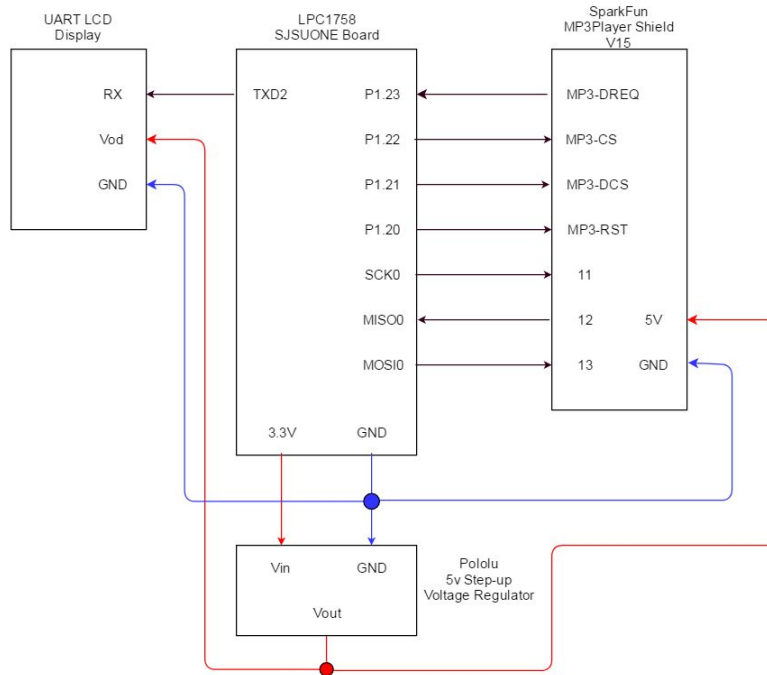


# CMPE146 MP3 Project

Ameer Ali, Ajay Curam, Andrew Hu, Ryan  
Will



# System diagram and Hardware



## Core Components Used:

SJSUOne Board (LPC 1758  $\mu$ C)

Onboard SD Card interface (SSP1)

Sparkfun MP3Player Shield V15 -  
VS1053B MP3 Audio Decoder (SSP0)

Sparkfun LCD- 09395 Display Screen  
with onboard PIC 16F88 (UART2)

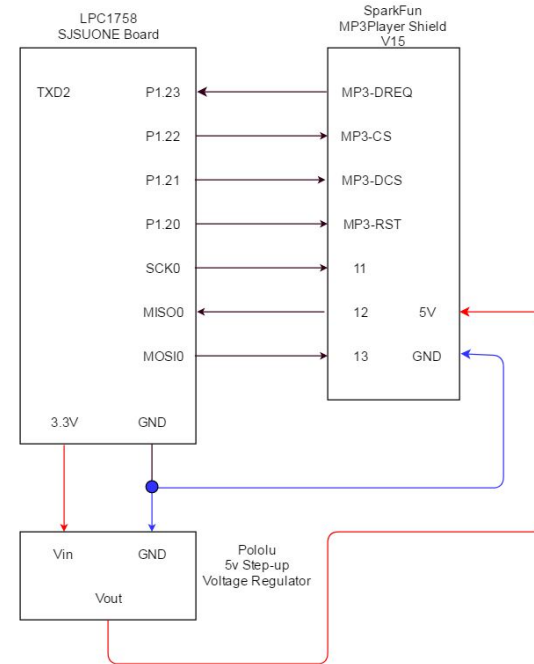
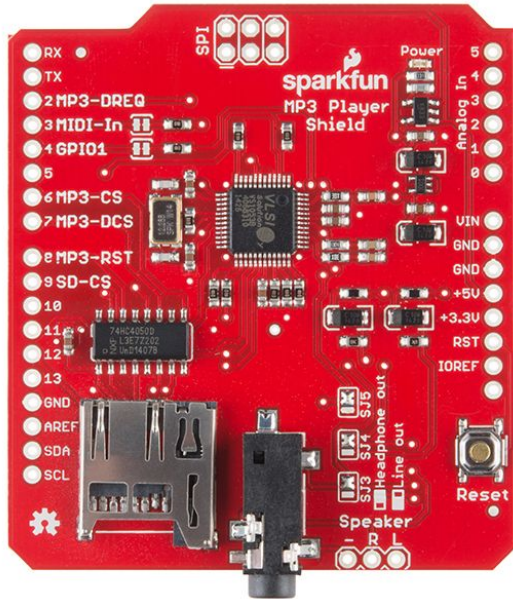
Pololu 5V Step-Up Regulator



# MP3 File Transfer Algorithm

- Scans through all files in the SD card path directory (1:) using the `f_opendir` and `f_readdir` libraries and puts all relevant mp3 file paths into a 2D character array that is generated by the SJOOne board on boot. LFname - long file name.
- After these files are found, we `f_mount` once, then `f_open` and `f_read` each file within our character array of file paths that point directly to our source mp3 files.
- Utilizing `f_read`, we constantly read 512 bytes of the file into a buffer that is 512 bytes. This buffer is then sent to the MP3Queue utilizing the `xQueueSend` API for further processing by the `MP3DecoderReadDataTask`.
- Our MP3 Player can support any SD card with MP3 Files on the root directory.

