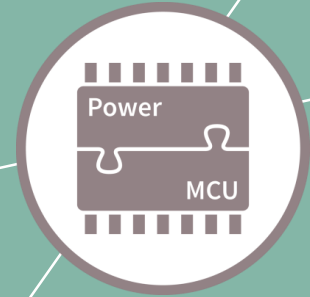


Evaluation Board and Evaluation Kit Getting Started for MOTIX™ TLE986x/7x

Toolchain Setup for:

- › TLE9869_EVALKIT
- › TLE9879_EVALKIT
- › TLE986x_EVALB_JLINK
- › TLE987x_EVALB_TQFP
- › TLE987x_EVALB_VQFN



October 2021



Table of contents

1	Toolchain installation for	3
	Keil μ Vision5	4
	IAR Embedded Workbench	5
2	Getting Started for	6
	Keil μ Vision5	7
	IAR Embedded Workbench	8
3	Evaluation board and kit overview	9
4	Product information links	10

Table of contents

1	Toolchain installation for	3
	Keil μ Vision5	4
	IAR Embedded Workbench	5
2	Getting Started for	6
	Keil μ Vision5	7
	IAR Embedded Workbench	8
3	Evaluation board and kit overview	9
4	Product information links	10

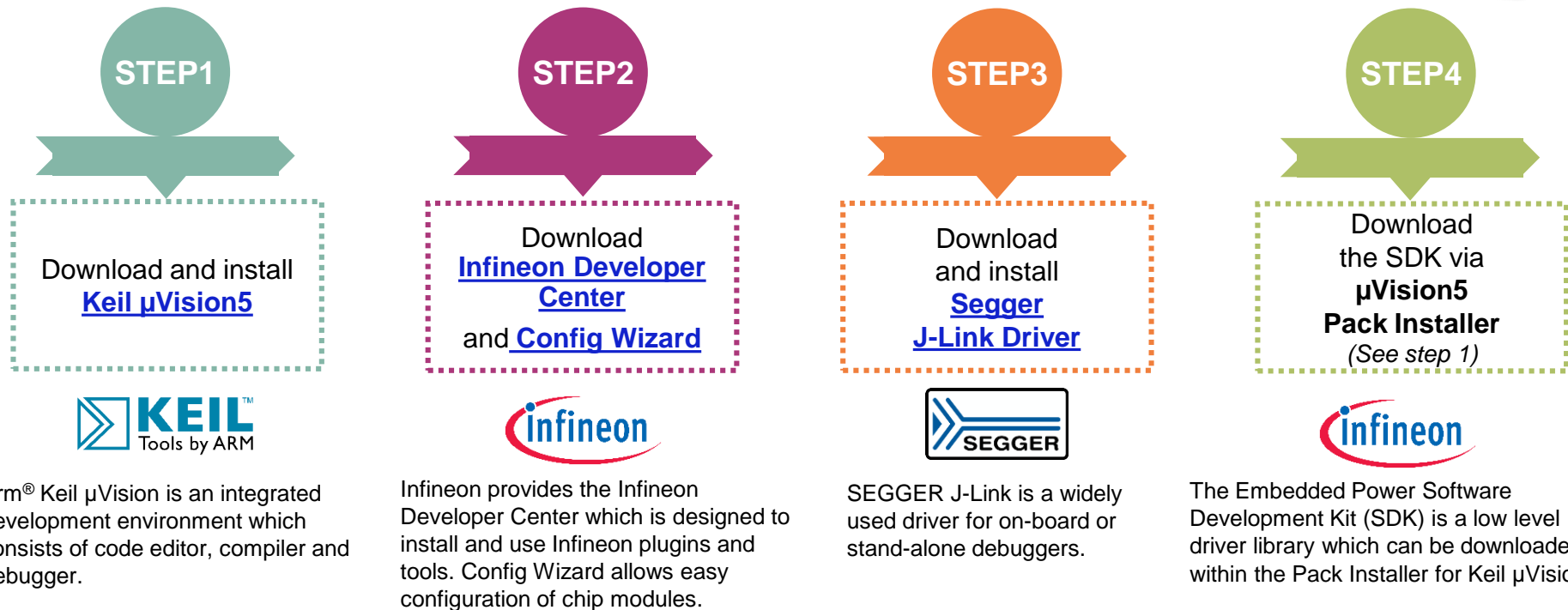
Toolchain Installation for Keil μ Vision5

General Overview



The information in steps 1-3 on how to install these tools can also be found as a video tutorial [here](#)

MOTIX™ Embedded Power ICs are supported by a complete development toolchain provided by Infineon and third party vendors.



Toolchain installation for Keil μ Vision5

Step 1: Keil μ Vision5

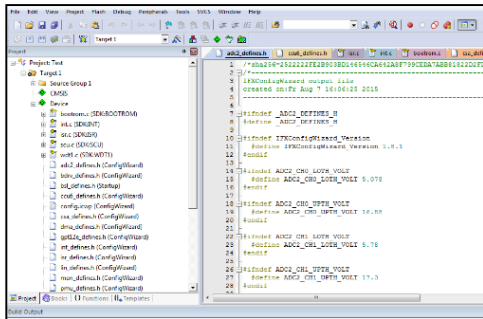


Keil μ Vision5



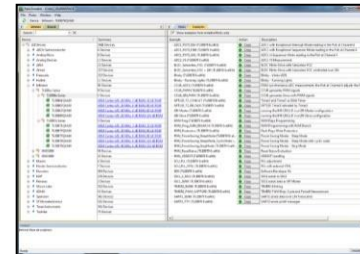
➤ Download from: <https://www.keil.com/demo/eval/arm.htm>

- Software development environment for Arm[®] Cortex-M[®] based microcontroller devices
- Code editor, online debugger and Arm C/C++ compiler



Pack Installer

- For downloading, installing, updating and managing the different software packages
- Quick access to example projects that can be used as a reference for own development



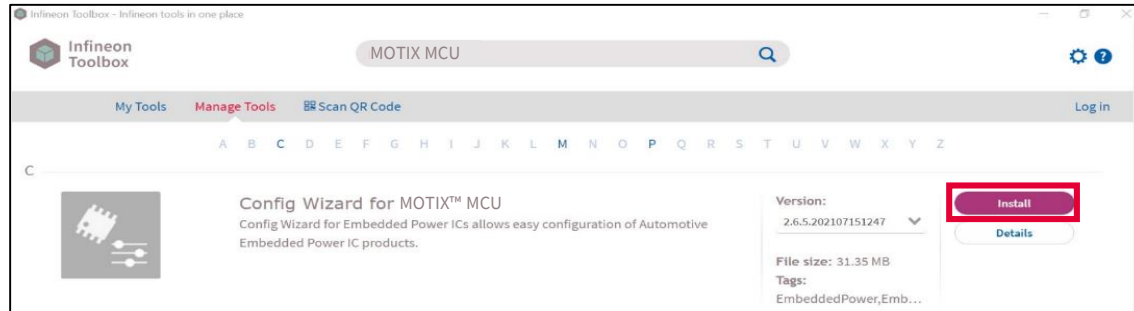
Toolchain installation for Keil μ Vision5

Step 2: Infineon Developer Center and Config Wizard (1)



Infineon Developer Center & Config Wizard for MOTIX™ MCU

- › Install the [Infineon Developer Center](#) (*) and start the tool
- › Within the Infineon Developer Center:
 1. Select the tab *Manage tools*, search for *Config Wizard for MOTIX™ MCU* and click on *Install*



(*) For more information about the Infineon Developer Center installation, please refer to the [Installation Manual](#).

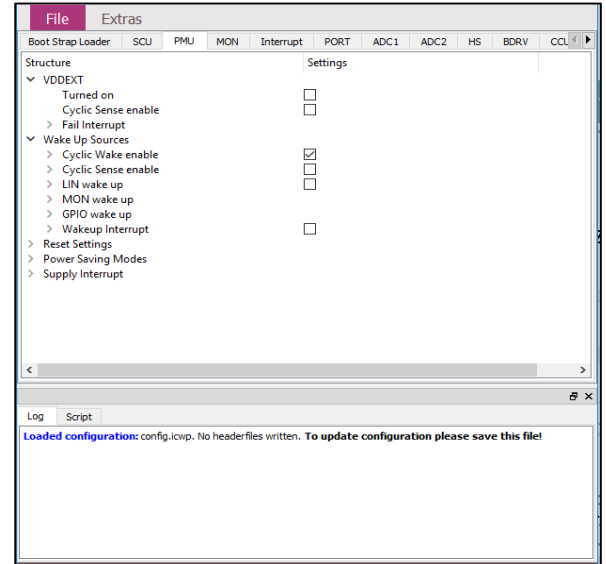
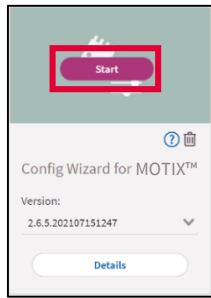
Toolchain installation for Keil μ Vision5

Step 2: Infineon Developer Center and Config Wizard (2)



Infineon Developer Center & Config Wizard for MOTIX™ MCU

2. Start the tool *Config Wizard for MOTIX™ MCU* once to trigger the integration into Keil μ Vision5



3. Close Config Wizard and the Infineon Developer Center

- > Config Wizard for MOTIX™ MCU enables the easy configuration of Automotive Embedded Power IC products.

Toolchain installation for Keil μ Vision5

Step 3: Segger J-Link Driver



Segger J-Link Driver

- > Driver for 'on-board' or 'stand-alone' debugger
- > Install driver from:
https://www.segger.com/downloads/jlink/JLink_Windows.exe



Toolchain installation for Keil μ Vision5

Step 4: Pack File



Pack File TLE986x and TLE987x for Keil μ Vision5

1. From Keil μ Vision5, open the *Pack Installer*
2. On the left side, select *Infineon*
3. On the right side, install the pack *Infineon::TLE987x_DFP*

- > Includes SDK (Software Development Kit) and Example code
- > Device support for flashing/erasing
- > SFR description for register debugging
- > Device description for Config Wizard

The screenshot illustrates the steps in the Keil Pack Installer:

- Step 1:** The Pack Installer window is open, showing the instruction: "Install or update Software Packs that contain Software Components".
- Step 2:** The "Devices" tab is selected. The tree view shows the "Infineon" folder expanded, with the "TLE987x Series" folder selected. The "TLE9877QXA40" device is highlighted.
- Step 3:** The "Packs" tab is selected. The "Infineon::TLE987x_DFP" pack is selected, and the "Install" button is highlighted.

