

雾膜软件

WMH605 倾角仪开发板

版本 20240818

1. 概述

开发板的主要功能是输出三轴角度。

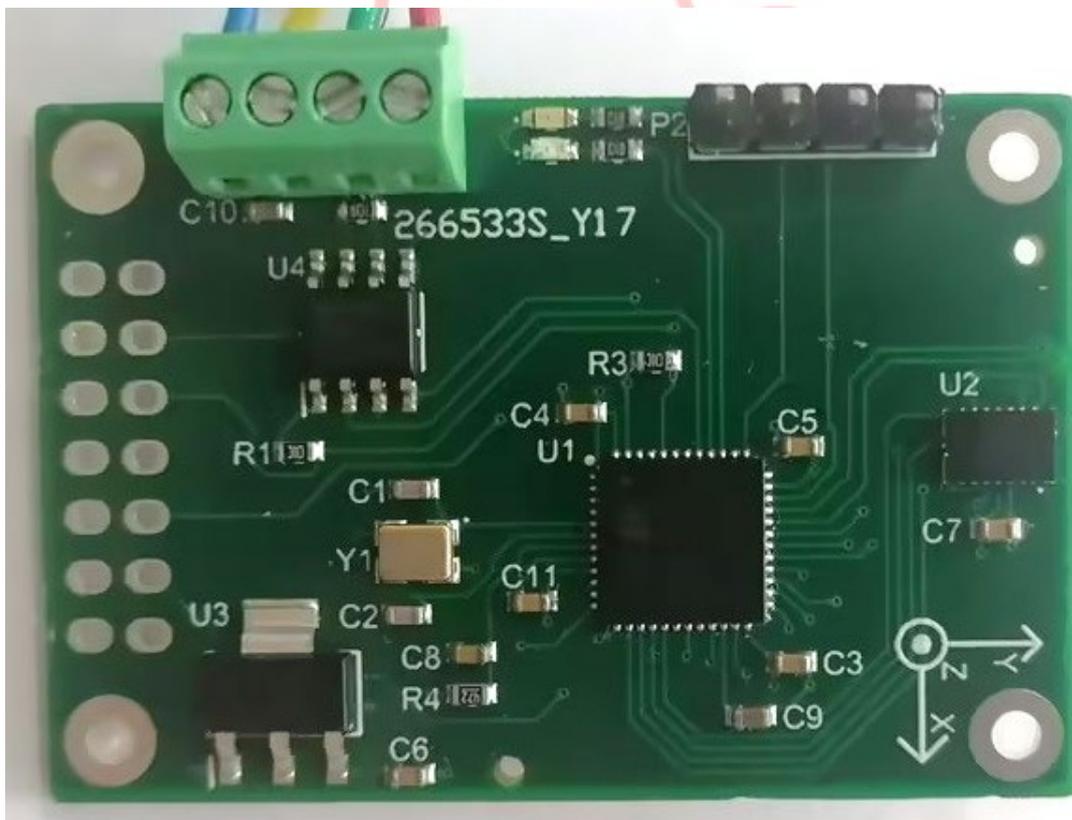
开发板组成包括：处理器 STM32F411CEU6，传感器 BMI055，电路附件。开发板附件包括串口-USB 转换器。

开发板提供示例代码。

主要特点：

- (1) 输出欧拉角，或四元数，或 6 轴原始数值。
- (2) 可在静止、振动、角振动、平稳运动的情况下工作。
- (3) 采用 float 型数据计算。
- (4) RS485 接口。

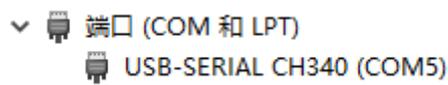
代码包适用于：组合导航的学习、研究、教学、科研。商业用途请联系本店另行定作。



2. 操作方法

2.1. 准备

串口 USB 转换器直接连接计算机。注意，有些机型需要安装串口 USB 转换器的驱动。如果驱动正常，在硬件管理器中应显示如下标志：



配置串口助手。注意串口号应当与硬件管理器一致；默认波特率 230400；其余配置参

照下图。



2.2. 读取数据

注意，上电 1~2 秒时，倾角仪会进行初始对准，此时应该保持基本静止状态。对准结束后，红灯转为绿灯，此时可以晃动倾角仪。

打开串口，即可读取数据。默认数据格式如下。

```
$-014.17,-088.12,-063.68,+0.6510,-0.6330,-0.3032,+0.2887,
```

依次为 yaw,pitch,roll,q0,q1,q2,q3，包含 3 个欧拉角和 4 个四元数。

2.3. 烧写程序

电路板发货时已经烧写默认程序，可以不烧写程序、直接使用。

需要烧写程序时，要准备的软件开发环境为：Keil uVision5；推荐同时使用 STM32CubeMX。需要准备硬件的仿真器，如 ST-LINK V2（需要自行购买，本店发货时不包含硬件仿真器）。

