



# A FIRST PROJECT

With the Toadstool *Mega328*



[www.crash-bang.com](http://www.crash-bang.com)

@crashbang\_proto

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

# A First Project

In four quick steps we'll get you up and running with your first project, the "hello world" of embedded systems: Blinking an LED.

## You Will Need

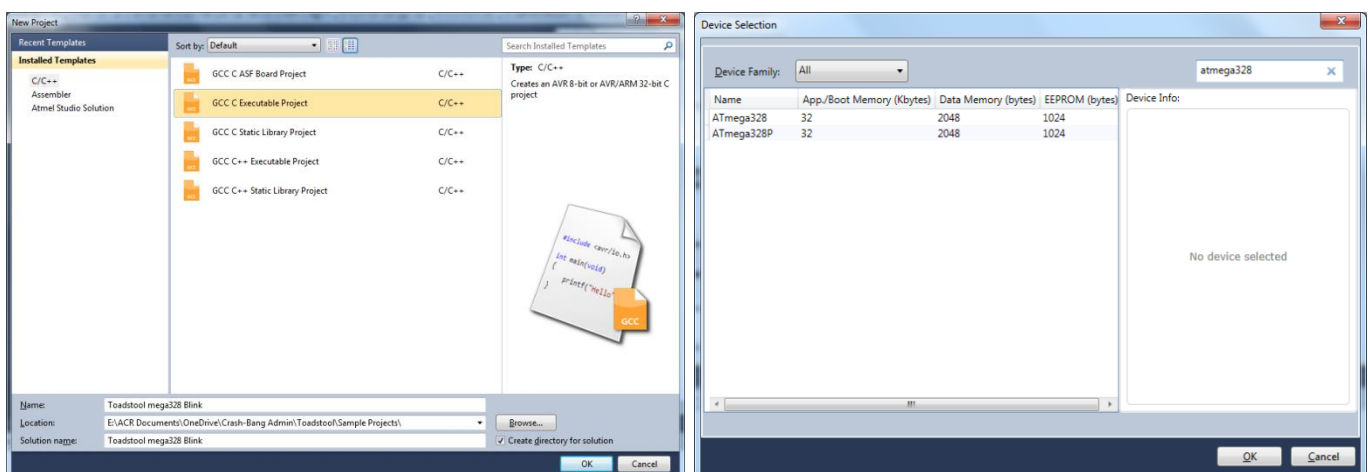
- ✓ Your Toadstool
- ✓ A Breadboard
- ✓ 1 LED
- ✓ 1 Resistor (a 330Ω works for most LEDs)
- ✓ 1 Jumper Wire
- ✓ Atmel Studio installed on your PC
- ✓ An ISP programmer

## 1 Setup your Breadboard

1. Use the Voltage Selection Jumper to choose whether to run your project at 5V or 3.3V. Either option will work for this project.
2. Place the Toadstool onto the breadboard, so that the semi-circular portion hangs off the edge of the breadboard – this will save you precious space on your breadboard for larger projects.
3. Connect the jumper wire from the **GND** pin on the Toadstool to the negative power rail on the breadboard
4. Place the LED on the breadboard: the anode in the same row as the Toadstool's pin **PB0**; the cathode in an empty row.
5. Place the resistor on the breadboard: one leg in the same row as the LED's cathode; the other leg to the negative power rail.

## 2 Create a new Project

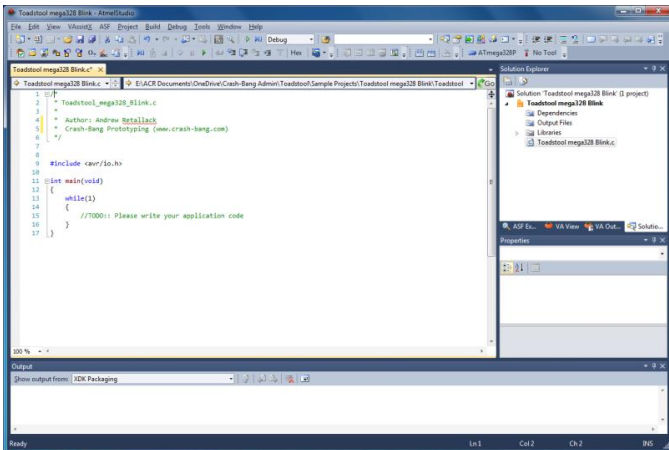
Once Atmel Studio is installed and fired up, create a new project (if you choose to use another IDE, then create a new project and move onto the next step):



