

INDEX

Name: Gayatri Marshal Kolhe Class Roll No: 05
 Branch: Computer Scie. and Engg. Year / Semester: 3rd / 6th
 Name of Subject: Hardware Lab.

S.No.	Name of Experiment	Page No.	Date	Remarks
1.	Study of IOT development -ut iot and formalization with intel gallileo intel.	1	13/01/23	
2.	To interface LED with intel Gallileo board and write a program to turn on LED repeatedly.	3	26/01/23	
3.	To interface push button with intel gallileo board and write a program to turn on LED.	5	2/2/23	Agile Day C
4.	To Interface buzzers with intel gallileo board and write a program to buzz ON and OFF.	8	09/2/23	
5.	To interface temperature sensor with intel gallileo board to print temperature reading.	10	26/02/23	
6.	To interface touch sensor with intel gallileo board to sense a finger when it is placed.	13	17/03/23	
7.	To interface light sensor with intel gallileo board, to detect something light intensity.	15	30/3/23	
8.	To interface sound sensor with intel gallileo board, to detect surrounding sound intensity.	17	6/4/23	
9.	To interface the LCD display with intel gallileo board and execute a program	19	15/4/23	

14/4/23 to display some text

EXPERIMENT NO. 1

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

Tools:-

SL. NO.	Tools	
01	IOT 3500 Dev. Kit	Intel gallileo kit
02	Computer system	4 GB RAM + 1TB HDD
03	Bootable pen drive	32 GB

Quantity

1
1
1

Process:- Steps to configure intel Galileo Board.

- 1) assemble the hardware from IOT kit ubox intel / Galileo and connect galileo board to power through adapter given.
- 2) Take out the sandisk 32 GB pen drive that is already loaded with Galileo Linux supportable OS username as IOT & attach it to computer , restart your computer as IOT , and select pen drive in 1st position of boot order
- 3) save the settings , new OS will load

Teacher's Signature : _____

```

root@quark01:~# commandit
commandit> services
-WiFi Enabled
      ethernet_984Fee01e95a_cable
      wifi_0cd292591721_hidden_managed_pk
      wifi_0cd292591721_4d696e63_managed_pk
      wifi_0cd292591721_6d6ab632_managed_pk
      wifi_0cd292591721_hidden_managed_leee8021x
      wifi_0cd292591721_49564c_managed_pk
      wifi_0cd292591721_hidden_managed_wep
      wifi_0cd292591721_managed_pk

commandit> agent on
agent registered
commandit> connect wifi_0cd292591721_6d696e63_managed_pk
Agent RequestingInput wifi_0cd292591721_6d696e63_managed_pk
  Passphrase? [ Type=psk, Requirement=mandatory ]
Connected with wifi_0cd292591721_6d696e63_managed_pk
Passphrase?

```

MINGW 0.2.7

```

root@quark01:~# commandit
commandit> services
-WiFi Enabled
      ethernet_984Fee01e95a_cable
      wifi_0cd292591721_hidden_managed_pk
      wifi_0cd292591721_4d696e63_managed_pk
      wifi_0cd292591721_6d6ab632_managed_pk
      wifi_0cd292591721_hidden_managed_leee8021x
      wifi_0cd292591721_49564c_managed_pk
      wifi_0cd292591721_hidden_managed_wep
      wifi_0cd292591721_managed_pk

commandit> agent on
agent registered
commandit> connect wifi_0cd292591721_6d696e63_managed_pk
Tab Tab
By pressing tab, it
was completed for
you

```

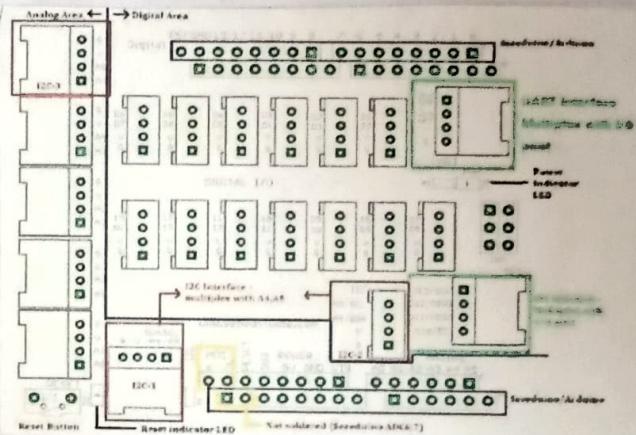
MINGW 0.2.7

```

root@quark01:~# commandit
commandit> enable wifi
Enabled wifi
commandit> scan wifi
Scan completed for wifi
commandit> services
-WiFi Enabled
      wifi
      wifi2
      Hnsguest
      TUL

commandit> agent on
agent registered
commandit>

