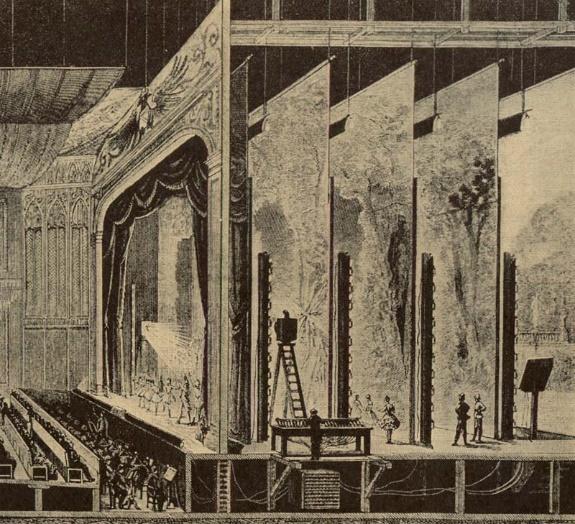
CHAPTER 1

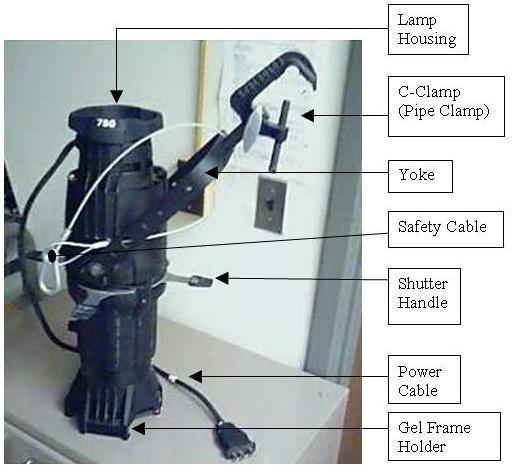
Introduction

* 1. Background
* The earliest known form of **stage lighting** was during the early Grecian (and later the Roman) theaters. They would build their theatres facing east to west so that in the afternoon they could perform plays and have the **natural sunlight** hit the actors, but not those seatedin the orchestra.





* Year by year from sunlight to **candles** and from candles to **oil lamps** , stage lighting systems got diversified, developed and modified.
* From sunlight to automated LED’s, the journey of the stage lighting system has been changing day by day and is becoming more functional and convenient.
  1. Relevance
* This project is based on a stage lighting system. We have worked on automating the lighting used to focus on a particular performer.
* Using suitable sensors interfaced with a microcontroller, stage lighting systems can be more efficient, automatic and effective.
* This project will help improve the experience of performers as well as production houses in the theatre business who can save on the lighting expenses as power is saved and also it will be a visual treat for the audience who have no obstruction in their view of the performer.
  1. Literature Survey
* In 1878, Joseph Swan patented the world’s first incandescent **electric lamp**. Gas was quickly out of the theatres and within one year the Paris Opera introduced this new type of lighting.
* LED lights which started to become popular in 2007 allowed for the colour of light to be changed from a console without any lighting gel or material needed. These lights instantly change colour and although more expensive, allow theatres to cut down on the amounts of lights needed.
* **More recently moveable** LED lights are coming to the stages. These lights can be programmed to move to a certain spot on stage and cut down the need for hundreds of lights even more.
* Stage lighting is very important in theatrical performances. But in the actual process of stage lighting control, there is a deviation between the expected lighting scene and the actual. Stage lighting operator needs a lot of debugging before or on the show. **Automatic tracking** can be used to detect whether there is an object movement, and tracking. In this way, we can keep the subject in the range of illumination without the manual adjustment. At the same time, it can also be switched to manual operation, so as to change the detection area or deal with special circumstances.
* The great advantage of the Track Systems is the flexibility of the system. Install it to fit almost any stage set-up with the need to create moving stage props or lights. Create intelligent stage lighting by implementing Track Systems in your basic scene design.



* 1. Motivation

As we all know lights play an important role in catching the attention of the audience in any stage performance. In the theatre there is a single man who operates the spotlight so we thought why not automate and make the task easier.

* 1. Aim of the Project

Design and develop a motion detecting spotlight.

* 1. Scope and Objectives
* Study the existing spotlight system in a theatre to identify limitations and involve electronics control for it.
* Design and develop effective system to address problems identified in study
* Test and Verify spotlight system for its effectiveness.
  1. Technical Approach

RGB LED should be mounted on rack and pinion arrangement.

A PIR sensor is used to detect motion on the right and left part of the stage which is analysed using a microcontroller.

Microcontroller controls the motion of the LED lamp horizontally across the stage using a stepper motor.

CHAPTER 2

Introduction

2.1 Introduction

Lighting is a critical element of theater.Modern stages contain tens to hundreds of lights, and setting each light source's parameters individually is both tedious and requires expert skill.So by using electronic components and gear assembly, we have made the task easier and automated.

2.2 Block diagram

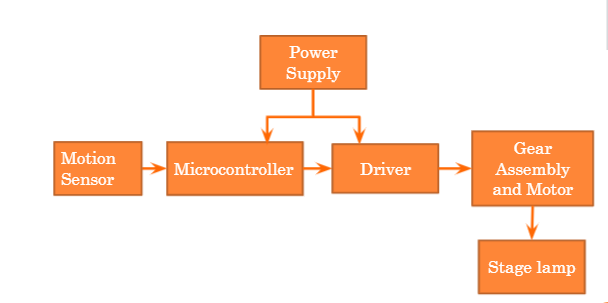
****

Fig.1. Block schematic of system

* + 1. **Motion Sensor** will detect the motion on the stage and will send the signal to the microcontroller.
    2. **Microcontroller** after receiving the signal , will control the stepper motor using the ULN2003 driver.
    3. **Driver** is used to provide required amount of current to the stepper motor assembly.
    4. **LED lamp** will be placed on the gear and motor assembly which will move based upon the movement on the stage.

