

## Computer-Driven Experiences with Familiar Objects Week 2

### Prompt: A similarly designed object and it's evolution

While giving a variation of my elevator to some of the Texas Instruments representative at the networking event I was at last week, they reminded me of a term that I had completely neglected to consider when proposing my Capstone. That term was “Internet of Things (IoT)” and in my complete immersion in Mark Weiser’s 20-year-old work, I had completely forgotten that the devices he describes are referred to nowadays as IoT devices. As such, the similarly designed object that I can look at is Amazon’s Echo home speaker, or really any home “smart” speaker or device that falls under the umbrella of IoT. The Echo speaker came out in 2014 and it’s main appeal, judging by the [first-generation product page](#), was the ability to play music hands-free. Since 2014, Amazon has tied together all of its products to interface with each other, using the Echo and its many microphones as the “ears,” and its hardware partners (Samsung, Phillips, Lenovo, etc.) have released a plethora of “Alexa-enabled” devices that all interact with each other in a way that resembles a pervasive computing environment. Admittedly, there is some form of learning curve, since the devices don’t actually understand human speech and the commands are clunky at times. Once you get used to it, though, it yields an experience where simply speaking in your house can fire up your “Alexa-enabled” smart tv, play music over your “Alexa-enabled” speakers, or dim your wireless “smart” lights.

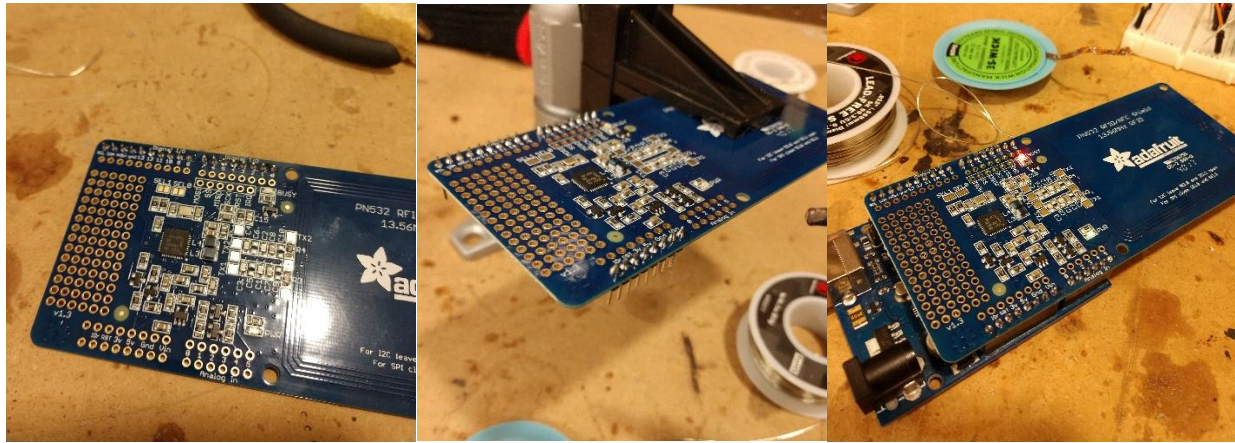
Back to the networking event, I took the last 30 minutes of last week’s class to go to this event where a bunch of UMD’s corporate sponsors were gathering to talk to Aerospace, Mechanical, and Computer Engineers about internship and employment opportunities. Naturally, I shamelessly plugged my Capstone project into every conversation I had in order to talk myself up. In the process of doing this however, I received some very valuable input about how to pitch my idea in the future. The majority of the advice pointed out that, even in the abstracted form I had it, there was still too much technical lingo being tossed around. A lot of what I was saying could be summed up with the term “Internet of Things.” This term alone will help immensely when talking to stakeholders, as there are a lot of ideas, like the theory of “pervasive computing”, baked into it that an expert in the field will already know. This will allow me to keep my questions short and sweet, without the clunkiness that was present in my Elevator Pitch.

