i2cscan

Version 1.0.0 – 16 June 2016
Kit Bishop

Document History		
Version	**Date**	**Change Details**
Version 1.0.0	16 June 2016	First released version

Contents

1. Overview ____ *2*

2. Files Supplied ____ *2*

3. Usage and Installation ____ *2*

3.1. Using i2cscan statically linked program ____ 2

3.2. Using and Installing i2cscan dynamically linked program ____ 3

4. Using Makefile ____ *3*

4.1. Modify Makefile ____ 3

4.2. Makefile targets ____ 3

5. Running the i2cscan Program ____ *4*

6. Further Development ____ *5*

1. Overview

i2cscan is C++ program that scans the I2C bus on the Omega and reports on I2C devices found.

Notes:

- The **i2cscan** library uses I2C to access the I2C bus via the **libnewi2c** library that is available and documented at https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/libnewi2c

i2cscan consists of single program in static and dynamic forms that scans the I2C bus for devices.

These program is described in more detail in this document.

The program was developed on a KUbuntu-14.04 system running in a VirtualBox VM and uses the OpenWrt toolchain for building the code:

The toolchain used can be found at:

- https://s3-us-west-2.amazonaws.com/onion-cdn/community/openwrt/OpenWrt-Toolchain-ar71xx-generic_gcc-4.8-linaro_uClibc-0.9.33.2.Linux-x86_64.tar.bz2

and details of its setup and usage can be found at:

- https://community.onion.io/topic/9/how-to-install-gcc/22

i2cscan comes with **NO GUARANTEES** ☺ but you are free to use it and do what you want with it.

2. Files Supplied

i2cscan is supplied in files in a GitHub repository at https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/i2cscan. This repository contains the following important directories and files:

- **i2cscan.pdf** – this documentation as a PDF file
- **Makefile** – the Makefile for **i2cscan** library
- **src** – directory containing source (***.cpp**) files for **i2cscan** library
- **bin** – directory containing the built program code:
    - **dynamic/i2cscan** – the dynamically linked version of the program
    - **static/i2cscan** – the statically linked version of the program

3. Usage and Installation

Installing and using the program is simple. It primarily consists of linking the program and if needed (see below) the i2clibrary code.

3.1. Using i2cscan statically linked program

To use **i2cscan** statically linked program you simply need to copy the program to the Omega and run it.

3.2. Using and Installing i2cscan dynamically linked program

To use **i2cscan** dynamically linked program you need to copy it and the **libnewi2c** library to your Omega and then run the program.

Since **i2cscan** dynamically linked program makes use of **libnewi2c** library for the I2C communication, you will also need to dynamically link to copy **libnewi2c.so** to the **/lib** directory on your Omega

Alternatively, you can copy the library to any location that may be set up in any **LD_LIBRARY_PATH** directory on your Omega. For example, I use the following for testing:

- Created directory **/root/lib**
- Copied the libraries to **/root/lib**
- Added the following lines to my **/etc/profile** file:

```
LD_LIBRARY_PATH=/root/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

4. Using Makefile

A **Makefile** is supplied that can be used to build the library.

4.1. Modify Makefile

The **Makefile** will need modifying:

- You **NEED** to and **MUST** change **TOOL_BIN_DIR** to the "bin" directory of your OpenWrt uClibc toolchain. E.G. make appropriate change to **<xxxx>** in:

  TOOL_BIN_DIR=<xxxx>/OpenWrt-Toolchain-ar71xx-generic_gcc-4.8-linaro_uClibc-0.9.33.2.Linux-x86_64/toolchain-mips_34kc_gcc-4.8-linaro_uClibc-0.9.33.2/bin

- You **MAY** need to change **LIBNEWI2C_DIR** to relative directory of libnewi2c if you are not using the sources as originally supplied.
  The default if using the standard **source** directory structure as supplied is:

  LIBNEW-GPIO_DIR=../libnewi2c

4.2. Makefile targets

The **Makefile** implements the following set of targets:

- **make**
  The default target. Performs a complete build of both static and dynamic link versions of the program.
  This is directly equivalent to:
  make static dynamic

- **make static**
  Performs a complete build of just the static link version of the program.

- **make dynamic**
  Performs a complete build of just the dynamic link version of the program.

- **make clean**
  Removes all previous build files, both static and dynamic link versions.
  This is directly equivalent to:
  make clean-static clean-dynamic

- **make clean-static**
  Removes all previous build files for static link versions only
  .
- **make clean-dynamic**
  Removes all previous build files for dynamic link versions only

If the following is added to the **make** command line:

builddep=1

then **libnewi2c** library that the program depends on will also be built before building the program.

5. Running the i2cscan Program

The i2cscan program is simply run by using the following command in the directory where the program is placed:

```
./i2cscan
```

The program takes no parameters.

The program scans all I2C addresses from 0x03 to 0x7f and for each reports:

- r – read access OK
- w – write access OK
- q – quick probe access Ok

The following is an example of the output when I2C devices are found with full access at addresses **0x08**, **0x10**, **0x17**, **0x20**, **0x60** and **0x70** and with read only access at address **0x7e**:

```
Scanning I2C
Result:RWQ where R=r=read; W=w=write; Q=q=quick probe; .=no valid response
    0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f
    --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
00:             ... ... ... ... ... rwq ... ... ... ... ... ... ...
10: rwq ... ... ... ... ... ... rwq ... ... ... ... ... ... ... ...
20: rwq ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
30: ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
40: ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
50: ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
```

```
60: rwq ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
70: rwq ... ... ... ... ... ... ... ... ... ... ... ... ... r.q ...
```

6. Further Development

Development of **i2cscan** is on-going. There will be changes and additions to the code in the future.