**Định dạng khung truyền dữ liệu**

**Rockstar Machine**

ALTA Media June 20,2025

Phần cứng gồm 1 board độc lập

Địa chỉ: 0x03

Kết nối : USB/Serial

Baudrate: 9600;

Coding system: 8 bit binary, hexadecimal 0-9, A-F

Bits/byte: 1 start bit, 8 data bits, no parity, 2 stop bit

LSB truyền trước. (Least Significant Bit)

Contents

[FUNCTION 0x07: Số lượng người quét mã QR 4](#_Toc199772057)

[FUNCTION 0x06: Trạng thái các động cơ trong máy 5](#_Toc199772058)

[FUNCTION 0x08: Số lượng lon rơi 6](#_Toc199772059)

[FUNCTION 0x18: Đọc trạng thái nhả quà 7](#_Toc199772060)

[FUNCTION 0x19: Nhận lệnh nhả quà 8](#_Toc199772061)

[FUNCTION 0x10: Reset Order command Hi / mã nhảy về mặc định 9](#_Toc199772062)

[FUNCTION 0x09: Xóa lỗi máy 10](#_Toc199772063)

[FUNCTION 0x80: Điều khiển động cơ trong máy RVM 11](#_Toc199772064)

[FUNCTION 0xFA: Trạng thái cơ khí trong máy ROCSTAR 12](#_Toc199772065)

[FUNCTION 0xFB: Kiểm tra model, firmware phần cứng 13](#_Toc199772066)

[FUNCTION 0xFE: Báo lỗi truyền dữ liệu 15](#_Toc199772067)

[Định dạng khung truyền dữ liệu giữa các thiết bị. 16](#_Toc199772068)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

# FUNCTION 0x07: Số lượng người quét mã QR

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03 mặc định** |
| FUNTION | **07** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh gửi Hi** |
| Order command Lo | **XX** | **Số thứ tự lệnh gửi Lo (lệnh mới thì tăng giá trị HiLo này lên, để tránh trùng lệnh)** |
| Byte Count | **02** |  |
| Data Hi | **XX** | **Giá trị là**  Data Hi **=0** |
| Data Lo | **XX** | **Số lượng người muốn nhả là:**  **=0x01: 1 người**  **=0x02: 2 người**  **=0x03: 3 người** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ phần cứng:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định** |
| FUNTION | **07** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** |  |
| Byte Count | **02** | **Số Byte Data đã nhận** |
| Status | **XX** | **=00: xác nhận đã nhận dữ liệu ok**  **~~=01: Có lỗi nhận dữ liệu sai CRC~~**  **~~=02: Lỗi không hỗ trợ FUNTION này.~~**  **~~=03: Lỗi mã lệnh nhỏ hơn hoặc bằng lệnh cũ.~~** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

# FUNCTION 0x06: Trạng thái các động cơ trong máy

Thiết bị -> PC

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03 mặc định** |
| FUNTION | **06** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh gửi Hi** |
| Order command Lo | **XX** | **Số thứ tự lệnh gửi Lo (lệnh mới thì tăng giá trị HiLo này lên, để tránh trùng lệnh)** |
| Byte Count | **02** |  |
| Data 0 | **XX** | **Động cơ:**  **=0x01: thang máy**  **=0x02: motor nha qua** |
| Data 1 | **XX** | **Trạng thái động cơ:**  **=0xFF: có lỗi**  **=0x00: không có lỗi** |
| Data 2 | **XX** | **Bổ sung sau nếu có** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ PC:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định** |
| FUNTION | **06** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** |  |
| Byte Count | **02** | **Số Byte Data đã nhận** |
| Status | **XX** | **=00: xác nhận đã nhận dữ liệu ok**  **~~=01: Có lỗi nhận dữ liệu sai CRC~~**  **~~=02: Lỗi không hỗ trợ FUNTION này.~~**  **~~=03: Lỗi mã lệnh nhỏ hơn hoặc bằng lệnh cũ.~~** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

