Software Overview

Year: 2016 Semester: Fall Team: 7 Project: ANPR Parking System

Creation Date: Nov. 7, 2016 Last Modified: December 11, 2016

Author: Zhihao Liu Email: liu869@purdue.edu

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Software Overview** | 5 | x2 | 10 |  |
| **Description of Algorithms** | 5 | x2 | 10 |  |
| **Description of Data Structures** | 5 | x2 | 10 |  |
| **Program Flowcharts** | 5 | x3 | 15 |  |
| **State Machine Diagrams** | 4 | x3 | 12 |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** | 3 | x2 | 6 |  |
| **Formatting and Citations** | 4 | x1 | 4 |  |
| **Figures and Graphs** | 5 | x2 | 10 |  |
| **Technical Writing Style** | 5 | x3 | 15 |  |
| **Total Score** | 92 | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

*This document is in general acceptable. Especially, great details are shown in the figures.*

*The biggest issue for it is writing. However, it is beyond the scope of this course, so some help out of this class is highly recommended. Besides that, some formatting issues shown before are still there. And the student may not have a solid understanding on what a state diagram is in general.*

*Still, an effort can be seen for the writing part, compared to this group’s past assignments.*

1.0 Software Overview

The automatic number plate recognition (ANPR) parking system requires two major hardware components. One is the microcontroller, which can raise and lower the gate bar, collect IR sensors’ status, change LEDs’ color to green or red and display message and information on LCD screen. The other one is Raspberry Pi, which can take photos and recognize the plate numbers captured.

The microcontroller will be responsible for the ability to take in IR sensors’ data indicating current status on parking spots and use it to change LEDs’ color accordingly. The microcontroller will collect IR sensors’ data and transmit data to Raspberry Pi. Message and information about available amount of parking spots will receive from Raspberry Pi. The gate bar is controlled by microcontroller. When microcontroller receives the data about the action for gate bar, the microcontroller will send signal to gate bar module to raise or lower the gate bar.

The Raspberry Pi 3 will take charge in taking photos and recognizing plate. The process is basically described as follows. When a car decides to drive into the parking garage. It will stop at the entrance stop line first. Then Raspberry Pi takes in the signal from infrared sensor indicating that a car is coming and Raspberry Pi controls the camera to take a picture of the license plate from the back of car. After this, by using plate recognition algorithm, the Raspberry Pi will record the plate number and store it into memory. After a long time when the car is going to leave, it will stop in front of the exit line. Then the camera will take the image again and calculate the time that the current car stays in the parking garage. Then, Raspberry Pi sends amount of parking fee to microcontroller and microcontroller shows the price on LCD screen.

2.0 Description of Algorithms

We plan to use “openalpr” - An open source project for plate recognition [1]. After Raspberry Pi gets the signal from infrared sensor via microcontroller indicating a car is in coming, the camera module will take the image from end of car and store it into disk memory. Then plate recognition algorithm will be used to recognize the number of the plate from the image. When Raspberry Pi finished recognizing the plate, it will send a signal back to microcontroller, controlling gate bar motor to raise the gate bar. The details of the algorithm will be not mentioned here because it is an open source program that we are going to just utilize it.

The microcontroller will run as a state machine and use interrupt module to retrieve infrared sensors’ data. After the car enters the parking garage, the microcontroller will lower the gate bar and get into waiting state of state machine to wait for next incoming car. There will be a buffer structure in microcontroller stored the availability parking spots, which means we can know if there is a car on parking spot from this buffer. When there is any change in this buffer, microcontroller will run a function to update the LEDs’ color accordingly, green for available parking spot and red for not available parking spot. Microcontroller can also receive message from Raspberry Pi via SPI module and display the message on LCD screen.

3.0 Description of Data Structures

In microcontroller, there will be an integer buffer for storing current status for all parking spots. 0 means the parking spot is empty and 1 will means there is a car on the parking spot.

In Raspberry Pi, a class will be construct to store license plate number, entry time and other information.

4.0 Sources Cited:

[1] GitHub, 'openalpr/openalpr', 2014. [Online]. Available: https://github.com/openalpr/openalpr/wiki/OpenALPR-Design. [Accessed: 01- Mar- 2015].

Appendix 1: Program Flowcharts



Appendix 2: State Machine Diagrams

