|  |  |
| --- | --- |
| **­** | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

|  |  |
| --- | --- |
| **Report 3** | |
| **Parking Guidance System Solution** | |
| **Group 1** | |
| **Group members** | Trần Nguyễn Minh Trung – Team Leader – SE61496  Bùi Phú Hiệp – Team Member – SE61438  Nguyễn Đỗ Phương Huy – Team Member – SE61358 |
| **Supervisor** | Nguyễn Đức Lợi |
| **Ext. Supervisor** | N/A |
| **Capstone Project Code** | PGSS |

- Ho Chi Minh City, Jan, 2017

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 1](#_Toc475466556)

[List of Tables 3](#_Toc475466557)

[List of Figures 3](#_Toc475466558)

[Definitions, Acronyms and Abbreviations 3](#_Toc475466559)

[C. Software – Hardware Requirement Specification 4](#_Toc475466560)

[1. User Requirement Specification 4](#_Toc475466561)

[1.1. Parking Guidance System 4](#_Toc475466562)

[1.2. Mobile Application 4](#_Toc475466563)

[2. System Requirement Specification 4](#_Toc475466564)

[2.1. External Interface Requirement 4](#_Toc475466565)

[2.1.1. User Interface 4](#_Toc475466566)

[2.1.2. Hardware Interface 5](#_Toc475466567)

[2.1.2.1. Block Diagram 5](#_Toc475466568)

[2.1.2.2. Raspberry Pi 3 6](#_Toc475466569)

[2.1.2.3. Arduino Nano 7](#_Toc475466570)

[2.1.2.4. Compass Module 3-Axis HMC5883L 8](#_Toc475466571)

[2.1.2.5. RF module nRF24L01+ 9](#_Toc475466572)

[2.1.2.6. Information LED Display Module 10](#_Toc475466573)

[2.1.2.7. Indicator LED Module 13](#_Toc475466574)

[2.1.3. Software Interface 16](#_Toc475466575)

[2.1.4. Communication Protocol 16](#_Toc475466576)

[2.2. System Overview Use Case 17](#_Toc475466577)

[2.3. List of Use Case 18](#_Toc475466578)

[2.3.1. Manager Use Case 18](#_Toc475466579)

[2.3.2. Administrator Use Case 22](#_Toc475466580)

[2.3.3. End User Use Case 23](#_Toc475466581)

[3. Software System Attribute 26](#_Toc475466582)

[3.1. Usability 26](#_Toc475466583)

[3.2. Reliability 26](#_Toc475466584)

[3.3. Availability 26](#_Toc475466585)

[3.4. Security 26](#_Toc475466586)

[3.5. Maintainability 26](#_Toc475466587)

[3.6. Portability 26](#_Toc475466588)

[3.7. Performance 26](#_Toc475466589)

[4. Conceptual Diagram 27](#_Toc475466590)

# List of Tables

[Table 1: Definitions, Acronyms and Abbreviations 3](#_Toc475613430)

[Table 2: Raspberry Pi 3 – Specification 6](#_Toc475613431)

[Table 3: Arduino Nano - Specification 8](#_Toc475613432)

[Table 4: The Compass Module 3-Axis HMC5883L - Pin Function 9](#_Toc475613433)

# List of Figures

[Figure 1: PGSS Block Diagram 5](#_Toc475466537)

[Figure 2: Raspberry Pi 3 6](#_Toc475466538)

[Figure 3: Arduino Nano 7](#_Toc475466539)

[Figure 4: Compass Module 3-Axis HMC5883L 8](#_Toc475466540)

[Figure 5: RF module nRF24L01+ 9](#_Toc475466541)

[Figure 6: RF module nRF24L01+ - Specification 10](#_Toc475466542)

[Figure 7: 7-segment LED Display 11](file:///C:\Users\TrungTNM\Documents\-FPT-CAPSTONE-PGSS\Common\Reports\Report%203.docx#_Toc475466543)

[Figure 8: TPIC6B595 Power Logic 8-Bit Shift Register 12](#_Toc475466544)

[Figure 9: TPIC6B595 Pin outs 13](#_Toc475466545)

[Figure 10: RGB LED common anode 14](#_Toc475466546)

[Figure 11: RGB LED common anode pin-out 15](#_Toc475466547)

[Figure 12: TIP122 Transistor 16](#_Toc475466548)

[Figure 13: Overview use case diagram 17](#_Toc475466549)

[Figure 14: Manager Use case diagram 18](#_Toc475466550)

[Figure 15: Conceptual Diagram 27](#_Toc475466551)

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| PGS | Parking Guidance System |
| Parking area | An area set aside for parking vehicles, aircraft, etc. |
| Parking lot | A place inside parking area that provide space for one vehicle |
| IoT | Internet of Things |
| CCU | Central Control Unit |

Table 1: Definitions, Acronyms and Abbreviations

# C. Software – Hardware Requirement Specification

## Software Requirement Specification

### Software Requirement

Manager can show the information of their car park to the end user, which will increase the interaction between car park provider and end user. The information include:

* Address
* Contact info
* Number of empty parking lot

End user can find the nearest car park, which has empty parking lot.

Manager can manage their car park easily; make an automatic system to guide the end user base on the interaction panel, which show number of empty parking lot in each area and the status light on each parking lot.

Users can see empty slot and detail information about parking area by touching a marker on map.

User can reserve a parking slot.

### GUI Requirement

User interface of mobile app must be simple, clearly and easy to use.

The color of mobile app must be elegant, not garish.

Each UI element must be arranged logically, allowing user access easily.

Meet all main function requirements.

