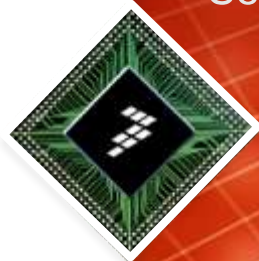




Hands-on Workshop: Designing Applications with the S12 MagniV Family

AMF-AUT-T0665

Derrick Klotz
Regional FAE
Senior Member of Technical Staff

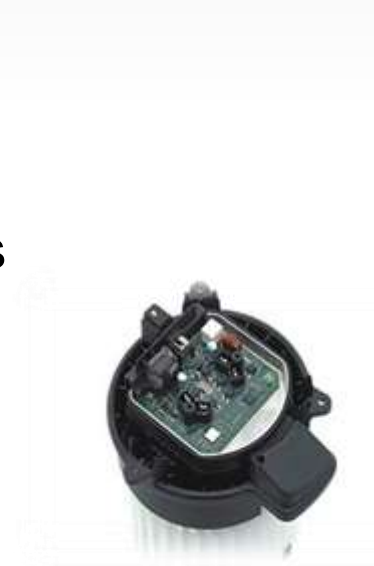


November 2013

Freescale, the Freescale logo, AllWin, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Wire, the Energy Efficient Solutions logo, iMote, iMote2GT, PGG, PowerQUICC, Processor Expert, QoIQ, QoIQv2, SafeAssure, the SafeAssure logo, StarCore, Symphony and Viterbi are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. AirBot, BeeBee, BeeStack, Coherent, Flexio, LayerScope, MagniV, M6C, Platform in a Package, QoIQ Converge, QUICC Engine, Ready Plug, SMARTMOS, Tower, TurboLink, Vybrid and Vybrid are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2013 Freescale Semiconductor, Inc.

Automotive Market Trends – Sensors and Actuators

- **Reduction of power consumption**
 - LIN to reduce cables weight
 - Smarter motor control techniques
 - Smarter and more sensors
- **Reduction of physical size**
 - Electro-mechanical integration
 - High temperature $>125^{\circ}\text{C T}_{\text{a}}$
- **High Growth**
 - Driven by affordability of LIN
 - Driven by comfort and convenience features

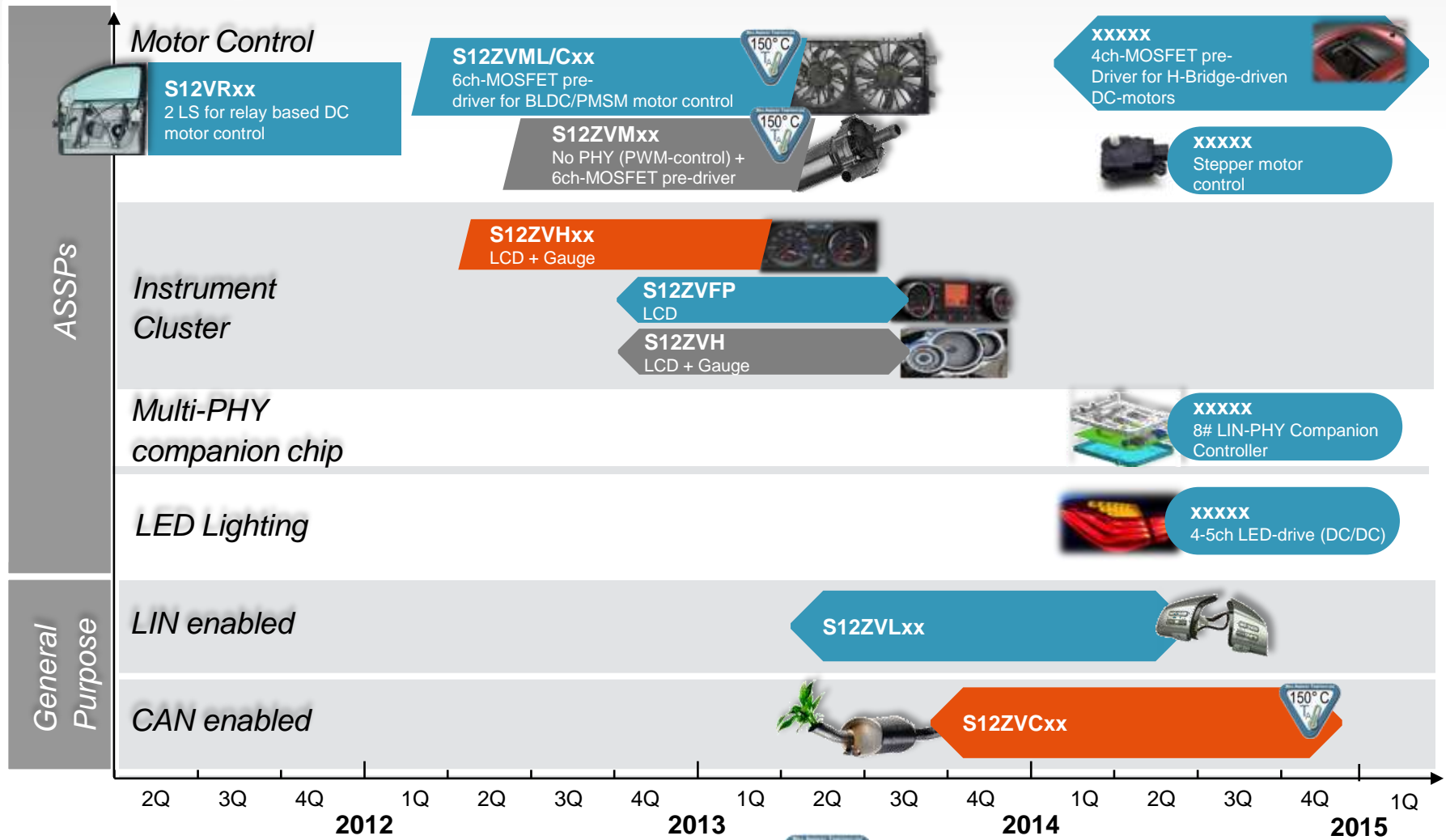


MagniV building blocks

High-Voltage Components		Digital Components		MCU Core and Memories	5V Analogue Components	Packaging	
VREG for tot. supply: •70mA w/o ext comp. or •170mA with ext. ballast		MS-CAN	SCI	S12- or S12Z-CPU 25/32/50MHz bus	ADC 10-12Bit resolution 1-2 S/H-units Up to 16ch total	LQFP: 32/48/64/100/144-pin	
LIN-PHY	CAN-PHY	SPI	IIC	Flash (ECC) 8kB – 128kB	Temp Sense	LQFP-EP: 48/64-pin	
V-BAT SENSE	V-SUP SENSE	GPIO	PGPIO 20mA	NGPIO 25mA	EEPROM (ECC) 128B – 4kB		
HVI (12V-input with ADC)		BDM/BDC		RAM (ECC) 512B - 8kB	Current Sense (2 x Op-Amp)	QFN: 32-pin (5x5mm)	
LS-drivers	HS-drivers	Key Wakeups		Win Wdog	Pierce Osc.		
Charge Pump		RTC			RCosc. +/-1.3%		
4-6ch MOSFET Predriver (50-100nC)		Timer 16Bit (25-64MHz)	PWM 8/16Bit (25-50MHz)		PLL		
		Motorcontrol PWM With Fault protection					
		Prog. Trigger Unit					
		Sound Generator					
		Segment LCD (4x40)					
		Stepper Motor Driver with SSD					



