adafruit learning system

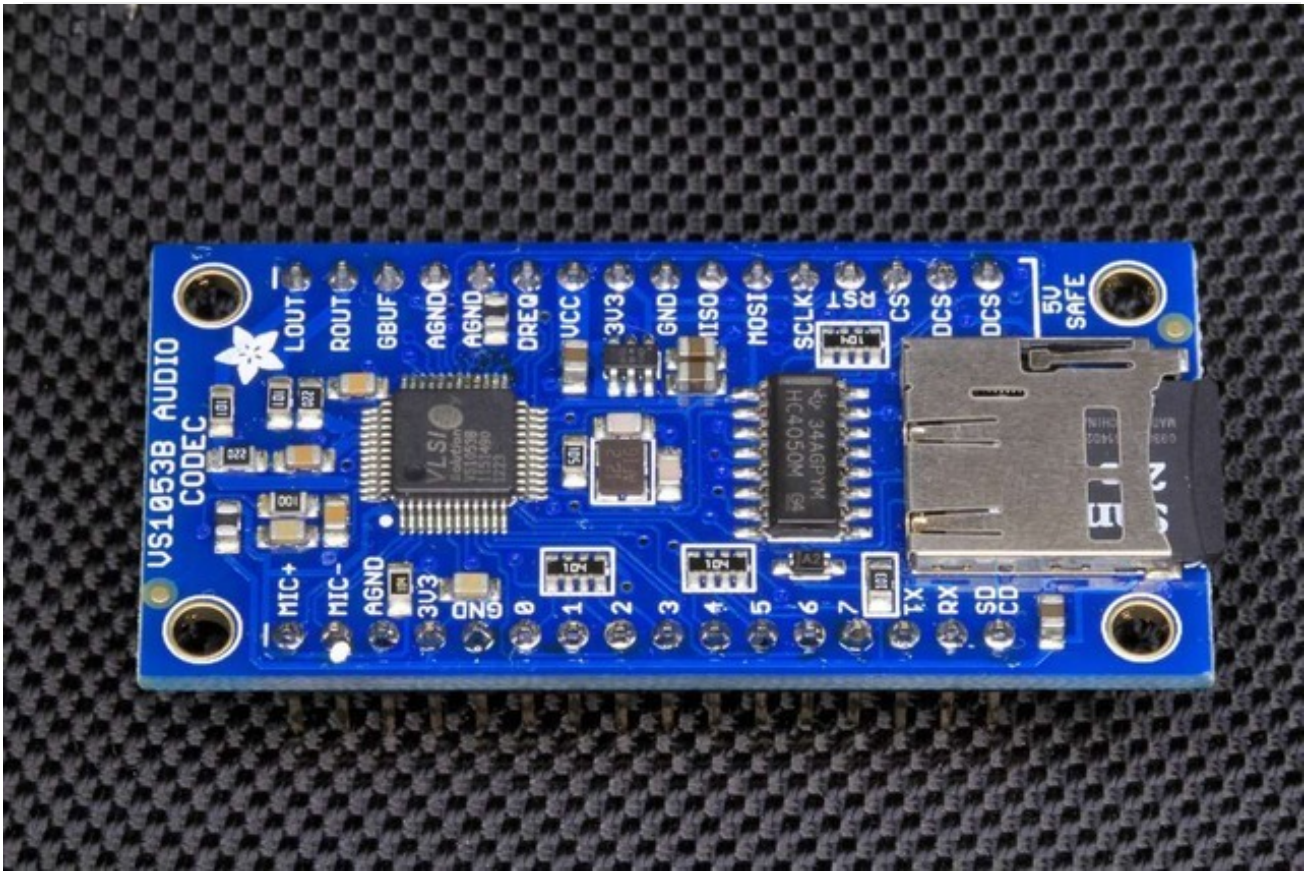
Adafruit VS1053 MP3/AAC/Ogg/MIDI/WAV Codec Breakout Tutorial

Created by Bill Earl

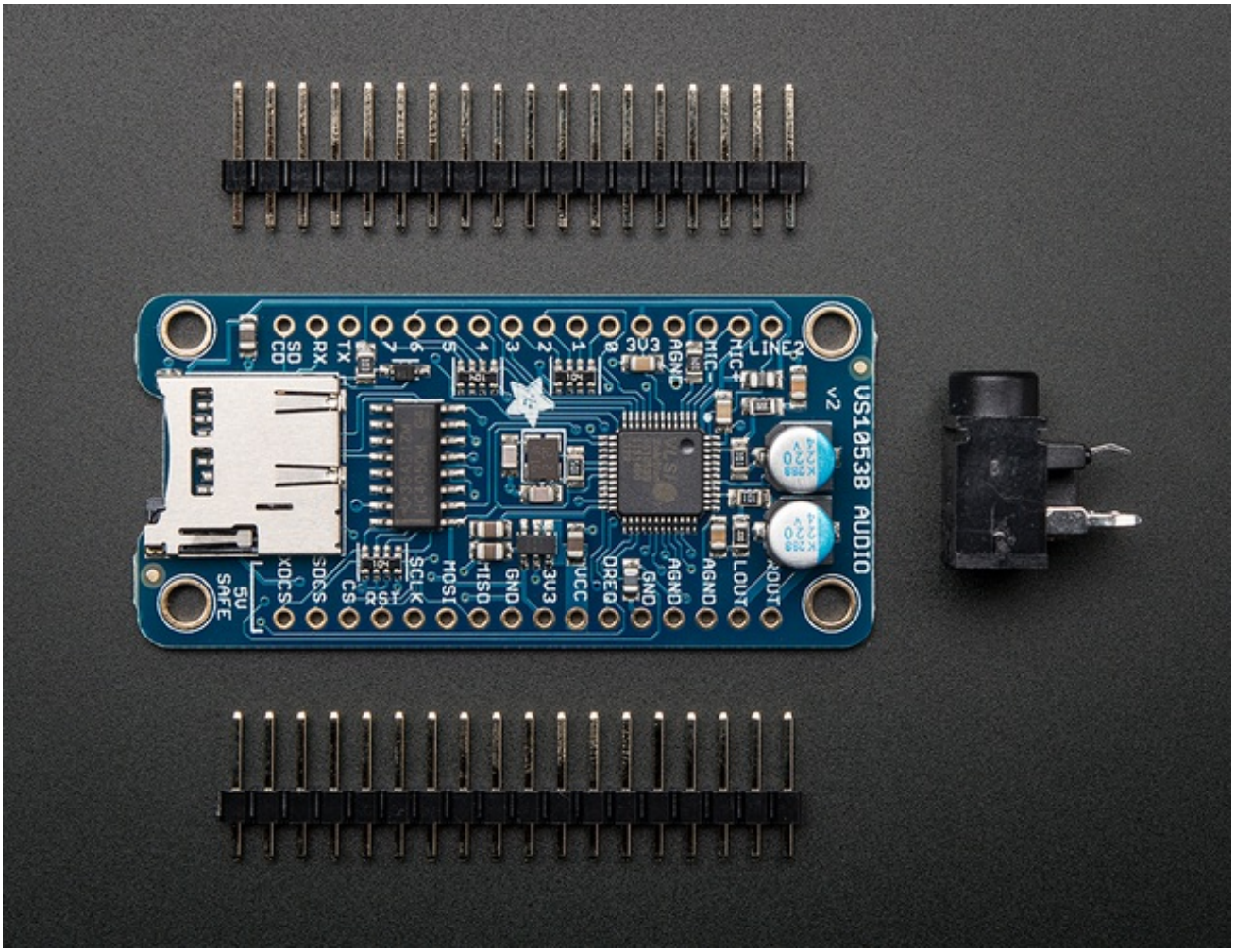
Last updated on 2014-05-16 04:15:17 PM EDT