## Hardware Requirement Specification

### Hardware Requirement

#### Hardware Interface

The hardware interface must satisfied the following requirements:

* Easy to replace
* Low-cost module
* Easy to implement

Based on project requirement we have choose following hardware components.

##### Block Diagram

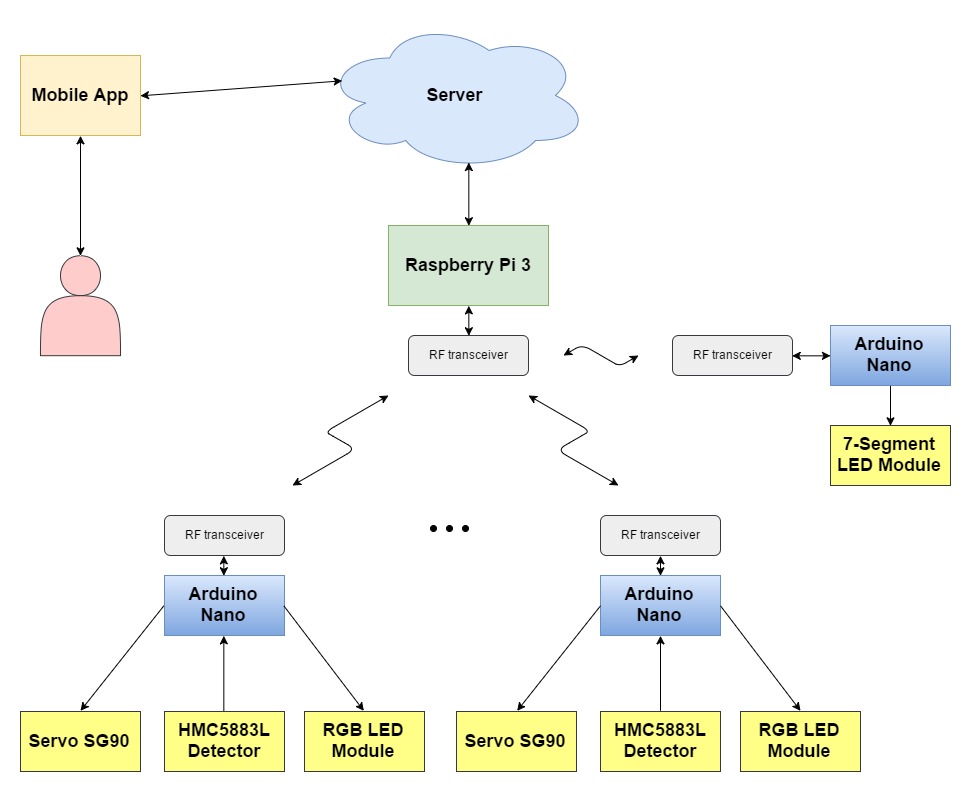


Figure 1: PGSS Block Diagram

##### Raspberry Pi 3



Figure 2: Raspberry Pi 3

**Overview:** To communicate with all other hardware component and processing value, we must have a Central control unit, there are many kind of central control unit in the market. After evaluate requirement of project, we decide to choose Raspberry Pi 3. Raspberry Pi 3 is powerful mini-computer with many features.

**Specification:**

|  |  |
| --- | --- |
| SoC | Broadcom BCM2837 |
| CPU | 4× ARM Cortex-A53, 1.2GHz |
| GPU | Broadcom VideoCore IV |
| RAM | 1GB LPDDR2 (900 MHz) |
| Network | 10/100 Ethernet, 2.4GHz 802.11n wireless |
| Bluetooth | Bluetooth 4.1 Classic, Bluetooth Low Energy |
| Storage | microSD |
| GPIO | 40 |

Table 2: Raspberry Pi 3 – Specification

More details about Raspberry Pi 3:

