Senior Design
ENG EC 463

Memo

UPR

To: Professor Pisano

From: UPR Level Security: Juan Peralta, Eugenia Almandoz, Namir Fawaz, Brock Guzman, Brandon Webster

Team: 24

Date: 11/19/17

Subject: First Deliverable Test Report

1.0 Project Objective

1.1: The objective of the UPR Security Patrol Robot is to provide a safer means of detection throughout an electrical subsystem. In these high voltage situations, many dangerous faults could injure workers during routine checks. The Patrol Robot would not replace workers, but allow for safer, more consistent surveillance to detect errors before they become drastic problems.

1.2: An important aspect of the Patrol Robot is its ability to navigate the grid autonomously and avoid obstacles that may have fallen in its path. The Robot will make rounds multiple times a day to check for intruders, check gauges, and survey for faults.

1.3: The Patrol Robot will provide a security aspect that detects unauthorized intruders with a thermal camera. The camera will measure temperature and size of the object to differentiate between small animals and the large ones, including humans, that could cause damage to the system.

1.4: The robot will also provide an early fault detection for the subsystem to alert remote operators of small faults that could lead to a larger issue, including hot spots, partial discharges, and even arching. It will also provide measurement readings of the gauges at scheduled times throughout the day and relay these to the operator.

1.5: The Patrol Robot will communicate with an operator through a web app that streams live video, displays alerts, and allows for remote override of the robot. Override will include navigational controls and shut down.

2.0 Test Objective and Significance

2.1: For our initial test we implemented a web application that serves as the central point for the user of our robot. Our application had 3 main components: a live video feed, an alerts bar and then the hardware aspect of our deliverable. The live video is produced using a raspberry pi camera which we plan to expand to do image detection for objects

as well. The alerts bar updates based on our sensor detections and whether the object is close enough to populate an alert. Lastly, for the hardware we used an arduino, a raspberry pi and three ultrasonic sensors.

2.2: This deliverable is crucial for our final project because of the importance of updating the user of important events the robot detects or to be able to have the user communicate with the robot itself. Although, having a robot does not replace employee's jobs it is a way for them to monitor the substation in the absence of personnel. In order for the robot to do its job efficiently we must have a way to communicate important events and video back to the employees. This application will also serve as the basis for users to override the robot and control it from their screen. They will be able to navigate the robot and drive it to locations they want to further investigate. This feature will be implemented later once the robot is able to navigate around a certain place/area,

2.3 **Live Video**

2.3.1 The live video is important because it helps the user see the area as if he/she were walking through it themselves. It serves as a security check factor through the user's eyes in case the robot is not able to spot every fault or danger in the substation. The live video is also important for when the user takes control of the robot. In order to see where the robot is going, the user can simply maneuver it based on what it sees through this live video.

2.4 **Alerts Bar**

2.4.1 The alert bar is significant because it not only helps the user but also is a self check for the group. In order for the robot to operate correctly it must detect certain objects as mentioned above. Documenting these faults the robot picks up is crucial for the company to find information quickly and efficiently, as well as be able to locate the spot the fault was detected. It is also a check for the group to assure that our robot is detecting all objects necessary and not missing any big events.

3.0 Equipment and Setup

3.1 Arduino

3.1.1 Ultrasonic Sensors: Three ultrasonic sensors were placed on a box which acted as the future robot. One was placed in front of the box and the other two placed were placed to the left and right of the box. A built-in arduino function calculates the time is takes for the sound wave to travel from the trigger to an object and reflect back to the echo of the sensor. To calculate the distance of an object, we just use equation: $d = v * t/2.$ Where d is distance, v is velocity, and t is time. We divide by two since the time recorded is the time to get to object and back. Knowing the distance we can calculate when an object is within 10 cm and send that information to the serial port.

3.1.2 Data Transfer: The information gathered from the Arduino is sent to the Raspberry Pi, which is then sent to the alert feed of the web application. Since the Arduino sends its information to its serial port, we can just plug in the Raspberry Pi to that port. On the Raspberry Pi we look for what port is connected to the Arduino and we tell it to read in that data.

3.2 Raspberry Pi

3.2.1 Camera Module: Since the camera we got was the Raspberry Pi's NoIR camera, integrating it was fairly simple. The camera is able to take both video and picture in 1080p. The camera is connected to the Pi's camera port, and its configured on the Pi.

