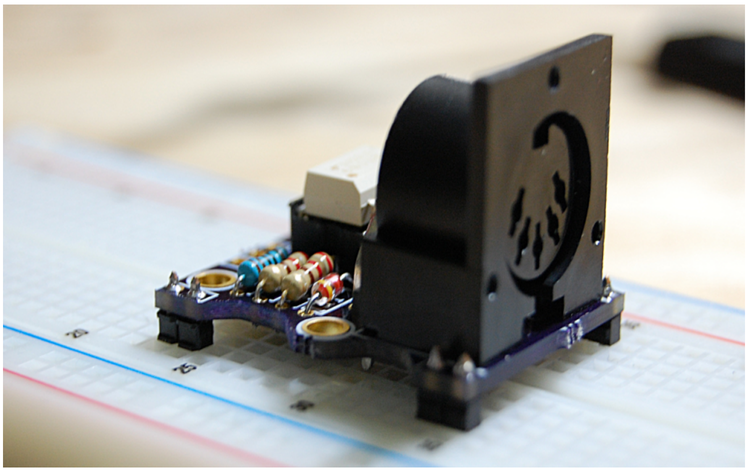
The Toymakers @ tymkrs.com
Questions? Please contact us:
feedback@tymkrs.com

DATASHEET



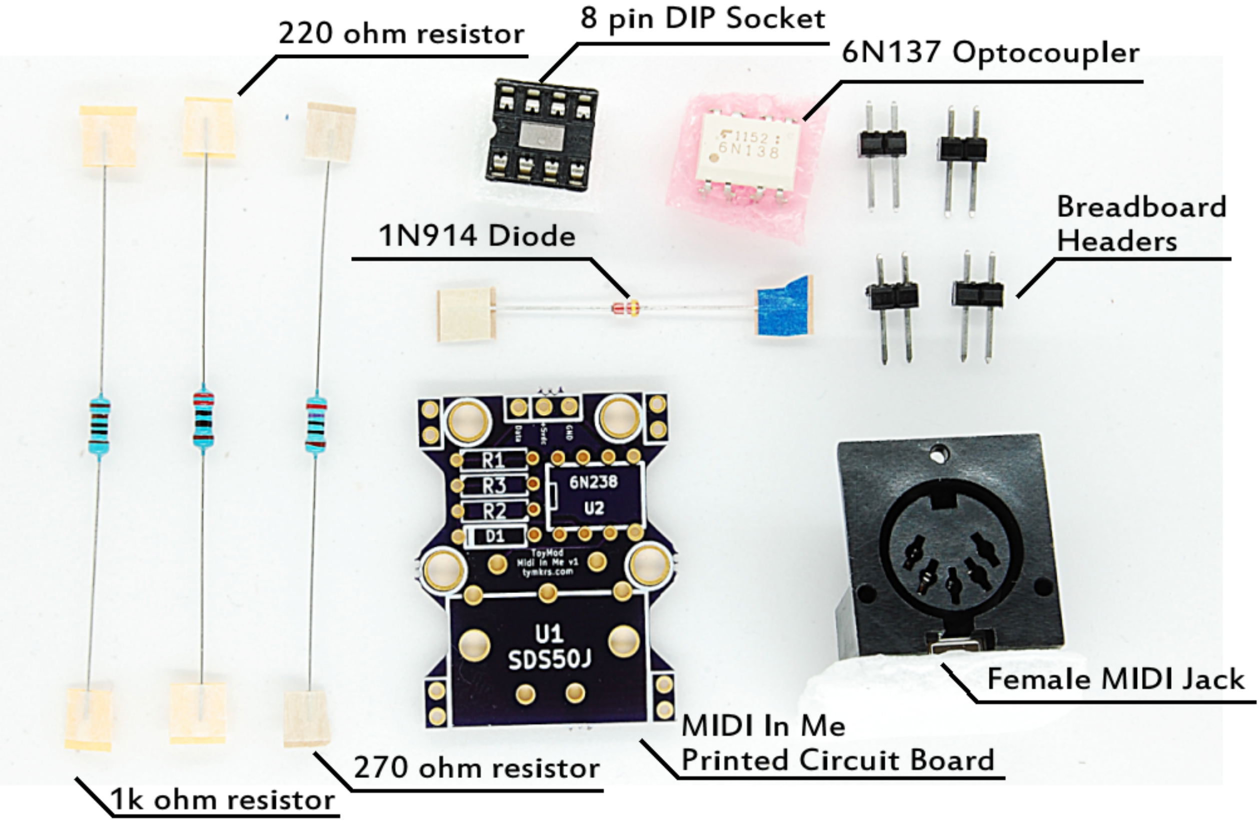
MIDI In Me

MIDI to Microcontroller Kit

The MIDI In Me kit allows you to interface any MIDI controller with your microcontroller of choice for MIDI-controlled...anything!

- Kit Type: Through-hole soldering
- Assembly instructions: In datasheet
- Function: MIDI controller to Microcontroller kit
- Uses 3 pins on the MCU to monitor incoming MIDI signals
- Designed with MIDI specifications

KIT CONTENTS



Contents of the Midi In Me Kit:

- Midi In Me printed circuit board (25.5 x 33.12 x 1.60mm)
- 4 – 1x2 male headers
