

ME Labs Advanced D-Stick (PIC18F47K40)

Chapter 1: Software

1.1 Download meProg Programming Software

The meProg software from ME Labs is used to open a Microchip-compatible (HEX) file and program the D-Stick with the code contain therein. The installation also includes the driver for MELABS COM Port, which allows serial communications with the D-Stick through USB.

Software, driver files and program examples can be found at:

<http://melabs.com/dstick>

The hex file mentioned above may be created with a variety of software tools. Any hex file suitable for programming the PIC18F47K40 may be used with the D-Stick. We recommend the PICBASIC PRO™ Compiler (PBP), discussed below. You will need PBP Gold Edition to support the Advanced D-Stick.

PICBASIC PRO™ Compiler activation keys may be purchased at:

<http://store.melabs.com/cat/PBP.html>

When creating the hex file, it is customary to include Configuration Data that sets the initial parameters for the PIC18F47K40. It is important to note that the D-Stick uses Low-Voltage-Programming to program the PIC18F47K40. Here is an example of the #CONFIG directive that you might use in PBP:

```
#CONFIG
CONFIG FEXTOSC = OFF ;Oscillator not enabled
CONFIG RSTOSC = HFINTOSC_64MHZ ;HFINTOSC with HFFRQ = 64 Mhz and CDIV = 1:1
CONFIG CLKOUTEN = OFF ;CLKOUT function is disabled
CONFIG CSWEN = ON ;Writing to NOSC and NDIV is allowed
CONFIG FCMEN = ON ;Fail-Safe Clock Monitor enabled
CONFIG MCLRE = EXTMCCLR ;If LVP = 0, MCLR pin is MCLR; If LVP = 1, RE3 pin function is MCLR
CONFIG PWRTE = ON ;Power up timer enabled
CONFIG LPBOREN = OFF ;ULPBOR disabled
CONFIG BOREN = ON ;Brown-out Reset enabled according to SBOREN
CONFIG BORV = VBOR_245 ;Brown-out Reset Voltage (VBOR) set to 2.45V
CONFIG ZCD = OFF ;ZCD disabled. ZCD can be enabled by setting the ZCDSEN bit of ZCDCON
CONFIG PPS1WAY = OFF ;PPSLOCK bit can be set and cleared repeatedly (subject to the unlock sequence)
CONFIG STVREN = ON ;Stack full/underflow will cause Reset
CONFIG DEBUG = OFF ;Background debugger disabled
CONFIG XINST = OFF ;Extended Instruction Set and Indexed Addressing Mode disabled
CONFIG WDTCPSS = WDTCPSS_31 ;Divider ratio 1:65536; software control of WDTPS
CONFIG WDTE = ON ;WDT enabled regardless of sleep
CONFIG WDTCS = WDTCS_7 ;window always open (100%); software control; keyed access not required
CONFIG WDTCCS = LFINTOSC ;WDT reference clock is the 31.0 kHz LFINTOSC
CONFIG WRT0 = OFF ;Block 0 (000800-003FFFh) not write-protected
CONFIG WRT1 = OFF ;Block 1 (004000-007FFFh) not write-protected
CONFIG WRT2 = OFF ;Block 2 (008000-00BFFFh) not write-protected
CONFIG WRT3 = OFF ;Block 3 (00C000-00FFFFh) not write-protected
CONFIG WRT4 = OFF ;Block 4 (010000-013FFFh) not write-protected
CONFIG WRT5 = OFF ;Block 5 (014000-017FFFh) not write-protected
CONFIG WRT6 = OFF ;Block 6 (018000-01BFFFh) not write-protected
CONFIG WRT7 = OFF ;Block 7 (01C000-01FFFFh) not write-protected
CONFIG WRTC = OFF ;Configuration registers (300000-30000Bh) not write-protected
CONFIG WRTB = OFF ;Boot Block (000000-0007FFFh) not write-protected
CONFIG WRTD = OFF ;Data EEPROM not write-protected
CONFIG SCANE = ON ;Scanner module is available for use, SCANMD bit can control the module
CONFIG LVP = ON ;Low voltage programming enabled. MCLR/VPP pin function is MCLR. MCLRE configuration bit is ignored
CONFIG CP = OFF ;UserNVM code protection disabled
CONFIG CPD = OFF ;DataNVM code protection disabled
CONFIG EBTR0 = OFF ;Block 0 (000800-003FFFh) not protected from table reads executed in other blocks
CONFIG EBTR1 = OFF ;Block 1 (004000-007FFFh) not protected from table reads executed in other blocks
CONFIG EBTR2 = OFF ;Block 2 (008000-00BFFFh) not protected from table reads executed in other blocks
CONFIG EBTR3 = OFF ;Block 3 (00C000-00FFFFh) not protected from table reads executed in other blocks
CONFIG EBTR4 = OFF ;Block 4 (010000-013FFFh) not protected from table reads executed in other blocks
CONFIG EBTR5 = OFF ;Block 5 (014000-017FFFh) not protected from table reads executed in other blocks
CONFIG EBTR6 = OFF ;Block 6 (018000-01BFFFh) not protected from table reads executed in other blocks
CONFIG EBTR7 = OFF ;Block 7 (01C000-01FFFFh) not protected from table reads executed in other blocks
CONFIG EBTRB = OFF ;Boot Block (000000-0007FFFh) not protected from table reads executed in other blocks
#ENDCONFIG
```

