Clock Control

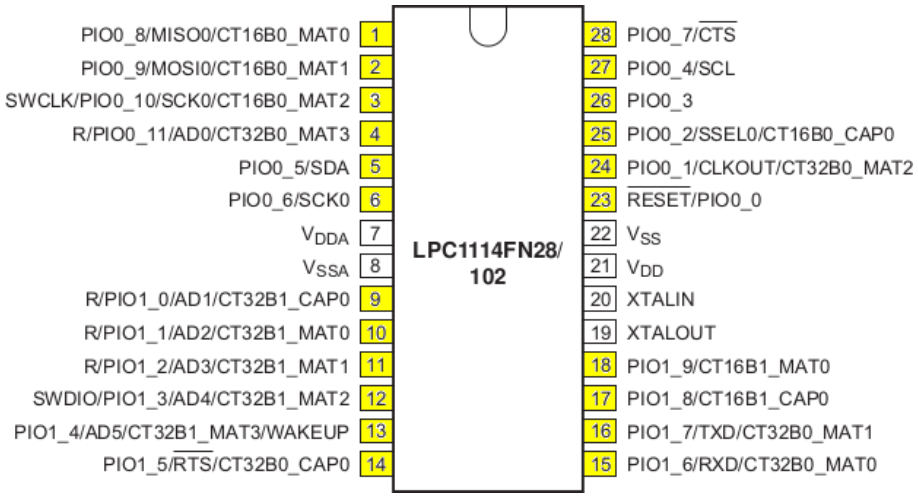
Power management

Parallel digital IO

Key terms: Ports, Direction registers, alternative functions, pull-up resistors.

The LCP1114 uses memory mapped I/O. From a programmer's perspective, the effect of this is that writing to particular memory locations changes the voltage of pins on the side of the microcontroller chip. These memory locations are called *output ports*. Furthermore, read operations from certain memory locations return the states (whether the voltage is high or low) of pins on the side of the chip. These memory locations are called *input ports*.

The LPC1114 family of microcontrollers supports up to 4 ports numbered 0 to 3. Each of these ports can have up to 12 I/O pins many of which can also have other functions. The particular version of the LPC1114 we use in this course is the LPC1114FN28/102 and this has 22 digital I/O pins of this type.



Input/Output pins on the LPC1114FN12/102. I/O pins are yellow.

As you can see from the diagram above, most of the I/O pins have alternative functions. The default function is the first one listed. For example, if we look at pin 25 we see that the default function of this pin is to behave as a digital I/O pin; in particular, port 0, bit 2. A secondary function of this pin is a Slave Select Output for Synchronous Serial Interface 0. A tertiary function for this is an event capture input for 16-bit counter/timer 0. The pin is rewired to the various internal subsystems by changing bits in a particular memory location.

