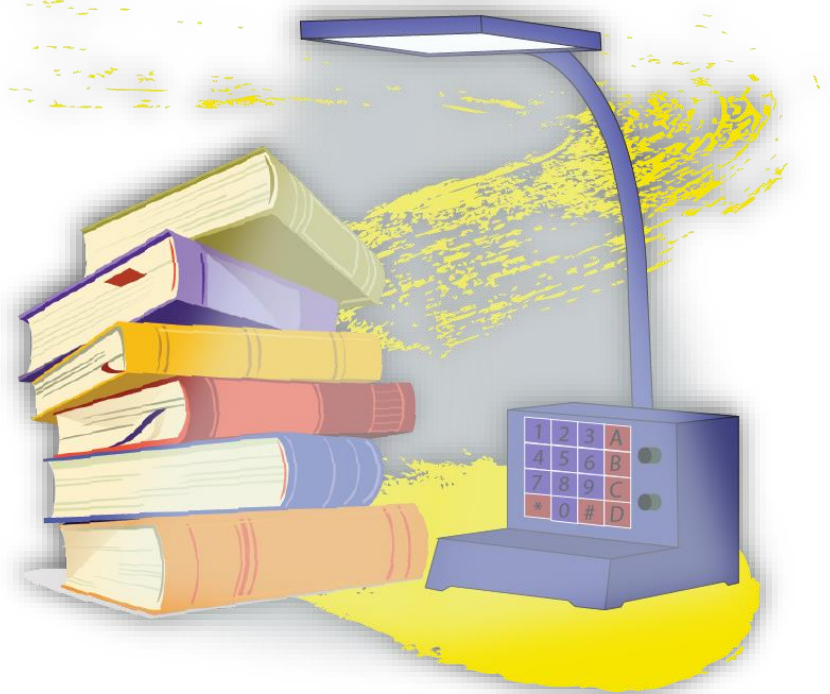


Study Buddy table lamp



AS2019912
T.K.Hashini Dilhara
ICT 305 2.0
Embedded project final report

CONTENT

1. Abstract -----	3
2. Introduction-----	4-6
2.1 Reason for selection	4
2.2 What is study buddy lamp?	5
3.System Models-----	6-8
3.1 System Design	
3.1.1 System Architecture (Block Diagram)	6
3.1.2. Flow chart.	7
3.1.3. Simulator Review.	8
4.System Implementation-----	9-14
4.1. Explanations of Components.	9
4.2. Materials	
4.2.1Sowtware Requirements.	14
5.Budget-----	14
6.Physical Review and code-----	15-35
7.Issues-----	35
8.Discusstion-----	35
9.Conclussion-----	35
10.Reference-----	36

1. Abstract

The concept of product creation is growing daily in the world of modern technology during an effort to satisfy customers. Multi-featured items have significantly changed society in the modern era. This project will develop the idea of a multifunctional table light as its focal point. The enhanced table lamp will offer the ability to purchase a specific quantity of items. This project demonstrates how to build a versatile table lamp. this project is specifically intended for those that strive to achieve their goals

2. Introduction

2.1 Reason for selection

Most students prefer to study at night to avoid any disturbance. Eye strain is a common result of reading in dim light. Table lamp is a daily need for most of the students and official personals. Earlier industries focus only on its aesthetic look. But now-a-days along with its aesthetic look, multipurpose use of a lamp has become a vital issue. Today customers are more likely to have a product of different features in a single hand. In this project, I try my best to create a smart desk lamp. As students we spend a lot of time studying. Trying to come up with a cool project, I thought about the pains that we have in the studying process. I believed that there must be a better way to manage studying habits in a nicer way.

2.2 What is study buddy lamp?

The Arduino mega board was used to design the lamp. It contains a keypad, display, LED panel and sensors. There are four operating modes for the lamp.

1. Automatic mode

The lamp can be turned on and off automatically in automatic mode. The lamp will come turn on when the room is dark. If not, the lamp is not on. The LdR sensor detects it.

2. Manual mode

Keypad or Bluetooth controls can be used to turn on and off the lamp manually.

Students are the best candidates for this table lamp. When studying, a student has the option of studying or taking exams.

3. Exam mode

Students can set the number of hours and minutes before the exam when selecting the exam mode. then the lamp will begin the timer and show the results on the LCD display. The busser will sound when the allocated time has passed.

The student can use a keypad or Bluetooth to input the hours and minutes. It can input a time in two digits.

And enter by using the keypad's "#" button. You can press the "E" button once more to end the exam mode. then the clock will stop, and the exam mode will terminate.

4. Study mode

The study mode on this lamp is crucial. Time management while studying is a common issue among students. Therefore, I created the study mode using the pomodoro method.

The Pomodoro Technique

1. Identify a task or tasks that you need to complete.
2. Set a timer for 25 minutes.
3. Work on a task with no distractions.
4. When the alarm sounds, take a 5-minute break.
5. Repeat the process 3 more times.
6. Take a longer 30-minute break and start again.

