

# FREEMASTER QUICKSTART: DEVELOPMENT AND DEBUG MADE EASY

Mike Cao 曹学余

GC AUTOMOTIVE FAE

[XUEYU.CAO@NXP.COM](mailto:XUEYU.CAO@NXP.COM)

+86 18616552690



SECURE CONNECTIONS  
FOR A SMARTER WORLD

NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2017 NXP B.V.  
PUBLIC



# AGENDA

- What is FreeMASTER
- How to get FreeMASTER
- FreeMASTER features
  - as a Real-Time Monitor
  - as a Control GUI
  - vs. a Debugger
- FREEMASTER Window Description



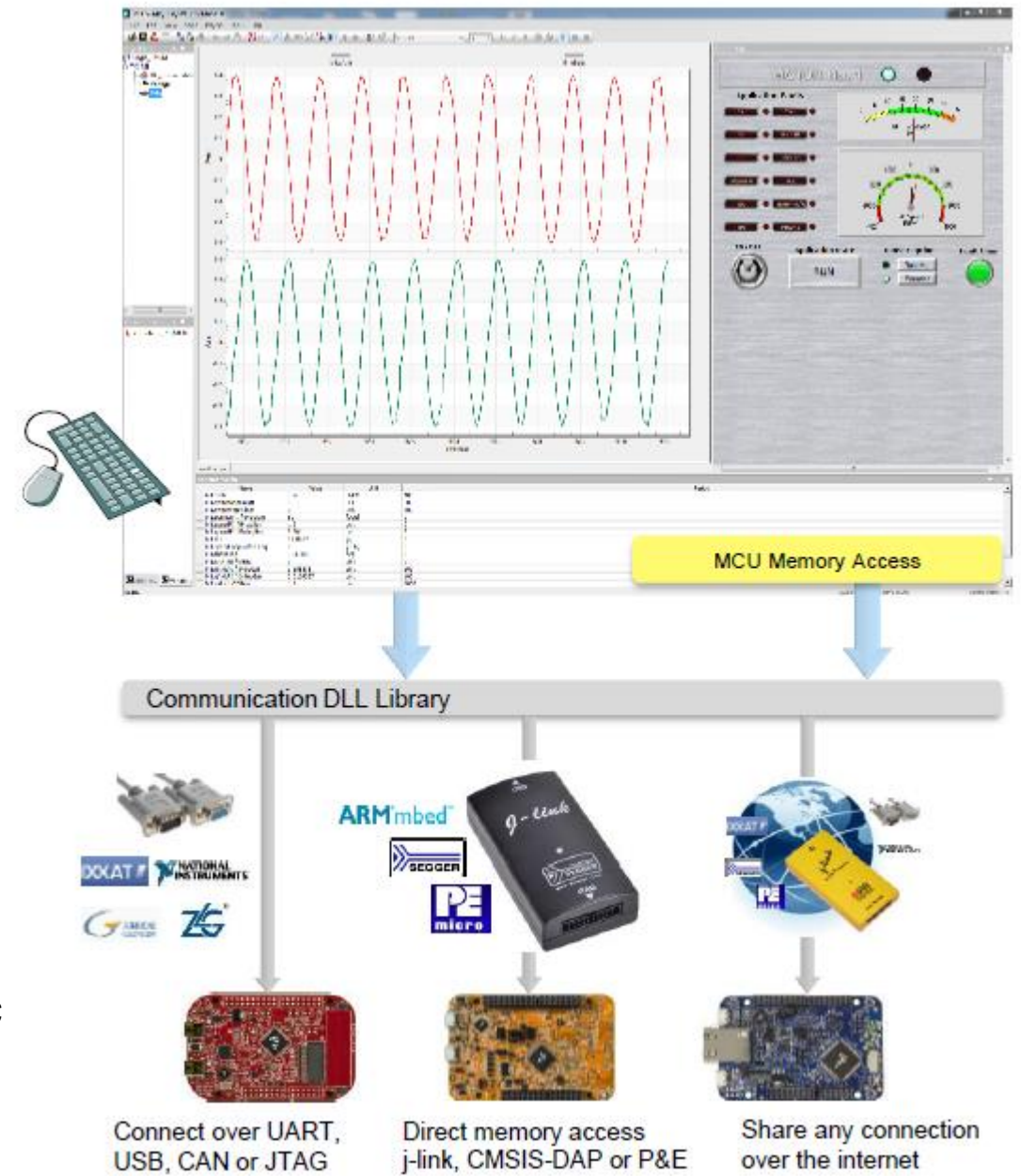
# What is **FreeMASTER**

FreeMASTER is a user-friendly **real-time debug monitor and data visualization** tool that can be used for application development and information management.

- Supports **non-intrusive monitoring of variables** on a running system.
- Display multiple variables changing over time on an **oscilloscope-like display**, or view **data in text form**.
- Supports **additional capabilities** and targets **with an on-target driver** for transmitting data from the target to the host computer.

## What do we do with FreeMASTER?

- **Connect:** to target MCU over UART, CAN, BDM, JTAG etc
- **Monitor:** read & show variables in run-time
- **Control:** set variables, send commands
- **Share:** enable Excel, Matlab or a script engine to add hardware to the control loop

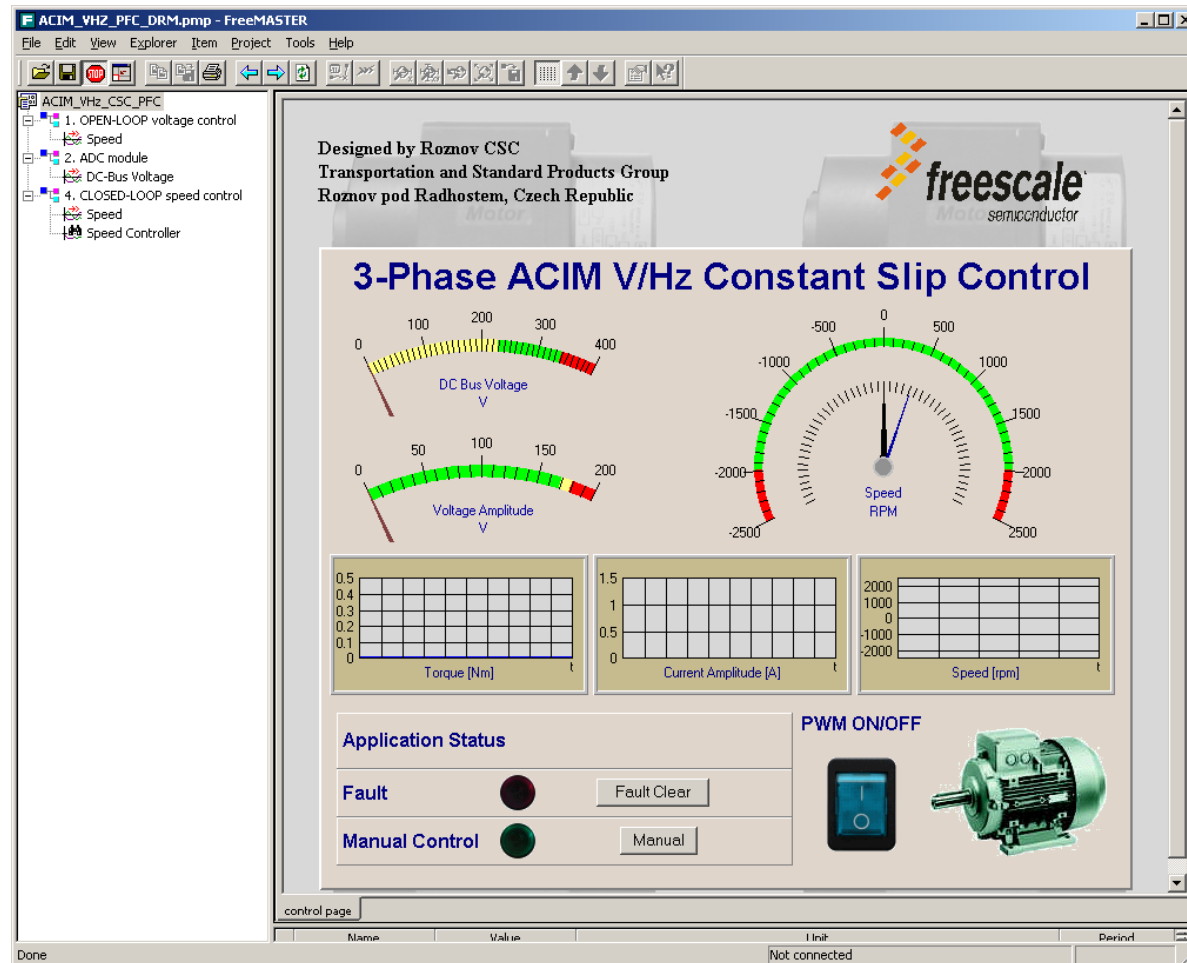


# What is FREEMASTER

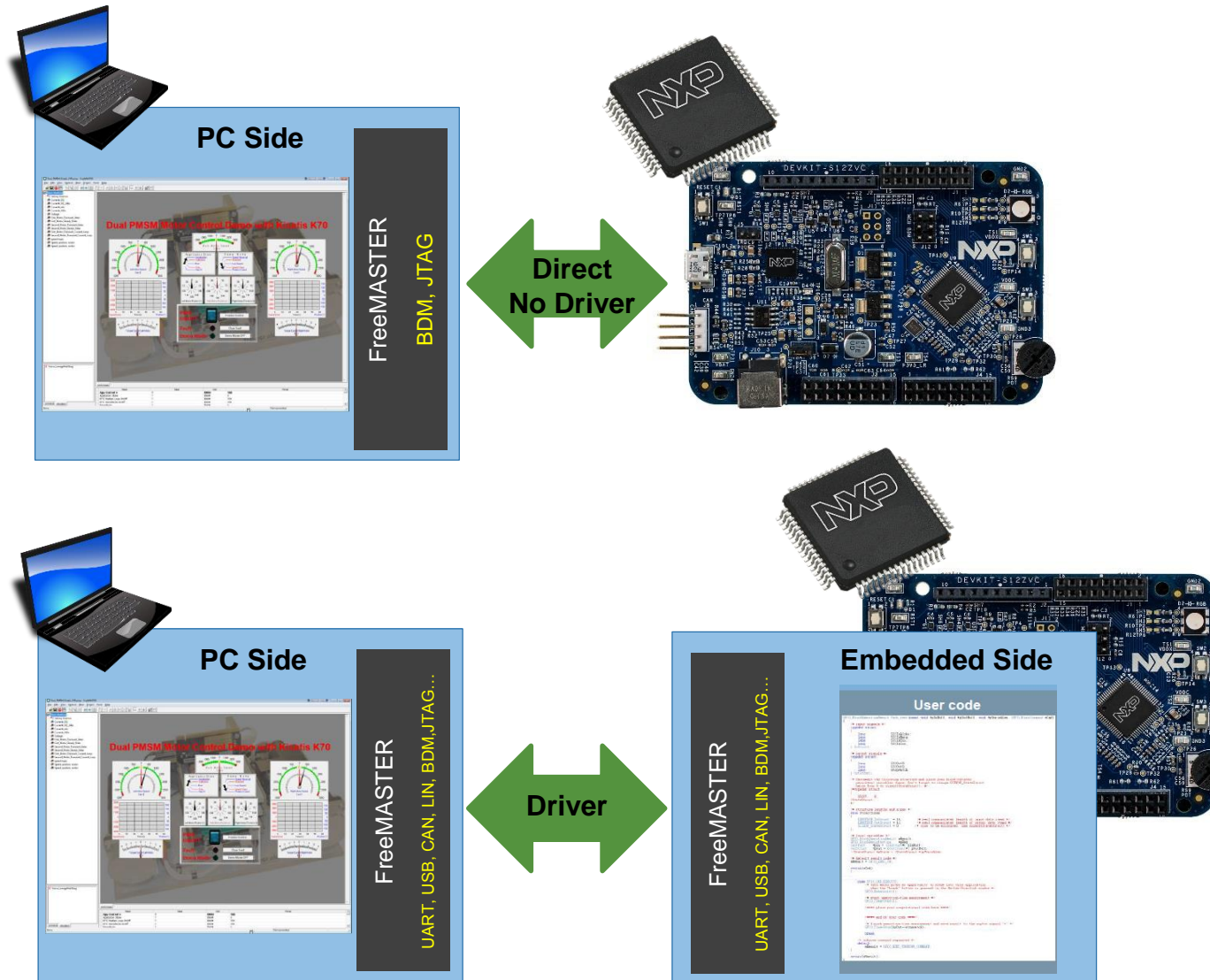
Application control  
and monitor

Live graphs,  
variable watches,  
and graphical  
control page

Real-time  
operation monitor



# What is **FreeMASTER** - Connection options



## • Supported Devices

- S08
- DSC
- ARM Cortex-M (Kinetis/S32)
- S12/S12X/S12Z(MagniV),
- MPC56xx, MPC57xx
- ColdFire V1/V2

## • Supported Interfaces

- BDM
- JTAG (Segger,PE,CMSIS DAP,etc)
- Serial
- CAN
- LIN
- USB

# What is **FREEMASTER** - Supported Devices & Interfaces

	No Driver	Target Driver Required						
MCU Families	BDM /JTAG	Packet-Driven BDM	Serial	CAN	LIN	USB	MQX IO	eOnce /JTAG
S12 MagniV® Mixed Signal, S12 and S12X MCUs	✓	✓	✓	✓	✓			
S32 MCUs based on ARM Cortex-M	✓	✓	✓	✓				
MPC56xx MCUs based on Power Architecture	✓	✓	✓	✓				
MPC57xx based on Power Architecture	✓	✓	✓	✓				
Kinetis MCUs based on ARM Cortex-M	✓	✓	✓	✓		✓		
S08 MCUs	✓	✓	✓	✓		✓		
DSC			✓	✓				✓
ColdFire MCUs	✓	✓	✓	✓		✓	✓	

## System Requirements

- Host side operating system: Windows XP to through Windows 10 32/64bit
- Required software: Internet Explorer 8 or higher installed beforehand.
- Hard drive space: 50 MB
- Other hardware requirements: Serial RS-232 port for local control or USB-to Serial converter.