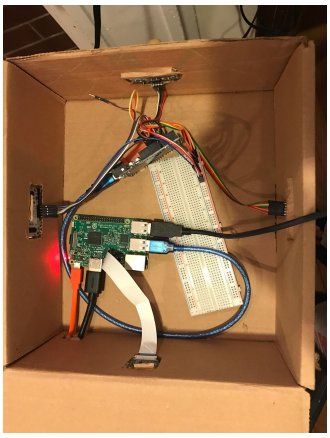

Figure 1. This picture shows the box we used to represent our robot and test our equipment. There is three sensors, located on the front, right, and left side of the box. The camera is located on the back of the box.

3.3 Web Application

3.3.1 Web Application: The web application is created using a combination of HTML and flask. For storing data we implemented a SQLAlchemy database in order to store data received from Raspberry Pi. The website, for now, has two main functions, to display live video from robot, and receive alerts from the Raspberry Pi.

3.3.2 Video Feed: This part of the test plan integrates the video stream on the Raspberry Pi with the web application. After the camera has been configured with the Pi, we use a mjpg streamer to send the video to a specified ip address. The web application then uses istream to play video from that ip address.

3.3.3 Alert Feed: After the Arduino sends sensor data to the Raspberry Pi, we need to send this information to our website's Alert Feed. In order to do so, the Raspberry Pi creates a JSON Post request which is sent to the server at (server_ip_address):5000/api/alert. The API then receives this request and decodes the JSON into an entry on the database. The website can now take information from the database and populate the Alert Feed. The alerts

have a title specifying what kind of alert it is, and then a description of the alert. For now it is only for object detection.

Figure 2. Screenshot of our website. The video would appear on the left side, and the alert feed populates on the right.

4.0 Measurements and Data

4.1 Summary:

The three main testable pieces of our deliverable, the live video feed, alerts, and hardware components were tested on a pass/fail basis. Successful viewing of the video feed and alerts in the web application guaranteed that the test was passing on all levels of the implementation. All components of our deliverable were able to pass their pass/fail test.

4.2 Video Feed:

The video feed presents in real time from the Raspberry Pi to the web application. This passes the test that we set up to ensure that the video feed works, proving that the Raspberry Pi is transmitting video properly, and the web application is receiving and displaying the video.

4.3 Alerts:

When an object is inserted within 10 cm of an ultrasonic sensor, the web application will display an alert on the web page (after the page is refreshed). This demonstrates alert functionality on every level of our web app: the ultrasonic sensors, the Arduino, the Raspberry Pi, and the web application and its database. If any piece of hardware throughout this hierarchy was not working, the end server would not display the alert that should be created.

4.4 Hardware:

Due to the passage of the prior two tests, all hardware components demonstrate that they are functional for the deliverable, and pass this test of functionality.

5.0 Conclusions

5.1 Summary:

There were three main purposes for this first deliverable. The first goal of the deliverable was to build a webapp which could be used to remotely monitor, and eventually control the UPR. The second goal was to demonstrate the Raspberry Pi's ability to stream live video to the web app that we built. Finally, the third objective was to demonstrate the Arduino's ability to receive a signal from an Ultrasonic Sensor then send the signal to the Raspberry Pi which would in turn send an alert to our web app in real time.

5.2 Web App:

These goals were all met, and the work that we were able to accomplish as a team gave us a strong platform to build on. The web app has been architected effectively to be able to store all alerts in a database along with a timestamp, so that an operator would be able to keep track of every anomaly detected. With this base layer completed, we will be able to develop a very powerful tool that will be easy to setup and use for any remote operator.

5.3 Video Streaming:

We were able to effectively stream live video from the Raspberry Pi to our web app, but this portion of the project will still require a decent amount of work. The camera we used to stream the video was the built in Raspberry Pi camera, but this will most likely not be the camera we decide to use in the final project. Though it has solid quality, it is not a camera that is very easy to use because it is attached by a very short and thin wire to the Pi itself. We will need to find a way to connect a separate camera to the Raspberry Pi and still transmit the video to our app.

This leads into the next point, which is to change the way we are currently streaming the video. As of now, the Pi uploads the stream directly to an unprotected IP address, which the app streams from directly. This is a very insecure way of streaming the video, and with the UPR security will need to be a priority. We will need to find a way to update this process so that the live video will be transmitted in a much more secure way.

5.4 Alert System:

The alerts system worked very well for our first deliverable. We were easily able to translate signals from the Ultrasonic Sensors into alerts on our app. The next step will be connecting different types of sensors using them to again send alerts into the Alerts List in our app.