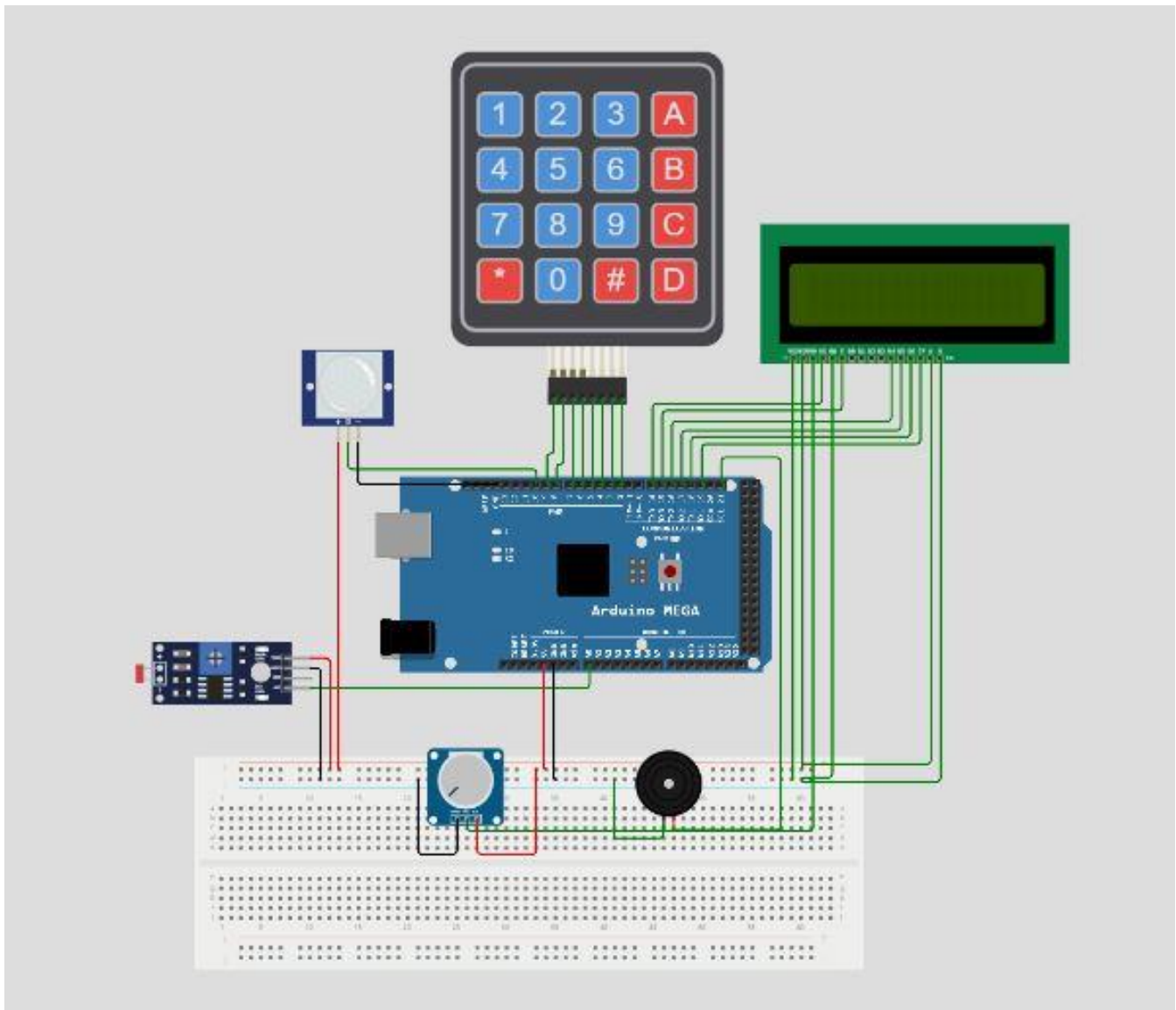
Pomodoro technique is highly effective as it helps you effectively manage your time and work on a task without distractions. It is also beneficial as it helps you become more disciplined and think about your work. This technique is designed to combat multitasking and improve concentration.

The study mode is activated when the learner presses the 'S' on keypad. Afterward, the learner can specify the time in hours, minutes, and intervals. The timer then starts, and the buzzer will sound at the beginning and finish of the break. And enter by using the keypad's '#' button. You can press the 'S' button once more to end the study mode. then the clock will stop, and the study mode will terminate.

