

# Lab #6: Time-Multiplexed LED Display using Programmable Timers

## Introduction

Real-time systems often require precise timing of data sampling and control operations. This is usually achieved through the use of programmable timer circuits that periodically interrupt the main processor at precise time intervals. To study timed interrupt-driven operation, this lab will require you to design and implement a time-multiplexed LED display using one of the microcontroller's programmable timers to interrupt the CPU.

The human eye is capable only capable of perceiving things that happen below a certain speed. Because of this, we are able to trick the human eye into perceiving things that aren't actually happening; one example of this would be varying the duty cycle of an LED in order to trick the human eye into believing the LED is glowing dimmer. In this lab, you will trick the human eye into perceiving a single image displayed on an LED matrix by turning on all of the LEDs in a single row or column at once and quickly cycling through all of the rows or columns.

## The Luckylight KWM-R30881CUAB LED Matrix

You will be provided a Luckylight KWM-R30881CUAB LED Matrix in lab as part of the gamepad. Figure 1 shows the circuit schematic from the datasheet, while Figure 2 presents the same information in an easier to read format. This matrix is known as a "common-cathode" or "cathode-row" matrix, where the cathode of each LED shares an electrical node with every other LED on that row. Note the difficulty in simultaneously addressing each individual LED in order to display complex images. For example, you cannot "checkerboard" the entire matrix at once. This is the need for time-multiplexing, as discussed above. Quickly displaying the image one row or column at a time allows you to display any combination of LEDs "simultaneously".

