

Starting laboratory session

Methods and tools for software quality

Antonio Martí Campoy



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

amarti@disca.upv.es

Goals



- Getting basic knowledge about the Integrated Development Environment (IDE) of Texas Instrument.
 - Code Composer Studio (CCS)

Plan

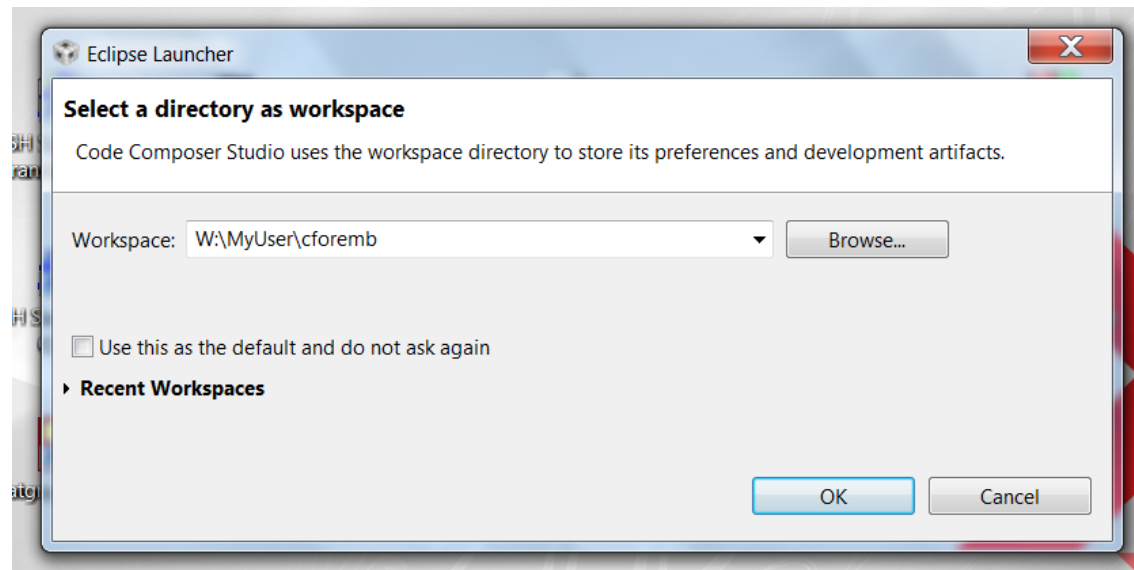


- Creating a new project.
- Setting options for the project.
- Adding robot_library.
- Testing the project.
- Testing the hardware.

Code Composer Studio



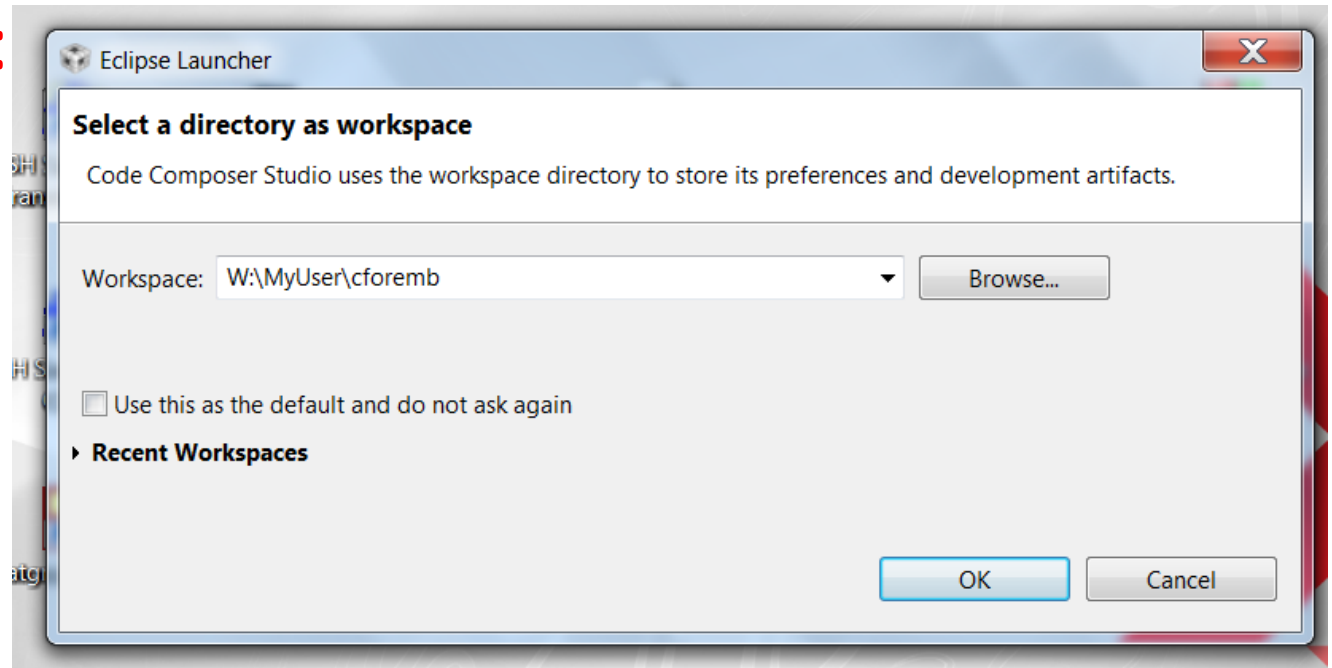
- Execute CCS
 - From shortcut in desktop or folder “Texas Instruments” in menu Start
 - There may be several versions of CCS
 - Choose CCS6 for MSP430



