

DAVE™ (Version 4) – Introduction

Learning Outcome

- › DAVE™ development platform for software development
- › DAVE™ highlights
 - Component based programming
 - GUI based configuration
 - Code repository
 - Hardware resource manager
 - Code generation
 - Support 3rd party tools
 - Expert support
 - DAVE™ SDK



DAVE™

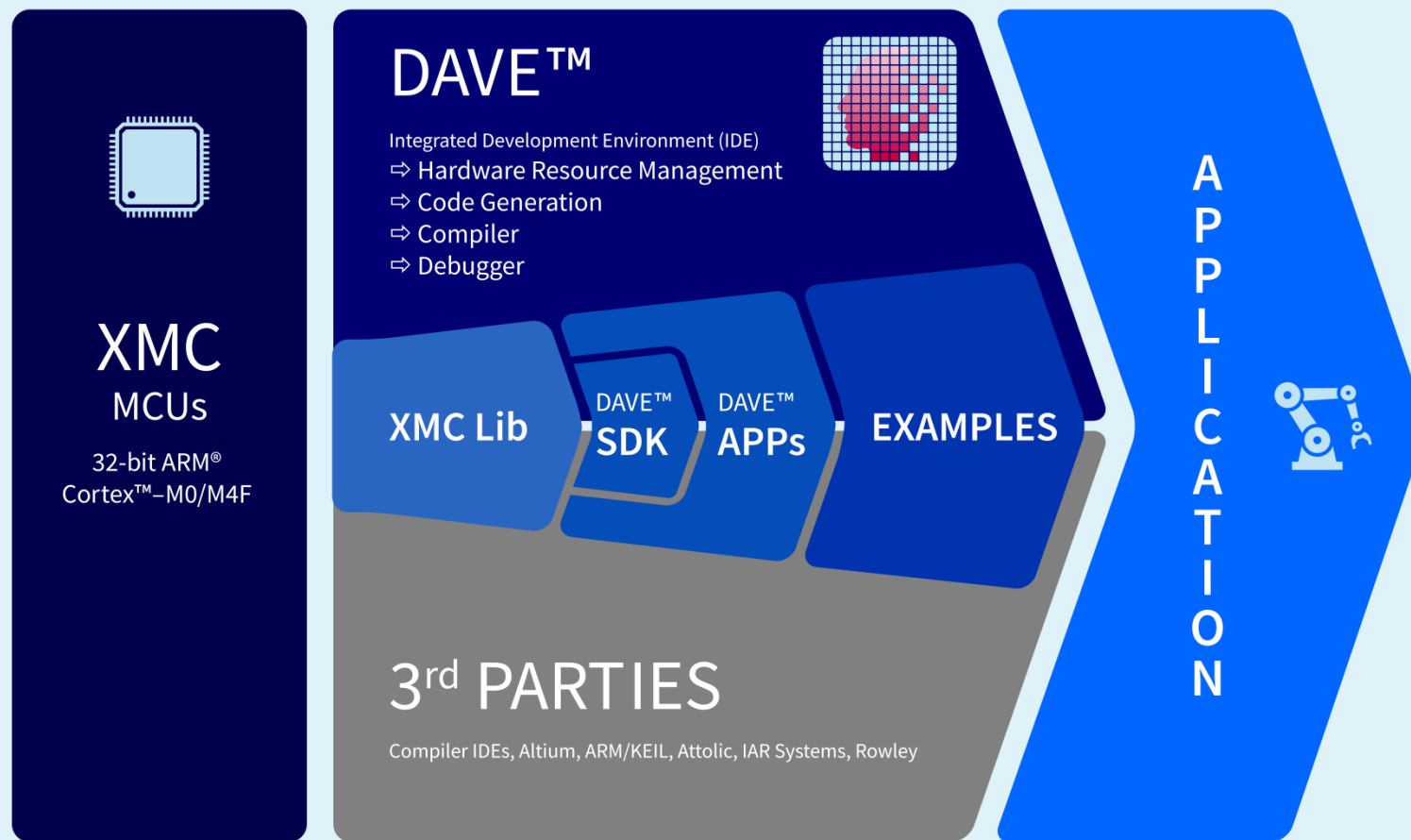
Free Eclipse based IDE offering code repository, graphical system design methods, and automatic code generator to guide XMC microcontroller user along the entire process – from evaluation to production (E2P).

XMC Lib and DAVE™ generated code can be used with other 3rd party tool chains.

Digital
Application
Virtual
Engineer

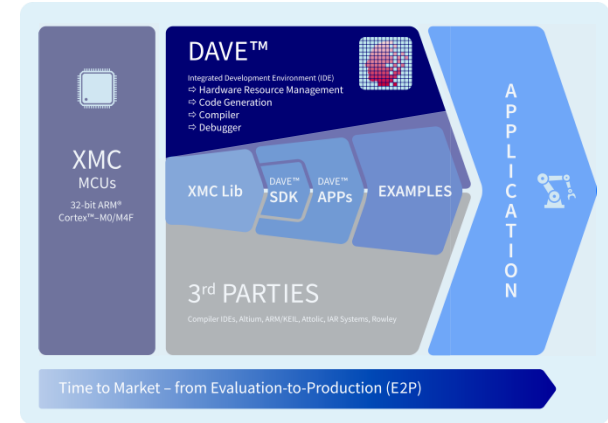
XMC Microcontroller

– Software development made easy

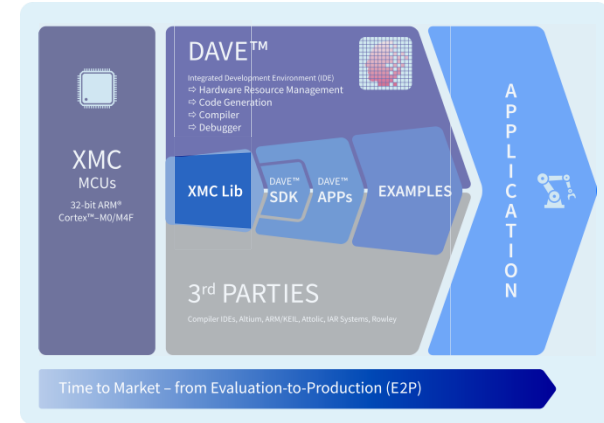


Time to Market – from Evaluation-to-Production (E2P)

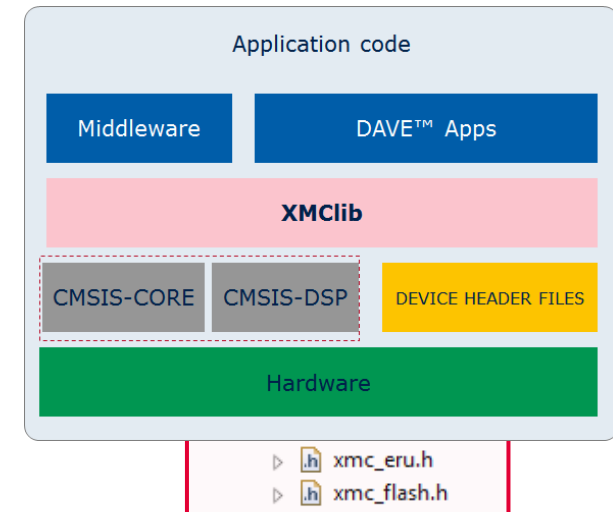
- Eclipse based Integrated Development Environment (IDE)
 - ARM GNU C-Compiler
 - GDB SEGGER J-Link debugger
 - Code generation based on user configuration using graphical user interface (GUI)
 - Hardware resource solver



