Aim: - To write C++ code to understand the ADC and Potentiometer.

Software Used: -

Cloud Keil Studio IDE and Tera Term

Theory: -

The mbed NXP LPC11U24 Microcontroller in particular is designed for prototyping low-cost USB devices, battery powered applications and 32-bit ARM® Cortex™-M0 based designs. It is packaged as a small DIP form-factor for prototyping with through-hole PCBs, stripboard and breadboard, and includes a built-in USB FLASH programmer.

Specifications of LPC11U24:

* NXP LPC11U24 MCU
* Low power ARM® Cortex™-M0 Core
* 48MHz, 8KB RAM, 32KB FLASH
* USB Device, 2xSPI, I2C, UART, 6xADC, GPIO
* Prototyping form-factor
* 40-pin 0.1" pitch DIP package, 54x26mm
* 5V USB, 4.5-9V supply or 2.4-3.3V battery
* Built-in USB drag 'n' drop FLASH programmer
* mbed.org Developer Website
* Lightweight Online Compiler
* High level C/C++ SDK
* Cookbook of published libraries and projects

API Used:

* Syntax used for digital output:
* DigitalOut (PinName pin)
* For LED blinking we’ve used:
* DigitalOut variable(LEDn); where n= 1,2,3,4
* For delay:
* wait(t); where ‘t’ is in seconds

Task 1: - Write a C++ code with mbed APIs to print the current voltage value of a potentiometer on PC serial monitor using ADC and serial port.

Code: -

#include "mbed.h"

Serial pc(USBTX, USBRX);

AnalogIn potent(A0);

int main()

{

    while(1)

    {

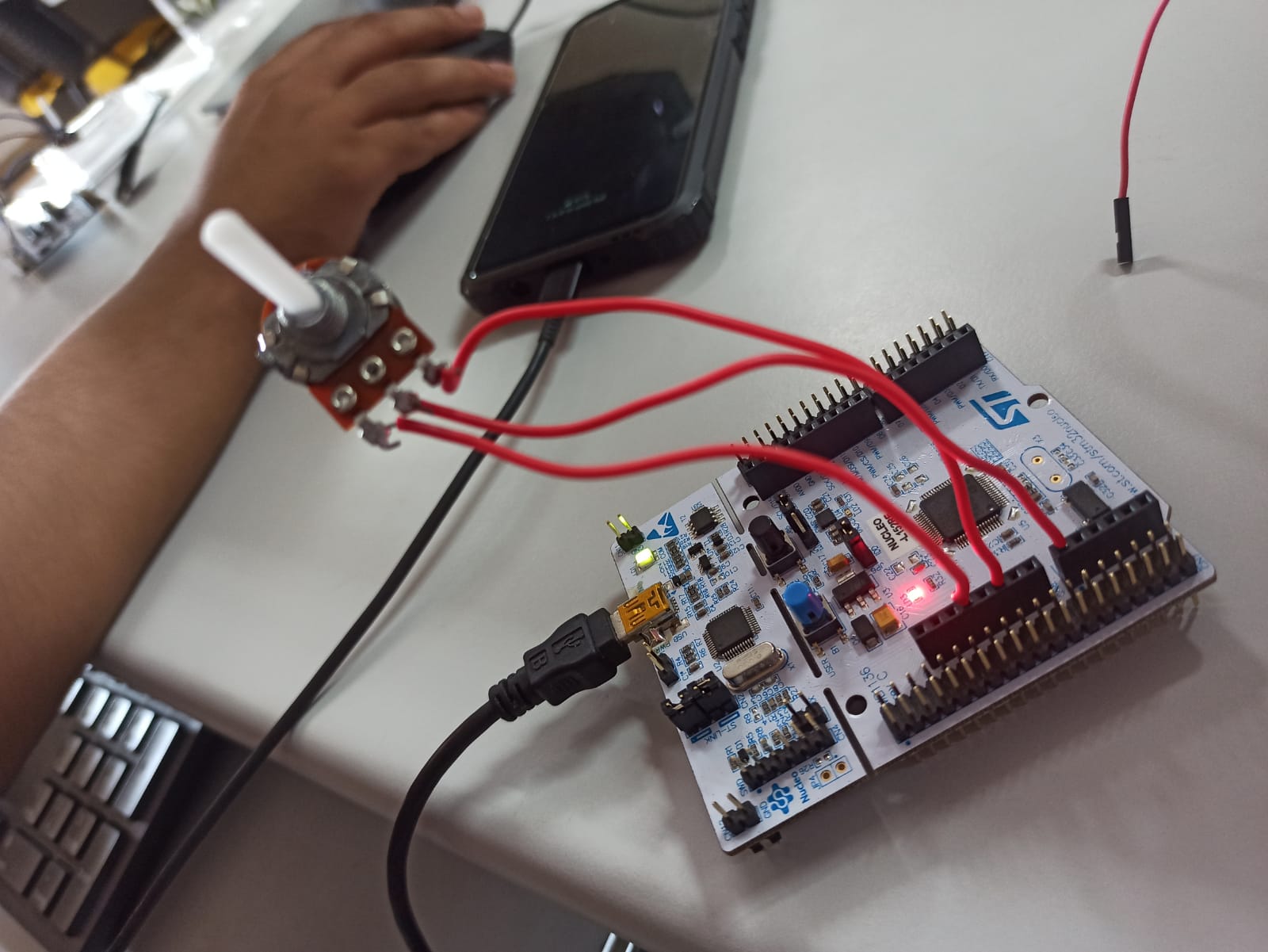
        pc.printf("Current volatge value %0.2f",potent.read());

