

## **Raspberry Pi Project – Home Alarm System**

Marcus Jeffrey and Simran Vig

### **Objective:**

The objective of this project is to use a Raspberry Pi 4 to create a home alarm system. It will include multiple components, including the use of hardware, software, a GUI, and Raspberry Pi's input and output functionalities. The user will be alerted of movement with email, a buzzer and LED lights when movement is detected in the area they want to secure. They will also have the option to disarm the alarm, or to put it on "away mode." The project may include a small model of a house made of cardboard to be used for the demonstration.

### **Hardware:**

Below is the list of hardware components used in this project.

- PIR Sensor
- Buzzer
- RGB LED
- Camera
- Wires (male-to-male and male-to-female)
- Resistors (220 ohm)
- Miniature House Model

### **Steps of Execution**

Below is a general plan of how this project should execute, and how a user might interact with it.

1. By default, the alarm system's "away mode" is off. This means that the PIR sensor is not currently sensing movement, and potential alerts are turned off.
2. When the user decides they want to set their alarm, they will use the GUI to set the alarm to "away mode."
3. When "away mode" is set using the GUI, the LED connected to the breadboard turns red, and the PIR sensor begins sensing nearby movement.
4. If the sensor detects movement, four things will happen:
  - a) The buzzer will go off in timed intervals.
  - b) The LED will flash red in timed intervals.
  - c) The camera, which is aimed to capture potential movement, takes a photo.
  - d) The photo is sent to the user via email, along with a warning message.
5. The user can deselect "away mode" on the GUI to disarm the alarm.
  - If the user deselects "away mode" while the alarm is on, the buzzer will stop emitting noise, and the LED will stop flashing. However, an email with a photo of the

movement will still be sent to the user as a cautionary measure – this could be helpful if they simply want to turn off the noise and flashing lights, but still want to see who made the movement.

## Circuit Design

Below is a draft for the circuit design of this project. It shows circuits that connect an LED, a buzzer, and a PIR sensor to the breadboard.

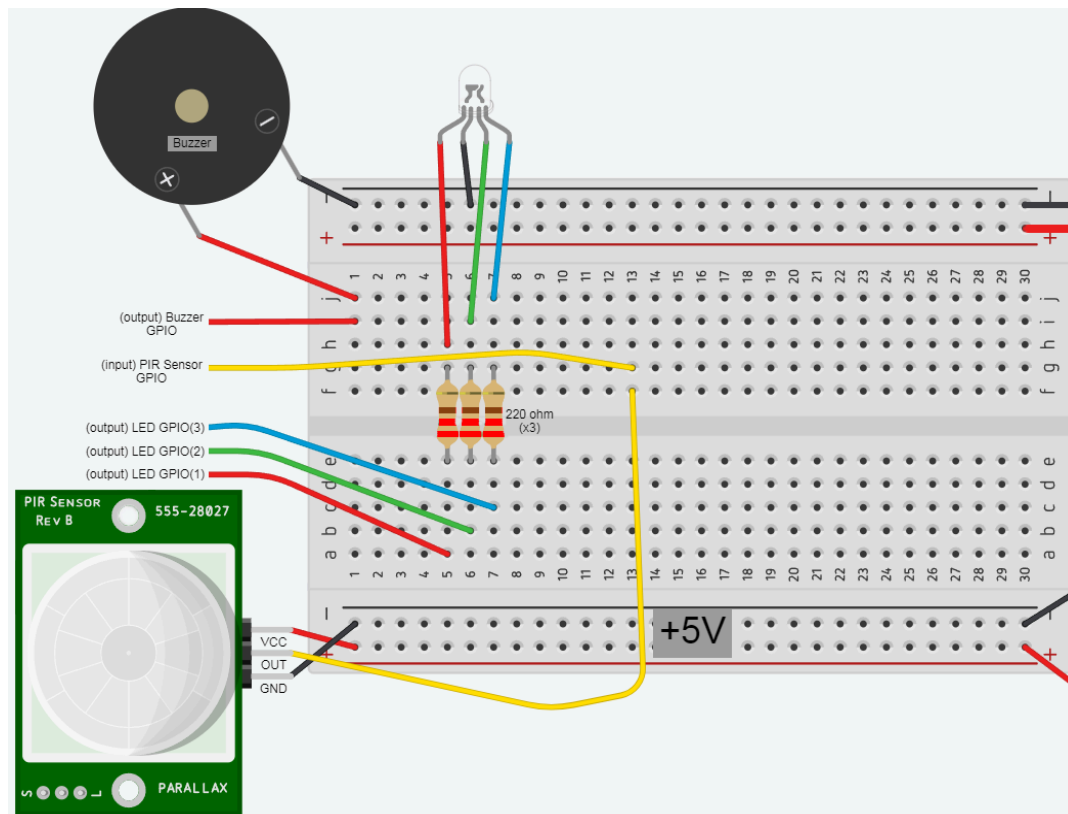


Figure 1: Diagram that shows the circuits for a PIR sensor, RGB LED, and buzzer, created using Tinkercad.com.

