PicKit2™ Usage from within the IDE

2014-12-31

1 Contents

2 Purpose of this document

Show a manner to use MicroChip's **PicKit2™ programmer** from within the IDE of an mE PIC compiler. Only tested with the "MikroPascal for PIC" v6.4.0 compiler, but other mE PIC compilers should be no problem.

3 Considerations

- Only using PicKit2™ as a PIC programmer will be possible from the compile's IDE, no **I**n **C**ircuit **D**ebugging.
- PicKit2™ has no "Multiplexer" as the MikroProg™ programmer does, only the lines to the PIC are there, not the lines to the "board". See section 5.

4 Software Installation

The PicKit2™ programmer is a well known PIC programmer. It can be controlled from within the compiler's IDE by installing the following software:

4.1 Download Pk2Cmd.exe and PK2DeviceFile.dat

Download the executable "PK2Cmd.exe" and the data file "PK2DeviceFile.dat" from http://ww1.microchip.com/downloads/en/DeviceDoc/PK2CMDv1-20.zip and place them somewhere in a distinct hard disk directory.

4.2 Make a batch file "Pk2Cmd.bat"

Make a batch file (named "pk2cmd.bat") in the same directory as above, with the following content:

```
@echo off
C:\Tools\PicKit2\pk2cmd.exe -BC:\Tools\PicKit2 -I -P%1 -F%2 -E -Z -J1 -MP -MC -T -R
pause
```

Of course "C:\Tools\PicKit2" should be replaced with the directory where you did put the exe and the batchfile.

For explanation of the PK2Cmd parameters simply type "Pk2Cmd" in a command prompt box in the installation directory of PicKit2™ exe file.

The batch file is made such that the PIC/board to be programmed is powered from the PicKit2.
After programming the PicKit2 will power the programmed device (board) and release MCLR.

Important note:

For some Pic's another pk2cmd command should be used:

```
C:\Tools\PicKit2\pk2cmd.exe -BC:\Tools\PicKit2 -I -P%1 -F%2 -E -Z -J1 -MP -T -R
```

The "-MC" parameter has to be left out for PIC's that don't have an actual configuration memory, but where the configuration bits reside in program memory. This is the case for Pic's where the datasheet states:

The last four words of Flash program memory,
known as the Flash Configuration Words (FCW), are
used to store the configuration data.

The batchfile can be adapted e.g. as follows:

```
@echo off

set Pth=C:\Tools\PicKit2

FOR /F "tokens=1 delims= " %%A IN (%Pth%\PICS_NoConfig.txt) DO if %1==%%A GOTO J5x

"%Pth%\pk2cmd.exe" -B"%Pth%" -I -P%1 -F%2 -E -Z -J1 -MP -MC -T -R
GOTO End

:J5x
"%Pth%\pk2cmd.exe" -B"%Pth%" -I -P%1 -F%2 -E -Z -J1 -MP -T -R
:End

pause
```

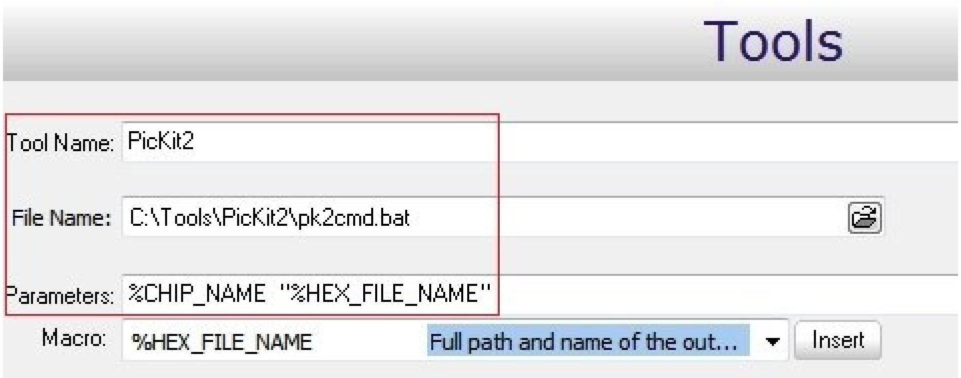
And the file "PICS_NoConfig.txt" in the same directory as the batchfile then contains the names of the "special" PICs (with config bits in program memory):

```
PIC18F27J53
PIC18F26J50
PIC24FJ64GA002
PIC24FJ64GB002
```

Of course the above pic's file has to be extended with other PIC's you use.

4.3 Create a new “Tool” in the compiler’s IDE

Create in the compiler’s IDE (Tools -> Options -> Tools) a new “tool” entry with the following content:



Tool Name: PicKit2

File Name: C:\Tools\PicKit2\pk2cmd.bat

Parameters: %CHIP_NAME "%HEX_FILE_NAME"

Macro: %HEX_FILE_NAME Full path and name of the out...

Figure 1

Of course “C:\Tools\PicKit2” should be replaced with the directory where you did put the exe and the batchfile. Mind the double quotes around HEX_FILE_NAME!

