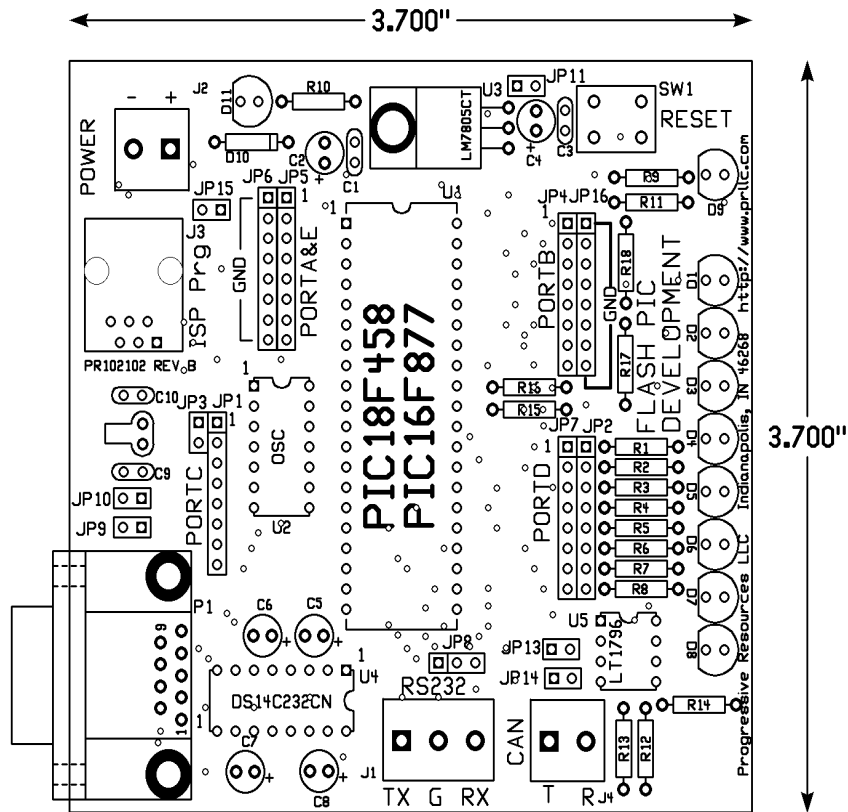


Other features of the development board include:

- RS232 through a 9-Pin D-Shell as well as screw terminals and a jumper header.
- Up to 32K words of In-System Programmable FLASH memory with up to 256 bytes of EEPROM and up to 1.5K of Internal RAM (depending on processor selection).
- Up to 8, 10 bit, Analog Inputs, using either internal or user supplied reference.
- 9 I/O controlled LEDs, 8 of which are jumper selectable.
- 32KHz “watch” crystal for on-board Real-Time operations.
- A universal clock socket allows for “canned oscillators”, as well as a variety of crystals, ceramic resonators, and passive terminations.
- 0.1” centered headers provide for simple connection to the processor special function pins and I/O.
- A 6-pin, ICD connection is provided for in-system programming and debugging. This connection is directly compatible with the Microchip ICD, ICD2 and CCS ICD-S programming hardware. Flash PICs can also be programmed through RS232 using an appropriate boot loader application.
- On-board regulation allows for power inputs from 8-38VDC with an LED power indicator.
- Termination is provided for 5VDC output at 250ma

SPECIFICATIONS

- | | |
|-------------------------|--|
| - Voltage range | 8 V to 38 VDC |
| - Power consumption | 250 mW (nominal) |
| - Dimensions: | W 3.7 inches x H 3.7 inches |
| - Mounting | Rubber feet, 4 places |
| - Weight | ~3 OZ |
| - Operating temperature | 0 deg. C to +60 deg. C |
| - Storage temperature | 0 deg. C to +85 deg. C |
| - Humidity | 0% to 95% at +50 deg. C (non-condensing) |



APPLICATION NOTES:

Power

J2 (screw terminal connector) is the power input point. Acceptable voltages are 8 – 38 VDC (J1-1 is +, J1-2 is ground).

JP11 is an output of the regulated 5 VDC that may be used to power other devices. Note, however, that the LM7805 does not have a heat sink and so the actual available power output is somewhat limited, depending on the input voltage and power being consumed. Check the LM7805 regulator specification for details.

Serial Connection

P1 is a standard DB-9 connector usually used to connect to a PC. The TX signal is on P1-2 and the RX signal on P1-3. These are RS-232 level signals.

