CS-308-2015 Final Report

Home Automation

Sahil Jindal 110020043

Rohan Gyani 110040001

Abhishek Gupta 110040067

Mridul Ravi Jain 110040083

Table of Contents

1. Introduction

One of the primary concerns regarding the modern day electrical appliances is their heavy consumption of electricity. Majority of electricity consumption in our homes and office space is done by lights, fans etc. While we need to use these appliances for daily activities, unnecessary usage of them in unoccupied spaces is what we are more concerned about. Switching off lights and appliances is one of the best known ways to save electricity. Many a times we forget to switch them off when not needed. Most of the times this happens for convenience or because of forgetfulness. But all this can be avoided using some form of automated sensing. The light or any other electrical appliance can be activated automatically by the active presence of a human body within the room & when there is no presence the lights can be deactivated automatically.

2. Problem Statement

Our objective is to control electrical appliances (lights, fans, etc.) of a room based on whether the room is occupied or not. The room occupancy is determined using smart door and PIR sensors.

If room is occupied, then switch on PIR sensors in the room to detect which part of room is occupied and switch on the appliances only in that part of the room. If room is unoccupied for some time switch off the appliances and if room remains unoccupied for some more time switch off the PIR sensors as well.

3. Requirements

3.1 Functional Requirements

- The room entrance should have a smart door that can detect whenever person enters or exits the room and sends this data to our circuit
- The PIR sensor is moved with help of a servomotor so that it can cover a relatively larger part of room
- The power mains of the rooms are controlled via relays connected to our circuit
- PIR sensors and servomotors are switched on when room occupancy becomes non-zero
- PIR sensors and servomotors are switched off after some time when the room occupancy becomes zero
- The lights are switched on when PIR sensors detect some motion
- The lights are switched off when PIR sensors detect no motion for some time

3.2 Non-Functional Requirements

- There is only a single entrance to the room
- PIR sensors should be accurately positioned so as to cover the entire room
- Room entries and exits should be accurately determined by the smart door
- Only human motion must be detected by the PIR sensors
- Make sure that PIR sensors and servomotors are operated in their respective voltage ratings

3.3 Hardware Requirements

- Simple smart door (implemented in SEIL)
- Raspberry Pi
- Arduino
- Servomotor
- Passive Infrared Sensor
- 4-Relay board
- 5-V DC source
- Bread board
- Jumper wires and LEDs

