

Análisis del consumo de energía de Arduino

Pedro Pillado García-Gesto

Resumen

- **Evaluación del consumo energético del Arduino Uno.**
- **Estudio del efecto de las modificaciones en la configuración en el consumo energético.**
- **Posibles mejoras y estudio de las diferentes alternativas.**
- **Valoración de alternativas al Arduino uno.**
- **Estudio detallado del consumo de energía de otras alternativas al Arduino uno.**

Hardware utilizado

- **Arduino uno**
- **Mini Ultra 8Mhz**
- **Arduino Mega**
- **Xbee**
- **Multímetro y osciloscopio**

Introducción

- Se hace una introducción a la materia para aquellos ajenos a ella.
- Introduce los tipos de arduino que existen, así como sus usos.

Característica de Arduino	UNO	Mega 2560	Leonardo	DUE
Tipo de microcontrolador	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E
Velocidad de reloj	16 MHz	16 MHz	16 MHz	84 MHz
Pines digitales de E/S	14	54	20	54
Entradas analógicas	6	16	12	12
Salidas analógicas	0	0	0	2 (DAC)
Memoria de programa (Flash)	32 Kb	256 Kb	32 Kb	512 Kb
Memoria de datos (SRAM)	2 Kb	8 Kb	2.5 Kb	96 Kb
Memoria auxiliar (EEPROM)	1 Kb	4 Kb	1 Kb	0 Kb

Introducción

- Introduce la materia del proyecto.
- Tiempo de uso que diferentes baterías pueden proporcionar al Arduino uno.

Tipo	Capacidad (mAh)	Descarga (%mes)	Auto descargado (μ A)
CR1212	18	1	0,25
CR1620	68	1	0,95
CR2032	210	1	3
NiMH AAA	900	30	375
Alkaline AAA	1250	2	35
NiMH AA	2400	30	1000
Alkaline AA	2890	2	80

Soluciones software

• Modos de sleep:

Sleep Mode	Active Clock Domains					Oscillators		Wake-up Sources							Software BOD Disable
	clk _{CPU}	clk _{FLASH}	clk _{I/O}	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	
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

Desactivación de periféricos

- **Recorre la lista de periféricos existentes de serie en el Arduino uno para estudiar los beneficios que pueda aportar su configuración.**
- **ADC: Su desactivación no aporta mejora apreciable.**
- **PRR: Mejora el consumo en 1,25 mA (el 10% del consumo de la placa).**
- **Configurar pines en in o out y en low o high refleja pequeños cambios en el consumo. Las mediciones son dispares, por lo que no se sacan conclusiones.**

Desactivación de periféricos

- **Power Reduction Register:** Permite detener el reloj de algunos periféricos para ahorrar energía. Al desactivar todos los pines se obtiene un ahorro de 3 mA. Solo funciona en modo activo o sleep idle.
- En el mejor de los casos se obtiene un consumo de 9,7mA, en modo sleep.

Frecuencia de reloj

- **A continuación se estudia cómo la frecuencia de trabajo afecta al consumo.**
- **Debido al cristal del arduino la frecuencia máxima es de 16Mhz y solo puede dividirse, nunca incrementarse.**
- **Es posible disminuir la frecuencia modificando los ficheros de configuración de Arduino.**

Librerías

- Se hace un recorrido por las librerías más populares de gestión de modos de bajo consumo.

- Las más populares son:

- Jeelib

- N0m1

- Low.power

- **Es una colección de cabeceras, clases y programas sencillos para usar con el Arduino IDE.**
- **Incluye un modo de bajo consumo controlado por el WatchDog Timer (WDT)**
- **Permite desactivar periféricos para el modo sleep**

N0m1

Permite programar la activación de un modo sleep determinado durante un periodo determinado o hasta que una interrupción lo despierte.

```
void loop()
{
  delay(100); //tiempo para escribir via serial
  Serial.println("ejecuta aquí tu código");
  Serial.print("durmiendo hasta interrupción");
  delay(100); //tiempo para escribir via serial
  sleep.pwrDownMode(); //escoge sleep mode
  sleep.sleepInterrupt(0,FALLING); //duerme hasta interrupción
}
```

Low-power

Permite definir muchas variables en una sola instrucción.

Permite seleccionar modo sleep y configurar qué elementos queremos activos.

