

# 嵌入式工作室二轮招新-robotics方向

祝贺你们通过一轮考核！

顺利来到STM32二轮方向的考核~

## 前言

经过一轮的考核，相信大家已经学会诸多知识，成为一位点灯大师：

- 对嵌入式系统有了一定的见解
- 了解了MCU、外设、并行串行、同步异步、串口、GPIO、DMA、ADC...等基本概念
- 大致了解了stm32f103c8t6结构
- 拥有扎实的C语言基础
- 熟悉Keil5的基本操作
- 理解GPIO外设相关电路图和输入输出模式等概念，掌握其基本配置方法

此次二轮考核所制作的项目，是基于廖院长开设的“进阶式挑战性综合课程设计--多旋翼飞行器的设计与实现”课程中一个小挑战的简化版。大家在完成二轮挑战的过程中，不妨多思考一下加速度计、陀螺仪、蓝牙在一架无人机里的作用，如果有兴趣制作一架四轴飞行器，可以在此次二轮项目的基础上进行扩展。



## 要求

- 对于每个挑战点，撰写一篇 markdown 文档
  - 该挑战点所需知识的学习笔记。
  - 实现该挑战点的步骤讲解。(需要画时序图的地方可以使用此工具[draw.io](https://draw.io))
  - 记录挑战过程中的理解和感悟。

- (想象这是一篇将要发布到CSDN、知乎等平台的教学文章。愿意的话可以发到CSDN上面哦!)
- 对代码尽可能的理解, 体会各个步骤的用意
- 切忌copy, 我们都知道很多up主会提供源码, 尽可能自己学会后独立复现

## 项目提交

**2025年11月30日晚23:59:59前** 将以下资料**打包**发送到[13820992079@163.com](mailto:13820992079@163.com)

- 各挑战点的markdown文档(导出为PDF)
- 引脚配置说明图, 时序图 (附加题)
- 拍一段该挑战点的演示视频
- Keil5项目工程文件压缩包

**邮件标题&压缩包文件名: 姓名-robotics方向-第二期**

## 学习方法

“To learn by doing.”——廖院长

嵌入式覆盖面广, 既要了解硬件, 又要掌握软件。对于此类项目式学习, 并非是如同高中一般学好了所有必备知识, 再如同期末考试一般集中火力把项目制作出来。在项目开始前, 我们很难全面的了解需要的知识、没办法预测会遇到什么问题, 所以我们的学习应该是“**项目规划(需要哪些功能, 涉及哪些知识) -->做项目-->遇到问题或知识欠缺-->学-->回归项目**”这样的一个过程。嵌入式绝不是可以纸上谈兵的方向, 要在做中学, 实践出真知, 在实操中体会硬件与软件交汇的魅力。

在二轮考核中, 我们将带领大家体验一遍“To learn by doing”的学习模式。该项目被划分为四个挑战点, 每个挑战点又有小挑战点。在每个挑战点中自顶向下分析一遍, 学习该挑战点需求的知识。在非常有限的时间内自主找到新知识, 并通过知识集成, 完成起初认为几乎不能完成的任务。

## 有疑惑别害羞(///N///)

“君子生非异也, 善假于物也。”--《劝学》

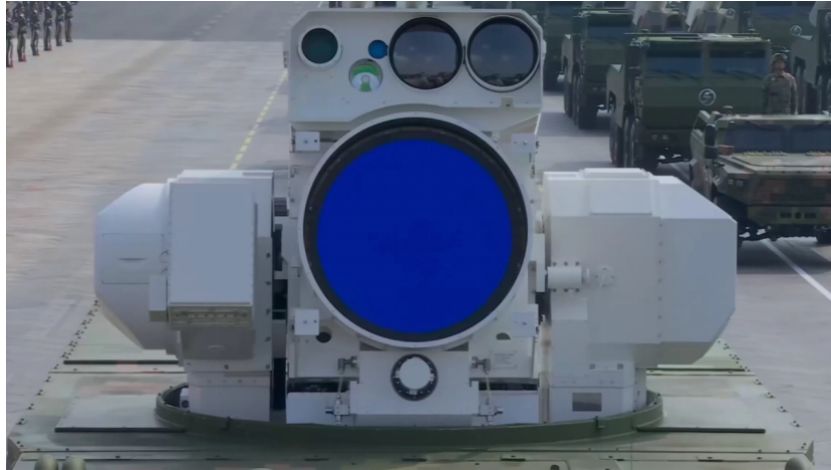
虽然二轮挑战难度较高, 但冷静分析, 划分成简单的小任务, 多问多思考, 一定能成功的~