- 1 – 1x3 female header
- 8 pin DIP socket
- Electrical Components

Electrical Components:

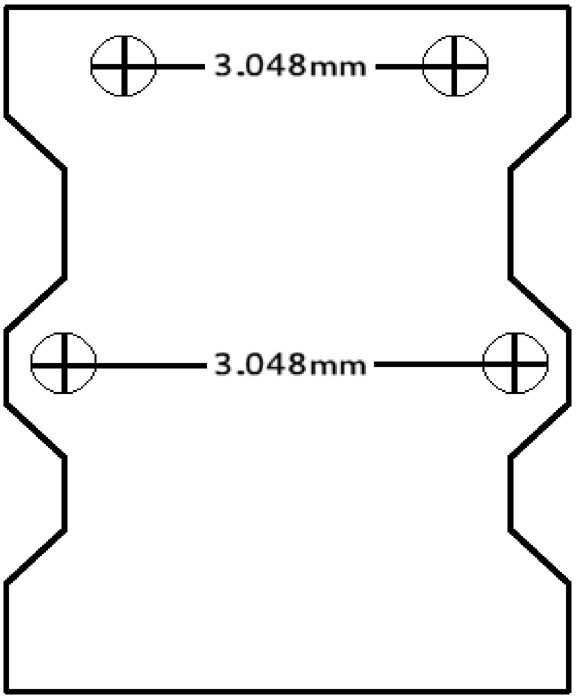
Reference	Quantity	Type	Value
R1	1	Resistor, 1/4W	1k ohm
R2	1	Resistor, 1/4W	220 ohm
R3	1	Resistor, 1/4W	270 ohm
D1	1	Diode	1N914
U1	1	Female MIDI Jack	---
U2	1	Optocoupler IC Chip	6N137/6N138

Recommended Operating Conditions

Parameter	Ratings	Unit
Supply Voltage	4.5 – 5.5	V
Operating Temperature	-40 to +85	ºC

The MIDI In Me circuit is built to MIDI specifications. Its circuit is not electrically connected to the MIDI cable and the two are optically isolated to prevent ground loops.

Mounting Holes:



Tools and material required for assembly (not included with the kit):

