

# ARDUINO AVANZADO

Miguel Angel Ruiz Gálvez

Visita: [miguelo.me](http://miguelo.me)

Material en: <http://goo.gl/UO3xix>

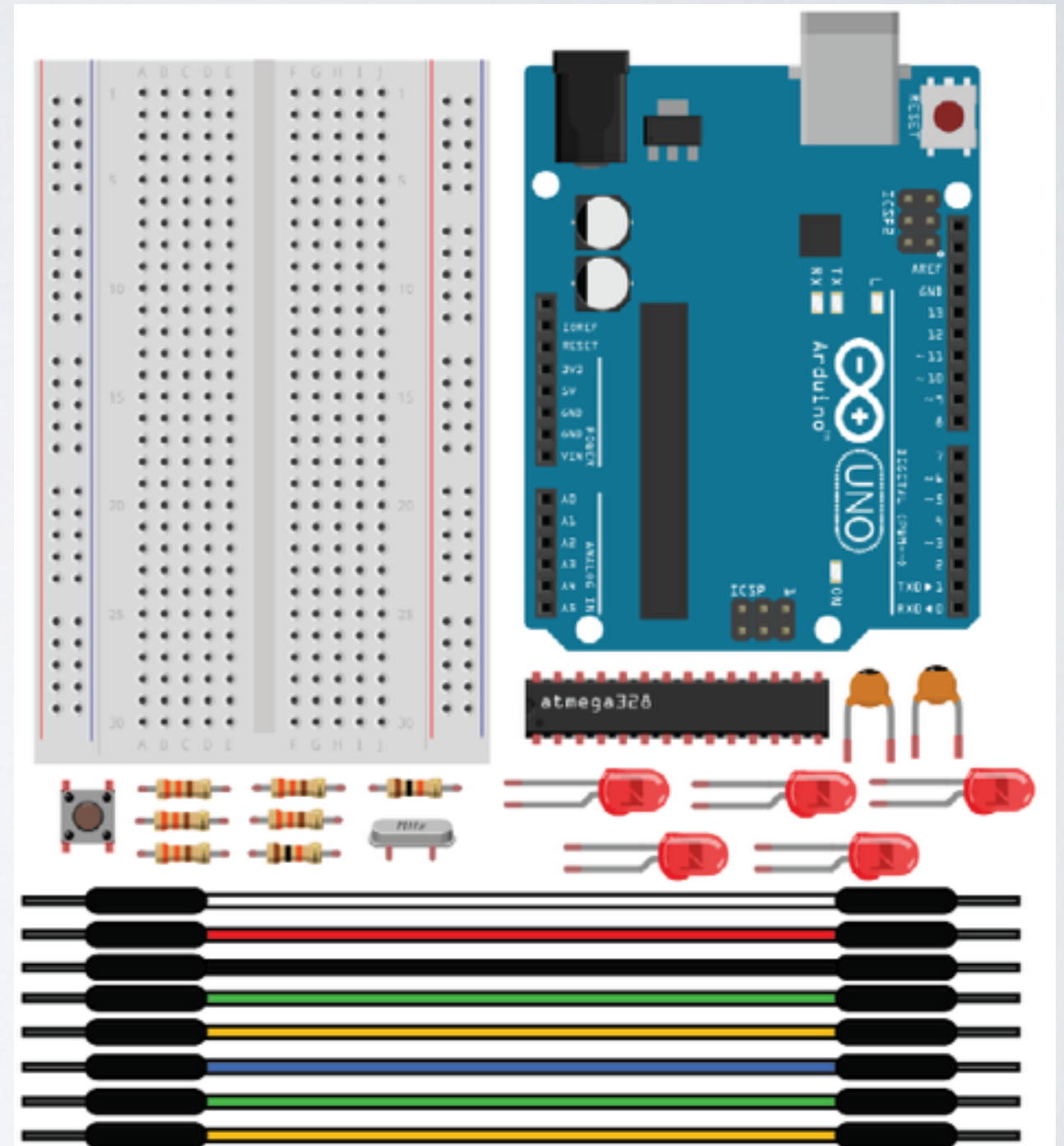
---

SOMEFI

Este documento está licenciado bajo la Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita <http://creativecommons.org/licenses/by-nc-sa/4.0/>.

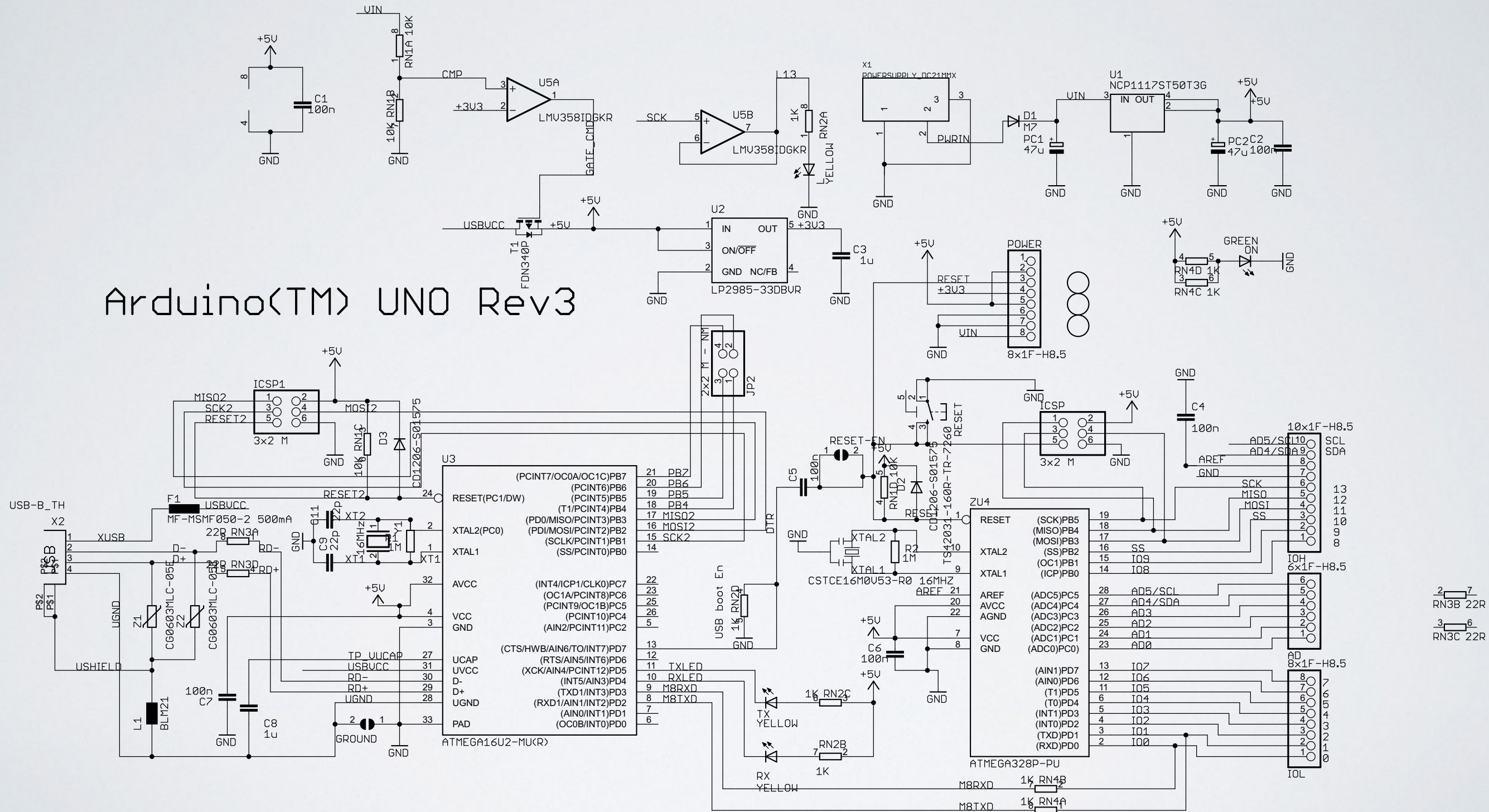
# MATERIAL A UTILIZAR

- Tarjeta de desarrollo Arduino UNO y cable.
- 1 Protoboard
- 1 Microcontrolador ATmega 328-UP
- 1 Cristal oscilador de 16 MHz
- 5 LED's
- 5 Resistencias de  $300\ \Omega$ ,  $\frac{1}{4}$  [W]
- 1 Pushbutton
- 2 Resistencias de  $10\ \text{K}\Omega$ ,  $\frac{1}{4}$  [W]
- 2 Capacitores de 22pf
- 20 Jumpers