Code Composer Studio



- Select your Workspace
- Option A:



- Option B:
menu File -> Switch workspace -> Other

Create a new project



The screenshot shows the 'New CCS Project' dialog box in Code Composer Studio. The dialog is titled 'New CCS Project' and has a subtitle 'Project name must be specified'. It contains several fields and options:

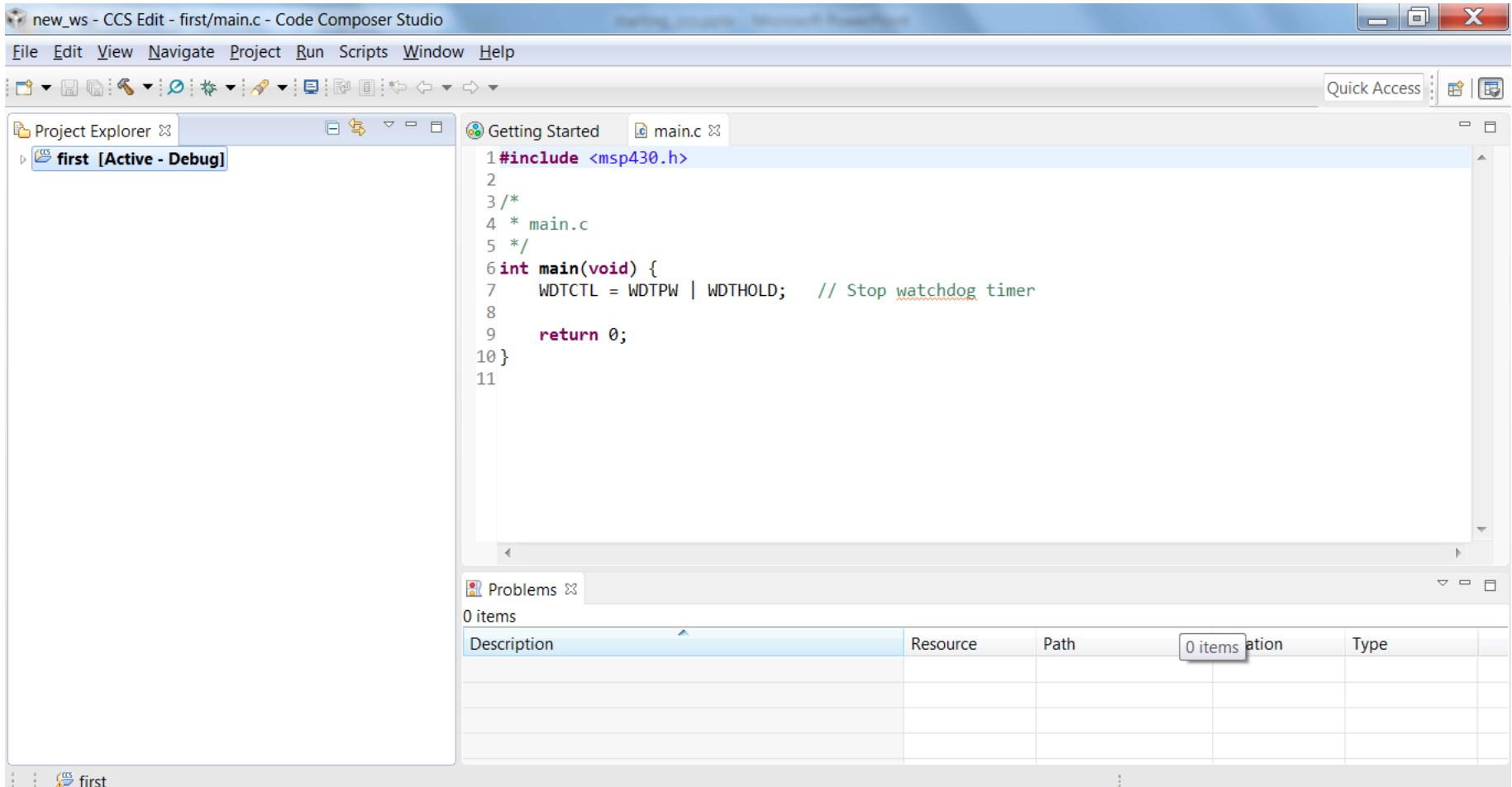
- Target:** A dropdown menu with '<select or type filter text>' as the current selection. A red arrow labeled '1' points to the 'File' menu in the background, and another red arrow labeled '2' points to the 'Target' dropdown.
- Connection:** A dropdown menu with 'TI MSP430 USB1' selected.
- MSP430:** A list of MSP430 microcontroller models. 'MSP430G2553' is highlighted in blue. A red arrow labeled '3' points to this list, and a red box labeled 'Select MSP430G2553' is positioned next to it.
- Project name:** A text input field.
- Use default location:** A checked checkbox.
- Location:** A text input field with the value 'C:\Users\amarti\Dropbox\docencia'.
- Compiler version:** A dropdown menu with 'TI v16.12.0.STS' selected.
- Advanced settings:** A section with a 'Project templates and examples' subsection.
- Project templates and examples:** A list of project templates. 'Empty Project (with main.c)' is selected. A red arrow labeled '4' points to this list, and a red box labeled 'With main.c' is positioned next to it.

At the bottom of the dialog, there are '< Back' and 'Next >' buttons. The background shows the CCS Edit interface with a 'Getting Started' page.

The project



- Here your new project



Preparing main.c

