Ghid de utilizare a mediului de dezvoltare Code Composer Studio v.6 pentru MSP430

1. Instalare

Cea mai recentă versiune a mediului de dezvoltare Code Composer Studio (CCS) poate fi descarcată de pe site-ul producatorului – Texas Instruments (TI) – de la adresa: <u>http://www.ti.com/tool/ccstudio</u>. În cele ce urmează se va discuta despre versiunile 6.x.

Se recomandă dezactivarea aplicației antivirus (dacă există pe sistemul pe care lucrați) pe durata instalării în special dacă aceasta exercită un control excesiv asupra proceselor executate. Lansați kitul de instalare descarcat pentru a porni procesul de instalare și urmați pașii inițiali privind acceptarea termenilor de utilizare și selecția locației de instalare.

În pasul dedicat selecției produselor ce urmează a fi instalate alegeți produsele destinate familiei MSP430. Figura următoare ilustrează selecția produselor necesare în acest pas.

Code Composer Studio v6 Setup Processor Support Select Product Families to be installed.	×
 MSP Ultra Low Power MCUs MSP430 Ultra Low Power MCUs MSP432 Ultra Low Power MCUs TI MSP430 Compiler TI ARM Compiler GCC ARM Compiler GCC ARM Compiler SimpleLink Wireless MCUs Sitara 32-bit ARM Processors Media Processors Single Core DSPs Multi Core Processors 	 Description MSP430 Ultra Low Power MCUs
□ Select All	Install Size: 723.14 MB. Download Size: 241.05 MB.
Texas Instruments 	Next > Finish Cancel

Pasul următor este dedicat alegerii driverelor necesare pentru interfețele de debug pe care le veți folosi. Asigurați-vă că opțiunea pentru **MSP430 USB FET** este selectată.

<u>~</u>
🖗 Code Composer Studio v6 Setup 🛛 🗙
Select Debug Probes
Select the debug probes you want installed and deselect the debug probes you want to leave out.
 TI XDS Debug Probe Support Blackhawk Debug Probes Spectrum Digital Debug Probes and Boards MSP430 USB FET Tiva/Stellaris ICDI Debug Probe
< >
Select All Install Size: 1290.83 MB.
Download Size: 430.28 MB.
Texas Instruments
< Back Next > Finish Cancel

În continuare se pot alege produse suplimentare precum compilatorul GCC pentru MSP430. Alegerea acestui compilator este opâională, compilatorul oferit de Texas Instrument fiind sufficient pentru îndeplinirea sarcinilor proiectului.

🗘 Code Composer !	Studio v6 Setup	o Cent	er	×
There are additiona Center. Selected add App Center when you Tools (for App MSP430 GCC EVE Compile Software (for A MSPWare GUI Compos	I products and featu d-ons will be downl ou run Code Compo Center background er App Center backgro ser	ures ("add-ons' oaded in the b ser Studio for t downlc downlc GNI dev und dov	") available from t ackground and ins the first time. cription U Compiler for MS ices.	he CCS App stalled by the
< └ Select All		> Insta Dow	ll Size: 1290.83 ME nload Size: 430.28	3. MB.
Texas Instruments —				
	< Back	Next >	Finish	Cancel

Instalarea propriu-zisă începe odată cu apăsarea butonului *Finish* și necesită o conexiune la internet pentru că implică descărcarea unor pachete de pe site-ul producătorului.



Fereastra urmatoare apare dupa finalizarea instalarii oferind posibilitarea de a crea scurtaturi si de a lansa Code Composer Studio.



2. Importarea proiectelor

Pentru a importa un proiect existent în CCS6 selectați funcția *Import* din meniul *File*. Din fereastra nou deschisă selectați *Code ComposerStudio > CCS Projects* și apăsați butonul *Next* ca în imaginea de mai jos.

💱 Import	_		×
Select Imports existing CCS Eclipse projects into workspace.			Ľ
Select an import source:			
type filter text			
 > General > C/C++ Code Composer Studio Build Variables CCS Projects Legacy CCSv3.3 Projects > Energia > Git > Install 			~
(?) < Back Next > Finish		Ca	ncel

În continuare selectați calea către proiectul deja existent ca în figura de mai jos și apăsați butonul *Finish*. Dacă se dorește copierea fișierelor proiectului în directorul current de lucru selectați *Copy projects into workspace*.

