

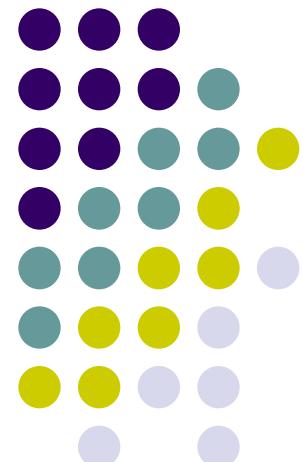


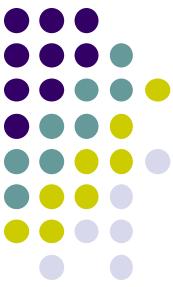
Chapter 3: Digital Input and Output

EE2405

嵌入式系統與實驗

Embedded System Lab





DigitalOut Example

```
#include "mbed.h"

DigitalOut myled(LED1);

int main()
{
    // check that myled object is initialized and connected to a pin
    if (myled.is_connected()) {
        printf("myled is initialized and connected!\n\r");
    }

    // Blink LED
    while (1) {
        myled = 1;          // set LED1 pin to high
        printf("myled = %d \n\r", (uint8_t)myled);
        ThisThread::sleep_for(500ms);

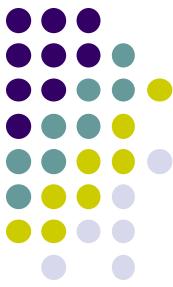
        myled.write(0);      // set LED1 pin to low
        printf("myled = %d \n\r", myled.read());
        ThisThread::sleep_for(500ms);
    }
}
```



DigitalOut

- DigitalOut is a C++ class

Function	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 (overload operator)
operator int()	A shorthand for read (overload operator)
is_connected()	Return non-zero if connected to a GPIO pin



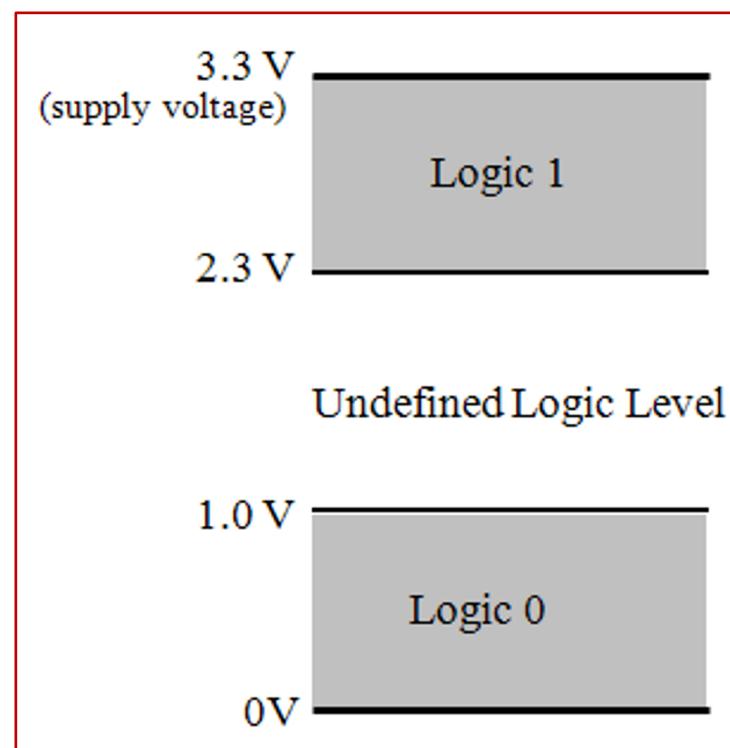
sleep_for() functions

- Sleep for a specified time period in millisec
 - Mbed OS will suspend the current thread.
- Defined under ThisThread:: namespace
 - Control the current running thread



