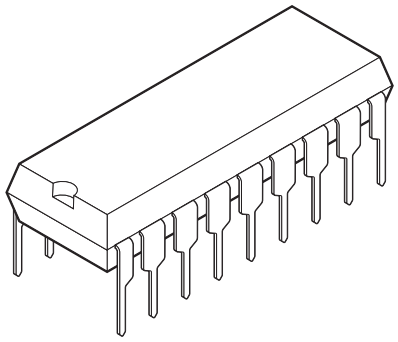
SERIAL LCD FIRMWARE

Order Code

FRM010 Serial LCD Firmware Chip



Serial Printer Firmware Contents

1 x PIC16C620A (programmed)
1 x 4MHz ceramic resonator
1 x data sheet

Introduction

The serial LCD firmware is used to allow serial control of an alphanumeric LCD. This allows microcontrollers (and microcontroller based systems such as the PICAXE or Basic Stamp) to visually output user instructions or readings onto a text screen without the need for a host computer. This is especially useful when working, for example, with analogue sensors, as the analogue reading can easily be displayed on the LCD module. All LCD commands are transmitted serially via a single microcontroller pin. A sample instruction, using the serout command is as follows:

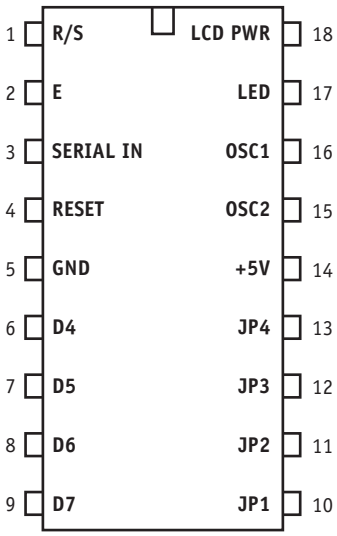
to print the text 'Hello' the instruction is simply

```
serout 7,T2400,("Hello")
```

The firmware can also be connected to the serial port of a computer using a 22k resistor.

Pin Definitions

Firmware Pin	LCD Pin	Description
Pin 1 - LCD RS	Pin 4	LCD Register Select Line
Pin 2 - LCD E	Pin 6	LCD Enable Line
Pin 3 - Serial Input	-	Serial Input Line
Pin 4 - Reset	-	Reset Pin - tie high with 4k7 resistor
Pin 5 - Ground	Pins 1, 5, 7-10	Ground
Pin 6 - LCD Data 4	Pin 11	LCD Data Line 4
Pin 7 - LCD Data 5	Pin 12	LCD Data Line 5
Pin 8 - LCD Data 6	Pin 13	LCD Data Line 6
Pin 9 - LCD Data 7	Pin 14	LCD Data Line 7
Pin 10 - Jumper 1	-	'Welcome message' jumper
Pin 11 - Jumper 2	-	'Baud rate' jumper
Pin 12 - Jumper 3	-	'Polarity' jumper
Pin 13 - Jumper 4	-	'Number of lines' jumper
Pin 14 - +5V Power	-	Power - connect to +5V
Pin 15 - Osc2	-	Clock - connect to 4MHz resonator
Pin 16 - Osc1	-	Clock - connect to 4MHz resonator
Pin 17 - LED	-	LED - connect to optional indicator LED
Pin 18 - LCD Output PWR	Pin 2	LCD Power - NB DO NOT Connect to +5V
-	Pin 3	LCD Contrast Line



Constructing the Full LCD Circuit

Parts List

1 Alphanumeric LCD Module (16x2)
1 10k preset potentiometer
1 1k resistor (22k for PC connection)
1 10k resistor
1 4k7 resistor
1 3-pin 4MHz ceramic resonator

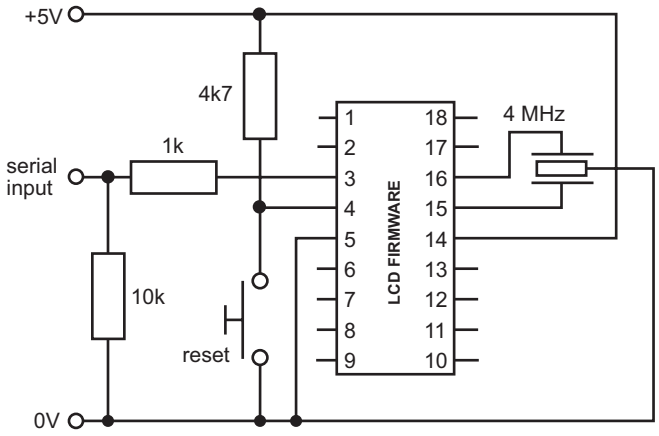
Optional:
1 680R resistor
1 LED
1 reset switch (push to make)

Construction

For ease of understanding the circuit diagram is broken down into two sections.

a) Firmware Connections
b) LCD Module Connections

A) FIRMWARE CONNECTIONS



a) Power and Oscillator

The circuit requires a +5V supply, which should be connected to pin 14 (+5V) and pin 5 (0V) of the firmware chip. The 4MHz 3-pin ceramic resonator should be connected to pins 15 and 16 as shown (the centre pin of the ceramic resonator is connected to ground).

b) Reset Switch

The optional reset switch is connected to firmware chip pin 4 as shown in the diagram. Note that the 4k7 resistor is ESSENTIAL and must be used even if the reset switch is omitted.

c) Serial Connection

PICAXE/Basic Stamp - The output pin of the PICAXE should be connected to the firmware chip pin 3 via a 1k resistor as shown in the diagram. The 10k pull down resistor is not required. DO NOT CONNECT VIA THE DARLINGTON DRIVER output on PICAXE boards - connect directly to the microcontroller output pin.

