Datasheet for PAS12 Serial Adaptor rev 1.50

© 2001 Working Technologies Ltd.
shane@workingtex.com
fax: +64 3 3793885

http://www.workingtex.com/htpic/
ph. +64 21 2977741
ph. +64 3 3793883

Introduction

To communicate from a computer to a standard microcontroller, a serial adaptor is needed.

The adaptor shown below is the PAS12 adaptor. This serial adaptor converts the +/-13V of an RS232 serial port to the +5V and 0V that a microcontroller reads into its hardware serial port.

Most microcontrollers work at +5V / 0V logic levels. Therefore, this adaptor will enable serial communications for the Microchip, Atmel, Motorola, Scenix, Texas Instruments, Philips, 8051 series and Hitachi microcontrollers, to name a few.

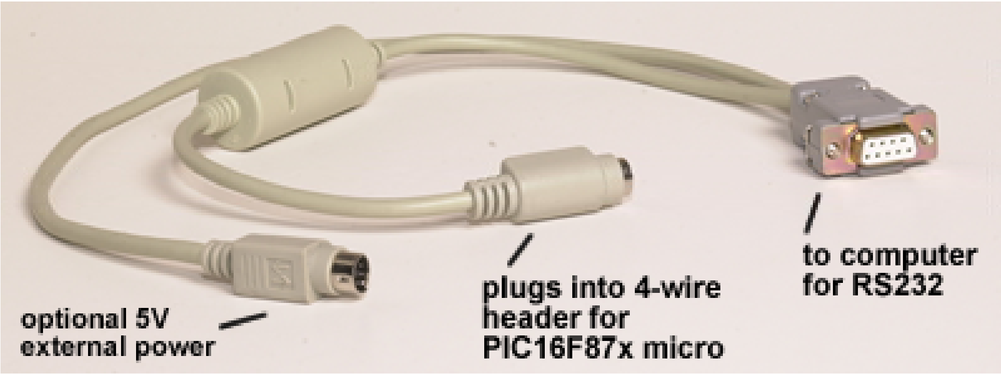
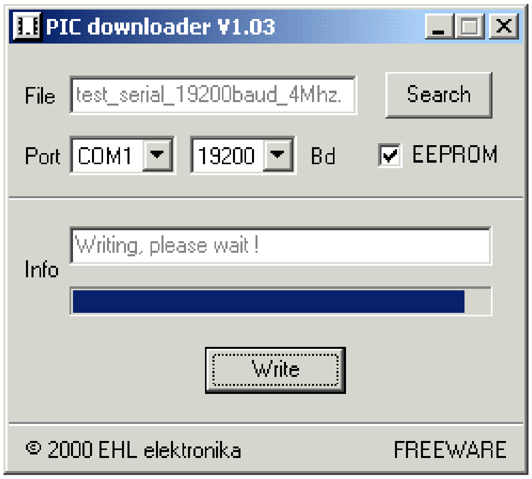


Figure 1: Product photo

Using the PAS12 adaptor in a PIC16F87x bootloader

Introduction

A bootloader is used to quickly download a new program into your PIC16F87x micro, in seconds.



File test_serial_19200baud_4Mhz.

Port COM1 19200 Bd ☑ EEPROM

Info Writing, please wait !

© 2000 EHL elektronika FREEWARE

Figure 2: PIC downloader screenshot

Bootloading is easily performed in-circuit, with the PIC micro still plugged into your PCB.

No modification to your circuit board is required.

Features

PIC16F87x serial bootloader. Currently in use by hundreds of users worldwide.

- 255 instructions long, fits into top of memory.
- Only uses two wires - TX and RX. This is unlike some bootloaders that need 5

wires!
- Example .hex files for 16F873/4/6/7.
- Example .hex files for 3.6864, 4,16 and 20Mhz.
- Downloads at 19200bps. This takes about 8 seconds for a complete program.
- Writes to EEPROM and ignores config bits.
- Circuit diagram supplied as a .gif or Protel 99 .sch file.
- Full source code supplied for bootloader, in assembly and C
- Compatible with any circuit - **no** trigger pins or jumpers needed to activate bootloader.
- Fits into space that the MPLab-ICD (an Incircuit Debugger) would normally use, if present.
- Extensively tested, with all combinations of crystal speeds, and 16F873/4/6/7.
- Windows interface plus full Delphi source code.