J1 and JP8 also provide connections for the RS-232 level serial signals. On each connector, pin 1 is the TX signal, pin 3 is the Rx signal and pin 2 is ground.

JP9 and JP10 are jumpers, which connect the processor serial signals (RXD and TXD) to/from the RS-232 driver chip. These jumpers must be in place for the RS-232 serial connections to work. Removing the jumpers allows use of the Port D, bits 0 and 1, for TTL-level I/O.

SPI Connection

Use of the SPI bus is by making connections to the SPI bus signals on Port C (JP1-4, 5, and 6).

Parallel Ports

Parallel ports A/E, B, C, and D are connected to JP5, JP4, JP1, and JP7, respectively (and labeled clearly on the board). Bits are connected sequentially with bit 0 on pin 1, bit 1 on pin2, etc. These are normal TTL-level signals with or without pull-ups depending on the port initialization set up in the software.

Port A/E (JP5) has a parallel row of ground pins next to it (JP6) providing a convenient ground reference when measuring analog voltages with the internal A/D converter (ADC).

Port B (JP4) has a parallel row of ground pins next to it (JP16) so that enabling the built-in pull-up resistors and then using two-pin jumpers to ground any pins that need a logic 0 applied for input purposes can effect simple input signals.

Port C (JP1) has a parallel set of pins (JP3) at positions 1 and 2. A clock crystal (32.768 KHz) is connected to JP3. JP3 is located adjacent to JP1 (Port C) to provide an easy connection of the clock crystal to Port C bits 0 and 1 for use as a real-time clock.

Port D (JP7) has a parallel row of pins (JP2), each of which is connected to an LED through a 510-ohm series resistor to +5 VDC. Jumping any of the pins of JP7 to the corresponding pin of JP2 allows the use of the on-board LED's as an output. Because the LED's are connected to +5 VDC and the port is sinking the LED current, the LED will be on for any pin that outputs logic 0.

System Clock

As supplied, the system clock is 10 MHz. U2 contains the crystal and caps necessary for the oscillator. Replacing U2 with a TTL, crystal, ceramic resonator, or RC oscillator, or a different integrated oscillator unit allows changing the system clock if necessary.

CAN Interface

A CAN interface driver socket it provided (U5) for a Linear Technologies, LT1796 CAN bus interface. Jumpers JP13 and JP14 connect the CAN interface to the appropriate pins on the controller (CANTX and CANRX). The CAN interface is intended for use with the PIC18F45x microprocessors that have an actual CAN

transceiver built in. Refer to Microchips website and datasheets for details on the CAN bus controller and its features.

PIC16F877/18F458 PIC® Boot Loader

The boot loader runs on the PIC's UART at 9600 baud, XON, XOFF handshaking, 8 data bits and 1 stop bit.

The boot loader code is executed upon reset or power up of the processor. If the boot loader does not receive instructions to load a new file with 5 seconds, then it jumps to the application code.

