# ERIKA Enterprise Tutorial

for the dsPIC (R) DSC platform

version: 1.1.11 January 18, 2011



#### About Evidence S.r.l.

Evidence is a spin-off company of the ReTiS Lab of the Scuola Superiore S. Anna, Pisa, Italy. We are experts in the domain of embedded and real-time systems with a deep knowledge of the design and specification of embedded SW. We keep providing significant advances in the state of the art of real-time analysis and multiprocessor scheduling. Our methodologies and tools aim at bringing innovative solutions for next-generation embedded systems architectures and designs, such as multiprocessor-on-a-chip, reconfigurable hardware, dynamic scheduling and much more!

#### Contact Info

Address: Evidence Srl, Via Carducci 64/A Località Ghezzano 56010 S.Giuliano Terme Pisa - Italy Tel: +39 050 991 1122, +39 050 991 1224 Fax: +39 050 991 0812, +39 050 991 0855

For more information on Evidence Products, please send an e-mail to the following address: info@evidence.eu.com. Other informations about the Evidence product line can be found at the Evidence web site at: http://www.evidence.eu.com.



This document is Copyright 2005-2010 Evidence S.r.l.

Information and images contained within this document are copyright and the property of Evidence S.r.l. All trademarks are hereby acknowledged to be the properties of their respective owners. The information, text and graphics contained in this document are provided for information purposes only by Evidence S.r.l. Evidence S.r.l. does not warrant the accuracy, or completeness of the information, text, and other items contained in this document. Matlab, Simulink, Mathworks are registered trademarks of Matworks Inc. Microsoft, Windows are registered trademarks of Microsoft Inc. Java is a registered trademark of Sun Microsystems. OSEK is a registered trademark of Siemens AG. The Microchip Name and Logo, and Microchip In Control are registered trademarks or trademarks of Microchip Technology Inc. in the USA. and other countries, and are used under license. All other trademarks used are properties of their respective owners. This document has been written using LaTeX and LyX.

## Contents

1	RT-Druid and Erika Enterprise tutorial for dsPIC (R) DSC	4
2	Note for Windows XP and Windows Vista users	5
3	Installing Erika Enterprise and RT-Druid on Microsoft Windows	6
4	First RT-Druid startup and configuration	9
5	Compiling your first Erika Enterprise demo for dsPIC (R) DSC	13
6	History	22

### 1 RT-Druid and Erika Enterprise tutorial for dsPIC (R) DSC

This small tutorial describes a set of steps needed to compile a simple application that shows the main features of Erika Enterprise and RT-Druid for the dsPIC (R) DSC platform.

This tutorial has been tested on a FLEX board produced by Evidence and Embedded Solutions and on a Microchip Explorer 16 development board from Microchip.

We presume that the reader is familiar with the MPLAB IDE debug environment provided by Microchip.

# 2 Note for Windows XP and Windows Vista users

If you are using Windows, and especially if you are using Windows Vista, please look carefully at the following warnings:

Warning: The Evidence package installation path should NOT contain directory name with spaces. For e.g. use c:/Evidence/Evidence

Warning: The Scilab package installation path should NOT contain directory name with spaces. For e.g. use c:/Evidence/scilab-4.1.2

Warning: If using Vista, be aware that directories like c:/Programmi, c:/Users/Documenti are not REAL directories but are aliases. DO NOT USE THEM. Put your RT-Druid workspace under c:/Users/yourusername/workspace.

Warning: Please install cygwin into its default directory, c:/cygwin.

Warning: Also if from the Windows Vista Explorer your Microchip compiler seems to be installed under c:/Programmi/Microchip/..., please remember to specify the REAL pathname. In particular, c:/Programmi DOES NOT EXISTS, whereas the correct name is c:/Program Files.

#### 3 Installing Erika Enterprise and RT-Druid on Microsoft Windows

This chapter will guide the developer on the installation procedure of Erika Enterprise and RT-Druid for the dsPIC (R) DSC platform.

The installation of Erika Enterprise and RT-Druid is composed of following packages:

- The Eclipse development environment, which is used by RT-Druid to provide the basic development environment for Erika Enterprise applications.
- The Eclipse environment is based on the Java platform, so that a working Java Runtime Environment must be present for using RT-Druid.
- The RT-Druid plugins, which provide the code generation for Erika Enterprise for Eclipse.
- The Erika Enterprise source code.
- The Microchip MPLAB IDE.
- The Microchip C30 Compiler.
- A version of the Microchip C30 compiler recompiled from the GCC sources, which enables basic C language compilation without the need to buy the full fledged Microchip C30 Compiler.
- A set of examples for the dsPIC (R) DSC Platform, which can be used to compile a first set of running examples for the Evidence/Embedded Solutions FLEX board, the Microchip Explorer 16 board, and others. These applications are organized in "templates", available at project creation.
- A subset of the Cygwin environment [1], including a set of utilities like make, gawk, and few others, which are used during the compilation process of an Erika Enterprise application.

