



Manual PIC 40 Basic Development Board

Amer Iqbal Qureshi

40 Pin PIC Microcontroller Basic Board

PIC Microcontrollers are fairly popular in industry as well as with students and hobbyists. Their easy availability, reasonably low price and rich integration of peripheral devices have made them a hot cake.

Each microcontroller project would need a special customized circuitry to handle the specific job, yet there is always need for a basic microcontroller circuit as well. Many students and hobbyists find it difficult to make the same basic circuit every time. This board has been made for this purpose. It has the basic layout of PIC microcontroller, along with most commonly used plug-ins like LCD, switches, status LEDs etc.

The beauty of this board is that all plug-ins have their own connection headers, and you can connect whatever combination you want to the main controller. Rest of the I-Os are free for your specialized external circuit that might be on a breadboard or a veroboard.

In this way the same controller board can be used in wide variety of projects. When the prototype experiments are done, you can make a complete project on your own board.

The board supports all 40-pin PIC microcontrollers that are pin compatible.



Each on-board module like LCD for example has a header with all the interfacing lines labeled. You can use simple jumper wires to connect the pins to I-O lines of microcontroller of your choice. We have tried to keep the most common I-O line connections close to the header so that long messy jumpers are not needed. You are however free to connect the device in any combination you like.

Two rails of GND and 5V supply are also provided close to 15 I-O lines. So that interfacing with external devices like sensors etc is made easy.

The board supports In circuit serial programming. This ensures that the controller remains in its place and you do not need to unplug it.

Moreover these 40 pin Microcontrollers can be programmed with a bootloader. Which will eliminate the need of an external programmer altogether. A short guide of this is given at the end of this

manual.

On Board Features:

USART with MAX 232 Level Conversion

The USART module contains MAX232 level converter IC for easy communication with PC and other devices requiring level conversion. The DB-9 Female type connector is standard serial port connector. You will need a standard serial cable with male and female connectors to connect this port to your PC serial port. Only Basic communication design has been followed the output of MAX 232 is provided as a two pin header labeled as TX and RX. The TX pin will bring data in, and Rx pin will take data out to the PC.

Close to it a header with RC7 and RC6 of microcontroller pins is located. These pins are RX and TX of hardware USART module present in most PIC microcontrollers. Most of the times all you need is to connect the jumpers from Tx to Rx on these jumpers as shown above.

In case your application needs to implement serial communication through some other I-O lines, just remove the jumpers, and using standard No. 22 gauge wire connect the TX and RX to your desired connections



Character LCD Interface

Character LCDs are fairly commonly used by students and hobbyists in their prototype projects to see various results of their project. This board has hardware design for most popular HD44780 based character LCDs. All types can be used, 1x16, 2x16 or 4x16 or even 20 x 4. Usually the LCD interface has a variable resistor to adjust contrast we have however replaced it with fixed 1.5K resistor.

The LCD interface allows you to connect the LCD either in 8 or 4 bit mode. The data pins are labeled as D0 to D7. E is enable pin and RS is register select.

RW pin is directly connected to GND to make it always write.

The backlight connections are directly made. Pin 16 is GND and pin 15 is VCC for back light.



Status LEDs

4 Status LEDs are provided. These have been configured as active high. The cathode is connected to GND and anodes have 220 ohms current limiting resistors.

The jumpers are provided to connect the status LEDs to I-O Line of your choice.

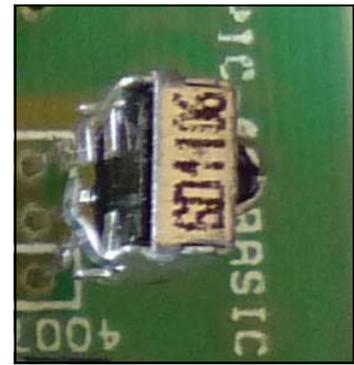
38 KHz IR sensor

38 KHz Infra red sensor is provided with output line pulled up using a 10K resistor. The Sensor therefore gives an active low signal when a 38KHZ modulated Infra red beam is received.

Most commercial remote controls, for TV, DVD or Refrigerators use 38KHz modulated signals.

You can use them as an input device to replace the keypad if you want to.

For example consider a SONY remote control, it has so many buttons, you can program your controller to decode for each button, so in other words you get a wireless keypad.



Push Switches

Two Push switches named SW1 and SW2 are provided to get momentary input. The switches are configured as active low.