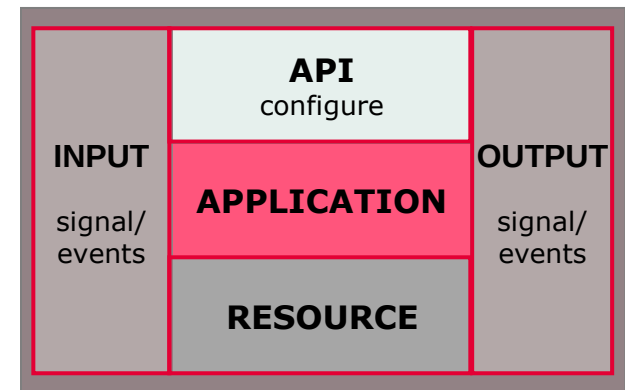
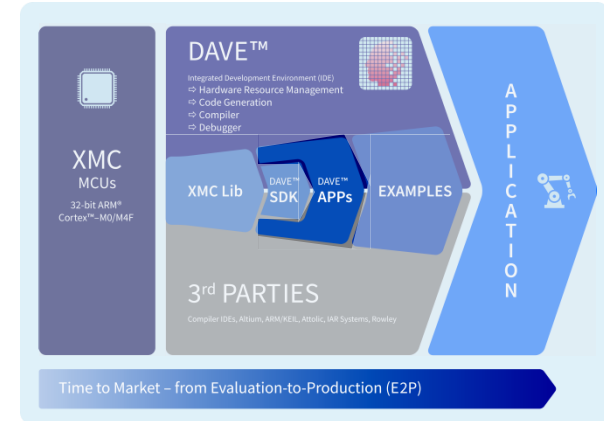
- Fully tested low level driver library for XMC peripherals (APIs)
- CMSIS / MISRA 2004 compliant
- Covers all peripheral functions and features
- XMC Lib is part of any new DAVE™ project



C/C++ Projects Project Expl

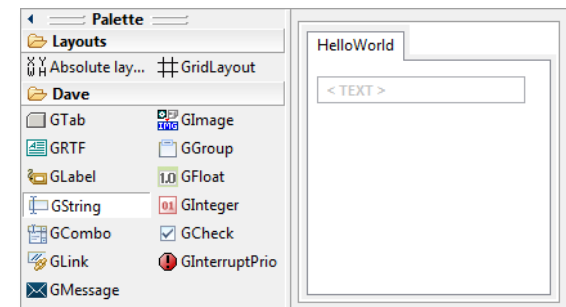
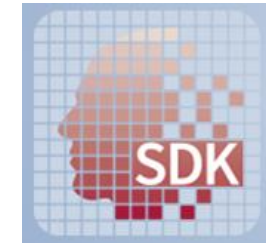
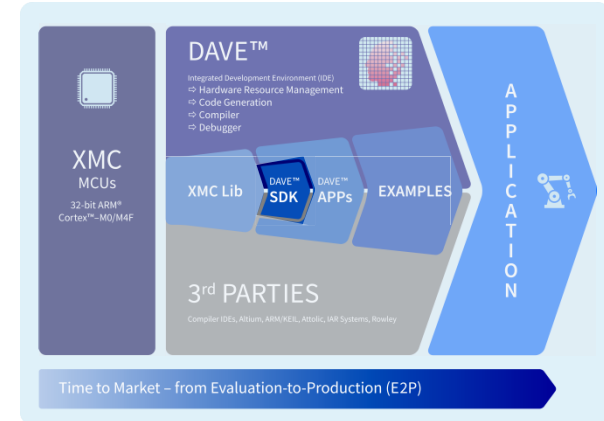


- Application oriented software components that abstract a certain use case
- Built on top of XMC Lib
- Graphical User Interface (GUI) configurable

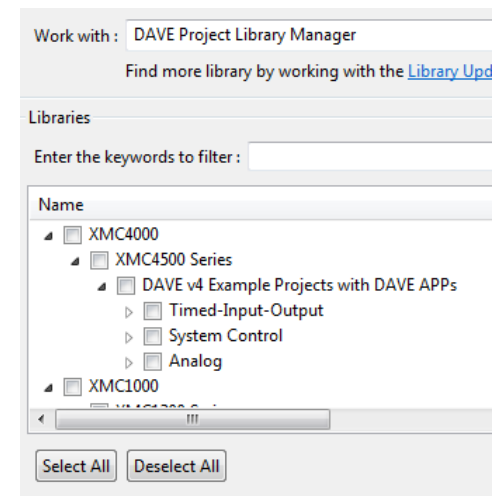
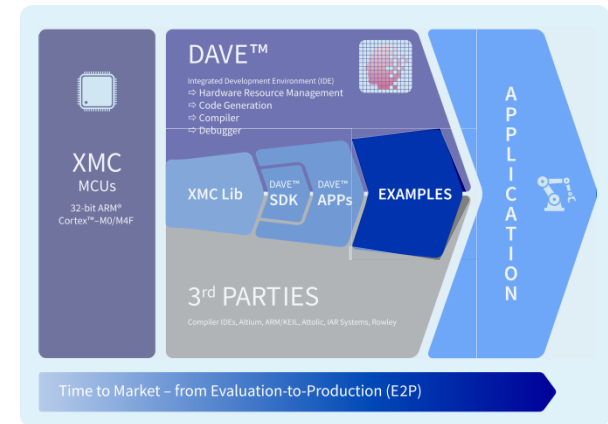


DAVE™ version 4 support a new set of enhanced DAVE™ APPs

- Software Development Kit for DAVE™ APPs
- Modify, enhance and create new DAVE™ APPs for any use case
- Code templates based on groovy script
- GUI designer with drag and drop functionality for widgets
- Debug features supported



- XMC Lib and DAVE™ APPs composed to applications
- Example projects as references for complete applications
- Download examples with DAVE project library manager



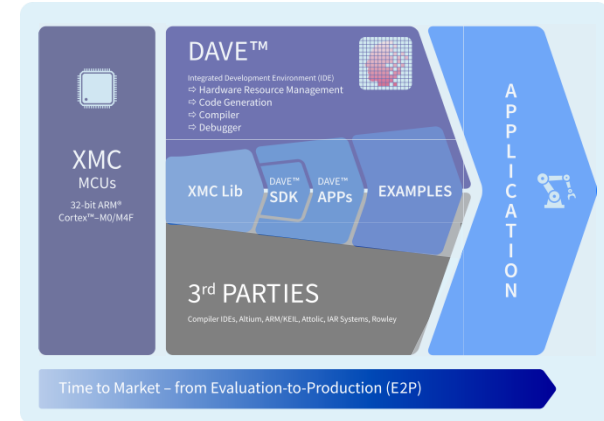
■ XMC Lib and DAVE™ generated Code

are tested with

- GCC compiler
- ARM® compiler
- TASKING compiler

■ And released for

- Altium
- ARM/KEIL
- Atollic
- IAR Systems
- Rowley



Compiler IDEs

TASKING

KEIL™
Tools by ARM

a atollic



IAR
SYSTEMS

Motor Control



- Asynchronous Motors (FOC, Frequency Control)
- PMSM, BLDC (FOC, Scalar, Hall Sensor)
- PWM Generation
 - ▢ Space Vector
 - ▢ Block Commutation
- Position Detection (Hall, Encoder, Resolver)
- Drive Automation

Power Conversion



- Buck Converter
 - ▢ Peak Current Control
 - ▢ Voltage Control
- PWM Generation using HRPWM

Lighting



- LED Lamp Control
- DALI
- DMX512

Communication



- USIC
 - ▢ UART
 - ▢ SPI
 - ▢ I2C
- USB
- Ethernet

General Purpose



- Timer/PWM (Capture, Compare)
- ADC
- DAC
- GPIO

HMI



- SEGGER GUI Library
- Display
- Touch

System



- Interrupt
- DMA
- AES
- CRC
- RTOS
- File System
- Emulated EEPROM

XMC Lib	CMSIS / MISRA 2004 compliant low level driver libraries (APIs) for XMC MCU peripherals				
	System	Timer/PWM	Analog-mixed Signal	Communication	Application specific
	• DMA • ERU • FCE • FLASH • GPIO • MATH • PAU • PRNG • RTC • SCU • WDT	• CCU4 • CCU8 • HRPWM • POSIF	• ACMP • ADC • DAC	• CAN • I2C • SPI • UART • USB • USIC	• BCCU • LEDTS • MATH • POSIF • HRPWM
XMC APPs	Graphical User Interface (GUI) configurable application oriented software components using XMC Lib (Low Level Driver); arranged in a library (APIs)				
	General Purpose		Application specific	And many more ...	
	• ADC_MEASUREMENT • ADC_QUEUE • ADC_SCAN • CAN_NODE • CAPTURE • CLOCK • COUNTER • CRC_HW • CRC_SW • CRYPTO_AES • DAC • DAC_BCCU • DAC_LUT • DAC_SWEEP • DBG_PIN • DMA_CH • E_EEPROM • EVENT_DETECT • EVENT_GEN • EXT_INTERRUPT • GPIO • INTERRUPT • IO_PORT • PRNG_HW • PWM • PWM_BC • PWM_CCU4 • PWM_CCU8 • PWM_SVM • RTC • SYSTIMER • TEMPERATURE • TIMER • WATCHDOG	• ACIM_FOC • ACIM_FREQ_CTRL • BLDC_SCALAR_CTRL • MOTOR_LIB • PMSM_FOC • PMSM_SCALAR_CTRL • QUAD_ENC_POSIF • RESOLVER • BUCK_CONVERTER • CURRENTCTRL • HRPWM • POWER_MGMT • PDM_BCCU	• DALI_CD • DALI.CG • DISPLAY_14SEG • DISPLAY_7SEG • FTP • GUI_KeyBoard • GUI_LCD • GUI_Mouse • GUI_SeggerLib • HALL_POSIF • HTTP • I2C_MASTER • I2C_SLAVE • LED_LAMP • LED_MATRIX • MANCHESTER_SW • MODBUS	• RTOS • SNMP • SP_MASTER • SPI_SLAVE • TOUCH_BUTTONS • TOUCH_PAD • TOUCH_SLIDER • TOUCH_WHEEL • UART • USBD • USBD_HID • USBD_MS • USBD_VCOM • USBD_WINUSB • USBH • USBH_HID	
DAVE™ EXAMPLES	XMC Lib (Low Level Driver for XMC MCUs) and DAVE™ APPs composed to applications				