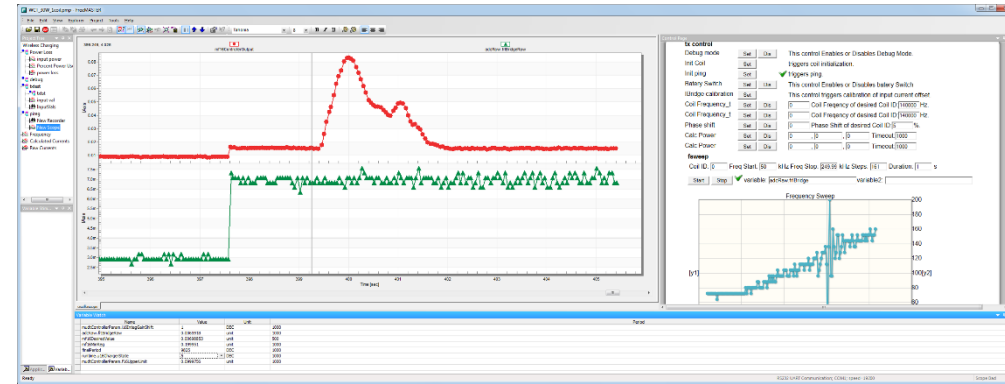
## NOTE:

If it is desired to run the debugger and **FREEMASTER** concurrently, the target driver option is required!

# What is **FREEMASTER** - Features and Usage

## Real Time Monitor

- Watching on-board variables or memory locations in various formats
- Text (name, value, min, max, enumerated labels...)
- Real-time waveform (real-time oscilloscope)
- High-speed recorded data (on-board memory oscilloscope)
- User-defined dashboard for data visualization

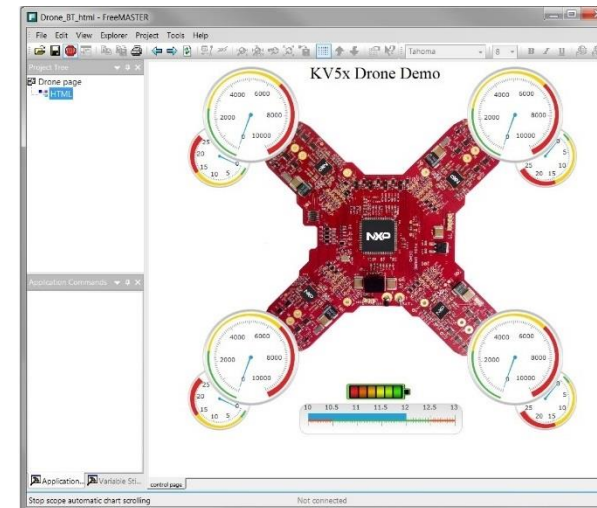


## Control Panel

- Direct setting of the variable value from the variable watch
- Time-table stimulation of the variable value
- User command/message control
- Visual Basic script or JScript-powered HTML Forms (with push buttons, indicators and sliders) or custom HTML5 gauges
- By external application like Excel, Matlab or other which support ActiveX embedding

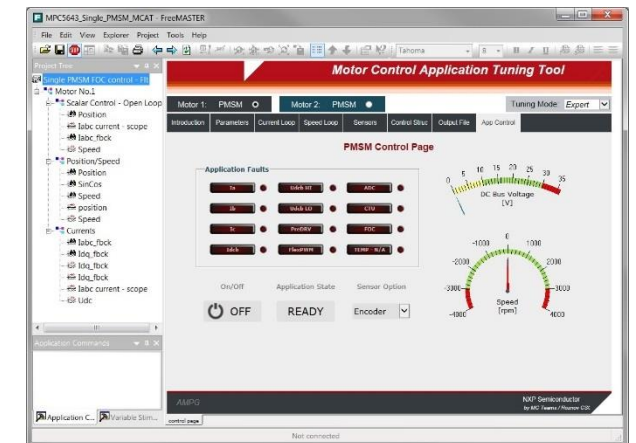
## Demonstration Platform

- You can both describe and demonstrate your embedded application by HTML pages that contain pictures, sounds, video sequences, links or any web content
- Display simultaneous real-time data monitoring
- Browse through the functional blocks of the embedded application

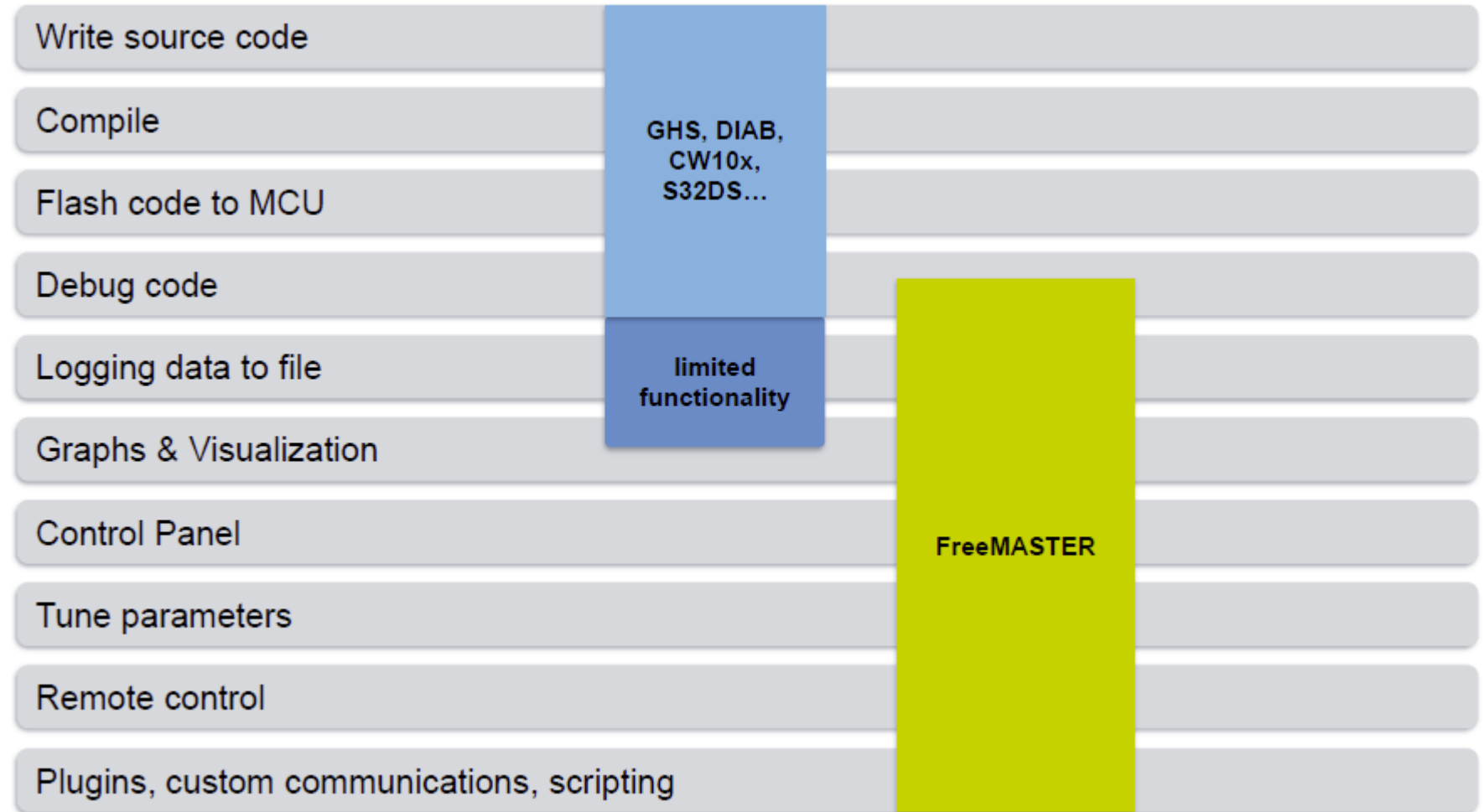


## Easy Project Deployment

- Entire project saved to a single file
- All resources/files packed in the project file
- "Demo mode" with password protection available
- New in v2.0: Project files embedded in target MCU Flash memory



# What is **FreeMASTER** - FreeMASTER vs. IDE/Debugger



# Tuning application constants with help of **FREEMASTER**

- The most challenging task for the developer is the setting of the application constants, sometimes trial-error method must be used when the system (drive) parameters are difficult to identify:
  - $P$  and  $I$  constants of the regulators
  - Filter constants
  - Constants of the position estimation algorithms
  - Tuning the merging process when switching from the open loop start-up to full sensorless mode

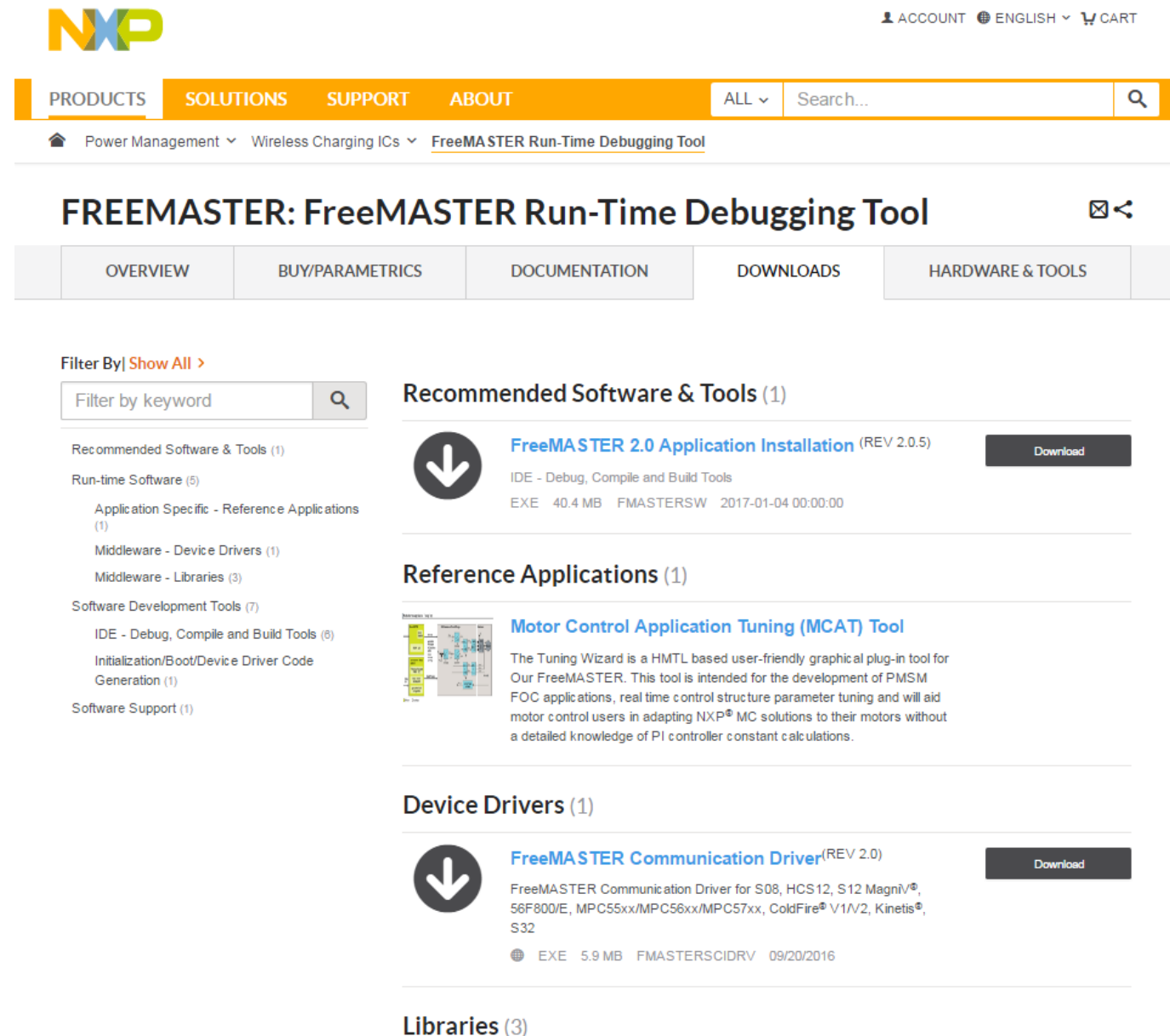
# **FreeMASTER** Highlights

- FreeMASTER helps developers to debug or tune their applications
- Replaces debugger in situations when the processor core can not be simply stopped (e.g. motor control)
- Recorder may be used to visualize transitions in near 10-us resolution
- **No EXTRA code is required on the embedded side to interface to FreeMASTER via BDM/OpenSDA/OSJTAG plug in modules.**

# How to get FreeMASTER

**FREEMASTER**

is a **FREE** download from  
[www.nxp.com/freemaster](http://www.nxp.com/freemaster)



The screenshot shows the NXP website's product page for the FreeMASTER Run-Time Debugging Tool. The page features a navigation bar with 'PRODUCTS', 'SOLUTIONS', 'SUPPORT', and 'ABOUT'. Below the navigation bar, there's a breadcrumb trail: 'Power Management > Wireless Charging ICs > FreeMASTER Run-Time Debugging Tool'. The main heading is 'FREEMASTER: FreeMASTER Run-Time Debugging Tool'. A horizontal menu contains 'OVERVIEW', 'BUY/PARAMETRICS', 'DOCUMENTATION', 'DOWNLOADS', and 'HARDWARE & TOOLS'. On the left, a 'Filter By' section lists categories like 'Recommended Software & Tools (1)', 'Run-time Software (5)', 'Software Development Tools (7)', and 'Software Support (1)'. The main content area is divided into three sections: 'Recommended Software & Tools (1)' featuring 'FreeMASTER 2.0 Application Installation (REV 2.0.5)', 'Reference Applications (1)' featuring 'Motor Control Application Tuning (MCAT) Tool', and 'Device Drivers (1)' featuring 'FreeMASTER Communication Driver (REV 2.0)'. Each item includes a download button and details like file size and date.

