Aim: - To write C++ source code to understand the digital port access.

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 to transmit a message “VIT SENSE” using the STM port serial to pc for every 1 second.

Code: -

#include "mbed.h"

Serial pc(USBTX, USBRX);

int main()

{

    while(1)

    {

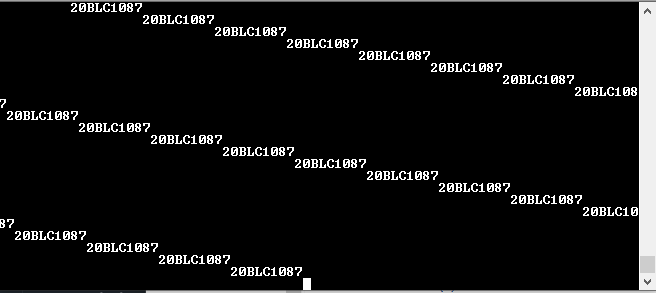
        pc.printf("20BLC1087\n");

        wait(1);

    }

}

Output: -



Task 2: - Write a C++ code with mbed APIs to receive a character(s) from PC and Switch ON the on-board LED. For all other character LED must be in OFF state.

Code: -

#include "mbed.h"

Serial pc(USBTX, USBRX);

DigitalOut myled(LED1);

int main()

{

    pc.printf("Press a character");

    while (1)

    {

        char y = pc.getc();

        if (y=='s')

        {

            myled=1;

            wait(1);

        }

        else

        {

            myled=0;

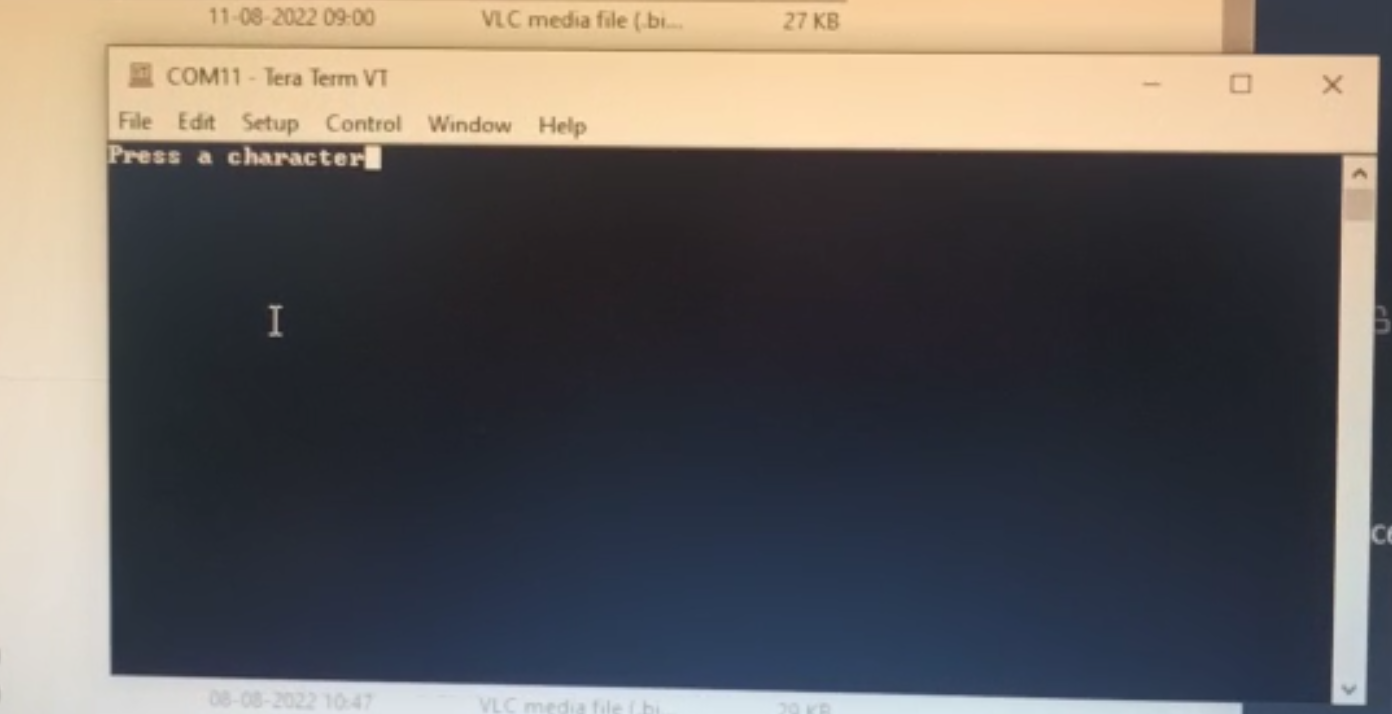
            wait(1);

