VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN

LAB RECORD

Submitted By

20BLC1027 – SHYAAM S

Submitted To

Dr. SOFANA REKA S

LAB – 01

Embedded System Design

Experiment-1

Name- Shyaam S

Aim-Glowing LEDs in different patterns on a NUCLEO-L152RE Board using the ARM mbed Compiler(Online)

Software Used-

Arm MBED online Compiler

Hardware Used-

Micro USB cable, NUCLEO-L152RE Development Board, Bread board

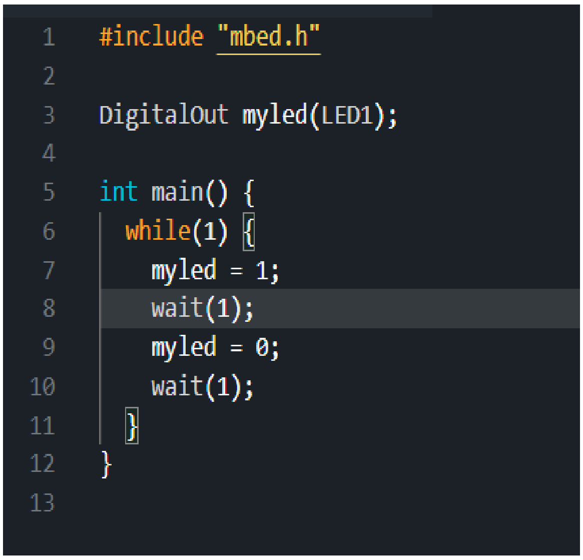
Procedure

1. Log-in into the mbed website by creating an account
2. Set-up the NUCLEO-L152RE Board.
3. Create A New-Project with template either Blinky LED Hello World or a Blank Template.
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the Board
8. Press Reset Button on the Board.
9. The code will now start its Execution.

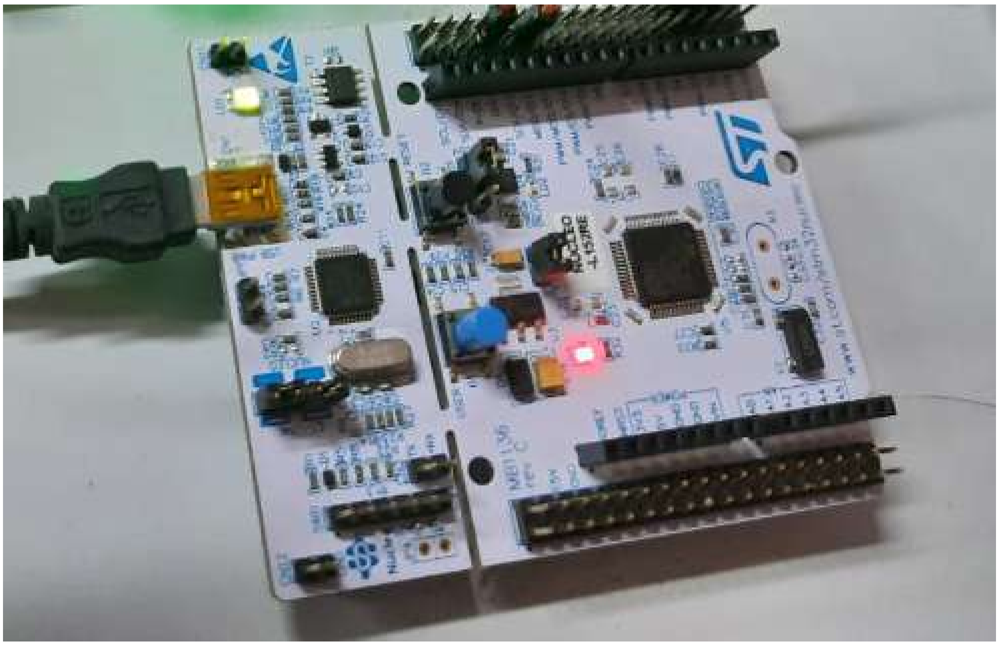
SHYAAM S 20BLC1027 Date 02/08/2022

Task-1:

Blinking the Primary Led on the Board

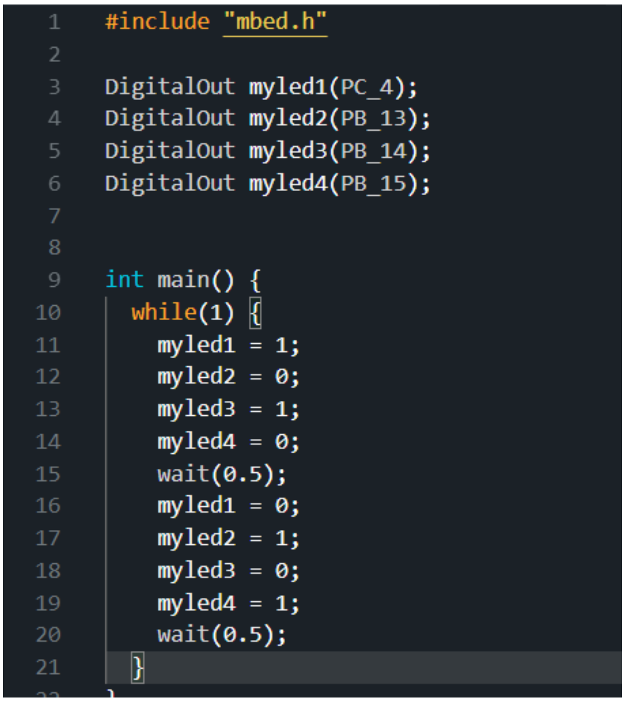


```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
  while(1) {
    myled = 1;
    wait(1);
    myled = 0;
    wait(1);
  }
}

```

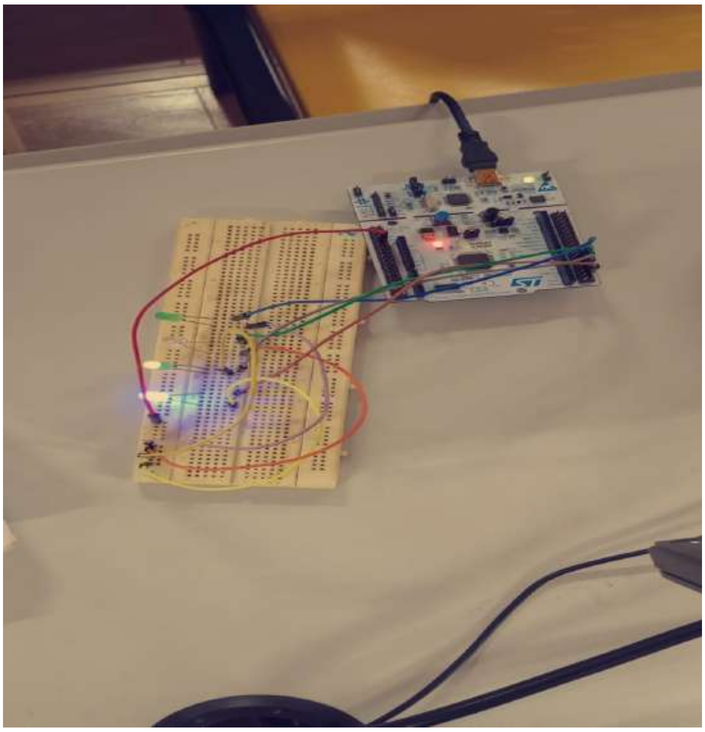


SHYAAM S 20BLC1027 Date 02/08/2022

Blinking LEDs Alternatively in pairs

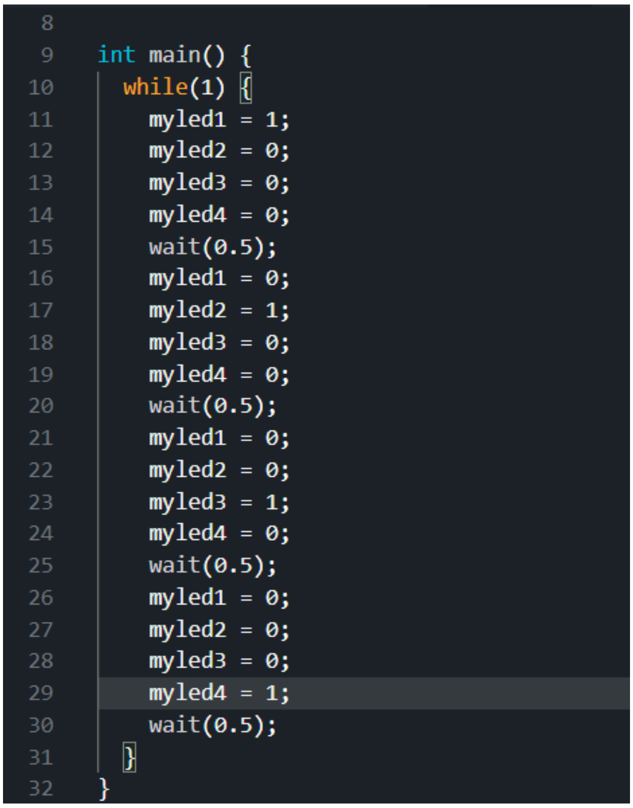


```
#include "mbed.h"

DigitalOut myled1(PC_4);
DigitalOut myled2(PB_13);
DigitalOut myled3(PB_14);
DigitalOut myled4(PB_15);


int main() {
  while(1) {
    myled1 = 1;
    myled2 = 0;
    myled3 = 1;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 1;
    myled3 = 0;
    myled4 = 1;
    wait(0.5);
  }
}
```

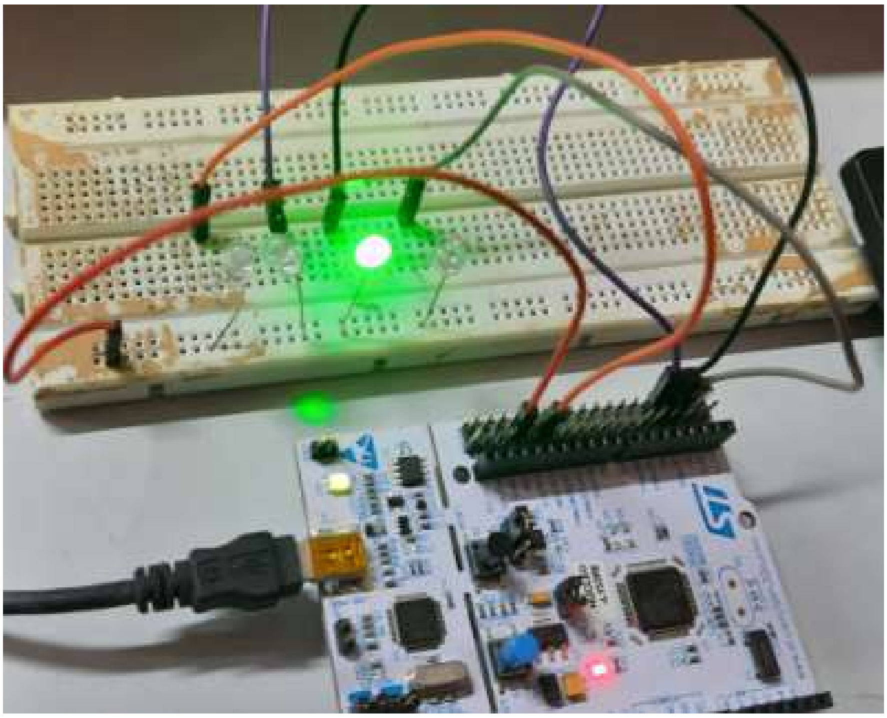


SHYAAM S 20BLC1027 Date 02/08/2022

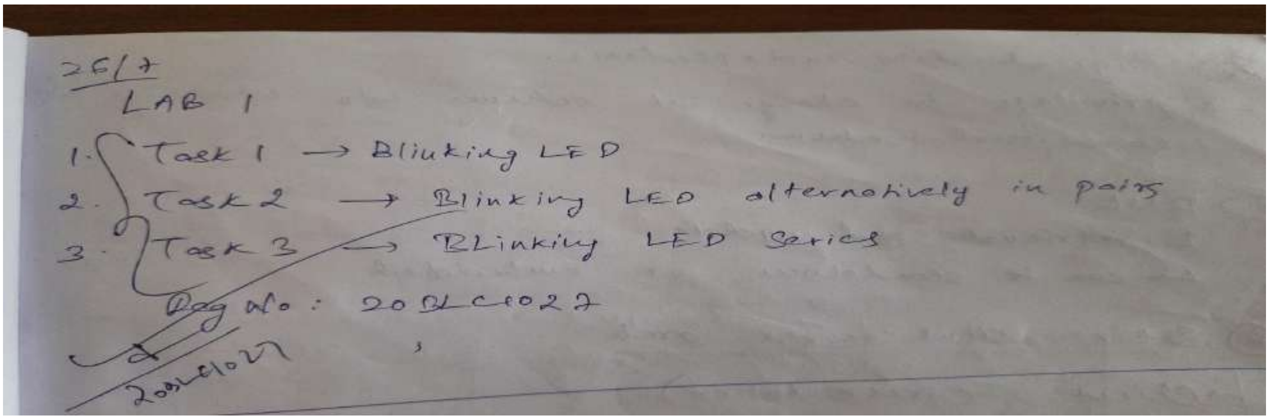
Blinking LED in series



```

int main() {
  while(1) {
    myled1 = 1;
    myled2 = 0;
    myled3 = 0;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 1;
    myled3 = 0;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 0;
    myled3 = 1;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 0;
    myled3 = 0;
    myled4 = 1;
    wait(0.5);
  }
}
```



Verification



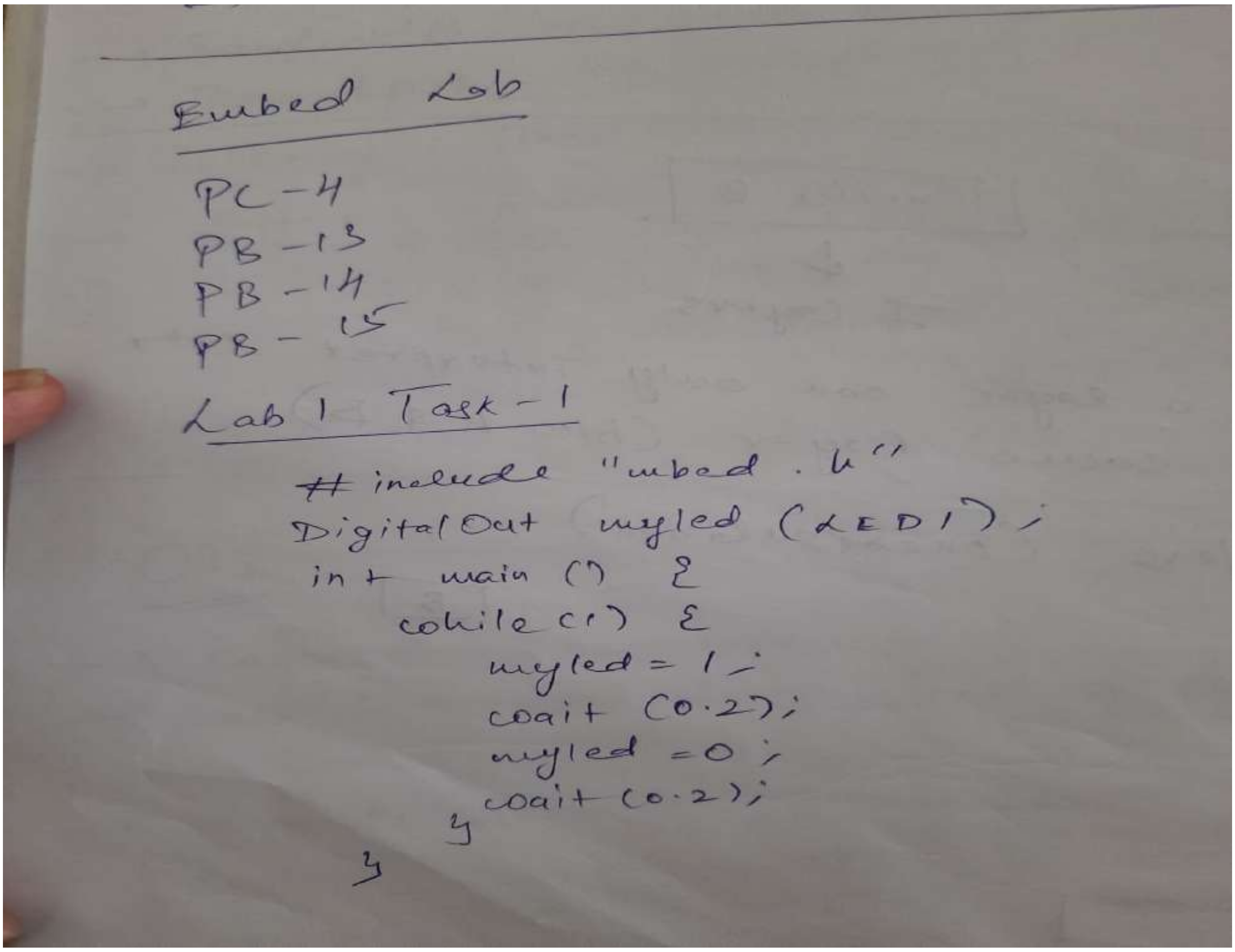
26/7

LAB 1

1. Task 1 → Blinking LED
2. Task 2 → Blinking LED alternatively in pairs
3. Task 3 → BLinking LED Series

Reg No : 20BLC1027

20BLC1027



Embed Lab

PC-4
PB-13
PB-14
PB-15

Lab 1 Task-1

```
#include "mbed.h"
DigitalOut myled (LED1);
int main () {
    while (1) {
        myled = 1;
        wait (0.2);
        myled =0;
        wait (0.2);
    }
}
```

Inference

1. Understood the specifications of the NUCLEO-L152RE Development Board and used the ARM online compiler to execute some basic

programs.

2. Understood the basic features of code like header files, Pin Declaration, delay(wait) etc.

Result

We were able to glow the LEDs connected to the development board in various patterns using the ARM online compiler .

SHYAAM S 20BLC1027 Date 02/08/2022

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN

LAB RECORD

Submitted By

20BLC1027 – SHYAAM S

Submitted To

Dr. SOFANA REKA S

LAB – 02

Embedded System Design

Experiment-2

Name- Shyaam S

Aim-Using the MBED BusIn and Busout APIs to perform different functions on the MBED board.

Software Used-

Arm MBED online Compiler

Hardware Used-

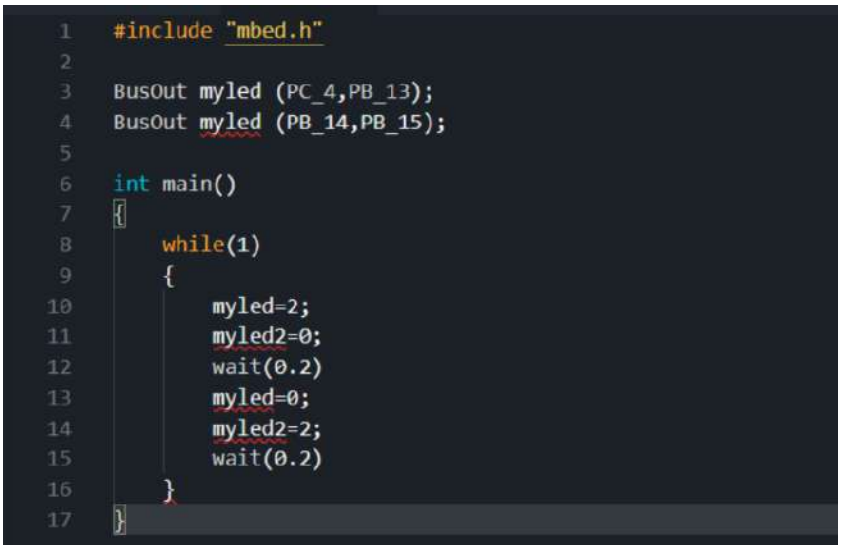
Micro USB cable, NUCLEO-L152RE Development Board, Bread board

Procedure

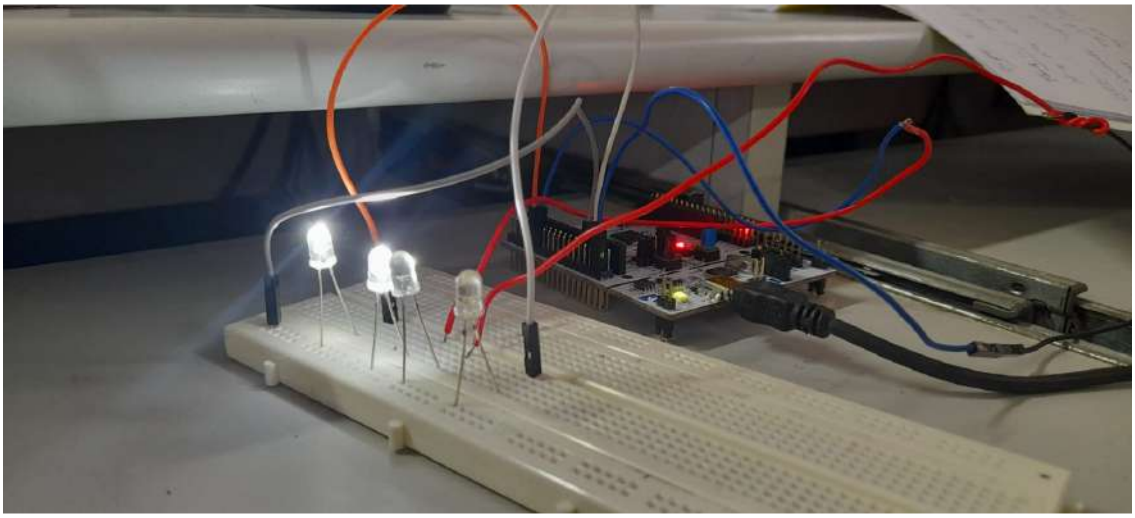
1. Log-in into the mbed website by creating an account
2. Set-up the NUCLEO-L152RE Board.
3. Create A New-Project with template either Blinky LED Hello World or a Blank Template.
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the Board
8. Press Reset Button on the Board.
9. The code will now start its Execution.

Task-1:

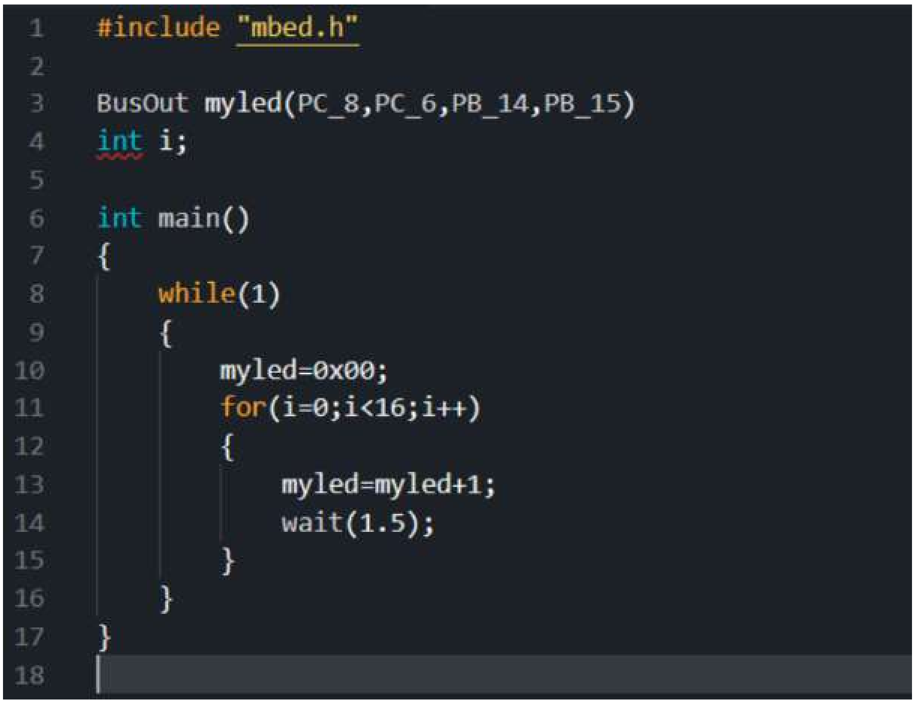
Simple Program to lit two-two LEDs at a time



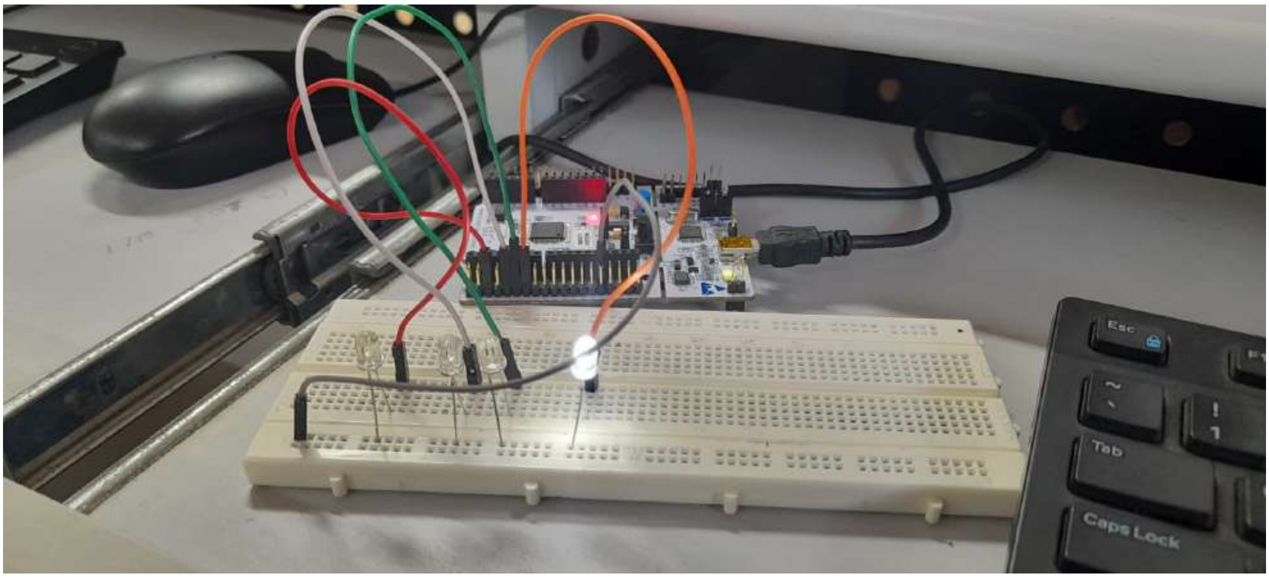
```
#include "mbed.h"

BusOut myled (PC_4,PB_13);
BusOut myled (PB_14,PB_15);

int main()
{
    while(1)
    {
        myled=2;
        myled2=0;
        wait(0.2)
        myled=0;
        myled2=2;
        wait(0.2)
    }
}
```

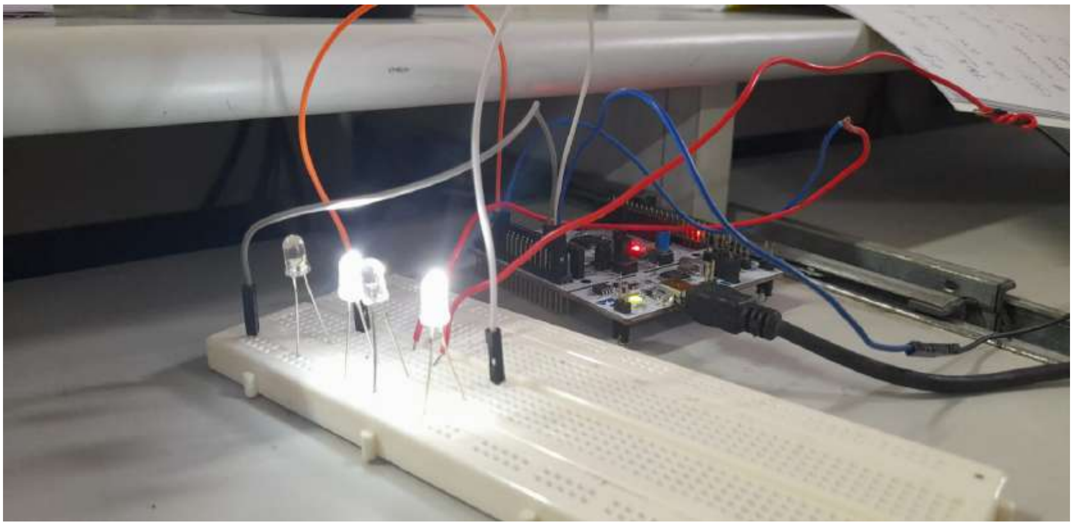


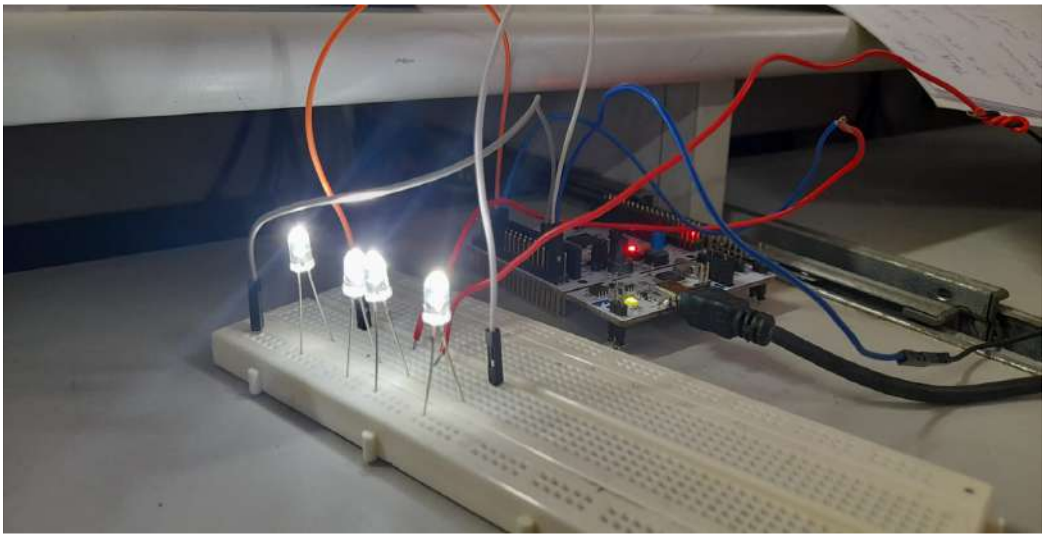
Task 2: Write a C++ code with code with mbed APIs to display hexadecimal counting patterns from 0-15 by blinking LEDs using BusOut function.