CHAPTER 3

System design

3.1 Detailed Design

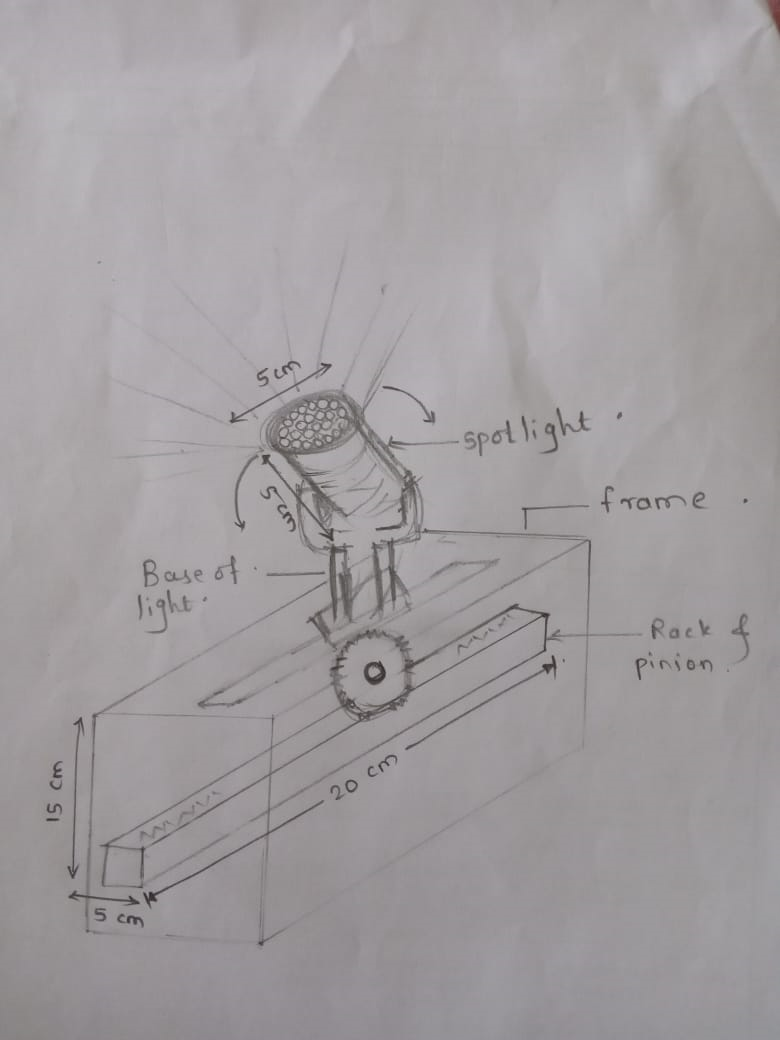
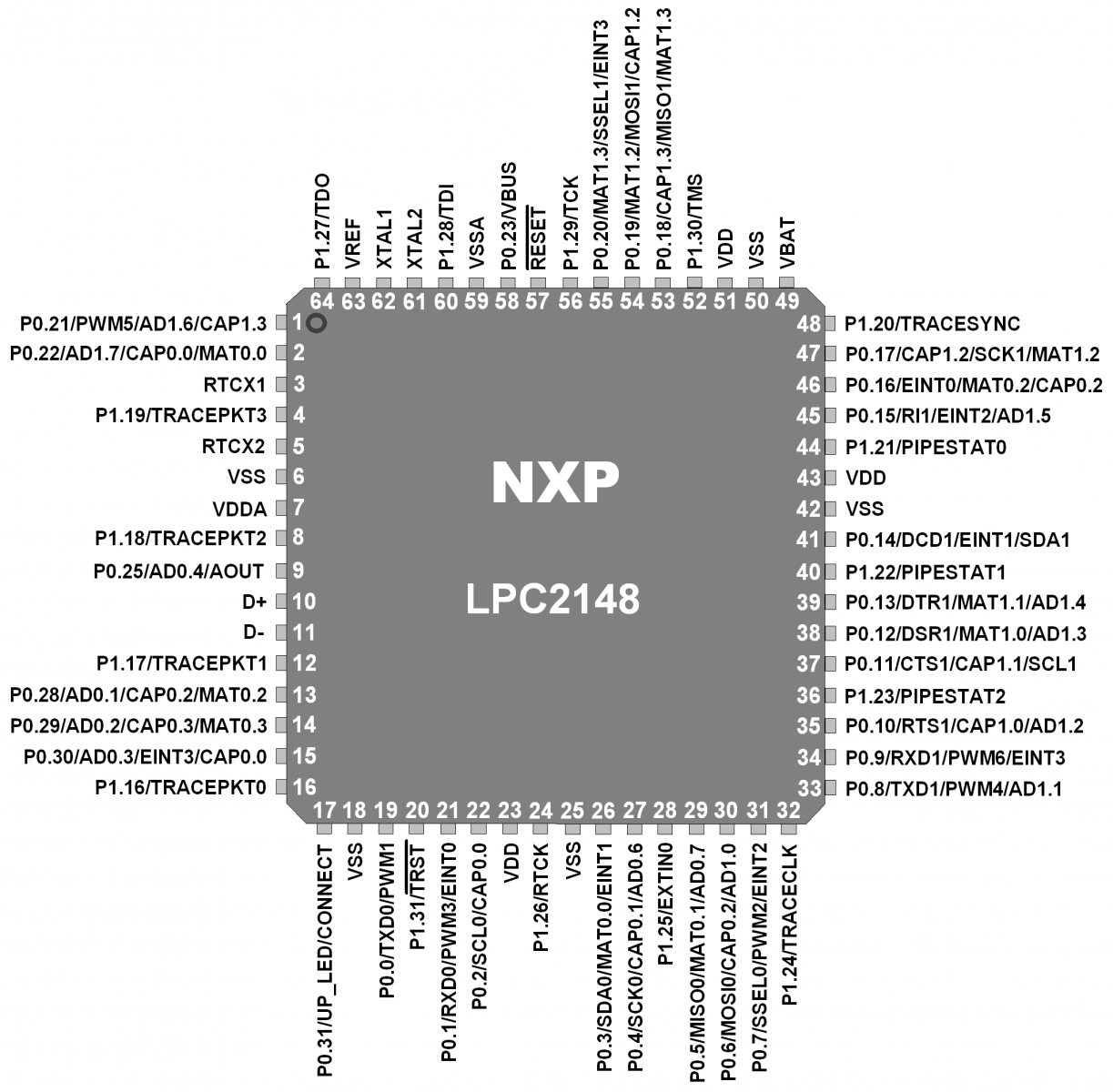


Fig.2. Detailed Diagram of system

3.2 Selection of components

Microcontroller (LPC 2148)

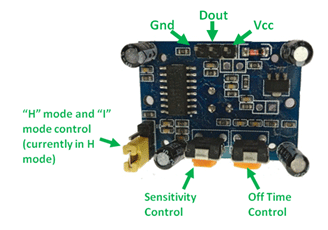
32 bit microcontroller

45 I/O pins

Supply voltage:0.5 - 3.6 V

Stepper Motor(28BYJ48)

Voltage: 0-12V DC

Phase: 4

Resistance: 300ohm

Motion Sensor (HC-SR501):

Voltage: 5-20V

Sensing Range: less than 120deg,within 7m

Operating Temperature: 15deg C-70deg C

Rack and Pinion Gear Set

RGB LED Lamp

Power: 9W

Voltage:110V-240V/50-60Hz

Width:5cm

Height:5cm

For model we will be using led lamp with above specification. In auditorium LED par lights which are of high power rating and bigger dimension are used.