S12 MagniV Roadmap



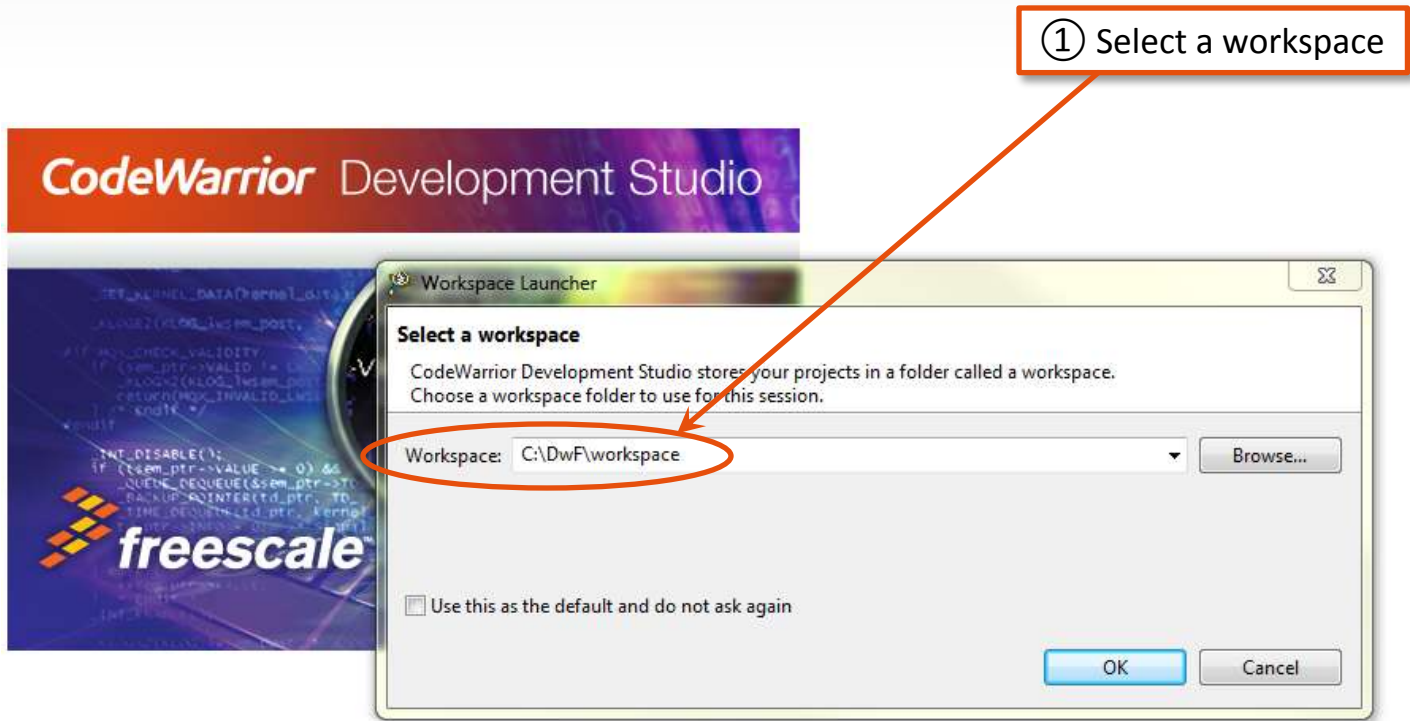
■ With LIN-PHY ■ No comms (PWM-control)
■ With CAN-PHY High temp (Grade 0)

freescale.com/MagniV

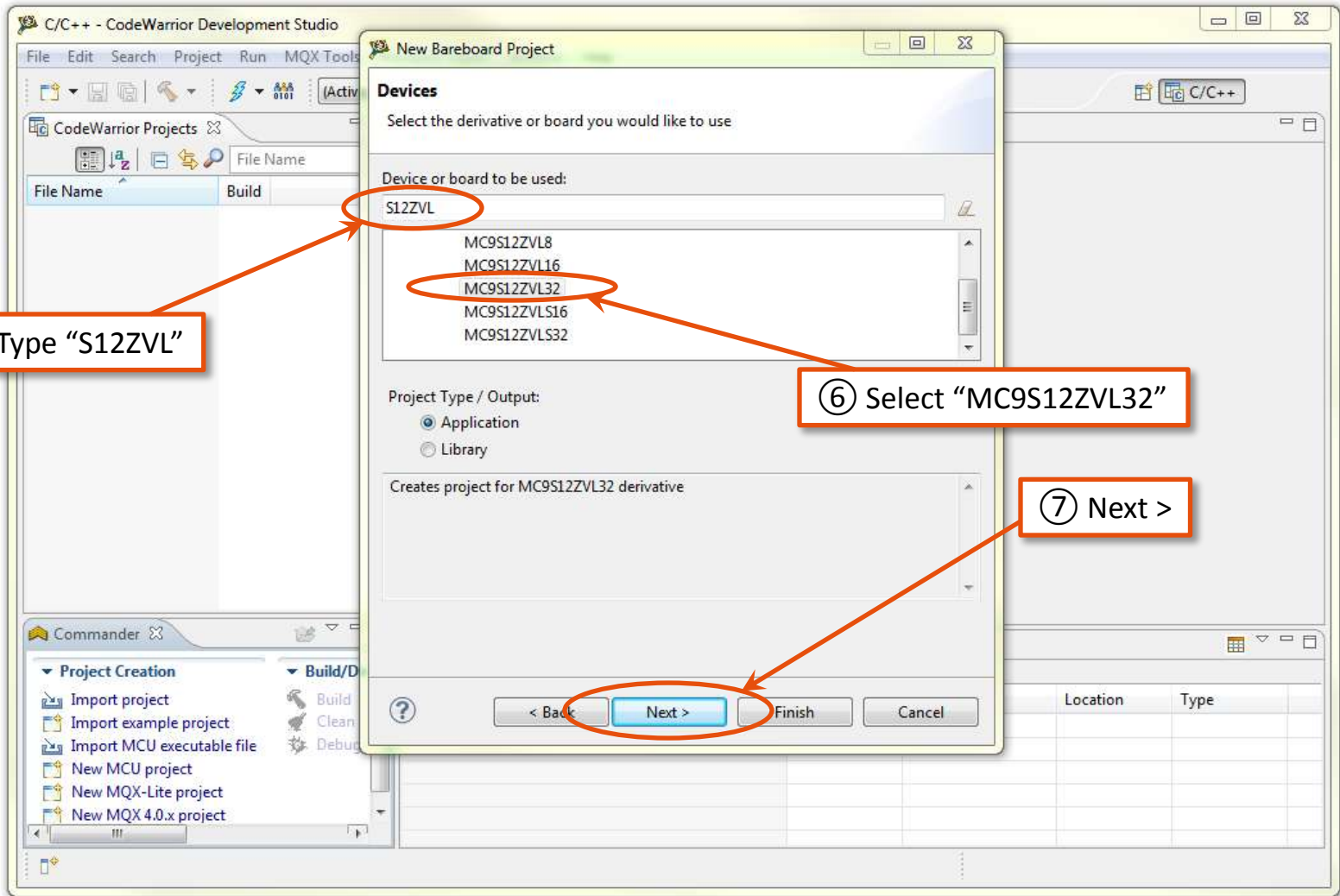


Freescale, the Freescale logo, AN1180, D.S., CodeTalker, CodeWarrior, CoreLink, CorePower, CoreSite, the Energy Efficient Solutions logo, iMESH, iNFI01, iP60, PowerQUICC, Processor Expert, QorIQ, Qorima, QorLabs, the QorLabs logo, StarCore, Symphony and Vorti are trademarks of Freescale Semiconductor, Inc. Reg. U.S. Pat. & Tm. Off. Artix, BeMicro, BestStack, CanFoc, Focite, LayerWatch, MagniV, MIC, Platform in a Package, QorIQ Developer, QorIQ Engine, iReady Play, SMARTMOS, Tivity, TurboLink, Hybrid and iSystem are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2013 Freescale Semiconductor, Inc.

Launch CodeWarrior 10.4



Select the Target Device



Component Inspector - Cpu

The screenshot shows the CodeWarrior Development Studio interface. The 'Component Inspector - Cpu' window is open, displaying a table of properties. The 'Expert' tab is selected in the top right corner. The 'Clock settings' section is expanded, showing various clock configuration options.