# ARDUINO

## Arduino(TM) UNO Rev3



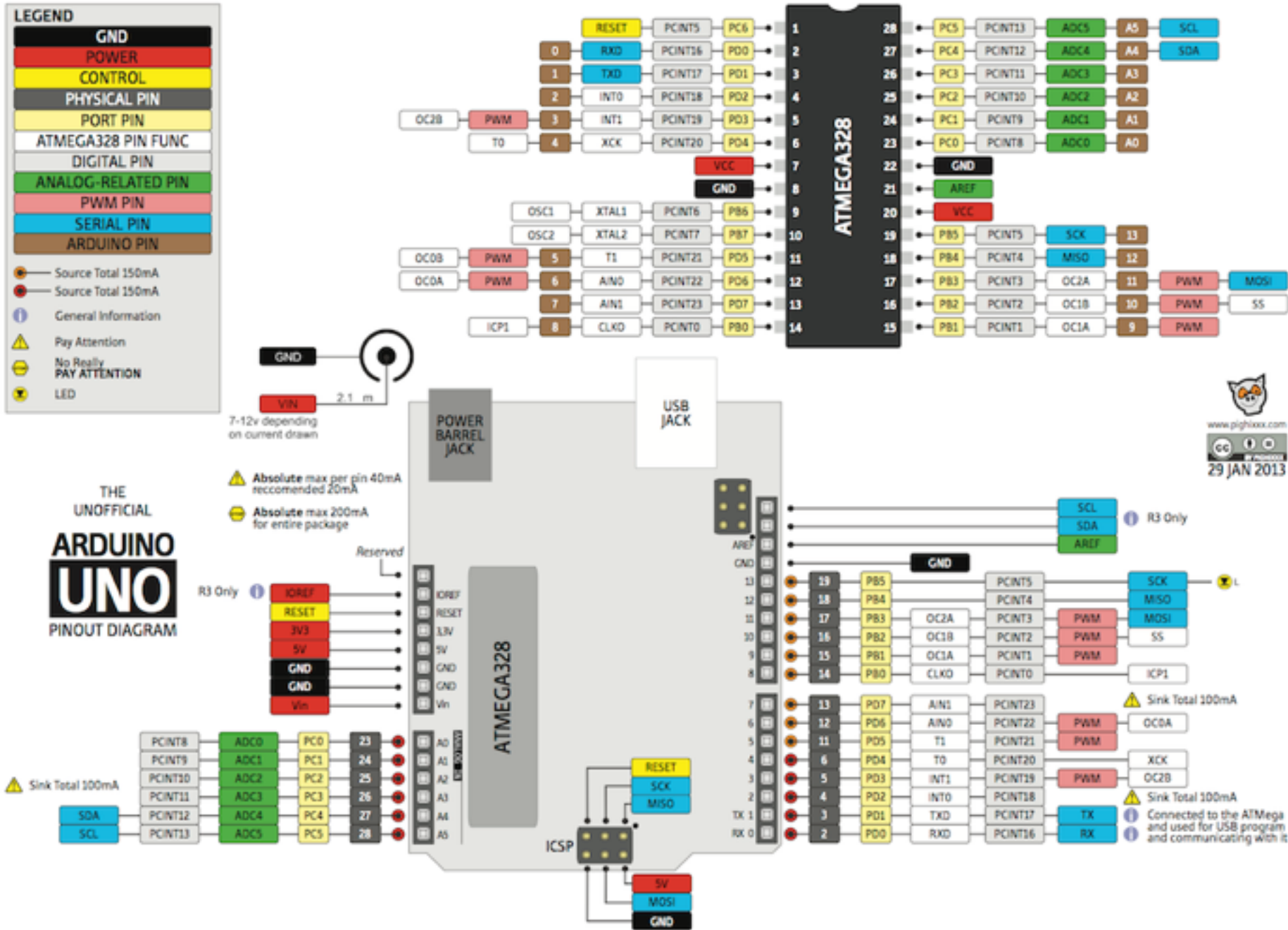


# ATMEGA 328-UP

## High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller.

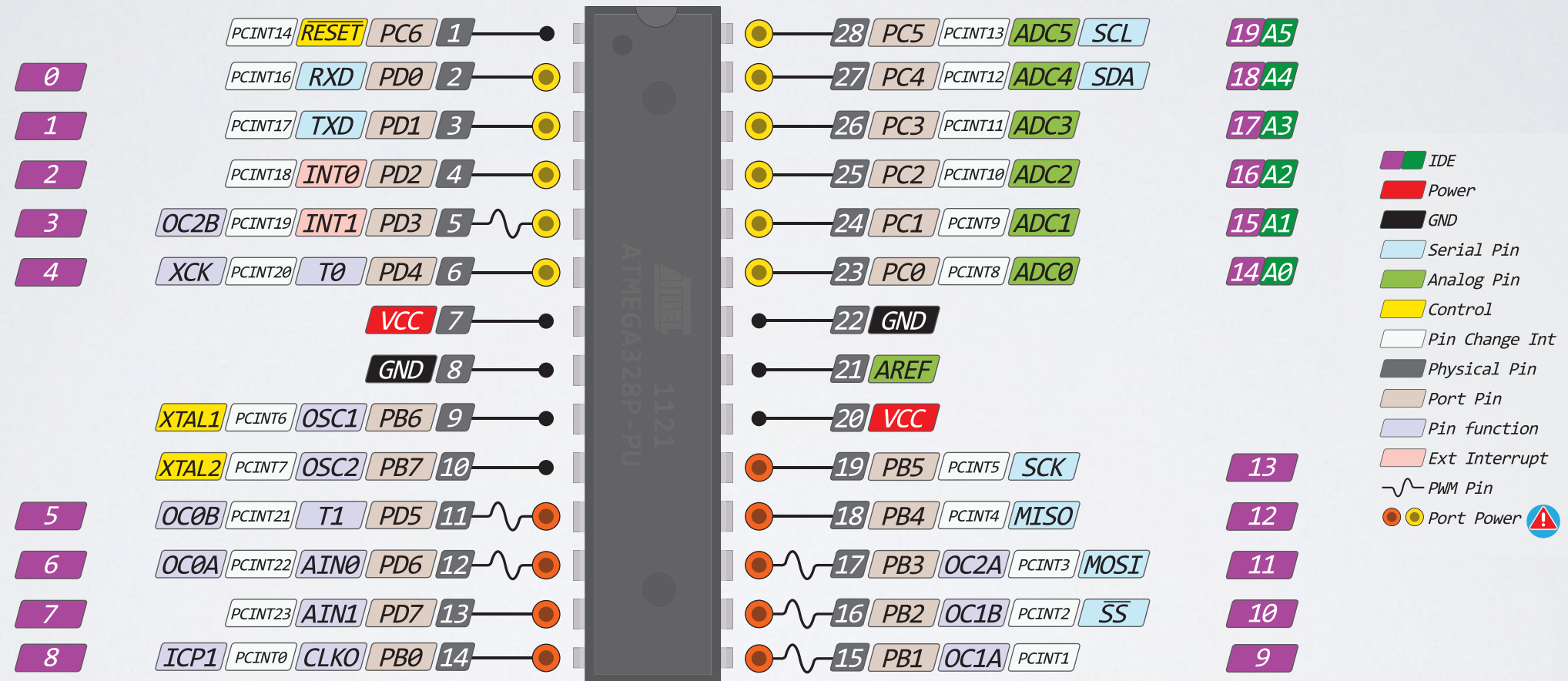
- Voltaje de Operacion: 1.8 - 5.5V / 0-20MHz
- 32K Bytes en Flash/1K Bytes en EEPROM/ 2K Bytes Internal SRAM.
- Ciclos de memoria: 10,000 Flash/100,000 EEPROM. Retención de información: 20 años a 85°C/100 años a 25°C.
- 2 Timer/Counters 8-bit con preescladores y comparadores independientes.
- 1 Timer/Counters 16-bit con modulo de captura, comparación y prescalador independiente.
- 6 canales de PWM.
- 6-canales 10-bit ADC en PDIP (8-canales 10-bit ADC en TQFP y QFN/MLF).
- 23 I/O.
- 6 Sleep Modes y 3 Low Power Consumption
- Comunicaciones USART, SPI, I<sup>2</sup>C.
- Watchdog Timer con Oscilador independiente. Interrupción y Wake-up con cambio de flanco.

# ATMEGA 328-UP



# ATMEGA 328-UP

## ATMEGA328 PINOUT



⚠ Absolute MAX per pin 40mA  
recommended 20mA

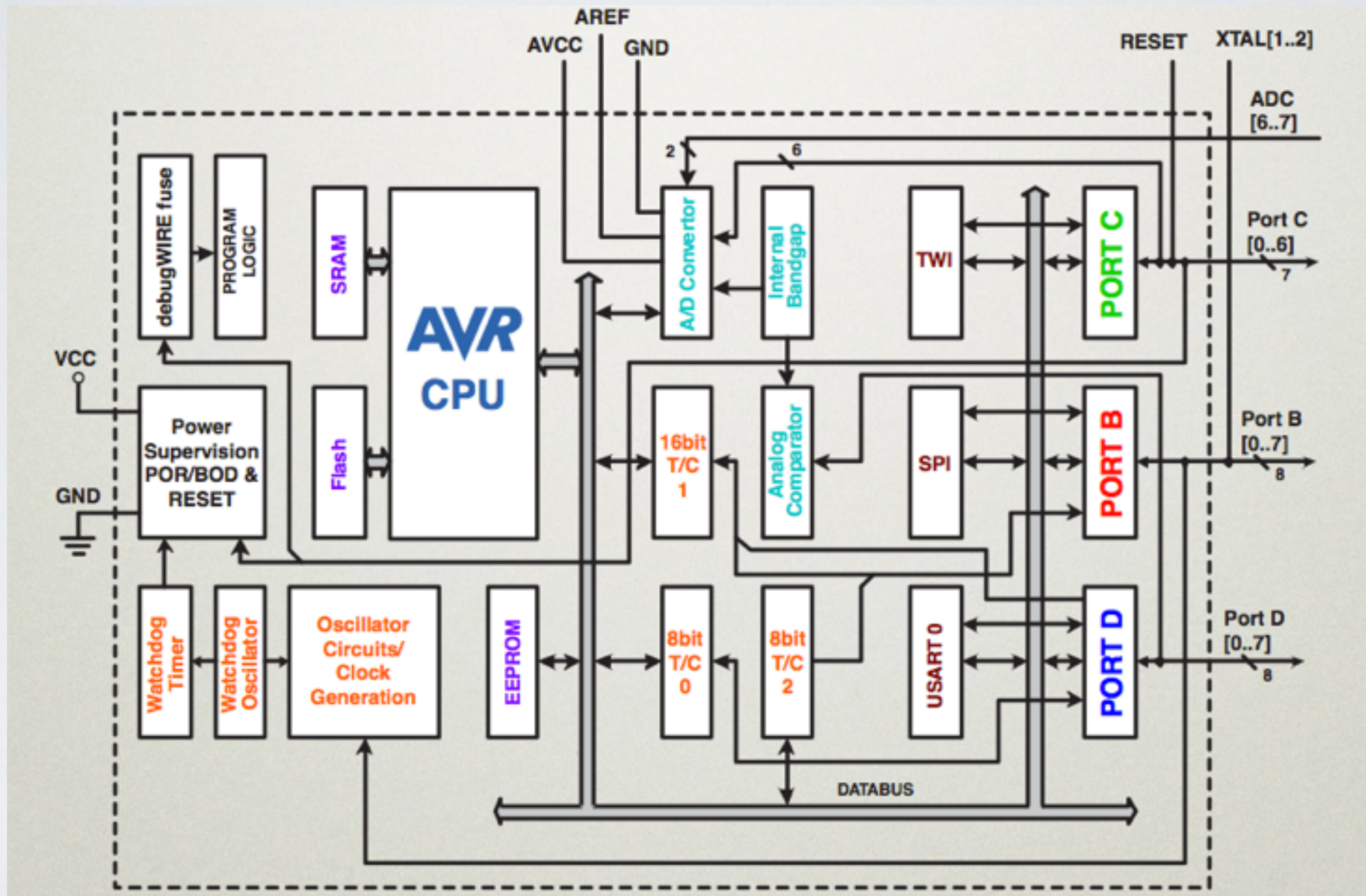
⚠ Absolute MAX 200mA  
for entire package

