ASEE's 123rd Annual • Conference & Exposition • New Orleans, LA • June 26-29, 2016

New Orleans JAZZED about Engineering Education

Paper ID #15677

# WORK IN PROGRESS: An Integrated DSP and Embedded Microcontroller Laboratory Curriculum

**Prof. Todd D. Morton, Western Washington University**

Todd Morton has been teaching the upper level embedded systems and senior project courses for Western Washington University's Electrical Engineering and Electronics Engineering Technology program for 27 years. He is the author of the text 'Embedded Microcontrollers', which covers assembly and C programming in small real-time embedded systems and has worked as a design engineer at Physio Control Corporation and at NASA's Jet Propulsion Laboratory as an ASEE-NASA Summer Faculty Fellow. He has a BSEE and MSEE from the University of Washington.

**Dr. Ying Lin, Western Washington University**

Ying Lin has been with the faculty of Engineering and Design Department at Western Washington University since September 2010 after she taught for two years at SUNY, New Platz. She received her MS in Applied Statistics and Ph.D. in Electrical Engineering from Syracuse University, NY, respectively. Her teaching interests include first-year Intro to Electrical Engineering, and upper-division Signals and Systems, and Digital Signal Processing courses. Her research areas focus on statistical signal processing for wireless sensor network applications and secure communications in wireless networks.

# Work-in-Progress: An Integrated DSP and Embedded Microcontroller Laboratory Curriculum


## Abstract

In this paper, we present our ongoing effort and progress in developing digital signal processing (DSP) laboratory coursework based on an embedded microcontroller (MCU)-based development platform. In particular, the MCU platform adopted for the DSP course uses the same ARM Cortex-M4 platform used in our embedded systems courses at Western Washington University. The goals of this work are twofold. One, by reducing student time spent on learning new development hardware and software, additional concepts can be introduced in the lab and enhanced learning outcomes can be added to the DSP course. Two, by introducing DSP implementation in embedded MCUs, student success in completing DSP-based capstone projects that meet realistic design constraints will be improved. Learning outcomes and assessment measures are introduced to determine the success of the work. If shown to be successful, this work may be adapted to the EE curricula in other institutions and may be adopted in other curricular areas such as controls and communications.


## I Introduction

Digital signal processing (DSP) is one of the essential topics for modern electrical and computer engineering curriculum[1]. It is widely acknowledged that well-designed hands-on laboratory experiments enhance students' understanding of DSP theory and fundamentals and are an effective complement to lectures[2]. Traditionally, implementing real-time DSP algorithms has been conducted using dedicated and expensive DSP hardware. In this case, we used the Texas Instrument's TMS320C6713DSK board with the CCS software development tool. Such platforms offer powerful processing capabilities for DSP applications, however, they also require students to learn the necessary hardware structure and the software development tool in order to use the platform. This demands much effort and time for students before they are able to work on the hands-on lab experiments. In our case, this poses a great challenge because the DSP course is a one-quarter course (a term of ten weeks). In addition, they are no longer the best solution to most designs that must meet realistic constraints such as cost, size, and power consumption.

Recent advances in DSP-enhanced MCUs (e.g. ARM Cortex-M MCUs) have made the inexpensive embedded microcontroller an option for most DSP applications and therefore a practical option for the DSP laboratory[2].

Our objectives have been to identify and to develop a common hardware and software platform that meet the requirements of the DSP course, the embedded microcontroller courses, and the culminating project. This unified approach of adopting common hardware and software tools for multiple courses greatly reduces students' time to learn how to use different tools, thus permits more time and effort to be focused on understanding the course concepts and attaining the desired learning outcomes.

We have completed work on the hardware and software platform and have identified the important learning outcomes and assessment measures. We are now working on the new DSP course laboratory experiments that incorporate these new tools and address the new learning outcomes.

These hands-on laboratory experiments are aimed to enhance students' understanding of the DSP fundamental concepts including sampling theory, spectrum analysis through DFT/FFT, and digital filtering. Students will implement these concepts in the lab using both stand-alone C programs and C programs that use the ARM CMSIS DSP library. These experiments will be introduced to the students for the first time spring quarter 2016.

