Development of an Embedded RTOS Educational Platform – Hardware Design and Development

Gang Sun,
Northern Kentucky University

Abstract

A Real-time Operating System (RTOS) is an operating system that effectively manages the hardware resources of an embedded system that requires very precise timing and high reliability. Because of the lack of time and lab facilities, most U.S. Electronic Engineering Technology (EET) programs do not usually offer the course related to embedded real-time systems development or they just emphasize concepts of RTOS and introduce basic theoretic topics in their microcontroller courses. As a result, students who have a good understanding of theory and concepts of RTOS do not have the opportunity to map their knowledge onto real-world implementations. To bridge the gap between conceptual understanding and concrete implementations, an embedded RTOS educational platform has been established for EET students in the Department of Engineering Technology & Industrial Distribution at Texas A&M University as well as in the Engineering Technology program at Northern Kentucky University. This paper only focuses on hardware design and development of the embedded RTOS platform. The laboratory curriculum development and student learning outcomes/feedback will be discussed in more detail in the follow-up manuscript.

Introduction

A Real-time Operating System (RTOS) is an operating system that effectively manages the hardware resources of an embedded system and performs more urgent tasks among various task-code subroutines at any given time. It is the key element for designing embedded products that require very precise timing and high reliability, such as cell phones, medical devices, and real-time automation systems, etc., in order to achieve more efficient use of CPU through multithreading, modularize product development and maintenance, and simplify application porting.

Due to the lack of time and lab facilities, most U.S. Electronic Engineering Technology (EET) programs do not offer any courses related to real-time operating systems or they just emphasize concepts of RTOS and introduce basic theoretic topics (e.g., various software architectures, real-time multi-task scheduling strategies, etc.) in their microcontroller courses [1]~[4]. As a result, students who may have a good understanding of theory and concepts of RTOS do not have the opportunity to map their knowledge onto real-world implementations. To bridge the gap between conceptual understanding and concrete implementations, a Real Time Operating System educational platform has been established for EET students in the Department of Engineering Technology & Industrial Distribution at Texas A&M University as well as in the Engineering Technology program at Northern Kentucky University.

Development of the Embedded RTOS Educational Platform

The embedded RTOS educational platform consists of two parts: the Modular Integrated Stackable Layer (MISL) intelligent layer and the analog system environment (ASE) board (Figure 1). The MISL intelligence layer, typically the TI-MSP430F5438A, can be directly interfaced to the ASE board. The RTOS platform has integrated numerous typical analog devices and new communication technologies into RTOS curriculum education, which mainly includes MISL architecture, inputs and outputs, analog and digital signal conversion, and wired & wireless communications. This platform is also designed for embedded systems education from entry-level courses at the sophomore level to final Capstone Design projects.