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

##### Arduino Nano

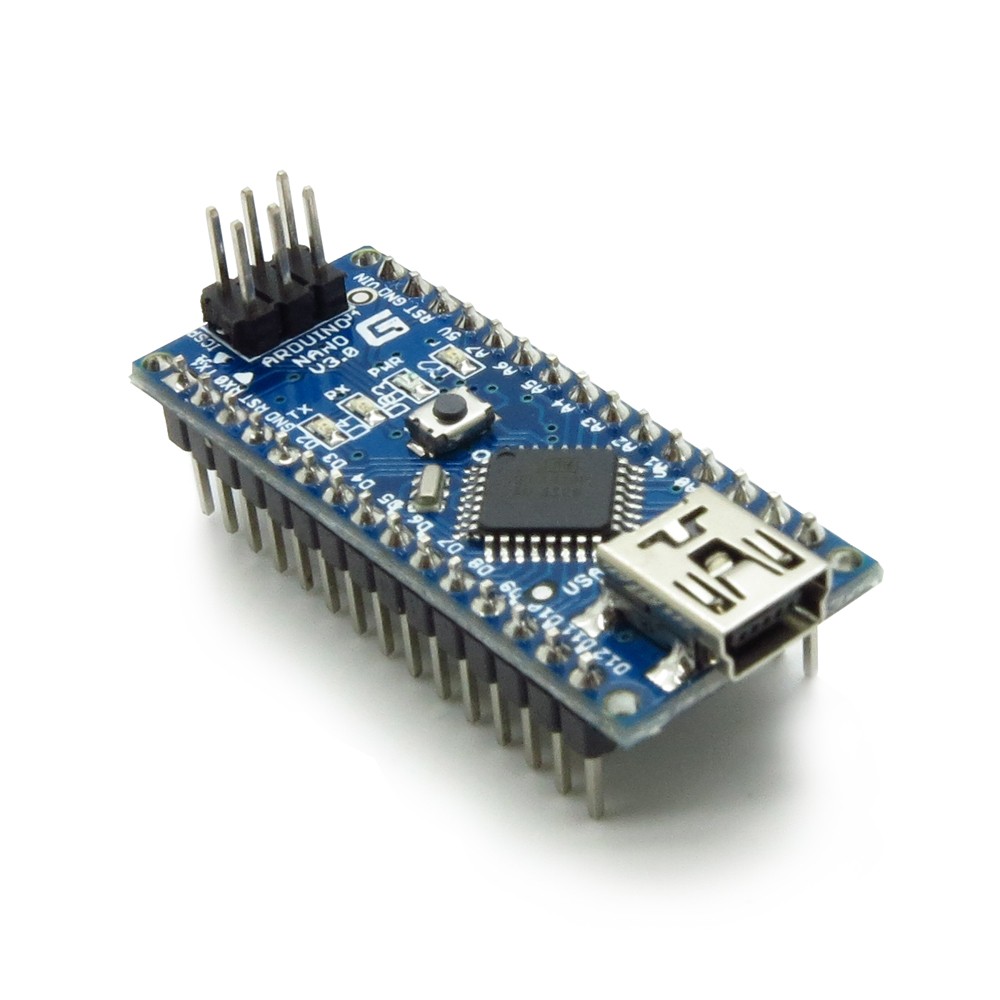


Figure 3: Arduino Nano

**Overview:** The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x).

**Specification:**

|  |  |
| --- | --- |
| Microcontroller | ATmega328 |
| Architecture | AVR |
| Operating Voltage | 5 V |
| Flash Memory | 32 KB of which 2 KB used by bootloader |
| SRAM | 2 KB |
| Clock Speed | 16 MHz |
| Analog I/O Pins | 8 |
| EEPROM | 1 KB |
| DC Current per I/O Pins | 40 mA (I/O Pins) |
| Input Voltage | 7-12 V |
| Digital I/O Pins | 22 |
| PWM Output | 6 |
| Power Consumption | 19 mA |
| PCB Size | 18 x 45 mm |
| Weight | 7 g |
| Product Code | A000005 |

Table 3: Arduino Nano - Specification

More detail about Arduino Nano:

<https://www.arduino.cc/en/Main/arduinoBoardNano>

##### Compass Module 3-Axis HMC5883L

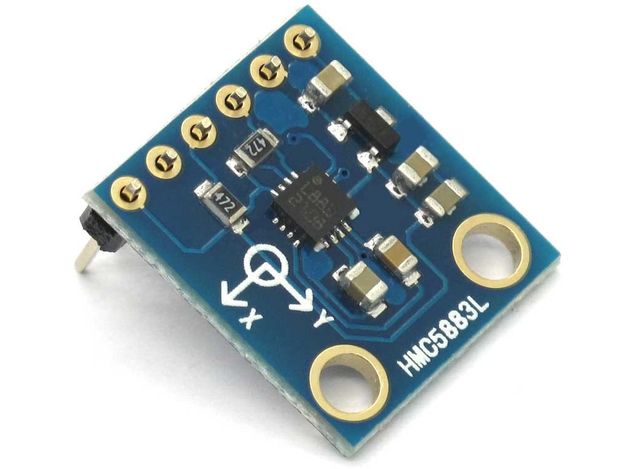


Figure 4: Compass Module 3-Axis HMC5883L

**Overview:** For detecting obstacle, we choose The Compass Module 3-Axis HMC5883L instead of ultrasonic sensor because ultrasonic sensor has many weaknesses, they are not accuracy, cannot be used outdoor in the bad weather in Vietnam.

The Compass Module 3-Axis HMC5883L is a low-field magnetic sensing device with a digital interface.

We choose The Compass Module 3-Axis HMC5883L because:

* It has reasonable price.
* Compatible with arduino and other board.
* Compact size.

**Specification:**

Input and Output Pins:

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | | **I/O** | **Function** |
| **Name** | **No.** |
| **VIN** | **1** |  | Supply Voltage – 2.7 to 6.5 VDC |
| **GND** | **2** |  | Ground |
| **SCL** | **3** | **I** | I²C Clock |
| **SDA** | **4** | **IO** | I²C Data |
| **RDY** | **5** | **I** | Data Ready |

Table 4: The Compass Module 3-Axis HMC5883L - Pin Function

##### RF module nRF24L01+

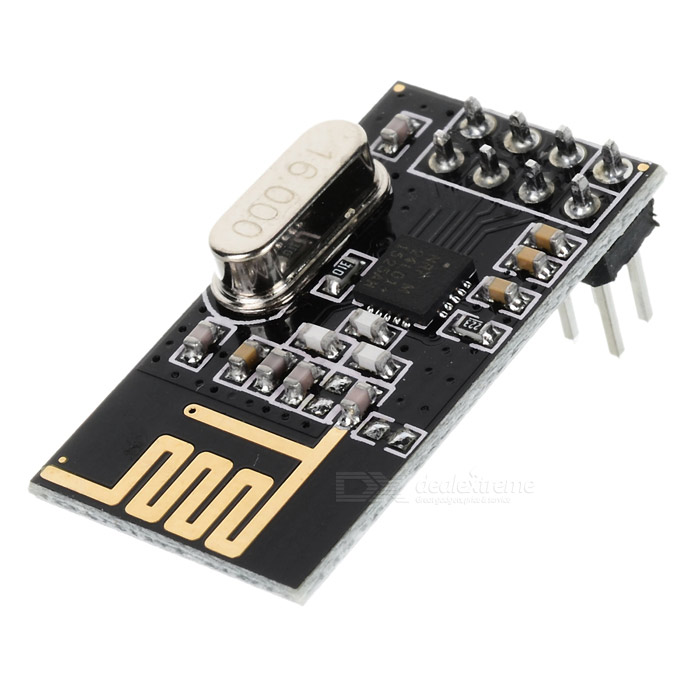


Figure 5: RF module nRF24L01+

**Overview:** Reason for PGSS use RF module nRF24L01 to communicate between Central control unit and other hardware component:

* It has reasonable price.
* Easy to buy.
* Ultra low power consumption.

