Yeacode TCP/IP MODBUS

Version No

Version	Issue date	Version Description	Author
V1.0.0.1	2020/03/30		Tanbo
V1.0.0.2	2020/04/13		Tanbo
V1.0.0.3	2020/12/29	Add description of callback interface	Tanbo
V1.0.0.4	2021/07/19	Description update	amos
V1.0.0.5	2022/07/15	Increase the number of acquired cache description	tanbo
V1.0.0.6	2022/12/15	Increase cleanup cache	tanbo
V1.0.0.7	2023/01/31	Add suspended printing, continued printing, and test information	tanbo

Content

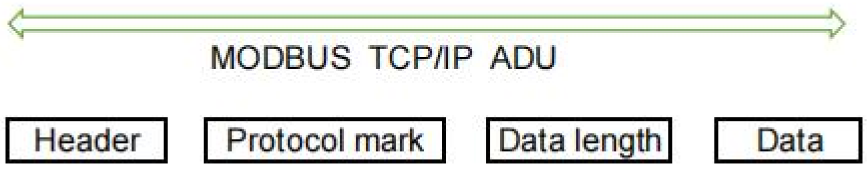
（一）Preface

This article defines the MODBUS protocol service of yeacode printer based on TCP-IP. The printer serves as the server to provide communication services, receives the connection request from client and responds to the client's MODBUS function application.

Both PLC and printer software can be set as the IP address of the server; the port is 20001.

Note: When the client-side generates an ADU service request, a TCP sending frame can only contain one Modbus TCP/IP ADU, and it must wait for the previous ADU to return before sending a new ADU frame.

MODBUS application data request format as shown



Message format definition

Domain	Type	number of bytes	Description
Header	bytes	2	Transmission symbol：0xEB 0x01
Protocol mark	bytes	2	Action command
Data length	bytes	4	Transmission parameter length (contain \0 at the end)
Data	bytes	JSON string (Utf8)	Parameter content (end in \0)Network byte order

Protocol mark definition

Value	Description
0x00 0x01	Get system status
0x00 0x02	Get printing status
0x00 0x03	Get printing log

0x00 0x04	Set dynamic data
0x00 0x05	Start printing
0x00 0x06	Stop printing
0x00 0x07	Get printing cache
0x00 0x08	Special parameter setting interface
0x00 0x09	Register log/output callback
0x00 0x0A	Logout log/output callback
0x00 0x12	Get the cache quantity
0x00 0x14	Clean up the cache
0x00 0x15	Suspension of printing
0x00 0x16	Continue to print
0x00 0x63	Test information

Response message format definition

Domain		Type	number of bytes	Description
Header		bytes	2	Transmission symbol
Protocol mark		bytes	2	Action command
Data length		bytes	4	Return parameter length,Data interception according to length
Data		bytes	Variable length	JSON string (Utf8) (end in \0)

（二）API Description

1． Send dynamic data

Send the printing content to the printer and return the processing result, and return the response content in JSON format .

注：实际接收到的字段在本文档中未描述时，请忽略该字段。

Request format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x04	2 bytes
Data length	0x00 0x10 0x00 0x10 (contain \0 at the end)	4 bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Data format /*JSON ,support UTF-8/GBK format */:

```
    {
        "text":   [
        {// If it contains images, write in the following format
          "metaname": "img1",
          "is_image": 1,//1=image, 0=text
          "metadata_pos": 0,//The starting position of the current image data in all images data at the end.
          "metadata_len": 18494// The content length of the current image data in all images data at the end .
          "hide_flag": 0,// Indicates whether to hide all objects containing dynamic data sources (not hidden by default: 0)
          "metaimage_dpi":300,//range（0-1200）image DPI value
          "metaimage_has_rolate":0,//range（0-360）rotation angle
          "compress":0//0=decompression,1=compression
    },
    {//If it contains images, write in the following format
          "metaname": "img2",
          "is_image": 1,//1=image,0=text
          "metadata_pos": 18494,//The starting position of the current image data in all image data at the end.
          "metadata_len": 10000//The starting position of the current image data in all image data at the end.
          "hide_flag": 0,//Indicates whether to hide all objects containing dynamic data sources (not hidden by default: 0)
          "metaimage_dpi":300,//image DPI value,value range(0-1200)
          "metaimage_has_rolate": 0,//image rotation angle,default 0,value range (0-360)
          "compress":0//image compressed or not,default 0, 0=decompression,1=compression
          },
    {//If it contains images, write in the following format
          "metaname": "txt",
                "is_image": 0,//1=image,0=text
                "metadata": "12321321"// the printed text data
                "hide_flag": 0,//Indicates whether to hide all objects containing dynamic data sources (not hidden by default: 0)
             }
          ],
          "repeat_times": 1,// the printing times of current data source (-1： cycle printing of currently sent data )
          "app_mode": 1000, // The current value passed into the assigned number
          "direct": -1, //the printing direction of current data (-1：no setting, 0：left to right，1：right to left)
          "cover_flag":0 ,//0: not override previous data , 1： override previous data. In addition to the special requirements based on the last data, non-overwrite mode is preferred (0) .
          "hide_flag": 0,//Indicates whether to hide all objects containing dynamic data sources (not hidden by default: 0)
       }+\0+all images data（0x42 0x4d ........）
```

Note： Follow below operation for sending large images need to be compressed (compress=1)：

1, Import the libImageProcessing.a static library in ImageProcessing \ Lib \ x86 into the project while referencing the header file ImageProcessing\Include\yccompress.h。

2, Call ym_bmp_rlecompress function to compress .

