



# Hands-On Workshop: How to Use Freescale's **FreeMASTER Tool for Development and Debugging**

FTF-ACC-F1244

Michal Hanak | Application Engineer

John H. Floros | Field Application Engineer

J U N E . 2 0 1 5

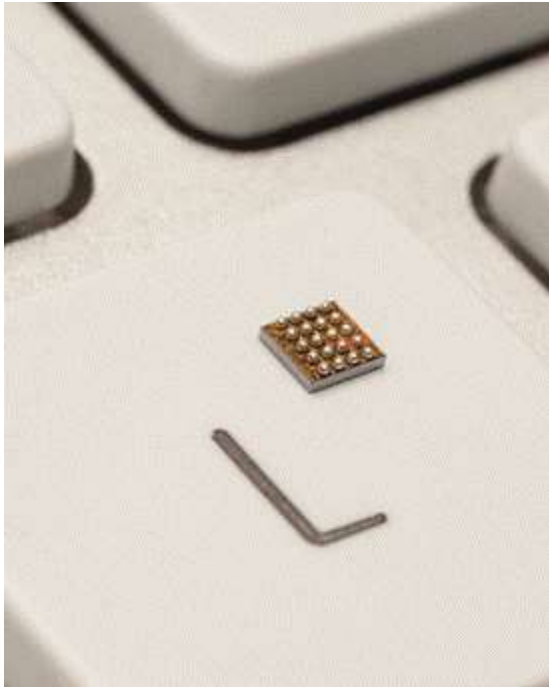


External Use

Freescale, the Freescale logo, ARMv6, C-S, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetic, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Converge, Qorivos, ReadyPilot, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Akrat, BoreX, BioStack, CoreNet, Flexis, Layerscape, M3C, Platform in a Package, QNCC Engine, SMARTMO2, Tower, TurboLink and UMEMS are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2015 Freescale Semiconductor, Inc.



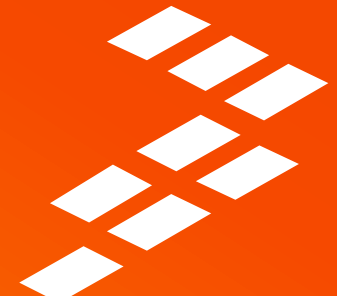
# Introduction: MOTIVATION FOR FREEMASTER



- FreeMASTER was created as an internal development tool by our Motor Control team - “by the engineers for the engineers” in year 2000. Today it is maintained in the SW Libs team in Freescale.
- The original motivation was to be able to visualize and tune parameters without stopping the MCU core in the debugger. Breakpoints in code are often a luxury which you cannot afford in real time applications.
- As it matured a customizable and scriptable HTML rendering engine was added to enable custom GUI pages to be created and used to control, demonstrate or sell embedded applications.

# FreeMASTER Overview

## A Real-Time Monitor for Your Freescale MCU



# Agenda

- What is FreeMASTER?
- FreeMASTER as a Real-Time Monitor
- FreeMASTER as a Control GUI
- FreeMASTER vs. Debugger
- FreeMASTER Replacing Custom GUI Applications
- FreeMASTER Internal Application Structure
- Summary

# Objectives

After this FreeMASTER Overview, you will know:

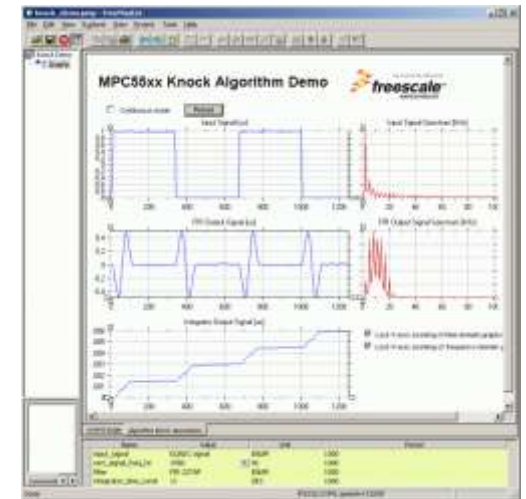
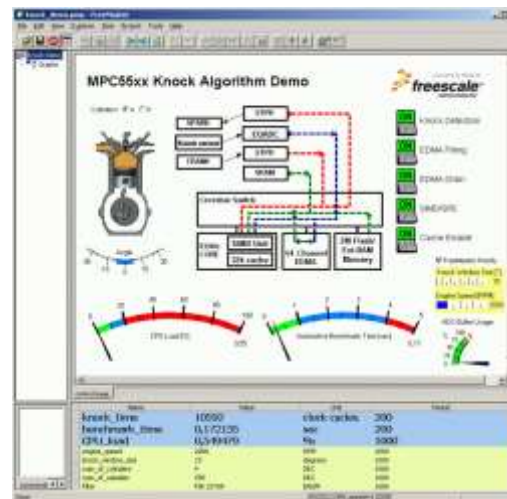
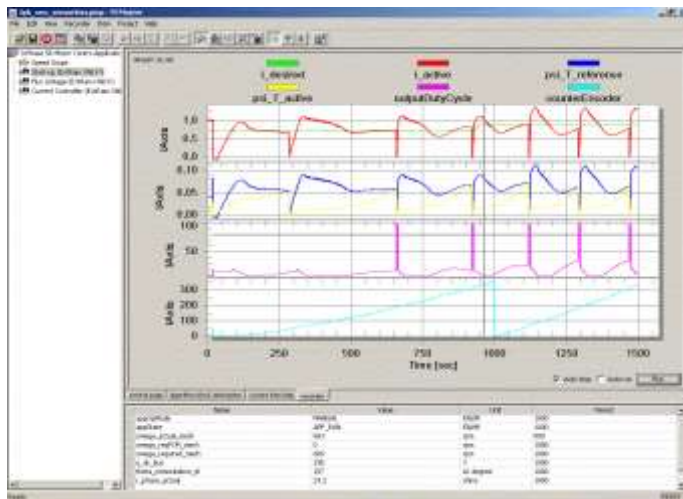
- ✓ What FreeMASTER is and what features it contains for real time monitoring of your application on the MCU
- ✓ How to configure some of the features available in the FreeMASTER user interface
- ✓ The steps required to enable FreeMASTER in your application at a high level

# What is FreeMASTER?

- Real-time Monitor
- Graphical Control Panel
- Demonstration Platform & Selling Tool



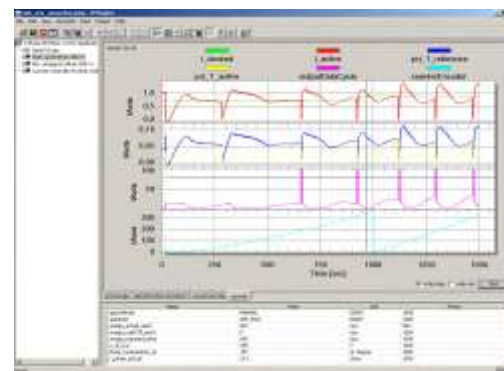
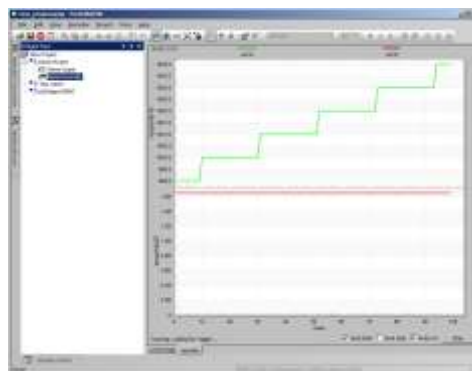
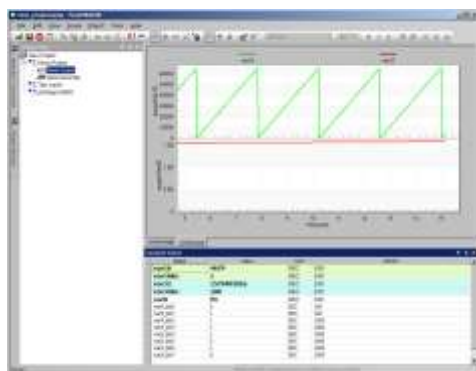
For your  
embedded  
application



# FreeMASTER as a Real-Time Monitor

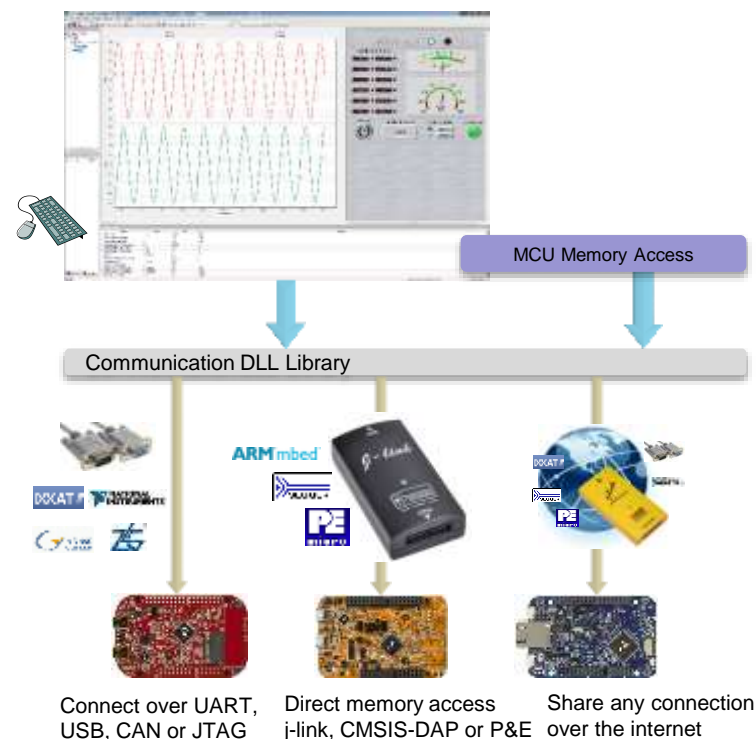
## FreeMASTER can Real-time Monitor

- 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.





