OVERVIEW

This application note explains the necessary changes to the existing MLA software stack to migrate from older MRF24WB0MA/MB software stack to incorporate the MRF24WG0MA/MB modules.

GETTING STARTED

- MRF24WG0MA/MB driver code is deeply embedded in the MLA releases.
- Macro definitions are used to differentiate between MRF24WB0MA/MB and MRF24WG0MA/MB.
  - #define MRF24WG (HardwareProfile.h)
- Substitute newly modified WiFi™ directory files that have updates from MRF24WB0MA/MB projects.
- Six WiFi files have specific post-fix _24WG versions.
- Certain C function prototypes are changed or removed, depending on which earlier version is being compared to, for example, SaveAppConfig(), power throttle control functions.
- A good starting point is to start with the TCP/IP demonstration application as a reference for porting.

The best approach to use in migrating applications based on MRF24WB0MA/MB to MRF24WG0MA/MB is adding application files to the new software stack.

- Start with the new version of the same demonstration application that was used to spawn the original application, or use the TCP/IP demonstration application.
- Add custom application files from the older application.
- Header files that typically need modifications are TCPIPConfig.h, WF_Config.h and HardwareProfile.h. The TCPIPConfig.h and WF_Config.h files can be used to set up WiFi connections and TCP/IP configurations. The HardwareProfile.h file can be used to configure the stack for custom circuit board.

Examples of more customized application files are: CustomHTTPApp.c, CustomSNMPApp.c, MainDemo.c, WF_Config.c, and so on.

- For changed Application Programming Interfaces (APIs), refer to the following release notes and help files:
  - \Microchip\TCPIP Stack\TCPIP Stack Version.txt
  - \Microchip\Help\Readme for Microchip Application Libraries.htm
  - \Microchip\Help\TCPIP Stack Help.chm

LIST OF CHANGES

The following section describes the changes to the existing MLA software stacks.

TCP/IP Stack v5.41 Based Application

Related WiFi files are located at \Microchip Solutions v20xx-xx-xx\Microchip\TCPIP Stack\WiFi.

Where, Microchip Solutions v20xx-xx-xx indicates the MLA installation directory location.

For example, \Microchip Solutions v2012-07-18\Microchip\TCPIP Stack\WiFi

Replace existing header files that has WF prefix with the following files:

- WFConsoleIwconfig.h
- WFConsoleMsgs.h
- WFMgmtMsg.h
- WFMac.h
- WFConsoleMsgHandler.h
- WFDriverPrv.h
- WFConsole.h
- WFApi.h
- WFConsoleIfconfig.h
- WFEasyConfig.h
- WFConsoleIwpriv.h
- WFRaw.h
Table 1 lists the files that have MRF24WG0MA/MB specific versions.

**TABLE 1: FILES WITH MRF24WG0MA/MB SPECIFIC VERSIONS**

<table>
<thead>
<tr>
<th>Existing Files</th>
<th>Replace With</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFDriverCom.c</td>
<td>WFDriverCom_24G.c</td>
</tr>
<tr>
<td>WFDriverRaw.c</td>
<td>WFDriverRaw_24G.c</td>
</tr>
<tr>
<td>WFMsgMgmt.c</td>
<td>WFMsgMgmt_24G.c</td>
</tr>
<tr>
<td>WFParamMsg.c</td>
<td>WFParamMsg_24G.c</td>
</tr>
<tr>
<td>FDriverPrv.h</td>
<td>WFDriverPrv_24G.h</td>
</tr>
<tr>
<td>WFRaw.h</td>
<td>WFRaw_24G.h</td>
</tr>
</tbody>
</table>

**Additional Migration from TCP/IP Stack v5.25**

- **SaveAppConfig** has different input parameters:
  
  \[ V5.25: \text{void \text{SaveAppConfig}(void)} \]
  
  \[ V5.36: \text{void \text{SaveAppConfig}(const \text{APP\_CONFIG} *\text{prtAppConfig})} \]

- **Search for function call** `SaveAppConfig()` in the following files:
  
  MainDemo.c, MainDemo.h, CustomHTTPApp.c, UARTConfig.c, WFEasyConfig.c, SNMPv3.c

- **Add #define MRF24WG definition to project definition files.**

- **Power throttle table control support was removed.**
  
  If the following API’s were used, they must be removed from the application to use the current version of the TCP/IP stack:
  
  - static INT8 WFPowerToAppPower(UINT8 wfPower)
  - static UINT8 AppPowerToWFPower(INT8 appPower)
  - void WF_ThrottleTableSet(tWFThrottleTable *p_table)
  - void WF_ThrottleTableGet(tWFThrottleTable *p_table)
  - void WF_ThrottleTableEnable()
  - void WF_ThrottleTableDisable(UINT8 bitRate)
  - void WF_ThrottleTableGetState(BOOL *p_state, UINT8 *p_bitRate)

**Other Changes**

- **WF\_Config.c, WF\_Config.h**
  
  `void WF\_ProcessEvent(UINT8 event, UINT16 eventInfo)`
  
  is changed to
  
  `void WF\_ProcessEvent(UINT8 event, UINT16 eventInfo, UINT8 *extraInfo).`

  *Where, UINT8 *extraInfo is the WPA passphrase that is sent to the host. The host can speed up the connection by making the passphrase to key calculation.*

- **WF\_Config.h**
  
  Depending on the version, the contents in the WF\_Config.h file changes. The format change occurs between MLA July 18, 2012 and MLA April 3, 2012 versions.

