Product Description

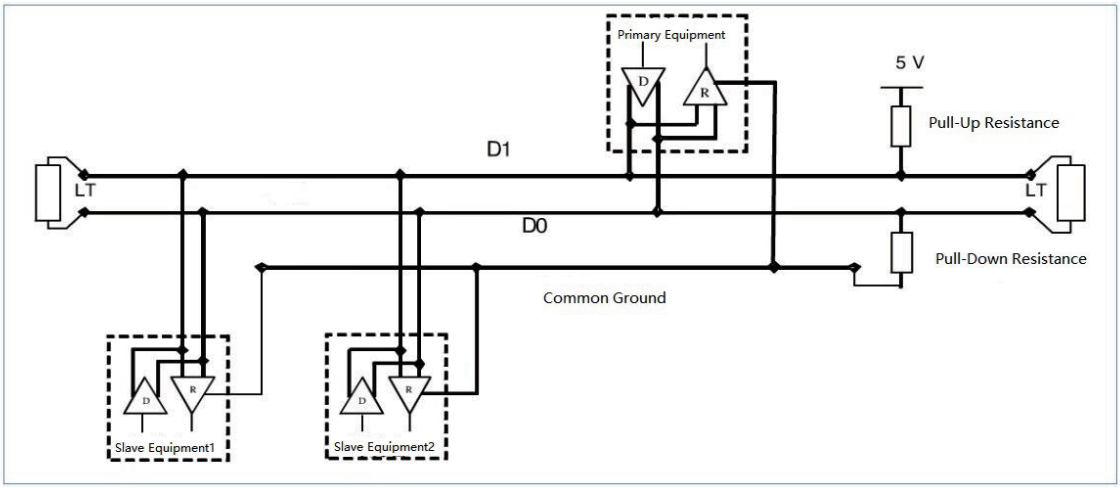
DRFA12S series is a multi-channel bus control voltage regulator. It adopts RS 485 bus control with Modbus RTU communication protocol. Designed to work with random-on solid-state relays (SSR), it can realize the voltage regulation function and can independently control 12 channels of SSRs, with channel 11 and 12 specifically available for cooling fan control. The module supports multiple types of thermocouples. The address can be configured from 1 to 8. Various baud rates are available for the 485 communication, with default settings at 9600 baud, no parity, and 2 stop bits.

Setting Introduction

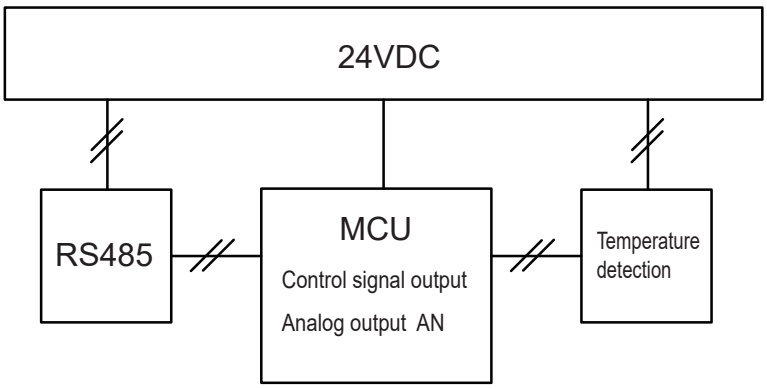
Please follow the wiring diagram in the datasheet for wiring, turn the address encoder to the required address, when the power is on then it can work normally.If you need to change the address, you need to power off first, and then turn the address encoder to the required address, when the power is on it can work normally.Note that the address cannot be set to 0.

Address	Fixed Communication Parameters
9	9600, even parity check, 1 stop bit

Topology Structure of Two-line Modbus



Isolation Mode Diagram



“ // ”: represents isolation

Register Introduction

DRFA12S has 6 input register and 26 holding register, their address and definition are as follows:

Address	Definition	Whether to save when power is off
	Input Register	
0	Working Status Register REG_STATUS	No
7	Temperature Register of T1 thermocouple REG_TEMP_T1	No
8	Temperature Register of T2 thermocouple REG_TEMP_T2	No
9	Temperature Register of T3 thermocouple REG_TEMP_T3	No
10	Thermocouple Break Register REG_T_ERR	No
11	AN Real-time Status Register REG_AN	No
	Holding Register	
20	U1 Phase Control Register REG_CON_U1	Yes
21	U2 Phase Control Register REG_CON_U2	Yes
22	U3 Phase Control Register REG_CON_U3	Yes
23	U4 Phase Control Register REG_CON_U4	Yes
24	V1 Phase Control Register REG_CON_V1	Yes
25	V2 Phase Control Register REG_CON_V2	Yes
26	V3 Phase Control Register REG_CON_V3	Yes
27	V4 Phase Control Register REG_CON_V4	Yes
28	W1 Phase Control Register REG_CON_W1	Yes
29	W2 Phase Control Register REG_CON_W2	Yes
30	W3 Phase Control Register REG_CON_W3	Yes
31	W4 Phase Control Register REG_CON_W4	Yes
32	Communication Fault Turn-off Register REG_OFF	Yes
33	Communication Parameter Setting Register REG_TX	Yes
34	Soft Start Register REG_SOFT_START	Yes
35	On-off Register REG_ON_OFF	Yes
36	Target Temperature 1 Setting Register REG_T1_SET	Yes
37	Temperature Range Setting Register REG_RANGE	Yes
38	Thermocouple Type Register REG_T_TYPE	Yes
39	Output Voltage Register REG_VOLTAGE	Yes
41	Motor Power Lower Limit Register REG_POWER_MIN	Yes
42	Motor Power Upper Limit Register REG_POWER_MAX	Yes
43	Gain Register REG_GAIN	Yes
44	Analog Control Register REG_CON_AN	No
45	W4 Function Selection Register REG_W4_SELEC	Yes
46	Target Temperature 2 Setting Register REG_T2_SET	Yes

The definition and usage description of input registers are as follows:

(1) Working Status Register REG_STATUS（Add:0）

Data format: 16 bit unsigned integer

This register is used to indicate the corresponding working status, as follows(1):

Bit5: When this bit is 1, it indicates that the U-phase of DRFA12S is disconnected or has no voltage; When this bit is equal to 0, it indicates that the wiring is normal and there is voltage.

Bit6: When this bit is 1, it indicates that the V phase of DRFA12S is disconnected or has no voltage; When this bit is equal to 0, it indicates that the wiring is normal and there is voltage.

Bit7: When this bit is 1, it indicates that the W-phase of DRFA12S is disconnected or has no voltage; When this bit is equal to 0, it indicates that the wiring is normal and there is voltage.

Note: (1) For 380T products, when one of the three wires of the three-phase power supply is disconnected, the corresponding position is 1. When two or three wires are disconnected at the same time, all three positions are set to 1.

(2) Temperature Register thermocouple REG_TEMP_TX（Add:7~9）
Data format: 16 bit unsigned integer
This register is the real-time temperature of the thermocouple, the unit is 0.1 ℃. When the thermocouple breaks, its value is 20479.

(3) Thermocouple Break Register REG_T_ERR（Add:10）
Data format: 16 bit unsigned integer
When the value of the register is zero, it indicates that the thermocouple is normal. In case of abnormality, it is as follows:
Bit0: When this bit is 1, it indicates that thermocouple T1 is disconnected;
Bit1: When this bit is 1, it indicates that thermocouple T2 is disconnected;
Bit2: When this bit is 1, it indicates that thermocouple T3 is disconnected.

(4) AN Real-time Status Register REG_AN（Add:11）
Data format: 16 bit unsigned integer
This register displays the control value of AN in real-time.

The definition and usage description of holding registers are as follows:

(1) Phase Control Register REG_CON_xy（Add:20~31）
Data format: 16 bit unsigned integer
This register is used to control the output power value of each solid-state relay. This value ranges from 0 to 1000 and is used to set the size of the output power.
For example, if the register value is 500 and the output voltage is set to 220VAC, then the output power of this channel is 50% of the power of the load at 220VAC.

(2) Communication Fault Turn-off Register REG_OFF（Add:32）
This register is used to set the time of automatically shutting down the output when communication is interrupted, the unit is seconds. This value ranges from 0 to 100, 0 indicates that the function is turned off, and the default value is 0. Other values indicate that when the controller does not receive communication information with the local address within that time, the output will be turned off.
For example, if the register value is 8, the controller will turn off the output if it does not receive communication information with its own address within 8 seconds.

(3) Communication Parameter Setting Register REG_TX（Add:33）
Data format: 16 bit unsigned integer
This register is used to set communication parameters, as shown in the table below:

Register value	Baud rate	Data bit	Check bit	Stop bit
300	9600	8 Bit data bits	No check	2 stop bits
301	19200	8 Bit data bits	No check	2 stop bits
302	38400	8 Bit data bits	No check	2 stop bits
303	57600	8 Bit data bits	No check	2 stop bits
304	115200	8 Bit data bits	No check	2 stop bits
310	9600	8 Bit data bits	even parity check	1 stop bit
311	19200	8 Bit data bits	even parity check	1 stop bit
312	38400	8 Bit data bits	even parity check	1 stop bit
313	57600	8 Bit data bits	even parity check	1 stop bit
314	115200	8 Bit data bits	even parity check	1 stop bit
320	9600	8 Bit data bits	odd parity check	1 stop bit
321	19200	8 Bit data bits	odd parity check	1 stop bit
322	38400	8 Bit data bits	odd parity check	1 stop bit
323	57600	8 Bit data bits	odd parity check	1 stop bit
324	115200	8 Bit data bits	odd parity check	1 stop bit

The default value of this register is 300. To change the communication parameters, such as 9600, even check, and 1 stop bit, simply write 310 into this register and then power it on again.

