

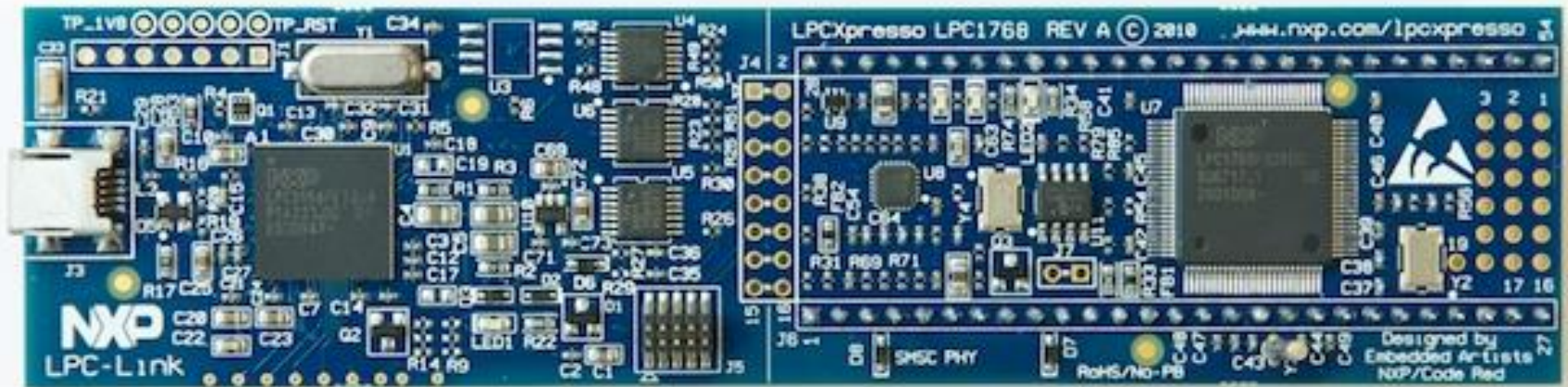
Electrónica Digital III

Temas

- Placa de desarrollo
- Descarga e instalación del entorno LPCXpresso
- Componentes del IDE, Interfaz gráfica
- Primer paso para generar nuestro primer proyecto
- Hardware de la placa de desarrollo, led y LPCXpresso Pinout
- Registros FIODIR, FIOSET y FIOCLR
- Programa de encendido y apagado de un Led
- Compilación y Debug

CortexM3

El diseño del núcleo CortexM3 corresponde ARM, quien vende la licencia para usar su arquitectura a los diferentes fabricantes, el microcontrolador es del fabricante NXP, las placas que vamos a utilizar son de la firma Embedded Artist, el entorno de desarrollo fue creado por CodeRed.



Descarga del IDE <http://www.nxp.com/>

www.nxp.com

NXP Sign In or Register English Cart

ALL Search...

PRODUCTS APPLICATIONS SUPPORT ABOUT

CONGRATULATIONS COLUMBUS, OHIO- US DOT SMART CITY

NXP is proud to work with you (and all the finalists) on smart infrastructure powered by NXP's secure IOT driven solutions

Let's Start Now

SECURE CONNECTIONS FOR A SMARTER WORLD >

PRODUCTS APPLICATIONS SUPPORT ABOUT

Descarga del IDE <http://www.nxp.com/>

www.nxp.com



[Sign In or Register](#)

[English](#) ▼

[Cart](#)

ALL ▼

Search...



PRODUCTS

APPLICATIONS

SUPPORT

ABOUT

ARM® PROCESSORS >

- Kinetis Cortex®-M Microcontrollers
- LPC Cortex-M Microcontrollers
- LPC ARM7/ARM9 Microcontrollers
- i.MX Applications Processors
- QorIQ Multicore Processors

POWER ARCHITECTURE® PROCESSORS >

MORE PROCESSORS >

DISCRETE & LOGIC >

- Bipolar Transistors
- Diodes
- ESD Protection, TVS, Filters
- Logic
- MOSFETs

IDENTIFICATION & SECURITY >

- MIFARE
- NFC

INTERFACE AND CONNECTIVITY >

- I²C
- USB Type-C

MEDIA AND AUDIO PROCESSING >

POWER MANAGEMENT >

RF >

SENSORS >

AUTOMOTIVE PRODUCTS >

- Automotive Radar
- CAN/LIN/FlexRay Transceivers
- Car Access & Immobilizers
- Multi Standard Digital Radio
- S12 & S32 MCUs/MPUs

DEVELOPMENT TOOLS >

SOFTWARE CENTER >

MORE PRODUCT INFORMATION

- All Products
- Competitor Cross Reference Tool
- Packages
- Product Longevity Program
- Product Selector

Discrete & Logic

NFC

Product Selector

SOFTWARE CENTER >

Online Academy

COMMUNITIES >

Descarga del IDE <http://www.nxp.com/>

www.nxp.com/products/software-and-tools:DEVELOPER_HOME

S

APPLICATIONS

SUPPORT

ABOUT

and

gic

Security

nectivity

rocessing

nt

es

icts

s

re

pment

opment

ns and
s

t and

data

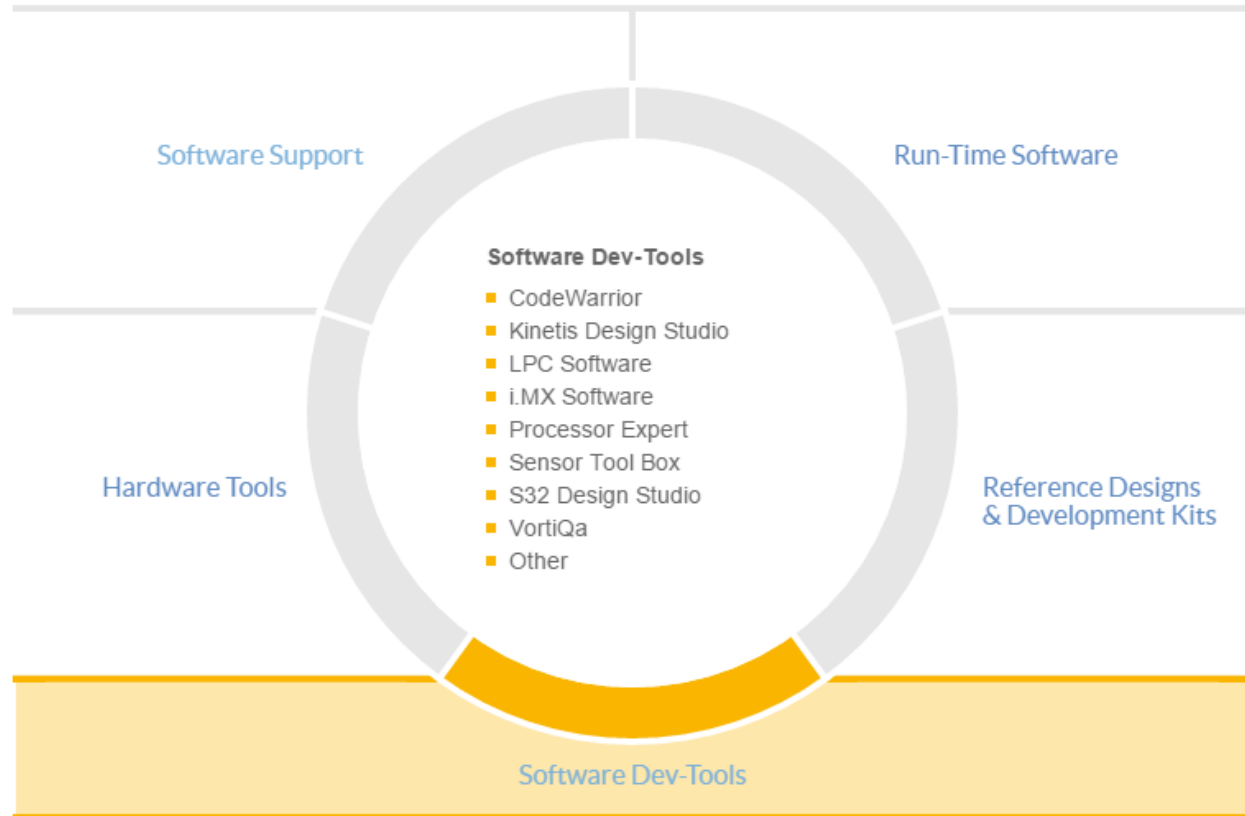
CTOR

real product
specifications.

NXP > Software and Tools



Software and Tools



Descarga del IDE <http://www.nxp.com/>

www.nxp.com/products/software-and-tools/software-development-tools:DEVELOPMENT

Sign In or Register English Cart

ALL Search...

PRODUCTS APPLICATIONS SUPPORT ABOUT

Microcontrollers and Processors
Discretes and Logic
Identification and Security
Interface and Connectivity
Media and Audio Processing
Power Management
RF
Sensors
Single Chip Modules
Automotive Products
Software and Tools

Run-time Software

Software Development Tools

- CodeWarrior Development Tools
- NXP Freedom Sensor Toolbox: Sensor Development Ecosystem
- VortiQa Software for Networking
- Kinetis Software and Tools
- Processor Expert and Embedded Components
- Software & Tools
- i.MX Software and Tools

Hardware Development Tools

Reference Designs and Development Kits

NXP > Software and Tools > Software Development Tools

Software Development Tools

CodeWarrior S32 Design Studio

Kinetis Design Studio Processor Expert

i.MX Software Sensor Toolbox


VortiQa

LPC Software

Software and Tools for LPC MCUs

- LPC software development tools
- LPC evaluation & development boards
- LPC debug & trace probes
- RTOS, drivers & middleware
- Production programming

Descarga del IDE <http://www.nxp.com/>



[Sign In or Register](#) [English](#) [Cart](#)

ALL

PRODUCTS APPLICATIONS SUPPORT ABOUT

Microcontrollers and Processors

ARM® Processors

i.MX Applications Processors

Kinetis Cortex-M MCUs

LPC Cortex-M MCUs

Software & Tools

LPC Partner IDEs and Development Tools

LPC Microcontroller Utilities

LPCXpresso Boards

LPC Partner Evaluation and Development Boards

LPC Debug and Trace Probes

LPCOpen Libraries and Examples

LPC Resources from Ecosystem Partners

LPC Production Programming

LPC IDHs and Consultants

LPC Cortex-M0+/M0

LPC Cortex-M3

LPC Cortex-M4

LPC ARM7/ARM9 MCUs

MAC57Dxxx

QorIQ ARM Processors

S32 ARM Processors & Microcontrollers

Vfxxx Controller

Power Architecture® Processors

More Processors

Discretes and Logic

Identification and Security

Interface and Connectivity

NXP > Microcontrollers and Processors > ARM® Processors > LPC Cortex-M MCUs > Software & Tools

Software and Tools for LPC MCUs

NXP® brings together world-class development platforms, tools, boards, and software from NXP and partners to get you started developing on NXP LPC microcontrollers in just minutes. With advanced yet low-cost ways to evaluate, develop, and debug with LPC MCUs and a choice of toolchains that support the entire Cortex-M portfolio, the LPC developer ecosystem gives you a familiar set of tools no matter which LPC family you choose.

LPC software development tools

LPC evaluation & development boards

LPC debug & trace probes

RTOS, drivers & middleware

Production programming

LPC Software and Tools

LPC software development tools

LPCXpresso IDE

LPC partner IDEs & development tools

LPC evaluation & development boards

LPCXpresso boards

LPCOpen Libraries and Examples

LPC partner evaluation & development boards

LPC debug & trace probes

RTOS, drivers & middleware

LPCOpen libraries & examples

Resources from ecosystem partners

LPC production programming

NXP and ecosystem partners

Software development tools

Application

Debug and trace probes

RTOS and middleware

Evaluation and development boards

RTOS

Board device drivers

Device drivers

Chip device drivers

Production programming

LPC MCU

Design Resources

NXP support and services

LPC community

LPC FAQ

LPC IDHs & consultants

Documentation

LPC Application Notes

LPC Datasheets

Descarga del IDE <http://www.nxp.com/>

NXP

Sign in or Register English Cart

ALL Search

PRODUCTS APPLICATIONS SUPPORT ABOUT

100% > Software and Tools > Software Development Tools > Software & Tools > LPC Microcontroller Utilities

LPCXpresso IDE v8.2.0 UPDATED ☆

Overview Documentation Downloads Buy / Specifications Hardware & Tools Training & Support

Jump To
Overview
Supported Devices
LPCXpresso IDE

Overview

The LPCXpresso IDE gives developers a low-cost way to create high-quality applications for LPC microcontrollers (MCUs). Based on the Eclipse platform, it has many enhancements to simplify application development and debugging. It features the industry-standard GNU toolchain with a choice of libraries: a proprietary, optimized C library or the standard GNU Newlib library. The LPCXpresso IDE can be used to build an executable of any size with full code optimization.

