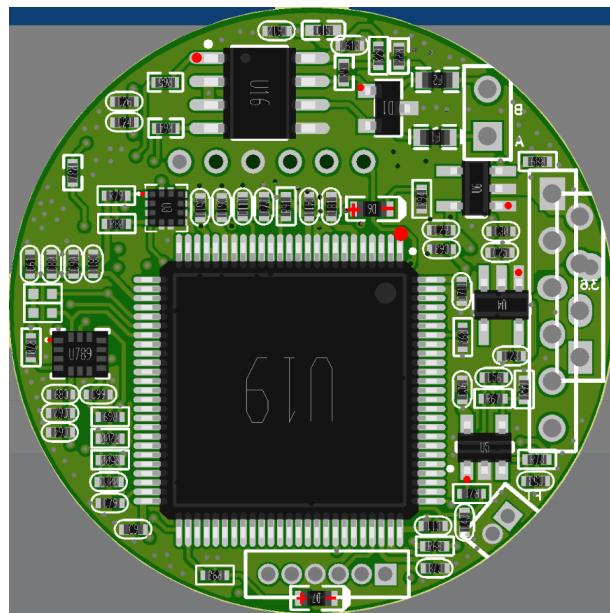


# 三轴振动传感器 PCBA – Quick start Guide

v. 1.1

IOT6-Z2/微功耗/主频率/加速度峰值/加速度峰峰值/加速度/峭度/速度  
4G/485/Lora

2025 年 8 月 31 日



## Introduction

IOT6-Z2 微功耗振动传感器集成了数字三轴传感器，通过 FFT 技术，高精度输出主频率/加速度峰值/加速度峰峰值/加速度 RMS/峭度/速度 RMS。同时集成了高精度模拟三轴传感器，通过积分电路，输出振动速度。

IOT6-Z2 采用 Arm Cortex-M4 32b MCU+FPUd 单片机，1MB Flash/192+4KB RAM 设计，保证系统能在 1024HZ 下完整采样运行，系统休眠电流只有 7uA，为电池运行提供基础支撑。

TTL 串口支持 JSON 报文输出，输出时间和运行方式可以自主定义。方便 Lora、4G 通信以及二次开发。

7uA 待机，三轴传感器一直在工作，振动毫秒唤醒，微功耗首选！

# 目录

1 IOT6-Z2 温度振动传感器简介	3
2 认证证书	5
3 IOT6-Z2 专利算法	5
3.1 三轴加速度计数学模型	5
3.2 静态模型（重力场测量）	6
3.3 六面校准法	6
3.4 运动参数计算，加速度积分	6
3.5 振动分析	7
3.6 频域分析	7
3.7 温度测量模型	7
3.8 振动测量模型	8
3.9 振动特征参数计算	8
3.10 温度补偿算法	8
3.11 集成算法流程	9
4 振动测试床测试结果，本 PCBA 输出	9
4.1 振动测量数据分析	10
4.2 数据分析结论	10
5 主要 BOM 表格，不包括阻容，不同型号配置不同	11
6 应用接口	11
6.1 JSON 接口	11
6.2 模拟部分参数	11
6.3 数字部分参数	13
6.4 JSON 格式	15
7 4G 模块和软件分析接口	18
7.1 基于 Luat 的微功耗 4G 模组，集成 GPS	18
7.2 Web 端接口展示	21
8 Appendix	25

# 1 IOT6-Z2 温度振动传感器简介

设备贴片图版本 1.1 见 9.

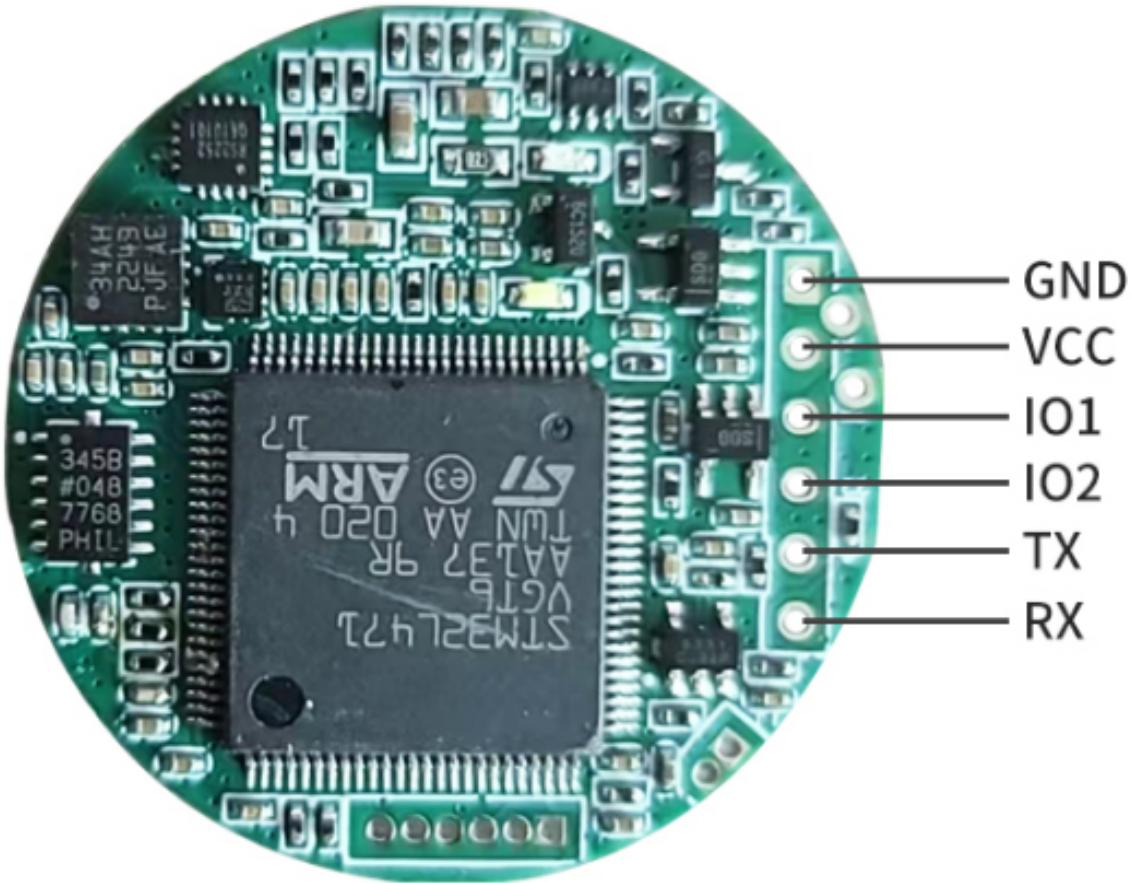


图 1: IOT6-Z2 PCB 和接口, 直径 33mm, TTL 或者 485 数字接口

型号: 485-ST (Rev 1.1)

