



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Universidad de Santiago de Chile
Facultad de Ingeniería
Departamento de Ingeniería Mecánica



DEPARTAMENTO
DE INGENIERÍA
MECÁNICA

Ayudantía N°8

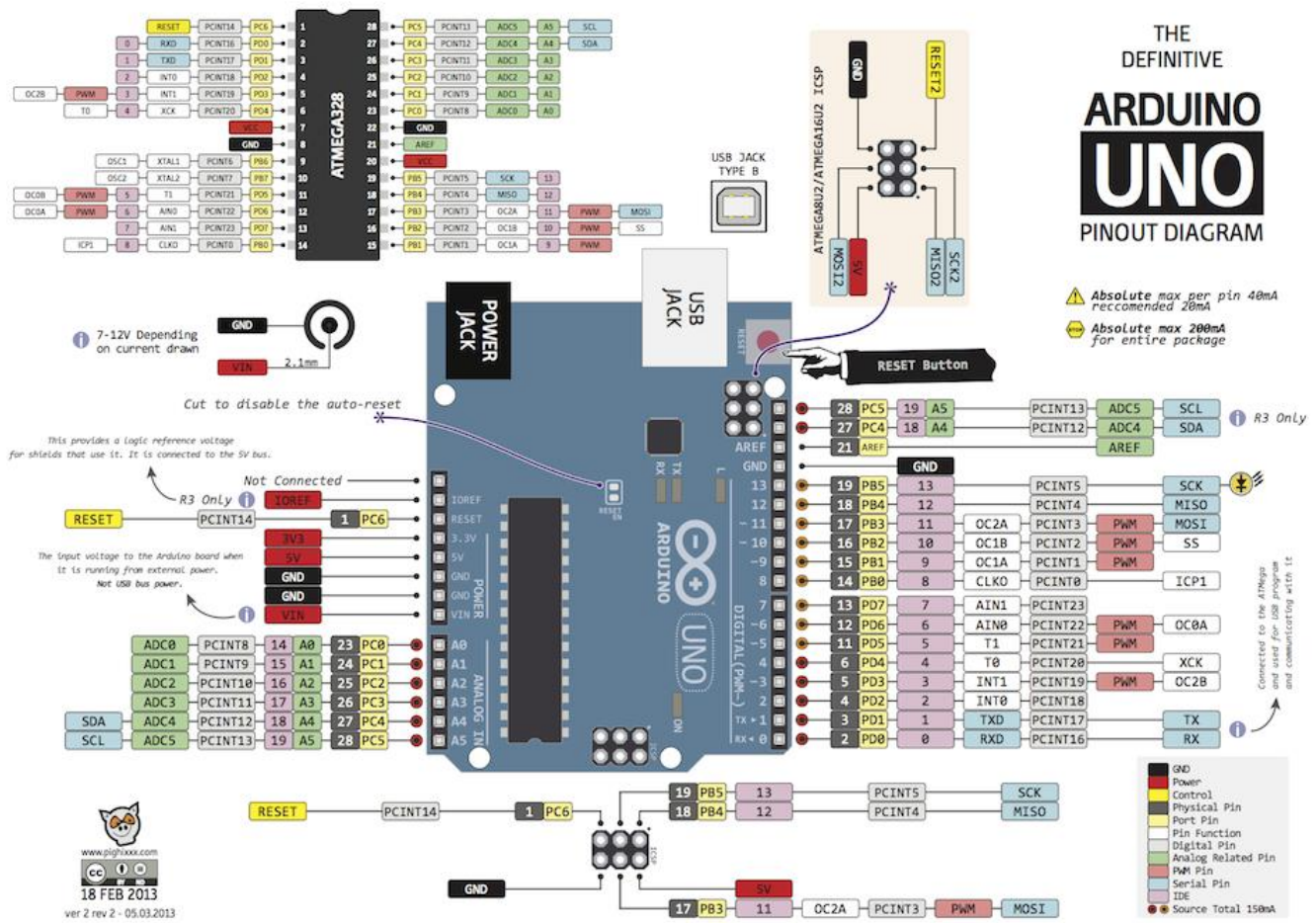


Código C introducción:

Resumen:

- i) `_BV()`: La función Big value sirve para definir el Pin de trabajo.
- ii) `DDRX`: Sirve para definir los pines de salida, por ejemplo un led, donde en la parte que sale "X" se escribe la inicial de los puertos donde se trabaja. Es necesario para definirlos como "salidas", el resto de los pines serán considerados entradas
- iii) `PORTX`: Sirve para definir los puertos de entrada, por ejemplo un botón, es decir los pines que le entregan información al código. Además sirve para activar los pull ups. Lo que quiere decir las resistencias internas del procesador.
- iv) `while(true)`: se utiliza como ciclo while, donde la diferencia con los utilizados anteriormente es que se establece un valor "verdadero" cosa que haga el ciclo por siempre y así el microprocesador o el hardware con el que se esté comunicando no deje de realizar la tarea que se le encargo. Por lo general se rellena con un 1, ya que es un valor siempre verdadero.
- v) `bit_is_clear(puerto,Bit)`: Si el bit se encuentra "set" entregara como resultado un valor igual a cero. Sin embargo si el bit se encuentra "clear" entregará un valor "No-cero".
- vi) `bit_is_set(puerto,Bit)`: Si el bit se encuentra "set" entregara como resultado un valor "No-cero". Sin embargo si el bit se encuentra "clear" entregará un valor igual a cero.
- vii) `_delay_ms()`: Permite realizar una pausa al programa en milisegundos. Esto quiere decir que el programa simplemente se detiene el tiempo que se le introduzca. Es una herramienta que permite establecer funciones intermitentes como encender y apagar un led. Sin embargo el problema que trae consigo es que al detener por completo el programa, margina a este de realizar todo tipo de tareas secundarias como cálculos matemáticos o lectura de pines de entrada. Lo cual puede ser poco eficiente y problemático para problemas complejos, por eso es que no se emplea muy seguido.
- viii) `if(prueba)`: El comando 'if' permite establecer condiciones mediante una prueba lógica, si esta se cumple se ejecutará lo que este dentro del 'if', de lo contrario saltara a la siguiente línea después de cerrar el 'if'. Si se quisiera se puede agregar un "else" de ser necesario, que permite realizar otra tarea en caso de no cumplirse la condición inicial.
- ix) `for(partida ; prueba ; incremento de la variable a probar)`: el "for" permite realizar varias veces una tarea, tantas veces se desee, la cantidad de veces dependerá de la condición o prueba a la que se someta el contador. Este se define al principio del "for", donde sale partida. Por ejemplo $i=0$. Luego la prueba puede ser $i \leq 60$. Y el incremento puede ser sumándole un valor más a la condición como $i++$ ($i=i+1$). Una vez se cumpla el ciclo pasará a la siguiente línea después de cerrar el for.
- x) `{}`: Los corchetes sirven siempre para poder abrir y cerrar los ciclos, condiciones, el código etc.

- Ejemplo en arduino:



En la siguiente imagen se expone como se llaman los puertos y los pines en un Arduino uno.

De la imagen lo que más interesa saber es el nombre de los puertos, que en este caso se ve en los cuadros amarillos opacos, donde sale por ejemplo PD3. Donde la D indica que es el puerto D y el número tres indica que es el pin número 3.

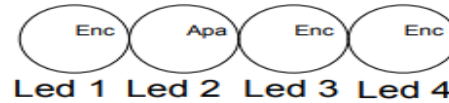
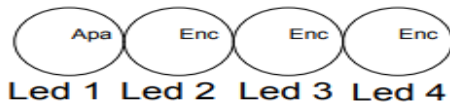
En función de esto es como se sabrá cómo se comunica el código con el Arduino uno.

El primer ejemplo es sobre cómo hacer que 4 leds se vayan apagando y prendiendo de a uno en una dirección y después vuelvan a apagarse de uno en la dirección contraria.

En la siguiente imagen se expresa mejor el problema:

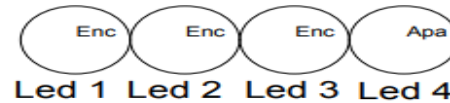
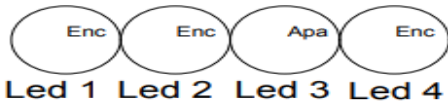


Primer Ciclo



Segundo Ciclo

Sexto Ciclo



Quinto Ciclo

Resultado:

```
#define F_CPU 12000
#include <avr/io.h>
#include <util/delay.h> //Libreria para usar delay

int main(void){

    DDRD |= _BV(PD2) | _BV(PD3) | _BV(PD4) | _BV(PD5); // Salidas leds
    unsigned char i,j; //Contadores de ciclo
    while (1){
        for (i=1 ; i<=4 ;i++){ // Recorre cada led, aplicándole un delay
            if (i==1) {
                PORTD |= _BV(PD3) |_BV(PD4) |_BV(PD5);
                PORTD &= ~_BV(PD2);
                _delay_ms(500);
            } if (i==2) {
                PORTD |= _BV(PD2) |_BV(PD4) |_BV(PD5);
                PORTD &= ~_BV(PD3);
                _delay_ms(500);
            } if (i==3) {
                PORTD |= _BV(PD2) |_BV(PD3) |_BV(PD5);
                PORTD &= ~_BV(PD4);
                _delay_ms(500);
            } else {
                PORTD |= _BV(PD2) |_BV(PD3) |_BV(PD4);
                PORTD &= ~_BV(PD5);
                _delay_ms(500);
            }
        }
        for (j=1 ; j<=4 ;j++){
            if (j==1) {
                PORTD |= _BV(PD2) |_BV(PD3) |_BV(PD4);
                PORTD &= ~_BV(PD5);
                _delay_ms(500);
            } if (j==2) {
                PORTD |= _BV(PD2) |_BV(PD3) |_BV(PD5);
                PORTD &= ~_BV(PD4);
                _delay_ms(500);
            } if (j==3) {
                PORTD |= _BV(PD2) |_BV(PD4) |_BV(PD5);
                PORTD &= ~_BV(PD3);
                _delay_ms(500);
            } else {
                PORTD |= _BV(PD3) |_BV(PD4) |_BV(PD5);
```

