2019 MASTERs Conference

23075 IoT6

Simplifying TCP/IP Applications with MPLAB® Harmony

Hands-On

Lab Manual

Instructors: Martin Ruppert Raji Shanmugasundaram Niklas Larsson

Microchip Technology Inc.



| Table of Contents | |
|--|----|
| Introduction | 3 |
| Hardware Requirements | 4 |
| Software Requirements | 5 |
| How to connect the USB and the CAT5 Network cable to the board | 6 |
| Lab 1 | 7 |
| Overview | 7 |
| Lab Procedure | 8 |
| Lab 2 | 27 |
| Overview | 27 |
| Lab Procedure | 31 |
| Lab 3 | 48 |
| Overview | 48 |
| Lab Procedure | 51 |
| MPLAB [®] Harmony TCP/IP Stack | 54 |
| TCP Module API Function List | 54 |
| UDP Module API Function List | 55 |
| Harmony TCP/IP API Subset For all Lab's | 56 |
| TCP Socket Management Functions | 56 |
| UDP Socket Management Functions | 60 |
| | 64 |

Introduction

This Lab Manual provides the step by step procedure to complete two labs in the MASTERs 23075 IoT6 Class.

If you don't know how TCP/IP works but you have to add network connectivity to your product, this is the right class for you. We will teach you the basics of TCP/IP, how the client-server model works, what ports and sockets are and how applications use them to create TCP/IP connections. The class will also teach you the fundamentals of network analysis with the well-known tool Wireshark. The hands-on part of the class utilizes Microchip's 32-bit MCUs with MPLAB[®] Harmony. You can learn the architecture and the fundamentals of the Harmony TCP/IP stack to interface your TCP/IP application with some common stack APIs. To make your life easier, the FreeRTOS[™] task schedular is used to simplify your application programming.

In Lab 1 we will open a TCP project, do some stack re-configuration and a connectivity check and in Lab 2 we will show an Application integration for local access, using the example of a Vending machine. Finally in Lab 3 we will make an Application integration for external access, using the example of a Weather Service.

Hardware Requirements

The following hardware is required:

- SAM E70 Xpained Ultra (Microchip Part Number: DM320113)
 - o <u>https://www.microchip.com/DevelopmentTools/ProductDetails/PartNO/DM320113</u>



- OLED1 Xplained Pro extension kit (Microchip Part Number: ATOLED1-XPRO)
 - o https://www.microchip.com/Developmenttools/ProductDetails/ATOLED1-XPRO



- Cat 5 Ethernet Patch Cable
- USB Male A to USB Male B Micro Cable



Software Requirements

The following software is required:

- Microchip MPLAB X IDE v5.20
 - o http://www.microchip.com/mplab
- Microchip MPLAB XC32 Compiler v2.15
 - o http://www.microchip.com/mplab/compilers
- Microchip MPLAB Harmony 3
 - o http://www.microchip.com/mplab/mplab-harmony
- Microchip MPLAB Harmony Configuration (MHC) Tool Plugin v3.3.0.1
- Tera Term v4.95

The Lab1, Lab2 and Lab3 class expects a Harmony 3 with the following subset of Versions (Tags)

To make it more easy in the class for the attendees, the folder C:/MASTERs/23075 already contains this H3 Checkout

| ect and | configure the packages the | at will be included in the current project | м а |
|--------------------------|---|--|-------------------------------------|
| oad N | Vame | Version | Dependencies |
| Vb | SD | v3.3.0 | csp(3.2.1) |
| V 0 | ore | v3.3.0 | csp(3.2.1) |
| V CI | rypto | v3.2.1 | core(v3.2.1) |
| V C | sp | v3.2.1 | |
| V n | et | v3.3.0 | core(3.2.1), csp(3.2.1), dev_packs(|
| | | | |
| onfigure | Device Family and CMSIS P | ack Paths: | |
| onfigure FP: | Device Family and CMSIS F | ack Paths: \SAME70_DFP\4.0.26\same70b\atdf\AT | TSAME70Q218.atdf |
| onfigure FP: MSIS: | Device Family and CMSIS F .\dev_packs\Microchip .\dev_packs\arm\CMS | ack Paths: \SAME70_DFP\4.0.26\same70b\atdf\AT IS\5.4.0 | rSAME70Q21B.atdf |

In the (optional) case a standard Harmony 3 installation is used, in the MHC configuration the **csp** v3.2.1 and **core** v3.2.1 must be selected. This can be done by clicking on the sub module and select in the drop down menu, the above mentioned version. The following checkout could take several minutes. The checkout is only local, no external Github access will be used.

| bad | Name | Version | Dependencies |
|----------------------|--|---|--|
| | audio | v3.3.0 | core(3.3.0), csb(3.3.0), usb(3.2.2), . |
| | bootloader | v3.0.0 | csp(3.0) |
| 1 | bsp | v3.3.0 | csp(3.2.1) |
| | bt | v3.3.0 | core(3.3.0), csb(3.3.0), usb(3.2.2), |
| 1 | core | v3.2.1 | csp(3.2.1) |
| 1 | crypto | v3.2.1 | core(v3.2.1) |
| 1 | csp | v3.2.1 | |
| | gfx | v3.3.0 | core(3.3.0), bsp(3.3.0) |
| | gfx_apps | v3.3.0 | usb(3.2.1), gfx(3.3.0) |
| | micrium_ucos3 | v3.0.0 | core(3.0) |
| | motor_control | v3.2.0 | csp(v3.2.0) |
| 1 | net | v3.3.0 | core(3.2.1), csp(3.2.1), dev_packs(. |
| | touch | v3.2.0 | csp(v3.3.0) |
| | | N3.2.2 | core(3.2.1) |
| nfigu P: 1SIS: | re Device Family and CMSIS Pa .\dev_packs\Microchip\! .\dev_packs\arm\CMSI | ck Paths: SAME70_DFP\4.0.26\same70b\atdf\AT s\5.4.0 | SAME70Q218.atdf |

How to connect the USB and the CAT5 Network cable to the board

- Ensure the Erase jumper is open
- Open the J805 jumper



- Insert the LAN8740 PHY daughter board on the ETHERNET PHY MODULE header.
- Connect the micro USB cable from the computer to the DEBUG USB connector on the SAM E70 Xplained Ultra Evaluation Kit
- Establish a connection between the router/switch with the SAM E70 Xplained Ultra Evaluation Kit through the RJ45 connector



Lab 1

Overview

Lab 1 will show you how to open an existing TCP/IP MPLAB Harmony 3 Project and using the MPLAB Harmony Configuration (MHC) Tool. The project will incorporate basic TCP/IP functionality to allow the SAM E70 Xplained Ultra to connect to an Ethernet Network, along with a simple application to flash a "Heartbeat" LED every 500ms. Once the project is generated and programmed onto the development kit, you will use a number of techniques to validate that the PIC is connected to a network and determine its IP Address. The concepts that will be covered in this lab include:

- Open a SAM E70 MPLAB X Project
- Configuring the MPLAB Harmony path
- Configuring the TCP/IP Stack options, including:
 - Network Configuration of the Host Name
 - TCP/IP Services including Dynamic Host Configuration Protocol Client, ICMPv4 Server (for Ping testing) & Announce Discovery Tool
 - Bandwidth testing with "iperf"
- Configuring the Harmony Console and Command Service for monitoring and control of the TCP/IP stack via a Terminal Client running on a USB CDC Interface (Emulated RS232 COM Port).
 - Toggling the IO Pin that drives USER_LED0 on the SAM E70 Xplained UltraSAM E70 Xplained Ultra
- Using the Windows Command Line Ping Tool and the Microchip TCP/IP Discovery tools to test connectivity of your SAM E70 Xplained UltraSAM E70 Xplained Ultra on the network
- Use the Console and Command System to get help on available TCPIP Commands and execute a command to get information about the network configuration.

Lab Procedure

Starting MPLAB X IDE

1.1. Start MPLAB X IDE by double clicking on the MPLAB X IDE v5.20 icon found on the Windows desktop.





Project Load, modify with MHC, Generate, build and run

1. Open Project by choosing File ➤ Open Project... from the main menu and select lab1 project

| 🗴 Open Proje | ect | | × |
|--------------|----------------|--|---------------------------------------|
| Ca | Look in: 🚺 M | IASTERs 💌 | Ē 💣 |
| Recent Items | ⊡] 23075 | | Project Name: |
| Recent Items | 🕀 🗎 🕂 | | web_net_server_nvm_mpfs_freertos_lab1 |
| | er 📗 lab1 | rmware | Open Required Projects: |
| Desktop | <u></u> | sam_e70_xult_freertos.X | |
| My Docum | | Manuals ts | |
| | . 1 vm_ | server | |
| Computer | | | |
| | File name: | C:\MASTERs\23075\lab1\firmware\sam_e70_xult_ | freertos.X Open Project |
| Network | Files of type: | Project Folder | ▼ Cancel |

- 2. Open Project Properties by choosing
 - a. File ➤ Project Properties from the main menu
 - b. Or select with a right click the project node in the project windows and select at the bottom

| _ | History | Þ | |
|---|------------|---|--|
| | Properties | | |

c. Or select the toolbox in the Dashboard



- 3. Select the XC32 Compiler v2.15
- 4. Select the SAME70 Xplained by clicking on the SN: Number

| Project Properties - web_net_server_n | vm_mpfs_freertos_lab1 | |
|--|--|---|
| Categories: General File Inclusion/Exclusion Conf: [sam_e70_xult_freertos] | Configuration Family: All Families | Device: ATSAME70Q21B 		▼ |
| ····· · Loading ····· · Libraries ····· · Building | Supported Debug Header: None | Supported Plugin Board: None |
| XC32 (Global Options) xc32-as xc32-gcc xc32-g++ xc32-ld xc32-ar | Packs: Packs SAME70_DFP SAME70_DFP SAME70_DFP Solution Solution Solution Solution Solution PM3 Real ICE Simulator Simul | Compiler Toolchain: Compiler Toolchains ARM ARM ARM (v6.3.1) [c:\Program File XC32 [Download Latest] XC32 [Download Latest] XC32 (v2.15) [C:\Program Fil XC32 (v2.05) [C:\Program Fil XC32 (v1.44) [C:\Program Fil XC32 (v1.40) [C:\Program Fil XC32 (v1.40) [C:\Program Fil XC32 (v1.40) [C:\Program Fil |
| | *Tip: double click on serial number (SN) to use a frien | dly name (FN) instead. |
| Manage Configurations | ОК С | ancel Apply Unlock Help |

5. Select from the Tools Menu the Harmony 3 Configurator



6. Ensure the H3 Path is set to "C:\MASTERs\23075\h3\"

| MPLAB Harmony | Launcher |
|--------------------------|--|
| Active Project Informati | on: |
| Name: sam_e70_xult_ | freertos |
| Select an option to use | for the launch paths for the MPLAB Harmony Framework tool suite: |
| MPLAB Harn | nony Project Path - The framework path that is stored with this MPLAB Harmony project. Reconfigure Paths |
| Harmony Path: | C:\MASTERs\23075\h3 |
| MPLABX Pat | ${f h}$ - The framework path that is stored with MPLABX (configured through Tools->Options->Harmony) |
| Harmony Path: | C:\microchip\h3 |
| Default Launch Action: | Prompt For Path Selection (This 🔻 |
| (Configurable through | Tools->Options->MPLAB Harmony) |
| | Launch |

7. Select "Launch"

| search kesul | | |
|--------------|---------------------------------------|---|
| , | Launching MPLAB® Harmony Configurator | 0 |
| Path: | C:\MASTERs\23075\h3\ | |
| Generating | Configuration Database | |
| | Path: Generating | Path: C:\MASTERs\23075\h3\ Generating Configuration Database |

The first time the MHC is started, it can take up to 2 Minutes before the Configuration Database is prepared

8. In the next window the H3 parts and their used Version Numbers are displayed

| | | Manajara | Deservice |
|---------------|---------------------------|--|------------------------------------|
| | vame | version | Dependencies |
| ✓ b | sp | v3.3.0 | csp(3.2.1) |
| V C | ore | V3.2.1 | csp(3.2.1) |
| V C | rypto | V3.2.1 | core(v3.2.1) |
| | sp | V3.2.1 | coro(2,2,1), cop(2,2,1), dov, pack |
| - | | 101010 | |
| | | | |
| 6 | | | |
| figure | Device Family and CMSIS F | Pack Paths: | |
| nfigure P: | Device Family and CMSIS F | Pack Paths: D\SAME70_DFP\4.0.26\same70b\atdf\AT | SAME70Q21B.atdf |

Click on "Launch"

9. Open the saved state file



10. The MHC is up and running

| MPLAB Harmony Configurator - sam_e/0_xult_treertos | • | | | |
|---|--|--|--------------------------------|---------------------------------|
| ile Generate Tools Utilities Window | | | | |
| 🗼 🛅 🎐 🕐 stiffere | | | | Framework: C:\MASTERs\23075\h3\ |
| 🗋 Available Components 🛛 🗕 🛃 🗖 | Project Graph | | 🕳 🛃 🗖 Configuration Options | - 20 |
| + | | View: Root | it 👻 | |
| P Board Support Packages (BSPs) Harmony Ubraries Packs Packs Parlpherals Third Party Libraries Tools | System C TIME System Service Ocre Service SYS_TIME O TMR SAM E70 Xplained Ultra BSP | MSIS Pack Presentation Cryptographic (Crypto) Library SYS_TIME LIB_CRYPTO TreatROS Third Party Library RTOS 0 TWIHSO Peripheral Library IDC 0 | on Layer ® | |
| | Core Harmony Cars Savice © RTOS Core Service © MEMORY Driver © Core Service | CONSOLE System Sarrice • Core Service UART SYS_CONSOLE TCP/IP STACK SYS_CONSOLE | | |
| | EFC Instances (=- (+)- Peripheral Library MEMORY / MEMOR | FILE SYSTEM System Service O Core Service DRV_MEDIA SYS_FS | | |
| Available Components Addive Components Console -red | | | C Help C Configuration Options | |
| vercome to the mir cho normony Collinguiatori | | | | |

11. Select Active Components (left below)

| | Available Components | Active Components | |
|-----|----------------------|-------------------|--|
| l n | Console - 🖼 🗗 | | |

12. Select the Instance 0 in Active Components



- 13. And change in the Configuration Options (on the right side) the Host Name to something meaningful for you.
 - The Host name can be identified in the Network.

| Configuration Options | | - 🗗 🗖 |
|--------------------------------|-------------------|-------|
| - + | | |
| | | |
| | 0 | |
| Interface | GMAC | |
| Host Name | MARTIN_RUPPERT | |
| Mac Address | 00:04:25:1C:A0:02 | |
| IPv4 Static Address | 192.168.100.11 | |
| IPv4 SubNet Mask | 255.255.255.0 | |
| IPv4 Default Gateway Address | 192.168.100.1 | |
| | 192.168.100.1 | |
| IPv4 Secondary DNS | 0.0.0 | |
| Power Mode | full | |
| Network Configuration Start-up | Flags | |
| Network MAC Driver | DRV_GMAC_Object | |
| | | - |

14. Select "TCP/IP Application Layer Configuration" in Active Components

| Active Components | - 🗗 🗖 |
|---------------------------------|---------|
| 🗙 🗔 🛤 🖻 🖏 🛧 | |
| CMSIS Pack | |
| CONSOLE | |
| Core | |
| Cryptographic (Crypto) Library | |
| EFC | |
| - FILE SYSTEM | |
| FreeRTOS | |
| | |
| Presentation Layer | |
| SAM E70 Xplained Ultra BSP | |
| System | |
| TC0 | |
| TCP/IP STACK | |
| | |
| ANNOUNCE | |
| - DHCP CLIENT | |
| DNS CLIENT | |
| HTTPNET SERVER | |
| IPERF | |
| - NBNS | |
| - SMTP CLIENT | |
| SNTP | |
| TCP/IP Application Layer Config | gurator |
| BASIC CONFIGURATION | |
| | |
| | |
| ITRANSPORT LAYER | |
| TIME | I |