The power sum for each pin's  
group should not exceed 100mA ⚠

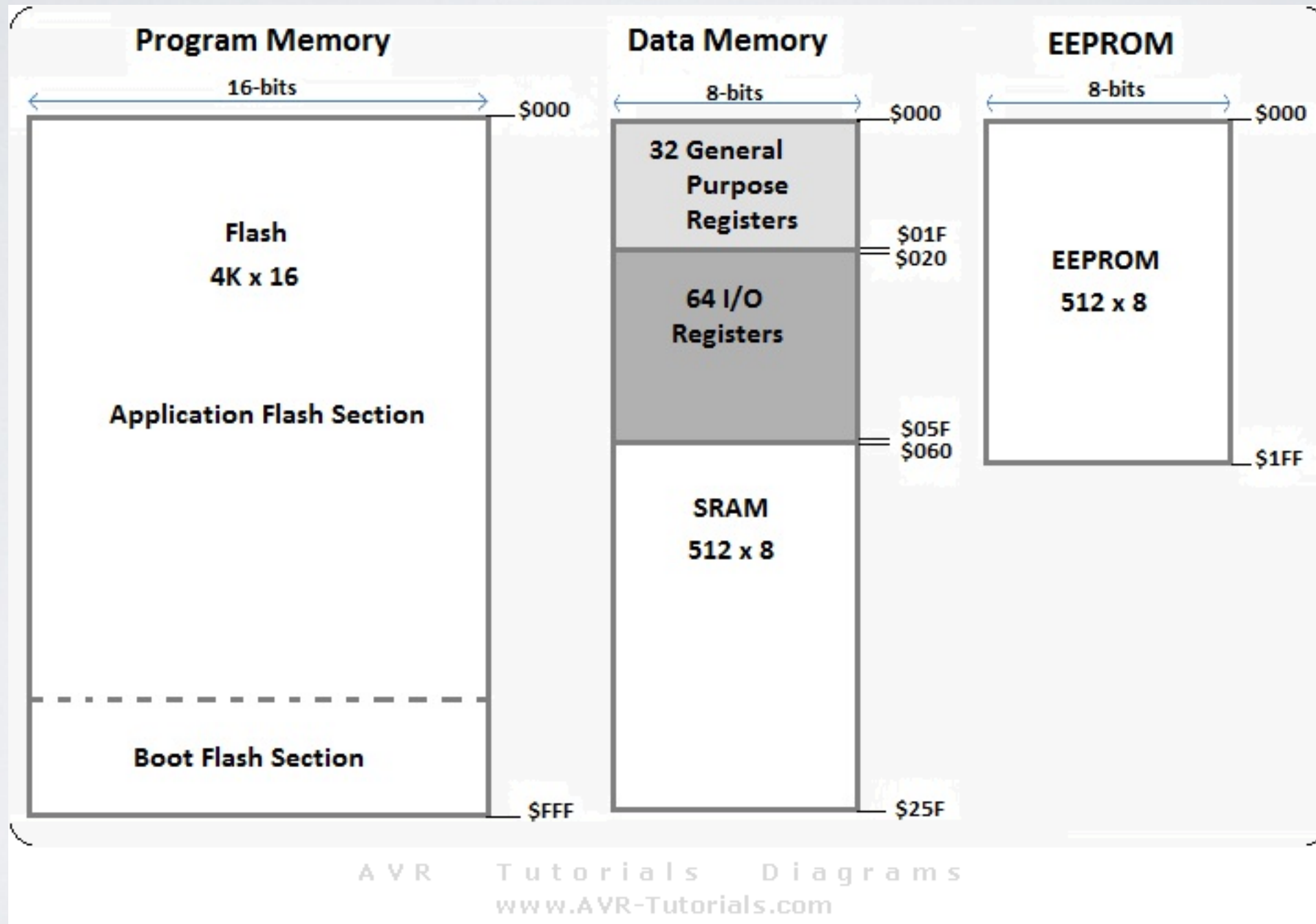




# ATMEGA 328-UP



# ATMEGA 328-UP





# ARDUINO/ C / ENSAMBLADOR

Función	Arduino	C	ASM
Facilidad	Facil	Medio	Complicado
Librerías	Muchas	Muchas	Muy pocas
Velocidad	Lento	Rapido	Máxima Velocidad
Acceso a procesos	Muy limitado	Poco Limitado	Completo
Comunidad	Muy grande	Media	Poca

# ESTRUCTURA BASICA CON C

## Arduino

```
void setup() {  
    }  
void loop(){  
    }  
}
```

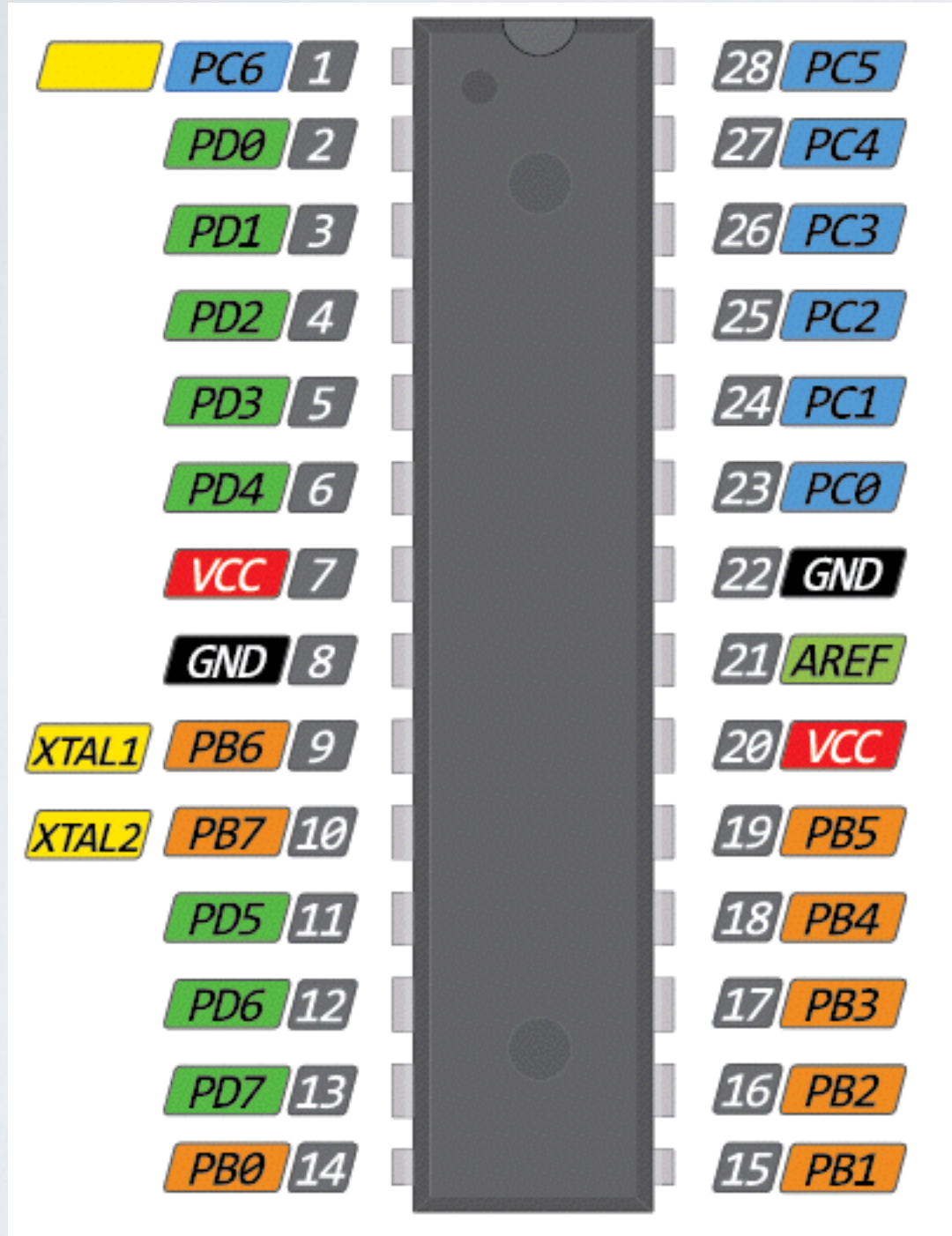
121 KHz

## C/C++

```
int main(void) {  
    // configuración  
    while (1) {  
        //loop  
    }  
    return (0);  
}
```

2.6 MHz

# GPIOS



Voltaje máximo aplicado I/O: 5.5v  
Corriente máxima aplicada I/O: 40mA  
Máxima corriente de salida I/O: 20mA  
Corriente máxima de Vcc: 200mA  
Corriente máxima de GND: 200mA  
Corriente recomendada: I/2 de máx.  
Resistencias Pullup: 20k $\Omega$   
Estado de alta Impedancia (Input): 100M $\Omega$



# CONTROL GPIOS

**PORTx:** controla el contenido de los puertos.

**DDRx:** controla la configuración de los puertos.  
(Entrada o Salida).

**PINx:** permite leer el valor de cada pin.

**x = B, C, D.**

# PUERTO B

XTAL 1 y 2

PORTB

Bit	7	6	5	4	3	2	1	0
0x25	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

0: HIGH  
1: LOW

DDRB

Bit	7	6	5	4	3	2	1	0
0x24	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

0: Input  
1: Output

PINB

Bit	7	6	5	4	3	2	1	0
0x23	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
R/W?	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

0: HIGH  
1: LOW

# PUERTO C

## RESET

PORTC

Bit	7	6	5	4	3	2	1	0
0x28	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

0: HIGH  
1: LOW

DDRC

Bit	7	6	5	4	3	2	1	0
0x27	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

0: Input  
1: Output

PINC

Bit	7	6	5	4	3	2	1	0
0x26	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
R/W?	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

0: HIGH  
1: LOW



# PUERTO D

PORD

Bit	7	6	5	4	3	2	1	0
0x2B	<b>PORTD7</b>	<b>PORTD6</b>	<b>PORTD5</b>	<b>PORTD4</b>	<b>PORTD3</b>	<b>PORTD2</b>	<b>PORTD1</b>	<b>PORTD0</b>
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

0: HIGH  
1: LOW

DDRD

Bit	7	6	5	4	3	2	1	0
0x2A	<b>DDRD7</b>	<b>DDRD6</b>	<b>DDRD5</b>	<b>DDRD4</b>	<b>DDRD3</b>	<b>DDRD2</b>	<b>DDRD1</b>	<b>DDRD0</b>
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

0: Input  
1: Output

PIND

Bit	7	6	5	4	3	2	1	0
0x29	<b>PIND7</b>	<b>PIND6</b>	<b>PIND5</b>	<b>PIND4</b>	<b>PIND3</b>	<b>PIND2</b>	<b>PIND1</b>	<b>PIND0</b>
R/W?	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