Pack	Action	Description
Device Specific	1 Pack	TLE9877QXA40 selected
Infineon::TLE987x_DFP	Install	Infineon TLE987x Series Device Support
-1.3.2	Unpack	Infineon TLE987x Series Device Support
Previous		Infineon::TLE987x_DFP - Previous Pack Versions
Generic	33 Packs	

Table of contents

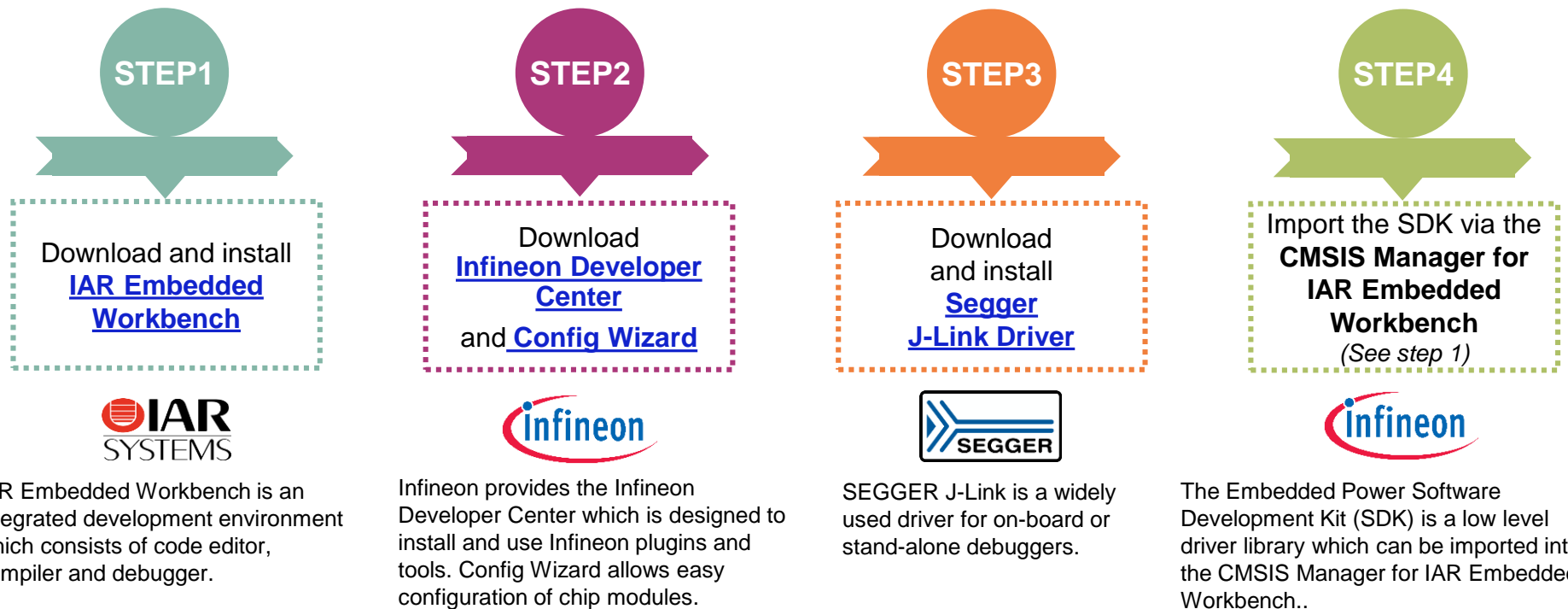
1	Toolchain installation for	3
	Keil μ Vision5	4
	IAR Embedded Workbench	5
2	Getting Started for	6
	Keil μ Vision5	7
	IAR Embedded Workbench	8
3	Evaluation board and kit overview	9
4	Product information links	10

Toolchain Installation for IAR Embedded Workbench

General Overview



MOTIX™ Embedded Power ICs are supported by a complete development toolchain provided by Infineon and third party vendors.



Toolchain Installation for IAR Embedded Workbench

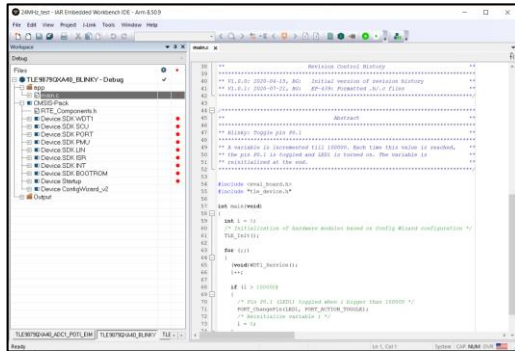
Step 1: IAR Embedded Workbench for ARM



IAR Embedded Workbench for ARM

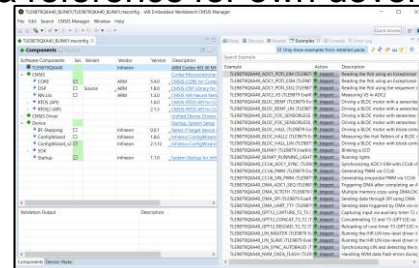


- Download from: <https://www.iar.com/products/architectures/arm/iar-embedded-workbench-for-arm/>
- Software development environment for Arm® Cortex-M® based microcontroller devices
- Code editor, online debugger and Arm C/C++ compiler



CMSIS Manager

- For importing, updating and managing the different software packages
- Quick access to example projects that can be used as a reference for own development



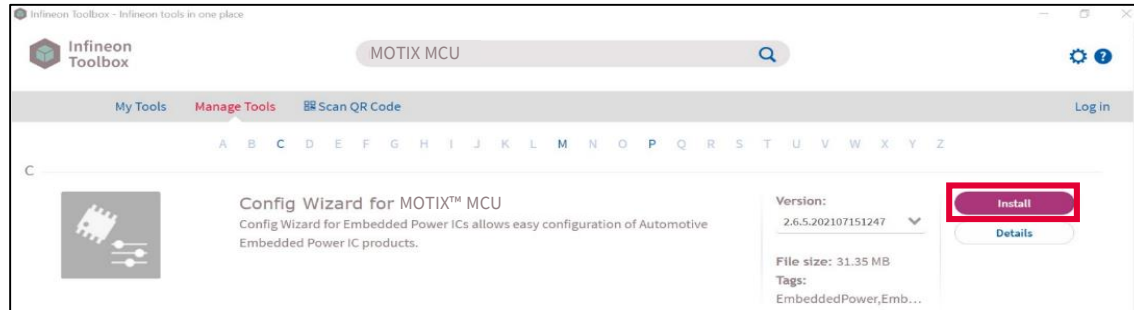
Toolchain installation for IAR Embedded Workbench

Step 2: Infineon Developer Center and Config Wizard (1)



Infineon Developer Center & Config Wizard for MOTIX™ MCU

- › Install the [Infineon Developer Center](#) (*) and start the tool
- › Within the Infineon Developer Center:
 1. Select the tab *Manage tools*, search for *Config Wizard for MOTIX™ MCU* and click on *Install*



(*) For more information about the Infineon Developer Center installation, please refer to the [Installation Manual](#).

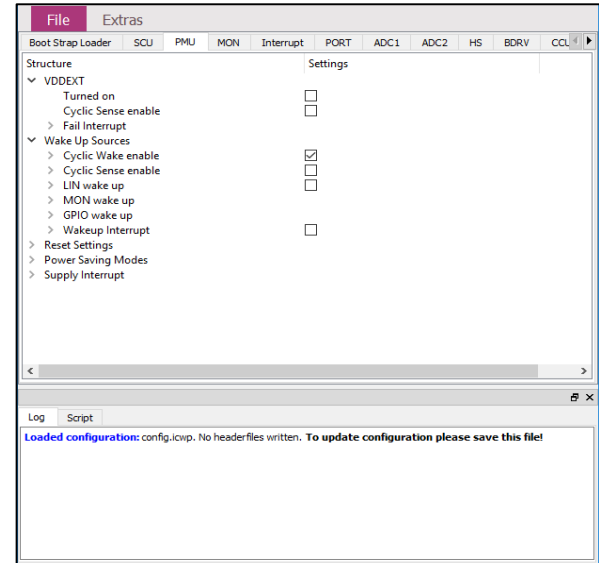
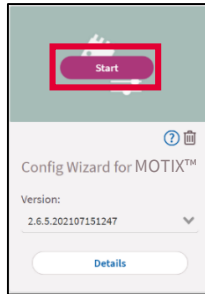
Toolchain Installation for IAR Embedded Workbench

Step 2: Infineon Developer Center and Config Wizard (2)



Infineon Developer Center & Config Wizard for MOTIX™ MCU

2. Start the tool *Config Wizard for MOTIX™ MCU* by clicking on Start



3. Close Config Wizard and the Infineon Developer Center
- > Config Wizard for MOTIX™ MCU enables the easy configuration of Automotive Embedded Power IC products.

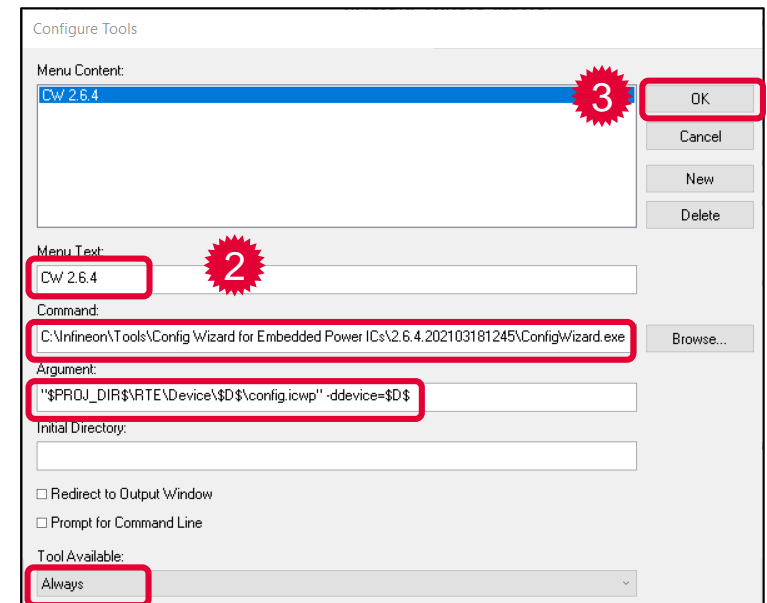
Toolchain Installation for IAR Embedded Workbench

Step 2: Infineon Developer Center and Config Wizard (3)



Infineon Developer Center & Config Wizard for MOTIX™ MCU

- › Unlike for Keil μ Vision5, the Config Wizard is not integrated automatically into the IAR Embedded Workbench.
- 1. Select the tab *Tools > Configure Tools...*
- 2. Define a name and set up the following:
 - › Command: Path to the tool exe
 - › Argument (from Config Wizard v2.6.4):
`"$PROJ_DIR$\RTE\Device\${D$\config.icwp" -dddevice=${D$`
 - › Tool Available: *Always*
- 3. Click on *OK*



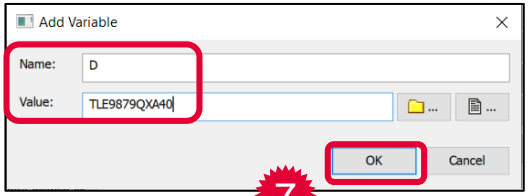
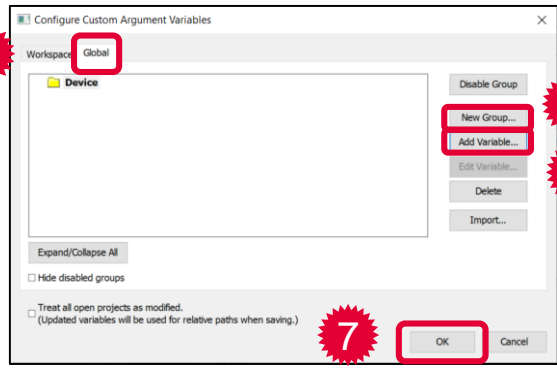
Toolchain Installation for IAR Embedded Workbench

Step 2: Infineon Developer Center and Config Wizard (4)



Infineon Developer Center & Config Wizard for MOTIX™ MCU

4. Click on *Tools > Configure Custom Argument Variables...* to define the variable `D` in the argument phrase
5. Select the tab *Global*, then define a name for the *New Group...* (here, *Device*)
6. Click on *Add Variable...* and define its name (here, *D*) and its value (here, *TLE9879QXA40*)
7. Click twice on *OK* to save your settings
8. Config Wizard is now fully accessible



Toolchain Installation for IAR Embedded Workbench

Step 3: Segger J-Link Driver



Segger J-Link Driver

- > Driver for 'on-board' or 'stand-alone' debugger
- > Install driver from:
https://www.segger.com/downloads/jlink/JLink_Windows.exe



Toolchain Installation for IAR Embedded Workbench

Step 4: Pack File



Pack File TLE987x for IAR Embedded Workbench

1. Open the *CMSIS Manager*
2. On the right side, select *Import Existing Packs...* and import the pack from your local disk (all packs are available [here](#), under Infineon)
3. The tab *Devices* shows the supported devices
4. The tab *Examples* shows the examples

- > Includes SDK and Example code
- > Device support for flashing/erasing
- > SFR description for register debugging
- > Device description for Config Wizard

The image shows four screenshots of the CMSIS Manager interface. Screenshot 1 shows the 'CMSIS-Pack Manager' dialog box with 'Open CMSIS-Pack Manager' button highlighted. Screenshot 2 shows the 'Import Existing Packs...' button highlighted. Screenshot 3 shows the 'Examples' tab with a table of examples. Screenshot 4 shows the 'Devices' tab with a list of devices.

