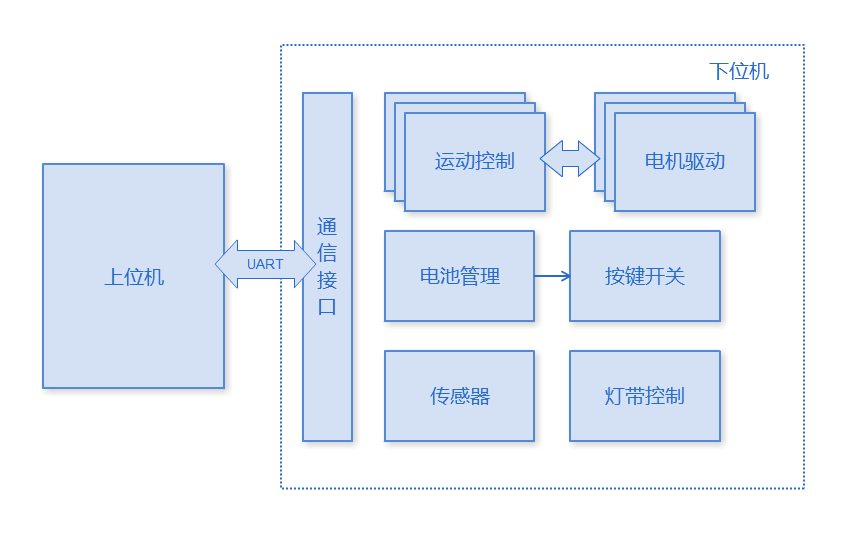
1. 总体框图

下位机功能主要包含电机的运动控制，按键和电池管理，灯带控制，传感器数据采样和处理几个部分，下位机的所有功能均设计为被动模式，也就是只会被动的接收指令并执行相应的功能，并在某些事件触发时向上位机发送通知，原则上不会保留状态信息或者主动进行一些操作或者运动。上位机与下位机的通信通过UART接口来进行。



1. 电机运动部分

主要包含电机驱动和运动控制两个部分。电机驱动完成给定速度和加速度驱动步进电机运动，运动控制基于驱动模块进行对速度或位置的控制，以达到平滑运动的目的，避免步进电机在过大的速度变化时导致失步的问题。

2.1. 电机驱动

2.1.1. 原理

步进电机通过STEP信号发出一个脉冲驱动电机运动一步，通过DIR信号控制步进电机方向。若希望电机按照设定的加速度a以及速度v运动，可以通过积分计算出运动距离s，并在s每增加一步或减一步的时候发出STEP脉冲和设置DIR方向信号：

2.1.2. 计算单位

为了优化计算的原因，在内部计算运动参数时使用统一的单位。

对于时间，使用积分时间为单位，积分时间是计算一次积分的时间，目前设置为25us，记为dt。对于距离，使用步为单位，记为step。相应的速度单位为step/dt，加速度单位为step/dt/dt。

2.1.3. 计算流程

在每个积分时间执行以下计算，更新速度和位置并输出STEP和DIR信号：

1. 更新速度：
2. 更新位置：
3. 如果，则输出STEP并更新DIR为正向，并执行；

如果，则输出STEP并更新DIR为负向，并执行step；

2.1.4. 速度和加速度限制

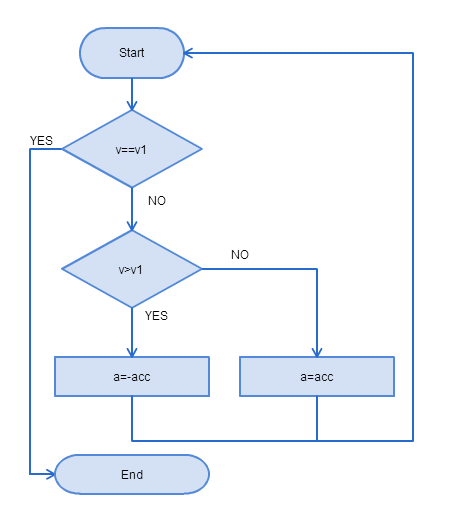
由于步进电机的输出扭矩与速度成反比关系，在较大的加速度和速度的情况下更容易出现失步的情况，因此需要将速度和加速度限制到一个范围，以避免失步情况的发生（速度和加速度的限制值需要通过测试得出）。

