INDEX

Name : Abhishek . S. Badukale Class Roll No.: 28

Branch : CSE Year / Semester : 3rd year / 6th sem

Name of Subject : Hardware Lab

S.No.	Name of Experiment	Page No.	Date	Remarks
1.	Study of eIOT 3500 development kit and familarization with Intel Galileo board and perform necessary software installation.		9/1/24	✓ map 16/01/24 A+
2.	To interface LED with Intel Galileo board and write a program to turn ON LED after every one sec interval.		16/1/24	✓ map 23/01/24 A+
3.	To interface push button with Intel galileo board and write a program to turn on LED when push button is pressed.		23/1/24	✓ map 30/01/24 A+
4.	To interface buzzer with intel galileo board and write a program to buzz on and off with a delay of one second.		30/1/24	✓ map 06/02/24 A+
5.	To interface temperature sensor with Intel galileo board and write a program to print temperature readings.		6/2/24	✓ map 13/02/24 A+

INDEX

Name : Abhishek Badukale Class Roll No.: 28

Branch : CSE Year / Semester : 3rd year / 6th sem

Name of Subject : Hardware lab

S.No.	Name of Experiment	Page No.	Date	Remarks
6.	To interface touch sensor with intel board and program to sense a finger when it placed.		13/2/24	Cmap 05/03/24 A+
7.	To interface light sensor with intel board and a program to detect surrounding light intensity.		5/3/24	Cmap 12/03/24 A+
8.	To interface sound sensor with intel board and program to detect surrounding sound intensity.		12/3/24	Cmap 19/03/24 A+
9.	To interface LED display with intel board and write a program to display some text.		19/3/24	Cmap 02/04/24 A+
10.	To interface rotary angle sensor/ potentiometer with the Galileo board and write a program to control the readings of sensor of changing the angle of potentiometer.		2/4/24	Cmap 05/04/24 A+

PAGE NO.:
DATE: 9 / 1 / 2024

Practical no. 1

- Aim: Study of IOT 3500 development kit and familiarization with intel Galileo board and perform necessary software installation.

- Tools:

Sr. No.	Tools	Specification	Qty
1	IOT 3500 Dev kit	Intel Galileo kit	1
2	Computer System	4GB+RAM 1TB HDD	1
3	Bootable pendrive	32 GB	1

- Theory: Steps to configure Intel Galileo Board.

1] Assemble the Hardware from IOT kit. Unbox Intel Galileo & connect Galileo Board to power through adapter given.

2] Takeout the sandisk 32 GB pendrive that is already loaded with Galileo linux supportable OS username as IOT. attach it to computer. restart your computer as IOT, and select pendrive in 1st position of Boot order in BIOS settings.

3] Save the settings, now OS will load into your computer. login by entering password iot123.

4] All necessary softwares are already installed in this OS, now go to terminal and write following command:

Teachers Signature _____

PAGE NO.:
DATE.: / / 20

make sure to attach Galileo to computer using SDIF to USB port cable.

5] Power ON Galileo and write command on OS terminal:
sudo screen /dev/ttyUSB0 115200

6] This will load Galileo terminal on OS terminal.

7] Now Galileo will require password enter "iot123".

8] Now type 'connmanctl' command in galileo prompt.

9] Type "search wifi" in terminal by pluging "wifi receiver module" to computer, then you will be listed some of SSID of available wifi.

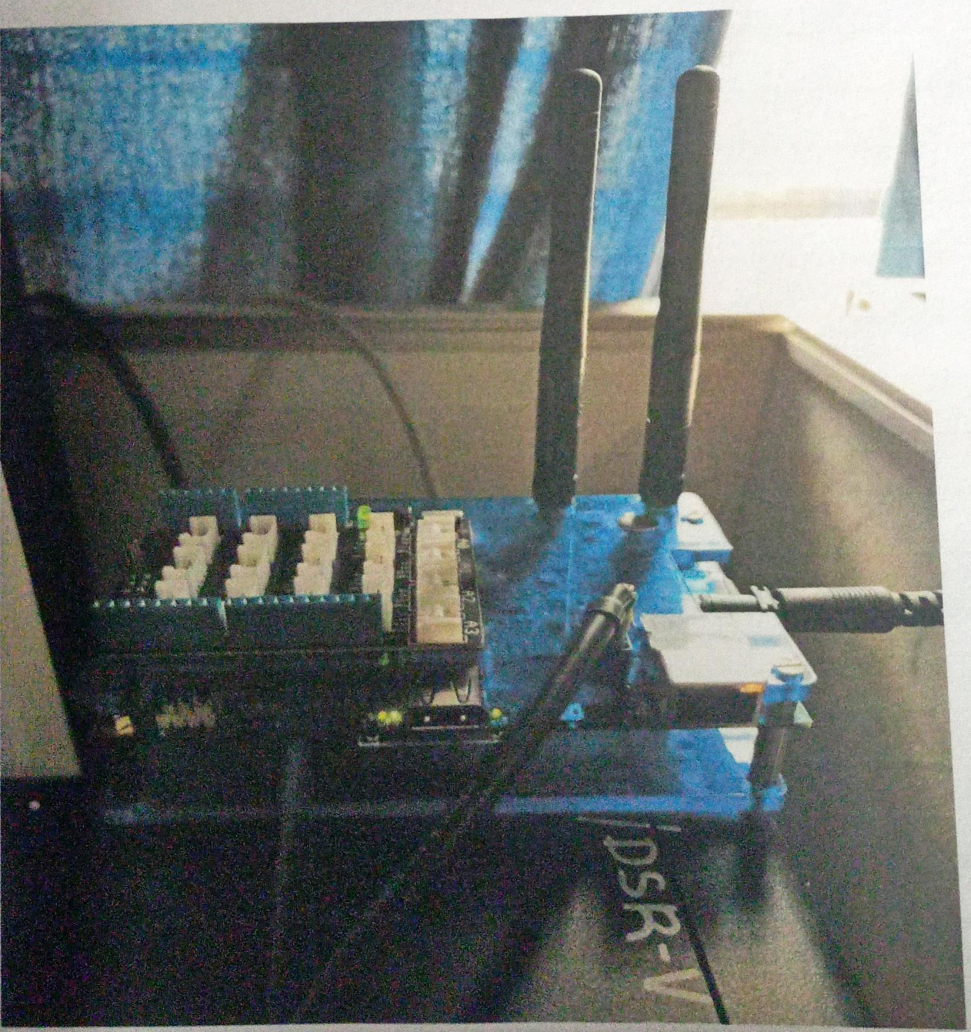
10] Copy id of your wifi, type add wifi wifi SSID & hit enter.

11] Now your galileo board is also connected to internet, exit by removing FTDI cable from computer.

12] Now all configurations has been done.

Teachers Signature _____

```
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          Interrupt:51 Base address:0x4000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:724 errors:0 dropped:0 overruns:0 frame:0
          TX packets:724 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:55172 (53.8 KiB)  TX bytes:55172 (53.8 KiB)

wlp1s0    Link encap:Ethernet  HWaddr 68:17:29:65:85:BA
          inet addr:192.168.1.168  Bcast:192.168.7.255  Mask:255.255.248.0
          inet6 addr: fe80::6a17:29ff:fe65:85ba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:88144 errors:0 dropped:72 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7631454 (7.2 MiB)  TX bytes:14588 (14.2 KiB)

root@galileo:~#
```



PAGE NO.:
DATE.: / / 20

- Program : None
- Conclusion : Study of IOT 3500 development kit & familiarization with intel galileo board & performed necessary software installations successfully.

PAGE NO.:
DATE.: 16 / 1 / 2024

Practical no. 2

Aim : To interface LED with Intel Galileo board and write a program to turn on LED repeatedly after every one sec interval.

Tools :

Sr. no.	Tools	Specifications	Qty
1	Development kit	IOT 3500 kit	1
2	Computer System	4GB + RAM, 1 TB HDD	1
3	LED Display	LD15, for galileo	1

Theory : Steps to implement above statement in galileo board.