Example	Action	Description
TLE9879QXA40_ADC1_POTI_EIM (TLE987)	Import	Reading the Poti using an Except...
TLE9879QXA40_ADC1_POTI_ESM (TLE987)	Import	Reading the Poti using an Except...
TLE9879QXA40_ADC1_POTI_SEQ (TLE987)	Import	Reading the Poti using the sequ...
TLE9879QXA40_ADC2_VS (TLE9879 EvalK)	Import	Measuring VS in ADC2
TLE9879QXA40_BLDC_BEMF (TLE9879 Ev)	Import	Driving a BLDC motor with a sens...
TLE9879QXA40_BLDC_BEMF_LIN (TLE987)	Import	Driving a BLDC motor with a sens...
TLE9879QXA40_BLDC_FOC_SENSORLESS	Import	Driving a BLDC motor with senso...
TLE9879QXA40_BLDC_FOC_SENSORLESS	Import	Driving a BLDC motor with senso...
TLE9879QXA40_BLDC_HALL (TLE9879 Ev)	Import	Driving a BLDC motor with block...
TLE9879QXA40_BLDC_HALL2 (TLE9879 E)	Import	Measuring the Hall Pattern of a B...
TLE9879QXA40_BLDC_HALL_LIN (TLE987)	Import	Driving a BLDC motor with block...
TLE9879QXA40_BLINKY (TLE9879 EvalK)	Import	Blinking a LED
TLE9879QXA40_BLINKY_RUNNING_LIGHT	Import	Running lights
TLE9879QXA40_CCUE_ADC1_SYNC (TLE9)	Import	Synchronizing ADC1-EIM with CC...
TLE9879QXA40_CCUE_PWM (TLE9879 Ev)	Import	Generating PWM via CCUE
TLE9879QXA40_CCUE_SIN_PWM (TLE987)	Import	Generating sinusoidal PWM vi...

Device	Summary
All Devices	81 Devices
ARM	34 Devices
Infineon	47 Devices
TLE987x Series	47 Devices
TLE984x Series	7 Devices
TLE985x Series	7 Devices
TLE986x Series	7 Devices
TLE987x Series	14 Devices
TLE9871QXA20	ARM Cortex-M3 24 MHz, 2 KB RAM, 36 KB ROM
TLE9872-QXA40	ARM Cortex-M3 40 MHz, 7 KB RAM, 256 KB ROM
TLE9872QTW40	ARM Cortex-M3 40 MHz, 7 KB RAM, 256 KB ROM
TLE9873QXA40	ARM Cortex-M3 40 MHz, 7 KB RAM, 251 KB ROM
TLE9873QXW40	ARM Cortex-M3 40 MHz, 2 KB RAM, 48 KB ROM
TLE9877QTW40	ARM Cortex-M3 40 MHz, 5 KB RAM, 64 KB ROM
TLE9877QXA20	ARM Cortex-M3 24 MHz, 5 KB RAM, 64 KB ROM
TLE9877QXA40	ARM Cortex-M3 40 MHz, 5 KB RAM, 64 KB ROM
TLE9877QXW40	ARM Cortex-M3 40 MHz, 5 KB RAM, 64 KB ROM
TLE9879-2QXA40	ARM Cortex-M3 40 MHz, 5 KB RAM, 128 KB ROM
TLE9879QTW40	ARM Cortex-M3 40 MHz, 5 KB RAM, 128 KB ROM
TLE9879QXA20	ARM Cortex-M3 24 MHz, 5 KB RAM, 128 KB ROM
TLE9879QXA40	ARM Cortex-M3 40 MHz, 5 KB RAM, 128 KB ROM
TLE9879QXW40	ARM Cortex-M3 40 MHz, 5 KB RAM, 128 KB ROM

Table of contents

1	Toolchain installation for	3
	Keil μ Vision5	4
	IAR Embedded Workbench	5
2	Getting Started for	6
	Keil μ Vision5	7
	IAR Embedded Workbench	8
3	Evaluation board and kit overview	9
4	Product information links	10

The example shown in this section is also covered by a video tutorial [here](#)

Keil μ Vision5 Template

- > Create new project with Infineon SDK
- > Write code

Infineon Config Wizard

- > Initialize modules
- > Set up Timers
- > Set up GPIOs

J-Link Configuration

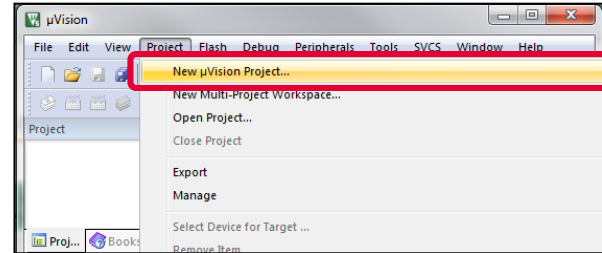
- > Connect device
- > Program flash
- > Use of debug window

Getting Started with Keil μ Vision5

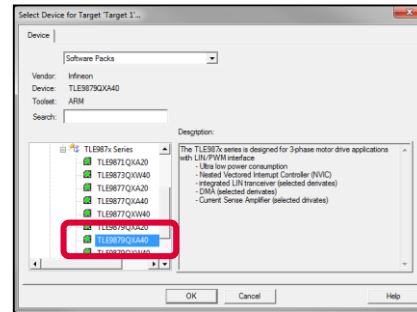
Step 1: Create New Project

1) Create New Project

- › Open Keil MDK
- › Select *Project > New μ Vision Project*
- › Name the project: (“TIMER2BLINK”)



- › Select the device
 - › e.g. TLE9879QXA40 for TLE9879_EVALKIT



Getting Started with Keil μ Vision5

Step 2: Configure the Run-Time Environment

2) Configure the Run-Time Environment

1. Expand *Device*

- › Check *ConfigWizard_v2*
- › Check *Startup*

➤ The selected cell background is **orange**

2. Click on *Resolve*

➤ The selected cell background is **green**

3. Click on *OK*

The image shows two screenshots of the 'Manage Run-Time Environment' dialog box in Keil μ Vision5. The top screenshot shows the 'Device' component expanded, with 'ConfigWizard_v2' and 'Startup' selected, highlighted with an orange background. A red circle with the number '1' is next to the 'Device' folder. A red circle with the number '2' is next to the 'Resolve' button. The bottom screenshot shows the same dialog box after clicking 'Resolve', with the 'ConfigWizard_v2' and 'Startup' cells highlighted in green. A red circle with the number '3' is next to the 'OK' button. A large grey arrow points from the 'Resolve' button in the top screenshot to the 'OK' button in the bottom screenshot.

Software Component	Sel.	Variant	Version	Description
CMISIS	<input checked="" type="checkbox"/>			Cortex-Microcontroller Software Interface Components
CMISIS Driver	<input checked="" type="checkbox"/>			Unified Device Drivers compliant to CMSIS-Driver Specifications
Compiler	<input checked="" type="checkbox"/>	ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
Device	<input checked="" type="checkbox"/>			Startup, System Setup
BF-Stepping	<input checked="" type="checkbox"/>		0.0.1	Select if target device is BF-Stepping
ConfigWizard	<input checked="" type="checkbox"/>		1.8.6	Infineon ConfigWizard Configuration File
ConfigWizard_v2	<input checked="" type="checkbox"/>		2.1.12	Infineon ConfigWizard Configuration File
Startup	<input checked="" type="checkbox"/>		1.1.0	System Startup for Infineon TLE987x device series
SDK	<input checked="" type="checkbox"/>			
File System	<input checked="" type="checkbox"/>	MDK-Plus	6.13.8	File Access on various storage devices
Graphics	<input checked="" type="checkbox"/>	MDK-Plus	6.10.8	User Interface on graphical LCD displays

Validation Output

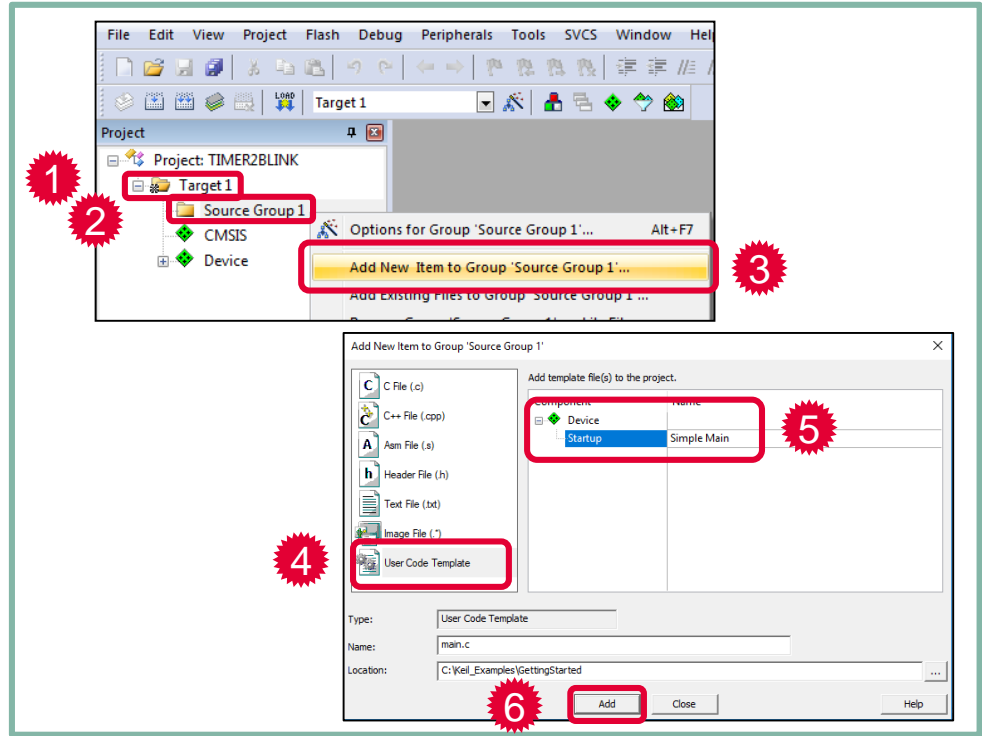
Infineon:Device:Startup	Description
require Device:SDK:PMU	Additional software components required
Infineon:Device:SDK:PMU	Select component from list
require Device:SDK:WDT1	Power Management Unit (SCU) driver for TLE987x
Infineon:Device:SDK:WDT1	Select component from list
require CMSIS:DFE:	Watchdog 1 (WDT1) driver for TLE987x
CMSIS:DFE:	Select component from list

