**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

# 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: \*

\* Conoder 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);

}

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

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) );

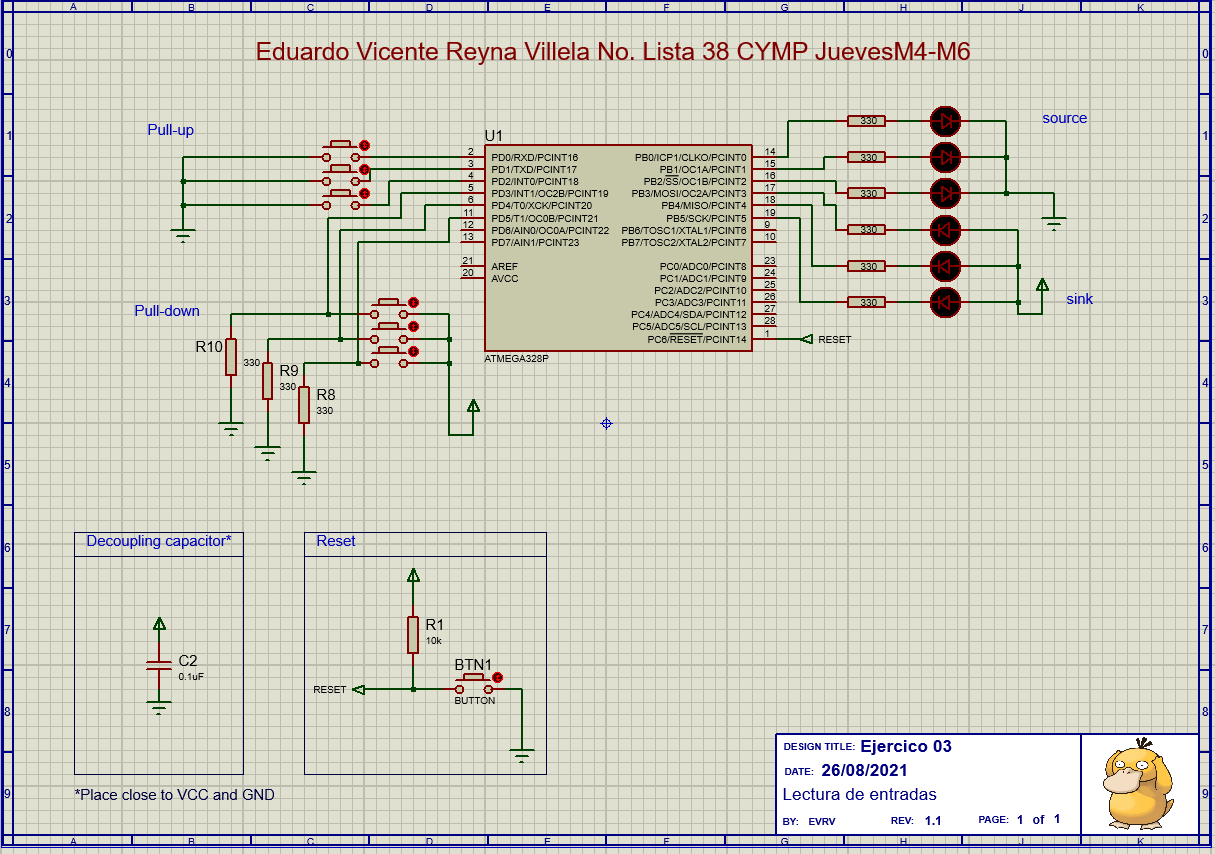
LEDS &= ~( \_BV(LED0) | \_BV(LED1) | \_BV(LED2) ); // apagadas - source

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