To install the software, execute the following steps:

- 1. Install your favourite Java runtime environment, which is needed to run RT-Druid; as RT-Druid is a plugin of the Eclipse editor, which requires Java to be executed.
- 2. Install the latest version of the Microchip MPLAB IDE; you can use the default install directory. At the end of the install process, accept the system reboot.

- 3. Install the Microchip C30 Compiler, available from the Microchip web site. Even in this case, you can use the default install directory. When it is asked to change the default environment, please do accept.
- 4. Run the Erika Enterprise and RT-Druid installer.
- 5. The installer will prompt a list of packages which can be installed. Select all the packages you wish to install and continue the installation procedure (see Figure 3.1).

🕝 Erika Enterprise for dsPIC DS	iC Setup								
Choose Components Choose which features of Erika Enterprise for dsPIC DSC you want to install.									
Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.									
Select components to install:	✓ Crika Enterprise	Description Position your mouse over a component to see its description.							
Space required: 287.1MB									
Erika Enterprise for dsPIC DSC —	< Back	Install Cancel							

Figure 3.1: This screenshot shows the dialog box with the available install packages.

**Note:** The Erika Enterprise install package provides a version of the Microchip C30 compiler recompiled from the GCC sources made available by Microchip. Although that compiler is able to compile Erika Enterprise applications, it does not include Microchip include files and libraries which are distributed with the Microchip package only.

- 6. The installer will ask for a destination directory. If possible, please use c:/Evidence/Evidence (see Figure 3.2).
- 7. At this point, please check the *first* line of the file evidencedir\bin\mymake\_cygwin.bat (where evidencedir is the directory you chose during the installation). For example, if Cygwin is installed inside C:\cygwin, then the first line of the file should look like the following one:

@set EE\_BASH\_PATH=C:\cygwin\bin\bash

...that is, the line contains the correct path to the bash.exe file in your Cygwin installation. If you accepted the default settings, the correct pathname should be C:\cygwin\bin\bash as specified in the example before.

3 Installing Erika Enterprise and RT-Druid on Microsoft Windows

F	
🕞 Erika Enterprise for dsPIC DSC Setup	
Choose Install Location Choose the folder in which to install Erika Enterprise for dsPIC DSC.	
Setup will install Erika Enterprise for dsPIC DSC in the following folder. To install folder, click Browse and select another folder. Click Install to start the installatic	
Destination Folder	
C:\Evidence\Evidence Brow	wse
Space required: 110.8MB Space available: 8.4GB Erika Enterprise for dsPIC DSC <back install<="" td=""><td>Cancel</td></back>	Cancel

Figure 3.2: This screenshot shows the preferred destination dir for installing Erika Enterprise.

**Note:** We ask to perform this check because it seems that on some Windows machines the Cygwin installer does not correctly set the registry keys used by the Erika Enterprise installer.

The rest of this tutorial assumes that the Microchip MPLAB IDE is installed within the C:\Programmi\Microchip directory and that, consequently, the GNU Assembler for dsPIC (R) DSC is installed within C:\Programmi\Microchip\MPLAB ASM30 Suite\bin. Please note that these values may be different from the settings you have chosen on your machine. Please also read the chapter with the Windows Vista recommendations.

After all the required packages have been installed, you are ready to start **RT-Druid** for the first time.

Please follow the next steps:

- 1. As the first step, run the Eclipse IDE from the Evidence menu inside the Start menu of your Windows machine, choosing Start/Programs/Evidence/RT-Druid.
- A dialog box will appear, prompting to choose the right workspace (see Figure 4.1). Leave the default workpackage directory as it is, and proceed by pressing "OK".

😂 Workspace	: Launcher		×
Select a w	orkspace		
	orm stores your projects in a folder called a workspace. orkspace folder to use for this session.		
Workspace:	C:\Users\pj\workspace	<b>•</b>	Browse
🕅 l ise this a	s the default and do not ask again		
E ose uns a	s ar arian an ao no os again	ОК	Cancel

Figure 4.1: This screenshot shows the dialog box for the choice of the current workspace directory.

Warning: The workspace pathname MUST NOT contain any blank space, otherwise Erika Enterprise and RT-Druid may not work properly.

**Note:** If you are using Windows Vista, then the workspace directory c:/Users/<username>/workspace works.

- 3. The Eclipse Welcome screen appears, like in Figure 4.2.
- 4. Before being able to correctly build your application, you should set the path to the Microchip C30 compiler and the MPLAB ASM30 assembler programs. For doing so, please go to the "Preference" menu, as shown in Figure 4.3, and find the "RT-Druid/Oil/PIC30 Configurator" form as depicted in Figure 4.4. The first



Figure 4.2: The Eclipse Welcome screen.

textbox, labeled Gcc path, refers to the installation directory of the Microchip C30 compiler. The second textbox, labeled Asm path, refers to the installation directory of the ASM30 assembler provided with the MPLAB IDE.

Warning: The install directories specified in the two textboxes of Figure 4.4 does *not* include the bin directory!

That is, c:\Programmi\Microchip\MPLAB C30 is correct, whereas c:\Programmi\Microchip\MPLAB C30\bin is not.

