

# Using Button Switches

Connect and read input from button switches.

Site: [iCODE](#)

Course: Machine Science Guides (Arduino Version)

Book: Using Button Switches

Printed by: Ivan Rudnicki

Date: Wednesday, July 30, 2014, 03:30 PM

# Contents

---

[About Button Switches](#)

[Adding a Button Switch](#)

[Controlling the LED](#)

# About Button Switches

---

A button is a switch that completes or interrupts an electric circuit, much like a switch for a household appliance. Buttons are commonly used to provide input to electronic devices, such as the texting device and game controller shown in Figure 1.



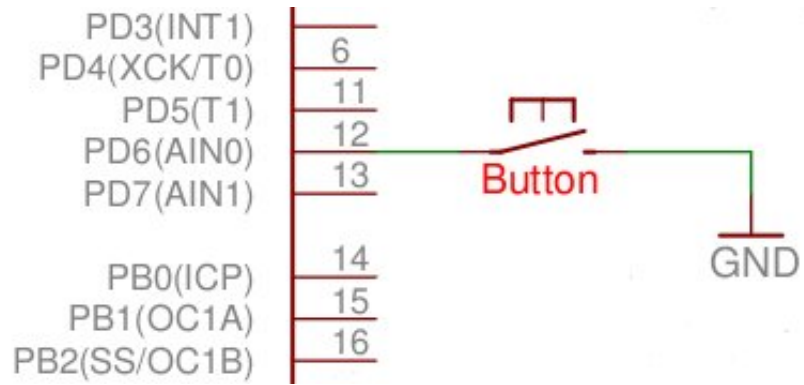
**Figure 1. Electronic devices with buttons.**

The button switches included with the kit are called *momentary* switches, because they make an electrical connection when pressed and break the connection when released.

# Adding a Button Switch

---

Add a button switch that connects Port D6 to ground, as shown in Figure 2.



**Figure 2. Adding a button switch to the ATmega Board.**

# Controlling the LED

---

A physical resistor is not needed when adding the button switch, because the ATmega168 has internal pull-up resistors available on most pins. These can be used to hold the value of an input pin high, just as you would with a physical resistor on the board. The functions `pullup_on` and `pullup_off` are used to activate and deactivate a pin's internal pull-up resistor, as shown below:

```
pullup_on(PORT_D6); //Turn on pull-up resistor for Port D6
pullup_off(PORT_D6); //Turn off pull-up resistor for Port D6
```

With the following code, you can make the button control the LED, using the internal pull-up resistor on Port D6.

1. Rename your code file `button.c`.
2. Type or paste the following code:

```
#include <mxapi.h>

int main(void)
{
    input_pin(PORT_D6); //Set up Port D6 as an input
    output_pin(PORT_B0); //Set up Port B0 as an output
    pullup_on(PORT_D6); //Activate pull-up resistor on Port D6
    while(1==1) //Remain in an infinite loop
    {
        if(pin_value(PORT_D6)==0) //If button is pressed
        {
            high_pin(PORT_B0); //Light LED
        }
        else if(pin_value(PORT_D6)==1) //If button is not pressed
        {
            low_pin(PORT_B0); //Turn off LED
        }
    }
}
```

3. Compile and test your new code.



### Programming Challenge

Change your code so that the LED turns off when the button is pressed. Then, modify your code so that the LED turns on when you first press the button and turns off when you press it again.