ECE 4564 - Team 20

Ethan Brooks
ethanb98@vt.edu

Chris Honaker
chris98@vt.edu

Kishan Parikh
kjp2376@vt.edu

Final Project Report

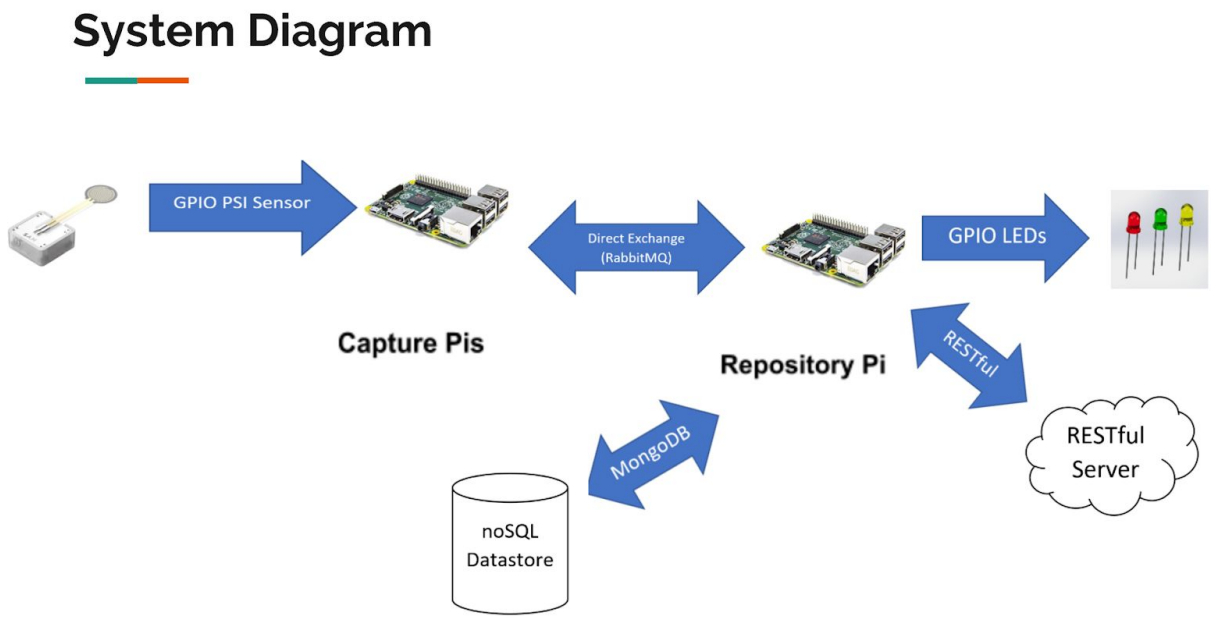
Concept of operations

The main purpose of our final project is to give students a way to know if tables are open at the 2nd floor of the library, the 4th floor of the library, and Torgersen Bridge. This is a major problem for many students at Virginia Tech that may not know if there are openings in popular study spaces. This would save people time and effort of looking for places to go study. The goal is to give students real-time updates of openings at all of these locations.

Students will interact with the system without even realizing, simply by placing books and items onto a table. The weight of these items will cause the pressure sensor on the table to update that the table is currently occupied. Once the items have been placed on the tables, the capture raspberry pi measuring the pressure pad voltage will receive a high signal, sending this data to the repository raspberry pi.

The repository raspberry pi will then turn the LED attached to it different colors based on availability of the study spaces. All study spaces being free will result in a green LED and all study spaces taken will result in a red LED. The user is also able to access a RESTful web server with three subject titled “4th Lib”, “2nd Lib”, or “Torg Bridge”, allowing them to observe location information. If the library is all full, the user will see that and will go to Torgersen Bridge which has more availability instead.

System Overview



ECE 4564 - Team 20

Ethan Brooks
ethanb98@vt.edu

Chris Honaker
chris98@vt.edu

Kishan Parikh
kjp2376@vt.edu

Description of system modules

1. Step one in our project is to connect the pressure sensors to the two capture raspberry pi's to measure if a textbook has been placed on it or not. The pressure sensors range from nearly 10M□ with no force applied, to around 1k□ with a good amount of force. The raspberry pi will be measuring the voltage coming in from the pressure sensors, to decide whether to activate signal high saying the table is being used or not.
2. Step two is to deliver the information read in from the pressure pads on the Capture Pi to the Repository Pi using RabbitMQ.
3. Step three is to send the data to MongoDB for storage, where 1 warehouse houses two collections, one for table availability received from the RabbitMQ exchange, and one keeping track of hokie PIDs and passwords for user authentication.
4. Step four is to connect the Repository Pi to an RGB LED to output availability of the tables. Green if the availability of tables is open and red if availability taken.
5. Step five is to connect the Repository Pi to a Flask server using RESTful API, to create a URL where the user can look up availability at three different locations, Library 4th floor, Library 2nd floor, and Torgersen Hall. The RESTful server will display the color of the LED at each location, green for spots open, red for all tables taken.

