

# 12 Application Program Samples

---

In this chapter, we make some samples about pulse output instruction, Modbus communication instructions and free format communication instructions etc.

12-1 . Pulse Output Sample

12-2 . Modbus Communication Sample

12-3 . Free Format Communication Sample

## 12-1 . Pulse Output Application

Example: below is the example program to send high/low pulse in turn

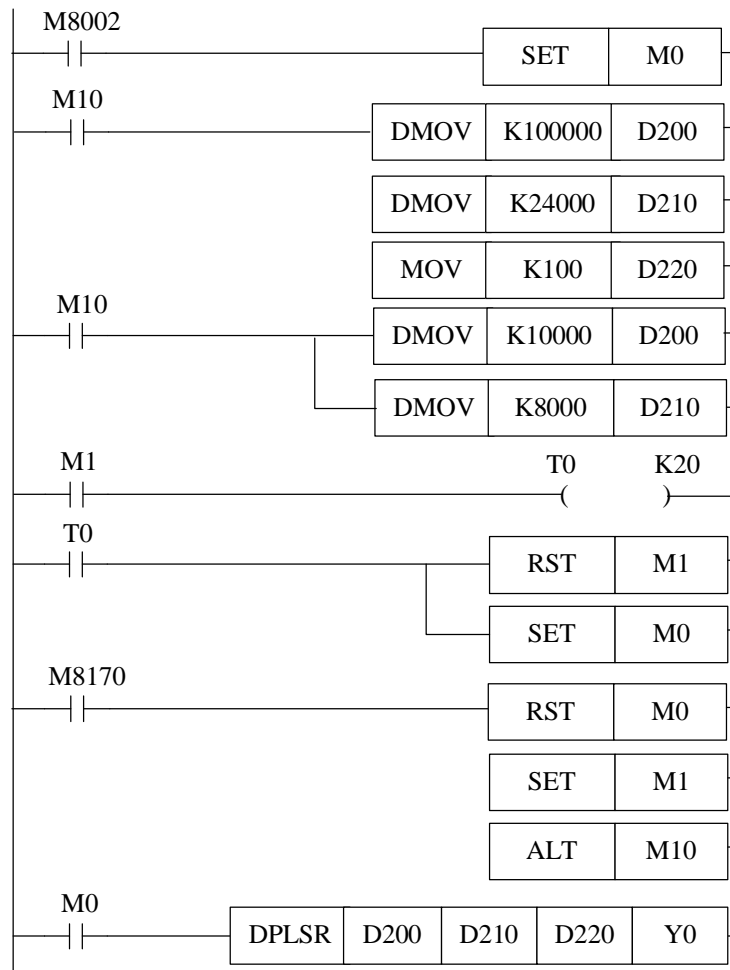
Each Parameter:

Stepping motor parameters: step angle= 1.8 degrees/step, scale=40, pulse number per rotate is 8000

High frequency pulse: maximum frequency is 100KHz, total pulse number is 24000 (3 rotates)

Low frequency pulse: maximum frequency is 10KHz, total pulse number is 8000 (1 rotates)

### Ladder Program:



### Instruction List:

LD	M8002	//initial positive pulse coil
SET	M0	//set M0 ON
LDF	M10	//M10 falling edge activate condition
OR	M8002	//Initial data
DMOV	K100000 D200	//move decimal data 100000 into DWORD D200
DMOV	K24000 D210	// move decimal data 24000 into DWORD D210
MOV	K100 D220	// move decimal data 100 into DWORD D220
LDP	M10	//M10 rising edge activate condition
DMOV	K10000 D200	// move decimal data 10000 into DWORD D200

DMOV	K8000	D210	// move decimal data 8000 into DWORD D210
LD	M1		//M1 status activate condition
OUT	T0	K20	//100ms timer T0, time 2 seconds
LD	T0		//T0 status activate condition
RST	M1		//reset M1
SET	M0		//set M0
LDF	M8170		//M8170 falling edge activate condition
RST	M0		//reset M0
SET	M1		//set M1
ALT	M10		//M10 status NOT
LD	M0		//M0 status activate condition
DPLSR	D200	D210 D220 Y0	//value in D200 is frequency、value in D210 is pulse number、 value is D220 is acceleration/deceleration time, send pulse via Y0;

