

Introduction to the 16-bit Tower Boards Using the MC9S12GN32

by: **Luis Olea**
Mexico

1 Introduction

This application note is a comprehensive guideline on how to get started using Tower boards, information on its available expansion peripherals, and how to start rapid prototyping with Freescale 16-bit microcontrollers.

The Tower controller modules are demonstration boards and kits that can be used for low cost microcontroller development and evaluation. The tower boards are designed to interface with the Freescale Tower System, a modular development platform which aids in rapid prototyping and tool-reuse. An integrated open-source background debug module, software tools, and examples provided with the tower boards make application development and debug quick and easy. The TWR-S12GN32 board features the MC9S12GN32, a 16-bit MCU focused on low-cost, high-performance, and a low pin-count. The MC9S12G family is intended to bridge the gap between high-end 8-bit MCUs and high-performance 16-bit MCUs. The MC9S12G family is targeted to generic automotive applications requiring CAN or LIN/J2602 communications. This guide shows how to quickly connect the board to a host PC, and execute a demonstration application for getting this device up and running.

After reading this application note you will be familiar with the 16-bit tower boards available, have a clear understanding of how, when, and why the tower boards are used, and which expansion modules are available for extending the prototyping

Contents

1	Introduction.....	1
2	Getting Started with Tower Boards.....	2
2.1	Tower board concepts.....	2
2.2	Use of Freescale Tower boards.....	2
2.3	Available peripheral modules.....	2
2.4	Available controller modules.....	3
3	Using the TWR-S12GN32 board.....	3
3.1	Creating a project in CodeWarrior.....	3
3.2	Conclusion.....	4
4	Troubleshooting.....	4
5	Reference Material.....	5
6	Glossary.....	5

capabilities of a given tower board. Additionally, you are provided an example application that can be run on the S12GN32 tower board.

2 Getting Started with Tower Boards

The following sections explain some advantages of using the Tower board concepts. A step by step example is provided in the use of the S12GN32 Tower board.

2.1 Tower board concepts

The following concepts are introduced in the Freescale Tower System for easy reference.

- **Controller module** — A stand alone expandible evaluation board that includes the microcontroller, an onboard debugger, and provides some leds, buttons and a potentiometer for interacting with the microcontroller.
- **Elevator modules** — These two boards plug into the two expansion buses available in the tower boards to route MCU signals to peripheral modules and easily accessible headers. Elevator modules allow up to one controller module and three peripheral modules to be connected to the tower expansion bus. Additionally, one side mounting board can also be plugged in (that is, TWR-LCD).
- **Peripheral module** — A hardware expansion board pluggable into the elevator boards that provides special functionality (that is ethernet, sensors).
- **Complete kit** — A kit includes a controller module, elevator modules, and one or more peripheral modules suited to a special application. The kit can always be reconfigured, expanded, or diminished to suit specific development needs.

2.2 Use of Freescale Tower boards

The Freescale Tower system allows you to explore and evaluate different microcontrollers and peripheral hardware to quickly prototype an embedded system that fits your needs, that is without any hardware development. Software development is facilitated through our documentation.

Use Freescale Tower system for:

- Evaluating a microcontroller derivative using a controller module that can function as a stand alone system.
- Adding hardware to fit your needs using a Freescale catalog of peripheral modules is quick and easy.
- Adding custom hardware through the ports available in the elevator modules, or in the controller module itself (all MCU signals are available on the elevator bus) is possible.
- Change the MCU of your embedded system by swapping the controller module.
- Enable SW development with the examples and applications provided.
- All low cost configurable modules enable rapid prototyping and development at a low cost.

2.3 Available peripheral modules

Among the growing portfolio of peripheral modules some are:

- TWR-ADCDAC-LTC — High Precision Analog Module
- TWR-ELEV — Provide power regulation circuitry, standardized signal assignments, and act as common backplane for all assembled Tower System modules
- TWR-LCD — Graphical LCD module
- TWR-MEM — Memory module featuring serial flash, MRAM, CPLD, SD Card, and compact flash
- TWR-PROTO — Prototyping module featuring an ample perfboard area for custom circuitry
- TWR-RF-SNAP — Tower System RF Wireless Mesh Networking Module

- TWR-SENSOR-PAK — Sensor module featuring multiple plug-in sensors, including inertial, pressure, and touch sensors
- TWR-SER — Serial module featuring Ethernet, USB, RS232/485, and CAN
- TWR-SER2 — Tower System Enhanced Serial module featuring dual Ethernet and high-speed USB

For more information go to the Freescale <http://www.freescale.com/tower> website.

2.4 Available controller modules

Freescale currently provides controller modules for the S12GN32 and S12G128 16-bit microcontrollers, with more boards to be available soon. There are also 8-bit and 32 bit boards and more to come in the 16-bit range.

3 Using the TWR-S12GN32 board

The following is a quick guide for developing an application with the TWR-S12GN32 board.