3.3 Working of each subcircuit

LPC2148 with PIR Sensor :

A passive infrared sensor (PIR sensor) is an electronic [sensor](https://en.wikipedia.org/wiki/Sensor) that measures [infrared](https://en.wikipedia.org/wiki/Infrared) (IR) light radiating from objects in its field of view. PIR sensors detect general movement, but do not give information on who or what moved. outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected which is converted to digital signal, then it is given to the microcontroller.

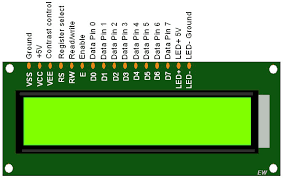
LPC2148 with stepper motor:

Stepper motor is a brushless DC motor, which can be rotated in small angles, these angles are called steps. We can rotate the stepper motor step by step by giving digital pulses to its pins. Stepper motors are inexpensive and have a rugged design. Speed of the motor can be controlled by changing frequency of digital pulses.

But we need a Motor Drive IC like ULN2003 to drive it, because stepper motors consume high current and it may damage microcontrollers.

LPC2148 with LCD:

Depending upon the signal received from the microcontroller and the movement of stepper motor LCD will display the message to which side of the stage the spotlight is moving.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pin number** | **Symbol** | **Level** | **I/O** | **Function** |
| 1 | Vss | - | - | Power supply (GND) |
| 2 | Vcc | - | - | Power supply (+5V) |
| 3 | Vee | - | - | Contrast adjust |
| 4 | RS | 0/1 | I | 0 = Instruction input  1 = Data input |
| 5 | R/W | 0/1 | I | 0 = Write to LCD module  1 = Read from LCD module |
| 6 | E | 1, 1->0 | I | Enable signal |
| 7 | DB0 | 0/1 | I/O | Data bus line 0 (LSB) |
| 8 | DB1 | 0/1 | I/O | Data bus line 1 |
| 9 | DB2 | 0/1 | I/O | Data bus line 2 |
| 10 | DB3 | 0/1 | I/O | Data bus line 3 |
| 11 | DB4 | 0/1 | I/O | Data bus line 4 |
| 12 | DB5 | 0/1 | I/O | Data bus line 5 |
| 13 | DB6 | 0/1 | I/O | Data bus line 6 |
| 14 | DB7 | 0/1 | I/O | Data bus line 7 (MSB) |
| 15 | VB+ | 1 | - | Backlight Supply |
| 16 | VB- | 0 | - |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Command to LCD** | **Code (Hex)** |
| 01 | Clear display screen | 01 |
| 02 | Return home | 02 |
| 03 | Decrement cursor (shift cursor to left) | 04 |
| 04 | Increment cursor (shift cursor to right) | 06 |
| 05 | Shift display right | 05 |
| 06 | Shift display left | 07 |
| 07 | Display off, cursor off | 08 |
| 08 | Display off, cursor on | 0A |
| 09 | Display on, cursor off | 0C |
| 10 | Display on cursor blinking | 0E |
| 11 | Shift cursor position to left | 10 |
| 12 | Shift cursor position to right | 14 |
| 13 | Shift the entire display to left | 18 |
| 14 | Shift the entire display to right | 1C |
| 15 | Force cursor to beginning of 1st line. | 80 |
| 16 | Force cursor to beginning of 2nd line. | C0 |
| 17 | 2 lines and 5x7 matrixes. | 38 |

CHAPTER 4

Implementation, Testing and Debugging

4.1 Implementation

Since components were not available due to the Covid 19 pandemic, entire implementation is done using Proteus Simulation.