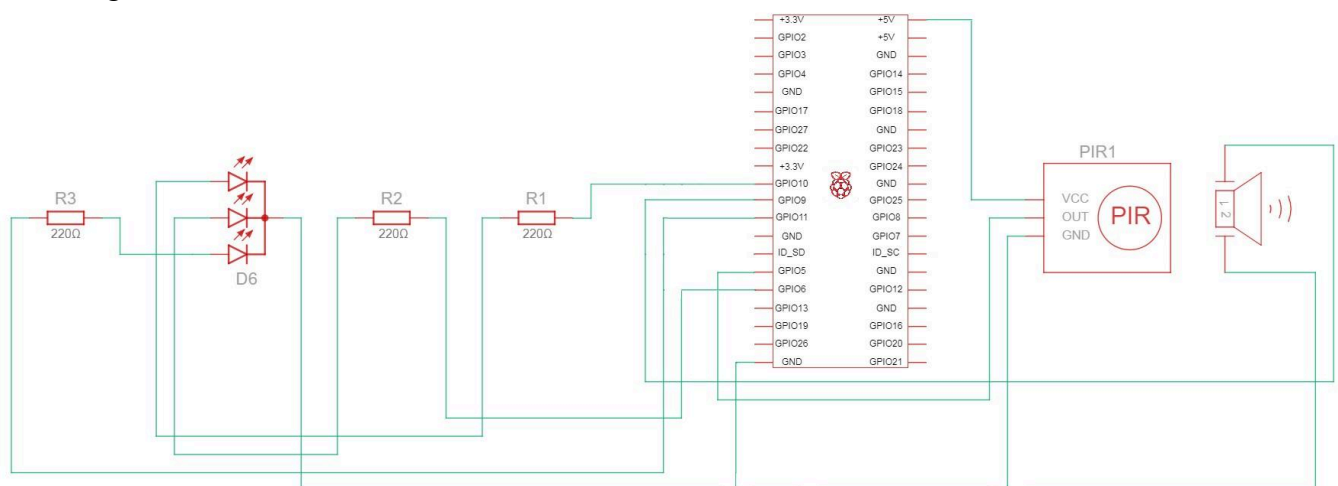


Figure 2: Circuit schematic of Figure 1 showing the connections to the Raspberry Pi GPIOs

## Software Plans

The software will have to control the LED, camera, PIR sensor, buzzer, GUI, and email. Each part of the code will be completed separately, and then integrated to ensure that the components are activated at the correct times. They will all make use of the GPIO and time libraries.

Below are general plans for how the code for each component will be implemented.

### a) **LED:**

The code will connect the LED's output pins and cause the LED to output green light when "away mode" is off. When "away mode" is turned on, the LED will consistently output red light. Then, when movement is detected, the LED will flash red light in set intervals.

Different functions will be created for each scenario, and then these functions will be called at the appropriate times.

### b) **PIR Sensor:**

The code will connect to the PIR sensor's output pin. If "away mode" is turned on, the code will start listening to the PIR's output. If enough motion is detected while the code's in "away mode" it will enter the alarm state. This might be achieved using a function. Deselecting away mode might also require another function to be called to turn off the buzzer and LED.

### c) **Buzzer:**

The code will connect the buzzer to output and will output in accordance with the PIR sensor's input pin. When the sensor detects movement while the alarm is on "away mode," the buzzer should go off in intervals. When the user deselects "away mode," the buzzer will stop emitting sound.

### d) **Camera:**

Using the picamera library, the code will use the capture method to take a photo and save it when the PIR sensor senses movement.

### e) **Email:**

Using the yagmail library, the code will read the content of the file with the captured photo (with the attachment parameter of the send method) into an email alongside a warning message (using the contents parameter of the send method).

### f) **GUI:**

The project will use a simple GUI with a checkbox, and an "Enter" button to submit the state of the checkbox. The checkbox will allow the user to select or deselect the

alarm's "away mode." Once they press the enter button, their preferences will be set until they change them.

Once each area of code is done, the next step will be to integrate all pieces so that they function according to the steps of execution. Generally, the GUI should lead to the buzzer, camera, LED, and email functioning simultaneously. The LED light will be the only hardware component that is consistently outputting.

### **Schedule and Implementation Stages**

After submitting the project's proposal on March 15<sup>th</sup>, the following is a tentative schedule for when the project will be implemented.

**March 15<sup>th</sup>:** Test the circuits with basic programs to ensure that they connect to the hardware components properly.

**March 15<sup>th</sup>- 21<sup>st</sup>:** Write the code for the project. Make sure it is comprehensive enough for testing the next session. Begin drafting or gathering materials for the model house.

**March 22<sup>nd</sup>:** Test the code along with the circuits from the previous weeks. Adjust the code or circuits if necessary. Begin building the model house and observing how it would work with the project

**March 22<sup>nd</sup>-28<sup>th</sup>:** Draft the project's report, add details to project or fix errors. Fix any errors with the model house.

**March 29<sup>th</sup>:** Practice running the full project, from start to finish, in preparation for the demo.

**March 29<sup>th</sup> - April 4<sup>th</sup>:** Continue to write the final report, editing the draft and adjusting any outdated information.

**April 5<sup>th</sup>:** Demonstrate the project in front of class.

**April 5<sup>th</sup>-12<sup>th</sup>:** Finish the report. Include any thoughts from the class demonstration, answers to questions that may have been asked, and notes for improvement for others who want to replicate the project.

### **Sources:**

Li, Link. "Introducing Electronics." Designing with Raspberry Pi, Jan 19, 2024, Douglas College. Lecture.

Li, Link. "Build Circuit & Control GPIOs." Designing with Raspberry Pi, Jan 26, 2024, Douglas College. Lecture.

Li, Link. "Sensors." Designing with Raspberry Pi, Feb 9, 2024, Douglas College. Lecture.

Li, Link. "Email, Camera, File." Designing with Raspberry Pi, March 8, 2024, Douglas College. Lecture.