

# ICRA 2018 DJI RoboMaster

## Artificial Intelligence Challenge Competition Referee System

### Conventions of Usage

V1.1 2018.04

# Disclaimer

Please read this disclaimer carefully before using this product. By using this product, you hereby signify that you have read and agree to this disclaimer. Install and use this product in strict accordance with the User Guide. SZ DJI TECHNOLOGY CO., LTD. (hereinafter referred to as "DJI") and its affiliated companies assume no liability for damage(s) or injuries incurred directly or indirectly from using, installing or modifying this product improperly, including but not limited to using non-designated accessories. DJI™ and RoboMaster™ are trademarks or registered trademarks of DJI and its affiliates. Names of products, brands, etc. appearing in this document are trademarks or registered trademarks of their respective owner companies. This product, the manual, and the software and driver that are compatible with the Referee System (RoboMaster Client, RoboMaster Assistant, RoboMaster Server and DJI WIN driver) are copyrighted by DJI with all rights reserved. No parts of this product or document shall be modified, reproduced or transmitted in any form without the prior written consent or authorization of DJI. DJI reserves the right to final interpretation of this document and all its related documents. All contents are subjected to the latest version of the manual.

## Product Usage Precautions

1. Please ensure that the robot ports of the referee monitoring device are installed correctly and firmly before use.
2. Please ensure that the wiring connection is correct before use.
3. Please ensure that the components are intact before use. Replace any worn or damaged components.

## Reading Tips

### Symbol Descriptions



Important Note



Operation/Use Tips



Glossary & Reference

### Related Documents

1. RoboMaster Referee System User Manual
2. Referee System Module Instructions

## Revision Log

Date	Version	Revision Records
2017.11.1	1.0	Convention of installation and function description related to ICRA rules.
2018.04.16	1.1	Modified "Referee System Interface Protocols" (p. 19)

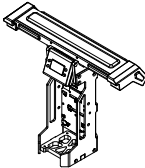

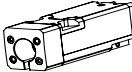

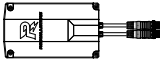
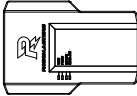
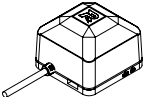
# Contents

<b>Disclaimer</b>	2
<b>Product Usage Precautions</b>	2
<b>Reading Tips</b>	2
Symbol Descriptions	2
Related Documents	2
<b>Revision Log</b>	2
<b>Referee System Usage Overview</b>	4
<b>Referee System Composition</b>	4
<b>Module Configurations</b>	5
<b>Installation Specifications</b>	5
Main Control Module	5
Armor Module	7
Speed Measurement Module	11
RFID Interaction Module	13
Camera Transmission Module	14
Localization Module	15
<b>Functions and Usage Rules</b>	16
Robot Specifications	16
Armor Module ID Setting Specifications	16
Module Status Monitoring Specifications	17
Launching Speed and Firing Rate Limits	18
Main Control Module Statuses	18
20-Second System Initialization Period	19
Competition Information Transmission Interface	19
<b>Appendix</b>	20
<b>Referee System Interface Protocols</b>	20
Communication Protocol Format	20
FrameHeader Format	20
<b>Command Code IDs</b>	20
Summary	20
Details	20
CRC Verification Example Code	25

## Referee System Usage Overview

To ensure fairness, the RoboMaster 2018 ICRA Artificial Intelligence Competition is evaluated electronically by the referee system. Each team must follow the usage conventions and correctly install the referee system. If a team violates the usage conventions, teams shall bear the consequences of failing to pass the pre-match inspection.

## Referee System Composition

Module Name	Appearance	Description
Main Control Module		The control unit of the referee system that monitors all the modules' statuses. Integrates functions such as power management, human-computer interaction, wireless networking, status display, robot health calculation, and more.
Armor Module		Perceives hits from projectiles. Transmits the hit data back to the Main Control Module and decreases Health Points accordingly.
17mm Speed Measurement Module		Detects projectile launching speed and frequency data in real time.
RFID Interaction Module		Performs information exchange with the battlefield module through NFC (near-field communication).
VTM Transmitter		Wirelessly transmits captured real-time HD video data through a camera to the receiving terminal.
VTM Receiver		Receives real-time HD video data and outputs it through the HDMI interface.
Localization Module		A real-time indoor positioning module. Outputs the two-dimensional coordinates of the venue information.

## Module Configurations

Module to Install (Required)	
Module Name	Quantity
Main Control Module	1
Small Armor Module	4
17mm Speed Measurement Module	1
RFID Interaction Module	1
Module to Install (Optional)	
Module Name	Quantity
VTM Transmitter	1
Localization Module	1

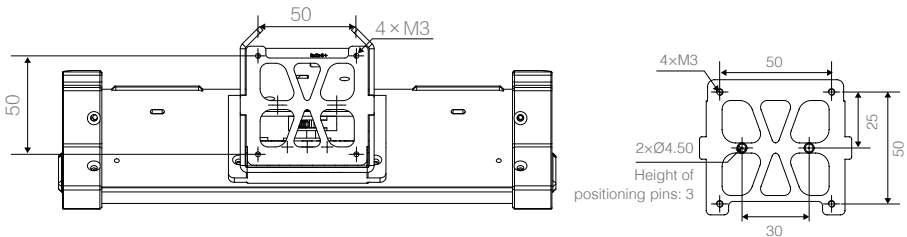
## Installation Specifications

This referee system is officially provided by the RMOC. This system monitors the damage status of the robot on which it is installed, including health points, projectile launching speed, and chassis power. The referee system server and the operating computers that receive the real-time data from the referee systems determine the outcome of the competition automatically and without bias. Teams must reserve mechanical and electrical interfaces for installing the referee system on their robots.

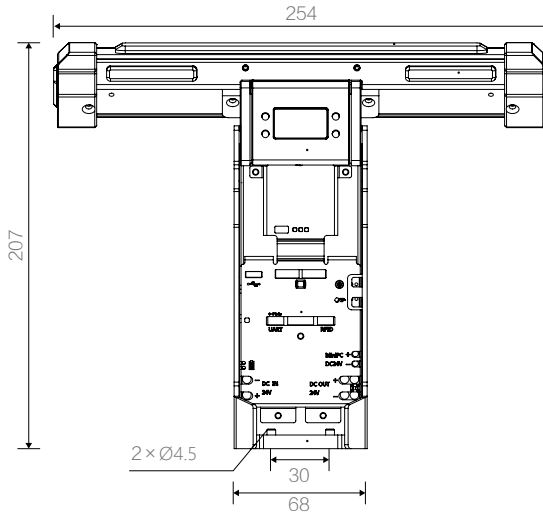
### Main Control Module

#### Installation Procedure

1. Reserve mounting holes on the robot chassis to match the dimensions of the Main Control Module. The Main Control Module holder can be flipped up or down. If needed, use the two 4.5-mm positioning pins that are 30 mm apart on one side of the module (shown in the figure below).