Select "Expert"

Expand "Clock settings"

Name	Value	Details
Component name	Cpu	
Clock generator	CPMU	CPMU
CPU type	MC9S12ZVL32MLF	
▲ Clock settings		
▲ Internal clock		
Internal oscillator frequency	1000.0	1000 kHz
▶ External clock	Disabled	
▲ Output clock		
▶ ECLK clock	Disabled	
▶ Low-power modes settings		
▶ Internal resource mapping		
▶ Operating mode settings		
▶ Internal peripherals		
▶ CPU interrupts/resets		
▶ Interrupts initialization		
▲ Enabled speed modes		
▲ High speed mode	Enabled	
High speed clock	Internal Clock	12.5 MHz
Internal bus clock	6.25	6.25 MHz
▲ PLL clock	Enabled	
Reference divider	Auto select	1
PLL multiplication factor	Auto select	25
PLL post divider	Auto select	4
VCO clock frequency [kHz]	50	
Frequency Modulation	FM off	
▶ Low speed mode	Disabled	
▶ Slow speed mode	Disabled	



Investigate Processor Expert Generated Code

The screenshot shows the CodeWarrior Development Studio interface. The 'CodeWarrior Projects' pane on the left shows a project tree where the 'Generated_Code' folder is circled in red. An arrow points from this folder to a callout box containing the text 'Expand "Generated Code"'. The 'Component Inspector - Cpu' pane on the right shows the 'Expert' tab with a table of properties for the 'Cpu' component.

Name	Value	Details
Component name	Cpu	
Clock generator	CPMU	CPMU
CPU type	MC9S12ZVL32MLF	
Clock settings		
Internal clock		
Internal oscillator freq	1000.0	1000 kHz
External clock	Disabled	
Output clock		

The 'Commander' pane at the bottom left shows project creation options. The 'Problems' and 'Console' panes at the bottom right show 0 items.

Investigate Assembly Language Code

The screenshot shows the CodeWarrior Development Studio interface. The title bar reads "C/C++ - Lab/FLASH/Generated_Code/Cpu_c1160692580758754539.lst - CodeWarrior Development Studio". The menu bar includes "File", "Edit", "Search", "Project", "Run", "MQX Tools", "ProcessorExpert", "Window", and "Help". The toolbar contains various icons for file operations and editing. The active window displays assembly code for "Cpu.c". The code includes comments such as "/* Disable protection of clock configuration registers */", "/* Enable the PLL to allow write to divider registers */", "/* Set the post divider register */", "/* Set the multiplier register */", "/* Set the PLL frequency modulation */", and "/* Set the multiplier register */".

Annotations on the screenshot:

- A red circle highlights the "Cpu.c" tab in the editor's tab bar.
- A red circle highlights the "Cpu_c1160692580758754539.lst" tab in the editor's tab bar.
- A red arrow points from the "Cpu_c1160692580758754539.lst" tab to a callout box that says "Close 'Cpu_cxxxx.lst' tab".
- A red arrow points from the "Cpu.c" tab to a callout box that says "Double-Left-Mouse-Click on 'Cpu.c' tab".



Agenda

- Automotive Body Electronics
- MagniV Solutions and Roadmap
- Lab 1 _____
 - Getting Started with CodeWarrior 10.4
- **S12Z Enhanced CPU Core**
- S12ZVM for BLDC Motor Control
- S12ZVL for General Purpose LIN
- S12ZVC for General Purpose CAN
- Lab 2 _____
 - Developing a Project for the S12ZVL



freescale.com/automotive

S12Z vs. S12X – Features

Attribute	S12Z	S12XE
Architecture	Harvard	Von Neumann
Address space	16MByte Linear	64KByte Linear (up to 4MByte Paged)
Data Bus Width	32-Bit RAM & Flash, 16-Bit I/O	16-Bit
ALU Width	32-Bit	16-Bit
Data Registers	2 x 8-Bit, 4 x 16-Bit, 2 x 32-Bit	2 x 8-Bit (can be used as a single 16-Bit accumulator)
Pointers	2 x 24-Bit	2 x 16-Bit

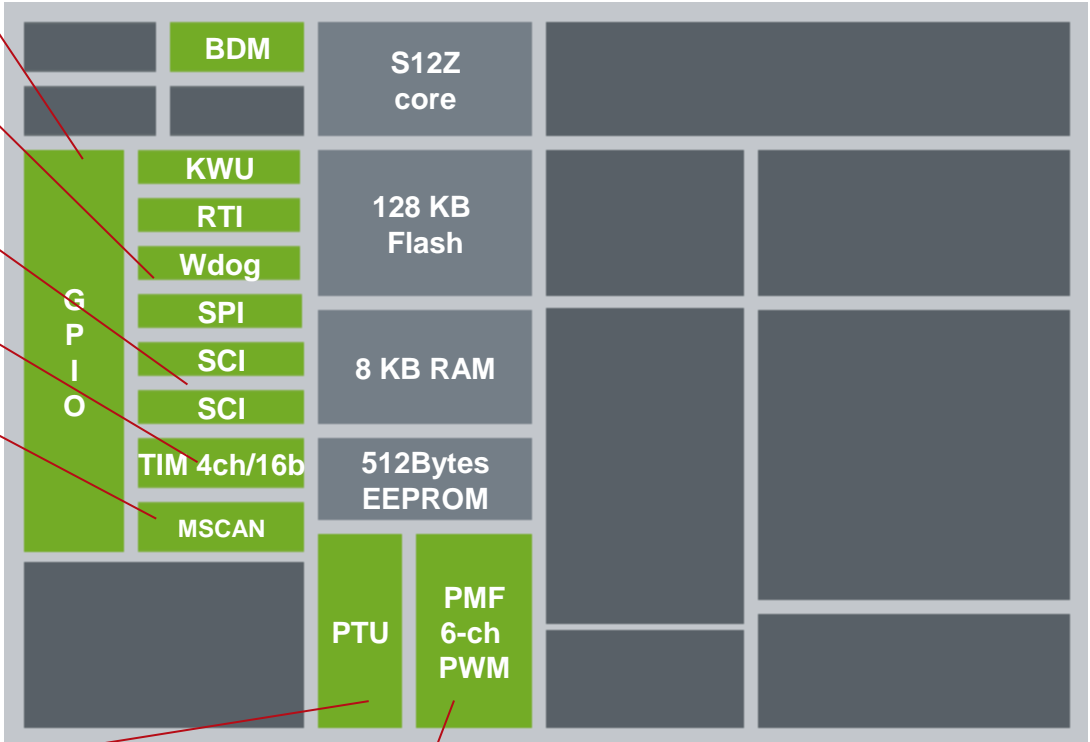
S12Z vs. S12X – Performance

Attribute	S12Z		S12XE	
Bit Shifter	32-bit multi-bit	1 cycle	16-bit single-bit	2cycles
Multiplier	32*32	2.5cycles	16*16	1cycle
	16*16	1cycle		
Divider	32 = 32/32	18.5cycles	32 = 32/16	11cycles
MAC	32 += 32*32	3.5cycles	32 += 16*16	13cycles
Fractional math support	Yes		No	
Bus speed	50MHz		50MHz	

- CPU operates at up to 100MHz, bus at 50 MHz
 - RAM bus can load and store w/ 100MHz
 - I/O buses @50MHz to reduce power consumption and die area

Overview of S12ZVM Feature Set

- Sensor Supply**
 - 20mA I/O for sensor
- Window Watchdog**
 - Independent RC osc
- 2 UARTs**
 - one linked to LIN Phy
 - 2nd as test interface
- Multiple timers**
 - IOC/periodic wakeup
- CAN Option**
 - CAN controller