Guide Contents

Overview



Above: v1.0 codec board. Below: v2.0 codec board



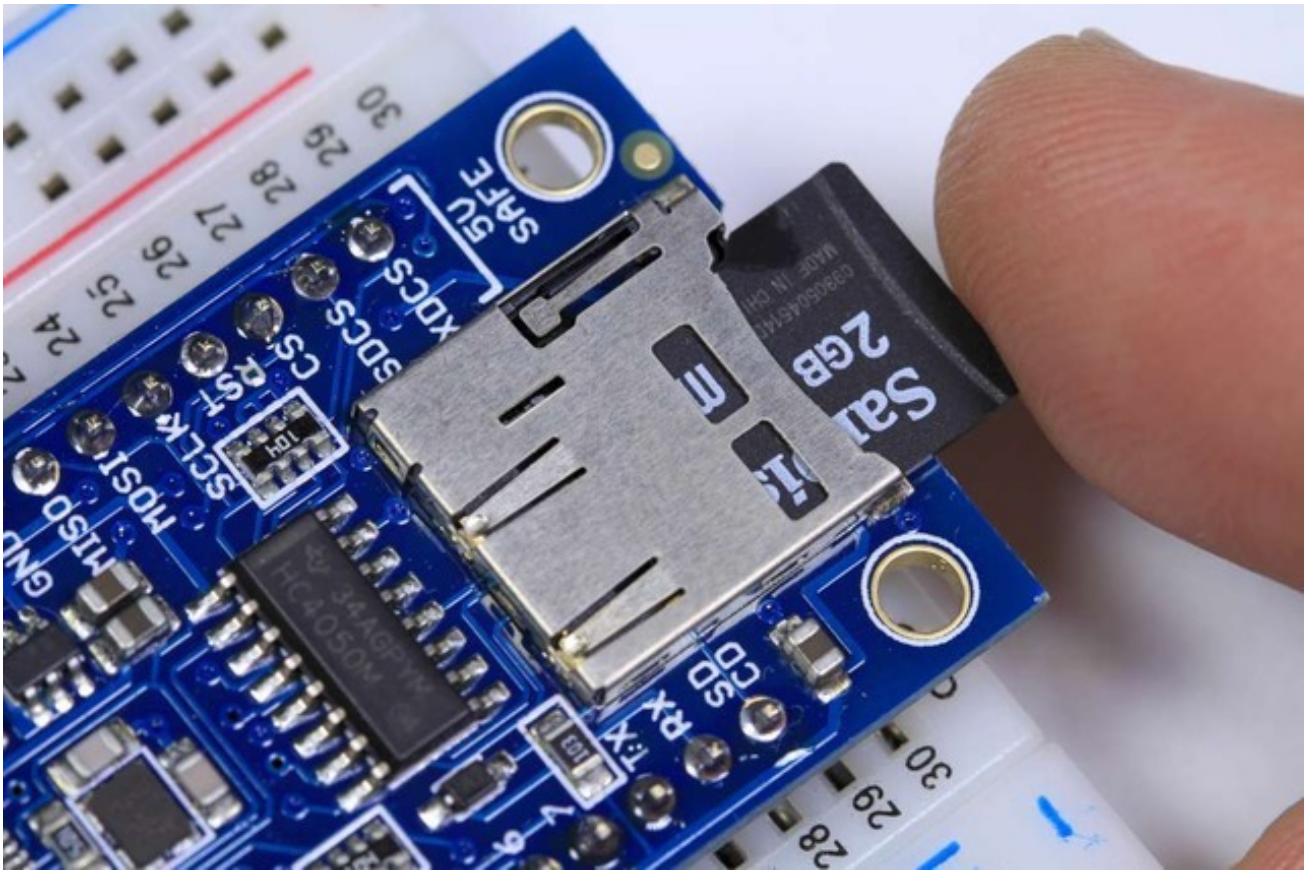
This breakout board is the ultimate companion for the VLSI VS1053B DSP codec chip. The VS1053 can decode a wide variety of audio formats such as MP3, AAC, Ogg Vorbis, WMA, MIDI, FLAC, WAV (PCM and ADPCM). It can also be used to record audio in both PCM (WAV) and compressed Ogg Vorbis. You can do all sorts of stuff with the audio as well such as adjusting bass, treble, and volume digitally. There are also 8 GPIO pins that can be used for stuff like lighting up small LEDs or reading buttons.

All this functionality is implemented in a light-weight SPI interface so nearly any microcontroller can play audio from an SD card. There's also a special MIDI mode that you can boot the chip into that will read 'classic' 31250Kbaud MIDI data on a UART pin and act like a synth/drum machine - there are dozens of built-in drum and sample effects! But the chip is a pain to solder, and needs a lot of extras. That's why we spun up the best breakout, perfect for use with an Arduino but also good for other microcontrollers that just don't have the computational power to decode MP3s.

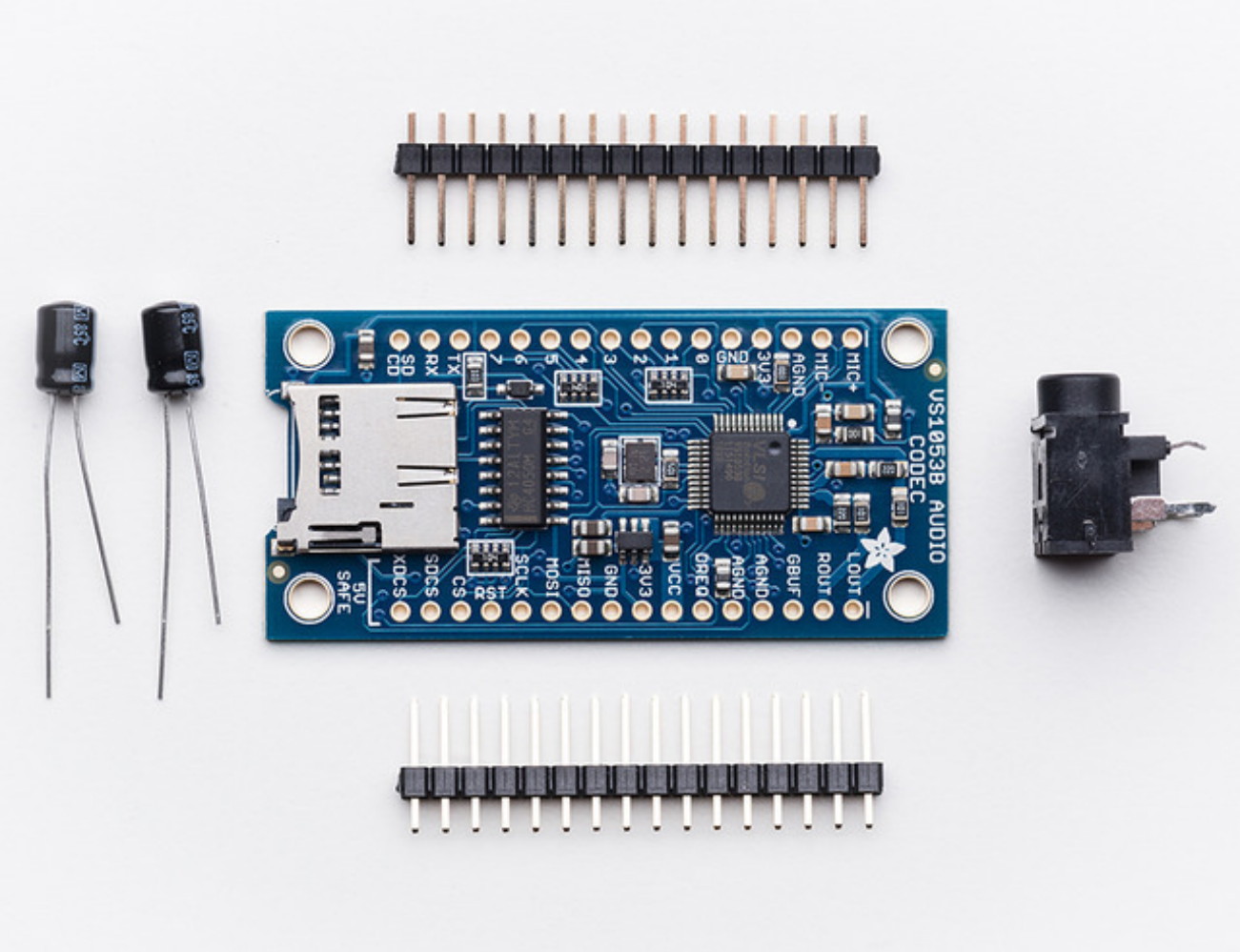
The breakout is slim enough to fit on a breadboard with 1 row of holes left over for wiring. There are 3.3v and 1.8v regulators onboard with ferrite beads and analog filtering for quality analog output. There's a microphone input port that you can wire up a line-in or mic to and record compressed audio. All 8 GPIO are broken out and they all have built in 100K pulldowns, simply connect your button from the GPIO pin to 3.3V for an active-high

connection. You'll likely want to play music from a microSD card so we added a holder on-board. And since we know so many of our customers use 5V microcontrollers like the Arduino, we made the interface pins all 5V compliant with level shifters so you can use the chip at 3V or 5V power/logic!

The breakout board comes fully assembled and tested. All v1.0 boards come with two 100uF electrolytic capacitors for line-level output coupling, v2.0 have the capacitors integrated into the PCB. All kits come with some 0.1" male header you can solder to the breakout so it plugs into a breadboard and a bonus stereo headphone jack that will be handy when you want to plug headphones in!

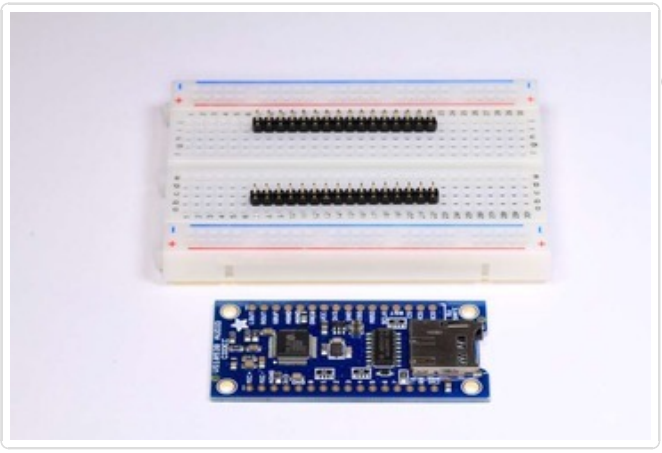


Assembly



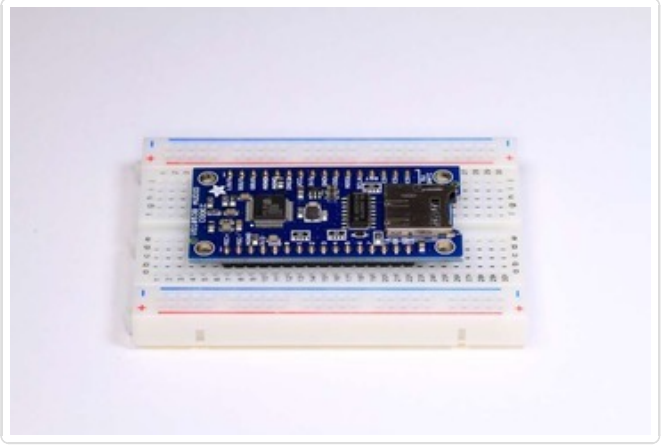
The board comes with all surface-mount components assembled and tested. For use on a breadboard, you will want to install the included header strips. This is a simple process and should only take a few minute

Assembly is the same for both v1 and v2 breakouts, even if the boards look a bit different!