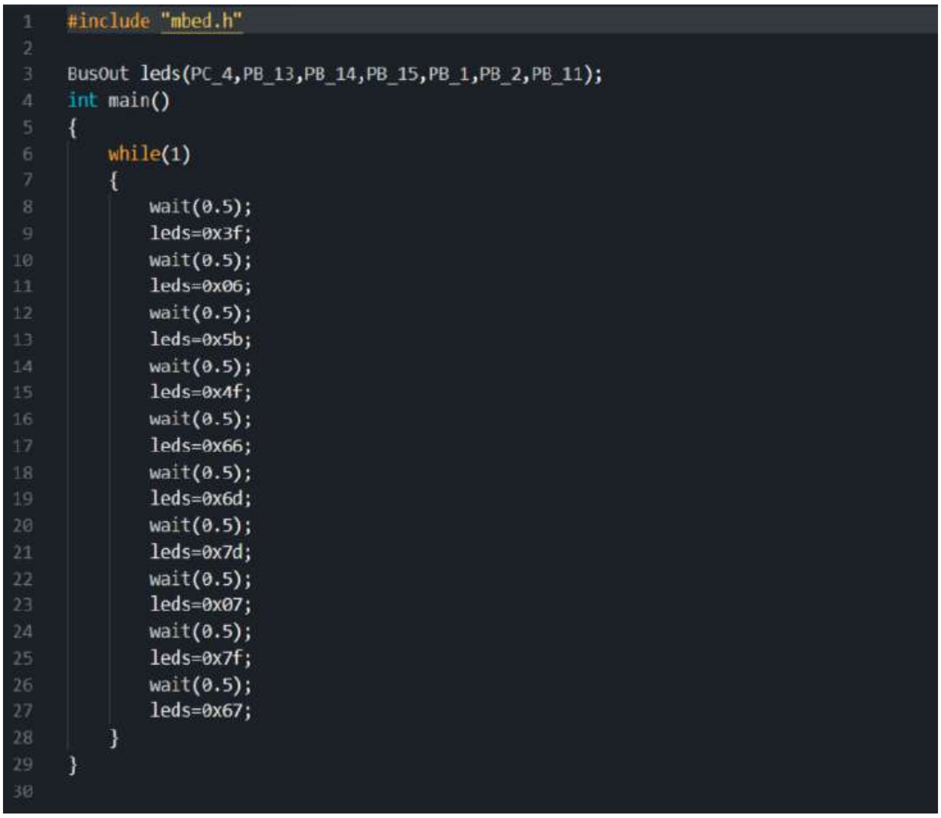
```
#include "mbed.h"

BusOut myled(PC_8,PC_6,PB_14,PB_15)
int i;

int main()
{
    while(1)
    {
        myled=0x00;
        for(i=0;i<16;i++)
        {
            myled=myled+1;
            wait(1.5);
        }
    }
}

```



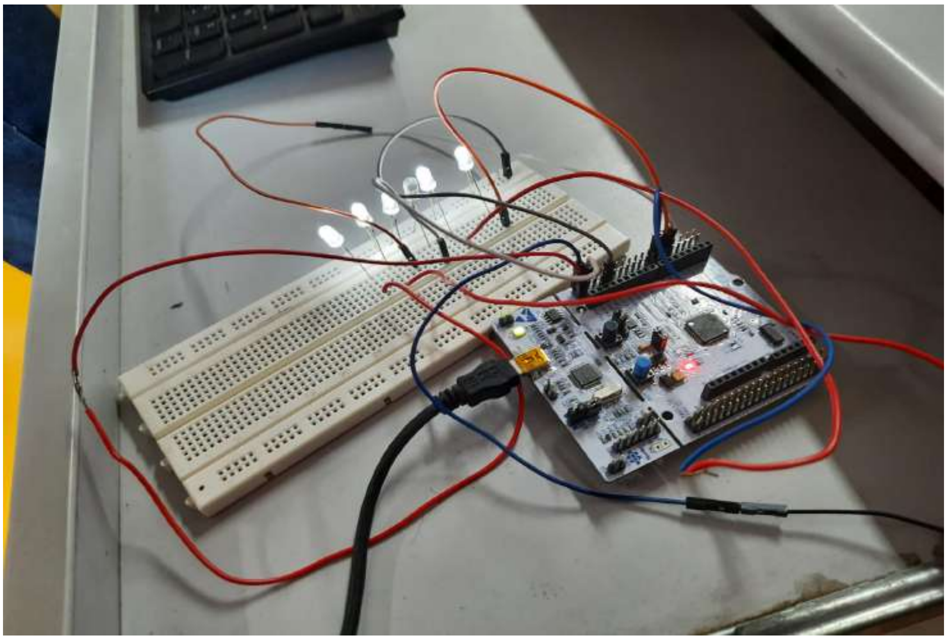


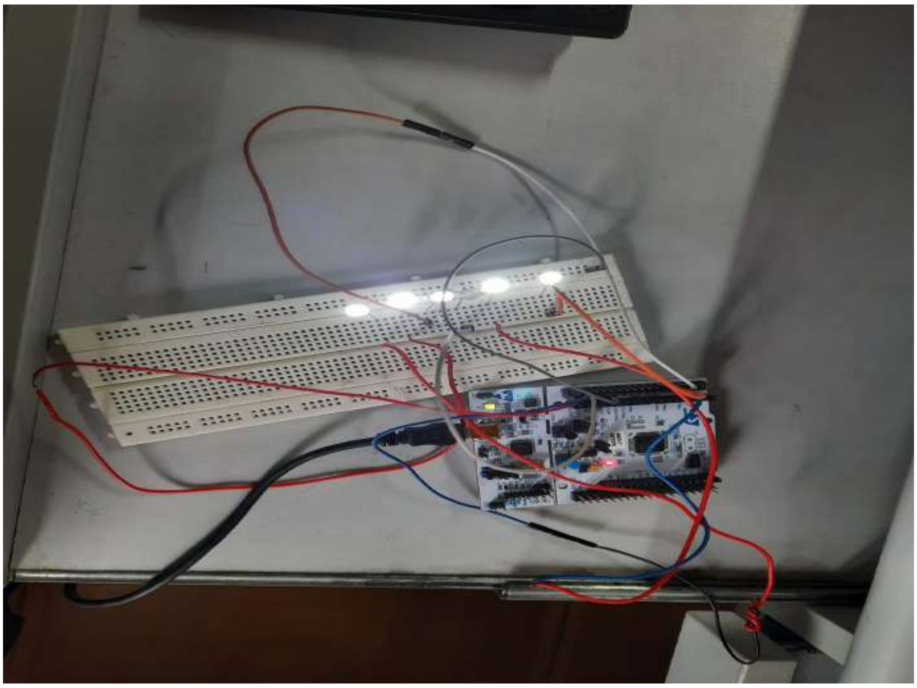


Task 3: Connecting a Seven-Segment Display and write a Code to display a continuous count from 0 to 9.

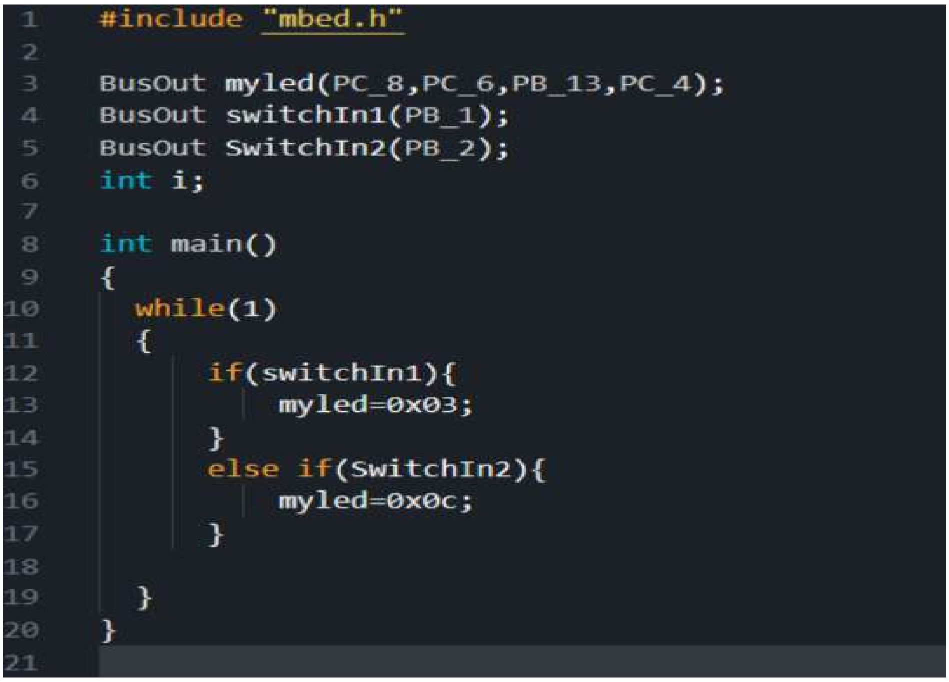


```
#include "mbed.h"

BusOut leds(PC_4,PB_13,PB_14,PB_15,PB_1,PB_2,PB_11);
int main()
{
    while(1)
    {
        wait(0.5);
        leds=0x3f;
        wait(0.5);
        leds=0x06;
        wait(0.5);
        leds=0x5b;
        wait(0.5);
        leds=0x4f;
        wait(0.5);
        leds=0x66;
        wait(0.5);
        leds=0x6d;
        wait(0.5);
        leds=0x7d;
        wait(0.5);
        leds=0x07;
        wait(0.5);
        leds=0x7f;
        wait(0.5);
        leds=0x67;
    }
}
```

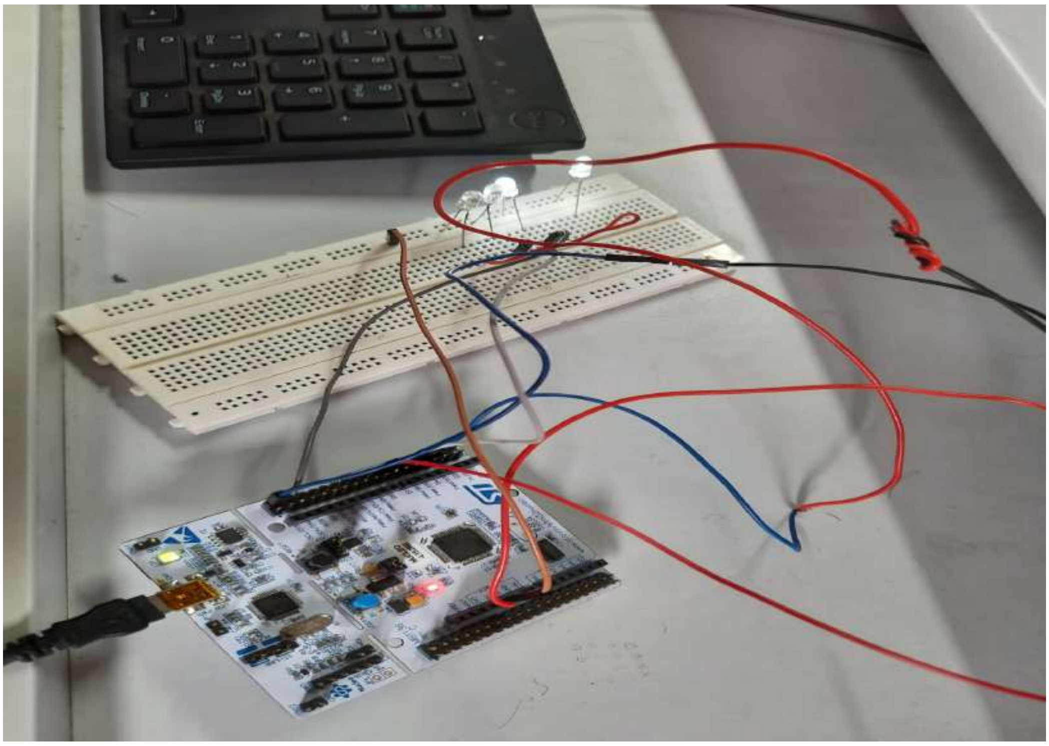


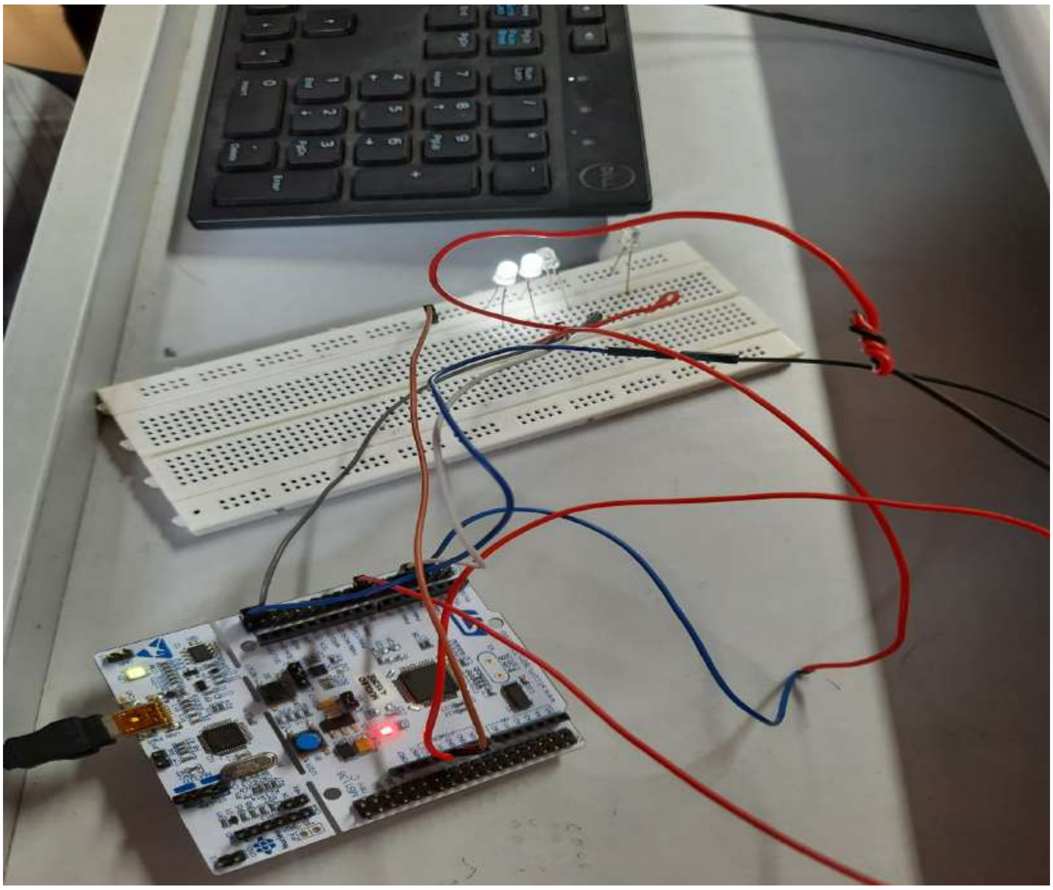


<u>CHALLENGING TASK</u>: **Read the multiple switch values and display the status of the switches in multiple LEDs. If the first switch is on then display 0x3 and if the second switch is on then display 0xC**

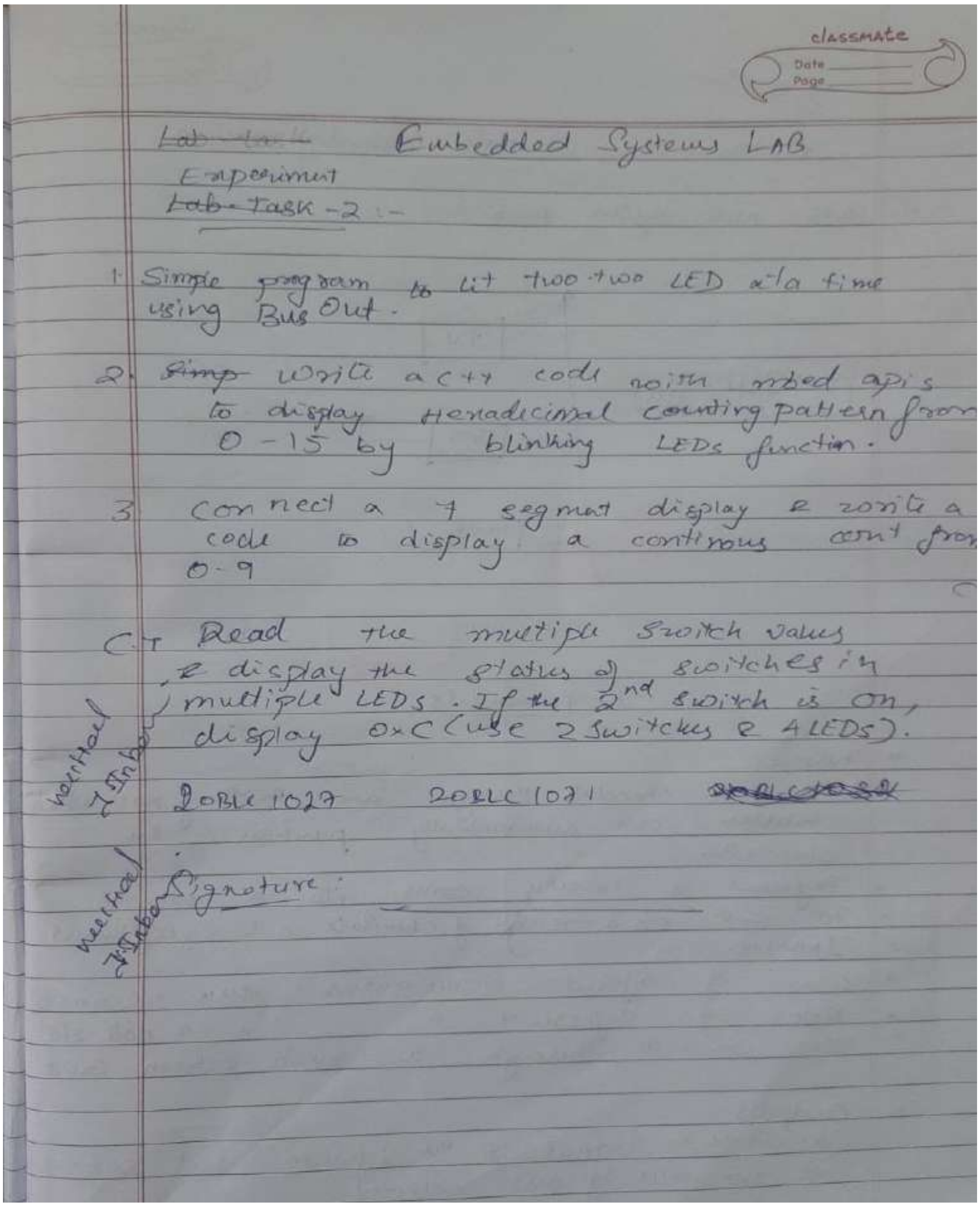


```
#include "mbed.h"

BusOut myled(PC_8,PC_6,PB_13,PC_4);
BusOut switchIn1(PB_1);
BusOut SwitchIn2(PB_2);
int i;

int main()
{
  while(1)
  {
      if(switchIn1){
         myled=0x03;
      }
      else if(SwitchIn2){
         myled=0x0c;
      }

  }
}
```





Verification



classmate
Date
Page

~~Lab task~~ Embedded Systems LAB
Experiment
~~Lab~~ Task -2 :-

1. Simple program to lit two-two LED at a time using Bus Out.

2. ~~Simp~~ Write a C++ code with mbed api's to display Hexadicimal counting pattern from 0 - 15 by blinking LEDs function.

3. Connect a 7 segment display & write a code to display a continous count from 0-9

C.T Read the multiple switch values & display the status of switches in multiple LEDs. If the 2nd switch is on, display 0xC (use 2 switches & 4 LEDs).

20BLC1027 20BLC1071

Signature:

Inference

1.We acquired knowledge of the fundamental elements like the protocols BusIn and BusOut.

2. Learnt about the basic functionalities of Nucleo64-STM32L152 board and explore various connections using BusIn and BusOut protocols.

Result

We used the BusIn and BusOut APIs to glow the LEDs in the requested pattern in the NUCLEO board.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B.Tech - Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN LAB RECORD

Submitted By

20BLC1027 – Shyaam S

Submitted To

Dr. Sofana Reka S

Experiment 3

16/08/2022

Working with Serial Ports and ADC

Aim: To perform operations using Serial Ports and ADC on a NUCLEO-L152RE board using MBED compiler online.

Software required:

Keil ARM MBED online compiler, Tera Term

Hardware required:

Micro USB cable, NUCLEO-L152RE dev board, breadboard, wires, LEDs, potentiometer, LM35 sensor.

Procedure:

1. Log in to the studio.keil.arm.com website after creating an account.

2. Set up the NUCLEO-L152RE board by connecting it to the system using the micro-USB cable.

3. Create a new project by going to File, selecting New, and then New MBED Project.

4. Select the "mbed2-example-blinky" template and then name the project.

5. Select the target hardware as Nucleo-L152RE board.

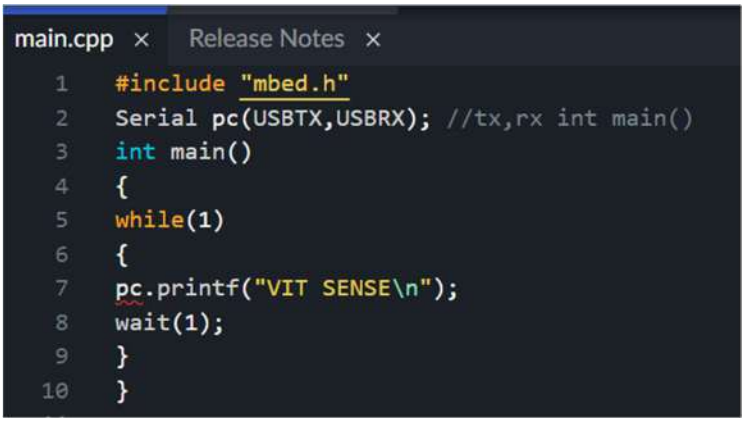
5. In the main.cpp file write/modify the code as per the requirement.

6. Click Build Project and download the bin file as prompted by the compiler.

7. After the bin file is downloaded, copy the file into the memory of the board. The code will now start its execution.

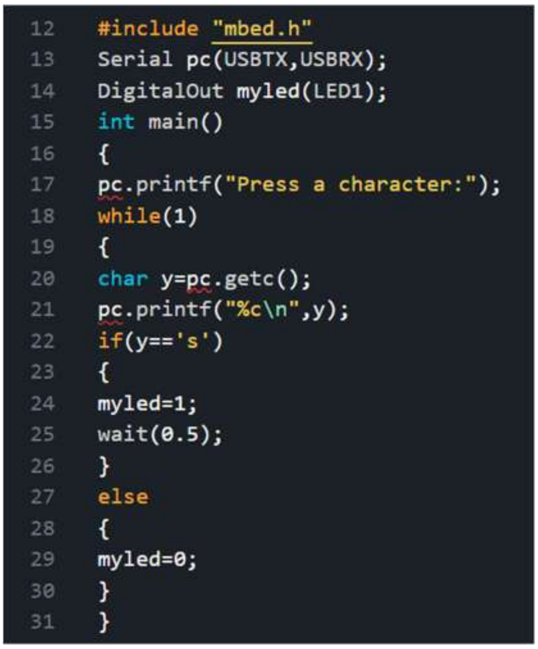
8. To use Tera Term, connect to COM port and establish new connection.

Task 1: Write a C++ code with MBED APIs to transmit a message "VIT SENSE" using MBED serial port to PC for every 0.5s.



```
#include "mbed.h"
Serial pc(USBTX,USBRX); //tx,rx int main()
int main()
{
while(1)
{
pc.printf("VIT SENSE\n");
wait(1);
}
}
```

Task 2: Write a C++ code with MBED APIs to receive a character (s) from the PC and switch ON the onboard LED1. For all other characters, LED1 must be in OFF state.

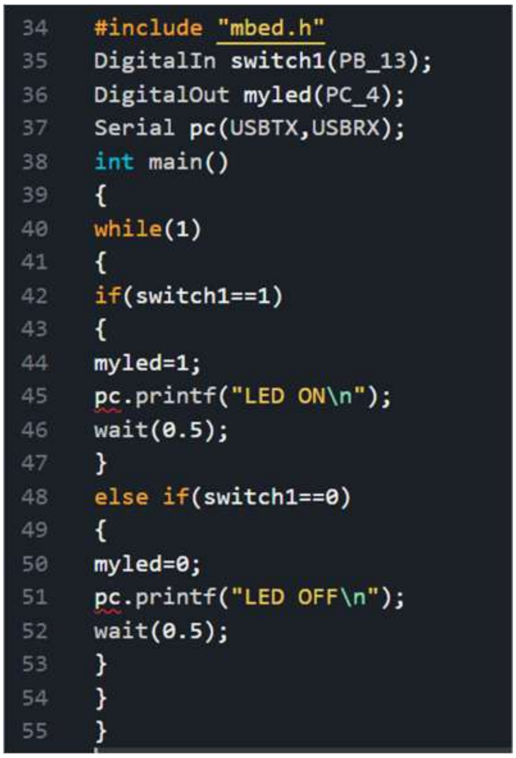


```
#include "mbed.h"
Serial pc(USBTX,USBRX);
DigitalOut myled(LED1);
int main()
{
pc.printf("Press a character:");
while(1)
{
char y=pc.getc();
pc.printf("%c\n",y);
if(y=='s')
{
myled=1;
wait(0.5);
}
else
{
myled=0;
}
}
```

Task 3: Write a C++ code with MBED APIs to interface a serial port, switch (PB_13) and LED (PC_4). Perform the operation below:

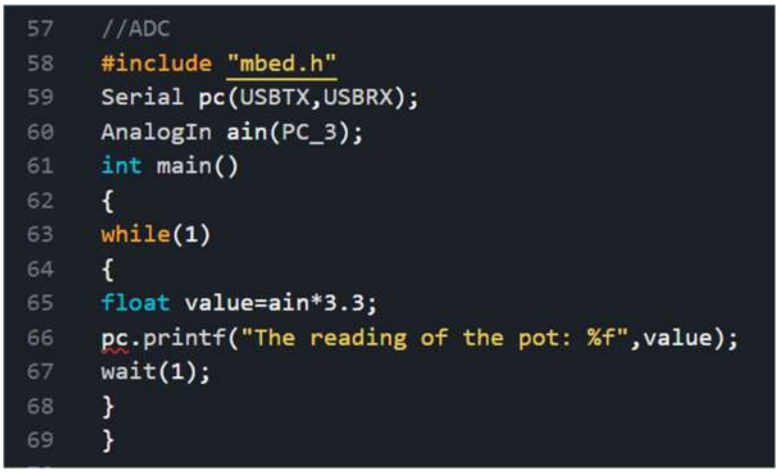
a) When switch = 1, make LED ON and send a message “LED ON” to PC serial monitor.

b) When switch = 0, make LED OFF and send a message “LED OFF” to PC serial monitor.



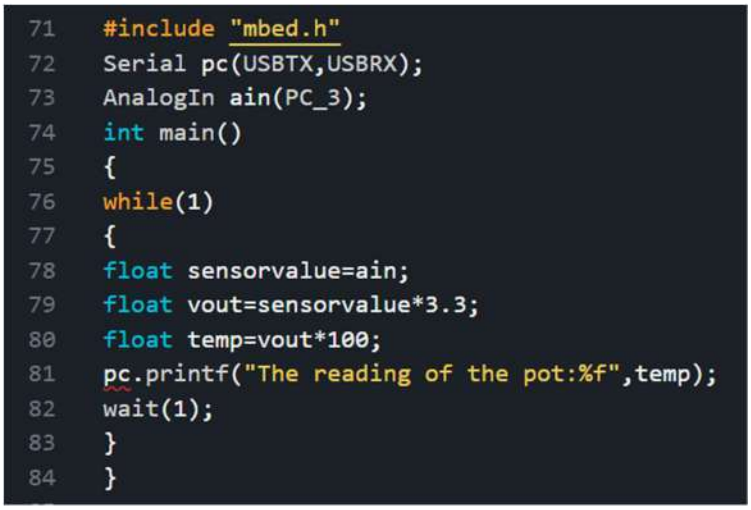
```
#include "mbed.h"
DigitalIn switch1(PB_13);
DigitalOut myled(PC_4);
Serial pc(USBTX,USBRX);
int main()
{
while(1)
{
if(switch1==1)
{
myled=1;
pc.printf("LED ON\n");
wait(0.5);
}
else if(switch1==0)
{
myled=0;
pc.printf("LED OFF\n");
wait(0.5);
}
}
}
```

Task 4: Write a C++ code with MBED APIs to print current voltage value of a potentiometer on PC serial monitor using ADC and serial port.



```
//ADC
#include "mbed.h"
Serial pc(USBTX,USBRX);
AnalogIn ain(PC_3);
int main()
{
while(1)
{
float value=ain*3.3;
pc.printf("The reading of the pot: %f",value);
wait(1);
}
}
```

Task 5: Write a C++ code with MBED APIs to interface a LM35 with MBED board and display current temperature value on PC serial monitor.

