固定翼飞机竖直飞行控制系统

- **题目名称：**固定翼飞机竖直飞行控制系统
- **比赛编号：**1000074
- **日　　期：**2007.11.27
- **单位名称：**东冠电子厂
- **联 络 人：**马山（中文）
- **ＮＡＭＥ：**Ma shan（英文）
- **职　　业：**研发副理
- **通讯地址：**广东省东莞市大朗镇洋乌工业区 东冠电子厂
- **邮　　编：**523789
- **电子邮箱：**5483@vegamedical.cn
- **电　　话：**0796-87080999-892
- **传　　真：**0769-83186099
- **手　　机：**13728344251

内容目录

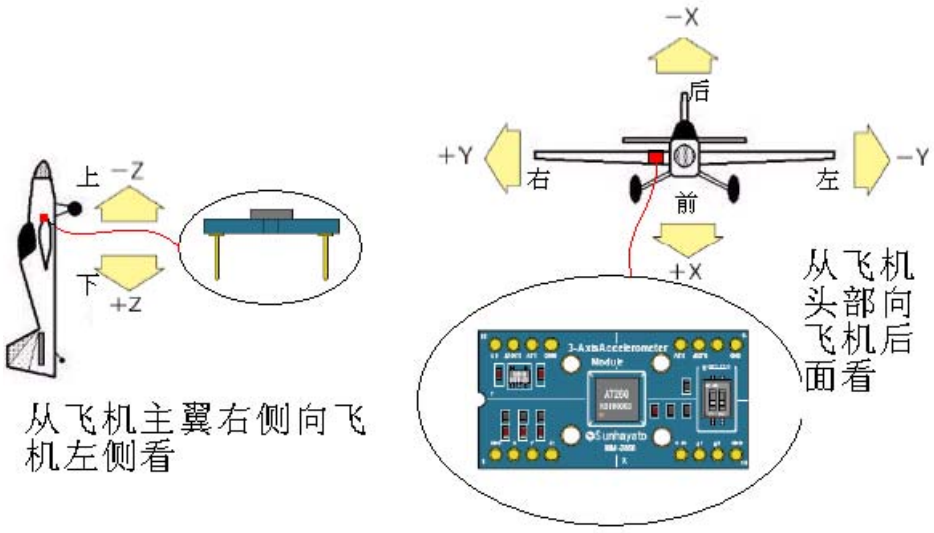
1　引言

一般的固定翼飞机水平飞行是最正常的飞行状态，如让其做大仰角飞行则因为各种原因动作不能一直保持，如让其垂直于地面飞行保持的时间则更短，例如有名的“普加乔夫眼镜蛇”（见图一和下述网址录像）也只能在垂直状态保持一到两秒。

http://www.tudou.com/programs/view/514g7jBkGdQ/

图一

本项目要实现的是一个 可以让飞机垂直于地面飞行并可以稳定在固定高度 的控制系统。通过在机身加入三轴向加速度传感器检测垂直于地面飞行的飞机的姿态（图二），当检测到飞机前后摇摆时控制水平尾翼保持前后方向的稳定性，当检测到飞机左右摇摆时控制垂直尾翼使飞机保持左右方向的稳定性，当检测到飞机上下窜动时控制油门的大小，也就是螺旋桨的转速保持飞机飞行高度的稳定性，当这些控制都很精准到位时，固定翼飞机也可以象直升机一样稳稳的悬停在空中了。



图二

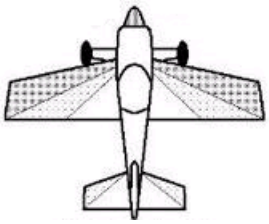
无论是战斗还是表演，让飞机保持机头向上并稳定在固定高度都是有用的特技，如果此项目可行，飞机将做出比“普加乔夫眼镜蛇”更有欣赏性和实用性的动作。

作为飞行表演，这应该是一个激动人心的时刻，眼前是一架巨大的飞机在身边悬停，耳边是螺旋桨的轰鸣声，悬停几秒后飞机咆哮着冲向云霄……观众想不鼓掌都困难。

在战斗中，飞机的特技飞行也是可以和高科技导弹相提并论的，飞机的战斗中一般都是后面的打前面的，前面的飞机要想摆脱被动状态只有想办法到后面的飞机的后面去，“普加乔夫眼镜蛇”就相当于飞机的刹车可以使飞机的速度突降，当后面的飞机冲到前面时，主被

动状态也就反过来了。但是“普加乔夫眼镜蛇”有一个弱点就是当这个动作在执行时他对后面飞机的投影面积剧增（见图三，图四，图三是从后面的飞机看到的前面的飞机正常飞行时的投影；图四是前面的飞机作“普加乔夫眼镜蛇”时的投影，可以看出，投影面积明显增大），这样就增加了被打中的机会，也就是说还在你做这个动作时，后面飞机的飞行员可能已经手疾眼快的扣动了机炮的开关，你也就垂直的掉下去了。利用本文提到的控制系统可以让飞机在做垂直飞行减速的同时让飞机顺时针或逆时针转 90 度（见图五，投影面积比图四小了很多），这样就可以把暴露给后面飞机的投影面积减小至水平飞行时的大小，减小被击中的机会，而“普加乔夫眼镜蛇”是无论如何也做不到这点的。

图三



图四

图五

2 设计概述

为了让飞机能竖直飞行，控制电路需要有以下功能。

2.1 判断此时飞机是否要竖直飞行：此项比较简单，对于真飞机，它可能只是一个可防止误触发的拨动开关，在本项目的实验品遥控飞机里，它就是某一个通道的脉宽信号。当脉宽大于某一设定值时，控制电路就知道它该上场了。

