

V4.8.8

**Copyright ECOTRONS LLC** 

**All Rights Reserved** 

# **Contact us:**

Web:	http://www.ecotrons.com
Email:	info@ecotrons.com
	ev-support@ecotrons.com
	ev-support@ecotrons.cor

Address: 13115 Barton Rd, STE H Whittier, CA, 90605 United States

Tel: +1 562-758-3039

+1 562-713-1105

Date	Version	Comments	Editor
6/20/2019	4.8.2		Ted Wang
7/24/2019	4.8.3	NVM Demo	David Wang
8/11/2019	4.8.4	Model Reference Section	Ted Wang
05/11/2020	4.8.5	Contact info update	Zack Li
07/10/2020	4.8.6	Screenshot update	Yibo Wang
07/16/2020	4.8.7	NVM update	Jake Li
07/30/2020	4.8.8	Add Task Monitor	Jake Li

# Contents

СН	APTE	ER 1 GENERAL INFORMATION10
1.1	Abo	ut EcoCoder
1.2	Syst	em Requirements11
1.3	МАТ	LAB Installation Requirements11
1.4	Supp	ported MATLAB Version 12
1.5	Deve	eloper's Kit
СН	APTE	ER 2 ECOCODER DEVELOPMENT ENVIRONMENT14
2.1	Soft	ware Installation List14
2.2	Code	eWarrior Installation16
2.3	MinC	SW-GCC Compiler Installation16
2.4	C++	Compiler Installation19
2.	4.1	Installation of Compiler for MATLAB 32-Bit20
2.	4.2	Compiler Selection for MATLAB 64-Bit21
2.5	Eco	Coder Installation 21
2.6	Link	S32DS_Power_Win32 to EcoCoder25
2.7	Link	HighTec TriCore Tool Chain to EcoCoder28
2.8	Link	CS+ to EcoCoder
2.9	Activ	vate EcoCoder
2.	9.1	Get Key File
2.	9.2	Activate EcoCoder by License (.dat) File
СН	APTE	ER 3 QUICK START ON APPLICATION SOFTWARE

CHAPTER 4 ECOCODER LIBRARY			
4.1	4.1 EcoCoder Target Definition 38		
4.2	Task	Scheduler	
4	4.2.1	Task Trigger	
	4.2.2	Task Monitor	
4.3	ADC		
	4.3.1	Read ADC Value	
	4.3.2	Read Fixed-Point ADC Volt	
	4.3.3	Read Float ADC Volt	
4.4	CAN	Communication	
	4.4.1	CAN Channel Definition	
	4.4.2	CAN Wake-up Frame Definition	
	4.4.3	Read Fixed-Point CAN Message	
	4.4.4	Send Fixed-Point CAN Message53	
	4.4.5	Read/Send CAN Message55	
4	4.4.6	CAN Receive Counter55	
	4.4.7	Set CAN Mode	
4	4.4.8	Send CAN Data57	
	4.4.9	Unpack Signals to CAN Data	
4	4.4.10	Pack Signals to CAN Data59	
	4.4.11	Receive CAN Message60	
	4.4.12	Transmit CAN Message61	
4.5	Seria	al Communication Interface (SCI) Block61	
	4.5.1	SCI Definition	
	4.5.2	Read SCI Data	
	4.5.3	Send SCI Data	
4.6	b Digit	al I/O 64	
	4.6.1	Switch Input	
4	4.6.2	KeyOn Input65	
	4.6.3	Switch Output	
	4.6.4	IPM Read	

4.6.5	PWM Definition	68
4.6.6	PWM Output	69
4.6.7	WakeUp Input	70
4.6.8	H-bridge Definition	71
4.6.9	H-bridge Output	72
4.6.10	PWM IO Frequency Range Definition	73
4.6.11	IPWM Read	74
47 101	O manual institution	
4.7 LIN		
4.7.1		
4.7.2	LIN Get Status	76
4.7.3	LIN Receive Date	77
4.7.4	LIN Transmit Data	78
4.8 Non	n-Volatile Memory Blocks	80
4.8.1	NVM Definition	80
4.8.2	NVM Variable Definition	81
4.8.3	Read NVM	82
4.8.4	Write NVM	83
4.8.5	8.5 Fixed NVM Definition	
4.8.6	4.8.6 Read Fixed NVM	
4.8.7	Write Fixed NVM	87
4.8.8	Store All NVM Data	88
4.8.9	Restore All NVM Data	89
	anastis Blacks	
4.9 Diag		
4.9.1		
4.9.2	Software Core Diagnostic	
4.9.3		
4.9.4		92
4.10 Ca	Calibration & Measurement	93
4.10.1	Calibration Definition	93
4.10.2	Read Calibration	94
4.10.3	Write Measurement	95
4.10.4	Write and Read Measurement	96

4.10.5	.5 Override Probe	97
4.10.6	.6 1-D Lookup Table	99
4.10.7	.7 2-D Lookup Table	100
4.10.8	.8 Calibration Data Check	101
4.11	System Management Blocks	102
4.11.1	.1 Power Management Example	102
4.11.2	.2 Shutdown Power	104
4.11.3	.3 Set ECU Mode	105
4.11.4	.4 ECU Master Chip Wake-Up Definition	106
4.11.5	.5 Watchdog Definition	107
4.11.6	.6 Software Reset	108
4.11.7	.7 Read System Free Counter	109
4.11.8	.8 Power Control Output	109
4.11.9	.9 Service Software Watchdog	110
4.12	CCP	111
4.12.1	.1 Fixed CCP Slave Definition	111
4.12.2	.2 CCP/CAL Seed&Key Security Definition	112
4.12.3	.3 CCP DAQ Seed&Key Security Definition	113
4.12.4	.4 CCP PGM Seed&Key Security Definition	113
4.12.5	.5 CCP Get Seed Trigger	114
4.12.6	.6 CCP Set Seed	114
4.12.7	.7 CCP Generate Seed Demo	115
4.13	Programming Blocks	115
4.13.1	.1 Online Programming Definition	115
4.13.2	.2 Programming Seed&Key Definition	117
4.13.3	.3 Entry UDS Programming	119
4.14	Sensors Blocks	120
4.14.1	.1 Read Gyro Hex Value	120
4.14.2	.2 Read Gyro Phy Value	120
4.15	Advanced Data Blocks	121
4.15.1	.1 Read OTP	121
4.15.2	.2 Read OTP (Input port)	122

4.1	5.3	Write OTP	
4.1	5.4	Write OTP (Input port)	
4.1	5.5	Read Data by Address	125
4.1	5.6	Read Data by Address (Input port)	126
4.1	5.7	Read String Value	127
4.1	5.8	EEPROM Emulation Definition	
4.1	5.9	Clear ALL EEPROM Emulation Record	128
4.1	5.10	Clear One EEPROM Emulation Record	129
4.1	5.11	Read EEPROM Emulation Record	129
4.1	5.12	Write EEPROM Emulation Record	130
4.1	5.13	Read Signals from EEPROM Emulation Record	
4.1	5.14	Write Signals to EEPROM Emulation Record	
4.1	5.15	Program First Run Flag	133
4.16	Appl	lication Base Blocks	133
4.1	6.1	Rising Edge	
4.1	6.2	Falling Edge	
4.1	6.3	Online Programming by SoftReset	
4.1	6.4	Online Programming by Entry UDS Programming	135
4.17	Mod	el Reference	136
4.1	7.1	Configurations for Parent Models and Referenced Models	137
4.1	7.2	Configuration Reference	138
4.1	7.3	Copy Parent Model Configuration File to Referenced Model	140
4.1	7.4	EcoCoder Blocks in Model Reference	140
СНА	PTER	5 CAN THEORY OF ECOTRONS	143
5.1	Introdu	ction	143
5.2	CAN Im	plementation	143
5.2	.1 Co	onvert DBC to m File	
5.2	.2 Eo	coCoder CAN Blocks	146
5.2	.3 Se	elect m file	
5.2	.4 Se	elect Message	
5.2	.5 Se	elect Sample Time	148

### **EC**ITRON

CHAPTER 6 MEMORY MANAGEMENT150			
6.1	Introduction		
6.2	Storage device		
<b>6.3</b> 6.	Data Storage       150         3.1       Calibration/Measurement Variable       150		
6.	3.2 Non-Volatile Variable		
СН	APTER 7 CUSTOM VARIABLE TYPE152		
7.1	Customize Variable Types 152		
7.2	Add Variables to Workspace153		
7.3	Customize Calibration Variables155		
7.4	Customize measurement Variables156		
7.5	Customize NVM Variables 157		
7.6	Save the Variables to M file158		
7.7 Load M file to Workspace159			
7.8	Model Example		
CHAPTER 8 PROGRAMMING VCU WITH ECOFLASH			
СН	APTER 9 MEASUREMENT AND CALIBRATION WITH ECOCAL162		
СН	APTER 10 UNINSTALL ECOCODER		
10.1	Uninstall EcoCoder from MATLAB163		
10.2	Uninstall EcoCoder from Windows System164		
СН	CHAPTER 11 FAQS		



APPENDIX A - NONVOLATILE VARIABLES THEORY	170
Non-volatile Variables	170
Fixed Non-volatile Variables	170
Battery Input	171

# Chapter 1 General Information

### **1.1 About EcoCoder**

EcoCoder is an advanced auto code generation library added on top of Simulink generic libraries. It links the user's Simulink models directly to Ecotrons target controller.

EcoCoder encapsulates the lower level driver software, or basic software, also abstracts the specific hardware, like Freescale or Infineon microprocessor-based controllers. It enables the controls engineer to develop their control systems completely in MATLAB/Simulink environment.

Plus, EcoCoder is only an add-on package on top of Simulink. It enables engineers to maximize the usage of Simulink generic library. It adds the necessary library blocks which bridge the gap between application software and the specific controller hardware. Meaning the application software will not be dependent on the specific hardware, and you can port your models to any other hardware which supports the Simulink. In short, you are not stuck with EcoCoder by using it.

Features:

- Auto-code generation of Simulink/Stateflow models using Embedded Coder/Stateflow Coder
- Calibration using EcoCAL or other CCP based software
- Programming using EcoFlash through CAN bus
- OTA upgrade of application software
- Available for both prototyping and production
- Manual C-code integration is available in addition to model-based design (MBD) with Simulink/EcoCoder



Benefits:

- Control engineers can be freed from time-consuming learning curve of hardware, C programming, and specific microprocessor settings
- Responsive support services from Ecotrons
   Application software development is isolated from a specific hardware, and it has transparency and easy migration to other platforms.

### **1.2** System Requirements

OS	Windows XP/Windows 7/Windows 10
СРИ	Intel CORE 2 Duo or higher
Memory	2 GB or higher
Hard drive	1 GB free hard disk space

### **1.3 MATLAB Installation Requirements**

Mandatory Components:

- MATLAB
- Simulink
- MATLAB Coder
- Simulink Coder
- Embedded Coder

Highly recommended components to be installed:

- Stateflow
- Stateflow Coder



#### 1.4 Supported MATLAB Version

- MATLAB R2010b 32-bit/64-bit
- MATLAB R2011a 32-bit/64-bit
- MATLAB R2011b 32-bit/64-bit
- MATLAB R2012a 32-bit/64-bit
- MATLAB R2012b 32-bit/64-bit
- MATLAB R2013a 32-bit/64-bit
- MATLAB R2013b 32-bit/64-bit
- MATLAB R2014a 32-bit/64-bit
- MATLAB R2014b 32-bit/64-bit
- MATLAB R2015a 32-bit/64-bit
- MATLAB R2015b 32-bit/64-bit
- MATLAB R2016a 64-bit
- MATLAB R2016b 64-bit
- MATLAB R2017a 64-bit
- MATLAB R2017b 64-bit
- MATLAB R2018a 64-bit
- MATLAB R2018b 64-bit
- MATLAB R2019a 64-bit
- MATLAB R2019b 64-bit

Note: some of the MATLAB versions (old) may require extra configurations to make the EcoCoder work. Contact us if you have compatibility issues.

#### **1.5 Developer's Kit**

- VCU
- Test Harness\*
- Ecotrons CAN (USB-CAN Adapter) \*

\* Test harness is available from Ecotrons; however, users can also make their own by using recommended connector parts.

\* Ecotrons CAN needs to be compatible with CAN Calibration Protocol (CCP). Third party adaptors like Kvaser



or PeakCAN should be compatible with Ecotrons products.



**Test Harness** 

# **Chapter 2** EcoCoder Development Environment

### 2.1 Software Installation List

Please install software tools in the following order:

#### 1. Integrated development environment for generating executable files

Main Chip	Integrated development environment
Infineon TC27x	HighTec TriCore Tool Chain
NXP SPC57xx	S32DS_Power_Win32_v2017.R1_b171019.exe
NXP SPC56xx	CodeWarrior for MPC55xxMPC56xx v2.10.exe
Renesas RH850	CS+(CSPlus_CC_Package_V70000.EXE)

#### 2. Compiler for generating DLL file (optional)

Compiler	Integrated development environment
	GCC compiler can generate DLL file, to set calibration,
Tdm64-gcc-4.9.2.exe	measurement, and program flashing permission.
	Support all MATLAB versions to generate DLL file.

- 3. EcoFlash Vxxxx Setup.exe
- 4. EcoCAL Vxxxx Setup.exe, or INCA
- 5. Ecotrons USB-CAN adapter Driver, or other CAN adapter Driver, such as Kvaser

product

6. Stateflow Coder (optional)

Compiler	Supported MATLAB version
C++ Compiler	Support Stateflow of MATLAB 32/64-bit
Lcc-win32	Support Stateflow of MATLAB 32-bit
MinGW-GCC	Support Stateflow of some MATLAB versions
	Please see the corresponding relationship between
	the MATLAB version and the supported MinGW
	version

#### 7. EcoCoder Setup.msi

#### Compilers and its Applications:

Compiler or integrated development environment utility	Application
HighTec TriCore Tool Chain	Compilation and link; HEX file generation for target EH2175A
S32DS_Power_Win32_v2017.R1_b171019.exe	Compilation and link; MOT file generation for NXP MPC5744 based units.
CodeWarrior for MPC55xxMPC56xx v2.10.exe	Compilation and link; MOT file generation for target EV2206B03, ET3206A
C++ Compiler	Support Stateflow of 32-bit and 64-bit MATLAB
Lcc-win32	Support Stateflow of 32-bit MATLAB
WinGW-GCC	<ol> <li>Generate DLL file for calibration, measurement and programming permission, support all versions of MATLAB</li> <li>Support Stateflow of some MATLAB versions, you can go to MATLAB official website to check whether it supports specific MATLAB version</li> </ol>
CS+(CSPlus_CC_Package_V70000.EXE)	Compilation and link, MOT file generation for target GWRH850

The corresponding relationship between the MATLAB version and the supported MinGW

version:

MATLAB version	Supported MinGW version
MATLAB2015a or below	Not support
MATLAB2015b	Support MinGW 4.9.2 (Distributor: TDM-GCC)
MATLAB 2016a	Support MinGW 4.9.2 (Distributor: TDM-GCC)
MATLAB 2016b	Support MinGW 4.9.2 (Distributor: TDM-GCC)
MATLAB 2017a	Support MinGW 4.9.2 (Distributor: TDM-GCC)
MATLAB 2017b	Support MinGW 5.3 (Distributor: TDM-GCC)
MATLAB 2018a	Support MinGW 5.3 (Distributor: TDM-GCC)

#### 2.2 CodeWarrior Installation

The installation instruction is for *CodeWarrior MPC55xxMPC56xx v2.10.exe*, if you installed other CodeWarrior version, please do the following after installation:

1. Run the "regserves.bat" file in the installation directory "Freescale\GW for MPC55xx and MPC56xx 2.10\bin". When the window appears, press any key to exit.

Di Windows (system oz (en diexe		
Registering core DLLs and IDE		
DllRegisterServer in .\Plugins\Support\MWComHelpers.d	1 succeeded.	
DllRegisterServer in .\Plugins\Support\MWRadModel.dll	succeeded.	
DllRegisterServer in .\Plugins\Support\CPlusSourceGen	dll succeeded.	
IDE.exe		
Done.		

2. If you follow the step 1 and other versions are called by default during compilation, you will need to uninstall other versions of CodeWarrior.

#### 2.3 MinGW-GCC Compiler Installation

The MinGW-GCC compiler can be installed via TDM-GCC. TDM-GCC is a compiler integration package for Windows that combines the latest version of the GCC toolset and includes API of open source MinGW or MinGW-w64. The installation steps are as follows:

1. First check "Check for updated files on the TDM-GCC server"

Vizard Action Choose which action you want the setup wizard to perform.	Σ
Create	Installations
: Create a new TDM-GCC installation	
Manage : Manage an existing TDM-GCC installation	
Remove	-
: Remove a TDM-GCC installation	
Check for updated files on the TDM-GCC server	
M-GCC Setup 1.1309.0	



#### 2. Choose MinGW-w64



#### 3. Choose the installation directory

H TDM-GCC Setup	X
New Installation: Installation Directory Choose the installation directory to use.	A
Setup will install TDM-GCC in the following folder. To install in a different folder, clic Browse and select another folder. Click Next to continue.	¢
Installation Directory	
C:\TDM-GCC-64\ Browse	
Space available: 19.7GB	Cancel

4. Choose TDM-GCC Recommended, C/C++

Check th	he components you	want installed and uncheck the com	ponents you don't want
Select t	he type of install:	TDM-GCC Recommended, C/C++	•
Or, sele	ct the optional com	ponents you wish to have installed: rent: 5.1.0-tdm64-1) : 2.25-tdm64-1) ie (MinGW-w64 Runtir (MinGW Stable: 3.82.	Description Position your mouse over a component to see its description,

Note: If the user uses MinGW-GCC as the Stateflow complier, proceed to the next step.

5. Add environment variables

Variable	Value	
HTC_DEVELOPM HTC_LICENSES	. D:\HighTec D:\HighTec\licenses	=
MOZ_PLUGIN_PA.	C:\Program Files (x86)\Foxit Softwar	e\
Path	C:\Program Files\Intel\WiFi\bin\;C:\P	ro
DI LI TOPUOP	5 100 1 <del>2</del> 10	
ustem variables	New Edit D	elete
ystem variables	New Edit D	elete
rstem variables Variable EP_NO_HOST_CH	New Edit D	elete
ystem variables Variable FP_NO_HOST_CH MW_MINGW64_EPC	New Edit D Value ECK. NO OC C:\TDM-GCC-64	elete
vstem variables Variable FP_NO_HOST_CH MW_MINGW64_L NUMBER_OF_PRO	New     Edit     D       Value     Value       ECK     NO       OC     C:\TDM-GCC-64       CESS     4	elete
vstem variables Variable FP_NO_HOST_CH MW_MINGW64_L NUMBER_OF_PRC	New         Edit         D           Value         Edit         D           ECK.         NO         NO           OC         C:\TDM-GCC-64         CESS 4           III         III         III	elete



- 6. Restart or open MATLAB.
- 7. Enter "mex -setup C++"

```
Command Window

    mex -setup C++
MEX configured to use 'MinGW64Compiler (C++)' for C++ language compilation.
Warning: The MATLAB C and Fortran API has changed to support MATLAB
    variables with more than 2^32-1 elements. In the near future
    you will be required to update your code to utilize the
    new API. You can find more information about this at:
    http://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html.
To choose a different C++ compiler, select one from the following:
    MinGW64 Compiler (C++) mex -setup:C:\Users\LEI\AppData\Roaming\MathWorks\MATLAB\R2016a\mex_C++_win64.xml C++
    Microsoft Visual C++ 2010 mex -setup:'H:\Program Files\MATLAB\R2016a\bin\win64\mexopts\msvcpp2010.xml' C++
    fx >>
```

#### 8. Choose MinGW64 Complier



#### 2.4 C++ Compiler Installation

MATLAB 32-bit system comes with a 'LCC' compiler which supports Stateflow automatic code generation. MATLAB 64-bit system does not provide compiler. To use Stateflow coder, it is necessary to install a third-party C++ Compiler that supports MATLAB 64-Bit version.