What's New on v8.2.0

- Upgraded GNU tools to ARM launched GCC 5 update 1
- Updated supported C/C++ dialects in IDE preferences and wizards
- Fixed issue with optimization level of CMA4-HardABI Redlib C library build
- Fixed issue with Redlib stm32cortexm0 function incorrectly matching for some input strings
- Corrected size of third RAM bank from 32KB to 16KB on LPC1820, LPC1810 and LPC1810
- Fixed issue causing some peripheral registers not to be displayed debugging LPC5411x MCUs
- Target CPU automatically selected if possible when debugging multicore MCUs, based on project's CPU

More

Download

LPCXpresso IDE

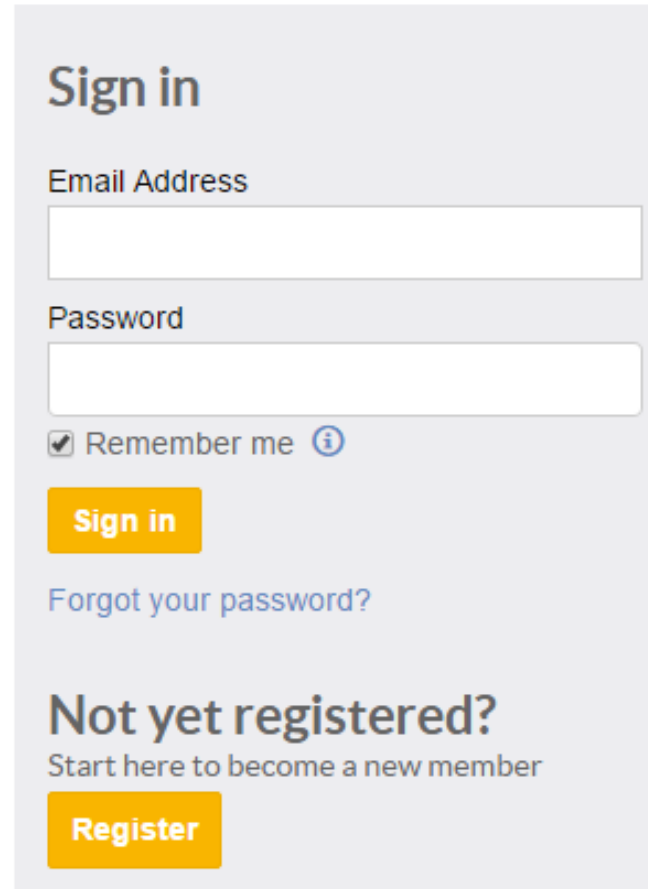
Thumbnail images: LPCXpresso IDE, Debugging, Instruction Trace, Multicore debugging, LPCXpresso, Host Embedded

- Fixed issue causing some perip
- Target CPU automatically select

More

Download

Descarga del IDE <http://www.nxp.com/>



The image shows a login and registration form for NXP. It has a light gray background. At the top, the text 'Sign in' is in a large, bold, dark gray font. Below it, the label 'Email Address' is in a smaller, dark gray font, followed by a white rectangular input field. Below that, the label 'Password' is in a smaller, dark gray font, followed by another white rectangular input field. Under the password field, there is a checkbox with a checkmark inside, followed by the text 'Remember me' and a small blue circular icon with a lowercase 'i'. Below this is a yellow rectangular button with the text 'Sign in' in bold black font. Under the button, the text 'Forgot your password?' is in a blue, italicized font. At the bottom, the text 'Not yet registered?' is in a large, bold, dark gray font. Below it, the text 'Start here to become a new member' is in a smaller, dark gray font. At the very bottom is a yellow rectangular button with the text 'Register' in bold black font.

Sign in

Email Address

Password

☒ Remember me ⓘ

Sign in

Forgot your password?

Not yet registered?

Start here to become a new member

Register

En este punto van a tener que tener una cuenta de NXP para poder descargar el IDE. Si al IDE ya lo tienen de todos modos es necesario crearse una cuenta de NXP para poder activar el IDE y tener disponible todos los kbytes de compilación y debugging que ofrece el software

Descarga del IDE <http://www.nxp.com/>



marcos English Cart

ALL ▾

Search...



PRODUCTS

APPLICATIONS

SUPPORT

ABOUT

NXP > Software & Support > Product Information : LPCXpresso IDE

Software & Support

Product List

Product Search

Order History

Recent Product Releases

Recent Updates

Licensing

License Lists

Offline Activation

FAQ

Download Help

Table of Contents

FAQs

Product Information

LPCXpresso IDE

Select a version. To access older versions, click on the " Previous " tab

Current

Previous

Version	Description	
8.X.X	LPCXpresso IDE for Linux v8.x.x	Download Log
8.X.X	LPCXpresso IDE for MAC v8.x.x	Download Log
8.X.X	LPCXpresso IDE for Windows v8.x.x	Download Log

Descarga del IDE <http://www.nxp.com/>

Software Terms and Conditions

LPCXpresso IDE for Windows v8.x.x

Please read the following agreement and click "I AGREE" at the bottom before downloading your software.

NXP SEMICONDUCTORS USA, INC.

End-User License Agreement for LPCXpresso Software Development Tools

(Rev. 2013-06-28)

THIS END USER LICENSE AGREEMENT ("LICENSE") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR SINGLE LEGAL ENTITY) AND NXP SEMICONDUCTORS USA, INC. ("NXP") FOR THE USE OF THE SOFTWARE ACCOMPANYING THIS LICENSE. NXP IS ONLY WILLING TO LICENSE THE SOFTWARE TO YOU ON CONDITION THAT YOU ACCEPT ALL OF THE TERMS IN THIS LICENSE. BY CLICKING "I AGREE" OR BY INSTALLING OR OTHERWISE USING OR COPYING THE SOFTWARE YOU INDICATE THAT YOU AGREE TO BE BOUND BY ALL THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, NXP IS UNWILLING TO LICENSE THE SOFTWARE TO YOU; DO NOT DOWNLOAD, INSTALL, USE, OR COPY THE SOFTWARE.

I Agree

Cancel

Descarga del IDE <http://www.nxp.com/>

Product Download

LPCXpresso IDE for Windows v8.x.x

Once downloaded and installed, you can [activate LPCXpresso IDE \(Free Edition\)](#) or purchase LPCXpresso IDE (Pro Edition) directly from the [NXP LPCXpresso IDE webstore](#) and then [activate LPCXpresso IDE \(Pro Edition\)](#).

For more information and assistance, please see the [LPCXpresso IDE Installation and Licensing Guide](#) included within the product installation or visit the [LPCXpresso IDE Forums and FAQs](#).

Show All Files 

5 Files

+	File Description	File Size	File Name
+	Installer: LPCXpresso v8.2.0 for Windows	437.4 MB	↓ LPCXpresso_8.2.0_647.exe
+	Installer: LPCXpresso v8.1.4 Windows	438.2 MB	↓ LPCXpresso_8.1.4_606.exe
+	Installer: LPCXpresso v8.1.2 Windows	438.2 MB	↓ LPCXpresso_8.1.2_603.exe
+	Installer: LPCXpresso 8.1.0 Windows	437.9 MB	↓ LPCXpresso_8.1.0_597.exe
+	Installer: LPCXpresso ,8.0.0 Windows	427.8 MB	↓ LPCXpresso_8.0.0_526.exe



Setup - LPCXpresso



Select Additional Tasks

Which additional tasks should be performed?



Select the additional tasks you would like Setup to perform while installing LPCXpresso, then click Next.

Additional icons:

- ☒ Create a desktop icon
- ☒ For all users
- ☐ For current user only

v7.3.0_186

<http://www.nxp.com/lpcxpresso>

< Back

Next >

Cancel



Setup - LPCXpresso



Ready to Install

Setup is now ready to begin installing LPCXpresso on your computer.



LPCXPRESSO

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:

C:\nxp\LPCXpresso_7.3.0_186

Start Menu folder:

LPCXpresso v7.3.0_186

Additional tasks:

Additional icons:

Create a desktop icon

For all users

Install drivers:

NXP Debug drivers

v7.3.0_186

<http://www.nxp.com/lpcxpresso>

< Back

Install

Cancel



Setup - LPCXpresso



Installing

Please wait while Setup installs LPCXpresso on your computer.



LPCXPRESSO

Extracting files...