#### **Explanation:**

When PLC changes from STOP to be RUN, M8002 gets a scan cycle; set the high frequency pulse parameters into D200、 D210, set the acceleration/deceleration speed to D220, set M0, the motor starts to run 3 rounds with high frequency. Meantime M8170 sets; the motor runs 3 rounds and decelerate, stop, coil M8170 reset; then reset M0, set M1, **NOT** M10; set the low frequency pulse parameters into D200、 D210; the timer time lags 2sec, when time reaches, reset M1; set M0, the motors starts to run 1 round with low frequency; after this starts to run with high frequency. Repeat this alternation time by time;

## 12-2 . MODBUS COMMUNICATION SAMPLES

E.g.1: realize Modbus read/write among one master and three slaves

**Operation:** (1) write content in D10~D14 to D10~D14 of 2# slave;  
(2) read D15~D19 of the slaves to D15~D19 of the mater; anyhow, write the first five registers' content to the slaves, the left five registers are used to store the content from the slaves;  
( 3 ) 3# 、 4# slaves are similar;

**Soft component's comments:**

D0: communication station number

D1: offset

M2: 2# communication error

M3: 3# communication error

M4: 4# communication error

M8137: COM2 communication error end signal

M8138: COM2 communication correct end signal

S0: write the target station

S1: read the target station

S2: judge the communication status

S3: offset the communication ID

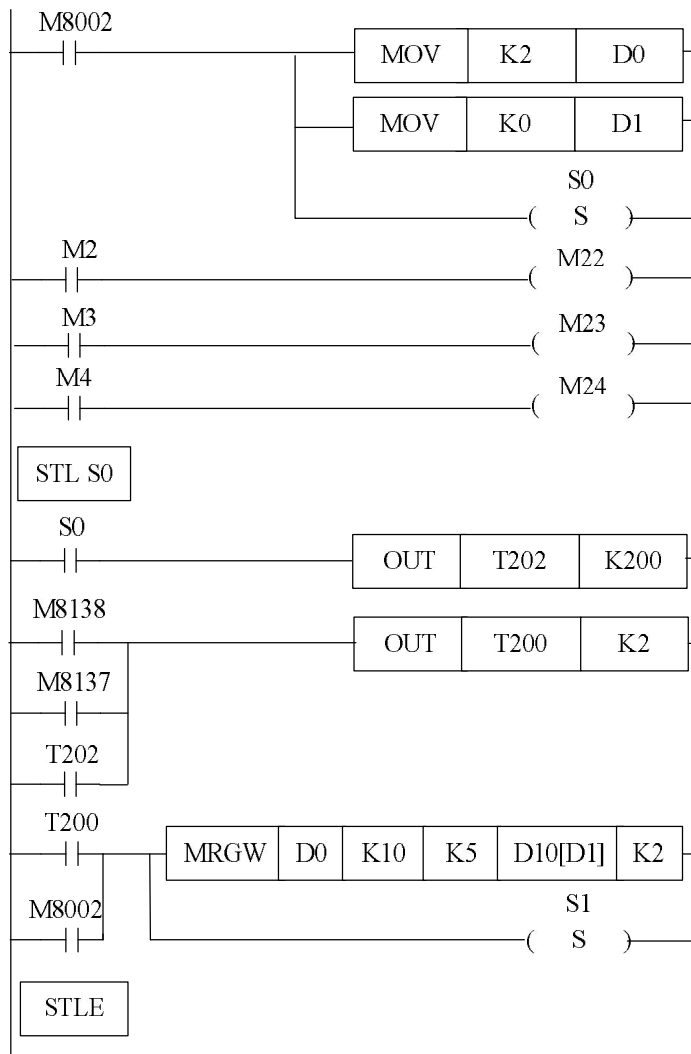
T200: communication interval 1

T201: communication interval 2

T202: self reset 1 of communication error

T203: self reset 2 of communication error

**Ladder**



In PLC's first scan cycle, evaluate the "communication station" to be 2;