**Warning:** The install directory of the assembler refers to the assembler provided with MPLAB IDE and *not* the assembler provided with the C30 compiler. The reason is that the directory is used to call the assembler and *also* to copy the crt0.s file, which has a different position in the two assembler distributions made by Microchip.

Warning: If you are using a Student Editon of the Microchip C30 compiler with an expired license, please check the "Use EE gcc to resolve dependencies" checkbox in Figure 4.4.

C/C+++Eclipse Platform File Edit Refactor Navigate Search Run	Project (MG	odawi biala				1	- 0 🗱
11 - 回合 計・日合 計・引・中心・	id • (		\$ • 0 • <b>Q</b> •	🍅 🇀 🛷 🔹   5		12 (12 c/c	••
Project Explorer 23		Open Perspective   Show View				E Outli 🛛 🧕	Mak 🔍 🗆
		Customize Perspective Save Perspective As Reset Perspective Close Perspective Close All Perspectives			A	An outline is not av	ailable.
		Navigation +					
		Preferences					
	🚺 Problems 🖇	🕄 🤕 Tasks 🔲 Console 🗔 Pr	operties				\$ ~ <b>.</b>
	ntems Description	*	Resource	Path	Locatio	on Type	
0 items selected				- Fi		(a)	2 🛛 💙 🔶

Figure 4.3: Go to the "Preference" menu.

Figure 4.4: Select paths for compiler and assembler.

5. Before creating and building your application, please deselect the "Build Automatically" flag inside the "Project" menu, as shown in Figure 4.5.

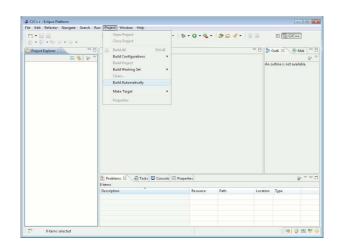


Figure 4.5: Deselect the "Build Automatically" flag in the "Project" menu.

You are now ready to compile your first Erika Enterprise demo. Please execute the following steps:

1. Please select "New Project", then "RT-Druid Oil and c/c++ Project" from the "File menu", as in Figure 5.1.

	New A	Alt+Shift+N ►	C++	C Project	1 - 0	🧆 😂 🛷 👻 🗐		😭 📴 C/C+	•
	Open File			C++ Project		100 1 100	liter.		<u> </u>
	Close	Ctrl+W	4	RT-Druid Oil and C/C++ Project	RT-Druid Oil and C/C++ Project	- 01	🗆 🗄 Outli 🛛 💿 Mak 👘		
	Close All Cl	trl+Shift+W	E3	Project					29 <sup>1</sup>
	Save Save As	Ctrl+S	63	Convert to a C/C++ Make Project Source Folder			Ā	In outline is not ava	
	Save All Ctrl+Shift+S Revert F2 Move Rename F2 Refresh Convert Line Delimiters To F Print Ctrl+P		200	Folder Source File					
<u> </u>			6	Header File File from Template Class					
80			<b>F</b> 9	Other Ctrl+N					
6									
	Switch Workspace Restart	•							
	Import Export								
	Properties	Alt+Enter							
	Exit								
_		💽 Pr		ns 🕸 🧑 Tasks 📮 Console 🔲 Prop	erties				<u>ا ا م</u>
		0 item		*					
		Desc	nptio	1	Resource	Path	Locatio	in Type	

Figure 5.1: Select "New project" from the "File" menu.

- 2. A Dialog box appear. Please select a template for the new project, as in Figure 5.2.
- 3. Press "Next".
- 4. Insert the name of the new project. Please type taskdemo (you can choose other names of course). Please see Figure 5.3. Press the "Finish" button.

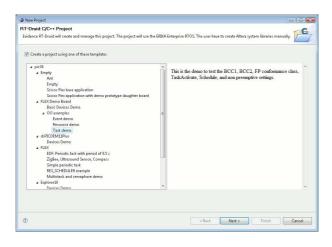


Figure 5.2: Select a template for your project.

New Project		
T-Druid C/C++ Project		G
Evidence RT-Druid will create and manage this project. The project will use the ERIKA Enterprise R	RTOS. The user have to create Altera system libraries manually.	
Project name: taskdemo		
V Use default location		
	Browse.	
2	<back next=""> Finish Cance</back>	el

Figure 5.3: Type a name for the new project.

- 5. We are now ready to build the demo. Right click on the project name in the Eclipse navigation bar, and choose "Build Project"<sup>1</sup> (see Figure 5.4).
- 6. Then, the compilation process starts as depicted in Figure 5.5. Please note the message that appears when the compilation is successfull.

Note: If the error depicted in Figure 5.6 appears (means that mymake\_cygwin.bat is unable to find a file), then please follow the instructions at the last point of Chapter 3.

<sup>&</sup>lt;sup>1</sup> "Build Project" only appears if the "Build Automatically" flag is not selected in the "Project" menu.

