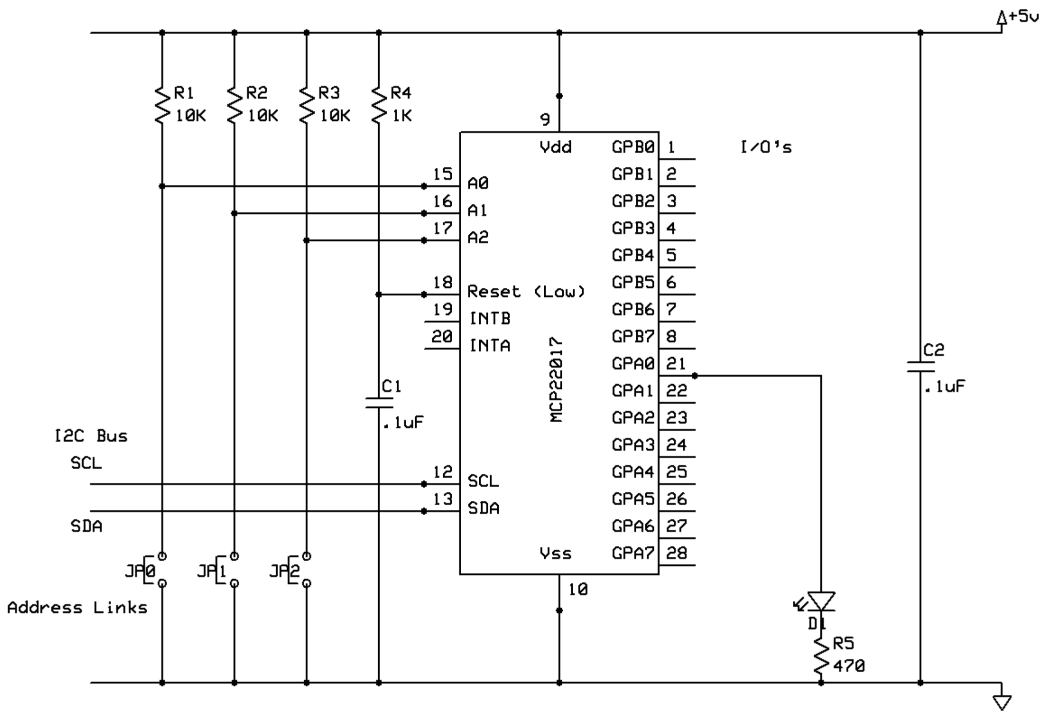
I2C Interface for QL Emulators Part 6

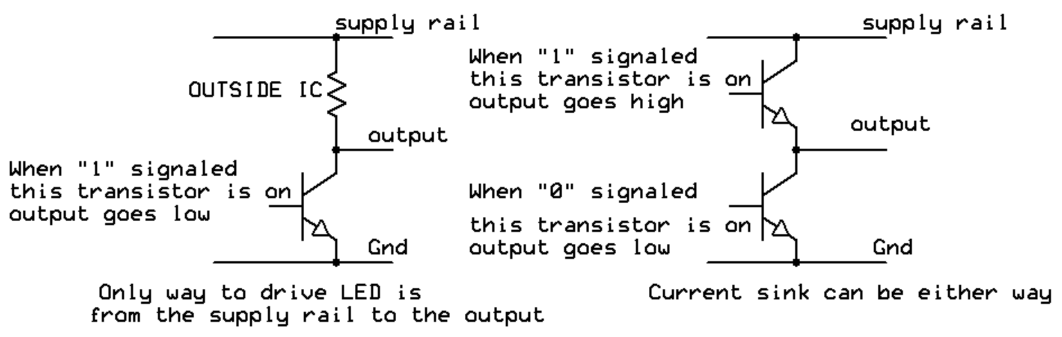
Originally Published in QL Today Vol 17, Issue 2, Dec-Feb 2012/2013

I recently purchased some PCF8574A's and found the price was now fairly high at £4.68 plus VAT each from Farnell and £1.63 plus VAT at RS Components. So I started to look to see if there were any other devices about. I came across a device from Microchip of PIC fame. Called MCP23017 (Farnell part number 1332088) at 95p plus VAT or from RS Components (part number 403-806) for 78p plus VAT. Please beware there may be a minimum order value from these companies, so worth checking out other suppliers. It is even better value than that since it has 2 * 8 bit ports, so 16 GPI (general purpose interface) lines, which otherwise would need 2 of the PCF devices, so a major saving either way.