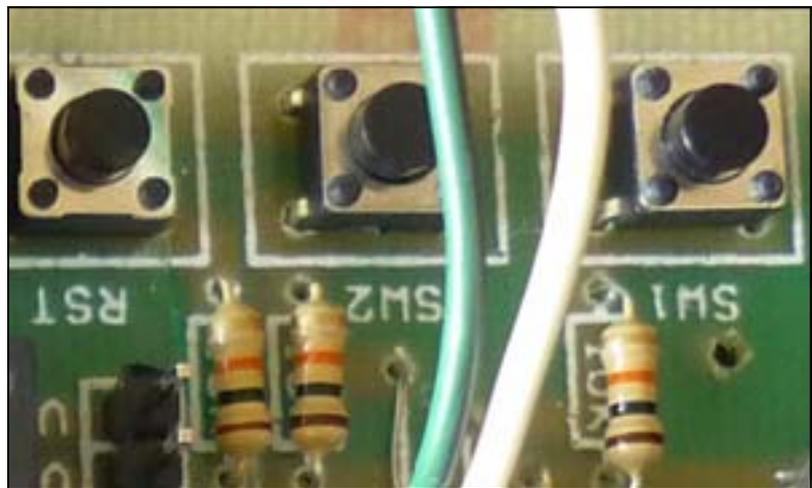
The output of each switch is connected through a 10K pull up resistor to +5V. The other end of switch is connected to GND.

Thus when switch is not pressed the Digital I-O of controller will see a logical High.

And when switch is pressed a logical Low will be seen at the pin.

A third switch RST, is provided.

This is reset switch. Pressing it will connect the MCLR pin to GND and reset the microcontroller.



I-O Lines

All I-O lines of the microcontroller are available through headers. Each pin is labeled by the corresponding Port Name. You must see the data-sheet to find out other functions like PWM, SPI etc associated with each pin.

The lines can be individually connected to any connections on board, or to external boards for interfacing. Fifteen I-O lines have associated +5V (VDD) and GND rails as well. These can be used to connect other boards, or to power them up.

Remember the board is powered by 7805 voltage regulator, that can supply only up to 1A current.



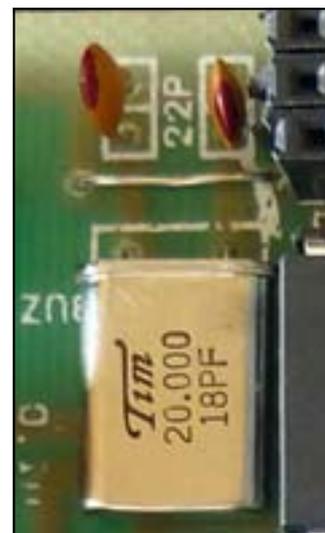
Crystal Oscillator

All microcontrollers require a clock signal for their proper operation. The speed of this oscillator will determine the speed of processing. This board is fitted with 20 MHz Crystal oscillator. This is the highest frequency at which most commonly used PIC microcontrollers can run.

PIC18F series controllers can indeed internally multiply this frequency and run the code at up to 48MHz.

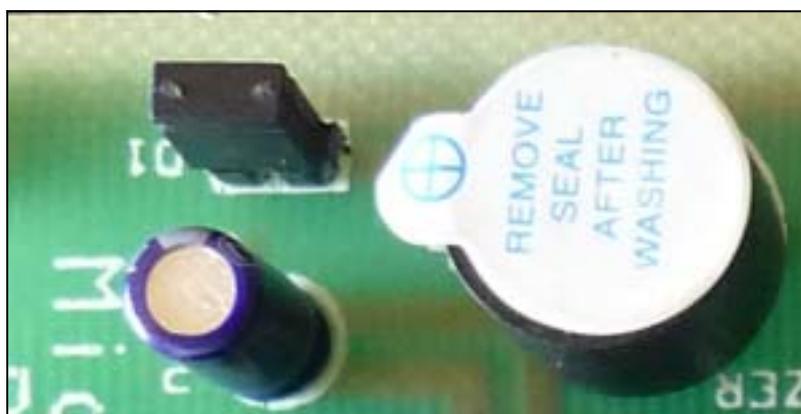
Slower processors like 16F877A although usually work well at this frequency, but when used with a debugger can not response that fast and thus debugging fails.

We therefore recommend using 18F series of microcontrollers with this board. In case you are using 16F series, you might need to change this crystal.



Buzzer

A 6V fixed frequency buzzer is provided. The buzzer has built-in oscillator to produce a sound of around 2KHz. The positive end of buzzer has been connected to +5V on board. The negative side is connected to Microcontroller RD1 through a jumper. In case you want to use RD1 for some other purpose, just disconnect this jumper, and you can connect buzzer to some other free I-O line.



The buzzer will produce sound when the corresponding I-O pin is taken Low.

In Circuit Programming Header

The usual practice is to program the microcontroller on a standalone programmer. This would require the controller to be removed and re-inserted back every time the new program is burnt into. This practice is obsolete now, and may even damage the controller.

