Hanna Lee

April 5, 2016

6.115 Final Project Proposal

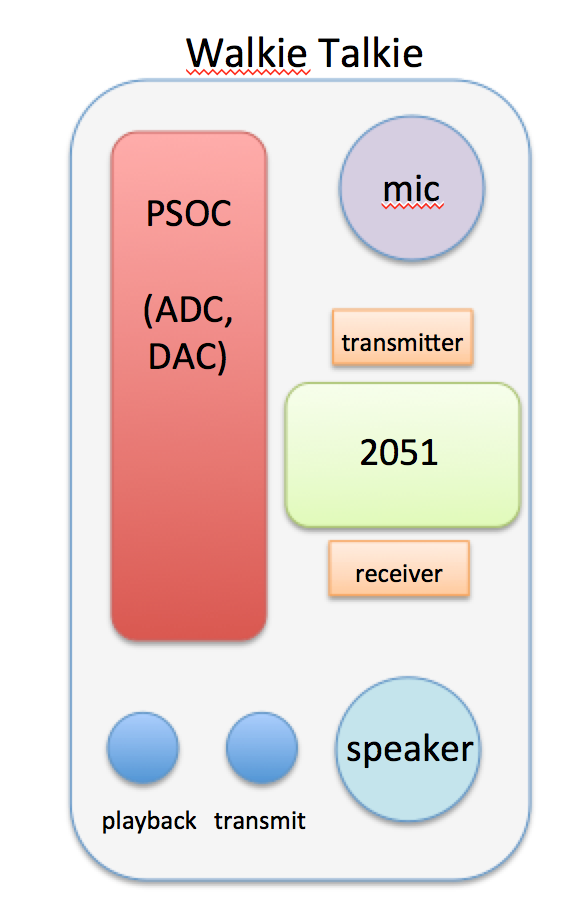
Walkie Talkies with Encryption and Modulation