```



into your computer. login by entering password (IOT123)

- 4) All necessary software are already installed in this now go to terminal and write following command : Make sure to attach gallileo to computer using SDTF to USB port cable
- 5) Power on Gallileo and write command os terminal:

sudo screen usbtty 020511

- 6) This will load Gallileo terminal on os terminal.

- 7) New Gallileo will require password. enter "iot123"

- 8) Now type !command! command in gallileo prompt.

- 9) Type 'search wifi' in terminal by plugging "wifi receiver module" to computer then you will be listed some of SSID's of available wifi.

- 10) copy id of your wifi wifi-ssid and his enter

- 11) Now your Gallileo board is also connected to internet exit by removing FTDI cable from computer.

- 12) Now all configurations has been done

Program:- None.

EXPERIMENT NO. 2

Aim:- To interface LCD with gallileo board and write a program to form on LCD repeatedly after every one sec interval.

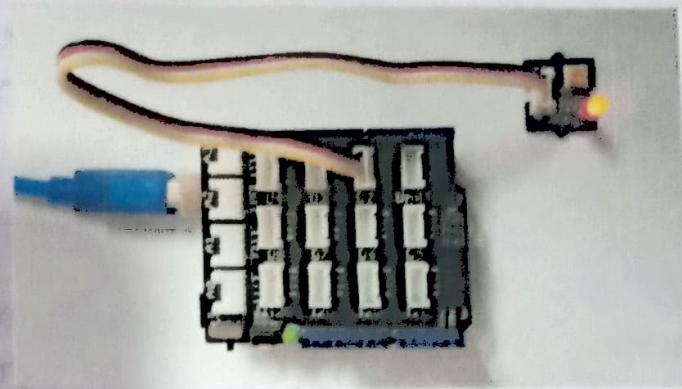
Tools:-

Sr. No.	Tools	Specification	Qty.
1.	Development kit	IOT 3500 kit	1
2.	Computer system	4GB RAM, 1TB HDD	1
3.	LCD Display	LDS15, for Gallileo	1

Theory:- steps to implement above statement in gallileo board.

- 1) Plug Gallileo to computer, and power ON.
- 2) Power computer system, enter password.
- 3) Open "Eclipse IDE" in Linux U. i.e. already installed.
- 4) All the programs were given in workspace of eclipse
- 5) click on the create connection → new connection → enter Gallileo ip address
- 6) click ~~NEXT~~ → right click on connection → click start connection → connection will now change to open.
- 7) write appropriate program and run by clicking on run button.
- 8) write command to interface LCD to gallileo
- 9) upload the code to gallileo and click on run.

Teacher's Signature:



Output :

```
Problems Tasks Console Properties IoT Sensor Support  
<terminated> Light [C/C++ Remote Application] /home/iot/workspace_iot/Light  
root@galileo:~# chmod 755 /tmp/Light;/tmp/Light ;exit  
Button was turned ON  
Times button was pushed : 1  
LED On  
Button was turned OFF  
Times button was pushed : 2  
LED OFF  
Button was turned ON  
Times button was pushed : 3  
LED On  
Button was turned OFF  
Times button was pushed : 4  
LED OFF  
Exiting...BBye!logout
```

Program:-

#include <avr/delay.h>

// Pin 13 has an LED connected to gallileo
board

// Give it a name

int led = 13;

// Setup routine runs once when you
press reset:-

void setup()

PinMode(led, output); //Initialize
digital pin as an output

void loop()

digitalWrite(led, HIGH); // turn LED
on high voltage delay (1000);

}

After upload completes, you should see a tiny,
green LED blinding on and off every second.