How to use the bootloader

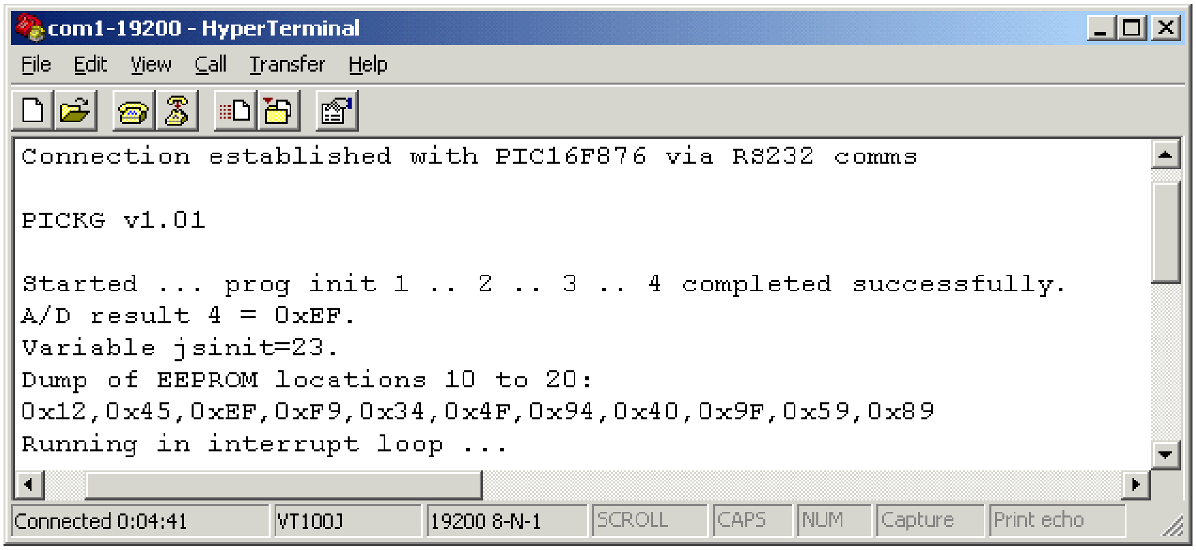
You start with your application circuit, complete with a 16F87x.

- Program the PIC micro in your target circuit with the bootloader code. This is a once-off operation.
- Plug in the PAS12 serial adaptor to your PC. 4 wires plug into the PIC micro.
- Add an extra line in your program to reserve the top 255 bytes in memory for the bootloader. Full Instructions provided.
- Reset your target circuit, and hit the *Write* button on the Windows software.
- Finished, less than 15 seconds for an 4k download

See http://www.workingtex.com/htpic for more, and to download the free bootloader.

Using the PAS12 adaptor for debugging

Once you have a PAS12 adaptor, a serial connection can be established between the PIC and a computer. This enables you to perform complicated debugging. The screenshot below illustrates this perfectly.



```
Connection established with PIC16F876 via RS232 comms

PICKG v1.01

Started ... prog init 1 .. 2 .. 3 .. 4 completed successfully.
A/D result 4 = 0xEF.
Variable jsinit=23.
Dump of EEPROM locations 10 to 20:
0x12,0x45,0xEF,0xF9,0x34,0x4F,0x94,0x40,0x9F,0x59,0x89
Running in interrupt loop ...
```

Figure 3: Debugging via serial port

An in-circuit debugger such as the MPLab-ICD can almost provide the same information. However, a serial link is superior in some ways. It provides a history of program events, unlike a debugger which can only provide the current state of the program. The best tools are a combination of an MPLab-ICD and a serial link.

To receive/transmit keystrokes in the terminal, simply include the following C statements in your program:

```
setup_serial(); //setup RX and TX for hardware serial
x=getch(); //waits for a character pressed in the terminal, eg: returns 'a' if keystroke 'a' typed

putch('a'); //prints 'a' on the terminal
putch(0x4A); //prints binary 0x4A to the terminal
putst("PICKG v1.01"); //prints strings to the terminal
putchhex(0x4A); //prints an ascii number to the terminal, namely the 2 ascii characters "4A"
putchdec(56789); //prints a decimal number to the terminal, the 5 ascii characters "56789"
```

See http://www.workingtex.com/htpic for the C libraries needed to transmit and receive serial characters. Assembly libraries for communicating to a computer via a serial link are widely available.