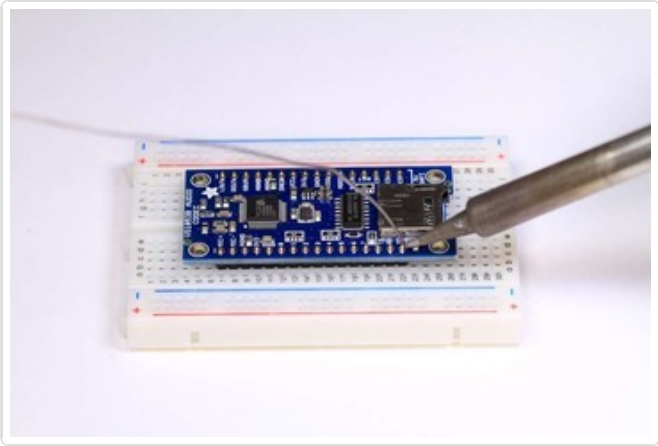
Prepare the Headers

Cut the header strips to length if necessary and place the long-pins-down in the breadboard.



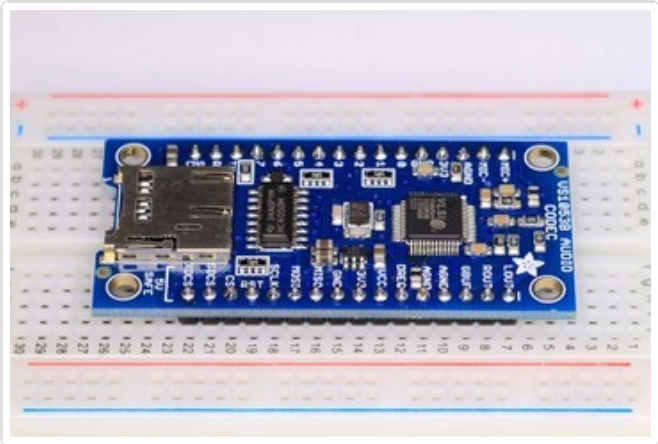
Position the Board

Place the breakout board over the header pins.



And Solder!

Solder every pin to assure good electrical contact.



Getting Started - VS1053 Software

To get started with the VS1053 breakout (any version) you will first need to download the Adafruit VS1053 Library:

Download Adafruit VS1053 Library

http://adafru.it/cDQ

Copy the folder inside the zip file to the Libraries folder inside your Arduino Sketchbook folder and re-name it to Adafruit_VS1053 For more details on how to install Arduino libraries, check out our detailed tutorial using the link below:

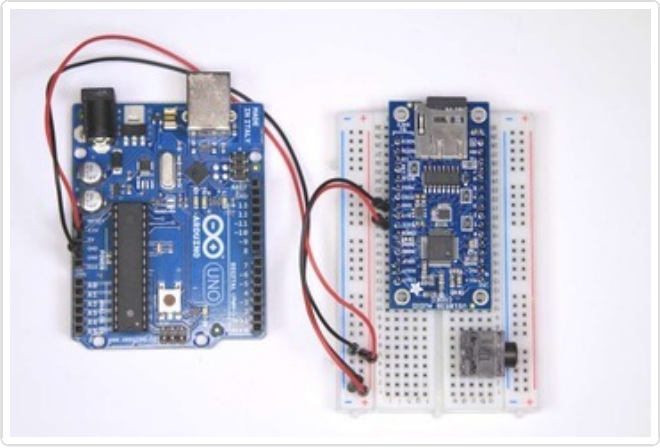
All About Arduino Libraries

http://adafru.it/aYM

Simple Audio Player Wiring

The VS1053 can be configured as a simple audio player under control of the Arduino. The Arduino reads data from the on-board SD card, then plays it back through the CODEC via the SPI interface.

These steps are identical for both v1 and v2 codec breakouts, even if the boards look a little bit different

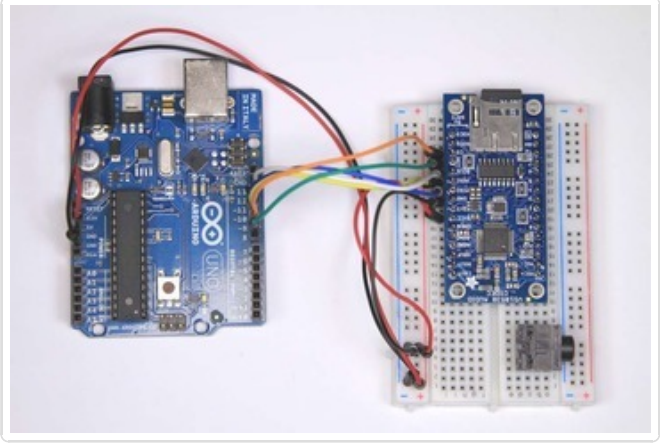


Prepare the breadboard

Place the VS1053 breakout on the breadboard. Center it so that there is one row of holes on each side.

Add the breakout friendly headphone jack and wire power and ground jumpers as shown.

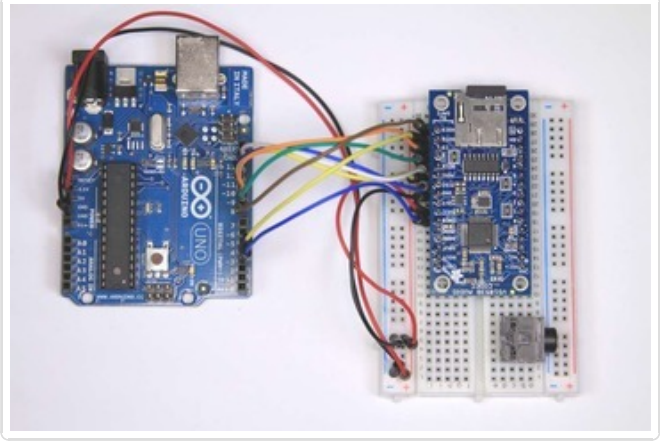
- **VCC -> 5v**
- **GND -> GND**



Connect SPI and Reset

Add jumpers for:

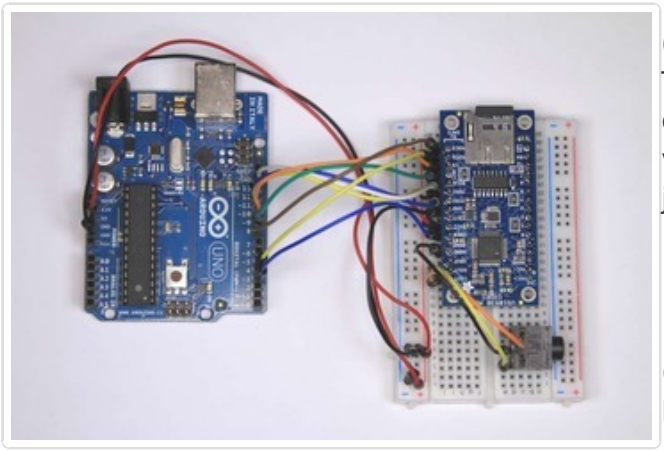
- **CLK -> Arduino #13 (Mega 52)**
- **MISO -> Arduino #12 (Mega 50)**
- **MOSI -> Arduino #11 (Mega 51)**
- **CS -> Arduino #10**
- **RST -> Arduino #9**



Connect the rest of the digital signals

Add more jumpers for:

- **XDCS-> Arduino #8**
- **SDCS -> Arduino #4**
- **DREQ -> Arduino #3**



Connect the Headphone Jack

This step will be a little bit different depending on which version you have. If your breakout is v2 and has the round silver capacitors, add jumpers for:

- **AGND -> Center 'ground' Pin**
- **LOUT -> Left Pin**
- **ROUT -> Right Pin**

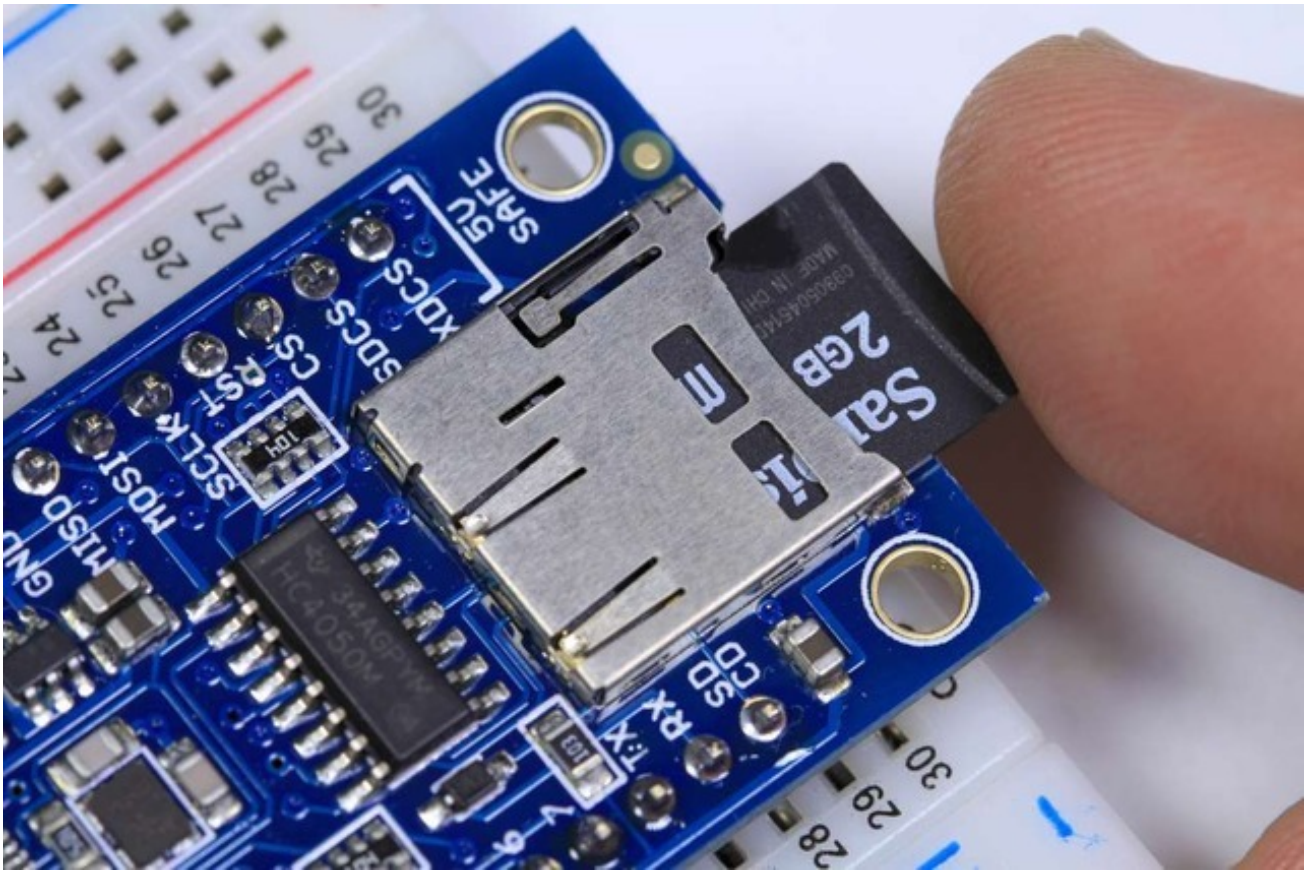
(these are in the corner in a row)
If your breakout is v1, the pins will be labeled

- **GBUF -> Center 'ground' Pin**
- **LOUT -> Left Pin**
- **ROUT -> Right Pin**

For v1, do not use this wiring for powered speakers or a line-in connection to a computer/stereo. **Only for headphones!** For v2, you can use this wiring for any type of connection without worry.

Load some MP3 files

Copy 2 MP3 files to a micro SD card and name them **track001.mp3** and **track002.mp3** (this is just for the test, you can re-name them later). Then install the card in the slot on the VS1053 breakout.



Make sure you have a good quality SD card, some cheap SD cards won't work, causing confusion! Especially 'non-brand' knockoffs.

Simple Audio Player Sketch

Connect the Arduino to your computer with a USB cable and plug your headphones into the headphone jack. Select **File->Examples->Adafruit_VS1053_Codec->player_simple** to load the example code.
You should hear your MP3 files in the headphones.

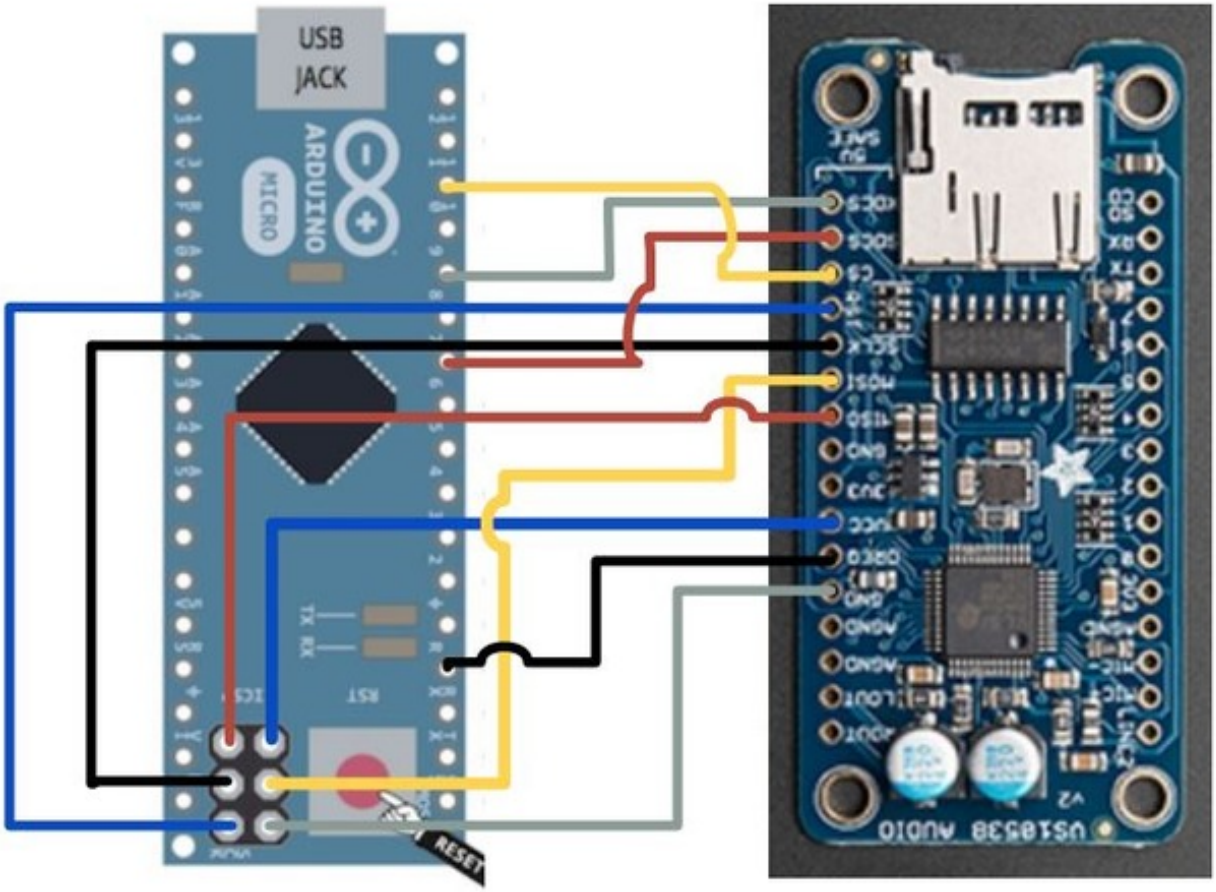
You can also run **File->Examples->Adafruit_VS1053_Codec->player_interrupts.** This example demonstrates playing files in the background using interrupts. This allows you to do other things in your sketch while the music plays!

With both examples, you can change the file names in the code to play different files - or even extend the code to search the SD card for files to play.

For the Micro:

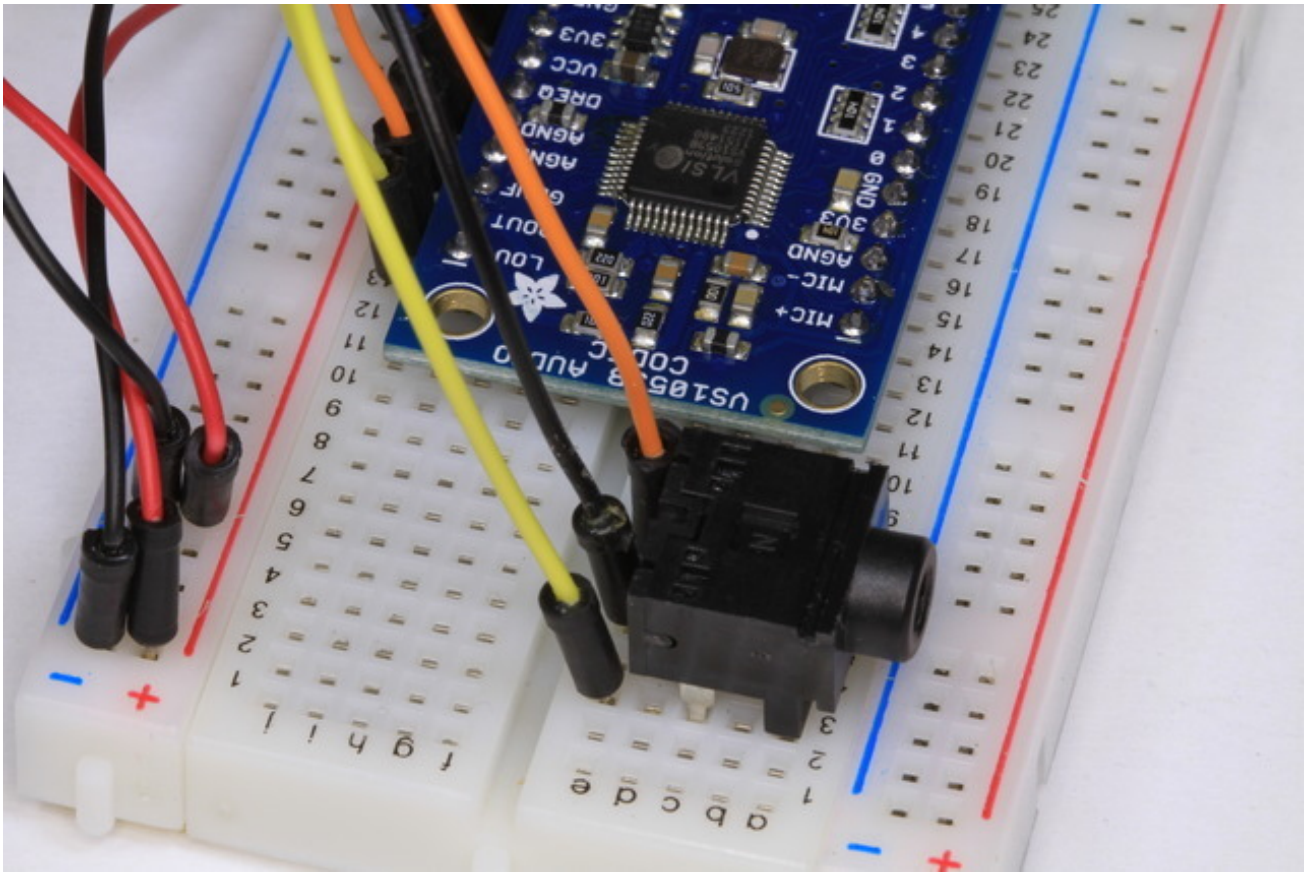
Frank Cohen has submitted a wiring diagram for the FilePlayer example on a Micro. See his code posted here as well: http://votsh.files.wordpress.com/2014/02/vs1053-arduino-micro-connections.pdf

