24-4-2021

ADRIAN OROZCO ANZURES 18300328  
CARLOS DANIEL LOZANO VAZQUEZ 18300249

SiSTEMAS EMBEBIDOS I

CETI COLOMOS

**Timer**

Actividad 16 – Práctica 9

Logotipo

Descripción generada automáticamente

**Objetivo**

El objetivo de esta práctica es aprender a utilizar los timer del Arduino para realizar una aplicación demostrativa.

**Descripción**

Utilizar Arduino MEGA y conectarle dos LEDs. (LED 1, LED 2).

Programar un timer como contador de pulsos internos para hacer encender “LED1” durante 5 segundos y apagarlo 5 segundos, de forma cíclica infinita.

Programar un timer como contador de pulsos externos para hacer encender “LED 2” cada que lleguen 30 pulsos externos a un pin de entrada del Arduino y apagarlo cada que lleguen otros 30 pulsos externos, de forma cíclica infinita.

Entonces se deben tener al mismo tiempo 2 LEDs parpadeando de acuerdo con lo especificado anteriormente.

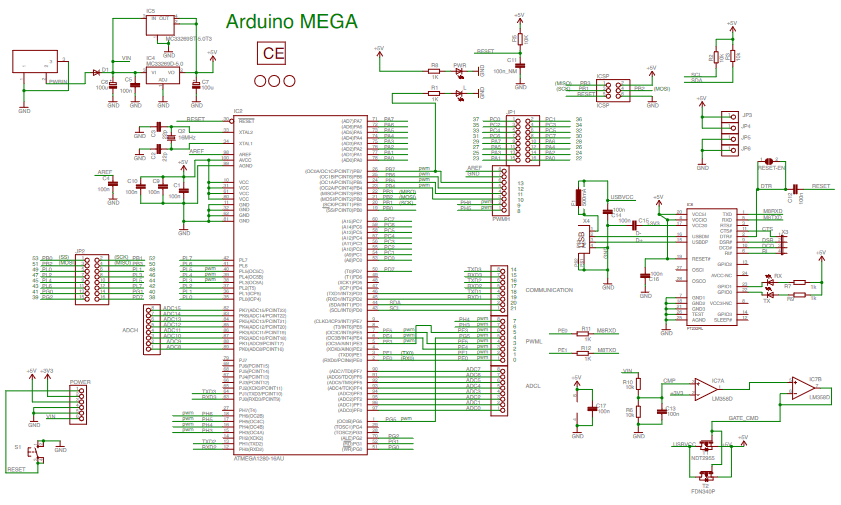
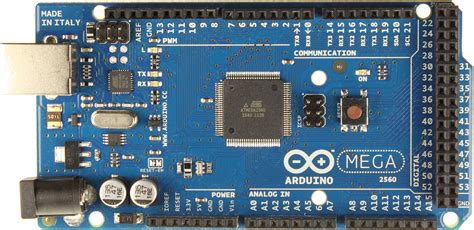
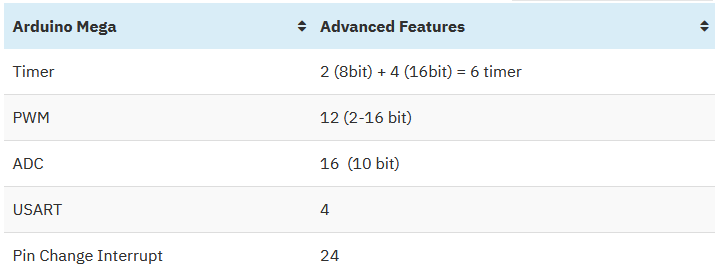
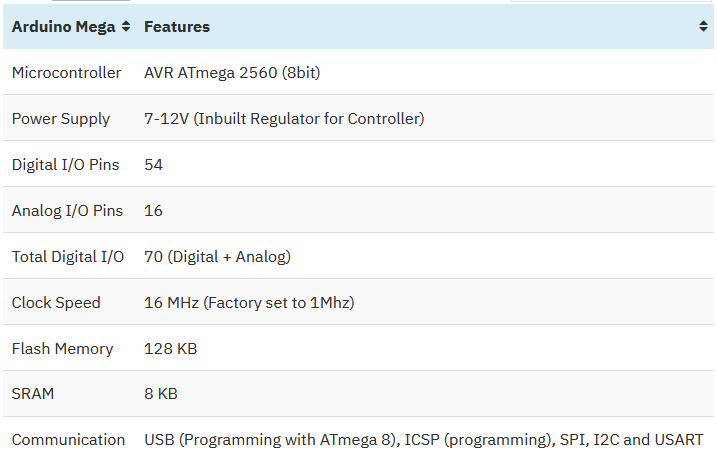
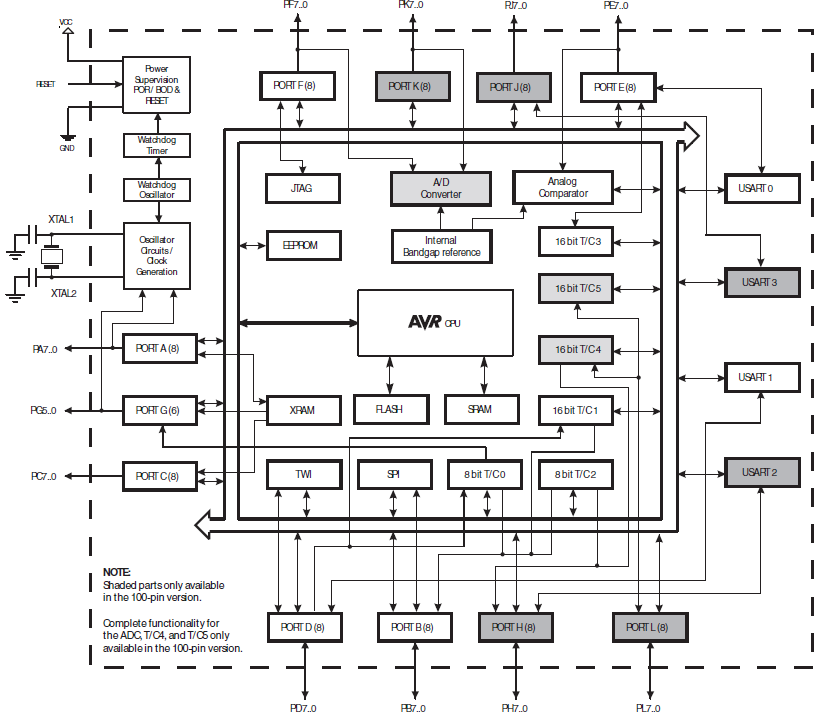
**Resumen**