1. Plug galileo board to computer & power ON
2. Power computer system, enter passwords.
3. Open 'Eclipse IDE' in linux OS, i.e already installed.
4. All the programs were given in workspace of Eclipse.
5. Click on create connection → new connection → enter galileo ip address & name your connection.
6. Click next → right click on connection → click start connection → connection will now change to green.
7. Write appropriate program and run by clicking on run button.
8. Write command to interface LED to galileo.

Teachers Signature _____

Code: -

```
#include "mraa.h"

#include <stdio.h>
#include <unistd.h>

int main()
{
 mraa_gpio_context d_pin = NULL;
 d_pin = mraa_gpio_init(13);
 if(d_pin == NULL){ //error check
 fprintf(stderr, "MRAA couldn't initialized.");
 return MRAA_ERROR_UNSPECIFIED;
}
// configure the GPIO pin as output
 if(mraa_gpio_dir(d_pin, MRAA_GPIO_OUT != MRAA_SUCCESS){
 fprintf("Can't set digital pin as output, existing");
 return MRAA_ERROR_UNSPECIFIED;
};
//loop to toggle on-board every second for 10 times
for(int i=10;i>0;i--){
 printf("LED OFF\n");
 mraa_gpio_write(d_pin,0); //turn off LED
 sleep(1); //wait 1 second
 printf("LED ON\n");
else mraa_gpio_write(d_pin, 1); //turn on LED
sleep(1); //wait 1 second
}
return MRAA_SUCCESS;
}
```

Output: -

```
Problems  Tasks  Console  Properties
<terminated> Blinky [C/C++ Remote Application] /home/iot/Downloads/exercises_code/
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
LED ON
logout
```

PAGE NO.:
DATE.: / / 20

9. Upload the code to Galileo and click on run.

After upload completes, you should see a tiny green LED blinking on and off every second. This LED is connected to pin 13 on galileo.

- Results: Compilation and installation of LED after every one second is done.

- Conclusion: Target board of such types can be used in low cost system design using very less amount of components and can be used for many user defined application.

23/11/24 (A1)

PAGE NO.:
DATE.: 23/ 1 / 2024

Practical no. 3

- Aim : To interface push button with intel galileo board and write program to turn on LED when push button is pressed.

- Tools :-

Sr no.	Tools	Specification	Qty
1	IOT development kit	IOT 3500	1
2	Computer system	4GB + RAM / 1 TB HDD	1
3	LED	LD15	1

- Theory : Steps to interface push button

material used :

1) Intel Galileo board 2) Push Button
3) LED 4) Resistor 5) Connecting wires

Hardware Setup Steps :-

1] Connect one terminal of push button to a digital input pin (eg P_2) on galileo board.
2] Connect jumper wire from P2 on galileo board to some row on push button.
3] Connect LED to galileo board using another connecting wires.
4] Power ON galileo and connect with computer system.
5] Open Eclipse and connect to your galileo connection.

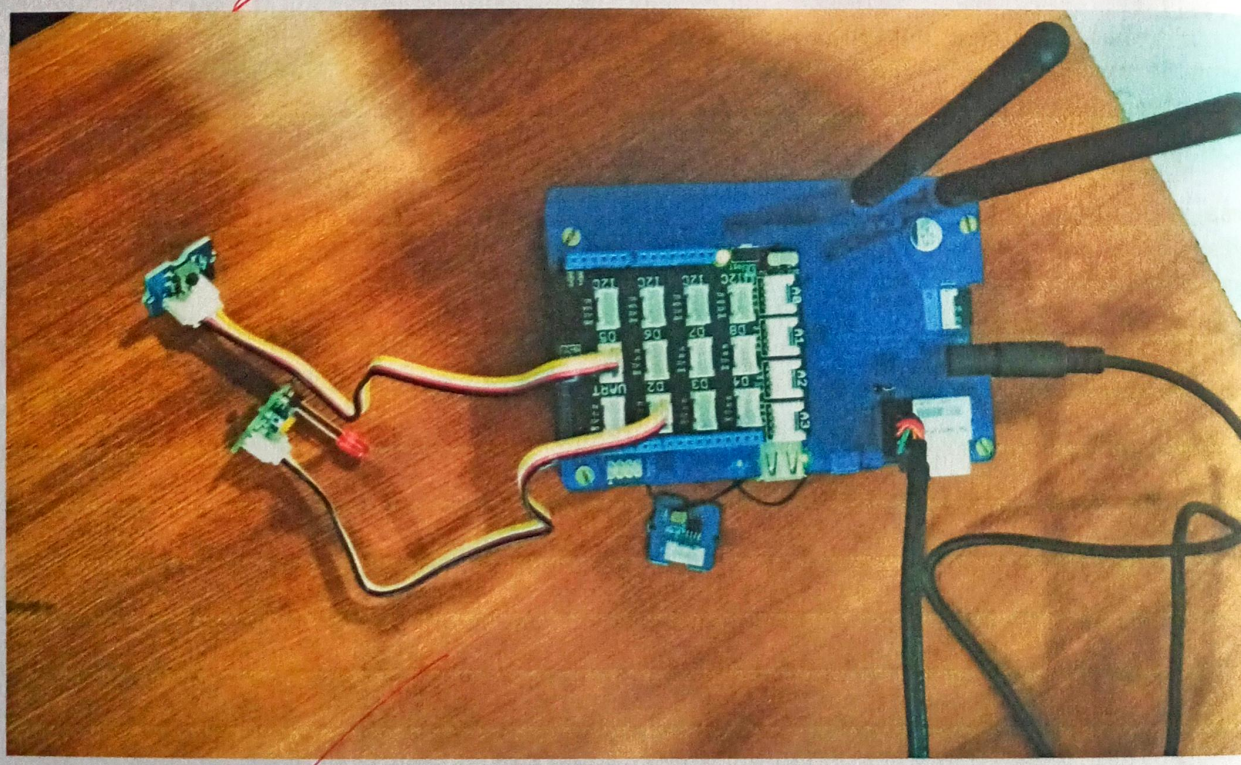
Teachers Signature _____

Code: -

```c
#include "mraa.h"
#include <stdio.h>
#include <unistd.h>

#define BUTTON_PIN 2
#define LED_PIN 13

int main()
{
    mraa_init();
    mraa_gpio_context button = mraa_gpio_init(BUTTON_PIN);
    mraa_gpio_dir(button, MRAA_GPIO_IN);

    mraa_gpio_context led = mraa_gpio_init(LED_PIN);
    mraa_gpio_dir(led, MRAA_GPIO_OUT);

    for(;;) {
        if (mraa_gpio_read(button)) mraa_gpio_write(led, 1);
        else mraa_gpio_write(led, 0);
        usleep(100000);
    }

    mraa_gpio_close(led);
    mraa_gpio_close(button);
    mraa_deinit();

    return MRAA_SUCCESS;
}
```

Output: -

```
Problems  Tasks  Console  Properties
<terminated> Button [C/C++ Remote Application] /home/iot/Downloads/ex
root@galileo:~# chmod 755 /tmp/Button;/tmp/Button ;exit
Program will exit after 5 button presses
Button value is: 1
 Program will exit after 4 button presses
Button value is: 0
 Program will exit after 4 button presses
Button value is: 0
 Program will exit after 4 button presses
Button value is: 0
 Program will exit after 4 button presses
Button value is: 0
 Program will exit after 4 button presses
Button value is: 1
 Program will exit after 3 button presses
Button value is: 1
 Program will exit after 2 button presses
Button value is: 1
 Program will exit after 1 button presses
Button value is: 1
 Exiting, bye!logout
```



PAGE NO.:
DATE.: / / 20

6] Upload into galileo and click on RUN.

- Result : Given problems statement successfully implemented.

- Conclusion : Program to interface push button to LED to turn ON when button pressed is implemented successfully.

Dmap (A)
30/01/20

PAGE NO.:
DATE.: 30/ 1 / 2024

Practical no. 4

Aim: To interface buzzer with intel galileo board and write a program to buzz on and off with a delay of one second.