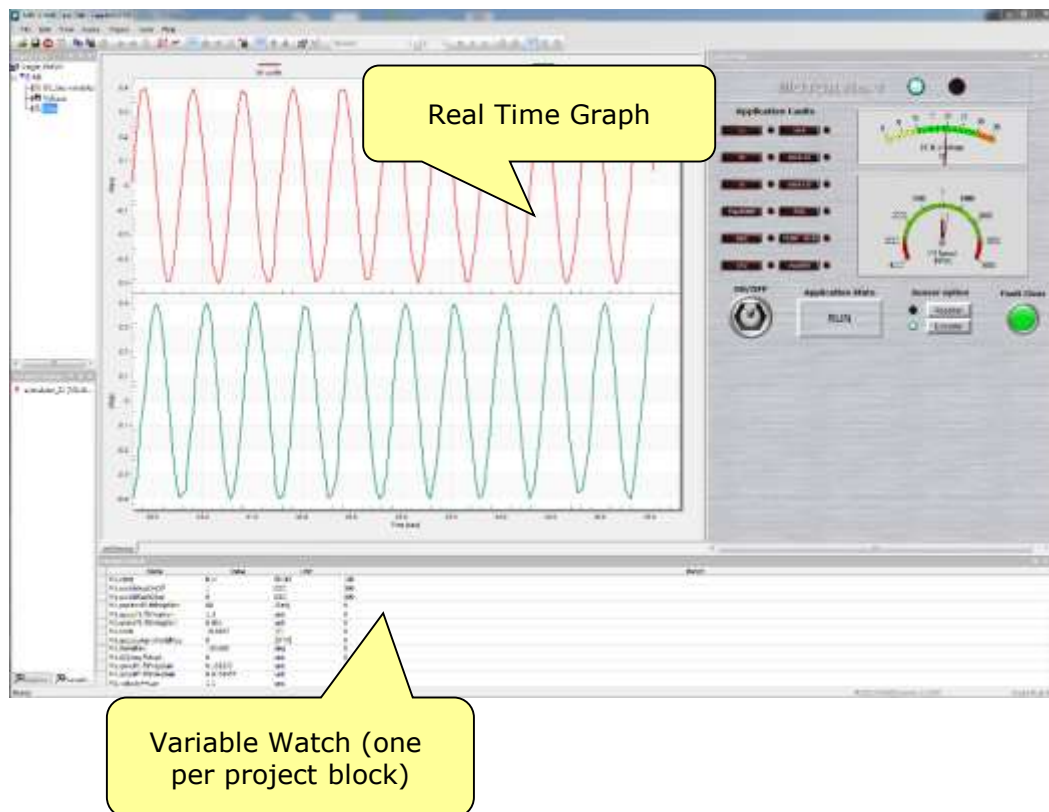
# FreeMASTER Features

- Graphical environment
- Easy to understand navigation
- Real-time access to embedded-side C variables
- Visualization of real-time data in the Scope window
- Acquisition of fast data changes using the on-target Recorder
- Built-in support for standard variable types (integer, floating point, bit-fields)
- Demo mode with password protection support
- HTML-based description or navigation pages
- ActiveX interface to enable VBScript or JScript control over embedded applications
- Remote Communication Server enabling a connection to target board over a network, including the Internet

# 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**

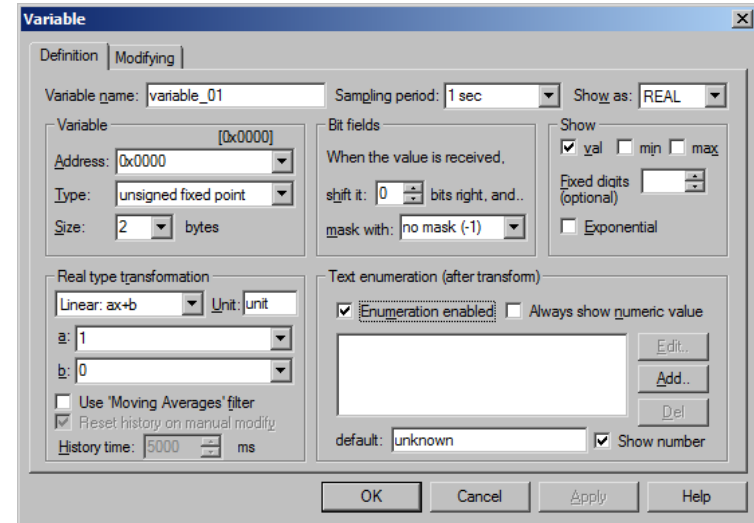
- Value can be transformed to custom units
- Transformations may reference other variable values
- Inverse-transformation applied when writing a new value to the variable

- **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

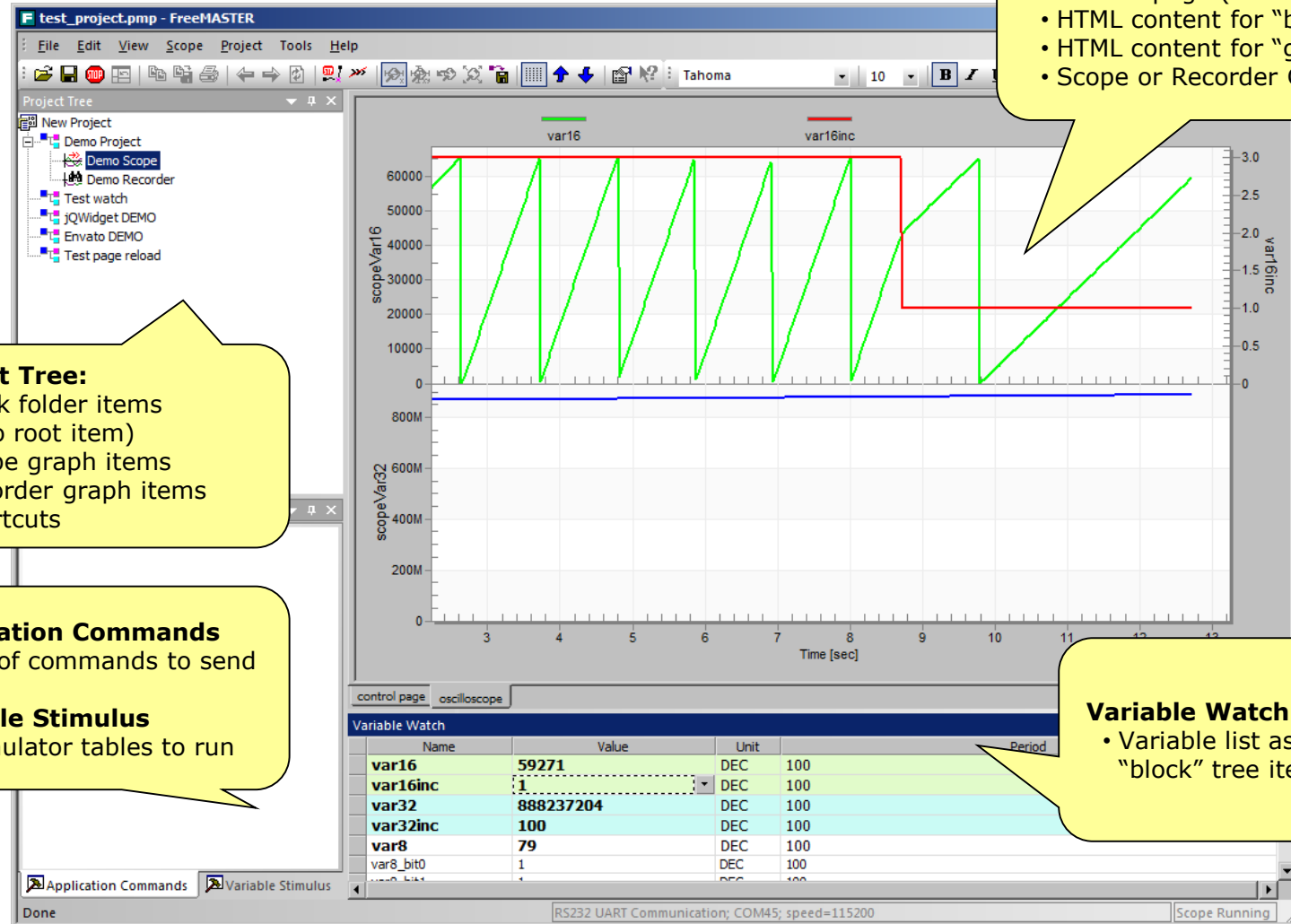
- **Application Commands**

- 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)



