UM11555

FreeMASTER sensor tool user manual

Rev. 1.4 — 6 June 2023 **User manual**

Document Information

Information	Content
Keywords	FreeMASTER, ISSDK, Sensors, Evaluation, Application Development, Real-time monitoring, Data visualization
Abstract	FreeMASTER-Sensor-Tool User manual

Revision history

Rev	Date	Description
1.4	20230606	• Document information: added keywords "Evaluation, Application Development". • Section 3.4, revised the FreeMASTER sensor tool directory structure shown in Step 2 adding directories for fxls8961af, fxls8971cf, and nmh1000. • Section 4.1, added five sensor evalution boards to Step 1 and added new note after Step 2. • Section 4.2, updated the directory structure in Step 1 and added a note with steps in Step 2. • Section 4.3, Step 7: revised "FRDMK22F-A8974_FXLS8974CF_Demo" to "FRDM_STBI_A8974_FXLS8974CF_Demo" and updated the image in Figure 6.
1.3	20230601	• Section 3.3, revised "Download the installer file..." to "Download the latest installer file.…" • Section 6, revised the URL for reference [7] linking to the download area for the current set of Windows installers.
1.2	20220707	• Section 2, inserted new bullet "Real-Time sensor data logging.…" • Section 3, revised the section name from "FreeMASTER sensors tool bring-up" to "Installing Free MASTER sensor tool". • Section 3.3, removed "3.1" from the first paragraph. • Section 4.1, revised as follows: – Revised the formatting for several unordered bulleted items into ordered list steps and a note. – Under "The current release supports..." revised the first bullet from "FRDM-K22F-A896x (FXLS896xAF Demo)" to "FRDM-K22F-A896x (FXLS8964AF, FXLS8967AF Demo)". – Revised "This document refers to FRDM-K64F-AGM01 (FXOS8700 Demo) for execution steps." to "This document refers to FRDM-K22F-A8974 (FXLS8974CF Demo) for execution steps." – Revised "Connect FRDM-K64F-AGM01 evaluation kit..." to "Connect FRDM-K22F-A8974 evaluation kit...." • Section 4.2, revised both steps including the FreeMASTER sensor tool directory structure shown in Step 1. • Section 4.3, revised as follows: – Step 4, Figure 3, updated the image. – Step 7, revised the content of the step and updated the image in Figure 6, – Step 8, revised the content of the step and updated the image in Figure 7. – Step 9, revised the content of the step and updated the image in Figure 8. – Step 10, Figure 9, updated the image. • Section 5.1, Step 10, revised the evaluation kit name from "FRDM-K64F-AGM01" to "FRDM-K22F-A8974". • Section 5.2, Step 3, revised the evaluation kit name from "FRDM-K64F-AGM01" to "FRDM-K22F-A8974". • Section 6, updated "3.1" to "3.x" in reference 7.
1.1	20220111	• Section 3.4, revised the sensor tool directory structure below step 2 adding "fxls896xaf" and "fxls8974cf". • Section 4.1, inserted "FRDM-K22F-A896x (FXLS896xAF Demo)" and "FRDM-K22F-A8974 (FXLS8974CF Demo)" to the list of supported sensor evaluation boards • Revision history, relocated the revision history from the end to the start of the document to conform with NXP document content guidelines.
1	20210128	Initial release

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

1 Overview

FreeMASTER sensor tool is the evaluation and application development software based on NXP FreeMASTER framework for IoT, industrial, and medical sensors. It provides an out-of-box sensor demonstration GUI for quick sensor evaluation. It also provides a development framework for end users to extend/create their custom applications.

FreeMASTER[1] is a real-time debug monitor and data visualization tool ideal for application development in support of IoT, industrial, and medical applications. FreeMASTER:

- supports non-intrusive monitoring of variables on a running system
- displays multiple variables on oscilloscope-like displays as standard widgets (gauges, sliders, and more)
- displays variable data in text form, offering simple-to-use data recorders.

FreeMASTER can link with custom HTML, MATLAB, or Excel to other scriptable frameworks to add MCU hardware into control loops.

The FreeMASTER sensor tool enables quick out of the box sensor demonstrations and provides flexibility for end users to rapidly prototype and create custom GUIs.

2 Features

- **Real-time sensor output monitoring**: Monitor multiple memory variables at individual sampling rates and chart up to eight streams in the oscilloscope/graph view.
- **Real-time sensor register control**: Modify memory variables and registers in real-time. Control hardware with real-time registers, and control the write capability of variables.
- **Data visualization**: Enables third-party instrumentation components inserted into the HTML code as embedded ActiveX objects. Allows for the creation of user-friendly displays of complex real-time data dashboards.
- **Sensor register page**: Provides a register map of the sensors and allows quick read and write of different register bits in real-time, allowing detailed sensor evaluation.
- **Out of the box sensor demonstration**: Provides quick visualization of sensor data and other sensor outputs based on the pre-configured sensor settings in the firmware.
- **Ease of use: Development platforms integration**
  - Integration with MCUXpresso[2] SDK and ISSDK[3] with multiple toolchain support (MCUXpresso IDE, IAR, KEIL, Arm GCC) for embedded application development.
  - JavaScript-powered HTML control forms can be extended. Users can provide an arbitrary collection of open source instrumentation gauges, dials, knobs, and sliders to create complex, elegant custom visual dashboards.
- **Real-time sensor data logging**: Supports real-time data logging using the "Pipes" feature at higher data rates (tested up to 400 Hz).

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

3 Installing FreeMASTER sensor tool

3.1 Prerequisite

This document assumes completion of the following prerequisites prior to attempting to run the FreeMASTER Sensors tool.

- Availability of supported Sensor evaluation board[4].
- User understanding of the debug environment set up for the Freedom family of development boards using OpenSDA[5] or third-party debugger with their IDE of choice
- Recommended toolchain/IDE (MCUXpresso IDE[6]) is installed on the development PC.
- User familiarity with the MCUXpresso SDK and ISSDK.
- Windows drivers are installed on the development PC corresponding to the OpenSDA application selected.

3.2 System configuration

Table 1 shows the system configuration required for the sensor evaluation tool.

Table 1. Recommended system configuration

Parameter	Recommended Configuration
Operating System	Microsoft Windows 7 or later
Communication to the target hardware	Serial RS-232 or USB port
Processor Speed	2.8 GHz
RAM	8 GB
Free hard disk space	20+ GB

3.3 Download and install FreeMASTER

The FreeMASTER application is distributed as a standalone, single-file, self-extracting, executable file.