🐯 Import CCS Eclipse Pro	jects				×
Select CCS Projects to Select a directory to sear	Import ch for existing (CCS Eclipse projects	5.	1	
 Select search-directory: Select archive file: 	C:_Work\eZ	430-RF2500 Wirele	ss Sensor Monitor\Er	B <u>r</u> ow	se
Discovered projects:					
✓ [□] eZ430-RF2500_V	WSM [C:_Work	t∖eZ430-RF2500 Wi	reless Sensor Monitc	Select	t All
				<u>D</u> esele	ct All
				R <u>e</u> fre	esh
<			>		
Automatically import <u>re</u>	eferenced proje ospace	ects found in same s	search-directory		
Open the <u>Resource Explore</u>	er to browse av	ailable example pr	ojects		
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Canc	el

3. Utilizarea proiectului demo EZ430-RF2500

3.1. Descrierea proiectului demonstrativ

Pentru a ușura introducerea în dezvoltarea de aplicații folosind kit-ul EZ430-RF2500, TI oferă pe pagina dedicată produsului un exemplu de proiect ce poate fi implementat pe această platformă.

Exemplul de proiect deservește o aplicație cu o rețea de senzori ce monitorizează temperatura în diverse puncte alte perimetrului în care aceștia sunt plasați. Fiecare sensor măsoară periodic temperatura și o transmite către o unitate master conectată la un computer. Unitatea master transmite toate informațiile primite prin interfața serială către o aplicație ce afișează datele recepționate de la toți senzorii.



Imaginea de mai sus ilustrează configurația sistemului deservit de aplicația demonstrativă. Sistemul este compus din trei componente SW:

- Aplicația de monitorizare ce ruleaza pe PC
- Aplicația Access-point ce rulează pe dispozitivul MSP430 master conectat la PC
- Aplicația End-device ce ruleazp pe dispozitivele MSP430 slave

3.2. Importarea și configurarea proiectului în CCS 6

Descărcați proiectul de pe pagina de produs EZ430-RF2500 (<u>http://www.ti.com/tool/ez430-rf2500#</u> <u>whatsIncluded</u>). Pachetul descărcat conține cele 3 componente SW amintite în secțiunea anterioară. Aplicația pentru PC se regăsește în folderul PC sub forma unui installer. Aplicațiile pentru microcontroller le găsim in folderul Embedded în două versiuni. O versiune este construită pentru CCS iar cealaltă pentru mediul de dezvoltare IAR Embedded Workbench.

Extrageți conținutul pachetului descărcat și importați în CCS proiectul destinat acestui mediu de dezvoltare folosind pașii descriși în secțiunea 2. Proiectul conține atât codul sursă pentru componenta Access-point cât și cel pentru End-device. Pentru a selecta ca proiect activ unul din cele două faceți click dreapta pe titlul proiectului din secțiunea *Project Explorer* selectând mai apoi din meniul deschis optiunea *Build Configurations > Set Active* și configurația dorită, ca în imaginea următoare.

Project Explorer 🛛							
eZ430-RF2500_WSM [Active - Access I	Point	1		L			
> 🖑 Binaries		New	>				
> 🔊 Includes		Show In	>				
> 🗁 Access Point		Add Files					
Applications		Сору	Ctrl+C				
> Components	B	Paste	Ctrl+V				
> Configuration	×	Delete	Delete				
> End Device		Refactor	>				
> End Device		Source	>				
Interpretentings		Move					
Ink msp430f2274.cmd.back		Rename	F2				
macros.ini_initial		Import	>				
☑ msp430f2274.ld	2	Export					
project.log		Show Build Settings					
		Build Project					
		Clean Project					
		Rebuild Project					
	8	Refresh	F5				
		Close Project					
		Make Targets	>	3 🖳 Console			
		Index	>				
		Build Configurations	>	Manage			Reso
		Debug As	>	Set Active >	~	1 Access Point	
		Compare With	\$	Build All		2 Access Point_GNU	J
		Restore from Local History	ŕ	Clean All		3 End Device	
		Team	>	Build Selected	Γ		
		Description	Alta Catan		_		

Ca alternativă aceeași setare se poate face diin meniul *Project > Build Configurations > Set Active*.

Proiectul descărcat a fost construit cu o versiune anterioară de CCS. Pentru a putea fi utilizat cu versiunea curentă de CCS proiectul necesită adaptarea unor setări de compilator. În campul *Project Explorer* faceți click dreapta pe proiectul deschis si selectati optiunea properties.

In fereastra deschisa la sectiunea General asigurati-va ca aveti setarile precum in imaginea urmatoare.