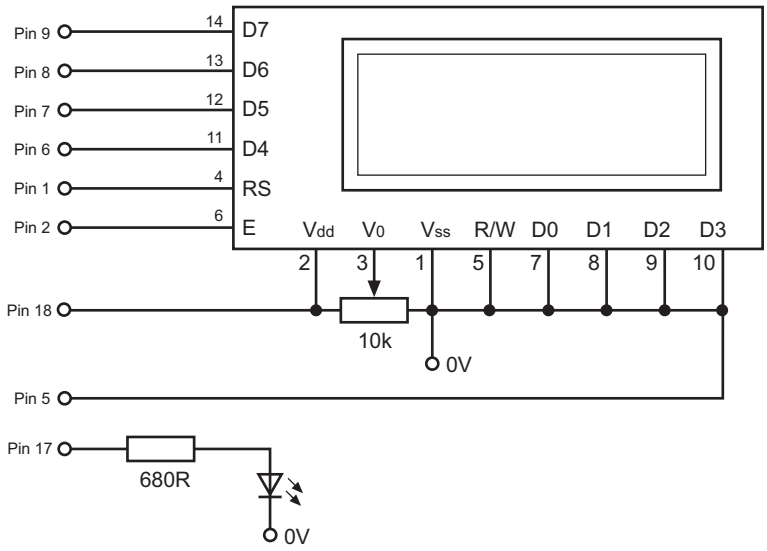
Serial Port of a PC - The TXD pin (pin 3 of the 9 way D serial connector) should be connected to the firmware chip pin 3 via a 22k resistor (ie replace the IK resistor shown in the diagram above with a 22k resistor). The GND pin (pin 5 of the 9 way D serial connector) should also be connected to the firmware chip pin 5.
IMPORTANT NOTE: The 22k resistor must be used when connecting directly to the serial port of a PC. If it is omitted damage may occur to the firmware chip and/or the computer serial port due to excessive current flow.

d) Jumper Settings

The serial LCD firmware chip has four optional jumpers (pins 10 to 13). The default setting for each of these jumpers is 'open', and so in most cases the pin can be left unconnected (there is an internal pull-up resistor within the chip itself). If you wish to 'close' the jumper this should be achieved by connected the pin to GND (0V)

Jumper 1	When closed the power-up 'welcome' message is disabled.
Jumper 2	When closed the baud rate is set to 9600 (open value is 2400)
Jumper 3	When closed the serial protocol is inverted, for direct connection to a PC.
Jumper 4	When closed the LCD module is configured to use just one line instead of two

B) LCD MODULE CONNECTIONS



e) LCD

The LCD is connected to the firmware chip as shown in the diagram above.
Note that the +5V supply is NOT directly connected to the LCD module as the LCD is powered via pin 18 of the firmware chip. Note also the use of the 10k preset potentiometer. This is used to alter the contrast of the LCD module.

f) LED

The optional LED lights when serial transmission is in progress. It is connected to firmware chip pin 17 via a 680 ohm resistor as shown above.

Using the LCD Instruction set

The codes for the LCD instruction set are given below. These commands are used to move around the display, start new lines and enable/disable the cursor. Each code can be sent to the LCD module by sending the number 254 followed by the command below. These instructions can be used to make the LCD messages more interesting - for instance by flashing the screen or creating 'moving' messages which scroll across the screen.

Code	Instruction
1	Clear display and move to the start of the first line
2	Move the cursor and display 'window' to the start of the first line
4	Set 'right to left printing' mode
5	Set 'scroll printing to the left' mode
6	Set 'left to right printing' mode
7	Set 'scroll printing to the right' mode
10	Turn visual LCD screen off
12	Hide cursor
13	Make cursor flash
14	Turn visual LCD screen (and cursor) on
16	Move cursor left one position
20	Move cursor right one position
24	Scroll display 'window' left one position
28	Scroll display 'window' right one position
128	Move cursor to the start of the first line
192	Move cursor to the start of the second line

A sample program using two lines of the LCD is as follows:

```
serout 7,T2400,(254,1)                    ' blank the screen
pause 30                                  ' short delay to enable blank to complete
serout 7,T2400,(254,128,"Top Line")       ' top line message
serout 7,T2400,(254,192,"Bottom Line")    ' bottom line message
pause 2000

loop:                                     ' start a loop
let b1 = b1 + 1                           ' increment variable
pause 500                                 ' short delay
serout 7,T2400,(254,192,"Count = ", #b1)  ' output value on bottom line
goto loop
```