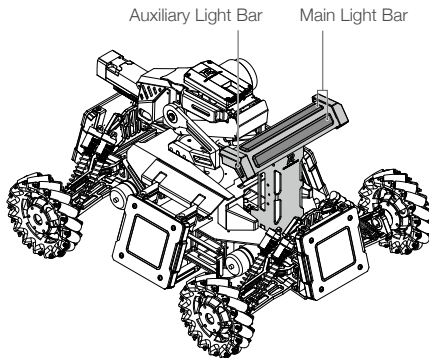


Uni: mm



Uni: mm

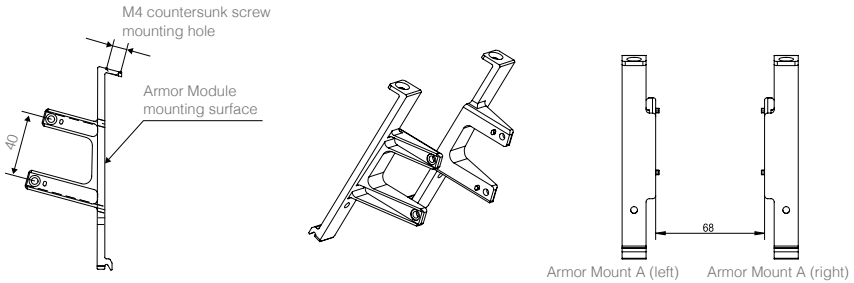
2. Use four M3 screws to mount the Main Control Module to the chassis.
3. Ensure that an imaginary line between the two auxiliary light bars is parallel to the ground, and that the status of the main light bar is completely visible when viewing the robot from at least one direction.



## Armor Module

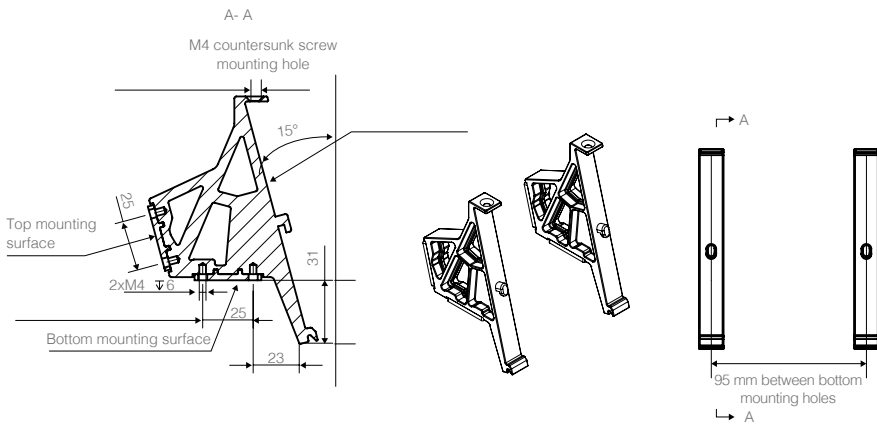
### Description

Use the armor mounts to install the Armor Module onto the robot. There are two types of armor mounts: Mount A and Mount B. For Armor Mount A, there is a left version and a right version (shown in the figure below):



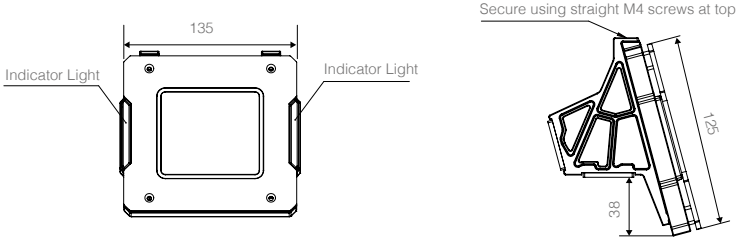
Uni: mm

Armor Mount B is shown in the figure below:



Uni: mm

The Small Armor Module is shown in the figure below:



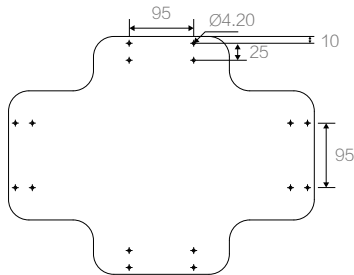
Use bottom mounting surface for installation

Uni: mm

The figure above uses Armor Mount B as an example. The labels in the figure apply to both Armor Mount A and Armor Mount B.

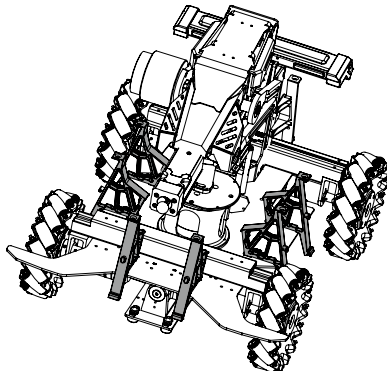
**Installation Procedure**

1. Using the dimensions shown in the figure below as a reference, reserve mounting holes on the chassis. Keep the sizes and the relative positions of the four mounting holes consistent.



Uni: mm

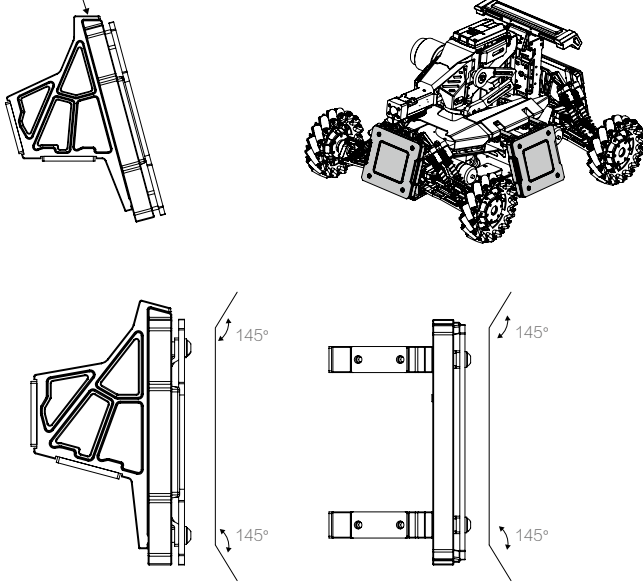
2. Use M4 screws to mount the Armor Mount B to the chassis.



