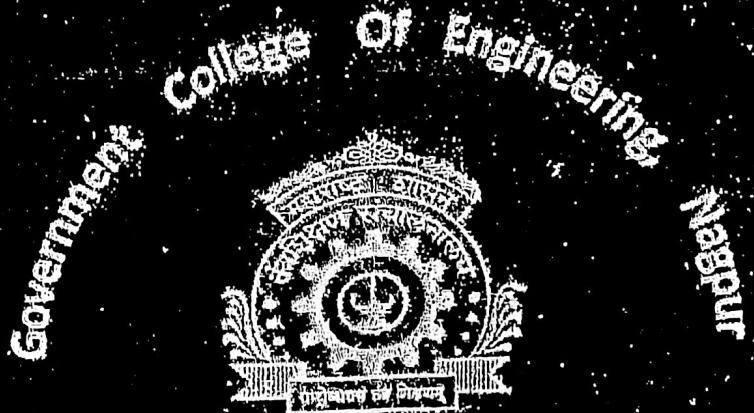


IOT Practical List

- 1 Study of eloT 3500 development kit and familiarization with Intel Galileo board and perform necessary software installation.
- 2 To interface LED with intel galileo board and write a program to turn on LED repetately after every one sec interval.
- 3 To interface push button with intel galileo board and write a program to turn on LED when push button is pressed.
- 4 To interface buzzer with intel galileo board and write a program to buzz on and off with a delay of one second.

- 5 To interface temperature sensor with intel galileo board and write a program to print temperature readings.
- 6 To interface touch sensor with intel galileo board and write a program to sense a finger when it is placed on intel galileo board.
- 7 To interface light sensor with intel galileo board and write a program to detect surrounding light intensity.
- 8 To interface sound sensor with intel galileo board and write a program to detect surrounding sound intensity.
- 9 To interface LCD with intel galileo board and write a program to display any text.



Certificate

Name of the Department Computer Science and Engineering This

is to certify that this practical record contains the bonafide practical work of

Mr./ Ms. Gayatri H. Roline

of 3rd year / semester 6th sem bearing Roll No. 05

has successfully completed the practical term work in the subject Hardware
Lab

during the academic session 2021 - 2022.

Signature of Teacher

(1) Ajedwakar

Date: 28/04/2023

Datesh
Head of Department

INDEX

Name : Gayatri Harshal 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 3500 development kit IoT and formalization with intel Galileo Intel.	1	19/01/23	
2.	To interface LCD with intel Galileo board and write a program to turn on LED repeatedly.	3	26/01/23	
3.	To interface push button with intel Galileo board and write a program to turn on LCD.	5	21/2/23	A good day
4.	To interface buzzers with intel Galileo board and write a program to buzz ON and OFF.	8	09/2/23	
5.	To interface temperature sensor with intel Galileo board to print temperature reading.	10	26/02/23	
6.	To interface touch sensor with intel Galileo board to sense a finger when it is placed.	13	17/03/23	
7.	To interface light sensor with intel Galileo board, to detect something light intensity.	15	30/3/23	
8.	To interface sound sensor with intel Galileo board, to detect surrounding sound intensity.	17	6/4/23	
9.	To interface the LCD display with intel Galileo board and execute a program to display some text	19	15/4/23	

EXPERIMENT NO.1

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

Tools:-

Sr. No.	Tools	
01	IOT 3500 Dev. Kit	Intel gallileo kit
02	Computer system	4 GB RAM, 1 TB HDD
03	Removable pendrive	32 GB

Quantity

1

1

1

Theory:- Steps to configure Intel Galileo Board.

- 1) Assemble the hardware from IOT kit which includes intel / Galileo and connect Galileo board to power through adapter given.
- 2) Take out the sdcard 32 GB pendrive that is already loaded with Galileo Linux supportable OS username as IOT and attach it to computer, restart your computer as IOT, and reset pendrive in 1st position of boot order.
- 3) Save the settings, now OS will load

Teacher's Signature : _____

```

MINGW32/
root@quark01:~/efdf:~# systemctl
systemctl> services
● Rc Wired
  ● Mincguest
  ● minc
  ● mjk2
  ● IUL
  ● ethernet_984fee01e95a_cable
  ● wifi_0cd292591721_hidden_managed_psk
  ● wifi_0cd292591721_4d986e636775657374_managed_none
  ● wifi_0cd292591721_6d696e63_managed_psk
  ● wifi_0cd292591721_6d6a6b32_managed_psk
  ● wifi_0cd292591721_hidden_managed_ieee8021x
  ● wifi_0cd292591721_49564c_managed_psk
  ● wifi_0cd292591721_hidden_managed_wep

systemctl> agent on
Agent registered
systemctl> connect wifi_0cd292591721_6d696e63_managed_psk
Agent RequestInput wifi_0cd292591721_6d696e63_managed_psk
  Passphrase : [ Type=psk, Requirement=mandatory ]
Connected wifi_0cd292591721_6d696e63_managed_psk
Passphrase?