This LED is connected to pin 13 on Gallileo.

Result:- Compilation and installation of LED
in faint after every one second is done
and process understand.

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.

EXPERIMENT NO. 3

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

Tools:-

SL.NO.	Tools	specification	qty
1.	IOT development kit	IOT 3500	1
2.	Computer system	4GB RAM 1TB HDD	1
3.	LED	LD15	1

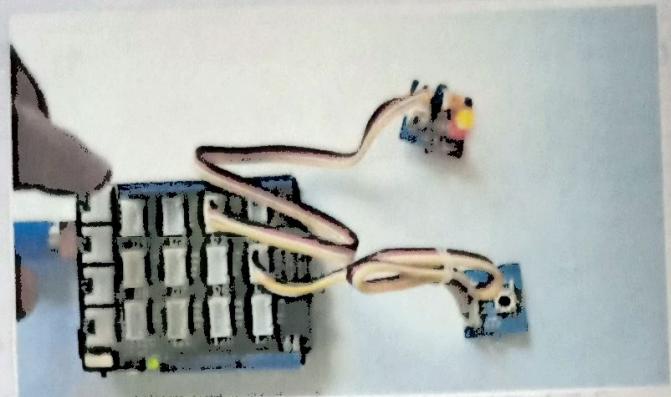
Theory:- Steps to interface push button

Materials need:

- 1) Intel Gallileo 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. P2) on gallileo board.
- 2) connect Jumper wire from P2 on gallileo board to some now on push button.
- 3) connect LED to gallileo board using another connecting wires.
- 4) Power ON gallileo and connect with computer system
- 5) Open Eclipse and connect to your gallileo connection.
- 6) upload into gallileo and click on RUN.



Output :

```
Problems Tasks Console Properties IoT Sensor Support

<terminated> Light [C/C++ Remote Application] /home/iot/workspace_iot/Light
root@galileo:~# chmod 755 /tmp/Light;/tmp/Light ;exit
LED On
LED Off
Exiting .. bbye!logout
```

Program:-

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

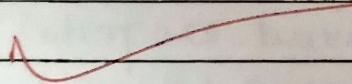
#define BUTTON_PIN 2
#define LED_PIN 3
```

```
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);
    while (1) {
        int button_state = mraa::gpio::read(button);
        if (button_state == 0) {
            mraa::gpio::write(led, 1);
        } else {
            mraa::gpio::write(led, 0);
        }
        sleep(1000);
    }
}
```

```
mraa::gpio::close(button);
mraa::gpio::close(led);
mraa::deinit();
return 0;
```

Results:- Copic problem statement successfully implemented.

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



EXPERIMENT NO.4.

Aim:- To interface buzzers with the intel gallileo board and write a program to buzz on and off with a delay of one second

Tools:-

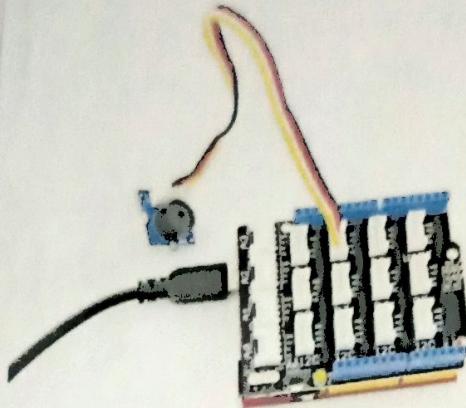
Sl.No.	Tools	specifications	qty.
1.	IOT 3500 development kit		1
2.	Computer System.		1
3.	Buzzer, connecting wires	4GB+RAM .1TB HDD	1
4.	Eclipse IDE		1

Theory:- Steps to interface buzzers with intel / Gallileo

- Materials used : Buzzer, connecting wires, Gallileo board.

Hardware setup:-

- Connect Buzzer to gallileo board using connecting wire to digital input pin (eg. D5)
- connect power adapter of board and turn it on.
- open eclipse IDE in computer system.
- connect to gallileo connection.
- write code in new file
- upload code in gallileo using console
- click on RUN button.