Properties for eZ430-RF2500_WS	SM								×
type filter text	CCS General							← ▼ □	⇒ ▼ ▼
	Configuration:	Access Poir	nt [Active]			~	Manage	Configura	tions
MSP430 Hex Utility [Disak	Main								
Builders	Device								
> C/C++ Build	Eamily: N	/ISP430							\sim
Debug	Variant:	<select or="" t<="" td=""><td>ype filter text></td><td>~</td><td>MSP430F2274</td><td></td><td></td><td></td><td>\sim</td></select>	ype filter text>	~	MSP430F2274				\sim
Project References	Connection: T	1 MSP430 (JSB1 [Default]	\sim	Identify	(app	lies to wh	ole projec	ct)
Run/Debug Settings	\checkmark	Manage t	the project's target-conf	iguration aut	tomatically				
	Advanced sett	ings							
	Compiler versi	on:	TI v15.12.1.LTS				\sim	More	
	Output type:		Executable				\sim		
	Output format	:	eabi (ELF)				\sim		
	Device endian	ness:	little				\sim		
	Linker comma	nd file:	Ink_msp430f2274.cmd				~	B <u>r</u> owse	
	Runtime suppo	ort library:	rts430_eabi.lib				\sim	B <u>r</u> owse	
< >>									
? Hide advanced settings						OK		Cance	el

Astfel, conform imaginii setările din următoarele trei câmpuri trebuie modificate după cum urmează:

- *Compiler version* versiunea compilatorului trebuie configurată ca versiunea cea mai recentă disponibilă a compilatorului Ti;
- Output format formatul fișierului rezultat în urma compilării trebuie sa fie eabi (ELF);
- Runtime support library.- librăria utilizată este rts430_eabi.lib.

Aceste schimbări sunt necesare pentru ambele configurații ale proiectului. Pentru schimbarea proprietăților configurația poate fi selectată din campul *Configuration* din cadrul ferestrei de proprietăți. Asigurați-vă că adaptați schimbările menționate anterior pentru ambele configurații ale proiectului.

3.3. Compilare, linkeditare, programare și debug

Pentru a genera codul obiect pentru una din componente selectați-o drept configurație activă și faceți click pe iconița reprezentând un ciocan ⁶S. Ca alternativă puteți porni procesul din meniul *Project* > *Build Project*.

Adaptarea fișierului de linkeditare

În urma compilării și linkeditării vor fi generate 2 warninguri. Unul din ele se referă la crearea secțiunii ".data", iar cel de-al doilea este legat de vectorul de întreruperi. Ambele warninguri se datorează diferențelor dintre comportamentul compilatorul pentru care a fost inițial configurat proiectul și cel al versiunii nou configurate. Nerezolvarea celor două warninguri duce la probleme în execuția codului. Mai exact problemele manifestate sunt blocarea aplicației End-Point în faza de configurare a stackului de comunicație și neincluderea rutinelor de tratare ale întreruperilor în codul obiect generat.

Pentru a rezolva problema înlocuiți fișierul de linkeditare aflat în directorul proiectului la calea relativă *eZ430-RF2500_WSM**lnk_msp430f2274.cmd* cu cea mai nouă versiune a acestui fișier pe care o puteți găsi în directorul de instalare a CCS6: *ccsv6**ccs_base**msp430include**lnk_msp430f2274.cmd*.

Pentru a vă asigura că folosiți o versiune adecvată a fișierului de linkeditare verificați dacă în zona "SECTIONS" este definită secțiunea ".data" precum este ilustrat în secvența de mai jos.

SECTIONS {			
.bss	: {} > RAM	/* Global & static vars	*/
.data	: {} > RAM	<pre>/* Global & static vars</pre>	*/
.TI.noinit	: {} > RAM	/* For #pragma noinit	*/
.sysmem	: {} > RAM	<pre>/* Dynamic memory allocation area</pre>	*/
.stack	: {} > RAM (HIGH)	<pre>/* Software system stack</pre>	*/

De asemenea trebuie să vă asigurați că fișierul de linkeditare conține definiții pentru vectorul de întreruperi precum în exemplul de mai jos.