# FUNCTION 0x08: Số lượng lon rơi

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03 mặc định** |
| FUNTION | **08** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh gửi Hi** |
| Order command Lo | **XX** | **Số thứ tự lệnh gửi Lo (lệnh mới thì tăng giá trị HiLo này lên, để tránh trùng lệnh)** |
| Byte Count | **02** |  |
| Data Hi | **XX** | **Giá trị là**  Data Hi **=0** |
| Data Lo | **XX** | **Số lượng lon đã rớt:**  **=0x01: 1 người**  **=0x02: 2 người** |
| Data 2 | **XX** | **Bổ sung sau nếu có** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ phần cứng:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định** |
| FUNTION | **08** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** |  |
| Byte Count | **02** | **Số Byte Data đã nhận** |
| Status | **XX** | **=00: xác nhận đã nhận dữ liệu ok**  **~~=01: Có lỗi nhận dữ liệu sai CRC~~**  **~~=02: Lỗi không hỗ trợ FUNTION này.~~**  **~~=03: Lỗi mã lệnh nhỏ hơn hoặc bằng lệnh cũ.~~** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

# FUNCTION 0x18: Đọc trạng thái nhả quà

Hardware gửi trạng thái lên PC

FUNCTION: 0x18

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** |  |
| FUNTION | **18** | **default** |
| Order command Hi | **XX** | **default** |
| Order command Lo | **XX** | **default** |
| Byte Count | **03** | **Số Byte data đọc được** |
| Data 0 | **XX** | **Trạng thái hoạt động của motor nghiền chái:**  +FF: Có Quà Rơi  +00: Không có quà rơi, hoặc lỗi động cơ nhả quà |
| Data 1 | **XX** | Vị trí khay sản phẩm |
| Data 2 | **XX** | Vị trí ô sản phẩm |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

**Phản hồi từ PC**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03** |
| FUNTION | **18** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh gửi Hi** |
| Order command Lo | **XX** | **Số thứ tự lệnh gửi Lo (lệnh mới thì tăng giá trị HiLo này lên, để tránh trùng lệnh)** |
| Byte Count | **00** | **Không có data gửi đi** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

# FUNCTION 0x19: Nhận lệnh nhả quà

Hardware gửi trạng thái lên PC

FUNCTION: 0x19

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** |  |
| FUNTION | **19** | **default** |
| Order command Hi | **XX** | **default** |
| Order command Lo | **XX** | **default** |
| Byte Count | **02** | **Số Byte data đọc được** |
| Data 0 | **XX** | **Vị trí khay cần nhả**  **=01:khay số một**  **=02:khay số hai**  **=03:khay số ba** |
| Data 1 | **XX** | **Vị trí ô cần nhả: ô cần rơi quà (int).Thí dụ:1,2,3.** |
|  |  | **Thêm nếu phát sinh** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

**Phản hồi từ PC**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03** |
| FUNTION | **19** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh gửi Hi** |
| Order command Lo | **XX** | **Số thứ tự lệnh gửi Lo (lệnh mới thì tăng giá trị HiLo này lên, để tránh trùng lệnh)** |
| Byte Count | **00** | **Không có data gửi đi** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

# FUNCTION 0x10: Reset Order command Hi / mã nhảy về mặc định

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **10** | **default** |
| Order command Hi | **00** | **default** |
| Order command Lo | **00** |  |
| Byte Count | **02** | **default** |
| Data Hi | **00** | **Giá trị là 0** |
| Data Lo | **00** | **Giá trị là 0** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ phần cứng:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **10** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** | **Default** |
| Byte Count | **02** | **Số Byte Data đã nhận** |
| Status | **XX** | **=00: xác nhận đã nhận dữ liệu ok**  **~~=01: Có lỗi nhận dữ liệu sai CRC~~**  **~~=02: Lỗi không hỗ trợ FUNTION này.~~** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

# FUNCTION 0x09: Xóa lỗi máy

**PC🡪THIẾT BỊ**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **09** | **default** |
| Order command Hi | **XX** | **default** |
| Order command Lo | **XX** | **Default** |
| Byte Count | **02** | **Số Byte data đọc được** |
| Data 0 | **00** | Giá trị là 0x00 |
| Data 1 | **01** | +0x01: Xóa lỗi chung của thiết bị |
|  |  | Nếu phát sinh bổ sung tiếp |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ phần cứng:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **09** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** | **Default** |
| Byte Count | **02** | **Số Byte Data đã nhận** |
| Status | **XX** | **=00: xác nhận đã nhận dữ liệu ok**  **~~=01: Có lỗi nhận dữ liệu sai CRC~~**  **~~=02: Lỗi không hỗ trợ FUNTION này.~~** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

