# ECE 362 Lab Verification / Evaluation Form

Experiment 9

## Evaluation:

**IMPORTANT! You must complete this experiment during your scheduled lab period. All work for this experiment must be demonstrated to and verified by your lab instructor *before the end* of your scheduled lab period.**

| STEP | DESCRIPTION | MAX | SCORE |
|---|---|---|---|
| DP 1 | Demo program load/run, potentiometer verification | **1** | |
| DP 2 | MSO frequency spectrum display | **1** | |
| DP 3 | LPF cutoff frequency calculation/verification | **1** | |
| DP 4 | Input waveform type/frequency trials | **4** | |
| SW 1 | Software (completed by the end of your scheduled lab period) | **8*** | |
| SW 2 | Input/output sampling frequency trials | **4** | |
| SW 3 | Submission (completed immediately following demonstration) | **4*** | |
| TQ | Thought questions | **2** | |
| Bonus | Sawtooth Waveform Generator | **5** | |
| | **TOTAL** | **25+** | |

** code must function as specified to receive full credit for software and submission scores (score for non-functioning code will be 20% of max for both software and submission)*

**Signature of Evaluator:** ______________________________________________

## Academic Honesty Statement:

**IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. *You will not receive credit for this lab experiment unless this statement is signed in the presence of your lab instructor.***

*"In signing this statement, I hereby certify that the work on this experiment is my own and that I have not copied the work of any other student (past or present) while completing this experiment. I understand that if I fail to honor this agreement, I will receive a score of ZERO for this experiment and be subject to possible disciplinary action."*

Printed Name: ______________________ Class No. __ __ __ __ - __

Signature: ____________________________________ Date: ________

# Experiment 9: Analog Signal Sampling and Reconstruction

**Instructional Objectives**

- To illustrate how analog signals can be sampled using the ATD and reconstructed using the PWM (used in conjunction with a low-pass filter)
- To illustrate the effects of input sampling rate and output sampling rate on the quality of signal reconstruction

**Parts Required**

- 2 x 16 LCD and 16-pin single-row header (previous experiment)
- GAL22V10 programmed as 8-bit shift register (previous experiment)
- 10 KΩ potentiometer
- Breadboard and wire

**Preparation**

- Read this document in its entirety
- Review material on the ATD, PWM, TIM, and SPI
- Complete interface wiring on your breadboard

**Introduction**

In lecture you have been introduced to the concept of data acquisition using a microcontroller. Here, the analog-to-digital converter (ATD) can be used to uniformly sample a continuous-time input signal, an encoding process referred to as *pulse code modulation* (PCM). Recall that the minimum frequency at which a continuous time signal should be sampled (Fs) is twice the highest frequency component that signal contains (referred to as the Nyquist frequency). Traditionally, PCM data can be reconstructed using a digital-to-analog (DTA) converter, the output of which is low-pass filtered (to remove "images" of the reconstructed signal centered at integer multiples of Fs). In the absence of an integrated DTA, another peripheral that can be used for signal reconstruction is the PWM unit. Here, the uniformly sampled data from the ATD is converted into a time-varying PWM duty cycle. The resulting "naturally sampled" square wave is then "transformed" into a continuous time signal by low-pass filtering it. One of the goals of this experiment is for you to witness this seemingly "magical transformation" first hand, as well as qualitatively compare the effects of input and output sampling frequencies on the reconstructed waveform.

**Demo Program**

A simple demo program is provided to help you get started as well as test your low-pass filter and function generator setup (Code Warrior project folder **Lab9-demo**). This program initializes the TIM and ATD modules to sample an analog input at **10,000 Hz** (0.1 ms timer interrupt) and then reconstruct the input signal with a **47,059 Hz** PWM output sampling frequency. It also includes provisions for a "digital volume control".

**DP Step 1a:** Set the MSO function generator to produce a **1000 Hz, 5.0 V** ***peak-to-peak sine wave*** that is **<u>offset</u>** by **+2.5 V DC** (*carefully verify this before connecting the MSO function generator output to your board!*).

 

**DP Step 1b:** Connect the MSO function generator output to **PAD0** (ATD input channel 0) and to oscilloscope channel 1. Connect the wiper of your potentiometer (with the legs connected to +5V and GND) to **PAD1** (ATD input channel 1). Connect channel 2 of the oscilloscope to **PT0** (PWM channel 0). Load and run the **Lab9-demo** program provided on the course web site. Verify that the potentiometer works as expected (i.e. as a "digital volume control") and set it to its maximum output.

