Introduction to FreeMASTER Dashboard Coding

Using HTML, JavaScript, ActiveX and JSON-RPC

Michal Hanak
Software Engineer

MAY 2020

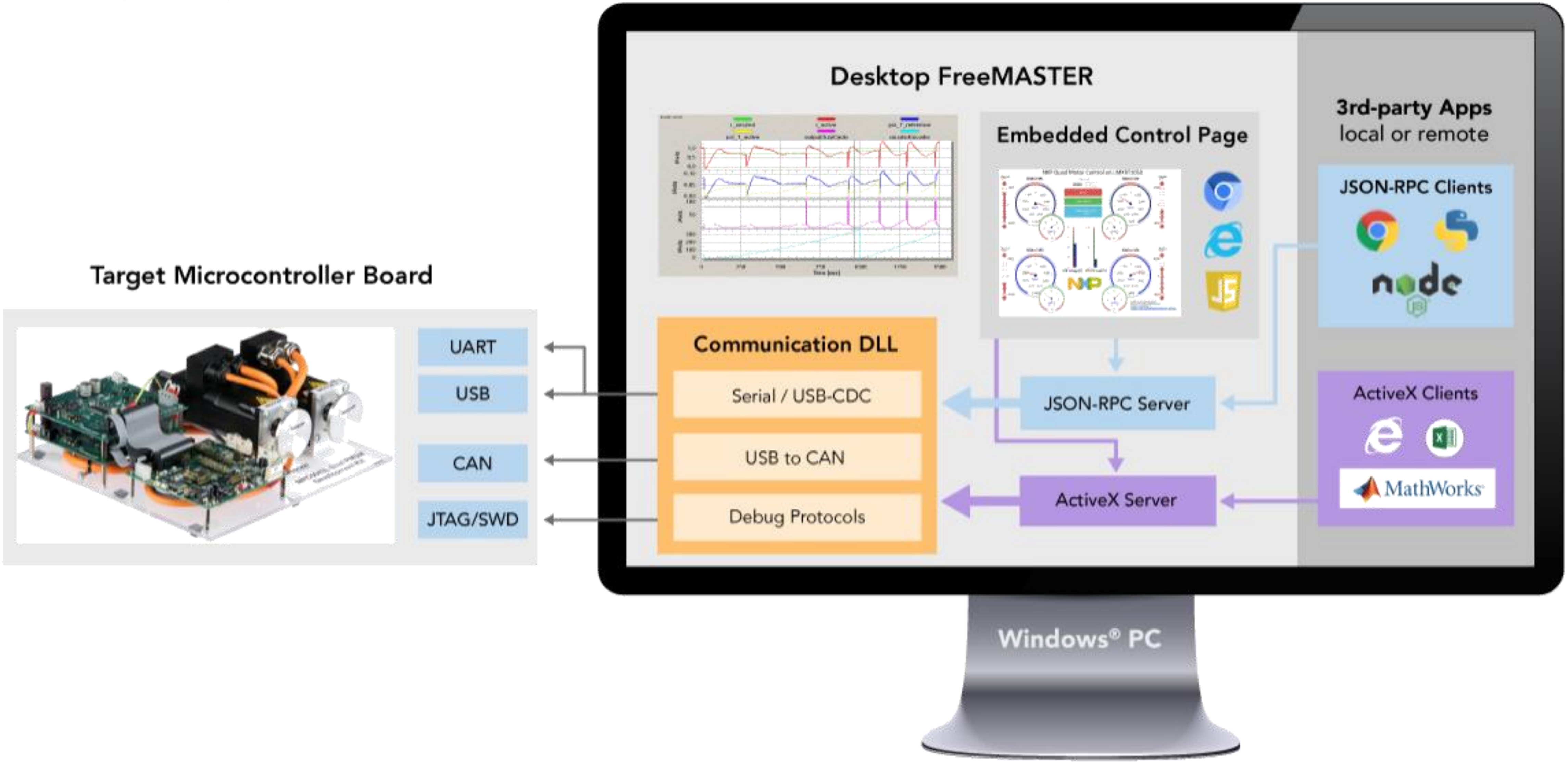
SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

PRESENTATION AGENDA

- FreeMASTER overview
- FreeMASTER and 3rd party application connectivity
- Example walk through
  - Basic dashboard using ActiveX
  - Accessing ActiveX from Excel VBA
  - Basic dashboard using JSON-RPC
  - Adding graphics, styles and more features
- Q & A