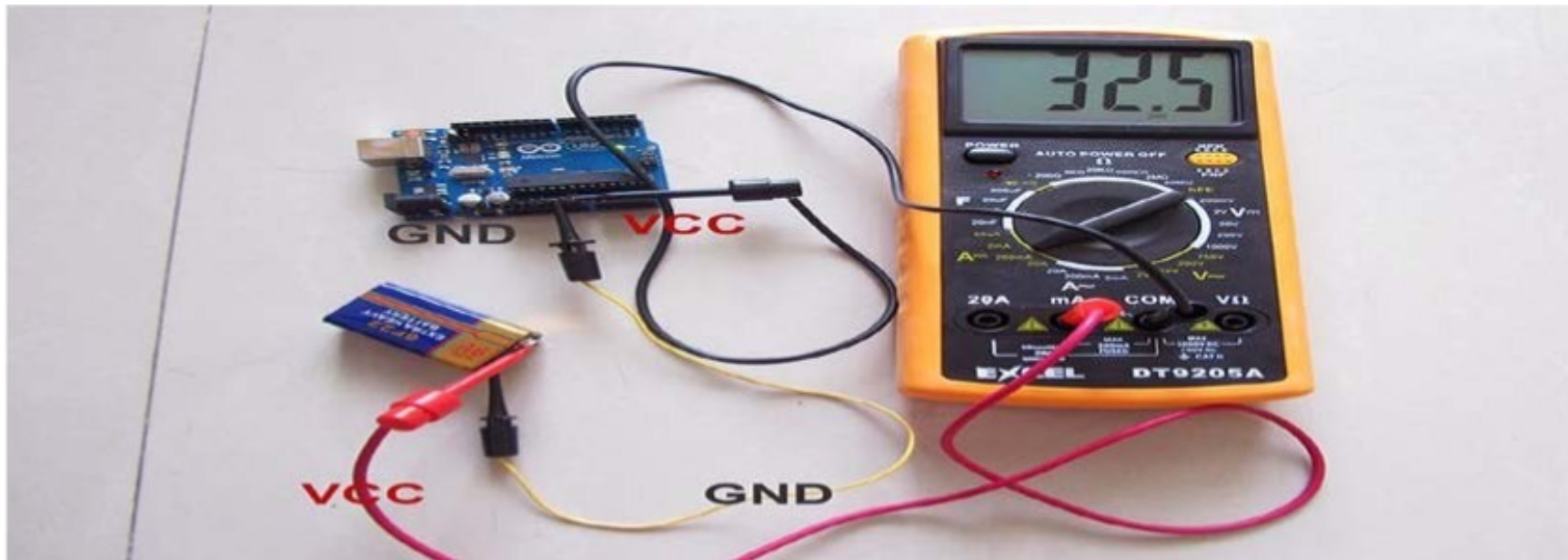
```
void loop()
{
    // La interrupción salta cuando el pin wake up es LOW
    attachInterrupt(0, wakeUp, LOW);
    // Entra en modo powerdown con ADC y BOD desactivados
    // Despierta cuando el pin wake up es LOW
    LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF);
    // Desactiva interrupción.
    detachInterrupt(0);
    // Realiza cualquier acción. Leer sensor, transmitir, etc.
}
```

