**Communication Protocol**

**ⅰ、Sub-layer parameter promise of data link**

Baud rate：115200

Parity check digit：None

Data digit： 8 bits

Stop digit：1bit

**ⅱ、Frame format of link layer**

**1. Frame format**

Message format：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STX** | **LENGTH** | **Message** | **ETX** | **LRC** |

**2. Definition of each field and control word**

|  |  |  |
| --- | --- | --- |
| **Name** | **Signification** | **Value** |
| STX | Frame begin symbol | 0x02 |
| LENGTH | Length (2 bytes) | Decided by the length of Message |
| Message | Sent data | Decided by the content of Message |
| ETX | Frame finish symbol | 0x03 |
| LRC | Checkout sum | Decided by the content between STX and ETX, not including STX |

**ⅲ、Message field structure**

**1. Message format**

|  |  |  |  |
| --- | --- | --- | --- |
| **LENGTH** | **TYPE** | **DATA TYPE** | **DATA** |

Field definition：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field | Field length | Content |
| 1 | LENGTH | 2 | Data portion length |
| 2 | TYPE | 0x38 | Message type |
| 3 | DATA TYPE | 0x01 | Sort |
| 3 | DATA | xx | Data |

**ⅳ、Measurement result transmission**

**Various DATA field format and content：**

Data content from high to low is：

1. 21 parameter（same with interface）, each 2 bytes, totally 38 bytes.
2. WBC histogram，256 byte
3. RBC histogram，256 byte
4. PLT histogram，128 bytes.
5. Surveyor’s pole parameter, sequence：WBC surveyor’s pole，4 bytes；RBC surveyor’s pole，2 bytes；PLT surveyor’s pole，2 bytes；totally 8 bytes。
6. R1，R2，R3，R4，PM alarm, each 1 byte，totally 5 bytes.
7. WBC counting time，2 bytes
8. RBC counting time，2 bytes
9. Sample No., 6 bytes
10. Date/Time , 6 bytes

**ⅴ、Program of transfer end**

case TRANSFER\_RESULT:

{

tx\_temp[0] = 0x02; //**STX**

tx\_temp[1] = 0x02; //**LENGHT**

tx\_temp[2] = 0xc5;

tx\_temp[3] = 0x38; // **TYPE**

tx\_temp[4] = 0x01; //**DATA TYPE**

tx\_temp[5] = (uchar)(wbc >> 8); // WBC

tx\_temp[6] = (uchar)wbc;

tx\_temp[7] = (uchar)(w\_scr >> 8); // LY%

tx\_temp[8] = (uchar)w\_scr;

tx\_temp[9] = (uchar)(w\_mcr >> 8); // MO%

tx\_temp[10] = (uchar)w\_mcr;

tx\_temp[11] = (uchar)(w\_lcr >> 8); // GR%

tx\_temp[12] = (uchar)w\_lcr;

tx\_temp[13] = (uchar)(w\_scc >> 8); // LY#

tx\_temp[14] = (uchar)w\_scc;

tx\_temp[15] = (uchar)(w\_mcc >> 8); // MO#

tx\_temp[16] = (uchar)w\_mcc;

tx\_temp[17] = (uchar)(w\_lcc >> 8); // GR#

tx\_temp[18] = (uchar)w\_lcc;

tx\_temp[19] = (uchar)(rbc >> 8); // RBC

tx\_temp[20] = (uchar)rbc;

tx\_temp[21] = (uchar)(hgb >> 8); // HGB

tx\_temp[22] = (uchar)hgb;

tx\_temp[23] = (uchar)(hct >> 8); // HCT

tx\_temp[24] = (uchar)hct;

tx\_temp[25] = (uchar)(mcv >> 8); // MCV

tx\_temp[26] = (uchar)mcv;

tx\_temp[27] = (uchar)(mch >> 8); // MCH

tx\_temp[28] = (uchar)mch;

tx\_temp[29] = (uchar)(mchc >> 8); // MCHC

tx\_temp[30] = (uchar)mchc;

tx\_temp[31] = (uchar)(rdw\_cv >> 8); // RDW\_cv

tx\_temp[32] = (uchar)rdw\_cv;

tx\_temp[33] = (uchar)(rdw\_sd >> 8); //rdw\_sd

tx\_temp[34] = (uchar)rdw\_sd;

tx\_temp[35] = (uchar)(plt >> 8); // PLT

tx\_temp[36] = (uchar)plt;

tx\_temp[37] = (uchar)(mpv >> 8); // MPV

tx\_temp[38] = (uchar)mpv;

tx\_temp[39] = (uchar)(pdw >> 8); // PDW

tx\_temp[40] = (uchar)pdw;

tx\_temp[41] = (uchar)(pct >> 8); // MPV

tx\_temp[42] = (uchar)pct;

for( i = 0; i < 256; i++ )

{

tx\_temp[43 + i] = wbchist[i];

tx\_temp[299 + i] = rbchist[i];

if( i < 128 )

tx\_temp[555 + i] = plthist[i];

}

tx\_temp[683] = swbclline;

tx\_temp[684] = swbchline;

tx\_temp[685] = lwbclline;

tx\_temp[686] = lwbchline;

tx\_temp[687] = rbclline;

tx\_temp[688] = rbchline;

tx\_temp[689] = pltlline;

tx\_temp[690] = plthline;

tx\_temp[691] = alarm.r1;

tx\_temp[692] = alarm.r2;

tx\_temp[693] = alarm.r3;

tx\_temp[694] = alarm.r4;

tx\_temp[695] = alarm.pm;

tx\_temp[696] = alarm.wbctime >> 8;

tx\_temp[697] = alarm.wbctime;

tx\_temp[697] = alarm.rbctime >> 8;

tx\_temp[699] = alarm.rbctime;

tx\_temp[700] = bh[5];

tx\_temp[701] = bh[4];

tx\_temp[702] = bh[3];

tx\_temp[703] = bh[2];

tx\_temp[704] = bh[1];

tx\_temp[705] = bh[0];

tx\_temp[706] = timer.year;

tx\_temp[707] = timer.month;

tx\_temp[708] = timer.date;

tx\_temp[709] = timer.hour;

tx\_temp[710] = timer.minute;

tx\_temp[711] = timer.second;

tx\_temp[712] = 0x03; //**ETX;**

tx\_temp[713] = 0;

for( i = 1; i < 713; i++ )

{

tx\_temp[713] = tx\_temp[i] ^ tx\_temp[713]; //**LRC**

}

}

break;

**Cable Connect with PC**

RS232 connector (DB9)

First RS232 Second RS232

1-------------------------------------------- (Nothing)

2---------------------------------------------3

3---------------------------------------------2

4-------------------------------------------- (Nothing)

5---------------------------------------------5

6-------------------------------------------- (Nothing)

7-------------------------------------------- (Nothing)

8-------------------------------------------- (Nothing)

9-------------------------------------------- (Nothing)

1

5

1

5

Ground line

Crust

If your computer don’t have the COM&RS232 (9pin) interface, you could use one USB TO RS232 cable connect the cable you made yourself.