**Software Formalization**

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

**Creation Date:**Oct. 1, 2016 **Last Modified:** Dec. 11, 2016

**Author:** Tian Qiu **Email:** tq@purdue.edu

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Third Party Software** | 5 | x2 | 10 |  |
| **Description of Components** | 5 | X3 | 15 |  |
| **Testing Plan** | 4.5 | x3 | 13.5 |  |
| **Software Component Diagram** | 5 | x4 | 20 |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** | 4 | x2 | 8 |  |
| **Formatting and Citations** | 5 | x1 | 5 |  |
| **Figures and Graphs** | 5 | x2 | 10 |  |
| **Technical Writing Style** | 5 | x3 | 15 |  |
| **Total Score** | 96.5 | | |  |

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

General Comments:

*It is clear that you have put much thought into the software part of your project.*

*Besides, there are grammar mistakes here and there. And some parts of the document can be improved to eliminate confusion.*

*Good job in general! The diagrams are very good and they have shown many details clearly.*

1. Utilization of Third Party Software

|  |  |  |  |
| --- | --- | --- | --- |
| Name | License | Description | Usage |
| RPIO | LGPLv3+ | RPIO is an advanced GPIO module for the Raspberry Pi, which includes GPIO input, output and interrupts. [1] | We will use RPIO library for our SPI communication on the Raspberry Pi’s end. |
| OpenALPR | GPLv3 | OpenALPR is an open source Automatic License Plate Recognition library written in C++ with bindings in C#, Java, Node.js, Go, and Python. The library analyzes images and video streams to identify license plates. The output is the text representation of any license plate characters. [2] | We will use openALPR to recognize the plate number automatically. |
| C language | GPL | C is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations. [3] | We will use C standard library to develop program in microcontroller. |

*Table 1. Third Party Libraries*

1. Description of Software Components

The software for ANPR Parking System has four major components. These include a function to detect whether a car comes, a function to recognize the plate number, a function to control the gate bar, and a function to display parking statues. The use of a function call structure and software component diagram is attached in the appendix.

2.1 Camera Module

The camera module is used to take the images when car is coming or leaving. There are two types of cameras that we can use, one is USB camera and the other one is raspberry Pi camera module. We bought both of them for the entrance of parking lot and also exit of parking lot. The problem is that for a single raspberry pi, only one camera module plugin is available. We plan to keep two cameras in the same type because in this way we can get the same resolution images input for plate recognition system. If we want to keep two cameras in the same type, the only way is to use two USB camera which we chose Logitech C290 camera. The final plan has not been decided yet, but I already tested both of them and made sure that both of them are suitable and programmable.

2.2 Raspberry Pi 3

The Raspberry Pi component is responsible for image processing. 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 build-in third party plate recognition algorithm aforementioned, 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 the amount of parking fee to microcontroller and microcontroller shows the price on LCD screen.

2.3 Microcontroller

The responsibility of microcontroller is to control all electrical components other than the Raspberry Pi module. These include controlling motor module to raise and lower gate barrier, controlling segments displays and collecting sensors data. When a car stops at the entrance of parking garage, the infrared sensors on the ground will detect a car above it and set voltage high to an assigned pin at microcontroller indicating a car stops at entrance. After microcontroller receives the signal from infrared sensor, microcontroller will send a signal to Raspberry Pi via I/O pin. Then microcontroller will receive a signal from Raspberry Pi via I/O pin indicating that Raspberry Pi has already process the license plate image. Then, microcontroller will raise the gate barrier via motor module, using two I/O pins from microcontroller.

2.4 16-Segments display

Serial 16-Segments display will be used to display license plate number at entrance and exit. Microcontroller will send signal to led drivers that drive the segments display.

1. Testing Plan

The importance of successfully verifying that component is as follows.

3.1 Microcontroller

1. Test whether microcontroller can collect infrared sensors’ data via assigned I/O pin.
2. Test whether microcontroller can send signal to motor module via two assigned I/O pins.
3. Test whether the data can be sent to and received from the Raspberry Pi over SPI.