2.2. 运动控制

运动控制建立在电机驱动的基础上，控制电机以平滑的运动（在加速度限制和速度限制之内）以一定的速度曲线达到目标速度目标位置。

2.2.1. 速度控制

控制电机以一定的加速度acc达到目标速度v1，控制频率为1dt，当前速度为v，控制流程如下：

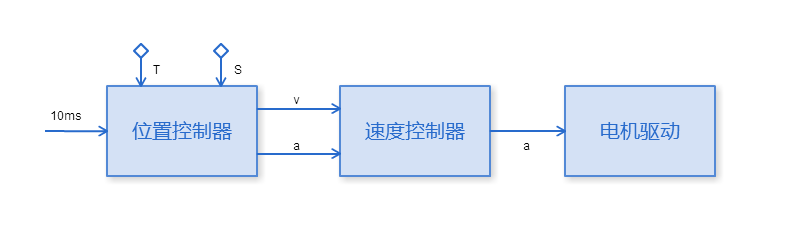


2.2.2. 位置控制

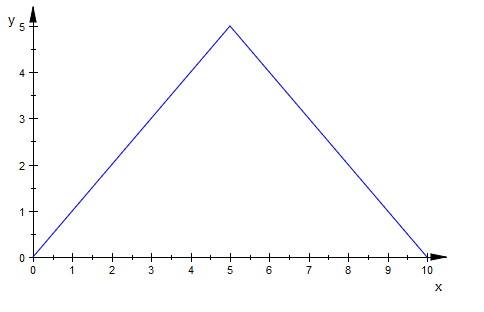
位置控制功能是驱动电机在指定的时间运动到指定位置。由于有速度和加速度限制，电机有可能不能完成指定的任务，因此引入了两种运动模式，1. 保证运动时间，也就是说在指定的运动时间内停止运动，但是极限情况下会运动不到位；2. 保证运动位置，也就是说一定会运动到指定的位置，但是极限情况下会超出时间。这两种运动模式，均需要保证可以在任意的初始速度下运动并且速度和加速度不超过限制。

1. 保证运动时间的控制

通过给定的时间的和位置，以及给定的速度曲线模式，在一定的控制频率下，得到当前所需的加速度和目标速度，驱动内环速度控制器（2.2.1），直到到达指定的时间（速度控制到0）。