仿真器连接到开发板的 P2 口，注意引线对应关系。

源代码最好解压到纯英文路径中，使用 Keil uVision5 打开\MDK-ARM\qjy.uvprojx。然后点击 LOAD 即可烧写程序。



注意，烧写程序时，切勿选择擦除全部 FLASH；否则会导致配置失效、工作异常。

3. 性能

项目	数值	单位	备注
----	----	----	----

角速度量程	250	deg/s	最大 2000
加速度量程	2	g	最大 16
角度稳定性	0.5	deg	限 pitch 和 roll
航向稳定性	5	deg/min	低频振动
数据刷新率	200	Hz	可自行修改

本产品仅用于教学实验、演示工作原理，不保证恶劣环境下的精度。

默认代码具有静态检测功能，静态下三轴欧拉角基本稳定。默认代码具有传感器融合计算功能；动态环境下，pitch 和 roll 基本稳定，yaw 会缓慢漂移。

注意，运动过程中不能出现超量程情况，否则会导致精度下降。试验默认代码时，应当低速转动；需要高速转动时，应更改代码，改变传感器量程。

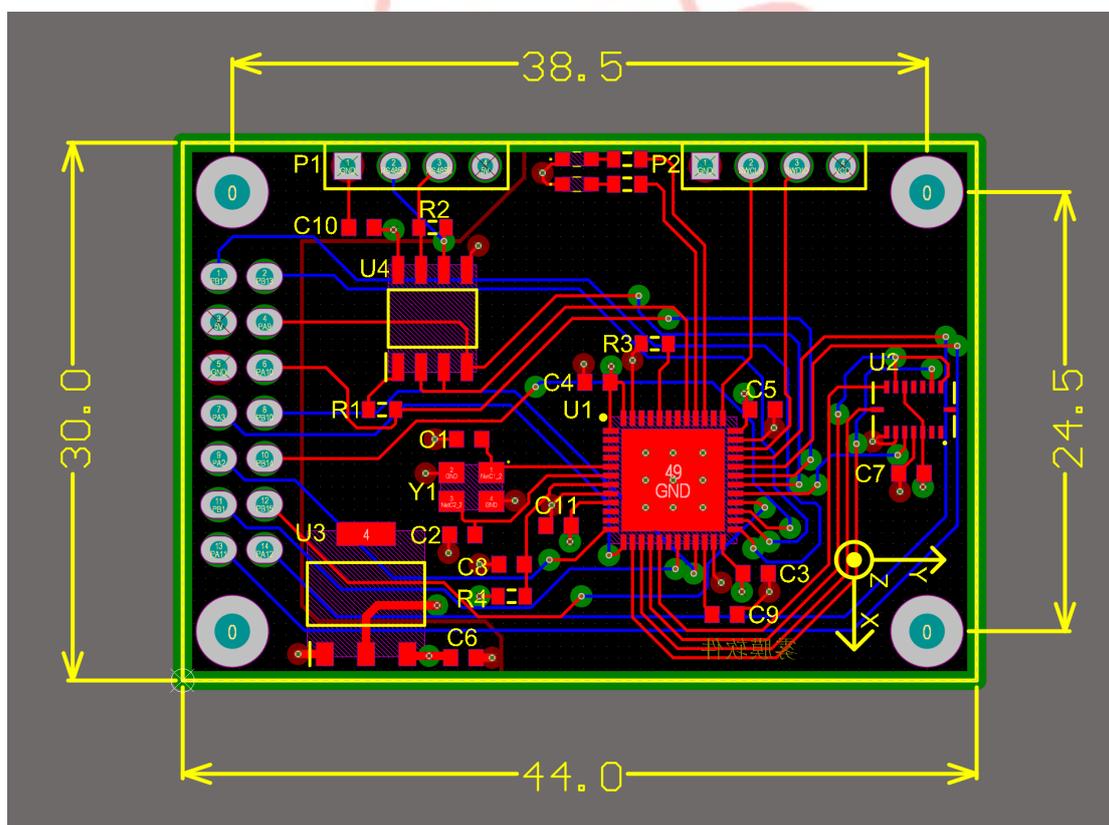
注意，yaw 为相对值，不是绝对值。本产品不具有寻北功能。

可在静止、振动、角振动、平稳运动的情况下工作。在离心运动、长时间加速等情况下，精度下降。

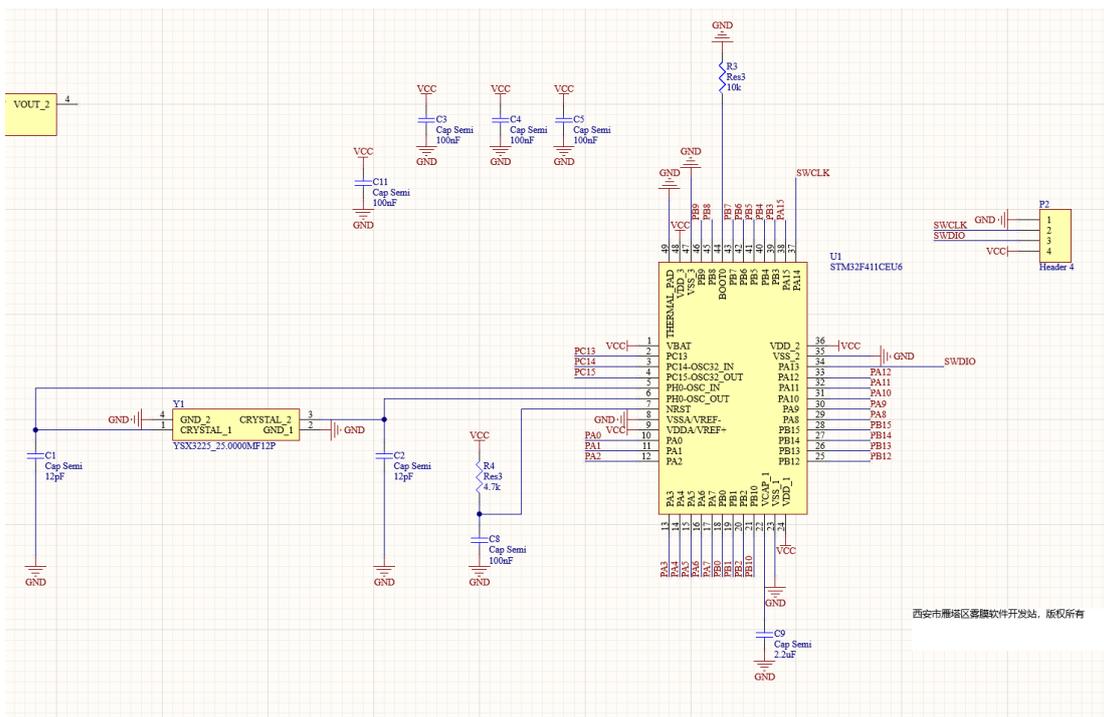
4. 外形和接口

4.1. 机械外形

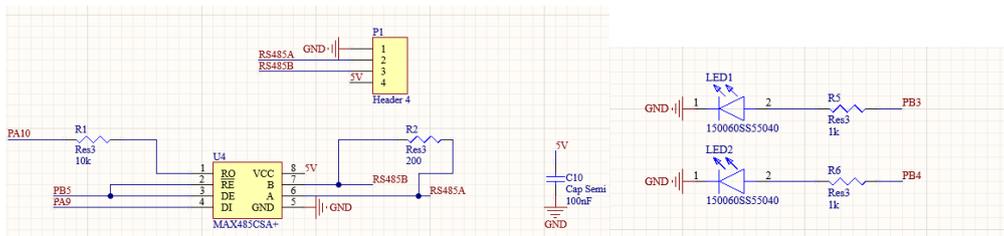
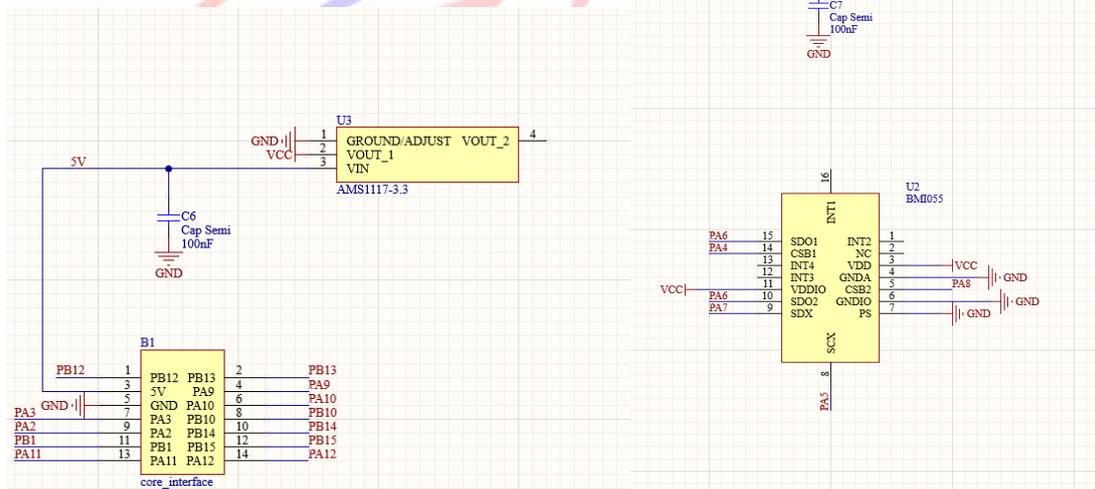
单位 mm。