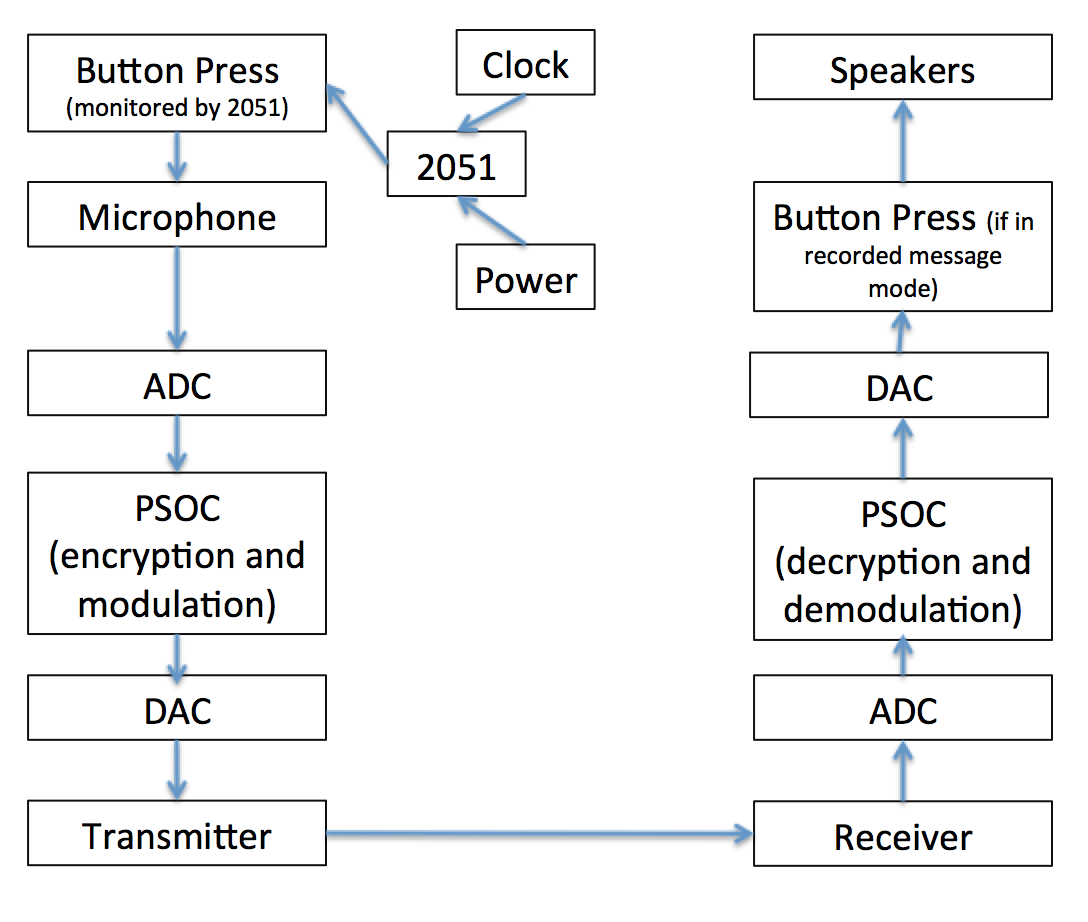
**Background/Introduction**

This project will involve creating two identical walkie talkies, that communicate with each other over either FM or Wi-Fi or Bluetooth (haven’t decided yet). The communication will be encrypted and decrypted using RSA, a widely used public key cryptography scheme (or a symmetric key scheme, I haven’t decided that either…) The walkie talkies will also provide options for modulating and manipulating the audio in different ways, such as lowering the pitch of a voice. There will be two options: one more live transmission, and one for “leaving a message,” such that the recorded message will not be played out of the speakers until the receiving party presses a button. Finally, time permitting, and if I choose to use Wi-Fi or Bluetooth, I will make a web app that allows me to record a messaged on my iPhone and send it to a walkie talkie.

This project excites me because it will allow me to learn about radio transmission, and play around with audio, and implement a cryptographic scheme that I’ve learned about in 6.857 this semester. I’m interested in security and cryptography, so I thought this would be a good way to integrate that interest into a microcontroller project. Additionally, I wanted to explore audio, so I will also play around with modulating the sound.

**Hardware Description**

I will create two identical walkie talkie devices. Each one will have a small microphone, small speaker, a psoc4, an 2051, and two buttons. I will use the microphone and speaker circuits in Figure 3, as described by Professor Leeb. The buttons will be hooked up to an external interrupt on the 2051, so that I will be able to detect button presses to know what mode the walkie talkie is in. I will create a DAC and ADC on the psoc4, for converting the mic signal to digital on the transmission end, and converting the digital signal to analog to feed into the speaker on the receiving end.



Wi-Fi or Bluetooth or FM (I am indecisive)

Figure 2. Block diagram and flowchart for a single transmit and receive interaction.

Figure 1. General outline of walkie talkie layout.