**Specification:**

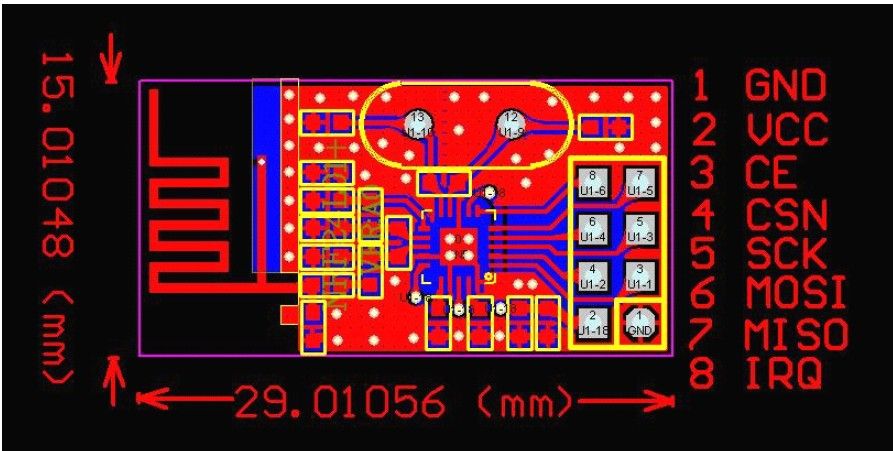


Figure 6: RF module nRF24L01+ - Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | | **I/O** | **Description** |
| **No.** | **Name** |
| 1 | GND |  | Power Supply Ground |
| 2 | VCC |  | 3.3V |
| 3 | CE | I | Chip Enable |
| 4 | CSN | I | SPI Chip Select |
| 5 | SCK | I | SPI Clock |
| 6 | MOSI | I | SPI Slave Data Input |
| 7 | MISO | O | SPI Slave Data Output |
| 8 | IRQ | O | Maskable Interrupt Pin |

Table 5: RF module nRF24L01+ - Pin Function

##### Information LED Display Module

Information LED Display Module include: 7-segment LED Display, TPIC6B595 Power Logic 8-Bit Shift Register

###### **F:\CP\Image\7seg.jpg7-segment LED Display**

Figure 7: 7-segment LED Display

**Specification:**

* 0.56 inch digit height
* Super Red emitting color
* White segment color, gray face
* Low current operation
* Easy mounting on PCB boards or sockets