4.2. 原理图

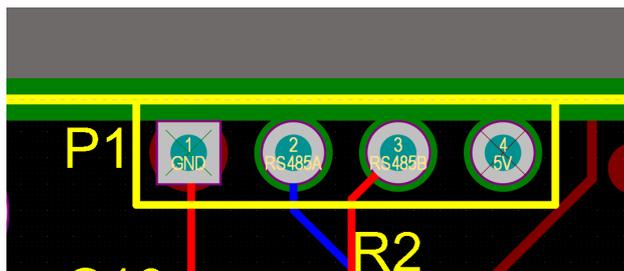


西安市雁塔区赛格软件开发站，版权所有

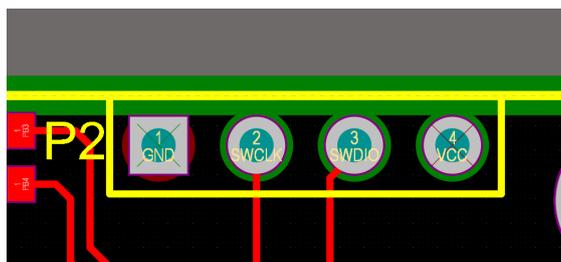


4. 3. 电气接口

RS485 接口如下：



程序烧写口如下:



扩展接口如下



4. 4. 通信协议

4. 4. 1. 协议配置

默认代码为字符串协议。如果要更改为字节流协议，需要更改代码：把 main()中 sendresult()改为 senddata1()。

4. 4. 2. 字符串

注意，使用字符串时，串口助手需要取消“十六进制显示”。

字符串协议形如：

```
$-014.17,-088.12,-063.68,+0.6510,-0.6330,-0.3032,+0.2887,
```

依次为 yaw,pitch,roll,q0,q1,q2,q3，包含 3 个欧拉角和 4 个四元数。每帧数据开头为\$，结尾为\r\n。

4.4.3. 字节流

注意，使用字节流时，串口助手需要勾选“十六进制显示”。

基本规则：（1）长整数（比如 32 位整数或 16 位整数），拆分为字节，低字节在前、高字节在后；但是文本或帧头按照表中的顺序排列，不再调换顺序。（2）数据除以比例系数为物理量的值；物理量的值乘以比例系数为数据。（3）一般数据为有符号数；但是帧头、状态指示、标志、校验码等为无符号数。

字节号	内容	比例系数	单位/备注
0~1	0xeb 0x90		帧头
2	计数		0~255 循环
3~4	陀螺仪 x	131.07	deg/s
5~6	陀螺仪 y	131.07	deg/s
7~8	陀螺仪 z	131.07	deg/s
9~10	加速度计 x	16383	g
11~12	加速度计 y	16383	g
13~14	加速度计 z	16383	g
15	校验码		2~14 字节求和

5. 姿态定义

5.1. 矩阵、欧拉角

有 3 种方法表示姿态：矩阵、四元数、欧拉角。

传感器 xyz 指向右、前、上，设为 b 坐标系。基准坐标系的 xy 在水平方向，设为 n 系。三维空间有 3 个旋转自由度。类似式(6-3)，依次绕三个坐标轴旋转，则坐标变换矩阵为

$$C_n^b = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5-1)$$

其中 C_n^b 就是表示姿态的矩阵，三个角度即欧拉角。上式即反映了矩阵和欧拉角之间的换算关系。注意，欧拉角多种不同定义，本文的定义可能与他人的定义不同。注意，三个角度是依次旋转的，不具有对称性、交换性。

注意，俯仰角在 90 deg 或 -90 deg 附近时，姿态角会突变。建议大角度时优先使用四元数表示姿态。

5.2. 四元数

实际导航系统中，为了防止计算误差导致姿态矩阵失去正交性，也为了减少计算量，往往采用四元数代替姿态矩阵进行姿态更新。四元数定义为

$$\mathbf{q} = \left[\cos \frac{\theta}{2} \quad u_x \sin \frac{\theta}{2} \quad u_y \sin \frac{\theta}{2} \quad u_z \sin \frac{\theta}{2} \right]^T \quad (5-2)$$

其中 θ 是旋转的角度， $[u_x \quad u_y \quad u_z]^T$ 是旋转轴的单位向量。

四元数也可以表示为

$$\mathbf{q} = \cos \frac{\theta}{2} + \mathbf{A} \sin \frac{\theta}{2} \quad (5-3)$$

其中 \mathbf{A} 是旋转轴的单位向量。

注意，四元数是按照三轴同时旋转定义的，不同于欧拉角的依次旋转。

6. 工作原理

6.1. 概述

程序主要工作流程如下：

初始化

```

解析对准
while(1)
{
    惯性导航;
    以速度为 0 作为条件, 计算扩展卡尔曼滤波, 得到惯性导航误差
    根据滤波结果修正惯性导航

    if(处于静态)
    {
        静态修正陀螺仪零偏
    }
}