*Save the display obtained for* ***one period of the input sine wave*** (*show both channels: the input sine wave and the "raw" PWM output signal*) to a USB drive.

*Place all saved displays in a single document for grading; when printing, "tile" multiple displays (suggest 4) per page with* ***inverted colors*** *before printing.*

**DP Step 2:** Plotting Frequency Spectrum. Now adjust the MSO to display roughly 10 periods of the sine wave. On the MSO, press the "Math" key. From the menu, select Operator: "FFT", Span: "50.0 kHz", Center: "25.0 kHz", and Vertical Units: "V RMS." The purple line will now display an approximate frequency spectrum from 0 to 50 kHz obtained by calculating the Fast Fourier Transform. *Save the FFT displays obtained for both the input sine wave and the "raw" PWM out.*

*Strongest frequency peaks in raw PWM signal:*

*Lowest frequency of PWM "artifacts":*

*Operation needed for "perfect" reconstruction:*

**DP Step 3:** Using a **1 KΩ** resistor and a **0.1 μF** capacitor, construct a single-pole low-pass filter (LPF) for the PWM output and connect it between the PWM output pin and the oscilloscope input (channel 2). Leave channel 1 of the oscilloscope connected to the MSO function generator. *Save the waveform and FFT display obtained for the low-pass filtered PWM output and compare it to the input.* Calculate the -3 dB *cutoff frequency* ($f_{-3\text{ dB}}$) of the LPF.

*LPF -3 dB cutoff frequency* ($f_{-3\text{ dB}}$) =

*Comparison of input, "raw PWM", and filtered spectrums:*

 

**DP Step 4a:** Set the function generator to produce sine waves at 100 Hz, 1000 Hz, 2000 Hz, 8000 Hz, 9000 Hz, 9900 Hz, 10100 Hz, 11000 Hz, and 12000 Hz. *Save MSO displays comparing input waveform to filtered PWM output.* Note the interesting relationship between the first three, middle three, and last three reconstructed waveforms. What is the name of this phenomena, and how does it relate to the Nyquist frequency?

| |
|---|
| *Interesting relationship of reconstructed waves and Nyquist frequency:* |
| *Name of phenomena:* |

**DP Step 4b:** Now adjust the FFT to Span: "10.0kHz", Center: "5.00kHz". Set the function generator to produce sine, square, triangle, and sawtooth (triangle with 0% symmetry) waves at 1000 Hz. *Save an MSO display for each waveform input and filtered output with their corresponding FFT*. Record the prominent frequency peaks for each waveform, along with the quality of reconstruction.

| | *Input frequency peaks* | *Output frequency peaks* | *Reconstruction quality* |
|---|---|---|---|
| *Sine* | | | |
| *Square* | | | |
| *Triangle* | | | |
| *Sawtooth* | | | |
| *Suggest two ways to improve reconstruction quality:* | | | |

 

**Software Description**

Write an application program in C that allows the input and output sampling frequencies to be changed, thus facilitating an efficient comparison of signal reconstruction quality. Similar to the demo program provided, ATD Ch 0 will be used to input the analog signal from the function generator, ATD Ch 1 will be interfaced to a 10 KΩ potentiometer ("digital volume control"), and PWM Ch 0 (output on PT0) will be used to provide the output signal. The left and right pushbuttons on the docking module will be used to "cycle" among the input and output sampling frequency choices available, respectively. Pushing the left pushbutton will allow the user to cycle through the choices available for input sampling frequency: 5000 Hz, 10,000 Hz, and 20,000 Hz. Note that the input sampling frequency can be controlled by changing the value loaded in <u>TC7</u>. Pressing the right pushbutton will allow the user to cycle through the choices available for output sampling frequency: 23,529 Hz, 47,059 Hz, and 94,118 Hz. Note that the output sampling frequency can be controlled by the <u>PWM clock pre-scalar</u>.