1



1

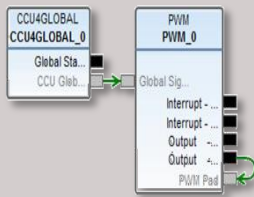
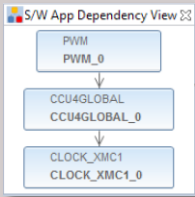


4

- 

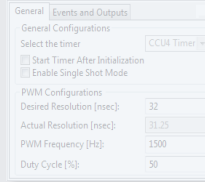


1



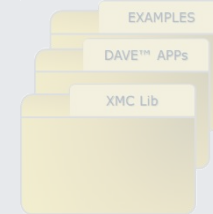
Component based programming

1



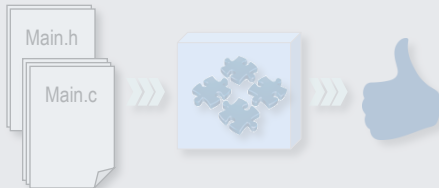
GUI based configuration

2



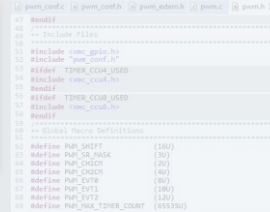
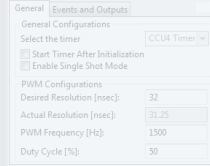
Code repository

3



Hardware resource manager

4



Code generation

5

- Released for
- Altium Limited
 - Atollic
 - ARM/KEIL
 - IAR Systems
 - Rowley Associates

TASKING

a atollic

KEIL
Tools by ARMIAR
SYSTEMS

Support 3rd party tools

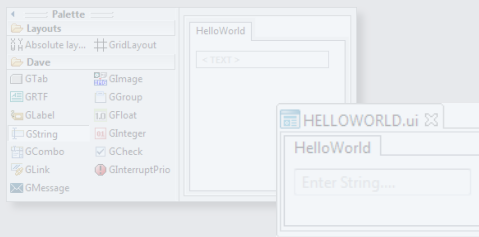
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

8



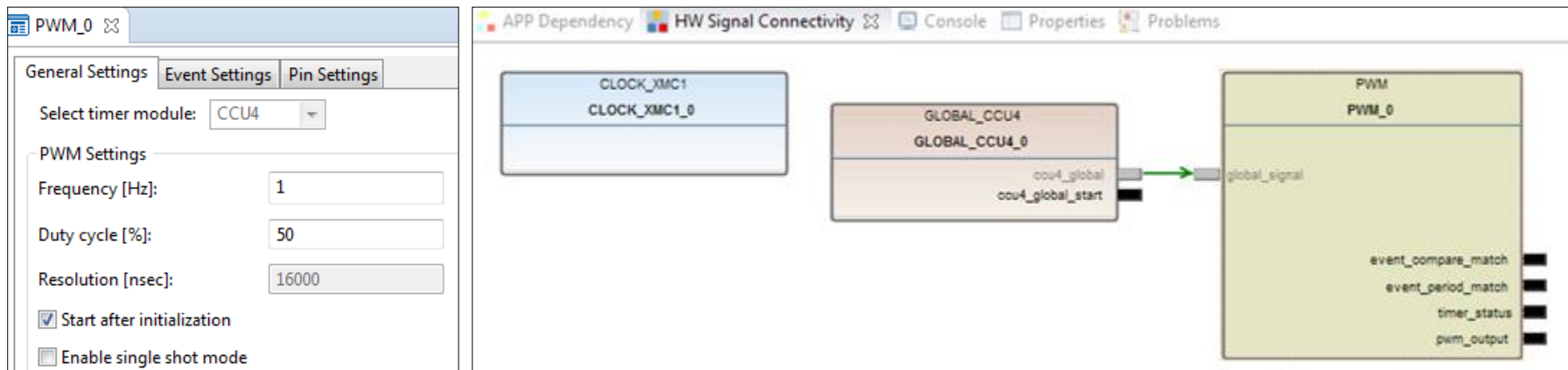
Free of charge

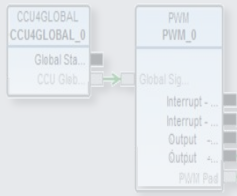
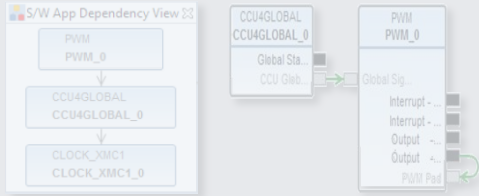
9

DAVE™

Component Based Programming

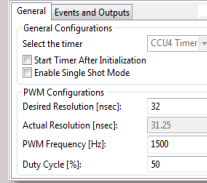
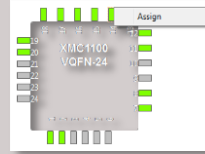
- › Build application with use-case oriented, reusable software components DAVE™ APPs
- › Configure DAVE™ APPs and connect hardware signals with graphical user interface
- › APIs of the XMC Lib can be used with DAVE™ APPs for full flexibility of peripherals and connectivity





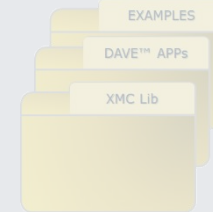
Component based programming

1



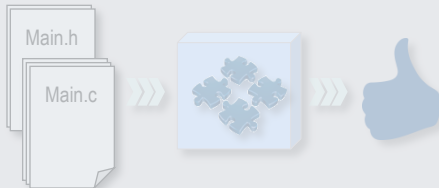
GUI based configuration

2



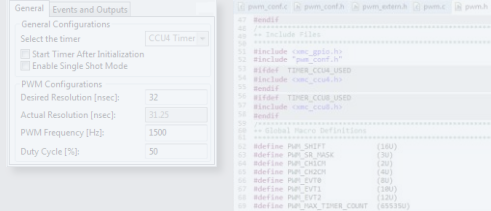
Code repository

3



Hardware resource manager

4



Code generation

5



Support 3rd party tools

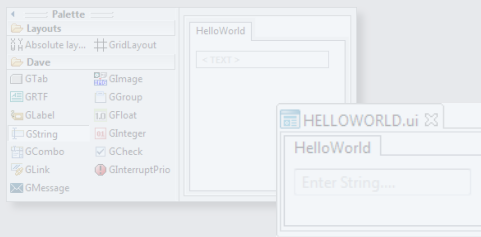
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