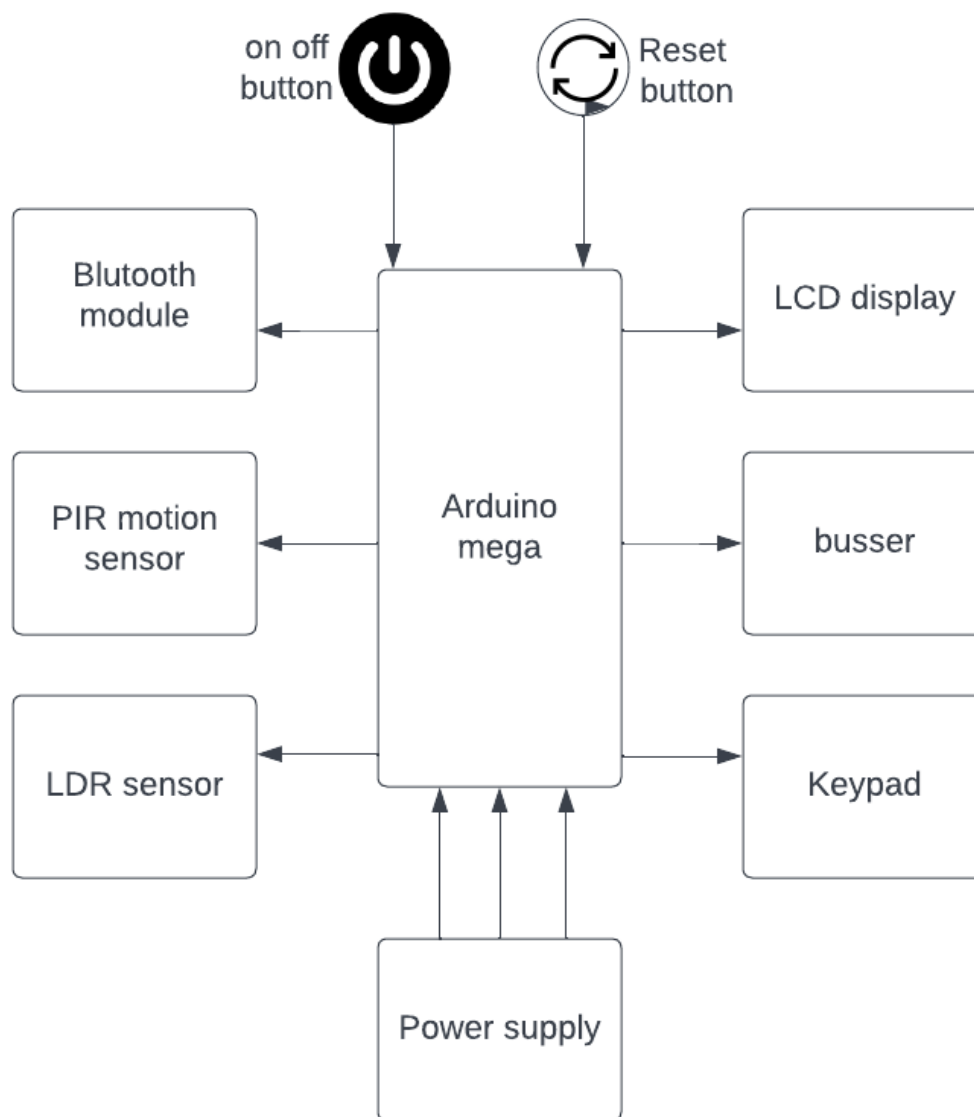
When a lamp is turned on, the movement might be awful. As a result, if the youngster falls asleep, the lamp will detect it and sound the alarm.

3. System Models

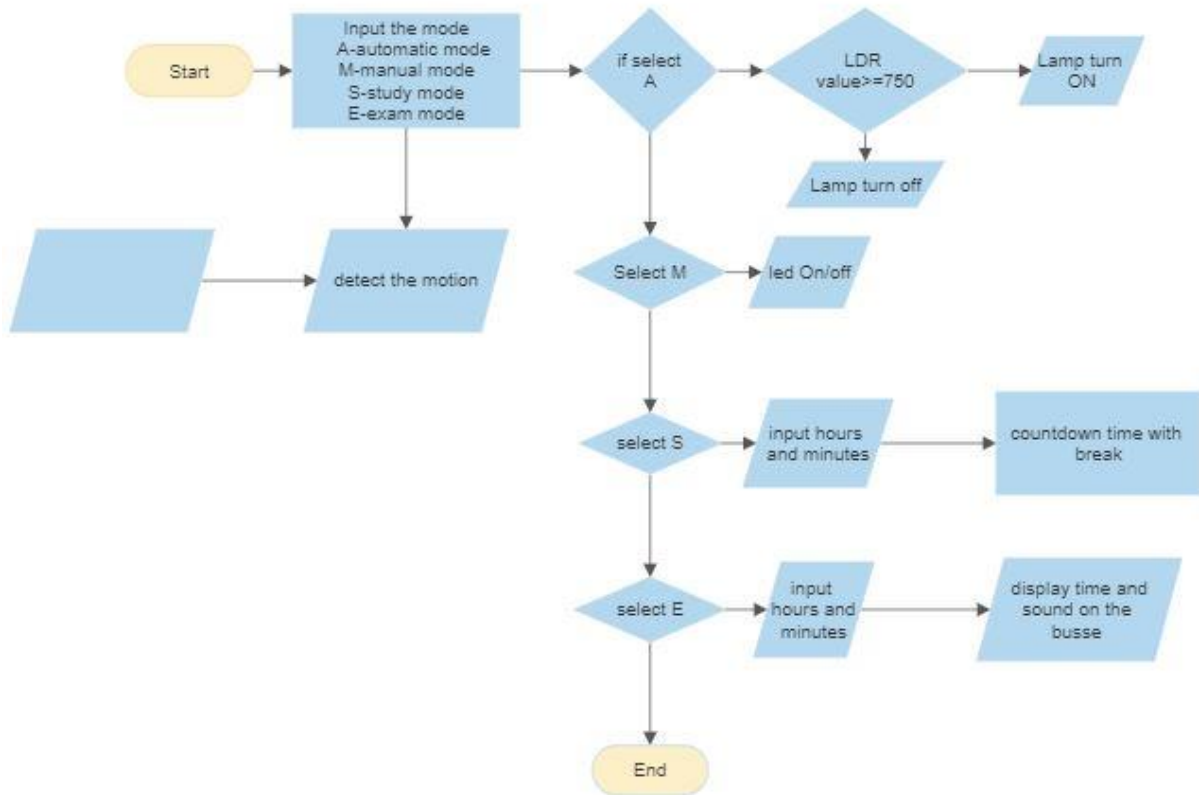
3.1 System Design



3.1.1 System Architecture (Block Diagram)



3.1.2. Flow chart



4. System Implementation

4.1. Explanations of Components

4.1.1. Arduino mega



The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

We can also use an Arduino Uno for this project.

