George Alexandris

N01163751

Project Name: IR Break Beam Sensor

CENG317-0NC

December 13, 2019

Table of Contents

[1. Abstract 3](#_Toc27342511)

[2. Introduction to Report 3](#_Toc27342512)

[3. Proposal 3](#_Toc27342513)

[3.1. Introduction and Background for Proposal 3](#_Toc27342514)

[3.2. Skills and Courses Related: 3](#_Toc27342515)

[3.3. Hardware 4](#_Toc27342516)

[3.4. Schedule 4](#_Toc27342517)

[3.5. Similar Products 4](#_Toc27342518)

[3.6. Conclusion to Proposal 4](#_Toc27342519)

[4. Components Used 4](#_Toc27342520)

[4.1. Raspberry Pi 4 Model B 4](#_Toc27342521)

[4.2. IR Break Beam Sensor 5](#_Toc27342522)

[4.3. Red LED 5](#_Toc27342523)

[4.4. 1 Kilo-Ohm Resistor 6](#_Toc27342524)

[4.5. 4.7 Kilo-Ohm Pull Up Resistor 6](#_Toc27342525)

[5. Hardware Design 7](#_Toc27342526)

[6. Breadboard Testing 8](#_Toc27342527)

[7. PCB Design 9](#_Toc27342528)

[8. Enclosure Design 10](#_Toc27342529)

[9. Final Testing 10](#_Toc27342530)

[10. Troubleshooting 11](#_Toc27342531)

[11. Conclusion 11](#_Toc27342532)

# Abstract

This report will explain the IR Break Beam Sensor project. The purpose of this project is to show how the IR Break Beam sensor interacts with the Raspberry Pi. The IR break beam sensor comes in two components, one is the receiver and the other is the transmitter. Simply put, the transmitter sends the IR beam to receiver. The receiver outputs a digital signal based on whether it detects the infrared beam from the transmitter. The Raspberry Pi can read this digital signal from one of it GPIO pins and the value can then be displayed on the terminal. Components used in the project will be discussed as well as why they were chosen for this project. The proposal and the background of the project will be discussed in this report. The idea of the use for the IR break beam sensor came from the thought that it will be used in the final project. The IR break beam sensor will be used for the gate control on the final project. The first step for the hardware design is the breadboard setup. All the hardware designs were created in a program called Fritzing. The design process of the breadboard, schematic, and PCB will be mentioned in the report. Test results of all hardware designs will be shown in the report. Problems occurred in each of the hardware designs and were also troubleshooted. The troubleshooting procedures will be written down here in this report.

# Introduction to Report

The purpose of this project will be to show how the IR break beam sensor is used. This report will look into how the project overall will be effective in the parking lot final project. The function of the IR break beam sensor is to detect interference between the transmitter and the receiver. The first part of the sensor is known as the transmitter which sends an infrared beam to the second part of the sensor. The second part is the receiver which detects if it receives the infrared beam. If the receiver does not detect the infrared signal it will send a digital output signal through the output wire. Finally, this report will go over in depth about the IR break beam sensor and how it interacts with the Raspberry Pi and how it will fit into the larger project.

# Proposal

# Introduction and Background for Proposal

The name of the project is Park Smart, the people that are involved with this project are George Alexandris, Vikas Sharma, and Daniel O’Donnell. The project is supposed to be a SMART parking lot system that incorporates a phone app to manage a user’s tickets, account, and where to park in the parking lot. The idea of this project came up when the group realized that they can develop an easier way to find parking spots by connecting all the spots to a parking app that detects if there is spot open.

# Skills and Courses Related:

* Circuits from CENG 215 Digital and Interface Systems
* Gannt charts that plan out the project CENG 216 Intro to Software Engineering,
* Raspberry Pi and Micro computing from CENG 252 Embedded Systems,
* Dealing with SQL and databases from CENG 254 Database With Java,
* Web access of databases from CENG 256 Internet Scripting;

# Hardware

The hardware tested has been successful in the Fall 2019 semester. The IR break beam sensor will be used for the gate control. Vikas will set up proximity sensors for the parking spots to detect if a car has parked. Daniel will set up the camera for authentication of license plates and stepper motor for the gate to move up and down. The data will be stored on the Firebase database that was set up in the CENG 319 course. All the database information for an account will have the name of the person, balance, and parking spot and this information will be able to be sent and retrieved from the phone app.