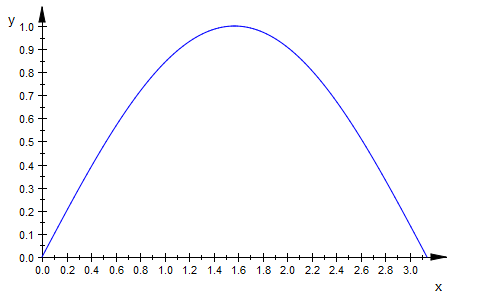
a). 一次速度曲线



在T时间内运动距离S，可计算得出速度，加速度

控制器的输出为：

b). SINE速度曲线

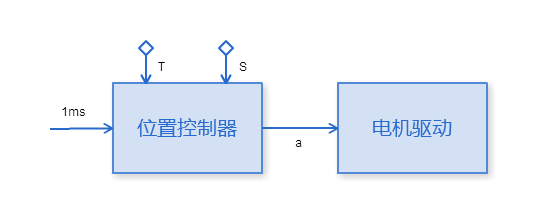


在T时间内运动距离S，可计算出速度曲线：；

因此控制器输出为：

1. 保证运动位置的控制

通过给定的时间的和位置，以及给定的速度曲线模式，在一定的控制频率下，得到当前所需的加速度，直接驱动运动器，直到到达指定的位置（并且速度到0）。

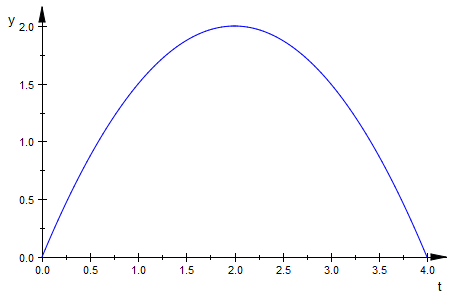


1. 一次速度曲线

这种模式下位置控制器的输出仅有三种状态：加速，减速，匀速。加速度值为给定值acc。

控制器计算流程可以描述为：

1. 如果目标位置已达到，并且速度到达0，则结束控制
2. 如果目标方向与速度方向相反，则输出a = acc，符号与v相反
3. 计算以当前的速度减速到0的距离s1
4. 如果目标位置与当前位置的距离ds<s1则减速，输出a=acc，符号与v相反；否则执行5
5. 计算以当前速度运动到目标位置的最短时间t2。
6. 如果t2>t，则加速，输出a=acc，符号与v相同；否则匀速，输出a=0；
7. 二次速度曲线



这种模式下，速度曲线可以设为，在v0的初始速度的条件下，经过时间T，运动距离S，可以得出：

求解得到：，

因此控制器的输出：

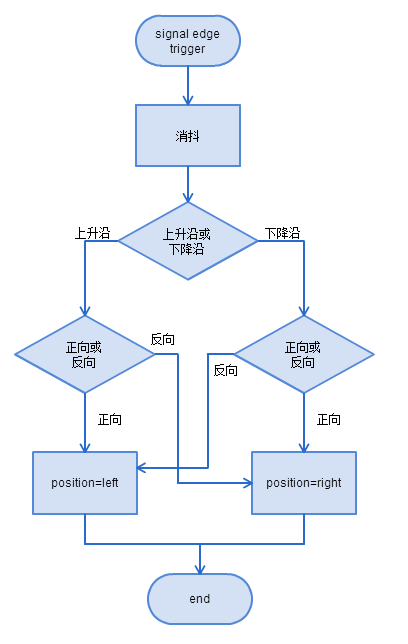
2.2.3. 复位

1. 复位传感器

机体运动使挡片经过复位传感器时，传感器会输出一个脉冲。如果电机正向运动，则脉冲信号的上升沿为挡片的右侧，下降沿为挡片的左侧；反向运动，脉冲信号的上升沿为挡片的左侧，下降沿为挡片的右侧。

2. 位置校正

当捕获到复位信号的上升沿或下降沿时，应该对电机的位置进行校准。假设挡片左侧对应的位置为left，右侧对应的位置为right，当前位置为position，校准流程如：



1. 机体正向位置

由于在唤醒时会改变机体的正向位置，需要维护自己的正向位置，在进行运动和复位时，均应相对于该位置。该正向位置记为forward，取值范围为-180~180度，初始值为0度，更新为forward\_new的流程如下：

* 1. 更新当前位置position+=(forward-forward\_new)
  2. 更新复位校正位置left+=(forward-forward\_new), right+=(forward-forward\_new)
  3. 更新正向位置forward = forward\_new

1. 硬复位

如果电机运动中失步或者受到外力而改变自身位置后，内部记录的位置信息已经不正确，则需要进行硬复位。硬复位需要使电机运动通过复位传感器并校正位置，然后运动到零位。

另外在上电后也应进行硬复位。

1. 软复位

直接通过位置控制器运动到零位。

1. 正向复位

直接进行硬复位或软复位，最终电机会运动到当前设置的正向位置上。如果需要复位到机体的初始位置则需要先将正向位置设置为0，然后再进行复位操作。

2.3. 按键开关

2.3.1. 按键短按检测

在每次按键上升沿和下降沿的时候触发中断并记录当前的按键状态和时间，在当前状态为按键松开并且上一次的状态为按下时，判断时间间隔，如果时间间隔大于短按时间（50ms）并小于长按时间（4000ms），则触发短按事件。