Arduino Micro to Adafruit VS1053 Codec Connections
Chart by Frank Cohen, fcohen@votsh.com
Distributed Under Creative Commons License
February 24, 2014

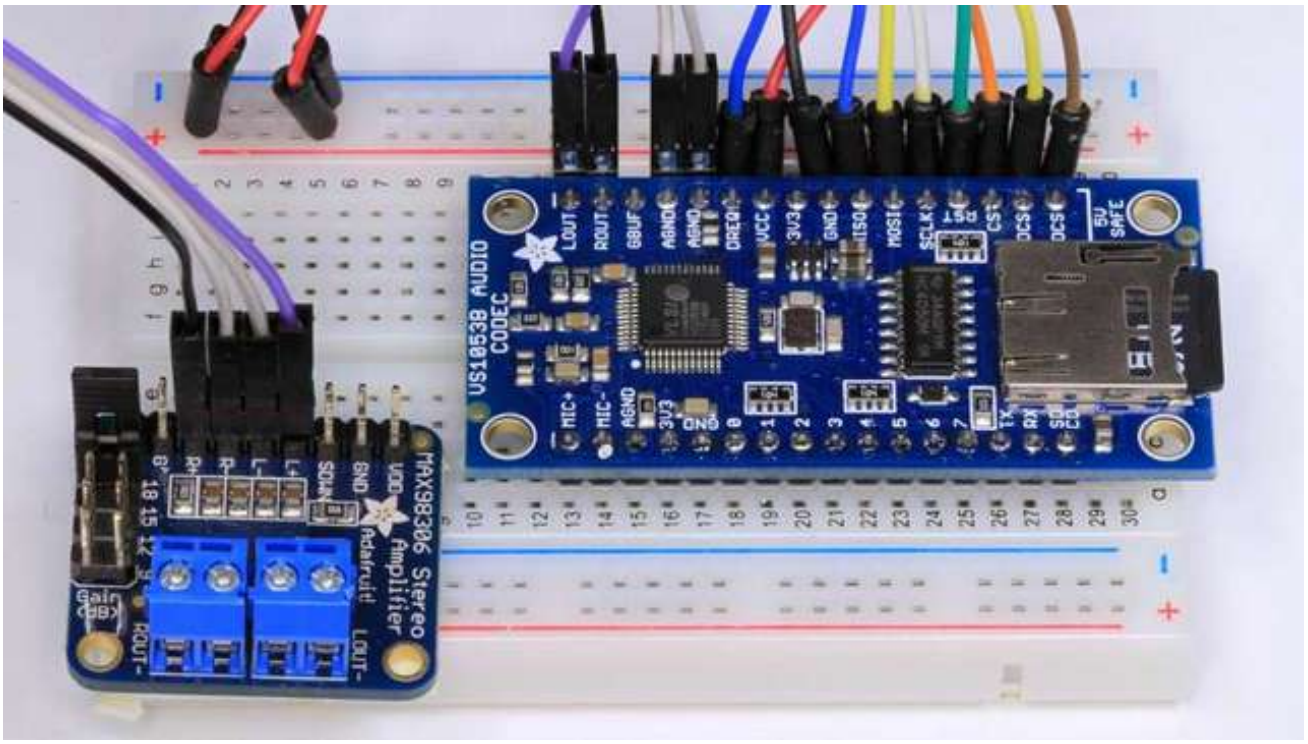


Audio Connections

Both the VS1053 version 1 (shown above, does not have big round silver capacitors at the top) **and** the version 2 (has a v2 label, has two big round silver caps) can drive a pair of headphones directly. Just connect the LOUT and ROUT to the left and right pins of the headphone jack. The center pin should be connected to GBUF.



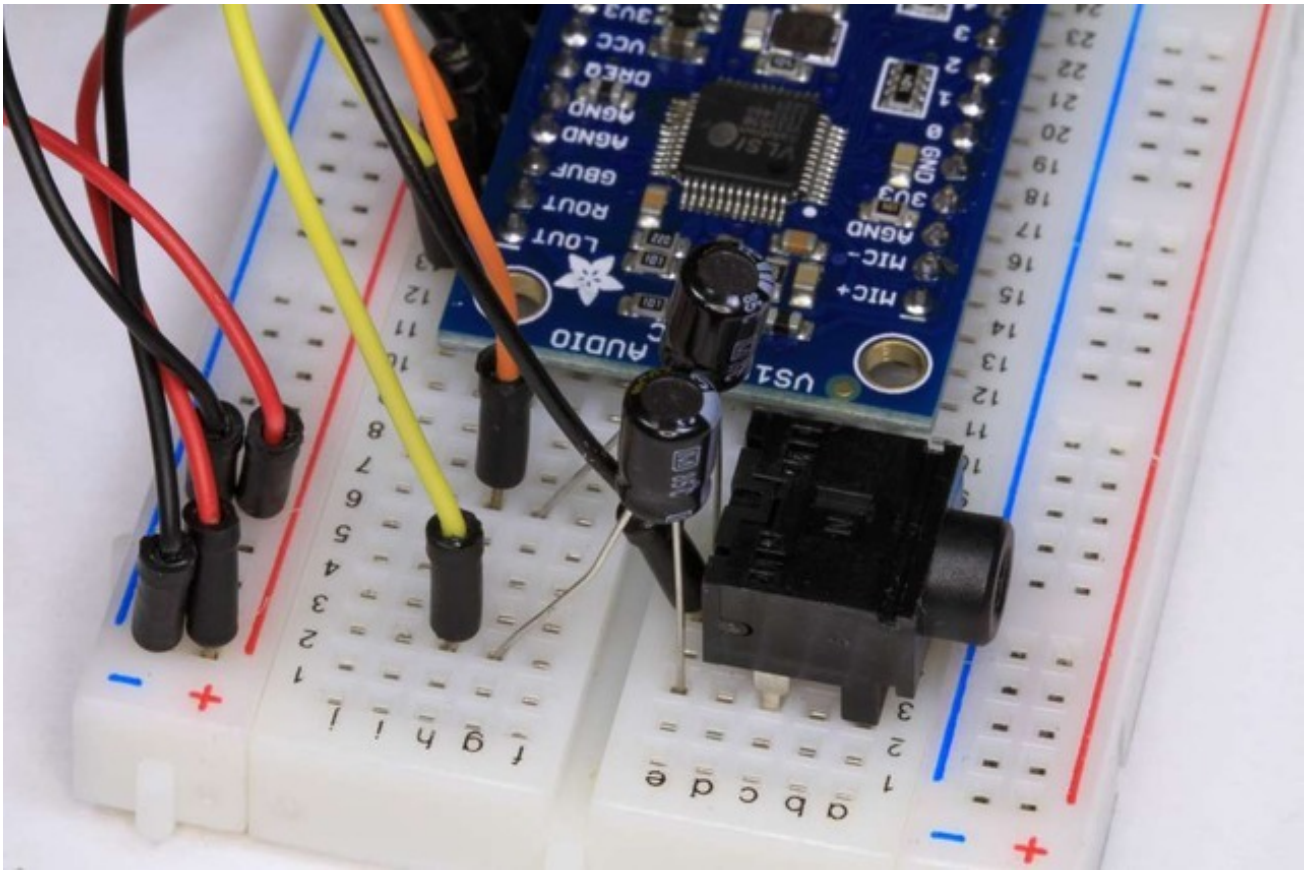
Both v1 and v2 can also directly drive audio devices with differential inputs such as our 3.7 Watt Stereo Class D Amplifier. Connect LOUT and ROUT to the L+ and R+ pins of the amplifier. Connect L- and R- to the AGND pins.



If you have a v1 breakout, you can connect to powered speakers/stereo BUT YOU MUST USE THE INCLUDED CAPACITORS!

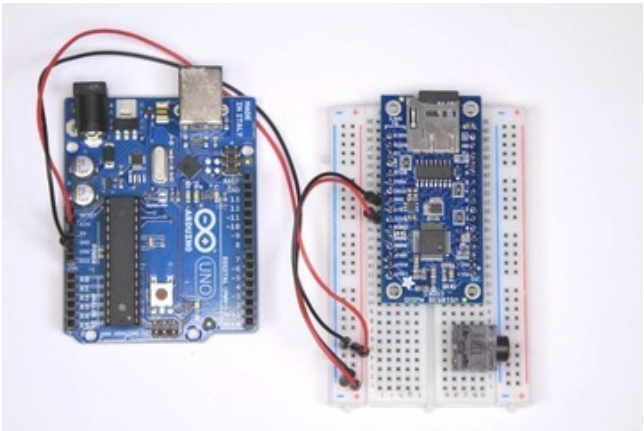
For audio devices requiring a 'line level' input, use the included capacitors. These are polarized electrolytic capacitors, so it matters which way you connect them. Connect the positive (longer) leads to the LOUT and ROUT pins, and the negative leads to the L & R pins of the headphone jack or L and R inputs of the audio device. The common (center) pin of the audio jack should be connected to the AGND pin.