4.1.2. PIR sensor



Human movement is picked up by an PIR sensor within about 10 meters of the sensor. PIRs are fundamentally made of a pyroelectric sensor, which can detect levels of infrared radiation. This figure is an average because the actual detection range is between 5m and 12m. For many important tasks or objects that must be aware of when someone has entered or departed the area. PIR sensors are

wonderful, they are flat control and low effort, have a large lens range, and are simple to interface with.

It used to take the motion detect when student in study mode and exam mode. It aimed to the student and table area. If motion is stopped, we can tell if someone is sleeping.

4.1.3. HC-05 Bluetooth Module



A well-liked module called the HC-05 can give your projects two-way (full-duplex) wireless connectivity. This module can be used to connect with any Bluetooth-enabled device, including a phone or laptop, as well as two microcontrollers like an Arduino. This approach is greatly facilitated by the abundance of existing Android applications. The module can easily be interfaced with any microcontroller that supports USART because it communicates using USART at a 9600 baud rate.

In this project, a light was utilized to turn on the modes so that the lamp could be controlled by a phone or laptop.

Change the name of Bluetooth HC-05

The default name of this module is HC-05. And the default password is 1234. I changed this default name to Study buddy. I used AT command for change the name when Bluetooth Module in command mode.

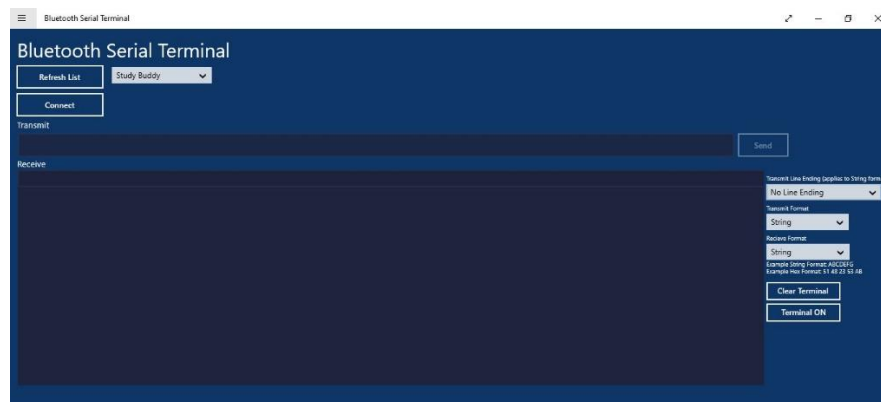


Figure 1. Interface of terminal

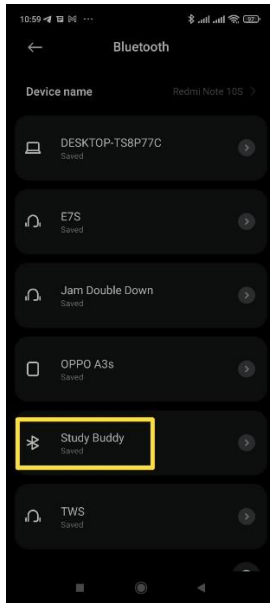


Figure 2. connect lamp with mobile phone

4.1.5.LDR Sensor module



LDR sensor module is used to detect the intensity of light. It is associated with both analog output pin and digital output pin labelled as AO and DO respectively on the board. When there is light, the resistance of LDR will become low according to the intensity of light.

It used to detect the room is dark or light. When room is dark the lamp is turn on automatically.

4.1.6.LCD Display



A type of flat panel display known as an LCD (Liquid Crystal Display) operates primarily using liquid crystals. Since they are frequently used in cellphones, televisions, computers, and instrument panels, LEDs offer a wide range of applications for consumers and enterprises.

It used to display messages and timer countdown in study mode.

4.1.7. BUSSER



A buzzer made by Arduino is like a beeper. When an electric current is fed through the Arduino buzzer, it emits sound. By applying various frequency electric pulses to the buzzer, which is directly connected to the Arduino, multiple tones can be produced.

4.1.8. Keypad 4x4



The buttons on a keypad are arranged in rows and columns. A 3X4 keypad has 4 rows and 3 columns, and a 4X4 keypad has 4 rows and 4 columns. Beneath each key is a membrane switch. Each switch in a row is connected to the other switches in the row by a conductive trace underneath the pad. Each switch in a column is connected the same way – one side of the switch is connected to all the other switches in that column by a conductive trace.

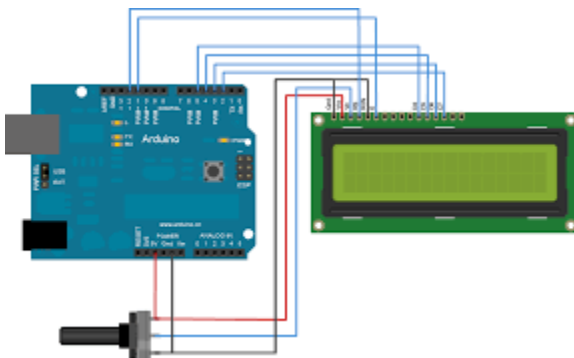
Each row and column is brought out to a single pin, for a total of 8 pins on a 4X4 keypad.

