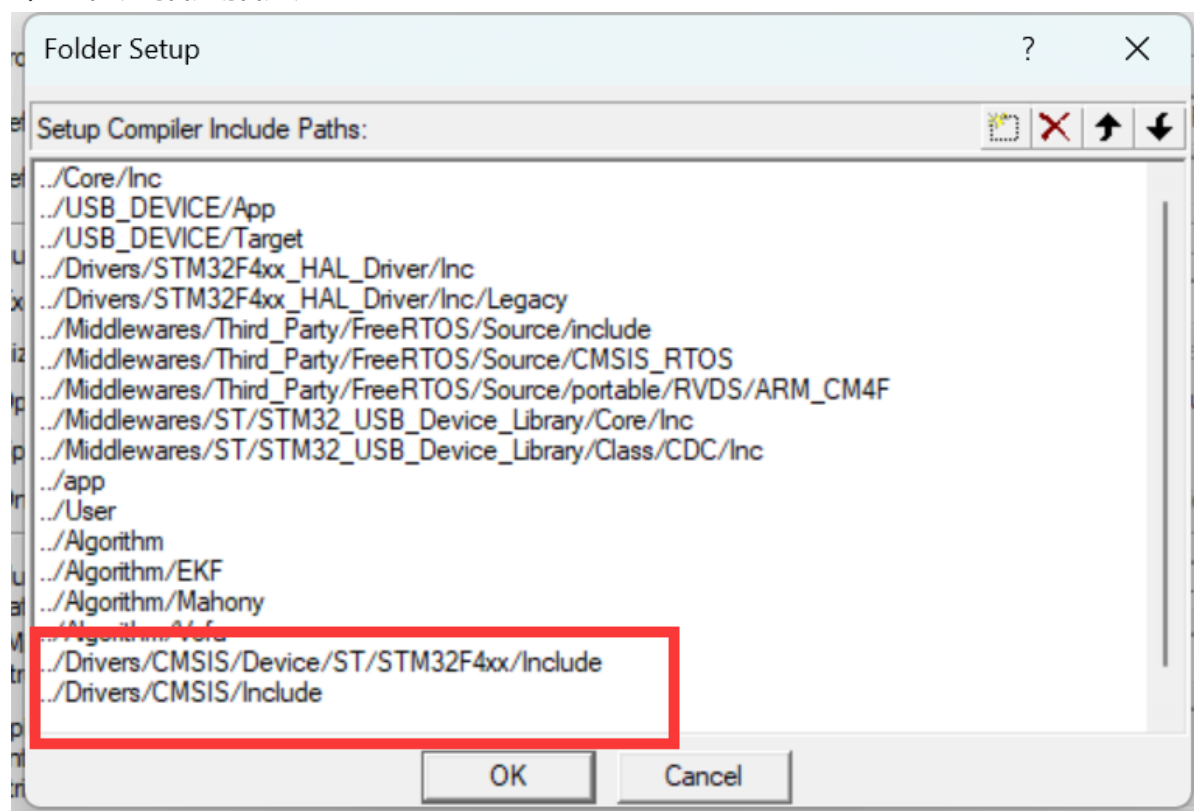


达妙F446开发板姿态解算

此惯导姿态解算算法EKF部分移植于：[WangHongxi2001/RoboMaster-C-Board-INS-Example \(github.com\)](https://github.com/WangHongxi2001/RoboMaster-C-Board-INS-Example)，Mahony部分我魔改了一部分。具体原理请看上面链接，同时还有开源的Mahony算法作为对比测试，开启了H7的cache作为优化，同时大量使用arm的dsp库进行优化。具体移植注意事项请看下文。