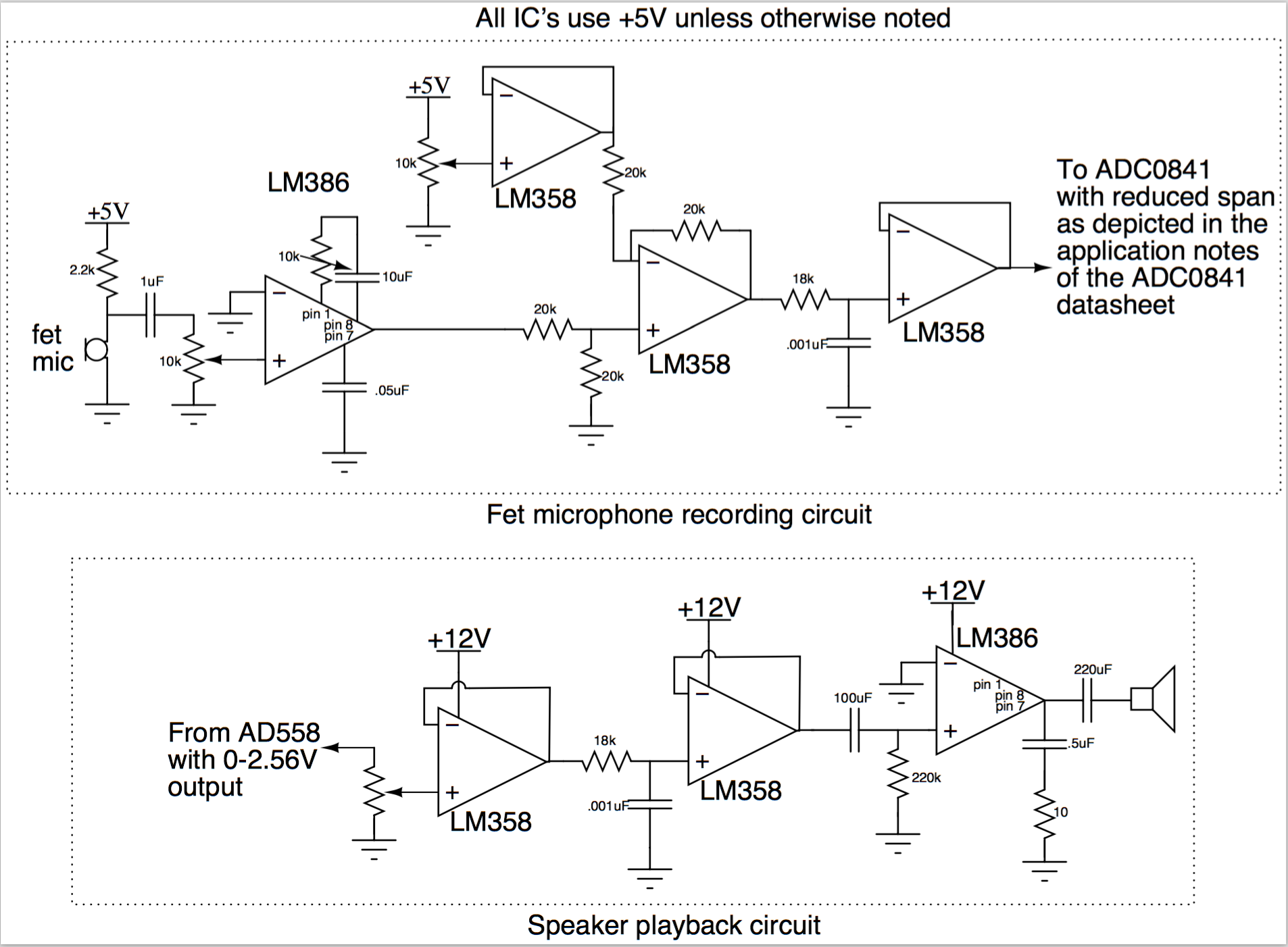


Figure 3. Professor Leeb’s suggested microphone and speaker playback circuits. I plant o connect them with the AD558 and the ADC0804

**Software Description**

*On the transmitting side:*

The 2051 will be responsible for reading button presses to see when the user is recording. This code will simply wait for a button press, identify the button, and then check for when the button is released. The psoc will receive a signal indicating that the user is beginning a transmission. There will be a header that is 00 if the transmission should be read live, or 01 if the transmission should be buffered. The buffering happens on the receiving side, but the transmitting side still needs to know, so that it can broadcast the header correctly. Next, a stream of bytes from the microphone will come in. After the user has stopped recording or the message has exceeded a maximum length, I add a footer to the message, that is a string of FF bytes (some fixed number of them, like 20 or so), to denote the end of the message. Then, I convert these bytes to their decimal representation, and package the bytes up as blocks, and encrypt each block using RSA, and then broadcast that over Wi-Fi.

*On the receiving side:*

I will receive bytes from the Wi-Fi/Bluetooth/FM receiver, and the psoc will decrypt the bytes. The psoc will look at the header, and see if the transmitted data should be played back immediately or only after a button press. If the playback is immediate, I will begin decrypting the data, and sending it back out through amplifiers, to the speaker. Otherwise, I will decrypt the data and buffer it, then wait for an external interrupt on the 2051, which will feed into the psoc to tell the psoc to begin sending the data out through the DAC, through the amplifier, and into the speakers. The psoc will recognize the footer, which is a string of FF bytes.