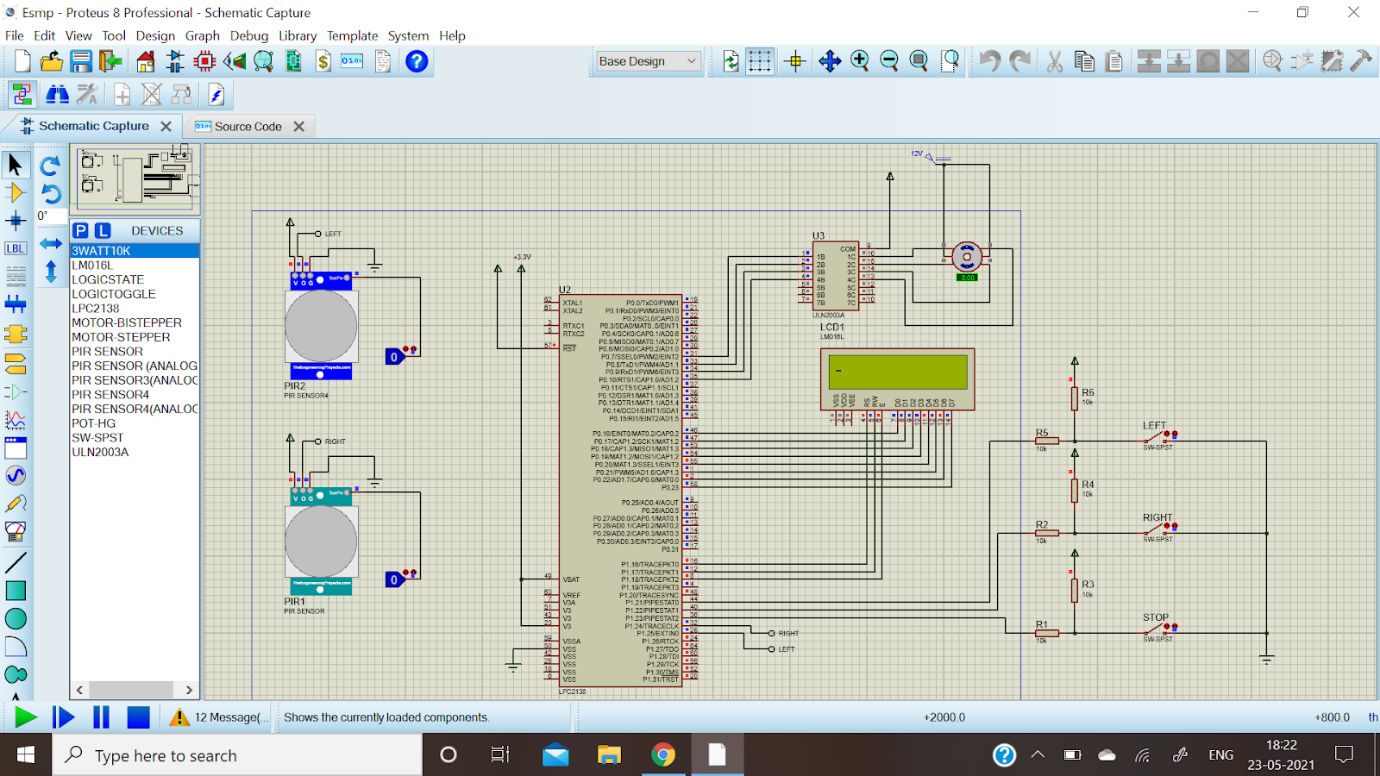


Fig.3. Implementation/Testing of system

4.2 Testing, Debugging

Initially switches are used in place of PIR Sensor and output is checked on LCD without the Stepper Motor and ULN2003 Driver.

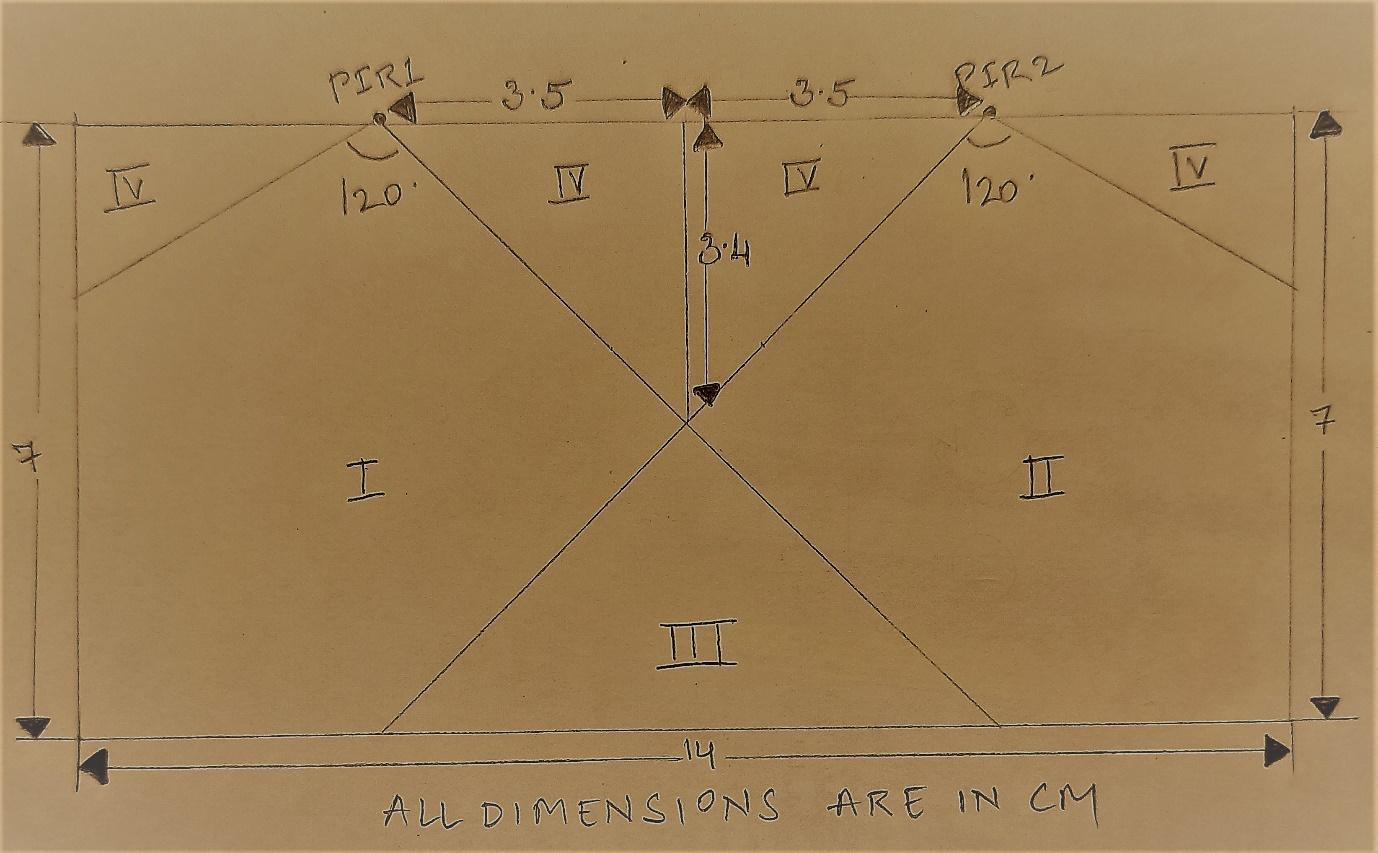


Fig.4. Testing of PIR sensor

According to the datasheet of PIR Sensor HC-SR501, the sensing range is 120 degrees and within 7 meters.

From the above diagram it is observed that for

REGION 1: PIR1 sensor will detect motion on the Left part of the stage hence the Lamp will move Left.

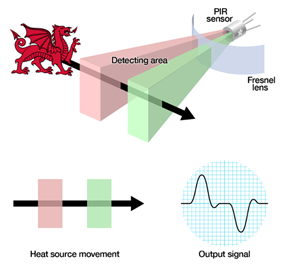
REGION 2: PIR2 sensor will detect motion on the Right part of the stage hence the Lamp will move Right.

REGION 3: Both PIR sensors will detect motion on stage hence the Lamp stays where it is.

REGION 4: Both PIR sensors will not detect motion on stage since it is out of range hence the Lamp stays where it is.