15. Ensure that ANNOUNCE and IPERF are selected



16. Select ICMPv4 in the Active Components

| Active Components | - 20 | 3 |
|--------------------------------|------|---|
| 🗙 🗔 🛤 🖭 🖷 🛧 | | |
| CMSIS Pack | | |
| CONSOLE | | |
| Core | | |
| Cryptographic (Crypto) Library | | |
| EFC | | |
| FILE SYSTEM | | |
| FreeRTOS | | |
| | | |
| Presentation Layer | | |
| SAM E70 Xplained Ultra BSP | | |
| - TCO | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| ARP | | |
| ICMPv4 | | |
| ICMPv6 | | |
| - IPv4 | | |
| IPv6 | | |
| NDP | | |
| TCP/IP Network Layer Configura | tor | |
| | | |
| TIME | | |
| TWIHSO | | |
| USART1 | | |

17. And ensure that the "Use ICMPv4 Client" is selected



18. Select "Code" (Generate Code)

| ÷ | MPL | AB Harm | ony Configur | ator - sam_e/(|
|-----|--------|-----------|----------------|----------------|
| ile | Ger | nerate To | ols Iltilities | Window |
| 1 | B | ₿ € | Code | |
| | Active | e Compone | nts | |

19. Select Don't Save

| P Modified Configuration | × |
|---|---------------------|
| Current configuration has been modified. Do you want to save it befor | re file generation? |
| | |
| | Don't Save Save As |

20. Select Generate

| 👙 Generate Project | | | × |
|----------------------------|--------------------|----------|----------|
| 1. Configure Generation Se | ettings | | |
| Generate Settings | | | |
| Merge Strategy | USER RECEN | IT | ▼? |
| 5 57 | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| (Mouse over a property | for detailed help) | | |
| | | | |
| | | | |
| | | | |
| | | | |
| 2. View Warnings | | | |
| Ture | Description | | |
| туре | Description | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| 3. Click Generate | | | |
| 5. Check Generate | | (| |
| | | Generate | Cancel |
| | | [] | |

21. Some Files will be changed and the MHC is asking in a "diff" window, if the changes should be taken over.

Accept all changes in the file by clicking on the Arrow in the middle above.