Tools :-

Sr. No.	Tools	Qty
1	IOT 3500 Dev Kit	1
2	Computer System	1
3.	Buzzer connecting wires	1
4.	Eclipse IDE	1

Theory :-

Steps to interface buzzer with Intel galileo:

- Material used: Buzzer , Connecting wires, galileo board.
- Hardware setup:

1) Connect Buzzer to galileo board using connecting wire to digital input pin.
2) Connect power adapter of board and turn it on.
3) Open Eclipse IDE in computer system.
4) Connect to Galileo connection.
5) Write code in new line.
6) Upload code in Galileo using console.
7) Click on RUN button.

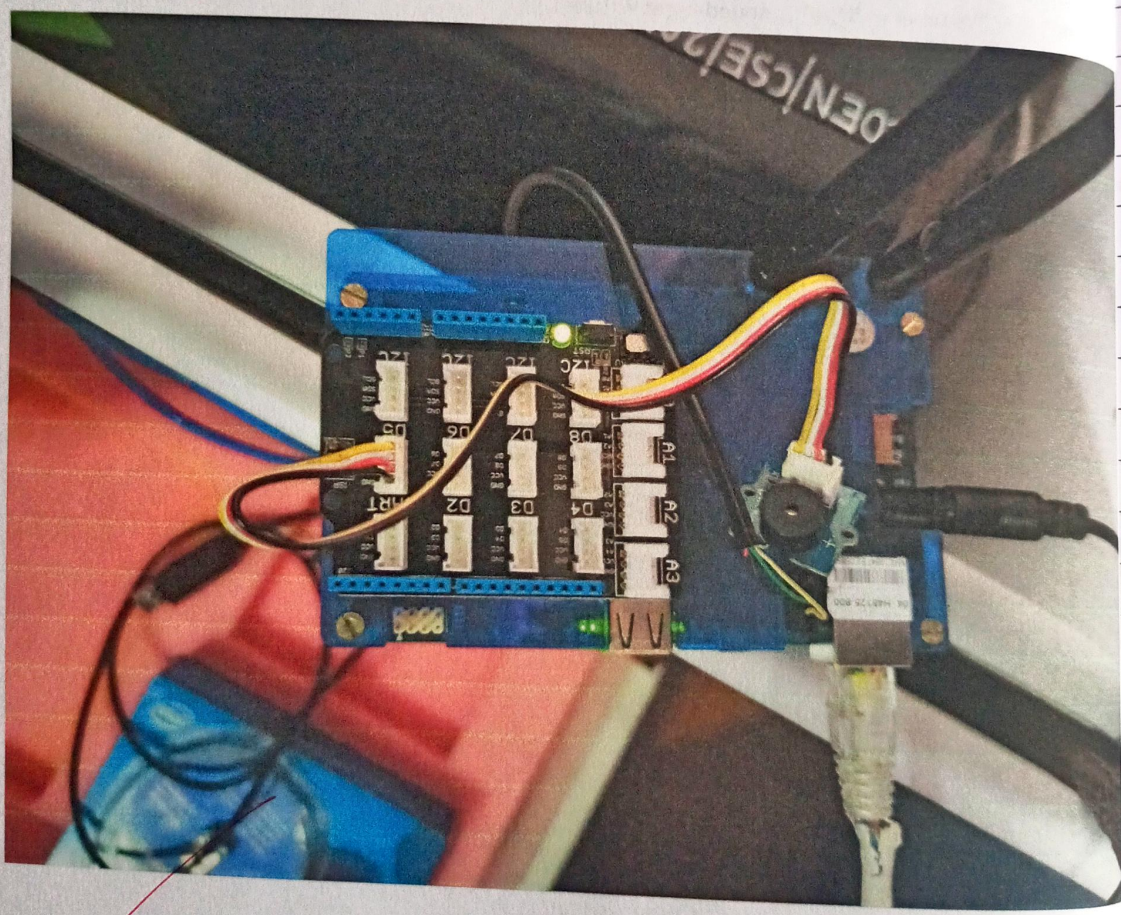
Teachers Signature _____

Code: -

```
#include <buzzer.hpp>
#include <stdio.h> // for printf()

#include <unistd.h> //for sleep()

int main()
{
    int chord[] = { DO, RE, FA, MI, SOL, LA, SI, DO, SI};
        // create Buzzer instance
        upm::Buzzer* sound = new upm::Buzzer(5);

        printf("Volume = %f\n", sound->getVolume());
        sound->setVolume(0.1);
        printf("Volume = %f\n", sound->getVolume());
        fflush(stdout);

        // play sound (DO, RE, MI, etc...), pausing for 0.1 seconds between notes
        printf("\nPlaying notes, pausing for 0.1 seconds between notes...\n");
        fflush(stdout);

        for (int chord_ind = 0; chord_ind < 7; chord_ind++) {
            // play each note for one second
            printf(" %d\n",  sound->playSound(chord[chord_ind], 500000) );
            usleep(100000);
        }
        printf("Exiting, bbye!\n");

        delete sound;
}
```

Output: -

```
Problems  Tasks  Console  Properties
<terminated> Buzzer [C/C++ Remote Application] /home/iot/Downloads/exercises_code
root@galileo:~# chmod 755 /tmp/Buzzer;/tmp/Buzzer ;exit
Volume = 1.000000
Volume = 0.600000

Playing notes, pausing for 0.1 seconds between notes...
 3800
 3400
 3000
 2900
 2550
 2270
 2000
Exiting, bbye!
logout
```



PAGE NO.:
DATE.: / / 20

- Result : Program implemented correctly & problem understood clearly.
- Conclusion : Program to interface buzzer with Intel Galileo board to buzz ON & OFF after delay of 1 second implemented successfully.

(P) map (A+)
06/02/24

PAGE NO.:
DATE: 6 / 2 / 2024

Practical no. 5

- Aim: To interface temperature sensor with intel galileo board and write a program to print temperature readings.

- Tools:

Sr. no.	Tools	Qty
1	IOT 3500 Development Kit	1
2	Computer System	1
3	Temperature Sensor	1

- Theory:

Steps to interface temperature sensor with Intel Galileo board.

- Material used: Temperature Sensor, connecting wires, galileo board

- Hardware / Software Setup:-

1) Connect temperature sensor to Galileo board by using connecting wires to input pin (eg. A75) to read analog values.
2) Power ON Galileo board, connect it to computer system.
3) Open Eclipse IDE in computer, connect to connect, write code in new file.
4) Mention temperature sensor pin no. on program.

