Integrating FreeModbus TCP/IP in to a uEZ Application

Covers the following products:

uEZ

Future Designs, Inc.
Your Development Partner
996 A Cleaner Way, Huntsville, AL 35805

1. Introduction

This document outlines the steps required to integrate FreeModbus TCP/IP into an existing application. For additional information about FreeModbus refer to their website, http://www.freemodbus.org/.

This document assumes some knowledge of Modbus, C programming, and familiarity with the compiler being used. Examples are provided using the current NXP supported compilers, IAR, CrossWorks, and Keil.

2. Getting Started

Provided with uEZ® as of v2.07 are the ported FreeModbBus source files, LPC1788 and LPC4088 port files, example Ruby Scripts, and a demo task to handle Master TCP/IP communications.

This document assumes that there is an existing application the files will be integrated into, if a project is not available use the uEZ Project Maker to create a template project for the selected hardware and compiler.

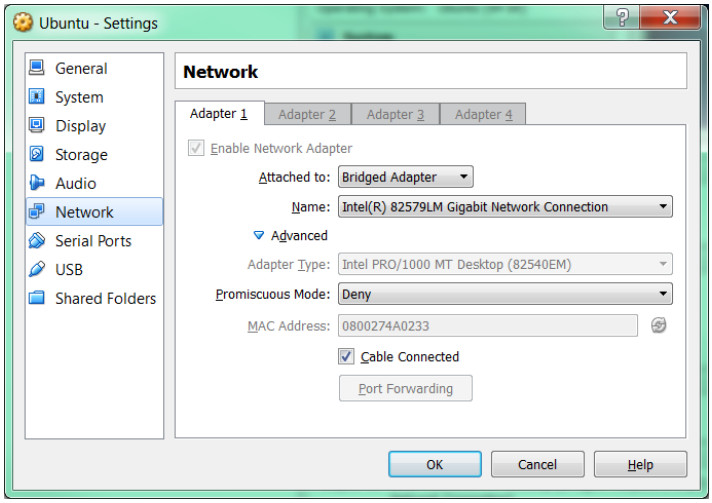
2.1 Setting up a Virtual Machine

Several options exist to run Virtual Machines, assuming you do not have access to a Linux machine or some other way to run Ruby Script files, two known working examples would be VMWare or Virtual Box, see links below.

http://www.vmware.com/
https://www.virtualbox.org/

2.1.1 Virtual Machine Network Settings

By default the uEZGUIs are set to use a static IP address so the Virtual Machine will need to be set with a static IP address on the same network. For VirtualBox the following settings need to be changed before opening the Virtual Machine.



In this example Intel(R) 82579LM Gigabit Network Connections refers to the device that is connected to the same physical network connection of the uEZ GUI.

2.1.2 Setting the IP Address in Linux

The simplest way to set the IP address is to use the command line. In Ubuntu the following command can be run.

```
sudo ifconfig eth0 192.168.10.21
```

You can then confirm that the VM can communicate with the uEZ GUI by pinging it.

```
ping 192.168.10.20
```

2.2 Linux Setup

Testing was done using Virtual Box and Ubuntu 14.04 LTS. The steps below show how to setup the machine to run the script files, these commands are run from the terminal. These instructions assume that none of the following programs have been previously installed on the virtual box.

1. *sudo apt-get install curl*
2. *curl –sSL https://rvm.io/mpapis.asc | gpg --import*
3. *\curl -sSL https://get.rvm.io | bash -s stable --ruby*
4. *source ~/.rvm/scripts/rvm*
5. Find the current list of Rubies
    a. *rvm list known*
6. Select the newly installed ruby
    a. *rvm use 2.2*
7. Verify the correct version
    a. *ruby –v*
8. Install rmodmus gem
    a. *gem install rmodbus*

2.3 Configuring Scripts

Copy the script files form “uEZ/Source/Library/Modbus/freemodbus/Client Scripts” to the machine that will be running them. Next configure the mbc_config.rb file with the correct IP Address and Port for the target hardware, this file will need to be updated and rerun when the IP Address of the target changes.

```
#! /usr/bin/env ruby

# this file allows you to update the config file programmatically
# modify and run
# ./mbc_config.rb

require 'yaml'

File.write("config.yml", ['192.168.1.168', 502].to_yaml)
```

After the file has been updated run the command "./mbc_config.rb" from the command line.

NOTE: Script files are provided as reference and initial testing purposes only.

2.4 Integrating Files in the Project

The first step is to take the Demo Application files, located in "uEZ\Source\Library\Modbus\freemodbus\Demo Application" and copy them to your application folder. Typically we recommend adding these files to the same folder as main.c. Next modify NetworkStartup.c to add the include file for the Modbus task and the function call to start the task.