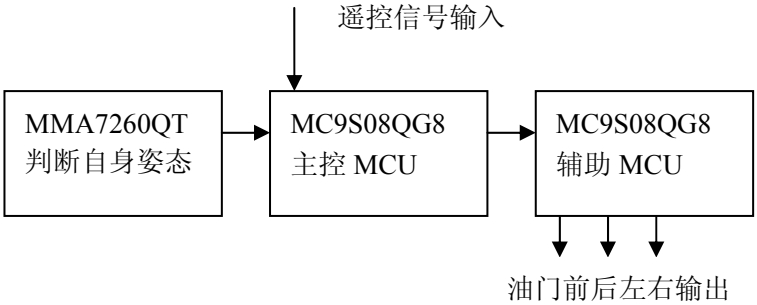
2.2 判断自身姿态：这是本控制电路的核心部分之一，要想控制自己保持竖直，得先判断自己现在是否竖直，高度有无变化，知道了自身姿态才好对症下药，对飞机各相应机构进行操纵以维持竖直姿态。本项目用的是 MMA7260QT 传感器检测飞机的姿态的，MMA7260QT 传感器的规格如下：

http://www.freescale.com/files/sensors/doc/data_sheet/MMA7260QT.pdf

2.3 控制：这是本项目设计制作过程中最耗时的部分，但其原理很容易理解，姿态电路已经了解了飞机的姿态，控制电路只要作相应动作就好了。飞机前倾它就让飞机尾巴向前移动一些，后倾就向后移动一些。同样，还有左右，仔细想一下，这和杂技顶杆有何不同，用手指顶一根长杆，杆向左倒，手指就向左移动，右倒就右移。前后也一样。这个动作人实现简单，让一堆硅元素加上代码来实现还是有点难呢。这里还没讨论另外一个需要控制的量——高度，飞机在作此动作时高度是要保持稳定的，机身上冲时要减小油门，下降时要加大油门。这样比较下来，此项目比顶杆是要复杂多了。要完成如此控制，没的说，要用 MC9S08QG8，还得两片，MC9S08QG8 MCU 的规格如下

http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08QG8.pdf

为了达到上述功能，本项目应有如下面图六所示的硬件电路。

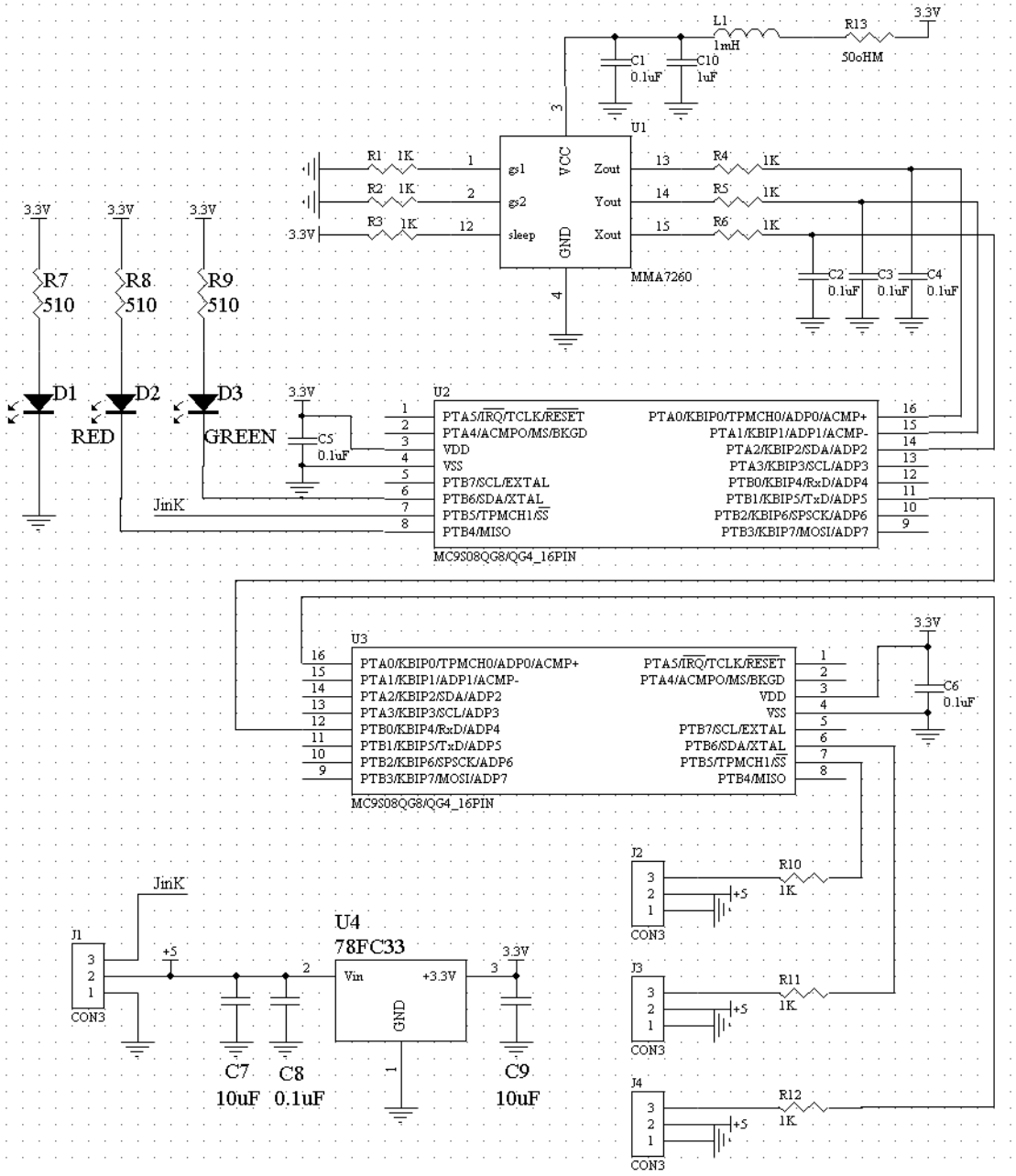


图六

2.4 选用两片 MC9S08QG8 是因为在本项目里有两处对时间比较严格的地方：一是读从遥控器送出的脉宽；二是把经过运算的脉宽精确地发送出去，两处都要 10ｕS 以内的误差，如用一片来做即使用中断也会有冲突等无法处理。本项目里两片工作的大体分配是：一片作为主控，用于读取遥控器送来的由接收机收到的脉宽（高实时性）、读取飞机姿态（低实时性）、控制运算（PID 运算）和数据发送（把需要对飞机怎么控制的数据，也就是上面 PID 运算的结果送给另一片 MCU）；另外一片作为辅助，只要把读到的数据实时发给飞机的执行机构就好了。