Logic values and voltages

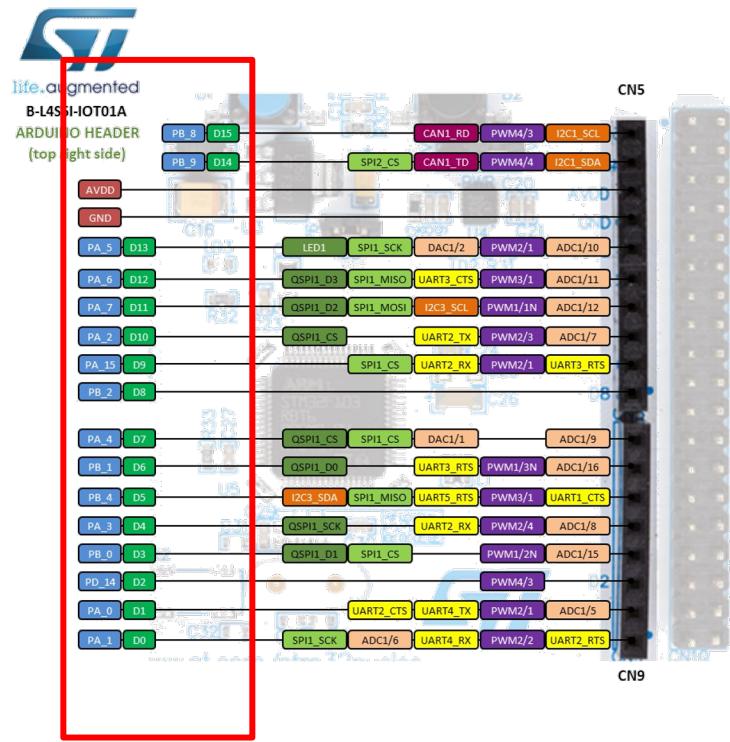
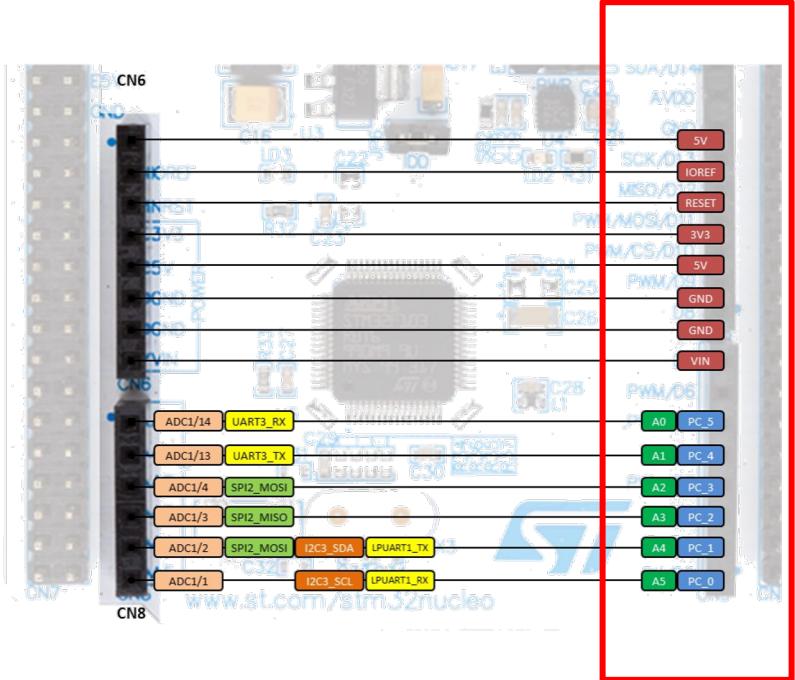
- Logic values are represented as electrical voltages:
 - Logic 0 == 0 - 1.0V
 - Logic 1 == 2.3-3.3V
- In this example, both logic 0 and 0 has a margin of 1V.
 - To provide noise margin to tolerate variations and environment changes.