C:\...\Drivers\pc18xx_43xx_winusb_drivers_v1.00\x86\WUDFUpdate_01009.dll

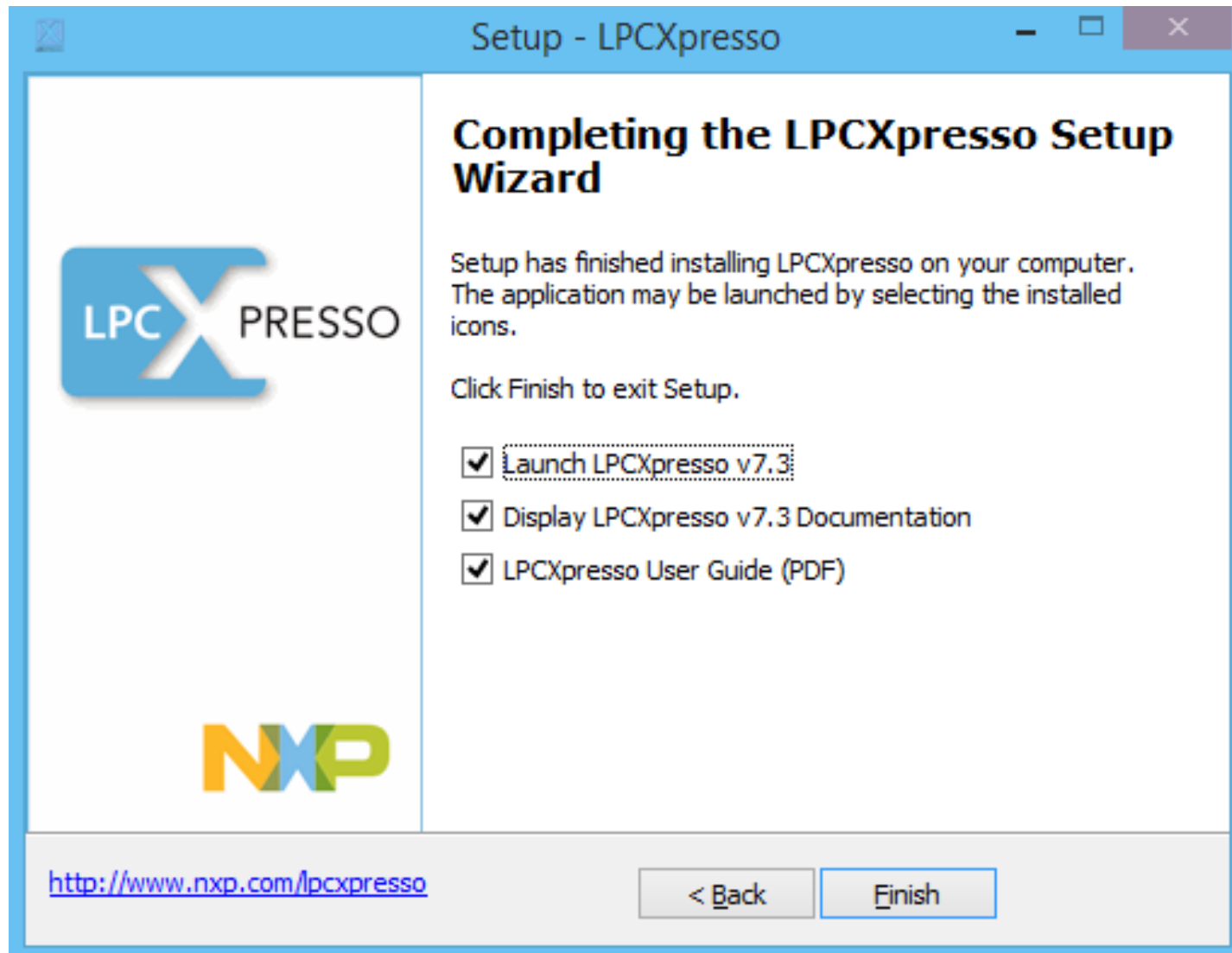


v7.3.0_186

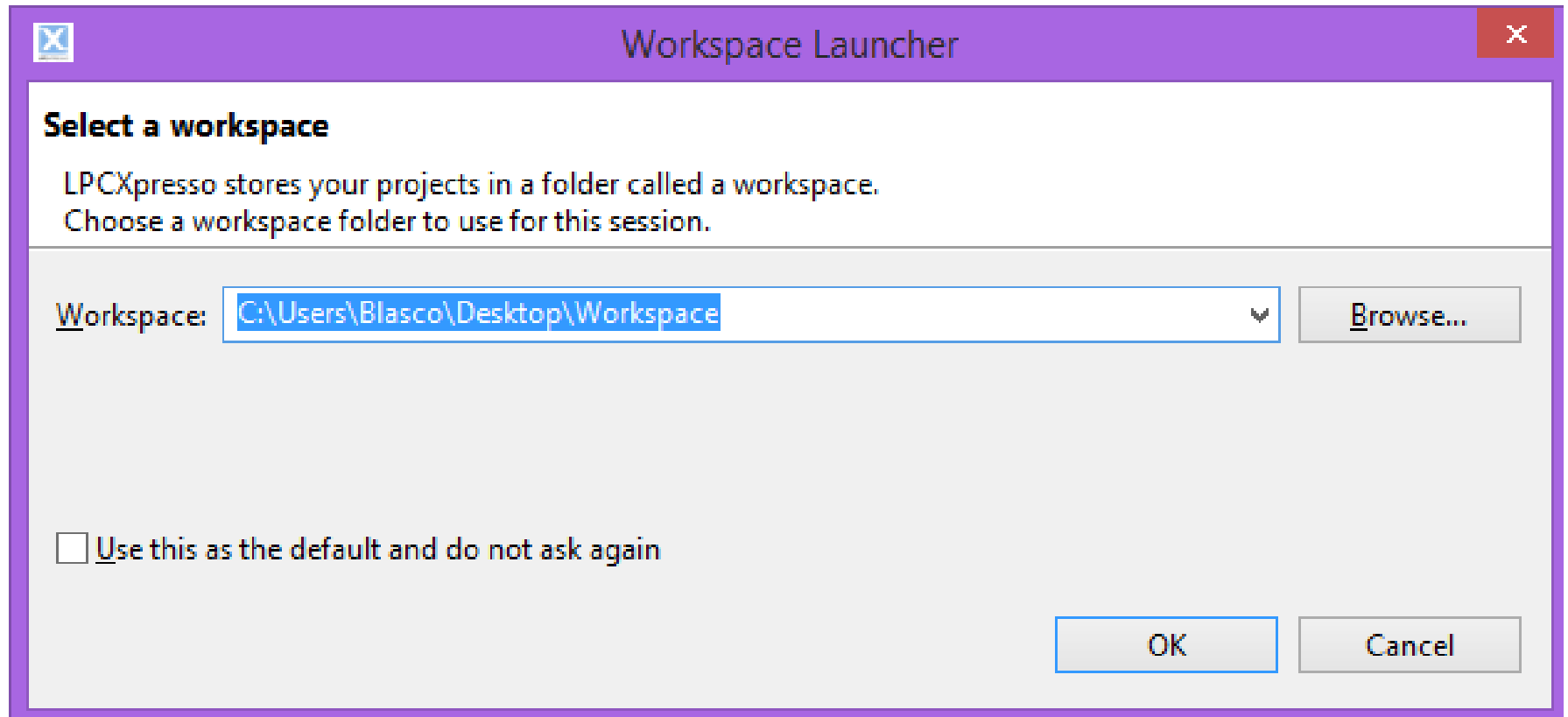
<http://www.nxp.com/lpcxpresso>

Cancel

Finalizamos y ejecutamos el entorno de desarrollo



Seleccionamos el Workspace





© 2013 NXP Semiconductors, B.V. All rights reserved.



Componentes del IDE

Integrated Development Environment

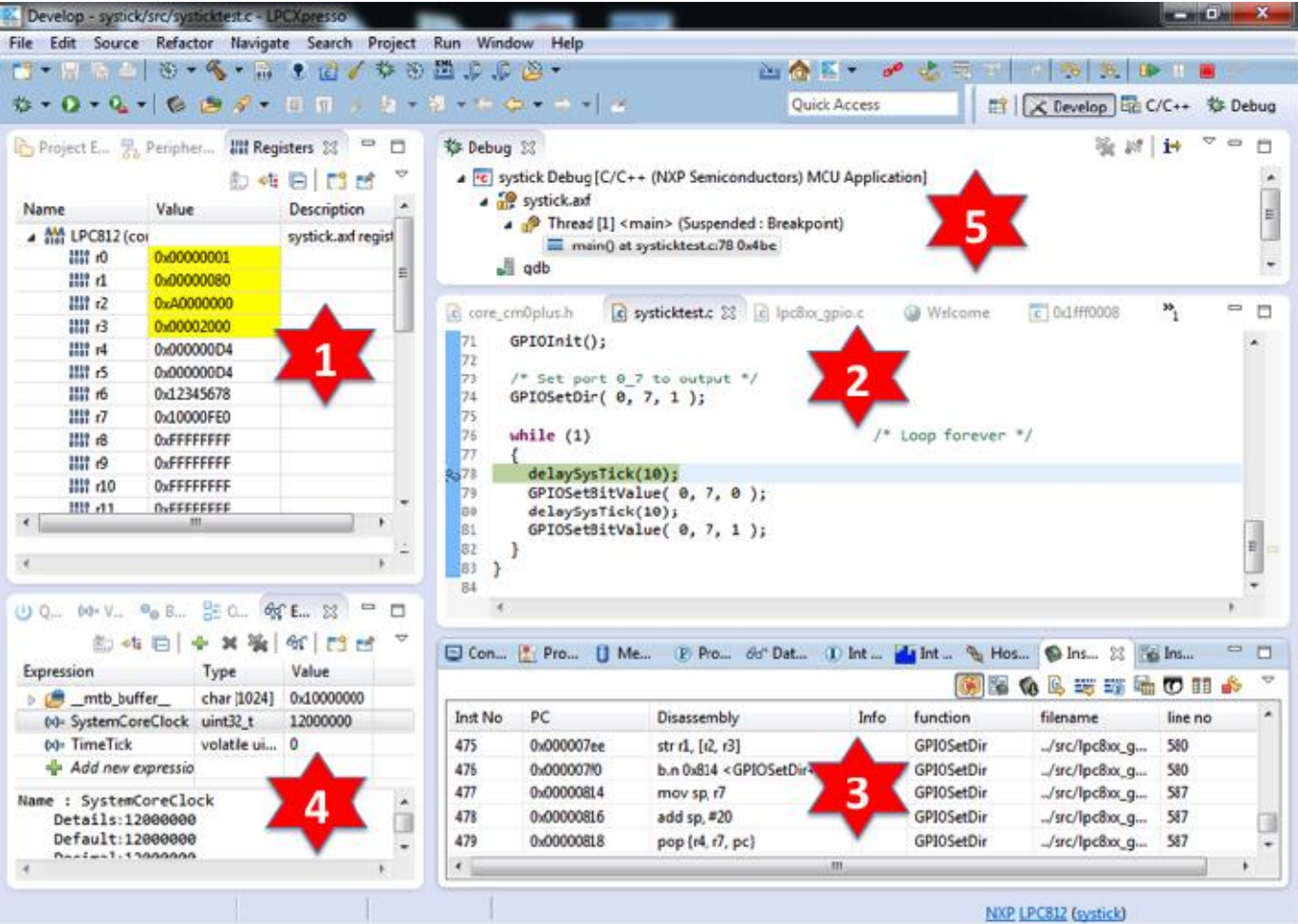
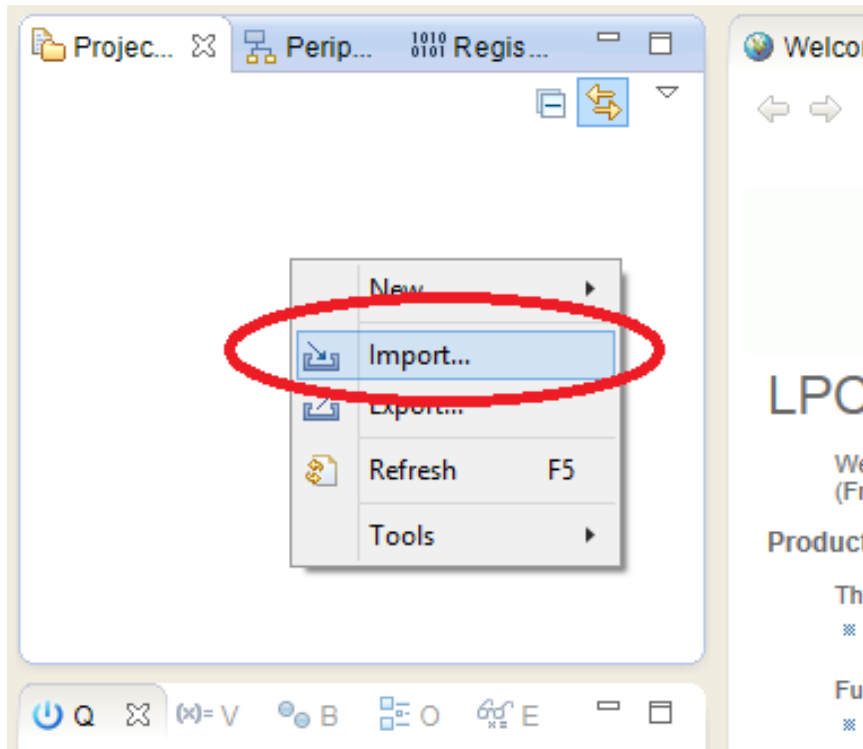


Figure 3.3. Develop Perspective (whilst debugging)

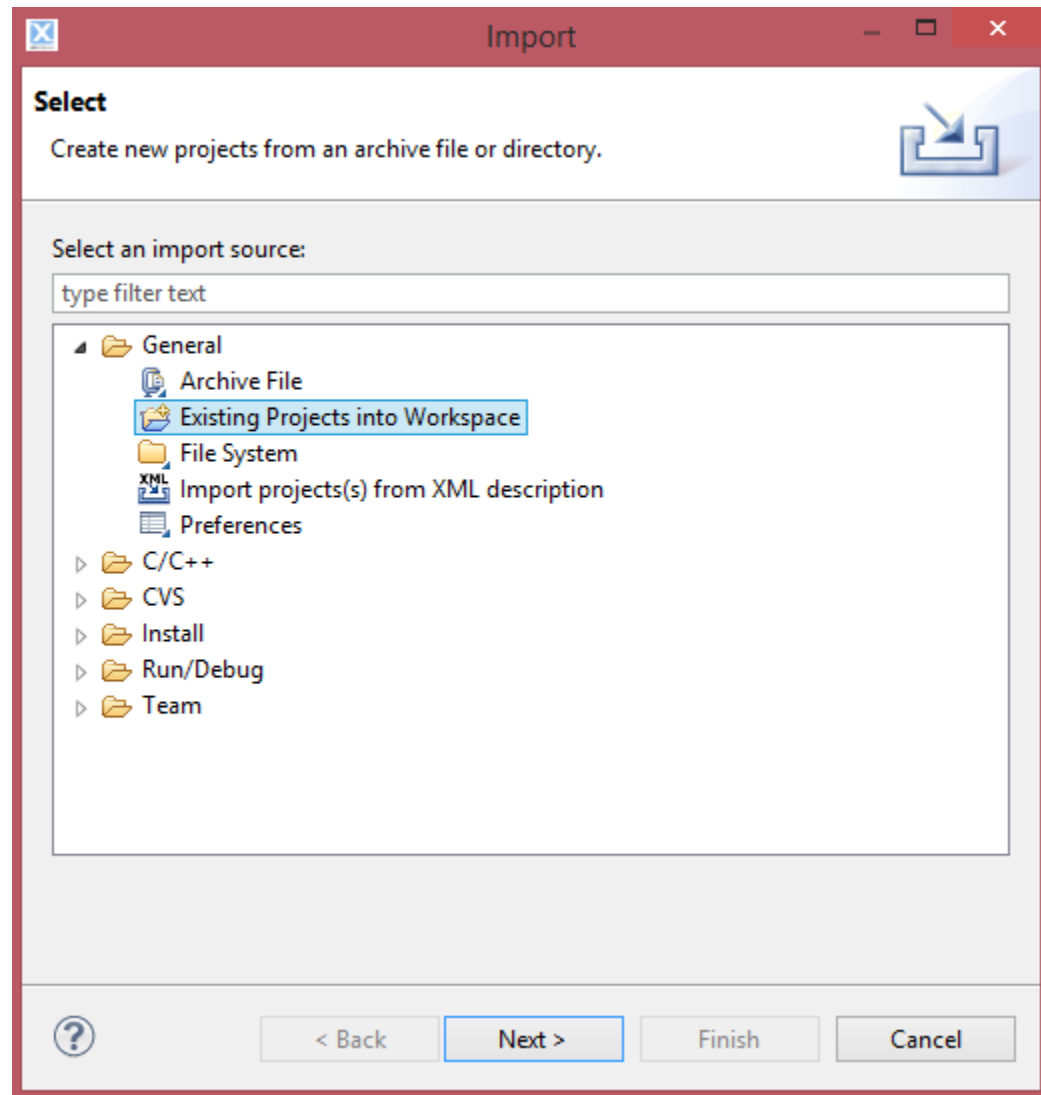
- 1) Project explorer/Peripherals/registers
- 2) Editor
- 3) Consola/Problemas
- 4) Quick start/variables/breakpoint/expression view
- 5) Debug view