0: HIGH  
1: LOW

# CONFIGURACIÓN

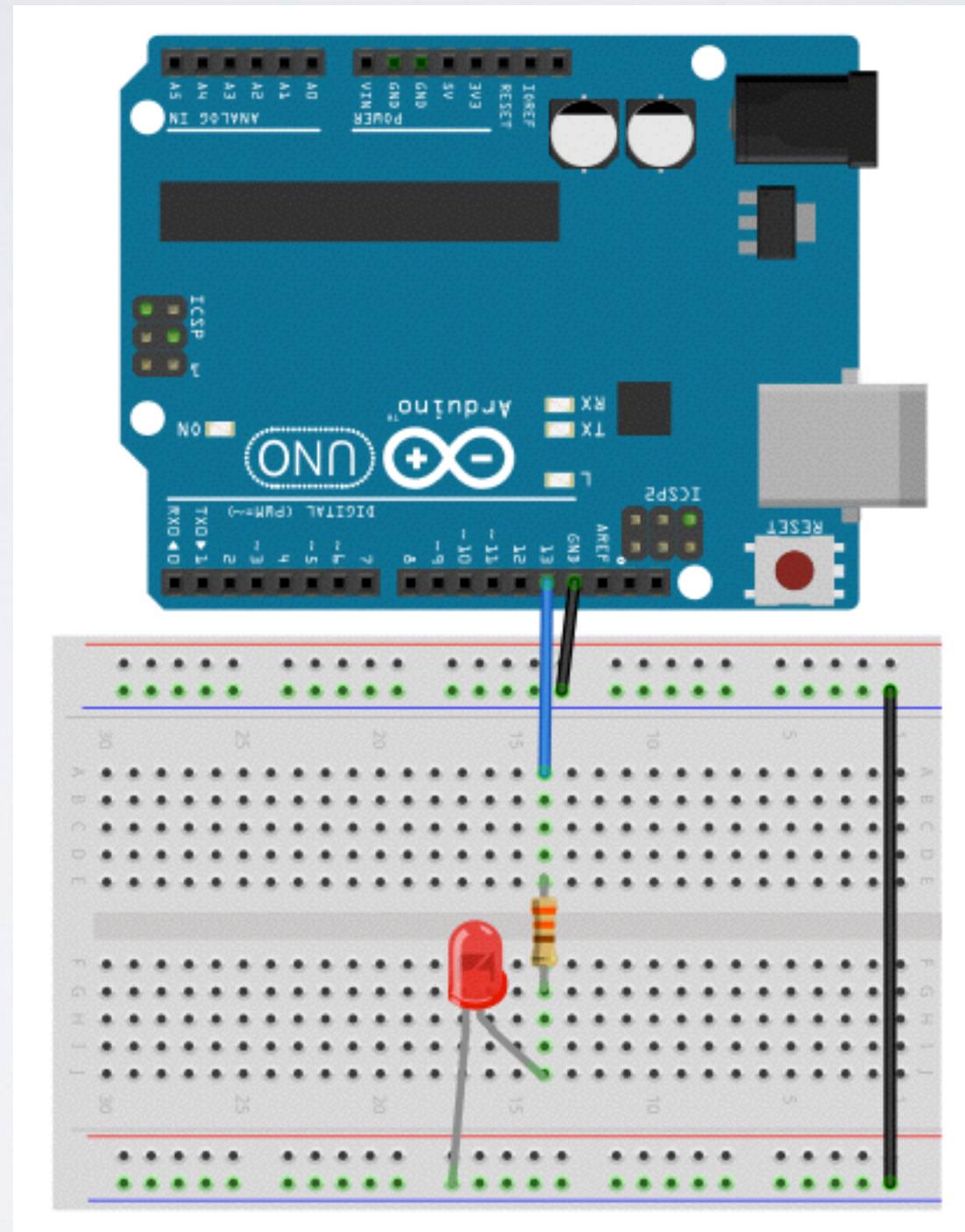
Acción	DDR	PORT	PUD	I/O	Pull-Up
Tri-estado (Hi-Z)	0	x	X	Input	No
Tri-estado (Hi-Z) con pull-up	0	1	0	Input	Si
LOW	1	0	x	Output	No
HIGH	1	1	x	Output	No

MCUCR

Bit	7	6	5	4	3	2	1	0
0x23	-	<b>BODS</b>	<b>BODSE</b>	<b>PUD</b>	-	-	<b>IVSEL</b>	<b>IVCE</b>
R/W?	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

\*Pull Up Desable

# BLINK EN C





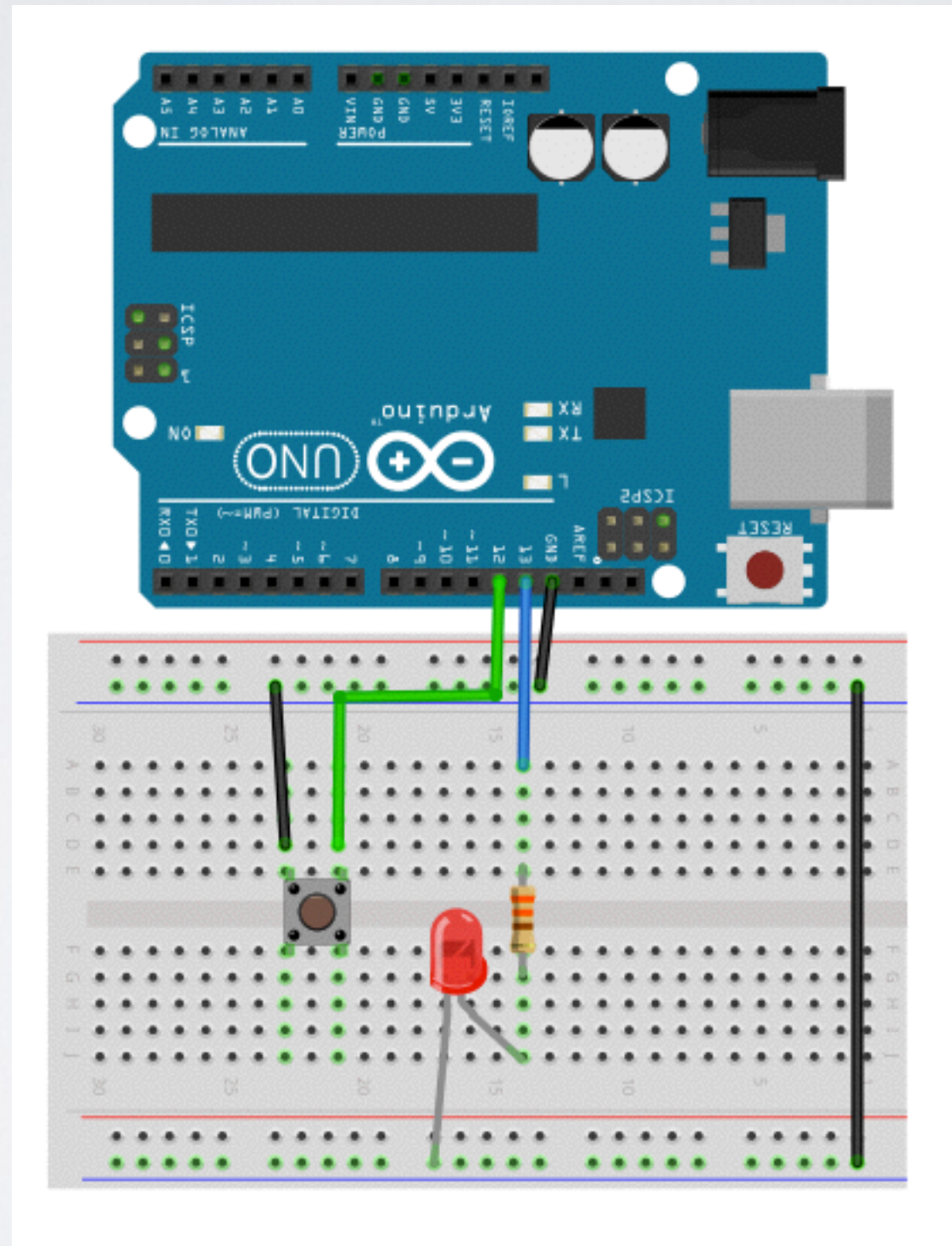
# MASCARAS

Las mascararas de utilizan para modificar un bits específico dentro de un byte facilitando en manejo de registros.

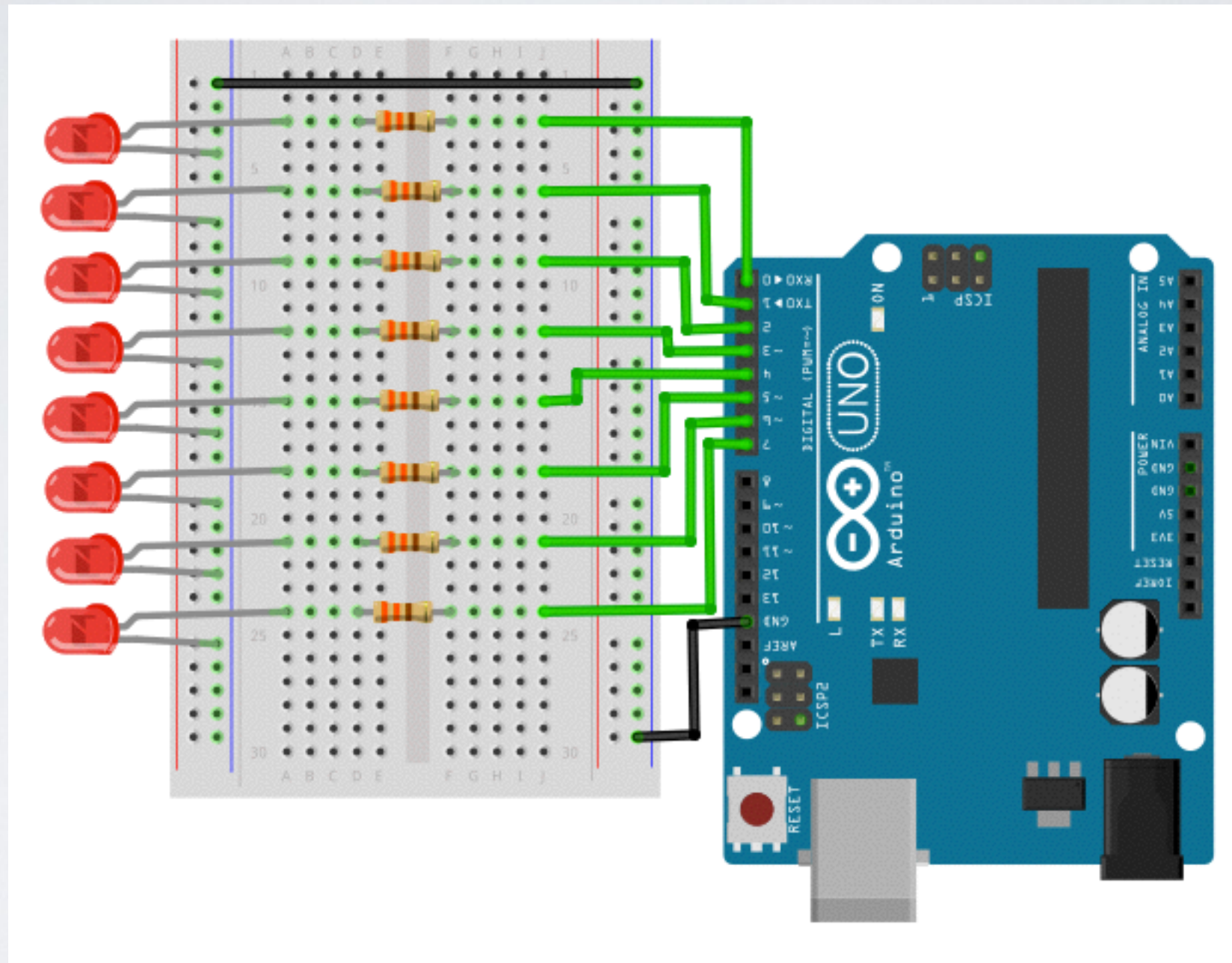
**byte** registro = 0b11110000;

Simbolo	Nombre	Ejemplo
=	OR	registro  = 0b00001010;
&=	AND	registro &= 0b01011111;
(1 << casilla )	Asignación de byte	registro  = (1 << 3); registro &= (1 << 5);

# BOTTON



# KIT C





# TIMERS

- Un Timer es una pieza de hardware interna, que se utiliza para medir acontecimientos del tiempo.
- El uC ATmega328 tiene 3 timers (Timer 0, Timer1 y Timer2). Los Timer0 y Timer2 tienen un contador de 8 bits (0 a 255) , mientras que el Timer1 tiene un contador de 16 bits. (0 a 65535).
- El uC Atmega2560 (Arduino Mega) cuenta con 6 timers, los Timer 0, Timer1 y Timer2 son idénticos al ATmega328, mientras el Timer 3, Timer4 y Timer5 son de 16 bits.
- Todos los Timers depende del reloj del sistema, normalmente 16MHz.
- En el firmware Arduino todos los temporizadores se configuraron a una frecuencia de 1 kHz (preescala a 64) y las interrupciones no están habilitadas.