Evaluate the "offset" to be 0

2# communication error reset

3# communication error reset

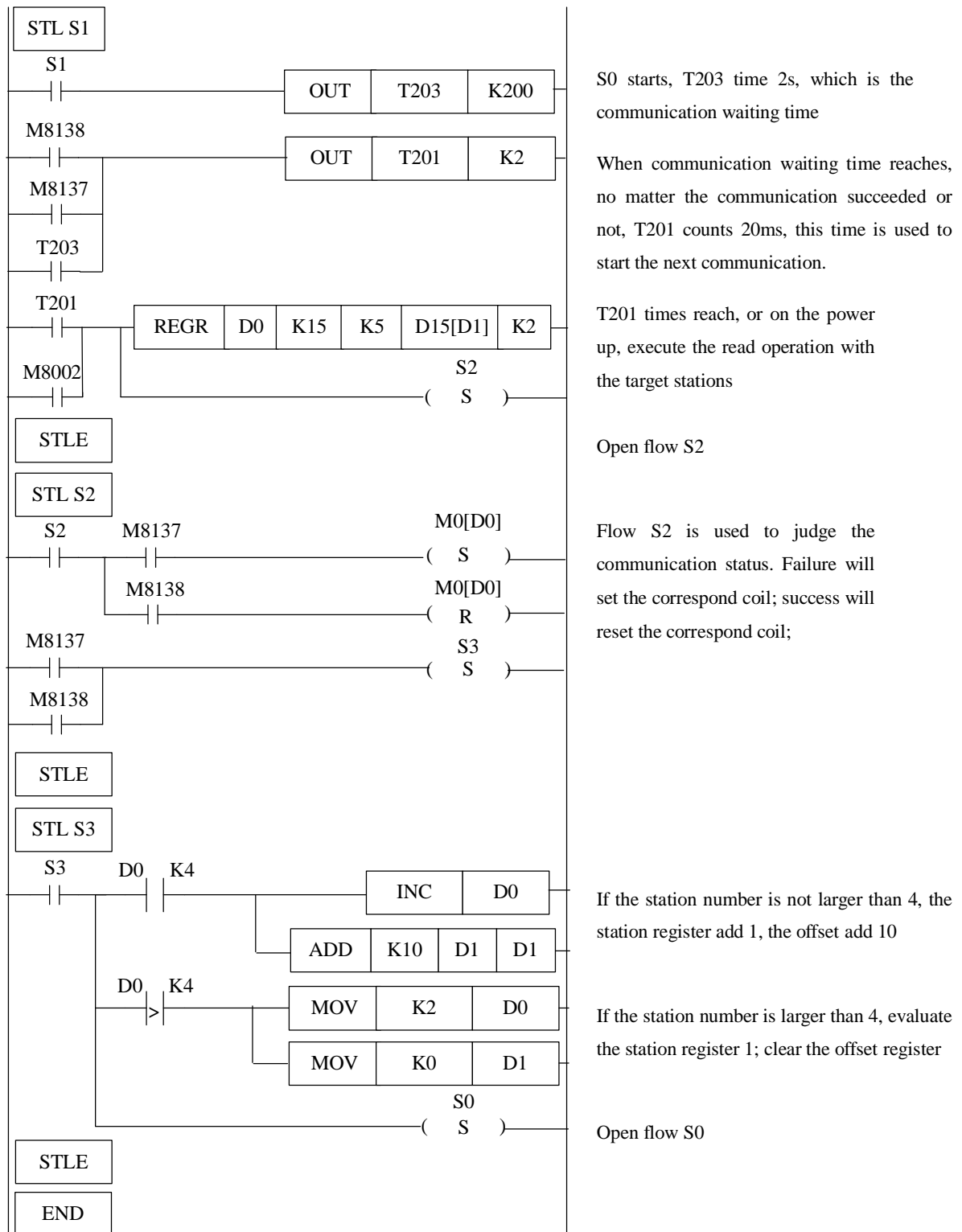
4# communication error reset

S0 starts, T202 counts 2S, which is the communication wait time

When the communication wait time reaches, no matter the communication succeeds or not, T200 time 20ms, this time is used start the next communication

T200 time reaches, or on the power up, execute the RUN operation to the target station