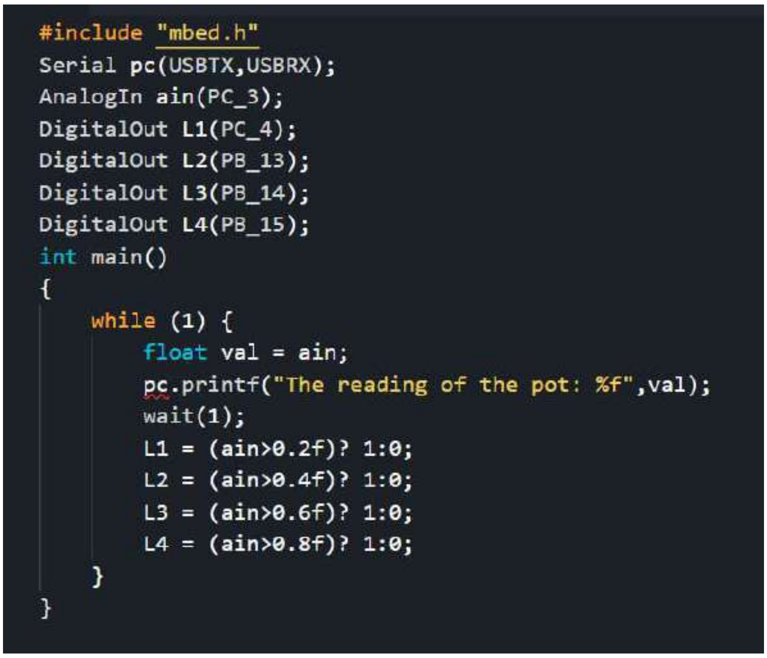


```
#include "mbed.h"
Serial pc(USBTX,USBRX);
AnalogIn ain(PC_3);
int main()
{
while(1)
{
float sensorvalue=ain;
float vout=sensorvalue*3.3;
float temp=vout*100;
pc.printf("The reading of the pot:%f",temp);
wait(1);
}
}
```

Challenging Task:

Write a C++ code with MBED APIs to implement a voltage level indicator using potentiometers and LEDs. Refer table below.

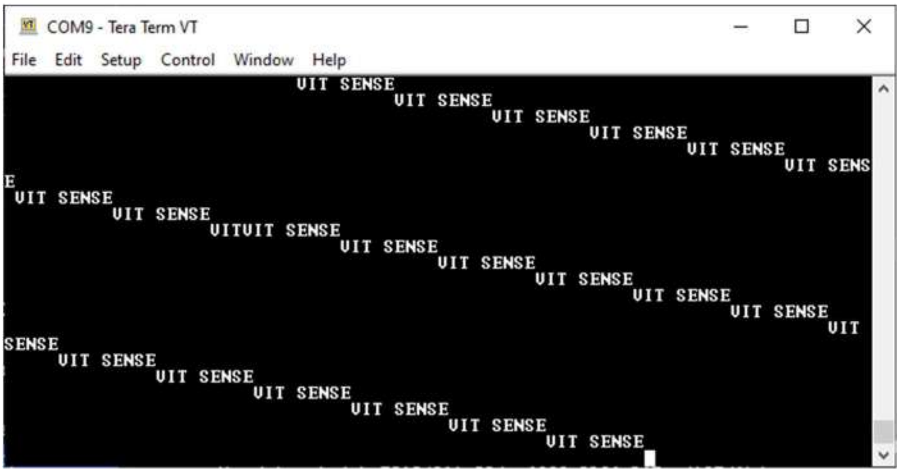
Analog Input value x	LED 1	LED 2	LED 3	LED 4
x<=0.2	0	0	0	0
0.2<x<=0.4	1	0	0	0
0.4<x<=0.6	1	1	0	0
0.6<x<=0.8	1	1	1	0
0.8<x<=1.0	1	1	1	1



```
#include "mbed.h"
Serial pc(USBTX,USBRX);
AnalogIn ain(PC_3);
DigitalOut L1(PC_4);
DigitalOut L2(PB_13);
DigitalOut L3(PB_14);
DigitalOut L4(PB_15);
int main()
{
    while (1) {
        float val = ain;
        pc.printf("The reading of the pot: %f",val);
        wait(1);
        L1 = (ain>0.2f)? 1:0;
        L2 = (ain>0.4f)? 1:0;
        L3 = (ain>0.6f)? 1:0;
        L4 = (ain>0.8f)? 1:0;
    }
}
```

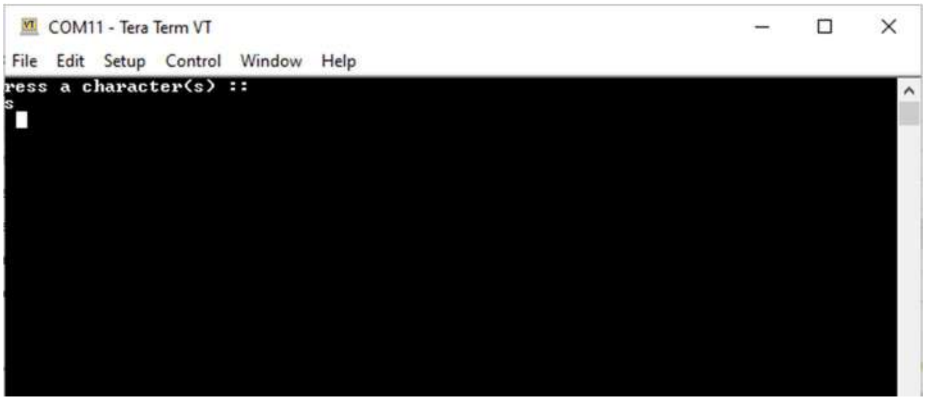
Output:

1.

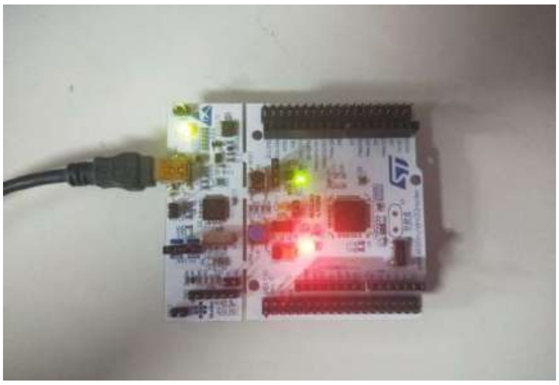


```
                           UIT SENSE
                                    UIT SENSE
                                             UIT SENSE
                                                      UIT SENSE
                                                               UIT SENSE
                                                                        UIT SENS
E
 UIT SENSE
          UIT SENSE
                   UITUIT SENSE
                               UIT SENSE
                                        UIT SENSE
                                                 UIT SENSE
                                                          UIT SENSE
                                                                   UIT SENSE
                                                                            UIT
SENSE
     UIT SENSE
              UIT SENSE
                       UIT SENSE
                                UIT SENSE
                                         UIT SENSE
                                                  UIT SENSE
```

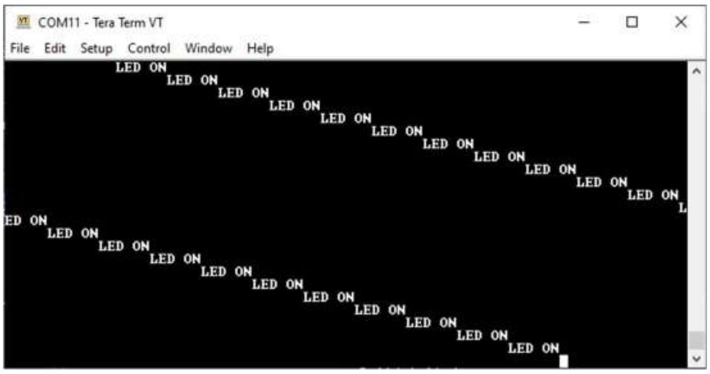
2.



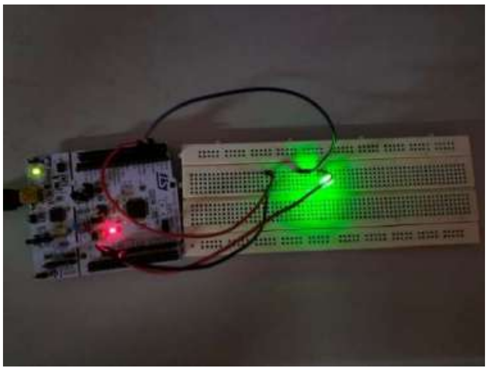
```
ress a character(s) ::
s
```

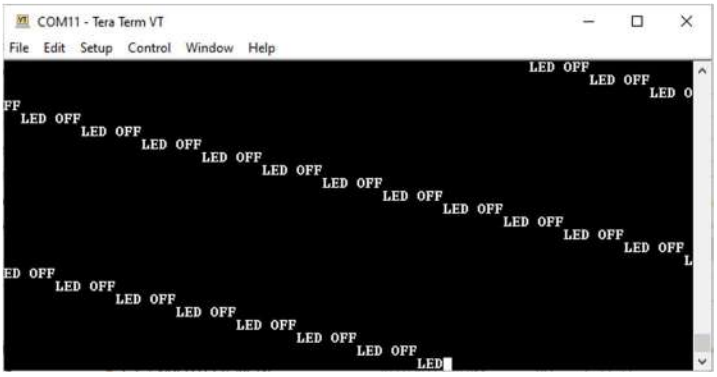


3.

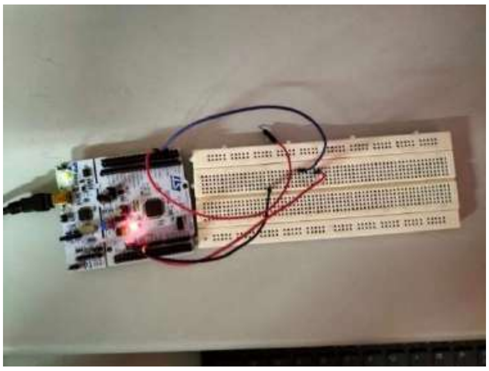


```
             LED ON
                   LED ON
                         LED ON
                               LED ON
                                     LED ON
                                           LED ON
                                                 LED ON
                                                       LED ON
                                                             LED ON
                                                                   LED ON
                                                                         LED ON L
ED ON
     LED ON
           LED ON
                 LED ON
                       LED ON
                             LED ON
                                   LED ON
                                         LED ON
                                               LED ON
                                                     LED ON
                                                           LED ON
```

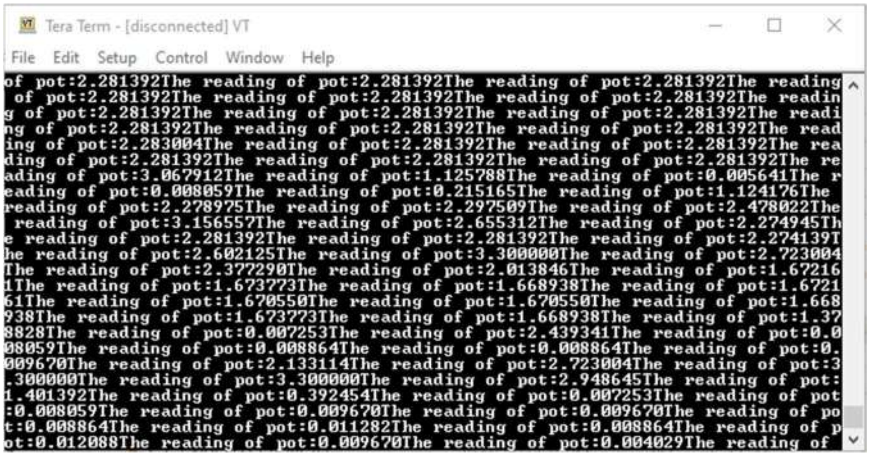




```
                                                             LED OFF
                                                                    LED OFF
                                                                           LED O
FF
  LED OFF
         LED OFF
                LED OFF
                       LED OFF
                              LED OFF
                                     LED OFF
                                            LED OFF
                                                   LED OFF
                                                          LED OFF
                                                                 LED OFF
                                                                        LED OFF
                                                                               L
ED OFF
      LED OFF
             LED OFF
                    LED OFF
                           LED OFF
                                  LED OFF
                                         LED OFF
                                                LED
```

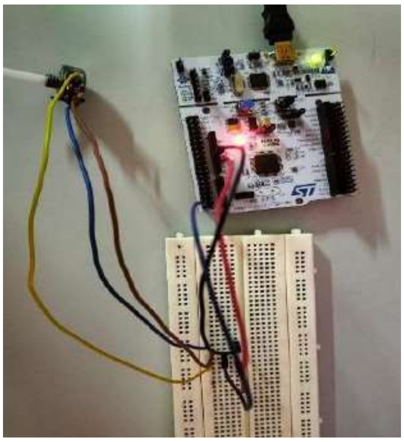


4.

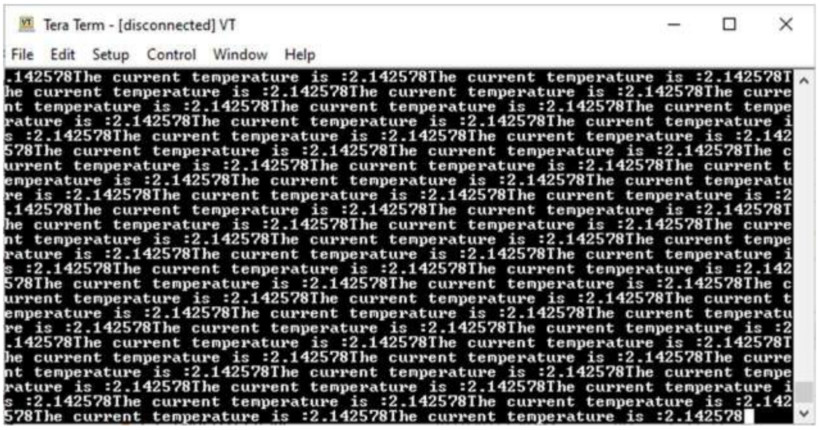


Tera Term - [disconnected] VT
File Edit Setup Control Window Help

```
of pot:2.281392The reading of pot:2.281392The reading of pot:2.281392The reading
 of pot:2.281392The reading of pot:2.281392The reading of pot:2.281392The readin
g of pot:2.281392The reading of pot:2.281392The reading of pot:2.281392The readi
ng of pot:2.281392The reading of pot:2.281392The reading of pot:2.281392The read
ing of pot:2.283004The reading of pot:2.281392The reading of pot:2.281392The rea
ding of pot:2.281392The reading of pot:2.281392The reading of pot:2.281392The re
ading of pot:3.067912The reading of pot:1.125788The reading of pot:0.005641The r
eading of pot:0.008059The reading of pot:0.215165The reading of pot:1.124176The
reading of pot:2.278975The reading of pot:2.297509The reading of pot:2.478022The
 reading of pot:3.156557The reading of pot:2.655312The reading of pot:2.274945Th
e reading of pot:2.281392The reading of pot:2.281392The reading of pot:2.274139T
he reading of pot:2.602125The reading of pot:3.300000The reading of pot:2.723004
The reading of pot:2.377290The reading of pot:2.013846The reading of pot:1.67216
1The reading of pot:1.673773The reading of pot:1.668938The reading of pot:1.6721
61The reading of pot:1.670550The reading of pot:1.670550The reading of pot:1.668
938The reading of pot:1.673773The reading of pot:1.668938The reading of pot:1.37
8828The reading of pot:0.007253The reading of pot:2.439341The reading of pot:0.0
08059The reading of pot:0.008864The reading of pot:0.008864The reading of pot:0.
009670The reading of pot:2.133114The reading of pot:2.723004The reading of pot:3
.300000The reading of pot:3.300000The reading of pot:2.948645The reading of pot:
1.401392The reading of pot:0.392454The reading of pot:0.007253The reading of pot
:0.008059The reading of pot:0.009670The reading of pot:0.009670The reading of po
t:0.008864The reading of pot:0.011282The reading of pot:0.008864The reading of p
ot:0.012088The reading of pot:0.009670The reading of pot:0.004029The reading of
```

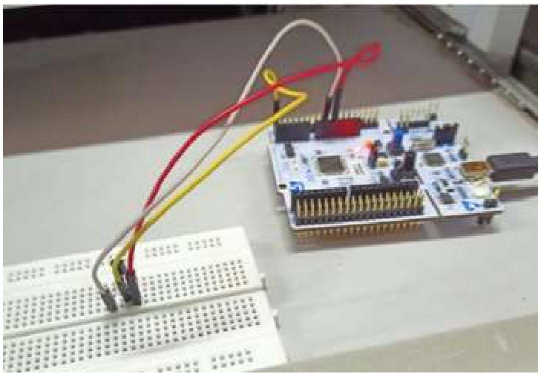


5.

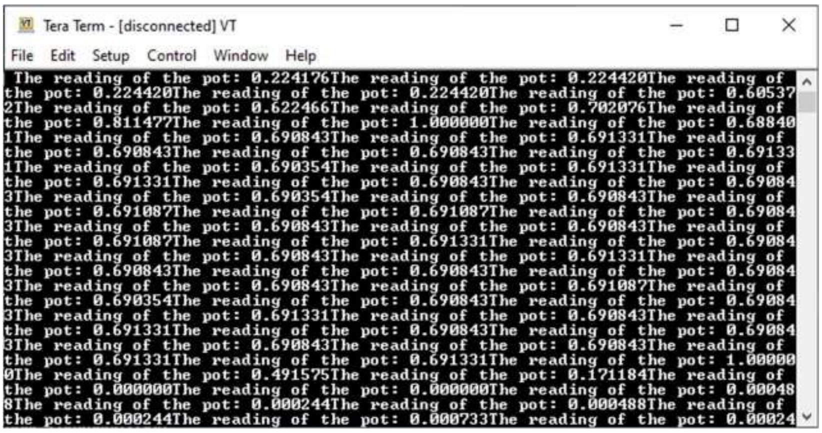


Tera Term - [disconnected] VT
File Edit Setup Control Window Help

```
.142578The current temperature is :2.142578The current temperature is :2.142578T
he current temperature is :2.142578The current temperature is :2.142578The curre
nt temperature is :2.142578The current temperature is :2.142578The current tempe
rature is :2.142578The current temperature is :2.142578The current temperature i
s :2.142578The current temperature is :2.142578The current temperature is :2.142
578The current temperature is :2.142578The current temperature is :2.142578The c
urrent temperature is :2.142578The current temperature is :2.142578The current t
emperature is :2.142578The current temperature is :2.142578The current temperatu
re is :2.142578The current temperature is :2.142578The current temperature is :2
.142578The current temperature is :2.142578The current temperature is :2.142578T
he current temperature is :2.142578The current temperature is :2.142578The curre
nt temperature is :2.142578The current temperature is :2.142578The current tempe
rature is :2.142578The current temperature is :2.142578The current temperature i
s :2.142578The current temperature is :2.142578The current temperature is :2.142
578The current temperature is :2.142578The current temperature is :2.142578The c
urrent temperature is :2.142578The current temperature is :2.142578The current t
emperature is :2.142578The current temperature is :2.142578The current temperatu
re is :2.142578The current temperature is :2.142578The current temperature is :2
.142578The current temperature is :2.142578The current temperature is :2.142578T
he current temperature is :2.142578The current temperature is :2.142578The curre
nt temperature is :2.142578The current temperature is :2.142578The current tempe
rature is :2.142578The current temperature is :2.142578The current temperature i
s :2.142578The current temperature is :2.142578The current temperature is :2.142
578The current temperature is :2.142578The current temperature is :2.142578
```

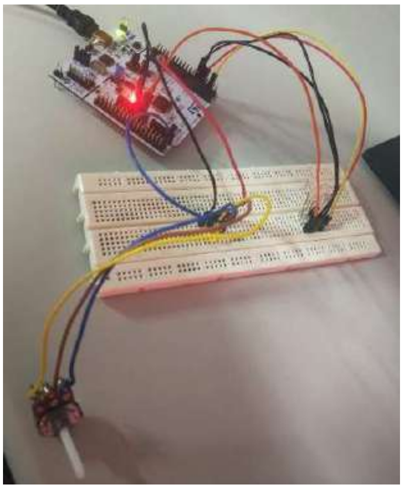


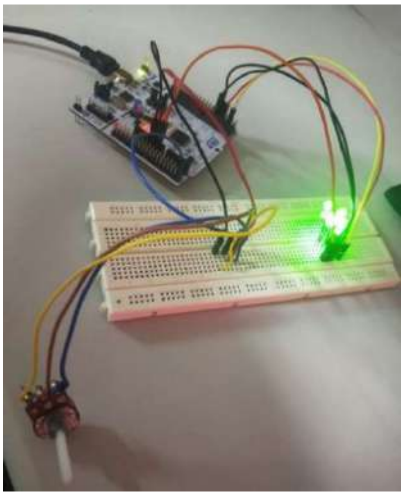
Challenging Task:



Tera Term - [disconnected] VT
File Edit Setup Control Window Help

```
 The reading of the pot: 0.224176The reading of the pot: 0.224420The reading of
the pot: 0.224420The reading of the pot: 0.224420The reading of the pot: 0.60537
2The reading of the pot: 0.622466The reading of the pot: 0.702076The reading of
the pot: 0.811477The reading of the pot: 1.000000The reading of the pot: 0.68840
1The reading of the pot: 0.690843The reading of the pot: 0.691331The reading of
the pot: 0.690843The reading of the pot: 0.690843The reading of the pot: 0.69133
1The reading of the pot: 0.690354The reading of the pot: 0.691331The reading of
the pot: 0.691331The reading of the pot: 0.690843The reading of the pot: 0.69084
3The reading of the pot: 0.690354The reading of the pot: 0.690843The reading of
the pot: 0.691087The reading of the pot: 0.691087The reading of the pot: 0.69084
3The reading of the pot: 0.690843The reading of the pot: 0.690843The reading of
the pot: 0.691087The reading of the pot: 0.691331The reading of the pot: 0.69084
3The reading of the pot: 0.690843The reading of the pot: 0.691331The reading of
the pot: 0.690843The reading of the pot: 0.690843The reading of the pot: 0.69084
3The reading of the pot: 0.690843The reading of the pot: 0.691087The reading of
the pot: 0.690354The reading of the pot: 0.690843The reading of the pot: 0.69084
3The reading of the pot: 0.691331The reading of the pot: 0.690843The reading of
the pot: 0.691331The reading of the pot: 0.690843The reading of the pot: 0.69084
3The reading of the pot: 0.690843The reading of the pot: 0.690843The reading of
the pot: 0.691331The reading of the pot: 0.691331The reading of the pot: 1.00000
0The reading of the pot: 0.491575The reading of the pot: 0.171184The reading of
the pot: 0.000000The reading of the pot: 0.000000The reading of the pot: 0.00048
8The reading of the pot: 0.000244The reading of the pot: 0.000488The reading of
the pot: 0.000244The reading of the pot: 0.000733The reading of the pot: 0.00024
```





Inference:

1. Learnt about working with serial ports and AnalogIn API.
2. Learnt how to work with Tera Term software.

Result:

Thus, several operations have been performed using Serial ports and ADC on a NUCLEO-L152RE board using MBED compiler online.

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B.Tech - Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN LAB RECORD

Submitted By

20BLC1027 – Shyaam S

Submitted To

Dr. Sofana Reka S

Experiment 4

20/08/2022

Virtual Lab Revision Exercise

Aim: To perform operations using the APIs learnt on a NUCLEO-L152RE board using MBED compiler online.

Software required:

Keil ARM MBED online compiler

Hardware required:

Micro USB cable, NUCLEO-L152RE dev board, breadboard, wires, LEDs

Procedure:

1. Log in to the studio.keil.arm.com website after creating an account.

2. Set up the NUCLEO-L152RE board by connecting it to the system using the micro-USB cable.

3. Create a new project by going to File, selecting New, and then New MBED Project.

4. Select the “mbed2-example-blinky” template and then name the project.

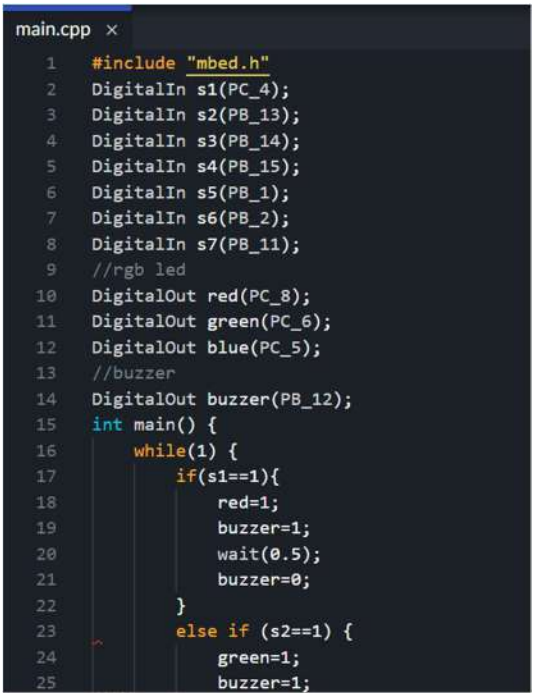
5. Select the target hardware as Nucleo-L152RE board.

5. In the main.cpp file write/modify the code as per the requirement.

6. Click Build Project and download the bin file as prompted by the compiler.

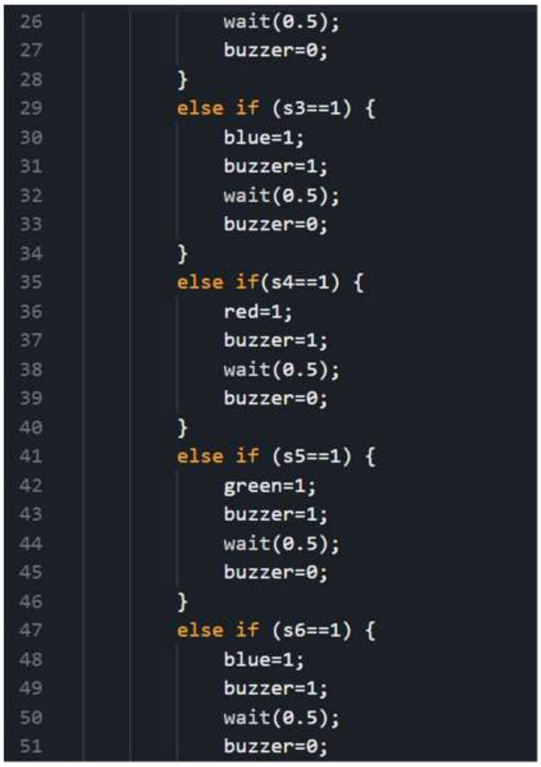
7. After the bin file is downloaded, copy the file into the memory of the board. The code will now start its execution.

Task 1: Write a program to design a piano that consists of 7 push button, 1 RGB LED, 1 Buzzer. This is to control two LEDs (Green, Red) state using a switch as per following logic. Each switch is configured to generate a music note SA-RE-GA-MA-PA-THA-NEE on the buzzer whenever respective switch is gets pressed (For switch-1 generate “SA”, for switch-2 generate “RE”, etc). Also assign a specific colour on RGB LED whenever respective key is press. Therefore, for every key press the buzzer generate a music tone and RGB LEB glow with particular colour.

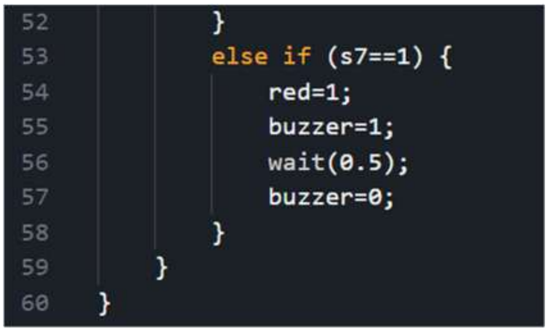


main.cpp ×

```cpp
#include "mbed.h"
DigitalIn s1(PC_4);
DigitalIn s2(PB_13);
DigitalIn s3(PB_14);
DigitalIn s4(PB_15);
DigitalIn s5(PB_1);
DigitalIn s6(PB_2);
DigitalIn s7(PB_11);
//rgb led
DigitalOut red(PC_8);
DigitalOut green(PC_6);
DigitalOut blue(PC_5);
//buzzer
DigitalOut buzzer(PB_12);
int main() {
    while(1) {
        if(s1==1){
            red=1;
            buzzer=1;
            wait(0.5);
            buzzer=0;
        }
        else if (s2==1) {
            green=1;
            buzzer=1;
```

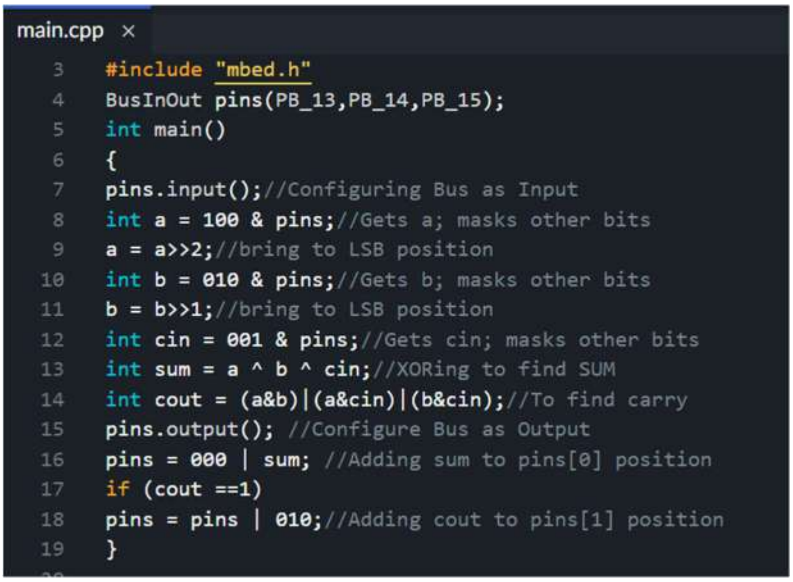


```cpp
            wait(0.5);
            buzzer=0;
        }
        else if (s3==1) {
            blue=1;
            buzzer=1;
            wait(0.5);
            buzzer=0;
        }
        else if(s4==1) {
            red=1;
            buzzer=1;
            wait(0.5);
            buzzer=0;
        }
        else if (s5==1) {
            green=1;
            buzzer=1;
            wait(0.5);
            buzzer=0;
        }
        else if (s6==1) {
            blue=1;
            buzzer=1;
            wait(0.5);
            buzzer=0;
```



```cpp
        }
        else if (s7==1) {
            red=1;
            buzzer=1;
            wait(0.5);
            buzzer=0;
        }
    }
}
```

Task 2: Develop A Full Adder Circuit: Based on the API studied: Bus INOUT



```
#include "mbed.h"
BusInOut pins(PB_13,PB_14,PB_15);
int main()
{
pins.input();//Configuring Bus as Input
int a = 100 & pins;//Gets a; masks other bits
a = a>>2;//bring to LSB position
int b = 010 & pins;//Gets b; masks other bits
b = b>>1;//bring to LSB position
int cin = 001 & pins;//Gets cin; masks other bits
int sum = a ^ b ^ cin;//XORing to find SUM
int cout = (a&b)|(a&cin)|(b&cin);//To find carry
pins.output(); //Configure Bus as Output
pins = 000 | sum; //Adding sum to pins[0] position
if (cout ==1)
pins = pins | 010;//Adding cout to pins[1] position
}
```

Inference:

1. Learnt to use the DigitalIn and DigitalOut APIs.
2. Learnt to develop a full adder circuit using BusInOut API.

Result:

Thus, the given tasks have been successfully implemented.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN

LAB RECORD

Submitted By

20BLC1027 – Shyaam S

Submitted To

Dr. SOFANA REKA S

LAB – 05

Shyaam S 20BLC1027 Date 23/08/2022

Embedded System Design

Experiment-5

Name – Shyaam S

Aim-Working with the MBED board PWM generation.

