

```
/******
```

THE SOFTWARE: For PlanxTone aka The Poluphonic Plank

Philip Catterall 27th April 2018

Development History

V01: Renamed from Midi piano

Vo2: Piano changed to Shachuchi

Vo3: Proximity range changed

V04: Octave Shifter

V05: Instrument changer

V06: TBD Key shifter / C, G, D, F plus octave down on original white notes (add up subtract multiples of 8 from note

V06a Changed Octave shifting, also bug, extra Note in array meant starting at D not C (counting backwards!!

V07 : Mode changer

```
*****/
```

```
// compiler error handling
```

```
#include "Compiler_Errors.h"
```

```
// serial rate
```

```
#define baudRate 57600
```

```
// include the relevant libraries
```

```
#include <MPR121.h>
```

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial mySerial(12, 10); // Soft TX on 10, we don't use RX in this code
```

```
//Control Definitions
```

```
//Pins P5 P4 Red Button (LH Binary) (Middle Rotary Switch MRS} Key Changer
```

```
//Pos 1 0 0
```

```
//Pos 2 1 0
```

```
//Pos 3 0 1
```

```
//Pos 4 1 1
```

```
//Pins P7 P6 Blue Button (RH Binary) (RHRSw) Instrument Changer
```

```
//Pos 1 0 0
```

```
//Pos 2 1 0
```

```
//Pos 3 0 1
```

```
//Pos 4 1 1
```

```
//Pin AO (LHRSw ~ Left Hand Rotary Switch}
```

```
//Pos 1 0
```

```
//Pos 2 203
```

```
//Pos 3 409
```

```
//Pos 4 614
```

```
//Pos 5 818 // CHECK rotating too far and causing short still, restrict to 4 rotations, or sub with binary
```

```
byte ValP4 = HIGH;
```

```
byte ValP5 = HIGH;
```

```
byte ValP6 = HIGH;
```

```
byte ValP7 = HIGH;
```

```
byte preValP4 = HIGH;
```

```
byte preValP5 = HIGH;
```

```
byte preValP6 = HIGH;
byte preValP7 = HIGH;
```

```
int sensorValueA0= 0;
```

```
byte Oct_1st = 0;
byte Oct_2nd = 0;
byte Oct_3rd = 0;
byte Oct_4th = 0;
byte Prev_Oct_1st = 0;
byte Prev_Oct_2nd = 0;
byte Prev_Oct_3rd = 0;
byte Prev_Oct_4th = 0;
```

```
//Instruments
```

```
int acoustic_grand_piano = 1;
int organ3 = 18;
int electric_guitar_clean = 28;
int shakuhachi = 75; // ethnic flute
int instrument = 1;// MIDI Default Piano
int prev_instrument = 1;
```

```
// Touch Board Setup variables
```

```
#define firstPin 0
```

```
#define lastPin 11
```

```
// VS1053 setup
```

```
byte note = 0; // The MIDI note value to be played
byte resetMIDI = 8; // Tied to VS1053 Reset line
byte ledPin = 13; // MIDI traffic indicator
//int instrument = 0; original script
```

```
// key definitions
```

```
                                440Hz
//                                4C, 4D, 4E, 4F, 4G, 4A, 4B, 5C, 5D, 5E, 5F, 5G, 5A
//const byte WhiteNotes[] = {60, 62, 64, 65, 67, 69, 71, 72, 74, 76, 77, 79}; //Original default WhiteNotes Key of C
byte KeyNotes[] = {79, 77, 76, 74, 72, 71, 69, 67, 65, 64, 62, 60}; //
//                                5G, 5F, 5E, 5D, 5C, 4B, 4A, 4G, 4F, 4E, 4D, 4C} Key of C
//                                5F# = 78 4F# = 66
//                                5G, 5F, 5E, 5D, 5C, 4B, 4A, 4G, 4F, 4E, 4D, 4C
//                                i = 0 1 2 3 4 5 6 7 8 9 10 11
//                                5C#=73 i=4 4C#=61 i= 11 Key of D
//                                4Bf (4A#)=70 i=5
//const byte KeyNotes[] = {36, 38, 40, 41, 43, 45, 47, 48, 50, 52, 53, 55, 57}; //Default is KeyNotes_24
//byte KeyNotes[] = {57, 55, 53, 52, 50, 48, 47, 45, 43, 41, 40, 38, 36}; //Default is KeyNotes_24 NB This is
reversed
//byte notes_key_C[] = {57, 55, 53, 52, 50, 48, 47, 45, 43, 41, 40, 38, 36};
```