3.4 Software Requirements

- Arduino IDE
- PuTTY
- Xming display server

4. System Design

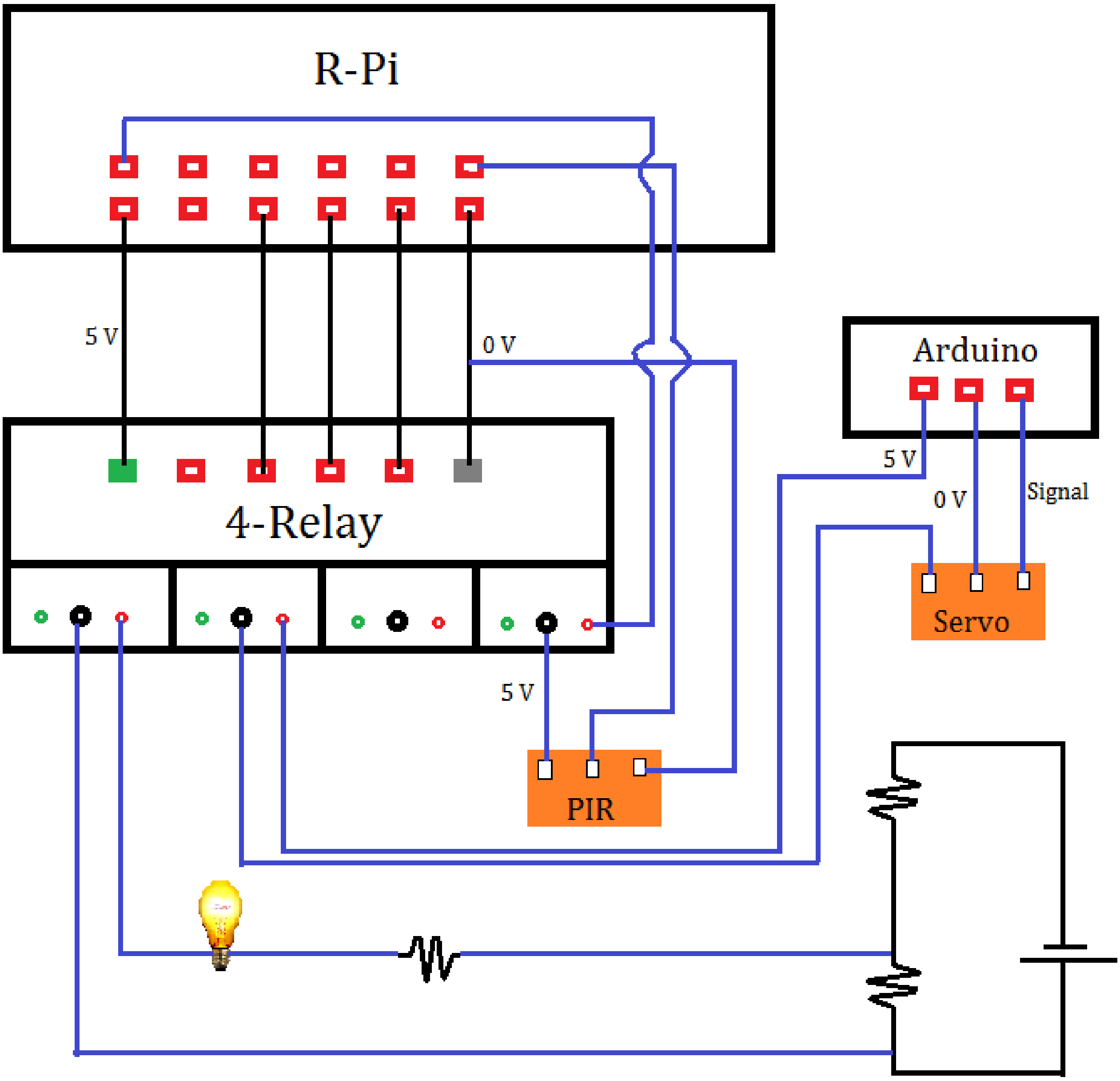
Smart door functionality:
Smart door is placed on the only entrance of the room and detects whenever someone enters or exits from the room. It uses an R-Pi and combination of laser sensor to do this. When a person enters or exits through the smart door, he cuts through the lasers and the signals is sent to pins 11 and 16 on R-Pi. We have used the already implemented smart door in SIEL.

Sending signals from Smart door to R-Pi:
Smart door implements a whole lot of functionalities that we do not need. We needed to detect only the entry and exits. Therefore we used a different R-Pi to implement our system and send signals from the smart door R-Pi to our R-Pi whenever entry or exits happen. This required using socket programming. Our R-Pi runs a server that waits for connection from the smart door R-Pi, and expects integers ‘1’ and ‘2’ from it. The smart door R-Pi acts as a client that sends ‘1’ when an entry happens and ‘2’ during an exit. Our R-Pi receives this data and processes it accordingly.

Working of PIR
PIR sensors are able to detect a change in IR levels of their detection zone (e.g. when a human enters a room) and hence sense motion. The PIR sensors we used have three pins: ground, digital out and 3-5VDC in. At idle, when no motion has been detected, the digital out will remain low, however when motion is detected, the digital out will pulse high (3.3V) and we use our R-Pi to sense this. The PIR sensors we use have a range of approximately 7 meters, and a 110° x 70° detection range, so it’s great for monitoring a door or the corner of a room.

