Final Project Report

WCWindow

Table of contents

1. Introduction
2. Functional Specification
3. Design
4. Implementation and Testing
5. User Guide/ Instructions

Introduction

A Window that does no require manual handling but instead can be opened and closed using your and app on your smartphone , tablet etc as long as there is a internet connection.

Functional Specification

The Smart Window is a window that is controlled by a smartphone. It can be controlled by commands sent from your phone or device and does not need manual handling. It uses a internet connection to allow the commands to be sent back and forth from window to device. It also can be programmed to be closed or opened at certain intervals.

I/O

Inputs:

WindowOpenButton(digital)

WindowCloseButton(digital)

Outputs:

ClosingWindow(analogue)
OpeningWindow(analogue)

Non-Functional requirements:

The Smart window is both electric and battery powered at all times. The Dc motor is battery powered whereas the Raspberry Pi is electrically powered. The smart window requires an internet connection between the smart-phone device, server and the smart window.

Design

WCWindow consists of :

- D.C motor
- Pinion gear and rack
- Card-board box – Frame
- 2x 1.5V baterries
- Plastic Window Pane
- Electrical circuit
- Raspberry Pi
- Android Studio App.

Implementation / Testing

Specification:

WCWindow uses D.C motor, Electrical circuit , Raspberry Pi and Android Studio App.

WCWindow code was written using the programming language python in the linux terminal of the Rasberry Pi.

WCWindow ran by calling a command on the linux terminal to run the python script.

Problems:

WCWindow was tested using Android studio app but it was not responding to command sent from the app. Problem may be within setting up of the server or setting up of the app.

UserGuide/Instructions

1. Set up electrical circuit
2. Set up Rasberry Pi
3. Connect Battery power source
4. Open up linux Teminal
5. Run code on specified WCW directory

Appendices

1. *Commented Code*
2. *Powerpoint Presentation*

Commented Code:

```
Gpio.setup(7,GPIO.OUT)
Gpio.setup(11,GPIO.OUT)
//pulse width modulated pins


p=GPIO.PWM(7,207)
q=GPIO.PWM(11,50)

p.start(0)
q.start(0)

try:
    while True:
                for I in range(100):
                        p.changeDutycycle(i)
                        time.sleep(0.02)
                 / /increase the motor
                 for I in range(100):
                        p.changeDutycycle(100-i)
                        time.sleep(0.02)
                  //drecrease speed of motor and continue cause its in a while loop **for demonstration purposes

                   p.changeChangeDutyCycle(0)
for I in range(100):
```

q.changeDutycycle(i)
time.sleep(0.02)
/ /increase the motor opposite direction
for I in range(100):
q.changeDutycycle(100-i)
time.sleep(0.02)
//drecrease speed of motor and continue cause its in a while loop **for demonstration purposes in opposite direction

q.changeChangeDutyCycle
except KeyBoardInterrupt:
pass
p.stop()
Gpio.cleanup()

Power point->