Software Used-

Arm MBED online Compiler

Hardware Used-

Micro USB cable, NUCLEO-L152RE Development Board, Bread board

Procedure

1. Log-in into the mbed website by creating an account
2. Set-up the NUCLEO-L152RE Board.
3. Create A New-Project with template either Blinky LED Hello World or a Blank Template.
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the Board
8. Press Reset Button on the Board.
9. The code will now start its Execution.

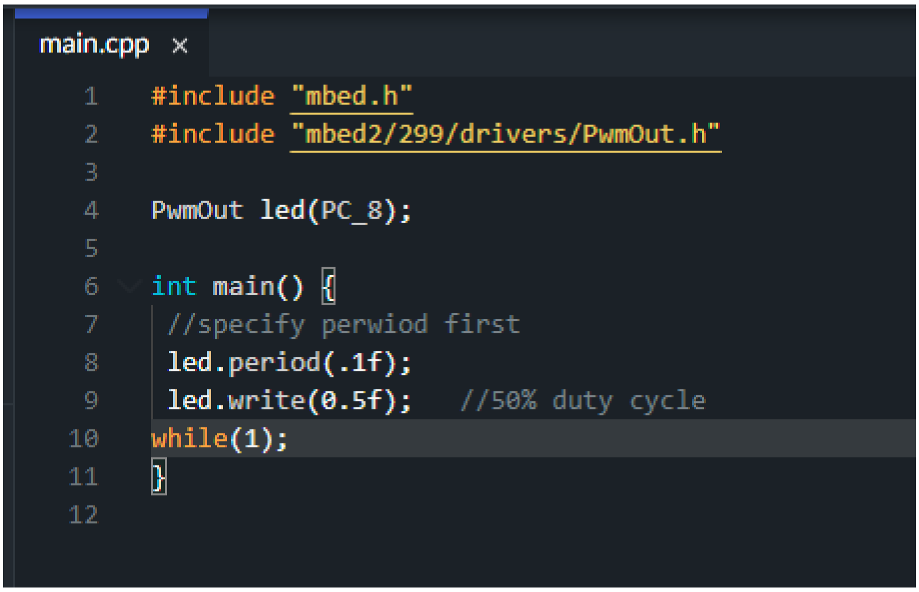
Shyaam S 20BLC1027 Date 23/08/2022

Task-1:

Write a C++ code with mbd APIs to generate a PWM signal with duty cycle 50%.verify the output using

(i)LED

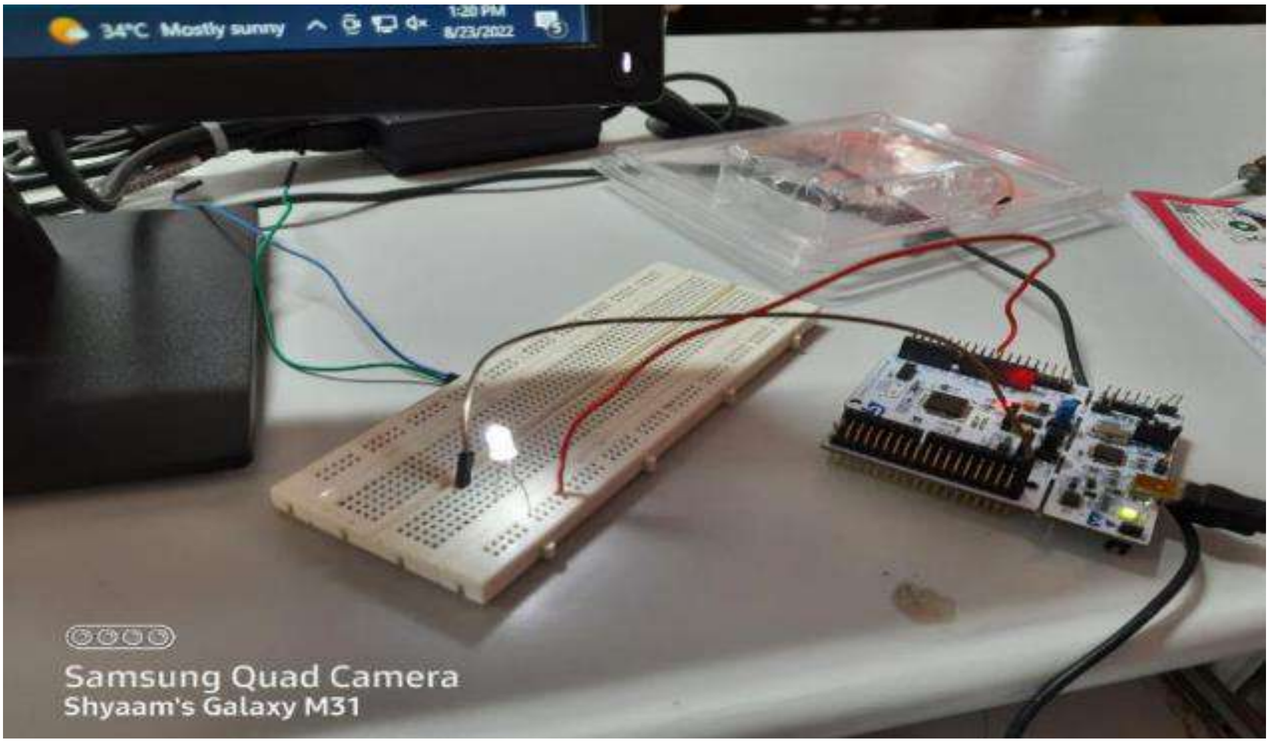
(ii)Digital Ocilloscope

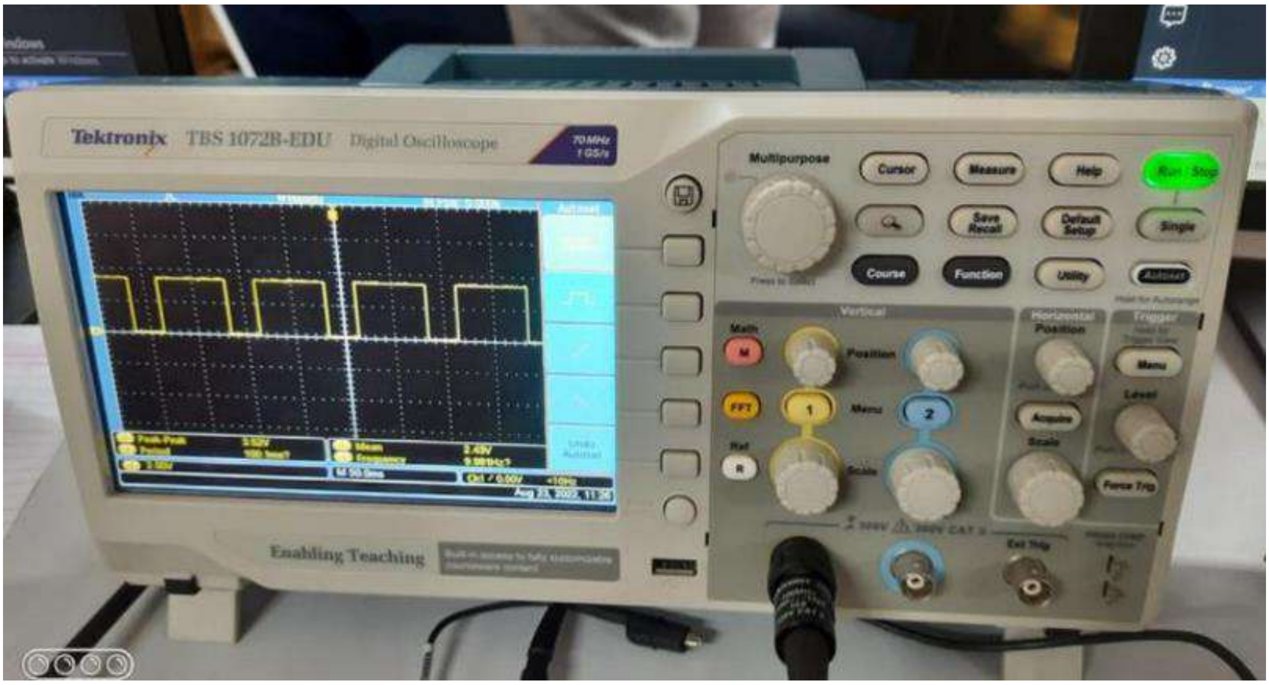


```
#include "mbed.h"
#include "mbed2/299/drivers/PwmOut.h"

PwmOut led(PC_8);

int main() {
 //specify perwiod first
 led.period(.1f);
 led.write(0.5f);   //50% duty cycle
while(1);
}

```

Shyaam S 20BLC1027 Date 23/08/2022

OUTPUT:





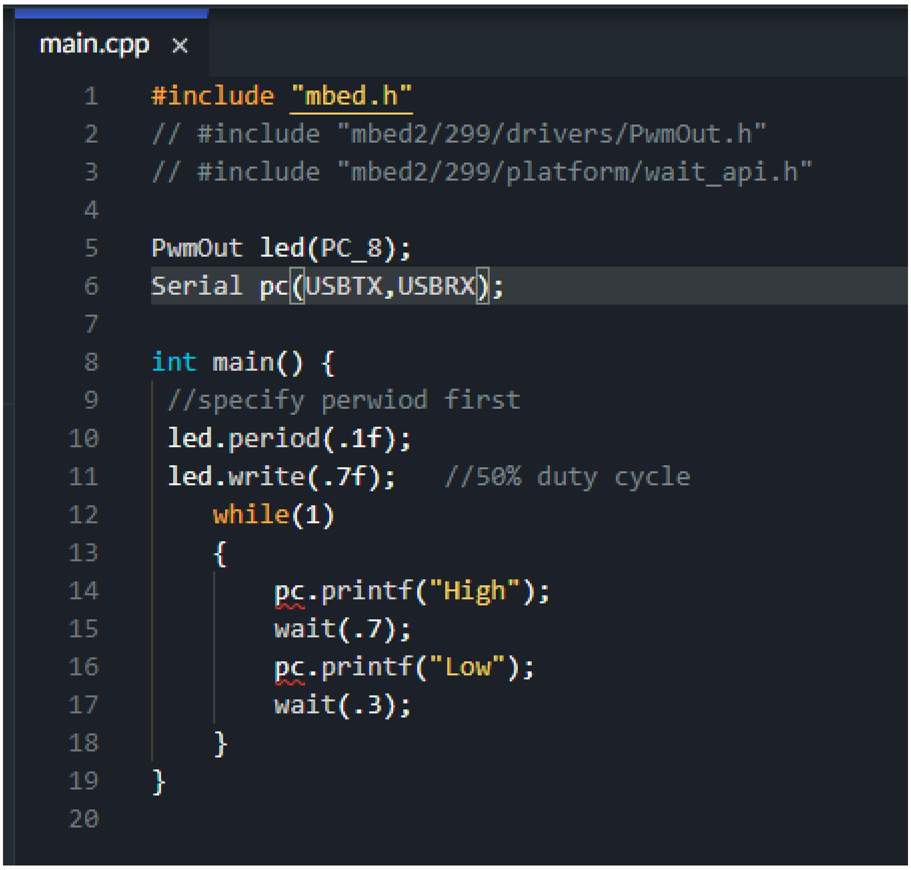
Shyaam S 20BLC1027 Date 23/08/2022

Task 2: Write a C++ code with code with mbed APIs to generate a PWM signal with duty cycle 70%.Verify the output using

(i)LED

(ii)Digital Ocilloscope

(iii) PC monitor

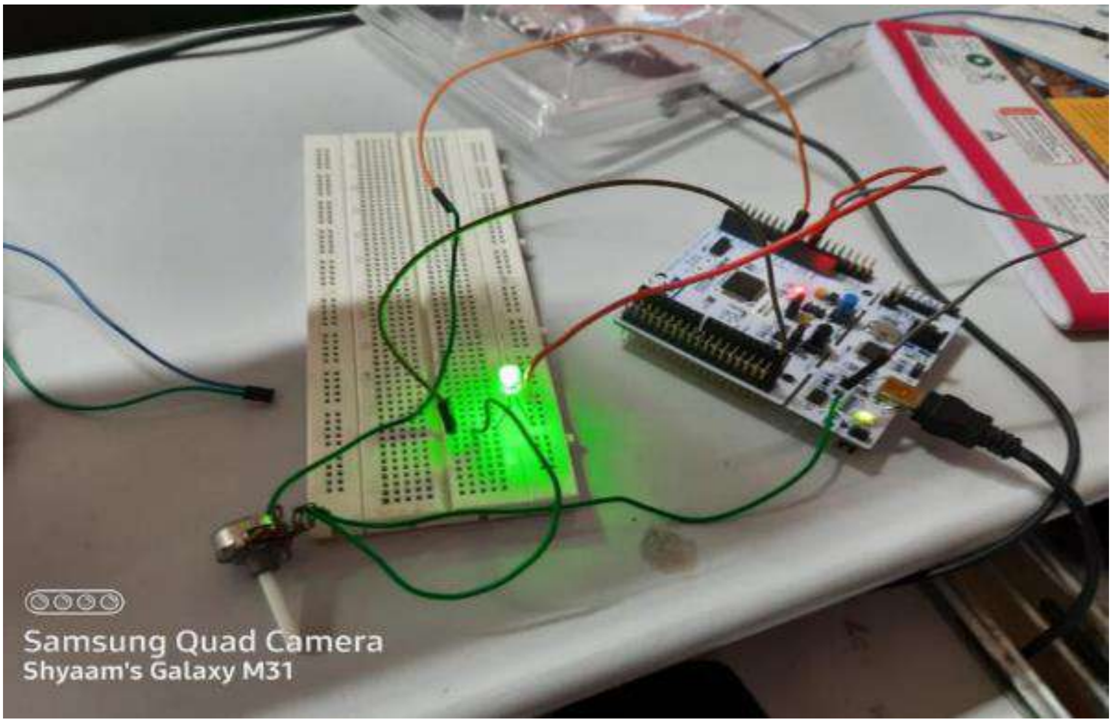


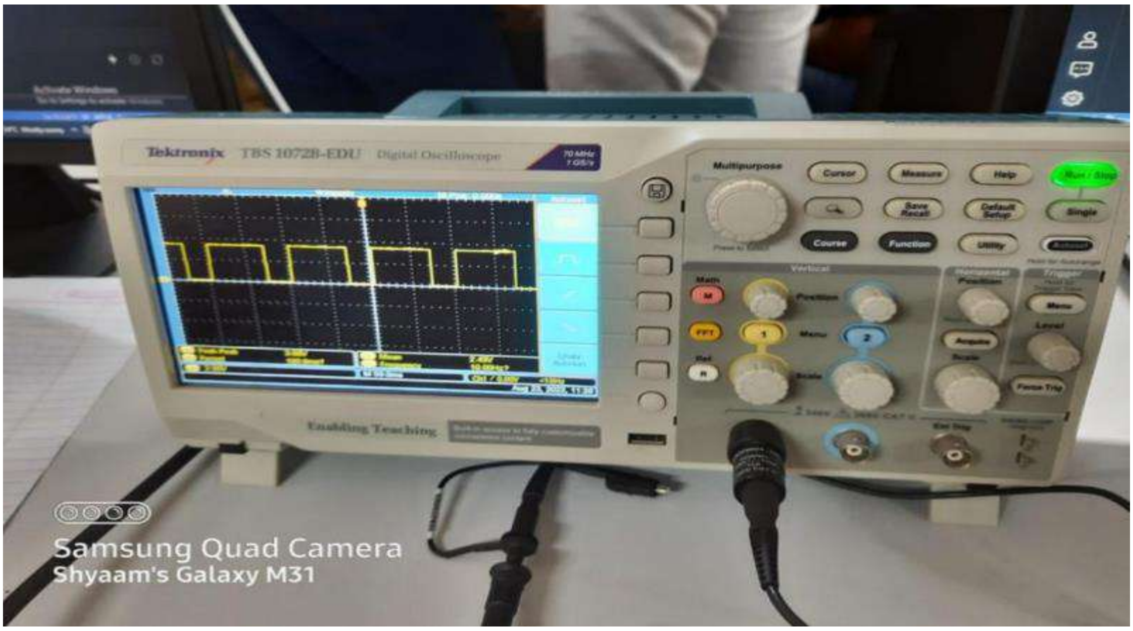
main.cpp ×

```cpp
#include "mbed.h"
// #include "mbed2/299/drivers/PwmOut.h"
// #include "mbed2/299/platform/wait_api.h"

PwmOut led(PC_8);
Serial pc(USBTX,USBRX);

int main() {
 //specify perwiod first
 led.period(.1f);
 led.write(.7f);   //50% duty cycle
    while(1)
    {
        pc.printf("High");
        wait(.7);
        pc.printf("Low");
        wait(.3);
    }
}

```

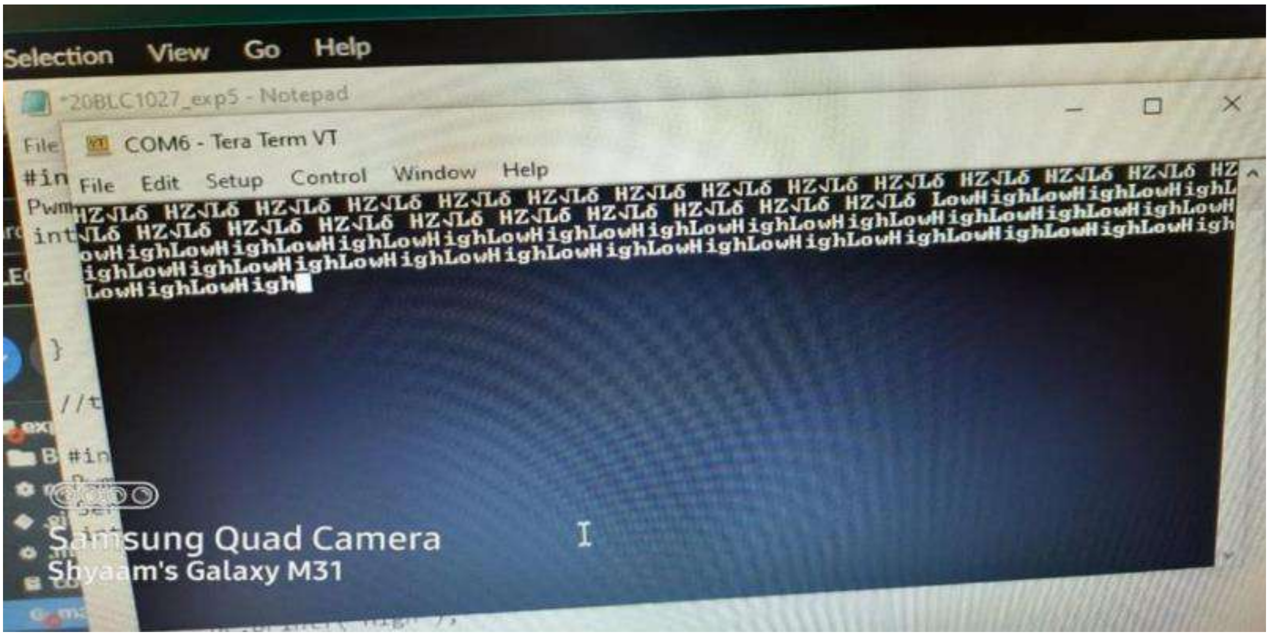
Shyaam S 20BLC1027 Date 23/08/2022

OUTPUT





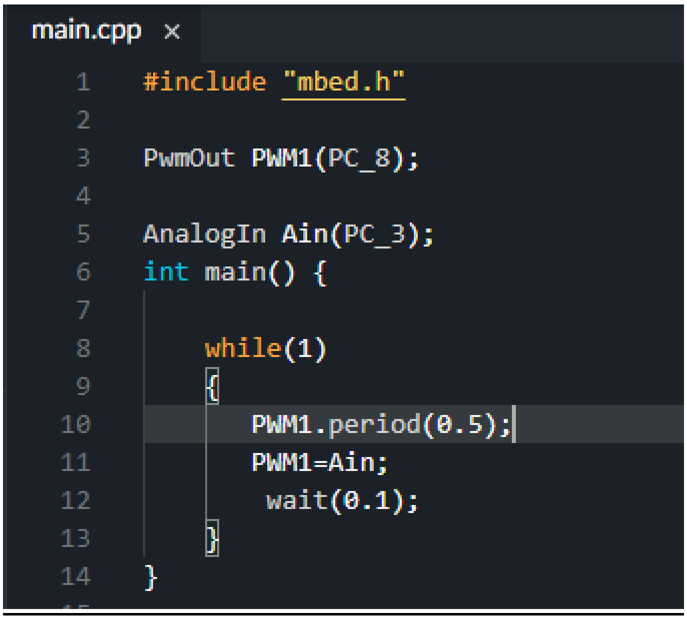
Shyaam S 20BLC1027 Date 23/08/2022



```
HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ
√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ HZ√Lδ LowHighLowHighLowHighL
owHighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowH
ighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowHighLowHigh
LowHighLowHigh
```

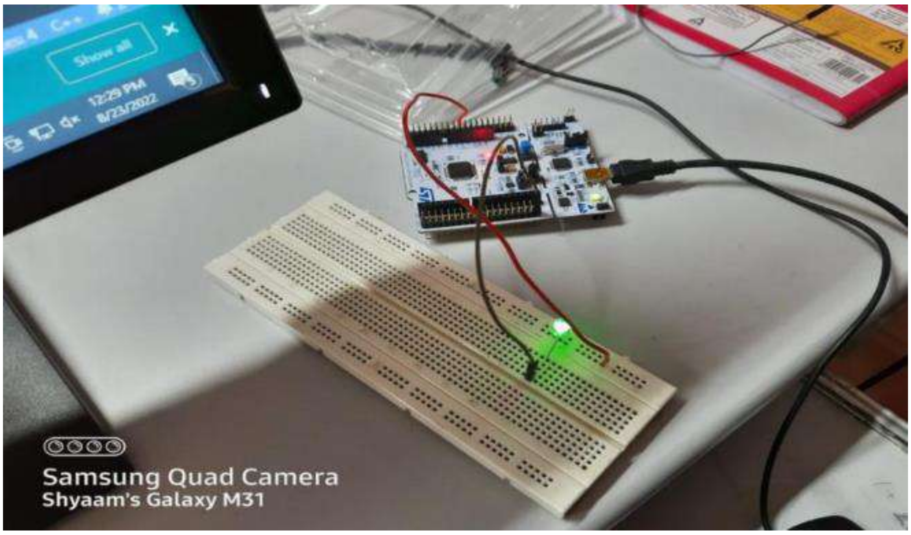
Shyaam S 20BLC1027 Date 23/08/2022

Task 3: Controlling LED brightness by using a POT on one of the analog input pin while PWM period remains the same.



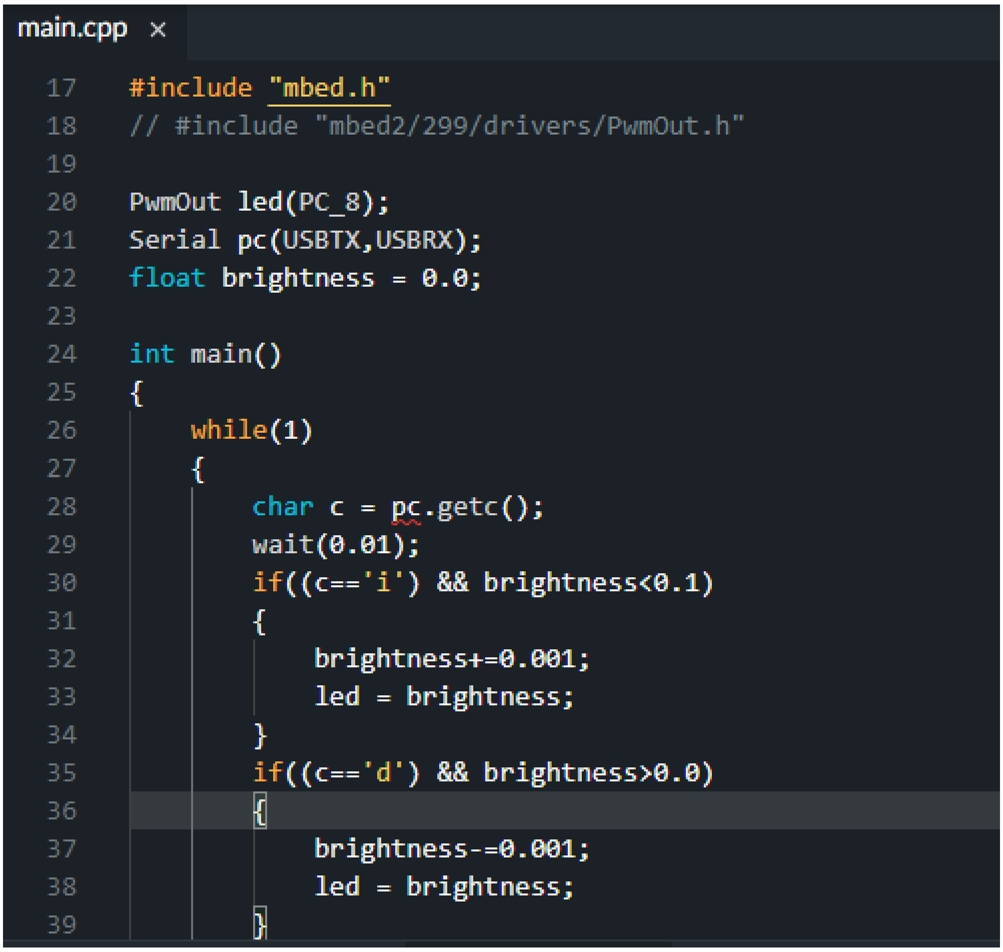
```
#include "mbed.h"

PwmOut PWM1(PC_8);

AnalogIn Ain(PC_3);
int main() {

    while(1)
    {
       PWM1.period(0.5);
       PWM1=Ain;
        wait(0.1);
    }
}
```

OUTPUT



Shyaam S 20BLC1027 Date 23/08/2022

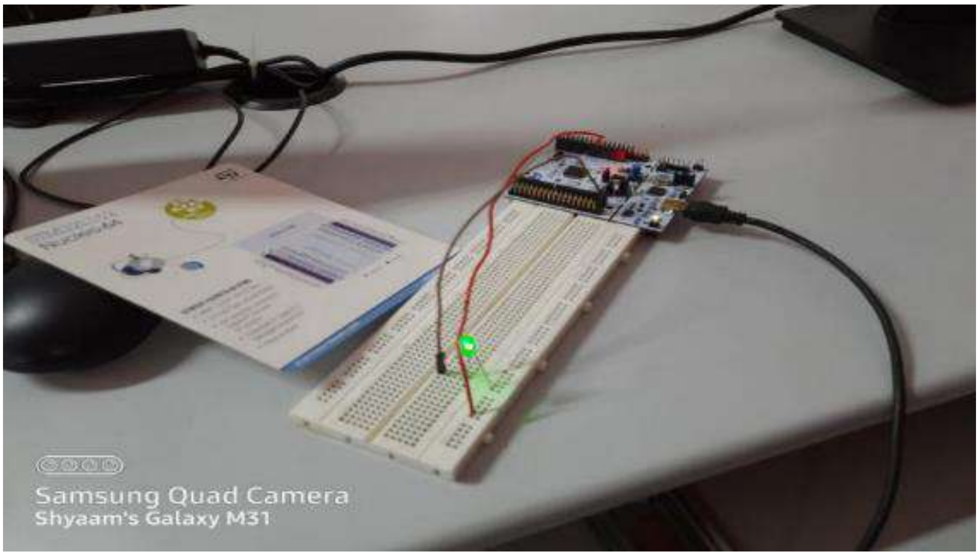
Task4: Generate a PWM signal to increase and decrease the brightness of the LED The program requires the use of a host terminal application to communicate the brightness value to the mbed by using the ‘i’ to increase the brightness and ‘d’ to decrease the brightness.

