**Digital Input Functions**

Digital input is accomplished in two steps. You must configure the digital pin as an input, and you must read the state of the digital pin.

**pinMode()**

As in the previous tutorial for [Digital Output](#), we need to call `pinMode()` to specify the pin direction. For the case of digital input, we pass the value `INPUT`.

Example code:

```c
pinMode(2, INPUT);
```

**digitalRead()**

Digital input on an Arduino is done using the `digitalRead()` function. `digitalRead()` takes a single parameter, the pin number, and returns an `int` taking the value either `HIGH` or `LOW`.

Example code:

```c
int iDigitalValue = digitalRead(2);
```

**digitalRead() Example**

A simple example of digital input uses a momentary pushbutton switch, generally called a button. Most hobby buttons or switches can be used for this example. Simply connect one side of the button or switch to Vcc (5V for Arduino Megas and Unos). Connect the other side to the digital pin being used (2 in this example). Finally, connect a 1MΩ resistor between the pin being used and GND. This is called a pull-down resistor, and has the effect of pulling the voltage low when the circuit is open (when the button is not pressed).
These schematics are attached and can be viewed with Fritzing.

**Blinking Light with Button Arduino Sketch**

```c
#define BUTTON_PIN 2
#define LED_PIN 13

void setup()
{
    // Initialize LED pin as an output
    pinMode(LED_PIN, OUTPUT);
    // Initialize Button pin as an input
    pinMode(BUTTON_PIN, INPUT);
}

void loop()
{
    // Read button value
    int iDigitalValue = digitalRead(BUTTON_PIN);
    if (iDigitalValue == HIGH)
    {
        // Turn LED on
        digitalWrite(LED_PIN, HIGH);
    }
    else
    {
        // Turn LED off
        digitalWrite(LED_PIN, LOW);
    }
}
```

digitalRead() Example
// Wait 10ms
delay( 10 );
}
This code can be downloaded here.