

PUNTEROS A ESTRUCTURAS Y FUNCIONES

INTRODUCCIÓN DE ESTRUCTURAS Y PUNTEROS
USO DE MACROS (BÁSICAS) - EJEMPLOS
CALLBACKS

Repaso de estructuras.

- Es un tipo de dato creado por el usuario
- Sirve para agrupar los datos de distintos tipos dentro de una única variable
- Ejemplo: deseamos crear una función que realice la inicialización de un puerto de un microcontrolador. Además deseamos realizar algún tipo de acción que debe ser introducida por otro programador.
- Un puerto en estas condiciones, tiene tres parámetros
 - Número de puerto
 - Estado del puerto
 - acción a ejecutar



Generamos
La estructura

```
typedef struct {  
    uint8_t port ;  
    uint8_t state;  
    /// ¿Acción a ejecutar?  
}led_t
```

PUNTEROS A FUNCIONES

- Se utilizan cuando se desconoce la acción a realizar.
- Por ejemplo pueden utilizarse dentro de ISR o IRQ.
- Cada función tiene un espacio asignado de memoria.
- Este puntero en lugar de apuntar una variable apunta al inicio de la función.

Sintaxis de C para definir un puntero a función:

`<tipo de dato> (*nombre)(tipo1 var1, .., tipoN varN)`

- Por ejemplo:
 - `Int (*función)(int,int)`: Este puntero puede apuntar a cualquier función de este tipo

¿Acción a ejecutar por parte del puerto?

- Definimos un campo de tipo puntero a función.
- La estructura queda de la siguiente forma:

```
typedef void (*callback_fn)(uint16_t a);  
typedef struct {  
    uint8_t port;  
    uint8_t state;  
    callback_fn fn;  
} led_t;
```

PUNTEROS A ESTRUCTURAS

- EL PUNTERO APUNTA A UNA VARIABLE DEL TIPO CREADAS POR EL USUARIO
- La sintaxis es identica a la de los punteros ya vista, solo se cambia el tipo de dato.
- Para acceder a un campo utilizando un puntero a la estructura se utiliza el operador “->”
- Lo utilizamos para apuntar la estructura anterior
- Creamos tres funciones: inicializar la estructura, cambiar el estado y que realice la ejecución al cambiar el estado.



Veremos todos estos conceptos con un ejemplo

OFF-TOPIC

- IRQ o INTERRUPCIONES MEDIANTE TIMERS
- VEREMOS EL uC Atmega328P
- POSEE TRES TIMERS
- Dentro de la Hoja de datos (datasheet se denominan TIMER0-TIMER 1 y TIMER 2). Haremos uso del TIMER2.

ARDUINO – UNO ATMEGA328p

CADA TIMER POSEE CUATRO MODOS

CTC

MODO FAST PWM

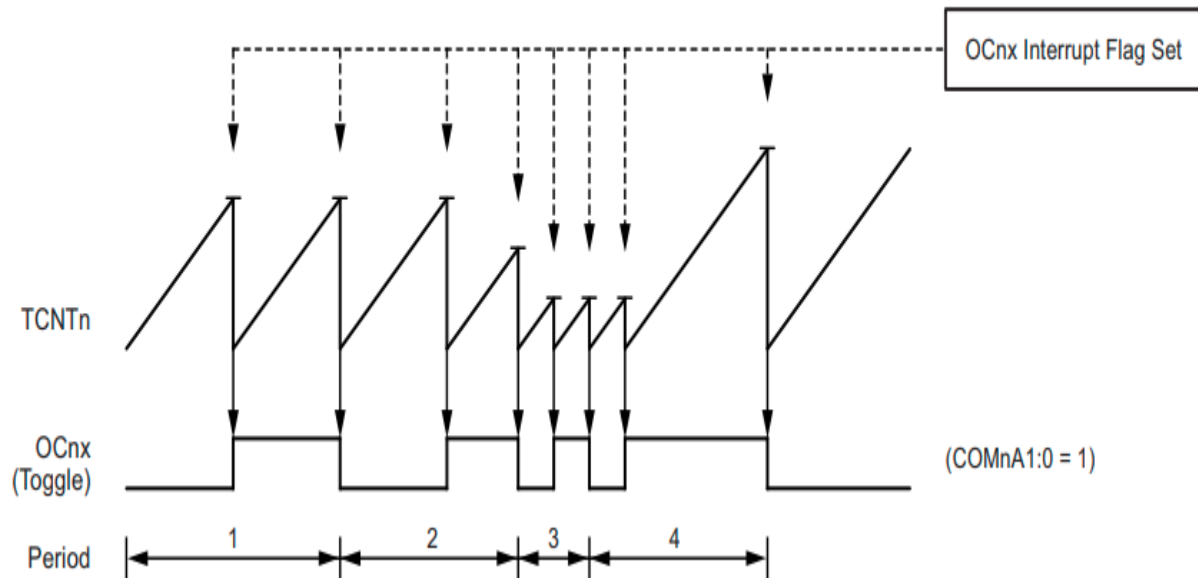
MODO PHASE
CORRECT

NORMAL

FUNCIONAMIENTO MODO CTC

$$f_{OCnx} = \frac{f_{clk_I/O}}{2 \times N \times (1 + OCRnx)}$$

Figure 17-5. CTC Mode, Timing Diagram



Ejemplo:

$F_{clk_I/O} = 16 \text{ Mhz}$

$N = 1, 8, 32, 64, 128, 256, 1024$

$OCRnX = 8 \text{ BITS (0 A 255)}$

Ejemplo:

si $N=8$, $OCRnx=9$

$F_{ocnx} = 100 \text{ KHz}$

CONFIGURACIÓN TIMER 2

17.11.1 TCCR2A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0
(0xB0)	COM2A1	COM2A0	COM2B1	COM2B0	—	—	WGM21	WGM20
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

TCCR2A

17.11.2 TCCR2B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0
(0xB1)	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20
Read/Write	W	W	R	R	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

TCCR2B

17.11.4 OCR2A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
(0xB3)	OCR2A[7:0]								OCR2A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

17.11.6 TIMSK2 – Timer/Counter2 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0
(0x70)	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

TIMSK2

CONFIGURACIÓN TIMER 2

Table 17-9. Clock Select Bit Description

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stop)
0	0	1	$\text{clk}_{T2S}/(\text{no prescaling})$
0	1	0	$\text{clk}_{T2S}/8$ (from prescaler)
0	1	1	$\text{clk}_{T2S}/32$ (from prescaler)
1	0	0	$\text{clk}_{T2S}/64$ (from prescaler)
1	0	1	$\text{clk}_{T2S}/128$ (from prescaler)
1	1	0	$\text{clk}_{T2S}/256$ (from prescaler)
1	1	1	$\text{clk}_{T2S}/1024$ (from prescaler)

```
SREG = (SREG & 0b01111111); // deshabilito interrupciones globales
TCNT2 = 0 ;
TIMSK2 = TIMSK2 | 0b00000010 ; //
TCCR2A = 0b00000010;
TCCR2B = 0b00000100; // 250 kHz n= 249
OCR2A = 249; //valor de comparacion
SREG = (SREG & 0b01111111) | 0b10000000 ;// habilito interrupciones globales
```

Table 17-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, phase correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF

¿Cada cuanto se ejecuta la IRQ?



