

Micro Inspector Pro

Getting started

Version 2.1



Micro Inspector Pro

Feature List



Feature	Micro Inspector Pro
View workspace	☑
Create/Edit workspace	☑
Save workspace	☑
View oscilloscope	☑
Create/Edit oscilloscope	☑
Target device selection	☑
Symbol browser	☑
Create/Edit memory dump screen	☑
View memory dump screen	☑

Getting started: Micro Inspector Pro

Chapter 1: Preconditions

- › Necessary preconditions

Chapter 2: Installing Micro Inspector Pro

- › Download Infineon Toolbox
- › Install the tool
- › How to get a license

Chapter 3: Working with Micro Inspector Pro

- › Informative notes
- › Creating a Dashboard
- › Creating an Oscilloscope

Getting started: Micro Inspector Pro

Chapter 1: Necessary Preconditions

1. Have an Evalkit available and connected
(in this example the TLE9879)

2. A compiled program for that board as an ELF/axf file
(or a file with respective information)

3. HEX file/compiled program flashed to the board
(in this example the TLE9879)

4. SEGGER JLink driver installed in host PC
<https://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack>



Getting started: Micro Inspector Pro




Chapter 2: Infineon Toolbox Installation

Open <https://www.infineon.com/cms/en/tools/landing/infineontoolbox.html>

› Download and install the Infineon Toolbox

› Home › Tools › Infineon Tools

Please follow these steps to install the Toolbox offline launcher.

Optional step: Backup Toolbox Launcher (< 2021.1.0)	Step 1: Download & Install	Step 2: Launch and discover tools
		
Backup your previous Infineon Toolbox Launcher directory (C:\Infineon\Toolbox)	Download Infineon Toolbox Launcher and follow the installation instructions.	Launch the Infineon Toolbox Launcher and discover/install/manage the available tools for you.

System Requirements

1. Infineon Toolbox is compatible with Windows 7 or later.
2. A recent multi-core processor, 2 GB of RAM and at least 200 MB of disk space are recommend. Installation and usage of tools will require additional disk space and RAM.
3. An Internet connection allowing HTTPS connections is required to install and update tools.

Get the latest version here:

Download now

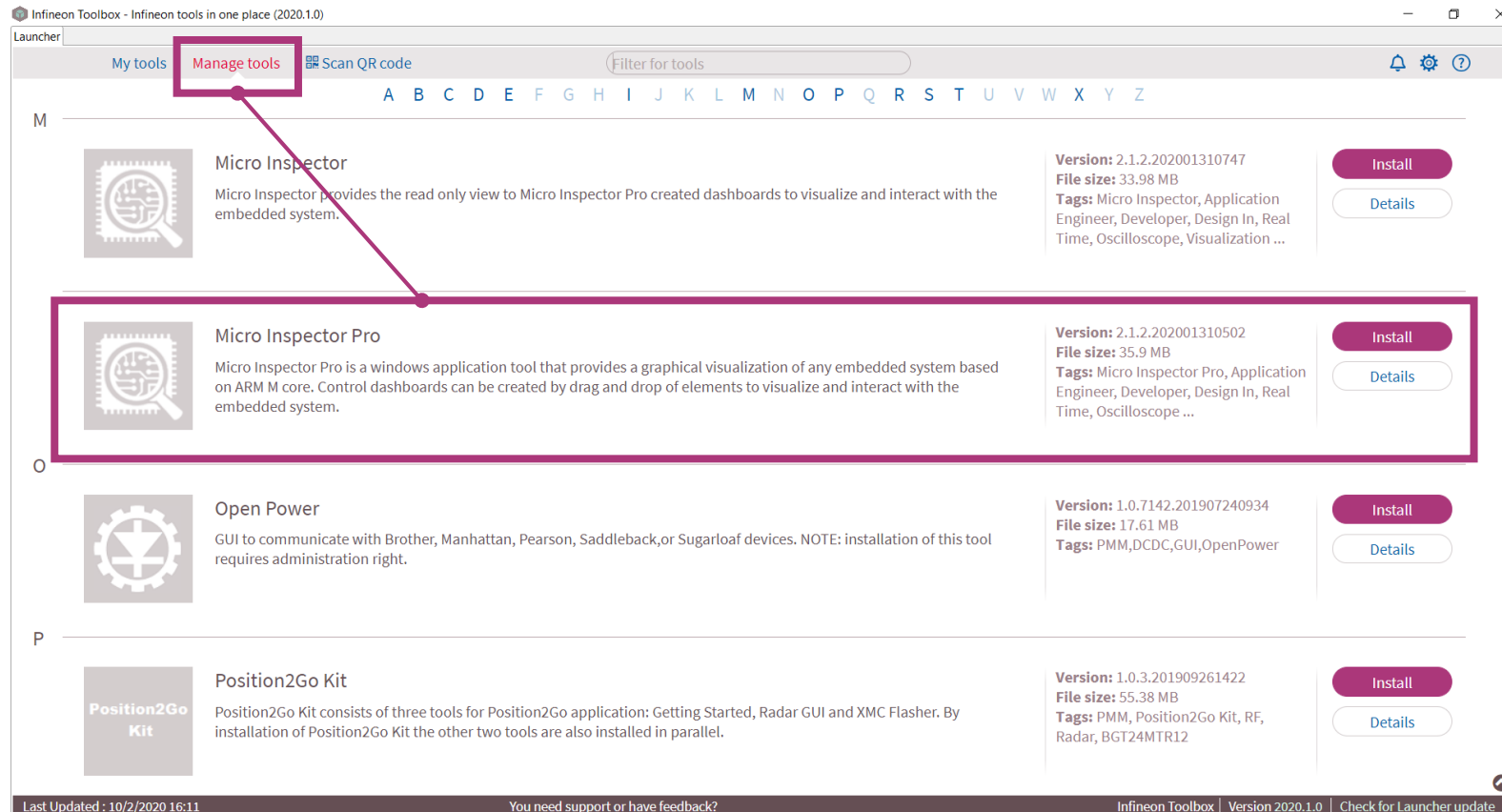
[Download Old Infineon Toolbox Launcher](#)

Getting started: Micro Inspector Pro

Chapter 2: Micro Inspector Pro installation

Open Infineon Toolbox

- › Navigate to “Manage Tools”
- › Install the Micro Inspector Pro

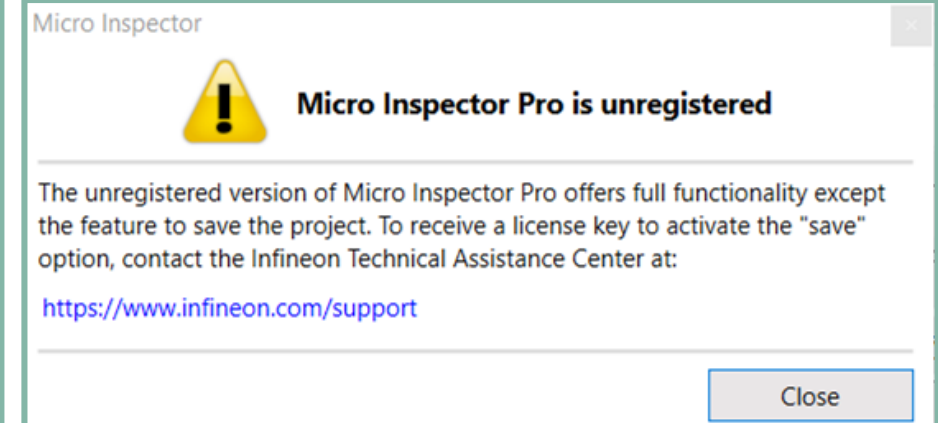


Getting started: Micro Inspector Pro

Chapter 2: How to upgrade from unregistered to registered version

Open the Micro Inspector Pro tool

- › This message will be displayed
- › Click on the link inside the pop-up window
- › <https://www.infineon.com/tac>



- › Provide the necessary information on the Infineon support page and submit your request
- › You get your license details via e-mail (approx. after 2 working days)
- › Open the tool and click on "Unregistered" in the bottom right corner
- › Enter the license details in the window that opens
- › Proceed with the licensed tool





Micro Inspector Pro has two modes:

Default mode: Not connected to device.