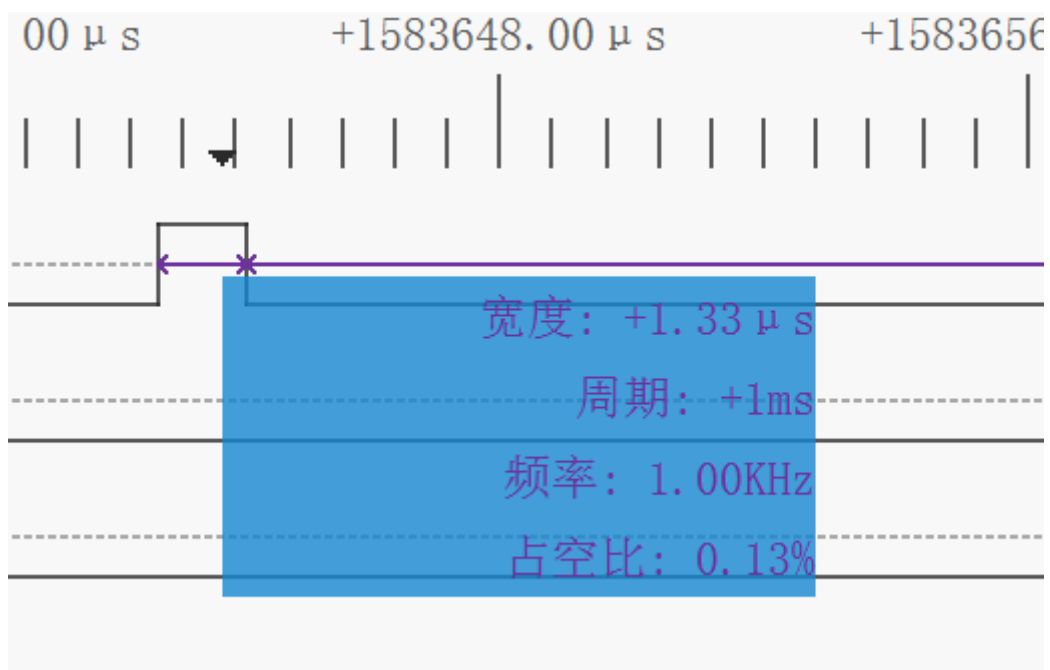
注意：

AC5的文件夹是使用AC5编译器的Code，AC6文件夹装着的是AC6的Code。需要注意的是AC6，如果使用cubemx重新生成后需要替换freertos中RVDS中的文件为armgcc中的文件，否则会导致编译失败。具体可以查阅相关资料。要注意如果是用cubemx重新生成了，需要将这里的两个path删除，才能正常编译，否则会出现arm_math重复引用等问题很玄学我不懂。希望有人指点指点

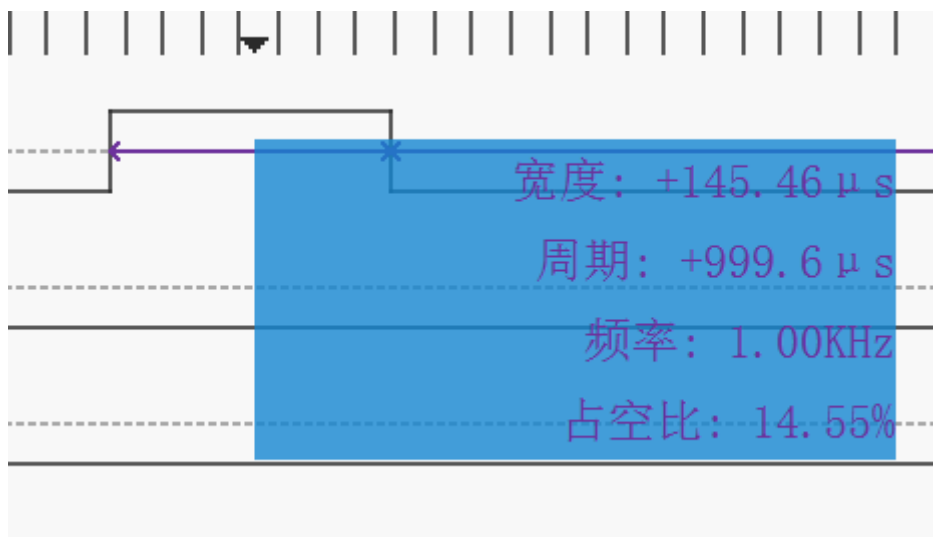


实际测试

开源Mahony算法-包含四元数转欧拉角的部分为1.33us,加四元数转欧拉角只需要520ns（这个是达妙H723的时间，f446的我懒得测了 23333）



开源EKF姿态解算算法-包含四元数转欧拉角的部分，总共为145.46us.