## 1. 概述

紧凑型嵌入式通信控制模块, 集成工业级 RS-485、数字加速度计 (IIS3DWBTR)、无线 (433MHz) 及高精度温度采集 (PT100) 功能, 适用于物联网节点、远程监测及微功耗传感场景。一节电池用 8 年。

## 2. 机械规格

外形: 圆形 PCB, 直径 33mm, 厚度 1mm (符合紧凑安装需求)。

工艺: FR4 材质, 4 层沉金工艺, 阻抗控制布线。

## 3. 通信接口

RS-485: 支持 Modbus RTU 协议, 15kV ESD 保护, 120Ω 终端电阻可配置。

### 三路 TTL UART:

UART5: 通用调试接口，通常接入串口屏。

UART4: 433MHz 无线: 支持 ASK/OOK 调制，最大发射功率 +20dBm，预留天线匹配网络调试位。

UART3: 支持硬件流控 (RTS/CTS)，集成 2 路 GPIO (可配置为 4G 模块电源使能控制，典型应用：发送前拉高电平唤醒，通信结束后强制断电以降低功耗)。

### 4. 传感器接口

PT100: 全量程补偿算法，支持校准。

IIS3DWBTR 或者其他型号: 内置三轴 MEMS 加速度计，SPI 接口，量程2g 到16g，支持振动监测。

LIS344: 高精度模拟积分传感器 (部分型号配置)。

### 5. 电源管理

输入电压: 3.3V-5V (低纹波 LDO 稳压)。

动态功耗控制: 支持通过 UART3 GPIO 关断 4G 模块电源，静态电流 < 8A (休眠模式)。如果取消振动唤醒，设备休眠电流为 2.5uA。

### 6. 扩展与调试

预留 SWD 编程接口，支持固件在线 OTA 升级。

板载状态 LED (可软件禁用)，其中白灯代表 PT100 读取完毕，红灯代表 TTL 数据发送完毕。

### 7. 典型应用

工业设备振动监测 (IIS3DWBTR + RS-485 上传)。

远程温度采集 (PT100 + 433MHz 无线传输)。

低功耗 4G 数传终端 (UART3 控制模块启停)。

### 8. 认证与兼容性

符合 EMC Class B 标准，RoHS 认证。

工作温度: -40 +85 (工业级)。

外壳可以是多种选择，一般应该选择磁吸，价格约为 35 元以上，低端采用磁吸设计，振动传感器 PCB 需要采用灌胶方式进行固定。下图给出一个某宝的外壳例子。我方协助提供

## 2 认证证书



苏州朗博校准检测有限公司  
Suzhou Lonbo Calibration and Testing Co., LTD

证书编号: 37SJ240716-8189  
Cer. No.

### 校准结果 (Calibration Results) :

1、外观及功能检查(Appearance and Function Check): 符合要求

2、测振仪幅值线性度 (Calibration of the amplitude linearity) :

频率(Hz)	加速度(mm/s)	显示值(mm/s)	相对误差(%)	允差(%)	扩展不确定度 (k=2)
160	10	10.1	1.0	±5.0	$U_{\text{rel}}=1.0\%$
160	15	15.2	1.3	±5.0	$U_{\text{rel}}=1.0\%$
160	20	20.3	1.5	±5.0	$U_{\text{rel}}=1.0\%$
160	25	25.3	1.2	±5.0	$U_{\text{rel}}=1.0\%$
160	30	30.4	1.3	±5.0	$U_{\text{rel}}=1.0\%$
160	35	35.5	1.4	±5.0	$U_{\text{rel}}=1.0\%$

3、位移幅值线性度 (Calibration of the amplitude linearity) :

频率(Hz)	位移(mm)	显示值(mm)	相对误差(%)	允差(%)	扩展不确定度 (k=2)
40	10	10.1	1.0	±3.0	$U_{\text{rel}}=2.0\%$
60	10	10.1	1.0	±3.0	$U_{\text{rel}}=2.0\%$
80	10	10.1	1.0	±3.0	$U_{\text{rel}}=2.0\%$
160	10	10.0	0.0	±3.0	$U_{\text{rel}}=2.0\%$

### 附(Appendix) :

关于测量结果不确定度的说明

Directions of measurement uncertainty in the calibration

1、依据: JJF 1059.1-2012 测量不确定度评定与表示

Reference document: JJF 1059.1-2012 Evaluation and Expression of Uncertainty  
in Measurement

-----以下空白 (hereafter no date) -----

图 2: 传感器校准证书

## 3 IOT6-Z2 专利算法

### 3.1 三轴加速度计数学模型

三轴加速度计输出模型可表示为:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{bmatrix} \quad (1)$$

其中:

- $a_x, a_y, a_z$  为传感器输出量 (单位: g 或 m/s<sup>2</sup>)
- $A_x, A_y, A_z$  为真实加速度 (单位: g 或 m/s<sup>2</sup>)
- R 为 3×3 旋转矩阵 (包含安装误差和轴间耦合)
- $b_x, b_y, b_z$  为零偏 (bias)
- $\eta_x, \eta_y, \eta_z$  为测量噪声 (通常建模为高斯白噪声)

## 3.2 静态模型 (重力场测量)

在静态情况下, 加速度计仅测量重力分量:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + b + \eta \quad (2)$$

