USB Logic Tool

Contents

The USB Logic Tool

The device is meant to be a prototyping aid and designed to work well by minimizing breadboard space used, interaction through a visual interface, easy setup and low cost. The low cost allows it to be built at home or a PCB can be purchased for a low price. It is also reprogrammable through USB so other functions can easily be loaded (such as a Virtual COM Port) or a custom one (variable switching mode power supply).

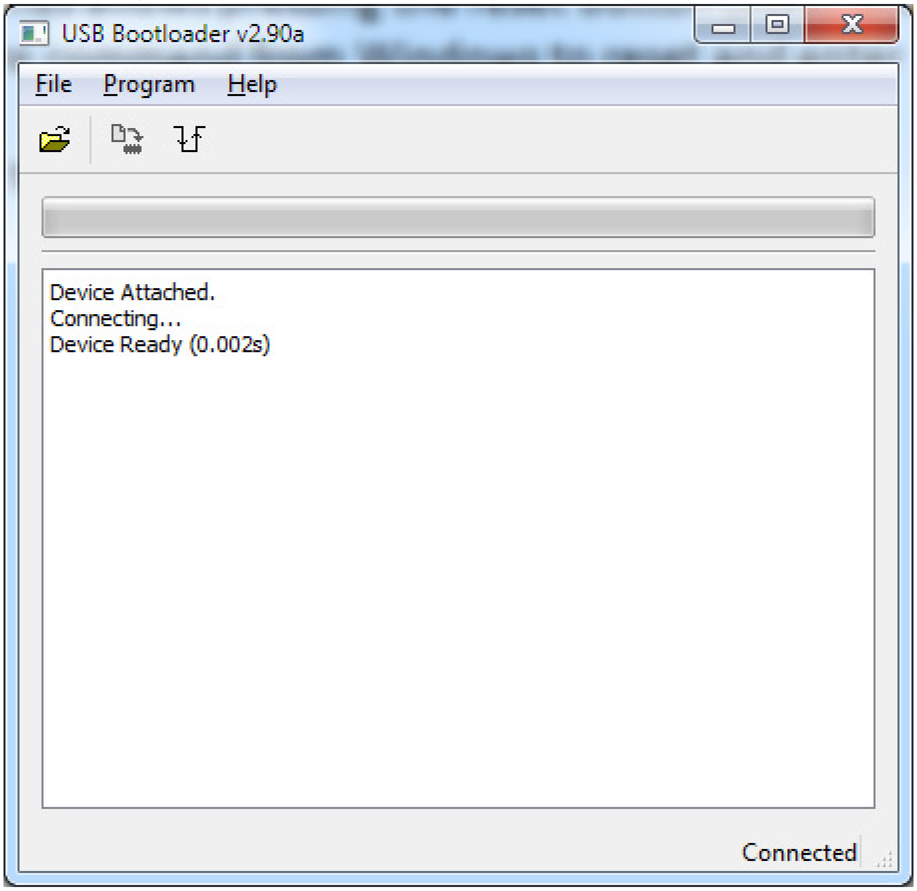
Programming

The PIC18F14K50 on the device can be re-programmed as needed and the programs can be the "USB Logic Tool", "Virtual COM Port" or a custom program developed by the user. With "USB Logic Tool" the pins function as set in the visual interface (more information in "Using the USB Logic Tool"). The "Virtual COM Port" allows UART communication through USB and pin RB5 will act as a receiver and RB7 will act as the transmitter. The "Virtual COM Port" drivers will need to be installed with the drivers provided by Microchip (included in the project files as well). To get started with custom programming visit: http://www.libstock.com/projects/view/55/microchip-hid-bootloader-template-and-adc-over-usb-for-pic18f-and-pic24f-pic18f14k50-pic18f27j53-pic18f2553-pic18f4553-pic24fj64gb002-tested

A standard programmer (such as PICkit 2) is needed to program the bootloader HEX file into PIC18F14K50. Once that is complete the HID bootloader is used to download any real program HEX file (the default one will be for the USB Logic Tool) and done as follows:

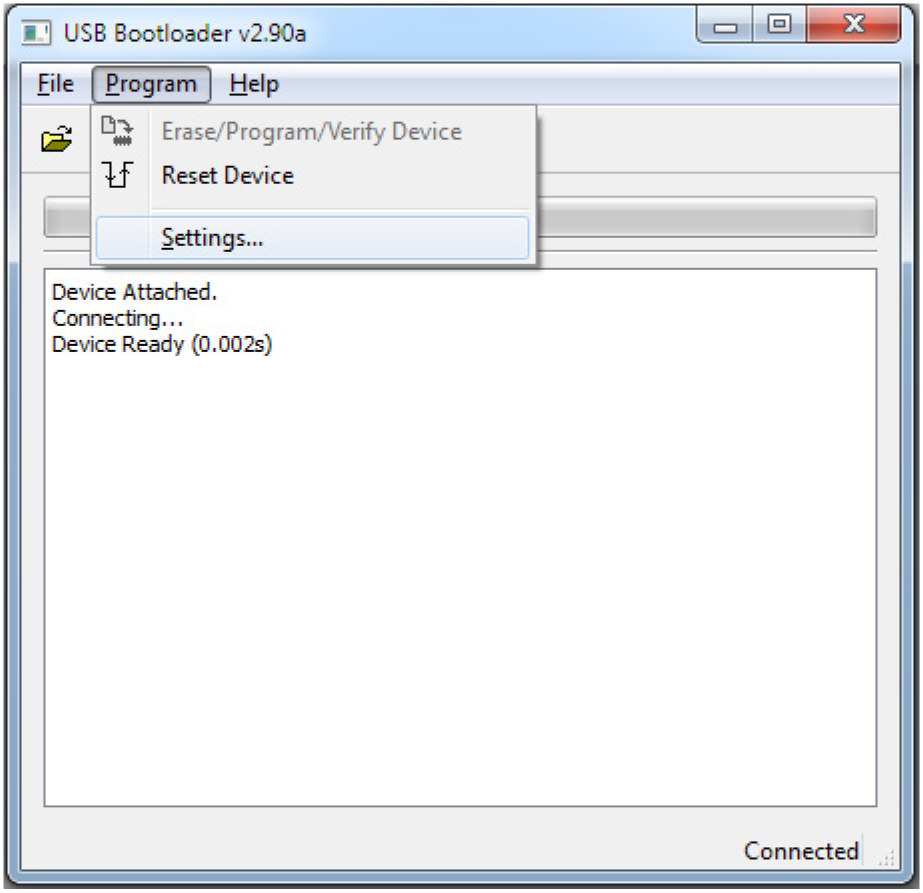
1) Enter bootloader mode:
    a. If the PIC is empty then it will automatically go into bootloader mode.
    b. If the program is loaded then holding the reset button (SW2) at power will enter bootloader mode.
    c. The USB Logic Tool program has allows pressing the reset button (SW2) while running or sending a software command from Windows to reset and enter bootloader mode.
2) Use Microchip's Bootloader software to download the program:

a. The display if the device is successfully connected in bootloader mode.