Download the latest installer file from the FreeMASTER tool Windows installation page[7], run it, and proceed according to the onscreen instructions. For more details, refer to the FreeMASTER for Embedded Applications user guide[8].

The FreeMASTER installation comes with several plug-in modules, enabling it to access the target hardware over alternative communication interfaces.

- The BDM Communication plug-in enables FreeMASTER to perform basic memory access operations on the supported platforms without any target CPU intervention. The BDM plug-in supports the BDM and JTAG debugging probes like the PE Micro Multilink, SEGGER J-Link, and Arm CMSISDAP.
- The packet-driven BDM Communication plug-in can be used as an additional layer on top of the BDM plug-in to enable high-level protocol commands like Recorder, TSA, or memory protection. The BDM plug-in uses either the JTAG or BDM interface to exchange communication protocol frames with the target driver, rather than just accessing the data directly.
- The FreeMASTER-over-CAN plug-in enables using FreeMASTER services over a CAN interface.

For more details about the communication plug-in modules, see the "readme" documents installed together with the latest FreeMASTER application.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

3.4 Download and install FreeMASTER sensor tool

The FreeMASTER sensor tool provides the following:

- Sensors FreeMASTER embedded application: an ISSDK-based sensor embedded application, integrated with FreeMASTER drivers, using a proprietary communication protocol to communicate to FreeMASTER host side.
- Sensors FreeMASTER Host GUI: FreeMASTER based host GUI application enabling direct read/write access mapping to sensor registers, variables along with control page, and register page implementation.

To download and install the FreeMASTER sensor tool, perform following steps:

1. Download the windows installer from the "Download" section on the FreeMASTER sensor tool webpage.
2. Run the installer and follow any onscreen instructions. By default, the installer app installs the package into c:\nxp\FreeMASTER_Sensor_Tool folder and refer to it as "<FreeMASTER Sensor Tool directory>".

```
<FreeMASTER sensor tool directory>
.
├── css       <Folder containing CSS files>
├── docs      <Folder containing user manual document>
├── js        <Folder containing required JS packages>
├── resources <Folder containing images>
└── sensors   <Folder containing sensor specific GUIs>
      ├── fxls8471
      ├── fxls8962
      ├── fxls896xaf
      ├── fxls8974cf
      ├── fxls8961af
      ├── fxls8971cf
      ├── nmh1000
      ├── fxos8700
      └── mma865x
```

4 Running Out-of-box sensors demonstrations

4.1 Plug-in sensor evaluation board

1. Connect the sensor evaluation board to the PC via the USB cable between the OpenSDA USB port on the board and the USB connector on the PC.

   The current release supports the following sensor evaluation boards:
   - FRDM-K22F-A896x (FXLS8964AF, FXLS8967AF Demo)
   - FRDM-STBA-A896x (FXLS8964AF, FXLS8967AF Demo using LPCXpresso55S16-EVK)
   - FRDM-K22F-A8974 (FXLS8974CF Demo)
   - FRDM-STBI-A8974 (FXLS8974CF Demo using LPCXpresso55S16-EVK)
   - FRDM-STBI-A8971-A8971 (FXLS8971CF Demo using LPCXpresso55S16-EVK)
   - FRDM-STBA-A8961 (FXLS8961AF Demo using LPCXpresso55S16-EVK)
   - FRDMSTBI-NMH1000 (NMH1000 Demo using FRDM-KE15Z)
   - FRDM-K64F-AGM01 (FXOS8700 Demo)
   - FRDM-K64F-AGM04 (MMA8652 Demo)
   - FRDMKL27Z-A8471 (FXLS8471 Demo)
   - FRDM-K22F-AGMP03 (FXLS8962 Demo)

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Note: *This document refers to FRDM-K22F-A8974 (FXLS8974CF Demo) for execution steps. For other supported sensor demos, apply similar steps.*

2. Connect FRDM-K22F-A8974 evaluation kit to the development PC via the USB cable between the OpenSDA USB port on the board and the USB connector on the PC.

   Note: *In order to run FXLS8974CF with LPCXpresso55S16-EVK, Connect USB cable from LINK2 port on LPC55S16-EVK to your test machine.*

4.2 Download sensor demo firmware

1. Go to “<FreeMASTER sensor tool directory>/sensors/fxls8974cf” folder.

```
< FreeMASTER sensor tool directory >
.
└── sensors <Folder containing sensor specific GUIs>
      └── fxls8974cf
            ├── FRDM_STBI_A8974_FXLS8974CF_Demo.pmpx
            ├── demo_details.htm
            ├── frdmk22f_a89xx_i2c_project.bin
            ├── frdmk22f_a89xx_spi_project.bin
            ├── hardware_configuration.htm
            ├── index.html
            ├── lpcxpresso55s16_a89xx_i2c_project.bin
            ├── lpcxpresso55s16_a89xx_spi_project.bin
            └── reg.csv
```

2. Drag and drop “frdmk22f_a89xx_i2c_project.bin” firmware into “FRDM-K22FD” virtual mass drive on the Windows PC. When the file copy operation completes, unplug and replug the board into the Windows PC.

