# QIC PROCESSOR CORE

V16

USER'S MANUAL

## How to contact Quanser Consulting:


| | |
|---|---|
| (905) 940-3575 | Telephone |
| (905) 940-3576 | Facsimile |
| 80 Esna Park Drive<br>Markham, ON<br>Canada L3R 2K8 | Mail |
| http://www.quanser.com/qic/ | Web |
| mailto://qic@quanser.com<br>mailto://support@quasner.com<br>mailto://info@quanser.com | Information about QIC<br>Support for QIC<br>General information |


## User's Manual



Printed in Canada.

# TABLE OF CONTENTS

# FIGURE INDEX

## Index of Figures

# INDEX OF TABLES

## Index of Tables



# 1 INTRODUCTION

The QIC suite of development components is aimed at simplifying the development of microcontroller-based applications. By integrating several commonly used functions into one convenient package, development time is greatly reduced. The user can quickly commence development of the external interfaces and the software, rather than spend time building the base microcontroller system. Using the QIC system as a reference design can greatly reduce implementation time of your designs.

The QIC system consists of two main hardware components. The first is the QIC Processor Core. This component hosts the microcontroller and the associated components to facilitate communications and programming. The second component is the QIC Carrier Board, which hosts an array of interfaces and signal conditioning options. Several different carrier boards are available, and are suited to different applications. Together, the two boards form an excellent reference design for many projects such as robotics, controllers, and various signal processing applications.

# 2 QIC PROCESSOR CORE

Based on the PIC16F87X series of microcontrollers, the QIC Processor Core offers all the features of the processor, plus the added ease of interfacing, programming and prototyping. Designed to fit into a standard prototyping breadboard, the QIC allows the user to design interface circuitry on a breadboard to quickly test new ideas.

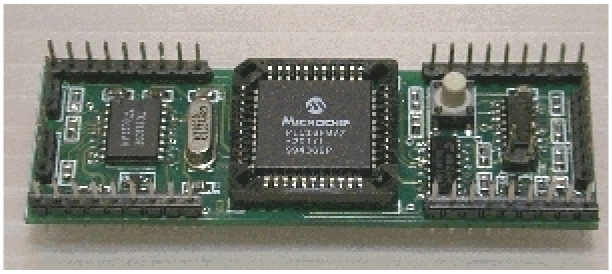


*Figure 1. The QIC Processor Core.*

## 2.1 PROCESSOR CORE FEATURES

Designed around the PIC16F877 microcontroller, the QIC Processor Core allows the user to take advantage of the vast amounts of freely available source code on the Internet. No special tools are required to program the processor core – the programming support hardware is already included. The user just needs to use the QIC Loader to download a new program. The code can be compiled using any of the available compilers for the PIC16FXXX series of microcontrollers.

- Based on the Microchip PIC16F877
  - 8K Flash program memory
  - 368 bytes RAM
  - 256 bytes EEPROM memory
  - 14 hardware interrupts
  - 3 hardware timers
  - 10 bit A/D (8 channels)
  - 10 bit PWM (2 channels)
- RS 232 compatible serial port – connects directly to the PC
- In-circuit serial programming support hardware built in
- On board reset switch.
- Single +5V supply required

## 2.2 PROCESSOR CORE PIN NAMES

Figure 2 shows the location of the pins and their associated names for the QIC Processor Core. For detailed information on the PIC microcontroller, programming, and special function details, refer to the Microchip website, at http://www.microchip.com. Document number **30292c** provides detailed programming information on the PIC16F87X series of microcontrollers. In most instances, the signal from the microcontroller is directly available on the pins of the QIC Processor Core. Please refer to the **schematics** section for details on this.

## 2.3 PROCESSOR CORE PIN LOCATIONS

Figure 2 also shows the pin locations and their numbering. Note how the pins are offset in their alignment to prevent the QIC Processor Core from being inserted incorrectly. The pins are numbered consecutively, from 1, counter clockwise from the top left.

