

DIY MIDI controllers using PIC microcontrollers and Basic Stamps

Ross Bencina

[home page](#) [email](#)

Page last updated 6 June 2007



Corporate Web Filter

Filter Websites, Messenger, Gaming, SMB
Market Leader! Download Trial

Extreme Low Power MCUs

20nA Sleep, Real Time Clock & Watchdog
Timer

Ads by Google

Introduction

By combining a PIC microcontroller or Basic Stamp II with a few passive components it is not difficult to construct your own knob box, trigger box, or other MIDI input device. A knowledge of electronics may be required for interfacing to some sensors. Both the Stamp and PIC provide an economical (under US\$100) entry point into the world of alternative MIDI controllers.

This page was created to provide a starting point for people interested in building their own MIDI control devices using Microchip PIC microcontrollers or Parallax Basic Stamps. If you have any comments or suggestions please feel free to email me.

Basic Stamps and PICs



A Basic Stamp consists of a tiny circuit board on which is soldered a PIC CPU, a timing crystal, and some EEPROM memory for program storage. It provides 16 bidirectional data (digital input/output) pins, which can be interfaced to potentiometers, a simple MIDI output circuit, A/D converters, etc. As its name suggests the Basic Stamp executes code written in BASIC, which is downloaded onto the stamp using a serial interface. [Parallax, Inc.](#) manufacture a number of different Basic Stamp models, the Basic Stamp II and IISx

are generally considered fast enough for MIDI transmission applications, but neither supports MIDI input satisfactorily.



[Microchip Technology Inc.](#) manufacture a large range of PIC microcontrollers. For small projects the PIC16F84 is a common choice as it is cheap and uses reprogrammable flash ROM for program storage. The PIC16F84 chip contains 1k words of program flash ROM, 68 bytes of RAM and 64 bytes of EEPROM for persistent data storage. The PIC16F84 has 13 bidirectional data pins which can be used in a similar fashion to those on the Basic Stamp. PIC chips are programmed in assembly language which is then stored onto the chip using a special programmer. For PC users a parallel port based programmer for the PIC16F84 can be built for around \$30. Both Microchip and Parallax distribute free PIC assembler software for the PC.

The general opinion is that the Basic Stamp II is a better starting point for people with little or no programming or electronics experience, as using a PIC directly requires you to learn PIC assembly language which can be tedious if you're not into that kind of thing. If you're a programmer or you're looking at using more than one unit the PIC16F84 is good choice as it's cheap and can often be made to do things faster than the Basic Stamp. You can assemble a programmer, a few PIC chips and some power supply circuitry for less than the cost of a Basic Stamp II.

See Peter H. Anderson's article [Working with Stamps and PICs](#) for an extensive discussion of the relative merits of PICs and Basic Stamps.

Interfacing to MIDI

For both the Basic Stamp II and medium range PIC chips, MIDI (i.e. serial) output is implemented in software by manipulating the state of a data pin. On the Basic Stamp this is achieved using the SEROUT command, on the PIC you have to roll your own MIDI output routine - this can be tricky as the serial timing is determined by the number of instructions executed and the clock speed of the PIC, however there are plenty of examples available on the web to get you started.

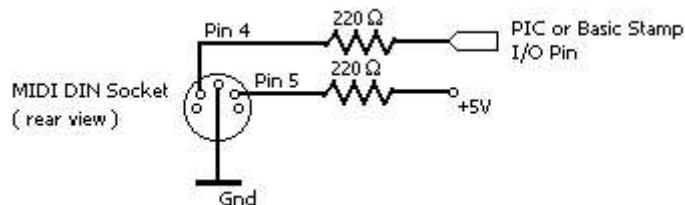
Neither the medium range PICs (such as the PIC16F84) nor the Basic Stamp (any model) are particularly suited to simultaneous MIDI reception and transmission as they do not support buffered serial communications. On the Basic Stamp II, any MIDI coming in while the Stamp is processing will be lost. On a PIC, code must be carefully designed to interleave serial reads and other processing at the sub-MIDI bit level to avoid loss of data.