- Soldering iron
- Solder
- Wire clippers

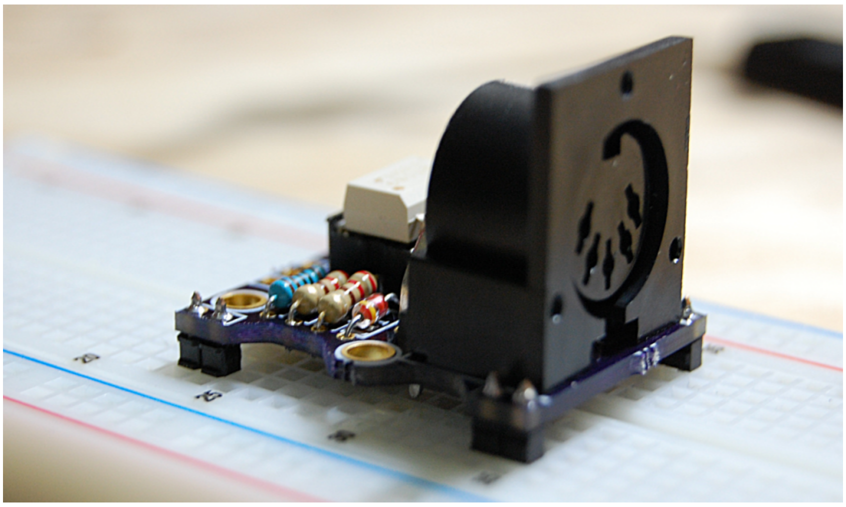
User provided items required for intended function:

- MIDI Controller
- MIDI Cable

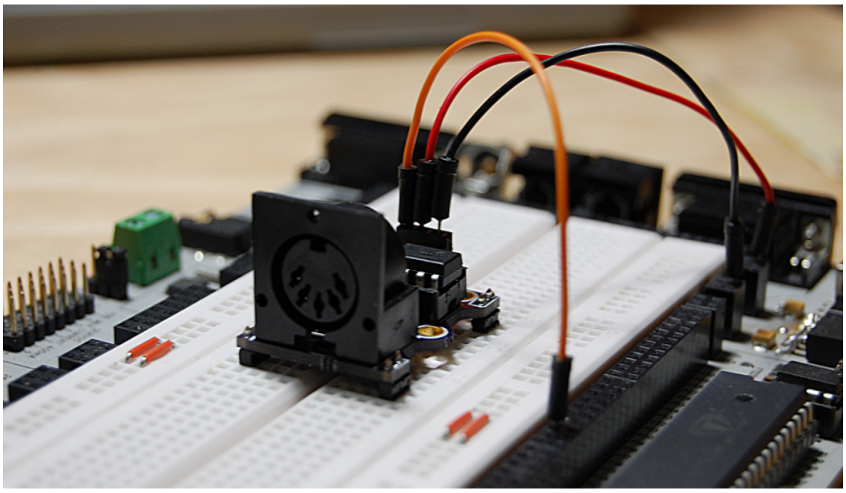
Additional physical/electrical specifications:

- Printed Circuit Board size: 1.00 x 1.30 x 0.063" (25.5 x 33.12 x 1.60mm)
- PCB thickness: 0.063" (1.60mm), not including any components
- PCB thickness: 0.945" (24mm), max height with MIDI jack
- Mounting holes: 4 holes provided. See drawings for locations and size.
- Breadboard headers are not connected to the circuit electrically – they are for stability only.

Additional Picture:



Assembled PCB
on Breadboard



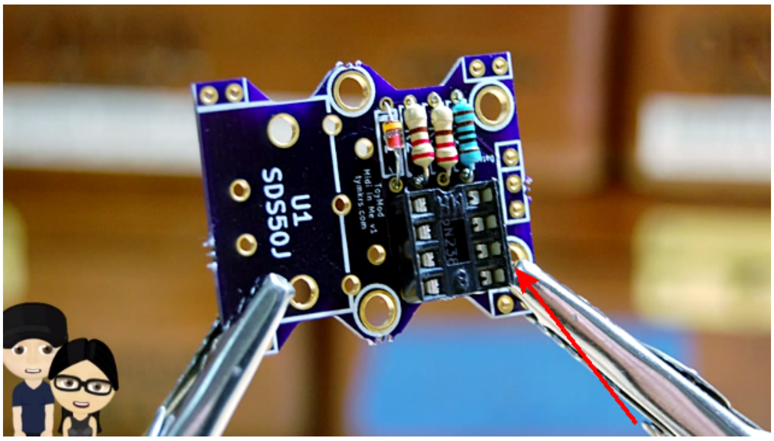
Assembled PCB
on Breadboard with
connections to microcontroller