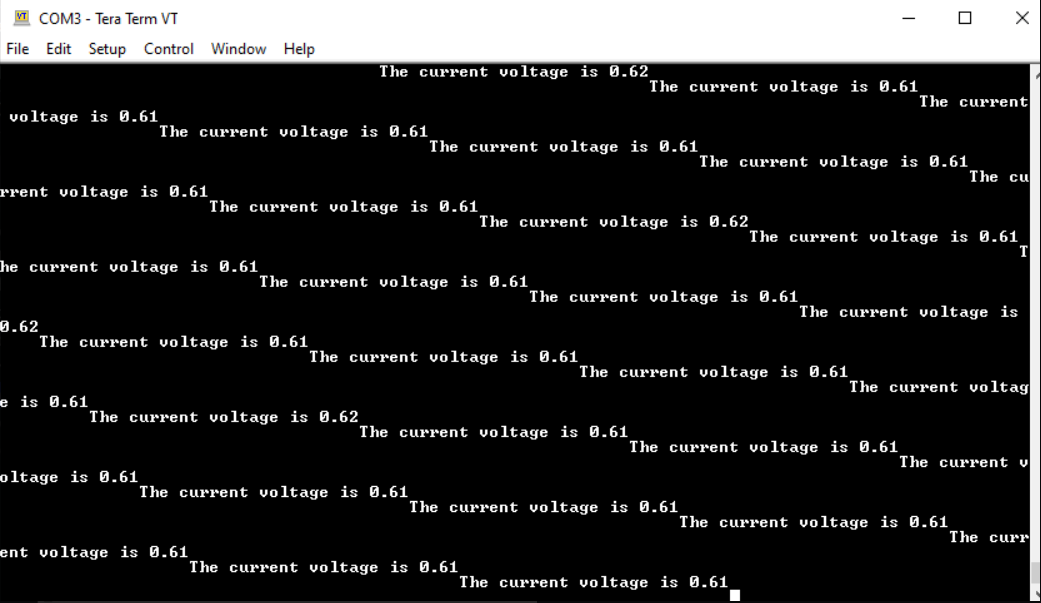
        wait(2);

    }

}

Output: -





Task 2: -Write a C++ code with mbed APIs to interface a LM35 with Nucleo board and display current temperature value on PC serial monitor

Code: -

#include "mbed.h"

#include "mbed2/299/drivers/AnalogIn.h"

Serial pc(USBTX, USBRX);

AnalogIn ain(PC\_3);

int main()

{

    while(1)

    {

        float sensorvalue ain = ain;

        float vout = sensorvalue3.3;

        float tempc = vout100;