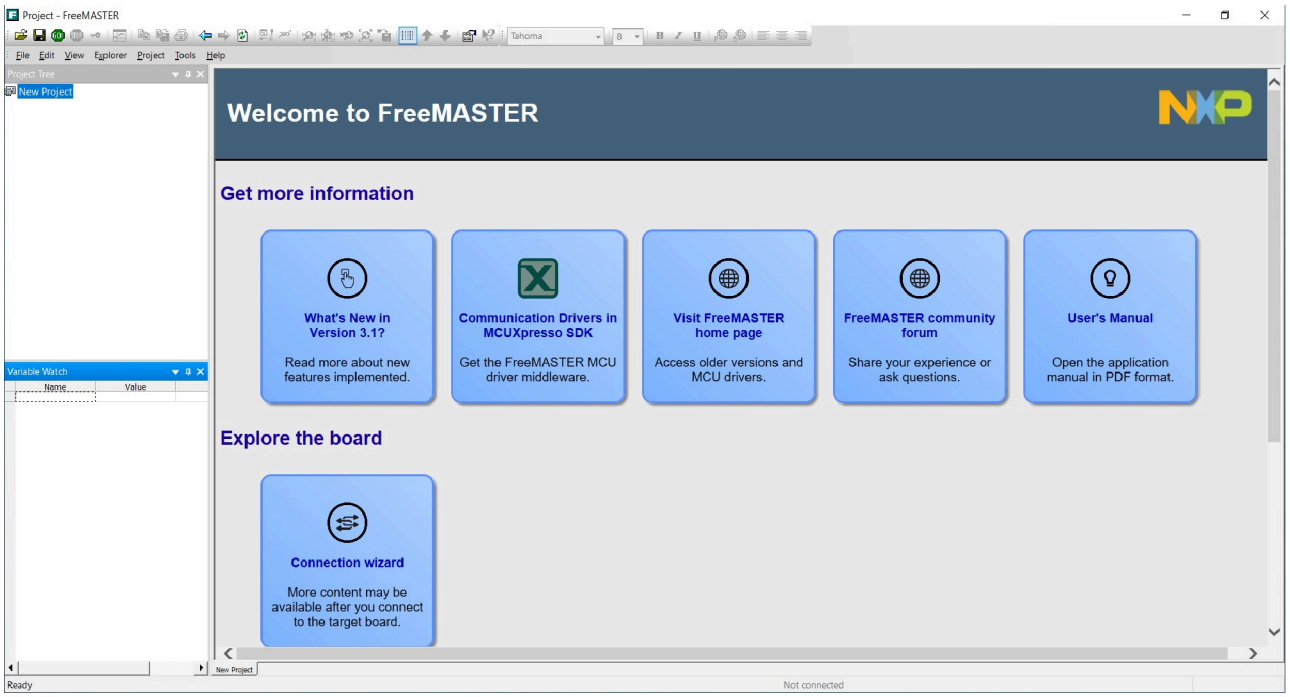
   Note: *In order to run FXLS8974CF demo with LPCXpresso55S16-EVK, follow these steps:*
   - *Check your test machine "Device Manager", you should see LPC-Link|| UCom Port (COMx) popping up.*
   - *Go to “<FreeMASTER sensor tool directory>/sensors/fxls8974cf/lpc55s16_loader” folder.*
   - *Open a command prompt to lpc55s16_loader folder.*
   - *Run "lpc55s16_load.bat COMx ../<*.bin>" on command prompt.*
     - *COMx is the COM port identified under your test machine's "Device Manager".*
     - **.bin is the bin file e.g. "lpcxpresso55s16_a89xx_i2c_project.bin" or "lpcxpresso55s16_a89xx_spi_project.bin".*
   - *Follow the instructions when lpc55s16_load.bat is executed.*

4.3 Run sensor demo

1. Launch FreeMASTER Windows application.
2. Click “Connection Wizard” and click “Next>”.

All information provided in this document is subject to legal disclaimers.

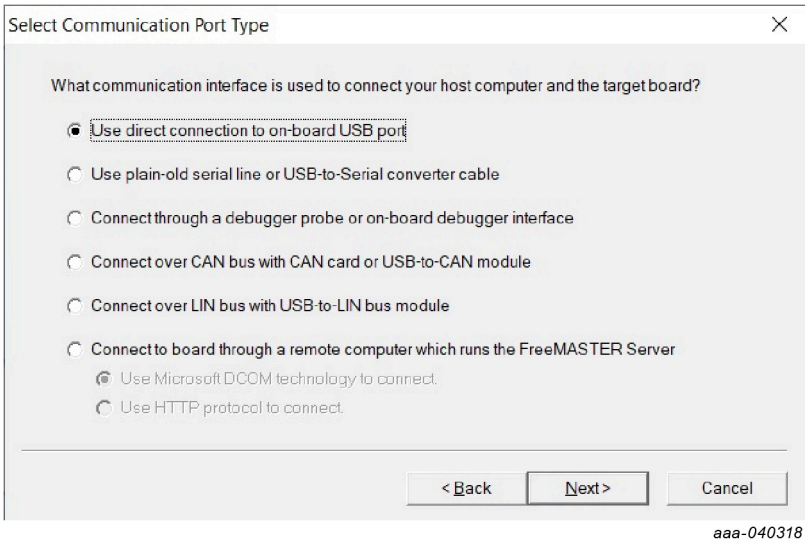
© 2023 NXP B.V. All rights reserved.



aaa-040317

Figure 1. FreeMASTER welcome screen

3. Select “Use direct connection to on board USB port” and click “Next>”.



aaa-040318

Figure 2. Communication port selection dialog

4. The FreeMASTER tool detects the COM port with the configured baud-rate automatically. Confirm the COM port and baud-rate, click “Next>”.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

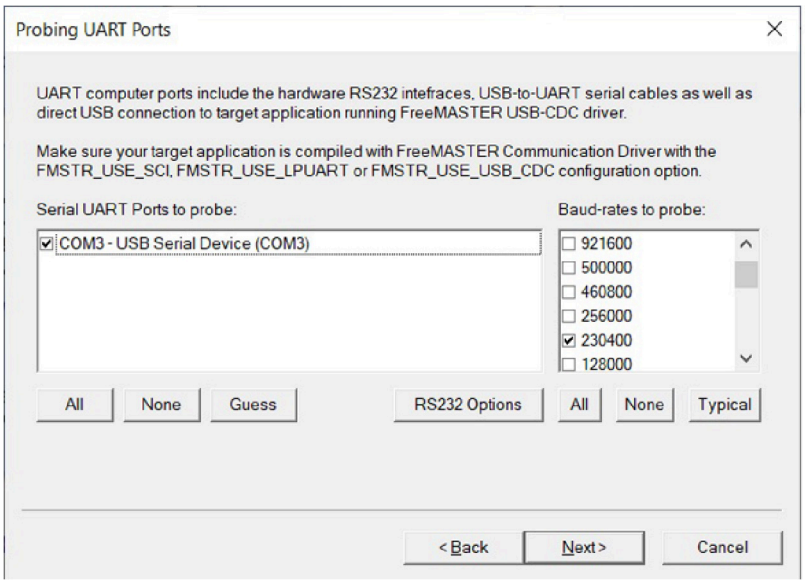


Figure 3. Probing UART ports panel

5. FreeMASTER detects the board connection and asks to confirm the detected settings. Confirm by selecting “Yes” and click “Finish”.