In this Work-in-Progress paper, we will present the detailed descriptions and justification of the selected hardware and software, course learning outcomes, assessment measures, and the scope of developed lab experiments. Assessment results to demonstrate the effectiveness of the developed lab coursework and proposed modifications will be presented in a future full paper.

## II Summary of selected MCU platform, CODEC, and software

The goal of this project is to use the same hardware and software platforms in the DSP course as those used in the embedded microcontroller courses. This would allow students in the DSP course to focus immediately on the DSP concepts without having to spend time learning how to use new hardware or software. The challenge has been to find a platform that meets the various constraints placed on practical embedded systems combined with the requirements for DSP. With the introduction of the ARM Cortex-M microcontrollers, especially the Cortex-M4 microcontrollers with both DSP and floating-point extensions, this goal has been made much more realizable.

### II.A Hardware tools adopted

Given that the ARM Cortex-M microcontrollers are offered by multiple vendors in a very large range of form factors and peripheral content, there are several vendors, each with many part variations that would fit the requirements for this work. We chose to use the NXP (formerly Freescale) Kinetis family because of its broad range of parts. This allows our seniors to choose an MCU that fits the constraints of their projects, while using the same hardware and software development eco-system.

The hardware tool adapted for this work includes the NXP Kinetis TWR-K65F180M development board and an in-house designed CODEC board. Both of these boards are used as part of the NXP Tower System Modular Development Board Platform as shown in Figure 1. This system allows for additional boards to be added for additional features. For example, we have an in-house developed user interface board with a character LCD module and keypad.

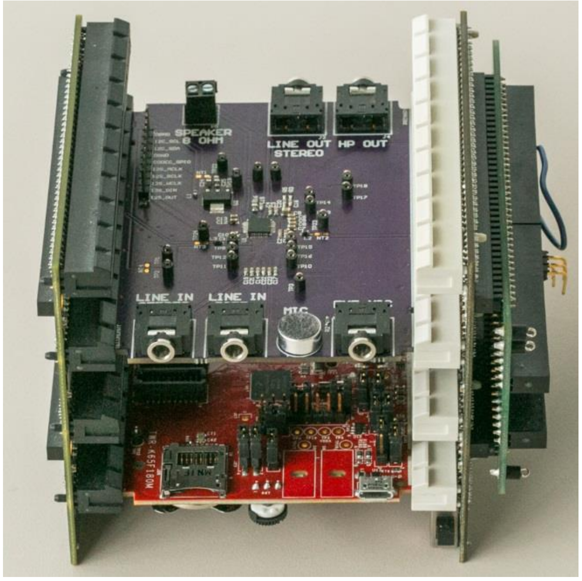


Figure 1: DSP platform based on TWR-K65F180M board and custom CODEC.

## Kinetis TWR-K65F180M development board

The development board is based on the NXP Kinetis K65 microcontroller. The K65 microcontroller is ARM Cortex-M4 based, runs at a clock speed of 180 MHZ, and has floating point and DSP extensions. It has 256kB of on-chip SRAM, which has been adequate for the large array blocks required for sample buffering and DSP processing. It also includes DMA, $I^2S$, and $I^2C$, which are all used in this work. This microcontroller allows for both fixed-point and floating point DSP.

The board itself does not include audio resources and the K65's DAC is only 12-bits. Because audio is the primary application used in the DSP labs, another board (i.e., a CODEC board) is required to provide these features.

## In-House designed CODEC board

The CODEC board is based on Texas Instrument's TLV320AIC3007 Stereo Audio CODEC. This CODEC has many configuration options and requires a low chip count for basic audio inputs and outputs as can be seen in Figure 1. The block diagram of the CODEC board is shown in Figure 2. The CODEC board includes an on-board microphone, an external microphone jack, stereo line-in, differential line-in, stereo headphone out, stereo line-out, and a terminal block for a small speaker. The digital audio to/from the microcontroller is over the $I^2S$ bus and CODEC configuration from the microcontroller is over $I^2C$.