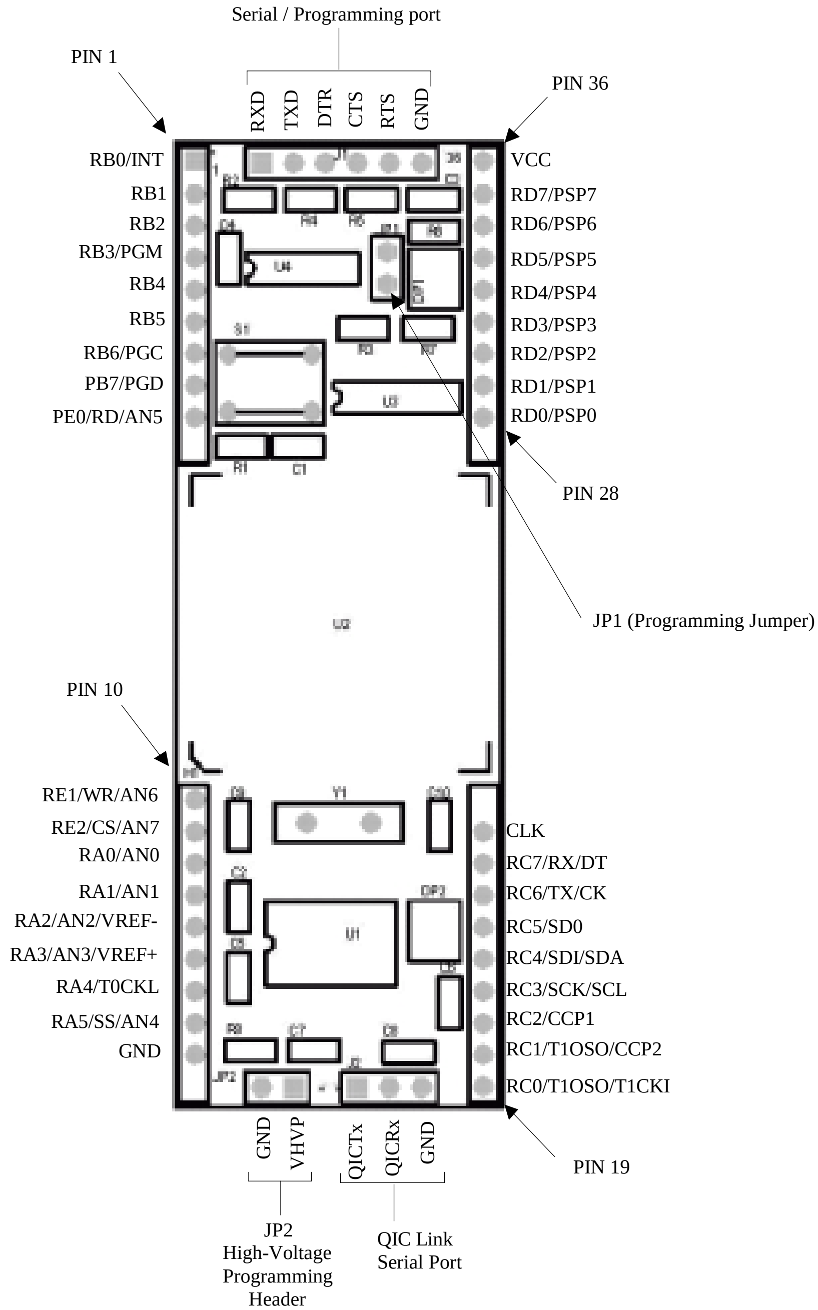



*Figure 2. The QIC Processor Core – Pin locations and functions.*

| QIC Core Pin Descriptions – Pins 1 – 18 | | | |
|---|---|---|---|
| **QIC Pin Number** | **Pin Name** | **Alternate Names** | **Description** |
| | | | PORTB is a bi-directional I/O port. The pins on this port can be programmed to use the internal weak pull ups. |
| 1 | RB0 | INT | Also serves as an external interrupt pin. |
| 2 | RB1 | | |
| 3 | RB2 | | |
| 4 | RB3 | PGM | Applying +5V volts will allow low voltage programming when the configuration bits are set appropriately. |
| 5 | RB4 | | |
| 6 | RB5 | | |
| 7 | RB6 | PGC | Interrupt-on-change pin. Also used as the serial programming clock |
| 8 | RB7 | PGD | Interrupt-on-change pin. Also used as the serial programming data |
| | | | PORTE is a bi-directional I/O port. |
| 9 | RE0 | /RD, AN5 | Read control for the slave port, or analog input 5 |
| 10 | RE1 | /WR, AN6 | Write control for the slave port, or analog input 6 |
| 11 | RE2 | /CS, AN7 | Chip select for the slave port, or analog input 7 |
| | | | PORTA is a bi-directional I/O port |
| 12 | RA0 | AN0 | Analog input 0 |
| 13 | RA1 | AN1 | Analog input 1 |
| 14 | RA2 | AN2, $V_{REF-}$ | Analog input 2, or negative analog reference voltage |
| 15 | RA3 | AN3, $V_{REF+}$ | Analog input 3, or positive analog reference voltage |
| 16 | RA4 | T0CKI | Clock input to Timer0 timer / counter. (open drain type) |
| 17 | RA5 | /SS, AN4 | Analog input 4, or slave select for the synchronous serial port |
| 18 | GND | GND | Ground / power return pin |

*Table 1. Pinout description for pins 1 - 18.*

| QIC Core Pin Descriptions – Pins 19 – 36 | | | |
|---|---|---|---|
| ***QIC Pin Number*** | ***Pin Name*** | ***Alternate Names*** | ***Description*** |
| | | | PORTC is a bi-directional I/O port |
| 19 | RC0 | T1OSO, T1CKI | Timer1 oscillator output or Timer1 clock input |
| 20 | RC1 | T1OSI, CCP2 | Timer1 oscillator input or Capture 2 input/Compare 2 output/PWM2 output |
| 21 | RC2 | CCP1 | Capture 1 input, Compare 1 output/PWM1 output |
| 22 | RC3 | SCK, SCL | Synchronous serial clock input/output for SPI and $I^2C$ modes |
| 23 | RC4 | SDI, SDA | SPI data in in SPI mode, or data I/O in $I^2C$ mode |
| 24 | RC5 | SDO | SPI data out in SPI mode |
| 25 | RC6 | TX, CK | USART Asynchronous transmit or synchronous clock |
| 26 | RC7 | RX, DT | USART Asynchronous receive or synchronous data |
| 27 | CLK | | Processor clock source. This is an external clock (20MHz) for use with external peripherals |
| | | | PORTD is a bi-directional I/O port. It also serves as the parallel slave port when interfacing to a microprocessor bus |
| 28 | RD0 | | |
| 29 | RD1 | | |
| 30 | RD2 | | |
| 31 | RD3 | | |
| 32 | RD4 | | |
| 33 | RD5 | | |
| 34 | RD6 | | |
| 35 | RD7 | | |
| 36 | VCC | | +5VDC Power |

*Table 2. Pinout description for pins 19 - 36*

| Serial / Programming Port Pin Descriptions | | |
|---|---|---|
| **Pin Number** | **Pin Name** | **Description** |
| 1 | RxD | PC Receive data line (RS232 level). Data from the QIC is transmitted on this line to the PC |
| 2 | TxD | PC Transmit data line (RS232 level). Data is received by the QIC on this line from the PC |
| 3 | DTR | PC Data Terminal Ready line. Used for programming purposes |
| 4 | CTS | PC Clear To Send line. Used for programming purposes |
| 5 | RTS | PC Ready To Send line. Used for programming purposes |
| 6 | GND | Ground line |