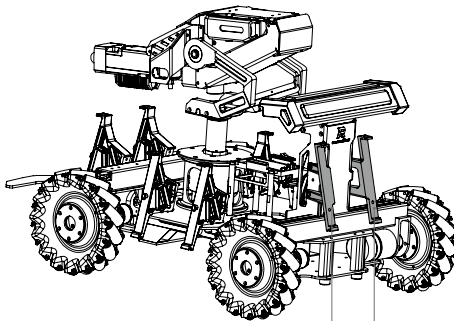


3. Install the Armor Module to the Armor Mount B and secure with M4 screws. Note: When correctly installed, the threaded hole on top of the armor mount is perpendicular to the horizontal ground, not perpendicular to the top surface of the armor mount. Ensure that the front 145° of the Armor Module's exposure surface is not blocked.

Use straight M4 screws to secure on top

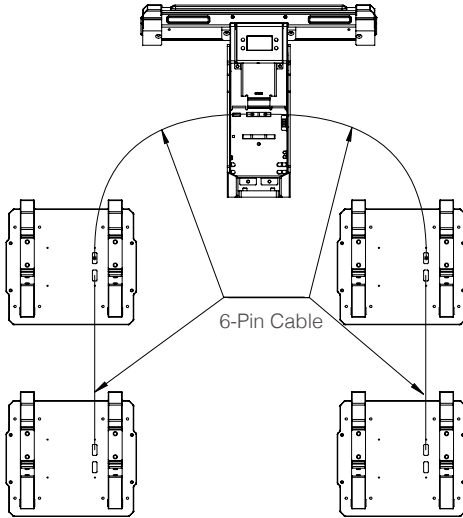


4. (Optional) Mount Armor Mount A at the rear of the robot using the M4 screw holes on the two sides of the Main Control Module. First, mount the Armor Module to Armor Mount A. Then, install Armor Mount A to the Main Control Module. Ensure that the mounting direction of the armor mount is correct to secure the installation.




Armor Mount A (left) Armor Mount A (right)

5. Use the 6-pin cable provided in the package to connect the Armor Modules in sequence to the Armor Module interface on the Main Control Panel. The two 6-pin interfaces on the Armor Module are equivalent to each other. It is recommended to split the total number of Armor Modules in sequence between the two equivalent interfaces to balance the current load.



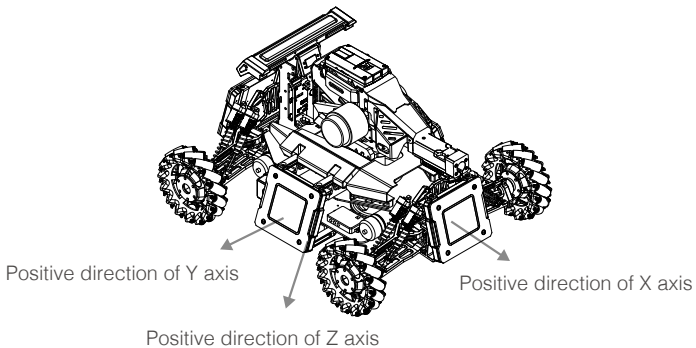
---

 • The lower edge of the Armor Module on the robot should be 50mm to 200mm from the ground.

---

### Installation Specifications and Requirements

In the following section, the robot body coordinate system is a standard XYZ Cartesian coordinate system, with the origin being the robot's center of mass, as shown in the following figure:



The kinematics equation of the robot should be based on the Cartesian coordinate system. If the team chooses to use a non-Cartesian coordinate system, the robot's coordinate system is defined using the following guidelines: Imagine that the launching mechanism of the largest caliber of the robot in its initial state launches a projectile. Define the positive X direction as the projected vector on the XY plane of this projectile's projective direction. Establish the positive Y direction by using the positive X direction and the positive Z direction (pointing to the earth's core) according to the right-hand rule, with the robot's center of mass as the origin.

**Installation on the Sides**

When an Armor Module is mounted on the side, its exposure surface must be firmly connected to the armor mount. Keep the bottom surface of the armor mount parallel to the XY plane, so that the acute angle between the normal vector of the Armor Module's exposure surface and the negative Z-axis is 75°.

Keep the two sides without the indicator light parallel to the XY plane. Define the directional vector of any Armor Module as the projected vector of the exposure surface's normal vector (forming an acute angle with the negative Z axis) on the XY plane. The unit vectors of the direction vectors of the 4 Armor Modules must be equal to the positive X-axis, negative X-axis, positive Y-axis, and negative Y-axis of the robot's coordinate system respectively. The angular error between the direction vector and the corresponding coordinate vector cannot exceed 5°.

The kinematics equations of the robot should also be based on the Cartesian coordinate system in this case. Armor Modules must use the same reference frame as the robot's own structural or kinematic characteristics. The imaginary connection line between the geometric centers of the Armor Modules mounted on the X-axis should be perpendicular to the counterparts of those mounted on the Y-axis. The two imaginary lines should also pass through the robot's geometric center. Armor Modules mounted on the X- or Y- axis can deviate by ±50 mm from the geometric center.

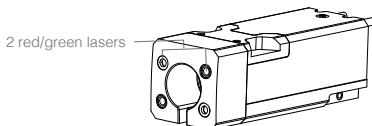
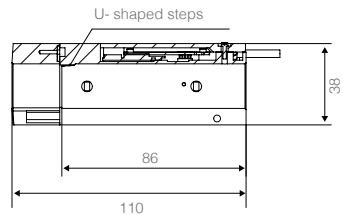
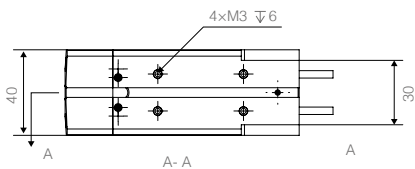


- Self-designed protective armor cannot have any contact with the official Armor Module.
- Do not change or decorate the official Armor Module.