- 多去B站看看教程
- 灵活使用搜索引擎和AI
- 学会去Google, Github等外网查找资料, 你甚至可以向卖家索要资料

有问题在招新群里提问, 或者私信出题人: 王晨博 蒋丰泽 张馨予 郭涵怡

# 引言

看过9.3阅兵，阿冉学长对出现的新式武器产生了浓厚的兴趣，其中尤其喜欢激光武器LY-1



阿冉学长仔细观察后，发现激光武器的朝向，主要由基座的两个转轴控制，阿冉学长想到能不能用舵机来自  
己设计一个可以遥控的激光武器的基座呢？

但控制舵机还要先学习一些知识，通过网上查照资料，阿冉学长找出了需要掌握的知识。

| doing    | 驱动舵机 | 信息发送接收 | 读取姿态    |
|----------|------|--------|---------|
| learning | PWM  | 蓝牙模块   | MPU6050 |

## 挑战点一：使用PWM驱动舵机

**舵机一定要会动**，👉👉👉👐😭👐😭👐😭👐😭，要动就要 PWM。

- 目标：**按键1**每次按下**舵机**顺时针旋转30度。超过180度则重置位置到0度
- 要求：加入**按键1**，并使用 PWM 控制**舵机**进行可控制转向。按下**按键1**产生**外部中断**，在**外部中断**里改变参数使**舵机**顺时针旋转30度，达到180度最大范围时位置回到零度。
- 问题：
  - 是什么产生的 PWM？PWM 的参数有哪些？分别有什么作用？
  - 为什么舵机要用 PWM 控制角度？PWM 的高电平持续时间与转角之间的关系是什么？
  - GPIO 需要配置成什么模式？这与普通 GPIO 输出模式的核心区别是什么？为什么这种模式适合 PWM 输出？
  - 假设舵机需要 50Hz 的 PWM 信号（周期 20ms），若 STM32 的定时器时钟为 72MHz，如何计算定时器的 ARR（自动重装载值）和 PSC（预分频系数）？试写出计算过程。
  - 按键按下时如何触发外部中断？从“引脚电平变化→EXTI 线检测→NVIC 中断响应”的流程，说明配置外部中断的关键步骤（可结合电路图中按键的连接方式分析）。

## 挑战点二：使用蓝牙模块进行信息的发送和接收

上位机（PC或手机）→蓝牙→OLED（搭建“交互桥梁”）

- 目标：上位机发送角度信息，HC05蓝牙模块接收并用于调整舵机角度，同时将角度显示在 OLED 屏幕上。
- 要求：通过AT指令配置蓝牙模块（可使用串口转TTL将蓝牙与PC连接），通过蓝牙模块实现STM32与PC（或手机）的连接。
- 问题：
  - 蓝牙模块（如 HC-05）与 STM32 之间通常采用 UART 通信，这种“串行异步通信”为什么适合远距离（相对 GPIO 直连）的数据传输？它通过哪些信号引脚实现数据收发？
  - 初次使用蓝牙模块时，如何通过 AT 指令配置波特率、设备名称？配置时需要注意哪些硬件细节？
  - 当蓝牙模块接收到数据时，为什么用“UART 接收中断”比“主循环轮询”更高效？中断服务函数中应如何处理数据？
  - 结合挑战点一，当蓝牙接收到“旋转舵机”指令后，如何复用之前的 PWM 控制逻辑实现转动，指令中的“角度参数”如何转化为 PWM 的占空比？

## 挑战点三：使用MPU6050读取姿态

我是谁，我在哪（位置），我要到哪里去（加速度），MPU6050 知道。

- 目标：使用 MPU6050 读取3轴加速度和角度数据。并使用 MPU6050 的加速度传感器计算横滚角和俯仰角，使用一阶滞后滤波平滑数据
- 问题：
  - MPU6050 采用 I2C 协议与 STM32 通信，这种“双线同步通信”（SCL/SDA）相比 UART，在多设备连接时有什么优势？如何通过地址引脚（AD0）区分多个 I2C 设备？