# ARDUINO Y TIMERS

Timer	Uso
Timer 0	<code>delay()</code> , <code>millis()</code> y <code>micros()</code>
Timer 1	<code>Servo()</code> , <code>analogWrite()</code>
Timer 2	<code>tone()</code> , <code>analogWrite()</code>

# TIMERS

- Timer/Counter Control Register A:  $TCCR_nA$
- Timer/Counter Control Register B:  $TCCR_nB$
- Timer Counter:  $TCNT_n$
- Output Compare Register A:  $OCR_nA$
- Output Compare Register B:  $OCR_nB$
- Timer/Counter Interrupt Mask Register:  $TIMSK_n$

$n = 0, 1, 2$



# TIMER 0

TCCR0A

Bit	7	6	5	4	3	2	1	0
0x44	COM0A1	COM0A0	COM0B1	COM0B0	—	—	WGM01	WGM00
R/W?	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

TCCR0B

Bit	7	6	5	4	3	2	1	0
0x44	FOC0A	FOC0B	—	—	WGM02	CS02	CS01	CS00
R/W?	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

\***C**OMpare **M**atch Output Timer **0 A/B** Mode **Bit**

\***F**orce **O**utput **C**ompare Timer **0 A/B**

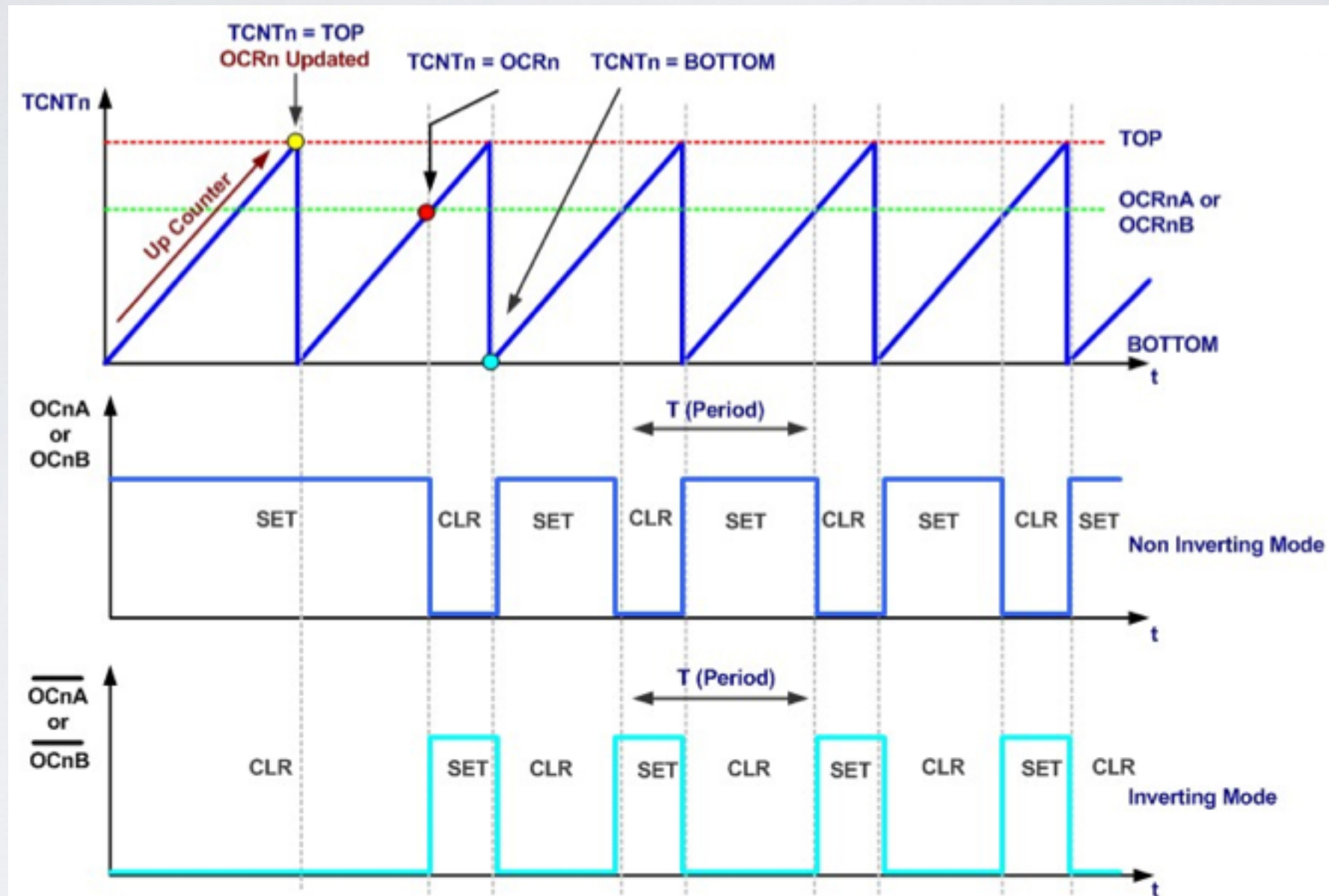
# TIMER 0

## Waveform Generation Mode Timer 0 Bit

**Table 14-8.** Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on <sup>(1)(2)</sup>
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

# FAST PWM



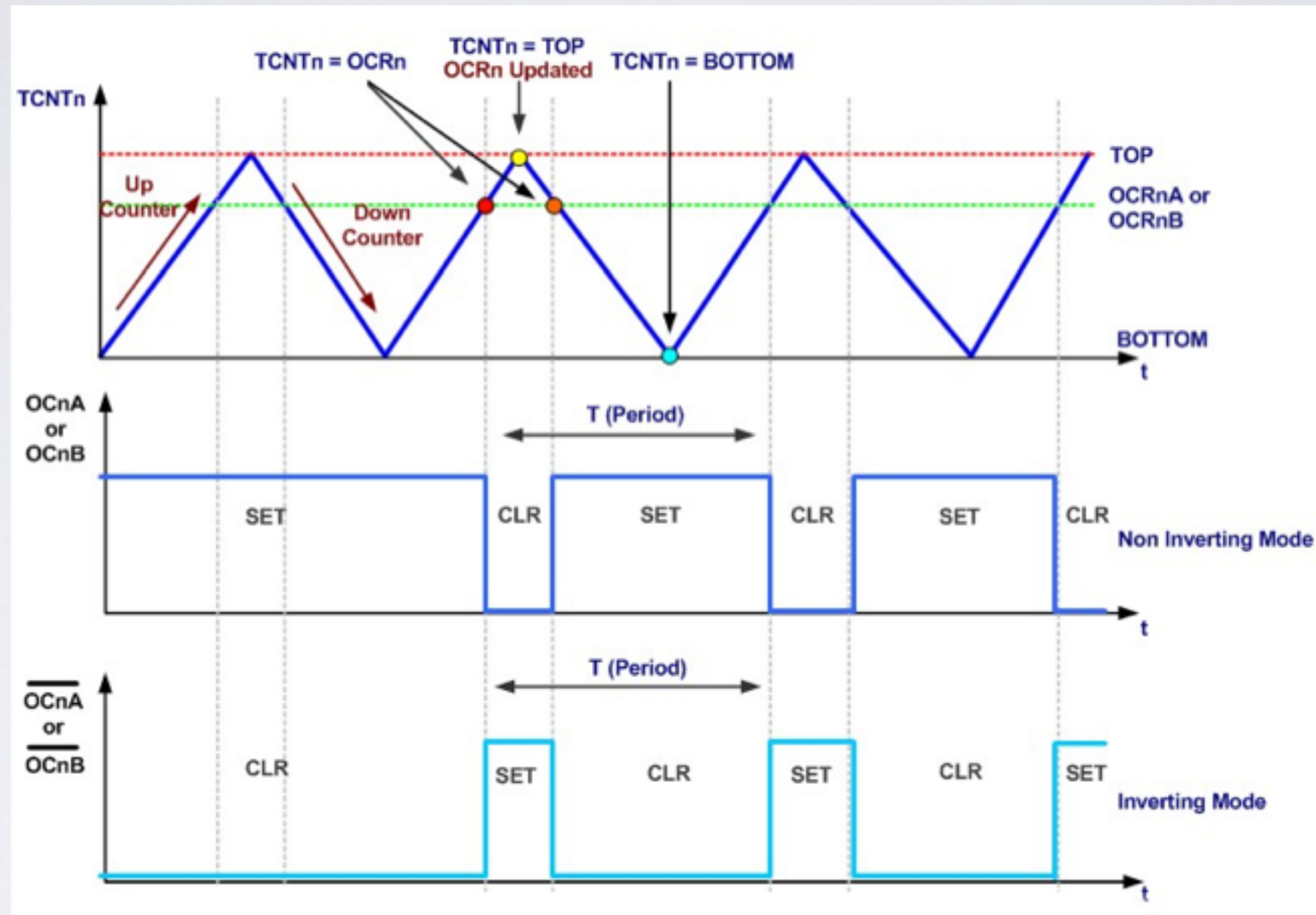
Frecuencia

$$f_{PWM} = f_{clk} / 256N$$

$N$  = prescalador



# PHASE CORRECT PWM



Frecuencia

$$f_{PWM} = f_{clk} / 510N$$

$N$  = prescalador

# TIMER 0

## Clock Select Timer 0 Bit

**Table 14-9.** Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$\text{clk}_{\text{I/O}}$ /(No prescaling)
0	1	0	$\text{clk}_{\text{I/O}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{I/O}}/64$ (From prescaler)
1	0	0	$\text{clk}_{\text{I/O}}/256$ (From prescaler)
1	0	1	$\text{clk}_{\text{I/O}}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