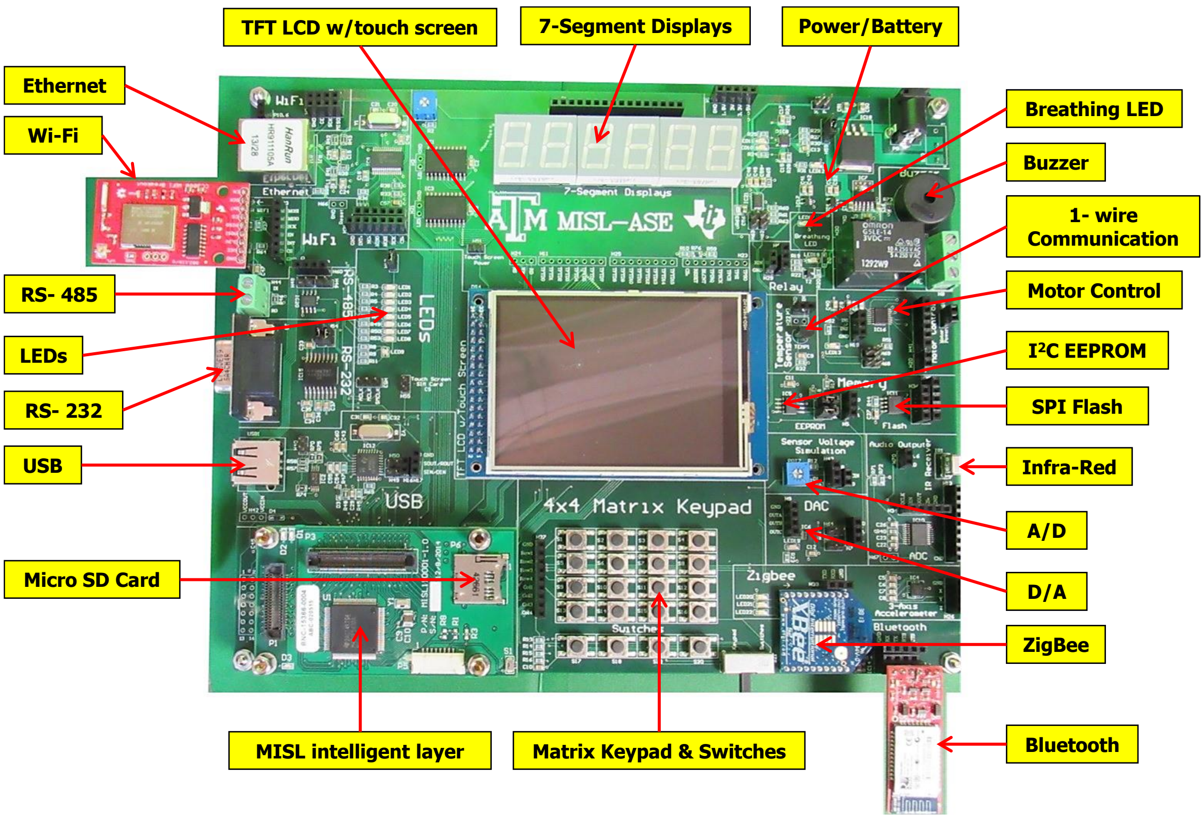

Figure 1. The RTOS educational platform.

1. MISL architecture

The Command and Data Handling branch of the Johnson Space Center, NASA created the Modular Integrated Stackable Layers (MISL) system [5]. The MISL system is a rack-and-stack space qualified system which allows the user to mix and match a variety of developed boards to most effectively fit the project's needs. The MISL system architecture utilizes a standardized power bus as well as a standardized data bus which provides communication and power distribution between layers. The standards for the busses are set by NASA. The MISL architecture is an open source environment which encourages any interested person to develop their own layer and provides all the necessary documentation a person would need to recreate any of the previously developed boards. The ESET program, Texas A&M University has partnered with NASA in the development of new layers for the MISL architecture [6].

Figure 2 illustrates the MISL architecture: bus connectors and MISL board layout. The data bus connector is always positioned horizontally on the relative top of the board while the power bus connector is always positioned vertical at the left side of the board. This standard placement allows for the MISL board to be any size but still keep the data and power bus connectors correctly located.