<mark>13 • 묘 손</mark> 성 • 원 •		) ≱ • ⇔ •	<b>·</b> •	<b>% • ⊗ • ‡</b>	• • • •	001	•		C/C++
Project Explo	irer	S Conf.oil						E Outli 🛛	Mak
		E 🙀 🔝 🍸 1/*					*		1
a 😂 taskden	~		terpri	se - a tiny RTOS	5 for small	microcontro	ilers	An outline is no	it available.
> C code		New	,	002-2008 Evide	ence Srl		E		
0 cor		Open in New Window							
	(PA)	Copy	Ctrl+C	ct of ERIKA Ent	cerprise.				
	-	Paste	Ctrl+V	e is free soft	are: you c	an radistrik	inte it		
		Delete	Delete	t under the ter					
		Remove from Context Ctrl+Alt+Shift		olished by the			on,		
		Move	TDOWN	exception desc	cribed belo	α).			
		Bename	F7	de statically (	or dvnamica	lly with oth	er mor		
		itename	12	ed work based o					
	<u>eès</u>	Import		he GNU General	Public Lic	ense cover t	he who		
	4	Export							
	Build Project Clean Project (2) Refresh	Build Project		ception, the co					
		Clean Project		ink this code a					
		Refresh	FS	ardless of the copy and dist					
		Close Project		hoice, provided					
		Close Unrelated Projects		ule, the terms					
		Build Configurations		s library. If					
		Make targets		to your version					
		Index		m			•		
				Console 🔲 Proper	ties				<b>\$</b> 2 ♥
		Run As	,						
		Debug As	,	-	Resource	Path	Locat	tion Type	
		Profile As	,						
		Team							
		Compare With	,						
		Restore from Local History							
		Properties All	t+Enter						

Figure 5.4: We are now able to build the project.

- 7. At the end of the compiling process you will be able to find a file named 'pic30.cof inside the Debug directory inside the project, as shown in Figure 5.7.
- 8. You are now ready to import the produced COFF file inside Microchip MPLAB IDE. To do that, open MPLAB IDE as in Figure 5.8.
- 9. Choose "Import..." from the "File" menu, as in Figure 5.9.
- 10. A dialog box appear. Please select the pic30.cof file that has been produced by the compilation process in Eclipse, as shown in Figure 5.10. You can find that file inside the Eclipse workspace you selected at the beginning in Figure 4.1. In this example, the file is stored inside the directory c:\Programmi\Evidence\eclipse\workspace\pic30\_oo\_mono\Debug.
- 11. You have now imported the COFF file inside MPLAB IDE. There is no need to create a MPLAB IDE Project, because the compilation process is handled by Eclipse. Figure 5.11 shows the "Disassembly Listing" and the "Program Memory" window. Please note that MPLAB IDE correctly recognizes the debug symbols of the source code produced inside Eclipse.
- 12. You can now start debugging the demo application using MPLAB IDE.

Figure 5.12 shows the Explorer 16 board running pic30\explorer16\Devices Demo demo application, which uses the Explorer 16 onboard devices to monitor and display the environment temperature.

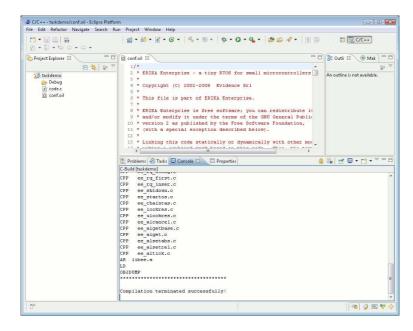


Figure 5.5: The compilation process.

```
Note: If you get an MPLAB IDE error like the following:
ICDWarn0015: Program memory has changed since
last program operation?
Continue with Debug operation?
Running Target
ICD0083: Debug: Unable to enter debug mode.
Please double click this
message for more information.
```

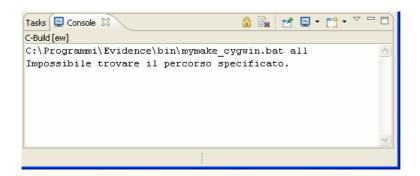


Figure 5.6: An error that shows up on some Windows machines. Please check the mymake\_cygwin.bat file as explained in the last point of Chapter 3.

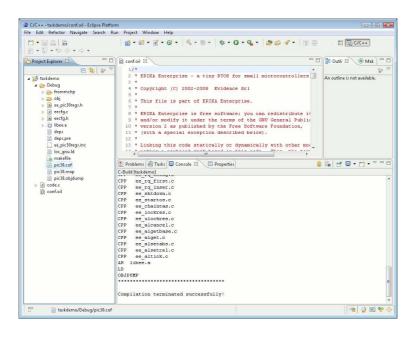


Figure 5.7: The output file is ready to be programmed on the target board.

Please make sure that you have entered the debug mode and programmed the device *from the Debugger Mode* and not from the Programmer Mode.

**Note:** If you are using a FLEX board, please remember to set the device correctly uinder the "Configure / Select device..." menu of MPLABIDE.