3.2 Raspberry Pi 3

1. Test whether the data can be sent to and received from the microcontroller over SPI.
2. Test the connection between Raspberry Pi 3 and cameras by taking photos.
3. Test whether when receive car existing signal from microcontroller, the Raspberry Pi 3 can launch the plate recognition program.
4. Test whether the plate recognition program can produce the credible results based on sample images.
5. Test whether the Raspberry Pi 3 can send the results back to microcontroller.

3.3 Camera Module

1. Use Raspberry Pi 3 camera module and C290 camera to take three images for sample plate in different angle respectively.
2. Store the images into Raspberry Pi 3 disk memory.
3. Use ALPR algorithm to test all the images and store all the results into database.
4. Compare all the results to the sample plate to see the credibility of the sample images.

3.4 16-Segments display

1. Test whether the signal from microcontroller can correctly display.

4.0 Sources Cited:

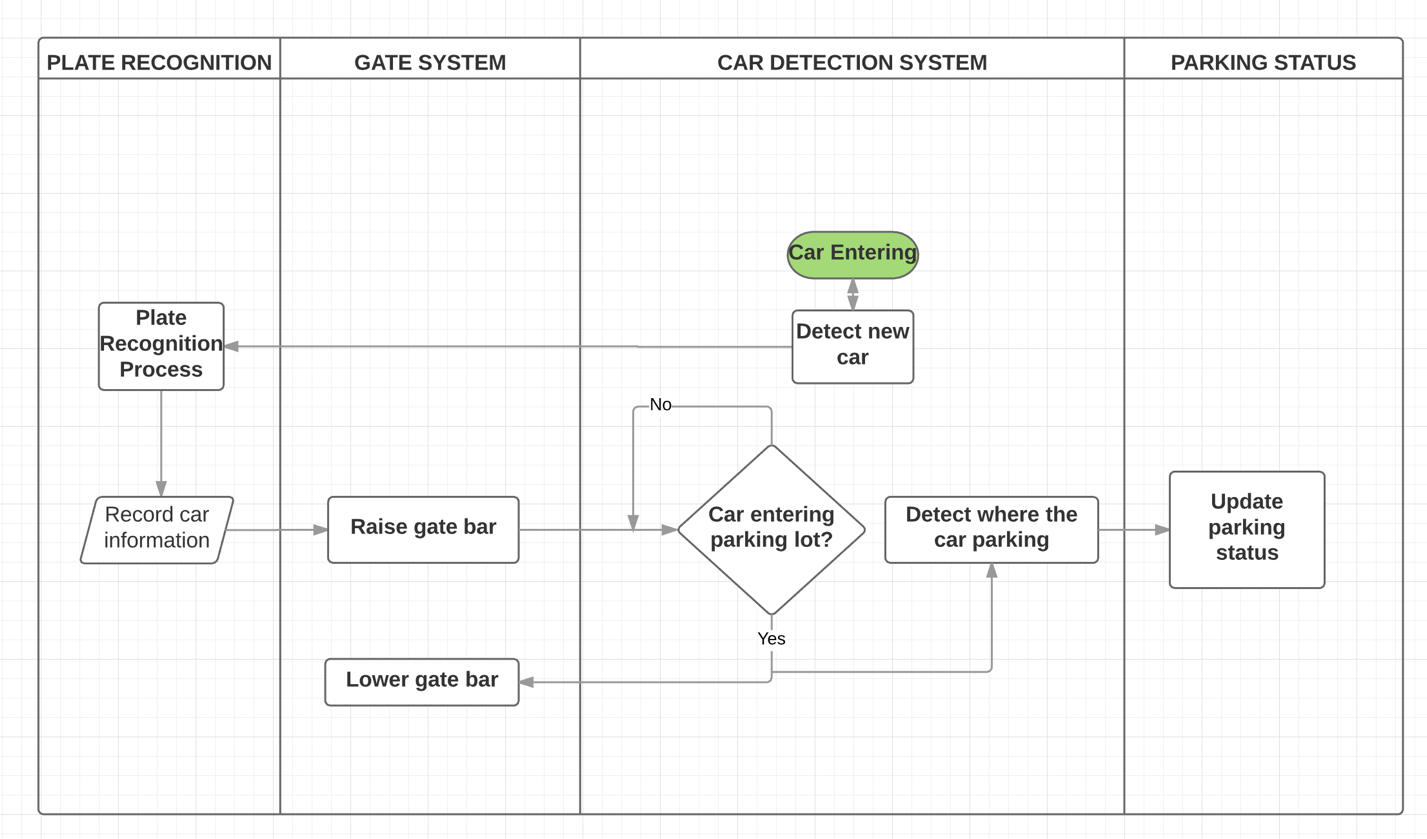
[1] Raspberry Pi Inc.(n.d.) BCM2835 ARM Peripherals [Online]. Available: <https://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>