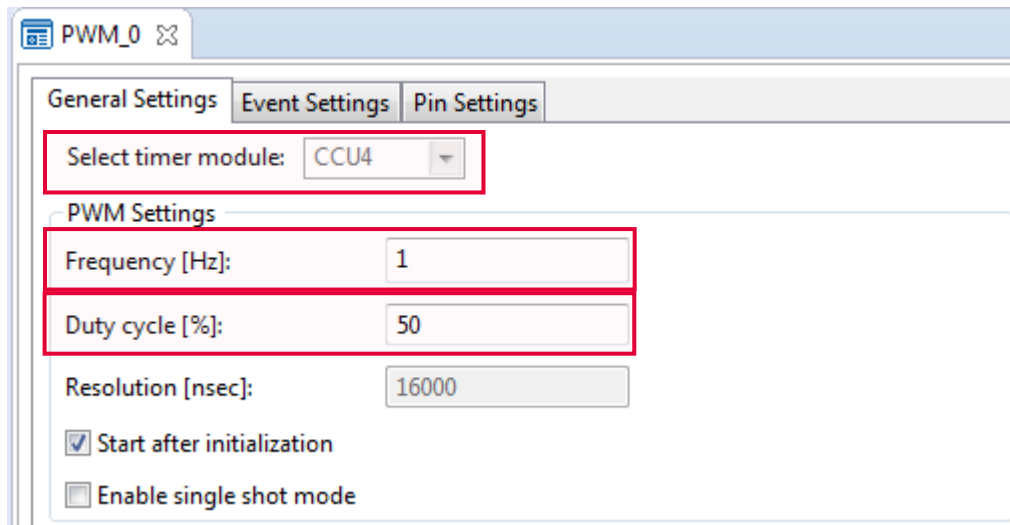
8



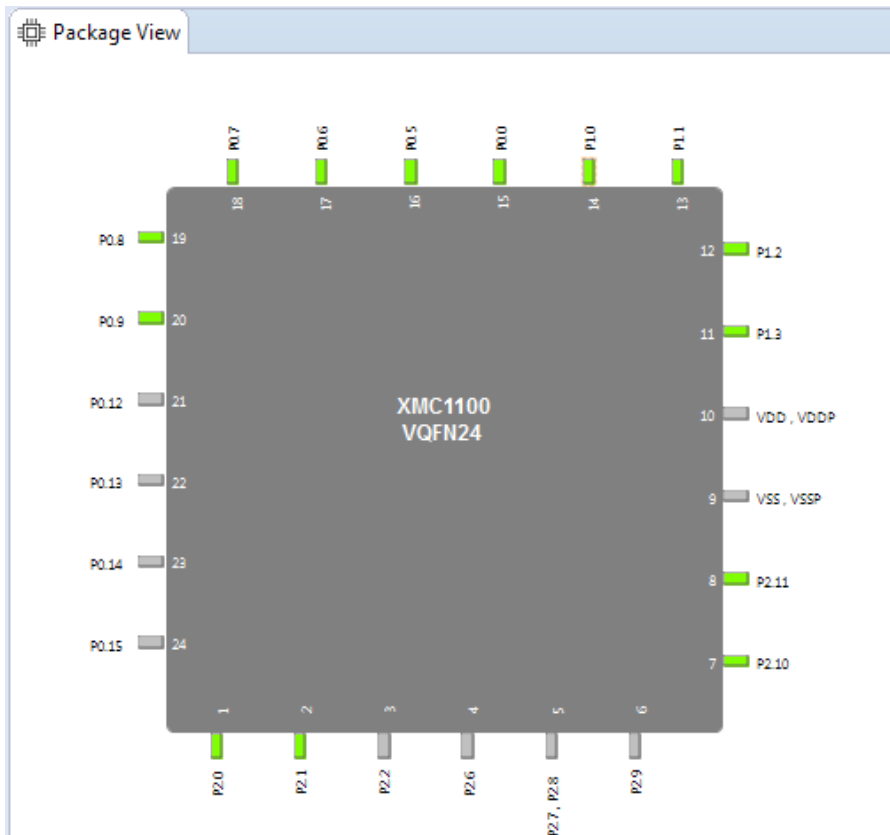
Free of charge

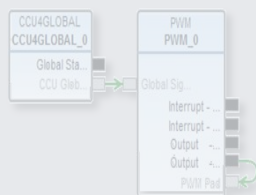
9

- › An application use case can be configured quickly with a graphical user interface
 - Example: Configure timer, frequency and duty cycle to generate PWM waveform with PWM APP



- › Graphical pin mapping
 - Pin package represented in a graphical view
 - Assign signal to one of the possible pins marked in green





Component based programming

1



General Events and Outputs	
General Configurations	
Select the timer: CCUR Timer	
☐ Start Timer After Initialization ☐ Enable Single Shot Mode	
PWM Configurations	
Desired Resolution [nsec]:	32
Actual Resolution [nsec]:	31.25
PWM Frequency [Hz]:	1300
Duty Cycle [%]:	50

GUI based configuration

2



Code repository

3



Hardware resource manager

4

General Events and Outputs	
General Configurations	
Select the timer: CCUR Timer	
☐ Start Timer After Initialization ☐ Enable Single Shot Mode	
PWM Configurations	
Desired Resolution [nsec]:	32
Actual Resolution [nsec]:	31.25
PWM Frequency [Hz]:	1300
Duty Cycle [%]:	50

```

1 // PWM configuration
2 #include "pwm_conf.h"
3 #include "pwm_conf.h"
4 #include "pwm_conf.h"
5 #include "pwm_conf.h"
6 #include "pwm_conf.h"
7 #include "pwm_conf.h"
8 #include "pwm_conf.h"
9 #include "pwm_conf.h"
10 #include "pwm_conf.h"
11 #include "pwm_conf.h"
12 #include "pwm_conf.h"
13 #include "pwm_conf.h"
14 #include "pwm_conf.h"
15 #include "pwm_conf.h"
16 #include "pwm_conf.h"
17 #include "pwm_conf.h"
18 #include "pwm_conf.h"
19 #include "pwm_conf.h"
20 #include "pwm_conf.h"
21 #include "pwm_conf.h"
22 #include "pwm_conf.h"
23 #include "pwm_conf.h"
24 #include "pwm_conf.h"
25 #include "pwm_conf.h"
26 #include "pwm_conf.h"
27 #include "pwm_conf.h"
28 #include "pwm_conf.h"
29 #include "pwm_conf.h"
30 #include "pwm_conf.h"
31 #include "pwm_conf.h"
32 #include "pwm_conf.h"
33 #include "pwm_conf.h"
34 #include "pwm_conf.h"
35 #include "pwm_conf.h"
36 #include "pwm_conf.h"
37 #include "pwm_conf.h"
38 #include "pwm_conf.h"
39 #include "pwm_conf.h"
40 #include "pwm_conf.h"
41 #include "pwm_conf.h"
42 #include "pwm_conf.h"
43 #include "pwm_conf.h"
44 #include "pwm_conf.h"
45 #include "pwm_conf.h"
46 #include "pwm_conf.h"
47 #include "pwm_conf.h"
48 #include "pwm_conf.h"
49 #include "pwm_conf.h"
50 #include "pwm_conf.h"
51 #include "pwm_conf.h"
52 #include "pwm_conf.h"
53 #include "pwm_conf.h"
54 #include "pwm_conf.h"
55 #include "pwm_conf.h"
56 #include "pwm_conf.h"
57 #include "pwm_conf.h"
58 #include "pwm_conf.h"
59 #include "pwm_conf.h"
60 #include "pwm_conf.h"
61 #include "pwm_conf.h"
62 #include "pwm_conf.h"
63 #include "pwm_conf.h"
64 #include "pwm_conf.h"
65 #include "pwm_conf.h"
66 #include "pwm_conf.h"
67 #include "pwm_conf.h"
68 #include "pwm_conf.h"
69 #include "pwm_conf.h"
70 #include "pwm_conf.h"
71 #include "pwm_conf.h"
72 #include "pwm_conf.h"
73 #include "pwm_conf.h"
74 #include "pwm_conf.h"
75 #include "pwm_conf.h"
76 #include "pwm_conf.h"
77 #include "pwm_conf.h"
78 #include "pwm_conf.h"
79 #include "pwm_conf.h"
80 #include "pwm_conf.h"
81 #include "pwm_conf.h"
82 #include "pwm_conf.h"
83 #include "pwm_conf.h"
84 #include "pwm_conf.h"
85 #include "pwm_conf.h"
86 #include "pwm_conf.h"
87 #include "pwm_conf.h"
88 #include "pwm_conf.h"
89 #include "pwm_conf.h"
90 #include "pwm_conf.h"
91 #include "pwm_conf.h"
92 #include "pwm_conf.h"
93 #include "pwm_conf.h"
94 #include "pwm_conf.h"
95 #include "pwm_conf.h"
96 #include "pwm_conf.h"
97 #include "pwm_conf.h"
98 #include "pwm_conf.h"
99 #include "pwm_conf.h"
100 #include "pwm_conf.h"

```

Code generation

5

- Released for
- Altium Limited
 - Atollic
 - ARM/KEIL
 - IAR Systems
 - Rowley Associates