3 硬件描述

硬件电路见图七

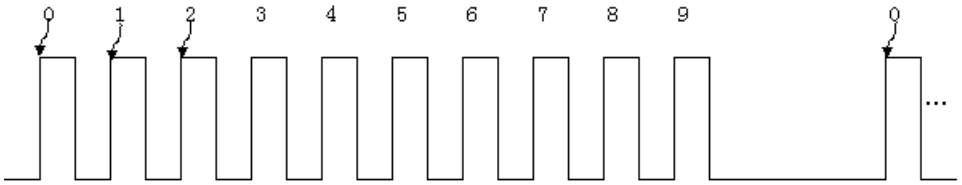


图七

3.1 传感器 MMA7260QT 输出的 3 个轴的数据分别送入主控 MCU 的 3 个 A/D 输入端 ADP0、ADP1 和 ADP2，电源接一个电阻和电感可防止电源对传感器的干扰。

3.2 整个电路的电源由 J1 从接收机引入，通过 78FC33 变为 3.3V 后给 MCU 和传感器供电，J2,J3,J4 分别是左右，油门和前后控制。

3.3 主控 MCU 的第二路 TPM 的捕捉输入端 TPMCH1 接接收机的信号输出端。本遥控系统是 9 通道的，也就是可同时控制 9 个机构，这种 9 通道的信号见下面图八



图八

它是由间隔为 20ms 的一个个脉冲串组成，每个脉冲串有 9 个脉冲，连续两个脉冲的上升沿的间隔就是遥控器送出的有效控制信号，此间隔最小 1ms，最大 2ms。TPM 的捕捉得到 9 个脉冲的宽度后，就可以根据需要直接送给舵机或 PID 运算后再送给舵机了。至于怎么识别某一脉冲是上一个脉冲串的结尾还是下一个脉冲串的开头，这里涉及到一个同步脉冲的概念，属于软件内容，在此先按下不表。

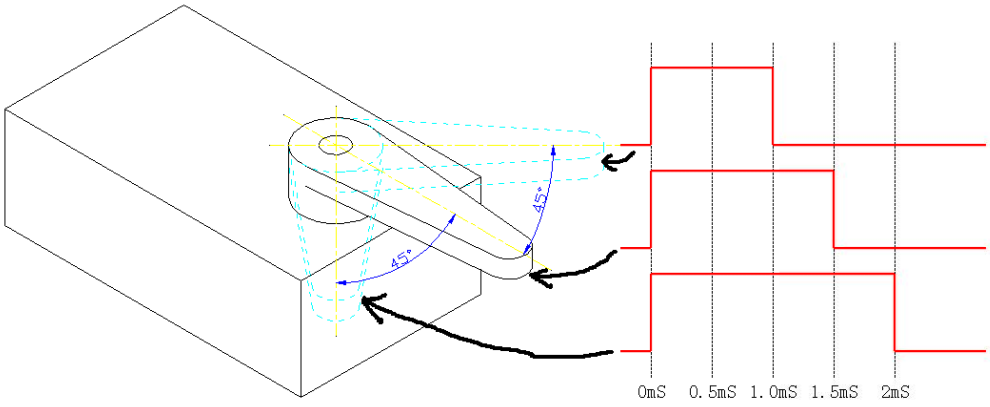
3.4 主控 MCU 的串行输出端接辅助 MCU 的串行输入端。主控 MCU 想怎么控制飞机动作，只要通过串口把数据发送给辅助 MCU 就好了，发送格式为：

0xAA	0x55	控制飞机前后动作的高位	控制飞机前后动作的低位	控制飞机左右动作的高位	控制飞机左右动作的低位	控制飞机油门的高位	控制飞机油门的低位
1	2	3	4	5	6	7	8

共 8 个字符，其中 0xAA、0x55 为引导位……这又属于软件内容了

3.5 三个 LED 好解释，D1 是电源指示灯，D3 用于指示“我是控制电路，我已经开始控制飞机竖直飞行了”，D2 是“出错了，小心出问题，快切换回由你控制吧。”