# Audio Decoder Diagram





# Audio Decoder Hardware - Setup

Pin 2 MP3 - Dreq - **Data Request Pin**, Interrupt based pin that tells the SJSUOne board if the Decoder's 2048-byte FIFO can take any more data. If high 32 bytes of SDI data or one SCI command can be sent

Pin 6 MP3-CS - **Chip Select** Active low Pin that tells decoder when data is being sent to it and for the purposes of this project set low when an address is being written to

Pin 7 MP3-DCS - **Data Chip Select**, Active Low Pin that tells decoder when music data is being sent and for the purpose of this project set low when you send music data to the decoder

Pin 8 MP3-RST - **Reset Pin** Reset Input for the decoder, set high initially



# Audio Decoder Hardware - MicroController connections

Communications between Microcontroller and Decoder happened over SPI using SSP0 on the board specifically.

Pin 11 - MOSI

Pin 12 - MISO

Pin 13 - SCLK



# Audio Decoder- Write To function

- Poll the Dreq signal to make sure decoder has room to take Data and if high begin transmission
- Set CS low and begin sending data starting with an opcode (for write that is 0x02), then the 8 bit address, and then finally the 16 bit data in two 8-bit chunks starting with the higher byte then the lower byte
- Set CS back up to high when done



# Audio Decoder - Read From function

- Give 8 bit address as input for this function as the address you wish to read from
- Wait for Dreq to go high and then set CS to low as well and begin transmission
- Begin with the opcode for reading (0x03), then the 8 bit address after which any data sent will be ignored
- You get the first high byte of data by sending a dummy byte, then you check if DREQ is still high and then get the other lower byte by sending another dummy byte.
- You get the bytes and OR them together to get the 16 bit data within the register



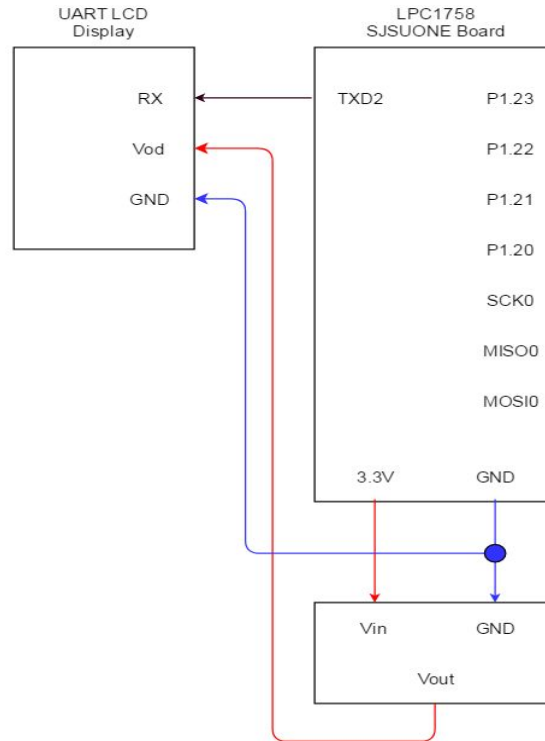


# Audio Decoder - Playing MP3 File

- In order to play an MP3 File, DREQ pin is polled until it sends a high signal, which means that there is space on the internal 2048 byte FIFO on the Audio Decoder to store MP3 data.
- The Data Chip Select must then be set to low in order to enable the decoder to send MP3 data to the FIFO.
- Only 32 bytes are sent to the decoder at a time because the DREQ pin needs to be checked if it sends a high signal after every 32 bytes. So out of the 512 bytes that are sent over the MP3FileReadTask, 32 bytes chunks are read at a time.

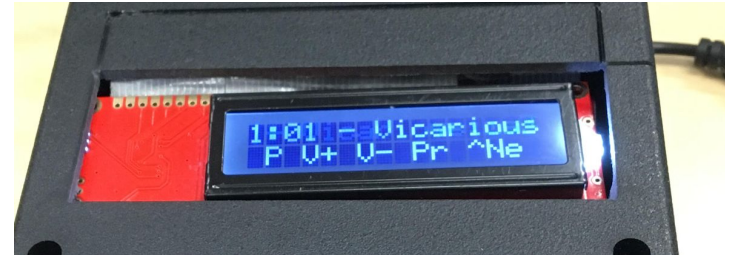


# LCD Display Diagram



# Display Hardware description

- Sparkfun LCD-09395  
(16x2)
- 3.3V Serial, UART 9600bps  
default baud
- 8 bits, 1 stop bit, no parity  
bit





# Display Algorithm explained

- Task is added that initializes the LCD display and updates it every second
- The first row of the LCD displays the path to the song. To accommodate 16+ character strings, we created a **scrolling function** that takes in the path as a character array.
- This function checks if there is a readable character in the *next offscreen element*. If there is, the buffer *shifts* and prints the new 16 characters. If there isn't the buffer resets to the beginning 16 characters.
- The second row of the LCD displays the menu. Depending on a *global menu select variable*, the LCD displays one of two menus.

TODO: ADD UML Diagram

# Control Instructions Explained

- Our MP3 Player has 5 main functionalities:
- Play/Pause Playback of MP3 File
- Volume Up/Down
- Previous/Next Track on SD Card.
- All of these are mapped to buttons on the SJOne Board.





# Live Demo