FREEMASTER 3.0



Target Microcontroller Board

UART

USB

CAN

JTAG/SWD

Desktop FreeMASTER

Embedded Control Page

Communication DLL

Serial / USB-CDC

USB to CAN

Debug Protocols

JSON-RPC Server

ActiveX Server

3rd-party Apps
local or remote

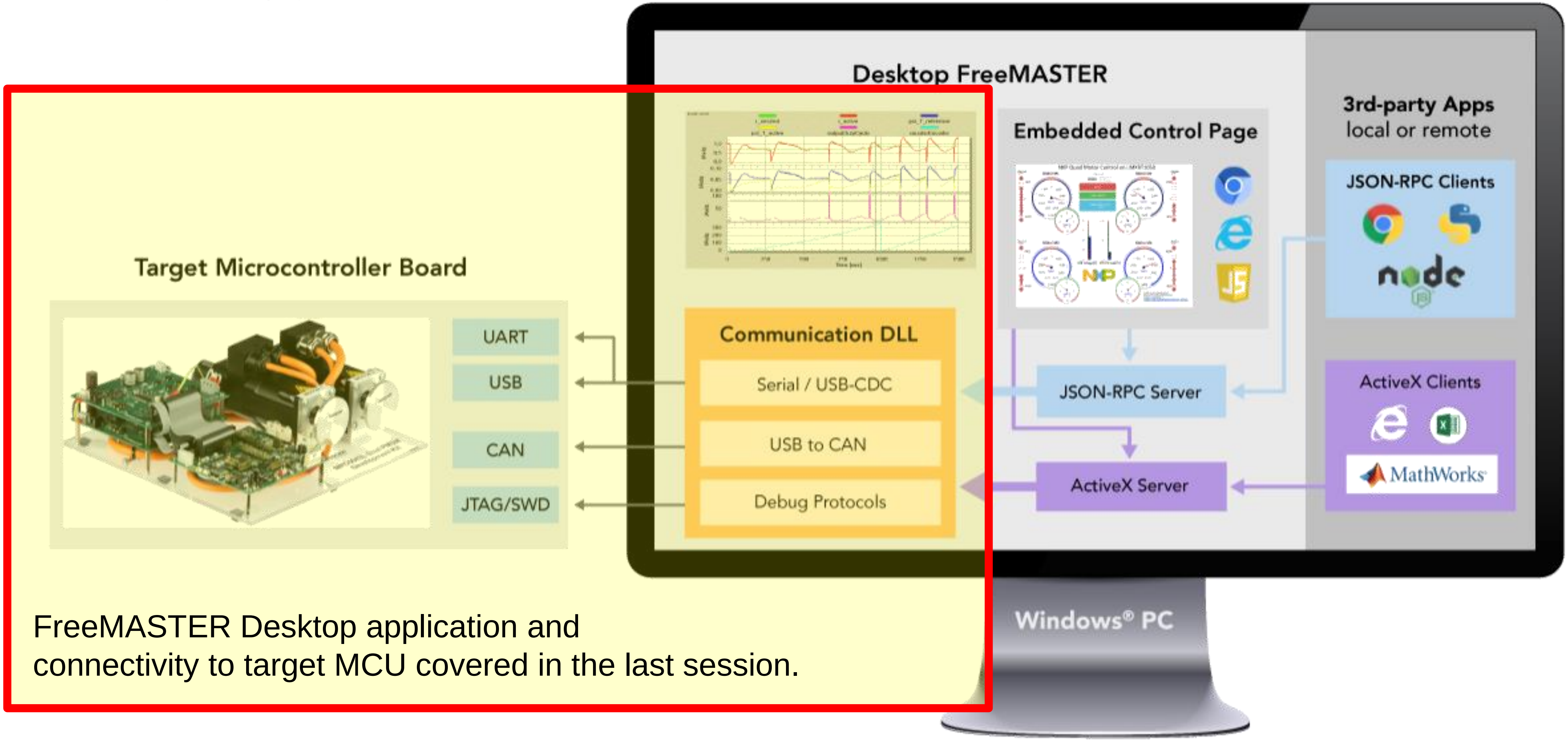
JSON-RPC Clients

ActiveX Clients

MathWorks

Windows® PC

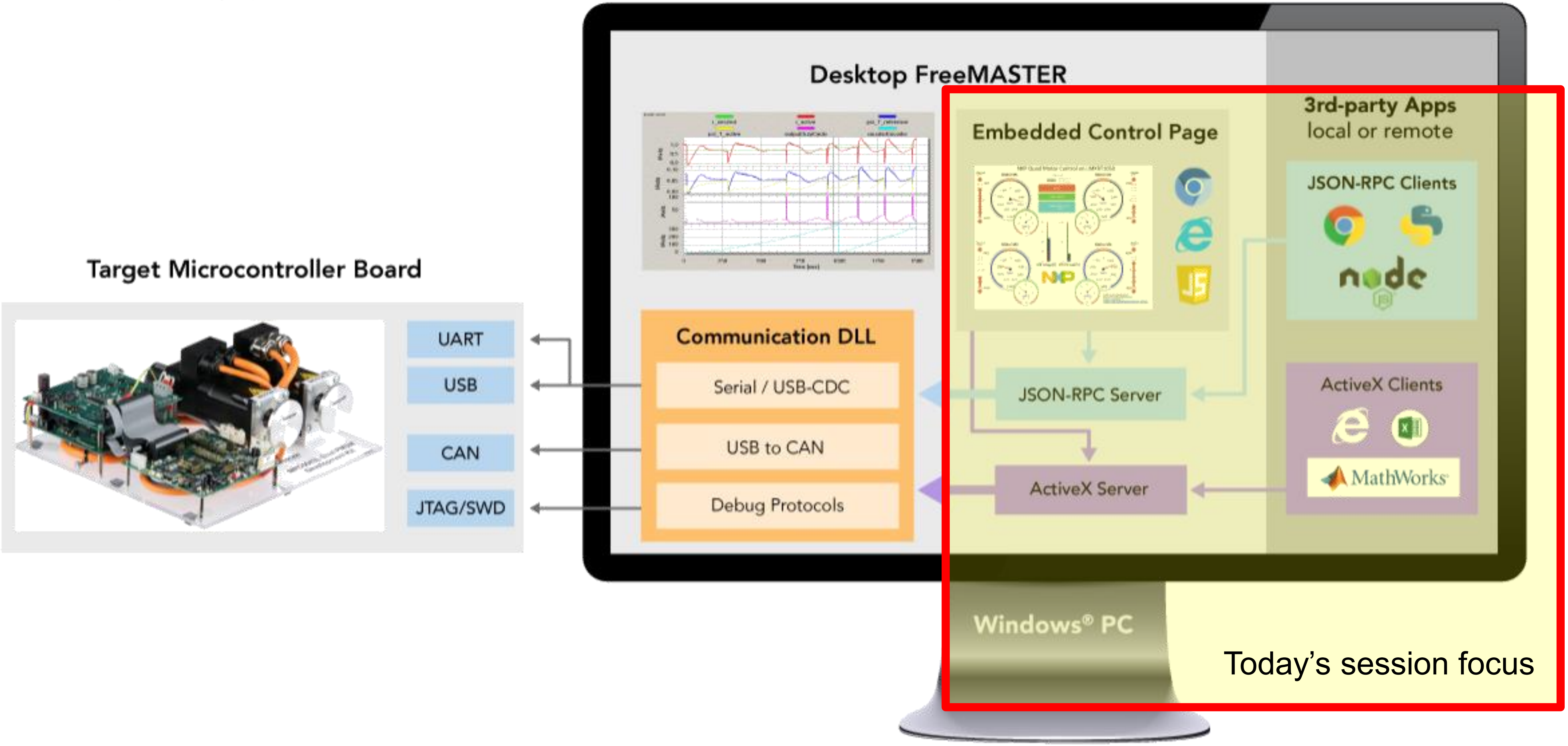
FREEMASTER 3.0



FreeMASTER Desktop application and connectivity to target MCU covered in the last session.

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 3.0



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.