2.3.2. 按键长按检测

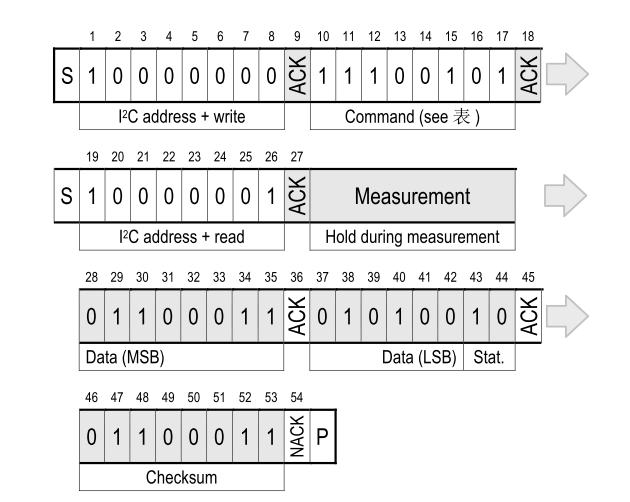
在tick中断中检测当前时间和上一次触发按键上升沿或下降沿的时间之差，如果值大于长按时间（4000ms）则触发长按事件。

2.4. 温湿度传感器SHT2x

2.4.1 I2C

系统通过I2C接口来读取温湿度传感器SHT2x数据。MCU 与传感器之间的通讯有两种不同的工作方式：主机模式或非主机模式。在第一种情况下，在测量的过程中， SCL 线被封锁（由传感器进行控制），在第二种情况下，当传感器在执行测量任务时，SCL 线仍然保持开放状态，可进行其他通讯。非主机模式允许传感器进行测量时在总线上处理其他 I2C 总线通讯任务。

系统主要使用主机模式，在主机模式下测量时，SHT2x 将 SCL 拉低强制主机进入等待状态。通过释放 SCL 线，表示传感器内部处理工作结束，进而可以继续数据传送。



其中COMMAND字段代表测量命令，包括0XE3:温度测量，0XE4:湿度测量

DATA字段代表测量值：温度为14bit，湿度为12bit

2.4.2. 数据处理

1. 相对湿度 RH 都可以根据输出的DATA相对湿度数据通过如下公式计算获

得 ，其中RH代表相对湿度，单位为%：

2. 温度值T可以通过将温度输出信号代入到下面的公式计算得到（结果以温

度°C 表示）：

2.5. 空气质量传感器

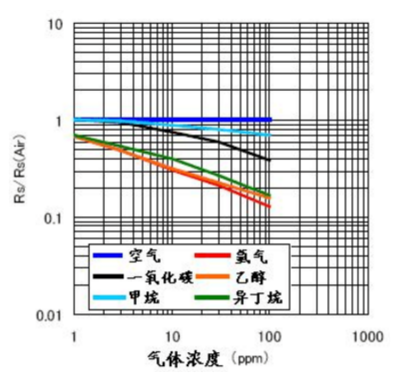
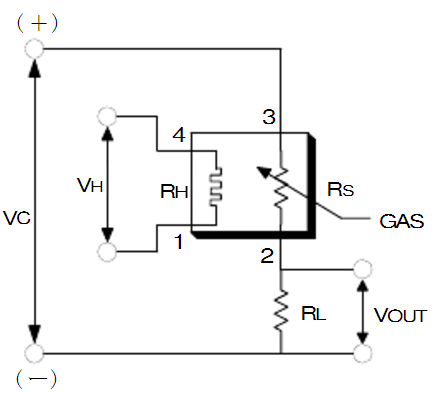
2.5.1. ppm计算

TGS2600由集成的加热器以及在氧化铝基板上形成的金属氧化物半导体构成。当可检知的气体存在时，空气中该气体的浓度越高，传感器的电导率就越高。使用简单的电路就可将这种电导率的变化变换为与气体浓度对应的输出信号。

下图是典型的灵敏度特性，全部是在标准试验条件下得出的结果。纵轴以传感器电阻比Rs/Rs(Air)表示，Rs､Rs(Air)的定义如下：

Rs＝各种浓度气体中的传感器电阻值

Rs(Air)＝清洁大气中的传感器电阻值

在上图的传感器工作电路下，通过ADC采集VOUT输出可以计算出Rs，计算方式如下：

通过对传感器的灵敏度特性进行拟合，生成一条大致的曲线，则可以通过Rs的值，估算出敏感气体的浓度ppm。

2.5.2. aqi计算

通过下表进行AQI的近似计算：

|  |  |  |
| --- | --- | --- |
| PPM,(Clow~Chight) | AQI(Ilow~Ihight) | AQI\_LEVEL |
| 0.0-4.4 | 0-50 | GOOD |
| 4.4-9.4 | 50-100 | MODERATE |
| 9.4-12.4 | 100-150 | UNHEALTHY FOR SENSITIVE |
| 12.4-15.4 | 150-200 | UNHEALTHY |
| 15.4-30.4 | 200-300 | VERY UNHEALTHY |
| 30.4-40.4 | 300-400 | HAZARDOUS |
| 40.4-50.4 | 400-500 | HAZARDOUS |