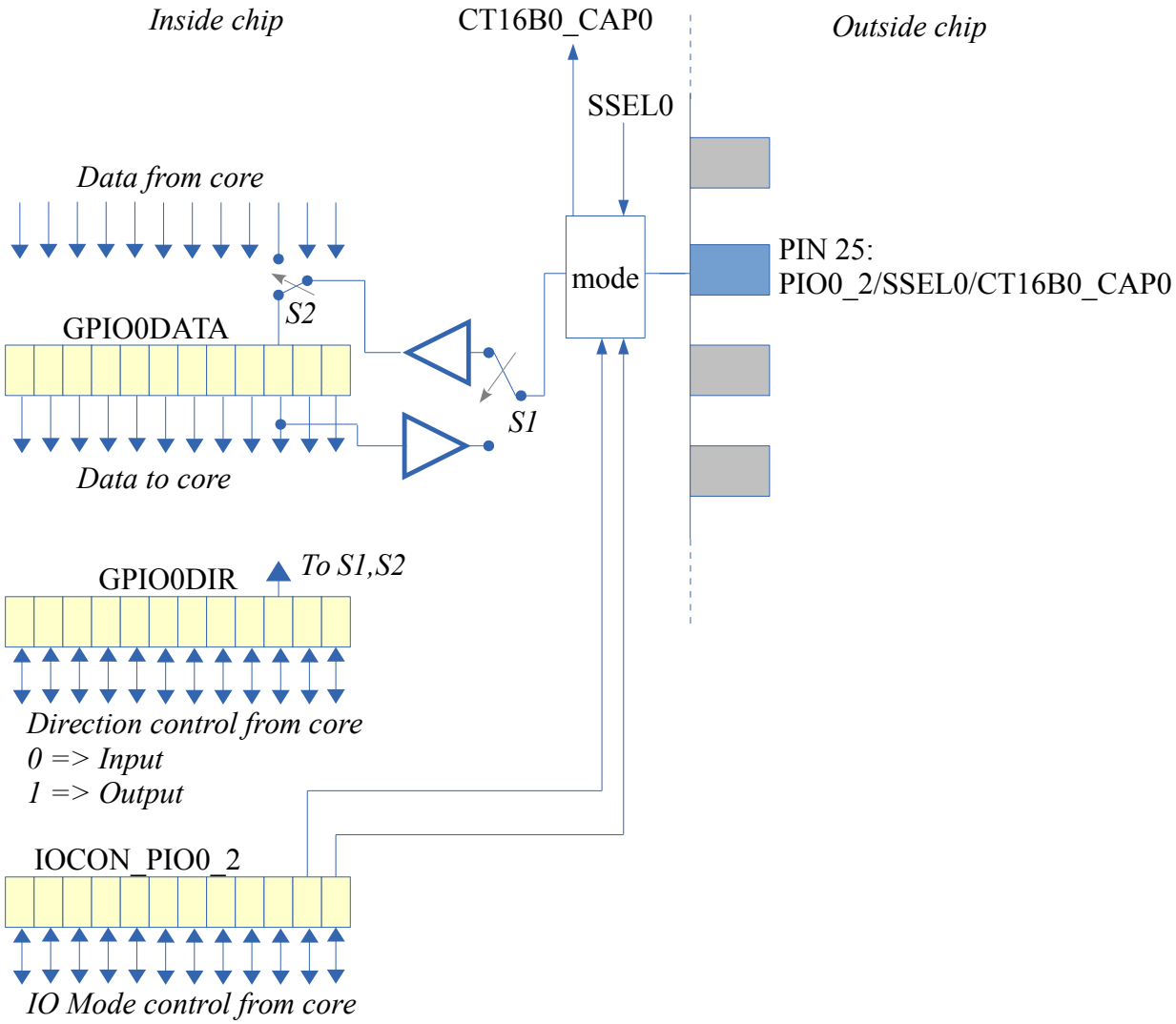
If a pin on the chip is configured as a digital input/output pin it can be further set up as an input pin or an output pin. This “directionality” is determined by a data direction register. Each digital I/O bit has a corresponding bit in a data direction register. If a direction register bit is a 0 then the corresponding digital I/O bit is an input, otherwise its an output.

The figure below shows (almost) the full equivalent circuit for Pin 25. If the register IOCON_PIO0_2 is zero, the default function for the pin is selected (a digital I/O pin). In this configuration, the direction register (GPIO0DIR) is used to configure the pin as an output or an input.

If the pin is an input, a voltage above about 1.5V on Pin 25 causes a logic '1' to appear in bit 2 of GPIO0DATA.

If the pin is an output, a logic '1' in GPIO0DATA bit 2 causes 3V (approx) to appear on pin 25. A value of 0 in this bit location produces a pin voltage of 0V (approx).