Primer paso para generar nuestro primer proyecto

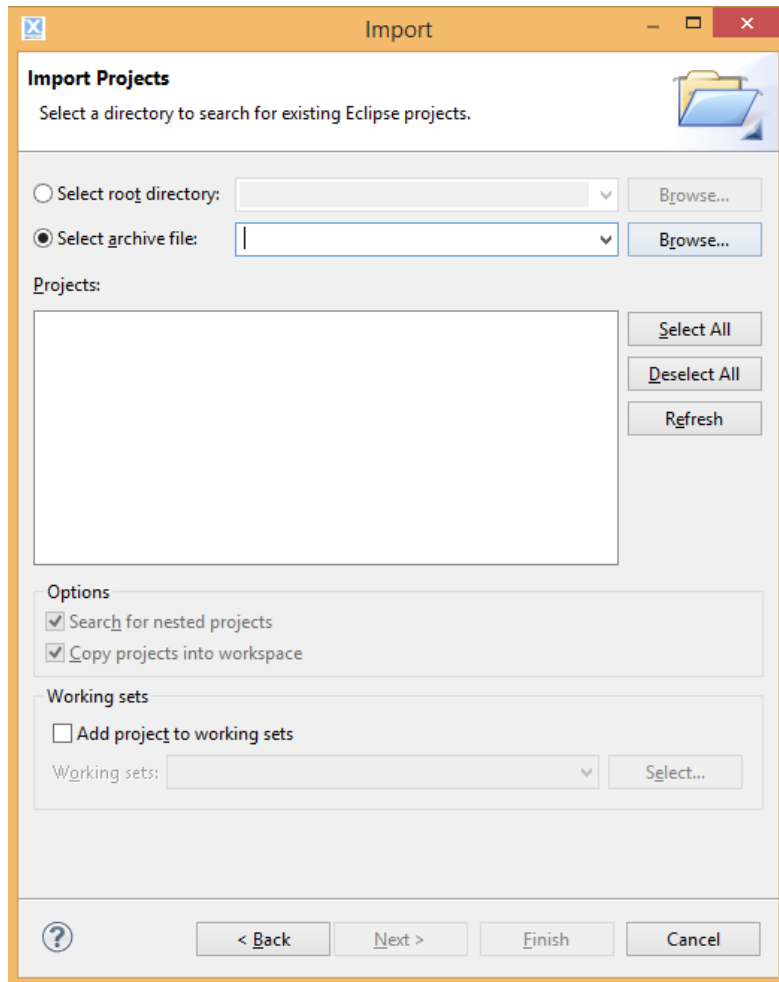
1) Click secundario en Project Explorer y seleccionamos la opción Import



2) Se abre la siguiente ventana de diálogo, dentro de la categoría “General” seleccionamos la opción “Existing Project into Workspace” y hacemos click en Next

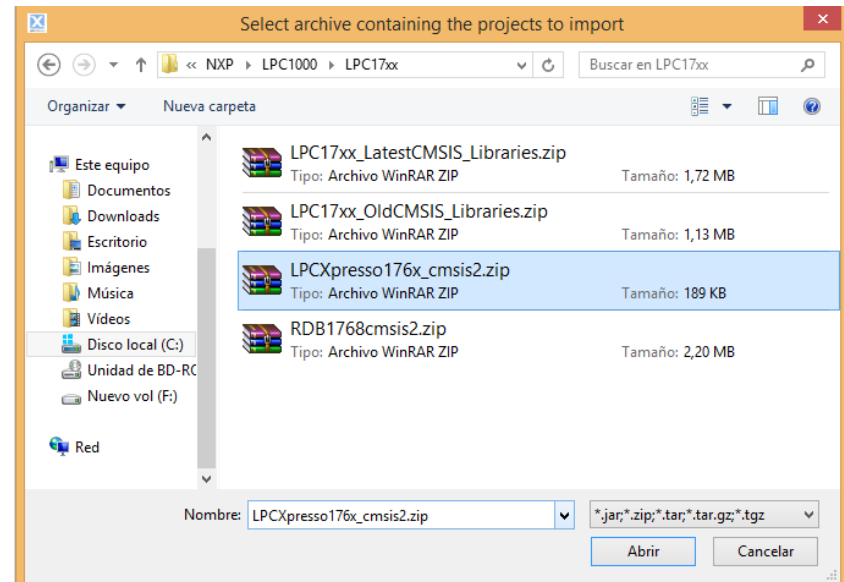


3) Buscamos el archivo de definiciones, de manera similar a como hacíamos en MPLAB, buscamos el archivo donde teníamos las definiciones de todos los registros de los periféricos, como hacíamos con .inc de MPLAB.

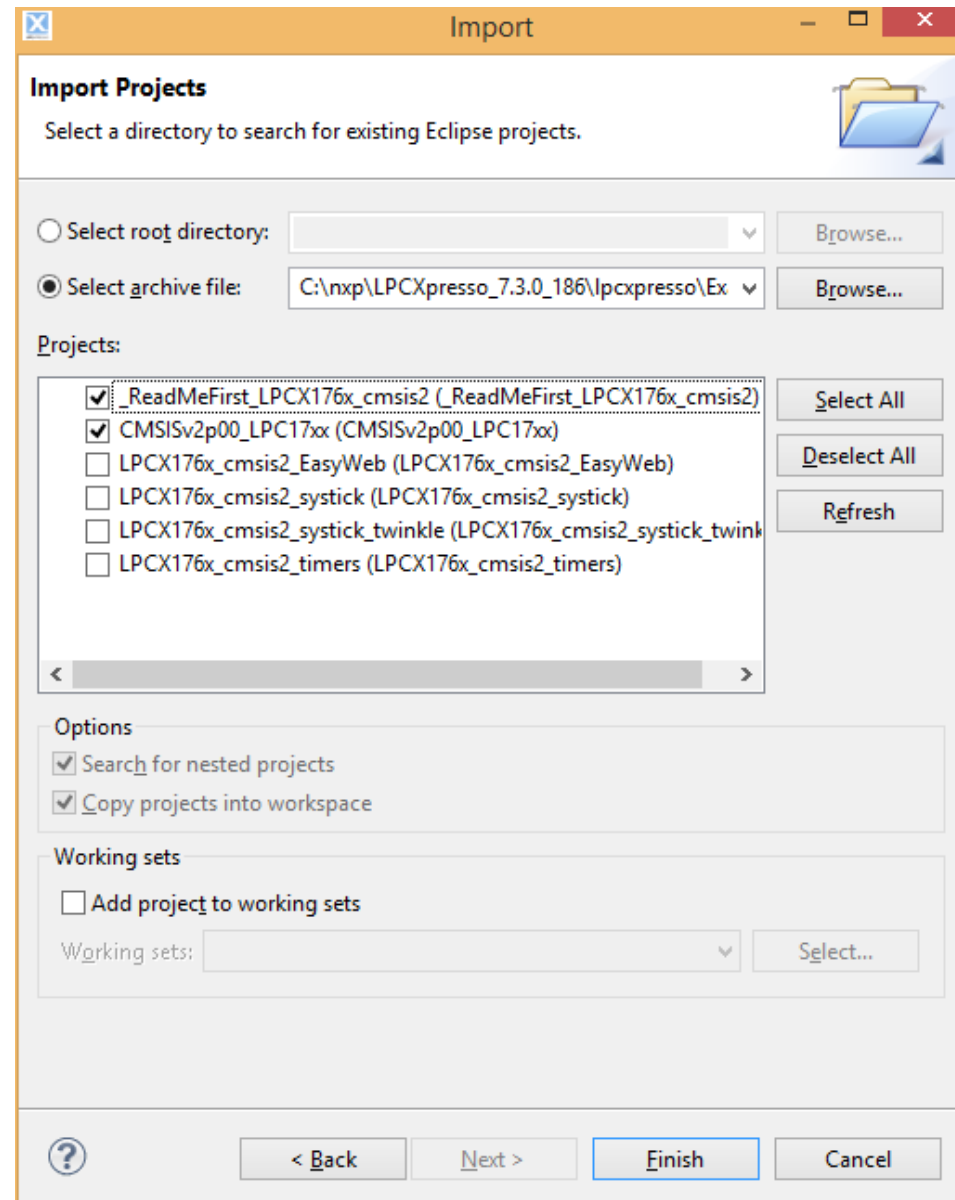


4) Este archivo de definiciones normalmente está en el directorio de instalación de LPCXpresso

C:\nxp\LPCXpresso_7.3.0_186\lpcxpresso\Examples\NXP\LPC1000\LPC17xx

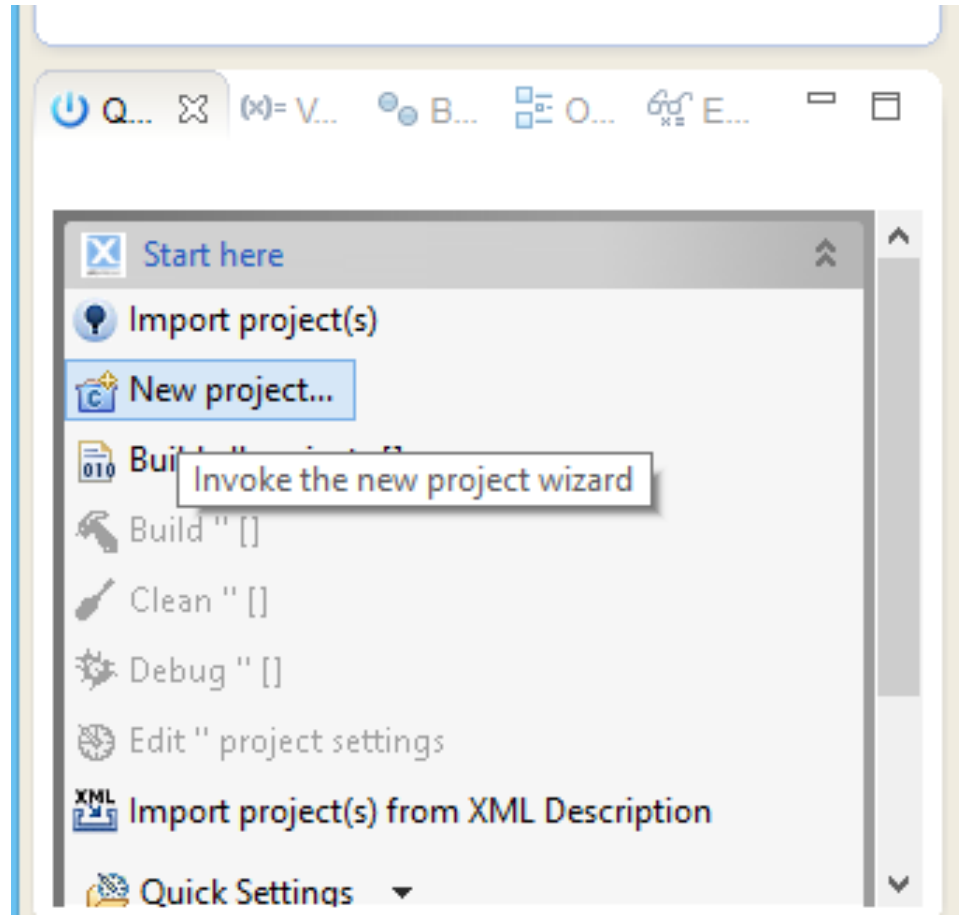


5) De todos los archivos seleccionamos los que necesitamos, en este caso solo los 2 primeros. Hacemos click en finish para importar estos archivos a nuestro Workspace, podemos verificar que se copiaron a nuestra carpeta de Workspace y ahora está disponible para todos los proyectos que coloquemos ahí.