Assembly Instructions

Build Notes:

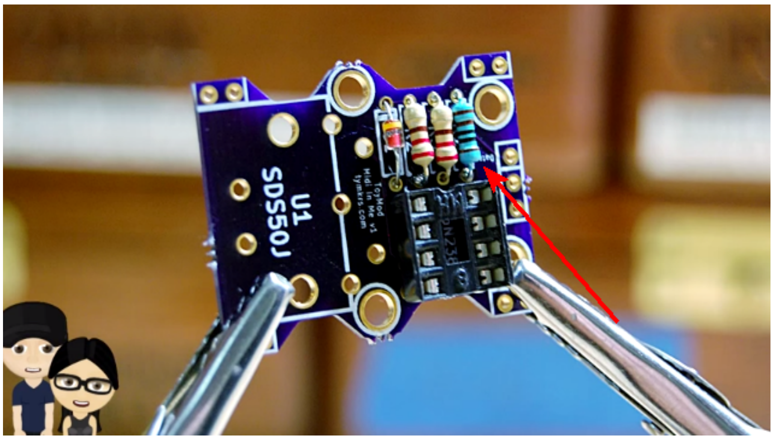
- **Datasheet for 6n137/138:** http://www.mouser.com/ds/2/149/6N138-193856.pdf
- **Method of use:** The connections run from the midi keyboard via 5pin DIN cable to Midi In Me. Midi In Me is connected to power, ground, and a monitored pin on Propeller (or other MCU). The microcontroller is then connected through USB to your computer/terminal.

Step 1: Put in the components



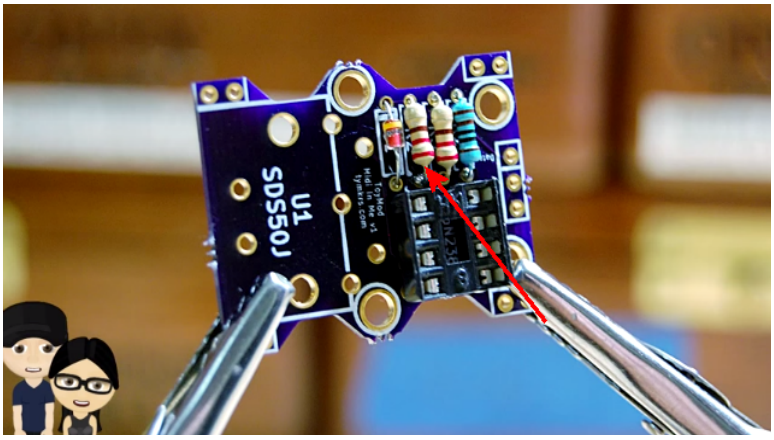
8 Pin DIP Socket

Match the semi-circular divot to the graphic on the PCB



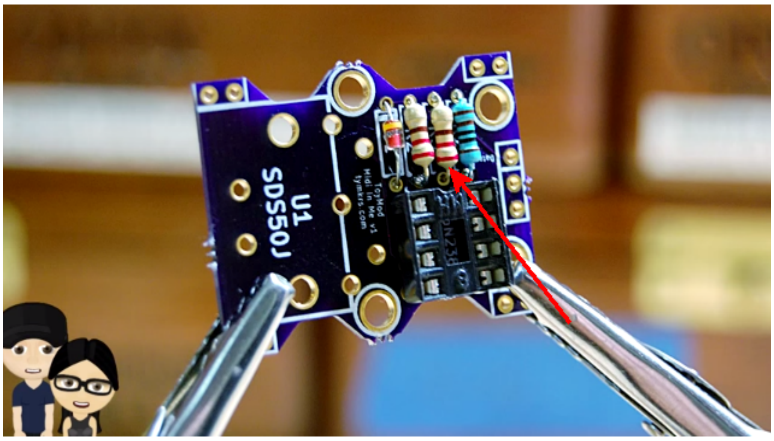
R1: 1k ohm resistor

You can bend the leads before putting it in the PCB. Polarity does not matter.