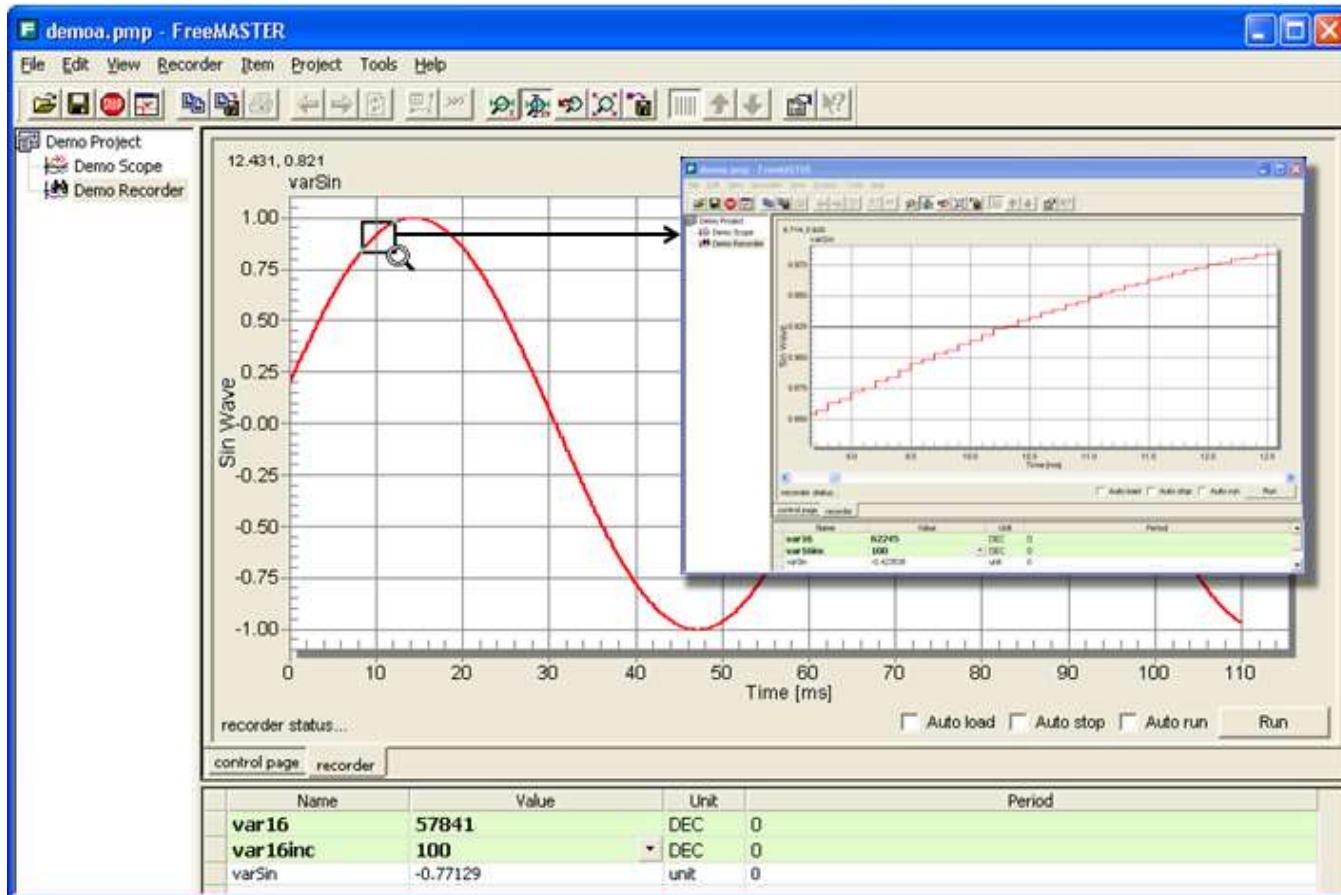
# FreeMASTER as a Real-Time Monitor

## Anatomy of the main window

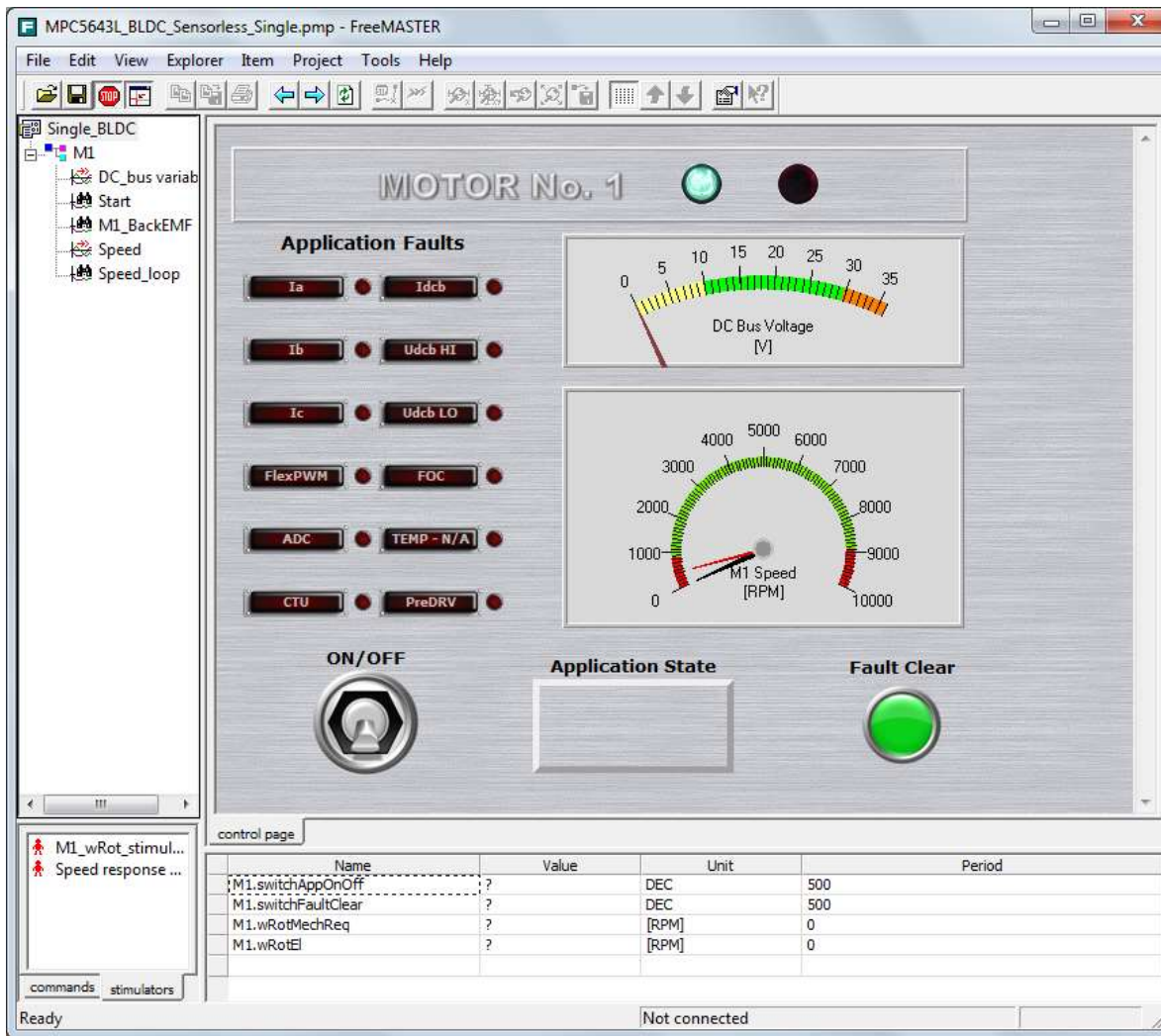


# FreeMASTER as a Real-Time Monitor

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



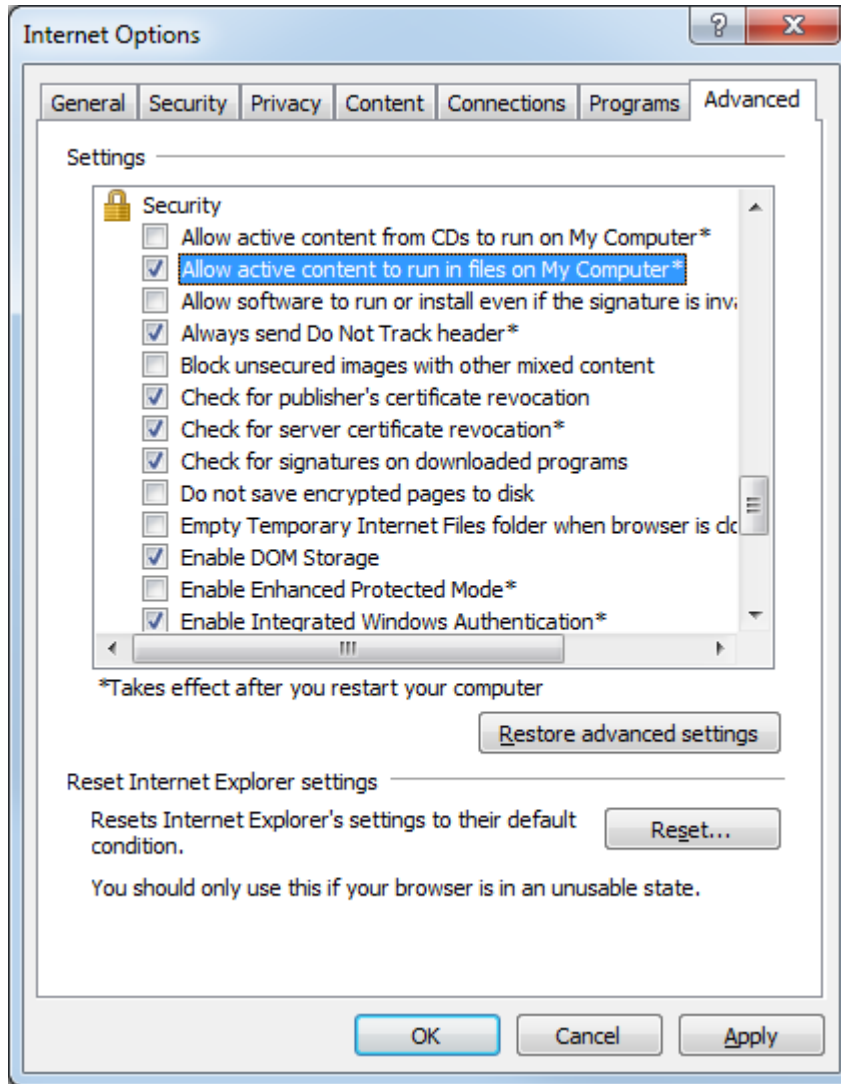
# FreeMASTER as a Real-Time Monitor



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 