```

```

MINGW32/
root@quark01:~/efdf:~# systemctl
systemctl> services
● Rc Wired
  ● Mincguest
  ● minc
  ● mjk2
  ● IUL
  ● ethernet_984fee01e95a_cable
  ● wifi_0cd292591721_hidden_managed_psk
  ● wifi_0cd292591721_4d986e636775657374_managed_none
  ● wifi_0cd292591721_6d696e63_managed_psk
  ● wifi_0cd292591721_6d6a6b32_managed_psk
  ● wifi_0cd292591721_hidden_managed_ieee8021x
  ● wifi_0cd292591721_49564c_managed_psk
  ● wifi_0cd292591721_hidden_managed_wep

systemctl> agent on
Agent registered
systemctl> connect wifi_0cd292591721_6d696e63_managed_psk
Tab           Tab
By pressing tab, it
auto completes for
you

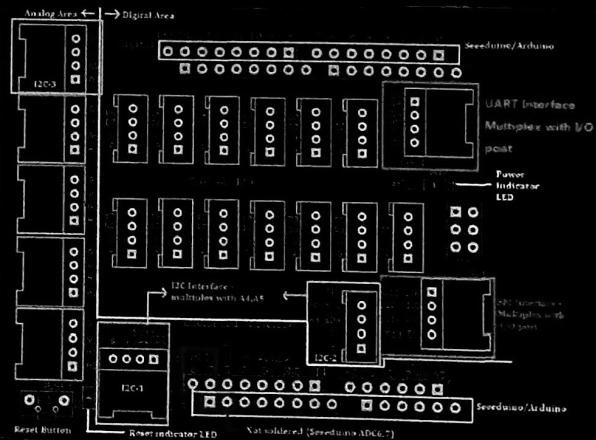
```

```

MINGW32/
root@quark01:~/efdf:~# systemctl
systemctl> enable wifi
Enabled wifi
systemctl> scan wifi
Scan completed for wifi
systemctl> services
● Rc Wired
  ● Mincguest
  ● minc
  ● mjk2
  ● IUL
  ● ethernet_984fee01e95a_cable
  ● wifi_0cd292591721_hidden_managed_psk
  ● wifi_0cd292591721_6d696e63_managed_psk
  ● wifi_0cd292591721_6d6a6b32_managed_psk
  ● wifi_0cd292591721_4d986e636775657374_managed_none
  ● wifi_0cd292591721_hidden_managed_ieee8021x
  ● wifi_0cd292591721_49564c_managed_psk
  ● wifi_0cd292591721_hidden_managed_wep

systemctl> agent on
Agent registered
systemctl>

```



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 020\$11

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

- 7) Now 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.

Kanha Conclusion :- Study of IOT 3500 development IoT one formalization with Intel gallileo board and perform necessary software installation performed successfully.

EXPERIMENT NO. 2

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

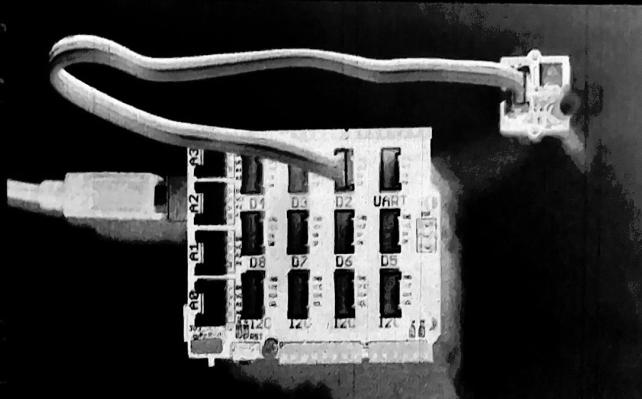
Tools:-

No.	Tools	Specification	Qty.
1.	Development Kit	IOT 3500 kit	1
2.	Computer system	4GB RAM 1TB HDD	1
3.	IBD Display	TDIS. 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 IBD to gallileo
- 9) upload the code to gallileo and