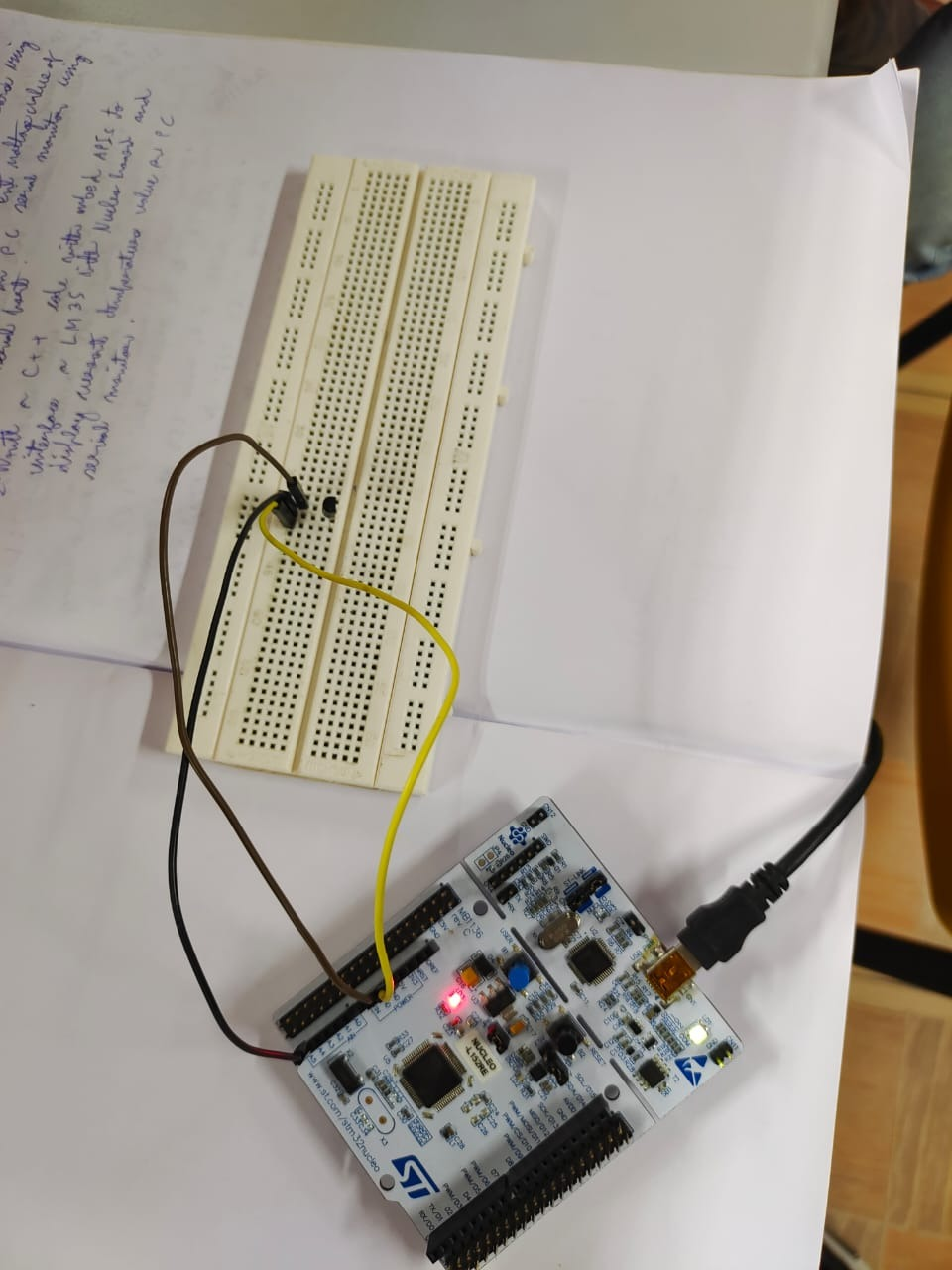
        pc.printf("The reading of the pot: %0.2f\n",tempc);

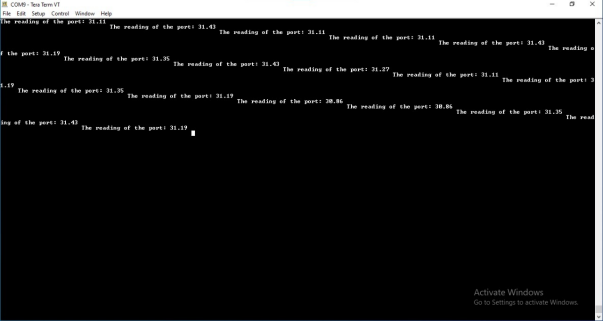
        wait(1);

    }

}

Output: -





Task 3: -Write a C++code with mbed APIs to implement a voltage level indicator using potentiometer and LEDs. (Refer table below)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Analog input value x | LED 1 | LED 2 | LED 3 | LED 4 |
| X<=0.2 | 0 | 0 | 0 | 0 |
| 0.2<x<=0.4 | 1 | 0 | 0 | 0 |
| 0.4<x<=0.6 | 1 | 1 | 0 | 0 |
| 0.6<x<=0.8 | 1 | 1 | 1 | 0 |
| 0.8<x<=1.0 | 1 | 1 | 1 | 1 |

Code: -

#include "mbed.h"

Serial pc(USBTX,USBRX);

AnalogIn ang(A0);

BusOut led(PC\_0,PC\_1,PB\_0,PA\_4);

int main() {

    float ang\_val;

    while(1) {

        ang\_val = ang.read();

        if (ang\_val<=0.2)

        {

            led = 0b0000;

        }

        else if (0.2<ang\_val && ang\_val<=0.4)

        {

            led = 0b1000;

        }

        else if (0.4<ang\_val && ang\_val<=0.6)

        {

            led = 0b1100;

        }

        else if (0.6<ang\_val && ang\_val<=0.8)

        {

            led = 0b1110;

        }

        else if (0.8<ang\_val && ang\_val<=1)

        {

            led = 0b1111;