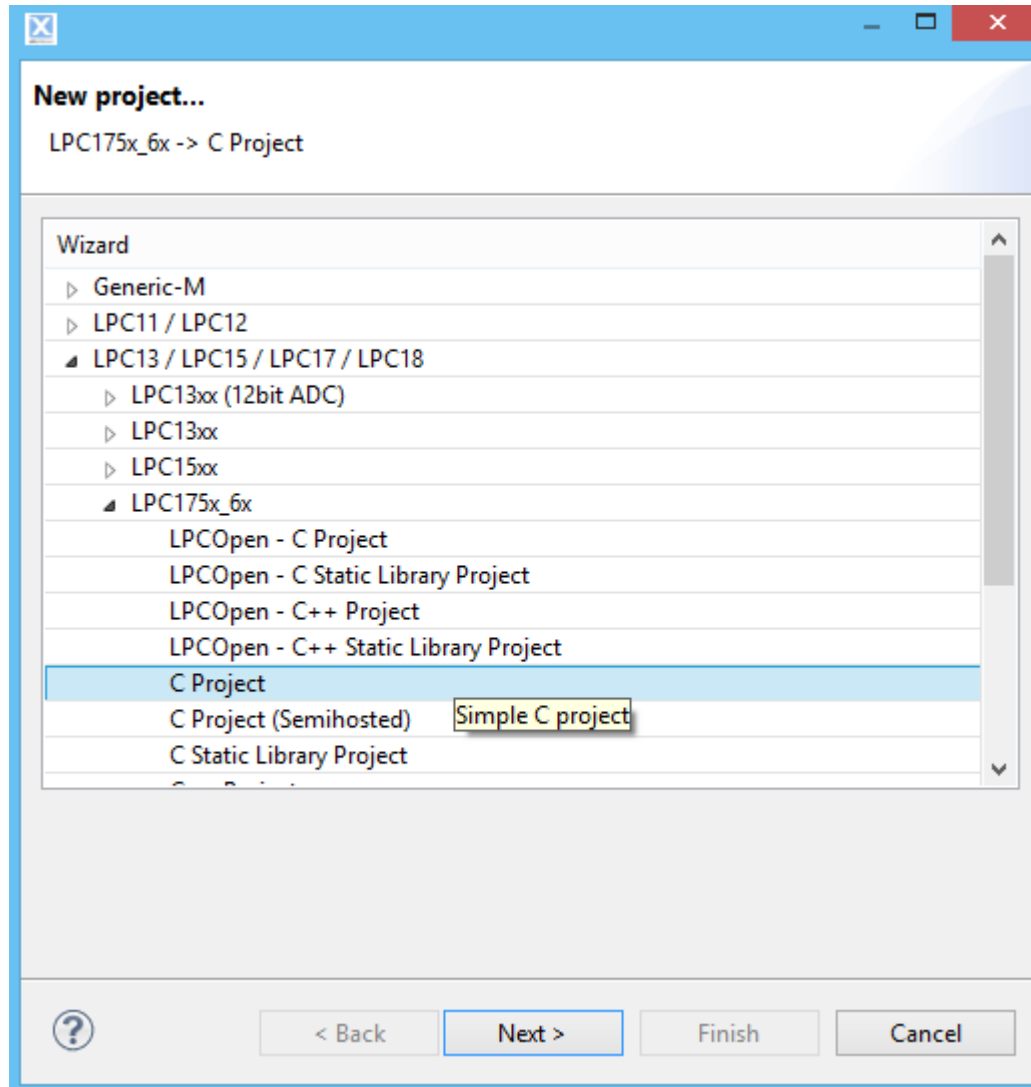


6) Ahora crearemos un nuevo proyecto

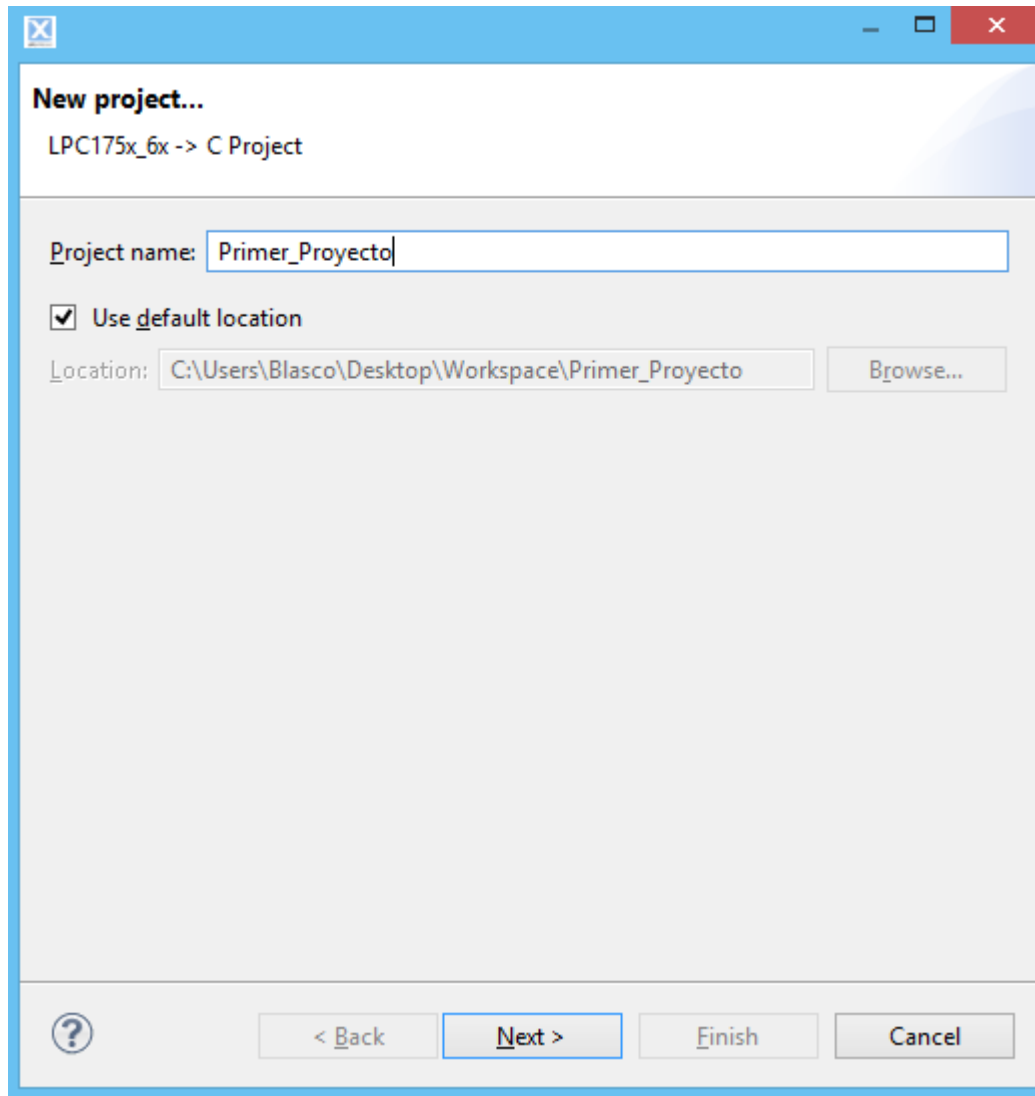
Dentro de quickstart vamos a la opción New project



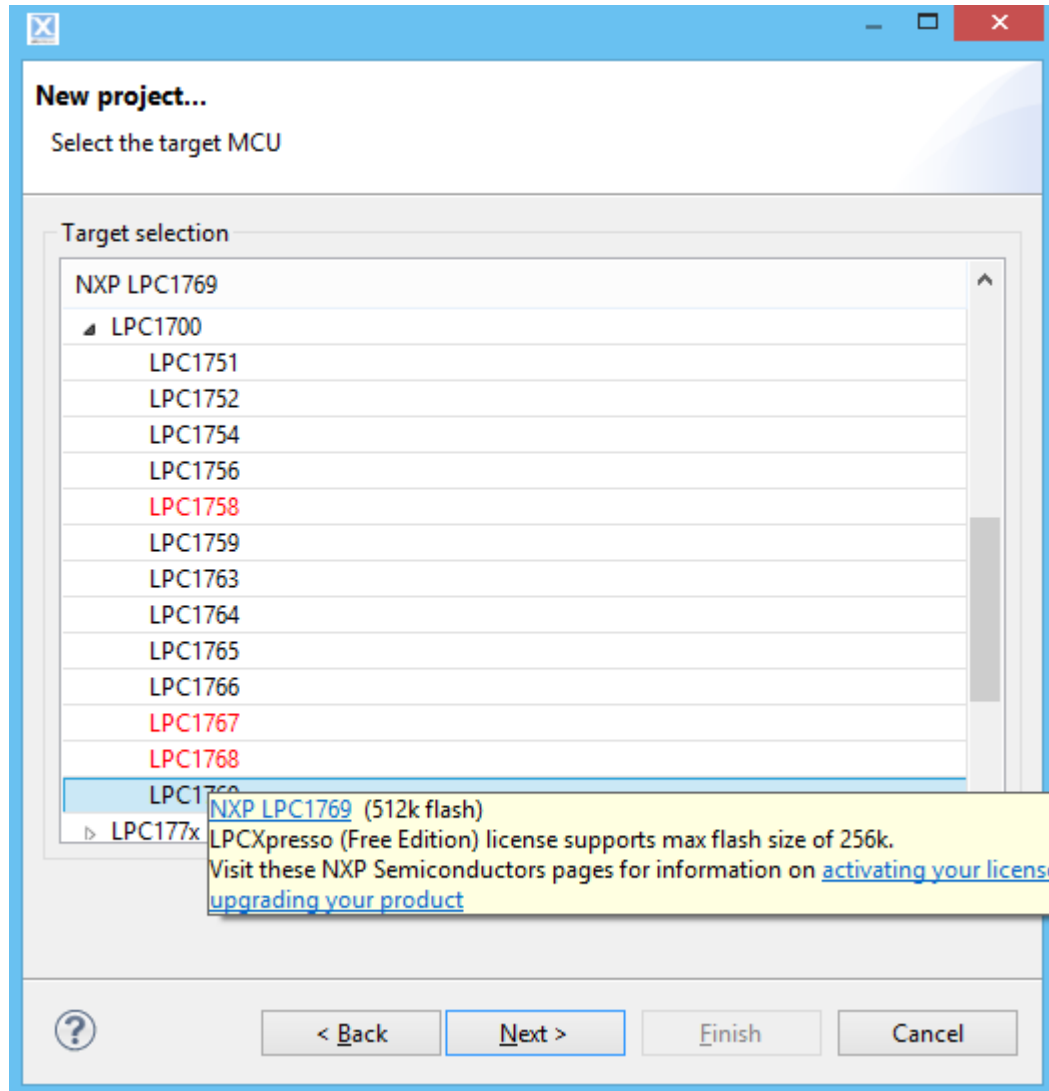
7) Se abre la siguiente ventana, de la cual seleccionaremos la opción que puede verse resaltada en celeste y presionamos Next



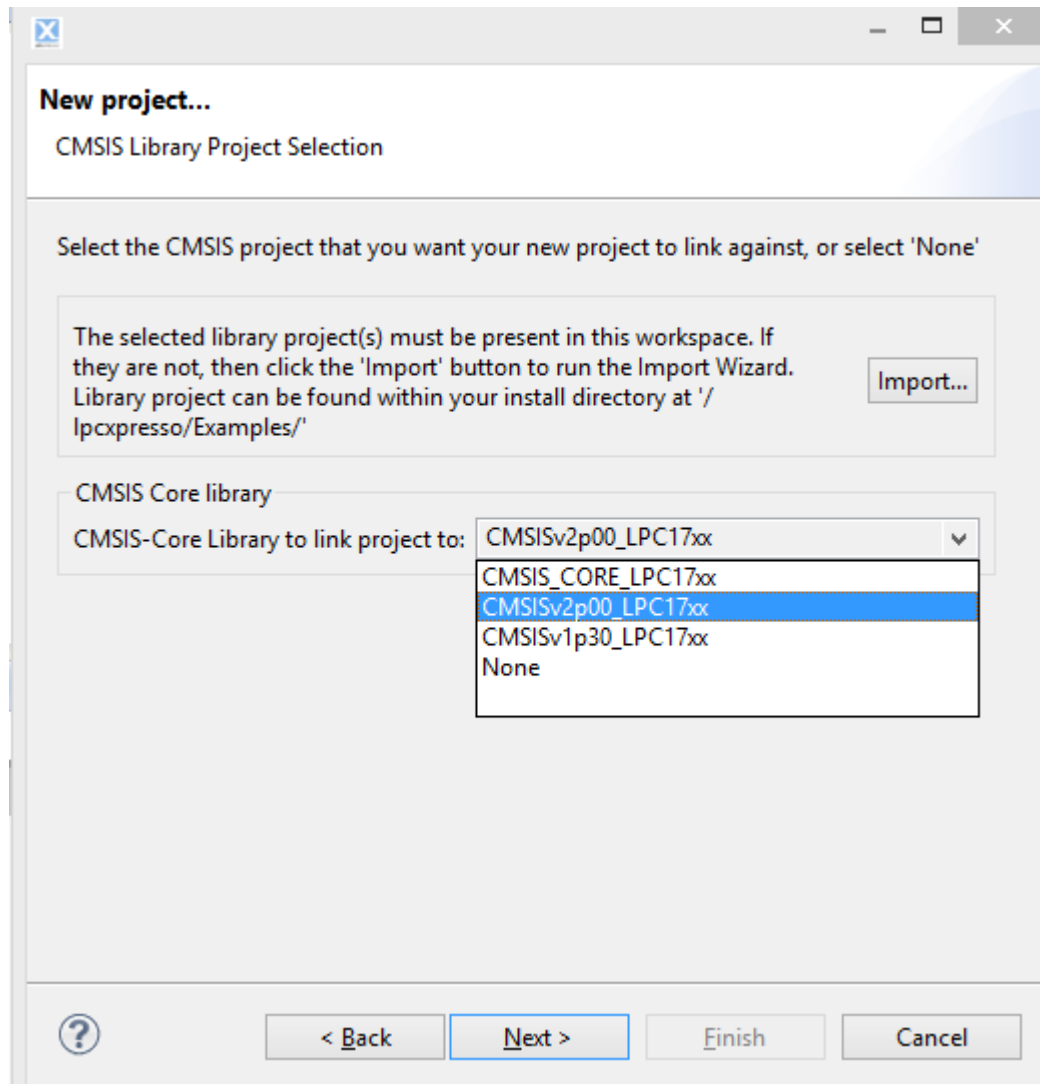
8) Colocamos nombre a nuestro proyecto y presionamos Next, esto creará la carpeta de proyecto