###### **TPIC6B595 Power Logic 8-Bit Shift Register**

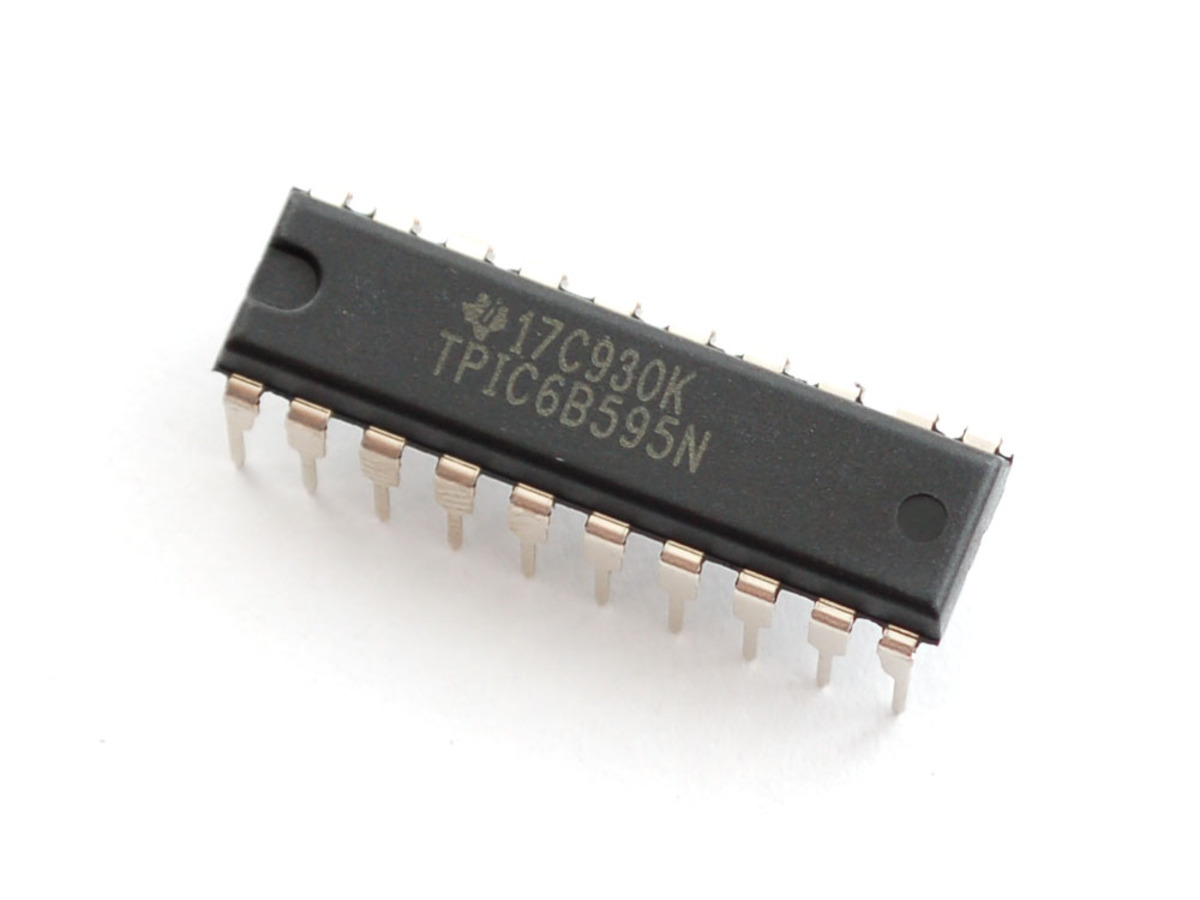


Figure 8: TPIC6B595 Power Logic 8-Bit Shift Register

**Specification:**

To display high power 7-segment display, we choose IC TPIC6B595 instead of IC 74HC595 because TPIC6B595 is asimple shift register IC that can control high-voltage/high-current devices directly**.** Each output pin can sink 150mA and then supports the maximum of Load Voltage at 50V.

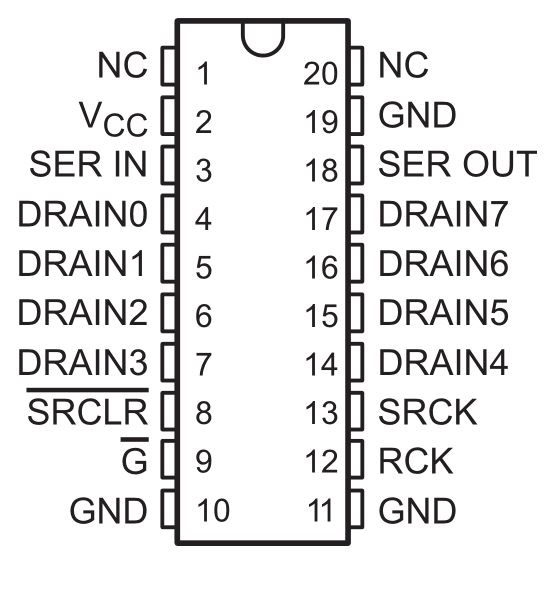


Figure 9: TPIC6B595 Pin-outs

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | | **I/O** | **Description** |
| **Name** | **No.** | O | Open-drain output |
| DRAIN0 | 4 |
| DRAIN1 | 5 |
| DRAIN2 | 6 |
| DRAIN3 | 7 |
| DRAIN4 | 14 |
| DRAIN5 | 15 |
| DRAIN6 | 16 |
| DRAIN7 | 17 |
| G | 9 | I | Output enable, active-low |
| GND | 10,11,19 | - | Power ground |
| NC | 1, 20 | - | No internal connection |
| RCK | 12 | I | Register clock |
| SERIN | 3 | I | Serial data input |
| SEROUT | 18 | O | Serial data output |
| SRCK | 15 | I | Shift register clock |
| SRCLR | 8 | I | Shift register clear, active-low |
| VCC | 2 | I | Power supply |

Table 6: IC TPIC6B595 - Pin Function

##### Indicator LED Module

Indicator LED Module include: Common anode RGB LED, TIP122 Transistor

###### **RGB LED common anode**



Figure 10: RGB LED common anode

**Overview:**

RGB LED allows you to change the lights to any color to show state of parking slot.

**Specification:**

* Forward Voltage (RGB): (2.0, 3.2, 3.2)V
* Max Forward Current (RGB): (20, 20, 20)mA
* Max Luminosity (RGB): (2800, 6500, 1200)mcd