**Speed Measurement Module**

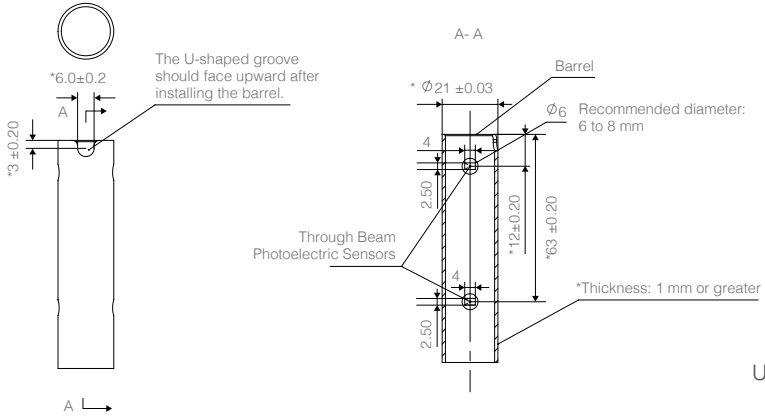
**Installation**

Speed Measurement Module (17-mm projectile):



Uni: mm

17-mm barrel size restrictions:



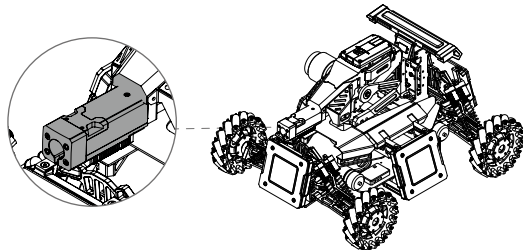
Uni: mm

**Barrel Requirements**

1. The barrel length must be greater than 90 mm.
2. Pay special attention to maintaining dimensions marked with the \* sign.
3. Ensure that the Through Beam Photoelectric Sensors are not blocked.
4. Transparent materials are not allowed.

**Installation Procedure**

1. Set the Speed Measurement module on the barrel with its cable pointing at the Main Control Module and the U-shaped step aligning with the U-shaped groove on the barrel.
2. Use M3 screws to tighten the barrel through the screw holes in the rear of the speed measurement module.
3. Connect the aviation plugs on the Speed Measurement Module to the plug on the Main Control Panel's Speed Measurement interface. The completed installation is shown in the figure below.

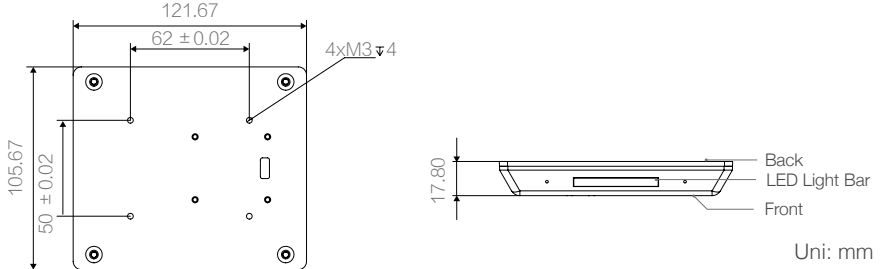


- Ensure that the mounting holes of the Through Beam Photoelectric Sensors are not blocked. Otherwise, the Speed Measurement Module will be unable to initialize.
- Ensure that the Speed Measurement Module is firmly secured to avoid relative displacement between the barrel and the module.

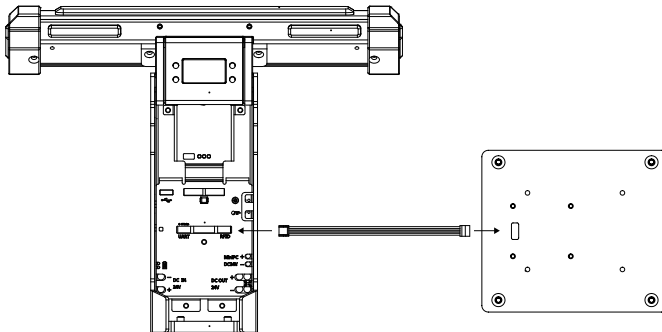
## RFID Interaction Module

### Installation Procedure

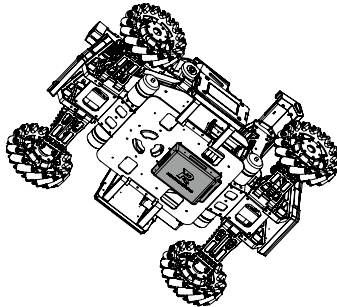
- Using the dimensions of the RFID Interaction Module as a reference, reserve mounting holes on the chassis.



- Use the 4-pin cable provided in the package to connect the RFID Interaction Module to the RFID interface on the Main Control Panel.



- Use M3 screws to mount the RFID Interaction Module to the chassis. Do not press or bend the 4-pin cable during installation, and choose an installation location away from the ground.

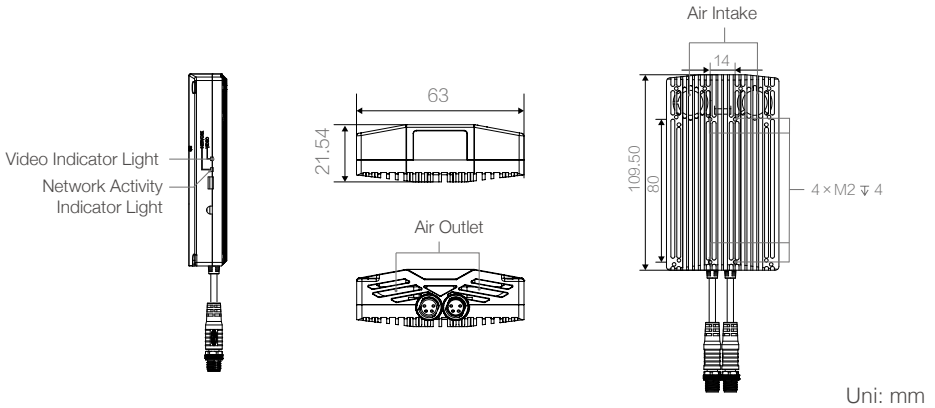


- ⚠ • Ensure that the logo side of RFID Interaction Module is not blocked by any metal objects. Test the effective detection distance of the module. If it decreases, check whether the module is installed properly.
- The complimentary RFID Interaction testing card can increase the robot's damage strength by 50%.