| Methy Merge Action: 2 Image: The mode Image: The mode Image: The mode Image: The mode Interaction of the mode | Merging: configuration.h. 8107676870428926504.old | | | | | | | |
|--|--|--|----------------|-----|--|--|----------------------|-------------|
| Corested Code Code Corested Code | Pending Merge Actions: 2 | | 🕂 🕂 Next | | Previous | | Clos | se |
| Concerned Gole 1/2 Current File: Configuration, BURD/SURD/SURD/SURD/SURD/SURD/SURD/SURD/S | | | _ | | Note: The c | ontent of the right pane displays the curr | ent state of this me | erge. |
| seties Size: 12:0:0: 12:0: 12:0: 12:0:0: 12:0: 12:0: 12:0: 12: | Gener | ated Code | | 1/2 | Current File: configuration.h_81076 | 76870428926504.old | | |
| <pre>Medica COPY_PAY NOT BETACHABLE TIME 30 Medica COPY_PAY NOT BETACHABLE TIME 3 Medica COPY_PAY NOT AND 1 NERVER AND 1 Medica COPY_PAY NOT AND 1 MEDICA MEDICA</pre> | #define TCPIP IPV6 NDP MAX NEIGHBOR ADVERTISEMENT | 3 | 132 | 332 | #define TCPIP IPV6 NDP MAX NEIGHBOR ADVERTISEMENT | 3 | | A [] |
| <pre>Madriam TCPIP TYPE REP BATTANE TIRER 1 1 334 034 define TCPIP TYPE REP BATTANE TIRER 1 4 446tim TCPIP TYPE REP CLAY, TIRE YRANGE, THERE 1 446tim TCPIP TYPE REP CLAY, TIRE YRANGE, THERE 1 446tim TCPIP TYPE REP CLAY, TIRE YRANGE, THERE YALL 0 335 035 046tim TCPIP TYPE REP CLAY, TIRE YRANGE, THERE YALL 0 336 036 046tim TCPIP TYPE REP CLAY, TIRE YRANGE, THERE YALL 0 337 037 046tim TCPIP TYPE REP CLAY, TIRE YRANGE, THERE YALL 0 446tim TCPIP TYPE REP CLAY, TIRE YRANGE, THERE YALL 0 446tim TCPIP TYPE REP CLAY, THERE YALL 0 44</pre> | #define TCPIP IPV6 NDP REACHABLE TIME | 30 | 333 | 333 | #define TCPIP IPV6 NDP REACHABLE TIME | 30 | | |
| <pre>Median CCPIP [PYCe NUP_VILST. PROOR_TIME 5 define TCPIP [PYCe NUP_VILST. PROOR_TIME 7 define TCPIP NETWOOR DEFAULT NOT NAME INCO *BARCT* define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:02* 4 define TCPIP NETWOOR DEFAULT NOT NAME INCO *DIO:04:25:1C:A0:00* *DIO:04:25:1C:A0:00* *DIO:04:25:1C:A0:</pre> | #define TCPIP IPV6 NDP RETRANS TIMER | 1 | 334 | 334 | #define TCPIP IPV6 NDP RETRANS TIMER | 1 | | |
| <pre>define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE_TWO MOORS (60 * 60 * 2) define TCPIP FIVE NUE VALUE LITETINE, TUE NUE VALUE LITETINE,</pre> | #define TCPIP_IPV6_NDP_DELAY_FIRST_PROBE_TIME | 5 | 335 | 335 | <pre>#define TCPIP_IPV6_NDP_DELAY_FIRST_PROBE_TIME</pre> | 5 | | |
| <pre>define TCPLP FPC MTU_INCREASE_TIMEOUT 600 define TCPLP IPVG MDT_INCREASE_TIMEOUT 600 define TCPLP IPVG MDT_TARK_TIMER_BATE 32 338 define TCPLP IPVG MDT_TARK_TIMER_BATE 32 339 339 4* Network Configuration Index 0 */ define TCPLP IPVG MDT_TARK_TIMER_BATE 32 341 341 /* Network Configuration Index 0 */ define TCPLP IPVG MDT_TARK_TIMER_BATE 32 446 fine TCPLP IPVG MDT_TARK_DARALDE NAME IDX0 **MARTIN RUPERT* ** 342 446 fine TCPLP IF GAAC. 344 344 446 fine TCPLP INTWORK DEFAULT_INST NAME IDX0 **MARTIN RUPERT* ** 345 345 446 fine TCPLP INTWORK DEFAULT_INST NAME IDX0 **MARTIN GAAC DAGA IDX0 **GAAC 344 344 446 fine TCPLP INTWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 446 fine TCPLP NETWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 446 fine TCPLP NETWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 446 fine TCPLP NETWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 446 fine TCPLP NETWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 446 fine TCPLP NETWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 146 fine TCPLP NETWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 146 fine TCPLP NETWORK DEFAULT_IP ADDRESS IDX0 **192.166.100.11* 346 346 146 fine TCPLP NETWORK DEFAULT_INTERACE FLASS IDX0 **192.166.100.1* 355 355 354 466 fine TCPLP NETWORK DEFAULT_INTERACE FLASS IDX0 **192.166.100.1* 356 356 466 fine TCPLP NETWORK DEFAULT_INTERACE FLASS IDX0 **192.166.100.1* 357 357 357 357 358 466 fine TCPLP NETWORK DEFAULT_INTERACE FLASS IDX0 **1011* 356 356 356 356 466 fine TCPLP NETWORK DEFAULT_INTERACE FLASS IDX0 **1011* 356 356 356 356 466 fine TCPLP NETWORK DEFAULT_INTERACE FLASS IDX0 **1011* 357 357 357 357 357 357 357 357 358 358 466 fine TCPLP NETWORK DEFAULT_INTERACE FLASS IDX0 **1011* 356 356 356 356 466 fine TCPLP NETWORK DEFAUL</pre> | #define TCPIP_IPV6_NDP_VALID_LIFETIME_TWO_HOURS | (60 * 60 * 2) | 336 | 336 | #define TCPIP_IPV6_NDP_VALID_LIFETIME_TWO_HOURS | (60 * 60 * 2) | | |
| #define TCPLP IFY% HEP_TASK_TIMER_BATE 32 338 #define TCPLP IFY% HEP_TASK_TIMER_BATE 52 /* Network Configuration Index 0 */ 400 340 340 340 340 340 340 340 340 340 340 340 340 340 340 340 340 340 340 341 341 /* Network Configuration Index 0 */ */ 342 4define TCPLP INTWORK DEFAULT_INTERFACE_NAME_IDX0 *GARC* 343 343 4define TCPLP INTWORK DEFAULT HOST NAME IDX0 *GARC* 344 345 344 </td <td>#define TCPIP IPV6 MTU_INCREASE_TIMEOUT</td> <td>600</td> <td>337</td> <td>337</td> <td>#define TCPIP IPV6 MTU_INCREASE_TIMEOUT</td> <td>600</td> <td></td> <td></td> | #define TCPIP IPV6 MTU_INCREASE_TIMEOUT | 600 | 337 | 337 | #define TCPIP IPV6 MTU_INCREASE_TIMEOUT | 600 | | |
| ** Network Configuration Index 0 */ #define ICPTP NETWORK_DEFAULT INTERFACE_NAME_LDX0 "GRAC" #define ICPTP NETWORK_DEFAULT_INTERFACE_NAME_LDX0 "GRAC" #define ICPTP NETWORK_DEFAULT_IP_ADDRESS_IDX0 "192.166.100.11" #define ICPTP NETWORK_DEFAULT_IP_ADDRESS_IDX0 "192.166.100.11" #define ICPTP NETWORK_DEFAULT_IP_ADDRESS_IDX0 "192.166.100.11" #define ICPTP NETWORK_DEFAULT_IP_ADDRESS_IDX0 "192.166.100.11" #define ICPTP NETWORK_DEFAULT_IP_NETWORK_DEFAULT_IP_ADDRESS_IDX0 "192.166.100.11" #define ICPTP NETWORK_DEFAULT_IP_NETWORK_DEFAULT_IP_ADDRESS_IDX0 "192.166.100.11" #define ICPTP NETWORK_DEFAULT_IP_NETWORK_DEFAULT_IP_NETWORK_DEFAULT_IP_NETWORK_DEFAULT_ING NETWORK DEFAULT_ING NETWORK DEFAULT_INTERFACE_FLASS_IDX0 "10.10" #define ICPTP NETWORK_DEFAULT_INTERFACE_FLASS_IDX0 "10.10" ICPTP NETWORK_CONFIG DESC_LIENT ON () ICPTP NETWORK_CONFIG | #define TCPIP_IPV6_NDP_TASK_TIMER_RATE | 32 | 338 | 338 | #define TCPIP_IPV6_NDP_TASK_TIMER_RATE | 32 | | |
| * Network Configuration Index 0 */ define TCPIP NETWORK DEFAULT INTERFACE NAME IDXO "GMAC" define TCPIP NETWORK DEFAULT INTERFACE NAME IDXO "GMAC" define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NOT NAME IDXO "GMAC" define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NOT NAME define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NOT NAME define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NOT NAME define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NOT NAME define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NOT NAME define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NAME IDXO DEFINITON NOT NAME define TCPIP NETWORK DEFAULT NOT NAME IDXO DEFINITON NOT NAME define TCPIP STACK COMMAND ENALIE | | | 339 | 339 | | | | |
| <pre>/* Metwork Configuration Index 0 */ define ICPTP_NETWORK_DEFAULT_INTERFACE_NAME_LDX0 "BMC" define ICPTP_NETWORK_DEFAULT_INTERFACE_FAULS_INTON "BMC" define ICPTP_NETWORK_DEFAULT_INTERFACE_FAULS_INTON "CONTOR DEFAULT_INTERFACE_FAULS_INTON "CONTOR DEFA</pre> | | | 340 | 340 | | | | |
| define TCPLP NETWORK DEFAULT_INTERFACE_NAME_LEXO "GMAC" define TCPLP NETWORK DEFAULT_NOT NAME_IEXO "MARTIN RUPPER" *** ************************************ | /* Network Configuration Index 0 */ | | 341 | 341 | /* Network Configuration Index 0 */ | | | |
| #define TCPIP_IF_GMAC 343 343 define TCPIP_IF_GMAC #define TCPIP_IF_GMAC 344 344 344 #define TCPIP_INTWORK DEFAULT NOT NAME INCO "WCHEPOARD G" ************************************ | <pre>#define TCPIP_NETWORK_DEFAULT_INTERFACE_NAME_IDX0</pre> | "GMAC" | 342 | 342 | <pre>#define TCPIP_NETWORK_DEFAULT_INTERFACE_NAME_IDX0</pre> | "GMAC" | | |
| 344 344 4define TCPIP NETWORK DEFAULT HOST NAME IDXO "MARENE RUPPERT" ************************************ | #define TCPIP_IF_GMAC | | 343 | 343 | #define TCPIP_IF_GMAC | | | |
| #define TCPIP NETWORK DEFAULT NOT NAME IDX0 "WEAREN RUPPERT" ************************************ | | | 344 | 344 | | | | _ |
| <pre>define TCPIP NETWORK_DEFAULT INC ADOR IDS0 *00:04:25:1C:A0:02* define TCPIP NETWORK_DEFAULT INC ADOR IDS0 *00:04:25:1C:A0:02* define TCPIP NETWORK_DEFAULT IP ADDRESS IDS0 *192:168:100.11* define TCPIP NETWORK_DEFAULT IP ADDRESS IDS0 *192:168:100.1* define TCPIP NETWORK_DEFAULT SSIDU0 *192:168:100.1* define TCPIP NETWORK_DEFAULT NETWORK_DEFAULT SSIDU0 *192:168:100.1* define TCPIP NETWORK_DEFAULT INTERFACE FLASS IDS0 *0.0.0.0* *101* *100* *101*</pre> | #define TCPIP NETWORK DEFAULT HOST NAME IDX0 | "MARTIN RUPPERT" | - ≫ 345 | 345 | #define TCPIP NETWORK DEFAULT HOST NAME IDX0 | "MCHPBOARD C" | | |
| <pre>idefine TCPIP NETWORK_DEFAULT IF ADDRESS IDX0 ''''''''''''''''''''''''''''''''''''</pre> | #define TCPIP_NETWORK_DEFAULT_MAC_ADDR_IDX0 | "00:04:25:1C:A0:02" | 346 | 346 | #define TCPIP_NETWORK_DEFAULT_MAC_ADDR_IDX0 | "00:04:25:1C:A0:02" | | |
| <pre>section CEPLE NETWORK_DEFAULT_IE_MARK LING *255.255.255.0**************************</pre> | Adefine TOPIP NETWORK DEFAULT IP ADDRESS IDXO | "192 168 100 11" | 348 | 348 | Adefine TOPIE NETWORK DEFAULT IN ADDRESS IDXO | "192 168 100 11" | | |
| <pre>indefine TCPIP_NETWORK_DEFAULT_ONTENA_TING *192.148.100.1* indefine TCPIP_NETWORK_DEFAULT_DNETRON *192.148.100.1* indefine TCPIP_NETWORK_DEFAULT_DNETRON *192.148.100.1* indefine TCPIP_NETWORK_DEFAULT_NETROR *192.148.100.1* *192.1</pre> | #define TCPIP NETWORK DEFAULT IP MASK IDX0 | "255.255.255.0" | 349 | 349 | #define TCPIP NETWORK DEFAULT IP MASK IDXO | "255.255.255.0" | | |
| <pre>section TCELE NETWORK_DEFAULT_DNS_IDV0 "192.148.100.1" store TCELE NETWORK_DEFAULT_DNS_IDV0 "192.148.100.1" store TCELE NETWORK_DEFAULT_DSCOME DNS_IDV0 "192.148.100.1" store TCELE NETWORK_DEFAULT_SCOME DNS_IDV0 "192.148.100.1" store TCELE NETWORK_DEFAULT_SCOME DNS_IDV0 "192.148.100.1" store TCELE NETWORK_DEFAULT_FORE NODE_IDV0 "192.148.100.1" store TCELE NETWORK_DEFAULT_INTERFACE FLAGS_IDV0 '10 '10 '10 '10 '10 '10 '10 '10 '10 '1</pre> | Adefine TOPIP NETWORK DEFAULT GATEWAY IDXO | "192.168.100.1" | 350 | 350 | Adefine TOPIE NETWORK DEFAULT GATEWAY IDXO | "192.168.100.1" | | |
| <pre>idefine TCPIP NETWORK_DEFAULT_SECOND_DNS_IDX0</pre> *0.0.0.0" *352 #354 #define TCPIP_NETWORK_DEFAULT_SECOND_DNS_IDX0 *0.0.0.0" *55 #354 #define TCPIP_NETWORK_DEFAULT_SECOND_DNS_IDX0 *0.0.0.0" *55 #354 #define TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 *0.0.0.0" *56 #364 #define TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 *0.0.0.0" *57 #357 *58 #define TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 *0.0.0.0" *56 #define TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 *0.0.0.0" *0.0.0.0" *56 #define TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 *0.0.0.0" *0.0.0.0" *56 #define TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 *0.0.0" *1000 *0.0.0.0" *0.0.0.0 | #define TOPIP NETWORK DEFAULT DNS IDX0 | "192.168.100.1" | 351 | 351 | #define TCPIP NETWORK DEFAULT DNS IDX0 | "192.168.100.1" | | |
| <pre>define TCFIP NETWORK_DEFAULT_FORE_NOOD_IDX0 **ult* define TCFIP NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 **ult* define TCFIP NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 **ult* interfault_INTERFACE_FLAGS_IDX0 **ult* int</pre> | #define TCPIP NETWORK DEFAULT SECOND DNS IDX0 | "0.0.0.0" | 352 | 352 | #define TCPIP NETWORK DEFAULT SECOND DNS IDX0 | "0.0.0.0" | | |
| #define TCFIP NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 354 355 356 | #define TCPIP NETWORK DEFAULT POWER MODE IDX0 | "full" | 353 | 353 | #define TCPIP NETWORK DEFAULT POWER MODE IDX0 | "full" | | = * |
| TCPLP NETWORK CONFIG DECY CLEBRIC ON I/V TCPLP NETWORK CONFIG DECY CLEBRIC ON I/V TCPLP NETWORK CONFIG DECHESTION I/V TCPLP NETWORK CONFIDENCE I/V TCPLP NETWORK CONFIDENCE I/V TCPLP NETWORK CONFIDENCE I/V TCPLP NETWORK CONFID DECHESTION I/V TCPLP NETWORK CONF | #define TCPIP NETWORK DEFAULT INTERFACE FLAGS IDX0 | Λ | 354 | 354 | #define TCPIP NETWORK DEFAULT INTERFACE FLAGS IDX0 | X | | |
| TCPEP NETWORK CONFIG DNS CLIENT ON 1/\ 356 356 356 TCPEP NETWORK CONFIG DNS CLIENT ON 1/\ TCPEP NETWORK | | TCPIP NETWORK CONFIG DHCP CLIENT ON \ | 355 | 355 | | TCPIP NETWORK CONFIG DHCP CL | IENT ON \ | |
| TCPIP NETWORK CONFIG IP STATIC 357 357 357 TCPIP NETWORK CONFIG IP STATIC #define TCPIP NETWORK DEFAULT MAC DRIVER IDX0 DRV GMAC Object 359 4define TCPIP NETWORK DEFAULT MAC DRIVER IDX0 DRV GMAC Object *** Lopip cmd Configuration ***/ 360 360 360 #define TCPIP STACK COMMAND EXABLE 363 364 *** #define TCPIP STACK COMMAND INABLE 4 365 364 #define TCPIP STACK COMMAND INABLE 4 365 364 #define TCPIP STACK COMMAND INABLE 4 365 365 #define TCPIP STACK COMMAND INABLE 4 365 364 #define TCPIP STACK COMMAND INABLE 4 365 365 #define TCPIP STACK COMMAND INABLE 4 365 365 #define TCPIP STACK COMMAND INTERCONFERDERT 4 365 366 #define TCPIP STACK COMMAND INTERCONFERDERT 1000 4 #define TCPIP STACK COMMAND INTERCONFERDERT 1000 1000 #define TCPIP STACK COMMAND INTERCONFERDERT 5000 5000 #define TCPIP STACK COMMAND INTERCONFERDERT 5000 5000 #define TCPIP STACK COMMAND INTERCONFERDERT 5000 5000 | | TCPIP NETWORK CONFIG DNS CLIENT ON \ | 356 | 356 | | TCPIP NETWORK CONFIG DNS CLI | ENT ON I | - |
| 358 358 #define TCPIP NETWORK DEFAULT NAC DRIVER IDXO DRV GNAC Object 360 360 361 361 362 362 363 363 364 363 365 364 366 361 367 363 368 364 369 364 360 361 361 361 362 362 363 /*** Lopip cmd Configuration ***/ #define ICPIP STACK_COMPAND ENABLE 363 #define ICPIP STACK_COMPAND ENABLE 364 #define ICPIP STACK_COMPAND ENABLE 4 #define ICPIP STACK_COMPAND ENABLE 4 #define ICPIP STACK_COMPAND ENABLE 4 #define ICPIP STACK_COMPAND ENABLE 1000 #define ICPIP STACK_COMPANDE INFIL ENABLE 1010 | | TCPIP NETWORK CONFIG IP STATIC | 357 | 357 | | TCPIP NETWORK CONFIG IP STAT | IC | |
| #define TCPIP NETWORK DEFAULT MAC DRIVER IDX0 DRV GMAC Object 359 #define TCPIP NETWORK DEFAULT MAC DRIVER IDX0 DRV GMAC Object /*** LOPID End Configuration ***/ 360 360 360 /*** LOPID End Configuration ***/ 363 361 361 /*** LOPID End Configuration ***/ 363 361 361 /*** LOPID End Configuration ***/ 363 363 /*** LOPID END ENDELE #define TCPIP STACK COMMAND ENABLE 363 364 364 #define TCPIP STACK COMMAND LOWE END EQUESTS 4 365 365 #define TCPIP STACK COMMAND LOWE END EQUESTS LOWE 1000 1000 1000 1000 #define TCPIP STACK COMMAND LINE END END END ELGO TIMEOUT 5000 367 367 366 | | | 358 | 358 | | | | |
| 360 360 361 361 362 362 4define ICFLP_STACK_COMBAND_INABLE 363 4define ICFLP_STACK_COMBAND_INABLE 364 4define ICFLP_STACK_COMBAND_INABLE 364 4define ICFLP_STACK_COMBAND_INABLE 364 4define ICFLP_STACK_COMBAND_INABLE 364 4define ICFLP_STACK_COMBAND_INABLE 4 4define ICFLP_STACK_COMBAND_INABLE 4 4define ICFLP_STACK_COMBAND_INABLE 4 4define ICFLP_STACK_COMBAND_INABLE 4 4define ICFLP_STACK_COMBAND_INAP_INF_INF_INF_INF_INF_INF 4 4define ICFLP_STACK_COMBAND_INF_INF_INF_INF_INF_INF_INF_INF 4 4define ICFLP_STACK_COMBAND_INF_INF_INF_INF_INF_INF 4 4define ICFLP_STACK_COMBAND_INF_INF_INF_INF_INF_INF_INF 4 4define ICFLP_STACK_COMBANDS_INF_INF_INF_INF_INF_INF_INF 4 4define ICFLP_STACK_COMBANDS_INF_INF_INF_INF_INF_INF_INF 5000 4define ICFLP_STACK_COMBANDS_INF_INF_INF_INF_INF_INF 5012 | #define TCPIP NETWORK DEFAULT MAC DRIVER IDX0 | DRV GMAC Object | 359 | 359 | #define TCPIP NETWORK DEFAULT MAC DRIVER IDX0 | DRV GMAC Object | | |
| 361 361 /*** LOPID End Configuration ***/ 362 362 /*** ICPID STACK_COMMAND_ENABLE 363 361 /*** ICPID_ENABLE 363 /*** ICPID STACK_COMMAND_ENABLE 363 364 /*** ICPID_ENABLE 364 /*** ICPID_STACK_COMMAND_ENABLE 364 364 /*** ICPID_ENABLE 4 /*** ICPID_STACK_COMMAND_INABLE 365 365 /*** ICPID_ENABLE 4 /*** ICPID_STACK_COMMAND_INABLE 366 366 /*** ICPID_ENABLE 4 /*** ICPID_STACK_COMMAND_INABLE 100 366 /*** ICPID_ENABLE 100 /*** ICPID_STACK_COMMAND_INF_INABLE 100 367 /*** ICPID_ENABLE 100 /*** ICPID_STACK_COMMAND_INF_INF_INF_INF_INF_INF_INF 100 367 /*** 367 /*** | | | 360 | 360 | | | | |
| 362 362 *** Lopip md Configuration ***/ 363 363 *** Lopip md Configuration ***/ \$define TCERF_STACK_COMMAND EMALE 364 364 refere TCER_STACK_COMMAND EMALE 364 \$define TCERF_STACK_COMMAND EMALE 364 364 refere TCERF_STACK_COMMAND EMALE 4 \$define TCERF_STACK_COMMAND EMALE 365 365 define TCERF_STACK_COMMAND IONE DEMALES 4 \$define TCERF_STACK_COMMANDS IONE ECON REQUESTS 4 365 366 define TCERF_STACK_COMMANDS IONE ECON REQUESTS 4 \$define TCERF_STACK_COMMANDS IONE ECON REQUEST DELAY 1000 366 366 define TCERF_STACK_COMMANDS IONE ECON REQUEST DELAY 1000 \$define TCERF_STACK_COMMANDS INFIL FUNALE false 368 368 define TCERF_STACK_COMMANDS IONE ECON REQUEST 5000 | | | 361 | 361 | | | | |
| /*** <u>topip and</u> Configuration ***/ 363 363 /*** <u>topip and</u> Configuration ***/ #define <u>TCPIP</u> STACK_COMMUND LNABLE 364 364 364 364 364 #define <u>TCPIP</u> STACK_COMMUND LNABLE 364 364 364 364 364 364 #define <u>TCPIP</u> STACK_COMMUND LNABLE 364 364 364 364 364 364 #define <u>TCPIP</u> STACK_COMMUND LNABLE 365 365 464 365 365 365 366 | | | 362 | 362 | | | | |
| idefine [CFLF_STACK_COMMAND_EXABLE 364 364 idefine [CFLF_STACK_COMMAND_EXABLE idefine [CFLF_STACK_COMMAND_ICMP_ECHO_REQUEST] 4 365 366 idefine [CFLF_STACK_COMMAND_ICMP_ECHO_REQUEST] 4 idefine [CFLF_STACK_COMMANDS_ICMP_ECHO_REQUEST] 4 365 366 idefine [CFLF_STACK_COMMANDS_ICMP_ECHO_REQUEST] 4 idefine [CFLF_STACK_COMMANDS_ICMP_ECHO_REQUEST] 1000 366 366 idefine [CFLF_STACK_COMMANDS_ICMP_ECHO_REQUEST] 1000 idefine [CFLF_STACK_COMMANDS_ICMP_ECHO_REQUEST] 000 367 367 idefine [CFLF_STACK_COMMANDS_ICMP_ECHO_TINEOUT] 5000 idefine [CFLF_STACK_COMMANDS_ICMF_ECHO_TINEOUT] 5000 368 368 idefine [CFLF_STACK_COMMANDS_ICMF_ECHO_TINEOUT] 5000 | /*** topip cmd Configuration ***/ | | 363 | 363 | /*** topip cmd Configuration ***/ | | | |
| definit CPFLP STACK_COMMANDS COMPANDS | #define TCPIP_STACK_COMMAND_ENABLE | | 364 | 364 | #define TCPIP_STACK_COMMAND_ENABLE | | | |
| #define ICPLF STACK_COMMANDS ICMLF ECHO PECHO #define ICPLF STACK_COMMANDS ICMLF ICMLF 1000 #define ICPLF STACK_COMMANDS ICMLF ICMLF 5000 #define ICPLF STACK_COMMANDS ICMLF ICMLF 5000 #define ICPLF STACK_COMMANDS ICMLF ICMLF ICMLF ICMLF | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUESTS | 4 | 365 | 365 | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUESTS | 4 | | |
| #define TCPIP_STACK_COMMANDS_ICMP_ECND_TIMEOUT 5000 367 346 fine TCPIP_STACK_COMMANDS_ICMP_ECND_TIMEOUT 5000 #define TCPIP_STACK_COMMANDS_INTE_INABLE false 368 368 define TCPIP_STACK_COMMANDS_INTE_INABLE false | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_DEL# | AY 1000 | 366 | 366 | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_DELA | Y 1000 | | |
| #define TCPIP_STACK_COMMANDS_WIFI_ENABLE false 368 #define TCPIP_STACK_COMMANDS_WIFI_ENABLE false | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_TIMEOUT | 5000 | 367 | 367 | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_TIMEOUT | 5000 | | |
| | #define TCPIP_STACK_COMMANDS_WIFI_ENABLE | false | 368 | 368 | #define TCPIP_STACK_COMMANDS_WIFI_ENABLE | false | | |
| #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_BUFF_SIZE 2000 369 #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_BUFF_SIZE 2000 | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_BUFF | 5_SIZE 2000 | 369 | 369 | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_BUFF | _SIZE 2000 | | |
| #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_DATA_SIZE 100 370 #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_DATA_SIZE 100 | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_DATA | A_SIZE 100 | 370 | 370 | #define TCPIP_STACK_COMMANDS_ICMP_ECHO_REQUEST_DATA | SIZE 100 | | |
| 371 371 | | | 371 | 371 | | | | |
| 372 372 | | | 372 | 372 | | | | Ŧ |

