Department of Computer Science

CPS-342 Embedded Linux

Final Project

Camera Boom

Date:05/12/2016	Spring Semester 2017

Group Members	Department	Major Contribution
Joseph Abel	CE	Programmed Servos
Nathan Dalling	CS	Programmed Flask web server

Lecture Instructor: Professor Chirakkal Easwaran

Abstract

The objective of this project is to be able to take photos in any direction and control the positioning through a web interface. In order to do this we have two servos which control the raspberry pi camera and allow it to move by inputting in coordinates into the web interface. Then once the pictures are taken they get stored in a web interface.

Table Of Contents

I. Introduction

The Camera Boom project was a project designed in the computer science course embedded Linux. In this project we developed a Graphical User Interface (GUI) using a flask web server which we programmed in python. This GUI that was developed will be able to remotely send over coordinates to the servos. This will result in the servos going in a certain location and allow the raspberry pi camera to take a picture. Our project that we developed is a functioning Camera Boom which uses 2 continuous servos to move the raspberry pi camera. These servos communicate through a Flask web server. The raspberry pi camera then sends images to our gallery on our Flask Web server.

I.I Project Goals

The goals in this project was to successfully create a flask web server which would provide us with our GUI in order to communicate between the GUI and the raspberry pi camera as well as the servos. The goals for the servo are the following: (1) create a reference point for the servos location, (2) design a mechanism in which the 2 servos can rotate the raspberry pi camera, (3) how we can run both servos simultaneously, and (4) get an accurate desired location for the servos. The goal for our raspberry pi camera is:(1) take a picture and (2) send it up to our flask web server. The main goal of the website is for it be user friendly, allow for the pictures to be stored in an organized fashion and display the servos location.

II Implementation

Camera boom was broken up into four parts the raspberry pi web server, programming of the servos, the creating of the stand and the programming of the camera

II.I Flask Web Server

Flask is a web application which provides frameworks to users. Flask web servers are developed using the python programming language. The Flasks web server has a specific functionality which is: communicate to the servos by being able to input in a certain number of degrees for the servo to go to, store the pictures taken from the raspberry pi camera in a gallery and provide a user friendly approach to web interfacing. In figure 1 below our Flask web server is being displayed. As you can see there is a (“x =”, “y =”) this will return and display the current position of the servo. The reset button below will reset the servos to our original position which is a preprogrammed the position. Next we have preview this button allows you to capture a photo and puts it in the gallery folder as preview.jpg. Then next the blank text box allows you to enter a picture image title. This picture image title will then be saved to the Gallery folder once you click capture. This is displayed on our flask web server. Next the button set set’s the location of the servos and the up, down, left, right buttons allow the camera and the servos to move in that direction. Under this it calls the web server calls all the images and the names of images in the gallery folder and displays it.