建议使用软件I2C，硬件I2C端口易被占用
  - 加速度计的 X、Y、Z 轴数据反映了什么物理量？为什么可以通过这些数据计算横滚角（Roll）和俯仰角（Pitch）？试推导计算公式（结合重力加速度在各轴的分量）。
  - 当设备运动时，加速度计的角度计算会出现较大误差，这是为什么？一阶滞后滤波（如  $filtered = \alpha * raw + (1-\alpha) * last\_filtered$ ）能如何改善这种误差？ $\alpha$  的取值大小对滤波效果有什么影响？
  - 从“MPU6050 寄存器地址→I2C 读写时序→数据解析”的流程，说明读取加速度数据的具体步骤。如何判断数据读取是否成功？
  - 若要在 OLED 上实时显示角度，如何将计算出的角度值（浮点数）转化为可显示的字符串？

## 挑战点四：整合上面的三个挑战点

终于要做完了，只剩最后一步了，做完了就可以给老登看看实力了😁😁😁

- 目标1：双模式舵机控制（一个舵机），通过pc端或手机端串口助手实现蓝牙控制舵机、MPU6050姿态解算得到俯仰角或者横滚角同步控制舵机角度（需注意输出角度不要超出舵机最大范围）。通过按键来切换蓝牙模式和MPU6050控制模式，用 OLED 屏幕显示当前模式（如“Bluetooth Mode”/“MPU6050 Mode”）
- 目标2：编写一份引脚选择说明图，并给出至少两个可选择版本。

不同模块可能共用 GPIO 或定时器资源（如 PWM 和 UART 可能共用同一定时器），建议在引脚说明中标注潜在冲突点及解决方案（更换定时器通道、复用引脚功能），帮助理解硬件设计的权衡思路。

恭喜你，到这里就已经完成了一个多途径控制的转向装置👏👏👏。想一想你的作品可以应用在哪里呢？

## 附加题：

现在有小登会问了，“🤖👉老登老登，你出的题太简单了，有更有挑战性的题吗？”有的兄弟，有的。为子折磨小登 为了学有余力的小登去实践自己的学习成果，我们精心挑选了一些附加题：

## 附加题一：配置寄存器

寄存器是单片机的基础，调库本质上是改变寄存器，那不妨通过直接配置寄存器点亮小灯。

- 目标：通过直接操作**寄存器**完成 GPIO 的配置
- 问题：
  - 要通过寄存器控制 GPIO，首先需要明确哪些核心信息？

可从“寄存器地址在哪找”“寄存器里的位对应什么功能”“GPIO 工作需要哪些外设支持”三个角度思考，比如参考 STM32F103 中文手册的“存储器映射”和“GPIO 寄存器描述”章节

- 若小灯点亮后不亮，可能的硬件或寄存器配置问题有哪些？

可通过“在 Keil 调试模式下观察寄存器地址的值”排查问题，比如时钟未使能、模式配置错误、引脚电平方向反了

## 附加题二：DMA 传输优化“系统效率”

- 目标：每隔1s读取 MPU6050 的加速度数据，并通过**蓝牙模块**发送给，数据显示在电脑上。

- 要求：使用 DMA 将 MPU6050 的数据存储，再通过另一个 DMA 通道将数据搬运到 USART 的发送数据寄存器。通过 TIM 定时产生中断
- 问题：
  - 当需要每隔 1 秒读取 MPU6050 数据并通过蓝牙发送时，若用 CPU 直接操作（轮询或中断），会对系统效率产生什么影响？DMA 为什么能解决这个问题（从“CPU 干预”的角度分析）？
  - 设计两个 DMA 通道分别完成“MPU6050→内存”和“内存→USART 发送寄存器”的传输，试说明每个通道的“源地址”“目标地址”“传输方向”应如何配置？
  - DMA 传输完成后，如何通知 CPU 进行后续处理（如数据校验、触发下一次传输）？这需要配置 DMA 的哪种中断？
  - 定时器如何实现 1 秒定时触发？从“时钟源选择→PSC/ARR 配置→更新中断使能”的步骤，说明定时逻辑的实现。
  - 结合前三个挑战点，DMA、定时器、UART、I2C 之间如何配合工作？试画出整个数据流程的时序图（从定时器触发→DMA 读 MPU6050→DMA 发蓝牙→等待下一次触发）。

