ARDUINO AVANZADO

Miguel Angel Ruiz Gálvez
Visita: miguelo.me
Material en: http://goo.gl/UO3xix

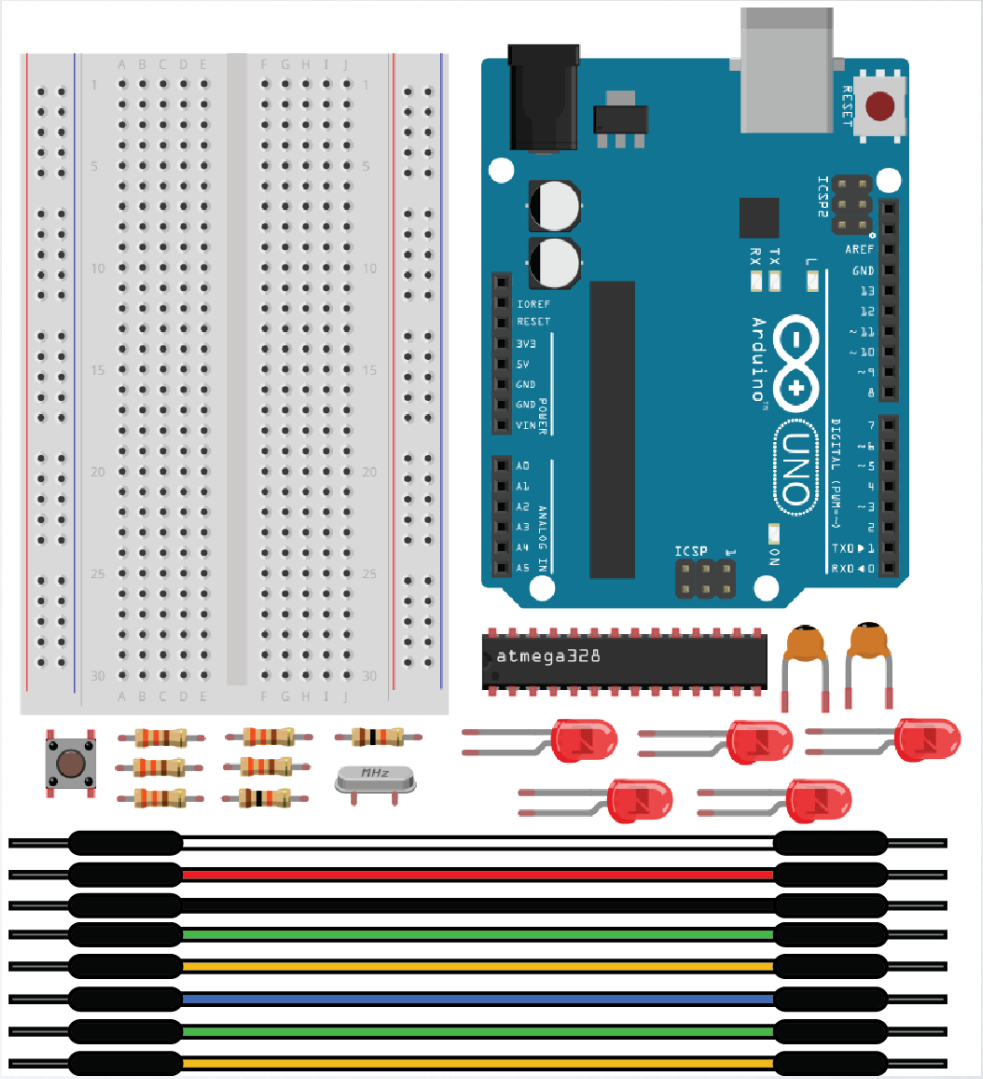
SOMEFI

MIGUELRUIZ
MECHATRONIC ENGINEER

CC BY NC SA

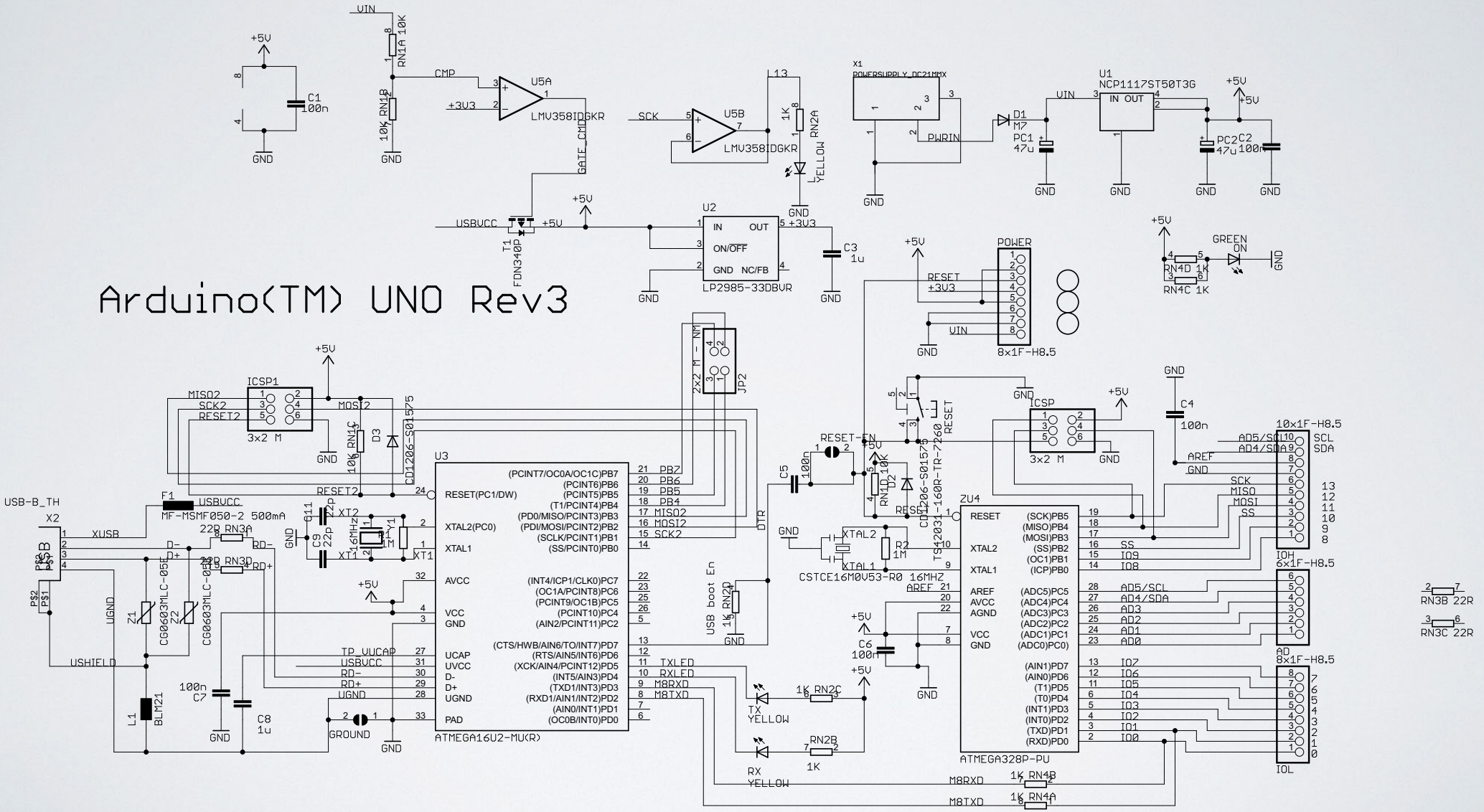
MATERIAL A UTILIZAR

- Tarjeta de desarrollo Arduino UNO y cable.
- 1 Protoboard
- 1 Micorocontrolador ATmega 328-UP
- 1 Cristal oscilador de 16 MHz
- 5 LED´s
- 5 Resistencias de 300 Ω, ¼ [W]
- 1 Pushbutton
- 2 Resistencias de 10 KΩ, ¼ [W]
- 2 Capacitores de 22pf
- 20 Jumpers



MIGUELRUIZ
MECHATRONIC ENGINEER

ARDUINO



MIGUEL RUIZ
MECHATRONIC ENGINEER

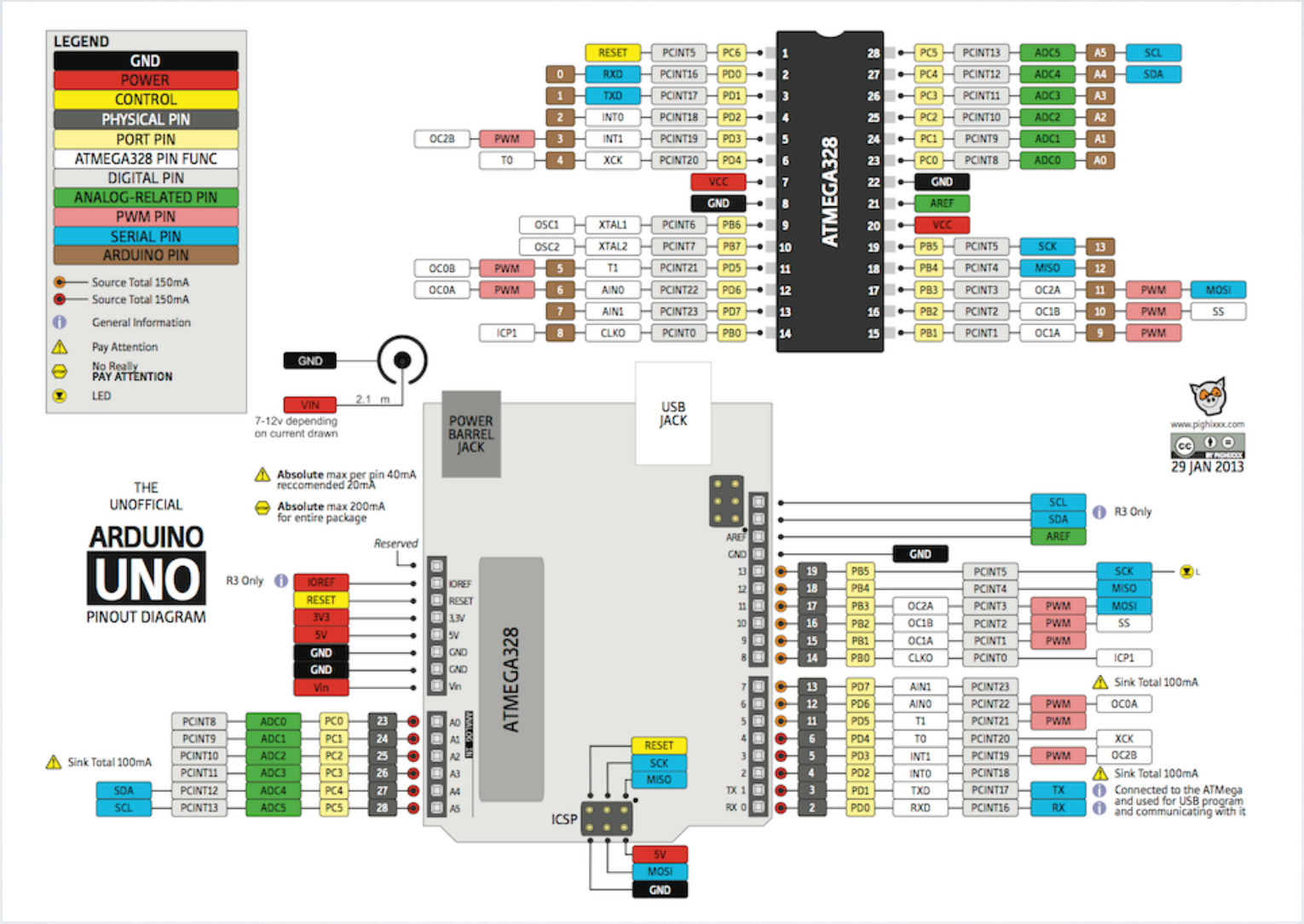
ATMEGA 328-UP

High Performance, Low Power AVR® 8-Bit Microcontroller.

- Voltaje de Operacion: 1.8 - 5.5V / 0-20MHz
- 32K Bytes en Flash/1K Byte 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 presescalador independiente.
- 6 canales de PWM.
- 6-canales10-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^2C.
- Watchdog Timer con Oscilador independiente. Interrupción y Wake-up con cambio de flanco.

