Project 3: Design Project

UART Audio Sequencer

Brandon Nowak, Kelechi Igwe, Reed Slobodin

CPE 316 - 01

Spring 2020

Dr. John Oliver

1. Behavior Description:

Summary

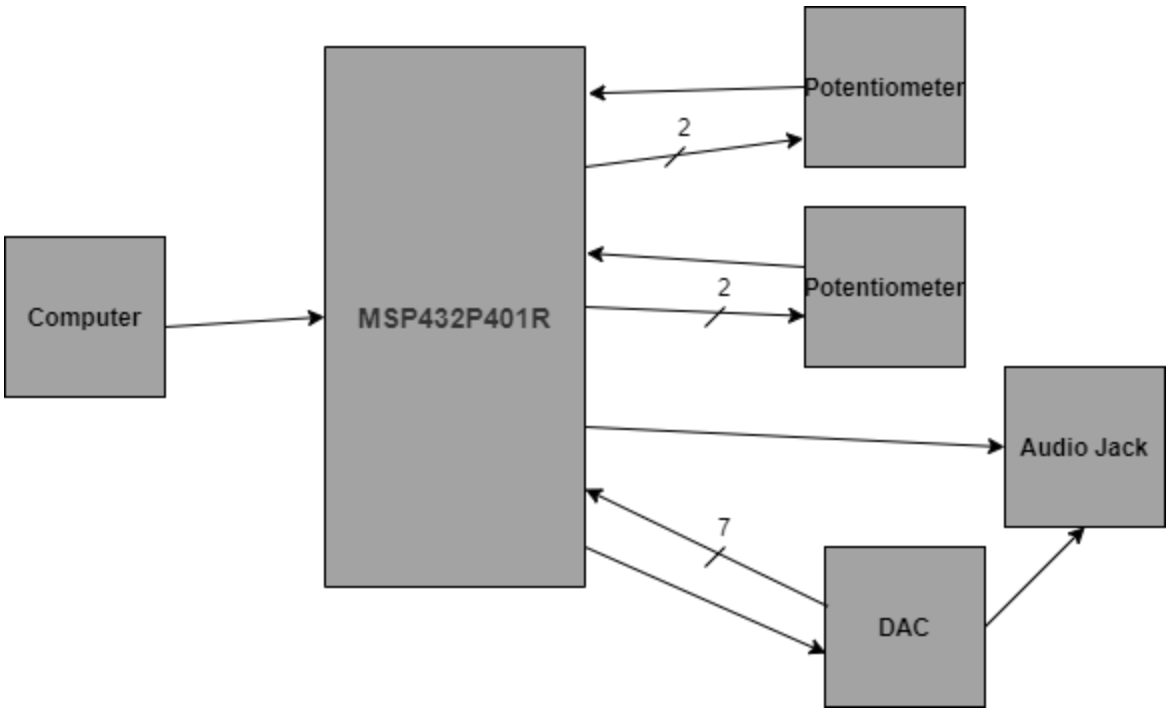
For our final design project, our group decided on an audio synthesizer, controlled using a program on a computer that communicates with the MSP432P401R via the UART serial protocol. The program, coded with Python, provides a GUI that allows the user to select a port to communicate with as well as a baud rate for the communication. Once the connection has been established between the transmitter (computer program) and the receiver (MSP432), the program allows the user to build any sequence of musical notes between A2 and A5. Once the user has formulated a musical sequence to play, the program takes that sequence and arranges it in a byte array so that the C program can properly receive and arrange the sequence onto the MSP432.

Once the MSP432 receives the character bytes from the Python program, the microcontroller proceeds to play the sequence. The MSP432 uses timers and interrupts to manage and coordinate the different modules at work while the microcontroller is playing the sequence, including: output to the DAC, control of the sound volume, and control of the sequence's tempo. For the DAC, the MSP432 uses the SPI protocol to send values to the MCP4912 DAC. The output is a square wave at a specific frequency, based on the note that needs to be played (Ex: To play A4, the DAC would produce a square waveform at 440 Hz). This output is then connected to an audio jack, and the sound produced from the waveform can be heard through any audio device that can be connected via 3.5mm jack. The volume and the sequence tempo are both controlled by rotary potentiometers connected to the MSP432's ADC with a 14-bit resolution. The values of the potentiometers are read through the ADC, and change either the beats per minute (BPM) of the sequence or the volume of the notes (BPM ranges from 60-180 quarter notes per minute). Once the sequence is done playing, if the user selected in the Python program to play the sequence once, the MSP432 will wait until a new sequence is sent to the program. If the user selected in the program to loop the sequence, the sequence will restart and play continuously until a new sequence is sent to the program.

