

ESP-NOW Test Report

- [1. 单设备](#)
 - [单程延时](#)
 - [往返延时](#)
 - [吞吐量](#)
 - [安全传输](#)
- [2. 多设备](#)
 - [多设备响应时间](#)
 - [多设备往返延时](#)
 - [OTA](#)
 - [安全握手](#)
 - [配网](#)

单设备

- 测试工具：
 - 一台电脑
 - 两块 ESP32 设备或者两块 ESP32C3 设备
 - 示波器
- 测试代码：
 - 基于 ESP-NOW [solution](#) 示例进行测试。
 - 主设备配置：

```
idf.py menuconfig > Example Configuration > ESP-NOW Mode > ESP-NOW initiator Mode
```
 - 从设备配置：

```
idf.py menuconfig > Example Configuration > ESP-NOW Mode > ESP-NOW responder Mode
```
 - ESP-IDF 版本使用 v4.4.1
- 测试场景：办公室内环境，设备距离地面 1.2m

用户应注意，测试的性能与工作环境密切相关。在没有特别说明时，以下测试数据未加密。不加密配置：

```
idf.py menuconfig > Example Configuration > [ ] Enable ESPNOW security
```

 并且设置

```
idf.py menuconfig > Component config > ESP-NOW Security Configuration > [ ] Application data security
```

 默认的工作信道为信号 1

单程延时

发送端发送开始到接收端接收到数据的时间差。

- 测试工具
 - 示波器
- 测试场景
 - 办公环境，主设备和从设备距离 0.5m
- 测试步骤
 - 主设备 IO17 连示波器的一个通道，低电平触发，发送一笔广播数据
 - 从设备 IO17 连示波器的另一个通道，收到数据后，低电平触发
 - 从设备执行命令

```
espnw_iperf -s -b 0 -G 17
```

，并记录从设备的 Mac 地址 `xx:xx:xx:xx:xx:xx`，其中 `-G 17` 表示 GPIO 17。
 - 主设备执行命令

```
espnw_iperf -c xx:xx:xx:xx:xx:xx -b 0 -p -C 1 -l 10 -G 17
```

，其中 `-l 10` 表示发送的 Payload 长度是 10 Bytes，默认长度是 230 Bytes。
 - 示波器测量两个电平下降沿的时间差。发送十笔包，计算最大值，最小值，平均值。
- 测试结果

Payload size(bytes)	10	50	100	150	200	230
Response Time (min)	2.28 ms	2.23 ms	2.29 ms	2.78 ms	3.26 ms	3.36 ms
Response Time (max)	3.06 ms	6.09 ms	9.94 ms	9.14 ms	13.60 ms	15.66 ms
Response Time (avg)	2.43 ms	3.57 ms	4.22 ms	4.88 ms	5.74 ms	5.37 ms

往返延时

从发送端开始发送数据到收到接收端回应的的时间差。

使用 `espnw_iperf -p ping` 命令，发送一笔数据后，设备端回复同样长度的数据，统计过程中的时间差，丢包率。

具体可以参考 [cmd_iperf.c](#) 文件。

- 测试场景

- 办公环境，主设备和从设备距离 3m

- 测试步骤

- 从设备执行命令 `espnw_iperf -s`，并记录从设备的 MAC 地址 `xx:xx:xx:xx:xx:xx`
- 主设备执行命令 `espnw_iperf -c xx:xx:xx:xx:xx:xx -p -C 10000 -l 10`，其中
 - `-b 0` 或者 `-b 1` 表示用广播包发送，且只发送一次，不重发。
 - `-C 10000` 表示一共发送 10000 笔数据。
 - `-l 10` 表示发送的 Payload 长度是 10 Bytes，默认长度是 230 Bytes。
- 记录主设备端 log 中的丢包率和延时