MIGUELRUIZ
MECHATRONIC ENGINEER

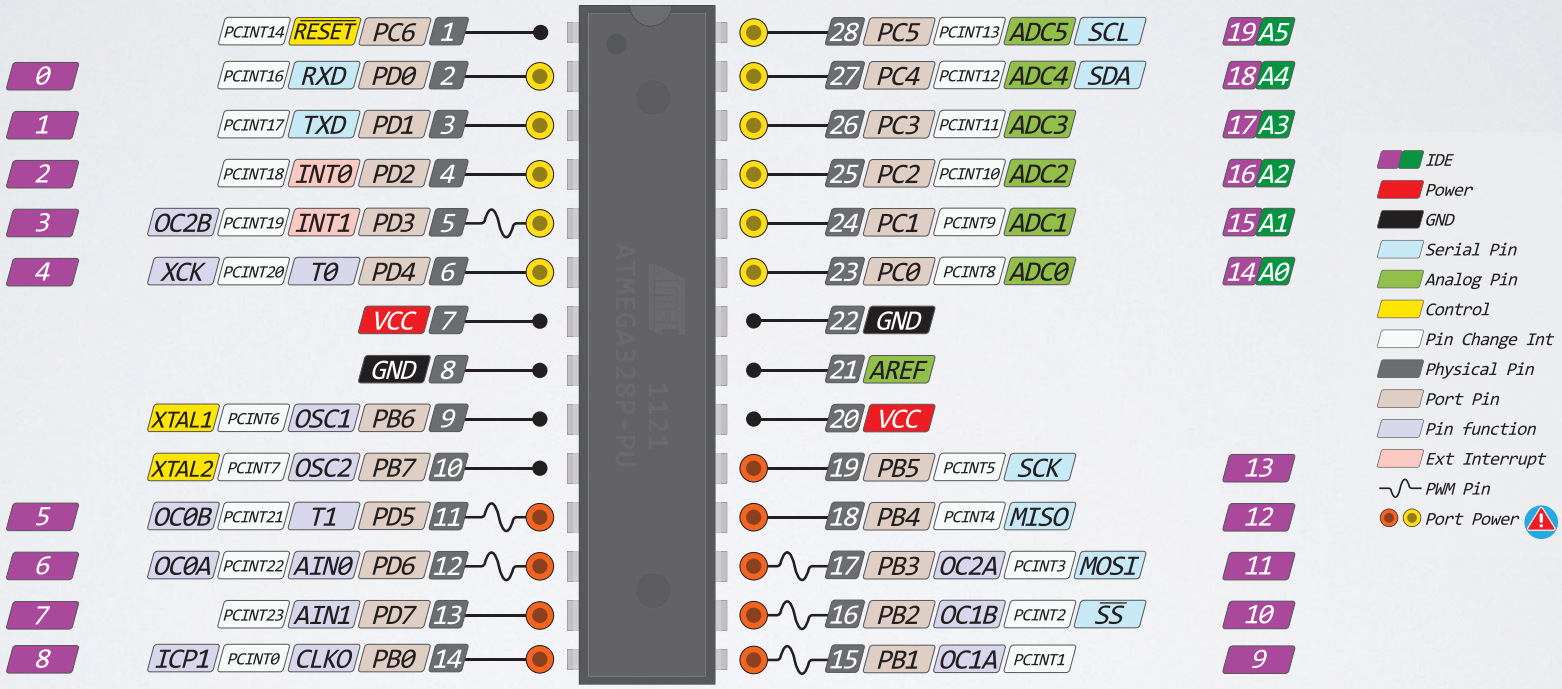
ATMEGA 328-UP



MIGUELRUIZ
MECHATRONIC ENGINEER

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

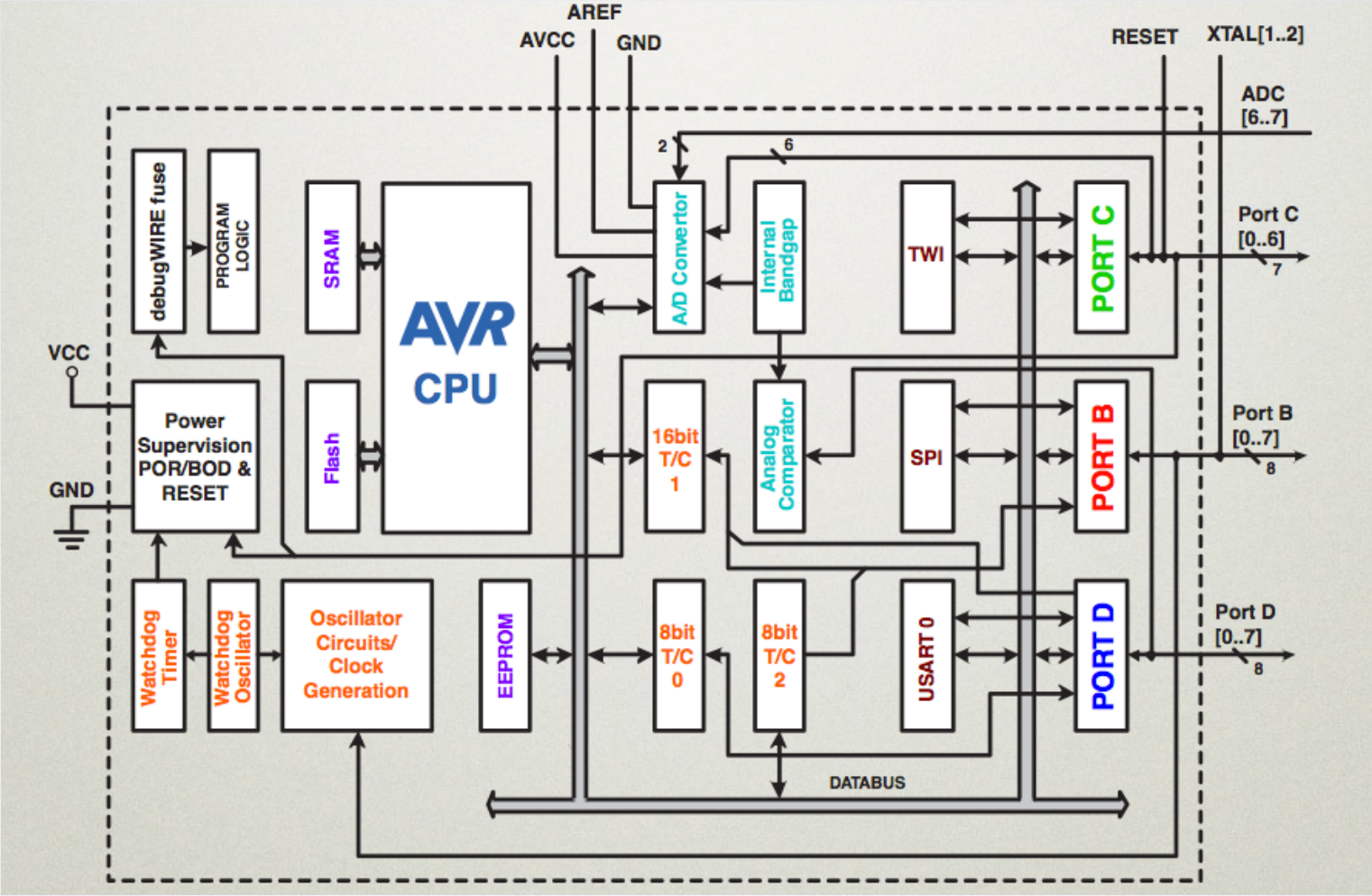
www.pighixxx.com

16 JUL 2014

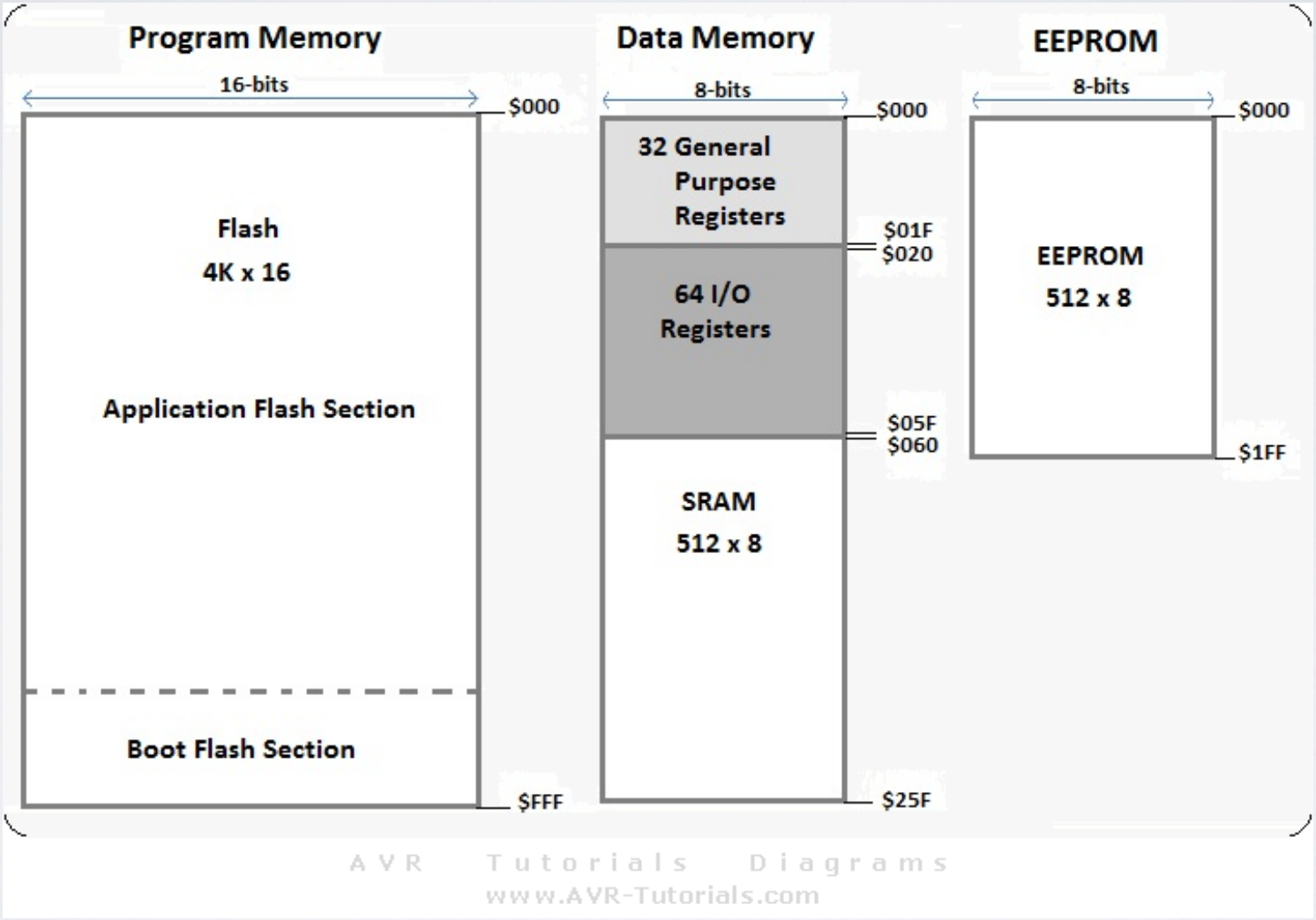
ver 3 rev 0

MIGUELRUIZ
MECHATRONIC ENGINEER

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

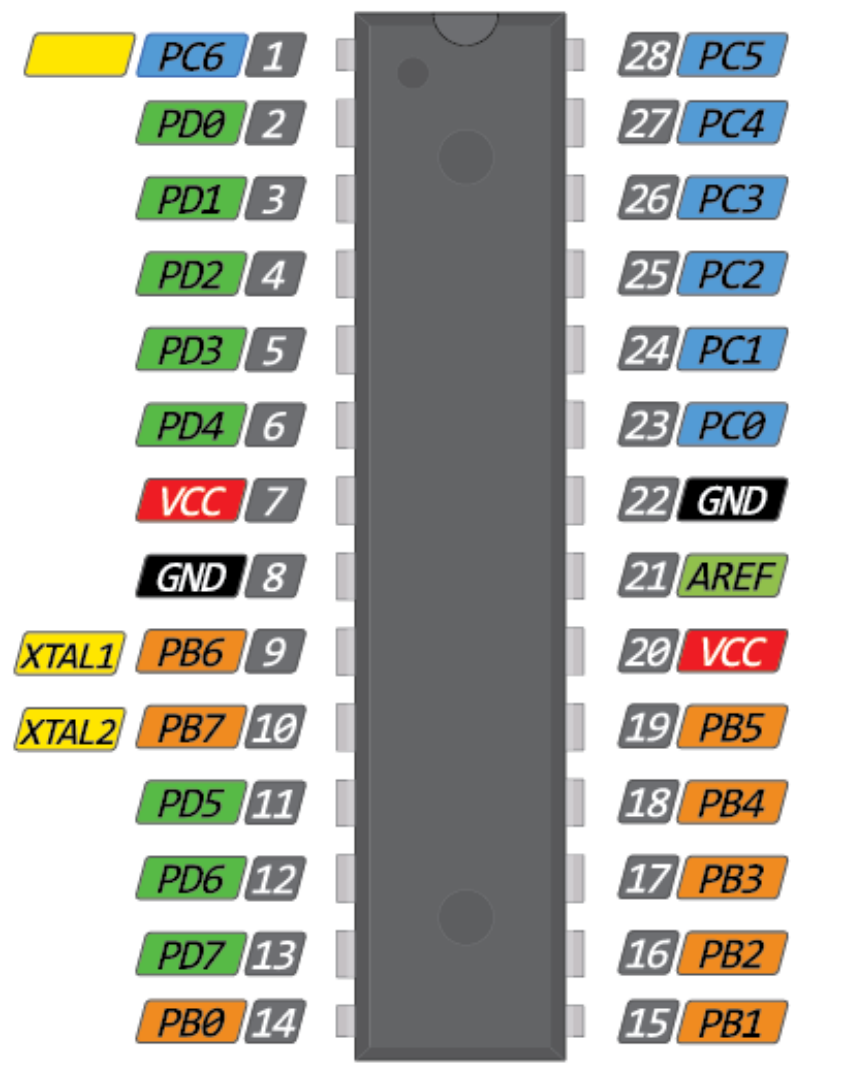
C/C++

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

2.6 MHz

MIGUELRUIZ
MECHATRONIC ENGINEER

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: 1/2 de máx.
Resistencias Pullup: 20kΩ
Estado de alta Impedancia (Input): 100MΩ

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.