Elements can be added to the Dashboard, variables can be assigned but elements are not clickable

Play mode: Connected to device.

Graphical elements can be clicked

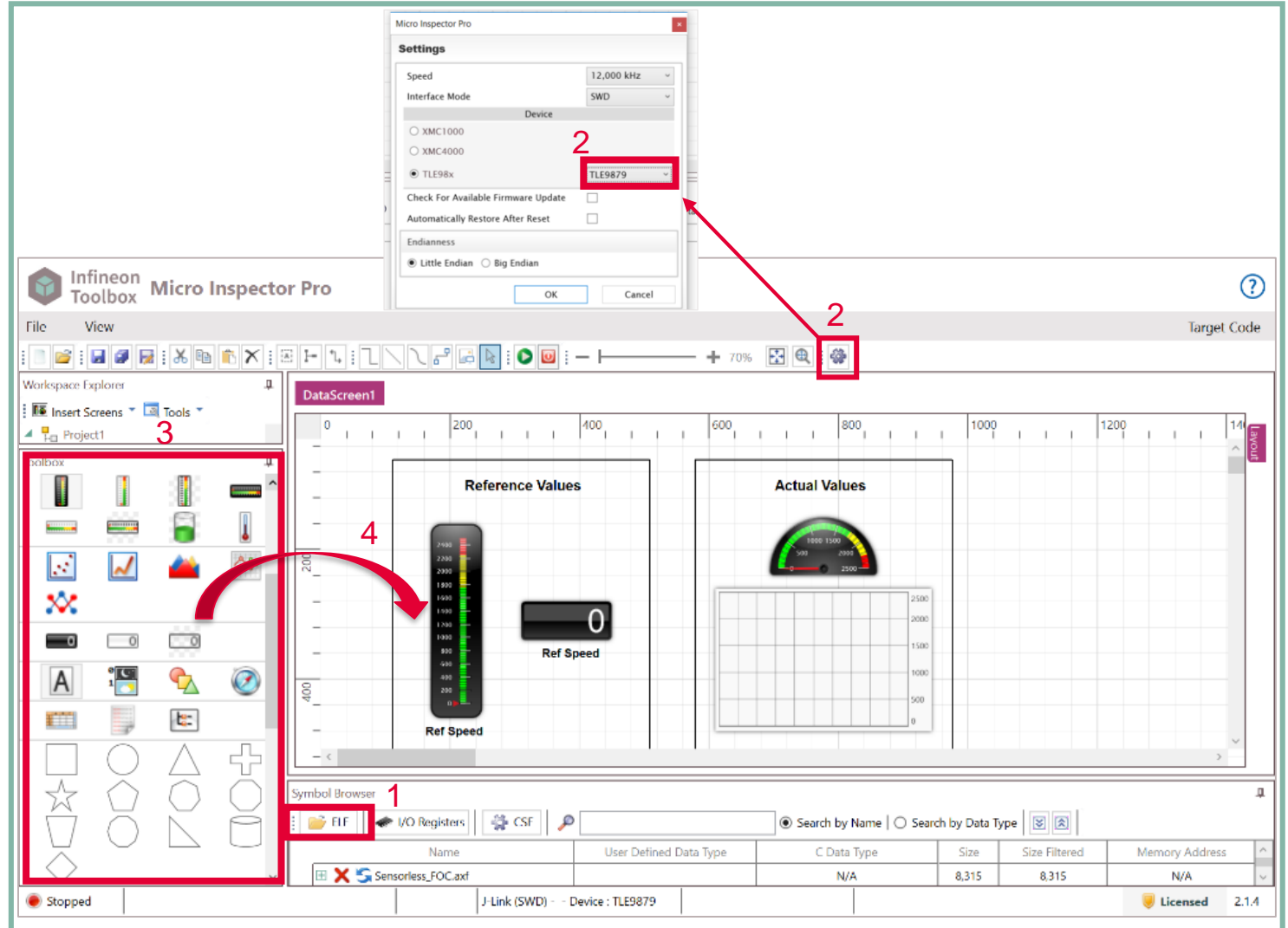
Comments:

- › To ensure a smooth Dashboard creation, make sure that the zooming level is set to 100%.
- › The use of writing variables can lead to changes in your program.

