此飞控板对遥控器的功能有一定要求，通道数不少于 5 个，
其中一个为两段式或者三段式开关通道，需要有通道中立点和行程调整功能。

从未使用过的 MWC，需按照以下步骤设置和安装好，才能开始飞行时的调试：

1. 用 Arduino 编辑 MWC 程序，然后用 FTDI 工具把程序上传到飞控；
2. 安装到机架上，接好所有相关的连接线；
3. 飞行前用 MWC GUI 配置程序，对飞控进行基本设置；
4. 外场飞行时用电脑、蓝牙模块或者 LCD 模块来调整 PID 及其他参数。

接下来将按照以上步骤开始配置您的飞控。

1. 请先准备好以下驱动和程序：
   以下驱动和程序都可以用于苹果 MacOS、Linux 与 Windows 操作系统，我们以 Windows 7/32bit 为例进行说明。

   FTDI 工具驱动，FTDI 是一种 USB 转 TTL 电平的信号转换工具，我们用它来上传需要的程序到飞控，调试时也会用到。驱动下载后需要手动安装，安装好以后，电脑会出现一个 COM 口：



   例如本机上分配到的是 COM3，在不同电脑上分配到的端口可能会不一样，但不影响使用。请务必完成此安装步骤，否则无法上传程序到飞控。
   下载地址：
   http://www.ftdichip.com/Drivers/CDM/CDM20814_WHQL_Certified.zip

   MWC 程序源代码。MWC 程序升级较为频繁，每次更新都会出现实用的新功能，或者某方面性能得到提高，方便我们随时享用最好的效果。下载过来的 MWC 程序包，包括源代码和 GUI 配置程序两部分，请解压文件。
   下载地址：
   http://code.google.com/p/multiwii/downloads/list

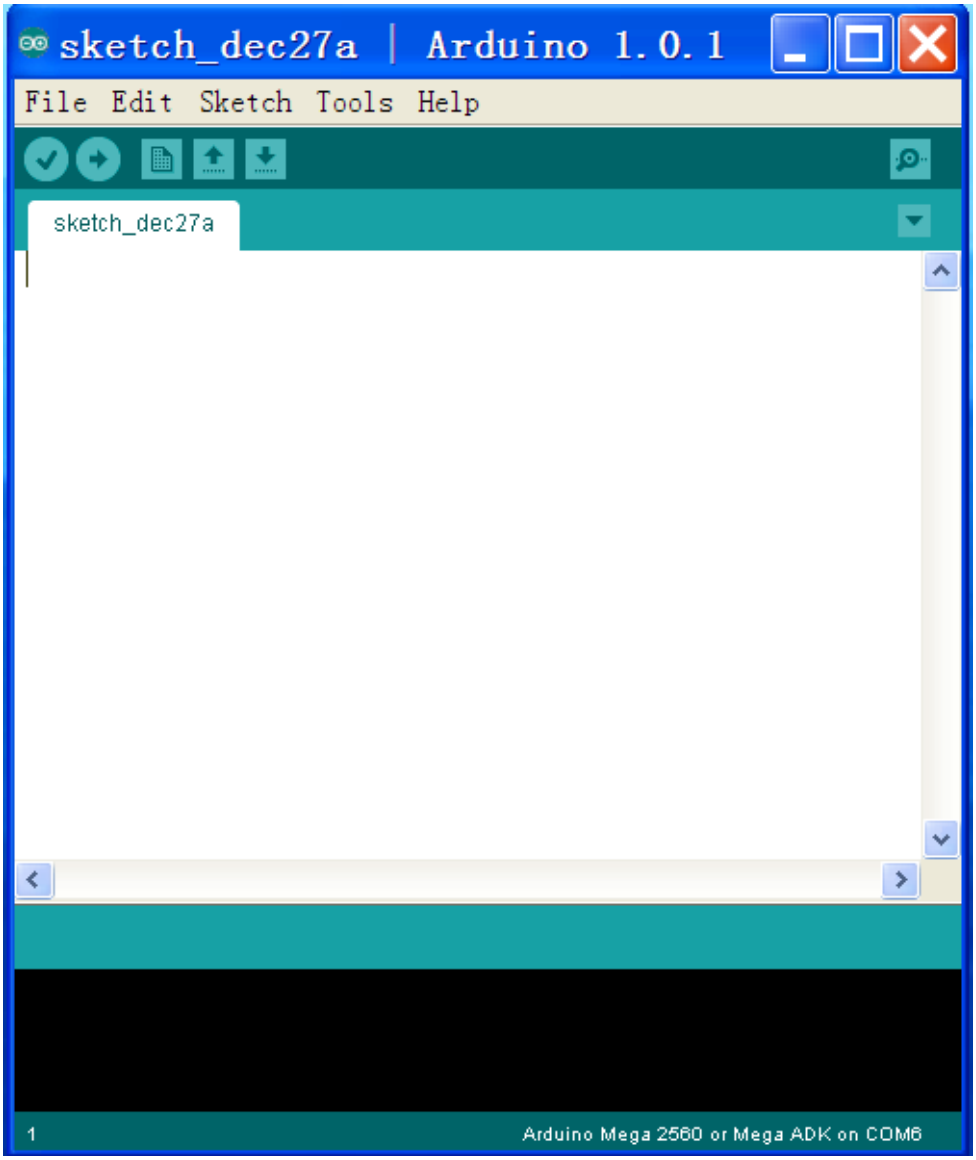
   Arduino 集成开发环境。我们使用 Arduino 这个工具来编辑 MWC 源代码，进行简单配置后，点击上传，程序就会通过 FTDI 传输到飞控中。如有需要调整程序的基础功能(例如切换飞行模式)，也将通过它来进行。下载后请解压文件，程序无需安装。
   下载地址：
   http://arduino.googlecode.com/files/arduino-0023.zip

   JAVA 虚拟机。也许您的电脑已经安装过，不过为了更好地运行 Arduino 与 GUI 程序，请升级到最新版。
   下载地址：

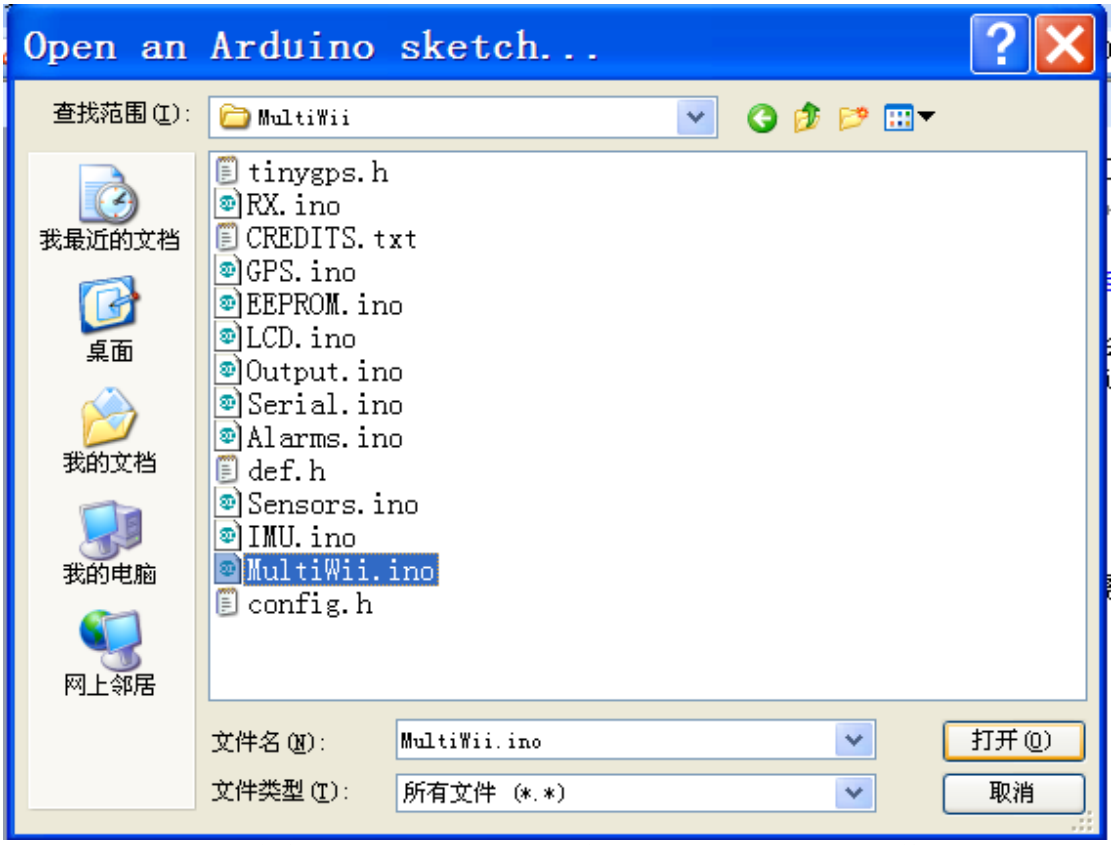
http://java.com/zh_CN/download/manual.jsp?locale=zh_CN

另外推荐收藏两个网址，分别是 MWC 官网与 RCGroups 论坛的 MWC 主帖，会有最新升级信息以及玩家讨论。
http://www.multiwii.com/
http://www.rcgroups.com/forums/showthread.php?t=1340771