4.3 Simulation

Observations



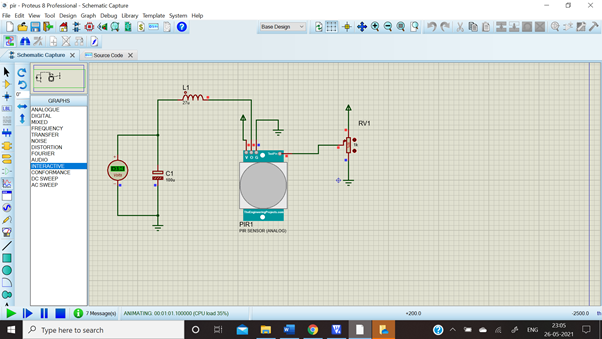


Fig.5. Testing/simulation of PIR sensor

|  |  |  |
| --- | --- | --- |
| Sr. No. | Potentiometer Resistance(ohm) | Voltage(V) |
| 1 | 10% = 100 | 0.50 |
| 2 | 20% = 200 | 1.03 |
| 3 | 30% = 300 | 1.50 |
| 4 | 40% = 400 | 1.97 |
| 5 | 50% = 500 | 2.46 |
| 6 | 60% = 600 | 2.97 |
| 7 | 70% = 700 | 3.45 |
| 8 | 80% = 800 | 3.94 |
| 9 | 90% = 900 | 4.46 |
| 10 | 100% = 1000 | 4.97 |

Table 1. Result of PIR sensor simulation

The motion of the performer will be detected by the PIR Sensor if he is present in its range

(120 degree and within 7m).

Detection of motion is treated as HIGH (digital value 1) ,that is , voltage above 2.46V.

Detection of no motion is treated as LOW (digital value 0) ,that is , voltage below 2.46V.

|  |  |
| --- | --- |
| Voltage(V) | Angle subtended by the performer on sensor |
| 2.46 | 40 degree |
| 2.97 | 50 degree |
| 3.45 | 60 degree |
| 3.94 | 70 degree |
| 4.46 | 80 degree |
| 4.97 | 90 degree |
| 4.46 | 100 degree |
| 3.94 | 110 degree |
| 3.45 | 120 degree |
| 2.97 | 130 degree |
| 2.46 | 140 degree |

Table 2. Result of PIR sensor simulation

Detection is most accurate when the performer is at 90 degree to the PIR Sensor, that is, value HIGH at 4.97V.

For all other voltage values above 2.46V, motion will be detected when the object is placed on either side of 90 degree.

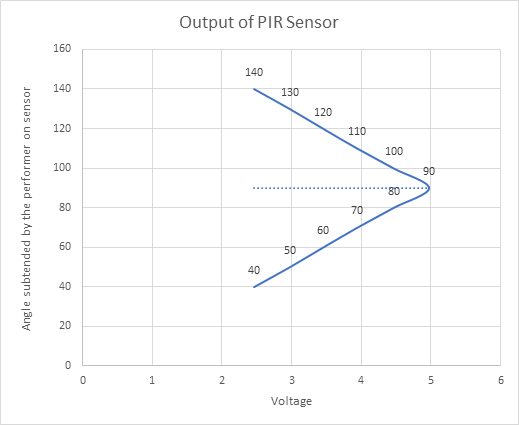
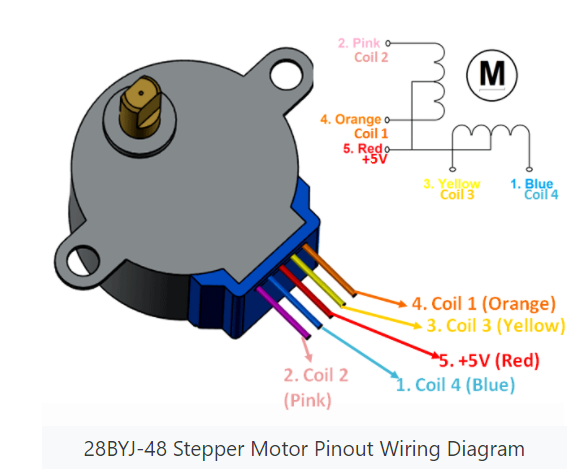


Fig 6.Graph of PIR sensor simulation



|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No | Parameters of Stepper Motor | Value | Unit |
| 1 | Type: 3-wire, 4-wire, 5 wire | 4 | wire |
| 2 | Type: Permanent magnet (PM)/ Variable reluctance (VR)/ Hybrid | Permanent magnet |  |
| 3 | Colour codes of wire | 1.Orange  2.Pink  3.Yellow  4.Blue  5.Red |  |
| 4 | DC voltage required | 5 | V |
| 5 | Coil resistance | 50 | Ohm |
| 6 | Current required for each coil (V/R) | 0.1 | A |
| 7 | Pulse frequency | 100 | Hz |
| 8 | Pulse width | 0.01 | S |
| 9 | Duty cycle | 50 | % |
| 10 | Min rotation angle | 5.625 | o |
| 11 | No. of pulses required for 1 rotation (360 o) | 2048 | -- |
| 12 | No. of steps required to achieve 90 o rotation | 8 |  |
| 13 | No. of pulses required for 1 step (0.72 o) | 1 |  |
| 14 | Frequency for 10 rotation /min |  | Hz |
| 15 | Revolution per minute = 60 x steps per second/ steps per revolution | 1920 |  |

Table 3. Stepper Motor Observations

Code

#include <LPC214X.h>

void msdelay(int c);

void lcd\_cmd(unsigned char cmd);

void lcd\_data(unsigned char data);

void lcd\_write\_string(unsigned char \*str);

void lcd\_init(void);

void lcd\_init(){

lcd\_cmd(0x38);

msdelay(1000);

lcd\_cmd(0x0E);

msdelay(1000);

lcd\_cmd(0x01);

msdelay(1000);

lcd\_cmd(0x80);

msdelay(1000);

}

void lcd\_cmd(unsigned char cmd){

IOCLR0=0xFF<<16;

IOCLR1=0x01<<16;

IOCLR1=0x01<<17;

IOSET0=cmd<<16;

IOSET1=0x01<<18;

msdelay(1000);

IOCLR1=0x01<<18;

}

void lcd\_data(unsigned char data){

IOCLR0=0xFF<<16;

IOSET1=0x01<<16;

IOCLR1=0x01<<17;

IOSET0=data<<16;

IOSET1=0x01<<18;

msdelay(1000);

IOCLR1=0x01<<18;

}

void lcd\_write\_string(unsigned char \*str){

int i=0;

while(str[i]!='\0')

{

lcd\_data(str[i]);

i++;

}

}

void msdelay(int c)

{

unsigned int i,j;

for(i=0;i<c;i++)

{

for(j=0;j<165;j++)

{

}

}

}

int main()