# Schedule

The Android app met expectations just a few more things need to be implemented, and the hardware for the IR Break Beam sensor is completed. The project has met all the deadlines stated on the GANNT chart. All the problems have been resolved and the project has met the exceptional standards before implementing them into the final project next year.

# Similar Products

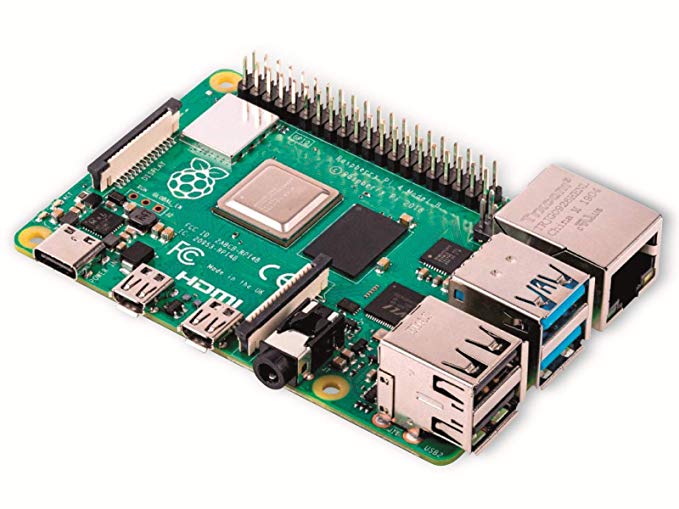
There are many parking lot systems that are similar to our product such as EasyPark. EasyPark is an app that allows the users, who purchase a monthly parking pass or daily parking pass, to park in the designated area EasyPark owns. This is similar to the software app that is being designed for the project. The software app for the project has a payment system that allows customers to get in a reserved space for them.

# Conclusion to Proposal

In conclusion, with the projects the group has completed, the group can build and implement the parking lot system on time. There will be an IoT solution by the end of next semester when the final project is started.

# Components Used

# Raspberry Pi 4 Model B



*Image of Raspberry Pi 4 Model B*

The choice of this Microcomputer was for the fact that it offers Wi-Fi and Bluetooth, and it aims to be a lot faster at computing than the previous models. The model for this project uses 4GB of RAM compared to two other models of this RPi 4 there were a 1GB RAM and 2GB RAM option. The 4GB option was chosen for more processing power. The Raspberry Pi also has 40 pins that have various uses. The pins on the Pi can power and ground a ciruit using 3.3V or 5V. There are also pins that are called General-Purpose Input/Output (GPIO). The GPIO pins can be set to either input or output depending on however the user sets it up. The 3.3V pin and GND pin on the Raspberry Pi are being used to power and ground the PCB designed for the IR break beam sensor. One of the many GPIO pins on the Raspberry Pi are set as an input to read the digital signal sent from the IR break beam sensor.

# IR Break Beam Sensor



*Image of IR Break Beam Sensor -- Transmitter on the left, has 2 wires (power, ground) and Receiver on the right, has 3 wires (power, ground and output)*

IR break beam sensor is two parts one transmitter and the other is the receiver. The two components can be powered by 3.3V or 5V. The transmitter has two wires, one for power and the other for ground. Transmitter sends an IR beam when it is powered up. The transmitter does not require any type of resistor to powered and can be connected directly to power and ground and it will work perfectly fine. When the transmitter is connected to 3.3V it offers a short distance of the travelling IR beam but when attached to 5v the transmitter can send an infrared beam up to a distance of 25cm. In the PCB designed for this project the transmitter is connected to 3.3V just to be safe of not damaging the component.

Receiver has three wires one for power, another for ground and one output wire. The power and output wires can be directly connected to a 3.3V or 5V it does not matter but requires the right amount of resistance. In the PCB designed for this project the receiver is connected to 3.3V just to be safe of not damaging the component. Output wire sends a digital signal through the wire when the receiver does not receive an infrared beam. The output wire requires a pull up resistor around 4.7kOhm for the digital signal to be detected by the Raspberry Pi. The power and ground wire, however, can be directly connected to 3.3V and ground directly.