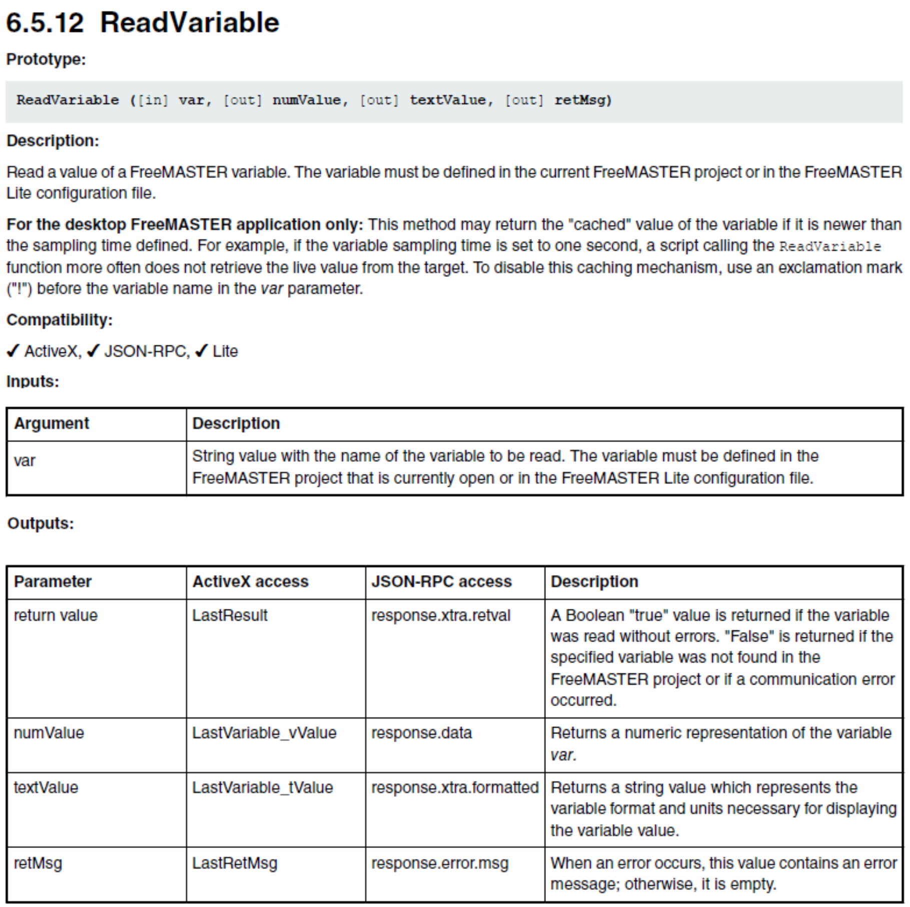
Options to connect to FreeMASTER

- **COM+ / ActiveX**
  - Backward compatible with older FreeMASTER versions since 2000
  - Default for dashboards hosted in Internet Explorer view or in standalone Internet Explorer browser
- **JSON-RPC**
  - Introduced in FreeMASTER 3.0
  - To be used by dashboards hosted in Chromium view or in standalone Chrome browser
  - Recommended for new designs

FreeMASTER RPC methods available

- Reading and writing variables or memory
- Sending application commands
- Detecting board and MCU application parameters
- Configuring and controlling the application
- Communicating using FreeMASTER pipes
- Accessing locally-stored files

- Full API reference is available in FreeMASTER User Guide



6.5.12 ReadVariable

Prototype:

```
ReadVariable ([in] var, [out] numValue, [out] textValue, [out] retMsg)
```

Description:

Read a value of a FreeMASTER variable. The variable must be defined in the current FreeMASTER project or in the FreeMASTER Lite configuration file.

For the desktop FreeMASTER application only: This method may return the "cached" value of the variable if it is newer than the sampling time defined. For example, if the variable sampling time is set to one second, a script calling the `ReadVariable` function more often does not retrieve the live value from the target. To disable this caching mechanism, use an exclamation mark ("!") before the variable name in the *var* parameter.

Compatibility:

✓ ActiveX, ✓ JSON-RPC, ✓ Lite

Inputs:

Argument	Description
var	String value with the name of the variable to be read. The variable must be defined in the FreeMASTER project that is currently open or in the FreeMASTER Lite configuration file.

Outputs:

Parameter	ActiveX access	JSON-RPC access	Description
return value	LastResult	response.xtra.retval	A Boolean "true" value is returned if the variable was read without errors. "False" is returned if the specified variable was not found in the FreeMASTER project or if a communication error occurred.
numValue	LastVariable_vValue	response.data	Returns a numeric representation of the variable *var*.
textValue	LastVariable_tValue	response.xtra.formatted	Returns a string value which represents the variable format and units necessary for displaying the variable value.
retMsg	LastRetMsg	response.error.msg	When an error occurs, this value contains an error message; otherwise, it is empty.

Displaying HTML content in FreeMASTER

– Tabbed HTML Views

- Host HTML pages embedded directly in the FreeMASTER main window
- Context-sensitive description when browsing through project
- Dedicated dashboard tab called “Control page”

– IE vs. Chromium rendering

- Global project option applied to all views: select IE mode or Chromium mode
- Chromium leverages the CEF project – core of the Chrome browser
- Use IE for backward compatible designs, use Chromium for new designs

Todays examples

- Use with out of box MCUXpresso SDK demo application for **FRDM-K64F board**
- SDK Builder at **mcuxpresso.nxp.com**
- Application code:
  - Trivial endless loop incrementing “var8”, “var16” and “var32” variables.
  - Increment amount controlled by the “varXinc” variables.