{

unsigned char anticlockwise[4] = {0x01,0x02,0x04,0x08};

unsigned char clockwise[4] = {0x08,0x04,0x02,0x01};

int no\_of\_steps = 2; //value for required number of steps rotation

unsigned int sw0,sw1,sw2,i,j,PIR1,PIR2;

unsigned char var1[]={"Moving Left"};

unsigned char var2[]={"Moving Right"};

unsigned char var3[]={"Paused"};

PINSEL0=0x00000000;

PINSEL1=0x00000000;

PINSEL2=0x00000000;

IODIR0=0xFFFFFFFF;

IODIR1=0x000F0000;

IO0CLR = 0x0000F780;

lcd\_init();

while(1)

{

PIR1 = (IO1PIN & (1<<24));//right

PIR2 = (IO1PIN & (1<<25));//left

/\*

sw0=(IOPIN1 & 0x00000000);

sw1=(IOPIN1 & 0x00000000);

sw2=(IOPIN1 & 0x00800000);

if(sw0==0)

val=1;

if(sw1==0)

val=2;

if(sw2==0)

val=3;

\*/

if(PIR1==0 && PIR2==0)

{

lcd\_cmd(0x01);

lcd\_write\_string(var3);

for(j=0; j<no\_of\_steps; j++)

{

for(i=0; i<4; i++)

{

IOPIN0 = 0x00;

}

}

}

else if(PIR1==0)

{

lcd\_cmd(0x01);

lcd\_write\_string(var1);//left

for(j=0; j<no\_of\_steps; j++)

{

for(i=0; i<4; i++)

{

IOPIN0 = anticlockwise[i]<<7;

msdelay(10000);

}

}

}

else if(PIR2==0)

{

lcd\_cmd(0x01);

lcd\_write\_string(var2);//right

for(j=0; j<no\_of\_steps; j++)

{

for(i=0; i<4; i++)

{

IOPIN0 = clockwise[i]<<7;

msdelay(10000);

}

}

}

else

{

lcd\_cmd(0x01);

lcd\_write\_string(var3);

for(j=0; j<no\_of\_steps; j++)

{

for(i=0; i<4; i++)

{

IOPIN0 = 0x00;