Working of Servomotor
In order to minimize the number of these sensors we mounted each PIR sensor on a servomotor. This enhances the range of sensing and reduces the number of PIR sensors. The servomotor we used has three pins: ground, digital-in and 4.8-6 V DC in. The digital-in receives pulses to that

Circuit diagram of implementation:



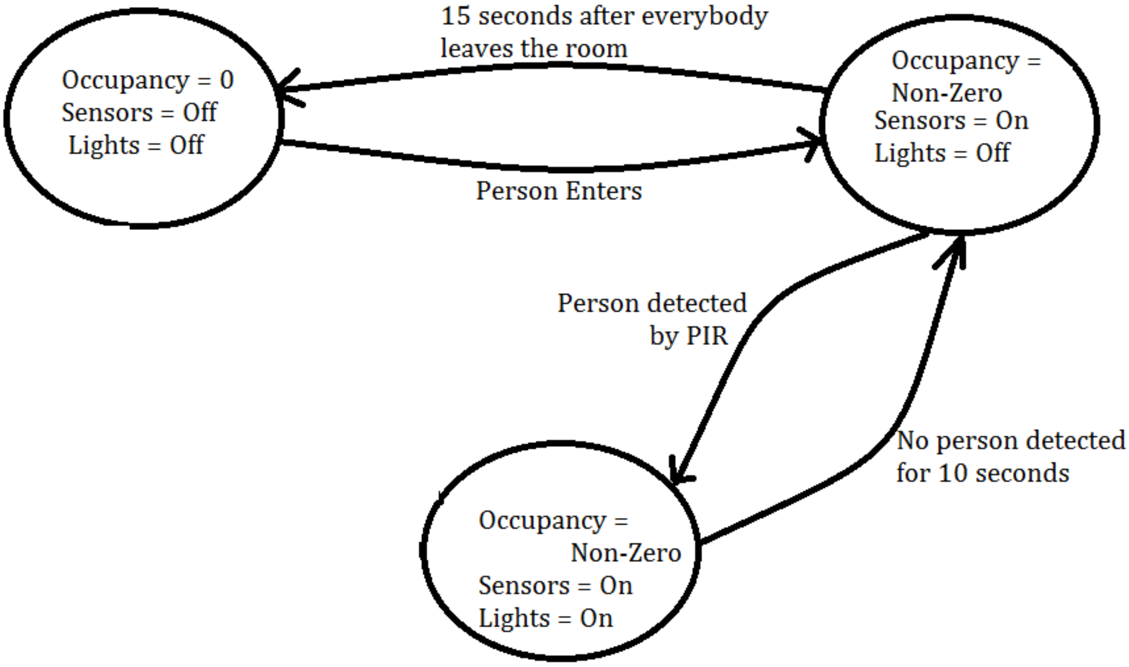
control the angle that servomotor sweeps. Due to the unavailability of pulse width modulation (PWM) interface in R-Pi there were some problems in implementation of servomotor using R-Pi. Hence we used Arduino to send the pulses to the servomotor. The servomotor is set and PIR sensor is mounted on it. Sensor is moved with the help of servomotor.

