

The MorseMeter

Don't look—listen—to your “talking” CW multimeter!

Recently I overheard an intriguing on-air discussion of speech-equipped digital multimeters. A blind amateur was explaining to a friend that his Micronta “Talking Multimeter” had failed and he could not find the custom ICs to repair it. Later that day I did a search on the Internet for other makers of speech-capable multimeters and was surprised, in this modern day of digital integration, that there was only one model being manufactured. It was very expensive and it lacked what I would consider an important feature—an audible continuity indication. This oversight is especially strange in a meter that is otherwise intended for use by ears rather than eyes.

A Solution to a Need

Over the next few days I considered the need for a multimeter that would announce its readings. With a chill down my spine, I imagined what it would be like if I were to lose my sight. Among many other problems, I would really miss being able to make voltage, current and resistance measurements. I'd been looking for a project that would stimulate me to learn more about PIC (programmable interrupt controller) ICs. It struck me that a PIC device combined with one of the digital multimeters that have a RS-232C serial communications port might make it possible to announce the meter readings.

An Internet Web search or two, discussions with friends, and a bit more thought got me started. I learned that the PIC chips were programmed using a simple interface connected to the parallel printer port of a PC. The software was written in a relatively simple language and then compiled for loading into the chip's non-volatile memory. And then, the IC was transferred to your own project.

Recalling that I'd seen some PIC programmer and training kits on the DIY Electronics “Kits-R-Us” Web site (a great source for electronic kits of all varieties),¹ I selected their “Simple PICmicro Programmer/Experimenter Kit #81.” This kit² is intended for use with the 16F84A microcontroller from MicroChip,³ which seemed very popular with experimenters. The kit includes a PIC, a tutorial manual,⁴ a programmer board, programming software and some code examples.

The DIY K-81 is available from dealers worldwide, but I purchased mine for a mere \$15 from Amazon Electronics in Columbiana, Ohio.⁵ They also sell a variety of other suitable PIC programmers.

While waiting for the kit to arrive I researched the technical details of digital multimeters with RS-232C serial ports. These meters are equipped with matching software to run on a personal computer. I already owned one such meter, the Metex ME-11, but I'd never tried its computer interface. The sup-

plied software loaded easily and ran fine on my Windows 95 PC. Clearly, it would be very handy for graphically displaying the results of testing and the gathering data over time on the hard drive of the computer. One could probably add one of the speech synthesizer programs for the visually impaired and read the on-screen results aloud, but my interest for this project was to eliminate the computer and build a small device to attach directly to the multimeter.

Fortunately, the Metex meter manual included the details of the serial communications protocol. The meter has a female DB-9 connector, and transmits and receives data as 7-bit ASCII with no parity and two stop bits—at 600 baud. In the shorthand often used for RS-232C connections this would be described as “600-7N2.” I ran one of my favorite terminal programs on a laptop computer, SIMPTERM,⁶ and confirmed that when I sent a letter “D” or a carriage return to the meter, it replied with the readings. For example, after I hit the ENTER key it might reply, DC 42.034 V <CR>. “Very cool,” I thought, “This is going to work!”

But what about other makes and models of digital multimeters? I found that while most meters equipped with a PC interface use normal ASCII data on a regular RS-232C port, some do their own thing. For example, the currently available RadioShack model 22-812 with a DB-9 connector looked like a good candidate but after I bought one I found all is not as it seems. The data you get from it is not a value, as you might expect, but rather, it sends information about the state of the segments of the LCD display. It works with its own program, but not for our project. Luckily, there are many meters available that do respond with easy to interpret data. The Metex⁷ line of meters is a good choice, as well as the Mastech⁸ brand models, but there are many others that will work. The RadioShack 22-805 appears to be the same meter as the Metex ME-11, so that's a possibility, as well. Do an Internet search for “PC interface multimeters,” or something similar, and you'll find a bunch.

Programming

The DIY kit arrived and I built the programmer board in a few minutes—it can be seen in Figure 1. The tutorial manual



¹Notes appear on page 00.