you 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

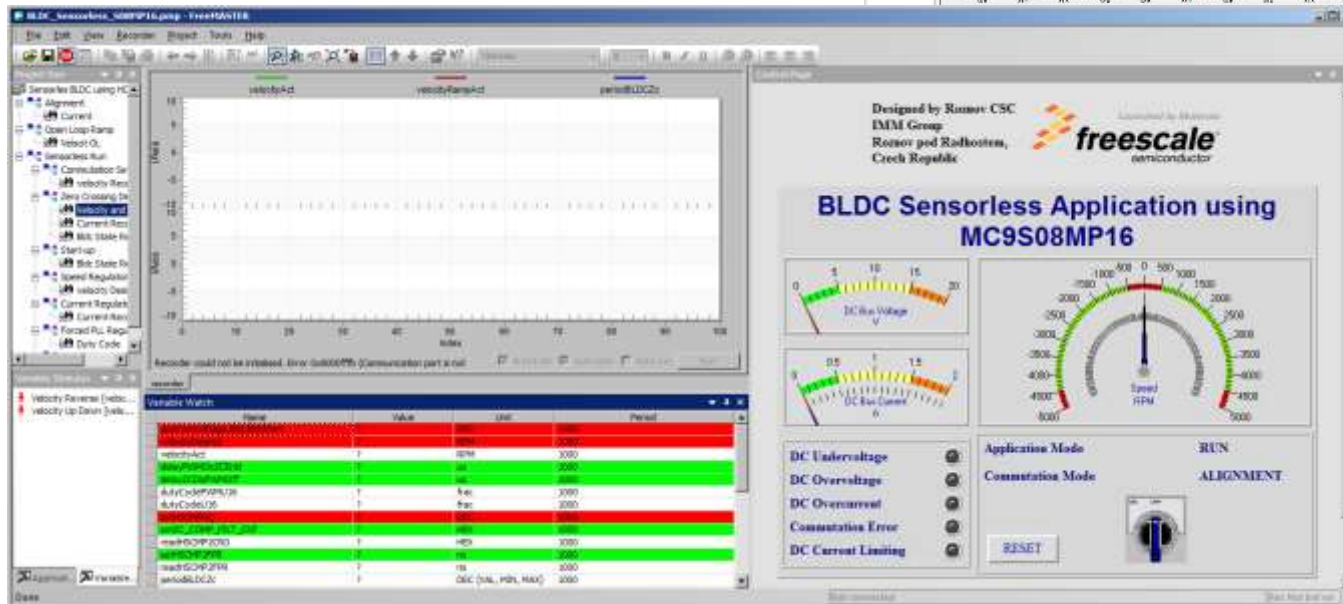
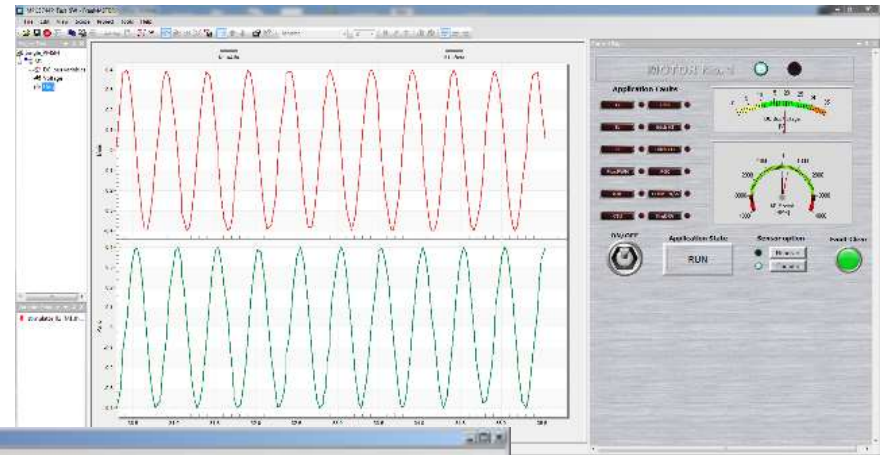
## Highlights

- 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:

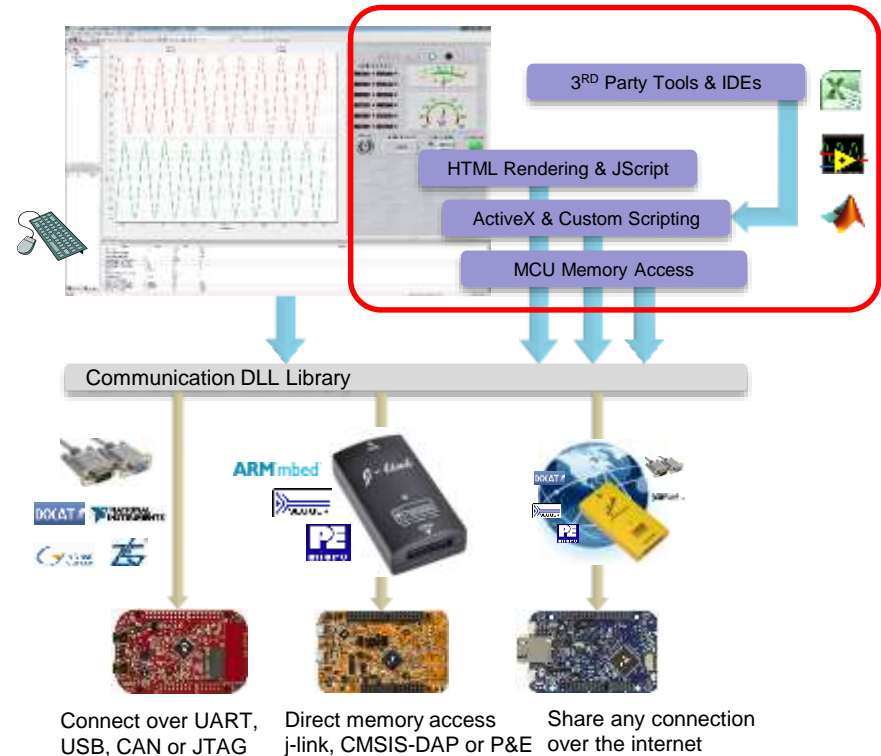
- rendering HTML-encoded GUI
- scriptable in JScript or VBScript
- script access to target memory



