LISTING PROGRAM

```
boolean clipping = 0;
byte newData = 0;
byte prevData = 0;
unsigned int time = 0;//keeps time and sends vales to store in timer[] occasionally
int timer[10];//storage for timing of events
int slope[10];//storage for slope of events
unsigned int totalTimer;
unsigned int period;
byte index = 0;//current storage index
float frequency;
int maxSlope = 0;//used to te max slope as trigger point
int newSlope;//storage for incoming slope data

byte noMatch = 0;//counts how many non-matches you've received to reset variables if
it's been too long
byte slopeTol = 3;//slope tolerance- adjust this if you need
int timerTol = 7;//timer tolerance- adjust this if you need

unsigned int ampTimer = 0;
byte maxAmp = 0;
byte checkMaxAmp;
byte ampThreshold = 32;
int correctFrequency;

void setup(){

 Serial.begin(9600);
  //LED pins
  pinMode(8,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(39,OUTPUT);
  pinMode(37,OUTPUT);
  pinMode(35,OUTPUT);
  pinMode(33,OUTPUT);
```

```
  pinMode(31,OUTPUT);
  pinMode(29,OUTPUT);



  cli();//disable interrupts

  //mengatur pin a0 menjadi 38.5kHz

  //clear ADCSRA and ADCSRB registers
  ADCSRA = 0;
  ADCSRB = 0;

  ADMUX |= (1 << REFS0); //SET TEGANGAN REFERENSI
  ADMUX |= (1 << ADLAR); //SET ADC MENJADI 8-BIT

  ADCSRA |= (1 << ADPS2) | (1 << ADPS0); //SET ADC CLOCK MENJADI 500kHz
  ADCSRA |= (1 << ADATE); //enabble auto trigger
  ADCSRA |= (1 << ADIE); //enable interrupts jika pengukuran selesai
  ADCSRA |= (1 << ADEN); //enable ADC
  ADCSRA |= (1 << ADSC); //memulai pengukuran adc



  sei();//enable interrupts
}

ISR(ADC_vect) {//when new ADC value ready

  PORTB &= B11101111;//set pin 12 low
  prevData = newData;//
  newData = ADCH;//MENGAMBIL DATA DARI A0
  if (prevData < 127 && newData >=127){//if increasing and crossing midpoint
     newSlope = newData - prevData;//calculate slope

    if (abs(newSlope-maxSlope)<slopeTol){
       //record new data and reset time
       slope[index] = newSlope;
       timer[index] = time;
       time = 0;
```

```
    if (index == 0)
        {//new max slope just reset
          PORTB |= B00010000;//set pin 12 high
          noMatch = 0;
          index++;//increment index
           }

     else if (abs(timer[0]-timer[index])<timerTol && abs(slope[0]-
newSlope)<slopeTol){//if timer duration and slopes
          //sum timer values
          totalTimer = 0;
          for (byte i=0;i<index;i++){
          totalTimer+=timer[i];}
          period = totalTimer;//set period

          //reset new zero index values to compare with
          timer[0] = timer[index];
          slope[0] = slope[index];
          index = 1;//set index to 1
          PORTB |= B00010000;//set pin 12 high
          noMatch = 0;
        }

     else{//crossing midpoint but not match
          index++;//increment index
          if (index > 9){
           reset();
        }

    }
  }

  else if (newSlope>maxSlope){//if new slope is much larger than max slope
    maxSlope = newSlope;
    time = 0;//reset clock
    noMatch = 0;
    index = 0;//reset index
    }

  else{//slope not steep enough
    noMatch++;//increment no match counter
```

```
    if (noMatch>9){
       reset();
     }
    }
 }

 time++;//increment timer at rate of 38.5kHz

 ampTimer++;//increment amplitude timer
 if (abs(127-ADCH)>maxAmp){
  maxAmp = abs(127-ADCH);
 }
 if (ampTimer==1000){
  ampTimer = 0;
  checkMaxAmp = maxAmp;
  maxAmp = 0;
 }

}


void E(){
 digitalWrite(2,0);
 digitalWrite(3,1);
 digitalWrite(4,1);
 digitalWrite(5,0);
 digitalWrite(6,0);
 digitalWrite(7,0);
 digitalWrite(8,0);
 }


void e1(){
 digitalWrite(2,0);
 digitalWrite(3,0);
 digitalWrite(4,1);
 digitalWrite(5,0);
 digitalWrite(6,0);
 digitalWrite(7,0);
 digitalWrite(8,0);
 }
```

```
void f(){
  digitalWrite(2,0);
  digitalWrite(3,1);
  digitalWrite(4,1);
  digitalWrite(5,1);
  digitalWrite(6,0);
  digitalWrite(7,0);
  digitalWrite(8,0);
  }

void A(){
  digitalWrite(2,0);
  digitalWrite(3,0);
  digitalWrite(4,0);
  digitalWrite(5,1);
  digitalWrite(6,0);
  digitalWrite(7,0);
  digitalWrite(8,0);
  }

void D(){
  digitalWrite(2,1);
  digitalWrite(3,0);
  digitalWrite(4,0);
  digitalWrite(5,0);
  digitalWrite(6,0);
  digitalWrite(7,1);
