

# ECE 362 Lab Verification / Evaluation Form

## Experiment 7

---

### 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
1	Interfacing (completed prior to your scheduled lab period)	10	
2	Software (completed during your scheduled lab period)	10	
3	Submission (completed following demonstration)	5	
4	Bonus Credit	5	
	<b>TOTAL</b>	<b>25+</b>	

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 7: Two-Channel LED Bar Graph Display (BGD)

### Instructional Objectives:

- To illustrate I/O expansion using a shift register
- To illustrate an ATD application
- To illustrate an RTI application

### Parts Required:

- One GAL22V10 (from ECE 270 parts kit, also available from Instrument Room)
- Two 10 K $\Omega$  potentiometers (from ECE 270 parts kit, also available from Instrument Room)
- Four RED, four YELLOW, and two GREEN LEDs (from ECE 270 parts kit, also available from Instrument Room)
- Six 150  $\Omega$  resistors (from ECE 270 parts kit, also available from Instrument Room)
- Breadboard and wire (from ECE 270 parts kit)
- BONUS CREDIT OPTION: various diodes, resistors, and capacitors along with a connector/cable compatible with your favorite “personal listening device”

### Prelab Preparation:

- Read this document in its entirety
- Review the material on the RTI and ATD subsystems
- Wire and test GAL22V10 shift register/bar graph displays
- Read the *Programming Your Microcontroller in C* document

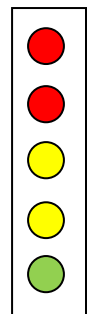
**NOTE:** All “C” code for this experiment must be written and debugged *during your scheduled lab period.*

### Overview

The objective of this lab is to create a two-channel LED bar-graph display (BGD) consisting of a 9S12C32 development kit interfaced to two potentiometers (referenced to 5 VDC). The BGD will sample the two analog input channels every 0.1 second (approx.) and display the converted values on two 5-segment LED displays, respectively.

The left pushbutton on the docking board will be used to “freeze” (stop) the BGD, while the right pushbutton will be used to start/ resume the BGD. The left LED on the docking board should be on when the BGD is stopped, while the right LED should be on when the BGD is running.

A set of five LEDs (two RED, two YELLOW, and one GREEN) will be used to indicate the approximate range of the voltage input on each channel (0 and 1), together creating an LED “bar graph” display (BGD). The GREEN LED for a given channel should be illuminated if the input voltage on that channel is greater than or equal to THRESH1. The “bottom” YELLOW LED for a given channel should be illuminated if the input voltage on that channel is greater than or equal to THRESH2. The “top” YELLOW LED for a given channel should be illuminated if the input voltage on that channel is greater than or equal to THRESH3. The “bottom” RED LED for a given channel should be illuminated if the input voltage on that channel is greater than or equal to THRESH4. Finally, the “top” RED LED for a given channel should be illuminated if the input voltage on that channel is greater than or equal to THRESH5. *The threshold values (THRESH1 – THRESH5) are constants that you will define.*



### Step 1. Interfacing

A 10-bit shift register, realized using a GAL22V10, will be used to interface the two LED bar graph displays to your microcontroller kit, as illustrated in Figure 1. The basic underlying concept was introduced in problems 4, 5, and 6 of the Module 2 homework set. Here, however, you will be configuring the GAL22V10 as a 10-bit shift register (instead of 8-bit), and driving individual LEDs (instead of a 7-segment display). Note that the GAL22V10 output pins should be configured as active low, i.e., to sink current when the corresponding shift register bit is “1”. Use pin 3 of Port T (“PT3”) to provide the *shift data* to your GAL22V10, and pin 4 of Port T (“PT4”) to provide the *shift clocking* signal. Write an ABEL program that realizes a 10-bit shift register for your GAL22V10 (this should be a simple adaptation of the code shown at the bottom of page 5 on the Module 2 homework solution). Test the operation of your shift register using DIP switches before interfacing it to your microcontroller kit, and demonstrate your hardware functionality to your lab T.A. *before you begin coding*.

**NOTE:** The locally-available GREEN and YELLOW LEDs are NON-RESISTOR, while the RED LEDs have an integrated current limiting resistor. Use a 150Ω current limiting resistor in series with each GREEN/YELLOW LED (anode connect to +5 VDC, cathode connected to GAL22V10 output pin).

