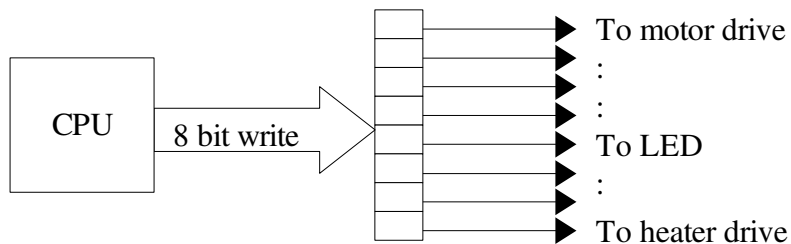


DT228-2 Microprocessor Systems 1

Lab 2. Input/Output masks.

An output port may be used to control several devices as shown below:



Each port bit is connected to a different device that may be turned on or off. The MSP430's CPU, in common with many others is unable to write single bits to a port to selectively turn on or off a device. It must instead write a set of bits – eight being the smallest number (a byte). How then can the CPU just turn on or off an individual device without affecting the others? By using **masks**.

Example 1

An 8 bit output port contains the following pattern of 1's and zero's:

B7	B6	B5	B4	B3	B2	B1	B1
1	0	1	1	0	0	1	1

We could set Bit 2 alone by the following sequence of operations:

Step 1: Read current contents of port output register: 1011 0011
 Create a mask with bit 2 only set: 0000 0100
 OR both numbers 1011 0111
 Write the result back to the port.

The port contents will now be:

B7	B6	B5	B4	B3	B2	B1	B1
1	0	1	1	0	1	1	1

As you can see only one of the bits has changed state.

Example 2.

An 8 bit output port contains the following pattern of 1's and zero's:

B7	B6	B5	B4	B3	B2	B1	B1
1	0	1	1	0	0	1	1

We could clear bit 5 alone by the following sequence of operations.

Step 1: Read current contents of port output register: 1011 0011
 Create a mask with bit 5 only clear: 1101 1111
 AND both numbers 1001 0011
 Write the result back to the port.

The port contents will now be:

B7	B6	B5	B4	B3	B2	B1	B1
1	0	0	1	0	0	1	1

Again only one bit is changed.

To summarize:

If you want to set a single bit in a number OR it with a number which has a 1 (only) in the relevant bit position – all other bits are 0.

If you want to clear a single bit in a number AND it with a number which has a 0 (only) in the relevant bit position – all other bits are 1.

Exercise

Enter the program shown in Listing 1 below and show that it can be used to buzz the buzzer and control one of the relays.

Questions/tasks:

How does the program work (in detail) (see Figure 1 below)

The tilde character is used in the program – what for?

How would you extend the program to allow control by another button of the second relay?

How do #define statements work and why is it good to use them for constants like mask values.

What is a relay and how can it be used to control electrical devices?

```
#include <msp430x14x.h>
```

```
#define BTN1MASK 0x10  
#define BTN2MASK 0x20  
#define RLY1MASK 0x20  
#define RLY2MASK 0x40  
#define BUZZ1MASK 0x04  
#define BUZZ2MASK 0x08
```

```
int Btn1Pressed()  
{  
    int Result;  
    Result = P4IN & BTN1MASK;  
    if (Result == 0)  
        return 1;
```

```

    else
        return 0;
}
int Btn2Pressed()
{
    int Result;
    Result = P4IN & BTN2MASK;
    if (Result == 0)
        return 1;
    else
        return 0;
}
void Relay1On() {
    P1OUT = P1OUT | RLY1MASK;
}
void Relay1Off() {
    P1OUT = P1OUT & ~RLY1MASK;
}
void InitializeBoard() {
    P4DIR = BUZZ1MASK+BUZZ2MASK; // setup bits 2 and 3 on Port 4 as
                                // outputs to drive the buzzer
    P1DIR = RLY1MASK+RLY2MASK;   // bits 5 and 6 on port 1 drive the
                                //relays
}
void delay() {
    int dly;
    for (dly=0;dly<50;dly++);
}
void Beep() {
    int count;
    for (count = 0; count < 50; count ++ ) {
        P4OUT = BUZZ1MASK;
        delay();
        P4OUT = BUZZ2MASK;
        delay();
    }
    P4OUT = 0;
}
void main( void )
{
    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;
    InitializeBoard();
    while (1) {
        if (Btn1Pressed())
            Beep();

        if (Btn2Pressed())
            Relay1On();
        else
            Relay1Off();
    }
}