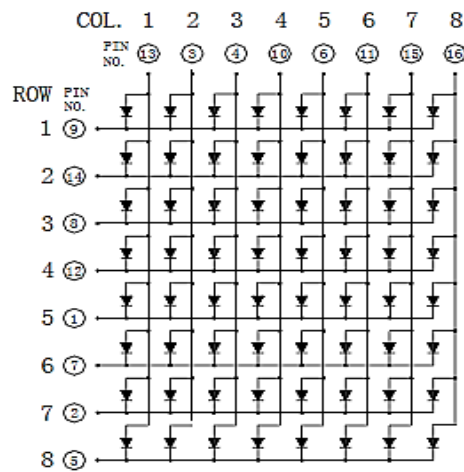


Figure 1: KWM-R30881CUAB LED Matrix Schematic from Datasheet

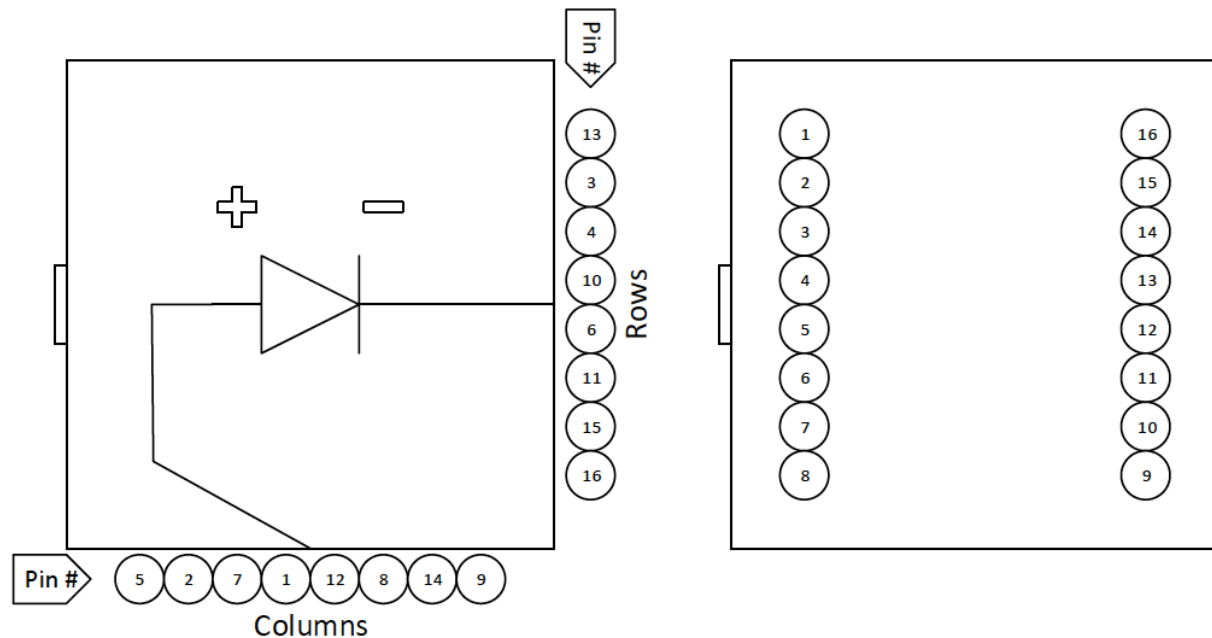


Figure 2: Simplified Schematic of KWM-R30881CUAB

## Pre-Lab Assignment

### Reading

Study the Monday lab lecture slides, *STM32L100 Reference Manual (Chapter 18 – General-Purpose Timers TM9/TM10/TM11)*, and Chapter 14 of the Cady text, 2<sup>nd</sup> edition, which describes the HCS12 timers, or chapter 9 of the Valvano text, which describes the Cortex-M and LM3S/TM4C microcontroller timers. Links to ELEC 2220 lectures on the STM32 microcontroller programmable timers and interrupt operation are available on Prof. Nelson's ELEC 3040/3050 web page. See the *Lab #6: Digital Stopwatch Design with Programmable Timers* lab manual for more information.

### Software Design

Building on your previous projects, design a C program that initializes timer module TIM10 to 1ms, initializes the displayed picture to the symbol chosen for Button 1, then enters a “do-nothing” loop to wait for timer or D-Pad interrupts. LED Display operation is to be as follows:

1. When one of the four arrow keys is pressed, the corresponding arrow should be displayed on the LED display.
2. When Button 1 is pressed, the LED display is to display an image of your choice, or the number '1'.
3. When the 'Reset' button is pressed, the LED display is to toggle between displaying the images in positive or in negative, as shown in Figure 3.

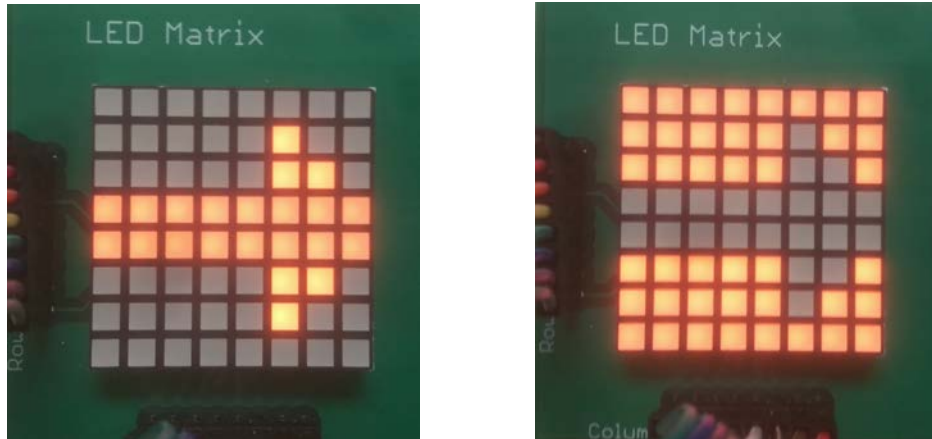


Figure 3: Example of Positive Display (Left) and Negative Display (Right)

### Software Considerations

In designing your software, be sure to take the following into consideration:

1. Refer to the ARM STM32L100RC datasheet to verify the amount of current each GPIO pin can supply and sink. Since the GPIO pins can sink more current than they can supply, it is recommended that you set one Row to logical LOW and all of the columns that need to be turned on in that row to logical HIGH, as this allows the current from each GPIO pin at logical HIGH to light up only one LED and maximize brightness. For reference on how this can be implemented, see Figure 4.

