



Exclusively designed for the FluxMaster M2025 P19 Brushless DC Motor and C620 Brushless DC Motor Speed Controller, this M2025 Accessories KIT includes several cables and a terminal block.

Reflexes System Expansion Manual,  
Reflexes System User Manual, Introduction  
of Reflexes System Module

The 10000 Alexander 12 includes several cables and a control panel, creating a complete propulsion system driven by two

# 木鸢通讯协议BCT 用户手册



# 目录

<b>1. 简介.....</b>	<b>3</b>
<b>2. 通讯协议介绍.....</b>	<b>4</b>
2.1 通讯帧基本格式.....	4
2.2 基本配置.....	5
2.2.1 快速开始 .....	5
2.2.2 输入源设置.....	5
2.2.3 COM 设置 .....	5
2.2.4 波特率配置.....	6
<b>3. 数据帧介绍.....</b>	<b>7</b>
3.1 心跳数据.....	7
3.2 云台控制.....	7
3.3 枪管控制.....	8
3.4 模式控制.....	8
<b>4. 数据帧速查.....</b>	<b>9</b>
4.1 目标地址速查 .....	9
4.2 功能码速查 .....	9

# 1. 简介

RoboMaster 机甲大师赛，是由大疆创新发起，专为全球科技爱好者打造的机器人竞技与学术交流平台。自 2013 年创办至今，始终秉承“为青春赋予荣耀，让思考拥有力量，服务全球青年工程师成为践行梦想的实干家”的使命，致力于培养具有工程思维的综合素质人才，并将科技之美、科技创新理念向公众广泛传递。

木鸢 Birdiebot 战队作为以工科为主导，多领域交叉合作的学生组织，致力于打造学校范围内学科覆盖面最广、管理制度最全面、具有一定区域影响力的一流高水平现代化工程应用型队伍，培养中国制造 2025 卓越的预备工程师。

木鸢 Birdiebot 使用 BTP&DM (Birdiebot Target Prception And Decision Making Framework, Birdiebot 目标感知与决策框架) 对赛场复杂的环境进行感知与决策。为了适应 Robomaster 赛场多种数据类型的传输，保证高效的通信效率，所有在 BTP&DM 数据的通信，均需要遵守 BCP (Birdiebot Communication Protocol, Birdiebot 通讯协议) 格式。本协议在确保通信高效、源码简单、可移植性高的基础上，实现数据正确性判断，有效避免数据传输过程中出现的错误数据的错误解析。

## 2. 通讯协议介绍

### 2.1 通讯帧基本格式

DATA 数据内容中的数据，采用小端模式，低字节在前，高字节在后。

帧头	目标地址	功能码	数据长度	数据内容	和校验	附加校验
HEAD	D_ADDR	ID	LEN	DATA	SC	AC
1	1	1	1	N	1	1
0xFF 固定值	本帧数据发送的接收机器人，参考硬件地址定义	机器人实现的目标位置	表示数据内容字段的长度	具体数据内容，n=数据长度	计算方法	附加后

为了提高数据传输的效率，当有浮点数类型数据需要传输时，根据数据类型的特点，适当截取小数点后固定几位，将浮点数转化成整数类型进行传输，可缩短数据长度，并且避免浮点数传输时发生异常，解析成非法浮点数。类似数据会在协议中标注，如  $A \times 100$ ，就代表将数据 A 只保留两位小数，乘以 100 进行传输，下位机使用时将收到的数据除以 100 即可。

■ 和校验 SUM CHECK 计算方法：从帧头 0xFF 字节开始，一直到 DATA 区结束，对每一字节进行累加操作，只取低 8 位

■ 附加校验 ADD CHECK 计算方法：计算和校验时，每进行一字节的加法运算，同时进行一次 SUM CHECK 的累加操作，只取低 8 位。

■ 校验计算示例：假设数据帧缓存为 data\_buf 数组，0xFF 存放于数组起始位置，那么 data\_buf[3] 存放的是数据长度，校验程序如下：

```
def sumcheck_cal(self):
    sumcheck = 0
    addcheck = 0
    for i in [(k, v) for k, v in self.INFO.items()][:3]:
        sumcheck += i[1]
        addcheck += sumcheck

    for i in self.INFO["DATA"]:
        sumcheck += i
        addcheck += sumcheck

    self.INFO["SUM_CHECK"] = int(sumcheck) & 0xFF
    self.INFO["ADD_CHECK"] = int(addcheck) & 0xFF
```

## 2.2 基本配置

### 2.2.1 快速开始

通过 clone BTP&DM 可以快速使用 BCP 搭建机器人上位机系统。

```
>>> git clone https://gitee.com/zyhbum/birdiebot2021.git
>>> cd birdiebot2021
>>> python3 main.py
```

BCP 使用 UART/USART 进行上下位机交互，通过设置 UART/USART 信息，能够快速实现消息的收发。这些硬件设置在不同平台上可能有所不同，可以根据具体设备型号和下位机类型，进行收发和控制。

硬件设置位于 BTP&DM/config/devConfig 中，包括了输入源设置、COM 设置、波特率设置、超时设置。

### 2.2.2 输入源设置

BTP&DM 在设计中，考虑到真实比赛环境及调试需求，可以选择不同的图像输入源进行操作，目前 BTP&DM 支持一下的图像输入源：

- 视频文件：使用视频或文件作为输入源。将 source 设置为文件路径即可，例如：source=test.mp4
- 普通 USB 相机：使用 USB 相机作为输入源。将 source 设置为 openCV 中设备编号即可，例如：source=0
- 工业摄像机：使用工业摄像机作为输入源，目前，BTP&DM 仅实现了对大恒工业相机的支持。将 source 设置为相机编号即可，例如：source="DAHENG0"