# Red LED



*Image of Red LED*

This LED emits a red light when powered on, it requires a 1Kohm resistor to be lit up to control the voltage and current levels going into it. It is connected to the output wire from the IR break beam sensor and then to a 1kohm pull up resistor. When the output wire sends the digital signal, it turns on the LED. The LED is used to indicate if the beam is detected or not by the IR sensor.

# 1 Kilo-Ohm Resistor



*Image of 1k-ohm resistor*

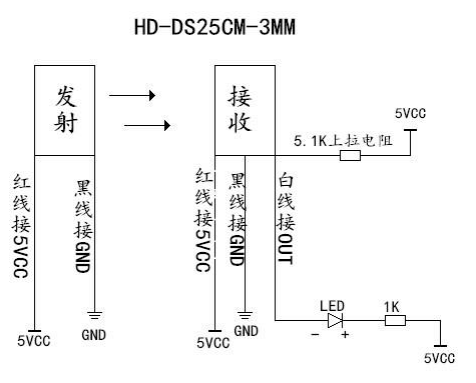
This is the pull up resistor for the Red LED. This will keep the LED from exceeding the highest current it can handle

# 4.7 Kilo-Ohm Pull Up Resistor



*Image of 4.7 Kilo-Ohm resistor*

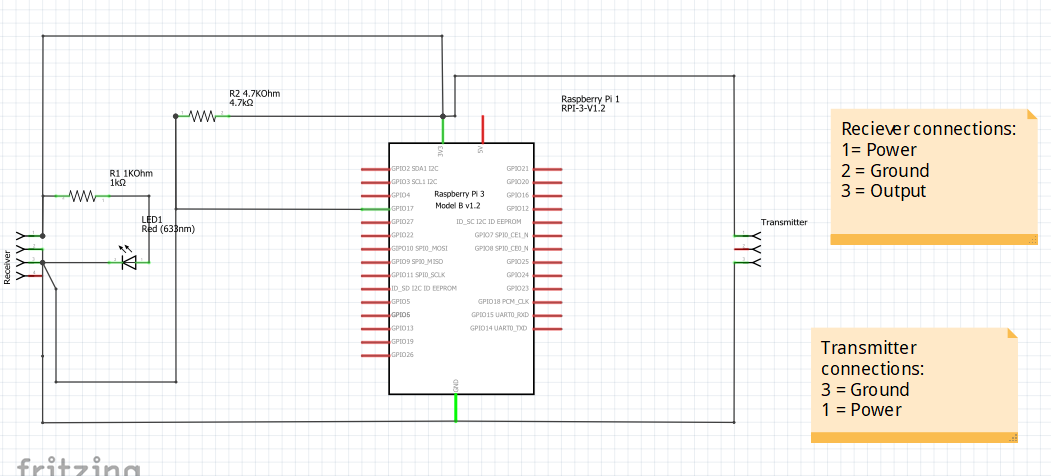
The 4.7Kohm resistor is needed in order to read the digital signal from the output. This was specified in the datasheet from the IR break beam sensor:



*Image from datasheet to show simple circuit set up*

Although it shows a 5.1 Kilo-Ohm resistor which are not in the parts kit provided. However, you can either use a 4.7 Kilo-Ohm which is used in this project or a 5.6 Kilo-Ohm resistor to handle the signal.