Teacher's Signature : 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
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...Bye!logout
```

Program:-

~~/* Blink */~~

// Pin 13 has an LED connected to Galileo
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 Galileo.

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

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 LCD when push button is pressed.

Tools:-

S.R.NO.	Tools	Specification	Qty
1.	IOT development kit	IOT 3500	1
2.	Computer system	4GB RAM 1TB HDD	1
3.	LCD	LCD1602	1

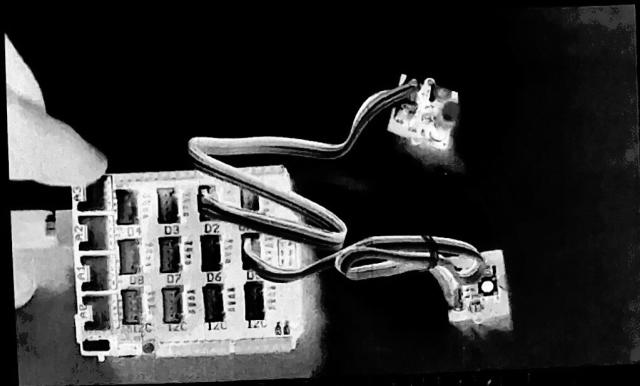
Theory:- Steps to interface push button

Materials need:

- 1) Intel Gallileo board
- 2) Push Button
- 3) LCD
- 4) Resistor
- 5) connecting wires.

Hardware setup steps:

- 1) connect one terminal of push button to a digital input pin (e.g. P2) on gallileo board.
- 2) connect Jumper wire from P2 on gallileo board to some row on push button.
- 3) connect LCD 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 .. bbyelogout
```

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);
```

1

```
mraa.gpio.close(button);
```

```
mraa.gpio.close(led);
```

```
mraa_deinit();
```

```
return 0;
```

2

PAGE NO.: 7
DATE.: / / 20

Results:- Cyclic problem statement successfully implemented.

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

Teachers Signature _____

EXPERIMENT NO.4.

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

Tools:-

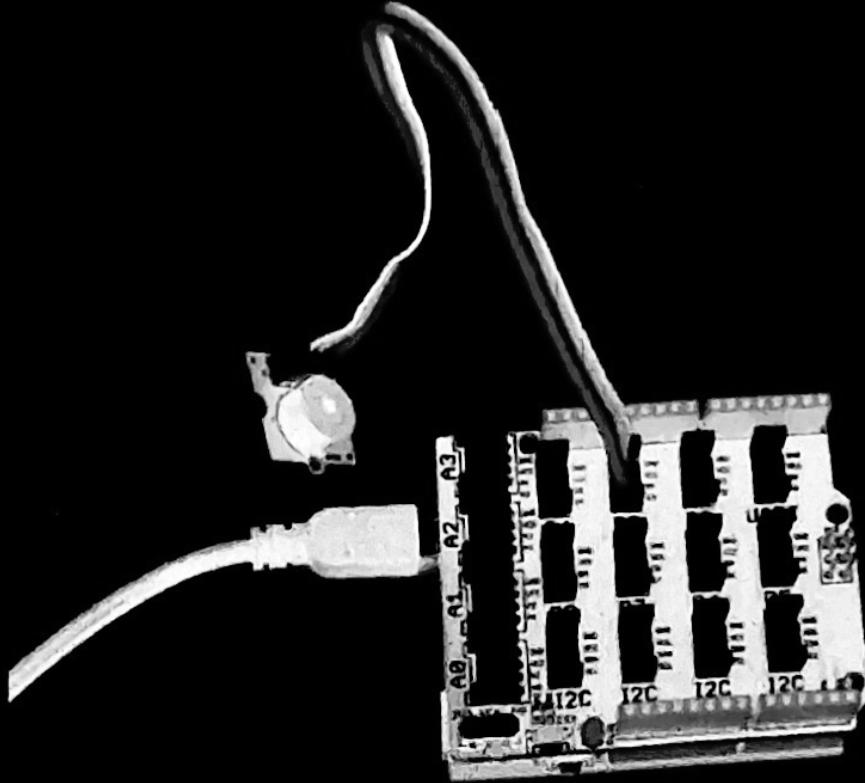
sr.no.	Tools	specifications	qty.
1.	IOT 3500 development kit		1
2.	Computer system.		1
3.	Buzzer, connecting wires	4GB+RAM .1TB HDD	1
4.	Eclipse IDE		1

Theory:- steps to interface buzzer with intel / Gallile

- 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.



A9 I2C P
pushbutton connect
buzzer

```
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, bby!
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