**PRODUCTS** SOLUTIONS SUPPORT ABOUT ALL Search...

Power Management > Wireless Charging ICs > FreeMASTER Run-Time Debugging Tool

## FREEMASTER: FreeMASTER Run-Time Debugging Tool

OVERVIEW BUY/PARAMETRICS DOCUMENTATION DOWNLOADS HARDWARE & TOOLS

Filter By **Show All** >

Filter by keyword

Recommended Software & Tools (1)

Run-time Software (5)

- Application Specific - Reference Applications (1)
- Middleware - Device Drivers (1)
- Middleware - Libraries (3)

Software Development Tools (7)

- IDE - Debug, Compile and Build Tools (6)
- Initialization/Boot/Device Driver Code Generation (1)

Software Support (1)

### Recommended Software & Tools (1)

**FreeMASTER 2.0 Application Installation** (REV 2.0.5) [Download](#)

IDE - Debug, Compile and Build Tools  
EXE 40.4 MB FMASTERSW 2017-01-04 00:00:00

### Reference Applications (1)

**Motor Control Application Tuning (MCAT) Tool**

The Tuning Wizard is a HTML based user-friendly graphical plug-in tool for Our FreeMASTER. This tool is intended for the development of PMSM FOC applications, real time control structure parameter tuning and will aid motor control users in adapting NXP® MC solutions to their motors without a detailed knowledge of PI controller constant calculations.

### Device Drivers (1)

**FreeMASTER Communication Driver** (REV 2.0) [Download](#)

FreeMASTER Communication Driver for S08, HCS12, S12 MagniV®, 56F800/E, MPC55xx/MPC56xx/MPC57xx, ColdFire® V1/V2, Kinetis®, S32  
EXE 5.9 MB FMASTERSCIDRV 09/20/2016

### Libraries (3)



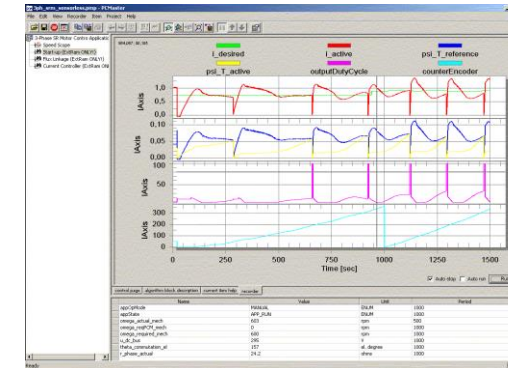
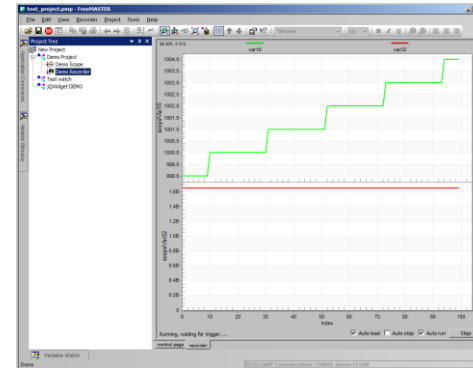
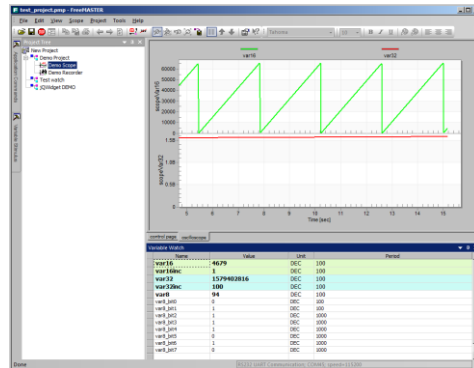
# 01.

## FreeMASTER Overview

# FreeMASTER as a Real-Time Monitor

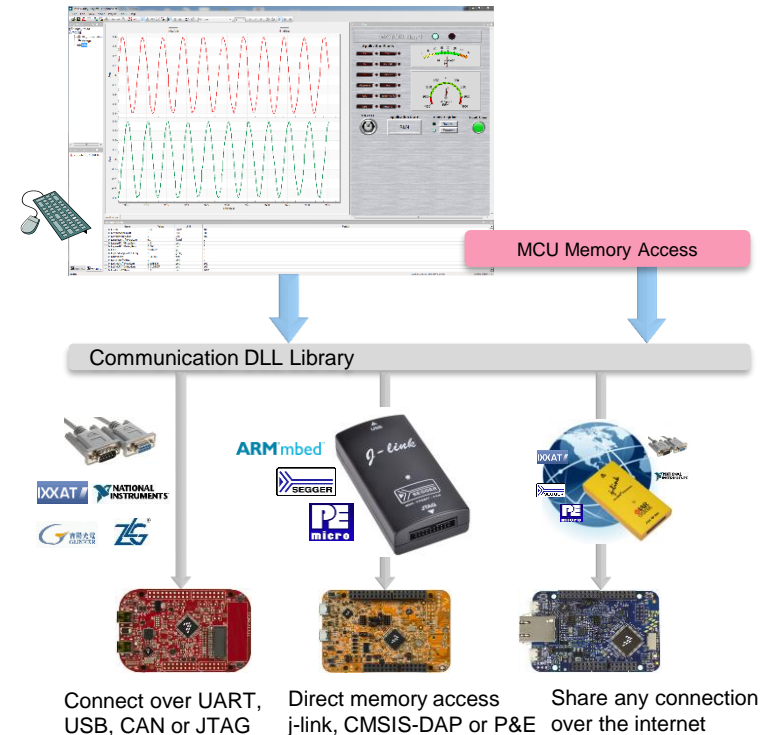
## FreeMASTER can Real-time Monitor

- Internal variables
- Processes & algorithms
- Application states



# FreeMASTER as a Real-Time Monitor

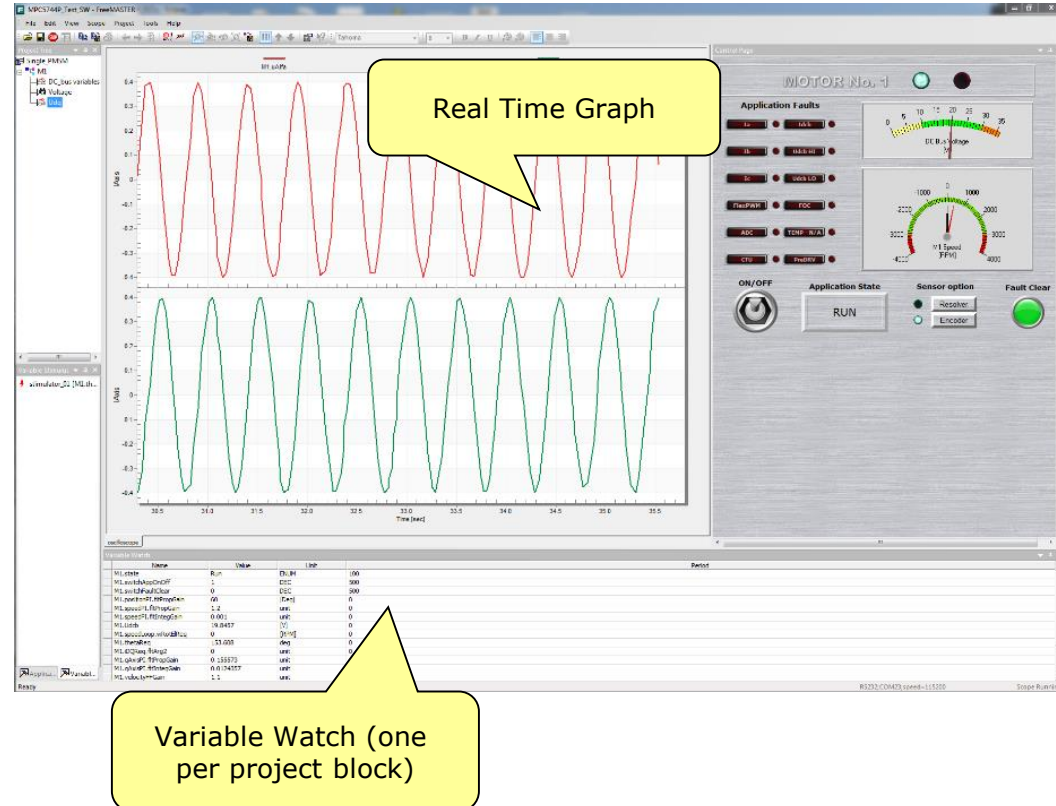
- PC Host Connects to an embedded application over unified DLL library
  - SCI, UART
  - USB-CDC - Kinetis, ColdFire V2
  - CAN - msCAN, FlexCAN with PC interface from IXXAT, Vector, NI, Glinker, ZLG
  - JTAG/EOnCE (56F8xxx only)
  - BDM - Kinetis, PowerPC, ColdFire, HCS with Segger, P&E Micro, CMSIS-DAP, iSystem, ...
  - Any of the above remotely over the IP network
- Enables access to application memory
  - Parses ELF application executable file
  - Parses DWARF debugging information in the ELF file
  - Knows addresses of global and static C-variables
  - Knows variable sizes, structure types, array dimensions etc.



# FreeMASTER as a Real-Time Monitor

Display the variable values in various formats:

- **Text**, tabular grid
  - variable name
  - numeric value
  - peak detector
  - number-to-text enumeration
- **Real-time waveforms**
  - up to 8 variables simultaneously in an oscilloscope-like graph
- **High-speed recorded data**
  - up to 8 variables in on-board memory **transient recorder**



# FreeMASTER as a Real-Time Monitor

- **Variable Transformations**

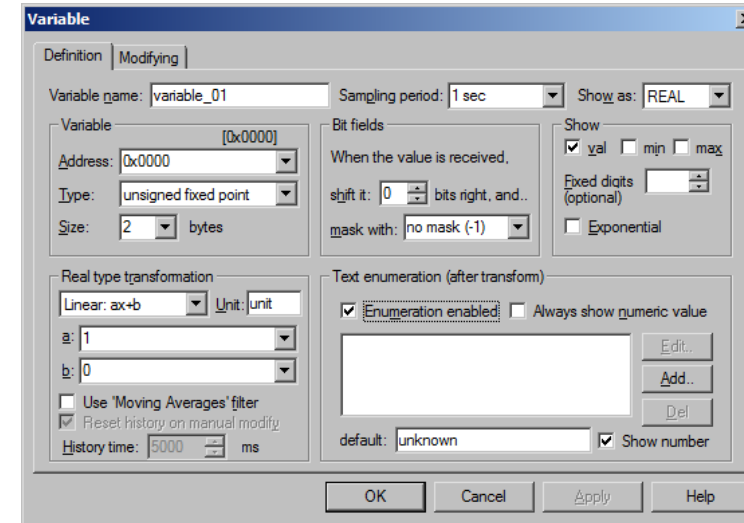
- Value can be transformed to custom units
- Transformations may reference other variable values
- Inverse-transformation applied when writing a new value to the variable

- **Ability to Protect Memory Regions (TSA)**

- Describing variables visible to FreeMASTER
- Declaring variables as read-write to read-only for FreeMASTER - the access is guarded by the embedded-side driver

- **Application Commands**

- Command code and parameters are delivered to an application for arbitrary processing
- After processed (asynchronously to a command delivery) the command result code is returned to the PC
- Legacy feature, not used in today's applications (requires target-side driver)



# FreeMASTER as a Real-Time Monitor

## Anatomy of the main window

The screenshot displays the FreeMASTER software interface for a project named 'test\_project.pmp'. The interface is divided into several sections:

- Project Tree:** Located on the left, it shows a hierarchical view of the project components, including 'New Project', 'Demo Project', 'Demo Scope', 'Demo Recorder', 'Test watch', 'jqWidget DEMO', 'Envato DEMO', and 'Test page reload'.
- Main pane:** The central area displays two graphs. The top graph, labeled 'var16' and 'var16inc', shows a sawtooth waveform with a red line indicating a threshold. The bottom graph, labeled 'scopeVar32', shows a blue line representing a signal over time. The x-axis for both graphs is 'Time [sec]' ranging from 3 to 12.
- Variable Watch:** A table at the bottom right lists monitored variables and their current values.
- Application Commands and Variable Stimulus:** These are located at the bottom left of the interface.

