Design and simulation of Home Automation

- Bluetooth controlled LED light
- GSM Module controlled Windows Blinds
- LDR or Photoresistor Light.

Izuchukwu George Enekwa
dept. Electronics Engineering
Bluetooth controlled LED
Hochschule Hamm Lippstadt

Giwa Habeeb Riwan
dept. Electronics Engineering
LDR or Photoresistor
Hochschule Hamm Lippstadt

Patrick Nonki
dept. Electronics Engineering
GSM Module Windows Blinds
Hochschule Hamm Lippstadt

Abstract

This paper documentation presents the design and simulation of a bluetooth controlled LED light using a mobile phone application, GSM module controlled Windows Blinds and LDR or Photoresistor Light system. These Home Automated devices are designed to provide homeowners comfort, convenience and energy efficiency by allowing them to control the smart devices with the automated actions. The system was designed and simulated using proteus 8, the hex file was loaded on the Proteus schematic design. The Proteus software was used to design and simulate the main circuit. The Tinkercad was also used to simulate the design for the LDR Photoresistor light. These simulations results were in accordance with the design specification.

I. Introduction

with the help of their mobile phones or automaticToday with the advancement of technology, automation has become part of our lives. Home automation allows us to control household appliances like light, doors, heating windows blinds and so on. A home is usually the most occupied place in any culture, therefore having our home automated will not only reduce human efforts but also energy efficiency and time saving. The system helps both young and old people by controlling home appliances sensors as they do not need to go to different locations for turning ON or OFF. Therefore, a system that works automatically needs to be put in place.

The main purpose of these home automation projects is to reduce the need to use physical contact and implement a system that works automatically by controlling the home appliances using different techniques like android application, Bluetooth, GSM Module and LDR Photoresistor.

II. Bluetooth Controlled LED

This lighting system is designed to be controlled using a mobile phone application where it is connected locally to the Bluetooth in a short range.

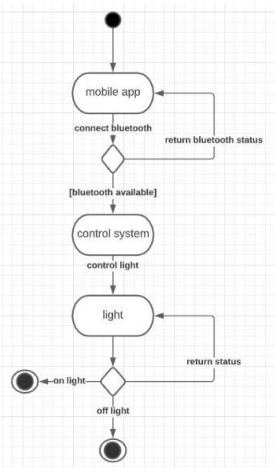
This design was made possible using Arduino. The Arduino is an open source single board-based micro-controller. It is a very popular platform forked from the Wiring platform and firstly designed to popularize the use of electronics in interaction design university students' projects. [1]

With the help of the Arduino input and output pins, I was able to connect several equipment such as the Bluetooth, LED Light and resistor. I used the MIT App inventor to create the mobile application and also used the Arduino IDE to write the Arduino code.

WHAT DOES THE IDE DO?

The IDE provides a graphical interface in which you can write your code, debug it, compile it, and upload it, basically. [1]

III. METHODS (FIGURE 1 State diagram)



The bluetooth controlled LED light comprises two (2) main subsystems: Mobile application system unit and the control/Switching system unit as shown in the system *State machine diagram* in Fig. 1.

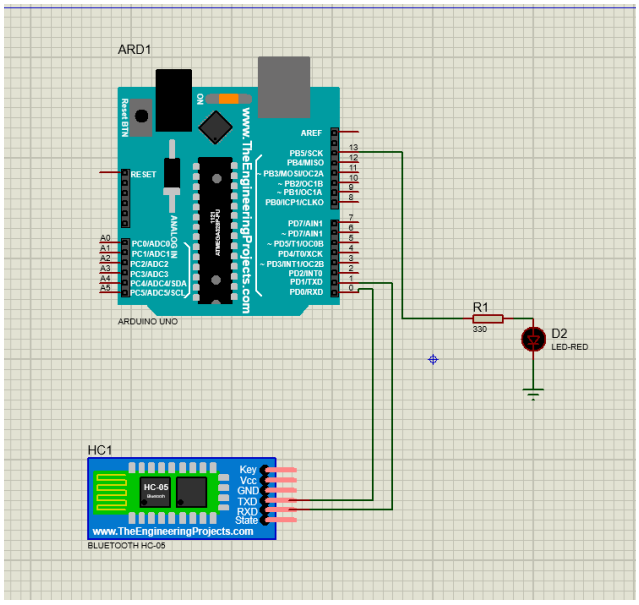
Fig. 1 consists of three blocks, each containing several components. The user defined input consists of the mobile app, where the user connects to the LED light using the mobile app bluetooth and it is also for the switching purpose.The control system unit is the heart of the system that consists of the microcontroller and the HC-05 bluetooth. Fig. 2.