22. Then click on close in the upper right corner



23. Same for the next 2 diff windows

| ····· | | | | | |
|--|-----------|-----|-----|---|---------------------------------------|
| Merging: tcpip_private.h_5816637879687640463.old | | | | | × |
| Pending Merge Actions: 1 | Vext Next | | | Previous | Close |
| | | | | Note: The content of the right name displays the cur | rent state of this merce |
| Committed Carte | | 1/1 | | Current Files taria animta la E016603020602640460 ald | |
| Generated Code | 53 | 1/1 | 0.0 | Current File: tcpip_private.n_3616637879687640463.0id | |
| // private stack manager interface | 94 | ~~ | 94 | // private stack manager interface | ^ _ |
| #include "tcpip/src/tcpip_manager_control.h" | 95 | | 05 | // pirvate state manager interiate | |
| | 96 | | 55 | *include - copip/sic/copip_manager_control.in- | |
| #include "tcpip/src/tcpip_announce_manager.h" | 97 | | 07 | Finaluda Branin/ana/ranin announce managen bB | |
| #include "tcpip/src/ndp_manager.h" | 98 | | 0.0 | finglude "topip/sto/copip_announce_manager.m | |
| #include "tcpip/src/ipv4_manager.h" | 99 | | 00 | Findlude Branin/and/and/ manager bl | |
| #include "tcpip/src/ipv6_manager.h" | 100 | | 100 | finglude "topip/sic/ipvf_manager.h" | |
| #include "tcpip/src/icmpv6_manager.h" | 101 | | 101 | Findlude Brann/arot/imput manager bl | |
| #include "tcpip/src/dhcpv6_manager.h" | 102 | | 102 | findlude "topip/sic/idmpv6_manager.h" | |
| <pre>#include "tcpip/src/icmp_manager.h"</pre> | 103 | | 103 | finclude "topip/sic/amp manager h" | |
| <pre>#include "tcpip/src/dhcp_manager.h"</pre> | 104 | | 104 | finglude #toppin/sto/doop_manager h# | |
| #include "tcpip/src/arp_manager.h" | 105 | | 105 | finclude "topip/sic/and manager h" | |
| <pre>#include "tcpip/src/dns_manager.h"</pre> | 106 | | 106 | finclude #topin/sto/dag manager h | |
| #include "tcpip/src/tcp_manager.h" | 107 | | 107 | finclude "topip/sto/dns_manager.h" | |
| #include "topip/src/nbns_manager.h" | 108 | | 108 | findlude "topip/sic/top_manager.h" | |
| #include "tcpip/src/http_net_manager.h" | 109 | | 109 | finclude "topip/sto/http net manager h" | |
| <pre>#include "tcpip/src/tcpip_commands_manager.h"</pre> | 110 | | 110 | finglude "topip/sic/rospin commands manager h" | |
| #include "tcpip/src/udp_manager.h" | 111 | | 111 | Finaluda #tanin/ara/uda managar b | |
| #include "topip/src/sntp_manager.h" | 112 | | 112 | findlude "topip/sic/sup_manager.h" | |
| finclude "topip/src/lldp manager.h" | 113 | | 113 | finclude "topip/sto/shop_manager h" | |
| <pre>#include "tcpip/src/iperf manager.h"</pre> | -> 114 | > | 114 | finclude "topip/sic/itdp manager.h" | |
| <pre>#include "tcpip/src/smtpc_manager.h"</pre> | 115 | | 115 | finclude "topip/sto/smopo_manager.n | |
| <pre>#include "tcpip/src/tcpip_packet.h"</pre> | 116 | | 116 | finclude "tropin/stc/tropin_plottern | |
| <pre>#include "topip/src/topip_helpers_private.h"</pre> | 117 | | 117 | finclude "topip/sto/copig_neipers_privatern | |
| <pre>#include "tcpip/src/oahash.h"</pre> | 118 | | 118 | Findlada "Formin/arc/hash fny h" | |
| #include "tcpip/src/hash_fnv.h" | 119 | | 119 | finclude "topip/sto/masi_intin | |
| #include "tcpip/src/tcpip_notify.h" | 120 | | 120 | rustate ooptp/sto/ooptp_nost1/m | |
| | 121 | | 121 | fendif // TCDIP STACK PRIVATE H | |
| <pre>#endif // _TCPIP_STACK_PRIVATE_H</pre> | 122 | | 122 | | |
| | 123 | | | | · · · · · · · · · · · · · · · · · · · |
| | | | | | |
| | | | | | |

24. The whole process is displayed with a progress bar

| Generating Project | | | × |
|--|-----------|-------------|-----|
| Task Type | Remaining | Total | |
| File Markup | 0 | | 482 |
| File Copy | 52 | | 482 |
| Libraries | 0 | | 0 |
| Settings | 8 | | 8 |
| Source Paths | 0 | | 0 |
| | | | |
| Generating file: C:\MASTERs\23075\h3\crypto\src\des3 | .c | | |
| | 94% | Please Wait | |

25. Back again in the main window of MPLABX, click on the "Make and Program Device" button



26. After successful build, the SAME70 is programmed automatically



27. The programming take about 30 seconds with on Board Debugger Check whether the USER_LED0 Activity is Blinking



Figure 1. SAM E70 Xplained Ultra Evaluation Kit with PHY Daughter Board

4-MB QSPI Flash (SST26VF032BA)



28. Open Terra Term Terminal Program Tera Term and select under Setup->Serial Port the COM Port and the Baudrate 115200 (in this case a COM96, but could be a different COM Port number in your case).

The press the Reset Button of the Board and let the Firmware restart. You should see the start message with the Lab name and the build timestamp at first, followed by the MAC Address from EEPROM.

29. After some small time, the IP Address should be change from 0.0.0.0 to a valid address

| 🐸 COM96:115200baud - Tera Term VT | |
|---|---|
| File Edit Setup Control Window Help | |
| | * |
| > | |
| <pre>web_net_server_nvm_mpfs_freertos_lab1 Jun 3 2019 19:30:07 SYS_Initialize: The MPFS2 File System is mounted MAC TCPIP_HOSTS_CONFIGURATIONI01.macAddr: fc:c2:3d:0c:20:44 TCP/IP Stack: Initialization Started TCP/IP Stack: Initialization Ended - success Interface GMAC on host MARTIN_RUPPERT - NBNS enabled GMAC IP Address: 0.0.0.0 GMAC IP Address: 192.168.0.17</pre> | |
| | = |

30. To check the basic information about the network enter the netinfo command and press Enter.

```
💐 COM93:115200baud - Tera Term VT
File Edit Setup Control Window Help
⊳netinfo
  ----- Interface <eth0/GMAC> ------
Host Name: MARTIN_RUPPERT – NBNS enabled
IPv4 Address: 192.168.0.24
Mask: 255.255.255.0
Gateway: 192.168.0.1
DNS: 192.168.0.1
MAC Address: fc:c2:3d:0b:bf:f9
IPv6 Unicast addresses:
    fe80:0:0:0:fec2:3dff:fe0b:bff9
    2a02:908:1d41:d520:fec2:3dff:fe0b:bff9
IPv6 Multicast addresses:
    ff02:0:0:0:0:1:ff0b:bff9
    ff02:0:0:0:0:0:0:1
dhcp is ON
Link is UP
Status: Ready
) | I
```

31. A help shows the available commands

```
🌉 COM93:115200baud - Tera Term VT
File Edit Setup Control Window Help
≻help
----- Supported command groups -----
 *** iperf: iperf commands ***
 *** topip: stack commands ***
 ----- Built in commands -
 *** reset: Reset host ***
 *** g: guit command processor ***
 *** help: help ***
≻help tcpip
 *** netinfo: Get network information ***
 *** defnet: Set/Get default interface ***
 *** dhcp: DHCP client commands ***
 *** dhcps: Turn DHCP server on/off ***
 *** zcll: Turn ZCLL on/off ***
 *** setdns: Set DNS address ***
 *** setip: Set IP address and mask ***
 *** setgw: Set Gateway address ***
 *** setbios: Set host's NetBIOS name ***
 *** setmac: Set MAC address ***
 *** if: Bring an interface up/down ***
 *** stack: Stack turn on/off ***
 *** heapinfo: Check heap status ***
 *** ping: Ping an IP address ***
 *** arp: ARP commands ***
 *** dnsc: DNS client commands ***
 ***
```

32. As a first simple test you can ping an external Server or anything else you like (Maybe the board of your class neighbor?)

```
COM96:115200baud - Tera Term VT
File Edit Setup Control Window Help

web_net_server_nvm_mpfs_freertos_lab1 Jun 3 2019 19:30:07
SYS_Initialize: The MPFS2 File System is mounted
MAC TCPIP_HOSTS_CONFIGURATION[0].macAddr: fc:c2:3d:0c:20:44
TCP/IP Stack: Initialization Ended - success
Interface GMAC on host MARTIN_RUPPERT - NBNS enabled
GMAC IP Address: 192.168.0.17
>ping www.google.com
Ping: reply[1] from 172.217.23.132: time = 25ms
Ping: reply[2] from 172.217.23.132: time = 25ms
Ping: reply[4] from 172.217.23.132: time = 25ms
Ping: reply[4] from 172.217.23.132: time = 25ms
Ping: reply[4] from 172.217.23.132: time = 25ms
Ping: done. Sent 4 requests, received 4 replies.
```

33. Select the tcpip_discoverer tool from

| G System ► Computer ► System | n (C:) ► MASTERs ► 23075 ► h3 ► net ► utili | ities 🕨 tcpip_discoverer 🕨 | ✓ ← Search t |
|---------------------------------|---|----------------------------|---------------------|
| Organize 👻 Include in library 🔻 | Share with 👻 Burn New folder | | :::• |
| MASTERs | Name | Date modified | Туре |
| 4 23075 | src | 02.06.2019 18:27 | File folder |
| h3 | tcpip_discoverer | 02.06.2019 18:27 | Executable Jar File |
| | | | |
| | | | |
| | | | |
| l cipito | | | |
| dev packs | | | |
| mbc | | | |
| l net | | | |
| L apps | | | |
| 👢 config | | | |
| 👢 doc | = | | |
| 👢 docs | - | | |
| 👢 driver | | | |
| 👢 net_pres | | | |
| 👢 sys_adapter | | | |
| 👢 tcpip | | | |
| L templates | | | |
| Utilities | | | |
| l mib2bib | | | |
| | | | |
| tcpip_discoverer | | | |
| src | | | |
| web_pages | | | |
| | | | |
| lab1 | | | • |
| 2 items | | | |

34. If Windows is asking for permissions allow the access



35. The TCP Discover should list all boards in the classroom. We have made this tool to help you to find your board in the network. The source codes of this tool (Java) are part of the H3.

You can identify your board by the Host Name that has select in an earlier step. The Host Name is also known to the DHCP server and is listed in their typical Web Interfaces as a connected device.

| ▲ Microchip TCPIP Discoverer | × |
|---|---|
| Discover Devices Network Direct Broadcast Exit | |
| Microchip Devices Harmony MAC-Address - fc:c2:3d:0c:20:44 MAC-Type Hostname MARTIN_RUPPERT IP Addresses 192.168.0.17 F680:00:0:fec2:3dff:fe0c:2044 202:908:1d41:d520:fec2:3dff:fe0c:2044 Multicast Listeners Ff02:0:0:0:0:11 Hostname Ff02:0:0:0:0:0:1 | |
| Date - Nov, 12 2018 Version - TCPIP Discoverer 3.0 | |

Press the **Discover Devices** button: The tool will send a UDP broadcast on port 30303, with the packet "Discovery, who is out there?" All H3 devices running the Announce service will respond to this broadcast, by sending a return broadcast on port 30303. The broadcast packet contains data on the type of interface used, the Host Name, MAC and IP Address. The Discover tool listens to all broadcasts on port 30303 and will show found devices under the Microchip

Devices tree. You can identify your device by looking for the host name that you entered in MHC Setup process. The Microchip TCPIP Discoverer tool also shows the IP address for your board.

36. A double click on the MAC-Address line will put you in your default Internet Browser

| 🚳 Microchip TCPIP Discoverer | × |
|---|---|
| Discover Devices Network Direct Broadcast Exit | |
| Microchip Devices Harmony MAC-Address - fc:c2:3d:0c:20:44 MAC-Type GMAC Hostname MARTIN_RUPPERT IP Addresses 192.168.0.17 Fe80:0:0:0:fec2:3dff:fe0c:2044 2a02:908:1d41:d520:fec2:3dff:fe0c:2044 Multicast Listeners ff02:0:0:0:0:11 ff02:0:0:0:0:0:1 | |
| Date - Nov, 12 2018 Version - TCPIP Discoverer 3.0 | |

37. And the Webpage is displayed. Please take some time an play with the sub menus to find out the capabilities of or H3 Web Server



37. Congratulations, you have completed Lab 1!

Lab 2

Overview

You will be implementing an embedded network application to update and track status of an Vending Machine (VM). The implementation is sectioned into two modules.

ss rou'll see this on the second seco

Fanta

9

Dasani

0

TASK 1:

The task is to interface the up/down and select buttons and an OLED of a Vending Machine to the HTTP-Server. The HTTP-Server keeps track of the items in the Vending Machine.

The HTTP- Client (PC-web page) talks to the HTTP- server and displays the status of the Vending Machine. The HTTP-Client also sends out an update request to the HTTP- server using Ethernet Interface.

Vending Machine Demo The GET method appends the data to the end of the URL following the question mark (?) in your browser's address GET is automatically decoded, and stored in the current H

Pepsi

0

Copyright © 2018 Microchip Technology, Inc

ITEM 0 • COUNT 0 • UPDATE

> Dr Pepper

9

Minute Maid

0

Let us exconstically decoded, and stored in the cu buffer. Your application will handle the data in the r callback. TCPI_HTTP_Arguest function provides an e submitted values for processing. As an example, this GET form gets count of the VM updates the table

> Coca Cola Diet

Coca



TASK 2:

The task is to make all the VMs in the class network to talk to a server. When a Bay of a VM is empty it sends out a message to a common server. An example Message:

"Message: 1 from Martin: The Pepsi Bay is empty"

Міскоснір

The Message number (1) and the Host name (Martin) together forms a unique Message and makes it easy to differentiate your message from other VMs on the network server and the number helps to find the recent ones.





Application Implementation

The Vending Machine demo (Task 1 and 2) is implemented using "SAME70XplainedUltraEvaluationKit "and "OLED1XplainedProExtension Kit". The OLED Xplained pro has three buttons, LEDs and an OLED.

- The VM application uses Button 1(Down) and 2(UP) to scroll through the VM items and Button 3 to select an item from the Vending Machine.
- The LEDs above the button indicates a button press with a blink.
- The OLED is used to display the name and their corresponding number of the items in a Vending Machine.
- When select button(B3) is pressed the HTTP-Server will decrement the count of an item and updates the HTTP-Client(Web page)through ethernet interface.



E70 WEB SERVER /Vending Machine





The Vending machine demo runs the below user applications in conjunction with MPLAB Harmony TCP/IP Stack:

- **OLED and Button Contoller** (MMI_Tasks): Manages the operation of the OLED display, Buttons and LEDs.
- Application Contoller (app_Tasks()): manages all high level network communications with client and server.





Objectives

In Lab 1, you configured a new MPLAB Harmony Project from scratch that included the TCP/IP Stack middleware and a LED Flasher. The project you created in Lab 1 will be used as the starting point for Lab 2.. The source code for the OLED and Button Controller are included into the project.

You will learn about several Harmony TCP/IP API functions by adding the necessary APIs into the application source code to manage the TCP Sockets and data exchanged with socket. You will also learn how the dynamic variables in a web page are handled by the server. Finally, you will get to use the Packet Sender software tool to perform isolated testing of the embedded application prior to connecting your VM implementation to the server on the network.

Lab Procedure

Project Setup

1.2. Close Lab 1 project by choosing File>Close Project(lab1) in the main menu.



1.3. The project for Lab 2 has already been setup in advance. The project is a working implementation of Lab 1 that has been renamed to web_net_server_nvm_mpfs_lab2, and has a number of files added including mmi.c and modified app.c file are added to the project. To open the Lab 2 project, choose <u>File>Open Project</u> in the main menu.



- 1.4. In the Open Project window, enter C:\MASTERs\21070\ web_net_server_nvm_mpfs_lab2\firmware into the File name text box or navigate to the file .
- 1.5. Click on *sam_e70_xult_freetos.X* icon in the file list.