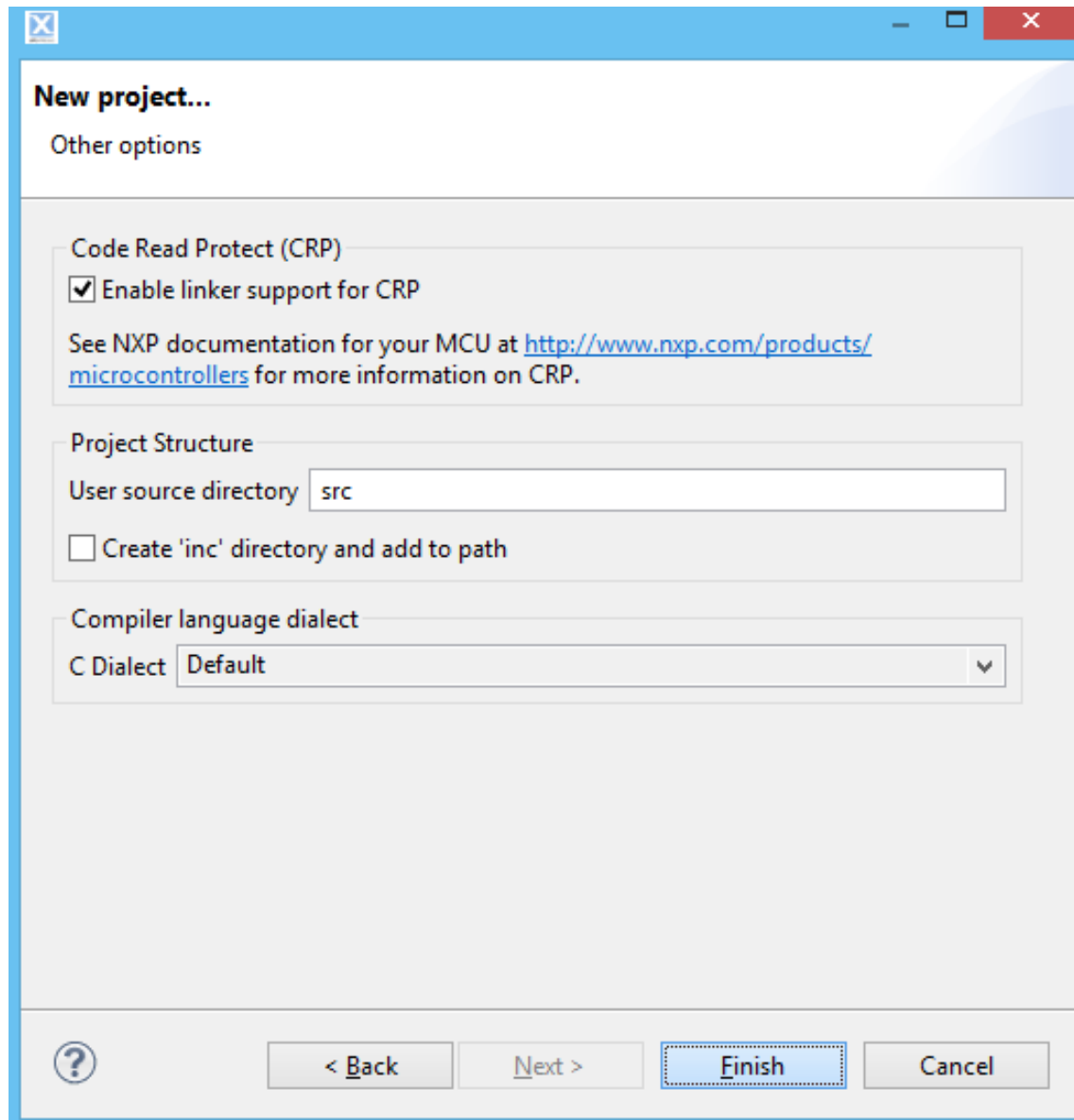
9) Seleccionamos el microcontrolador, aparece en rojo, esto es porque este micro tiene más memoria de la que podemos compilar con la versión gratuita, esto no es problema dado que no llegaremos en este curso a hacer aplicaciones que excedan esa cantidad.



10) Seleccionamos la librería CMSIS que agregamos a nuestro proyecto anteriormente.



11) Seleccionamos Next hasta que llegamos a esta ventana y presionamos Finish.



New project...

Other options

Code Read Protect (CRP)

☒ Enable linker support for CRP

See NXP documentation for your MCU at <http://www.nxp.com/products/microcontrollers> for more information on CRP.


Project Structure

User source directory

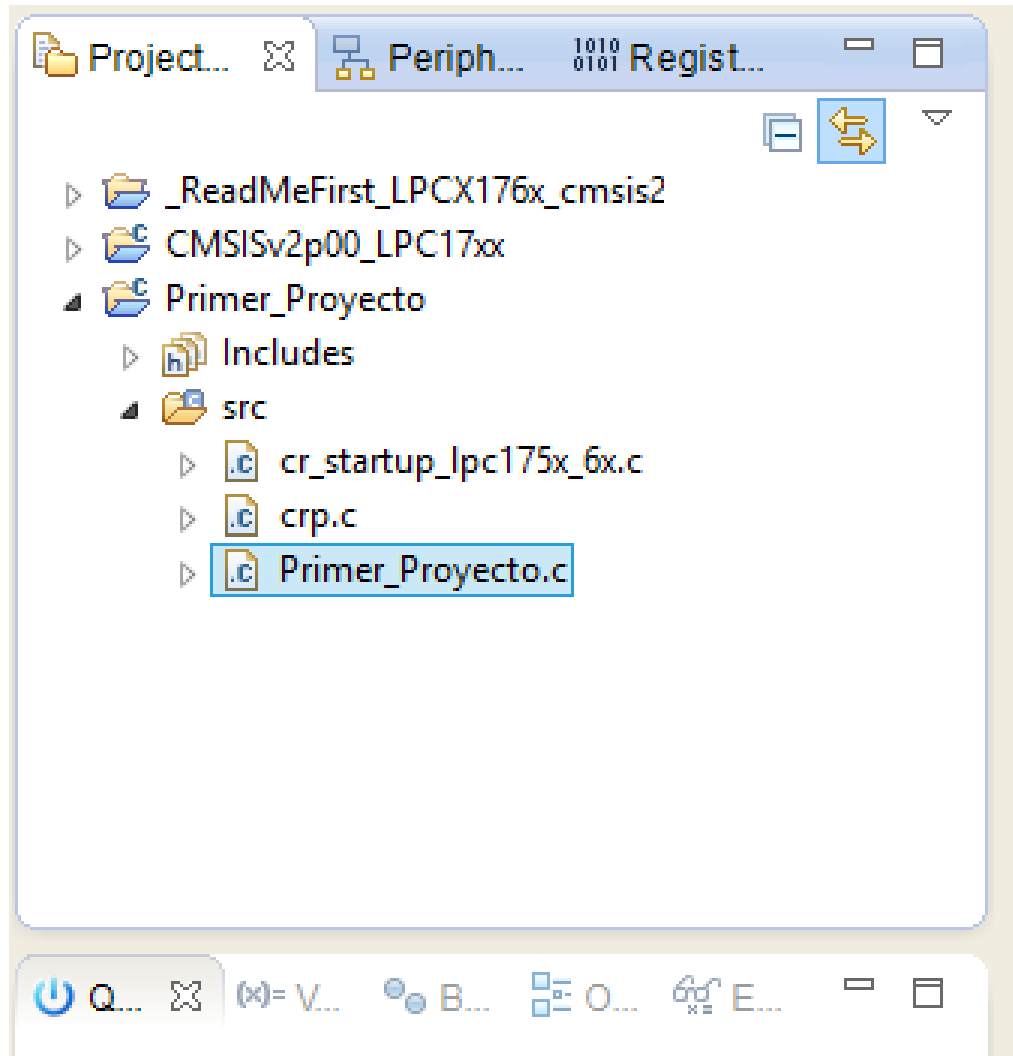
☐ Create 'inc' directory and add to path

Compiler language dialect

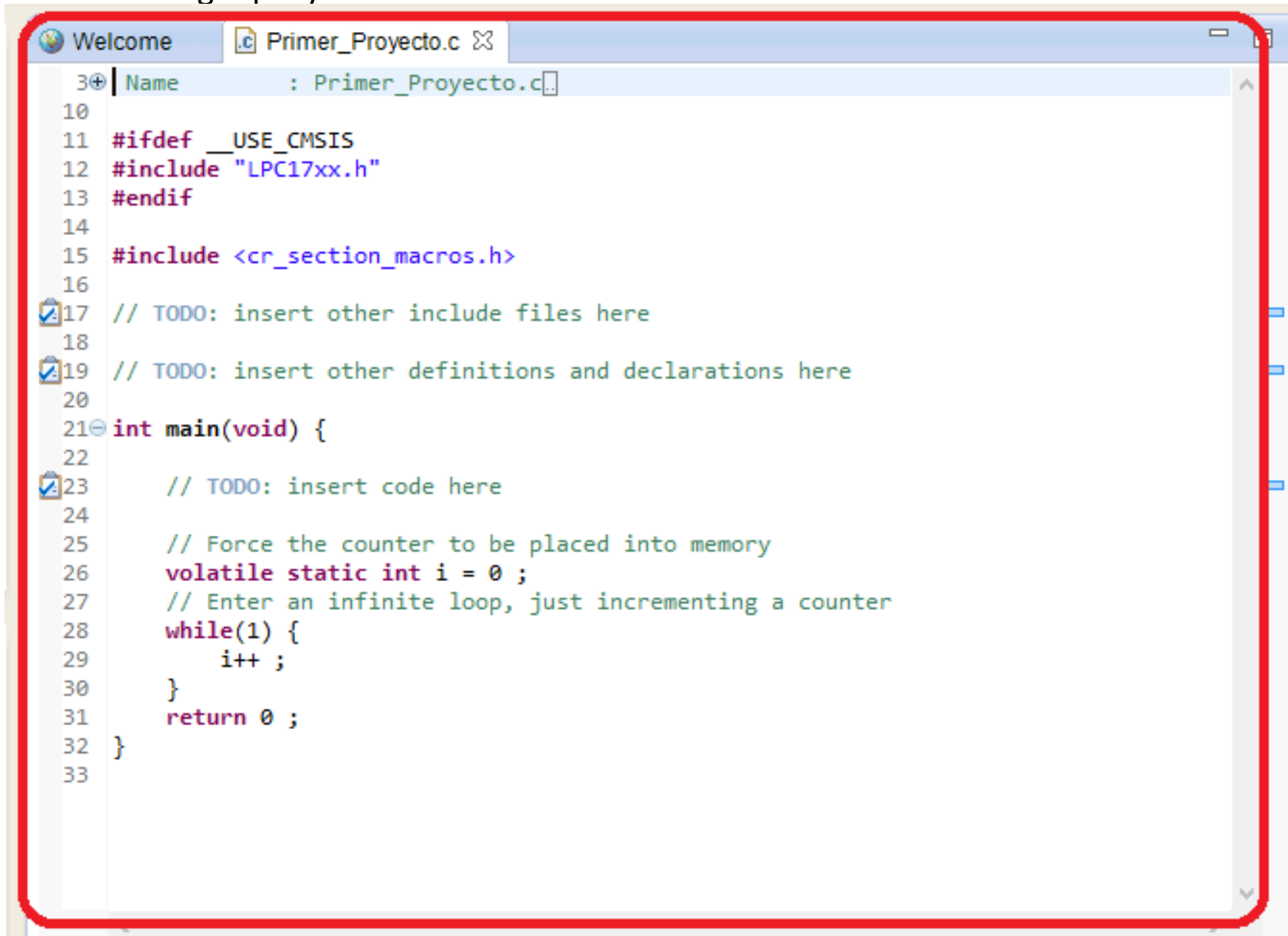
C Dialect



12) Podemos ver en el Project explorer que se generó la carpeta con nuestro proyecto que llamamos “Primer_Proyecto”.