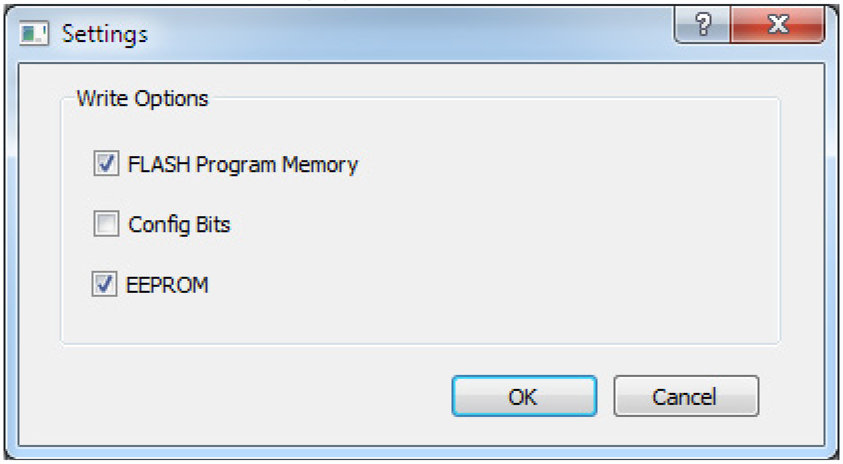
```
Device Attached.
Connecting…
Device Ready (0.002s)
```

b. Double check the settings by going to “Program->Settings…”.



```
Device Attached.
Connecting…
Device Ready (0.002s)
```

c. Make sure the following options are set.

