**I2C Interface for QL Emulators Part 7**
(and now for non Minerva MK2 Black Box QL's)
Originally Published in QL Today Vol 17, Issue 3, March-May 2013

Some of you may have been put off by the cost of the By-Vac BV4221-V2 since, at the time of writing, this device costs £24.50. Another down side of this device is it will only work with PC based emulators since it uses the USB port. So is there another way ? Yes there is, it is both cheaper, well the main hardware solution anyway, and will work with 'Black Box' QL's. However you will have to construct it, program a PIC microprocessor and provide a 5VDC regulated supply. So there is more work involved. But this is a really good way of using that old QL you have hidden away. I am using one as a model railway layout controller. Without delving (at least too far) into the 'Black Box' itself. As to the BT connectors used on standard UK QL's please see my comments in QLT 17, Issue 2, page 9.

The cost of the main active components is as follows:-

| | | |
|---|---|---|
| From Farnell | | |
| PIC12F675I/P | 9759018 | £1.08 |
| MAX232CPE+ | 9725172 | £3.67 |
| From Rapid | | |
| PIC12F675I/P | 73-3284 | £0.84 |
| MAX232CPE+ | 82-0148 | £1.41 |
| 10uF 25V Cap x 4 | 11-3686` | £0.031 each |
| 0.1uF Cap x 2 | 08-0235 | £0.021 each |

So depending where you purchase the components, and adding things like connectors, this converter can be built for around £10. Prices correct at time of writing, also there may be minimum order requirements, so do check carefully when ordering parts.

However you do need the ability to program PIC chips. There are several ways of doing this. For example the Microchip PICkit2 made by the manufactures of the PIC micro's themselves, costs £32.26 from Farnell. An alternative is the Velleman kit (K8048) at £34.99 from Maplin. The Velleman kit is limited in the range of PIC devices it can program. However it will program the PIC12F675 as used in this converter and also the PIC16F84 as used in the PS2 mouse to game port converter by flyer.gio from Italy see my review in QLT 17, Issue 2.. The Velleman kit is very good if you want to learn more about PIC's, since it is a programmer and development board, so you takes your choice. I, in fact have both these devices and they work very well. But this does add to the overall cost, but is a good investment, for other PIC based projects and if you wish learn more about PIC devices.

There is a third (the cheapest) solution called WIN PIC. This is a PC based programmer, like the ones above, but with fairly simple hardware which you can make on some strip board. WIN PIC will program a wide variety of PIC devices, including the PIC16F84A for the PS Mouse project and the PIC12F675 for this project. Do use the circuit diagram at http://www.qsl.net/dl4yhf/winpic/index.htm#eprom_prog. Since this has a better arrangement for the all important programming voltage. The power for the programming voltage can be provided with two (R22) PP3 batteries connected in

series to give a voltage of about 18 volts, the on board regulator then reduces this to the required 12.7 volts for programming the PIC's. All the PC software and information for this programmer can be found at http://www.qsl.net/dl4yhf/winpicpr.html, then scroll down the page until the WIN PIC manual for the user manual. You will find the software download further down the opening page.

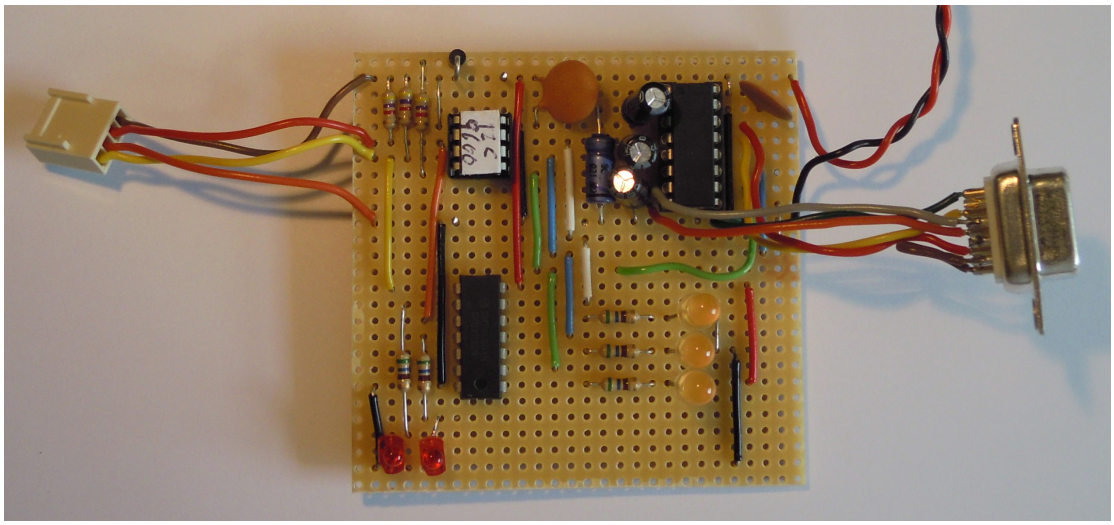
I have tested the WIN PIC programmer on both the 'PS Mouse' and the 'RS232 to I2C' projects and it works with no problem.