 digitalWrite(8,0);
  }

void G(){
  digitalWrite(2,0);
  digitalWrite(3,1);
  digitalWrite(4,0);
  digitalWrite(5,0);
  digitalWrite(6,0);
  digitalWrite(7,0);
  digitalWrite(8,0);
  }
```

```
void B(){
  digitalWrite(2,1);
  digitalWrite(3,1);
  digitalWrite(4,0);
  digitalWrite(5,0);
  digitalWrite(6,0);
  digitalWrite(7,0);
  digitalWrite(8,0);
  }
void C(){
  digitalWrite(2,0);
  digitalWrite(3,1);
  digitalWrite(4,1);
  digitalWrite(5,0);
  digitalWrite(6,0);
  digitalWrite(7,0);
  digitalWrite(8,1);
  }



void reset(){
  index = 0;//reset index
  noMatch = 0;//reset match couner
  maxSlope = 0;//reset slope
}

void stringCheck(){


  if( frequency>105&&frequency<116){
    A();
    Serial.println("A");
    correctFrequency = 110;
    }

  if(frequency>209&&frequency<232){
    A();
    Serial.println("A");
    correctFrequency = 220;
```

```
  }

 if(frequency>416&&frequency<465){
  A();
  Serial.println("A");
  correctFrequency = 440;
  }

 if(frequency>832&&frequency<931){
  A();
  Serial.println("A");
  correctFrequency = 880;
  }

 if(frequency>240&&frequency<255){
  B();
  Serial.println("B");
  correctFrequency = 247;
  }

if( frequency>119&&frequency<126){
  B();
  Serial.println("B");
  correctFrequency = 123;
  }

 if(frequency>467&&frequency<522){
  B();
  Serial.println("B");
  correctFrequency = 494;
  }

 if(frequency>933&&frequency<1046){
  B();
  Serial.println("B");
  correctFrequency = 988;
  }

 if(frequency>127&&frequency<139){
  C();
  Serial.println("C");
```

```
 correctFrequency = 131;
 }

if(frequency>256&&frequency<276){
 C();
 Serial.println("C");
 correctFrequency = 262;
 }

if(frequency>495&&frequency<553){
 C();
 Serial.println("C");
 correctFrequency = 523;
 }

if(frequency>989&&frequency<1975){
 C();
 Serial.println("C");
 correctFrequency = 1047;
 }

if(frequency>140&&frequency<155){
 D();
 Serial.println("D");
 correctFrequency = 146.8;
 }

if( frequency>278&&frequency<310){
 D();
 Serial.println("D");
 correctFrequency = 294;
 }

if( frequency>555&&frequency<621){
 D();
 Serial.println("D");
 correctFrequency = 587;
 }

if(frequency>77&&frequency<86){
 E();
```

```
 Serial.println("E");
 correctFrequency = 82.4;
 }

if(frequency>157&&frequency<170){
 E();
 Serial.println("E");
 correctFrequency = 165;
 }

if(frequency>313&&frequency<340){
 e1();
 Serial.println("E");
 correctFrequency = 329;
 }

if(frequency>623&&frequency<697){
 e1();
 Serial.println("E");
 correctFrequency = 659;
 }

if(frequency>170&&frequency<185){
 f();
 Serial.println("F");
 correctFrequency = 175;
 }

if(frequency>88&&frequency<91){
 f();
 Serial.println("F");
 correctFrequency = 87;
 }

if(frequency>342&&frequency<369){
 f();
 Serial.println("F");
 correctFrequency = 349;
 }

if(frequency>660&&frequency<739){
```

```
  f();
  Serial.println("F");
  correctFrequency = 698;
  }

 if(frequency>93&&frequency<103){
  G();
  Serial.println("G");
  correctFrequency = 98;
  }

 if(frequency>371&&frequency<414){
  G();
  Serial.println("G");
  correctFrequency = 392;
  }

 if(frequency>186&&frequency<207){
  G();
  Serial.println("G");
  correctFrequency = 196;
  }

 if(frequency>741&&frequency<830){
  G();
  Serial.println("G");
  correctFrequency = 784;
  }


}

void frequencyCheck(){
 int acuration = frequency - correctFrequency ;
 if(acuration >= 1 && acuration <= 4){
  digitalWrite(33,1);
 }
 else if(acuration >= 4 && acuration <= 7){
  digitalWrite(31,1);
 }
 else if(acuration <= - 1 && acuration >= -4){
```

```
   digitalWrite(39,1);
  }
  else if(acuration <= -4 && acuration >= -7){
   digitalWrite(37,1);
  }
  else if(acuration == 0){
   digitalWrite(35,1);
   digitalWrite(29,1);
  }

}

void allLEDsOff(){
  digitalWrite(2,1);
  digitalWrite(3,1);
  digitalWrite(4,1);
  digitalWrite(5,1);
  digitalWrite(6,1);
  digitalWrite(7,1);
  digitalWrite(8,1);
  digitalWrite(39,0);
  digitalWrite(37,0);
  digitalWrite(35,0);
  digitalWrite(33,0);
  digitalWrite(31,0);
  digitalWrite(29,0);
}

void loop(){

  allLEDsOff();

  if (checkMaxAmp>ampThreshold){
frequency = 38462/float(period);//calculate frequency timer rate/period
   }
   Serial.println(frequency);
  stringCheck();
  frequencyCheck();
  delay(100);

}
```