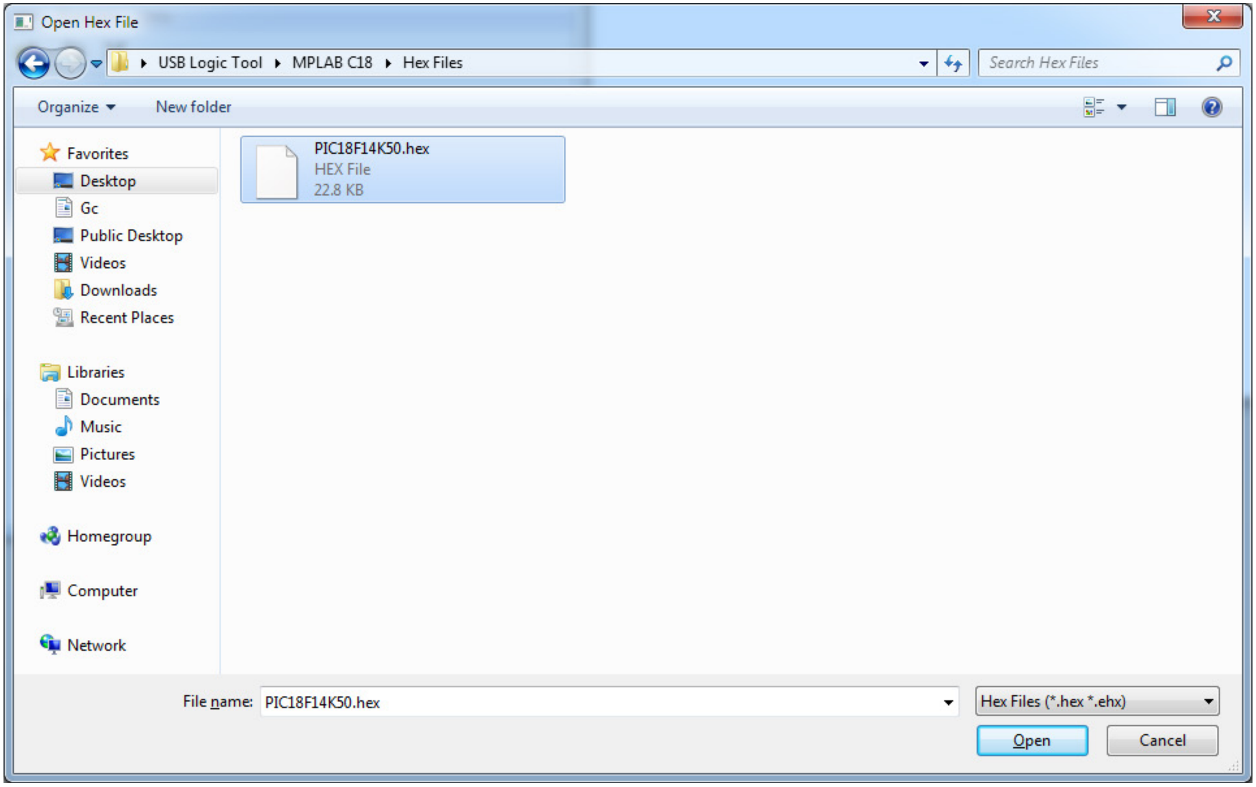


☑ FLASH Program Memory

□ Config Bits

☑ EEPROM

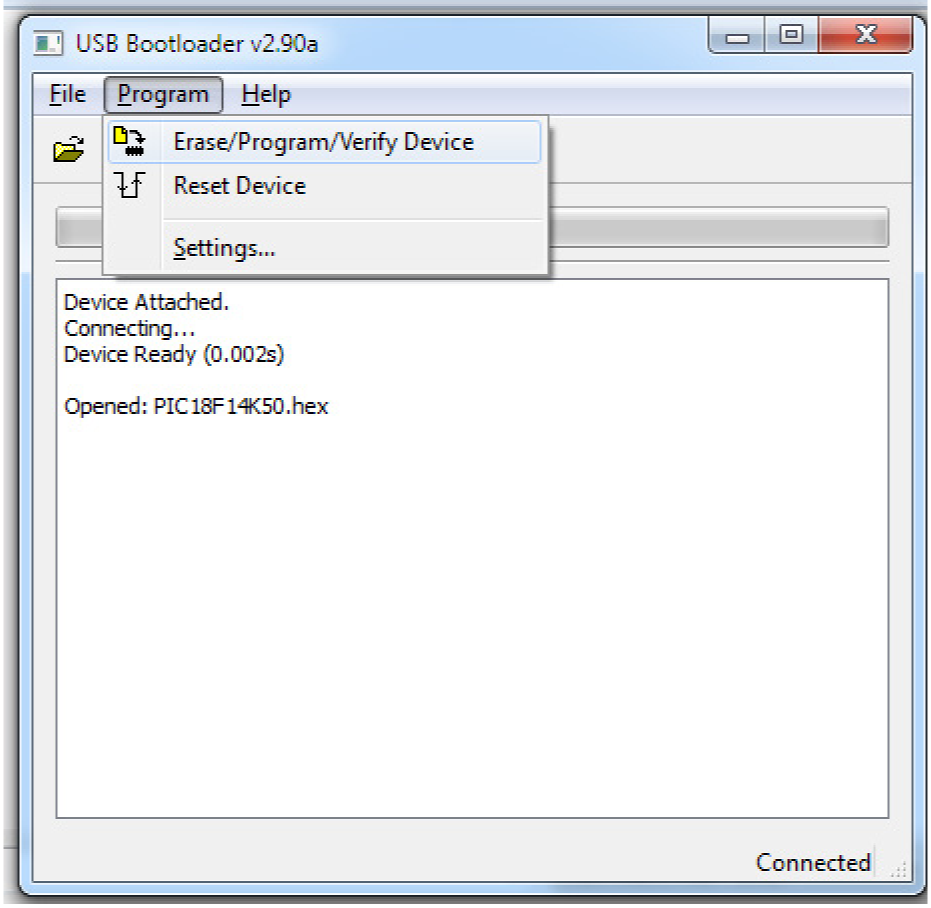
d. Press “CTRL+O” then select the HEX file to program.



File name: PIC18F14K50.hex

Hex Files (*.hex *.ehx)

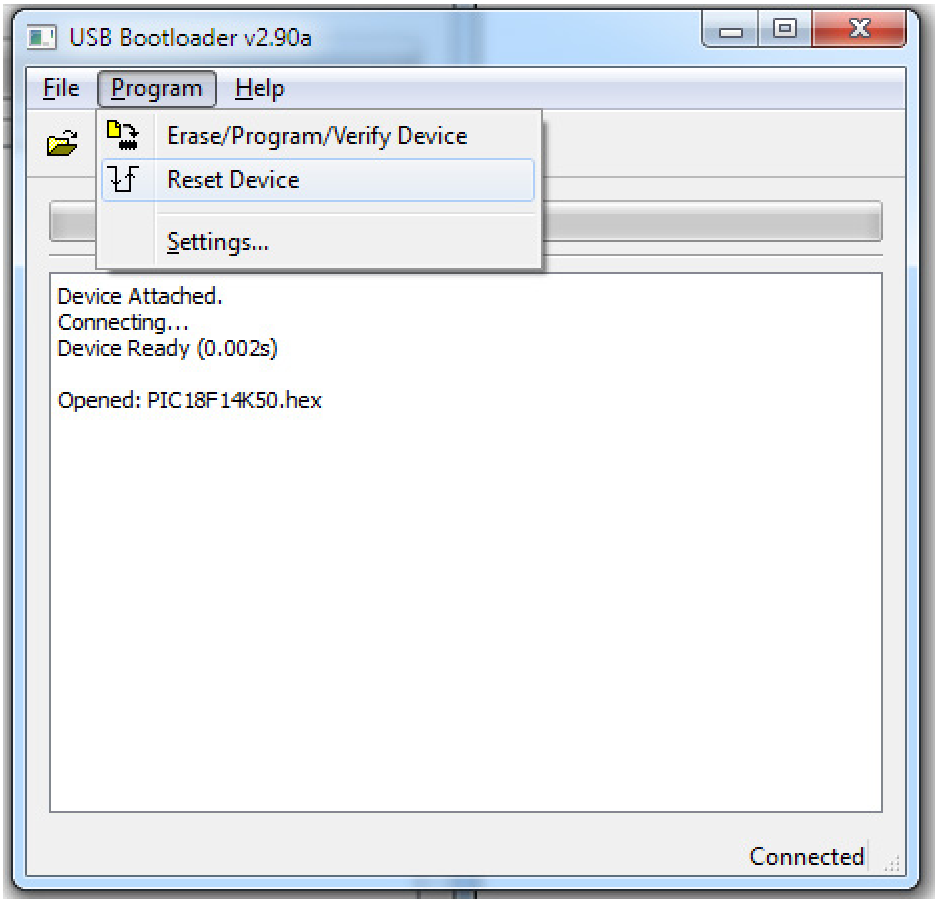
e. Start programming the device by going to “Program->Erase/Program/Verify Device”.



```
Device Attached.
Connecting…
Device Ready (0.002s)

Opened: PIC18F14K50.hex
```

f. Reset the device by going to “Program->Reset Device”.

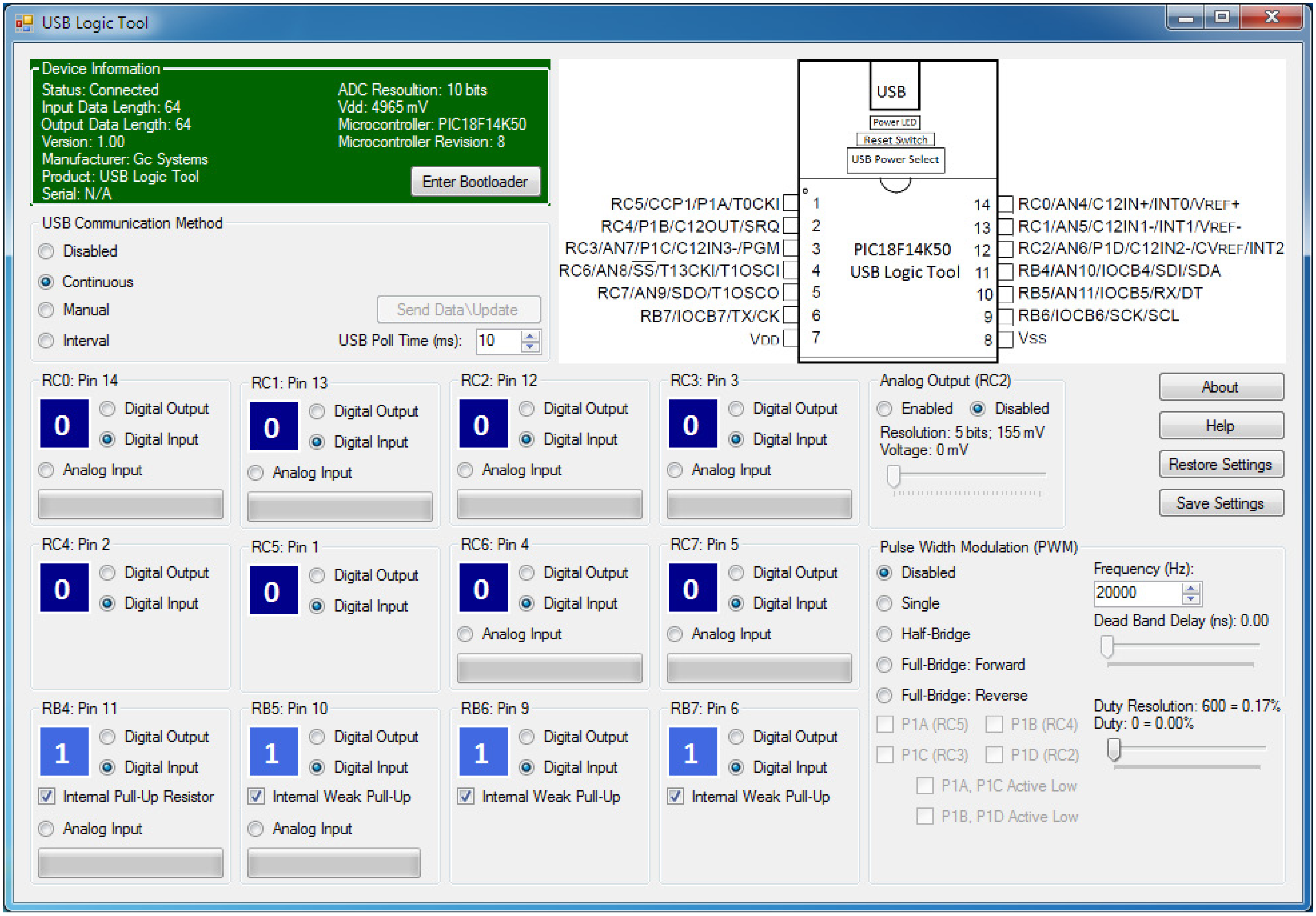


```
Device Attached.
Connecting…
Device Ready (0.002s)

Opened: PIC18F14K50.hex
```