Usando una tarjeta de Arduino MEGA, un push button y dos LEDs, se debe crear un programa que utilice un timer interno y un timer externo. Uno de estos LEDs debe prenderse y apagarse por 5 segundos, mientras que el otro debe prenderse y apagarse cada 30 repeticiones.

En este caso, el timer externo funcionará como un botón al que se le debe presionar 30 veces para que prenda y apague por igual.

El programa es creado en un editor de texto, ya sea Visual Studio Code, Sublime Text, Atom, o hasta el mismo Arduino IDE; pero para utilizar el código este debe ser verificado en el IDE de Arduino.

**Diagrama Eléctrico**

********

**Circuito**

**Imagen que contiene tabla, cuarto

Descripción generada automáticamente**

**Código**

#include <TimerOne.h>

#define BTN 2

#define LED\_1 52

#define LED\_2 53

unsigned contIn = 0, contEx = 0;*// Contadores para el timer interno y externo*

static bool outIn = LOW, outEx = LOW;*// Salidas del timer interno y externo*

void setup () {

    pinMode(LED\_1, OUTPUT);

    pinMode(LED\_2, OUTPUT);

    pinMode(BTN, INPUT\_PULLUP);

    Timer1.initialize(1000000);*// Inicializar el timer en 1 seg*

    Timer1.attachInterrupt([] {

        contIn++;

*// Cada 5 repeticiones de 1 segundo activar o desactivar LED 1*

        if (!(contIn % 5)) {

            outIn = !(outIn);

            digitalWrite(LED\_1, outIn);

        }

        return;

    });

    digitalWrite(LED\_1, LOW);

    digitalWrite(LED\_2, LOW);

}

void loop () {

    if (digitalRead(BTN) == LOW) {

        contEx++;

*// Cada 30 pulsaciones del boton activar o desactivar el LED 2*

        if (!(contEx % 30)) {

            outEx = !(outEx);

            digitalWrite(LED\_2, outEx);

        }

    }

}

**Explicación**

El programa comienza en setup(), donde se inicializa el Timer1 cada segundo. A su vez, se programa una función anónima que aumenta el contador interno cada que se ejecuta la función del timer y, cuando el contador es múltiplo de 5 (pues cinco segundos han pasado), se activa o desactiva el LED 1 usando la variable out interna.

En la función loop(), se tiene el código para el timer externo, cada que se presione un botón, se aumentará el contador externo y, cuando el contador sea múltiplo de 30 (pues se han dado 30 pulsaciones), se activa o desactiva el LED 2 usando la variable out externo.

**Observaciones**

Una práctica muy sencilla de realizar, tuve un poco de problemas configurando el botón por el efecto rebote, pero todo lo demás fue muy sencillo.

**Conclusión**

Una práctica muy potente para, saber manejar timers nos permite crear aplicaciones muy potentes, pues podemos realizar funciones asíncronas y síncronas para realizar diferente tipo de acciones.