Getting started: Micro Inspector Pro

Creating a Dashboard I

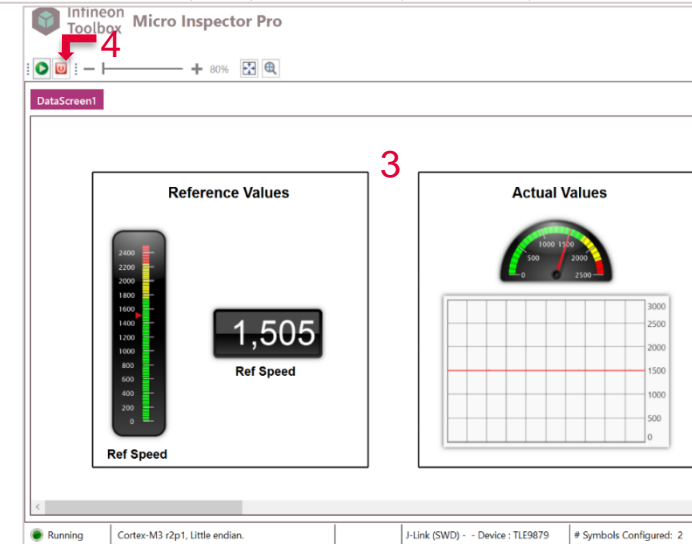
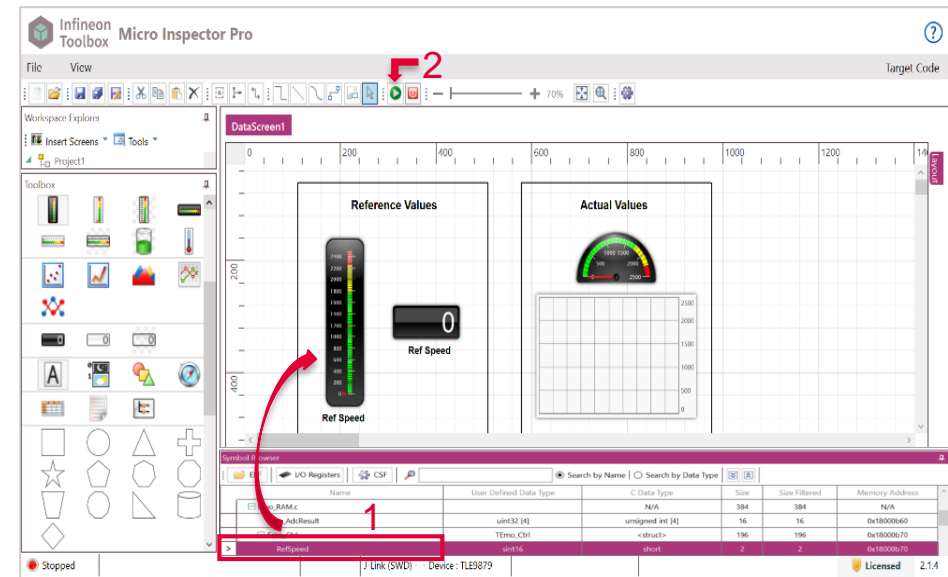
1. Load the ELF/axf file of your project in the Symbol Browser located in:
`<example_project_path>\Boards\Infineon\TLE9879_EvalKit\BLDC_SENSORLESS_FOC_EXAMPLE_TLE987X\Objects` (Make sure that the program is already compiled)
2. Choose TLE9879 in the Settings Menu
3. Choose the elements you need from the Micro Inspector Pro Toolbox
4. Add them to the data screen using drag and drop



Getting started: Micro Inspector Pro

Creating a Dashboard II

1. Assign the right variables (only global or static) from your ELF/axf file to the respective elements via drag and drop from the Symbol Browser
2. Click on 'Run' to run your Dashboard
3. The executed example Dashboard looks like this
4. Clicking on 'Stop' terminates the execution.