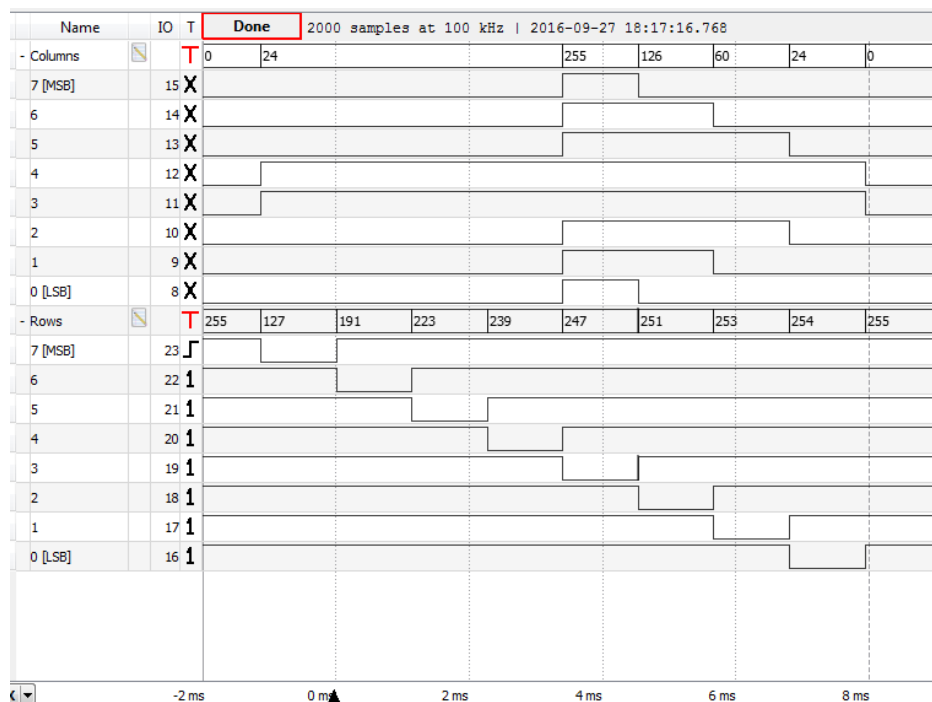


Figure 4: Logic Analyzer Showing Arrow Displayed

In your laboratory notebook, record the following **prior to lab**.

1. Flowcharts for the program and two interrupt service routines (D-pad and timer)
2. Draft program and the two interrupt service routines (or directions to content on H: drive)
3. A plan for testing the display methodology
4. Wiring diagram for connecting the KWM-R30881CUAB LED Matrix to your microcontroller

## Lab Procedure

1. Implement your hardware design, using connections from previous labs. This project requires the D-Pad (PB[7-0] and PA[1]) and 16 pins to control the LED display: PC[15:0]. You may also find it useful to connect additional LEDs to unused GPIO pins to display debugging information.
2. Compile, download, and test your program. To begin, verify that you can still use your D-Pad algorithm to read the correct value. Next, ensure that the LED matrix is connected and will operate the way you intend by connecting the wires to +3V and GND on your microcontroller to light up individual LEDs.
3. Change the period of TIM10 to observe the time-multiplex effect. What happens if you lengthen the period? What happens if you shorten the period?

## Deliverables this Week

Memos are to be submitted this week by 3pm Friday.

The Students will be required to show the GTA the following:

1. LED Matrix displaying an image and changing with button press
2. Oscilloscope measurement of the TIM10 period
3. Logic Analyzer measurement of time-multiplexed operation