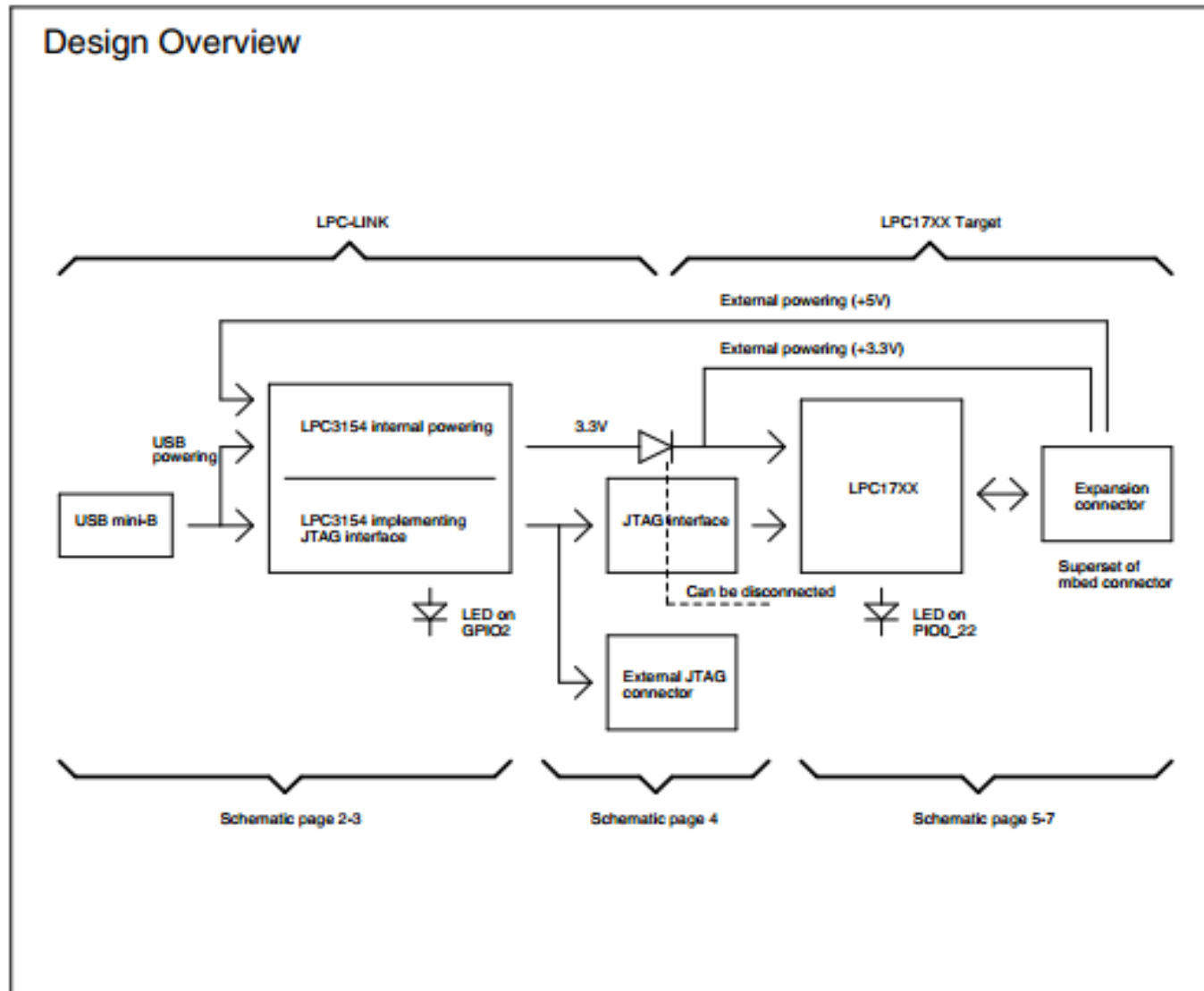


13) Hacemos doble click en “Primer_Proyecto.c” para visualizar el código de este en el editor de código que ya habíamos mencionado



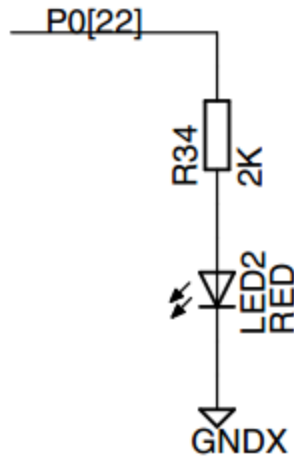
```
10
11 #ifndef __USE_CMSIS
12 #include "LPC17xx.h"
13 #endif
14
15 #include <cr_section_macros.h>
16
17 // TODO: insert other include files here
18
19 // TODO: insert other definitions and declarations here
20
21 int main(void) {
22
23     // TODO: insert code here
24
25     // Force the counter to be placed into memory
26     volatile static int i = 0 ;
27     // Enter an infinite loop, just incrementing a counter
28     while(1) {
29         i++ ;
30     }
31     return 0 ;
32 }
33
```


14) Intentaremos ahora encender y apagar un led. Para esto veremos por un lado el hardware disponible en nuestra placa, por otro los nombres de los registros que debemos escribir. En la siguiente imagen se ve a nivel de bloques la los elementos disponibles en la placa LPCXpresso

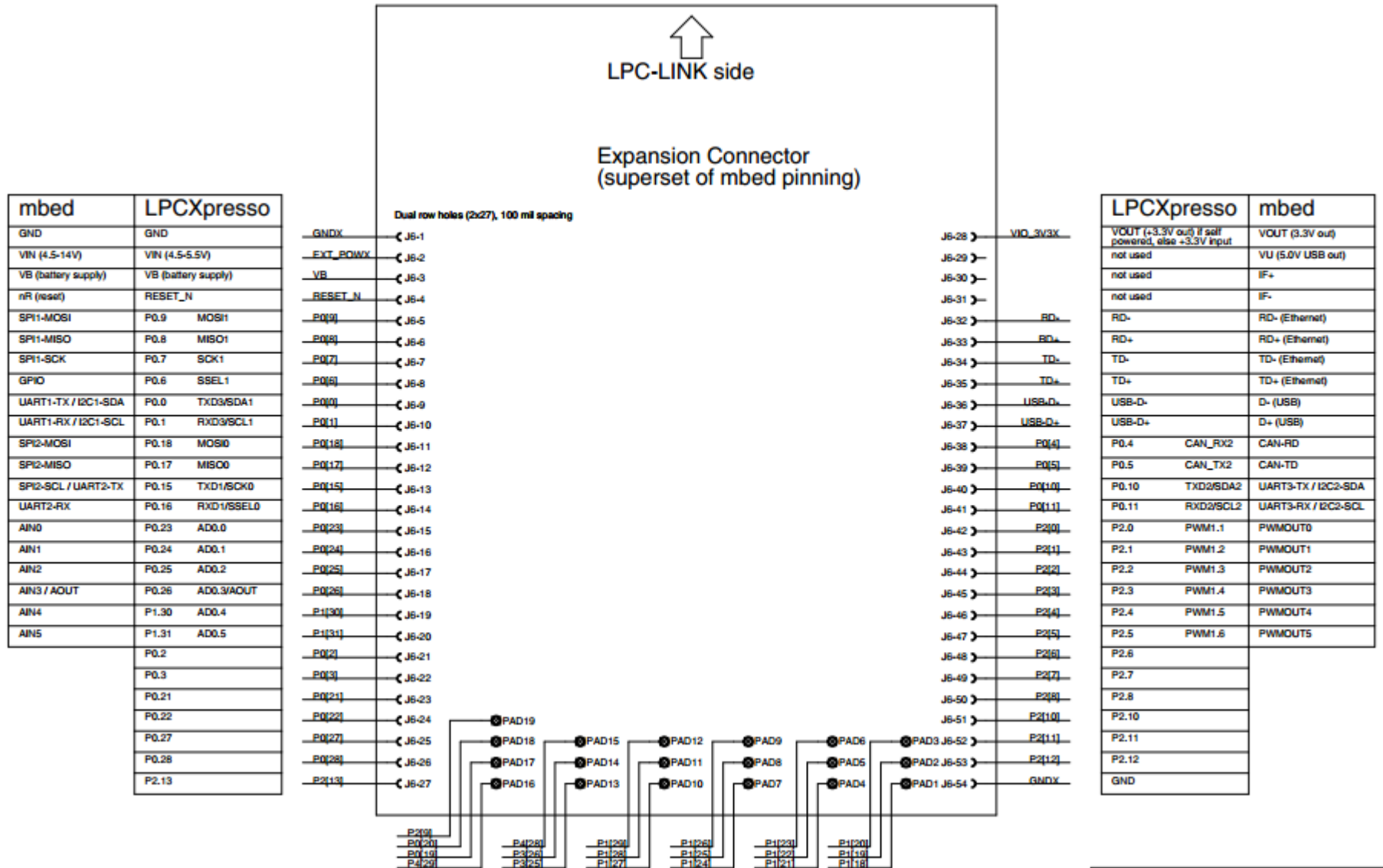


15) Led disponible en nuestra placa LPC1769 conectada en el pin 22 del puerto cero

LED



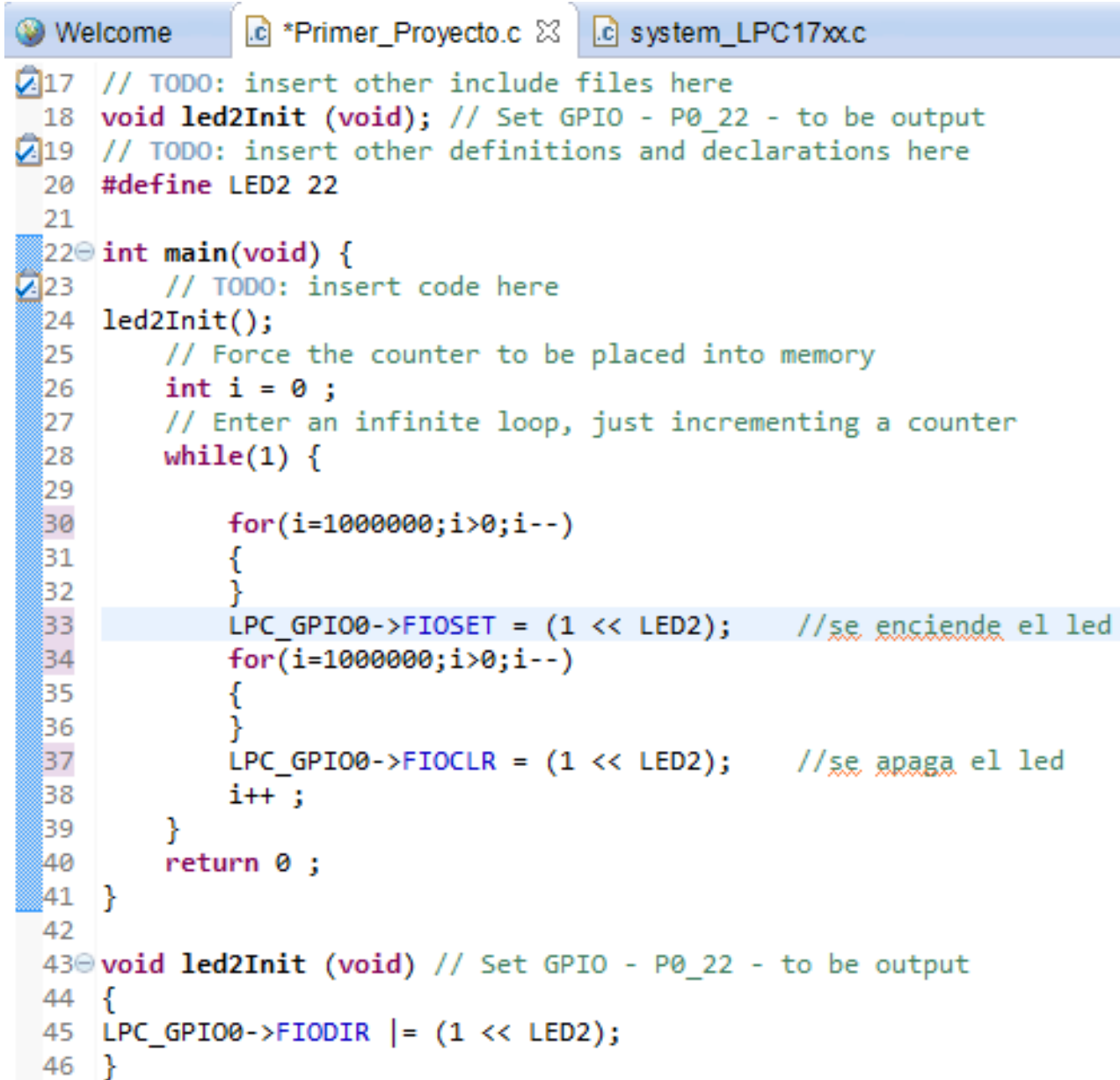
16) Disposición de pines de placa mbed y LPCXpresso.



