SAFE PET SYSTEM – PROJECT ARTEFACT

Thomas Morgan - 221392134

o Background

This prototype fundamentally combines pet surveillance and feeding devices into one smart home system.

One device is fixed below a pet door/flap and uses its RFID sensor to scan a tag that is attached to a pet's collar. This door device publishes particle cloud data to the feeder device to update the universal "state" of the pet. (Inside or outside). The pet's state is visually represented by one of two coloured LEDs on the door device.

The feeder device also incorporates RFID, scanning the specific tag of a pet to operate its feeding mechanism. The feeding mechanism operates based off the current time, and the scheduling that the user has input into the device. The scheduling is set by operating the button, when pressed, actuates a single LED on the LED ring. Each LED on the ring represent 1 hour of a 12-hour format. If the pet approaches the feeder RFID, and the current time meets the user set schedule, the feeder will disperse pet food into the bowl, and lift the lid allowing the pet to eat.

When any behaviour is observed by any device (goes inside/outside, attempts to eat or is eating), the system will notify the pet owner via IFTTT, where the owner will receive an email notification updating the pet's behaviour.

o Problem Statement

This system is developed to resolve a set of specific problems:

- An owner is away from the home and wishes to be notified on the pet's whereabouts and keep their pet fed while they are away.
- Owners of pets want an automated feeding schedule allowing the owner to limit the amount of food a pet can eat without supervision.
- Delegating a feeding device to a specific pet allows for un-intended pets/other animals to eat food meant for the delegated pet.
- Owners want to be updated in real-time on their pet's behaviour via online notifications.

o Requirements

Hardware:

- 2x Particle Argon
- Addressable LED Ring - 12 Bit WS2812 RGB LED
- SER0047 - 9g 180 Metal Servo with Analog Feedback (1.5kg)
- POLOLU-2820 - FEETECH FS90R Micro Continuous Rotation Servo
- 2x COM-09190 - Momentary Push Button Switch - 12mm Square
- ADA1536 - Buzzer 5V - Breadboard friendly
- LED Rainbow Pack - 5mm PTH
- 2x MFRC522 - RFID Read and Write Module Kit
- 3x 220 Resistors

Software:

- Particle Cloud user account
- Particle Workbench extension for VS Code
- IFTTT user account
- BOTH Safe pet Door system **and** Safe pet Feeder system files imported from GitHub repo, separate project for door device with src/libs for the door system, feeder device src/libs for the feeder device.

- Design Principles

This prototype utilises a component-based approach. The software is Modular programmed, where each sensor, actuator and data transmission are activated via a specific method call. The main objective of this project is to develop the most robust, fault-tolerant system applicable that utilizes a Real-Time application.

Considering that each device communicates with each other via Particle Publish() and Subscribe(), the intent of this design is to ensure complete communication/actuator/sensor fault-tolerance. System State is semi-dependant on Particle Cloud data, If the data transmission does not reach either device due to an acute internet drop out, the system has redundancy plans built in through hardware buttons and secondary function calls that enable Particle cloud data transmission when desired.