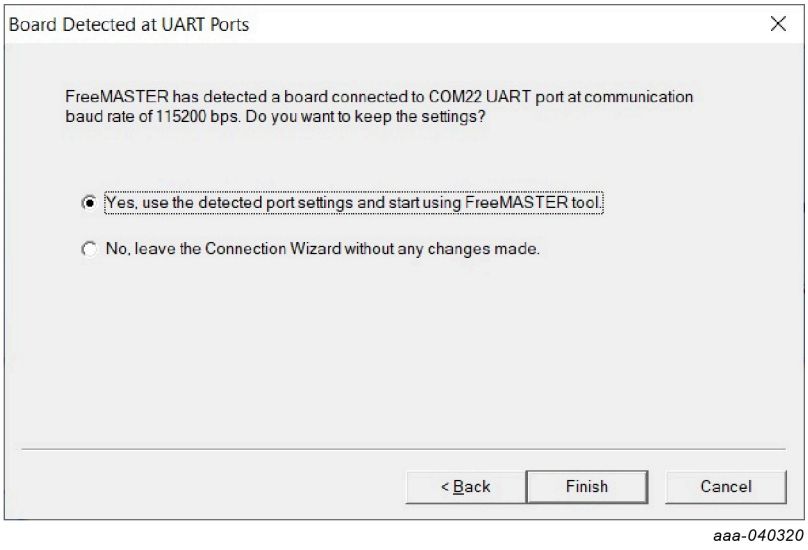
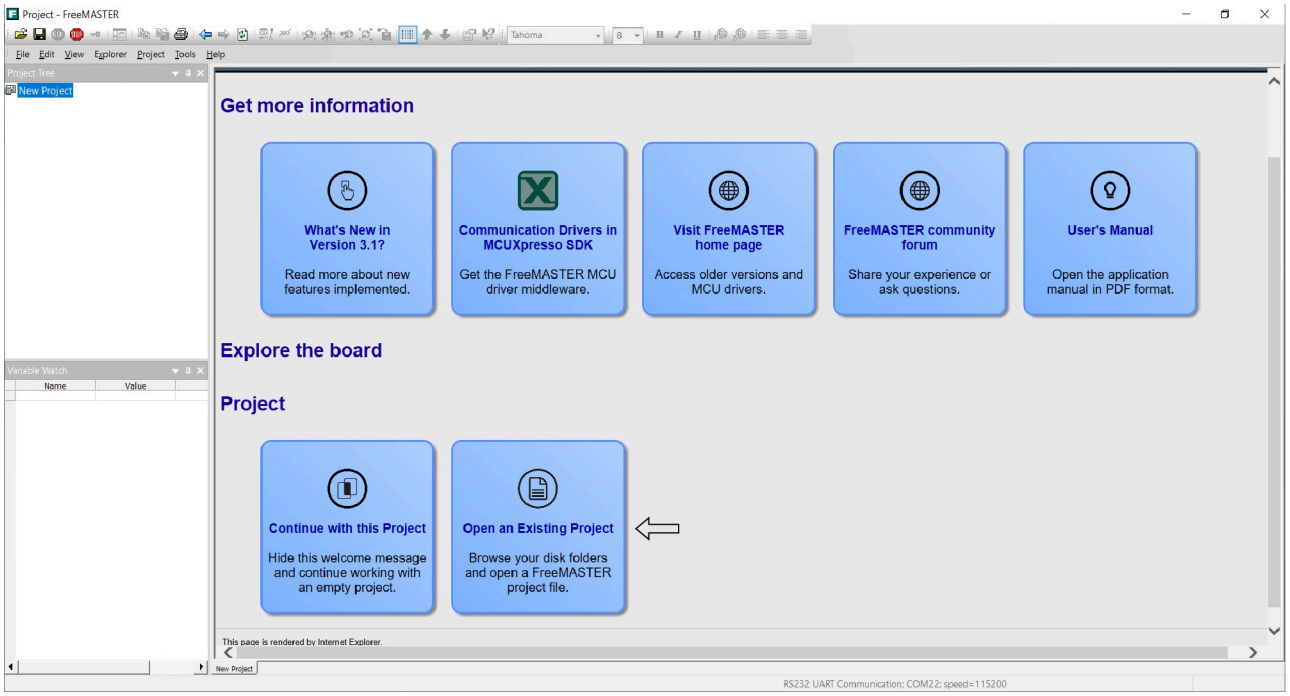


Figure 4. Board detection confirmation panel

6. FreeMASTER opens an option to “Open an Existing Project”. Select the option.

All information provided in this document is subject to legal disclaimers.

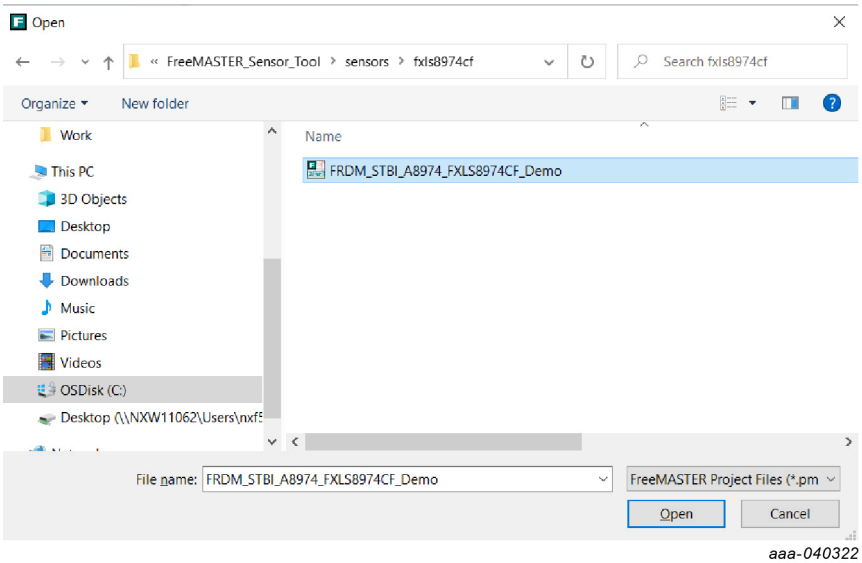
© 2023 NXP B.V. All rights reserved.



aaa-040321

Figure 5. Open existing project dialog

7. Browse to “<FreeMASTER sensor tool directory>/sensors/fxls8974cf” folder and select “FRDM_STBI_A8974_FXLS8974CF_Demo” Sensor demo project. Click “Open”.