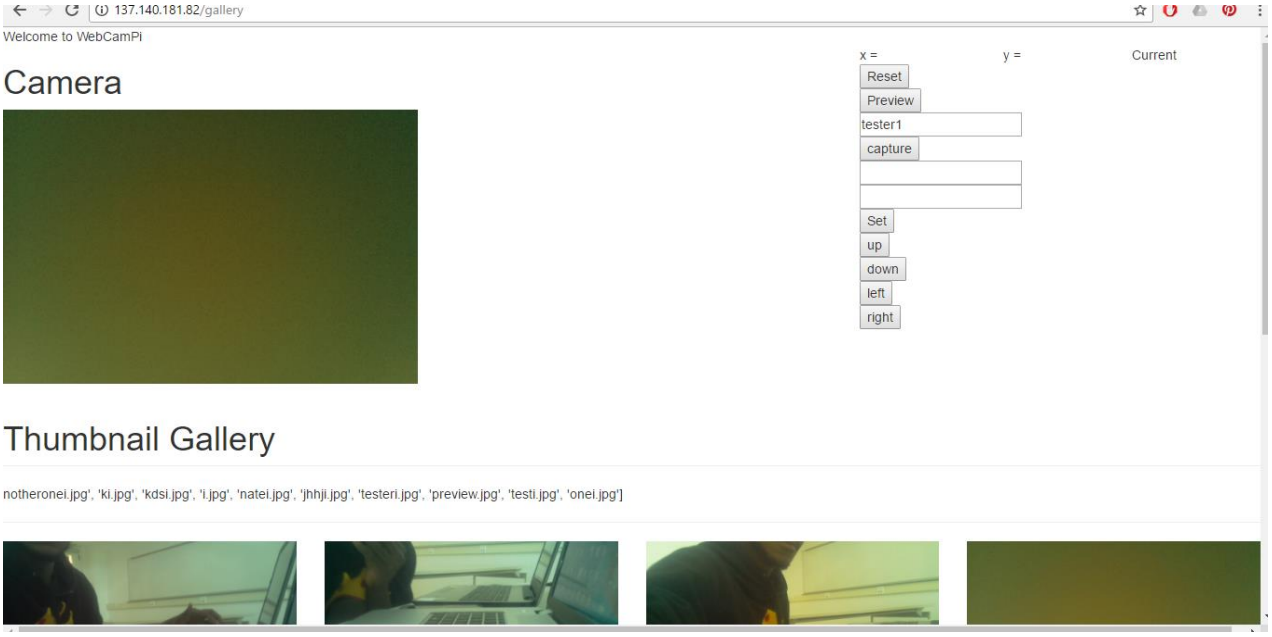


Figure 1: Flask web server home page

II.II Camera

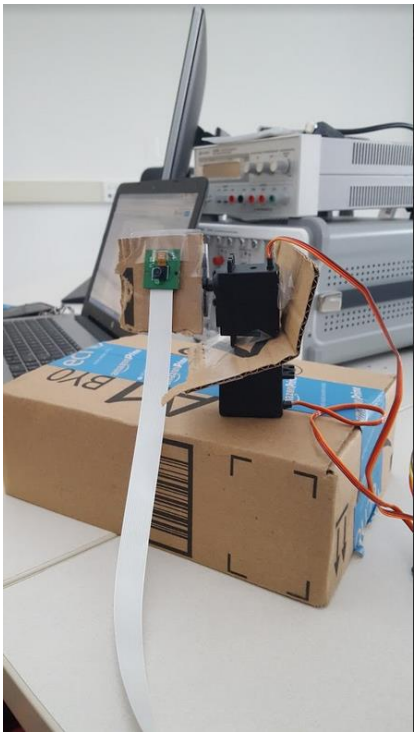
The raspberry pi camera uses very basic properties. When the camera is on the light will turn on and the raspberry pi will take a picture and then after it takes a picture it will proceed to store into a folder for the gallery.

II.III Servos

The servo motors need a minimum of 5V to keep it continuously spinning 360 degrees. These continuous servos that we used allowed for full rotation of the camera. These servos also allowed the camera to be accessed in all three dimensions. In this project we needed to spin 2 servos but this turned out to be a problem since we only had 1 pwm port on the raspberry pi. In order to combat this problem given to us inside our kit for this project was an Adafruit 16 channel 12-bit PWM/Servo Driver. This servo driver needed to have its pins soldered on which is what we did. Next, we needed to use an external voltage supply in order to power the servos because we did not want to use the voltage being emitted off from the raspberry pi because that can lead to servo malfunction and potentially blow out the circuit especially if the servo changes directions it gains heat. In order to drive these servos correctly and make sure not too much current is drawn I soldered on a 470uF capacitor. This capacitor will limit the torque and the servo current draw. This driver allowed us to control multiple servos at once in order to do this we used the Adafruit_Python_PCA9685 API in order for us to call the functions which would perform certain tasks. Each of the servos are using servo slot 0 and 2 on the PWM/Servo Driver in order to adjust the speed of the servo's we sent over different frequencies to each servo.

II.IV Creation of the Stand

The stand that we developed uses an amazon box and the amazon box is used as a base for the mechanism. The base for this mechanism is then connected to one of the continuous servos which will rotate the mechanism around the x-axis. Then the next continuous servo is sitting on a piece of cardboard ripped off from the amazon box this rotates the raspberry pi camera around the y-axis. After the contraption is built we have the raspberry pi camera sticking out on another piece of cardboard. This will revolve and take the pictures based off the user input. In order to combat the problem of the servos intertwining we programmed each servo to go to a certain minimum and maximum. We also taped down the wires in order to combat this issue.



III. Summary components

The project has two servos set on the camera stand. Servo A holds and rotates the camera on the x axis and servo B that holds the other servo rotating on the y axis. The web server takes the user's input from buttons and input boxes to operate the camera and servos. The web server uses the camera to take a picture and names it's as preview or user's input and puts it in a folder. These pictures are then called from the web server in the gallery and in the preview portion of the webpage. The web server initializes the x position and y position and displays the position each time it is changed. It then uses the user input to move the servos to a specific position or a relative position. This is done through an algorithm based on its current position, voltage and time.

IV. Overall Project thoughts

This project had a wide range of bad, good and tough characteristics. The negatives that were associated with this project was that we burned out a raspberry pi due to the fact that we inputted to much voltage into the pi, the raspberry pi camera burned out because the external voltage supply that the Adafruit PWM Driver uses sent out too much voltage for the raspberry pi camera. and soldered in correctly the Adafruit PWM driver. These mistakes resulted in the learning of different pieces of hardware and they all interact with each other. The positive aspects of this project is that we were able to move both servos using the PWM driver, we got the entire system without using a heat sync, we also got the raspberry pi camera to turn power on and off, we got it store into the gallery and finally we have everything integrated together. The tough aspects for this project was the mechanical design of the mechanism that is built out of a amazon box, another tough aspect of this project was configuring the Flask web server in order to work for our usage. For future goals we would have liked to 3d-print a stand and test the servos on that stand.

V. Group Members

Joseph Abel	Programmed Servos, Designed Contraption, Debugged Flask Web Server
Nathan Dalling	Program Flask Web Server, Designed Contraption, Tested the raspberry pi camera

VI. References

Nathan Dalling GitHub : https://github.com/n02723913/ELSpring2017
Joseph Abel GitHub: https://github.com/earthkid123/ELSpringg2017
Adafruit ,Adafruit_Python_PCA9685 :https://github.com/adafruit/Adafruit_Python_PCA9685