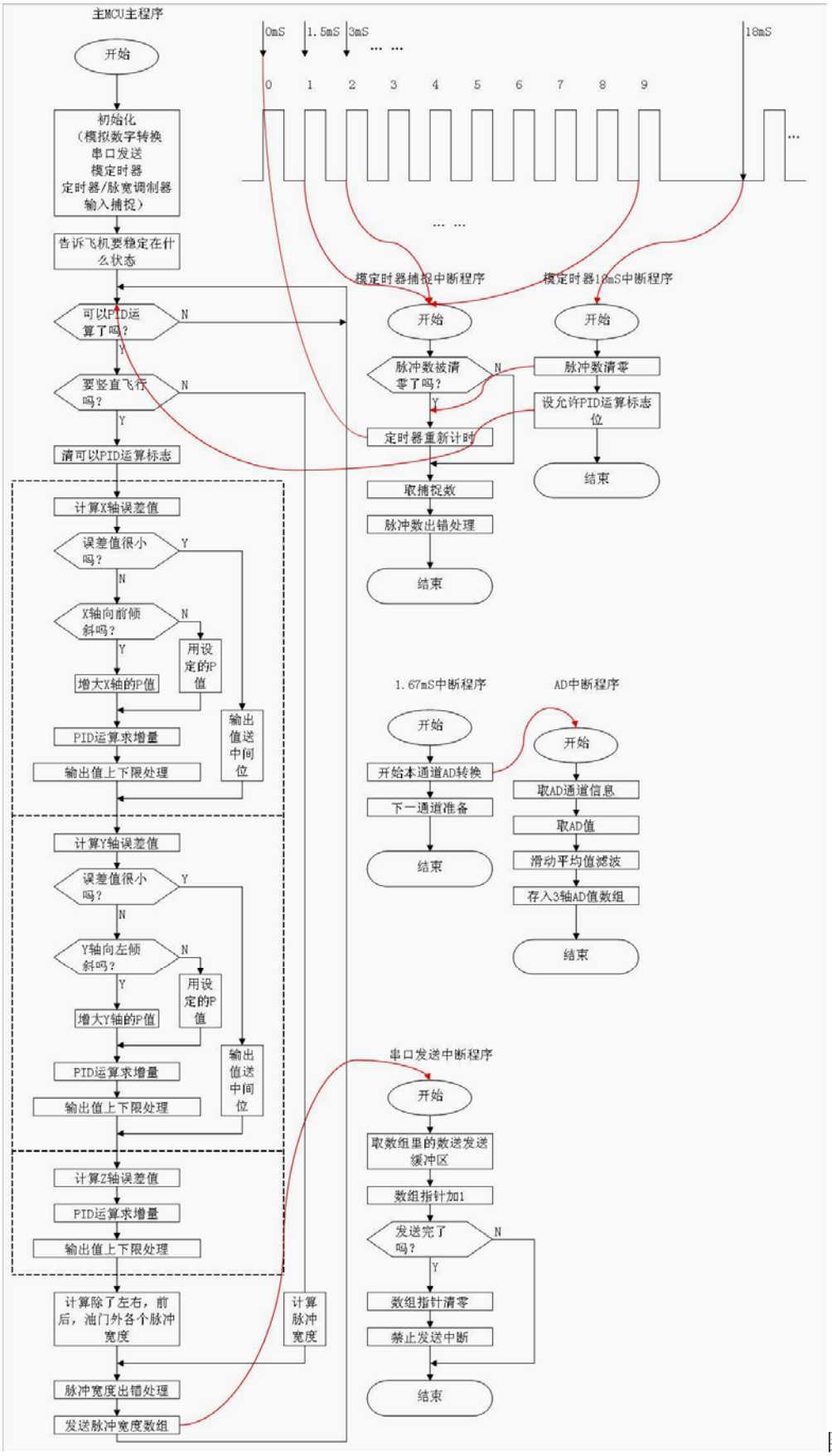
3.6 辅助 MCU 有两个工作：一是读主控 MCU 送来的串行数据，找到引导字符后，把前后左右上下（油门）的数据送入相应的数组贮存；二是每 20ms 把相应的数组里的数据以脉冲形式发送出去。脉冲最小 1ms，最大 2ms。这个信号送入飞机的执行机构——舵机后对应的输出规律见下图。1ms 的脉冲使舵机摇臂在最左边的位置；1.5ms 的脉冲使舵机摇臂位于中位；2ms 的脉冲使舵机摇臂在最右边的位置；舵机的摇臂连接飞机的各执行机构，摇臂摇动时飞机的执行机构如尾翼等也跟着摆动。



图九

4 软件描述

4.1 主控 MCU 部分，流程图见图十



图九

4.1.1 读取遥控信号：这个遥控信号见图十右上角，是 9 个脉冲组成的脉冲串，每 20ms 一串，读取的方法如下，由 TPM 捕捉到脉冲的上升沿进入模定时器捕捉中断程序后，先看是不是脉冲串里的第一个，如是就把定时器清零，然后把捕捉值存入数组，这样 9 次后，数组里就由低到高地存下了每个脉冲发生的时间，用减法一算，各脉冲宽度就出来了。为防止出错，加了个脉冲数错误预防，就是对脉冲个数进行计数，如果在一个脉冲串里读到了多于 9 个的脉冲那就是有干扰了，赶紧把所有脉宽按中间值 1.5ms 处理。这样虽然和遥控器发出的指令不一样，但减少了出大问题的机会。

4.1.2 18ms 定时中断：系统有一个模定时器 18ms 中断程序，因正常的脉冲串 9 个脉冲不会超过 18ms，所以可以让定时器在第一个脉冲开始时计数，计到 18ms 就认为一个脉冲串结束，把脉冲数计数器清零。模定时器捕捉中断程序里看到这个计数器被清零了就知道下一个脉冲串开始了。这个中断还为 PID 的定时运算提供时间标志，每进入一次中断就把标志置位，主程序读到标志被置位就进行 PID 运算，这样就保证了 PID 运算的周期是固定的。

4.1.3 1.67ms 中断程序和 AD 中断程序：为了读取 MMA7260QT 输出的数据需要用 AD 读值，这里每 1.6667ms 进入中断一次设定 AD 通道，读完 3 个通道用时 5ms，在两个脉冲串的标准间隔 20ms 内可以各读 4 次，用滑动平均值算法后可以滤除干扰。