Using the USB Logic Tool

1) Start USB Logic Tool software and connect the USB Logic Tool device (order does not matter).
   a. Windows will automatically install drivers if they do not exist.
2) This is what the software will look like when the device is connected:



USB Logic Tool

Device Information
Status: Connected
Input Data Length: 64
Output Data Length: 64
Version: 1.00
Manufacturer: Gc Systems
Product: USB Logic Tool
Serial: N/A
ADC Resoultion: 10 bits
Vdd: 4965 mV
Microcontroller: PIC18F14K50
Microcontroller Revision: 8
Enter Bootloader

USB Communication Method
○ Disabled
● Continuous
○ Manual
○ Interval
Send Data\Update
USB Poll Time (ms): 10

RC0: Pin 14
0
○ Digital Output
● Digital Input
○ Analog Input

RC1: Pin 13
0
○ Digital Output
● Digital Input
○ Analog Input

RC2: Pin 12
0
○ Digital Output
● Digital Input
○ Analog Input

RC3: Pin 3
0
○ Digital Output
● Digital Input
○ Analog Input

Analog Output (RC2)
○ Enabled ● Disabled
Resolution: 5 bits; 155 mV
Voltage: 0 mV

About
Help
Restore Settings
Save Settings

RC4: Pin 2
0
○ Digital Output
● Digital Input

RC5: Pin 1
0
○ Digital Output
● Digital Input

RC6: Pin 4
0
○ Digital Output
● Digital Input
○ Analog Input

RC7: Pin 5
0
○ Digital Output
● Digital Input
○ Analog Input

Pulse Width Modulation (PWM)
● Disabled
○ Single
○ Half-Bridge
○ Full-Bridge: Forward
○ Full-Bridge: Reverse
□ P1A (RC5) □ P1B (RC4)
□ P1C (RC3) □ P1D (RC2)
□ P1A, P1C Active Low
□ P1B, P1D Active Low
Frequency (Hz): 20000
Dead Band Delay (ns): 0.00
Duty Resolution: 600 = 0.17%
Duty: 0 = 0.00%

RB4: Pin 11
1
○ Digital Output
● Digital Input
☑ Internal Pull-Up Resistor
○ Analog Input

RB5: Pin 10
1
○ Digital Output
● Digital Input
☑ Internal Weak Pull-Up
○ Analog Input

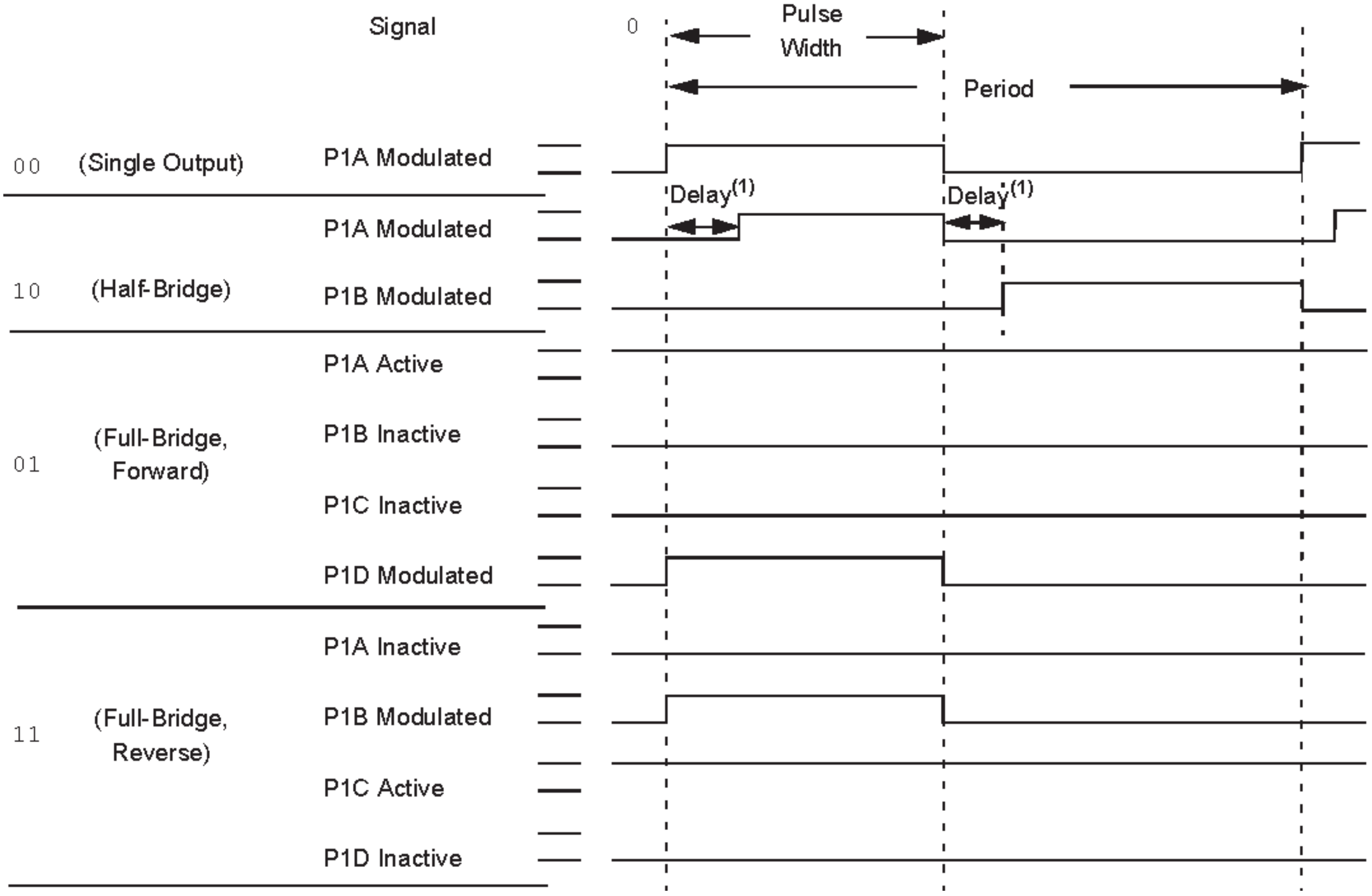
RB6: Pin 9
1
○ Digital Output
● Digital Input
☑ Internal Weak Pull-Up