The correct settings for the dsPIC on the FLEX Full and the FLEX Light is shown in Figure 5.13. The correct settings for the PIC18 on the FLEX Full is shown in Figure 5.14.

MPLA8 IDE v8.00			
File Edit View Project Debugger Programm		Checksum: 0×039c	
I Uehted W ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )			
💽 Output			
Build Version Control Find in Files			
dsPIC33FJ256M	C710 oab sab IP0 dc n	ovzc	

Figure 5.8: The Microchip MPLAB IDE.

New	Ctrl+N	- 💣 🖨 🖬 🚯	Checksum: 0×039c	
Add New File to Project				
Open	Ctrl+O			
Close				
Save	Ctrl+S			
Save As				
Save All				
Dpen Workspace				
Save Workspace				
Save Workspace As				
Close Workspace				
mport				
ixport				
Print	Ctrl+P			
Recent Files	•			
Recent Workspaces	•			
xit				
Build Version Control Fin	nd in Files			

Figure 5.9: Choose "Import..." from the "File" menu to import the coff file produced in Eclipse.

MPLAB IDE V8.00	
File       Edit. View       Project       Debugger       Programmer       Tools:       Configure       Window       Help         D </th <th></th>	
Cerca in: 🔒 Debug 👻 😨 🐨	
Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbols       Image: Symbols     Image: Symbols     Image: Symbols     Image: Symbo	<u></u>
Nome file: pic30.cof - Apri	
Tipo file: All Load Files (",hex,",co(,",cod,",ell) 🗸 Annula	
dsPIC336/256MC710 oab seb IP0 dc n ovz.c	

Figure 5.10: Select the COFF file you want to import.