1.2 Using the D-Stick with PBP 3.1 Gold Edition

The instructions here are for PBP 3.1 and later (Gold Edition required). If you already use a version of PBP that supports the PIC18F47K40, you can use it to program the D-Stick.

When using PBP with the D-Stick, the default development environment is MicroCode Studio. This is the code editor that is included with PBP. Select “18F47K40” in the Microcontroller dropdown list on the MicroCode Studio toolbar. MicroCode Studio will automatically find the meProg software when it is installed on the computer. Note that older versions of meProg do not support the D-Stick. You should have meProg version 4.63 or later. The download link appears above.

The programmer selection in MicroCode Studio should be set to “microEngineering Labs Programmer”. This setting is found in the View menu under “Compile and Program Options”. MicroCode Studio will automatically launch meProg when you click the “Compile Program” button.

1.3 Using meProg



If launched from MicroCode Studio, the newly-created hex file will already be open. If meProg is launched from its Windows shortcut, click File > Open and select the hex file that you want to program into the D-Stick. When the file is open, its name will show in the title bar in meProg. To program the D-Stick, click Program > Program or press Ctrl-P on the keyboard. A full description of meProg and how to operate it can be found in its Help menu.

1.4 The MELABS COM Port

When meProg is installed, it will also install a COM Port Driver for the D-Stick. This driver also serves for other ME Labs devices. When the D-Stick is connected to USB, Windows will automatically create the COM Port and assign a number to it.

If you plan to use the COM Port in MicroCode Studio, make sure the D-Stick is connected before you launch the software. If you connect the D-Stick while MicroCode Studio is running, you will need to shut it down and restart it to connect to the COM Port.

If you have multiple COM Ports on your computer and need to identify which one is the D-Stick, you can run the Windows Device Manager from Control Panel and look under “Ports (COM & LPT)”. When the D-Stick is connected, you should see the MELABS COM Port displayed with its number.

The MELABS COM Port accesses the USART on the PIC18F47K40. It can be used with MicroCode Studio Plus for in-circuit debugging. It can also be used to communicate with other software on the computer. You may use terminal-emulation software or your own custom applications to interact with the program running on the D-Stick.

1.5 MCSPX In Circuit Debugging (ICD)

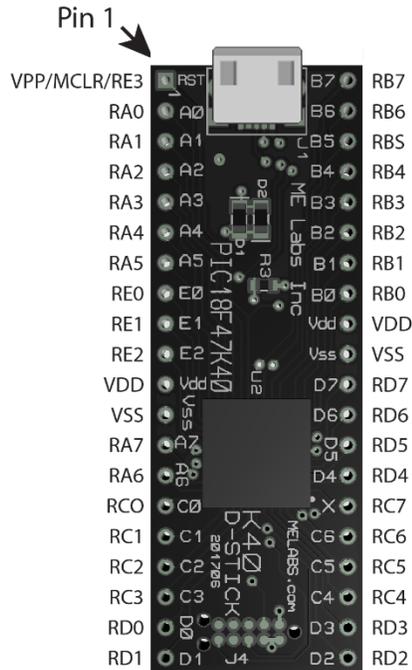
To use the ICD function in MicroCode Studio, you must purchase the PLUS version of the software (MicroCode Studio Plus, aka MCSPX). If the ICD toolbar buttons are disabled when you select the 18F47K40, you will need to copy the ICD model file into the installation folder. A newly-created model file (18F47K40.icd) is included with the PBP program examples that are found on the D-Stick support page. After you download and extract the examples, copy the file and save to the “ICDModels” folder within the MCSPX installation.