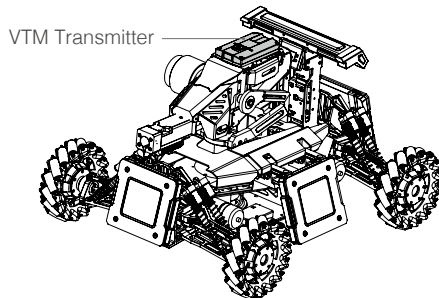
## Camera Transmission Module

### Installation of the VTM Transmitter

- Using the dimensions of the VTM Transmitter and its interfaces as a reference, reserve mounting holes in the desired location.



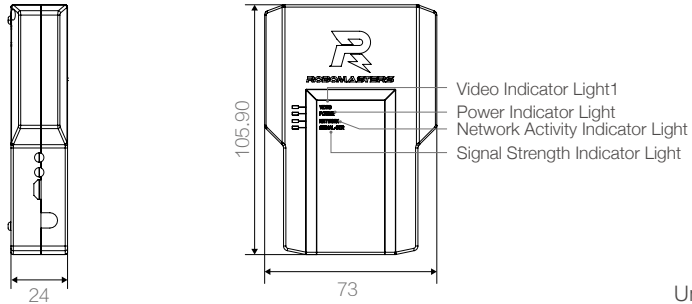
Use four M2 screws to mount the VTM Transmitter to a proper location. To ensure that the VTM Transmitter works properly, keep its air intake and outlet free of blockage. Also ensure that no metal components are blocking the top of the module, where its antenna is located.



- Connect the aviation plug of the VTM Transmitter to the plug on the main control panel's camera transmission interface.

### Installation of the VTM Receiver

Use the complimentary mounting clip to mount the VTM Receiver on the monitor or any other support. Ensure that the mounting location is at least 1 m above the ground and is not blocked by any metal objects. Check the quality of the receiving image to find the most suitable mounting location. The VTM Receiver is shown in the figure below:

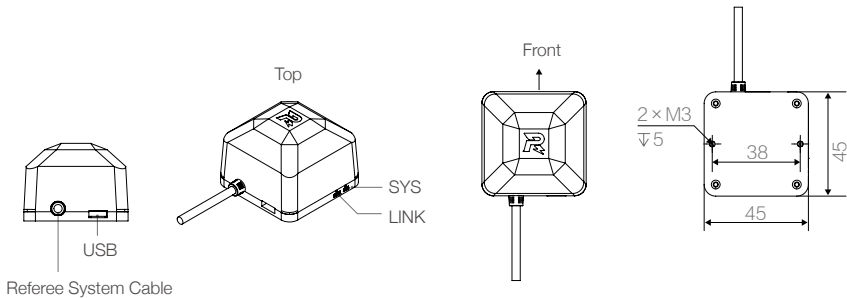


Uni: mm

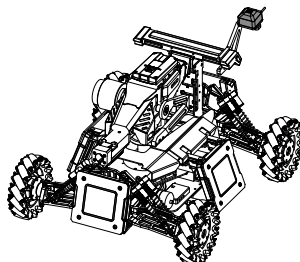
## Localization Module

### Installation Procedure

- Using the dimensions of the Localization Module as a reference, reserve mounting holes in the specified locations.



- Use two M3 screws to mount the Localization Module to the specified location. The front of the Localization Module must be in line with the front of the robot, horizontally installed with the top facing up.



- Use the aviation plug extension cords in the package to connect the Localization Module to the aviation plug of the Camera Transmission Module.



- The aviation plugs on the VTM Transmitter, the VTM Receiver, Localization Module and the Main Control Panel are all equivalent interfaces.
- The localization base stations are fixed at the top of the fence around the site. While the robot is moving, ensure that the straight-line connection between the Localization Module and each base station is not blocked by the robot itself. It is recommended to install the Localization Module at the highest point on the robot.
- It is recommended to maintain greater than 20 cm between the installation location of the Localization Module and the motor, the camera transmitter module, and any components that have or can generate a strong magnetic field. A distance of less than 10 cm may cause motor or module failure.

## Functions and Usage Rules

### Robot Specifications

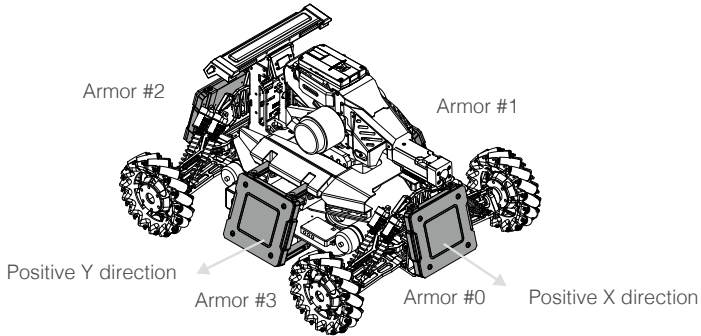
Specification	Value
Robot initial value	2000 HP(Health Points)
Max. launching speed	20 m/s
Max. launching rate	10 Hz
Damage caused by single projectile	50 HP (Health Points)

### Armor Module ID Setting Specifications

The modules of the referee system communicate using a CAN network, so they need unique IDs to ensure smooth communication. During production, each Armor Module is assigned a default ID. Therefore, you need to set sequenced IDs the first time you install multiple Armor Modules and each time you replace them. The ID setting procedure is as follows:

1. On the referee system interactive main page, hold the "Confirm" button to enter the "Function" page of the referee system.
2. On the "Function" page, press the up and down buttons to find and select "System Setup" to enter the "System Setup" page.
3. On the "System Setup" page, press the up and down buttons to find and select "Armor ID Setup". Then, press the "Confirm" button to enter the armor ID setting mode page. Select "Armor ID Reset" to enter the armor ID reset status page. Then armor light will flash at a certain frequency (The red light flashes if the robot ID is red, and the blue light flashes if the ID is blue).
4. Using a 42-mm projectile (or a similar-sized object), strike each Armor Module in the sequence of Armor #0, #1, #2, #3. If the armor light stops flashing, the armor ID has been set successfully. The ID number of the Armor Module that was hit first is 0, and the ID number is sequentially incremented according to the hitting order. After completing the above operations, check whether the armor ID has been set successfully by querying the version number of the Armor Modules. If the read number of effective Armor Modules is the same as the number of Armor Modules installed, the process is complete. Armor Module ID numbers have a strict sequential requirement, as shown in the figure below:





- Please set the Armor Module IDs in advance of the competition. If an ID is set incorrectly, the referee system will not be able to detect it during initialization. The referee system will consider it an offline Armor Module and automatically deduct HP from the robot during the competition.