```
sound->setVolume(0.5);

Problems Tasks Console Properties IoT Sensor Support
terminated> Buzzer [C/C++ Remote Application] /home/iot/workspace_iot/Buz
root@galileo:~# chmod 755 /tmp/Buzzer;/tmp/Buzzer ;exit
Volume = 1.000000
Volume = 0.500000
Playing notes, pausing for 0.1 seconds between notes...
3800
3400
3000
2900
2550
2270
2000
3800
2000
3800
Exiting, bbyey
logout
```

Program:-

```
#include<stdio.h>
#include<stdlib.h>
#include<mraa.h>
#define BUZZER_PIN 2
int main()
{
    mraa_init();
    mraa_gpio_context buzzer=mraa_gpio_init
        (BUZZER_PIN);
    mraa_gpio_dir(BUZZER,MRAA_GPIO_OUT);
    while(1)
    {
        mraa_gpio_write(buzzer,1);
        sleep(1);
        mraa_gpio_write(buzzer,0);
        sleep(1);
    }
    mraa_gpio_close(buzzer);
    mraa_deinit();
    return 0;
}
```

Result:- Program implemented correctly and problem understood clearly.

Conclusion:- Program to interface buzzer with intel galileo board to buzz ON or OFF after delay of 1 sec implemented successfully.

EXPERIMENT NO.5

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

Tools:-

- 01 - IOT 3500 Development kit - 1
- 02 - Computer system - 1
- 03. Temperature sensor - 1

Theory:- steps to interface temperature sensor with Intel gallileo board :-

Materials used of kit:- Temperature sensor, connecting wires, gallileo board.

Hardware / software setup:-

1] connect temperature sensor to gallileo board by using connecting wires to input pin (eg. A#5) to read analog values.

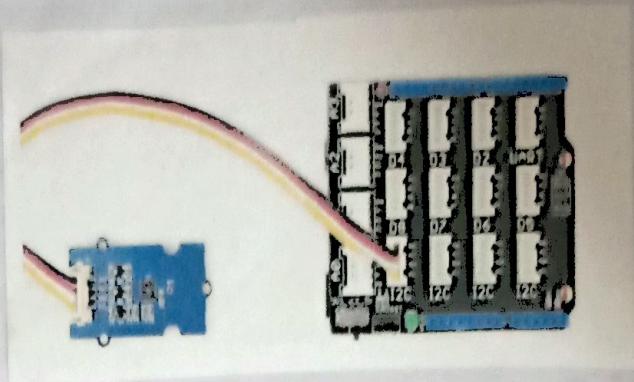
2] Powers ON Gallileo board, connect it to computer system.

3] Open Eclipse IDE in computer, connect to connection, write code in new file.

4] Mention temperature sensor pin number - e on program.

5] upload the code to gallileo board.

Teacher's Signature:



```
Problems Tasks Console Properties IoT Sensor Support
<terminated> Temperature [C/C++ Remote Application] /home/iot/workspace_iot/Temperature/Debug/Ti
root@galileo:~# chmod 755 /tmp/Temperature;/tmp/Temperature ;exit
ADC A0 read value : 616
Temperature value : 33.87 Degree Celsius
ADC A0 read value : 619
Temperature value : 34.14 Degree Celsius
ADC A0 read value : 617
Temperature value : 33.96 Degree Celsius
ADC A0 read value : 616
Temperature value : 33.87 Degree Celsius
ADC A0 read value : 583
Temperature value : 30.97 Degree Celsius
ADC A0 read value : 576
Temperature value : 30.37 Degree Celsius
ADC A0 read value : 572
Temperature value : 30.03 Degree Celsius
ADC A0 read value : 568
Temperature value : 29.69 Degree Celsius
ADC A0 read value : 564
Temperature value : 29.35 Degree Celsius
ADC A0 read value : 567
Temperature value : 29.60 Degree Celsius
Exiting .. Byeelogout
```

6] click on RUN

Program:-

```

#include <stdio.h>
#include <stdlib.h>
#include <linux/i2c-dev.h>
#define I2C_BUS "/dev/i2c-1"
#define TEMP_SENSOR_ADDR 0x48
int main() {
    int i2cfd; int e-p temperature.
    float temp-celsius;
    i2cfd = open(I2C_BUS, O_RDWR);
    if (i2cfd < 0) perror("failed");
    return 1;
}

while(1) {
    if (read(i2cfd, &temp-raw, sizeof(temp-raw)) != sizeof(temp-raw)) {
        perror("failed");
        close(i2cfd);
        return 1;
    }

    temp-celsius = temp-raw * 0.0625;
    printf("Temperature: %.2f°C\n",
           temp-celsius);