其中  $g$  为当地重力加速度 (约 9.81 m/s<sup>2</sup>)。

## 3.3 六面校准法

通过六个正交位置的测量数据计算校准参数:

$$\min_{R, b} \sum_{k=1}^6 \|a_k - (Rg_k + b)\|^2 \quad (3)$$

其中  $g_k$  为第  $k$  个位置的理论重力向量 (如  $[0, 0, g]^T, [0, 0, -g]^T$  等)。

## 3.4 运动参数计算, 加速度积分

通过时间积分获得速度和位移:

$$v(t) = v_0 + \int_{t_0}^t a(\tau) d\tau \quad (4)$$

$$p(t) = p_0 + \int_{t_0}^t v(\tau) d\tau \quad (5)$$

数字实现采用梯形积分法:

$$v[n] = v[n-1] + \frac{T_s}{2}(a[n] + a[n-1]) \quad (6)$$

其中  $T_s$  为采样周期。

### 3.5 振动分析

加速度信号处理:

1. 有效值 (RMS):

$$a_{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (a[n] - \bar{a})^2} \quad (7)$$

2. 峰峰值:

$$a_{pp} = \max(a[n]) - \min(a[n]) \quad (8)$$

### 3.6 频域分析

对加速度信号进行 1024 点 FFT:

$$A(k) = \sum_{n=0}^{1023} a[n] e^{-j \frac{2\pi}{1024} kn}, \quad k = 0, 1, \dots, 1023 \quad (9)$$

计算功率谱密度:

$$P(k) = \frac{1}{1024} |A(k)|^2 \quad (10)$$

### 3.7 温度测量模型

温度传感器通常使用以下模型将 ADC 值转换为温度:

$$T = \frac{V_{out} - V_{offset}}{S} \quad (11)$$

其中:

- $T$  为温度值 (°)
- $V_{out}$  为传感器输出电压
- $V_{offset}$  为 0 时输出电压
- $S$  为传感器灵敏度 (mV/°)

### 3.8 振动测量模型

振动传感器通常输出加速度信号，通过积分可获得速度和位移：

1. 加速度积分获得速度：

$$v(t) = \int a(t)dt + v_0 \quad (12)$$

2. 速度积分获得位移：

$$d(t) = \int v(t)dt + d_0 \quad (13)$$

数字实现时使用梯形积分法：

$$v[n] = v[n-1] + \frac{T_s}{2}(a[n] + a[n-1]) \quad (14)$$

其中  $T_s$  为采样周期。

### 3.9 振动特征参数计算

1. 有效值 (RMS):

$$a_{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} a[n]^2} \quad (15)$$

2. 峰峰值：

$$a_{pp} = \max(a[n]) - \min(a[n]) \quad (16)$$

3. 峭度指标：

$$K = \frac{\frac{1}{N} \sum_{n=0}^{N-1} (a[n] - \bar{a})^4}{\left( \frac{1}{N} \sum_{n=0}^{N-1} (a[n] - \bar{a})^2 \right)^2} \quad (17)$$

### 3.10 温度补偿算法

振动传感器灵敏度常受温度影响，需进行补偿：

$$a_{comp} = a_{raw} \times (1 + \alpha(T - T_{ref})) \quad (18)$$

其中：

- $a_{comp}$  为补偿后加速度
- $a_{raw}$  为原始加速度
- $\alpha$  为温度系数
- $T_{ref}$  为参考温度

### 3.11 集成算法流程

- (a) 采集 1024 点温度和振动数据
- (b) 对振动数据应用窗函数 (如汉宁窗)
- (c) 执行 1024 点 FFT
- (d) 计算振动频谱特征
- (e) 读取温度数据并进行校准
- (f) 应用温度补偿到振动数据
- (g) 计算时域振动参数 (RMS, 峰峰值, 峭度等)
- (h) 输出结果

## 4 振动测试床测试结果，本 PCBA 输出

设备可以每隔 X 分钟输出此表格，表格在我方服务器 Web 方式提供接口。方便集成商集成。

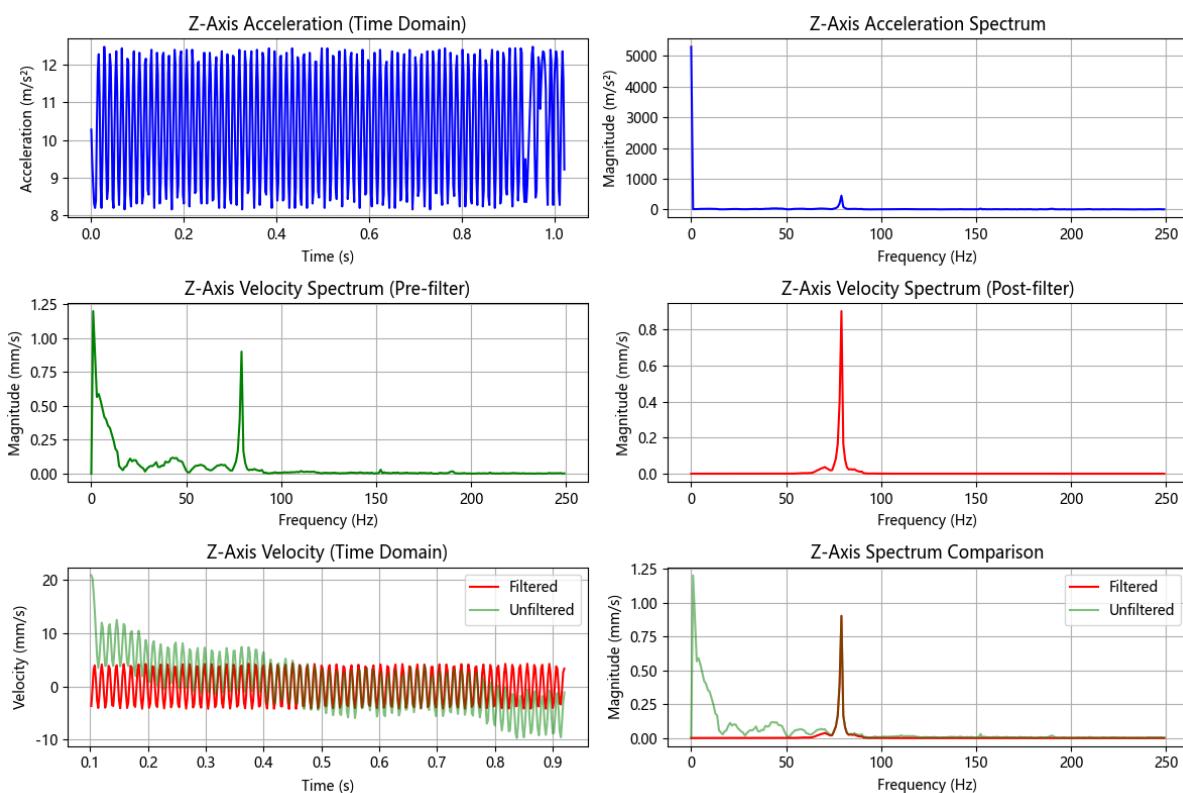


图 3: 振动台设置为 3mm/s 的输出，本 PCBA 输出

## 4.1 振动测量数据分析

表 1: 三轴振动参数测量结果 (主频 79.10 Hz 对比)