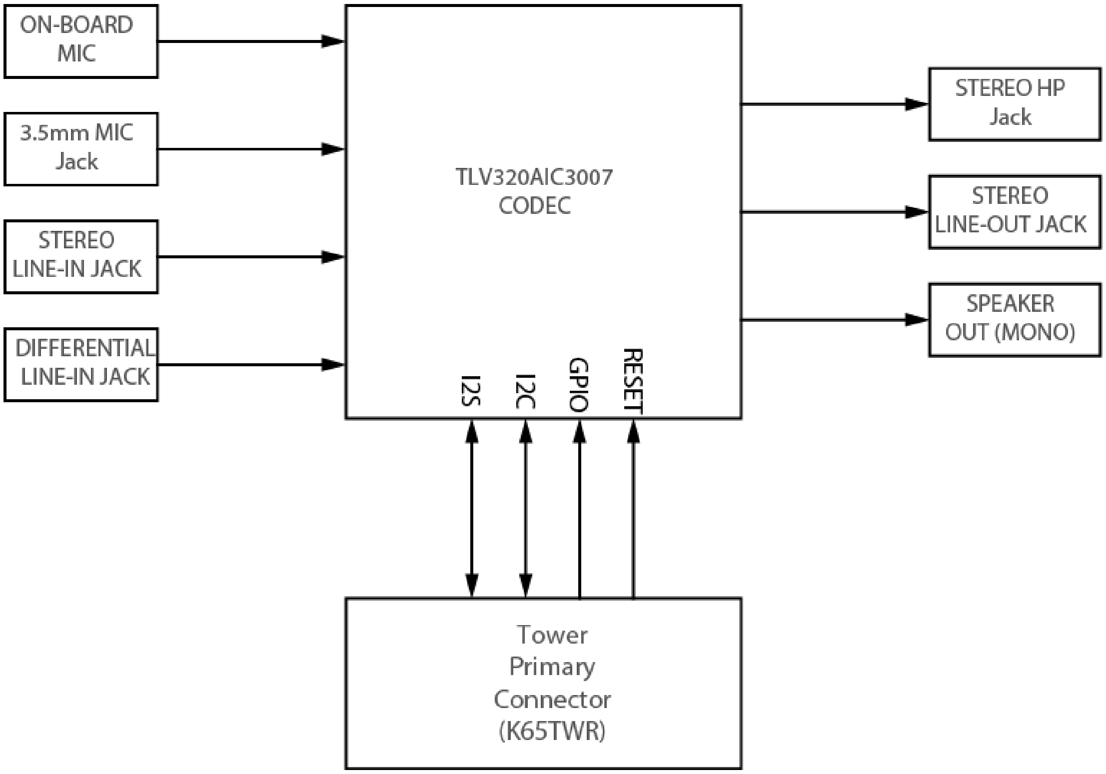


Figure 2: CODEC board block diagram.

**II.B Software tools adopted**

The software tools adapted for this work are the NXP Kinetis Design Studio (KDS) and the ARM CMSIS DSP code library. Optionally, the students can write their programs to run in a real-time kernel environment as they are already proficient in writing MicroC/OS applications.

The KDS IDE is an Eclipse-based code development system that is available at no cost from NXP. Most importantly, the students have used this development system in multiple courses for three quarters prior to taking the DSP course. Table 1 shows a summary of the course sequence, in order. What has been a long learning process to learn a new IDE, should now be eliminated.

| **Course Number/Title** | **Scope of the Course** |
|---|---|
| EE244 Embedded Microcontrollers I | Introduction to microcontrollers, software development using KDS, assembly language programming |
| EE344 Embedded Microcontrollers II | Simple multitasking, peripheral drivers, C language programming, using GIT in KDS |

| | |
|---|---|
| EE444 Embedded Systems | Real-time kernels, team-based software development, and advanced applications |

Table 1: Summary of Courses using the KDS IDE software tool and the MCU platform

The ARM CMSIS DSP code library is available for free from either ARM or the Kinetis SDK. It includes all of the necessary DSP functions for the DSP course. It includes support for 32-bit floating-point, 32-bit fixed-point, 16-bit fixed-point, and 8-bit fixed-point. Students will need to learn how to use this library, which does come with good documentation.

Firmware has been developed and will be available to the student through KDS project imports and GIT repositories. The firmware includes the drivers for the CODEC, $I^2S$, $I^2C$, and DMA along with code to complete the sample stream from/to the CODEC to sample blocks available for the student's processing code. This allows the student to focus on the processing algorithms. However, the sample stream is highly configurable so sample size and type, sample rate, number of channels, and block size can all be set by the students.

