



## Module 2 – Interrupts / Alarm Clock

Copyright © 2007-2021, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

### Introduction

Interrupts are very important and useful for embedded systems. An interrupt is an event that triggers the automatic execution of a predetermined piece of code. The code used to handle interrupts is often referred as the Interrupt Service Routine (ISR). The normal or default flow of code in a microcontroller system can be interrupted by different events, for example: timers overflowing, external events, critical errors (such as division by zero or illegal op-codes), power failure, communication events, etc. In this module you will be using timer interrupts in the EFM8 microcontroller in assembly language.

### References

A51 (8051 assembler) user manual included with CrossIDE. ‘A51.pdf’ not only describes how to use the assembler, but also includes a detailed description of all the instructions supported by the 8051 family of microcontrollers.

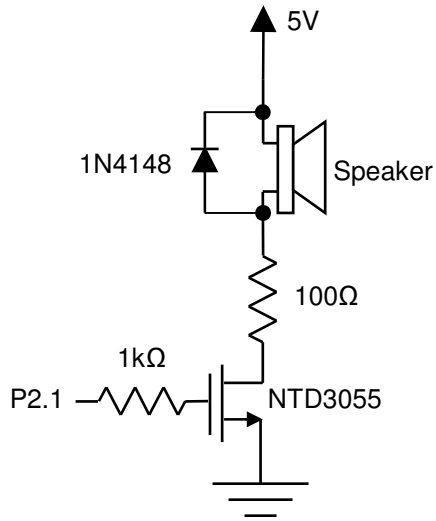
EFM8 Reference Manual. Information about timers and other hardware peripherals in the EFM8 microcontroller can be found in the datasheet.

### Pre-laboratory

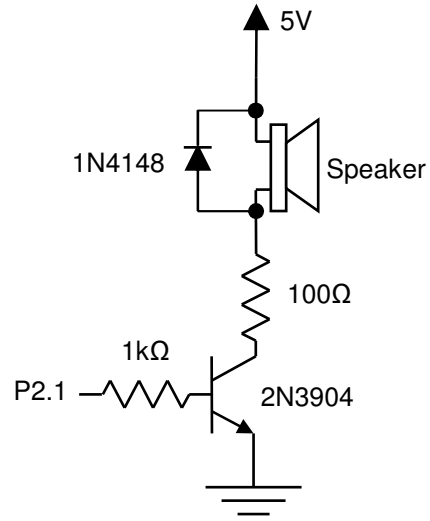
- 1) Sketch the block diagram of timer 1 working as a 16-bit timer.
- 2) Write down the assembly instructions to set and enable timer 1 interrupts using the EFM8 microcontroller.

### Laboratory

1) Provided in the web page for the course is the example “ISR\_example.asm” for the EFM8 board with a LCD attached as in lab #1. This demonstration program carries three ‘concurrent’ tasks: a) Increments (when P0.0=1) or decrements (when P0.0=0) a BCD variable every half second using an ISR for timer 2; b) Generates a 440Hz intermittent square wave at pin P2.1 using an ISR for timer 0; and c) in the ‘main’ loop it displays the variable incremented/decremented using the ISR for timer 2 on the LCD and resets it to zero if the ‘BOOT’ pushbutton connected to pin P3.7 of the EFM8LB1 microcontroller is pressed. Download, compile, and load the program into the EFM8 board and verify its correct operation by attaching the speaker available in your kit to a transistorized driver connected to P2.1 as shown in the figure below. You should hear a ‘beep’ every second.



Speaker driver using a MOSFET



Speaker driver using a BJT

2) Write/compile/run an assembly program for the EFM8 board with LCD for an alarm clock. The alarm clock must display hours (12 hour mode with AM/PM indication), minutes, and seconds, using the LCD. The clock's current time (hours, minutes, seconds, AM/PM), must be settable using push buttons. The clock must also include a settable alarm. When an alarm is trigger, a speaker should produce an alarm sound. Use the speaker available in the ELEC291/292 parts kit for this purpose. Don't forget to add extra functionality and/or features for bonus marks! You must submit a video of your working clock showing all its functionality as well as your assembly code to Canvas.