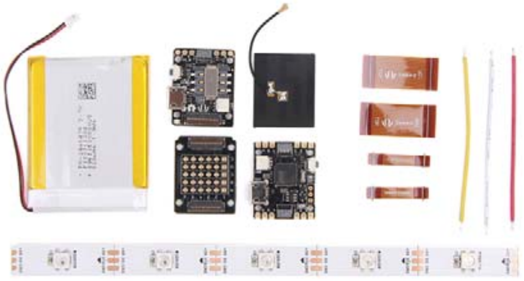
RePhone Lumi Kit

SKU 113060002



- Slim and small modulesPowered by Xadow GSM+BLE
- Programmable LED Driver
- Various rich-featured add-on modules for function extension

Description

RePhone Lumi Kit is specially designed for LED lovers - embedding the Xadow GSM+BLE module into your awesome lighting device and control the light via SMS or Bluetooth!

TheXadow Duino is specially designed for building up awesome Lighting Devices using the most popular RGB LED WS2812b. Based on the MCU ATmega32u4 and the built-in micro USB, Xadow Duino allows you to either control the RGB LEDs directly from the software in RePhone Kit Create, or program the board with Arduino IDE to make it work dependently.

FEATURES

- • Slim and small modules
- • Powered by Xadow GSM+BLE
- • Programmable LED Driver
- • Various rich-featured add-on modules for function extension
- • Stackable, chainable and sewable with other Xadow Modules

SPECIFICATION

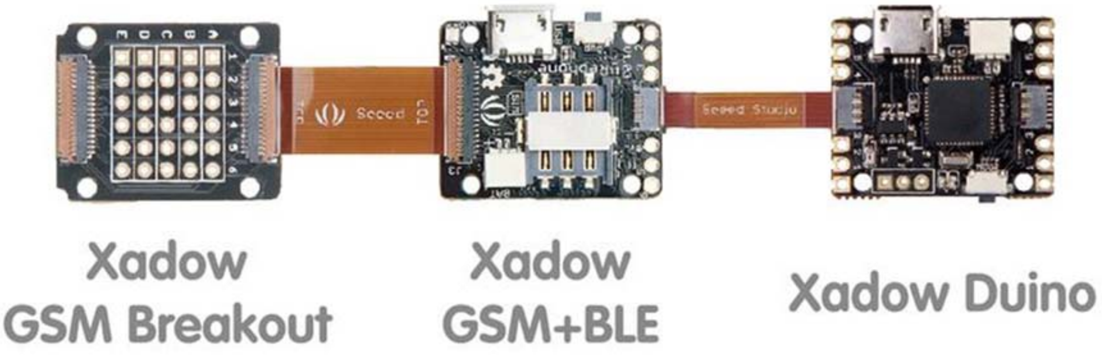
- • Core Module: Xadow GSM+BLE
- • Microcontroller: MT2502
- • MCU Core: 32-bit ARM7EJ-STM RISC processor
- • RAM: 4MB
- • Flash Memory: 16MB
- • Power Supply: 3.3 – 4.2V
- • Power Consumption: 20mW/30mW/52mW @ standby(no radio)/standby(GSM)/standby(BT)
- • Quad-band: 850/900/1800/1900 MHz
- • GPRS: Class 12 modem
- • Clock Speed: 260 MHz
- • Connector: 35 PIN Connector & 11 PIN Connector for Xadow, SMD Female JST battery connector
- • Interfaces: External ports for LCD, camera, I2C, SPI, UART, and GPIOs etc.
- • Dimension: 25.37mm X 20.30mm / 1" X 0.8"

PART LIST

- • Xadow GSM+BLE * 1
- • Xadow Duino * 1
- • Xadow GSM Breakout * 1
- • FPC Cable 35 PIN * 2
- • FPC Cable 11 PIN * 2
- • Battery 520mAh * 1
- • User Guide * 1

RePhone Lumi Kit

Introduction



RePhone Lumi Kit is specially designed for LED lovers - embedding the Xadow GSM+BLE module into your awesome lighting device and control the light via SMS or Bluetooth!