Getting Started with Keil μ Vision5

Step 3: Use the main.c template

3) Use the main.c Template

1. Expand *Target 1*
2. Right click on *Source Group 1*
3. Choose *Add New Item to Group 'Source Group 1'*
4. Choose *User Code Template*
5. Expand *Device*
 - > Choose *Startup*
6. Click on *Add*

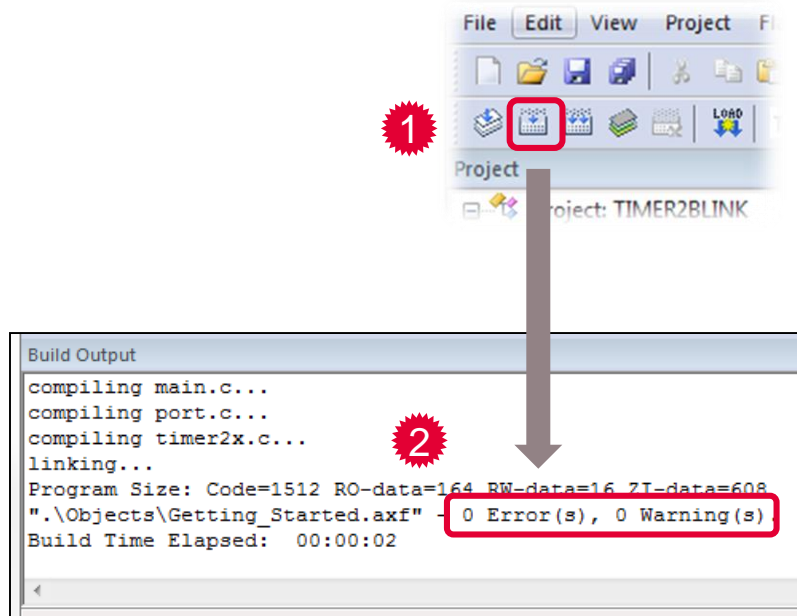


Getting Started with Keil μ Vision5

Step 4: Compile the Project

4) Compile the Project

1. Compile Project:
 - > Press the *Build* button or press *F7*
2. The Build Output window shows 0 Error(s) , 0 Warning(s)

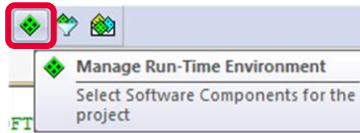


Getting Started with Keil μ Vision5

Step 5: Add Modules from the Run-Time Environment

5) Add Modules from the Run-Time Environment

1. Click on *Manage Run-Time Environment*



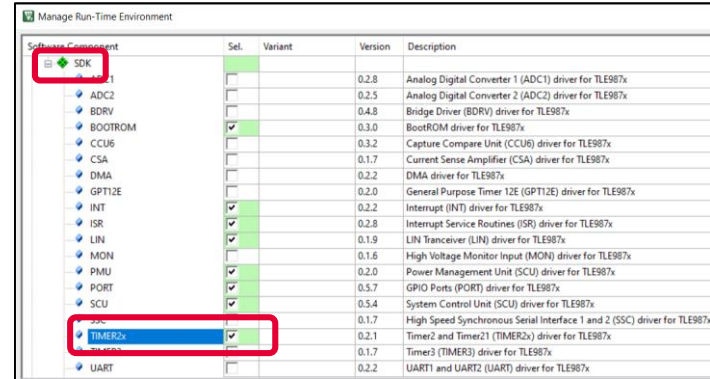
2. Expand *Device*, then the *SDK* section

3. Select *TIMER2x* and click *OK*

4. Add code to the main.c file:

- i. Include Libraries
- ii. Initialize Modules

2



3

```
49 /*****  
50 ** Includes  
51 *****/  
52 #include "tle_device.h"  
53  
54 #include "port.h"  
55 #include "timer2x.h"
```

i

```
82 int main(void)  
83 {  
84 /* Initialization of the hardware modules  
85 based on Config Wizard config */  
86 TLE_Init();
```

ii

Getting Started with Keil μ Vision5

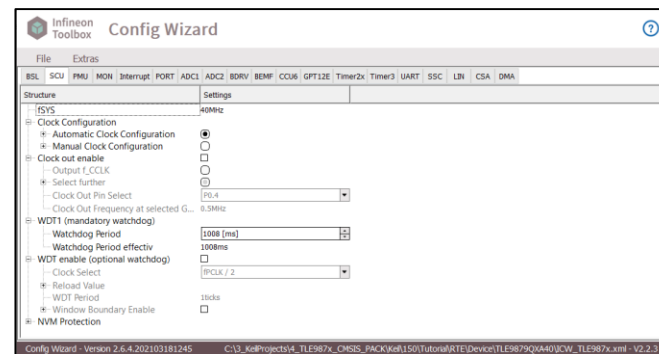
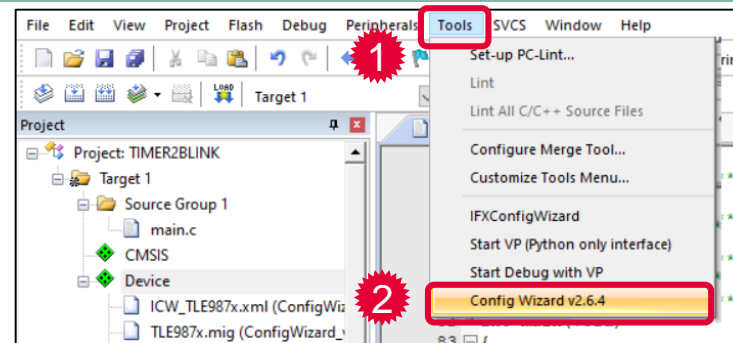
Step 6: Use the Config Wizard v2 (1)

6) Use the Config Wizard v2 (1)

1. Select the tab *Tools*

2. Open *Config Wizard v2*

3. Config Wizard opens in a separate window



Getting Started with Keil μ Vision5

Step 6: Use the Config Wizard v2 (2)

6) Use the Config Wizard v2 (2)

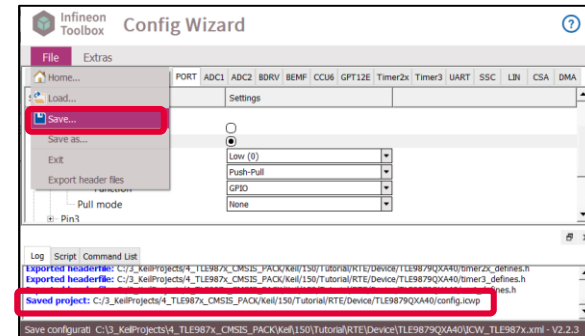
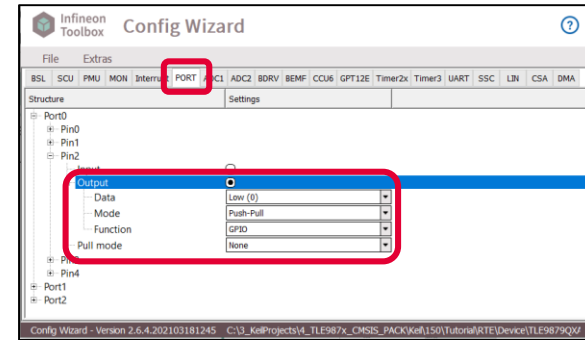
1. Select the tab *PORT*
2. Go to the *Port0 > Pin2* section
3. Configure pin to *Output* mode
4. Save with *File > Save*

1

2

3

4

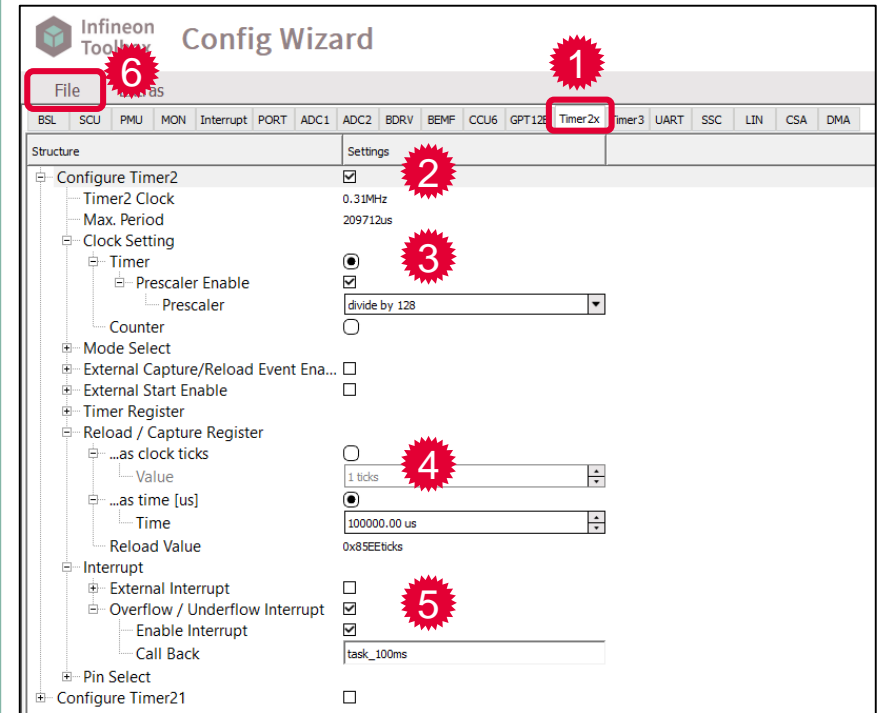


Getting Started with Keil μ Vision5

Step 6: Use the Config Wizard v2 (3)

6) Use the Config Wizard v2 (3)

1. Click on the tab *Timer2x*
2. Enable the checkbox *Configure Timer2*
3. In the section *Clock Setting*:
 - › Enable the prescaler and select *divide by 128*
4. In the section *Reload / Capture Register*:
 - › Select *...as time[us]* and enter *100000.00 us*
5. In the section *Interrupt*:
 - › Enable the checkboxes *Overflow / Underflow Interrupt* and *Enable Interrupt*
 - › Call Back Name: *task_100ms*
6. Save with *File > Save*



The screenshot shows the Infineon Config Wizard v2 interface. The top menu bar includes 'File', 'Tools', and 'Help'. The 'Timer2x' tab is selected. The 'Structure' tree on the left shows the configuration for 'Configure Timer2'. The 'Settings' panel on the right shows the following configuration:

- Configure Timer2
- Timer2 Clock: 0.31MHz
- Max. Period: 209712us
- Clock Setting
 - Timer
 - Prescaler Enable
 - Prescaler: divide by 128
 - Counter:
 - Mode Select
 - External Capture/Reload Event Enable
 - External Start Enable
 - Timer Register
 - Reload / Capture Register
 - ...as clock ticks
 - ...as time [us]
 - Time: 100000.00 us
 - Reload Value: 0x85EEticks
- Interrupt
 - External Interrupt
 - Overflow / Underflow Interrupt
 - Enable Interrupt
 - Call Back: task_100ms
 - Pin Select
- Configure Timer21