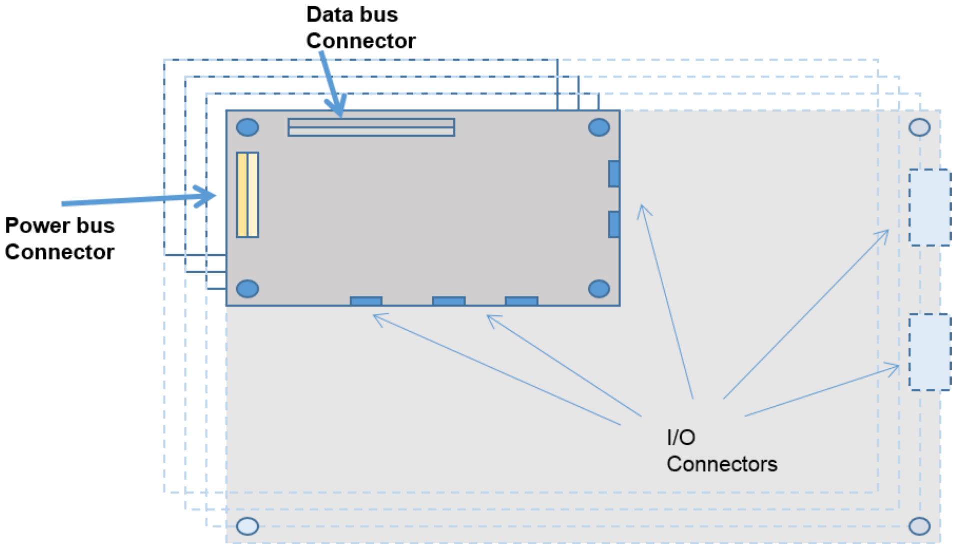

Figure 2. MISL architecture: bus connectors and MISL board layout

The height of the MISL stack varies depending on the number of boards used. As shown in Figure 3, there are two possible heights to both the male and female connectors: 0.354 and 0.276 inches.

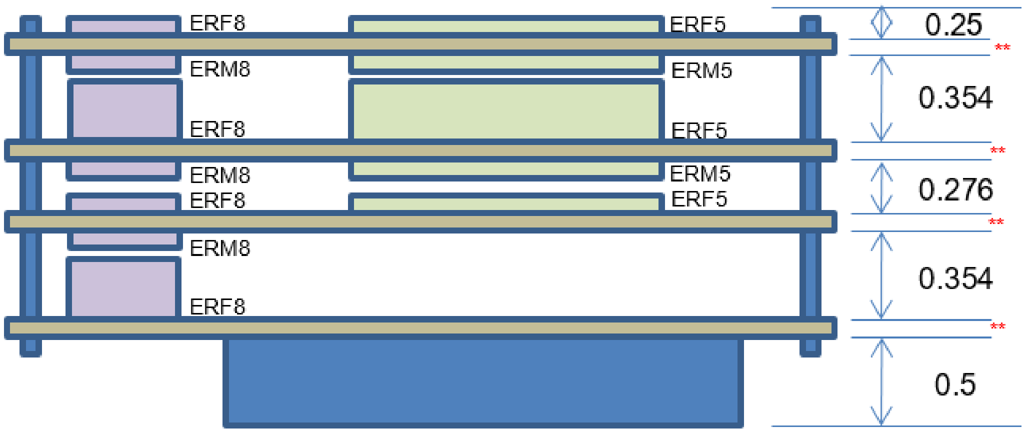

Figure 3. Stack Height

The RTOS platform employs the MISL- MSP430F54238A intelligence layer as shown in the lower-left corner of Figure 1. This intelligence layer is plugged into the ASE board. It should be noted that the robust design using MISL architecture could allow a number of other embedded intelligence boards, such as the Launchpad development system or other microcontroller layers, to be interfaced to the ASE board.

2. The ASE Board

The ASE board encompasses various analog and digital peripherals and new wired/wireless communications.

The inputs and outputs part allows students to configure the GPIOs of the MSP430F5438A as either outputs to control on-board RGB LEDs, breathing LED (PWM), multiple 7-segment displays, audio system, buzzer, and TFT LCD with touch screen, etc. or inputs to read the levels from 4x4 matrix keypad and 4 independent switches.

The analog-digital signal conversion part includes a potentiometer to simulate different input voltages from analog sensors; a 12-bit 3-axis accelerometer (ADXL335) to detect the static acceleration of gravity in dynamic acceleration resulting from vibration, shock and motion; and a 16-bit, 8-channel SAR external ADC (ADS8345) that allows students to work with high-resolution ADCs instead of internal 12-bit ADCs inside the MSP430 microcontroller.

The communication part contains most popular wired & wireless communication interfaces and protocols, such as four-wire communication networks: SPI (Ethernet, 2.4 G Wi-Fi (TI-CC3000 and nRF24L01), Micro SD card, and flash memory SST25VF016B). Via SPI interfaces, numerous devices and equipment in the field can be remotely accessed, monitored, and controlled through internet and terminals; two-wire communication networks: UART (USB, RS-232/485, Bluetooth RN-42, and ZigBee-XBee2). Through ZigBee or other UART interfaces, students are able to develop wireless distributed sensor networks to control all sorts of automation systems in manufacturing factories, office, home or other places; two-wire communication networks: I^2C (DAC5574 and EEPROM ATMEL24C16); and 1-wire communication networks: DS18B20. The one-wire communication technology would allow student to design a novel distributed temperature monitoring systems instead of the traditional temperature measuring method using analog temperature sensors, signal conditioning circuits, and A/D converters.

