TI-RSLK MAX

Texas Instruments Robotics System Learning Kit

Texas Instruments

Module 9

Introduction: SysTick Timer

Introduction: SysTick Timer

Educational Objectives:

LEARN SysTick timer fundamentals
USE SysTick to generate accurate time delays
LEARN how to measure pulse times, and period with a logic analyzer
LEARN how to measure amplitude and period with an oscilloscope
CREATE an analog low pass filter (LPF) using an RC circuit
USE PWM and an LPF to create a digital to analog converter (DAC)
DESIGN, TEST & DEBUG A SYSTEM
Control the brightness of an LED using PWM

Prerequisites (Modules 5, 7, and 8)

- Voltage, current, resistor, capacitor (Module 5)
- Microcontroller GPIO (Module 7)
- Switch and LED interfaces (Module 8)

Recommended reading materials for students:

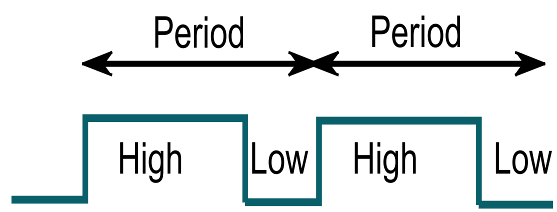
- Chapter 9, **Embedded Systems: Introduction to Robotics,** Jonathan W. Valvano, ISBN: 9781074544300, copyright © 2019

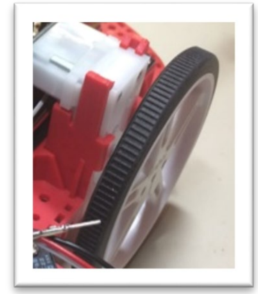
Time is an important parameter for an embedded system. As an **input**, measuring time includes measuring frequency, period or pulse width. For example, in the GPIO module, we saw the optical reflectance of the line sensor translated to a voltage versus time response, and the microcontroller converted this sensor data into digital form by measuring the length of time it took for the response to change from a logic high to a logic low.

As an **output**, the microcontroller will create signals that affect its environment. In the **8. Switches and LED** module we needed to manage time in order to oscillate the LED at 5 Hz. In this module, we introduce **pulse width modulation** (PWM), which is a method using time to deliver an adjustable power to a device. With PWM, the software generates a digital output of fixed frequency. Let *Period* be the fixed period of this digital wave, let *High* be the time the signal is high, and let *Low* be the time the signal is low. Typically, when the signal is high, power is applied to the external device. The software adjusts the *High* and *Low* times, such that *Period=High+Low* is fixed. In many systems, the delivered power is linearly proportional to the **duty cycle**:

$$Duty\ cycle = \frac{High}{Period} = \frac{High}{High+Low}$$

Since *Period* is fixed, power will be linearly proportional to the software parameter *High*.





In the lab associated with this module, we will use PWM to dim the brightness of an LED. By passing the PWM output to an **analog low pass filter**, with one resistor and one capacitor, we can create a digital to analog converter (**DAC**). Using the RC filter at this point is a good way to explain how motors respond to the PWM wave. In a future module (**12. DC Motors**), our system uses a software generated PWM to set the power delivered to the robot motors. PWM generation is so important to embedded systems, we will show you in the **13. Timers** module how to create multiple PWM waveforms off-loading the waveform generation into hardware. In this approach, the software still sets the duty cycle and the period, but the timer hardware does the work of generating the digital waves.

ti.com/rslk

IMPORTANT NOTICE AND DISCLAIMER