TASKING

atollic

KEIL
Tools by ARM

IAR
SYSTEMS



Support 3rd party tools

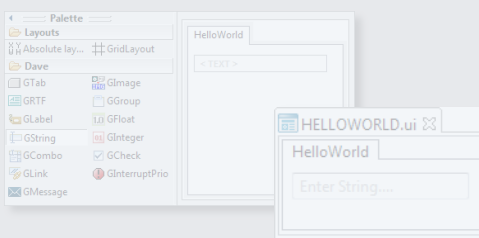
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

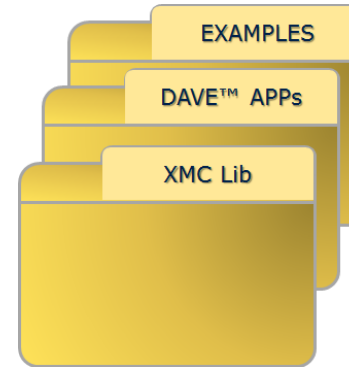
8

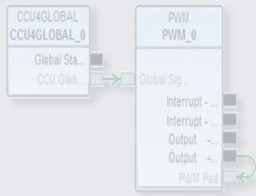
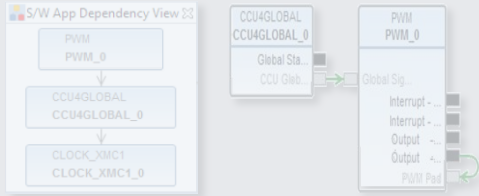


Free of charge

9

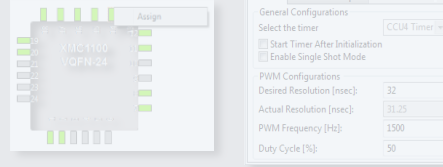
- › Access to free code libraries
 - DAVE™ APPs
 - EXAMPLES
 - XMC Lib (Low Level Driver)
- › Customizable for specific application or use case
- Get new APPs releases and updates with one-click update in DAVE™ IDE





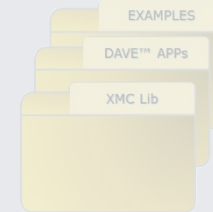
Component based programming

1



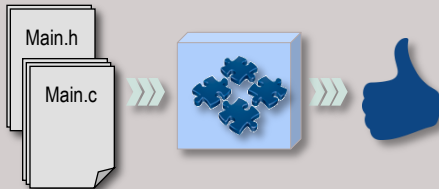
GUI based configuration

2



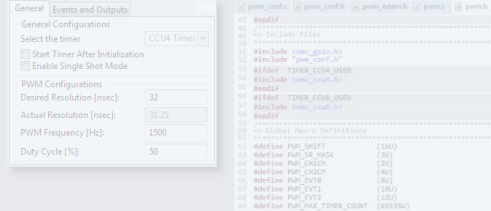
Code repository

3



Hardware resource manager

4



Code generation

5



Support 3rd party tools

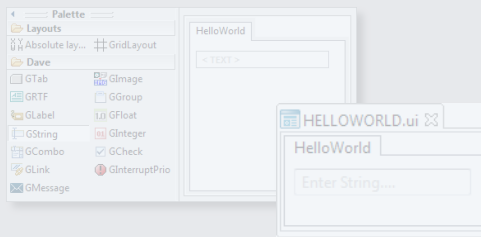
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

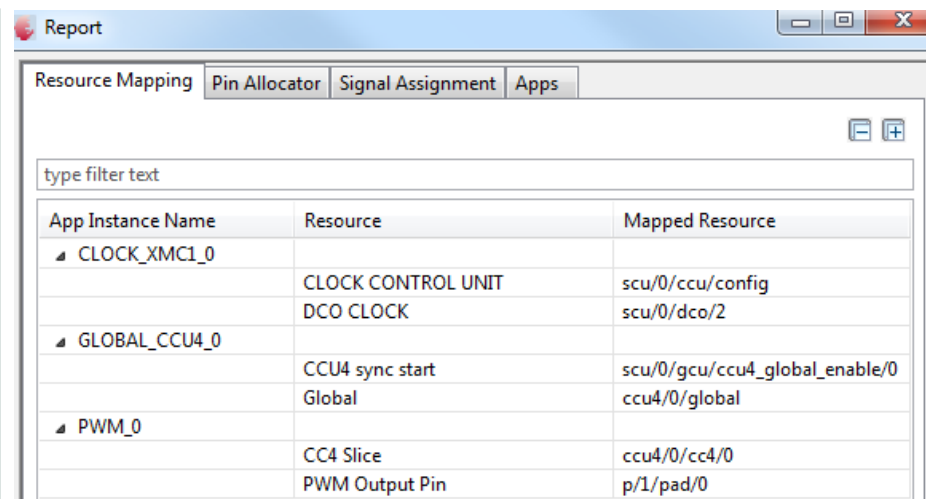
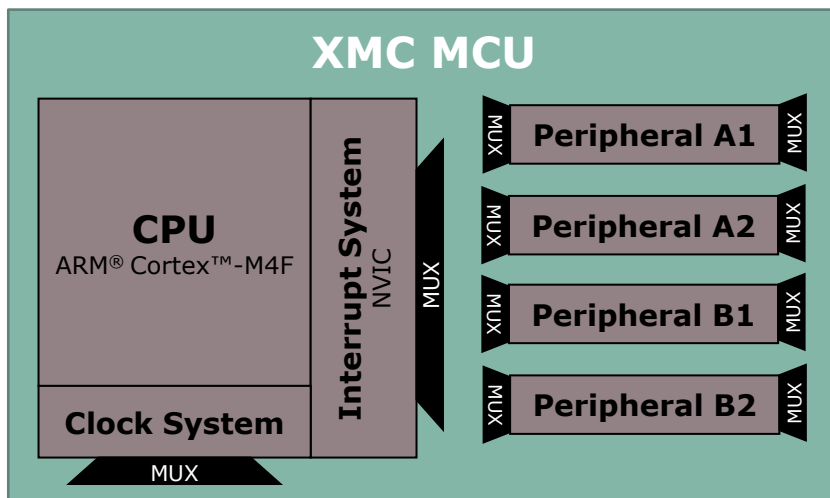
8



Free of charge

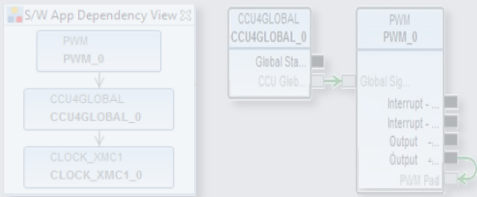
9

- › Resource solver manages chip hardware resources
 - Signal connections
 - Pin constraints
- › Automatically maps DAVE™ APPs to real peripherals and determines the right multiplexer settings
- › Automatically maps signal to pin (solver assigned pins)



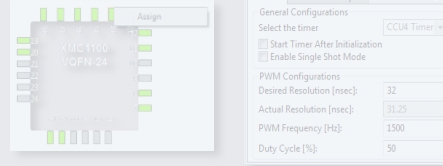
The screenshot shows the 'Report' window in the DAVE software, specifically the 'Resource Mapping' tab. It displays a table mapping application instance names to hardware resources and their mapped locations.

App Instance Name	Resource	Mapped Resource
▲ CLOCK_XMC1_0	CLOCK CONTROL UNIT	scu/0/ccu/config
	DCO CLOCK	scu/0/dco/2
▲ GLOBAL_CC4_0	CCU4 sync start	scu/0/gcu/ccu4_global_enable/0
	Global	ccu4/0/global
▲ PWM_0	CC4 Slice	ccu4/0/cc4/0
	PWM Output Pin	p/1/pad/0



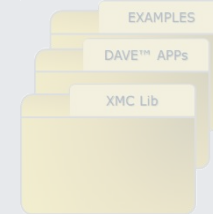
**Component
based programming**

1



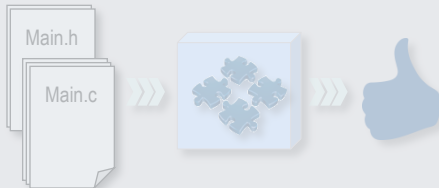
**GUI
based configuration**

2



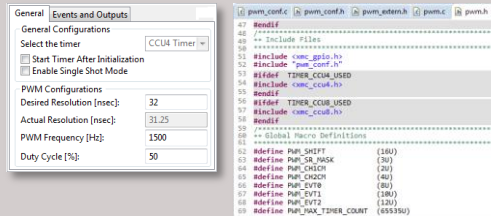
**Code
repository**

3



**Hardware
resource manager**

4



**Code
generation**

5



**Support
3rd party tools**

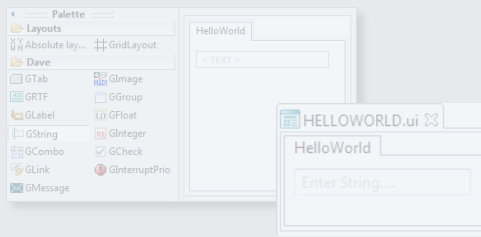
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



