

## EASY GUIDE TO THE MPLAB IDE SIMULATOR

### SIMULATION OF AN ASSEMBLY LANGUAGE PROGRAM TO HANDLE 8 LEDS IN PORT B AND A DIGITAL INPUT IN BIT RA4

Description: MPLAB IDE software includes a simulator program, which is a very useful development tool, not only for debugging but also as a teaching assistant.

You can follow step by step each instruction of the program, insert breakpoints, generating bit digital inputs at any port and observe the contents of any internal register of the microcontroller and the memory locations of data and code, EEPROM and Stack.

The simulator also has the option of a "Stopwatch" function to measure execution times.

The example explained below, will simulate a program written in assembly language using the MPLAB IDE. The user may observe the behavior of the bits of Port B, and include a manual simulation of digital inputs (stimulus), namely the bit RA4.

If you have the 18F2550 Bolt system, you may additionally observe the program running in real time, using the leds of PORT B and the nmicroswitch SW1 for PORT A.

1. Create a folder on your PC containing the following two files: "BOLT-PING-PONG.asm" and the auxiliary file "KEYBOLT2.inc" Download these files from the following link:

<http://puntoflotante.net/BOLT-SYSTEM-CONTROL-LEDS-MICROSWITCHES-RELAY.htm>

2. Within the program MPLAB IDE v8.63, click "File" and "Open" file: "BOLT-PING-PONG.asm"

3. This program, written in assembly language, makes the shift to the right or left, of bits on the port B. The shift direction is controlled by the state of bit RA4. In Bolt system 18F2550, this bit is controlled by the microswitch SW1.

4. For the simulator to work properly, edit directive and write: org 0x000

**Continue to configure the simulator:**

5. >Configure> Select Device> 18F2550

6. >Debugger> Select Tool> MPLAB SIM

7. >Debugger> Settings> Processor frequency= 48 Mhz

8. >Debugger> Stopwatch → a window with the stopwatch appears.

9. >Debugger> Stimulus > New Workbook (to configure digital inputs)

10.>Pin/SFR choose RA4 from list.

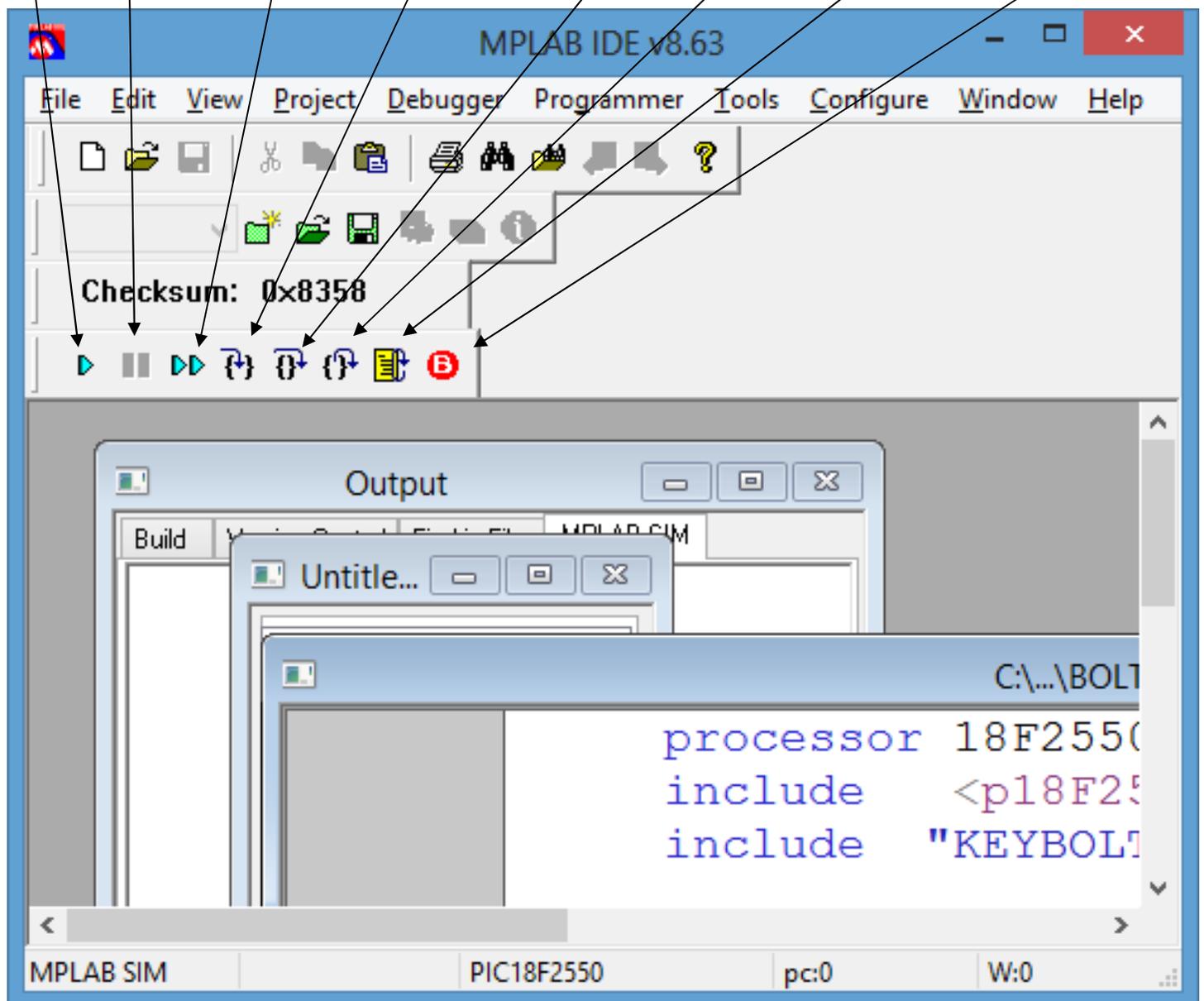
11.>Action: choose "Set High". Modify the size of the window to your needs, as shown in the windows below.

