USAR ARDUINO PARA REPROGRAMAR EL BOOTLOADER DE OTRO ARDUINO



El bootloader de Arduino es un software alojado en la memoria flash que nos permite programar Arduino a través del puerto serie sin necesidad de usar un programador externo.

De forma resumida, durante el arranque de Arduino el bootloader comprueba si se está intentando programar el Arduino. Si es así, graba el programa en la memoria, y reinicia Arduino. En caso contrario, el bootloader ejecuta el último programa grabado.

El bootloader de Arduino es una de las partes esenciales en las que reside la comodidad y sencillez de uso de Arduino (y por tanto de su éxito). En general **lo normal es que no necesitemos lidiar con él**.

Sin embargo, hay varias circustancias en las que necesitaremos ser capaces de modificar el bootloader de Arduino. Por ejemplo:

- Los usuarios avanzados pueden querer modificar y personalizar el proceso de arranque.
- En proyectos grandes, podemos querer aprovechar el espacio ocupado por el bootloader.
- Algunos fabricantes envían sus placas sin el bootloader precargado.
- En alguna circunstancia el bootloader puede corromperse.

En cualquiera de los casos, **no necesitamos disponer de un programador externo** si no que podemos grabar el bootloader de un Arduino usando otro Arduino como programador.

Al Arduino que actúa como programador lo llamaremos **master**, y al que vamos a programar **slave**. La comunicación entre PC y master se realiza a través de puerto serie, mientras que el master se comunicará con el slave a través de SPI.

PREPARAR EL ARDUINO MASTER

En primer lugar vamos a configurar nuestro Arduino Master como programador. Para ello, simplemente tenemos que carga el Sketch "Arduino ISP" de la librería de ejemplos de Arduino. Lo cargamos en el Master como cualquier otro sketch.

Blink Arduino 1.6.5					×	
Archivo Editar Program Nuevo Abrir Open Recent Proyecto Ejemplos Cerrar Salvar Guardar Como Configurar Página	ma Herramientas Ctrl+N Ctrl+O > Ctrl+W Ctrl+S Ctrl+Mayús+S Ctrl+Mayús+P	Ayuda 01.Basics 02.Digital 03.Analog 04.Communication 05.Control 06.Sensors 07.Display 08.Strings 09.USB	> > > > > > > > > > > > > > > > > > > >	Ly.		^
Imprimir Preferencias	Ctrl+P Ctrl+Coma	10.StarterKit ArduinoISP	>			
Salir Void setup() { // initialize the d pinMode(led, OUTPUT } // the loop routine r void loop() { digitalWrite(led, H delay(1000);	Ctrl+Q igital pin as an); uns over and ove IGH); // turn // wait	Teensy Adafruit ADXL345 Adafruit BMP085 Library Adafruit HMC5883 Unified Adafruit ILI9341 Bridge DHT sensor library	> > > > > >	svel)		
<pre>digitalWrite(led, L delay(1000); }</pre>	OW); // turn // wait	EEPROM Esplora Ethernet Firmata GSM LiquidCrystal Robot Control Bobot Motor	> > > > > > > > > > > >	LOW		~

Así de fácil. Con esto hemos convertido nuestro Arduino en un programador ISP.

CONECTAR ARDUINO MASTER CON SLAVE

Desconectamos el master del ordenador, y conectamos master y slave. Para ello, alimentamos el Arduino Slave desde el Master conectando Vcc y Gnd de ambas placas. Por otro lado, conectamos los pines del bus SPI entra ambas placas.

Si, por ejemplo, estamos usando un Arduino UNO como programador Master y un Arduino Mini como slave, el esquema de conexión en el master a los pines del slave quedarían de la siguiente forma.



Mientras que la conexión vista desde slave (Mini en el ejemplo) a los pines del master serían las siguientes.



GRABANDO EL BOOTLOADER

Ahora podemos proceder a quemar el bootloader en el Arduino. Conectamos el master al ordenador a través del puerto serie, manteniendo la conexión entre master y slave.

Configuramos el IDE de Arduino. Seleccionamos:

- El puerto COM al que tenemos conectado el master.
- El modelo de placa del slave
- Como programador, "Arduino ISP"



Finalmente seleccionamos "Quemar bootloader" para iniciar el proceso de grabación.

Blink Arduino 1.6.5		-						
Archivo Editar Programa He	erramientas Ayuda							
Blink	Auto Formato Archivo de programa. Reparar codificación & Recargar. Monitor Serie	Ctrl+T Ctrl+Mayús+M	₽ ▼					
Turns on an LED on for This example code is in */ // Pin 13 has an LED conr // Pin 11 has the LED on	Placa: "Arduino Nano" Procesador: "ATmega328" Puerto Programador: "Arduino as ISP"	>						
<pre>// Pin 6 has the LED on // Pin 13 has the LED on // give it a name: int led = 13; // the setup routine runs of void setup() { // initialize the digital pinMode(led, OUTPUT); }</pre>	Quemar Bootloader]					
<pre>// the loop routine runs ov void loop() { digitalWrite(led, HIGH); delay(1000); digitalWrite(led, LOW); delay(1000); }</pre>	er and over again forever: // turn the LED on (HIGH is the // wait for a second // turn the LED off by making th // wait for a second	voltage level) e voltage LOW	~					
Arduino Nano, ATmega328 on COM11								

Tras un montón de lucecitas y parpadeos en los LED de ambas placas **veremos un mensaje** en el IDE informando de que el bootloader se he grabado correctamente.

Listo, hemos grabado con éxito el bootloader, y podemos desconectar ambos Arduinos y usarlos con normalidad.