[2] Chris Hager (2013) RPIO’s documentation [Online]. Available:

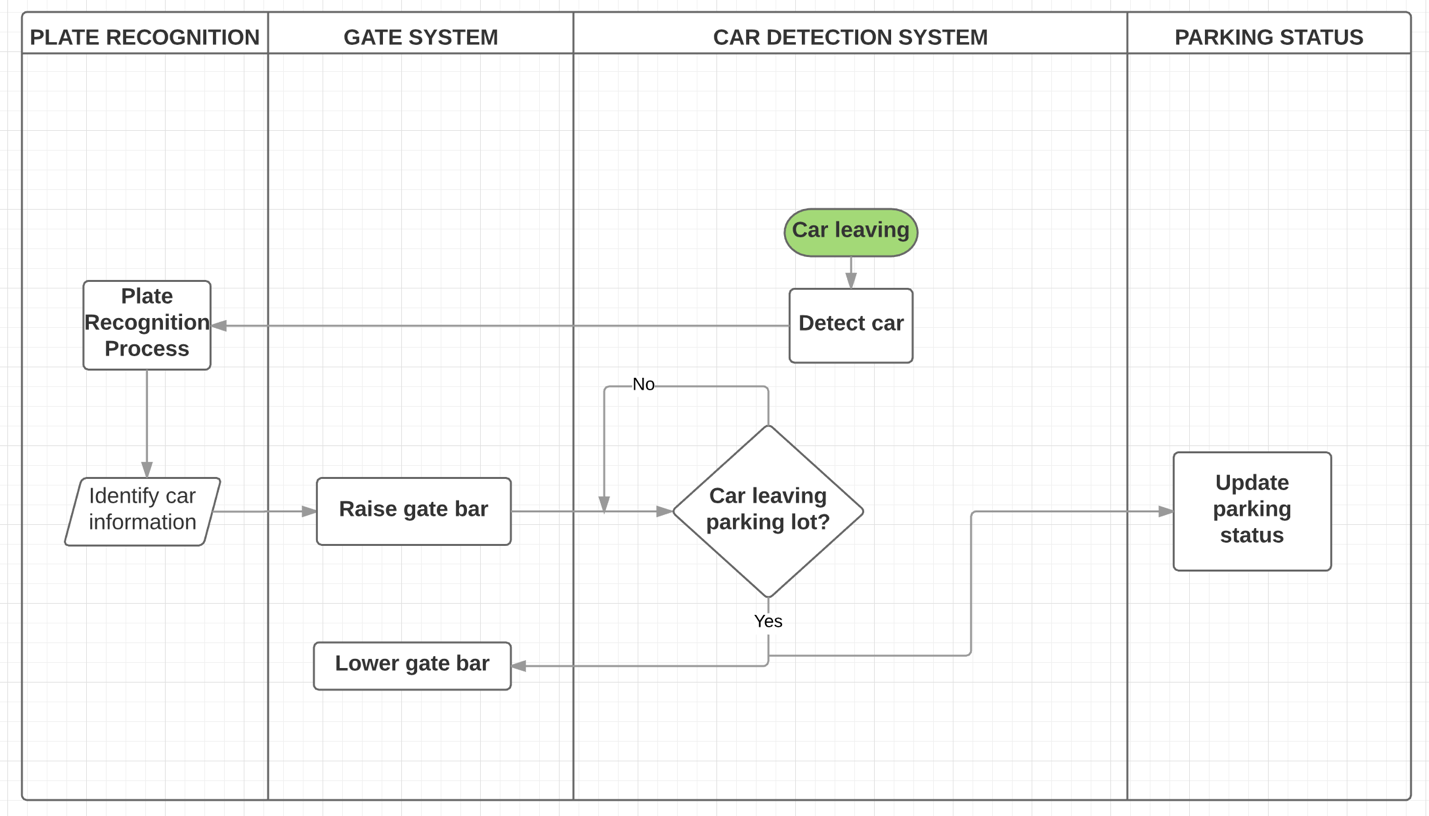
<https://pythonhosted.org/RPIO/>

[3] Free Software Foundation, Inc. (n.d.) GCC, the GNU Compiler Collection [Online]. Available: <https://gcc.gnu.org>