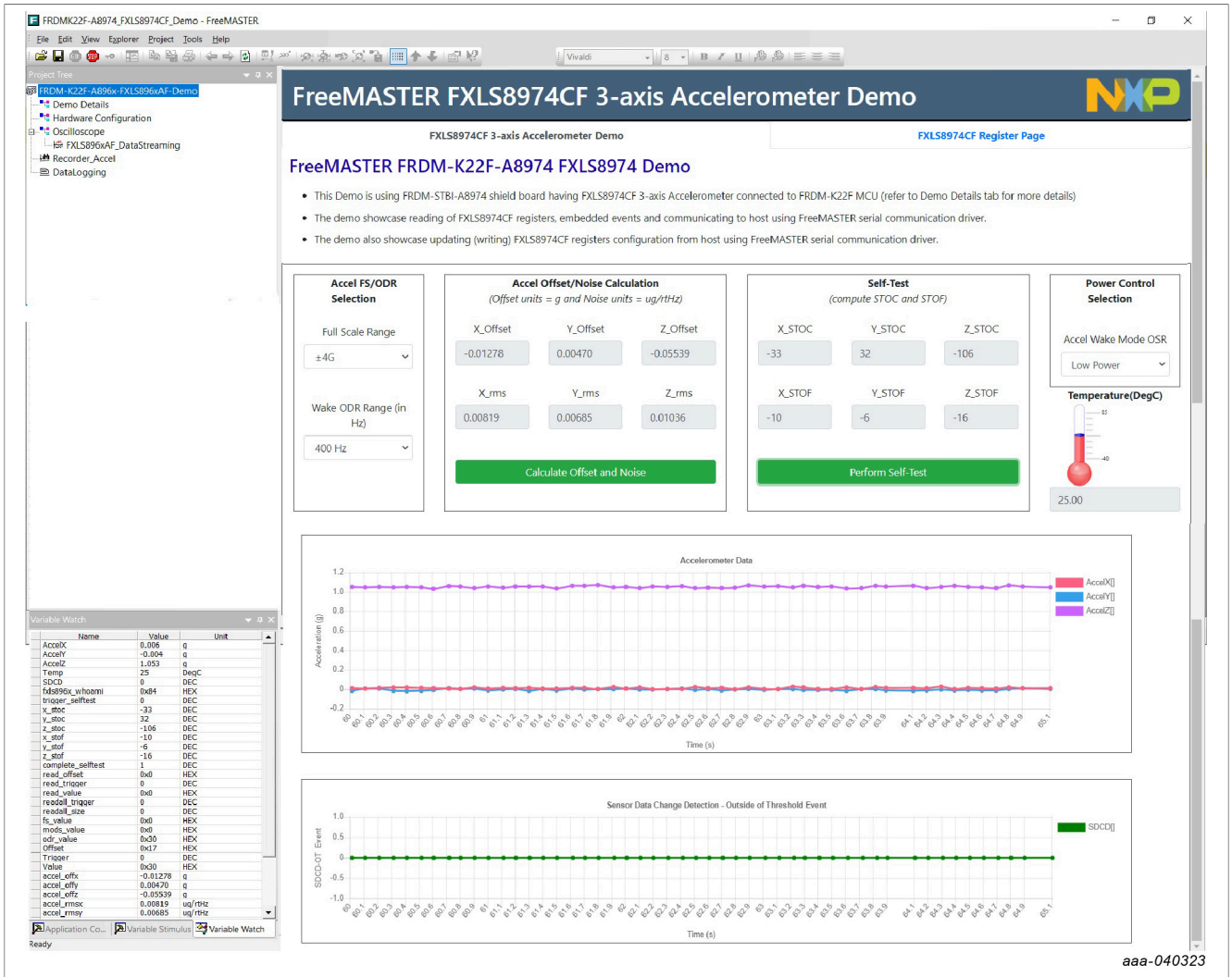
aaa-040322

Figure 6. Open sensor demonstration project

8. FreeMASTER launches the FXLS8974CF sensor demo and opens the control page. On this page, the user views sensor power control selections, FS/ODR selections, Offset/Noise measurement selection, along with time-series charts for accelerometer samples.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.



aaa-040323

Figure 7. FreeMASTER sensor demonstration: Control Page

9. To access the FXLS8974CF register set click “FXLS8974CF Register Page” tab. Click “Read All” to view instantaneous values of the FXLS8974CF sensor registers in real-time.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