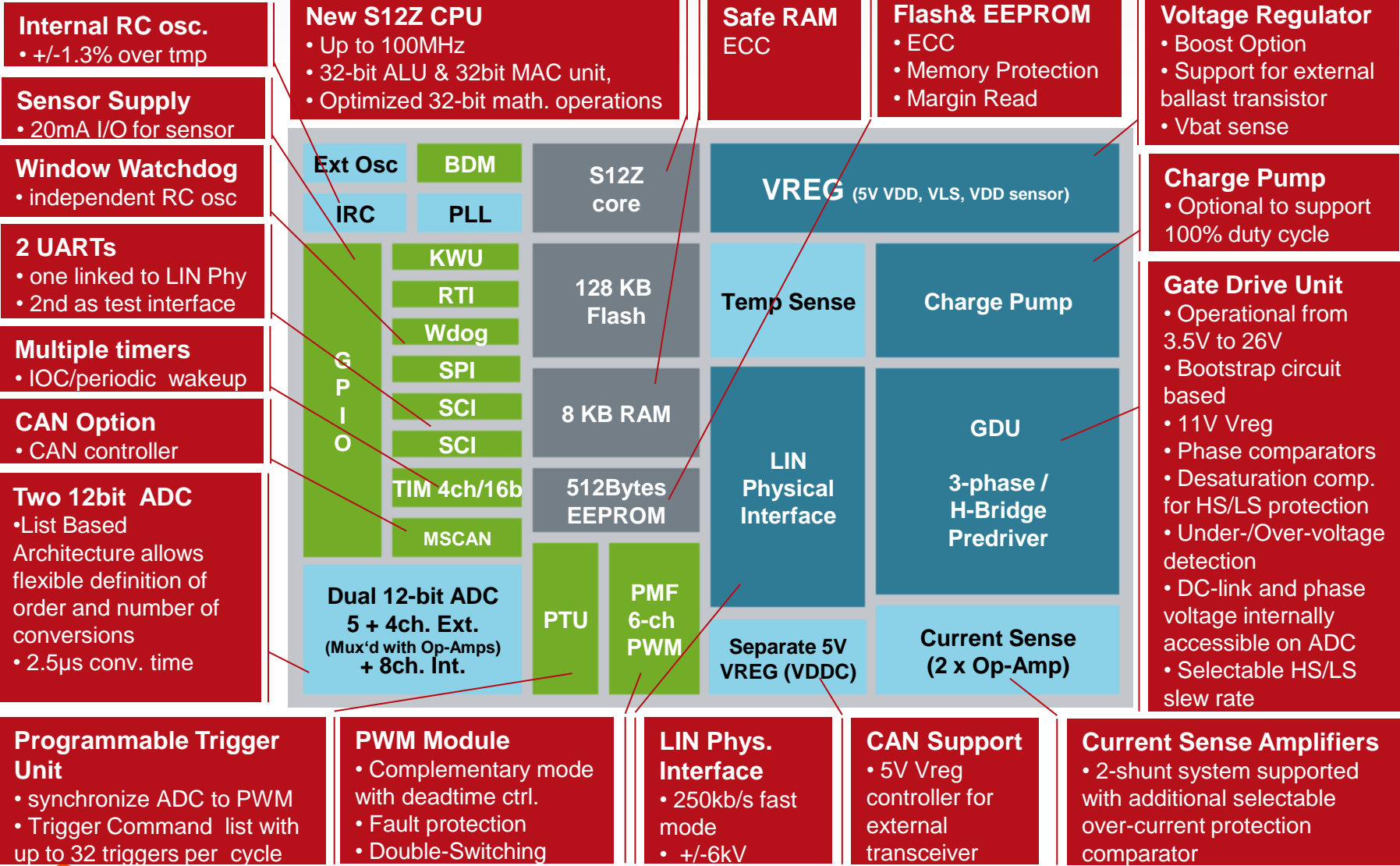


- Programmable Trigger Unit**
 - synchronize ADC to PWM
 - Trigger Command list with up to 32 triggers per cycle

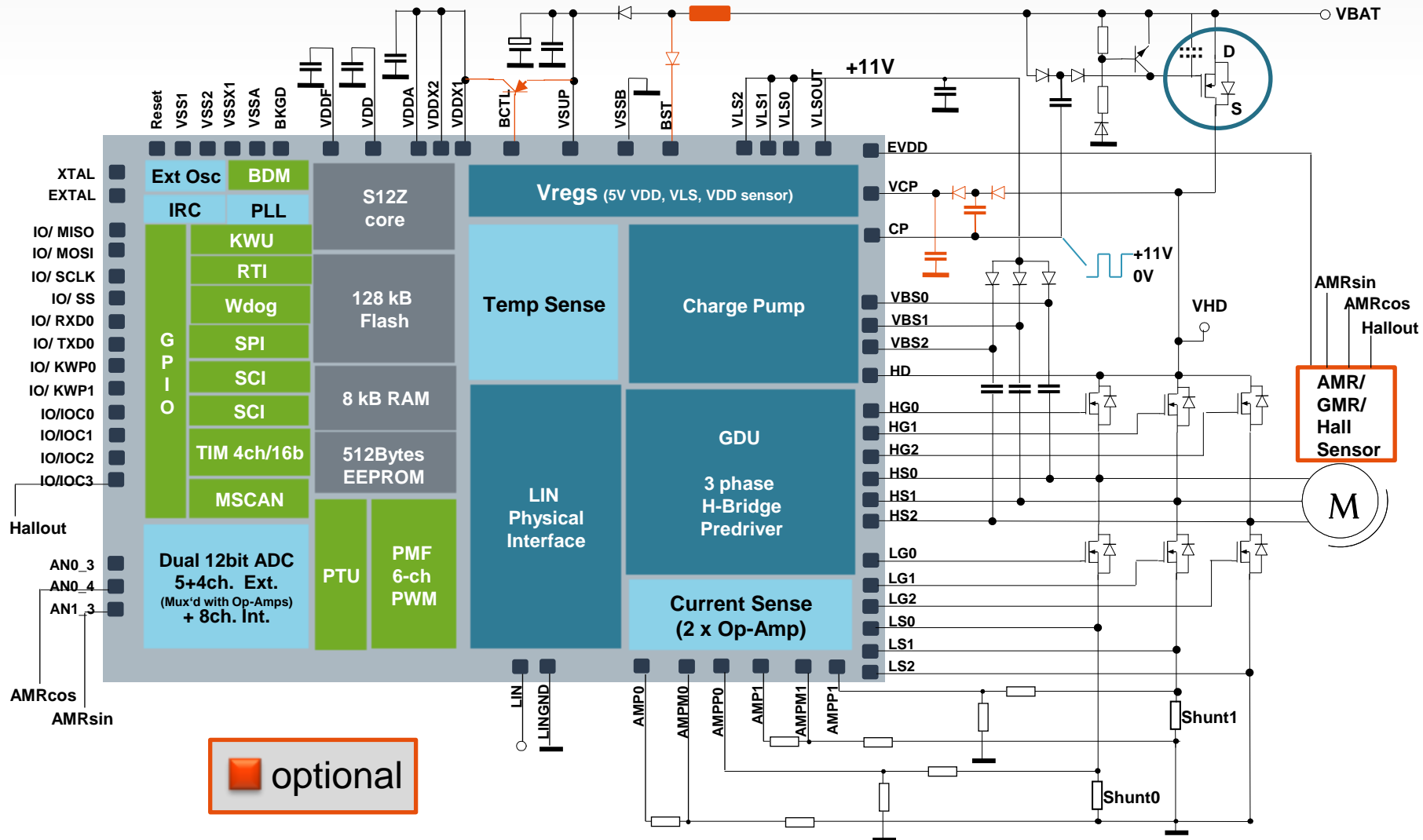
- PWM Module**
 - Complementary mode with deadtime ctrl.
 - Fault protection
 - Double-Switching



Overview of S12ZVM Feature Set



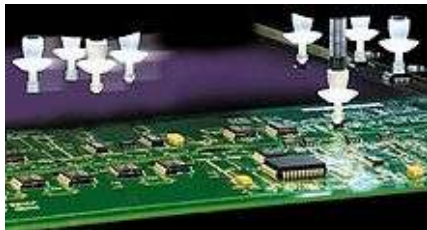
S12ZVML Application Schematic



Manufacturing Cost Benefits

Multiple effects of eliminating ICs or passive components:

- Material : PCB space, Cu tracks, test points, soldering
- Pick & place
- Testing at board level
- Inventory management



SMTA (Surface Mount Technology Association):

- NMACPCs (Non-Material Assembly Cost Per Component) range from \$0.01 to \$0.15 per component assembled
- NMACPI/O (Non-Material Assembly Cost Per Input/Output) range from \$0.005 to \$0.01 per I/O assembled.