测量参数	X 轴	Y 轴	Z 轴
<b>加速度参数</b>			
主频 (Hz)	79.10	79.10	79.10
峰值 ( $m/s^2$ )	0.7061	0.2746	12.4741
峰峰值 ( $m/s^2$ )	0.4315	0.2354	4.3150
RMS ( $m/s^2$ )	0.4661	0.1639	10.4440
峭度	-0.85	-0.13	-1.50
<b>速度参数</b>			
滤波 RMS ( $mm/s$ )	0.16	0.06	2.96
未滤波 RMS ( $mm/s$ )	0.60	0.29	6.48
RMS 差值 ( $mm/s$ )	0.44	0.23	3.52

注: 所有测量均在相同工况下获取, 采样频率 1000 Hz, FFT 点数 1024。对于高频率采集, 可以选择高端 MCU。

## 4.2 数据分析结论

- 轴向对比:** Z 轴振动能量显著高于 X/Y 轴 (RMS 值达  $10.444 m/s^2$ , 是 X 轴的 22.4 倍)
- 峭度分析:** 所有轴向峭度均为负值 (Z 轴最低 -1.50), 表明振动信号分布较平坦
- 滤波影响:** 速度 RMS 差值在 Z 轴最大 ( $3.52 mm/s$ ), 显示高频成分最多
- 频率特征:** 三轴主频均为 79.10 Hz, 建议检查该频率对应的机械共振源

表 2: 各轴振动能量占比

轴向	能量占比 (%)
X 轴	4.2
Y 轴	1.5
Z 轴	94.3

## 5 主要 BOM 表格，不包括阻容，不同型号配置不同

Comment	Footprint	Pins	顶层数	元件立创编号
LIS2DH12TR	LGA-12	12	1	C10926
GS8331-TR	SOT-23_5P	5	1	C157712
SM712	SOT-23-3	3	1	C521963
TLC2254AIDR	SOIC16	16	1	C41754
SP3485EEN	SOP8	8	1	C8963
IIS3DWBTR	LGA-14L	14	1	C717702
LIS344ALHTR	LGA-16	16	1	C967634
TPS70933	SOT-23_5P	5	2	C89347
407VGT6	P-TQFP-100-1	100	1	C12345

## 6 应用接口

### 6.1 JSON 接口

为了方便集成，我们采用 JSON 方式进行参数设置，比如通过串口 3 下发如下参数。

- (a) 例子 1: { "zr": 30, "yr": 20, "xr": 30, "yz": 71, "fac": 55, "tmin": 120, "mg": 2, "pt100switch": 0, "pt100calib": 10019, "data50": 1, "data51": 1, "data52": 1, "data53": 1, "data54": 1, "data55": 1, "data56": 1, "data57": 1, "data58": 1, "data59": 1, "data60": 1, "data61": 1, "data62": 1, "data63": 1 }
- (b) 例子 2: { "data50": 1, "data51": 1, "data52": 1, "data53": 1, "data54": 1, "data55": 1, "data56": 1, "data57": 1, "data58": 1, "data59": 1, "data60": 1, "data61": 1, "data62": 1, "data63": 1 }
- (c) 例子 3: { "mg": 2, "pt100switch": 0, "pt100calib": 10019 }
- (d) 例子 4: { "data59": 1, "data60": 1, "data61": 1, "data62": 1, "data63": 1 }

下面对 JSON 进行详细描述

### 6.2 模拟部分参数

#### 1. 模拟振动传感器校准

表 3: 模拟信号配置参数

参数	类型	默认值	描述
zr	uint8_t	30	Z 轴电阻抽头值
yr	uint8_t	20	Y 轴电阻抽头值
xr	uint8_t	30	X 轴电阻抽头值
yz	uint8_t	71	模拟振动阈值
fac	uint8_t	55	模拟振动比率
tmin	uint8_t	120	定时唤醒间隔 (分钟)
mg	uint8_t	2	振动启动阈值
pt100switch	uint8_t	0	PT100 校准开关
pt100calib	int	10019	PT100 校准值 (单位:0.01)

### LIS344 三轴传感器

- 芯片型号: [LIS344ALH](#) (超高精度积分电路)
- 默认校准值: X 轴: 30 Y 轴: 20 Z 轴: 30
- 标准校准方法:
  - 使用振动测试仪设置 1mm/s 输出, 依次调整 zr/yr/xr 参数直到读数匹配, 输出是线性
- 快速校准方案:
  - 保持默认 zr/yr/xr 值不变, 仅需调整 `fac` 比率参数 (默认 55), 物理意义是电压 2.5V 输出 50mm/s

## 2. 无模拟传感器情况

### 硬件兼容性说明

- 若电路板未搭载 LIS344 传感器 (即将停产, 我司购买的时候 50 元, 现在嘉立创涨价到 200 元):
  - 忽略所有模拟传感器相关参数
  - 设置 `data56=0` 禁用模拟通道
- 可通过检查 PCB 版本号确认:
  - V2.1+: 标配 LIS344
  - V1.x: 无模拟传感器

### 3. PT100 温度校准

#### 铂电阻校准指南

- 校准准备：
  - 100标准电阻（精度 0.01）
  - 示例：100.19应输入10019
- 校准步骤：
  - i. 连接标准电阻到 PT100 接口
  - ii. 设置pt100\_switch=1启用校准模式
  - iii. 输入标准值pt100\_calib=10019，代表 100.19 欧姆
- 安全限制：
  - 最大校准值：101.00 (pt100\_max=10100)

