Please ask about problems and questions regarding this tutorial on answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

# BlinkM Tutorial

**Description:** Control a BlinkM through ROS using an Arduino and rosserial

**Tutorial Level:** ADVANCED

**Next Tutorial:** Arduino Oscilloscope (/rosserial_arduino/Tutorials/Arduino%20Oscilloscope)

| electric | fuerte | groovy | hydro | indigo | jade | kinetic |
|---|---|---|---|---|---|---|

**Contents**



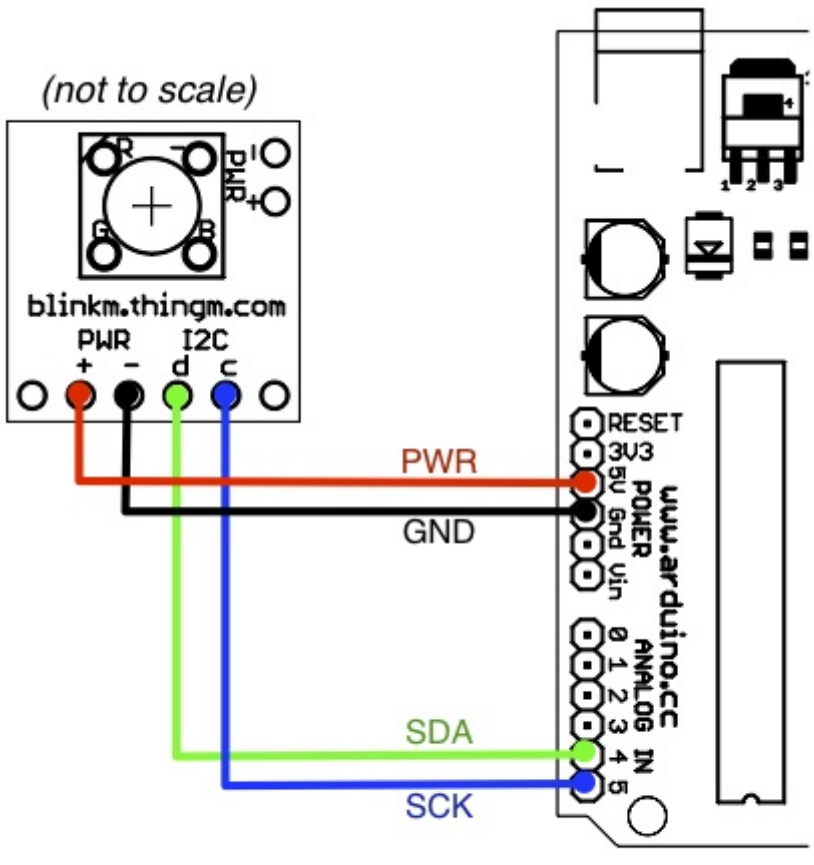
One thing every robot needs is a good indicator light. If you are looking for something more fancy or just plane brighter than just a single tiny LED, the BlinkM is a good choice for your robotics project. The BlinkM (http://thingm.com/products/blinkm) is a I2C controlled multi-colored LED which can change colors and run lighting scripts. In this tutorial, we are going to use the default scripts of a BlinkM to create a multicolored blinking or solid indicator light.


**BlinkM**

For this tutorial, you will be using the example provided with ros_lib. In your Arduino IDE, go to File>Examples>ros_lib>BlinkM. The code can also be found in rosserial_arduino/src/ros_lib/examples.

For a full reference and getting started guide for the BlinkM, see the BlinkM datasheet (http://docs.google.com/viewer?url=http%3A%2F%2Fthingm.com%2Ffileadmin%2Fthingm%2Fdownloads%2FBlinkM_datasheet.pdf).

## 1. Hardware




**BlinkM / MinM**

Diagram from the BlinkM datasheet (http://docs.google.com/viewer?url=http%3A%2F%2Fthingm.com%2Ffileadmin%2Fthingm%2Fdownloads%2FBlinkM_datasheet.pdf)

The hardware for this tutorial is relatively simple. All that is neccesary is a BlinkM and an Arduino. The BlinkM needs to be connected to 5V, GND, and the Arduino's I2C connections (SDA and SCL). You can purchase both a BlinkM (http://www.sparkfun.com/products/8579) and an Arduino (http://www.sparkfun.com/products/9950) from Sparkfun.