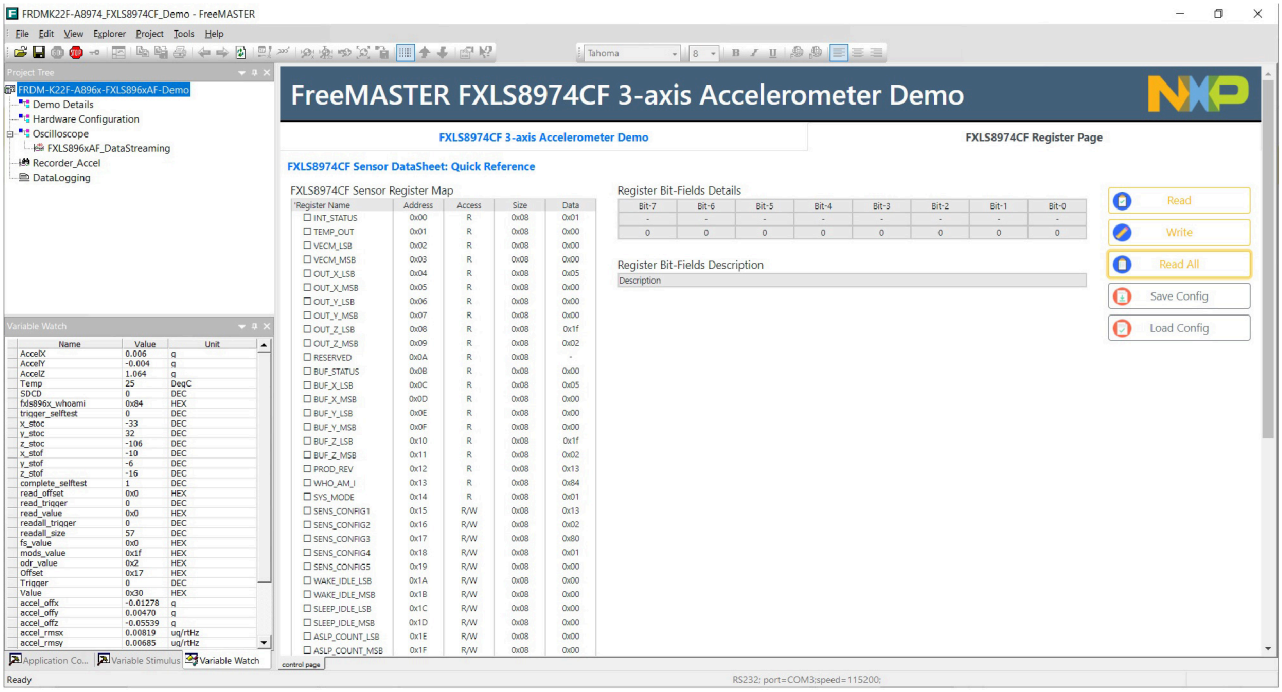


Figure 8. FreeMASTER sensor demonstration: Register Page

10. Users can select specific registers and perform single register read or write actions in real-time. For a chosen sensor register with read/write access, users can toggle bit-fields to change the register value. Users can click “Write” to perform register write operation and/or perform register read by clicking “Read”.

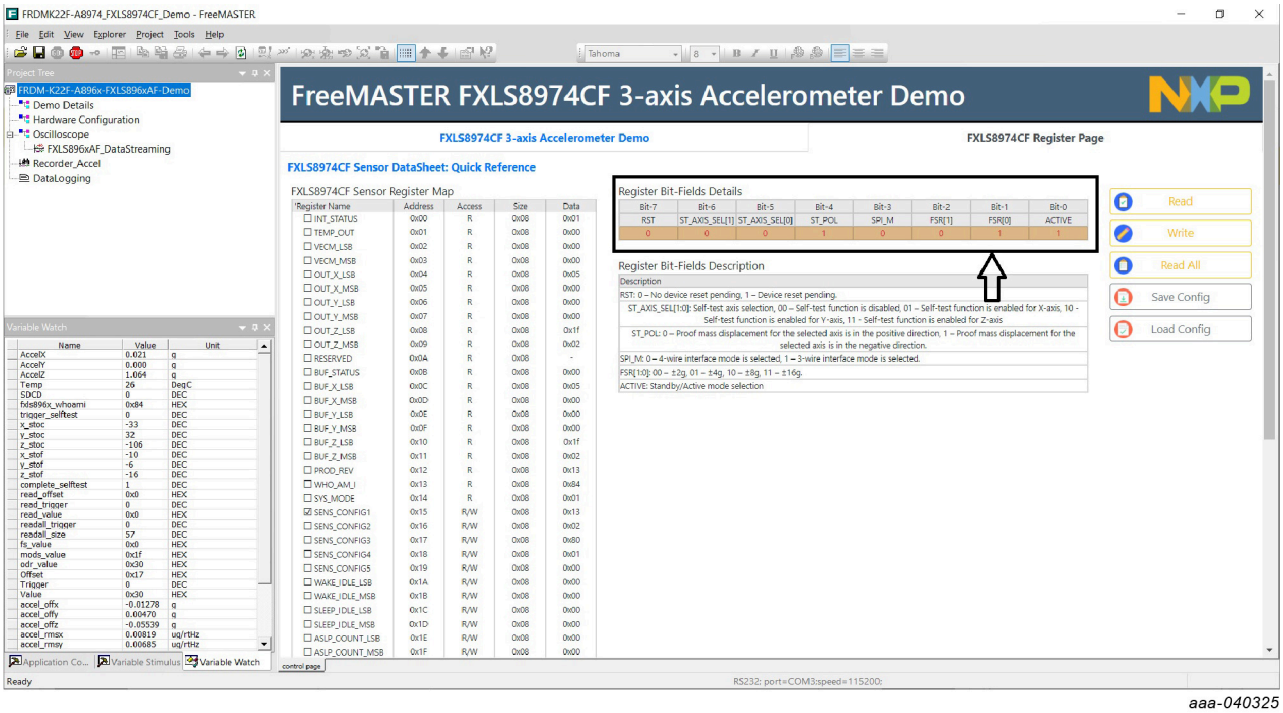


Figure 9. User controlled interactive register editing panel

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

5 FreeMASTER sensor tool development flow

5.1 Creating sensor embedded application

1. Migrate the ISSDK-based embedded application to use FreeMASTER drivers and FreeMASTER Proprietary Comm Protocol to communicate to host side.
2. Figure 10 shows the sensor embedded application development flow:

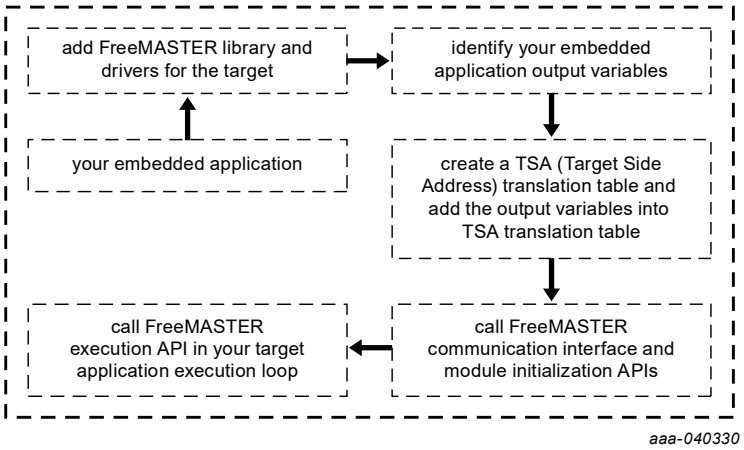


Figure 10. Sensor embedded application development flow

3. Start with the ISSDK out-of-box embedded application project for the supported sensor toolbox evaluation kit.
4. Add FreeMASTER drivers for the desired platform into the embedded project.
5. Identify the target output variables to control, monitor, and visualize using the sensor host application.
6. Create a TSA (Target Side Addressable) translation table using FreeMASTER macros and add user identified output variables into TSA.
7. Call FreeMASTER communication to initialize the API and module initialize the API as part of the application initializations.
8. Call the FreeMASTER execution API in the application execution loop.
9. With these changes completed in the embedded application, the FreeMASTER sensor embedded application is ready.
10. Rebuild the embedded project and program the FRDM-K22F-A8974 evaluation kit.