MIDI input without output is possible on a PIC, but is not described here - see the links section for some examples. Implementing simultaneous MIDI in and out, or soft thru / merge is impossible using a Basic Stamp II. It is difficult to implement on the PIC16F84 and leaves little room for other processing to be performed. A suggested solution is to use two PICs - one for input and one for output, or to use a more expensive PIC such as the PIC16C7x which has an on-chip UART.

MIDI output

Wiring up a PIC or Basic Stamp for MIDI output couldn't be simpler:

- Connect pin 4 of the MIDI connector to the desired output pin of the PIC or Basic Stamp via a 220 Ohm resistor
- Connect pin 5 of the MIDI connector to the +5V via a 220 Ohm resistor
- Connect pin 2 of the MIDI connector to ground



Important Note! A couple of people have written to me suggesting that the pins 4 and 5 on the MIDI output wiring diagram are reversed. I haven't verified whether this is true or not. You may need to experiment by swapping the connections if it doesn't work.

See either David B. Thomas' [MIDI Sender](#) (Parallax assembler) or my [midisend](#) (Microchip assembler) program for an example of sending MIDI using a PIC chip. For the Basic Stamp II see Jeff Mann's [MIDI out program for BASIC Stamp II](#).

If you're looking for information about MIDI, visit [The MIDI Technical Fanatic's Brainwashing Center](#). The [MIDI electrical specification diagram](#) along with other useful information is available from the [MIDI Manufacturers Association website](#).

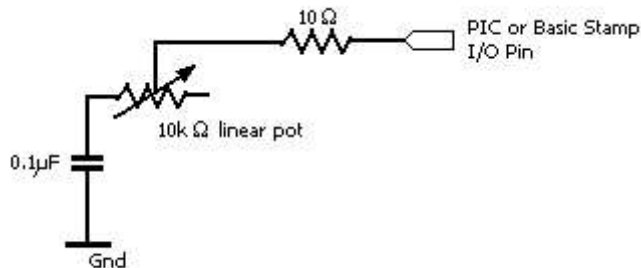
Interfacing to variable resistances

Although this section discusses potentiometers (as used with Knobs, Faders and Joysticks), the information can also be applied to other variable resistance devices such as Light Dependent Resistors (LDRs) or Forces Sensing Resistors (FSRs).

The dual input / output nature of PIC and Basic Stamp data pins allows a simple circuit to be constructed that can measure the value of a potentiometer. There are two different circuits commonly used for performing this measurement: the first was used in conjunction with the Basic Stamp I POT command and is useful on the PIC as source code to drive it is freely available (this is the version described below). The Basic Stamp II replaced the POT command with the RCTIME command which requires a different circuit - see the Stamp II manual for details or John Radenberg's [Basic Stamp DIY knob box page](#) for a functioning example.

The POT method

A measurement is made by first charging the capacitor (making the data pin output a high), then setting the pin to input and slowly discharging it (by toggling the pin between input and output). The time taken for the pin to go low corresponds to the time it takes for the capacitor to discharge which is related to the resistance of the potentiometer.

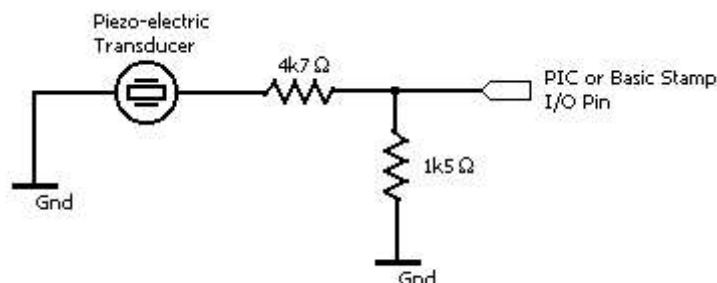


The Basic Stamp I POT command performs the above process and returns a scaled output value. See [Scot Edwards' Parallax assembler version of the POT command in The PIC Source Book](#) for an example of using the above circuit with a PIC chip.

On a 10Mhz PIC16F84 the above circuit takes 6ms to charge and at maximum resistance takes 138 counts (0.5ms) to decay. An advantage of the PIC over the Basic Stamp when multiple pots are used is that all RC circuits can be charged at once, providing a significant speed increase over sequentially charging and discharging each circuit.