# 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

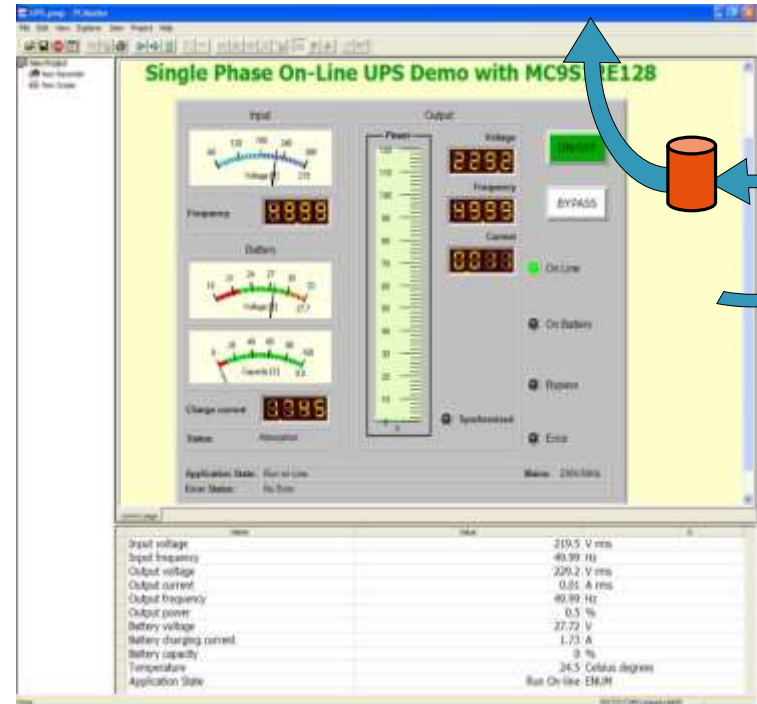


## Sending Application Commands

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

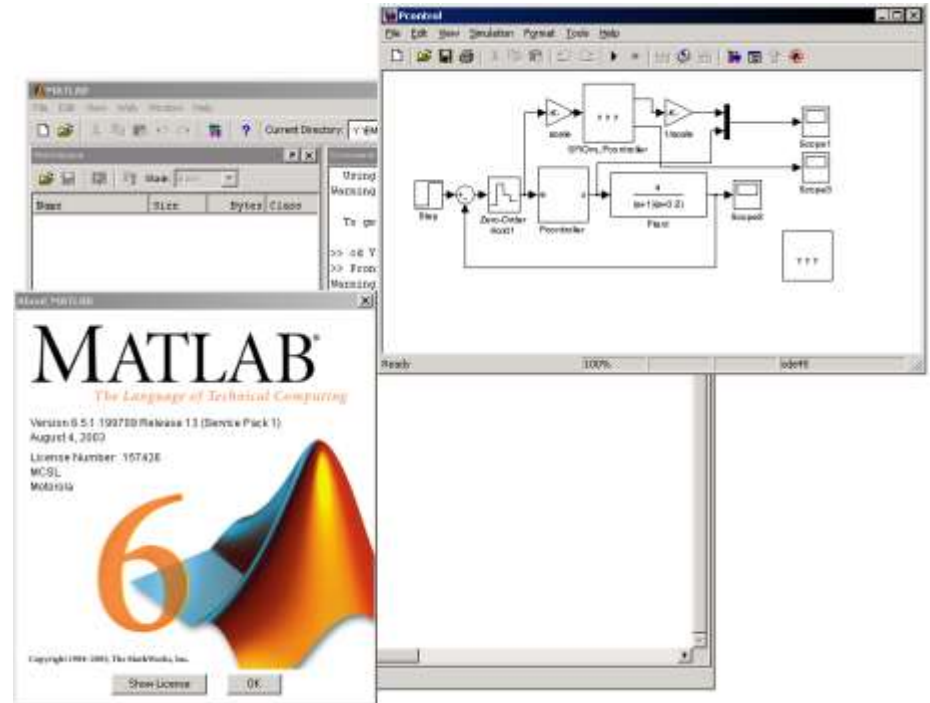
# FreeMASTER as a Control GUI

- 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

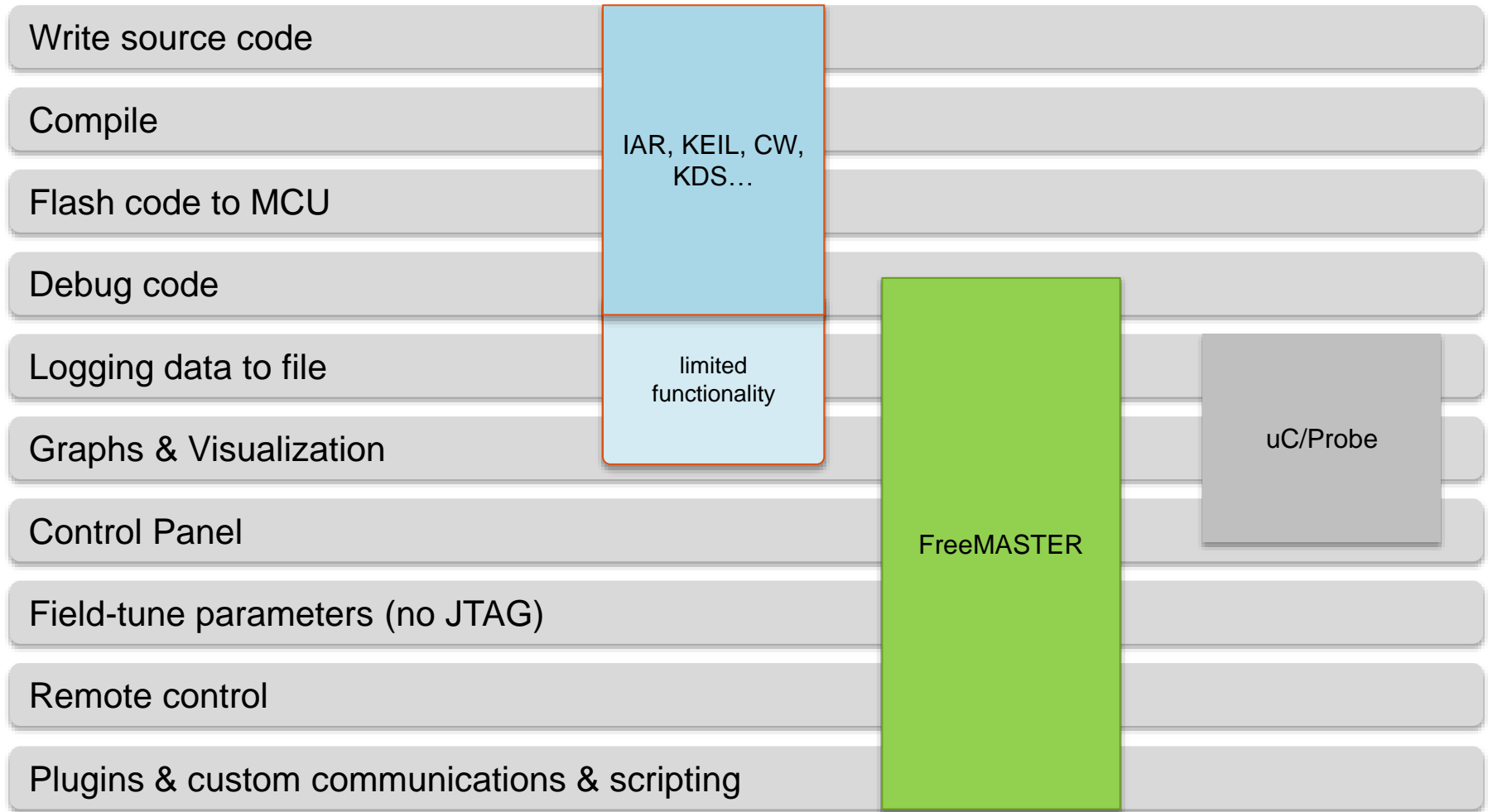


# FreeMASTER as a Control GUI

- 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



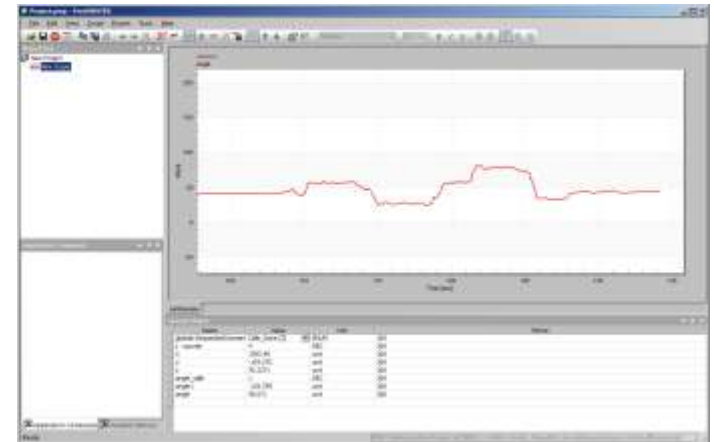
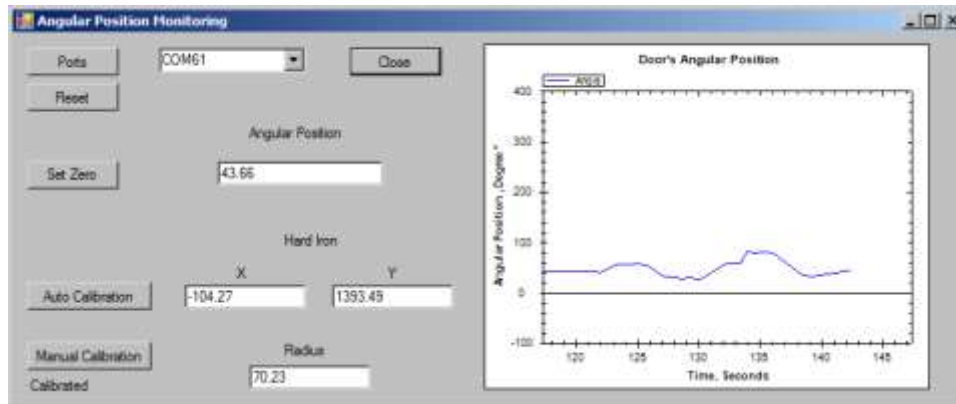
# FreeMASTER vs. Debugger