5.2 Creating sensor host gui application

1. Create the sensor host GUI application using out-of-box demo template.
2. Figure 11 shows the sensor host GUI application development flow:

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

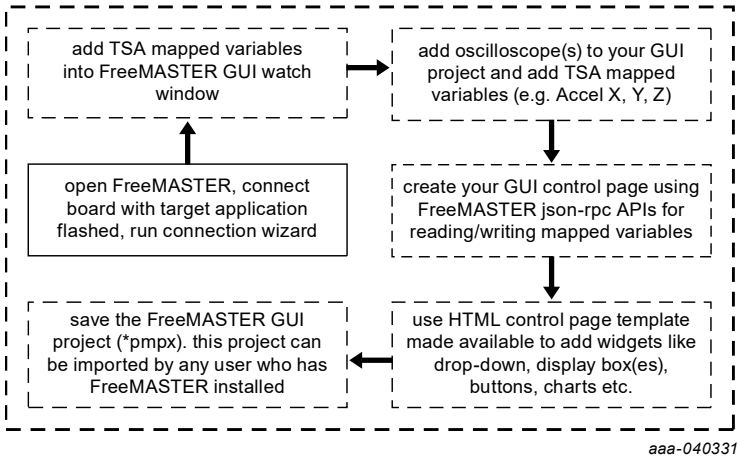


Figure 11. GUI Application flow

3. Launch the FreeMASTER Windows application. Connect the FRDM-K22F-A8974 evaluation kit to the development PC using a USB cable connected between the OpenSDA USB port on the board and a USB connector on the PC.
4. Run the connection wizard on the FreeMASTER application. The connection wizard detects the COM port and starts an empty project.

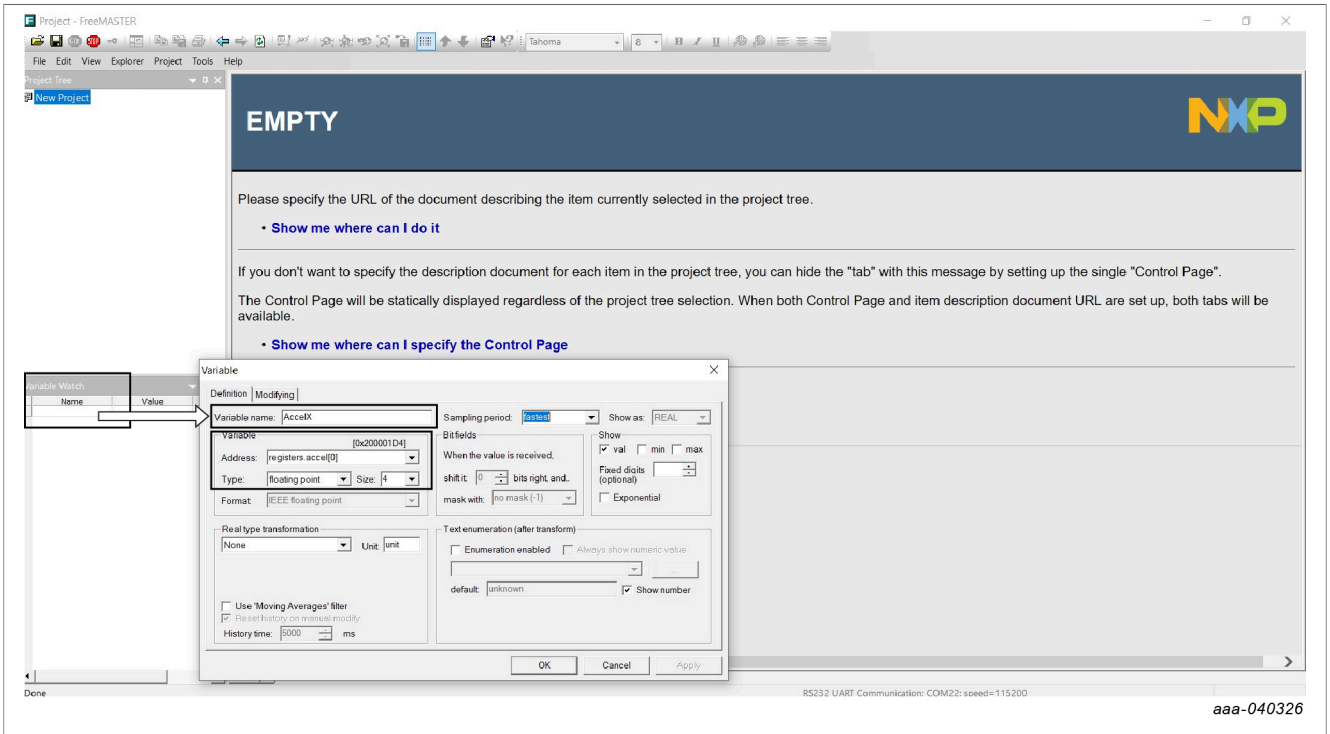


Figure 12. Create empty GUI project

5. Add TSA mapped variables into the FreeMASTER “Variable Watch” window.
   - Double-click in the “Variable Watch” window, select the variable address (TSA mapped variable in the embedded application) and assign a variable name for it.
   - Check the variable type and size.
   - Make the sampling period as “fastest”.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

- Change the variable unit field, for example, G for acceleration output.