After that the software installation is ready. The new tool PicKit2™ will be visible in the IDE’s “tool” bar. When it is clicked the batch file will receive the PIC’s name and the hexfile to be programmed. Since the hexfile contains the Pic’s configuration words they will also be programmed.

Note: The command in the batchfile can probably be inserted in the IDE’s new “tool” directly, avoiding usage of a batchfile. The reason for using it is an “ease of test” purpose.

5 Hardware Connections

This section described how to connect a PicKit2 programmer to a circuit board for In Circuit Programming.

5.1 To an mE IDC10 connector (mE board pinning)

The big difference between the PicKit2™ programmer and e.g. the MikroProg™ for PIC programmer is that PicKit2™ does not contain a "Multiplexer", see the manual of the MikroProg™ programmer for explanation.

If you want to connect the PicKit2™ to a board with the mE IDC10 connector (using 2x5 male headers), then this is how to connect:

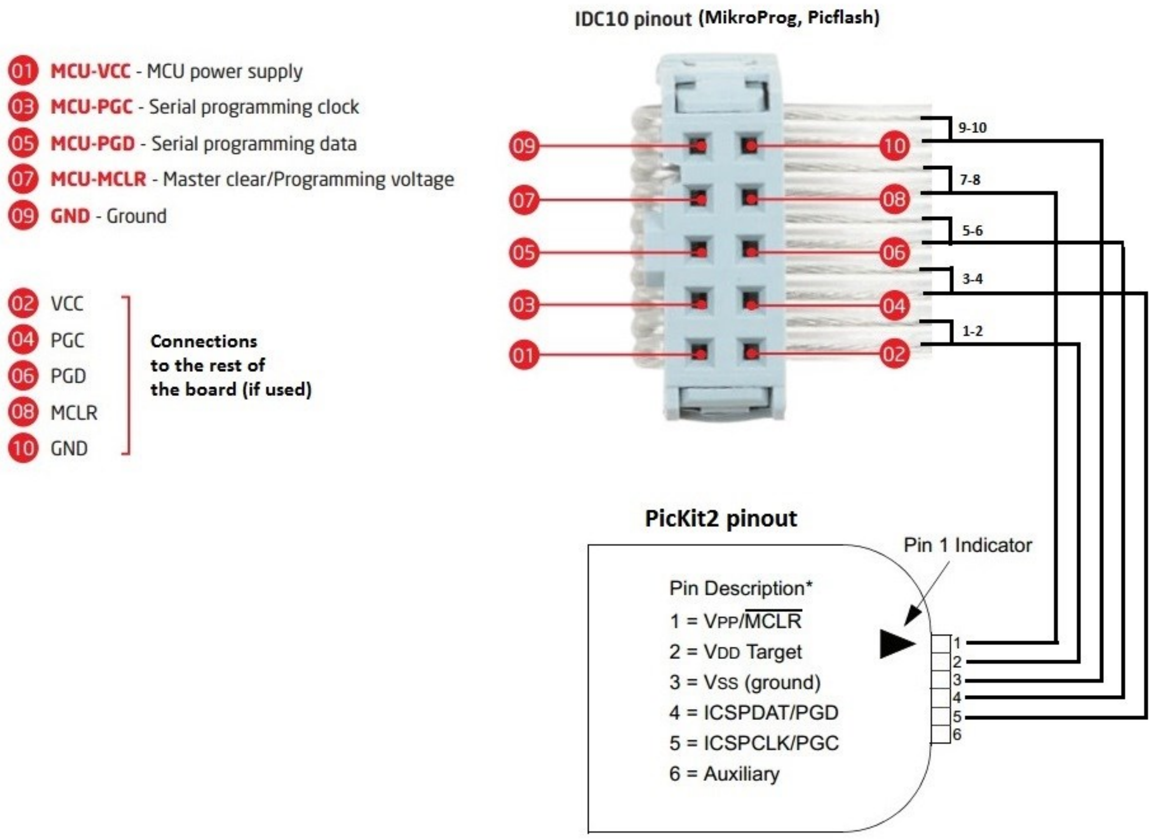

IDC10 pinout (MikroProg, Picflash)
01 MCU-VCC - MCU power supply
03 MCU-PGC - Serial programming clock
05 MCU-PGD - Serial programming data
07 MCU-MCLR - Master clear/Programming voltage
09 GND - Ground
02 VCC
04 PGC
06 PGD
08 MCLR
10 GND
Connections to the rest of the board (if used)
09
07
05
03
01
10
08
06
04
02
9-10
7-8
5-6
3-4
1-2
PicKit2 pinout
Pin 1 Indicator
Pin Description*
1 = VPP/MCLR
2 = VDD Target
3 = VSS (ground)
4 = ICSPDAT/PGD
5 = ICSPCLK/PGC
6 = Auxiliary
1
2
3
4
5
6

Figure 2

For boards with only an mE 1x5 male header simply omit the "even" pinning shown above, only the left side of the IDC10 is present.