As you can see the circuit is straight forward, the main difference other than the 16 I/O's is the reset pin (18) this must be held high. As you may have seen in the address table in the part 5 of this series, the MCP23017 has the same address range as the PCF8574. So 8 of these device can be used on each I2C bus giving a maximum of 128 I/O lines. Or another way, the same as using 8 * PCF8574 and 8 * PCF8574A. A lot less wiring to do.

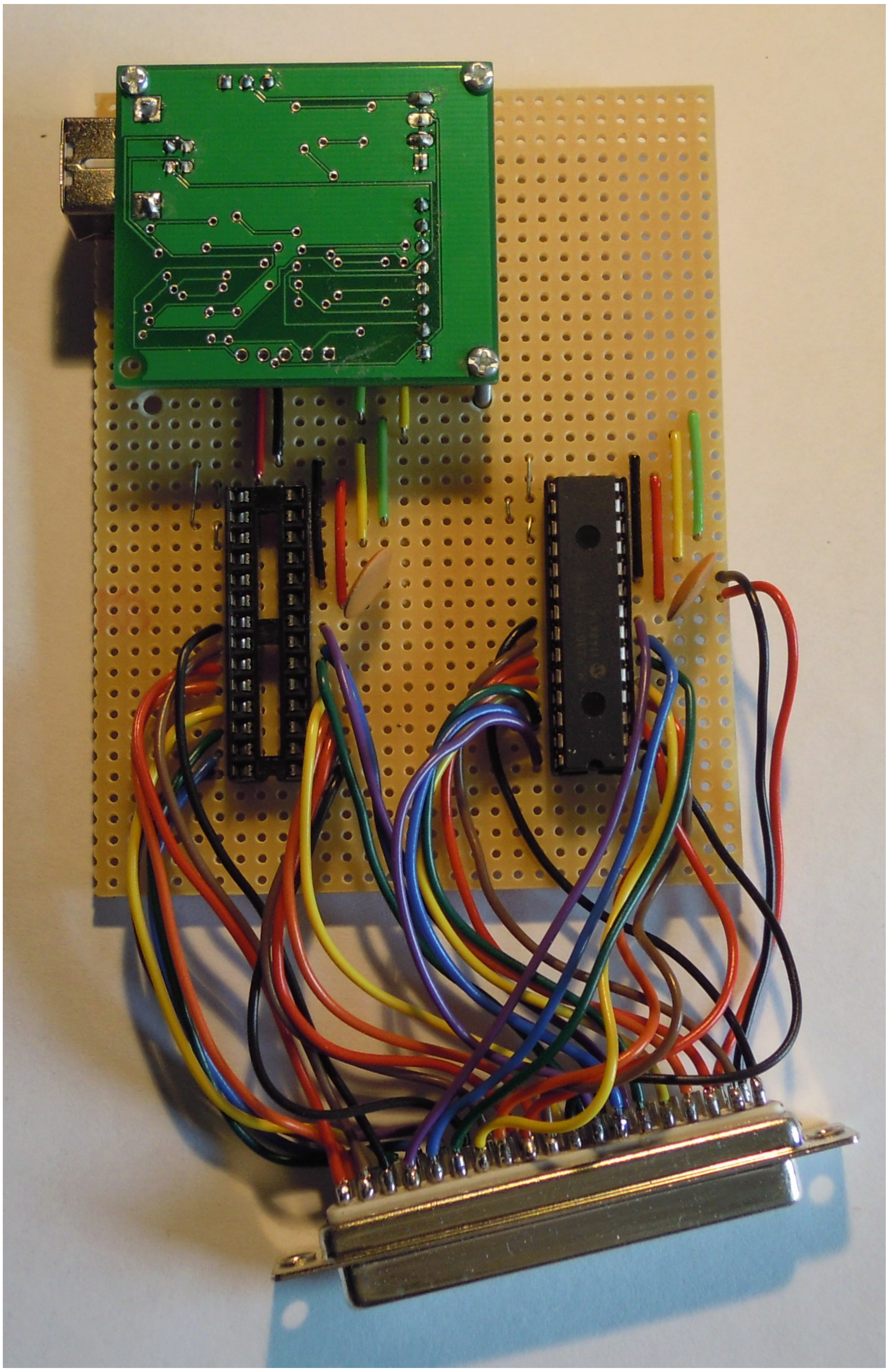
One other major hardware difference is the way the outputs works. With both the PCF8574 and PCF8574A, when in output mode, the output is what is called sink only. That is, there is in effect a transistor between the output pin and the GND, with a weak resistor from the output pin to the 5V supply rail. Which is why when a given pin is given a 'high' it is in fact low. Or put another way, it is inverted. One of the reasons, I put a driver chip ULN2803 to drive the LED's on my test card, see part 1 of this series. So the LED's display in the correct sense. In the case of the MCP23017, it has a more normal TTL type output. Sometimes referred to as a totem pole output.



