arm Education

Embedded Systems Essentials with Arm: Getting Started

Module 3

KV2 (3): GPIOS and Peripherals

A general purpose input-output, or GPIO, is a generic digital signal pin. It has no set purpose - its function is defined by the board's programmer or designer at run time.

The term "general purpose" refers to the fact that a GPIO can be configured to interface a whole range of complex digital and analog I/O signals and protocols.

Each GPIO port pin is connected to a configurable internal interface circuit. The connection between the interface circuit and the external circuit is a pin or a pad. The "function select" block allows us to configure the GPIO to interface with a range of different devices. Various properties can be configured using the control registers, and various types of data can either be written out or read in. There may also be analog paths for analog to digital conversion.

This approach allows a few GPIO pins to serve a wide range of functions, saving space and improving flexibility.

In real-life applications, GPIOs control 'switches,' such as transistors, to disable or enable the power supply for other components. GPIOs can only provide two voltage levels. If more than two are required, an analog input or output, or the trick of a pulse width modulation (PWM), must be used.

When utilizing GPIO, a microcontroller usually provides voltages at the dedicated levels of 0 and its fixed power supply, known as vdd. Until recently, vdd was usually 5 volts; however, today's ultra-low power processors have reduced this voltage to as low as 1.8 volts. This may require adding a level converter to shift the voltage levels produced by the processor to reach the needed levels.

Let's consider how we would use a GPIO to light up one of two LEDs.

This circuit diagram shows how we could connect one switch and two LEDs to our Cortex processor. Note the use of additional resistors and the fact that the red LED needs a different resistor from the blue LED.

To calculate the value for these resistors we can use Ohm's law. We will use the formula R = (VDD – VLED) / ILED. (R equals VDD minus VLED over ILED)

- R is the value of the resistor that we are trying to find. VDD is the source of current and we will set it as equal to 3 volts.
- VLED is the voltage required by the LED and depends on the type of LED, in this case red equals about 1.8 volts and blue equals about 2.7 volts.
- ILED is the current we want to flow through the resistor, which in this case will be 4 milliamps or 0.004 amperes.

Plugging these numbers into the equation will give us a resistor value of 300 ohms for the red LED and 75 ohms for the blue LED.

Our program can determine if the input signal is 1 or 0, and can set the output to 1 or 0. In one position, the processor will light the red LED, and in the other position, the processor will light the blue LED.

We'll use one GPIO as an input, to read the signal level from the switch, and two further GPIOs as outputs, to drive the red and blue LEDs.

Pull-up and pull-down resistors are typically used in combination with switches. You can see we've done this for the GPIO switches in our circuit design.

The purpose of pull-up and pull-down resistors is to ensure that there is a default or "known" value on the input pin. This prevents an unknown state called "floating". "Floating" is a situation in which there is no input and the program cannot decide whether the switch should be high or low.

In this diagram, if we want the switch SW1 (IN) to pull the pin to the ground, we would use the pull-up resistor layout.