*Table 3. Serial / Programming port connections.*

| QIC Link Pin Descriptions | | |
|---|---|---|
| **Pin Number** | **Pin Name** | **Description** |
| 1 | QIC Link Rx | Data transmitted from other QIC modules is sent to the serial port, bypassing the QIC (RS232 level). This data is seen by the host only. |
| 2 | QIC Link Tx | Data received at the serial port is retransmitted via this pin to other QIC modules (RS232 level). The data appears as if it came from the host directly. |
| 3 | GND | Ground connection. |

*Table 4. QIC Link serial port connections.*

| High Voltage Programming Pin Descriptions | | |
|---|---|---|
| **Pin Number** | **Pin Name** | **Description** |
| 1 | VHVP | +12VDC input for high-voltage programming. |
| 2 | GND | Ground connection for high-voltage programming |

*Table 5. High-Voltage programming support connections.*

# 3 QIC PROCESSOR CORE ACCESSORIES

To simplify the development process, the QIC Processor Core has built in serial support and in-circuit programming support. This section has detailed information on these two interfaces.

## 3.1 IN-CIRCUIT SERIAL PROGRAMMING HARDWARE

To make programming of the QIC Processor Core easier, the circuitry required for in-circuit serial programming has been included directly on the board. The circuitry is designed to allow for in-circuit serial programming when required, but to leave the programming pins available for other functions when the device is not being programmed. To enable this dual configuration, a jumper, **JP1** is used to select between programming mode and normal operation. In the event that the programming pins (RB6 and RB7) are not required for the user's application, the jumper can be left in place without affecting the normal operation of the system. In effect, installing the jumper (JP1) connects the pins RB6 and RB7 to the in-circuit serial programming hardware. Removing the jumper connects pin RB6 to pin 7 of the QIC core, and pin RB7 to pin 8 of the QIC core. Figure 3 shows the circuit details that achieve this functionality.

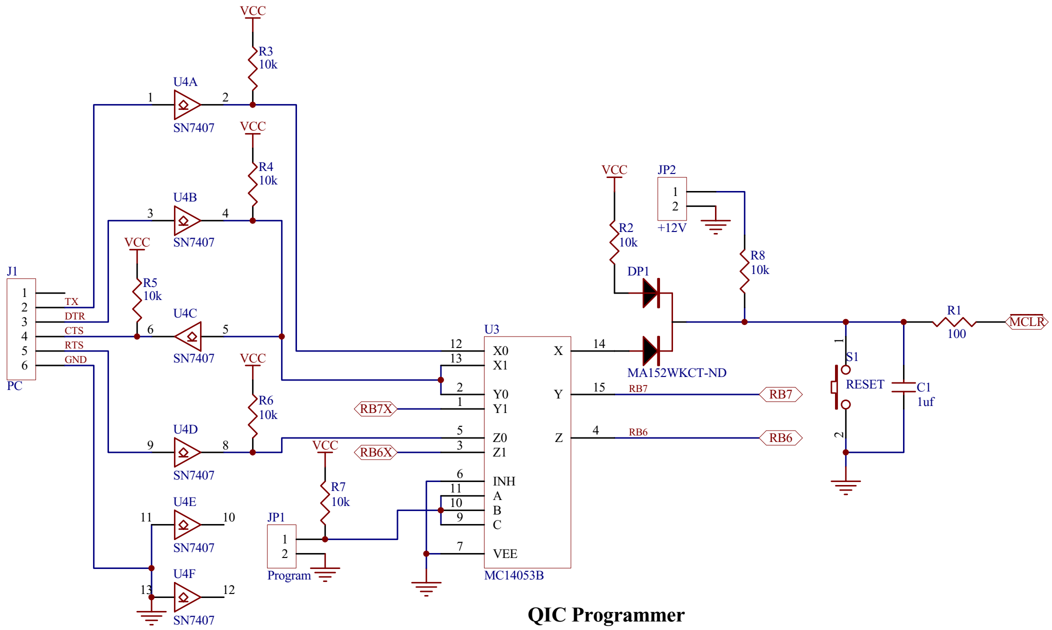



*Figure 3. In-circuit serial programming support circuitry.*

### 3.1.1

### 3.1.2 HIGH-VOLTAGE PROGRAMMING

Programming voltage is applied to header **JP2** (see Figure 5). +12VDC should be applied to pin 1, and the ground for the external source should be connected to pin 2. Refer to Figure 3 for details. Applying the voltage at this connector supplies the required high voltage signal to put the PIC16F877 into the programming mode. The voltage for high voltage programming can be obtained from any regulated external power source. Using the QIC Loader, the user can quickly program the device with their own code. In addition, the user can write their own programmer to use the on-board in-circuit serial programming hardware. Microchip has documents detailing in-circuit serial programming of the 16F877 processor.

Please note that the PIC16F877 will be put into programming mode when both RB6 and RB7 are held low, and MCLR is brought up to the required high voltage programming voltage. This condition is most easily obtained by setting up the high voltage programming voltage, holding RB6 and RB7 low, and waiting for the user to press the reset button on the QIC Processor Core. This procedure guarantees that when the reset button is released the PIC16F877 will go into the program mode.

##### 3.1.2.1 High-Voltage Programming Procedure.

- Install a jumper on **JP1**, shorting the pins together.
- Apply **+12V** to **JP2**. +12VDC should be on pin 1, ground (GND) on pin 2.
- Connect the programming cable to **J1**, the programming/serial port.
- Start the programming software and follow the directions.

### 3.1.3 LOW-VOLTAGE PROGRAMMING

With the configuration register of the PIC16F877 set appropriately, it is possible to program the device using only +5VDC. By default, when fully erased, the PIC16F877 will allow low-voltage programming (LVP)[1]. To put the PIC16F877 into low-voltage programming mode, the user will need to apply +5VDC to the PGM pin (RB3), or pin 4 of the QIC Processor Core (see Figure 5). Note that when in this mode, RB3 cannot be used for any other purpose. In effect, this pin is lost to the user under software control. The only function of the pin is to put the PIC16F877 into low-voltage programming mode when the LVP bit of the configuration register is set[2]. As with the high-voltage programming mode, the user can either use the QIC Loader, or write their own software to program the device. Microchip has documents detailing in-circuit serial programming of the 16F877 processor.