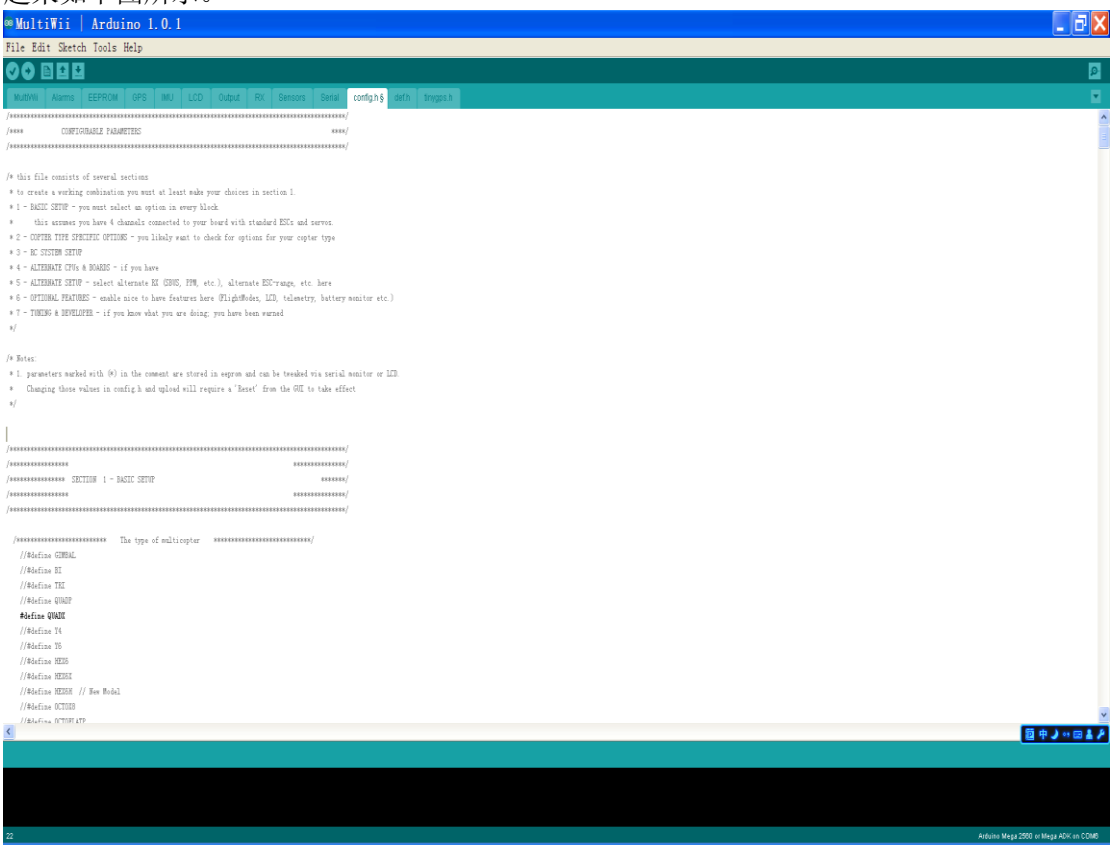
如果您购买飞控时已经指定好飞行模式，我们会对程序做好初步的修改，尽量让您可以直接使用，飞行时调整参数即可（除了 Y3），如果发现工作不正常，再来修改程序即可。
本文将以 4 轴×模式为例，讲解如何配置飞控程序，其他模式于此相同，只有 Y3 需要单独配置舵机。
准备好以上程序后，我们先打开 MWC 源代码，首先启动 Arduino 程序，看到如下窗口：



点击菜单中的“File”—“Open”
选择好您准备好的 MWC 主程序工程文件：

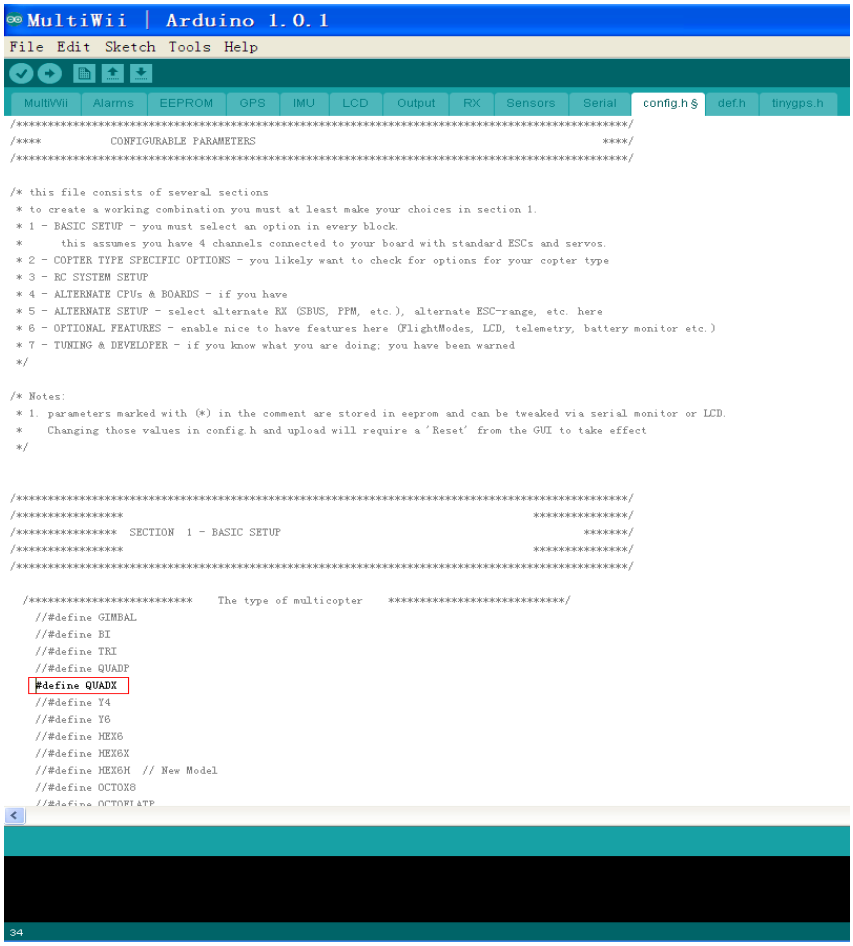


这里 MWC 的程序版本为 MultiWii_dev_r1240，请选择打开 MultiWii.ino 文件，打开后即可看到该工程文件包含多个源代码文本，会在新窗口中显示，之前空白的窗口可以关闭。我们一般只需修改 config.h 中的一些程序语句即可，请用鼠标切换至 config.h 标签，看起来如下图所示。



修改程序语句的方法：

每行语句左边带有“/*”或者“//”的时候，显示为灰色，表示该句为注释，是无效的程序语句。如果需要它生效，只需去掉“//”，它会以黑色显示出来，表示该语句有效。首先要清楚飞行模式也是从程序中去指定，但考虑到 MWC 硬件环境以及电调搭配的不同，有些其他关键语句是需要修改的，如果配置有误，会造成程序无法上传或者飞控工作不正常。我们不会对所有语句进行说明，仅仅修改必要的部分。请注意，不要轻易改动太多没有把握的语句，否则可能会出现问题而找不出原因在哪里。



我们从上往下看，第一个关键的地方是飞行模式的选择：

- The type of multicopter */
- //#define GIMBAL
  //#define BI
  //#define TRI
  //#define QUADP
  #define QUADX
  //#define Y4
  //#define Y6
  //#define HEX6
  //#define HEX6X
  //#define HEX6H // New Model
  //#define OCTOX8

这里我们选择的是 X 模式，您可以根据需要修改成其他模式，需要注意的是，MWC-S V1.5 并不支持上面所有飞行模式。

电调的怠速设置：

/* Set the minimum throttle command sent to the ESC (Electronic Speed Controller)
This is the minimum value that allow motors to run at a idle speed */
//#define MINTHROTTLE 1300 // for Turnigy Plush ESCs 10A
//#define MINTHROTTLE 1120 // for Super Simple ESCs 10A
//#define MINTHROTTLE 1064 // special ESC (simonk)
//#define MINTHROTTLE 1050 // for brushed ESCs like ladybird
#define MINTHROTTLE 1150
标准电调通过 1000~2000us 范围内的信号进行控制，如果您需要飞控解锁后电机就开始慢速旋转，请设定一个合适的怠速信号值(默认是 1150，大多数电调会开始输出电流)请注意此值不合适太小，否则会影响飞行器回中的效果，如果您喜欢解锁后电机停转，请启用“MOTOR_STOP“代码（请参见下文）。上传程序后如果发现不符合需求，可以再次修改后上传。

关于 I2C 信号频率的选择：
/*I2C speed*/
//#define I2C_SPEED 100000L //100kHz normal mode, this value must be used for a genuine WMP
#define I2C_SPEED 400000L //400kHz fast mode, it works only with some WMP clones
MWC-S V1.5 支持 400Khz 高速模式，请启用 400000L 这一行，注释掉第一行。