12.>View > Watch > Add SFR (choose to watch registers PORTA and PORTB)

13.>Project> Quickbuild> BOLT-PING-PONG.asm

Identify the commands in the simulator menu (see table below):

Run    Halt    Animate    Step Into    Step Over    Step Out    Reset    Breakpoint



Command	Comments
RUN	Ejecute program until breakpoint found
HALT	Stop executing program
ANIMATE	Step by step automatic
STEP INTO	Step by step manual
STEP OVER	Execution of subroutine until return
STEP OUT	Finishes execution until end of subroutine
RESET	Reset to microcontroller → a green arrow appears
BREAKPOINT	Inserts breakpoint in a line of code

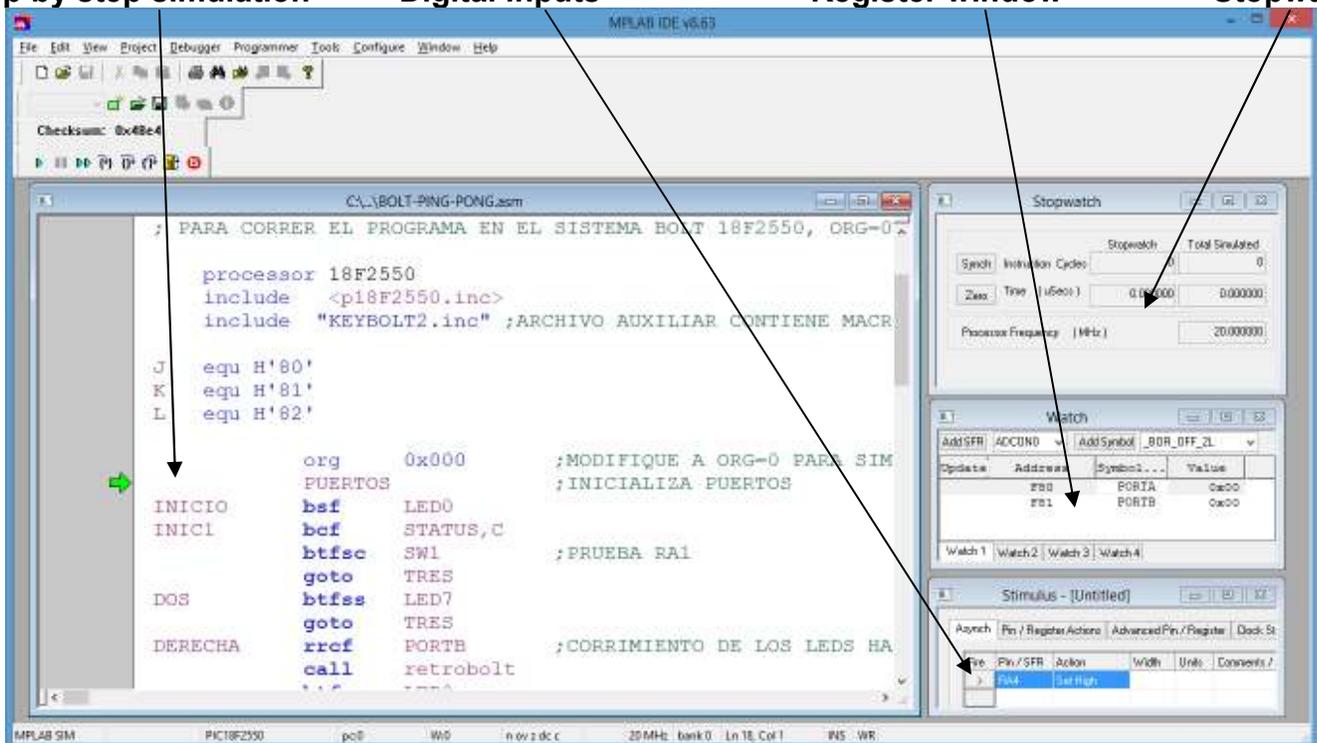
14. During simulation, you may click on button "Fire" to the left of RA4, to activate RA4=1. If you want RA4=0, then choose "Set Low" and click on "Fire".

Start simulation:

15. Click on the "Step Over" option of the simulator menu and perform the simulation step by step and see the shift in the bits of PORT B with values: 01-02-04-08-10-20-40-80 . Remember that all these values are in hexadecimal notation.

16. Continue the simulation with RA4=0 and now observe the values of PORT B with the shift pattern reversed: 80-40-20-10-08-02-01.

Step by step simulation      Digital inputs      Register window      Stopwatch



17. Once you have finished making tests on the simulator, modify again directive org 0x0800

**Assemble the file once again to run in real time:**

18. >Project> Quickbuild> BOLT-PING-PONG.asm

19. Using Bolt software, load the executable file .hex and observe the program running in real time in Bolt 18F2550 system. You may control the shift in leds by changing manually the position of microswitch SW1.

**Any question, please contact:**

**[atencionclientes@puntoflotante.net](mailto:atencionclientes@puntoflotante.net)**