#### 2.4.1 Installation of Compiler for MATLAB 32-Bit

1. Type 'mex -setup' at MATLAB Command Window.



2. Type 'y' at Command Window.

Please choose your compiler for building external interface (MEX) files: Would you like mex to locate installed compilers [y]/n? y

3. Type '1' at Command Window.

Select a compiler: [1] <mark>Lcc-win32</mark> C 2.4.1 in C:\PROGRA~1\MATLAB\R2010b\	\sys\lcc
[0] None	
Compiler: 1	

4. Type 'y' at Command Window.

Compiler:	Lcc-win3	32 C 2.4.1
Location:	C:\PROGE	RA~1\MATLAB\R2010b\sys\lcc
Are these	correct	[y]/n? y

5. When the following information is displayed, the installation is successful.

Trying to update options file: C:\Documents and Settings\Administrator\Application Data\MathWorks\MATLAB\R2010b\mexopts.bat From template: C:\PROGRA~1\MATLAB\R2010b\bin\win32\mexopts\lccopts.bat Done . . .

#### 2.4.2 Compiler Selection for MATLAB 64-Bit

- Go to the official website of MathWorks https://www.mathworks.com/support/sysreg/previous releases.html
- 2. Click 'Details' under 'Supported Compilers' of MATLAB version on customer PC.
- 3. For Windows 64-bit system, refer to the page titled: 'Windows 64bit'.
- 4. After finishing the compiler installation, follow the steps in previous section to configure compiler for MATLAB.

#### **2.5** EcoCoder Installation

Note: Please keep MATLAB closed during the entire installation and licensing process.

1. Double-click 'EcoCoder 56xx Vx.x.x Setup.msi', click '*Next*' at the following screen.



2. Choose installation path, click 'Next'.

*Note: it is recommended to install EcoCoder under the system drive.* 

블 EcoCoder	
Select Installation Folder	
The installer will install EcoCoder to the following folder.	
To install in this folder, click "Next". To install to a different folder, enter it be	iow or click "Browse".
Folder	
C:\Program Files\ECO\ECS\	Browse
Ň	Disk Cost
Install EcoCoder for yourself, or for anyone who uses this computer:	$\backslash$
everyone	
⊘ Just me	
Cancel < Back	Next >

3. Click 'Next'.



4. Choose the version of MATLAB you want to install EcoCoder to, then select 'Install EcoCoder to selected MATLAB version', click 'OK'. You can also install EcoCoder for all MATLAB versions on computer.

EcoCoder Loader		
Tools		
Matlabroot: D:\Program Files\MATLAB\R2012b		
Select Matlab: MATLAB 8.0 (R2012b) x64		
EcoCoder_MPC		
Install EcoCoder to selected MATLAB version		
C Install EcoCoder to all supported MATLAB version		
C Uninstall EcoCoder from selected MATLAB version		
O Uninstall EcoCoder from all supported MATLAB version		
ОК		

5. Click '*Close*'.

EcoCoder				x
Installation Complete				
EcoCoder has been successfully install	led.			
Click "Close" to exit.				
	Lancel	< Back	Llose	



After successfully install it, the icon '*EcoCoder Loader*' will appear on the desktop.
 EcoCoder Loader will be used to generate the license file and activate EcoCoder.



 If you run MATLAB then, it will prompt message 'EcoCoder has been installed successfully' as shown in following red box. It indicates that EcoCoder is successfully installed to MATLAB.



### 2.6 Link S32DS\_Power\_Win32 to EcoCoder

Please refer to this video for setting up the S32DS IDE:

https://youtu.be/CiChf1 JZcw

Link for copy and paste:

https://www.nxp.com/support/developer-resources/run-time-software/s32-designstudio-ide/s32-design-studio-ide-for-power-architecture-based-mcus:S32DS-PA?tab=Design Tools Tab

NXP provides it for free, but you need to register your own account.

http://tdm-gcc.tdragon.net/download

And this is the step where you can copy and paste the path:

Add two directories where powerpc-eabivle-gcc.exe and make.exe are located to the environment variables. Split them by semicolon. Add at the end of the Variable value:

C:\NXP\S32DS\_Power\_v2017.R1\utils\msys32\usr\bin;C:\NXP\S32DS\_Power\_v2017.R1\ Cross\_Tools\powerpc-eabivle-4\_9\bin

In case you're not comfortable with video, here are the text steps:

If you use any MPC5744 based unit, you need to install the S32DS as a compiler, such as *S32DS\_Power\_Win32\_v2017.R1\_b171019.exe*, which can be downloaded from the NXP official website for free. After installing EcoCoder and S32DS\_Power\_Win32, you also need to use EcoCoder to select the path of *makefile* and add two directories where *powerpc-eabivle-gcc.exe* and *make.exe* are located to the environment variables.



1. Open EcoCoder, choose Tools > Select S32DS

EcoCoder Loader	×
Tools	
Get EcoCoder Key	LAB\R2010b
Active EcoCoder	
Select S32DS	(R2010b) x64
EcoCoder_MPC	
Install EcoCoder to selecte	d MATLAB version
C Install EcoCoder to all sup	ported MATLAB version
C Uninstall EcoCoder from se	elected MATLAB version
C Uninstall EcoCoder from a	I supported MATLAB version
	ОК

2. Click Browse, choose *makefile* under e200\_ew12, for example, the full path is:

🕑 Open				
💬 💬 👢 « S32DS 🕽	• e200_ewl2 ►	✓  ✓ Search e200_ewl2	Q	
Organize 🔻 New fold	er			
A	Name	Date modified	Туре	
Libraries	👢 EWL_C	2018/4/27 16:54	File folder	
A Music	ル EWL_C++	2018/4/27 16:54	File folder	
Distures	EWL_ReleaseNotes	2018/4/27 16:54	File folder	
Subversion	👢 EWL_Runtime	2018/4/27 16:54	File folder	
Subversion	👢 lib	2018/4/27 16:54	File folder	
Julieos	Build_Tag.txt	2016/10/18 22:40	Text Docu	
Computer	Doxyfile	2016/10/18 22:40	File	
Sustem LIQ (C)	🗋 makefile	2016/10/18 22:40	File	
System H0 (C.)				
Svstem (E:)				
File na	me:	▼ (*.*)		
		Open 🔽 Ca	ancel	Brows
		OK		
	L	UK		



3. Right click on Computer, then click on properties

**EC**ITRON



4. Choose Advanced system settings > Environment Variables

	L Surtan and Carutin Surtan
File Edit View Tools Help	)
Control Panel Home	System Properties
<ul> <li>Device Manager</li> <li>Remote settings</li> </ul>	You must be logged on as an Administrator to make most of these changes.
<ul> <li>System protection</li> <li>Advanced system settings</li> </ul>	Performance Visual effects, processor scheduling, memory usage, and virtual memory
	Settings
	User Profiles Desktop settings related to your logon Settings
	Startup and Recovery System startup, system failure, and debugging information
See also	Environment Variables
Action Center	
Windows Update Performance Information ar	OK Cancel Apply



5. Add two directories where powerpc-eabivle-gcc.exe and make.exe are located to the environment variables. Split them by semicolon. Add at the end of the Variable value:

C:\NXP\S32DS\_Power\_v2017.R1\utils\msys32\usr\bin;C:\NXP\S32DS\_Power\_v201 7.R1\Cross\_Tools\powerpc-eabivle-4\_9\bin

	able
Variable name:	Path
Variable value:	werpc-eabivle-4_9\bin;C:\TDM-GCC-64\bi
	OK Cancel
vetom variabloc	A
Variable	Value
Variable OS	Value Windows_NT
Variable OS Path	Value Windows_NT C:\Program Files (x86)\Vector vFlash 2
Variable OS Path PATHEXT	Value Windows_NT C:\Program Files (x86)\Vector vFlash 2 .COM;.EXE;.BAT; CMD;.VBS;.VBE;.JS;
Variable OS Path PATHEXT PICO_DRIVER	Value Windows_NT C:\Program Files (x86)\Vector vFlash 2 .COM;.EXE;.BAT; CMD;.VBS;.VBE;.JS; C.\Program Files Vector_CANoe_CANal

## 2.7 Link HighTec TriCore Tool Chain to EcoCoder

For EH2175A/EH2275A/EAXVA03, the installation of HighTec TriCore tool chain is required. After the installation of HighTec TriCore Tool Chain, users need to specify the *makefile* directory in EcoCoder Loader.

Open EcoCoder Loader, click Tools, then click "Select HighTec".

EcoCoder Loader	×
Tools	-
Get EcoCoder Key	B\R2014b
Active EcoCoder	
Select S32DS	2014b) x64 🗸 🗸 🗸
Select HighTec	
Select Tasking	IATLAB version
C Install EcoCoder to all support	ted MATLAB version
O Uninstall EcoCoder from sele	cted MATLAB version
C Uninstall EcoCoder from all st	upported MATLAB version
	ок

Then in the pop-up window, click "Browse", locate the "make.exe" in the HIGHTEC

installation path, in "bin" folder under the folder 'toolchains'.

For example, the full path can be: "C:\HIGHTEC\toolchains\tricore\v4.9.1.0-infineon-

1.1\bin\make.exe"



#### 2.8 Link CS+ to EcoCoder

If target is GWRH850, CS+ should be installed as a compiler, for example, you can install *CSPlus\_CC\_Package\_V70000.exe*. After installation, you need to add the directory where the file *CubeSuite+.exe* is located to the system environment path, then restart MATLAB. Regarding how to add environment variables, please refer to the operations in Link <u>S32DS\_Power\_Win32 to EcoCoder</u>.

#### 2.9 Activate EcoCoder

There are two ways to activate EcoCoder and other Ecotrons software.

*Note: Please close MATLAB for the activation process.* 

1. Dongle

The hardware dongle released by Ecotrons can activate software once it is plugged in PC.

2. '*.dat'* file

'.*dat*' file is linked to Windows SN, meaning the '.*dat*' file is bound to a specific PC and not allowed to be transferred to another PC. New '.*dat*' file must be issued if customer shift to new PC.

Note: all Ecotrons software would 'remember' license once it is activated even after it is upgraded to new version. It is mandatory to use Windows Add/Remove programs to uninstall all previously installed versions of EcoCoder. For safety concern, please install new version to same folder as previous EcoCoder.

### 2.9.1 Get Key File

1. Double-click "EcoCoder Loader" on the desktop.



2. Select Tools  $\rightarrow$  Get EcoCoder Key.

😴 EcoCoder Loader
Tools
Get EcoCoder Key R2010b
Active EcoCoder
THE MATLES 7. IT (HZ010b) x64
EcoCoder_MPC
Install EcoCoder to selected MATLAB version
Install EcoCoder to all supported MATLAB version
C Uninstall EcoCoder from selected MATLAB version
C Uninstall EcoCoder from all supported MATLAB version
ОК

3. Click 'Export'.

Export key license file	
C Export serial number	





4. Save the key file.



5. Please send the key file to *EV-Support@ecotrons.com* for license file.

# 2.9.2 Activate EcoCoder by License (.dat) File

1. Double-click '*EcoCoder Loader*' at the Desktop.



2. Select Tools->Activate EcoCoder.

EcoCoder Loader
Tools
Get EcoCoder Key R2012b
Active EcoCoder
EcoCoder_MPC
Install EcoCoder to selected MATLAB version
C Install EcoCoder to all supported MATLAB version
O Uninstall EcoCoder from selected MATLAB version
C Uninstall EcoCoder from all supported MATLAB version
OK

3. Click 'Browse'.

Active EcoCoder		Concernance.	
Activation mode			]
By the license file			
C By serial number			
			Browse
	ОК		

5. Open license file, for example, select 'EcoCoder\_license.dat', then select 'Open'.

Organize 👻 Nev	w folder		
E Desktop	* Name	Date modified	Туре
Downloads	EcoCoder_license.dat	2007/12/26 12:07	DAT File
词 Libraries			
Documents	E		
Documents			
<ul> <li>Documents</li> <li>Music</li> <li>Pictures</li> <li>Subversion</li> </ul>	E		
<ul> <li>Documents</li> <li>Music</li> <li>Pictures</li> <li>Subversion</li> <li>Videos</li> </ul>			
Documents  Music  Pictures Subversion Videos			
Documents Music Pictures Subversion Videos Local Disk (C:)			

6. Click 'OK'. The activation is successful if the pop-up window is displayed as following.



7. If following message shows up in MATLAB command window, EcoCoder should be ready to use.

Loadin	B BOOOder to Mithind
Type '	'EcoCoder_Reset" at Command Window to uninstall EcoCoder V2.5 or earlier vers
EcoCod	der Version: V2.6.10.7
EcoCod	der Directory: C:\Program Files (x86)\ECO\ECS\EcoCoder\EcoCoder_MPC\
Matlab	proot: F:\Program Files\MATLAB\R2013a
EcoCod	ler has been installed succesfully

# Chapter 3 Quick Start on Application Software

The purpose of this chapter is to give users a quick start to use EcoCoder for control system development. If you don't have any Simulink model yet, and want to have something to start with, or if you want to port your existing Simulink models into EcoCoder platform, this is a quick start. Because EcoCoder will provide an outline (a basic EV control model) for you, to fill in your existing model.

- 1. Change path to desired folder other than MATLAB installation directory.
- 2. Type the command '*EcoCoder\_Prj ('DemoTest')*' in Command Window.



3. A model 'DemoTest.mdl', and a '.m' file as shown in the following figure will be generated. In Simulink, by using shortcut key 'Ctrl + B' or click 'Build Model' button in Simulink task bar, 'a2l', '.mot', '.cal' file would be generated. (Note: for the latest version of EcoCoder, if users choose CCP type as 'configurable', EcoCoder will only generate '.a2l' and 'mot'/'hex'. )



If the 'CodeWarrior' window pops up, do not manually intervene, it will automatically compile and close when the process is finished.

As shown in below picture, you can find the generated '.mot', '.cal', and '.a2l' files.


# Chapter 4 EcoCoder Library

The EcoCoder library is an add-on library in Simulink. EcoCoder library mainly provides interfaces for application software to handle I/Os, VCU power, communication and calibration / measurement setup, etc.



~~



# 4.1 EcoCoder Target Definition

### Folder: EcoCoder Blocks

#### **Description:**

	BIOCK Parameters: Eco	Loder Target Definition		~
	EcoCoder Target Definition (mask) (link)			-
	General Parameters Advanced Parameters		eters	=
	Target: EV2106A-2		•	
EV2106A-2	Enable all required	execution modules t	o be scheduled	-
EcoCoder Target Setting	ОК	Cancel	Help	/
larget :EV2106A-2				
EcoCoder Tarr General Para User Coder RC [hex2dec('000 Calibration RC [hex2dec('400 Measurement [hex2dec('400 Default Data(, [hex2dec('400 Fixed NVM(Ad [hex2dec('400 Non-fixed NVM [hex2dec('400 Stack(Address	get Definition (mask) (link)           meters         Advanced Parameters           DM(Address, Length)         D20000'),hex2dec('000E0000')]           DM(Address, Length)         D10000'),hex2dec('00002000')]           AM(Address, Length)         D0F000'),hex2dec('00002000')]           Data(Address, Length)         D00000'),hex2dec('00002000')]           Data(Address, Length)         D00000'),hex2dec('00002000')]           Address, Length)         D00000'),hex2dec('00000000')]           Address, Length)         D02000'),hex2dec('00001000')]           dress, Length)         D11000'),hex2dec('00001000')]           f(Address, Length)         D12000'),hex2dec('00001000')]           i, Length)         D12000'),hex2dec('00001000')]           i, Length)         D12000'),hex2dec('00001000')]			
	OK Cancel	Help Apply		

Under the 'General Parameters' tab, this block defines the specific model of Ecotrons VCU that you are using.

Place this block in application model, usually at the top level, to select the VCU model for users' application. The 'Advanced Parameters' tab enables the user to work with part of the ROM and RAM memory addresses. If you would like to do the adjustment regarding all the addresses, please contact Ecotrons Tech Support, otherwise, please keep it as default.

### **Block Parameters:**

Parameter Field	Value	Comments/Description
Target	Drop-down list	Pick target VCU
Enable all required	Check box	If enabled:
execution modules to be		All subsystems that are not
scheduled		assigned to tasking triggers
		would be assigned to L1ms
		trigger*
Advanced Parameters	Memory addresses (RAM,	Contact our tech support
	ROM)	for adjustments

\* Please refer to Task Scheduler for knowledge of 'Tasking'.

### 4.2 Task Scheduler

### 4.2.1 Task Trigger

### Folder: EcoCoder Blocks/Task Scheduler

**Description:** 

		ſ	Source Block Parameters	s: Task Trigger	x
_		_	Subsystem (mask) (lin	k)	
	Task_L20ms	f() >	Parameters		
V	Task Trigger	-1	Set Task time Task_L2	20ms	-
	lusk mggo		Task_ir Task_L1 Task_L5 Task_L1 Task_L2	ni 1ms 5ms 10ms 20ms	
			Task_L5 Task_L1 Task_L2 Task_L5	50ms 100ms 200ms 500ms	
			Task_L1 Task_H1 Task_H5 Task_H1 Task_H1 Task_H2	1000ms 1ms 5ms 10ms 20ms	

This block is for task scheduling and prioritization.

Definition / initialization blocks need to be executed when the VCU power on, for

variables initialization/parameter definitions.

All other blocks/subsystems should be triggered by this block, for task prioritization and scheduling.

### **Block Parameters**

Parameter Field	Value	Comments/Description
Set Task Time	Drop-down list	Task type and execution
		period selection*

\* H represents high priority, tasks will be implemented by interruption. L represents low priority, tasks will be implemented by software timer function call.

If two tasks are assigned to the same task type, then the user needs to specify priority of the two tasks to determine execution order. Please refer to the example in the following link for more information: <u>https://www.mathworks.com/help/simulink/examples/block-priority.html</u>

\* For CAN bus applications, they are recommended to be set up in 10ms tasks.

## 4.2.2 Task Monitor

### Folder: EcoCoder Blocks/Task Scheduler

### **Description:**

	🔁 Source Block Parameters: Task Monitor
	Task Monitor (mask) (link)
	Parameters
Task_H1ms ms ►	Set Task time Task_H1ms  TRTC Frequency(KHz)
Task Monitor	5000
	OK Cancel Help Apply

This module is used to monitor task scheduling time.

### **Block Parameters**

Parameter Field	Value	Comments/Description
Set Task Time	Drop-down list	Task type selection
		NXP5606: 150,000
		NXP5744: 5,000
RTC Frequency (KHz)	Numeric	TC275: 100,000
		TC297: 200,000
		TC397: 100,000

### Block Output:

- 1. ms: The actual execution cycle of the task.
- 2. %: Load factor of chosen task type.

### 4.3 ADC

### 4.3.1 Read ADC Value

### Folder: EcoCoder Blocks/ADC

**Description:** 



In most cases, there are voltage dividing and shifting circuits on the target VCU hardware, they map the physical voltages being measured into the range that the microcontroller chip(s) can read, usually 0 to 5V. The resolution at which this pre-processed voltage by dividing circuits can be measured depends on the controller chip, usually 10 or 12 bits (1023 or 4095 maximum value, respectively). A reading of 0 represents the minimum voltage specified for these external circuits and a maximum value (1023 or 4095) represents the highest specified voltage.

This EcoCoder block outputs values of the A/D converter channel connected to corresponding physical pin. The output value of this block is the output of AD converter chip (10- or 12-bits binary value).

EcoCoder has predefined input voltage range and resolution of each channel, please refer to datasheet of the specific VCU.

Channel	ADC Predefined	RAW ADC	Raw ADC (binary)
Copyright ECOTRONS LLC	<u>www.</u>	ecotrons.com	Page 43

See examples below:



	Resolution (bits)		
1	10	500	000111110100
2	12	500	00000111110100

### **Block Parameters:**

Parameter Field	Value	Comments/Description
Analog input channel	0 – n (determined by hardware resource of specific VCU)	Pick specific analog input channel

# 4.3.2 Read Fixed-Point ADC Volt

Folder: EcoCoder Blocks/ADC

Description:



	Source Block Parameters: Read Fixed-Point ADC Volt
Channel:A0 Type:Custom Voltage Ratio Resolution:12Bit vo Custom Voltage Ratio:1 Reference Voltage(V):5 Conversion factor:0.001221	Subsystem (mask) (link)         Parameters         Analog input channel:         Input type:         Volt_0_5V
Read Fixed-Point ADC Volt	AD Resolution: 12Bit Custom Voltage Ratio: 1 Reference Voltage:(V) 5 QK Cancel Help Apply

This block enables user to read the physical voltage at the physical pin on the connector, and block output data type is fixed-point.