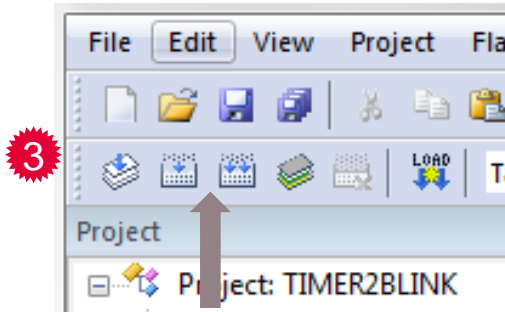
Getting Started with Keil μ Vision5

Step 7: Edit the File main.c

7) Edit the File main.c

In the file main.c in Keil MDK:

1. Start Timer2 before the “for (;;)” loop
2. Declare and define the function of the interrupt call back
 - › Use API function “PORT_ChangePin()”
3. Save and build (F7) the project



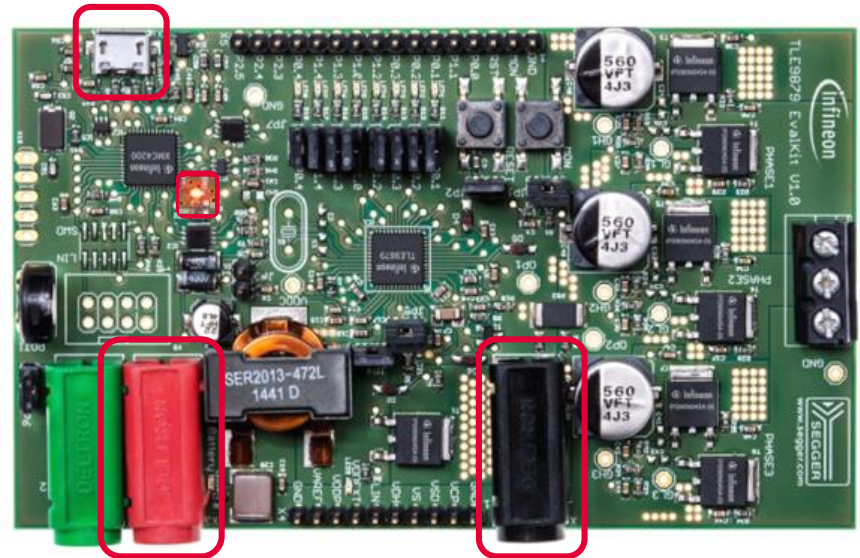
```
81  /*****  
82  ** Place your application code here  
83  *****/  
84  TIMER2_Start();  
85  /*****  
86  ** Main endless loop  
87  *****/  
88  for (;;)   
89  {
```

```
66  /*****  
67  ** Private Function Declaration  
68  *****/  
69  void task_100ms(void);
```

```
106 void task_100ms(void)  
107 {  
108     PORT_ChangePin(0x02, PORT_ACTION_TOGGLE);  
109 }
```

8) Power up the Evaluation Board

- › Connect micro USB cable
- › Supply board via banana jacks (VBAT, GND)
- › Debug LED lights up

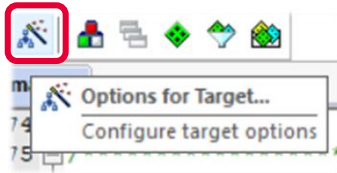


Getting Started with Keil μ Vision5

Step 9: Connect the Debugger

9) Connect the Debugger

1. Click on *Options for Target*

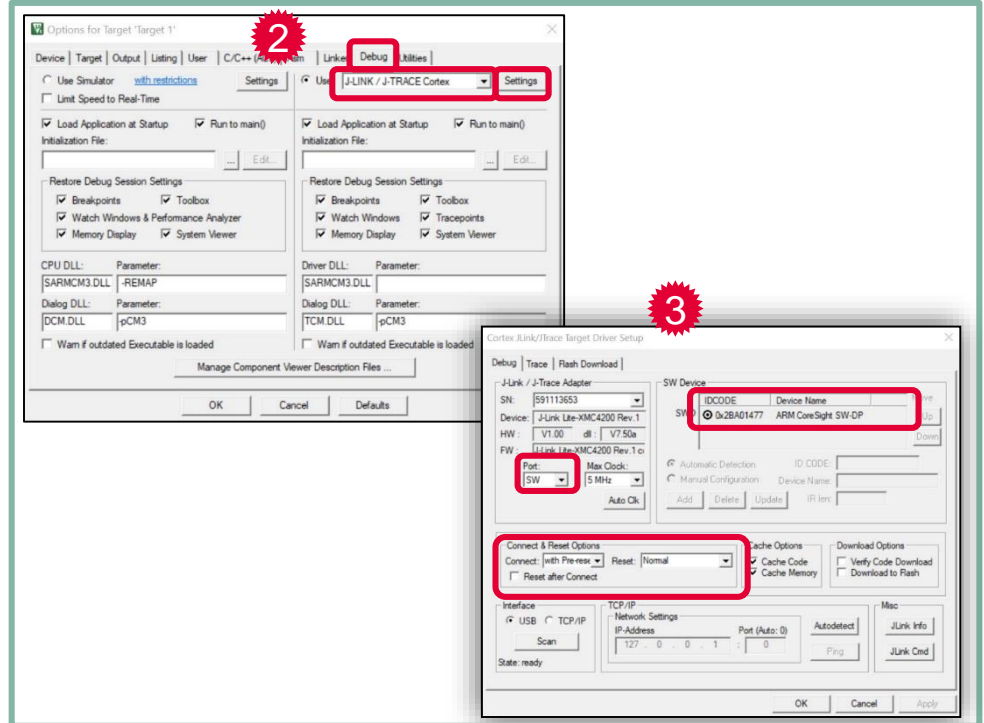


2. Click on the tab *Debug* and select *J-Link*

3. Click on *Settings* and set up the following:

- › Port: *SW*
- › Connect: *with Pre-Reset*
- › Reset: *Normal*

- SWD connection established when “IDCODE” is visible



Getting Started with Keil μ Vision5

Step 10: Download and Run Code

10) Download and Run Code

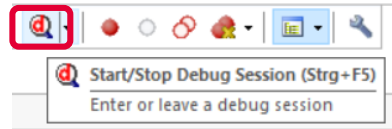
1. Load the code into the target or press *F8*
 - > *Flash Load finished* is shown in the *Build Output* window
2. Press the *Reset* button on the evaluation board
3. The LED on Port P0.2 lights up
4. The port toggles every 100 ms

The diagram illustrates the process of downloading and running code on a target device. It consists of four numbered steps:

1. A screenshot of the Keil IDE showing the 'Target 1' window with the 'Download (F8)' button highlighted by a red box and a red '1' in a starburst.
2. A screenshot of the 'Build Output' window showing the message 'Flash Load finished at 12:33:59' highlighted by a red box and a red '2' in a starburst.
3. A photograph of an evaluation board with a red box around the 'RST' button and a red '3' in a starburst.
4. A square wave timing diagram showing a period of 100ms, with a red '4' in a starburst below it.

11) Use the Runtime Debug

1. Click on *Start/Stop Debug Session*



2. Left click in the dark grey area left of the code to place a breakpoint

3. Click on *Run* or press *F5* to start the code execution

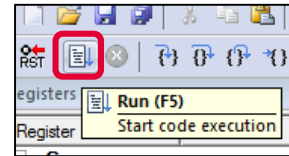
4. The code is executed and stops at the breakpoint

- › In this example, every time you click on Run, the LED P0.2 is toggled

2

```
146  | 146 | 146 | /***** Private Function Definitions *****/
147  | 147 | 147 | **
148  | 148 | 148 | *****/
149  | 149 | 149 | void task_100ms(void)
150  | 150 | 150 | {
151  | 151 | 151 |     PORT_ChangePin(0x02,PORT_ACTION_TOGGLE);
152  | 152 | 152 | }
153  | 153 | 153 |
```

3



4

```
146  | 146 | 146 | /***** Private Function Definitions *****/
147  | 147 | 147 | **
148  | 148 | 148 | *****/
149  | 149 | 149 | void task_100ms(void)
150  | 150 | 150 | {
151  | 151 | 151 |     PORT_ChangePin(0x02,PORT_ACTION_TOGGLE);
152  | 152 | 152 | }
153  | 153 | 153 |
```

Table of contents

1	Toolchain installation for	3
	Keil μ Vision5	4
	IAR Embedded Workbench	5
2	Getting Started for	6
	Keil μ Vision5	7
	IAR Embedded Workbench	8
3	Evaluation board and kit overview	9
4	Product information links	10

IAR Embedded Workbench Template

- > Create new project with Infineon SDK
- > Write code

Infineon Config Wizard

- > Initialize modules
- > Set up Timers
- > Set up GPIOs

J-Link Configuration

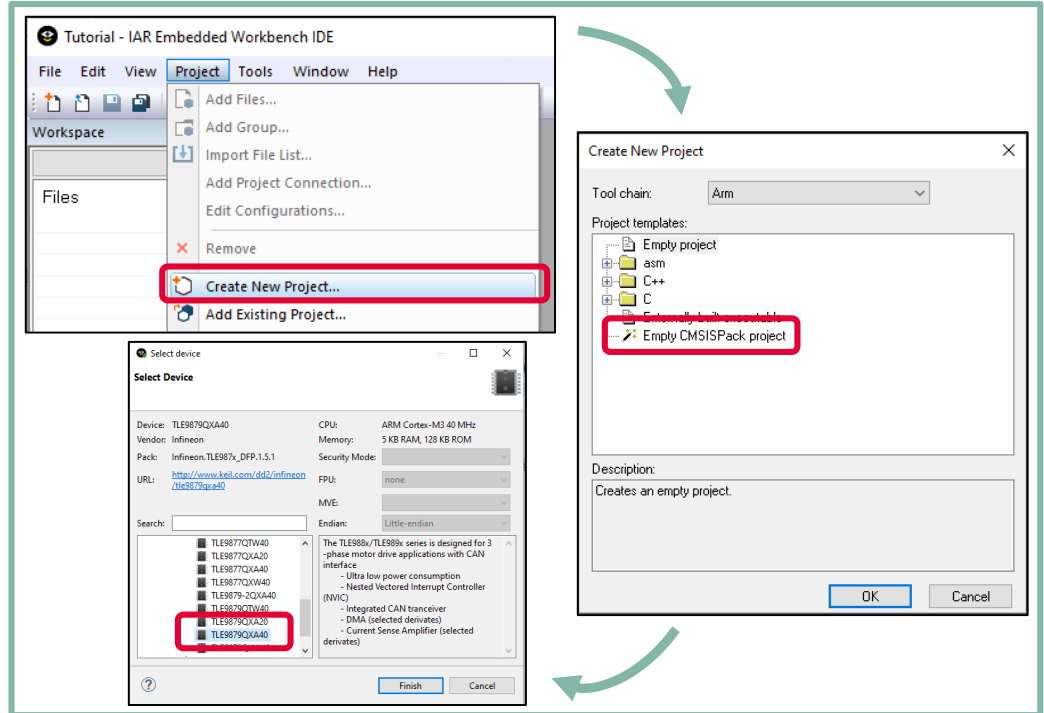
- > Connect device
- > Program flash
- > Use of debug window

Getting Started with IAR Embedded Workbench

Step 1: Create New Project

1) Create New Project

- › Open IAR Embedded Workbench
- › Select a workspace to contain your project
- › Select *Project > Create New Project...*
- › Select *Empty CMSISPack project*
- › Name the project: ("TIMER2BLINK")
- › Select the device
 - › e.g. TLE9879QXA40 for TLE9879_EVALKIT



Getting Started with IAR Embedded Workbench

Step 2: Configure the Run-Time Environment

2) Configure the Run-Time Environment

In the `TIMER2BLINK.rteconfig` file that opens in the CMSIS Manager:

1. Expand *Device*

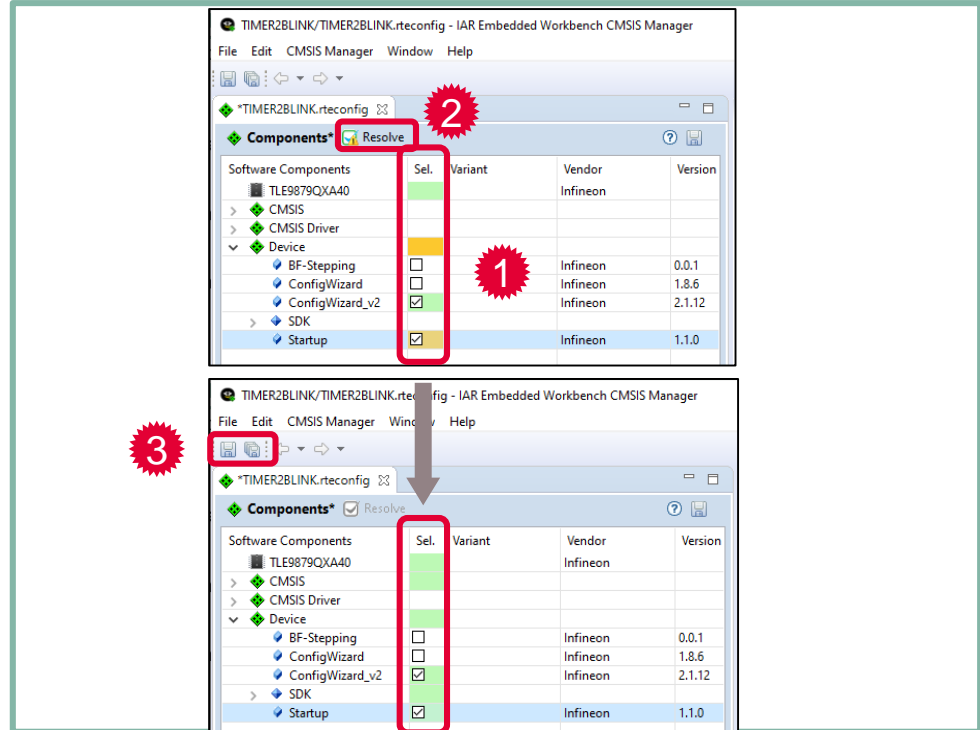
- › Check *ConfigWizard_v2*
- › Check *Startup*

➤ The selected cell background is **orange**

2. Click on *Resolve*

➤ The selected cell background is **green**

3. Save your configuration: the selected modules appear under the section CMSIS-Pack in your workspace



Getting Started with IAR Embedded Workbench

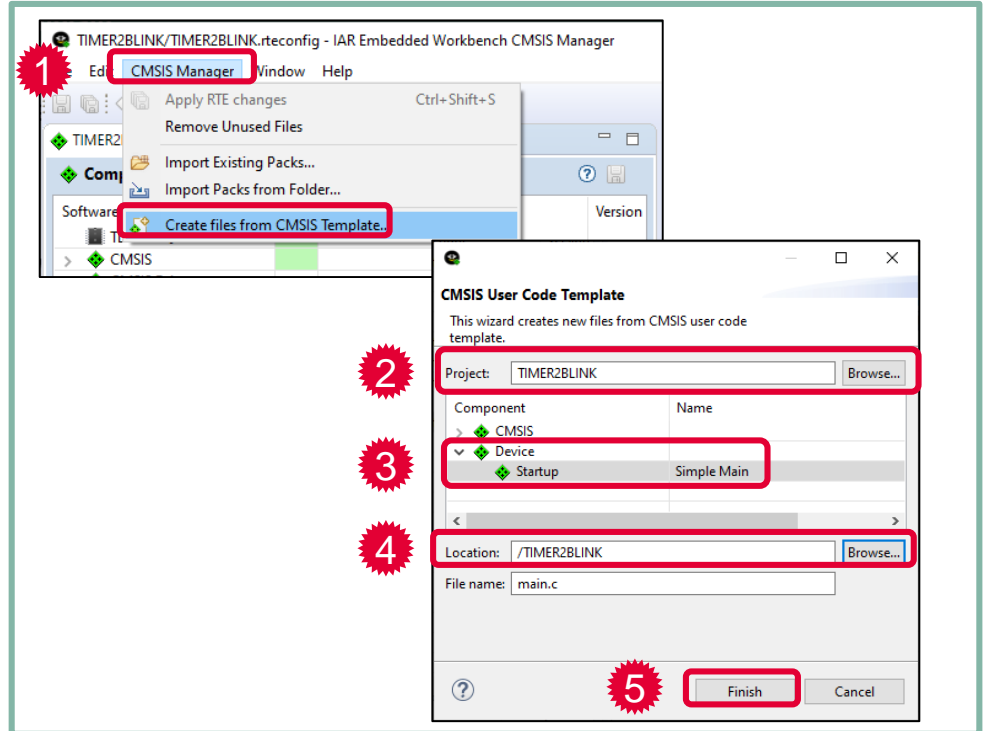
Step 3: Use the main.c template (1)

3) Use the main.c Template (1)

In the CMSIS Manager:

1. Open the menu *CMSIS Manager* and select *Create files from CMSIS Template...*
2. Select the project **TIMER2BLINK**
3. Expand *Device* and select *Startup/Simple Main*
4. Browse for the file location
5. Click on *Finish*

The file was added to your project but still does not appear in your workspace.



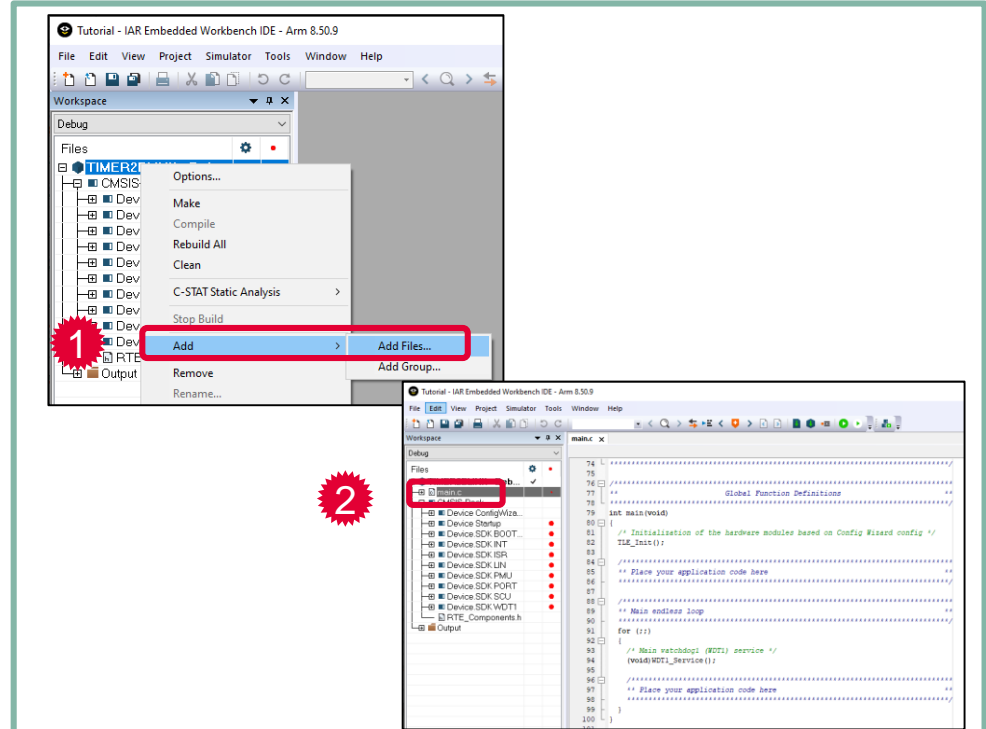
Getting Started with IAR Embedded Workbench

Step 3: Use the main.c template (2)

3) Use the main.c Template (2)

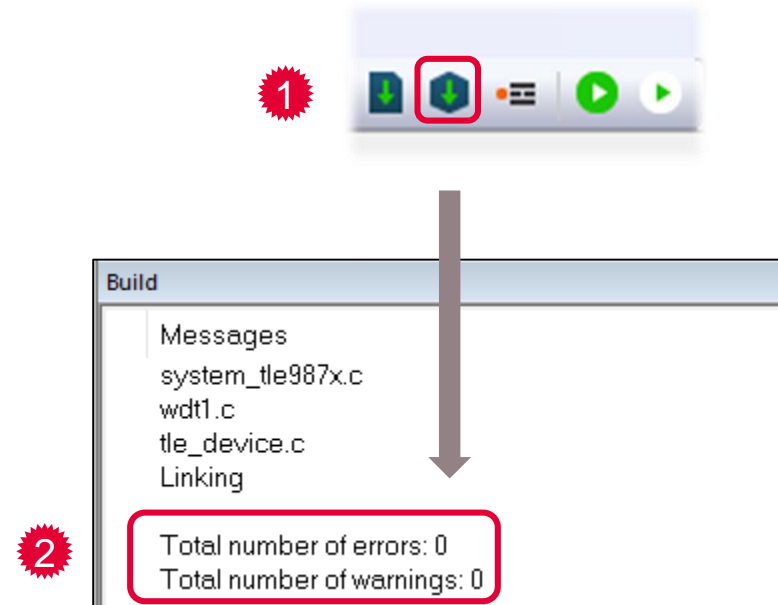
In your workspace:

1. Right click on the project name and select *Add > Add Files...*
 - Select the *main.c* file in the Windows Explorer and click on Open
2. The *main.c* file is visible in your project



4) Compile the Project

1. Compile the project:
 - > Press the *Make* button or press *F7*
2. The Build window shows
0 Error(s) , 0 Warning(s)



Getting Started with IAR Embedded Workbench

Step 5: Add Modules from the Run-Time Environment

5) Add Modules from the Run-Time Environment

In the `TIMER2BLINK.rteconfig` file in the CMSIS Manager:

1. Expand *Device*, then the *SDK* section
2. Select *TIMER2x* and save your configuration
3. Add code to the `main.c` file:
 - i. Include Libraries
 - ii. Initialize Modules

The screenshot shows the IAR Embedded Workbench interface. The top window is the 'Components' manager for the project 'TIMER2BLINK.rteconfig'. It displays a list of software components with columns for 'Sel.', 'Variant', 'Vendor', 'Version', and 'Description'. The 'SDK' section is expanded, and 'TIMER2x' is selected. A red box highlights the 'TIMER2x' row, and a red starburst with the number '2' is next to it. Below the components window, a snippet of the `main.c` file is shown. It includes the following code:

```
54 //***** Includes *****
55 /**
56 *****
57 #include "tle_device.h"
58 #include "port.h"
59 #include "timer2x.h"
60
61 int main(void)
62 {
63     /* Initialization of the hardware modules
64     based on Config Wizard config */
65     TLE_Init();
66 }
```

Red boxes highlight the include statements in the code, and a red starburst with the number 'i' is next to the first include statement. Another red starburst with the number 'ii' is next to the `TLE_Init();` call.