**Expert
support**

7



**DAVE™
SDK**

8

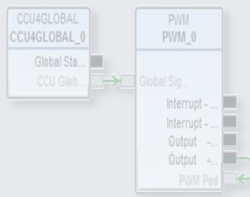


**Free
of charge**

9

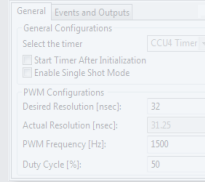
- › Automatic code generation in a single click 
- › Generate code based on DAVE™ APPs configuration
- › More efficient and readable code based on XMC Lib
- › Use generated code with
 - APIs of XMC Lib in DAVE™
 - 3rd party compiler tools

```
pwm_conf.c  pwm_conf.h  pwm.c  pwm.h  x
70  */
71  typedef enum PWM_TIMER_TYPE
72  {
73      PWM_TIMER_SLICE_CC04 = 0U,
74      PWM_TIMER_SLICE_CC08
75  } PWM_TIMER_TYPE_t;
76
77  /**
78   * @brief The type identifies the timer status.
79   */
80  typedef enum PWM_TIMER_STATUS
81  {
82      PWM_TIMER_IDLE = 0U,
83      PWM_TIMER_RUNNING
84  } PWM_TIMER_STATUS_t;
85
86  /**
87   * @brief The type identifies the timer interrupts.
88   */
89  typedef enum PWM_InterruptType
90  {
91      PWM_PERIODMATCH_INTERRUPT = 0U,    /**< Period match interrupt while counting up */
92      PWM_COMPAREMATCH_INTERRUPT = 2U,   /**< Compare match interrupt while counting up */
93  } PWM_InterruptType_t;
94
95  /**
96   * @brief The type identifies the timer interrupts.
97   */
98  typedef enum PWM_OutputPassiveLevelType
99  {
100      PWM_OUTPUT_PASSIVE_LEVEL_LOW = 0, /**< Passive level = Low */
101      PWM_OUTPUT_PASSIVE_LEVEL_HIGH, /**< Passive level = High */
102      PWM_OUTPUT_PASSIVE_LEVEL_MAX
103  } PWM_OutputPassiveLevelType_t;
104
```

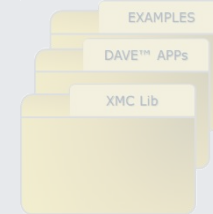
Component based programming

1



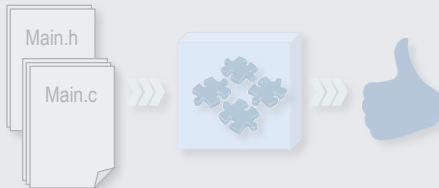
GUI based configuration

2



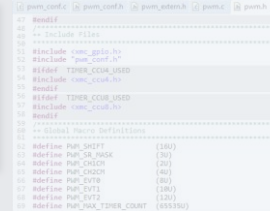
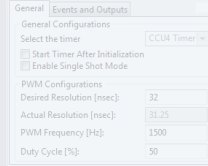
Code repository

3



Hardware resource manager

4



Code generation

5

- Released for
- Altium Limited
 - Atollic
 - ARM/KEIL
 - IAR Systems
 - Rowley Associates

TASKING

a10t0llic

KEIL
Tools by ARMIAR
SYSTEMS

Support 3rd party tools

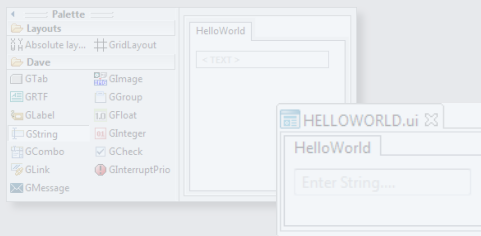
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

8



Free of charge

9

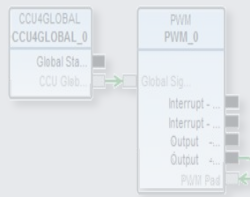
DAVE™

Support 3rd Party Tools



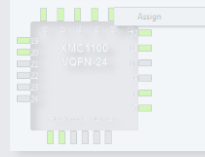
- › Interface to third party and commercial tools
- › XMC Lib can be used independent of DAVE™ or DAVE™ APPs with 3rd party tools
- › Easy import of DAVE™ generated code to 3rd party tools





Component based programming

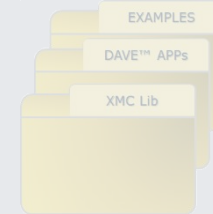
1



General Events and Outputs	
General Configurations	
Select the timer	CCUR Timer
☐ Start Timer After Initialization ☐ Enable Single Shot Mode	
PWM Configurations	
Desired Resolution [nsec]	32
Actual Resolution [nsec]	31.25
PWM Frequency [Hz]	1300
Duty Cycle [%]	50

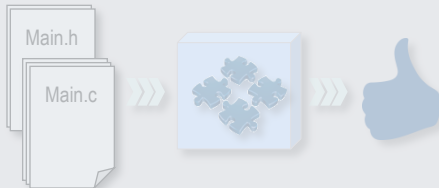
GUI based configuration

2



Code repository

3



Hardware resource manager

4

General Events and Outputs	
General Configurations	
Select the timer	CCUR Timer
☐ Start Timer After Initialization ☐ Enable Single Shot Mode	
PWM Configurations	
Desired Resolution [nsec]	32
Actual Resolution [nsec]	31.25
PWM Frequency [Hz]	1300
Duty Cycle [%]	50

```

1 // PWM configuration
2 #include "pwm_conf.h"
3 #include "pwm_conf.h"
4 #include "pwm_conf.h"
5 #include "pwm_conf.h"
6 #include "pwm_conf.h"
7 #include "pwm_conf.h"
8 #include "pwm_conf.h"
9 #include "pwm_conf.h"
10 #include "pwm_conf.h"
11 #include "pwm_conf.h"
12 #include "pwm_conf.h"
13 #include "pwm_conf.h"
14 #include "pwm_conf.h"
15 #include "pwm_conf.h"
16 #include "pwm_conf.h"
17 #include "pwm_conf.h"
18 #include "pwm_conf.h"
19 #include "pwm_conf.h"
20 #include "pwm_conf.h"
21 #include "pwm_conf.h"
22 #include "pwm_conf.h"
23 #include "pwm_conf.h"
24 #include "pwm_conf.h"
25 #include "pwm_conf.h"
26 #include "pwm_conf.h"
27 #include "pwm_conf.h"
28 #include "pwm_conf.h"
29 #include "pwm_conf.h"
30 #include "pwm_conf.h"
31 #include "pwm_conf.h"
32 #include "pwm_conf.h"
33 #include "pwm_conf.h"
34 #include "pwm_conf.h"
35 #include "pwm_conf.h"
36 #include "pwm_conf.h"
37 #include "pwm_conf.h"
38 #include "pwm_conf.h"
39 #include "pwm_conf.h"
40 #include "pwm_conf.h"
41 #include "pwm_conf.h"
42 #include "pwm_conf.h"
43 #include "pwm_conf.h"
44 #include "pwm_conf.h"
45 #include "pwm_conf.h"
46 #include "pwm_conf.h"
47 #include "pwm_conf.h"
48 #include "pwm_conf.h"
49 #include "pwm_conf.h"
50 #include "pwm_conf.h"
51 #include "pwm_conf.h"
52 #include "pwm_conf.h"
53 #include "pwm_conf.h"
54 #include "pwm_conf.h"
55 #include "pwm_conf.h"
56 #include "pwm_conf.h"
57 #include "pwm_conf.h"
58 #include "pwm_conf.h"
59 #include "pwm_conf.h"
60 #include "pwm_conf.h"
61 #include "pwm_conf.h"
62 #include "pwm_conf.h"
63 #include "pwm_conf.h"
64 #include "pwm_conf.h"
65 #include "pwm_conf.h"
66 #include "pwm_conf.h"
67 #include "pwm_conf.h"
68 #include "pwm_conf.h"
69 #include "pwm_conf.h"
70 #include "pwm_conf.h"
71 #include "pwm_conf.h"
72 #include "pwm_conf.h"
73 #include "pwm_conf.h"
74 #include "pwm_conf.h"
75 #include "pwm_conf.h"
76 #include "pwm_conf.h"
77 #include "pwm_conf.h"
78 #include "pwm_conf.h"
79 #include "pwm_conf.h"
80 #include "pwm_conf.h"
81 #include "pwm_conf.h"
82 #include "pwm_conf.h"
83 #include "pwm_conf.h"
84 #include "pwm_conf.h"
85 #include "pwm_conf.h"
86 #include "pwm_conf.h"
87 #include "pwm_conf.h"
88 #include "pwm_conf.h"
89 #include "pwm_conf.h"
90 #include "pwm_conf.h"
91 #include "pwm_conf.h"
92 #include "pwm_conf.h"
93 #include "pwm_conf.h"
94 #include "pwm_conf.h"
95 #include "pwm_conf.h"
96 #include "pwm_conf.h"
97 #include "pwm_conf.h"
98 #include "pwm_conf.h"
99 #include "pwm_conf.h"
100 #include "pwm_conf.h"