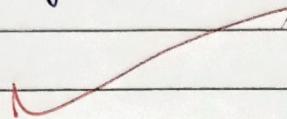
    sleep(1);
}

close(i2cfd);

```

Conclusion:- Program to interface temperature sensor with gallileo board to read temp and print it, implemented successful

-g -



EXPERIMENT NO.6

Aim:- To interface touch sensor with Intel gallileo board and write a program to sense to a finger when it is placed on intel gallileo board.

Tools:-

- 01- IOT 3500 Development Kit - 1
- 02- Computer system - 1

Theory:- Steps to Interface touch sensor with Gallileo
Material used to kit: Touch sensor, connecting wires, gallileo board.

(eg. A5)

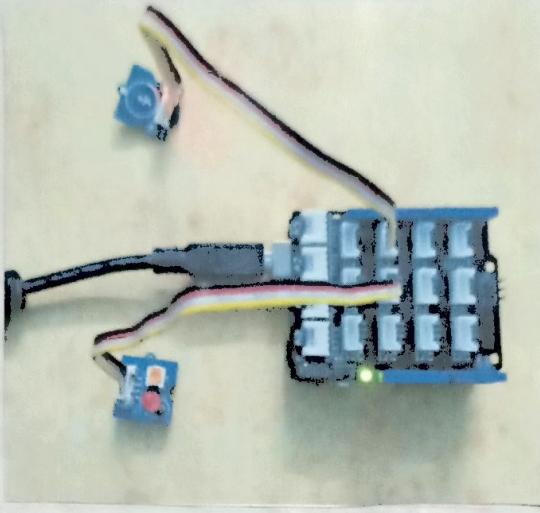
1. Connect touch sensor to gallileo board using connecting wires to analog pin of gallileo board (eg. A5)
2. Power ON gallileo, connect to computer system
3. Open Eclipse IDE on computer, connect to existing connection.
4. ~~write a program~~ on new file
5. ~~save the program~~
6. upload to gallileo board using console command
7. Install method package.

Enter Pkg update

Pkg install i2c-tools

i2c-devtest -x -y <i2c-bus-number>

Teacher's Signature : _____



```
Problems Tasks Console Properties IoT Sensor Support
 TouchInterrupt [C/C++ Remote Application] /home/iot/workspace_iot/Tou
[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 .. ByeLogout
```

8. click on RUN.

9. check the output on gallileo board.

Program:-

```
#include < stdio.h >
#include <unistd.h >
#include <fcntl.h >
#define TOUCH_SENSOR_PIN 4
int main() {
    int touch;
    bool finger_detected = false;
    touch_fd = open("/sys/class/gpio/gpio4/value", O_RDONLY);
    if (touch < 0) perror("failed");
    return 1;
}

while (1) {
    char value;
    if (value == '0') if (!finger_detected)
        printf("Finger detected!\n");
    finger_detected = true;
}