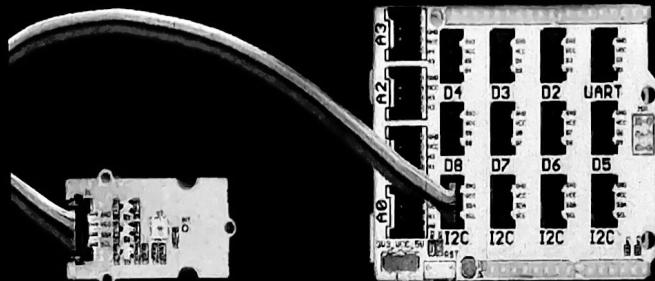
Theory:- steps to interface temperature sensor with intel gallileo board :-

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. A15) to read analog values.
- 2] Power 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/Temperature
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 : 38.97 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 : 29.66 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 TMP_SENSOR_ADDR 0x48
int main() {
    int i2cfd; int temp;
    float temp_celcius;
    i2cfd = open(I2C_BUS, O_RDWR);
    if (i2cfd < 0) perror("failed");
    return 1;
}

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

temp_celcius = temp * 0.0625;
printf("Temperature : %.2f\n", temp_celcius);
sleep(1);

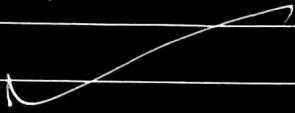
close(i2cfd);
}
```

Teacher's Signature :

Page No	12
Date	

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

-g-



Teacher's Signature : _____

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 do kit: Touch sensor, connecting wires, gallileo board.

(eg. A5)

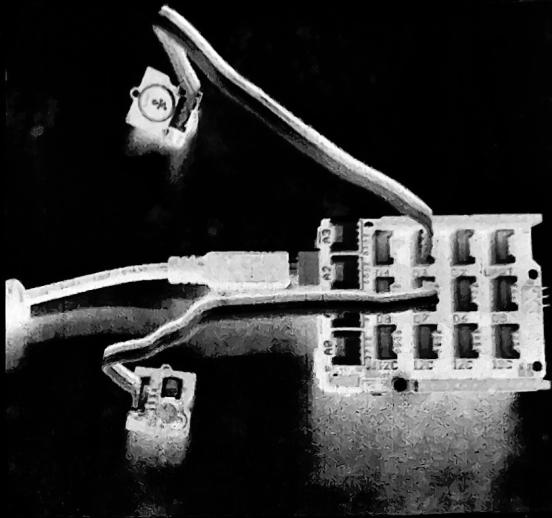
1. Connect touch sensor to gallileo board using connecting wires to analog pin of gallileo board (eg. A5)
2. Power ON gallileo, connect to computer system
3. Open Eclipse IDE on computer, connect to existing connection.
4. Write a program on new file
5. Save the program
6. upload to gallileo board using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

Enter Pkg update

Pkg install i2c-tools

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

Teacher's Signature :



```
Problems Tasks Console Properties IoT Sensor Support

```

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 3
int main() {
    int touch;
    bool finger_detected = false;
    touch_fd = open("sys/class/gpio/gpio7/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 { 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:-

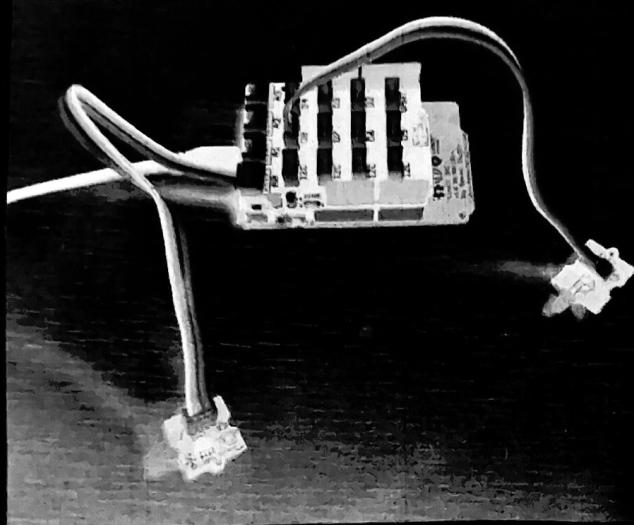
- 01 - Intel 3500 Development Kit - 1
- 02 - Computer system - 1
- 03 - Light sensor - 1

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.

Teacher's Signature :