```

Code generation

5

- Released for
- Altium Limited
 - Atollic
 - ARM/KEIL
 - IAR Systems
 - Rowley Associates

TASKING

atollic

KEIL
Tools by ARM

IAR
SYSTEMS



Support 3rd party tools

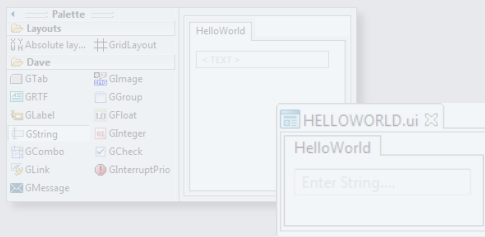
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

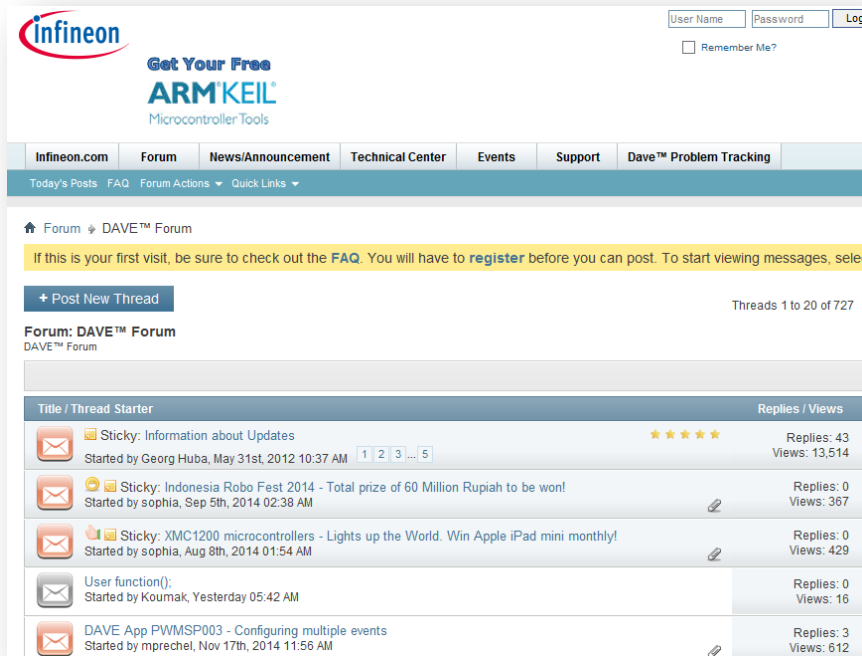
8



Free of charge

9

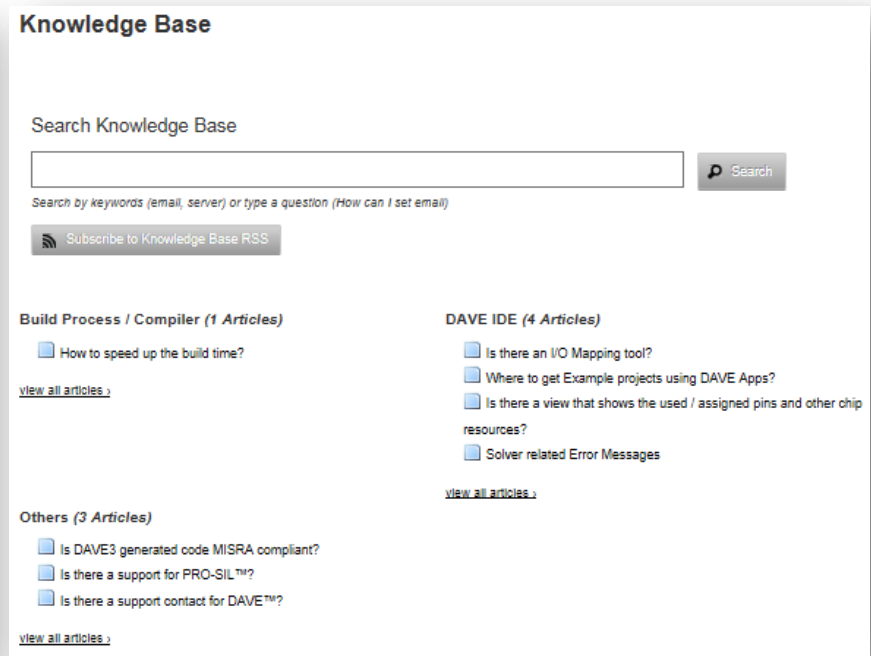
- › Easy access to DAVE™ technical support, downloads and information updates



The screenshot shows the Infineon DAVE™ Forum interface. At the top, there's a navigation bar with links to Infineon.com, Forum, News/Announcement, Technical Center, Events, Support, and DAVE™ Problem Tracking. Below this, a banner for 'Get Your Free ARM KEIL Microcontroller Tools' is visible. The main content area is titled 'Forum: DAVE™ Forum' and contains a list of threads. The threads are displayed in a table with columns for Title / Thread Starter, Replies, and Views. The first thread is 'Sticky: Information about Updates' by Georg Huba, dated May 31st, 2012, with 43 replies and 13,514 views. The second thread is 'Sticky: Indonesia Robo Fest 2014 - Total prize of 60 Million Rupiah to be won!' by sophia, dated Sep 5th, 2014, with 0 replies and 367 views. The third thread is 'Sticky: XMC1200 microcontrollers - Lights up the World. Win Apple iPad mini monthly!' by sophia, dated Aug 8th, 2014, with 0 replies and 429 views. The fourth thread is 'User function()' by Koumak, dated Yesterday 05:42 AM, with 0 replies and 16 views. The fifth thread is 'DAVE App PWMSP003 - Configuring multiple events' by mprechel, dated Nov 17th, 2014, with 3 replies and 612 views.

Title / Thread Starter	Replies / Views
Sticky: Information about Updates Started by Georg Huba, May 31st, 2012 10:37 AM	Replies: 43 Views: 13,514
Sticky: Indonesia Robo Fest 2014 - Total prize of 60 Million Rupiah to be won! Started by sophia, Sep 5th, 2014 02:38 AM	Replies: 0 Views: 367
Sticky: XMC1200 microcontrollers - Lights up the World. Win Apple iPad mini monthly! Started by sophia, Aug 8th, 2014 01:54 AM	Replies: 0 Views: 429
User function() Started by Koumak, Yesterday 05:42 AM	Replies: 0 Views: 16
DAVE App PWMSP003 - Configuring multiple events Started by mprechel, Nov 17th, 2014 11:56 AM	Replies: 3 Views: 612

DAVE™ Forum



The screenshot shows the Infineon Knowledge Base interface. At the top, there's a search bar labeled 'Search Knowledge Base' with a search button. Below the search bar, there's a link to 'Subscribe to Knowledge Base RSS'. The main content area is divided into two columns. The left column is titled 'Build Process / Compiler (1 Articles)' and contains a link to 'How to speed up the build time?'. The right column is titled 'DAVE IDE (4 Articles)' and contains links to 'Is there an I/O Mapping tool?', 'Where to get Example projects using DAVE Apps?', 'Is there a view that shows the used / assigned pins and other chip resources?', and 'Solver related Error Messages'. Below these columns, there's a section titled 'Others (3 Articles)' with links to 'Is DAVE3 generated code MISRA compliant?', 'Is there a support for PRO-SIL™?', and 'Is there a support contact for DAVE™?'.