**Project Tree:**

- Block folder items (also root item)
- Scope graph items
- Recorder graph items
- Shortcuts

**Application Commands**

- List of commands to send

**Variable Stimulus**

- Stimulator tables to run

**Main pane:**

- Control page (if not floating)
- HTML content for "block" items
- HTML content for "graph" items
- Scope or Recorder Graphs

**Variable Watch:**

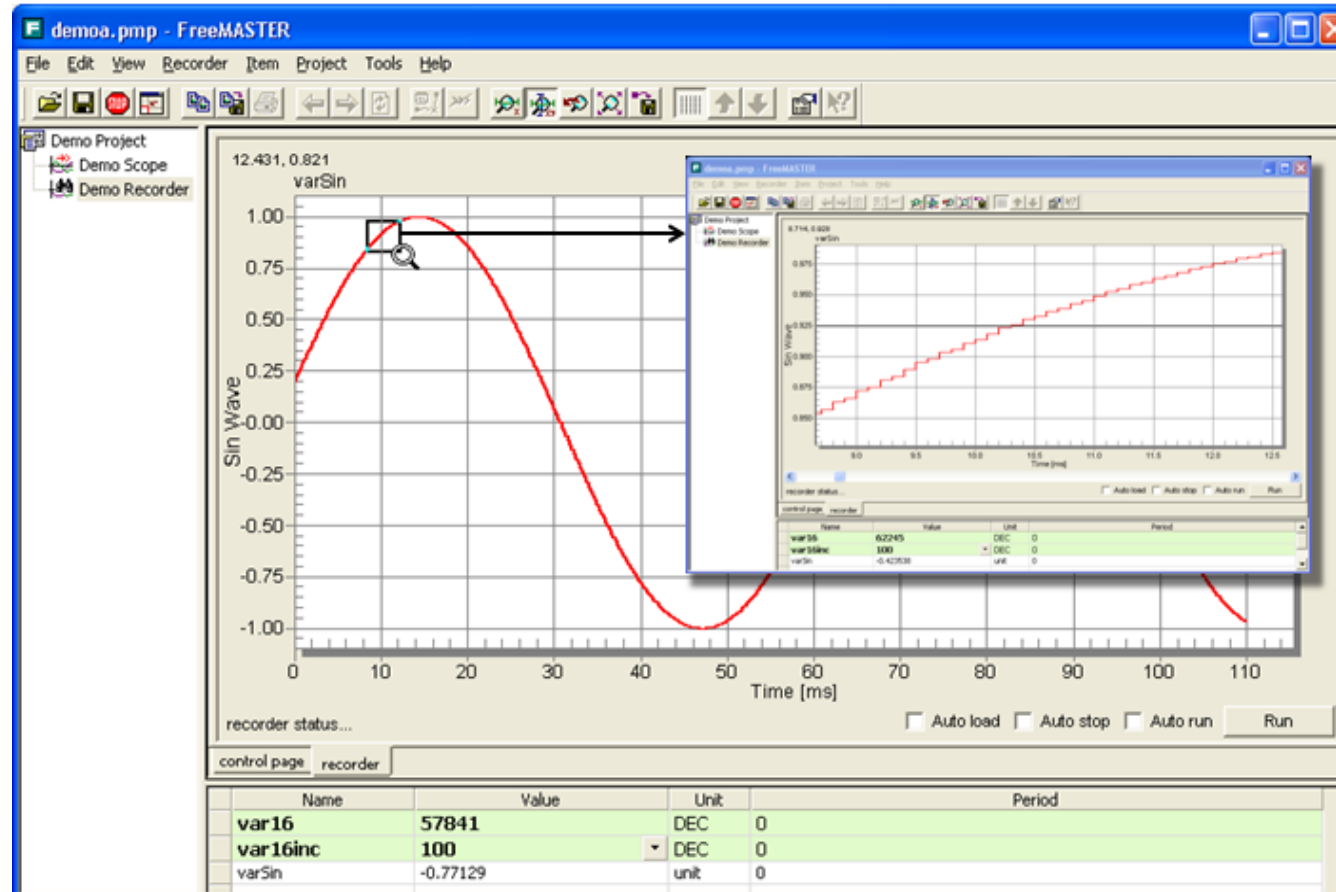
- Variable list assigned to "block" tree item

Name	Value	Unit	Period
var16	59271	DEC	100
var16inc	1	DEC	100
var32	888237204	DEC	100
var32inc	100	DEC	100
var8	79	DEC	100
var8_bit0	1	DEC	100
var8_bit1	1	DEC	100

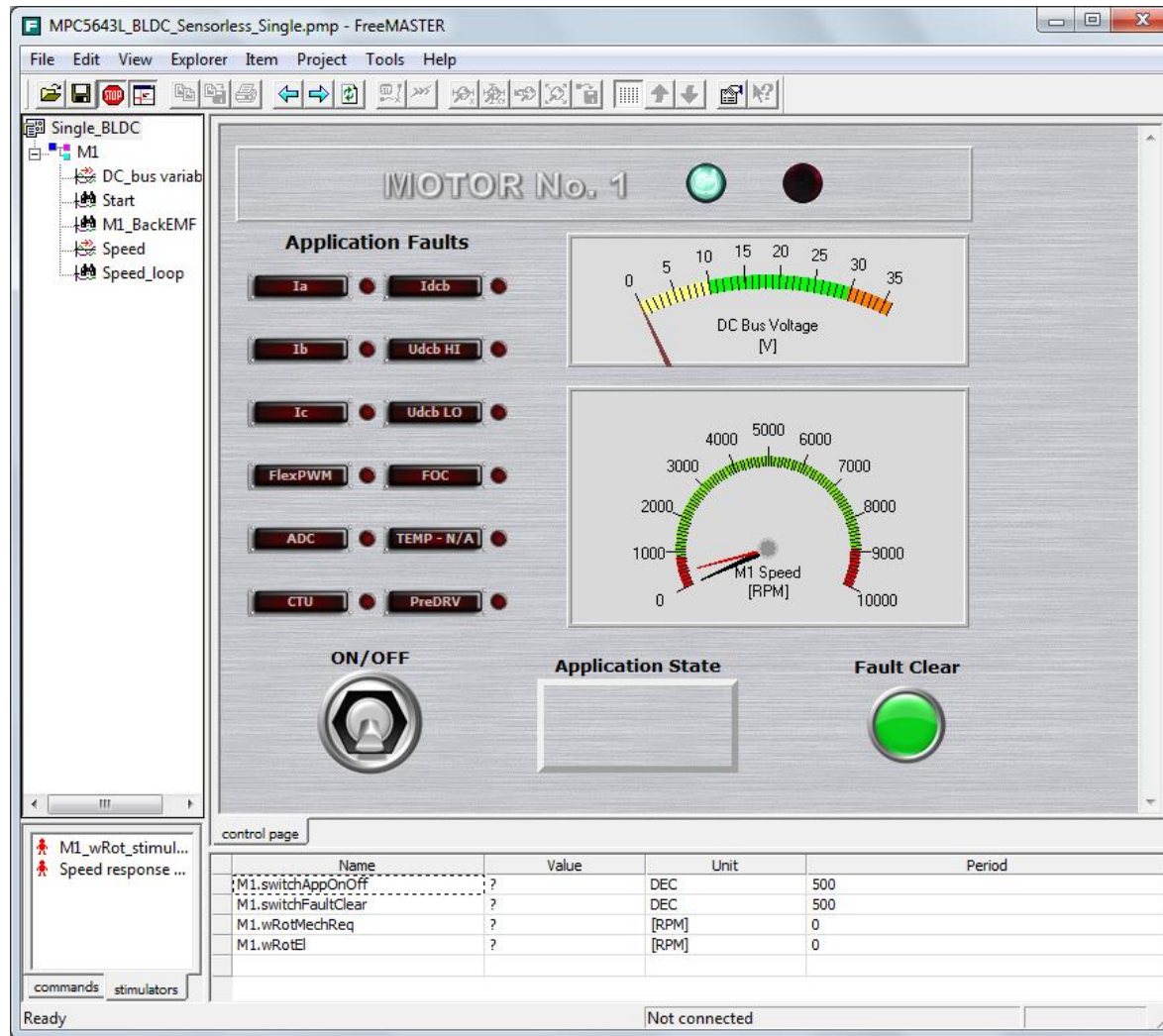
Done RS232 UART Communication; COM45; speed=115200 Scope Running

# FreeMASTER as a Real-Time Monitor

- **Establish a Data Trace on Target**
  - Set up buffer (up to 64KB), sampling rate and trigger



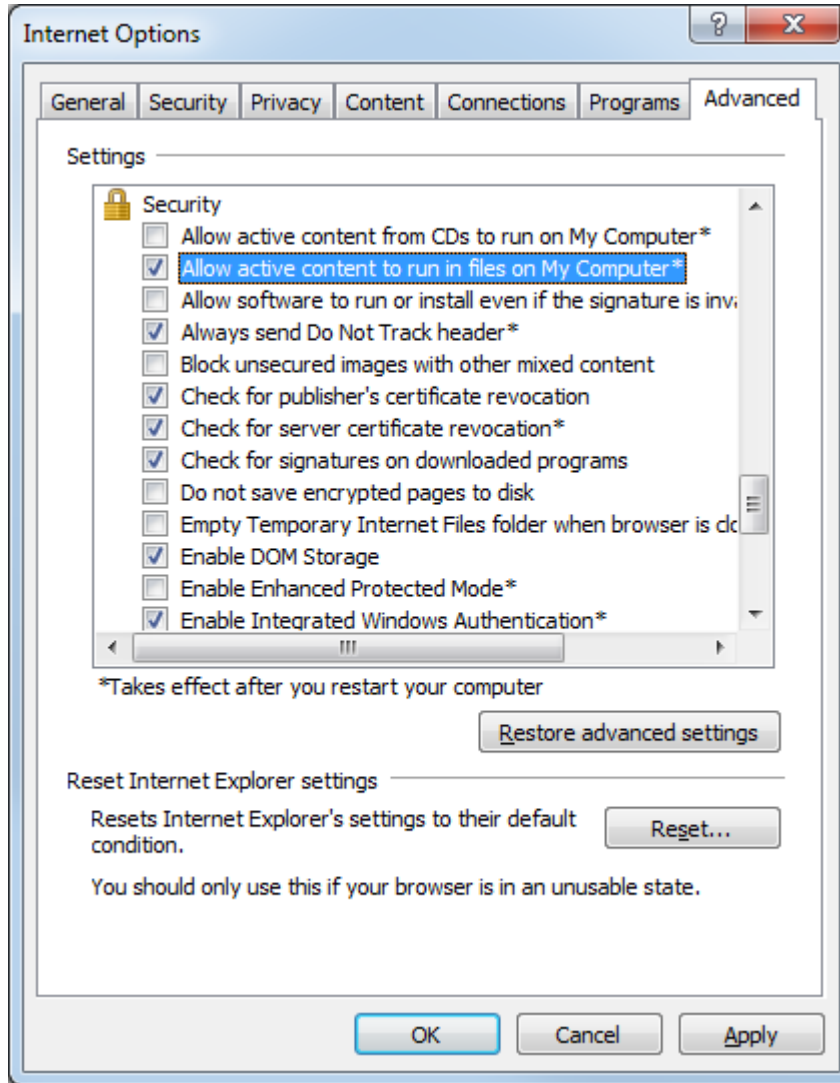
# FreeMASTER as a Real-Time Monitor



The HTML-based data visualization area. The user can provide any collection of ActiveX-based instrumentation to create custom visual dashboards as complex or elegant as desired.

The data visualization area may also be used to display arbitrary information, such as presentations, help files and fact sheets.

# FreeMASTER as a Real-Time Monitor



In order to allow the ActiveX – based instrumentation to run it may be necessary to set your Internet options to allow the active content to run.

# FreeMASTER as a Real-Time Monitor: Demo Mode

- To prevent modification, the project's author can lock the project against changes by switching it into the *Demo Mode*.
- An important part of the FreeMASTER's capabilities is the demonstration and description of the target board application. It is essential that the demonstration project, once prepared, is not accidentally modified.
- In the Demo Mode, the user cannot change the *Project Tree item properties*, *cannot add or remove the tree items*, and *cannot change any project options*.

# FreeMASTER as a Real-Time Monitor

## Steps to integrate FreeMASTER in your Application

- Include the files under the **FreeMASTER Serial Communication Vxx\src\_common** in your application code project with no changes.
- One file changed in **FreeMASTER Serial Communication V1.6\src\_platforms\MPC56xx** directory:
  - renamed freemaster\_cfg.h.example to freemaster\_cgh.h
  - Configure freemaster by changing macro definitions
- Addition to main.c
  - Add function call FMSTR\_Init() after system init
  - Add function call FMSTR\_Poll(); in main loop
- To build with FreeMASTER support for MPC56xx, include all files under **FreeMASTER Serial Communication V1.6\src\_platforms\MPC56xx** and **FreeMASTER Serial Communication V1.6\src\_platforms\MPC56xx** directories.