Pre-requisites

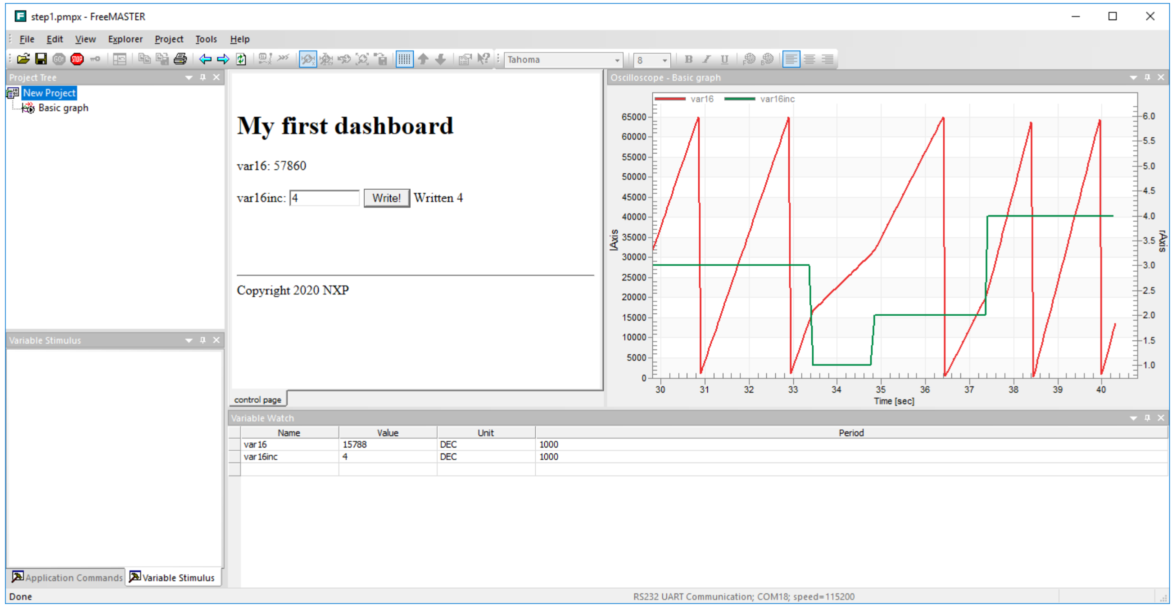
- FreeMASTER 3.0, basic project with var16 and var16inc variables defined
- Text editor, Chrome browser

ActiveX Exercise #1: Basic IE control page

- classic HTML page with <html>, <head> and <body> sections
- FreeMASTER COM+ object reference:

```
<object id="pcm" height="0" width="0" classid=
"clsid:48A185F1-FFDB-11D3-80E3-00C04F176153">
</object>
```

- Standard HTML elements
  - `<span>` to display text
  - `<input type=“text”>`
  - `<input type=“button”>`



My first dashboard

var16: 57860

var16inc: 4 Write! Written 4

Copyright 2020 NXP

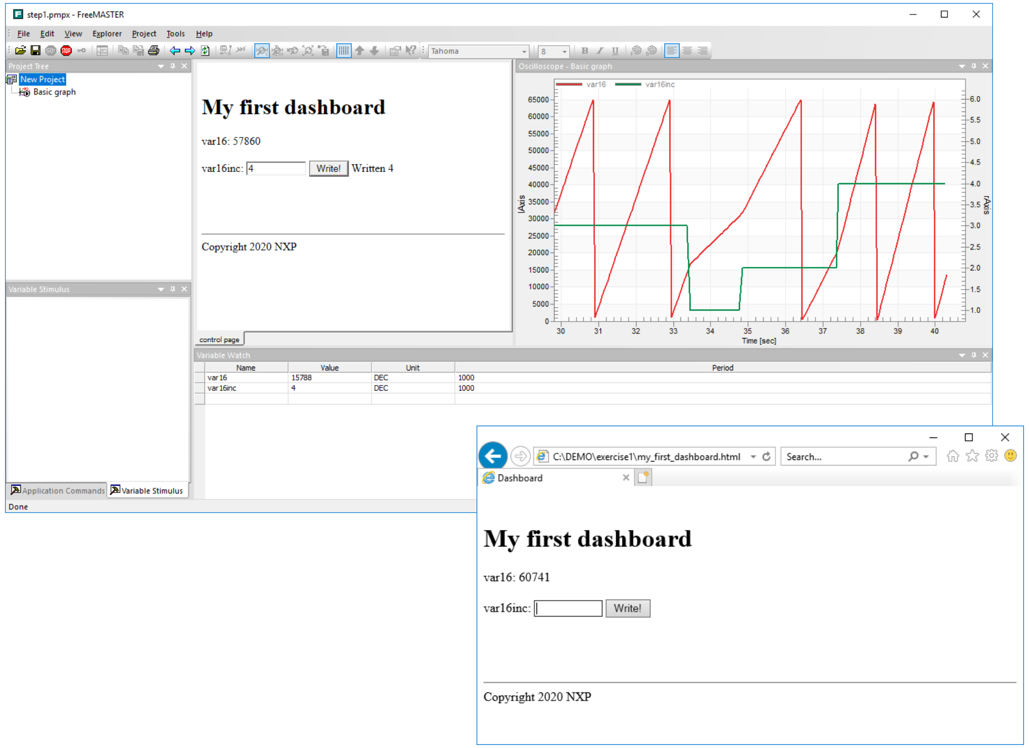
Name	Value	Unit	Period
var16	15788	DEC	1000
var16inc	4	DEC	1000

Done

RS232 UART Communication; COM18; speed=115200

Exercise #1 also running standalone

- Internet Explorer to open dashboard HTML
- FreeMASTER launches automatically by COM+ automation
- Open project manually (or by script)



My first dashboard

var16: 57860

var16inc: 4 Write! Written 4

Copyright 2020 NXP

Name	Value	Unit	Period
var16	15788	DEC	1000
var16inc	4	DEC	1000

My first dashboard

var16: 60741

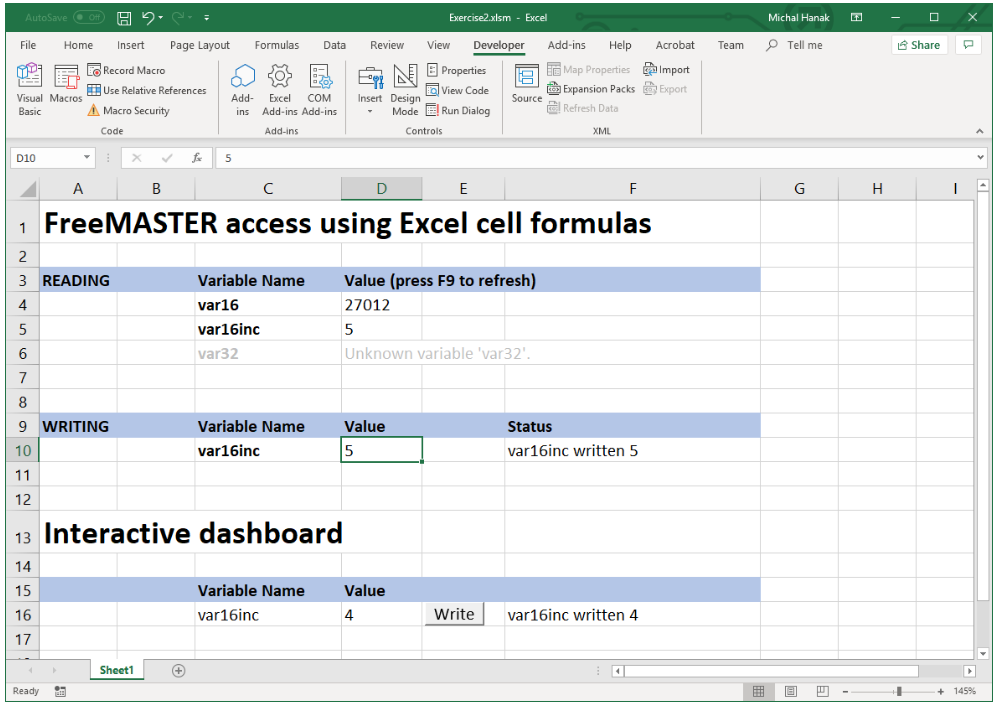
var16inc: Write!