The state machine mapped code can be found on git. [Team-6/SMART HOME/Bluetooth LED Light/State machine C++/State machine C++ at master · Elijah-Endee/Team-6 (github.com)]

List of materials.

Resistor	330 Ohms
LED	Red
Bluetooth	HC-05
Arduino	Uno

Figure 2. Circuit diagram of Automated LED Light control showing all components of the system.



The microcontroller was programmed using C language and the programmed code was compiled using Arduino IDE. The code used can be found below. code 1. The Arduino IDE automatically generates the hexadecimal file (HEX) which was later exported into the proteus file for simulation.

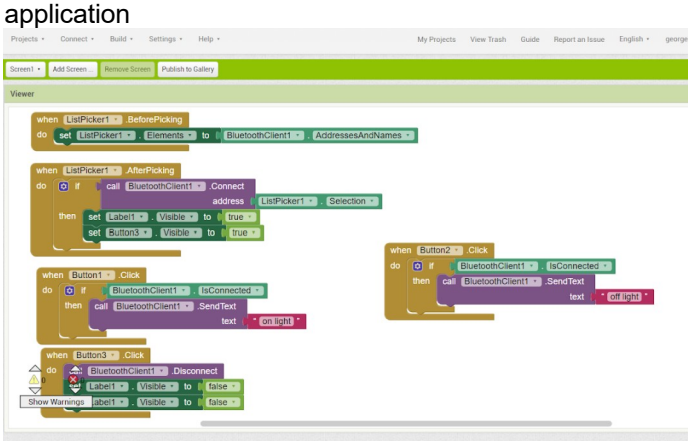
Code 1

```
File Edit Sketch Tools Help
NEWW_Light_Text_Command_C___Light
String text;

void setup() {
pinMode(13, OUTPUT);
Serial.begin(9600);
}

void loop() {
while(Serial.available()){
  delay(10);
  char c = Serial.read();
  text+=c;}

if(text.length() >0){
  Serial.println(text);
```

The Arduino Pinmode used to write the C programming language is the output Pin 13 as seen in fig. 2.

I have conducted testing of the designed system (Bluetooth controlled LED Light) via simulation using Proteus. [Team-6/Arduino code at master · Elijah-Endee/Team-6 (github.com)]

Code. 2. Block code used to generate the Mobile application



The ListPicker1 refers to the connection with the bluetooth server, while the Button1 sends "on light" and the Button2 sends "off light" to the Arduino. The Button3 is the function to disconnect the bluetooth from the connected LED Light.

The Video of the proper simulation of this project can be found here https://youtu.be/0Rs9Riy4K1E

IV. Lighting System Using LDR or Photoresistor

This is the design and implementation of a lighting System using Light Dependent Resistor (LDR). We all want our home appliances to be controlled automatically based on some conditions and that's called Home automation. We are going to control the light based on darkness outside, the light turns ON automatically when it is dark outside and turns off when it gets bright. For this, we need a light sensor to detect the light condition and some circuitry to control the Light sensor. It's like a Dark and light detector circuit but this time we are using Arduino to get more control over light. This design can be used in different areas like Street lights, Public parks, and lights outside of houses.

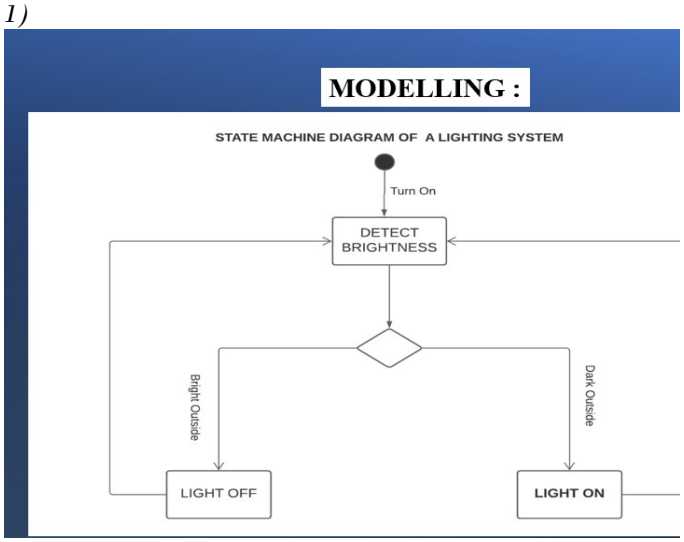
Methods

The method for the implementation of the Light System using LDR is in four (4) stages including design analysis, simulation, construction, and testing. The design is carried out using LDR light sensor and Arduino Uno microcontrollers. The microcontroller is programmed using the Arduino C language and simulated using Tinkercad software.