```

本文采用扩展卡尔曼滤波方法, 实现传感器融合和姿态稳定。

6.2. 坐标系

载体系 b 定义为与载体固定连接的坐标系, 不妨取 xyz 轴为右前上。

导航坐标系 n 是表示导航结果的坐标系。

惯性参考系 i 。惯性参考系主要用于描述概念。惯性导航中一般不需要真正地在惯性参考系中投影, 所以不必在惯性参考系中规定坐标系。

完整地描述角速率、姿态、加速度、速度、位移等需要 3 个坐标系。坐标系 β 相对于坐标系 α 的变化量 x 在坐标系 γ 的投影表示为 $x_{\alpha\beta}^{\gamma}$ 。在这种表示方法下, 一些简单的计算规则如下:

同一个坐标系内表示的变量符合向量加法规则, 即

$$x_{AB}^{\gamma} + x_{BC}^{\gamma} = x_{AC}^{\gamma} \quad (6-1)$$

同一个变量在不同坐标系的换算可以用矩阵表示。

$$x_{\alpha\beta}^{\mu} = C_{\gamma}^{\mu} x_{\alpha\beta}^{\gamma} \quad (6-2)$$

坐标变换矩阵表示旋转关系。例如二维的坐标变换矩阵为

$$C_n^b = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (6-3)$$

三维的坐标旋转有 3 个自由度, 可以看作是类似形式矩阵相乘。

坐标变换矩阵是正交矩阵, 逆矩阵是原矩阵的转置

$$C_{\mu}^{\gamma} = (C_{\gamma}^{\mu})^{-1} = (C_{\gamma}^{\mu})^T \quad (6-4)$$

6.3. 惯性导航

6.3.1. 基本原理

惯性导航的基本原理是: 陀螺仪测量角速度, 角速度积分得到姿态。加速度计测量加速度, 加速度积分得到速度, 速度积分得到位置。

实际情况中有一些因素导致上述计算变得复杂。1.需要进行一些坐标系变换。2.需要考虑地球的自转、重力、以及球形形状。

6.3.2. 矩阵姿态更新

如果每次旋转的角度很小, 则坐标变换矩阵近似为

$$dC = \begin{bmatrix} 1 & 0 & -d\theta_y \\ 0 & 1 & 0 \\ d\theta_y & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & d\theta_x \\ 0 & -d\theta_x & 1 \end{bmatrix} \begin{bmatrix} 1 & d\theta_z & 0 \\ -d\theta_z & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6-5)$$

略去二阶小量, 则有

$$dC = \begin{bmatrix} 1 & d\theta_z & -d\theta_y \\ -d\theta_z & 1 & d\theta_x \\ d\theta_y & -d\theta_x & 1 \end{bmatrix} \quad (6-6)$$

上式表示了坐标旋力矩阵与旋转角度的关系。如果旋转角度很小，则不必考虑旋转顺序。为了表示的方便，引入角增量反对称矩阵

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \quad (6-7)$$

那么姿态矩阵更新公式为

$$\mathbf{C}_b^i(t+T) = \mathbf{C}_b^i(t) \lim_{k \rightarrow +\infty} \left(\mathbf{I} + \frac{[\boldsymbol{\theta}_b^i]}{k} \right)^k = \mathbf{C}_b^i(t) \exp([\boldsymbol{\theta}_b^i]) \quad (6-8)$$

其中 \exp 表示自然常数 e 为底数的指数函数。 $\mathbf{C}_b^i(t)$ 是上一时刻的姿态矩阵， $\mathbf{C}_b^i(t+T)$ 是下一时刻的姿态矩阵。上式即姿态更新公式。

利用麦克劳林公式，能得到更便于计算的如下公式

$$\exp([\boldsymbol{\theta}]) = \mathbf{I} + \frac{\sin|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|} [\boldsymbol{\theta}] + \frac{1 - \cos|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|^2} [\boldsymbol{\theta}]^2 \quad (6-9)$$

如果旋转角度较小，同时为了避免分母为 0，可以采用如下近似公式

$$\exp([\boldsymbol{\theta}]) \approx \mathbf{I} + [\boldsymbol{\theta}] \quad (6-10)$$

矩阵法仅用于演示原理。代码实际采用四元数法更新姿态。

6.3.3. 四元数姿态更新

四元数姿态微分方程为

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \mathbf{q} \quad (6-11)$$

引入 4 维的角增量矩阵

$$[\boldsymbol{\theta}] = \begin{bmatrix} 0 & -\theta_x & -\theta_y & -\theta_z \\ \theta_x & 0 & \theta_z & -\theta_y \\ \theta_y & -\theta_z & 0 & \theta_x \\ \theta_z & \theta_y & -\theta_x & 0 \end{bmatrix} \quad (6-12)$$

四元数更新姿态的公式为

$$\mathbf{q}(t+T) = \left(\cos \frac{|\boldsymbol{\theta}|}{2} \mathbf{I} + \frac{\sin \frac{|\boldsymbol{\theta}|}{2}}{|\boldsymbol{\theta}|} [\boldsymbol{\theta}] \right) \mathbf{q}(t) \quad (6-13)$$