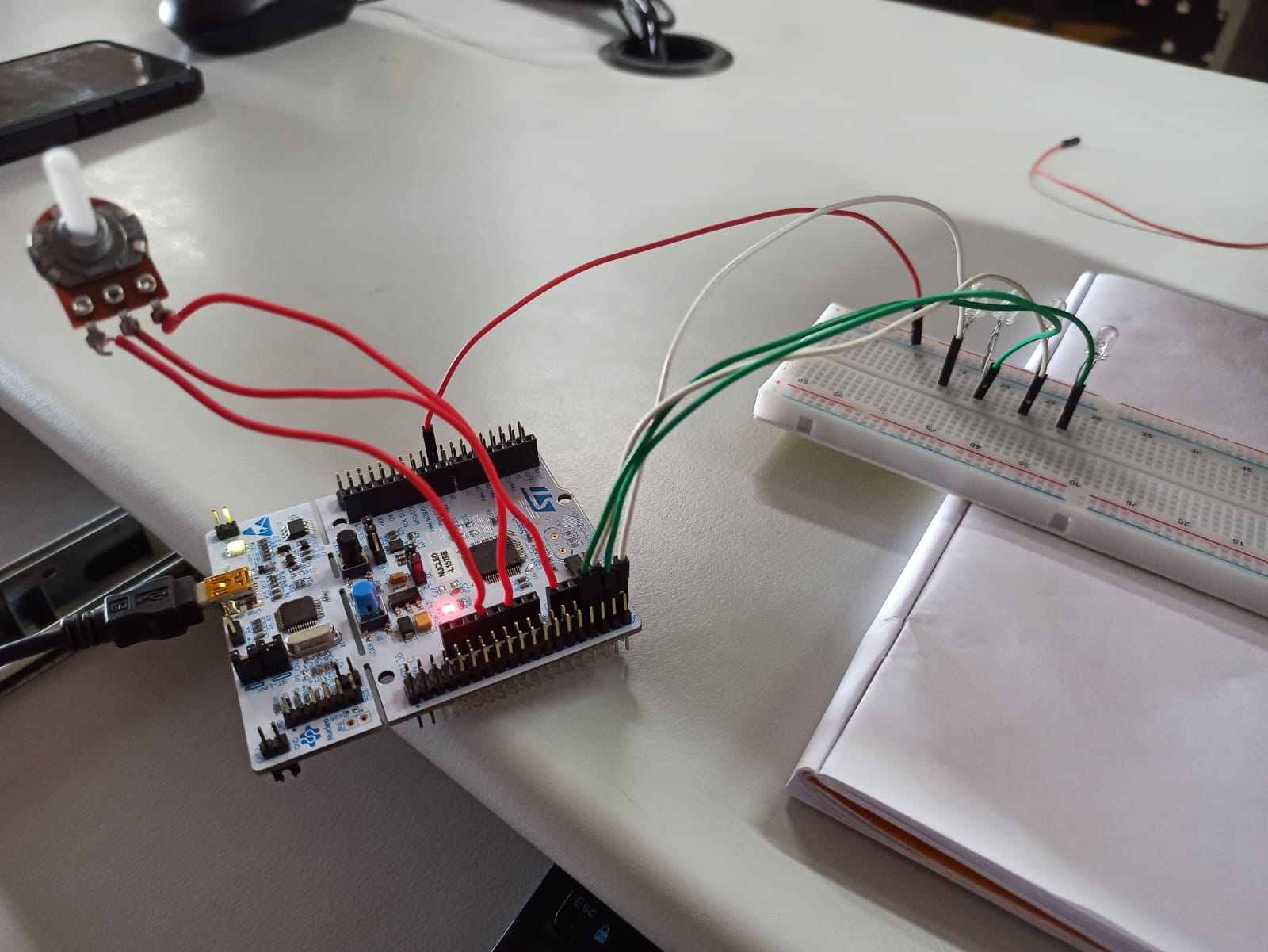
        }

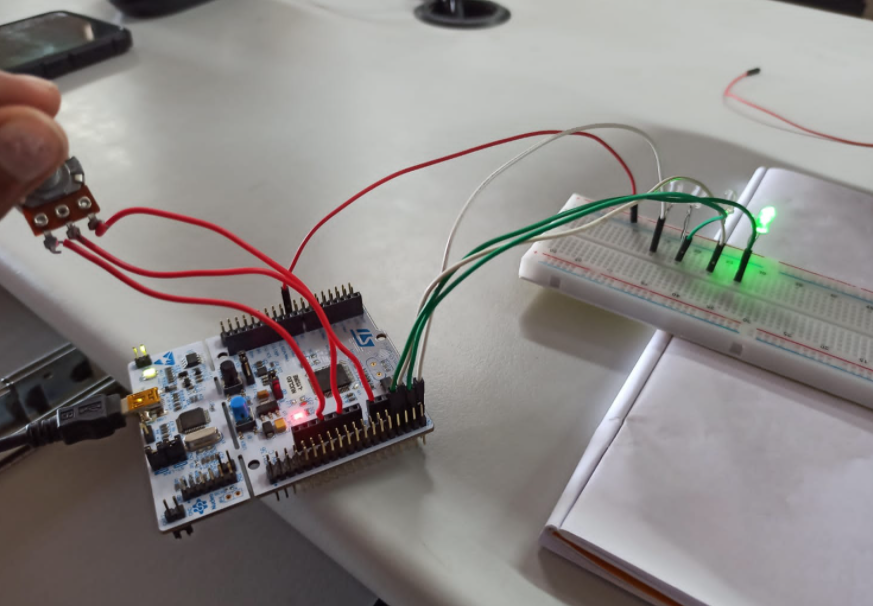
        wait(0.2);