Interfacing to piezo-electric transducers

In the February 1998 issue (vol. 27, no. 3) of [Electronics Today International](#), an article by Tom Scarff entitled *MIDI Drum Pads* describes the construction of an 8 input piezo trigger to MIDI unit based on a PIC16C84 microcontroller. The unit generates MIDI messages when the triggers are struck, no velocity information is measured however. [Tom's article including source code and schematics is available online.](#)



The circuit above illustrates the interface between a piezo transducer and a PIC data pin. When the transducer (or a surface to which the transducer is attached) is struck, a voltage is generated which may be detected at the data pin. Suitable software would scan the data pin(s) and transmit a MIDI message when a high (1) logic state is detected.

[PAiA](#) have some more complex [schematics for interfacing percussion sensors.](#)

Further directions

The web contains many examples of interfacing PICs, Basic Stamps and other microcontrollers to switches, A/D converters and other devices that may be usefully employed to construct interesting MIDI controllers. Dan O'Sullivan's [Physical Computing Page](#) is a great source of information about mediating between the physical and digital worlds.

By combining an A/D converter and a multiplexer chip a MIDI controller with increased resolution, faster update time and a large number of pots, sensors or other input devices could easily be constructed. For the more adventurous, Microchip manufactures more complex PICs supporting faster clock speeds, on-chip A/D converters, timers and serial interfaces.

PIC and Basic Stamp MIDI links

Some of the links below are no longer valid - if you know of the new location of any of these pages, please let me know so I can update the links. If you want to access the content of a broken link, try the [wayback machine](#) - just enter the URL of the document which wasn't there and the wayback machine will retrieve an archived version.

MIDI output

- [Jeff Mann's MIDI out program for the BASIC Stamp II](#)
- [John Radenberg's Basic Stamp Knob Box DIY Project](#)
- [The Trueman-Cook R-Bow uses a Basic Stamp II to generate MIDI from force sensing resistors and accelerometers, among other things](#)
- [Tom Scarff's MIDI projects include a MIDI voice harmoniser and MIDI drum pads as published in ETI magazine - both use the PIC16C84](#)
- [Mike Whitco's PIC Microcontroller based Midi kick pedal](#)

- [David Sorlien's MIDI Footswitch Project](#)
- [Thorsten Klose's recently redesigned MIDIbox Plus is a PIC16F84 based knob box utilising a Maxim MAX186 A/D converter](#)
- [The MIDI Program Change Unit by Maila Fatticcioni and Massimo Grasso performs MIDI input and output](#)
- [Paul Messick's designs for a PIC16C54 based MIDI Metronome and PIC16C57 based MIDI Controller Footpedal](#)
- [DeeT's PIC Page includes a MIDI Sender program](#)
- [ISE1 Project: MIDI Output From Piano Foot Pedals](#)
- [Jerome Dumas' plans for a MIDI Knob Box using the PIC16C7x](#)
- [Dylan Menzies sells MANMIDI \(Multichannel Analog to MIDI Converter\), a pre-programmed PIC16C71. You can now download the hex image from his site and make your own.](#)
- [Kashi's PSX \(PlayStation\) controller to PC MIDI converter is interesting because it connects directly to the PC game port MIDI interface without any buffering circuitry.](#)
- [Everyday Practical Electronics Magazine \(EPE\) have recently published some MIDI PIC projects, source code is available from The Official EPE PIC Source Code Page.](#)
- [Marco Della Rocca's schematics, source code, manuals and prebuilt 8 knob controller using the PIC 16F84. All in German](#)
- [Admir Salahovic's eDrum - Trigger MIDI Converter using PIC 16F877. With full schematics, PCB designs and firmware.](#)
- [Kazuhiro Iida's MIDI Control Chip Kit are preprogrammed PIC chips for adding keyboard-to-MIDI support to an existing piano or organ keyboard. Includes a tutorial on retrofitting MIDI sensors to a piano action.](#)
- [Angelo Fraietta has a number of PIC based DIY kits including the Dumb Controller Kit and MiniCV. Angelo is happy to help and will give you his schematics if you ask him.](#)
- [Nicholas Appleton has schematics and source code for an easily expandable device to fit into existing analogue organs to give them MIDI out using the PIC16F77A.](#)
- [Josh Harle's source code and circuit schematics for a 4Mhz RC based PIC16F84 MIDI controller.](#)
- [Otis Irace provides source code and schematics for his PIC MIDI Patch Controller which uses a PIC16F877A Micro and a 2 pin LCD Character Display and sends MIDI control changes in response to buttons and potentiometers.](#)