Embedded RTOS Course Introduction

Final debug and testing of the embedded RTOS educational platform was completed in 2015. An initial order of 200 boards has been procured and populated. Moreover, a new real-time embedded systems design course has been created using this platform in the lab. This new RTOS course and lab provide an opportunity for EET students to (1) further understand embedded real-time kernel and software design; (2) demonstrate ability to use semaphores, mailboxes, messages queues and pipes; and (3) demonstrate ability to design and implement a real-time embedded application. The main topics covered in the class include:

- *Introduction* part talks about the concepts of software structures, foreground/background, real-time kernel, and time slice, and so on.

- *Operating system concepts* part discusses interrupts, tasks and states, task switching pending, and timer operation (one-shot timers, periodic timers, and time of day), and so on.
- *uC/OS-III* part introduces installation and use of uC/OS-III development environment and creation of multiple tasks.
- *Inter-task communications* part discusses semaphores (binary Semaphores and counting semaphores), mailboxes, message queues, priority inversion, deadlocks, and mutexes, and so on.
- *RTOS design application* part gives an example of environmental parameter monitoring system development. In this monitoring system, the real-time kernel (uC/OS-III) executes multiple tasks with different priorities, such as temperate/relative humidity sampling (high priority), data output on UART (medium priority), touchscreen display (medium priority), and keyboard (low priority).

Conclusions

This paper presents the overall hardware design and development of the embedded RTOS platform. A new real-time embedded systems design course using this platform is also briefly discussed. Unfortunately, the RTOS platform and new course evaluation is not included as the author would like to wait until he can collect more feedback from students and other users. The RTOS laboratory curriculum development and student learning outcomes will be discussed in more detail in the follow-up manuscript.

Acknowledgement

The author wishes to acknowledge the financial support of Texas Instruments, ECETDHA grant, the MISL design work of the C&DH Branch-NASA/JSC, and the Look College of Engineering at Texas A&M University.

References

1. Kornecki, Andrew and Farahazad Behi. "Real-Time Software Course for Undergraduate Computing Programs: Experience Report". http://pages.erau.edu/~kornecka/papers/INEER07korbeh.pdf
2. Bob Rinker. "CS452/552 - Real Time Operating Systems, University of Idaho". http://www2.cs.uidaho.edu/~rinker/
3. He, Nannan and Han-way Huang. "Use of FreeRTOS in Teaching Real-time Embedded Systems Design Course". In ASEE Annual Conference and Exposition, Conference Proceedings, Indianapolis, IN, United states, 2014.
4. Dick, Robert and Peter Dinda. "Introduction to Real-time Systems Syllabus, Northwestern University". http://www.ece.northwestern.edu/~dickrp
5. Modular Integrated Stackable Layers – http://www.nasa.gov/centers/johnson/engineering/projects/modular_instrumentation_system/index.html
6. J. Morgan. "Modular Integrated Stackable Layers (MISL): An Academic-Public Sector Partnership for Rapid Prototyping and Development," Proceedings of ASEE Annual Conference and Exposition, Conference Proceedings, Seattle, WA, United states, 2015.

Dr. Gang Sun, Northern Kentucky University

Dr. Gang Sun is currently an assistant professor in Engineering Technology program at Northern Kentucky University. From 2013 and 2016, he was an assistant professor in the Department of Engineering Technology and Industrial Distribution at Texas A&M University. His research involves designing Mechatronic Systems that integrate embedded systems, programmable logic controllers, ubiquitous computing, real-time operation systems (RTOS), wired/wireless communication, sensor networks, and actuator control in various industrial, agricultural, and environmental applications.