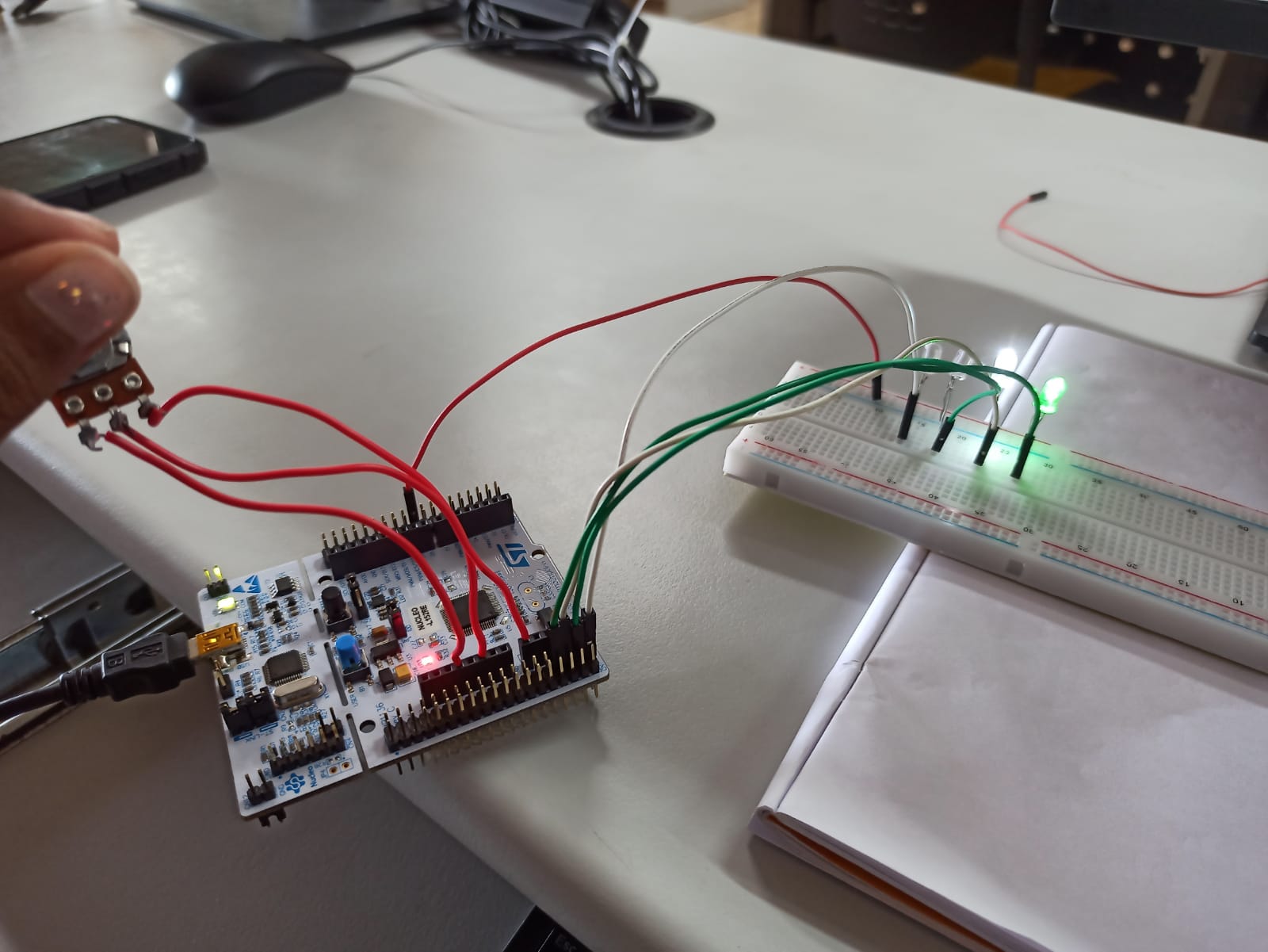
    }

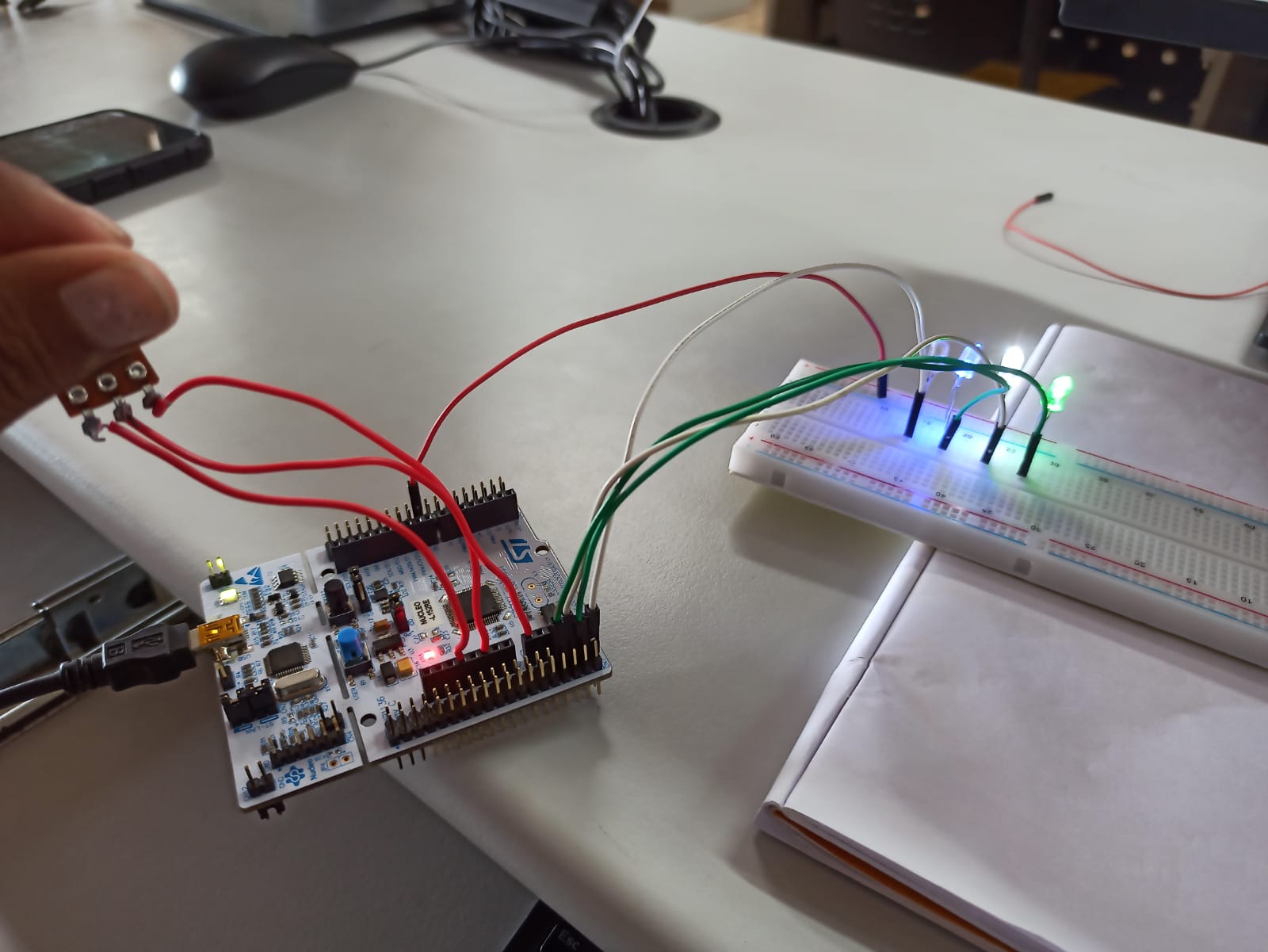