# FreeMASTER as a Real-Time Monitor

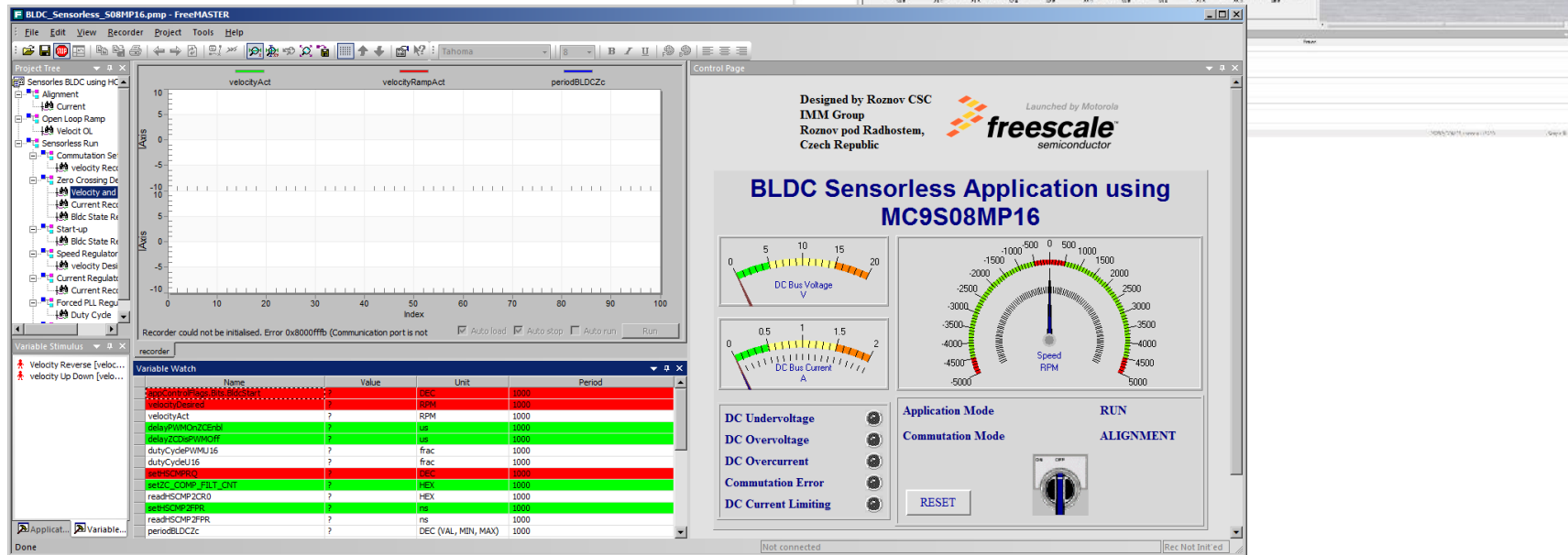
## Highlights

- Access to target variables, symbols and data types
- Safe access over UART, CAN or USB with target-side driver
- Transparent access over BDM (no target-side driver needed)
- Addresses parsed from ELF file or provided by target (TSA)
- Fine tuning parameters or direct control via variable modifications
- Scope graphs with real time data in [ms] resolution
- Recorder visualization transitions close to 10[us] resolution

# FreeMASTER as a Control GUI

## What the FreeMASTER Control GUI can do:

- rendering HTML-encoded GUI
- scriptable in JScript or VBScript
- script access to target memory



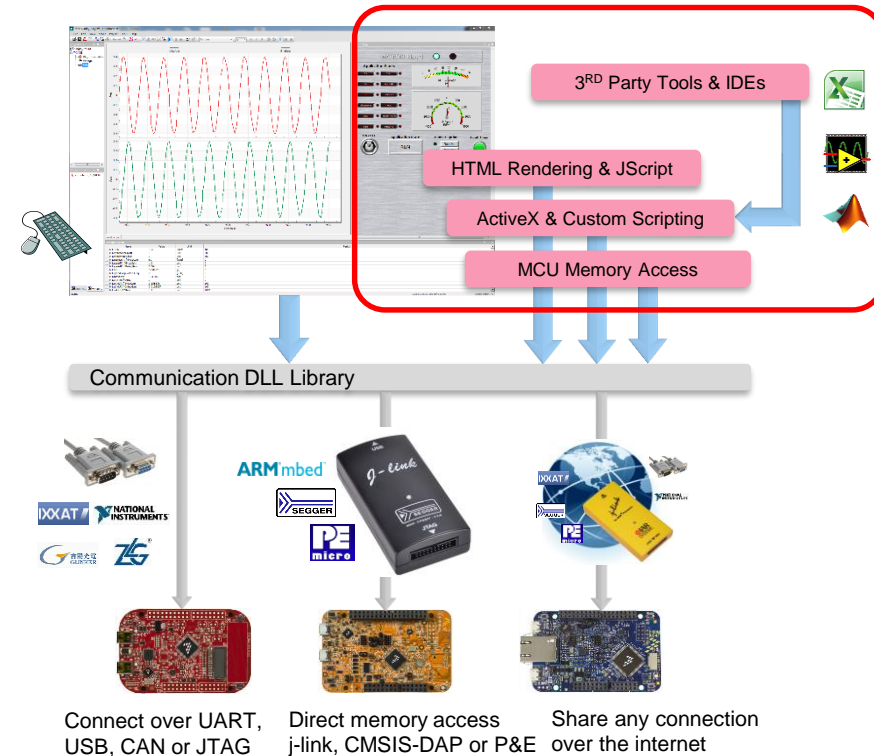
# FreeMASTER as a Control GUI

## Variable access and modification

- Manually in the Watch pane
- Time-tables & stimuli modification
- Script-based read/write directly from GUI
  - mouse-clicks and keyboard control
  - push buttons and forms
  - sliders, gauges or other ActiveX/HTML5 widgets
  - custom intelligence and control algorithms
- ActiveX clients external to FreeMASTER
  - Excel or Matlab – typical programmable clients
  - FreeMASTER enables HW-in-loop simulations
- Works over all communication interfaces

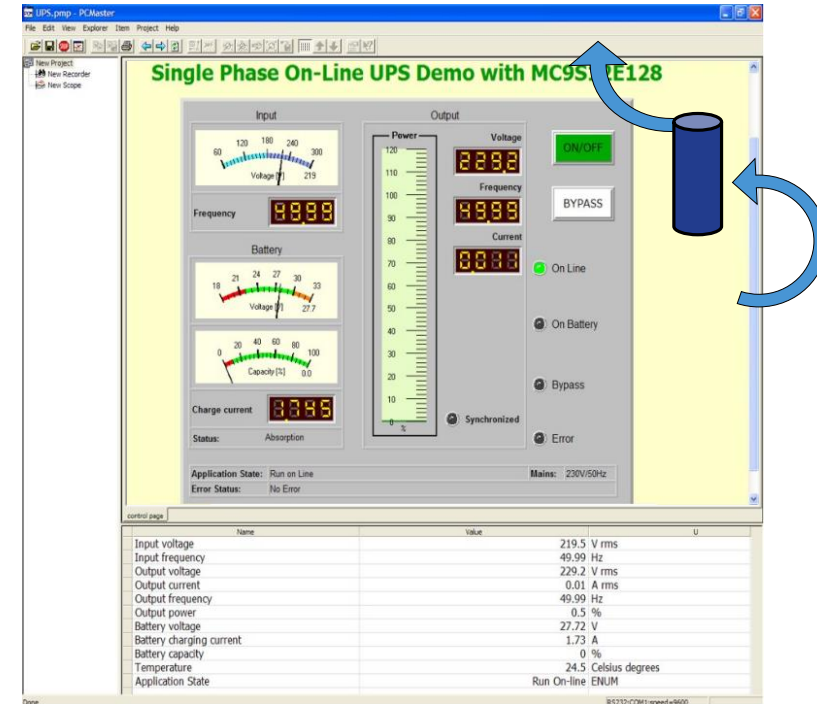
## Sending Application Commands

- “Traditional” control approach
- Not recommended as it is limited to systems with target-side agents (UART & CAN).



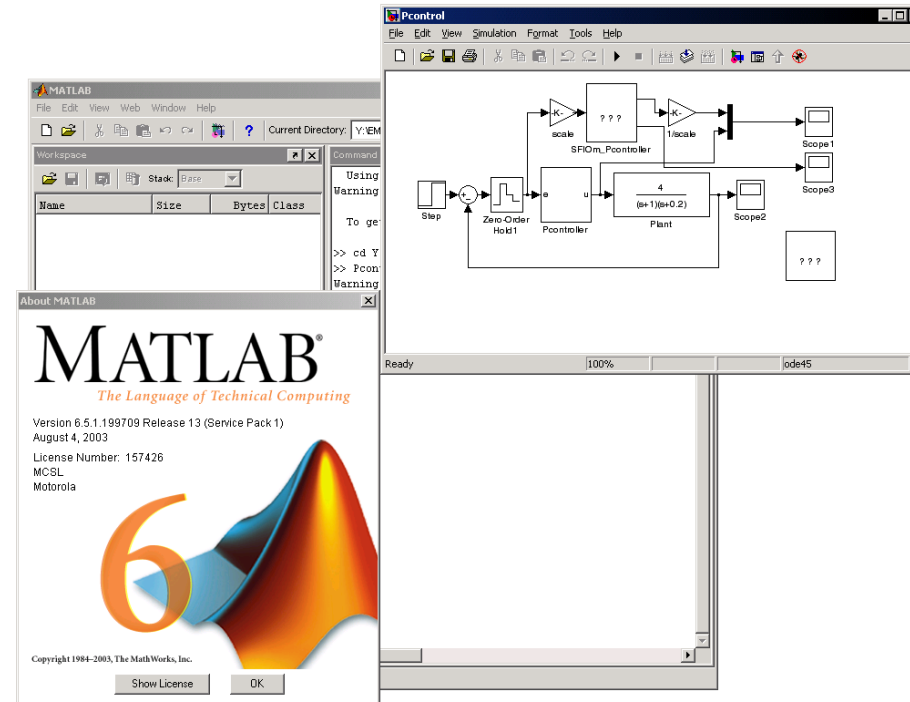
# FreeMASTER as a Control GUI

- Scripting in FreeMASTER
  - HTML pages are displayed directly in the FreeMASTER window
  - InternetExplorer v10 used as the rendering engine
  - HTML may contain scripts and ActiveX objects
- FreeMASTER invisible ActiveX object
  - Script accesses the FreeMASTER functionality through this object
  - Variable access
  - Direct memory access
  - Stimulator access
  - Application Commands
  - Recorder Data
  - Symbol and data type information



# FreeMASTER as a Control GUI

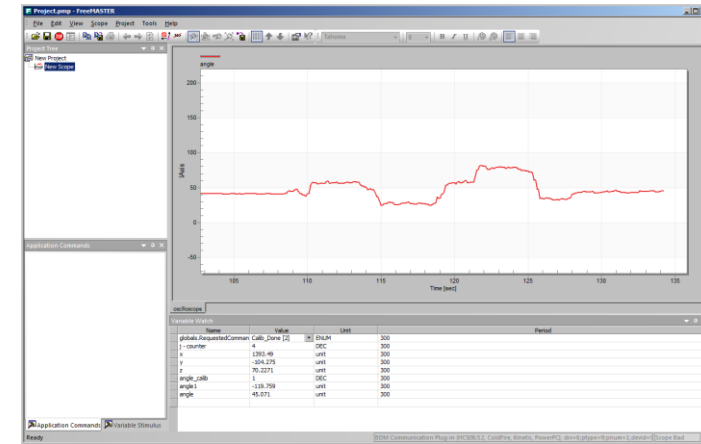
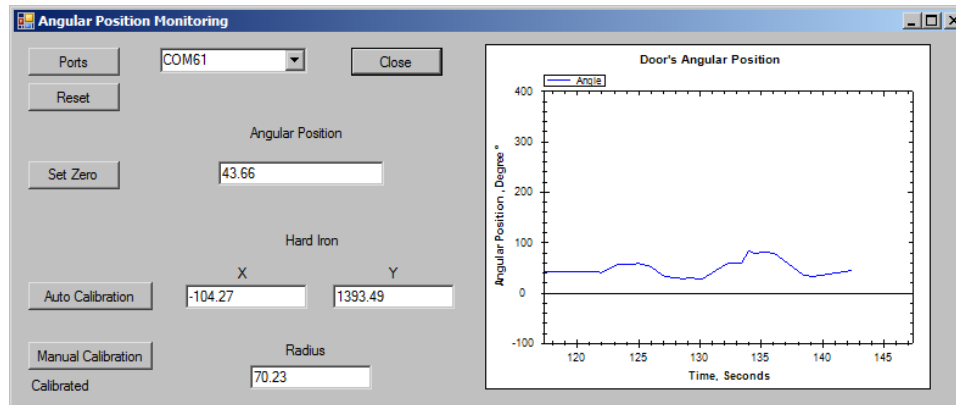
- Target-in-loop Simulations
  - FreeMASTER invisible ActiveX object is accessible also by external standalone applications
  - Standard C++ or VB applications
  - Excel & Visual Basic for Applications
  - Matlab, Simulink
- Target-in-loop Simulation
  - Matlab or Simulink engine lets embedded application to perform calculations
  - SFIO toolbox for direct HW-in-loop simulations and algorithm testing