Teachers Signature _____

Code: -

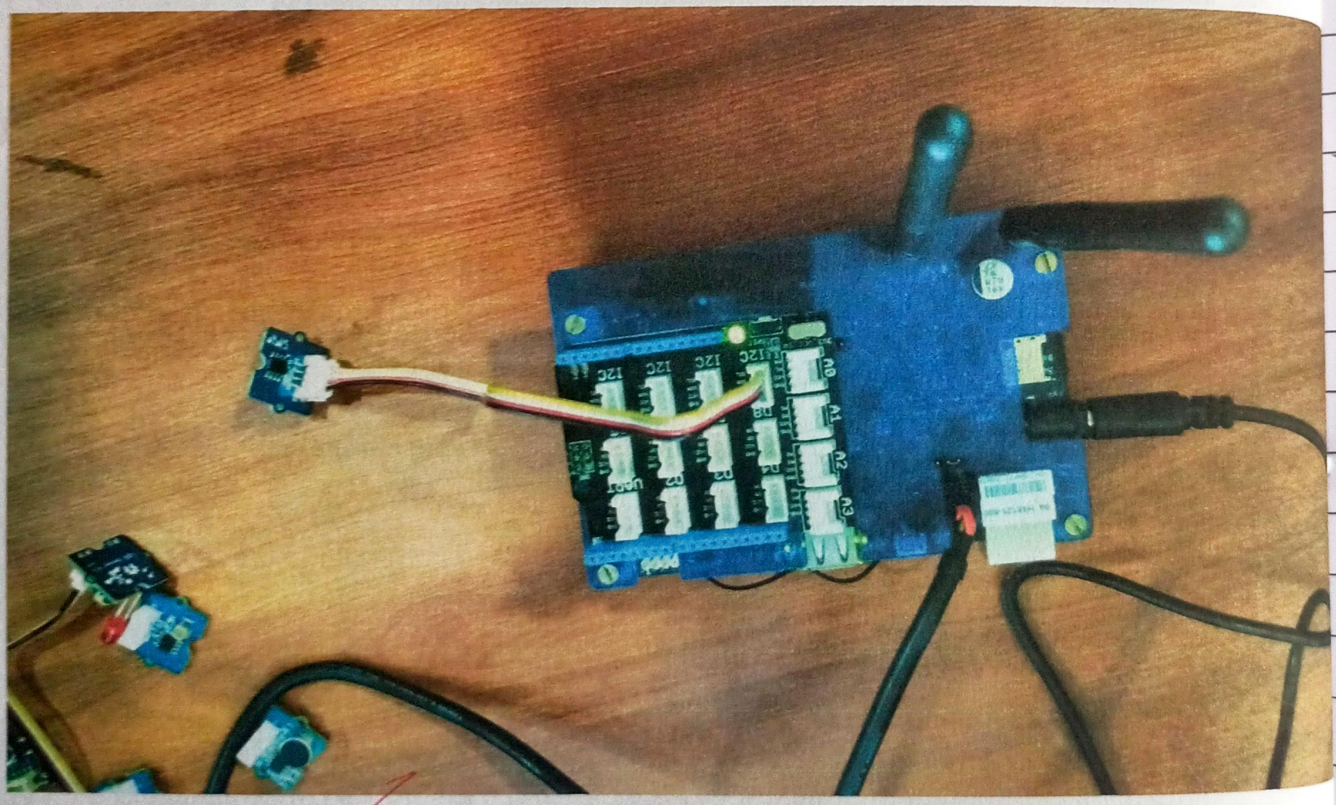
```
#include "mraa/aio.h" //for mraa_aio_read()

#include <math.h> // for math functions

#include <stdio.h> // for printf()

#include <unistd.h> //for sleep()
int main()
{
    mraa_aio_context adc_a0;
    uint16_t adc_value = 0;

    const int B=4275;                 // B value of the thermistor
    const int R0 = 100000;            // R0 = 100k

    //create object of analog input class using mraa lib
    adc_a0 = mraa_aio_init(0);

    if (adc_a0 == NULL) {
        return 1;
    }
    for (int i=10; i>0;i--) {
        adc_value = mraa_aio_read(adc_a0); //Max value @ 5V = 1024
        printf("ADC A0 read value : %d\n", adc_value);

        float R = 1023.0/((float)adc_value)-1.0;
        R = 100000.0*R;
        float temperature=1.0/(log(R/100000.0)/B+1/298.15)-273.15;//convert to temperature
as per datasheet ;

        printf("Temperature value : %.2f Degree Celsius\n", temperature);

        sleep(3);
    }
    mraa_aio_close(adc_a0);
    printf("Exiting .. Bbye!");
    return MRAA_SUCCESS;
}
```

Output: -

Problems Tasks Console Properties

<terminated> Temperature [C/C++ Remote Application] /home/iot/Downloads/exercises

```
Temperature value : 29.01 Degree Celsius
ADC A0 read value : 548
Temperature value : 28.00 Degree Celsius
ADC A0 read value : 558
Temperature value : 28.84 Degree Celsius
ADC A0 read value : 554
Temperature value : 28.50 Degree Celsius
ADC A0 read value : 556
Temperature value : 28.67 Degree Celsius
ADC A0 read value : 559
Temperature value : 28.92 Degree Celsius
ADC A0 read value : 610
Temperature value : 33.34 Degree Celsius
ADC A0 read value : 612
Temperature value : 33.52 Degree Celsius
Exiting .. Bbye!logout
```



PAGE NO.:
DATE.: / / 20

5) Upload the code to galileo board.
6) Click on RUN.

- Conclusion :- Program to interface temperature sensor with Galileo board to read temperature and print it implemented successfully.

13/02/24

PAGE NO.:
DATE.: 13 / 2 / 2024

Practical no. 6

Aim : To interface touch sensor with intel galileo board & write a program to sense a finger when it is placed on intel galileo board.

Tools :-

01	IOT 3500 Development Kit	1
02	Computer system	1

Theory :-

Steps to interface touch sensor with Galileo.

- Materials used from Kit :- Touch sensor, connecting wires, Galileo board.

- Hardware setup :-

1. Connect touch sensor to galileo board using connecting wires to input Analog pin of Galileo board.
2. Power ON Galileo, connect to computer system.
3. Open Eclipse IDE on computer, connect to existing connection.
4. Write program on new file.
5. Save the program.
6. Upload to galileo board using console commend.

Teachers Signature

Code: -

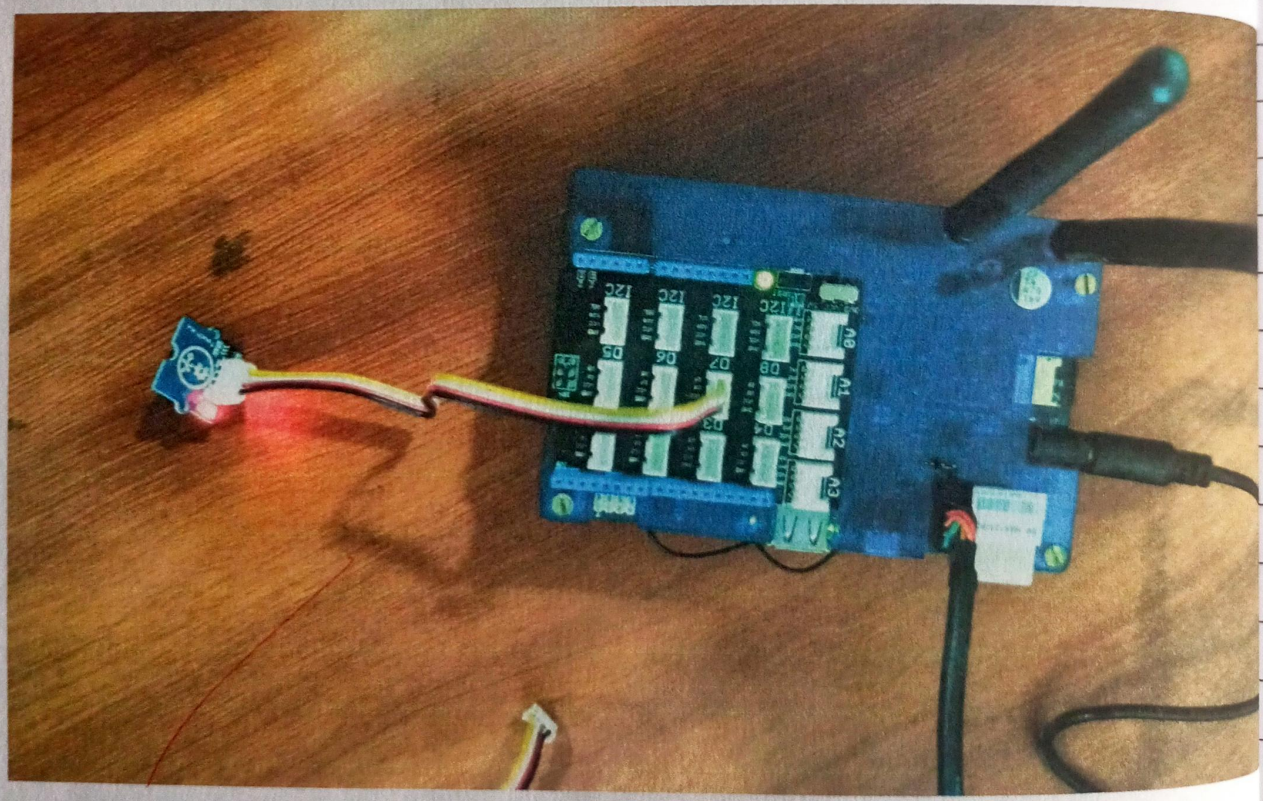
```
/* UPM library includes */
#include "ttp223.h" // for button->value()

/* Standard IO includes */
#include <stdio.h> // for printf()

#include <unistd.h> //for sleep()

void touchISR (void*);
int count = 5;

void touchISR(void*)
{
    count--;
    printf("\nHello World from ISR, will exit after %d touch events", count);
    fflush(stdout);
}

int main()
{
    // Create the button object using GPIO pin 4
    upm::TTP223* touch = new upm::TTP223(4);
    // Read the input and print, waiting one second between readings

    touch->installISR(mraa::EDGE_FALLING, &touchISR, NULL);

    printf("\nWelcome, waiting for touch event.\nWill exit after 5 events");
    fflush(stdout);

    while(count>0);

    printf("\nExiting .. Bbye!");

    delete touch;      // Delete the button object

}
```