Build Process / Compiler (1 Articles)

- How to speed up the build time?

[view all articles](#)

DAVE IDE (4 Articles)

- Is there an I/O Mapping tool?
- Where to get Example projects using DAVE Apps?
- Is there a view that shows the used / assigned pins and other chip resources?
- Solver related Error Messages

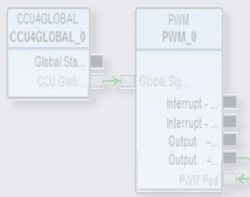
[view all articles](#)

Others (3 Articles)

- Is DAVE3 generated code MISRA compliant?
- Is there a support for PRO-SIL™?
- Is there a support contact for DAVE™?

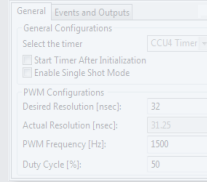
[view all articles](#)

Knowledge Base



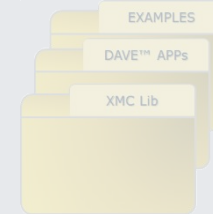
Component based programming

1



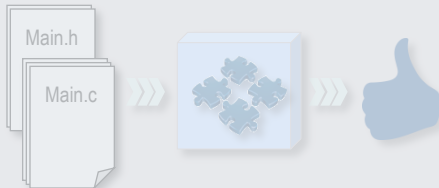
GUI based configuration

2



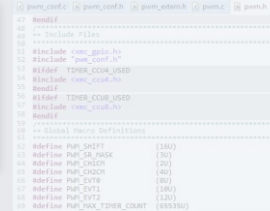
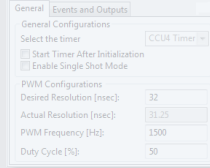
Code repository

3



Hardware resource manager

4



Code generation

5

- Released for
- Altium Limited
 - Atollic
 - ARM/KEIL
 - IAR Systems
 - Rowley Associates

TASKING

a atollic

KEIL
Tools by ARMIAR
SYSTEMS

Support 3rd party tools

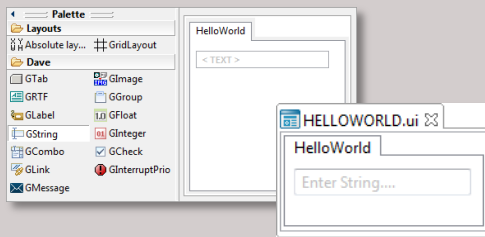
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

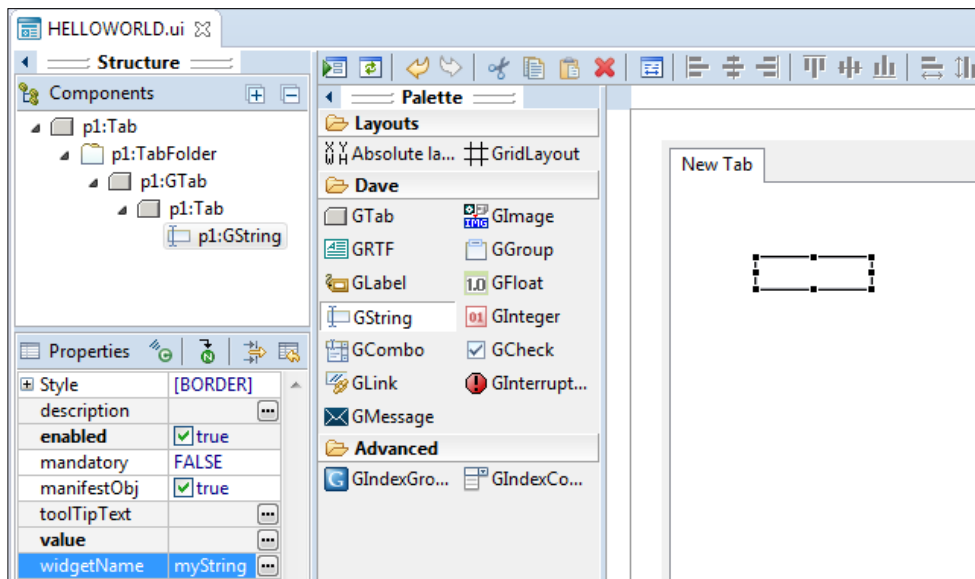
8



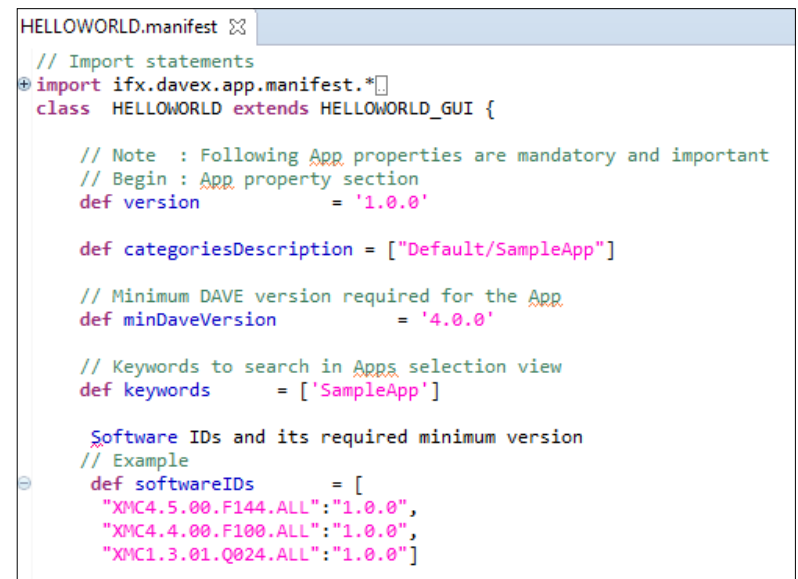
Free of charge

9

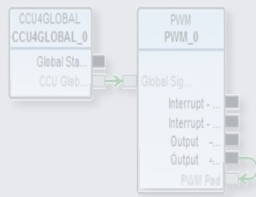
- › Develop new DAVE™ APPs or modify DAVE™ APPs
- › GUI designer with drag and drop functionality
- › Professional software editor with syntax highlights and completion functions for efficient coding
- › Debugging features



GUI designer

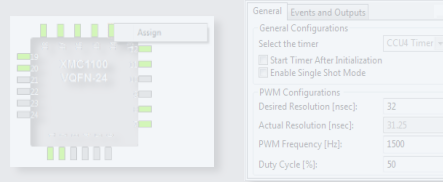


Software editor



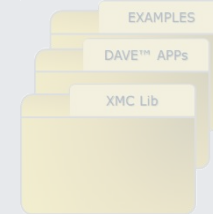
Component based programming

1



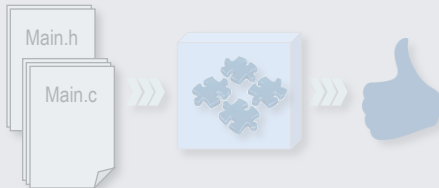
GUI based configuration

2



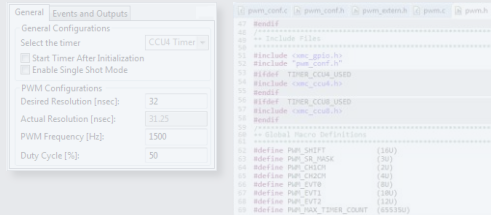
Code repository

3



Hardware resource manager

4



Code generation

5



Support 3rd party tools

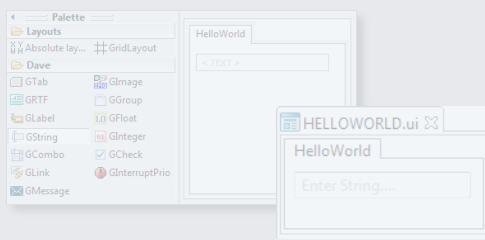
6

- FAQ
- FORUM
- Knowledge Base
- eMAIL: dave@infineon.com



Expert support

7



DAVE™ SDK

8



Free of charge

9

DAVE™

Free of charge



- › Download DAVE™ for free

www.infineon.com/DAVE

- › Download package includes DAVE™ v4.0.0 and DAVE™ SDK v4.0.0



Part of your life. Part of tomorrow.

