

LABORATORIO DE JUGUETE

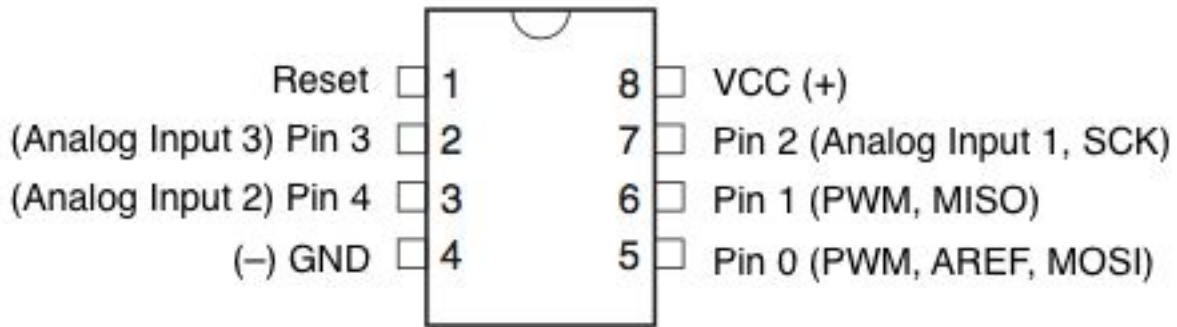
Tutorial: cómo programar el attiny85 en la IDE de Arduino

Antes de empezar refrescamos un poco la data acerca de la familia ATTINY25/45/85: Son los microcontroladores ideales para hacer tareas sencillas que no demanden muchos pines ni memoria.



Ésta familia de microcontroladores tiene apenas 8 patitas y su memoria flash (el lugar donde se alberga nuestro código) es de 8, 4 o 2 kilobytes (de acuerdo al número en su nombre). Pero para muchas de las cosas que solemos hacer con arduino suele ser suficiente!

ATtiny45 / ATtiny85



Como verás en el esquema de arriba, tiene 5 entradas/ salidas digitales de las cuales tres, pueden ser usadas como entradas analógicas. A que varios de tus proyectos en Arduino no usan más que esto, no?

Y cómo hacemos para programar estas cucarachitas usando la [IDE](#) de Arduino?

Éstos son los pasos a seguir:

- Si no has instalado la IDE de Arduino o tienes una versión anterior a la 1.7, ingresa al siguiente link, descarga e instala la última versión disponible de acuerdo a tu sistema operativo y arquitectura:

[WINDOWS](#)

[MAC](#)

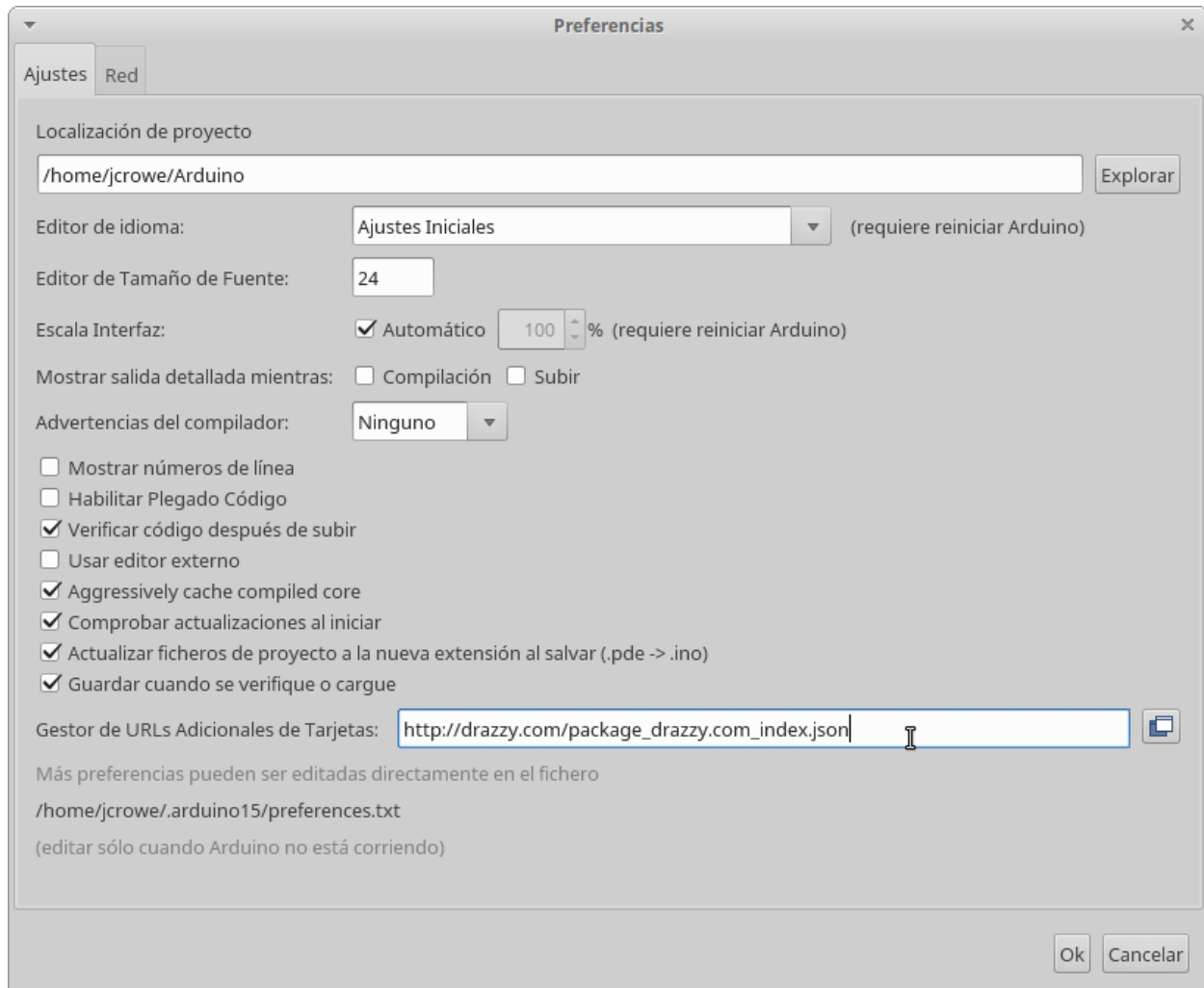
[LINUX \(32bit\)](#)