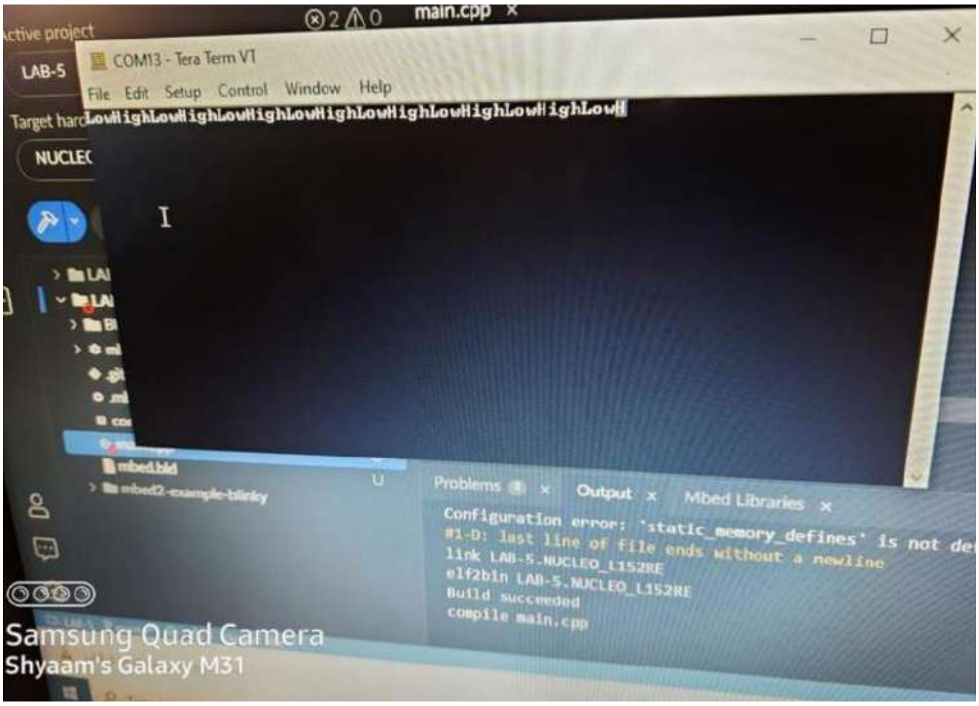


```cpp
#include "mbed.h"
// #include "mbed2/299/drivers/PwmOut.h"

PwmOut led(PC_8);
Serial pc(USBTX,USBRX);
float brightness = 0.0;

int main()
{
    while(1)
    {
        char c = pc.getc();
        wait(0.01);
        if((c=='i') && brightness<0.1)
        {
            brightness+=0.001;
            led = brightness;
        }
        if((c=='d') && brightness>0.0)
        {
            brightness-=0.001;
            led = brightness;
        }
```

OUTPUT

Shyaam S 20BLC1027 Date 23/08/2022

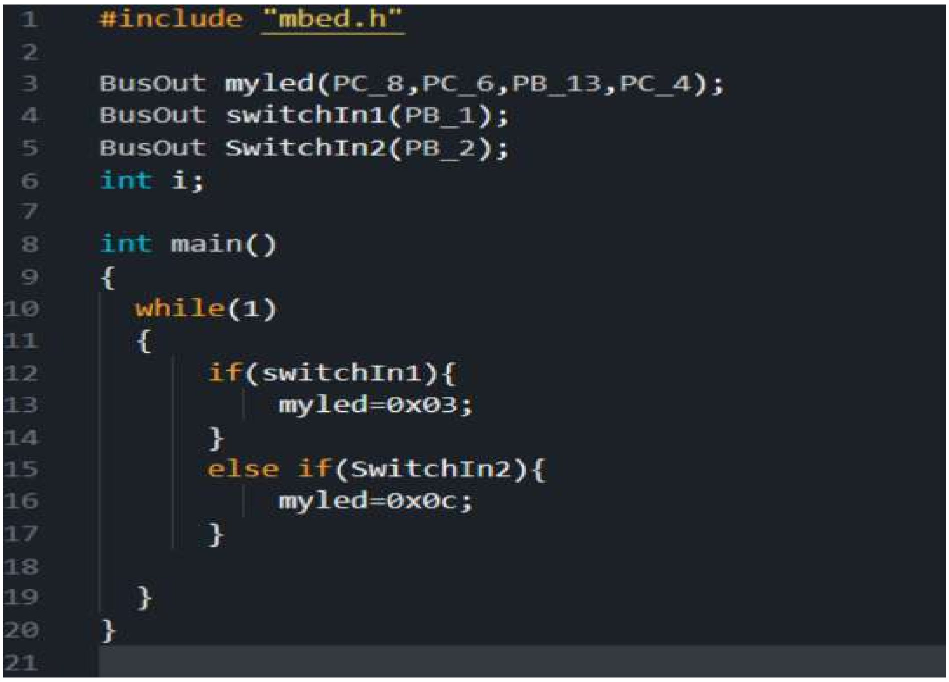




```
LowHighLowHighLowHighLowHighLowHighLowHighLowHighLowH
```

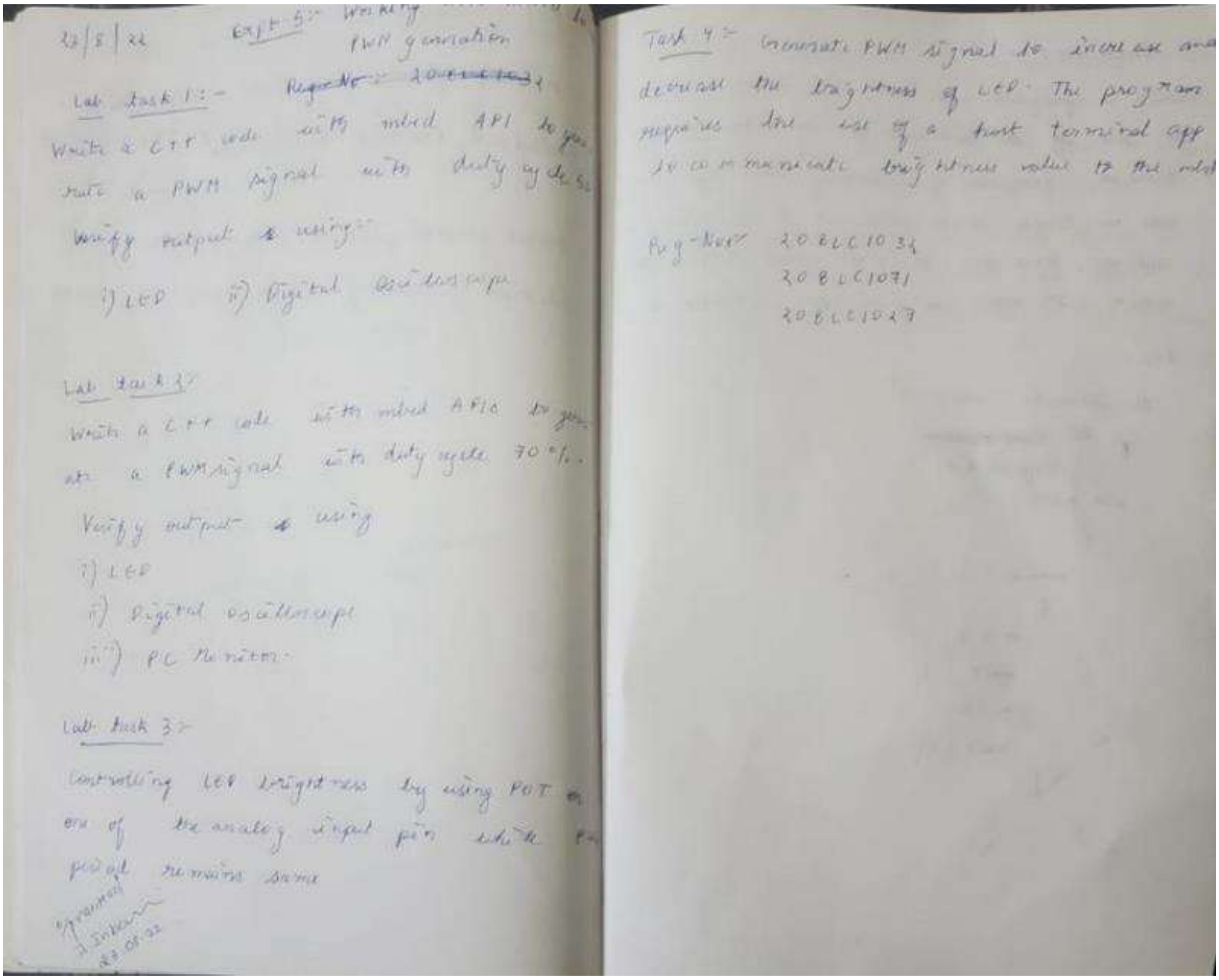
```
Configuration error: 'static_memory_defines' is not de
#1-D: last line of file ends without a newline
link LAB-5.NUCLEO_L152RE
elf2bin LAB-5.NUCLEO_L152RE
Build succeeded
compile main.cpp
```

CHALLENGING TASK: Read the multiple switch values and display the status of the switches in multiple LEDs. If the first switch is on then display 0x3 and if the second switch is on then display 0xC



```
#include "mbed.h"

BusOut myled(PC_8,PC_6,PB_13,PC_4);
BusOut switchIn1(PB_1);
BusOut SwitchIn2(PB_2);
int i;

int main()
{
  while(1)
  {
      if(switchIn1){
          myled=0x03;
      }
      else if(SwitchIn2){
          myled=0x0c;
      }

  }
}
```

Verification



22/8/22 Expt-5:- Working [illegible] PWM generation

Lab task 1:- Reg No :- [illegible]

Write a C++ code with mbed API to generate a PWM signal with duty cycle 50

Verify output using:-

i) LED ii) Digital oscilloscope

Lab task 2:-

Write a C++ code with mbed APIs to generate a PWM signal with duty cycle 70%.

Verify output using

i) LED

ii) Digital oscilloscope

iii) PC Monitor.

Lab task 3:-

Controlling LED brightness by using POT on one of the analog input pin while the period remains same

Approved
J Intern
23.08.22

Task 4:- Generate PWM signal to increase and decrease the brightness of LED. The program requires the use of a host terminal app to communicate brightness value to the mbed

Reg-No:- 20BLC1034
20BLC1071
20BLC1027

Inference

1. Learnt about working with Pulse Width Modulation generation on NUCLEO-L152RE
2. Learnt about connecting and interfacing the CRO to visualize the output.

Result

We used the PWM API to perform several tasks in the NUCLEO board.

Shyaam S 20BLC1027 Date 23/08/2022

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN

LAB RECORD

Submitted By

20BLC1027 – Shyaam S

Submitted To

Dr. SOFANA REKA S

LAB – 06

Embedded System Design

Experiment-6

Aim- Displaying several operations on the mbed board using the TextLCD library on the LCD .

Software Used-

Arm MBED online Compiler

Hardware Used-

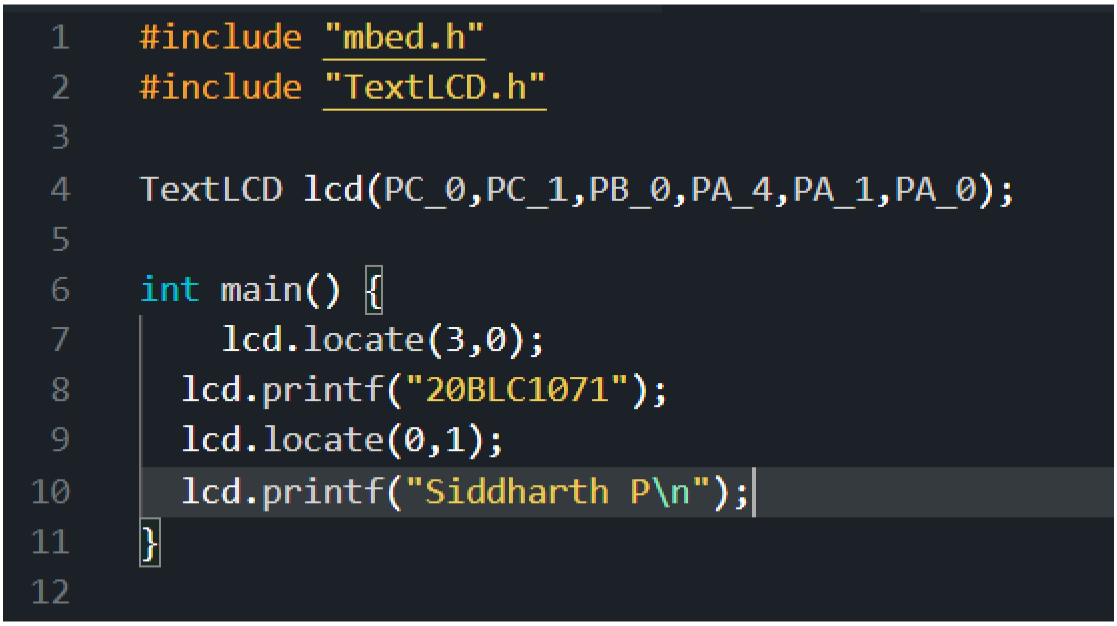
Micro USB cable, NUCLEO-L152RE Development Board, Breadboard,Jumper Wires, LCD,Terraterm

Procedure

1. Log-in into the mbed website by creating an account
2. Set-up the NUCLEO-L152RE Board.
3. Create A New-Project with template either Blinky LED Hello World or a Blank Template.
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the Board
8. Press Reset Button on the Board.
9. The code will now start its Execution.

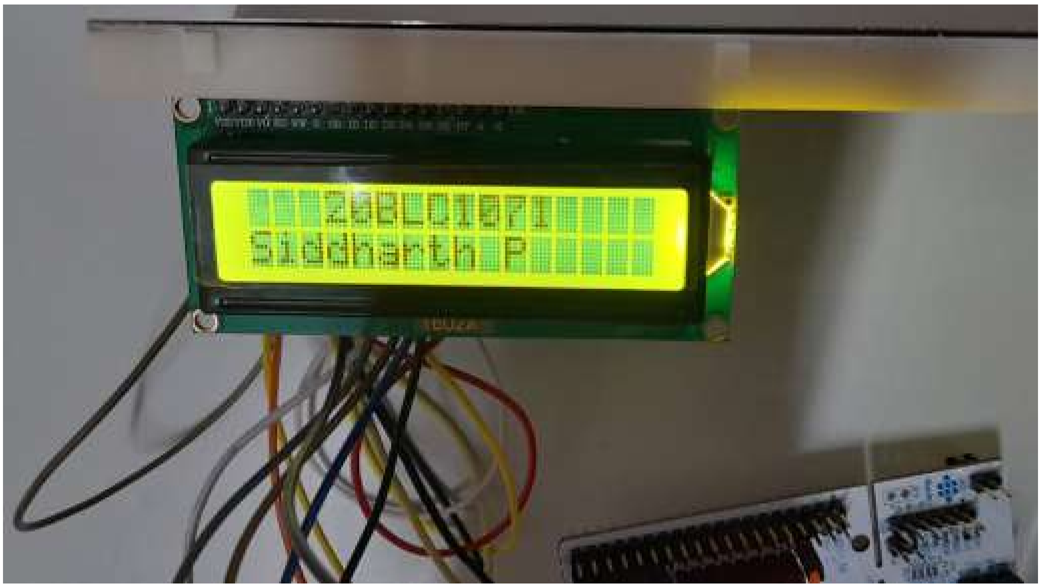
Task-1:

Display your register number at row 0, 3rd position and your name at the beginning of row 1. Assume LCD operates on 4 bit when EN and RS active state. Design and verify this logic .



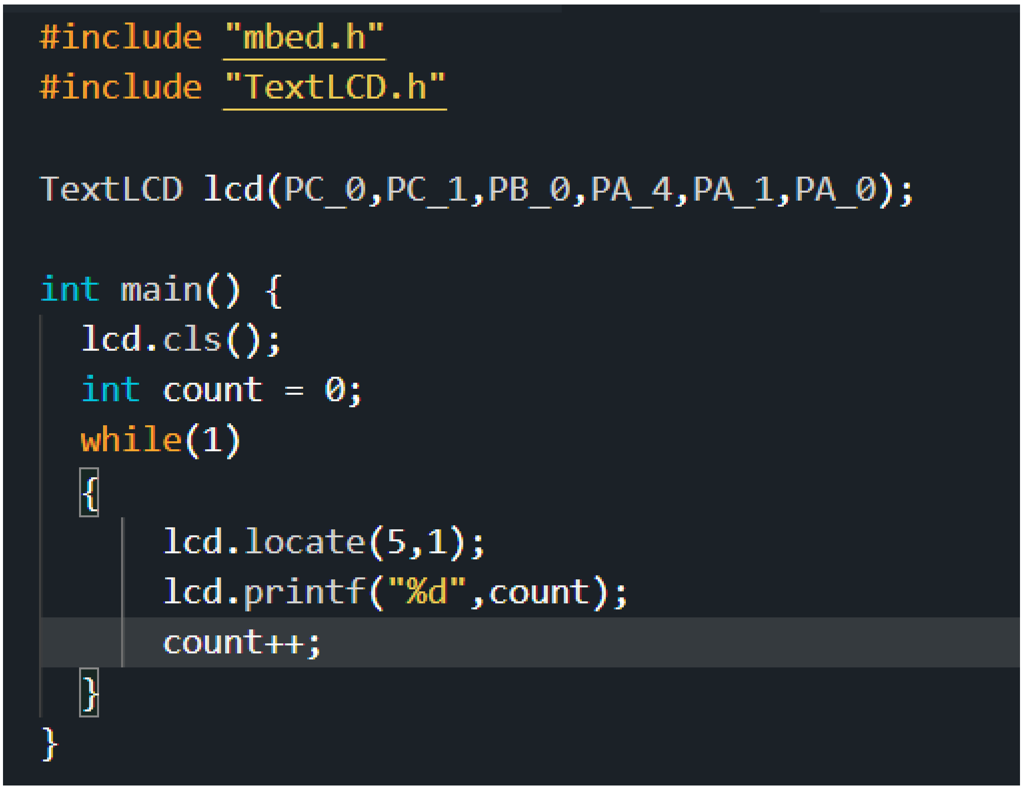
```
#include "mbed.h"
#include "TextLCD.h"

TextLCD lcd(PC_0,PC_1,PB_0,PA_4,PA_1,PA_0);

int main() {
  lcd.locate(3,0);
 lcd.printf("20BLC1071");
 lcd.locate(0,1);
 lcd.printf("Siddharth P\n");
}
```

OUTPUT:



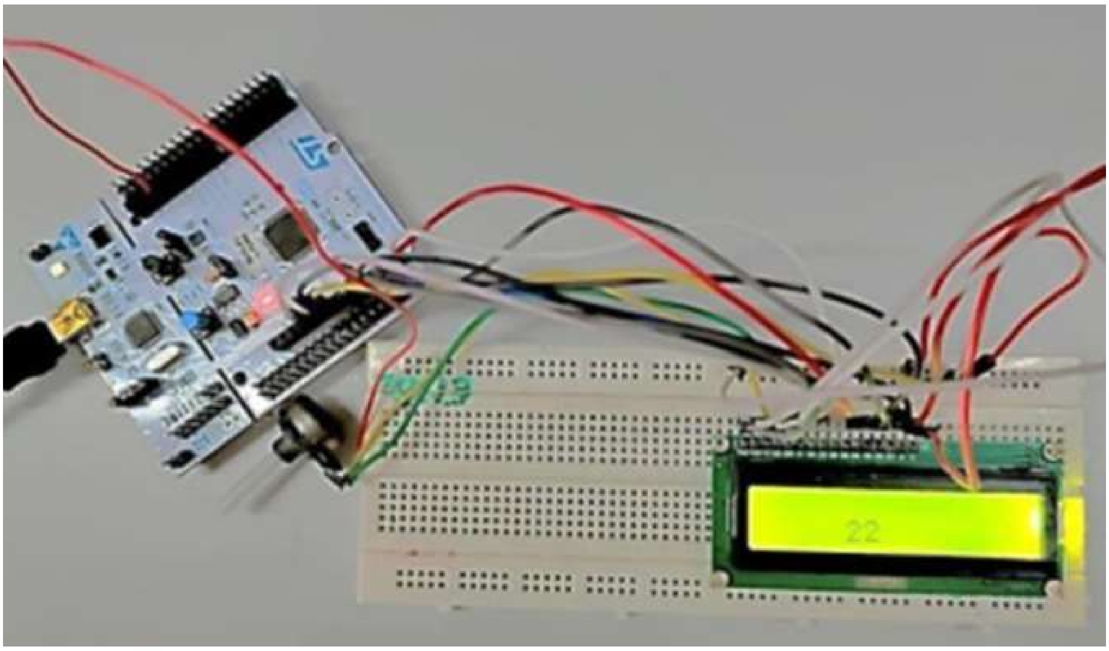
Task 2: Display a continuous count variable on the LCD display on row 1 5th position. Assume LCD operates on 4 bit when EN and RS active state. Design and verify this logic.

Code:



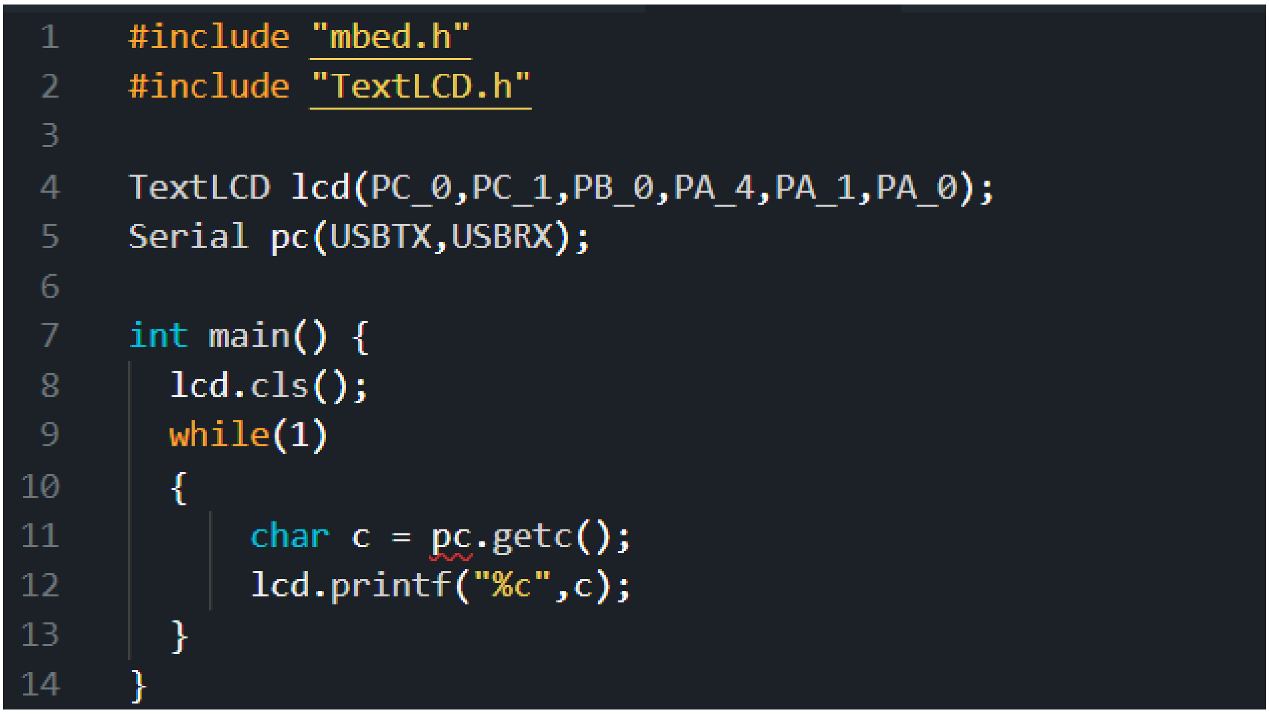
```
#include "mbed.h"
#include "TextLCD.h"

TextLCD lcd(PC_0,PC_1,PB_0,PA_4,PA_1,PA_0);

int main() {
    lcd.cls();
    int count = 0;
    while(1)
    {
        lcd.locate(5,1);
        lcd.printf("%d",count);
        count++;
    }
}
```

Output:



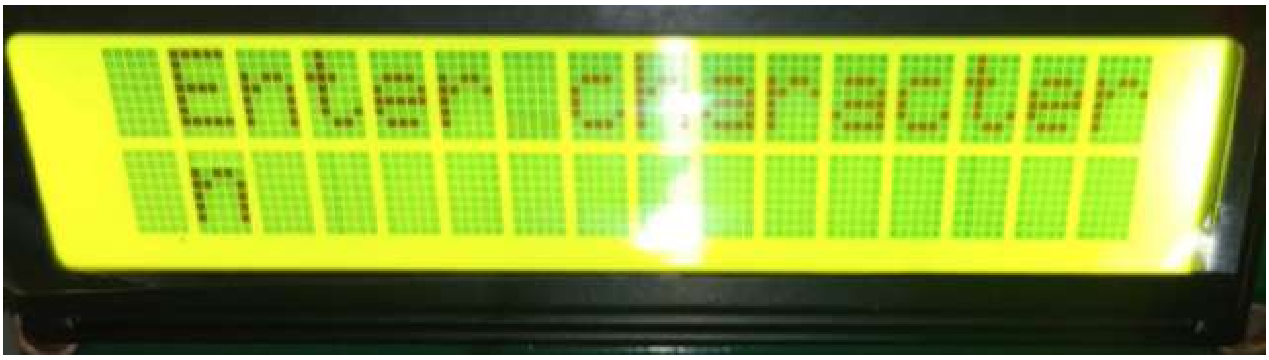
Task 3: Accept serial input character via teraterm software from host computer and display on LCD.

Code:

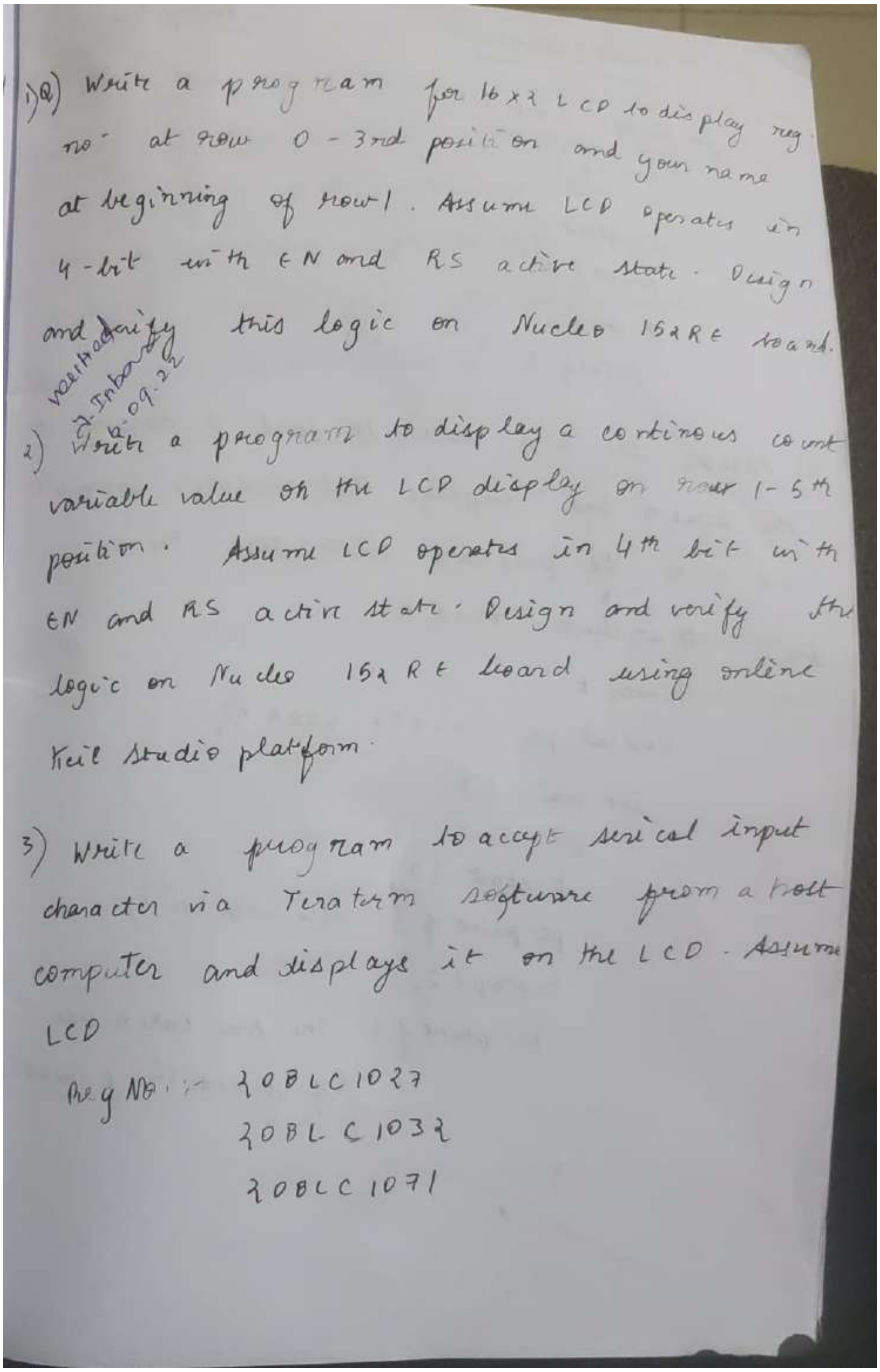


```
#include "mbed.h"
#include "TextLCD.h"

TextLCD lcd(PC_0,PC_1,PB_0,PA_4,PA_1,PA_0);
Serial pc(USBTX,USBRX);

int main() {
  lcd.cls();
  while(1)
  {
      char c = pc.getc();
      lcd.printf("%c",c);
  }
}
```

Output



Verification



1)Q) Write a program for 16x2 LCD to display reg. no. at row 0 - 3rd position and your name at beginning of row 1. Assume LCD operates in 4-bit with EN and RS active state. Design and verify this logic on Nucleo 152RE board.

verified J. Inbaraj 2.09.22

2) Write a program to display a continous count variable value on the LCD display on row 1- 5th position. Assume LCD operates in 4th bit with EN and RS active state. Design and verify the logic on Nucleo 152RE board using online Keil studio platform.