#### **Block Parameters:**

1. Analog input channel: Choose analog channel.

2. Input type: Channel type of the voltage input, 4 types are supported: '0-5V', '0-12V',

'0-24V' and 'Custom Voltage Ratio'.

3. AD Resolution: Please refer to the datasheet of VCU for resolution selection.

4. *Custom Voltage Ratio*: This option is available for the fourth input type 'Custom Voltage Ratio' only. Previously, every voltage input type has fixed resistor divider, as a result, the user just needs to select voltage type. Different resistor divider is introduced in new hardware datasheet, which gives the introduction of new input type and this input option. 5. *Reference Voltage*: By default, it will be set as 5V. **Please do not change**.

#### Block Output:

*Volt*: Physical value of input voltage of specified channel; unit: V; 'single' data type.

(\*) For fixed point toolbox advantages, refer to

https://www.mathworks.com/help/simulink/fixed-point.html

(\*) Every channel has its unique configuration defined in firmware, please refer to

datasheet of VCU and select correct setting for the channel

# 4.3.3 Read Float ADC Volt

(	🔁 Source Block Parameters: Read Float ADC Volt	X
	Subsystem (mask) (link)	
Channel:A0 Type:Custom Voltage Ratio Resolution:12Bit Oustom Voltage Ratio:1 Reference Voltage(V):5 Conversion factor:0.001221 Read Float ADC Volt	Parameters Analog input channel: 0 Input type: Custom Voltage Ratio AD Resolution: 12Bit Custom Voltage Ratio: 1 Reference Voltage:(V) 5	•
	<u>QK</u> <u>Cancel</u> <u>Help</u> A	pply
	<u>OK</u> <u>Cancel</u> <u>Help</u> <u>A</u>	рріу

This block enables user to read the physical voltage at the physical pin on the connector, and block output data type is float-point.

### **Block Parameters:**

1. Analog input channel: Choose analog channel.

2. *Input type*: Channel type of the voltage input, 4 types are supported: '0-5V', '0-12V', '0-24V' and 'Custom Voltage Ratio'.

3. *AD Resolution*: Please refer to the datasheet of VCU for resolution selection, since different VCUs have different AI channels setup.

4. *Custom Voltage Ratio*: This option is available for fourth input type 'Custom Voltage Ratio' only. Previously, every voltage input type has fixed resistor divider, as a result, the user just needs to select voltage type. Different resistor divider is introduced in new hardware, which explain the introduction of new input type and this input option.

5. *Reference Voltage*: By default, it will be set as 5V. Please do not change.

### **Block Output:**

*Volt*: Physical value of input voltage of specified channel.

## 4.4 CAN Communication

Please Refer to Chapter 5 CAN theory of Ecotrons before using EcoCoder CAN blocks.

Chapter 5, combined with CAN bus communication protocol, will give the user

preliminary knowledge of implementing CAN on Ecotrons VCU.

### 4.4.1 CAN Channel Definition

Folder: EcoCoder Blocks/CAN

Description:



CAN_A :on ID Filter Enable:off Extended Frame:off Mask:7FF ID:0x100 Baud Rate:500kbps Tx Queue:=30 messages Rx Queue:=1 messages CAN Channel Definition2	Block Parameters: CAN Channel Definition2         CAN Channel Definition (mask) (link)         Define the CAN Channel of the ECU.         Parameters         CAN_Channel CAN_A         CAN_Enable         CAN ID Filter Enable         CAN ID Filter Enable         CAN ID Ask(uint32 Hex)         hex2dec('7ff')         CAN Baud Rate(k bps)         S00         CAN TxBuffer Size         30         CAN RxBuffer Size         1
	CAN TXBUffer Size 30 CAN RxBuffer Size 1 <u>OK</u> <u>Cancel</u> <u>Help</u> <u>Apply</u> <u>Area</u> <u>Apply</u>

This block provides configuration interface for CAN low level protocol parameters. It is recommended to read through and understand CAN low level protocol prior to designing CAN related application software.

Parameter Field	Value	Comments/Description
CAN_Channel	Drop-down list	Please refer to datasheet to select supported CAN channels. In some cases, CAN A is represented by CAN 0.
CAN_Enable	Check box	If checked: the selected CAN channel would be activated

ECOTRON

EcoCoder Manual

CAN ID Filter* Enable	Check box	If checked: message with ID in filter list would be filtered out by VCU on the selected CAN bus.
CAN Extended	Check box	If checked: CAN ID input would be interpreted as extended format
CAN ID Mask (uint32 Hex)	Not Configurable	fixed at '7ff', which means only IDs with lower 11 bits same as input of 'CAN ID Filter' will be accepted by VCU if 'CAN ID Filter Enable' checked
CAN ID Filter (uint32 Hex)	Numeric	Specify the filter
CAN Baud Rate (bps)	Drop-down list	Specify baud rate
CAN TxBuffer Size	Numeric	It is used to specify software buffer size to help store the sequence of message to be sent
CAN RxBuffer Size	Numeric	It is used to specify software buffer size to help store the sequence of incoming message.

# 4.4.2 CAN Wake-up Frame Definition

	🚡 Block Parameters: CAN Wake-up Frame Definition	η
	CAN Channel Definition (mask) (link)	
	Define the CAN Channel of the ECU.	
Channel:CAN_A Wake-up Mode:Specific Frames Baud Rate(k bps):500 II Extended:off IID Setting.fnex2dec(100') ID Mask.hez2dec(100') Data Length CorF OxFF OxFF Data-0:Mesk.OxFF OxFF OxFF Data-0:Mesk.OxFF OxFF OxFF Data-0:Mesk.OxFF OxFF OxFF Data-0:Mesk-up Frame Definition	Parameters CAN Channel CAN_A Wake-up Mode Specific Frames Baud Rate(k bps) 500 ID Extended ID Setting hex2dec('100') ID Mask hex2dec('7ff') Data Setting Enable Data Length	
	8	
	Data Mask	1
	:('ff'),hex2dec('ff'),hex2dec('ff'),hex2dec('ff'),hex2dec('ff')]  OK Cancel Help Apply	
	Ø Data Setting Enable         Data Length         8         8         Data Mask         .('ff'),hex2dec('ff'),hex2dec('ff'),hex2dec('ff')]         OK       Cancel         Help       Apply	

This block is used to define the VCU wake-up CAN message, it is only supported by several new VCU models. For more information, please consult Ecotrons support.

#### **Parameters:**

- 1. CAN Channel: Selecting CAN channel number for this function.
- 2. *Wake-up Mode*: Setting the wake-up mode, including *Disable* (disable CAN wake up function), *All Frames* (any frame on the specified bus can wake VCU up), and *Specific Frames* (User specify the frame that can wake up the VCU).
- 3. Baud Rate: CAN baud rate set up.
- 4. *ID Extended*: If checked, the specified VCU-waking-up message will have to use extended CAN ID. If not checked, the message has to use standard CAN ID.

- 5. *ID Setting*: Specify the ID here for the wake-up message.
- 6. ID Mask: The mask for VCU-waking-up message ID.
- 7. *Data Setting Enable*: If checked, not only a specific waking-up message ID is needed, but the user also needs to specify the data in the message. Only message with matching ID and data can wake up the VCU. Data can be specified in the following blank.
- 8. *Data Length*: Set the wake-up message data length. Only when the data length of the wake-up message matches this value, the message can wake up the VCU.
- 9. *Data Mask*: The mask for wake-up message data. The message data bitwise AND with this mask value, if one or more bit of the result of bitwise AND is (are) not 0, the message can wake up the VCU.

### 4.4.3 Read Fixed-Point CAN Message

### Folder: EcoCoder Blocks/CAN

#### Description:



This block provides CAN messages receiving and unpacking functions. It requires a .m file of CAN message definition to help unpack CAN messages. The generation process (from *.dbc* file to .m file) is explained in Chapter 5.

Parameter Field	Value	Comments/Description
Select CAN Channel	Drop-down list	The CAN channel has to be defined before applied.
Select M File	Check box	If checked: please enter the name of m file in the blank space under check box.
Select Message	Drop-down list	Specify CAN message to be received and processed by the block.
Show Message Available Port	Check box	If checked: the block will provide a signal flag to help tell the availability of this CAN message.
Show Message Count Port	Check box	Message counter, if checked: every time the message is received, the counter increments by 1.
Show Signal Name	Check box	If checked: the names of the signals will be displayed.
Define Signal	Check box	If checked: signals parsed out from the block will be cast as measurement variables. <i>'Show Signal</i> <i>Name'</i> must be checked before checking this item.
Signal prefix	Alpha-numeric text	Specify prefix to parsed out signal names, remember to



EcoCoder Manual

		use single quote.
Output Data Type: Inherit	Check box	If checked: the data type of the signal is inherited from input data type. If not checked: the signal type is automatically defined using fixed point tool.
Sample time	Drop-down list	Please refer to section 5.2.5

# 4.4.4 Send Fixed-Point CAN Message

### Folder: EcoCoder Blocks/CAN

Description:

									1	Block Parameters: Send Fixed-Point CAN Message
									-5	Send Fixed-Point CAN Message (mask) (link)
Demand_Speed Demand_Tonjue FaulReset	CAN V2.7.2 Send Message : HCU_C ID : 0x00000113(standard	COMMAND	:0 Length:8(b)	vtes) Int	erval Inherit(In	herit)		Remote >	-F -S	Parameters Select CAN Channel State Connected  Select CAN Channel CAN_A
Demand_LimitValid	Signal Units name	s  Start Les   (LSB)  (	ngth  Data  bit)  type	Byte Fa order	ctor Offset  	Multiplex   1 type	fultiplex value			Select M Flie
Demand_LimitLow	Demand_Speed  Demand_Torque	I 561 I 401	16  unsigned m 14  unsigned m	otorola  otorola	1 -15000  1  -5000	Standard  Standard	0			EcoCanM_Demo
Demand_LimitHigh	FaultReset  Demand_LimitValid	I 391 I 381	1  unsigned m 1  unsigned m	otorola; otorola;	1  0  1  0	Standard  Standard	0			Select Message HCU_COMMAND
Live_Counter	Demand_LimitLow  Demand_LimitHigh	241	12  unsigned m 12  unsigned m	otorola  otorola	41 01 41 01	Standard  Standard	0	Length >	ſ	
Control_Mode	Live_Counter  Control_Mode	41	4  unsigned m 3  unsigned m	otorola  otorola	1 0	Standard  Standard	0			
MCU_Enable	MCU_Enable(	1 01	i) unsigned(m	otorola	11 01	Standard)	U	Data		
			Send Foed-PointC	AN Message						
										OK Cancel Help Apply

Parameter Field	Value	Comments/Description
Select CAN Channel State	Drop-down list - Connected	Connected: Message will be sent out from the CAN

ECOTRON

	- Disconnected	channel selected under 'Select CAN Channel' Disconnected: User will have to assign the output port manually.
Select CAN Channel	Drop-down list	CAN channel selection
Select M File	Check box	If checked: please enter the name of m file in the blank space under check box.
Select Message	Drop-down list	Specify CAN message to be sent and processed by the block.
Input Data Type: Inherit	Check box	If checked: the data type of the signal is inherited from input data type. If not checked: the signal type is automatically defined using fixed point tool.
Sample time	Drop-down list	Please refer to section 5.2.5

### **Block Inputs:**

Signals to be sent out.

Block Outputs (if 'Disconnected' is selected under 'Select CAN Channel State'):

- 1. Remote: frame type-1 is remote frame, 0 is standard frame
- 2. Extended: frame type- 1 is extended frame, 0 is standard frame.
- 3. ID: message ID.
- 4. Length: message data length.
- 5. Data: message data

### 4.4.5 Read/Send CAN Message

These two blocks are similar to previous two blocks which is generic version of CAN read/send for customer who did not purchase fixed point toolbox of Simulink. However, if the toolbox available, it is recommended to use fixed point version of read/send blocks.

# 4.4.6 CAN Receive Counter

### Folder: EcoCoder Blocks/CAN

**Description:** 



This module can be used to count the number of frames received by specified CAN channel.

### **Block Parameters**

Parameter Field	Value	Comments/Description
CAN channel	Dron-down list	Specify the channel to be
	Drop-down list	monitored

### Block Output:

cnt: If the selected channel receives one frame, cnt value increments by 1.

# 4.4.7 Set CAN Mode

### Folder: EcoCoder Blocks/CAN

### **Description:**

(	Block Parameters: Set CAN Mode		
	Subsystem (mask) (link)		
	Parameters		
A	Trigger type rising		
Channel:CAN_A Mode:ListenOnly	CAN Channel CAN_A		
	CAN Mode ListenOnly		
Set CAN Mode			
	OK Cancel Help Apply		

This module can be used to switch CAN operating mode between 'listen only' and

'normal'.

### **Block Parameters**

Parameter Field	Value	Comments/Description
Trigger type	Drop-down list	Trigger type selection
CAN Channel	Drop-down list	Specify CAN channel to be controlled
CAN Mode	Drop-down list	Specify the CAN mode to be triggered by the block

### Block Input:

Trigger signal: the signal input to trigger the execution of this block.

# 4.4.8 Send CAN Data

	Block Parameters: Send CAN Data
	Parameters CAN Channel CAN_A
data CAN_AID(11):101	8 CANID
Send CAN Data	hex2dec('101')         FrameType         Show error status
	OK Cancel Help Apply

### **Block Parameters:**

- 1. CAN Channel: CAN channel selection.
- 2. Data Length: Message data length, in bytes.
- 3. CANID: The ID of the message to be sent. HEX value.
- 4. Frame Type: Drop-down list for frame type selection.

### **Block Inputs:**

data: The message data to be sent.

# 4.4.9 Unpack Signals to CAN Data



### **Block Parameter:**

Signal (Array): the signal definition matrix of CAN frame.

#### Block Inputs:

Data: the message data to be unpacked.

### **Block Outputs:**

Unpacked signals from the CAN data.

# 4.4.10Pack Signals to CAN Data

Pack CAN signals to CAN message, usually used together with Send CAN Data block.

G		Block Parameters: Pack Signals To CAN Data	×
2	BMS_HCU_0_Heatbeat (0:2:uint8:Motorola)	Pack Signals To CAN Data (mask) (link)	
2	AC_CC (2:1:boolean:Motorola)	Devenue de un	
2	DC_CC (3:1:boolean:Motorola)	Parameters	
2	BMS_Md (4:1:boolean:Motorola)		
2	BMS_SysFitLvI (5:3:uint8:Motorola)		
>	BMS_SOC (8:8:uint8:Motorola)	OK Cancel Help A	Apply
2	BMS_CellHVoltNum (16:8:uint8:Motorola)	(	
>	BMS_CellHVolt (32:16:uint16:Motorola)		
>	BMS_CellLVoltNum (40:8:uint8:Motorola)		
2	BMS_CellLVolt (58:16:uint16:Motorola)		
Ľ	Pack Signals To CAN Data		

#### **Block Parameters:**

Out Signal (Array): The definition array of the signals to be packed.

### **Block Inputs:**

Signals to be packed, values are in HEX.

### **Block Output:**

Data: the packed CAN message data.

### 4.4.11 Receive CAN Message

	Source Block Parameters: Receive CAN Message		
	Receive CAN Message (mask)		
	Parameters		
Message Available 🕨	CAN Channel CAN_C		
Channel:CAN_C Remote	CAN ID		
Extended:on Extended	hex2dec('101')		
Sample Time:Task_Inherit	✓ Extended		
Data 🕨 —	Data Length		
Receive CAN Message	8		
	Sample Time Task_Inherit		
	OK Cancel Help Apply		

#### **Block Parameters:**

- 1. CAN channel: CAN channel selection.
- 2. CAN ID: The ID of the message to be received.
- 3. Extended: Message type to be received. If checked: extended frame. Otherwise, standard frame.
- 4. Data Length: The data length of the to-be-received message.
- 5. Sample Time: Define the task scheduling time of this block being triggered.

#### Block Outputs:

- 1. Message Available: Flag for message availability, 1 stands for message valid and available.
- 2. Remote: Flag for frame type, 1 stands for remote frame. 0 stands for data frame.
- 3. Extended: Flag for frame type, 1 stands for extended frame. 0 stands for standard frame.
- 4. ID: Message ID.
- 5. Length: Message data length.
- 6. Data: Message data.

### 4.4.12Transmit CAN Message

	Sink Block Parameters: Transmit CAN Message		
	Transmit CAN Message (mask)		
Remote			
Channel:CAN_C ID Sample Time:Task_Inherit	Parameters CAN Channel CAN_C		
Data Transmit CAN Message	Sample Time Task_Inherit		
	OK Cancel Help Apply		

**Block Parameters:** 

- 1. CAN Channel: Channel selection
- 2. Sample Time: Define the task scheduling time of this block.

Block Inputs:

- 1. Remote: Flag for frame type, 1 stands for remote frame. 0 stands for data frame.
- 2. Extended: Flag for frame type, 1 stands for extended frame. 0 stands for standard frame.
- 3. ID: Message ID.
- 4. Length: Message data length.
- 5. Data: Message data.

# 4.5 Serial Communication Interface (SCI) Block

The SCI mode includes SCI\_RxData and SCI\_TxData. Currently, only SCI\_A channel is supported.

# 4.5.1 SCI Definition

	Block Parameters: SCI Definition		
	HCU SCI (mask) (link)		
	Configure the SCI module of the HCU.		
	Parameters		
SCIA = off	SCI Channel SCI_A		
	SCI_Enable		
SCI Definition	SCI Baudrate 2400 💌		
	SCI Parity Mode NONE		
	UN Cancel Help Apply		

#### Block Parameters:

- 3. SCI Channel: Communication channel selection.
- 4. SCI\_Enable: Enable selected channel.
- 5. SCI Baud rate: Channel baud rate setup.
- 6. SCI Parity Mode: Parity check mode setup.

### 4.5.2 Read SCI Data

This block enables the VCU to read data from specific SCI port.

	Source Block Parameters: Read SCI Data		
SCI_A f() data	Parameters		
Read SCI Data	SCI channel SCI_A		
	OK Cancel Help Apply		

#### **Block Parameter:**

SCI\_Channel: SCI communication channel selection.

### **Block Outputs:**

1. f (): Flag for receiving data. If data received, the flag will be 1. This signal can

be used as a trigger signal.

2. Data: Output received 8-bit data.

### 4.5.3 Send SCI Data

This block will send SCI data to assigned channel.

	Sink Block Parameters: Send SCI Data		
	S-Function (mask) (link)		
data SCI_A	Parameters		
Send SCI Data	SCI channel SCI_A -		
	OK Cancel Help Apply		

#### **Block Parameter:**

SCI\_Channel: SCI channel selection.

### **Block Input:**

Data: The 8-bit data to be sent out.

# 4.6 Digital I/O

These blocks are used for measuring digital input/output. Including Switch Input, Switch Output, PWM input and PWM output.

# 4.6.1 Switch Input

### Folder: EcoCoder Blocks/Digital I/O

### Description:



This block reads the physical voltage level of digital input channels and output Boolean

variable to application layer.

Parameter Field	Value	Comments/Description
Switch input channel	Drop-down list	Digital input channel
Switch input chainer		selection

### 4.6.2 KeyOn Input

#### Folder: EcoCoder Blocks/Digital I/O

**Description:** 

(	Source Block Parameters: KeyOn Input	
	S-Function (mask) (link)	
	Parameters	
	Switch input channel DI_KEYON	
	Key AD2Volt Factor:	
DIRead:DI_KEYON bool	109/27846	
KeyOn Input	Key Off Threshold Volt:(V)	
	7	
	Key On Hysteresis Volt:(V)	
	2	
	OK Cancel Help Apply	

KeyOn signal is recommended to be used for powering up and shutting down the VCU.

For different VCUs, KeyOn signal inputs are different (refer to the VCU datasheet to confirm the KeyOn signal input type) - if KeyOn signal is digital input, leave the configuration as default; If KeyOn signal is read through analog input channel, user will have to configure factor according to voltage divider parameter - for this parameter, please refer to VCU datasheet.

