# Usart Boot loader Protocol

# Status

* Boot loader
* App execute

Figure 1. Boo loader Code Sequence with USART

No

Yes

Receive command?

Yes

App execute

No

Yes

Jump to App Address

Run App

Set update flag

Update command

Go command routine

Is update flag set?

Yes

No

Command

Is GO command?

Table 1.Memory mapping of boot loader

|  |  |  |
| --- | --- | --- |
|  | Start Address | End Address |
| Bootloader | 0x0800 0000 | 0x0800 FFFF |
| App | 0x0801 0000 | 0x0803 F7FF |
| State | 0x0803 F800 |  |
| Update |  | 0x0803 FFFF |

## State and update flag

# Commands

* Get Version: *get boot loader version.*
* Get ID: *get chip ID*
* Read Memory:
* Go: *jump to app.*
* Write Memory:
* Erase:
* Set Command : *reset*.
* Update: *need update firmware after restart.*

Command code:

|  |  |  |
| --- | --- | --- |
| Command | Command code | Command description |
| Get Version | 0x01 | Gets the version of boot loader. |
| Get ID | 0x02 | Gets the chip ID |
| Read Memory | 0x11 | Reads up to 256 bytes of memory starting from an address specified by the application |
| Go | 0x21 | Jumps to user application and executes it. |
| Write Memory | 0x31 | Writes up to 256 bytes of memory to the RAM of flash starting from an address specified by the application. |
| Erase | 0x43 | Erases from one to all the flash memory pages. |
| Set Command | 0x63 | Reset command |
| Update | 0x83 | Sets update flag. |

*\*Each packet is either accepted (ACK answer) or discarded (NACK answer):*

*•* ***ACK = 0x79*** *•* ***NACK = 0x1F***

**Command Type:**

Command-type (2+13+5 bytes = 20bytes)

Data-type (2+6+1024+5bytes = 1037bytes)

Host <-> Device : H->D 0x2\_ ; D->H 0x8\_

**Ex: H->D C-type: 0x2C ; D->H D-type: 0x8D**

1. Get Version

Figure 2. Get Version Command Workflow

1. Send Get Version Command
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D & Type (1) | Command (1) | Zero(13) | End (1) | CRC32 (4) |
| 0x2C | 0x01 | 0 | 0xFF |  |

1. Receive Command and Data (ACK)
   1. C-type: 20 bytes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| H-D&Type(1) | Command(1) | Version(1) | Zero(12) | End(1) | CRC32(4) |
| 0x8C | 0x79 | 0x01 | 0 | 0xFF |  |

\*Version=0x01 =>version 0.1

1. Receive Command and Data (NACK)
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D & Type (1) | Command (1) | Zero(13) | End (1) | CRC32 (4) |
| 0x8C | 0x1F | 0 | 0xFF |  |

1. Get ID

Figure 3. Get ID Command Workflow

1. Send Get ID Command
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D & Type (1) | Command (1) | Zero(13) | End (1) | CRC32 (4) |
| 0x2C | 0x02 | 0 | 0xFF |  |

1. Receive Command and Data (ACK)
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D&Type(1) | Command(1) | ID(13) | End(1) | CRC32(4) |
| 0x8C | 0x79 | Ex: “STM32F107VCT6” | 0xFF |  |

1. Receive Command and Data (NACK)
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D & Type (1) | Command (1) | Zero(13) | End (1) | CRC32 (4) |
| 0x8C | 0x1F | 0 | 0xFF |  |

1. Read Memory

Figure 4. Read Memory Command Workflow

1. Send Read Memory Command
   1. C-type: 20 bytes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| H-D&Type(1) | Cmd (1) | Start\_addr (4) | | Size(2) | Zero(7) | End (1) | CRC32 (4) |
| 0x2C | 0x11 | | 0x0800 4000 | 0x0400 | 0 | 0xFF |  |

1. Receive Command and Data (ACK)
   1. D-type: 1031 bytes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| H-D&Type(1) | Command(1) | Zero(6) | Memory Data(1024) | End(1) | CRC32(4) |
| 0x8D | 0x79 | 0 | Memory Data | 0xFF |  |

1. Receive Command and Data (NACK)
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D & Type (1) | Command (1) | Zero(13) | End (1) | CRC32 (4) |
| 0x8C | 0x1F | 0 | 0xFF |  |

1. Go

Figure 5. Go Command Workflow

1. Send Go Command
   1. C-type: 20 bytes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| H-D&Type(1) | Cmd (1) | Jump\_addr (4) | | Zero(9) | End (1) | CRC32 (4) |
| 0x2C | 0x21 | | 0x0801 0000 | 0 | 0xFF |  |

1. Receive Command and Data (ACK)
   1. D-type: 1031 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D&Type(1) | Command(1) | Zero(13) | End(1) | CRC32(4) |
| 0x8C | 0x79 | 0 | 0xFF |  |

1. Receive Command and Data (NACK)
   1. C-type: 20 bytes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| H-D&Type(1) | Command (1) | Error(1) | Zero(12) | End (1) | CRC32 (4) |
| 0x8C | 0x1F | 0x00(check sum error) | 0 | 0xFF |  |
| 0x8C | 0x1F | 0xFF(state error) | 0 | 0xFF |  |

1. Write Memory

Figure 6. Write Command Workflow

1. Send Read Memory Command
   1. C-type: 1037 bytes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| H-D&Type(1) | Command(1) | Start\_addr (4) | Size(2) | Write Data(1024) | End(1) | CRC32(4) |
| 0x2D | 0x31 | 0x0801 0000 | 0x0400 | Data | 0xFF |  |

1. Receive Command and Data (ACK)
   1. D-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D&Type(1) | Command(1) | Zero(13) | End(1) | CRC32(4) |
| 0x8C | 0x79 | 0 | 0xFF |  |

1. Receive Command and Data (NACK)
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D & Type (1) | Command (1) | Zero(13) | End (1) | CRC32 (4) |
| 0x8C | 0x1F | 0 | 0xFF |  |

1. Erase Memory Command

Figure 7.Erase Command Workflow

1. Send Go Command
   1. C-type: 20 bytes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| H-D&Type(1) | Cmd (1) | Erase Page (1) | Zero(12) | End (1) | CRC32 (4) |
| 0x2C | 0x43 | 0x00~80 (128 page) | 0 | 0xFF |  |

1. Receive Command and Data (ACK)
   1. D-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D&Type(1) | Command(1) | Zero(13) | End(1) | CRC32(4) |
| 0x8C | 0x79 | 0 | 0xFF |  |

1. Receive Command and Data (NACK)
   1. C-type: 20 bytes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| H-D&Type(1) | Command (1) | Zero(13) | End (1) | CRC32 (4) |
| 0x8C | 0x1F | 0 | 0xFF |  |

1. Set Command
2. Update