下面是非常关键的传感器类型选项：
/*Combined IMU Boards*/
/* if you use a specific sensor board:
please submit any correction to this list.
Note from Alex: I only own some boards, for other boards, I'm not sure, the info was gathered via rc forums, be cautious */
//#define FFIMUv1 // first 9DOF+baro board from Jussi, with HMC5843 <- confirmed by Alex
//#define FFIMUv2 // second version of 9DOF+baro board from Jussi, with HMC5883 <- confirmed by Alex
//#define FREEIMUv1 // v0.1 & v0.2 & v0.3 version of 9DOF board from Fabio
//#define FREEIMUv03 // FreeIMU v0.3 and v0.3.1
//#define FREEIMUv035 // FreeIMU v0.3.5 no baro
//#define FREEIMUv035_MS // FreeIMU v0.3.5_MS <- confirmed by Alex
//#define FREEIMUv035_BMP // FreeIMU v0.3.5_BMP
//#define FREEIMUv04 // FreeIMU v0.4 with MPU6050, HMC5883L, MS561101BA <- confirmed by Alex
//#define FREEIMUv043 // same as FREEIMUv04 with final MPU6050 (with the right ACC scale)
//#define NANOWII // the smallest multiwii FC based on MPU6050 + pro micro based proc <- confirmed by Alex
//#define PIPO // 9DOF board from erazz
//#define QUADRINO // full FC board 9DOF+baro board from witespy with BMP085 baro <- confirmed by Alex
//#define QUADRINO_ZOOM // full FC board 9DOF+baro board from witespy second edition
//#define QUADRINO_ZOOM_MS// full FC board 9DOF+baro board from witespy second edition <- confirmed by Alex
//#define ALLINONE // full FC board or standalone 9DOF+baro board from CSG_EU
//#define AEROQUADSHIELDv2

```
//#define ATAVRSBIN1          // Atmel 9DOF (Contribution by EOSBandi). requires 3.3V power.
//#define SIRIUS                                    // Sirius Navigator IMU <- confirmed by Alex
//#define SIRIUSGPS           // Sirius Navigator IMU using external MAG on GPS board
//#define SIRIUS600           // Sirius Navigator IMU  using the WMP for the gyro
//#define MINIWII                                   // Jussi's MiniWii Flight Controller <- confirmed by Alex
//#define MICROWII            // MicroWii 10DOF with ATmega32u4, MPU6050, HMC5883L, MS561101BA from http://flyduino.net/
//#define CITRUSv2_1          // CITRUS from qcrc.ca
//#define CHERRY6DOFv1_0
//#define DROTEK_10DOF        // Drotek 10DOF with ITG3200, BMA180, HMC5883, BMP085, w or w/o LLC
//#define DROTEK_10DOF_MS // Drotek 10DOF with ITG3200, BMA180, HMC5883, MS5611, LLC
//#define DROTEK_6DOFv2       // Drotek 6DOF v2
//#define DROTEK_6DOF_MPU // Drotek 6DOF with MPU6050
//#define DROTEK_10DOF_MPU//
//#define MONGOOSE1_0         // mongoose 1.0        http://store.ckdevices.com/
//#define CRIUS_LITE          // Crius MultiWii Lite
//#define CRIUS_SE            // Crius MultiWii SE
//#define OPENLRSv2MULTI    // OpenLRS v2 Multi Rc Receiver board including ITG3205 and ADXL345
//#define BOARD_PROTO_1       // with MPU6050 + HMC5883L + MS baro
//#define BOARD_PROTO_2       // with MPU6050 + slave    MAG3110 + MS baro
//#define  GY_80              // Chinese 10 DOF with  L3G4200D ADXL345 HMC5883L BMP085, LLC
//#define GY_85               // Chinese 9 DOF with    ITG3205 ADXL345 HMC5883L LLC
//#define GY_86               // Chinese 10 DOF with    MPU6050 HMC5883L MS5611, LLC
//#define GY_521              // Chinese 6    DOF with    MPU6050, LLC
//#define INNOVWORKS_10DOF // with ITG3200, BMA180, HMC5883, BMP085 available here http://www.diymulticopter.com
//#define INNOVWORKS_6DOF     // with ITG3200, BMA180 available here http://www.diymulticopter.com
//#define MultiWiiMega        // MEGA + MPU6050+HMC5883L+MS5611 available here http://www.diymulticopter.com
//#define PROTO_DIY           // 10DOF mega board
//#define IOI_MINI_MULTIWII// www.bambucopter.com
//#define Bobs_6DOF_V1        // BobsQuads 6DOF V1 with ITG3200 & BMA180
//#define Bobs_9DOF_V1        // BobsQuads 9DOF V1 with ITG3200, BMA180 & HMC5883L
//#define Bobs_10DOF_BMP_V1 // BobsQuads 10DOF V1 with ITG3200, BMA180, HMC5883L & BMP180 - BMP180 is software compatible with BMP085
#define FLYDUINO_MPU
//#define CRIUS_AIO_PRO_V1
//#define DESQUARED6DOFV2GO    // DEsquared V2 with ITG3200 only
//#define DESQUARED6DOFV4       // DEsquared V4 with MPU6050
//#define LADYBIRD
//#define MEGAWAP_V2_STD              // available here: http://www.multircshop.com <- confirmed by Alex
//#define MEGAWAP_V2_ADV
//#define HK_MultiWii_SE_V2    // Hobbyking board with MPU6050 + HMC5883L + BMP085
//#define HK_MultiWii_328P     // Also labeled "Hobbybro" on the back.  ITG3205 + BMA180 + BMP085 + NMC5583L + DSM2 Connector (Spektrum Satellite)
```

请直接启用#define FLYDUINO_MPU，否则飞控将不能正常工作

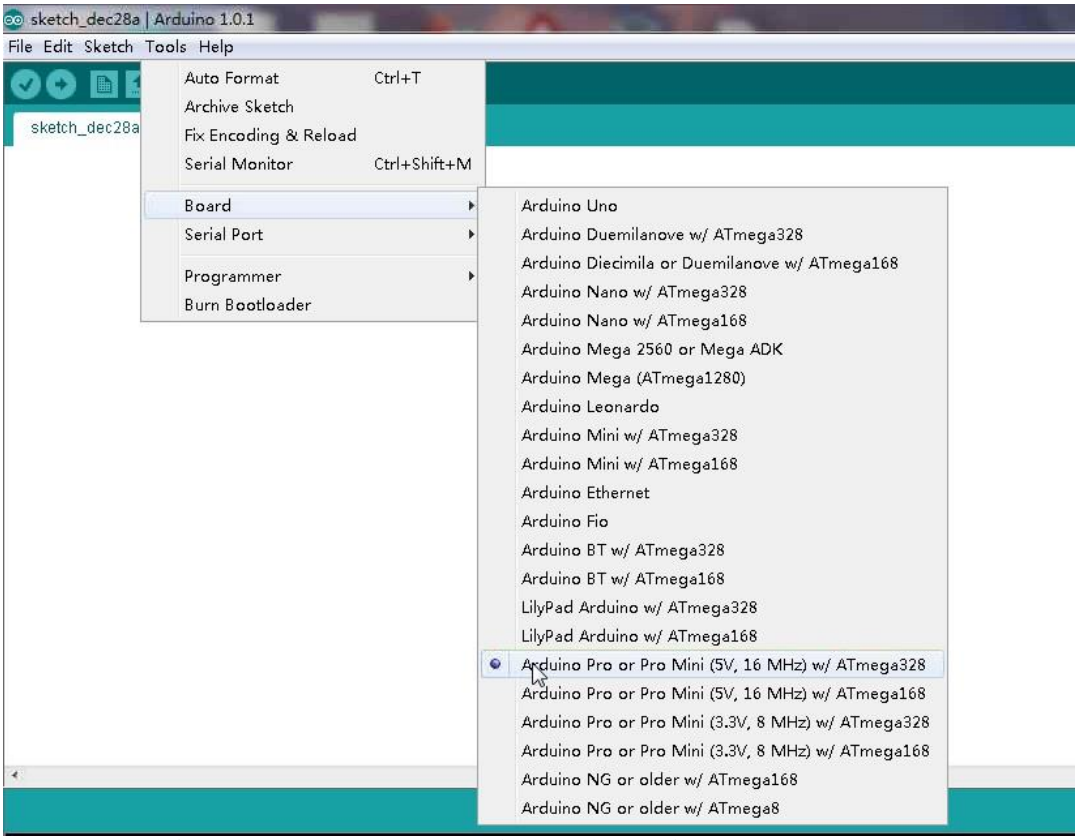
启用 A0 与 A1 端口输出：
/*Hexa Motor 5 & 6 Pins*/
/* PIN A0 and A1 instead of PIN D5 & D6 for 6 motors config and promini config
This mod allow the use of a standard receiver on a pro mini
(no need to use a PPM sum receiver) */
#define A0_A1_PIN_HEX
这两个端口原本用于连接云台稳定用的舵机，如果您没有 PPM SUM 类型接收机又想让 MWC-S V1.5 飞控支持 6 轴，请启用它，这样会让云台增稳功能失效。另外一种情况是，您想使用 6 轴模式，同时又想启用云台增稳功能，那必须使用 PPM SUM 接收机/JR 卫星/S.Bus.接收机，或者换用 PRO 版全功能飞控。

启用 D8 或者 D12 端口：
/*Aux 2 Pin*/
/* possibility to use PIN8 or PIN12 as the AUX2 RC input (only one, not both)
it deactivates in this case the POWER PIN (pin 12) or the BUZZER PIN (pin 8) */
//#define RCAUXPIN8
#define RCAUXPIN12
MWC-S V1.5 飞控带有 D7(AUX1)端口，通过三段开关的信号可以控制传感器状态的开和关，如果您觉得还需要 AUX2 进行更详细的状态控制，包括云台增稳开关或者作为解锁开关来用，可以启用 D12 作为 AUX2 的信号输入端口。

信号死区宽度：
/* introduce a deadband around the stick center
Must be greater than zero, comment if you dont want a deadband on roll, pitch and yaw */
#define DEADBAND 6
通常接收机出来的信号脉宽会有 1~3us 的误差（这是我们用信号测试仪检测的结果），设置一个合理的死区宽度，可以让飞控忽略这种轻微的误差，避免误差的信号直接去控制飞控动作。设置得过大的话，会让摇杆附近范围内的微小动作被飞控忽略到，反而降低飞控灵敏度，默认值为 6。