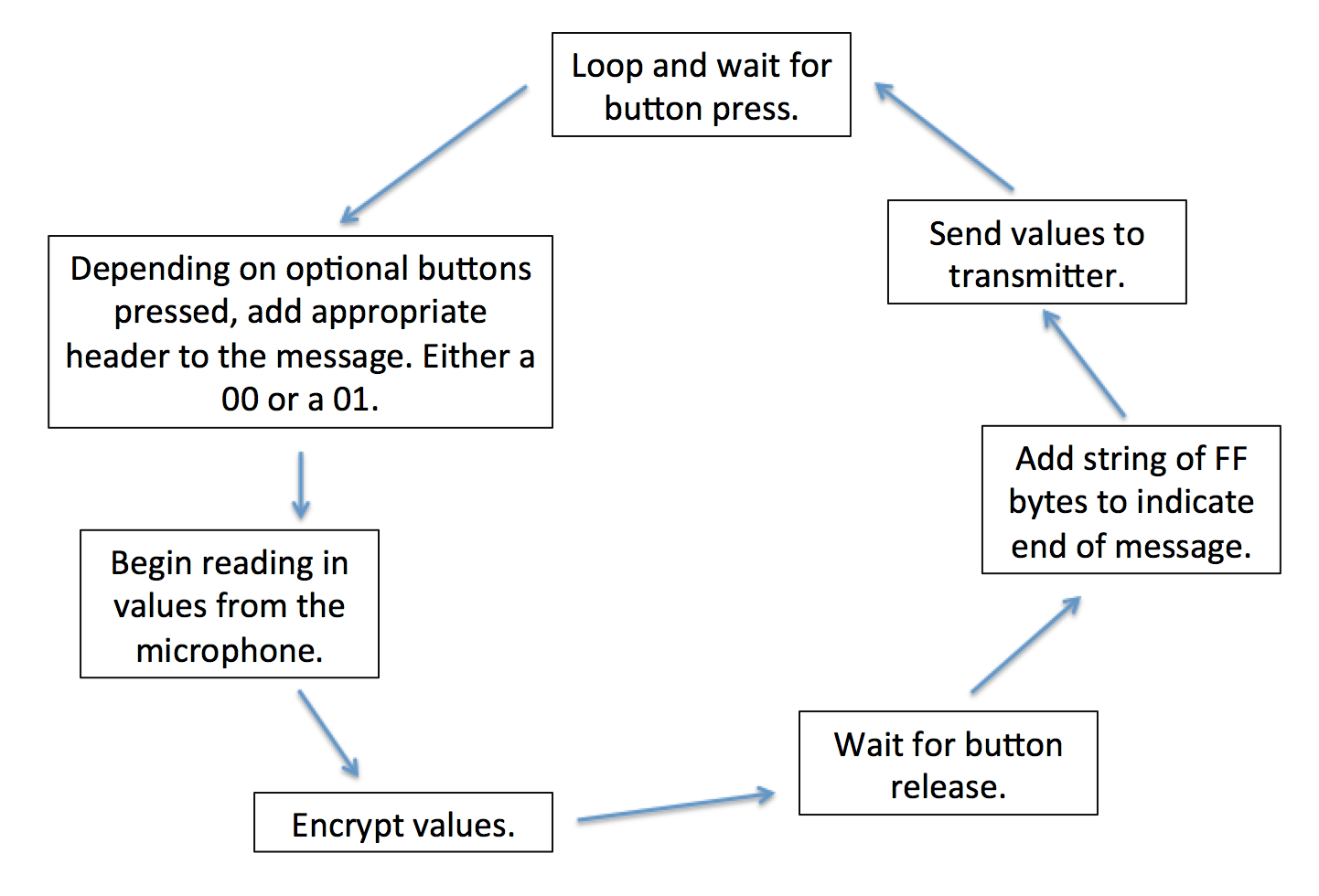
****

Figure 4. Transmission side software flow. 