Copyright 2020 NXP

ActiveX Exercise #2: Accessing FreeMASTER from Excel VBA

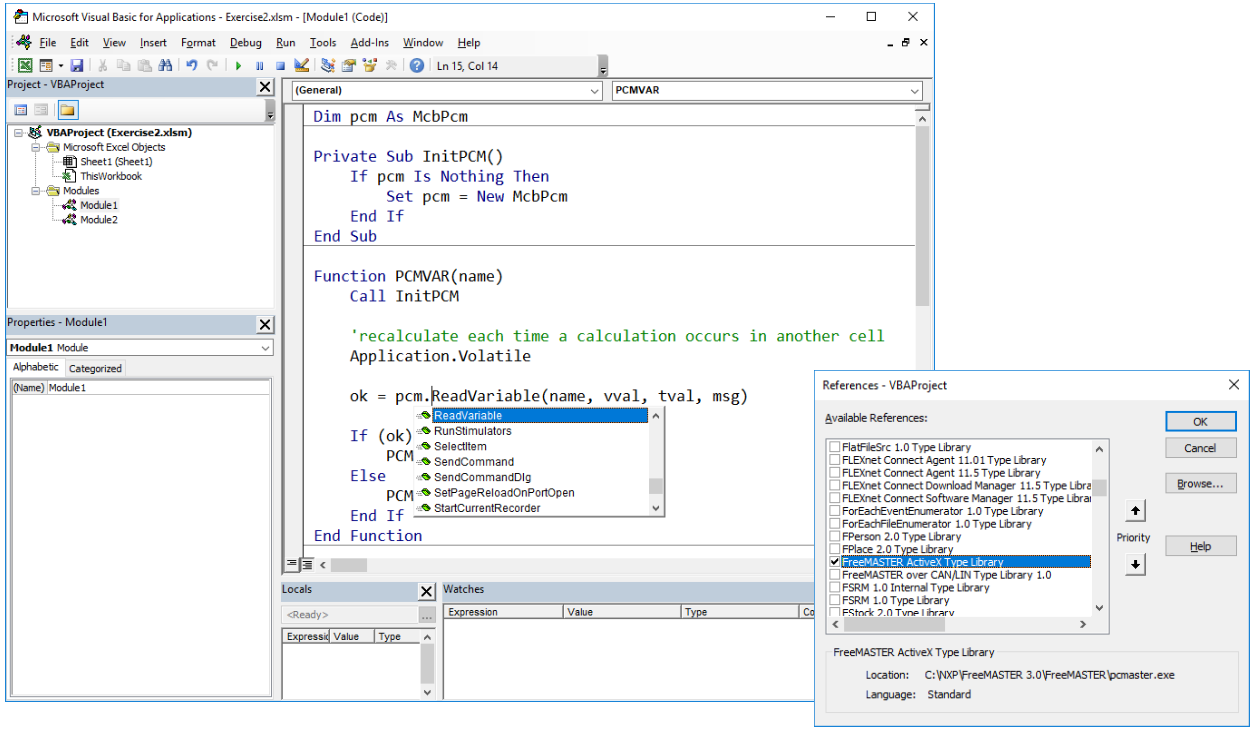
– Excel VBA supports ActiveX natively. Use to create Cell Formulas or custom Forms

– FreeMASTER Type Lib needs to be registered, object is named *McbPcm*



D10 = 5

	A	B	C	D	E	F
1	**FreeMASTER access using Excel cell formulas**					
2						
3	**READING**		**Variable Name**	**Value (press F9 to refresh)**		
4			**var16**	27012		
5			**var16inc**	5		
6			**var32**	Unknown variable 'var32'.		
7						
8						
9	**WRITING**		**Variable Name**	**Value**		**Status**
10			**var16inc**	5		var16inc written 5
11						
12						
13	**Interactive dashboard**					
14						
15			**Variable Name**	**Value**		
16			var16inc	4	Write	var16inc written 4
17						



```
Dim pcm As McbPcm

Private Sub InitPCM()
    If pcm Is Nothing Then
        Set pcm = New McbPcm
    End If
End Sub

Function PCMVAR(name)
    Call InitPCM

    'recalculate each time a calculation occurs in another cell
    Application.Volatile

    ok = pcm.ReadVariable(name, vval, tval, msg)

    If (ok)
        PCM
    Else
        PCM
    End If
End Function
```

- ☐ FlatFileSrc 1.0 Type Library
- ☐ FLEXnet Connect Agent 11.01 Type Library
- ☐ FLEXnet Connect Agent 11.5 Type Library
- ☐ FLEXnet Connect Download Manager 11.5 Type Libra
- ☐ FLEXnet Connect Software Manager 11.5 Type Librai
- ☐ ForEachEventEnumerator 1.0 Type Library
- ☐ ForEachFileEnumerator 1.0 Type Library
- ☐ FPerson 2.0 Type Library
- ☐ FPlace 2.0 Type Library
- ☑ FreeMASTER ActiveX Type Library
- ☐ FreeMASTER over CAN/LIN Type Library 1.0
- ☐ FSRM 1.0 Internal Type Library
- ☐ FSRM 1.0 Type Library
- ☐ FStock 2.0 Type Library

FreeMASTER ActiveX Type Library

Location: C:\NXP\FreeMASTER 3.0\FreeMASTER\pcmaster.exe

Language: Standard

Pros and Cons of using ActiveX and Internet Explorer

Pros

- Backward compatible with older FreeMASTER versions
- Synchronous method execution simplifies JavaScript coding