Please note that the PIC16F877 will enter programming mode when both RB6 and RB7 are held low, and +5VDC is applied to the PGM pin (pin 4 of the QIC core). With these conditions present, pressing the reset button will guarantee that the PIC16F877 will enter programming mode when the reset switch is released.

##### 3.1.3.1 Low-Voltage Programming Procedure

- Install a jumper on **JP1**, shorting the pins together.

1 Please consult the Microchip document,"EEPROM Memory Programming Specification" document number 30262c (30262c.pdf, www.microchip.com )

2 Note that if the LVP (low voltage programming) bit is cleared – disabling LVP – this method of programming will no longer be available. The user will have to resort to high-voltage programming.

- Apply **+5VDC** to the **PGM** pin (**pin 4** of the QIC core). A good source for this is the power going to pin 36, VCC.
- Connect the programming cable to **J1**, the programming/serial port.
- Start the programming software and follow the directions.

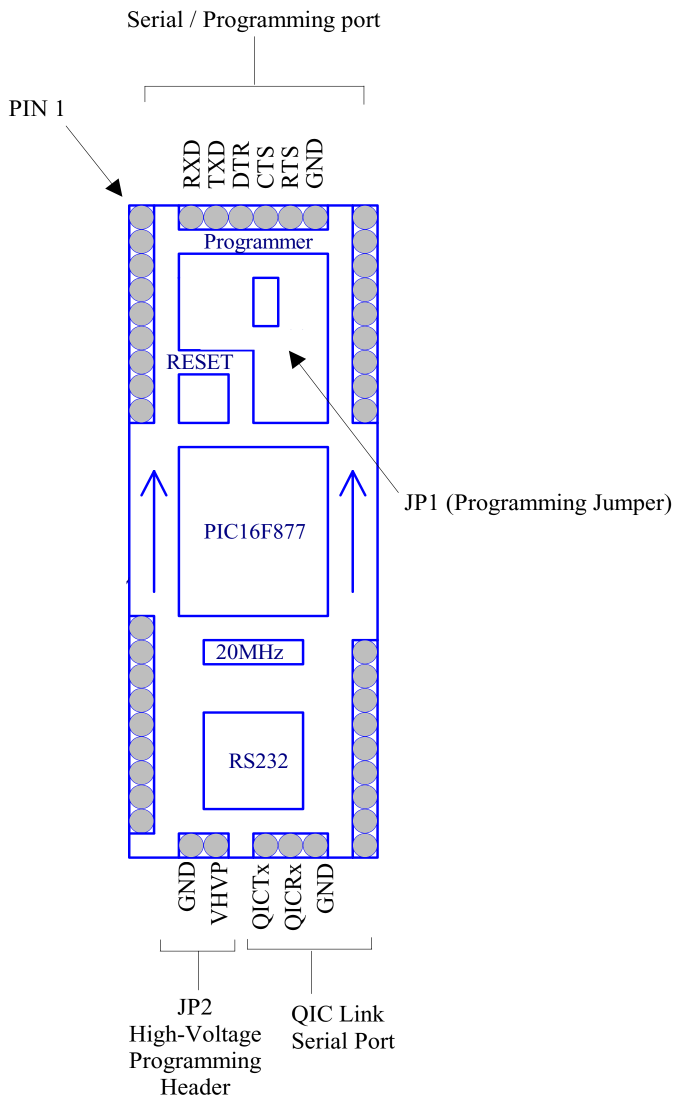



*Figure 4. Serial pin locations.*

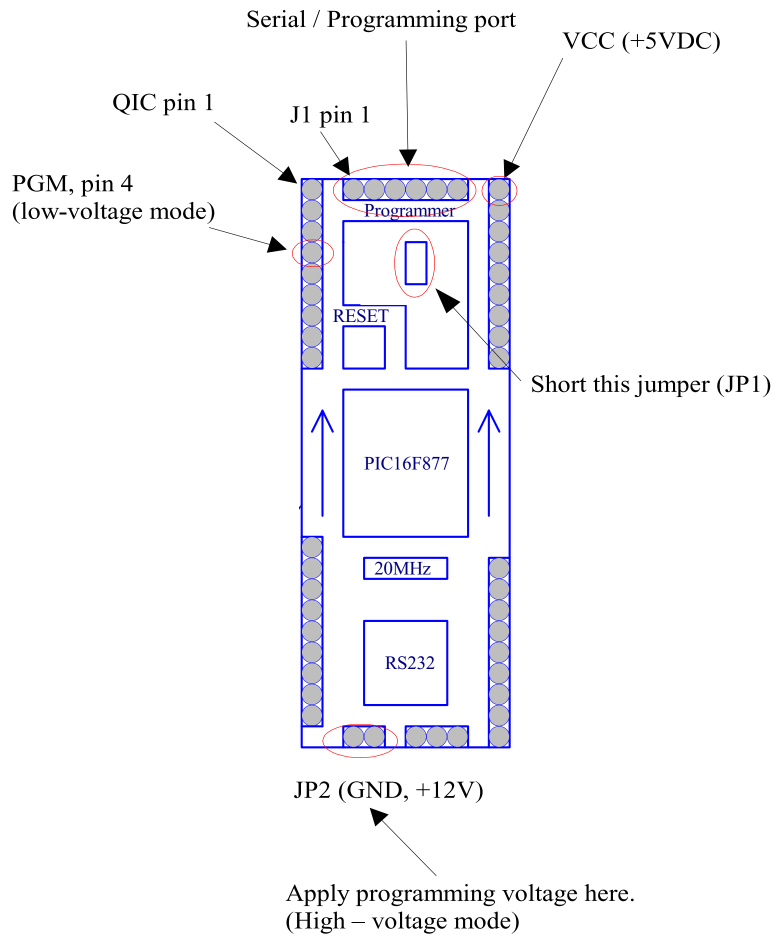



*Figure 5. Serial programming and connection setup details.*

## 3.2 ON-BOARD SERIAL SUPPORT

The on-board serial interface is shown in logical format in Figure 6. With an external connection to J1, the serial port connector, the device will be able to communicate with the QIC Processor Core. Signals transmitted from the PC (TxD) are received by the PIC on RC7, which is connected directly to the microcontroller's USART. Using the USART allows for the greatest flexibility in serial support, while minimizing the required software overhead.