4 Testable Requirements

1. Upon a student laying a textbook onto the pressure sensor, the server will be updated that the table is occupied.
2. The system will have GPIO LEDs that will display red for all tables are occupied, and green for tables being free.
3. The raspberry pis will communicate using RabbitMQ.
4. Have a Flask server server with three locations titled "4th Lib", "2nd Lib", or "Torg Bridge" for users to observe the location information using the RESTful API.

ECE 4564 - Team 20

Ethan Brooks
ethanb98@vt.edu

Chris Honaker
chris98@vt.edu

Kishan Parikh
kjp2376@vt.edu

Hardware List (to be bought)

Part	Quantity	Price per Part	Total	URL
Pressure Sensor	1	$7.95 (+$5 S&H)	$12.95	https://www.amazon.com/SENSING-RESISTOR-CIRCLE-1oz-22LB-FLEXIBLE/dp/B00B887CLS/ref=sr_1_1?keywords=Flexiforce&linkCode=ll2&linkId=aef56fddd05dff109203c5fe70d84ec2&qid=1553374514&s=gateway&sr=8-1
GPIO Breakout Boards	**1**	$9.99	$9.99	https://www.amazon.com/iUniker-GPIO-Breakout-Expansion-Raspberry/dp/B07DR9TTDP/ref=sr_1_8?keywords=raspberry+pi+gpio+breadboard&qid=1553374810&s=gateway&sr=8-8
Total	**2**		**$22.95**	

Hardware List (already owned)

Part	Quantity
Raspberry Pi	3
Monitor	3
GPIO LED	1
GPIO Breakout Board	1
Library Table	1
Textbook	1
Resistor and Capacitor Kit	1

ECE 4564 - Team 20

Ethan Brooks
ethanb98@vt.edu

Chris Honaker
chris98@vt.edu

Kishan Parikh
kjp2376@vt.edu

Project Schedule

	3/24	3/31	4/24	4/25	4/26	4/27	4/28	5/4	5/7
Proposal Submitted									
Project Approval									
Create Presentation									
Project Proposal Presentation									
LED sending green/red light display based on input									
PSI Pads sending high/low values from weight									
Notice Sent to RPi changing LED light									
RabbitMQ Tx/Rx data from PSI RPis to LED RPi									
Debugging rest of code									
Make final presentation									
Final Project Demonstration									

Team Member Tasks

Chris: RabbitMQ connections of the raspberry pi
Ethan: RESTful server connection, Electrical components (LED and PSI pads)
Kishan: MongoDB server setup

Project Reflections

The PSI Sensor is a variable resistor that emits a resistance of 10MΩ when there is no weight applied to it, and emits a resistance of 1kΩ when a textbook is placed on top of it. Using a simple voltage divider, and adding a 1MΩ resistor as R2, the formula $Vout = Vin * \frac{R2}{R1+R2}$ proves that under a 1kΩ strain provided by a textbook, 3.29V is input to the GPIO pin, more than enough for it to read a high value. Once read as a high voltage, a signal is sent to the RabbitMQ portion to be sent from the Capture Pi to the Repository Pi.

The RabbitMQ portion of the code serves as a way for the two Raspberry Pis to communicate with one another. Our RabbitMQ code has 1 exchange called Places. This exchange has 3 queues, 4th floor Library, 2nd floor Library, and Torgersen Bridge. Places exchange serves as a way for table availability at each location to be accurately

ECE 4564 - Team 20

Ethan Brooks ethanb98@vt.edu
Chris Honaker chris98@vt.edu
Kishan Parikh kjp2376@vt.edu

documented and reported. All of the exchanges were direct exchanged in RabbitMQ. Also, all of the queues were given a routing key that is the same as the name of the queue. This allows every queue to have a unique key so that every message goes to the right place. Each queue would relay its data to the necessary components, such as the RGB LED and the Flask Server.

The RGB LED that we received is directly connected to the board using the Raspberry Pi's GPIO pins. The RGB LED is connected to red, ground, green, blue pins, and each of the color pins has a resistor attached to it to ensure current is not high for the Raspberry Pi to handle. On the Pi, the program created multiple functions to turn on and turn off the red and green LED lights. Once a study table is read by the PSI sensor as being used, the LED will output a red light. Otherwise, if the table is open, the LED will output a green light, showing the availability.

MongoDB is used to store commands received from the RabbitMQ exchange. These commands are parsed and stored into a collection contained within one warehouse. The warehouse houses two collections that keep track of study table availability and user authentication. The commands are also parsed into document format and then stored into their respective database. The mongoDB instance is not started by the python code, but instead it is run as a service on the pi. This allows the server to automatically be started when the pi turns on.

The Flask server is created using the RESTful API, which allows the user to logon, given that they are a Hokie, through user authentication via the MongoDB server. The URL that the user sends must end in /4thLib, /2ndLib, or /Torg. Once logged onto the Flask server, the user will be presented with information regarding the availability of tables at each of these locations. If all are taken, the color field will display red, signaling that now is not a good time to go to that location. If the majority of table are free, the color field will display green, signaling that the location being looked at has many tables available.