Cons

- Old technology, being abandoned.
- Global IE settings affect FreeMASTER behavior
- Remote dashboards possible with DCOM, but not trivial to set up
- Synchronous method execution may cause UI to freeze

Pros and Cons of migrating to JSON-RPC

Pros

- Modern approach, widely supported by JavaScript, Python and other languages
- Chromium view, Chrome (and other) browser support
- Asynchronous method execution prevents UI freezing
- Seamless remote dashboard connections

Cons

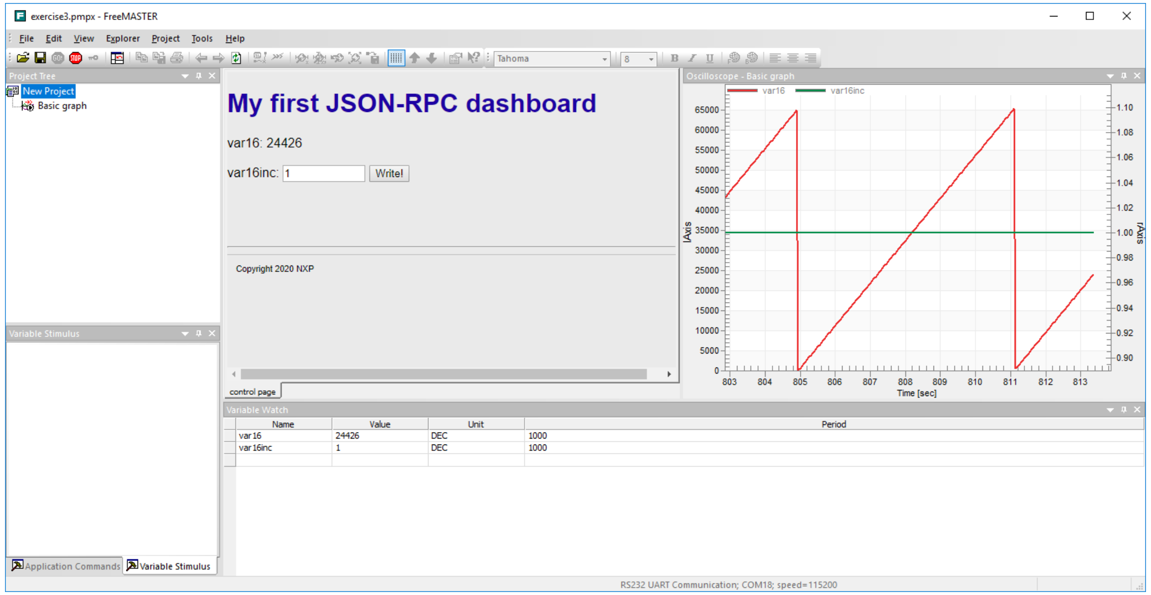
- Need helper code to wrap JSON-RPC layer (freemaster-client.js)
- Asynchronous programming and Promise interface may be more difficult to learn

JSON-RPC Exercise #3: Basic Chromium control page

- With **freemaster-client.js** wrapper object, JSON-RPC becomes as easy as a local JavaScript calls

- JSON-RPC interface almost identical to ActiveX.

- Asynchronous JavaScript programming is based on the **“Promise” interface**.



My first JSON-RPC dashboard

var16: 24426

var16inc: 1 Write!

Copyright 2020 NXP

Name	Value	Unit	Period
var16	24426	DEC	1000
var16inc	1	DEC	1000

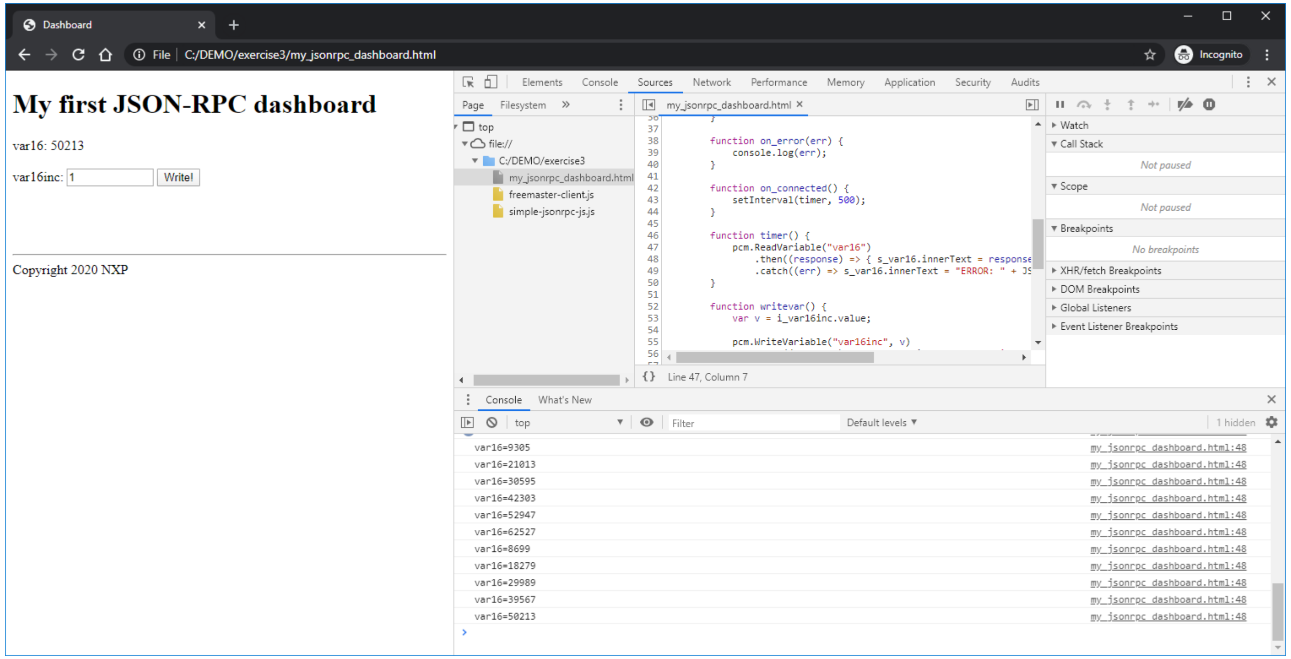
RS232 UART Communication; COM18; speed=115200

Exercise #3 in standalone Chrome with JavaScript debugger

- Same behavior in standalone Chrome browser as in FreeMASTER view

- Powerful Chrome built-in JavaScript debugger

- Easy to set up remote access from Chrome on tablets & phones



My first JSON-RPC dashboard

var16: 50213

var16inc: 1 Write!