# SELECCIÓN DE ESCALA

Tiempo deseado : 30 KHz

clk: 16Mhz

$$T_{\text{deseado}} = 1 / 30 \text{ KHz} = 0.033 \text{ ms}$$

CS0	Escala	Frecuencia	Tick [ms]	Tdeseado/Ticks	Precarga
0 0 1	clk/1	16 MHz	0.0000625	528	no cabe
0 1 0	clk/8	2 MHz	0.0005	66	255 - 66 = 189
0 1 1	clk/64	250 KHz	0.004	8.25	255 - 8 = 247
1 0 0	clk/256	62.5 KHz	0.016	2.0625	255 - 2 = 253
1 0 1	clk/1024	15.625 KHz	0.064	0.556	255 - 1 = 255



# TIMER 0

TCNT0

Bit	7	6	5	4	3	2	1	0
0x46	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

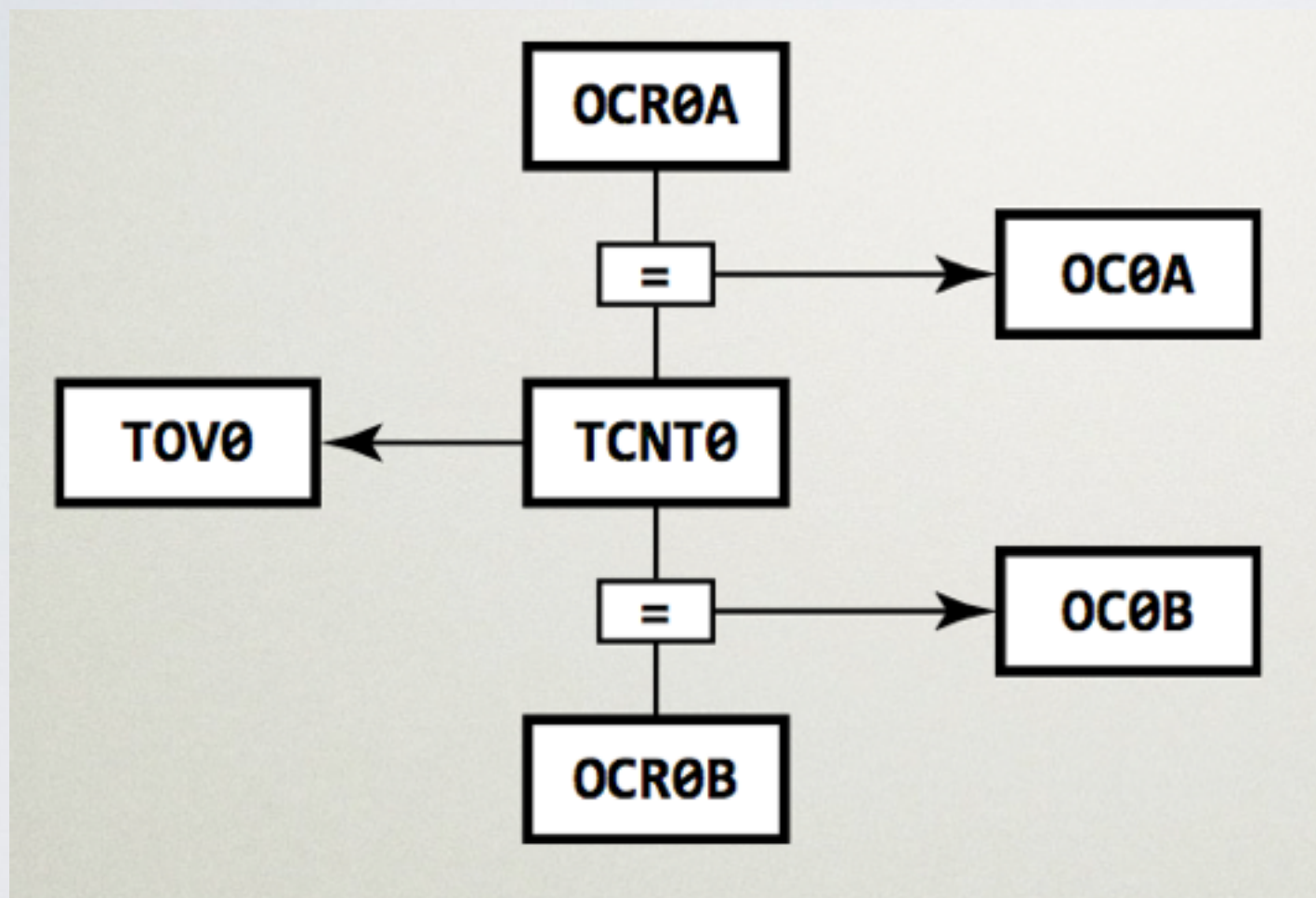
OCR0A

Bit	7	6	5	4	3	2	1	0
0x47	OCR0A7	OCR0A6	OCR0A5	OCR0A4	OCR0A3	OCR0A2	OCR0A1	OCR0A0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

OCR0B

Bit	7	6	5	4	3	2	1	0
0x48	OCR0B7	OCR0B6	OCR0B5	OCR0B4	OCR0B3	OCR0B2	OCR0B1	OCR0B0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

# TIMER 0 INTERRUPTCIONES



- Overflow interrupt (**TOV0**)
- Output compare match interrupts (**OCF0A** y **OCF0B**)

# TIMER 0 INTERRUPTCIONES

TIMSK0

Bit	7	6	5	4	3	2	1	0
0x6E	—	—	—	—	—	OCIE0B	OCIE0A	TOIE0
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

TIFR0

Bit	7	6	5	4	3	2	1	0
0x35	—	—	—	—	—	OCF0B	OCF0A	TOV0
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0



# TIMER 0 INTERRUPTACIONES

```
ISR(TIMERO_OVF_vect){
```

```
}
```

```
ISR(TIMERO_COMPA_vect){
```

```
}
```

```
ISR(TIMERO_COMPB_vect){
```

```
}
```

# TIMER 2

TCCR2A

Bit	7	6	5	4	3	2	1	0
0xB0	COM2A1	COM2A0	COM2B3	COM2B4	—	—	WGM21	WGM20
R/W?	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

TCCR2B

Bit	7	6	5	4	3	2	1	0
0xB1	FOC2A	FOC2B	—	—	WGM22	CS22	CS21	CS20
R/W?	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

\***CO**mpare **M**atch Output Timer **2 A/B** Mode **Bit**

\***F**orce **O**utput **C**ompare Timer **2 A/B**

# TIMER 2

## Clock Select Timer 2 Bit

**Table 17-9.** Clock Select Bit Description

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
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)



# TIMER 2

TCNT2

Bit	7	6	5	4	3	2	1	0
0xB2	TCNT27	TCNT26	TCNT25	TCNT24	TCNT23	TCNT22	TCNT21	TCNT20
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

OCR2A

Bit	7	6	5	4	3	2	1	0
0xB3	OCR2A7	OCR2A6	OCR2A5	OCR2A4	OCR2A3	OCR2A2	OCR2A1	OCR2A0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

OCR2B

Bit	7	6	5	4	3	2	1	0
0xB4	OCR2B7	OCR2B6	OCR2B5	OCR2B4	OCR2B3	OCR2B2	OCR2B1	OCR2B0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

# TIMER 2 INTERRUPTCIONES

TIMSK2

Bit	7	6	5	4	3	2	1	0
0x70	—	—	—	—	—	OCIE2B	OCIE2A	TOIE2
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

TIFR2

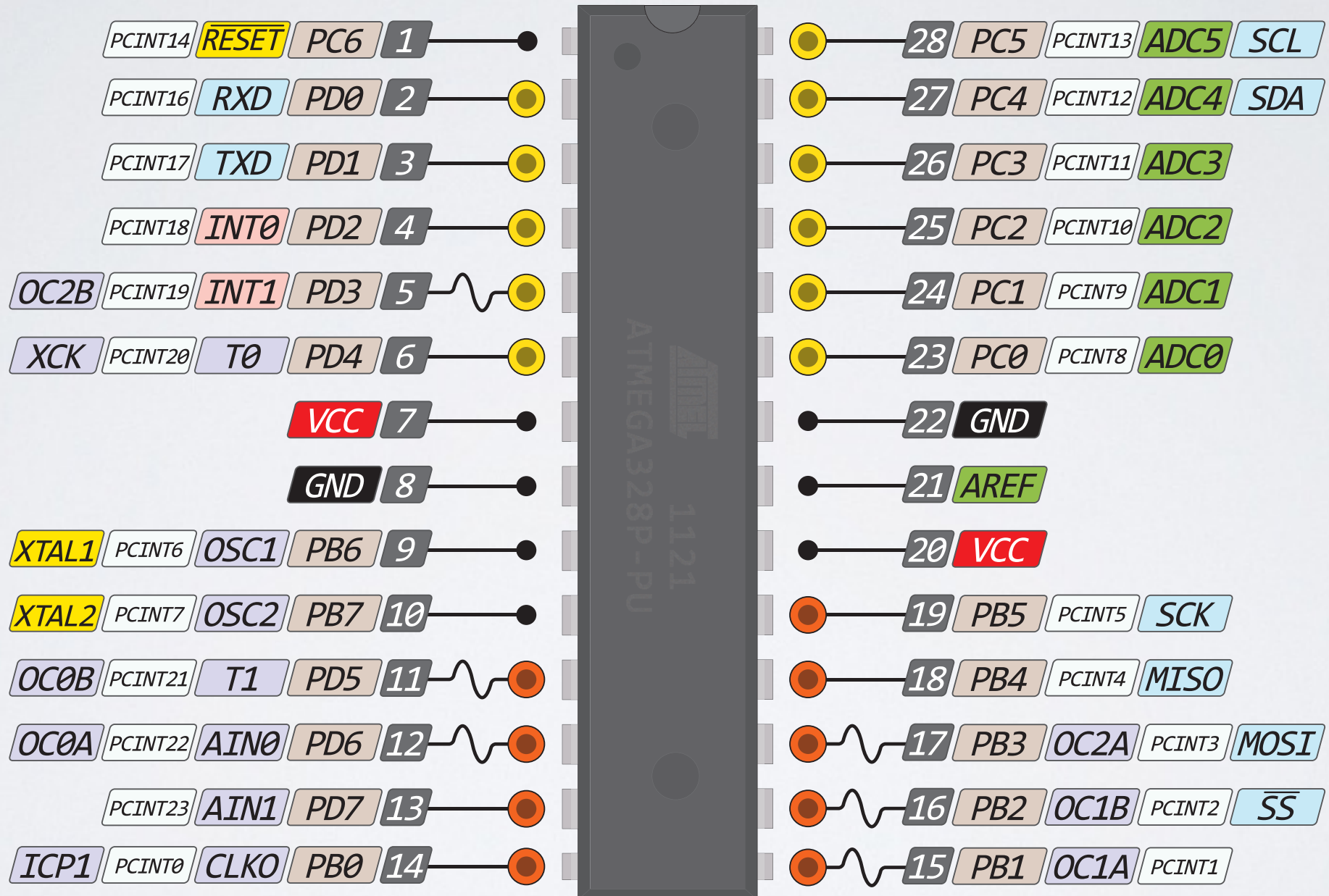
Bit	7	6	5	4	3	2	1	0
0x71	—	—	—	—	—	OCF2B	OCF2A	TOV2
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