**III: DSP course improvement goals and changes**

The goals of the proposed changes to the DSP course laboratories are to achieve a higher attainment level of the current course learning outcomes and the attainment of new learning outcomes made possible by this new system. To assess the effectiveness of the proposed lab tools and lab coursework, we will adopt several assessment measures as elaborated below.

**III.A DSP course – desired outcomes and assessment measures:**

Table 2 summarizes the desired course learning outcomes upon students' completion of the course.

Note that in our EE program, prior taking the DSP course, students have completed a pre-requisite discrete-time systems course in which students learn foundational topics such as sampling and the Discrete Fourier Transform (DFT). In the DSP class, the focus is on Fast Fourier Transform (FFT), spectrum analysis, digital filtering, and the implementation of DSP algorithms in the lab.

| Learning Outcomes | |
|---|---|
| 1. | Implement Fast Fourier transform (FFT) to analyze the frequency spectrum of digital signals. |
| 2. | Design and implement FIR filters and IIR filters and analyze filter performances including passband ripples/gain, passband roll-off (transition region width), and stop band attenuation. |
| 3. | Obtain filter impulse response and frequency responses given FIR and IIR filter structures, difference equations, or transfer functions. |
| 4. | Recognize the impact of quantization and fixed-point and floating-point hardware. |
| 5. | Apply DSP techniques to design solutions for practical applications. |
| 6. | Use software tools for DSP modeling and simulations. |
| 7. | Implement DSP algorithms in real-time embedded hardware. |

Table 2: DSP course learning outcomes

In our institution, the courses are offered in 10-week quarters. Due to the limited time, in the past, students were not able to implement certain more complicated lab coursework, e.g., the integration of digital filtering and FFT algorithm to carry out real-time adaptive filtering (proposed Lab #5 in Section III.B). This is because a portion of lecture and lab time were needed to introduce students to the dedicated DSP hardware and software before students were ready to program the hardware to implement DSP labs. Our goal is, by adopting the MCU-based platform, to greatly reduce the time students spend on learning a new tool so they can dedicate more time and effort to work on more content and concepts in the labs. More discussions and summaries of the existing lab coursework and the proposed new lab coursework are provided in Section III.B.

To gauge the effectiveness of the selected MCU-based platform compared to the traditional dedicated DSP hardware, we propose to use the following assessment measures:

- First, students' feedback in the form of survey questionnaires will be collected. We plan to conduct two sets of surveys: One from students who completed the DSP class in spring 2015. In this course, the dedicated DSP hardware was used. As a comparison, we will also obtain surveys from students who will complete the DSP class in spring 2016. In this course, the MCU-based DSP hardware will be used. Such feedback will help us to assess how the students feel about the effectiveness of the proposed MCU-based platform compared to the DSP-based platform.
- Secondly, students' performance in completing the DSP lab coursework will be tracked, especially the labs on fixed-point DSP implementation and adaptive filtering. How successful students complete such lab coursework can be a useful indicator to show the effectiveness of the chosen MCU-based platform.

Some of the detailed assessment results will be available by the end of spring 2016 quarter and will be presented in a future full paper.

### III.B DSP course – proposed lab experiments based on TWR-K65F180M board

In this section, we present the scope of the developed hands-on lab experiments, which are aimed to reinforce students understanding of the DSP concepts covered in the lectures and also provide opportunities to implement real-time DSP algorithms. To make the laboratory coursework more engaging, we chose real-life audio signals as the input sources, e.g., music from students' cell phones or ipod, human voice signals, or other types of audio input.

The contents of the developed lab experiments are tied closely to the concepts introduced in lectures. Table 3 shows the scope of each proposed lab.

The detailed lab experiment descriptions will be presented in a future full paper.

| **Lab Experiment** | **Concepts/Topics Related** | **Scope of the Lab Experiment** |
|---|---|---|
| #1 | FFT, spectrum analysis | Given a set of different input signals, implement a FFT algorithm to conduct spectrum analysis. |
| #2 | FIR filter<br>Discrete convolution<br>Frequency response | FIR filter implementation and frequency response analysis. |
| #3 | IIR filter<br>Filter implementation structures | IIR filter implementation and frequency response analysis. |
| #4 | Impact of fixed-point | Given an input audio signal, conduct IIR filtering to explore the impact of using fixed-point numbers. |
| #5 | Adaptive filtering | Integration of digital filtering and FFT to realize real-time adaptive filtering implementation based on input signal spectrum analysis. |