## 附加题三：使用STM32读取MPU6050三轴加速度并通过蓝牙无线控制舵机云台

太好了，你学会了PWM驱动舵机，蓝牙模块通讯，MPU6050传感器，现在让我们开始造LY-4造云台吧

- 目标：
  - 执行端：一个最小系统板配备蓝牙接收和舵机
  - 控制端：另一个系统板配备mpu6050和蓝牙发送
  - 二维云台：mpu6050横滚角控制云台下面的舵机转角，俯仰角控制上面的舵机。
  - 使用DMA搬运接执行端收到的数据。
  - 使用卡尔曼滤波替代一阶滞后滤波
  - 使用mpu6050的三个加速度值和三个陀螺仪值，计算横滚角、偏航角和俯仰角，偏航角控制下面的舵机，俯仰角控制上面的舵机



- 参考视频: [如何使用STM32读取MPU6050三轴加速度并通过蓝牙无线控制舵机云台? 而且还很丝滑!](#)
- 补充: 此题需要较好的学习能力和动手能力, 需要的综合能力比较强, 能完成者可以尝试复现, 如果觉得较难复现了解原理, 明白是怎样工作即可。(小登加油👉)

## 附加题四: 手写mpu6050四元数姿态解算

InvenSense 官方提供了 MPU6050 DMP 库 (常见是 `inv_mpu.h`、`inv_mpu_dmp_motion_driver.h`), 以便我们直接调取DMP解算好的四元数。调库谁不会啊, 试试自己写一个

- 目标: 不用 DMP 库, 采用 **Madgwick/Mahony 算法**, 利用 MPU6050 的 **陀螺仪+加速度计** 数据, 实时解算四元数并得到欧拉角。
- 要求:
  1. **数据采集**: 通过 I2C 读取 MPU6050 的加速度和陀螺仪数据。
  2. **预处理**: 对加速度数据归一化, 消除量纲差异。
  3. **四元数更新**:
    - 利用陀螺仪角速度积分预测四元数;
    - 使用加速度 (重力方向) 作为参考修正误差。
  4. **归一化处理**: 保证四元数模长为 1。
  5. **姿态输出**: 将四元数转换为欧拉角 (Roll/Pitch/Yaw) 。

