USING NXP FREEMASTER

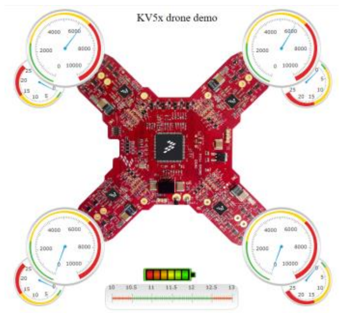
TO DEBUG, TUNE, CONTROL AND DEMONSTRATE EMBEDDED APPLICATIONS

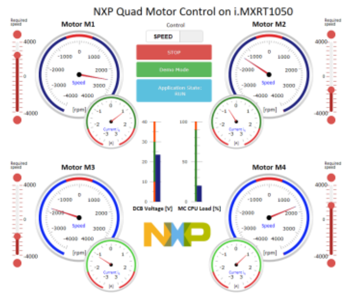
MICHAL HANAK
JANUARY 2020

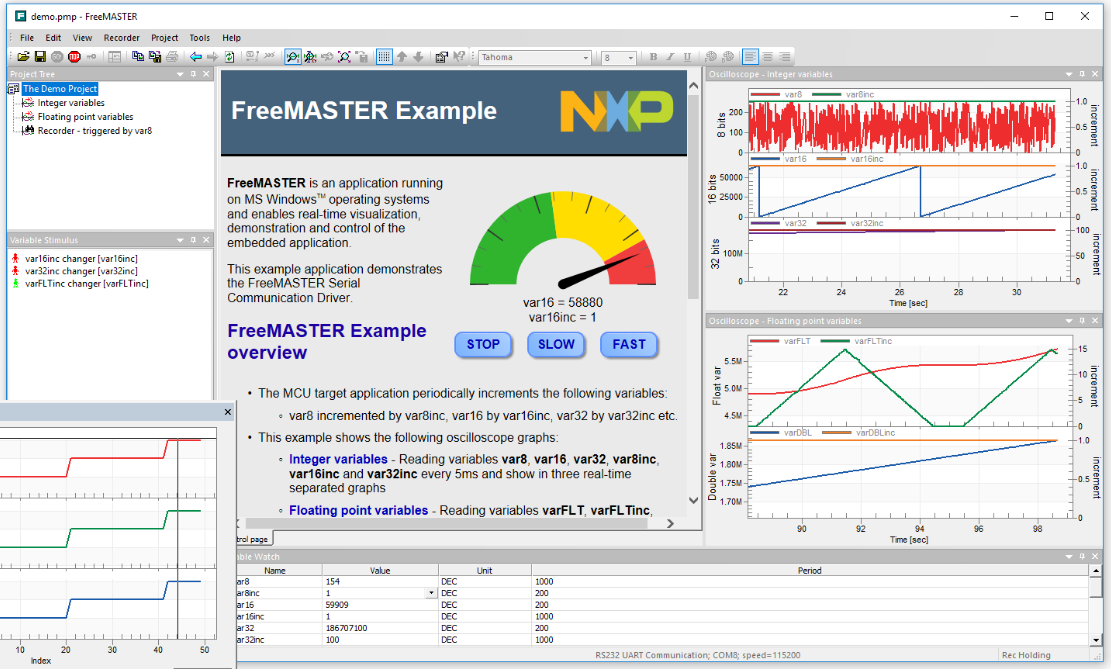
SECURE CONNECTIONS
FOR A SMARTER WORLD

AGENDA

- What is FreeMASTER?
- FreeMASTER as a Real-time Monitor
- FreeMASTER as a Control GUI
- FreeMASTER vs. Debugger
- FreeMASTER Replacing Custom GUI Apps
- FreeMASTER Lite
- Inside FreeMASTER






demo.pmp - FreeMASTER
File
Edit
View
Recorder
Project
Tools
Help
Project Tree
The Demo Project
Integer variables
Floating point variables
Recorder - triggered by var8
Variable Stimulus
var16inc changer [var16inc]
var32inc changer [var32inc]
varFLTinc changer [varFLTinc]
Oscilloscope - Integer variables
Oscilloscope - Floating point variables
Time [sec]
STOP
SLOW
FAST

FreeMASTER Example

FreeMASTER is an application running on MS Windows™ operating systems and enables real-time visualization, demonstration and control of the embedded application.

This example application demonstrates the FreeMASTER Serial Communication Driver.

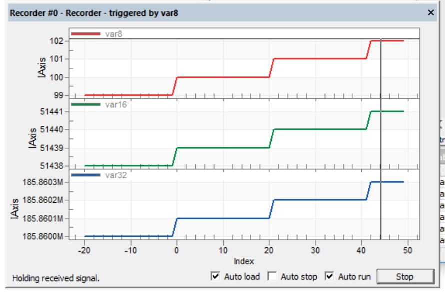
var16 = 58880
var16inc = 1

FreeMASTER Example overview

- The MCU target application periodically increments the following variables:
  - var8 incremented by var8inc, var16 by var16inc, var32 by var32inc etc.
- This example shows the following oscilloscope graphs:
  - **Integer variables** - Reading variables **var8**, **var16**, **var32**, **var8inc**, **var16inc** and **var32inc** every 5ms and show in three real-time separated graphs
  - **Floating point variables** - Reading variables **varFLT**, **varFLTinc**,

Name	Value	Unit	Period
ar8	154	DEC	1000
ar8inc	1	DEC	200
ar16	59909	DEC	200
ar16inc	1	DEC	1000
ar32	186707100	DEC	200
ar32inc	100	DEC	1000

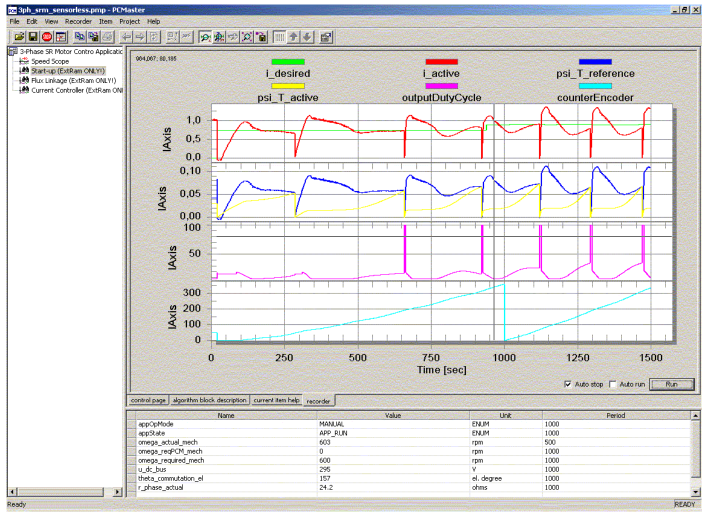
RS232 UART Communication; COM8; speed=115200 Rec Holding



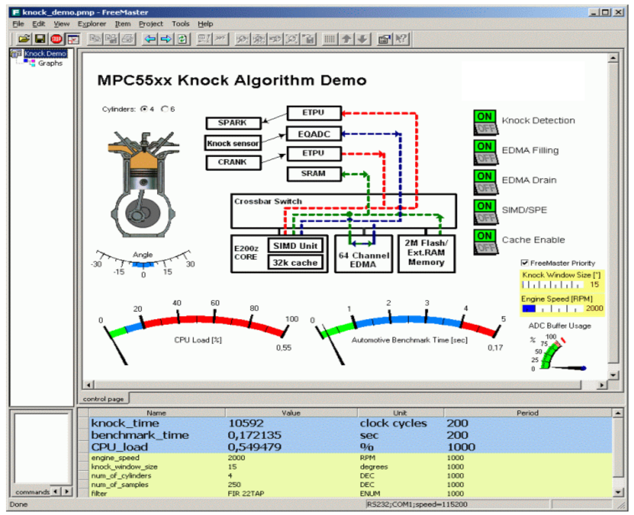
What is FreeMASTER?

- Real-time Monitor
- Graphical Control Panel
- Demonstration Platform

FOR YOUR EMBEDDED APPLICATION

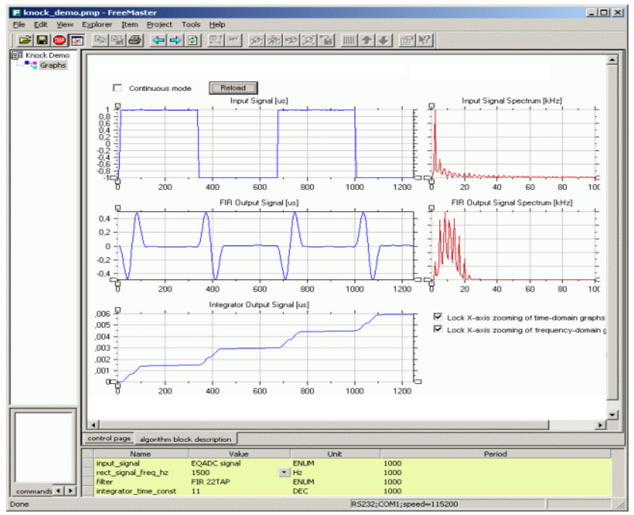


Name	Value	Unit	Period
appOpMode	MANUAL	ENUM	1000
appState	APP_RUN	ENUM	1000
omega_actual_mech	603	rpm	500
omega_reqPCM_mech	0	rpm	1000
omega_required_mech	600	rpm	1000
u_dc_bus	295	V	1000
theta_commutation_el	157	el. degree	1000
r_phase_actual	24.2	ohms	1000



Name	Value	Unit	Period
knock_time	10592	clock cycles	200
benchmark_time	0,172135	sec	200
CPU_load	0,549479	%	1000
engine_speed	2000	RPM	1000
knock_window_size	15	degrees	1000
num_of_cylinders	4	DEC	1000
num_of_samples	250	DEC	1000
filter	FIR 22TAP	ENUM	1000

RS232;COM1;speed=115200


knock_demo.pmp - FreeMaster
Continuous mode
Reload
Input Signal [us]
Input Signal Spectrum [kHz]
FIR Output Signal [us]
FIR Output Signal Spectrum [kHz]
Integrator Output Signal [us]
Lock X-axis zooming of time-domain graphs
Lock X-axis zooming of frequency-domain
control page
algorithm block description