}

}

}

}

}

4.4 Final project working

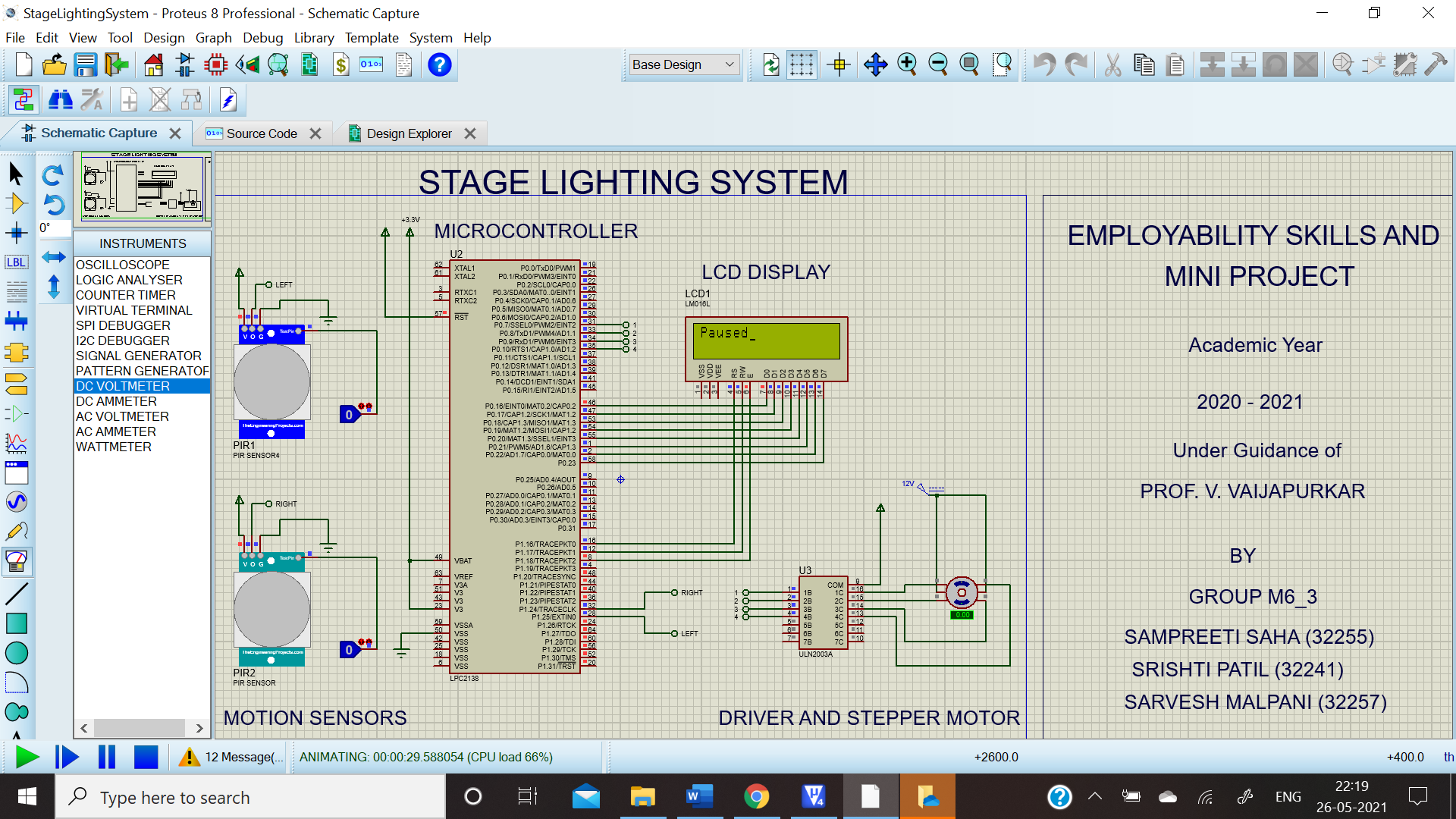


Fig.8. Output of Paused System when no motion detected

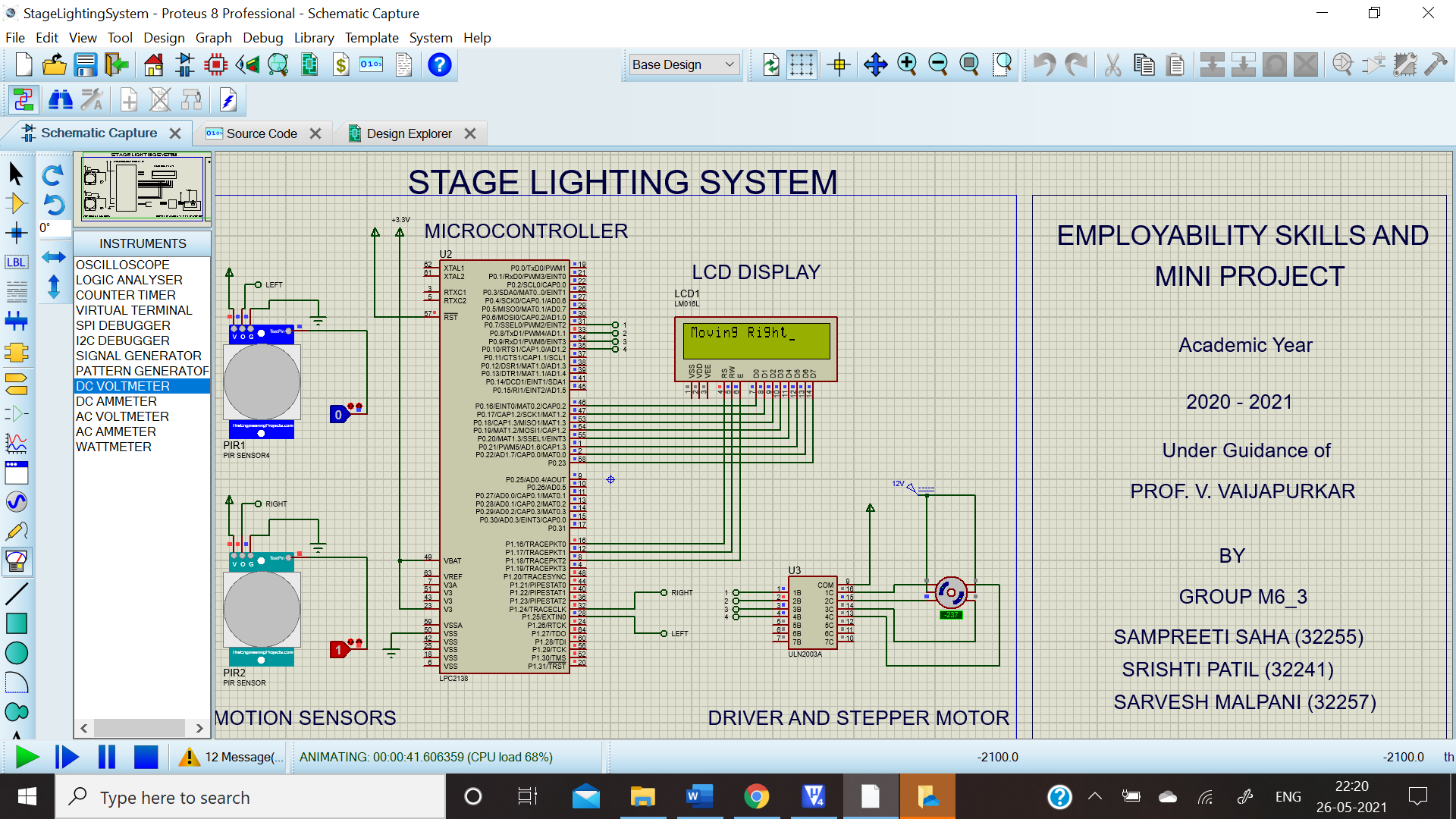


Fig.9. Output of System moving right

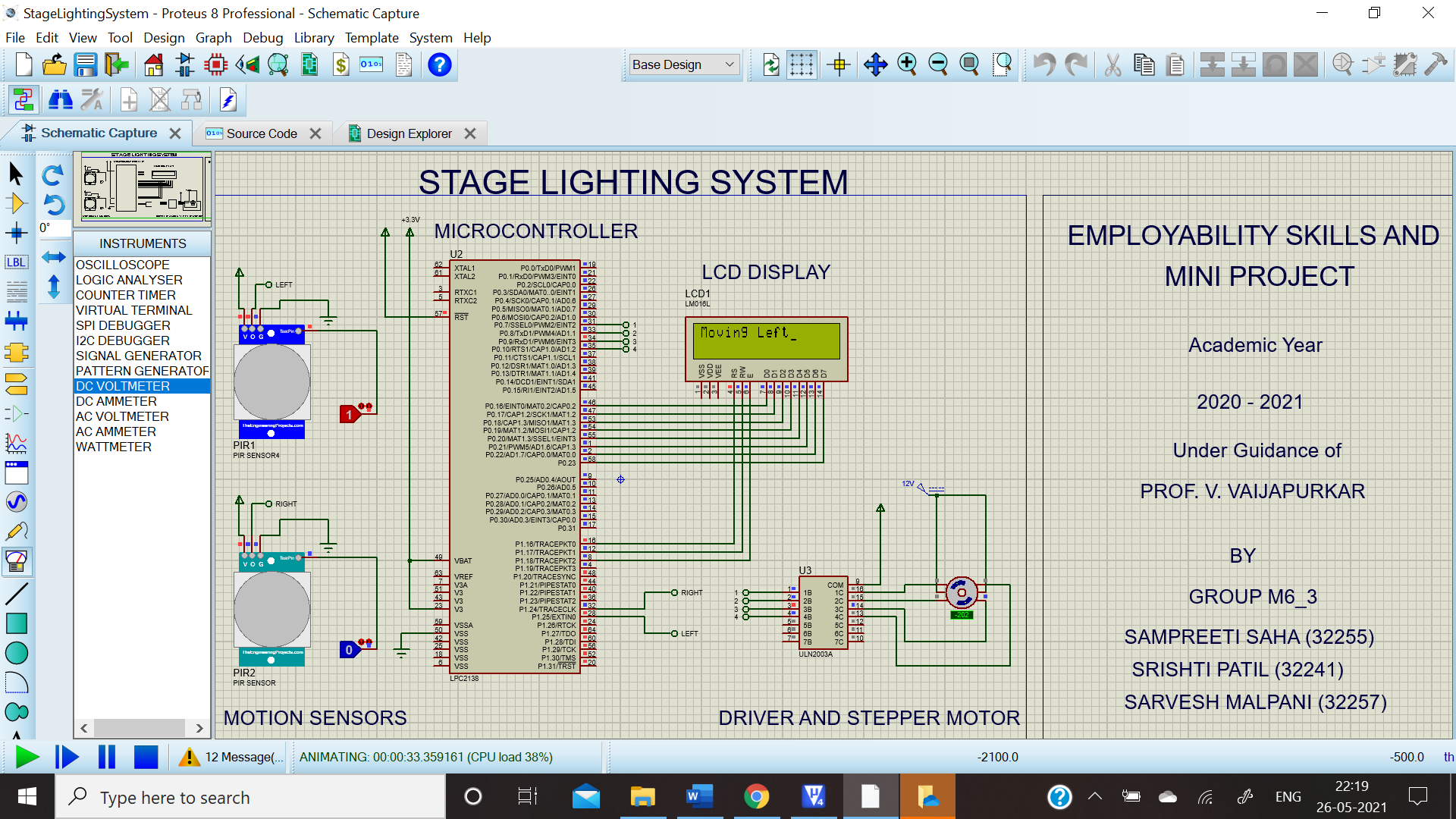


Fig.10. Output of System moving left

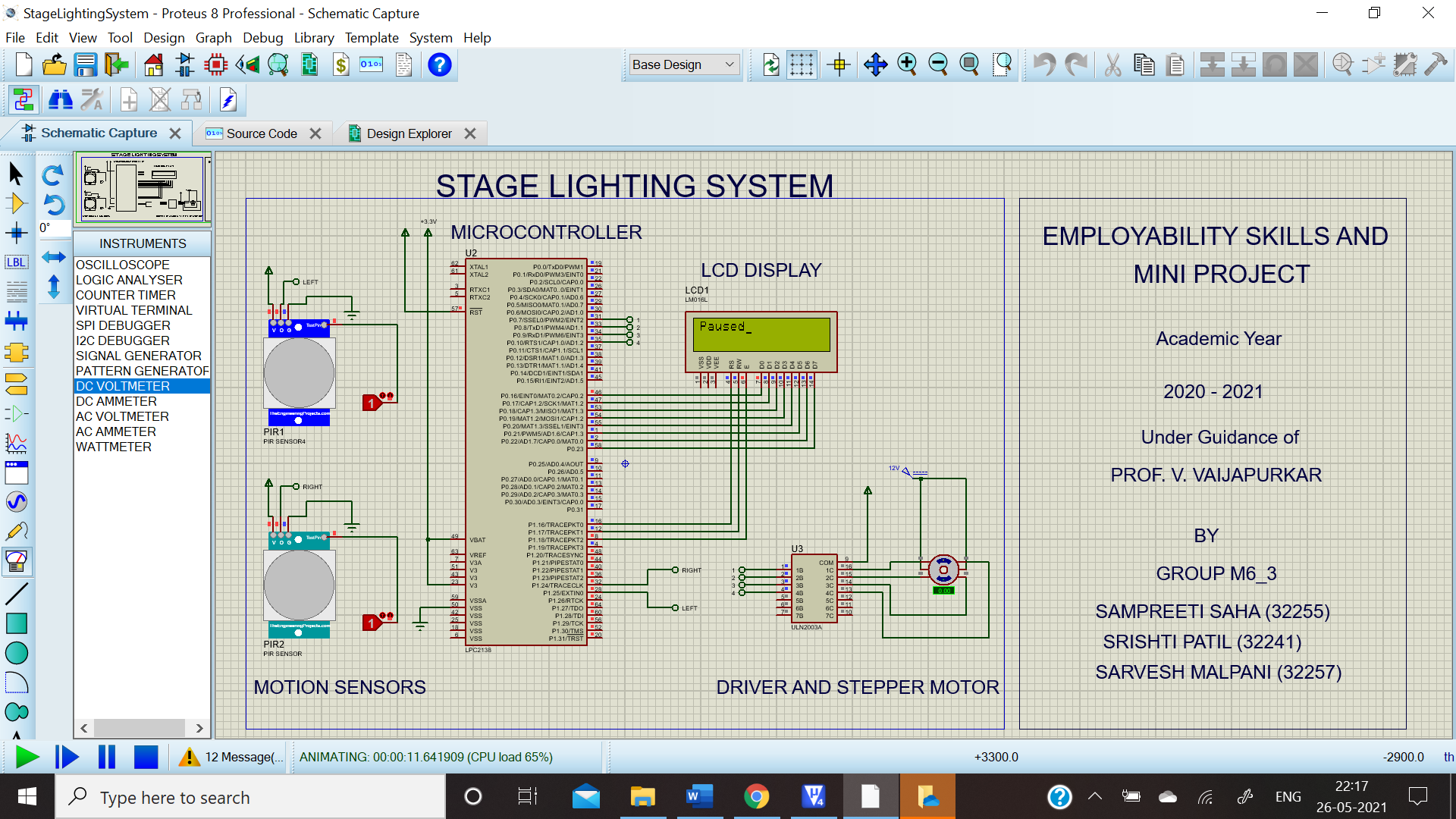


Fig.11. Output of Paused System when both sensors detect motion

CHAPTER 5

Future Scope and Conclusion

5.1 Future Scope

* The color of the spotlight may be changed from the fixed white tone, according to the scene on stage.
* The change of color may be automatic using a camera as a sensor to detect the facial expressions of the main actor.

5.2 Conclusion

Through this project we learnt about the LPC2148 microcontroller and its interfacing with switches, stepper motor and ULN2003 and most importantly about the PIR Sensor which helps in motion detection.