Output: -

Problems Tasks Console Properties

<terminated> TouchInterrupt [C/C++ Remote Application] /home/iot/Downloads/exercis

```
root@galileo:~# chmod 755 /tmp/TouchInterrupt;/tmp/TouchInterrupt ;exit

Welcome, waiting for touch event.
Will exit after 5 events

Hello World from ISR, will exit after 4 touch events

Hello World from ISR, will exit after 3 touch events

Hello World from ISR, will exit after 2 touch events

Hello World from ISR, will exit after 1 touch events

Hello World from ISR, will exit after 0 touch events
Exiting .. Bbye!logout
```



PAGE NO.:
DATE.: / / 20

7. Install related package
   Enter = pkg update
   pkg install ize-tools
   ize detect -r -y < izc_bus_number >
8. Click on RUN
9. Check the output on galileo board.

- Conclusion : Program to interface touch sensor with galileo board & sense finger if it is placed on galileo board implemented successfully.

05/03/24 (A+)

Practical no. 7

PAGE NO.:
DATE.: 5 / 3 / 2024

- Aim : To interface light sensor with galileo board & write a program to detect surrounding light intensity.

- Tools :-
1. Intel 3500 Development Kit 1
2. Computer system 1
3. light sensor 1

- Theory :-
Steps to interface light sensor to galileo

- Materials required : light sensor, Galileo board, connecting wires.

- Hardware Setup :
1. Connect light sensor to galileo board. using connecting wire to pin input analog.
2. Power on galileo board, connect it to computer system using FTDI to USB Cable.
3. Open Eclipse IDE in computer connect to existing connection user.
4. Write program in new file of eclipse, upload it to galileo board after interfacing light sensor using console command.

Teachers Signature _____

Code: -

```
#include "grove.h" //for light->name(), light->value()

/* Standard IO includes */
#include <stdio.h> // for printf()

#include <unistd.h> //for sleep()

int main()
{
      // Create the light sensor object using AIO pin 0
        upm::GroveLight* light = new upm::GroveLight(0);
        // Read the input and print both the raw value and a rough lux value,
        // waiting one second between readings
        for (int i=10;i>0;i--) {
            printf(" Light value is %f which is roughly %d lux \n", light->raw_value(),
light->value());
            fflush(stdout);
            sleep(2);
        }
        // Delete the light sensor object
        printf("Exiting .. bbye!");
        delete light;
}
```

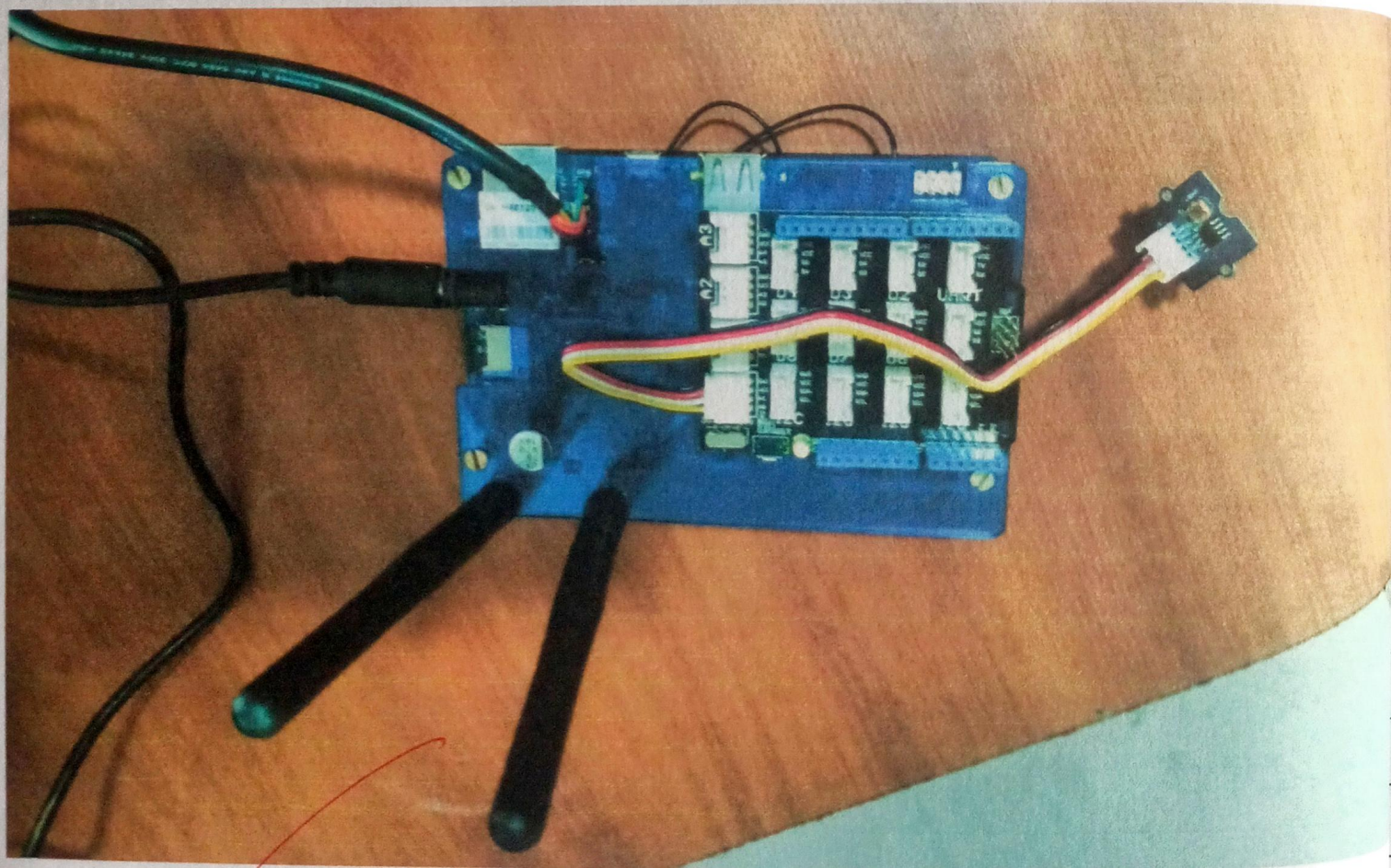
Code: -

```
#include "grove.h" //for light->name(), light->value()

/* Standard IO includes */
#include <stdio.h> // for printf()

#include <unistd.h> //for sleep()

int main()
{
      // Create the light sensor object using AIO pin 0
        upm::GroveLight* light = new upm::GroveLight(0);
        // Read the input and print both the raw value and a rough lux value,
        // waiting one second between readings
        for (int i=10;i>0;i--) {
            printf(" Light value is %f which is roughly %d lux \n", light->raw_value(),
light->value());
            fflush(stdout);
            sleep(2);
        }
        // Delete the light sensor object
        printf("Exiting .. bbye!");
        delete light;
}
```

Problems Tasks Console Properties

<terminated> Light [C/C++ Remote Application] /home/iot/Downloads/exercises_code/l

```
 Light value is 764.000000 which is roughly 53 lux
 Light value is 764.000000 which is roughly 53 lux
 Light value is 764.000000 which is roughly 53 lux
 Light value is 764.000000 which is roughly 53 lux
 Light value is 764.000000 which is roughly 53 lux
 Light value is 655.000000 which is roughly 27 lux
 Light value is 657.000000 which is roughly 27 lux
 Light value is 674.000000 which is roughly 31 lux
 Light value is 764.000000 which is roughly 53 lux
 Light value is 764.000000 which is roughly 53 lux
 Light value is 764.000000 which is roughly 53 lux
 Light value is 450.000000 which is roughly 9 lux
 Light value is 365.000000 which is roughly 6 lux
 Light value is 323.000000 which is roughly 5 lux
 Light value is 306.000000 which is roughly 4 lux
Exiting .. bbye!logout
```



DATE.: / / 20

5. Save file, click on RUN.
6. Measure output in console window.

Conclusion : Program to interface light sensor with galileo board & detect surrounding light intensity implemented successfully.

