Book: Microcontroladores PIC – Programación en C con ejemplos

Microcontroladores PIC – Programación en C con ejemplos

4.8 Ejemplo 6

Home

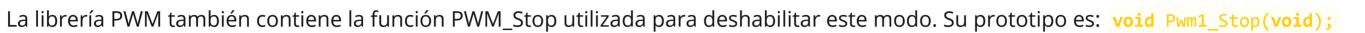
ebooks

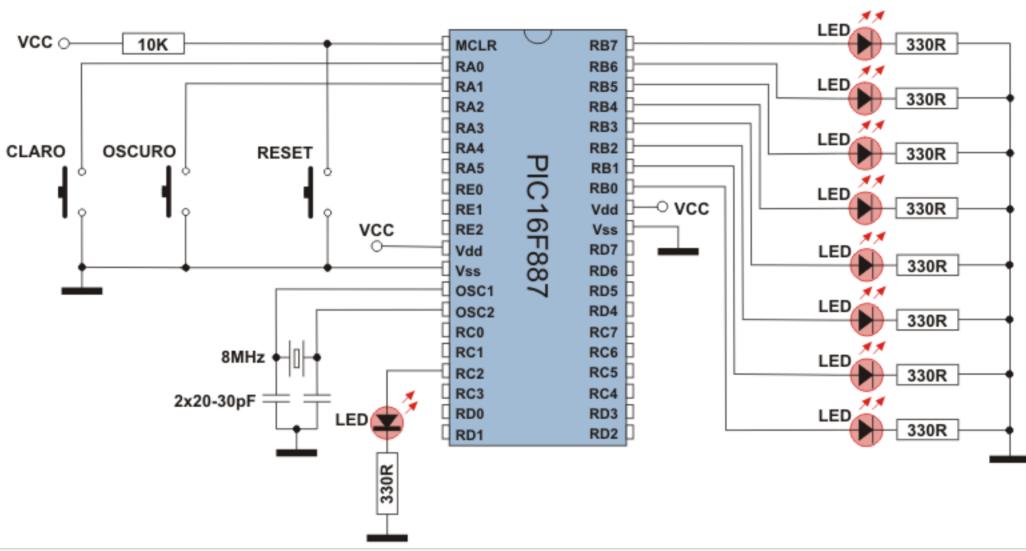
Módulo CCP1 como generador de señal PWM

Este ejemplo muestra el uso del módulo CCP1 en modo PWM. Para hacer las cosas más interesantes, la duración de los pulsos en la salida P1A (PORTC,2) se puede cambiar por medio de los botones de presión simbólicamente denominados 'OSCURO' y 'CLARO'. La duración ajustada se visualiza como una combinación binaria en el puerto PORTB. El funcionamiento de este módulo está bajo el control de las funciones pertenecientes a la librería especializada PWM. Aquí se utilizan las tres de ellas:

ejemplo-6

- 1. **PWM1_init** tiene el prototipo: void Pwm1_Init(long freq); El parámetro freq ajusta la frecuencia de la señal PWM expresada en hercios. En este ejemplo equivale a 5kHz.
- 2. **PWM1_Start** tiene el prototipo: void Pwm1_Start(void);
- 3. PWM1_Set_Duty tiene el prototipo: void Pwm1_Set_Duty(unsigned short duty_ratio); El parámetro duty_ratio ajusta la duración de pulsos en una secuencia de pulsos.