Mediciones

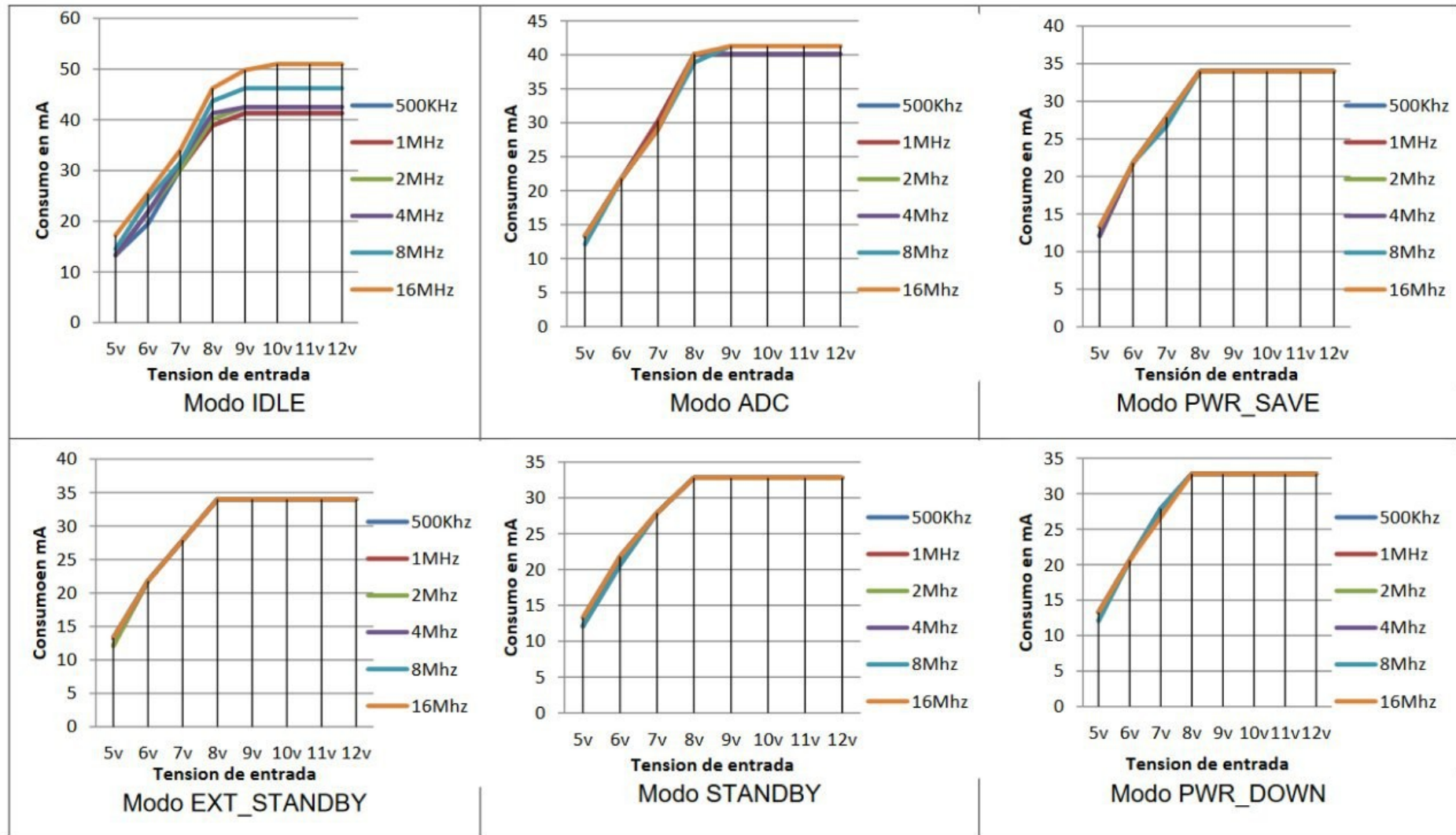
- Para las mediciones estáticas se utiliza un multímetro digital estándar.
- Para las mediciones dinámicas se mide la tensión sobre una resistencia de control colocada en serie con el Arduino, realizada sobre una impedancia flotante de $32,9\Omega$ nominales calculada en función de la corriente esperada. Para ello se usa un osciloscopio portátil.

Mediciones

Equipo utilizado para las mediciones:



Modos sleep



Frecuencia en modos sleep

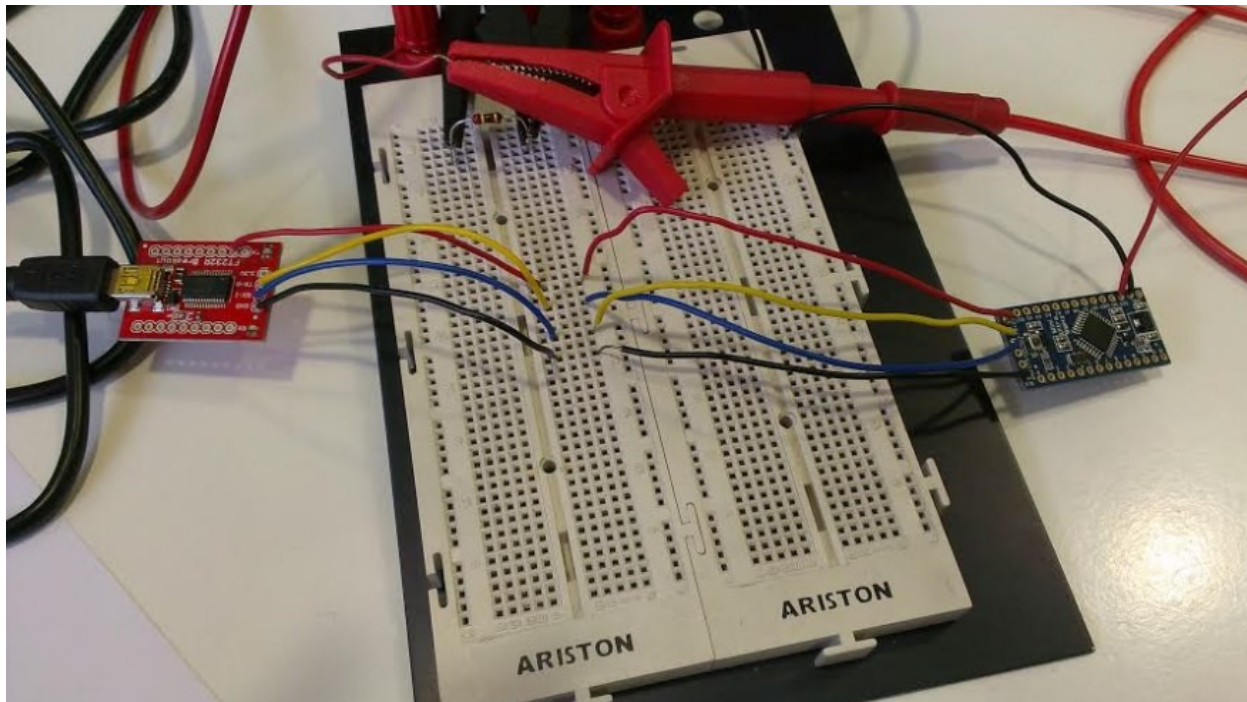
Se hace constantemente referencia a que es posible usar la máxima frecuencia del micro en los modos sleep sin consumir más energía.

Pero se debe a que la señal de reloj principal se desactiva.

Más adelante se contradice explicando esto último.

Mini Ultra 8Mhz

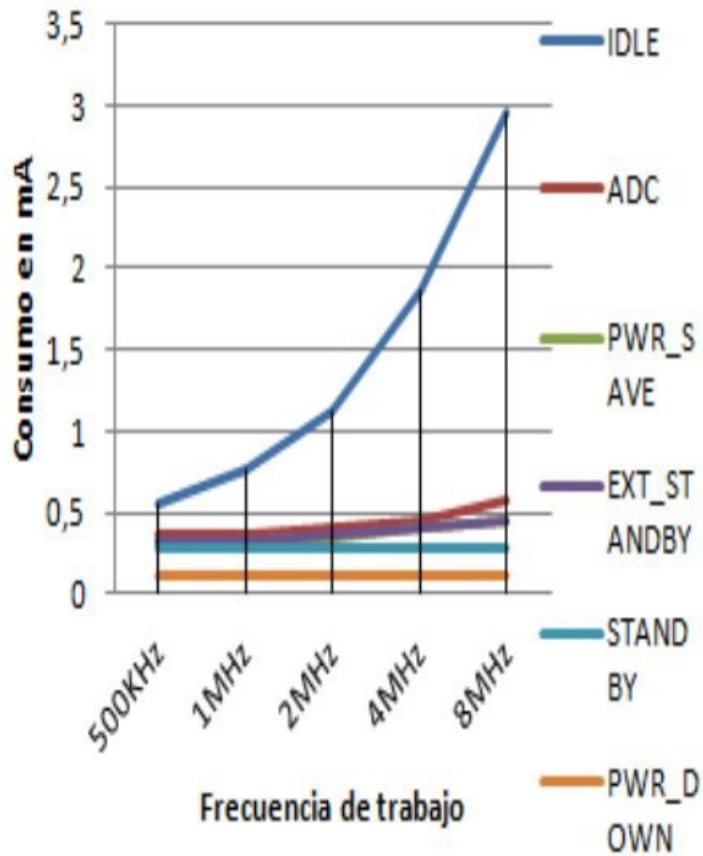
Se trata de un microcontrolador de pequeño tamaño y de reducidas prestaciones orientado al ahorro de consumo energético