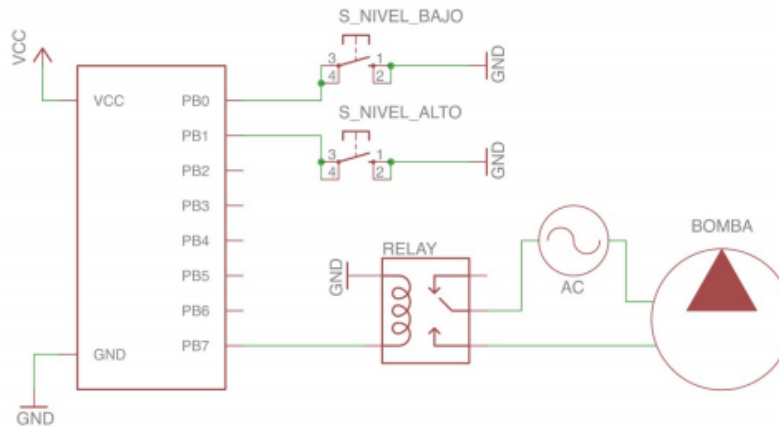
```

PORTD &= ~_BV(PD2);
_delay_ms(500);
}
}
}

```

Problema 2. (2.0 Pts.) Se desea programar un microcontrolador para controlar el llenado de un estanque, para esto se dispone de dos sensores de nivel llamados S_NIVEL_BAJO, que se activa cuando el nivel del contenedor está bajo el mínimo y S_NIVEL_ALTO que se activa cuando el nivel del estanque está por sobre el nivel máximo. Ambos sensores cuentan con un circuito interno que elimina la transición del interruptor y entregan una señal limpia. Adicionalmente se cuenta con un relé que está conectado a una bomba, esta bomba está conectada a una fuente de agua inagotable. Como se puede observar en la figura todos los dispositivos mencionados están conectados al puerto B, la documentación de dicho puerto se encuentra a continuación

Desarrolle un código en lenguaje C para mantener siempre el estanque con una cota de agua aceptable que se encuentre entre los sensores de nivel.



Cuando un pin del puerto B está configurado como salida el valor lógico corresponde al valor del pin en el registro PORTB, en caso de estar configurado como entrada un bit activo implica activar el pull-up en el pin físico.

Problema de PEP:

Bit	7	6	5	4	3	2	1	0
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Val Inicial	0	0	0	0	0	0	0	0

DDRB – Registro de dirección del puerto B

Un valor lógico de 1 implica que el pin está configurado como salida, un valor 0 significa que el pin está configurado como entrada.

Bit	7	6	5	4	3	2	1	0
DDRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Val Inicial	0	0	0	0	0	0	0	0

PINB – Registro de estado del puerto B

Los valores del registro PINB dependen de los valores lógicos de cada uno de los pines físicos.

Bit	7	6	5	4	3	2	1	0
PINB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Val Inicial	0	0	0	0	0	0	0	0



Solución:

```
#include<avr/io.h>

int main(void){
    PORTB |= _BV(PB0) | _BV(PB1); //Sensores
    DDRB |= _BV(PB7); //Bomba
    unsigned char contador_logico=0; //Para determinar el estado lógico del puerto PB7
    while(1){
        if (bit_is_clear(PINB,PB0)){ //Si se activa el sensor de que el nivel está bajo
            contador_logico=1; //El contador logico tomar el valor de verdadero
        } else{
            while (contador_logico){ //Un while que va verificando el llenado del estanque
                if(bit_is_clear(PINB,PB1)){ //Si se prende el sensor de que esta alto el nivel
                    PORTB &= ~_BV(PB7); //Apaga el puerto PB7
                    contador_logico=0; //Cambia el estado logico para que se acabe el while
                }else{
                    PORTB |= _BV(PB7); //Mantiene prendido el puerto PB7 hasta que se llene
                }
            }
        }
    }
}
```

de agua