- 此处提供一份由gpt生成的伪代码：

```
1 // ahrs_mpu.c
2 // Madgwick & Mahony IMU-only implementations for MPU6050
3 // Usage: call *_updateIMU(...) each sensor sample with correct dt (s).
4 #include <math.h>
5 #include <stdint.h>
6
7 // ----- 配置 -----
8 #define ACC_LSB_2G 16384.0f // ±2g
9 #define GYRO_LSB_250DPS 131.0f // ±250 °/s
10 #define PI 3.14159265358979323846f
11 static const float deg2rad = PI / 180.0f;
12 static const float rad2deg = 180.0f / PI;
13
14 // ----- 状态（四元数） -----
15 static float q0 = 1.0f, q1 = 0.0f, q2 = 0.0f, q3 = 0.0f; // w,x,y,z
16
17 // ----- Gyro bias -----
18 static float gyro_bias_x = 0.0f, gyro_bias_y = 0.0f, gyro_bias_z = 0.0f;
19
20 // ----- Madgwick 参数 -----
21 static float madgwick_beta = 0.1f; // 推荐 0.01 ~ 0.5 视噪声和采样率
22
23 // ----- Mahony 参数 -----
24 static float mahony_twoKp = 2.0f * 0.5f; // 2*Kp
25 static float mahony_twoKi = 2.0f * 0.0f; // 2*Ki
26 static float mahony_integralFBx = 0, mahony_integralFBy = 0,
mahony_integralFBz = 0;
27
28 // ----- 工具函数 -----
29 static inline float invSqrt(float x) {
30     if (x <= 0.0f) return 0.0f;
31     return 1.0f / sqrtf(x);
32 }
33
34 // ----- 校准函数：静止时调用多次平均 -----
35 void gyro_calibrate_from_raw(const int16_t *gx_samples, const int16_t
*gy_samples, const int16_t *gz_samples, int n) {
36     if (n <= 0) return;
37     long sx=0, sy=0, sz=0;
38     for (int i=0;i<n;i++) { sx += gx_samples[i]; sy += gy_samples[i]; sz +=
gz_samples[i]; }
39     float gx_avg = (float)sx / n;
40     float gy_avg = (float)sy / n;
41     float gz_avg = (float)sz / n;
42     // 转为 rad/s
43     gyro_bias_x = (gx_avg / GYRO_LSB_250DPS) * deg2rad;
44     gyro_bias_y = (gy_avg / GYRO_LSB_250DPS) * deg2rad;
45     gyro_bias_z = (gz_avg / GYRO_LSB_250DPS) * deg2rad;
46 }
47
```

```

48 // ----- Madgwick IMU-only 更新 -----
49 void MadgwickAHRSupdateIMU(float gx, float gy, float gz, float ax, float ay,
float az, float dt) {
50     // 输入: gx,gy,gz in rad/s (已减去 bias), ax,ay,az in g (或任意, 但会归一化),
dt in s
51     float q0l=q0, q1l=q1, q2l=q2, q3l=q3;
52     // 1. 归一化加速度
53     float norm = ax*ax + ay*ay + az*az;
54     if (norm == 0.0f) return;
55     norm = invSqrt(norm);
56     ax *= norm; ay *= norm; az *= norm;
57
58     // 2. 四元数导数 (陀螺贡献)
59     float qDot0 = 0.5f * (-q1l * gx - q2l * gy - q3l * gz);
60     float qDot1 = 0.5f * ( q0l * gx + q2l * gz - q3l * gy);
61     float qDot2 = 0.5f * ( q0l * gy - q1l * gz + q3l * gx);
62     float qDot3 = 0.5f * ( q0l * gz + q1l * gy - q2l * gx);
63
64     // 3. 梯度下降校正项 (Madgwick 原文 IMU-only 简化形式)
65     float _2q0 = 2.0f*q0l, _2q1 = 2.0f*q1l, _2q2 = 2.0f*q2l, _2q3 = 2.0f*q3l;
66     float _4q0 = 4.0f*q0l, _4q1 = 4.0f*q1l, _4q2 = 4.0f*q2l;
67     float _8q1 = 8.0f*q1l, _8q2 = 8.0f*q2l;
68     float q0q0 = q0l*q0l, q1q1 = q1l*q1l, q2q2 = q2l*q2l, q3q3 = q3l*q3l;
69
70     // objective function gradient (常见实现)
71     float s0 = _4q0 * q2q2 + _2q2 * ax + _4q0 * q1q1 - _2q1 * ay;
72     float s1 = _4q1 * q3q3 - _2q3 * ax + 4.0f * q0q0 * q1l - _2q0 * ay - _4q1
+ _8q1 * q1q1 + _8q1 * q2q2 + _4q1 * az;
73     float s2 = 4.0f * q0q0 * q2l + _2q0 * ax + _4q2 * q3q3 - _2q3 * ay - _4q2
+ _8q2 * q1q1 + _8q2 * q2q2 + _4q2 * az;
74     float s3 = 4.0f * q1q1 * q3l - _2q1 * ax + 4.0f * q2q2 * q3l - _2q2 * ay;
75
76     // 归一化梯度
77     norm = s0*s0 + s1*s1 + s2*s2 + s3*s3;
78     if (norm > 0.0f) {
79         norm = invSqrt(norm);
80         s0 *= norm; s1 *= norm; s2 *= norm; s3 *= norm;
81     } else {
82         s0=s1=s2=s3=0.0f;
83     }
84
85     // 4. 应用反馈
86     qDot0 -= madgwick_beta * s0;
87     qDot1 -= madgwick_beta * s1;
88     qDot2 -= madgwick_beta * s2;
89     qDot3 -= madgwick_beta * s3;
90
91     // 5. 积分
92     q0l += qDot0 * dt;
93     q1l += qDot1 * dt;
94     q2l += qDot2 * dt;
95     q3l += qDot3 * dt;