Parameter Field	Value	Comments/Description
Switch input channel	Drop-down list	Only one channel selectable for KeyOn
		The voltage factor for KeyOn voltage
Key AD2Volt Factor	Numeric	detection (only valid when KeyOn is
		read from AI)
Kay Off Thrashold Valt	Numorio	If the input voltage is lower than this
Key On Threshold Volt	Numeric	value, output is '0'.



	Numeric	The hysteresis value between upper
		and lower threshold. If the interpreted
KeyOn Hysteresis Volt		voltage is larger than the sum of 'Key
		Off Threshold Volt' value and this
		value, output is '1'.

# 4.6.3 Switch Output

### Folder: EcoCoder Blocks/Digital I/O

### **Description:**



Parameter Field	Value	Comments/Description
	Drop-down list	Select switch channels to
Switch output channel		be controlled
Input	Numeric (bool)	0 or 1, switch control value

### 4.6.4 IPM Read

### Folder: EcoCoder Blocks/Digital I/O

### **Description:**



This block measures the frequency of input PWM signal and returns the PWM signal period.

Parameter Field	Value	Comments/Description
IDW/M channel	Drop-down list	Select channel to measure
		PWM input
Period (output)	Numeric	PWM period, Unit is 0.1ms

# 4.6.5 PWM Definition

Folder: EcoCoder Blocks/Digital I/O

**Description:** 

	LSO09		
	LSO07 LSO08		
	LSO06		
	LS004		
0	LSO03		
OPWM Duty	LSO02		
	OPWM_Channel0	1	
1	OPWM_Channel0	0	
OPWM FRO	Hbridge3		
OPWM Enabl	Hbridge1		^
OPWM channel	Hbridge1		
Parameters			
PWM Definition	(mask) (link)		
DIOCK Paramet	Leis, P wivi Demindon		,
	Block Paramet PWM Definition Parameters OPWM channel OPWM Enable OPWM FRQ 1 OPWM Duty 0	▶       Block Parameters: PWM Definition         PWM Definition (mask) (link)         Parameters         OPWM channel         Hbridge1         Hbridge2         OPWM FRQ         Hbridge3         0PWM_Channel0         OPWM_Channel00         0PWM_Channel00         0PWM_Channel00         0PWM Duty         LS001         LS004         LS005         LS006         LS007         LS008	Block Parameters: PWM Definition         PWM Definition (mask) (link)         Parameters         OPWM channel         Hbridge1         Hbridge2         OPWM FRQ         Hbridge3         1         OPWM Channel01         OPWM LSO01         LSO02         0         LSO03         LSO05         LSO06         LSO07         LSO08

This block enables channels for PWM output, initializes the PWM output parameters for corresponding channels.

Channels with PWM output capability (H-bridge, LSO, HSO) can be found in the pinout table of VCU datasheet.

Parameter Field	Value	Comments/Description
OPWM channel	Drop-down list	Specify the channel for PWM output
ODW/M Enable	Chack how	If checked, enable PWM output
	able Check box	function of specified channel
	Theoretical range is 1-2000000 Hz.	
	Recommended frequency range for	
OPWM FRQ	M FRQ Numeric	perfect square wave output is 15 –
		2000 Hz.
	The unit for input value is configurable	

# **EC** TRON

#### EcoCoder Manual

		in the block PWM IO Frequency Range
		Definition.
		Control the duty cycle of the selected
OPWM Duty	Numeric	channel signal. Expected value is 0-
		10000, corresponding to 0-100%.

### 4.6.6 PWM Output

Folder: EcoCoder Blocks/Digital I/O

**Description:** 



This block configures PWM outputs.

Parameter Field	Value	Comments/Description
OPWM channel	Drop-down list	Specify the PWM output channel
freq (input)	Numeric	Theoretical range is 1-2000000 Hz.

# **EC**ITRON

		Recommended range for perfect
		square wave output is 15 – 2000 Hz.
		For the input value unit, refer to <i>PWM</i>
		IO Frequency Range Definition.
		Control the duty cycle of the selected
duty (input)	Numeric	channel output, value 0-10000
		corresponds to 0-100%.

# 4.6.7 WakeUp Input

### Folder: EcoCoder Blocks/Digital I/O

Description:

	Source Block Parameters: WakeUp Input	
	S-Function (mask) (link)	
DIRead:DI_WAKEUP1 bool >	Parameters	
WakeUp Input	Switch input channel DI_WAKEUP1	
	DI_WAKEUP2 DI_CANWAKE	
	QK DI_LINWAKE	

This block can read wake-up signal status.

Parameter Field	Value	Comments/Description
Switch input channel	Drop-down list	Wakeup source selection
Output	Numeric (Boolean)	'1' is active

# 4.6.8 H-bridge Definition

### Folder: EcoCoder Blocks/Digital I/O

### Description:

Hbridge Channel:Hbridge1 Hbridge Enable:1 Current Direction A->B:1 Hbridge Initial Frequency:1000 Hbridge Initial Duty Cycle:5000 Hbridge Definition	Block Parameters: Hbridge Definition         S-Function (mask) (link)         Parameters         Hbridge Channel: Hbridge1         Image: Hbridge Enable:         Image: Hbridge Current Direction A->B:         Hbridge Initial Frequency:         1000         Hbridge Initial Duty Cycle:         5000
	QK Cancel Help Apply

This block is used for setting up the VCU H-bridge(s).

Parameter Field	Value	Comments/Description
H-bridge Channel	Drop-down list	Select H-bridge channel
H-bridge Enable	Check box	If checked: Enable H-bridge
H-bridge Current Direction A->B	Check box	If checked: The current direction is from A->B. If not checked: the current direction is B->A.



		(A and B are the two outputs of H- bridge, see the VCU data sheet for more information)
H-bridge Initial Frequency	Numeric (Hz)	The theoretical range is 1-2000000 Hz. Recommended range for perfect square wave output is 15 – 2000 Hz. Input value unit is configurable in the <i>PWIM IO Frequency Range Definition</i> block.
H-bridge Initial Duty Cycle	Numeric	0-10000 corresponds to 0-100%.

# 4.6.9 H-bridge Output

### Folder: EcoCoder Blocks/Digital I/O

### Description:

	Block Parameters: Hbridge Output		
	Hbridge Output (mask) (link)		
en curA2B frq Hbridge1 duty	Parameters Hbridge Channel Hbridge1		
Hbridge Output	<u>O</u> K <u>Cancel H</u> elp <u>Apple</u>	ý	

This block controls H-bridge output.

Parameter Field	Value	Comments/Description
H-bridge Channel	Drop-down list	Select H-bridge channel
en (input)	Numeric (bool)	'1' to enable H-bridge


EcoCoder Manual

curA2B (input)	Numeric (bool)	'1': current flows from A to B;
		'0': currents flow from B to A.
		The theoretical range is 1-2000000
		Hz. Recommended range for perfect
frq (input)	Numeric	square wave output is 15 – 2000 Hz.
		Input value unit is configurable in the
		PWM IO Frequency Range Definition
		block.
duty (input)	Numeric	0-10000 corresponds to 0-100%.

## 4.6.10PWM IO Frequency Range Definition



This block defines the frequency range and accuracy of PWM IOs. If it is not implemented in the model, the accuracy will be default value, 1 Hz.

- 1. Frequency Range: Frequency range selection, changes in this option will alter the frequency range and accuracy of all the frequency related block in the model.
- Frequency Accuracy: Accuracy adjustment. Value in the box means the unit frequency for outputs/inputs of frequency related blocks. For example, if the Frequency Accuracy is 0.01 Hz, it means that when frequency related block outputs/inputs value is 5, the actual physical frequency value is 5x0.01Hz = 0.05 Hz.

## 4.6.11IPWM Read



#### **Block Parameter:**

Select IPWM Channel: PWM inputs channel selection

### **Block Outputs:**

- 1. freq: the input PWM frequency of the signal
- 2. duty: the input PWM signal duty cycle

## 4.7 LIN Communication

## 4.7.1 LIN Channel Definition

Folder: EcoCoder Blocks/LIN

## Description:

		_
	Block Parameters: LIN Channel Definition	×
	LIN Channel Definition (mask) (link)	
	Parameters	
	LIN Channel: LIN_A	•
LIN Channel:LIN_A LIN Mode:LIN_Mode_Master	LIN Mode: LIN_Mode_Master	•
Sync Break Field:13bit	LIN Baud Rate(bps):	
LIN Channel Definition	9600	:
	Sync Break Field: 13bit	•
	OK Cancel Help Apply	

This block provides configuration interface for LIN parameters.

Parameter Field	Value	Comments/Description
LIN Channel	Drop-down list	Please refer to datasheet to select supported LIN channel.
LIN Mode	Drop-down list	Select Lin mode. (Master or Slave)
LIN Baud Rate (bps):	Numeric	Input LIN baud rate.

## 4.7.2 LIN Get Status

## Folder: EcoCoder Blocks/LIN

## Description:

	Source Block Parameters: LIN Get Status	
	LIN Get Status (mask) (link)	
LIN Channel:LIN_A st ▶	Parameters	
LIN Get Status	LIN Channel: LIN_A	
	OK Cancel Help Apply	

This module is used to get the status of LIN channel.

### **Block Parameters**

Parameter Field	Value	Comments/Description
LIN Channel	Drop-down list	Please refer to datasheet to select supported LIN channel.

## **Block Output:**

1. St: (status)

Value	Description
0	Normal
1	Error
6	Busy



## 4.7.3 LIN Receive Date

### Folder: EcoCoder Blocks/LIN

## **Description:**

	🔁 Source Block Parameters: LIN Receive Data
	LIN Receive Data (mask) (link)
	Parameters
	LIN Channel: LIN_A
LIN Channel:LIN_A ID:0x0 Available >	ID:
Data Length:2	hex2dec('0')
Classic Checksum:Classic Data	Data Length:
Sample Time:Task_Inherit	2
LIN Receive Data	Direction:
	MasterSend •
	Classic Checksum: Classic 🔹
	Sample Time: Task_Inherit
	OK Cancel Help Apply

This module is used to receive data from the LIN bus.

Parameter Field	Value	Comments/Description
LIN Channel	Drop-down list	Please refer to datasheet to select supported LIN channel.
ID	Numeric	Data address to receive
Data Length	Numeric	Data length
Direction	Drop-down list	Select according to LIN mode
Classic Checksum	Drop-down list	Checksum category



## **Block Outputs:**

- 1. Available: output of 1 means data is valid, otherwise invalid.
- 2. Date: receiving data

## 4.7.4 LIN Transmit Data

### Folder: EcoCoder Blocks/LIN

### **Description:**

	Sink Block Parameters: LIN Transmit Data
	-LIN Transmit Data (mask) (link)
	Parameters
	LIN Channel: LIN_A
LIN Channel:LIN_A ID:0x0 Data Length:3	ID: hex2dec('0')
Driection:MasterSend Classic Checksum:Enhanced Sample Time:Task_Inherit	Data Length: 3
LIN Transmit Data	Driection:
	MasterSend 🔹
	Classic Checksum:
	Enhanced
	Sample Time: Task_Inherit
	OK Cancel Help Apply

This module is used to send data to the LIN bus.

Parameter Field	Value	Comments/Description
LIN Channel	Drop-down list	Please refer to datasheet to

# **EC** TRON

EcoCoder Manual

		select supported LIN channel.
ID	Numeric	Data address to receive
Data Length	Numeric	Data length
Direction	Drop-down list	Select according to LIN mode
Classic Checksum	Drop-down list	Checksum category
Sample Time	Drop-down list	Task sample time

## Block Input:

1. Date: sending data

## 4.8 Non-Volatile Memory Blocks

There are two types of Non-Volatile Memory. The first type is NVM, and the second type is Fixed NVM. Data stored in NVM will not be lost between power cycles. Data stored in Fixed NVM will not be lost after the VCU is programmed.

For more information, please refer to Appendix A.

## 4.8.1 NVM Definition

#### Folder: EcoCoder Blocks/Non-volatile Memory Blocks

#### Description:

	Block Parameters: NVM Definition1	×
Restore from factory default values in .mot	-NVM Definition Block (mask) (link)	
NVM Definition1	Parameters	
	Load NVM values in the following way	Restore from factory default values in .mot 🔻
		Load previous saved values in flash
		Restore from factory default values in .mot           DK         Cancel         Help         Apply

This block is used to initialize NVM variables and specify the NVM variable initialization method after every time the VCU being programmed by .mot file.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Load NVM value in the following way	Drop-down list	Load previous saved values in flash: The corresponding NVM variable value would be initialized from the NVM memory area, instead of .mot file. Restore from factory default values in .mot: The corresponding NVM variable value would be initialized from .mot file.

## 4.8.2 NVM Variable Definition

## Folder: EcoCoder Blocks/Non-volatile Memory Blocks

### Description:

	Block Parameters: NVM Variable Definition
	Subsystem (mask) (link)
	Parameters
	Variable Name
nvm:demo_st_nvmd	demo_st_nvmd
NVM Variable Definition	Initial Value
value:0 type:uint8 D:1	0
	Variable Type uint8
	Dimensions
	1
	Description
	OK Cancel Help Apply

This block is used to define regular NVM variables.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Variable Name	Alpha-numeric text	Variable name
Initial Value	Numoric	Initial value of the to-be
Initial value	Numeric	defined variable
Variable Type	Drop-down list	Select variable data type
Dimension	Numeric	Variable dimension
Unit	Alpha-numeric text	User-defined variable unit
Description	Alpha numeric toxt	User-defined variable
Description	Aipila-numeric text	description

## 4.8.3 Read NVM

### Folder: EcoCoder Blocks/Non-volatile Memory Blocks

## Description:

	Subsystem (mask) (link)
nvm:demo_trq_nvm     Read NVM     typesingle	Parameters variable_name <pre>demo_trq_nvm variable_type single</pre>
	OK Cancel Help Apply

This module is used for reading regular NVM variables.

# ECOTRON

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Variable_name	Alpha-numeric text	Specify variable name
Variable_type	Drop-down list	Variable data type

### Block Output:

The NVM variable value

## 4.8.4 Write NVM

### Folder: EcoCoder Blocks/Non-volatile Memory Blocks

**Description:** 

	Sink Block Parameters: Write NVM			
	Subsystem (mask) (link)			
	Parameters Variable Name demo_st_nvmr Variable Variable Definition			
	Initial Value			
nvm:demo_st_nvmr	0			
Write NVM	Variable Type uint8			
value:0 type:uint8 D:1	Dimensions			
	1			
	Unit			
	Description			
	OK Cancel Help Apply			

This module is used for writing regular NVM variables into RAM. To save changed variables into VCU flash between power cycles, users need to use another block 'Store All NVM Data'.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Variable Name	Alpha-numeric text	NVM variable name
		If checked:
Enable Variable Definition	Check box	Define and write NVM variable
Enable Variable Definition		If not checked:
		Only write NVM
Initial Value	Numeric	NVM variable initial value (for
		NVM variable definition)
Variable Type	Drop-down list	NVM variable type
Dimension	Numeric	Dimension of NVM variable
Unit	Alpha-numeric text	User-defined variable unit
Description	Alpha-numeric text	Memo

## 4.8.5 Fixed NVM Definition

Folder: EcoCoder Blocks/Non-volatile Memory Blocks

Description:

Order	Name	Туреі	Size I	nit		v	alu	e
1	Fnvm_double	double	11	01[				1]
2	Fnvm_int8	int8	2	1 [			1	2]
31	Fnvm_uint8	uint8	11	010				1]
4	Fnvm_boolean	boolean	4	010	1	2	3	4]
51	Fnvm uint16	uint16	2	1 [			1	2]
61	Fnvm int16	int16	2	1 [			1	2]
71	Fnvm single	single	11	010				1]
81	Fnvm int32	int32	11	010				1]
91	Fnvm uint32	uint32	41	110	1	2	3	41

Block Parameters: Fixed NVM Definition
S-Function (mask) (link)
Parameters
Select the m file:
ecocoder_getFixedNvmList
<u>OK</u> <u>Cancel</u> <u>H</u> elp <u>A</u> pply

This module is used to define and initialize fixed NVM variables.

This Fixed NVM Definition block will only be executed once, during the first power-up

process of VCU application software, every time after the .mot file being flashed into VCU.

**Block Parameters** 

Parameter Field	Value	Comments/Description	
Select the m file	.m file	Select the .m file defining	
		NVM variables.	

\*The m file can be defined as the picture below:

```
Function NVMList=ecocoder_getFixedNvmList()
NVMList={...
struct('name', {'Fnvm_double'},'type', {'double'},'size', 1,'init', 0,'value',1), ...
struct('name', {'Fnvm_int8'},'type', {'uint8'},'size', 2,'init', 1,'value',[1 2]), ...
struct('name', {'Fnvm_boolean'},'type', {'uint8'},'size', 4,'init', 0,'value',1), ...
struct('name', {'Fnvm_uint16'},'type', {'uint16'},'size', 2,'init', 1,'value',[1 2]), ...
struct('name', {'Fnvm_uint16'},'type', {'uint16'},'size', 2,'init', 1,'value',[1 2]), ...
struct('name', {'Fnvm_uint16'},'type', {'uint16'},'size', 2,'init', 1,'value',[1 2]), ...
struct('name', {'Fnvm_uint16'},'type', {'int16'},'size', 1,'init', 0,'value',[1 2]), ...
struct('name', {'Fnvm_single'},'type', {'int32'},'size', 1,'init', 0,'value',1), ...
struct('name', {'Fnvm_uint32'},'type', {'uint32'},'size', 4,'init', 0,'value',[1 2 3 4]), ...
}:
end
```

The .m file needs to be added under MATLAB path. The 'init' in the .m file is the flag for NVM variable initialization.

init = 1: The corresponding NVM variable value(s) will be loaded from .mot file during the first time of VCU starting process, every time after .mot file being flashed into VCU.

init = 0: The corresponding NVM variable value(s) will be loaded from original VCU NVM memory block during the first time of VCU starting process every time after .mot file being flashed.

If the VCU that you are operating is a brand new VCU and will be flashed for the very first time, no matter what the 'init' value is, the NVM variables will be initialized from .mot file.

## 4.8.6 Read Fixed NVM

## Folder: EcoCoder Blocks/Non-volatile Memory Blocks

### **Description:**

	Source Block Parameters: Read Fixed NVM			
	S-Function (mask) (link)			
R	Parameters			
Fnvm_int8	Fixed NVM Varible:			
Read Fixed NVM	Fnvm_int8			
	OK Cancel Help Apply			

This module is used for reading fixed NVM variables.

#### **Block Parameters**

Parameter Field	Value	Comments/Description	
Fixed NVM Variable	Variable name	Specify the variable name	
		to be read	

## **Block Output:**

The value of corresponding variables.

## 4.8.7 Write Fixed NVM

## Folder: EcoCoder Blocks/Non-volatile Memory Blocks

## Description:

	Sink Block Parameters: Write Fixed NVM	
	S-Function (mask) (link)	
	Parameters	
Fnvm_int8	Fixed NVM Varible:	
Write Fixed NVM	Prom_int8	
	OK Cancel Help Apply	

This module is used for writing fixed NVM variables.

Parameter Field	Value	Comments/Description
Fixed NVM Variable	Variable name	Specify the variable to be
		written.

## 4.8.8 Store All NVM Data

#### Folder: EcoCoder Blocks/Non-volatile Memory Blocks

#### **Description:**

	Sink Block Parameters: Store All NVM Data
	Subsystem (mask)
<u>4</u>	Store NVM Data
Store All NVM Data	Parameters Select Trigger Type rising
Store All NVM Data	
	OK Cancel Help Apply

When this module is triggered, all NVM variable data will be written from RAM to flash,

so that the NVM data will be stored in the VCU.

It is recommended to call this block before VCU power-off. And please do not call this block too frequently. For example, if a 5ms task is assigned to this block, flash would quickly burn out because flash memory blocks have life span, and frequent programming/erasing of memory block will cause program/erase cycles running out.