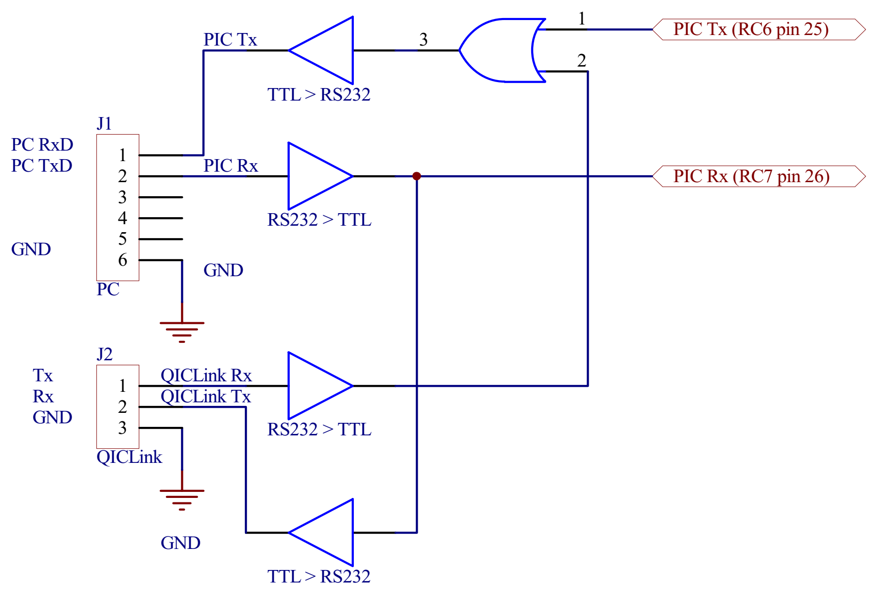



*Figure 6. Serial interface logical diagram.*

Similarly, serial data is shifted out of the USART on RC6, and is sent to the PC's RxD pin on the serial port. The serial interface uses a combined RS232 to TTL / TTL to RS232 converter to provide the proper signal levels for the PC and the PIC16F877.

The QIC Link port echoes the serial data received at J1 on connector J2.By connecting additional devices, to the QIC Link port, they will also be able to communicate with the host PC (or host device).

The full serial interface schematic is shown in Figure 7.

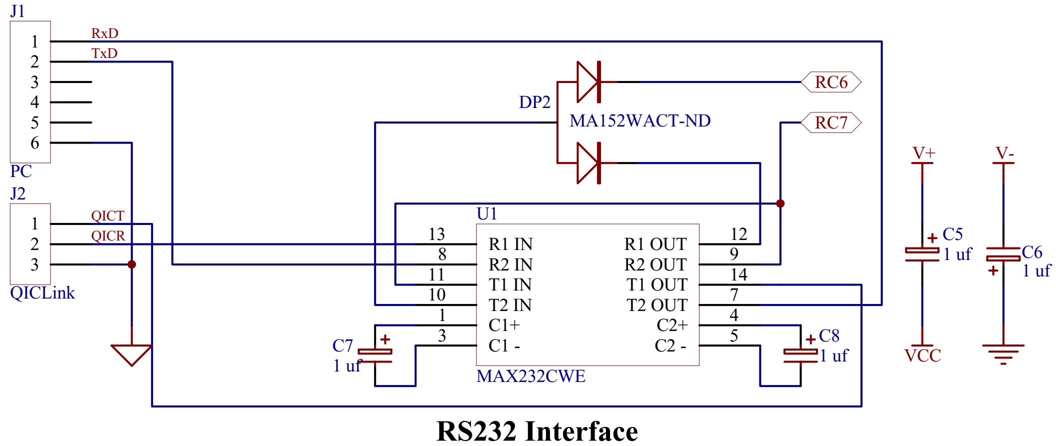



*Figure 7. The serial interface circuitry.*

### 3.2.1 RS232 COMPATIBLE SERIAL PORT

Details of the connections for the serial port interface are shown in Figure 8. These connections will need to be made in order to connect the PC to the QIC Processor Core. The connections shown are for a female DB9 connector. Note that only the signals TxD, RxD and GND are required for serial communications. The other lines are required for in-circuit serial programming only, but are shown here for reference. Referring to Figure 7, the MAX232CWE IC takes care of the conversions between TTL and RS232. All signals at connector J1 are expected to be at standard RS232 levels.

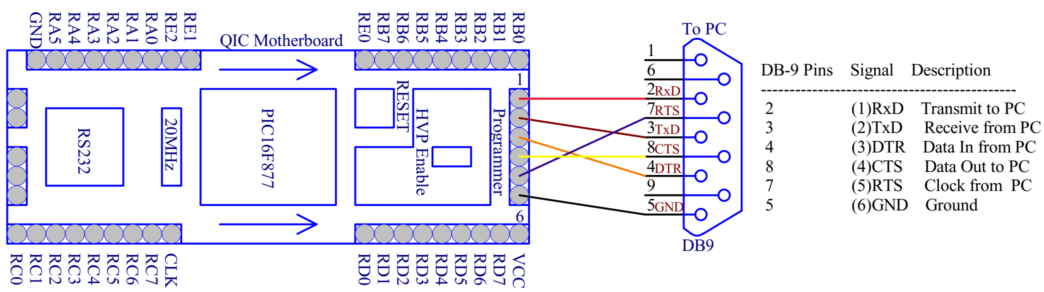



*Figure 8. Serial interface to PC wiring.*

When writing software to use the serial port, the user must be careful to specify RC6 for transmitting serial data, and RC7 for receiving serial data. Some compilers allow for software implemented serial ports on *any* I/O pins of the device. The user will also need to take care of the RS232 to TTL level conversion. RS232 level signals should not be directly applied to any pins except the serial port interface or the QIC Link interface.