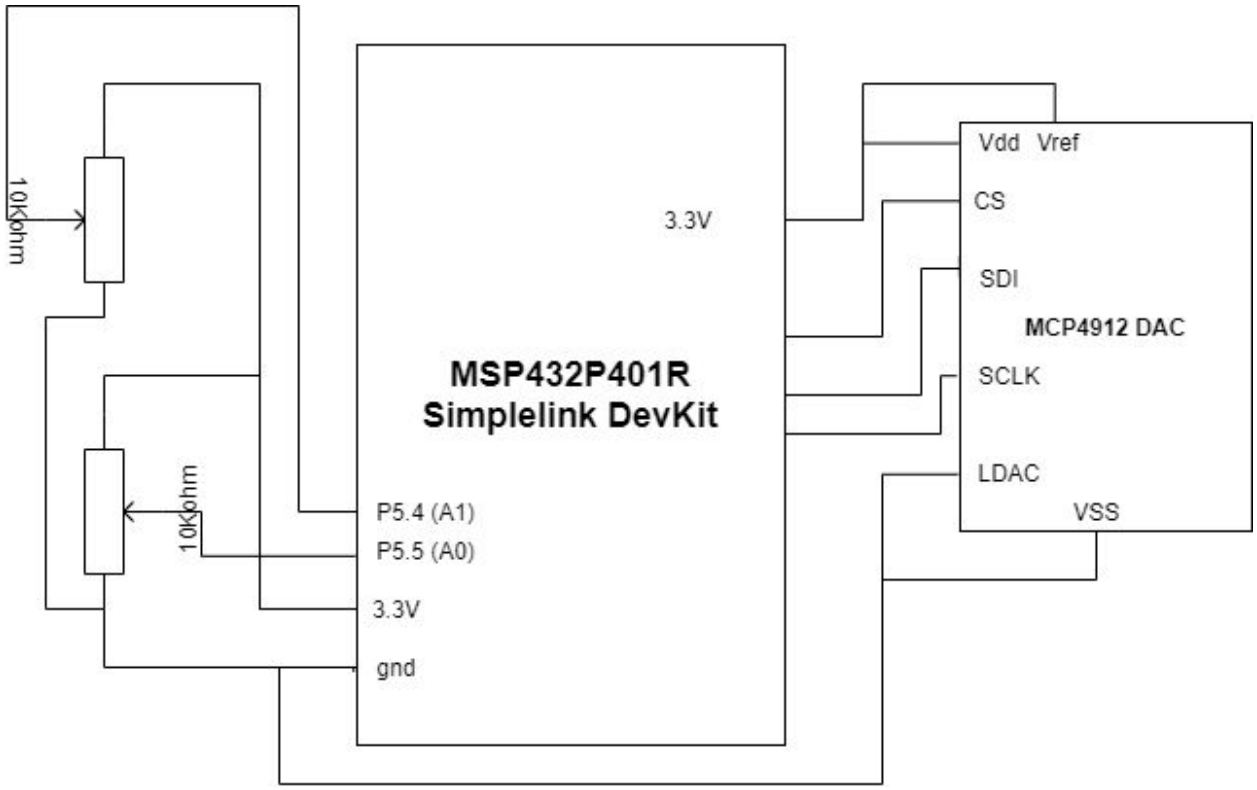
2. System Specification:

Specification	Value	Range (if applicable)
MSP432P401R [1]		
Power Supply Voltage	3.0V	1.6V to 3.7V
Power Consumption	~3.72mA	~2mA to ~4.59mA
ADC Resolution (bits)	14	N/A
Clock Frequency	24MHz	1.5MHz to 48MHz
Physical Size	14mm x 14mm	N/A
MCP 4912 DAC [2]		
Power supply Voltage	3.3V	2.7V to 5.5V
DAC resolution	10 bits	N/A
Number of channels	2	N/A
Voltage reference	External 3.3V	External
Clock frequency	12 MHz	12 MHz
Physical Size	0.3in (Package Width)	N/A
Weight	0.95g	N/A
Rotary Potentiometer [3]		
Resistance	10KΩ	N/A
Mounting Height	~10mm	N/A
Power Rating	0.05W	N/A

3. System Architecture:



4. System Schematic:

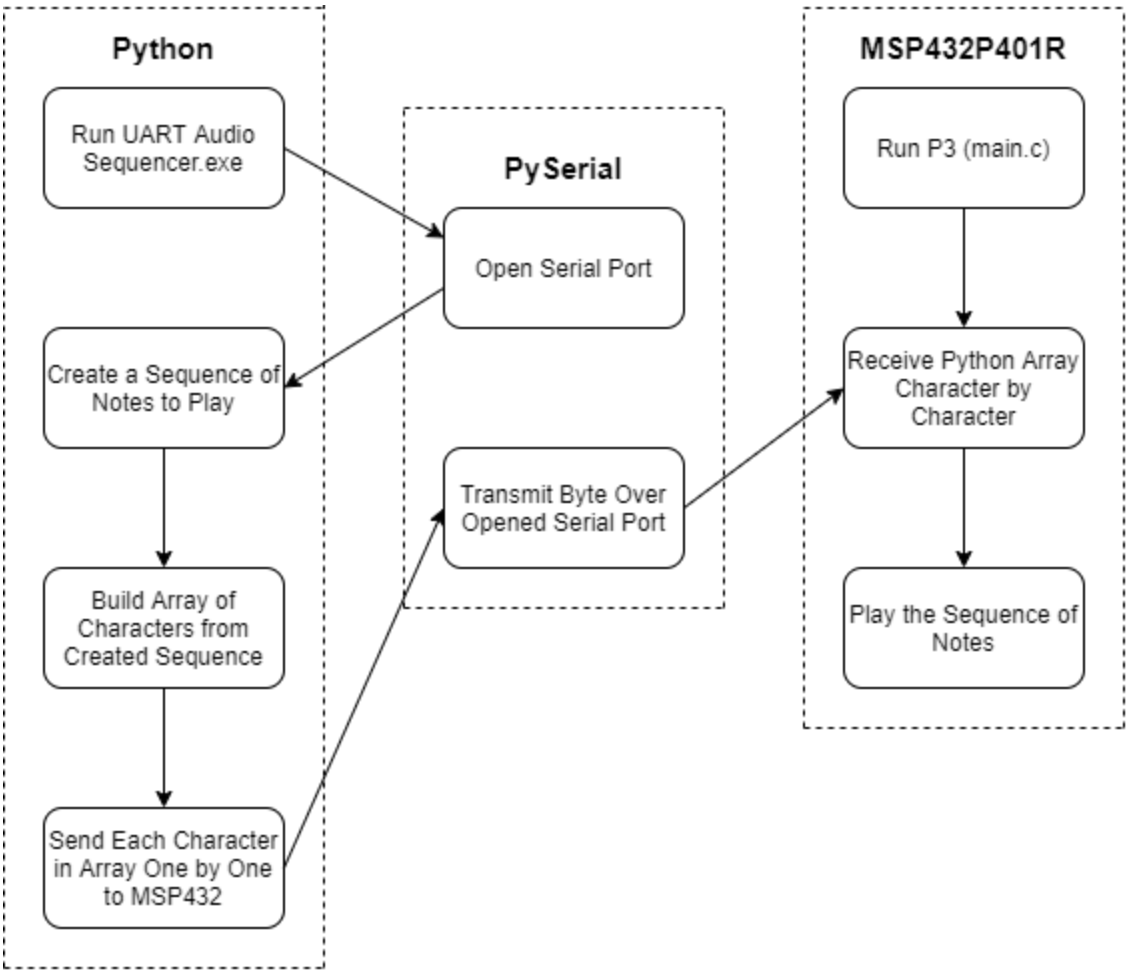


5. Software Architecture:

Prerequisites for Python Code (in Appendix B)