Includes section:

```
#include "ModbusTCPIPTask.h"
```

Function Call from NetworkStartupTask ()

```
// Start Modbus TCPIP
ModbusTCPIPTask_Start();
```

NOTE: the ModbusTCPIPTask_Start() assumes the network has already been initialized.

Then just add the files to the current project.

2.5 Adding FreeModbus Files

The files that need to be added are all located in the directory "uEZ\Source\Library\Modbus\", this is the assumed base directory for the files listed below.

- freemodbus/modbus
  - mb.c
- freemodbus/modbus/functions
  - mbfunccoils.c
  - mbfuncdiag.c
  - mbfuncdisc.c
  - mbfuncholding.c
  - mbfuncinput.c
  - mbfuncother.c
  - mbutils.c
- freemodbus/modbus/include
  - mb.h
  - mbconfig.h
  - mbframe.h
  - mbfunc.h
  - mbport.h
  - mbproto.h
  - mbutils.h
- freemodbus/modbus/tcp

  - mbtcp.c
  - mbtcp.h
- freemodbus/demo/FDI1788/port
  - port.h
  - portevent.c
  - portother.c
  - porttcp.c

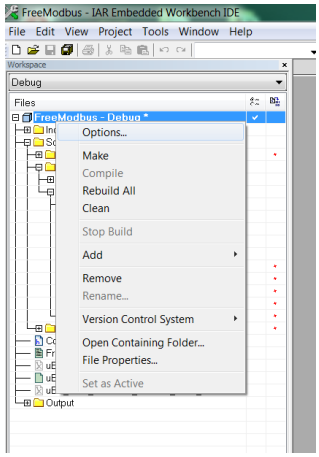
2.6 Include Directories

Since each compiler is different the supported compilers will be explained separately.

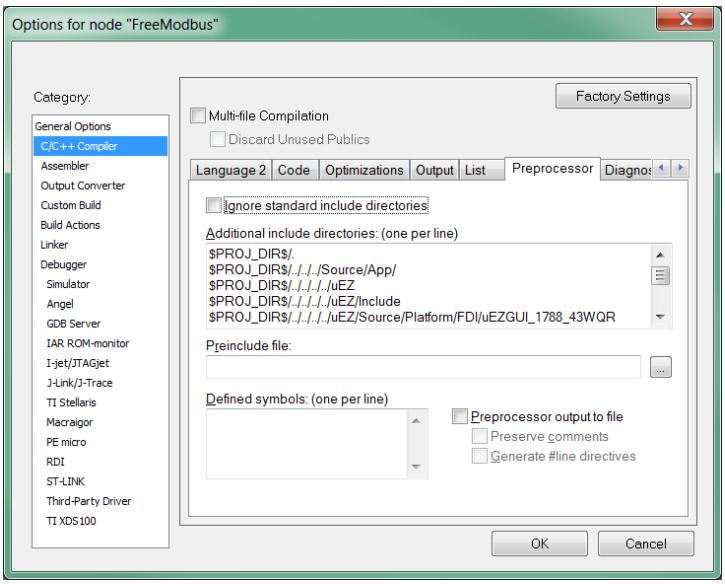
2.6.1 IAR

To add the include directories for IAR follow the steps below.

1. Right click on the project in the workspace.



2. Select options.
3. In the options dialog select C/C++ Compiler then select the Preprocessor tab.



4. Scroll to the bottom of the Additional include directories and add these lines. NOTE the paths are relative to the project file location.