```

```

96
97 // 6. 归一化四元数
98 norm = q01*q01 + q11*q11 + q21*q21 + q31*q31;
99 norm = invSqrt(norm);
100 q0 = q01*norm; q1 = q11*norm; q2 = q21*norm; q3 = q31*norm;
101 }
102
103 // ----- Mahony IMU-only 更新 -----
104 void MahonyAHRSupdateIMU(float gx, float gy, float gz, float ax, float ay,
105 float az, float dt) {
106     // gx,gy,gz in rad/s (bias removed), ax,ay,az in g (or any, will
107     normalize)
108     float recipNorm, halfvx, halfvy, halfvz, halfex, halfey, halfez;
109
110     // Normalize acc
111     recipNorm = ax*ax + ay*ay + az*az;
112     if (recipNorm == 0.0f) return;
113     recipNorm = invSqrt(recipNorm);
114     ax *= recipNorm; ay *= recipNorm; az *= recipNorm;
115
116     // estimated direction of gravity (from quaternion)
117     halfvx = q1*q3 - q0*q2;
118     halfvy = q0*q1 + q2*q3;
119     halfvz = q0*q0 - 0.5f + q3*q3;
120
121     // error = measured x estimated (cross product)
122     halfex = (ay*halfvz - az*halfvy);
123     halfey = (az*halfvx - ax*halfvz);
124     halfez = (ax*halfvy - ay*halfvx);
125
126     // integral
127     mahony_integralFBx += mahony_twoki * halfex * dt;
128     mahony_integralFBy += mahony_twoki * halfey * dt;
129     mahony_integralFBz += mahony_twoki * halfez * dt;
130
131     // apply feedback
132     gx += mahony_twokp * halfex + mahony_integralFBx;
133     gy += mahony_twokp * halfey + mahony_integralFBy;
134     gz += mahony_twokp * halfez + mahony_integralFBz;
135
136     // integrate quaternion rate
137     gx *= (0.5f * dt);
138     gy *= (0.5f * dt);
139     gz *= (0.5f * dt);
140     float qa=q0, qb=q1, qc=q2;
141     q0 += (-qb*gx - qc*gy - q3*gz);
142     q1 += (qa*gx + qc*gz - q3*gy);
143     q2 += (qa*gy - qb*gz + q3*gx);
144     q3 += (qa*gz + qb*gy - qc*gx);
145
146     // normalize quaternion
147     recipNorm = invSqrt(q0*q0 + q1*q1 + q2*q2 + q3*q3);

```

```

146     q0 *= recipNorm; q1 *= recipNorm; q2 *= recipNorm; q3 *= recipNorm;
147 }
148
149 // ----- 读取四元数 / 欧拉 -----
150 void getQuaternion(float *out_q0, float *out_q1, float *out_q2, float *out_q3)
151 {
152     *out_q0 = q0; *out_q1 = q1; *out_q2 = q2; *out_q3 = q3;
153 }
154 void getEulerDeg(float *roll, float *pitch, float *yaw) {
155     // roll (x), pitch (y), yaw (z)
156     float sinr = 2.0f * (q0*q1 + q2*q3);
157     float cosr = 1.0f - 2.0f * (q1*q1 + q2*q2);
158     *roll = atan2f(sinr, cosr) * rad2deg;
159
160     float sinp = 2.0f * (q0*q2 - q3*q1);
161     if (fabsf(sinp) >= 1.0f) *pitch = copysignf(90.0f, sinp);
162     else *pitch = asinf(sinp) * rad2deg;
163
164     float siny = 2.0f * (q0*q3 + q1*q2);
165     float cosy = 1.0f - 2.0f * (q2*q2 + q3*q3);
166     *yaw = atan2f(siny, cosy) * rad2deg;
167 }
168
169 // ----- 使用示例（伪代码，放在主循环或定时器回调） -----
170 /*
171 void sensor_task(float dt) {
172     // 1) 从 MPU6050 读 raw (int16_t)
173     int16_t axr = ...; int16_t ayr = ...; int16_t azr = ...;
174     int16_t gxr = ...; int16_t gyr = ...; int16_t gzh = ...;
175
176     // 2) 转单位
177     float ax = (float)axr / ACC_LSB_2G; // 单位 g
178     float ay = (float)ayr / ACC_LSB_2G;
179     float az = (float)azr / ACC_LSB_2G;
180
181     float gx = ((float)gxr / GYRO_LSB_250DPS) * deg2rad - gyro_bias_x;
182     float gy = ((float)gyr / GYRO_LSB_250DPS) * deg2rad - gyro_bias_y;
183     float gz = ((float)gzh / GYRO_LSB_250DPS) * deg2rad - gyro_bias_z;
184
185     // 3) 调用滤波器（任选其一）
186     MadgwickAHRSupdateIMU(gx, gy, gz, ax, ay, az, dt);
187     // 或
188     MahonyAHRSupdateIMU(gx, gy, gz, ax, ay, az, dt);
189
190     // 4) 取角度
191     float roll, pitch, yaw;
192     getEulerDeg(&roll, &pitch, &yaw);
193
194     // 5) 上报 / 发送（通过 DMA UART 等）
195 }
196 */

