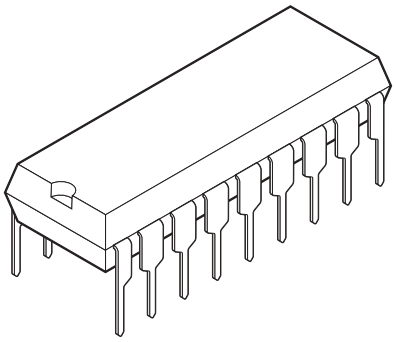
# Serial Printer Firmware

## Order Code

FRM015 Serial Printer Firmware Chip



## Serial Printer Firmware Contents

1 x PIC16C620A (programmed)
1 x data sheet

## Introduction

The serial printer firmware is designed to allow printing on most parallel interface printers via PICAXE microcontrollers and BASIC stamps. It allows such devices the ability to create hard copy output of data without the need for a PC.

Data is written to the printer firmware over a single serial data wire at 2400 Baud. The firmware generates the appropriate printer control signals, as well as monitors printer status to cause the incoming serial stream to be printed on the parallel printer. The firmware provides basic text output in the printers default font.

## Connections

It is important that the printer is switched off when the cables, modules etc. are connected together. If not it is possible for random characters to be printed due to noise spikes as the modules are connected. In practice we would recommend that the printer is only switched on directly before the PICAXE program is downloaded, and switched off again after the program has run.

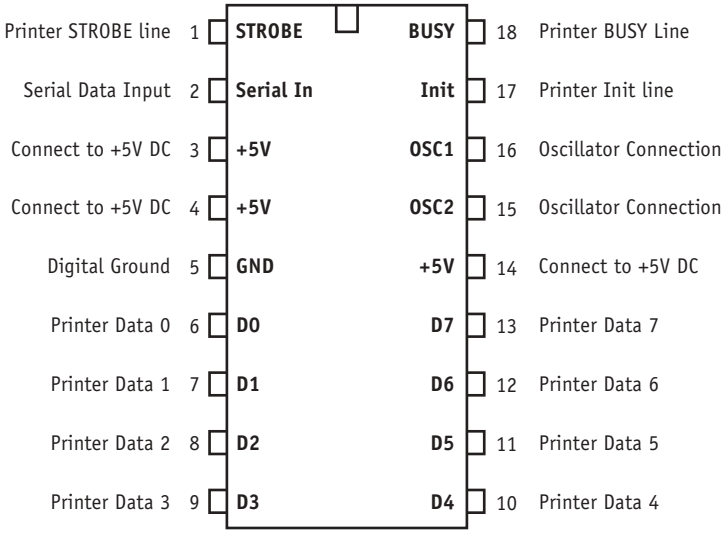
Note that on dot-matrix (and some ink-jet) printers each line is printed separately when the carriage return (CR-13) character is sent. On most inkjets and all laser printers the whole page is printed when the form feed (FF-12) character is sent

The printer firmware requires three connections to the PICAXE: V+ (5V), Ground (0V), and serial pin. Serial communication is often made via pin 7, and that is the pin used in all the example programs.
WARNING: Always switch the printer OFF when making connections!