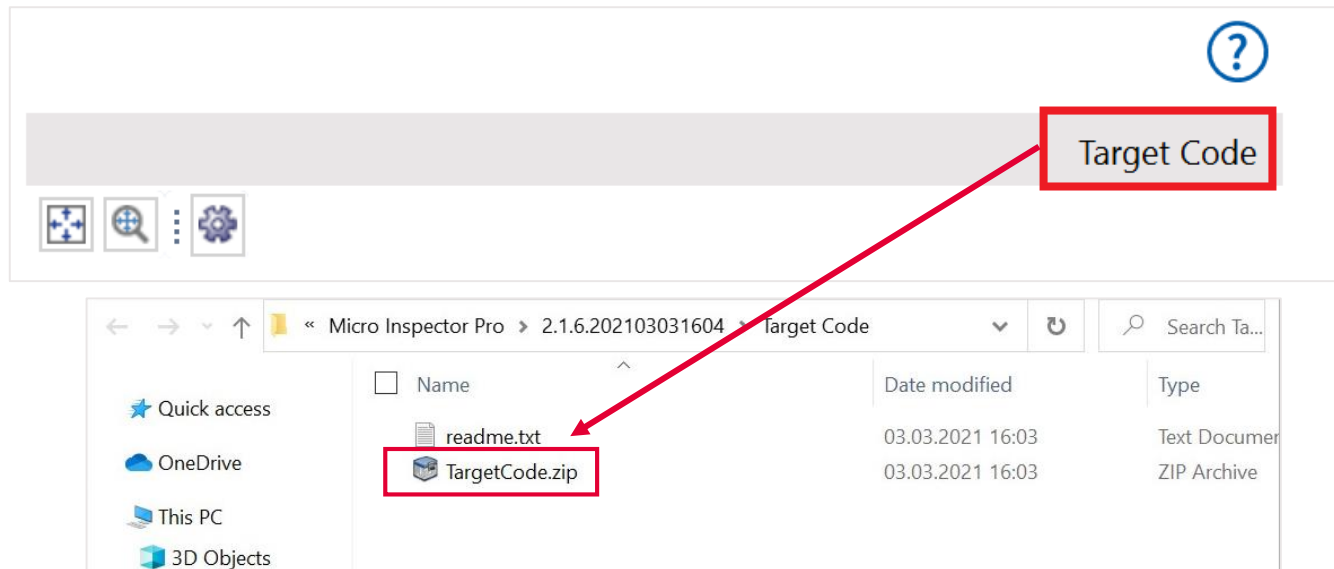
Getting started: Micro Inspector Pro

Creating an Oscilloscope I

Project adaption to support the Oscilloscope feature

Preconditions:

- › Any IDE (in this case KEIL μ Vision is used): <https://www.keil.com/demo/eval/arm.htm>
- › “Config Wizard for Embedded Power ICs” installed in the Infineon Toolbox
- › Target code to use the Oscilloscope of Micro Inspector Pro:
You can find this code by clicking on 'Target Code' in the menu bar of Micro Inspector Pro (TargetCode.zip)