下列语句是关于解锁后，油门保持最低时油门停转
/* motors will not spin when the throttle command is in low position
this is an alternative method to stop immediately the motors */
#define MOTOR_STOP
如果不启用此项，解锁后电机会开始转起来，请根据需要来调整。

修改到这里就可以上传程序到飞控进行试飞了，上传之前还有两件事情需要准备好，一个是 FTDI 端口和飞控类型的选择，另外一个是您的电脑、FTDI 和飞控之间连接要准备好。MWC-S V1.5 是 328P 单片机，所以需要在 Arduino 的菜单“Tools”-“Board”中选择对应的选项 5V,16Mhz ATmega328：



FTDI 工具在本机分配到的端口为 COM3（每台电脑可能会不同），所以在“Tools”-“Serial Port”中需要选择“COM3”，如果没找到这个 COM 口，请检查与电脑的连接是否正常，还有驱动是否已经安装好，可以到“设备管理器”里面检查。





这两个地方设置完以后，需要确认飞控与 FTDI 工具是否连接好（见上图），飞控在使用 FTDI 的时候无需单独供电，只需与 FTDI 直接对接即可，接上后飞控就会开始启动，LED 会亮并且闪动。这样上传程序前的硬件连接工作就已经完成了。

修改好的程序可以点按保存按钮进行保存，方便日后直接调用。



上传好修改的程序，请点此按钮



然后会看到 Arduino 底部出现相应的状态信息

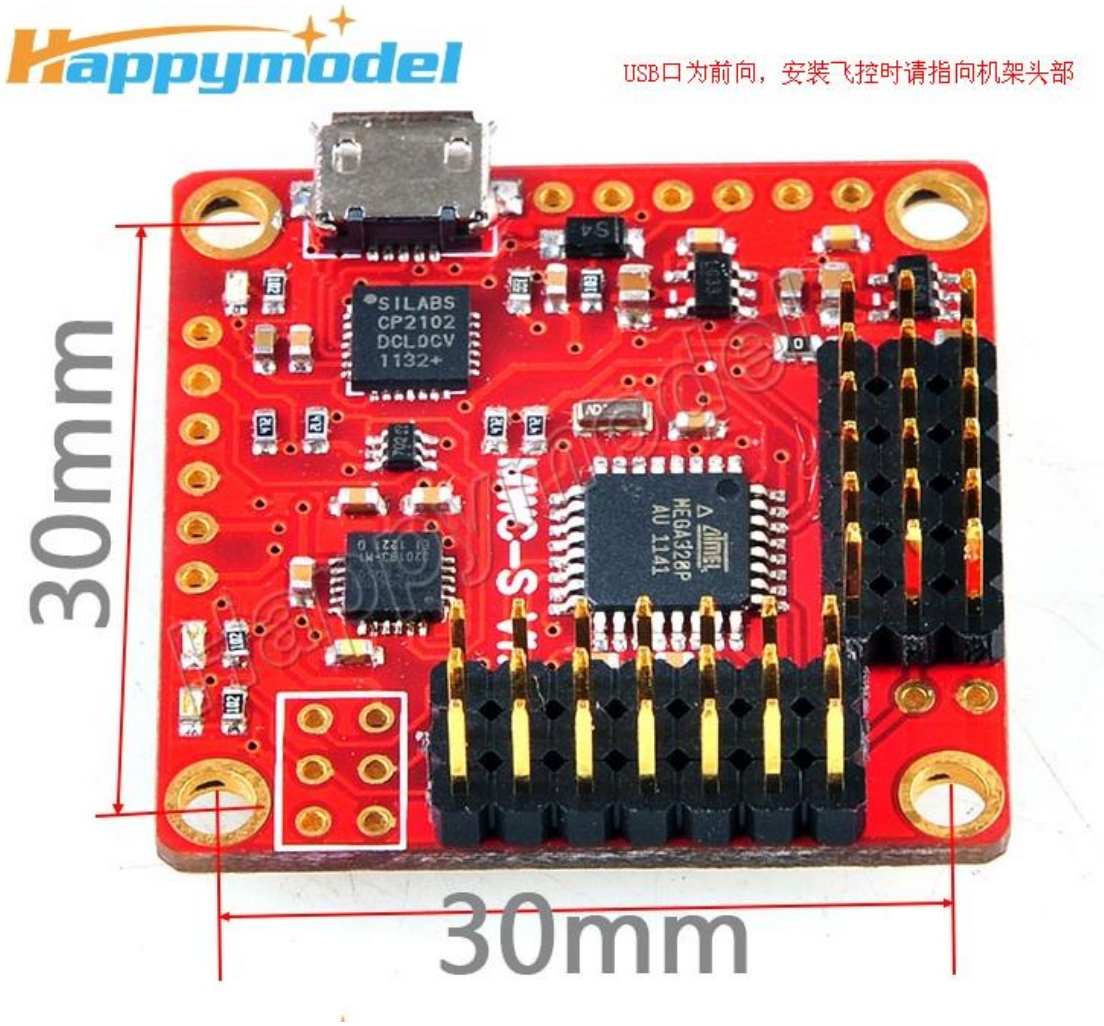


顺利上传后，会显示 Done uploading.



现在飞控已经可以正常工作，可以断开 FTDI 与飞控之间的连接。

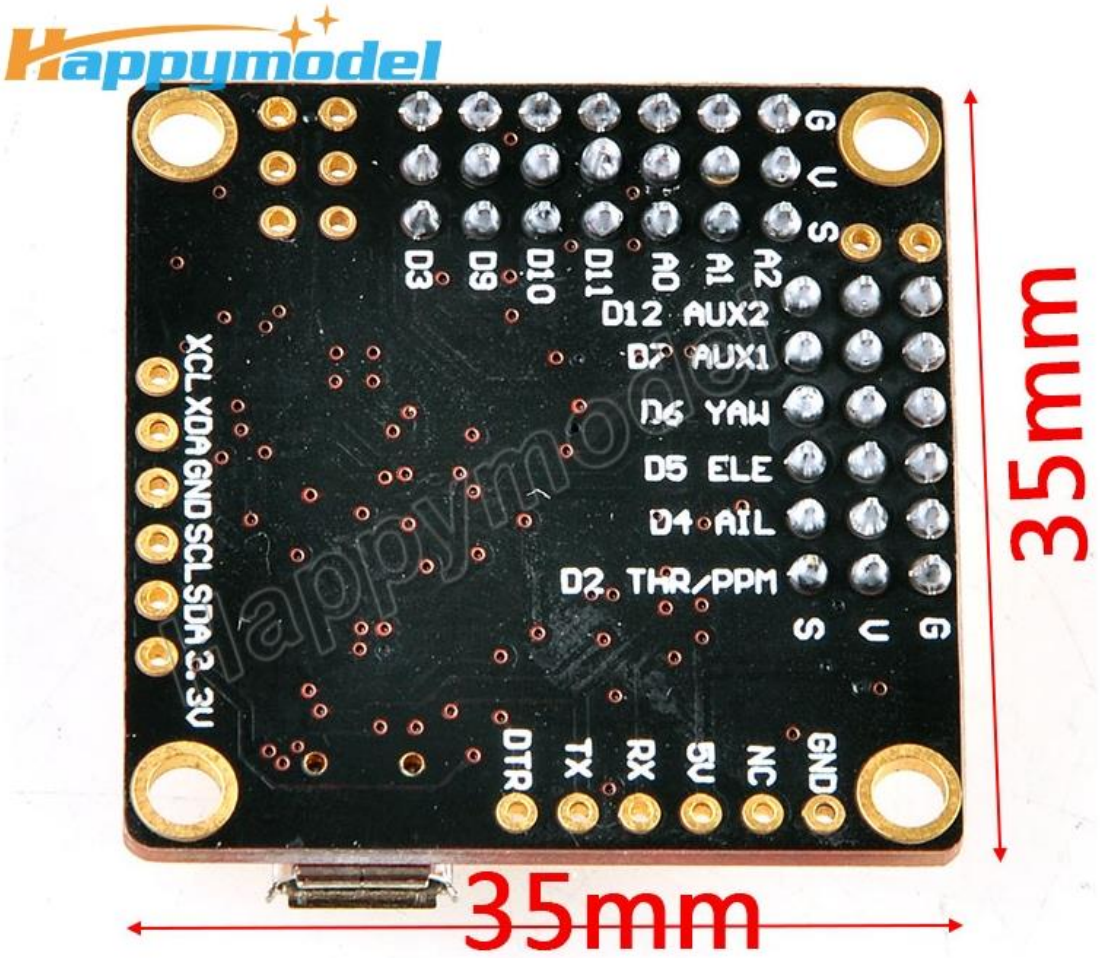
3. 请安装好您的机架及其他设备，飞控安装时请注意以下几个问题：



如有条件，请尽量使用减震方式安装飞控，譬如减震橡胶或者直接用泡棉双面胶固定在机架上，如果您确定不使用螺丝固定，请确保粘贴飞控时的角度准确，否则可能会严重影响飞行效果。

飞控与接收机之间的连接：
使用附带的杜邦连接线分别连接您的接收机与飞控，需要注意的是所有通道的信号级位置都是最靠飞控最内侧，标注有“S”的那一排插针。

飞控通道	接收机通道
THR	油门
ROLL（AIL）	副翼
PITCH （ELE）	升降舵
YAW（RUDD）	方向舵
D7	任意三段式开关通道，作为 AUX1 信号输入
D12	依程序选项而定，作为 AUX2 信号输入，也是任意三段式开关通道