Open the flow S1



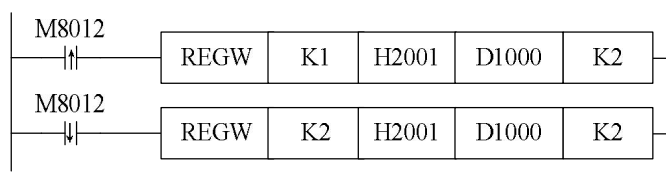
### Program Explanation:

When PLC turns from STOP to RUN, M8002 gets a scan cycle. S0 flow open, write the master's D10—D14 to slave 2# D10—D14. no matter the communication is success or not, turn to S1 flow; check the previous communication written condition. After certain time delay, continue to read D15~D19 data from 2#. After this reading enter S2 flow, check if the communication is success. If failed, set M23, enter alarming. After finishing the communication with 2#, enter S3,

then flow S3 will judge with the station number. If the station number is less than 1, the offset add 10; or else start from 2# again.

e.g. 2: Below is the sample of THINGET XC series PLC with two of THINGET inverters, they communicate via Modbus communication, XC series PLC write the frequency to the two inverters;

set the first inverter's station to be 1; set the second inverter's station to be 2; store the frequency's set value in D1000 and D2000. execute the frequency setting order via COM ports;



**Program Description:**

On the rising edge of M8012, write frequency to the first inverter; on the falling edge of M8012, write frequency to the second inverter;

## 12-3 . Free Format Communication Example

In this example, we use DH107/DH108 series instruments;

### 1、Interface Specifications

DH107/DH108 series instruments use asynchronous serial communication interface, the interface level fits RS232C or RS485 standard. The data format is: 1 start bit, 8 data bits, no parity, one/two stop bit. The baud rate can be 1200~19200bit/s .

### 2、Communication Instruction Format

DH107/108 instruments use Hex data form to represent each instruction code and data;

Read/write instructions:

**Read: address code +52H (82) +the para.(to read) code +0+0+CRC parity code**

**Write: address code +43H( 67 )+ the para.(to write) code +low bytes of the wrote data + high bytes of the wrote data +CRC parity code**

**The read instruction's CRC parity code is: the para. (to read) code \*256+82+ADDR**

**ADDR** is instrument's address para., the range is 0~100 (pay attention not to add 80H). CRC is the remainder from the addition of the above data (binary 16bits integral). The remainder is 2 bytes, the high byte is behind the low byte;

**The write instruction's CRC parity code is: the para. (to write) code \*256+67+ the para. value (to write) +ADDR**

The para. to write represents with 16 bits binary integral;

No matter to write or read, the instrument should return data as shown below:

**The test value PV+ given value SV+ output value MV and alarm status +read/write parameters value +CRC parity code**

Among in, PV、SV and the read parameters are all in integral form, each occupies two bytes, MV occupies one byte, the value range is 0~220, alarm status occupies one byte, CRC parity code occupies two bytes, totally 10 bytes.

**CRC parity code is the reminder from the result of PV+SV+ (alarm status \*256+MV)+ para. value +ADDR;**

(for details, please refer to AIBUS communication description)

### 3、Write the program

After power on the PLC, the PLC read the current temperature every 40ms. During this period, the user can write the set temperature.