Name	Value	Unit	Period
input_signal	EQADC signal	ENUM	1000
rect_signal_freq_hz	1500	Hz	1000
filter	FIR 22TAP	ENUM	1000
integrator_time_const	11	DEC	1000

RS232;COM1;speed=115200

FreeMASTER as a Real Time Monitor

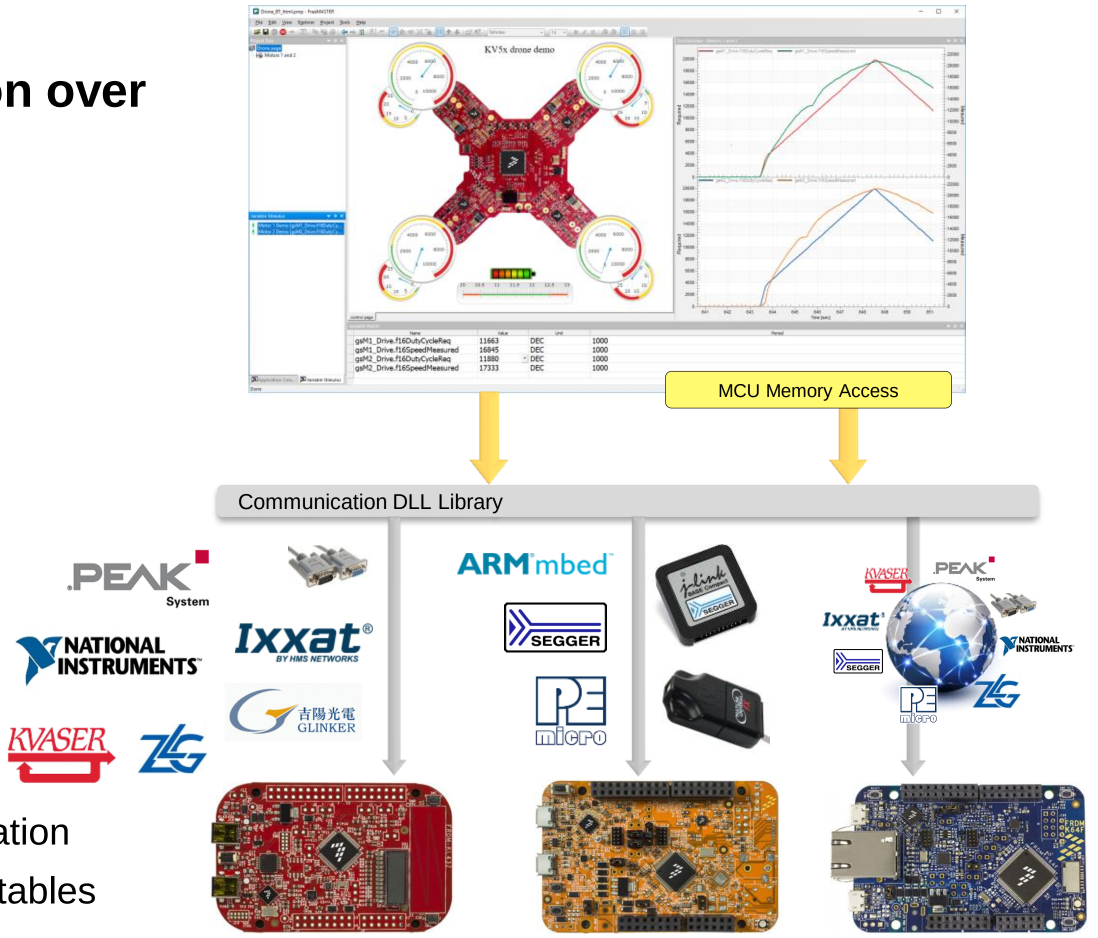
of

Internal variables
Processes & algorithms
Application states

FreeMASTER as a Real-time Monitor

- **FreeMASTER connects to target embedded application over**
  - **UART s**erial communication (SCI, LPUART, UART, USART, …)
  - **USB-CDC** – virtual serial line
  - **CAN** – msCAN, FlexCAN, MCAN
  - **SWD/JTAG/BDM** – non-intrusive access using debugger interface
  - **JTAG/EOnCE** – real time data exchange port of 56F8xxx family
  - Any of the above remotely over the network using FreeMASTER Remote Server or JSON-RPC API
  - Any of the above from 3rd party application using ActiveX
  - Any of the above using unified Communication DLL

- **…and enables access to application memory**
  - Parses ELF file and extracts DWARF debugging and symbolic information
  - Reads symbolic information from runtime Target-Side Address (TSA) tables
  - Knows global and static variables address, type and size
  - Knows complex data types (structures and arrays)



Name	Value	Unit	Period
gsM1_Drive.f16DutyCycleReq	11663	DEC	1000
gsM1_Drive.f16SpeedMeasured	16845	DEC	1000
gsM2_Drive.f16DutyCycleReq	11880	DEC	1000
gsM2_Drive.f16SpeedMeasured	17333	DEC	1000

Connect over UART, USB-CDC or CAN

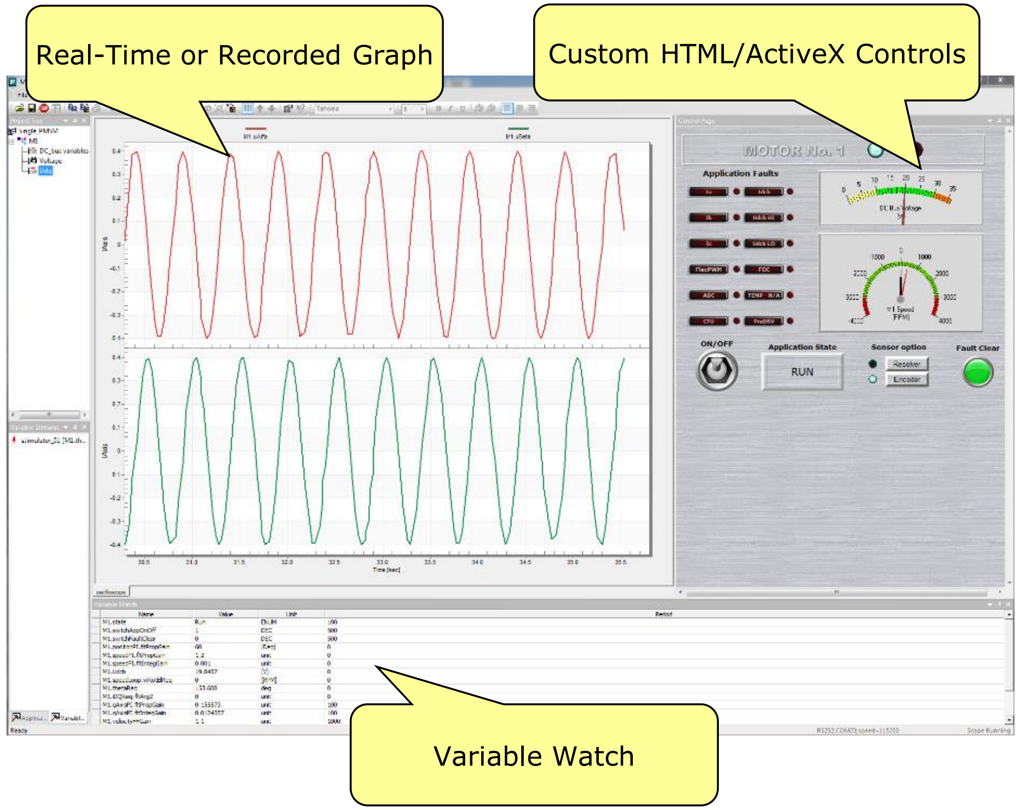
Direct memory access using j-Link, CMSIS-DAP or P&E

Share any connection over the internet

FreeMASTER as a Real-time Monitor

Display the variable values in various formats:

- **Table View**
  - variable values updated at specified rate
- **Real-Time Chart**
  - values read in real time and plotted in oscilloscope-like waveform
- **High-Speed Recorded Graph**
  - values recorded by on-board memory **transient recorder** and plotted all at once
- **Custom ActiveX/HTML Controls**
  - values displayed ActiveX/HTML controls (sliders, gauges, etc.)



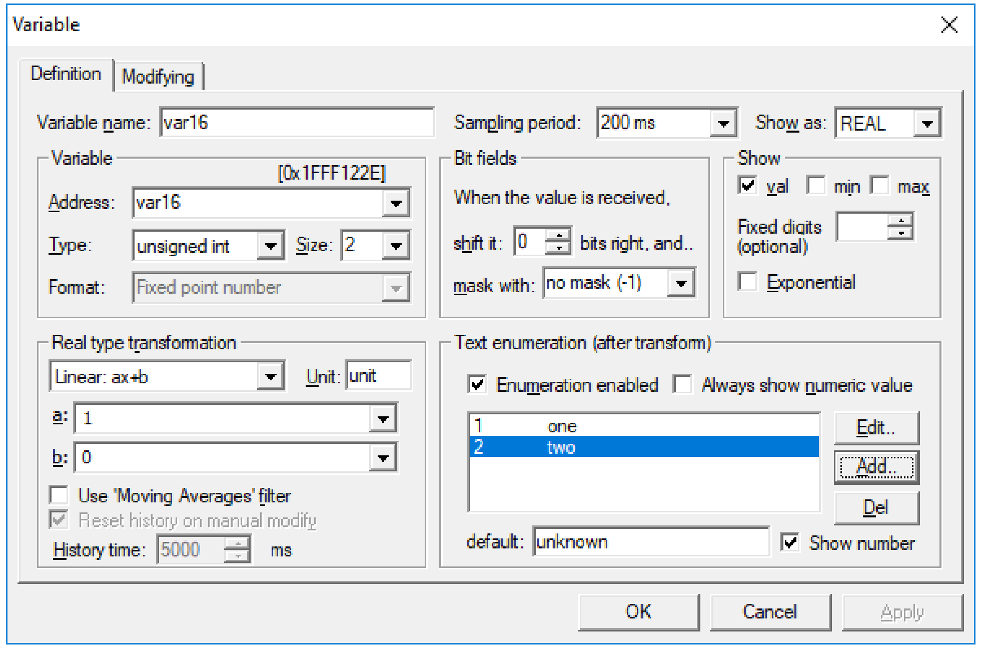
FreeMASTER as a Real-time Monitor

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)

- User TSA tables 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 name: var16
Sampling period: 200 ms
Show as: REAL

Variable [0x1FFF122E]
Address: var16
Type: unsigned int
Size: 2
Format: Fixed point number

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
Linear: ax+b
Unit: unit
a: 1
b: 0
[ ] Use 'Moving Averages' filter
[✓] Reset history on manual modify
History time: 5000 ms

Text enumeration (after transform)
[✓] Enumeration enabled [ ] Always show numeric value

1	one
2	two

default: unknown
[✓] Show number

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 as it requires target-side driver. Does not work over direct SWD/JTAG/BDM.