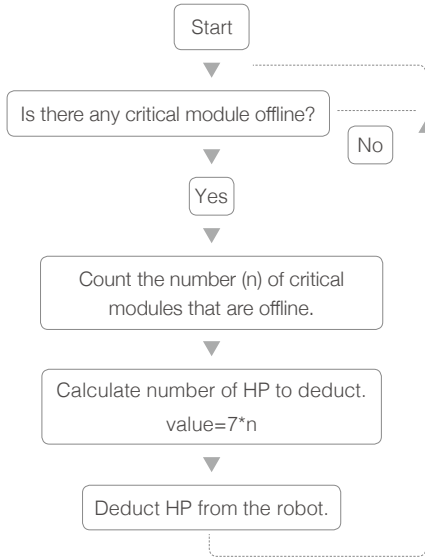
## Module Status Monitoring Specifications

To ensure fair competition, it is important that all of the referee system modules are working properly. Therefore, the referee system initializes each module after powering up. Depending on the amount of impact each module has on the competition, it is classified into one of the following categories: critical modules or general modules.

Critical modules: Armor Modules, Speed Measurement Module, and Main Control Module

General modules: RFID Interaction Modules, VTM Transmitter, VTM Receiver, and Localization Module

Critical modules affect the fairness of the competition. For example, an Armor Module will not detect enemy attacks if it is malfunctioning. In the official competition, the referee system checks each module's connection at a frequency of 2 Hz. If a critical module is detected as offline, the referee system will automatically deduct HP from the robot. The HP deduction mechanism is shown in the following flowchart:



General module offline is detrimental to the friendly team’s performance overall. Therefore, the referee system will not deduct robot’s health point but flash the yellow auxiliary light to remind the team of this situation.

**Launching Speed and Firing Rate Limits**

Categories	Max. Value	Penalty
Launching speed	20 m/s	a. 10% of HP deducted if exceeded by 10% b. 20% of HP deducted if exceeded by more than 10% but less than or equal to 20%. c. 40% of HP deducted if exceeded by more than 20%.
Launching rate	10 Hz	10% of HP deducted for each projectile fired exceeding the launching rate

**Main Control Module Statuses**

Normal Statuses		
Main Control Module light bar status	Auxiliary light bar status	Description
Some red/blue LEDs steadily lit; some LEDs off	Red or blue LEDs steadily lit	The light color indicates the robot’s ID. A steadily lit light bar indicates the percentage of HP remaining. If all LEDs are steadily lit, the robot is at full health.
Red or blue LEDs steadily lit	Flashing white light	Robot has gained a 50% attack boost.

**Warning or Error Statuses**

Main Control Module light bar status	Auxiliary light bar status	Description
Yellow light steadily lit	Yellow light steadily lit	Not competing; critical module offline
Normal status	Flashing yellow light	Not competing; general module offline

**20-Second System Initialization Period**

After the two-minute preparation phase preceding the competition, there is a 20-second referee system initialization period. After the system initialization, the competition officially starts. During the initialization, the competition server automatically detects the client connection status, robot wireless connection status, robot module status, and the status of the venue props. If the status cannot support the game's proper initialization (e.g., a client, robot, or prop is offline), the system initialization countdown is suspended. After all faults are cleared, the referee will manually restart the system initialization and the countdown continues. Upon entering the 20-second initialization period, the competition server restores all robots' HP completely. During this period, the operators must stay in the operator room and ensure the mice and the keyboards are functioning properly.

**Competition Information Transmission Interface**

To help contestants better design the automatic control system and obtain real-time statuses during the competition, the referee system includes an UART input and output interface. This system outputs data from the robots and the battlefield during the competition. It can also transmit data customized by the users and display it on the RoboMaster Client's UI. This output contains the remaining time in the competition, the robots' remaining HP and location data, and the IDs of the attacked armor. The open uplink data interface reserves three floating-point data types. The user can make the uplink data appear on the corresponding RoboMaster Client's UI as described to the "Referee System Interface Protocol Description".

## Appendix

### Referee System Interface Protocols

#### Communication Protocol Format

FrameHeader(5-Byte)	CmdID(2-Byte)	Data(n-Byte)	FrameTail(2-Byte, CRC16)
---------------------	---------------	--------------	--------------------------

#### FrameHeader Format

Region	Offset position	Size (byte)	Description
SOF	0	1	The FrameHeader is fixed at 0xA5
DataLength	1	2	Data segment length
Seq	3	1	Sequence number
CRC8	4	1	Frame head CRC8 check

### Command Code IDs

#### Summary

Command Code	Data segment length (Byte)	Function
0x0001	8	Robot status, transmitted periodically at a frequency of 10 Hz
0x0002	1	Damage data, transmitted when the robot is hit
0x0003	6	Real-time launch data, transmitted when a projectile is launched
0x0005	2	Real-time RFID interaction data; transmitted periodically at a frequency of 10 Hz when the RFID card is detected
0x0006	1	Match results data is sent at the end of the match
0x0007	2	Buff status, sent when the status of any Buff changes
0x0008	16	Robot position and orientation, transmitted periodically at a frequency of 50 Hz
0x0100	13	Team custom data displayed in the operating interface is limited to a frequency of 10 Hz

#### Details

##### 0x0001 (Robot's status in the competition)

Byte offset	Size	Description
0	2	Remaining time in the current round (unit: seconds)

2	1	Current competition stage 0: pre-competition stage 1: preparation stage 2: initialization stage 3: 5-second countdown 4: in combat 5: calculating competition results
3	1	Reserved
4	2	Robot's current HP
6	2	Robot's maximum HP

Definition:

```
typedef __packed struct
{
    uint16_t stageRemianTime;
    uint8_t gameProgress;
    uint8_t reserved;
    uint16_t remainHP;
    uint16_t maxHP;
}extGameRobotState_t;
```

**0x0002 (Damage data)**

Byte offset	Size	Description
0	1	Bits 0-3: indicate the armor ID if the type of change is armor damage 0x0: armor #0 (front) 0x1: armor #1 (left) 0x2: armor #2 (rear) 0x3: armor #3 (right) Other bits reserved  Bits 4-7: changes in HP 0x0: armor damage (attack received) 0x1: module offline

Definition:

```
typedef __packed struct
{
    uint8_t armorType : 4;
    uint8_t hurtType : 4;
}extRobotHurt_t;
```