Data zone definition: buffer area of sending data D10~D19

buffer area of accepting data D20~D29

instruction's station number: D30

read command's value: D31=52 H

write command's value: D32=43 H

parameter's code: D33

temperature setting: D34



CRC parity code: D36

Temperature display: D200,D201

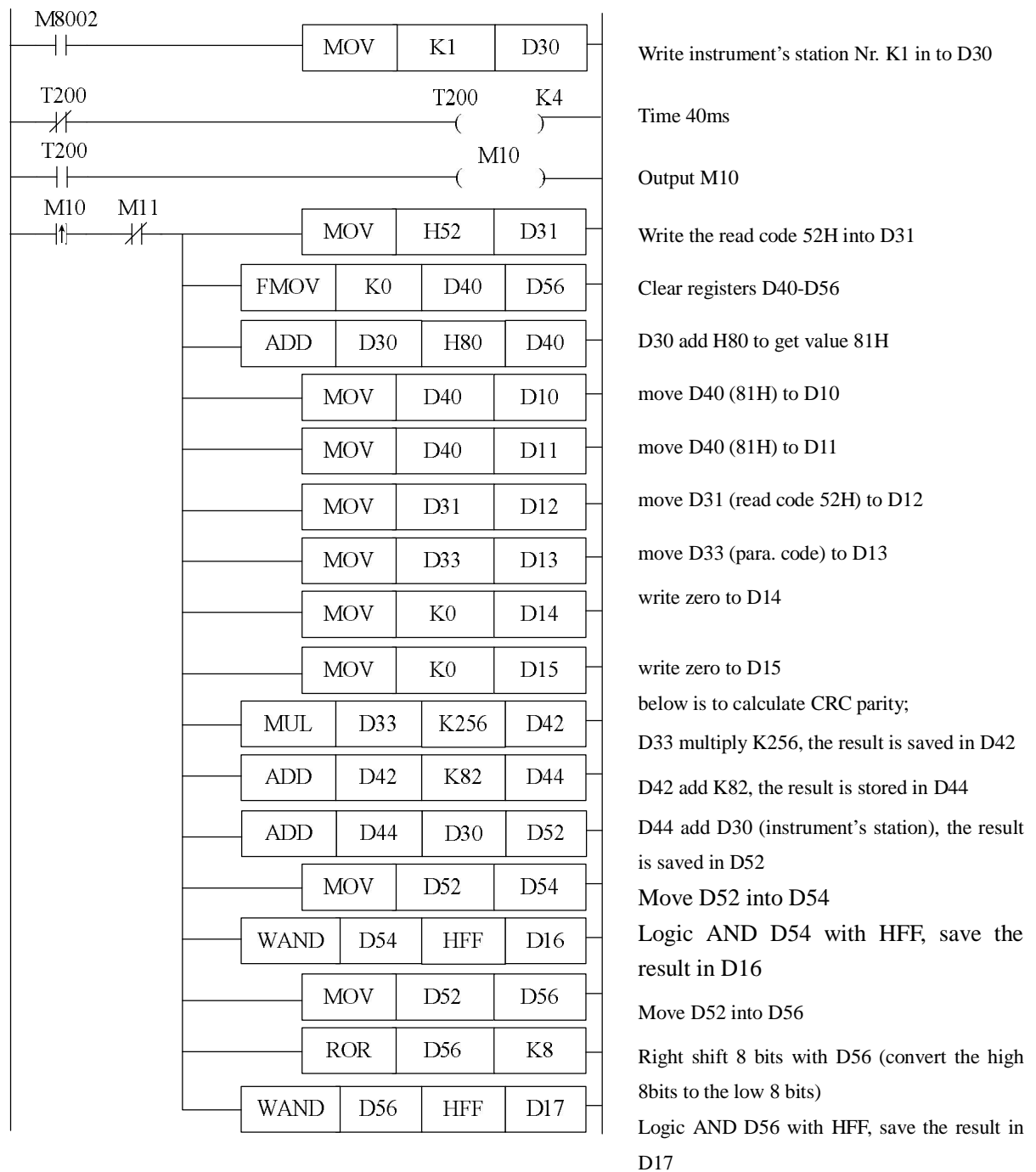
The send data form: 81H 81H 43H 00H c8H 00H 0cH 01H (current temperature display)