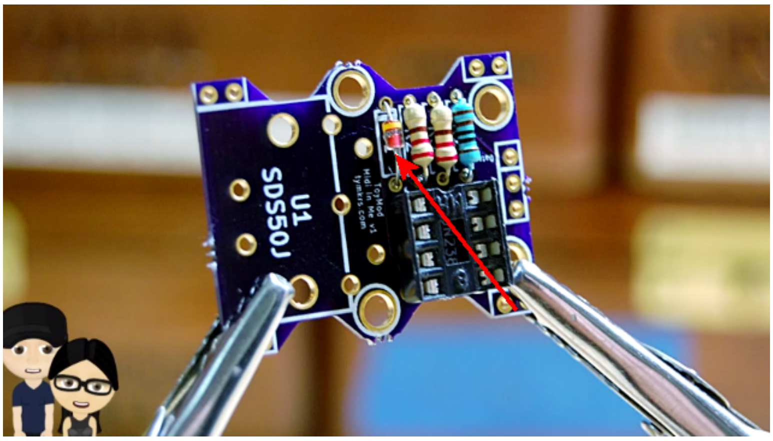
R2: 220 ohm resistor

You can bend the leads before putting it in the PCB. Polarity does not matter.



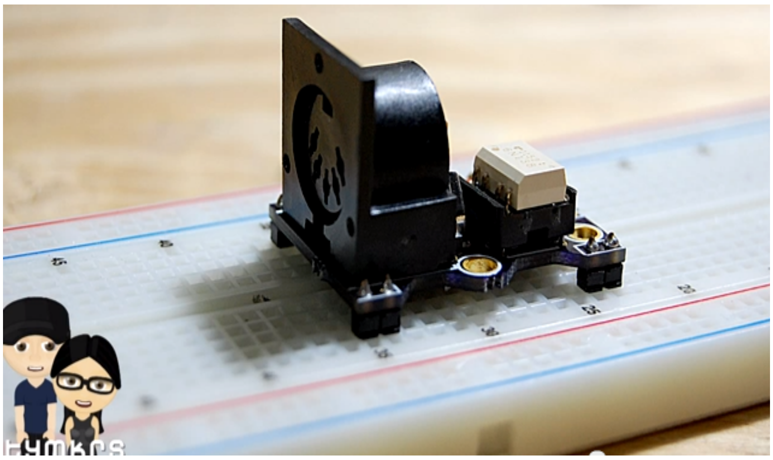
R3: 270 ohm resistor

You can bend the leads before putting it in the PCB. Polarity does not matter.



D1: 1N714 diode

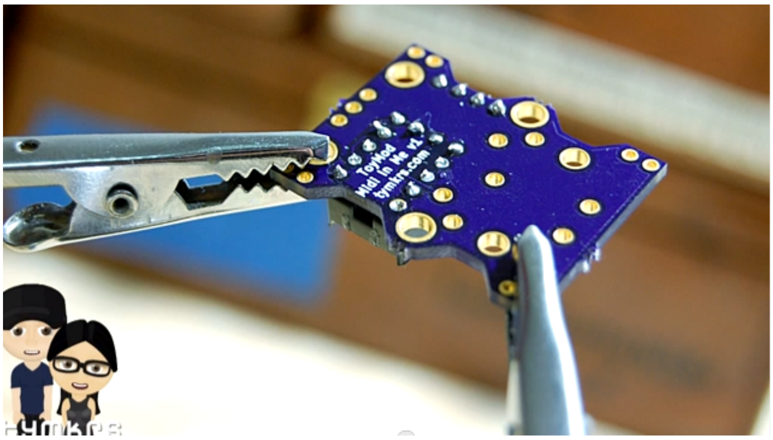
You can bend the leads before sticking it into the holes. The peach colored line should match the line drawn on the PCB



U1: MIDI Jack

There's only one way to put the jack in – it will face outwards so you can plug the MIDI cable in!