Specification

Xadow GSM+BLE

Microcontroller	MT2502
MCU Core	32-bit ARM7EJ-STM RISC processor
RAM	4 MB
Flash Memory	16 MB
Power Supply	3.3 ~ 4.2V(no SIM)/3.5 ~ 4.2V(with SIM)
Power Consumption	20mW/30mW/52mW @ standby(no radio)/standby(GSM)/standby(BT)
Quad-band	850/900/1800/1900 MHz
GPRS	Class 12 modem
Clock Speed	260 MHz
Connectors	35 PIN Connector & 11 PIN Connector for Xadow Modules; JST 1.0 Connector for the battery
Interfaces	LCD, Audio, I2C, SPI, UART, and GPIOs etc
Dimensions	25.37mm × 20.30mm / 1” × 0.8”

Xadow Duino

Microcontroller	ATmega32u4
Power Supply	5V via USB, current draw is 1.5A at Max 3.3 ~ 4.2V via battery socket, current draw is 1.5A at Max 3.3 ~ 6V via breakout pins, current draw is 500 mA at Max 3.3 ~ 4.2 V via Xadow GSM+BLE, current draw is 500 mA at Max

Charging Current	500 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
Connector	2 × 11 PIN Xadow Connector Bonding pads for RGB LED WS2812b
Interfaces	Interface with Xadow GSM+BLE through I2C (7-bit address 0x23)
Dimensions	25.37mm × 20.30mm / 1”× 0.8”

Xadow GSM Breakout

Dimensions	25.37mm × 20.30mm / 1” × 0.8”

Module Connection

The Xadow GSM+BLE integrates a Xadow 36 Pin connector and a Xadow 11 Pin connector. As the image shows here, you can connect the Xadow GSM+BLE to Xadow GSM breakout with a Xadow 36 Pin cable, but do bear in mind that the cable has to be inserted **firmly** and **flat**, and most importantly, **keep the ‘TOP’ side upwards**. The Xadow 11 Pin connector have much more flexible design, you can connect the Xadow Duino and the mainboard with a Xadow 11 Pin ribbon cable in whatever way you want, again just make sure cable is firmly inserted.

Control the Light with SMS!

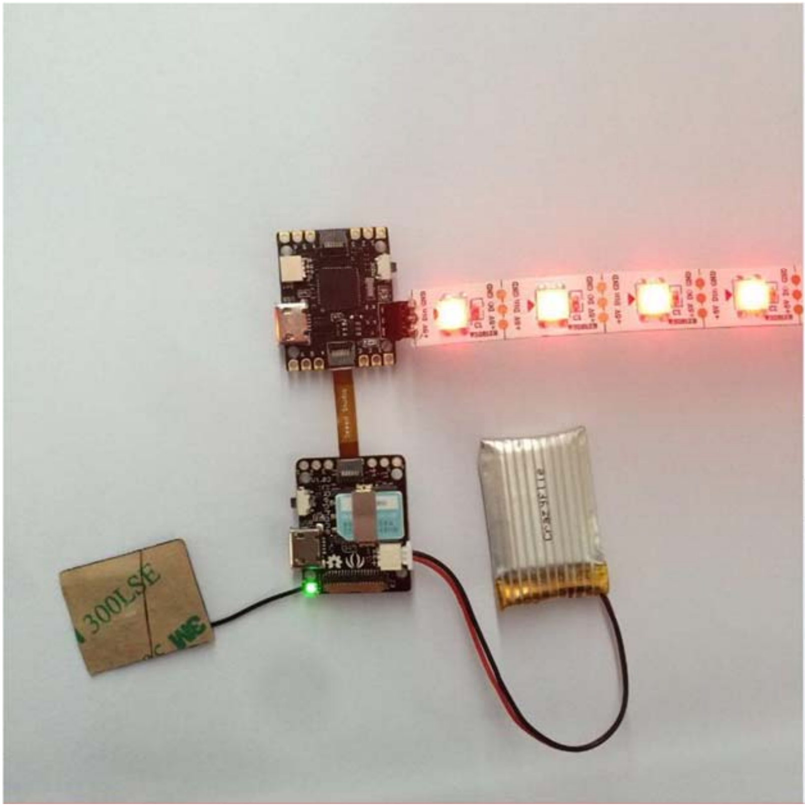
Here is an example of controlling you RGB LEDs by sending a message to your RePhone.

