**Lab 4 Report**

Sage Marks, Matt Krueger

3360:0001 - Embedded Systems

Professor Beichel

University of Iowa, College of Engineering

## Introduction

Embedded Systems Lab 4 introduces a new concept in programming at the hardware level: interrupts. An interrupt is implemented using software to configure a path from an interrupt vector (we used external ATmega328p pins in lab 4) to the processor. This enables the processor to map the incoming signal to its corresponding interrupt service routine in memory. An ISR is simply a subroutine which momentarily stops - or interrupts - the programs current execution to execute the instructions inside of the subroutine jumped to by the interrupt vector.

Lab 4 utilizes this new concept to more easily program the user interface of an embedded system. In previous labs, we have utilized extremely fast polling rates using timers to perceive any change in signals at external pins. The overhead of coordinating timers is drastically reduced using interrupts, not to mention it reduces the latency from a user interaction to an output. We applied interrupts in lab 4 to improve the function of a push button and a rotary pulse generator (RPG) used for a user to control the rpm of a fan using pulse width modulation (PWM).

Listed below is an exhaustive list of hardware required to replicate the lab 4 circuit. Please note that an Arduino Uno contains the microcontroller needed and is sufficient for use in lab. We opted to use the Arduino Uno for simplicity when developing the circuit.

|  |  |  |
| --- | --- | --- |
| Hardware | Quantity | Description |
| Atmega 328P µC | 1 | Programmable µC |
| Enable Low Push Button | 1 | Control on/off of PWM Fan |
| Rotary Pulse Generator | 1 | Control duty cycle of PWM Fan |
| 16x2 LCD Display | 1 | Display fan status |
| EFB0412VHD-SP05 Fan | 1 | PWM fan |
| 100KΩ Resistor | 1 | Button debouncing |
| 10KΩ Resistor | 6 | RPG, button, and V0 voltage divider |
| 1KΩ Resistor | 1 | V0 voltage divider |
| 0.01µF Capacitor | 4 | Decoupling and filtering |

Figure 1: Materials List

## Schematic

A diagram of a computer

AI-generated content may be incorrect.

Figure 2: Electrical circuit schematic created using KiCAD

## 3. Discussion

## 4. Conclusion

Following the brief interruption of the Embedded Systems midterm, we applied our new-found knowledge of interrupts to build a fan monitoring system. Already having learned the lowest level of program flow of using timers to do routine checks on the status of input pins, interrupts were easily picked up. This change in program flow can be treated conceptually as a higher level: functions responding to events. Though we move towards more abstracted logic, the fundamentals of coding remain the same - the processing and transfer of bits. Interrupt.

## 5. Appendix A: Source Code

## 6. Appendix B: References

Beichel, Reinhard. *Embedded Systems Lab 2, ECE:3360. The University of Iowa, 2025* <<https://uiowa.instructure.com/courses/248357/files/29567265?module_item_id=8134634>>

Beichel, Reinard. *Embedded Systems, Rotary Pulse Generators and Lab 3. The University of Iowa, 2025 <*[*https://uiowa.instructure.com/courses/248357/files/29778930?module\_item\_id=8162158*](https://uiowa.instructure.com/courses/248357/files/29778930?module_item_id=8162158) *>*

Beichel, Reinard. *Embedded Systems, Lecture 8: Timers. The University of Iowa, 2025 <*[*https://uiowa.instructure.com/courses/248357/files/29853139?module\_item\_id=8165716*](https://uiowa.instructure.com/courses/248357/files/29853139?module_item_id=8165716) *>*

Delta Electronics. EFB0412VHD-SP05 DC Fan Specification. January 4, 2008. <<https://www.delta-fan.com/Download/Spec/EFB0412VHD-SP05.pdf>>

Microchip Technology. *ATmega328P – 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*. January 2015. <<https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>>

Hitachi. HD44780U (LCD-II) Dot Matrix Liquid Crystal Display Controller/Driver. September 1999. <<https://cdn.sparkfun.com/assets/9/5/f/7/b/HD44780.pdf>>

Pighixxx. *The Definitive Arduino Uno Pinout Diagram*. *May 5, 2013. <*<https://uiowa.instructure.com/courses/248357/files/29320694?module_item_id=8042318>>