### 2.2.3 COM 设置

#### 2.2.3.1 Window 设备配置

在 NUC/普通 X86 设备的 Windows 系统下，UART/USART 应使用 USB to TTL 模块进行转发（未保障数据的正确传输，推荐使用 FT232 芯片的转接模块，最好不要使用 CH340 芯片的模块）。安装驱动后，在命令行中使用 mod 命令，查看模块占用端口。

```
>>>mod

设备状态 COM23:
-----
波特率:      19200
奇偶校验:    None
数据位:      8
停止位:      1
超时:        OFF
XON/XOFF:    OFF
```

```
CTS 握手: OFF
DSR 握手: OFF
DSR 敏感度: OFF
DTR 电路: ON
RTS 电路: ON
```

设备状态 CON:

-----

```
行: 9001
列: 120
键盘速度: 31
键盘延迟: 1
代码页: 936
```

例如，在该设备中，可配置 `PORTX = "COM23"`

### 2.2.3.2 Jetson Ubuntu 设备配置

在 Manifold、Jetson AGX 等开发套件中，能够通过开发套件中自带的引脚，完成相关 COM 设置，在 Manifold2-G Ubuntu16.04 条件下，UART0 在系统内核中对应的设备是 `ttyS0`，UART1 对应的设备是 `ttyTHS2`。Manifold 2-G 的 UART 接口最大支持 3M 波特率。其中，921600 波特率为标称值，实际为 910000，在使用 921600 波特率进行传输时，这可能导致此 UART 接口无法与其他波特率亦有误差的设备(如一些 OSDK 设备)进行通讯。此时建议切换至其他波特率使用。

例如，在 Manifold2-G 使用 UART1 口，可配置 `PORTX = "/dev/ttyTHS2"`。

## 2.2.4 波特率配置

BCT 波特率指的是有效数据讯号调制载波的速率，即单位时间内载波调制状态变化的次数。在使用 BCT 时，只需将波特率调整为与下位机匹配即可，默认为 500000，但推荐不应小于 115200。

### 2.2.4.1 超时设置

BCT 在接收下位机时的等待操作时间，默认为 0，可根据实际场景自行设置，None 代表永远等待操作，0 为立即返回请求结果，其他值为等待超时时间(单位为秒)。

## 3. 数据帧介绍

### 3.1 心跳数据

为保证上位机与下位机正常通讯，在 BTP&DM 开始工作时，会始终向下位机发送心跳数据。心跳数据默认每隔 50ms 发送，数据内容为 0/1 交替发送。当下位机一段时间无法收到心跳数据时，说明 BTP&DM 与下位机连接以断开，应执行相应操作。

帧头	目标地址	功能码	数据长度	数据内容	和校验	附加校验
HEAD	D_ADDR	ID	LEN	DATA	SC	AC
0xFF	机器人编号	0xAA	2	格式如下	程序计算	程序计算

数据类型	UINT8
数据内容	交替发送 0/1

### 3.2 云台控制

发送云台偏移位置角度，角度为角度制下云台偏转相对角度，可使用以下语句快速发送：

```
self.robot.gimbal(yaw_angle, pitch_angle)
```

帧头	目标地址	功能码	数据长度	数据内容	和校验	附加校验
HEAD	D_ADDR	ID	LEN	DATA	SC	AC
0xFF	机器人编号	0x02	4	格式如下	程序计算	程序计算

数据类型	UINT8	UINT16	UINT8	UINT16
数据内容	Yaw 轴偏转角度正负	Yaw 轴偏转角度 绝对值	Pitch 轴偏转角度 正负	Pitch 轴偏转角度 绝对值



## 3.3 枪管控制

发送枪口是否发射弹丸，可使用以下语句快速发送：

```
self.robot.barrel(speed, isShoot)
```

帧头	目标地址	功能码	数据长度	数据内容	和校验	附加校验
HEAD	D_ADDR	ID	LEN	DATA	SC	AC
0xFF	机器人编号	0x04	2	格式如下	程序计算	程序计算

数据类型	UINT8	UINT8
数据内容	枪口发送速度	枪口是否发射

## 3.4 模式控制

发送是否由上位机接管，可使用以下语句快速发送：

```
self.robot.mode_ctrl(speed, isShoot)
```

帧头	目标地址	功能码	数据长度	数据内容	和校验	附加校验
HEAD	D_ADDR	ID	LEN	DATA	SC	AC
0xFF	机器人编号	0x06	2	格式如下	程序计算	程序计算

数据类型	UINT8
数据内容	1 为上位机接管 0 为下位机自行控制 下位机根据需求自行选择是否读取该值



## 4. 数据帧速查

### 4.1 目标地址速查

目标字段	目标名称	目标编号
broadcast	广播	0x00
mainfold	上位机	0x01
sentry_up	哨兵机器人上云台	0x02
sentry_down	哨兵机器人下云台	0x03
infantry	步兵机器人	0x04
engineer	工程机器人	0x05
hero	英雄机器人	0x06
air	空中机器人	0x07
radar	雷达站	0x08

### 4.2 功能码速查

功能码字段	功能码名词	功能码
pathway	哨兵轨道	0x01
gimbal	云台	0x02
barrel	枪管	0x04
referee_system	裁判系统	0x05
mode	模式控制	0x06
heartbeat	心跳数据	0xAA



木鸢机甲工作室

<https://www.sues.edu.cn>

上海市大学生创新创业教育实践基地

上海市松江区龙腾路333号上海工程技术大学实训楼1157室