FreeMASTER as a Real-time Monitor

Anatomy of the desktop window

Project tree:
- Custom project hierarchy
- Folder-like block items
- Oscilloscope graph items
- Recorder graph items
- Pipe console items

Variable stimulus
- Stimulator time tables

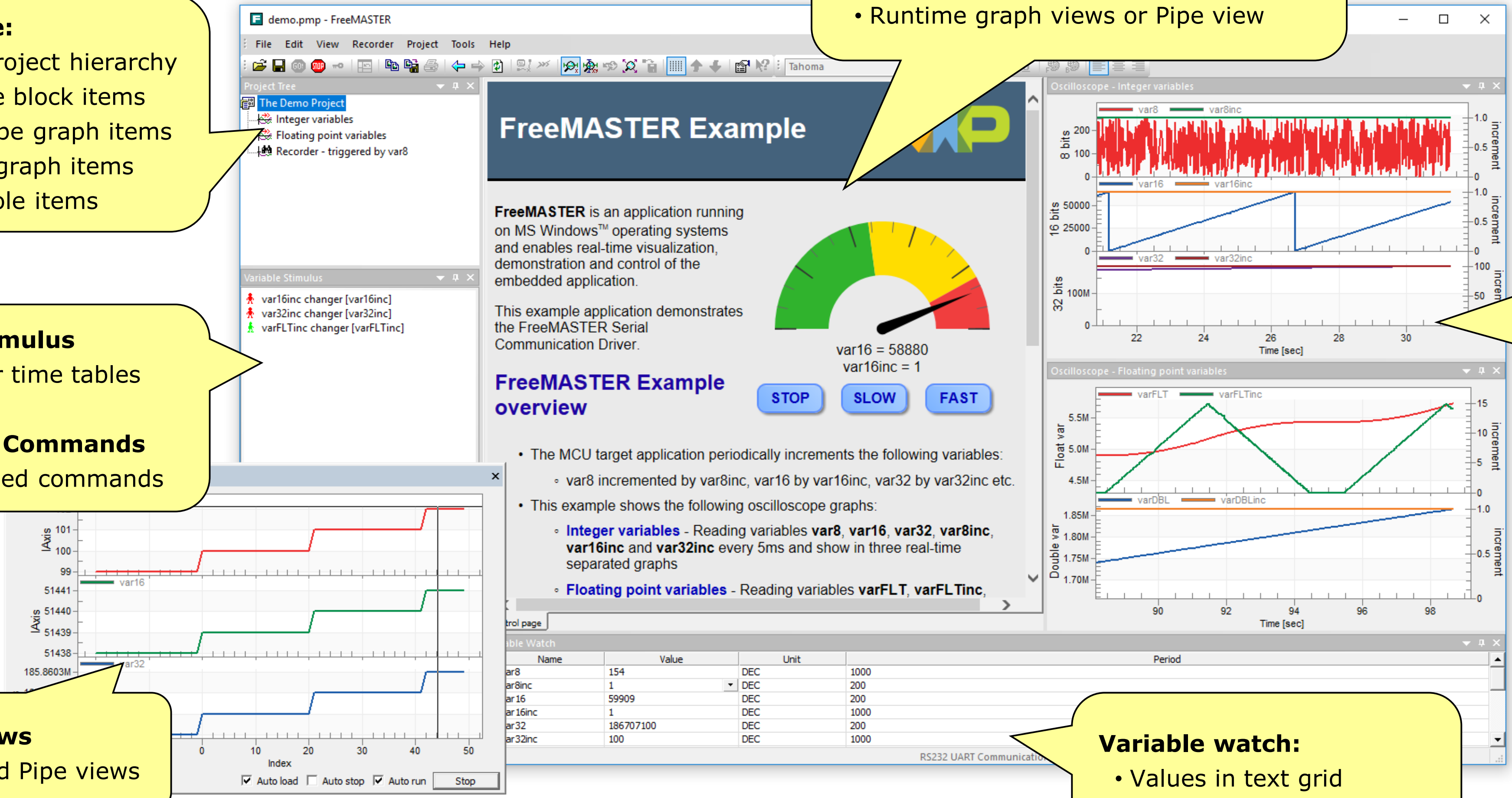
Application Commands
- User-defined commands

Floating views
- Graphs and Pipe views

Main tabbed pane:
- Custom control UI page in HTML
- Selected tree item description in HTML
- Runtime graph views or Pipe view

Docked views
- Graphs and Pipe views
- Docked to window side
- Auto-hide option
- Overlap with other views in a tabbed pane

Variable watch:
- Values in text grid
- Variables assigned to selected "block" tree item



FreeMASTER Example

FreeMASTER is an application running on MS Windows™ operating systems and enables real-time visualization, demonstration and control of the embedded application.

This example application demonstrates the FreeMASTER Serial Communication Driver.

FreeMASTER Example overview

- The MCU target application periodically increments the following variables:
  - var8 incremented by var8inc, var16 by var16inc, var32 by var32inc etc.
- This example shows the following oscilloscope graphs:
  - **Integer variables** - Reading variables **var8**, **var16**, **var32**, **var8inc**, **var16inc** and **var32inc** every 5ms and show in three real-time separated graphs
  - **Floating point variables** - Reading variables **varFLT**, **varFLTinc**,

Name	Value	Unit	Period
ar8	154	DEC	1000
ar8inc	1	DEC	200
ar16	59909	DEC	200
ar16inc	1	DEC	1000
ar32	186707100	DEC	200
ar32inc	100	DEC	1000

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
- Direct memory access with J-Link, P&E Multilink or CMSIS-DAP, no target-side driver needed
- Addresses are parsed from ELF file or provided by user-defined TSA tables
- Enables fine tuning parameters or direct application control via variable modifications
- Oscilloscope graphs with real time data in [ms] resolution
- On-board Recorder visualization of transitions in [µs] resolution

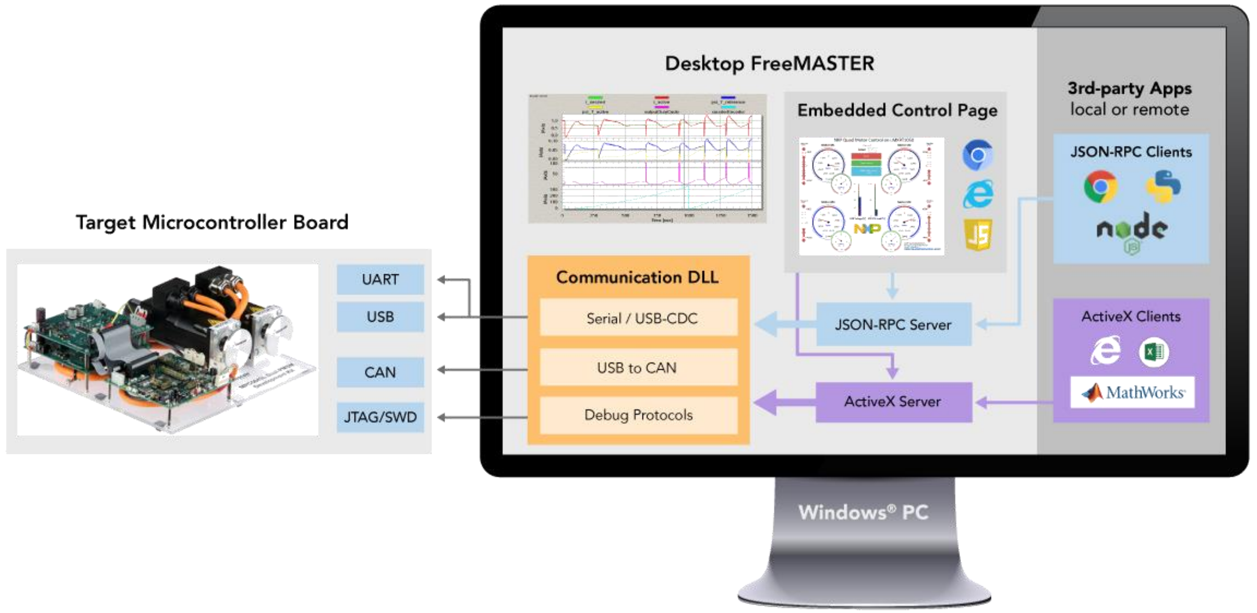
FreeMASTER as a Control GUI

Rendering HTML-encoded GUI
Native JavaScript scripting
ActiveX or JSON-RPC interface
3rd party applications and scripts

FreeMASTER as a Control GUI

- **Control Application by Variable Modification**
  - Manually in the Variable Watch table view
  - Time-tables & stimuli modification
  - Script-based variable access directly from custom GUI
    - Handle mouse-clicks and keyboard control
    - Custom graphics, 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
    - Ideal for Hardware-in-the-loop simulations
  - JSON-RPC clients using WebSocket or TCP
    - Standalone web apps running in Chrome
    - Node.js, Python and other scripts
  - Works over UART, CAN and also with non-intrusive SWD/JTAG/BDM direct memory access.

- **Control by Sending "Application Commands"**
  - "Traditional" control approach
  - Scripts and 3rd party applications support
  - Only works when target driver is used (UART, CAN, PDBDM)
  - Does not work with direct-access SWD/JTAG/BDM



MathWorks and MATLAB are trademarks or registered trademarks of The MathWorks,Inc.
TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.