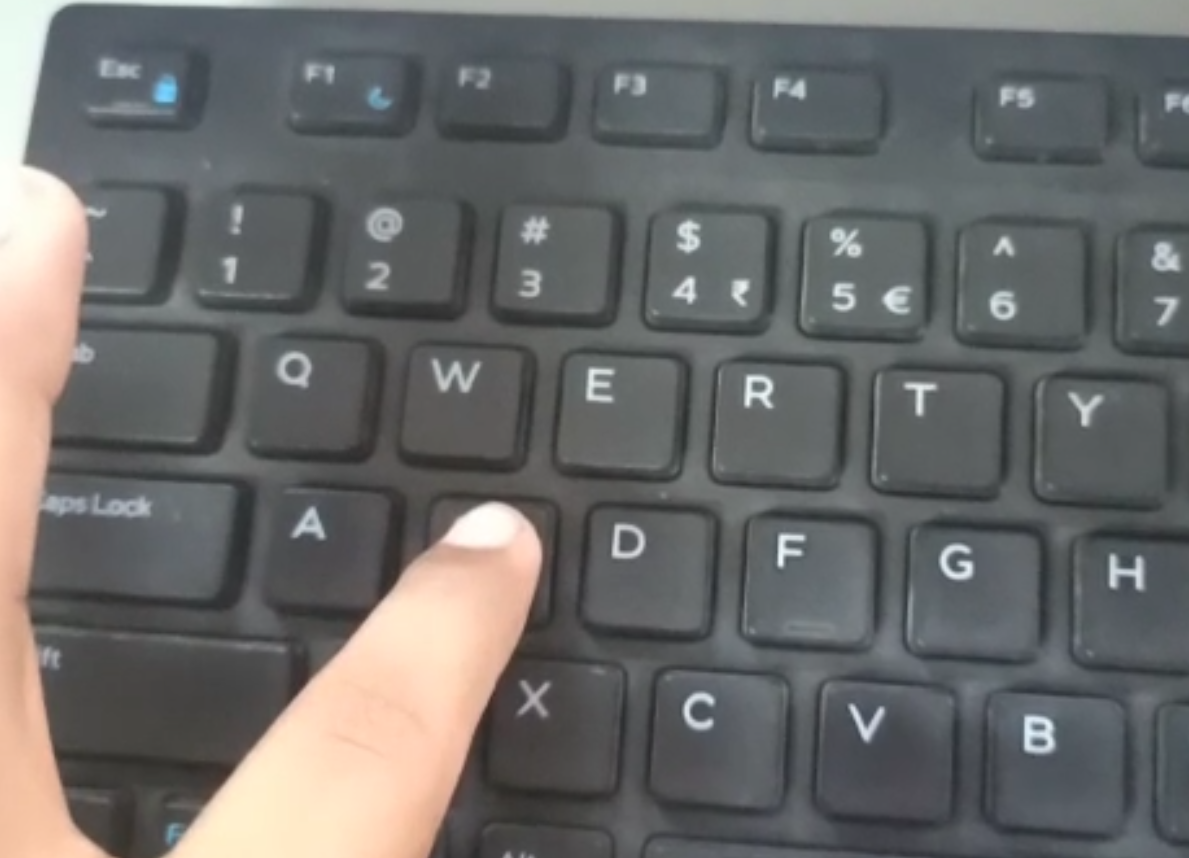
        }

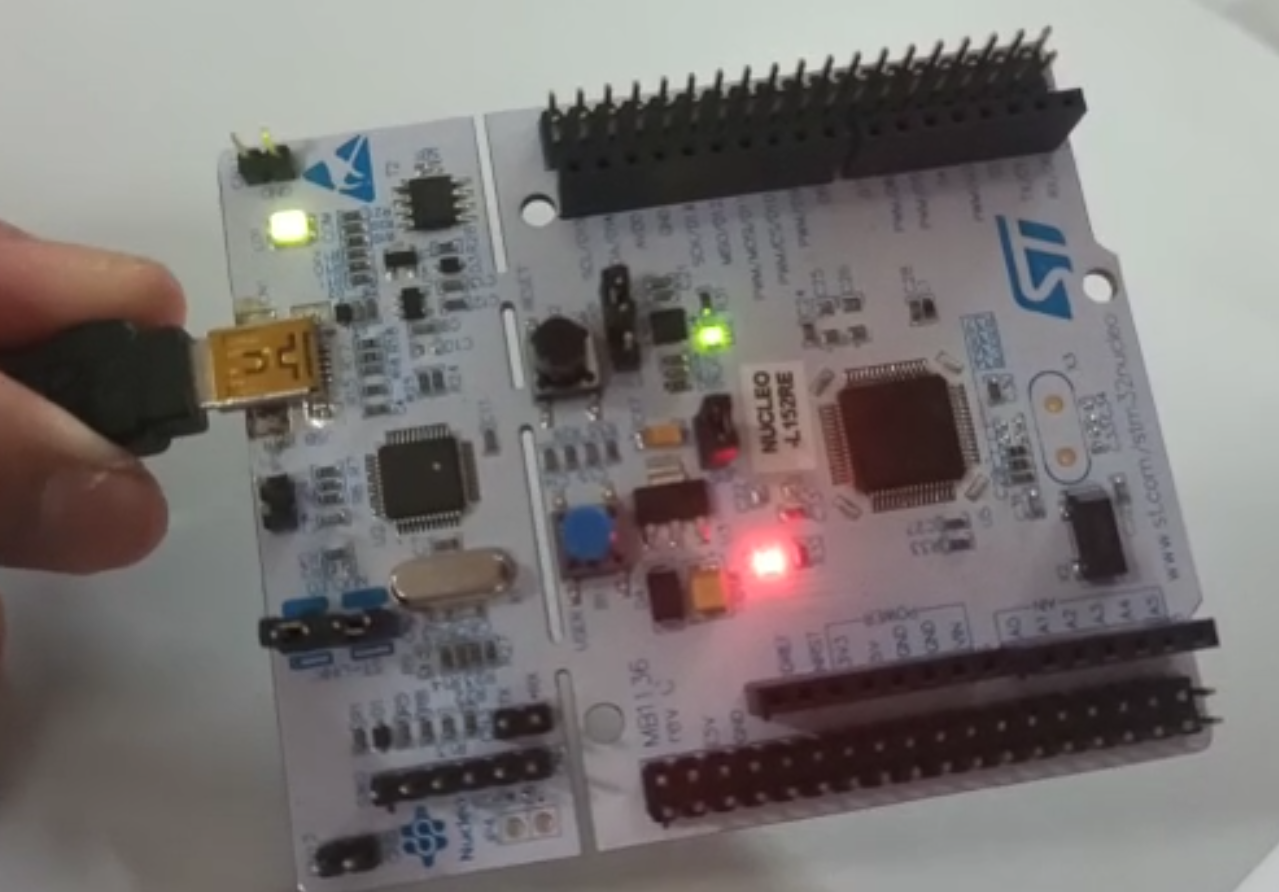
    }

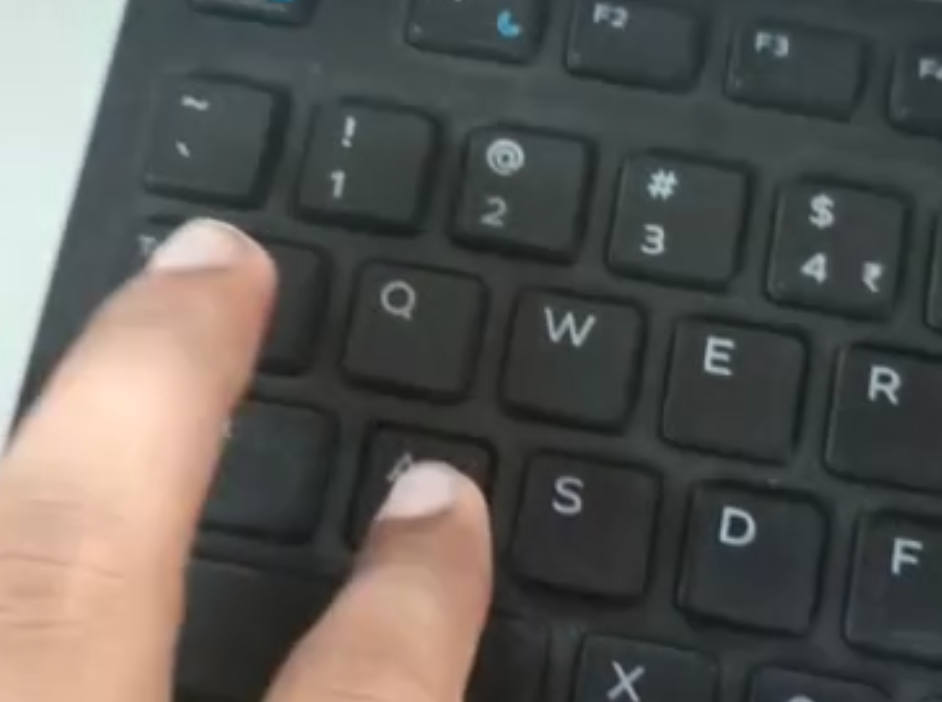
}

Output: -











Task 3: -

Write a C++ code with mbed APIs to interface a serial port, switch (PB\_13) and LED (PC\_4). Perform the below operation:

When switch=1, make LED ON and sent a message “LED ON” to PC serial monitor.  
When switch=0, make LED OFF and sent a message “LED OFF” to PC serial monitor.

Code: -

#include "mbed.h"

Serial pc(USBTX, USBRX);

BusIn inps (PB\_13);

DigitalOut myled(PC\_4);

int main()

{

    if (inps == 0b01)

    {

        myled=1;

        pc.printf("LED ON");

        wait(1);

    }

    else if (inps == 0b00)

    {

        myled=0;

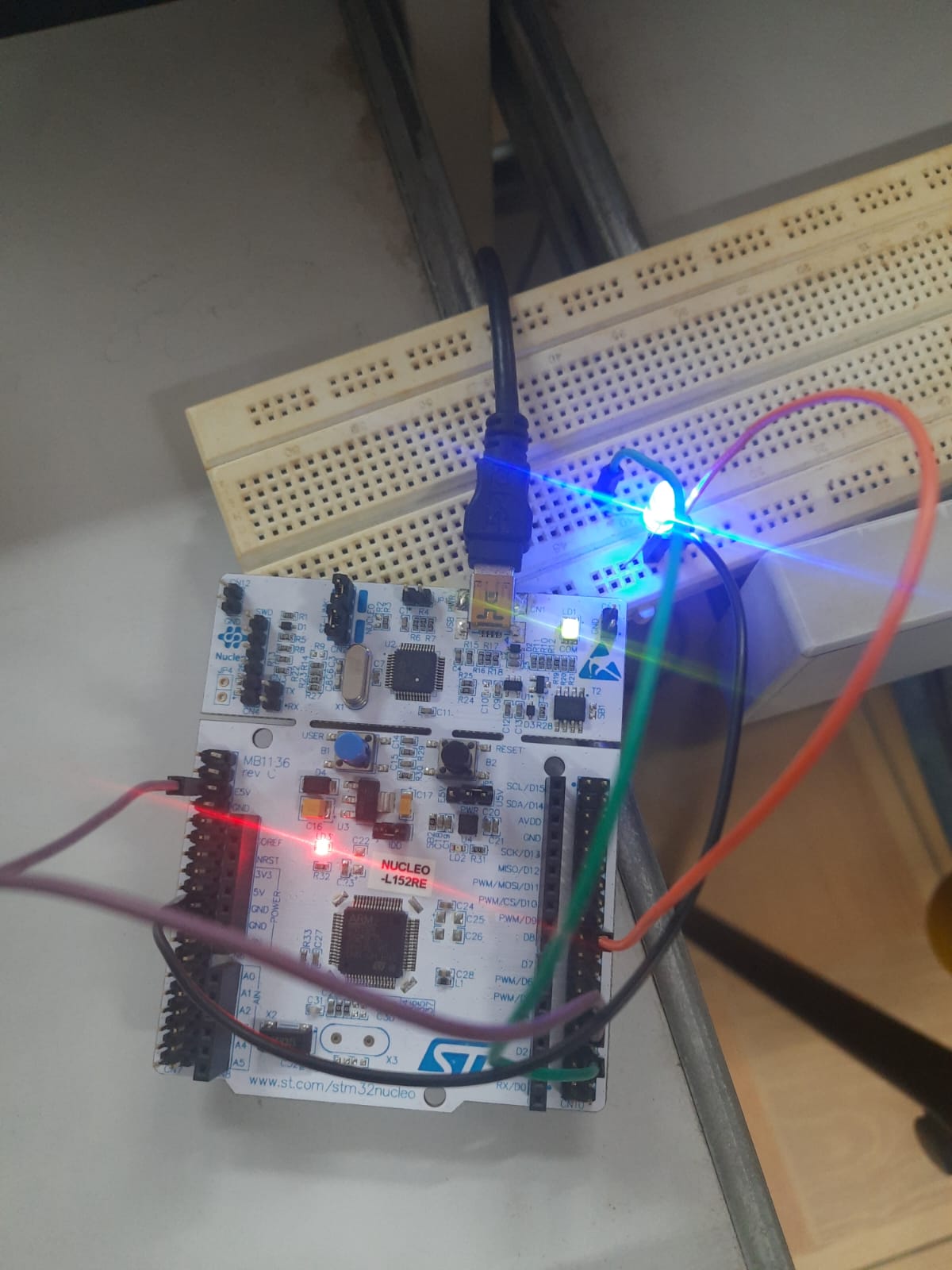
        pc.printf("LED OFF");