4.1.4 主程序：

4.1.4.1 上面准备工作都做完了，主程序就开始工作了，初始化就不解释了，需要多说几句的是告诉飞机要稳定在什么状态这个子程序，我一开始就按照传感器水平放置时各轴的计算输出作为飞机要稳定的目标值，实际工作时飞机上窜下跳，几天的 DEBUG 后发现每次上电时，虽然传感器都是水平放置，可因为温度电压等的不同三个轴的输出和计算的输出是不一样的，会有少许偏差，为了解决这个问题，我在上电后先对三个轴的值连续读 64 次然后平均，把这个平均值作为飞机要稳定的目标值，这样飞机就基本稳定了。见下面连接的录像，因为当时只控制了油门，前后左右还没控制，所以在飞机的两侧粘了两根管子把前后左右限制住，用细铁丝穿过后绷紧，飞机就可以上下滑动了，这是控制好油门飞机就能“悬停”了。

http://www.tudou.com/programs/view/PqAPNmoWEfs/

```
for (i = 0; i < 3; i++){                    //三个轴
    AdcTimes = 0;                           //64 次的计数器清零
    AdcTmp = 0;                             //读出的 AD 值放在这里
    do{
```

```
        ADCSC1 = i + 1;              //设定 AD 通道
        while(!ADCSC1_COCO){;} //等它转换
        AdcTmp += ADCR;              //把这次的值加起来
    }while(++AdcTimes < 64);         //加够 64 次了吗
    Must[i] = AdcTmp / 64;}                  //加够了，取平均值
```

4.1.4.2 提到稳定，PID 功不可没，主程序里的 PID 有两处要提一下，一是因为刚刚提到的传感器的稳定性的关系还有飞机的抖动，AD 输出值差个 2-3 个数是很正常的事，所以在主程序里要判断一下，如果误差值在 3 以内，则不进行 PID 运算，这样就减少了舵机无谓的抖动；二是因为飞机的前后左右重量不一致，所以在飞机前倾后仰左右摆动时要用不同的 P 值计算。

4.1.4.3 主程序每 20ms 更新一次数据，数据就是前后左右上下的控制脉冲的宽度，这个数据是直接把模定时器捕捉中断程序里捕捉到的数拿来用还是要拿 PID 运算的数据由一个脉宽（决定此时飞机是否要竖直飞行的脉宽）决定，这里用到的是第 7 个脉冲的脉宽，也就是遥控器的第七通道，我把这个重要的参数命名为 Helen(这是我爱人的英文名字，以此来感谢她对我这次比赛的全力支持)，当 Helen > 1500（1.5ms）时就是操纵者要竖直飞行，要把 PID 运算的数据输出，当 Helen < 1500 时就是正常飞行，直接把模定时器捕捉中断程序里捕捉到的数输出就好了。

```
if(Helen > 1500){        //要竖直飞行？
    XPid();              //PID 运算
    YPid();
    ZPid();
}else{                   //正常飞行
    PlusWCalAll();       //直接把捕捉到的数输出
}
```

4.1.4.4 在 4.1.1 里提到要得到各个脉冲宽度只要把从低到高的每个脉冲发生的时间一减就好了，这个减法运算也是在主程序里完成的，减法完成后还要对脉冲宽度进行出错处理，要检查一下，小于 1ms 的按 1ms 算，大于 2ms 的按 2ms 算，处理完了就可以调用发送中断发送给辅助 MCU 了。调用方式为把数组排列好，把发送数据计数器清零，打开发送中断允许位，这时发送缓冲区是空的，发送中断一允许，它就立刻进中断发送去了。