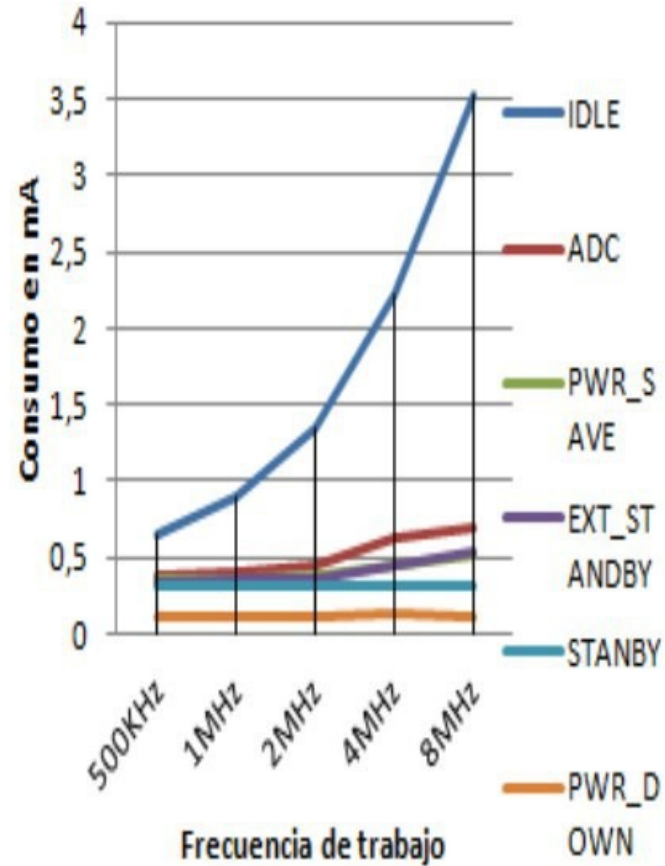
Mini Ultra 8Mhz

- Se usa un adaptador FTDI para conectar el arduino al PC, se sueldan los cables a las placas.
- Tiene un rango de alimentación limitado entre 3,3 y 6V que hay que tener en cuenta para alimentarlo.
- Para probar el consumo se usará de nuevo el programa Blink en combinación con los diferentes modos sleep.

Consumo del Mini Ultra



Gráfica comparativa modos Sleep Mini Ultra a 2.8V



Gráfica comparativa modos Sleep Mini Ultra a 3.3V

Mini Ultra con librerías

·Estos son los consumos del Mini Ultra en el programa Blink con las diferentes librerías ya vistas de gestión de energía en combinación con modificaciones en la frecuencia de trabajo.

	1MHz	2MHz	4MHz	8MHz
Jeelib	1,5	1,5	1,5	1,6
N0m1	1,6	1,7	1,8	1,8
Low.power	1,5	1,5	1,4	1,5

(mA)

Arduino Mega y Xbee

- **La placa Arduino Mega es una placa de mayor tamaño y mejores prestaciones que el Arduino uno. Ofrece más pines de I/O así como un consumo más elevado.**
- **Los módulos Xbee se utilizan para comunicaciones inalámbricas entre Arduinos a hasta 100m de distancia de forma sencilla.**

Arduino Mega

Microcontrolador	ATmega1280
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines E/S digitales	54 (14 proporcionan salida PWM)
Pines de entrada analógica	16
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria Flash	128 KB de las cuales 4 KB las usa el gestor de arranque(bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

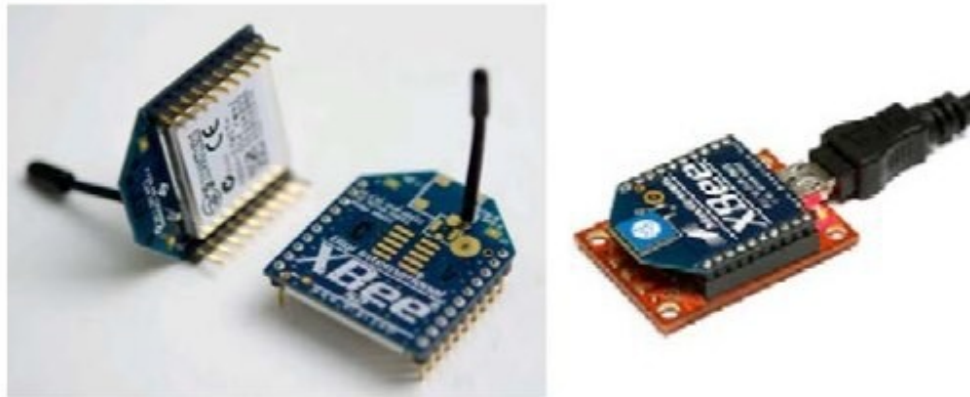
Arduino Mega

·Se observa como a partir de 8V los valores se mantienen estables, siendo recomendable alimentar a 5V para ahorrar energía.