3) Write a program to accept serial input character via Teraterm software from a host computer and displays it on the LCD. Assume LCD

Reg No. :- 20BLC1027
20BLC1032
20BLC1071

Inference

1. Learnt about interfacing the LCD to the NUCLEO board and about the TextLCD api.
2. Understood the basics of the TextLCD api.

Result

We used the TextLCD API to perform several tasks in the NUCLEO board and display the results on the LCD.

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN

LAB RECORD

Submitted By

20BLC1027 - Shyaam S

Submitted To

Dr. SOFANA REKA S

LAB – 07

Embedded System Design

Experiment-7

Name- Shyaam S

Aim- Performing tasks using the servo motor and LDR with the help of the STM 32board.

Software Used-

Arm MBED online Compiler

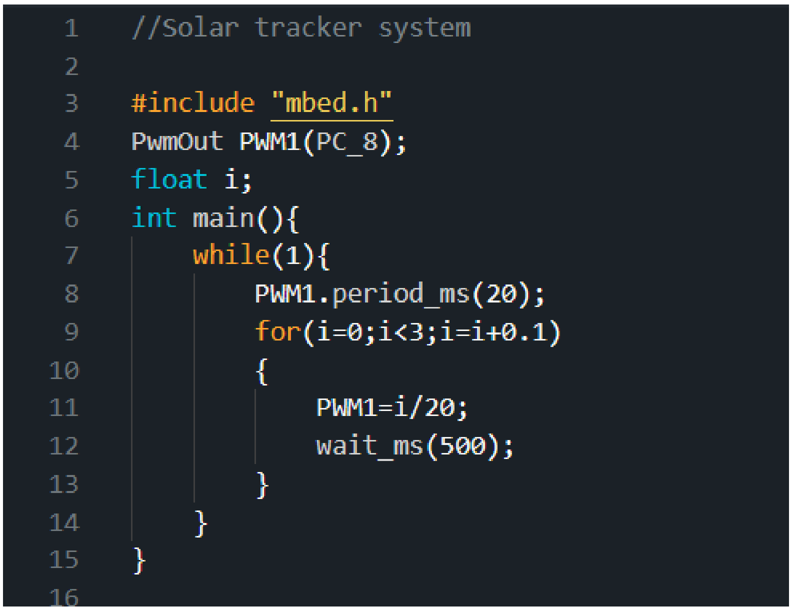
Hardware Used-

Micro USB cable, NUCLEO-L152RE Development Board, Bread board, potentiometer, 16x2 LCD Display.

Procedure

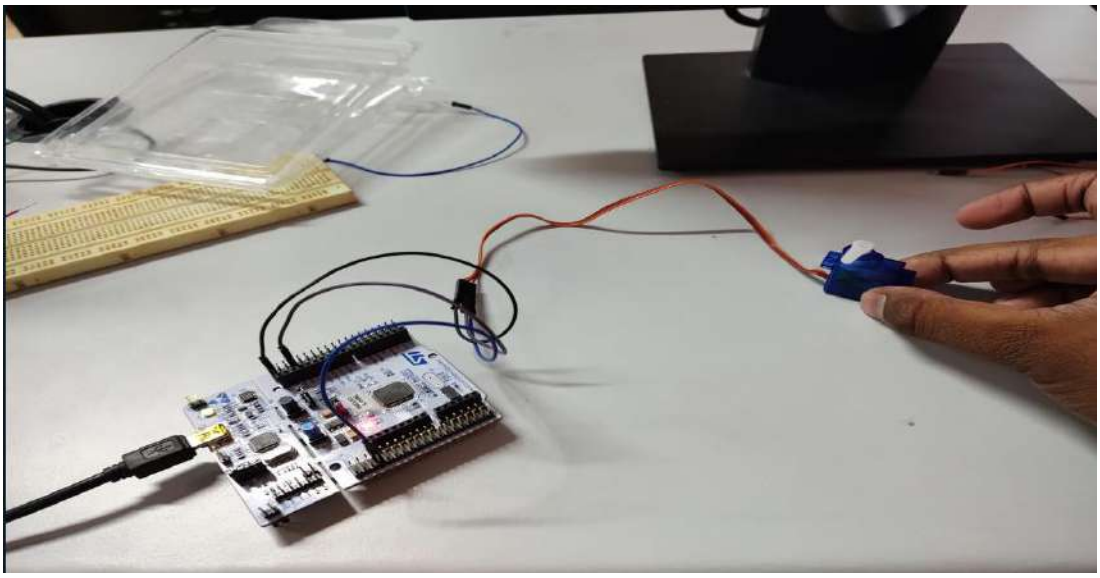
1. Log-in into the mbed website.
2. Set-up the NUCLEO-L152RE Board.
3. Create A New-Project with template either Blinky LED Hello World or a Blank Template.
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the Board
8. Press Reset Button on the Board.
9. The code will now start its Execution.

Task 1: Controlling the servo motor at 1 degree at a time
Write a program to control the servo motor by rotating slowly from 0 degrees to 180 degrees, one degree at a time. When the motor has to be rotated 180degrees, it will begin to rotate in the other direction until it returns to the initial position. Simulate and verify this login on the STM32 board.



```
//Solar tracker system

#include "mbed.h"
PwmOut PWM1(PC_8);
float i;
int main(){
    while(1){
        PWM1.period_ms(20);
        for(i=0;i<3;i=i+0.1)
        {
            PWM1=i/20;
            wait_ms(500);
        }
    }
}

```

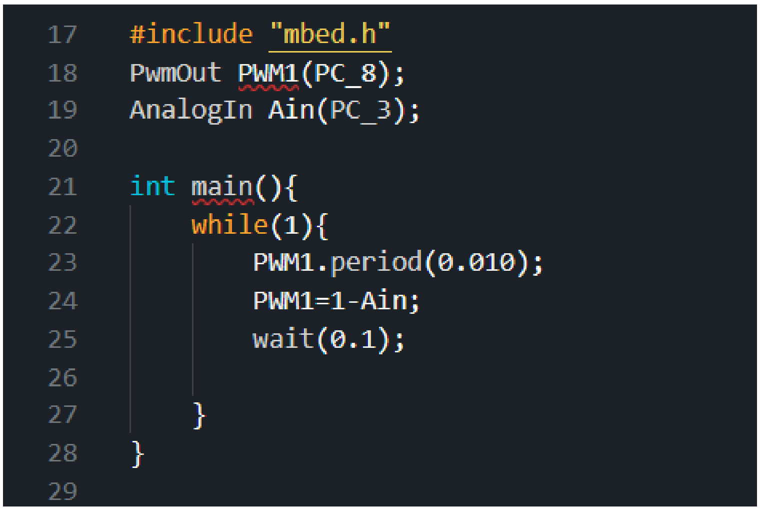
<u>OUTPUT:</u>



Task 2: Auto Intensity Street light controller

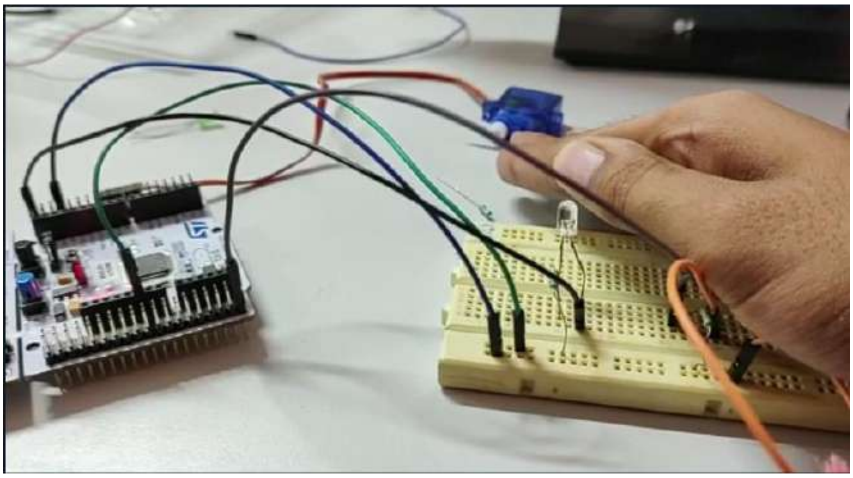
Write a program to design an auto-intensity street light controller. This system helps the street light to get switched on automatically as per surrounding brightness. For example sometimes when the weather becomes hazy it's quite difficult to see anything then at that point this auto-intensity street light gets switched on based on present lighting conditions.

CODE:



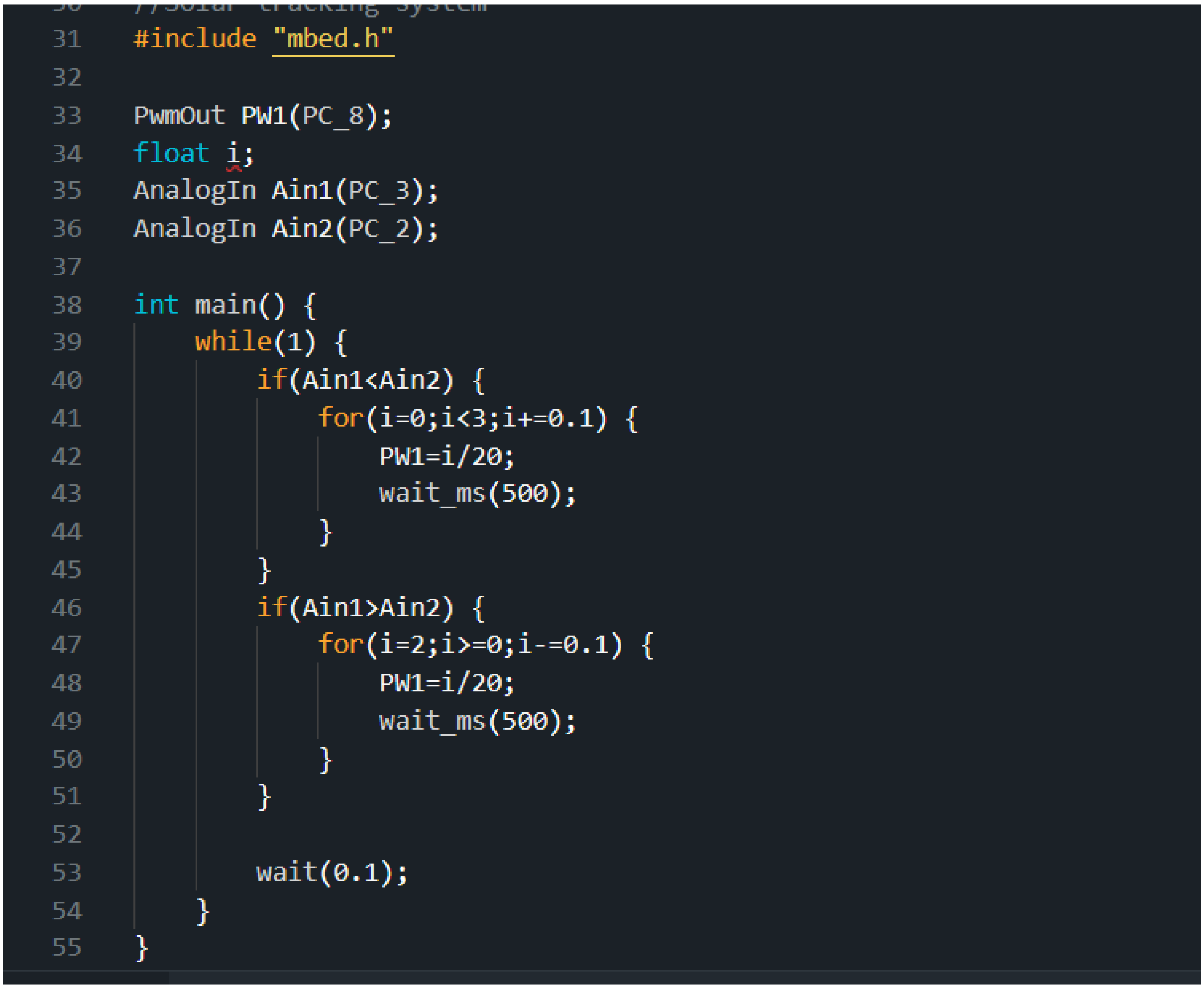
```
#include "mbed.h"
PwmOut PWM1(PC_8);
AnalogIn Ain(PC_3);

int main(){
    while(1){
        PWM1.period(0.010);
        PWM1=1-Ain;
        wait(0.1);

    }
}

```

<u>OUTPUT :</u>



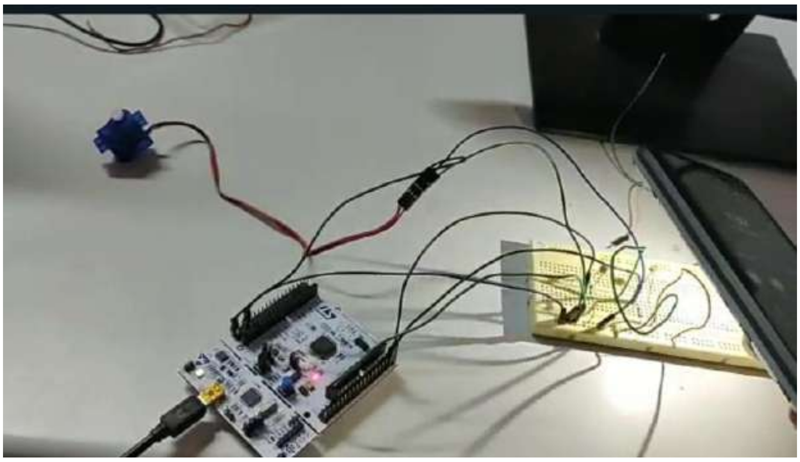
Challenging Task : Write a program to design a solar tracking system for harvesting solar energy efficiently by the solar panels. This system is constructed by fitting two LDRs angled away from each other by around 90, to a servo. Continuously read the light value sensed by the two LDRs and rotate the servo so that each is receiving equal light. Asa sun-tracking system will be located to track the sun from sunrise to sunset, i.e notmore than 180. Also, display both LDR values and present servo motor position on LCD.

CODE:

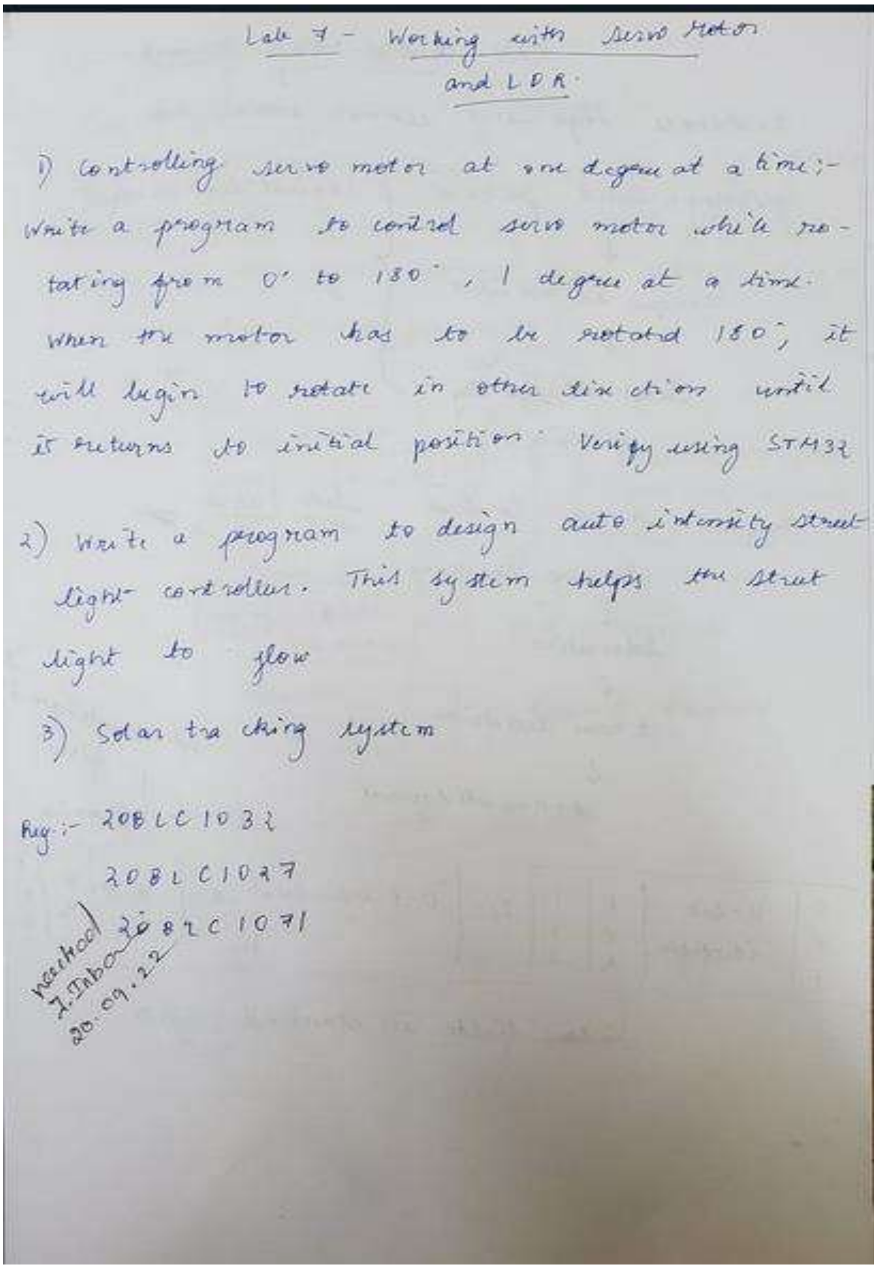


```
//Solar tracking system
#include "mbed.h"

PwmOut PW1(PC_8);
float i;
AnalogIn Ain1(PC_3);
AnalogIn Ain2(PC_2);

int main() {
    while(1) {
        if(Ain1<Ain2) {
            for(i=0;i<3;i+=0.1) {
                PW1=i/20;
                wait_ms(500);
            }
        }
        if(Ain1>Ain2) {
            for(i=2;i>=0;i-=0.1) {
                PW1=i/20;
                wait_ms(500);
            }
        }

        wait(0.1);
    }
}
```

OUTPUT :



Verification



Lab 7 - Working with Servo Motor and LDR.

1) Controlling servo motor at one degree at a time:-
Write a program to control servo motor while rotating from 0° to 180°, 1 degree at a time. When the motor has to be rotated 180°, it will begin to rotate in other direction until it returns to initial position. Verify using STM32

2) Write a program to design auto intensity street light controller. This system helps the street light to flow

3) Solar tracking system

Reg:- 20BLC1032
20BLC1027
20BLC1071

reached
I.Inba
20.09.22

Inference

1. Get familiar with the fundamental capabilities and features of the mbed nucleo board and mbed online compiler.
2. Acquired knowledge of libraries such as TextLCd.

Result

Hence the tasks were completed and verified on the NUCLEO-L152RE

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN

LAB RECORD

Submitted By

20BLC1027 – SHYAAM S

Submitted To

Dr. SOFANA REKA S

LAB – 08

Embedded System Design

Experiment-7

Name- Shyaam S

Aim- - Performing several operations on the Mbed board using the

SPI, Digital Out and Serial API to establish communication between Master

and Slave boards.

Software Used-

Arm MBED online Compiler

Hardware Used-

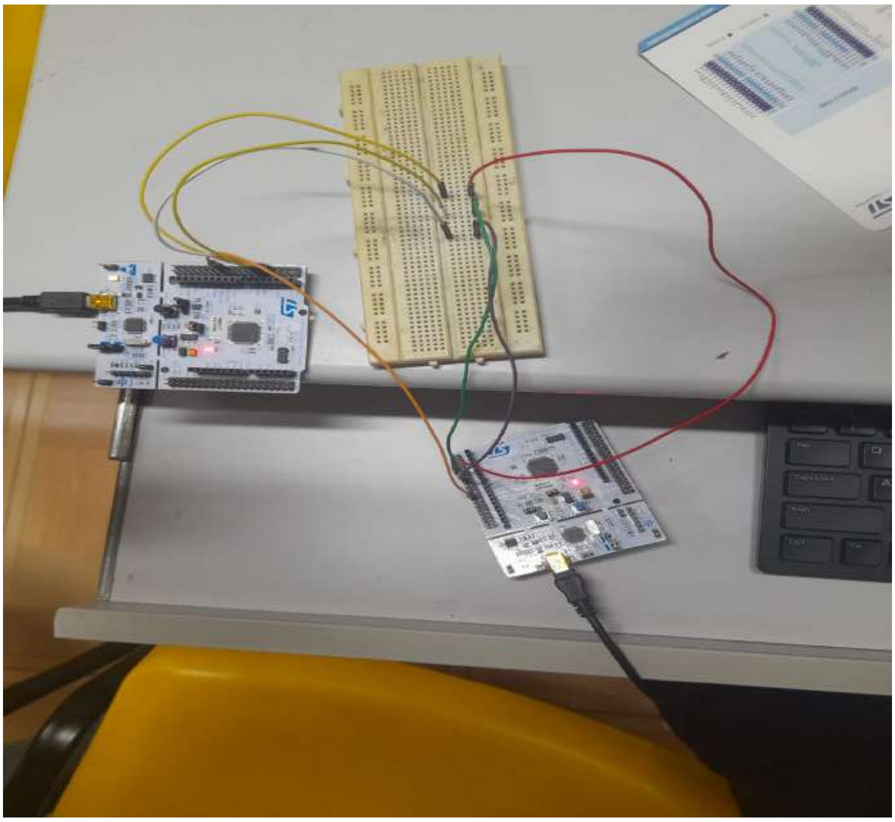
Micro USB cable, NUCLEO-L152RE Development Board, Breadboard,Jumper Wires, Servo motor,LDR

Procedure

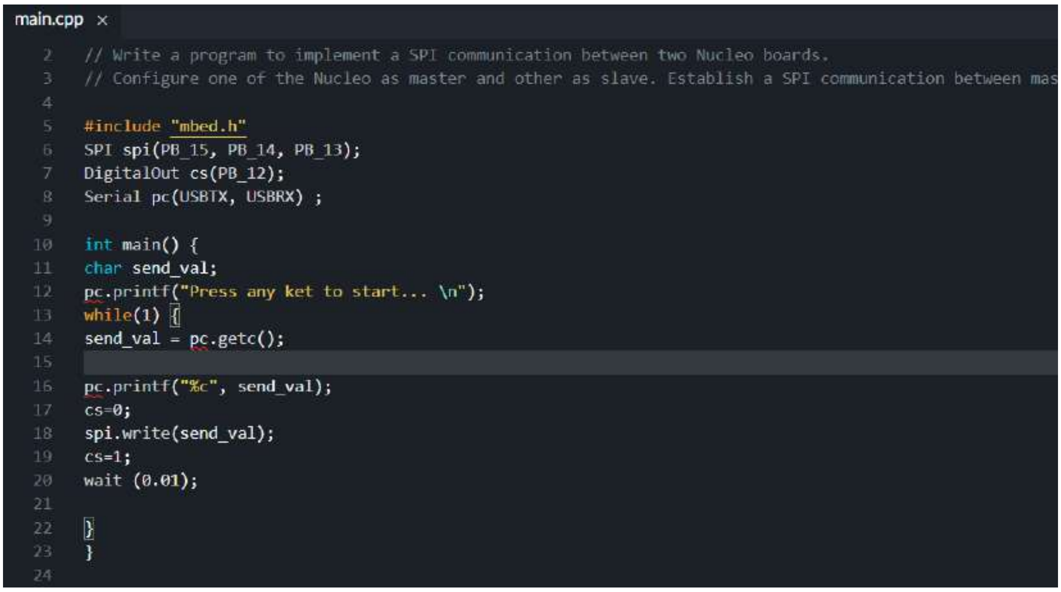
1. Log-in into the mbed website by creating an account
2. Set-up the NUCLEO-L152RE Board.
3. Create A New-Project with template either Blinky LED Hello World or a Blank Template.
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the Board
8. Press Reset Button on the Board.
9. The code will now start its Execution.

Task-1:

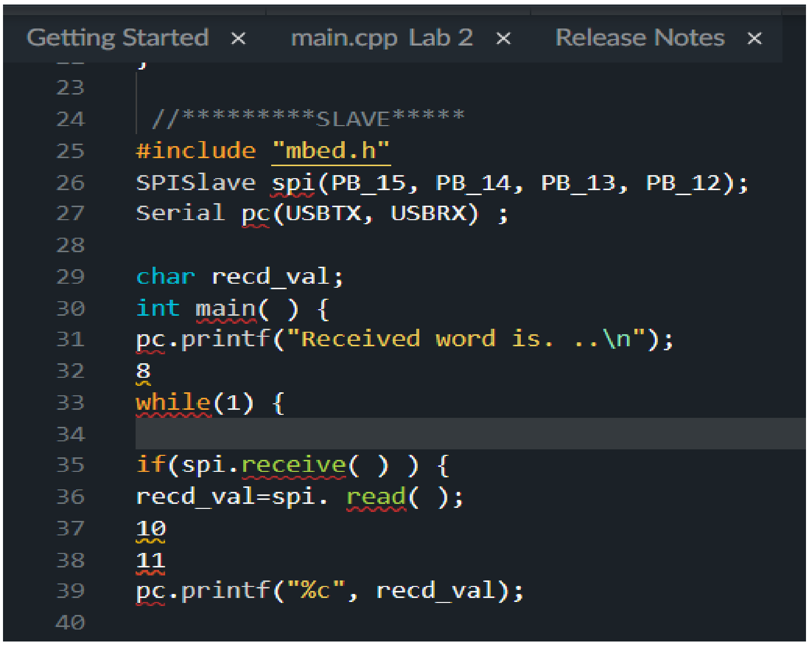
Write a program to implement a SPI communication between two Nucleo boards. Configure one of the Nucleo board as master and other as slave. Establish a SPI communication between master and slave display each key press on the master's Tera term to the slave Tera term terminal.