Working of Relays

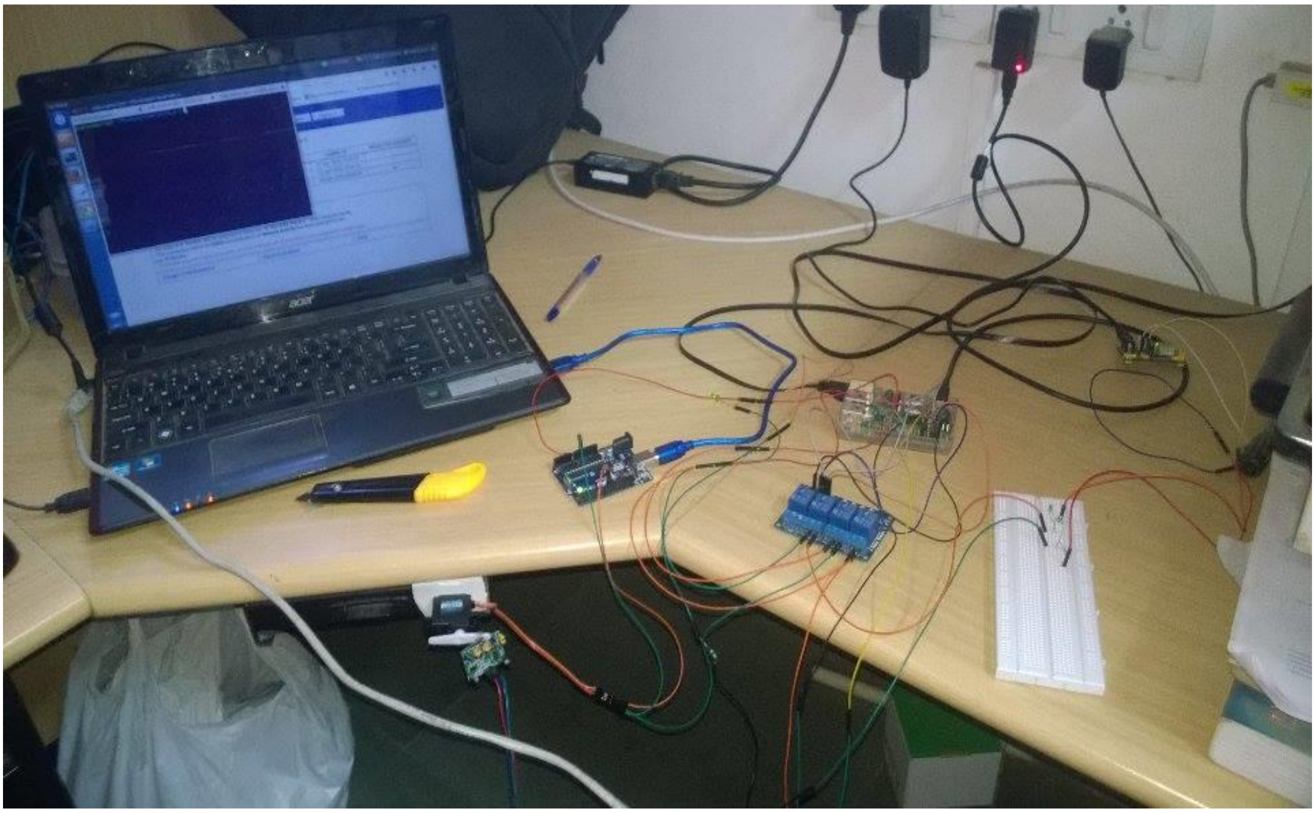
We used a 4-Relay board to control the power to PIR sensor, servomotor and the power mains of the light we control. The signal to switch on/off these devices is sent from the R-Pi to relay board. Each relay consists of three screw type terminals labelled: NO, COM, NC. The COM is connected to NC when relay if off and COM is connected to NO when relay is on.

5. Working of the System and Test results

Finite State Machine of our implementation:



Snapshot of the implementation:



6. Discussion of System

a) What all components of our project worked as per plan?

We successfully implemented the proposed design for both part (a) [Grade-B] and part (b) [Grade-A]

In **part (a)** of the project, we controlled the electrical appliances in the room directly using the input from the smart door. This part did not use the PIR sensors and servomotors. The smart door notifies whenever someone exits or enters the room and using this data we determine the room occupancy. When room occupancy becomes zero then all the lights in the room are switched off (after 10 second) and when room occupancy becomes non-zero then all lights in the room are switched on.