	500KHz	1MHz	2MHz	4MHz	8MHz	16MHz
5v	18,1	18,8	19,6	21,8	25,4	35,2
6v	26,2	26,9	28	30,7	35,3	44,2
7v	49	49,8	51,2	54,3	58,3	67,4
8v	49,7	50,3	51,4	54,8	59,7	71,3
9v	49,7	50,3	51,4	54,8	59,7	71,3
10v	49,7	50,3	51,4	54,8	59,7	71,3
11v	49,7	50,3	51,4	54,8	59,7	71,3
12v	49,7	50,3	51,4	54,8	59,7	71,3

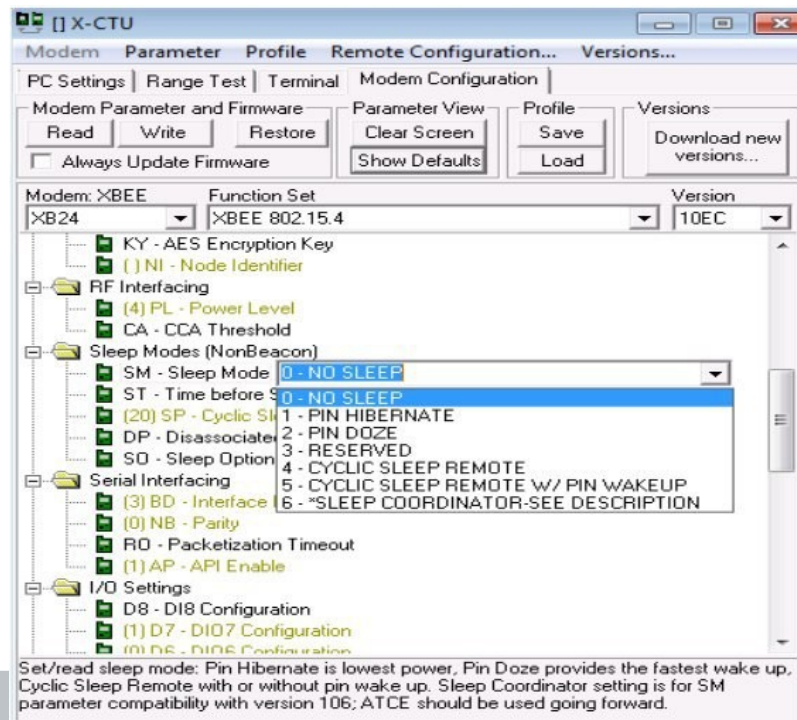
Xbee

- Se alimenta de la propia placa Arduino.
- La forma más sencilla de programar los módulos Xbee es a través de un adaptador FTDI para conectarlo al PC por USB.



Xbee

•El programa X-CTU pone a disposición del programador varios modos sleep mediante una interfaz que permite elegir el modo sleep y el tiempo que deseamos que permanezca en él.



Arduino Mega + Xbee

·A continuación se mide el consumo del conjunto del Arduino Mega con el Xbee probando diferentes combinaciones de modos sleep.

	Consumo total en mA	Consumo XBEE en mA
SM1	69,4	0,1
SM2	70,1	0,8
SM3	114,1	44,8
SM4	114,1	44,8
SM5	114,1	44,8
SM6	114,1	44,8

Comparativas de consumo

La única placa viable para trabajo autónomo es el Arduino Mini Ultra como se ve en la figura.

	Arduino Uno	Arduino Mini Ultra 8MHz	Arduino Mega	Waspnote Libelium
Consumo operativo	12.1 mA	0.76 mA	18.1 mA	15 mA
Consumo en reposo	12.1 mA	0.12 mA	16.7mA	0.05 mA
Precio	Kit 97€	Unidad 9€	Unidad 47,19 €	Kit 199€
	Unidad 14€			

Conclusiones

- **Tras las mediciones se comprueba que el arduino uno no es apto para entornos donde se requiera un bajo consumo.**
- **Solo se han estudiado soluciones software, las cuales son de baja efectividad.**
- **El Arduino Mini Ultra sí permite ser usado como dispositivo de bajo consumo, pero sus prestaciones son muy bajas.**
- **El Arduino Mega + Xbee resultó inviable para entornos de bajo consumo.**