1.6. Press Open Project.

| 🗴 Open Proje | ct | × |
|--------------|--|--------------|
| Recent Items | Look in: lab2 Forget Name: Forget Name: Forget Nam | _lab2 |
| Desktop | | |
| Documents | | |
| This PC | | |
| 4 | File name: C:\WASTERs\23075\Jab2\frmware\sam_e70_xult_freertos.X |)pen Project |
| Network | Files of type: Project Folder | Cancel |

1.7. The source and header structure for the project is shown below.



MHC Code Generation

- **1.8.** The project configurations are already set, and we are going to look on the configurations and generate the code. (steps 1.8 -1.22)
- 1.9. Before the MHC Configuration tool can be used, you need to set lab2 as the main project, by right clicking on the web_net_server_nvm_mpfs_lab2 folder under the Projects window, and choosing Set as Main Project in the popup menu.



1.10. To launch MHC, choose **<u>Tools > Embedded > MPLAB Harmony 3 Configurator</u>** in the main menu.

| Tools | Window Help | | |
|-------|-------------|---|---------------------------------------|
| E | Embedded | ٢ | MPLAB® Harmony 3 Framework Downloader |
| L | icenses | | MPLAB® Harmony 3 Configurator |

1.11. In the MPLAB Harmony Launcher dialog box, make sure the path states

C:\MASTERs\23075\h3 and click on Launch.

| 😤 MPLAB Harmony Launcher | | | | | | |
|---|--|--|--|--|--|--|
| Active Project Information: | | | | | | |
| Name: sam_e70_xult_freertos | | | | | | |
| Select an option to use for the launch paths for the MPLAB Harmony Framework tool suite: | | | | | | |
| MPLAB Harmony Project Path - The framework path that is stored with this MPLAB Harmony project. Reconfigure Paths Harmony Path: C:\MASTERs\23075\h3 | | | | | | |
| MPLABX Path - The framework path that is stored with MPLABX (configured through Tools->Options->Harmony) | | | | | | |
| Harmony Path: C:\microchip\h3 | | | | | | |
| Default Launch Action: Prompt For Path Selection (This (Configurable through Tools->Options->MPLAB Harmony) Launch Cancel | | | | | | |
| | | | | | | |

1.12. In the nest window the H3 parts and their used Version Number are displayed. Click on Launch

| oad | Name | Version | Dependencies |
|---------------|------------------------------|---|------------------------------------|
| 1 | bsp | v3.3.0 | csp(3.2.1) |
| 1 | core | v3.2.1 | csp(3.2.1) |
| | crypto | v3.2.1 | core(v3.2.1) |
| ~ | csp | v3.2.1 | |
| v | net | v3.3.0 | core(3.2.1), csp(3.2.1), dev_packs |
| | | | |
| nfigure | e Device Family and CMSIS Pa | ck Paths: | |
| nfigure P: | a Device Family and CMSIS Pa | ck Paths: SAME70_DFP\4.0.26\same70b\atdf\AT: | 5AME70Q21B.atdf |

1.13. Open the saved state file

| 🚑 Open Default Saved State | \times |
|---|----------|
| Open default saved state file? | |
| C:\MASTERs\23075\ab2\firmware\src\config\sam_e70_xult_freertos\sam_e70_xult_freertos. | xml |
| Open Cancel | |

- **1.14.** Explore and look on the configurations tree by clicking on the modules under Active Components (step 1.14 and 1.15- can be skipped and continue from 1.16).
- 1.15. Click and check the "Core" option under Active Components. On right hand side the Configuration options are displayed, check the "Number of Applications "options under Configuration Options which is set to 2.Expand both Application 0 and 1 Configuration trees, and the Application Name is set to app (Application Interface) for Application 0, and mmi(Man Managed Interface for buttons, leds and OLED) for Application 1.
- **1.16.** Check on **TCP/IP Stack** under Active components and its configuration under **Configuration options**.

| File Jiects Active Components | Configuration Options Help |
|--------------------------------|--|
| | |
| CMSIS Pack | enerate Harmony Application Files 🗹 |
| | Application Configuration Number of Applications 2 |
| Cryptographic (Crypto) Library | - Application 0 Configuration |
| EFC | Application Name app |
| FILE SYSTEM | The state of the s |
| FreeRTOS | Application 1 Configuration |
| | Application Name mmi |
| Presentation Layer | Application name must be valid C-Language identif |
| SAM E70 Xplained Ultra BSP | RTOS Configuration |
| SPIO | Enable System Interrupt 🗹 |
| System | Enable System Ports |
| TCO | Enable System Cache |
| | Enable System DMA |
| TCP/IP STACK | Enable OSAL |

1.17. Select the Instance 0 in the TCP/IP stack under Active Components to enter the HOST Name (TCP/IP STACK->BASIC CONFIGURATION->Instance 0). And change in the Configuration Options (displayed on the right side of MPlabx) the Host Name to your firstname_lastname. The Host name can be identified in the Network.

| 🗋 Active Components 💼 🛃 🗖 | | |
|----------------------------|----------------------------------|-------------------|
| | Configuration Options* Help | |
| CMSIS Pack | - + | |
| CONSOLE | — — | |
| Core | | |
| EFC | ····Network Configurations Index | 0 |
| FILE SYSTEM | Interface | GMAC |
| FreeRTOS | Host Name | RAJI_SHAN |
| Presentation Layer | Mac Address | 00:04:25:1C:A0:02 |
| SAM E70 Xplained Ultra BSP | | 192.168.100.11 |
| System | IPv4 SubNet Mask | 255.255.255.0 |
| | | 192.168.100.1 |
| BASIC CONFIGURATION | IPv4 Primary DNS | 192.168.100.1 |
| | - IPv4 Secondary DNS | 0.0.0.0 |
| TCP/IP Basic Configurator | Power Mode | full |
| TCPIP CMD | Network Configuration Start-up | Flags |
| TCPIP CORE | Network MAC Driver | DRV_GMAC_Object |
| TCPIP File System Wrapper | | |
| | | |
| | | |
| | | |
| I THE | | |

1.18. Select "Code" (Generate Code) which is under the menu tabs.

| ÷ | MPL | AB Harr | mony Configur | ator - sam_e/ |
|-----|--------|----------|-----------------|---------------|
| ile | Ger | nerate 1 | Tools Htilities | Window |
| 1 | ₿ | 9 🕈 | Code | |
| n, | Active | e Compoi | nents | |

1.19. Select Don't Save



1.20. Select Generate

| 🚇 Generate Proje | ect 💽 💌 |
|--|----------------------------|
| 1. Configure Gener | ation Settings |
| Generate Settir Merge Strategy | Igs USER RECENT ▼? |
| (Mouse over a p | roperty for detailed help) |
| 2. View Warnings | |
| Туре | Description |
| | |
| 3. Click Generate | Generate Cancel |

1.21. Some Files will be changed, and the MHC is asking in a "diff" window, if the changes should be taken over. Don't accept the changes for "custom_http_net_app.c" file just click on close(step 2 in pic). Accept all changes for the other files by clicking on the Arrow in the middle and then close.

| Merging: configuration.h_7137651041604723379.old | | 1 | | | |
|--|--|--|-----|--|--|
| nding Merge Actions: 1 | | - Next | - 1 | Previous | |
| | | | | | Note: The content of the right pane displays the current state |
| Gener | ated Code | 1/1 | | Current File: configuration.h_71376 | 551041604723379.old |
| | | 324 🕸 | 24 | | |
| | | 325 | 325 | | |
| ine TCPIP IPV6 NDP MAX RTR SOLICITATION DELAY | 1 | 326 | 326 | #define TCPIP IPV6 NDP MAX RTR SOLICITATION DELAY | 1 |
| THE ICEPT IPV6 NDP KIK SOLICITATION INTERVAL | 1 | 327 | 327 | Adeline TOPIP IPV6 NDP KIK SOLICITATION INTERVAL | 9 |
| THE TOPIE IPV6 NDE MAX KIR SUBJOIRTIONS | 3 | 320 | 320 | Adefine TOPID IDVE NDD MAX MULTICAST SOLICITATIONS | 3 |
| The ICPIP IPV6 NDP MAX MULTICASI SOLICII | 3 | 329 | 329 | #define TCPIP IPV6 NDP MAX MOLIICASI SOLICII | 3 |
| THE TOPIP IPV6 NDP MAX UNICASI SUBTCIT | 3 | 330 | 330 | Wdefine ICPIP IPV6 NDP MAX ONICASI SOLICII | 3 |
| THE ICEIP IEVS NOP MAX ANICASI DELAI TIME | 1 | 331 | 331 | Adefine TCPIP IPV6 NDP MAX ANICASI DELAI TIME | 1 |
| THE TOPIP IPV6 NDP MAX ALIGHBOR ADVERTISEMENT | 3 | 334 | 334 | Adefine TOPID IDVS NDD DEACHADIE TIME | 3 |
| ING TOPID IDUE NDD DETRING TIMED | 1 | 333 | 224 | tdefine TOPIP IPV6 NDP REACHABLE TIME | 1 |
| THE TOPIC IPVO NOT REINARD TIMER | - | 225 | 332 | Adding TONTO TONG NOD DELAY PIDET DOOR TIME | - |
| THE TOPIE INVENDE UNITE LIFETIME THE HOUSE | (60 * 60 * 2) | 335 | 226 | Adefine TODID TOUS NOD UNITE LIPSTING TWO HOUDS | (60 * 60 * 2) |
| ING TODID IDUS MTU INCREASE TIMEOUT | (60 - 60 - 2) | 330 | 227 | #define TOPID IDVE MTU INCREASE TIMEOUT | (60 - 60 - 2) |
| ING TOPID IDUE NDD TASK TIMED DATE | 32 | 337 | 220 | tdefine TOPIP IPV6 MIG INCREASE TIMEOUT | 22 |
| | 02 | 339 | 336 | Facture Total Trive and Triber Table | 00 |
| | | 340 | 340 | | |
| twork Configuration Index 0 */ | | 341 | 341 | /* Network Configuration Index 0 */ | |
| TO TOTE NETWORK DEFNUET INTERENCE NAME TOYO | IIGMD CI | 242 | 242 | fdefine TORIE NETWORK DEFAULT INTERFACE NAME IDVO | "GMD.C" |
| TOTT TE CANC | on ac | 343 | 343 | tdefine TOPTE IE GMAC | VIII C |
| | | 344 | 344 | | |
| ine TCPIP NETWORK DEFAULT HOST NAME IDX0 | "RAJI SHAN" | ··>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> | 345 | #define TCPIP NETWORK DEFAULT HOST NAME IDX0 | "MCHPBOARD C" |
| ine TCPIP NETWORK DEFAULT MAC ADDR IDX0 | "00:04:25:1C:A0:02" | 346 | 346 | #define TCPIP NETWORK DEFAULT MAC ADDR IDX0 | "00:04:25:1C:A0:02" |
| | | 347 | 347 | | |
| ine TCPIP NETWORK DEFAULT IF ADDRESS IDX0 | "192.168.100.11" | 348 | 348 | #define TCPIP NETWORK DEFAULT IP ADDRESS IDX0 | "192.168.100.11" |
| ine TCPIP NETWORK DEFAULT IP MASK IDX0 | "255.255.255.0" | 349 | 349 | #define TCPIP NETWORK_DEFAULT_IP_MASK_IDX0 | "255.255.255.0" |
| ine TCPIP NETWORK DEFAULT GATEWAY IDX0 | "192.168.100.1" | 350 | 350 | #define TCPIP NETWORK DEFAULT GATEWAY IDX0 | "192.168.100.1" |
| ne TCPIP NETWORK DEFAULT DNS IDX0 | "192.168.100.1" | 351 | 351 | #define TCPIP NETWORK DEFAULT DNS IDX0 | "192.168.100.1" |
| ine TCPIP NETWORK_DEFAULT_SECOND_DNS_IDX0 | "0.0.0.0" | 352 | 352 | #define TCPIP NETWORK DEFAULT SECOND DNS IDX0 | "0.0.0.0" |
| ne TCPIP NETWORK DEFAULT POWER MODE IDX0 | "full" | 353 | 353 | #define TCPIP NETWORK_DEFAULT_POWER_MODE_IDX0 | "full" |
| ine TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 | Λ | 354 | 354 | #define TCPIP_NETWORK_DEFAULT_INTERFACE_FLAGS_IDX0 | χ |
| | TCPIP_NETWORK_CONFIG_DHCP_CLIENT_ON \ | 355 | 355 | | TCPIP_NETWORK_CONFIG_DHCP_CLIENT_ON \ |
| | TCPIP_NETWORK_CONFIG_DNS_CLIENT_ON \ | 356 | 356 | | TCPIP_NETWORK_CONFIG_DNS_CLIENT_ON \ |
| | TCPIP_NETWORK_CONFIG_IP_STATIC | 357 | 357 | | TCPIP_NETWORK_CONFIG_IP_STATIC |
| | | 358 | 358 | | |
| ne TCPIP_NETWORK_DEFAULT_MAC_DRIVER_IDX0 | DRV_GMAC_Object | 359 | 359 | #define TCPIP_NETWORK_DEFAULT_MAC_DRIVER_IDX0 | DRV_GMAC_Object |
| | | 360 | 360 | | |
| | | 361 | 361 | | |
| | | 362 | 362 | | |
| topip cmd Configuration ***/ | | 363 | 363 | /*** topip cmd Configuration ***/ | |
| ine TCPIP STACK COMMAND ENABLE | | 364 | 364 | #define TCPIP_STACK_COMMAND_ENABLE | |
| | | | | | |

1.22. The whole process is displayed with a progress bar

| Generating Project | | | | × |
|--|-----------|----|-------|-------------|
| Task Type | Remaining | | Total | |
| File Markup | | 0 | | 482 |
| File Copy | | 52 | | 482 |
| Libraries | | 0 | | 0 |
| Settings | | 8 | | 8 |
| Source Paths | | 0 | | 0 |
| | | | | |
| Generating file: C:\MASTERs\23075\h3\crypto\src\de | es3.c | | | |
| | 94% | | | Please Wait |

- 1.23. After the MHC has finished generating the project, go to the Projects Window and expand the Header Files and Source Files to see the source/header files for the app and mmi application file and TCP/IP stack files.
 - > The WEB page source code for the VM application is found under the folder web-pages->VM.htm
 - The hex file for the web page is in mpfs_net_img.c which is generated using the utility MPFS- generator which comes along with Harmony net package.



Application Source and Header File Setup TASK1:

In the VM application when an VM ITEM update is initiated by the HTTP- client (WEB PAGE) it sends out a GET command that needs to be processed by the HTTP-Server. The get command is sent along with the argument of the VM "Item" to be updated and its "Count" which are processed by the HTTP-Server. In Harmony TCP/IP stack GET command is handled by the function TCPIP HTTP NET ConnectionGetExecute ().

Follow the procedure under to include the code that process the GET request from HTTP-CLient: -

1.24. Click and Open the file custom_http_net_app.c. Search for the function TCPIP_HTTP_NET_ConnectionGetExecute () (Approx. Line.No. 247) in which the below code snippets need to be inserted.