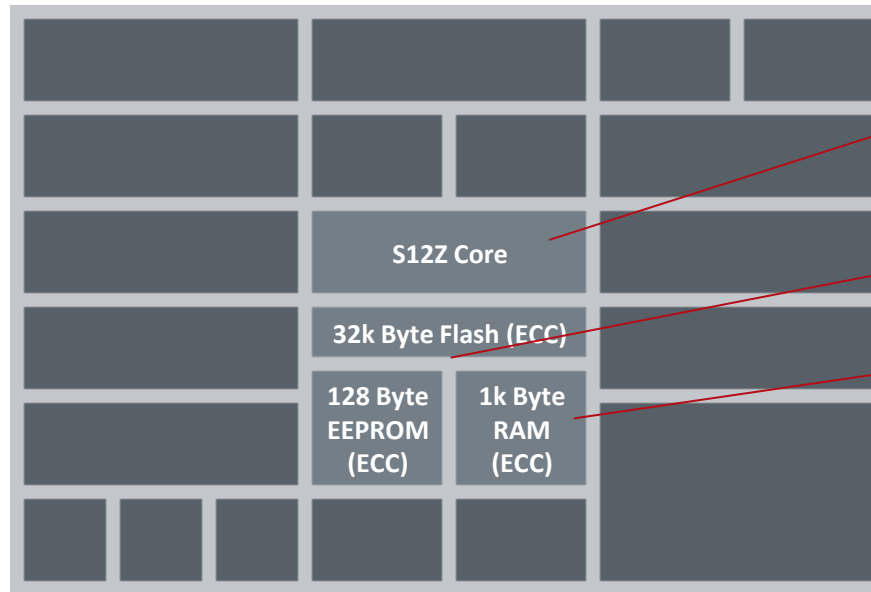
Agenda

- Automotive Body Electronics
- MagniV Solutions and Roadmap
- Lab 1 _____
 - Getting Started with CodeWarrior 10.4
- S12Z Enhanced CPU Core
- S12ZVM for BLDC Motor Control
- **S12ZVL for General Purpose LIN**
- **S12ZVC for General Purpose CAN**
- Lab 2 _____
 - Developing a Project for the S12ZVL



freescale.com/automotive

Overview of S12ZVL Feature Set



New S12Z CPU

- up to 50MHz
- 32-bit ALU & 32-bit MAC unit
- Optimized for 32-bit math operations

Flash & EEPROM

- with ECC
- Memory Protection
- Margin Read

Safe RAM

- with ECC

Overview of S12ZVL Feature Set

Internal RC oscillator
• ±1.3% over temp

		Pierce Oscillator		Temp Sense	10-bit ADC
SCI	RC osc ±1.3%	PLL	16-bit Timer 6-ch + 2-ch		
SCI	S12Z Core		PWM 8-ch, 8-bit (or 4-ch, 16-bit)		
SPI	32k Byte Flash (ECC)		BDM		
I2C	128 Byte EEPROM (ECC)	1k Byte RAM (ECC)			
GPIO	EVDD × 1	NGPIO × 3			

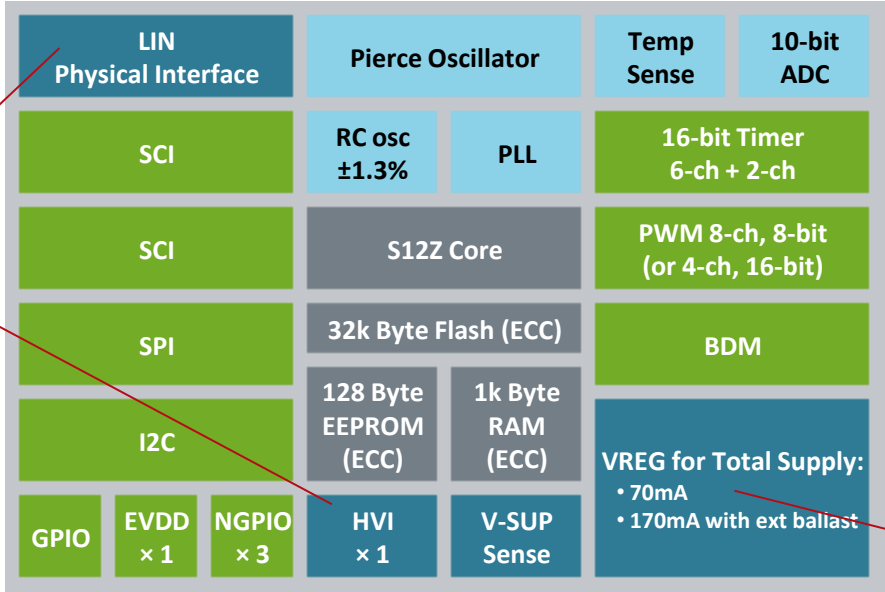
Overview of S12ZVL Feature Set

LIN Physical Interface

- 250kbps Fast Mode
- ±6kV (IEC61000-4-2)

1 × High Voltage Input

- with Interrupt capability



Voltage Regulator

- 5.5V to 18V normal operating range
- protected operation up to 40V
- 3.5V minimum operating voltage
- support for external ballast transistor

S12ZVL Key Features



RGB ambient lighting



Sensor (airflow)



Switchpanels / Userinterfaces

LIN Physical Interface		Pierce Oscillator		Temp Sense	10-bit ADC
SCI		RC osc ±1.3%	PLL	16-bit Timer 6-ch + 2-ch	
SCI		S12Z Core		PWM 8-ch, 8-bit (or 4-ch, 16-bit)	
SPI		32k Byte Flash (ECC)		BDM	
I2C		128 Byte EEPROM (ECC)	1k Byte RAM (ECC)	VREG for Total Supply: • 70mA • 170mA with ext ballast	
GPIO	EVDD × 1	NGPIO × 3	HVI × 1	V-SUP Sense	

Key Feature`	Benefits
S12Z core, 25 MHz Bus	Improved code-efficiency & core performance vs. S12
8 KB to 32 KB flash	Family-concept, no need for external Flash/EPROM/ROM
Up to 128 Bytes EEPROM	4B erasable page; Easier to use vs. data flash; no need for EE-emul.
all memories (Flash, RAM, EE) with ECC	Error Code Correction provides high reliability
Built-in automotive voltage regulator operating between 3.5 and 40 V	Operates directly from car-battery, no need for extra Voltage-regulator saving PCB-board-space. handles automotive design issues, such as double battery, crank voltage and load dump conditions
Built-in LIN physical layer	No need for an external LIN physical layer device, saving space and design time. Meets automotive OEM specifications for LIN conformance and EMC requirements
NGPIO & EVDD	EVDD able to supply 5V/20 mA offchip; N-GPIO able to sink up to 3 x 25mA off Chip (useful for RGB-LED-drive)
Protected 12V input (HVI)	Allows automotive battery voltage-level inputs (with ADC-capability & ESD-protection)
on Chip RC oscillator trimmed to 1,3% tolerance	Due to accurate on chip clock generation LIN-communication can be done without external Crystal or Resonator and without the need of SW-intensive synchronization

Package Options:

- 48 LQFP, 32 LQFP
- 32QFN (5 x 5 mm)

S12ZVC Key Features



Seatbelt pretention



Sensor (Powertrain)