```
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
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
3.2 Exiting .. bbye!logout
```

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: %f\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.

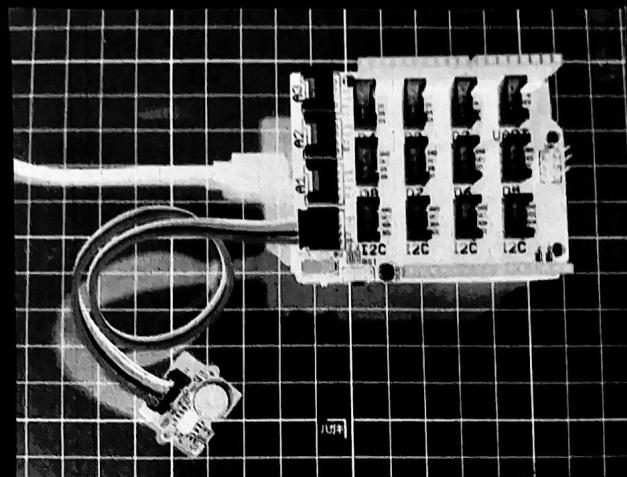
Tools:-

- 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.

Teacher's Signature :



Output :

The screenshot shows a terminal window with the following details:

- Header:** Problems, Tasks, Console, Properties, IoT Sensor Support.
- Title Bar:** <terminated> Mic [C/C++ Remote Application] /home/iot/workspace_iot/Mic/Debug/Mic (24/04/23 2:46 pm)
- Content:** The terminal window is mostly blank, showing a large number of horizontal dotted lines, which typically indicates a long delay or a stuck loop in the application's output.

Program:-

```
#include <stdio.h>
#include "mraa.h"
const int sound_pin = 0;
int main() {
    mraa_init();
    mraa_aio_context sound_sensor = mraa_aio_init(sound_pin);
    while(1) {
        int raw_value = mraa_aio_read(sound_sensor);
        printf("sound Intensity : %d\n", raw_value);
        sleep(1);
    }
    mraa_aio_close(sound_sensor);
    mraa_deinit();
    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.

Teacher's Signature :

EXPERIMENT NO. 9

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

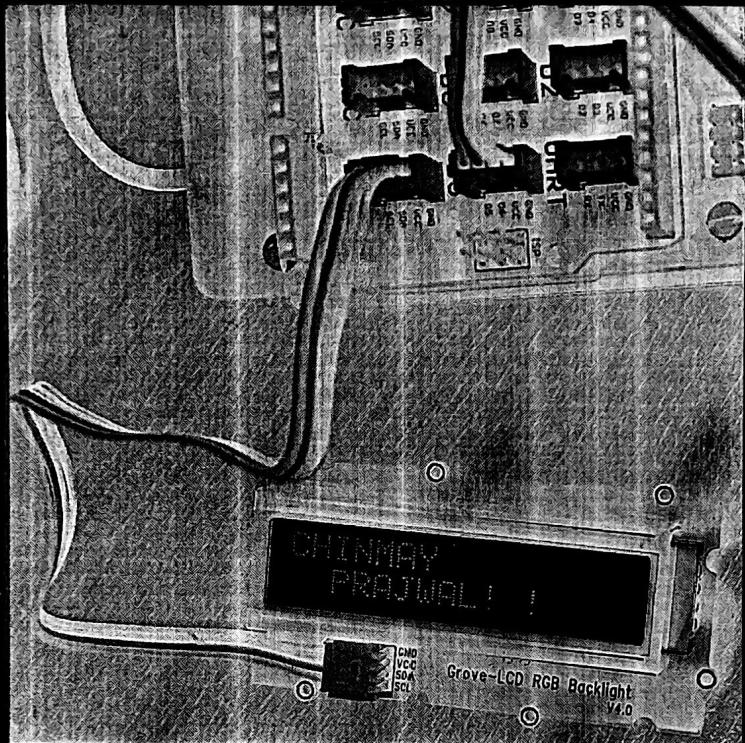
Tools:-

- 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 Frances Teacher's Signature:



Program:-

```
#include <wire.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
```

```
    lcd.begin(16, 2);
```

```
    lcd.backlight();
```

```
    lcd.print("Hello. Galileo!");
```

}

```
void loop()
```

}

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

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

Teacher's Signature :

Output :

The screenshot shows a terminal window with the following interface elements:

- Top bar: Problems (checkbox), Tasks (checkbox), Console (checkbox), Properties (checkbox), IoT Sensor Support (checkbox).
- Icons: A set of small icons including a square, a cross, a gear, a document, a magnifying glass, a play button, and a refresh symbol.

The terminal output is as follows:

```
<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
```