Vmap (AP)
12/03/24

PAGE NO.:
DATE: 12/3/2024

Practical no. 8

Aim: To interface sound sensor with galileo board & write a program to detect surrounding sound intensity.

Tools :-

01	Intel 3500 Development Kit	1
02	Computer System	1
03	Sound sensor, connecting wires	1

Theory :

Steps to interface sound sensor with galileo.

- Materials used : Sound sensor, connecting wires, Galileo board.

Hardware Setup :

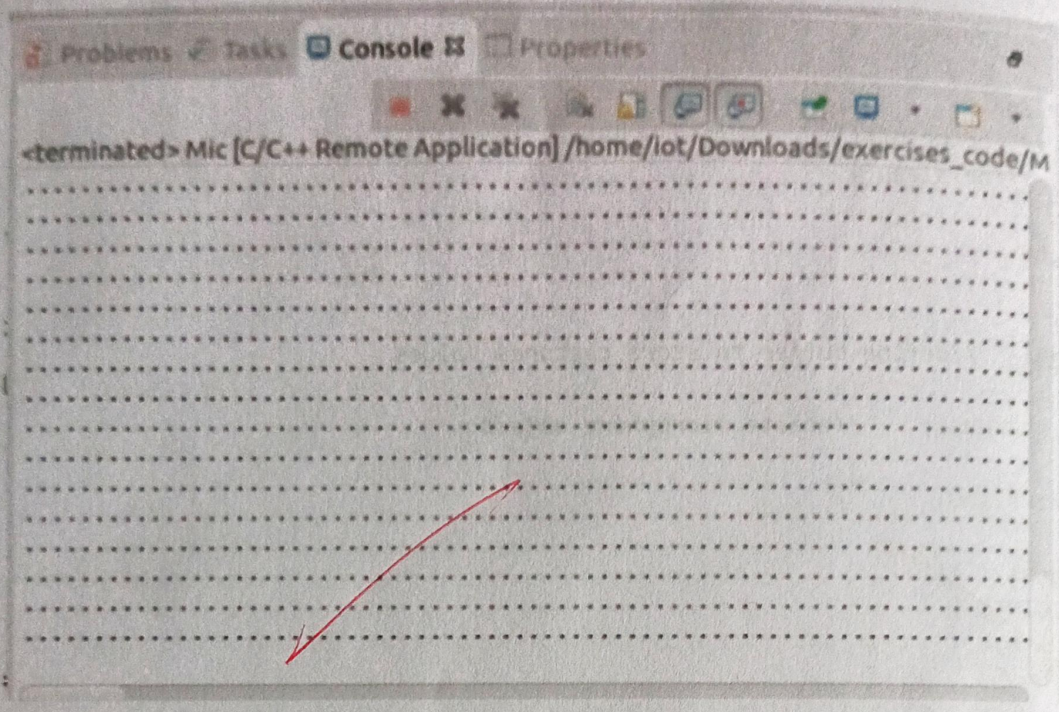
1. Connect sound sensor to galileo board using connecting wires to Analog input pin.
2. Power ON Galileo board, connect to computer system using FTDI to USB cable.
3. Open Eclipse workspace in computer by attaching OS pendrive given in development kit.
4. Write program in new file, connect to galileo connection. Save file.

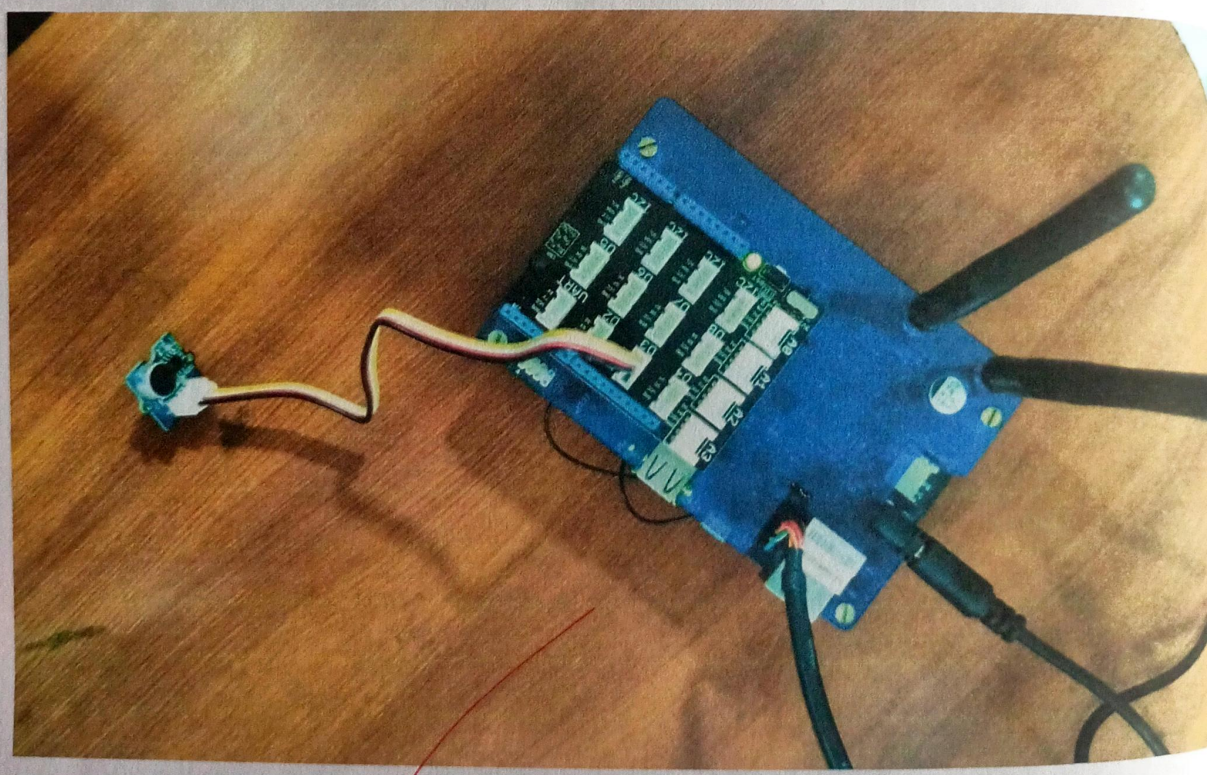
Teachers Signature _____

Code: -

```
#include "mic.h" // for
#include <stdio.h> // for printf()
#include <unistd.h> //for sleep()
#include <signal.h>
#include <sys/time.h>

int is_running = 1;
uint16_t buffer [128];  //define buffer to store captures values

upm::Microphone *mic = NULL;    //create microphone object

void sig_handler(int signo)
{
    printf("got signal\n");
    if (signo == SIGINT) {
        is_running = 0;
    }
}

int main(int argc, char **argv)
{
    // Attach microphone to analog port A0
    mic = new upm::Microphone(0);

    if (signal(SIGINT, sig_handler) == SIG_ERR)
          printf("\ncan't catch SIGINT\n");

    thresholdContext ctx;
    ctx.averageReading = 0;
    ctx.runningAverage = 0;
    ctx.averagedOver   = 2;

    while (is_running) {
        int len = mic->getSampledWindow (2, 128, buffer);
        if (len) {
            int thresh = mic->findThreshold (&ctx, 30, buffer, len);
            mic->printGraph(&ctx);

            if (thresh) {
                // do something ....
            }
        }
    }
    printf ("exiting application\n");
    delete mic;
    return 0;

}
```

Output: -


Problems
Tasks
Console
Properties
<terminated> Mic [C/C++ Remote Application] /home/iot/Downloads/exercises_code/M



PAGE NO.:
DATE.: / / 20

5. Upload program to galileo board using console command.
6. Click on RUN button of IDE.
7. Measure the output on console window.