PUERTOB

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
Defaul	0	0	0	0	0	0	0	0

0:HIGH
1:LOW

DDRB

Bit	7	6	5	4	3	2	I	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	I	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

MIGUELRUIZ
MECHATRONIC ENGINEER

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
Defaul	0	0	0	0	0	0	0	0

0:HIGH
1:LOW

DDRC

Bit	7	6	5	4	3	2	I	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	I	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

MIGUELRUIZ
MECHATRONIC ENGINEER

PUERTO D

PORD

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

0:HIGH
1:LOW

DDRD

Bit	7	6	5	4	3	2	I	0
0x2A	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
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	I	0
0x29	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
R/W?	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

0:HIGH
1:LOW

MIGUELRUIZ
MECHATRONIC ENGINEER

CONFIGURACIÓN

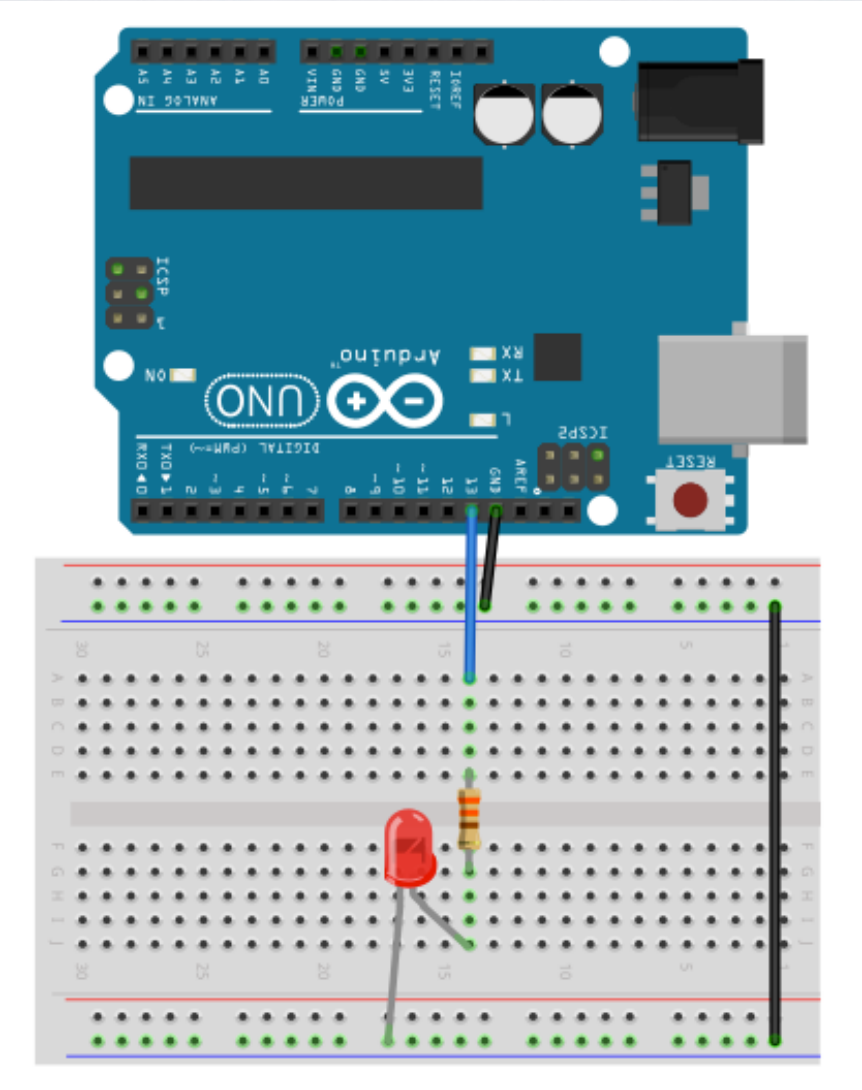
Acción	DDR	PORT	PUD	I/O	Pull-Up
Tri-estado (Hi-Z)	0	x	X	Input	No
Tri-estado (Hi-Z) con pull-up	0	1	0	Input	Si
LOW	1	0	x	Output	No
HIGH	1	1	x	Output	No

MCUCR

Bit	7	6	5	4	3	2	1	0
0x35	-	BODS	BODSE	PUD	-	-	IVSEL	IVCE
R/W?	R	R/W	R/W	R/W	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

*Pull Up Desable

BLINK EN C



MIGUELRUIZ
MECHATRONIC ENGINEER

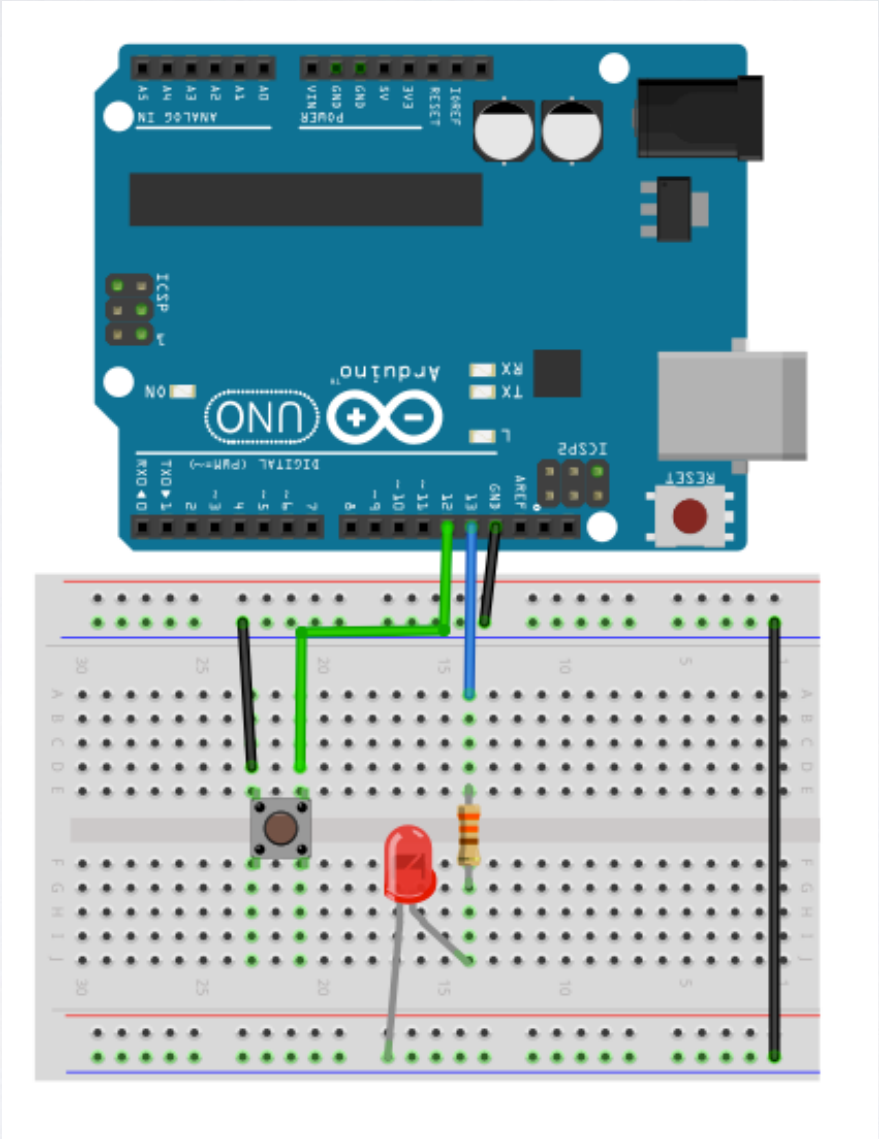
MASCARAS

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

```
byte registro = 0b11110000;
```

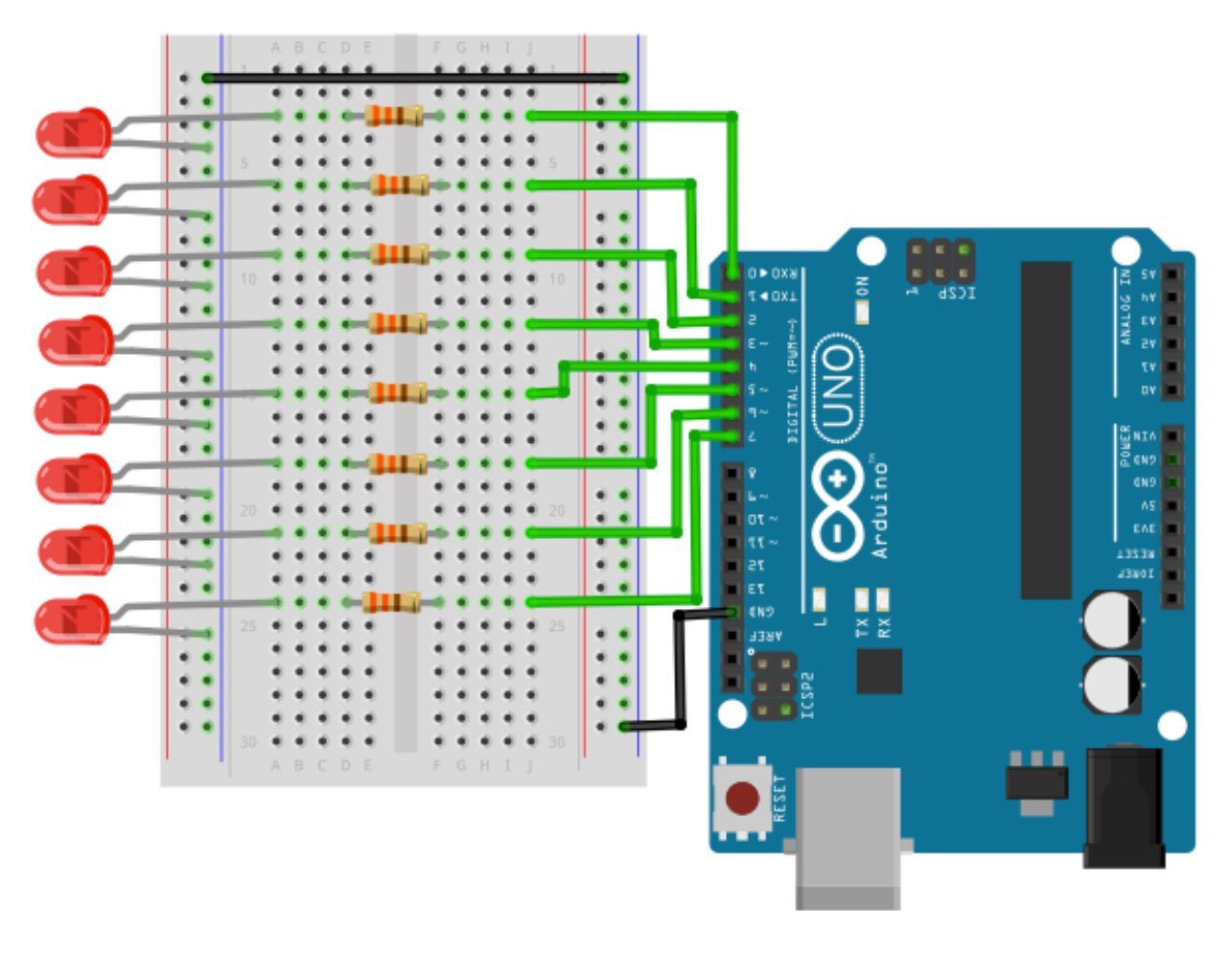
Simbolo	Nombre	Ejemplo
|=	**OR**	`registro |= 0b00001010;`
&=	**AND**	`registro &= 0b01011111;`
(1 << casilla)	**Asignación de byte**	`registro |= (1 << 3);` `registro &= (1 << 5);`

BOTTON



MIGUELRUIZ
MECHATRONIC ENGINEER

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 256) , mientras que el Timer1 tiene un contador de16 bits. (0 a 65 536).
- El uC Atmega2560 (Arduino Mega) cuneta 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	`delay(), millis() y micros()`
Timer 1	`Servo(), analogWrite()`
Timer 2	`tone(), analogWrite()`

TIMERS

- Timer/Counter Control Register A: TCCR*n*A
- Timer/Counter Control Register B: TCCR*n*B
- Timer CouNT: TCNT*n*
- Output Compare Register A: OCR*n*A
- Output Compare Register B: OCR*n*B
- Timer/Counter Interrupt Mask Register: TIMSKn

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
Defaul	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
Defaul	0	0	0	0	0	0	0	0