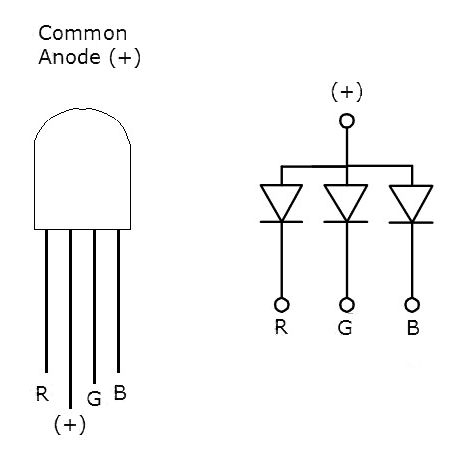


Figure 11: RGB LED common anode pin-out

###### **TIP122 Transistor**

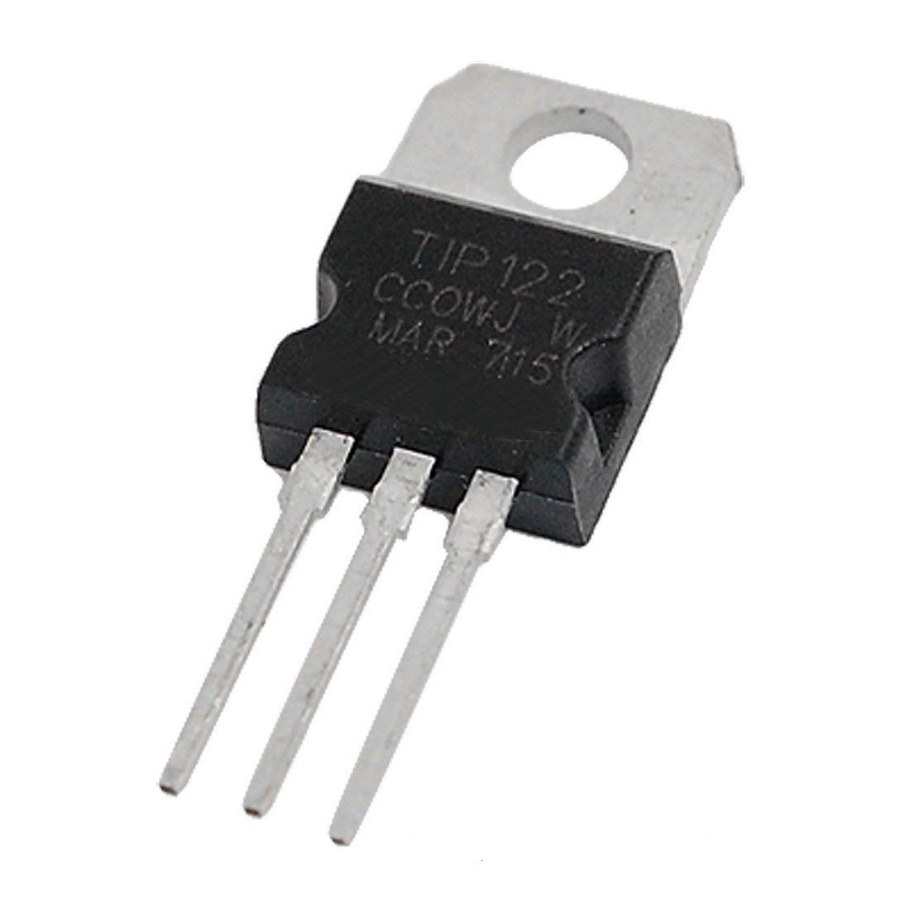


Figure 12:: TIP122 Transistor

**Overview:** A single digital pin on Arduino Nano do not provide enough current to power RGB LED, A solution for this situation is to use an NPN Darlington Transistor designed for medium power linear switching applications, so we use TIP122 Transistor to provide RGB LED with power from an external source. It can power devices up to 100VDC at 5 Amps.

**Specification:**

* TIP122 is power transistors
* Collector Current: 5 ampere
* Collector-Emitter Volt: 100 volts
* Power Dissipation: 65 watts