ASSR

Bit	7	6	5	4	3	2	1	0
0x71	—	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

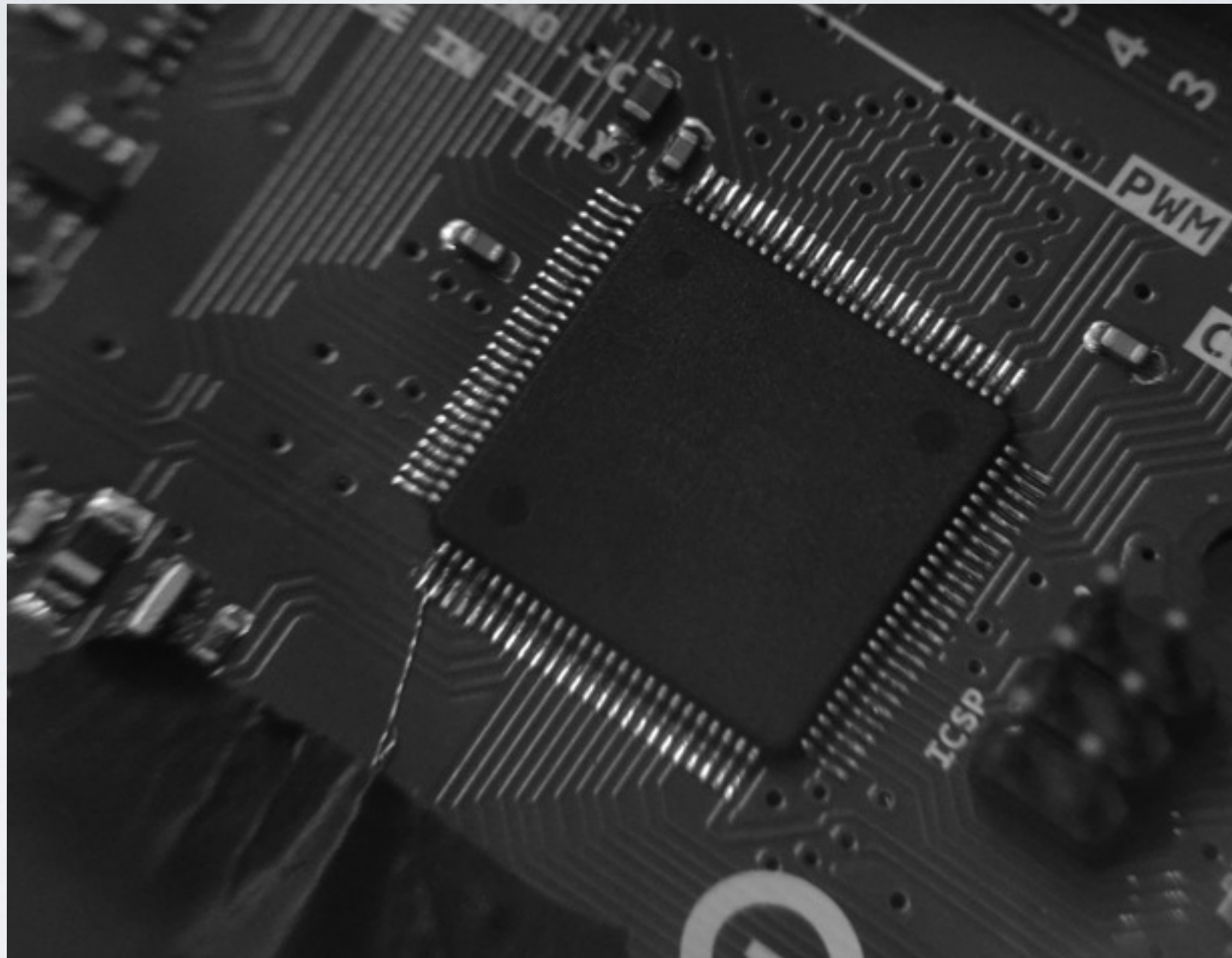
\*ASynchronous Status Register

# PROBLEMAS TIMER 2





# PROBLEMAS TIMER 2



# TIMER 2 INTERRUPTCIONES

```
ISR(TIMER2_OVF_vect){
```

```
}
```

```
ISR(TIMER2_COMPA_vect){
```

```
}
```

```
ISR(TIMER2_COMPB_vect){
```

```
}
```

# TIMER1

TCCR1A

Bit	7	6	5	4	3	2	1	0
0x80	COM1A1	COM1A0	COM1B3	COM1B4	—	—	WGM11	WGM10
R/W?	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

TCCR1B

Bit	7	6	5	4	3	2	1	0
0x81	ICNC1	ICNC2	—	WGM13	WGM12	CS12	CS11	CS10
R/W?	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

TCCR1B

\*COMpare MATCH Output Timer 1A/B Mode Bit

\*Input CAPture NOise CAnceler

\*Input CAPture Edge Select



# TIMER1

## Clock Select Timer 1 Bit

**Table 15-5.** Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{\text{I/O}}/1$ (No prescaling)
0	1	0	$\text{clk}_{\text{I/O}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{I/O}}/64$ (From prescaler)
1	0	0	$\text{clk}_{\text{I/O}}/256$ (From prescaler)
1	0	1	$\text{clk}_{\text{I/O}}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

# TIMER I

TCNT1

Bit	7	6	5	4	3	2	1	0
0xB2	TCNT27	TCNT26	TCNT25	TCNT24	TCNT23	TCNT22	TCNT21	TCNT20
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

OCR2A

Bit	7	6	5	4	3	2	1	0
0xB3	OCR2A7	OCR2A6	OCR2A5	OCR2A4	OCR2A3	OCR2A2	OCR2A1	OCR2A0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

OCR2B

Bit	7	6	5	4	3	2	1	0
0xB4	OCR2B7	OCR2B6	OCR2B5	OCR2B4	OCR2B3	OCR2B2	OCR2B1	OCR2B0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

# TIMER 1

TCNT1H

Bit	7	6	5	4	3	2	1	0
0x85	TCNT115	TCNT114	TCNT113	TCNT112	TCNT111	TCNT110	TCNT19	TCNT18
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

TCNT1L

Bit	7	6	5	4	3	2	1	0
0x84	TCNT17	TCNT16	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

OCR1AH

OCR1BH

ICR1H

y

y

y

OCR1AL

OCR1BL

ICR1H



# TIMER | INTERRUPCIONES

TIMSK1

Bit	7	6	5	4	3	2	1	0
0x70	—	—	ICIE1	—	—	OCIE1B	OCIE1A	TOIE1
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

TIFR1

Bit	7	6	5	4	3	2	1	0
0x71	—	—	ICF1	—	—	OCF1B	OCF1A	TOV1
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

# TIMER | INTERRUPTIONES

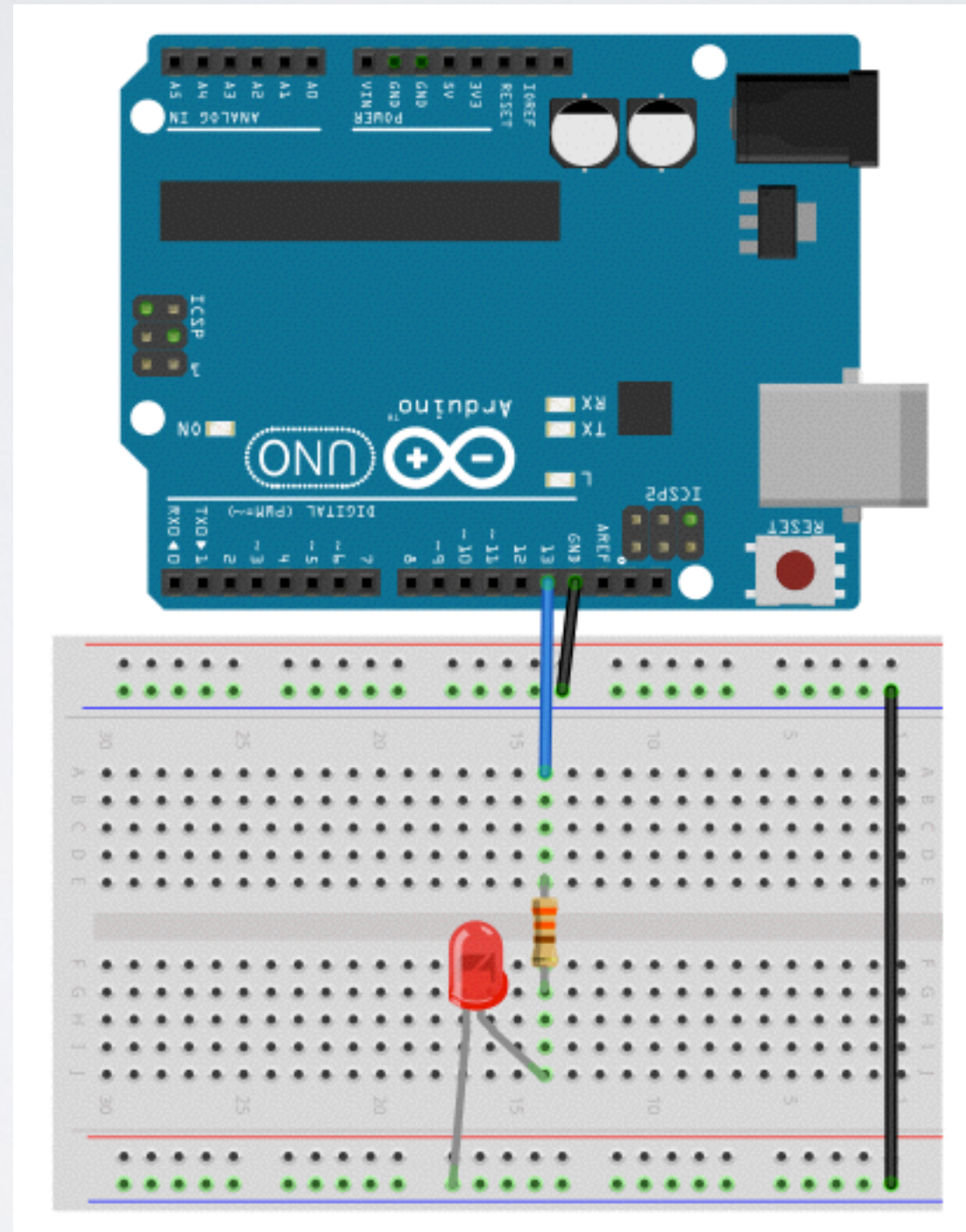
```
ISR(TIMER1_OVF_vect){  
  
}
```

```
ISR(TIMER1_COMPB_vect){  
  
}
```

```
ISR(TIMER1_COMPA_vect){  
  
}
```