姿态可以用 3*3 矩阵或者 4*1 的四元数表示。本代码包采用四元数计算姿态，这是主流方法。但是矩阵对于坐标系变换的计算比较方便，所以坐标变换的地方使用了矩阵表示姿态。

6.3.4. 速度更新

加速度计测量的加速度是 \mathbf{b} 系的，要把加速度换算为 \mathbf{n} 系，才可以用于计算速度。

对于局部直角坐标系，加速度积分即速度，速度积分即位置。

本代码不需要计算天向速度，只计算了水平方向的速度。

6.4. 组合导航

6.4.1. 原理概述

卡尔曼滤波可以理解为：根据方差求权重，做加权平均。

原始的卡尔曼滤波适用于线性系统。因为导航系统不是线性的，所以采用扩展卡尔曼滤波。扩展卡尔曼滤波的主要方法是，选用误差量，利用一阶微分近似为线性系统。滤波得到误差量估计值后，立刻补偿误差。

有的文献把 EKF 算法进一步细化为 ESKF 算法，严格意义上本代码包的方法属于 ESKF 算法。但是大量的文献没有把 EKF 算法进行如此细致的划分，本代码包的算法完全可以说就是 EKF 算法。

6.4.2. 卡尔曼滤波

比较复杂的系统中，一方面系统具有多个自由度，另一方面被测量随着时间而变化。因此用状态空间方程的形式描述系统的关系，并把加权平均数计算方法用矩阵表示，则得到卡尔曼滤波。

系统表示为：

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (6-14)$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k \quad (6-15)$$

其中是 \mathbf{x} 状态量，是希望获得而又难以准确测量的量。式(6-14)描述了被测量的变化关系，这里是离散形式。 \mathbf{z} 表示量测量，是能测量得到但是包含随机误差的量。式(6-15)描述了量测量与状态量的关系。 \mathbf{w} 和 \mathbf{v} 是随机噪声。有的系统中 \mathbf{w} 和 \mathbf{v} 会乘以系数矩阵，但是大多数惯性导航装置的三轴传感器精度大体相当，因此没必要引入标准卡尔曼滤波的 Γ 矩阵。

状态量的变化也可以描述为连续方程

$$\dot{\mathbf{x}}_k = \mathbf{F} \mathbf{x}_{k-1} \quad (6-16)$$

如果采样间隔足够小，离散方程与连续方程的关系为

$$\Phi = \mathbf{I} + \mathbf{F}T \quad (6-17)$$

其中 T 为采样间隔， \mathbf{I} 为单位矩阵。

卡尔曼滤波的解算过程就是根据 \mathbf{z} 估计 \mathbf{x} ，具体方法如下：

如果不考虑误差，前后时刻的 \mathbf{x} 具有关系

$$\hat{\mathbf{X}}_{k|k-1} = \Phi \hat{\mathbf{X}}_{k-1} \quad (6-18)$$

$\hat{\mathbf{X}}_{k-1}$ 是前一时刻 \mathbf{x} 的估计值， $\hat{\mathbf{X}}_{k|k-1}$ 是推算的后一时刻的 \mathbf{x} 。但是因为误差的存在，这个推算并不准确，需要根据 \mathbf{z} 修正，因此取

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{X}}_{k|k-1}) \quad (6-19)$$

其中 \mathbf{K}_k 是反映权重的滤波增益。这个增益由如下方法计算

$$\mathbf{P}_{k|k-1} = \Phi \mathbf{P}_{k-1} \Phi^T + \mathbf{Q} \quad (6-20)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R})^{-1} \quad (6-21)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (6-22)$$

其中 \mathbf{P} 、 \mathbf{Q} 、 \mathbf{R} 分别是 $\hat{\mathbf{X}}$ 、 \mathbf{w} 、 \mathbf{v} 的方差矩阵。

上述公式给出了线性系统的卡尔曼滤波方法。导航系统是非线性系统。非线性系统可以局部微分而近似为线性系统，采用扩展卡尔曼滤波方法解算。扩展卡曼滤波中的 \mathbf{x} 是误差量，扩展卡尔曼滤波获得误差量后，及时修正，使得误差量总维持在较小范围内；在误差量较小时，局部微分得到的线性系统与原始的非线性系统基本一致，卡尔曼滤波能取得较好效果。

代码包采用闭环反馈校正的方式，滤波后修正惯导误差，所以标准卡尔曼滤波中的 $\hat{\mathbf{X}}_{k-1}$ 取0，简化后的计算公式为

$$\hat{\mathbf{X}}_k = \mathbf{K}_k \mathbf{z}_k \quad (6-23)$$

6.5. 倾角仪的滤波

取扩展卡尔曼滤波的状态量 \mathbf{x} 为8维向量，包含速度误差*2、姿态误差*3、陀螺仪零偏

*3。

用扩展卡尔曼滤波进行组合导航的步骤是：1.进行惯性导航解算。2.惯性导航速度即 \mathbf{z} 。3.用卡尔曼滤波计算 \mathbf{x} 。4.根据 \mathbf{x} 修正惯性导航的结果，并返回步骤1。