Parameter Field	Value	Comments/Description
Select Trigger Type	Drop-down list	Select trigger type

## 4.8.9 Restore All NVM Data

## Folder: EcoCoder Blocks/Non-volatile Memory Blocks

## **Description:**

	🛐 Sink Block Parameters: Restore All NVM Data		
	-Subsystem (mask)		
	Restore NVM Data		
Restore All NVM Data	Parameters		
÷	Select Trigger Type rising 💌		
Restore All NVM Data			
	OK Cancel Help Apply		

The module reads NVM data from ROM (flash) back to RAM.

It is recommended to call this block when VCU powers on. This block can be triggered by 'Task\_ini'.

Parameter Field	Value	Comments/Description
Select Trigger Type	Drop-down list	Select trigger type

## 4.9 Diagnostic Blocks

Diagnostic blocks are designed to realize VCU diagnostic functions.

## 4.9.1 Hardware Output DTC

#### Folder: EcoCoder Blocks/Diagnostic Blocks

**Description:** 



This block can realize the hardware diagnosis of supported LSO, HSO and H-bridge.

Please refer to VCU datasheet for the channels that support diagnostic functions.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
DTC_Channel	Drop-down list	Select hardware channel

#### Block Output:

DTC: The diagnostic trouble code of the specified channel.

## 4.9.2 DTC Parser

#### Folder: EcoCoder Blocks/Diagnostic Blocks

#### Description:



This block can help parse out specific fault of DTC.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Index of bit	Drop-down list	Select the fault to be analyzed

#### **Block Input:**

DTC: The diagnostic trouble code.

#### **Block Output:**

bool: If the output value is one, the specific fault selected in the Block Parameter happened; If output value is 0, the fault did not happen.



## 4.9.3 Software Core Diagnostic

## Folder: EcoCoder Blocks/Diagnostic Blocks

## **Description:**

👔 Source Block Parameters: Software Core Diagnostic
S-Function (mask) (link)
=0:No fault !=0:Fault
Parameters
DTC_Channel ROM
ROM
RAM
MainChip
SalfChashill

This block provides memory/chip fault check.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
DTC Channel	Drop-down list	Select the memory or chip
		to be diagnosed.

## 4.9.4 Clear H-bridge DTC

### Folder: EcoCoder Blocks/Diagnostic Blocks

Description:

Sink Block Parameters: Clear Hbridge DTC		
	Subsystem (mask) (link)	
	Parameters Hbridge Channel: Hbridge1	
Clear Hbridge DTC	OK Cancel Help Apply	

This block can clear the H-bridge channel faults, the trigger type to trigger this block is



rising edge.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
H-bridge Channel	Drop-down list	Select the channel of H-bridge

## 4.10 Calibration & Measurement

## 4.10.1 Calibration Definition

#### Folder: EcoCoder Blocks/Calibration & Measurement

## **Description:**



value:0 type:single max:[] Min:[]

Block Parameters: Calibration Defin	e X
Subsystem (mask) (link)	
Parameters	
variable_name	
demo_trq_val	
variable_value	
0	
variable_type single	▼ >>
variable_min	variable_max
[]	[]
Unit	
Description	
	Cancel Help Apply

This block can help define and initialize calibration variable.

Parameter Field	Value	Comments/Description
variable_name	Alpha-numeric text	Calibration variable name
variable_value	Numeric	Calibration variable initial value

# **EC**ITRON

variable_type	Drop-down list	Calibration variable data type
variable_min	Numeric	Calibration variable lower limit
variable_max	Numeric	Calibration variable upper limit
Unit	Alpha-numeric text	User-defined calibration variable unit
Description	Alpha-numeric text	Memo

## 4.10.2 Read Calibration

## Folder: EcoCoder Blocks/Calibration & Measurement

### Description:

	Source Block Parameters: Read Calibration	
	Subsystem (mask) (link)	
	Parameters	
	variable_name	
	demo_trg_val	
	variable_value	
	0	
demo_trq_val	variable_type single	
value:0 type:single max [] Min:[]	variable_min variable_max	
	Unit	
	Description	
	OK Cancel Help Apply	

This block defines and reads calibration variables.

Parameter Field	Value	Comments/Description
variable_name	Alpha-numeric text	Calibration variable name
variable_value	Numeric	Calibration variable initial value
variable_type	Drop-down list	Calibration variable data type
variable_min	Numeric	Calibration variable lower limit



variable_max	Numeric	Calibration variable upper limit
Unit	Alpha-numeric text	User-defined calibration variable unit
Description	Alpha-numeric text	Memo

### **Block Output:**

Calibration variable value.

## 4.10.3 Write Measurement

## Folder: EcoCoder Blocks/Calibration & Measurement

### Description

	🔁 Sink Block Parameters: Write Measurement	×
	Write Measurement (mask) (link)	
	Parameters variable_name	
	demo_trq	
demo_trq Write Measurement type:single Dimensions:1	variable_type single >> >> Dimensions 1 Unit	
	Description OK Cancel Help App	ly

This block can help define measurement variables.

Parameter Field	Value	Comments/Description
variable_name	Alpha-numeric text	Measurement variable name
variable_type	Drop-down list	Variable data type



Dimensions	Numeric	The dimension of measurement
		variable
Unit	Alpha-numeric text	User-defined measurement variable
		unit
Description	Alpha-numeric text	Memo

### **Block Input:**

To-be-measured variable.

## 4.10.4 Write and Read Measurement

## Folder: EcoCoder Blocks/Calibration & Measurement

## Description:

		_
	Tunction Block Parameters: Write and Read Measurement	×
	Write and Read Measurement (mask) (link)	
	Parameters	
	variable_name	
	demo_trq	]
	variable_type single >>	
domo tra	Dimensions	
Write and Read Measurement	1	]
type single Dimensions:1	Unit	
type.ongle Dimensione.		
	Description	
	OK Cancel Help Apply	

This block is an inline block, it helps read measurement variables.

Parameter Field	Value	Comments/Description
variable_name	Alpha-numeric text	Measurement variable name

# ECITRON

variable_type	Drop-down list	Variable data type
Dimensions	Numeric	The dimension of measurement variable
Unit	Alpha-numeric text	User-defined measurement variable
om		unit
Description	Alpha-numeric text	Memo

### Block Input:

To-be-measured variable.

#### **Block Output:**

Same as input. (This block is used for variable measurements, it does not change variable values).

## 4.10.5 Override Probe

This block is used for overriding signal values for calibration.

		Function Block Parameters: Override Probe         X           Override Probe (mask) (link)
>	Test_flgOvrCal_val = 0 Test_flgOvrEn_val = 0 Test_flgOvrMsr	Parameters Base Variable Name Test_flg Override Calibration Initial Value 0 Override Enabled Initial Value 0
	Override Probe	Data Type single     >>       Unit

In calibration software, 'Variable\_nameOvrCal\_val' is calibration variable, 'Variable\_nameOvrMsr' is the measurement variable, 'Variable\_nameOvrEn\_val' is the

# ECITRON

control signal – when control signal is '1', the calibration variable will override the original signal that passes through the block, and the block output will be the calibration variable value. When the control signal is '0', this block will not override the passing-through signal, the measurement variable will have the same value as block input and block output would be the same as the block input.

### **Block Parameters:**

- 1. Base Variable Name: user-defined name of the overriding variable.
- 2. Override Calibration Initial Value: initial value of the calibration variable.
- 3. Override Enable Initial Value: initial value of control signal.
- 4. Data Type: data type of calibration variable.
- 5. Unit: user-defined measurement variable unit
- 6. Description: the description of the variable.

#### Block Input:

Variable to be overridden.

Output:

If control signal is 1, the output of the block would be the overriding calibration variable value;

If the control signal is 0, the output of the block would be the same as the input. (No overriding)

# 4.10.6 1-D Lookup Table

## Folder: EcoCoder Blocks/Calibration & Measurement

## **Description:**

	The Function Block Parameters: 1-D Lookup Table
	Subsystem (mask) (link)
	Parameters
	Name:
	demo_trq1
	Breakpoint type single
1-D Lookup Table	Breakpoint data:
	[1 2 4]
	Table type: single
	Table data:
	[3 6 7]
	OK Cancel Help Apply

This block defines 1-D look-up table. 1-D look-up table supports online calibration.

Parameter Field	Value	Comments/Description
Name	Alpha-numeric text	Variable name
Breakpoint type	Drop-down list	Variable type
Breakpoint data	Numeric (Matrix)	Breakpoint data
Table type	Drop-down list	Table variable type
Table data	Numeric (Matrix)	Table data

## 4.10.7 2-D Lookup Table

## Folder: EcoCoder Blocks/Calibration & Measurement

### **Description:**

	Function Block Parameters: 2-D Lookup Table       Subsystem (mask) (link)
,	Parameters Name: demo_trq2 Breakpoints 1 type: single  >> Breakpoints 1 data(Row):
2-D Lookup Table	[1 2 3] Breakpoints 2 type: single    See >> Breakpoints 2 data(Columm): [1 2 3 5]
	Table type:     single     >>       Table data:     [2 4 4 3 ;2 5 4 3;2 4 5 8]
	OK Cancel Help Apply

This block defines 2-D look-up table, 2-D look-up table supports online calibration.

Parameter Field	Value	Comments/Description	
Name	Alpha-numeric text	2-D look-up table name	
Breakpoints 1 type	Drop-down list	Breakpoints 1 variable data type	
Breakpoints 1	Numoric (Matrix)	Ducely einte 1 verieble dete	
data(Row)	Numeric (Matrix)		
Breakpoints 2 type	Drop-down list	Breakpoints 2 variable data type	
Breakpoints 2	Numoric (Matrix)	Brockpoints 2 variable data	
data(Column)	Numeric (Matrix)	breakpoints 2 variable data	
Table type	Drop-down list	Select table variable data type	
Table data	Numeric (Matrix)	Initialize table data	

## 4.10.8 Calibration Data Check

This module is used for checking the calibration data at the VCU power-on process. If there is any corrupted calibration data, the controller software will enter an infinite loop to avoid potential catastrophic results due to corrupted calibration data.

	Source Block Parameters: Calibration Data Check	
	Override Probe (mask) (link)	
	Parameters	
Enable Calibration Data Check : on f()	Enable Calibration Data Check	
Calibration Data Check	OK Cancel Help Apply	

#### **Block Parameters:**

Enable Calibration Data Check: If checked: enable the function.

Output:

f() : Flag for checking calibration data. If there is a problem with the calibration data, the flag will be set to 1. This signal can be used as a trigger signal.

## 4.11 System Management Blocks

## 4.11.1 Power Management Example

Folder: EcoCoder Blocks/System Management Blocks

Description:

	Block Parameters: Power Management Example	x
	Power Management Example (mask) (link)	
	Parameters	
	Key Signal From Input	
	Power Off Delay(S):	
Key Cycle Reset Enable:1	0.5	
Key Cycle Of Duration(s):0.1 Key Cycle On Duration(s):0.1	Key Cycle Reset Enable:	
Key Off Threshold Volt(V):3 Key On Hysteresis Volt(V):6	1	
Power Off Fail Enable Reset:0 Enable NVM Store And Restore:1	Key Cycle Off Duration(S):	
₽	0.1	
Power Management Example	Key Cycle On Duration(S):	
	0.1	
	Key AD2Volt Factor:	
	109/27846	
	Key Off Threshold Volt(V):	
	3	
	Key On Hysteresis Volt(V):	Ŧ
	OK Cancel Help A	pply

This block integrates power-off logic control and operations. It can be regarded as a reference/demo design of VCU power-off logic. Users are encouraged to understand the block first by looking down mask and then make necessary modification to the block for customized implementations.

Parameter Field	Value	Comments/Description	
Key Cignel from Inc.	Charlehov	If checked, the key switch signal	
Key Signal from input	Check box	can be read from VCU input.	
Power Off Delay(S)	Numeric	Power off delay time	
		If set to 0, VCU will not be reset if	
		key switch turns back on before	
Kay Cycla Dasat Enabla		VCU power-off delay period ends.	
Rey Cycle Reset Enable	Numeric (Boolean)	If set to 1, VCU will be reset if key	
		switch turns back on before VCU	
		power-off delay ends.	
	Numeric	The duration (time threshold) after	
Kow Cycle Off Duration (S)		the key-off moment (KeyOn signal	
key Cycle Off Duration(S)		absent) to the time when VCU	
		starts power-off process.	
		VCU starts power-up process if	
Key Cycle On Duration(S)	Numeric	KeyOn signal is detected for more	
		than this time threshold.	
		The factor to be multiplied that	
Key AD2Volt Factor	Numeric	convert AD to voltage, see section	
		4.2.1 for details.	
		If the input KeyOn voltage is less	
Key Off Threshold Volt(V)	Numeric	than this value, KeyOn signal is	
		interpreted as '0'.	
Key On Hysteresis Volt(V)	Numeric	If the KeyOn input voltage is larger	



		than the sum of 'Key Off Threshold
		Volt' and this hysteresis value,
		KeyOn signal is '1'.
		If this setting is '1', VCU would keep
Power Off Fail Enable	Numeric (Peeleen)	trying to power off at certain
Reset	Numeric (Boolean)	frequency when power-down
		process fails.
Enable NVM Store and	Chack box	If Checked: Enable NVM control
Restore	Check box	option.
Set The Waiting Time(ms)	Numeric	Power-off delay time

## 4.11.2 Shutdown Power

## Folder: EcoCoder Blocks/System Management Blocks

## Description:

	Block Parameters: Shutdown Power X	
Shutdown Power	Shutdown Power (mask) (link)	
	Parameters	
÷	Select Trigger Type rising •	
Shutdown Power		
	OK Cancel Help Apply	

This block can be called to start VCU power-off process.

### **Block Parameters**

Parameter Field	Value	Comments/Description
Soloct Triggor Typo	Drop.down.list	Block trigger signal type
Select mgger Type	Drop-down list	selection.
Cot The Meiting Time (me)	Numerie	Set the time of power-off
Set the waiting time(ms)	Numeric	delay waiting time.

## 4.11.3 Set ECU Mode

### Folder: EcoCoder Blocks/System Management Blocks

**Description:** 

	Block Parameters: Set ECU Mode	
	Subsystem (mask) (link)	
	Parameters	
	Trigger type rising -	
Mode:MasterChipSleep	Select ECU Mode MasterChipSleep	
Set ECU Mode		
	OK Cancel Help Apply	

This module can set the working mode of ECU.

Parameter Field	Value	Comments/Description
Trigger Type	Drop-down list	Block trigger signal type
		selection.
Select ECU Mode	Drop-down list	Work mode selection

# 4.11.4 ECU Master Chip Wake-Up Definition

## Folder: EcoCoder Blocks/System Management Blocks

### **Description:**

	Block Parameters: ECU Master Chip WakeUp Definition		
	ECU WakeUp (mask) (link)		
	Configure the WakeUp Channel of the ECU.		
	Parameters		
	CAN Channel CAN_A		
Channel = CANLA	Wakeup En CAN R		
Wakeup = off	CAN_D CAN_C		
ECU Master Chip WakeUp Definition	<u>O</u> K <u>C</u> ancel <u>H</u> elp <u>A</u> pply		

The block specifies the CAN channel that wakes up the VCU.

Parameter Field	Value	Comments/Description
Trigger Type	Drop-down list	Wake-up CAN channel
		selection
Wakeup Enable	Check box	If checked: the specified
		CAN channel can wake up
		VCU.

# 4.11.5 Watchdog Definition

## Folder: EcoCoder Blocks/System Management Blocks

### **Description:**

	Block Parameters: Watchdog Definition
	S-Function (mask) (link)
SWDG:on Wait Time:500 HWDG:off	Parameters
	🗹 Software Watchdog Enable:
	Software Watchdog Wait Time(ms):
Watchdog Definition	500
	🗌 Hardware Watchdog Enable:
	OK Cancel Help Apply

Settings for software watchdog and hardware watchdog.

Parameter Field	Value	Comments/Description
Software Watchdog Enable	Check box	If checked: Software watchdog is
		enabled.
Software Watchdog Wait Time(ms)	Numeric	The 'feeding dog' operation is
		executed at Task_L1ms, software will
		reset when timeout.
Hardware Watchdog Enable	Check box	If checked: Hardware watchdog
		enabled.
		(If this icon is greyed out, the specified VCU has
		no hardware watchdog built in)

## 4.11.6 Software Reset

## Folder: EcoCoder Blocks/System Management Blocks

### **Description:**

	Sink Block Parameters: Software Reset
A A	Subsystem (mask) (link) Software Reset
Software Reset	Parameters Select Trigger Type rising 🔹
	OK Cancel Help Apply

This block is used for triggering VCU software reset. If this block is called, the VCU software

will be reset immediately.

Parameter Field	Value	Comments/Description
Select Trigger Type	Drop-down list	Block trigger signal type
		selection.
## 4.11.7 Read System Free Counter

#### Folder: EcoCoder Blocks/System Management Blocks

#### **Description:**



By calling the block, VCU main controller 32-bit free-running counter value will be read. The value can be used to calculate time interval between certain events or to generate random numbers, etc.

#### **Block Output:**

System free counter value.

#### 4.11.8 Power Control Output

#### Folder: EcoCoder Blocks/System Management Blocks

Description:

	Sink Block Parameters: Power Control Output		
	Power Control Output (mask) (link)		
	Parameters		
Power Channel:PWR5V2	Power Control Channel PWR5V2		
Deverse Constant Output	PWR5V2		
Power Control Output	PWR5V3		
	OK PWR5V4		
	PWR5V5		
	PWR12V_DRVP		

#### **Block Parameters**

Parameter Field	Value	Comments/Description
Power Control Channel	Drop-down list	Power channel selection.
input	Boolean	1: turning on power for
		corresponding channel.
		0: turning off power for
		corresponding channel.

## 4.11.9 Service Software Watchdog

#### Folder: EcoCoder Blocks/System Management Blocks

#### Description:

Software watchdog is used for resetting VCU software if the watchdog internal counter times out.

To enable this block, simply drag this block into your application software and schedule it as a low priority task using task scheduler. Every time this block being triggered by task scheduler will be taken as 'feed dog'. As a result, the scheduling period should be less than software watchdog timeout threshold.

Service Software Watchdog

Service Software Watchdog

## 4.12 CCP

This block set includes CCP related implementations.

## 4.12.1 Fixed CCP Slave Definition

#### Folder: EcoCoder Blocks/CCP

#### **Description:**

	Block Parameters: Fixed CCP Slave Definiton
	CCP21 Slave Definiton (mask) (link)
CCP Type: Configurable Enable CCP: on CAN Channel:CAN_C Station Address(Intel): 0x112 CRO ID: 0x100 DTO ID: 0x101 DAQID: [hex2dec(101') hex2dec(101') hex2dec(102')] DAQ First PID: [hex2dec(0') hex2dec(3C') hex2dec(78') hex2dec(0')] DTO Length: [hex2dec(0') hex2dec(20') hex2dec(30') hex2dec(30')] DAQ Period(ms): [5 20 100 0] Fixed CCP Slave Definiton	Parameters         CCP Type Configurable         ✓ CCP Enable         CAN Channel CAN_C         Station Address(Intel)         hex2dec('0112')         Command Receive Object(CRO) ID         hex2dec('100')         Data Transmit Object(DTO) ID         hex2dec('101')         Data Acquisition (DAQ) ID         [hex2dec('101') hex2dec('101') hex2dec('102')]         Data Acquisition (DAQ) First PID         [hex2dec('0') hex2dec('3C') hex2dec('78') hex2dec('0')]         Data Acquisition (DAQ) Length         [hex2dec('0') hex2dec('20') hex2dec('30') hex2dec('30')]         Data Acquisition (DAQ) Period(ms)         [5 20 100 0]

ОК

Cancel

Help

Apply

This block is used for setting up CCP related slave station definitions

Parameter Field	Value	Comments/Description
ССР Туре	Drop-down list	Simple:
		Under this option, the only
		configurable parameter in this block is
		CAN channel. After building, users will

#### **Block Parameters**



		get A2L, CAL and Mot (or Hex) file. To
		use EcoCAL, both A2L and CAL files
		need to be loaded.
		Configurable:
		Under this option, after building, users
		will get A2L and Mot (or Hex) file. To
		use EcoCAL, A2L and Mot (or Hex) files
		are needed.
CCP Enable	Check box	CCP enable
CAN Channel	Drop-down list	Specify CAN channel for CCP
Station Address(Intel)	Numeric	CCP Station Address
Command Receive	Numeric	Specify Command Receive Object
Object(CRO) ID	Numeric	(CRO) ID (Master->Slave)
Data Transmit Object(DTO)	Numeric	Data Transmit Object (DTO) ID (Slave
ID	Numeric	-> Master)
Data Acquisition (DAQ) ID	Numeric	CCP DAQ ID
Data Acquisition (DAQ) PID	Numeric	The first PID in the DAQ list.
Data Acquisition (DAQ)	Numoric	DAQ list longth
Length	Numeric	DAQ IISTIELIBUI
Data Acquisition (DAQ)	Numoric	CCP DAO pariod
Period(ms)	Numeric	

## 4.12.2 CCP/CAL Seed&Key Security Definition

This block is used to add authentication for VCU program change or calibration, the encryption algorithm of which can be customized. It also generates DLL file based on the user-provided seeds.