State Machine Diagram

This diagram shows discrete behavior of a part of the designed system through finite state transitions.

1)

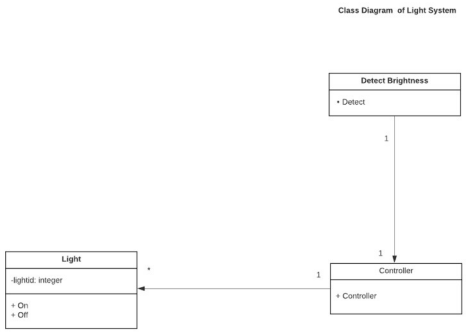


https://github.com/Elijah-Endee/Team-6/blob/master/SMART%20HOME/Control%20light%20with%20LDR/State%20Maschine%20diagram%20%20(2).png

The detect brightness uses LDR to control when the light should be turned on and when the light should be turned off. Whenever outside gets dark due to a fused bulb or any other factors, a light bulb automatically turns on. And when there's a brightness, it causes the light bulb to turn off automatically.

Class Diagram

Classes in a class diagram correspond with classes in the source code. The diagram shows the names and attributes of the classes, connections between the classes, and sometimes also the methods of the classes.

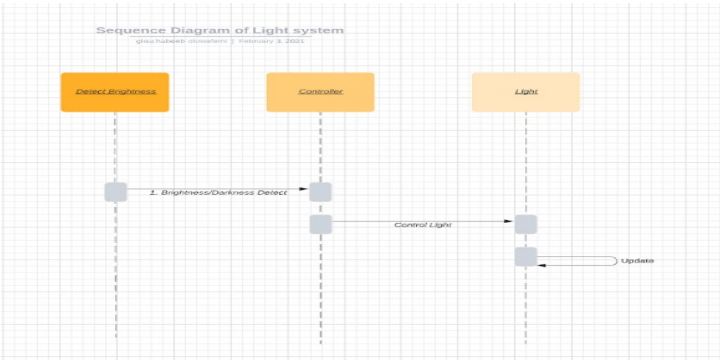


https://github.com/Elijah-Endee/Team-6/blob/master/SMART%20HOME/Control%20light%20with%20LDR/UML%20class.png

This Detect brightness is controlled using LDR sensors and an array of LED's which are interfaced with the microcontroller. According to the value of LDR sensors the brightness is controlled . At the daytime, when there is a need of less light, LDR sensors sense the light and accordingly the programmed LED's glows, but when there is a need of more light LDR sensors sense the light in nearby area and based on the amount of light required the LED's are turned on.

Sequence Diagram

This diagram shows the interactions between objects in the sequential order that those interactions occur.



IDE Code from State machine diagram

A state machine consists of a number of states and the transition between states. To create a state machine, you start by creating a controller class, and then create a sub-state machine diagram from the controller class. Moreover, you can generate source code based on the sub-state machine diagram.

Also we had to write test codes for each element to ensure that the expected values are returned.

An overview of the codes can be seen at:

https://github.com/Elijah-Endee/Team-6/blob/master/SMART%20HOME/Control%20light%20with%20LDR/lightSystemMain2%20(1).c

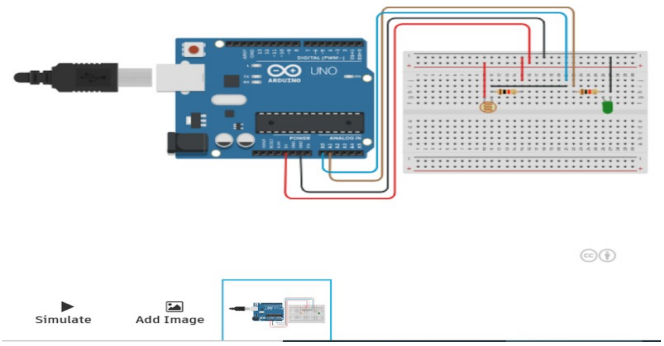
https://github.com/Elijah-Endee/Team-6/blob/master/SMART%20HOME/Control%20light%20with%20LDR/LightSystemTestCode.c

HardWare Components Required:

- LDR
- LED
- 100k Resistor
- Bread Board
- Connecting wires
- Arduino

Arduino LDR Sensor Connections

First, you need to connect the LDR to the analog input pin 0 on the Arduino. You have to use a voltage divider configuration to do this. The connection diagram for the Arduino is as given below.



https://github.com/Elijah-Endee/Team-6/commit/744e3d13e96ca658fe38a58c94ef51549911f953

One leg of the LDR is connected to VCC (5V) on the Arduino, and the other to the analog pin 0 on the Arduino. A 100K resistor is also connected to the same leg and grounded.