Copyright 2020 NXP

```
        function on_error(err) {
            console.log(err);
        }

        function on_connected() {
            setInterval(timer, 500);
        }

        function timer() {
            pcm.ReadVariable("var16")
                .then((response) => { s_var16.innerText = response
                .catch((err) => s_var16.innerText = "ERROR: " + JS
        }

        function writevar() {
            var v = i_var16inc.value;

            pcm.WriteVariable("var16inc", v)
```

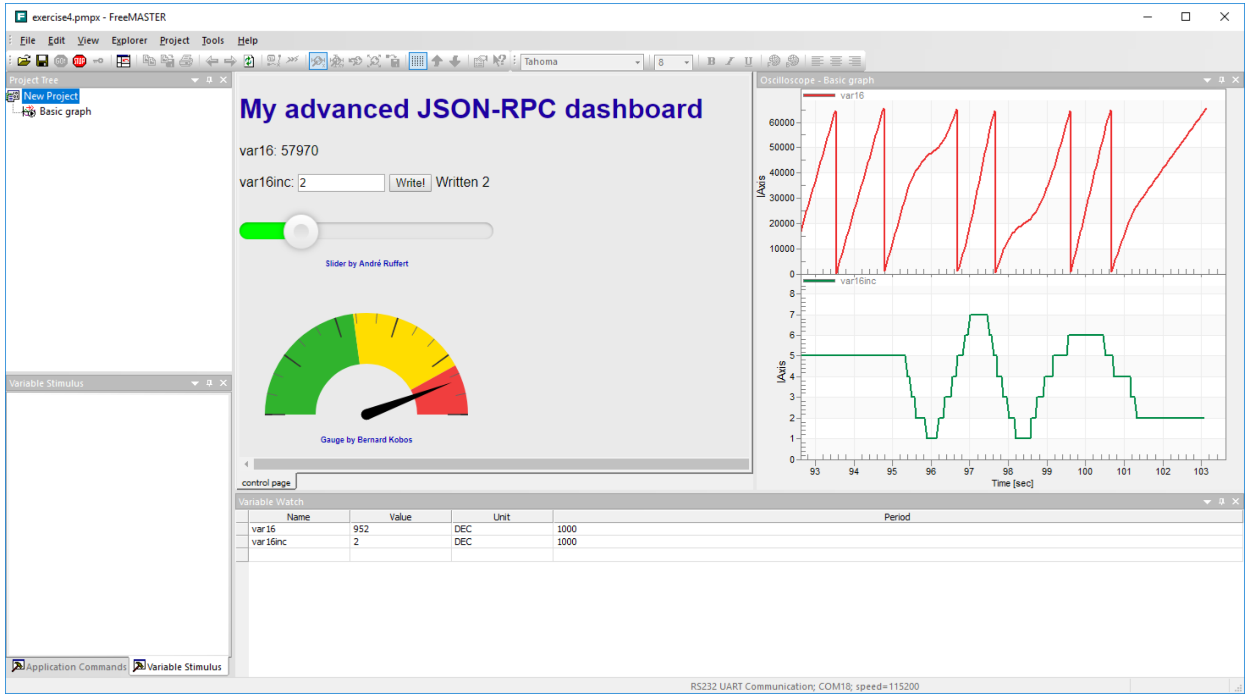
```
var16=9305                                   my_jsonrpc_dashboard.html:48
var16=21013                                  my_jsonrpc_dashboard.html:48
var16=30595                                  my_jsonrpc_dashboard.html:48
var16=42303                                  my_jsonrpc_dashboard.html:48
var16=52947                                  my_jsonrpc_dashboard.html:48
var16=62527                                  my_jsonrpc_dashboard.html:48
var16=8699                                   my_jsonrpc_dashboard.html:48
var16=18279                                  my_jsonrpc_dashboard.html:48
var16=29989                                  my_jsonrpc_dashboard.html:48
var16=39567                                  my_jsonrpc_dashboard.html:48
var16=50213                                  my_jsonrpc_dashboard.html:48
```

JSON-RPC Exercise #4: Adding HTML5 widgets

- With a Chromium support, any modern HTML5 widget may be added to page

- Two free (MIT license) widgets demonstrated today
  - Gauge by Bernard Kobos (link)
  - jQuery Slider by André Ruffert (link)

- Plenty of HTML5 widgets available
  - jqWidgets, Google Charts
  - Chart.js, Plotly.js, ..



