

UNIVERSIDAD DE SANTIAGO DE CHILE

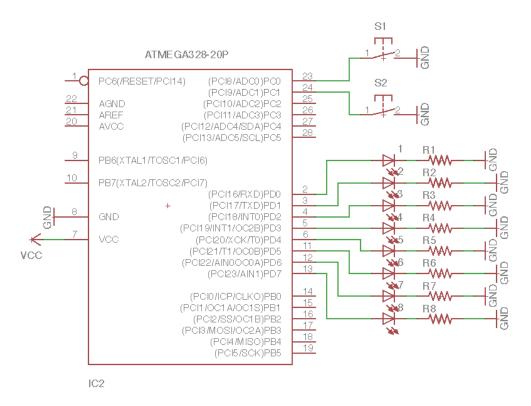
FACULTAD DE INGENIERÍA





Secuencias LED y Pulsadores:

En el siguiente esquemático se muestra el circuito utilizado para realizar secuencias con luces LED y realizar cambios a través de switch momentáneos.



Se pide lo siguiente:

- Configurar las entradas y salidas de los puertos correspondientes al esquemático, activando las resistencias Pull-Up de los switch *S1* y *S2* ubicados en el puerto C.
- Realizar las siguientes secuencias de leds en el puerto PORTD:
 - Representar de forma binaria en el puerto, un contador numérico que varíe entre 0 y 255, con un retraso de 10ms entre cada aumento del contador
 - Realizar secuencia bit a bit de forma ascendente y luego descendente en PORTD, con un retraso de 100ms entre cada cambio de bit.
 - Realizar parpadeos en los led ubicados en PDO y PD1, con 120ms entre parpadeo, y 500ms cada 3 parpadeos, alternando el estado de ambos led entre sí (PDO:0,PD1:1 - PDO:1,PD1:0)
- Utilizar S1 para alternar entre las 3 secuencias anteriores, utilizando delay de 1seg para realizar "debounce".
- Utilizar S2 para apagar PORTD o volver a las secuencias, utilizando delay de 1seg para realizar "debounce".

Solución

```
#define F CPU 16000000//Frecuencia oscilador arduino mega
#include <avr/io.h>//Librería I/0
#include <util/delay.h>//Librería delay
#define Puerto_botones PINC
#define S1 PC0
#define S2 PC1
void setup_hardware()//Configuración puertos
{
       DDRD = BV(PD0) \mid BV(PD1) \mid BV(PD2) \mid BV(PD3) \mid BV(PD4) \mid BV(PD5) \mid BV(PD6)
       LBV(PD7);//Configuración salidas puerto B
       //DDRD = 255;
       //DDRD = 0b111111111;
       //DDRD = (1<<0) | (1<<1) | (1<<2) | (1<<3) | (1<<4) | (1<<5) | (1<<6) | (1<<7);
       PORTC |= BV(S1) | BV(S2); //Resistencias Pull-Up para S1 y S2
}
int main(void)
       setup hardware();
       unsigned char secuencia = 0; //Crea la variable que permite seleccionar la secuencia
       unsigned char permiso = 1;
      while(1)
       {
              if(bit is clear(Puerto botones,S1))//Cambio de secuencia
              {
                     _delay_ms(1000);//Debounce
                     secuencia++;
                     if(secuencia > 2)
                     {
                            secuencia = 0;
                     }
              }
              if(bit_is_clear(Puerto_botones,S2))//Apagar puerto o seguir con secuencias
                     delay ms(1000);//Debounce
                     if(permiso == 1)
                     {
                            permiso = 0;//Desactiva secuencias
                            PORTD = 0;//Apaga puerto
                            } else {
                            permiso = 1;//Activa secuencias
                     }
              }
```

```
if(permiso == 1) {//Si esta activado (SW2 cambia esta variable)
       if(secuencia == 0)//Primera secuencia, contador a binario
       {
              PORTD = 0; //Apaga cualquier estado anterior.
              for(unsigned int i = 1; i <= 254; i++)//contador i recorre numeros de 1 en 1</pre>
              {
                     PORTD = i; //Escribe el contador en PORTD, de forma binaria
                     _delay_ms(10);
              }
       } else if(secuencia == 1)//Secuencia 2, bit a bit
              PORTD = 0;
              for(unsigned char i = 0; i <= 6; i++)//Recorre bits de forma ascendente</pre>
              {
                     PORTD = _BV(i);//Enciende el pin correspondiente al contador
                     _delay_ms(100);
              for(unsigned char i = 7; i >= 1; i--)//Recorre bits de forma descendente
                     PORTD = BV(i);
                     _delay_ms(100);//Enciende el pin correspondiente al contador
       } else if(secuencia == 2)//Parpadeos
              PORTD |= _BV(PD0);
              PORTD &= ~_BV(PD1);
              _delay_ms(120);
              PORTD = BV(PD1);
              PORTD &= ~ BV(PD0);
              _delay_ms(120);
              PORTD |= _BV(PD0);
              PORTD &= ~ BV(PD1);
              delay ms(120);
              PORTD |= _BV(PD1);
              PORTD &= ~ BV(PD0);
              delay ms(120);
              PORTD |= _BV(PD0);
              PORTD &= ~_BV(PD1);
              _delay_ms(120);
              PORTD |= _BV(PD1);
              PORTD &= ~ BV(PD0);
              delay ms(120);
              PORTD |= BV(PD0);
              PORTD &= ~_BV(PD1);
             _delay_ms(500);
              //
                                   PORTD = 0;
                                   PORTD |= BV(PD1);
              //
              //
                                   for(unsigned char i = 0; i <= 6;i++)</pre>
              //
                                   {
              //
                                          PORTD ^= _BV(PD0);
              //
                                          PORTD ^= BV(PD1);
              //
                                          _delay_ms(120);
                                   }
              //
              //
                                   _delay_ms(500);
      }
}
```

}

}