1.25. Find the TODO[1] and insert the below code starting from the line mentioned as "//<--Insert the solution for TODO[4] starting on this line". Code can be inserted either by typing or copy and paste which is provided for your convenience at the end of the file (custom_http_net_app.c).</p>

ptr = TCPIP_HTTP_NET_ArgGet(httpDataBuff, (const uint8_t *)"ITEM");

1.26. Find the TODO[2] and insert the code snippet starting from the line mentioned as "//<--Insert the solution for Item 2 starting on this line". Code can be inserted either by typing or copy and paste which is provided for your convenience at the end of the file (custom_http_net_app.c).</p>

ptr = TCPIP_HTTP_NET_ArgGet(httpDataBuff, (const uint8_t *)"COUNT");

VM_Count[Update_Item]=*ptr;

TASK 2:

When a Bay of the VM is empty it sends out the message to the common server. The sate machine to open a socket to send message and close socket is shown below



In this section, you will gain some experience with the use of Harmony TCP/IP API functions. The app.c source is missing lines of code. All missing code specifically relates to management of the setting up the TCP Client such as opening or closing the socket, checking if the socket is connected, checking if data is available, and writing data to the socket. Your task is to read the description for each missing item, select the appropriate Harmony TCP/IP API, and fill in the missing line of code.

- 1.27. The first step to do is get the common server's IP address. Get the address from the presenter and make a note.
- **1.28.** Click and Open the file app.c. and find the function APP_Tasks (). (Approx. Line.No. 200) in which the below code snippets are inserted.



1.29. Find the TODO[3] in which common SERVER IP address is entered. Replace the x with the common server IP address (). For example: if the server IP address is 192.168.0.108 replace

- * x1 with 192
- * x2 with 168
- * x3 with 0
- * x4 with 108
- **1.30.** Find TODO[4] and add the below snippet to Open a socket for TCP_Client . Code can be inserted either by typing or copy and paste the code which is provided for your convenience at the end of the file (app.c).

appData.socket = TCPIP_TCP_ClientOpen(IP_ADDRESS_TYPE_IPV4,

appData.port,

(IP_MULTI_ADDRESS*) & addr);

1.31. Find TODO[5] and add the below snippet to check the TCP connection is established 0.

Insert the code in the line mentioned as *"//<--Insert the solution for TODO[5] starting on this line"* Code can be inserted either by typing or copy and paste the code which is provided for your convenience at the end of the file (app.c). if (!TCPIP_TCP_IsConnected(appData.socket))

1.32. Find TODO[6] and add the below snippet to check the TCP connection is established .

Insert the code in the line mentioned as *"//<--Insert the solution for TODO[6] starting on this line"* Code can be inserted either by typing or copy and paste the code which is provided for your convenience at the end of the file (app.c).

if (TCPIP_TCP_PutIsReady(appData.socket) == 0)

1.33. Find TODO[7] and add the below snippet to send the message to the common server once a TCP connection at port 80 is established.

Insert the code in the line mentioned as "//<--Insert the solution for TODO[7] starting on this line" Code can be inserted either by typing or copy and paste the code which is provided for your convenience at the end of the file (app.c).

sprintf(buffer, "MSG:%d from %s : %s is empty", (int) MessageCounter++, (char *)
TCPIP_HOSTS_CONFIGURATION[0].macAddr, (char *) VM_Items[bay_index - 1]);

SYS_CONSOLE_PRINT("Sending message: %s\r\n", buffer);

TCPIP_TCP_ArrayPut(appData.socket, (uint8_t*) buffer, strlen(buffer));

1.34. Find TODO[8] and add the below snippet to close the TCP Client socket.

Insert the code in the line mentioned as *"//<--Insert the solution for TODO[8] starting on this line"* Code can be inserted either by typing or copy and paste the code which is provided for your convenience at the end of the file (app.c).

TCPIP_TCP_Close(appData.socket);

Hardware Setup

This hardware setup is common to both TASK1 and TASK2. After adding the required code for the application, make the hardware connections to programme the device and to see the output



1.35. Network Connection

Cable: CAT5 Ethernet Cable supplied with Starter Kit Connection: RJ45 Jack from the class network to PCB Top RJ45 Jack(make sure the jumper below the Ethernet PHY module is taken off)

1.36. Programming and Console Connection

OLED1 X PLAINED PRO



Cable USB Male A to USB Male B Micro cable Connection: USB Debug Port on PCB Top to Laptop USB Port

3

1.37. OLED1 Xplained Pro Connection

Connection: Connect the OLED x PLAINED PRO to the EXT 2 of the board (Extension close to Ethernet PHY)

1.38. PC



PC needs to be connected to the Class network or It can be connected to the wireless network

1.39. In the main window of MPLABX, click on the "Make and Program Device" button



1.40. After successful build, the SAME70 is programmed automatically. Build and Programming are monitored on the output window at the bottom of the MPLAB X IDE screen.



1.41. Immediately after programming, you can see the text on the OLED screen in series like the screens below.

Use the Buttons 1 and 2 (Previous and Next item) on the OLED1 Xplained pro to scroll through the items of the Vending Machine.



1.42. Now open a Tera Term to see the status of the HTTP server and its IP address.

If Tera Term is still open from Lab 1, click on the window title bar to make it active. If Tera Term needs to be re-opened and configured, refer the below steps.



Open Terra Term Terminal Program Tera Term and select under" Setup->Serial Port" (in this case a COM14, but could be a different COM Port number in your case) set the speed to 115200.Match the below set-up and click "OK "

| Tera Term: Serial port setu | ip | | \times |
|-----------------------------|---------------|--------|----------|
| Port: | COM14 | ~ | ок |
| Speed: | 115200 | ~ | |
| Data: | 8 bit | ~ | Cancel |
| Parity: | none | \sim | |
| Stop bits: | 1 bit | \sim | Help |
| Flow control: | none | \sim | |
| Transmit dela | y c/char 0 | ms | ec/line |

1.43. Reset the Board and let the Firmware restart.



1.44. After some small time, the IP Address should change from 0.0.0.0 to a valid address. Take time to go through the Initialization messages, MAC address, Host name and the IP address of the HTTP- server displayed on the tera term terminal.

| File Edit Setup Control Window Help web_net_server_num_mpfs_freertos_lab1 Jun 6 2019 09:01:39 SYS_Initialize: The MPFS2 File System is mounted MAC TCPIP HOSTS_CONFIGURATIONION.macAddr: fc:c2:3d:0d:21:d7 CCP/IP Stack: Initialization Started CCP/IP Stack: Initialization Ended - success Interface GMAC on host RAJI_SHAN - NBNS enabled (null) -sends message to the server when a Bay is empty MAC IP Address: 0.0.0.2 MAC IP Address: 10.13.33.76 | |
|---|--|
| web_net_server_nvm_mpfs_freertos_lab1 Jun 6 2019 09:01:39 YS_Initialize: The MPFS2 File System is mounted MCC TCPIP_HOSTS_CONFIGURATION[0].macAddr: fc:c2:3d:0d:21:d7 CP/IP Stack: Initialization Started ICP/IP Stack: Initialization Ended - success Interface GMAC on host RAJI_SHAN - NBNS enabled (null) -sends message to the server when a Bay is empty MAC IP Address: 0.0.0.0 MAC IP Address: 10.13.33.76 | |
| web_net_server_nvm_mpfs_freertos_lab1 Jun 6 2019 09:01:39 YS_lnitialize: The MPFS2 File System is mounted MAC TCPIP_HOSTS_CONFIGURATION[0].macAddr: fc:c2:3d:0d:21:d7 CP/IP Stack: Initialization Started ICP/IP Stack: Initialization Ended - success Interface GMAC on host RAJI_SHAN - NBNS enabled (null) -sends message to the server when a Bay is empty MAC IP Address: 0.0.0.0 MAC IP Address: 10.13.33.76 | |
| <pre>web_net_server_nvm_mpfs_freertos_lab1 Jun 6 2019 09:01:39 YSL_Initialize: The MPFS2 File System is mounted MAC TCPIP_HOSTS_CONFIGURATION[0].macAddr: fc:c2:3d:0d:21:d7 CP/IP Stack: Initialization Started ICP/IP Stack: Initialization Ended - success Interface GMAC on host RAJI_SHAN - NBNS enabled Cnull) -sends message to the server when a Bay is empty MAC IP Address: 0.0.0 MAC IP Address: 10.13.33.76</pre> | |
| MAC IP Address: 10.13.33.76 | |
| AC TCPIP_HOSTS_CONFIGURATION[0]_macAddr: fc:c2:3d:0d:21:d7 CP/IP Stack: Initialization Started CP/IP Stack: Initialization Ended - success Interface GMAC on host RAJI_SHAN - NBNS enabled null> -sends message to the server when a Bay is empty MAC IP Address: 2.2.2.2 MAC IP Address: 10.13.33.76 | |
| CP/IP Stack: Initialization Ended - success Interface GMAC on host RAJI_SHAN - NBNS enabled null> -sends message to the server when a Bay is empty MAC IP Address: 2.2.2.2 MAC IP Address: 10.13.33.76 | |
| Interface GMAC on host RAJI_SHAN - NBNS enabled null> -sends message to the server when a Bay is empty MAC IP Address: 9.9.9.9 MAC IP Address: 10.13.33.76 | |
| null) -sends message to the server when a Bay is empty MAC IP Address: <u>9.9.9.9</u> MAC IP Address: 10.13.33.76 | |
| MAC IP Address: 2.2.2.2 MAC IP Address: 10.13.33.76 | |
| MAC IP Address: 10.13.33.76 MAC IP Address: 10.13.33.76 | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

1.45. Open a web browser and enter the HTTP- Server IP address that you got on the tera term in the previous step. This opens a web page (HTTP-Client)

| | | | HARMONY |
|--------------------------|---|---|---|
| | | | TCP/IP Stack Demo Applicati |
| Overview | Welcome! | | LED: |
| Dynamic Variables | Stack Version: | 7.32 - H3.2 | • |
| SSI Processing | Build Date: | Jun 6 2019 09:01:40 | Buttons: |
| Processing Forms | File System | FLASH | Random Number: |
| Authentication | File System Type: | MPFS2 | 12993 |
| Cookies | This site demonstrates | the power, flexibilit | y, and scalability of a 32-bit |
| Uploading Files | embedded web server. microcontroller running | Everything you see the Harmony Micro | is powered by a Microchip PIC ochip TCP/IP Stack. |
| Sending Emails | On the right you'll see t | the current status o | f the demo board. For a quick |
| Dynamic DNS | buttons (except MCLR!) | and you'll see the | s on the board. Press the push status update immediately. This |
| Network Configuration | This site is provided as | a tutorial for the va | arious features of the HTTP web |
| SNMP Configuration | Dynamic Variab Server Side Inc | le Substitution - | display real-time data |
| Vending Machine | Processing Forr Authentication Cookies - store s Uploading Files | ns - handle input fr - require a user nar session state inform - parse files for cor | om the client ne and password lation for richer applications nfiguration settings and more |
| | Several example applic parameters, sending er to built-in GZIP compre the 32kB of on-board M | ations are also prov nails, and controllin ssion support, all ti lemory. | vided for updating configuration ig the Dynamic DNS client. Thank: hese tutorials and examples fit in |
| | For more information o TCP/IP Stack Libraries I on your computer as pa | n the Harmony TCP Help paragraph in t art of the Harmony | /IP Stack, please refer to the he MPLAB Harmony Help installed distribution. |
| | Copyright © 2018 | Microchip Technology, I | nc. |

1.46. Click on the Vending Machine tab to see the page for the Vending Machine Demo.

| | | | HARMONY | 🕂 Міскоснір | | | | | | MPL | B |
|--|--|--|--|--|------------------------|--------------------------|---------------|------------------|-------------|-------------------|----------|
| | | | TCP/IP Stack Demo Application | | | | | | | | |
| verview | Welcome | | | | | | | 1 | CP/IP Sta | ick Demo | Applicat |
| mamic Variables | Stack Varciani | 7 22 - 112 2 | LED: | Overview | Vend | dina I | Mach i | ine De | emo | | |
| Processing | Build Date: | Jun 6 2019 | Buttons: | Dynamic Variables | The OFT | | | | | • Marchill and | |
| ocessing Forms | File System Location: FLASH Random Number: 12993 SSI Processing Following the question mark (2) in your browser's address bar. Data se GET is automatically decoded, and stored in the current HTTP connectif | | | | | | | | ection of | | |
| kies | File System Type: MPF52 | | | | | | | | | xecute etrieve | |
| nding Emails namic DNS twork nfiguration | On the right you'll see example, click the LED: buttons (except MCLR! examples uses AJAX te This site is provided as server, including: | the current status of s to toggle the lights) and you'll see the s ichniques to provide a tutorial for the var | the demo board. For a quick on the board. Press the push tatus update immediately. This eal-time feedback. ious features of the HTTP web | Cookies Uploading Files Sending Emails | As an exa updates t | mple, this (he table | SET form g | 1 COUN UPDATE | the VM item | ns as user i | nput an |
| NMP onfiguration • Server Side Includes - process real-time SSL commands | | | | Dynamic DNS | VM TRACKER | | | | | | |
| iding Machine | Authentication Cookies - store Uploading Files Several example applic parameters, sending ei | require a user nam session state information parse files for configurations are also provi- mails, and controlling | e and password tion for richer applications īguration settings and more ded for updating configuration the Dynamic DNS client. Thanks | Configuration SNMP Configuration | Coca- Cola | Coca- Cola Diet | Pepsi | Dr Pepper | Sprite | Fanta | Dasa |
| | to built-in GZIP compre the 32kB of on-board N | ession support, all the 1emory. | ese tutorials and examples fit in | Vending Machine | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| | For more information o TCP/IP Stack Libraries on your computer as p | on the Harmony TCP/ Help paragraph in th art of the Harmony d | IP Stack, please refer to the e MPLAB Harmony Help installed istribution. | | | | | 1 | | | I |
| | | | | | C | opyright © 20 | 18 Microchip | Technology, Inc. | | | |

1.47. Use the Button 3 on the OLED to pick an item from the VM. The count of the corresponding item on the OLED scren will be decremented by 1 and is reflected on the Web page as well as on the OLED screen.



loav In

1.48. Check on the VM update. Pick an item and the count to be changed on the WEB page from the ITEM and COUNT drop down menu. Select update. When an update is selected the WEB page will send a GET command along with the ITEM number and COUNT to the HTTP-server. The HTTP-server decodes the GET command and displays it on the VM diaply (OLED).