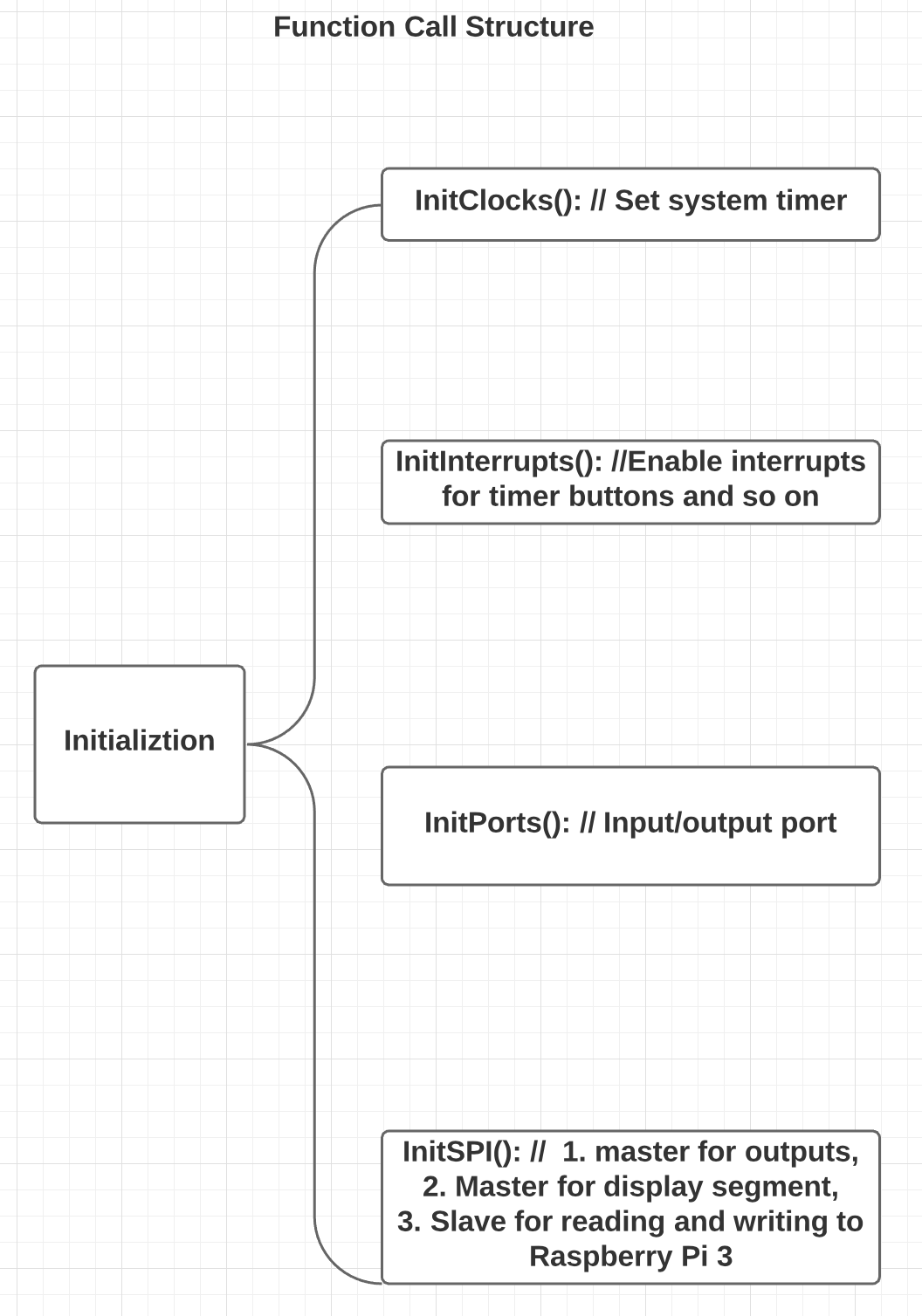
Appendix 1: Software Component Diagram