We hope that this project will help improve the experience of actors as well as production houses in the theatre business and be a visual treat for the theatre audience in the future.

References

Books:

[1]ARM Simplified: with LPC2148 by Prof K Subramani

Automated Lighting: The Art and Science of Moving Light in Theatre, Live Performance, Broadcast, and Entertainment, Book by Richard Cadena

Websites:

[2]<https://www.theengineeringprojects.com/2016/01/pir-sensor-library-proteus.html>

[3]<https://embetronicx.com/tutorials/microcontrollers/lpc2148/pir-sensor-interfacing-lpc2148/>

[4]<https://circuitdigest.com/microcontroller-projects/interfacing-stepper-motor-with-arm7-lpc2148>

Datasheets

1. HC-SR 501 [PIR Sensor](https://www.nxp.com/docs/en/data-sheet/LPC2141_42_44_46_48.pdf)
2. [LPC2141\_42\_44\_46\_48](https://www.nxp.com/docs/en/data-sheet/LPC2141_42_44_46_48.pdf)
3. [LED\_RGB\_DISK](https://www.nxp.com/docs/en/data-sheet/LPC2141_42_44_46_48.pdf)
4. 24BYJ48 [Stepper Motor](https://www.nxp.com/docs/en/data-sheet/LPC2141_42_44_46_48.pdf)