	SeedKeyInputTxt	_ C X
	Enable Custom Algorithm     boolean_T EcoCoder_Seed2Key(uint8_T *Seed,uint16_T SizeSeei     uint16_T MaxSizeKey,uint16_T	d,uint8_T *Key, *SizeKey)
	uint8_T i=0; for(i=0;i <maxsizekev;i++)< th=""><th>*</th></maxsizekev;i++)<>	*
CCP CAL Seed&Key Security Definiton	{ Key[i]=Seed[i]+3; } *SizeKey=MaxSizeKey;	
CCP CAL Seed&Key Security Definiton	return 1;	
		~
	ОК	Cancel

## 4.12.3 CCP DAQ Seed&Key Security Definition

This block is used to add authentication to the data measurement and calibration, the algorithm of which can be customized. It also generates DLL file based on the seeds.



## 4.12.4 CCP PGM Seed&Key Security Definition

This block is used to add authentication to the data flash, the algorithm of which can be customized. It also generates DLL file according to the seeds. The DLL file name is the name of MOT file plus "\_PGM".



## 4.12.5 CCP Get Seed Trigger

This is getting seed task trigger block. It is valid when the *CCP Type* of the *Fixed CCP Slave Definition* block is chosen "Configurable". Please refer to the *CCP Generate Seed Demo* block for details.



## 4.12.6 CCP Set Seed

This is setting seed block. It is valid when the CCP Type of the *Fixed CCP Slave Definition* block is chosen "*Configurable*". Please refer to the *CCP Generate Seed Demo* block for details.



## 4.12.7 CCP Generate Seed Demo

This is a demo block for generating seeds. It is valid when the CCP Type of the *Fixed CCP Slave Definition* block is chosen to be "Configurable". This module is implemented using the CCP Get Seed Trigger and CCP Set Seed blocks, it can be used as a demonstration Seed&Key function.



## 4.13 Programming Blocks

## 4.13.1 Online Programming Definition

Folder: EcoCoder Blocks/CCP

Description:

	EcoFlash Online Programming Definition (mask) (link)	
	Parameters	
	CAN Channel CAN_A	
CAN A:on	🗵 Enable	
Station Address(Intel):0x235 CRO ID(11-bit):0x100	Station Address(Intel)	
	hex2dec('235')	
D101D(11-bit)0x101	Command Receive Object(CRO) ID(11-bit)	
	hex2dec('100')	
Online Programming Definition1	Data Transmit Object(DTO) ID(11-bit)	
	hex2dec('101')	

This block is used for the online programming parameter definition. Note that this block

can only work with some specific VCU models. If the VCU that the user is working on does not support this function, there will be a pop-up notice when using this block.

Users can use more than 1 of these blocks in their Simulink model to make multiple CAN channels support online programming function.

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Select CAN Channel	Drop-down list	Online programming CAN channel selection.
Enable KeyCycle Program	Check box	If checked: programming will require key cycle to start programming.
Enable Online Program	Check box	If checked: VCU programming will not require a key cycle.
Station Address (Intel)	Greyed out	This value cannot be changed for now.
Command Receive Object (CRO) ID (11-bit)	Alpha-numeric text	Specify Command Receive Object (CRO) ID (Master->Slave) It is recommended to remain as default.
Data Transmit Object (DTO) ID (11-bit)	Alpha-numeric text	Data Transmit Object (DTO) ID (Slave -> Master) It is recommended to remain as default.

Note: if you have more than 1 VCU on one CAN bus and you want to flash one of them, you need to change the CRO and DTO both in this block and in EcoFlash to make them the same, like shown below.

	Block Paramete	rs: Online Programming Definition	×
	- EcoFlash Online I	Programming Definition (mask) (link)	
	Parameters		
	CAN Channel C/	AN_C	•
	Enable KeyCyl	e Program	
Channel:CAN_C Enable KeyCyle Program:on Enable Online Program:off	Enable Online	Program	
Station Address(Intel):0x235 CRO ID(11-bit):0x100 DTO ID(11-bit):0x101	Station Address(	Intel)	
Online Programming Definition	hex2dec('235')		:
	Command Receiv	ve Object(CRO) ID(11-bit)	
	hex2dec('77')		
	Data Transmit O	bject(DTO) ID(11-bit)	
	hex2dec(*/8')		:
		OK Cancel Help	Арріу
EcoFlash V1.1.1.8 Help     Settings     CAN Settings     Device Type Kvaser     Device Index 0     Channel No. 1     Baud Rate 500kbs     C ALM     C     BuildChkSum	AN  Flash Open  How to f Step 1: P Step 2: C C C C C C C C C C C C C C C C C C C	File Flash Cancel ash the \$19/Mot/Hex file?	× × 29bit
	Step 5: C	DTO ID(Hex): 78	29bit
Open Device Cl	ose Device		
		OK	Cancel

## 4.13.2 Programming Seed&Key Definition

The SeedKey function prevents unknown mot/hex files from being flashed to the VCU. The customer can use this block to define their own "password". The DLL file and the MOT file will be generated at the same time when the building and compilation process is finished. The DLL file can be loaded in EcoFlash to authenticate the flashing process. Without this DLL file you generate, the data on VCU cannot be erased via EcoFlash. Please refer to the EcoFlash manual for more details.

This block can define and modify the flashing key. If the parameters keep unchanged, the default parameters will be used for flashing (0x12-0x34-0x56-0x78-0x9A-0xBC). If changed, the new parameters in the module will be used for generating the DLL file.

# ECITRON

#### EcoCoder Manual

	SeedKeyInputTxt
	Enable Custom Algorithm
	Parameter 1
	hex2dec('12')
	Parameter 2
	hex2dec('34')
KeyWord:0x12-0x34-0x56-0x78-0x9A-0xBC	Parameter 3
	hex2dec('56')
	Parameter 4
Programming SeedKey Definition	hex2dec('78')
	Parameter 5
	hex2dec('9A')
	Parameter 6
	hex2dec('BC')
	OK Cancel

SeedKeyInputTxt	
Enable Custom Algorithm boolean_T EcoCoder_Seed2Key(uint8_T *Seed,uint16_T Si uint16_T MaxSizeKey,uin	zeSeed,uint8_T *Key, it16_T *SizeKey)
uint8_T i=0; for(i=0;i <maxsizekey;i++) { Key[i]=2; } *SizeKey=MaxSizeKey; return 1;</maxsizekey;i++) 	
} OK	Cancel

Parameters:

1. Enable Custom Algorithm: Enable the user-defined algorithm. If enabled, you can use a piece of C code to define key algorithm. If not, the key will be defined by setting the parameters.

2. Parameter1 - Parameter6: Key flash setting, 6-byte key.

After automatically compiling and generating code, system will generate DLL file for key flash. EcoFlash will load the DLL file to match the key, if successfully matched, the flash is authenticated. Please refer the EcoFlash manual for details.

T\_EV2106B01\_Program\_1.a2l
 T\_EV2106B01\_Program\_1.cal
 T\_EV2106B01\_Program\_1.mot
 T\_EV2106B01\_Program\_1\_PG.dll

## 4.13.3 Entry UDS Programming

Enter the UDS programming mode by this block, which enables controller to update the program through the UDS protocol.



Parameter Field	Value	Comments/Description
Select trigger type	Drop-down list	Trigger type selection.

## 4.14 Sensors Blocks

## 4.14.1 Read Gyro Hex Value

Note that this block can only work with some specific VCU models. If the VCU that the user is working on does not support this function, there will be a pop-up notice when using this block.

The module reads the Hex values of the angular acceleration of the three axes of the gyroscope and outputs the raw data in uint16.

Raw data need to be multiplied by a slope of 0.05 and an accounted for an offset of - 819.15 to give the acceleration physical value in deg / sec.



## 4.14.2 Read Gyro Phy Value

Note that this block can only work with some specific VCU models. If the VCU that the user is working on does not support this function, there will be a pop-up notice when using this block.

The module reads the angular acceleration of the three axes of the gyroscope and the output is physical value in deg / sec, data type is single.



## 4.15 Advanced Data Blocks

**OTP** One-Time Programmable memory

#### 4.15.1 Read OTP



#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Start Position Offset (Byte)	Numeric	Start address offset value
Data Length (Byte)	Numeric	Data length (the number of bytes the data takes)

#### **Block Outputs:**

data: The data read from OTP area.

st: Data reading status, 0 stands for data reading successfully.

## 4.15.2 Read OTP (Input port)



#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Data Length (Byte)	Numeric	The number of bytes that the data takes in OTP area.

#### Block Input:

Offset (byte): the start position offset.

#### **Block Output:**

data: The data that has been read from OTP area.

st: Data reading status, 0 stands for data reading successfully.

## 4.15.3 Write OTP



#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Start Position Offset (8Bytes)	Numeric	Start address offset, the unit is 8 bytes.
Select Trigger Type	Drop-down list	Trigger type selection.
Data Length (Byte)	Numeric	The length of the data to be written.
Data (Block input)		The data to be written in

EcoCoder Manual

		OTP area.
st (Block output)	boolean	Data write status, 0 stands for writing successful; Non- 0 value stands for writing unsuccessful.

# 4.15.4 Write OTP (Input port)



Function Block Parameters: Write OTP(Input port)		
Write OTP (mask) (link)		
Parameters		
Select Trigger Type rising		
Data Length(Byte)		
30		
OK Cancel Help Apply		

#### **Block Parameters:**

Parameter Field	Value	Comments/Description
Copyright ECOTRONS LLC	www.ecotrons.com	Page 124

# **EC**ITRON

Block Input: Start Position Offset (8Bytes)	Numeric	OTP writing start address offset, the unit is 8 bytes.
Select Trigger Type	Drop-down list	Block trigger type selection.
Data Length (Byte)	Numeric	The data length in byte.
Block Input: data		Data input
<b>Block Output:</b> st	boolean	Data write status, 0 stands for writing successful; Non- 0 value stands for writing unsuccessful.

## 4.15.5 Read Data by Address

This block enables users to have access to memory by address.

Read Data By Address Start Address:hex2dec('00018100') data Data Length(Byte):1

Read Data By Address



Source Block Parameters: Read Data By Address
Read Data By Address (mask) (link)
Parameters
Start Address:
hex2dec('00018100')
Data Length(Byte)
1
OK Cancel Help Apply

#### **Block Parameters**

Parameter Field	Value	Comments/Description
Start address	Numeric	Start address
Data Length (Byte)	Numeric	Data length
<b>Block Output:</b> Data		Data output

## 4.15.6 Read Data by Address (Input port)

This block is the same as the "Read Data by Address" block, except for the method of specifying the address is changed. For this block, the address is specified by block input signal.



Read Data By Address(Input port)



Parameters: Read Data By Address(Input port)	x
Read Data By Address (mask) (link)	
Parameters	
Data Length(Byte)	
OK Cancel Help Appl	y

#### **Block Parameters**

Parameter Field	Value	Comments/Description
Start address(input)	Numeric	Start address
Data Length(Byte)	Numeric	Data read length
Block Output: data		Data output

## 4.15.7 Read String Value

This block can translate strings to corresponding ASCII numeric arrays.

	Source Block Parameters: Read String Value
	Read String Value (mask) (link)
String:EcoCoderVCU data ►	Parameters
Read String Value	String Input:
	['EcoCoderVCU'
	OK Cancel Help Apply

#### **Block Parameters**

Parameter Field	Value	<b>Comments/Description</b>
Copyright ECOTRONS LLC	www.ecotrons.com	Page 127



String Input	String	The string to be parsed
Block Output:	Numoric	The numeric arrays of the
Data	Numeric	string ASCII.

## 4.15.8 EEPROM Emulation Definition

	Block Parameters: EEPROM Emulation Definition
	EEPROM Emulation Definition (mask) (link) EEPROM Emulation Definition
EEPROM Emulation Definition Enable Record Cache:off Record Cache Size(Record):2	Parameters  Enable Record Cache Record Cache Size(Record)
EEPROM Emulation Definition	2
	OK Cancel Help Apply

Parameter Field	Value	Comments/Description
Enable Record Cache	Check box	Enable logging of information, non-self- recording.
Record Cache Size (Record)	Numeric	The number of the cached record message.

## 4.15.9 Clear ALL EEPROM Emulation Record



# **EC** TRON

Parameter Field	Value	Comments/Description
Select Trigger Type	Drop-down list	Trigger type selection
Block Output: st	Numeric	Output state, refer to Appendix Table 1.

## 4.15.10 Clear One EEPROM Emulation Record



Parameter Field	Value	Comments/Description
Select Trigger Type	Drop-down list	Start address
<b>Block Input:</b> id		Record id
Block Output: st	Numeric	Output state, refer to Appendix Table 1

## 4.15.11 Read EEPROM Emulation Record



# ECITRON

#### EcoCoder Manual

Parameter Field	Value	Comments/Description
Data Length (Byte)	Numeric	Data length in byte
Block Input: 1. id 2. type		<ol> <li>The id of records to be read</li> <li>The type of records to be read, which inherits the data type of the previous block</li> </ol>
Block Output: 1. data 2. st		<ol> <li>Read Data</li> <li>Output state, refer to Appendix Table 1</li> </ol>

## 4.15.12 Write EEPROM Emulation Record



Parameter Field	Value	Comments/Description
Select Trigger Type	Drop-down	Trigger type selection
Data Length	Numeric	Data length
Block Input: 1. id 2. type 3. data		<ol> <li>Id of records to be written</li> <li>Type of records to be written</li> <li>Data written</li> </ol>
Block Output:	Numeric	Output state, refer to Appendix

Table 1

# 4.15.13 Read Signals from EEPROM Emulation Record

	EEPROM	1 V2.8.5 me : AllTypes	Type Val	ue: 0x00	0000011	Length : 2	7(bytes)				Boolean         Parameters           Boolean         Parameters
	Order	Signal Units  name	Start 1   (LSB)	Length  (bit)	Data  type	Byte Fa order	ctor Of	fset  	Multiplex  type	Multiplex value	U88 Select M File RecordTypeDefDemo_V1 Select Type JalBaseTypes
Record ID	11	Boolean	01	11	unsigned	intel	11	01	Standard  Standard	0	U16 Distance (ppc) Indexes (ppc)
	31	108  U16	161	81	signed  unsigned	intel	11	01	Standard  Standard	0	III6 Enable Custom Variable Name Rules
	51	I16  U32	401	16  32	signed  unsigned	intel  intel	11	01	Standard  Standard	0	U32 · "
	71 81	I32  Float32	88    120	32   32	signed  float32	intel  intel	11	01	Standard  Standard	0	132 Function packaging Parameterless
											Float32

Parameter Field	Value	Comments/Description
Select M file	Check box	Add M file to MATLAB path, specify the M file name, then click "OK" and "APPLY", finally double-click the module again to select the Message.
Select Type	Drop-down list	Record type selection
Show Signal Names	Check box	if enabled: the name of the signal will be displayed on the output signal line.
Signal Prefix		User-defined variable prefix on the signal line.
Block Input: Record ID		The record ID to be read and parsed
<b>Block Output:</b> st		Output state, refer Appendix Table 1 Note: The value of the signal after parsing record is the actual physical value

# 4.15.14 Write Signals to EEPROM Emulation Record

Enable Record ID Boolean	EEPROM Type Nai	1 V2.8.5 me: AllTypes	т	ype Valu	ie: 0x00	0000011	Length : 2	7(bytes)				Block Parameters: Write Signals to EEPROM Emulation Reco Write Signals to EEPROM Emulation Record (mask) (link)	d
U08	Order  	Signal Uni name	ts  	Start L (LSB)	ength  (bit)	Data  type	Byte Fa order	ctor Of: 	fset  	Multiplex  type	Multiplex value	Parameters	
108	1	Boolean		01	1	unsigned	intel	1	0	Standard	0		
116	2	U081	1	8	8	unsigned	intel	1	0	Standard	0	st Record TypeDerDemo_V1	
10	3	1081	1	16	8	signed	intel	1	0	Standard	0	Select Type AllBaseTypes	
16	4	U16	1	24	16	unsigned	intel	1	0	Standard	0		
	5	I16	1	40	16	signed	intel	1	0	Standard	0	Function packaging Parameterless	
J32	6	U32	1	56	32	unsigned	intel	1	0	Standard	0		
	7	I32	1	881	32	signed	intel	1	0	Standard	0		
I32 Float32	8	Float32		120	32	float32	intel	1	0	Standard	0	OK Cancel Help	/

Parameter Field	Value	Comments/Description			
Select M file	Check box	Add M file to MATLAB path, (users can change the default M file name) then click "OK" and "APPLY", finally double- click the module again to select the Message.			
Select Type	Drop-down list	Record type selection			
Block Input: 1. Enable 2. Record ID		<ol> <li>Write record enable control, it is recommended to control by edge triggering signal.</li> <li>The record ID to be written</li> </ol>			
Block Output: st		Output state, refer Appendix Table 1			

## 4.15.15 Program First Run Flag



## 4.16 Application Base Blocks

## 4.16.1 Rising Edge

This module is used to judge whether there is a rising edge trigger or not.



Parameter Field	Value	Comments/Description
Initial Condition is true	Check box	This is an initialize configuration option.
		2 422



	If checked, the default initialization value
	is 1.

## 4.16.2 Falling Edge

This module is used to judge whether there is a falling edge trigger or not.



Parameter Field	Value	Comments/Description
Initial Condition is true	Check box	This is an initialize configuration option. If checked, the default initialization value is 1.

## 4.16.3 Online Programming by SoftReset

This module is suitable for controllers that support the CCP protocol update procedure, with a CRO ID of 0x100, a DTO ID of 0x101, and a station address of 0x3502.



## 4.16.4 Online Programming by Entry UDS Programming

This module is suitable for controllers that support the CCP protocol update procedure, with a CRO ID of 0x100, a DTO ID of 0x101, and a station address of 0x3502.

	🔁 Block Parameters: Online Programming By Entry UDS Progra		
	Online Programming By Entry UDS Programming (mask)		
	Online Programming By Entry UDS Programming		
Online Programming By Entry UDS Programming CAN Channel:CAN_C	Parameters Select CAN Channel		
Online Programming By Entry UDS Programming	CAN_C •		
	OK Cancel Help Apply		

Parameter Field	Value	Comments/Description
Select CAN Channel	Drop-down list	CAN channel selection

## 4.17 Model Reference

Model reference in Matlab Simulink:



The 'Model' block in Simulink library 'Ports & Subsystems' could be used to include a submodel in a parent model. The blocks included in the 'Model' block are regarded as referenced models, and the model that includes the referenced models is named Parent Model. The referenced model can be used as an independent model to run simulation independently; it can also be used as the model reference and take part in the simulation in the parent model.

For more information about the Model Reference, please refer to:

https://www.mathworks.com/help/simulink/model-reference.html

Add the Simulink block 'Model' to the current Simulink model, and double click, the user will be able to add the referenced model in the popup window Block Parameters.



After the referenced model is successfully added, the inputs and outputs of the referenced model will appear on both sides of the 'Model' block. The user can connect the inputs and outputs on the 'Model' block to the parent model.