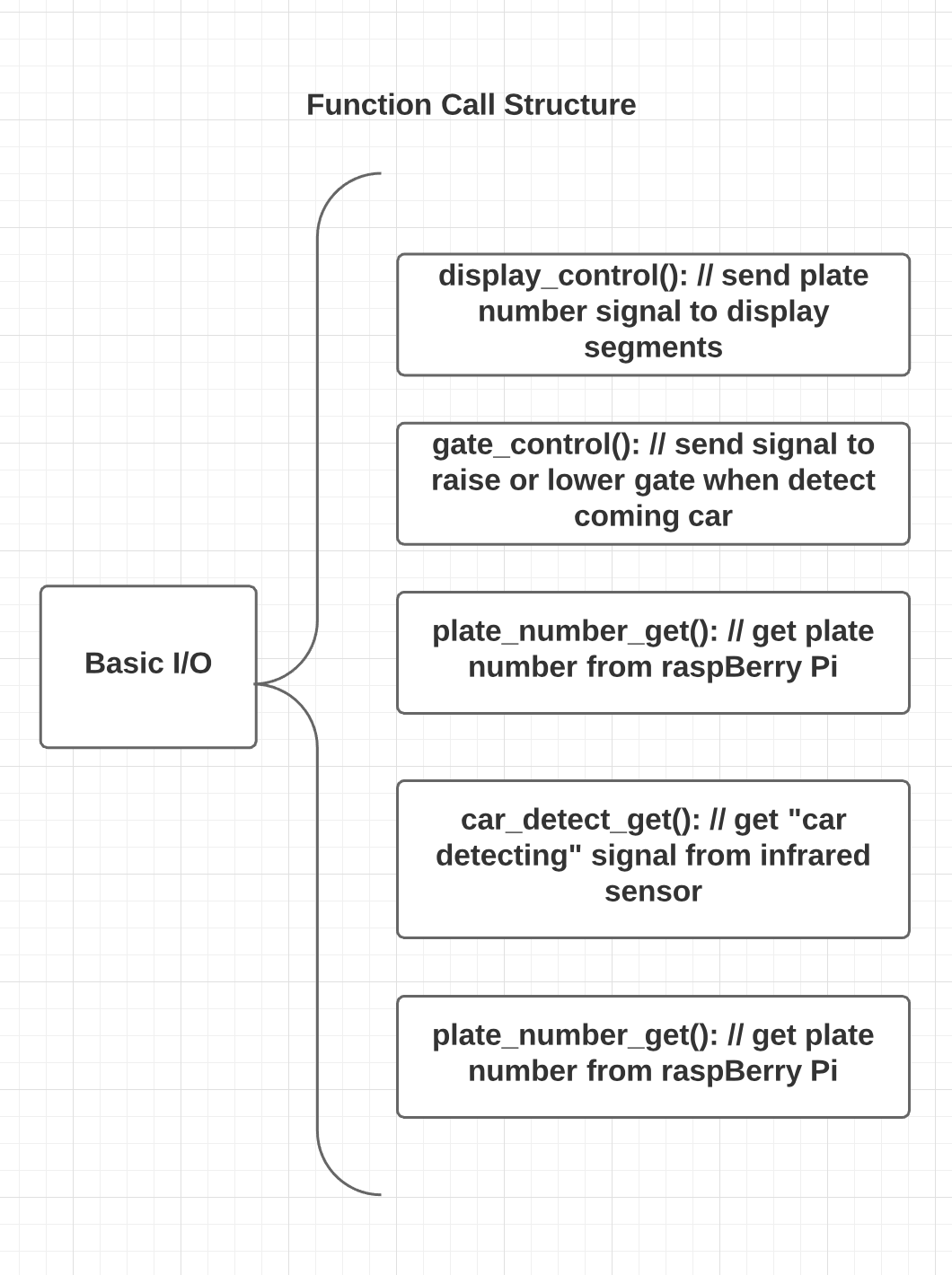
*Diagram 1. High level system structure for entering car*