# 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

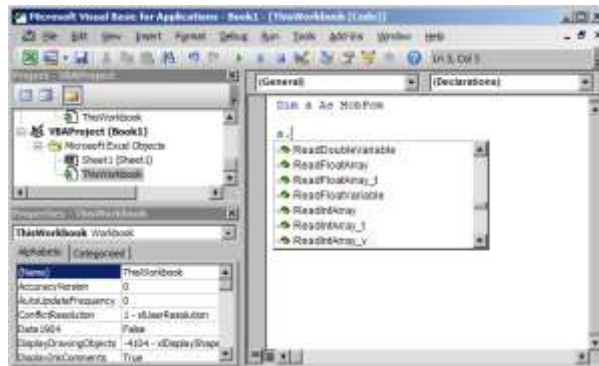
# From Custom GUI to FreeMASTER

- **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



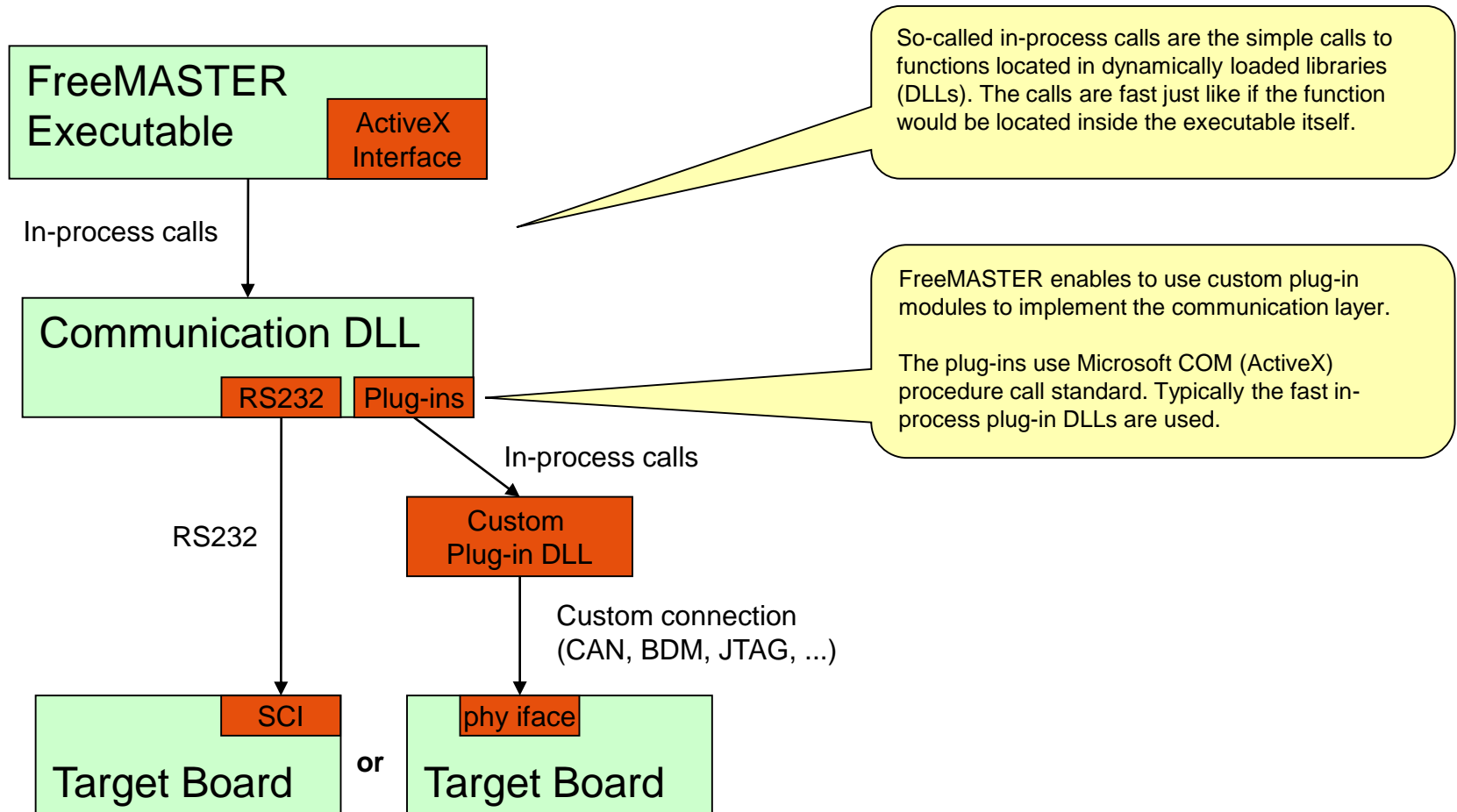
```
Dim a As McbPcm
```

```
a.
```

- ReadDoubleVariable
- ReadFloatArray
- ReadFloatArray\_t
- ReadFloatVariable
- ReadIntArray
- ReadIntArray\_t
- ReadIntArray\_v

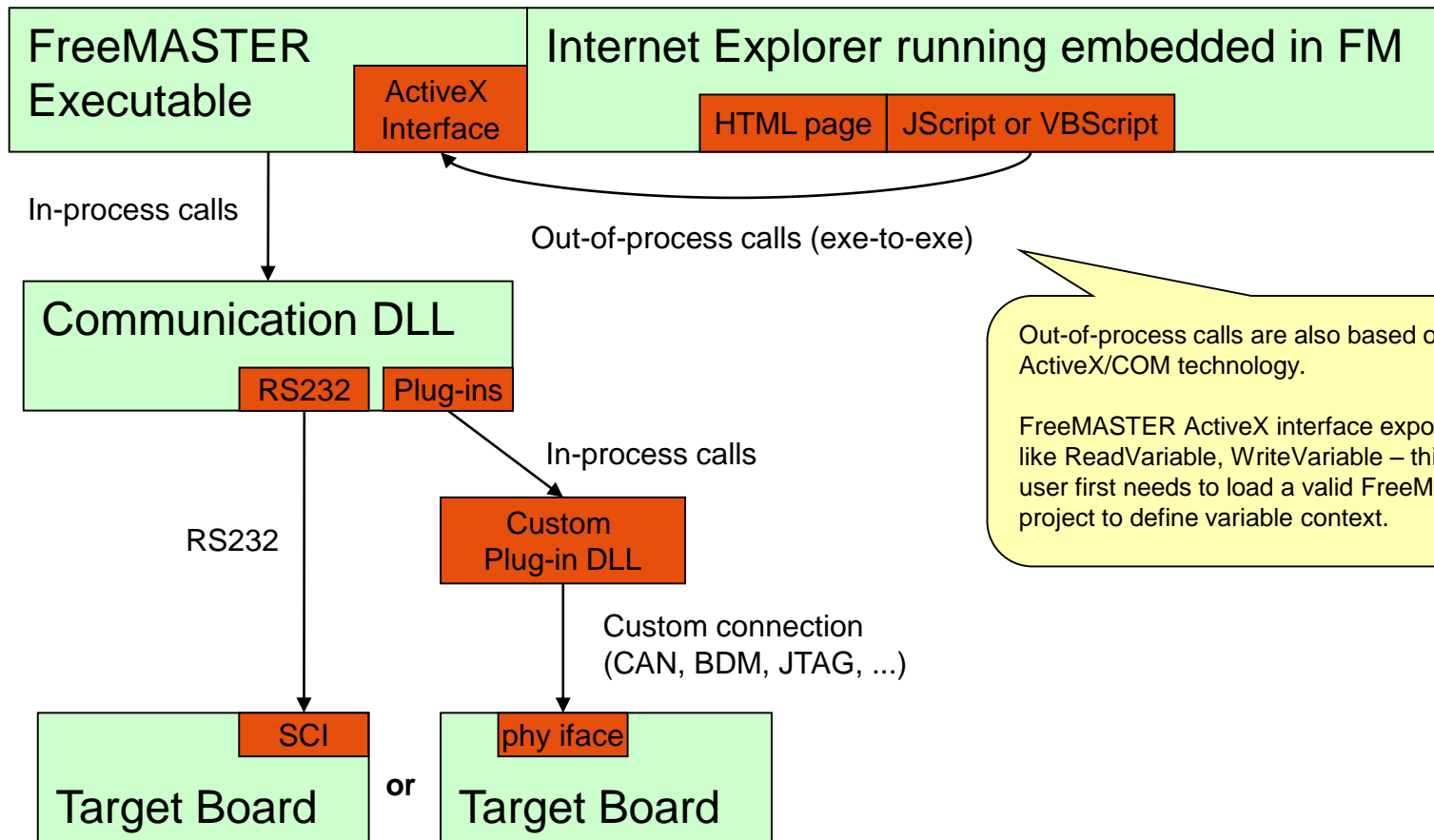
# FreeMASTER Internal Application Structure