Microchip now supports In Circuit Serial Programming (ICSP). However to do so your programmer must have this facility.

The header provided on this board is in accordance with Microchip guidelines. The arrangement of pins has been labeled. Your ICSP must have pins in this order:

- 1: VPP
- 2: VDD (+5V)
- 3: GND
- 4: PGD
- 5: PGC
- 6: NC (No Connection)



Power Supply

The board is supplied by a external 6-12V DC adapter. The center pin of DC adapter should give positive supply. A reverse polarity protection diode is placed in series to protect against reversed

polarity power supply.

The supply is then regulated by 7805 low drop-out voltage regulator.

An On-Off switch conveniently allows to cut-off supply instead of removing adapter pin again and again.

An LED indicator shows presence of 5V supply on board.

Note:

The In circuit programmer also gives out 5V supply, and when connected will light up the power LED. However this power may not be enough for the board and you will need to turn the board power ON as well.

Take special precaution when connecting the VDD supply rail to other boards. If other boards have their own supply, it is better only to connect GND. The VDD line from this rail is directly connected to VDD chain on board, so a wrongly connected line may damage your processor, or other ICs.



Using Bootloader

Microchip PIC 16F876 and above support a new method of programming, called self-programming. This feature allows a program to write into the program memory, without the need of an external programmer, and using PGD or PGC pins. Instead it uses the standard serial communication pins and serial communication system to get new program and write into program memory.

In order to use this feature you therefore need two things:

1. Your board must have serial communication system with PC. Our PIC-40 board has this feature so we can proceed.
2. You need a piece of special software that will reside inside the PIC to monitor if a new program is arriving down the serial link?

The small piece of program that needs to reside inside the PIC is called bootloader. Many commercial and non-commercial bootloader programs are available. Some are also supplied along with popular compilers, so that the program is easily compiled and transferred to the PIC from the same environment.

Here I shall demonstrate you how to install a bootloader supplied with Proton BASIC compiler. Every bootloader has two parts, one that resides in the PIC. This must be compiled for your particular controller and oscillator it has. The second part is kept in PC that communicates with the PIC part. In our case the PC part will be in side the PROTON BASIC IDE. So we do not need to install an other third party program to manage this.

Our PIC-40 board has PIC18F452 with 20 MHz crystal oscillator. So first thing is to get the Proton Basic bootloader for this. Open the installation folder of proton Basic and locate a folder named **MCloader**. Open this folder, this folder will have few files, that Proton IDE will use to communicate. This folder fill have another folder named **Loader-Hex**, this folder will have a number of pre-compiled .hex files for most



commonly used processors that support self programming. Each processor will have files for 4 and 20 MHz oscillators. Now Locate **18F452_20.hex** file.

The first step is to burn this file into your microcontroller using a standard programmer. You can use any programmer to do that. We used our PIC PG-II programmer to transfer this file into PI18F452 microcontroller through In circuit serial programmer header.