飞控与电调之间的连接：

接上电调前，应该单独把每个电调接到接收机的油门通道上做一次油门校准，然后才接上飞控。我们不建议直接把每个电调的插头全部接到飞控上(虽然这样看起来似乎没问题)，并联了多组电调电源，可能会造成电调故障，如果供电出现问题，会导致危险！解决办法之一是让某个电调的 3 线插头直接连接到飞控上(这组负责供电给飞控与接收机)，其余每个电调都只连接信号级(S)，通常这条线为白色，这样可以避免电调供电并联。

补图：接收机、电调与飞控的连接

电调连接顺序以及电机旋转方向非常重要。为了安全，连接好以后先不要装上螺旋桨。电机上的数字表示连接到飞控上对应的输出端口号，例如“11”表示连接到飞控上的“D11”端口。请根据需要的模按照接线指示图来接好所有电机、电调和舵机

3 轴 Tricopter 在 config.h 中以“TRI”表示。



其中 3 是舵机，用这个模式还需要在 config.h 中调整舵机相关的选项。

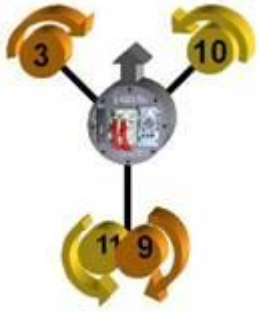
4 轴×模式/QuadcopterX 在 config.h 中以“QUADX”表示。



4 轴＋模式/Quadcopter＋ 在 config.h 中以“QUADP”表示。



4 轴 Y4 模式 在 config.h 中以“Y4”表示。



6 轴 Y6 模式 在 config.h 中以“Y6”表示。



6 轴 HEX6＋模式　在 config.h 中以“HEX6”表示。



6 轴 HEX6 X 模式　在 config.h 中以“HEX6X”表示。

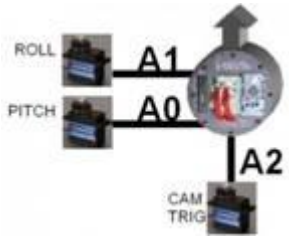


以上 3 种 6 轴模式，电机 5 和 6，如果开启 A0 与 A1，则由 A0 取代 5，A1 取代 6，否则您要使用 PPM SUM 接收机/JR 卫星/S.Bus 信号输入才能使用 6 轴，当然不用 PPM 输入又要使用 6 轴，就不能用云台增稳功能，除非使用 PRO 专业版飞控。

2 轴阿凡达模式 BICOPTER　在 config.h 中以“BI”表示。

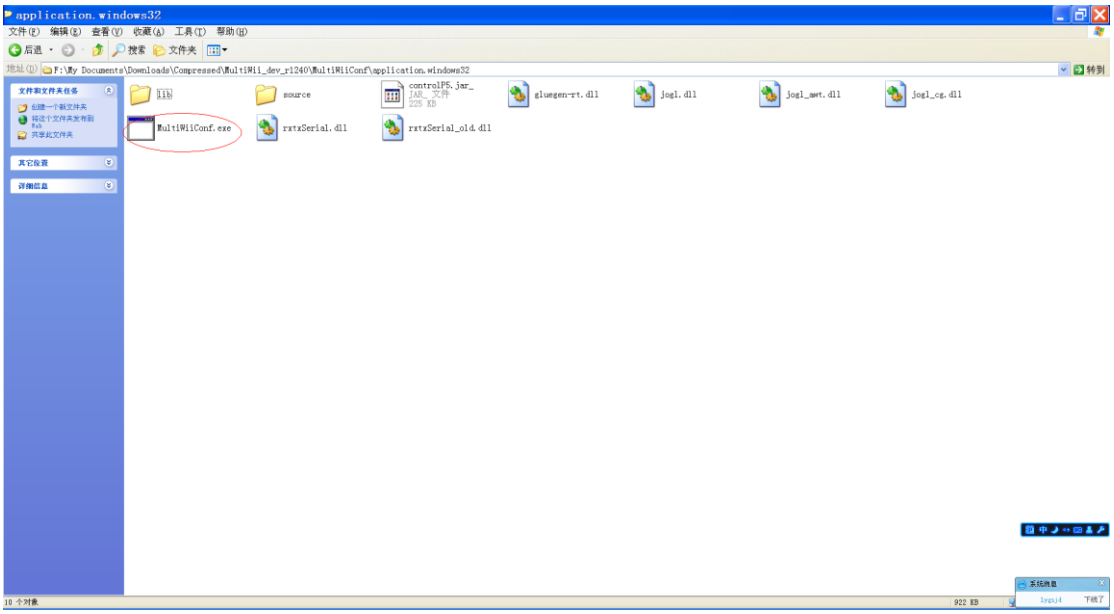


不作为飞控的云台稳定模式　在 config.h 中以“GIMBAL”表示。



4. 飞行前，需要用 GUI 程序观察和调整必要的选项，主要包括以下几个方面：

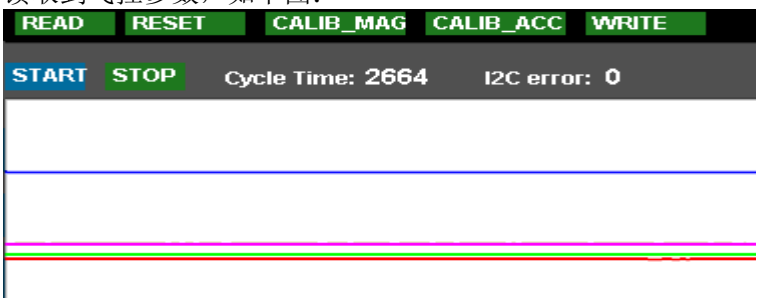
设置通道正反向，校准通道中立点和通道行程；
设置传感器开关(AUX1/AUX2)；
检查传感器输出信号是否正确(如果不正常则代表飞控有故障)；
测试启动电机后，飞控动作是否正常。

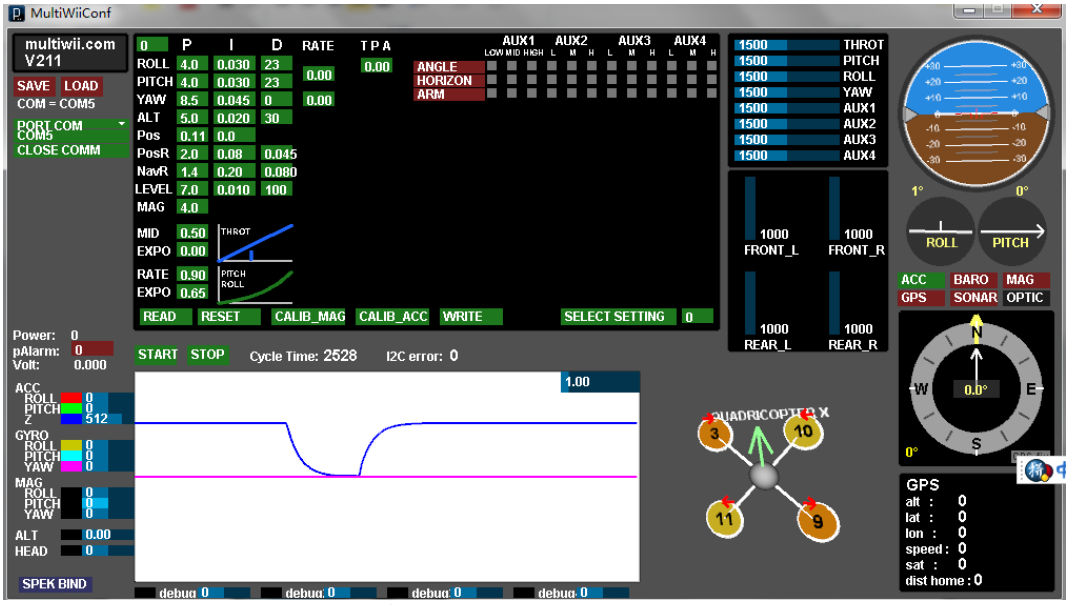


调试过程中可以不需要连接电池，只需接上 FTDI。飞控启动时，请保持机架在尽可能水平的位置，蓝色 LED 灯会在一阵快速闪动之后熄灭，表示已经初始化完成。请打开 GUI 程序，选好端口号（本例为 COM3）。



然后点按 START 进行数据读取，这时马上会看到所有传感器的输出状态，点按 READ 可读取到飞控参数，如下图：





请先看遥控器信号监控，垂直的两条分别是油门与升降舵通道，水平的是副翼、升降舵与 AUX1 通道：



设置通道正反向：

油门/升降舵摇杆往上下方向推，图中对应的滑块也应该往对应方向滑动，如果不正确，则需要把对应的通道反向；

副翼/发射电摇杆往左右方向推，图中对应的滑块也应该往对应方向滑动，如果不正确，则需要把对应的通道反向；

校准通道中立点：

油门通道无需校准，观察升降舵/副翼/方向舵摇杆在中间位置时，对应的滑块是否显示为 1500 左右（允许±3 误差），如果正确，请修改对应通道的中立点，让这三个通道摇杆在中立点时，对应的数值也在 1500 左右。请注意，如果中立点信号位置不准确，会导致飞行误动作。

