

# Arduino Hello World

Jorge Nunes

2014/03/08

## Abstract

What this article is all about.

## 1 Introduction

Mapping between Arduino and Atmega328 ports: <http://arduino.cc/en/Hacking/-PinMapping168#.UxrYyEEux6Y>

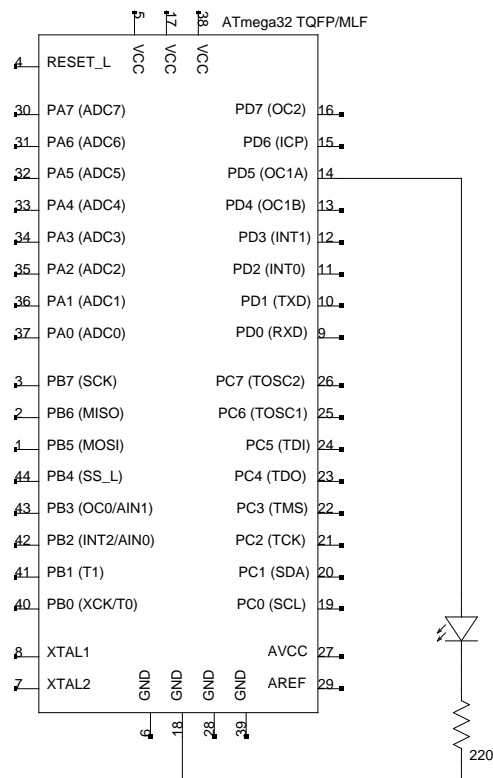


Figure 1: Circuit diagram showing how to connect the LED to the Arduino board.

And the actual real setup.

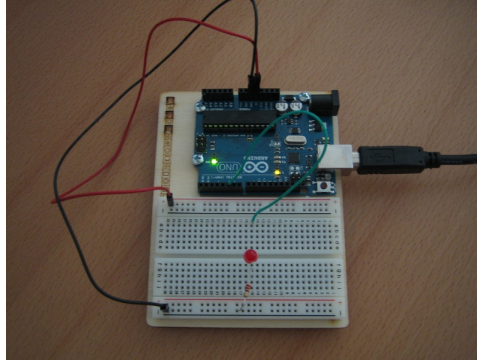


Figure 2: An Arduino Uno board with the circuit already set up.

## 2 Required tools

The following tools are needed for creating code to be run on the Arduino board:

- `avr-gcc` — GCC cross compiler for the Atmel processor.
- `avr-libc` — Minimal libC implementation.
- `avrdude` — Tool for uploading the code into the Arduino board.

```
1 yum install avr-gcc avr-libc avrdude
```

## 3 The code

This is the code we are going to compile and run in the Arduino board.

### Example code for the blinking LED.

```
1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 const int BLINK_DELAY_MS = 500;
5
6 int main (void) {
7     /* Set pin 5 of PORTD for output*/
8     DDRD |= _BV(DDD5);
9
10    while(1) {
11        /* Set pin 5 high to turn led on */
12        PORTD |= _BV(PORTD5);
```

```

13     _delay_ms (BLINK_DELAY_MS);
14
15     /* Set pin 5 low to turn led off */
16     PORTD &= ~_BV (PORTD5);
17     _delay_ms (BLINK_DELAY_MS);
18 }
19 return 0;
20 }

```

## 4 Compiling the code

To compile the code:

```

1 avr-gcc -Os -DF_CPU=16000000UL -mmcu=atmega328p -o blink ↵
    blink.c
2 avr-objcopy -O ihex -R .eeprom ./blink ./blink.hex

```

## 5 Uploading the compiled code to Arduino

Finally, to upload the code to the Arduino board:

```

1 avrdude -c arduino -p ATMEGA328P -P /dev/ttyACM0 -U flash ↵
    :w:./blink.hex:i

```

## 6 Colophon

The circuit diagram was designed with the `gschem` tool of the [gEDA project](#) software suite.

The video was edited and transcoded to MPEG-2 using the [FFmpeg](#) tools.