

```

/*
 2-12-2011
 Spark Fun Electronics 2011
 Nathan Seidle

 This code is public domain but you buy me a beer if you use this and
 we meet someday (Beerware license).

 This code works with the VS1053 Breakout Board and controls the
 VS1053 in what is called Real Time MIDI mode.
 To get the VS1053 into RT MIDI mode, power up the VS1053 breakout
 board with GPIO0 tied low, GPIO1 tied high.

 I use the NewSoftSerial library to send out the MIDI serial at
 31250bps. This allows me to print regular messages
 for debugging to the terminal window. This helped me out a ton.

 5V : VS1053 VCC
 GND : VS1053 GND
 D3 (SoftSerial TX) : VS1053 RX
 D4 : VS1053 RESET

 Attach a headphone breakout board to the VS1053:
 VS1053 LEFT : TSH
 VS1053 RIGHT : RSH
 VS1053 GBUF : GND

 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.

 To play "Sticks" (31):
 talkMIDI(0xB0, 0, 0x78); //Bank select: drums
 talkMIDI(0xC0, 5, 0); //Set instrument number
 //Play note on channel 1 (0x90), some note value (note), middle
 velocity (60):
 noteOn(0, 31, 60);

 */
#include <NewSoftSerial.h>
NewSoftSerial mySerial(2, 3); //Soft TX on 3, we don't use RX in this
code

byte note = 0; //The MIDI note value to be played
byte resetMIDI = 4; //Tied to VS1053 Reset line
byte ledPin = 13; //MIDI traffic inidicator
int instrument = 0;

void setup() {
  Serial.begin(57600);

  //Setup soft serial for MIDI control
  mySerial.begin(31250);

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

```

```

}

void loop() {

    talkMIDI(0xB0, 0x07, 120); //0xB0 is channel message, set channel
    volume to near max (127)

    /*
    //Demo Basic MIDI instruments, GM1
    //=====
    Serial.println("Basic Instruments");
    talkMIDI(0xB0, 0, 0x00); //Default bank GM1

    //Change to different instrument
    for(instrument = 0 ; instrument < 127 ; instrument++) {

        Serial.print(" Instrument: ");
        Serial.println(instrument, DEC);

        talkMIDI(0xC0, instrument, 0); //Set instrument number. 0xC0 is a
1 data byte command

        //Play notes from F#-0 (30) to F#-5 (90):
        for (note = 30 ; note < 40 ; note++) {
            Serial.print("N:");
            Serial.println(note, DEC);

            //Note on channel 1 (0x90), some note value (note), middle
velocity (0x45):
            noteOn(0, note, 60);
            delay(50);

            //Turn off the note with a given off/release velocity
            noteOff(0, note, 60);
            delay(50);
        }

        delay(100); //Delay between instruments
    }
    //=====
    */

    //Demo GM2 / Fancy sounds
    //=====
    Serial.println("Demo Fancy Sounds");
    talkMIDI(0xB0, 0, 0x78); //Bank select drums

    //For this bank 0x78, the instrument does not matter, only the note
for(instrument = 30 ; instrument < 31 ; instrument++) {

        Serial.print(" Instrument: ");
        Serial.println(instrument, DEC);

        talkMIDI(0xC0, instrument, 0); //Set instrument number. 0xC0 is a
1 data byte command

        //Play fancy sounds from 'High Q' to 'Open Surdo [EXC 6]'
        for (note = 27 ; note < 87 ; note++) {
            Serial.print("N:");
            Serial.println(note, DEC);

```

```

        //Note on channel 1 (0x90), some note value (note), middle
velocity (0x45):
        noteOn(0, note, 60);
        delay(50);

        //Turn off the note with a given off/release velocity
        noteOff(0, note, 60);
        delay(50);
    }

    delay(100); //Delay between instruments
}

/*
//Demo Melodic
//=====
Serial.println("Demo Melodic? Sounds");
talkMIDI(0xB0, 0, 0x79); //Bank select Melodic
//These don't sound different from the main bank to me

//Change to different instrument
for(instrument = 27 ; instrument < 87 ; instrument++) {

    Serial.print(" Instrument: ");
    Serial.println(instrument, DEC);

    talkMIDI(0xC0, instrument, 0); //Set instrument number. 0xC0 is a
1 data byte command

    //Play notes from F#-0 (30) to F#-5 (90):
    for (note = 30 ; note < 40 ; note++) {
        Serial.print("N:");
        Serial.println(note, DEC);

        //Note on channel 1 (0x90), some note value (note), middle
velocity (0x45):
        noteOn(0, note, 60);
        delay(50);

        //Turn off the note with a given off/release velocity
        noteOff(0, note, 60);
        delay(50);
    }

    delay(100); //Delay between instruments
}
*/
}

//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);
}
}

```

```
//Plays a MIDI note. 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.print(cmd, BYTE);
  mySerial.print(data1, BYTE);

  //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.print(data2, BYTE);

  digitalWrite(ledPin, LOW);
}
```