Getting Started with IAR Embedded Workbench

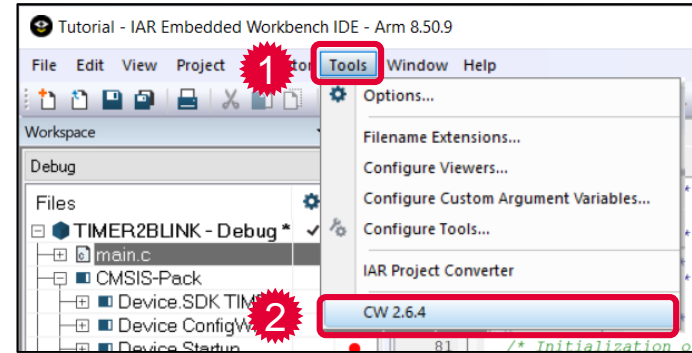
Step 6: Use the Config Wizard v2 (1)

6) Use the Config Wizard v2 (1)

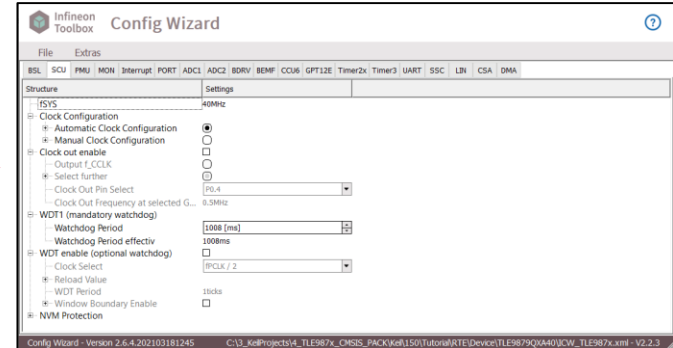
1. Select the tab *Tools*

2. Open *Config Wizard v2*

3. Config Wizard opens in a separate window



3



Getting Started with IAR Embedded Workbench

Step 6: Use the Config Wizard v2 (2)

6) Use the Config Wizard v2 (2)

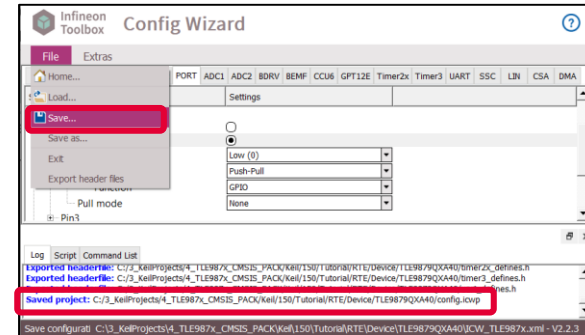
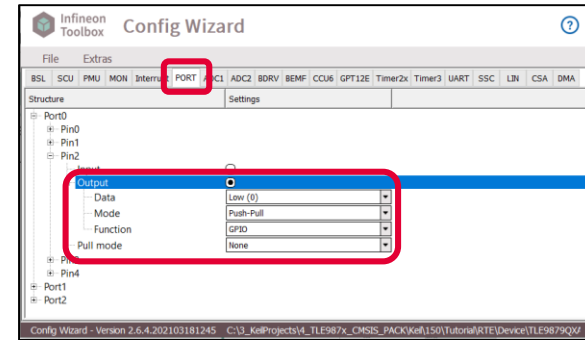
1. Select the tab *PORT*
2. Go to the *Port0 > Pin2* section
3. Configure pin to *Output* mode
4. Save with *File > Save*

1

2

3

4

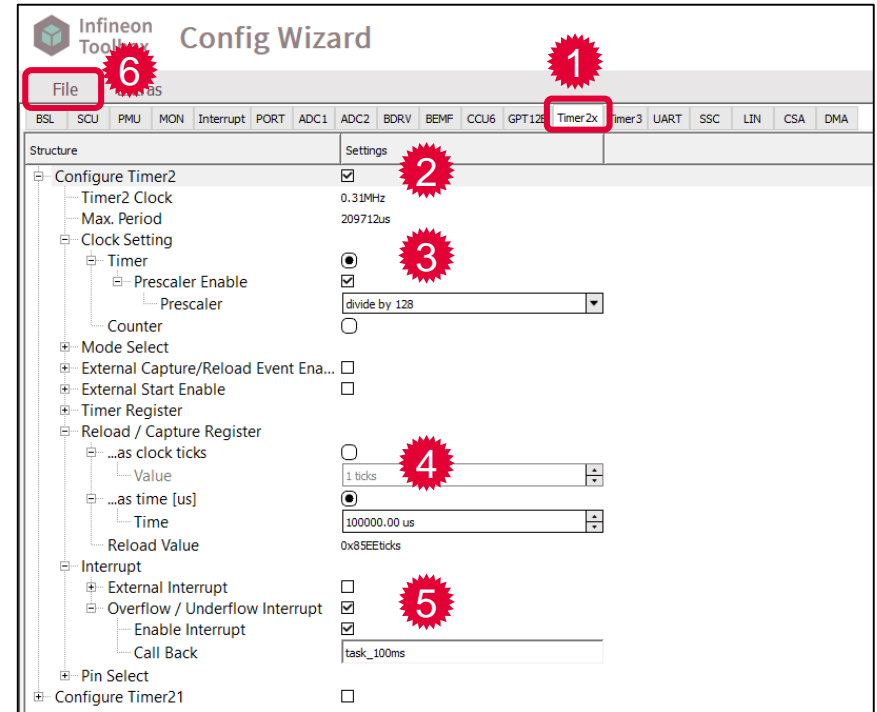


Getting Started with IAR Embedded Workbench

Step 6: Use the Config Wizard v2 (3)

6) Use the Config Wizard v2 (3)

1. Click on the tab *Timer2x*
2. Enable the checkbox *Configure Timer2*
3. In the section *Clock Setting*:
 - › Enable the prescaler and select *divide by 128*
4. In the section *Reload / Capture Register*:
 - › Select *...as time[us]* and enter *100000.00 us*
5. In the section *Interrupt*:
 - › Enable the checkboxes *Overflow / Underflow Interrupt* and *Enable Interrupt*
 - › Call Back Name: *task_100ms*
6. Save with *File > Save*



The screenshot shows the Infineon Config Wizard v2 interface. The 'File' menu is highlighted with a red circle and the number 6. The 'Timer2x' tab is selected, highlighted with a red circle and the number 1. The 'Configure Timer2' checkbox is checked, highlighted with a red circle and the number 2. The 'Prescaler Enable' checkbox is checked, and the 'Prescaler' dropdown is set to 'divide by 128', both highlighted with a red circle and the number 3. The '...as time [us]' radio button is selected, and the 'Time' field is set to '100000.00 us', both highlighted with a red circle and the number 4. The 'Overflow / Underflow Interrupt' and 'Enable Interrupt' checkboxes are checked, and the 'Call Back' field is set to 'task_100ms', all highlighted with a red circle and the number 5.

7) Edit the File main.c

In the file main.c in Keil MDK:

1. Start Timer2 before the “for (;;)” loop
2. Declare and define the function of the interrupt call back
 - › Use API function “PORT_ChangePin()”
3. Save and build (F7) the project



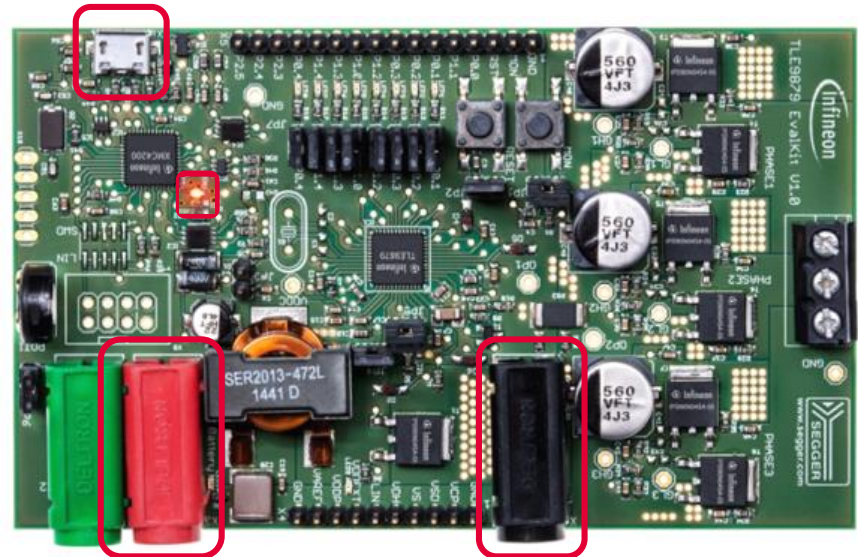
```
87 | /******  
88 | ** Place your application code here  
89 | *****  
90 | TIMER2_Start();  
91 |  
92 | /******  
93 | ** Main endless loop  
94 | *****  
95 | for (;;)   
96 | {
```

```
66 | /******  
67 | ** Private Function Declarations  
68 | *****  
69 | void task_100ms(void);
```

```
106 | void task_100ms(void)  
107 | {  
108 | PORT_ChangePin(0x02, PORT_ACTION_TOGGLE);  
109 | }
```

8) Power up the Evaluation Board

- › Connect micro USB cable
- › Supply board via banana jacks (VBAT, GND)
- › Debug LED lights up

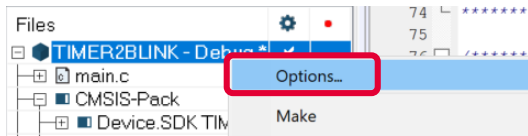


Getting Started with IAR Embedded Workbench

Step 9: Connect the Debugger

9) Connect the Debugger

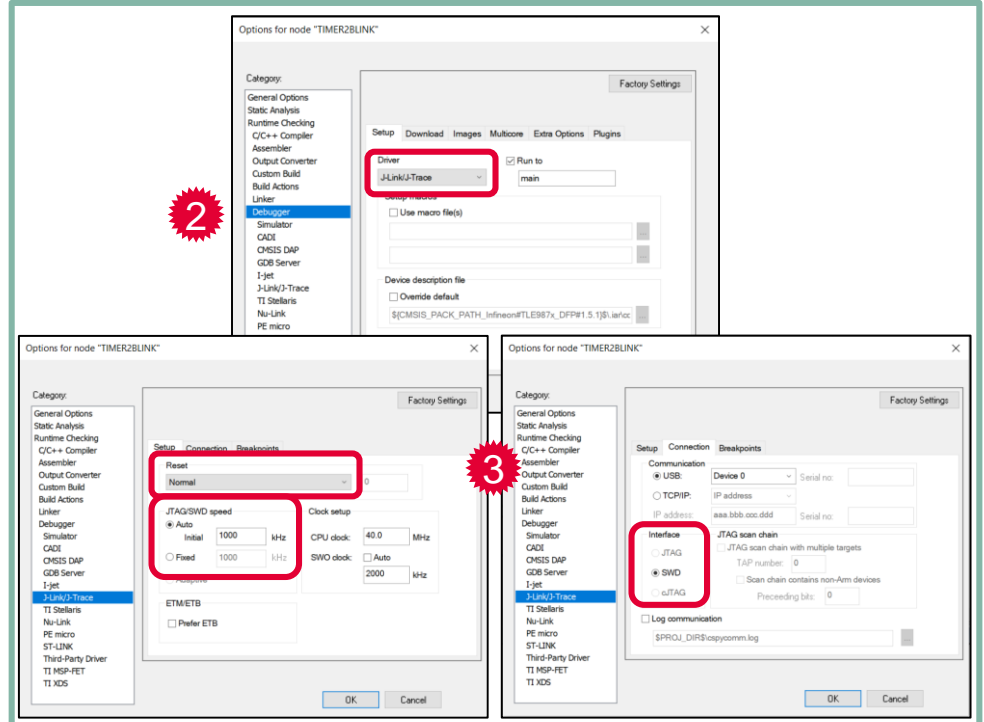
1. Right click on the project name and select *Options*