RB7: Pin 6
1
○ Digital Output
● Digital Input
☑ Internal Weak Pull-Up

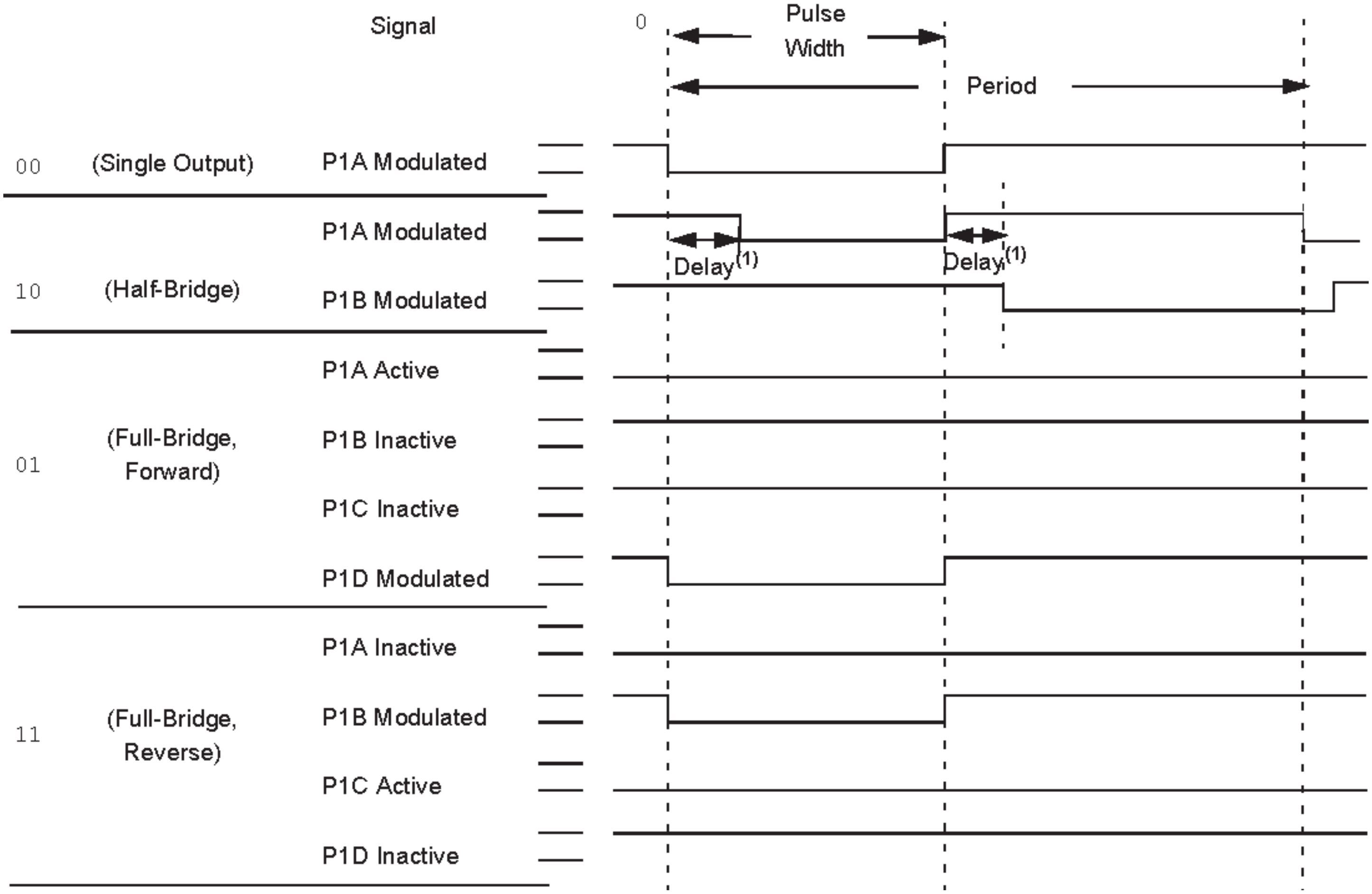
3) The device features are changed by pressing the large numbers (in digital output mode), selecting the radio buttons, checking the checkboxes, sliding the slide bars or pressing the standard buttons.
4) Device Information: Green when connected, otherwise it is red. Shows the ADC resolutions of the device, the current voltage powering the PIC, the specific microcontroller and its revision.
5) USB Communication Method:
   a. *Disabled:* There is no communication between the PIC and PC but all settings (before disconnect) will remain.
   b. *Continuous (Default):* PIC and PC communicate as fast as possible.
   c. *Manual:* PIC and PC communicate when “Send Data\Update” is pressed.
   d. *Interval:* PIC and PC at a specified interval (in milliseconds) that is set in “USB Poll Time (ms)”.
6) About*:* Shows information about the program.
7) Help: Shows a pop-up that explains what all of the features do.

