Teensy Air Guitar ADXL343

Documentation

Technical Specifications

Code Base

Group 8

Milind Singh / N12392135/ ms10997

Table Of Contents

2.1 Theory of Operation

- The device is meant to emulate playback from an Acoustic Guitar , with capability to play the **I–V–vi–IV progression** chords. (Am , G , F , C)
- The device is composed of 3 sub-systems
  - ADXL 343 Accelerometer sub-system
  - Teensy Audio Shield sub-system
  - Touch Pad sub-system
- The systems interact with each other using the I^2C Protocol.
- ADXL343 is a device capable of throwing independent hardware-based event-driven interrupts through dedicated pins , these pins can be mapped to the Teensy 3.2 microcontroller Interrupt Controller to throw interrupts.
- The Teensy Audio Shield provides audio playback capability and routing through various internal and external buses , this device uses the I2S communication bus.
- Touch Pads are simple switches using INPUT_PULLUP mode on GPIO pins. When the pads are contacted together , it reads a digital HIGH , else LOW.
- The device operates on the interactions on these sub-systems , the capacitive pads act as a select line for 4 different chord combinations , ADXL343 is meant to be used as the guitar pick , it reads swing and converts it into real-time sound , Teensy Audio Shield is used to pass audio to a Mixer through RCA jack.

2.2 Initial Concept of Operation:

1. Two ADXL343 Accelerometers with 3 -Axis Detection capability will be used to map control to the Strumming hand and capacitive touch pads to be used as Fretting Keys.
2. The Z and X axis of ADXL343 map to upstrokes and downstrokes , two ADXL343 provide more accurate detection of TAP and ACTIVITY events by using interrupt inputs from both ADXL343 devices to detect events only when interrupt is fired on both ADXL343 MEMS devices.
3. The TAP event is defined as the sudden change in acceleration above the Threshold Value and for shorter duration than Duration specified in configuration. This can be configured as a way to detect strokes of hand by increasing threshold sufficiently. This INT will be mapped to INT1 pin of ADXL343 and attached to an Teensy Native Interrupt using attachInterrupt(digitalPintoInterrupt()) function of Arduino library.
4. The Teensy Audio Shield will be used to playback audio samples recorded from actual guitars corresponding to the input combination of chord and strum pattern. Teensy Audio Library is capable of playback from Teensy 3.2 on-board memory via Wav2Sketch , which is a script which converts Wav Audio format files into Arrays in .h and .cpp formats.
5. Teensy Audio Shield communicates through I2C and supports SDCard via SPI interface.
6. These files can be played back spontaneously without any delays , since on-board memory is very high-speed.

2.3 Design Considerations:

1. **Memory Limits:** The On-System memory on Teensy 3.2 is 256Kb , this is not sufficient to store the large number of high-fidelity audio samples required for the air instrument. Thus , a MicroSD card was added to enhance memory limits.
2. **Reduced Sample Size:** The samples were limited to a duration of approximately 2 sec/sample , this reduced size of individual sample , allowing faster reading from SDcard.
3. **Stereo to Mono:** The samples were converted from stereo to mono 44.1Khz WAV files , Teensy Audio Shield can only playback limited codecs.
4. **ACTIVITY instead of TAP Interrupt:** The TAP event used a duration counter which is limited to approximately 0.015 seconds of max duration for the event. Which is highly impractical to be the time of a guitar strum. The ACTIVITY event on the other hand only fires an interrupt when the defined threshold is crossed in an activated Axis. This is more suitable for the application.
5. **Only 1 ADXL343 :** The ADXL343 accelerometer is a highly accurate and precise device , it was deemed unnecessary to have redundancy in sensors , it would waste more processing power than noticeable benefits. So preferring simplicity of design to achieve goal, only 1 ADXL343 is used as the pick of AirGuitar.
6. **Conductive Tape as Switch:** The capacitive touch buttons were replaced by Conductive Tape , which is taped onto a glove in specific patterns to allow touch capability on two fingers to select one of four total chords. Two GPIO pins are configured as Inputs with Pull-Up enabled, One large pattern of conductive tape is connected to Ground and two small patterns on finger tips are connected to 2 GPIO pins. When a finger tip touches ground , it will change the chord to corresponding pattern.
7. **Multiple Samples:** 4 variations of downstroke and upstroke each of each individual chord were recorded for a total of 32 samples. These samples change randomly with each strum providing for a more natural sounding guitar.

2.4 Special Considerations:

1. **Audio Playback Queuing:** The audio playback queueing system was designed to minimize overlapping and harsh cuts to different sounds per detection. This system uses two playback objects and two fade objects from Teensy Audio Library to control hand-offs and transitions from one playback to another without conflicting. ***REFER AudioQueue.h***
2. **Random Sample Playback:** The system plays back audio samples with designated names being generated in pseudo random order using random() and randomSeed() functions of Arduino library . The random seed is tied to an Analog Read Pin which is floating so the pattern of random numbers generated are different in each iteration , making the system pseudo random. ***Refer FileMap.h***

3. Directions of Use:

1. The Glove hand has conductive patches on the finger tips and a large conductive strip on the wrist section . The movement of finger tips to touch the strip will cause the chords to change.
2. The chords corresponding to the finger patterns are as follows:

1st Finger	3rd Finger	Chord Played
Open	Open	A min
Closed	Open	G maj
Open	Closed	C maj
Closed	Closed	F maj

3. Now , if strumming hand is moved up or down in a motion similar to a guitar strum , then system will play corresponding upstroke or downstroke chord sample.
4. Debugging and live log can be accessed through Serial Monitor at 9600 baud.

4.1. Code Base:

Code is accessible in the root folder of the project.
Code was split up into smaller components and header files to ease readability and accessibility.
Code Base is made up of 5 Components , as described:

1. **AirGuitar.ino :** This contains the setup() and loop() of the program.
2. **ADXL.h :** This contains ADXL Setup and supplementary functions.
3. **AudioQueue.h :** This is the Audio Playback Object queue system.
4. **AudioShield.**h : Setup Teensy Audio Library and Audio Shield
5. **FileMap.**h : Random file name generation

4.2. Scalability/Possible Enhancements:

1. **Improved Range of Sounds :** The system is capable of easy expansion and improvements are possible even with current hardware setup. The recorded sample quality can be increased dramatically for a more natural sounding device. The number of samples can also be easily increased and incorporated into the code base with simple changes to FileMap.h .
2. **More Chord Patterns:** The fret board detection system can be improved by adding a touch screen panel at the expense of cost but making much more complex fret patterns possible , increasing the maximum number of possible patterns to infinity.

   On the other hand , up to 32 patterns are possible by using 5 GPIO pins and mapping them to Conductive Tape switches on the five fingers of the hand.
3. **Better Queuing and Mixing:** Improvements on mixing and queuing of playback objects can be achieved by applying sound and signal processing techniques. This will improve natural sounding nature of device without incurring additional cost.
4. **Inertial Measurement Units:** The usage of IMU devices instead of ADXL343 can offer enhanced real time 3d vector position detection , this can be used to incorporate elements such as strum speed , pressure , amplitude and even map to individual notes , offering many possible application scenarios.
5. **Improved Robust Design:** The device is a barebones device with no robustness or safety features as of now , it can be highly improved to give a reliable , dependable device.

4.3. Cost:

The total cost of the device is under 20$, making it economically feasible.

4.4. Debugging:

1. **Serial Monitor:** Serial Monitor will display Initialization status , current playback status and error codes , if occurred.
2. ADXL343 , Teensy Audio Shield , SD Card and Serial Communications throw errors if not initialized properly. Check connections in this case.