2. Select the category *Debugger*, then the driver *J-Link/J-Trace*

3. In the category *J-Link/J-Trace*, the following setups should be per default:


- > Setup: *Reset* > *Normal*
- > Setup: *SWD speed* > *Auto*
- > Connection: *Interface* > *SWD*

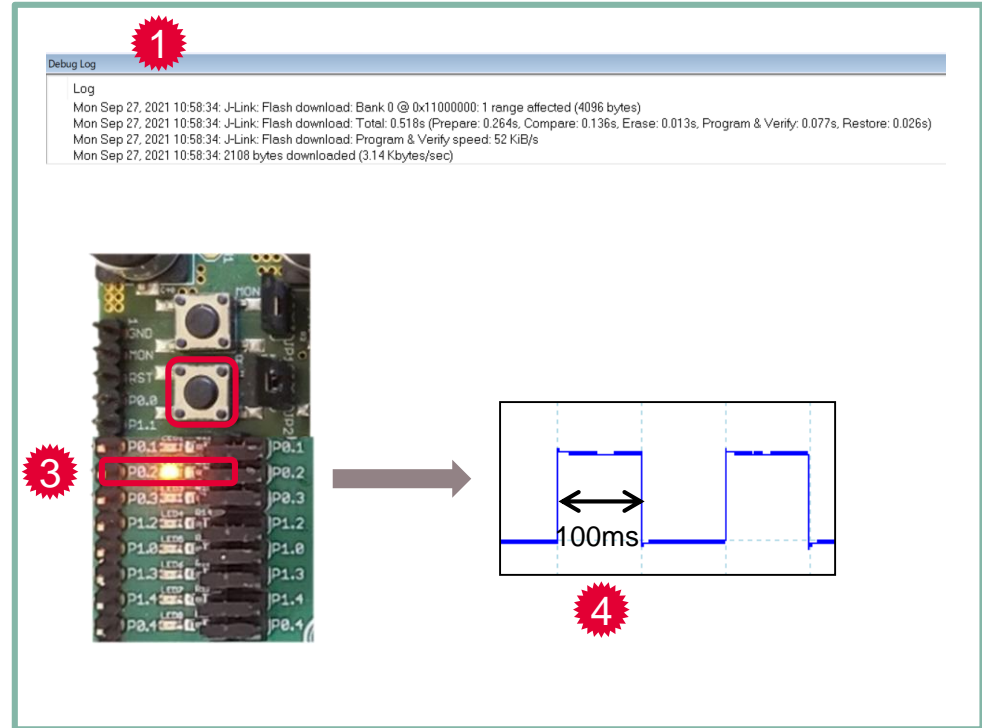


Getting Started with IAR Embedded Workbench

Step 10: Download and Run Code

10) Download and Run Code

1. Select *Project > Download > Download active application* to load the code into the target
 - > *Flash download* is shown in the *Debug Log* window
- > You can also click on  to download the code into the target and start a debug session
2. Press the *Reset* button on the evaluation board
3. The LED on Port P0.2 lights up
4. The port toggles every 100ms



Debug Log

```
Log
Mon Sep 27, 2021 10:58:34: J-Link: Flash download: Bank 0 @ 0x11000000: 1 range effected (4096 bytes)
Mon Sep 27, 2021 10:58:34: J-Link: Flash download: Total: 0.518s (Prepare: 0.264s, Compare: 0.136s, Erase: 0.013s, Program & Verify: 0.077s, Restore: 0.026s)
Mon Sep 27, 2021 10:58:34: J-Link: Flash download: Program & Verify speed: 52 KiB/s
Mon Sep 27, 2021 10:58:34: 2108 bytes downloaded (3.14 Kbytes/sec)
```

1

3

4

100ms

11) Use the Runtime Debug

1. Click on *Download and Debug*



2. Left click in the dark grey area left of the code to place a breakpoint

3. Click on *Go* or press *F5* to start the code execution

4. The code is executed and stops at the breakpoint

- > In this example, every time you click on Run, the LED P0.2 is toggled

2

```
106 void task_100ms(void)
107 {
108     PORT_ChangePin(0x02, PORT_ACTION_TOGGLE);
109 }
```

3



4

```
106 void task_100ms(void)
107 {
108     PORT_ChangePin(0x02, PORT_ACTION_TOGGLE);
109 }
```

Table of contents

1	Toolchain installation for	3
	Keil μ Vision5	4
	IAR Embedded Workbench	5
2	Getting Started for	6
	Keil μ Vision5	7
	IAR Embedded Workbench	8
3	Evaluation board and kit overview	9
4	Product information links	10

MOTIX™ Embedded Power ICs Product Portfolio based on Arm® Cortex®-M processor

Smart Relay IC DC Motor	Smart Half Bridge	Smart H-Bridge Driver	Smart BLDC Driver IC
<p>TLE9842/3/4</p> <ul style="list-style-type: none"> > Window Lift > Sunroof > Wiper 	<p>TLE9845/TLE9851</p> <ul style="list-style-type: none"> > HVAC Fan > Engine Cooling Fan > Fuel Pump > Water Pump 	<p>TLE985x/TLE986x</p> <ul style="list-style-type: none"> > Window Lift > Sunroof > Wiper > Power Folding Roof > Power Sliding Door > Power Trunk/ Tailgate 	<p>TLE987x</p> <ul style="list-style-type: none"> > Fuel Pump > HVAC Fan > Engine Cooling Fan > Water Pump > Oil Pump > Sunroof > Wiper

TLE9869QX – Evaluation Kit

- › 2-Phase N-MOS Bridge
- › Single Shunt in GND path
- › Integrated LIN (inside device)
- › Virtual Com Port via J-Link
- › Debug LEDs
- › Onboard Segger J-Link Debugger

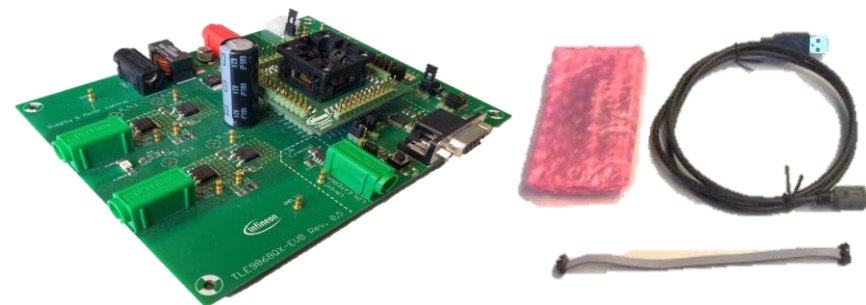
TLE9869 EVALKIT: **SP001388252**



TLE986x – 2-Phase Board with Socket

- › H-Bridge N-MOS Bridge
- › Integrated LIN
- › External LIN Transceiver
- › RS232
- › Debug LEDs
- › Debug Connector SWD
- › J-Link Lite Debugger

TLE986x EVALB_JLINK: **SP001253678**



TLE9879QX – Evaluation Kit

- › 3-Phase N-MOS Bridge
 - › Single Shunt in GND path
 - › Integrated LIN (inside device)
 - › Virtual Com Port via J-Link
 - › Debug LEDs
 - › Onboard Segger J-Link Debugger
- TLE9879 EVALKIT: **SP001389172**



TLE987x – 3-Phase Board with Socket

- › 3-Phase N-MOS Bridge
 - › Integrated LIN
 - › External LIN Transceiver
 - › RS232
 - › Debug LEDs
 - › Debug Connector SWD
 - › J-Link Lite Debugger
- TLE987x EVALB_TQFP: **SP005421936**
TLE987x EVALB_VQFN: **SP005421934**

TLE987x EVALB_TQFP



TLE987x EVALB_VQFN



Table of contents

1	Toolchain installation for	3
	Keil μ Vision5	4
	IAR Embedded Workbench	5
2	Getting Started for	6
	Keil μ Vision5	7
	IAR Embedded Workbench	8
3	Evaluation board and kit overview	9
4	Product information links	10

Product Information Links

Overview

- › [Product Brief](#)
- › [Selection Guides](#)
- › [Product Presentations](#)

- › [Embedded Power IC Overview](#)
- › [TLE986xQX Overview](#)
- › [TLE987xQX Overview](#)

Technical Material

- › [Datasheets](#)
- › [Application Notes](#)
- › [Getting Started](#)
- › [PCB Design Data](#)

- › [TLE986xQX Documents](#)
- › [TLE987xQX Documents](#)

Evaluation Boards

- › [Evaluation Boards](#)
- › [Application Kits](#)

- › [Kits and Boards Overview](#)
- › [Information about TLE9879 Evalkit](#)
- › [Information about TLE9869 Evalkit](#)

Software & Tools

- › [Config Wizard for MOTIX™ MCU](#)
- › [Keil μVision5](#)
- › [IAR Embedded Workbench](#)
- › [Software Examples](#)

- › [Link to Software & Tools](#)

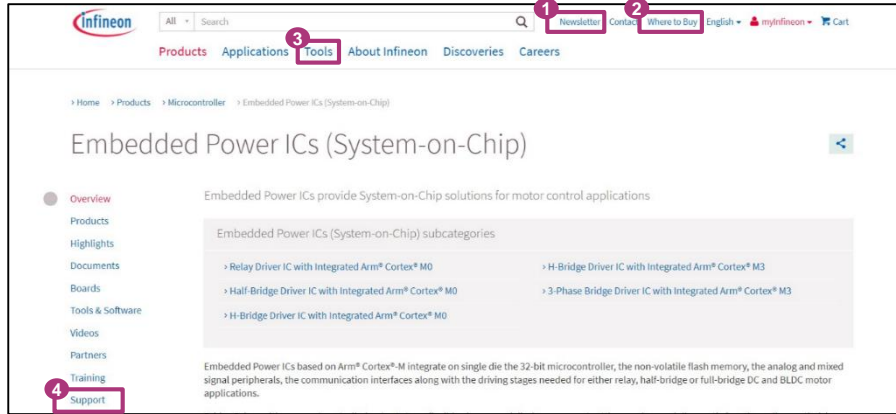
Videos

- › [More than 30 videos](#)

- › [Link to Videos](#)

Support Online Tools and Services

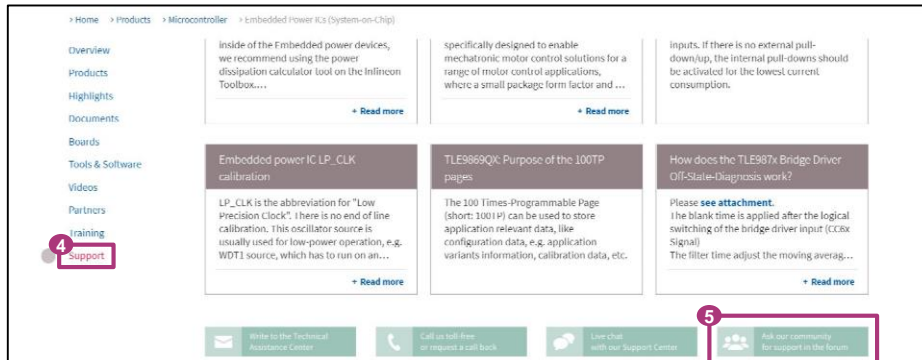
> infineon.com/embeddedpower



1
**Subscribe to
Newsletter**

2
Where to Buy

3
**Tools, Finders
and Selectors**



4
Support

5
Forum



Part of your life. Part of tomorrow.