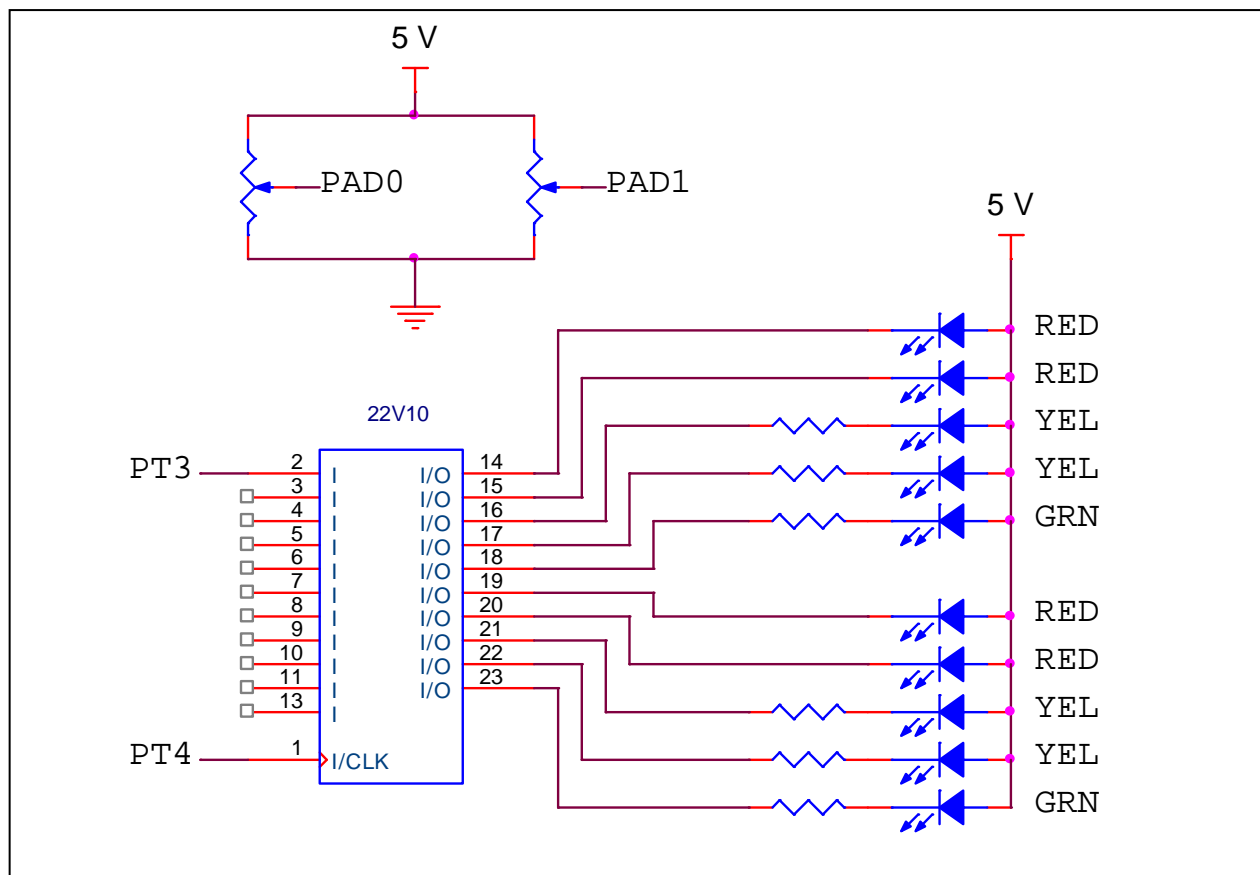


Figure 1. Interface of 10KΩ potentiometers and LEDs to 9S12C32 Development Kit.

**Step 2. Software**

**The “C” code for this application must be written, debugged, and demonstrated during your scheduled lab period.** The software modules needed are as follows:

1. Peripheral device initialization (RTI and ATD) – done before main loop commences.
2. Main program that checks left/right pushbutton flags as well as the tenths (of second) flag and takes appropriate actions – this should be performed in a continuous non-terminating loop, as outlined below:

```

If the "tenth second" flag is set, then
- clear the "tenth second" flag
- if "run/stop" flag is set, then
  - initiate ATD conversion sequence
  - apply thresholds to converted values
  - determine 5-bit bar graph bit settings for each input channel
  - transmit 10-bit data to external shift register
- endif
Endif

```

```

If the left pushbutton ("stop BGD") flag is set, then:
- clear the left pushbutton flag
- clear the "run/stop" flag ("freeze" BGD)
- turn on left LED/turn off right LED (on docking module)
Endif

```

```

If the right pushbutton ("start BGD") flag is set, then
- clear the right pushbutton flag
- set the "run/stop" flag (enable BGD updates)
- turn off left LED/turn on right LED (on docking module)
Endif

```

3. RTI interrupt service routine that detects pushbutton presses and sets left/right pushbutton flags (similar to Lab 6); also, determines when 0.1 second of RTI interrupts have occurred (track in global variable RTICNT), and sets “tenths” flag accordingly. The RTI interrupt rate should be on the order of 5-10 ms (approx.) to ensure accurate sampling of pushbuttons.
4. ATD device driver that performs a conversion sequence on input channels 0 and 1 (i.e., sequence length should be 2).
5. Shift register device driver, similar to the code outlined in problem 5 of the Module 2 homework – should shift out 10 bits of data to the GAL22V10 shift register/LED BGD.