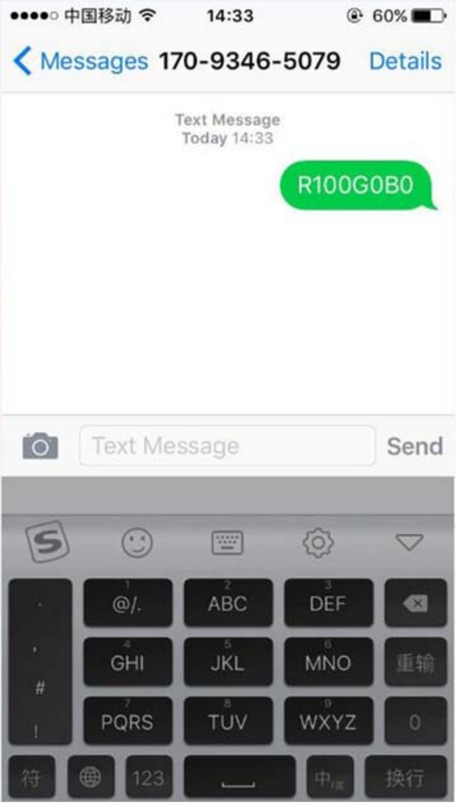
Firstly you need to follow the instructions in the **Arduino IDE for RePhone Kit** to download the Arduino IDE for RePhone Kit as well as install the drivers. Then copy the following code to your Arduino IDE for RePhone. Choose the **board as "RePhone"** and **download it with the "MTK USB Debug Port"**.

```
#include <LCheckSIM.h>
#include <LGSM.h>
#include <LWs2812.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

//SMS command 0 : "OFF" - off display.
//SMS command 1 : "R100G0B0" - red color is 100, green color is 0, blue coler is 0.
char num[20];
char buf_contex[200];


void setup()
{
    Serial.begin(115200);
    Serial.print("Ws2812 test.\r\n");
}

void loop()
{
    unsigned long rgb_color;
    if(LSMS.available())
    {
        LSMS.remoteNumber(num, 20);
        LSMS.remoteContent(buf_contex, 50);

        Serial.println("Get new sms: ");

        Serial.print("Number:");
        Serial.println(num);

        Serial.print("Content:");
        Serial.println(buf_contex);

        if(strcmp("OFF", buf_contex) == 0)
        {
            Serial.print("RGB off.\r\n");
            LWs2812.monochrome(5, 0, 1);
        }
        else
        {
            if(StingFind(buf_contex, "R", 1))
            {
                if(StingFind(buf_contex, "G", 1))
                {
                    if(StingFind(buf_contex, "B", 1))
                    {
                        Serial.print("RGB flag is find.\r\n");
                        char color_r[4] = {0};
                        char color_g[4] = {0};
                        char color_b[4] = {0};

                        sscanf(buf_contex + 1,"%[^G]", color_r);
                        sscanf(buf_contex,"%*[^G]G%[^B]", color_g);
                        sscanf(buf_contex,"%*[^B]B%[^\0]", color_b);

                        rgb_color = atol(color_r)*256*256 + atol(color_g)*256 + atol(color_b);

                        Serial.print("rgb color is ");
                        Serial.println(rgb_color);
                        delay(100);
                        LWs2812.monochrome(5, rgb_color, 0);
                        delay(100);
                        LWs2812.monochrome(5, rgb_color, 0);
                    }
                }
            }
        }
    }
}

int StingFind(char* string, char* find, int number)
{
    char* pos = string;
    char* p = string;
    int count = 0;

    while(number > 0)
    {
        pos = strstr(p, find);
        if(pos == 0)return -1;

        while(p <= pos)
        {
            if(*p > 0x80 || *p < 0)
            {
                p ++;
            }
            p ++;
            count ++;
        }
        number --;
    }
    return count;
}
```

In this code we set the SMS command as: ··· **“OFF”** - off display. **“R100G0B0”** - red color is 100, green color is 0, blue coler is 0. ···

So if we send **“R100G0B0”** to the RePhone, the LEDs will become red as showing below:





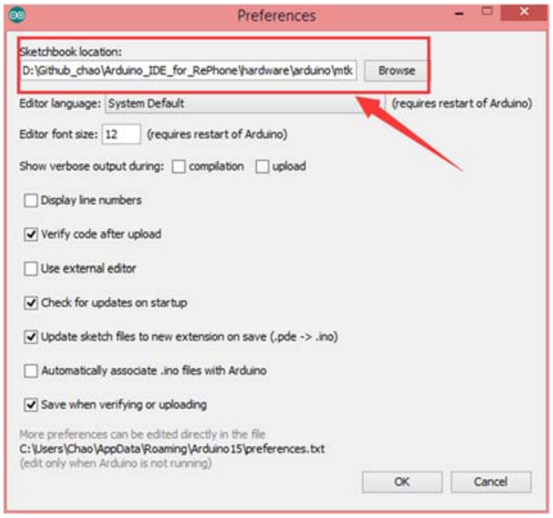
Text Message
Today 14:33

R100G0B0

More Example Code

Playing with some example code would a good start if you are heading into some even more interesting project.