Results: Sound sensor interfaced with galileo board & understood all working clearly.

Conclusion: Program to interface sound sensor with galileo board & detecting surrounding sound implemented successfully.

Dmap
13/03/24

A+

PAGE NO.:
DATE: 13 / 3 / 2024

Practical no. 9

- Aim: To interface LCD with intel galileo board and write a program to display any text.

- Tools :-

Sr. no.	Tools	qty
1.	Intel Galileo Board	1
2.	LCD display	1
3.	Jumper wires	1

- Theory: Steps to interface the LCD display and write a program to display text.

1. Connect the VCE pin of the LCD display to the 5V pin on the Galileo board.
2. Connect the GND pin of the LCD display to any GND pin on the Galileo board.
3. Connect the SDD pin of the LCD display to any digital pin on the Galileo board.
4. Connect the SCL pin of the LCD display to any digital pin on the Galileo board.
5. Connect the Galileo board to a computers and open the Arduino IDE.
6. Go to "Sketch" → "Include library" → "Manage libraries"
7. In the library manager - search for "liquid crystal I2C".
8. Install the "liquid crystal I2C" library by Arduino.

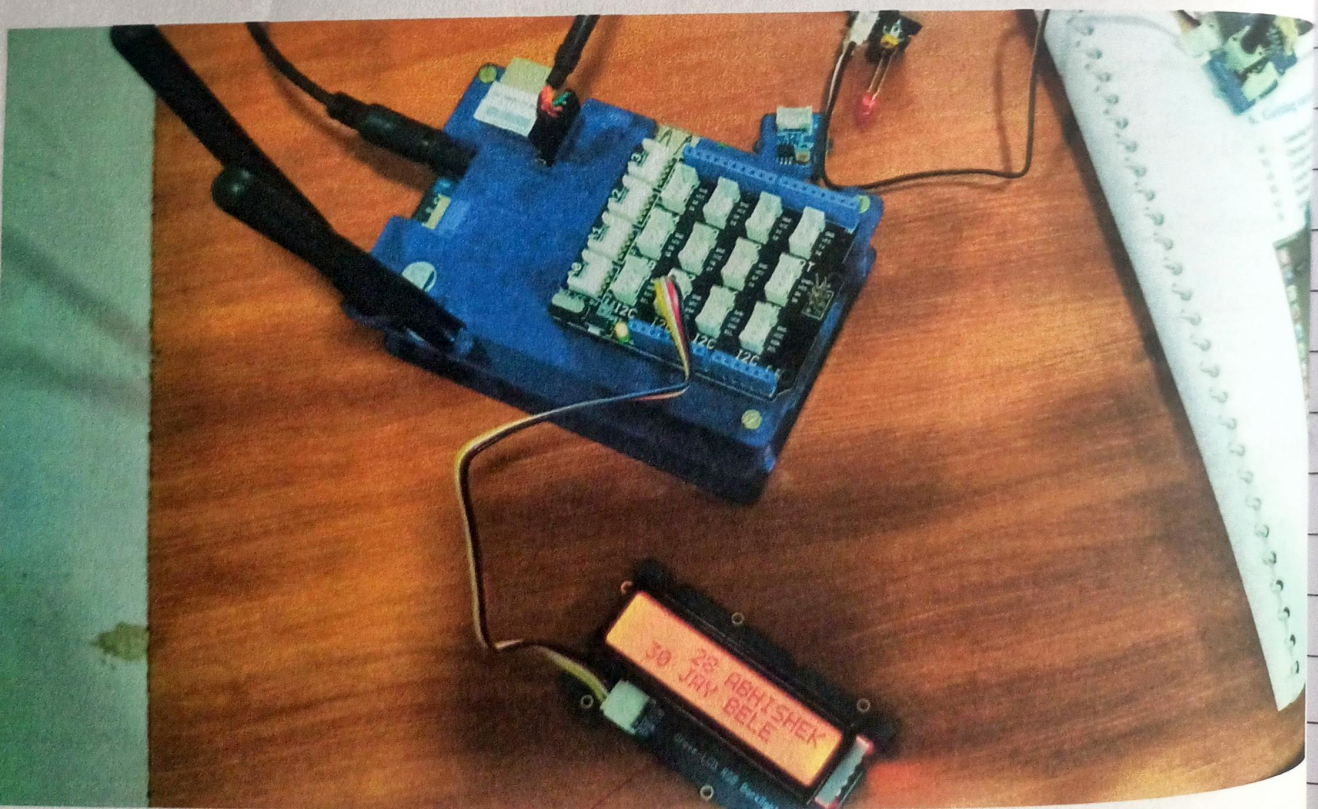
Code: -

```
#include "jhd1313m1.h"
#include <stdio.h> // for printf()

#include <unistd.h> //for sleep()
int main(void)
{
    // 0x62 RGB_ADDRESS, 0x3E LCD_ADDRESS
    upm::Jhd1313m1 *lcd;
    lcd = new upm::Jhd1313m1(0, 0x3E, 0x62); //Create lcd instance
    //arguments: I2C addresses of LCD controller and LED backlight controller

    printf("Display text on LCD\n");

    lcd->setCursor(0,0);    //bring cursor to top left corner
    lcd->write("28 ABHISHEK");  //print text
    lcd->setCursor(1,2);    //bring cursor to second row
    lcd->write("30 JAY BELE"); //print text

    printf("Sleeping for 10 seconds\n");
    sleep(10);
    printf("Starting Color loop...\n");

    //Run loop for toggling backlight color between Red->Green->Blue x 5 times
    for (int i = 5; i>0 ;i--){
    lcd->setColor(255,0,0); //set backlight color to Red
    sleep(1);
    lcd->setColor(0,255,0); //set backlight color to Green
    sleep(1);
    lcd->setColor(0,0,255); //set backlight color to Blue
    sleep(1);
    }

    printf("Exiting .. bbye!\n");

    delete lcd; //free up memory.
    return 0;
}
```

Output: -

Problems | Tasks | Console | Properties

```
LCD [C/C++ Remote Application] /home/iot/Downloads/exercises_code/LCD/Debug/LCD
root@galileo:~# chmod 755 /tmp/LCD;/tmp/LCD ;exit
Display text on LCD
Sleeping for 5 seconds
Starting Color loop...
```



PAGE NO.:
DATE.: / / 20

Result : After uploading the process is complete, The LCD display should start showing the message.

Conclusion : To interface LCD display with intel galileo board and write a program to display some text has been implemented successfully.

A+
02/04/24

PAGE NO.:
DATE.: 2 / 4 / 2024

Practical no. 10

Aim: To interface rotary angle sensor/potentiometer with the Galileo board and write a program to control the readings of sensors of changing the angle of potentiometer.