(default location: C:\Program Files (x86)\Mecanique\MCSPX\ICDModels)

2.1 Pin Locations

The ME Labs D-Stick functions exactly like the PIC microcontroller that it is based on (PIC18F47K40), with the addition of an on-board, in-circuit programmer and a USB-to-serial converter that allows communication to the microcontroller's USART. The pin locations on the D-Stick match the pin locations on the 40-pin DIP package of Microchip's part number: PIC18F47K40-I/P. This allows you to replace existing microcontrollers with the D-Stick to add programming and serial communications functionality. It also allows you to replace the D-Stick with a less expensive device after development is finished.

Pin diagram:



See the Microchip PIC18F47K40 data sheet for complete descriptions of the pins and their functions.

2.2 Shared connections

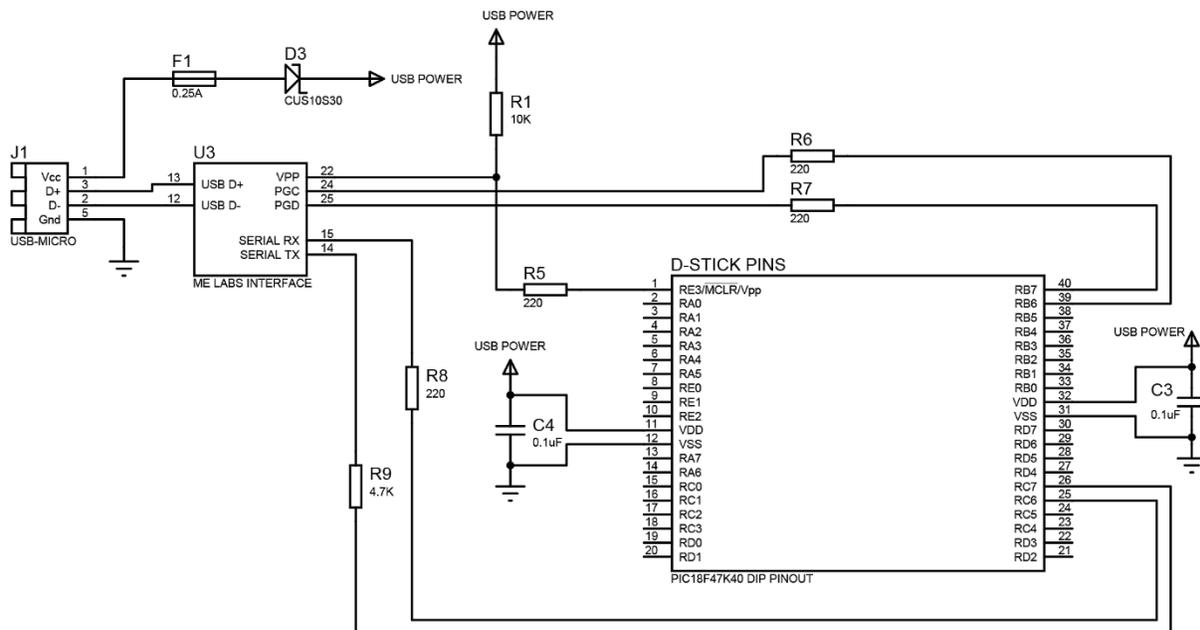
On the simplified schematic diagram below, you can see that most of the pins on the D-Stick are isolated connections to the equivalent pins on the PIC18F47K40. There are exceptions, though, to accommodate programming and serial communications.

Of special concern is the USART RX, pin-26, labeled “RC7”. On the PIC18F47K40, the RX pin can be used as a high-impedance input. On the D-Stick, the RX pin is driven by the USB-to-serial converter on the ME Labs interface chip through a resistor. This may interfere with serial (or other input signals) that are routed to pin-26 of the D-Stick.

Care should also be taken when connecting to the programming pins (RB6 and RB7). If you load these pins heavily or with capacitive/inductive elements, you may receive errors when programming the D-Stick. It is best to avoid connecting to RB6 and RB7, but they can be used for high-impedance connections.

All the interface connections to the PIC18F47K40 are made with series resistors to avoid damage to the interface chip. The resistors, shown on the schematic below, may be de-soldered and removed if desired. They are located on the underside of the D-Stick PCB and clearly labeled.

2.3 Simplified Schematic



2.4 Power

Note the connection to J1, Pin-1 on the schematic above. This is how the D-Stick is powered by the USB connection. It is protected by a diode so that the computer’s USB ports won’t be damaged if you connect a second power source to the VDD pins on the D-Stick.

Since the USB connection supplies power from the PC, you can use this to power the circuit in which the D-Stick is inserted. When the USB cable is connected, the 5V power is connected to the VDD pins shown above (pins 11 and 32). The available current is 250mA. If the built-in fuse is triggered by excessive current, it will reset automatically if power is disconnected for a few seconds.