else if (finger_detected == false) {
    usleep(1000);
    close(touch);
    return 0;
}
```

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

Teacher's Signature:

EXPERIMENT NO. 7.

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

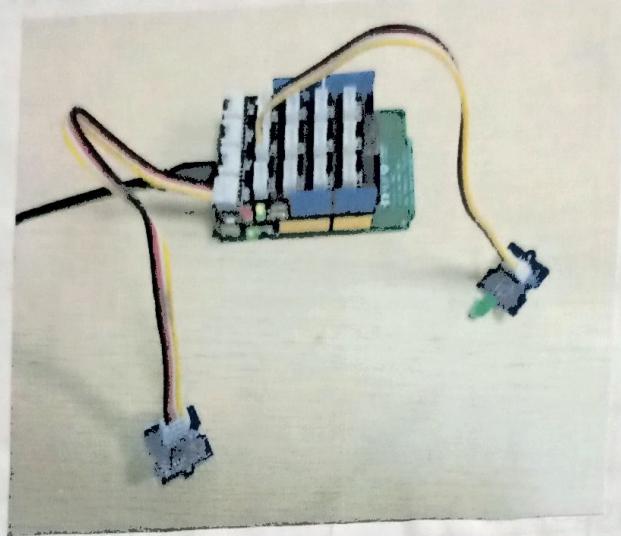
Tools:-

01	- Intel 3500 Development Kit	-	1
02	- Computer system	-	1
03	- Light sensor	-	1

Theory:- Steps to Interface light sensor to gallileo
 - Material required : light sensor, gallileo board, connecting wires.

Hardware setup:

- 1) Connect light sensor to gallileo board using connecting wires to pin input analog eg A5
- 2) Power ON gallileo board, connect it to computer system using FTDI to USB cable
- 3) Open Eclipse IDE in Computer, connect to existing connection user.
- 4) Write a program in new file of eclipse, upload it to gallileo board after interfacing light sensor using console command
- 5) Save file, click on RUN.
- 6) Measure output in console window.



```
Problems Tasks Console Properties IoT Sensor Support
 Light [C/C++ Remote Application] /home/iot/workspace_iot/Light
root@galileo:~# chmod 755 /tmp/Light;/tmp/Light ;exit
Light value is 309.000000 which is roughly 4 lux
Light value is 310.000000 which is roughly 4 lux
Light value is 165.000000 which is roughly 1 lux
Light value is 146.000000 which is roughly 1 lux
Light value is 571.000000 which is roughly 17 lux
Light value is 538.000000 which is roughly 14 lux
Light value is 581.000000 which is roughly 18 lux
Light value is 479.000000 which is roughly 11 lux
Light value is 366.000000 which is roughly 6 lux
Light value is 382.000000 which is roughly 6 lux
.2 Exiting .. bbyelogout
```

Program:-

```

#include <stdio.h>
#include <mraa.h>
#include <unistd.h>
const int light_sensor = 0;
int main() {
    mraa_init();
    mraa_aio_context light_sensor = mraa_aio_init(light_sensor);
    while(1) {
        int raw_val = mraa_aio_read(light_sensor);
        float voltage = (float) raw_val * 5.0 / 1023.0;
        float light_intensity = (voltage / 5.0) * 100.0;
        printf("Light intensity: %.2f\n", light_intensity);
        sleep(1);
    }
    close(light_sensor);
    mraa_deinit();
    return 0;
}

```

Conclusion:- Program to interface light sensor with galileo board and detect surroundings light intensity implemented successfully.

Teacher's Signature : _____

EXPERIMENT NO. 8

Aim:- To interface sound sensor with gallileo board and write a program to detect surrounding sound intensity.

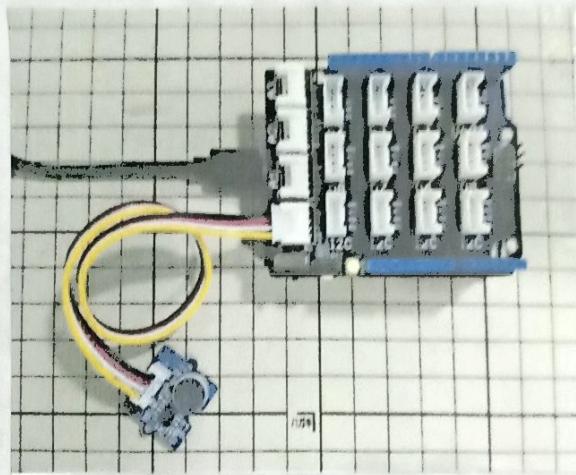
Tools:-

- 01 - Intel 3500 Development kit - 1
- 02 - Computer system - 1
- 03 - sound sensor, connecting wires - 1!!

Theory:- steps to interface sound sensor with gallileo

Material used : sound sensor, connecting wires, gallileo board.

- 1) Connect sound sensor to gallileo board using connecting wires to analog input pin (eg. A5)
- 2) Power on gallileo 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 gallileo connection, save file
- 5) upload program to gallileo board using console command
- 6) click on RUN button of IDE
- 7) Measure the output on console window.



Output :