## 4.17.1 Configurations for Parent Models and Referenced Models

When building Simulink models with EcoCoder, the configurations of the referenced model(s) and the parent model must be kept the same, otherwise there will be errors during the code generation or simulation and the ongoing process will stop due to the error. There are two methods to keep the same configuration for referenced model(s) and parent model:

- 1. Use 'Configuration Reference'
  - Or

2. Copy the configuration of the parent model to the referenced model(s)

#### 4.17.2 Configuration Reference

In order to keep the same configuration for parent model and referenced model(s), the user can use 'Configuration Reference' function. In Model Explorer, by clicking 'Convert Configuration Reference', the configuration of the parent model will be converted to configuration reference.

Model Explorer	
File Edit View Tools Add Help	
R: □ / 4 6 %   = = = + 2 + 6 0 = + <i>f f</i>   111	
Search: by Name Vame:	
Model Hierarchy 🖉 🚾 🔂 Contents of: …f_V2/Configuration (only) Filter Contents	Configuration Parameters: Demo_Ref_V2/Configuration (Active)
Simulink Root       Column View:       Default       Show Details 10 object(s)         Base Vorkspace       BookTypace       BlockTyps         Model Vorkspace       Solver       BlockTyps         Model Vorkspace       Open       Ctrl+1         Eabedde       Copy       Copy         > Base Norkspace       Copy       Paste         Copy       Paste       Ctrl+V         Activate       Rename       Find Referenced Variables         Export.       Convert to Configuration Reference       Properties	Model Configuration       *         In the Configuration is a set of 'Configuration Components' that individually define specific settings for a particular model's execution (simulation and/or deployment (code generation). A siven model can have more than one Configuration is used for current operation (simulation or code seneration) on the model. Only one Configuration is active at a time.         Configuration       Simulation Same beach of the different Configuration is active at a time.         Configuration       Same beach of the different Configuration's active at a time.         Configuration       Same beach of the different Configuration's active at a time.         Configuration       Demos. Bef V2         Nume:       Configuration         Description:       Same defiguration
Contents Search Results Convert to Configuration Reference from the Active Configuration of the Model	Revert Help Apply

Then right click 'Reference', select 'Propagates to Referenced Models' in the menu to apply the configuration in the referenced model(s).



#### EcoCoder Manual

Model Explorer	a mart man start	a halings have not	
File Edit View Tools Add Help	1		
🔁 🛅   2 🖞 🗳 💥   🖽 📖 📲	🖉 🕶 🖿 📽 🕲 🕲 🖗 률 fx 🔳	🔺 🔡 🖶 R. 🖓 R.	
Search: by Name 👻	Name:	& Search	
Vodel Hierarchy     Image: Constraint of the second	Contents of: ···_Ref_V2/Reference (c Column View: Block Data Types  Sl Name BlockType OutDataTypeStr_OutNir Out Open Ctrl+I Cut Cut Cut Cut Cut Cut Cut Find Referenced Variables Propagate to Referenced Models Properties	anly) Filter Contents	Configuration Reference Desc.Ref_V2/Reference (Active) Configuration Reference A configuration Reference allows multiple models to use the same externally stored configuration set. In a data dictionary or the base workspace, a configuration reference can be used to select the configuration for multiple models without modifying the models. Referenced Configuration Name: Configuration Name: Configuration Configuration Description:
4 III +	Contents Search Resu	lts	
Propagate the Configuration Reference of th	he Model to its Referenced Models.		h.

Configuration Reference Propagation to Refer	renced Models		×		
- Description Propagate configuration reference ( Reference ) of top model ( Demo_Ref_V2 ) to each selected referenced model. The current configuration sets will be saved and can be restored after propagation.					
Propagation report Click the Propagate button to propagate the configuration reference.					
Total: 6 🗹 Converted: 0 🖉 Rest	ored: 0 🗹 Skipped: 0 👿	Failed: 0 Search model	name		
✓ Model (6/6)	Status	Message	Undo/Redo		
✓ actr	Initial		(5) (2)		
✓ esmc	Initial		(5) (2)		
V pwrtrn	Initial		(5) (2)		
▼ sens	Initial		(5) (c)		
🔽 thrml	Initial		(5) (c)		
VehDrv	Initial		(5) (c)		
	Propagate Pause(s)	Restore All	Help OK		

After the conversion of the configuration file, please save the configuration file in the Matlab Workspace, the default name is 'configuration'.

Please remember to load the saved configuration file to Matlab Workspace before opening the parent model.

For more information about Configuration Reference, please refer to: <u>https://www.mathworks.com/help/simulink/ug/more-about-configuration-</u> references.html?#responsive\_offcanvas

#### 4.17.3 Copy Parent Model Configuration File to Referenced Model

In Model Explorer, the configuration file for different models could by copied by right clicking the file. In this case, the user can also copy the configuration file of the parent model to the referenced model(s) and activate the configuration file in the referenced model by right clicking the configuration.

#### 4.17.4 EcoCoder Blocks in Model Reference

All blocks in the EcoCoder have been updated to be compatible with Model Reference. But in order to keep the generated code with good normative and readability, users need to pay special attention to some of the EcoCoder blocks. Specifically, all EcoCoder blocks with blue edges could be used in both parent model and referenced model(s); while the EcoCoder blocks with red edges will only be valid in the parent model (<u>They can be used</u> in the referenced models to pass the simulation, but will not be generated in the C code)

#### ADC

All the blocks in ADC can be used and will be valid in both parent model and referenced model(s).

#### Advanced Data Blocks

All the blocks in Advanced Data Blocks can be used in both parent model and referenced model(s).

However, for the EEPROM Emulation Definition block, only the definition in parent model is valid, and will be generated into C code. If this block is defined in the referenced model(s), the definition will not be generated into C code.

#### Application Base Blocks

# ECITRON

All the blocks in Application Base blocks can be used and will be valid in both parent model and referenced model(s).

#### **Calibration & Measurement**

In Calibration & Measurement, the Calibration Data Check can only be used in in the parent model, for all other blocks in Calibration & Measurement, they can be used in both parent model and referenced model(s). If these blocks are defined in the referenced model(s), the definitions will not be generated into C code.

#### CAN

All the blocks in CAN can be used in both parent model and referenced model(s).

The CAN channel Definition block and CAN Wake-up Frame Definition block are valid and will be generated into C code only when they are defined in the parent model.

#### ССР

In CCP, all blocks can only be used in the parent model and shall not be used in the referenced model(s).

#### Diagnostic Blocks

All blocks in Diagnostic Block can be used and will be valid in both parent model and referenced model(s).

#### Digital I/O

All the blocks in Digital I/O can be used in both parent model and referenced model(s).

However, the H-Bridge Definition block, IPWM interrupt Handler Definition block, PWM Definition block and PWM IO Frequency Range Definition blocks are valid and will be generated into C code only when they are defined in the parent model. If these blocks are defined in the referenced model(s), they will not be generated into C code.

#### Non-Volatile Memory Blocks

In the Non-Volatile Memory Blocks, NVM Definition block can only be used in the parent model and shall not be used in the referenced model(s).

And Fixed NVM Definition block is valid and will be generated into C code only when it is defined in the parent model, if Fixed NVM Definition block is defined in the referenced model(s), it will not be generated into C code (Unlike Non-Volatile Memory block, user can still keep the Fixed NVM Definition block in the referenced model(s)). The parameters of the Fixed NVM Definition in all models shall be kept the same.

# ECITRON

All other blocks in Non-Volatile Memory blocks can be used in both parent model and referenced model(s).

#### **Programming Blocks**

All blocks in the Programming Blocks can only be used in the parent model.

#### SCI

All the blocks in SCI can be used in both parent model and referenced model(s).

However, the SCI Definition block is valid and will be generated into C code only when they are defined in the parent model. If this block is defined in the referenced model(s), it will not be generated into C code.

#### System Management Blocks

All the blocks in System Management Blocks can be used in both parent model and referenced model(s).

However, the ECU Master Chip Wake-up Definition block, Stack Overflow Detection Definition block and Watchdog Definition block are valid and will be generated into C code only when they are defined in the parent model. If these blocks are defined in the referenced model(s), they will not be generated into C code.

#### Task Scheduler

All the blocks in Task Scheduler can be used and will be valid in both parent model and referenced model(s).

# **Chapter 5 CAN Theory of Ecotrons**

## 5.1 Introduction

<u>Controller Area Network</u> (CAN) nowadays is very widely used on the vehicle control system. Ecotrons VCUs provide multiple CAN channels (3-5 channels, depending on the specific VCU model) and enables the VCUs to communicate with multiple electronic control units on the vehicle.

## 5.2 CAN Implementation

To use Ecotrons VCUs, the user needs to convert DBC file into .m file and then use the .m file to define and initialize the CAN communications. The process is intuitive, user-friendly, and can give users more flexibility for CAN communication implementation.



## 5.2.1 Convert DBC to m File

User can convert DBC to .m file automatically using the software EcoCAN that can be found in EcoCAL. If you want to know more about EcoCAL, please refer to the manual *EcoCAL manual for EV*.



#### **Process:**

1. Open the DBC file to be converted in EcoCAN.

E EcoCAN v1.42								
Open Doc Elle Organize New folder Bounloads Recent Places EV_Demo.dbc 2017/3/27 13.47 DBC File 7 KB Filtures Subversion Videos EV_Demo.dbc On the file File name: EV_Demo.dbc File name: EV_Demo.dbc Open Cancel	EcoCAN v1.42				-			
	No. 1							
Open DDC File    Organize  New folder Ile  Organize  New folder Ile  Ile  Ile  Organize  New folder Ile								
Open DDC File   Organize New folder   Downloads Name   Date modified Type   Downloads Name   Date modified Type   Documents Documents   Music EV_Demo.dbc   Documents Subversion   Videos Videos     Ev_Demo.dbc Disk(C:)   Local Disk (C:) Local Disk (C:)   File name: EV_Demo.dbc   EV_Demo.dbc Dit*(*.dbc)     Computer     Computer     Computer     Decal Disk (C:)     File name: EV_Demo.dbc								
Organize • New folder   © Organize • New folder   © Downloads   © Recent Places   © Libraries   © Documents   Music   © Pictures   © Videos     Image: Computer   Local Disk (C:)   © Local Disk (E:)   • Ele name: EV_Demo.dbc   • Det(t.dbc)   • Concel	(	😑 Open DoC File					<u> </u>	×
Organize < New folder     Downloads     Name     Date modified     Type     Size     EV_Demo.dbc     2017/3/27 13:47        Documents     Music   Pictures   Subversion   Videos     Videos     File name:   EV_Demo.dbc     Pictures   Docal Disk (D:)     File name:   EV_Demo.dbc     Pictures   Open     Cancel		🕒 🗢 📕 🕨 Compute	r 🕨 Local Disk (C:) 🕨 Demo		• •;	Search Demo		٩
Downloads   Recent Places   Libraries   Documents   Music   Pictures   Subversion   Videos     Videos     File name:   EV_Demo.dbc     Date modified   Type   Size     OBC File   7 KB     7 KB     Pictures   Subversion     Videos     File name:   EV_Demo.dbc     Pictures   Open   Cancel		Organize 👻 New folde	a de la companya de l					0
Recent Places          EV_Demo.dbc       2017/3/27 13:47       DBC File       7 KB         Documents       Music       Pictures       Subversion         Videos       Videos       Videos       Pictures         Local Disk (C:)       Local Disk (C:)       Pictures       Pictures         Documents       File name:       EV_Demo.dbc       DBC file       7 KB		Downloads 🔦	Name	Date modified	Тур	e Size		
Ibraies         Documents         Music         Pictures         Subversion         Videos         Local Disk (C:)         Local Disk (E)         File name:         EV_Demo.dbc         EV_Demo.dbc		Recent Places	EV_Demo.dbc	2017/3/27 13:47	DBC	File	7 KB	
Documents         Music         Pictures         Subversion         Videos         Local Disk (C:)         Local Disk (D:)         Local Disk (E)         File name:         EV_Demo.dbc         PBU(".dbc)         Open ▼         Cancel		😭 Libraries		Υ.				
		Documents		$\langle \rangle$				
Subversion Videos Local Disk (C:) Local Disk (E:) File name: EV_Demo.dbc VDem(*.dbc) Cancel		Pictures		\				
Image: Second Disk (C:)         Image: Local Disk (C:)         Image: Local Disk (C:)         Image: Local Disk (C:)         Image: Local Disk (E:)         Image: File name: EV_Demo.dbc         Image: File name: EV_Demo.dbc         Image: Del(".dbc)         Image: Computer         Image: Computer <t< th=""><th></th><td>Subversion</td><td></td><td>·</td><td></td><td></td><td></td><td></td></t<>		Subversion		·				
Computer Local Disk (C:) Local Disk (D:) Local Disk (E:) File name: EV_Demo.dbc Copen Cancel		Videos						
Local Disk (C:)     Local Disk (D:)     Local Disk (E:)     File name: EV_Demo.dbc     Open ▼ Cancel		🖳 Computer						
Local Disk (D:)     Local Disk (E:)     File name: EV_Demo.dbc     Open ▼ Cancel		🚢 Local Disk (C:)			- \			
Gancel Disk (E) File name: EV_Demo.dbc ↓ Db (',dbc) ↓ Cancel		👝 Local Disk (D:)						
File name: EV_Demo.dbc		🧫 Local Disk (E:) 🖕						
Open  Cancel		File na	ime: EV Demo.dbc		•	Db.L(*.dbc)	-	-
			-		ſ	Open 🚽	Cancel	
					L			

4


2. After DBC file being loaded, the following window will pop-up.

≡ EcoCAN v1.42 - [ListView] ≔		
<u> </u>		
Network Nodes     HCU     HCU_COMMA     HCU_BMS_0     BMS_HCU_3     BMS_HCU_2     BMS_HCU_1     BMS_HCU_0     MCU_STATUS     MCU_STATUS     MCU_STATUS     MCU_STATUS     MCU_STATUS     MCU_STATUS	Address Comment HCU 0x0 BMS 0x0 MCU 0x0 4 5 2 3 1	
G:\to	vout_EcoTools\Setup Tools\EcoCAN V1.42\EV_Demo.dbc	

3. Click the indicated button and export the DBC file to m file, users can specify the saving path.



## 5.2.2 EcoCoder CAN Blocks

Please select 'Read CAN Message' or 'Send CAN Message' if fixed-point tool has not been installed in Matlab.

EcoCoder Blocks     ADC     Calibration & Measurement	EMS_ACTrq (12:12 uint16:Motorola Data EMS_DmdTrq (16:12:uint16:Motorola
CAN	Pack Signals To CAN Data Read CAN Message
CCP Diagnostic Blocks Digital I/O	Read Fixed-Point CAN Message Send CAN Data
SCI > Sensors Blocks	And
System Management Blocks Task Scheduler	Send CAN Message Send Fixed-Point CAN Message

#### 5.2.3 Select m file

This m file can help parse out signals in messages. Users need to save the .m file in the folder where your model is.

🕶 🕶 🖽 🕫 👘 🖓 G. 🕐 SOTTA	[ <sup>-03</sup>	process อกเรากุดตรายุเสตร			
Current Folder	۲	🖻 EV_Demo 🕨 🖻 ASW_Demo 🕨	1 nrocoss input signals 1 0 02 procoss hms input signals		
🗋 Name 🔺			Source Block Parameters: ecocoder_sfun_read_float		
🗏 👃 asw_libs	Ξ		S-Function (mask)		
🗏 👢 canMx_m	2				
🖄 Ev_Demo_M.m	⇒				
🖽 📕 libs	AE		Parameters		
📅 EV_Demo.dbc		ECO-EV CAN V1.0.9C	Select CAN Channel CAN_A		
± 📕 asw_m		ID : 0x00000211(standard) RTR			
■ L DI_E_C_m		Signa come  Un:			
Image: Mot_andEcoCAL_INI EV Domo mdl		rx CANABMS DC_CC	Ev_Demo_M		
MDL setup m		TX CANABMSAC_CC	Soloct Mossage BMS HCIL 0		
		rxCANAEMSEMS_CellLVoltNum	Select Message DHS_NCO_0		
		rxCANABMSBMS_CellHVolt   rxCANABMSBMS_CellHVoltNum	BMS_HCU_0		
		rxCANAEMSEMS_SOC	Show Data Available Port		
		rxCANAEMSEMS_Md			
		xxCANABMSBMS_VCU_0_Heatbeat	Snow Age Count Port		
			Show Signal Names		
			Signal Prefix		
			'rxCANABMS'		
Detaile					
50 uno					
			OK Cancel Help Apply		

## 5.2.4 Select Message

This step will let users pick the specific CAN messages that need to be parsed.

Source block Parameters, ecocoder_stun_read_noat
S-Function (mask)
Parameters
Select M Flie
Ev_Demo_M
Select Message BMS_HCU_0
BMS_HCU_0 HCU BMS_0
Show Data Av BMS_HCU_3
BMS_HCU_2 Show Age Cot BMS_HCU_1
Show Signal NBMS_HCU_0
Signal Prefix MCU_STATUS_4 MCU_STATUS_5
'rxCANABMS' MCU_STATUS_2
MCU_STATUS_3 Sample Time Ta MCU_STATUS_1
OK Cancel Help Apply

### 5.2.5 Select Sample Time

Source Block Parameters: ecocoder_sfun_read_float	X
S-Function (mask)	
Parameters	
Select CAN Channel CAN_A	•
Select M Flie	
Ev_Demo_M	
Select Message BMS_HCU_0	•
BMS_HCU_0	
Show Data Available Port	
Show Age Count Port	
Show Signal Names	
Signal Prefix	
'rxCANABMS'	
Sample Time Task MDef	•
Task_MDef	
Task_Inherit	
Task_ini	

#### Task\_Inherit:

If 'Task\_Inherit' is selected, the block will be executed every time when the subsystem that includes this block is executed.

#### Task\_ini:

The block will only be executed during the initialization process when VCU is powered on.

#### Task\_MDef:

The sample time will be decided according to the interval value in the .m file that is shown below. (This value comes from DBC file and is editable).

3		%Message Number:1
4	-	case 'HCU_COMMAND'
5	-	ECOCAN.HCU_COMMAND = struct;
6	-	ECOCAN. HCU_COMMAND . name = 'HCU_COMMAND';
7	-	ECOCAN.HCU_COMMAND.description = 'HCU_COMMAND';
8	-	ECOCAN. HCU_COMMAND. protocol = "ECOCAN";
9	-	ECOCAN.HCU_COMMAND.id = hex2dec('113');
0	-	ECOCAN.HCU_COMMAND.idext = 'SIANDARD';
1	-	ECOCAN.HCU_COMMAND.payload_size =8;
2	-	ECOCAN.HCU_COMMAND.interval =-1;
3		

Interval	(-t, -1)	(0,0.005)	[0.005,0.01)	[0.01,0.02)	[0.02,0.05)
Sample Time	Task_Inherit	Task_H1ms	Task_H5ms	Task_H10ms	Task_L20ms
Interval	[0.05,0.1)	[0.1,0.2)	[0.2,0.5)	[0.5,1)	[1,10)
Sample Time	Task_L50ms	Task_L100ms	Task_L200ms	Task_L500ms	Task_L1000ms

## **Chapter 6 Memory Management**

## 6.1 Introduction

When application software gets more complicated and larger, memory management will become an important aspect of VCU software development.

## 6.2 Storage device

Ecotrons VCU includes two types of storage device, Flash and RAM.

Flash is the memory which stores basic software, application software, constant, calibration and NVM variable data, the data in Flash will not be lost after powering off the VCU. Contents in Flash would be copied to RAM during VCU power-up process. NVM variable data is recommended to be saved to Flash before VCU power off. Calibration can be implemented 'on the fly', and calibration variable data can be burned back to Flash manually through EcoCAL, the calibration software developed by Ecotrons.

RAM (Random Access Memory) directly works with CPU by storing software needed data and code during VCU runtime. Different from Flash, the data in RAM would be lost when VCU powers down.

### 6.3 Data Storage

### 6.3.1 Calibration/Measurement Variable

Please refer to section 4.10.1 to 4.10.4 for definition, initialization, reading and writing calibration and measurement variables.

The only special part is writing of calibration variable. It is achieved through EcoCAL 'Program' or 'Download' function. EcoCAL is an advanced calibration tool developed by Ecotrons. 'Download' option can save calibration data to VCU RAM, while 'Program' option can write calibration data to VCU flash. 'Upload' can help upload the existing calibration data from VCU flash to PC.

