Introduction to FreeMASTER
Dashboard Coding

Using HTML, JavaScript, ActiveX and JSON-RPC

Michal Hanak
Software Engineer

MAY 2020
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

Communication DLL
- Serial / USB-CDC
- USB to CAN
- Debug Protocols

JSON-RPC Server

ActiveX Server

Desktop FreeMASTER

3rd-party Apps
- local or remote

JSON-RPC Clients

ActiveX Clients

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 Desktop application and connectivity to target MCU covered in the last session.
Today’s session focus
3RD PARTY APPLICATION CONNECTIVITY OPTIONS

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
3RD PARTY APPLICATION CONNECTIVITY OPTIONS

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
3RD PARTY APPLICATION CONNECTIVITY OPTIONS

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
EXAM PLES

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:

  ```html
  <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">`
**FREEMASTER ACTIVEX INTERFACE**

**Exercise #1 also running standalone**

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

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
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
FREEMASTER ACTIVEX INTERFACE

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
FREEMASTER JSON-RPC INTERFACE

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.
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
FREEMASTER JSON-RPC INTERFACE

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, ..
Today’s session focus
NEXT SESSION: FREEMASTER LITE

FreeMASTER Lite Service

- Embedded Control Page
- Communication DLL
  - Serial / USB-CDC
  - USB to CAN
  - Debug Protocols
- JSON-RPC Server
- JSON-RPC Clients
- 3rd-party Apps
  - local or remote
- Web Browser
  - Control page
- Web Server
- Retrieved pages

Target Microcontroller Board

- UART
- USB
- CAN
- JTAG

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