### 3.2.2 QIC-LINK SERIAL PORT

The QIC Link serial port is intended to facilitate multi-device communications. The topology of the communication network is "host-to-slave". What this means is that the host PC can communicate with any of the QIC Processor Core modules. The QIC cores can in turn talk to the host PC. Refer to Figure 6 and Figure 7 for details on the implementation of this circuit.

Figure 9 details the network connections necessary to link several QIC Processor Core boards together. **Although it appears that data flows through the previous QIC core in the network chain, this is not the case**. QIC cores connected in this manner will only be able to talk directly to the host PC. Other QIC cores will not see the serial communications data. **However, every QIC on the network will see serial communications sent from the host PC**. For this reason, the user will need to implement some type of protocol to manage the network traffic. Each QIC Processor Core on the network will need to be assigned an address so that it can determine if the data being sent is intended for it. Many different methods exist for implementing this type of network negotiation. The exact type of protocol used will depend on the actual application requirements.

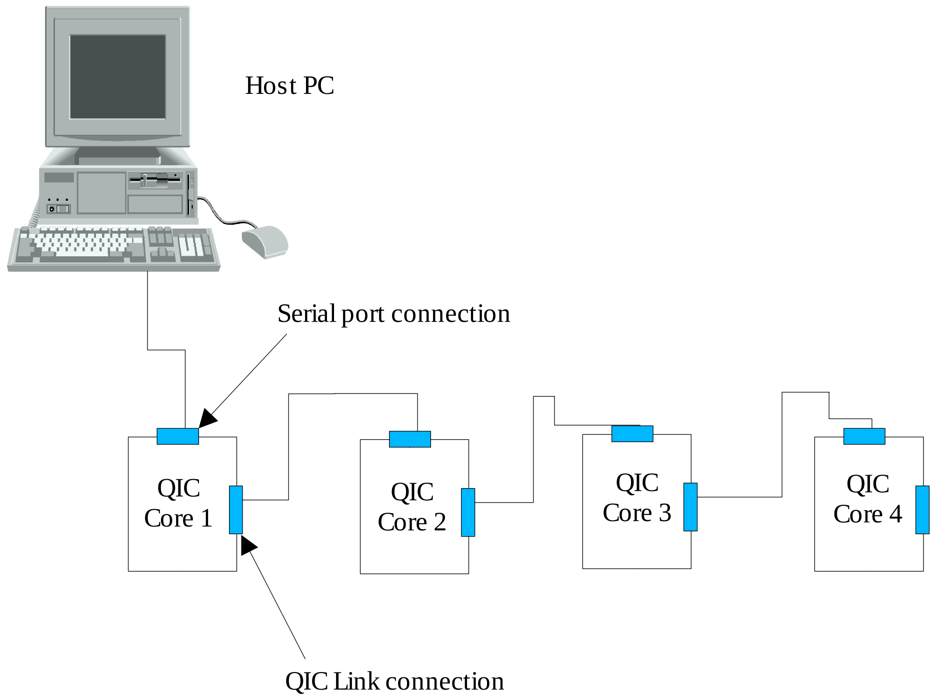



*Figure 9. QIC Link network connections.*

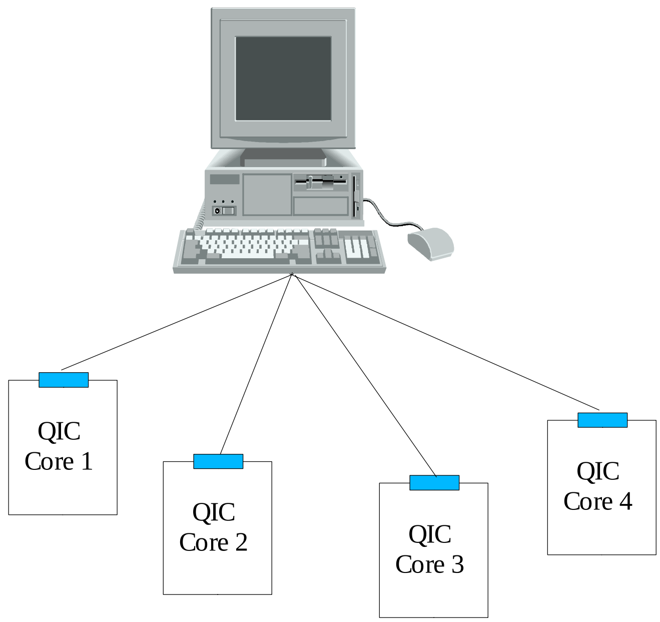



*Figure 10. "Star" network topology of QIC Link.*

# 4 SUPPORT SOFTWARE

Programming of the QIC Processor Core can be achieved in a variety of ways. Using the included support software such as the QIC Loader and the QIC Bootloader, the user can quickly develop and test new applications. Depending on the end use of the application, the user may choose the QIC Loader approach over the QIC Bootloader approach or vise-versa.

## 4.1 USING THE QIC LOADER

The QIC Loader approach allows programming of the QIC Processor Core while it is still in the user's circuit. Using the loader approach is the more traditional of the programming methods in that the programmer erases the entire memory contents of the device and rewrites it with the user's new code. Using this approach, all areas of the memory space can be modified; code memory, data EEPROM memory, and configuration memory. Since this method allows setting of the code protection bits in the configuration memory registers, it is usually the preferred method for final product deployment.