2)

3)

4)

*5) **Testing the Code for the Arduino LDR Sensor***

After connecting the LDR to your Arduino, you can check for the values coming from the LDR via the Arduino. To do this, connect the Arduino via USB to your PC and open up the Arduino IDE or software. Next, paste this code and upload it to your Arduino:

TINKERCAD CODE SNIPPET

```
int brightness;
void setup()
{
  pinMode(A1, OUTPUT);
  pinMode(A0, INPUT);
}

void loop()

{

  brightness= analogRead(A0);

  if(brightness <350)
  {
    digitalWrite(A1, HIGH);
  }
  else
  {
     digitalWrite(A1, LOW);
  }
```

IDE CODE SNIPPET

https://github.com/Elijah-Endee/Team-6/blob/master/SMART%20HOME/Control%20light%20with%20LDR/Arduino%20code.txt

Simulation Method

Programming of the microcontroller was done using the Arduino software while the circuit design was done using Tinkercad.

An overview of the simulation can be found at:

https://youtu.be/7_blEmUca9E

V. Windows blinds controlled via GSM

To be able to control the windows blinds, we had to use two different possibilities :

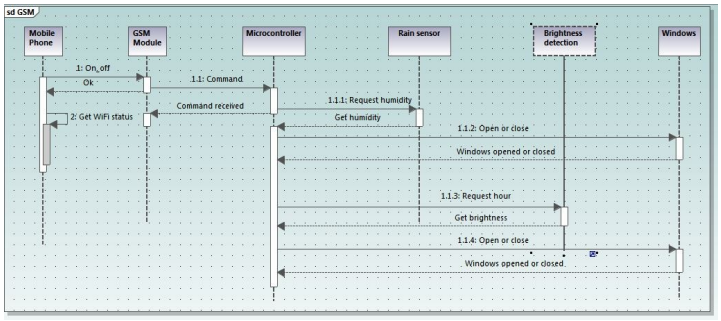
- Either by choosing to control it via a GSM Module by sending a ping to it by phone
- Either automatically, by detecting the rain outside and also the brightness by detecting if it's night or day.

The main steps that we used to develop our system is the model-based analysis where all the specifications are developed step by step by first developing the UML diagrams, move it to C++ and make tests, and then implement it on our arduino and then making simulation using Proteus professional.

UML Diagrams

Sequence diagram

This UML diagram shows the different interactions between all the components of our system and is seen as the user interface.



Here we can see that the user sends instructions to open or close the windows via GSM Module and after it there are 2 actions happening at the same time : get the humidity and the brightness by the Microcontroller and then make the required action.

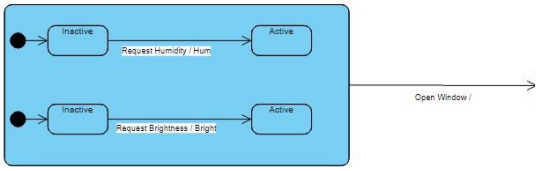
State machine diagram

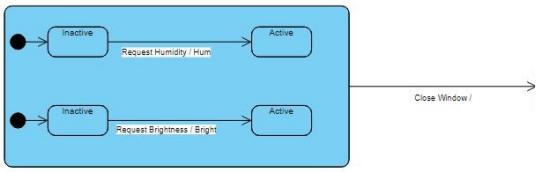
This diagram is designed to show the main behaviour of all the elements of our system depending on the inputs and operations required.

- State machine diagram of the mobile phone

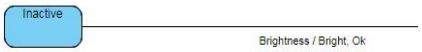
- State machine diagram of GSM Module

- State machine diagram of Controller





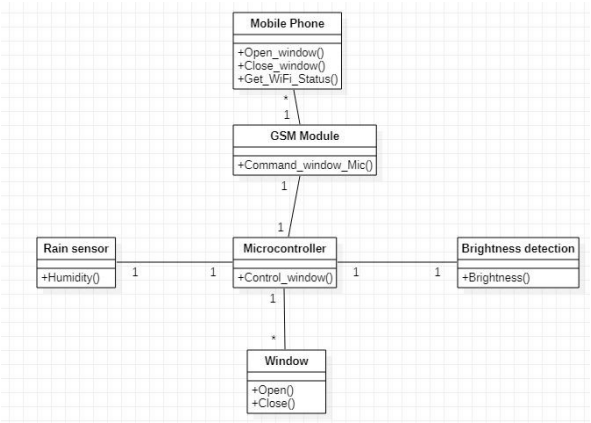
- State machine diagram of Brightness detection



- State machine diagram of Rain sensor

Class diagram

These diagrams are designed to see the interaction between all the elements considered as classes, the operations within each of them and the relationships.