### **Week 2-3:**

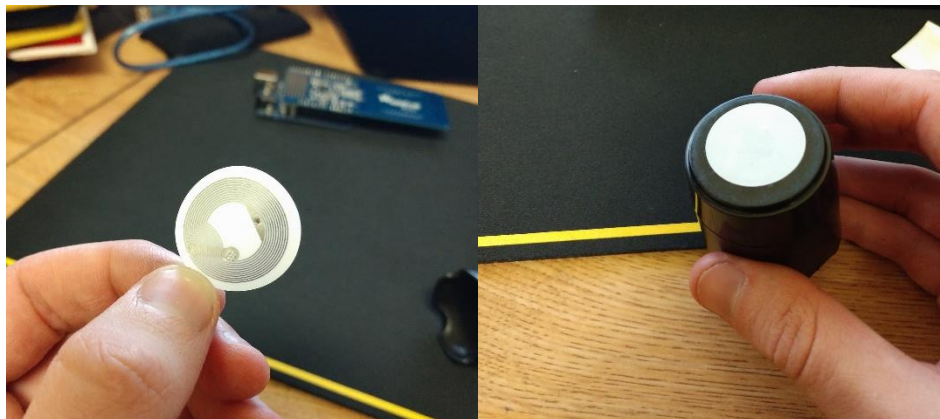
- Find more user experiences (continuous)
- Start working on the C++ library (2 days)
  - Try to at least get down function prototypes that apply to the user experiences gathered so far.
- Establish serial communication between the Host Machine and the Arduino (3 days total)
  - Can I get the Arduino to run a command on Windows/Linux/Unix? (1 day)
  - What are the limitations of data transmission over USB? (2 days)
- Establish a reliability standard for the NFC communication (2 days)
  - How often does communication drop?
  - What is the effective distance?
  - Etc.

Now, onto the progress for Week 2 of Week 2-3!

So, the Adafruit PN532 shield arrived, and I was able to solder all of the pins to it, despite the horrible DCC soldering irons. This took me about an hour overall.

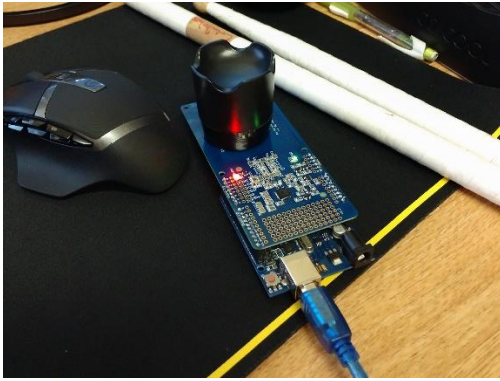


I've done some research into what scripting languages would be best to write in so that there's a) plenty of support in case I get stumped, and b) lots of libraries that support functionality like Google searching or Spotify integration. I have decided to ditch Batch/ZSH for Python. The extra bonus is that I already own plenty of literature on Python that I've been studying over the Summer, so refreshing my knowledge should be easy, and that Visual Studio has a Python plugin that is free for students called PyCharm! The Arch Linux environment has been setup, but because it's literally just a CLI, I can't really take a screenshot of what it looks like. The Arduino IDE has also been installed, and I wrote a sketch that writes to an NFC tag and was able to the successfully write the message "Hello World!" and then read it. Because the cymbal is a little too large to have on my desk all of the time, I used the little plastic stand for my Wacom Pen and put the NFC tag on the bottom (the white sticker). Experimenting and searching took me around 6 hours this week.



So, as this pertains to my actual schedule, I haven't found new user experiences yet, I started on the C++ library last week but I haven't finished the function prototypes, I've setup the NFC and found the [Python library](#) required to run the scripts from the Arduino, and from the testing I've done, the NFC communication is pretty flawless once its established, but the effective distance is very very small and is easily obstructed by something the thickness of a wallet (see below).

Works!



Does not work!

