

# ESP8266 Joylink2.0 Demo 介绍

---

## 1. 概述

ESP8266 平台实现了京东微联 Joylink2.0 协议。用户可以参考 Espressif 提供的设备端 Demo 进行二次开发，快速接入京东微联云平台。

Demo 参考京东官方 [Joylink2.0 SDK](#)，添加了 WiFi 相关、OTA、Flash 存储、button 等适配层，用户只需要关心少部分 API，如初始化、数据收发、事件回调等，加快了用户的二次开发速度。适配层具有一定的通用性且开源，用户可根据产品需求自行修改。

Demo 使用的通讯方式为非透传模式。即云端上传的 lua 脚本为 only\_trans.lua

本文档章节介绍：

- **1. 概述：**介绍背景。
- **2. Demo 使用：**介绍 Demo 如何跑起来。包括环境搭建，编译下载，设备控制。
- **3. 开发指南：**介绍 Demo 如何二次开发。包括文件结构，功能介绍，API介绍，资源占用。
- **4. 相关链接：**给出与 joylink 相关的链接。包括 Demo 下载，微联文档。

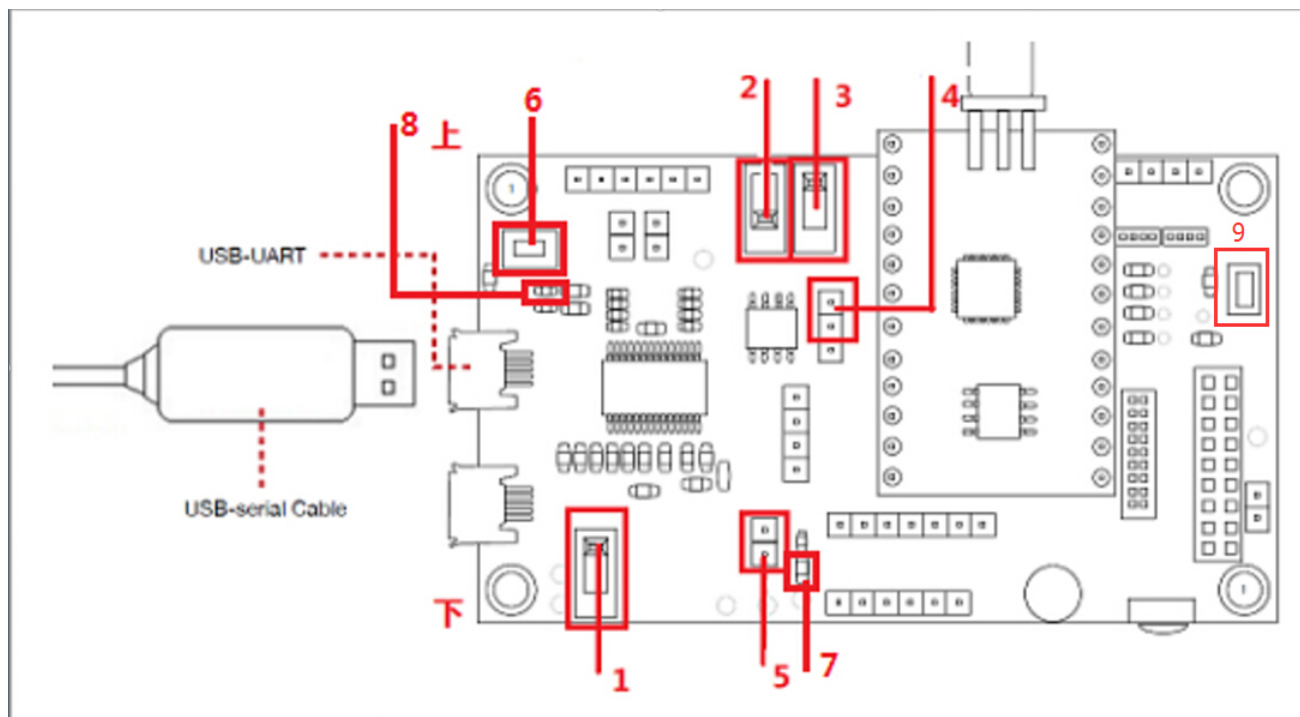
## 2. Demo 使用

用户拿到乐鑫提供 Joylink2.0 Demo 后，编译下载固件到乐鑫 ESP8266 开发板。使用京东微联 APP 扫描测试设备的二维码进行配置。配置激活成功后便可进行设备控制。此 Demo 对应的测试设备类型为“智能家居 / 生活电器 / 灯具”。