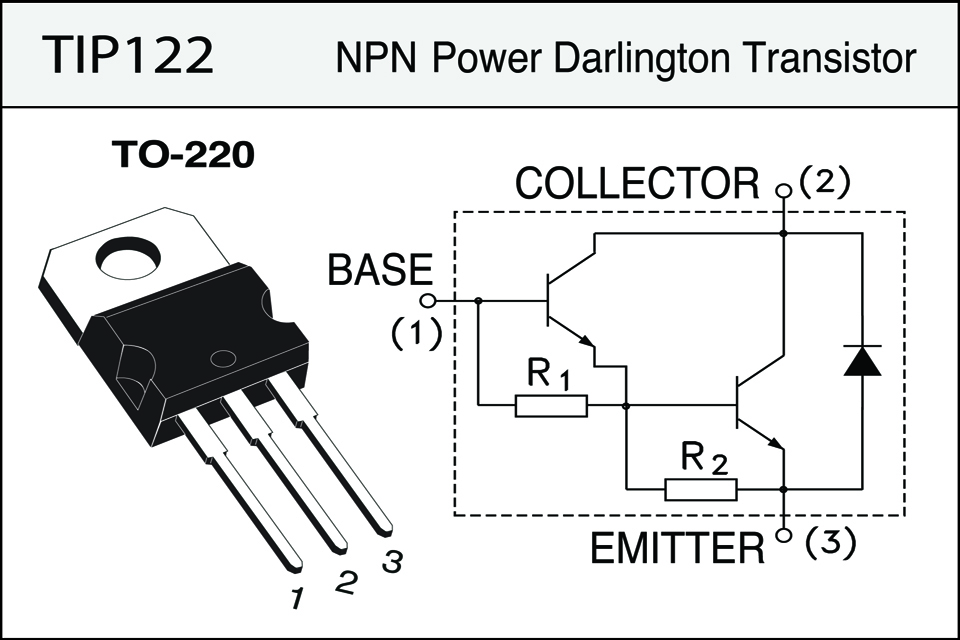


Figure 12: TIP122 Transistor- Pin Layout

##### Servo Motor SG90

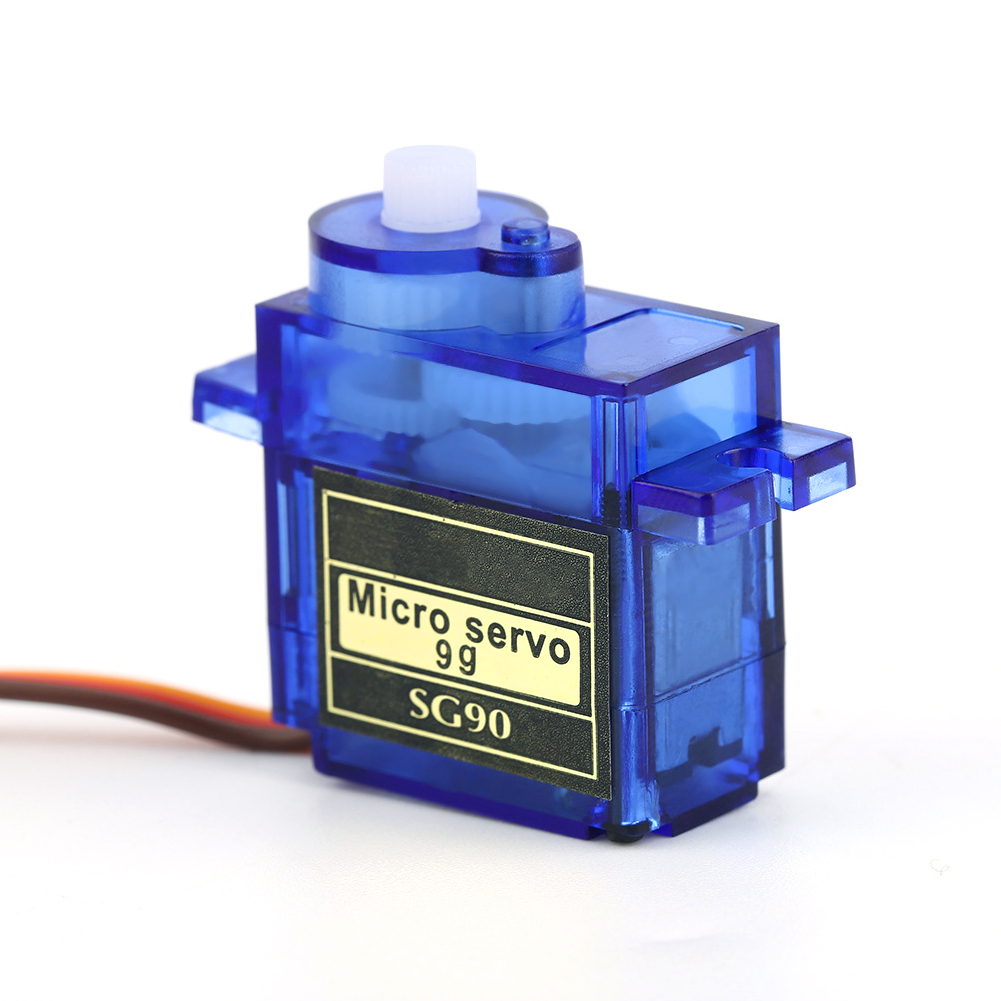


Figure 13:: Servo Motor – Tower Pro SG90

**Overview:** PGSS use Servo Motor SG90 to control barrier at each parking slot.

**Specification:**

|  |  |
| --- | --- |
| **Torque** | 1.80 kg-cm at 4.8V |
| **Speed** | 0.1sec/60° at 4.8V |
| **Voltage** | 4.0V to 7.2V, 4.6V - 5.2V nominal |
| **Dimensions** | 23mm x 12.2mm x 29mm |
| **Rotation range** | 180o |
| **Weight** | 9g |
| **Pulse width** | 500-2400uS |
| **Operating Temperature range** | 30°C to 60°C |

Table 7: Servo Motor SG90 – Specification

|  |  |  |
| --- | --- | --- |
| **Pin of Servo SG90** | **Name** | **Description** |
| Red | VCC | Power supply 5V |
| Black | GND | Power supply ground |
| Yellow | Signal | The servo will move based on the signal sent to signal wire. |

Table 5: Servo Motor SG90 – Pin-outs

#### Communication Protocol

* We communicate between hardware component and board through GPIO pin.
* Arduino Nano board communicate with Raspberry Pi 3 via RF Module.

### System Overview Use Case

E:\Downloads\CapstoneUseCases.png

Figure 13: Overview use case diagram

### List of Use Case

#### Manager Use Case

E:\Downloads\ManagerUseCase.png

Figure 14: Manager Use case diagram

**Use case specifications**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-1 specification** | | | |
| **Use-case no.** | PGSS01 | **Use-case version** | 1.0 |
| **Use-case name** | Configuration System | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * Manager   **Summary:**   * This use case allow user to change the configuration of their system.   **Goal:**   * Manager can change the information of car park, which show to the end user.   **Triggers:**   * User click on “Setting” button.   **Preconditions:**   * Mobile application is already launch. * Manager has been logged in   **Post Conditions:**   * **On Success:** New configuration is apply and save to server * **On Failure:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User click on “Setting” button | Application navigate to “Setting” menu | | 2 | User select option in the Menu  Change by click toggle or change value in the text box  Select “Submit” button | Change the value and save to server |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Business Rules:**   * N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-2 specification** | | | |
| **Use-case no.** | PGSS02 | **Use-case version** | 1.0 |
| **Use-case name** | Manage Car Park | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * Manager   **Summary:**   * This use case allow user to change their car park info.   **Goal:**   * Manager can change the information of car park, which show to the end user.   **Triggers:**   * User select their car park. * Click “Edit”   **Preconditions:**   * Mobile application is already launch. * Manager has been logged in   **Post Conditions:**   * **On Success:** New information of edited car park saved to server. * **On Failure:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User select car park.  User click on “Edit” button | Application navigate to “Setting” menu | | 2 | User select option in the Menu  Change by click toggle or change value in the text box  Select “Submit” button | Change the value and save to server |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Business Rules:**   * N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-3 specification** | | | |
| **Use-case no.** | PGSS03 | **Use-case version** | 1.0 |
| **Use-case name** | Manage Area | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * Manager   **Summary:**   * This use case allow user to change the status of each area.   **Goal:**   * The status of selected area updated and change in mobile app.   **Triggers:**   * User select their car park. * User select area in selected car park.   **Preconditions:**   * Mobile application is already launch. * Manager has been logged in   **Post Conditions:**   * **On Success:** New configuration is apply and save to server * **On Failure:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User select car park | Application change to car park detail page | | 2 | User select area in the selected car park | Application change to area detail page | | 3 | User select status in the drop down list.  Click “Update” button | The status of the area will change on server and update in mobile application |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Business Rules:**   * N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-4 specification** | | | |
| **Use-case no.** | PGSS04 | **Use-case version** | 1.0 |
| **Use-case name** | Manage Parking Slot | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * Manager   **Summary:**   * This use case allow user to manage the parking slot.   **Goal:**   * The status of selected area updated and change in mobile app.   **Triggers:**   * User select their car park. * User select area in selected car park. * Then select parking slot   **Preconditions:**   * Mobile application is already launch. * Manager has been logged in   **Post Conditions:**   * **On Success:** New configuration is apply and save to server * **On Failure:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User select car park | Application change to car park detail page | | 2 | User select area in the selected car park | Application change to area detail page | | 3 | User select parking slot to edit  After change information, select “Update” button | The information of parking slot is change on server and update in mobile application. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User select car park | Application change to car park detail page | | 2 | User select area in the selected car park | Application change to area detail page | | 3 | User click menu beside list parking spot to delete. | The parking spot will be set to deleted in server and update in mobile app. |   **Exceptions:**   * N/A   **Business Rules:**   * N/A | | | |