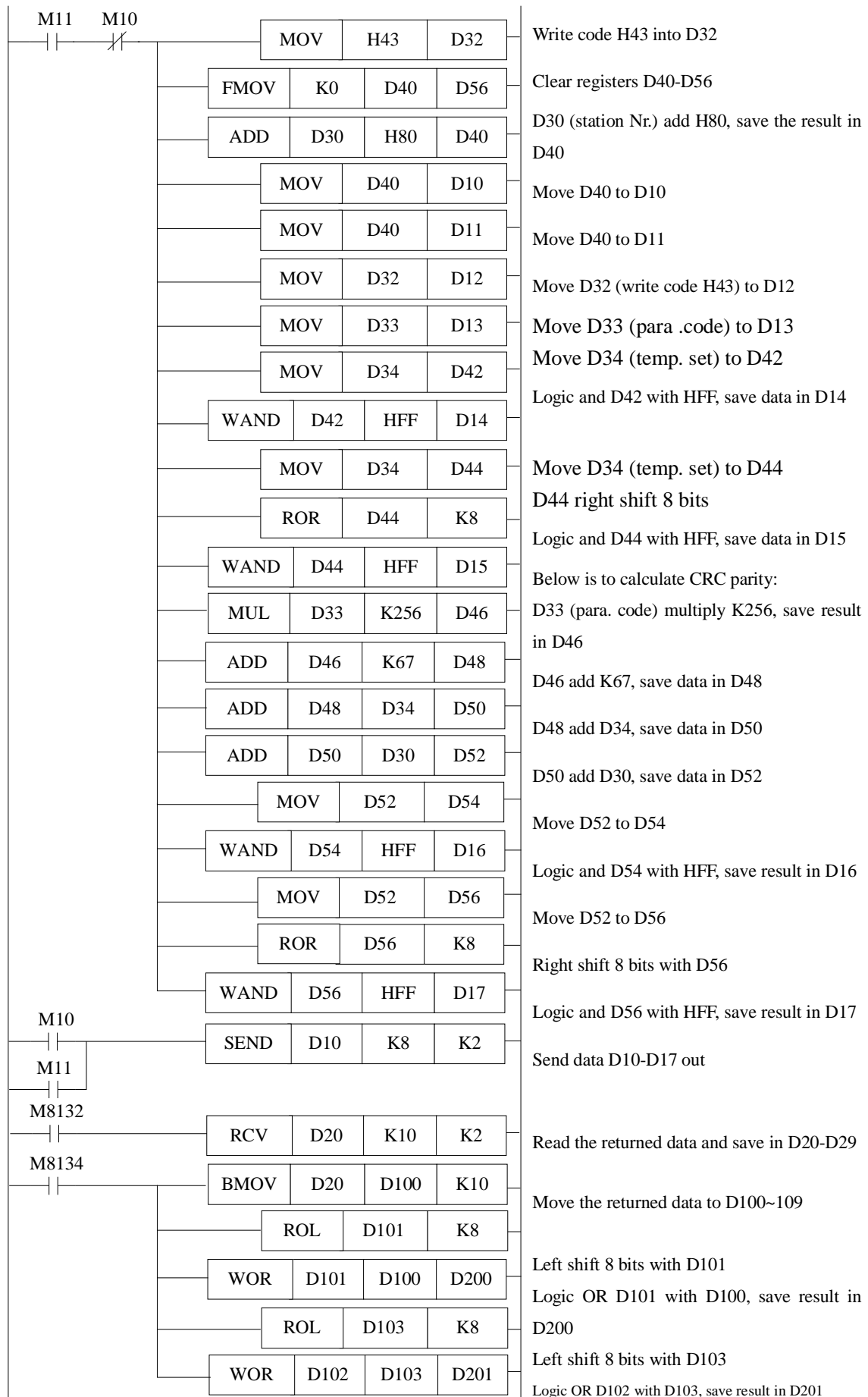
Communication parameters setting: baud rate: 9600, 8 data bits, 2 stop bits, no parity

Set FD8220=255; FD8221=5

( the hardware and software must be V2.4 or above)

## Ladder:





**Program Description:**

The above program is written according to DH instrument's communication protocol, the soft component's functions are listed below:

Relationship of sent (SEND) data string and registers:

	D10	D11	D12	D13	D14	D15	D16	D17
Read	Address code	Address code	Read code 52H	Parameters code	0	0	CRC low bytes	CRC high bytes
Write	Address code	Address code	Write code 42H	Parameters code	low bytes of the written data	high bytes of the written data	CRC low bytes	CRC high bytes

Relationship of received (RCV) data (data returned by the instrument) and the registers:

D20	D21	D22	D23	D24	D25	D26	D27	D28	D29
PV low bytes	PV high bytes	SV low bytes	SV high bytes	Output value	Alarm status	Read/write low bytes	Read/write high bytes	CRC low bytes	CRC high bytes

So, if write data string according to the communication objects' protocol, use SEND and RCV commands from free format communication, user will get the communication with the objects.