*COmpare Match Output Timer 0 A/B Mode Bit

*Force Output Compare Timer 0 A/B

MIGUELRUIZ
MECHATRONIC ENGINEER

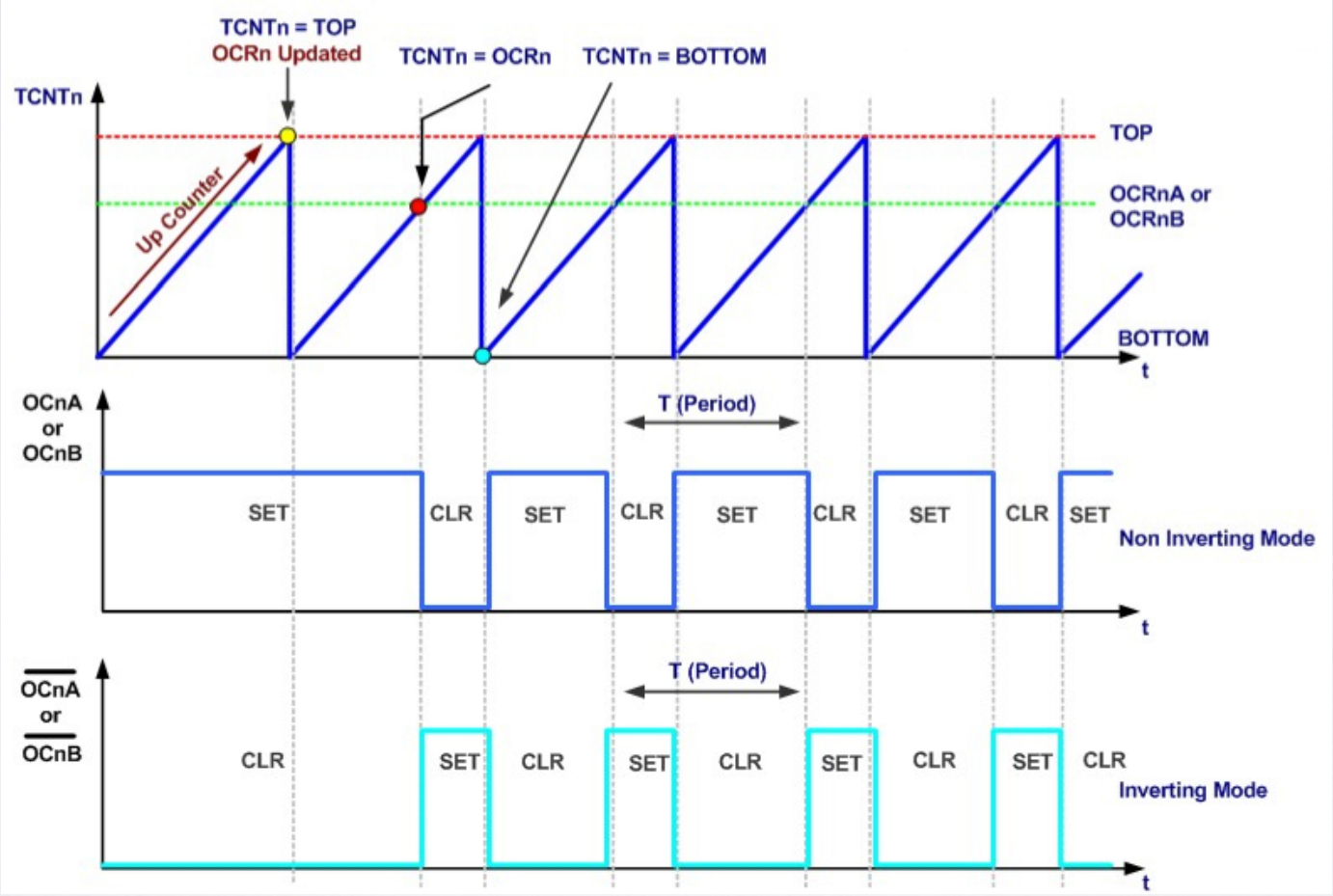
TIMER 0

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(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

FAST PWM

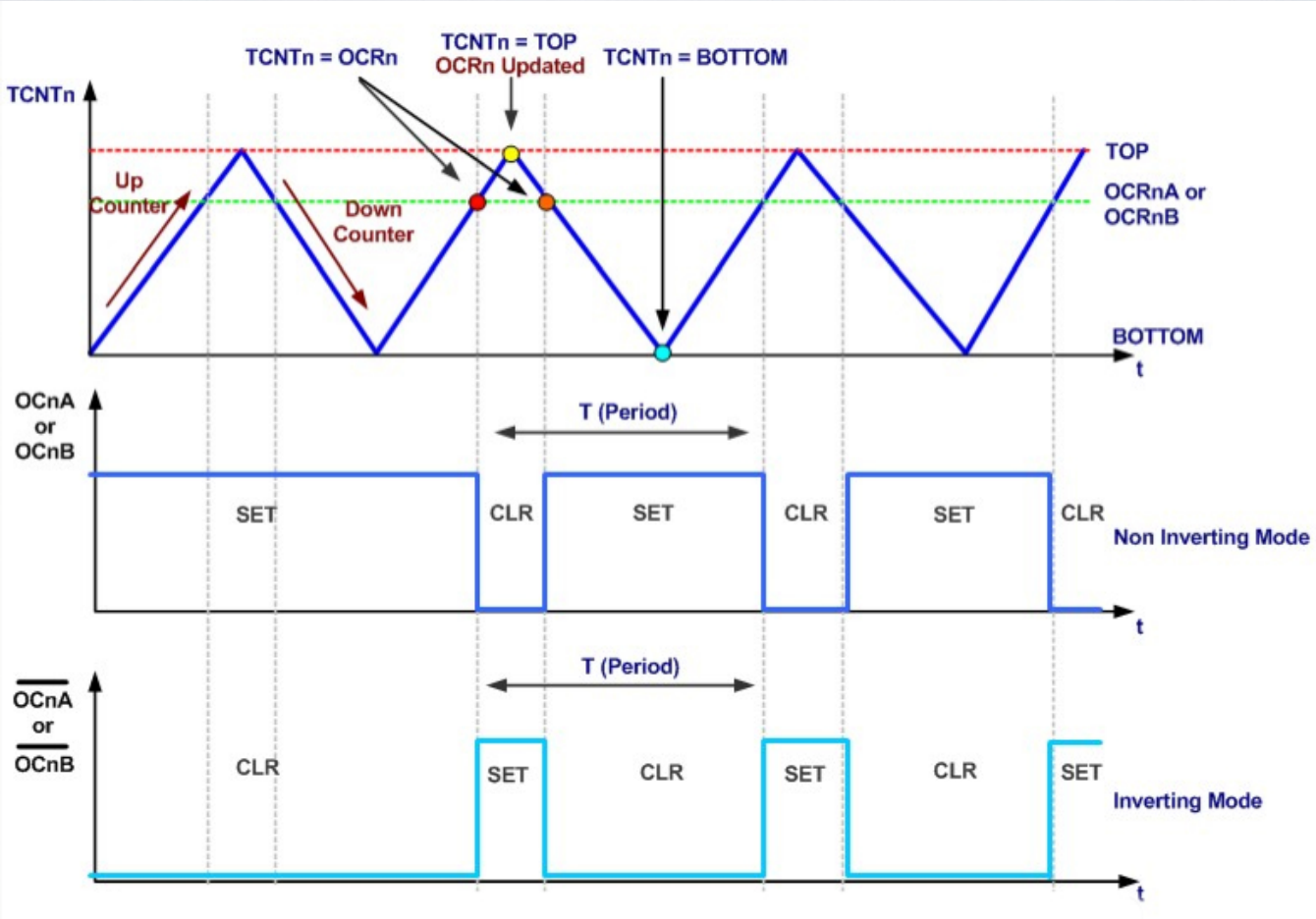


Frecuencia

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

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	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/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

Tdeseado= 1 / 30 KHz = 0.033 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
Defaul	0	0	0	0	0	0	0	0

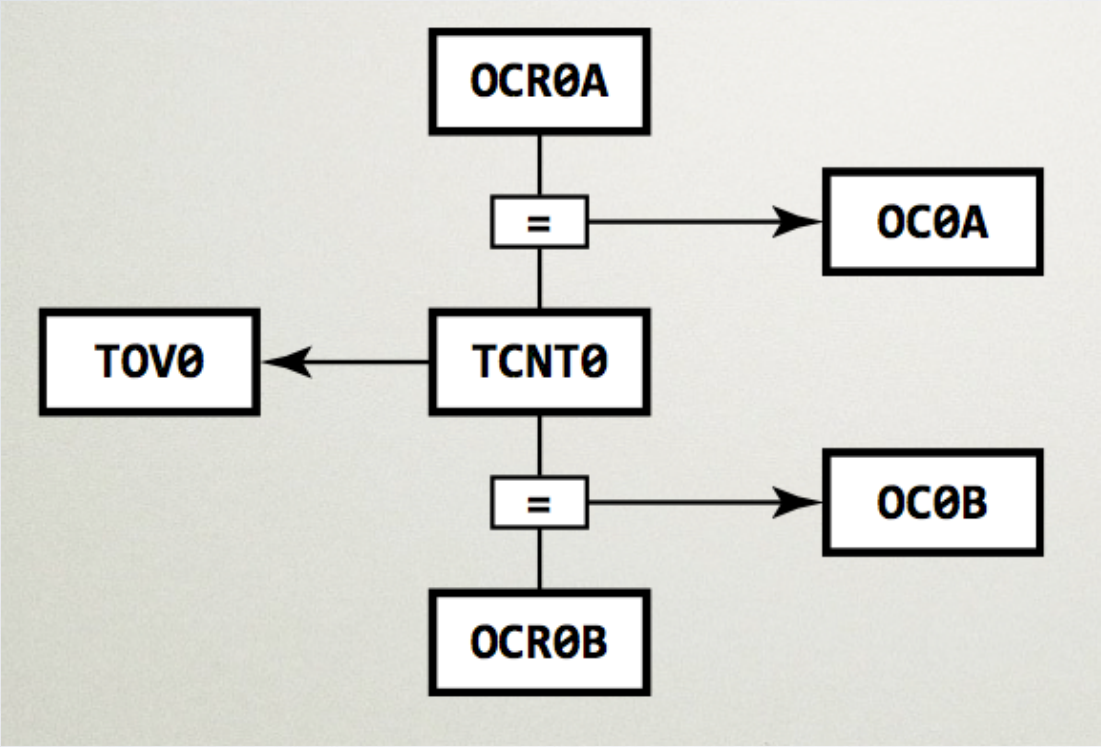
OCR0A

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
Defaul	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
Defaul	0	0	0	0	0	0	0	0

TIMER 0 INTERRUPCIONES



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

TIMER 0 INTERRUPCIONES

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
Defaul	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
Defaul	0	0	0	0	0	0	0	0

