

Arduino Air Drums



Aman Sohane

Chinmay Khandekar

Paritosh Meihar

Overview

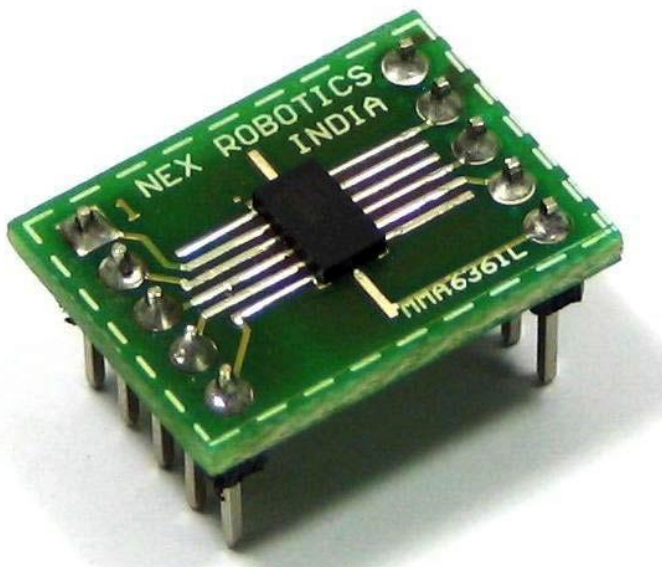
Many people want to become a drummer. Particularly because they are awesome, stylish and rocking. But most of them cannot fulfill their dream. Maybe because drum sets are too expensive or maybe some of them just want to have an experience of drumming. We can help them out with our drumming kit. Here we present a really cool drum kit that has a special twist. "NO DRUMS at all". Yes. It can be played in air.

We are using Arduino Uno, some MIDI devices, USB cables and other electronics devices for constructing our drum kit. We know its not going to challenge the true drummers and their drum set but we are sure all those who want to be a drummer would love to get their hand on our "Arduino Air Drums"

Hardware

The hardware consists of arduino micro-controller, two axis accelerometer, a *chappal* and 2 PCBs. We also use laptop with serial to MIDI converter and MIDI software to generate sounds.

Accelerometer



MMA6361L is a Two axis Low-g accelerometer Module which gives selectable acceleration range from $\pm 1.5g$ or $\pm 6g$.

Pin	Pin Name	Description
1	XOUT	X direction output Voltage
2	YOUT	Y direction output Voltage
4	Vss	Supply Ground
5	Vdd	3.3V supply voltage
6	Sleep (Active Low)	Logic input pin to enable product or sleep mode
8	g-select	Logic input pin to select g level

Pins 3,7,9,10 are not connected. The g-select feature allows to select between two sensitivities 800mV/g (logic 0) and 206mv/g (logic 1). We are using 800mv/g sensitivity so that pin 8 is connected to pin 4 (Supply Ground). Pin 6 is just to select normal mode of operation in which it consumes 400uA current . It is connected Vcc (pin 5).

Software

We are using Serial to MIDI converter to “DECODE” the message sent by arduino and then using any MIDI player to play the beat. The Serial to Midi converter is an application written in processing environment which monitors the data received on serial port and gives process it. Also a virtual route is required to map the output port of Serial to MIDI converter and input of MIDI player. We used LoopBe for the purpose.

Working

Our drums kit has 3 parts:

1. Snare
2. Hi-hat
3. Bass

Snare and Hi-hat

For snare and hi-hat, we use accelerometers so that they can be played in air. As can be seen from the picture, we have made two sticks for snare and hi-hat. The Y-output of snare accelerometer goes to analog pin 0 and Y-output of hi-hat snare accelerometer to analog pin 1 of Arduino. This output can be observed on serial monitor as decimal values from 0 to 1023. When stick stops midway in the air (condition of “hit” is met) , it gives extreme value because of infinite resistance and on the serial monitor we get zero or very small value of voltage (by small we mean less than 100 on serial monitor). Usually when accelerometer is kept steady, it gives value around 380 on serial monitor. And when we are moving it in one particular direction with increasing velocity it gives value around 600. So when it experiences transition, it suddenly drops from this value to 0. When this transition is detected we set ‘activepad’ HIGH and we send a MIDI signal through serial to computer. Serial to MIDI converter software converts this serial value to MIDI value which is used by a virtual MIDI device which generated corresponding beat.