```
//const byte notes_key_G[] =
//const byte notes_key_D[] =
//const byte notes_key_F[] =
const byte Octave02 = -24;
```

```

const byte  Octave03 = -12;
const byte  Octave04 = 0;
const byte  Octave05 = 12;
const byte  Octave06 = 24;
byte  OctShiftX = 0;
const byte  allNotes[] = {60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71};
// if you're using this with drum sounds instead of notes, you can choose each
// individual drum sound from the bank shown at the bottom of this code
// where it says "PERCUSSION INSTRUMENTS (GM1 + GM2)"
const byte  drumNotes[] = {60, 36, 41, 44, 56, 54, 58, 36, 72, 79, 76, 58};

```

```

void setup(){
  Serial.begin(baudRate);
  pinMode(4, INPUT_PULLUP);//
  pinMode(5, INPUT_PULLUP);//
  pinMode(6, INPUT_PULLUP);//
  pinMode(7, INPUT_PULLUP);//

  // uncomment the line below if you want to see Serial data from the start
  // while (!Serial);
  // Setup soft serial for MIDI control
  mySerial.begin(31250);
  Wire.begin();
  // 0x5C is the MPR121 I2C address on the Bare Touch Board
  if(!MPR121.begin(0x5C)){
    Serial.println("error setting up MPR121");
    switch(MPR121.getError()){
      case NO_ERROR:
        Serial.println("no error");
        break;
      case ADDRESS_UNKNOWN:
        Serial.println("incorrect address");
        break;
      case READBACK_FAIL:
        Serial.println("readback failure");
        break;
      case OVERCURRENT_FLAG:
        Serial.println("overcurrent on REXT pin");
        break;
      case OUT_OF_RANGE:
        Serial.println("electrode out of range");
        break;
      case NOT_INITED:
        Serial.println("not initialised");
        break;
      default:
        Serial.println("unknown error");
        break;
    }
    while(1);
  }

  // pin 4 is the MPR121 interrupt on the Bare Touch Board
  MPR121.setInterruptPin(4);

```

```

// initial data update
MPR121.updateTouchData();

// Reset the VS1053
pinMode(resetMIDI, OUTPUT);
digitalWrite(resetMIDI, LOW);
delay(100);
digitalWrite(resetMIDI, HIGH);
delay(100);

// initialise MIDI
setupMidi();

// V03 Proximity added
// this is the touch threshold - setting it low makes it more like a proximity trigger
// default value is 40 for touch
MPR121.setTouchThreshold(3); //8 in original //3 for 2cm range

// this is the release threshold - must ALWAYS be smaller than the touch threshold
// default value is 20 for touch
MPR121.setReleaseThreshold(2); //4 in original //2 for 2cm range
Serial.begin(9600); // setup serialchannel for time stamping
}
// = = = = = MAINPROGRAMME = = = = =
void loop(){
  if(MPR121.touchStatusChanged()){
    MPR121.updateAll();
    ValP4 = digitalRead(5); // number switched to correct wiring swap!
    ValP5 = digitalRead(4);
    ValP6 = digitalRead(7);
    ValP7 = digitalRead(6);
    // sensorValueA0 = analogRead(A0);

// = = = = = OctaveSelect = = = = =

// Pos 1 = 0 : Pos 2 = 203 : Pos 3 = 409 : Pos 4 = 614 : Pos 5 = 818
    if ( sensorValueA0 < 50 ) // Pos 1 = 0
    {OctShiftX = Octave02;}
    else if ( sensorValueA0 < 250 ) // Pos 2 203
    {OctShiftX = Octave03;}
    else if ( sensorValueA0 < 450 ) // Pos 3 409
    {OctShiftX = Octave04;}
    else if ( sensorValueA0 < 660 ) // Pos 4 614
    {OctShiftX = Octave05;}
    else if ( sensorValueA0 < 860 ) // Pos 5 818
    {OctShiftX = Octave06;}

// Serial.print("Pin A0 = "); Serial.print(sensorValueA0);
// Serial.print(" / OctShiftX = ");
// Serial.print(OctShiftX); Serial.print(" instrument = ");
// Serial.print( instrument ); Serial.println();