Complete the “C” skeleton file provided on the course website **during your scheduled lab period**. 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. Demonstrate your completed system to your lab T.A.

**Step 3. Submission**

Zip the ABEL program you wrote for Part 1 (.abl file only) together with the C code you wrote for Part 2 (.c file only) and submit your solution file on-line after demonstrating it to your lab T.A. (but before leaving lab). ***Be sure identifying information*** (i.e., name, class number, and lab division) ***is included in the files you submit – credit will not be awarded if identifying information is omitted.***

#### Step 4. Bonus Credit

To make the two-channel BGD more interesting (“socially redeeming”), you can convert it into a stereo VU (“volume units”) meter for use with your favorite “low information music player” (LIMP) with the addition of some simple analog circuitry. Basically, the AC waveforms produced when playing music (fed by your LIMP to low-impedance “earbuds”) need to be rectified and low-pass filtered in order to create a useful/meaningful VU meter. Also, the voltage thresholds used by your C program need to be adjusted based on the realization that your typical LIMP produces at most a 1-volt PP (peak-to-peak) AC output.

One possible realization is illustrated in Figure 2. Here, the resistor  $R_a$  and capacitor  $C$  determine the “attack time” (quickness of response time constant) to the fluctuating audio signal and  $R_d$  determines the “decay time” (persistence of display to an audio impulse). Typically,  $R_d$  is significantly larger than  $R_a$  (i.e., attack time has a smaller time constant than decay time), but these are often based on user preference (so some experimentation may be necessary). Use your  $10\text{K}\Omega$  potentiometers to adjust the attack time ( $R_a$ ) for each channel, and resistors about an order of magnitude greater for the decay time ( $R_d$ ). Generic PN junction diodes can be used as the rectifiers. The value of  $C$  should be on the order of several microfarads (e.g.,  $10\text{ }\mu\text{F}$ ).

You can test your VU input circuit in advance by connecting its outputs (PAD0 and PAD1) to an oscilloscope and playing various genre of music on your LIMP and watching the attack/decay of the corresponding DC envelope (this also provides an opportunity to determine the peak DC voltage produced by your circuit, so that you can determine the appropriate BGD voltage thresholds). Note: “High information” VU meters are calibrated in decibels. The maximum voltage produced by your circuit should correspond to illuminating the top RED LED (along with all those below it) and be designated “0 dB”. This will serve as the reference voltage for calculating the thresholds at which the “lower” LEDs illuminate ( $V_{0\text{dB}}$ ). So, for example, if each LED in the BGD represents 3 dB VU steps, the corresponding threshold voltage for the “-3 dB RED LED” can be calculated based on solving the equation:

$$-3\text{ dB} = 20 \log_{10} (V_{\text{thresh-3dB}} / V_{0\text{dB}})$$

The remaining thresholds (-6 dB, -9 dB, and -12 dB) can be calculated in a similar fashion.

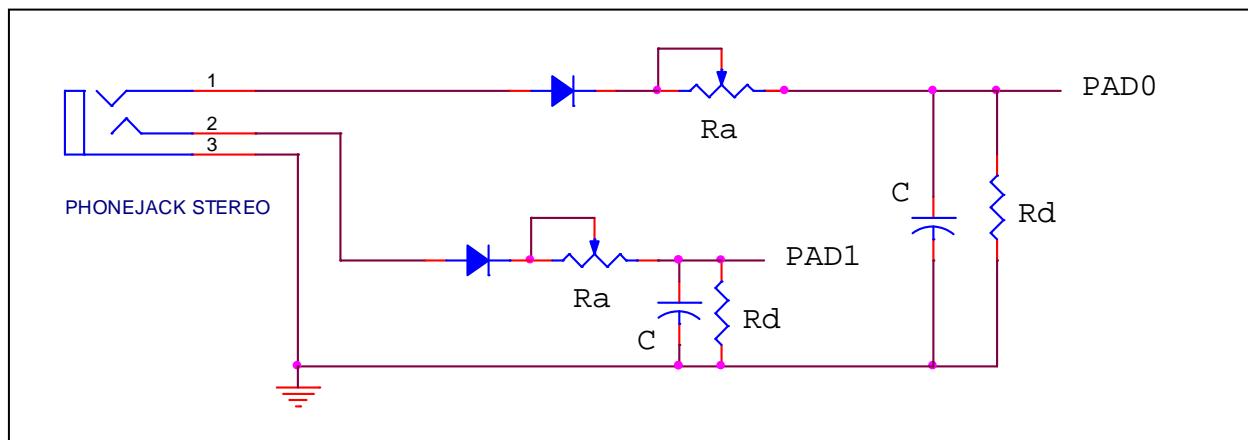


Figure 2. Circuit to Convert BGD into a Stereo VU Meter.