Library/Language	Download Link
Python (program built on version 3.6)	https://www.python.org/downloads/
PySerial	Pip: `pip install pyserial` Link: https://pypi.org/project/pyserial/#files
Tkinter	Included with Python Download

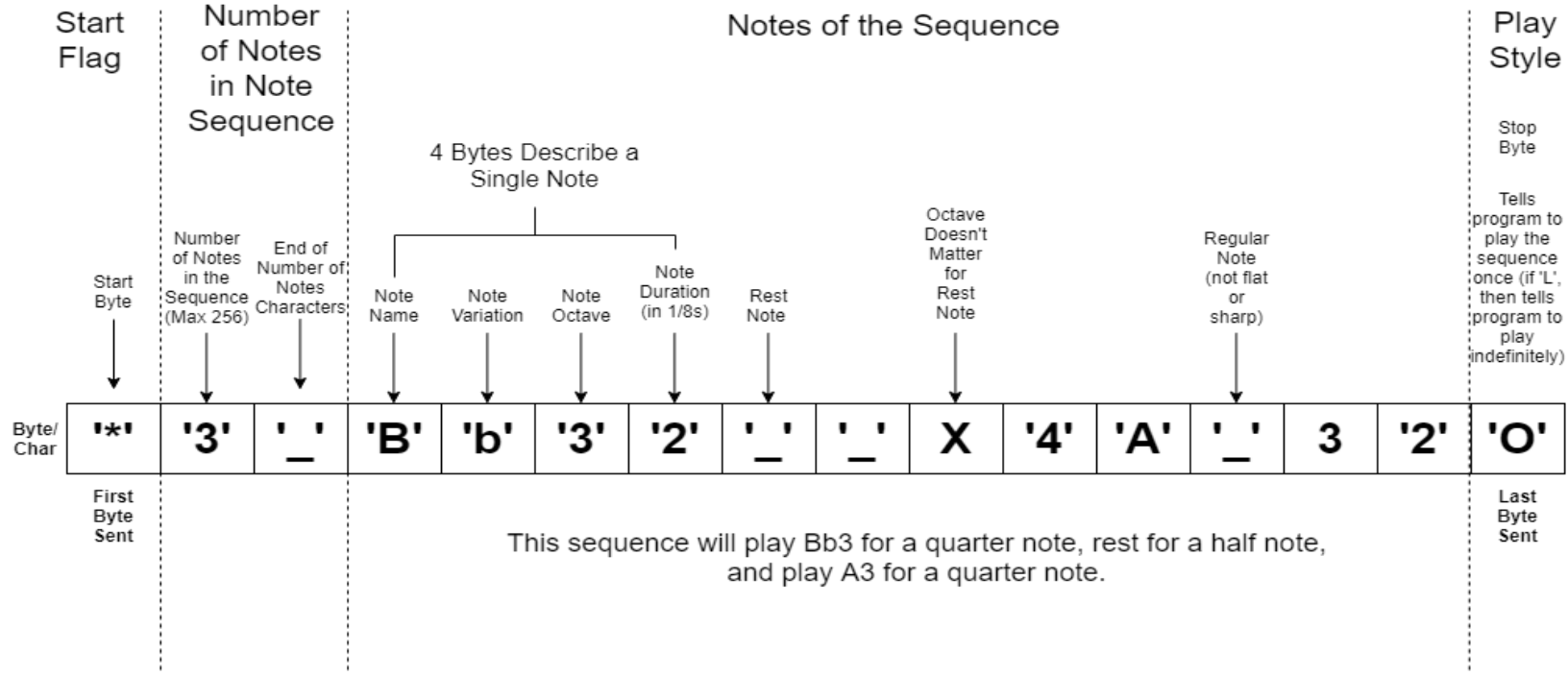
UART Audio Sequencer Software Architecture



