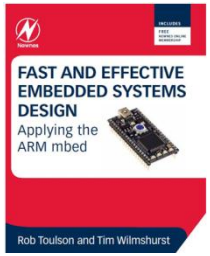
Embedded Systems Design Course

Applying the mbed microcontroller

Digital input and output

These course notes are written by R.Toulson (Anglia Ruskin University) and T.Wilmshurst (University of Derby). (c) ARM 2012

These course notes accompany the textbook “Fast and effective embedded system design : Applying the ARM mbed”



Digital input and output

- Introduction to digital terminology
- Digital outputs on the mbed
- Using LEDs on the mbed pins
- Connecting switches to the mbed
- Implementing a digital switch input on the mbed

Introduction to digital terminology

- The mbed uses a power rail of 3.3 Volts - 0 Volts indicates ‘off’ and 3.3 Volts indicates ‘on’.
- A number of terms are used interchangeably to describe on and off in digital systems:

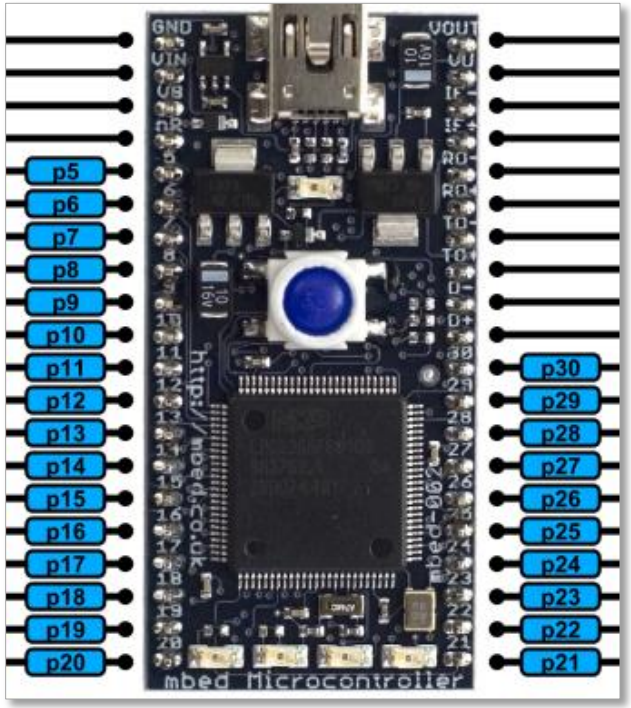
0V	3.3V
Open	Closed
Off	On
Low	High
Clear	Set
logic 0	logic 1
False	True

- Note: terms ‘logic 0’ and ‘logic 1’ may be simply referred to as ‘0’ and ‘1’

Digital outputs on the mbed

- On the mbed, the four on-board LEDs are digital outputs which have been specially configured to operate with no extra wires or connections needed.

- The mbed also has 26 digital IO pins (pins 5-30) which can be configured as inputs or outputs.



Digital outputs on the mbed

The available library functions are shown in the table below.

DigitalOut	**A digital output, used for setting the state of a pin**
Functions	**Usage**
DigitalOut	Create a DigitalOut connected to the specified pin
write	Set the output, specified as 0 or 1 (int)
read	Return the output setting, represented as 0 or 1 (int)
operator=	A shorthand for write
operator int()	A shorthand for read

Digital outputs on the mbed

- The digital IO pins are configured by defining them at the start of the program code.
- Each digital IO is given a name and associated pin, for example:

```
DigitalOut myname1(p5);
DigitalOut myname2(p6);
DigitalOut myname3(p7);
etc…
```

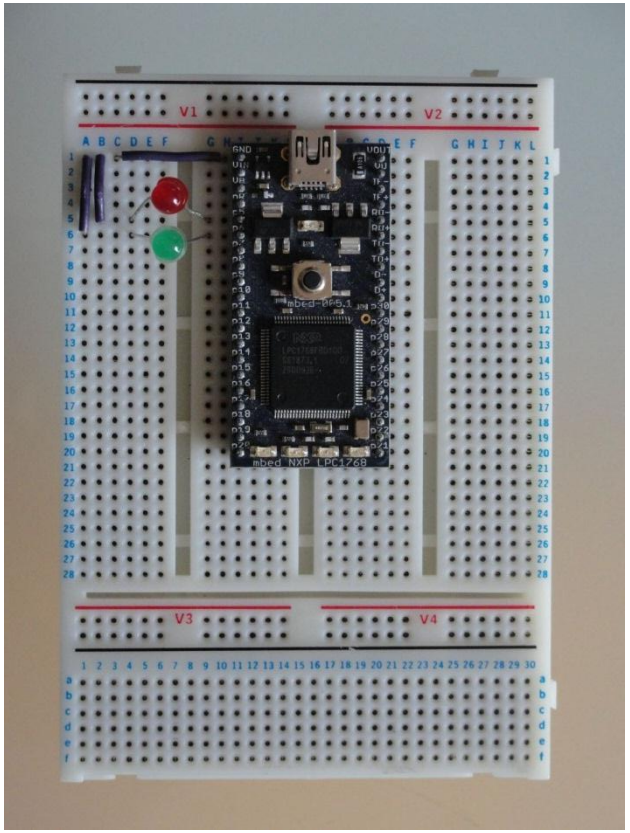
- The DigitalOut interface can be used to set the state of the output pin, and also read back the current output state.
- Set the DigitalOut to 0 to turn it off, or 1 to turn it on.