FreeMASTER as a Control GUI

- **HTML Content inside FreeMASTER**
  - HTML views embedded in the FreeMASTER window
  - Dedicated “control page” view always accessible
  - Optional “description” views showing HTML content based on project tree selection
  - Internet Explorer or Chromium rendering engines

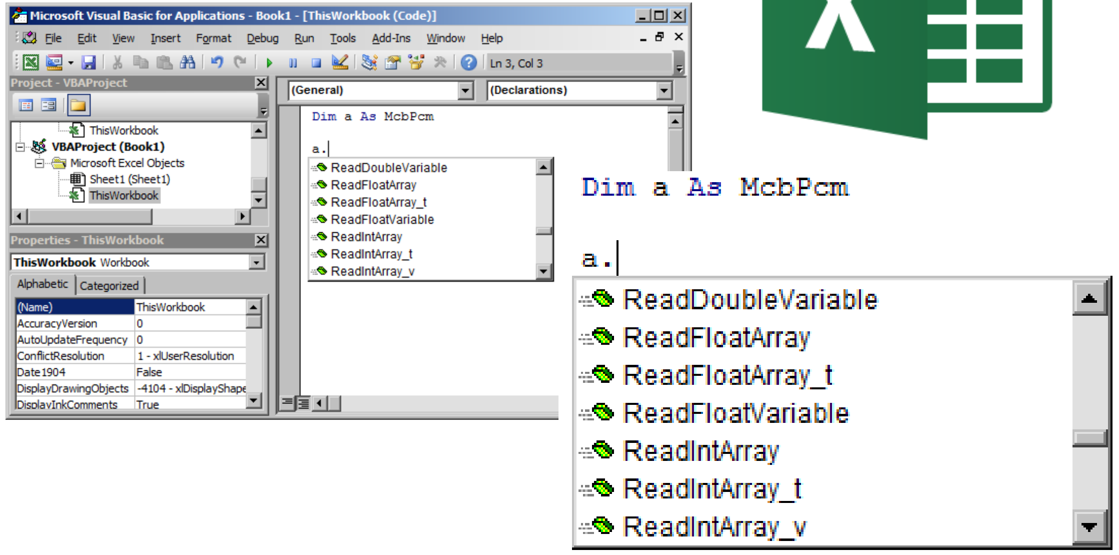
- **Scripting in FreeMASTER**
  - JavaScript may be natively embedded in HTML pages
  - ActiveX for Internet Explorer JavaScript, VBA Excel, etc.
  - JSON-RPC for Chrome JavaScript or 3rd party scripts Node.js, Python, Matlab, Octave, etc.
  - JSON-RPC over WebSocket or direct TCP
  - JSON-RPC also used with **FreeMASTER Lite** service

- **FreeMASTER Scripting Methods**
  - Read/write access using existing variable objects
  - Direct read/write memory access using C symbol names
  - Retrieving symbol and data type information
  - Project tree manipulation and navigation
  - Oscilloscope or Recorder data handling
  - Pipe communication
  - Project file and communication port control
  - Local file read/write access
  - Variable Stimulator control
  - Sending Application Commands

Refer to FreeMASTER User Guide for full script API reference and examples using JScript, VBScript, VBA Excel and Matlab

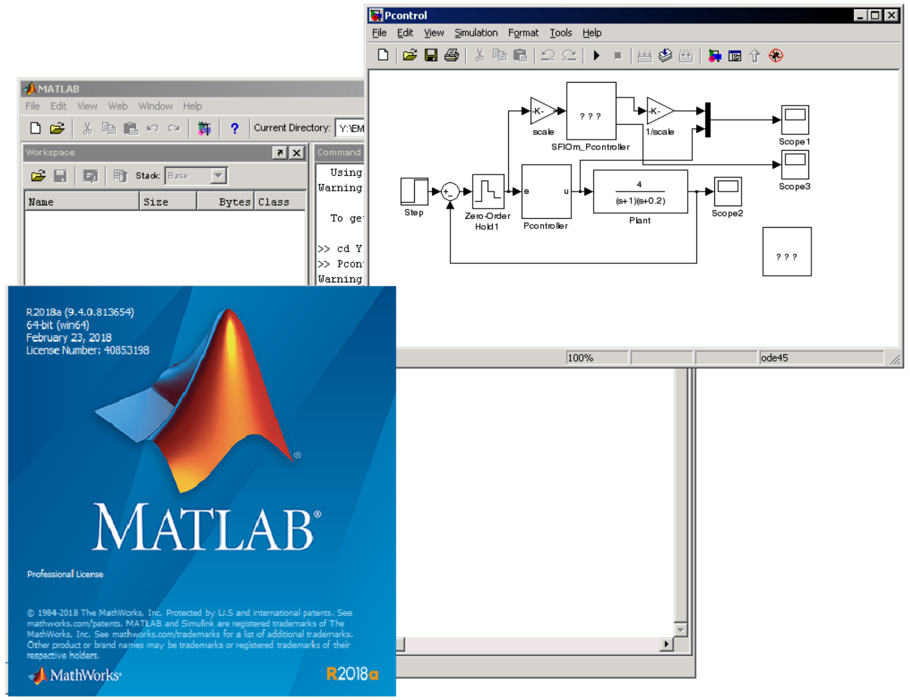
FreeMASTER as a Control GUI

- **Target-in-loop Simulations**
  - FreeMASTER ActiveX object or JSON-RPC interface is accessible by external standalone applications
  - Standard C++ or VB applications
  - Excel & Visual Basic for Applications
  - Octave, Matlab & Simulink



(Name)	ThisWorkbook
AccuracyVersion	0
AutoUpdateFrequency	0
ConflictResolution	1 - xlUserResolution
Date1904	False
DisplayDrawingObjects	-4104 - xlDisplayShape
DisplayInkComments	True

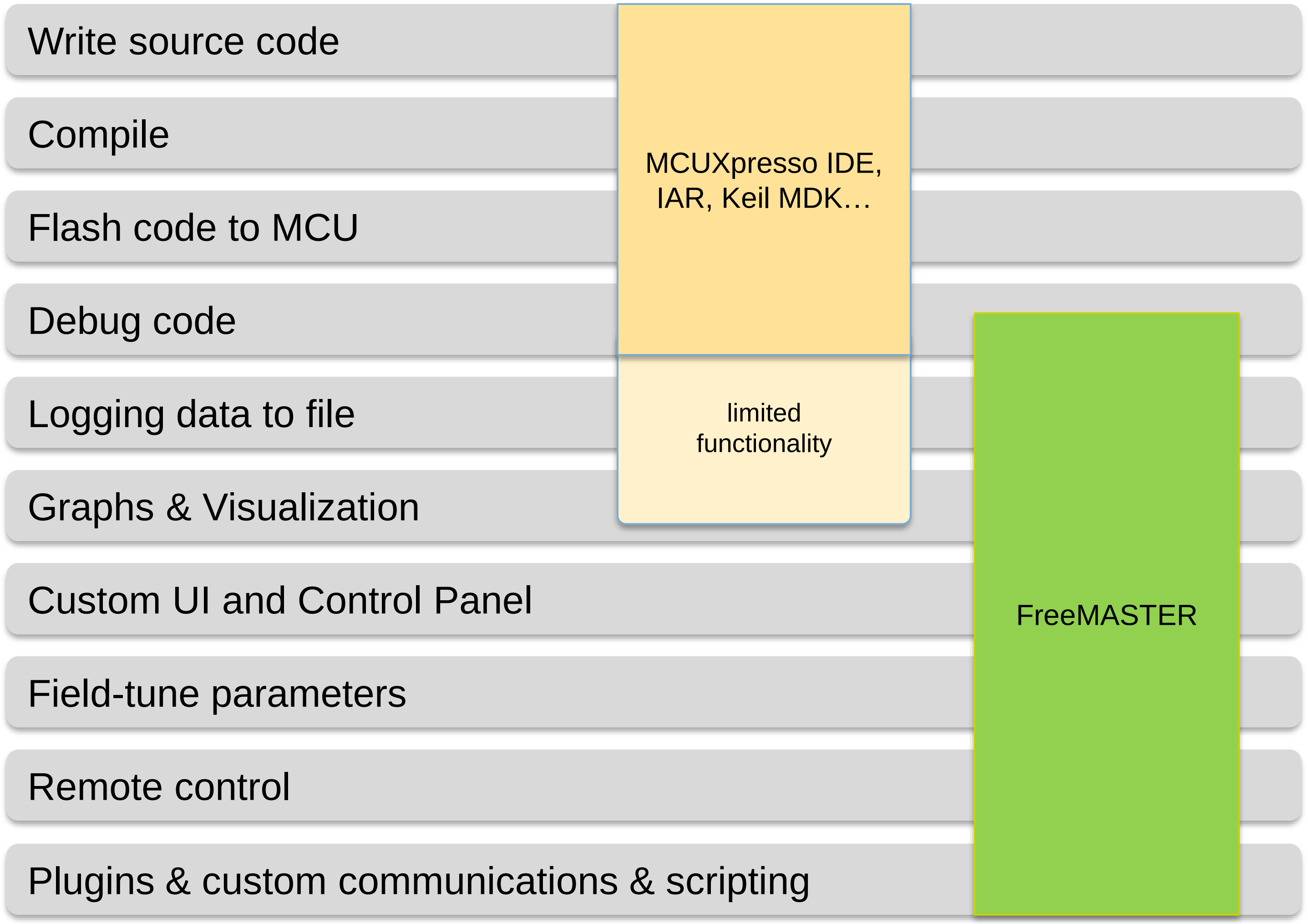
```
Dim a As McbPcm

a.
ReadDoubleVariable
ReadFloatArray
ReadFloatArray_t
ReadFloatVariable
ReadIntArray
ReadIntArray_t
ReadIntArray_v
```



```
Using
Warning
To ge
>> cd Y
>> Pcon
Warning
```