Unded W	) 🚅 🖬   🐰 🖬 (	🛢 🖨 🗛 🇀 🤋 🗍 🗌		Checksum: 0×768d	G (5 da II a
179:       /* Tarki: just call the ChristmasTree */         00315       A8042       beet.b 0x002.97         180:       TARK (Task1)         00320       A8042       beet.b 0x002.97         100:       A8042       beet.b 0x002.95         100:       A8042       beet.b 0x002.95         100:       A8042       beet.b 0x002.95         100:       A8042       beet.b 0x002.95         100:       Task1 (Tred+1)       0x000         00394       821999       lack.b (Tred+1)         00394       821999       lack.b (Tred+1)         00394       00036       lack.i (Tred+1)         00394       00010       mev.m 0x00000         00394       000392       lack.i (Tred+1)         00395       00010       mev.m 0x01,00000         00394       000300       mex.m 0x0,00000         00395       000000       mex.m 0x0,00000         00385       000000       mex.m 0x0,000000         00385 <td>Untitled W</td> <td></td> <td></td> <td></td> <td></td>	Untitled W				
Image: 179:       /* Task: just call the DrintmasTree */         00011: ASDC4:       betch: boock2; #         00012: CO200       mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */         10: mov.m (rotat); for the christmasTree */       10: mov.m (rotat); for the christmasTree */		Disassembly Listing			
Image: Solution of the set of t		178:			*
Image: 100:         TAR(Task)           00022: 02080         mov.w for2d, 56           00022: 02080         mov.w for2d, 50000           00024: 02080         mov.w for2d, 50000           00034: 02080         mov.w for2d, 50000           00038: 0707CC         mov.w for2d, 50000           00038: 00000         mov.w for2d, 50000           00038: 000000         mov.w for2d, 500000		179:	/* Taskl: just call the Chri	stmasTree */	
For         00320         ABC042         best.b 000022,66           00312         828400         mev.w 502000         b00005           101         00322         828420         betr.b 00000,00056           102         00322         828422         betr.b 00002,265           103         00322         828422         betr.b 00002,265           103         105         00322         828422           103         105         106         10702           105         104         1083         108           105         104         1083         108           105         104         108000         100398           106         00398         00787C         reall 600000           00385         00000         mov = 8004,00000         000000           00385         10000         mov = 8004,00000         00000           00385         20000         mov = 8004,00000         00000           00385         1080         reall 0x00036         109           109:         /* CONFICIENTUS         101         101           109:         stdet mrXCHENUS         101         101           109:         /* CONFICIENTUS		0031E A8E042	bset.b 0x0042,#7		
I Output         00312 202000         mov.w 80024,00000           00324 82030         mov.w 80024,000056         181           101         00324 82090         mov.w 80024,000056           00324 82090         mov.w 80024,80         181           103         0034 822990         mov.w 80024,000056           0034 822990         mov.w 80024,00006         184           0034 822990         mov.w 8002,00000         00380           0038 200010         mov.w 8004,00000         00380           0038 200000         mov.w 8004,00000         00380           0038 200000         mov.w 8004,00000         00380           0038 200000         mov.w 8004,000000         00380           130:         stifed WSON036         stifed WSON036           130:         st		180:	TASK(Task1)		
■ 00374 898.00         mov.w 0x0000_000950           181         (           00322 ASA42         blx.b 0x002, g5           1052         (           1053         (           1054         (           1055         (           1051         (           1051         (           1051         (           1051         (           1051         (           1051         (           1051         (           1052         (           00395 00700         (           00395 007000         mov.w 1001, (           00395 007000         mov.w 1001, (           00395 007000         mov.w 1002, (           1050         fididi (		00320 A8C042	bset.b 0x0042,#6		
Image: 108:         108:           00022         2.80042           108:         taskl fired+t;           00034         K2599           108:         /* first half of the christmas tree */           108:         /* first half of the christmas tree */           108:         /* first half of the christmas tree */           108:         144_blink(0x01);           00036         20000           0037         144_blink(0x02);           0038         20000           00395         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         00000           00385         fondit           199:         fifted mitstast tree */		003E2 202D80	mow.w #0x2d8,0x0000		
Prio:         0.0322 ASA042         betxb 0:00024,85           100:         0.034 KC299         inc.v9 C0099           101:         inc.v9 C0099         inc.v9 C0099           103:         1.41 f of the christmas tree "/           104:         /* First half of the christmas tree "/           104:         /* First half of the christmas tree "/           104:         /* First half of the christmas tree "/           105:         Teal inth(000);           106:         0.0398 (0000           00398 (0000         mov.vs f004, 00000           00398 (00000         senditional Schedule() here! */           109:         fide fifSchedule() f           109:         fide fifSchedule() here! */           109:         fide fifSchedule() here! */ <tr< td=""><td></td><td>003E4 884AE0</td><td>mov.w 0x0000,0x095c</td><td></td><td></td></tr<>		003E4 884AE0	mov.w 0x0000,0x095c		
Image: caskl_fired+i;           00394 8C3998           100:           101:           102:           103:           104:           105:           106:           107:           108:           00394 20000           00395 20000           00395 20000           00396 00000           00397 00000           00392 000000           00392 0000		101:	(		
Image: The system         0.0094 KC2999         imw 0.00998           104:         /* First half of the christmas tree */         104:           104:         /* First half of the christmas tree */         104:           105:         00010         weight (0001);           00099 07770         reall 0:000036           106:         104         leight (0001);           100:         100000         mev.* 0000;00000           00039:         0070700         reall 0:000036           107:         104_blink(0001);         104           108:         0077FC         reall 0:000036           109:         /* CONFIGURATION 7 and 4: we put an additional Schedule() here! */           100:         #identific (1)           100:<		00322 A9A042	bclr.b 0x0042,#5		
<pre>103: 103: 104: 105: 105: 105: 105: 105: 105: 105: 105</pre>		182:	taskl_fired++;		
Ima:         /* First half of the christmas tree */           185:         00395 20010           00395 20010         mov. # Dial, 00000           00396 20010         mov. # Dial, 00000           00397 Carl Log0038         mov. # Dial, 00000           00304 0797C4         reall 0x00036           187.         1e4_blink(0x04);           00304 0797C4         reall 0x00036           187.         1e4_blink(0x04);           00304 0797C4         reall 0x00036           188         finde Wrator           190.         fifdet Wrotnow           191.         fifdet Wrotnow           192.         fendstil           193.         fischedule()           193.         fischedule()           193.         /* Scondul hall of the christmas tree */           193.         /* Scondul hall of the christmas tree */           194.         /* Scondul hall of the christmas tree */           195.         reall 0x000036           196.         reall 0x000036           197.         reall 0x000036           197.			inc.w 0x0998		
Firs         105;         1ed_blink(loc0);           00399         07701         rcall 0x000038           165         1ed_blink(loc0);           00394         000780           105         1ed_blink(loc0);           00394         000780           105         1ed_blink(loc0);           00394         00000           00395         1ed_blink(loc0);           00394         00000           00395         1ed_blink(loc0);           00394         00000           00394         00000           00394         000000           00394         000000           00394         000000           00394         result 0x000036           100;         #iddet Weinon           101;         #iddet Weinon           102;         #endit();           103;         #endit();           104;         /* Second half of the christmas tree */           105;         1ed_blink(0x00);           1034         /* Second half of the christmas tree */           105;         1ed_blink(0x10);           1034         000000           1034         000000           1035;         1ed_blink(0		183:			
