电机测速方法：检测A相 每332个电平变化为1rad。捕捉到第一个电平变化后打开定时器，打开计数器计数储存rad。定时间时间到达1S后停止计时、计数。

速度为（rad/332）转每秒。

定时器中断：一秒后进入中断 停止外部中断 计算得到速度

外部中断：用来累加电平变化（测量角度可以用到） 进入条件为电平发生改变|上升沿或者下降沿

接收到测速命令

定时器初始化为一秒同时打开外部中断

时间到 关闭定时器 关闭外部中断

测量转过的角度的方法：A相 外部捕获中断，角度计数器加一 转过角度为1.07\*计数器

不操作的话一直加(设置最大转动圈数 超过以后归零)

一般不用读取 当需要读取的时候再读取（标志位实现）

按照指定速度旋转：编写函数，

第一步：比较设定速度和当前速度的大小

第二步：进入PID算法得出瞬间的空占比

第三步：将空占比应用改变。

转动指定角度：每一个电平变化为转过了360/334=1.07度，编写函数，

以什么速度转动？（模型可能是前期速度大 即将到达目标值时减小 理想状态是速度为零时角度刚好调整到位|PID可能可以完成相关计算）

多大的速度？（交给PID计算？输入目标角度 检测当前转过的角度 计算得到误差 由PID算法得到当前误差下的PWM空占比输出）

另外一种思路：不用PID调整 设定角度，确认更改角度，速度设定为可以平稳转动的最小值（测试出来的）转动 捕获一次判断一次当前角度 相等时设定速度为零 一个调节周期完成

PID相关思考：

PID算法要共用吗？（共用 这样不需要关定时器了 参数不能共用 参数要调整 ）

画 速度波形：

速度对时间的函数图像 横向以一个像素点作为一个时间刻度 230个点为一个周期 周期 到达后清屏从零开始计数 纵向输出当前速度 输出的为点 用画线函数将点连成线指示波形

画 角度波形：

方案一 当前角度关于时间的波形（采样率不够 画出的图像不够平滑）

方案二 也可以画成圆盘 划线指示当前角度值 起点坐标为圆心 重点坐标需要计算

Printf函数的实现：

难点：未指定参数个数

实现功能：可以识别出格式控制符 数据类型识别符 将数据发送到串口打印

入口参数应该为一个数组

参数内部

上位机传入指令控制MCU：串口接收数据，收到指令处理

逐个字符分割进行与判断 符合格式后进入进行相应调整

掉电保存：

两个方案：EEPROM或者内部flash（依赖电容）|可以不用 在值发生改变的时候去写入就可以了 除了第一次初始化用人工赋值 除此之外其他时间都在上电的时候用EEPROM的储存值读取 | 第一次写入数据后提供标志位 每次上电数据初始化后检测相应标志位 如果成立则用EEPROM的数据覆盖初始化

模式切换逻辑关系：上电后初始化各个函数，打开显示器，设定速度以70转每秒的速度（上限）进行正向旋转（称为状态0）

接收到串口或矩阵按键关于状态切换的指令后设定速度为0，手动辅助将空占比调节为0 PID参数切换 电机静止，检测到输入速度后打开角度检测， PID开始工作调整

先设定角度 然后通过按键将调整的角度传入PID开始调整 调整后将设定的角度清零

按键10需要干的事情：标志位取反做出反应 屏幕指定位置显示 速度调整到0 更换PID参数并初始化PID 开始等待调整角度

按键4需要干的事情：载模式1读取当前的PID参数并初始化PID 在模式2下将设定的角度传入角度调整函数

使用GIT做版本控制：

git add .        （注：别忘记后面的.，此操作是把Test文件夹下面的文件都添加进来）

git commit  -m  "提交信息"  （注：“提交信息”里面换成你需要，如“first commit”）

git push -u origin master   （注：此操作目的是把本地仓库push到github上面，此步骤需要你输入帐号和密码）