Register Introduction

(4) Soft Start Register REG_SOFT_START（Add:34）

Data format: 16 bit unsigned integer

This register is used to control the slope of soft start, with a value range of 0-60 and a default value of 0. When the value is 0, turn off the soft start function. When the value is n, the output control increases at a rate of 8/(n × 50mS).

For example, when the output control register REG_CON_xy has a value of 500 (i.e. output power of 50%) and the soft start register REG_SOFT-START has a value of 10, the output increases at a rate of 8/(10 × 50mS). After about 32 seconds, the output reaches the set value of 500 (i.e. output power of 50%).

(5) On-off Register REG_ON_OFF（Add:35）

Data format: 16 bit unsigned integer

This register is used to control the output of the product, with a value range of 0~8191 (i.e. 0~0x1FFF). The default value is 0 and it is not saved in case of power failure. The definitions of each position are as follows:

Bit0: Control U1 channel. When the bit is 0, turn off the output of U1 channel, otherwise turn it on.
Bit1: Control U2 channel. When the bit is 0, turn off the output of U2 channel, otherwise turn it on.
Bit2: Control U3 channel. When the bit is 0, turn off the output of U3 channel, otherwise turn it on.
Bit3: Control U4 channel. When the bit is 0, turn off the output of U4 channel, otherwise turn it on.
Bit4: Control V1 channel. When the bit is 0, turn off the output of V1 channel, and vice versa.
Bit5: Control the V2 channel. When the bit is 0, turn off the V2 output, and vice versa.
Bit6: Control V3 channel. When the bit is 0, turn off the output of V3 channel, otherwise turn it on.
Bit7: Control the V4 channel. When the bit is 0, turn off the V4 output, and vice versa.
Bit8: Control W1 channel. When the bit is 0, turn off the output of W1 channel, otherwise turn it on.
Bit9: Control W2 channel. When the bit is 0, turn off the output of W2 channel, otherwise turn it on.
Bit10: Control W3 channel. When the bit is 0, turn off W3 channel output, otherwise turn it on.
Bit11: Control the W4 (F+/F -) channel. When the bit is 0, turn off the output of W4 (F+/F -) channel, and vice versa.
Bit12: Control the analog quantity AN. When the bit is 0, turn off the AN output, and vice versa.

(6) Target Temperature 1 Setting Register REG_T1_SET（Add:36）

This register is used to set the target temperature, with a unit of 0.1 ℃ and a value range of 0~10000. The module compares the detected temperature with the target temperature and adjusts the output power of the cooling fan based on the comparison result. The priority of this register is higher than that of the analog control register REG_CON_AN (address: 44). When this register is non-zero, AN is controlled by the temperature calculation result.

When the register is zero, AN is controlled by the analog control register REG_CON_AN (address: 44). When the W4 function selection register REG_W4_SELEC is set to 1, the control value of W4 is consistent with AN.

(7) Temperature Range Setting Register REG_RANGE（Add:37）

Data format: 16 bit unsigned integer

This register is used to set the temperature control range, with a unit of 0.1 ℃ and a value range of 20~1000. For example, the target temperature 1 setting register REG_T1_SET is set to 1000, and the temperature range setting register REG_RANGE is set to 100. When the detected temperature is below the lower limit (REG_T1_SET-REG_RANGE/2), which is 95 ℃, the output power is zero; When the detected temperature is greater than the upper limit (REG_T1_SET+REG-Range/2), which is 105 ℃, the output power is 100%; When the detected temperature is between the upper and lower limits, the output power is linearly related to the temperature.

(8) Thermocouple Type Register REG_T_TYPE（Add:38）

This register is used to set the thermocouple type, as shown in the table below:

Register value	Thermocouple type
0	K
1	J
2	T
3	N
4	S
5	E
6	B
7	R

The default value of this register is 0, which means it defaults to K-type thermocouple. When changing other values, power must be turned off and then back on before the new type can be executed.

Attention: Isolation type thermocouples are preferred. If the number of thermocouples exceeds one, all isolated thermocouples need to be used.

(9) Output Voltage Register REG_VOLTAGE（Add:39）

Data format: 16 bit unsigned integer

This register is only applicable to the 380T model and is used to set the output voltage size. The default value of this register is 380. The output voltage regulation of the product will be adjusted according to this set value.

For example, when the register value is 380, the highest output voltage is 380VAC.

When the register value is 220, the highest output voltage is 220VAC.