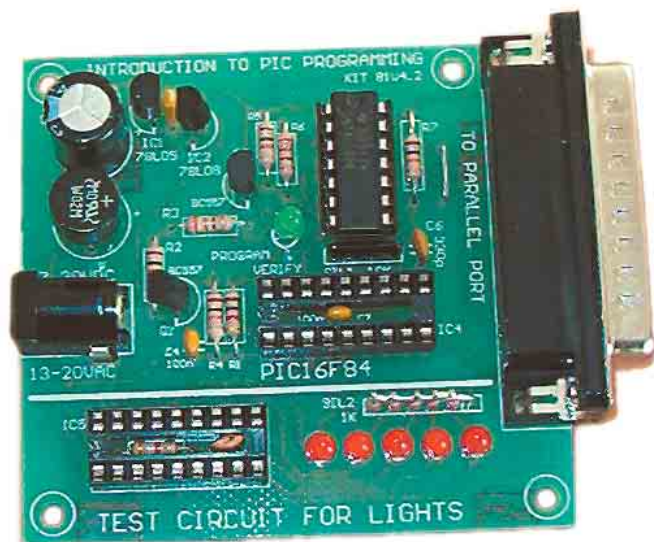


Figure 1—The DIY Kit-81 programmer board.

is excellent and within the hour I was able to successfully write a program to flash the test LEDs on the programmer board. Looking deeper into the datasheet documentation of the 16F84A microcontroller, I learned more about its serial communications and other I/O, and how to structure the programs in its memory. The chip understands only raw hexadecimal codes from the programmer, but there are a variety of compiler programs to create the HEX code from various computer languages. Generally speaking, the software is written in a text editor or a word processor to produce a file (filename.txt) and then compiled into a file (filename.hex) for loading into the PIC chip via the programmer board. The IC is then moved to the final circuit for operation.

Here's a handy tip to avoid damage to the PIC chip as it is frequently moved back and forth from the programmer socket to the prototype's IC socket during testing. Put the PIC chip into a machine-pin IC socket and move the PIC and the socket back and forth, as an assembly, from board to board. The bare PIC pins are fragile, but the machined pins will hold up a lot better. It may seem odd to plug a socket into another socket, but it works well.

As I asked questions of my friends more experienced with microcontrollers, one of my stalwart pals, Tim Hynde, KA8DDZ, stepped forward to assist. Tim has built many PIC projects, including a homebrew CW decoder, and said he could whip up some code for the 16F84A that would do everything I wanted, in short order. Tim prefers to work in the PicBasicPro environment from Micro-Engineering Labs,⁹ and this was of great interest to me as I had experience in BASIC years ago—during the older days of personal computing (with Commodore and RadioShack TRS computers). As we worked together by e-mail, we started to call this the MorseMeter project.

Tim produced a program that night which accomplished the basics. A momentary switch closure to ground starts the process, sending a character out on the serial transmit line, then the program reads the returning ASCII characters on the receive line. As each character is decoded, the program jumps to the appropriate subroutine to make the dits and dahs of CW on an output line. Using information in the PicBasicPro and MicroChip documentation, we designed a prototype of the MorseMeter device to test the software. After several edits and additions by each of us to both the hardware and the software, we came up with fully functional versions.