}

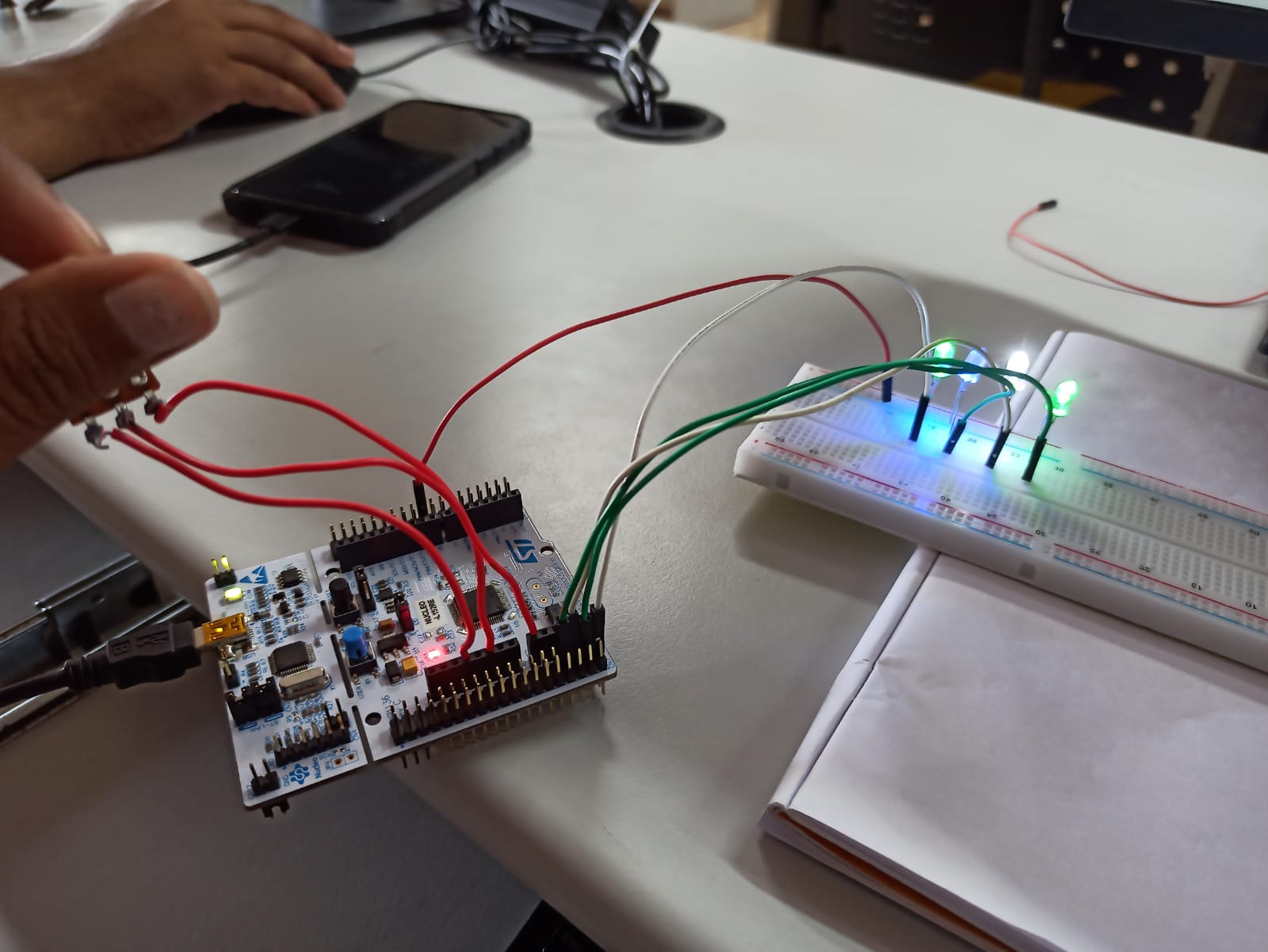
Output: -



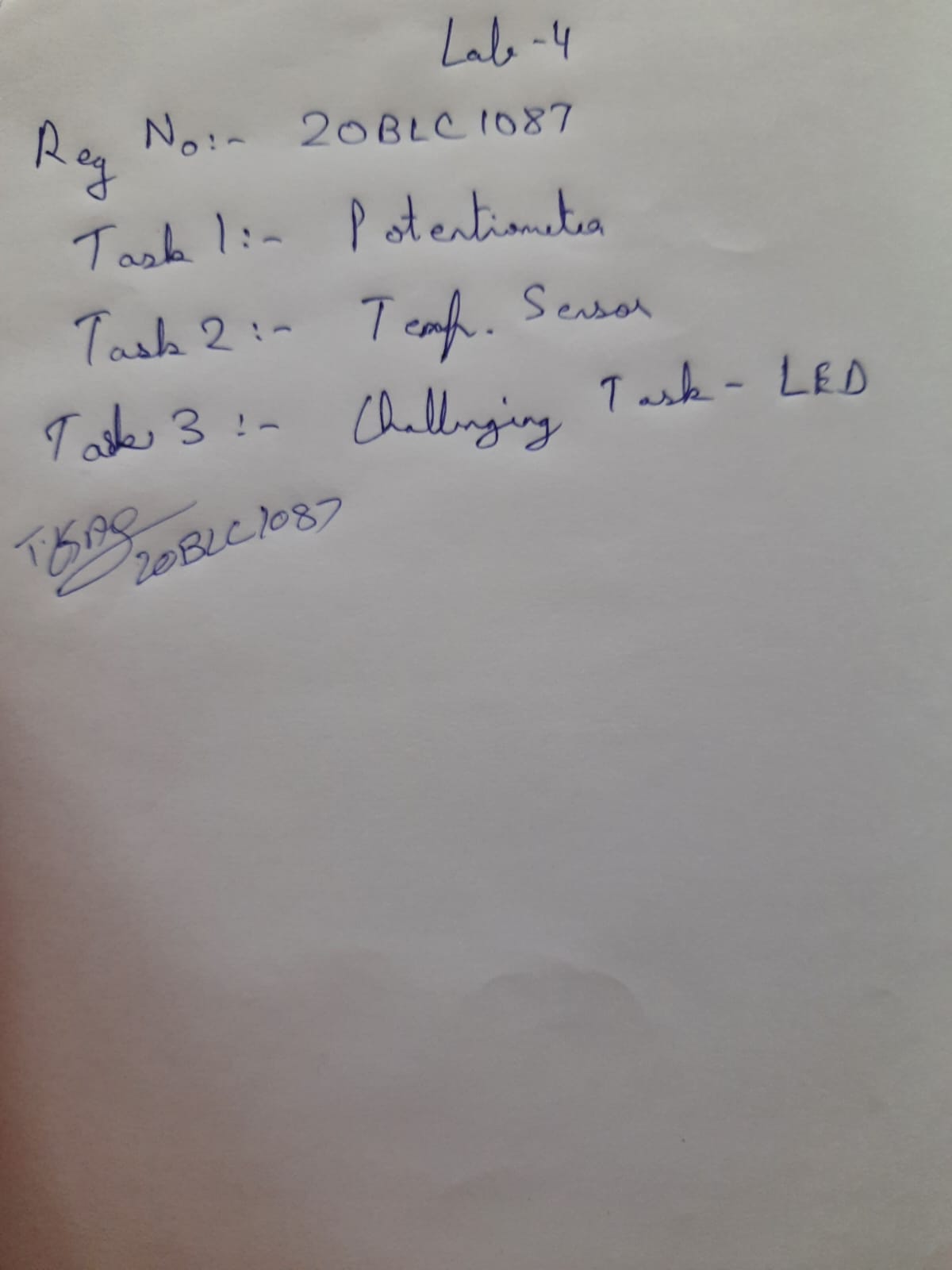