计算公式为：

其中I为空气质量指数，C为气体浓度（ppm），Ihigh，Ilow，Chigh，Clow为C对应范围的最大和最小值。

2.6. 灯带控制

一圈灯带包含20个LED灯（有RGB三色），每个LED可以独立的进行开关控制，并共享同一个RGB亮度控制，通过调节PWM的占空比来调节灯带RGB值。

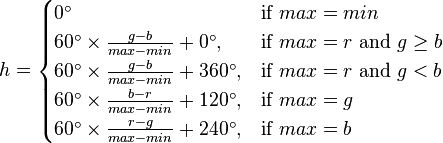
2.6.1. 呼吸效果

1. RGB与HSL

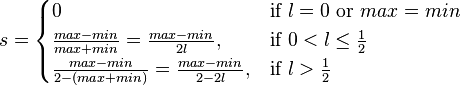
HSL 表示 **h**ue（[色相](http://zh.wikipedia.org/wiki/%E8%89%B2%E7%9B%B8)）、**s**aturation（[饱和度](http://zh.wikipedia.org/wiki/%E8%89%B2%E5%BA%A6_(%E8%89%B2%E5%BD%A9%E5%AD%A6))）、**l**ightness（[亮度](http://zh.wikipedia.org/wiki/%E4%BA%AE%E5%BA%A6)）。呼吸效果中，仅希望改变亮度，不改变颜色值。

从RGB转换为HSL：

设 (r, g, b) 分别是一个颜色的红、绿和蓝坐标，它们的值是在 0 到 1 之间的实数。设 max 等价于 r, g 和 b 中的最大者。设 min 等于这些值中的最小者。要找到在 HSL 空间中的 (h, s, l) 值，这里的 h ∈ [0, 360）是角度的色相角，而 s, l ∈ [0,1] 是饱和度和亮度，计算为：



l = \begin{matrix} \frac{1}{2} \end{matrix} (max + min)



h 的值通常规范化到位于 0 到 360°之间。而 h = 0 用于 max = min 的（就是灰色）时候而不是留下 h 未定义。

HSL 和 HSV 有同样的[色相](http://zh.wikipedia.org/wiki/%E8%89%B2%E7%9B%B8)定义，但是其他分量不同。HSV 颜色的 s 和 v 的值定义如下：

s = \begin{cases}0, & \mbox{if } max = 0 \\\frac{max - min}{max} = 1 - \frac{min}{max}, & \mbox{otherwise}\end{cases}

v = max \,

从HSL转换为RGB

给定 HSL 空间中的 (h, s, l) 值定义的一个颜色，带有 h 在指示色相角度的值域 [0, 360）中，分别表示饱和度和亮度的s 和 l 在值域 [0, 1] 中，相应在 RGB 空间中的 (r, g, b) 三原色，带有分别对应于红色、绿色和蓝色的 r, g 和 b也在值域 [0, 1] 中，它们可计算为：

首先，如果 s = 0，则结果的颜色是非彩色的、或灰色的。在这个特殊情况，r, g 和 b 都等于 l。注意 h 的值在这种情况下是未定义的。

当 s ≠ 0 的时候，可以使用下列过程：[[1]](http://zh.wikipedia.org/wiki/HSL%E5%92%8CHSV%E8%89%B2%E5%BD%A9%E7%A9%BA%E9%97%B4#cite_note-0)

q=\begin{cases}l \times (1+s), & \mbox{if } l < \frac{1}{2} \\l+s-(l \times s), & \mbox{if } l \ge \frac{1}{2}\end{cases}

p = 2 \times l - q \, 

h_k = {h \over 360} \, （*h* 规范化到值域 [0,1）内）

t_R = h_k+\frac{1}{3} \, 

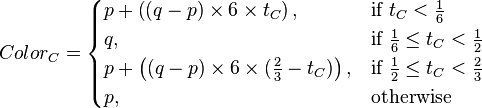
t_G = h_k \, 

t_B = h_k-\frac{1}{3} \, 

\mbox{if } t_C < 0 \rightarrow t_C = t_C + 1.0 \quad \mbox{for each}\,C \in \{R,G,B\}

