[Group 11] Assignment #3: Hands-on Experience with MQTT

1. TEAM MEMBER DETAILS:

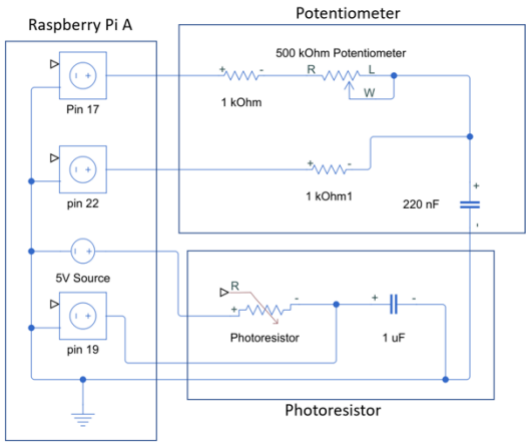
1) Priyam Garg pgarg6@ncsu.edu
2) Divyang Doshi ddoshi2@ncsu.edu
3) Brendan Driscoll bhdrisco@ncsu.edu
4) Jordan Boerger jwboerge@ncsu.edu
5) Vishal Veera Reddy vveerar2@ncsu.edu

Percent Contribution	
Priyam Garg	20%
Divyang Doshi	20%
Brendan Driscoll	20%
Jordan Boerger	20%
Vishal Veera Reddy	20%

Tasks		Members				
Topic	**Sub Tasks**	**Priyam Garg**	**Divyang Doshi**	**Brendan Driscoll**	**Jordan Boerger**	**Vishal Veera Reddy**
Hardware	Raspberry Pi A Hardware Connectivity and Software Integration of Hardware			100%		
	Raspberry Pi B Hardware Connectivity and Software Integration of Hardware					100%
Software	Raspberry Pi A MQTT Code				100%	
	Raspberry Pi B MQTT Code	100%				
	Raspberry Pi C MQTT Code		100%			
	Laptop #2 MQTT Code	100%				
	Laptop #1 requirements		100%			
Report/README		20%	20%	20%	20%	20%

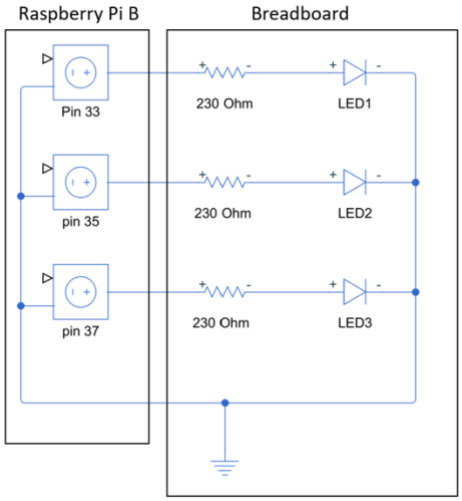
We believed that every teammate did their part well and had the equal contribution in the project.

2. SCHEMATICS FOR RASPBERRY PI A



Made in simulink.

3. SCHEMATICS FOR RASPBERRY PI B



Made in simulink.

4. DESIGN CHOICES

4.1. Choice 1 (MQTT Broker)

Our team chose the Mosquitto MQTT broker.
This broker is a popular choice for developing IoT applications and comes with a wealth of documentation.

- Mosquitto is simple to integrate into the Python environment in which our code is written.
- It also provides a powerful and easy-to-use API that allows us to simply specify callback routines for on_connect, on_disconnect, and on_message events.
- Mosquitto additionally has a loop mechanism that, when invoked, manages message publishing and subscribing to the broker while the user simply needs to indicate the topic for publishing and subscribing.

4.2 Choice 2 (ADC and Design)

For our implementation, we didn't use an ADC. Instead, we measured the time it took for capacitors to either charge or discharge by monitoring how long it takes for the raspberry pi input pins to change from high to low or vice versa. This works since the charge time for a capacitor in an RC circuit is dependent on the resistance, and the higher the resistance the longer it will take for the capacitor to charge.

Because of this implementation there is no set sampling rate since our measurements are time dependent, and each reading will take a different amount of time. As such we sampled these values every 50 ms to ensure that we fall within the 100 ms sampling time stated in the assignment requirements.

4.3 Choice 3 (Normalization)

We chose to normalize our readings for the LDR and potentiometer to range from 0-100, with the idea of making it a percentage of the max value for either component.

4.4 Choice 4 (range of raw values (min and max) for LDR and Potentiometer

The raw values observed front the potentiometer ranged from roughly 20 - 12000.
The raw values from the LDR varied much more, with a minimum of roughly 20 when a bright light was shined directly on it, resting in a well lit room it gave values of roughly 150, and when it was completely covered it was observed to go as high as 40000.

4.5 Choice 5 (Scaled Values)

Since we scaled our readings to represent a percentage of the maximum raw values, both the LDR and potentiometer range from 0-100.

5. LINKS

Github:
https://github.ncsu.edu/jwboerge/IoT_ASN3_G11

Google Drive:
https://drive.google.com/drive/folders/1cKj2L7G8Qy4bCCIyMD_sDCp5d-ssWQzH?usp=sharing