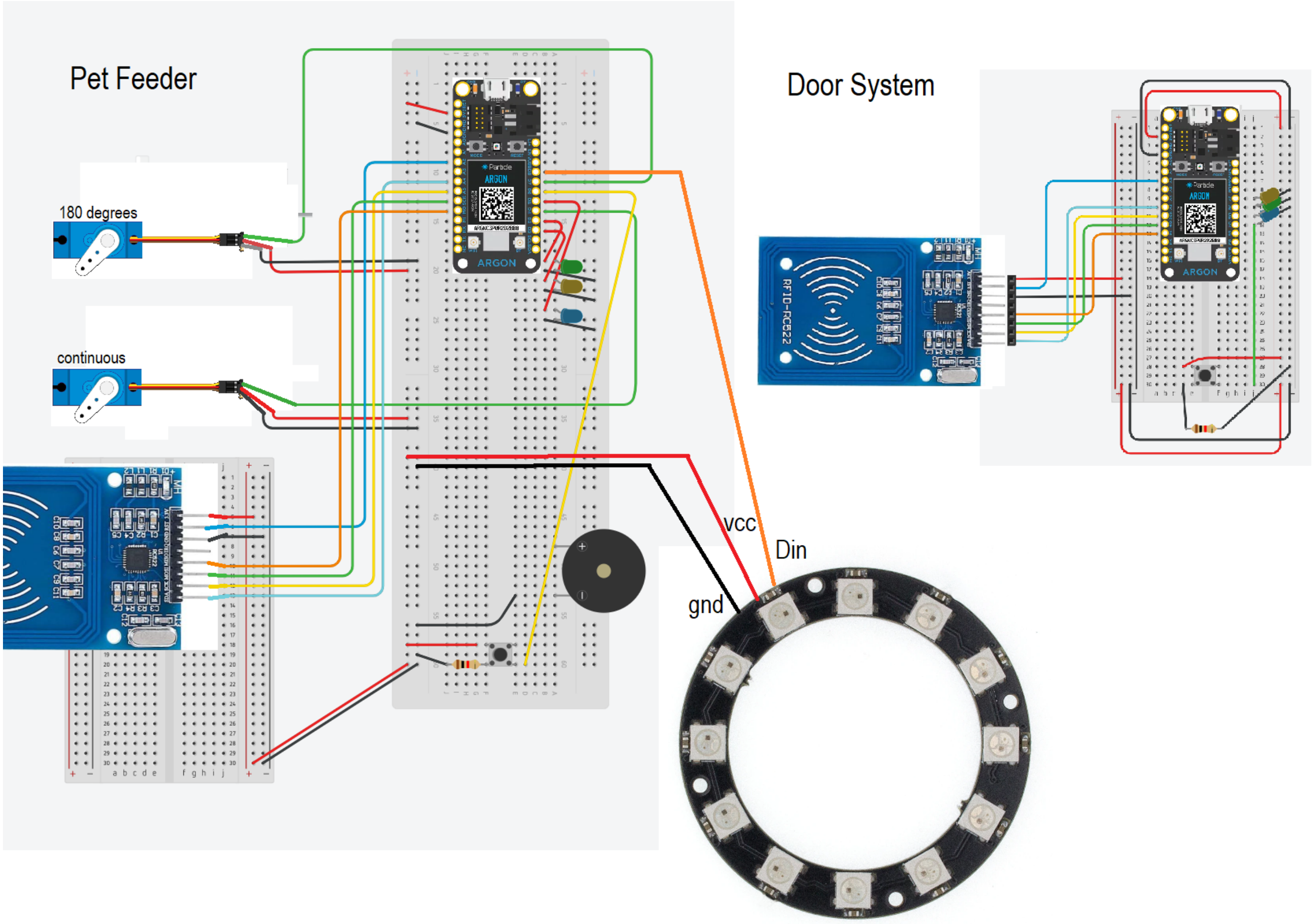
The device must also be responsive, in such the delay between sensing, thinking and acting must be at a minimum. This also applies to IFTTT notifications, where for the example, the feeder senses a pet, actuates the feeder and send a notification to the owner within a reasonable time frame (network speed permitted).

The fundamental point of this system is simplicity:

- Visual aids are used to represent to the user what the systems current state is in a clear objective manner.
- Buttons serve a single purpose that is well defined.
- The system operates autonomously without the need for user input outside initial setup.

The system is designed to incorporate real-time, using current AEST to compare to a set feeding schedule. The feeding schedule is set by a robust and simple button press that can be changed at anytime by the user.

- Prototype Architecture



- Link to prototype code on GitHub

https://github.com/Thumbness/SafePetHomeSystem.git

- Link to Project teaching Case video:

https://youtu.be/48-fUNYdTlw

Video Demonstration:

https://www.youtube.com/watch?v=66AhdNspRok

- Testing approach

In the wiring for the devices, I tested each piece of sensor/actuator hardware first, ensuring that the hardware communicates with the Argon, using the Serial for reading Card UID values, LED status, button presses etc. Once all hardware was initialised, I began to test the RFID functionality in storing the cards UID on the system then sending the UID to the door device via Particle Cloud. I designed the system such that both LED's would stop flashing on both devices if the UID matched upon initial "registration" of the tag/card.