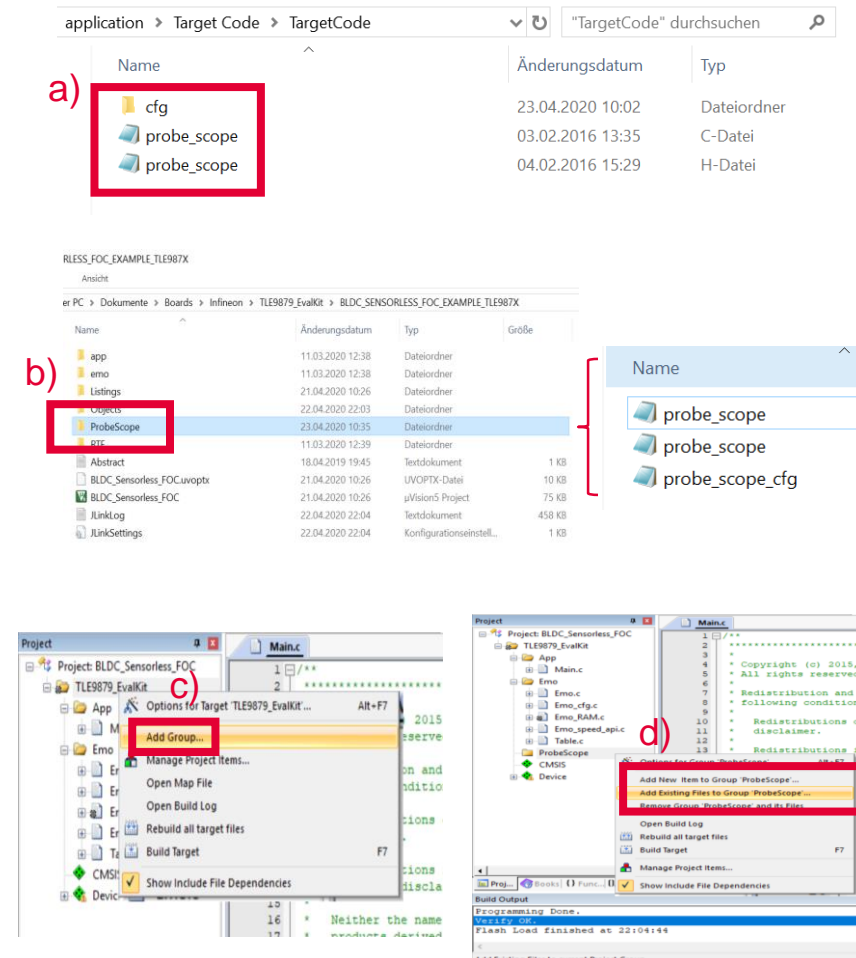
Getting started: Micro Inspector Pro

Creating an Oscilloscope II

Project adaption to support the Oscilloscope feature

1. Add the target code to your project, files:

- Copy the probe_scope.c, probe_scope.h and probe_scope_cfg.h files from the target code
- Create a new folder in the example project and paste the probe_scope files in this folder
- Add a new group under your main folder in your KEIL project
- Add the probe_scope.c file to the new group

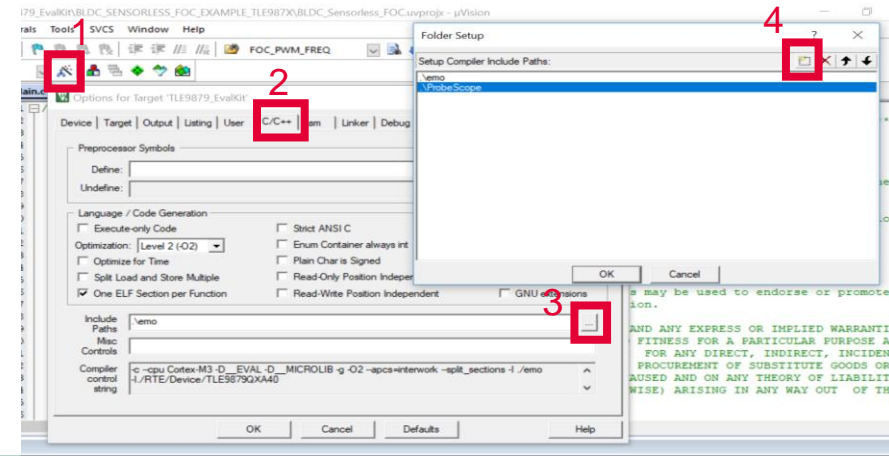


Getting started: Micro Inspector Pro

Creating an Oscilloscope III

Project adaption to support the Oscilloscope feature

- e) Setup the project to include the folder by
1. Clicking on "Options for target"
 2. "C/C++" perspective
 3. Include the path
 4. Select the folder with the probe_scope files