Sugerencia de continuación

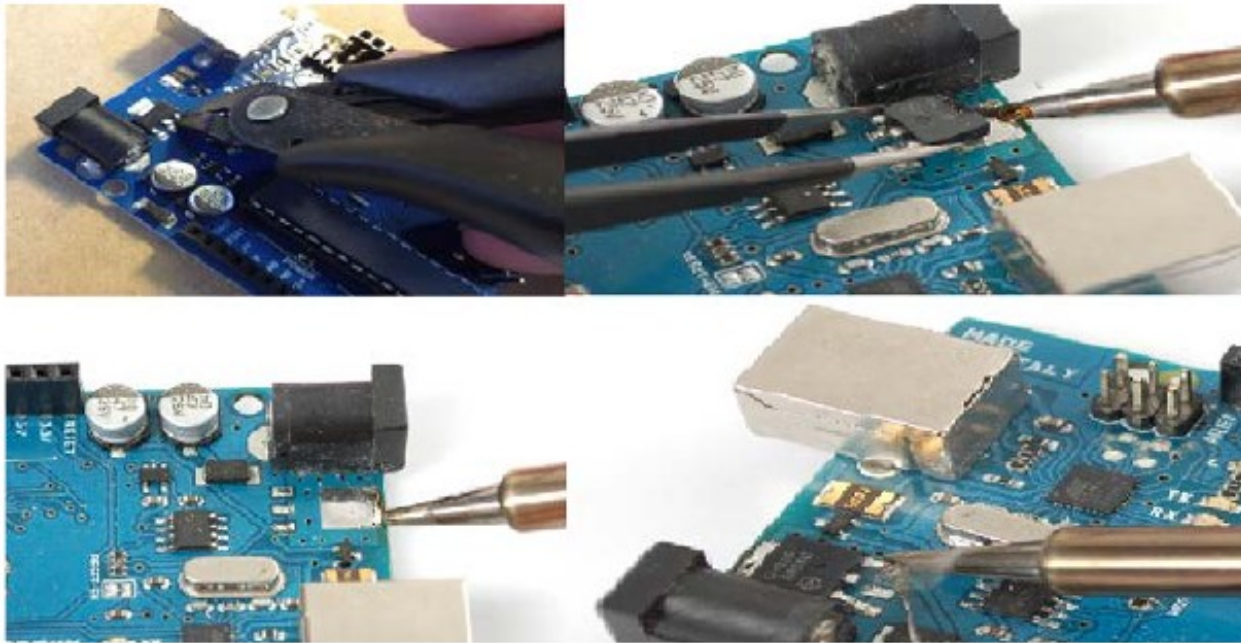
- **Se podría estudiar la viabilidad de formas de autoabastecimiento de energía (placas solares, generadores mecánicos...)**

Soluciones hardware

- **Sustituir el regulador de tensión.**
- **Supone una gran parte del consumo energético de la placa.**
- **Su sustitución es sencilla**
- **El modelo que usa el Arduino uno (LM117-5) es poco eficiente.**
- **Hasta el 58% de la energía consumida se desperdicia produciendo calor.**
- **Según indagaciones el modelo LTC3525 es el más adecuado. (~95% de eficiencia)**