# 2. Code

The ROS serial integration code for the blinkm is below. In this arduino sketch, the node subscribes to a std_msgs/String (http://docs.ros.org/api/std_msgs/html/msg/String.html) on the blinkm topic. In the callback, the the node parses the command to determine the LED's color and if it should be blinking. The LED can be red (r), blue (b), magenta(m), green(g), white(w), cyan(c), and yellow(y). If the command begins with an 'S' or 's' the LED is a solid color. Otherwise, the LED blinks that color.

Toggle line numbers

```
/*
*  RosSerial BlinkM Example
*  This program shows how to control a blinkm
*  from an arduino using RosSerial
*/

#include "WProgram.h" //include the Arduino library
#include <stdlib.h>


#include <ros.h>
#include <std_msgs/String.h>


//include Wire/ twi for the BlinkM
#include <Wire.h>
extern "C" {
#include "utility/twi.h"
}

#include "BlinkM_funcs.h"
const byte blinkm_addr = 0x09; //default blinkm address


void setLED( bool solid,  char color)
{

        if (solid)
        {
           switch (color)
                {

                case 'w':  // white
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0xff,0xff,0xff);
                        break;

                case 'r': //RED
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0xff,0,0);
                        break;

                case 'g':// Green
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0,0xff,0);
                        break;

                case 'b':// Blue
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0,0,0xff);
                        break;

                case 'c':// Cyan
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0,0xff,0xff);
                        break;

                case 'm': // Magenta
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0xff,0,0xff);
                        break;

                case 'y': // yellow
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0xff,0xff,0);
                        break;

                default: // Black
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_fadeToRGB( blinkm_addr, 0,0,0);
                        break;
                }
        }


        else
        {
```

```
                switch (color)
                {
                case 'r':  // Blink Red
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 3,0,0 );
                        break;
                case 'w':  // Blink white
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 2,0,0 );
                        break;
                case 'g':  // Blink Green
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 4,0,0 );
                        break;

                case 'b': // Blink Blue
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 5,0,0 );
                        break;

                case 'c': //Blink Cyan
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 6,0,0 );
                        break;

                case 'm': //Blink Magenta
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 7,0,0 );
                        break;

                case 'y': //Blink Yellow
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 8,0,0 );
                        break;

                default: //OFF
                        BlinkM_stopScript( blinkm_addr );
                        BlinkM_playScript( blinkm_addr, 9,0,0 );
                        break;
                }

        }
}

void light_cb( const std_msgs::String& light_cmd){
        bool solid =false;
        char color;
        if (strlen( (const char* ) light_cmd.data) ==2 ){
          solid  = (light_cmd.data[0] == 'S') || (light_cmd.data[0] == 's');
          color = light_cmd.data[1];
        }
        else{
          solid=  false;
          color = light_cmd.data[0];
        }

        setLED(solid, color);
}



ros::NodeHandle  nh;
ros::Subscriber<std_msgs::String> sub("blinkm" , light_cb);


void setup()
{

    pinMode(13, OUTPUT); //set up the LED

        BlinkM_beginWithPower();
        delay(100);
        BlinkM_stopScript(blinkm_addr);  // turn off startup script
        setLED(false, 0); //turn off the led

        nh.initNode();
        nh.subscribe(sub);
```

```
155
156 }
157
158 void loop()
159 {
160  nh.spinOnce();
161  delay(1);
162 }
```

Another key feature of this script that should be noted is that the I2C address of the BlinkM is set to the default BlinkM I2C. If yours has been reprogrammed, or if you want to control multiple BlinkMs (/BlinkMs), you will need to change this address.

Toggle line numbers

```
21 #include "BlinkM_funcs.h"
22
```

# 3. Testing

Program you Arduino with the BlinkM sketch. Open the blinkm sketch from the package://roserial_arduino_tutorials/sketches/BlinkM folder and program your Arduino.

Start up the roscore in a new termianl

```
roscore
```

Launch the rosserial_python serial_node. Make sure to choose the right serial port.

Now, look at the red blinking light!

```
rostopic pub blinkm std_msgs/String "br"
```

Blue light!

```
rostopic pub blinkm std_msgs/String "sb"
```



Wiki: rosserial_arduino/Tutorials/BlinkM (last edited 2014-09-22 16:30:23 by AustinHendrix (/AustinHendrix))