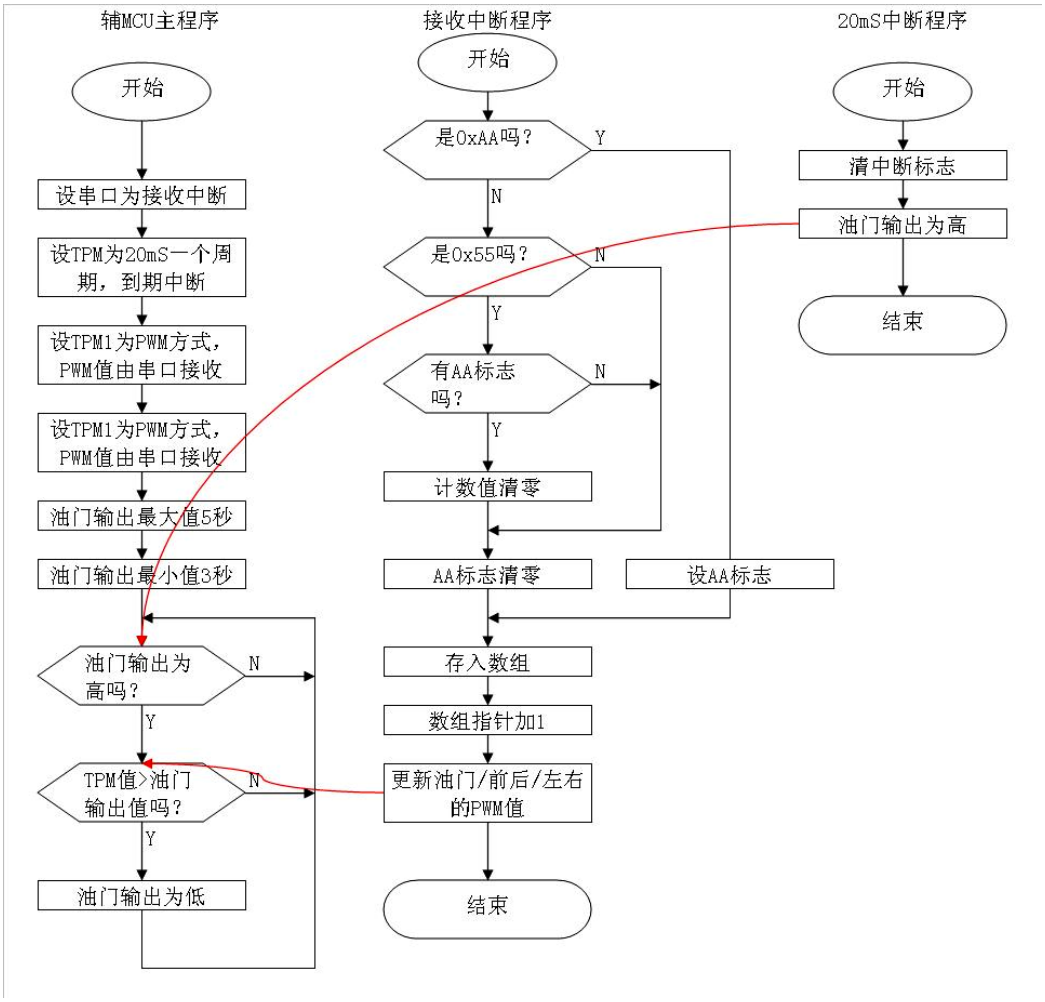
4.1.5 串口发送中断程序：程序进了这个中断后首先把发送数据计数器指示的数送

到发送缓冲区去发送，然后看看数组里的数发送完了吗，如发送完了就把发送中断允许位清零，以防止正在发送的数据发送完后再次进入中断；如没发送完就直接退出，等待下次中断。

4.2 辅助 MCU 程序，程序流程图见图十一

4.2.1 串口接收中断：辅助 MCU 程序的关键在串口接收中断，接收到的数要存入指定的数组，这个容易实现，要把引导头找出来要稍微动下脑筋，因为作为引导头的 0xAA,0x55 也有可能是数据，只有连起来的才是引导头，判断方法详见流程图。

4.2.2 因为本 MCU 只有两个定时器，可是需要控制三路脉冲，所以有两路脉冲用定时器的 PWM 功能实现，另外一路需要用软件模拟，我用的方法是用一个 20ms 的中断，进入中断后把 I/O 口置位，然后在主程序里循环判断定时器的值，当定时器的值大于要送的脉冲值时把 I/O 复位。



5 结语

本项目要实现的是一个非常复杂的现有的飞机还无法做出的动作，如能完成设计投入生产相信能推动相关领域更上一层楼！谢谢各位评委的细心阅读。

附件 1：关于飞机的一些小知识

一般的飞机由 6 部分组成，分别是螺旋桨，主翼，副翼，机身，垂直尾翼(方向舵)和水平尾翼(升降舵)，详见图 A，作用分别介绍如下

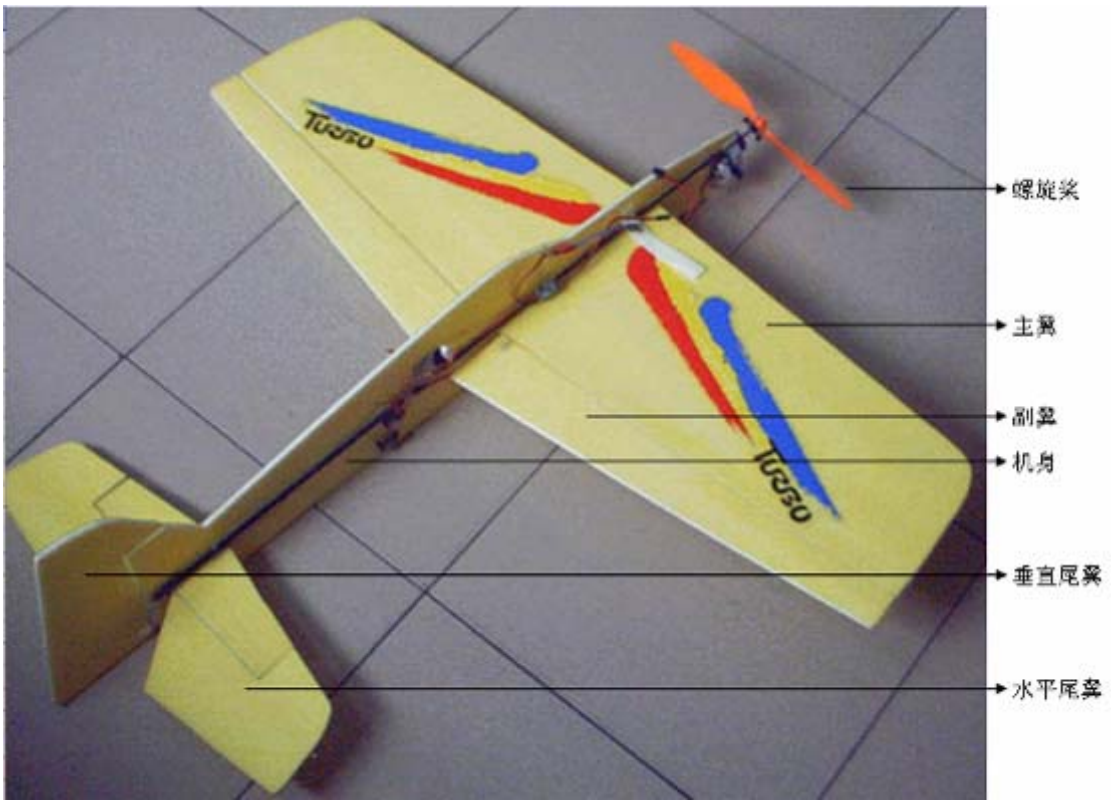


图 A

螺旋桨：飞机的前进动力，有了它飞机就能向前飞；
主翼：用来把飞机向前的速度转换为向上的升力，飞机就升起来了；
副翼：副翼的动作见图 B 图 C，当飞机前进时相对于飞机来讲向后吹的气流或螺旋桨吹过来的气流吹在副翼上时会产生使飞机沿着螺旋桨的拉力线横滚的力；

图 B　　图 C

机身：这个不用解释了，连接飞机各部分，动力设备，电子设备，驾驶舱都在这里了；
垂直尾翼(方向舵)：垂直尾翼的动作见图 D 图 E，当飞机前进时相对于飞机来讲向后吹的气流或螺旋桨吹过来的气流吹在垂直尾翼上时会产生一个让机尾摆动的力（作用在垂直尾翼上的力的水平分力）,见图 F，当垂直尾翼向左偏转时，螺旋桨吹过来的气流作用在垂直尾翼上