1. Simply setup your **sketchbook location** in the **preference** as where you install the Arduino IDE for RePhone, then you'll be able to see the example codes.



Sketchbook location:
D:\Github_chao\Arduino_IDE_for_RePhone\hardware\arduino\mtk

Editor language: System Default (requires restart of Arduino)

Editor font size: 12 (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

□ Display line numbers

☑ Verify code after upload

□ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

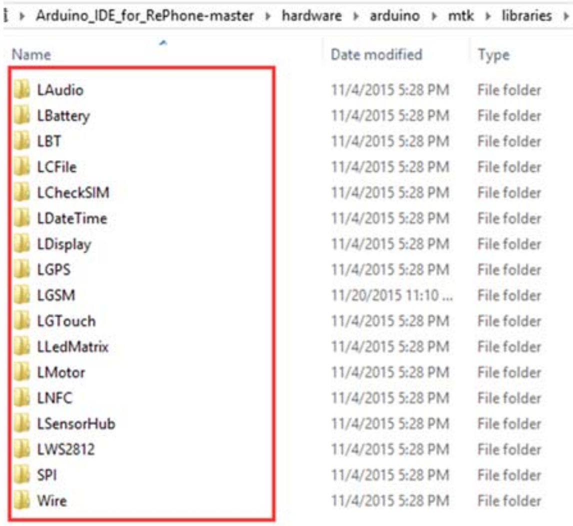
□ Automatically associate .ino files with Arduino

☑ Save when verifying or uploading

More preferences can be edited directly in the file
C:\Users\Chao\AppData\Roaming\Arduino15\preferences.txt
(edit only when Arduino is not running)

1. You may find the Arduino example coded manually for all the RePhone modules in the following path:

Aruino_IDE_for_RePhone-master\hardware\arduino\mtk\libraries



› Arduino_IDE_for_RePhone-master › hardware › arduino › mtk › libraries ›

Name	Date modified	Type
LAudio	11/4/2015 5:28 PM	File folder
LBattery	11/4/2015 5:28 PM	File folder
LBT	11/4/2015 5:28 PM	File folder
LCFile	11/4/2015 5:28 PM	File folder
LCheckSIM	11/4/2015 5:28 PM	File folder
LDateTime	11/4/2015 5:28 PM	File folder
LDisplay	11/4/2015 5:28 PM	File folder
LGPS	11/4/2015 5:28 PM	File folder
LGSM	11/20/2015 11:10 ...	File folder
LGTouch	11/4/2015 5:28 PM	File folder
LLedMatrix	11/4/2015 5:28 PM	File folder
LMotor	11/4/2015 5:28 PM	File folder
LNFC	11/4/2015 5:28 PM	File folder
LSensorHub	11/4/2015 5:28 PM	File folder
LWS2812	11/4/2015 5:28 PM	File folder
SPI	11/4/2015 5:28 PM	File folder
Wire	11/4/2015 5:28 PM	File folder

RePhone Community

We’ve been looking for a better place where our backers (RePhone Users) can sit together, warmly and comfortably, have conversations about RePhone, discuss technical problems, share ideas/projects, and give feedback on the modules’ development in the future. And then here we go, the RePhone Community.

Now join us in the RePhone Community! Together we seek answers, make interesting stuff, care about each other, and share our experiences.

Frequently Asked Questions

Some frequently asked questions in RePhone Community are collected and answered to the topic **“Frequently Asked Questions of RePhone (FAQ)”** , the topic will be kept updating whenever a new FAQ comes out.

Resources

Please go the Wiki page Xadow Duino for more info.

Help us make it better

Thank you for choosing Seeed. A couple of months ago we initiated a project to improve our documentation system. What you are looking at now is the first edition of the new documentation system. Comparing to the old one, here is the progresses that we made:

- Replaced the old documentation system with a new one that was developed from Mkdocs, a more widely used and cooler tool to develop documentation system.
- Integrated the documentation system with our official website, now you can go to Bazaar and other section like Forum and Community more conveniently.
- Reviewed and rewrote documents for hundreds of products for the system's first edition, and will continue migrate documents from old wiki to the new one.

An easy-to-use instruction is as important as the product itself. We are expecting this new system will improve your experience when using Seeed's products. However since this is the first edition, there are still many things need to improve, if you have any suggestions or findings, you are most welcome to submit the amended version as our contributor or give us suggestions in the survey below, Please don't forget to leave your email address so that we can reply.

Happy hacking