CODE:

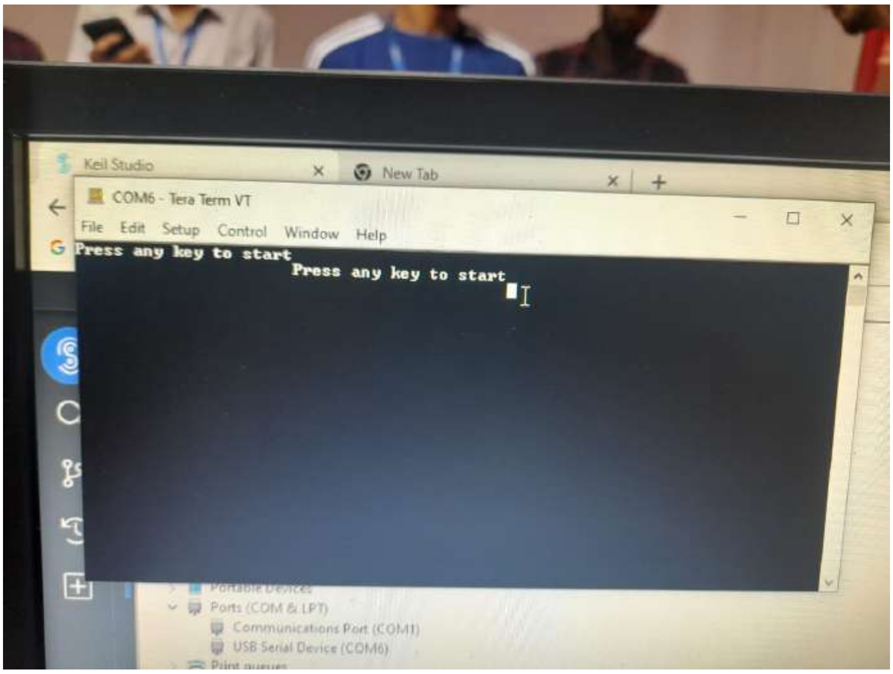


```
// Write a program to implement a SPI communication between two Nucleo boards.
// Configure one of the Nucleo as master and other as slave. Establish a SPI communication between mas

#include "mbed.h"
SPI spi(PB_15, PB_14, PB_13);
DigitalOut cs(PB_12);
Serial pc(USBTX, USBRX) ;

int main() {
char send_val;
pc.printf("Press any ket to start... \n");
while(1) {
send_val = pc.getc();

pc.printf("%c", send_val);
cs=0;
spi.write(send_val);
cs=1;
wait (0.01);

}
}
```

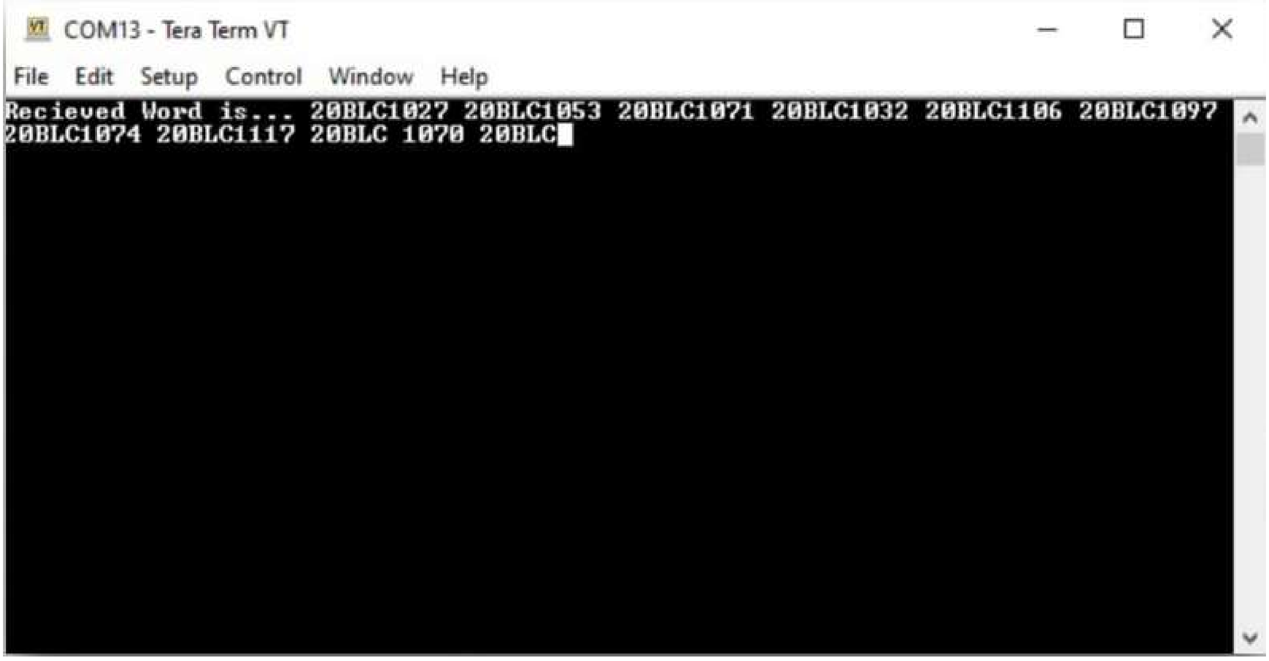


```
}

 //*********SLAVE*****
#include "mbed.h"
SPISlave spi(PB_15, PB_14, PB_13, PB_12);
Serial pc(USBTX, USBRX) ;

char recd_val;
int main( ) {
pc.printf("Received word is. ..\n");
8
while(1) {

if(spi.receive( ) ) {
recd_val=spi. read( );
10
11
pc.printf("%c", recd_val);
```

OUTPUT:



```
Press any key to start
                     Press any key to start
```

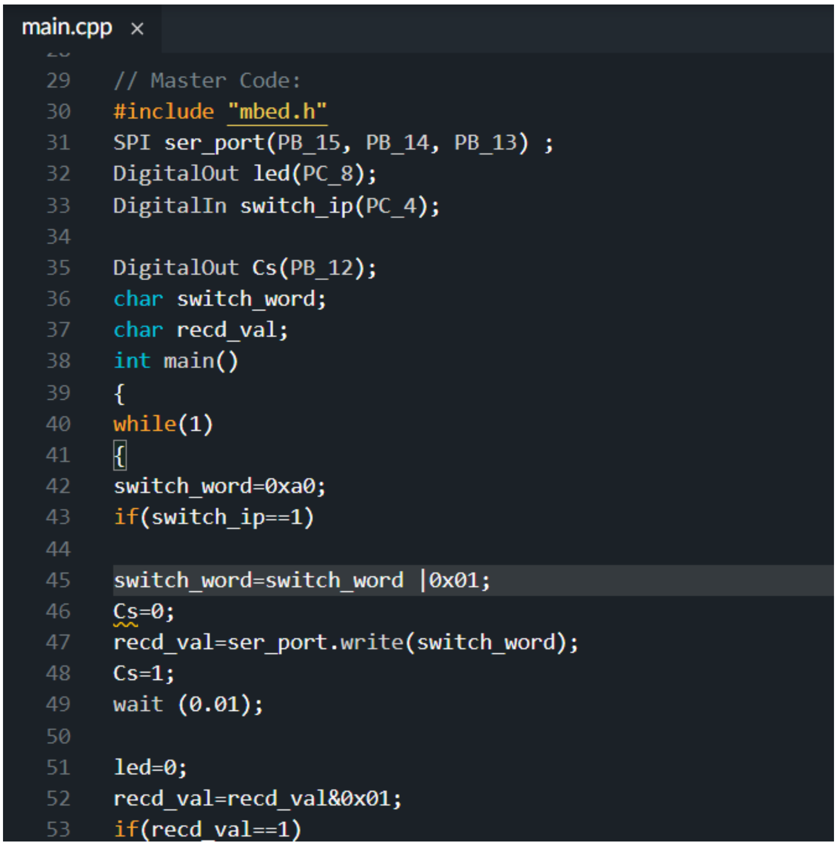


```
Recieved Word is... 20BLC1027 20BLC1053 20BLC1071 20BLC1032 20BLC1106 20BLC1097
20BLC1074 20BLC1117 20BLC 1070 20BLC
```

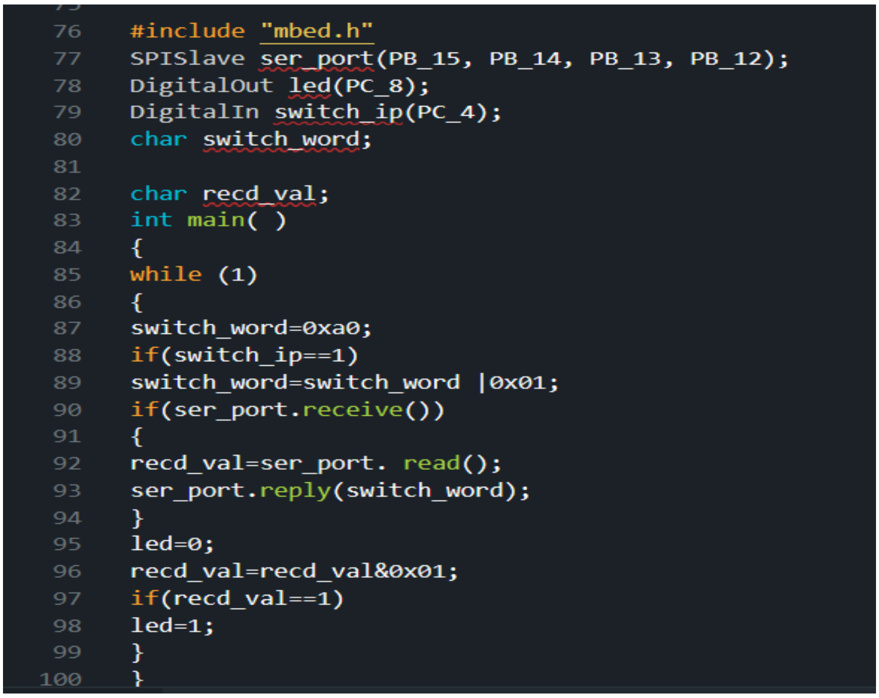
Task 2:

Write a program to implement a SPI communication between two Nucleo boards. Configure one of the Nucleo as master and other as slave. Both Nucleo are attached with a LED & a push button separately. Master LED can be controlled by using slave Nucleo's push button and slave Nucleo's LED can be controlled by master Nucleo's push button using SPI communication protocol.

Code:

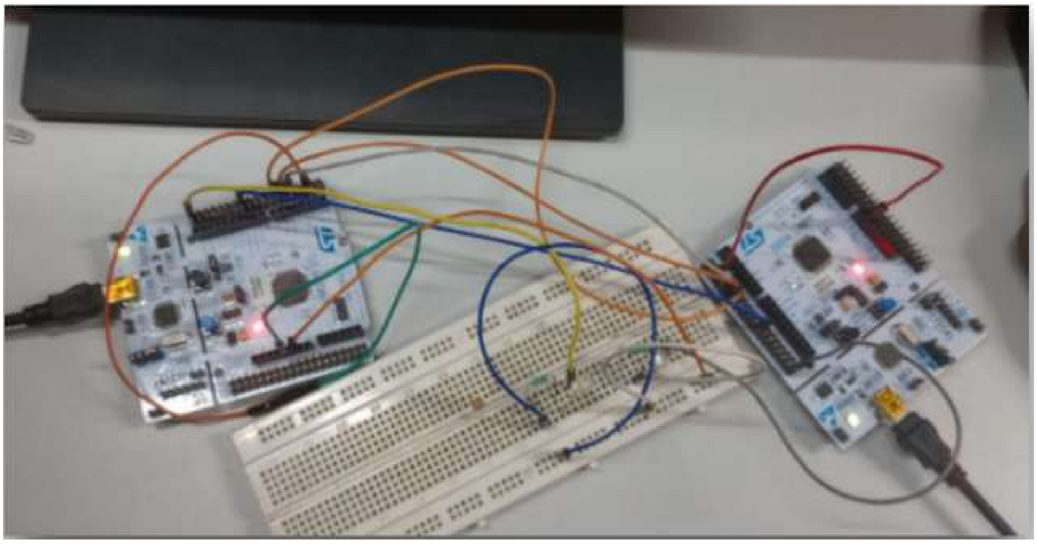


```
// Master Code:
#include "mbed.h"
SPI ser_port(PB_15, PB_14, PB_13) ;
DigitalOut led(PC_8);
DigitalIn switch_ip(PC_4);

DigitalOut Cs(PB_12);
char switch_word;
char recd_val;
int main()
{
while(1)
{
switch_word=0xa0;
if(switch_ip==1)

switch_word=switch_word |0x01;
Cs=0;
recd_val=ser_port.write(switch_word);
Cs=1;
wait (0.01);

led=0;
recd_val=recd_val&0x01;
if(recd_val==1)
```

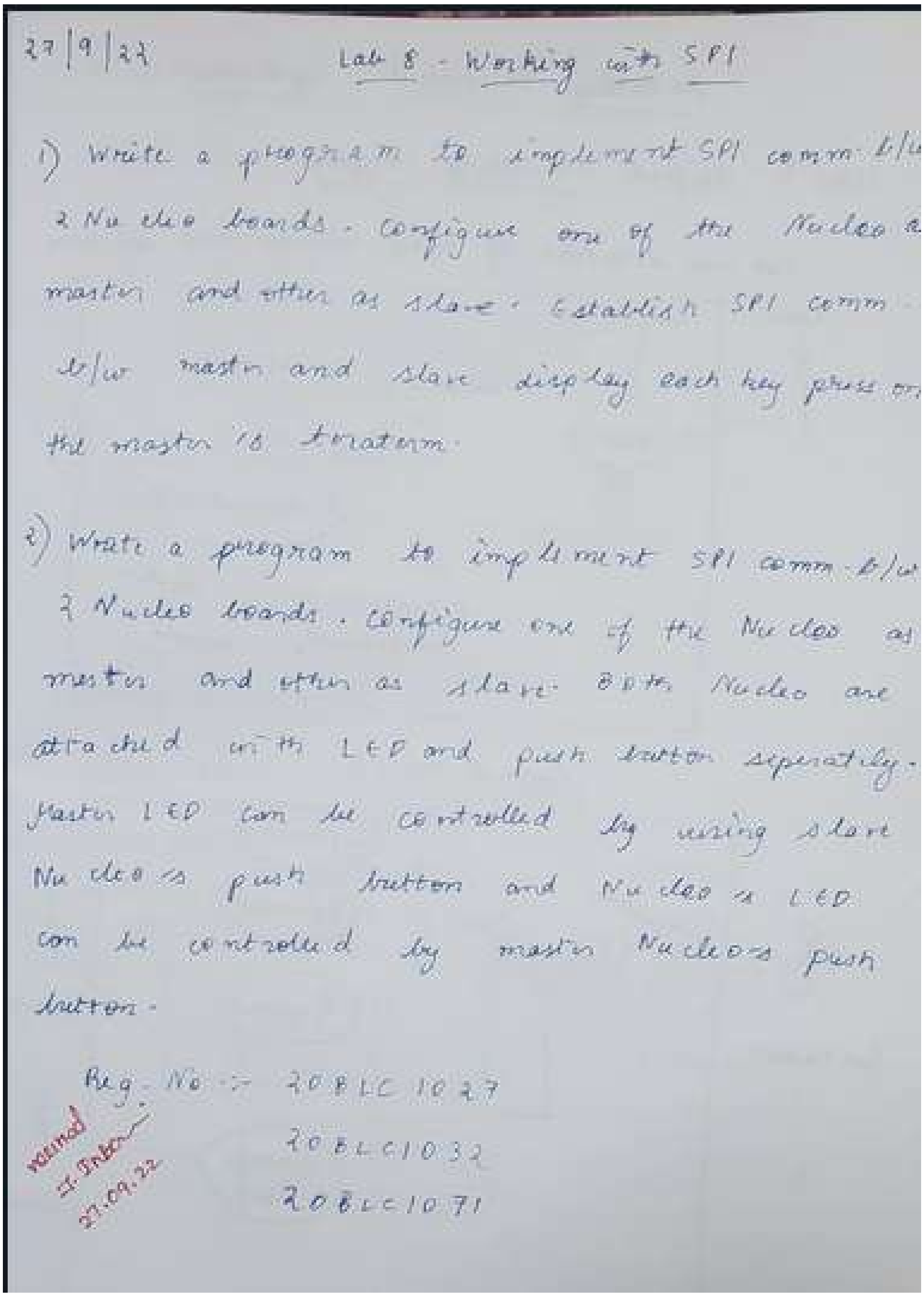


```
#include "mbed.h"
SPISlave ser_port(PB_15, PB_14, PB_13, PB_12);
DigitalOut led(PC_8);
DigitalIn switch_ip(PC_4);
char switch_word;

char recd_val;
int main( )
{
while (1)
{
switch_word=0xa0;
if(switch_ip==1)
switch_word=switch_word |0x01;
if(ser_port.receive())
{
recd_val=ser_port. read();
ser_port.reply(switch_word);
}
led=0;
recd_val=recd_val&0x01;
if(recd_val==1)
led=1;
}
}
```

Output:



VERIFICATION



27/9/22

Lab 8 - Working with SPI

1) Write a program to implement SPI comm. b/w 2 Nucleo boards. configure one of the Nucleo a master and other as slave. Establish SPI comm. b/w master and slave display each key press on the master is teraterm.

2) Write a program to implement SPI comm. b/w 2 Nucleo boards. configure one of the Nucleo as master and other as slave. Both Nucleo are attached with LED and push button seperately. Master LED can be controlled by using slave Nucleo's push button and Nucleo's LED can be controlled by master Nucleo's push button.

Reg. No :- 20BLC1027

20BLC1032

20BLC1071

verified

27.09.22

Inference

1. Worked with APIs such as SPI and SPISlave used for SPI communication.

2. Learnt about the basic functionalities of the Nucleo64-STM32L152 board and explore various connections to establish communication between Master and Slave board.

Result

The Nucleo 64-STM32L152 board APIs SPI and SPISlave has been studied and programs were written for SPI communication between the Master and the Slave boards.

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN

LAB RECORD

Submitted By

20BLC1027 – Shyaam S

Submitted To

Dr. SOFANA REKA S

LAB – 09

Embedded System Design

Experiment-9

Working with Bluetooth HC-05 module

Aim- To perform several operations on the MBED board using the Bluetooth HC-05 Bluetooth module.

TASK-1: Data transfer between HC-05 and PC

Write a C++ code with mbed APIs to communicate with smart phone via Bluetooth module by transferring ASCII values between them. Values received by mbed module will be displayed in PC Tera Term and value received by smartphones displaced on the App screen. Implement this logic on Nucleo board and HC-05 Bluetooth module using Keil studio cloud online compiler.

TASK-2: Bluetooth based home automation

Write a C++ code with mbed to control on board LED and DC motor using Bluetooth communication as per below logic. Get the value from smartphone through app and control the LED and DC motor accordingly. Implement the logic on Nucleo board and HC05 Bluetooth module using Keil studio cloud online compiler

"1" – LED OFF, "2" – LED ON, "3" – Motor OFF, "4" – Motor ON

TASK-3 (Challenging task): Wireless temperature transfer

Write a C++ code with mbed APIs to transfer room temperature value measure through LM35 wirelessly using Bluetooth module to the smartphone. Implement this logic on Nucleo board and HC-05 Bluetooth module using Keil studio cloud online compiler.

Software Used-

Arm MBED online Compiler

Hardware Used-

Micro USB Cable, Nucleo64 – STM32L152 Board, Jumper Wires, LEDs, Resistors, HC-05 Bluetooth module, Breadboard.

Procedure-

1. Log in to the Keil arm studio website by creating an account
2. Select File, and hover over new to start a new MBED project
3. Create A New-Project with template either Blinky LED
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the Board
8. Press Reset Button on the Board.
9. The code will now start its Execution.

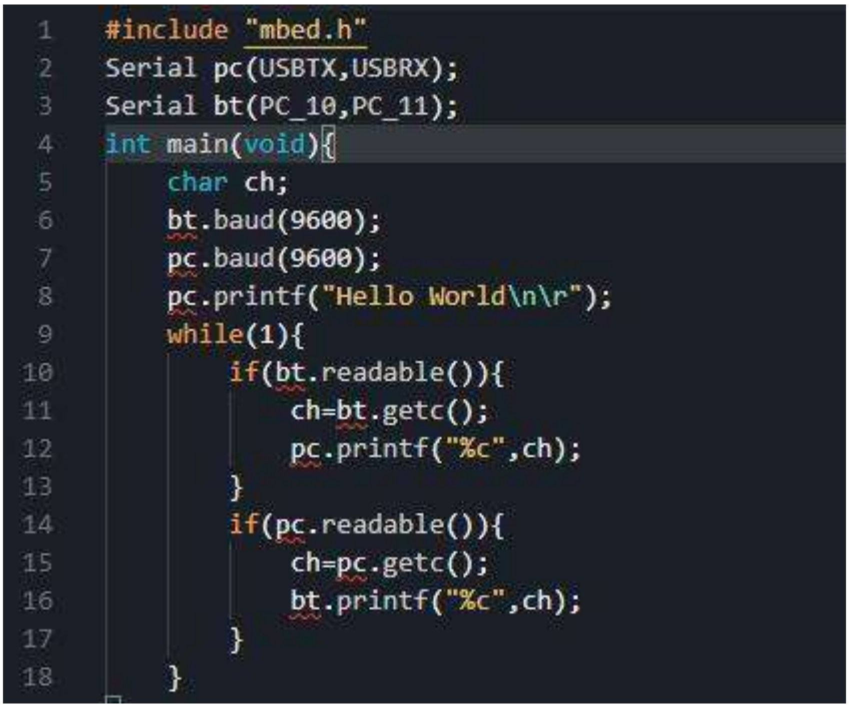
PROGRAM:

TASK-1

Data transfer between HC-05 and PC

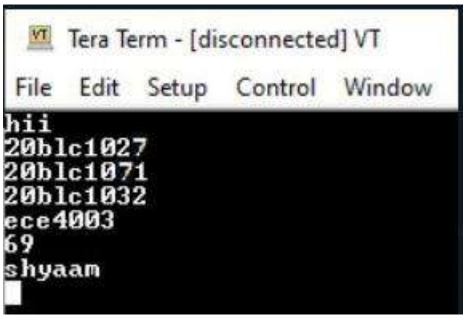
Write a C++ code with mbed APIs to communicate with smart phone via Bluetooth module by transferring ASCII values between them. Values received by mbed module will be displayed in PC Tera Term and value received by smartphones displaced on the App screen. Implement this logic on Nucleo board and HC-05 Bluetooth module using Keil studio cloud online compiler.

Code-

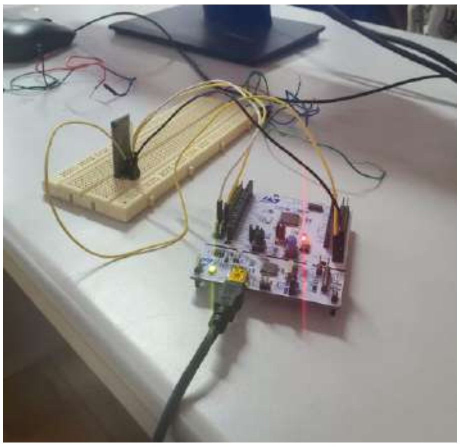


```
#include "mbed.h"
Serial pc(USBTX,USBRX);
Serial bt(PC_10,PC_11);
int main(void){
    char ch;
    bt.baud(9600);
    pc.baud(9600);
    pc.printf("Hello World\n\r");
    while(1){
        if(bt.readable()){
            ch=bt.getc();
            pc.printf("%c",ch);
        }
        if(pc.readable()){
            ch=pc.getc();
            bt.printf("%c",ch);
        }
    }
```

Output-



```
hii
20blc1027
20blc1071
20blc1032
ece4003
69
shyaam
```



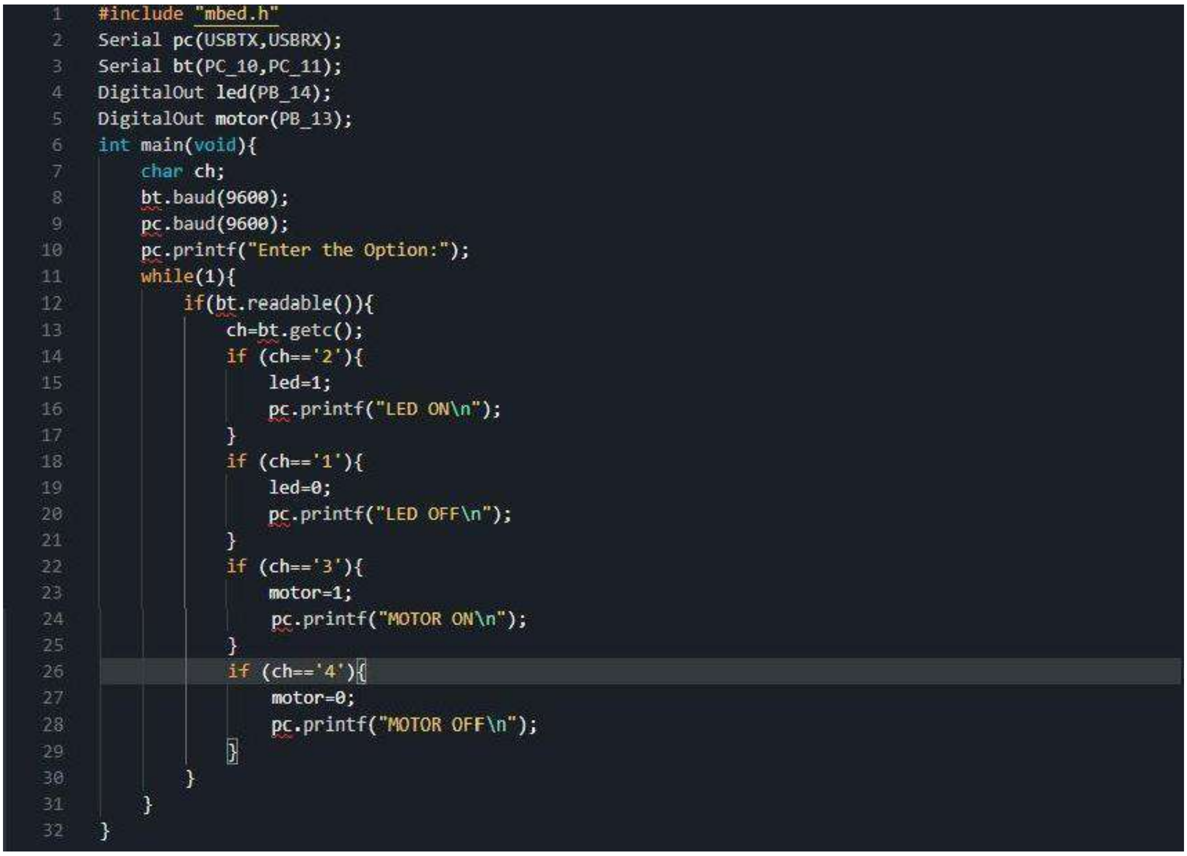
TASK – 2

Bluetooth based home automation

Write a C++ code with mbed to control on board LED and DC motor using Bluetooth communication as per below logic. Get the value from smartphone through app and control the LED and DC motor accordingly. Implement the logic on Nucleo board and HC05 Bluetooth module using Keil studio cloud online compiler.

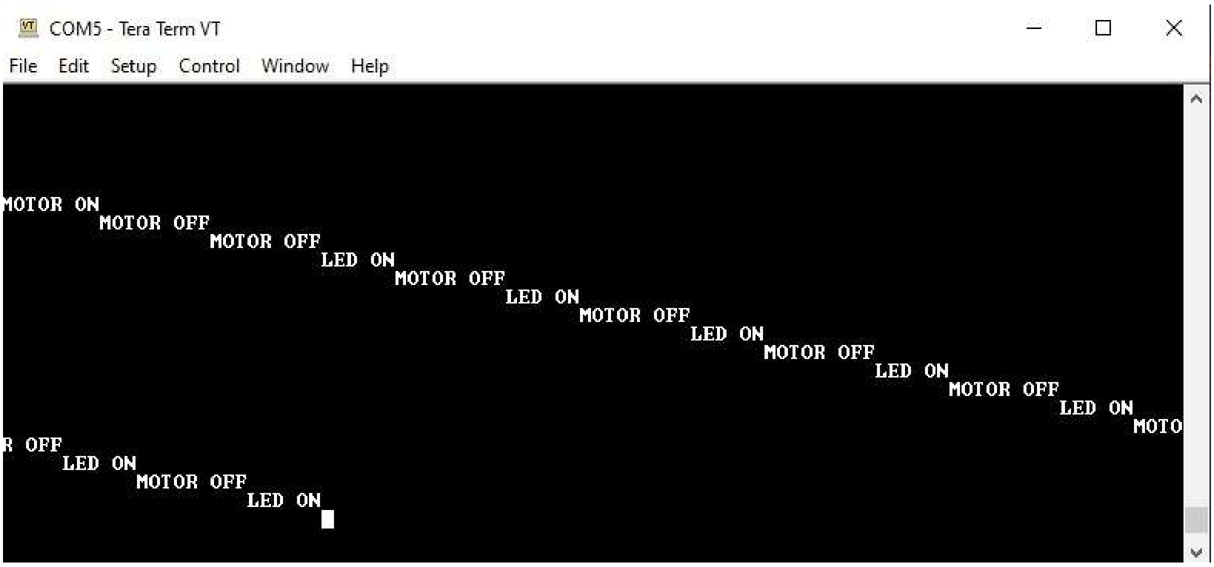
“1” – LED OFF, “2” – LED ON, “3” – Motor OFF, “4” – Motor ON

Code-

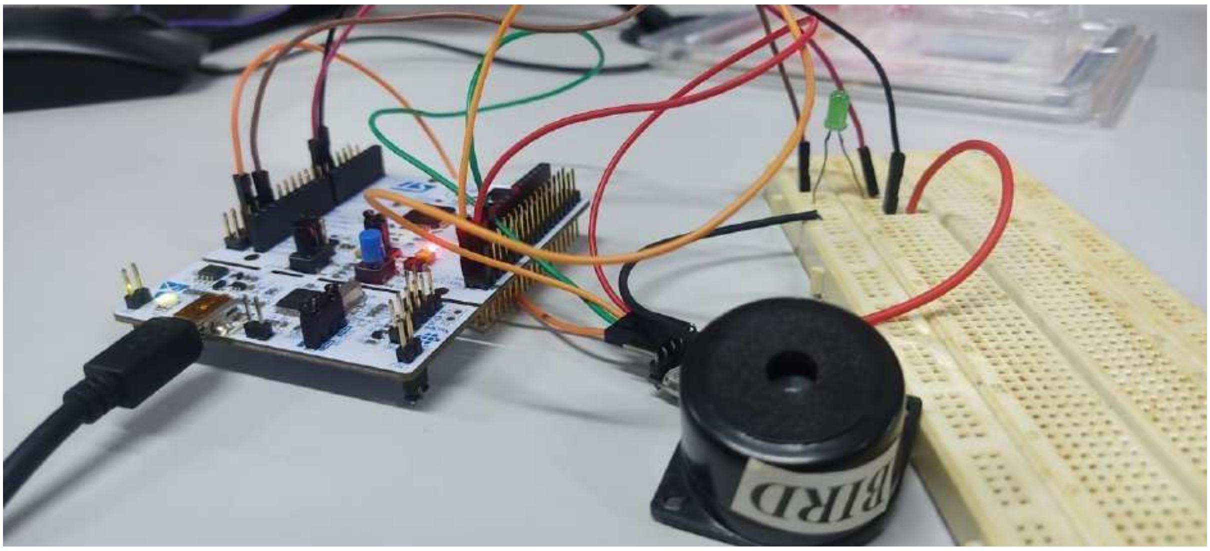


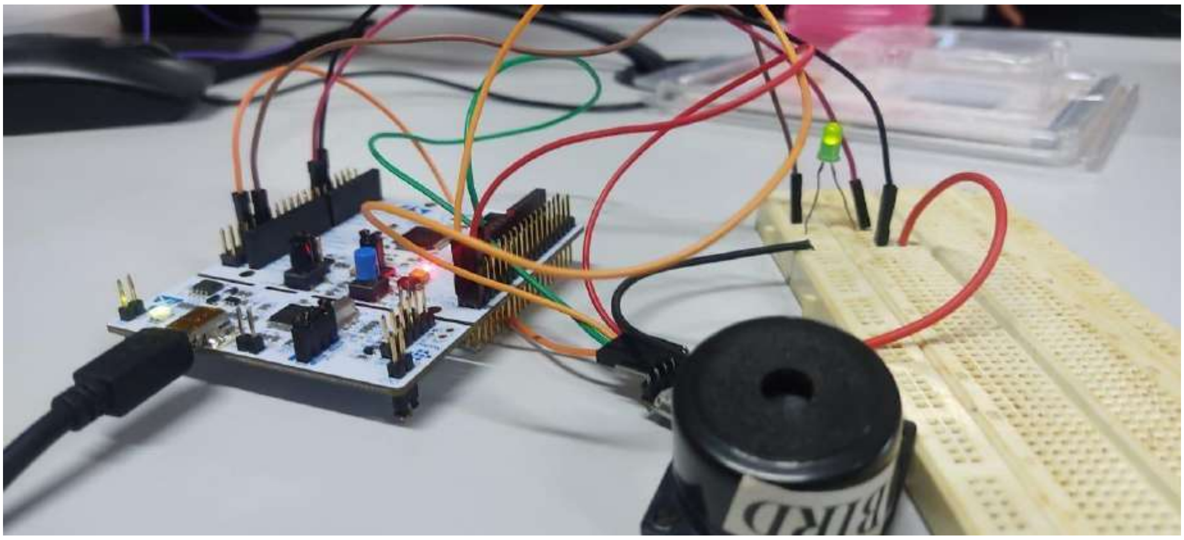
```
#include "mbed.h"
Serial pc(USBTX,USBRX);
Serial bt(PC_10,PC_11);
DigitalOut led(PB_14);
DigitalOut motor(PB_13);
int main(void){
    char ch;
    bt.baud(9600);
    pc.baud(9600);
    pc.printf("Enter the Option:");
    while(1){
        if(bt.readable()){
            ch=bt.getc();
            if (ch=='2'){
                led=1;
                pc.printf("LED ON\n");
            }
            if (ch=='1'){
                led=0;
                pc.printf("LED OFF\n");
            }
            if (ch=='3'){
                motor=1;
                pc.printf("MOTOR ON\n");
            }
            if (ch=='4'){
                motor=0;
                pc.printf("MOTOR OFF\n");
            }
        }
    }
}
```