Using LEDs on the mbed pins

- Connect the mbed to a breadboard and attach a red LED to pin 5 and a green LED to pin 6.

- Remember to attach the positive side of the led (the side with the longer leg) to the mbed. The negative side should be connected to ground.

- The mbed has a common ground on pin 1.



Using LEDs on the mbed pins

- Exercise 1: Create a new program for the external LED project.
- Modify the default main.cpp code to become the following:

```
#include "mbed.h"
DigitalOut redled(p5);
DigitalOut greenled(p6);
int main() {
    while(1) {
        redled = 1;
        greenled = 0;
        wait(0.2);
        redled = 0;
        greenled = 1;
        wait(0.2);
    }
}
```

- Compile, download and run the code on the mbed.

Using LEDs on the mbed pins

- Look at the example program and identify the key C programming elements as follows:
  - The mbed.h library file is linked to by the `#include' statement.
  - DigitalOut objects are defined with a name and a chosen mbed pin.
  - The main program function exists inside `int main() { ... program... }'.
  - An infinite loop is implemented by the while(1) statement, so that the program continuously loops forever, allowing the led to flash continuously.
  - Digital outputs are controlled simply by setting the relevant objects equal to 0 or 1.
  - The mbed wait() function is used for timing control.

Digital inputs on the mbed

- Digital inputs values can be read.
- As with digital outputs, the same 26 pins (pins 5-30) can be configured as digital inputs, as follows:

```
DigitalIn myname1(p5);
DigitalIn myname2(p6);
DigitalIn myname3(p7);
```

- The DigitalIn Interface determines the current logical state of the chosen input pin, e.g. logic '0' or logic '1'.
- Zero volts on a digital input pin returns a logical 0, whereas 3.3 Volts returns a logic 1.

Digital inputs on the mbed

The available library functions are shown in the Table below.

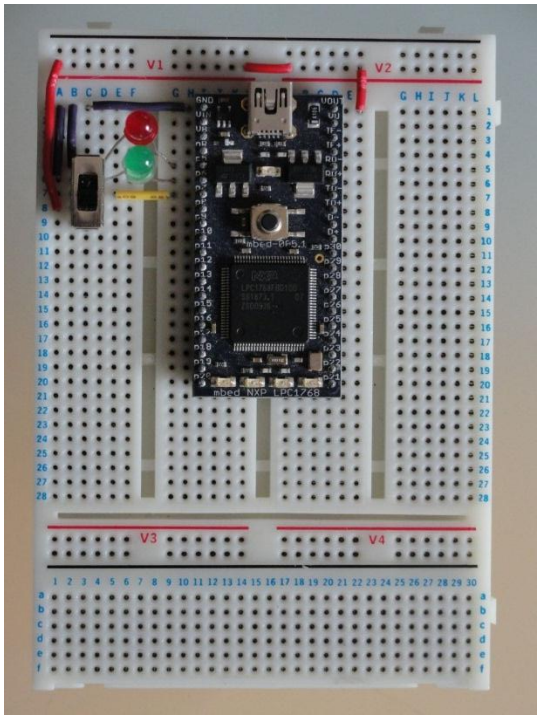
DigitalIn	**A digital input, used for reading the state of a pin**
Functions	**Usage**
DigitalIn	Create a DigitalIn connected to the specified pin
read	Read the input, represented as 0 or 1 (int)
mode	Set the input pin mode
operator int()	A shorthand for read

Connecting switches to the mbed

- A simple mechanical switch to set a digital input pin either high (‘1’) or low (‘0’) by connecting it to switch between the 3.3V and GND rails.

- To the setup included in the previous example, add a mechanical switch output to pin 7.

- Connect the switch inputs to 0V (GND, pin1) and 3.3V (Vout, Pin 40).



Implementing a digital switch input on the mbed

- Exercise 2: Create a new program for the LED switch project. Modify the default main.cpp code as shown.

- When two forward slash symbols (//) are used, the compiler ignores any proceeding text, so we can use this to write useful comments.

```
#include "mbed.h"
DigitalOut redled(p5);
DigitalOut greenled(p6);
DigitalIn  switchinput(p7);
int main() {
    while(1) {

      if (switchinput==1) {
        greenled = 0;    //green led is off
        redled = 1;      // flash red led
        wait(0.2);
        redled = 0;
        wait(0.2);
      }
      else if (switchinput==0) {
        redled = 0;      //red led is off
        greenled = 1;    // flash green led
        wait(0.2);
        greenled = 0;
        wait(0.2);
      }
    }
}
```

Implementing a digital switch input on the mbed

- Compile, download and run the code on the mbed.
- Look at the code, the if (switchinput==1) statement allows the code to operate in two different ways, dependent on the value of the digital input (i.e. the mechanical switch position).
- If the switch gives a value of 1, the green LED is set to zero (off) and the red LED is programmed to flash. If the digital input is low (0), we see the roles of the LEDs reversed.

Digital switch exercises

- Exercise 3: Create a 1000 Hz digital square wave output which can be analysed on an oscilloscope.

- Exercise 4: Using a digital input switch, create a square wave output that doubles in frequency when a digital input switch is switched on.

- Exercise 5: Create a system which counts the number of times a digital switch is pressed or changed, and lights an LED when 10 instances have been counted.

Summary

- Introduction to digital terminology
- Digital outputs on the mbed
- Using LEDs on the mbed pins
- Connecting switches to the mbed
- Implementing a digital switch input on the mbed