R2018a (9.4.0.813654)
64-bit (win64)
February 23, 2018
License Number: 40853198

FreeMASTER vs. Compiler / Debugger



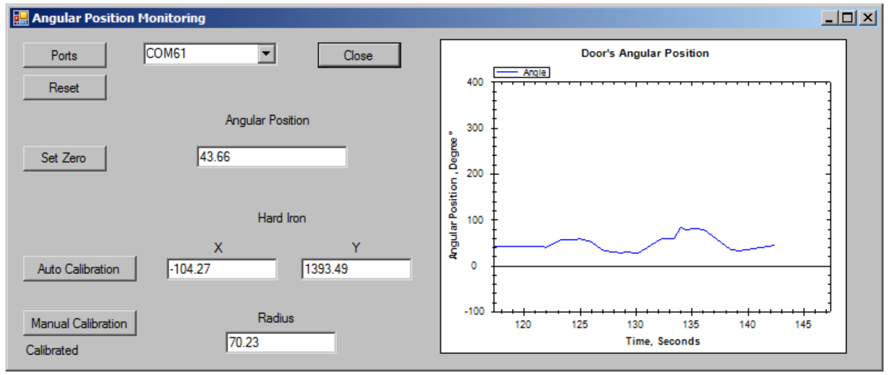
FreeMASTER Replacing Custom GUI Applications

FreeMASTER vs. custom GUI development
Typical use cases

From Custom GUI to FreeMASTER

- **Pitfalls of developing custom GUI**
  - Requires PC Host programming tools and skills
  - Never enough communication interfaces, communication issues
  - Time to develop a robust PC Host application
  - Deploying GUI to host PC
  - Single-purpose – typically bound to particular MCU demo app. firmware

- **Benefits of FreeMASTER**
  - Uniform approach – application control by variable modification
  - Works over UART/CAN but also over non-intrusive SWD/JTAG/BDM
  - One tool used with variety of GUIs
  - GUI easily extended by multimedia content (charts, documentation) local, online or embedded
  - Usable with user-modified applications **(!)**
  - GUI project can be extended by user to cover more functionality

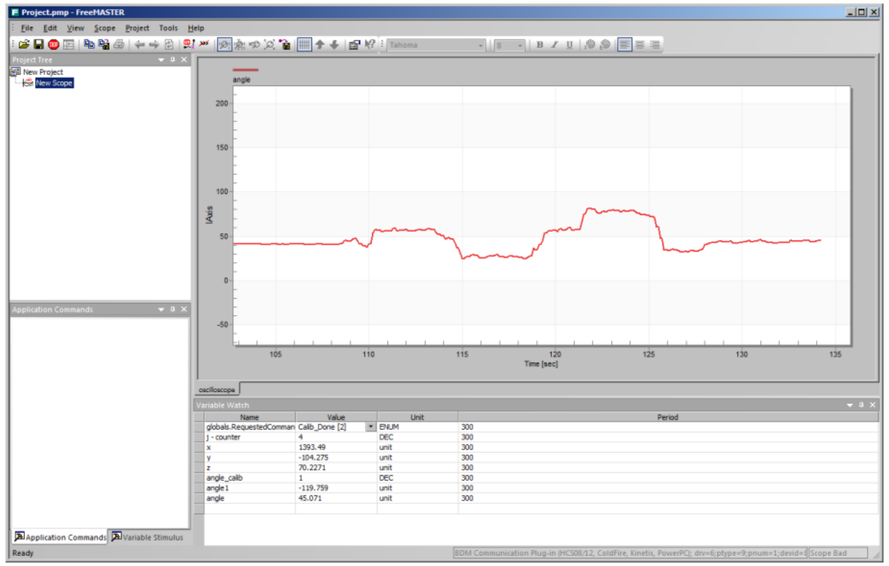


COM61

Angular Position: 43.66

Hard Iron — X: -104.27, Y: 1393.49

Radius: 70.23



Variable Watch

Name	Value	Unit	Period
globals.RequestedComman	Calib_Done [2]	ENUM	300
j - counter	4	DEC	300
x	1393.49	unit	300
y	-104.275	unit	300
z	70.2271	unit	300
angle_calib	1	DEC	300
angle1	-119.759	unit	300
angle	45.071	unit	300

BDM Communication Plug-in (HCS08/12, ColdFire, Kinetis, PowerPC); drv=6;ptype=9;pnum=1;devid=[| Scope Bad

From Custom GUI to FreeMASTER

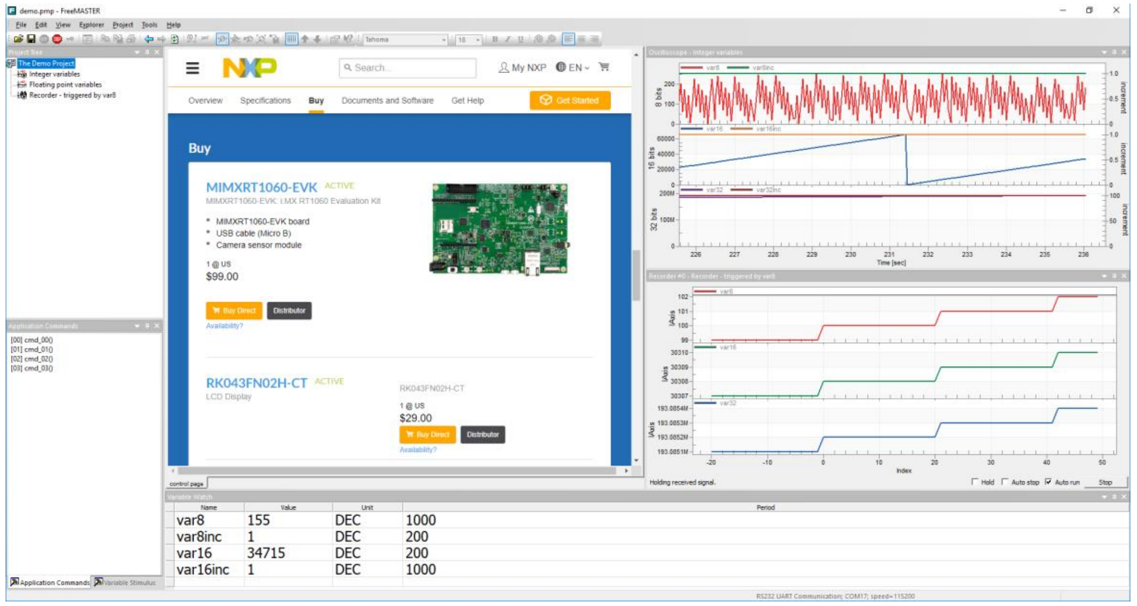
- **Typical custom GUI Approach**
  - Communication-driven data collection
  - Typically custom serial protocol
  - PC sends request, target processes and replies with data
    - **Pro**: under full control of developer
    - **Pro**: application and data processing logic done in native programming language
    - **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 target application not supported by single-purpose GUI.
    - **Con** : user modifications of target application or firmware makes the GUI to stop working.

- **FreeMASTER Approach**
  - Rich graphing and visualization features instantly available
  - Control application by modifying variables
  - Use either artificial variables dedicated for GUI control or modify state variables used also by the general application algorithm
    - **Pro** : works over standardized protocol or with SWD/JTAG/BDM direct memory access
    - **Pro** : easy to protect or restrict functionality

FreeMASTER as a Demonstration Tool

FreeMASTER as a Demonstration Tool

- Project GUI is a native HTML page with all benefits of online Internet resources
- Embed multimedia content and online shop links directly in the control UI
- Project hierarchy enables to describe all aspects of target application accompanied with live data
- NXP customers use FreeMASTER to demonstrate their products
- Complimentary use with any NXP MCU system