- 测试结果

- 主从设备都使用广播包发送，并且设置重发次数为 3，命令中设置参数 `-b 3`。

Payload size(bytes)	10	50	100	150	200	230
Delay Time (min)	2 ms	3 ms	4 ms	4 ms	5 ms	6 ms
Delay Time (max)	49 ms	88 ms	80 ms	85 ms	93 ms	78 ms
Delay Time (avg)	5.63 ms	7.43 ms	10.74 ms	13.59 ms	16.60 ms	17.78 ms
Loss Rate	0.22%	0.13%	0.17%	0.15%	0.31%	0.34%

- 主从设备都使用广播包发送，设置重发次数为 3，命令中设置参数 `-b 3`，并且都设置 ACK 机制(接收端回复确认消息)，命令中设置参数 `-A`。

Payload size(bytes)	10	50	100	150	200	230
Delay Time (min)	6 ms	7 ms	9 ms	11 ms	13 ms	15 ms
Delay Time (max)	123 ms	146 ms	135 ms	92 ms	85 ms	136 ms
Delay Time (avg)	11.54 ms	14.26 ms	17.67 ms	20.38 ms	23.20 ms	25.10 ms
Loss Rate	0.13%	0.19%	0.39%	0.39%	0.21%	0.27%

- 主从设备都使用单播包发送，命令中去掉参数 `-b xx`。

Payload size(bytes)	10	50	100	150	200	230
Delay Time (min)	2 ms	3 ms	4 ms	4 ms	5 ms	6 ms
Delay Time (max)	75 ms	79 ms	96 ms	73 ms	121 ms	94 ms
Delay Time (avg)	4.56 ms	6.01 ms	6.69 ms	7.83 ms	9.37 ms	11.45 ms
Loss Rate	0.11%	0.12%	0.13%	0.09%	0.08%	0.10%

吞吐量

使用 solution 示例时，为避免从设备发送 log 影响性能，主从设备都使用默认的

`idf.py menuconfig > Example Configuration > ESP-NOW Mode > ESP-NOW initiator Mode` 配置。使用 `espnw_iperf -c/-s` 命令，测试 `espnw` 在不同设置下的吞吐量。具体可以参考 [cmd_iperf.c](#) 文件。

- 测试场景

- 办公环境，主设备和从设备分别距离 3m

- 测试步骤

- 从设备和主设备分别执行如下命令进行测试，其中 `xx:xx:xx:xx:xx:xx` 表示从设备的 MAC 地址。
 - 单播包发送：从设备执行 `espnw_iperf -s`，主设备执行：`espnw_iperf -c xx:xx:xx:xx:xx:xx`
 - 广播包发送：从设备执行 `espnw_iperf -s -b 0`，主设备执行：`espnw_iperf -c xx:xx:xx:xx:xx:xx -b 0`
 - 广播包重发 3 次：从设备执行 `espnw_iperf -s -b 3`，主设备执行：`espnw_iperf -c xx:xx:xx:xx:xx:xx -b 3`
 - 广播包重发 3 次 + ACK 机制：从设备执行 `espnw_iperf -s -b 3 -A`，主设备执行：`espnw_iperf -c xx:xx:xx:xx:xx:xx -b 3 -A`
- 默认发送长度是 230 Bytes，添加 `-l xx` 命令指定发送的 Payload 长度。
- 记录从设备端 log 中的吞吐率和丢包率。

- 测试结果

- 吞吐量

Payload size(bytes)	10	50	100	150	200	230
unicast	0.03 Mbps	0.13 Mbps	0.22 Mbps	0.28 Mbps	0.33 Mbps	0.36 Mbps
broadcast 0	0.06 Mbps	0.22 Mbps	0.34 Mbps	0.41 Mbps	0.46 Mbps	0.49 Mbps
broadcast 3	0.02 Mbps	0.07 Mbps	0.11 Mbps	0.14 Mbps	0.16 Mbps	0.17 Mbps
broadcast 3 + ACK	0.01 Mbps	0.01 Mbps	0.02 Mbps	0.02 Mbps	0.04 Mbps	0.08 Mbps