**0x0003 (Real-time launching data)**

Byte offset	Size	Description
0	1	Reserved
1	1	Projectile launching rate
2	4	Projectile launching speed

Definition:

```
typedef __packed struct  
{  
    uint8_t reserved;  
    uint8_t bulletFreq;  
    float bulletSpeed;  
}extShootData_t;
```

**0x0005 (Real-time RFID interaction data)**

Byte offset	Size	Description
0	1	Card type 11: large Power Rune attack points card Other bits reserved
1	1	Card index number; used to distinguish different sections

Definition:

```
typedef __packed struct  
{  
    uint8_t cardType;  
    uint8_t cardIdx;  
}extRfidDetect_t;
```

**0x0006 (Competition result data)**

Byte offset	Size	Description
0	1	Competition result 0: Draw 1: Red team wins 2: Blue team wins

Definition:

```
typedef __packed struct  
{  
    uint8_t winner;  
}extGameResult_t;
```

**0x0007 (Buff status)**

Byte offset	Size	Description
0	2	Buff type (1 = active) bit 13: ICRA team member has received a small Power Rune bit 14: ICRA enemy has received a small Power Rune Other bits reserved

Definition:

```
typedef __packed struct
{
    uint16_t buffMusk;
} extGetBuff_t;
```

**0x0008 (Robot position and orientation)**

Byte offset	Size	Description
0	4	X value of position
4	4	Y value of position
8	4	Z value of position
12	4	Reserved

Definition:

```
typedef __packed struct
{
    float x;
    float y;
    float z;
    float reserved;
} extGameRobotPos_t;
```

**0x0100 (Customized data)**

Byte offset	Size	Description
0	4	Customized data 1
4	4	Customized data 2
8	4	Customized data 3
8	4	Customized data 4

Definition:

```
typedef __packed struct
```

```
{
```

```
    float data1;
```

```
    float data2;
```

```
float data3;
```

```
uint8_t mask;
```

```
}extShowData_t;
```



## CRC Verification Example Code

```

//crc8 generator polynomial:G(x)=x8+x5+x4+1
const unsigned char CRC8_INIT = 0xff;
const unsigned char CRC8_TAB[256] =
{
0x00, 0x5e, 0xbc, 0xe2, 0x61, 0x3f, 0xdd, 0x83, 0xc2, 0x9c, 0x7e, 0x20, 0xa3, 0xfd, 0x1f, 0x41,
0x9d, 0xc3, 0x21, 0x7f, 0xfc, 0xa2, 0x40, 0x1e, 0x5f, 0x01, 0xe3, 0xbd, 0x3e, 0x60, 0x82, 0xdc, 0x23,
0x7d, 0x9f, 0xc1, 0x42, 0x1c, 0xfe, 0xa0, 0xe1, 0xbf, 0x5d, 0x03, 0x80, 0xde, 0x3c, 0x62, 0xbe, 0xe0,
0x02, 0x5c, 0xdf, 0x81, 0x63, 0x3d, 0x7c, 0x22, 0xc0, 0x9e, 0x1d, 0x43, 0xa1, 0xff, 0x46, 0x18, 0xfa,
0xa4, 0x27, 0x79, 0x9b, 0xc5, 0x84, 0xda, 0x38, 0x66, 0xe5, 0xbb, 0x59, 0x07, 0xdb, 0x85, 0x67, 0x39,
0xba, 0xe4, 0x06, 0x58, 0x19, 0x47, 0xa5, 0xfb, 0x78, 0x26, 0xc4, 0x9a, 0x65, 0x3b, 0xd9, 0x87, 0x04,
0x5a, 0xb8, 0xe6, 0xa7, 0xf9, 0x1b, 0x45, 0xc6, 0x98, 0x7a, 0x24, 0xf8, 0xa6, 0x44, 0x1a, 0x99, 0xc7,
0x25, 0x7b, 0x3a, 0x64, 0x86, 0xd8, 0x5b, 0x05, 0xe7, 0xb9,
0x8c, 0xd2, 0x30, 0x6e, 0xed, 0xb3, 0x51, 0x0f, 0x4e, 0x10, 0xf2, 0xac, 0x2f, 0x71, 0x93, 0xcd, 0x11,
0x4f, 0xad, 0xf3, 0x70, 0x2e, 0xcc, 0x92, 0xd3, 0x8d, 0x6f, 0x31, 0xb2, 0xec, 0x0e, 0x50, 0xaf, 0xf1,
0x13, 0x4d, 0xce, 0x90, 0x72, 0x2c, 0x6d, 0x33, 0xd1, 0x8f, 0x0c, 0x52, 0xb0, 0xee, 0x32, 0x6c, 0x8e,
0xd0, 0x53, 0x0d, 0xef, 0xb1, 0xf0, 0xae, 0x4c, 0x12, 0x91, 0xcf, 0x2d, 0x73, 0xca, 0x94, 0x76, 0x28,
0xab, 0xf5, 0x17, 0x49, 0x08, 0x56, 0xb4, 0xea, 0x69, 0x37, 0xd5, 0x8b, 0x57, 0x09, 0xeb, 0xb5, 0x36,
0x68, 0x8a, 0xd4, 0x95, 0xcb, 0x29, 0x77, 0xf4, 0xaa, 0x48, 0x16, 0xe9, 0xb7, 0x55, 0x0b, 0x88, 0xd6,
0x34, 0x6a, 0x2b, 0x75, 0x97, 0xc9, 0x4a, 0x14, 0xf6, 0xa8,
0x74, 0x2a, 0xc8, 0x96, 0x15, 0x4b, 0xa9, 0xf7, 0xb6, 0xe8, 0x0a, 0x54, 0xd7, 0x89, 0x6b, 0x35,
};
unsigned char Get_CRC8_Check_Sum(unsigned char *pchMessage,unsigned int dwLength,unsigned char ucCRC8)
{
unsigned char ucIndex;
while (dwLength-->0)
{
ucIndex = ucCRC8^(*pchMessage++);
ucCRC8 = CRC8_TAB[ucIndex];
}
return(ucCRC8);
}
/*
** Descriptions: CRC8 Verify function
** Input: Data to Verify,Stream length = Data + checksum
** Output: True or False (CRC Verify Result)
*/