If you have a v2 breakout, these capacitors can be skipped, they are included onto the breakout board itself.



MIDI Connections

With a few jumper connections, the board will boot up in MIDI mode that will read 'classic' 31250Kbaud MIDI data on a UART pin and act like a synth/drum machine - there are dozens of built-in drum and sample effects.

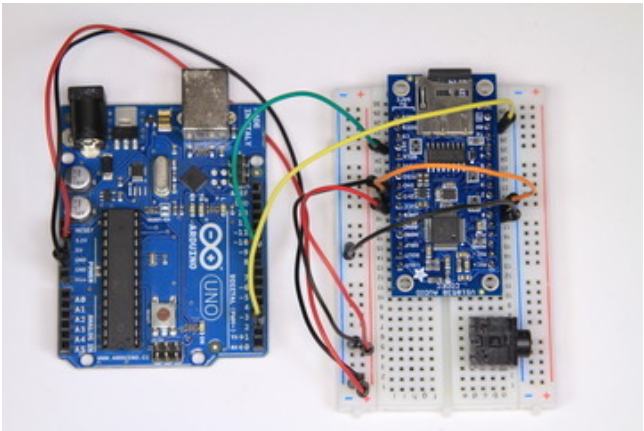


Prepare the breadboard

Place the VS1053 breakout on the breadboard. Center it so that there is one row of holes on each side.

Add the breakout friendly headphone jack and wire power and ground jumpers as shown.

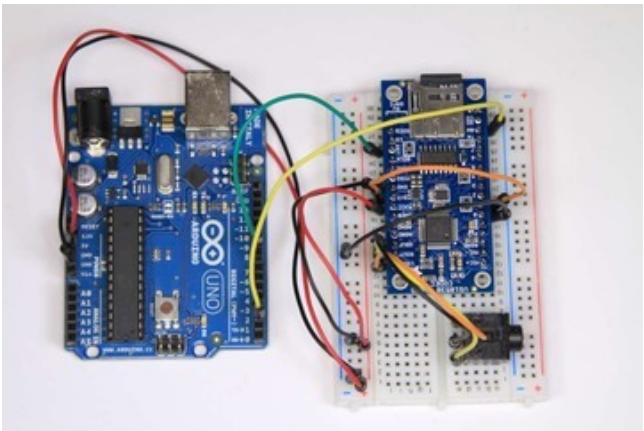
- **VCC -> 5v**
- **GND -> GND**



Configure for MIDI operation

Add jumpers as follows:

- **GPIO-0 -> GND**
- **GPIO-1 -> 3.3v**
- **Rx -> Arduino Digital #2**
- **RST -> Arduino Digital #9**



Connect the Headphone Jack

Same as the earlier tests, add jumpers for:

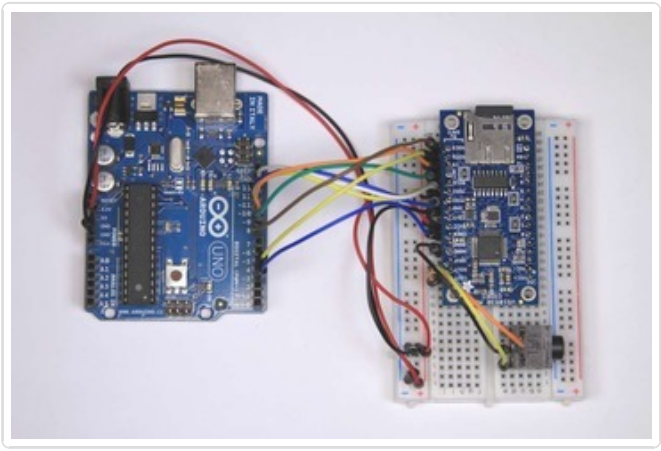
- **AGND (v2) or GBUF (v1) -> Center Pin**
- **LOUT -> Left Pin**
- **ROUT -> Right Pin**

Midi Example Sketch

Connect the Arduino to your computer with a USB cable and plug your headphones into the headphone jack. Select **File->Examples->Adafruit_VS1053_Codec->player_miditest** to load the example code for MIDI operation.. You should hear a repeating series of ascending tones from the MIDI player.

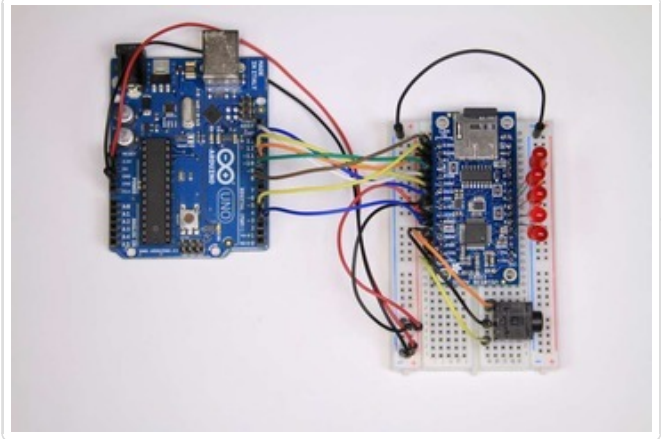
GPIO

The VS1053 has 8 GPIO pins that can be read and written via the library. The player_gpiotest sketch demonstrates how to do this.



Start with the basic wiring

Use the Simple Audio Player Wiring from the link on the left.



Add some LEDs

Add a jumper from the ground bus on the left to the ground bus on the right of the breadboard.

Then add some LEDs from the GPIO pins to ground. Be sure to connect the long lead (anode) to the GPOI pin and the short lead (cathode) to ground.

What? No current limiting resistors?

Strictly speaking, best practice is to use a current limiting resistor when driving an LED from a GPIO pin. In this case, the example sketch pulses each led only briefly, so there is no danger of damage. For more general use, you should select a resistor appropriate for the led you are using. See All About LEDs (http://adafru.it/clH) for more detail.

Run the player_gpiotest sketch

Connect the Arduino to your computer with a USB cable. Select **File->Examples->Adafruit_VS1053->player_gpiotest** to load the example code.

If you have headphones, you will hear a beep at the start to indicate that the sketch is running. Then you should see the LEDs flashed in sequence.

If you open the Serial Monitor, you can see the values that are written to and read from each GPIO pin.

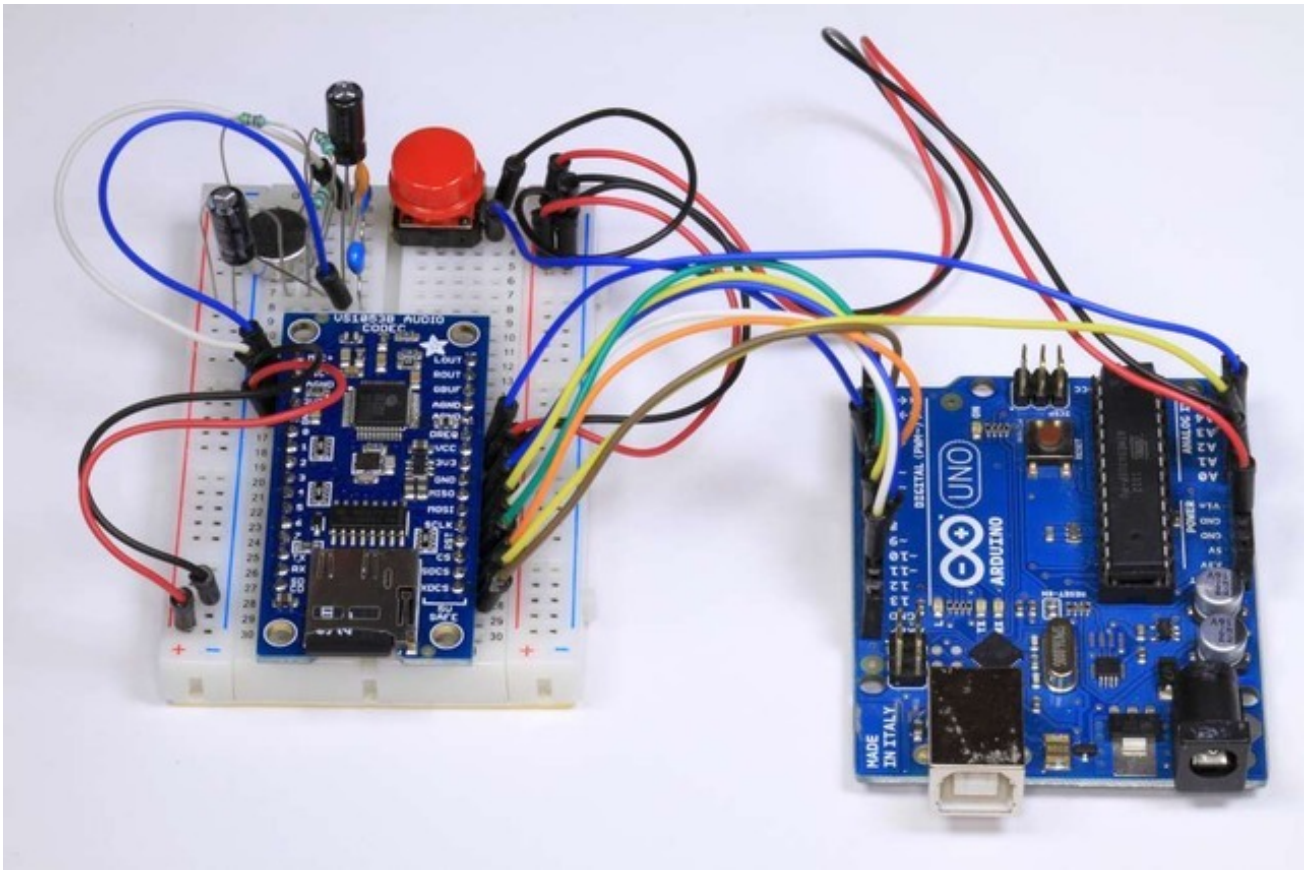
Note that the GPIO pins are NOT 5v safe. Do not connect them to voltages greater than 3.3v.