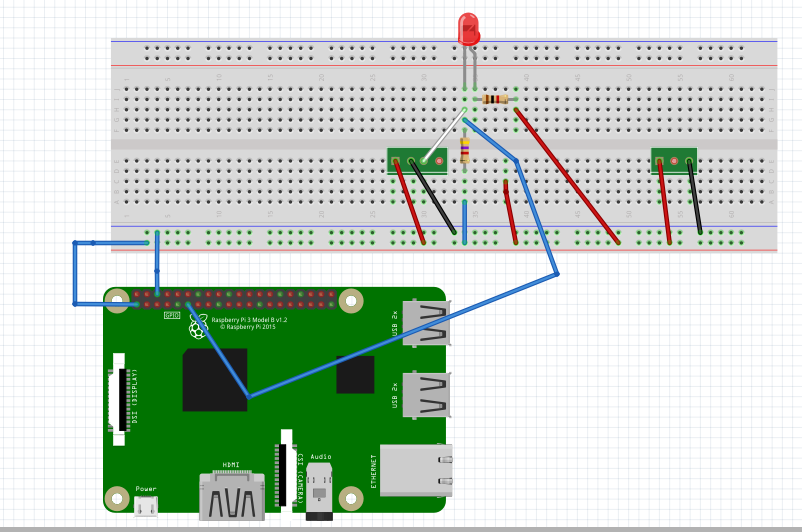
# Hardware Design



*Image of Schematic Design*

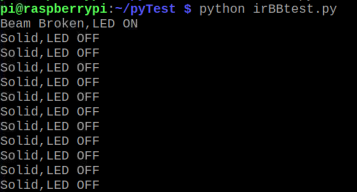
Hardware design of the IR break beam sensor project is shown here in the schematic design. The Raspberry Pi provides the power, ground and input for the full circuit. The transmitter just has ground and power directly connected to it. While on the receiver side, the power and ground wires are connected directly like the transmitter. The output wire on the receiver, however, has two resistors and an LED attached to it. The 4.7kOhm is directly connected to the 3.3V on the Raspberry Pi, and the LED is connected to the 1kOhm pull-up resistor. Then the last connection on the output wire it is connected directly to one of the GPIO pins of the Raspberry Pi 4, it can connect to any specific one. In this project I connected the output to the GPIO 17 pin on the Pi and set it up as an input. The GPIO pins on the Pi can either be set up as input or output depending on the code.

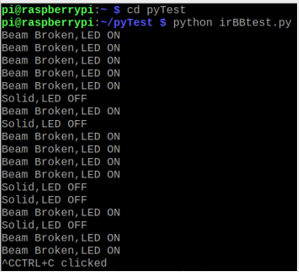
# Breadboard Testing



*Image of final breadboard design*

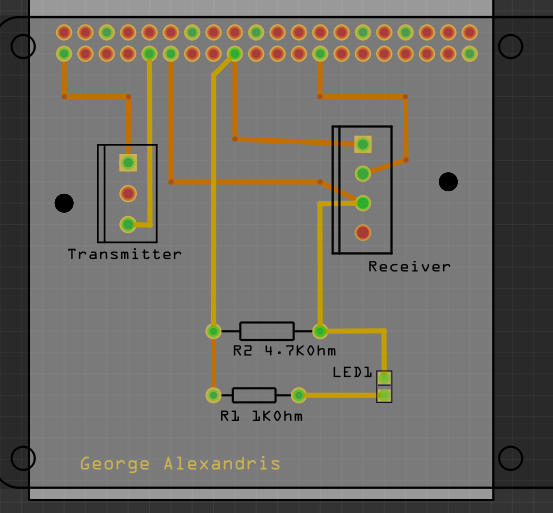
*Image of failed breadboard test*

Breadboard testing came from a few different designs, more generally from testing different designs specified from sources. Some were fails and some were generally fault in the code, second guessing the design of the circuit. The testing of the breadboard circuit came from multiple designs just to see if a digital signal can be read from the receiver to the Raspberry Pi’s GPIO pins. Some of the designs were faulty due to the code that was tested or some the pins were not reading the right signal. On the tests that failed the python code would send results showing that the pin was in LOW state and then would change to HIGH state and stay in the HIGH state forever.

*Image of successful breadboard test*

Due to the code being set to the wrong pin, the RPi was not reading the values from the correct pin. Once the code was set to the right GPIO pin the code was getting the correct values as shown to the left.