My advanced JSON-RPC dashboard

var16: 57970

var16inc: 2 Write! Written 2

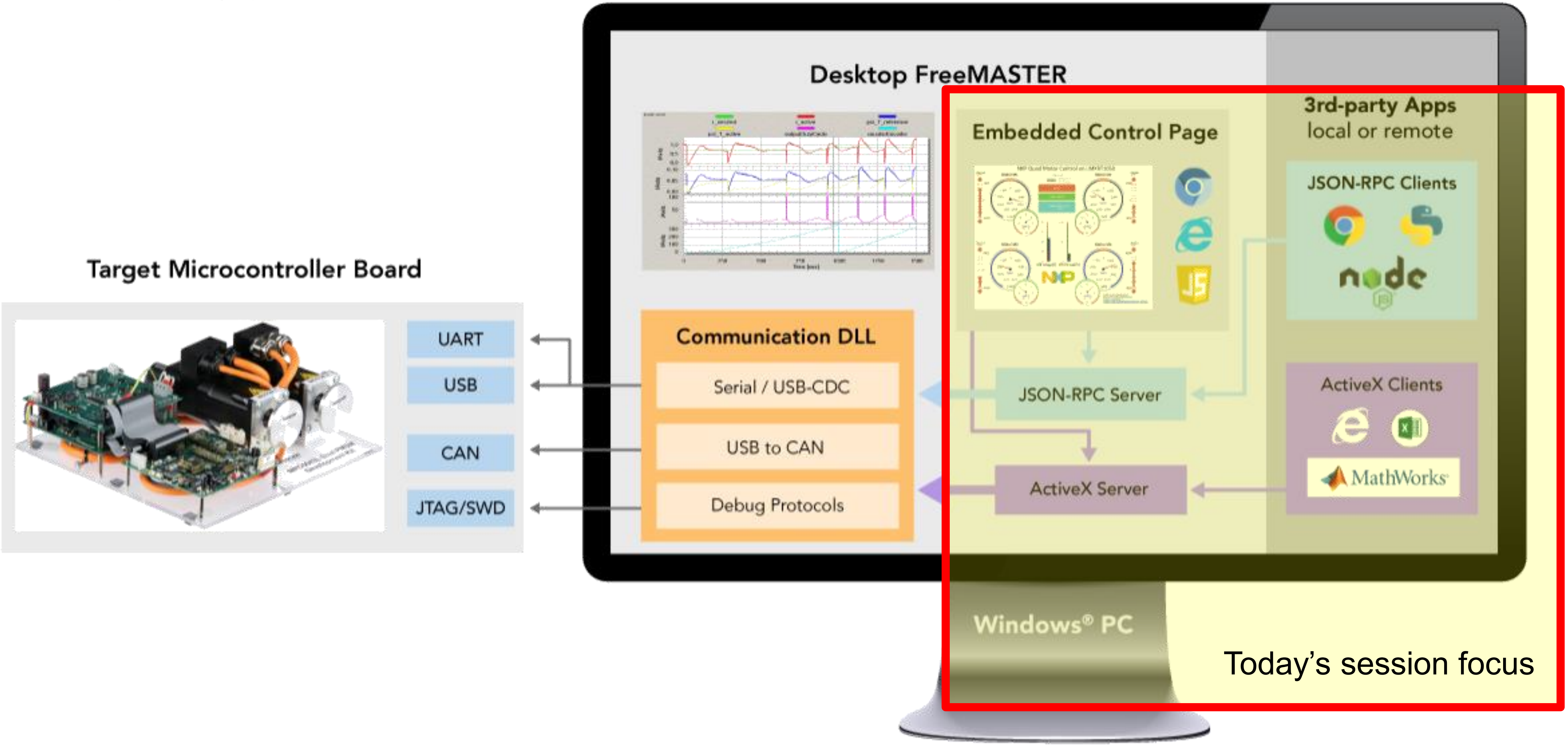
Slider by André Ruffert

Gauge by Bernard Kobos

Name	Value	Unit	Period
var16	952	DEC	1000
var16inc	2	DEC	1000

RS232 UART Communication; COM18; speed=115200

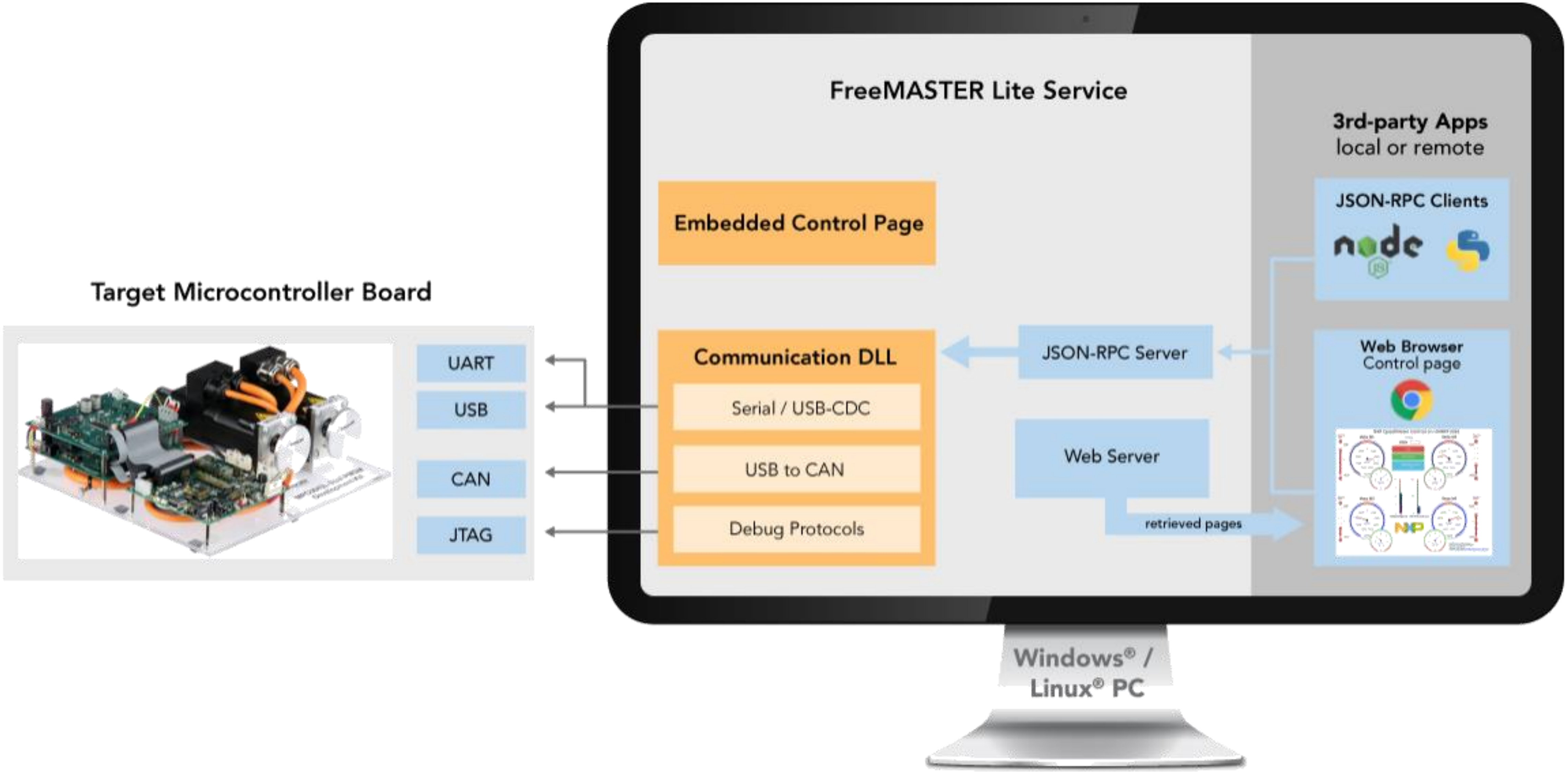
FREEMASTER 3.0



Today’s session focus

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.

NEXT SESSION: FREEMASTER LITE



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

ANY QUESTIONS?

SECURE CONNECTIONS
FOR A SMARTER WORLD