It used to input the time and modes.

Table of input characters and their actions

	A	B
1	Action	Input character
2	automatic light On/off	A
3		
4	manually light on/off	M
5		
6	studymode on/off	S
7		
8	Exam mode on /off	E
9		
10	set timer	t
11		
12		

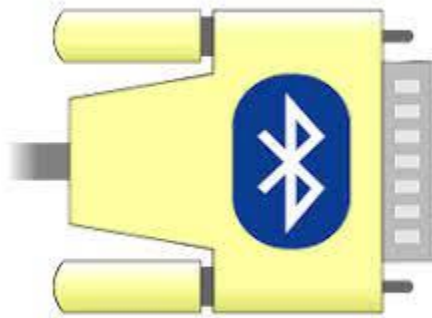
4.1.9 potentiometer



To display any output and input of Arduino we can use a liquid crystal display module (LCD). LCD is one of the essential components to get the visual output. The output can be a value of any sensor or a device such as potentiometer. This discourse explains the process of how we can display the potentiometer values on the LCD display in Arduino.

4.2. Materials

4.2.1. Software Requirements

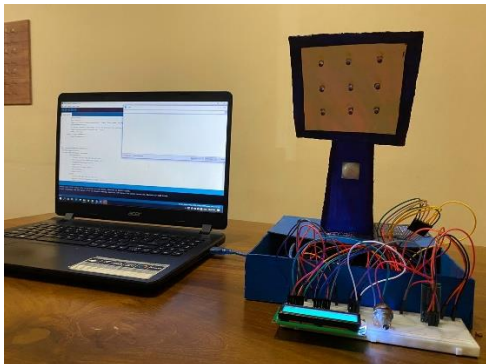


Serial Bluetooth Terminal' is a line-oriented terminal / console app for microcontrollers, Arduinos, and other devices with a serial / UART interface connected with a Bluetooth to serial converter to android device.

5. Budget

Item	Price
Arduino Mega Board	2200LKR
16*2 LCD Display	890LKR
Bluetooth Module HC-05	1250LKR
Keypad 4x4	250LKR
LDR sensor	215LKR
PIR motion Sensor	395LKR
potentiometer	120LKR

6. Physical Review and code



```
00:34:34.958
00:34:39.403 M
00:34:44.856 M
00:34:44.858 manually light off
00:34:55.693 A
00:34:59.713 *M
00:35:10.103
00:35:10.104
00:35:20.091 M
00:35:20.092 manually light on
00:35:28.012 S
00:35:35.947 *M
00:35:45.245
00:35:45.247
00:35:48.949 S
00:35:55.641 A
00:35:55.642 automatic light mode activated*M
00:36:11.484
00:36:12.474 Motion detected!
00:36:22.864 Motion detected!
00:36:22.870
00:36:22.870
00:36:33.758 Motion stopped!
00:36:33.760 S
00:36:34.255 Study Mode On
00:36:37.717 Set The Hours
00:36:41.678 9hours Entered
00:36:43.162 Enter Minutes! Minutes Entered
00:36:47.122 Enter Interval Time(Min)
00:36:48.113 Interval Entered
00:37:06.426 Interval Started
00:37:09.403 Waiting
00:37:10.387 Waiting
00:37:11.377 Waiting
00:37:12.862 Waiting
00:37:13.951 Waiting
00:37:14.842 00:00:00
00:37:24.098 S
00:37:24.515 *M
00:37:25.507 00:00:01
00:37:35.633
00:37:35.634

Automatic light mode a... Stu... Manual mod... Exa... M M M M
```

```
00:35:58.281 Connecting to Study Buddy ...
00:35:58.684 Connected
00:36:02.683 M
00:36:09.712 M
00:36:09.712 manually light on
00:36:14.210 M
00:36:24.564
00:36:34.958
00:36:34.958
00:36:39.403 M
00:36:44.856 M
00:36:44.858 manually light off
00:36:55.693 A
00:36:59.713 *M
00:37:10.103
00:37:10.104
00:37:20.091 M
00:37:20.092 manually light on
00:37:28.012 S
00:37:35.947 *M
00:37:45.245
00:37:45.247
00:37:48.949 S
00:37:55.641 A
00:37:55.642 automatic light mode activated*M
00:38:11.484
00:38:12.474 Motion detected!
00:38:22.864 Motion detected!
00:38:22.870
00:38:22.870
00:38:33.758 Motion stopped!
00:38:33.760 S
00:38:34.255 Study Mode On
00:38:37.717 Set The Hours
00:38:41.678 9hours Entered
00:38:43.162 Enter Minutes! Minutes Entered
00:38:47.122 Enter Interval Time(Min)
00:38:48.113 Interval Entered
```

Bluetooth Serial Terminal

Refresh List

Study Buddy

Disconnect

Transmit

Send

Receive

>A

A

automatic light mode activated

>A

>A

A

automatic light mode deactivated

>M

>M

A

automatic light mode activated

A

automatic light mode deactivated

M

manually light on

Transmit Line Ending (applies to String format)