How to connect the PAS12 adaptor to your circuit

A standard PS/2 computer extension cable is shown in figure 3 below.

Figure 3: PS/2 cable for interfacing to target circuit board

As in figures 4 and 5 below, cut the cable about 20cm from one end, and bare the wires. Cut all unused wires off short.

Key:

1. Orange, +5 volts
2. Red, Ground
3. Female plug

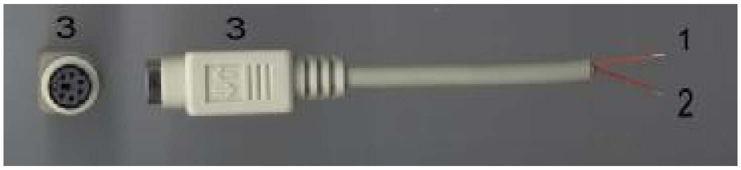


Figure 4: Female PS/2 plug, for external +5V power supply

Key:

1. Orange, to +5volts to PIC
2. Black, to TX on PIC
3. Yellow, to RX on PIC
4. Red, Ground to PIC
5. Male plug

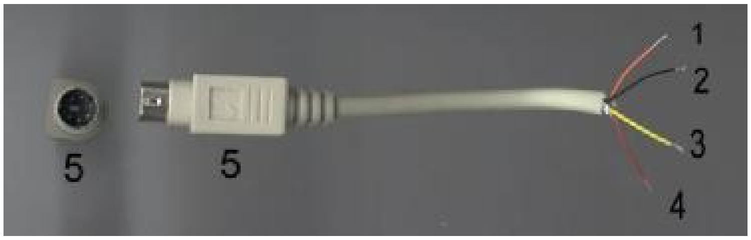


Figure 5: Male PS/2 plug pinouts, connects to target PIC

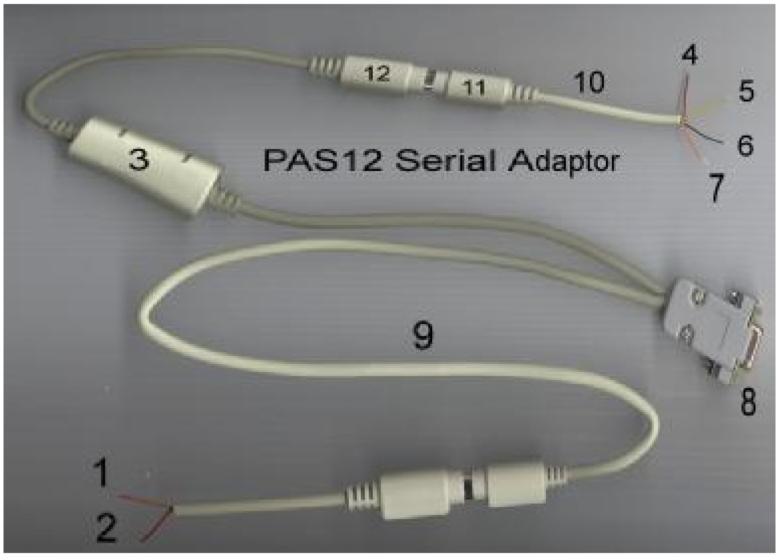


Figure 6: Connection Diagram

Plug the RS232 serial plug numbered **8** in figure 6 into a spare serial port on your computer. Choose between COM1, COM2, COM3 or COM4.

In your target circuit, locate the **+5V** pin on your PIC16F87x. Connect this to the **orange** wire numbered **7** in figure 6.

In your target circuit, locate the **ground** pin on your PIC16F87x. Connect this to the **red** wire numbered **4** in figure 6.

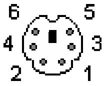
In your target circuit, locate the RC7/TX pin on your PIC16F87x. According to the data sheet, for 28-pin PIC16F873 / 876 devices RC7/TX is pin 18. For 40-pin PIC16F874 / 877 devices this is pin 26. Connect **RC7/TX** to the **black** wire numbered **6** in figure 6.