Pin Names

- Rules defined in
https://os.mbed.com/teams/ST/wiki/pinout_labels
- For discovery iot node, check PinNames.h in
https://github.com/ARMmbed/mbed-os/tree/master/targets/TARGET_STM/TARGET_STM32L4/TARGET_STM32L4S5xI/TARGET_B_L4S5I_IOT01A



Use only the labels written in **blue/white** or **green/white** in your codes.

Or their aliases, for example:

`LED1 = PA_5,`

`LED2 = PB_14,`

`LED3 = PC_9,`

`BUTTON1 = PC_13,`

Or use Arduino pin names:

`A0 = PC_5,`

`A1 = PC_4,`

`A2 = PC_3,`

`A3 = PC_2,`

`A4 = PC_1,`

`A5 = PC_0,`

`D0 = PA_1,`

`D1 = PA_0,`

`D2 = PD_14,`

`D3 = PB_0,`

`D4 = PA_3,`

`D5 = PB_4,`

`D6 = PB_1,`

`D7 = PA_4,`

`D8 = PB_2,`

`D9 = PA_15,`

`D10 = PA_2,`

`D11 = PA_7,`

`D12 = PA_6,`

`D13 = PA_5,`

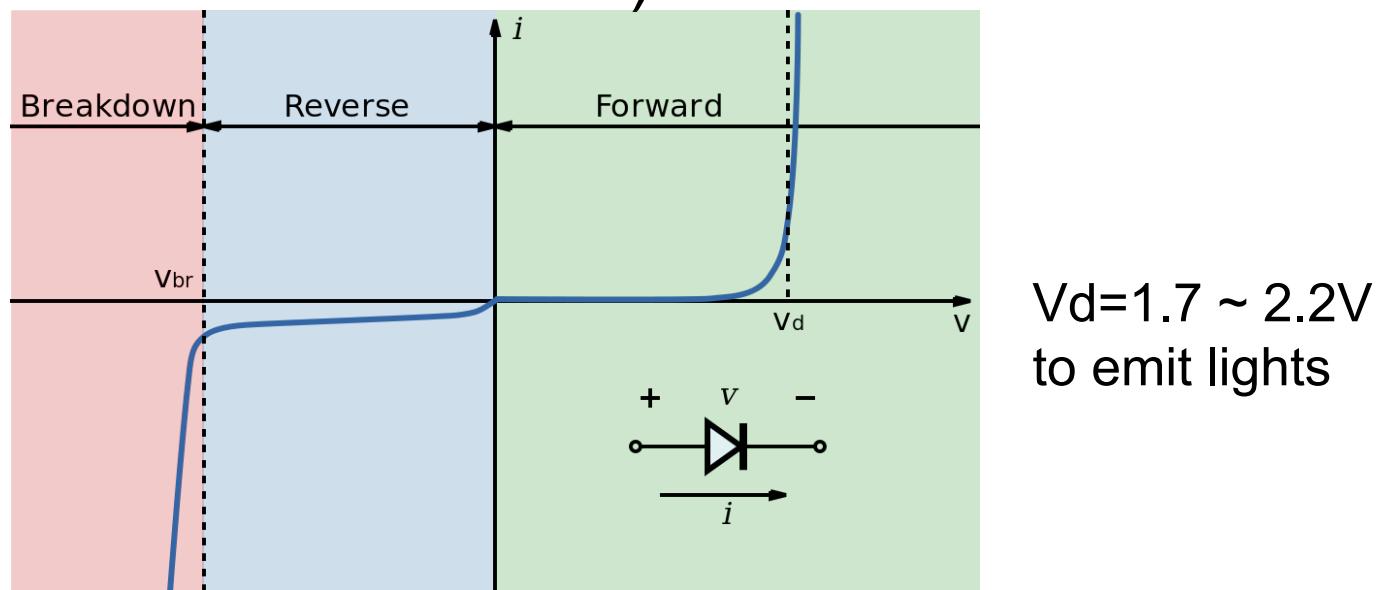
`D14 = PB_9,`

`D15 = PB_8,`



Light Emitting Diode (LED)