USB Logic Tool Features

- Pulse Width Modulation (PWM): Takes priority over Analog Output and digital/analog pin functions if selected
  - *Frequency:* Sets the frequency of the square wave which can be between 2933Hz and 1MHz
  - *Duty:* Sets the percentage of time that square wave is active (can be high or low)
  - Disabled: The PWM module is not active and none of the pins are reserved
  - Single Mode: A single PWM signal which can be copied to four different pins
  - Half-Bridge Mode: P1A and P1B are used and P1C and P1D are not used (logic functions are allowed) to create a wave form opposing (inverted) each other
  - Full-Bridge: Forward Mode: P1A is active, P1B and P1C are inactive, and P1D is modulated (square wave)
  - Full-Bridge: Reverse Mode: P1A is inactive, P1B is modulated (square wave), P1C is active, and P1D inactive
  - P1A, P1B, P1C, P1D Checkboxes: Can be checked to output square wave on those pins in single mode only; They are disabled and appropriately checked in all other modes
  - Active Low Checkboxes: If selected the signal on the pins will be inverted
  - Dead Band: Creates a delay in switching the signal in Half-Bridge of Full Bridge Mode
  - An example of output wave in the active-high setting (the delay is due to the dead-band):

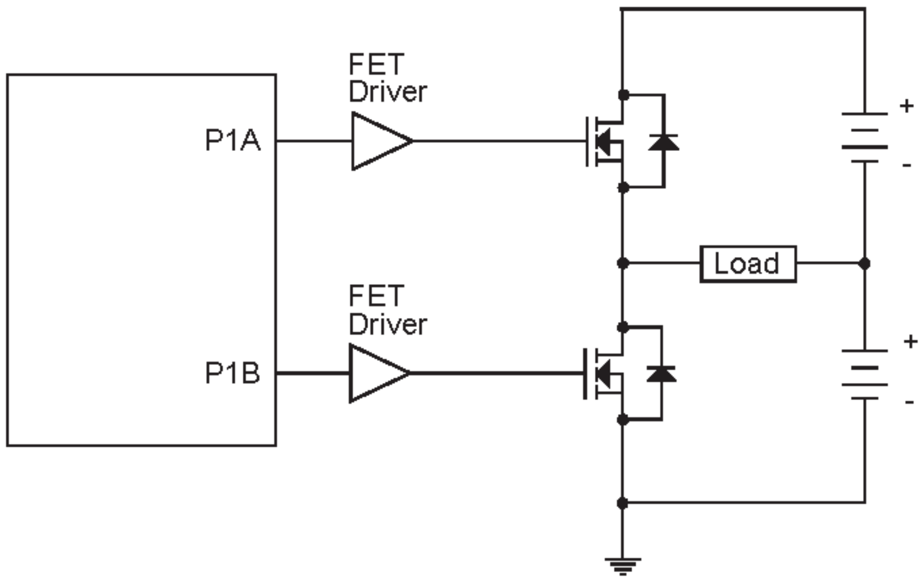


- An example of output wave in the active-low setting (the delay is due to the dead-band):

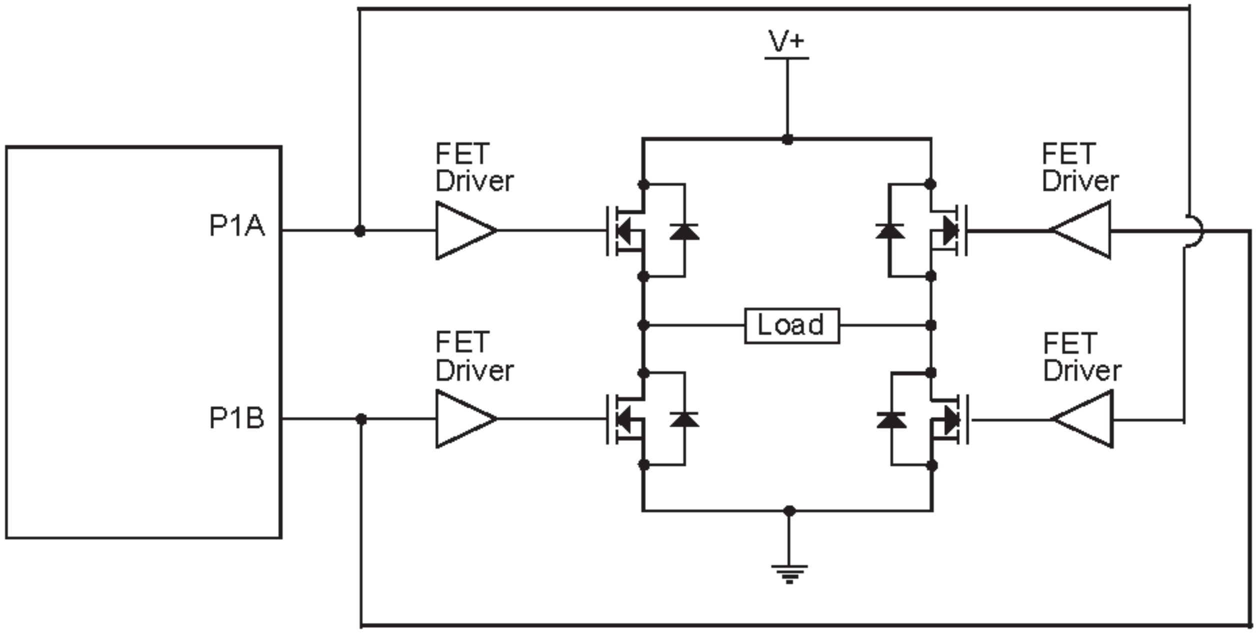


- Half-Bridge Sample Circuits:

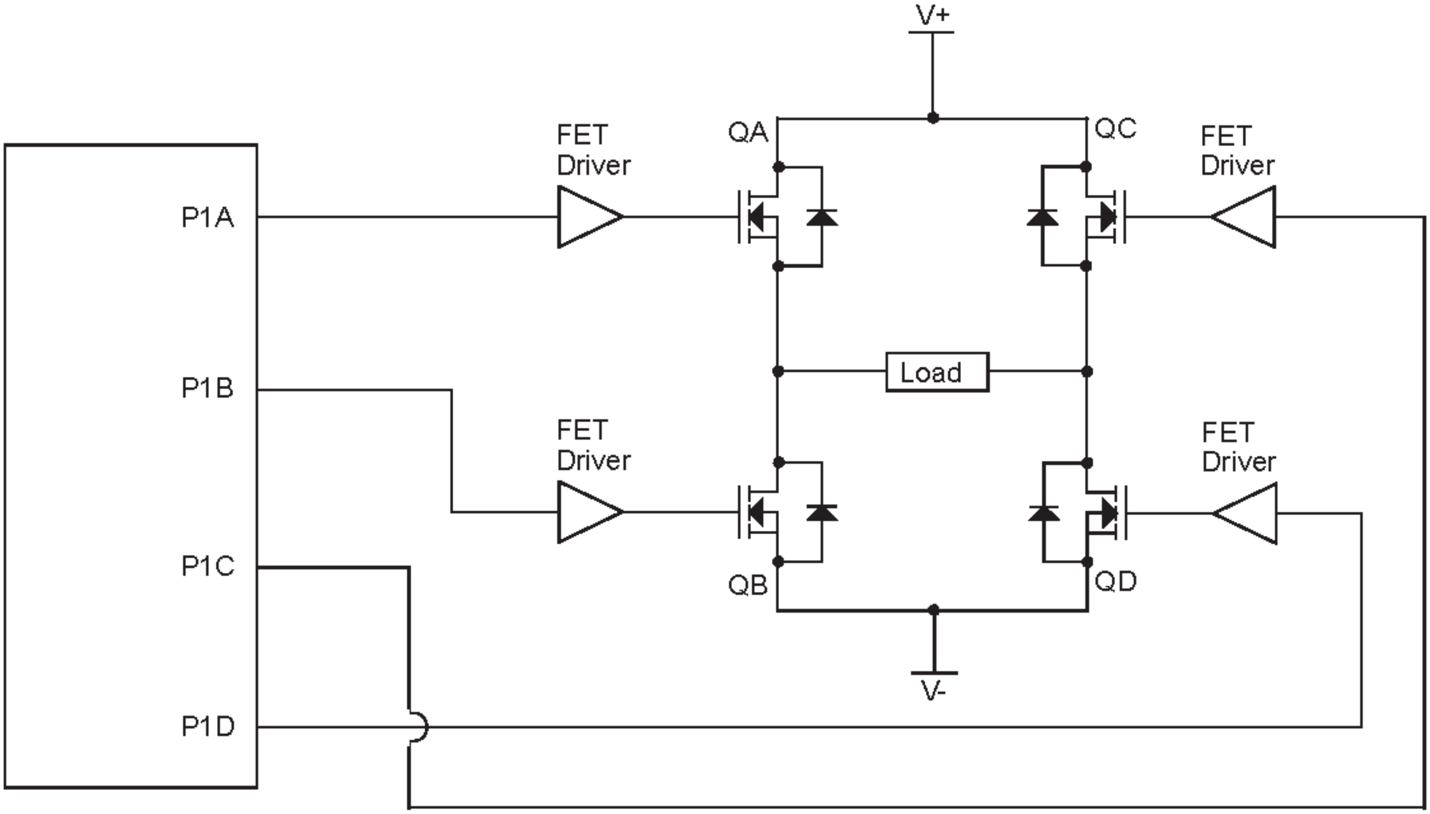
Standard Half-Bridge Circuit (“Push-Pull”)



Half-Bridge Output Driving a Full-Bridge Circuit



- Full-Bridge Sample Circuits:

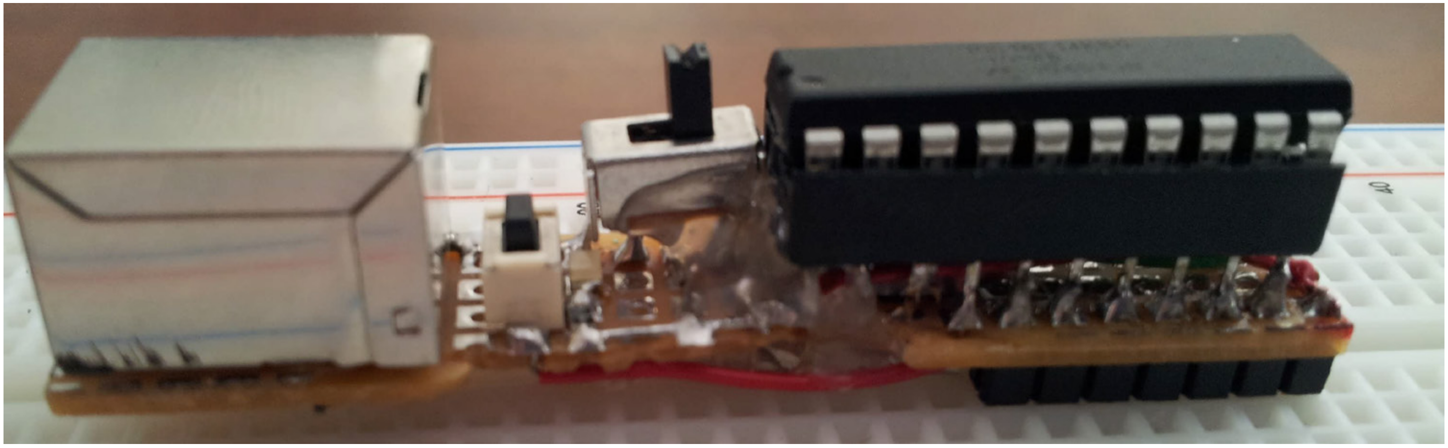


- Analog Output: Set a 5-bit resistor ladder circuit to give an analog output voltage; Take priority over pin RC2 functions
- Save Settings: Save all settings to the PICs internal EEPROM; If EEPROM settings are ever saved then they will be restored at power-up
- Restore Settings: Load settings saved in the PICs internal EEPROM
- Analog Output: Set pin to read an analog reading on the specified pin
- Digital Input: Set pin to read a digital input where 1 represents logic high and 0 represents logic low
  - *Internal Weak Pull-up:* Enable the internal pull-up resistor on the pin
- Digital Output: Set logic high (Vdd) or logic low (Vss) output by clicking on the large number to alternate the state

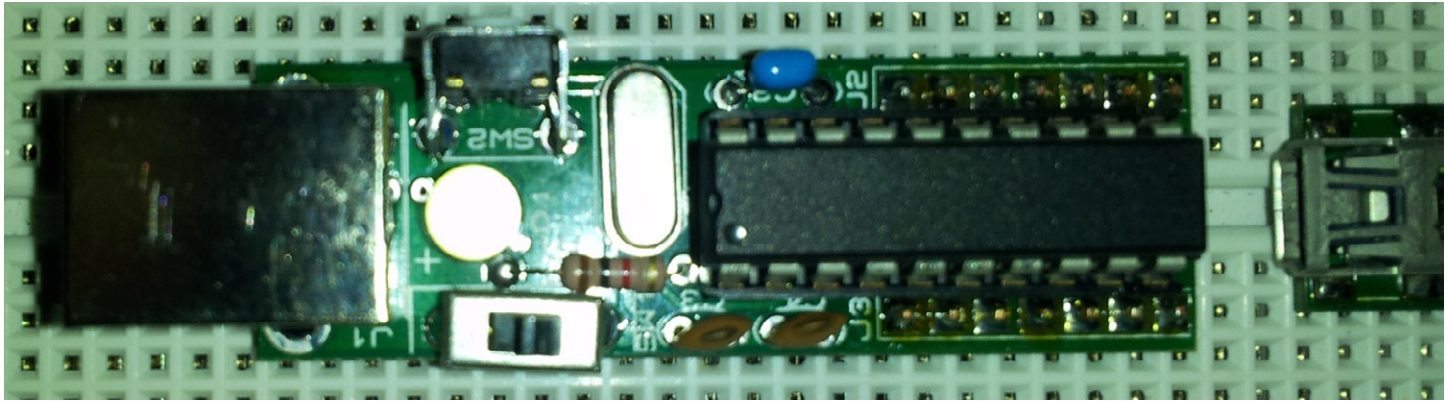
Device Hardware

A few different options are available to use the device and the schematic (Schematic.PDF) and bill of materials (Bill Of Materials.XLSX) are included in the project:

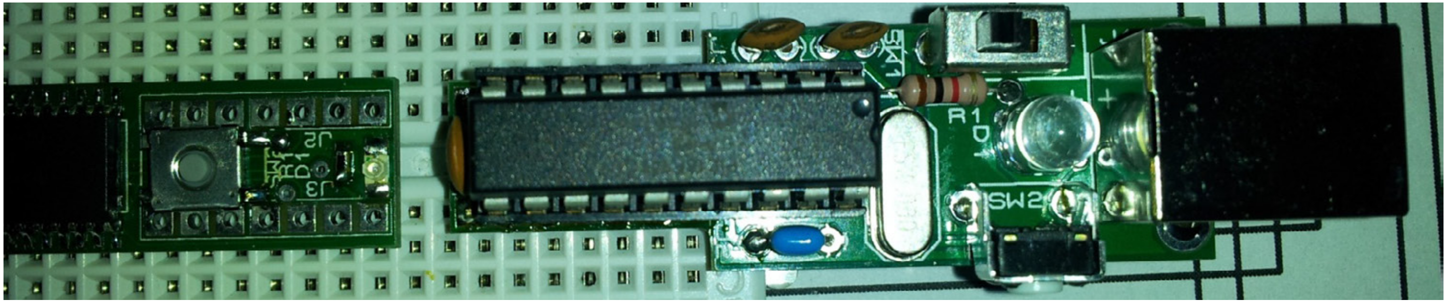
- Build on BreadBoard - Easy
- Build on ProtoBoard – Difficult/Tricky (MCU Pins are directly soldered to breadboard headers and crystal is under the IC)

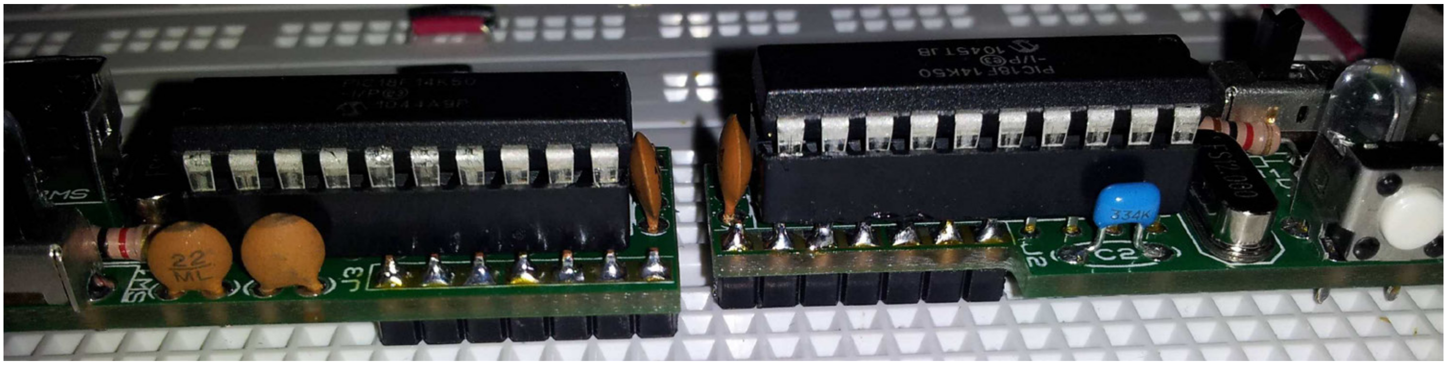


- Buy Through Hole Version PCB from BatchPCB – Easy
    - “PCB Layout (Through Hole).pdf”
    - http://www.batchpcb.com/index.php/Products/83920

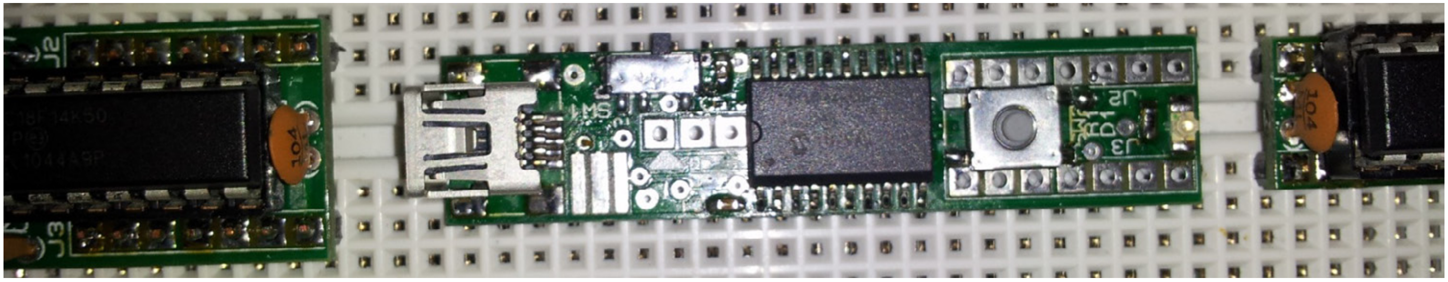


- Buy Through Hole Version PCB with space reduced by mounting IC on breadboard headers from BatchPCB – Average/Tricky (MCU Pins are directly soldered to breadboard headers)
    - “PCB Layout (Through Hole - Special).pdf”
    - http://www.batchpcb.com/index.php/Products/83922





- Buy Surface Mount Version PCB from BatchPCB – Difficult The First Time
    - “PCB Layout (Surface Mount).pdf”
    - http://www.batchpcb.com/index.php/Products/82975



Buying from BatchPCB will only provide the PCB needed and the components have to be purchased separately. The exact components for the PCB are listed in “Bill Of Materials.XLSX”.