

Arduino Hello World

Jorge Nunes

March 8, 2014

Abstract

What this article is all about.

1 Introduction

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

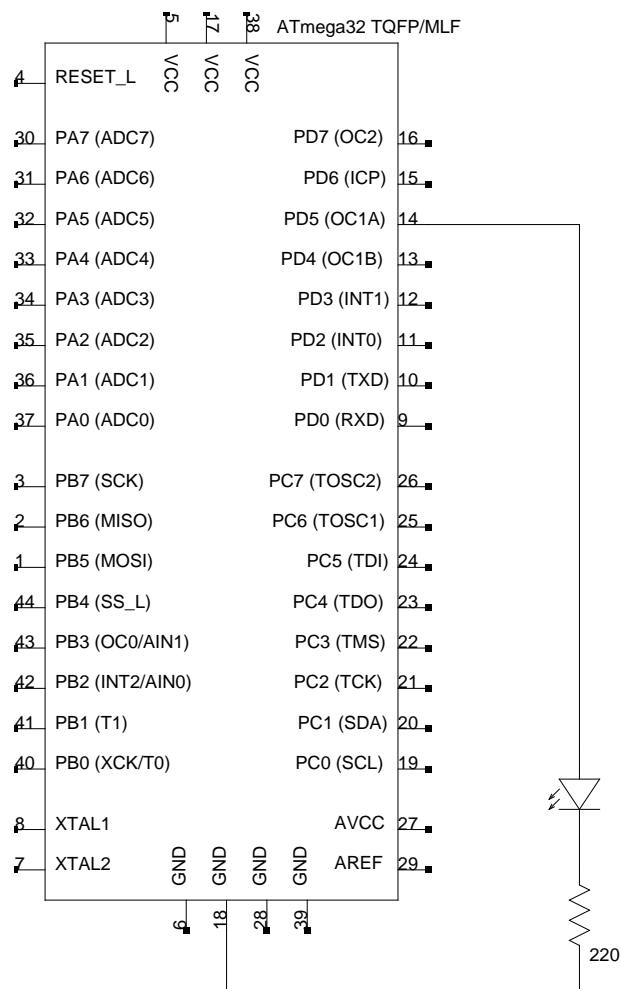


Figure 1: Circuit diagram showing how to connect the LED to the Arduino board.

And the actual real setup.

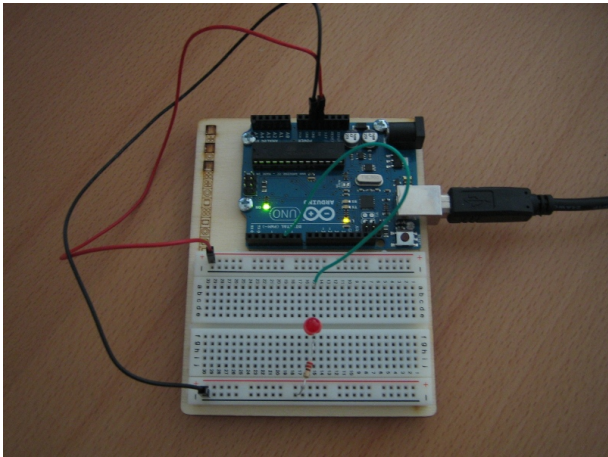


Figure 2: An Arduino Uno board with the circuit already set up.

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.

```
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.

```
#include <avr/io.h>
#include <util/delay.h>

const int BLINK_DELAY_MS = 500;

int main (void) {

    /* Set pin 5 of PORTD for output*/
    DDRD |= _BV(DDD5);

    while(1) {
        /* Set pin 5 high to turn led on */
        PORTD |= _BV(PORTD5);
        _delay_ms(BLINK_DELAY_MS);

        /* Set pin 5 low to turn led off */
        PORTD &= ~_BV(PORTD5);
        _delay_ms(BLINK_DELAY_MS);
    }

    return 0;
}
```

4 Compiling the code

To compile the code:

```
avr-gcc -Os -DF_CPU=16000000UL -mmcu=atmega328p -o blink blink.c
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:

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