In the phases of the development of our project, the incremental delivery for validation was taken in account and thus after validation of the diagrams, we moved to the C++ mapping of Class diagrams and state machine diagrams and tests.

C++ mapping from Class diagram

Here we have to map the code depending on the UML Class diagram that we designed earlier. To do so, we had to map each class independently with the source file and the header and also the functions contained in each class.

An overview of the code can be seen at:

https://github.com/Elijah-Endee/Team-6/tree/master/SMART%20HOME/C%2B%2B%20for%20Windows%20blinds/C%2B%2B%20mapping%20from%20Class%20diagram

C++ mapping from State machine diagram and tests

Here also we had to map from the UML state machine diagram but in addition, we had to consider the different behaviour of the system by switching between the states of all the entities.

Also we had to write test codes for each element to ensure that the expected values are returned.

An overview of the codes can be seen at:

https://github.com/Elijah-Endee/Team-6/tree/master/SMART%20HOME/C%2B%2B%20for%20Windows%20blinds/C%2B%2B%20mapping%20from%20State%20machine%20diagram

By following our working method, after validation of these steps we moved forward to the next ones.

After writing these codes we had to make it real by choosing the best way to implement it by choosing the best components:

- Arduino as microcontroller because of its range of possibilities and his powerful components

- Rain and Brightness sensors to detect the rain and the brightness outside
- GSM Module to control manually our system by sending ping requests.
- Proteus professional for the simulation of our system.

Arduino Ide and code

The code written takes in account all the inputs and outputs of our system and also the libraries and is valid just for the simulation part

- As inputs we used the pins as follow:
  - Digital pin 6 for the Rain sensor
  - Digital pin 8 for the brightness detection
  - Digital pin 7 for the GSM open instruction
  - Digital pin 12 for the GSM close instruction
  - As outputs we used the pins as follow:
  - Digital pin 4 for the motor to open the window
  - Digital pin 13 for the motor to close the window

The overall arduino code can be found at :

https://github.com/Elijah-Endee/Team-6/tree/master/SMART%20HOME/C%2B%2B%20for%20Windows%20blinds

Proteus professional and Simulation

Among all the simulation tools available, we decided to use Proteus Professional due to his large pallet of components and also real time simulation capability.

For the simulation we had to drag each component on the working area and upload the arduino code in the simulated arduino equipment by adding the path of the code file as parameters. An overview of the simulation can be found at:

https://youtu.be/cz2yGreuNTE

Conclusion

By doing this project we are able to see the working of Automated Bluetooth LED Light, LDR light and an automated windows blinds. This project involved designing and development of these home automation projects to reduce the need to use physical contact and implement a system that works automatically by controlling the home appliances using different techniques. We were motivated to do this project because, home automation has improved the way we humans live and will also do so in years to come, we want to be a part of the generation to make this huge impact in the most occupied spaces in human lives.

References

[1] Julien Bayle, C Programming for Arduino, First published: May 2013 Production Reference: 1070513, Published by Packt Publishing Ltd.

https://www.ijariit.com/manuscripts/v4i5/V4I5-1321.pdf

https://www.researchgate.net/publication/348560011_Design_and_construction_of_an_automatic_streetlight_controller_using_microcontroller_and_LDR [accessed Feb 03 2021].

https://ieeexplore.ieee.org/document/6014058/authors#authors

https://java-programming.mooc.fi/part-11/1-class-diagrams

https://create.arduino.cc/projecthub/SURYATEJA/automatic-street-light-controller-27159f

https://www.researchgate.net/publication/325722320_Designing_and_Modeling_of_Arduino_Based_Light_Sensor

https://www.krishisanskriti.org/vol_image/03Jul20150307485.pdf

https://www.researchgate.net/publication/348560011_Design_and_construction_of_an_automatic_streetlight_controller_using_microcontroller_and_LDR

AFFIDAVIT

We (Names; ENEKWA IZUCHUKWU GEORGE, PATRICK NONKI and GIWA RIWAN HABEEB) herewith declare that we have composed the present paper and work ourself and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form has not been submitted to any examination body and has not been published. This paper was not yet, even in part, used in another examination or as a course performance.