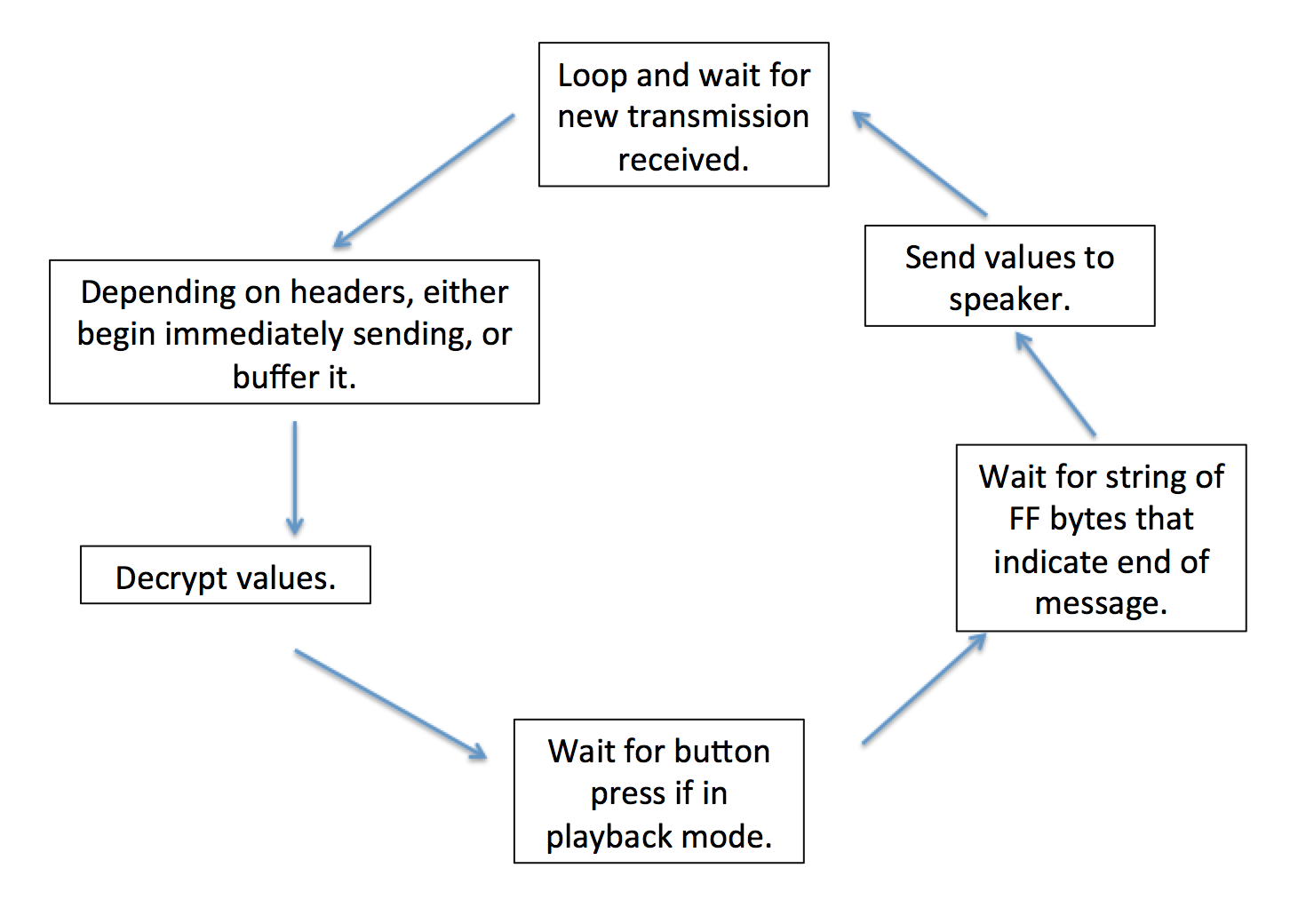
****

Figure 5. Receiver side software flow.

**Project Scope and Management**

This project can be broken down into multiple stages.

Modest Risk Level (“B” Grade): The initial stage is to simply get the two walkie talkies communicating with each other. I would be able to press the “talk” button on the transmitting side, speak into the walkie talkie, and a receiver in the other room with the other walkie talkie would hear my voice audibly and clearly.

Exciting Risk Level (“A” Grade): The more interesting stage of the project, which I anticipate being able to do, is to encrypt communication on the psoc before transmitting, and decrypt communication on the receiving end before playing back through the speakers. I would also add in a button for the immediate vs. recorded message modes. In “immediate” mode, the walkie talkie will behave as already described. In “recorded message” mode, the receiving walkie talkie will buffer the message in memory and only play it back through the speakers when the user presses the button. There will also be an LED to indicate that there is a message, or even multiple messages.

Spectacular Risk Level (“A+” Grade): I would add in cool voice modulation features, where if I pressed one button then transmitted, I would have a lower voice, or a higher voice. I would add in authentication, using MACs to verify the identity of the sender, so that even if someone sent encrypted messages to the walkie talkies, I would be able to detect that the sender was not the holder of the other walkie talkie. Maybe, if I choose to use Wi-Fi/Bluetooth, I’m thinking of also writing a simple web app that would allow me to speak into my phone’s microphone, and have that get broadcasted to both walkie talkies, but I’m not quite sure how that’d work with encryption.

**Special Component Needs**

First, peripheral chips – I will need a DAC (AD558), ADC (ADC0804), and either an FM transmit/receive pair of chips, or a chip that does Wi-Fi. I will also need several op amps to scale the signal out of the microphone.

I will need some small breadboards and a power supply, to hook up the 2051 to. I will also need the microphone and speakers that Professor Leeb emailed out about. The mic is: <http://www.digikey.com/product-search/en?keywords=102-1720-ND> and the speaker is http://www.allelectronics.com/make-a-store/item/sk-61/1-speaker-in-enclosure/1.html.

Finally, for mechanical parts, I will need the physical bases, likely rectangular, that will be the walkie talkies. These will likely be either plastic or wood, and I will acquire them from the Edgerton Center, or from CSAIL.

**Timetable**

*Week of April 11:* Get all of my parts together, including all peripheral chips, microphone/speaker, psoc4, mechanical base for walkie talkies. Hook up the DAC and ADC and be able to read in microphone input to the psoc.

*Week of April 18:* Hook up the wireless/Bluetooth/FM chips. Get the walkie talkies communicating with each other, such that I can hold the button down, speak into one, and have a receiver in another room hear and understand what I said.

*Week of April 25:* Get the software to do encryption and decryption. And, get the second mode of the walkie talkie, in which a message is buffered and only played back to the receiver when the receiving user presses a button, working properly.

*Week of May 2:* Add in capability to buffer multiple messages; add in cool voice modulation features.

*Week of May 9:* Debug, debug, more debug. Prepare final presentation and polish the writeup.