### 2.1. 环境搭建

- 硬件准备
  - 开发板：**ESP8266 开发板 ( [ESP-LAUNCHER](#) )；
  - 路由器：**可以连接外网；
  - 手机：**安装[京东微联 APP](#)。连接到路由器 2.4GHz 网络；

- 开发板介绍



标号 1 开关拨下（拨下为断电状态，拨上为上电状态）；

标号 2 开关拨下（拨下为下载模式，拨上为运行模式）；

标号 3 开关拨上（CHIP\_EN PIN，默认拨上即可）；

标号 4 跳线帽插入上方的两个针脚；

标号 5 插入跳线帽，3.3v 供电；

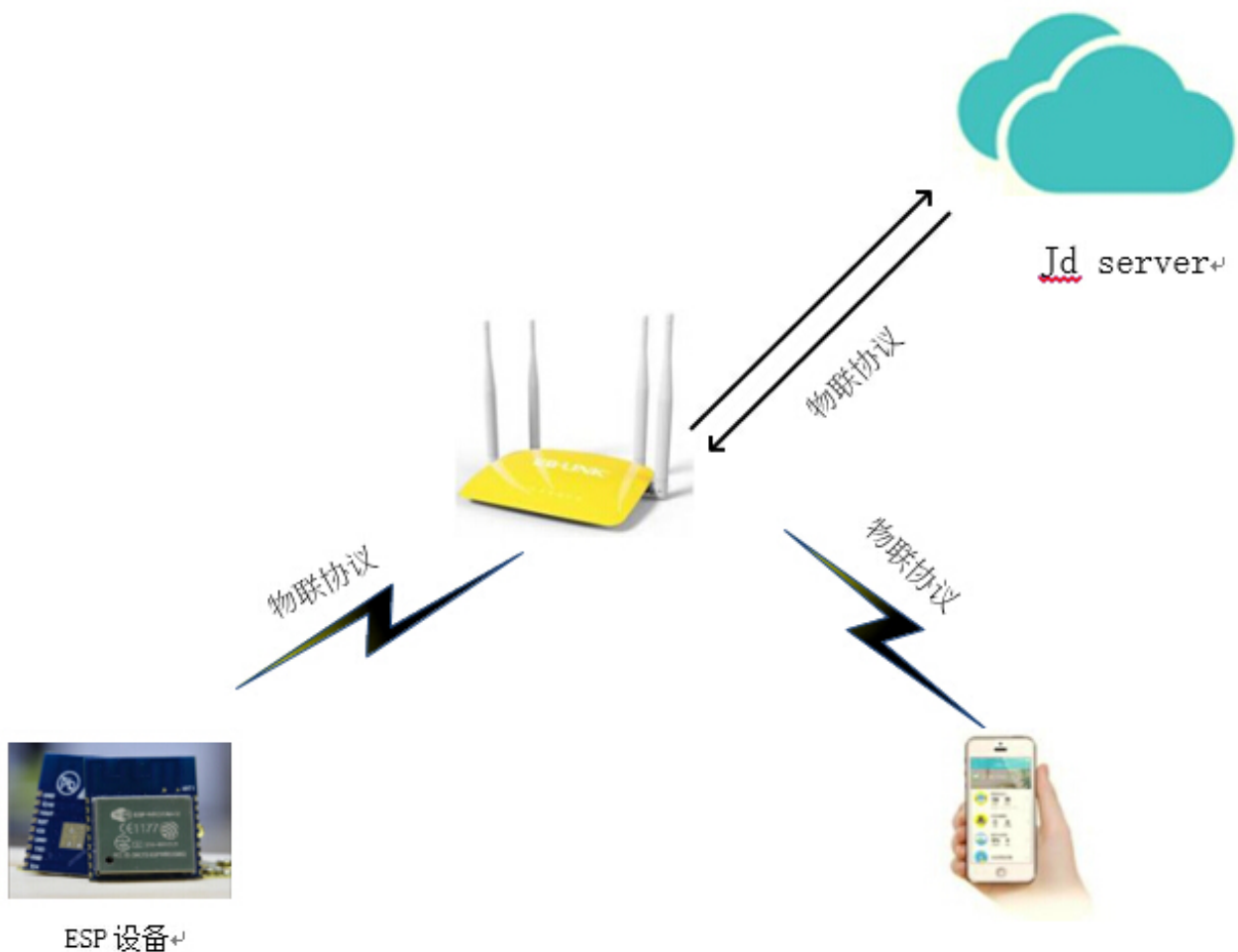
标号 6 配网开关，设备进入配网模式；

标号 7 状态指示灯，设备配网慢闪->连接服务器快闪->联网成功常亮->控制成功状态改变；

标号 9 出厂设置开关，设备端会解绑并恢复出厂设置；

标号 10 连接 GPIO14 与 SW2 旁边的 J70，使能 SW2 按键，即出厂设置开关。

- 环境拓扑结构



## 2.2 编译下载

- 文件下载

官网下载 Demo 工程和最新版本 [ESP8266 RTOS SDK](#)。修改 demo/gen\_misc.sh 脚本中 SDK 路径。

```
1. export SDK_PATH=$(pwd) / ../ESP8266_RTOS_SDK_1.5.0
2. export BIN_PATH=$(pwd) / ../ESP8266_RTOS_SDK_1.5.0/bin
```

- bin 文件

默认编译生成的文件为 user1.2048.new.5.bin, 至少使用 2MByte flash 下载。可作为 OTA 文件上传到服务器后台。其他 Bin 文件 SDK 自带, 非编译生成。