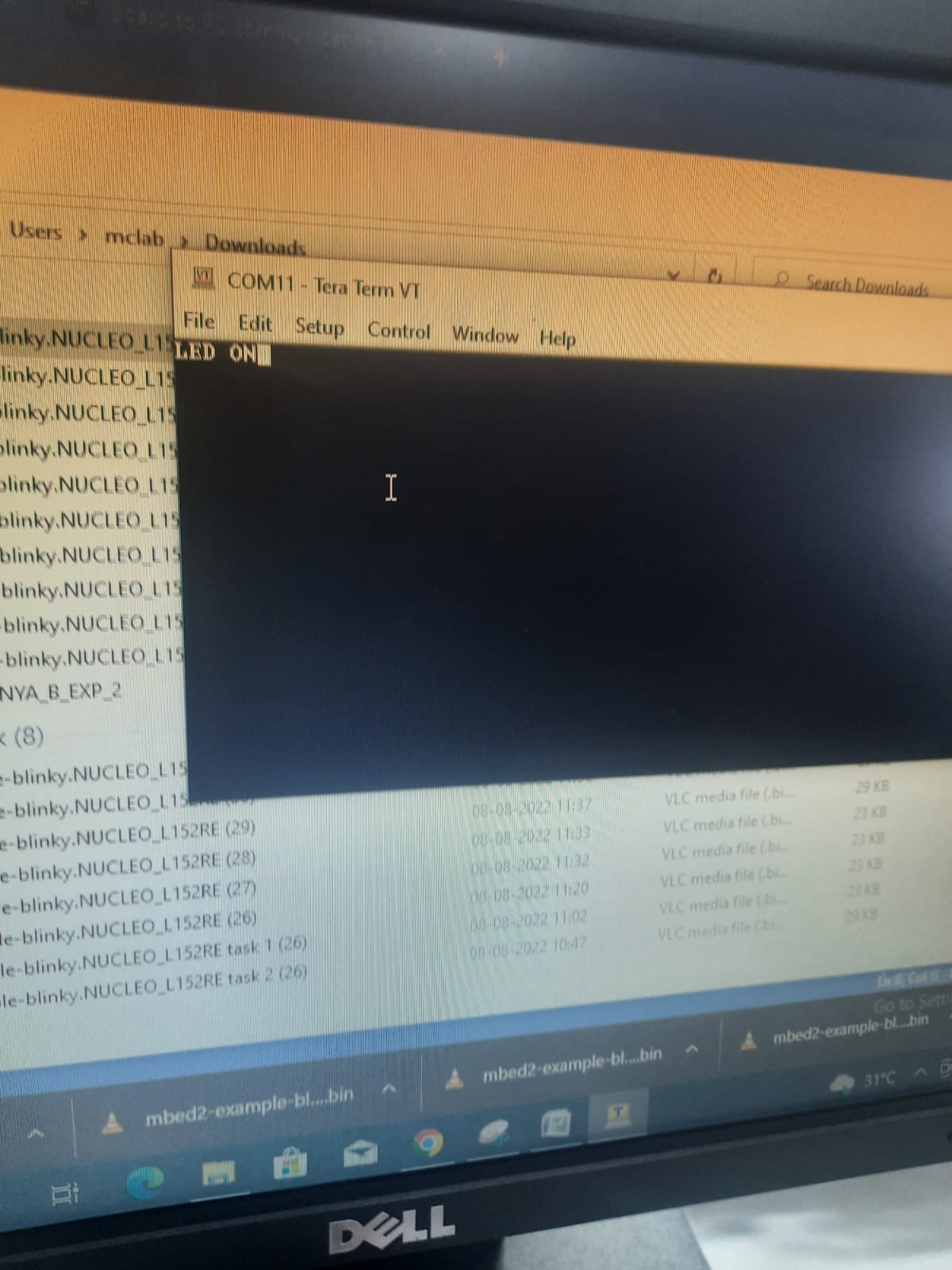
        wait(1);