# FreeMASTER Replacing Custom GUI Applications

## FreeMASTER instead of Custom GUIs

- Comparing FreeMASTER with custom GUI approach
- Typical use cases



# From Custom GUI to FreeMASTER

- **Typical pitfalls of using custom GUI**

- Requires PC Host programming tools and skills
- Never enough communication interfaces, communication issues over and over again
- Time to develop a robust PC Host application
- Deploying GUI to host PC
- Using custom GUI with modified user application

- **Benefits of FreeMASTER**

- uniform approach – application control by variable modification
- works over UART/CAN but also over non-intrusive BDM
- one tool used with variety of GUIs
- GUI easily extended by multimedia content (charts, documentation) local, online or embedded
- Can be used with user-modified content too. User able to mix “our” data with “his” data in common charts.
- GUI project can be extended by user to cover more functionality

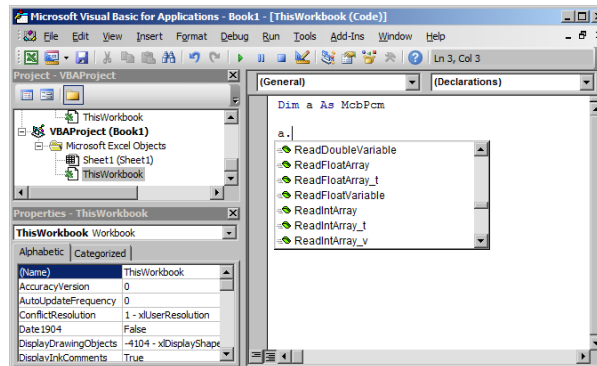
# From Custom GUI to FreeMASTER

- **Typical custom GUI approach:** communication driven data collection, custom protocol
  - PC sends request, Target processes and replies with data
    - pro: under full control of developer, may be shielded from the rest of application logic
    - con: communication development just for sake of GUI, typically not used for any other purpose
    - con: migration to different communication media is typically hard
    - con: user modifications of firmware makes the GUI to stop working
- **FreeMASTER approach:** control by modifying variables
  - use either artificial variables dedicated for GUI control
  - or modify state variables used also by the general application algorithm
    - con: typically requires to change existing applications with custom GUI
    - pro: works over standardized protocol or with BDM direct memory access
    - pro: easy to protect or restrict functionality
    - pro: easy to integrate this approach with additional user modifications to firmware
    - pro: the TSA feature – self-describing and automatic board discovery (FreeMASTER 2.0 in 2015)

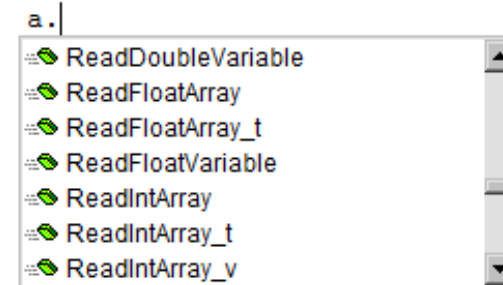
# FreeMASTER Internal Application Structure

## Inside FreeMASTER

- How to get maximum out of FreeMASTER
- Linking with other executables
- Reusing communication layer

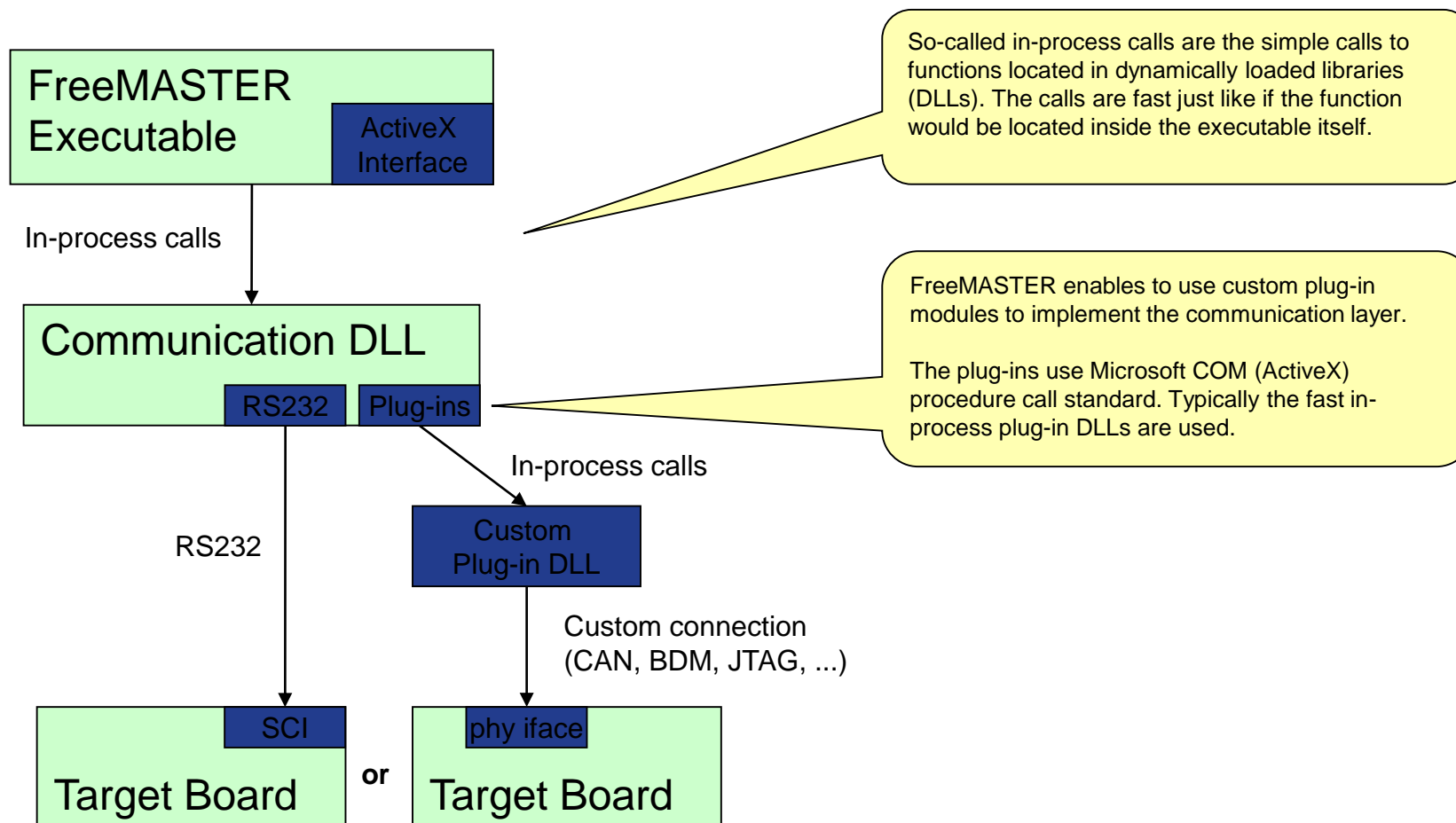


```
Dim a As McbPcm
```



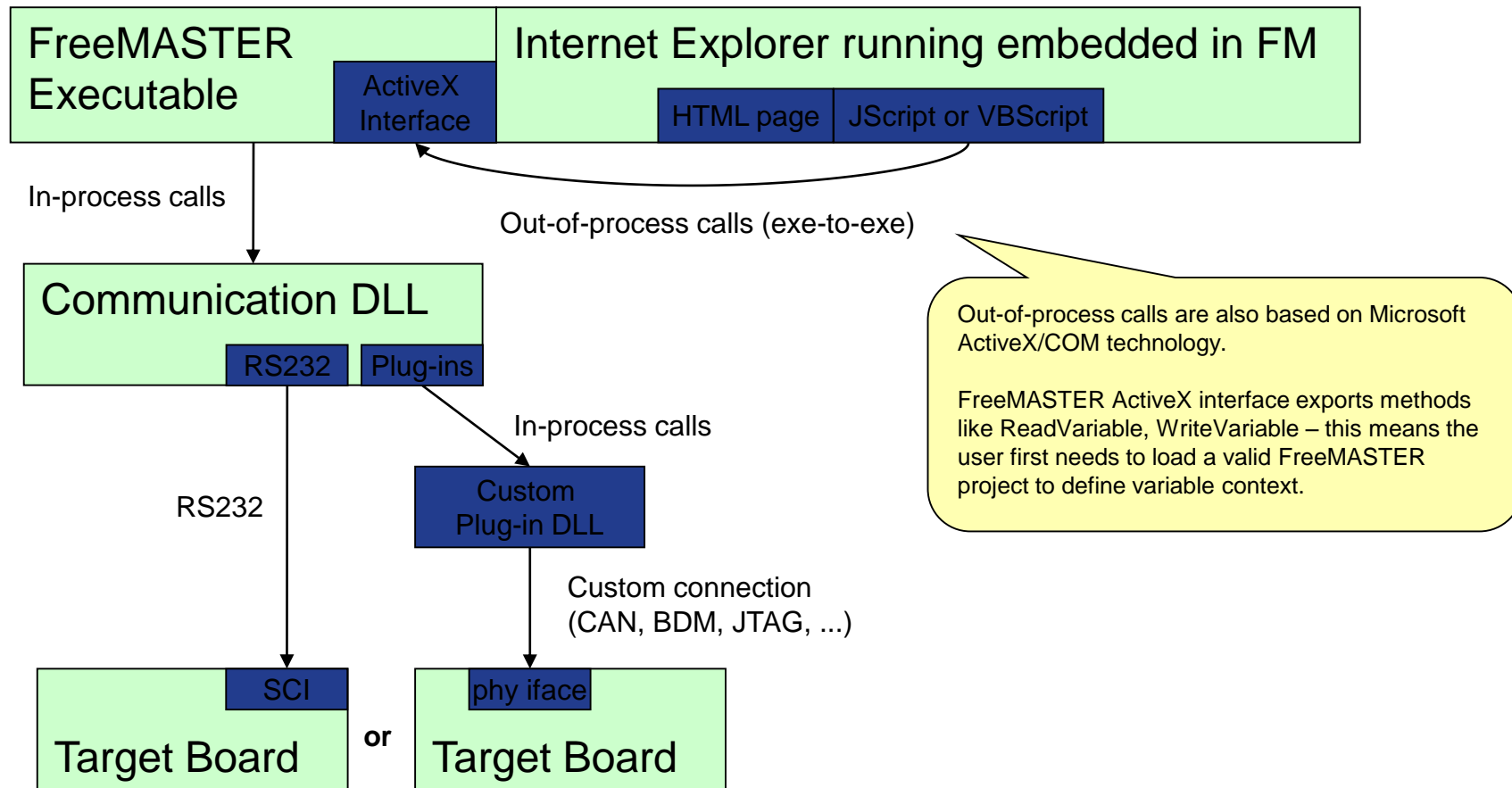
# FreeMASTER Internal Application Structure

## Basic FreeMASTER Communication Diagram



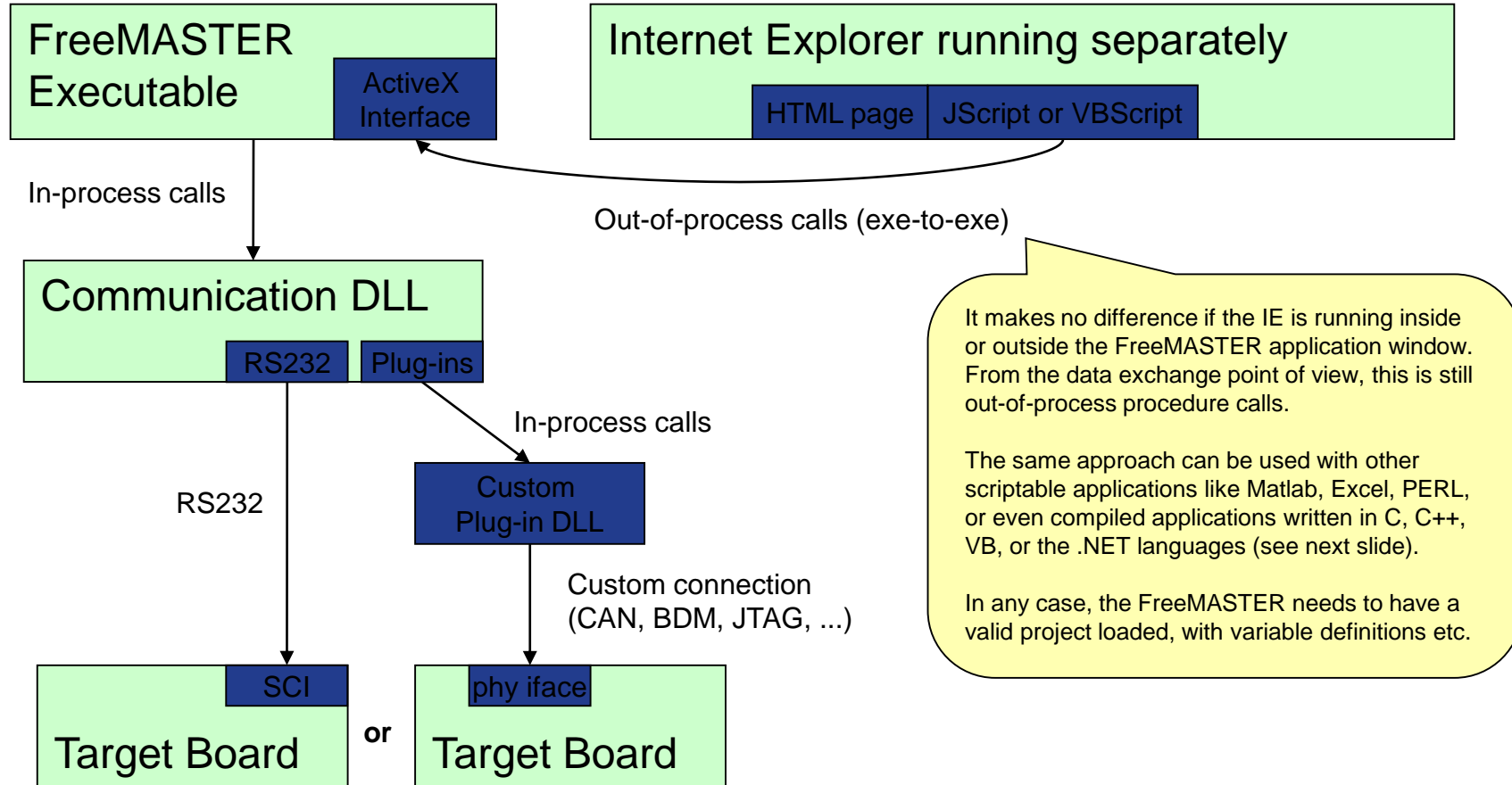
# FreeMASTER Internal Application Structure