2. Add the target code to your project, code
- a. Include the probe_scope.h file in your main.c file in the KEIL project

Code: `#include "probe_scope.h"`

```
44  /**                               Includes
45  *****
46  #include "tle_device.h"
47  #include "Emo_RAM.h"
48  #include "probe_scope.h"
```

- b. Call the ProbeScope_Init function in your main function

Code: `ProbeScope_Init(20000);`

```
82  int main(void)
83  {
84      /* Initialize device drivers
85      /* Note: Watchdog is already
86      TLE_Init();
87      Emo_Init();
88      ProbeScope_Init(20000);
89  }
```

Getting started: Micro Inspector Pro

Creating an Oscilloscope IV

Project adaption to support the Oscilloscope feature

- c) Open the Emo_RAM.c file and include the same "probe scope" header file as in main.c

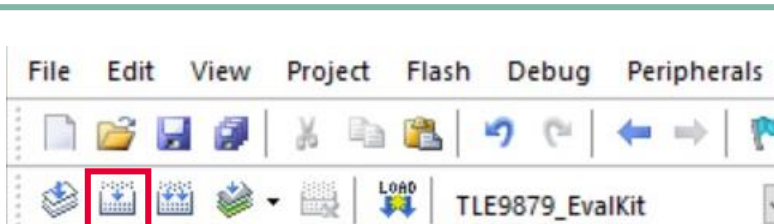
Code: `#include "probe_scope.h"`

```
40 //*****
41 /**
42 ..... Revision
43 .....
44 * V0.4.0: 2013-06-03, SS: DC-
45 * V0.3.0: 2013-02-28, SS: PI
46 * V0.2.0: 2012-12-13, SS: Ctr
47 * V0.1.0: 2012-11-12, SS: Ini
48 */
49 .....
50 .....
51 ..... Includes
52 .....
53 #include "Emo_RAM.h"
54 #include "probe_scope.h"
55 .....
56 ..... Private
57 .....
58 .....
```

- d) Call the `ProbeScope_Sampling();` function in the "Emo_handleFOC" function → This function returns the output value

```
374 .....
375 /* Perform space vector modulation */
376 Emo_lExeSvm(&Emo_Svm);
377 /* Filter for Iq */
378 Emo_Ctrl.RotCurrImagdisplay = Mat_Exel
379 ProbeScope_Sampling();
380 } /* End of Emo_HandleFoc */
381 .....
```

- e) Configure the probe_scope files:
Compile the project to load the header files, open the probe_scope_cfg.h file and configure the values according to your requirements (see example below for TLE9879 EvalKit)



```
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
19 .....
20 .....
21 .....
22 #define PROBE_SCOPE_MAX_CH 3 /* The maximum number of channels: [1,4]. */
23 #define PROBE_SCOPE_MAX_SAMPLES 250 /* The maximum number of samples per channel. */
24 #define PROBE_SCOPE_16_BIT_EN 1 /* The maximum size of each sample is 16-bits: [0,1]. */
25 #define PROBE_SCOPE_32_BIT_EN 0 /* The maximum size of each sample is 32-bits: [0,1]. */
26 #define PROBE_SCOPE_SAMPLING_CLK_HZ_DFLT 20000 /* Default freq (Hz) to configure the timer at init. */
27 .....
```