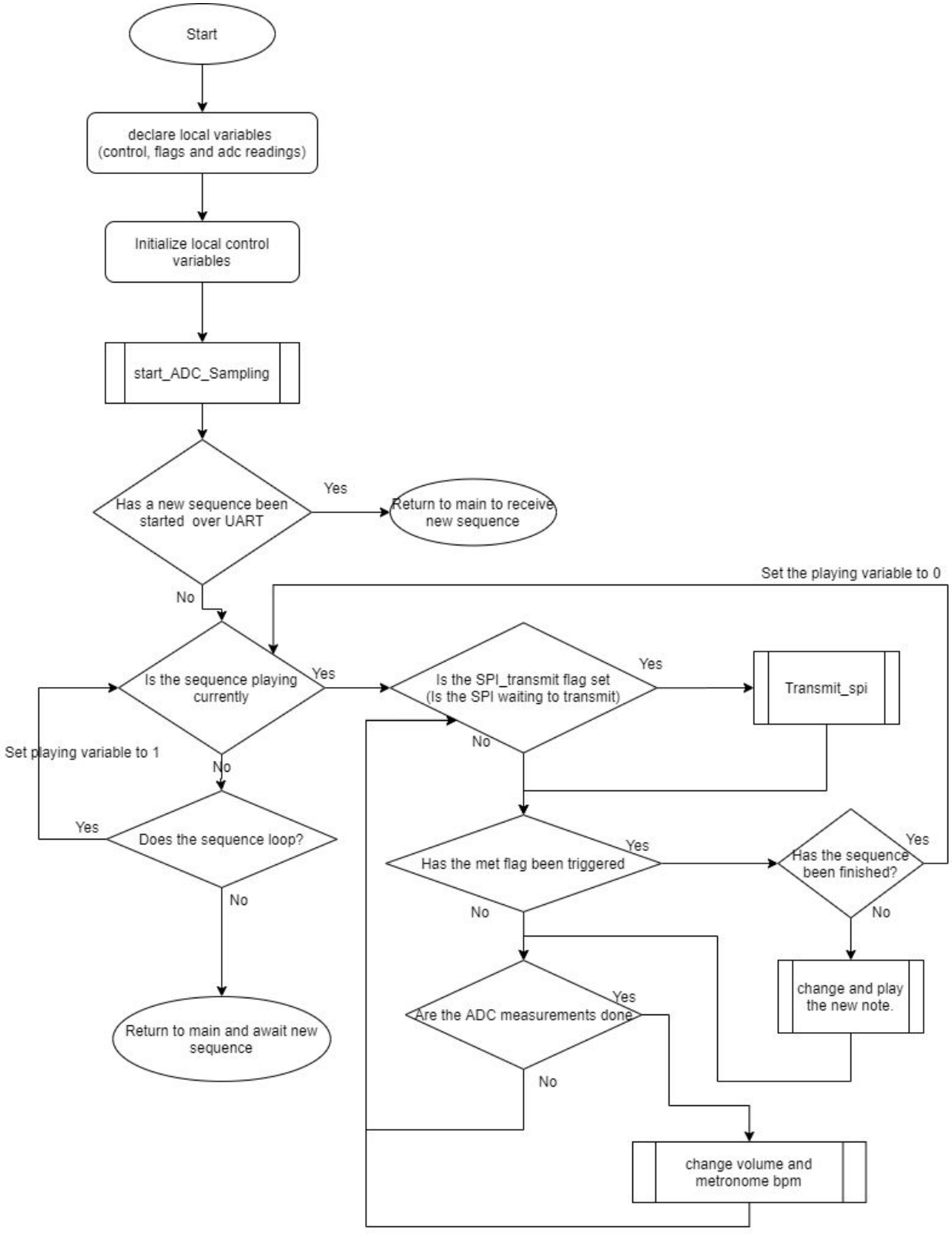
Character Descriptions

Character	Purpose
'*'	Start of a New Note Sequence
'A'-'G'	Musical Note
'b'	Flat Musical Note (A Semitone Down from Musical Note)
'_'	No Note (Break) / No Variation (No Sharp or Flat)
'L'	Stop Byte of Sequence / Command to Loop Sequence
'O'	Stop Byte of Sequence / Command to Play Sequence Once

Sequence Format



Play_sequence flowchart



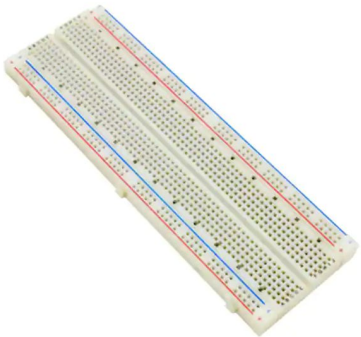
6. Bill of Materials:

1) Microcontroller LaunchPad: MSP-EXP432P401R (Estimated Cost per Unit: $23.59) (see figure below)

.