\mbox{if } t_C > 1 \rightarrow t_C = t_C - 1.0 \quad \mbox{for each}\,C \in \{R,G,B\}

对于每个颜色[向量](http://zh.wikipedia.org/w/index.php?title=%E5%9D%90%E6%A0%87%E5%90%91%E9%87%8F&action=edit&redlink=1) Color = (ColorR, ColorG, ColorB) = (r, g, b),



\mbox{for each}\,C \in \{R,G,B\}

2. 流程

开始时，设置RGB颜色值，并计算出HSL值，在每个FRAME对HSL中的L值进行增加或者减小，并从HSL反向计算出RGB，并设置到灯带上，直到时间达到。

2.6.2. 跑马灯

跑马灯的流程为设置一个RGB值，初始化一个亮段的起始位置为start，长度为lenth，在每个FRAME中对start值进行增加或减少，并根据当前的start和length信息，计算出20个led的亮灭状态进行设置，直到时间结束。

2.6.3. 情景模式

所有情景模式均为呼吸灯，跑马灯，常亮，常灭，闪烁的状态组合。

1. 开机效果：先常亮蓝色3s，然后跑马灯，直到开机完成
2. 低电量效果：常亮橙红色
3. 充电中：绿色呼吸灯
4. 网络异常：红色常亮
5. 网络连接成功：绿色跑马灯
6. 提醒效果：紫色闪烁5s
7. 警报效果：红蓝间隔闪烁5s
8. 关机效果：蓝色跑马灯10s
9. 聆听效果：白色呼吸
10. 说话效果：白色跑马灯
11. 音乐效果：七色交替闪烁

2.7. 通信部分

2.7.1. 协议

数据包格式：其中SYNC为同步头0x2B（单字节），CMD为命令代码（单字节），LENGTH为数据长度（单字节），DATA1~n（LENGTH个字节），CHKSUM为校准字节（单字节）



2.7.2. 接收命令说明

1. CMD\_MOVE\_RELATIVE

命令代码为1，功能为驱动电机在指定时间内运行到指定相对位置。DATA字段的结构为：



MotorID：电机ID，三个电机从下到上依次为0，1，2

Angle：运动的相对角度，单位度

Time：运动的时间，单位秒

1. CMD\_MOVE\_ABSOLUTE

命令代码为2，功能为驱动电机在指定时间内运动到指定的绝对位置。DATA字段的结构为：



MotorID：电机ID，三个电机从下到上依次为0，1，2

Angle：运动的绝对角度，单位度

Time：运动的时间，单位秒

1. CMD\_MOVE\_POSITION

命令代码为3，功能为驱动电机在指定时间内运动到指定的绝对位置（-180~180），并选择最短的距离运动。DATA字段的结构为：



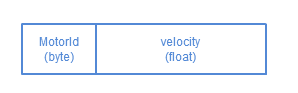
MotorID：电机ID，三个电机从下到上依次为0，1，2

Angle：运动的绝对角度，单位度

Time：运动的时间，单位秒

1. CMD\_MOVE\_SPEED

命令代码为5，功能为驱动电机以指定的速度运动。DATA字段的结构为：

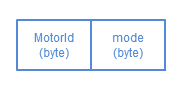


MotorID：电机ID，三个电机从下到上依次为0，1，2

Velocity：运动速度，单位度/秒

1. CMD\_MOVE\_RESET

命令代码为6，功能为驱动电机复位。DATA字段的结构为：



MotorID：电机ID，三个电机从下到上依次为0，1，2

Mode：复位模式：0-硬复位，1-软复位，2-圈数复位

1. CMD\_FORWARD\_SETTING

命令代码为8，功能为设置正向位置。DATA字段的结构为：



Type: 0-相对位置，1-绝对位置

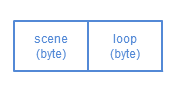
Angle：正向位置角度，单位度

1. CMD\_POWER\_DOWN

命令代码：0x50，功能为关机，无DATA

1. CMD\_LED\_SCENE\_CONTROL

命令代码0x51，功能为灯带控制，DATA字段的结构为：



Scene：情景模式，0-关闭灯带，1-开机，2-电量低，3-充电中，4-网络连接失败，5-网络连接成功，6-提醒，7-警报，8-关机，9-听，10-说，11-音乐。