Tools:-

Sr. no	Tools	Qty
1.	Intel Galileo Board	1
2.	Rotary Angle Sensor	1
3.	Jumper Wires	1

Theory:- Steps to interface rotary angle sensor with the galileo board and write a program to control the readings of the sensor are as follows:

1. Connect the Grove Rotary angle sensor to an analog port (A0, A1, etc) on the shield. Then connect the shield to the Galileo board.
2. Open Arduino IDE Go to Sketch > Include library > Manage libraries. Search for "UPM" and install the "Grove UPM" library by Arduino.
3. Copy the provided program code into a new sketch in the Arduino IDE.
4. Go to Tools > Board and select "Galileo" or "Galileo Gen2" depending on your board. Then go to Tools > port and

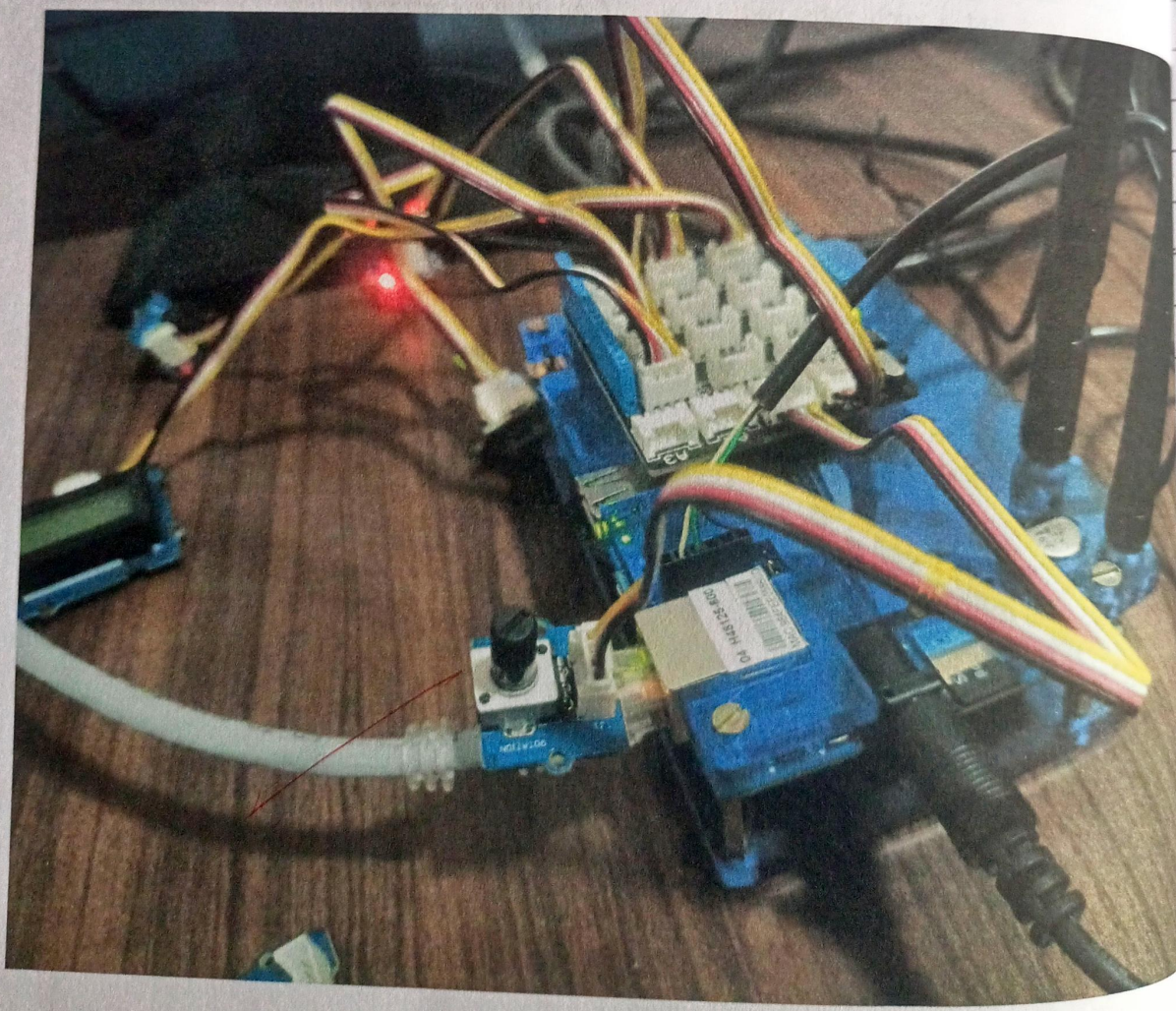
Teachers Signature _____

Code: -

```
#include "grove.h" // for button->value()

/* Standard IO includes */
#include <stdio.h> // for printf()

#include <unistd.h> //for sleep()

int main()
{
      // Instantiate a rotary sensor on analog pin A1
        upm::GroveRotary* knob = new upm::GroveRotary(1);
    // Read the input and print, waiting one second between readings
    while( 1 ) {
            float abs_value = knob->abs_value(); // Absolute raw value
            float abs_deg = knob->abs_deg();     // Absolute degrees
            float abs_rad = knob->abs_rad();     // Absolute radians
            float rel_value = knob->rel_value(); // Relative raw value
            float rel_deg = knob->rel_deg();     // Relative degrees
            float rel_rad = knob->rel_rad();     // Relative radians
            printf("Absolute: %4d raw %5.2f deg = %3.2f rad Relative: %4d raw %5.2f deg
%3.2f rad\n",
                    (int16_t)abs_value, abs_deg, abs_rad, (int16_t)rel_value, rel_deg,
rel_rad);
            sleep(1); // Sleep for 1s
        }
    // Delete the button object
    delete knob;
    //Will the program ever reach here? If no, then how can we delete knob?
    //Hint: See boiler plate for link to signal document
}
```

Output: -

```
Problems  Tasks  Console  Properties

RotaryAngle [C/C++ Remote Application] /home/iot/Downloads/exercises_code/Rotary/
Absolute:  1023 raw 299.71 deg = 5.24 rad Relative:   511 raw 149.85 deg 2
Absolute:  1022 raw 300.00 deg = 5.24 rad Relative:   511 raw 149.85 deg 2
Absolute:  1023 raw 300.00 deg = 5.24 rad Relative:   511 raw 149.85 deg 2
Absolute:  1023 raw 300.00 deg = 5.23 rad Relative:   511 raw 149.85 deg 2
Absolute:  1022 raw 300.00 deg = 5.24 rad Relative:   511 raw 149.85 deg 2
Absolute:   747 raw 219.06 deg = 3.82 rad Relative:   235 raw 68.91 deg 1.
Absolute:   733 raw 215.25 deg = 3.76 rad Relative:   223 raw 65.40 deg 1.
Absolute:   662 raw 194.43 deg = 3.39 rad Relative:   152 raw 44.57 deg 0.
Absolute:   704 raw 206.16 deg = 3.60 rad Relative:   192 raw 56.30 deg 0.
Absolute:   703 raw 206.16 deg = 3.60 rad Relative:   192 raw 56.30 deg 0.
Absolute:   687 raw 201.76 deg = 3.52 rad Relative:   176 raw 51.91 deg 0.
Absolute:   288 raw 84.46 deg = 1.47 rad Relative:  -224 raw -65.69 deg -1
Absolute:   377 raw 110.56 deg = 1.93 rad Relative:  -135 raw -39.59 deg -
Absolute:   579 raw 169.50 deg = 2.96 rad Relative:    68 raw 19.65 deg 0.
Absolute:   536 raw 157.18 deg = 2.75 rad Relative:    25 raw  7.33 deg 0.
Absolute:   453 raw 133.14 deg = 2.32 rad Relative:   -58 raw -17.01 deg -
```



Select the COM port your Galileo board is connected to.

5. Click the upload button to upload the program to your galileo board.
6. Open the serial monitor by going to Tools > Serial monitor, set baud rate to 9600 (usually the default).
7. Turn the knob of the rotary sensor, you should see readings on the serial monitor find that update every second.

- Result: When we run the program and rotate the knob of the rotary sensor, we see output on the serial monitor that updates every second.

- Conclusion: The practical successfully demonstrates how to interface a rotary angle sensor (or potentiometer) with a galileo board. The provided program reads the sensor values and displays them on the serial monitor, allowing to observe relationship between the sensors physical rotation and output readings.

05/04/24 A+

Teachers Signature _____