## FreeMASTER Communication with HTML/JScript Pages



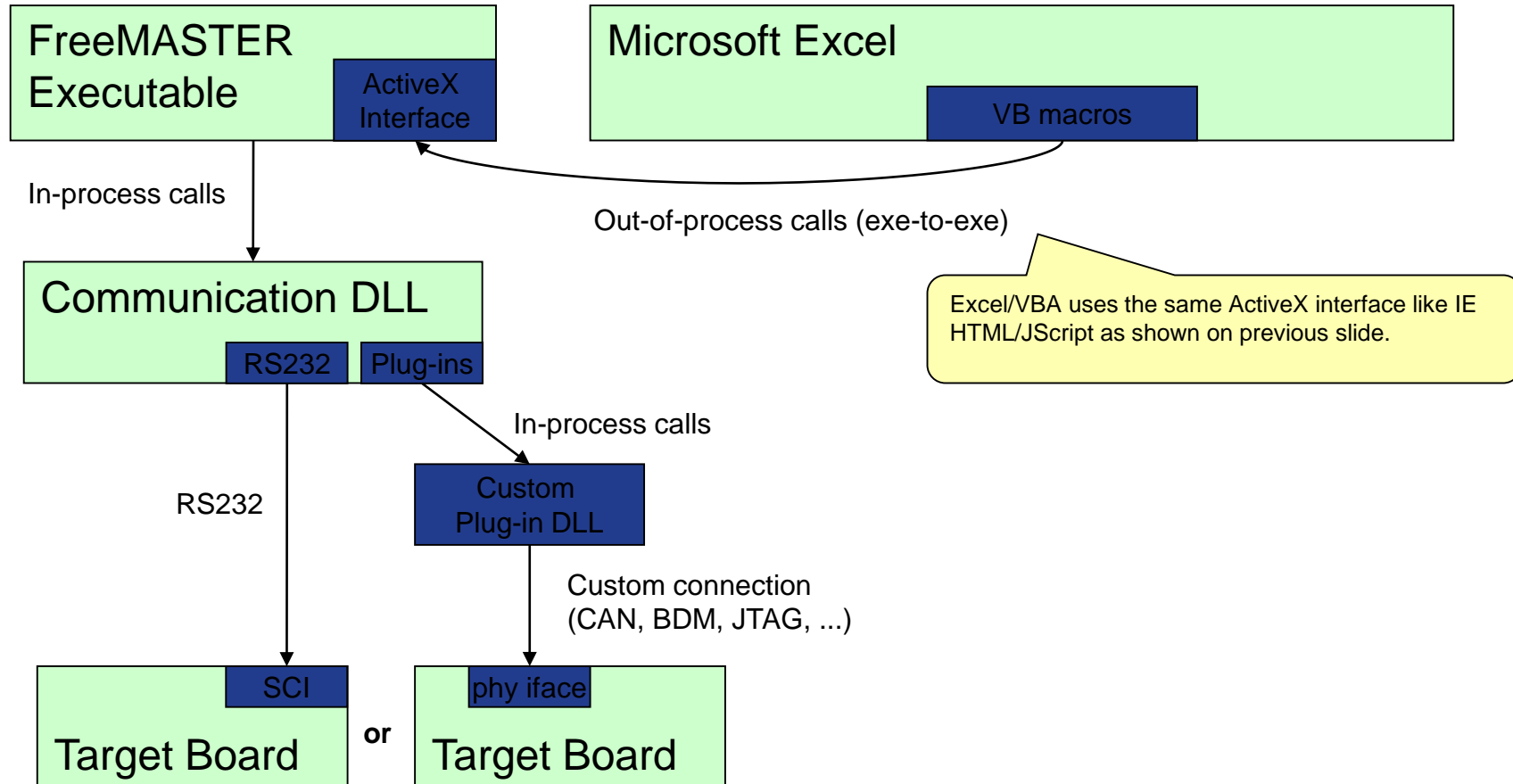
# FreeMASTER Internal Application Structure

## Internet Explorer Running Separately (no difference)



# FreeMASTER Internal Application Structure

## Excel (or other application) accessing FM ActiveX

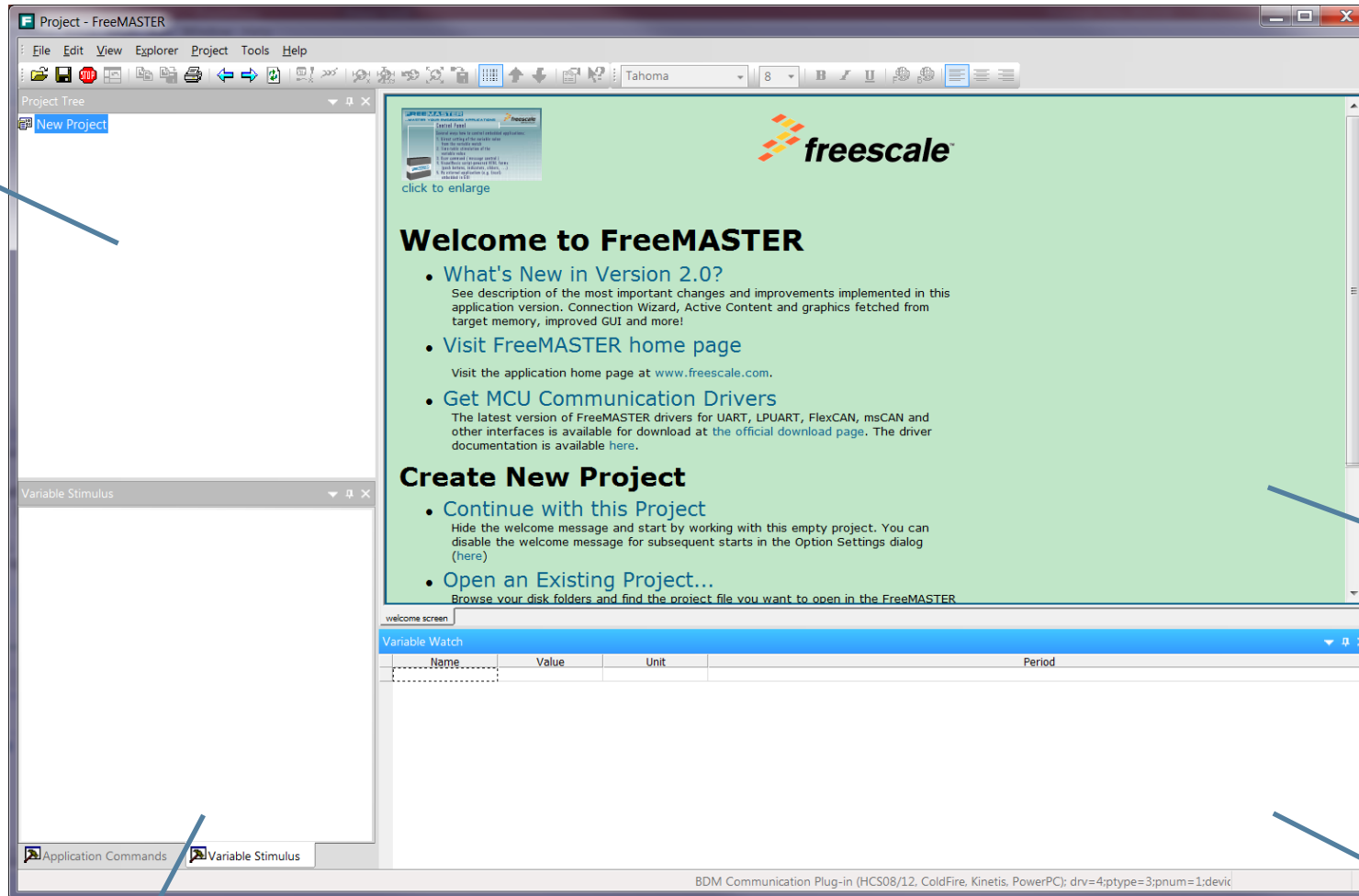




# 02.

## FREEMASTER Window Description

# Application Windows



Project Tree

Detail View

- Scope / Recorder
- Description / Control

Variable Stimulus /  
App Commands

Variable Watch

- Watch
- Control

# FREEMASTER Variable Watch

## Variable Transformations

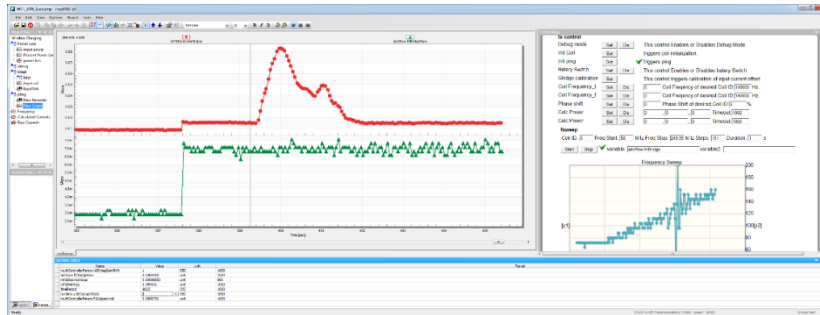
- Variable value can be transformed to a custom unit
- Variable transformations may reference other variable values
- Values are transformed back when writing a new value to the variable

## Ability to protect memory regions

- Describing variables visible to FreeMASTER
- Declaring variables as read-write to read-only for FreeMASTER
- the access is guarded by the embedded-side driver

## Variable update rate

- Choose rate of update based on type of variable
- Rate selectable from 100ms to 10s or as fast as possible (limited by communication interface)



Variable

Definition | Modifying

Variable name: counter

Variable: [0x110C]

Address: counter

Type: unsigned int Size: 4

Format: Fixed point number

Sampling period: fastest

Show as: REAL

Bit fields

When the value is received,

shift it: 0 bits right, and..

mask with: no mask (-1)

Show

☒ val ☐ min ☐ max

Fixed digits (optional)

☐ Exponential

Real type transformation

None

Unit: unit

Linear: ax+b

Lin: two points

Hyp: d/(ax+b)+c

☐ Use 'Moving Averages' filter

☒ Reset history on manual modify

History time: 5000 ms

Text enumeration (after transform)

☐ Enumeration enabled ☐ Always show numeric value

default: unknown ☒ Show number

OK Cancel Apply

Variable

Definition | Modifying

Modifying mode

☐ Don't allow any modifications

☒ Any value within proper limits

☐ One of listed (pre-defined) values only

Limit values

Min (optional):

Max (optional):

Edit style

☐ Simple edit box

☐ Edit box with spin control (change by step)

☐ Combo box with pre-defined values

☒ Hide edit interface at inactive cells

Pre-defined values

☒ All numbers from min to max (by step)

Step:

☐ Text enumeration (see transformation setup)

☐ Other (sep. .)