MIGUELRUIZ
MECHATRONIC ENGINEER

TIMER 0 INTERRUPCIONES

```
ISR(TIMER0_OVF_vect){

}

ISR(TIMER0_COMPA_vect){

}

ISR(TIMER0_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
Defaul	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
Defaul	0	0	0	0	0	0	0	0

***COmpare Match Output Timer 2 A/B Mode Bit**

***Force Output Compare 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	clk_{T2S}/(No prescaling)
0	1	0	clk_{T2S}/8 (From prescaler)
0	1	1	clk_{T2S}/32 (From prescaler)
1	0	0	clk_{T2S}/64 (From prescaler)
1	0	1	clk_{T2S}/128 (From prescaler)
1	1	0	clk_{T2S}/256 (From prescaler)
1	1	1	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
Defaul	0	0	0	0	0	0	0	0

OCR2A

Bit	7	6	5	4	3	2	1	0
0xB3	OCR2A7	OCR2A6	OCR0A5	OCR2A4	OCR2A3	OCR2A2	OCR2A1	OCR2A0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Defaul	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
Defaul	0	0	0	0	0	0	0	0

MIGUELRUIZ
MECHATRONIC ENGINEER

TIMER 2 INTERRUPCIONES

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
Defaul	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
Defaul	0	0	0	0	0	0	0	0

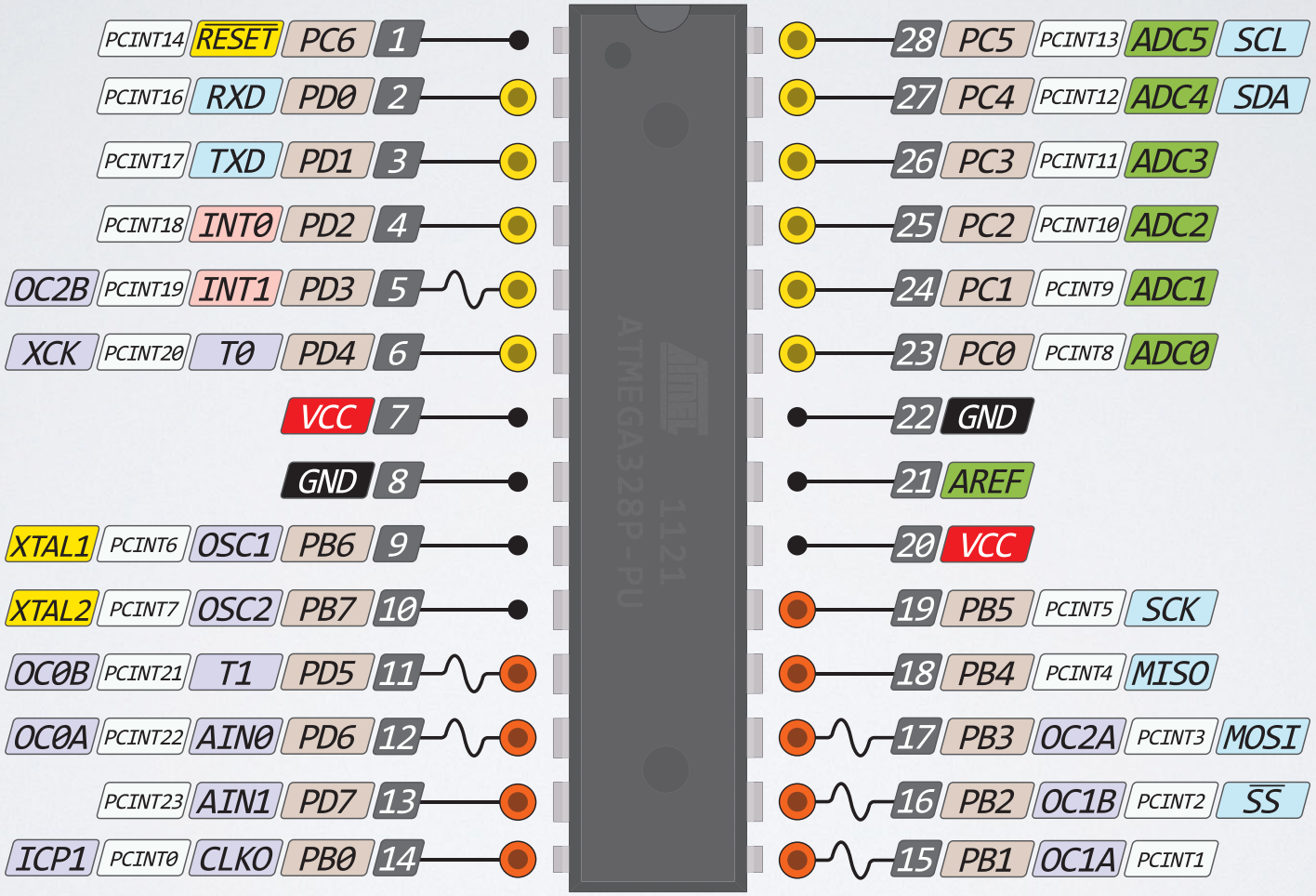
ASSR

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
Defaul	0	0	0	0	0	0	0	0

*ASynchronous Status Register

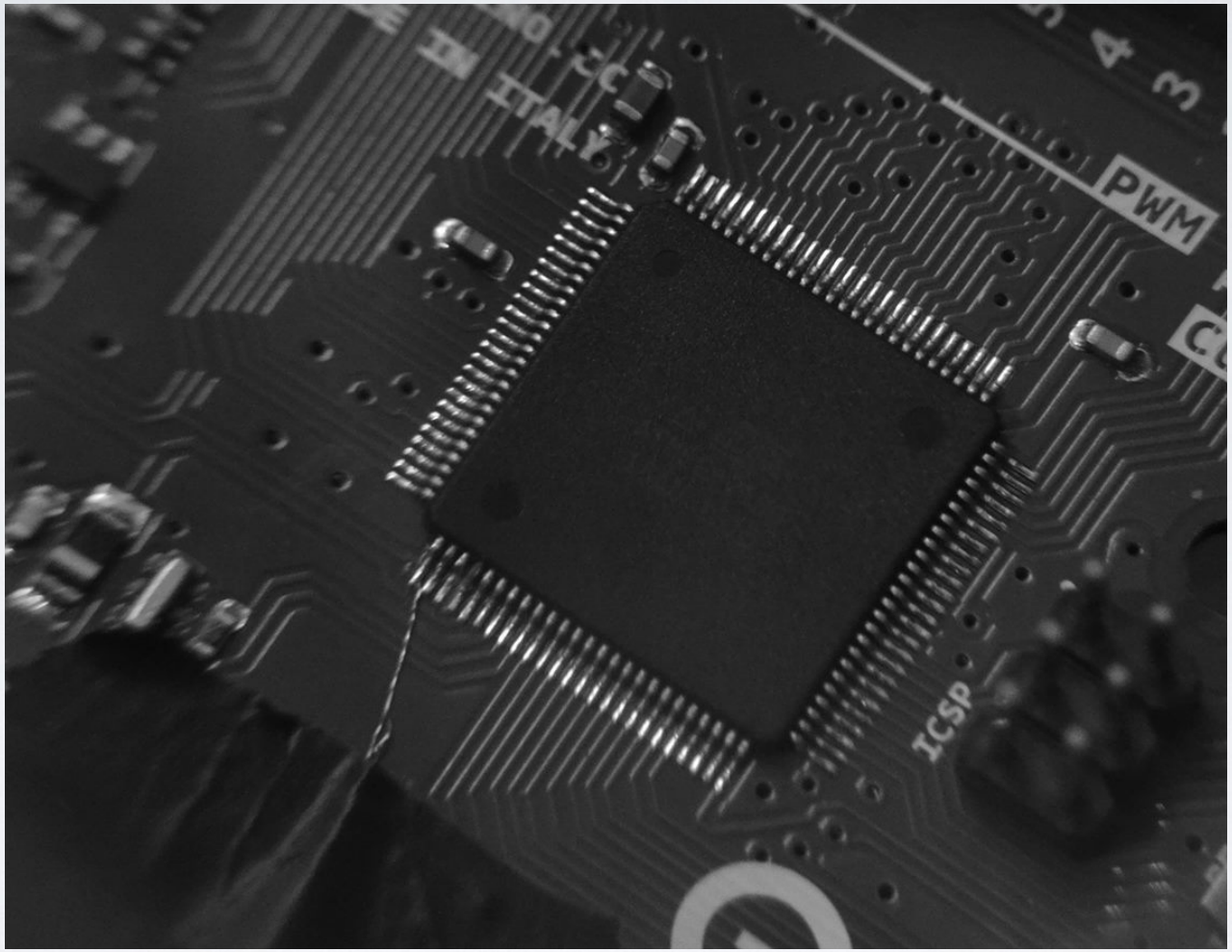
MIGUELRUIZ
MECHATRONIC ENGINEER

PROBLEMAS TIMER 2



MIGUELRUIZ
MECHATRONIC ENGINEER

PROBLEMAS TIMER 2



MIGUELRUIZ
MECHATRONIC ENGINEER

TIMER 2 INTERRUPCIONES

```
ISR(TIMER2_OVF_vect){

}

ISR(TIMER2_COMPA_vect){

}

ISR(TIMER2_COMPB_vect){

}
```

TIMER 1

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
Defaul	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
Defaul	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

TIMER I

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	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/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 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
Defaul	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
Defaul	0	0	0	0	0	0	0	0

OCR1AH y OCR1AL

OCR1BH y OCR1BL

ICR1H y ICR1H

TIMER I 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
Defaul	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
Defaul	0	0	0	0	0	0	0	0

TIMER I INTERRUPCIONES

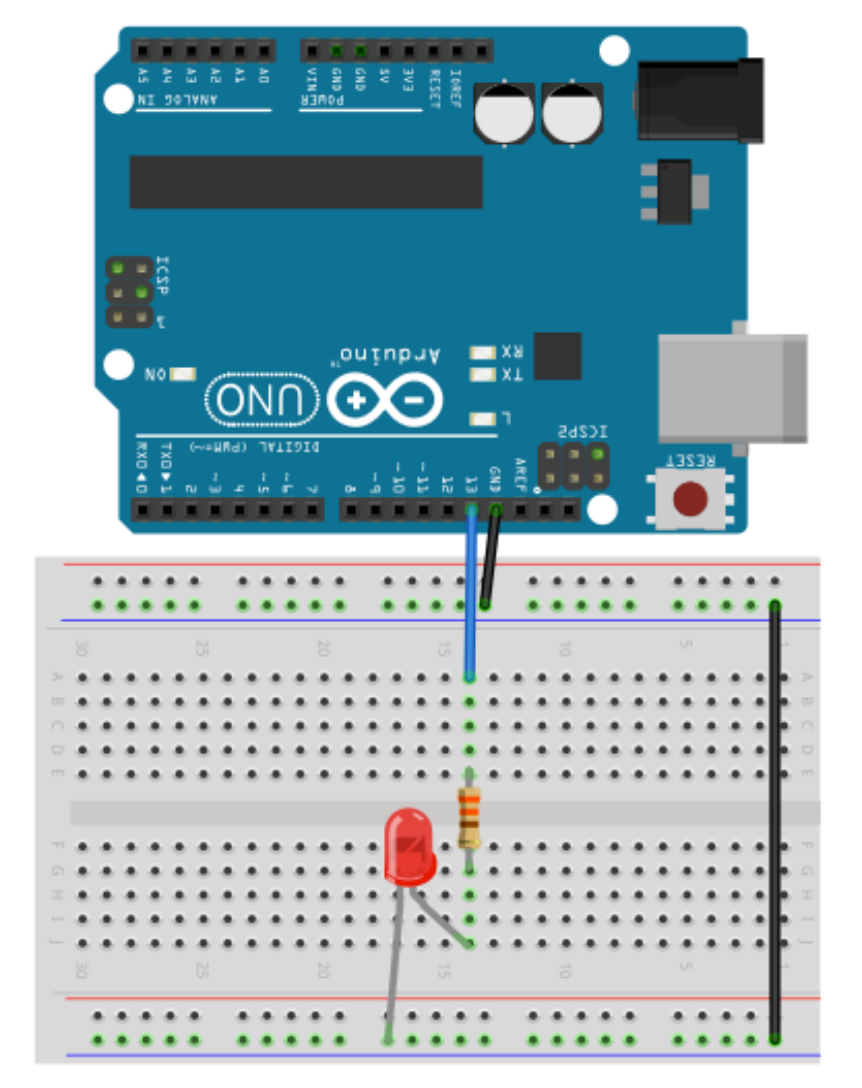
```
ISR(TIMER1_OVF_vect){

}
```

```
ISR(TIMER1_COMPB_vect){

}
```

```
ISR(TIMER1_COMPA_vect){

}
```

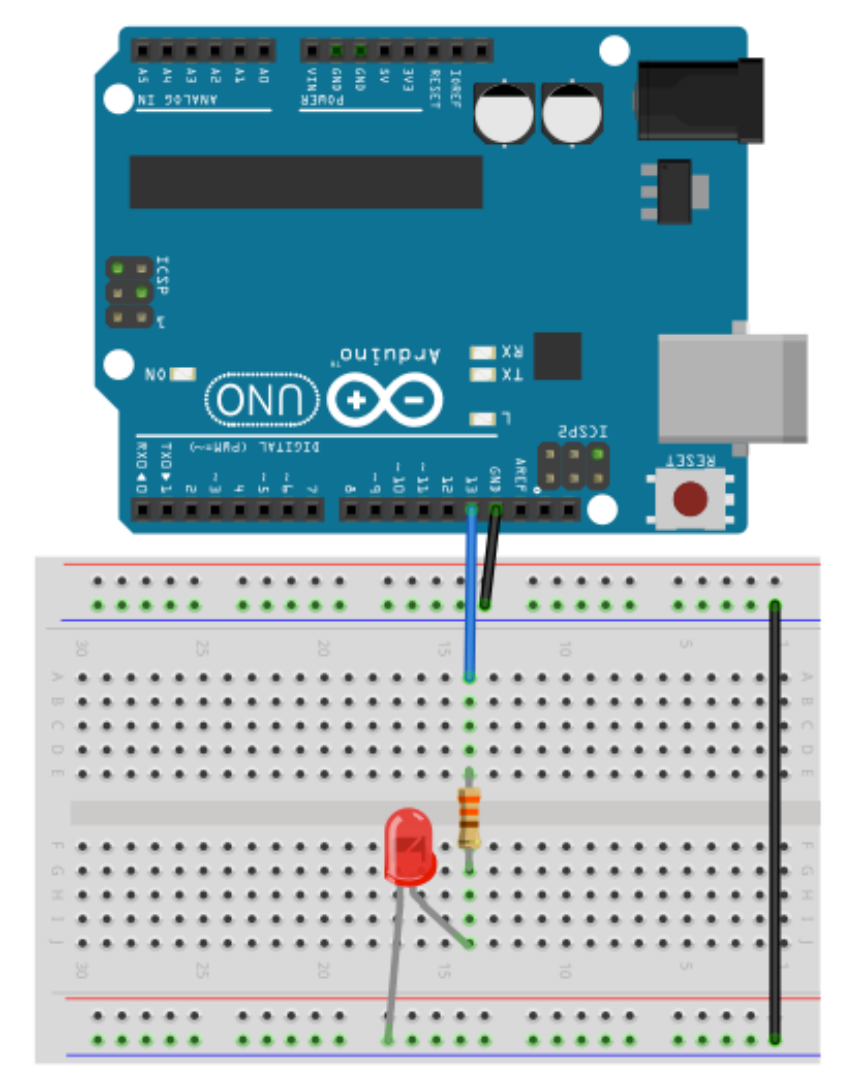
```
ISR(TIMER1_CAPT_vect){

}
```

TIMER 1 OVF



MIGUELRUIZ
MECHATRONIC ENGINEER

TIMER I COMPA



MIGUELRUIZ
MECHATRONIC ENGINEER

SLEEP MODE

El Sleep Mode es de utilidad para reducir el consumo de energía del dispositivo, esto puede ser muy útil y/o necesario cuando se energía el microcntrolador a través de una batería y / o panel solar.

El ATmega328-UP consume aproximadamente 20mA (30mA ARDUINO UNO) durante su funcionamiento normal mientras que en el modo POWER_DOWN puede llegara consumir 0.05mA (15mA ARDUINO UNO). Eso significa que una batería de 9V de 565 mAh puede nos puede dar 11300 horas en el modo de consumo mas bajo.

Existen 5 modos de ahorro :

- SLEEP_MODE_IDLE - mayor consumo
- SLEEP_MODE_ADC
- SLEEP_MODE_PWR_SAVE
- SLEEP_MODE_STANDBY
- SLEEP_MODE_PWR_DOWN - menor consumo

SLEEP MODE

Table 9-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes.

	Active Clock Domains					Oscillators		Wake-up Sources							
Sleep Mode	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT1, INT0 and Pin Change	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O	Software BOD Disable
Idle			X	X	X	X	X(2)	X	X	X	X	X	X	X	
ADC Noise Reduction				X	X	X	X(2)	X(3)	X	X(2)	X	X	X		
Power-down								X(3)	X				X		X
Power-save					X		X(2)	X(3)	X	X			X		X
Standby(1)						X		X(3)	X				X		X
Extended Standby					X(2)	X	X(2)	X(3)	X	X			X		X

Notes:
1. Only recommended with external crystal or resonator selected as clock source.
2. If Timer/Counter2 is running in asynchronous mode.
3. For INT1 and INT0, only level interrupt.

SLEEP MODE

MCUCR

Bit	7	6	5	4	3	2	1	0
0x35	-	BODS	BODSE	PUD	-	-	IVSEL	IVCE
R/W?	R	R/W	R/W	R/W	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

SMCR

Bit	7	6	5	4	3	2	1	0
0x33	-	-	-	-	SM2	SM1	SM0	SE
R/W?	R	R	R	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

PRR

Bit	7	6	5	4	3	2	1	0
0x64	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUSART0	PRADC
R/W?	R	R/W	R/W	R/W	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

MIGUELRUIZ
MECHATRONIC ENGINEER

SLEEP MODE

Sleep Mode Select Bit

Table 9-2. Sleep Mode Select

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby(1)
1	1	1	External Standby(1)

Note: 1. Standby mode is only recommended for use with external crystals or resonators.

SLEEP MODE EN C