PCF8574(A) Output arrangement MCP23017 Output arrangement

This is diagrammatic, this is not how it is really done within the chip, This is just for explanation.

Here there are in effect two transistors, one from the output pin to GND, as in the PCF example, but the resistor in the PCF is now replaced with another transistor. One transistor is turned on at a time. So the output can sink from supply to GND or to +5. So you can connect LED's for example from the output pin (LED anode), to GND (LED cathode) via a current limiting resistor. So no extra chip to invert the sense. So even more cost saving. However do not draw more than 20mA from any pin. If you need more current then you will still need something like the ULN2803 chip or drive power transistors. Still I will leave this to you since, this will all depend on your application. Also the BV4221 can only provide up to a maximum of 100mA. If your application requires more current, then you will have to provide an external power supply.



The picture above is my test circuit with two MCP23017 IC sockets with a BV4221-V2.

Now the software side of things.

See the I2C Interface article listing in QLT Vol 17 issue 1 page 40. You need lines 1000 to 3090 from this listing. Then merge it with the following listings.

```
10 REMark I2C test routines
20 I2C_init
30 I2C_Start
40 :
50 MCP_init
60 PRINT
70 PRINT "LED Flash (Port A)"
80 ledflash parallel8$,"A"
90 PRINT
100 PRINT "LED Flash (Port B)"
110 ledflash parallel8$,"B"
120 PRINT
130 PRINT "LED Flash (Port A)"
140 ledflash parallel7$,"A"
150 PRINT
160 PRINT "LED Flash (Port B)"
170 ledflash parallel7$,"B"
180 PRINT
190 PRINT "All LED Flash"
200 allledflash
210 PRINT
220 PRINT "LED Binary Count (Port A)"
230 ledcount parallel8$,"A"
240 PRINT
250 PRINT "LED Binary Count (Port B)"
260 ledcount parallel8$,"B"
270 PRINT
280 PRINT "LED Binary Count (Port A)"
290 ledcount parallel7$,"A"
300 PRINT
310 PRINT "LED Binary Count (Port B)"
320 ledcount parallel7$,"B"
330 PRINT
332 PRINT "LED Chase"
335 ledchase
340 :
350 PRINT "End        ":CLOSE#3:STOP
360 :
500 DEFine PROCedure monitor
510 c$=""
520 REPeat test
530 a$=INKEY$(#3)
540 IF a$="" THEN GO TO 530
550 c$=c$&a$
560 END REPeat test
```

```
570 END DEFine monitor
580 :
```