1. From the **File** menu, select **New**, then **Project**
2. From the dialog that appears, select the **GCC C Executable Project**
3. Enter the project **Name** and choose the **Location** you'd like to save it. Then click **OK**
4. The final step is to choose the microcontroller (or device as Atmel Studio calls it). The **Toadstool Mega328** ships with an ATmega328P, so select that from the list (use the search field on the top right to filter results) and click **OK**.

### 3 Enter and Compile the Program Code

You now have a blank project.



Type the following code into Atmel Studio – you will see that a basic template has already been created for you, named **main.c**.

```

/*
 * Sample Program for Toadstool mega328
 * =====
 * Blinks an LED connected to PB0 every second.
 * Crash-Bang Prototyping www.crash-bang.com
 */

#define F_CPU 16000000UL //We are running at 16MHz. Used to time the delay

#include <avr/io.h>
#include <util/delay.h> //Contains the _delay_ms function called later

int main(void)
{
    //Configure the port the LED is connected to (PB0)
    DDRB |= (1<<DDB0); //Set PB0 as an output

    while(1) //Loop indefinitely
    {
        //LED on
        PORTB |= (1<<PORTB0); //Turn the LED on, by making PB0 go high
        _delay_ms(1000); //Delay 1 second (1000 milliseconds)

        //LED off
        PORTB &= ~(1<<PORTB0); //Turn the LED off, by making PB0 go low
        _delay_ms(1000); //Delay 1 second (1000 milliseconds)
    }
}
  
```

If you aren't using Atmel Studio, this code will still work as long as you're using the GCC toolchain.

Hit **F7** to compile and build the code (or choose **Build Solution** from the **Build** menu). The Output pane at the bottom of the screen should show that the build was successful, with the message:

```
Build succeeded.  
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

## 4 Flash the Program on the Toadstool

The last step is to get the compiled code onto the Toadstool's microcontroller. In this example, an Atmel-ICE Basic<sup>1</sup> programmer/debugger is used – this is a good option for entry-level programming and debugging. If you're using a USBTiny or USBAsp, then you'll need to follow online instructions to configure Atmel Studio and flash the program.<sup>2</sup>

1. Connect the programmer that you'll be using to the ICSP header on the Toadstool. Plug it into your computer as well.
2. Some programmers are able to power your board – if yours doesn't allow this, then you'll need to power your board from a battery connected to the 2.1mm jack. The battery voltage should be at least 1.5V more than the voltage you've chosen with the Voltage Selection jumper.
3. From the **Project** menu, click the **...properties** option and then the **Tool** tab. Here you need to choose the programmer you're using. Note that it must be connected in order to be selectable.
4. To upload the program, choose **Start without Debugging** from the **Debug** menu (or press **Ctrl-Alt-F5**)

The LED on your breadboard should start blinking. You can now disconnect the Programmer if you like, and power the project from a battery – allowing it to operate without being tethered to your computer.

## Where to From Here?

Congratulations on getting your first Toadstool project up and running. There are more resources online to take you further: refer to our website<sup>3</sup> to access these.

---

<sup>1</sup> **Atmel Ice product page:** <http://www.atmel.com/tools/atatmel-ice.aspx>

<sup>2</sup> **Using a USBTiny programmer with Atmel Studio:** <http://www.crash-bang.com/getting-started-atmel-studio/>

<sup>3</sup> **Toadstool on the Internet:** [www.crash-bang.com/toadstool](http://www.crash-bang.com/toadstool)