- A LED is a semiconductor light source that emits light when current flows through it.
 - Electrons in the diode recombine with electron holes, releasing energy in the form of photons (electroluminescence)

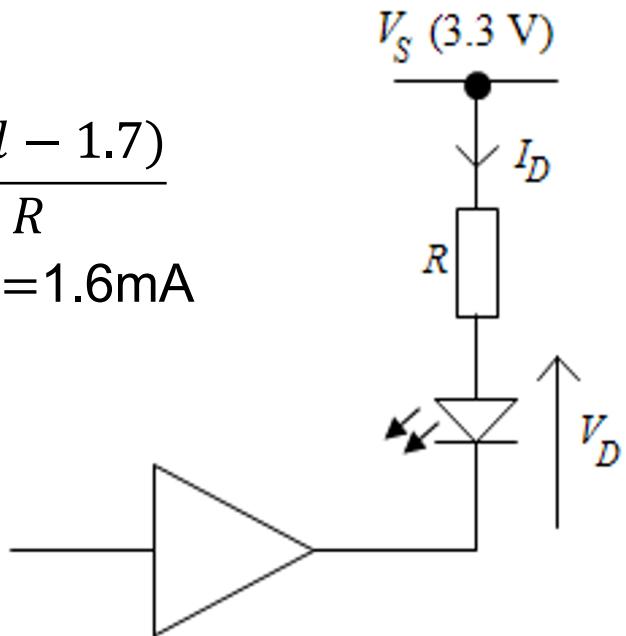




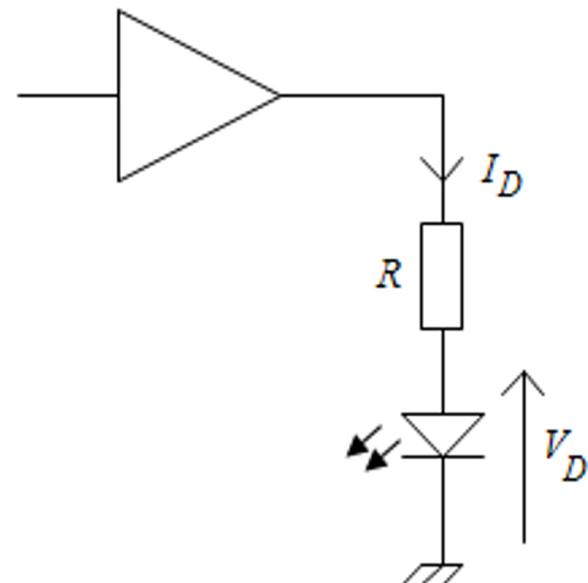
Drive a LED with a logic pin

- Pin includes a buffer and a series resistor.

$$I_D \sim \frac{(V_{dd} - 1.7)}{R}$$
$$= \frac{(3.3 - 1.7)}{100} = 1.6\text{mA}$$



