Automation of window blinds using Raspberry Pi

Akshat Goel
Computer Engineering Department
San Jose State University
San Jose, US
akshat.goel@sjsu.edu

Apurav Gupta
Computer Engineering Department
San Jose State University
San Jose, US
apurav.gupta@sjsu.edu

Hemang Behl
Computer Engineering Department
San Jose State University
San Jose, US
hemang.behl@sjsu.edu

Varun Shrivastav
Computer Engineering Department
San Jose State University
San Jose, US
varun.shrivastav@sjsu.edu

***Abstract*—We have built an application which smartly automates the window blinds with the help of light sensors. We can control the blinds with the application and control it using voice. This report will explain what technologies and devices were used and how they were implemented.**

Keywords—automation, blinds, raspberry pi, light sensor, web application

I. INTRODUCTION

In the current scenario, various products exist for automating the window blinds but each one of them suffers from the same issue being the high cost. Being unaffordable causes many people and places like hospitals loose out on availing the benefits of automated window blinds.

For this purpose, we have created a low-cost and a fully automated window blinds platform. We have also implemented light and motion sensing capabilities which helps the blinds in being truly called 'smart' as it opens, and closes based the sensors input.

II. SOFTWARE USED

A. *Raspbian OS*

Raspbian is Debian based Linux distribution. It promotes Python as the main programming language, The default firmware is closed source, while an unofficial open source is available.

B. *Flask Server*

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.

III. HARDWARE USED

We used Raspberry Pi 2.0, resistors of varying resistance, capacitor, LED bulb, light dependent resistor (LDR) sensor, IR sensor (motion sensor), breadboard, jumper wires, window blinds and a RC 180 degrees servo motor.

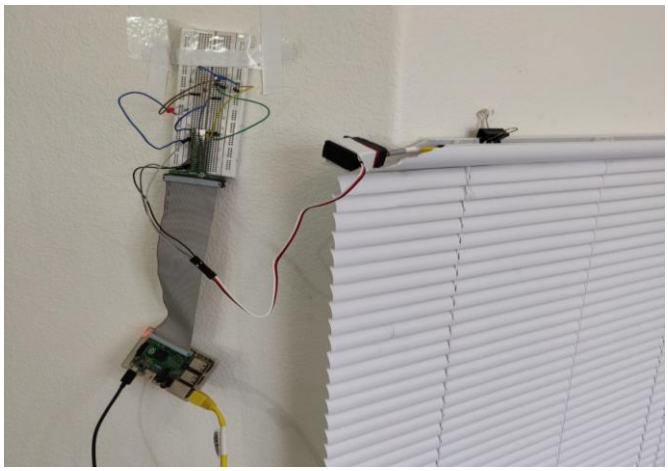


Fig 1: Setup of hardware

IV. APPROACH

We started with Arduino as our base platform on which we were going to build the application. However, we switched to Raspberry Pi as it was more suited to our needs which required it to be flexible enough to be connected to various devices. We connected the sensors on the breadboard which was in turn connected to the Raspberry Pi.

We installed and used Flask server in Raspbian OS for making REST-API calls from the front end being the web application. We got the added advantage of Python being the framework's language. We wrote the python code in the flask server to handle the REST API calls and make the Raspberry control the servo motor as required. We were able to completely open and close the window blinds and control the angle of the blinds. We set a threshold for the light sensor so that when a pre-defined level was reached, the blinds would automatically turn by a specific angle. The motion sensor was added as an extra feature to also close and open the blinds.

V. ARCHITECTURE

Below given is the architecture diagram for our breadboard connectivity:

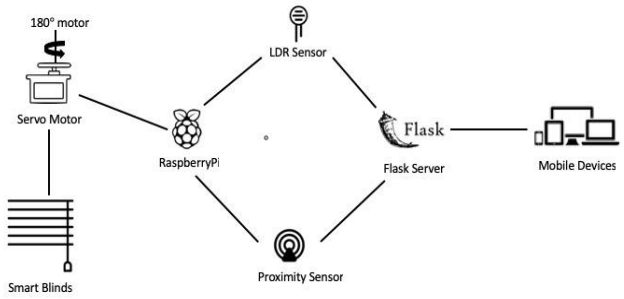
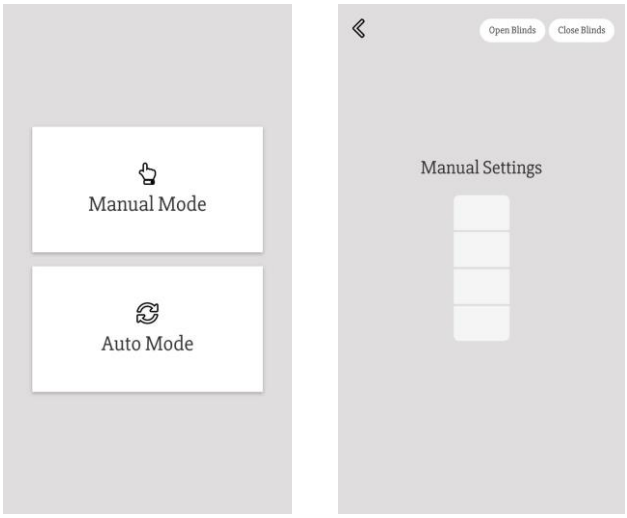


Fig 2: Architecture diagram

VI. Features

A. Multi-platform app

The below figures show the user interface of the web application which communicates with the Raspberry Pi using REST API calls.



(a) (b)

Fig 3: (a) Web app showing the two modes
(b) Web app showing the manual mode

B. Voice control enabled by Siri Kit iOS

Voice control was achieved by using Siri Kit in iPhone. By simply saying 'Hey Siri, open the window blinds' and 'Hey Siri, close the window blinds' we were able to open and close the window blinds respectively. Voice controls uses REST API to send commands to open and close.

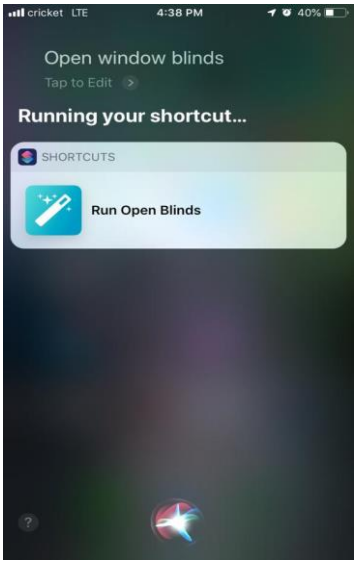


Fig 4: Screenshot showing Siri-Kit in action

C. Light sensing

Using a light-dependent resistor we were able to detect light and read that as numerical values in the flask server installed on the Raspbian OS. Four levels were pre-defined based on the light received. The blinds are adjusted to these four levels based on the light received. The 'auto' mode uses this feature.

D. Motion sensing

Using an infra-red sensor, motion was detected within a short distance. The windows blinds were toggled between open and close whenever any motion was detected.

VII. Future Work

We plan to integrate temperature and humidity sensor as well to the application. We also plan to integrate voice control with Google Home Mini and Amazon Alexa for a complete home automation experience.

Acknowledgment

Akshat Goel, Apurav Gupta, Hemang Behl and Varun Shrivastav thank Professor Rakesh Ranjan for his guidance and support given to us to complete this project.

The GitHub link for the project is as follows:

https://github.com/SJSU272LabF18/Project-Team-16

References

[1] https://en.wikipedia.org/wiki/Raspberry_Pi#Software
[2] https://en.wikipedia.org/wiki/Servomotor