Table 3: Summary of proposed lab coursework

As a comparison, we also provide the summary of the labs using the dedicated TMS320C6713DSK board in previous DSP courses in Table 4.

| **Lab Experiment** | **Concepts/Topics Related** | **Scope of the Lab Experiment** |
|---|---|---|
| #1 | Introduction to TMS320C6713DSK board and the CCS IDE | The DSK board basics, CCS functionalities, programming tips |

| | | |
|---|---|---|
| #2 | FFT using CCS | Spectrum analysis of different input signals |
| #3 | FIR filter implementation | FIR filter implementation and frequency response analysis. |
| #4 | IIR filter Filter implementation | IIR filter implementation and frequency response analysis. |

Table 4: Summary of existing Lab coursework using the dedicated TMS320C6713DSK board

The new proposed labs that use the MCU-based platform would facilitate students to attain more content and practice DSP topics with more depth and complexity.

**IV: Senior capstone project improvement goals and assessment measures**

The Electrical Engineering program at Western Washington University is in the process of transitioning from an ABET-ETAC Electronics Engineering Technology (EET) program to an ABET-EAC Electrical Engineering (EE) program. All aspects of the curriculum are being evaluated and, when necessary, modified in order to meet ABET-EAC criteria.

The capstone project courses are designed to meet the EAC Criterion 5 section that states[3]:

*Students must be prepared for engineering practice through a curriculum culminating in a major design experience based on the knowledge and skills acquired in earlier course work and incorporating appropriate engineering standards and multiple realistic constraints.*

This requires that a new emphasis on '*multiple realistic constraints*' must be made in the course content and assessment.

This is also expressed in the ABET-EAC student outcomes:

*(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability*

To assess the effectiveness of this work, we will focus on the following outcome:

1. Students are able to successfully complete a design project that uses DSP to meet realistic constraints.

The following analysis shows that most projects in the past five years that should have used DSP to meet realistic constraints were either avoided, did not meet the constraints, or were not successfully completed.  Data was compiled from the senior projects for the last five years (2011-2015).

Out of a total of 84 projects, 18 (21%) should have used DSP to meet realistic constraints. As an aside to the main focus of this work, this small percentage may indicate that students avoid projects that require a significant amount of DSP. Assessment will be made during the project proposal stage to see if more students choose a project that requires DSP.

Table 5 shows that, out of the 18 projects that should have used DSP, five used dedicated DSP hardware, two used microcontrollers, and 11 avoided using DSP by using analog signal processing.

| | DSP | MCU | Analog |
|---|---|---|---|
| Method Chosen | 5 | 2 | 11 |
| Met Constraints | 1 | 2 | 0 |
| Successful Completion | 1 | 1 | - |

Table 5: Historical DSP project success data

An assessment of these historical results to measure Outcome 1 above shows only three of these projects met realistic constraints – one used a dedicated DSP chip and two used a microcontroller. The others did not meet realistic constraints such as cost, size, and power consumption because they either tried to use the DSP hardware that was used in the DSP course or they had a much higher chip count.

Out of the three projects that met constraints, only two were successfully completed. This historic low attainment of Outcome 1 is one area this work is designed to improve. These assessment measures will continue to be made to see if the work described in this paper is successful at improving these outcomes.

## V. Conclusions

A new MCU-based lab platform has been developed and proposed to be used in the DSP course labs. The motivation of this change is to increase student attainment of learning outcomes in both the DSP course and culminating projects. Assessment measures have been developed to measure the effectiveness of the new system which will be used in the DSP course for the first time in spring 2016. The first senior cohort that will be assessed for this work will complete their culminating project during the spring 2017. If the system proves successful, it will be useful to other universities and to other areas of the EE curriculum such as controls and communications.

**References**


1. Cameron H.G. Wright, Thad B. Welch, and Michael G. Morrow, “Real-time DSP using “See-Through.”, Computers in Education Journal, Vol 6, No. 2, April – June 2015, pp 31-39.
2. Donald S. Reay, “Digital Signal Processing Using the ARM Cortex-M4”, John Wiley & Sons, 2016.
3. ABET, “Criteria for Accrediting Engineering Programs, 2016 – 2017.