/* MSP430 Interr	upt vectors */
.int00	: {} > INT00
.int01	: {} > INT01
PORT1	: { * (.int02) } > INT02 type = VECT_INIT
PORT2	: { * (.int03) } > INT03 type = VECT_INIT
.int04	: {} > INT04
ADC10	: { * (.int05) } > INT05 type = VECT_INIT
USCIAB0TX	: { * (.int06) } > INT06 type = VECT_INIT
USCIABØRX	: { * (.int07) } > INT07 type = VECT_INIT
TIMERA1	: { * (.int08) } > INT08 type = VECT_INIT
TIMERA0	: { * (.int09) } > INT09 type = VECT_INIT
WDT	: { * (.int10) } > INT10 type = VECT_INIT
.int11	: {} > INT11
TIMERB1	: { * (.int12) } > INT12 type = VECT_INIT
TIMERB0	: { * (.int13) } > INT13 type = VECT_INIT
NMI	: { * (.int14) } > INT14 type = VECT_INIT
.reset	: {} > RESET /* MSP430 Reset vector */

Programare

Pentru a programa aplicația pe dispozitivul embedded și pentru a face debug trebuie să creați o configurație de debug. Deschideți fereastra de creare și editare a configurațiilor de debug din meniul *Run > Debug Configurations...* sau din submeniul iconiței ³/₄. Pentru a crea o configurație de debug selectați din lista optiunea *Code Composer Studio – Device Debugging* și apăsați butonul New ¹/₄.

Create, manage, and run configurations	
Start CCS Debug launch	1
 Image: Image: Im	Configure launch settings from this dialog: Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. Press the 'Duplicate' button to copy the selected configuration. Press the 'Filter' button to configure filtering options. - Edit or view an existing configuration by selecting it. Configure launch perspective settings from the <u>'Perspectives'</u> preference page.
ilter matched 3 of 3 items	Debug Close
Debug Configurations	
ireate, manage, and run configurations Start CCS Debug launch	
reate, manage, and run configurations Start CCS Debug launch	Name: EZ430
reate, manage, and run configurations Start CCS Debug launch	Name: EZ430
reate, manage, and run configurations Start CCS Debug launch Start CCS Debug launch Start CCS Debug launch Start CCS Debug prime text C/C++ Remote Application ✓ Code Composer Studio - Device Debugging	Name: EZ430 Main Program Target Source Common Use default target configuration Use default target configuration Use default target configuration
reate, manage, and run configurations start CCS Debug launch Start CCS Debug launch Start CCS Debug launch Start CCS Debug launch Start CCS Debug Start	Name: EZ430 Main Program Use default target configuration Target Configuration Target Configuration
reate, manage, and run configurations Start CCS Debug launch Start CCS Debug launch Start CCS Debug launch type filter text C/C++ Remote Application C/C++ Remote Application C/C++ Remote Application New configuration ► Launch Group	Name: EZ430 Main Program Use default target configuration Target Configuration Target Configuration File System Workspace Initialization Script
ireate, manage, and run configurations Start CCS Debug launch	Name: EZ430 Main Program Use default target configuration Target Configuration Target Configuration Initialization Script File System Workspace Werkspace
reate, manage, and run configurations Start CCS Debug launch Start CCS Debug launch Uppe filter text € C/C++ Remote Application € C/C++ Remote Application € New_configuration ► Launch Group	Name: EZ430 Main Program Use default target configuration Target Configuration Target Configuration Initialization Script File System Workspace Witarget Configuration File System Workspace Workspace Witarget Configuration Synchronize the properties for all compatible CPUs Use the same console for the CIO of all CPUs
reate, manage, and run configurations Start CCS Debug launch Start CCS Debug launch Start CCS Debug launch Start CCS Debug launch CCC + Remote Application CCC Code Composer Studio - Device Debugging New configuration ► Launch Group	Name: EZ430 • Main • Program = Target Source Common Use default target configuration Target Configuration
iter matched 4 of 4 items	Name: EZ430 • Main Program = Target Source Common Use default target configuration Target Configuration Target Configuration Target Configuration Target Configuration Titalization Script Initialization Script File System Workspace V

Introduceți un nume relevant pentru configurația nou creată în câmpul *Name* și selectați un fișier pentru câmpul Target Configuration. Fișierul de configurare pentru dispozitivul țintă îl veți găsi între fișierele proiectului urmând calea relativă *\Embedded\CCS\eZ430-RF2500_WSM\targetConfigs\ MSP430F2274.ccxml*. Apăsați butonul *Apply* pentru a salva schimbările efectuate.

Pentru a lansa sesiunea de debug apăsați butonul *Debug* din fereastra *Debug Configurations* sau selectați *Debug* din meniul *Run*. Procesul pornit astfel începe cu stabilirea conexiunii cu dispozitivul embedded și programarea microcontrollerului cu aplicația generata prin procesul de build.