Gate output sinking
current from LED load

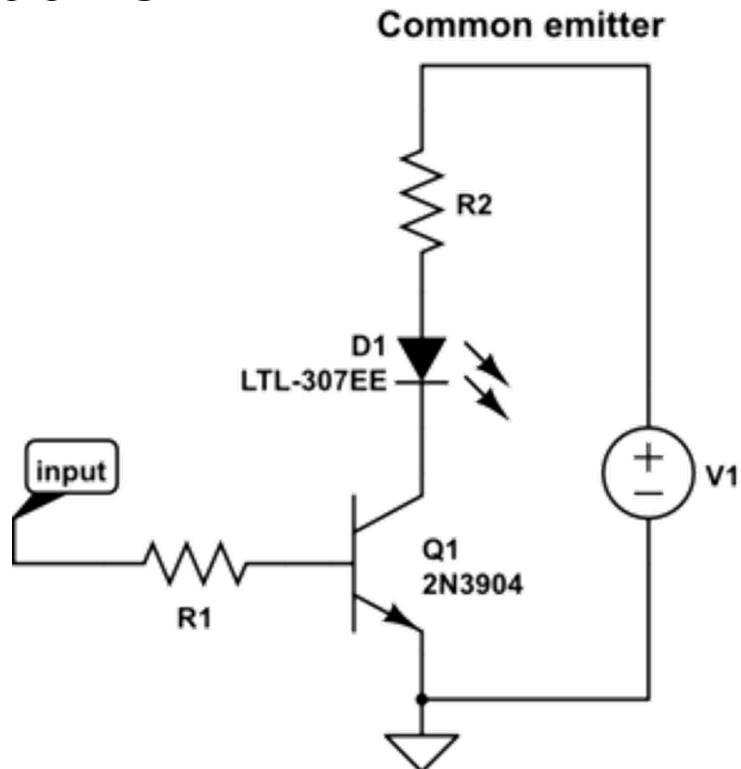


Gate output sourcing
current to LED load

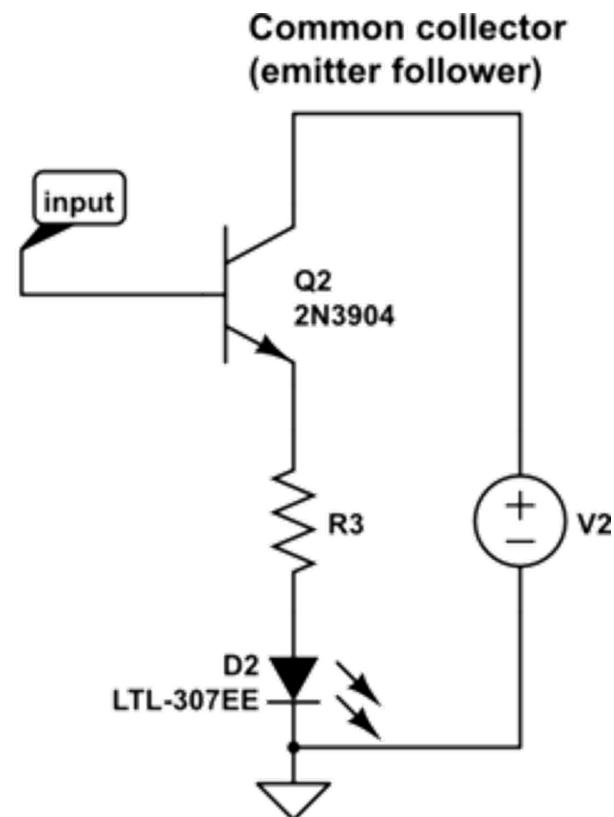


LED driver circuit

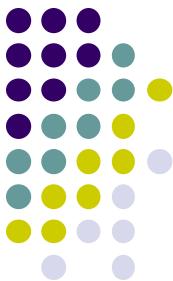
- The BJT transistor represents a part of the buffer.