[LINUX \(64bit\)](#)

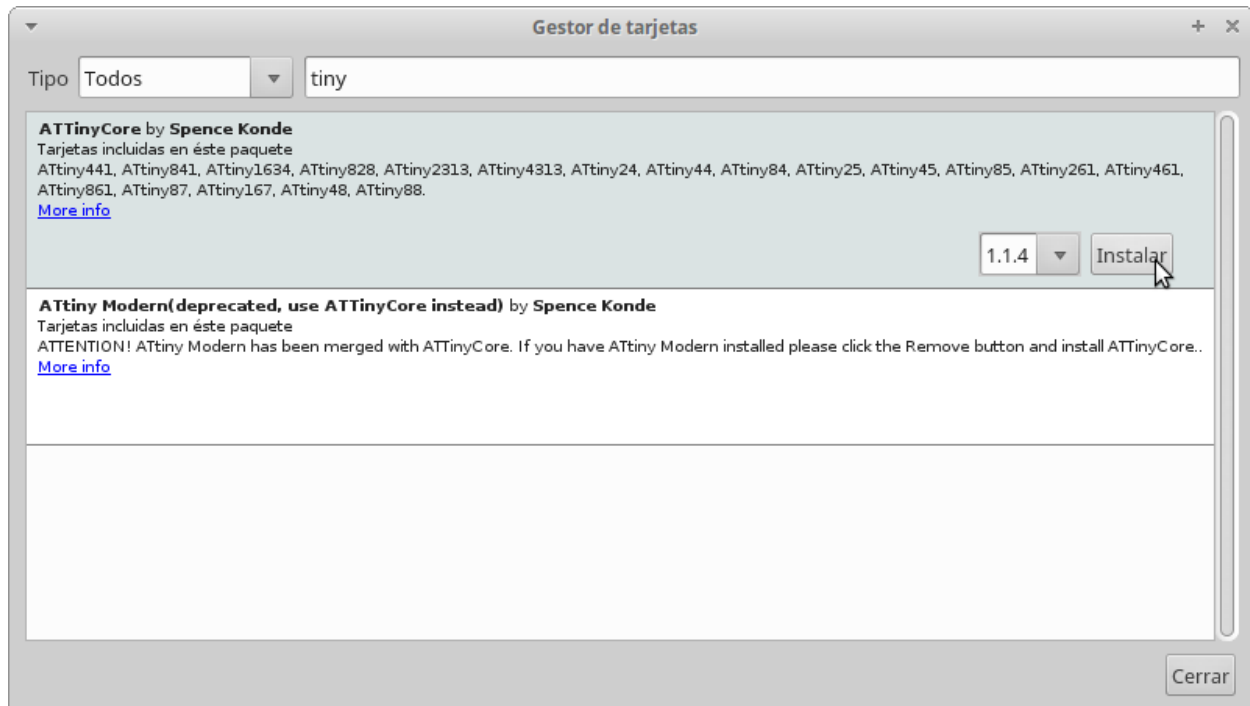
- A continuación te presentamos el Core (núcleo) que vamos a usar: [ATTinyCore \(de Spence Konde\)](#). Puedes echar un vistazo al repositorio o por ahora simplemente copiar la dirección de aquí debajo:

http://drazzy.com/package_drazzy.com_index.json

- Abrió la IDE de Arduino y dirígite a **Archivo/Preferencias**
- Allí encontrarás un cuadro con el nombre "Gestor de URLs adicionales de tarjetas ". Pegá en el cuadro la dirección que copiaste antes y hacé clic en OK