boot.bin----->0x000000 // 启动程序

user1.2048.new.5.bin----->0x01000 // 主程序, 编译生成, OTA文件。

blank.bin----->0xFC000 // 初始化用户参数区

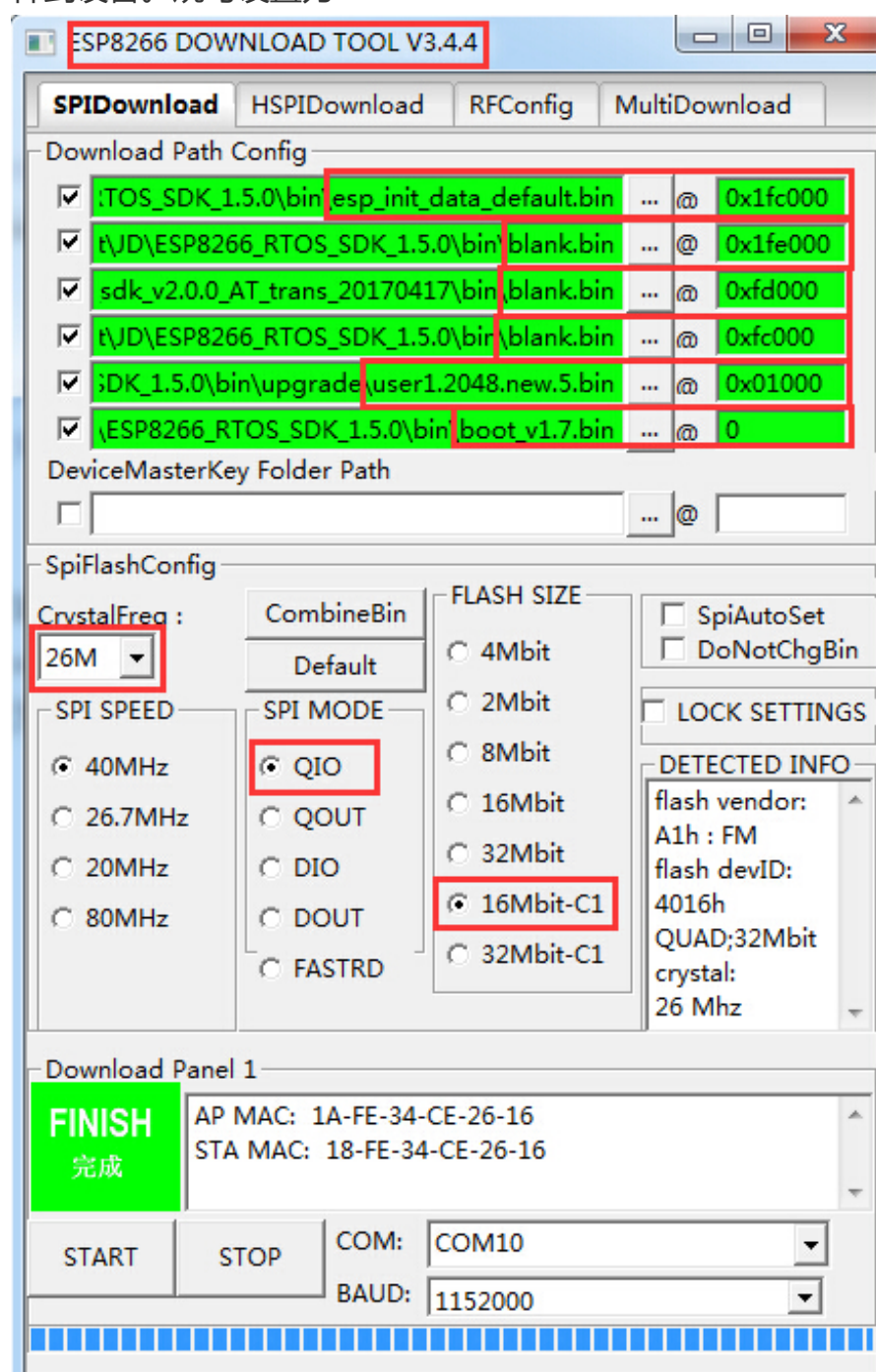
blank.bin----->0xFD000 // 初始化用户参数区。

esp\_init\_data\_default.bin-->0x1FC000 // 初始化其他射频参数区

blank.bin----->0x1FE000 // 初始化系统参数区

- 程序烧写

编译环境中运行 gen\_misc.sh 脚本，在指定路径生成 bin 文件。使用[下载工具](#)下载 bin 文件到设备。烧写设置为：



## 2.3 运行

● 操作步骤

序号	设备	状态灯	APP
1	上电，串口正常打印	不亮	-
2	配网按键，进入配网	慢闪	扫码开始配置
3	配网成功，连接服务器	快闪	显示可使用设备
4	连接服务器成功	常亮	显示设备控制界面
5	设备被控时状态灯会显示	亮灭	控制设备状态

\* 扫码配置

使用京东微联App扫码  
添加虚拟设备



● 配网界面



### 3. 开发指南

本章详细介绍 Demo 工程，如需二次开发请关注本章。

用户需要调用的 API 和参数配置在下面头文件中：

```
1. Demo/platforms/joylink/include/joylink_config.h
```

```
1. Demo/platforms/joylink/include/joylink_api.h
```

## 3.1 文件结构

```
1.  |— components                //sdk 功能模块
2.  |   |— adapter
3.  |   |— button
4.  |   |— ota
5.  |   |— param
6.  |   |— ringbuffer
7.  |   |— status_led
8.  |— document                //文档
9.  |— driver                  //ESP8266 驱动
10. |— include
11. |   |— driver              //驱动头文件
12. |— platforms              //各云平台相关
13. |   |— joylink            //joylink2.0 sdk
14. |       |— include        //用户需包含的 Joylink 头文件
15. |       |— jdinnet        //joylink2.0 配网程序
16. |       |— libjoylink
17. |           |— auth        //joylink2.0 加解密相关程序
18. |           |— extern      //ESP平台适配层代码
19. |           |— joylink      //joylink2.0 主逻辑流程
20. |           |— json         //joylink2.0 json解析
21. |           |— list
22. |— protocols              //可存放用户协议处理
23. |— user                   //用户应用程序
```

