

Computer-Driven Experiences with Familiar Objects Week 1

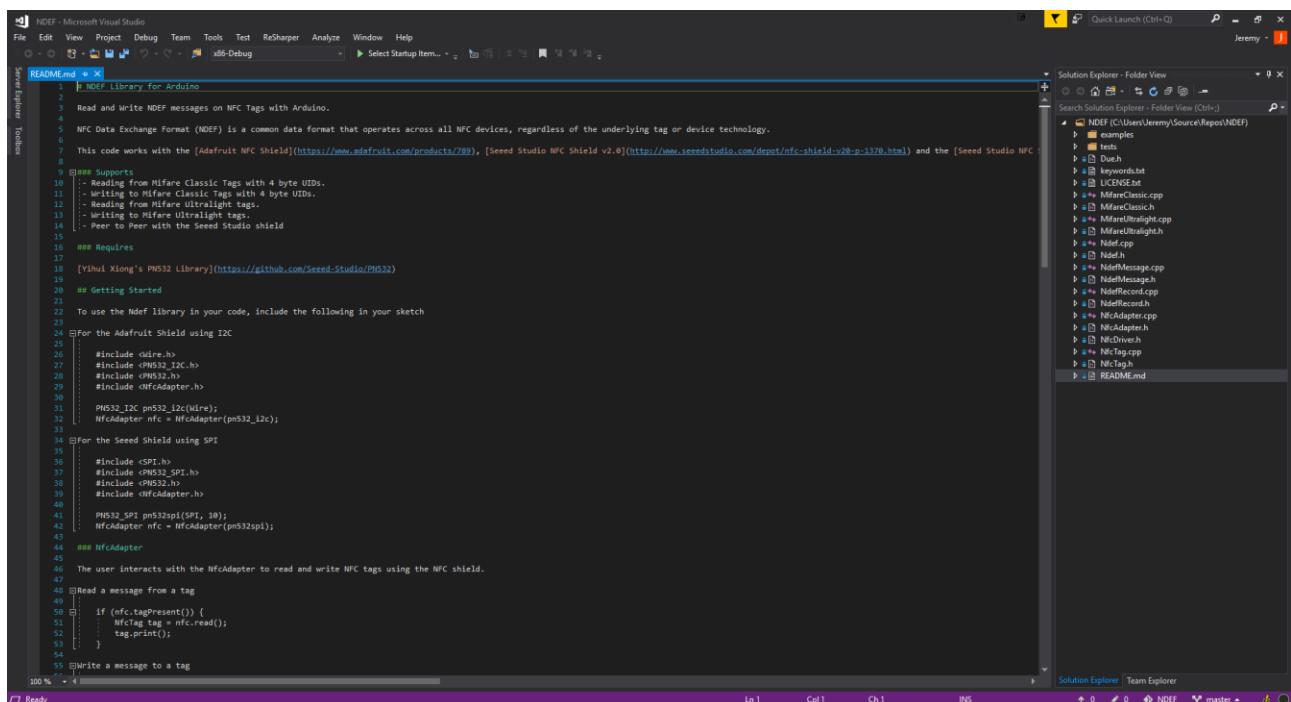
Having initially been confused about the whole process of getting the capstones approved and funded, I still find myself without one of the main components, the [Adafruit NFC shield module](#). Despite this, it hasn't pushed back my schedule too much, considering that I have begun the coding process, and the only thing this will prevent me from doing is testing the NFC functionality of the code. While the NFC functionality is the crux of the Capstone, there are a lot of things I can do that do not require the NFC, such as curate content, and make sure the Arduino can interface with the Host Machine. However, considering how ambitious my original timeline was, and the delay in getting components, I'm just going to push everything back 1 week in my final proposal. I'll use the time that this grants me to find more user experiences so that I can trim down the fat later if one particular experience doesn't pan out.