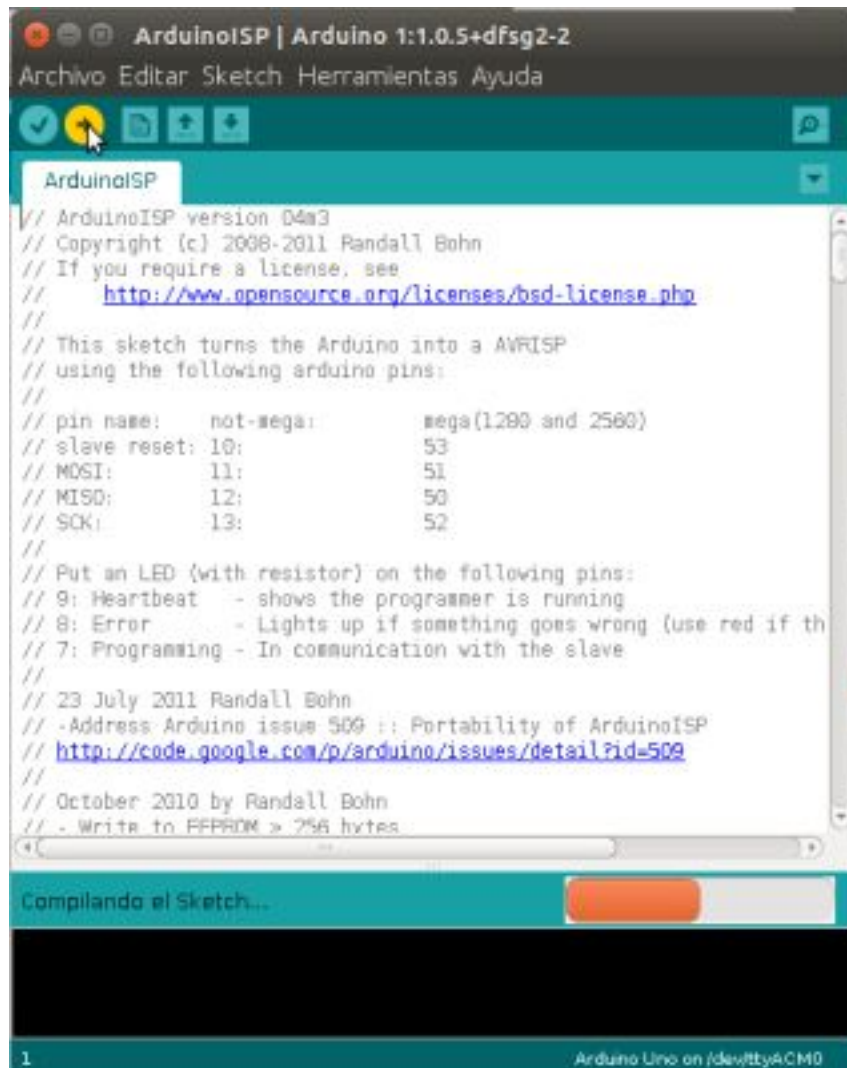
- Ahora dirígite a **Herramientas/Placa/Gestor de tarjetas**. Dale un momento al gestor para que se actualize e incorpore la ruta que pegaste en el paso anterior. Luego escribí "tiny" en el buscador. Aparecerá "ATTinyCore by Spence Konde"; hacé click en "instalar".



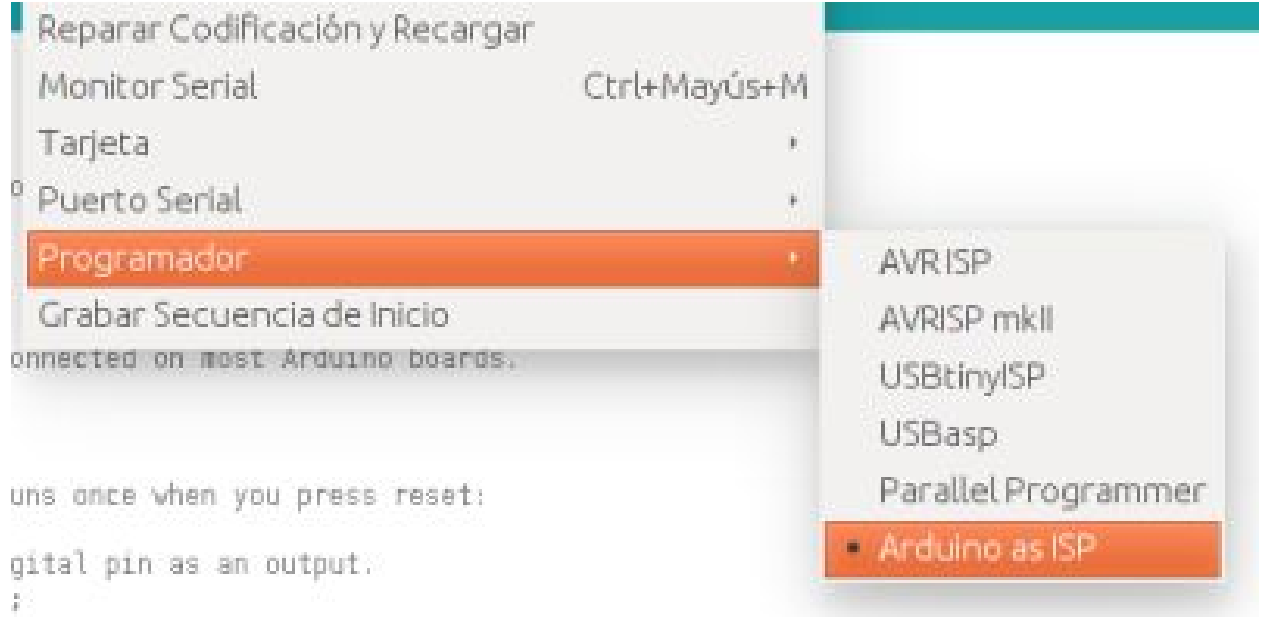
- La IDE de Arduino ya está lista para programar los attinys. Pero necesitamos un programador **hardware** también!. Vamos a usar un Arduino UNO como programador de attinys. Para eso tenemos que subirle un código especial. Tomá tu placa arduino o arduino-compatible y conectala a tu compu.
- En la pestaña **Archivo**, andá hasta **Ejemplos** y buscá **ArduinoISP**:



- Subí el sketch a tu placa Arduino Uno (damos por sentado que ya sabés cómo elegir la placa y el puerto de comunicación):



- Acabás de convertir a tu arduino en un programador de microcontroladores. Para usarlo como tal, sólo tenés que dar un paso más. Andá a la pestaña **Herramientas**, luego a **Programador** y elegí **Arduino as ISP**:



uns once when you press reset:

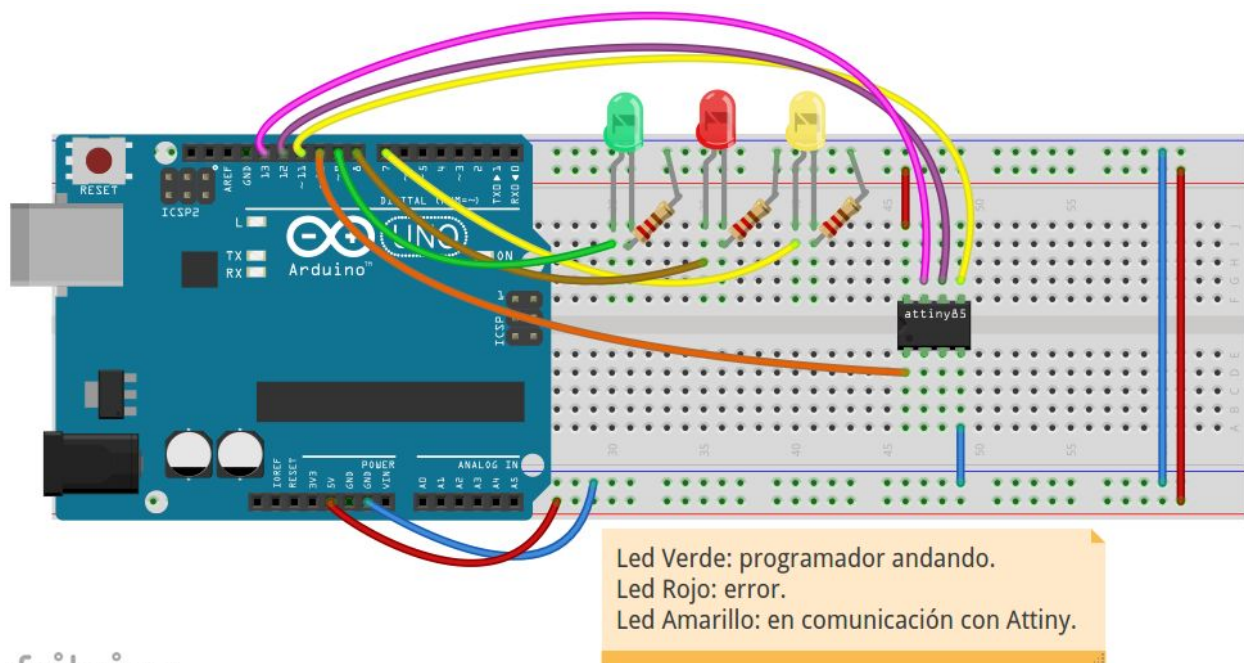
gital pin as an output:

;

ns over and over again forever:

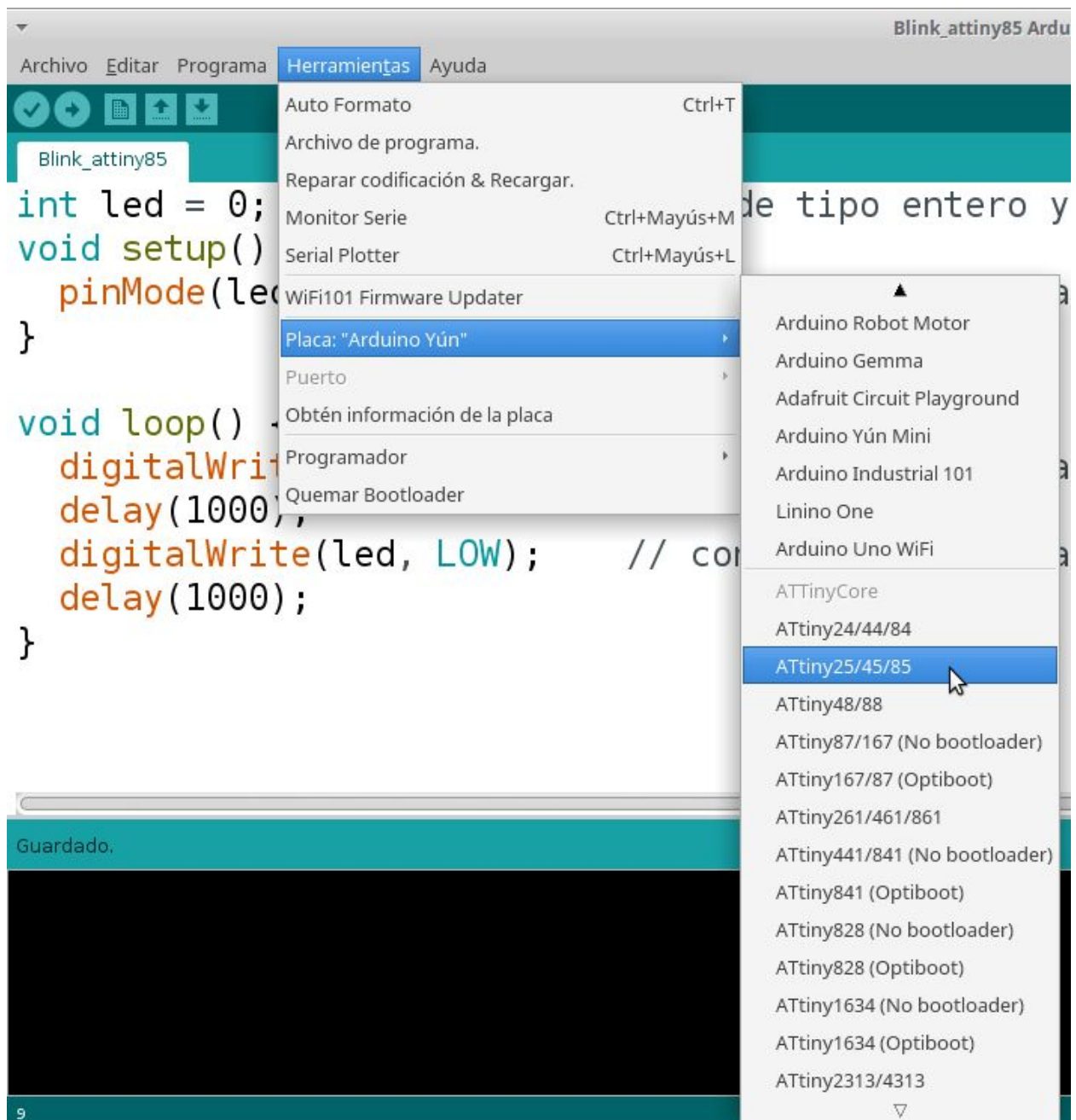
-

Tu Arduino ya está listo, ahora a conectar el Attiny85. Vas a necesitar, una protoboard y cables de protoboard. Opcionalmente podés agregar tres leds (uno verde, otro rojo y otro amarillo) y tres resis de 220ohm para tener una respuesta visual de lo que está sucediendo (el programador funciona igual si no conectás los leds). Conectá todo en la protoboard siguiendo la siguiente imagen o el esquemático (recordá que podés hacer doble click en las imágenes para verlas más grandes)

