vdd(p)

3.3V

GND

Vss

Vdd(p) - 16

SDA 20

SDA

Vss - 8

SCL 21

SCL

RESET

**ESP32**

**PCAL6408A**

3.3V

Vdd(p)

GND

Vss

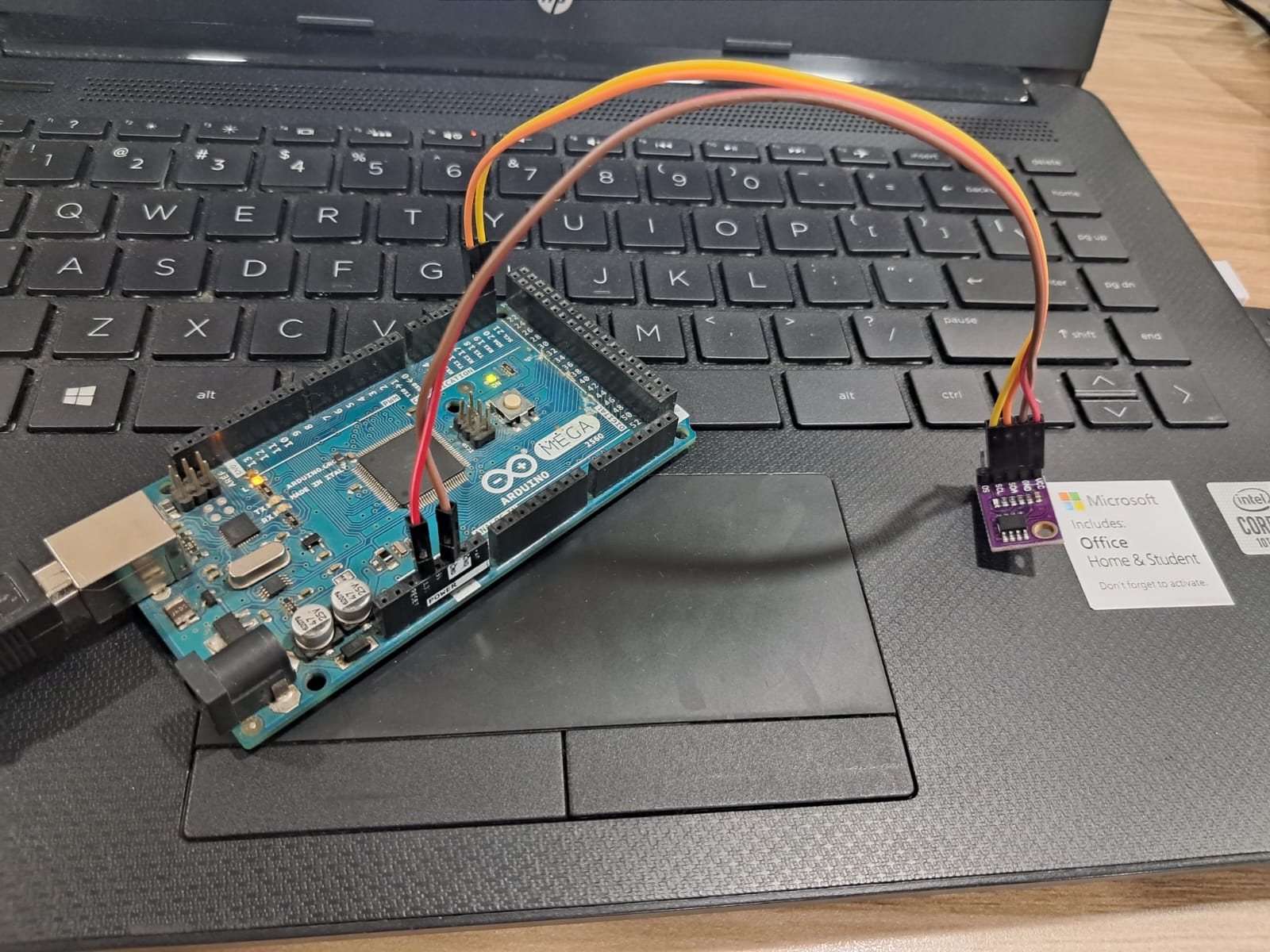
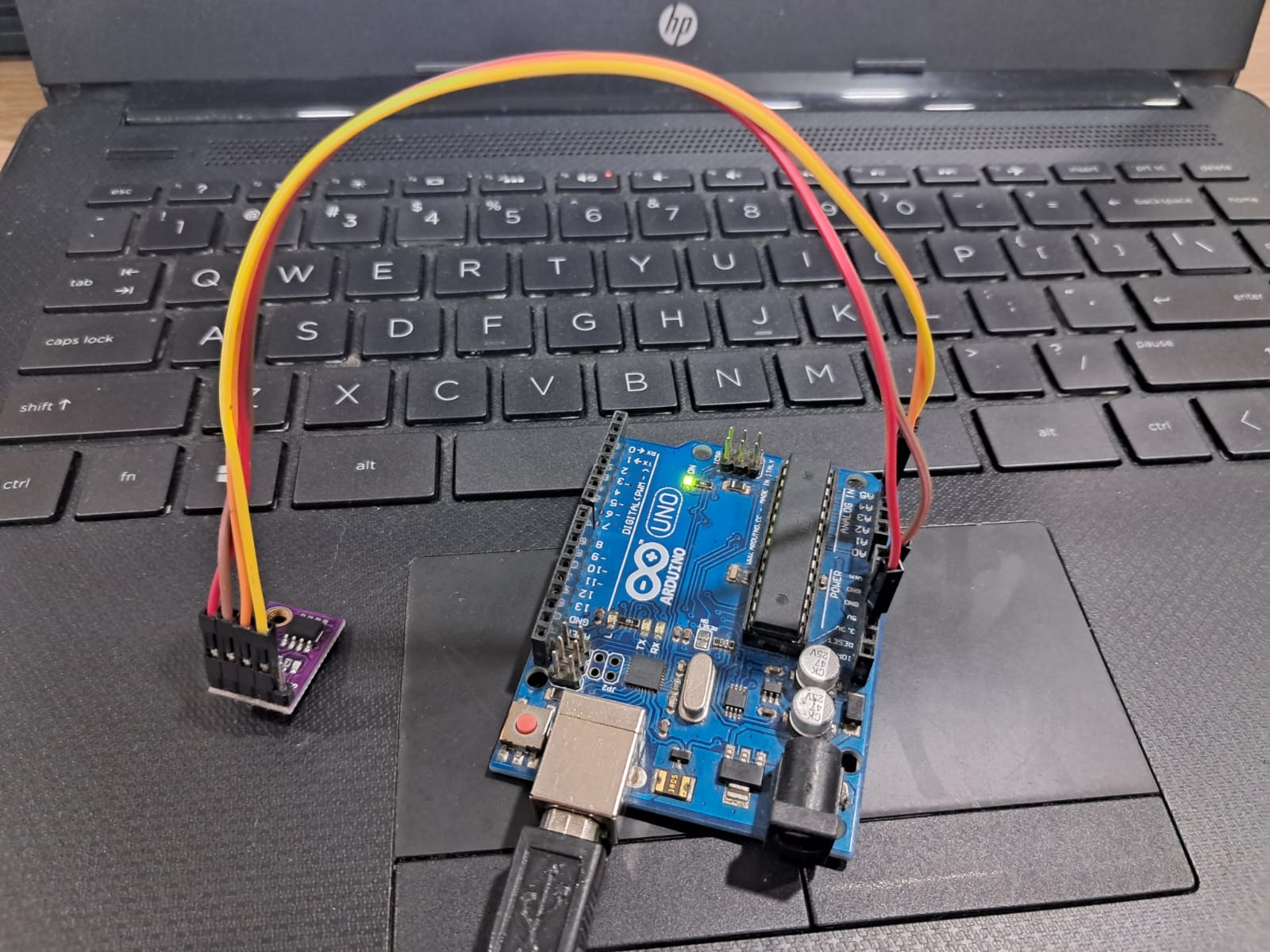
SDA 21