# FUNCTION 0x80: Điều khiển động cơ trong máy RVM

**PC🡪thiết bị**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **80** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh H** |
| Order command Lo | **XX** | **Số thứ tự lệnh L** |
| Byte Count | **02** | **Số Byte data đọc được** |
| Data 0 | **XX** | Vị trí motor điều khiển:  =0x01: Động cơ thang máy  =0x02: Động cơ tấm che  =0x03: Động cơ khoá cửa hộc quà |
| Data 1 | **XX** | Giá trị là  =0x00: dừng lại  =0x01: xoay trái/đóng lại  =0x02: xoay phải/mở ra |
|  |  | Nếu phát sinh bổ sung tiếp |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ phần cứng:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **80** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** | **Tương ứng lệnh đã nhận** |
| Byte Count | **02** | **Số Byte Data đã nhận** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

# FUNCTION 0xFA: Trạng thái cơ khí trong máy ROCSTAR

**Thiết bị🡪PC**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **FA** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh H** |
| Order command Lo | **XX** | **Số thứ tự lệnh L** |
| Byte Count | **02** | **Số Byte data đọc được** |
| Data 0 | **XX** | Loại cửa:  =0x01: Cửa hộc quà |
| Data 1 | **XX** | Giá trị là  =0xFF: cửa mở  =0x00: cửa đóng  =0x01: cửa đang khoá |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ PC:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **FA** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** | **Tương ứng lệnh đã nhận** |
| Byte Count | **02** | **Số Byte Data đã nhận** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

# FUNCTION 0xFB: Kiểm tra model, firmware phần cứng

PC 🡪 HW.

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **FB** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh gửi Hi** |
| Order command Lo | **XX** | **Số thứ tự lệnh gửi Lo (lệnh mới thì tăng giá trị HiLo này lên, để tránh trùng lệnh)** |
| Byte Count | **00** | **Default==00:không có data theo sau** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

**Phản hồi từ phần cứng:**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** | **address =03: mặc định cho máy RVM** |
| FUNTION | **FB** | **=FUNTION nhận được** |
| Order command Hi | **XX** | **Tương ứng lệnh đã nhận** |
| Order command Lo | **XX** |  |
| Byte Count | **04** | **Default** |
| DataHi | **XX** | **[firmware Hi]** |
| DataLo | **XX** | **[firmware Lo]** |
| DataHi | **XX** | **[model Hi]** |
| DataLo | **XX** | **[model Lo]** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

**Qui trước truyền: theo số gồm 2 byte:**

**Thí dụ firmware: 00 01: mapping lên app tương ứng là: RVM2.001**

**Thí dụ model: 00 01: mapping lên app tương ứng là: AM\_AQUA38/40**

# FUNCTION 0xFE: Báo lỗi truyền dữ liệu

* Khi PC gửi cho HW, hoặc HW gửi cho PC: mà phát hiện dữ liệu bị lỗi, sẽ dùng function này để thông báo.
* Tất cả lỗi liên quan đến truyền dữ liệu đều gửi qua function này:

-sai function

-sai CRC

-không hỗ trợ function…

**PC🡪HW hoặc HW🡪PC**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example**  **(HEX)** | **Note** |
| ADDRESS | **03** |  |
| FUNTION | **FE** | **default** |
| Order command Hi | **XX** | **Giá trị nhận được của dữ liệu** |
| Order command Lo | **XX** | **Giá trị nhận được của dữ liệu** |
| Byte Count | **02** | **Số Byte data đọc được** |
| Data 0 | **XX** | **Function bị sai** |
| Data 1 | **XX** | Trạng thái lỗi dữ liệu  **=01: Có lỗi nhận dữ liệu sai CRC**  **=02: Lỗi không hỗ trợ FUNTION này**  **=giá trị khác: không xác định** |
| CRC lo | **1 byte** | **Hardware tính CRC** |
| CRC hi | **1 byte** | **Hardware tính CRC** |