Write

☒ Write immediately after value changes

☐ Write after ENTER or cell-loose focus

☒ Finish edit mode after ENTER

Shadow write address:

OK Cancel Apply

## Explanation of Variables

The screenshot shows the 'Variable' dialog box in the FREEMASTER software, with the 'Modifying' tab selected. The dialog is annotated with several callouts explaining its components:

- Variable Name:** Points to the 'Variable name' field, which contains the text 'counter'.
- Variable Address/Type/Size and Format:** Points to the 'Variable' section, which includes:
  - 'Address': A dropdown menu showing 'counter' and a memory address '[0x110C]'.
  - 'Type': A dropdown menu showing 'unsigned int'.
  - 'Size': A dropdown menu showing '4'.
  - 'Format': A dropdown menu showing 'Fixed point number'.
- Variable Update Rate:** Points to the 'Sampling period' dropdown menu, which is set to 'fastest'.
- Display Format:** Points to the 'Show as' dropdown menu, which is set to 'REAL'.
- Display Value and optionally display min/max values:** Points to the 'Show' section, which includes checkboxes for 'val' (checked), 'min', and 'max'.
- Bit Field Manipulation:** Points to the 'Bit fields' section, which includes:
  - 'When the value is received, shift it': A field with '0' and a 'bits right, and..' label.
  - 'mask with': A dropdown menu showing 'no mask (-1)'.
- Variable Transformation if Show as: is REAL:** Points to the 'Real type transformation' section, which includes:
  - A dropdown menu with 'None' selected.
  - A list of transformation options: 'None', 'Linear: ax+b' (highlighted), 'Lin: two points', and 'Hyp: d/(ax+b)+c'.
  - Checkboxes for 'Use 'Moving Averages' filter' (unchecked) and 'Reset history on manual modify' (checked).
  - 'History time': A field with '5000' and a 'ms' unit.
- Enumerate values to text:** Points to the 'Text enumeration (after transform)' section, which includes:
  - Checkboxes for 'Enumeration enabled' (unchecked) and 'Always show numeric value' (unchecked).
  - A text area for enumeration values.
  - Buttons for 'Edit..', 'Add..', and 'Del'.
  - A 'default' dropdown menu showing 'unknown'.
  - A checkbox for 'Show number' (checked).

At the bottom of the dialog are 'OK', 'Cancel', and 'Apply' buttons.

**Variable**

Definition | **Modifying**

Modifying mode

- ☐ Don't allow any modifications
- ☒ Any value within proper limits
- ☐ One of listed (pre-defined) values only

Edit style

- ☐ Simple edit box
- ☐ Edit box with spin control (change by step)
- ☒ Combo box with pre-defined values
- ☒ Hide edit interface at inactive cells

Write

- ☒ Write immediately after value changes
- ☐ Write after ENTER or cell-loose focus
- ☒ Finish edit mode after ENTER

Limit values

Min (optional):

Max (optional):

Pre-defined values

- ☒ All numbers from min to max (by step)
- Step:
- ☐ Text enumeration (see transformation setup)
- ☐ Other:

Shadow write address:

OK Cancel Apply

Variable  
Protection

Define appearance  
of variable editing

Define when  
variable is actually  
written

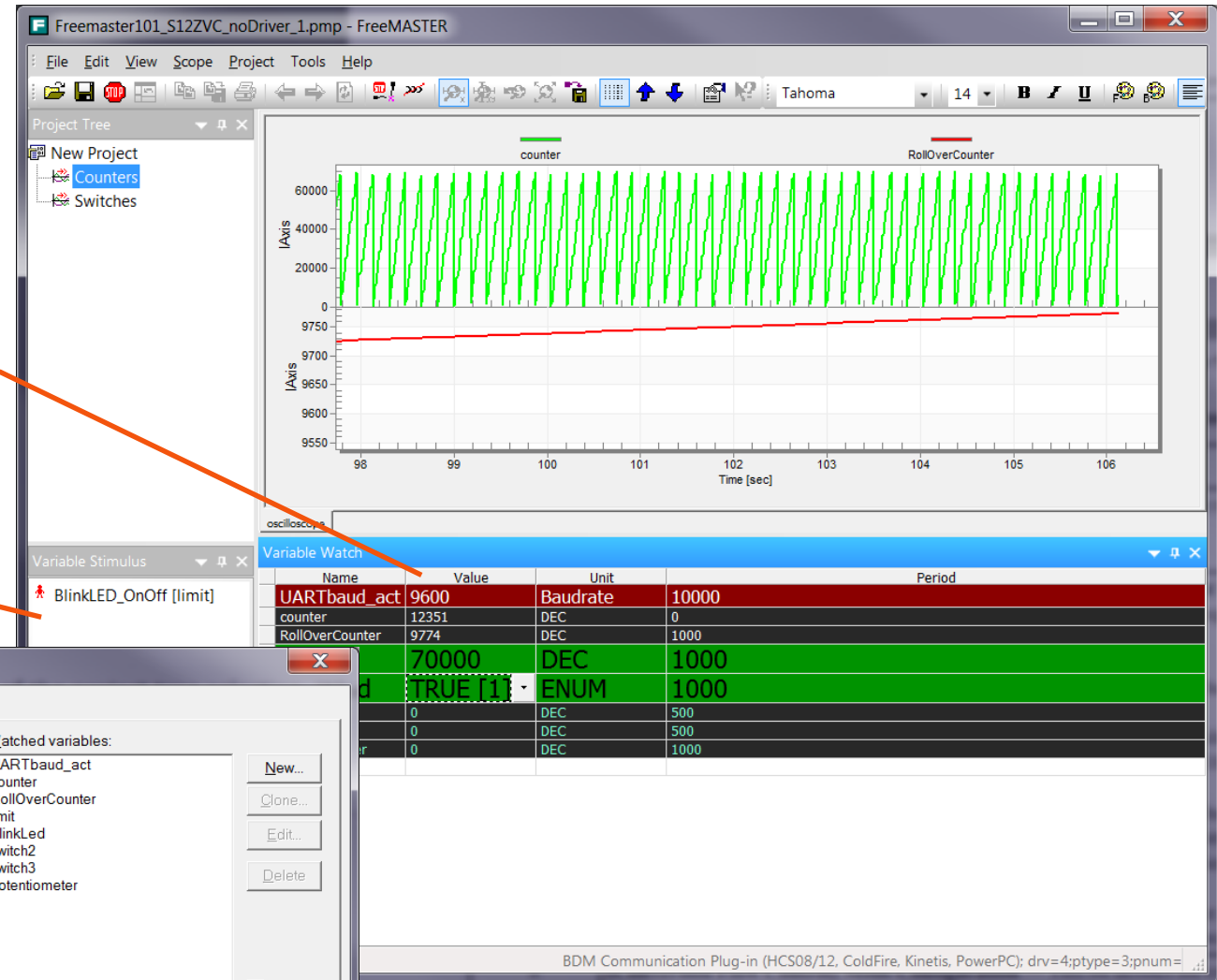
Limit variable modification as  
Min/Max

Limit variable by  
pre-defined  
values

Variable Color and Size

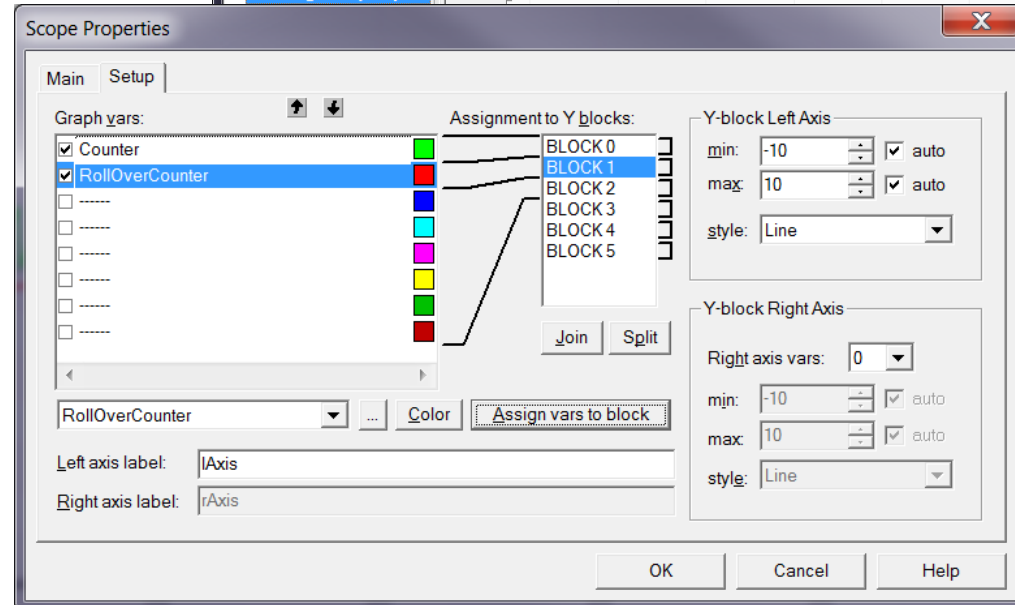
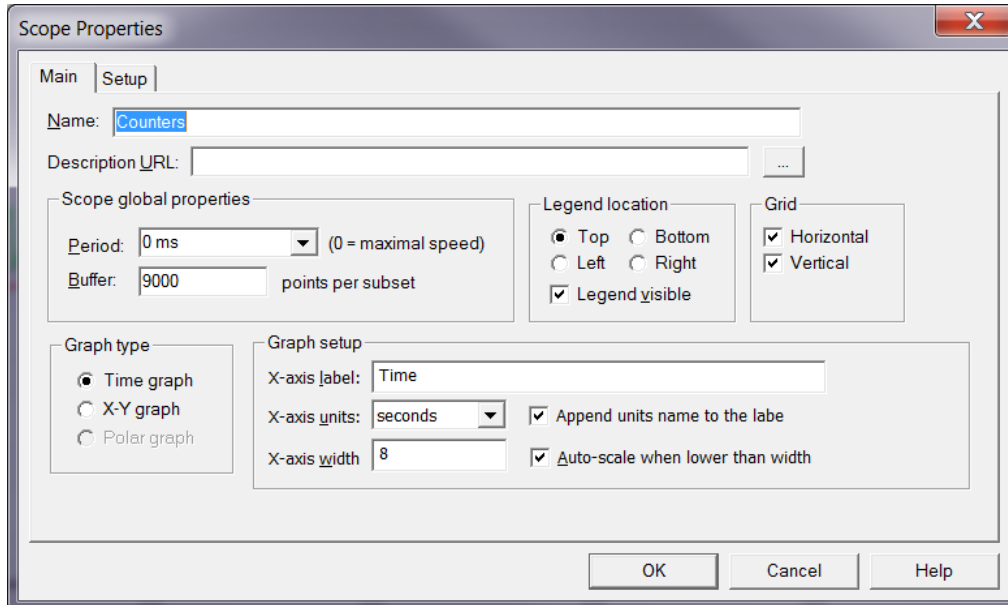
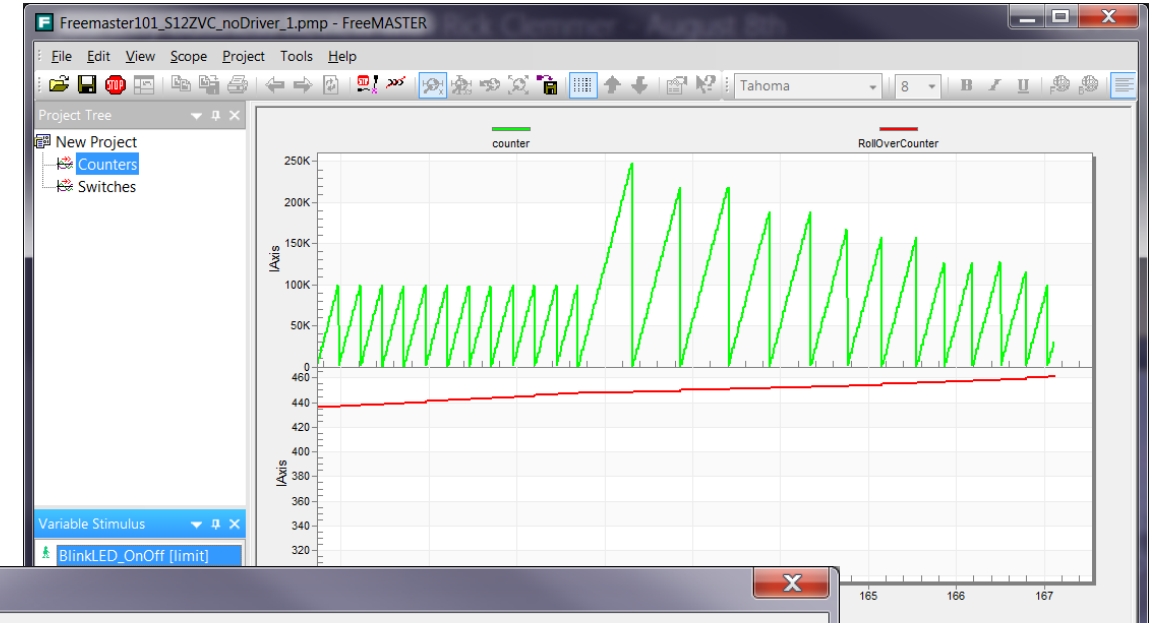
Variable Stimulus

Display Order



## Multiple Scope or Recorders can be configured

- Select between different Scope views
- Can do time graphing or X-Y graphing
- Scope can display multiple variables and/or multiple axis
- Can set up a left and right y-axis on same graph



# NXP Automotive FAE interface for TW customer

Mike Cao 曹学余

Xueyu.cao@nxp.com

Control&Network, Automotive Customer Application Solution and Support

**NXP** (China) Management Co.,Ltd

Mobile: +86 186 1655 2690

Tel: +86 21 2205 2745

地址: 中国上海市裕通路100号宝矿洲际商务中心20层 邮编: 200070

Add: 20F, BM InterContinental Business Center, 100 Yu Tong Road, Shanghai, P.R.C 200070



SECURE CONNECTIONS  
FOR A SMARTER WORLD