5.1.1 Isolation

Since the PicKit2™ has no "Multiplexer" it is necessary to add a so-called "isolation" between the PicKit2™ connections between the PIC and the rest of the circuit (to be controlled by the PIC when it is running) in the lines Vpp/MCLR, PGD and PGC:

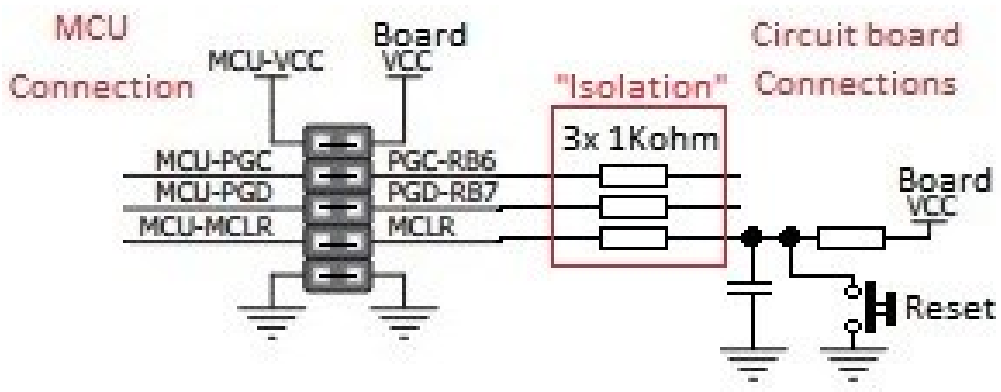


Figure 3

The circuit above is the one when using the PicKit2™ on an mE IDC10 (2x5 pin make header) or on an mE 1x5 male header. Of course the reset circuitry drawn is optional, some PIC's/circuits do not need it.

5.2 To a "standard" (MicroChip) ICSP connector

If using the "MicroChip" pinning for the ICSP connector (= the one of the PicKit2™), then the circuit looks like this:

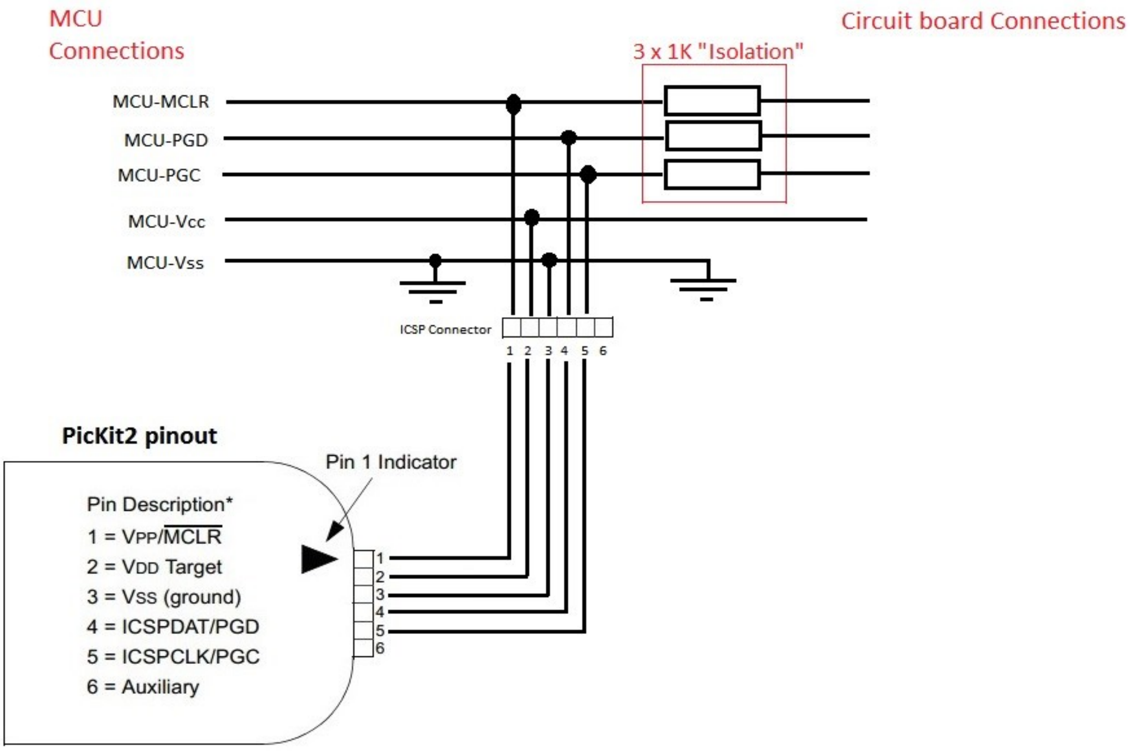


Figure 4

5.2.1 Isolation

See section 5.1.1 and Figure 4 for explanation.

6 Links

PicKit2 related documents:
http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120

[end of document]