**Local Communication Protocol**

# Serialport

## Physical layer:

* Biggest gap time between 2 bytes are 3 bytes time (base on the bard rate).
* Max response time is 250ms;
* Uart parameter fixed: 8,1,N; 115200;

## Basic frames definition:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 byte | 1 byte | 1 byte | 1 byte | 1 byte | Multi bytes | 2 byte |
| Address from | Address to | ID | Function code | length | data | CRC |

*#define MAX\_BUFF\_DATA\_LEN 255*

***typedef******union***

*{*

*uint8\_t data[MAX\_BUFF\_DATA\_LEN + 5];*

***struct*** *\_St1*

*{*

*uint8\_t dst;*

*uint8\_t src;*

*uint8\_t id;*

*uint8\_t function;*

*uint8\_t length;*

*uint8\_t dataBuff[MAX\_BUFF\_DATA\_LEN];*

*}St;*

*}Layer2Frame;*

## Address:

|  |  |  |
| --- | --- | --- |
| Value | comments |  |
| FF | Reserved for broadcasts |  |
| 0~0xFE | Address for all devices include master and slave |  |

## ID

The ID is normally increase by itself, the ID is unique for one request and response;

## Function code

|  |  |  |
| --- | --- | --- |
| bit 0~ bit 5 | Bit 6 | Bit 7 |
| Function code | Frame type | Status |
| 0 ~ 0x3F | * 0: request * 1: response | * 0: OK; * 1: Error |

Currently support commands below:

* *CMD\_POLL* = 0x0,
* *CMD\_READ\_OBJ* = 0x01,
* *CMD\_WRITE\_OBJ* = 0x02,
* *CMD\_READ\_TYPE* = 0x03,
* *CMD\_READ\_LENGTH* = 0x04,
* *CMD\_READ\_ATR\_NUM* = 0x05,
* *CMD\_READ\_RANGE* = 0x06,
* *CMD\_READ\_MEM* = 0x10,
* *CMD\_WRITE\_MEM* = 0x11,
* *CMD\_BURST\_VALUE* = 0x20,

|  |  |  |
| --- | --- | --- |
| CMD name | Request Data | Response |
| *CMD\_POLL* | Todo | Todo |
| *CMD\_READ\_OBJ* | 0: Sub ID  1: Obj ID  2: Atribute ID | 0: Sub ID  1: Obj ID  2: Atribute ID  3: Data Byte 0  4: Data Byte 1 |
| *CMD\_READ\_TYPE* |
| *CMD\_READ\_LENGTH* |
| *CMD\_READ\_ATR\_NUM* |
| *CMD\_READ\_RANGE* |
| *CMD\_WRITE\_OBJ* | 0: Sub ID  1: Obj ID  2: Atribute ID  3: Data Byte 0  4: Data Byte 1 | 0: Sub ID  1: Obj ID  2: Atribute ID  3: Return code |
| *CMD\_READ\_MEM* | 0~3: address  4: length (shall <= 250) | 0~3: address  4: length (shall <= 250)  5: data 0;  6: data 1; |
| *CMD\_WRITE\_MEM* | 0~3: address  4: length (shall <= 250)  5: data 0;  6: data 1; | 0~3: address  4: length (shall <= 250)  5: error code |
| *CMD\_BURST\_VALUE* | Todo | Todo |

# CAN2.0:

## Physical layer

Use Can 2.0 extend frames;

## Head

Total 29 bit for can 2.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0~7 | 8~15 | 16~21 | 22~23 | 24~25 | 26~28 |
| Dst address | Src address | Function code | Command Type | Frame type | Ex |

## Address:

Refer to Serial port

## CMD Type:

|  |  |
| --- | --- |
| 0x01 | Response ok |
| 0x02 | request |
| 0x00 | Broadcast |
| 0x03 | Response error |

## Frame type

|  |  |
| --- | --- |
| 0x01 | multi frames start 🡪 the data part not have the frame ID, the EX is 0; |
| 0x02 | Multi frames 🡪 the EX parts will increase by 1; |
| 0x03 | multi frames ends🡪 the EX parts will increase by 1; |
| 0x00 | No multi frames |

## Ex – multi-frames

The ID is normally increase by itself, the ID is unique for one request and response;

1. ID which is used to identify this is the response to the request id;
2. the EX parts will increase from 0 when multi frames; the data is sent with 8 bytes max for one frame;

## Function code

Refer to Serial port

## Data

Refer to Serial port