

Central control linker communication protocol (RS485)

Communication setting : Baud rate : 4800bps; Check bit: NONE ; Data bit 8; Stop bit 1.

One frame length is fixed to 10 bytes, The 9th and 10th bytes are the CRC-16 checksum of the first 8 bytes.

The central controller send the data to the linkers, The data format is as follows

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
0x3A	Linker Address	0x01	0x00	0x00				CRC-16 LowByte	CRC-16 HighByte

The first byte is the constant 0x3A, which is also the start byte of each frame;

The second byte is the destination address, to control the linker number 01, the value is 0x01; the linker number 02, the value is 0x02

The third byte is the constant 0x01;

The 4th byte is the constant 0x00;

The 5th byte is the constant 0x00;

The sixth byte function is as follows:

Byte 6							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	Drain	Swing	Pump	Exhaust	Fan

The Clean function should be turn off after 3-minute's running.

The 7th byte function is as follows:

Byte 7							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Allow to fill water	0	0	0	0	0	0	0

The 8th byte function is as follows:

Byte 8							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	0-15 means cooler fan speed 1-16			

The lower 8 bits of the 9th byte CRC-16.

The higher 8 bits of the 10th byte CRC-16.

When the linker receives the command from the central controller, it will send a frame of data to the central controller in 0.2 seconds. The data format is as follows:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
0x3A	Linker Address	0x01			Received Byte6	Received Byte7	Received Byte8	CRC-16 LowByte	CRC-16 HighByte

Communication setting : Baud Rate: 4800bps; Check bit: NONE; Data bit 8; Stop bit 1.

One frame length is fixed to 10 bytes, The 9th and 10th bytes are the CRC-16 checksum of the first 8 bytes.

The first byte is the constant 0x3A, which is also the start byte of each frame;

The second byte is the network address. to control the linker number 01, the value is 0x01; the linker number 02, the value is 0x02

The 3rd byte is the constant 0x01;

The 4th byte function as bellows:

Byte 4							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Over current Error code E1	Power supply, voltage too high error code E2	Power supply voltage too low Error code E3	0	0	0	0	0

The 5th byte function is as bellows;

Byte 5							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	Water supply failure	Filling water	Higher water level sensor without water	Lower water level sensor without water

The sixth byte is the sixth byte of the last command received;

The seventh byte is the seventh byte of the last command received;

The eighth byte is the eighth byte of the last command received;

The 9th byte is the lower 8 bits of CRC-16 (pay attention).

The 10th byte is the higher 8 bits of CRC-16 (pay attention).

Command Example:

The central controller control Linker number 1 to

Turn on fan、 pump, Swing, Set fan speed at 16, Then need to send the following command:

0x3A,0x01,0x01,0x00,0x00,0x0D,0x80,0x0F,0x62,0x76

If the cooler lack of water at the moment, after received above command, the linker 01 will send the following command to the central controller :

0x3A,0x01,0x01,0x00,0x03,0x0D,0x80,0x0F,0x62,0x32

The central controller controls Linker number 2 to

Turn on the fan only, At speed 7, should send following command :

0x3A,0x01,0x01,0x00,0x00,0x01,0x80,0x06,0x62,0x73

If the cooler has sufficient water, after receive above command, The linker number 2 will send the following command to the central controller:

0x3A,0x01,0x01,0x00,0x00,0x01,0x80,0x06,0x62,0x73

CRC check program reference

// Entry parameters: Array first address, array length

```
unsigned int CRCApp(unsigned char * dat, unsigned char len)
```

```
{
```

```
    unsigned char i,j;
```

```
    unsigned int x = 0;
```

```
    unsigned int crc = 0xFFFF;
```

```
    for(i=0; i<len; i++)
```

```
    {
```

```
        crc = crc ^ dat[i];
```

```
        for (j=8; j>0; j--)
```

```
        {
```

```
            x = crc & 0x0001u;
```

```
            crc >>= 1;
```

```
            if (x!=0) crc ^= 0xA001;
```

```
    }  
  }  
  return crc;  
}
```