### 4. 设备心跳时间

#### tmin 参数说明

- 功能定义：
  - 设备定时唤醒发送状态报告的时间间隔  
单位：分钟
- 参数特性：有效范围：1-255  
典型设置：120（2 小时）
- 特殊场景建议：
  - 一次性铁锂电池供电：建议 60-120 分钟，19000mAH 电池，4G 可以发送 3 万次
  - 太阳能供电：可设置 5-15 分钟

## 6.3 数字部分参数

### 数字配置参数详解

#### 1. data50：加速度计量程

表 4: 数字信号配置参数

参数	类型	默认值	描述
data50	uint8_t	1	1=16G, 2=2G
data52	uint8_t	1	功耗模式选择
data57	uint8_t	1	休眠振动阈值 (mm/s)
data59	uint8_t	1	发送串口 JSON 数据的间隔 (单位: 分钟)
data61	uint8_t	1	运行中原始数据发送间隔
data63	uint8_t	20	数据避免碰撞发送最小间隔秒

低速模式      高速模式

2G (data50=2)    16G (data50=1)

默认值: 1 (16G 量程)

## 2. data52: 功耗模式选择

模式 1 (data52=1)    微小功耗模式, 震动后检测 60 秒

模式 2 (data52=2)    微功耗, 震动后检测无震动立刻休眠

模式 3 (data52=3)    正常工作模式

默认值: 1 (微小功耗模式)

## 3. data57: 休眠振动阈值

单位: mm/s

连续 5 次采集小于此值进入休眠 (单位: mm/s), 大于这个数值, data63 秒内没有发送数据, 重新开始采集

典型值: 1 (1mm/s)

## 4. data59: 运行中 JSON 发送间隔

单位: 分钟 - 默认值: 1 (每分钟发送 1 次)

## 5. data61: 原始数据发送

启用 (data61=1) 发送 512CE3 字节 (XYZ 三包)

禁用 (data61=0) 不发送原始数据

默认值: 1 (启用)

## 6. data63: 数据避免碰撞发送最小间隔

单位: 秒 - 避免数据爆发的发送最小间隔 (单位: 秒) 默认值: 20

## 6.4 JSON 格式

设备日志如下:

```
Main Freq: X=79.10Hz Y=79.10Hz Z=79.10Hz
Accel Peak: X=0.6669 Y=0.5884 Z=11.2580 m/s2
Accel P-P: X=0.2354 Y=0.2746 Z=1.8828 m/s2
Accel RMS: X=0.5213 Y=0.4141 Z=10.3272 m/s2
Kurtosis: X=0.2383 Y=0.5540 Z=-1.4814
Velocity RMS: X=0.0419 Y=0.0272 Z=1.1763 mm/s
```

JSON发送数据如下: (保留2位小数)

```
{"id": "836", "b": 0, "pt": 22.99, "fx": 79.10, "fy": 79.10, "fz": 79.10,
"pkx": 0.67, "pky": 0.59, "pkz": 11.26, "p2px": 0.24, "p2py": 0.27, "p2pz": 1.88,
"amsx": 0.52, "amsy": 0.41, "amsz": 10.33, "ktsx": 0.24, "ktsy": 0.55, "ktsz": -1.48,
"rmsx": 0.04, "rmsy": 0.03, "rmsz": 1.18, "xdl": 0, "ydl": 0, "zdl": 0,
"mn": "869080078263029", "iccid": "898608331924C0058969"}
```

自主上报配置数据

```
{"id": "2682", "v": 26, "zr": 30, "yr": 20, "xr": 30, "yz": 71, "fac": 55, "tmin": 120,
"mg": 2, "data50": 1, "data51": 1, "data52": 1, "data53": 1, "data54": 1, "data55": 1,
"data56": 1, "data57": 1, "data58": 1, "data59": 1, "data60": 1, "data61": 1,
"data62": 1, "data63": 20}
```

自主上报GPS数据

```
{"sateCnt": 8, "altitude": 106.8000031, "lng": 116.1470795,
"imei": "869080078263029", "t": 1756622118, "speed": 0.62042,
"lat": 38.6640167, "id": 2682}
```

表 5: 三轴传感器数据转换说明

JSON 字段	物理量	单位	转换公式
<b>公共参数</b>			
pt	温度		原始值直接读取
b	电池电压	V	原始值/100
<b>X 轴参数</b>			
fx	主频	Hz	原始值直接读取
pkx	加速度峰值	m/s <sup>2</sup>	$1.001 \times pkx$
p2px	加速度峰峰值	m/s <sup>2</sup>	$1.001 \times p2px$
amsx	RMS 加速度	m/s <sup>2</sup>	$0.997 \times amsx$
ktsx	峭度	无单位	原始值直接读取
rmsx	滤波速度 RMS	mm/s	原始值直接读取
xdl	X 模拟传感器速度	mms	
<b>Y 轴参数</b>			
主频	Hz	原始值直接读取	
pk <sub>y</sub>	加速度峰值	m/s <sup>2</sup>	$1.002 \times pk_y$
p2py	加速度峰峰值	m/s <sup>2</sup>	$1.002 \times p2py$
ams <sub>y</sub>	RMS 加速度	m/s <sup>2</sup>	$0.998 \times ams_y$
kts <sub>y</sub>	峭度	无单位	原始值直接读取
rms <sub>y</sub>	滤波速度 RMS	mm/s	原始值直接读取
ydl	Y 模拟传感器	mms	
<b>Z 轴参数</b>			
f <sub>z</sub>	主频	Hz	原始值直接读取
pk <sub>z</sub>	加速度峰值	m/s <sup>2</sup>	$1.003 \times pk_z$
p2pz	加速度峰峰值	m/s <sup>2</sup>	$1.003 \times p2pz$
ams <sub>z</sub>	RMS 加速度	m/s <sup>2</sup>	$0.999 \times ams_z$
kts <sub>z</sub>	峭度	无单位	原始值直接读取
rms <sub>z</sub>	滤波速度 RMS	mm/s	原始值直接读取
zdl	Z 模拟传感器	mms	