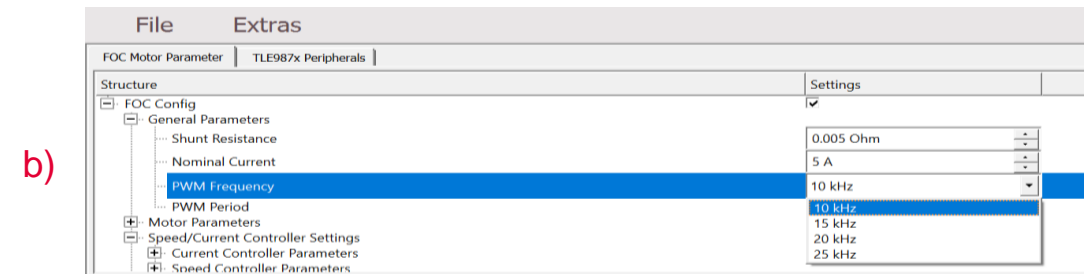
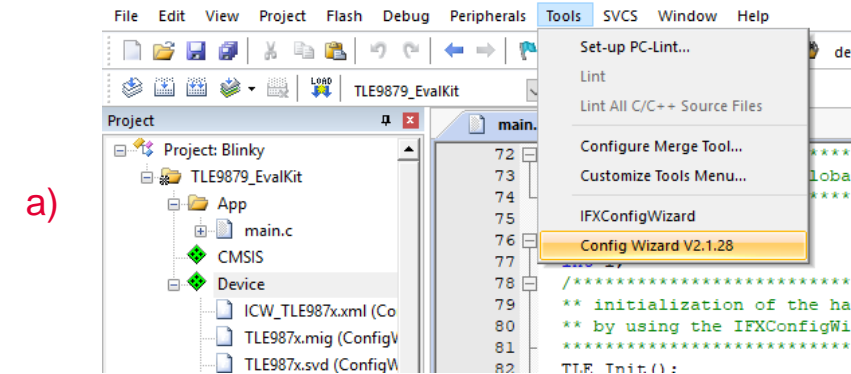
Getting started: Micro Inspector Pro

Creating an Oscilloscope V

Project adaption to support the Oscilloscope feature

3. Set the PWM Frequency:

- Open Config Wizard by choosing Tools > Config Wizard V2
- Set the PWM Frequency to 10 kHz → Save the change
- Go to Keil uVision → Compile project → Flash to target device

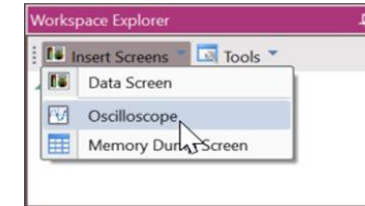


Getting started: Micro Inspector Pro

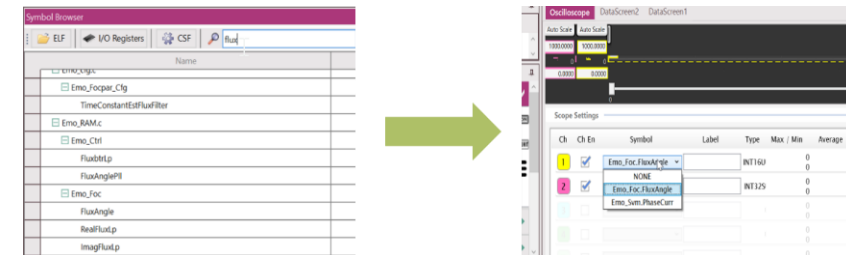
Creating an Oscilloscope VI

Project adaption to support the Oscilloscope feature

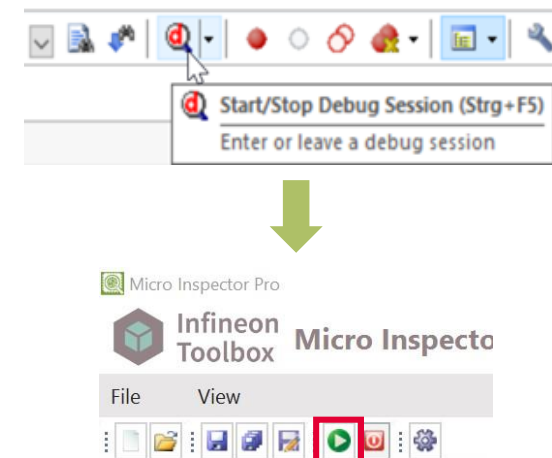
4. Start Micro Inspector Pro
 - a. Open the ELF/axf File
 - b. Add an Oscilloscope to your screens



5. Configure the Oscilloscope
 - a. Search for the necessary variables in the Symbol Browser
 - b. Add them to channels by double-clicking on them



6. Establish a debug connection to the target device
 - a. Initiate the debug session in KEIL. This enables Micro Inspector Pro to establish a connection to the debugger
 - b. Run the example project from Micro Inspector Pro





Part of your life. Part of tomorrow.