Free         Syma         00398         200010         mev. * Doil, 00000           00398         00398         1065         1e4_bitk(0x00);           166:         1e4_bitk(0x01);         000395           00395         00707C1         reall 0x000386           107         1e4_bitk(0x01);         000395           107         1e4_bitk(0x01);         000395           100395         200040         mov.* B0x4,0x0000           000385         200040         mov.* B0x4,0x0000           000385         200040         mov.* B0x4,0x0000           000385         200040         mov.* B0x4,0x0000           000385         200040         mov.* B0x4,0x0000           108:         /* CONTORNATION 3 and 4: ve put an additional Schedule() here! */           139:         Schedule():         192:           139:         fendit():         193:           139:         1e4_bitk(0x00);         00034           000342         000000         00034           000342         000000         mov.* B0x4,00000           000342         000000         mov.* B0x4,00000           000342         000000         mov.* B0x4,00000           000344         007000         mov.* B0x4,0		184:	/* First half of the chris	thas tree */	
Bad         O0039 0797C%         rcall 0x00036           106::         lebink(coc2);           00034 200220         mov. # 50x2,00000           00035 200200         mov. # 50x4,00000           00038 200200         mov. # 50x4,00000           00038 200200         mov. # 50x4,00000           00038 200200         fstdet myCHDDL           100:         fstdet myCHDDL           100:         fstdet myCHDDL           100:         fstdet myCHDDL           101:         fstdet myCHDDL           102:         fstdet myCHDDL           103:         fstdet myCHDDL           104:         /* Second half of the christmas tree */           105:         led_blink(coc0);           00324 200000         00000           00324 200000         mov. * s0x0,00000           00324 200000         mov. * s0x0,000000		185:	<pre>led_blink(0x01);</pre>		
Bad         O0039 0797C%         rcall 0x00036           106::         lebink(coc2);           00034 200220         mov. # 50x2,00000           00035 200200         mov. # 50x4,00000           00038 200200         mov. # 50x4,00000           00038 200200         mov. # 50x4,00000           00038 200200         fstdet myCHDDL           100:         fstdet myCHDDL           100:         fstdet myCHDDL           100:         fstdet myCHDDL           101:         fstdet myCHDDL           102:         fstdet myCHDDL           103:         fstdet myCHDDL           104:         /* Second half of the christmas tree */           105:         led_blink(coc0);           00324 200000         00000           00324 200000         mov. * s0x0,00000           00324 200000         mov. * s0x0,000000	Files C Sym	00396 200010	mow.w #0x1,0x0000		
I Output         00384_200200         mov.**00.2,00000           100352_20000         mov.**00.2,00000         00300_079FCC         reall 0x00036           Badd Version         5100_079FCC         reall 0x00036         00300_079FCC         reall 0x00036           Badd Version         5100_079FCC         reall 0x00036         00300_079FCC         reall 0x00036           130:         5100_0715(07.3700.3 and 4: we put an additional Schedule() here! */         190:         190:           130:         5100_0715(07.3700.3 and 4: we put an additional Schedule() here! */         190:         190:           130:         5100_0715(07.3700.3 and 4: we put an additional Schedule() here! */         190:         190:           130:         5100_0715(07.3700.3 and 4: we put an additional Schedule() here! */         190:         190:           130:         5100_0715(07.3700.3 and 4: we put an additional Schedule() here! */         190:         190:           130:         140_1100.0000.00000         003.400000.00000         003.400000.00000         003.400000.00000           00322_200000         mov.ve 00.00.000000         003.400000.000000         003.400000.000000         003.400000.000000           00324_2000000         00324_00000000         00324_00000000         003.400000000         003.400000000		00398 07FFCE	rcall 0x000336		
I Output         0038C 07FFCC reall 0x000366           100:         10 0x100 0x0000           Budd Version         0038C 100000           Budd Version         51 0x0000           Budd Version         51 0x0000           100:         100:           130:         7 CONFLOURATION 3 and 4: we put an additional 5chedule() here! */           130:         51 0x0000           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           130:         100:           131:         /* Scood half of the christmas tree */           135:         100:           136:         100:           137:         100:           138:         10:           139:         10:           139:         10:           139:         10:           139: <t< td=""><td></td><td>186:</td><td></td><td></td><td></td></t<>		186:			
1001put         1007;         1e4_blink(0s04);           003A0         07976, v S0A4_00000         003A0           100         100;         v S0A4_00000           100;         fifdef WFCMEDULE         and 4; ve put an additional Schedule() here! */           100;         fifdef WFCMEDULE         100;           100;         reall 0:0000;         00342         00000;           100;         mov.w 50:00; 0:0000;         00345;         00000;           00345; 2000;00         mov.w 50:00; 0:0000;         00345;         0:0000;					
Bad         Venion         00038         200040         mov.* 5004,00000           Bad         Venion         0         0         0.019707.4         rcsll 0.0200326           Bad         Venion         fs: dc4         1% CONFICUENTIAL TON 1 and 4: we put an additional Schedule() here! */           100:         fs: dc4         1% CONFICUENTIAL TON 1 and 4: we put an additional Schedule() here! */           101:         fs: dc4         1% CONFICUENTIAL TON 1 and 4: we put an additional Schedule() here! */           103:         fs: dc4         1% fs: dc4           103: <td></td> <td></td> <td></td> <td></td> <td></td>					
I Output         OOSA0 07FFCA         reall 0:000336           Buid Vemion         199:         /* CONFIDURATION 3 and 4: we put an additional Schedule() here! */           190:         ficket MYSCREDUCATION 3 and 4: we put an additional Schedule() here! */           191:         Schedule();           192:         fendit           193:         Schedule();           194:         /* Second half of the christmas tree */           195:         1e4_blink(log09);           00324 200000         mov.w s0s0, 00000           00324 200000         mov.w s0s0, 00000           00324 201000         mov.w s0s0, 00000           00324 201000         mov.w s0s0, 00000           00324 20100         mov.w s0s0, 00000           00324 20100         mov.w s0s0, 00000           00324 20100         mov.w s0s0, 00000					
Looput         100:           Baid Version         109:           100:         fieldstructure					
Bud         Venion         100: 100: 100: 100: 100: 100: 100: 100:	Cutput		rcall 0x000336		
190         fifdef WYGREPOLE           191:         5 Checkle(1)           192:         fendit           190:         /* Scond half of the christmas tree */           191:         10 Checkle(1)           192:         isolation           193:         10 Checkle(1)           194:         10 Checkle(1)           195:         10 Checkle(1)           196:         10 Checkle(1)           197:         10 Checkle(1)           198:         10 Checkle(1)           198:         10 Checkle(1)           198:         10 Checkle(1)           198:         10 Checkle(1)           199:         10 Checkle(1)					
191: Schodule(); 192: fendf 193: /* Second half of the christmas tree */ 194: /* Second half of the christmas tree */ 195: led blink(Gro0); 003A4 07FC0 reall 0x00036 106: led blink(0x10); 003A4 20100 mov x 50x10,00000 003A6 20100 mov x 50x10,00000 003A6 20100 mov x 50x10,00000	Build Version			we put an additional Schedule() her	iel */
192: fendit 193: 194: /* Second half of the christmas tree */ 195: 00342 200800 mov w for0.0c0000 00344 07FE00 real 0x000026 196: Led Disk(0x10)/ 00346 200100 mov.w 50210,0x0000 00346 07FE00 real.0x000036					
199: 194: /* Second half of the christmas tree */ 195: 1ed_blink(log08); 00324 20000 mov.w gbs0,00000 00334 079708 rcall 0x00036 16: 1ed_blink(log10); 00326 200100 mov.w gbs0,100000 00328 079760 rcall 0x00036					
134: /* Second haif of the christmas tree */ 135: 00342 200800 mov w 5040,0x0000 00344 078Fcs reall 0x00036 136: 144_bisk(0x10); 00345 200100 mov.w 5040,0x000 00348 078Fc5 reall 0x00036			#endif		
195: 1ed_blink(0x00); 00324_200000 mev: #500,00000 00324_079765 reall_0x00036 186: 1ed_blink(0x10); 00324_200100 mev: #500.000000 00326_200100 mev: #500.000000					
00342 200080 mev. w 20.8, 00000 00344 079F68 reall 0000336 1966 led_blank(01.0) 00348 079F68 reall 0000356				stmas tree */	
0034 07976 real 0x000306 196: 1e4pink (0x00) 003& 200100 www.w 80x10,0x0000 003& 07976 real 0x00036					
196: 1ed_blink(0x10); 0034& 800100 mov # 0510(000000 033& 079766 real_0 0000365					
003&6 200.00 mov. w 50x10,0x0000 003&8 07FFC6 real 0x00336					
003A8 07FFC6 kcall 0x000336					
107. Ved 514-5 (0.07)					
					+
	1	m			· · ·