Buy

MIMXRT1060-EVK ACTIVE
MIMXRT1060-EVK: i.MX RT1060 Evaluation Kit

- MIMXRT1060-EVK board
- USB cable (Micro B)
- Camera sensor module

1 @ US
$99.00

RK043FN02H-CT ACTIVE
LCD Display

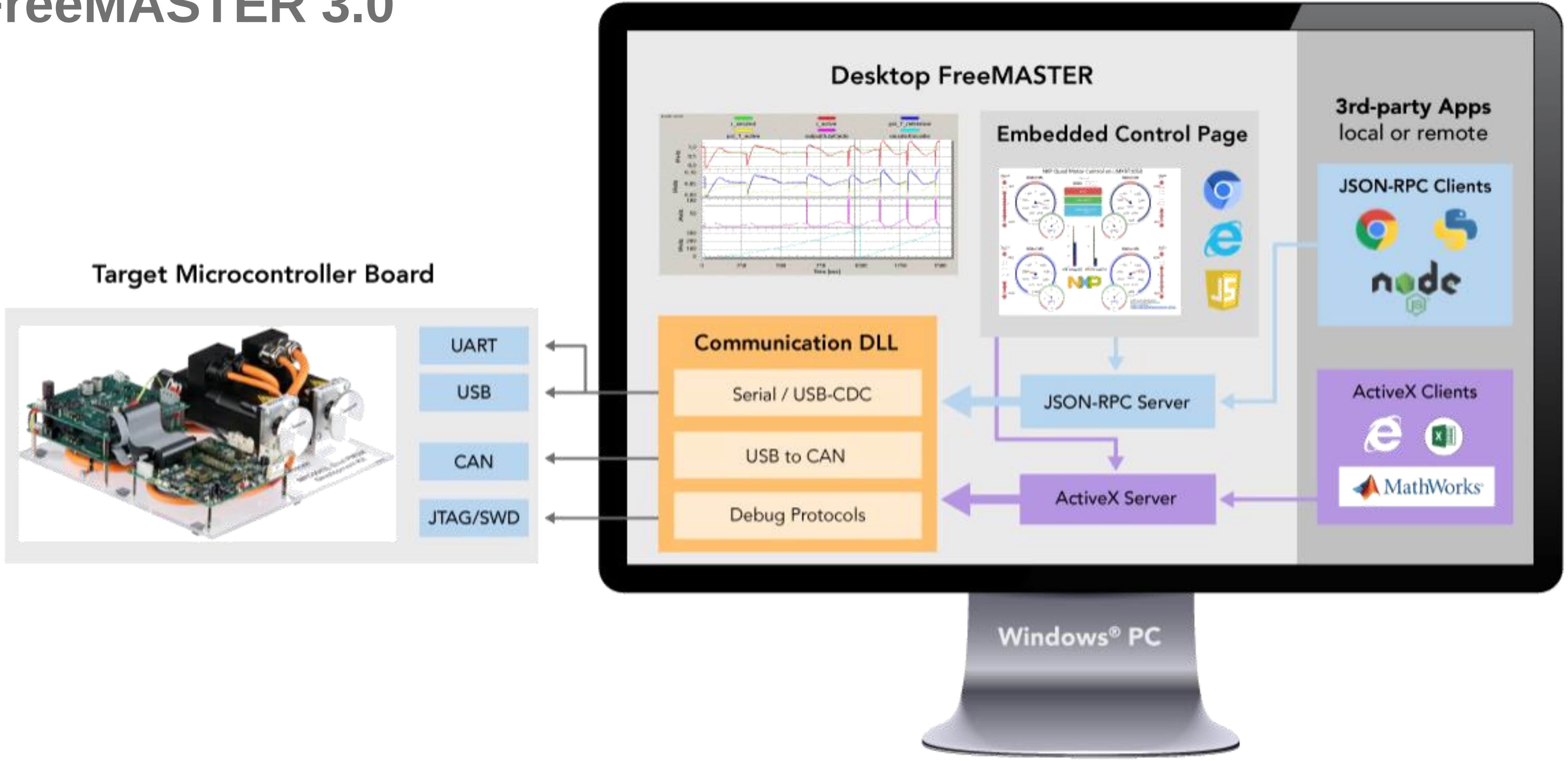
RK043FN02H-CT
1 @ US
$29.00

Variable Watch

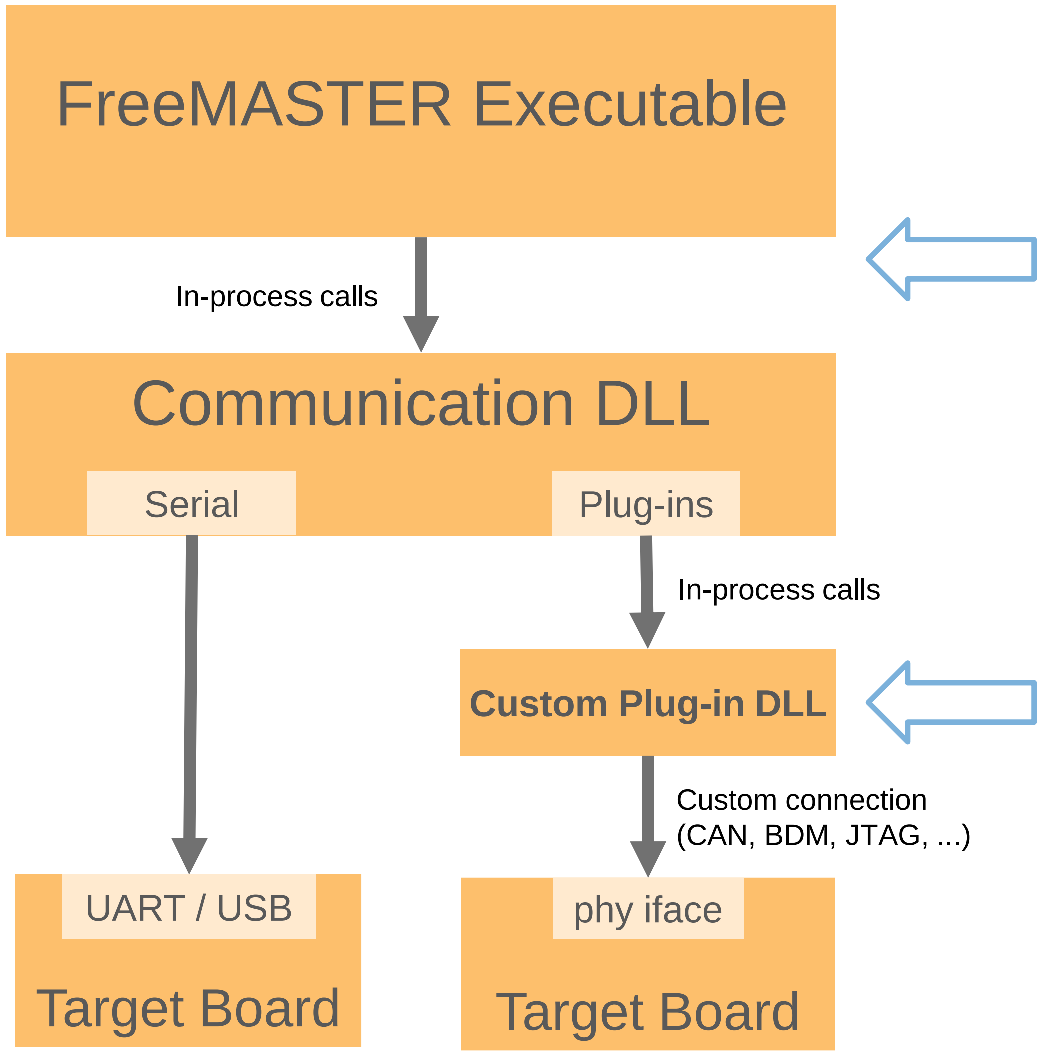
Name	Value	Unit	Period
var8	155	DEC	1000
var8inc	1	DEC	200
var16	34715	DEC	200
var16inc	1	DEC	1000

Inside FreeMASTER

FreeMASTER 3.0



Inside FreeMASTER – Communication Flow



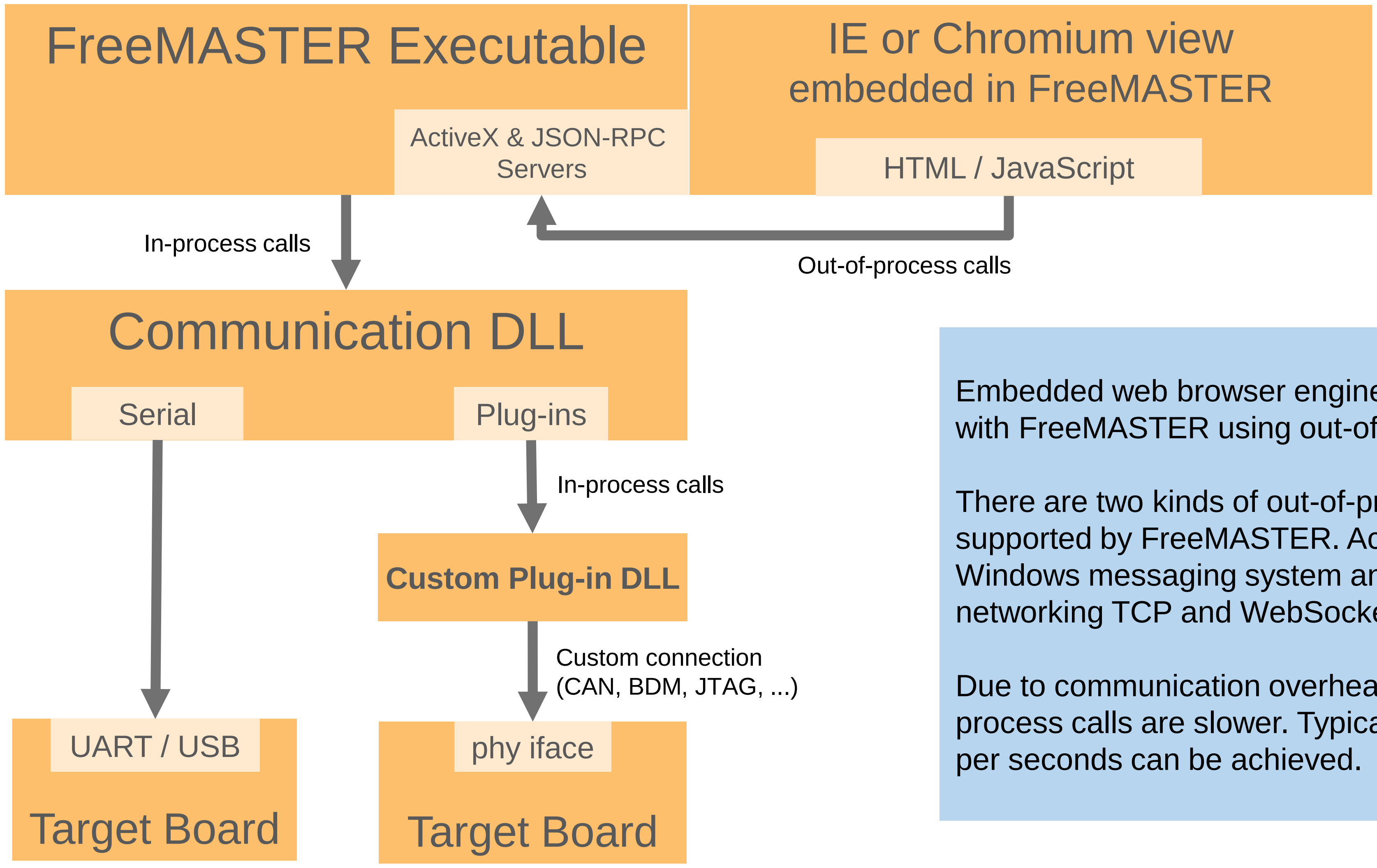
So-called “in-process calls” are simple calls to functions located in dynamically loaded libraries (DLLs). The calls are fast just like if the function would be located inside the executable itself.

FreeMASTER enables to use custom plug-in modules to implement the communication layer.

NXP delivers several plug-ins in the standard FreeMASTER distribution. CAN, BDM and other connections are possible. Plug-ins are not always fully featured; for example a direct memory SWD/JTAG/BDM plug-in allows memory reads and writes, not the recorder or TSA feature.

FreeMASTER plug-ins use Microsoft COM+ procedure call standard. Both in-process DLL (typical) and out-of-process EXE plug-ins are supported.