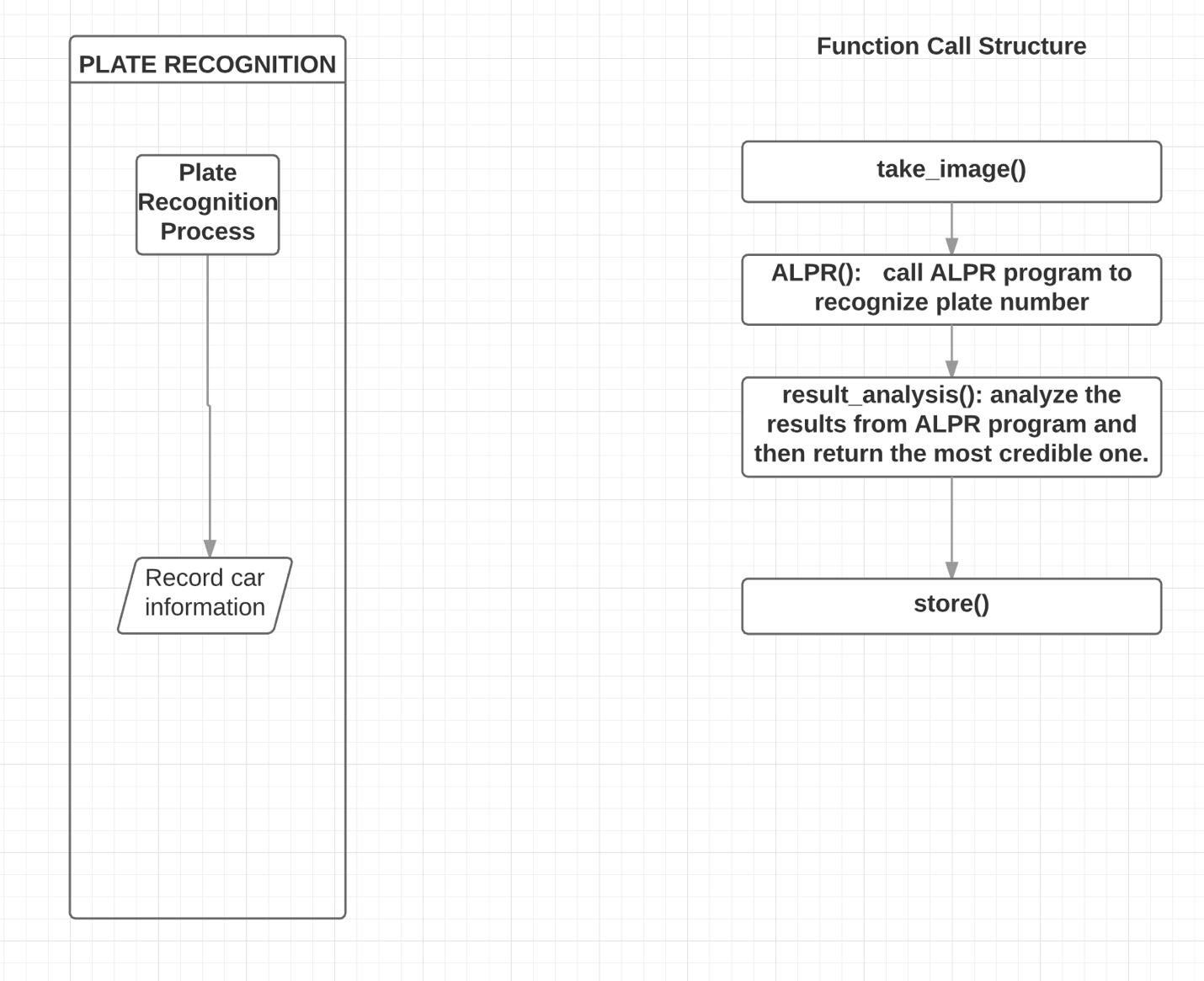
*Diagram 2. High level system structure for leaving car*



*Diagram 3. Function call structure for microcontroller initialization*



*Diagram 4. Function call structure for major I/O in microcontroller*



*Diagram 5. Function call structure for Raspberry Pi plate recognition*