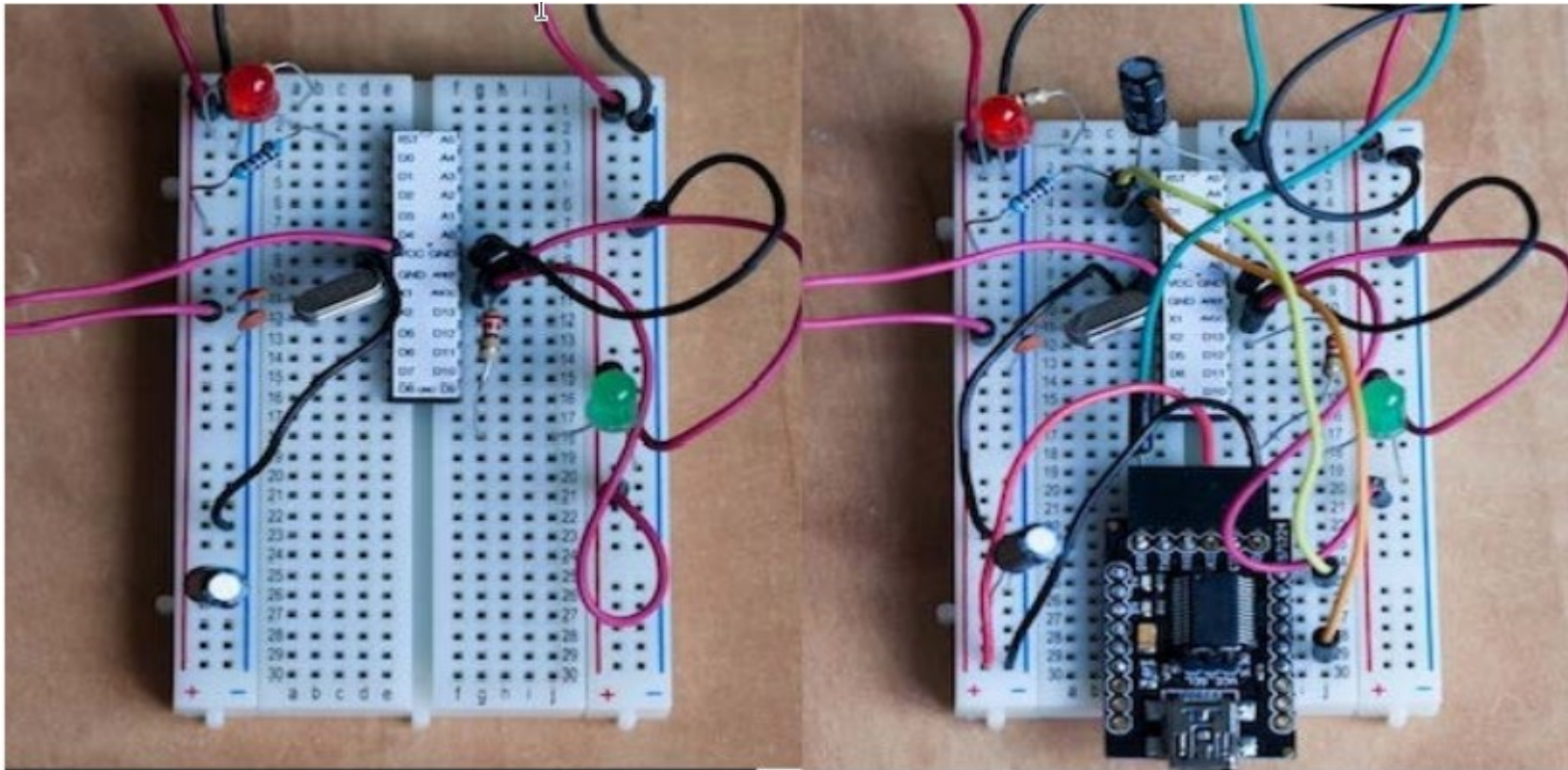
Soluciones hardware

- Este modelo permite alimentar a 3,3V.
- Permite una integración más conveniente con otros dispositivos.



Soluciones hardware

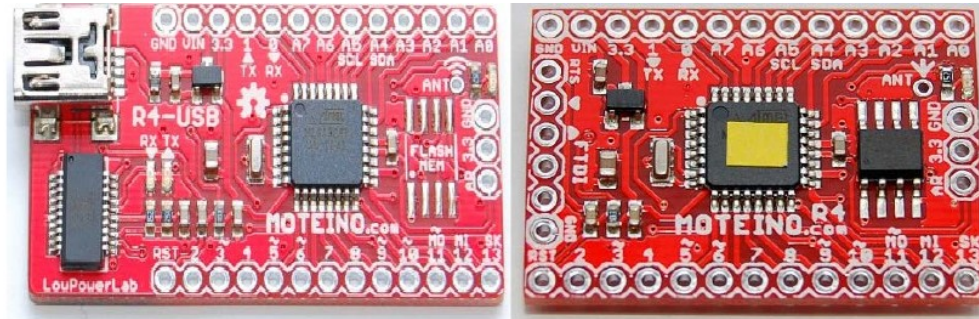
- Imitar la placa en un protoboard eliminando componentes inútiles para el uso requerido.



ATmega 328, reloj 16Mhz, condensadores, leds y una placa de carga FTDI.

Alternativas a arduino

• Moteino



• Mini Ultra 8Mhz



• Tinyduino