// = = = = = EndOctaveSelect = = = = =

```

```
//===== Key Change=====
```

```
//Pins P5 P4 Red Button (LH Binary) Key Changer
```

```
//Pos 1 0 0
```

```
//Pos 2 1 0
```

```
//Pos 3 0 1
```

```
//Pos 4 1 1
```

```
//const byte KeyNotes[] =
```

```
//const byte notes_key_C[] =
```

```
//const byte notes_key_G[] =
```

```
//const byte notes_key_D[] =
```

```
//const byte notes_key_F[] =
```

```
if ( ValP5==0 && ValP4==0) // Key of C, put notes back to C
```

```
{KeyNotes[1] =77;
```

```
KeyNotes[8] =65;
```

```
KeyNotes[4] =72;
```

```
KeyNotes[11] =60;
```

```
}
```

```
else if ( ValP5==0 && ValP4==1 ) // Key of G
```

```
{KeyNotes[1] =78;
```

```
KeyNotes[8] =66;
```

```
KeyNotes[4] =72;// estore back to original
```

```
KeyNotes[11] =60;// Ditto
```

```
}
```

```
else if ( ValP5==1 && ValP4==0 ) // Key of D
```

```
{ KeyNotes[1] =78;
```

```
KeyNotes[8] =66;
```

```
KeyNotes[4] =73;
```

```
KeyNotes[11] =61;
```

```
}
```

```
else if ( ValP5==1 && ValP4==1 ) // Key of F
```

```
{KeyNotes[1] =77; //set notes back to normal
```

```
KeyNotes[8] =65;
```

```
KeyNotes[4] =72;
```

```
KeyNotes[11] =60;
```

```
KeyNotes[5] =70; //B Flat
```

```
}
```

```
//===== EndKey Change=====
```

```
//===== InstrumenSelect=====
```

```
//Pins P7 P6 Blue Button (RH Binary) (RHRSw) Instrument Changer
```

```
//Pos 1 0 0
```

```
//Pos 2 1 0
```

```
//Pos 3 0 1
```

```
//Pos 4 1 1
```

```
//int acoustic_grand_piano = 1;
```

```
//int organ3 = 18;
```

```
//int electric_guitar_clean = 28;
```

```
//int shakuhachi = 75; // ethnic flute
```

```
//int instrument = 0 // MIDI Default Piano
```

```
if ( ValP7==0 && ValP6==0)
{instrument= acoustic_grand_piano;}
else if ( ValP7==0 && ValP6==1 ) //
{instrument= organ3;}
else if ( ValP7==1 && ValP6==0 ) //
{instrument= shakuhachi;}
else if ( ValP7==1 && ValP6==1 ) //
{instrument= electric_guitar_clean;}
```

```
if (instrument != prev_instrument)
{Serial.print(" Instrument = ");
 //Serial.print(instrument);
 // Serial.print(" Prev Instrument = ");
 // Serial.print(prev_instrument);
 //Serial.println();
 //delay(500);
 // pinMode(resetMIDI, OUTPUT);
 // digitalWrite(resetMIDI, LOW);
 // delay(100);
 // digitalWrite(resetMIDI, HIGH);
 delay(100);
 talkMIDI(0xC0, instrument, 0);
 prev_instrument = instrument;}
```

```
// = = = = = End of InstrumentSelect = = = = =
```

```
// code below sets up the board essentially like a piano, with an octave
// mapped to the 12 electrodes, each a semitone up from the previous
for(int i=firstPin; i<=lastPin; i++){
```

```
    // you can choose how to map the notes here - try replacing "KeyNotes" with
    // "allNotes" or "drumNotes"
    note = (KeyNotes[lastPin-i]+OctShiftX);
    if(MPR121.isNewTouch(i)){
        //Note on channel 1 (0x90), some note value (note), 75% velocity (0x60):
        noteOn(0, note, 0x60);
        Serial.print("Note ");
        Serial.print(note);
        Serial.println(" on");
    } else if(MPR121.isNewRelease(i)) {
        // Turn off the note with a given off/release velocity
        noteOff(0, note, 0x60);
        Serial.print("Note ");
        Serial.print(note);
        Serial.println(" off");
    }
}
```

```

}

// functions below are little helpers based on using the SoftwareSerial
// as a MIDI stream input to the VS1053 - all based on stuff from Nathan Seidle

// Send a MIDI note-on message. Like pressing a piano key.
// channel ranges from 0-15
void noteOn(byte channel, byte note, byte attack_velocity) {
  talkMIDI( (0x90 | channel), note, attack_velocity);
}

// Send a MIDI note-off message. Like releasing a piano key.
void noteOff(byte channel, byte note, byte release_velocity) {
  talkMIDI( (0x80 | channel), note, release_velocity);
}