## 3.2 参数配置

文件 `Demo/platforms/joylink/include/joylink_config.h` 为用户参数配置的文件。如果需要修改为自己的设备请在此修改。

```
1.  #define JOYLINK_WAIT_DEV_STATUS_TIMEOUT (1000) /*!< ms. when cmd down,
    joylink will wait cmd reply */
2.  #define JOYLINK_TASK_STACK_MAIN (1024-256)      /*!< word. joylink loop
    task stack, test by API "high water" */
3.  #define JOYLINK_TASK_PRIOTY (tskIDLE_PRIORITY + 2) /*!< The base priot
    y of joylink */
4.  #define JOYLINK_STORAGE_LOCATION (0xfc) /*!< joylink info save
    location, use 3 sector */
5.  #define JOYLINK_IS_PASSTHOUGH (0) /*!< 0/1. joylink mode. if
```

```

passthrough you should upload lua */
6.  #define JOYLINK_OTA_RX_MAXLEN      (1024)  /*!< max-1460. limit the ea
    ch packet length of OTA */
7.  #define JOYLINK_SNTP_SERVER1_NAME  "cn.pool.ntp.org"  /*!< first
    sntp server addr, sync internet time */
8.  #define JOYLINK_SNTP_SERVER2_NAME  "edu.ntp.org.cn"   /*!< second
    sntp time */
9.  #define JOYLINK_SMNT_BUTTON_NUM    (13) /*!< set button pin ,restart to
    smartconfig */
10. #define JOYLINK_RESET_BUTTON_NUM   (14) /*!< set button pin ,chear flas
    h info and restart to smartconfig */
11. #define LED_IO_NUM                  (15) /*!< set led gpio, led status */

```

以下参数为产品信息，需要在服务器后台产品基本信息里获取，或者向京东有关部门咨询。

```

1.  joylink_info_t product_info = {
2.      .innet_aes_key      = JOYLINK_AES_KEY,    //SecretKey
3.      .jlp.version        = JOYLINK_VERSION,    //固件版本，从1递增。需保
    证与服务器后台固件升级里设置的版本对应，
4.      .jlp.accesskey      = JOYLINK_ACCESSKEY,  //激活过程产生
5.      .jlp.localkey       = JOYLINK_LOCAL_KEY,  //激活过程产生
6.      .jlp.feedid        = JOYLINK_FEEDID,     //激活过程产生
7.      .jlp.devtype        = JOYLINK_DEVTYPE,    //设备类型，向京东获取
8.      .jlp.joylink_server = JOYLINK_SERVER,     //激活过程产生
9.      .jlp.server_port    = JOYLINK_SERVER_PORT, //激活过程产生
10.     .jlp.CID             = JOYLINK_CID,        //设备品类码，向京东获取
11.     .jlp.firmwareVersion = JOYLINK_FW_VERSION, //固件版本名称，需保证与
    服务器后台固件版本一直，固定格式，参考京东提供SDK文档
12.     .jlp.modelCode       = JOYLINK_MODEL_CODE, //设备型号
13.     .jlp.uuid            = JOYLINK_UUID,       // UUID，后台获取
14.     .jlp.lancon          = JOYLINK_LAN_CTRL,   // 设备是否支持局域网控制
15.     .jlp.cmd_tran_type   = JOYLINK_CMD_TYPE    // 数据是json或者scirpt
    方式传输
16. };

```

### 3.3 配网

京东一键配置方式包括：BLE 配网，京豆标准一键配置，京豆标准一键配置+SoftAp。Demo中使用的配网方式为京东标准一键配置，后台填写的联网方式代码为 `KMYXDI63HNBFF9CG`。如需其他配网方式代码请咨询京东。