开发板购买链接

<https://item.taobao.com/item.htm?ft=t&id=767305853444>

控制板

• STM32H723VGT6主控 •

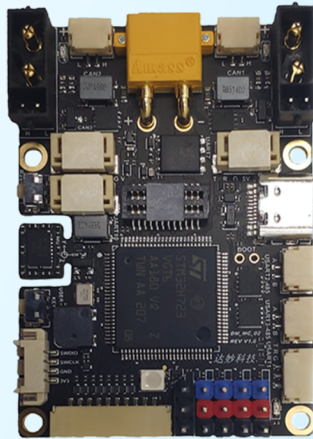
• 三路FDCAN两路485 •

• SBUS 4*PWM DCMI •

• 内置BMI088 •

• 多重防护 •

• 超高性价比 •



惊爆价 ¥199.00 当天发货 可开票

kit / DM-MC02

你当前开源项目尚未选择许可证 (LICENSE)，[点此选择并创建开源许可](#)

master 分支 1 标签 0

32 次提交

- 3D文件 update 5f2275c 19小时前
- image 修改说明和注册 17天前
- 例程 update 例程/Board-H7_W52812/Core/Src/main.c 21小时前
- 原理图 Fix labeling errors 5天前
- 数据手册 上传资料 19小时前
- README.en.md Initial commit 19小时前
- README.md update README.md 23小时前
- 达妙科技DM-MC-Board02电机开发板... update 18小时前

图 README.md

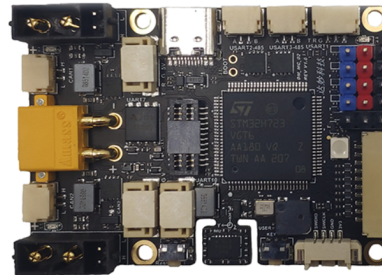
- DM_MC02
 - 介绍
 - 主控渲染图
 - 数据手册
 - 数据介绍
 - 购买链接

DM_MC02

介绍

达妙科技 DM_MC02 开发板资料

主控渲染图



MCU	STM32H723VGT6	
可控电源	x2	PMOS (XT30 2+2)
CANFD	x3	最大5Mbps
RS485	x2	最大10Mbps
USB	x1	
SBUS	x1	
PWM	x4	
串口	x3	
SWD	x1	
W2812彩灯	x1	
蜂鸣器	x1	
按键	x2	
IMU	x1	BMI088
LCD扩展口x1		SPI+IIC
QSPI Flash	8Mb	W25Q64JV
DCMI接口	x1	24Pin 无需转接
扩展接口	x1	2串口 1SPI 1IIC

达妙电机技术交流群
群号: 174204312

扫一扫二维码，加入群聊

QQ交流群

使用淘宝扫码购买

资料下载地址: <https://gitee.com/kit-miao/dm-mc02>

购买地址: <https://item.taobao.com/item.htm?ft=t&id=767305853444&skuld=5440731611044>

使用注意事项:

1. 上电后先进行恒温控制（温度控制我简单的调了一下PID，具体可以打开DEBUG进行设定，不同的电压PID可能稍有差别，需要对PID进行调整），当温度达到设定温度（40°），进行一个计数，当计数值达到阈值（目的是确保温度已经到40度附近）才进行到第二个状态

2.第二个状态，即attitude_flag==1，进行陀螺仪0飘初始化，此过程中需要保持开发板静止。初始化结束后进入第三个状态

3.attitude_flag==2，进行姿态解算

```
IMU_QuaternionEKF_Update(gyro[0],gyro[1],gyro[2],accel[0],
accel[1],accel[2]);//ekf姿态解算部分
//mahony姿态解算部分
//HAL_GPIO_WritePin(GPIOE,GPIO_PIN_13,GPIO_PIN_SET);
Mahony_update(gyro[0],gyro[1],gyro[2],accel[0],accel[1],ac
cel[2],0,0,0);
Mahony_computeAngles(); //角度计算
//HAL_GPIO_WritePin(GPIOE,GPIO_PIN_13,GPIO_PIN_RESET);
//=====
=====
//ekf获取姿态角度函数
pitch=Get_Pitch(); //获得pitch
roll=Get_Roll();//获得roll
yaw=Get_Yaw();//获得yaw
```