Lines 1000 to 3090 from QLT 17

```
4000 DEFine PROCedure ledflash(dadd$,port$)
4010 FOR a=1 TO 10
4020 IF port$="A" THEN PRINT#3;"s-";dadd$;" 12 ff p";CHR$(13);:REMark
s=start message to USB to I2C converter, p=end of message to USB to I2C converter.
4030 IF port$="B" THEN PRINT#3;"s-";dadd$;" 13 ff p";CHR$(13);:REMark s=start
message to USB to I2C converter, p=end of message to USB to I2C converter.
4040 non_print_reply:REMark Stops printing the USB to I2C reply also ensure serial
buffer is cleared.
4050 PAUSE 25
4060 IF port$="A" THEN PRINT#3;"s-";dadd$;" 12 00 p";CHR$(13);
4070 IF port$="B" THEN PRINT#3;"s-";dadd$;" 13 00 p";CHR$(13);
4080 non_print_reply:REMark Stops printing the USB to I2C reply also ensure serial
buffer is cleared.
4090 PAUSE 25
4100 NEXT a
4110 END DEFine ledflash
4120 :
5000 DEFine PROCedure ledcount(dadd$,port$)
5010 FOR a=0 TO 255
5020 hhex$=HEX$(a,8)
5030 PRINT a;" ";hhex$;" ";
5040 lhex$=hex_case_con(hhex$)
5050 IF port$="A" THEN PRINT#3;"s-";dadd$;" 12 "&lhex$&" p";CHR$(13);
5055 IF port$="B" THEN PRINT#3;"s-";dadd$;" 13 "&lhex$&" p";CHR$(13);
5060 non_print_reply:REMark Stops printing the USB to I2C reply also ensure serial
buffer is cleared.
5070 PAUSE 5
5080 NEXT a
5090 END DEFine ledcount
5100 :
6000 DEFine PROCedure MCP_init
6010 PRINT#3;"s-";parallel8$;" 00 00 p";CHR$(13);:REMark set mcp23017 port "A"
mode IODIRA to output
6020 non_print_reply
6030 PRINT#3;"s-";parallel8$;" 01 00 p";CHR$(13);:REMark set mcp23017 port "B"
mode B to output
6040 non_print_reply
6050 PRINT#3;"s-";parallel7$;" 00 00 p";CHR$(13);:REMark set mcp23017 port "A"
mode IODIRA to output
6060 non_print_reply

6070 PRINT#3;"s-";parallel7$;" 01 00 p";CHR$(13);:REMark set mcp23017 port "B"
mode B to output
6080 non_print_reply
6090 END DEFine MCP_init
```

```
6100 :
6110 DEFine PROCedure allledflash
6120 PRINT#3;"A40";CHR$(13);
6130 non_print_reply
6140 FOR a=1 TO 10
6150 PRINT#3;"s-";parallel8$;" 12 ff p s-";parallel8$;" 13 ff p s-";parallel7$;" 12 ff p
s-";parallel7$;" 13 ff p";CHR$(13);:REMark s=start message to USB to I2C
converter, p=end of message to USB to I2C converter.
6160 REMark PRINT#3;"s 13 ff p";CHR$(13);:REMark s=start message to USB to
I2C converter, p=end of message to USB to I2C converter.
6170 non_print_reply:REMark Stops printing the USB to I2C reply also ensure serial
buffer is cleared.
6180 PAUSE 25
6190 PRINT#3;"s-";parallel8$;" 12 00 p s-";parallel8$;" 13 00 p s-";parallel7$;" 12 00
p s-";parallel7$;" 13 00 p";CHR$(13);:REMark s=start message to USB to I2C
converter, p=end of message to USB to I2C converter.
6200 REMark PRINT#3;"s 13 00 p";CHR$(13);:REMark s=start message to USB to
I2C converter, p=end of message to USB to I2C converter.
6210 non_print_reply:REMark Stops printing the USB to I2C reply also ensure serial
buffer is cleared.
6220 PAUSE 25
6230 NEXT a
6240 END DEFine allledflash
6250 :
6260 DEFine PROCedure ledchaseup
6290 FOR a=1,2,4,8,16,32,64,128,0
6300 PRINT#3;"s-";parallel8$;" 12 ";HEX$(a,8);" p";CHR$(13);
6310 non_print_reply
6320 PAUSE 1
6330 NEXT a
6340 FOR a=1,2,4,8,16,32,64,128,0
6350 PRINT#3;"s-";parallel8$;" 13 ";HEX$(a,8);" p";CHR$(13);
6360 non_print_reply
6370 PAUSE 1
6380 NEXT a
6390 FOR a=1,2,4,8,16,32,64,128,0
6400 PRINT#3;"s-";parallel7$;" 12 ";HEX$(a,8);" p";CHR$(13);
6410 non_print_reply
6420 PAUSE 1
6430 NEXT a
6440 FOR a=1,2,4,8,16,32,64,128,0
6450 PRINT#3;"s-";parallel7$;" 13 ";HEX$(a,8);" p";CHR$(13);
6460 non_print_reply
6470 PAUSE 1
6480 NEXT a
6490 END DEFine ledchaseup
6500 :
6510 DEFine PROCedure ledchasedown
6520 FOR a=128,64,32,16,8,4,2,1,0
6530 PRINT#3;"s-";parallel7$;" 13 ";HEX$(a,8);" p";CHR$(13);
```

```
6540 non_print_reply
6550 PAUSE 1
6560 NEXT a
6580 FOR a=128,64,32,16,8,4,2,1,0
6590 PRINT#3;"s-";parallel7$;" 12 ";HEX$(a,8);" p";CHR$(13);
6600 non_print_reply
6610 PAUSE 1
6620 NEXT a
6630 FOR a=128,64,32,16,8,4,2,1,0
6640 PRINT#3;"s-";parallel8$;" 13 ";HEX$(a,8);" p";CHR$(13);
6650 non_print_reply
6660 PAUSE 1
6670 NEXT a
6680 FOR a=128,64,32,16,8,4,2,1,0
6690 PRINT#3;"s-";parallel8$;" 12 ";HEX$(a,8);" p";CHR$(13);
6700 non_print_reply
6710 PAUSE 1
6720 NEXT a
6730 END DEFine ledchasedown
6740 :
6750 DEFine PROCedure ledchase
6760 PRINT#3;"s-";parallel8$;" 12 00 p s-";parallel8$;" 13 00 p s-";parallel7$;" 12 00
p s-";parallel7$;" 13 00 p";CHR$(13);:REMark s=start message to USB to I2C
converter, p=end of message to USB to I2C converter.
6770 non_print_reply
6780 FOR cc=1 TO 10
6790 ledchaseup
6800 ledchasedown
6810 NEXT cc
6820 END DEFine ledchase
```