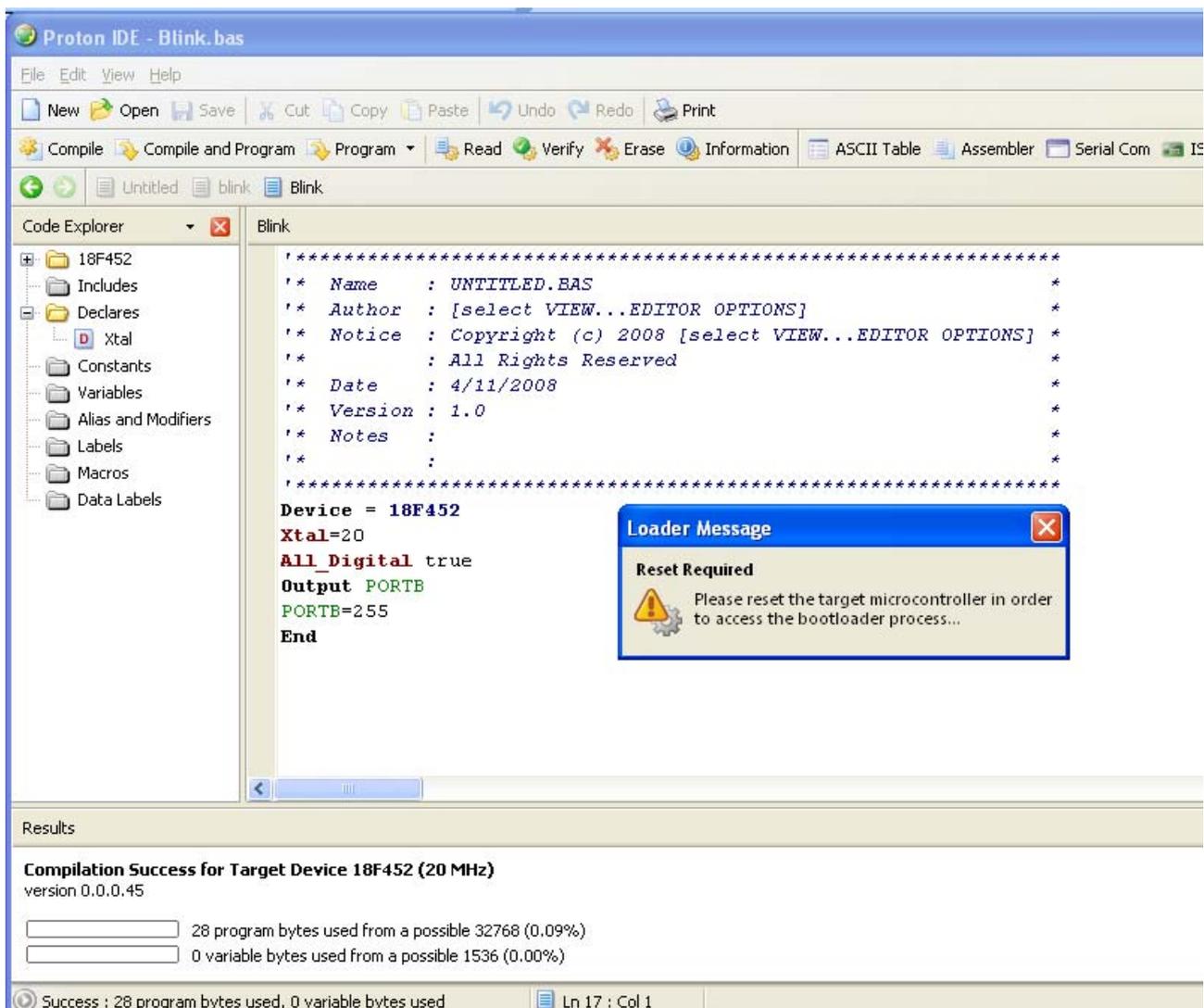
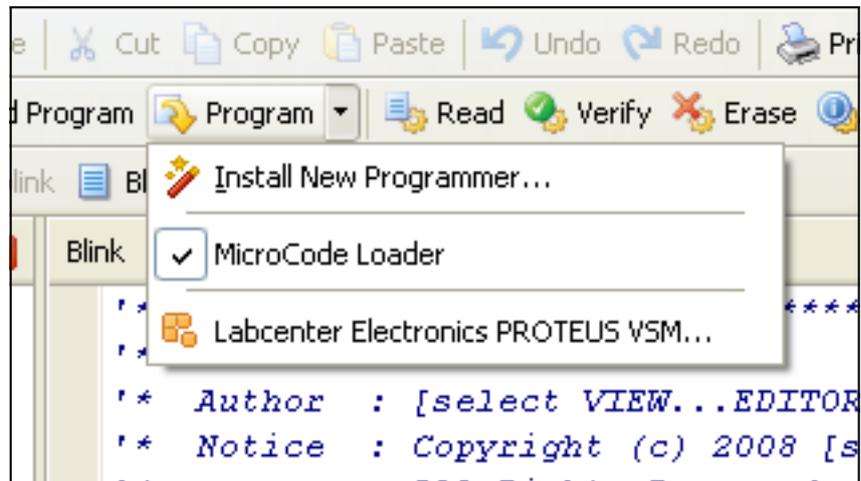
Once the file has been successfully transferred, disconnect the programmer. Attach serial cable, and connect the serial module with RC6 and RC7. As shown in photo.

Now start Proton Basic IDE and select Microcode loader as programmer.

This will select the bootloader to be used as programmer.

Once this has been selected,

write your Proton Basic program, and click the *Compile and Program* button on toolbar. The compiler will compile the Basic language program to produce .hex file. After that it will show up Microcode Loader message to Reset the controller. Now either turn the board On, if its Off, or just



press the reset button. If everything is OK, the progress Bar should appear and transfer the new program in to microcontroller, and immediately start executing it.

If you modify the source program, just click the Compile and program button, and press the rest button when asked, the new program will be updated into PIC microcontroller and start execution. That's it.

In case the PIC fails to communicate open the MCloder in Proton Basic and set the COM port and baud rate accordingly.

In case your particular compiler does not have a bootloader, then you can use a standalone boot-loader program, there are many available on internet, but I personally like Tiny Bootloader.

<http://www.etc.ugal.ro/cchiculita/software/picbootloader.htm>