MIDI input

- [Trevor Page's Simple PIC16F84 Based MIDI to CV Converter](#)
- [Callipygian.Com has software, documentation and schematics for receiving MIDI note ON commands to drive TTL outputs with a PIC16F84](#)
- [Patrice Rabby has code, schematics and PCB designs for decoding MIDI to 32 outputs to control switch relays, robots or lights using the PIC16F84.](#)
- [Old Crow's PIC projects including a MIDI-to-Trigger device](#)
- [Decoding MIDI with a 17c42 PIC Processor](#)
- [PIC 16C63 MIDI controlled Light dimmer](#)
- [John Blacet sells a pre-programmed PIC that converts MIDI clocks and sequencer commands into logic pulses](#)

- [Bob Lang's site includes: A MIDI controlled harpsichord using 16f877, A MIDI controlled xylophone using 16f877, and A MIDI interpreting through box using 16f873](#)

Other

- [Bill Coghill's Stamp MIDI Information Page](#)

General PIC links

The manufacturer of PIC chips [Microchip Technology Inc.](#) maintains a large website providing technical information, application notes and free development software for the PIC. [Parallax, Inc.](#) maker of Basic Stamps provide mailing lists for PIC and Stamp developers and also have free development software available online.

Peter H. Anderson teaches PIC and Basic Stamp programming, his [Embedded Processor Control](#) web site is a goldmine of projects, kits and other useful information. If you only look at one other site, make it this one.

[Dontronics](#) sell various PIC related products including the SimmStick development platform. Don maintains a great set of PIC links and resources, and also hosts an online version of Scot Edwards' [The PIC Source Book](#) which contains assembly language versions of the Basic Stamp instruction set - very useful!

Based in the United Kingdom, Russell Warburton's [Warburton Technology](#) specialises in the distribution of Parallax Basic Stamps and other microcontroller based products and development tools.

Many companies manufacture PIC programmers, most designed to work with the more expensive PIC chips. There are many DIY PIC programmer circuits on the web, especially for the cheaper PICs such as the PIC16F84. I use a programmer based on Michael Covington's eminently affordable [No Parts PIC Programmer](#) which can be purchased as a kit from [Oatly Electronics](#).

If you've decided to get involved in PIC development consider the [PICList](#), a high-traffic mailing list which provides a supportive environment for discussing PIC software and hardware development.

There's so much general information on the web about using PICs and Basic Stamps that your next stop should be a search engine - I use [Google](#).

Reader's comments

Jose Souto wrote to let me know that:

"There are a number of free C compilers for PIC and I'm very satisfied with the CC5xFree from [www.bknd.com](#). With this compiler fully integrated with MPLAB from Microchip, people willing to use PIC don't need to rely on Assembly to write code for MIDI.

Last year I wrote a small C snippet to read MIDI with a PIC12C508 using a bit bang C code written with CC5xFree. This is from my experience in writing a full MIDI receiver for a customer. I posted it at the [PicList](#) "

Peter M Olsen wrote to me about the Picaxe which can be programmed in BASIC:

"Have you looked at the Picaxe? ([www.picaxe.co.uk](#)). I've been using them for about 6 months now and as far as I am concerned they leave the Basic Stamp for dead! They are basically a Microchip PIC with a bootstrap loader. I'd love to get one working with MIDI."

Acknowledgements

Thanks to:

John Radenberg of [Basic Stamp DIY knob box page](#) fame, who introduced me to many of the ideas on this page.

Jim Sosnin whose work with PICs and MIDI gave me the confidence to attempt constructing my own circuits.

Jeff Mann for numerous suggestions and corrections.