In your target circuit, locate the RC6/RX pin on your PIC16F87x. According to the data sheet, for 28-pin PIC16F873 / 876 devices RC6/RX is pin 17. For 40-pin PIC16F874 / 877 devices this is pin 25. Connect **RC6/RX** to the **yellow** wire numbered **5** in figure 6.

The optional external power pins numbered **1** and **2** are **unconnected** in this configuration, however, they can be used to supply external power to the circuit. If you do decide to use wire **9** in figure 6 as external power, the **red** wire numbered **1** in figure 6 is **ground**. The **orange** wire numbered **2** in figure 6 is **+5V**. As stated, these two wires are unconnected in this configuration, as power to the MAX232 level shifter chip is being supplied from the target circuit itself.

Pinouts

For the plug numbered 12 in figure 6 above, the pinouts are as follows:



PS/2 pinouts for plug 12 in figure 6 above	
Pin	Description
1	SERIAL_RX >> black >> RC6/TX/CK on 16F87x
2	unconnected >> brown
3	GND >> red >> GND on PIC16F87x
4	VCC (+5V) >> orange >> POWER on 16F87x
5	SERIAL_TX >> yellow >> RC7/RX/DT on 16F87x
6	unconnected >> green

The transmitted characters from the PC (SERIAL_TX) go to the receive pin on the PIC16F87x micro (RC6/RX/DT). This is because characters are transmitted from the PC into the receive pin of the micro. Likewise, the received characters into the PC (SERIAL_RX) come from the transmit pin on the PIC16F87x micro (RC7/RX/DT).

If you connect pins 5 and 1 in the diagram above, it forms a loopback. Every serial character transmitted is received back into the serial port. If you use a terminal program, with local echo turned on, every character pressed will come up twice on the screen.

Using wire 9 as a Power Supply

It is possible to use the wire marked **9** in figure 6 as a power supply. Simply plug it into the PS/2 keyboard or mouse port of a computer or laptop, this will supply power to the target circuit, from the computer. However, if there is a problem with the target circuit, and a short between +5V and 0V is introduced, this may cause problems. It is not recommended that this option is used, unless the target circuit is known to be working perfectly.

Fill in the pin numbers marked ? using a continuity meter.

- Orange, goes to +5volts to PIC, goes to pin **?** in PS/2 plug
- Black, goes to TX on PIC, goes to pin **?** in PS/2 plug
- Yellow, goes to RX on PIC, goes to pin **?** in PS/2 plug
- Red, goes to Ground to PIC, goes to pin **?** in PS/2 plug

Figure 7 - Finding pinouts of PS/2 plug

Most computers since 1996 have current-limiting polyswitches on their keyboard and mouse ports, which limit the current to 200mA. However, some older computers have no current limit, and a short between the +5V and 0V lines of the keyboard will damage the motherboard. This can be repaired by replacing the motherboard fuse.

Using a PS/2 male plug on your circuit board

It is possible to replace wire **10** in diagram 6 with a PS/2 plug, that is soldered directly to the target circuit board. Since the pinouts of some PS/2 plugs vary, use a multimeter to find out the equivalent wires to attach to your application circuit.

Plug a female PS/2 connector into the PS/2 plug on your circuit board. Look at figure 4, and identify the four colored wires, orange, black, yellow, and red. You may have to bare 2 additional wires, the yellow and black.

Now, using a multimeter in continuity mode, match the colors to the pins on the plug.

For example:

If the orange wire in the PS/2 cable (figure 4) and pin 4 on the PS/2 plug (on target circuit) are electrically shorted
then attach pin 4 on the PS/2 plug (on application circuit) to the +5V pin on the PIC micro (on application circuit).

Conclusion

We have seen that a PAS12 serial adaptor can make ones programming style a lot more powerful. A bootloader decreases development time, and can be used for remote software field updates without a programmer. All that is needed is a laptop computer and an adaptor.

In addition, a PAS12 serial adaptor enables debugging data to be sent to and from a computer.

Where to get a PAS12 serial adaptor

The PAS12 serial adaptor is available from Working Technologies. See the website http://www.workingtex.com/htpic/ for more information.

Copyright

We welcome your comments! If you have any enquiries at all, send them to shane@workingtex.com.

Standard Disclaimer

Working Technologes and/or employees thereof will not be held responsible for any loss due to use of this product.