// Sends a generic MIDI message. Doesn't check to see that cmd is greater than 127,
// or that data values are less than 127.
void talkMIDI(byte cmd, byte data1, byte data2) {
  digitalWrite(ledPin, HIGH);
  mySerial.write(cmd);
  mySerial.write(data1);

  // Some commands only have one data byte. All cmds less than 0xBn have 2 data bytes
  // (sort of: http://253.ccarh.org/handout/midiprotocol/)
  if( (cmd & 0xF0) <= 0xB0)
    mySerial.write(data2);

  digitalWrite(ledPin, LOW);
}

// SETTING UP THE INSTRUMENT:
// The below function "setupMidi()" is where the instrument bank is defined. Use the VS1053 instrument library
// below to aid you in selecting your desire instrument from within the respective instrument bank

void setupMidi(){

  // Volume - don't comment out this code!
  talkMIDI(0xB0, 0x07, 127); //0xB0 is channel message, set channel volume to max (127)

  // -----
  // Melodic Instruments GM1
  // -----
  // To Play "Electric Piano" (5): Flute (72) ;To Play "Shakuhachi" (75):
  talkMIDI(0xB0, 0, 0x00); // Default bank GM1
  // We change the instrument by changing the middle number in the brackets
  // talkMIDI(0xC0, instrument, 0); "number" can be any number from the melodic table below
  talkMIDI(0xC0, 71, 0); // Set instrument number. 0xC0 is a 1 data byte command(55,0)
  // -----
  // Percussion Instruments (Drums, GM1 + GM2)
  // -----
  // uncomment the two lines of code below to use - you will also need to comment out the two "talkMIDI" lines
  // of code in the Melodic Instruments section above

```