The MCP23017 has 22 registers, do please look at the datasheet for all the functionality of these registers. However I will show in the following routines how to toggle the outputs for both “A” and “B” ports. The important thing to bear in mind is you have to send the start signal, then the device address, then the register address, the value for that register, then finally the stop signal. So for example to set the “A” port to output mode we send “s-40 00 00 p”. Then we can send “s-40 14 ff p”, this will then set all the outputs on port “A” high. 40 is the address “in hex” of the device if all the address links are in place, i.e. all the address pin are low. Also note all hex letters must be lower case.

I will not repeat the common routines you need to make all this work. See QLT Vol 16 issue 1. You need lines 1000 to 3090 from this listing.

Until next time, when I will be looking at a RS232 to I2C converter you can used with a ‘Black Box’ QL.

References

http://www.byvac.com/bv3/index.php?route=product/product&product_id=88

(Please note, I used the original V1 of the BV4221, ByVac now supply V2 which also has a SPI interface. The commands are the same, so the programs listed in the article should still work.)

http://www.byvac.com/bv3/index.php?route=product/category&path=44

PCF8570 Ram Data Sheet
http://www.nxp.com/documents/data_sheet/PCF8570.pdf

PCF8574(A) Data Sheet
http://www.nxp.com/documents/data_sheet/PCF8574.pdf
http://focus.ti.com/lit/ds/symlink/pcf8574.pdf

PCF8591 Data Sheet
http://www.nxp.com/documents/data_sheet/PCF8591.pdf

DS1307 RTC (Real Time Clock)
http://datasheets.maxim-ic.com/en/ds/DS1307.pdf

DS1803 Digital Potentiometer Data Sheet
http://datasheets.maxim-ic.com/en/ds/DS1803.pdf

MCP23017 Data Sheet
http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en023499

I2C Tutorials
http://www.robot-electronics.co.uk/acatalog/I2C_Tutorial.html
http://www.i2c.byvac.com/ar_foundation.php
http://doc.byvac.com/index.php5?title=I2C_Foundation#SPI

TF Services I2C manual
http://www.dilwyn.me.uk/docs/manuals/index.html

Advanced I2C information, but still worth a read to understand I2C protocols
http://www.nxp.com/documents/user_manual/UM10204.pdf