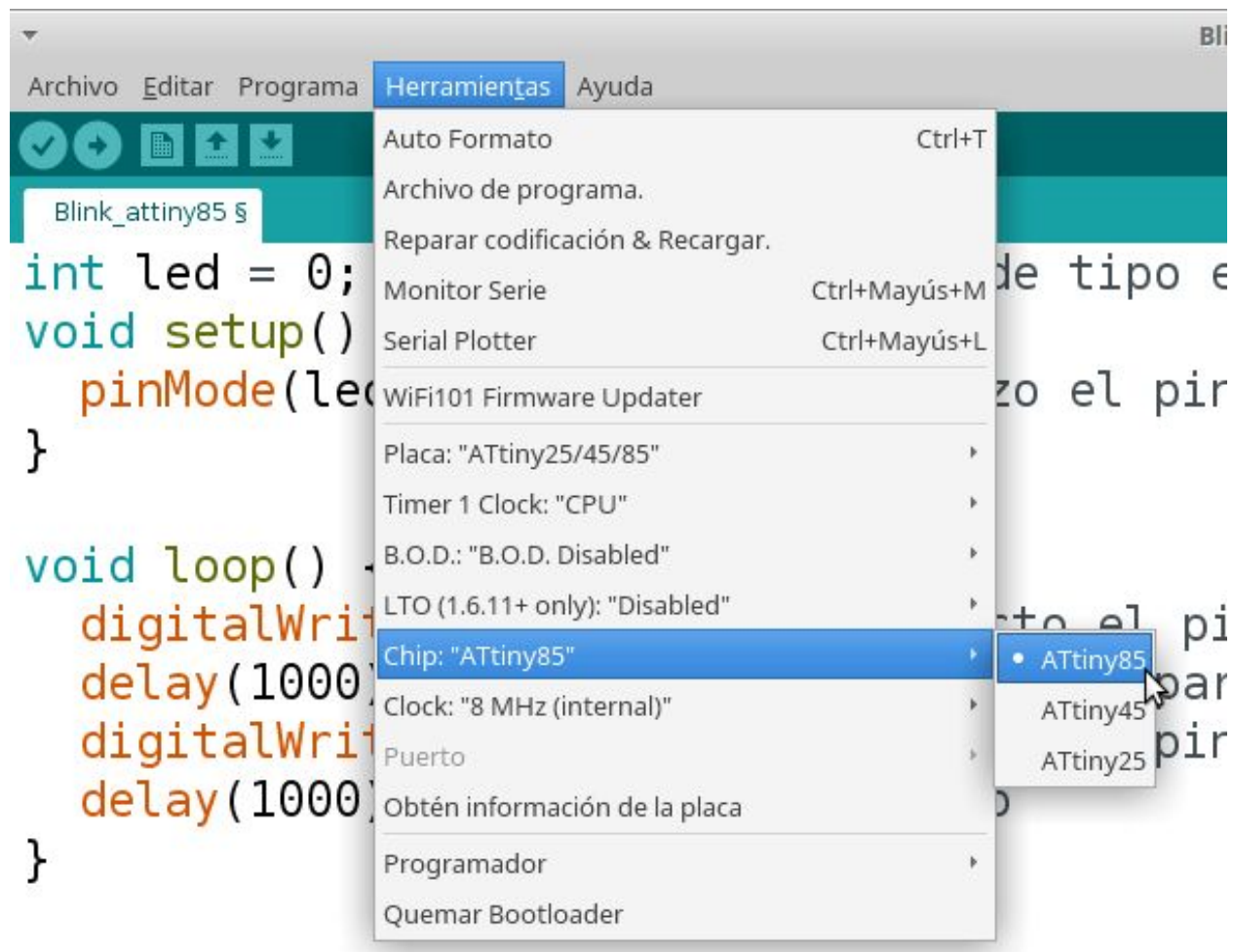




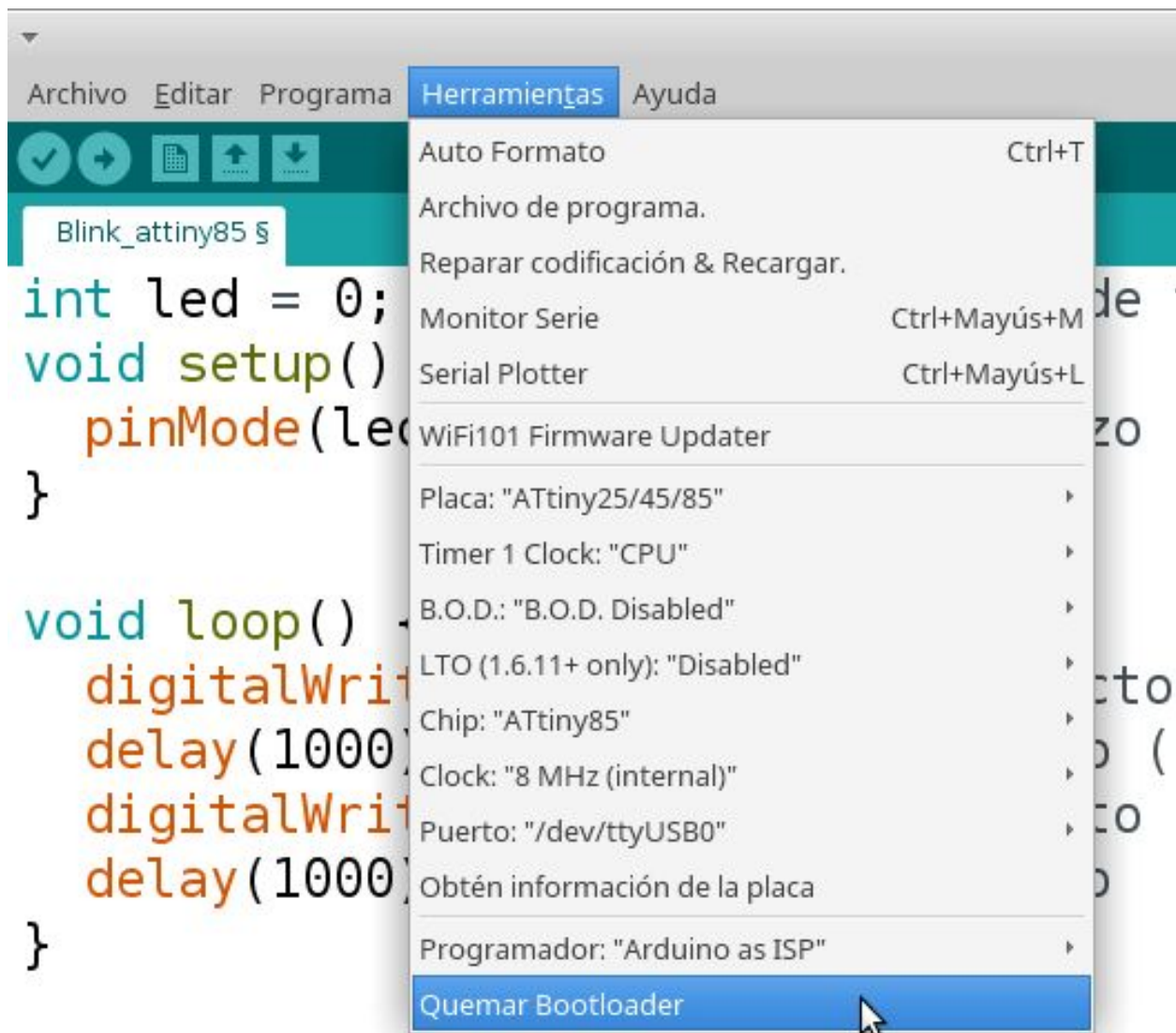
- Vamos a elegir el chip que estamos usando (en éste caso el attiny85). Andá a Herramientas/Placa y deslízate hacia abajo hasta que aparezca la opción attiny25/45/85. Seleccionala:



Si tu chip es efectivamente un Attiny85, no tenés que cambiar ninguna opción. Si en lugar de tener un attiny85, tenés un 25 o 45, vas a tener que cambiar el tipo de chip ingresando nuevamente a Herramientas.



La siguiente acción se realiza sólo una vez, al principio, con cada chip que compres. Es la carga del bootloader o sistema de arranque. Una vez quemado el bootloader, tu chip está listo para ser programado una y otra vez. Andá a Herramientas y presioná "Cargar bootloader (recordá tener seleccionado "Arduino as ISP" como programador).

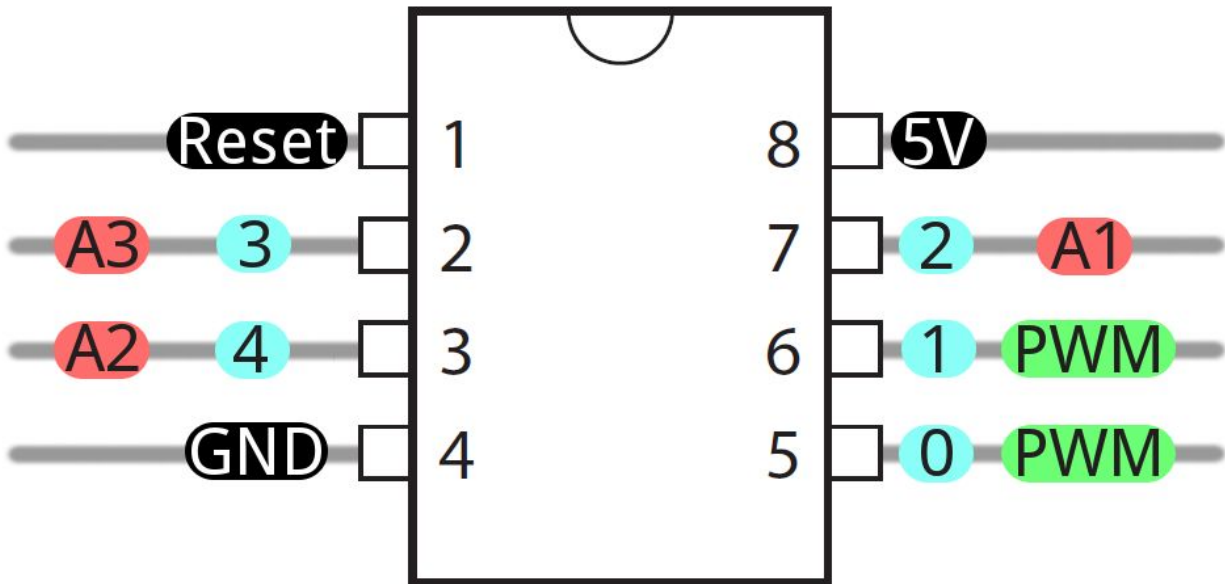


- Después de unos segundos, te aparecerá el siguiente cartel:



- Si obtuviste algún mensaje de error, tendrás que repasar conexiones y pasos anteriores.
- Finalmente está todo listo para cargar nuestro primer sketch. Vamos a repasar cómo se enumeran y declaran las entradas y salidas del attiny25/45/85 en el entorno de

programación de Arduino. En rojo verás el número con el que se declaran las entradas analógicas y en celeste las entradas/salidas digitales. En verde están indicadas las salidas digitales que permiten usar la función "analogWrite" (PWM).