## Basic FreeMASTER Communication Diagram



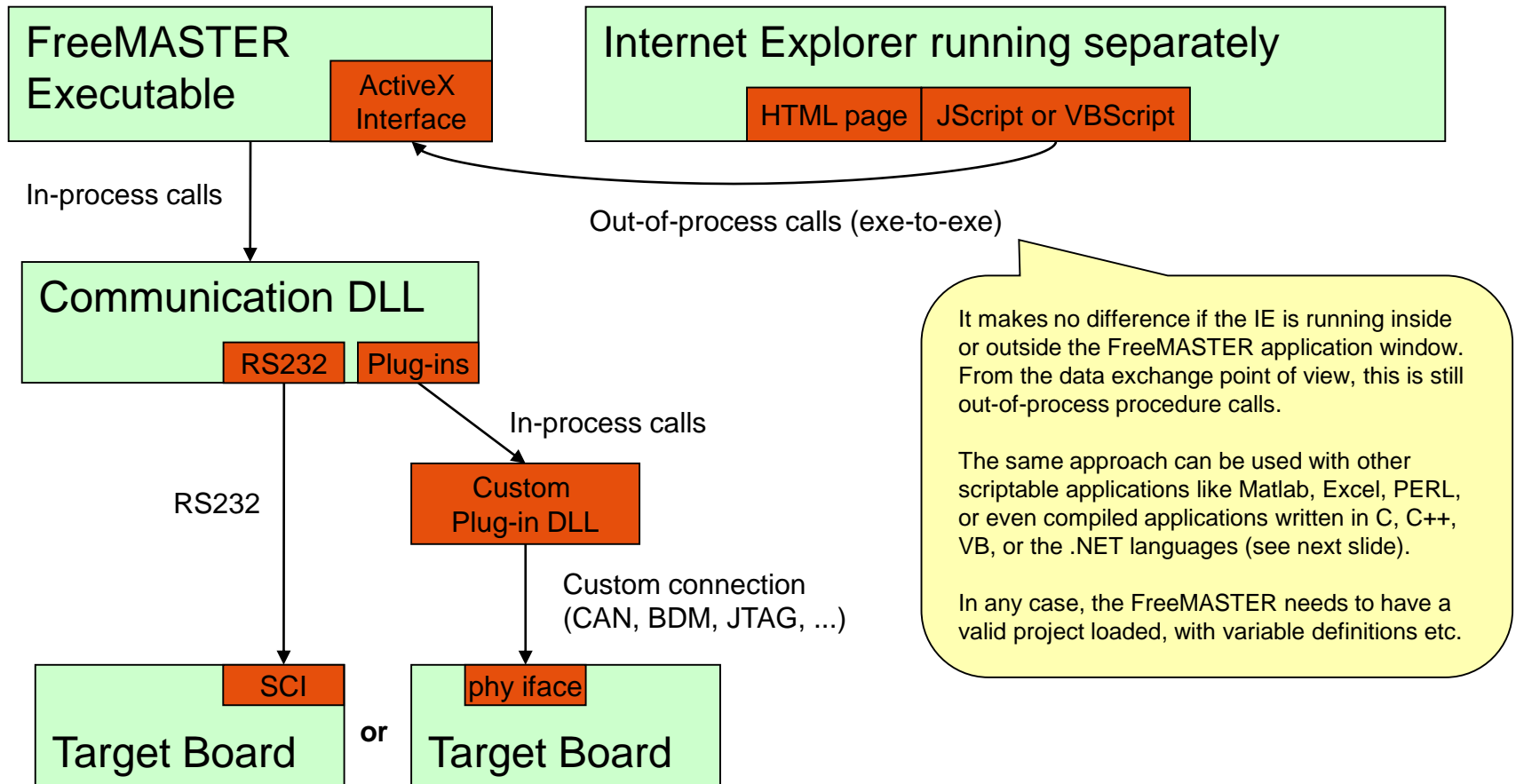
# FreeMASTER Internal Application Structure

## FreeMASTER Communication with HTML/JScript Pages



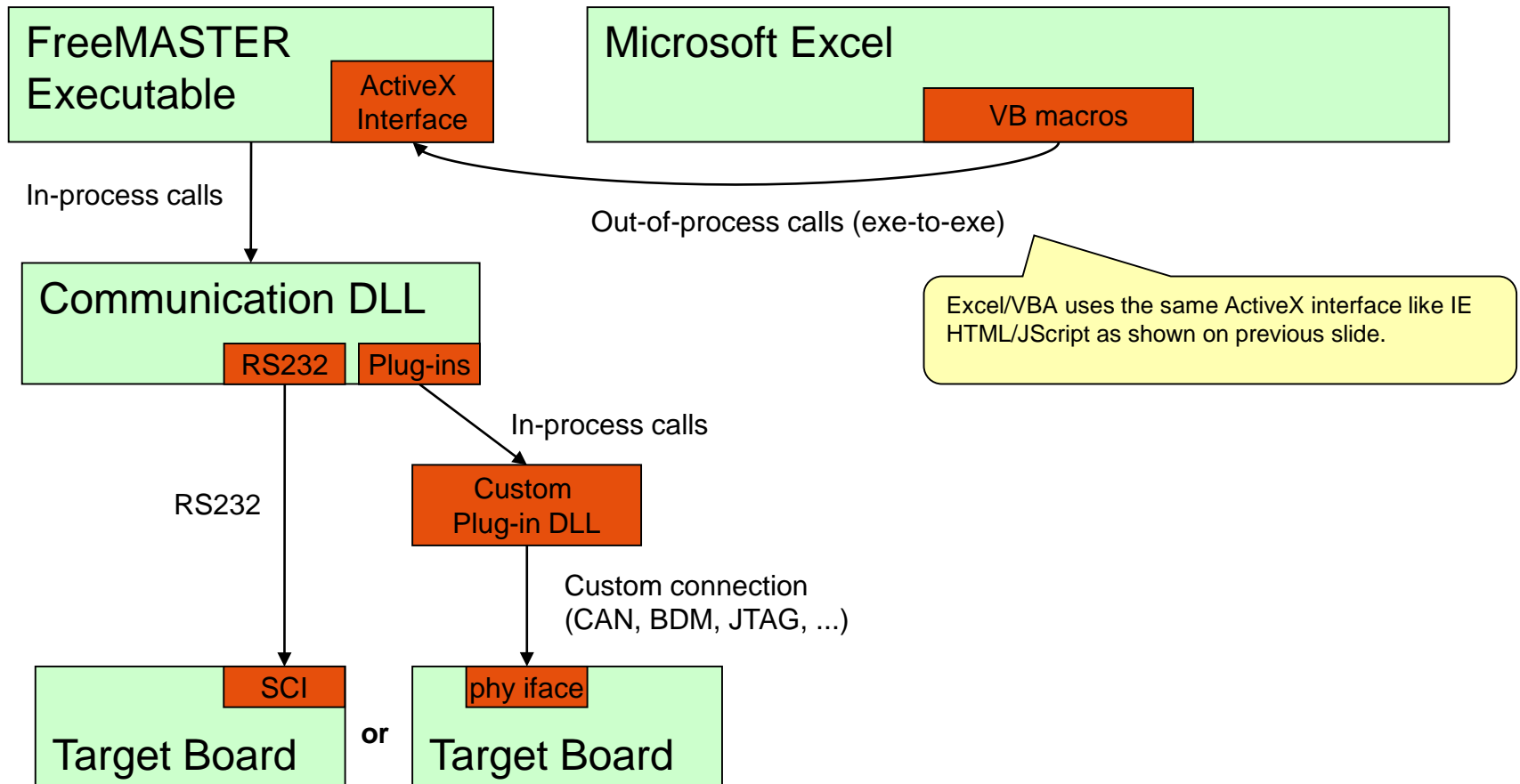
# FreeMASTER Internal Application Structure