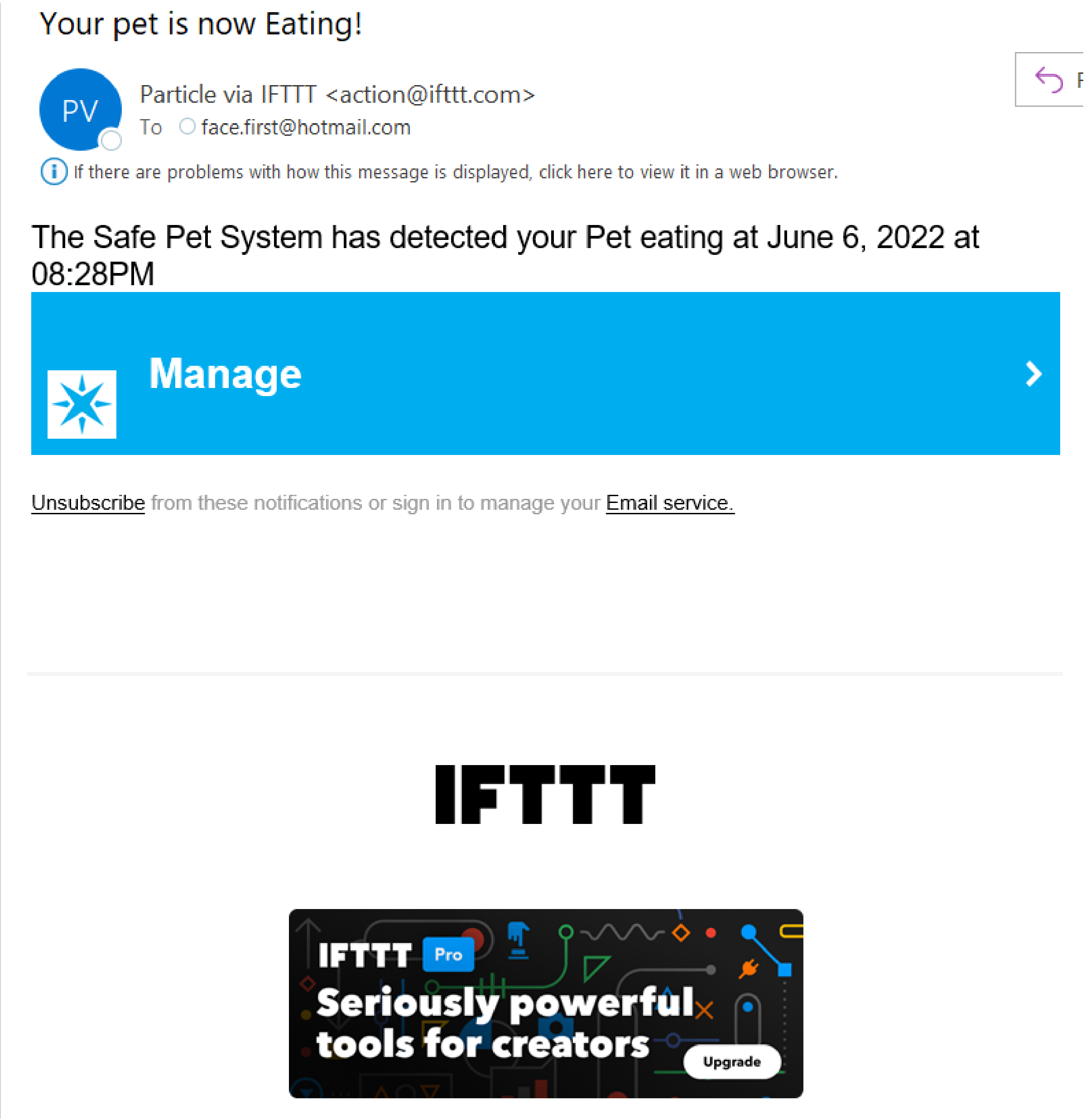
I tested the servo motors next, by initially using a button press to actuate each motor, setting the right angle and speed for the appropriate behaviour. Once the motor functionality had been set, instead of the button press, I used the tag/card and scanned it on the RFID sensor, given the previous testing of registered cards, it became a process of filtering comparisons between matching new card UID to registered card UID. If matched the servos would actuate, if not, do nothing.

The door system interacting with the feeder system became tricky, as it is important to have both devices synced together, it became apparent that defining the door recognising if the pet came through the door, the system does not know the direction, therefore I set a counter, using odd numbers for inside, even numbers for outside. So, when the devices came out of sync, and the pet approaches the feeder, both systems would re-sync setting this "counter" back to 1 (odd number = inside).

