int buttonarray[] = {13, 12, 11, 10, 9, 8}; // [E2, A2, D3, G3, B3, E4]

// each pin represents a guitar string

// next we create and array with frequencies matching each of the strings above

// such that when 13 is selected the freq matching the note e is selected).

float freqarray[] = {82.41, 110.00, 146.83, 196.00, 246.94, 329.63};//sll in Hz

int lowerLed = 7;

int higherLed = 6;

int justRight = 5;

#define LENGTH 512

byte rawData[LENGTH];

int count = 0;

// Sample Frequency in kHz

const float sample\_freq = 8919;

int len = sizeof(rawData);

int i,k;

long sum, sum\_old;

int thresh = 0;

float freq\_per = 0;

byte pd\_state = 0;

void setup(){

for (int i=0; i<=5; i++)

{

pinMode(buttonarray[i], INPUT\_PULLUP);

}

pinMode(lowerLed, OUTPUT);

pinMode(higherLed, OUTPUT);

pinMode(justRight, OUTPUT);

Serial.begin(115200);

}

void loop(){

if (count < LENGTH)

{

count++;

rawData[count] = analogRead(A0)>>2;

}

else {

sum = 0;

pd\_state = 0;

int period = 0;

for(i=0; i < len; i++)

{

// Autocorrelation

sum\_old = sum;

sum = 0;

for(k=0; k < len-i; k++) sum += (rawData[k]-128)\*(rawData[k+i]-128)/256;

// Serial.println(sum);

// Peak Detect State Machine

if (pd\_state == 2 && (sum-sum\_old) <=0)

{

period = i;

pd\_state = 3;

}

if (pd\_state == 1 && (sum > thresh) && (sum-sum\_old) > 0) pd\_state = 2;

if (!i) {

thresh = sum \* 0.5;

pd\_state = 1;

}

}

// Frequency identified in Hz

if (thresh >100) {

freq\_per = sample\_freq/period;

Serial.println(freq\_per);

for (int s=0; s<=5; s++)

{

if (digitalRead(buttonarray[i])== HIGH)

{

if (freq\_per - freqarray[i] < 0)

{

digitalWrite(lowerLed, HIGH);

}

else if(freq\_per - freqarray[i] > 10)

{

digitalWrite(higherLed, HIGH);

}

else

{

digitalWrite(justRight, HIGH);

}

}

}

}

count = 0;

}

}