BJT sinking current from LED load

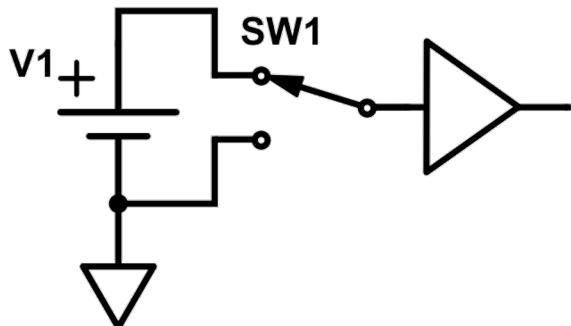


BJT sourcing current to LED load¹⁰

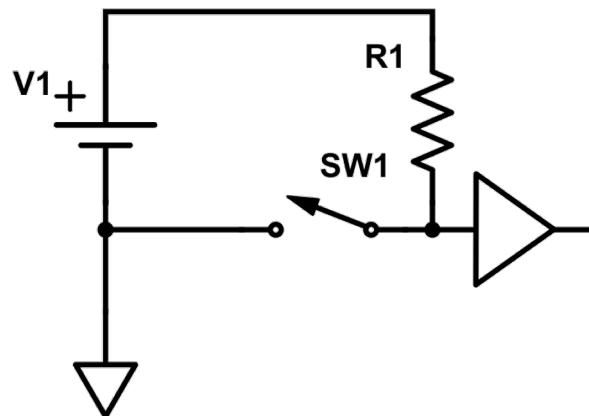


Switches as inputs

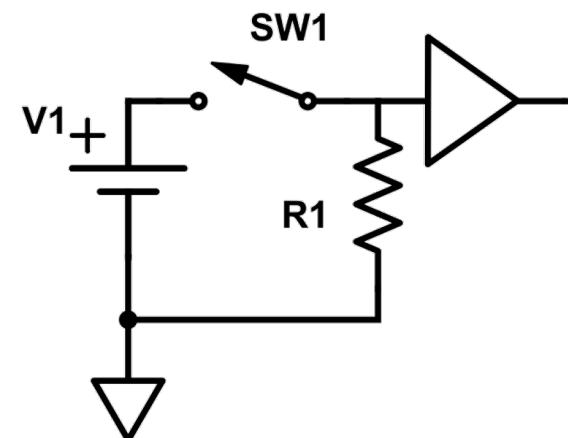
- We can use electromechanical switches to input logic levels:



(a) Single-pole double-throw (SPDT) connection.



(b) Single-pole single-throw (SPST) with pull-up resistor.



(c) SPST with pull-down resistor

Note: the diagrams show a logic buffer, which can in each case be a microcontroller/mbed input.

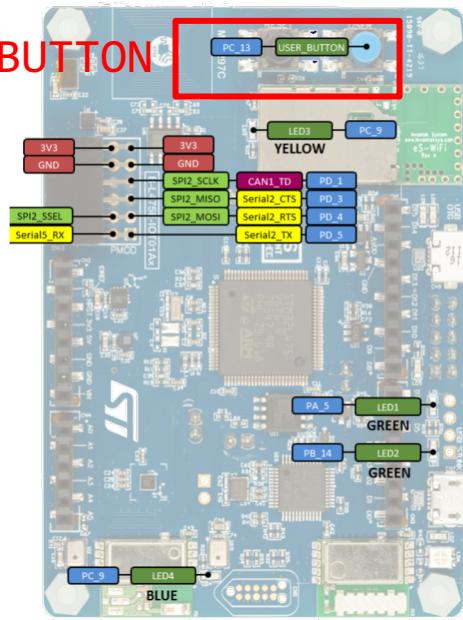


DigitalIn

- Note that use of DigitalIn enables by default the internal pull-down resistor, i.e. the input circuit is configured as Figure (c) in previous slide. This can be disabled, or an internal pull-up enabled, using the mode() function.

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, with parameter chosen from: PullUp, PullDown, PullNone, OpenDrain
operator int()	A shorthand for read()

