GPIO
General Purpose Input/Output

Gary J. Minden
September 13, 2017
Outline

• A little electronics
• General Purpose Input/Output (GPIO)
• Example
Energy and Circuits

\[ I \times R = V \]

(Ohm’s Law)
GPIO Simplified Circuit

To/From uC

Data Out

Output Buffer

Enb

Pin (Pad)

Data In

Input Buffer

© G. J. Minden 2013
LED Circuit (Blinky)

GPIO PortG<2>

Data Out

Load

V = 3.3 VDC
I = 0.006 A = 6 mA

Write ‘1’ to PortG<2>… LED is on
Write ‘0’ to PortG<2>… LED is off
Energy and Power

- **Energy** is a quantity… How much gas do you have in your car gas tank?
- **Power** is how fast you use energy… How fast do you drive?
- \( P = I \times V \) … Power is measured in **Watts**
- \( E_n = P \times T \) … Energy is Power integrated over time
  - You purchase \( E_n \) from the power company
  - However, the power company needs max( P ) capacity to meet maximum load, otherwise voltage drops and you have brown-outs
Battery Energy — AA Alkaline-Manganese Dioxide

- Nominal voltage: 1.5 V
- Impedance: 81 m-ohm @ 1 kHz
- Typical weight: 24 g (0.8 oz)
- Typical volume: 8.4 cm³ (0.5 in³)
- Terminals: Flat
- Storage temperature range: 5°C to 30°C (41°F to 86°F)
- Operating temperature range: -20°C to 54°C (-4°F to 130°F)
GPIO Electronic Connections

LED Output
0 (L) -- LED Off
1 (H) -- LED On
R6 limits current
$V_{LED} \approx 1.4 \text{ V}$

Switch Input
Open -- 1 (H)
Pressed -- 0 (L)
Pull-up in GPIO

Speaker Output
GPIO

- A basic hardware mechanism to move digital signals from external, physical devices to your program (sensor) and/or move data from your program to external, physical digital signals (actuator)
- Organized as 8-bit ports, e.g. PortA, PortB, ..., PortH
- Each bit of a port can be individually programmed to be an input bit or an output bit
- If an output, the current (I) drive is programmable
  - 2 mA, 4 mA, and 8 mA current drives are possible
- If an input, the input is Schmitt-triggered, 5 V tolerant
- A pin can be programmed to be neutral, weakly pulled-up, or weakly pulled-down
- The physical pins can be programmed to support different functions
  - For example, PortA<0> can be a GPIO function or a serial receiver function
GPIO

- The following documents describing GPIO are available from:
  http://www.ittc.ku.edu/~gminden/Embedded_Systems/Content/Datasheets.html
- The hardware GPIO is described in Chapter 10 of the TI Tiva TM4C1294 Microcontroller Datasheet
  - See: http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/TI_TM4C1294NCPDT.pdf
- The GPIO software library is described in Chapter 15 of the Tiva Peripheral Driver Library User Guide
  - See: http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/TI_TIVA_DriverLib_UG-2.1.0.12573.pdf
- The “external use” of a GPIO pin is determined by the external system design
GPIO Simplified Circuit

To/From uC

Data Out

Output Buffer (Driver)

Enb

Pin (Pad)

Input Buffer (Receiver)

Data In
Simplified GPIO Hardware Structure

- The “Physical Pin” is as it name implies
- A “Driver” and “Receiver” are circuits connected to the “Physical Pin”
- The “Driver” is controlled by a “Direction Register” (input or output) and a current drive level
  - If the pin is an input pin, the driver is disabled
  - The current drive level (“I”) can be 2 mA, 4 mA, or 8 mA
- The “Receiver” is a Schmitt-trigger
- The output signal can be selected from the port data register or an alternative hardware function. This is controlled by the “Fcn Select” (Function Select) register
- The input signal is distributed to the GPIO port and the alternative hardware function
Basic GPIO Programming Steps

• The basic steps to configuring a GPIO pin are:
  • Enable (power-up) the hardware component
  • Configure the GPIO pin to be an input pin or an output pin
  • Configure the GPIO pin for current level and pull-up/pull-down

• The basic steps for using a GPIO pin as an output are illustrated on next slide

• The basic steps to read and write values from/to a GPIO pin are illustrated on the “next next” slide

• The subroutines used in this example are documented in the Stellaris DriverLib User Manual
Blinky Initialization

/*--Task_Blinky.c
 *
  *  Author: Gary J. Minden
  *  Organization: KU/EECS/EECS 388
  *  Date: February 22, 2016
  *
  *  Description: Blinks LED D1 (PortN<1>) on Tiva TMC41294 Evaluation board
  *
  */

#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "inc/hw_uart.h"

#include <stddef.h>
#include <stdbool.h>
#include <stdint.h>
#include <stdarg.h>

#include "driverlib/sysctl.h"
#include "driverlib/pin_map.h"
#include "driverlib/gpio.h"

#include "FreeRTOS.h"
#include "task.h"