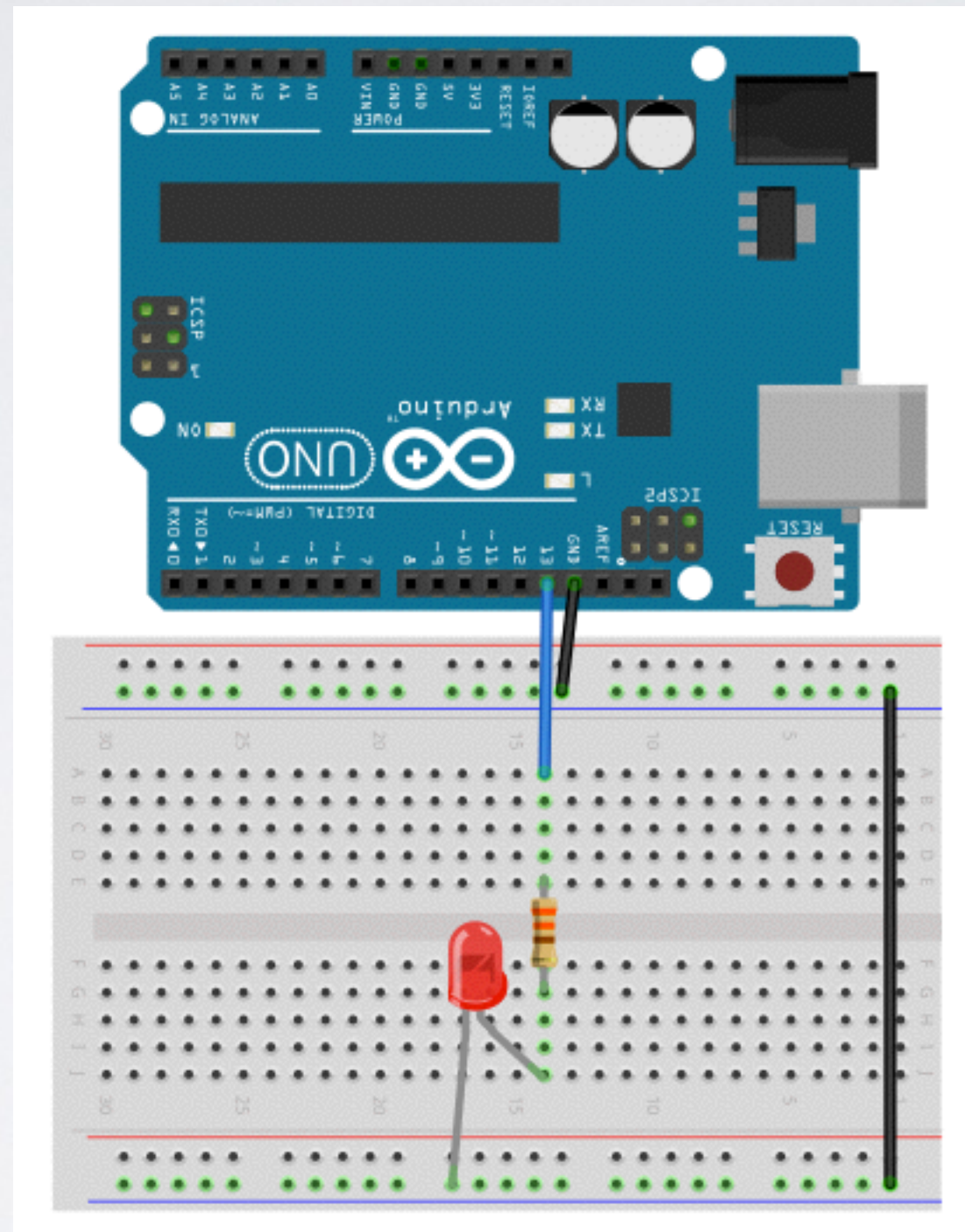
```
ISR(TIMER1_CAPT_vect){  
  
}
```

# TIMER 1 OVF





# TIMER | COMPA



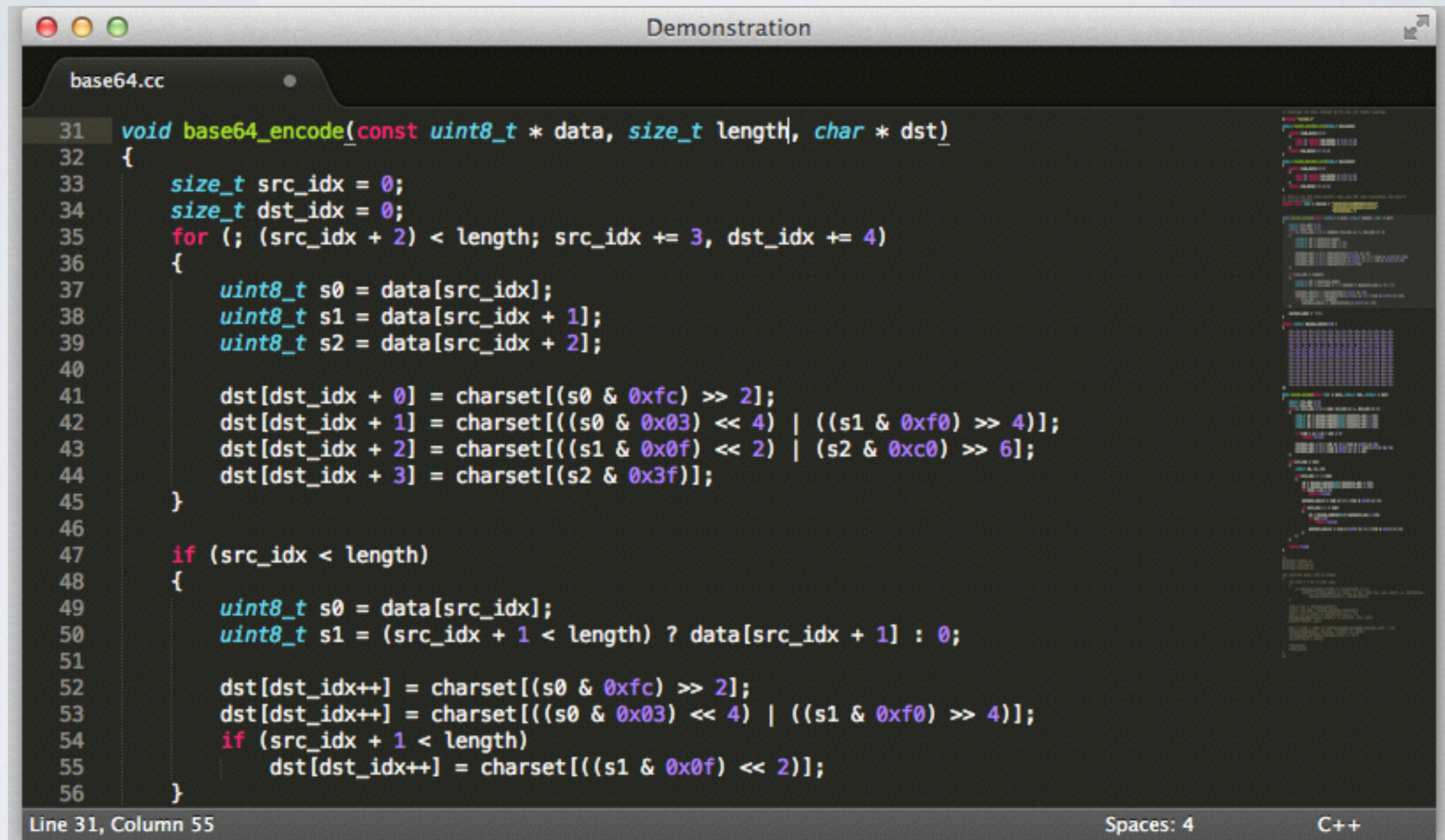
# SUBLIME TEXT

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia.

Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto, se puede obtener una licencia para su uso ilimitado, pero el no disponer de ésta no genera ninguna limitación más allá de una alerta cada cierto tiempo. Sitio oficial : <http://www.sublimetext.com>.

- **Minimapa**
- **Multi Selección**
- **Multi Cursor**
- **Multi Layout**
- **Soporte nativo para infinidad de lenguajes**
- **Búsqueda Dinámica**
- **Auto completado y marcado de llaves**
- **Soporte de Snippets y Plugins**
- **Configuración total de Keybindings**
- **Acceso rápido a línea o archivo (Cmd+P o Ctrl+P)**
- **Coloreado y envoltura de sintaxis**
- **Pestañas**
- **Resaltado de paréntesis e indentación.**

# SUBLIME TEXT



```
31 void base64_encode(const uint8_t * data, size_t length, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < length)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < length)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
```

Line 31, Column 55

Spaces: 4 C++



Comando (⌘/Ctrl)	Acción
Ctrl + L Ctrl + ⌥ + L Ctrl + ⌥ + K Ctrl + KK (igual en Mac) Ctrl + K + ⌘ (igual en Mac) Ctrl + ⌥ + D	Manejo de lienas
Ctrl + X	Corta la linea completa
Ctrl + ↵ Ctrl + ⌥ + ↵	Añadir lineas
Ctrl + ↑ Ctrl + ↓ Ctrl + ⌥ + ↑ Ctrl + ⌥ + ↓	Selección completa
Ctrl + D Ctrl + Click	Multicursor con selección
Ctrl + M Ctrl + ⌥ + M (igual en Mac)	Manejode contenido con corchetes

Comando (⌘/Ctrl)	Acción
Ctrl + P Ctrl + ⌘ + P	Abrir archivos y panel de control
Ctrl + R	Buscador de funciones
Ctrl + F2 F2 ⌘ + F2 Ctrl + ⌘ + F2	Bookmarks
Ctrl + KU Ctrl + KL	Mayúsculas/Minúsculas
Ctrl + / Ctrl + ⌘ + /	Comentarios
Ctrl + Space	Navegación en auto-compleat

Creado por: Miguel Angel Ruiz Gálvez  
Contacto: [miguelo.me](http://miguelo.me)

Agradecimientos a:

- Massimo Banzi
- Hernando Barragan
- David Cuartielles
- Pighixxx

Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita <http://creativecommons.org/licenses/by-nc-sa/4.0/>.