```

# 参考资料

---

[江科大STM32入门教程](#)

[keysking 超易懂的STM32教程](#)

[野火STM32F103标准外设库\(SPL\)开发教学视频](#)

[正点原子stm32入门](#)

[正点原子资料下载中心](#)

[意法半导体官网](#)

[stm32f103中文参考手册](#)

[MPU6050 datasheet](#)

希望大家不断提升自我的信息检索能力，学会自己找到相关的资料、教程、手册等。遇见问题，先经过自己的思考，多多在CSDN，Google，Github等网站查找解决方案，多问问学长学姐，拷打出题人。

## 硬件推荐

---

仅作参考学习推荐，不涉及任何商业性质；例图仅起辅助选择、价格参考的作用，不代表任何推荐店铺的行为。

可以参考江科大STM32开发套件的物品清单，包括STM32F103最小系统板、蓝牙模块HC-05，舵机，MPU6050，usb-ttl（CH340），面包板、OLED显示器、st-link烧录器、杜邦线也可以单独购买

天猫 | TOUGLESY “不贵且好用”

转动角度  
180度/360度

扭动稳定  
反应快

SG90



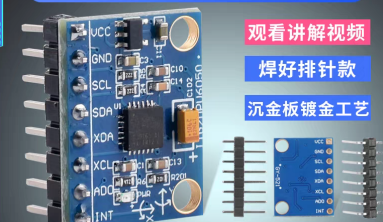
原厂供货 品质有保障 现货速发

天猫 | TOUGLESY 博主推荐

MPU6050舵机仪加速模块

GY-521模块 三维角度传感器 送资料+代码

观看讲解视频  
焊好排针款  
沉金板镀金工艺



厂家直销 量大详谈 现货速发 快速发货

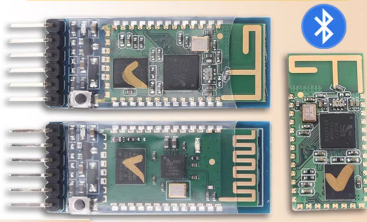
京源AI CH340串口模块

USB转TTL、单片机下载器



欣微 HC-05蓝牙串口透传模块

与蓝牙2.0设备连接 10米远程通讯连接

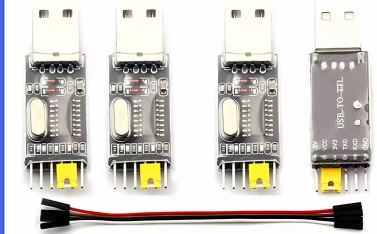


最大咨询客服优惠 提供网盘资料下载

DIY创作实验用蓝牙模块

USB转TTL CH340模块

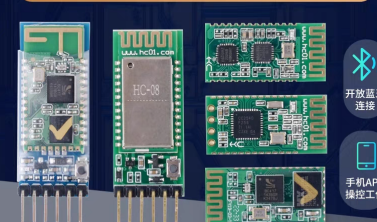
多种通信速率 稳定可靠 支持多个操作系统



专注元器件22年  
电子元器件全系列BOM表专业配单  
关注店铺+收藏商品优先发货

HC-05/2/6/8蓝牙串口模块

主从机一体 无线串口透传 功能稳定 多款可选



开放蓝牙连接  
手机APP操控工作

原厂供货 提供资料下载 现货速发

WANG HONG 0.96英寸OLED显示模块



颜色: 白色/蓝色/黄蓝双色 接口: SPI/IIC

尺寸: 0.96英寸 分辨率: 128\*64

驱动: SSD1306 字库: 中/英文

技术支持

STM8/STM32 仿真器

ST-LINK V2 STM8/STM32下载器线烧录器调试器

V2仿真器·支持win10系统·颜色随机发

送四根杜邦线



先领取详情页的优惠券再购物