## 6.3.2 Non-Volatile Variable

There are two types of non-volatile variables, NVM variables and Fixed NVM variables, for different application purposes. Please refer to section 4.8.1 to 4.8.9 for definition, reading and writing non-volatile variables.

Ecotrons non-volatile memory theory is described in Appendix A.

# Chapter 7 Custom Variable Type

There are two ways to define monitoring/calibration/NVM variables. One is to custom variable type, and the other is to use the definition block in the EcoCoder library.

The method in this chapter eliminates the need for software engineers to load multiple monitor/calibration/NVM variable blocks during simulation by simply defining the variables in "Base Workspace" and save them as m files.

EcoObj is a custom data storage definition package. It is an extension of the simulated signal object and the simulation parameter object. Define custom data objects and classes by using the ASAP2 standard. It generates the product code and the ASAP2 file (or a2l file). You can use EcoObj or MATLAB's "Model Explorer" to define types and variables in M files, the following sections describe the graphical definition method.

## 7.1 Customize Variable Types

1. Open "Model Explorer"

Demolect	CONTRACT OF A DESCRIPTION OF A DESCRIPTI	and the second here
He Edit Mew Display Disgram Simulation Analysis Code Jook Help		
5.86444880.80.80.00.00	10 D (Romal +) 🥥 + 🛗 +	

#### 2. Base Workspace > Add > Simulink Numeric Type

Model Explorer				
<u>File Edit View Tools</u>	Add	Help		
🛛 🖂 🖾 🖓 🖾 🖓		MATLAB Variable	Ctrl+M	<b>時品</b> 在
Search: by Name		Simulink <u>P</u> arameter	Ctrl+P	Seare
	τ	Simulit: Signal	Ctrl+5	<b>F</b>
Model Hierarchy		Simulink NumericType		ice (only)
A Simulink Root	<b>H</b>	Simulink AliasType		Show De
Base Workspace	=	Simulink <u>B</u> us		F
P a Demolest	<b>P</b>	Simulink Variant		
	٢	Configuration		
		Configuration Reference		am Files\EC
	6	Configuration for Concurrent Execution		am Files(EC
		Add <u>C</u> ustom		puble>
	祭	Event	Ctrl+E	
		Data	Ctrl+D	
	Æ	Input Trigger		
	fx	Function Call Output		
	۲	Stateflow Target	Ctrl+T	
	_			

3. Name the variable and set the properties through the window on the right

🖼 Model Explorer						
File Edit View Tools Add Help						
9. 🗀 🖁 🖧 🙀 🖽 🐘 🖿 🕲 🗉 🕸 🗜 🎾 📖 🕢 📲 🖓 리, 국 다						
Search: by Name	✓ Name:	🚳 Search				
Model Hierarchy 🖉 🚾	🔚 Contents of: …rkspace (only) - Fil	ter Contents	Simulink.NumericT	ype: DT_trq_S16		
Simulink Root     Base Workspace     Main DemoTest	Column View: Data Ob. • Show Datails Name Value EDT_trq_516	i_object(s)	Data type mode: Signedness: Word length: Slope: Sias:	Fixed-point: slope and bias scaling:  Signed  16  2'0  0		
			Lata type averite: Is alias - Code generation op Data scope: Auto Header file:	tions =		

#### 7.2 Add Variables to Workspace

Add "EcoObj.Signal" or "EcoObj.Parameter" to "Base Workspace" via "Model Explorer" as shown below.

1. Open "Model Explorer"



2. Base Workspace > Add > Add Custom...



3. Click "Add Custom", choose "EcoObj.Signal" or "EcoObj.Parameter".

Obj	
EcoObi.Parameter	
EcoObj.Parameter	
EcoObj.Signal	
AUTOSAR, Parameter AUTOSAR, Signal canlib, Parameter canlib, Signal SimulinkDemos, Parameter	Е
	EcoObi.Parameter EcoObj.Parameter EcoObj.Signal AUTOSAR.Parameter AUTOSAR.Signal canlib.Parameter canlib.Signal SimulinkDemos.Parameter SimulinkDemos.Signal

## 7.3 Customize Calibration Variables

Model Explorer - Select Object					
Object name(s) Demo trg_val					
Object class	EcoObj.Pa	rameter			
ок		Can	cel		

1. Choose "EcoObj.Parameter", name the variable then click "OK".

2. Set the properties through the window on the right. To define calibration variables, "Calibration(Custom)" must be chosen in "Storage Class".

) = = + + + = = = = = = = = = = = = = =			
GK Search			
Contents of: Base Workspace (only)	Pilter Contents	EcoObj.Parameter: Demo_trq_val	
Column View: Data Objects   Show Details	<u>1 object(s)</u> 🖓 🗸	Value: 10	
Diame Value DataType Min Max Dimension	s StorageClass Complexity Init	Data type: uint16	
Demo_trq_val 10 uint16 [] [] [1.1]	Calibration (C real	Dimensions: [1 1]	Complexit
		Minimum: []	Maximum
		Units:	
		Code generation options	
		Storage class: Calibration (Custom)	
		Custon attributes	
		Header file:	
		Alias:	
		Alignment: -1	
		Description:	

## 7.4 Customize measurement Variables

💮 Model Exp	lorer - Selec	t Object 🛛 💡 🗙
Object name(s)	: Demo_trq	
Object class:	EcoObj.Sigr	nal 🔻
OK	<	Cancel

1. Choose "EcoObj.Signal", name the variable then click "OK".

2. Set the properties through the window on the right. To define measurement variables "Measurement (Custom)" must be chosen in "Storage Class".

			~.						
51	Contents of: Ba	ise Workspace (or	nly)		F	iter Contents		EcoObj.Sign	ak Demo_trq
Col	umn View: Data C	Objects 🔹	Show Detai	s		2 object(s	) 🖓 -	Data type:	auto
	Name	Value	DataType	Min	Мах	Dimensions	StorageCl	Complexity:	auto
Del an	Demo_trq_val	10	uint16	0	[]	[1.1]	Calibratio	Dimensions:	-1
t	Demo_trq		auto	[]	[]	-1	Measuren	Sample time:	-1
								Minimum:	0
								Initial value:	
								Code gene	ration options
								Storage da	ss: Measurement (Custom)
								-Custom a	ttributes
								Header fi	le:

## 7.5 Customize NVM Variables

Model Expl	orer - Select Object	? ×
Object name(s):	Demo_trq_nvm	
Object class	EcoObj.Signal	•
ОК		Cancel

1. Choose "EcoObj.Signal", name the variable then click "OK".

2. Set the properties through the window on the right. To define NVM variables, "MeasurementNvm (Custom)" must be chosen in "Storage Class".

5.	Contents of: Ba	se Workspace (or	nly)		Ē	lter Contents		EcoO	)bj.Sign	al: Dcmo_trq_nvm
Col	umn View: Data C	bjects 🔻	Show Detail	<u>s</u>		3 object(s	) 🖓	Data	a type:	auto
	Name	Value	DataType	Min	Max	Dimensions	StorageCl	Com	plexity:	auto
[0] (14	Demo_trq_val	10	uint16	[]	[]	[1 1]	Calibratio	Dime	ensions:	-1
t	Demo_trq_nvm		auto	[]	[]	-1	Measuren	Sam	ple time:	-1
t	Demo_trq		auto	[]	[]	-1	Measurer			
								Minin	num:	
								Initia	al value;	
								Co	de gener	ation options
								Sto	xage da:	ss: MeasurementNVM (Custom)
								0	Custom a	tributes
								н	leader fi	e:



## 7.6 Save the Variables to M file

1. Base Workspace > Export...



2. Click "Export...", as shown below, save the file to "Demo\_data.m".

Export to File			x	J
G → + Computer + Local Disk (E:) + demo + 4 Search demo			Q	
Organize 💌 New folder	100	*	0	]
Name     Date modified		Туре		1
Image: Computer   Image: Local Disk (C:)   Image: Local Disk (D:)   Image: Local Disk (E:)   Image: Local Disk (C:)   Image: Local Disk (C:				
			+	
File name: Demo_data.m			-	
Save as type: MATLAB-files (*.m)			•	
Hide Folders     Save	G	ancel		

## 7.7 Load M file to Workspace

Drag "Demo\_data.m" file to the "Command Window".

🔥 MATLAB R	2012b					- 18 p.
HOME	PLOTS	APPS		SHORTCUTS		
New New Script 👻	Open	Import Data Wo	Save orkspace	🖶 New Variable	Analyze Code	s ▼ Libra
	FILE		VA	RIABLE	CODE	SIMUL
< 🔶 🔁 🍃	🖸 퉬 🕨 F: 🕨 test	DemoEV	Demo			
Current Fold	er 💿	Commar	nd Windo	w		
Demo_	data.m	fx; >>				

## 7.8 Model Example



## Chapter 8 Programming VCU with EcoFlash

EcoFlash is a user-friendly GUI for programming VCU. Please refer to EcoFlash manual for operation of the software. The below picture provides a quick glance of EcoFlash.

Settings Communication Mo CAN Settings Device Type Eco	ode CAN -	Flash Open File Flash Cancel	
Device Index 0 Channel No. 1 Baud Rate 100 ALM BuildChkSum	kbs ECU Options	How to flash the S19/Mot/Hex file? Step 1: Power on VCU. Step 2: Configure CAN settings to comply with hardware setting and software specification, then click <open device="">. Step 3: Load the S19/Mot/Hex file, click <open file=""> Step 4: Power off VCU and wait for 5 seconds. Step 5: Click <flash>, then quickly power on VCU.</flash></open></open>	•
Open Device Open Device	Close Device		~

## **Chapter 9 Measurement and Calibration with EcoCAL**

EcoCAL is dedicated for data measurement, calibration, logging and analysis. It is a professional calibration tool developed by Ecotrons. EcoCAL is based on CCP/XCP protocols and uses the CAN bus for communication between master-slave stations. It provides great convenience for VCU in-vehicle testing and prototype development.



# Chapter 10 Uninstall EcoCoder

### **10.1 Uninstall EcoCoder from MATLAB**

#### Note: You have to close all MATLAB applications before uninstalling.

1. Double-click 'EcoCoder Loader'.



2. Choose MATLAB version, and select 'Uninstall EcoCoder from selected MATLAB version', then click 'OK'.

EcoCoder Loader	×
Tools	
Matlabroot: D:\Program Files\MATLAB\R2012b	
Select Matlab: MATLAB 8.0 (R2012b) x64	•
EcoCoder_MPC	
C Install EcoCoder to selected MATLAB version	
C Install EcoCoder to all supported MATLAB version	
Uninstall EcoCoder from selected MATLAB version	
Uninstall EcoCoder from all supported MATLAB version	
ОК	

## 10.2 Uninstall EcoCoder from Windows System

For EcoCoder update, the user will need to uninstall older version EcoCoder from

Window before installing new version EcoCoder.

Note: You have to close all open MATLAB applications before uninstalling.

1. Click 'Start' and then click 'settings', follow steps in picture below

	- T Citck Settings		
ល	Home	Apps & features	
F	ind a setting	Installing apps	
Ар	ps	Choose where you can get apps from. Installing only apps from the Store helps protect your PC and keep it running smoothly.	ne
	Apps & features 2 Click Apps	Allow apps from anywhere $\sim$	
≣t	Default apps		
<u>m</u>	Offline maps	Apps & features	
ſt.	Apps for websites	Manage optional features Manage app execution aliases	
	Video playback	Search, sort, and filter by drive. If you would like to uninstall or move an app, select it from the list.	
Ť	Startup 3 Put keyword to Search box	EcoCoder ×	
		Sort by: Name $\checkmark$ Filter by: All drives $\vee$	
		EcoCoder 4 Single click 75.1 M Ecotrons 6/12/201	<b>B</b> 8
		EcoCoder 17.8 M	B

## **Chapter 11 FAQs**

Q1. The m file exported from DBC by 'EcoCAN' can't be used

A1. The name of the m file must match to the C Language variable naming requirement.

And it can't be the name of the existing model or m file.

Q2. Model created by 'EcoCoder\_Prj', emulation or code generation error

- Check if your MATLAB has Fixed-Point Tool license. If not, the use of fixed-point blocks will trigger errors.
- 2. Make sure all support files are added to path.
- 3. Check whether necessary MATLAB components are installed.

Q3. 'CAN' module is blank after being configured

Please check whether the CAN definition .m file is added to Path.

Q4. EcoCoder Loader Pop-up error



You may have to register the 'comdlg32.ocx' to windows.

Q5. How to update application model to be compatible with updated EcoCoder

a) EcoCoder Target Definition

The model must include the EcoCoder Target Definition



b) Model configuration module

'ECU\_Setting' module is divided into independent configuration blocks. Please add CAN,

OPWM, CCP and other configuration blocks if needed.



c) Resolve some disabled modules

If CAN library blocks used in the model are disabled, you need to firstly resolve all blocks and save the model before installing new EcoCoder, otherwise the original model will be stuck when using new EcoCoder.

1) Right-click on the disable block, and select Library link->Resolve Link



#### EcoCoder Manual

ECO-EV CAN V1.1.7 	BMS_H h :8(byte	Comment Through Comment Out Delete	Ctrl+Shift+Y Ctrl+Shift+X Del	Message Available(bool) Message Count(uint32) rxCANABMSDC_CC	→ BMS_fig0Availabi value:0 type:bockean → trXBMS_ont0 value:0 typeuint32 → trXBANS_ont0_rnBM value:0 typeuint32
Signal Units  Stan name    (LSI	art Lenç 5B)  (bi	Find Referenced <u>V</u> ariables <u>C</u> reate Subsystem from Selection	Ctrl+G	tip <b>leDA</b> NABMSAC_CC value CANABMSBMS_CeIILVoit	Value:0 type:boolean Value:0 type:boolean VABMSBMS_Cell
rxCANABMSD_CCC    rxCANABMSAC_CC    rxCANABMSAC_CC    rxCANABMSBMS_CELLIVOLL  V  ! rxCANABMSBMS_CELLIVOLL  V  : rxCANABMSBMS_CELLIVOLL  V  : rxCANABMSBMS_CELLIVOLNUM    rxCANABMSBMS_SCC] !	3  2  56  40  32  16  8	Test <u>H</u> arness <u>F</u> ormat Botate & Flip <u>A</u> rrange	> > >	0 IABMSBRIS_CellLVoltNum 0 CANABINJSBINS_CellHVolt 0 ABMSBNJS_CellHVoltNum	value:0 type:single BMSBMS_Cell.v value:0 type:uint8 Value:0 type:single BMSBMS_Cell value:0 type:single DMSBMS_Cell mCA
rxCANABMSBMS_SysFltLv1    rxCANABMSBMS_Md    rxCANABMSBMS_VCU_0_Heatbeat	51 41 01	<u>M</u> ask Library Link	•	0 nxCANABMSBMS_SOC <u>G</u> o To Library	Block Ctrl+L
		Signals & Ports Reguirements Traceability Coverage	•	Break Link Resolve Link View Change	5
		Model Advisor	,		value:0 type:uint8

#### 2) Restore all disable linked blocks.

Description This model contains linked blocks that are disabled in the following table. To see all of the disabled lin		
This model contains linked blocks that are disabled in the following table. To see all of the disabled lin		
	or edited. By defa ks, select 'Show all	ult, only the edited links are shown disabled links'.
To push changes to the library, in the Action column select 'Restore'.	n, select 'Push'. To r	restore the block from the library,
To push/restore the whole hierarchy use 'Hierarchy' 'Individual' mode.	mode and to push	n/restore individual links use
Options	Push/Restore N	lode
Show all disabled links	Hierarchy	Individual
Edited links		
Linked block Action		Library (version)
▲ ecocoder_sfun_read_float < Restore	-	EcoCanServLibs (1.173)
\Lambda ecocoder_sfun_read_float 🛛 < Restore		EcoCanServLibs (1.173)
Paths for select ed link Path in model : Test EV2106801 CAN/ASW Demo signals/ecocoder sfun read float Library source : CoCanServLibs/CAN Float/ecocod	1/01 process input ler sfun read float	signals/03 process bms input
Push all Restore all Clear all	ОК	Cancel Help Apply

Q6. Is there a way to get rid of popping up folder of generated file?

## **EC**ITRON

Yes. Please go through these steps, Configuration Parameters -> Code Generation -> EcoCoder Debug Options -> Compiled Action. Then you can configure.

Compiled Action includes: No Prompt, OpenTarget, PopupBox.

No Prompt: There is no any prompt when it finishes generating file.

OpenTarget: It will open folder which has generated files.

PopupBox: 'Software has been compiled successfully!' will pop up when it finishes generating files.

Select:	Pariet Files Onlines Descended	^
Solver	Project Files Options Reserved	
Data Import/Export	Compiled Action No Prompt	
Optimization PopunBox		
Diag U.J. OpenTarget		
Mrd No Prompt		
Simulation Target		E
Code Generation		
Report		
Comments		
Symbols		
Custom Code		
Debug		
Interface		
Code Style		
Templates		
Code Placement		
Data Type Replacement		
Memory Sections		
EcoCoder Target Options		
EcoCoder Debug Options		
besign vermer		
		-
		, , , , , , , , , , , , , , , , , , ,
		ancel <u>H</u> elp <u>Apply</u>

Q7. Is there a way to access project file in C code?

Yes, go same steps as Q6 then configure box of 'Project Files Options'.

Project Files Options includes Reserved, Removed.

Reserved: 'XX\_CWprj' will be reserved when it finishes generating file, so user can access C code from 'XX\_CWprj ' file.

Removed: 'XX\_CWprj' will be removed when it finishes generating files.



#### EcoCoder Manual

Fologt		
Solver       Project Files Op         Data Import/Export       Optimization         Diagnostics       Hardware Implementation         Model Referencing       Simulation Target         Code Generation       Report         Comments       Symbols         Custom Code       Debug         Interface       Verification         Code Style       Templates         Code Placement       Memory Sections         EcoCoder Debug Options       HDL Code Generation	tions Removed Reserved	E

## **Appendix A- Nonvolatile Variables Theory**

This is a description of assigning variable values to non-volatile memory and fixed non-volatile memory locations, how to change such values and notice for operation.

### Non-volatile Variables

The NVM variables are stored in flash, which can maintain information even the VCU is powered off (unlike RAM, which would lose data after the VCU power off).

NVM data values are read from flash and written into RAM when VCU is powered on. The RAM variables can be read/written as many times as needed since RAM has high Program/Erase cycle. When key off signal is detected, power off logic would trigger the process of storing NVM variables from RAM to flash. An example power down block, which includes this NVM variable storage logic, *Power Management Example*, is provided in EcoCoder library.

**Note**: The power input to VCU BATT is required to be uninterrupted to make sure the process of storing of NVM variables value is safe. If power is lost unexpectedly while application is running, the value of the NVM variables on the next key-on will be the same value as what had been saved into flash during the last controlled shutdown. If power is lost unexpectedly during the controlled shutdown procedure (when the process of saving NVM variables into flash is supposed to be happening), all NVM variables will revert to their default values (defined in the application software).

### Fixed Non-volatile Variables

The Fixed NVM variables will be kept the same even the VCU is programmed (unless it is required to be changed by user configuration), so critical data such as odometer data will not be lost even the VCU software update is performed. The fixed NVM variables are stored in specific space of flash and arranged in the order defined block, which means specific addresses in flash are reserved for specific variables.

If new variables need to be added to fixed NVM space, it is necessary to re-initialize by calling definition block.

## **Battery Input**

As mentioned previously, power supply has to be maintained at least for a short period after key-off, in order for the VCU to execute the shutdown process.

The shutdown process implemented by block 'Power Management Example' includes stopping the application and saving NVM variables that have been temporarily stored in RAM to flash, during the power-off delay, after key-off. This is the recommended way to save nonvolatile variables to flash. If the frequency of calling 'Store All NVM Data' block is too high, errors might occur.

#### Table 1.

Status Value	Supported MinGW version
0	Successful operation
1	Insufficient space, available active area block is less than set active block
2	Flash operation error
3	Block operation error
4	Block detection error
5	Not enough writing space
6	Need to erase
7	Abnormal block status
8	Parameters error
9	Record not found
10	Record type not match
11	Record deleted
12	Record replication succeeded
13	Writing a record
14	Executing a swap operation
15	Records need to be written to new activity area