17) Los registros que escribiremos son el FIODIR, FIOSET y FIOCLR de los cuales por ahora no se aclara más que la función de cada uno, FIODIR define la dirección del pin (si es entrada o salida), FIOSET pone en estado de '1' el pin al que se hace mención. FIOCLR pone en estado de '0' el pin al que se hace mención.

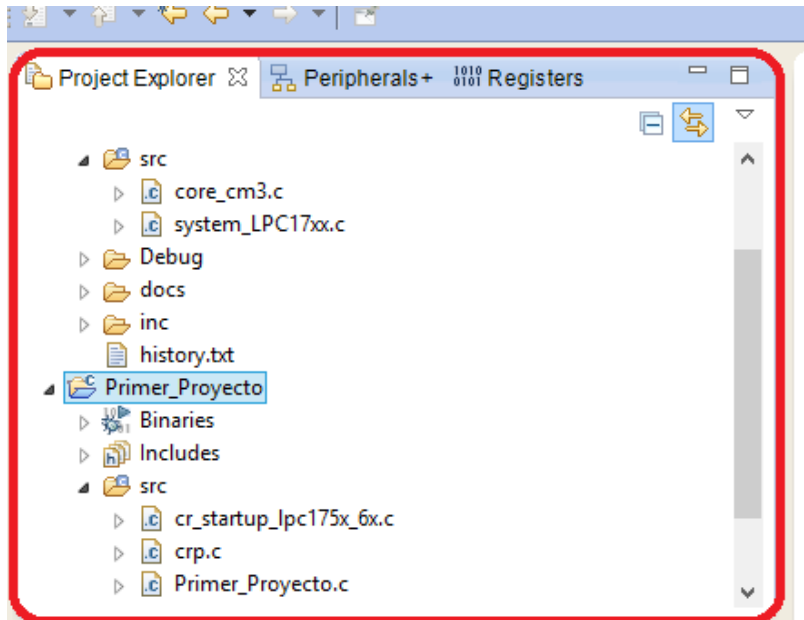
FIODIR	Fast GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0	FIO0DIR - 0x2009 C000 FIO1DIR - 0x2009 C020 FIO2DIR - 0x2009 C040 FIO3DIR - 0x2009 C060 FIO4DIR - 0x2009 C080
FIOSET	Fast Port Output Set register using FIOMASK. This register controls the state of output pins. Writing 1s produces highs at the corresponding port pins. Writing 0s has no effect. Reading this register returns the current contents of the port output register. Only bits enabled by 0 in FIOMASK can be altered.	R/W	0	FIO0SET - 0x2009 C018 FIO1SET - 0x2009 C038 FIO2SET - 0x2009 C058 FIO3SET - 0x2009 C078 FIO4SET - 0x2009 C098
FIOCLR	Fast Port Output Clear register using FIOMASK. This register controls the state of output pins. Writing 1s produces lows at the corresponding port pins. Writing 0s has no effect. Only bits enabled by 0 in FIOMASK can be altered.	WO	0	FIO0CLR - 0x2009 C01C FIO1CLR - 0x2009 C03C FIO2CLR - 0x2009 C05C FIO3CLR - 0x2009 C07C FIO4CLR - 0x2009 C09C

18) Modificaremos el código de modo que podamos encender y apagar un led de forma periódica haciendo un retardo por software

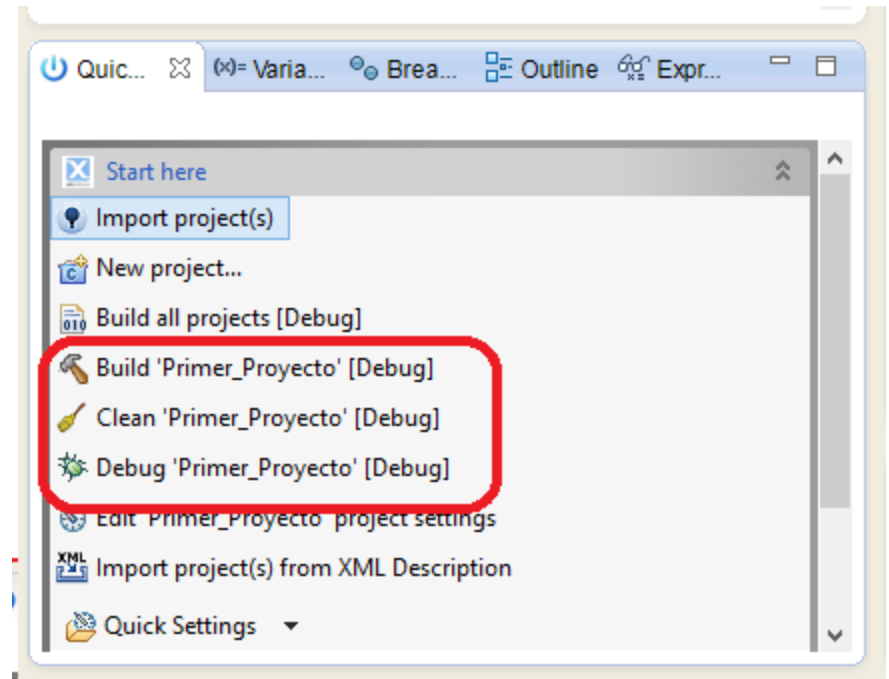


```
17 // TODO: insert other include files here
18 void led2Init (void); // Set GPIO - P0_22 - to be output
19 // TODO: insert other definitions and declarations here
20 #define LED2 22
21
22 int main(void) {
23     // TODO: insert code here
24     led2Init();
25     // Force the counter to be placed into memory
26     int i = 0 ;
27     // Enter an infinite loop, just incrementing a counter
28     while(1) {
29
30         for(i=1000000;i>0;i--)
31         {
32         }
33         LPC_GPIO0->FIOSET = (1 << LED2); //se enciende el led
34         for(i=1000000;i>0;i--)
35         {
36         }
37         LPC_GPIO0->FIOCLR = (1 << LED2); //se apaga el led
38         i++ ;
39     }
40     return 0 ;
41 }
42
43 void led2Init (void) // Set GPIO - P0_22 - to be output
44 {
45     LPC_GPIO0->FIODIR |= (1 << LED2);
46 }
```

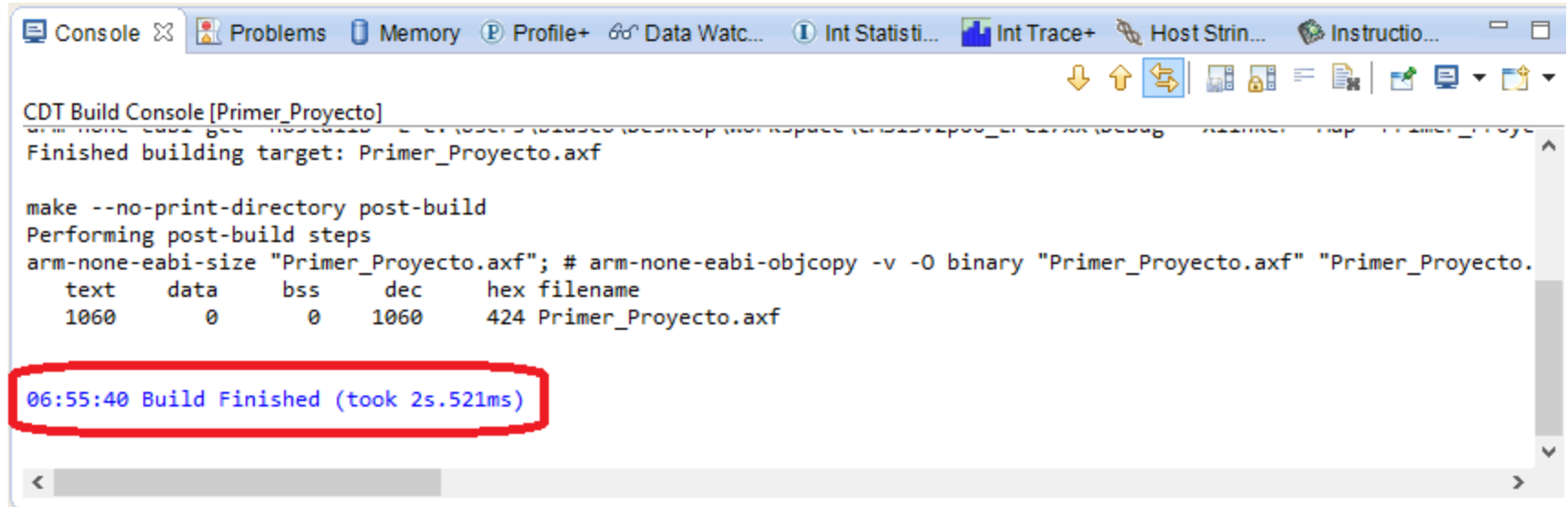
19) Una vez escrito nuestro código procedemos a compilarlo, para esto vamos al Project Explorer y seleccionamos nuestro proyecto como se ve en la siguiente imagen.



De esta manera nos aseguramos que en la ventana quickstart nos aparecen las opciones correspondientes a ese proyecto que nos figuran como
Build 'Primer_Proyecto'
Clean 'Primer_Proyecto'
Debug 'Primer_Proyecto'
Como se ve en la imagen a la derecha



20) Podremos ver en la consola si hubo algún problema de sintaxis o si falta agregar alguna librería o problemas de compilación en general.



The screenshot shows the CDT Build Console for a project named 'Primer_Proyecto'. The console output indicates that the build was successful. The output text is as follows:

```
CDT Build Console [Primer_Proyecto]
Finished building target: Primer_Proyecto.axf

make --no-print-directory post-build
Performing post-build steps
arm-none-eabi-size "Primer_Proyecto.axf"; # arm-none-eabi-objcopy -v -O binary "Primer_Proyecto.axf" "Primer_Proyecto.
text    data    bss    dec    hex filename
1060      0      0    1060    424 Primer_Proyecto.axf

06:55:40 Build Finished (took 2s.521ms)
```

The final line, '06:55:40 Build Finished (took 2s.521ms)', is highlighted with a red rectangle in the original image.

21) Una vez que el programa compila y no arroja ningún error procedemos al debug. Para esto vamos al Project Explorer y hacemos click en nuestro proyecto actual, que en nuestro caso es “Primer_Proyecto” y luego en la ventana QuickStart hacemos click en Debug ‘Primer_Proyecto’ tal como hicimos para compilar nuestro código.

22) Esto dispara el modo Debug y permite correr el programa en la placa, detenerlo, avanzar paso a paso, etc, al igual que como hacíamos las simulaciones en pic, solo que en este caso no se simula el avance del programa línea a línea sino que podemos hacer que el microcontrolador avance línea a línea, es decir podemos detener el microcontrolador en el punto del programa que necesitemos, lo que resulta una herramienta muy potente de debug. Para el debug tenemos herramientas, de las cuales podemos ver algunas en la siguiente imagen.



Se listan de Izquierda a derecha

- Ignorar breakpoint
- Run
- Pausa
- Stop Debug
- Desconectar
- Step Into
- Step Over

Step into: Avanza línea a línea, Si ocurre una llamada a una subrutina avanza paso a paso dentro de la subrutina

Step Over: avanza paso a paso si ocurre una llamada a subrutina no se detiene en cada línea de la subrutina

23) Vista del modo Debug

The screenshot displays the Keil uVision IDE in Debug mode. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, editing, and debugging.

The left sidebar shows the Project Explorer with the project 'Primer_Proyecto' expanded. Below it, the Peripherals list is visible, showing various hardware components like ADC, CANAFR, CANCON, and DAC. The Registers window is also open, showing the current state of the processor registers.

The main code editor displays the source code for 'Primer_Proyecto.c'. The code is in C and implements a simple LED blink using GPIO pins. The code is as follows:

```
1 // TODO: insert other include files here
2
3 void led2Init (void); // Set GPIO - P0_22 - to be output
4
5 // TODO: insert other definitions and declarations here
6 #define LED2 22
7
8
9
10 int main(void) {
11     // TODO: insert code here
12     led2Init();
13     // Force the counter to be placed into memory
14     int i = 0 ;
15     // Enter an infinite loop, just incrementing a counter
16     while(1) {
17         for(i=1000000;i>0;i--)
18         {
19             LPC_GPIO0->FIOSET = (1 << LED2); //se enciende el led
20             for(i=1000000;i>0;i--)
21             {
22             }
23             LPC_GPIO0->FIOCLR = (1 << LED2); //se apaga el led
24             i++;
25         }
26     }
27     return 0 ;
28 }
```

The code is written in C and implements a simple LED blink using GPIO pins. The code is as follows:

Pueden verse, si se desea, el estado de los periféricos, el estado de las variables, además de los registros de trabajo del CortexM3