**El hardware**

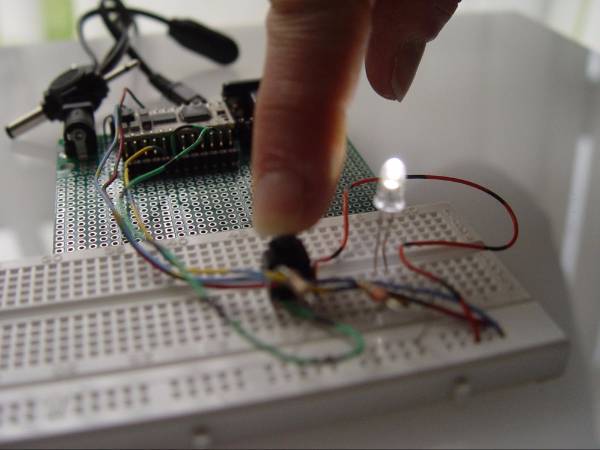
Desde el punto de vista de un diseñador, esta fue probablemente la parte más difícil de abordar. Pedí o compré tableros de evaluación de diferentes fabricantes de microcontroladores.

Aquí hay algunos momentos clave en el diseño de hardware para cableado.

**Prototipo 1**

El primer prototipo de Wiring utilizó el microcontrolador [Parallax](https://www.parallax.com/) Javelin Stamp. Era una opción natural ya que estaba programado en un subconjunto del lenguaje Java, que ya estaba siendo utilizado por Processing.

Problema: como se describe en el documento de tesis en la página 40, la compilación, vinculación y carga de los programas del usuario se basan en las herramientas patentadas de Parallax. Dado que Wiring se planeó como software de código abierto, el sello Javelin simplemente no era una opción viable.

[](https://arduinohistory.github.io/images/full/WiringPrototype1-JavelinStamp.jpg)

***Foto de Javelin Stamp utilizada para el primer prototipo de hardware de cableado.***

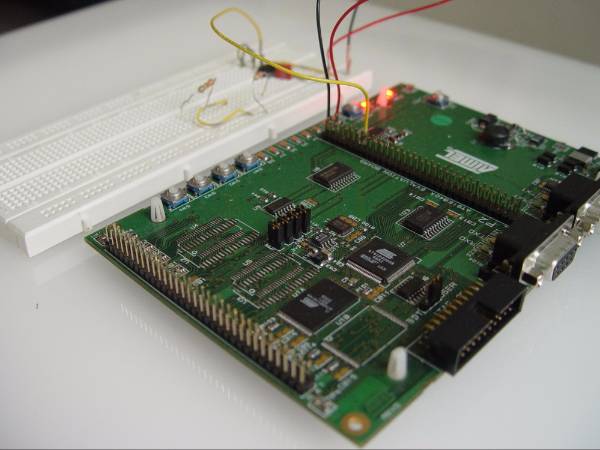
Para los siguientes prototipos, los microcontroladores se eligieron según la disponibilidad de herramientas de código abierto para compilar, vincular y cargar el código del usuario. Esto llevó a descartar la muy popular familia de microcontroladores PIC de Microchip, ya que, en ese momento (alrededor del año 2003), Microchip no tenía una cadena de herramientas de código abierto.

**Prototipo 2**

Para el segundo prototipo de hardware de cableado, se seleccionó el microcontrolador [AT91R40008](http://www.atmel.com/devices/R40008.aspx) basado en ARM [Atmel](http://www.atmel.com/) , que dio lugar a excelentes resultados. Se desarrollaron los primeros ejemplos de croquis y se iniciaron las pruebas de nomenclatura de comandos. Por ejemplo, solía ser el nombre del ahora ubicuo .pinWrite()digitalWrite()

El Atmel R40008 sirvió como banco de pruebas para la API de entrada / salida digital y la API de comunicaciones en serie, durante mi evaluación de sus capacidades. El Atmel R40008 era un microcontrolador muy potente, pero era demasiado complejo para un acercamiento práctico porque era casi imposible soldar a mano sobre una placa de circuito impreso.

Para obtener más información sobre este prototipo, consulte la página 42 en el documento de tesis.

