Experiment 3: More GPIO Registers and Digital Input

i

REVISION HISTORY							
NUMBER	DATE	DESCRIPTION	NAME				
141	Sep. 19, 2014		н				

Contents

1 Objectives										
2	Parts List									
3	Background	1								
	3.1 Digital Input	. 1								
	3.2 GPIO, Revisited	. 1								
	3.2.1 FIOPIN	. 1								
	3.2.2 FIOMASK	. 2								
	3.3 Pull-up/Pull-down Resistors	. 2								
	3.3.1 PINMODE	. 2								
4	Tasks	3								
	4.1 Modify Experiment 2	. 3								
	4.2 Two Switches, One LED	. 3								

1 Objectives

- · Connecting external digital input devices
- Introducing additional useful GPIO registers
- Understanding pull-up/down resistors

2 Parts List

- LPC1769 LPCXpresso board
- USB A-Type to Mini-B cable
- Breadboard
- LEDs
- · Push-buttons, toggle-switches, and switches
- 330-Ohm Resistors
- Jumper wires

3 Background

3.1 Digital Input

A GPIO pins can be configured to act as GPI (general-purpose input) by setting the corresponding bit in the FIODIR register to 0. A digital input pin is "digital" because it is *driven* by an external digital device that has only two states (HIGH or LOW); and it is "input" because its state is *read* by the microcontroller. That implies that some external device/circuit is needed to generate that digital input value (HIGH or LOW).

Examples for simple digital input devices include switches and push-buttons.

3.2 GPIO, Revisited

In addition to the three registers used in the previous experiment, there are few additional GPIO-related ones. The one that is particularly essential for reading from a digital input peripheral is FIOPIN.

3.2.1 FIOPIN

This register is a R/W register that stores the current state of a port's pins. In other words, you can write to FIOPIN to set and clear pins of a specific port. You can also read the state of port pins. FIOPEN is essential for the read operation, but since it is a R/W register, it can also be used with output pins. For instance, you can redo Experiment 2 using FIOPIN only instead of FIOSET and FIOCLR.

There is an FIOPIN register for each one of the five I/O ports, and it can be accessed in the same way FIOSET and FIOCLR are. For example, port 1's FIOPIN register can be accessed using:

LPC_GPIO1->FIOPIN

3.2.2 FIOMASK

If a bit in this register is 1, the corresponding port pin will not be affected by FIOPIN, FIOSET, or FIOCLR. The default is all `0`s, implying nothing is masked.

For example, you can mask all bits in port 0 except bits 7 and 8 using: LPC_GPIO0->FIOMASK = 0xFFFFFE7F;

3.3 Pull-up/Pull-down Resistors

It is possible for an input peripheral to have an open circuit. For example, a simple push-button is an open circuit when it is not pressed. In this case, an input pin connected to such switch would be floating, leading to an ambiguous pin state.

The solution is to have a pull-up resistor. The figure below shows how a pull-up resistor is used.



Figure 1: Pull-up resistor illustrated

Note

In the above schematic, if the connections to 3.3 volts and GND are swapped, the resistor would be called a *pull-down* resistor; in which case the input would be HIGH when the switch is closed and LOW when the switch is open.

Тір

In the Cortex family, a programmable pull up/down resistor is built in. The default is *pull-up*, but you can program it to be *pull-down*. You can also disable it and use an external resistor as discussed above.

3.3.1 PINMODE

The PINMODE registers are used to control the built-in pull-up/down resistors. There are 10 PINMODE registers, named: PINM OD0, PINMOD1, ..., PINMOD9. In LPC17xx.h, these registers are fields of the LPC_PINCON structure. It takes 2 bits in a PINMODE register to select the mode for a single pin. For example, the PINMOD0 register controls the mode for the pins P0.0 to P0.15. The 2 bits select one of the following 4 modes:

3	/	3

00	Pin has an on-chip pull-up resistor enabled.
11	Pin has an on-chip pull-down resistor enabled.
10	Pin has neither pull-up nor pull-down resistors enabled.
01	Repeater mode: the pin to retain its last known state and is not driven externally.

Examples

LPC_PINCON->PINMODE0	&=	~(0x3);		11	P0.0	has	an	on-ch	ip	pull-up	resistor		
LPC PINCON->PINMODE0	=	(1<<4)	1	(1<	<5);	11	P0.2	has	an	on-chip	pull-down	resisto	r

4 Tasks

4.1 Modify Experiment 2

- 1. Redo Experiment 2 without using FOIOSET and FIOCLR!
- 2. Use a switch to start and stop the scrolling.

4.2 Two Switches, One LED

1. Build a system in which a single LED is controlled by two switches. You should be able to switch the LED ON or OFF using either switch.



Caution A common mistake is to forget about or misuse the ${\tt FIODIR}$ register.