```
// talkMIDI(0xB0, 0, 0x78); // Bank select: drums
// talkMIDI(0xC0, 0, 0); // Set a dummy instrument number
// -----
}
```

/* MIDI INSTRUMENT LIBRARY:

MELODIC INSTRUMENTS (GM1)

When using the Melodic bank (0x79 - same as default), open chooses an instrument and the octave to map
To use these instruments below change "number" in talkMIDI(0xC0, number, 0) in setupMidi()

1 Acoustic Grand Piano	33 Acoustic Bass	65 Soprano Sax	97 Rain (FX 1)
2 Bright Acoustic Piano	34 Electric Bass (finger)	66 Alto Sax	98 Sound Track (FX 2)
3 Electric Grand Piano	35 Electric Bass (pick)	67 Tenor Sax	99 Crystal (FX 3)
4 Honky-tonk Piano	36 Fretless Bass	68 Baritone Sax	100 Atmosphere (FX 4)
5 Electric Piano 1	37 Slap Bass 1	69 Oboe	101 Brighthouse (FX 5)
6 Electric Piano 2	38 Slap Bass 2	70 English Horn	102 Goblins (FX 6)
7 Harpsichord	39 Synth Bass 1	71 Bassoon	103 Echoes (FX 7)
8 Clavi	40 Synth Bass 2	72 Clarinet	104 Sci-fi (FX 8)
9 Celesta	41 Violin	73 Piccolo	105 Sitar
10 Glockenspiel	42 Viola	74 Flute	106 Banjo
11 Music Box	43 Cello	75 Recorder	107 Shamisen
12 Vibraphone	44 Contrabass	76 Pan Flute	108 Koto
13 Marimba	45 Tremolo Strings	77 Blown Bottle	109 Kalimba
14 Xylophone	46 Pizzicato Strings	78 Shakuhachi	110 Bag Pipe
15 Tubular Bells	47 Orchestral Harp	79 Whistle	111 Fiddle
16 Dulcimer	48 Trimpani	80 Ocarina	112 Shanai
17 Drawbar Organ	49 String Ensembles 1	81 Square Lead (Lead 1)	113 Tinkle Bell
18 Percussive Organ	50 String Ensembles 2	82 Saw Lead (Lead)	114 Agogo
19 Rock Organ	51 Synth Strings 1	83 Calliope (Lead 3)	115 Pitched Percussion
20 Church Organ	52 Synth Strings 2	84 Chiff Lead (Lead 4)	116 Woodblock
21 Reed Organ	53 Choir Aahs	85 Charang Lead (Lead 5)	117 Taiko
22 Accordion	54 Voice oohs	86 Voice Lead (Lead)	118 Melodic Tom
23 Harmonica	55 Synth Voice	87 Fifths Lead (Lead 7)	119 Synth Drum
24 Tango Accordion	56 Orchestra Hit	88 Bass + Lead (Lead 8)	120 Reverse Cymbal
25 Acoustic Guitar (nylon)	57 Trumpet	89 New Age (Pad 1)	121 Guitar Fret Noise
26 Acoustic Guitar (steel)	58 Trombone	90 Warm Pad (Pad 2)	122 Breath Noise
27 Electric Guitar (jazz)	59 Tuba	91 Polysynth (Pad 3)	123 Seashore
28 Electric Guitar (clean)	60 Muted Trumpet	92 Choir (Pad 4)	124 Bird Tweet
29 Electric Guitar (muted)	61 French Horn	93 Bowed (Pad 5)	125 Telephone Ring
30 Overdriven Guitar	62 Brass Section	94 Metallic (Pad 6)	126 Helicopter
31 Distortion Guitar	63 Synth Brass 1	95 Halo (Pad 7)	127 Applause
32 Guitar Harmonics	64 Synth Brass 2	96 Sweep (Pad 8)	128 Gunshot

PERCUSSION INSTRUMENTS (GM1 + GM2)

When in the drum bank (0x78), there are not different instruments, only different notes.
To play the different sounds, select an instrument # like 5, then play notes 27 to 87.

27 High Q	43 High Floor Tom	59 Ride Cymbal 2	75 Claves
28 Slap	44 Pedal Hi-hat [EXC 1]	60 High Bongo	76 Hi Wood Block
29 Scratch Push [EXC 7]	45 Low Tom	61 Low Bongo	77 Low Wood Block

30 Srcatch Pull [EXC 7]	46 Open Hi-hat [EXC 1]	62 Mute Hi Conga	78 Mute Cuica [EXC 4]
31 Sticks	47 Low-Mid Tom	63 Open Hi Conga	79 Open Cuica [EXC 4]
32 Square Click	48 High Mid Tom	64 Low Conga	80 Mute Triangle [EXC 5]
33 Metronome Click	49 Crash Cymbal 1	65 High Timbale	81 Open Triangle [EXC 5]
34 Metronome Bell	50 High Tom	66 Low Timbale	82 Shaker
35 Acoustic Bass Drum	51 Ride Cymbal 1	67 High Agogo	83 Jingle bell
36 Bass Drum 1	52 Chinese Cymbal	68 Low Agogo	84 Bell tree
37 Side Stick	53 Ride Bell	69 Casbasa	85 Castanets
38 Acoustic Snare	54 Tambourine	70 Maracas	86 Mute Surdo [EXC 6]
39 Hand Clap	55 Splash Cymbal	71 Short Whistle [EXC 2]	87 Open Surdo [EXC 6]
40 Electric Snare	56 Cow bell	72 Long Whistle [EXC 2]	
41 Low Floor Tom	57 Crash Cymbal 2	73 Short Guiro [EXC 3]	
42 Closed Hi-hat [EXC 1]	58 Vibra-slap	74 Long Guiro [EXC 3]	

*/

/*****

Bare Conductive On Board Midi Piano (or other instrument... or drum kit!)

Midi_Piano.ino - basic sketch that defines the Touch Board as a particular instrument when placed in onboard MIDI mode.

To do this, connect the two solder bridges with "MIDI" and "MIDI ON" printed next the the solder pads on the Touch Board. You can do this with solder or using small blobs of Electric Paint.

By default, this is a 12-key keyboard with just the white notes (a diatonic scale). This can be changed to a full chromatic scale (black and white notes) or a drum kit by modifying the indicated sections in loop() (and also setupMIDI() if you want drums). You can also change this to be any instrument by modifying the Melodic Instruments section of setupMIDI().

Bare Conductive code written by Stefan Dzisiewski-Smith and Peter Krige. Much thievery from Nathan Seidle in this particular sketch. Thanks Nate - we owe you a cold beer!

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License (CC BY-SA 3.0) <http://creativecommons.org/licenses/by-sa/3.0/>

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Developmnt Hostoy
V01: Renamed from Midi piano
Vo2: Piano changed to Shachuchi
Vo3: Proximity range changed
vo4: TBD Key shifter

V05: Inst changer
V06 : Mode changer

//===== NEW NOTE DEFINITIONS Defs 1st July 13

