



# Module 8

Introduction: Interfacing input and output



# Introduction: Interfacing input and output

## Educational Objectives:

**LEARN** Switch & LED fundamentals,

**BUILD** interface circuits for switches and LEDs with TI's LaunchPad development board

**WRITE CODE** to configure switches as inputs and LEDs as outputs

**DESIGN, TEST & DEBUG A SYSTEM**

"Window Intruder Detect Security System"

### Prerequisites ( Modules 1, 2,3, 4, 6)

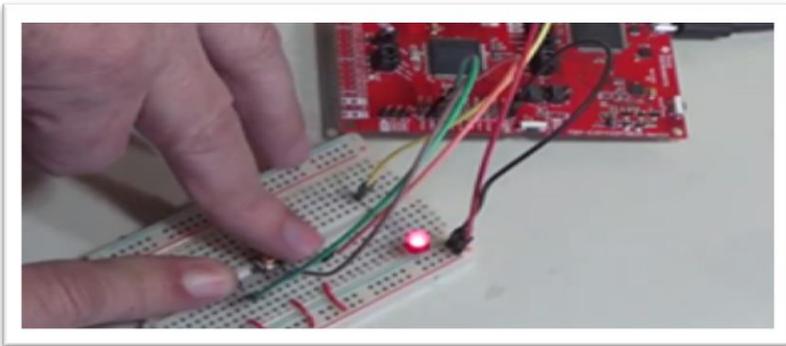
- Basic circuits, Ohm's Law ( Module 2)
- Running code on the Launchpad using CCS ( Module 1,4)
- GPIO ( Module 6)

### Recommended reading materials for students:

- Volume 1 Section 2.7, 4.1, 4.2.2, 4.3, and 4.6  
Embedded Systems: Introduction to the MSP432 Microcontroller  
ISBN: 978-1512185676, Jonathan Valvano, copyright (c) 2017

or

- Volume 2 Section 2.4, and 2.6  
Embedded Systems: Real-Time Interfacing to the MSP432  
Microcontroller, ISBN: 978-1514676585, Jonathan Valvano,  
copyright (c) 2017



The design, development and debugging of a robotic system involves many tasks. Building of any system involves knowledge of basic components and how they work and connect with other components commonly termed as **interfacing**. In fact, we can divide all of engineering into sub-systems and interfaces. Interfaces are used to combine multiple **sub-systems** together to form a more complex system. In many examples of interfacing with a microcontroller we use input devices to feed data into a computer and output devices to allow the computer to affect its surroundings.

Another name for sub-system is **module**, which can be a software module or hardware module. Another name for a software module is **device driver**. In order to build large systems, we need a method to manage complexity. Breaking a large system into modules, which are in turn broken into smaller modules, is the standard approach to dealing with complexity. There are two aspects of a module: what it does and how it works. Modular design provides an abstraction that allows us to separate what a device does from how it works. For example, consider the red LED on your LaunchPad. There are two prototypes in the **LaunchPad.h** header file that describe what this module does:

```
void LaunchPad_Init(void);  
void LaunchPad_LED(uint8_t data);
```

How the module works can be found in **the LaunchPad.c** code file. Good modularity maximizes the number of modules while minimizing the **coupling** between modules. One quantitative measure of coupling is **bandwidth**, or the amount of data flowing from one module to another.

Again, to connect modules together we need an **interface**. In the software world, one module is connected to another by the public functions that can be called. This means the header files in C define the interconnection between software modules. In the hardware world, physical devices, e.g., electrical, mechanical, chemical, biological, allow modules to be interconnected. Furthermore, when connecting software modules to hardware devices, we use a combination of software and hardware components to affect the connection. For example, the IR sensor is a hardware device that uses optics to measure distance. We will use optical devices (e.g., the sensor), electrical circuits (e.g., the analog to digital converter (ADC), and software (e.g., the ADC routines) to connect the sensor to the robotic software running on the microcontroller.



## Introduction: Interfacing input and output

A common example of an input device is the **switch**. Engineers use switches in a myriad of applications touching every industry such as aerospace, automotive, chemical, communication, marine, medical, military, petrochemical, and transportation. Switches are also found in the devices we use in our homes every day. We use switches in nearly all electrical and mechanical products.

We can categorize switches as:

- Momentary pushbutton
- Rotary
- Slide switch, and
- Toggle switch.

The state of the switch is either an “OPEN” or a “CLOSE”, which can be read as binary information by a microcontroller. A typical switch has a 100-M $\Omega$  resistance when “**not pressed**” and has a 0.1- $\Omega$  resistance when “**pressed**”. We can interface switches with either positive logic or negative logic. Most commonly, the switches require the use of internal pull-up resistors for negative logic switches and internal pull-down resistors for positive logic switch interfaces. The pull-up and pull-down functions are enabled by software during initialization.

A simple example of an output device is the light emitting diode Or LED. Like switches, LEDs are binary devices, in that they can be either “ON” or “OFF”. The software controls the state of LED explicitly by calling **the LaunchPad\_LED** function with a 1 or a 0. This learning module will use LEDs to report binary diagnostic information. However, we can find LED interfacing in many applications, such as optical cables, solid-state relays, digital isolation barriers, and infrared transmitters. LEDs have a nonlinear voltage current relationship. Interfacing an LED requires understanding of Ohm’s Law in resistors.

Interfacing switches and LEDs to the microcontroller is an appropriate place to start because the process is simple and the proper behavior is obvious to the learner. However, wrapped into the simple activity of connecting switches and LEDs to the microcontroller, we can expose our students to the fundamental processes of design, assembly, coding, and testing. As part of the lab, students will design **a window intruder detect system** the knowledge they gained in this module. Ultimately, the robot will use similar switches to detect an object. The students will use LEDs to provide visualizing of where and what the robot software is doing and also help with debugging, as the robotic system is put together to accomplish the planned task of solving the maze.

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2018, Texas Instruments Incorporated