Speaking of user experiences and my original goals, though. My original "Before Spring Break Goals" looked like this:

Before The Spring Semester:

- Setup the build environment for the C++ library
 - Find an Arduino NFC library to build off of
- Host Machine platform decided on
- Acquire the [Adafruit](#) NFC Shield and ~10 NFC tags
- One or two user experiences curated
 - Having ordered the parts for the necessary modules

And out of those goals, I accomplished every single one except for acquiring the [Adafruit NFC Shield](#). I have setup the C++ build environment in Visual Studio using the JetBrains Resharper plugin, and I'm going to use Arduino coding paradigms and style guides outlined on their [website](#). I cloned a copy of the NFC PN532 library from the NDEF Github and will use it to retrofit the serial functionality that I want to implement (See Below). NDEF is the name of the non-profit organization enforcing the NFC standard.



```
README.md
1 | Install library for Arduino
2
3 | Read and Write NDEF messages on NFC Tags with Arduino.
4
5 | NFC Data Exchange Format (NDEF) is a common data format that operates across all NFC devices, regardless of the underlying tag or device technology.
6
7 | This code works with the [Adafruit NFC Shield](https://www.adafruit.com/products/789), [Seed Studio NFC Shield v2.0](https://www.seedstudio.com/deposit/nfc-shield-v2.0-p-1278.html) and the [Seed Studio NFC
8 | ## Supports
9 | - Reading from Mifare Classic Tags with 4 byte UIDs.
10 | - Writing to Mifare Classic Tags with 4 byte UIDs.
11 | - Reading from Mifare Ultralight tags.
12 | - Writing to Mifare Ultralight tags.
13 | - Peer to Peer with the Seed Studio shield
14
15 | ## Requires
16
17 | [Yihui Xiong's PN532 Library](https://github.com/Seeed-Studio/PN532)
18
19 | ## Getting Started
20
21 | To use the Ndef library in your code, include the following in your sketch
22
23 | ## For the Adafruit Shield using I2C
24
25 | #include <i2c.h>
26 | #include <PN532_I2C.h>
27 | #include <PN532.h>
28 | #include <NfcAdapter.h>
29
30 | PN532_I2C pn532_i2c(i2c);
31 | NfcAdapter nfc = NfcAdapter(pn532_i2c);
32
33 | ## For the Seed Shield using SPI
34
35 | #include <SPI.h>
36 | #include <PN532_SPI.h>
37 | #include <PN532.h>
38 | #include <NfcAdapter.h>
39
40 | PN532_SPI pn532spi(SPI, 10);
41 | NfcAdapter nfc = NfcAdapter(pn532spi);
42
43 | ## NfcAdapter
44
45 | The user interacts with the NfcAdapter to read and write NFC tags using the NFC shield.
46
47 | ## Read a message from a tag
48
49 | if (nfc.tagPresent()) {
50 |     NfcTag tag = nfc.read();
51 |     tag.print();
52 | }
53
54 | ## Write a message to a tag
55
```

It's interesting that I'm going to be implementing both the RC modules and Host Machine functionality using "Serial", but those both means very different things in their context. When I refer to Serial in terms of the host machine, I mean over USB, and when I refer to the RC circuits I refer to the serial output pins on the Arduino. The only functional difference is that the baud rates will be different, and the microcontroller chip that facilitates the connections are in different locations. This is important to note, because translating Serial in both of these instances is going to be vastly different, with the Host Machine, I'm going to need to read key bytes that will be read by the machine that will queue commands on the machine, and I've decided that I'm going to be using a Linux CLI or TTYL for the majority of this, (Video content will need to be played through the Window Manager via VLC or some other easily scriptable video player). I'll look into a few of my Python scripting books to figure out what kinds of content I can serve entirely from the CLI/TTYL. **SCRIPTING IS MY FRIEND.**

I've also curated my first user experience! I know it's probably cheating, but it's going to be my own experience. I'm going to put an NFC tag on this cracked Sabian AAX Crash Cymbal!



This cymbal belonged to one of the drum sets at the summer camp where I've been going since 2011. I went to this camp as camper for 3 years, and I've been a counselor there for the last 5 years. This

cymbal has been on the drumset every year that I've been there, and I can safely say that I am most likely responsible for its demise, as it only ever cracked when I was playing on it. This year it was on its last legs, so I took it. There are a lot of musical memories stored in the object for me in the form of songs that I've played over the years at this camp, as well as the aesthetics and ambience of the recording room where it was stored. This object evokes a lot of physical feelings for me, even temperature! The room that this cymbal was in was always incredibly hot. I'll have to find a way to incorporate all of that into the experience.