- 丢包率

Payload size(bytes)	10	50	100	150	200	230
unicast	13/22823 (0.06%)	12/20010 (0.06%)	9/16639 (0.05%)	10/14079 (0.07%)	8/12465 (0.06%)	5/11740 (0.04%)
broadcast 0	1585/43849 (3.61%)	1504/33366 (4.51%)	1210/25292 (4.78%)	1124/20330 (5.53%)	932/17264 (5.40%)	948/15962 (5.94%)
broadcast 3	38/14659 (0.26%)	43/11059 (0.39%)	37/8331 (0.44%)	23/7232 (0.32%)	21/6056 (0.35%)	11/5595 (0.20%)
broadcast 3 + ACK	2/5009 (0.04%)	1/1620 (0.06%)	3/1356 (0.22%)	1/1106 (0.09%)	3/1600 (0.19%)	3/2552 (0.12%)

安全传输

启用应用数据安全加密功能：`idf.py menuconfig > Example Configuration > [*] Enable ESPNOW security` 并且设置

`idf.py menuconfig > Component config > ESP-NOW Security Configuration > [*] Application data security`

- 加解密耗时

- 使用一个设备，执行 `sec_test -C 10000 -l xx` 命令，其中 `-l xx` 指定被加密数据的长度，默认是 230。
`-c 10000` 表示加密解密 10000 次后计算加密解密的平均时间。
- 记录命令执行结果中的加解密平均时间。

Payload size(bytes)	10	50	100	150	200	230-4
Encrypt Time	47 us	109 us	171 us	235 us	297 us	339 us
Decrypt Time	47 us	109 us	173 us	236 us	299 us	341 us

- 单程延时

- 使用 solution 示例默认配置，先启动从设备，再启动主设备。
- 等主从设备完成安全握手，打印 `Get APP key` .
- 其他设置同上述 [单程延时](#)

Payload size(bytes)	10	50	100	150	200	230-4
Response Time (min)	2.16 ms	2.35 ms	2.82 ms	3.50 ms	3.98 ms	4.46 ms
Response Time (max)	5.27 ms	3.69 ms	6.86 ms	12.24 ms	6.84 ms	7.86 ms
Response Time (avg)	3.26 ms	2.95 ms	3.84 ms	5.00 ms	5.28 ms	5.41 ms

- 往返延时

- 使用 solution 示例默认配置，先启动从设备，再启动主设备。
- 等主从设备完成安全握手，打印 `Get APP key` .
- 使用 `espnow_iperf -p` 命令，测试广播包不重发，主设备和从设备距离 0.5m 时的往返延时
 - 从设备执行命令 `espnow_iperf -s -b 0`，并记录从设备的 MAC 地址 `xx:xx:xx:xx:xx:xx`
 - 主设备执行命令 `espnow_iperf -c xx:xx:xx:xx:xx:xx -b 0 -p -C 10000 -l 10`，其中 `-l xx` 指定被加密数据的长度，加密状态下默认是 226.
 - 记录主设备端 log 中的丢包率和延时

Payload size(bytes)	10	50	100	150	200	230-4
Delay Time (min)	3 ms	4 ms	5 ms	6 ms	7 ms	7 ms
Delay Time (max)	48 ms	57 ms	55 ms	56 ms	56 ms	61 ms
Delay Time (avg)	4.48 ms	5.55 ms	6.91 ms	8.08 ms	8.94 ms	9.44 ms
Loss Rate	0.35%	0.41%	0.47%	0.47%	0.47%	0.47%

多设备

- 测试工具：
 - 两台电脑
 - 若干 USB hub
 - 101 块 ESP32 设备，其中一块作为主设备和最多 100 块作为从设备
- 测试代码：
 - 基于 ESP-NOW [solution](#) 示例进行测试
 - ESP-IDF 版本使用 v4.4.1
- 测试场景：办公室内环境