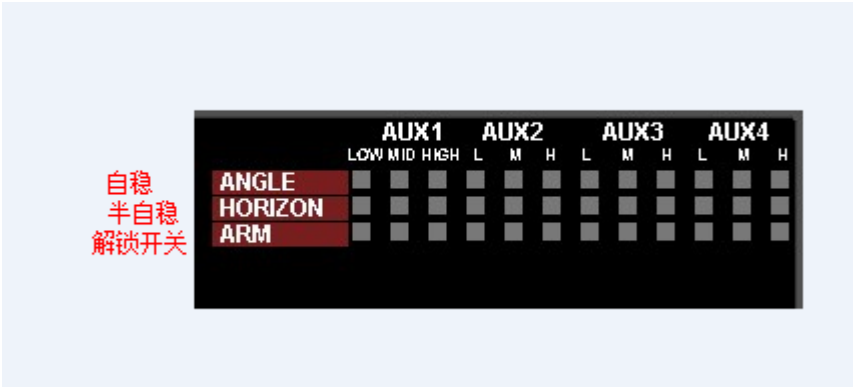
校准通道行程：

油门通道无需校准，观察升降舵/副翼/方向舵通道摇杆在推到最大/最小位置时，对应的滑块是否显示为最大 1900/最小 1100 左右，如果有偏差，请修改对应通道的行程，让其最大值刚好超过 1900，最小值刚好低于 1100。请注意，如果不处理好，会导致解锁/锁定油门、校准传感器和进入液晶编程模式出现故障。

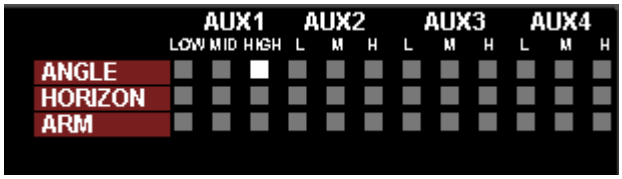
接下来需要设置传感器开关

遥控器的两段式或者三段式开关信号由飞控的 MODE 端口输入后，会作为 AUX1 显示出来，拨动开关，会看到 AUX1 的滑块跟随滑动，这里需要靠它来控制传感器的开关状态。下图中我们主要设置前三个传感器状态，灰色小方格表示当前设置无效，点选成白色后表示有效。

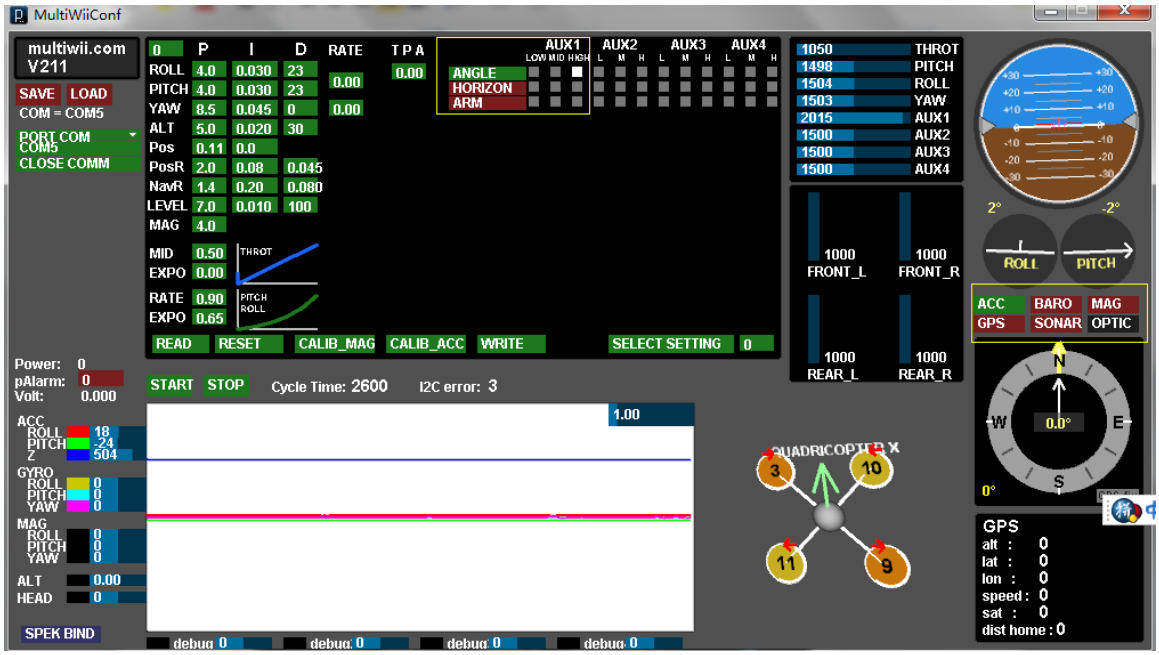
ANGLE 是自稳功能控制（需要加速度计，MWC-S V1.5 支持）
HORIZON
ARM



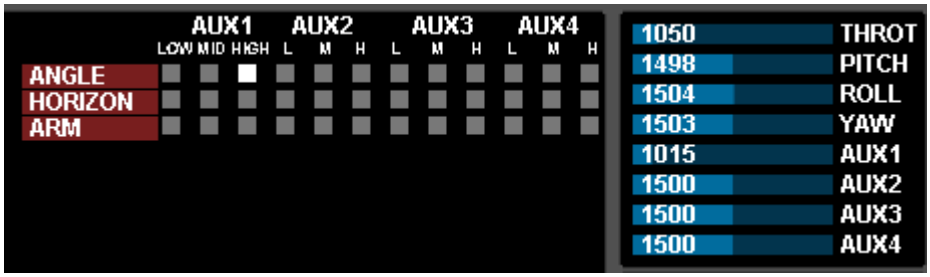
上图中 AUX1 下方的三列小方格，分别表示三段开关的三个状态，当开关处于低中高三个位置时，会对应选中左中右三个方块。我们举例说明这个设置方法，比如下图中点选其中 1 个小方格，并按 WRITE 保存设置。



那么当开关位置处于最高点时，AUX1 的滑块在右边（图中是 2015），ACC（加速度计）显示为绿色启用启用状态，BARO（气压计）和 MAG（磁阻）都为 OFF（关闭），对应 AUX1 下面左边 HIGH 这一列，开关在这个位置只有 ANGEL 的小方格是白色，并且“ANGLE”显示为绿色。

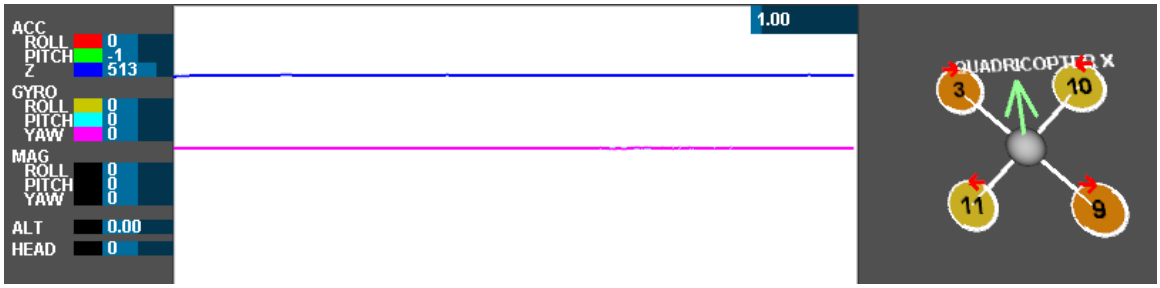


当开关位置处于最低点时 AUX1 的滑块在左边（图中是 1015），这时 ANGLE（加速度计）由绿色变成褐色（禁用）状态。



相信您已经掌握这种设置方法，上图传感器开关状态，绿色是启用，褐色是禁用，传感器的状态可以根据自己的需要来进行设置，并无标准。注意点选小方块时如果无法切换为白色或者灰色，请多试几次，完成后一定要按底部绿色的 WRITE 按键进行保存，否则设置会无效，设置完成后按一次 READ 以确认飞控是否保存正确，然后拨动开关观察状态是否是您想要的效果。

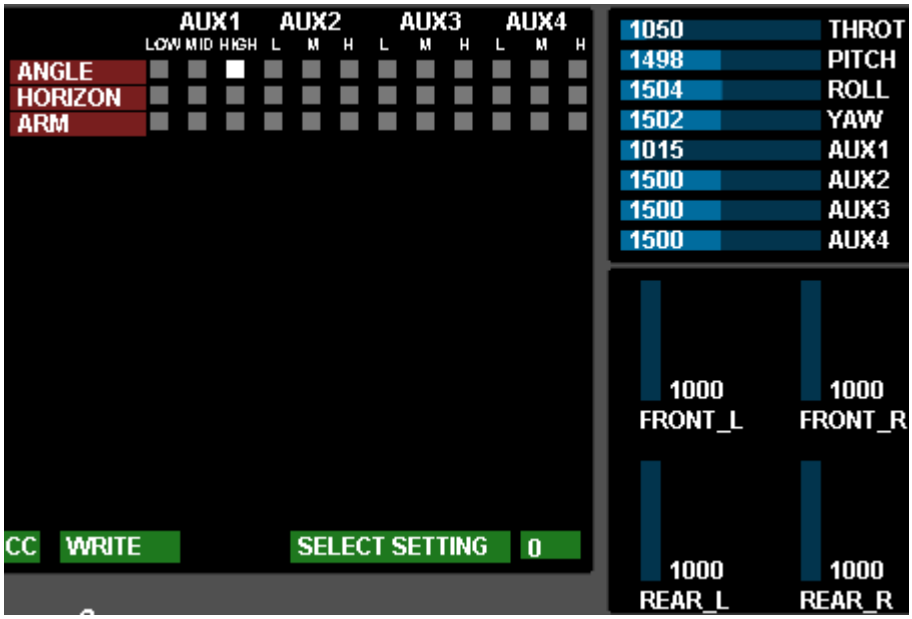
下面我们需要检查 ACC 传感器的数据输出是否正确，万一出现故障，请不要进行飞行，应联系我们进行维修。