- Para testear la comunicación entre arduino y attiny (en este caso un attiny85) , vamos a usar el clásico ejemplo Blink que hace parpadear un led a un ritmo constante de un segundo. Para ello usaremos un led (y una resistencia de 220 Ohms) conectado a la pata 5 (Digital 0) Acá tenés un [LINK](#) al archivo ya modificado para su uso en este tutorial. Copiá y pegá en un sketch vacío. Alternativamente podés abrir el ejemplo blink ubicado en Archivo/Ejemplo/Basics/Blink y reemplazar LED_BUILTIN por el valor 0 en todos los casos. Si copiaste y pegaste el texto del link, tu sketch se verá así:

```

Blink_attiny85 Arduino 1.8.2
int led = 0; // creo una variable de tipo entero y le asigno el valor 0
void setup() {
  pinMode(led, OUTPUT); //inicializo el pin digital 0 (pata 5 del chip) como salida
}

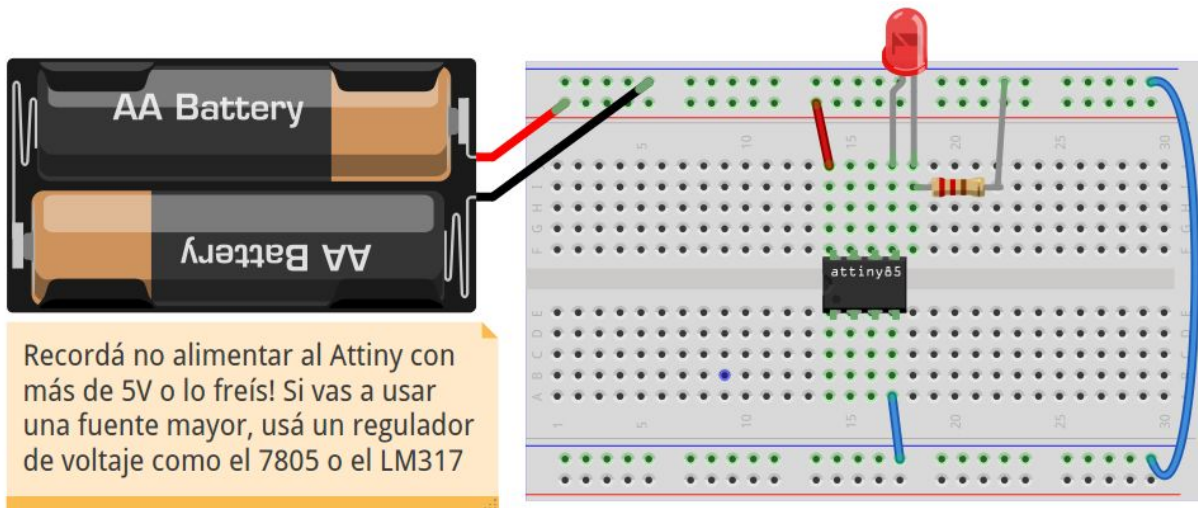
void loop() {
  digitalWrite(led, HIGH); // conecto el pin 0 a 5V
  delay(1000); // espero un segundo (el parámetro se expresa en milisegundos)
  digitalWrite(led, LOW); // conecto el pin 0 a tierra (0V/GND)
  delay(1000); // espero un segundo
}

```

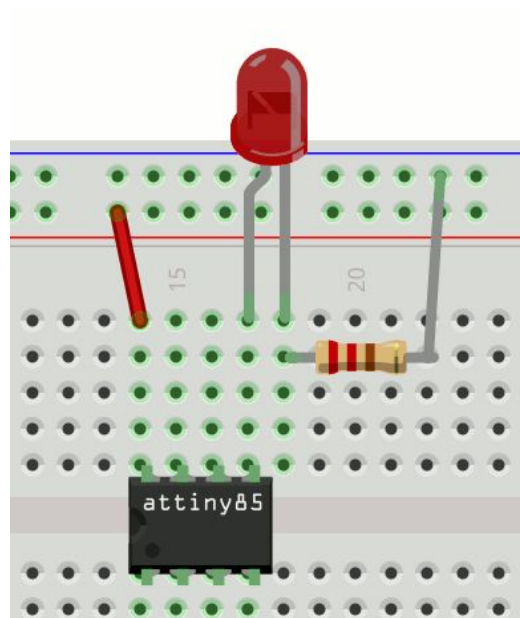
Guardado.

10 Arduino Yun en /dev/tty/USB0

- Subí el código al chip apretando el botón Cargar. Si colocaste los leds indicadores, deberías ver parpadear el led amarillo, lo que indica que está cargándose el código al attiny. Cuando veas el cartel de carga terminada, tu attiny ya estará programado! Ahora es momento de desarmar tu circuito anterior o buscar otra protoboard para ensamblar el circuito con el led (recordá no usar más de 5V!!!). El attiny anda fantástico con dos pilas AA (sumando 3V en total). Si no tenés ni pilas ni regulador, podés alimentar tu circuito con 5V y GND de tu arduino conectado a USB. El circuito es el siguiente:



- A esta altura, si todo está OK, deberías ver tu led parpadeando: Misión cumplida!



- Si algo falló en el camino, no te olvides de "triplechequear" instrucciones y conexiones. Ya sabés programar attinys, si querés experimentar con ejemplos más complejos, podés investigar proyectos del labo como el [Gamelín MINI](#) o [Neklops](#).