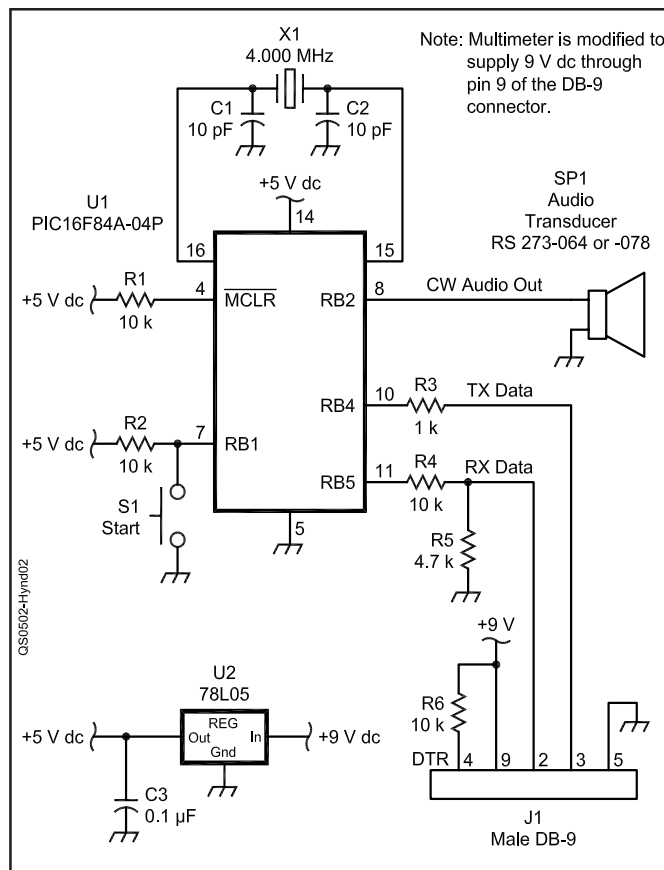


Figure 2—The MorseMeter schematic and parts list. Parts can be obtained from many suppliers, including Digi-Key (www.digikey.com), RadioShack (noted below as RS) (www.radioshack.com) and Mouser (www.mouser.com).

- C1, C2—10 pF capacitor (ceramic), RS 272-809, Digi-Key BC1001CT-ND.
- C3—0.1 μF capacitor (monolithic), RS 272-1069 or 272-109.
- J1—DB-9 male connector, RS 276-1537.
- R1, R2, R4, R6—10 kΩ, 1/4 W resistor, RS 271-1335.
- R3—1 kΩ, 1/4 W resistor, RS 271-1321.
- R5—4.7 kΩ, 1/4 W resistor, RS 271-1330.
- S1—Momentary pushbutton switch, RS 275-1571.
- SP1—Piezo audio transducer, RS 273-064 or 273-078.
- U1—IC, PIC16F84A-04/P (MicroChip), Digi-Key PIC16F84A-04/P-ND.
- U2—78L05 5 V regulator, Mouser 512-LM78L05ACZX, Digi-Key NJM78L05A-ND.
- X1—4.0 MHz crystal, Mouser 520-HCU400-S.
- 18-pin IC socket, RS 276-1992.
- 18-pin machine pin socket, Digi-Key AE7218-ND.
- Experimenter PC board, RS 276-148.
- Enclosure box, RS 270-1801.

Just as there are a variety of digital multimeters with serial ports, there are many variations possible on the MorseMeter program—different serial bit patterns and baud rates, unique sets of ASCII characters sent and received, or interface timing differences, for example. Different bits and bauds are simply software changes and a bit of study in the PicBasicPro manual will reveal the nature of the SEROUT and SERIN parameters the builder will need for the meter in question. If a different character starts the meter output, then just change the final parameter on the SEROUT2 line for the ASCII decimal value required. If different letters, numbers, or symbols are expected from the meter, one can add other MYBYTE subroutines with the dit and dah patterns for the desired letters.

I originally thought we would need to include RS-232C

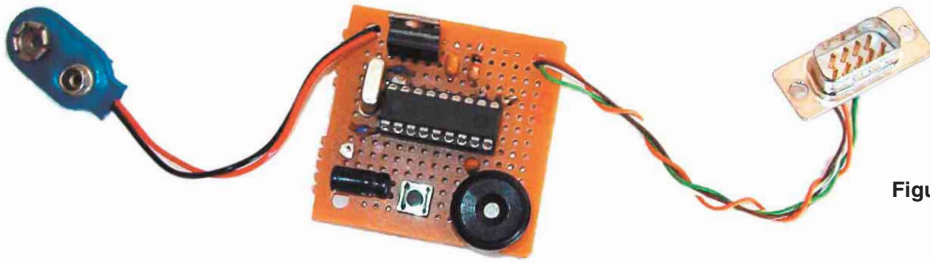


Figure 3—The prototype assembly board.