```
terminated> Mic [C/C++ Remote Application] /home/iot/workspace_iot/Mic/Debug/Mic (24/04/23 2:46 pm)
```

Program:-

```

#include <stdio.h>
# mraa.h
const int sound-pin=0;
int main() {
    mraa-init();
    mraa-qio-context sound-sensor = mraa-
    aio-int(sound-pin);
    while(1) {
        int raw-value = mraa-qio-read(sound-
            sensor);
        printf("sound Intensity : %d\n",
            raw-value);
        sleep(1);
    }
    mraa-aio-close(sound-sensor);
    mraa-delinit();
    return 0;
}

```

Results:- Sound sensor interfaced with gallileo board and understood all working clearly.

Conclusion:- Program to interface sound sensor with gallileo board and detecting surround sound implemented successfully.

EXPERIMENT NO. 9

Aim:- To interface LCD Display with intel gallileo board and write a program to display some text.

Tools:-

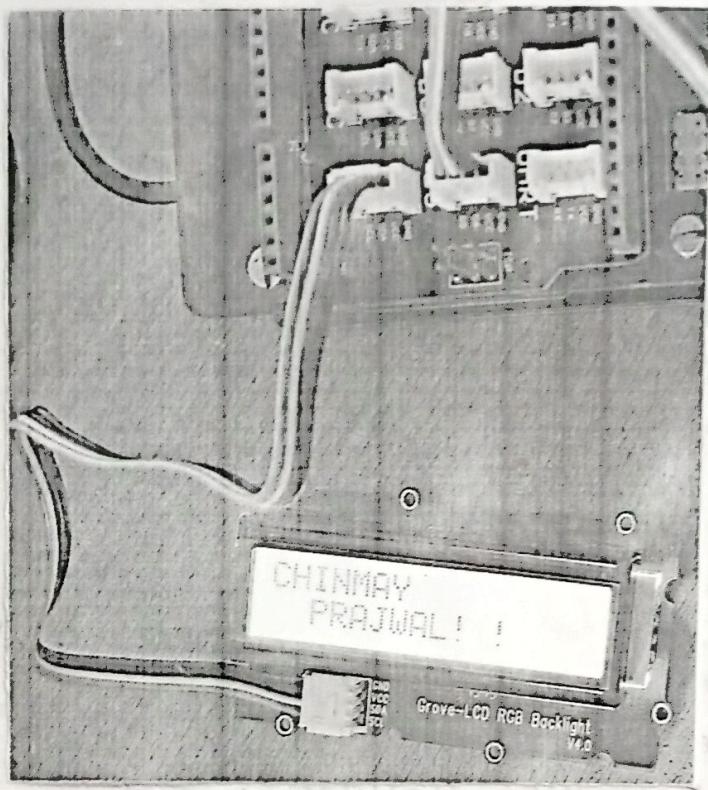
- 1) Intel Gallileo Board - 1
- 2) LCD display - 1
- 3) Jumper wires - 1
- 4) Breadboard - 1

Theory:-

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

- 1) connect the Vcc pin of the LCD display to the 5V pin on the gallileo board.
- 2) connect the GND pin of the LCD display to any GND pin on the gallileo board.
- 3) connect the SDA pin of the LCD display to any digital pin on the gallileo board.
- 4) connect the SCL pin of the LCD display to any digital pin on the gallileo board.
- 5) connect the gallileo board to a computer 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 Focude

Teacher's Signature:



Program:-

```
#include <wire.h>
#include <Liquid crystal-I2C.h>
```

```
Liquid crystal - I2C LCD (0x27, 1612);
void setup();
```

```
lcd.begin(1612);
lcd.backlight();
lcd.print("Hello. Gallileo!");
```

F
E
void loop();

Result:- After uploading the program to computer,
the LCD display should start showing
the message "Hello gallileo".

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

Output :

```
<terminated> LCD [C/C++ Remote Application] /home/iot/workspace_iot/LCD/  
chmod 755 /tmp/LCD;/tmp/LCD ;exit  
root@galileo:~# chmod 755 /tmp/LCD;/tmp/LCD ;exit  
Display text on LCD  
Sleeping for 2 seconds  
Starting Color loop...  
Exiting .. bby!  
logout
```