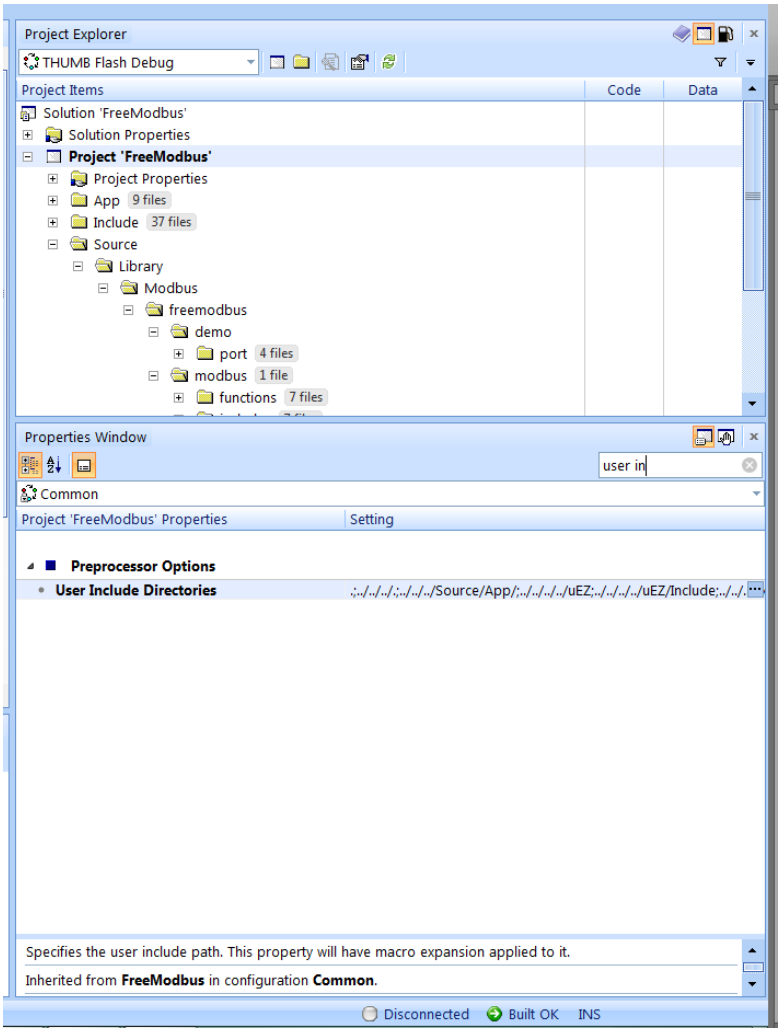
$PROJ_DIR$/../../../../uEZ/Source/Library/Modbus/freemodbus/modbus/include
$PROJ_DIR$/../../../../uEZ/Source/Library/Modbus/freemodbus/modbus/tcp
$PROJ_DIR$/../../../../uEZ/Source/Library/Modbus/freemodbus/demo/FDI1788/port

5. Repeat steps 1 – 4 for each project configuration.

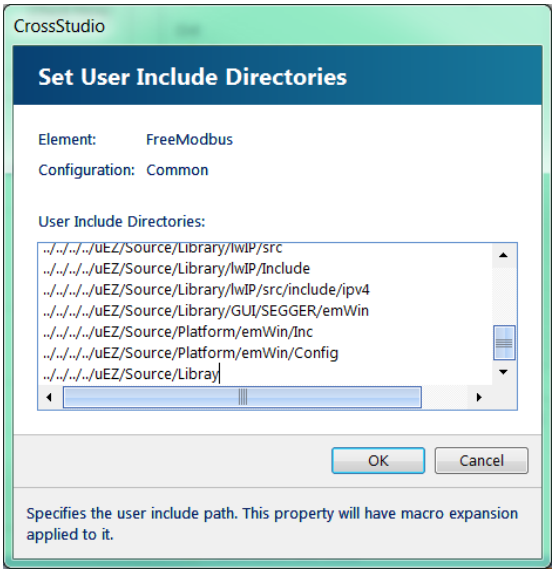
2.6.2 CrossWorks

To add the include directories for CrossWorks follow the steps below.

1. Select the Project in the Project Explorer Window and do a search for “user in” to find the User Include Directory Settings. Make sure the configuration is set for Common.



2. Double Click on the setting to open the dialog.



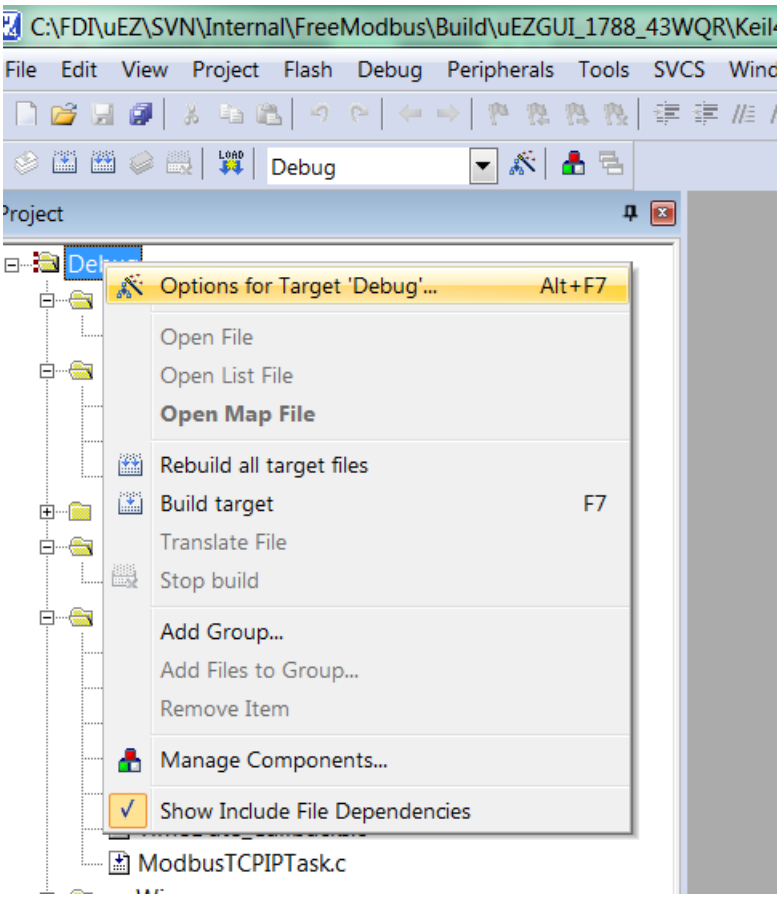
3. Add the following entries.