惯性和卫星组合导航系统关键在于具体列出状态矩阵 Φ ，即可实现组合导航的计算。扩展卡尔曼滤波的矩阵 \mathbf{F} 是雅可比矩阵，即偏微分矩阵。根据惯性导航的计算公式，可以得到 \mathbf{F} 如下。

$$\mathbf{F} = \begin{bmatrix} \mathbf{O}_2 & \mathbf{F}_{av} & \mathbf{O} \\ \mathbf{O} & \mathbf{O}_3 & -\mathbf{C}_b^n \\ \mathbf{O} & \mathbf{O}_3 & \mathbf{O}_3 \end{bmatrix} \quad (6-24)$$

反映姿态误差对速度误差影响的子矩阵为

$$\mathbf{F}_{av} = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \end{bmatrix} \quad (6-25)$$

其中 f_E 、 f_N 、 f_U 是换算到 n 系的加速度计数值，即不扣除重力的比力信息。

$$\mathbf{f}_n = \begin{bmatrix} f_E \\ f_N \\ f_U \end{bmatrix} = \mathbf{C}_b^n \mathbf{f}_b \quad (6-26)$$

7. 代码包预览

7.1. 部分代码截图

（由于版本迭代，实际代码可能与截图有轻微差别）

```

int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration----- */

    /* Reset of all peripherals, Initializes the Flash interface
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_USART1_UART_Init();
    MX_SPI1_Init();
    MX_TIM2_Init();
    /* USER CODE BEGIN 2 */
    //initflash();
    initbmi();
    para();
    HAL_TIM_Base_Start_IT(&htim2);

    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
        if(timeflag)
        {
            timeflag=0;
            readbmi();
            calcoqjy();
            //senddatal();
            if(fr>=200)
            {
                HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOB,GPIO_PIN_3,GPIO_PIN_SET);
                sendresult();
            }
        }
    }
}
/* USER CODE END 3 */
}

```

```

ins_gyroacc(gx, gy, gz, ax, ay, az);
getPhi();

Zl = speedl;

Pkk = op_AaddB(op_AB(op_AB(Phi, Pk), op_AT(Phi)), Ql);
K = op_ArightB(op_AB(Pkk, op_AT(H)), op_AaddB(op_AB(op_AB(H, Pkk), op_AT(H)), R));
Xl = op_AB(K, Zl); //因为每次滤波之后补偿了误差, 所以状态预测总是0.
IKH = op_AsubB(Eye, op_AB(K, H));
Pk = op_AaddB(op_AB(op_AB(IKH, Pkk), op_AT(IKH)), op_AB(op_AB(K, R), op_AT(K)));

speedl = op_AsubB(speedl, submat(Xl, 0, 0, 2, 1));
attil = quatupdate(attil, op_AB((op_AT(Cbn(attil))), submat(Xl, 2, 0, 3, 1)));
biasgyro = op_AsubB(biasgyro, submat(Xl, 5, 0, 3, 1));

//补偿传感器误差。注意方向
gyrol=op_AaddB(gyro,biasgyro);

//一、计算姿态
attil=quatupdate(attil,op_kA(dTins,gyrol)); //更新姿态
cbnm = Cbn(attil);

//二、计算速度
accn=op_AB(cbnm,accl); //更新这个数, 以便于卡尔曼滤波的部分使用

speedl.num[0][0] = speedl.num[0][0] + dTins*accn.num[0][0];
speedl.num[1][0] = speedl.num[1][0] + dTins*accn.num[1][0];
//不计算重力和天向速度
}

```

7.2. 矩阵计算库

为了矩阵计算, C 代码中有结构体 MAT。其内容为:

```

int m; //行数
int n; //列数
double num[MAT_MAX][MAT_MAX]; //矩阵数据内容