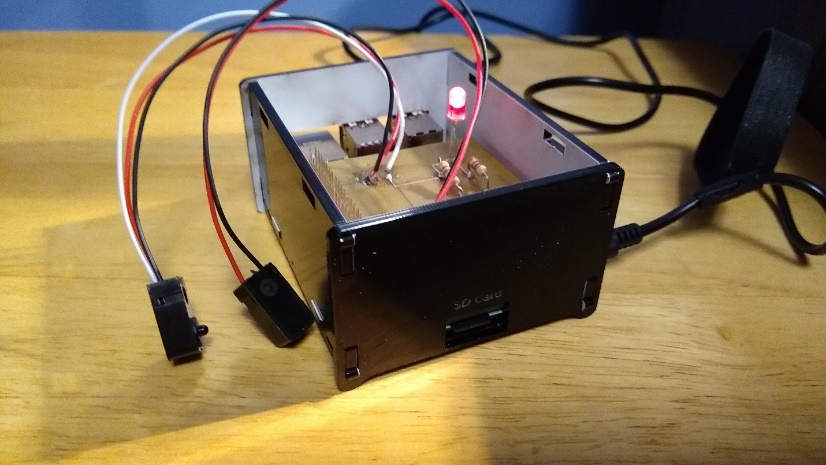
# PCB Design



*Image of PCB Design from Fritzing*

PCB designs were all created in a free software called Fritzing. All designs were done in the application from the breadboard design, schematic and then the PCB. The PCB was completed last in the design process. This is to make sure the breadboard tests go well and to see if there can anything being read from the sensor. After the breadboard design and testing is completed the PCB design needs to be finished. The final PCB design shown here above is made to represent something similar to the breadboard design. Transmitter is connected to both 3.3V power and ground, and the receiver is connected to 3.3V and ground directly, and then its output wire is connected to 4.7kohm pull up resistor, LED and then to another pull up resistor that is 1kohm. Then the third line from the output wire is connected to the GPIO pin 17.

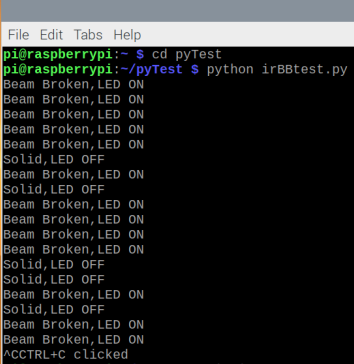
# Enclosure Design



*Image of PCB and Raspberry Pi in the enclosure*

Enclosure design went through a few design changes the first design for the Raspberry Pi did not fit because the part where the 4 USB ports and 1 ethernet port did not fit properly so there has to be a design change in the size of it. The first design was just for the Raspberry Pi and did not consider the PCB size. The second design for the enclosure did fit the Raspberry Pi but the PCB did not fit properly when put together. The third design had to consider the dimensions of the PCB and the Raspberry Pi fitting together. In the image above it shows the final design of the enclosure of the Raspberry Pi and PCB fitting together with the enclosure.

# Final Testing

The final tests on were performed on the Raspberry Pi. For example, the image on the right shows that result. The GPIO pin is setup as an input in the code and reads the output from the receiver. Receiver sends a digital signal to the GPIO pin. Then the program displays whether the value is high or low.

# Troubleshooting

Troubleshooting procedures that have been used in no particular order for hardware and software are going to be listed below.

For Hardware:

* Checking if the wires of the IR Break Beam Sensor are connected to the right places.
* Checking if the pins on the RPi are reading properly
* Checking if the output pin is connected properly to the GPIO pin
* Making sure there are no loose connections
* Making sure the sensor is not faulty by testing with an LED indicator.

For Software:

* Checking if the RPI.GPIO library is set up properly
* Checking if the pin is correct when setting up the GPIO in BCM mode
* Checking if pin is reading correct values

In the beginning of this project, there were some troubleshooting steps that were being missed. There were problems through different phases in the project, for example, not setting up the right pin in the code. In the project there has been use of one pin and knowledge has been gained on how it can be implemented into the project. Knowledge of how the IR break beam sensors has been gained as well.

# Conclusion

In conclusion, the report described how the IR break beam sensor works, how it was troubleshooted, noted any problems in software design or hardware design and the solutions. The purpose of this report (again) was to look at how the IR Break Beam Sensor can be used in a project. The topics described in this report will be used in the final project next semester in the final project for the parking lot system.