C++ as example：

```
      QByteArray bmpData;//Original image data obtained
      If(bmpData.isNull()){
            return;
      }
      char *ptrIn = bmpData.data();
      char *pcompressBtmData = NULL;
      int nCompressLen = 0;
      Int compressResult = ym_bmp_rlecompress(ptrIn, &pcompressBtmData,&nCompressLen);//compress data
      if(compressResult == 0){
            QByteArray byte(pcompressBtmData,nCompressLen);
            bmpData = byte;/replace data
      }
      if(pcompressBtmData){
            free(pcompressBtmData);//free
      }
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2 bytes

Protocol mark	0x00 0x04	2 bytes
Data length	0x00 0x10 0x00 0x10	4 bytes According to this data length, intercept the corresponding length content in the data
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format):

{

"status": 0//0: success，1: failure，49：Dynamic source database cache limit exceeded,

50：printing not start

}

2． Get printing status

Get operation status of current printer

Request format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x02	2 bytes
Data length	0x00 0x10 0x00 0x10 (contain \0 at the end)	4 bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON ,support UTF-8/GBK format */:

{

"group_id":0//printhead group ID

}

Response format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x02	2 bytes
Data length	0x00 0x10 0x00 0x10	4 bytes According to this data length, intercept the corresponding length content in the data

Data	JSON string (Utf8)	Variable length (end in \0)

Return value Utf8 string description (JSON data format)：

```
{
        "print_id": 0,//printhead group ID
        "print_status": 0,//0: print end , 1: printing
        "reprint_status": 0,//0: print end ,2：  re-print
        "meta_read_end": 0,//1 database，  2 Counter maximum mark
        "print_errno": 0,//printing error ,SDK_ERR_PRINT_DB_CONNECT = 8,
                                                          SDK_ERR_PRINT_DB_READ = 9,
        "print_yield": 0,//output
        "line_speed": 0.000000//line speed m/s
}
```

3． Get system status

Get current status of printing system

Request format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x01	2 bytes
Data length	0x00 0x00 0x00 0x00	4 bytes

Response format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x01	2 bytes
Data length	0x00 0x10 0x00 0x10	4 bytes According to this data length, intercept the corresponding length content in the data
Data	JSON string (Utf8)	Variable length (end in \0)

Return value Utf8 string description (JSON data format)：

```
{
        "device_name": “Inkjet”,
```

"tcp_version": 200330,
"net_status": 0,//0: normal,1: printhead network error,2: controller network error
3: controller IP conflict
"ink_status": 0,//-1: Ink cartridge incorrect identification , 0: normal, 5: ink is running out, 7: ink is used out
"ciss_status": 20,//17 &18: the bottled ink is running out , 19&20: the bottled ink is used out
"ph_status": 0,//0: normal, others： error
"elec_status": 0,//0:normal, others： error
"wheel_status": 0,//0: normal, others： error
"heat_status": 0,//0: normal, others： error
"uv_status": 0//0:uv disconnected, 1:uv connected
}

4． Get printing log

Get the printed data

Request format

Domain	Format	length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x03	2 bytes
Data length	0x00 0x10 0x00 0x10 (contain \0 at the end)	4 bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON, supportUTF-8/GBK format */:

```
{
    "group_id":0// printhead group ID
}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x03	2 bytes
Data length	0x00 0x10 0x00 0x10	4 bytes According to this data length, intercept the corresponding length

		content in the data
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{
        "text": [{
                "index": 3950312,
                "metatype": 3,
                "metaname": "counter 1",//data source name
                "metadata": "000003950311"//log content
        }],
        "time": 946779773,//Log generation time, the seconds since 1970
        "group_id": 0,
        "index": 3950312,
        "yield": 3950312,
        "status": 0//0： printed 1： covered
}
```

5. Start printing

Open the specified data for printing

Request format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x05	2 bytes
Data length	0x00 0x10 0x00 0x10 (contain \0 at the end)	4 bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON, support UTF-8/GBK format */:

```
        {
                "print_file": "222.ym"// the full name of printing data
        }
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x05	2 bytes

Data length	0x00 0x10 0x00 0x10	4 bytes According to this data length, intercept the corresponding length content in the data
Data	JSON string (Utf8)	Variable length (end in \0)