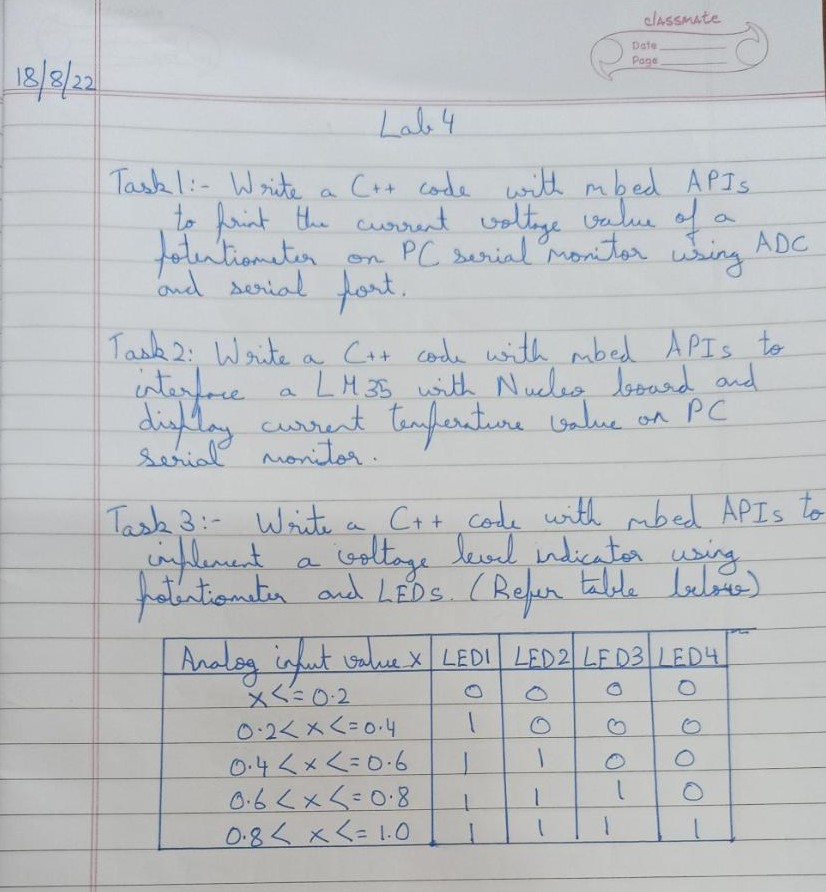






Verification Status: -





Result: -

Successfully understood ADC and Potentiometer concept on Keil Studio.