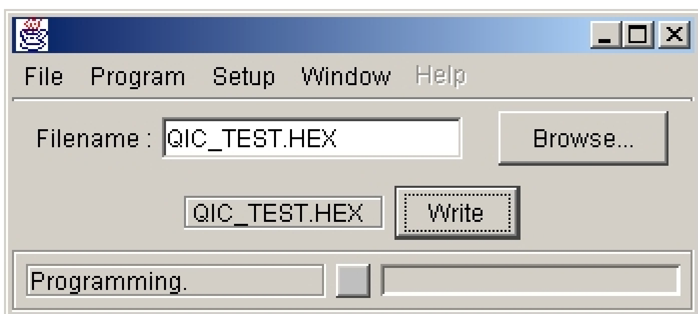



*Figure 11. QIC Loader window. Select a file by clicking on Browse.*

Programming can be accomplished using either high-voltage or low voltage mode. Depending on the programming mode selected, follow the appropriate procedure below.

### 4.1.1 HIGH VOLTAGE PROGRAMMING

- Install a jumper on **JP1**, shorting the pins together.
- Apply **+12V** to **JP2**. +12VDC should be on pin 1, ground (GND) on pin 2.
- Connect the programming cable to **J1**, the programming/serial port. The other end of the cable should connect to the serial port of the PC.
- Apply power to the QIC Processor Core.
- Start the QIC Loader and select an Intel Hex formatted file to load.
- Make sure that the correct serial port is selected. The default port is COM1. Select a different one by selecting Setup -> Serial Port and selecting from the available choices.
- Click the Write button to begin the programming process.
- Follow the directions on the screen. The Reset button referred to on the screen is the reset button on the QIC Processor Core.
- A dialog will appear confirming the success or failure of the programming procedure.

### 4.1.2 LOW VOLTAGE PROGRAMMING

- Install a jumper on **JP1**, shorting the pins together.
- Apply **+5VDC** to the **PGM** pin (**pin 4** of the QIC core). A good source for this is the power going to pin 36, VCC.
- Connect the programming cable to **J1**, the programming/serial port. The other end of the cable should connect to the serial port of the PC.
- Apply power to the QIC Processor Core.
- Start the QIC Loader and select an Intel Hex formatted file to load.
- Make sure that the correct serial port is selected. The default port is COM1. Select a different one by selecting Setup -> Serial Port and selecting from the available choices.
- Click the Write button to begin the programming process.
- Follow the directions on the screen. The Reset button referred to on the screen is the reset button on the QIC Processor Core.

- A dialog will appear confirming the success or failure of the programming procedure.

In most cases, the programming will complete successfully. However, there are times when errors will occur. The error message shown in Figure 12 typically means that the PC cannot detect the hardware. This error will occur before the programming begins.

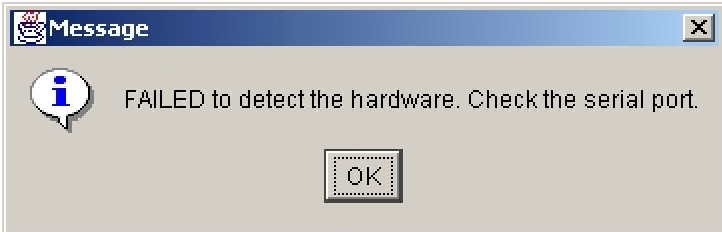



*Figure 12. Programmer error message.*

Typical causes of this are:

- Serial cable unplugged from the PC or the QIC Processor Core
- Wrong serial port selected in the QIC Loader
- Power is not applied to the QIC Processor Core.
- Damaged serial cable

The causes listed above should be checked to remedy the problem before trying to continue.

Other errors may occur during the programming procedure. The error shown in Figure 13 is indicative of a programming error.


*Figure 13. Typical error message encountered during programming.*

Typical causes of this are:

- Serial communications error; check symptoms of failed hardware first.
- Jumper JP1 has not been installed for programming.
- No programming voltage applied for high-voltage programming, or PGM voltage not present for low-voltage programming.
- Reset button not pressed when requested by the software.
- Code protection bits are set in the configuration memory area. Erase the entire device before continuing.
- Processor is faulty or damaged.

Use of the programming software is fairly straight-forward. There are additional tools for viewing the program and data memory, as well as for erasing an PIC16F877 and restoring to its default condition.

## 4.2 USING THE QIC BOOTLOADER

The QIC Bootloader approach takes advantage of the processor's ability to reprogram its *own* code space. To start, a small program called the bootloader is programmed into the device using the QIC Loader. This bootloader occupies a very small portion of the code space near the top of memory. Once loaded, upon processor reset, the bootloader looks for a serial "signature" at the serial port. If it receives this signature, it then goes into the bootloading mode, otherwise, it begins executing the previously loaded user code. In the bootloading mode, an Intel Hex type file is transferred to the device using a standard serial protocol. Typically, a serial terminal program such as Hyperterminal or the PIC Downloader can be used to send a new file. The previously installed bootloader receives the code and reprograms the PIC. This method is extremely useful for rapid development of new designs.

In order to use the bootloader, ***the bootloader code must be downloaded to the processor first.*** Otherwise, the bootloading will not be possible. Once the bootloader is installed, neither high-voltage programming, nor low-voltage programming will be required. For this reason, the user may wish to disable the low-voltage programming option by setting the appropriate configuration bits. ***Once the bootloader code is loaded into the program memory, only the serial port connections will be required.*** To use the PIC Downloader, start the program and select an Intel Hex formatted file for download. Search for this file by clicking the search button, or pressing F2.