的力分解为两个力，水平分力就是让机尾向右的，机尾向右摆动了，也就相当于机头向左摆动了，简而言之，当垂直尾翼向左偏转时机头就向左摆动，当垂直尾翼向右偏转时机头就向右摆动，所以垂直尾翼又叫方向舵；

图 D　　图 E

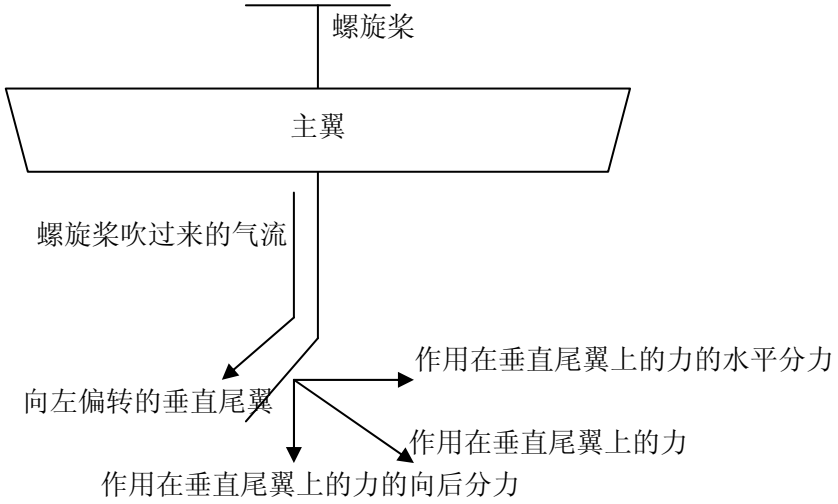


图 F

水平尾翼(升降舵)：水平尾翼的动作见图 G 图 H，它的动作原理同垂直尾翼的一样，只是垂直尾翼控制的是飞机的左右，水平尾翼控制的是飞机的上下，当水平尾翼向上偏转时机头就向上上升，当水平尾翼向下偏转时机头就向下下降，所以水平尾翼又叫升降舵。

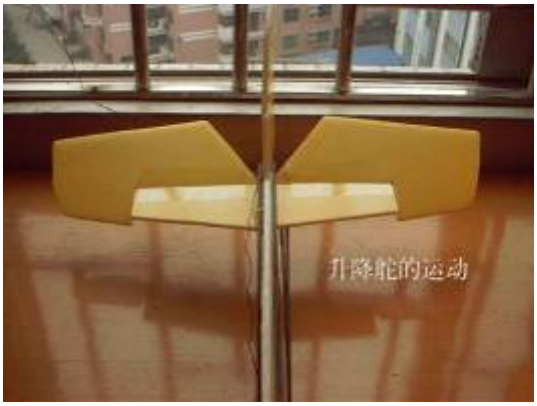


图 G　　图 H

以上介绍的各部分的功能是当飞机正常水平飞行时的，在本项目里，当飞机垂直于地面时，各部分的功能已经被扭曲，不再是正常的功能了，详述如下：

螺旋桨：它的拉力完全用于克服飞机的重力；

主翼：曾经用于产生升力的功臣现在没用了；
副翼：当飞机因为螺旋桨的旋转反向自旋时，它可以产生反自旋的力；
机身：嗯，它的作用还是没变，和正常飞行时一样；
垂直尾翼(方向舵)：还是让飞机左右摆动的，只不过正常飞行时是让飞机在水平面上左右摆动，现在是让飞机在垂直面上左右摆动；
水平尾翼(升降舵)：现在它是用来让飞机在垂直面上前后摆动了。

好了，现在可以在脑中描绘出这样一个情景了吧，“一架飞机竖直停在空中，它的升力由螺旋桨提供，副翼控制着自旋，垂直尾翼控制着左右，水平尾翼控制着前后，稳定的旋停着”，好了，飞机的知识介绍完毕。

附件 2

传感器 MMA7260QT 是加速度传感器吗

论点：文件 MMA7260QT Rev:2.06/2007(以下简称文件)里提到的传感器 MMA7260QT(以下简称传感器)不是加速度传感器，而应该是力传感器。

论据：当传感器水平静置时，它运动的加速度是零，而它的 Z 轴却有 800mv@1.5g 的输出，如果它是加速度传感器它的输出此时应是零。

论证：在论据里提到的例子如果以力传感器解释就完美了。

当传感器水平静置时，它只受到重力作用。它输出的是 beam(这个词不知如何翻译才准确)受到的重力。

下面详细讨论在某个轴上静止、匀速直线运动、匀加速直线运动和匀速圆周运动时它的输出分别如何：

1、 静止时

1.1 水平放置静止时(见图 1)：

图 1

为了模拟可动的 beam 会自动回中的特性，为它加了一绷紧的橡皮膜，此时传感器的加速度和速度都为零，只受重力和支持力，它们是一对平衡力，所以它保持静止。此时可动的 beam 的加速度和速度也都为零，只受重力和橡皮膜因弹性变形给它的支持力，这也是一对平衡力，所以它也保持静止。但因此时它已经因为重力的原因往下移动，离下 beam 近了些，因此，传感器有输出，输出按以下计算，