Switchpanels / Userinterfaces

CAN Physical Interface

- High Speed – up to 1Mbps
- ISO 11898-2 & ISO 11898-5

High-Resolution Timer

- 20ns with 25MHz bus

High-Resolution PWM

- 20ns with 25MHz bus

Seperate CAN Supply

- with external ballast transistor

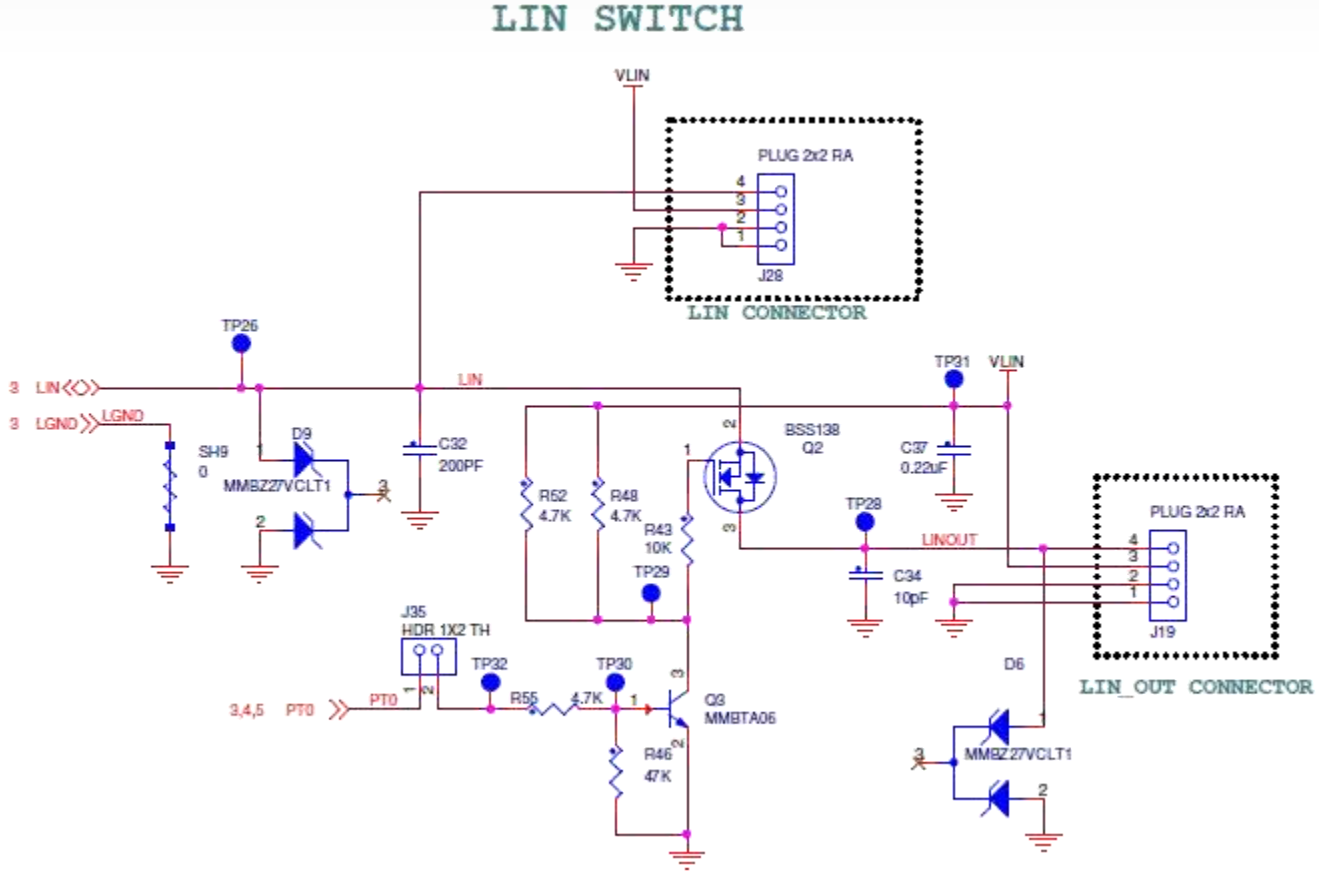
CAN Physical Interface	Pierce Oscillator		Temp Sense	12-bit ADC
MS-CAN	RC osc ±1.3%	PLL	Analog Comparator with DAC	
SCI	S12Z Core		High-Res. Timer 4-ch, 16-bit	
SCI	192k Byte Flash (ECC)		High-Res. PWM 4-ch, 16-bit	
SPI			Timer 8-ch, 16-bit	
I2C	2k Byte EEPROM (ECC)	8k Byte RAM (ECC)	PWM 4-ch, 16-bit	
BDM			VREG for Total Supply: • 70mA • 170mA with ext ballast	
GPIO 4-channel open drain	HVI × 2	V-SUP Sense	CAN Supply: • with ext ballast	

Key Feature	Benefits
S12Z core, 32 MHz Bus	Improved code-efficiency & core performance versus S12
64 KB to 192 KB flash	Family-concept, no need for external Flash/EPROM/ROM
Up to 2k Bytes EEPROM	4B erasable page; Easier to use vs. data flash; no need for EE-emul.
all memories (Flash, RAM, EE) with ECC	Error Code Correction provides high reliability
Built-in automotive voltage regulator operating between 3.5 and 40 V	Operates directly from car battery without the need for extra voltage regulator, saving PCB board space. Handles automotive design issues, such as double battery, crank voltage and load dump conditions
Built-in CAN physical layer	No need for an external CAN physical layer device, saving space as well as time & cost for design & testing. Meets automotive OEM requirements for CAN conformance and EMC.
Analog Comparator with DAC	Allows for fast reaction of ext. signals vs. preprogrammed threshold
Protected 12V input (HVI)	Allows automotive battery voltage-level inputs (with ADC-capability & ESD-protection)
12-Bit ADC; 16ns-Timer/PWM	High resolution mixed signal peripherals for many sensor-applications (e.g. ultrasonic)

Package Options:

- 64 LQFP
- 48 LQFP

TRK-S12ZVL: LIN Daisychain



S12ZVL: S12 MagniV Mixed-Signal MCU for LIN Applications ☆

Overview **Documentation** Software & Tools

Application Notes

PREPRODUCTION

The S12ZVL platform, part of the S12 MagniV mixed-signal MCU family, offers a low-cost, highly integrated solution that enables the design of smallest possible automotive LIN nodes, while the family concept provides scalability for platform design. The S12ZVL integrates a sophisticated S12Z core together with a 12V to 5V voltage regulator and a LIN physical layer for automotive applications such as sensors, actuators switch panels or lighting.

This page contains information on a preproduction product. Specifications and information herein are subject to change without notice.

For additional information and sample availability, contact your local [Freescale Sales Office](#).

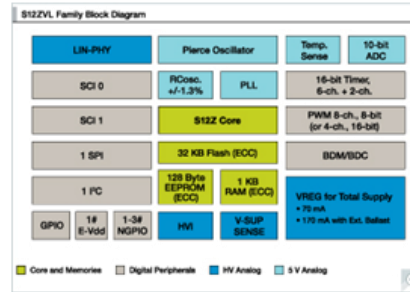
Features

- Enhanced S12Z core at 25 MHz bus speed
- Up to 32 KB Flash (with ECC)
- 128B EEPROM (with ECC)
- Up to 1 KB RAM (with ECC)
- Vreg for 3.5 to 20-volt operating range, scalable in supply for on- and off-chip systems
- LIN physical layer
- On-chip RC-oscillator 1.3% accurate

Related Products

- [S12ZR: S12 MagniV Mixed-Signal MCU for Relay Driven Motor Applications](#)
- [S12ZVC: S12 MagniV Mixed-Signal MCU for CAN Applications](#)
- [S12ZVH: S12 MagniV Mixed-Signal Microcontroller for Automotive Instrument Cluster Applications](#)

[More ▾](#)



Featured Documentation

- [S12ZVLFS: S12 MagniV S12ZVL Family - Fact Sheet](#)
- [BODYELECTRWP: Future Advances in Body Electronics - White Paper](#)

Target Applications

- **Automotive**
 - Heating Ventilation and Air Conditioning (HVAC)
 - Lighting
 - Doors, Window Lift and Seat Control
 - Steering Wheel Controllers
 - Watchdog Controller for Chassis/Safety/Powertrain
 - LIN Nodes
 - LIN User Interface
 - LIN Switch Panel
 - LIN Actuators, Sensors
- **Industrial**
 - Ambient Lighting Control