6. Go to the “Project Tree” and rename the project name by right-clicking on the “New Project” and selecting the properties settings.
7. Right-click the project name and select “Create Oscilloscope” to add one or more oscilloscopes to the project and add watch variables (for example, Accel X):
   - Add a name for the project oscilloscope in the “Main Settings”.
   - Go to “Variables” tab and add variable (AccelX)

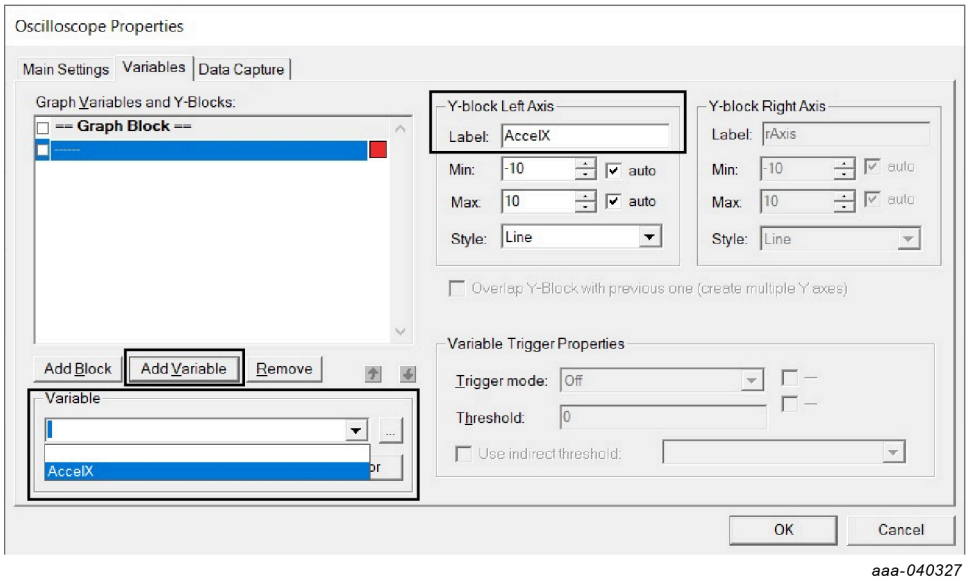
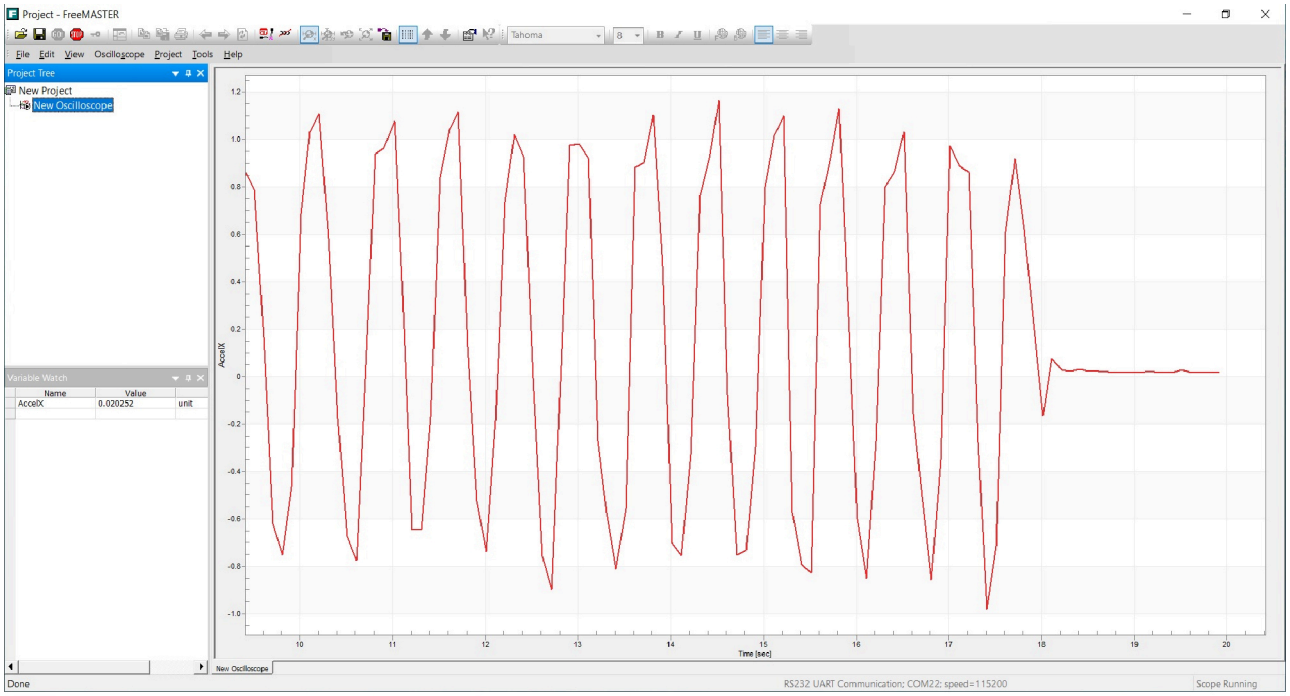


Figure 13. Add watch variable to oscilloscope

8. The watch variable is successfully added into the oscilloscope.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.



aaa-040328

Figure 14. Visualize watch variable on oscilloscope

9. Create the GUI control page using FreeMASTER json-rpc APIs for reading/writing mapped variables.
10. Use the available, out-of-box example HTML control page template to add custom widgets such as drop-down, display boxes, buttons, and charts.
11. Save the FreeMASTER GUI project (*pmpx).
12. After performing these steps, the sensor host application is ready for use. Any user with FreeMASTER installed can import this project (*pmPx).

6 References

[1] FreeMASTER Run-Time Debugging Tool, https://www.nxp.com/design/software/development-software/freemaster-run-time-debugging-tool:FREEMASTER

[2] MCUXpresso Software and Tools, https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-:MCUXPRESSO

[3] IoT Sensing Software Development Kit (ISSDK): Embedded Software Framework, https://www.nxp.com/design/software/development-software/sensor-toolbox-sensor-development-ecosystem/iot-sensing-software-development-kit-issdk-embedded-software-framework:IOT-SENSING-SDK

[4] Sensor evaluation boards, https://www.nxp.com/design/sensor-developer-resources/sensor-toolbox-sensor-development-ecosystem/evaluation-boards:SNSTOOLBOX

[5] OpenSDA Serial and Debug Adapter, https://www.nxp.com/design/microcontrollers-developer-resources/ides-for-kinetis-mcus/opensda-serial-and-debug-adapter:OPENSDA

[6] MCUXpresso Integrated Development Environment (IDE), https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE?&tab=Design_Tools_Tab

[7] FreeMASTER tool 3.x - Windows installer, FreeMASTER tool download page

[8] FreeMASTER for Embedded Applications user guide, https://www.nxp.com/docs/en/user-guide/FMSTERUG.pdf

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

7 Legal information

7.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

7.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Tables

Figures

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2023 NXP B.V. **All rights reserved.**

For more information, please visit: http://www.nxp.com

Date of release: 6 June 2023
Document identifier: UM11555