Inside FreeMASTER – Embedded Web Browser

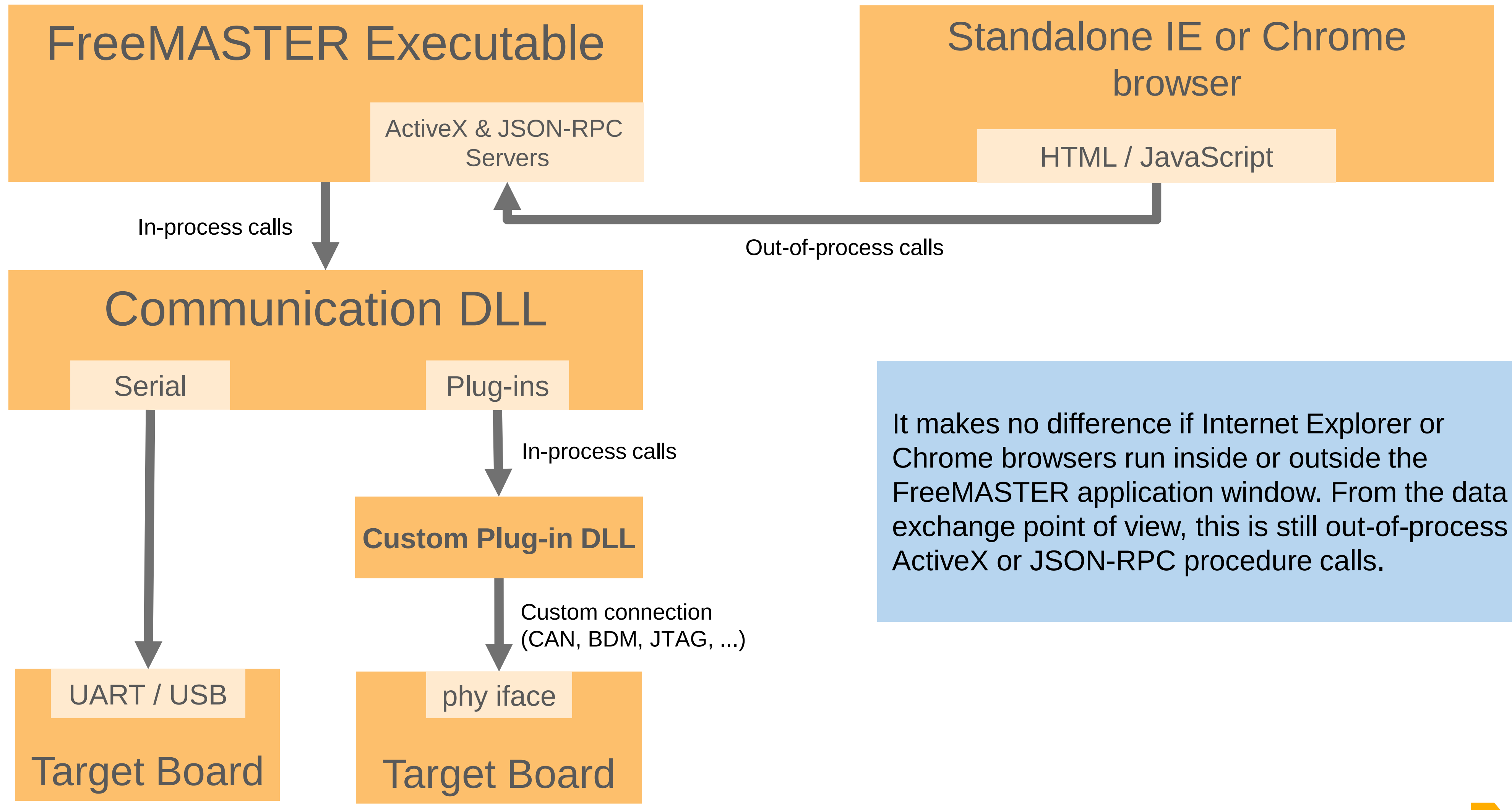


Embedded web browser engine communicates with FreeMASTER using out-of-process calls.

There are two kinds of out-of-process calls supported by FreeMASTER. ActiveX uses Windows messaging system and JSON-RPC use networking TCP and WebSocket technologies.

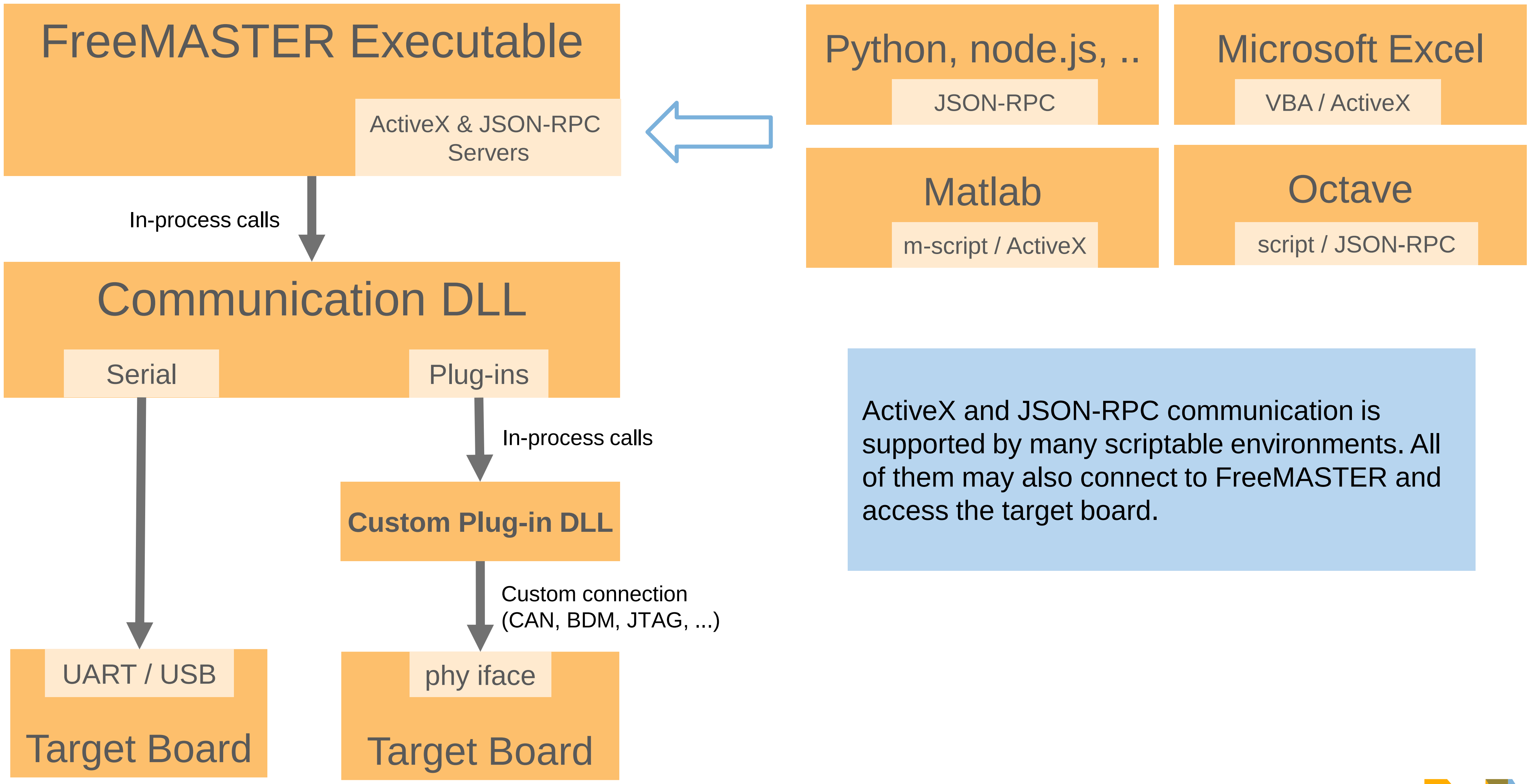
Due to communication overhead, the out-of-process calls are slower. Typically up to 500 calls per seconds can be achieved.

Inside FreeMASTER – Standalone Web Browser



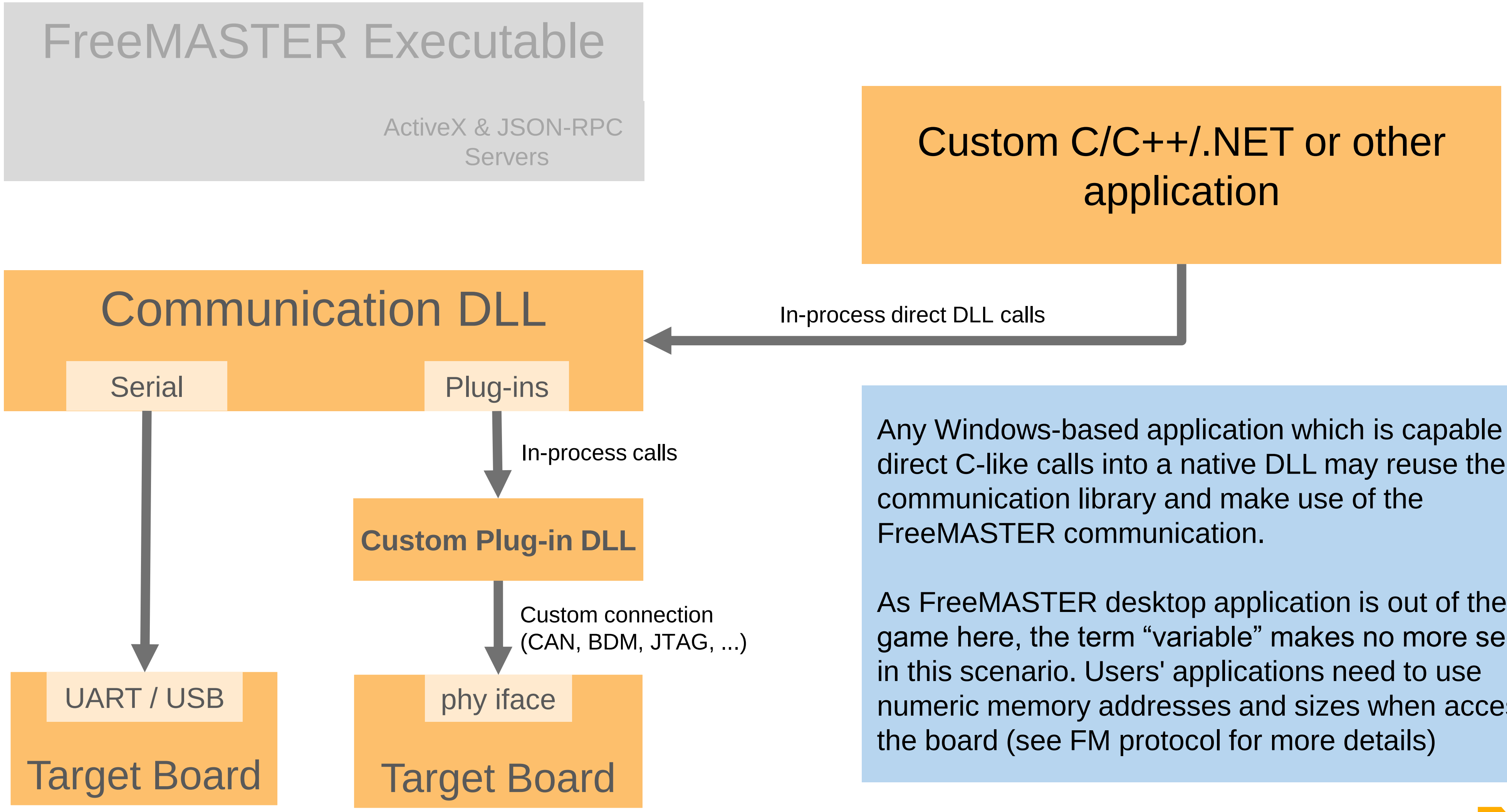
It makes no difference if Internet Explorer or Chrome browsers run inside or outside the FreeMASTER application window. From the data exchange point of view, this is still out-of-process ActiveX or JSON-RPC procedure calls.

