Mid-term Review – Part 1

- GPIO Configuration and Use
- Active Low and Active High Signals
- C Language Perspective of a Computer
- State Machine Concepts
- State Machine Implementation in C
- Automatic and Static Variables
- Scope of Variables
- C Coding Standards, Modularity in C
- Nine Potential Sections of a Well-Structured Embedded C Module
- Delay Loop Versus Hardware Tick Timer, Jitter
- Tick Timer Period Tradeoffs
- Serial Port Concepts and Terms
- Synchronous Versus Asynchronous Communications
- Interrupt Concepts and Terms
- Foreground (synchronous) code
- Background (asynchronous) code
- Vector table
- IRQ (interrupt request)
- ISR (interrupt service routine)
- Use of the stack to preserve context of foreground code upon return of ISR
- C Startup Code in startup_stm32l152xe.s

Microcontroller Peripherals and Interfaces

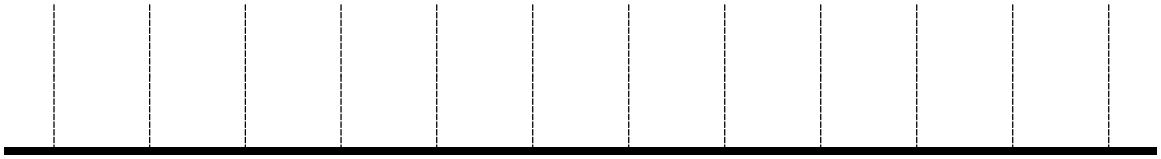
Today we will discuss the following common microcontroller peripherals and interfaces:

- Pulse Width Modulation (PWM)
- Inter-Integrated Circuit (I2C) synchronous serial interface
- Serial Peripheral Interface (SPI) synchronous serial interface
- Analog-to-Digital Converter (ADC)
- Digital-to-Analog Converter (DAC)
- Switch debouncing

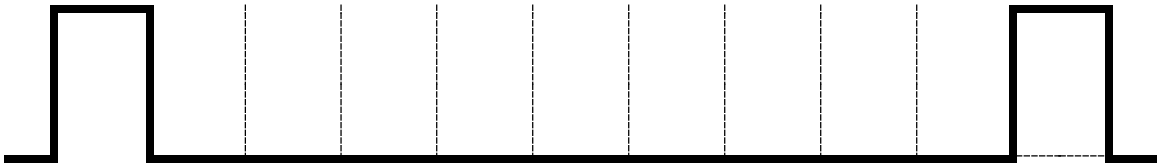
Pulse Width Modulation (PWM) Concepts

Pulse width modulation (PWM) is a method of digitally controlling the amplitude of a peripheral such as the brightness of an LED, the speed of a motor, or the loudness of a speaker. As the name suggests, the width of a periodic pulse is varied (this is referred to as the duty cycle of the pulse, in percent) to provide a smaller or larger average stimulus (voltage or current) to the peripheral. Here are some examples:

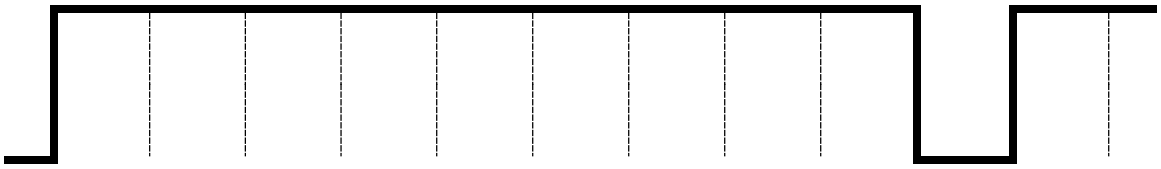
0% Duty Cycle Pulse Width Modulation (Active-High)



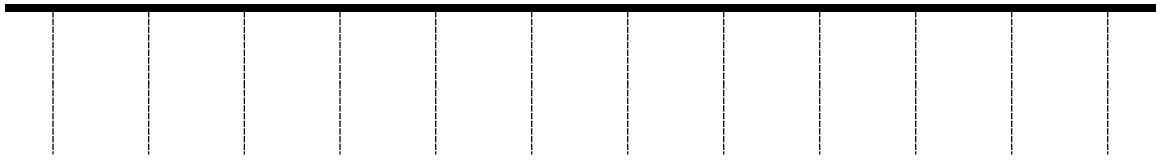
10% Duty Cycle Pulse Width Modulation (Active-High)



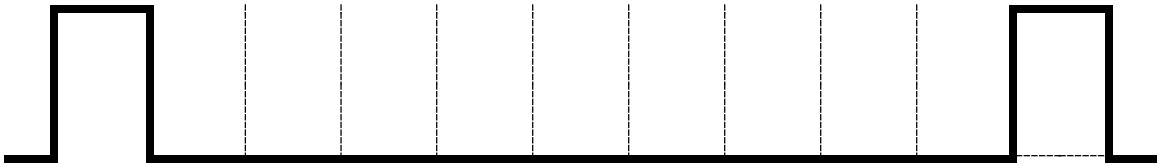
90% Duty Cycle Pulse Width Modulation (Active-High)



100% Duty Cycle Pulse Width Modulation (Active-High)



90% Duty Cycle Pulse Width Modulation (Active-Low)



I2C Signals and Concepts

- SCL (Serial Clock)
- SDA (Serial Data)
- All peripherals on an I2C bus have unique addresses

SPI Signals and Concepts

- CS (Chip Select)
- SCK (Serial Clock)
- MOSI (Master Out, Slave In)
- MISO (Master In, Slave Out)
- All peripherals on a SPI bus have unique chip select lines

ADC Signals and Concepts

- AIN (Analog Input)
- Sample rate, resolution, accuracy, monotonicity, settling time

DAC Signals and Concepts

- AOUT (Analog Output)
- Sample rate, resolution, accuracy, monotonicity, settling time

Switch Debouncing

- Mechanical switches do not produce a clean, single transition from open to closed
- Switch bounce can persist for up to 100 msec in some switches
- Debouncing requires a state machine that waits for a transition of the GPIO input connected to the switch, delays for the debounce interval, then checks to see if the transition has persisted

Class Notes