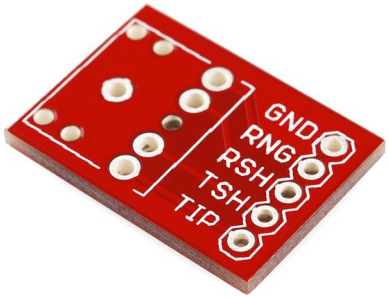
2) MCP 4912 Digital - Analog Converter (Estimated Cost per Unit: $1.83) (see figure below)

3) Pololu 830-Point Breadboard (Estimated Cost per Unit: $4.95) (see figure below)



4) Audio Jack 3.5mm (Estimated Cost per Unit: $1.50) (see figure below)

5) SparkFun Audio Jack Breakout (Estimated Cost per Unit: $0.95) (see figure below)



6) 2 Rotary Potentiometers - Linear (10KΩ) (Estimated Cost per Unit: $0.95) (see figure below)

Total Estimated Cost: $34.72

7. Ethical Implications:

Kelechi

Our UART Audio Sequencer is a simple product that can be used to play millions (okay maybe not millions but a lot) of musical sequences through a speaker using the MSP432. However, even the simplest of projects come with moral ethical issues that we as computer engineers must deal with. The two projects that we have had in this class dealt with hardware and software that were technologically local to the MSP432P401R and Code Composer Studio. However, our final project implements software that is out of the control of the MSP432. When our project is loaded and running on the MSP432, the complementary Python program has more control over the microcontroller than the microcontroller itself due to how our project is set up. In our project, the Python executable can start and stop any sequence playing through the microcontroller into the speakers. While this is useful for any audio sequencer, it can also be used as an entry point for hackers to affect the program. For example, if the user wants to play a sequence, they would enter the notes of the sequence into the Python program and then press the button that will send that sequence over a serial port on their computer into the microcontroller. However, somebody else that has access to the serial port can also send characters that the board will read. Therefore, if a hacker wants to prevent the user from playing a sequence, they can constantly be sending the start byte character to the board to prevent it from playing one sequence. As a result, we have an ethical dilemma where an added feature helps the user a lot, but at the same time puts their product at more of a risk (a double-edged sword).

Brandon

As a product for consumer leisure, this project is relatively free of ethical dilemmas. The main two that come to mind are to do with exposed circuitry and general misuse by the end user. As is the case with any circuit, there is risk of a short due do shoddy construction, or from the consumer's own misuse. However, using the MSP432 launch board, the majority of those issues are prevented. The only other item of note is the speakers used with this project. Those are not provided in our project - by design. This product is intended to be used with any set of standalone speakers which will take input from a 3.5mm jack. This is ostensibly a feature of our system and design - but has a downside. Our output can vary from 0-3.3V at max, which from our testing for our previous project was much larger than the input that a pair of earbuds expected. If a user has this system on full volume, that might damage their audio equipment or possibly cause minor hearing damage due to the sound caused by the 3.3V signal.

Reed

With the rise of electronically produced music, the role that intellectual property plays has changed greatly. New musical styles heavily focusing on sampling and remixing past musical pieces. As designers of the sequencer, we must take into account the possibility that users of our product will use it to violate intellectual property laws. Fortunately, right now our sequencer only plays square wave notes, and is unable to mimic more complex musical concepts like chords. It would be very difficult to recreate copyrighted music on our sequencer as a result. However, if the sequencer were more complex, these issues would be of concern to us. Much like copyrighted music can be played on any musical instrument (e.g. a guitar), it can be played on the sequencer. We cannot control what the end user uses the sequencer to create, but if we were to ship this out as a product, we could include information about music copyrights and how the user can use the sequencer without violating these laws.