```
#include <avr/sleep.h>
```

```
set_sleep_mode(mode):
```

- SLEEP_MODE_IDLE
- SLEEP_MODE_ADC
- SLEEP_MODE_PWR_SAVE
- SLEEP_MODE_STANDBY
- SLEEP_MODE_PWR_DOWN

```
sleep_enable();
sleep_mode();

sleep_disable();
```

- La función set nos permite elegir alguno de los 5 Sleep Modes.
- La función enable habilita el modo Sleep seleccionado.
- La función mode inicia el modo Sleep, mientras este activa esta función las funciones de microcntrolador estarán in activas.
- Finalmente disable habilita nuevamente el microcontrolador continuando la ejecución después de esta instrucción.

NOTA : Es muy importante activar las interrupciones de WAKE UP antes de habilitar el modo Sleep

SLEEP MODE EN C

```
#include <avr/sleep.h>

void enterSleep(void){
    set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    sleep_enable();
    sleep_mode();

    // se queda aqui

    sleep_disable();
}
```

POWER REDUCTION EN C

```
#include <avr/power.h>

power_adc_disable();
power_spi_disable();
power_timer0_disable();
power_timer1_disable();
power_timer2_disable();
power_twi_disable();
power_all_enable();
```

- Esta librería se utiliza para controlara el registro **PPR** (Power Reduction Register)

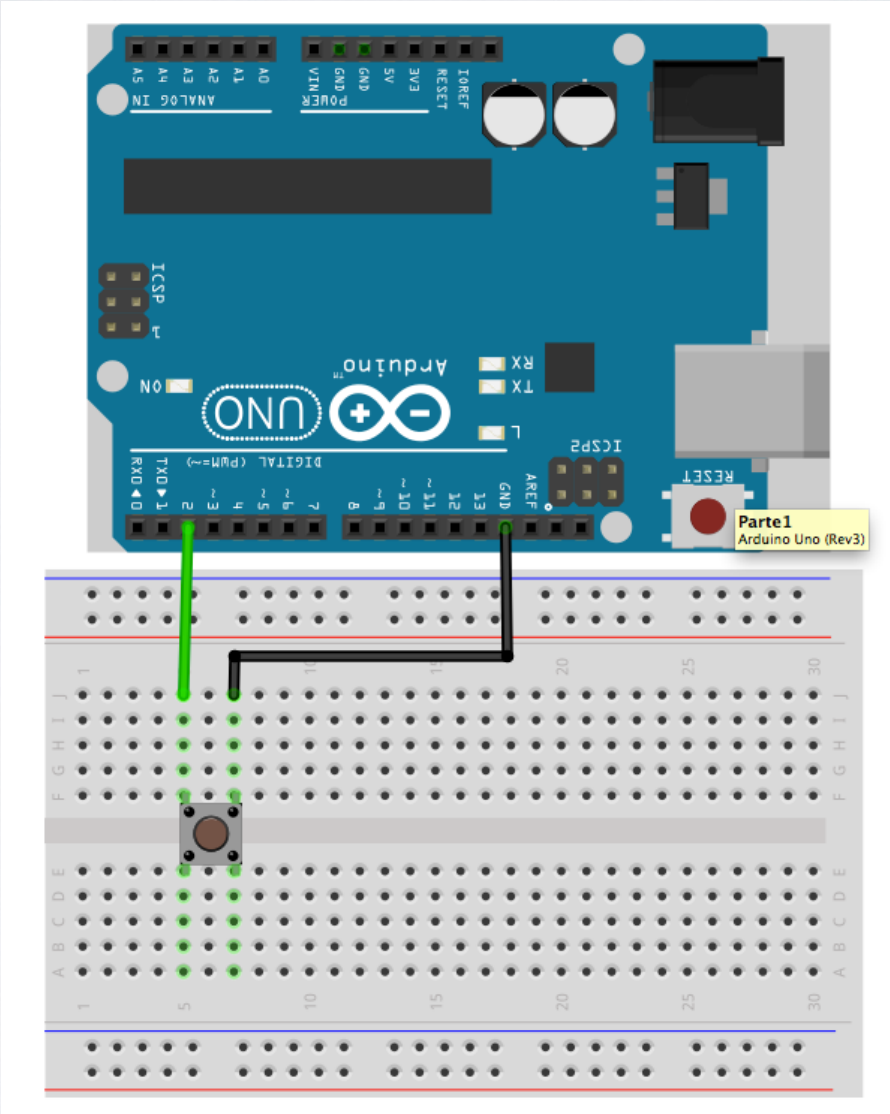
INTERRUPCIONES SE WAKE UP

I**nterrupción externa:** Sale del modo Sleep con ayuda de una interrupción externa (0 o 1).

Timers: Ese pueden utilizar las interrupciones de los timers para despertar el sistema periódicamente, llevar a cabo una acción y volver a dormir.

WDT. Despierta periódicamente al sistema, llevar a cabo una acción y volver a dormir. El WDT proporciona más tiempo de sueño y menor consumo de energía.

SLEEP MODE INTERRUPTION EXT

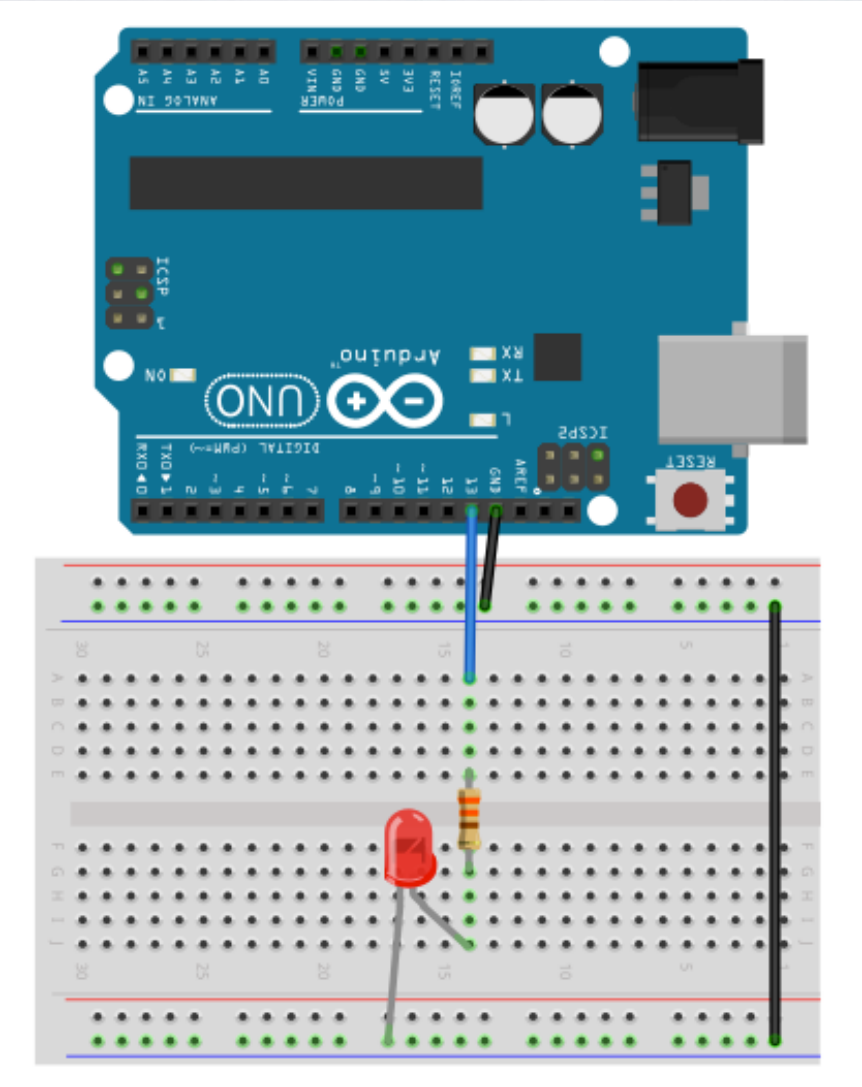


MIGUELRUIZ
MECHATRONIC ENGINEER

SLEEP MODE TIMERS

Timer	Tamaño	Max tiempo de espera	Modo de energía Min
Timer0	8 bit	16.4ms	IDLE
Timer1	16 bit	4.1s	IDLE
Timer2	8 bit	16.4ms	POWER_SAVE
Watch Dog Timer	N / A	8s	PWR_DOWN

SLEEP MODE TIMER I



MIGUELRUIZ
MECHATRONIC ENGINEER

WDT

El WDT en el ATmega328 tiene su propio oscilador interno de 128kHz a diferencia d ellos otros Timers.

Es este oscilador independiente que permite al WDT para funcionar en el modo de consumo mínimo: SLEEP_MODE_PWR_DOWN.

El WDT también tiene un pre-escalador, que se usa para configurar el período de tiempo de espera.

Puede ser utilizad como interrupción o también para hacer Reset al sistema.

WDT

MCUSR

Bit	7	6	5	4	3	2	1	0
0x55	-	-	-	-	WDRF	BORF	EXTRF	PORF
R/W?	R	R	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	-	-	-	-

WDTCSR

Bit	7	6	5	4	3	2	1	0
0x60	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0
R/W?	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	x	0	0	0

WDT

Table 10-1. Watchdog Timer Configuration

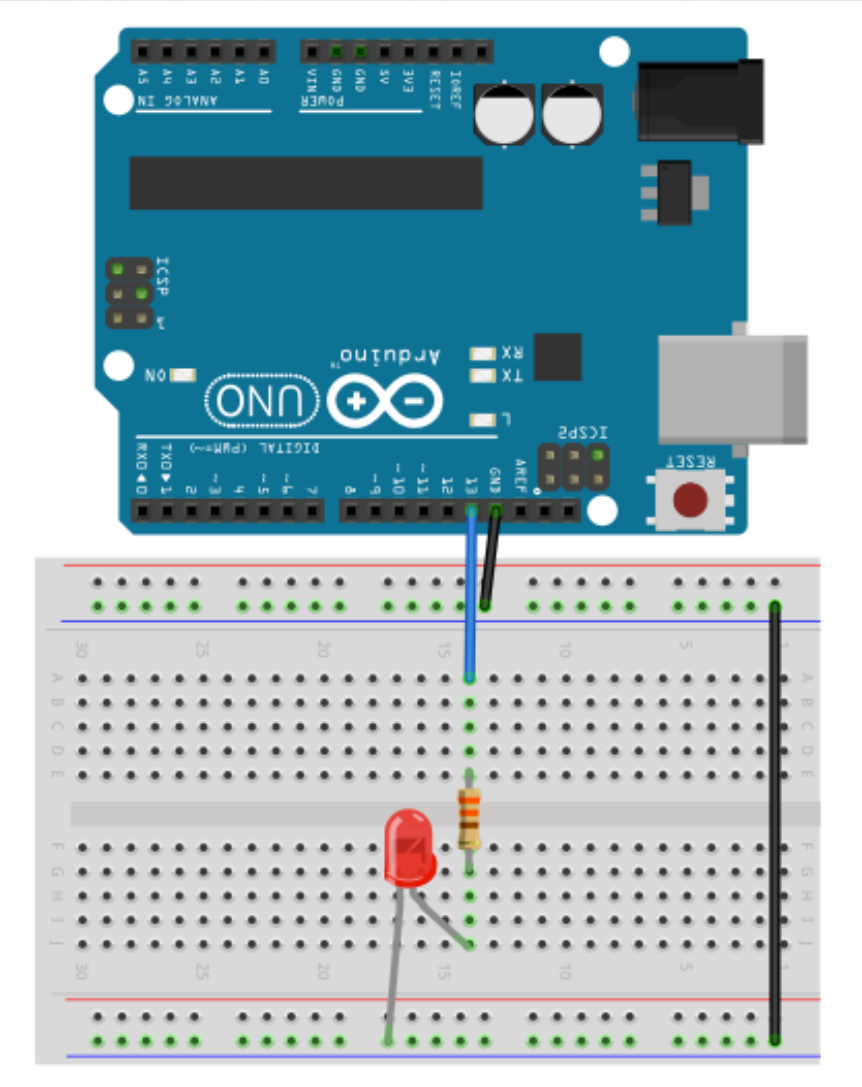
WDTON[1]	WDE	WDIE	Mode	Action on Time-out
1	0	0	Stopped	None
1	0	1	Interrupt Mode	Interrupt
1	1	0	System Reset Mode	Reset
1	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
0	x	x	System Reset Mode	Reset

WDT

Table 10-2. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 5.0V
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32 ms
0	0	1	0	8K (8192) cycles	64 ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32768) cycles	0.25 s
0	1	0	1	64K (65536) cycles	0.5 s
0	1	1	0	128K (131072) cycles	1.0 s
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s
1	0	1	0	Reserved	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

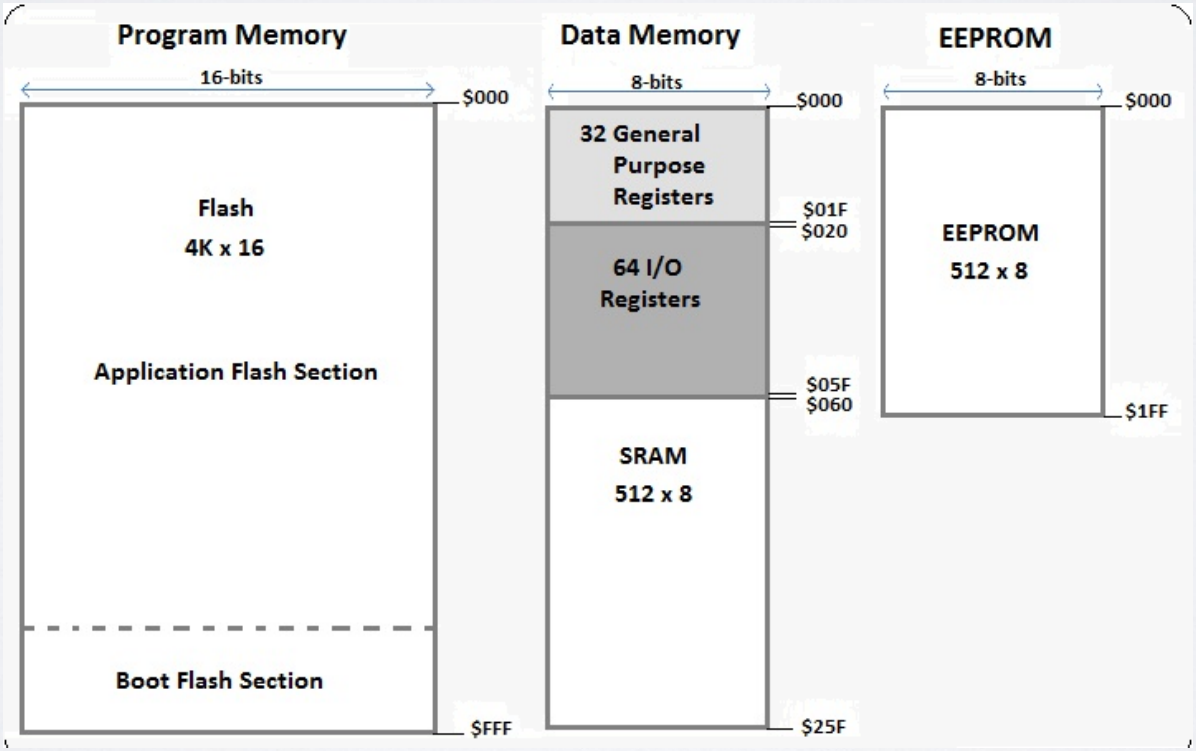
SLEEP WDT



MIGUELRUIZ
MECHATRONIC ENGINEER

EEPROM

La EEPROM (Electrically Erasable Programmable Read-Only Memory) es un tipo de memoria que no necesita energía para mantenerse (por ello la llaman memoria no volátil), permite almacenar información de forma "permanente", dado que la podemos usar para almacenar datos incluso cuando nuestro microcontrolador esta desconectado.



EEPROM

Placa	Memoria
UNO, Lilypad, Leonardo, Micro	1K
Mega	4K
DUE	-
Mano, Pro	512 Bytes

EEPROM