设橡皮膜弹性系数为 k，可动的 beam 在重力作用下向下移动了 D，则有

$$m_{beam}\, g_{重力加速度} = k\, D$$

所以，

$$D = m_{beam}\, g_{重力加速度} / k$$

当 D 变化时根据文件里提到的 $C=A\varepsilon/D$ 可知 C 就会变化，C 变化则传感器输出 V 变化。

小结：当因为重力作用橡皮膜给可动的 beam 支持力时有输出。

1.2 竖直放置静止时(见图 2)，

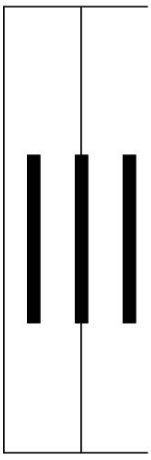


图 2

此时可动的 beam 位于中位传感器无输出，状态比较简单，不详细分析。

小结：当橡皮膜不给可动的 beam 施力时无输出。

2、 匀速直线运动

匀速直线运动的状态同竖直静止(见图 2)，中间的可动的 beam 没有在水平方向上受力，所以保持中立。

小结：当橡皮膜不给可动的 beam 施力时无输出。

3、 匀速圆周运动(见图 3)

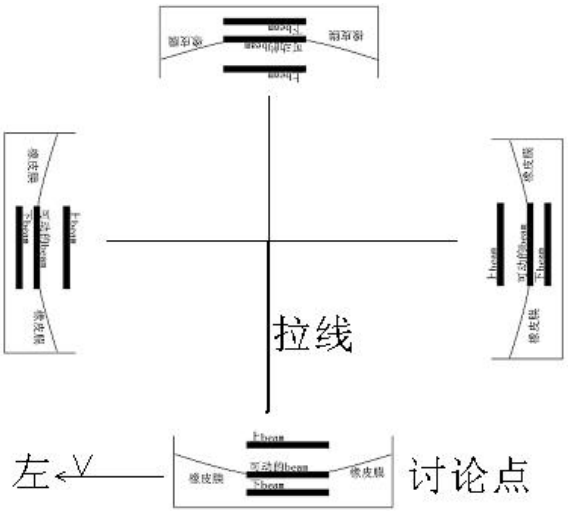


图 3

为了忽略重力影响，我们讨论在水平面内作匀速圆周运动的情况(顺时针运动，线长为 r)。设当传感器运行至讨论点时，它的线速度为 V，方向为水平向左，角速度为ω，则拉住传感器的线在此时对传感器的拉力也就是向心力

$$F_{传感器}=m_{传感器}r\omega^2$$

或

$$F_{传感器}=m_{传感器}V^2/r$$

这里 m 是指传感器的质量，包括外壳、3 个 beam、橡皮膜……

同时，由于可动的 beam 也是和传感器一起作匀速圆周运动，所以也受到一个力

$$F_{beam} = m_{beam} r\omega^2$$

这里的 m 是指可动的 beam 的质量。

不同的是传感器受到的向心力是拉线提供的，而可动的 beam 受到的向心力是由于橡皮膜变形产生的弹力提供的。

小结：当在圆周运动需要橡皮膜给可动的 beam 向心力时有输出。

4、匀加速直线运动

4.1 传感器竖直放置水平匀加速直线运动

设传感器如图 4 所示，竖直放置向右做匀加速直线运动，加速度为 a，则传感器需要拉线施加给它一个方向向右，大小为 $F_{传感器}=m_{本传感器}a$ 的力。

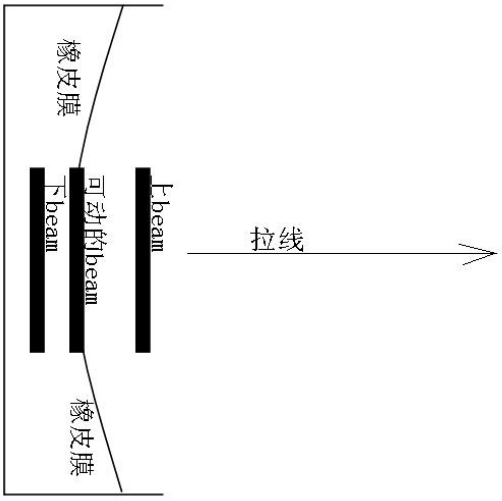


图 4

同时，由于可动的 beam 也是和传感器一起运动，所以它受到的力是

$$F_{beam}=m_{beam}a$$

传感器受到的力是拉线施加过来的，可动的 beam 受到的力是由于橡皮膜变形产生的弹力提供的。

小结：当橡皮膜提供向右的拉力时 beam 向右匀加速运动时有输出。

4.2 传感器水平倒置竖直匀加速运动(自由落体)

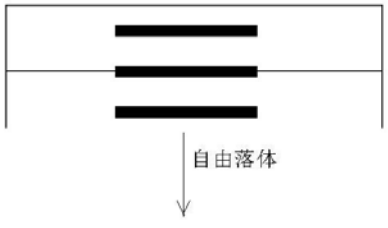


图 5

在上述运动条件时，传感器和可动的 beam 受力分别为

$$F_{传感器}=m_{传感器}g_{重力加速度}$$

和

$$F_{beam}=m_{beam}g_{重力加速度}$$

与 4.1 分析稍有不同，传感器受的力是由于它的质量在重力加速度的作用下产生，beam 受的力是由于 beam 的质量在重力加速度的作用下产生，不是橡皮膜提供。这时传感器是无输出的，因此可有以下小结。

小结：当橡皮膜没给可动的 beam 施力时无输出。

综上所述，得出下表：

状态	结论	输出
水平静止	橡皮膜给支持力	有
竖直静止	橡皮膜不施力	无
匀速直线运动	橡皮膜不施力	无
匀速圆周运动	橡皮膜给向心力	有
水平匀加速直线运动	橡皮膜给拉力	有
竖直匀加速直线运动	橡皮膜不施力	无

由上表总结为：无论传感器的运动状态为何，是否有加速度，只要橡皮膜对 beam 施力(不管此力是做支持力、向心力还是拉力)，则本传感器就有输出。

所以本传感器 MMA7260QT 应该是力传感器，而不是加速度传感器。