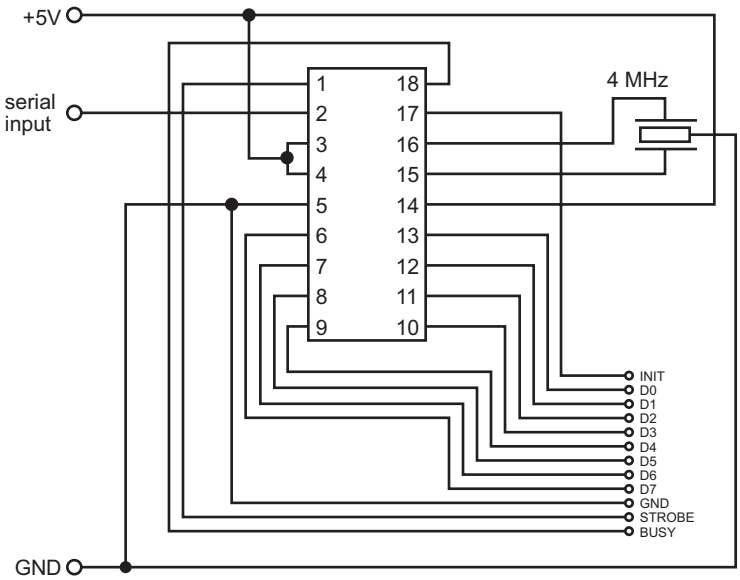
## Limitations

The printer firmware provides a basic text printing facility, using the default printer font (the font and font size cannot be altered). It does not support all the 'intelligent' features of modern printers (e.g. graphics). In particular the error-reporting is very basic, and so it is important to ensure the printer is correctly set-up (i.e. paper is present etc.) before trying to print.

## Firmware Pinout Diagram




## Connection to PICAXE




## Connection to Printer

| Signal Name | Firmware Pin | D25 Cable Pin | Connector on Printer |
|---|---|---|---|
| D0 | Pin 6 | Pin 2 | Pin 2 |
| D1 | Pin 7 | Pin 3 | Pin 3 |
| D2 | Pin 8 | Pin 4 | Pin 4 |
| D3 | Pin 9 | Pin 5 | Pin 5 |
| D4 | Pin 10 | Pin 6 | Pin 6 |
| D5 | Pin 11 | Pin 7 | Pin 7 |
| D6 | Pin 12 | Pin 8 | Pin 8 |
| D7 | Pin 13 | Pin 9 | Pin 9 |
| STROBE | Pin 1 | Pin 1 | Pin 1 |
| BUSY | Pin 18 | Pin 11 | Pin 11 |
| INIT | Pin 17 | Pin 16 | Pin 16 |
| GND | Pin 5 | Pin 18-25 | Pin 19-30,33 |

## Operation:

The Serial Printer Firmware must be 'woken-up' with a start of text (STX-2) command before printing, and should be disabled with an end of text (ETX-3) command after printing is over. All lines must be followed by a carriage return (CR-13) and line feed (LF-10) sequence. All pages must be terminated with a form feed (FF- 12) character. The following PICAXE program illustrates how printing is used with the **serout** command.

```
'PROGRAM: PRT001.BAS

'PIACXE program that demonstrates
'how to use the serial Printer Module.

'Connect the Printer Module to the PICAXE.
'Switch the printer OFF, connect the printer
'cable to the module. Power up the PICAXE,
'and then switch the printer ON. Then download
'this program.

symbol      S_OUT =     7     'serial Printer output (output)
symbol      i     =     b5    'scratchpad counter

'Define useful ASCII characters
symbol      STX   =     2     'Start of text ("wake-up")
symbol      ETX   =     3     'End of text ("sleep")
symbol      TAB   =     9     'Tab
symbol      FF    =     12    'Form Feed

'Note that CR and LF are pre-defined constants
'within the Prog. Editor and therefore do not require
'a separate 'symbol' definition.

init: 'Make serial pin an output, and leave high for 5ms
      'This provides time for the printer module serial
      'input to reset.

      high S_OUT              'Make pin7 high output.
      pause 5                 'Short pause.

main: 'Wake up printer module via sending STX command.
      serout S_OUT,T2400,(STX)

      'Print 'Hello' message.
      serout S_OUT,T2400,("Hello",CR,LF)

      'Note all lines must always be followed by CR then LF.

      'Print ten more lines.
      for i = 1 to 10
         serout S_OUT,T2400,(TAB,"This is line ",#i,CR,LF)
      next i

      'Note the # before the variable i. This ensures
      'variable numbers are transmitted as a string of
      'ASCII characters rather than as a single byte.

      'Note also some printers print very near left edge.
      'Printing a TAB before each line helps create a margin.

      'Now finish printing by sending form-feed character.
      serout S_OUT,T2400,(FF)

      'Finally send printer driver module to sleep.
      serout S_OUT,T2400,(ETX)

      end
```