```

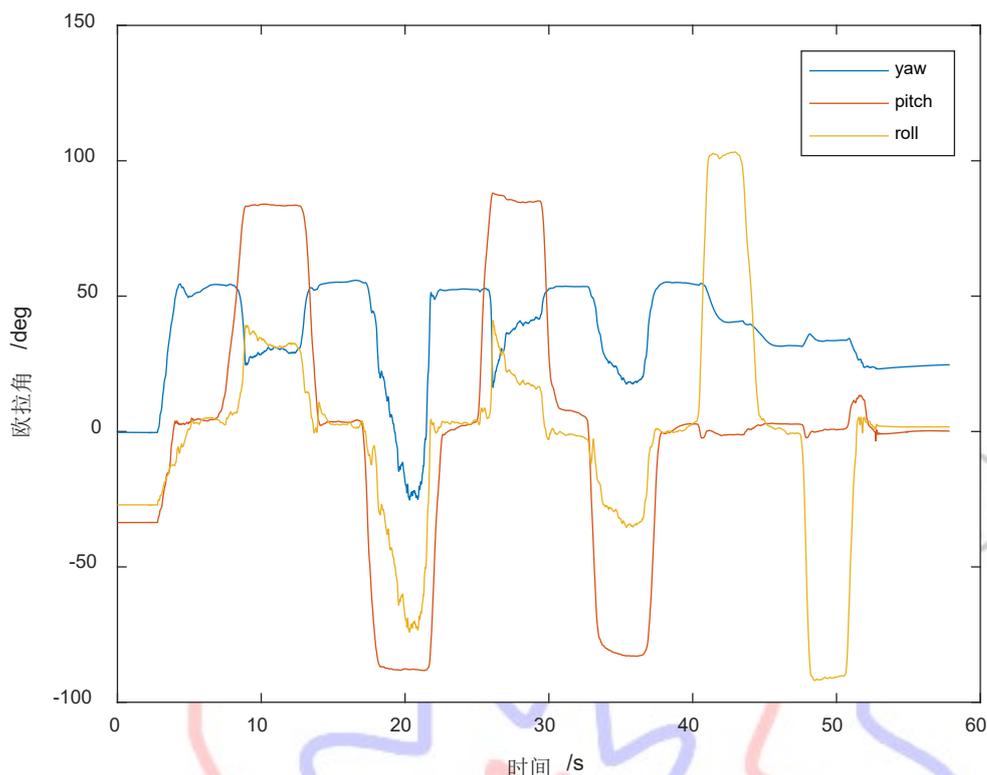
可以根据需要直接修改矩阵的数值。特别注意, C 或 C++ 中数组元素的下标从 0 开始; 而 MATLAB 的下标从 1 开始。

矩阵计算的功能已经包含在代码包中, 通常情况下不需要修改。

本代码包的矩阵运算是专门简化过的、另起炉灶的计算库, 与 Eigen、OpenCV 等常见矩阵库不兼容。

7.3. 输出数据曲线

徒手晃动, 输出数据为 txt 文件。借助 Matlab 绘制曲线, 三轴姿态角如下:



8. 常见问题

8.1. 代码容易看懂吗

本代码为了便于学习理解，力求简洁清楚。但是代码包技术含量较高，本店不保证用户能理解代码。用户可以参照前文的代码截图，预先评估自己是否可以理解代码包的内容。

为了防止非法转卖，代码包的注释比较简化。如有难以理解的部分，可以联系本店人工答疑。

用户可以自行修改代码，以调整功能。也可委托本店修改代码，但是需要额外费用。

8.2. 怎样标定

本产品传感器精度较低、稳定性较差，标定不具有实际意义。需要高精度的用户，请联系本店另行定做。

产品只有电路，缺乏可靠的基准面。如果要标定，需要额外配置机械外壳，以提供稳固的基准面。根据需求，采用温箱转台标定，或者采用稳固台面简易标定。

9. 著作权和服务

9.1. 工作原理参考什么资料

参考实体书《组合导航应用笔记》，东南大学出版社，2025年。

讲解视频，哔哩哔哩视频网搜索“大胡子刘师傅”。

9.2. 著作权声明

本店保留著作权。

电路、说明书、全部附属代码（以下简称本代码包）仅限于学习和研究用途的少量使用；包含改编文件、写入嵌入式系统的编译后程序，所有副本总计不得超过5份。

本代码包有偿使用。

严禁转卖或公开发布本代码包的全部或一部分。

大规模应用本代码包需要额外取得本店的授权。

对于违反上述要求的用户，本店有权要求停止销售、撤稿、赔偿损失等。

9.3. 服务内容

赠送 30 分钟语音答疑服务，用于解决较为复杂的疑问。

赠送长期文字答疑，用于解决简单的、零散的疑问。

答疑服务仅限直接购买人本人使用。答疑服务不能转让、不能共享。用户需要保留购买凭证截图；丢失购买凭证的，本店可以不提供答疑服务；不是从本店购买的，而是从其他渠道获得代码包的，不提供答疑服务。

本商品技术含量较高，本店不保证能在限时内解答所有疑问。有需要的用户，可以付费购买额外的语音答疑服务。

本店可提供少量的数据判读服务。但是大量的数据判读服务需要额外收费。较为复杂的数据处理，或者定制化修改代码，可能需要额外收费。

上述服务可能需要排队，本店不能保证服务的实时性。

9.4. 联系方式

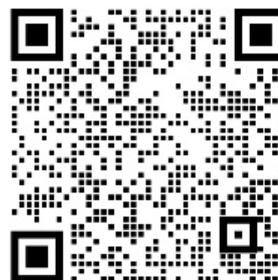
西安市雁塔区雾膜软件开发站

销售、答疑、定制开发：

微信：（扫码）



雾膜软件



电子邮箱：braun@wmsoft.wang

网站：<http://wmsoft.xyz>