You need the following pre-requisites to follow the steps to test this application.

- Install CodeWarrior Development Studio 5.1 (or later version) for S12(X) (available at www.freescale.com)
- Connect a TWR-S12GN32 board with a mini USB cable. The first time you connect it it will ask to install the drivers of the OSBDM. It should be installed automatically if CodeWarrior is already installed on the PC.

3.1 Creating a project in CodeWarrior

The following steps allow you to create a project for the S12GN32. Write a simple application, compile it and run it on the microcontroller.

To create a project targeting the S12GN32:

1. Open CodeWarrior Development Studio for S12(X) 5.1 (Start – Programs – Freescale CodeWarrior).
2. Create a new project (File – New project).
3. Select MC9S12GN32 in derivative, and Open source BDM as the default connection
4. Select a project name and location, leave language selection as default (C language)
5. Skip adding existing files to the project (click next).
6. Leave rapid application development as none (default), also leave startup code, memory model, and floating point support as defaults (ANSI startup code, small memory model, and none for floating point support are the defaults). Click next and finish. The project files are now created and you are provided with startup code and a main function with an infinite for loop.

Now that the project wizard has created the project and the main file, start coding the application:

1. Open main.c in the project tree view
2. Copy the following code to main.c inside the main function just below the “put your own code here” comment:

Initialization code:

```
/* Init gpio pins for led outputs */
#define LED1 PTT_PTT2
#define LED2 PTT_PTT3
#define LED3 PTT_PTT4
#define LED4 PTT_PTT5
PTT = 0xFF;
DDRT = 0x3C;
/* Init gpio pins for button inputs */
#define SW1 PT1AD_PT1AD4
#define SW2 PT1AD_PT1AD5
```

Troubleshooting

```
#define SW3 PT1AD_PT1AD6
#define SW4 PT1AD_PT1AD7
PER1AD = 0xF0;
ATDDIEN = 0x00F0;
/* Init timer for blinking leds */
TSCR2_PR = 0x06;
TSCR1_TEN = 1;
```

3. Copy the following code to main.c in the main function inside the infinite for loop:

LED and Switch code

```
unsigned char tmrFlag = 0;
/* SW1 turns LED1 on */
if(SW1){
    LED1 = 1;
} else {
    LED1 = 0;
}
/* SW2 turns LED2 on */
if(SW2){
    LED2 = 1;
} else {
    LED2 = 0;
}
/* Check if timer overflowed */
if(TFLG2_TOF){
    tmrFlag = 1;
}
/* SW3 blinks LED3 */
if( (!SW3) && (tmrFlag) ){
    LED3 ^= 1;
}
/* SW4 blinks LED4 */
if( (!SW4) && (tmrFlag) ){
    LED4 ^= 1;
}
/* Clear timer flag */
if(tmrFlag){
    tmrFlag=0;
    TFLG2_TOF = 1;
}
```

4. Debug the project (press F5 or go to menu Project – Debug). This will compile the project and program the target.

After the target device has been programmed the code can run using the green arrow run button, by pressing F5 or by going to menu Run – Start/Continue.

After the code is running press SW1 through SW4 to see the changes in LED1 through LED4. SW1 and SW2 turn on LED1 and LED2 respectively. SW3 and SW4 blink LED3 and LED4 after a noticeable amount of time dictated by a timer overflow.

3.2 Conclusion

This application note demonstrated how to quickly get the 16-bit Tower boards up and running with information about expanding the hardware to have a fully functional prototype for an embedded system at a low cost.

4 Troubleshooting

If the demonstration application fails to function as indicated above, please follow the steps below before contacting Freescale support.

- Ensure the option jumpers are set to the default positions, ensure the PWR_SEL option jumper is set correctly.
- Ensure the ON/OFF switch is set to ON.

- Ensure the VDD LED is on.
- Ensure the correct version (CodeWarrior for S12X) of CodeWarrior is installed and that the S12GN32 Service Pack is installed.

5 Reference Material

The following material is available at www.freescale.com.

Software development tools for 16-bit controllers

- CodeWarrior 5.1 for HCS12(X) Microcontrollers

Application notes regarding 16-bit devices

- AN3622 — Comparison of the S12XS CRG Module with S12P CPMU Module
- AN2612 — PWM Generation Using HCS12 Timer Channels
- AN2428 — An Overview of the HCS12 ATD Module
- AN2883 — Serial Communication Interface as UART on HCS12 MCUs
- AN2461 — Low Power Management using HCS12 and SBC devices

Reference Manual for the S12G 16-bit microcontrollers

- MC9S12G Family Reference Manual

6 Glossary

The following concepts are commonly used when working with tower boards:

- TWR — Tower board
- Elevator module — Boards that interconnect the different tower modules
- OSBDM — Open source background debug module that is included in the controller module for programming/debugging an application.
- Expansion module — Specialized hardware designed to be pluggable into the Freescale Tower System.

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2011 Freescale Semiconductor, Inc.