用户应注意，测试的性能与工作环境密切相关。在没有特别说明时，以下测试数据未加密。

多设备响应时间

- 由于主设备发送广播包，多设备响应时，不方便进行测量。
 - 如果通过网络时间的方式，误差有 10ms 以上，对于响应时间就会测量不准确。
 - 通过示波器测量比较准确，但是不方便接大量设备。
- 因为主设备发送广播包，从设备都可以进行接收，因此
 - 多设备单程延时可以参考单设备的[单程延时](#)。
 - 或者参考单设备的[往返延时](#)，单程延时 = 往返延时/2。
 - 考虑可能的转发次数，响应时间 = 单程延时 * 转发次数。

多设备往返延时

当多设备同时发送数据时，发送端发送数据到收到接收端回应的的时间差。

- 测试场景
 - 办公环境，主设备和从设备分别距离 0.5m, 3m
 - 一个主设备和最多 100 个从设备。
- 测试步骤
 - 主设备执行命令 `espnow_iperf -s`，并记录主设备的 Mac 地址 `xx:xx:xx:xx:xx:xx`
 - 设置从设备端 log 等级为 error: `command FF:FF:FF:FF:FF:FF "log -m espnow -l ERR"`，并将从设备固件中 `ping report` 结果打印从 `ESP_LOGI` 改为 `ESP_LOGE`。可通过固件升级的方式修改固件。这样主设备端之后可以清楚的显示从设备端的结果 log。
 - 从设备发送单播包：在主设备端执行命令 `command FF:FF:FF:FF:FF:FF "espnow_iperf -c xx:xx:xx:xx:xx:xx -C 100 -p"`，即可在从设备端批量执行命令。
 - 从设备发送广播包：在主设备端执行命令 `command FF:FF:FF:FF:FF:FF "espnow_iperf -c xx:xx:xx:xx:xx:xx -b 0 -C 100 -p"`，即可在从设备端批量执行命令。
 - 记录主设备端显示的从设备端 log 中的丢包率和延时，并且计算平均值。
- 测试结果
 - 子设备发送单播包时，往返延时和测试中的丢包率

Device Num	30	50	100
Average Delay Time (0.5m away)	384.50 ms	607.37 ms	1124.50 ms
Average Delay Time (3m away)	404.47 ms	611.09 ms	1312.76 ms

Device Num	30	50	100
MAX Delay Time (0.5m away)	858 ms	1237 ms	2285 ms
MAX Delay Time (3m away)	638 ms	1682 ms	2236 ms

Device Num	30	50	100
Average Loss Rate (0.5m away)	0.09%	0.08%	0.09%
Average Loss Rate (3m away)	0.17%	0.06%	0.29%

- 子设备发送广播包时，往返延时和测试中的丢包率

Device Num	30	50	100
Average Delay Time (0.5m away)	21.55 ms	26.57 ms	45.58 ms
Average Delay Time (3m away)	24.95 ms	26.98 ms	47.92 ms

Device Num	30	50	100
MAX Delay Time (0.5m away)	242 ms	260 ms	268 ms
MAX Delay Time (3m away)	198 ms	249 ms	315 ms

Device Num	30	50	100
Average Loss Rate (0.5m away)	15.43%	17.38%	25.35%
Average Loss Rate (3m away)	14.90%	18.70%	28.38%

注

Average Delay Time: 对每个设备的平均延时时间，求和取平均值。MAX Delay Time: 对每个设备的最大延时时间，取最大值。Average Loss Rate: 对每个设备的丢包率，求和取平均值。

OTA

- 测试场景
 - 办公环境，主设备和从设备距离 0.5m~1m
 - 一个主设备，1~100 个从设备
- 测试步骤
 - PC 端编译升级固件，连接路由器，并且设置路由器信道为 1，PC 端起 http server
 - 主设备用命令 `wifi_config -s ssid -p password` 连接上述路由器
 - 主设备用命令 `ota -d http://192.168.0.6:8070/Resp.bin` 从 http server 端下载固件
 - 主设备用命令 `ota -f 3000` 扫描从设备并且获取设备列表
 - 主设备用命令 `ota -s <xx:xx:xx:xx:xx:xx>,<xx:xx:xx:xx:xx:xx>,...` 发送升级固件到从设备
 - 待所有设备完成固件升级，记录主设备 log 中打印的完成时间。

