



Ejercicio 03:

Conceptos

Docente: M.I. Jesús Daniel Garza Camarena

Semestre: agosto - diciembre 2021

Datos del alumno:

Nombre	Matrícula	Carrera
Eduardo Vicente Reyna Villela	1868879	ITS

Frecuencia: Jueves

Hora: M4-M6

Grupo: 001

No. De Lista: 38



Controladores y Microcontroladores Lógicos Programables

Código

```
/* **** */
* Nombre: Eduardo Vicente Reyna Villela *
* Hora clase: M4-M6 *
* Día: Jueves *
* N° de lista: *
* N° de Equipo: No aplica *
* Dispositivo: Atmega328P *
* Rev: 1.0 *
* Propósito de la actividad: *
* Conocer y saber utilizar las pull up y pull down *
* Fecha: 26/08/2021 *
**** */

#include <avr/io.h>
#define F_CPU 1000000UL
#include <util/delay.h>

#define SWITCH PIND
#define SW_0 PIND0
#define SW_1 PIND1
#define SW_2 PIND2
#define SW_3 PIND3
#define SW_4 PIND4
#define SW_5 PIND5

#define LEDS PORTB
#define LED0 PORTB0
#define LED1 PORTB1
#define LED2 PORTB2
#define LED3 PORTB3
#define LED4 PORTB4
#define LED5 PORTB5

void init_ports(void);

int main(void)
{
    init_ports();
    while (1)
    {
        // LEDS = SWITCHES

        // PUL UP
        // Se usa una macro para verificar si el puerto está en LOW para
        encender el primer LED
        // De esta forma funciona bien ya que como el switch está en HIGH no se
        enciende el LED hasta que esté en LOW (activando el switch por la pull-up)
        if ( bit_is_clear(SWITCH, SW_0) ) // --> SWITCH & ( _BV(SW_0) ) == 0 --
        > SWITCH & ( 1 << SW_0 ) == 0
        {
            LEDS |= _BV(LED0);
        }
    }
}
```



Controladores y Microcontroladores Lógicos Programables

```
else
{
    LEDS &= ~_BV(LED0);
}

if ( bit_is_clear(SWITCH, SW_1) )
{
    LEDS |= _BV(LED1);
}
else
{
    LEDS &= ~_BV(LED1);
}

if ( bit_is_clear(SWITCH, SW_2) )
{
    LEDS |= _BV(LED2);
}
else
{
    LEDS &= ~_BV(LED2);
}

// PUL DOWN

if ( bit_is_clear(SWITCH, SW_3) )
{
    LEDS |= _BV(LED3);
}
else
{
    LEDS &= ~_BV(LED3);
}

if ( bit_is_clear(SWITCH, SW_4) )
{
    LEDS |= _BV(LED4);
}
else
{
    LEDS &= ~_BV(LED4);
}

if ( bit_is_clear(SWITCH, SW_5) )
{
    LEDS |= _BV(LED5);
}
else
{
    LEDS &= ~_BV(LED5);
}
}

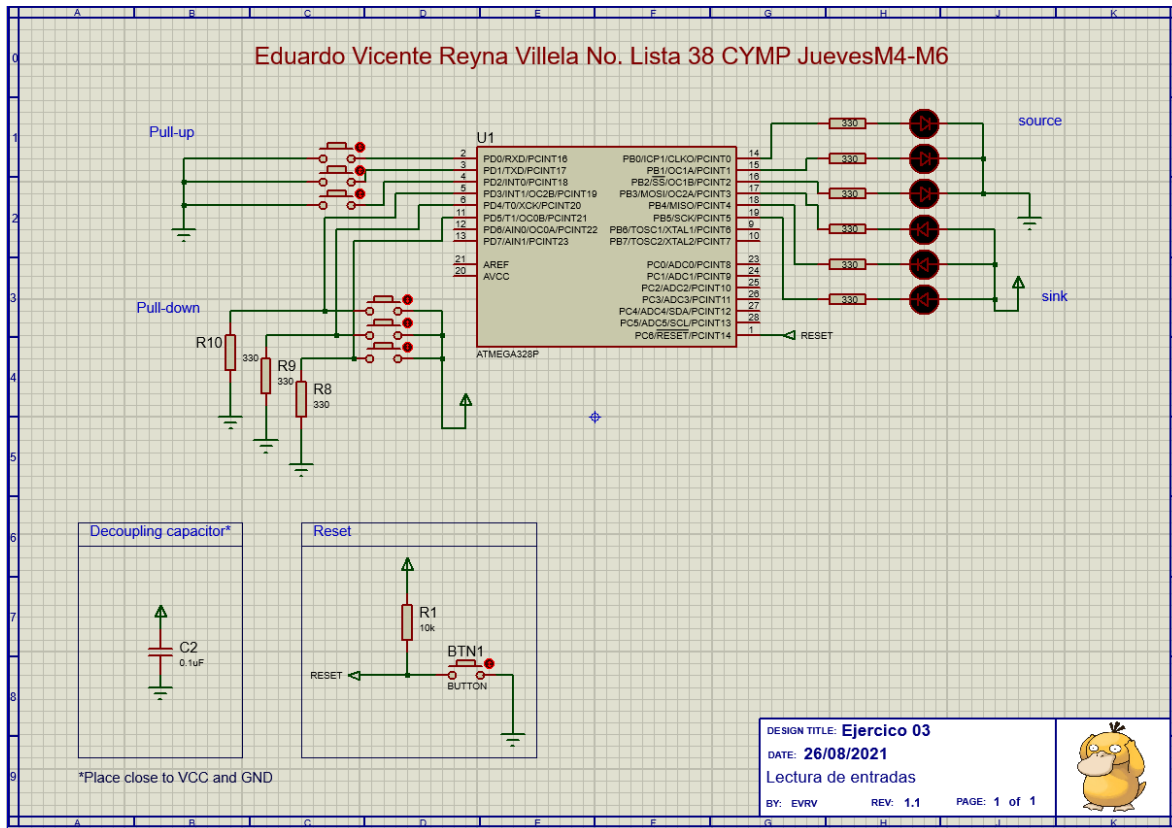
void init_ports(void)
{
    // Entradas
```



Controladores y Microcontroladores Lógicos Programables

```
DDRD  &= ~( _BV(SW_0) | _BV(SW_1) | _BV(SW_2));  
SWITCH |= ( _BV(SW_0) | _BV(SW_1) | _BV(SW_2)); // pull-up  
  
DDRD  &= ~( _BV(SW_3) | _BV(SW_4) | _BV(SW_5));  
SWITCH &= ~( _BV(SW_3) | _BV(SW_4) | _BV(SW_5)); // pull-down  
  
// Salidas  
  
DDRB |= ( _BV(LED0) | _BV(LED1) | _BV(LED2) );  
LEDS &= ~( _BV(LED0) | _BV(LED1) | _BV(LED2) ); // apagadas - source  
  
DDRB |= ( _BV(LED3) | _BV(LED4) | _BV(LED5) );  
LEDS |= ( _BV(LED3) | _BV(LED4) | _BV(LED5) ); // apagadas - sink  
}
```

Imagen del esquemático





Simulación en TINKERCAD o proteus

Se encuentra en archivos