Use the LCD to display the current input sampling frequency (ISF) and output sampling frequency (OSF) settings, as illustrated in the example.

```
ISF: 10000 Hz
OSF: 47059 Hz
```

The RTI will be used to sample the pushbuttons on the docking board every 2.048 milliseconds. Data for the LCD display will be shifted out to an external shift register (GAL22V10) and will be interfaced to the SPI module through Port M (PTM). The description of the LCD interface used previously is reproduced below for your convenience.

An external 8-bit shift register (GAL22V10) will be used to interface LCD to the microcontroller via the SPI module (MOSI, port pin PM[4]; and SCK, port pin PM[5]). The LCD will be interfaced as described to the microcontroller module as described in the table below:

Pin 1


| LCD Pin # | LCD Pin Description | Connected to Microcontroller |
|---|---|---|
| 1 | Vss (ground) | Vss (ground) |
| 2 | Vcc (+5V) | Vcc (+5V) |
| 3 | VEE (contrast adjust) | Vss (ground) |
| 4 | R/S (register select) | PTT[2] |
| 5 | R/W' (LCD read/write) | PTT[3] |
| 6 | LCD Clock | PTT[4] |
| 7 | DB[0] (LSb) | Q[0] |
| 8 | DB[1] | Q[1] |
| 9 | DB[2] | Q[2] |
| 10 | DB[3] | Q[3] |
| 11 | DB[4] | Q[4] |
| 12 | DB[5] | Q[5] |
| 13 | DB[6] | Q[6] |
| 14 | DB[7] (MSb) | Q[7] |
| 15 | Not connected | |
| 16 | Not connected | |

**<u>NOTE</u>: DB[#] are the LCD data inputs and Q[#] are the data outputs of the GAL22V10 shift register.**

Some of the LCD pins require a bit more explanation:

| Mnemonic | Name | Description |
|---|---|---|
| `RS` | `Register select` | `This pin is logic 0 when sending an instruction command over the LCD data bus and logic 1 when sending a character.` |
| `R/W'` | `Read/write` | `This pin is logic 0 when writing to the LCD, logic 1 when reading from it. For this lab, we will only write to the LCD.` |
| `LCDCLK` | `LCD clock` | `This pin latches in the data on the data[7:0] bus on the falling edge. Therefore, this line should idle as logic 1.` |

**SW Step 1:**
Complete the "C" skeleton file provided in the Code Warrior **Lab9** project folder on the course website. Note that the "finished product" should work in a "turn key" fashion, i.e., your application code should be stored in flash memory and begin running upon power-on or reset. After completing your C program, test its operation by selecting an input sampling frequency of 10,000 Hz and an output sampling frequency of 47,059 Hz. Verify that the results obtained (LPF output waveforms) with a 1000 Hz sine wave as input are identical to those produced by the demo program.

**SW Step 2:** With a 1000 Hz sine wave as input, cycle through each possible combination of input and output sampling frequencies (total of 9 cases). Note which waveforms are produced with the greatest accuracy, and which are produced with the least.

*Save a display of each low-pass filtered output with its corresponding FFT spectrum.*

| |
|---|
| *Waveform comparison notes:* |

**SW Step 3.**
Zip your completed **Lab 9** Code Warrior project folder and submit it on-line (using the link at the bottom of the Lab Experiments page) *immediately after* demonstrating it to your lab T.A. ***Be sure identifying information*** (i.e., name, class number, and lab division) ***is included in the*** **`main.asm`** ***file you submit – credit will not be awarded if identifying information is omitted.***

**Thought Questions**
Answer the following thought questions in the space provided below:

(a) In your own words, describe the difference between *uniform* sampling (PCM encoding) and *natural* sampling.

(b) Provided Nyquist-related input sampling constraints have been met, what is the most important parameter that influences the quality (accuracy) of the reconstructed output signal? What is a “good working value” for this parameter?

(c) Describe what you would expect to observe if the value of R and/or C in the LPF were changed (thus changing the cut-off frequency).

(d) Describe what you would expect to observe if a *higher order* LPF were used in place of the single-pole filter.

**BONUS CREDIT:** Sawtooth Waveform Generator

Disconnect the MSO’s waveform generator from PAD0 and connect a second potentiometer. Instead of reconstructing the signal on PAD0, synthesize a sawtooth wave. Use the potentiometer connected to PAD0 to vary the frequency of the sawtooth wave, while PAD1 still controls the volume. How might this digital sawtooth wave be used to synthesize a sine or any other arbitrary waveform?

 