You will also need Microchips MPLAB IDE development tool, this is free to download from Microchips web site, please see below for link. Be warned this is a big download.

In essence this PIC based converter is a RS232 to I2C converter. To use it with a PC based emulators such as QPC you will need either a RS232 on your PC or a USB to RS232 converter. These are available from such places as Maplin's. You may already have one of these. So again this may add to the cost. So the ByVac product could still be the best answer for you, if you don't have a USB to RS232 converter or PIC programmer.

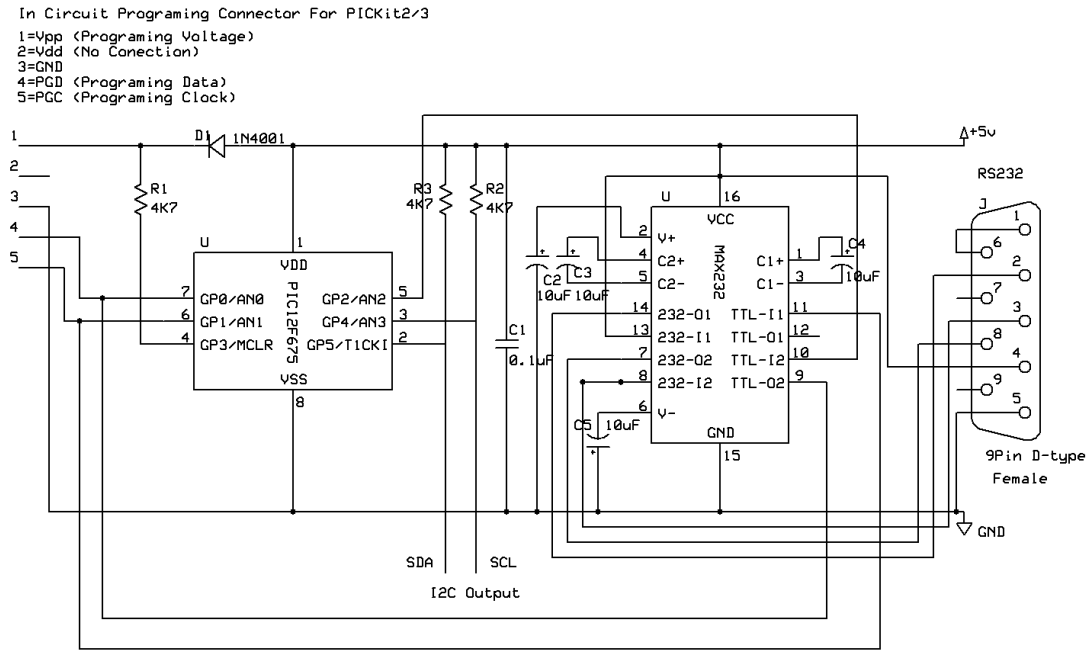
So to the RS232 to I2C converter itself. The hardware and firmware for this project was developed by Andrew M Bishop. He, as far as I know is not a QL user. This project was originally designed for Linux and PC based projects. See his web site for more information. You will also find other RS232 based converters as well, such as for PS2, SPI and Infra Red device. Which you may find of interest, but I will only be dealing with the I2C this time.