It is possible that arduino might detect more no. of zeros during transition because it takes around 300uS between 2 consecutive readings as observed on serial monitor. So we consider only the first zero and neglect next few values. This is explained in code.

Bass

We observed once during a lab session that when arduino’s input pin is kept floating , it becomes HIGH when it is touched by some conducting material. So we use this trick to detect a bass hit.



The Code.

```
//AIR DRUMS
```

```
//Aman Sohane
```

```
//Chinmay Khandekar
```

```
//Paritosh Meihar
```

```
unsigned char PadNote[3] = {52,100,10};    // MIDI notes from 0 to 127 (Mid C = 60)
```

```
int PadCutOff[3] = {50,50,0};    // condition to cause a drum hit
```

```
int MaxPlayTime[3] = {10,10,10};    // Cycles before a 2nd hit is allowed
```

```
#define midichannel  0;    // MIDI channel
```

```
boolean VelocityFlag = true;// Velocity ON (true) or OFF (false)
```

```
// Internal Use Variables
```

```
boolean activePad[3] = {0,0,0};    // Array of flags of pad currently playing
```

```
int PinPlayTime[3] = {0,0,0};    // Counter since pad started to play
```

```
unsigned char status;
```

```

int pin = 0;
int hitavg = 0;

// Setup
void setup()
{
  Serial.begin(9600);          // connect to the serial port 115200
}

// Main Program
void loop()
{
  for(int pin=0; pin < 2; pin++)
  {
    hitavg = analogRead(pin);    // read the input pin
    // Serial.println((hitavg));
    if(hitavg <= PadCutOff[pin] )    //compare to threshold
    {
      if((activePad[pin] == false))    //giving beat for first time
      {
        MIDI_TX(144,PadNote[pin],127);    //send midi value through serial
        PinPlayTime[pin] = 0;    //counting time to give a minimum time
        differencebetween two beats of same instrument
        activePad[pin] = true;    //set flag high that this instrument is played once
      }
      else
      {
        PinPlayTime[pin] = PinPlayTime[pin] + 1;    //if the flag is high just increment the counter
      }
    }
  }
}

```

```

    }
}
else if((activePad[pin] == true)) //increment the counter otherwise even if threshold
condition is not met
{
    PinPlayTime[pin] = PinPlayTime[pin] + 1;

    if(PinPlayTime[pin] > MaxPlayTime[pin]) //if the counter exceeds given time difference
between two beats then set flag low and send note off the signal.
    {
        activePad[pin] = false;
        MIDI_TX(128,PadNote[pin],127);
    }
}
}
}

// Transmit MIDI Message
void MIDI_TX(unsigned char MESSAGE, unsigned char PITCH, unsigned char VELOCITY)
{
    status = MESSAGE + midichannel;
    Serial.print(status);
    Serial.print(PITCH);
    Serial.print(VELOCITY);
}

```

Problems

Since we are using two axis accelerometers , we can detect both X-output and Y-output transitions and produce corresponding 2 beats using MIDI e.g. Hi-hat for X-output transition and Crash for Y-

output transition. However the accelerometer is so sensitive that when we move it in 1 particular direction and detect a zero, it also gives zero for perpendicular direction output of accelerometer. Hence both the beats are played simultaneously. This can be solved by noting down the threshold for both directions so that arduino sends only 1 serial signal.

Further Improvements

We can try to play 2 beats using only 1 stick as described above.

Also external casing can be improved and made more presentable.

Resources and Links

- For reference to MIDI protocol <http://en.wikipedia.org/wiki/MIDI>
- <http://www.geek.com/articles/geek-cetera/arduino-air-drums-made-from-garden-forks-and-flip-flops-20110729/>
- <http://www.arduino.cc/playground/Interfacing/Processing>
- Serial to MIDI converter -http://spikenzielabs.com/SpikenzieLabs/Serial_MIDI.html
- LoopBe-<http://nerds.de/en/loopbe1.html>