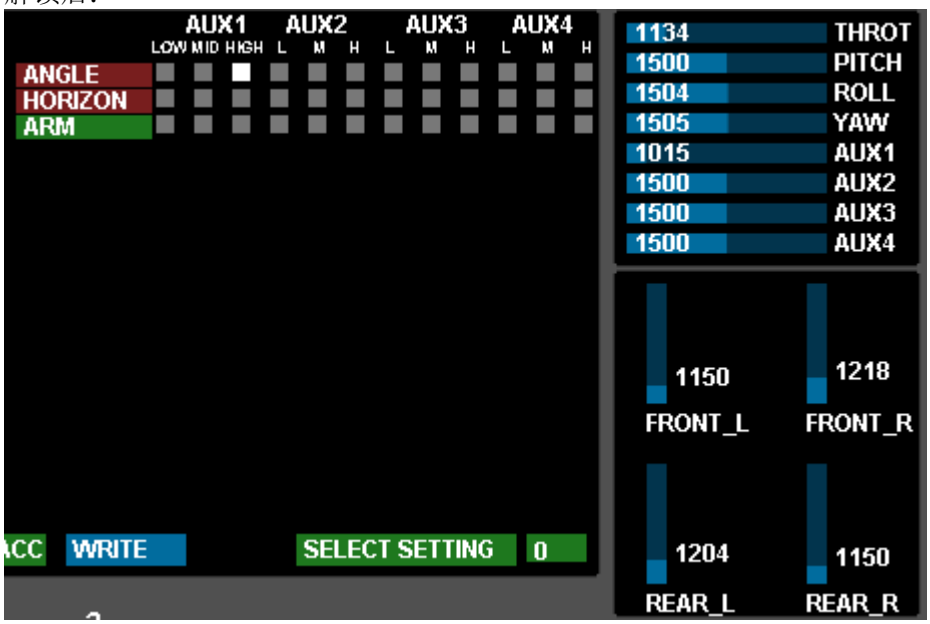
加速度计（ACC）在飞行器静止时，三轴读数会相对稳定，如果前后/左右/上下移动飞行器，对应的 PITCH/ROLL/Z 会有读数变化，这表面加速度计工作正常。这时如果需要校准加速度计，请将飞行器放在尽量平的地面上，点按 CALIB_ACC 按钮，等待大约 5 秒钟，加速度计校准完成后，ROLL/PITCH 会显示为 0，校准完成。

接下来应该请尝试解锁油门，油门摇杆保持最低，方向舵或者副翼摇杆推到最右，直到看到蓝色 LED 灯恒亮，表示解锁完成。本例我们设置的解锁后油门信号为 1060（见上文程序修改部分），请看下图对比解锁前后油门信号的变化。

解锁前：



解锁后：

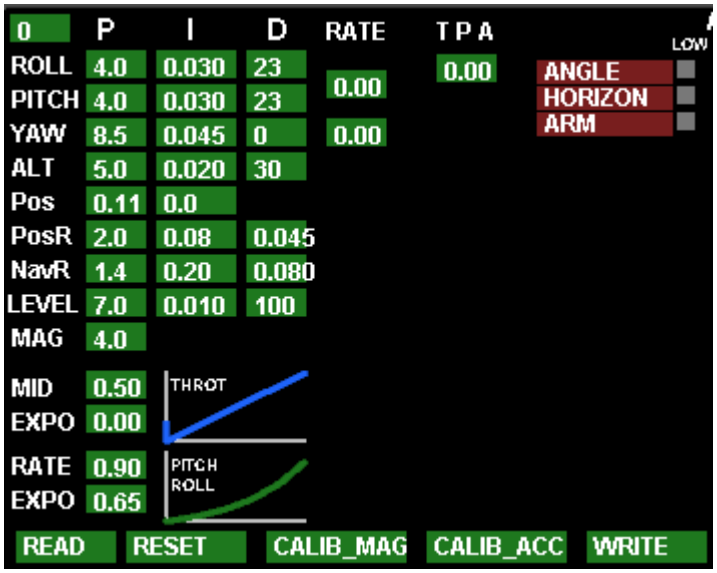


本例设置的飞行模式为 X 模式，上图中 4 个条状分别代表 4 个电调的控制信号，排列位置近似于×结构。

解锁前蓝色 LED 为熄灭状态，信号显示为 1000（默认值），解锁后推高油门电机就会开始快速转起来。请注意，进行这些测试的时候仍然不要装上螺旋桨。如果出现某些电机不转，请检查相关连接线是否有问题。

您还可以抓着飞行器，前后左右摆动，观察电机的加减速修正是否正确，比如往前倾斜，前面两个电机会明显加速，同时上图中 FRONT_L 与 FRONT_R 都会往上滑动，数值增大。同时还需要注意电机旋转方向是否正确，请参见前文中不同飞行模式的电调接线位置图。

飞控主要调整的参数是下图中绿色背景的数值，修改方式是用鼠标左键在需要调整的数值方框内点按住不动，左右滑动鼠标，修改完毕后松开鼠标左键即可。请记得每次修改完一系列参数后，点按 WRITE 进行保存。



PID 参数右边的 RATE（包含 ROLL/PITCH/YAW）表示飞控的灵活程度，数值越高飞行器越灵活，默认的 0.00 适合初学者，最大值 1.00 适合高机动性飞行，例如飞特技空翻或者其他高机动飞行动作，如果是 FPV 或者常规飞行可取值在 0.4~0.7 之间。

TPA(Throttle PID attenuation)可改善快速爬升时飞行器的稳定性，通常为默认值 0.00。

RC RATE 表示遥控信号的动作量（等同于遥控器的 D/R），这里针对 ROLL 与 PITCH，1.9 版本中默认为 0.9，如果您觉得稍微动一下摇杆，飞行器的动作量就很大，可以适当减小该值，反之应该加大。

RC EXPO 表示摇杆动作柔和度（等同于遥控器的 EXPO），如果该值增大，那么摇杆的上下左右动作在一小段范围内会显得更为柔和，意思是减小摇杆附近范围内输出的动作量，这样会让飞行器动作更为柔和，如果您想让操控飞行器的动作灵敏度更高，请适当减小该值。该值介于 0～1.0 之间，0 表示没有柔和度，1 表示柔和度最高。

LEVEL 的 P 与 I 值，LEVEL 这一项用于控制自稳效果（它必须在 LEVEL 功能启用时才有效果），如果飞行器飞行时自稳效果不好，可以通过调整 P 值改善。

ROLL/PITCH/YAW 的 PID 值我们会在后面另外说明。

恭喜您！飞行前的调试到此结束，接下来可以开始进行飞行测试和调整了。

5. 飞行调试。

MWC 是用 PID 算法进行飞行稳定性控制的程序，三个字母分别代表比例（P）-积分（I）-微分（D）。

PID 三个参数的直观作用：

P（比例）：这是一个增益因子，当多轴飞行器受风等的影响发生向一边倾斜时，P 值直接决定多轴飞行器的抵抗这种倾斜的力的大小。P 越大，多轴飞行器抵抗意外倾斜的能力越强，但 P 过于大时会引起多轴飞行器抖动甚至猛烈侧翻。P 越小，多轴飞行器抵抗意外倾斜的能力越弱，但 P 过小时会引起多轴飞行器自平衡能力不足甚至朝一边侧翻（如顺着风的方向）。

I（积分）：这个参数决定了飞行控制器对过往飞行状态的依赖程度。如果 I 值太小，会使飞行器过度依赖当前的误差，不能抑制“过敏”现象，从而造成飞行颠簸；如果 I 值太大，则会过度削弱系统对误差的反应能力，造成反应迟缓。

D（微分）：一旦多轴飞行器发生倾斜，则认为多轴飞行器会继续向同一方向倾斜，合适的 D 参数的能有效抑制未来可能发生的倾斜。如果 D 值太小，您会觉得多轴飞行器反应不够灵敏；如果 D 值太大，也会引起“过敏”。相较于 P 而言，D 反映得更多的是灵敏度，而 P 反映的是纠正误差的力度。