Output-



```
MOTOR ON
        MOTOR OFF
                 MOTOR OFF
                          LED ON
                                MOTOR OFF
                                         LED ON
                                               MOTOR OFF
                                                        LED ON
                                                              MOTOR OFF
                                                                       LED ON
                                                                             MOTOR OFF
                                                                                      LED ON
                                                                                            MOTO
R OFF
     LED ON
           MOTOR OFF
                    LED ON
```



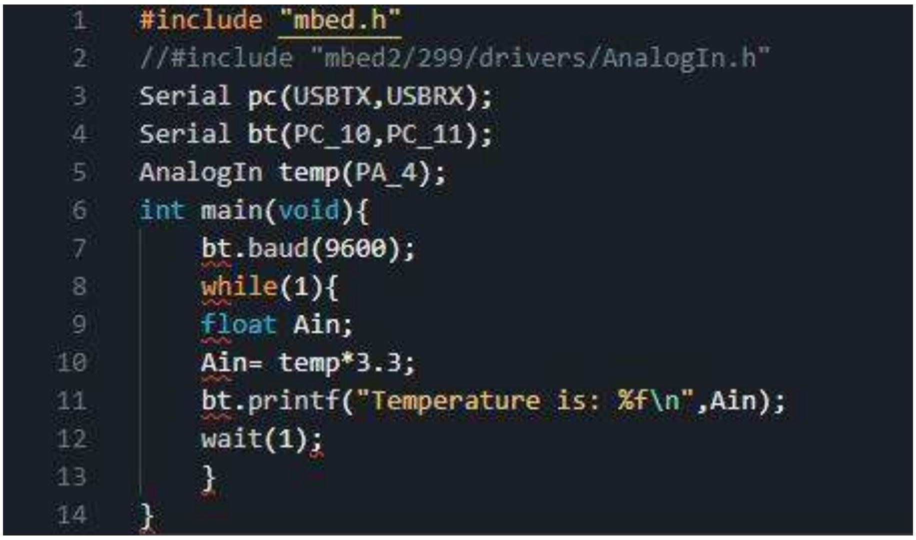


TASK – 3(Challenging task)

Wireless temperature transfer

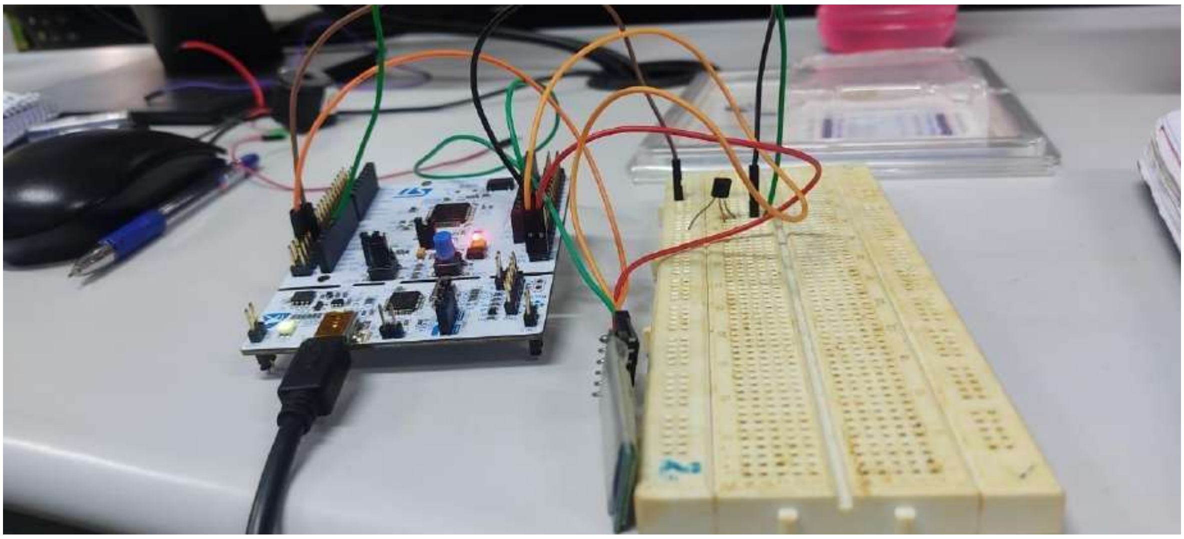
Write a C++ code with mbed APIs to transfer room temperature value measure through LM35 wirelessly using Bluetooth module to the smartphone. Implement this logic on Nucleo board and HC-05 Bluetooth module using Keil studio cloud online compiler.

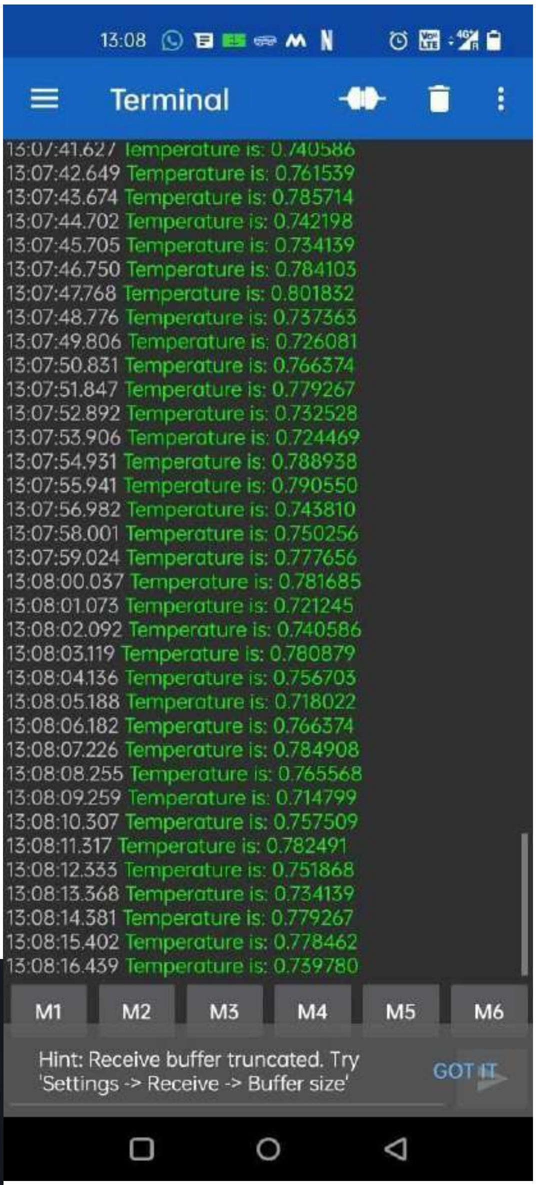
Code-



```
#include "mbed.h"
//#include "mbed2/299/drivers/AnalogIn.h"
Serial pc(USBTX,USBRX);
Serial bt(PC_10,PC_11);
AnalogIn temp(PA_4);
int main(void){
    bt.baud(9600);
    while(1){
    float Ain;
    Ain= temp*3.3;
    bt.printf("Temperature is: %f\n",Ain);
    wait(1);
    }
}
```

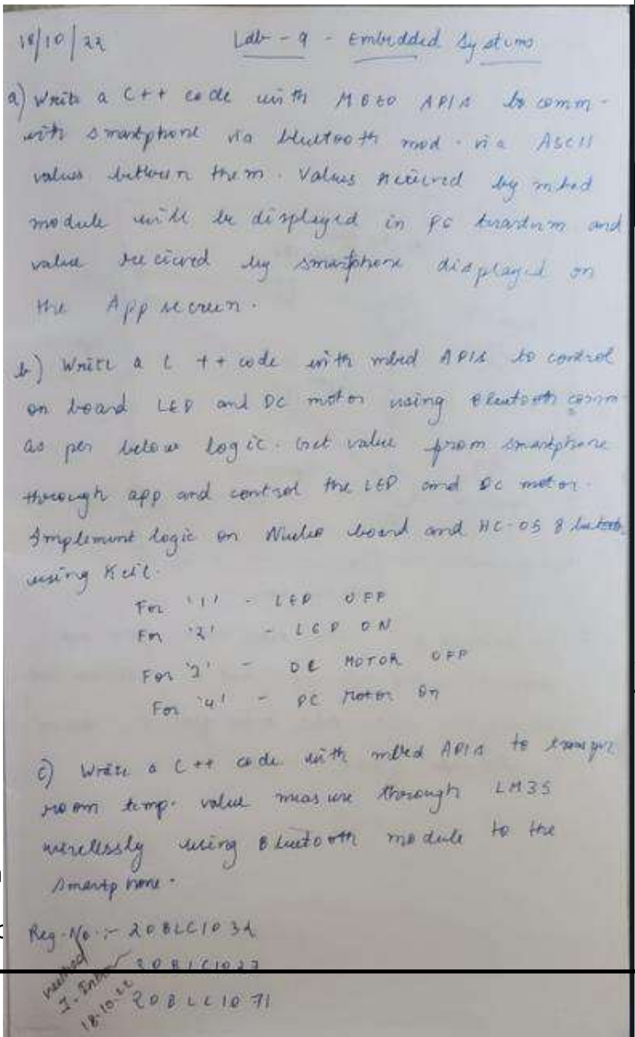
Output-





```
13:07:41.627 Temperature is: 0.740586
13:07:42.649 Temperature is: 0.761539
13:07:43.674 Temperature is: 0.785714
13:07:44.702 Temperature is: 0.742198
13:07:45.705 Temperature is: 0.734139
13:07:46.750 Temperature is: 0.784103
13:07:47.768 Temperature is: 0.801832
13:07:48.776 Temperature is: 0.737363
13:07:49.806 Temperature is: 0.726081
13:07:50.831 Temperature is: 0.766374
13:07:51.847 Temperature is: 0.779267
13:07:52.892 Temperature is: 0.732528
13:07:53.906 Temperature is: 0.724469
13:07:54.931 Temperature is: 0.788938
13:07:55.941 Temperature is: 0.790550
13:07:56.982 Temperature is: 0.743810
13:07:58.001 Temperature is: 0.750256
13:07:59.024 Temperature is: 0.777656
13:08:00.037 Temperature is: 0.781685
13:08:01.073 Temperature is: 0.721245
13:08:02.092 Temperature is: 0.740586
13:08:03.119 Temperature is: 0.780879
13:08:04.136 Temperature is: 0.756703
13:08:05.188 Temperature is: 0.718022
13:08:06.182 Temperature is: 0.766374
13:08:07.226 Temperature is: 0.784908
13:08:08.255 Temperature is: 0.765568
13:08:09.259 Temperature is: 0.714799
13:08:10.307 Temperature is: 0.757509
13:08:11.317 Temperature is: 0.782491
13:08:12.333 Temperature is: 0.751868
13:08:13.368 Temperature is: 0.734139
13:08:14.381 Temperature is: 0.779267
13:08:15.402 Temperature is: 0.778462
13:08:16.439 Temperature is: 0.739780
```

Hint: Receive buffer truncated. Try 'Settings -> Receive -> Buffer size'



18/10/22 Lab - 9 - Embedded Systems

a) Write a C++ code with Mbed APIs to comm- with smartphone via bluetooth mod. via ASCII values between them. Values received by mbed module will be displayed in PC terminal and value received by smartphone displayed on the App screen.

b) Write a C ++ code with mbed APIs to control on board LED and DC motor using Bluetooth comm as per below logic. Get value from smartphone through app and control the LED and DC motor. Implement logic on Nucleo board and HC-05 Bluetooth using Keil.

For '1' - LED OFF
For '2' - LED ON
For '3' - DC MOTOR OFF
For '4' - DC motor on

c) Write a C++ code with mbed APIs to transfer room temp. value measure through LM35 wirelessly using Bluetooth module to the smartphone.

Reg-No:- 20BLC1032
20BLC1023
20BLC1071

verified
I. Enkh
18.10.22

Verification-

Inference-

1. Get familiar with the fundamental capabilities and features of the Mbed Nucleo Board and Mbed Online Compiler.

2. Acquired knowledge of Bluetooth HC05 Module.

3. Learned how to interface the Nucleo64-STM32L152 board to the Bluetooth module and use it for various other tasks via Bluetooth connection to a mobile device.

Result-

Interfacing a Bluetooth module to the Nucleu64-STM32L152 board has been studied and programs were written for the specified tasks successfully.

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B.Tech – Electronics & Computer Engineering

ECE4003 – EMBEDDED SYSTEM DESIGN LAB RECORD

Submitted By

20BLC1027 – Shyaam S

Submitted To

Dr. SOFANA REKA S

Embedded System

Design

Experiment-10

Aim- Interfacing accelerometer and ultrasonic sensor with STM NUCLEO L152RE and performing several operations

Software Used-

Arm MBED online Compiler, Tera Term, Bluetooth Terminal HC-05 App

Hardware Used-

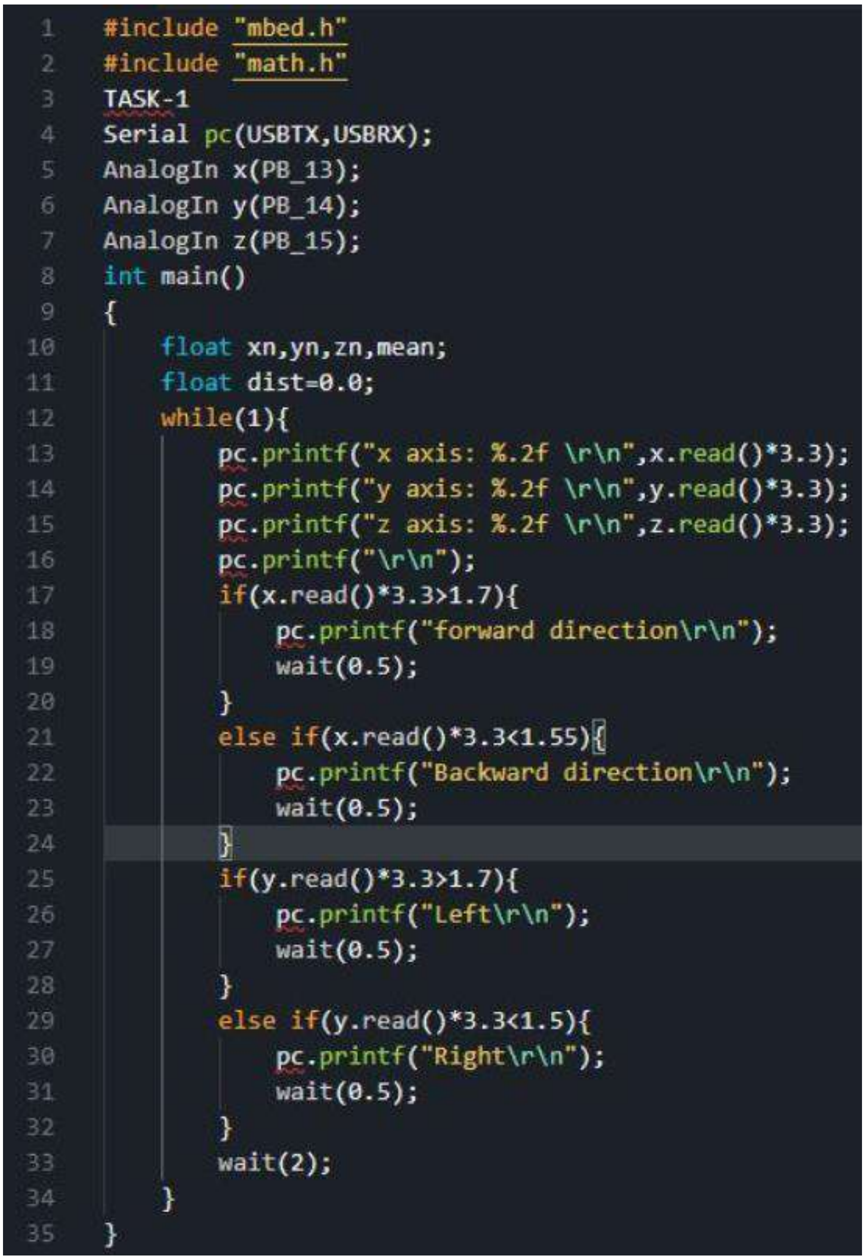
Micro USB cable, STM NUCLEO L152RE Development Board, accelerometer, HC-05 Bluetooth module, ultrasonic sensor, Bread Board, buzzer, Jumper Wires

Procedure-

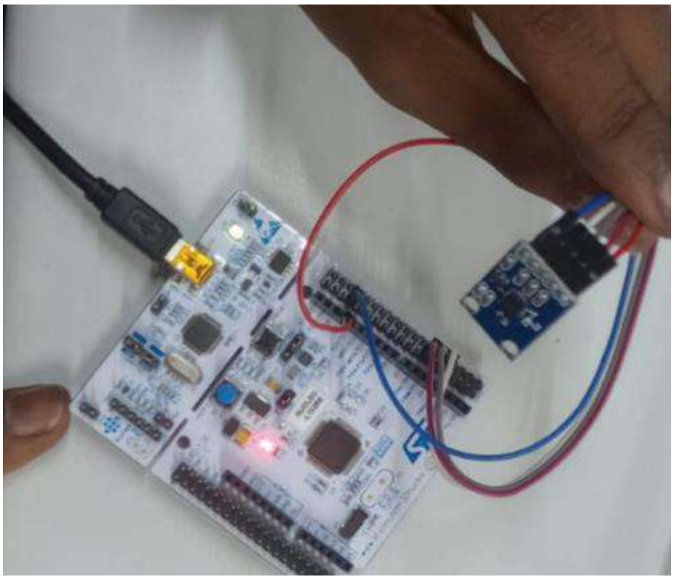
1. Log-in into the MBED website by creating an account
2. Set-up the STM NUCLEO L152RE Board.
3. Create A New-Project with template either Blinky LED Hello World or a Blank Template.
4. In the main.cpp file write/modify the code as per the requirements.
5. Once done save and compile the code.
6. Once Compiled you will be prompted to download the bin file.
7. Once the bin file is downloaded place it into the memory of the board.
8. Press Reset Button on the Board.
9. The code will now start its Execution.

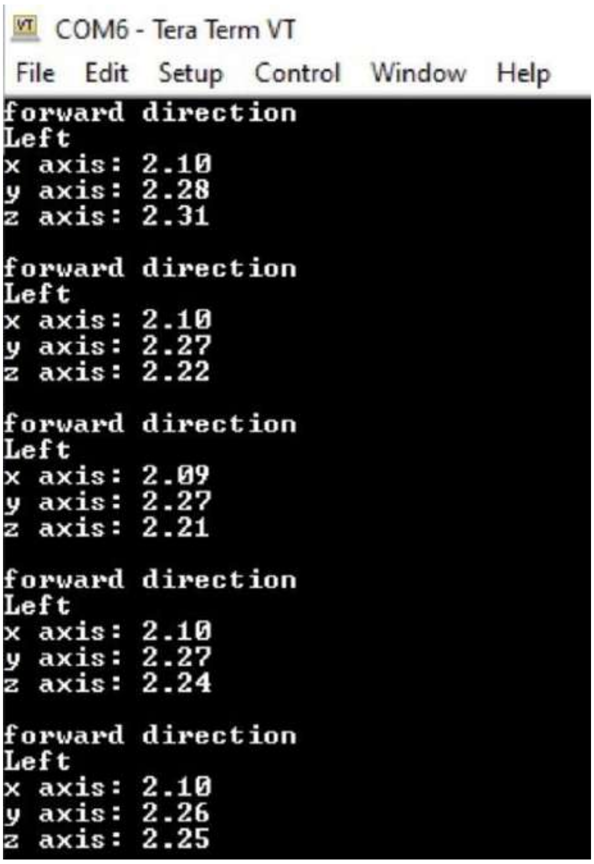
Tasks-

Task1 – **Write a C++ code with mbed APIs to sense the direction of tilt of the fitness tracker and display it on the serial monitor in pc.**



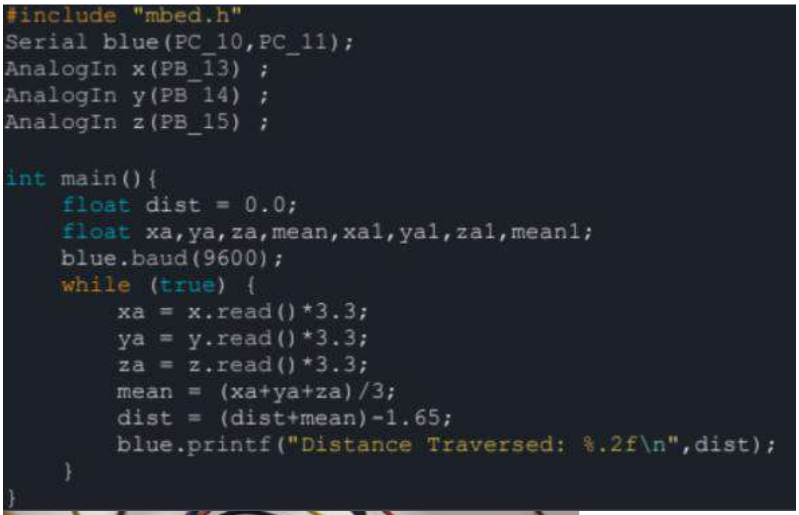
```
#include "mbed.h"
#include "math.h"
TASK-1
Serial pc(USBTX,USBRX);
AnalogIn x(PB_13);
AnalogIn y(PB_14);
AnalogIn z(PB_15);
int main()
{
    float xn,yn,zn,mean;
    float dist=0.0;
    while(1){
        pc.printf("x axis: %.2f \r\n",x.read()*3.3);
        pc.printf("y axis: %.2f \r\n",y.read()*3.3);
        pc.printf("z axis: %.2f \r\n",z.read()*3.3);
        pc.printf("\r\n");
        if(x.read()*3.3>1.7){
            pc.printf("forward direction\r\n");
            wait(0.5);
        }
        else if(x.read()*3.3<1.55){
            pc.printf("Backward direction\r\n");
            wait(0.5);
        }
        if(y.read()*3.3>1.7){
            pc.printf("Left\r\n");
            wait(0.5);
        }
        else if(y.read()*3.3<1.5){
            pc.printf("Right\r\n");
            wait(0.5);
        }
        wait(2);
    }
}
```



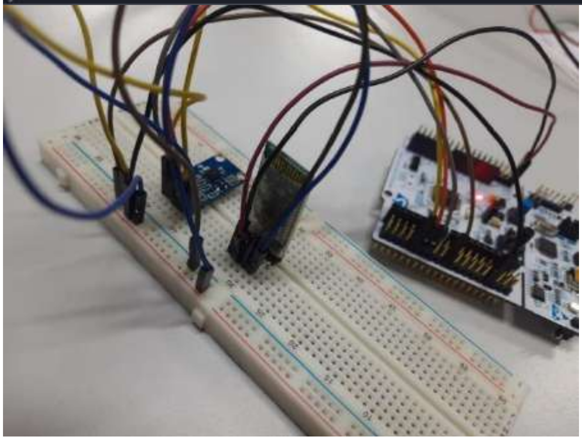


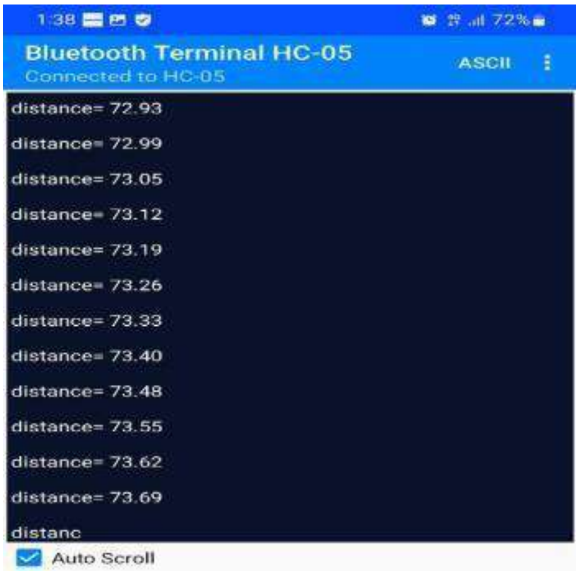
```
forward direction
Left
x axis: 2.10
y axis: 2.28
z axis: 2.31

forward direction
Left
x axis: 2.10
y axis: 2.27
z axis: 2.22

forward direction
Left
x axis: 2.09
y axis: 2.27
z axis: 2.21

forward direction
Left
x axis: 2.10
y axis: 2.27
z axis: 2.24

forward direction
Left
x axis: 2.10
y axis: 2.26
z axis: 2.25
```

<u>Task2 –</u> Write a C++ code with mbed APIs to read the values of the accelerometer and measure the distance travelled by it and display it on phone using Bluetooth module.

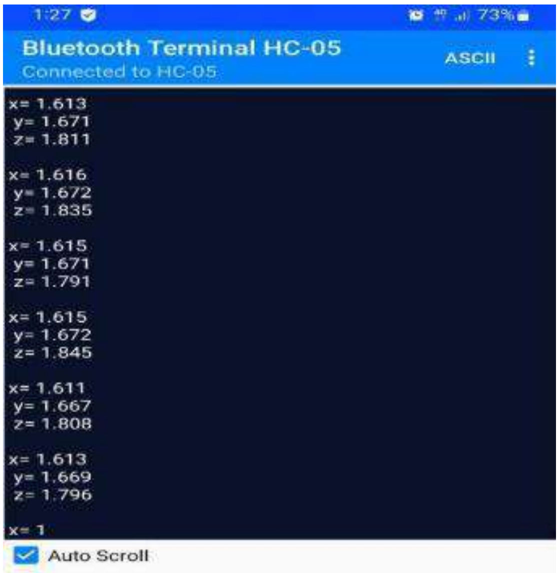


```
#include "mbed.h"
Serial blue(PC_10,PC_11);
AnalogIn x(PB_13) ;
AnalogIn y(PB_14) ;
AnalogIn z(PB_15) ;

int main(){
    float dist = 0.0;
    float xa,ya,za,mean,xa1,ya1,za1,mean1;
    blue.baud(9600);
    while (true) {
        xa = x.read()*3.3;
        ya = y.read()*3.3;
        za = z.read()*3.3;
        mean = (xa+ya+za)/3;
        dist = (dist+mean)-1.65;
        blue.printf("Distance Traversed: %.2f\n",dist);
    }
}
```



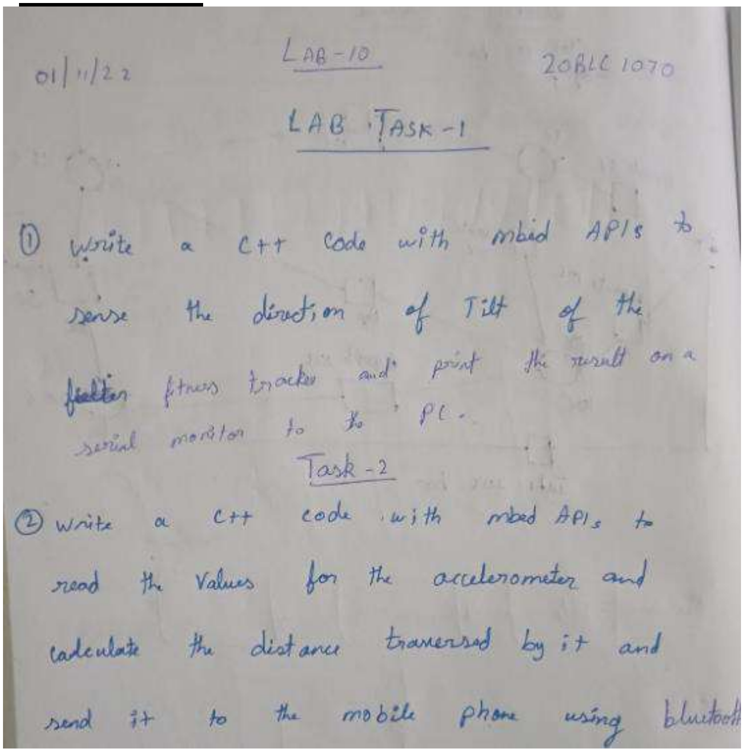


```
distance= 72.93
distance= 72.99
distance= 73.05
distance= 73.12
distance= 73.19
distance= 73.26
distance= 73.33
distance= 73.40
distance= 73.48
distance= 73.55
distance= 73.62
distance= 73.69
distanc
```



```
x= 1.613
 y= 1.671
 z= 1.811

x= 1.616
 y= 1.672
 z= 1.835

x= 1.615
 y= 1.671
 z= 1.791

x= 1.615
 y= 1.672
 z= 1.845

x= 1.611
 y= 1.667
 z= 1.808

x= 1.613
 y= 1.669
 z= 1.796

x= 1
```

Verification



01/11/22

LAB-10

20BLC1070

LAB TASK-1

① write a C++ Code with mbed APIs to sense the direction of Tilt of the fealter fitness tracker and print the result on a serial monitor to the PC.

Task-2

② write a C++ code with mbed APIs to read the Values for the accelerometer and calculate the distance traversed by it and send it to the mobile phone using bluetooth

Inference

1.) Learnt to interface HC-05 Bluetooth module, accelerometer with STM NUCLEO L152RE board.

Result

Hence, the given tasks have been implemented using STM NUCLEO L152RE Development Board.