Figure 5.11: Debug symbols are correctly recognized.



Figure 5.12: The Explorer 16 board with the running demo program.

Select Device						
Device: Device dsPI033FJ256MC710    ALL	Family:					
Microch	ip Tool Support					
Programmers						
PICSTART Plus     MPLAB ICD     PRO MATE II     PICkit 1     MPLAB PM3     MPLAB REA	-					
Language and Design Tools						
ASSEMBLER OCOMPILER v2.02 v2.02	O VDI					
Debuggers						
<ul> <li>MPLAB SIM</li> <li>MPLAB ICD</li> <li>MPLAB REAL ICE</li> </ul>	2 🥝 PICkit 2					
MPLABICE 2000 MPL	ABICE 4000 ICE/ICD Headers					
No Module     No Module     No Module						
OK Cancel Help						

Figure 5.13: Selecting the dsPIC MCU mounted on the  $\mathsf{FLEX}$  boards.



Figure 5.14: Selecting the PIC18 MCU mounted on the FLEX Full boards.

## 6 History

Version	Comment	
1.0.0	Initial revision.	
1.1.0	Updated with information and screenshots about Erika Enterprise	
	version 1.4.	
1.1.3	Updated with new screenshots about configuration; added warn-	
	ings about blanks in pathnames.	
1.1.4	Added some changes to the install instructions from comments by	
	Simone Mannori.	
1.1.5	Added Cygwin error workaround when the installer can not find	
	the cygwin registry value.	
1.1.6	Typos found when translating the document in italian.	
1.1.7	Updated some figures Figure, added a few warning boxes.	
1.1.8	Added notes about MPLAB IDE.	
1.1.9	Added recommendations for Windows Vista.	
1.1.10	Updated screenshots for EE 1.4.3. Erika Enterprise Basic renamed	
	to Erika Enterprise.	
1.1.11	Revised entire document.	

# Bibliography

[1] Cygwin. The cygwin project. http://www.cygwin.com, 2005.