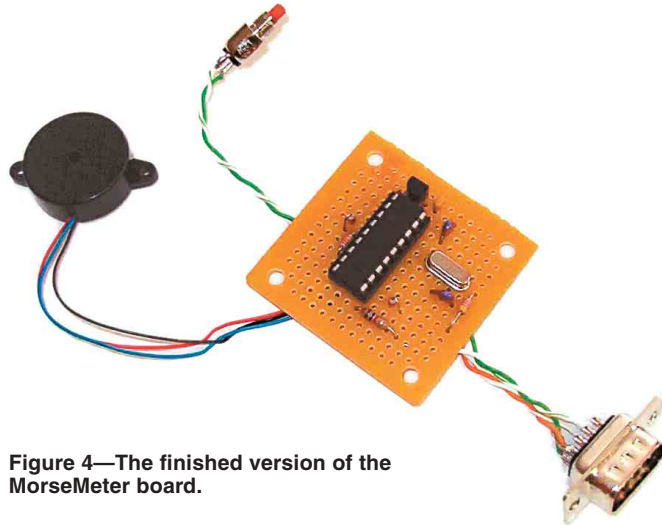


Figure 4—The finished version of the MorseMeter board.



Figure 5—The enclosed MorseMeter unit.

driver and receiver ICs (such as the MAX232 series), but we found that the 16F84A microcontroller is capable of doing a fair job of emulating an RS-232 port by itself. It might not work with every multimeter but, so far, it has done an excellent job. Experiment with the values of the resistors on the TX and RX data lines if it doesn't work immediately. I also expected that we would need to use the CW output of the PIC to key a tone oscillator, or a Sonalert-type "beeper," but Tim found that, with the right commands, the PIC could be made to output a tone. It's not particularly clean, and sounds rather harsh, but the simplicity is hard to beat—just connect a small audio transducer (RadioShack 273-078) directly to the output line and you're in business. Figure 2 shows the schematic of the MorseMeter, together with a parts list. The source code listing and the machine code HEX file, ready for programming, along with a PC board layout and other construction details can be found on the ARRL Web site.¹⁰

Building It

My original prototype of the design is shown in Figure 3. It worked flawlessly on my Metex ME-11, but with the pushbutton and transducer mounted right on the circuit board, it was not the physical design I wanted. I made a second version with low-profile components, and put it into a small plastic project box. The START pushbutton, DB-9 connector, and audio transducer were mounted on the sides of the enclosure. I modified the Metex meter to provide switched 9 V dc on pin 9 of its DB-9 connector, eliminating the need for a separate battery and power switch. Figure 4 shows the final, finished board and the completed unit can be seen in Figure 5.

The MorseMeter is a great workbench aid for visually impaired electronic technicians or just about anyone who needs an audible indication of a multimeter reading. Construction of fur-

ther examples of this design might be worthy community service projects for ham clubs or agencies devoted to support of the visually impaired. And, of course, it provides a great way to learn about PIC microcontrollers—I know it did it for me!

Notes

- ¹www.kitsrus.com/.
- ²www.kitsrus.com/upuc.html#k81.
- ³www.microchip.com/.
- ⁴www.kitsrus.com/zip/k81manual.zip.
- ⁵www.electronics123.com/.
- ⁶www.data-connect.com/shareware_communications_software.htm.
- ⁷www.metex.co.kr/.
- ⁸www.p-mastech.com/.
- ⁹www.melabs.com/.
- ¹⁰www.arrl.org/files/qst-binaries/morsemeter.zip.

Photos by the authors

Steve Johnston, WD8DAS, started taking apart radios as a youngster and became a ham at age 13, 29 years ago. A destroyer, builder, repairer and restorer of electronic gear for nearly all of his life, Steve's been a broadcast engineer for the past 22 years. He is Director of Engineering and Operations for Boise State Radio, a 20 station, network PBS system in the Northwest. Steve has a BA in history, and when he's not spending time with radio and electronics, he is a writer and a historian. He lives in Boise, Idaho with his wife Christy and their two children, Kaitlin and Noah. You can reach Steve at sbjohnston@aol.com.

Tim Hynde, KA8DDZ, was first licensed in 1978. He worked for Dentron Radio right out of high school; this led to a 20 year career in industrial automation involving motion control, computer-based machine controllers and automation interfacing. Currently employed as a Title Insurance Agent, Tim enjoys programming and has a strong interest in microcontrollers and their application to Amateur Radio. He can be reached at tim@marydudley.com. 