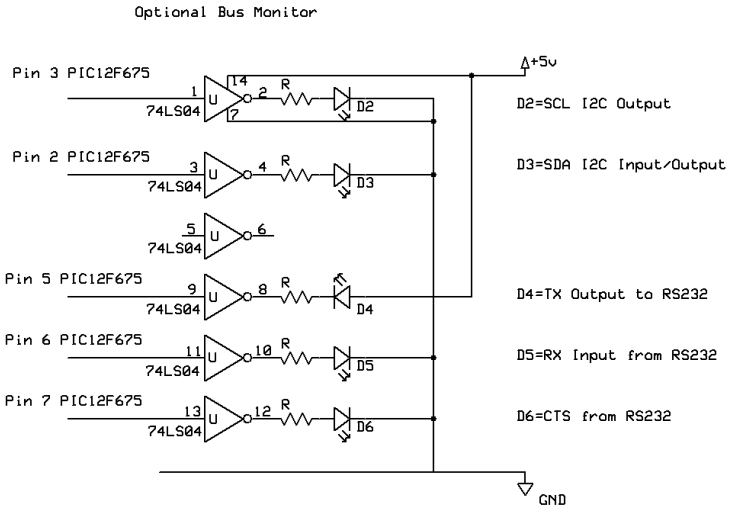
The nice thing about Andrew's projects is that all the source code as well at the compiled hex files are available in his amb-pic-code file. See below for the link for this. So you can see how these things work and make any changes you want for your own purposes.



In its original form the converter works at 38400 baud, which is fine for emulator users like QPC2, but is far to fast for ‘Black Box’ QL users. So I have modified Andrews code to work at 9600 baud. I will explain how you change Andrews code.

The circuit diagram below shows the complete project, it breaks down into three parts. The PIC which does all the clever stuff and does the conversion work. This is the top left integrated circuit with just the PIC and three resistors and a diode. For PICKit2/3 users I have shown the connections for the programmer. Second is the MAX232 which is a level converter to change the +5v signals to and from the PIC to the +/- 8 volts required for the RS232 interface, this is the integrated circuit and associated components, top right. The third and final part is a LED monitor, which shows the traffic on all the I2C and RS232 lines, this is optional. I just put this in for my own testing purposes. For this project to work you just need the PIC and MAX232 parts of the circuit in the top diagram below. Unlike the BV4221 which received it’s power from the USB connection, you do need to provide a regulated 5VDC supply to this circuit.




As you can see it is very simple, with a low component count. There's is not even any crystals, since the PIC is programmed to use an internal oscillator.

Using the original hex file 'program.hex' from Andrew's library, which you will find in the amb-pic-code-2010-09-19/rs232-i2c directory. Then just follow the instruction for your PIC programmer. This will give a working PIC converter that works at 38400 baud.

This high baud rate is fine if you just want to use this converter with QPC2. However this is far to fast for a 'Black Box QL'. The maximum speed with a standard QL is 9600 baud. Now to change the baud rate from 38400 to 9600 you will need the MPLAB IDE. So download this from the Microchip web site and install it on your PC. No choice here has to be a PC.

Now to make things a little easier to understand I put all the required files to compile the PIC code in one directory. They are listed below you will find them in the amb-pic-code-2010-09-19/rs232-i2c directory and the amb-pic-code-2010-09-19/common directories.

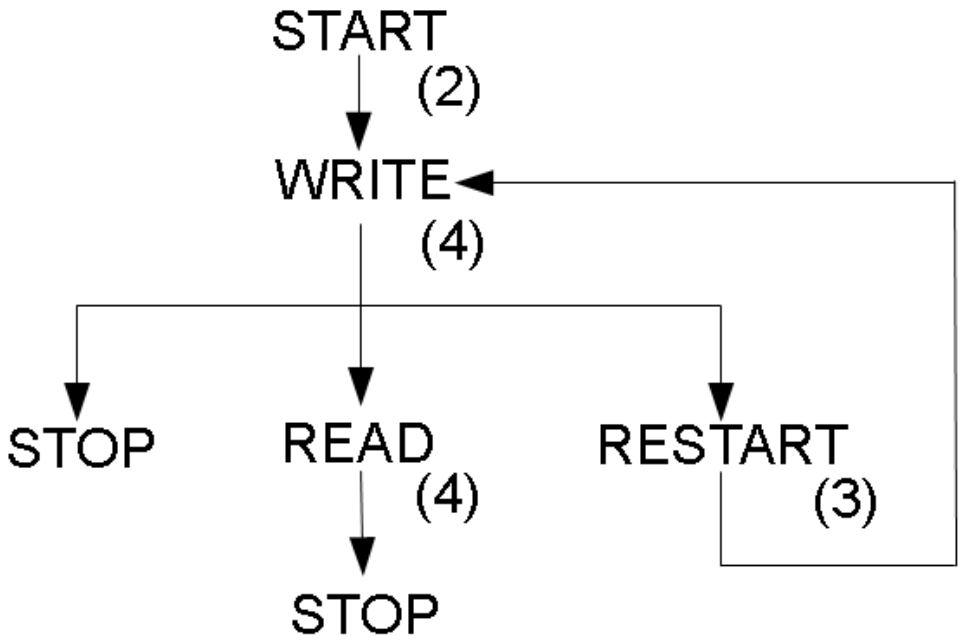
Delay.inc
Delay_cycle.inc
Delay_ms.inc
Delay_us.inc
Eeprom.inc
Generic.inc
I2c.inc
I2c_basic.inc
I2c_master.inc
I2c_slave.inc
I2c_ssp.inc
I2c_ssp_basic.inc
I2c_ssp_master.inc
I2x_ssp_slave.inc
Reset_device.inc
Rs232.inc
Rs232_basic.inc
Rs232_rx.inc
Rs232_time.inc
Rs232_tx.inc
Program.asm