- Replace the generic microcontroller header file with the actual microcontroller header:
 - `#include <msp430.h>` replace with `#include <msp430g2553.h>`
- Add header file for library “system”
 - `#include “system.h”`
- Add header file `<stdint.h>`
 - Allow the use of standard types: `uint8_t`, `int16_t`,...

Preparing main.c

- Change type of function main from int to void:
 - `void main(void)` } //An embedded system
- Remove `return 0;` } never returns
- Add header files “stdint.h” and “intrinsics.h” to have access to standard types and low level functions
- Add in the main function sentences to disable the watchdog.

```
// Stop watchdog timer to prevent time out reset
WDTCTL = WDTPW + WDTHOLD;
```

Setup the project

- Download file “library_ccs_v2_no_source.rar” from folder “robot_library” in moodle:
ent.esigelec.fr
 - Extract files in the folder of your workspace/project
 - Copy all .h files and library_ccs_v3.lib to the folder of your project.
- After disabling the watchdog, add a call to *Clock_graceInit_DCO_12M()*;
 - You have to include header “system.h”

Test the project

- Now, before adding your own code, test if everything is ok building the project:
 1. Compile all files and link them with “Project → Build All or Build Project”
 - Check window “Problems”: expected result is warnings: 0 errors:0
 2. Download and activate debug with “Run → Debug”
 - CCS changes to CCS Debug window.
 - First line of function main is highlighted.

Starting laboratory session

Methods and tools for software quality

Antonio Martí Campoy



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

amarti@disca.upv.es