No Line Ending

Transmit Format

String

Receive Format

String

Example String Format: ABCDEFG

Example Hex Format: 51 48 23 53 AB

Clear Terminal

Terminal ON

Code

```

/*
 * Name: Hashini Dilhara
 * AS2019912
 * study buddy table lamp
 */

#include "Countimer.h" // includes the timer Library
#include <Keypad.h> // includes the Keypad Library
#include <LiquidCrystal.h> // includes the LiquidCrystal Library

LiquidCrystal lcd(14, 15, 16, 17, 18, 20); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)

Countimer timer;

Countimer timer1ForStudyMode;

Countimer timer2ForStudyMode;

#define buzzerPin 21 //change to any output pin (not analog inputs)

static int hours=0;

static int minutes=0;

const byte ROWS = 4;

const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'E'},
  {'7', '8', '9', 'M'},
  {'*', '0', '#', 'S'}
};

byte rowPins[ROWS] = {9, 8, 7, 6};

byte colPins[COLS] = {5, 4, 3, 2};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

```



```

int ldr_pin=0;//analog pin

int pir_pin=10;//pir pin

int ledpin=12;//light pin


static char Incoming_value='0';


int val;//ldr value

int pir_state = LOW;      // by default, no motion detected

int pir_val = 0;          // variable to store the sensor status (value)


static bool isLedOn=false;

static bool isItAutomatic=false;


static bool isItExamMode=false;

static bool isTimeSet=false;

static bool isHourSet=false;

static bool isMinutesSe=false;


static bool studyMode=false;

static bool timerBreak=false;

static int  studyModeMin;

static int  studyModeHours;

static int  studyModeInterval;


void checkTheAvaliableLight();

void readSerialValues();

void ledOnOffFunction(bool ledOnOffState);

void checkTheMotion();

void refreshClock();

void keypadInputForTimerSet(bool selectVariable);

void readKeypadInputsForModes();

void turnOnOffStudyMode();

void turnOnOffAutomaticLightMode();

void turnOnOffManualLightMode();

void turnOnOffExamMode();

```

```

void activateBuzzer();

void timerTerminator();

void alertTheInterval();

void setTheTimerForStudyMode(int studyModeHours,int studyModeMin,int studyModeSec,int intervalInMin);

void alertTheEndOfTheStudyMode();

void displayTheCurrentTimerDetails();

void StudyModeTimerTerminator();

```

```

void StudyModeTimerTerminator(){

    studyMode=false;

    timer1ForStudyMode.stop();

    timer2ForStudyMode.stop();

    Serial.println("Study Mode Off");

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Study Mode Off");

    delay(3000);

    lcd.clear();

}

```

```

void turnOnOffStudyMode(){

    if(!studyMode){//set the new timer

        //turn on the study mode

        lcd.setCursor(0,0);

        lcd.print("Study Mode On");

        Serial.println("Study Mode On");

        delay(4000);

        lcd.clear();

        Serial.println("Set The Hours");

        lcd.setCursor(0,0);

        lcd.print("Set The Hours");

        keypadInputForStudyModeTimerSet(1);

        lcd.clear();

        Serial.println("Hours Entered");

        lcd.setCursor(0,0);
    }
}

```

```

lcd.print("Hours Entered");
delay(1000);
lcd.clear();
Serial.print("Enter Minutes");
lcd.setCursor(0,0);
lcd.print("Enter Minutes");
keypadInputForStudyModeTimerSet(2);
lcd.clear();
lcd.print("Minutes Entered");
Serial.println("Minutes Entered");
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Enter Interval Time(Min)");
Serial.println("Enter Interval Time(Min)");
keypadInputForStudyModeTimerSet(3);
lcd.clear();
lcd.print("Interval Entered");
Serial.println("Interval Entered");
delay(1000);
lcd.clear();
setTheTimerForStudyMode(studyModeHours,studyModeMin,0,studyModeInterval);
studyMode=true;
}else if(studyMode){
    StudyModeTimerTerminator();
}
}

```

```

void setTheTimerForStudyMode(int studyModeHours,int studyModeMin,int studyModeSec,int intervalInMin){
    timer1ForStudyMode.setCounter(studyModeHours, studyModeMin,studyModeSec,timer1ForStudyMode.COUNT_UP,
alertTheEndOfTheStudyMode);

    timer2ForStudyMode.setCounter(studyModeHours, studyModeMin,studyModeSec,timer2ForStudyMode.COUNT_UP,
alertTheEndOfTheStudyMode);

    //when every defined interval this alertTheInterval start

    timer1ForStudyMode.setInterval(alertTheInterval,intervalInMin*1000*60);

    //display the time in every seconds

    timer2ForStudyMode.setInterval(displayTheCurrentTimerDetails,1000);
}

```

```

timer1ForStudyMode.start();
timer2ForStudyMode.start();
}

void displayTheCurrentTimerDetails(){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(timer2ForStudyMode.getCurrentTime());
    Serial.println(timer2ForStudyMode.getCurrentTime());
}

void alertTheInterval(){
    //pause the timer2 for interval
    timer1ForStudyMode.pause();
    timer2ForStudyMode.pause();

    activateBuzzer();
    activateBuzzer();
    activateBuzzer();
    activateBuzzer();

    Serial.println("Interval Started");
    lcd.setCursor(0,1);
    lcd.print("Interval Started");
    delay(3000);
    lcd.clear();
    int interval=5000;
    while(interval!=0){
        lcd.setCursor(0,1);
        lcd.print("Wating");
        Serial.println("Wating");
        delay(1000);
        lcd.clear();
        interval=interval-1000;
    }

    timer1ForStudyMode.start();
    timer2ForStudyMode.start();