#### Administrator Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-5 specification** | | | |
| **Use-case no.** | PGSS05 | **Use-case version** | 1.0 |
| **Use-case name** | Add Car Park | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * Administrator   **Summary:**   * This use case allow user to add new car park to the system   **Goal:**   * New car park is added and save to server.   **Triggers:**   * User click on “Add” button.   **Preconditions:**   * Mobile application is already launch. * Administrator has been logged in   **Post Conditions:**   * **On Success:** New car park is save to server * **On Failure:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User click on “Add” or “+” button | Application navigate to add car park menu | | 2 | User fill in the textbox  Select “Submit” button | New car park with filled in info is added to server |   **Alternative Scenario:**   * N/A   **Exceptions:**   * Name of the car park is unique * Address of the car park is unique (don’t has same latitude and longitude)   **Business Rules:**   * N/A | | | |

#### End User Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-6 specification** | | | |
| **Use-case no.** | PGSS06 | **Use-case version** | 1.0 |
| **Use-case name** | Check number of empty slot | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * End User   **Summary:**   * This use case allow user view number of empty slot in each car park   **Goal:**   * Show number of empty slot   **Triggers:**   * User login to the mobile application   **Preconditions:**   * Mobile application is already launch. * End user had logged in.   **Post Conditions:**   * **On Success:** User know the number of empty slot in car park * **On Failure:** Don’t show number of empty slot in car park   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User log in to the application | Show the map with the marker as car park and the number, which indicate the number of empty slot |   **Alternative Scenario:**   * N/A   **Exceptions:**   * The number will have tick/ exclamation points to show that the number is recently update or not.   **Business Rules:**   * Tick: recently update * Exclamation points: number is not update in more than 1 hour. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-7 specification** | | | |
| **Use-case no.** | PGSS07 | **Use-case version** | 1.0 |
| **Use-case name** | Book parking slot | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * End User   **Summary:**   * This use case allow user to book parking slot before go to the car park   **Goal:**   * Book the parking slot before go to car park   **Triggers:**   * User has selected the car park to book   **Preconditions:**   * Mobile application is already launch. * End user had logged in.   **Post Conditions:**   * **On Success:** User book the parking slot success * **On Failure:** Show error message when book   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User log in to the application | Show the map with the marker as car park and the number, which indicate the number of empty slot | | 2 | User select the car park they want to book | Show the “Book” button if has empty slot | | 3 | Fill information for transaction  Click “Submit” | Make a transaction and set one parking slot to booked  Show the address of booked parking slot to the user |   **Alternative Scenario:**   * N/A   **Exceptions:**   * Transaction fail by 3rd party.   **Business Rules:**   * N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case-8 specification** | | | |
| **Use-case no.** | PGSS08 | **Use-case version** | 1.0 |
| **Use-case name** | Search car park | | |
| **Author** | Bui Phu Hiep | | |
| **Date** | 13/02/17 | **Priority** | High |
| **Actor:**   * End User   **Summary:**   * This use case allow user to search a car park by name or address   **Goal:**   * Show the searched car park   **Triggers:**   * User login to the mobile application   **Preconditions:**   * Mobile application is already launch. * End user had logged in.   **Post Conditions:**   * **On Success:** Show the searched car park on the map if success * **On Failure:** Show message error   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User log in to the application | Show the map with the marker as car park and the number, which indicate the number of empty slot | | 2 | Enter the name or address in the search bar  Press “Enter” or click “Search” | Find the car park base on name or address then focus on the map. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No.** | **Actor Action** | **System Response** | | 1 | User log in to the application | Show the map with the marker as car park and the number, which indicate the number of empty slot | | 2 | Enter the name or address in the search bar  Press “Enter” or click “Search” | Show message don’t have car park if the name or address is incorrect |   **Exceptions:**   * N/A   **Business Rules:**   * N/A | | | |

## Software System Attribute

### Usability

* User controls all system components via only mobile application.
* The system can install easily.
* User can learn how to use the system fast.

### Reliability

### Availability

* The mechanical component require electrical system to work well.
* Hardware components are easy to find in the market.

### Security

* Mobile application require authentication and authorization implement well because manager and end user use the same application.

### Maintainability

* Use plug and play component so we can easily replace it.

### Portability

* Easy to construct.

### Performance

* Detection car is fast, less then 50ms.
* The speed of server can scale base on the budget easily.

## Conceptual Diagram

E:\Downloads\Entity.png

Figure 15: Conceptual Diagram

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| CarPark | Descript all car park information in the system |
| Area | Describe all area detail in car park |
| ParkingLot | Describe parking lot information in the area |
| Item | Describe hardware item in each parking lot |
| BookingHistory | Describe the booking history of the user |
| Transaction | Save the transaction of each booking |