Inside FreeMASTER – Other Client Applications



ActiveX and JSON-RPC communication is supported by many scriptable environments. All of them may also connect to FreeMASTER and access the target board.

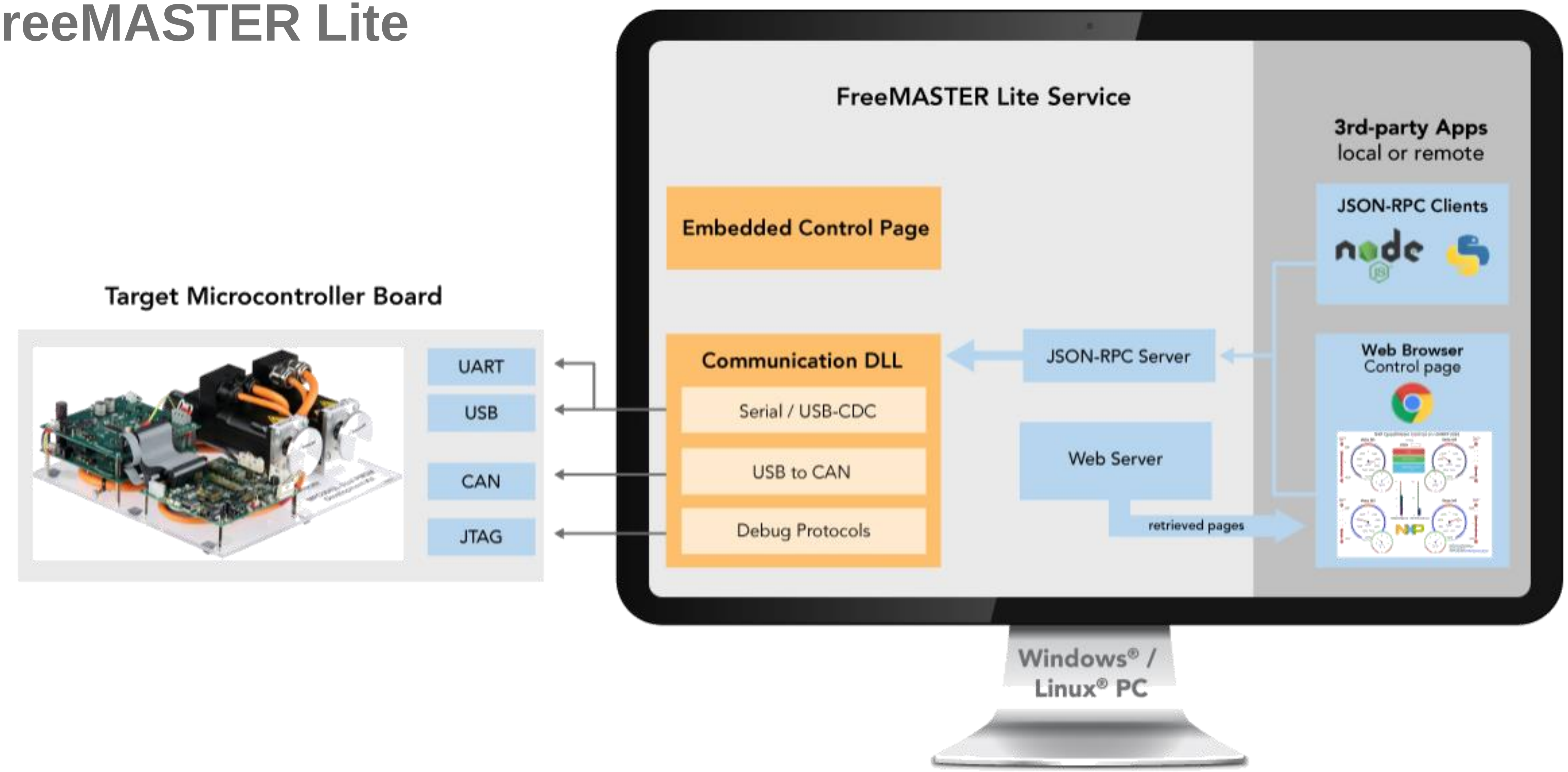
Inside FreeMASTER – Interfacing to Communication DLL


FreeMASTER Executable
ActiveX & JSON-RPC Servers
Custom C/C++/.NET or other application
In-process direct DLL calls
Communication DLL
Serial
Plug-ins
In-process calls
Custom Plug-in DLL
Custom connection (CAN, BDM, JTAG, ...)
UART / USB
Target Board
phy iface
Target Board

Any Windows-based application which is capable of direct C-like calls into a native DLL may reuse the communication library and make use of the FreeMASTER communication.

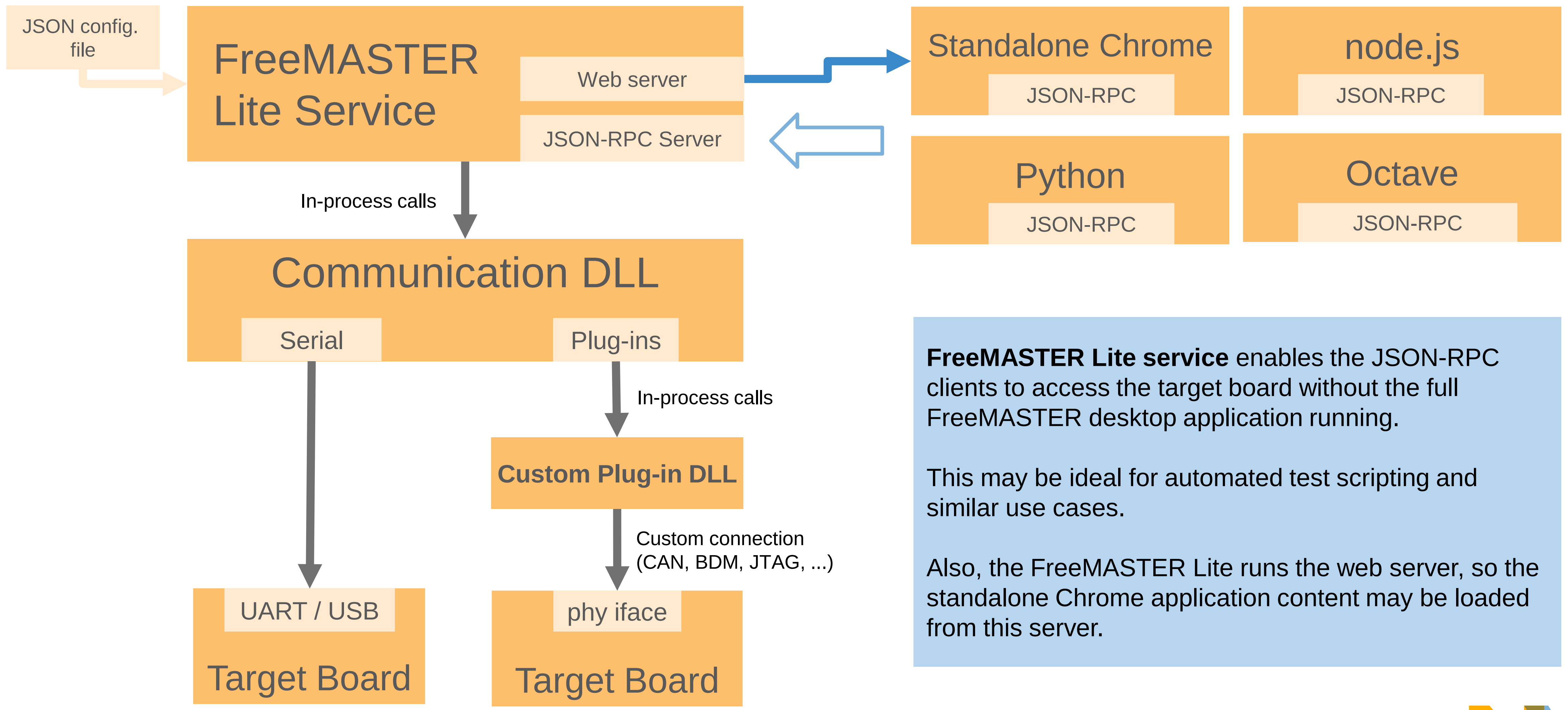
As FreeMASTER desktop application is out of the game here, the term “variable” makes no more sense in this scenario. Users' applications need to use numeric memory addresses and sizes when accessing the board (see FM protocol for more details)

FreeMASTER Lite



TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.

FreeMASTER Lite – Service without User Interface



FreeMASTER Lite service enables the JSON-RPC clients to access the target board without the full FreeMASTER desktop application running.

This may be ideal for automated test scripting and similar use cases.

Also, the FreeMASTER Lite runs the web server, so the standalone Chrome application content may be loaded from this server.

SECURE CONNECTIONS
FOR A SMARTER WORLD