Ogg Recorder

(Nipper photo by Bevery & Pack - Creative Commons Share Alike)

Recording with the VS1053

The record_ogg example sketch turns your VS1053 breakout into a recording device that generates OGG encoded files on the SD card in real-time.



Wiring

Wiring for the record_ogg sketch is as follows:

Power and Ground:

- **VCC -> 5v**
- **GND -> GND**

Basic SPI connections:

- **SCLK -> Arduino #13**
- **MOSI -> Arduino #11**
- **MISO -> Arduino #12**
- **CS -> Arduino #10**

Additional Control Signals:

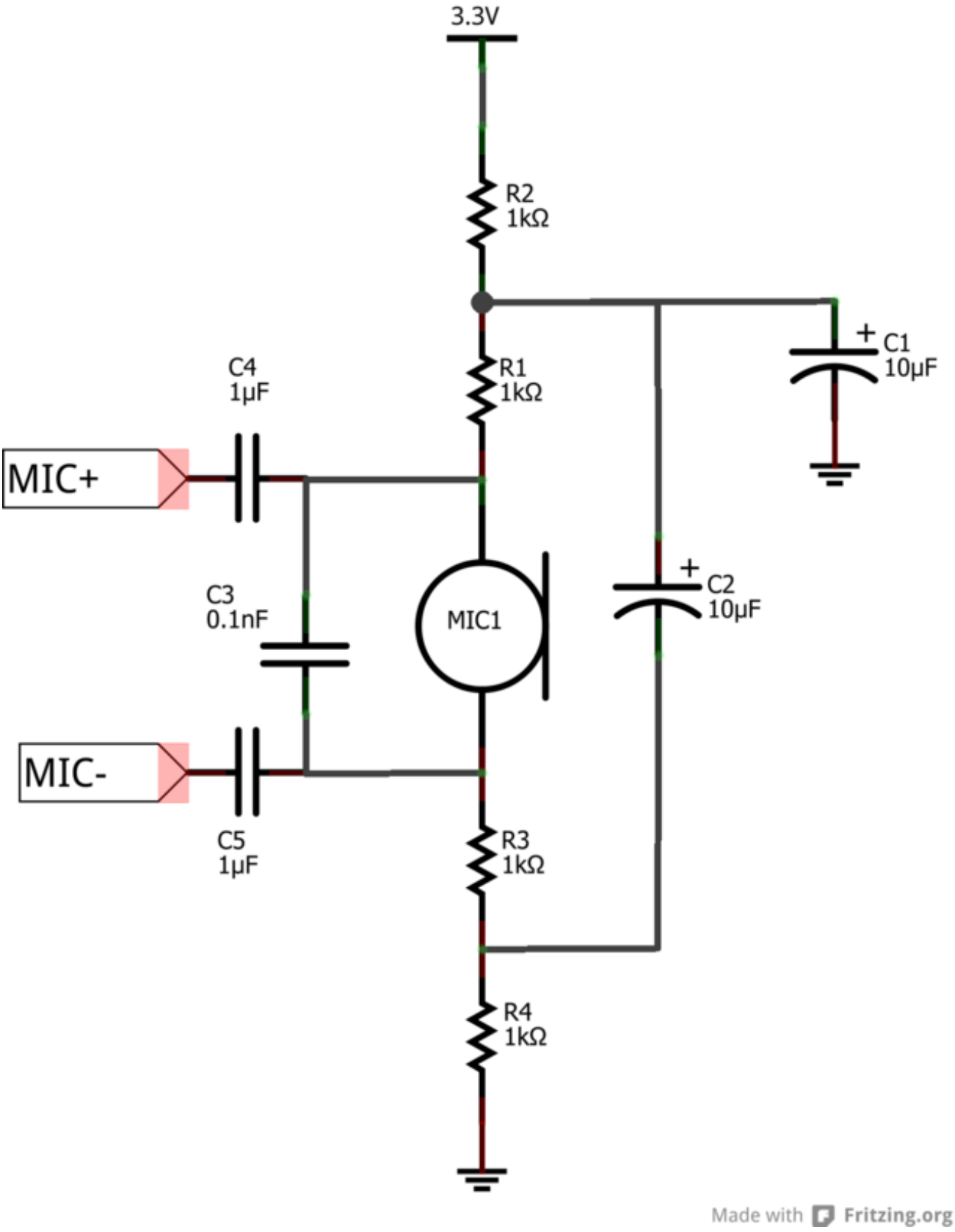
- **SDCS -> Arduino A0**
- **XDCS -> Arduino #8**
- **RST -> Arduino #9**
- **DREQ -> Arduino A1**

Start/Stop Button (Momentary):

- **from Arduino #7 to GND**

Electret Microphone Circuit:

The Microphone circuit is derived from the reference design shown in Figure 3 on Page 13 of the VS1053B Datasheet (http://adafru.it/cll). Note that the ground connections shown are to analog ground (AGND).



A note about microphone circuits:

Microphones, by their nature, are very sensitive devices. They are prone to picking up all sorts of noises (both electrical and acoustic) in addition to what you are trying to record. A microphone circuit built on a breadboard is not likely to sound good at all. For good sound, you will want to review the spec sheet recommendations and build this on a good quality circuit board.

Recording Sketch:

Connect the Arduino to your computer with a USB cable and plug your headphones into the headphone jack. Select **File->Examples->Adafruit_VS1053_Codec->record_ogg** to load the example code.

Plug-Ins

The record_ogg example sketch demonstrates a powerful feature of the VS1053 breakout: Plug-ins. OGG recording capability is not supported natively in the chip itself, but through a plug-in code module. A binary file containing the code image of the plug-in is loaded from the SD card at startup. Once loaded, the VS1053 becomes an OGG recording device.

To Record:

Press the start/stop button to start the recording. Press once again to stop. Each recording will be stored in a separate .OGG file on the SD card.

To Playback:

Use any OGG capable audio player on your computer - or playback through the VS1053 using the Simple Audio Player configuration (see link at left).

Library Reference

class Adafruit_VS1053_FilePlayer

The Adafruit_VS1053_FilePlayer class is derived from the Adafruit_VS1053 class and provides high level functions for playing files stored on the VS1053 breakout SC Card reader.

Public Methods:

Adafruit_VS1053_FilePlayer (uint8_t mosi, uint8_t miso, uint8_t clk, uint8_t rst, uint8_t cs, uint8_t dcs, uint8_t dreq, uint8_t cardCS) - Software SPI constructor. Uses Software SPI, so you must specify all SPI pins.

Adafruit_VS1053_FilePlayer (uint8_t rst, uint8_t cs, uint8_t dcs, uint8_t dreq, uint8_t cardCS) - Hardware SPI constructor. Uses Hardware SPI and assumes the default SPI pins.

boolean begin(void) - Initialize communication and reset the chip.

boolean useInterrupt(uint8_t type) - Specifies the interrupt to use for interrupt-driven playback. Valid arguments are:

- VS1053_FILEPLAYER_TIMER0_INT
- VS1053_FILEPLAYER_PIN_INT

boolean startPlayingFile(char *trackname) - Begin playing the specified file from the SD card using interrupt-driven playback. This allows your program to perform other tasks as the file is playing.

boolean playFullFile(char *trackname) - Play the complete file. This function will not return until the playback is complete.

Public Member Variables:

File currentTrack - File currently being played

boolean playingMusic - True if playback in progress

class Adafruit_VS1053

The Adafruit_VS1053 class implements an interface to the basic VS1053 functionality. For more detail on the operation of the VS1053 chip, please refer to the documentation on the Downloads page (see the link to the left).

public Methods:

Adafruit_VS1053(uint8_t mosi, uint8_t miso, uint8_t clk, uint8_t rst, uint8_t cs, uint8_t dcs, uint8_t dreq) - Software SPI constructor - must specify all pins.