It is best to load the elements above into MPLAB IDE to edit and then save them. You could do this in Notepad, but the layout will not be very clear. Now just go though all the above RS232 files and change any entry from 38400 to 9600. Recompile the code. If you do not want to do this then I have made available the hex file, which runs at 9600 to the editors of QLToday.

So that is the hardware now the software. The protocol for this converter is different from the BV4221 we have discussed during this series.

The commands are as follows:-
START = ‘S’
STOP = ‘P’
RESET = ‘R’
WRITE COMMAND = ’w’
WRITE DATA = ’x’, For use with EEPROMS
READ COMMAND = ’r’
READ DATA = ’s’, For use with EEPROMS

State machine that is implemented here.
(Brackets show error number if expected state is NOT reached)




State Machine

So the order of commands should look something like this :-

'S'TART, write *n* bytes, STO'P'
or
'S'TART, write *n* bytes, 'R'ESTART, read *m* bytes, STO'P'

A simple example SB program is show below, which is fully commented, this can drive a PCF8574A in output mode :-

```
10 REMark RS232 to I2C test
20 CLS
30 BAUD ser1,9600:REMark Sets the baud rate for the RS232 to I2C Converter, either 9600 or 38400, depending on how the PIC has been programmed
40 OPEN#3;ser1:REMark opens the required serial port.
50 PRINT#3;CHR$(83);:REMark HEX='53',ASCII code='S' for START I2C command
60 PAUSE 10:REMark Delay to allow the PIC to process the data and be ready for the next character
70 PRINT#3;CHR$(119);:REMark HEX='77', ASCII code="w" write to I2C device
80 PAUSE 10
90 PRINT#3;CHR$(2);:REMark HEX='02' number of bytes to follow.
100 PAUSE 10
110 PRINT#3;CHR$(126);:REMark HEX='7E', ASCII='~' the I2C device address.
120 PAUSE 10
130 PRINT#3;CHR$(129);:REMark DATA to be sent to the I2C device (0 to 255).
140 PAUSE 10
150 PRINT#3;CHR$(80);:REMark HEX='50', ASCII='P' for the STOP I2C command
160 PAUSE 10
```

```
170 FOR a=0 TO 2: REMark This FOR NEXT loop reads the return which is sent
after the STOP command has been sent to the RS232 to I2C converter.The return
from RS232 to I2C converter should be 'OK' if all is well. This also where the errors
will be returned if this go wrong, see State Machine above to help with debugging.
180 a$=INKEY$(#3)
190 IF a$="" THEN GO TO 180
200 PRINT a$;
210 NEXT a
220 CLOSE#3
```

I will leave it you to try reading data back from a device and amending my previous programs as you require.

I would like to thank Andrew M Bishop's help in the preparation of this article and his permission to feature his PIC development.

References

Andrew M Bishop's RS232 Converter page and software libraries
http://www.gedanken.org.uk/electronics/rs232-converters/i2c.html
http://www.gedanken.org.uk/electronics/amb-pic-code/

Microchip MPLAB IDE download site
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469

WIN PIC Programmer
http://www.qsl.net/dl4yhf/winpic/index.htm#eprom_prog
http://www.qsl.net/dl4yhf/winpicpr.html

DIY PS2 to game port converter by flyer.gio.
http://www.qlforum.co.uk/viewtopic.php?f=2&t=379
Software download for DIY PS2
http://www.mediafire.com/?xgmf9qx20sjxsbj