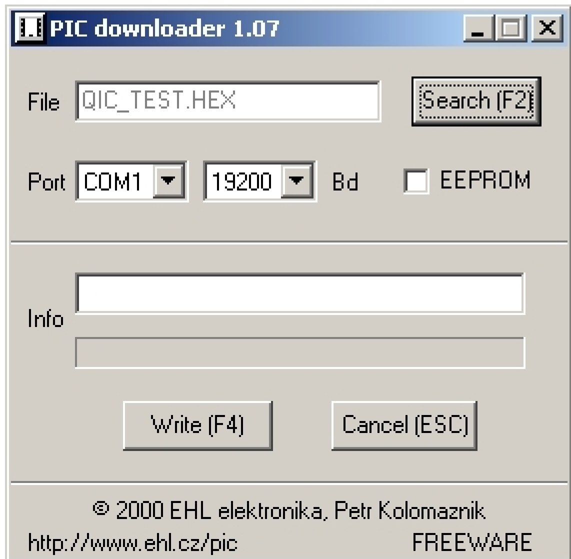



*Figure 14. PIC Downloader interface window.*

Once the file is loaded, press Write (or F4) to begin the programming process. The reset button on the QIC Processor Core should be pressed at this time. After a brief pause, the download will commence.

If problems arise during the programming process, check the following list of possible causes.

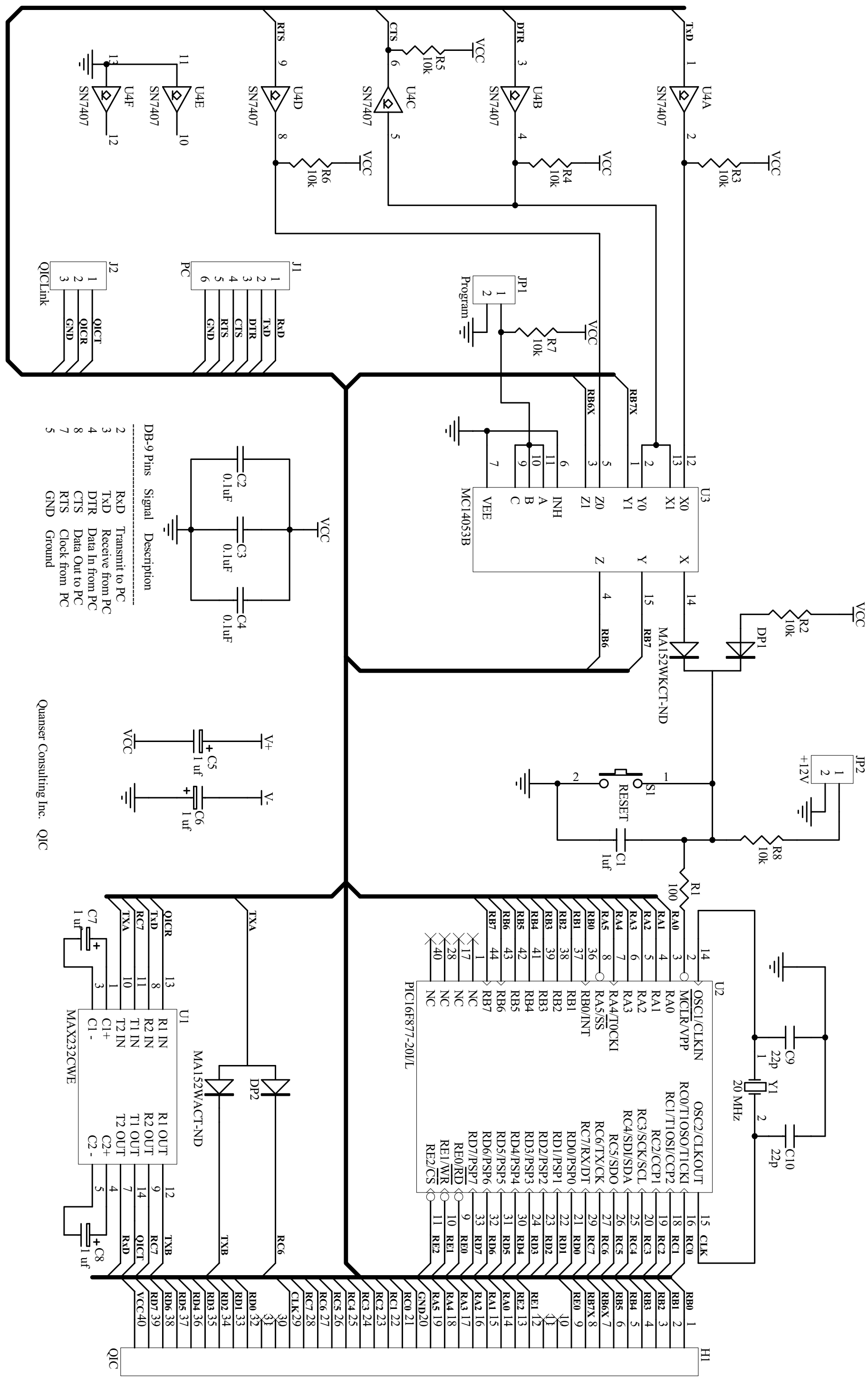
- Make sure the serial cable is attached, and that power is applied.
- Be certain that the bootloader code has already been written to the device using either high-voltage or low voltage programming.
- Verify that the serial port and the baud rate are correct for the bootloader that is being used.

# 5 SCHEMATICS



Quanser Consulting Inc.
80 Esna Park Drive,
Units 1,2,3
Markham, Ontario
Canada
L3R 2R6
Tel : +1 (905) 940 - 3575
www.quanser.com
mailto://qic@quanser.com

## 5.1 QIC PROCESSOR CORE – V16


MAX232CWE
PIC16F877-20I/L
MC14053B
MA152WACT-ND
MA152WKCT-ND
SN7407
RESET
20 MHz
Quanser Consulting Inc. QIC

# Alphabetical Index