Step 2: Solder in the electrical components and clip the extra leads!



I use 60/40 0.38mm gauge solder for these pads. But also have 1.3mm gauge solder for the larger solder pads.

Using nibbers or nail clippers, trim the extra leads off of the electrical components!

Example Code

Can be found: http://www.tymkrs.com/code/MidiInMe_Demo_1.spin.

```
File: MidiInMe_Demo_1.spin
Version: 1.0
Copyright (c) 2013 Tymkrs
See end of file for terms of use.

Author: Whisker
```

```
HISTORY:
  This demo is made as an example for using the MidiInMe ToyMod kit from
http://tymkrs.com/
  It uses the MidiIn object by Tom Dimock available here
http://obex.parallax.com/object/387

USAGE:
  • Connect MidiInMe Data pin to Propeller Pin MidiInPin.
  • Connect MidiInMe +5vdc pin to the regulated +5vdc (or regulated 3.3v) powering your
Propeller.
  • Connect MidiInMe GND pin to the same ground as your Propeller.
  • Connect MidiInMe 5 pin DIN (midi input) to a MIDI output device; such as a digital
piano, drum kit, or computer.
  • Use Parallax Serial Terminal to monitor NoteOn, NoteOff, and Controller midi events.
```

```
CON

  'Initialize the system clock speed.
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  'Which Pin is MidiInMe connected to?
  MidiInPin                     = 12

  'Mask values for selecting which Midi Events to monitor.
  doNoteOn                      = $00000001
  doNoteOff                     = $00000002
  doAftertouch                  = $00000004
  doController                  = $00000008
  doProgramChange               = $00000010
  doChannelPressure             = $00000020
  doPitchWheel                  = $00000040
  doSysex                       = $00000100
  doMTC                         = $00000200
  doSongPosPtr                  = $00000400
  doSongSelect                  = $00000800
  doTuneRequest                 = $00001000
  doMidiClock                   = $00002000
  doMidiTick                    = $00004000
  doMidiStart                   = $00008000
  doMidiContinue                = $00010000
  doMidiStop                    = $00020000
  doActiveSense                 = $00040000
  doReset                       = $00080000

OBJ

  PST: "Parallax Serial Terminal" 'As they occur, Midi events are printed to a Parallax
Serial Terminal window on your PC.
  MIDI: "MidiIn" 'The driver which reads the MIDI data stream, parses the MIDI commands,
and returns the MIDI events.

PUB Main

  'Start Parallax Serial Terminal.
  PST.Start(115_200)
  'Wait 1 second for user to enable Parallax Serial Terminal software.
  waitcnt(clkfreq + cnt)

  'Print greeting to serial terminal.
  PST.str(String("Midi In Me Demo, Running!", 13))

  'Start Midi In driver on MidiInPin, listening for NoteOn, NoteOff, and Controller
events.
  MIDI.start(MidiInPin, doNoteOn+doNoteOff+doController)

  'Do this over and over, forever.
  repeat
    ScanKeys 'Call the ScanKeys Function.

Pub ScanKeys | Event

  'This function checks if there was a midi event since the last time it checked.
  'If so, it checks what type of type of event (0, 1, or 3) (NoteOn, NoteOff, or
Controller).
  'If the event was one of these three types, it prints the event to the serial terminal.

  Event := MIDI.evtCheck
  case Event
    -1:
    other:
      case Event.byte[3]
        0:
          PST.str(String("Channel #"))
          PST.dec(Event.byte[2])
          PST.str(String(" Note On #"))
          PST.dec(Event.byte[1])
          PST.char(13)
        1:
          PST.str(String("Channel #"))
          PST.dec(Event.byte[2])
          PST.str(String(" Note Off #"))
          PST.dec(Event.byte[1])
          PST.char(13)
        3:
          PST.str(String("Channel #"))
          PST.dec(Event.byte[2])
          PST.str(String(" Controller #"))
          PST.dec(Event.byte[1])
          PST.str(String(" Value: "))
          PST.dec(Event.byte[0])
          PST.char(13)
```