| verview ynamic Variables SI Processing ocessing Forms Ithentication okies | Vend The GET m following tI GET is auto buffer. You callback. Tr submitted As an exan | ethod appered e question matically dd application PIP_HTTP_A values for propered aple, this GF | Iachi ands the da mark (?) i ecoded, an will hand urgGet fund processing. | T ne De ata to the end in your brows nd stored in t le the data in tion provides | CP/IP Sta MO I of the UR ier's addrese be current | ick Demo / | Applicatio | on | | Overview | Vend | lina N | 1achi | | TCP/I | P Sta | ck Demo |
|---|---|--|---|---|--|---|---|----|--|--|---|--|--|--|--|---|---|
| verview ynamic Variables SI Processing rocessing Forms thentication pokies | Vend The GET m following th GET is auto buffer. You callback. To submitted As an exan | ethod apper equestion matically de application .PIP_HTTP_A values for pr apple, this GF | nds the da mark (?) i ecoded, an will hand urgGet fund rocessing. | ne De ata to the end in your brows nd stored in t le the data in ction provides | mo d of the UR ter's addres | I. You'll see | | | | Overview | Vend | lina N | 1achi | | - | _ | |
| ynamic Variables SI Processing rocessing Forms uthentication pokies | The GET m following th GET is auto buffer. You callback. To submitted As an exan | ethod apper e question matically de application PIP_HTTP_A values for pr pple, this GF | ends the da mark (?) i ecoded, an will hand argGet fund rocessing. | ata to the end in your brows nd stored in t le the data in ction provides | d of the UR er's addres | I. You'll see | | | | | | mg r | acm | ne De | enne | | |
| | updates th | table | ET form ge | ets count of the | the TCPIP an easy m | ss bar. Data HTTP conn _HTTP_GetE: nethod to n | e this data a sent via ection data xecute etrieve nput and | a | | Dynamic Variables SSI Processing Processing Forms Authentication Cookies | The GET n following t GET is aut buffer. You callback. T submitted As an exar updates th | hethod app he question omatically r applicatio CPIP_HTTP_ values for nple, this C e table | ends the d n mark (?) decoded, a on will hand ArgGet fun processing GET form g | ata to the e in your brow nd stored in lle the data ction provid ets count of | nd of th vser's a the cu in the ⁻ es an e the VM | he UR: addres irrent TCPIP_ asy m 1 item | I. You'll se is bar. Dat HTTP conr HTTP_GetE lethod to r s as user i |
| ploading Files | TTEM 1 Violating Files TTEM 0 COUNT 0 Violating Files TEM 1 Violating Files TEM 0 Violating Files | | | | | | | | | | | | | | | | |
| mamic DNS | | | | 1 2 | | | | | | Dynamic DNS | | | | VM TRACKE | R 3 | | |
| twork nfiguration MP | Coca- Cola | Coca- Cola | Pepsi | 3 TRACKER 4 5 6 7 Pepper | Sprite | Fanta | Dasani | | | Configuration SNMP Configuration | Coca- Cola | Coca- Cola Diet | Pepsi | Dr Pepper | 5 6 7 8 9 | ite | Fanta |
| nfiguration | | Diet | | | | | | _ | | venting Machine | 8 | 9 | 9 | 9 | 4 | | 9 |
| | 8 | 9 | 9 | 9 | 9 | 9 | 9 | | | | Co | pyright © 20 | 18 Microchip | Fechnology, Inc | | | |
| | Co | oyright © 2018 | 8 Microchip 1 | ēchnology, Inc. | <u> </u> | AICROC | THIP | | | L | H P Stack D | | r | | | | |
| | | | | | | | | | | ТСР/1 | P Stack De | emo App | lication | | | | |



Copyright © 2018 Microchip Technology, In

Example:

On the web page ITEM 4 and its count will be changed to 4 and, also on the OLED.

| | | ITEM | 0 V COUN | ▼ 0 ▼ | | | | |
|---------------|-----------------------|-------|--------------|--------------|-------|--------|---|-------|
| | | | VM TRACKER | ı. | | | | |
| Coca- Cola | Coca- Cola Diet | Pepsi | Dr Pepper | Sprite | Fanta | Dasani | | 4. Dr |
| 8 | 9 | 9 | 4 | 9 | 9 | 9 | | |
| | | | | 1 | L | |] | |

You have successfully completed TASK 1 and let's move on to TASK 2.

- 1.49. To test Task 2, one or more Bay of the VM machine must be Zero. Press the select Button 3 and make one or more item to zero count. After few second you can see the messages " Sending VM status ", "Starting Connection" and "TCP Socket Connected" and the "BAY Empty message" and TCP client closed on the tera term.
- **1.50.** Look at the presenter's screen to identify your message.

MSG:338 from fc:c2:3d:0d:21:d7 : 2.Diet-Coke is empty

| Message No. | MAC address |
|-------------|-------------|

Lab 3

Overview

In many IoT applications, JSON is commonly used as a format in order to transport high-level data in an effective way. It is generally an alternative to XML. Consider the following example of describing a person named Raji-Niklas Ruppert in JSON-format:

```
"firstName" : "Raji-Niklas",
"lastName" : "Ruppert",
"age" : 30,
"address" : {
    "streetAddress" : "2355 W Chandler Blvd",
    "city" : "Chandler",
    "state" : "AZ",
    "postalCode" : "85224",
}
```

}

{

Using this format makes it very easy to communicate between applications requiring information about Raji-Niklas Ruppert. The advantage of using JSON in embedded applications is that because it is easy to read for humans, it is simple to parse and make use of. Due to this, it is commonly used to transmit data between a server and a web application. In this lab we are going to implement an embedded application fetching weather data from a web server. When the application accesses a specific URL specifying a command with a geographic location, the web server will respond by sending the current weather in JSON-format to the web application. The application will be running on our SAME70-boards.

In this lab we will only do very simple parsing (which is one of the strengths using JSON), using standard string operations. There are however more sophisticated parsers which can be used for more robust and complex applications, while still only consuming a very limited footprint.

The weather service used in this lab is <u>https://openweathermap.org/</u>. With OpenWeatherMap, there are several services such as hourly forecast, UV Index, Air pollution and more, all outputting in JSON. With the free account there are limited option to only use the "Current Weather Data" service. With this service you can request the current weather from different geographic locations. Depending on by which method (City ID, ZIP Code, Coordinates etc.) the URL call will be slightly different. A full description of the API can be found here: <u>https://openweathermap.org/current</u>. For this lab we will fetch current weather by city. The following URL for this is:

http://api.openweathermap.org/data/2.5/weather?q={CITY}&APPID={API Key}

The API Key is unique to each user. This is also how OpenWeatherMap tracks how many requests you attempt. The API Key is a 15byte long hexadecimal string. It can look like this:

ed3da58111974261002c2af4f8e8e81f

In most JSON API:s there is also a well defined format specified, which tells you where the different objects and strings are located in the JSON-message. From OpenWeatherMap:

```
{"coord":{"lon":-122.09,"lat":37.39},
"sys":{"type":3,"id":168940,"message":0.0297,"country":"US","sunrise":1427723751,"sunset":14
27768967},
"weather":[{"id":800,"main":"Clear","description":"Sky is Clear","icon":"01n"}],
"base":"stations",
"main":{"temp":285.68,"humidity":74,"pressure":1016.8,"temp_min":284.82,"temp_max":286.48},
"wind":{"speed":0.96,"deg":285.001},
"clouds":{"all":0},
"dt":1427700245,
"id":0,
"name":"Mountain View",
"cod":200}
```

Application Flow



Lab Outline

- In the the source code file app,c are TODO's. At this places you have to change the source codes. At the bottom of the app.c you find the solutions. Either you think about what you have to change, or you just copy the solution at the right place.
- The main purpose of this class is to point you to the crucial points in an Harmony 3 application and not to let you write a complete TCP application
- The pre-made template is built from the Harmony example project, tcpip_tcp_client.
- First, we will need to declare the APPID_KEY.
- We will then set the host & port of the remote connection static as we will only connect to OpenWeatherMap.
- After this, we will redirect the user input from the command console to a char* buffer to be used in the application.
- Now we have all information required to build the URL from the introduction.
- When we have connected and requested the data, we need to parse the resulting JSON- string (the whole JSON containing the current weather will be in one string).
- Typically, good practice when you debug JSON-strings is to print the resulting string for you to view with your own eyes that it looks correct.
- At last, redirect the application to go back to accepting user input.

Lab Procedure

- 1. Start by closing any open projects in MPLAB X IDE.
- 2. Open a new project and choose lab3 -> Firmware -> sam_e70_xult_freertos.X.
- 3. Open the file app.c located under source files.
- 4. Go to (CTRL+F) "TODO A". Enter the correct APPID_KEY. Either you create your own account on OpenWeatherMap or you
 - take the one written I the class.

| 37 | // ************************************ |
|----|---|
| 8 | //TODO A: Enter the correct APPID_KEY |
| 39 | <pre>static const char* APPID_KEY = "";</pre> |
| 40 | <pre>char jsonBuffer[1024];</pre> |
| 41 | <pre>char cityBuffer[128];</pre> |
| 42 | |
| 43 | // ************************************ |
| 43 | // ************************************ |

- 5. Now scroll down to "TODO B", the function APP_Initialize.
- 6. Set the application to connect to the host api.openweathermap.org and the port to 80. This is set to 80 because this call will

```
be over HTTP.
```

| 105 | | | | |
|-----|---|---|--|----|
| 106 | | | <pre>memset(jsonBuffer, 0, sizeof (jsonBuffer));</pre> | |
| 107 | | | <pre>memset(cityBuffer, 0, sizeof (cityBuffer));</pre> | |
| 108 | | | //TODO B: Set the application to connect to api.openweasthermap.org and port | 80 |
| 109 | | | <pre>appData.host = "";</pre> | |
| 110 | | | appData.port =; | |
| 111 | | | | |
| 112 | L | } | | |
| 113 | | | | |

7. Re-direct the user input from APP_URL_BUFFER to the cityBuffer array. This can be done in several ways, but one is to use the built-in C function snprintf(char* dest, size_t size, const char *format, ...). The first argument is the destination buffer (cityBuffer), the second one is the max size to be copied (128, because that is specified in the declaration) and the formatted input in this scenario is APP_URL_BUFFER. This can be found in "TODO C".

| 200 | ICPIP_DNS_RESOLI TESUIC; |
|-----|--|
| 8 | //TODO C: Re-direct the user input to cityBuffer from APP_URL_BUFFER |
| 202 | <pre>snprintf(,,);</pre> |
| 203 | <pre>SYS_CONSOLE_PRINT("cityBuffer: %s\r\n", cityBuffer);</pre> |
| 204 | |

8. Scroll down to "TODO D", the state APP_TCPIP_WAIT_FOR_CONNECTION. In this state we will wait for a connection to be established. Once established we will send a GET command with the full URL in the format specified in the introduction:

| http:// | /api.c | openv | veath | ermap | .org/o | data/2 | 2.5/ | weath | ner?c | q={C | CITY} | &APP | ID={ | <ey}.< th=""></ey}.<> |
|---------|--------|-------|-------|-------|--------|--------|------|-------|-------|------|-------|------|------|-----------------------|
| | 1.1 | | | | | | | | | | | | | |

| 277 | //TODO D: Build the full URL in pathBuffer. |
|-----|---|
| 278 | <pre>char pathBuffer[128];</pre> |
| 279 | <pre>snprintf(, 128, "data/2.5/weather?q=%s&APPID=%s", ,);</pre> |
| 280 | <pre>appData.path = pathBuffer;</pre> |
| 281 | |

9. Once the request is sent to the server, the application will go into the APP_TCIPIP_WAIT_FOR_RESPONSE state. Once the connection is closed, set the next state to be APP_STATE_JSON_PARSE_RETRIEVED_DATA.

| 298 | if (!TCPIP_TCP_IsConnected(appData.socket)) { |
|------|--|
| 299 | SYS_CONSOLE_MESSAGE("\r\nConnection Closed\r\n"); |
| 300 | //TODO E: Set the next state to be APP_STATE_JSON_PARSE_RETRIEVED_DATA |
| 301 | appData.state = ; |
| 302 | break; |
| 1000 | |

10. Now go down in the state APP_STATE_JSON_PARSE_RETRIEVED_DATA. One of the first things we want to do after we have sorted out the JSON-part of the retrieved data is to print the raw JSON-string. This helps us debug & analyse.

| 313 | <pre>char* resultingJson;</pre> |
|-----|--|
| 314 | char* pos; |
| 315 | |
| 316 | <pre>pos = strstr(jsonBuffer, "{\"");</pre> |
| 317 | <pre>*(&resultingJson) = pos;</pre> |
| 318 | |
| 319 | //TODO F: Print the resultingJson string |
| 320 | <pre>SYS_CONSOLE_PRINT("resultingJson: \r\n %s \r\n",);</pre> |

11. In a real application, we would need to first know the format of the JSON message in order to be able to parse it correctly. To make this lab more efficiently, we will do this backwards. If you look at this example piece of API response from OpenWeatherMap found in the introduction section to this lab. Looking at the format from the API, we need to calculate in what position the value of humidity start. The function strstr will cut the resulting Json string at the first occurrence of "humidity". A hint is to look at the other blocks where you parse the temperature, pressure and main weather.

| 322 | //Find Humidity | |
|-----|--|---|
| 323 | <pre>char* mainHumidityJson;</pre> | |
| 324 | <pre>char* mainHumidtyBuffer;</pre> | |
| 325 | | |
| 326 | //TODO G: Find the correct number of positions to move to the right after humidity | y |
| 327 | <pre>pos = strstr(resultingJson, "humidity");</pre> | |
| 328 | <pre>*(&mainHumidityJson) = pos + ;</pre> | |
| 329 | <pre>mainHumidtyBuffer = strtok(mainHumidityJson, ",");</pre> | |
| 330 | | |

12. Once the parsing is done, we wish to print the values of the main weather, pressure, temperature and humidity.



13. Now to complete the loop, we want to go back to the APP_TCPIP_WAITING_FOR_COMMAND state once the JSON-

parsing and printing is done.

| > | 359 | | | | | _ | , | | , - | | , | | | | | | |
|-------|------------|--|----|--------|----|---------|-------|-------|-------|---------|-----|---------|-------|----|----------|-------------|-----------|
| | 360 | | | //TODO | I: | Go back | to th | e APP | TCPIE | WAITING | FOR | COMMAND | state | to | continue | application | operation |
| d pre | 361 362 | | ŀ. | | | | | | | | | | | | | | |

- 14. When the build process was correct, then program the E70
- 15. Open the Terminal program, press the Reset Button
- 16. When ">" is prompter you can put in "requestWeather <city>"

Or the short form: "rw <city>"

For ex. "rw Phoenix"



17. At this point you reached the end of the class and can walk outside to check if the current weather report was correct

MPLAB® Harmony TCP/IP Stack

TCP Module API Function List

Socket Management Functions

| Opens a TCP socket as a server. |
|--|
| Opens a TCP socket as a client. |
| Disconnects an open socket and destroys the socket handle, releasing the associated resources. |
| Connects a client socket. |
| Binds a socket to a local address. |
| Binds a socket to a remote address. |
| Determines if a socket has an established connection. |
| Self-clearing semaphore indicating socket reset. |
| Disconnects an open socket. |
| Aborts a connection. |
| Allows getting the options for a socket like: current RX/TX buffer size, etc. |
| Allows setting options to a socket like adjust RX/TX buffer size, etc. |
| Obtains information about a currently open socket. |
| Gets the current network interface of an TCP socket. |
| Sets the interface for an TCP socket |
| Deregisters a previously registered TCP socket signal handler. |
| Registers a TCP socket signal handler. |
| Standard TCP/IP stack module task function. |
| |

Transmit Data Functions

| TCPIP_TCP_Put | Writes a single byte to a TCP socket. |
|-------------------------|---|
| TCPIP_TCP_PutIsReady | Determines how much free space is available in the TCP TX buffer. |
| TCPIP_TCP_StringPut | Writes a null-terminated string to a TCP socket. |
| TCPIP_TCP_ArrayPut | Writes an array from a buffer to a TCP socket. |
| TCPIP_TCP_Flush | Immediately transmits all pending TX data. |
| TCPIP_TCP_FifoTxFullGet | Determines how many bytes are pending in the TCP TX FIFO. |
| TCPIP_TCP_FifoTxFreeGet | Determines how many bytes are free and could be written in the TCP TX FIFO. |

Receive Data Transfer Functions

| TCPIP_TCP_ArrayFind | Searches for a string in the TCP RX buffer. |
|--------------------------|---|
| TCPIP_TCP_Find | Searches for a byte in the TCP RX buffer. |
| TCPIP_TCP_Get | Retrieves a single byte to a TCP socket. |
| TCPIP_TCP_Peek | Peaks at one byte in the TCP RX buffer/FIFO without removing it from the buffer. |
| TCPIP_TCP_Discard | Discards any pending data in the RCP RX FIFO. |
| TCPIP_TCP_FifoRxFreeGet | Determines how many bytes are free in the RX buffer/FIFO. |
| TCPIP_TCP_FifoSizeAdjust | Adjusts the relative sizes of the RX and TX buffers. |
| TCPIP_TCP_FifoRxFullGet | Determines how many bytes are pending in the RX buffer/FIFO. |
| TCPIP_TCP_GetIsReady | Determines how many bytes can be read from the TCP RX buffer. |
| TCPIP_TCP_ArrayGet | Reads an array of data bytes from a TCP socket's RX buffer/FIFO. |
| TCPIP_TCP_ArrayPeek | Reads a specified number of data bytes from the TCP RX buffer/FIFO without removing them from the buffer. |