**Phản hồi/hoặc không cần phản hồi gói này: biết để gửi lại gói khác**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Example** | **Note** |
| ADDRESS | **03** | **address =03** |
| FUNTION | **FE** | **default** |
| Order command Hi | **XX** | **Số thứ tự lệnh gửi Hi** |
| Order command Lo | **XX** | **Số thứ tự lệnh gửi Lo (lệnh mới thì tăng giá trị HiLo này lên, để tránh trùng lệnh)** |
| Byte Count | **02** | **Số byte data nhận được** |
| CRC lo | **1 byte** | **PC tính** |
| CRC hi | **1 byte** | **PC tính** |

# Định dạng khung truyền dữ liệu giữa các thiết bị.

ALTA Media June 02,2025

**Mục đích:** Tạo 1 khung truyền dữ liệu chuẩn giữa PC, các thiết bị phần cứng với nhau, sử dụng trong công ty ALTA Media.

**Thiết lập truyền thông: sử dụng RTU mode (Remote Terminal Unit)**

Coding system: 8 bit binary, hexadecimal 0-9, A-F.

Baud rate: mặc định là 115200

Bits/byte: 1 start bit, 8 data bits, no parity, 2 stop bit .

LSB truyền trước. (Least Significant Bit)

**Định dạng khung truyền :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| START | ADDRESS | Order command | FUNCTION | DATA | CRC | END |
| T1-T2-T3-T4 | 8 BITS | 16 BITS | 8 BITS | nx8 BITS | 16 BITS | T1-T2-T3-T4 |

**START**: In RTU mode, messages start with a silent interval of at least 3.5 character times.  
This is most easily implemented as a multiple of character times at the baud rate  
that is being used on the network (shown as T1–T2–T3–T4 in the figure below).  
The first field then transmitted is the device address.

**ADDRESS**: địa chỉ thiết bị nhận, 1 byte

**Order command**: số thứ tự lệnh điều khiển, 2 byte

**FUNTION**: mã điều khiển, 1 byte.

**DATA**: dữ liệu truyền

**CRC**: Cyclical Redundancy Check: kiểm tra lỗi, 2 bytes

**END**: Kết thúc khung truyền.

**CRC:**

Chức năng CRC: được dùng để kiểm tra khung truyền lỗi hay không, CRC do thiết bị gửi tính, gửi cùng khung truyền, thiết bị nhận tính lại dựa trên dữ liệu nhận được, và so sánh kết quả tính được với CRC nhận được: nếu 2 kết quả không bằng nhau thì xem như có lỗi xuất hiện.

**CRC Generation**

**The Cyclical Redundancy Check (CRC) field is two bytes, containing a 16–bit**

**binary value. The CRC value is calculated by the transmitting device, which**

**appends the CRC to the message. The receiving device recalculates a CRC**

**during receipt of the message, and compares the calculated value to the actual**

**value it received in the CRC field. If the two values are not equal, an error results.**

**The CRC is started by first preloading a 16–bit register to all 1’s. Then a process**

**begins of applying successive 8–bit bytes of the message to the current contents**

**of the register. Only the eight bits of data in each character are used for generating**

**the CRC. Start and stop bits, and the parity bit, do not apply to the CRC.**

**During generation of the CRC, each 8–bit character is exclusive ORed with the**

**register contents. Then the result is shifted in the direction of the least significant**

**bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is**

**extracted and examined. If the LSB was a 1, the register is then exclusive ORed**

**with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.**

**This process is repeated until eight shifts have been performed. After the last**

**(eighth) shift, the next 8–bit character is exclusive ORed with the register’s current**

**value, and the process repeats for eight more shifts as described above. The final**

**contents of the register, after all the characters of the message have been applied,**

**is the CRC value.**

**A procedure for generating a CRC is:**

**1, Load a 16–bit register with FFFF hex (all 1’s). Call this the CRC register.**

**2. Exclusive OR the first 8–bit byte of the message with the low–order byte**

**of the 16–bit CRC register, putting the result in the CRC register.**

**3. Shift the CRC register one bit to the right (toward the LSB), zero–filling the**

**MSB. Extract and examine the LSB.**

**4. (If the LSB was 0): Repeat Step 3 (another shift).**

**(If the LSB was 1): Exclusive OR the CRC register with the polynomial**

**value A001 hex (1010 0000 0000 0001).**

**5. Repeat Steps 3 and 4 until 8 shifts have been performed. When this is**

**done, a complete 8–bit byte will have been processed.**