Adafruit_VS1053(uint8_t rst, uint8_t cs, uint8_t dcs, uint8_t dreq) - Hardware SPI constructor - assumes hardware SPI pins.

uint8_t begin(void) - Initialize SPI communication and (hard) reset the chip.

void reset(void) - Performs a hard reset of the chip.

void softReset(void) - Attempts a soft reset of the chip.

uint16_t sciRead(uint8_t addr) - Reads from the specified register on the chip.

void sciWrite(uint8_t addr, uint16_t data) - Writes to the specified register on the chip.

void sineTest(uint8_t n, uint16_t ms) - Generate a sine-wave test signal.

void spiwrite(uint8_t d) - Low-level SPI write operation.

uint8_t spiread(void) - Low-level SPI read operation.

uint16_t decodeTime(void) - Reads the DECODETIME register from the chip.

void setVolume(uint8_t left, uint8_t right) - Set the output volume for the chip.

void dumpRegs(void) - Prints the contents of the MODE, STATUS, CLOCKF and VOLUME registers.

void playData(uint8_t *buffer, uint8_t buffsiz) - Decode and play the contents of the supplied buffer.

boolean readyForData(void) - Test if ready for more data.

void applyPatch(const uint16_t *patch, uint16_t patchsize) - Apply a code patch (See datasheet for details).

uint16_t loadPlugin(char *fn) - Load the specified plug-in.

void GPIO_digitalWrite(uint8_t i, uint8_t val) - Write to a GPIO pin.

void GPIO_digitalWrite(uint8_t i) - Write to all 8 GPIO pins at once.

uint16_t GPIO_digitalRead(void) - Read all 8 GPIO pins at once.

boolean GPIO_digitalRead(uint8_t i) - Read a single GPIO pin.

void GPIO_pinMode(uint8_t i, uint8_t dir) - Set the Pin Mode (INPUT/OUTPUT) for a GPIO pin.

boolean prepareRecordOgg(char *plugin) - Initialize chip for OGG recording.

void startRecordOgg(boolean mic) - Start recording (mic = true for microphone input).

void stopRecordOgg(void) - Stop the recording.

uint16_t recordedWordsWaiting(void) - Returns the number of words recorded.

uint16_t recordedReadWord(void) - Reads the next word from the buffer of recorded words.

uint16_t recordedReadWord(void) - Reads the next word from the buffer of recorded words.

Downloads and Links

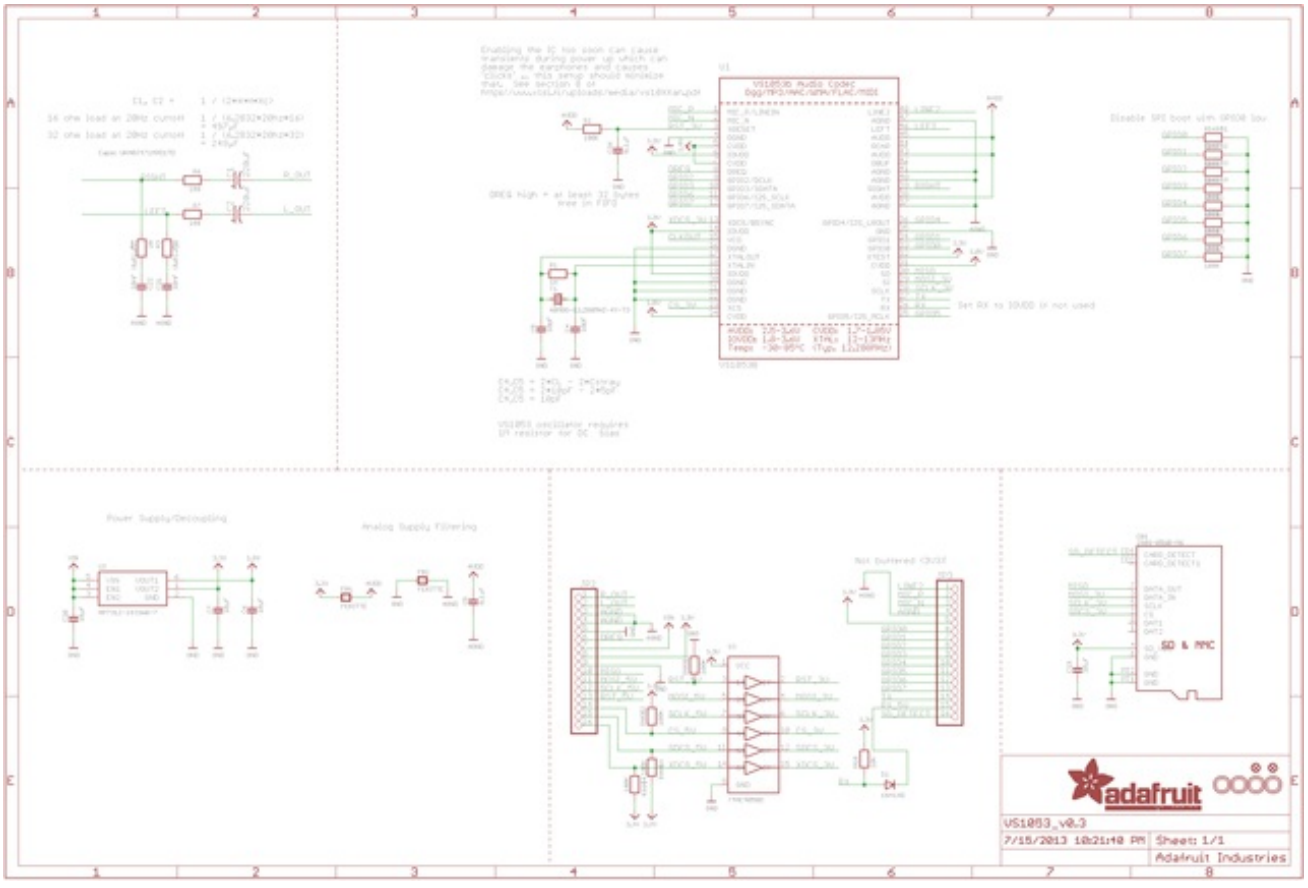
Library:

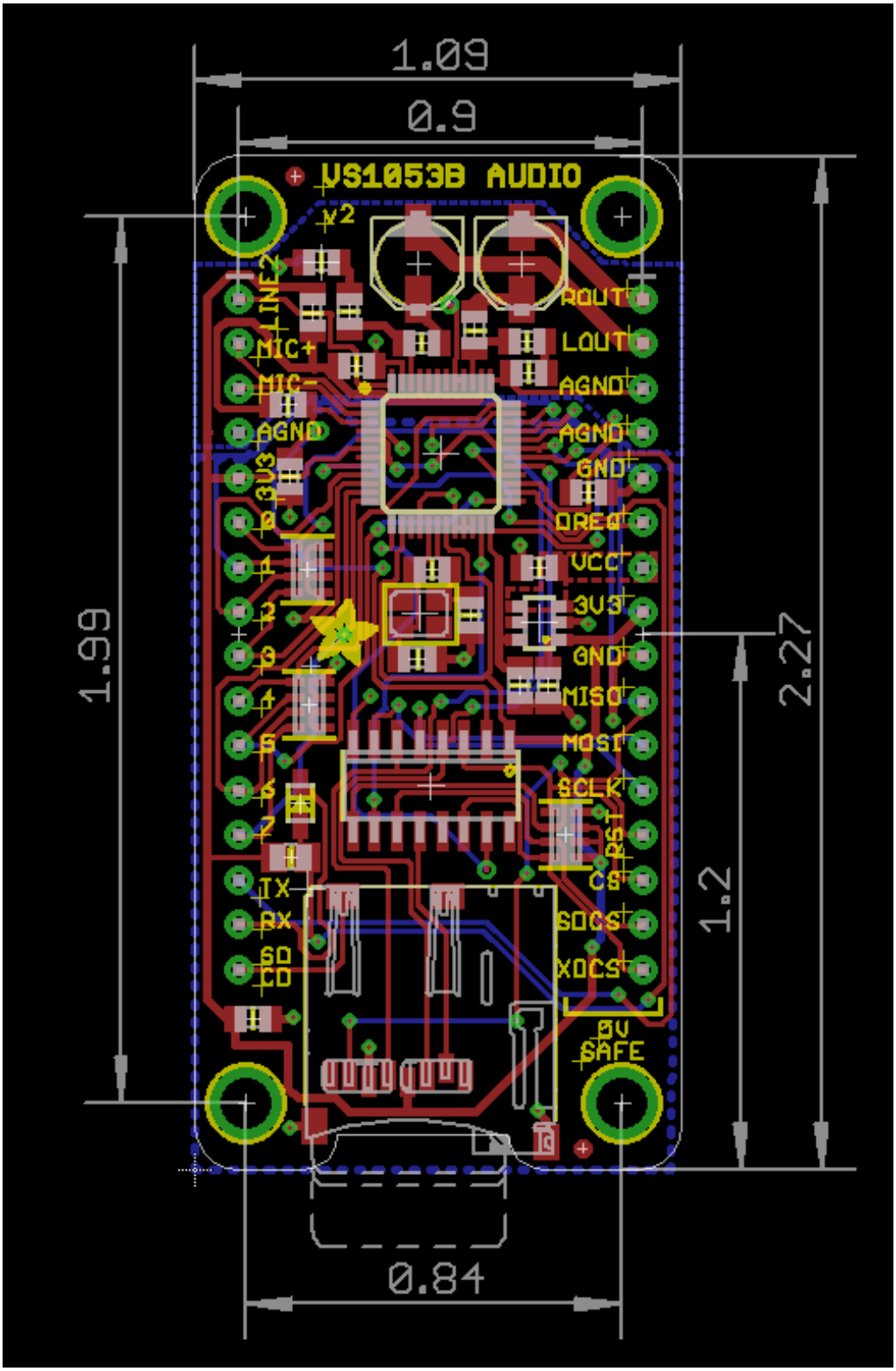
- Adafruit VS1053 Library (http://adafru.it/clE)

Technical Information:

- VS1053B (Codec chip) datasheet (http://adafru.it/clI)
- Details about the Ogg vorbis encoder/recorder (http://adafru.it/clJ)
- App note on connecting to analog outputs! (http://adafru.it/clK)
- Dimensions: 27.72mm / 1.09" x 58.08mm / 2.28"

Schematics for v2




1.09
0.9
VS1053B AUDIO
LINE2
MIC+
MIC-
AGND
3V3
ROUT
LOUT
AGND
AGND
GND
DREQ
VCC
3V3
GND
MISO
MOSI
SCLK
RST
CS
SDCS
XDCS
0V
SAFE
TX
RX
SD
1.99
2.27
1.2
0.84