我们在此引用国外模友的经验帖，相信能帮助您有效掌握一些调整技巧。

原文地址：http://www.rcgroups.com/forums/showthread.php?t=1375728

－－－

I – 对角度变化进行采样与取平均值计算的时间周期

存在偏差时，返回到初始位置有个修正的过程，在修正过程中力度会越来越大，直到达到最大值。

较高的 I 值会增强稳定的效果。

增大 I 值：

减小漂移和提升稳定效果，但过大的 I 值会延长稳定过程的周期，同时也会降低 P 的效果。

减小 I 值：

增强反应速度，但会增加漂移和降低稳定效果，同时会提升 P 的效果。

特技飞行：需要的 I 值稍低。

普通飞行：需要的 I 值稍高。

D – 飞行器回到初始位置的速度

较高的 D 值（数值与效果相反，较高的 D 值参数上会更接近 0）意味着飞行器会快速返到回初始位置。

增大 D 值：（请记住，增大效果等于减小设定的参数值）

提升修正的速度，也会提升产生修正过量与振荡的几率，同时也会提升 P 的效果。

减小 D 值：（请记住，减小效果等于增大设定的参数值）

降低由偏差状态返回初始位置的振荡，恢复到稳定效果的时间变长，同时也会降低 P 的效果。

特技飞行：增大 D 值（请注意是减小参数值）。

普通飞行：减小的 D 值（请注意是增大参数值）。

实际应用中的高级调整

（下面这些只是建议）

特技飞行：

增大 P 值直到出现振荡，然后稍微减小一点。

改变 I 值直到悬停漂移达到无法接受的程度，然后稍微增大一点。

增大 D 值（请注意是减小参数值）直到出现大动作控制时无法接受的振荡。

P 值现在可以稍微减小一点。

稳定飞行（常规/FPV）：

增大 P 值直到出现振荡，然后稍微减小一点。

改变 I 值直到偏差程度达到无法接受的程度，然后稍微增大一点。

减小 D 值（请注意是增大参数值）直到出现大动作控制时回中速度变慢，然后稍微增大一点 D 值（请注意是减小参数值）。

P 值现在可以稍微减小一点。

在一般的飞行中，你会找到一组折中的 PID 值，这是最优化的效果。

其他影响 PID 的因素

你会倾向于某组 PID 参数值，但不会有两个最佳状态的飞行器拥有完全同样的参数，这时因为下列的原因造成：

机架的重量/尺寸/材料/硬度
电机的功率/扭力/推力
电机的安装位置/电机的间距
螺旋桨的直径/螺距/材料
机架的平衡效果
操控者的技能

参考资料，维基百科中关于 PID 控制的介绍：
http://en.wikipedia.org/wiki/PID_controller

LED 指示方式及各种功能的摇杆动作说明

蓝色 LED 显示方式：

通电后初始化：快速闪动几秒钟后熄灭；
初始化陀螺/加速度计：快速闪动几秒钟后熄灭；
开机后无法顺利初始化：持续闪动（请重新上电）；
锁定时，飞行器倾角过大：闪动；
锁定时：熄灭；
解锁后：常亮。

启动（解锁）和锁定油门

解锁：保持油门摇杆最低，方向舵摇杆（或者副翼摇杆）最右，直到蓝色 LED 常亮。
如果怠速油门设置为低转速而不是停转，解锁后电机会马上转起来，注意安全！
锁定：保持油门摇杆最低，方向舵摇杆（或者副翼摇杆）最左，直到蓝色 LED 熄灭。

陀螺仪校准

在锁定状态下，保持油门摇杆最低，方向舵摇杆最左，然后升降舵摇杆拉到最低，蓝色 LED 会持续闪动几秒后熄灭，校准结束，松开摇杆。陀螺仪初始化过程中，请勿移动飞行器。

加速度计(自稳功能)校准

飞行前必须操作一次校准。飞行器应放在尽量水平的地面（或其他平面），在锁定状态下，保持油门摇杆最高，方向舵摇杆最左，然后升降舵摇杆拉到最低，蓝色 LED 会持续闪动一段时间后熄灭，校准结束。注意要先松开升降舵摇杆，才松开方向舵/油门摇杆。在 GUI 中的“CALIB_ACC”按键同样具有校准功能。

操作结束前请勿移动飞行器，如果校准没有正确完成，蓝色 LED 会一直闪动而不熄灭，这样您也无法解锁油门（安全设计），需要重新上电操作初始化。
请注意，如果校准后先松开方向舵摇杆，会误进入加速度微调功能。

加速度计（自稳功能）微调

飞行器在无风状态下悬停时如果会朝某个方向漂移，就需要进行加速度计微调。在锁定状态下，保持油门最高，升降舵摇杆和副翼摇杆可以根据情况打到最大/最小位置再回中，每次操作时蓝色 LED 会闪动，根据需要重复这个操作。微调完成后升降舵/副翼摇杆回中，油门摇杆回到最低位。
例如悬停时飞行器往正前方漂移，微调时就需要把升降舵摇杆往下打到底再回到中间，根据漂移的严重程度重复操作数次，每次摇杆打到底都会看到 LED 闪动，重新解锁起飞后观察效果，不行的话再次进行调整。

★以上 4 项操作如果出现问题，请检查 1~4 通道的正反向以及通道行程是否正确（GUI 中观察的每个通道最小值应小于 1100，最大值应大于 1900，否则可能会造成设置失败。）

使用液晶模块时，可进入编程模式用遥控器设定参数。

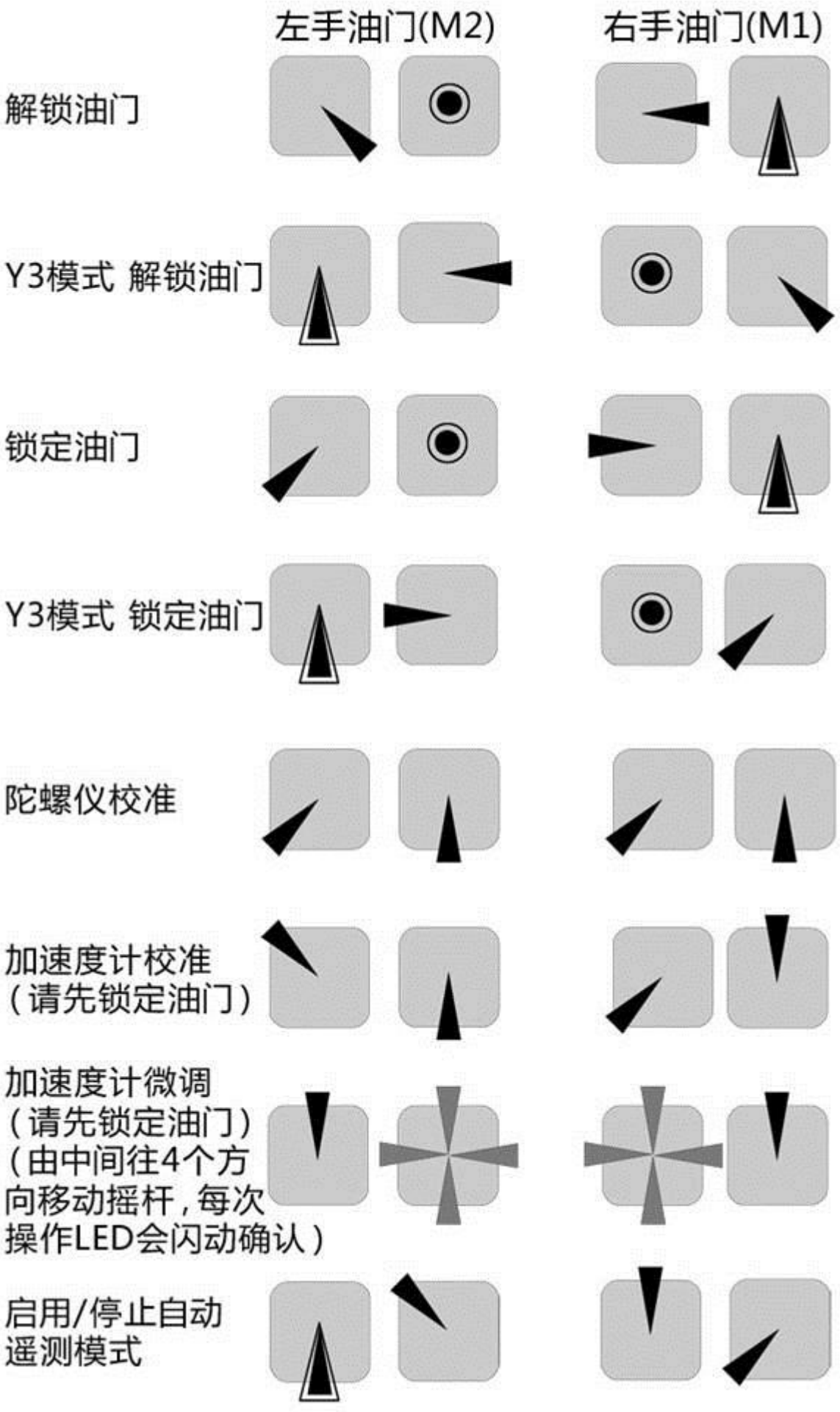
进入编程模式：在锁定状态下，先将升降舵摇杆推到最高并保持，然后油门摇杆最低，方向舵摇杆最右，蓝色 LED 会持续闪动，这时可以松开两个摇杆，继续等待几秒钟，LED 再次快速闪动后就表示进入编程模式。接着可以接上液晶模块（如果未进入编程模式而是先接上液晶模块，可能会显示不正常）。

参数项目选择：保持油门最低，用升降舵上下摇动选择参数项目。

参数值选择：保持油门最低，用方向舵摇杆左右摇动选择参数值。

保存参数并退出编程模式：在编程模式下，先将升降舵摇杆推到最高并保持，然后油门摇杆最低，方向舵摇杆最左，蓝色 LED 会持续闪动几秒后熄灭，这时已经保持并退出编程模式。如果再进行参数选择操作将会无效。

摇杆组合动作图


左手油门(M2)
右手油门(M1)
解锁油门
Y3模式 解锁油门
锁定油门
Y3模式 锁定油门
陀螺仪校准
加速度计校准
（请先锁定油门）
加速度计微调
（请先锁定油门）
（由中间往4个方
向移动摇杆，每次
操作LED会闪动确认）
启用/停止自动
遥测模式

编程模式摇杆组合动作图