```
const byte note0 = 0; //Octave -5 C midi note no 0 / freq 8.18 m1Cn
const byte note36 =36; //Octave 2 C midi note no 36
const byte note37 =37; //Octave 2 C# midi note no 37
const byte note38 =38; //Octave 2 D midi note no 38
const byte note39 =39; //Octave 2 D# midi note no 39/ freq 77.78 2Ds
const byte note40 =40; //Octave 2 E midi note no 40/ freq 82.41 2En
const byte note41 =41; //Octave 2 F midi note no 41/ freq 87.31 2Fn
const byte note42 =42; //Octave 2 F# midi note no 42/ freq 92.50 2Fs
const byte note43 =43; //Octave 2 G midi note no 43/ freq 98.00 2Gn
const byte note44 =44; //Octave 2 G# midi note no 44/ freq 103.83 2Gs
const byte note45 =45; //Octave 2 A midi note no 45/ freq 110.00 2An
const byte note46 =46; //Octave 2 A# midi note no 46/ freq 116.54 2As
const byte note47 =47; //Octave 2 B midi note no 47/ freq 123.47 2Bn
```

```
const byte note48 =48; //Octave 3 C midi note no 48/ freq 130.81 3Cn
const byte note49 =49; //Octave 3 C# midi note no 49/ freq 138.59 3Cs
const byte note50 =50; //Octave 3 D midi note no 50/ freq 146.83 3Dn
const byte note51 =51; //Octave 3 D# midi note no 51/ freq 155.56 3Ds
const byte note52 =52; //Octave 3 E midi note no 52/ freq 164.81 3En ( F3 )
const byte note53 =53; //Octave 3 F midi note no 53/ freq 174.61 3Fn
const byte note54 =54; //Octave 3 F# midi note no 54/ freq 185.00 3Fs
const byte note55 =55; //Octave 3 G midi note no 55/ freq 196.00 3Gn
const byte note56 =56; //Octave 3 G# midi note no 56/ freq 207.66 3Gs
const byte note57 =57; //Octave 3 A midi note no 57/ freq 220.00 3An
const byte note58 =58; //Octave 3 A# midi note no 58/ freq 233.08 3As
const byte note59 =59; //Octave 3 B midi note no 59/ freq 246.94 3Bn
```

```
const byte note60 =60; //Octave 4 C midi note no 60/ freq 261.63 4Cn ~ MIDDLE "C"
const byte note61 =61; //Octave 4 C# midi note no 61/ freq 277.18 4Cs
const byte note62 =62; //Octave 4 D midi note no 62/ freq 293.66 4Dn
const byte note63 =63; //Octave 4 D# midi note no 63/ freq 311.13 4Ds
const byte note64 =64; //Octave 4 E midi note no 64/ freq 329.63 4En
const byte note65 =65; //Octave 4 F midi note no 65/ freq 349.23 4Fn
const byte note66 =66; //Octave 4 F# midi note no 66/ freq 369.99 4Fs
const byte note67 =67; //Octave 4 G midi note no 67/ freq 392.00 4Gn
const byte note68 =68; //Octave 4 G# midi note no 68/ freq 415.30 4Gs
const byte note69 =69; //Octave 4 A midi note no 69/ freq 440.00 4An ~ MIDDLE "A"
const byte note70 =70; //Octave 4 A# midi note no 70/ freq 446.16 4As
const byte note71 =71; //Octave 4 B midi note no 71/ freq 493.88 4Bn
```

```
const byte note72 =72; //Octave 5 C midi note no 60/ freq 523 5Cn
const byte note73 =73; //Octave 5 C# midi note no 61/ freq 554 5Cs
const byte note74 =74; //Octave 5 D midi note no 62/ freq 587 5Dn
const byte note75 =75; //Octave 5 D# midi note no 63/ freq 622 5Ds
const byte note76 =76; //Octave 5 E midi note no 64/ freq 659 5En
const byte note77 =77; //Octave 5 F midi note no 65/ freq 698 5Fn
const byte note78 =78; //Octave 5 F# midi note no 66/ freq 739 5Fs
const byte note79 =79; //Octave 5 G midi note no 67/ freq 783 5Gn
const byte note80 =80; //Octave 5 G# midi note no 68/ freq 830 5Gs
const byte note82 =81; //Octave 5 A midi note no 69/ freq 880 5An
```

```
const byte note82 =82; //Octave 5 A# midi note no 70/ freq  5As
const byte note83 =83; //Octave 5 B  midi note no 71/ freq  5Bn

const byte note84 =84; //Octave 6 C midi note no 70/ freq  5As
const byte note85 =85; //Octave 6 C#
```

//===== END OF NOTE DEFINITIONS =====

*****/