Loop：是否循环播放

1. CMD\_READ\_POSITION

命令代码0x60，功能为读取电机的位置信息，无DATA字段。

收到该命令后，向上位机回发数据CMD\_UPDATE\_POSITION

1. CMD\_READ\_FORWARD

命令代码0x61，功能为读取目前的正向位置，无DATA字段。

收到该命令后，向上位机回发数据CMD\_UPDATE\_FORWARD

1. CMD\_READ\_MOTOR\_STOP

命令代码0x62，功能为获取目前电机是否均停止，无DATA字段。

收到该命令后，当电机运行停止时向上位机回发数据CMD\_MOTOR\_ALL\_STOP

1. CMD\_READ\_STATUS

命令代码0x63，功能为读取目前的传感器状态，无DATA字段。

收到该命令后，向上位机回发数据CMD\_UPDATE\_STATUS

1. CMD\_READ\_INFORMATION

命令代码0x64，功能为读取目前的版本信息，无DATA字段。

收到该命令后，向上位机回发数据CMD\_UPDATE\_ INFORMATION

1. CMD\_READ\_BATTERY\_STATUS

命令代码0x65，功能为读取目前的电池状态信息，无DATA字段。

收到该命令后，向上位机回发数据CMD\_UPDATE\_ BATTERY\_STATUS

2.7.3. 发送数据说明

1. CMD\_MOTOR\_ALL\_STOP

命令代码为2，表示电机已运行停止。该数据在电机停止时每2秒发送一次。

DATA字段的结构为：



Forward：正向位置，单位度

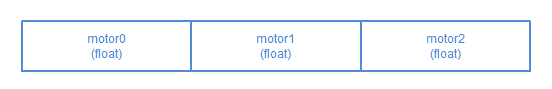
Motor0：电机0的位置，单位度

Motor1：电机1的位置，单位度

Motor2：电机2的位置，单位度

1. CMD\_UPDATE\_POSITION

命令代码0x60，在收到CMD\_READ\_POSITION命令后发送，DATA结构为：



Motor0：电机0的位置，单位度

Motor1：电机1的位置，单位度

Motor2：电机2的位置，单位度

1. CMD\_UPDATE\_FORWARD

命令代码0x61，在收到CMD\_READ\_FORWARD命令后发送，DATA为forward(float)信息，代表当前的正向位置，单位度。

1. CMD\_UPDATE\_STATUS

命令代码0x63，在收到CMD\_READ\_STATUS命令后发送，或每5秒发送一次，用来上传传感器状态等数据信息，DATA结构为：



Temperature：温度，单位摄氏度

Humidity：相对湿度，单位%

Aqi：空气质量指数

Aqi\_level：空气质量等级，1-6，数值越高，空气越差

Battery：电池电量，单位%

1. CMD\_UPDATE\_INFORMATION

命令代码0x64，在收到CMD\_READ\_INFORMATION命令后发送，DATA字段结构为：



Version：当前固件的版本号

Chksum：当前固件的校验码

1. CMD\_UPDATE\_BATTERY\_STATUS

命令代码0x65，在收到CMD\_READ\_BATTERY\_STATUS命令后发送，DATA字段结构为：



Voltage：电池电压，单位毫伏

Temperature：电池温度，单位摄氏度

Current：电池电流，单位毫安

Status：电池状态

Percent：电池电流，单位%

1. CMD\_NOTIFY

命令代码0xF0，向上位机发送通知，DATA字段为通知码notify（单字节）。Notify的类型包括：0-插上适配器，1-拔掉适配器，2-短按电源键，3-长按电源键