```

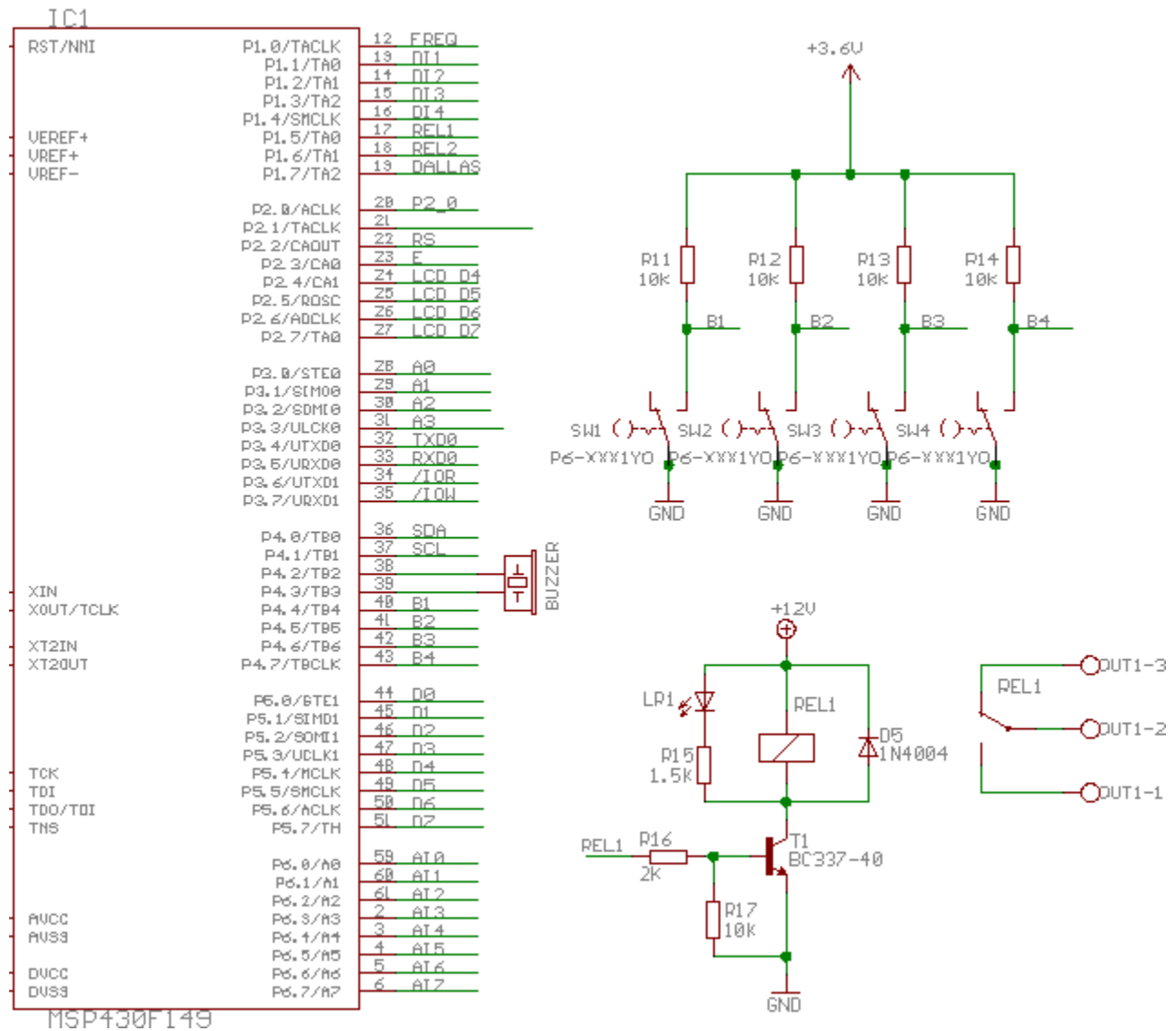


Figure 1: Extract from the EasyWeb2 circuit diagram