   ../../../../uEZ/Source/Library/Modbus/freemodbus/modbus/include
   ../../../../uEZ/Source/Library/Modbus/freemodbus/modbus/tcp
   ../../../../uEZ/Source/Library/Modbus/freemodbus/demo/FDI1788/port

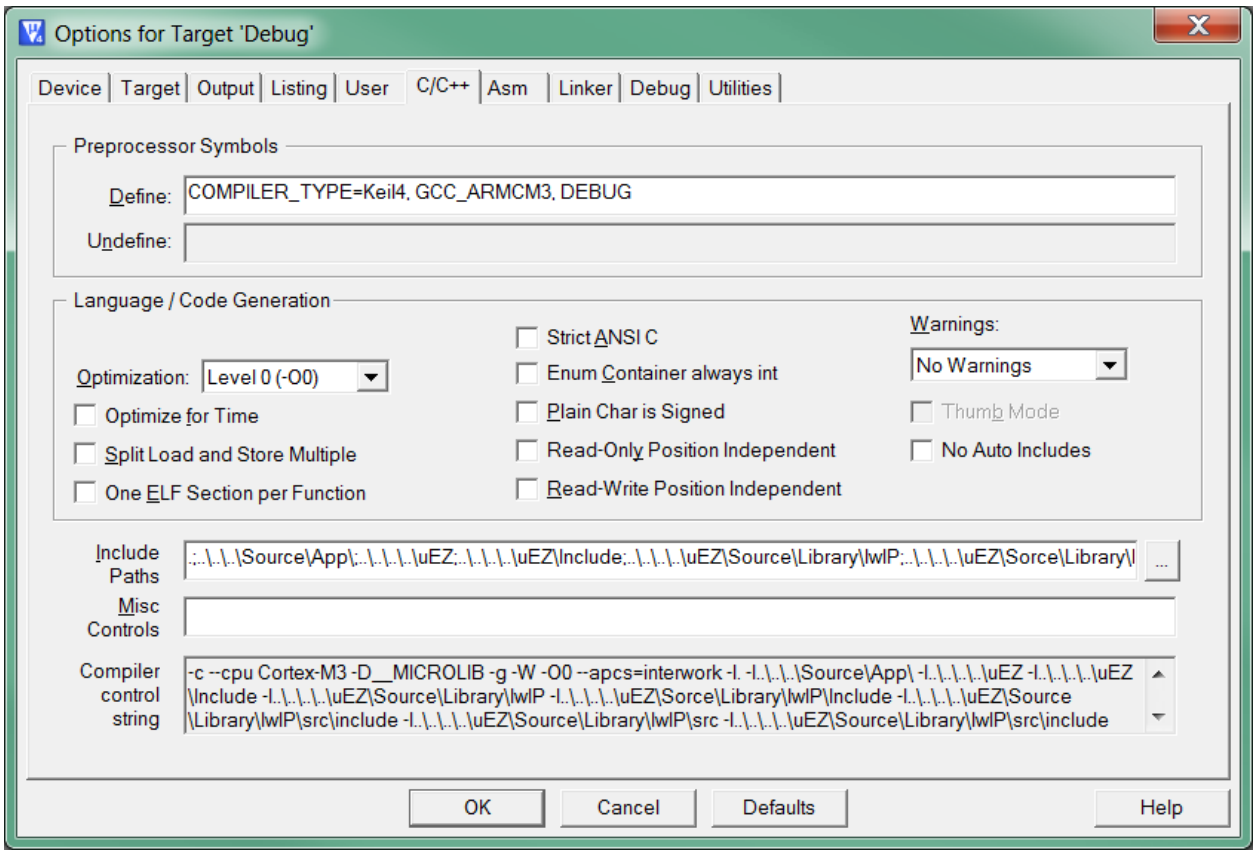
2.6.3 Keil

To add the include directories for Keil follow the steps below.

1. Select the project confirmation in the Project window and select options.



2. Select the C/C++ tab and then select the "…" button next to the include paths.



3. Add the following entries.

../../../../uEZ/Source/Library/Modbus/freemodbus/modbus/include
../../../../uEZ/Source/Library/Modbus/freemodbus/modbus/tcp
../../../../uEZ/Source/Library/Modbus/freemodbus/demo/FDI1788/port