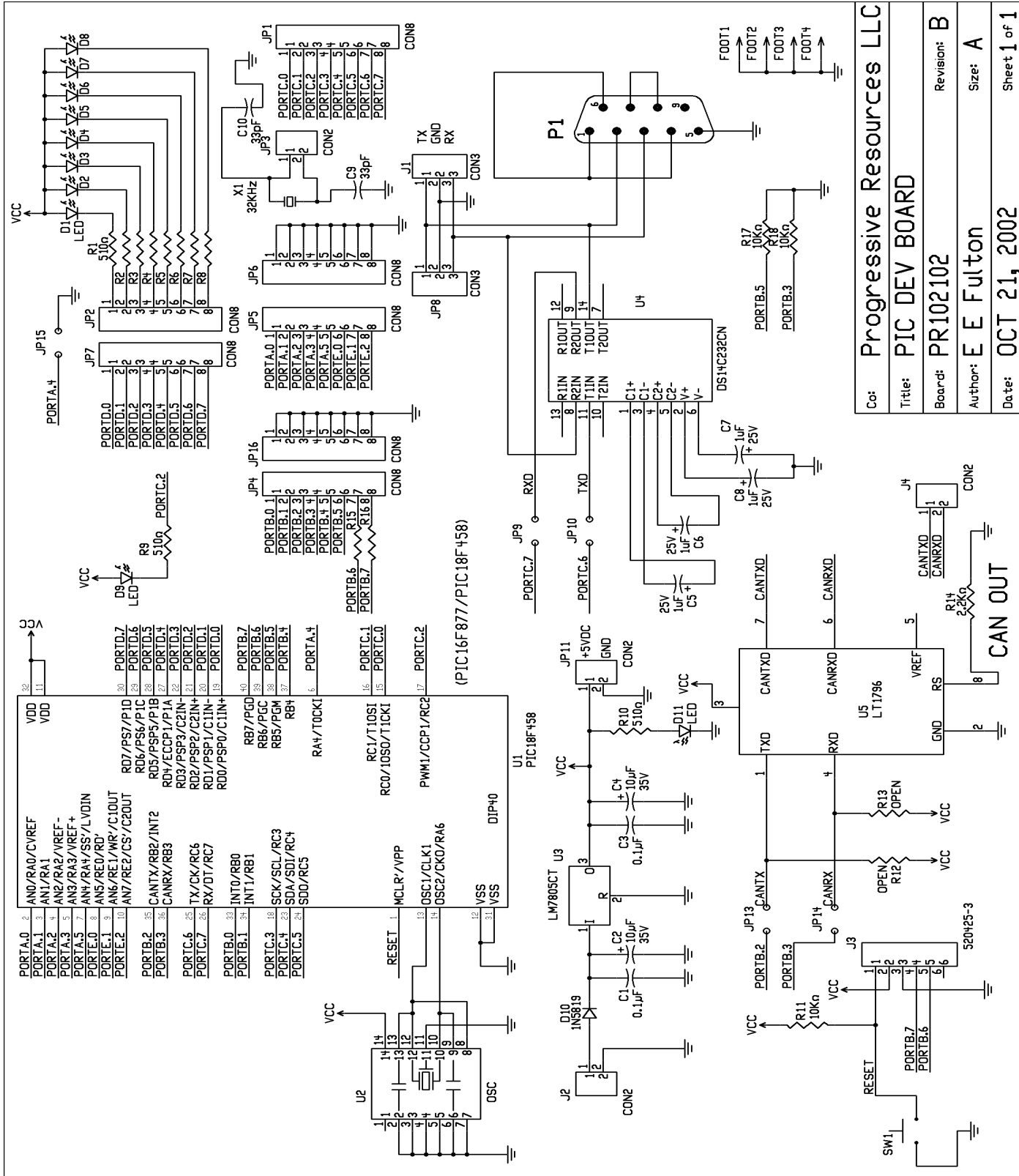
After the PICBL is started (via a reset or a power-up), the following protocol must be observed. The PC application provided handles the protocol for you with its default settings or you can create a custom application.

1. Upon power-up or reset, the PICBL sends a '^' (BOOTLOADER_ACTIVE_CHAR) at the selected baud rate. This is the character used to tell the external program that the PICBL is active and can enter "programming mode" at this time.
2. The host is then required to send the three-character entry sequence. This is used to prevent an inadvertent attempt of reprogramming from taking place. If the PICBL does not receive these characters within the timeout period, then the PICBL jumps to the application, if there is no application code loaded, execution stays within the PICBL indefinitely, waiting for the entry sequence.
3. Once the three-character entry sequence has been sent, the bootloader sends the version string (vSx.xx for PIC16 parts, vLx.xx for PIC18 parts) followed by the bootloader's starting address, the "goto bootloader" op code programmed at address 0x0000, and then a '?' (READY_CHAR).
4. Upon receipt of the READY_CHAR, the host application should send the hex file for the new/updated application program observing an X-ON / X-OFF handshaking protocol to control data flow. The handshaking is very important as the flash memory area may write data slower than the serial port can send data. The programming software continues sending the hex file until it is all sent. After each line of ".hex" file is received by the bootloader, one of two characters is transmitted by the bootloader:
 - '~' Line received with no errors.
 - '-' Checksum error the detected while receiving the line.
5. After the programming is complete, the PICBL sends either a '#', meaning the programming is all right, or an '@' indicating that an error has occurred and the program did not load 100% successfully. In most cases an error during programming means that the main application program is corrupted and will need to be resent.

6. The PICBL then jumps into the programmed application software.

One final note about the **PIC16F877/18F458 PIC® Boot Loader**, the boot loader source code is available for purchase and as such provides the opportunity for customized boot loader solutions. The PIC boot loader is available at <http://www.prllc.com>.

SCHEMATIC



Co:	Progressive Resources LLC
Title:	PIC DEV BOARD
Board:	PR102102
Author:	E E Fulton
Date:	OCT 21, 2002
Revision:	B
Size:	A
Sheet	1 of 1