When a specific method is called by with pet's behaviour, the system publishes this data ("inside", "outside", fed = true", "fed = false"). Both devices subscribe to this data to also keep the system state in-sync, and allow for the IFTTT email notification trigger to act accordingly.

Once the system was functioning with robust testing, negating potential causes for communication faults etc, the scheduling was implemented last. The LED RING catalyses the visual aid for setting the schedule. Using the in built TIME library, I set the systems time to current AEST in setup. The feeding systems functionality is dependant upon the current time and the desired hour scheduling, therefore I implemented another "counter" that both the Time and individual LED ring corresponds to (12-hour format between 1 -12). When the button is pressed, the counter increments by 1, and stops at 12 before resetting back to 1. The LED RING represents this choice. Therefore, when a card is scanned, the device compares the current time to the devices set counter. I tested the actuation of the feeder by setting the schedule to different counter time hours outside the current real-time hour. The feeder successfully only serves food during the scheduled hour if and only if the registered tag matches the newly scanned tag.

IFTTT triggers were set once all functionalities had been completed:



Your pet is now Eating!

Particle via IFTTT <action@ifttt.com>
To ○ face.first@hotmail.com

If there are problems with how this message is displayed, click here to view it in a web browser.

The Safe Pet System has detected your Pet eating at June 6, 2022 at 08:28PM

Unsubscribe from these notifications or sign in to manage your Email service.

IFTTT

- User Manual

Setting up the system:

Once all wiring is complete and prototype is powered on for the first time:

1. Claim both argons to own Particle account and flash respective code onto devices.
2. Blue LED's will flash on both devices.
3. Scan new tag/card on FEEDER device RFID sensor (card ID sent to Door device through the cloud)
4. Buzzer will sound for 3 seconds, and blue LED's will turn off indicating tag has been registered.
5. Scan registered tag on door system to initialise system LED status's.
6. Set feeding schedule by pressing BUTTON on feeder device, each button press will change the LED's on the LED RING, press until Blue LED on strip is set to desired AM/PM HOUR time.
7. Press tag/card against feeder RFID to check that system has successfully initialized. (Green led will light up on both devices to indicate that the pet is indoors and both devices are synced.

Operation

Once both devices are synced, and pet tag/card is registered:

- Attach registered tag to pet collar.
- **This system detects pet behaviour and notifies the user via email notification.**
- If tag/card is scanned on feeder device, the device will check the current time against the user set schedule. (Feeder will activate if for example: schedule is set to 8 on LED ring, and current time is between 8 and 9 am/pm) The system notifies the owner of each outcome.
- If pet enters/exits through door flap, door device will send that data to feeder system, and both devices will update their respective LED's (Yellow = Outside, Green = Inside) The system notifies the user of each outcome.
- If pet enters the house through another entrance, Door system will obviously become "out of sync" with the pet's current state. This will be re-synced when the pet either approaches the feeder, or can be manually reset by pressing the BUTTON on the DOOR device until the requires LED that indicates the pet's current status correctly.

Troubleshooting

If an error occurs:

- Reset Schedule on feeder by pressing button to desired hour time again.
- Press button on door device to resync system state.
- Check servo motor wheel is attached properly.

- If problems persist: Restart Door system and Feeder system until Blue LED's flash, restart setup.

- Conclusion

The design, development and implementation of this project has been incredibly rewarding. I planned to incorporate all the new technologies and concepts learned throughout the unit into this project. This involved research into hardware specifications, communication protocols and prototype design ideas. Having a small background in Arduino allowed a fast-learning curve into the Particle development environment using C++. Most of the issues I faced were during the initial connections between the RFID sensor yet utilising its libraries and playing with the methods given proved to be useful for the behaviour I intended for the hardware to achieve.

Another challenging task was designing the structure of the device itself, using cardboard, glue, and tape to create small instruments for the servo's, architecture of the box itself and the food container/bowl. If I had an opportunity to do this project again, I would attempt to finalize the wiring prototype by having the designed PCB manufactured and incorporated into the prototype, thus having the prototype a complete package without external wiring. I would also utilize some more sensors such as a Distance Sensor to enable greater ergonomics/fault-tolerance when the pet is still at the feeder.