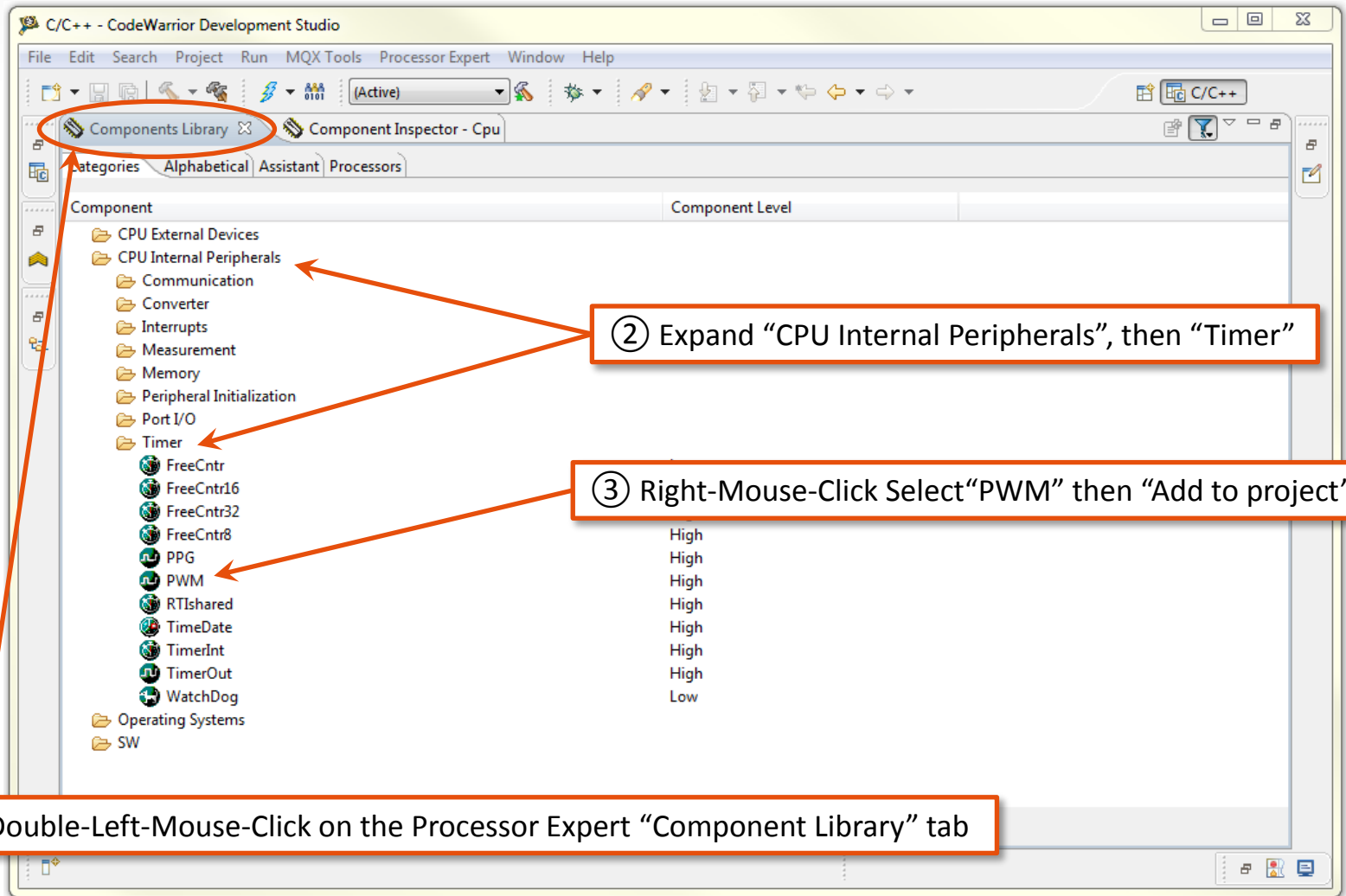
Functional Safety

SafeAssure Functional Safety Program Solutions targeted to help meet IEC 61508 and ISO 26262 functional safety compliance

Connect With Us

Automobility Blog by Freescale's automotive experts

Add a PWM Component to the Project



④ Double-Left-Mouse-Click on the Processor Expert "Component Library" tab

② Expand "CPU Internal Peripherals", then "Timer"

③ Right-Mouse-Click Select "PWM" then "Add to project"

Change PWM Component Properties

The screenshot shows the CodeWarrior Development Studio interface. The 'Component Inspector' window is open for the 'PWM_RED' component. The 'Expert' tab is selected. The 'Properties' tab is active, showing a table of component properties. Annotations are present:

- Annotation 7:** A box pointing to the 'Component name' and 'PWM or PPG device' rows in the table, containing the text: "Change the Component name to 'PWM_RED' and select the PWM channel 'PWMPER1'".
- Annotation 8:** A box pointing to the 'Period' row in the 'Value' column, containing the text: "Select anywhere in the Value column beside 'Period' and then select the elipsis ('...')".
- Annotation 6:** A box pointing to the 'Expert' tab in the top right, containing the text: "Select 'Expert'".

Name	Value	Details
Component name	PWM_RED	
PWM or PPG device	PWMPER1	PWMPER1
Duty compare	PWMDTY1	PWMDTY1
Output pin	PP1_KWP1_PWM1_IOC1_1	PP1_KWP1_PWM1_IOC1_1
Output pin signal		
Counter	PWM1	PWM1
Interrupt service/event	Disabled	
Period		Unassigned timing
Starting pulse width		Unassigned timing
Aligned	Left	
Initial polarity	low	
Same period in modes	no	
Component uses entire timer	no	
Initialization		
Enabled in init. code	yes	
Events enabled in init.	yes	
CPU clock/speed selection		
High speed mode	This component enabled	This component is enabled
		this component is disabled
		this component is disabled

⑦ Change the Component name to "PWM_RED" and select the PWM channel "PWMPER1"

⑧ Select anywhere in the Value column beside "Period" and then select the elipsis ("...")

⑥ Select "Expert"

Change PWM Component Properties

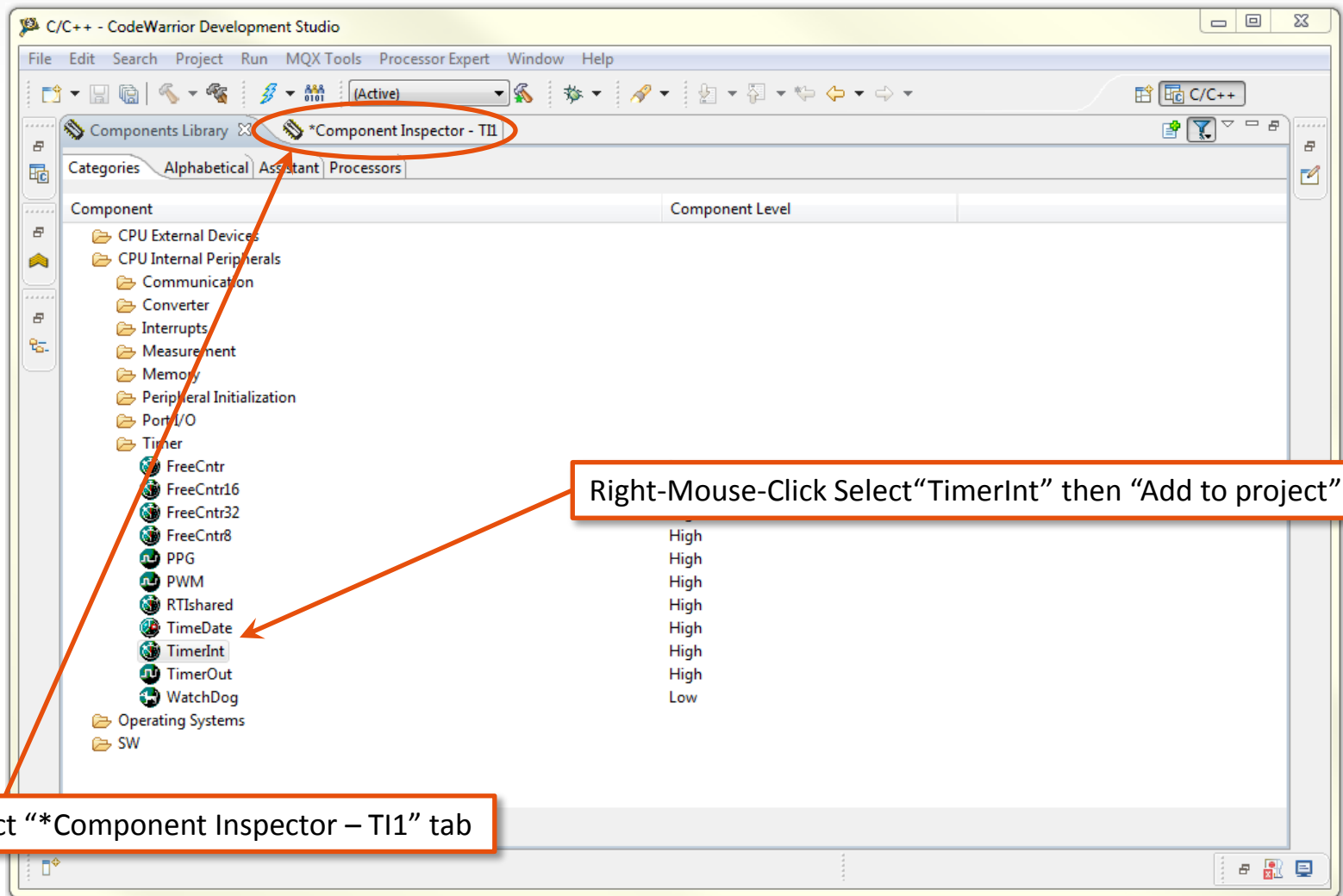
The screenshot displays the 'Component Inspector - PWM_RED' window. The 'Methods' tab is selected. The table below shows the component's properties:

Name	Value	Details
Component name	PWM_RED	
PWM or PPG device	PWMPER1	PWMPER1
Duty compare	PWMDTY1	PWMDTY1
Output pin	PP1_KWP1_PWM1_IOC1_1	PP1_KWP1_PWM1_IOC1_1
Output pin signal		
Counter	PWM1	PWM1
Interrupt service/event	Disabled	
Period	4 ms	3.815 ms
Starting pulse width	0 ms	0 ms
Aligned	Left	
Initial polarity	low	
Same period in modes	no	
Component uses entire timer	no	
Initialization		
Enabled in init. code	yes	
Events enabled in init.	yes	
CPU clock/speed selection		
	not enabled	This component is enabled
	not disabled	This component is disabled
	not disabled	This component is disabled

10 Repeat with the “Starting pulse width” by making it “0”ms.

11 Select the “Methods” tab

Add a Periodic Timer Component to the Project



Add Code to ProcessorExpert.c

```

C/C++ - Lab/Sources/ProcessorExpert.c - CodeWarrior Development Studio
File Edit Search Project Run MQX Tools ProcessorExpert Window Help
Cpu.c *ProcessorExpert.c
41 /* User includes (#include below this line is not maintained by Processor Expert) */
42
43 volatile uint8_t tick;
44 static uint8_t tick_last;
45
46 void demo (void)
47 {
48     static uint8_t state = 5;
49     static uint8_t duty_RED = 0x00;
50     static uint8_t duty_GREEN = 0xFF;
51     static uint8_t duty_BLUE = 0x00;
52
53     if (tick == 0)
54     {
55         state++;
56         if (state == 6)
57         {
58             state = 0;
59         }
60     }
61
62     switch (state)
63     {
64     case 0: // go to Red
65         duty_BLUE++;
66         break;
67     case 1: // go to Yellow
68         duty_GREEN--;
69         break;
70     case 2: // go to Green
71         duty_RED++;
72         break;
73     case 3: // go to Aqua
74         duty_BLUE--;
75         break;

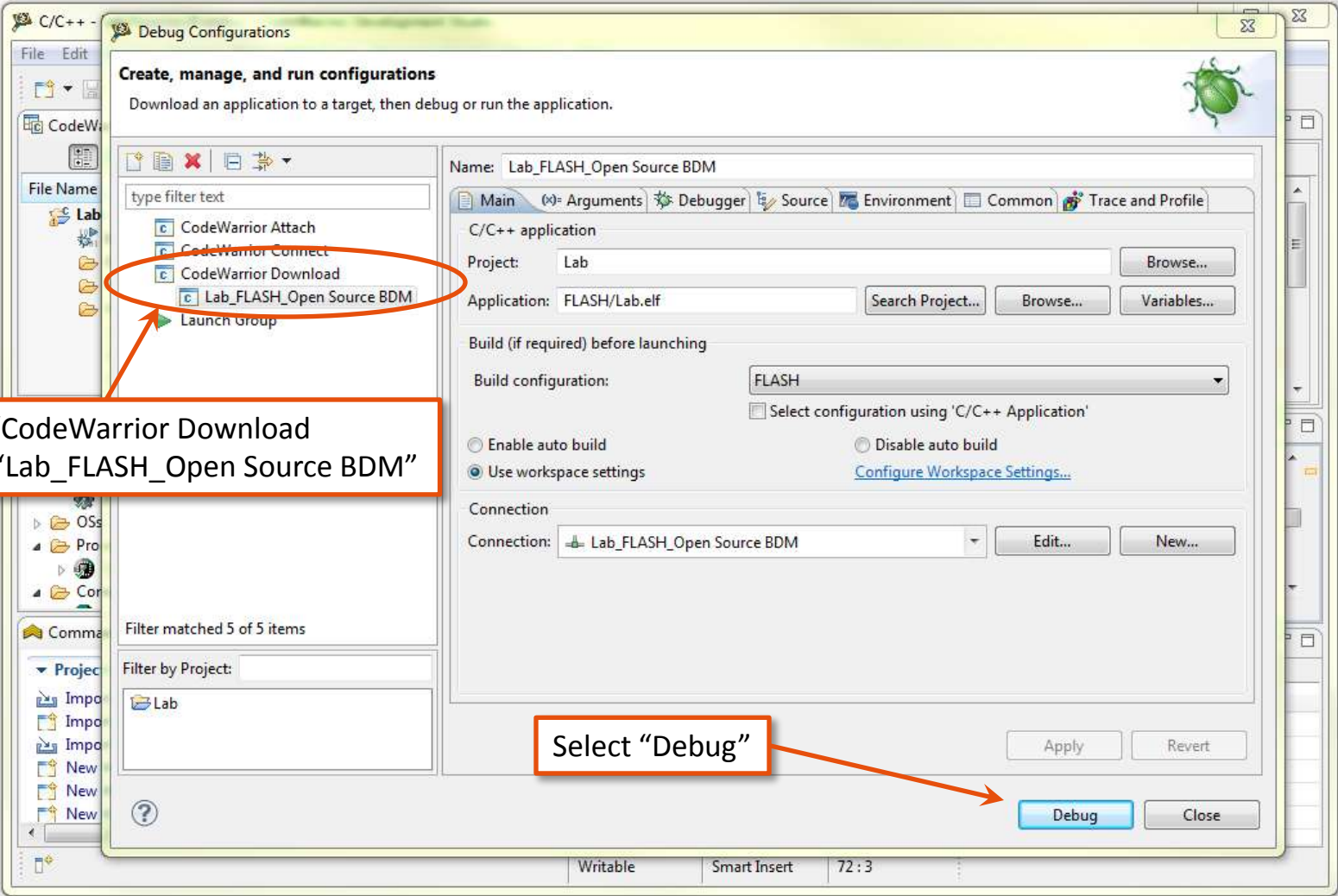
```

The image shows a code editor window with several annotations on the right side of the code:

- A bracket on the right groups lines 43 and 44, with a label **Global variables**.
- A bracket on the right groups lines 48 through 51, with a label **Local variables**.
- A bracket on the right groups lines 53 through 75, with a label **"state" control**.

At the bottom of the editor window, the status bar shows "Writable", "Smart Insert", and "45 : 1".

Select the Debug Configuration



Select "CodeWarrior Download then "Lab_FLASH_Open Source BDM"

Select "Debug"