- **Many changes are detailed in the WFApi.h file.**
  
  Refer to comments in the code for detailed function descriptions.
  
  - **Domains reduced to** WF\_DOMAIN\_FCC, WF\_DOMAIN\_ETSI, WF\_DOMAIN\_JAPAN, WF\_DOMAIN\_OTHER
  
  - New API calls specific to MRF24WG are listed in Table 2.

---

**Existing Files**

- Replace With
  
  - WFDriverCom.c
  - WFDriverCom_24G.c
  - WFDriverRaw.c
  - WFDriverRaw_24G.c
  - WFMsgMgmt.c
  - WFMsgMgmt_24G.c
  - WFParamMsg.c
  - WFParamMsg_24G.c
  - FDriverPrv.h
  - WFDriverPrv_24G.h
  - WFRaw.h
  - WFRaw_24G.h
### TABLE 2: NEW API CALLS SPECIFIC TO MRF24WG

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>New API Calls Specific to MRF24WG</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>void WF_CMGetConnectContext(tWFConnectContext *p_ctx);</td>
<td>WFConnectionManager.c</td>
</tr>
<tr>
<td>2</td>
<td>void WF_CPSetSsidType(UINT8 CpId, UINT8 hidden);</td>
<td>WFConnectionProfile.c</td>
</tr>
<tr>
<td>3</td>
<td>void WF_CFGetSsidType(UINT8 CpId, UINT8 *hidden);</td>
<td>WFConnectionProfile.c</td>
</tr>
<tr>
<td>4</td>
<td>void WF_CPSetWepKeyType(UINT8 CpId, UINT8 wepKeyType);</td>
<td>WFConnectionProfile.c</td>
</tr>
<tr>
<td>5</td>
<td>void WF_CFGetWepKeyType(UINT8 CpId, UINT8 *p_wepKeyType);</td>
<td>WFConnectionProfile.c</td>
</tr>
<tr>
<td>6</td>
<td>void WF_CPGetWPSCredentials(UINT8 CpId, tWFWpsCred *p_cred);</td>
<td>WFConnectionProfile.c</td>
</tr>
<tr>
<td>7</td>
<td>void WF_CASetDtimInterval(UINT16 dtimInterval);</td>
<td>WFConnectionAlgorithm.c</td>
</tr>
<tr>
<td>8</td>
<td>void WF_CAGetDtimInterval(UINT16 *p_dtimInterval);</td>
<td>WFConnectionAlgorithm.c</td>
</tr>
<tr>
<td>9</td>
<td>void WF_GetTxMode(UINT8 mode);</td>
<td>WFFparamMag_24G.c</td>
</tr>
<tr>
<td>10</td>
<td>void WF_ENABLEDEBUGPRINT(UINT8 option);</td>
<td>WFFparamMag_24G.c</td>
</tr>
<tr>
<td>11</td>
<td>void WF_SETSTACKVERSION(UINT8 major, UINT8 minor);</td>
<td>WFFparamMag_24G.c</td>
</tr>
<tr>
<td>12</td>
<td>void WF_TxPowerGetMax(INT8 *p_maxTxPower);</td>
<td>WFTxPower.c</td>
</tr>
<tr>
<td>13</td>
<td>void WF_MulticastSetConfig(tWFMultiCastConfig *p_config);</td>
<td>WFFparamMag_24G.c</td>
</tr>
<tr>
<td>14</td>
<td>void WF_EnableSWMultiCastFilter(void);</td>
<td>WFFparamMag_24G.c</td>
</tr>
<tr>
<td>15</td>
<td>void WF_DisplayModuleAssertInfo(void);</td>
<td>WFDebugStrings.c</td>
</tr>
</tbody>
</table>

### Software Aid

Use a file compare utility like WinMerge (freeware) to reduce the time to determine the differences between an existing application and the latest version of the TCP/IP stack. This also reduces the likelihood subtle changes will go unnoticed.

The WinMerge software can be downloaded from the following web site [http://winmerge.org](http://winmerge.org).
Note the following details of the code protection feature on Microchip devices:

• Microchip products meet the specification contained in their particular Microchip Data Sheet.

• Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.

• There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.

• Microchip is willing to work with the customer who is concerned about the integrity of their code.

• Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademark

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, rPIC, SST, SITest Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Silicon Storage Technology in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworkshops, dsSPEAK, ECAN, ECONECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, MINDI, MIWI, MPASM, MF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved. Printed on recycled paper.

ISBN: 978-1-62076-717-7

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.