注：如果需要回退版本，可以使用命令

```
command <xx:xx:xx:xx:xx:xx>,<xx:xx:xx:xx:xx:xx>,.. "rollback" , 再执行重启命令  
command <xx:xx:xx:xx:xx:xx>,<xx:xx:xx:xx:xx:xx>,.. "restart" , 使设备回退到之前的版本运行。
```

如果待升级设备比较多(大于 28 个), 受输入长度限制,

```
ota -s <xx:xx:xx:xx:xx:xx>,<xx:xx:xx:xx:xx:xx>,.. 命令可以改为执行  
ota -s FF:FF:FF:FF:FF:FF 命令, 命令执行中将再次扫描设备并且对扫描设备进行升级。
```

以免从设备发送太多 log 影响主设备接收, 可以通过命令

```
command FF:FF:FF:FF:FF:FF "log -m espnow -l WARN" 设置从设备发送的 log level 为 WARN .
```

• 测试结果

本次测试中升级固件长度为 1,029,248 字节。

Device Num	1	10	20	30	40	50	60	70	80	100
Spend Time (0.5m~1m away)	22s	25s	26s	33s	33s	35s	36s	39s	38s	39s

测试主设备和从设备不同距离下, 100 个从设备完成升级的时间:

Device Num	100
Spend Time (3m away)	50s
Spend Time (5m away)	65s
Spend Time (10m away)	75s

安全握手

• 测试场景

- 办公环境, 主设备和从设备分别距离 1m
- 一个主设备, 1~100 个从设备

• 测试步骤

- 从设备烧写开启加密功能的固件, 或者通过固件升级的方式下载开启加密功能的固件。
- 如果从设备已经获取加密密钥, 需要擦除重新获取时, 可以执行命令 `security -e`, 或者通过主设备执行命令 `command FF:FF:FF:FF:FF:FF "security -e"`, 对所有从设备批量擦除加密密钥。
- 从设备启动, 等待获取加密密钥
- 主设备启动, 待所有设备完成握手过程并获取密钥, 记录主设备 log 中打印的完成时间。

• 测试结果

Device Num	1	10	20	30	40	50	60	70	80	100
Spend Time (1 m away)	3.6s	6.4s	11.2s	13.4s	16.9s	20.9s	24.5s	27.6s	30.6s	37.2s

`Spend Time` 中包含大约 2520 ms 的扫描时间。

配网

• 测试场景

- 办公环境, 主设备和从设备分别距离 1m
- 一个主设备, 1~100 个从设备
- 需要连接的路由器为环境中同名 ssid 的路由器, 一共有 8 台, 分别处在 1, 3, 6, 8, 11, 13 信道。

• 测试步骤

- 从设备如果已配置网络，长按 provision 按键，会擦除配网信息并重新启动。或者主设备端使用 `command -a FF:FF:FF:FF:FF:FF "provisioning -e"` 命令擦除所有从设备配网信息。由于设备可能处于不同信道，`-a` 即全信道发送命令
- 从设备上双击 provision 按键，灯变为白色后表示进入配网状态。或者主设备端使用 `command FF:FF:FF:FF:FF:FF "provisioning -r"` 命令使所有从设备进入配网状态。
- 主设备发送配网信息，并且开始计时。使用命令 `provisioning -i 30000 ssid password`，其中 `30000` 表示配网开启时长为 30,000 ms，即 30s，ssid 和 password 为从设备需要连接的路由器名称和密码。
- 从设备灯全部变为绿色或者红色时，停止计时。

以免从设备发送太多 log 影响主设备接收，可以通过命令

`command FF:FF:FF:FF:FF:FF "log -m espnow -l WARN"` 设置从设备发送的 log level 为 `WARN`。

• 测试结果

Device Num	1	10	20	30	40	50	60	70	80	100
Spend Time (1m away)	5s	6s	7s	9s	13s	13s	13s	14s	16s	18s