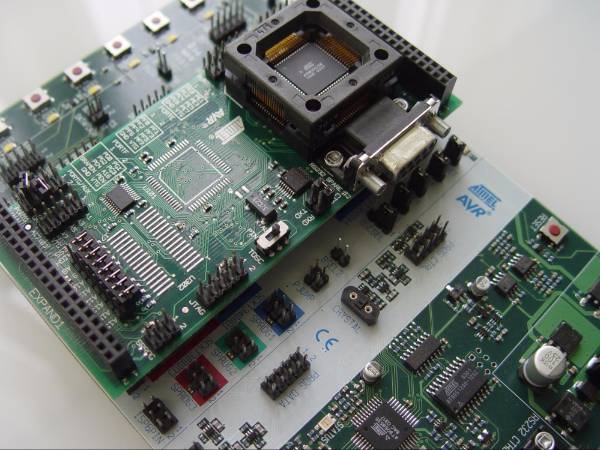
[](https://arduinohistory.github.io/images/full/WiringPrototype2-AtmelAT91R40008.jpg)

***Foto de Atmel AT91R40008 utilizada para el segundo prototipo de hardware de cableado.***

**Prototipo 3**

Los experimentos de prototipos anteriores llevaron al tercer prototipo, donde el microcontrolador se redujo a uno aún poderoso, pero con la posibilidad de modificarlo sin los requisitos de equipos especializados o periféricos adicionales a bordo.

Seleccioné el microcontrolador Atmel [ATmega128](http://www.atmel.com/devices/ATMEGA128.aspx) y compré una [placa de](http://www.atmel.com/tools/STK500.aspx)evaluación Atmel [STK500](http://www.atmel.com/tools/STK500.aspx) con un zócalo especial para el ATmega128.

[](https://arduinohistory.github.io/images/full/WiringPrototype3-AtmelATmega128.jpg)

***Foto de Atmel STK500 con la expansión ATmega128.***

Las pruebas con el STK500 fueron un éxito inmediato, por lo que compró un [MAVRIC](http://www.bdmicro.com/mavric-ib/) bordo de [BDMICRO](http://www.bdmicro.com/) con los ATmega128 soldadas. El trabajo de Brian Dean en sus tableros MAVRIC no tenía paralelo en ese momento, y su trabajo lo llevó a construir una herramienta de software para cargar fácilmente nuevos programas a su tablero. Todavía se usa hoy en el software Arduino, y se llama "avrdude".

Como los puertos COM tradicionales estaban desapareciendo de las computadoras, seleccioné el hardware [FTDI](http://www.ftdichip.com/) para la comunicación a través de un puerto USB en la computadora host. FTDI proporcionó controladores para Windows, Mac OS X y Linux que se requerían para que el entorno Wiring funcionara en todas las plataformas

El propósito de Arduino es el poder de ser usado por la mayoría de las personas, incluso sin tener un fuerte entrenamiento en programación y / o electrónica. Las principales ventajas de Arduino son: su costo, la facilidad de aprender la herramienta, un software de desarrollo sencillo, no se necesita un programador externo a la tarjeta y es un desarrollo de hardware.

Arduino también es una plataforma de desarrollo de Hardware libre, relativamente económica. Algunas tarjetas alternativas son la libertad de NXP, la tarjeta [Nucleo-l432kc](https://hetpro-store.com/TUTORIALES/nucleo-l432kc/) y [Raspberry](https://hetpro-store.com/TUTORIALES/que-es-raspberry/) . La [diferencia](https://hetpro-store.com/TUTORIALES/arduino-vs-raspberry-pi/) principal [entre Arduino y Raspberry](https://hetpro-store.com/TUTORIALES/arduino-vs-raspberry-pi/) es el tipo de dispositivo digital de procesamiento que tiene. Por ejemplo, el arduino es un microcontrolador, mientras que la frambuesa tiene un microprocesador. También es la diferencia entre un [Arduino vs Microcontrolador](https://hetpro-store.com/TUTORIALES/arduino-vs-microcontrolador/) es el lenguaje con el que se programa. Ambos se programan en C, pero el lenguaje Arduino es más sencillo y más rápido para aprender. Además, no se requiere programar los registros del microcontrolador.

## CARACTERÍSTICAS DE ARDUINO

Las principales características de un hijo de Arduino:

* Velocidad en Mhz.
* Tamaño de memoria RAM, FLASH y EEPROM.
* Cantidad de pinos de entrada / salida.
* Numero de pines analógicos.
* Cuantos puertos UART, [I2C](https://hetpro-store.com/TUTORIALES/i2c/) , SPI tiene.
* Tamaño de la tarjeta de evaluación.
* Bits del procesador.
* [Voltaje](https://hetpro-store.com/TUTORIALES/voltaje/) del procesador.

Por ejemplo, el Arduino 1 funciona aa 16Mhz, tiene 14 pines de entrada / salida digital y 6 canales de entrada para señales analógicas (adc). También tiene 1 puerto UART, 1 I2C, 1 SPI y su procesador es un ATmega328P de 8 bits. La memoria FLASH es de 32KB, una RAM de 2KB y de EEPROM tiene 1KB.

<https://hetpro-store.com/TUTORIALES/que-es-arduino/>