## 产品配置

联网方式

WIFI

KMYXDI63HNBFF9CG

京东标准一键配置

局域网协议支持

支持局域网协议V1.0 ▼

编辑配置说明

## 3.4 数据收发

数据收发我们使用了 事件 + read/write 函数的操作。用户在事件 callback 中收到不同的事件做相应的操作。

```
1.  typedef enum {
2.      JOYLINK_EVENT_NONE = 0,
3.      JOYLINK_EVENT_WIFI_START_SMARTCONFIG,
4.      JOYLINK_EVENT_WIFI_GOT_IP,
5.      JOYLINK_EVENT_WIFI_DISCONNECTED,
6.      JOYLINK_EVENT_CLOUD_CONNECTED,
7.      JOYLINK_EVENT_CLOUD_DISCONNECTED,
8.      JOYLINK_EVENT_GET_DEVICE_DATA,  /*!< should call esp_joylink_write
send dev status to server */
9.      JOYLINK_EVENT_SET_DEVICE_DATA,  /*!< should call esp_joylink_read
read the cmd from server */
10.     JOYLINK_EVENT_OTA_START,
11. } joylink_event_t;
12.
13. /**
14.  * @brief upload device info to joylink SDK.
15.  *
16.  * @param down_cmd  [out]get the cmd from server or lan.
17.  * @param len  [in]the lenth of down_cmd, should less than
JOYLINK_DATA_LEN.
18.  * @param micro_seconds  timeout.
19.  *
20.  * @return
21.  *      - >0 : real cmd lenth
22.  *      - <= : read err
23.  */
```

```

24.  int esp_joylink_read(void *down_cmd, size_t size, int micro_seconds);
25.
26.  /**
27.   * @brief upload device info to joylink SDK.
28.   *
29.   * @param up_cmd [in] must be upload cmd with the format.
30.   * @param len [in] the length of up_cmd, should include '\0', should be less than JOYLINK_DATA_LEN.
31.   * @param micro_seconds [in] timeout.
32.   *
33.   * @return
34.   * - >0 : real cmd length
35.   * - <= : read error
36.   */
37.  int esp_joylink_write(void *up_cmd, size_t len, int micro_seconds);

```

## 3.5 OTA

OTA 相关代码在 `Demo/platforms/joylink/libjoylink/extern/joylink_upgrade.c` 文件中处理。

实现 OTA 需要上传 ota 文件到服务器后台。并填写正确的版本信息。

**注意：** 厂商必须严格按照固定格式填写版本号，如（1.1.1.20151020\_release）。

如下图，后台设置在线升级界面。序号是上传 bin 文件后自动生成的。设备端上传的 version 如果与这个版本号不同，APP 会自动推送 OTA 的提示。



## 3.6 辅助功能

增加了按键重启，与 led 状态指示功能。用户可以修改宏使能。

## 3.7 资源使用

下面表格是 Demo 中各任务分配的栈空间及使用情况。单位是 word。

task - 指任务函数名

stack - 指创建任务时分配的栈空间

high water - 指运行过程中各任务栈空间的最高水纹

task	stack	high water
read_task_test	256	77
write_task_test	256	56
joylink_event_ctrl_task	192	72
joylink_task	948	183
jd_innet_start_task	896	72
sys_start_snmp_server	320	95
fota_begin	384	50

## 4. 相关链接

- Espressif 官网：<http://espressif.com>
- ESP8266 SDK 下载：  
<http://espressif.com/en/products/hardware/esp8266ex/resources>
- 烧录工具：<http://espressif.com/en/support/download/other-tools>
- 京东微联官网：<https://smartcloud.jd.com>
- 官方JoyLink协议SDK：[https://storage.jd.com/testsmartcloud/joylink\\_dev\\_sdk.zip](https://storage.jd.com/testsmartcloud/joylink_dev_sdk.zip)
- 设备入网的协议文档：<https://storage.jd.com/testsmartcloud/JoylinkConfigDoc.zip>