UDP Module API Function List

Socket Management Functions

| TCPIP_UDP_ServerOpen | Opens a UDP socket as a server. |
|-----------------------------------|---|
| TCPIP_UDP_ClientOpen | Opens a UDP socket as a client. |
| TCPIP_UDP_IsOpened | Determines if a socket was opened. |
| TCPIP_UDP_IsConnected | Determines if a socket has an established connection. |
| TCPIP_UDP_Bind | Bind a socket to a local address and port. This function is meant for client sockets. It assigns a specific source address and port for a socket. |
| TCPIP_UDP_RemoteBind | Bind a socket to a remote address This function is meant for server sockets. |
| TCPIP_UDP_Close | Closes a UDP socket and frees the handle. |
| TCPIP_UDP_OptionsGet | Allows getting the options for a socket such as current RX/TX buffer size, etc. |
| TCPIP_UDP_OptionsSet | Allows setting options to a socket like adjust RX/TX buffer size, etc |
| TCPIP_UDP_SocketInfoGet | Returns information about a selected UDP socket. |
| TCPIP_UDP_SocketNetGet | Gets the network interface of an UDP socket |
| TCPIP_UDP_SocketNetSet | Sets the network interface for an UDP socket |
| TCPIP_UDP_TxOffsetSet | Moves the pointer within the TX buffer. |
| TCPIP_UDP_SourceIPAddressSet | Sets the source IP address of a socket |
| TCPIP_UDP_BcastIPV4AddressSet | Sets the broadcast IP address of a socket Allows an UDP socket to send broadcasts. |
| TCPIP_UDP_DestinationIPAddressSet | Sets the destination IP address of a socket |
| TCPIP_UDP_DestinationPortSet | Sets the destination port of a socket |
| TCPIP_UDP_Disconnect | Disconnects a UDP socket and re-initializes it. |
| TCPIP_UDP_SignalHandlerDeregister | Deregisters a previously registered UDP socket signal handler. |
| TCPIP_UDP_SignalHandlerRegister | Registers a UDP socket signal handler. |
| TCPIP_UDP_Task Standard | TCP/IP stack module task function. |
| Transmit Data Functions | |
| TCPIP_UDP_PutIsReady | Determines how many bytes can be written to the UDP socket. |
| TCPIP_UDP_TxPutIsReady | Determines how many bytes can be written to the UDP socket. |
| TCPIP_UDP_ArrayPut | Writes an array of bytes to the UDP socket. |
| TCPIP_UDP_StringPut | Writes a null-terminated string to the UDP socket. |
| TCPIP_UDP_Put | Writes a byte to the UDP socket. |
| TCPIP_UDP_TxCountGet | Returns the amount of bytes written into the UDP socket. |
| TCPIP_UDP_Flush | Transmits all pending data in a UDP socket. |
| Receive Data Transfer Functions | |

| NECEIVE | Data | ITALISICI | unctions | |
|---------|------|-----------|----------|--|
| | | | | |
| | | | | |

| TCPIP_UDP_GetIsReady | Determines how many bytes can be read from the UDP socket. |
|-----------------------|--|
| TCPIP_UDP_ArrayGet | Reads an array of bytes from the UDP socket. |
| TCPIP_UDP_Get | Reads a byte from the UDP socket. |
| TCPIP_UDP_RxOffsetSet | Moves the read pointer within the socket RX buffer. |
| TCPIP_UDP_Discard | Discards any remaining RX data from a UDP socket. |

Harmony TCP/IP API Subset For all Lab's

TCP Socket Management Functions

TCPIP_TCP_ArrayGet Function

This function reads an array of data bytes from a TCP socket's RX buffer/FIFO. The data is removed from the FIFO in the process.

Function Prototype

```
uint16_t TCPIP_TCP_ArrayGet(
    TCP_SOCKET_hTCP,
    uint8_t* buffer,
    uint16_t len
);
```

Preconditions

TCP is initialized.

Parameters

| Parameter | Description |
|-----------|---|
| hTCP | The socket from which data is to be read. |
| buffer | Pointer to the array to store data that was read. |
| len | Number of bytes to be read. |

Returns

| Туре | Description |
|----------|---|
| uint16_t | The number of bytes read from the socket. If less than 1en , the RX FIFO |
| | buffer became empty or the socket is not connected. |

TCPIP_TCP_ClientOpen Function

Provides a unified method for opening TCP client sockets. Sockets are created at the TCP module initialization, and can be claimed with this function and freed using **TCPIP_TCP_Abort** or **TCPIP_TCP_Close**. If the remoteAddress != 0 (and the address pointed by remoteAddress != 0) then the socket will immediately initiate a connection to the remote host.

Function Prototoype

Preconditions

TCP is initialized.

Parameters

| Parameter | Description |
|---------------|---|
| addType | The type of address being used. Valid values are: IP_ADDRESS_TYPE_IPV4 or |
| | IP_ADDRESS_TYPE_IPV6 |
| remotePort | TCP port to connect to. The local port for client sockets will be automatically |
| | picked by the TCP module. |
| remoteAddress | The remote address to be used |

Returns

| Туре | Description |
|------------|---|
| TCP_SOCKET | Handle - Save this handle and use it when calling all other TCP APIs. If no |
| | sockets of the specified type were available to be opened, the handle will |
| | contain a value equal to INVALID_SOCKET. |

TCPIP_TCP_Close Function

Graceful Option Set: If the graceful option is set for the socket (default), a TCPIP_TCP_Disconnect will be tried. If the linger option is set (default) the TCPIP_TCP_Disconnect will try to send any queued TX data before issuing FIN. If the FIN send operation fails or the socket is not connected the abort is generated.

Graceful Option Not Set: If the graceful option is not set, or the previous step could not send the FIN, a TCPIP_TCP_Abort is called, sending a RST to the remote node. Communication is closed, the socket is no longer valid and the associated resources are freed.

Function Prototype

```
void TCPIP_TCP_Close(
        <u>TCP_SOCKET</u> hTCP
);
```

Preconditions

TCP socket should have been opened with TCPIP_TCP_ServerOpen/TCPIP_TCP_ClientOpen.

hTCP - valid socket

Void

Parameters

| | Parameter | Description |
|--------|-----------|---|
| | hTCP | Handle to the socket to disconnect and close. |
| Return | 15 | |
| | Туре | Description |

TCPIP_TCP_GetIsReady Function

Call this function to determine how many bytes can be read from the TCP RX buffer. If this function returns zero, the application must return to the main stack loop before continuing in order to wait for more data to arrive.

None

Function Prototype

Preconditions

TCP is initialized.

Parameters

| Parameter | Description |
|-----------|----------------------|
| hTCP | The socket to check. |

Returns

| Туре | Description |
|----------|--|
| uint16_t | The number of bytes available to be read from the TCP RX buffer. |

TCPIP_TCP_IsConnected Function

This function determines if a socket has an established connection to a remote node. Call this function after calling

TCPIP_TCP_ServerOpen()/TCPIP_TCP_ClientOpen() to determine when the connection is set up and ready for use.

Function Prototype

| <pre>bool TCPIP_TCP_IsConnected(</pre> | | |
|--|--|--|
| TCP SOCKET hTCP | | |
|); | | |

Preconditions

TCP is initialized.

Parameters

| Parameter | Description |
|-----------|--------------------------|
| hTCP | The TCP socket to check. |

Returns

| Туре | Description |
|------|-----------------------------------|
| bool | True: the socket is connected |
| | False: the socket is disconnected |

TCPIP_TCP_PutIsReady Function

Call this function to determine how many bytes can be written to the TCP TX buffer. If this function returns zero, the application must return to the main stack loop before continuing in order to transmit more data.

Function Prototype

Preconditions

TCP is initialized.

Parameters

| Parameter | Description |
|-----------|--|
| hTCP | The socket from which data is to be written. |
| ns | |

Returns

| Туре | Description |
|----------|---|
| uint16_t | The number of bytes available to be written in the TCP TX buffer. |

TCPIP_TCP_StringPut Function

This function writes a null-terminated string to a TCP socket. The null-terminator is not copied to the socket.

Function Prototype

```
const uint8_t* TCPIP_TCP_StringPut(
    TCP SOCKET hTCP,
    const uint8_t* Data
);
```

Preconditions

TCP is initialized.

Parameters

| Parameter | Description |
|----------------|--|
| hTCP | The socket from which data is to be written. |
| const uint8_t* | Data |

Returns

| Туре | Description |
|----------------|--|
| const uint8_t* | Pointer to the byte following the last byte written to the socket. If this |
| | pointer does not dereference to a NULL byte, the buffer became full or |
| | the socket is not connected. |

TCPIP_TCP_WasReset Function

This function is a self-clearing semaphore indicating whether or not a socket has been disconnected since the previous call. This function works for all possible disconnections: a call to **TCPIP_TCP_Disconnect**, a FIN from the remote node, or an acknowledgment timeout caused by the loss of a network link. It also returns true after the first call to **TCPIP_TCP_Initialize**. Applications should use this function to reset their state machines.

Function Prototype

```
bool TCPIP_TCP_WasReset(
        TCP_SOCKET hTCP
);
```

Preconditions

TCP is initialized.

Parameters

| | Parameter | Description |
|--------|-----------|--|
| | hTCP | The TCP socket to check. |
| Return | IS | |
| | Туре | Description |
| | bool | true: the socket was disconnected since the previous call |
| | | false: the socket remained connected since the previous call |

UDP Socket Management Functions

TCPIP_UDP_ArrayGet Function

This function reads an array of bytes from the UDP socket, while adjusting the current read pointer and decrementing the remaining bytes available. TCPIP_UDP_GetIsReady should be used before calling this function to get the number of the available bytes in the socket.

Function Prototype

```
uint16_t TCPIP_UDP_ArrayGet(
    <u>UDP_SOCKET</u> hUDP,
    uint8_t * cData,
    uint16_t wDataLen
);
```

Preconditions

UDP socket should have been opened with TCPIP_UDP_ServerOpen/TCPIP_UDP_ClientOpen.

hUDP - valid socket

Parameters

| Parameter | Description |
|-----------|---|
| hUDP | UDP Socket Handle |
| cData | The buffer to receive the bytes being read. If NULL, the bytes are simply |
| | discarded |
| wDataLen | Number of bytes to be read from the socket. |
| | |

Returns

| Туре | Description |
|-----------------|---|
| <u>uint16 t</u> | The number of bytes successfully read from the UDP buffer. If this value is |
| | less than wDataLen, then the buffer was emptied and no more data is |
| | available. |

TCPIP_UDP_Close Function

Closes a UDP socket and frees the handle. Call this function to release a socket and return it to the pool for use by future communications.

Function Prototoype

```
void TCPIP_UDP_Close(
        UDP_SOCKET_hUDP
);
```

Preconditions

UDP socket should have been opened with TCPIP_UDP_ServerOpen/TCPIP_UDP_ClientOpen.

hUDP - valid socket

Parameters

| | Parameter | Description |
|--------|-----------|-------------------|
| | hUDP | UDP Socket Handle |
| Return | IS | |

| Туре | Description |
|------|-------------|
| void | None |

TCPIP_UDP_GetIsReady Function

This function will return the number of bytes that are available in the specified UDP socket RX buffer. The UDP socket queues incoming RX packets in an internal queue. If currently there is no RX packet processed (as a result of retrieving all available bytes with TCPIP_UDP_ArrayGet, for example), this call will advance the RX packet to be processed to the next queued packet. If a RX packet is currently processed, the call will return the number of bytes left to be read from this packet.

Function Prototype

```
uint16_t TCPIP_UDP_GetIsReady(
     <u>UDP_SOCKET</u> hUDP
);
```

Preconditions

UDP socket should have been opened with TCPIP_UDP_ServerOpen/TCPIP_UDP_ClientOpen.

hUDP parameter is a valid socket

Parameters

| | Parameter | Description |
|--------|-----------|-------------------|
| | hUDP | UDP Socket Handle |
| Return | 15 | |
| | Туре | Description |

| уре | Description |
|-----------------|---|
| <u>iint16 t</u> | The number of bytes that can be read from the socket. |

TCPIP_UDP_ServerOpen Function

Provides a unified method for opening UDP server sockets.

Function Prototype

```
UDP SOCKET TCPIP_UDP_ServerOpen(
    IP ADDRESS TYPE addType,
    UDP PORT localPort,
    IP MULTI ADDRESS* localAddress
);
```

Preconditions

UDP is initialized.

Parameters

| Parameter | Description |
|---|---|
| IP_ADDRESS_TYPE addType | The type of address being used. |
| | IP_ADDRESS_TYPE_IPV4 or IP_ADDRESS_TYPE_IPV6. |
| <u>UDP PORT</u> localPort | UDP port on which to listen for connections |
| <pre>IP MULTI ADDRESS* localAddress</pre> | Local IP address to use. Can be 0 (NULL) if any incoming interface will do. |

Returns

| Туре | Description |
|------------|---|
| UDP SOCKET | Handle - Save this handle and use it when calling all other UDP APIs. If no |
| | sockets of the specified type were available to be opened, the handle will |
| | contain a value equal to INVALID_SOCKET. |

TCPIP_UDP_SocketInfoGet Function

This function will fill a user passed UDP_SOCKET_INFO structure with status of the selected socket

Function Prototype

```
bool TCPIP_UDP_SocketInfoGet(
    <u>UDP_SOCKET_hUDP,
    UDP_SOCKET_INFO</u>* pInfo
);
```

Preconditions

UDP socket should have been opened with <u>TCPIP_UDP_ServerOpen()/TCPIP_UDP_ClientOpen()()</u>.

hUDP - valid socket

pInfo - valid address of a UDP SOCKET INFO structure

Parameters

| Parameter | Description |
|-----------|--|
| hUDP | UDP Socket Handle |
| pInfo | Pointer to UDP SOCKET INFO to receive socket information |

Returns

| Туре | Description |
|------|---|
| bool | true if call succeeded |
| | false if no such socket or invalid pinfo. |

UDP_SOCKET_INFO Structure

Holds information about a UDP Socket

Structure

```
typedef struct {
    IP_ADDRESS_TYPE addressType;
    IP_MULTI_ADDRESS_remoteIPaddress;
    IP_MULTI_ADDRESS_localIPaddress;
    IP_MULTI_ADDRESS_sourceIPaddress;
    IP_MULTI_ADDRESS_destIPaddress;
    UDP_PORT_remotePort;
    UDP_PORT_localPort;
    TCPIP_NET_HANDLE_hNet;
} UDP_SOCKET_INFO;
```

Members

| Туре | Member Name | Description |
|------------------|-----------------|---|
| IP_ADDRESS_TYPE | addressType | address type of the socket |
| IP_MULTI_ADDRESS | remoteIPaddress | current socket destination address |
| IP_MULTI_ADDRESS | localIPaddress | current socket source address |
| IP_MULTI_ADDRESS | sourceIPaddress | source address of the last packet |
| IP_MULTI_ADDRESS | destIPaddress | destination address of the last packet |
| UDP_PORT | remotePort | Port number associated with remote node |
| UDP_PORT | localPort | local port number |
| TCPIP_NET_HANDLE | hNet | associated interface |