In **part (b),** we extended the above implementation to switch on lights only in the relevant parts of the room. To identify these relevant parts we used PIR sensors. The PIR sensors that we are using have a range of 110 degrees x 70 degrees up to around 20 feet (6 meters). In order to minimize the number of these sensors we mounted each PIR sensor on a servomotor. This enhanced the range of sensing and reduced the number of PIR sensors. Now the room occupancy (number of people in the room) is still determined using smart door but it does not control the lights directly. Rather,
PIR sensors and servomotors are

- switched on when room occupancy becomes non-zero
- switched off after 10 seconds when the room occupancy becomes zero

And, the lights are

- switched on when PIR sensors detect presence
- switched off when PIR sensors detect no motion for 10 time

b) What we added more than discussed in SRS?

In the SRS we planned to implement our system on the R-Pi used by the smart door. However since we did not need all the smart door functionalities and needed to detect only the entry and exits, we used a different R-Pi to implement our system and send signals from the smart door R-Pi to our R-Pi whenever entry or exits happen. This required using socket programming. Our R-Pi runs a server that waits for connection from the smart door R-Pi, and expects integers ‘1’ and ‘2’ from it. The smart door R-Pi acts as a client that sends ‘1’ when an entry happens and ‘2’ during an exit. Our R-Pi receives this data and processes it accordingly.

c) Changes made in plan from SRS and challenges:

The position of the servo motor is set by the length of a pulse. The servo expects to receive a pulse at least every 20 milliseconds. The angle is converted into a duty cycle between 0 and 100. This actually produces pulses shorter than the 1 millisecond expected minimum value and longer than 2 milliseconds maximum.

Due to the unavailability of pulse width modulation (PWM) interface in Raspberry Pi there were some problems in implementation of servomotor using R-Pi. We tried to manually generate pulses of various duty cycles using GPIO library, but it resulted in jitter with the servo.

While trying to solve this problem we found that the servomotor works fine on turning on the power source but doesn’t work smoothly afterwards. In order to solve this problem we used different power sources as well but it resulted in same problem.

Finally we found that since R-Pi just like every other operating system runs multiple processes at any point of time. Python (like everything else) is being interrupted by the operating system to let other programs run. They causes jitter in the pulse when our program is interrupted between the start and stop pulse lines.

Hence we used Arduino to send the pulses to the servomotor. The servomotor is set and PIR sensor is hinged about a point. Sensor is moved with the help of servomotor. If we want to avoid the use of Arduino we can use R-Pi with software solutions (such as servoblaster, piblaster, pipwm) which use DMA to time the pulses. DMA is not affected by operating system interrupts.

7. Future Work

Our project has scope for improvement in terms of scalability. The system can be extended for a multi-entrance room. Further, it can be extended to multiple rooms and corridors with the lights in the corridors being switched on during the exits.

Due to limitations of resources, we have used a single PIR sensor mounted on a servomotor to switch on a single light. For practical purposes, this can be further extended to multiple sensors controlling multiple lights in a room.

The complete system can be integrated with other systems that may include centralized control of lighting, air conditioning, security systems to provide improved convenience, comfort, energy efficiency and security.

8. Conclusions

Automatically turning off lights in unoccupied spaces such as classrooms, conference rooms, public spaces, dormitories, large offices can result in huge energy savings in the long run. Proper placement and orientation of system is essential.

Our primary objective was to implement the idea of power saving using the available technologies at the minimum cost. For practical purposes it needs scaling up using multiple components.

9. References

1. http://arduino-info.wikispaces.com/ArduinoPower#4-8
2. http://www.raspberrypi.org/documentation/usage/gpio/
3. http://www.raspberrypi.org/learning/quick-reaction-game/worksheet/
4. http://www.raspberrypi.org/forums/viewtopic.php?t=36225
5. https://www.youtube.com/watch?v=ddlDgUymbxc
6. https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=46771
7. http://raspi.tv/2013/how-to-make-your-own-raspberry-pi-flag-waving-demo
8. https://learn.adafruit.com/adafruits-raspberry-pi-lesson-8-using-a-servo-motor/software
9. http://www.modmypi.com/blog/raspberry-pi-gpio-sensing-motion-detection