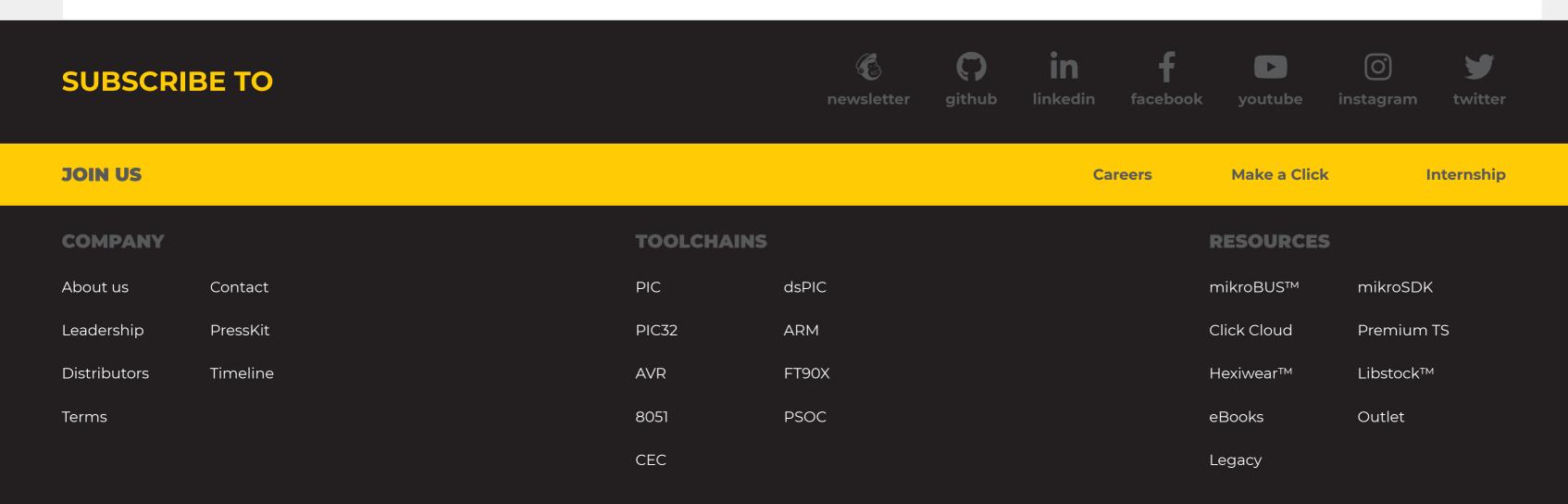




```
// Definir las variables ciclo_de_trabajo_actual,
// ciclo_de trabajo_anterior
unsigned short ciclo_de_trabajo_actual;
unsigned short ciclo_de trabajo_anterior;
void initMain() {
 ANSEL = 0;
                  // Todos los pines de E/S se configuran como digitales
 ANSELH = 0;
  PORTA = 255;
                  // Estado inicial del puerto PORTA
 TRISA = 255;
                 // Todos los pines del puerto PORTA se configuran como entradas
 PORTB = 0;
                  // Estado inicial del puerto PORTB
 TRISB = 0;
                  // Todos los pines del puerto PORTB se configuran como salidas
  PORTC = 0;
                  // Estado inicial del puerto PORTC
                  // Todos los pines del puerto PORTC se configuran
 TRISC = 0;
 // como salidas
 PWM1_Init(5000); // Inicialización del módulo PWM (5KHz)
void main() {
 initMain();
  ciclo_de_trabajo_actual = 16;  // Valor inicial de la variable ciclo_de_trabajo_actual
  ciclo_de trabajo_anterior = 0; // Reiniciar la variable ciclo_de trabajo_anterior
  PWM1_Start();
                                 // Iniciar el módulo PWM1
  while (1) {
                                 // Bucle infinito
                                 // Si se presiona el botón conectado al RA0
   if (Button(&PORTA, 0,1,1))
   ciclo_de_trabajo_actual++ ;
                               // incrementar el valor de la variable current_duty
                                 // Si se presiona el botón conectado al RA1
   if (Button(&PORTA, 1,1,1))
   ciclo_de_trabajo_actual--;
                                 // decrementar el valor de la variable current_duty
   if (old_duty != ciclo_de_trabajo_actual) {
                                                        // Si ciclo_de_trabajo_actual y
     // ciclo_de trabajo_anterior no son iguales
     PWM1_Set_Duty(ciclo_de_trabajo_actual);
                                                        // ajustar un nuevo valor a PWM,
     ciclo_de trabajo_anterior = ciclo_de_trabajo_actual; // Guardar el nuevo valor
     PORTB = ciclo_de trabajo_anterior;
                                                        // y visualizarlo en el puerto PORTB
   Delay_ms(200); // Tiempo de retardo de 200mS
```

Para que este ejemplo funcione apropiadamente, es necesario marcar las siguientes librerías en la ventana Library Manager antes de compilar el programa:

- PWM
- Button



Copyright© 2019 MikroElektronika d.o.o. **Privacy**