```

```
}
```

```
void alertTheEndOfTheStudyMode(){
```

```
    Serial.println("Timer Completed");
```

```
    lcd.clear();
```

```
    lcd.print("Timer Completed");
```

```
    activateBuzzer();
```

```
    activateBuzzer();
```

```
    activateBuzzer();
```

```
    activateBuzzer();
```

```
    delay(3000);
```

```
    lcd.clear();
```

```
}
```

```
void timerTerminator(){
```

```
    isTimeSet=false;
```

```
    isItExamMode=false;
```

```
    timer.stop();
```

```
    Serial.println("Exam Mode Off");
```

```
    lcd.clear();
```

```
    lcd.setCursor(0,0);
```

```
    lcd.print("Exam Mode Off");
```

```
    delay(3000);
```

```
    lcd.clear();
```

```
}
```

```
void activateBuzzer(){
```

```
    for(int i=0; i<255; i++) { //do this 255 times
```

```
        analogWrite(buzzerPin, i); //raise the voltage sent out of the pin by 1
```

```
        delay(10); //wait 10 milliseconds to repeat
```

```
    }
```

```
    for(int i=125; i>0; i--) { // do this 255 times
```

```
        analogWrite(buzzerPin, i); //lower the voltage sent out of the pin by 1
```

```
        delay(10); //wait 10 milliseconds to repeat
```

```

}
}

void turnOnOffExamMode(){
    if(!isItExamMode){//set the new timer
        //turn on the exam mode
        lcd.setCursor(0,0);
        lcd.print("Exam Mode On");
        Serial.println("Exam Mode On");
        delay(4000);
        lcd.clear();
        lcd.setCursor(0,0);
        Serial.println("Set The Hours");
        lcd.setCursor(0,0);
        lcd.print("Set The Hours");
        keypadInputForTimerSet(true);
        lcd.clear();
        Serial.println("Hours Entered");
        lcd.setCursor(0,0);
        lcd.print("Hours Entered");
        delay(1000);
        lcd.clear();
        Serial.print("Enter Minutes");
        lcd.setCursor(0,0);
        lcd.print("Enter Minutes");
        keypadInputForTimerSet(false);
        lcd.clear();
        lcd.print("Minutes Entereds");
        Serial.println("Minutes Entered");
        delay(1000);
        lcd.clear();
        timer.setCounter(hours,minutes,0, timer.COUNT_DOWN, onComplete);
        isItExamMode=true;
        // Print current time every 1s on serial port by calling method refreshClock().
        timer.setInterval(refreshClock, 1000);
        timer.start();
    }
}

```

```

}else if(isItExamMode){
    timerTerminator();    } }

void turnOnOffManualLightMode(){
    isItAutomatic=false;
    //turn off automatic led mode
    if(!isLedOn){
        //turn on the led if its not on
        Serial.println("manually light on");
        ledOnOffFunction(true);
        lcd.clear();
        lcd.print("Light ON");
        delay(5000);
        lcd.clear();
    }else{
        //turn off the led if its on
        Serial.println("manually light off");
        ledOnOffFunction(false);
        lcd.clear();
        lcd.print("Light OFF");
        delay(5000);
        lcd.clear();
    }
}

void turnOnOffAutomaticLightMode(){

    //automatic light on
    if(isItAutomatic==false){

        isItAutomatic=true;

        Serial.println("automatic light mode actiavted");

        lcd.clear();
        lcd.print("A-light ON");
        delay(5000);
        lcd.clear();
    }else{

```

```
//automatic light off

isItAutomatic=false;

if(isLedOn){
    ledOnOffFunction(false);
}

Serial.println("automatic light mode deactiavted");

lcd.clear();

lcd.print("A-light OFF");

delay(5000);

lcd.clear();  }

}

void refreshClock() {

    Serial.print("Current count time is: ");

    Serial.println(timer.getCurrentTime());

    lcd.clear();

    lcd.print(timer.getCurrentTime());

}

void keypadInputForTimerSet(bool selectVariable){

    String inputChar;

    int digit=0;

    while(1){

        char customKey = customKeypad.getKey();

        if (customKey and customKey!='#'){

            inputChar+=customKey;

            Serial.print(customKey);

            if(selectVariable==true){//display the user input for hours

                lcd.setCursor(digit,1);

            }

            if(selectVariable==false){//display the user input for minutes
```



```

        lcd.setCursor(digit+4,1);
    }
    lcd.print(customKey);
    digit=digit+1;
}

if(customKey=='#' || digit==2){
    lcd.clear();
    if(selectVariable==true){
        hours=inputChar.toInt();
        isHourSet=true;
    }

    if(selectVariable==false){
        minutes=inputChar.toInt();
        isMinutesSe=true;
    }
    digit=0;
    break;

} } }

void keypadInputForStudyModeTimerSet(int selectVariable){
    String inputChar;
    int digit=0;
    while(1){

        char customKey = customKeypad.getKey();

        if (customKey and customKey!='#'){
            inputChar+=customKey;
            Serial.print(customKey);
            if(selectVariable==1){//display the user input for hours
                lcd.setCursor(digit,1);

            }

```

```

if(selectVariable==2){display the user input for minutes

    lcd.setCursor(digit+3,1);
}

if(selectVariable==3){display the user input for interval time in minutes

    lcd.setCursor(digit+6,1);
}

lcd.print(customKey);
digit=digit+1;

}