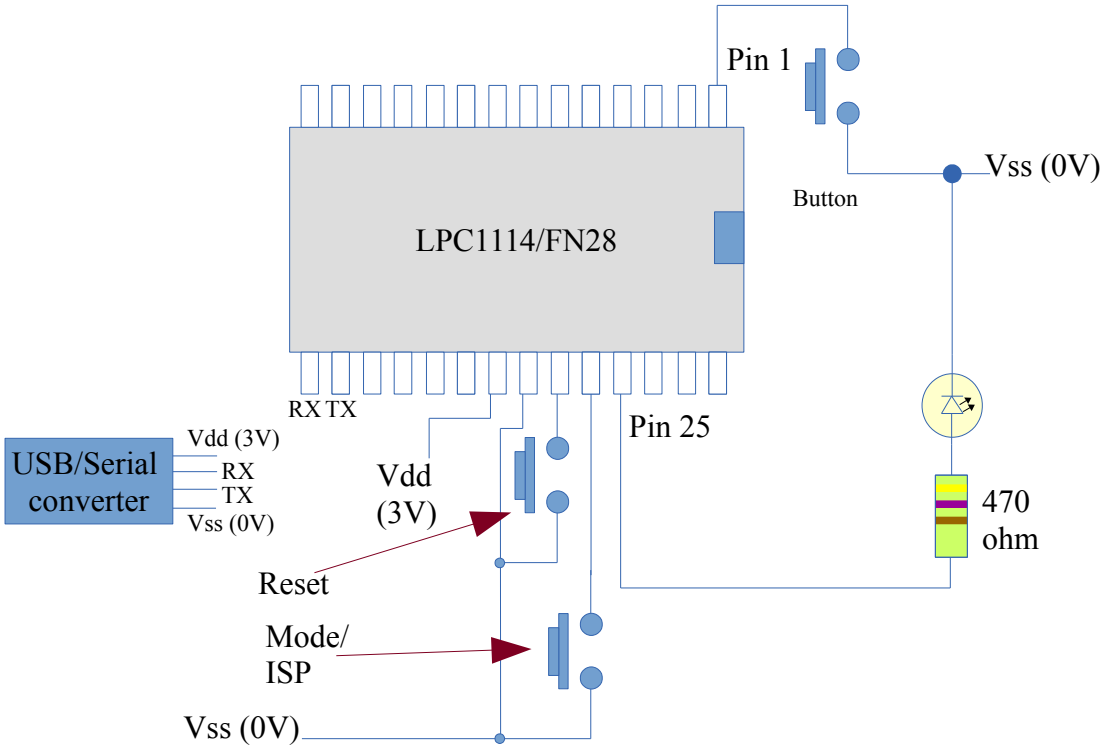
General Purpose Input Output (GPIO) pins such as pin 25 are used to turn devices on and off and to sense whether buttons have been pressed, liquid levels reached etc. in the area of automation and control.



Configuration systems for Pin 25 on the LPC1114FN28/102

Lighting an LED and sensing a button press.

The “Hello World” equivalent program in the embedded world is often called “Blinky”. The idea is that the program make an LED flash on and off repeatedly. A further refinement of this program also adds in button sensing. The figure below shows the wiring for such a setup. It also includes the wiring for the programming interface for the LPC1114.



*Wiring for lighting an LED and sensing a button press. Note: there is an internal **pull-up resistor** on Pin 1*

C-Code for above:

```
/* Blinky with a button.
  An LED is attached to Pin 25 and a button is attached to Pin 1
  When the button is NOT pressed, the LED should blink
*/
#include "lpc111x.h"
void delay(unsigned len)
{  // Software delay function
      while(len--);
}
void ConfigPins()
{
      SYSAHBCLKCTRL |= BIT6 + BIT16; // Turn on clock for GPIO and IOCON
      IOCON_PIO0_2 &= ~(BIT1+BIT0);  // ensure Pin 25 behaves as GPIO
```

```
	GPIO0DIR |= BIT2; // Make Pin 25 an output
	GPIO0DIR &= ~BIT8;	// Make Pin 0 an input
	GPIO0DATA = 0;		// 0 output initially
}
int main()
{
	ConfigPins();
	while(1)
	{
		if (GPIO0DATA & BIT8) // if the button is NOT pressed
		{
			GPIO0DATA ^= BIT2; // Toggle (XOR) LED
			delay(1000000);	// Wait
		}
	}
}
```