此代码中，cheat是通过一定的作弊手段，去掉了陀螺仪gyro[2]小的值从而使得yaw完全静止不太飘，如果应用场景角速度变化不明显建议去掉。

获得陀螺仪的pitch，roll，yaw通过调用函数。

四元数位于QEKF_INS.q 的数组中

串口显示曲线

默认开了一个线程通过USB CDC进行上位机数据发送，文件位于Algorithm.c。

上位机默认协议使用vofa的justfloat协议。vofa下载链接：[VOFA-Plus上位机 | VOFA-Plus上位机](#)

虚拟串口为自动波特率可以随意设置，自动识别。

发送的四个口分别为pitch，roll，yaw，temp。temp为陀螺仪温度可以用于调节温度控制PID

```
void vofa_demo(void)
```

```

{

    // call the function to store the data in the buffer

    //=====
    ===
    //ekf姿态解算的值
    vofa_send_data(0, pitch);
    vofa_send_data(1, roll);
    vofa_send_data(2, yaw);

    //=====
    ==

    //=====
    //mahony解算的值
    vofa_send_data(3, pitch_mahony);
    vofa_send_data(4, roll_mahony);
    vofa_send_data(5, yaw_mahony);

    //=====

    vofa_send_data(6, temp); //陀螺仪加热温度
    vofa_send_data(7, H723_Temperature); //h723内部温度
    // call the function to send the frame tail
    vofa_sendframetail();
}

```

移植注意

mahony代码只有mahonyahrs.c/.h文件，ekf在QuaternionEKF还有kalman_filter 的.c/.h中

EKF部分

需要实现这里的malloc，这里本文使用freertos的malloc来托管。

```
void* user_malloc(size_t size)
{
    void* tmp = 0;
    tmp = pvPortMalloc(size);
    return tmp;
}
```

如果移植到别的平台还需要优化kalman_filter.h里面的

```
#define mat_arm_matrix_instance_f32
#define Matrix_Init arm_mat_init_f32
#define Matrix_Add arm_mat_add_f32
#define Matrix_Subtract arm_mat_sub_f32
#define Matrix_Multiply arm_mat_mult_f32
#define Matrix_Transpose arm_mat_trans_f32
#define Matrix_Inverse arm_mat_inverse_f32
```

同时本文使用了dsp库中的函数加速，如果移植到别的平台还需要解决以下函数。

arm_atan2_f32 反正切函数 替换为相应平台的反正切

__sqrtf(x); 为dsp库内的 1/sqrt(t)函数。可以使用 mahonyahrs.c库中的 float Mahony_invSqrt(float x)来代替，

Mahony部分

这个函数由于只有arm平台可以使用，别的平台请使用另外一个函数：float Mahony_invSqrt(float x)，也在mahonyahrs文件中

```
static float invSqrt(float x) // if use other platform
please use float Mahony_invSqrt(float x)
{
    volatile float tmp = 1.0f;
    tmp /= __sqrtf(x);
    return tmp;
}
```

下面这个函数为所有平台都可以用的invSqrt函数，用于加速sqrt过程。

```

float Mahony_invSqrt(float x)
{
    float halfx = 0.5f * x;
    float y = x;
    long i = *(long*)&y;
    i = 0x5f3759df - (i>>1);
    y = *(float*)&i;
    y = y * (1.5f - (halfx * y * y));
    y = y * (1.5f - (halfx * y * y));
    return y;
}

```

如果移植到别的平台还需要解决，同ekf一样只要将arm_atan2_f32替换为自己平台的atan2。

```

void Mahony_computeAngles()
{
    arm_atan2_f32(q0*q1 + q2*q3, 0.5f - q1*q1 -
q2*q2,&roll_mahony);
    roll_mahony *= 57.29578f;
    pitch_mahony =57.29578f * asinf(-2.0f * (q1*q3 -
q0*q2));
    arm_atan2_f32(q1*q2 + q0*q3, 0.5f - q2*q2 -
q3*q3,&yaw_mahony);
    yaw_mahony *=57.29578f;
    anglesComputed = 1;
}

```