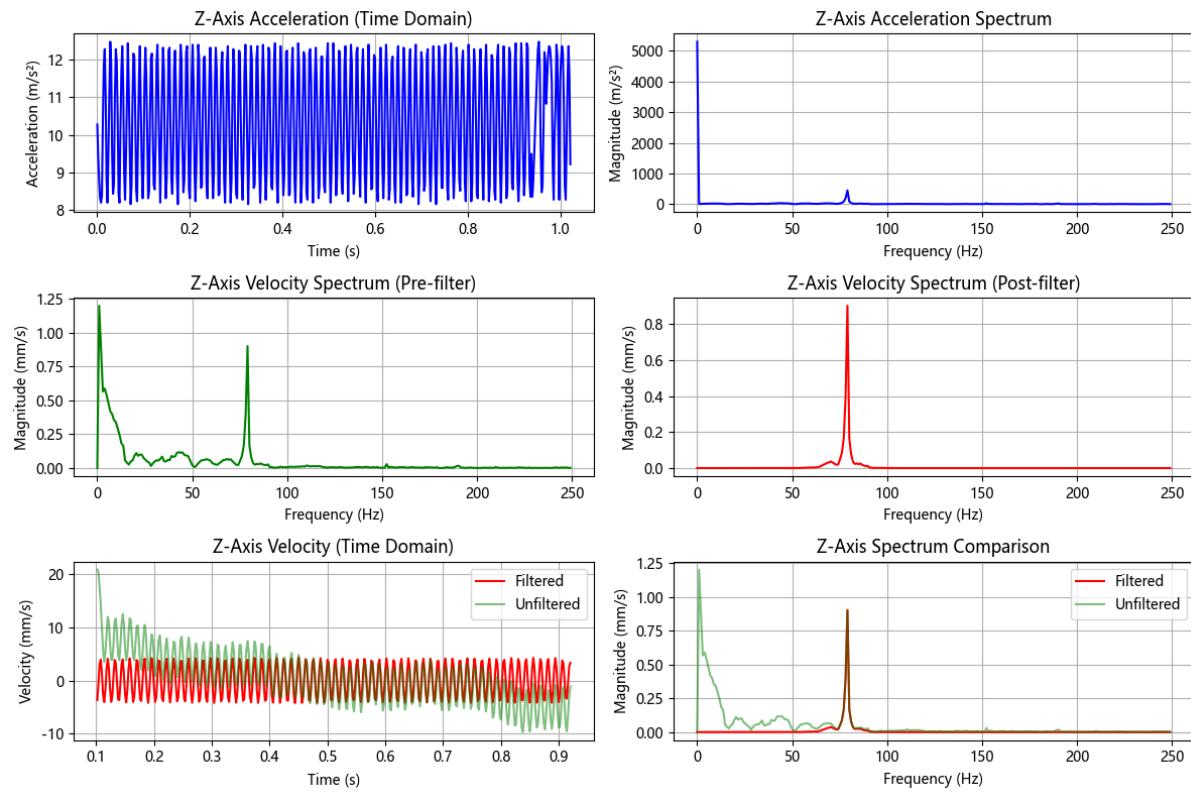


图 4: 振动台设置为 1.2mm/s 的输出, 本 PCBA 输出, 手机端可以查看

下面是Web端的日志, 在手机端展示的实时数据, web端的计算采用了志强CPU进行计算。

#### X-Axis Metrics:

Main Freq: 79.10 Hz  
 Accel Peak: 0.6669 m/s<sup>2</sup>  
 Accel P-P: 0.2354 m/s<sup>2</sup>  
 Accel RMS: 0.5213 m/s<sup>2</sup>  
 Kurtosis: 0.24  
 Filtered Velocity RMS: 0.04 mm/s  
 Unfiltered Velocity RMS: 0.39 mm/s  
 Difference: 0.34 mm/s

#### Y-Axis Metrics:

Main Freq: 79.10 Hz  
 Accel Peak: 0.5884 m/s<sup>2</sup>  
 Accel P-P: 0.2746 m/s<sup>2</sup>  
 Accel RMS: 0.4141 m/s<sup>2</sup>  
 Kurtosis: 0.55  
 Filtered Velocity RMS: 0.03 mm/s

Unfiltered Velocity RMS: 0.57 mm/s

Difference: 0.54 mm/s

Z-Axis Metrics:

Main Freq: 79.10 Hz

Accel Peak: 11.2580 m/s<sup>2</sup>

Accel P-P: 1.8828 m/s<sup>2</sup>

Accel RMS: 10.3272 m/s<sup>2</sup>

Kurtosis: -1.48

Filtered Velocity RMS: 1.19 mm/s

Unfiltered Velocity RMS: 2.35 mm/s

Difference: 1.16 mm/s

## 7 4G 模块和软件分析接口

### 7.1 基于 Luat 的微功耗 4G 模组，集成 GPS

配套的 4G 模块采用了 780EG Cat1+GPS 版本。PCBA 如下图所示。



图 5: Cat1, GPS 模组，内置 eSIM 卡

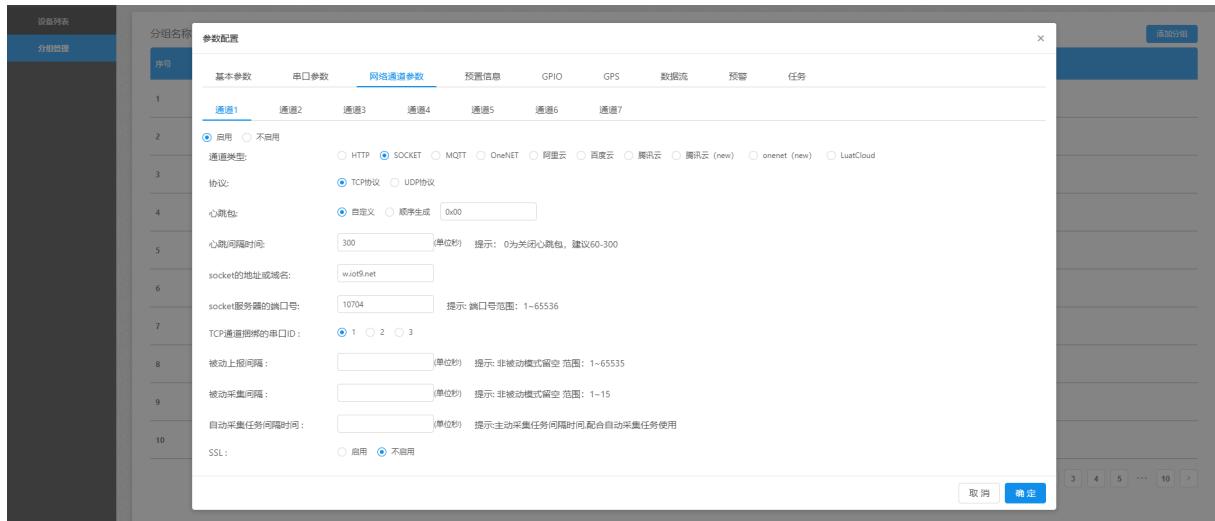


图 6: Web 配置, 支持 8 路协议

RT Thread设备端的报文封装

HEX 报文代码封装如下:

```
tempSZ[0] = 0x8E;
tempSZ[1] = 0x8E;
short value = 2; // 你要发送的值
tempSZ[2] = (value >> 8) & 0xFF; // 高字节
tempSZ[3] = value & 0xFF; // 低字节
tempSZ[4] = (jsonLength >> 8) & 0xFF;
tempSZ[5] = jsonLength & 0xFF ;
int totalLength = 6 + jsonLength;
trandata_uart2(tempSZ, totalLength);
```

//服务器端的解析代码

```
// ProtocolConstant.java
public interface ProtocolConstant {
    short MAGIC_NUMBER = (short) 0x8E8E;
    int BASE_LENGTH = 6;
    short TYPE_RAWX = (short) 0x03;
    short TYPE_RAWY = (short) 0x04;
    short TYPE_RAWZ = (short) 0x05;
    short TYPE_JSON = (short) 0x02;
}
import lombok.Data;
// RawDataPacket.java
```

```

@Data
public class RawDataPacket {
    private short xyz = 0;
    private final byte[] rawData;
    public RawDataPacket(byte[] rawData, short xyz) {
        this.xyz = xyz;
        this.rawData = rawData;
    }
}
// SmartProtocolDecoder.java
import io.netty.buffer.ByteBuf;
import io.netty.channel.ChannelHandlerContext;
import io.netty.handler.codec.ByteToMessageDecoder;
import java.nio.charset.StandardCharsets;
import java.util.List;
public class SmartProtocolDecoder extends ByteToMessageDecoder {
    @Override
    protected void decode(ChannelHandlerContext ctx, ByteBuf in,
    List<Object> out) throws Exception {
        if (in.readableBytes() < ProtocolConstant.BASE_LENGTH) {
            return;
        }
        System.err.println("in.readableBytes() 5010-->" + in.readableBytes());
        in.markReaderIndex();
        short magic = in.readShort();
        if (magic != ProtocolConstant.MAGIC_NUMBER) {
            in.resetReaderIndex();
            in.skipBytes(1);
            return;
        }
        short type = in.readShort();
        int dataLength = in.readShort();
        if (in.readableBytes() < dataLength) {
            in.resetReaderIndex();
            return;
        }
        byte[] data = new byte[dataLength];
    }
}

```

```
    in.readBytes(data);

    if (type == ProtocolConstant.TYPE_RAWX
    || type == ProtocolConstant.TYPE_RAWY
    || type == ProtocolConstant.TYPE_RAWZ)  {
        RawDataPacket packet = new RawDataPacket(data,type);
        out.add(packet);
    } else if (type == ProtocolConstant.TYPE_JSON)  {
        String jsonStr = new String(data, StandardCharsets.UTF_8);
        JsonPacket packet = new JsonPacket(jsonStr);
        out.add(packet);
    } else {
        ctx.close();
    }
}
```

## 7.2 Web 端接口展示



图 7: Web 接口 1.

## IOT6-Z2/微功耗/主频率/加速度峰值/加速度峰峰值/加速度/峭度/速度

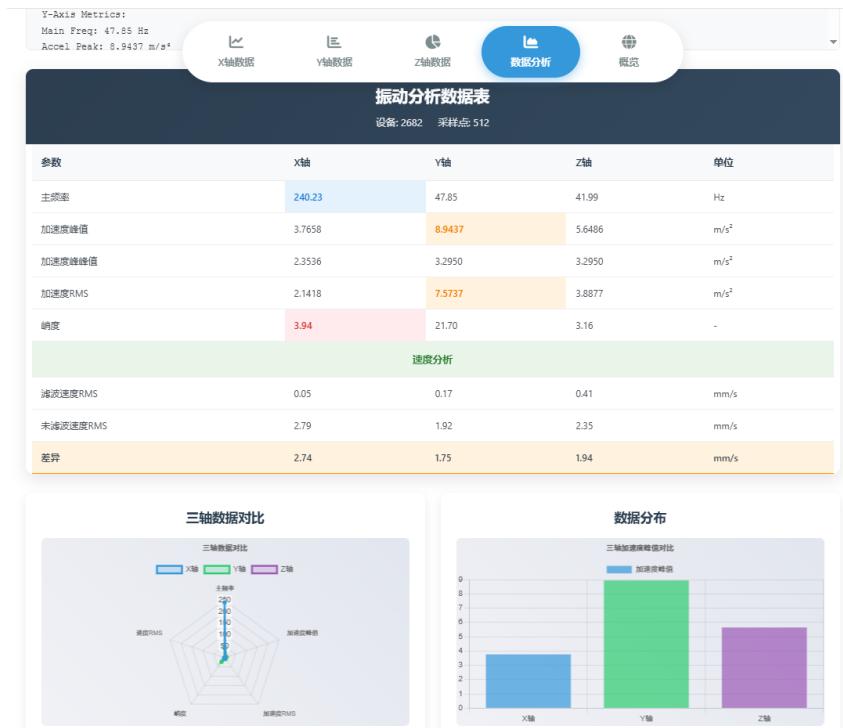


图 8: Web 接口 2.



图 9: Web 接口 3.