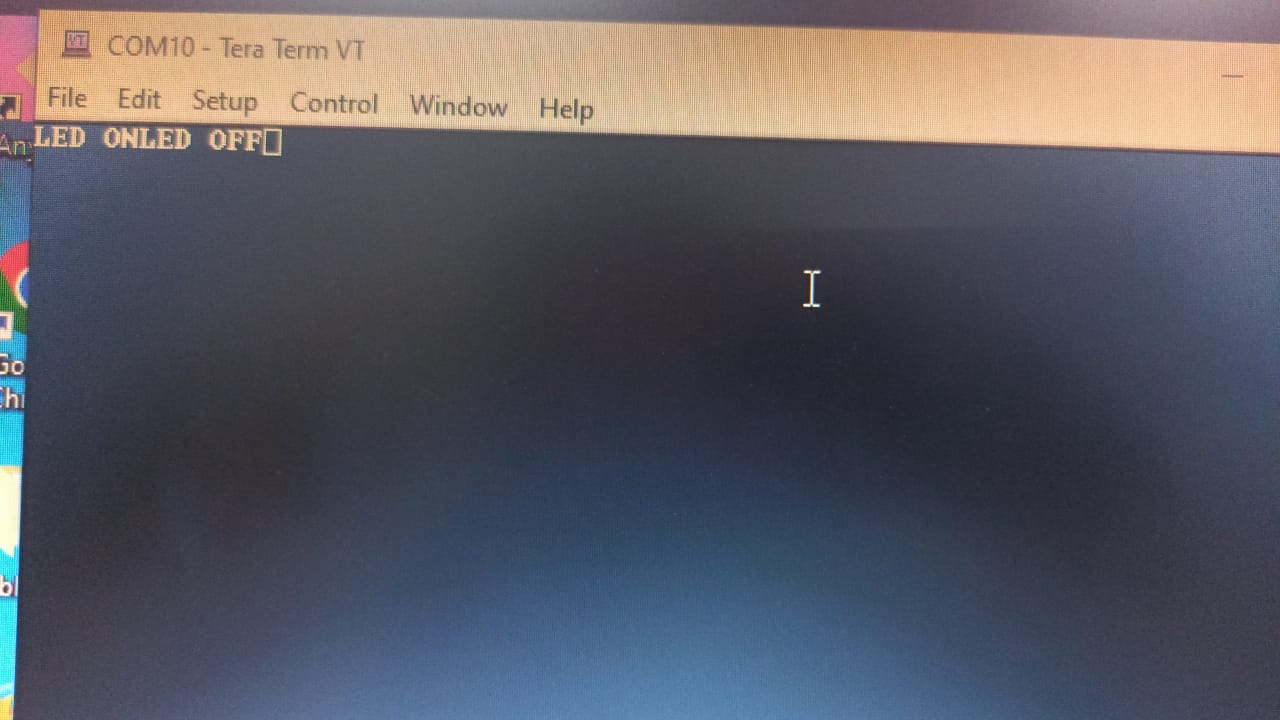
    }

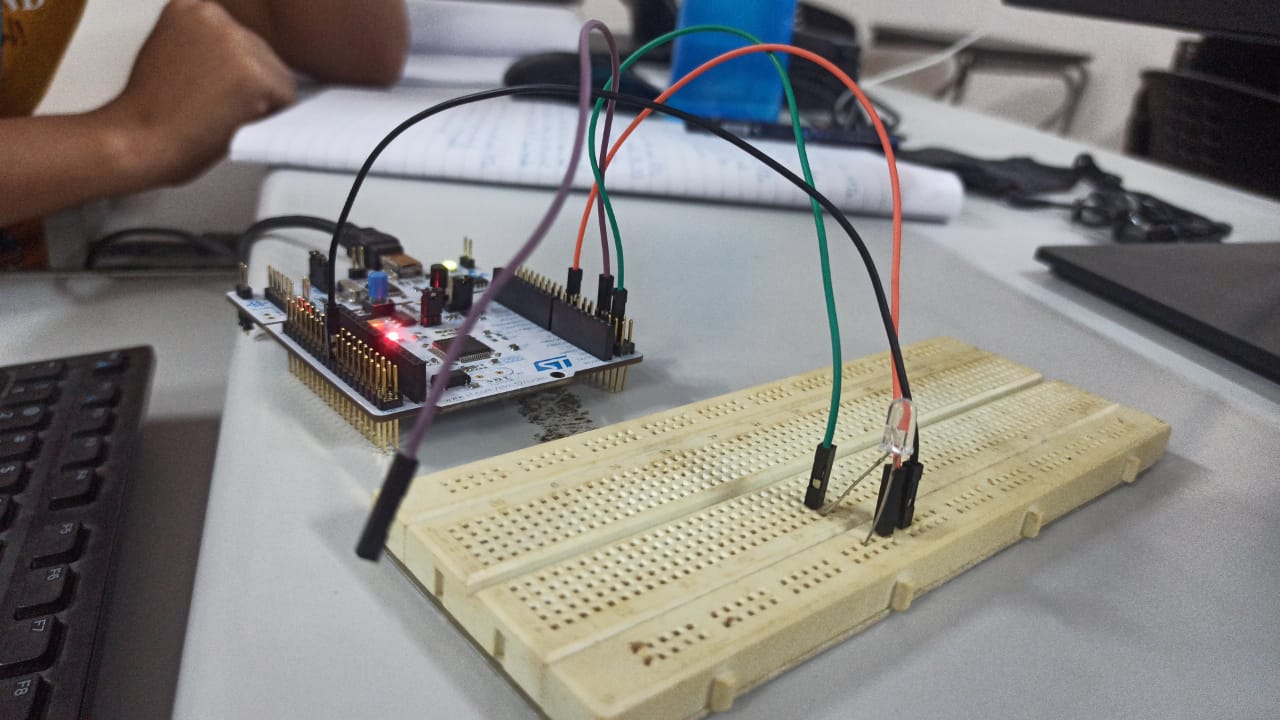
}

Output: -

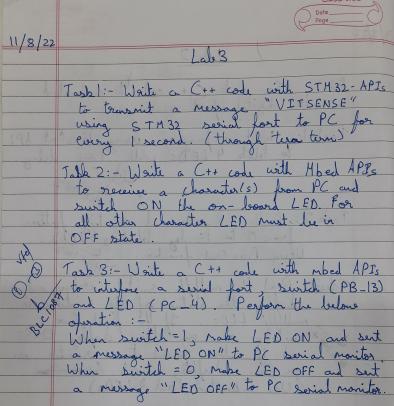








Verification Status



Result: - Therefore all the above tasks have been successfully understood, experimented and verified successfully.