## Internet Explorer Running Separately (no difference)

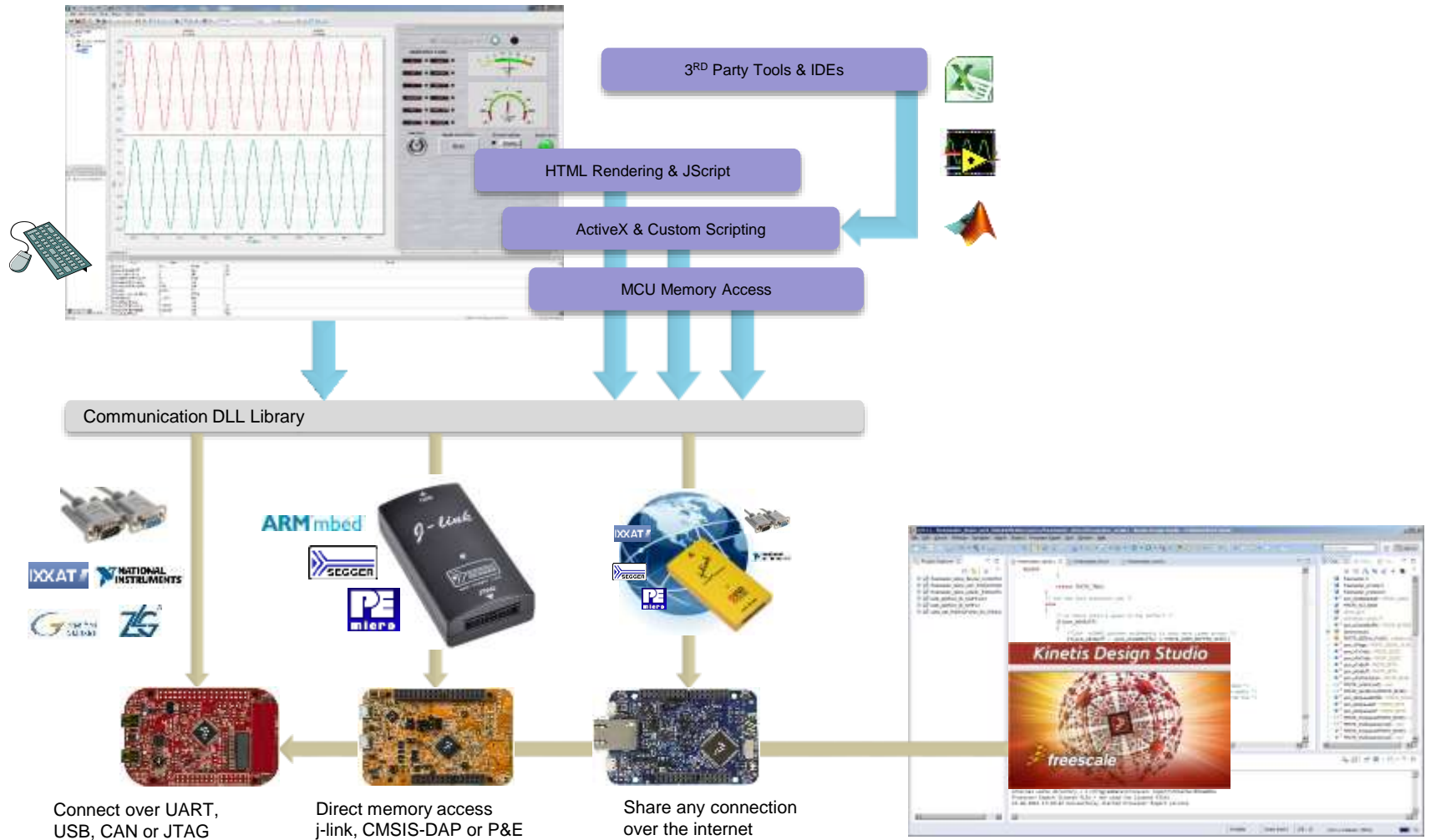


# FreeMASTER Internal Application Structure

## Excel (or other application) accessing FM ActiveX



# Thank you - Questions?





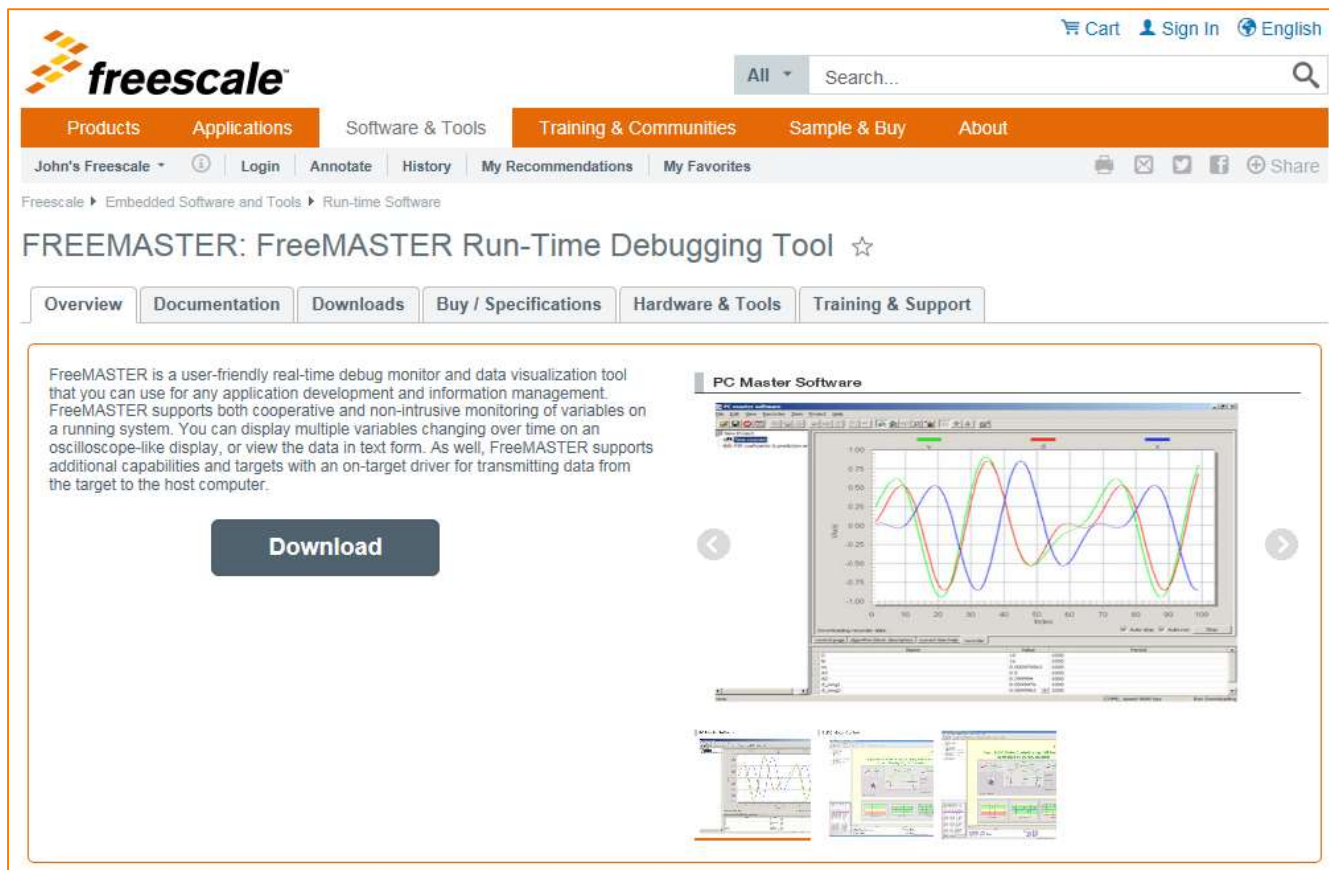
# Any Questions?



# Summary: More Information

For FreeMASTER, visit the Freescale website:

[www.freescale.com/Freemaster](http://www.freescale.com/Freemaster)



The screenshot shows the Freescale website's product page for FreeMASTER. The page features the Freescale logo, a navigation bar with links to Products, Applications, Software & Tools, Training & Communities, Sample & Buy, and About. Below the navigation bar, there's a section for 'John's Freescale' with links to Login, Annotate, History, My Recommendations, and My Favorites. The main heading is 'FREEMASTER: FreeMASTER Run-Time Debugging Tool'. Below this, there are tabs for Overview, Documentation, Downloads, Buy / Specifications, Hardware & Tools, and Training & Support. The 'Downloads' tab is selected. The content area describes FreeMASTER as a user-friendly real-time debug monitor and data visualization tool. It includes a 'Download' button and a preview of the PC Master Software interface, which shows a graph of multiple variables over time.

**freescale™**

All Search...

Products Applications Software & Tools Training & Communities Sample & Buy About

John's Freescale Login Annotate History My Recommendations My Favorites

Freescle ▶ Embedded Software and Tools ▶ Run-time Software

## FREEMASTER: FreeMASTER Run-Time Debugging Tool ☆

Overview Documentation Downloads Buy / Specifications Hardware & Tools Training & Support

FreeMASTER is a user-friendly real-time debug monitor and data visualization tool that you can use for any application development and information management. FreeMASTER supports both cooperative and non-intrusive monitoring of variables on a running system. You can display multiple variables changing over time on an oscilloscope-like display, or view the data in text form. As well, FreeMASTER supports additional capabilities and targets with an on-target driver for transmitting data from the target to the host computer.

**Download**

**PC Master Software**

The PC Master Software interface shows a graph of multiple variables over time, with a table of data below it.

# Summary

- Any Questions?
- Please Fill Out Your Surveys
- Thank you for your time





[www.Freescale.com](http://www.Freescale.com)