```

```
unsigned int Verify_CRC8_Check_Sum(unsigned char *pchMessage, unsigned int dwLength)
{
    unsigned char ucExpected = 0;
    if ((pchMessage == 0) || (dwLength <= 2)) return 0;
    ucExpected = Get_CRC8_Check_Sum (pchMessage, dwLength-1, CRC8_INIT);
    return ( ucExpected == pchMessage[dwLength-1] );
}
/*
** Descriptions: append CRC8 to the end of data
** Input: Data to CRC and append.Stream length = Data + checksum
** Output: True or False (CRC Verify Result)
*/
void Append_CRC8_Check_Sum(unsigned char *pchMessage, unsigned int dwLength)
{
    unsigned char ucCRC = 0;
    if ((pchMessage == 0) || (dwLength <= 2)) return;
    ucCRC = Get_CRC8_Check_Sum ( (unsigned char *)pchMessage, dwLength-1, CRC8_INIT);
    pchMessage[dwLength-1] = ucCRC;
}

uint16_t CRC_INIT = 0xffff;
const uint16_t wCRC_Table[256] =
{
    0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf,
    0x8c48, 0x9dc1, 0xaf5a, 0xbbed, 0xca6c, 0xdbed, 0xe97e, 0xf8f7,
    0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e,
    0x9cc9, 0x8d40, 0xbfdb, 0xae52, 0xdaed, 0xcb64, 0xf9ff, 0xe876,
    0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd,
    0xad4a, 0xbcc3, 0x8e58, 0x9fd1, 0xeb6e, 0xfae7, 0xc87c, 0xd9f5,
    0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c,
    0xbdcb, 0xac42, 0x9ed9, 0x8f50, 0xfbef, 0xea66, 0xd8fd, 0xc974,
    0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb,
    0xce4c, 0xdfc5, 0xed5e, 0xfcd7, 0x8868, 0x99e1, 0xab7a, 0xbaf3,
    0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a,
    0xdecd, 0xcf44, 0xfddf, 0xec56, 0x98e9, 0x8960, 0xbbfb, 0xaa72,
    0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9,
    0xef4e, 0xfec7, 0xcc5c, 0xdd5, 0xa96a, 0xb8e3, 0x8a78, 0x9bf1,
    0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738,
    0xffcf, 0xee46, 0xdcdd, 0xcd54, 0xb9eb, 0xa862, 0x9af9, 0x8b70,
```

```

0x8408, 0x9581, 0xa71a, 0xb693, 0xc22c, 0xd3a5, 0xe13e, 0xf0b7,
0x0840, 0x19c9, 0x2b52, 0x3adb, 0x4e64, 0x5fed, 0x6d76, 0x7cff,
0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036,
0x18c1, 0x0948, 0x3bd3, 0x2a5a, 0x5ee5, 0x4f6c, 0x7df7, 0x6c7e,
0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5,
0x2942, 0x38cb, 0x0a50, 0x1bd9, 0x6f66, 0x7eef, 0x4c74, 0x5dfd,
0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134,
0x39c3, 0x284a, 0x1ad1, 0x0b58, 0x7fe7, 0x6e6e, 0x5cf5, 0x4d7c,
0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3,
0x4a44, 0x5bcd, 0x6956, 0x78df, 0x0c60, 0x1de9, 0x2f72, 0x3efb,
0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232,
0x5ac5, 0x4b4c, 0x79d7, 0x685e, 0x1ce1, 0x0d68, 0x3ff3, 0x2e7a,
0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1,
0x6b46, 0x7acf, 0x4854, 0x59dd, 0x2d62, 0x3ceb, 0x0e70, 0x1ff9,
0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330,
0x7bc7, 0x6a4e, 0x58d5, 0x495c, 0x3de3, 0x2c6a, 0x1ef1, 0x0f78
};
/*
** Descriptions: CRC16 checksum function
** Input: Data to check,Stream length, initialized checksum
** Output: CRC checksum
*/
uint16_t Get_CRC16_Check_Sum(uint8_t *pchMessage,uint32_t dwLength,uint16_t wCRC)
{
    UInt8_t chData;
    if (pchMessage == NULL)
    {
        return 0xFFFF;
    }
    while(dwLength--)
    {
        chData = *pchMessage++;
        (wCRC) = ((uint16_t)(wCRC) >> 8) ^ wCRC_Table[((uint16_t)(wCRC) ^ (uint16_t)(chData)) & 0x00ff];
    }
    return wCRC;
}

/*
** Descriptions: CRC16 Verify function

```

\*\* Input: Data to Verify,Stream length = Data + checksum

\*\* Output: True or False (CRC Verify Result)

\*/

```
uint32_t Verify_CRC16_Check_Sum(uint8_t *pchMessage, uint32_t dwLength)
```

```
{  
    uint16_t wExpected = 0;  
    if ((pchMessage == NULL) || (dwLength <= 2))  
    {  
        return __FALSE;  
    }  
    wExpected = Get_CRC16_Check_Sum ( pchMessage, dwLength - 2, CRC_INIT);  
    return ((wExpected & 0xff) == pchMessage[dwLength - 2] && ((wExpected >> 8) & 0xff) ==  
    pchMessage[dwLength - 1]);  
}
```

/\*

\*\* Descriptions: append CRC16 to the end of data

\*\* Input: Data to CRC and append,Stream length = Data + checksum

\*\* Output: True or False (CRC Verify Result)

\*/

```
void Append_CRC16_Check_Sum(uint8_t * pchMessage,uint32_t dwLength)
```

```
{  
    uint16_t wCRC = 0;  
    if ((pchMessage == NULL) || (dwLength <= 2))  
    {  
        return;  
    }  
    wCRC = Get_CRC16_Check_Sum ( (U8 *)pchMessage, dwLength-2, CRC_INIT );  
    pchMessage[dwLength-2] = (U8)(wCRC & 0x00ff);  
    pchMessage[dwLength-1] = (U8)((wCRC >> 8)& 0x00ff);  
}
```



- Uart communication configuration, baud rate 115200, data bit 8, stop bit 1, parity bit none, flow control none.
-



## Contact Us

Email: [robomaster@dji.com](mailto:robomaster@dji.com)

ICRA 2018 official website: [icra2018.org/](http://icra2018.org/)

RoboMaster official website: [www.robomaster.com/en-US](http://www.robomaster.com/en-US)

Phone number: 0755-36383255 (Monday to Friday, 10 AM to 7 PM UTC-8)

WeChat



RoboMasterNews

Weibo



RoboMaster

**R** and **ROBOMASTER** are trademarks of DJI.