Return value Utf8 string description (JSON data format)：

```
{
    "status": "0" //0: success,1: failure,-1: failure ,4: printing ,32: ink used
out ,50: printing not start
}
```

6． Stop printing

Stop current printing

Request format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x06	2 bytes
Data length	0x00 0x00 0x00 0x00	4 bytes

Response format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x06	2 bytes
Data length	0x00 0x10 0x00 0x10	4 bytes According to this data length, intercept the corresponding length content in the data
Data	JSON string (Utf8)	Variable length (end in \0)

Return value Utf8 string description (JSON data format):

```
{
      "status": "0" //0: success,1: failure
}
```

7. Get printing cache

Get cache data of current printer

Request format

Domain	Format	Length
Header	0xEB 0x01	2 bytes
Protocol mark	0x00 0x07	2 bytes
Data length	0x00 0x10 0x00 0x10 (contain \0 at the end)	4 bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Data format /*JSON, support UTF-8/GBK format */:

```
{
"group_id":0//printhead group ID
}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x07	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes According to this data length, intercept the corresponding length content in the data
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format):

```
{
"print_cache": [[{
            "index": 60,//number
            "status": 0,//0： printer 1： not printed 2： under printing 3： about to print
            "metatype": 5,
            "metaname": "dynamic text 1",//data source name
            "metadata": "12121"/content to be sending
        }]]
    }
```

8． Special parameter setting interface

Special parameters for interaction between different systems

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x08	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes (Contains \0 ending)
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON, support UTF-8/GBK format */:

```
{"offset_value":1000}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x08	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{"status": 0}
```

9． Register log callback

Obtain log info in real time

Note：printhead software version should be higher than 210101

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x09	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes (Contains \0 ending)
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON data format, support UTF-8/GBK format*/:

```
{
   "regist_type":1//1： log callbacl, 2： output callback
}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x09	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：{"status": 0}

10. Logout log callback

stop obtaining log info in real time

Note： printhead software version should be higher than 210101

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x0A	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes (Contains \0 ending)

Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON data format, support UTF-8/GBK format*/:

```
{
   "regist_type":1//1： log callback ,2： output callback
}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x0A	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{"status": 0}
```

11． Log callback parameters

Log Info

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x0C	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{
    "group_id": 0,
    "index": 11343,
    "yield": 11343,//output
    "status": 0,//0： printed   1： overwritten
    "text": [{
        "metatype": 3,
            "metaname": "counter 1",//data source name
            "metadata": "1344"//content
      }, {
            "metatype": 4,
            "metaname": "shift 1",
            "metadata": "S1"
      }]
}
```

12. Registered output callback

obtain output info in real time

Note：printhead software version should be higher than 210101

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x09	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes (Contains \0 ending)
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON data format, support UTF-8/GBK format*/:

```
{
   "regist_type":2//1：log callback , 2：output callback
}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x09	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
      {"status": 0}
```

13． Logout output callback

stop obtaining output info in real time

Note：printhead software version should be higher than 210101

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x0A	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes (Contains \0 ending)
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON data format, support UTF-8/GBK format*/:

```
{
  "regist_type":2//1：log callback, 2：output callback
}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x0A	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{"status": 0}
```

14． Output callback parameter

Output info

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes

Protocol mark	0x00 0x0B	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{
    "yield":41620,//output
    "group_id":0//printhead group ID
}
```

15． Get the cache quantity

Get the cache quantity。

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x12	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Data format/*JSON data format, support UTF-8/GBK format*/:

```
{
"group_id":0//Spithead groupID
}
```

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x12	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{
       "status": "1"//Cache quantity
 }
```

16． Clean up the cache

Clean up the cache。

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x14	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x14	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format):

```
{
       "status": "0"//0：0: Success, other: failure
 }
```

17． Suspension of printing

Suspension of printing。

Request format

Domain	Format	Length

Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x15	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x15	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{
      "status": "0"//0：0: Success, other: failure
 }
```

18． Continue to print

Continue to print。

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x16	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Response format

Domain	Format	Length

Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x16	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{
        "status": "0"//0：0: Success, other: failure
 }
```

19． Test information

Test information。

Request format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x63	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Response format

Domain	Format	Length
Header	0xEB 0x01	2bytes
Protocol mark	0x00 0x63	2bytes
Data length	0x00 0x10 0x00 0x10	4bytes
Data	JSON string (Utf8)	Variable length (end in \0)

Return value JSON string (Utf8) description (JSON data format)：

```
{
        "sCissName": "Ciss0",
        "nCissStatus": 0,
        "fCapacity": 3000.000000,
        "fLeftPercent": 0.000000,
        "nLeftPrintNums": 0,
        "nMaxPrintNums": 0,
        "sInkModel": "P31X ",
        "nHeatOn": 1,
        "fTemperature": 35.000000,
        "fCurrrentTemperature": 20.000000,
        "nExtHeatOn": 0,
        "fExtTemperature": 0.000000,
        "fExtCurrrentTemperature": 0.000000
    }
```