```
#include <EEPROM.h>
```

```
EEPROM.write( dirección , byte );
byte valor = EEPROM.read( dirección );
```

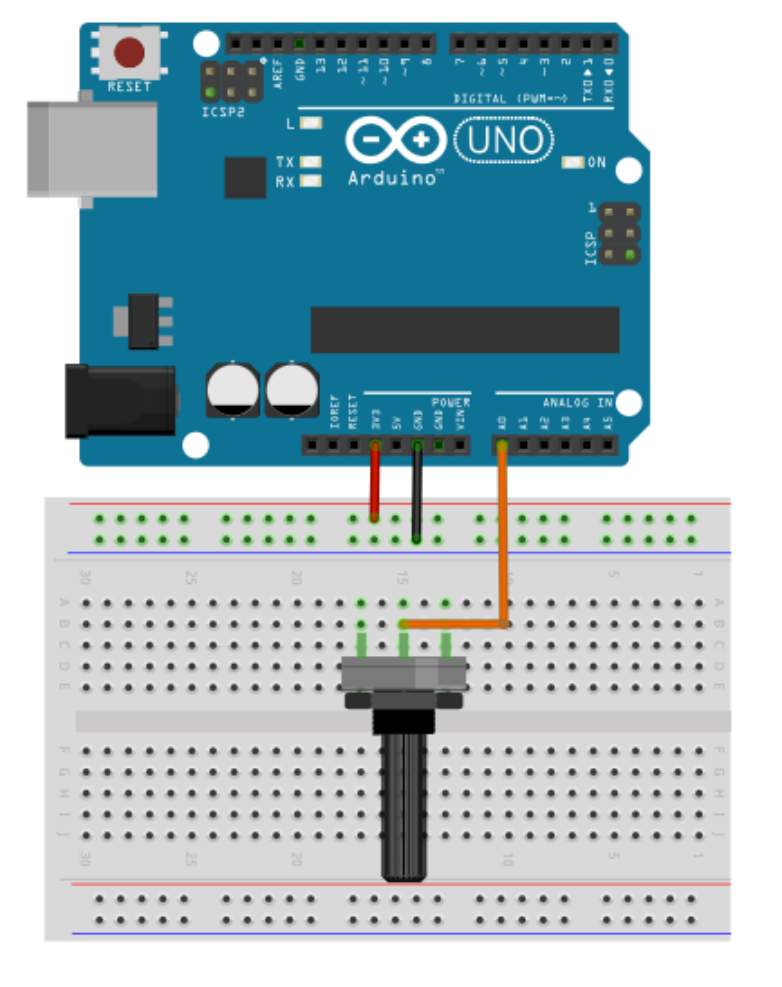
- Existen 512 direcciones, de 0 a 51.
- Se utiliza write para escribir en el registro deseado.
- Mientras read se utiliza para leer el registro.

SEPARACIÓN DE BYTES