USER_BUTTON



Example of DigitalIn

```
#include "mbed.h"

DigitalIn mypin(USER_BUTTON);
DigitalOut myled(LED1);

int main()
{
    // check mypin object is initialized and connected to a pin
    if (mypin.is_connected()) {
        printf("mypin is connected and initialized! \n\r");
    }

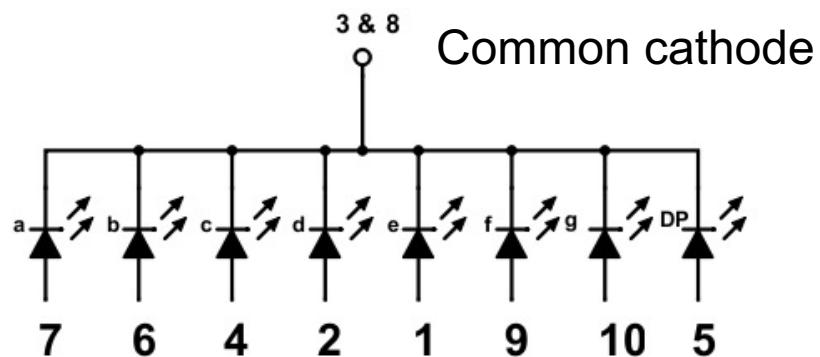
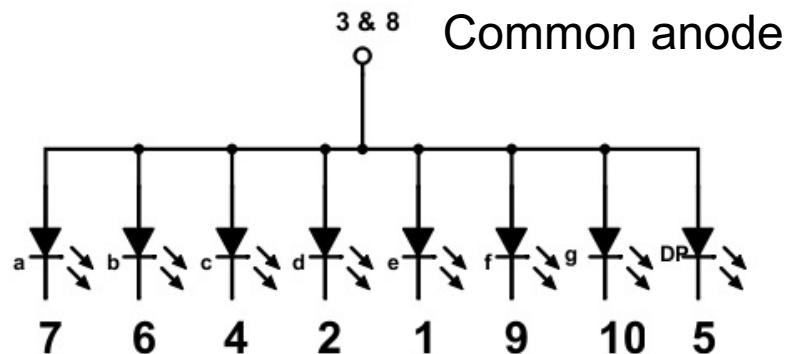
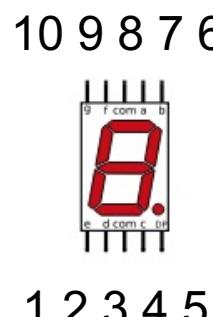
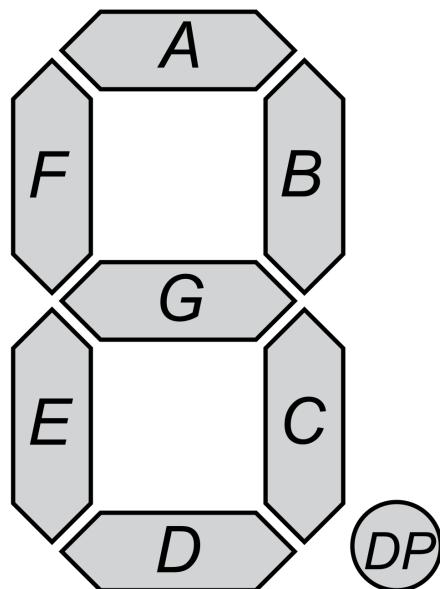
    // Optional: set mode as PullUp/PullDown/PullNone/OpenDrain
    mypin.mode(PullNone);

    // press the button and see the console / led change
    while (1) {
        printf("mypin has value : %d \n\r", mypin.read());
        myled = mypin; // toggle led based on value of button
        ThisThread::sleep_for(250ms);
    }
}
```



Seven Segment Display

- LEDs arranged in the following to display alphanumeric values (plus dot, DP).
 - Common anode
 - Common cathode