(10) Motor Power Lower Limit Register REG_POWER_MIN（Add:41）

Data format: 16 bit unsigned integer

The value range of this register is 0-400, used to set the lower limit of motor power regulation, with a default value of 0.

For example, when the value of this register is 100, it indicates that the lower limit of motor regulation is 100.

(11) Motor Power Upper Limit Register REG_POWER_MAX（Add:42）

Data format: 16 bit unsigned integer

The value range of this register is 400~1000, used to set the upper limit during the motor power regulation process, with a default value of 1000.

For example, when the value of this register is 800, it indicates that the upper limit of motor regulation is 800.

(12) Gain Register REG_GAIN（Add:43）

Data format: 16 bit unsigned integer

The value range of this register is 10-100, with a default value of 10. One tenth of this value multiplied by the control value is used as the output of this channel.

For example, when the value of the register is 40, if the first control register REG_CON_U1 of the U-phase is 250, then the output of the first U-phase output is 250 × 40/10=1000, which means that this channel is a full power output.

(13) Analog Control Register REG_CON_AN（Add:44）

Data format: 16 bit unsigned integer

This register is used to control the output voltage of analog signals. The value range of this register is 0~1000, corresponding to an output voltage of 0~10V.

For example, when the register value is 500, the analog AN output voltage is 5V.

(14) W4 Function Selection Register REG_W4_SELEC（Add:45）

Data format: 16 bit unsigned integer

This register is used to switch the output function of W4. The value range of this register is 0~1, and the default value is zero. When the register is zero, W4 is controlled by REG_CON_W4, consistent with other control methods. When the register is set to 1 and the target temperature register is non-zero, the output power of W4 is controlled by DRFA based on real-time temperature. At this time, the load (motor) power supply should be 220VAC from W and N.

(15) Target Temperature 2 Setting Register REG_T2_SET（Add:46）

Data format: 16 bit unsigned integer

This register is used to set the target temperature of 2, with a unit of 0.1 ℃ and a value range of 0~10000. The module compares the detected temperature T2 with the target temperature 2, and adjusts the output power of the cooling fan based on the comparison result. The priority of this register is higher than that of the W3 phase control register REG_CON_W3 (address: 30). When this register is zero, the output power of W3 is controlled by REG_CON_W3, consistent with other control methods. When the register is non-zero, the output power of W3 is controlled by DRFA based on real-time temperature. At this time, the load (motor) power supply should be 220VAC and come from W and N.

Communication Protocol

communication protocol uses the standard Modbus. Please refer to the official Modbus instructions for the relevant communication protocol. This document lists some formats for your reference:

0x03 - Read holding register
Request: Slave station address-0x03-data byte number-Data-CRCL-CRCH
Answer: Slave station address -0x03-data byte number-data-CRCL-CRCH
Error: Slave station address-0x83-Error Code-CRCL-CRCH
Example：Read the data from U1 and U2 control registers with address 05 of the slave station:
Request：0x05 0x03 0x00 0x14 0x00 0x02 CRCL CRCH
Answer：0x05 0x03 0x04 REG_CON1_H REG_CON1_L REG_CON2_H REG_CON2_L CRCL CRCH

0x06 - Write single register
Request: Slave station address -0x06-start address-registerquantities-CRCL-CRCH
Answer: Slave address -0x06- Register address - Value -CRCL-CRCH
Error: Slave address -0x86- Error code -CRCL-CRCH
Example：Write 80 to the U1 control register REG_CON_U1 with slave address 05:
Request：0x05 0x06 0x00 0x14 0x00 0x80 CRCL CRCH
Answer：0x05 0x06 0x00 0x14 0x00 0x80 CRCL CRCH

CRC Check

CRC check complies with Modbus,the following is a C language CRC check function:

The function uses two parameters:
unsigned char *puchMsg; Pointer to the buffer containing binary data messages for generation;
unsigned short usDstsLen; The number of bytes in the message buffer.

CRC generating function:

```
unsigned short CRC16(puchMsg,usDatalen)          /*function returns CRC as an unsigned short type*/
unsigned char *puchMsg;                          /*Message used to calculate CRC */
unsigned short usDataLen;                        /*The number of bytes in the message*/
{
      unsigned char uchCRCHI=0xFF;               /*High byte initialization of CRC*/
      unsigned char uchCRCLo=0xFF;               /*Low byte initialization of CRC*/
      unsigned uIndex;                           /*CRC query table index */
      while (usDataLen--)                        /*completes the entire message buffer */
           {
            uIndex=uchCRCLo^*puchMsgg++;         /*Calculate CRC */
            uchCRCLo=uchCRCHi^auchCRCHi[uIndex];
            uchCRCHi=uchCRCLo[uIndex];
           }
     return(uchCRCHi<<8|uchCRCLo);
}
```


CRC Check

High-Order Byte Table
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
};

Low-Order Byte Table
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
};