```

```

if(customKey=='#' || digit==2){

    lcd.clear();

    if(selectVariable==1){

        studyModeHours=inputChar.toInt();

        isHourSet=true;

    }

```

```

if(selectVariable==2){

    studyModeMin=inputChar.toInt();

    isMinutesSe=true;

}

if(selectVariable==3){

    studyModeInterval=inputChar.toInt();

    isMinutesSe=true;

}

digit=0;

break;

} } }

```

```

void setup() {

    Serial.begin(9600);beign the serial communication

```

```

lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display

//loading screen start

lcd.setCursor(4,0);

lcd.print("Study");

lcd.setCursor(4,1);

lcd.print("Buddy");

delay(4000);

lcd.clear();

//loading screen end

pinMode(buzzerPin, OUTPUT); //tell arduino the buzzer is an output device

pinMode(ledpin,OUTPUT);

pinMode(pir_pin, INPUT); // initialize sensor as an input

delay(1000);

}

//check the ldr light sensitivity

void checkTheAvaliableLight(){

    if(!isItAutomatic){

        val=analogRead(ldr_pin);

        if(val>=750){

            ledOnOffFunction(true);

        }else{

            ledOnOffFunction(false);

        }

        delay(1000);

    }

}

//timer on complete

void onComplete() {

    isTimeSet=false;

    activateBuzzer();

    Serial.println("Complete!!!");

}

```

```

void setTimer(int hours,int minutes){
    if(!isTimeSet){
        timer.setCounter(hours,minutes,0, timer.COUNT_DOWN, onComplete);
    }else{
        Serial.println("Already Time Set");
    } }

void readKeypadInputsForModes(){

char customKey = customKeypad.getKey();

if (customKey){
    //turn on or off study mode
    if(customKey=='S'){
        turnOnOffStudyMode();
    }

    //turn on or off exam mode
    if(customKey=='E'){
        turnOnOffExamMode();
    }

    //turn on or off manual mode
    if(customKey=='M'){
        turnOnOffManualLightMode();
    }

    //turn on or off automatic mode
    if(customKey=='A'){
        turnOnOffAutomaticLightMode();
    } } }

void readSerialValues(){
    String userInput;

    if(Serial.available()>0){

        Incoming_value = Serial.read(); //Read the incoming data and store it into variable Incoming_value
    }
}

```

```
Serial.println(Incoming_value);
```

```
//automatic light on off via ble or serial communication
```

```
if(Incoming_value=='A'){
```

```
    turnOnOffAutomaticLightMode();
```

```
}
```

```
//manual light on off via ble or serial communication
```

```
if(Incoming_value=='M'){
```

```
    turnOnOffManualLightMode();    }
```

```
if(Incoming_value=='S'){
```

```
    turnOnOffStudyMode();
```

```
}
```

```
if(Incoming_value=='E'){
```

```
    //check the study mode or not
```

```
    } } }
```

```
void ledOnOffFunction(bool ledOnOffState){
```

```
    if(ledOnOffState){
```

```
        digitalWrite(ledpin,HIGH);
```

```
        isLedOn=true;
```

```
    }else{
```

```
        digitalWrite(ledpin,LOW);
```

```
        isLedOn=false;
```

```
    } }
```

```
void checkTheMotion(){
```

```
    pir_val = digitalRead(pir_pin); // read sensor value
```

```
    if (pir_val == HIGH) {        // check if the sensor is HIGH
```

```
        Serial.println("Motion detected!");
```

```
        delay(10000);            // delay 10000 milliseconds
```

```
    if (pir_state == LOW) {
```

```
        Serial.println("Motion detected!");
```

```
        pir_state = HIGH;        // update variable state to HIGH
```

```
    }
```

```

}

else {

    //digitalWrite(led, LOW); // turn LED OFF

    delay(10000);    // delay 10000 milliseconds

    if (pir_state == HIGH){

        Serial.println("Motion stopped!");

        pir_state = LOW;    // update variable state to LOW

    }

}

}

void loop() {

    readSerialValues();

    readKeypadInputsForModes();

    checkTheAvaliableLight();

    // Run timer

    timer.run();

    timer1ForStudyMode.run();

    timer2ForStudyMode.run();

    // Now timer is running and listening for actions.

    // If you want to start the timer, you have to call start() method.

    checkTheMotion();

}

```

7.Issues

There is a motion sensor to determine whether or not the kids are sleeping. However, the motion sensor is unable to recognize the youngster. Any object's motion can be detected by it.

On Bluetooth Terminal, there is a problem. When it's functioning, every output has the "^M" symbol.

8.Discusstion

This study lamp is simple to use with the Pomodoro method. For the panel, I utilized 9 white LED bulbs. However, anyone can use a light bulb for the lamp, I favor led bulbs. We cannot use a bulb directly with an Arduino since they operate on different voltages (5V DC for Arduino and AC for bulbs). Typically, a yellow light bulb works well for lamps.

9.Conclusion

If you're a student, you already know that sometimes it's quite hard to stay focused on studying. However, you should consider utilizing a time tracker when studying to restore your focus and be more effective. So, if you keep a time diary, you'll be able to see how much time you spend studying. Additionally, it will be lot simpler for you to keep track of your deadlines and progress. Additionally, you can experiment with the Pomodoro method.

10.Reference

https://en.wikipedia.org/wiki/Pomodoro_Technique

<https://todoist.com/productivity-methods/pomodoro-technique>

<https://github.com/inflop/Countimer>