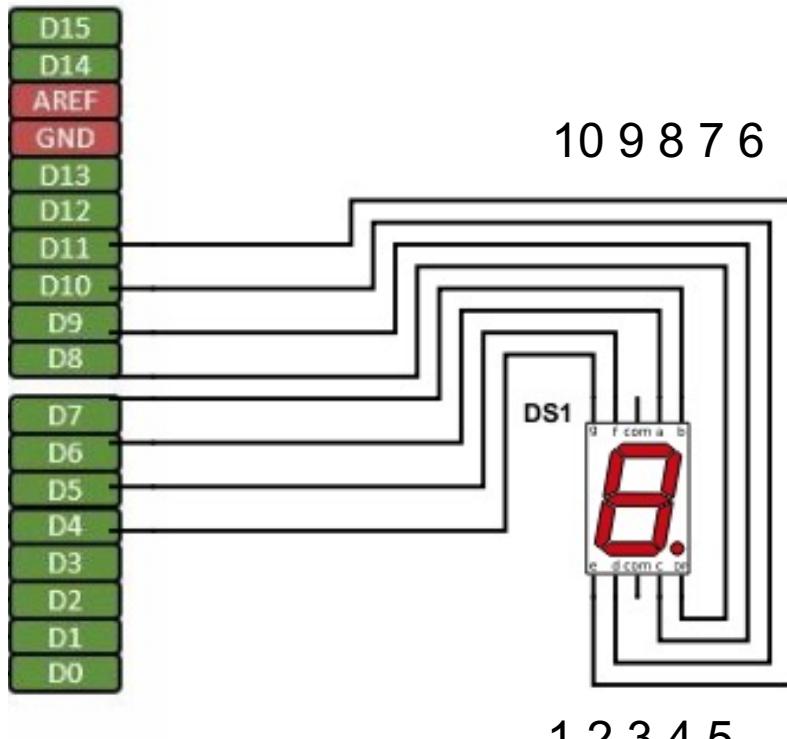


Example codes of display

Digit	Display	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0	0x3F	0x7E	on	on	on	on	on	on	off
1	1	0x06	0x30	off	on	on	off	off	off	off
2	2	0x5B	0x6D	on	on	off	on	on	off	on
3	3	0x4F	0x79	on	on	on	on	off	off	on
4	4	0x66	0x33	off	on	on	off	off	on	on
5	5	0x6D	0x5B	on	off	on	on	off	on	on
6	6	0x7D	0x5F	on	off	on	on	on	on	on
7	7	0x07	0x70	on	on	on	off	off	off	off
8	8	0x7F	0x7F	on						
9	9	0x6F	0x7B	on	on	on	on	off	on	on
A	A	0x77	0x77	on	on	on	off	on	on	on
b	B	0x7C	0x1F	off	off	on	on	on	on	on
C	C	0x39	0x4E	on	off	off	on	on	on	off
d	D	0x5E	0x3D	off	on	on	on	on	off	on
E	E	0x79	0x4F	on	off	off	on	on	on	on
F	F	0x71	0x47	on	off	off	off	on	on	on

Common cathode connection as shown in the next slides
https://en.wikipedia.org/wiki/Seven-segment_display

Connect to 7-segment display



Connection	pin
D4 → 10	g
D5 → 9	f
D6 → 7	a
D7 → 6	b
D8 → 5	DP
D9 → 4	c
D10 → 2	d
D11 → 1	e
GND → 3, 8	

Common cathode connection

mbed code example for 7-segment display



```
/* Display demonstration digits 0 to 9 in turn.  
*/  
#include "mbed.h"  
  
BusOut display(D6, D7, D9, D10, D11, D5, D4, D8);  
char table[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D,  
0x7D, 0x07, 0x7F, 0x6F}; //The table contains 0-9  
  
int main(){  
    while(1){  
        for (int i = 0; i<10; i = i+1){  
            display = table[i];  
            ThisThread::sleep_for(1s);  
        }  
    }  
}
```

Note: Busout is a collection of DigitalOut



Chapter Review

- Logic signals, expressed mathematically as 0 or 1, are represented in digital electronic circuits as voltages. One range of voltages represents 0, another represents 1.
- LEDs can be driven directly from the mbed digital outputs. They are a useful means of displaying a logic value, and of contributing to a simple human interface.
- Electromechanical switches can be connected to provide logic values to digital inputs.
- Multi-colored LEDs can be made by enclosing several individual LEDs of different colors in the same housing. The mbed application board contains one of these.
- A range of simple opto-sensors have almost digital outputs, and can with care be connected directly to mbed pins.
- Where the mbed pin cannot provide enough power to drive an electrical load directly, interface circuits must be used.