```
lowByte( datatype );
highByte( datatype );
```

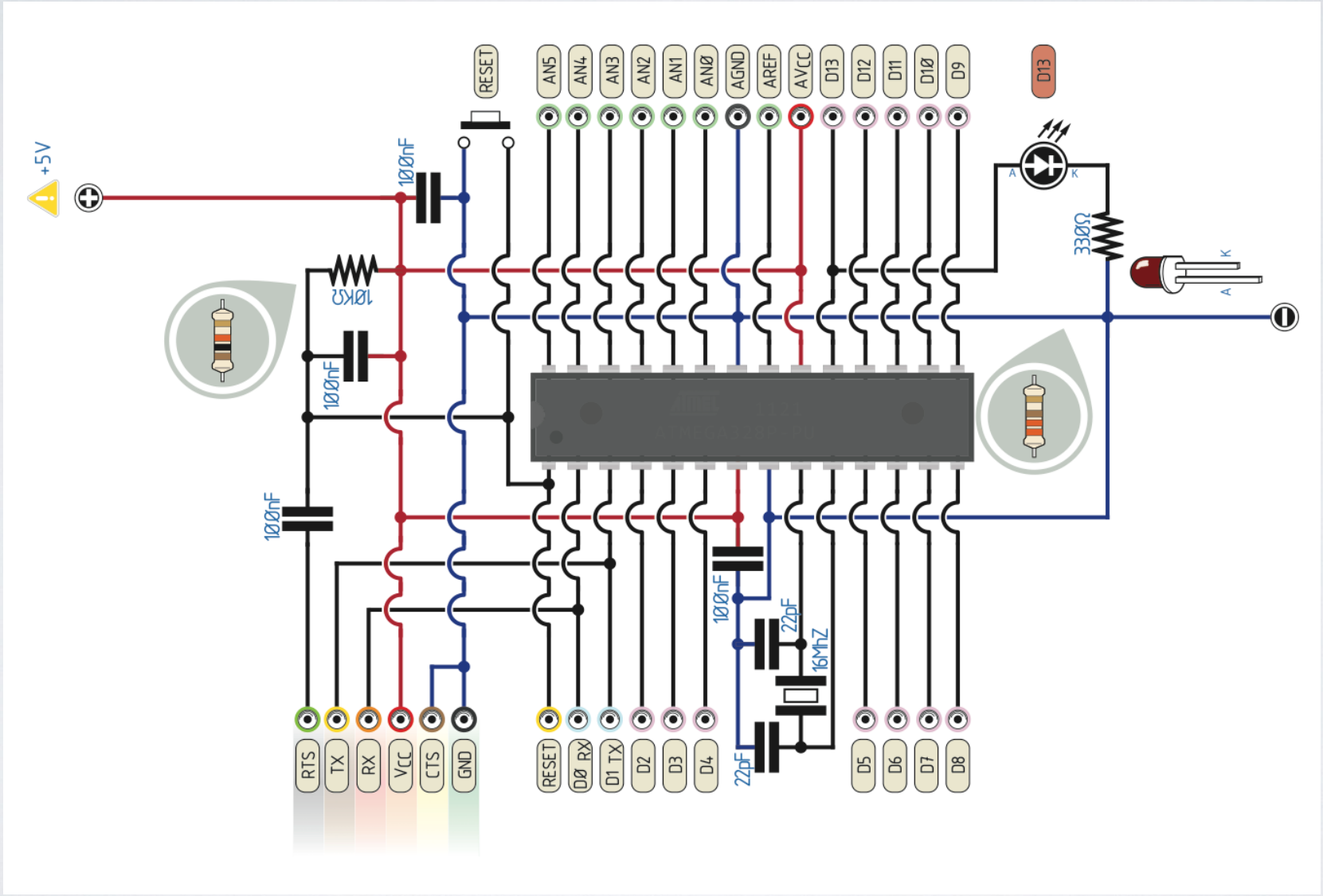
- Se utilizan esta dos funciones para separara registros de 16 bits en 2 registros de 8 bits.
- La función low devuelve el vector bajo de 16 bits.
- Mientras high retorna el vector alto de 16 bits.

ALMACENANDO



MIGUELRUIZ
MECHATRONIC ENGINEER

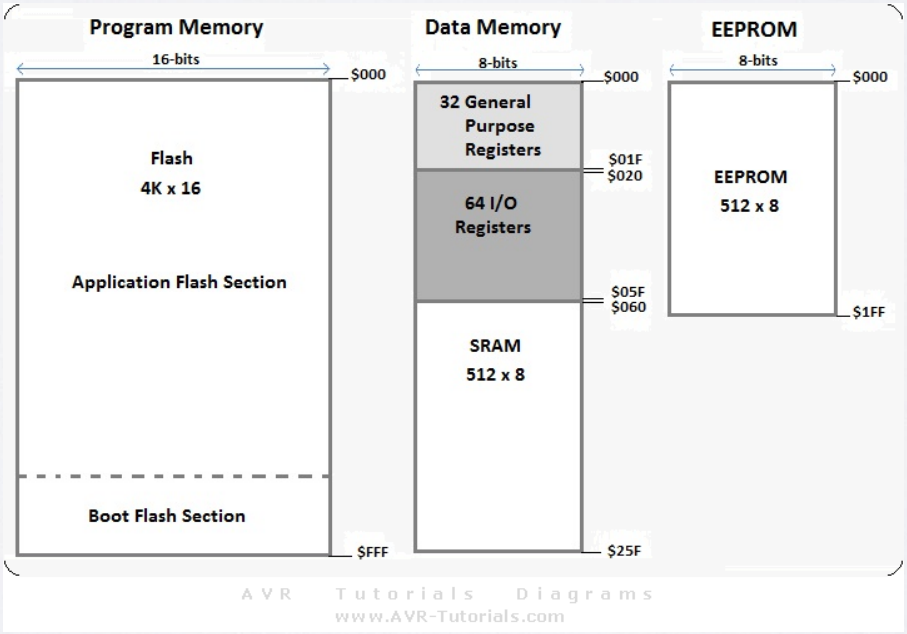
CIRCUITO BASICO



BOOTLOADER

Un gestor de arranque o bootloader es un programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, y que está diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar.

En la placa Arduino el microcontrolador contiene precargado el bootloader que es de gran utlidad para cargar aquel programa que quedo en nuestro microcontrolador cuando lo desconectamos por última vez.



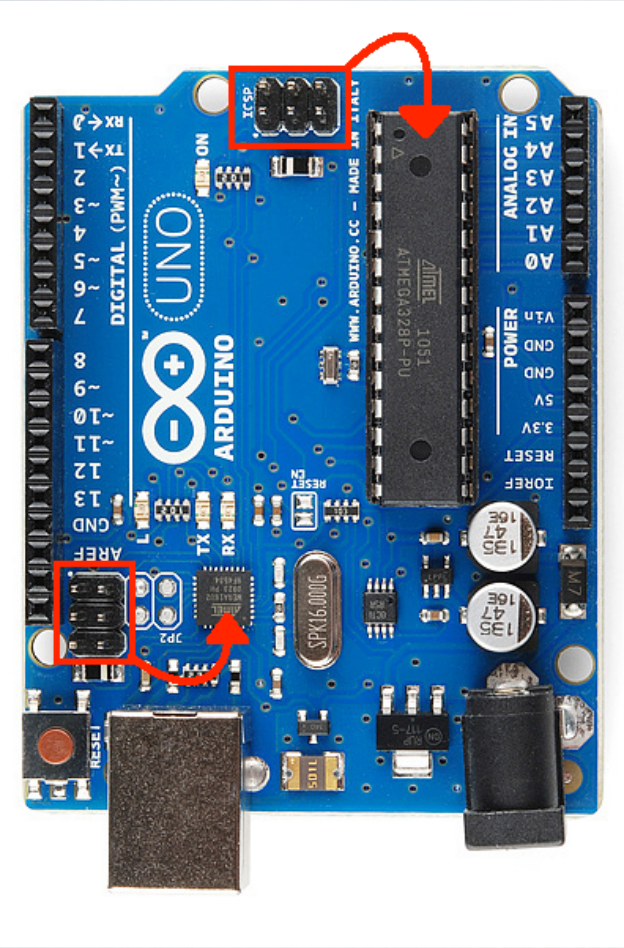
MIGUELRUIZ
MECHATRONIC ENGINEER

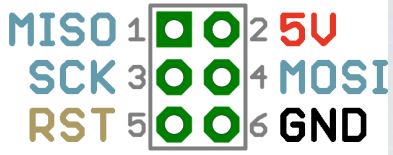
PROGRAMACIÓN ICSP

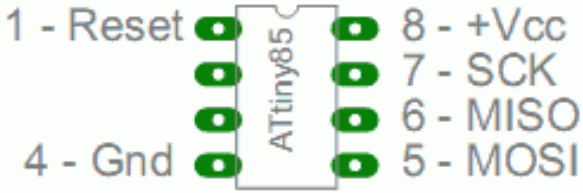
Cuando se desarrollan sistemas microcontrolados en altas producciones resulta muy ineficiente utilizar programadores por RS232 para una producción en serie.

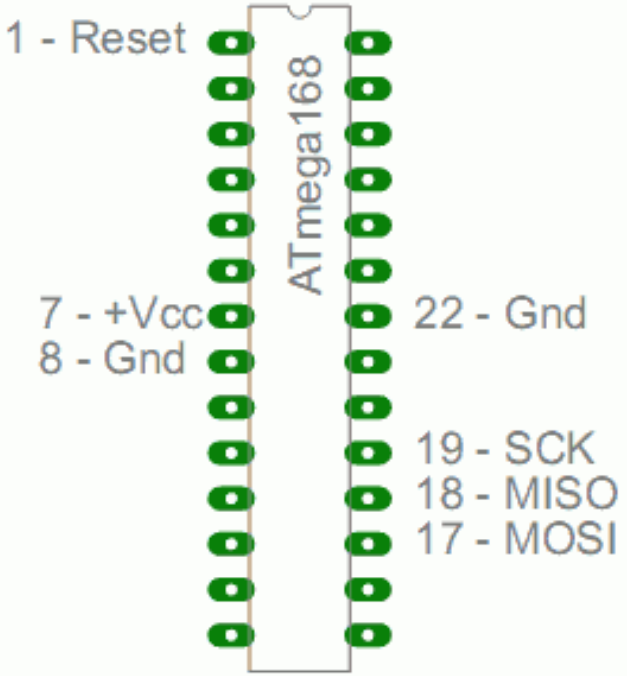
Es por ello que se diseño que existe la programación ICSP (Inter Circuit Serial Programer) para programar y reprogramar el micro controlador "en circuito".

PROGRAMACIÓN ICSP



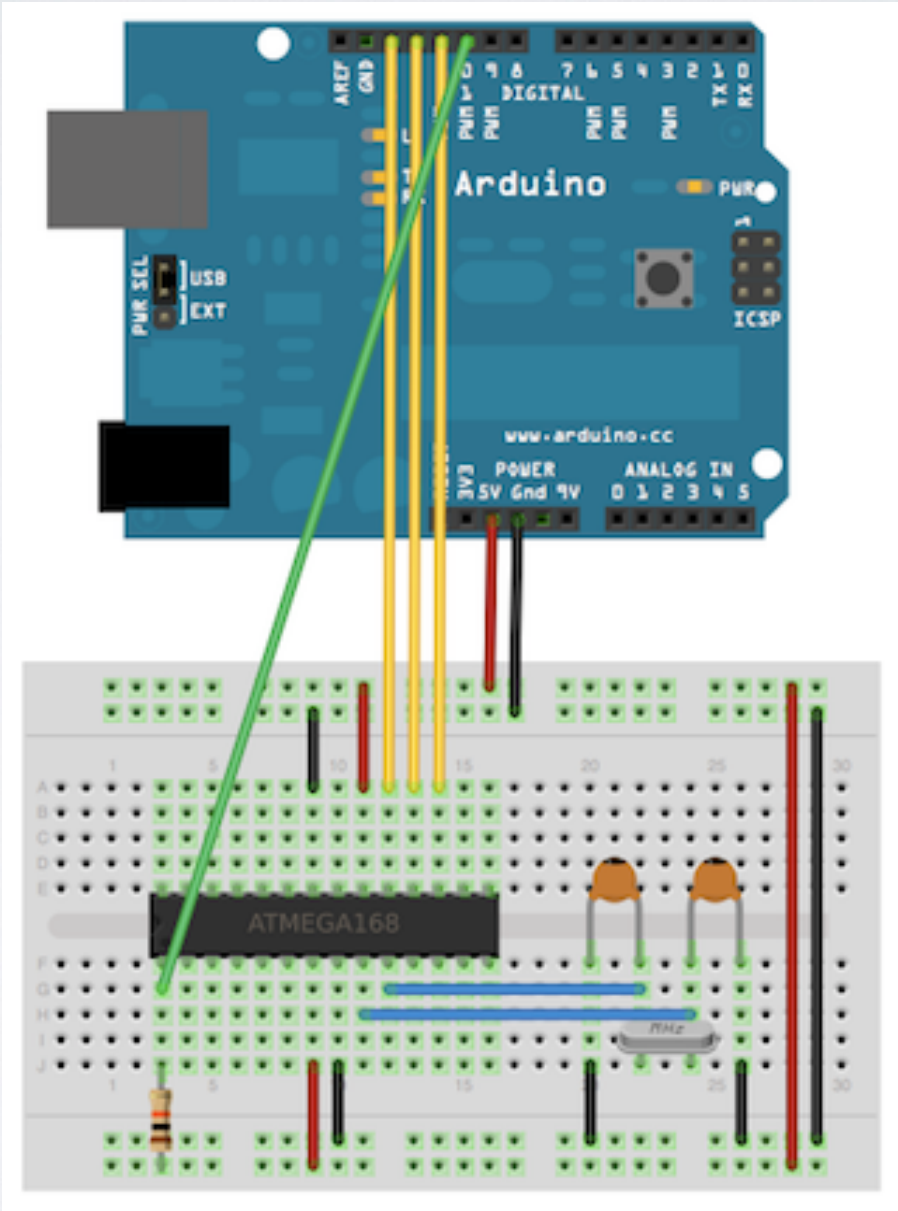






MIGUELRUIZ
MECHATRONIC ENGINEER

PROGRAMACIÓN ICSP



MIGUELRUIZ
MECHATRONIC ENGINEER

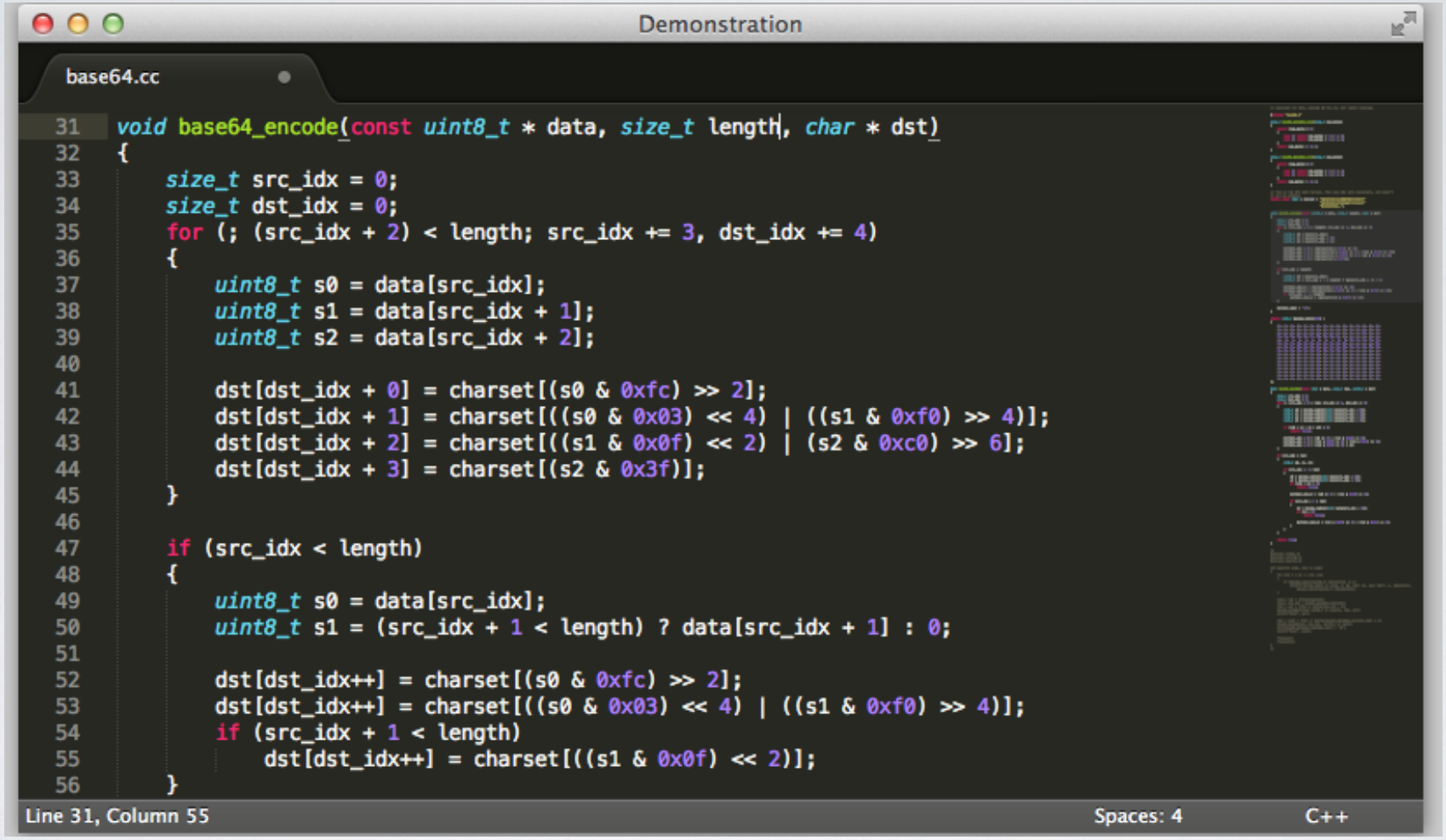
SUBLIME TEXT

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




```cpp
void base64_encode(const uint8_t * data, size_t length, char * dst)
{
    size_t src_idx = 0;
    size_t dst_idx = 0;
    for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = data[src_idx + 1];
        uint8_t s2 = data[src_idx + 2];

        dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
        dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
        dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
        dst[dst_idx + 3] = charset[(s2 & 0x3f)];
    }

    if (src_idx < length)
    {
        uint8_t s0 = data[src_idx];
        uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;

        dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
        dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
        if (src_idx + 1 < length)
            dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
    }
```

MIGUELRUIZ
MECHATRONIC ENGINEER

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

MIGUELRUIZ
MECHATRONIC ENGINEER

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/Minusculas
Ctrl + / Ctrl + ⇧ + /	Comentarios
Ctrl + Space	Navegación en auto-compleat

Creado por: Miguel Angel Ruiz Gálvez
Contacto: miguelo.me

Agradecimientos a:

- Massimo Banzi
- Hernando Barragan
- David Cuartielles
- Pighixxx

MIGUELRUIZ
MECHATRONIC ENGINEER

creative commons