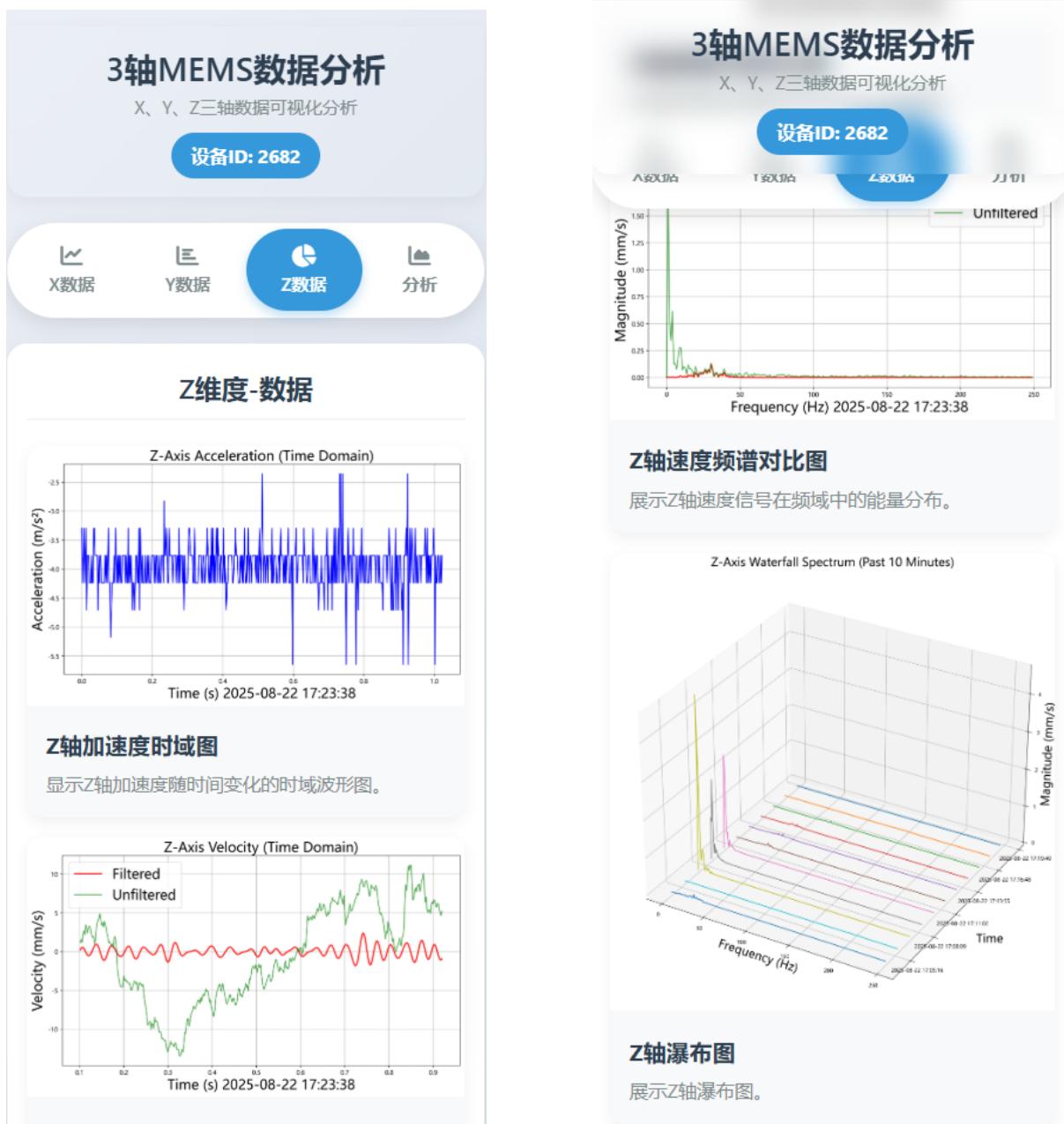


图 10: Web 接口 3

基于 HTTP 的 GET 接口，用于访问设备数据和分析结果。

网站首页

- URL: <http://172.iot6.net>
- 说明: 访问设备管理网站首页

移动端应用

- URL: <http://172.iot6.net/mobile.html?deviceid=2682>

- **参数:**

- `deviceid` - 设备 ID (示例: 2682)

Web 应用

- **URL:** `http://172.iot6.net/web.html?deviceid=2682`

- **参数:**

- `deviceid` - 设备 ID (示例: 2682)

数据接口

## JSON 数据接口

- **URL:** `http://172.iot6.net/api/rt?id=2682`

- **参数:**

- `id` - 设备 ID (示例: 2682)

- **返回格式:** JSON

GPS 数据接口

- **URL:** `http://172.iot6.net/api/gps?id=2682`

- **参数:**

- `id` - 设备 ID (示例: 2682)

- **返回格式:** GPS 数据

数据分析接口

X 轴数据分析

- 频谱对比图: `http://172.iot6.net/2682/X_spectrum_comparison.png`
- 加速度时间图: `http://172.iot6.net/2682/X_acceleration_time.png`
- 速度时间对比图: `http://172.iot6.net/2682/X_velocity_time_comparison.png`
- X 轴分析图: `http://172.iot6.net/2682/pbx.png`

Y 轴数据分析

- 频谱对比图: `http://172.iot6.net/2682/Y_spectrum_comparison.png`
- 加速度时间图: `http://172.iot6.net/2682/Y_acceleration_time.png`
- 速度时间对比图: `http://172.iot6.net/2682/Y_velocity_time_comparison.png`

- Y 轴分析图: <http://172.iot6.net/2682/pby.png>

## Z 轴数据分析

- 频谱对比图: [http://172.iot6.net/2682/Z\\_spectrum\\_comparison.png](http://172.iot6.net/2682/Z_spectrum_comparison.png)
- 加速度时间图: [http://172.iot6.net/2682/Z\\_acceleration\\_time.png](http://172.iot6.net/2682/Z_acceleration_time.png)
- 速度时间对比图: [http://172.iot6.net/2682/Z\\_velocity\\_time\\_comparison.png](http://172.iot6.net/2682/Z_velocity_time_comparison.png)
- Z 轴分析图: <http://172.iot6.net/2682/pbz.png>

## 8 Appendix

PCBA、嵌入式程序、OTA、Luat 客户端、服务器端、Java 以及 Web 程序，微信联系方式: tajamaya

## 参考文献

- [1] <https://so.szlcsc.com/global.html?k=LIS344&hot-key=GRM155R60J106ME44D/>
- [2] <https://so.szlcsc.com/global.html?k=STM32L496RGT6+LQFP-64>
- [3] <https://so.szlcsc.com/global.html?k=IIS3DWBTR&hot-key=LM258DT>
- [4] <https://docs.openluat.com/air780egh/product/>
- [5] <https://so.szlcsc.com/global.html?k=C41754&hot-key=AD7689BCPZRL7>
- [6] <https://docs.openluat.com/air780egh/product/>