```
#include <MIDI.h>

const byte octaves[12][6][5]  =
{
  {
    {0,   0,  0,0,0},//0
    {59, 60, 61,0,0},
    {55, 56, 0 ,0,0},
    {50, 51, 0 ,0,0},
    {45, 46, 0 ,0,0},
    {40, 41, 42,0,0}
  },
  {
    {64,65,66,0,0},//1
    {0, 0, 0,62,63},
    {0, 0, 57,58,0},
    {0, 0, 0 ,0,0},
    {0, 0, 47 ,48,49},
    {0, 0, 0,43,44}
  },
    {
    {0,67,68,0,0},//2
    {0, 0, 0,0,0},
    {0, 0, 59,60,61},
    {52, 53, 54 ,0,0},
    {0, 0, 0 ,50,51},
    {0, 0, 0  ,45,46}
  },
    {
    {69,70,0,0,0},//3
    {64, 65, 66,0,0},
    {0, 0, 62,63,0},
    {55,56, 0 ,0,0},
    {0, 0, 0 ,0,0},
    {0, 0, 47  ,48,49}
  },
    {
    {71,   72,  73,0,0},//4
    {0, 67, 68,0,0},
    {0, 0, 0 ,0,0},
    {57, 58, 0 ,0,0},
    {52, 53,54,0,0},
    {0, 0, 0,50,51}
  },
  {
    {0,74,75,0,0},//5
    {0, 69,70,0,0},
    {64, 65, 66,0,0},
    {59, 60, 61 ,0,0},
    {0, 55,56 ,0,0},
    {0, 0, 0,0,0}
  },
    {
    {0, 0,  0, 0,0 },//6
    {0, 0, 71,72,73},
    {0, 0, 67,68,0},
    {0, 0, 62 ,63,0},
    {0, 0, 57 ,58,0},
    {0, 0, 52 ,53,54}
  },
    {
    {76,77,78,0,0},//7
    {0, 0, 0,74,75},
    {0, 0, 69,70,0},
    {0, 0, 0 ,0,0},
    {0, 0, 59,60,61},
    {0, 0, 0  ,55,56}
  },
      {
    {0, 79, 80,0,0},//8
    {0, 0, 0,0,0},
    {0, 0, 71 ,72,73},
    {64, 65, 66 ,0,0},
    {0, 0, 0 ,62,63},
    {0,0, 0,57,58}
  },
  {
    {81,  82,  0,0,0},//9
    {76, 77,78  ,0 ,0},
    {0,   0, 74 ,75,0},
    {67, 68, 0 ,0,0},
    {0, 0, 0 ,0,0},
    {0,0, 59,60,61}
  },
    {
    {83, 84, 85,0,0},//10
    {0, 79, 80,0,0},
    {0, 0, 0 ,0,0},
    {69, 70, 0 ,0,0},
    {64, 65,66 ,0,0},
    {0,0, 0,62,63}
  },
    {
    {0,   0,  0,0,0},//11
    {0, 0, 0,0,0},
    {76, 0, 0 ,0,0},
    {71, 0, 0 ,0,0},
    {0, 0, 0 ,0,0},
    {0,0, 0,0,0}
  }
};




//int octave =0;
//String notename ;
//int SMnoteindex ;


void HandleNoteOn(byte channel, byte pitch, byte velocity)
{

    if (velocity != 0)
    {
        MIDImessage(1,  pitch, channel);// send noteon
    }else {

     MIDImessage(0,  pitch, channel);
    }
}

void HandleNoteOff(byte channel, byte pitch, byte velocity)
{
    // Do whatever you want when you receive a Note Off.
   MIDImessage(0,  pitch, channel);

}
    void MIDImessage(int command, int MIDInote,int MIDIchannel) {
       Serial.println(MIDInote);

    // octave = getOctave(MIDIchannel-10, MIDInote) ;

    // SMnoteindex=(MIDInote % 12)<=8 ?(MIDInote % 12+3) : (MIDInote % 12-9)      ;  // c=0
    //  A A# B  C C# D D# E F F# G G#
    //  0 1  2  3 4  5 6  7 8 9  10 11  //SMARTLIGHT SCALE
    //  9 10 11 0 1  2 3  4 5 6  7  8   // MIDI SCALE
          if(command==1){
                  Serial.write(B01000101);// command to trun led on
          }
             else  if(command==0){
                  Serial.write(B01000100);// command to trun led off
          }
                 Serial.write((MIDInote % 12)<=8 ?(MIDInote % 12+3) : (MIDInote % 12-9) );// note from array
                 Serial.write(getOctave(MIDIchannel-11, MIDInote));// octave  note chnnel here is -11 .. diff
library
    }


           int getOctave (int curString, int note ) {
      byte currOctave = 0;
            for (int octave = 0;  octave < 12;  octave++) {
                for (int fret = 0; fret < 5; fret++) {
                      if (octaves[ octave][curString][fret]==note) {
                          currOctave= octave ;
//                           Serial.print("found NOte");
//                           Serial.println(note);
//                           Serial.print("current oct");
//                            Serial.println(currOctave);
//                            Serial.println(octaves[octave][curString][fret]);
                          return currOctave;
                      }
                }
            }
       return currOctave;
      }

        void setup()
        {
             pinMode(8, OUTPUT);
          digitalWrite(8, HIGH);
            MIDI.setHandleNoteOn(HandleNoteOn);  // Put only the name of the function
            MIDI.setHandleNoteOff(HandleNoteOff);
              // Initiate MIDI communications, listen to all channels
          Serial.write(B01000000);// reset SMARTLIGHT
             delay(1000);

            // Connect the HandleNoteOn function to the library,
            // so it is called upon reception of a NoteOn.
        MIDI.begin(MIDI_CHANNEL_OMNI); // use midi setting to change baud rate
        }


void loop()
{
    // Call MIDI.read the fastest you can for real-time performance.
    MIDI.read();
}
```