Neftali Hernandez
EE250 Writeup
May 08, 2020

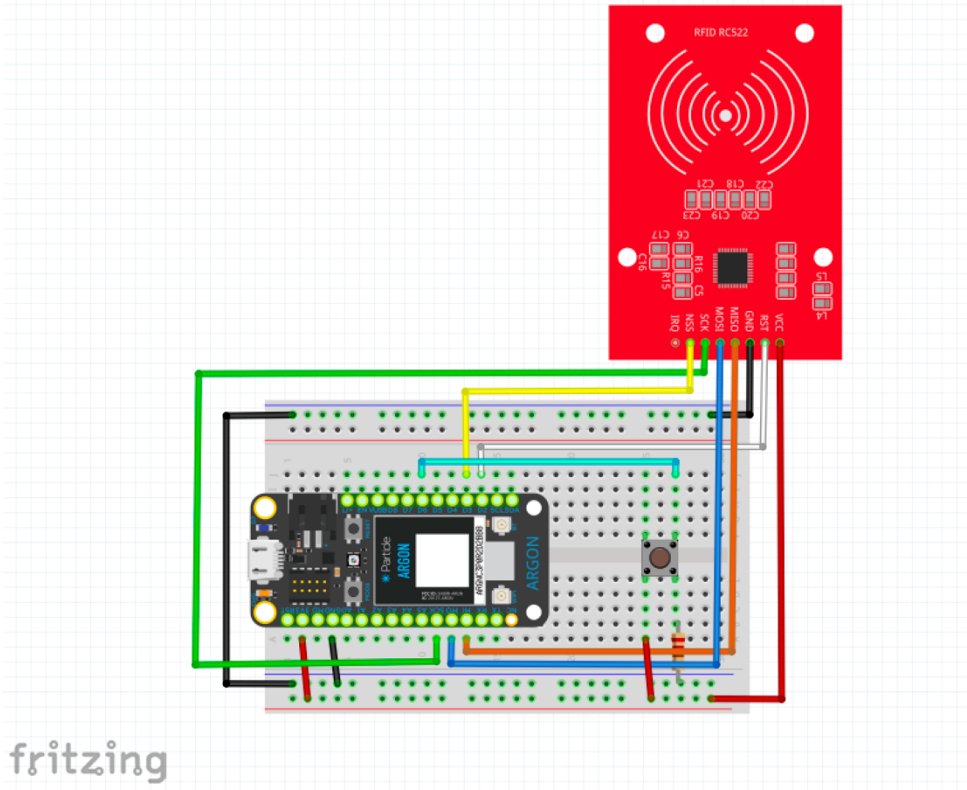
RFID Security System

Introduction:

For my project I decided to make a security system using two different devices. I incorporated both the Particle Argon device, which has WiFi and cloud capabilities, along with the Raspberry Pi 3. My IoT system aims to use two nodes, one to read data from an RFID scanner, and the other node to get that data without physically being connected to each other. Essentially, the Argon Particle device will be connected to the outside of the door scanning for incoming RFID tags. The Particle Argon then processes the data read from each Proximity Integrated Circuit Card (PICC) and produces the unique card ID, which it then publishes to the cloud privately. My Raspberry Pi, which is theoretically on the other side of the door, then connects to the appropriate channel through the Particle cloud and runs an event handler function to check the database if the PICC/RFID tag is a valid tag within the list of users. If it is valid then we can manually open the door for them.

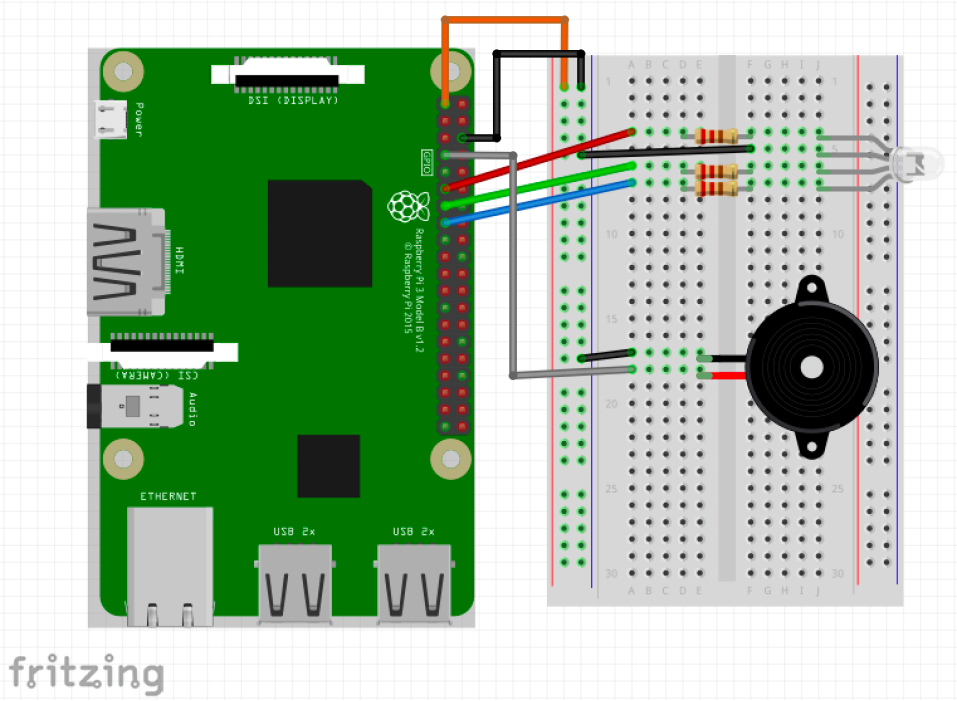
Setup and Wiring:

Particle Argon
(Made with Fritzing Software)



Argon	RC522
3V3	VCC
D2	RST
GND	GND
MI	MISO
MO	MOSI
SCK	SCK
D3	SS

Raspberry Pi 3
(Made with Fritzing Software)



RPI (GPIO PINS):
D0 – to buzzer. (Grey Wire)

D1 – to led leg #1. (Red Wire)

GND – to led leg #2. (Black Wire)

D2 – to led leg #3. (Green Wire)

D3 – to led leg #4. (Blue Wire)

RPI GPIO Reference

Peripherals	GPIO	Particle	Pin #			Pin #	Particle	GPIO	Peripherals
3.3V			1	X	X	2	5V		
I2C	GPIO2	SDA	3	X	X	4	5V		
	GPIO3	SCL	5	X	X	6	GND		
Digital I/O	GPIO4	D0	7	X	X	8	TX	GPIO14	UART
GND			9	X	X	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18	PWM 1
Digital I/O	GPIO27	D2	13	X	X	14	GND		
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23	Digital I/O
3.3V			17	X	X	18	D11/A2	GPIO24	Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND		
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25	Digital I/O
	GPIO11	SCK	23	X	X	24	CE0	GPIO8	SPI (chip enable)
GND			25	X	X	26	CE1	GPIO7	
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC	DO NOT USE
Digital I/O	GPIO5	D4	29	X	X	30	GND		
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12	Digital I/O
PWM 2	GPIO13	D6	33	X	X	34	GND		
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16	PWM 1
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20	Digital I/O
GND			39	X	X	40	D16/A7	GPIO21	Digital I/O

(https://core-electronics.com.au/tutorials/raspberry-pi-workshop-for-beginners.html#ch5)

Once you correctly wire your devices, you will need to register Particle Argon device through the app and Install Visual Studio Code on your Computer.

For the RPI, you will need to connect your RPI to a monitor, keyboard, and mouse. You will open up the terminal and type **bash <(curl -sL https://particle.io/install-pi)**

This will run set up for the RPI to be recognized by the Particle platform so it can communicate through the cloud.

Now we are ready to code projects. The argon code can be done on our regular laptop by opening up Visual Studio Code and creating a project. And our RPI code can be coded by opening Explorer on the RPI and going to build.particle.io

Performance & Challenges:

The system I created worked great with no noticeable errors. Publishing and subscribing to the events and reading what was being published worked fine on both ends. The particle cloud was pretty neat in that it encapsulated all the RSA public-private key pair connections and CoAP over DTLS protocols we needed to use to establish a connection. Their documents page goes into further detail. (https://docs.particle.io/tutorials/device-cloud/introduction/). One challenge I was having was figuring out how to use Particle cloud on the RPI since compatibility was discontinued a few years ago. Certain features were still working with this older version, so I managed to code a simple program after tedious hours of trying to download the particle agent to register my RPI with Particle.

As far as improvements, I would like to fit the RFID scanner in a neat container with an LCD screen and buzzer to display if the RFID tags are valid or not. I would also like to use a different cloud service that can connect with both Particle cloud and my RPI. A neat little feature I can add would be to utilize IFTTT to send text and email notifications to security official when an invalid RFID card has been scanned more than once.

Conclusion:

All in all, it was very fun getting to utilize concepts learned throughout the course to implement in simple IoT systems. Seeing how they can all work in parallel to make a connected system more diverse in functionality was pretty neat. Covering the topics, we did in class inspired me to look for other cloud services and nodes that could communicate to other devices like the Raspberry Pi. I definitely want to keep expanding on this project and be able to write my own ID tag to the card as well as be able to replicate my own USC ID card which is running at 125 Mhz. I will need to do further research to determine which sensors or devices would work well with writing to 125MHz ID cards.