SDA

SCL

SCL 22

RESET

****

****

**NAME OF THE LIBRARY:**  
#include<Wire.h>

**LIST OF FUNCTIONS IN THE LIBRARY:**

1. **Wire.begin()** - When this function is called, it performs the following actions:

Initializes the I2C hardware on the Arduino board, setting up the appropriate registers and configurations.

 Resets the I2C bus, ensuring a clean starting state for communication. Configures the Arduino board to act as the I2C master device.

2. **Wire.beginTransmission(Address)** - The Address parameter is the 7-bit address of the I2C slave device to which you want to send data. This address is typically provided in hexadecimal format (e.g., 0x68). When this function called, it starts the process of transmitting data to the specified I2C slave device. The function internally sets the I2C bus into transmission mode and sends the start condition (a start bit) to initiate communication. The address provided in the address parameter is sent on the I2C bus, indicating the target slave device for the upcoming data transmission.

3. **Wire.write(Data)** - This function is called to send data over the I2C bus. The data to be sent is provided as an argument to the function. It can be a single byte or an array of bytes. The function internally sends the data over the I2C bus in the form of bytes.

4. **Wire.endTransmission()** function is called to complete the transmission of data over the I2C bus. It returns an integer value that indicates the status of the transmission. The return value can be used to determine if the transmission was successful or if an error occurred. The function internally sends the appropriate I2C stop condition, indicating the end of the transmission.