Chapter 15 MemoryMap Communication

MemoryMap communication protocol is similar to IBM 3764R, it is used when memory data with low variation. High variation will cause MemoryMap overloading. MemoryMap is for two units communication. When setting the MemoryMap with two devices, one has to be set as Master, and the other is slave. In normal condition, Master and Slave do not communication except when one of them has change memory data which has assign for each other. After data is identical the communication will stop communicating. So this is used for to keep the consistent data between two device (Master and Slave) via corresponding register.

The corresponding memory is the same propoty as MT500's register MW(MB) from Master and slave (The 1000 words MW(MB) is reserved for MemoryMap in MT500 for communication.) The feature of memory: MB is correspondence with MW, according following list, MB0~MBf and MW0, MB10~MB1f and MW1…, they are all indicate the same register.

Device name	Format	Range
MB	ddd(h)	ddd:0~999 h:0~F(hex)
MW	ddd	ddd:0~999

When using MemoryMap communication procotol, the master and slave have to use the same communication setting. The way to connect as doing follow:

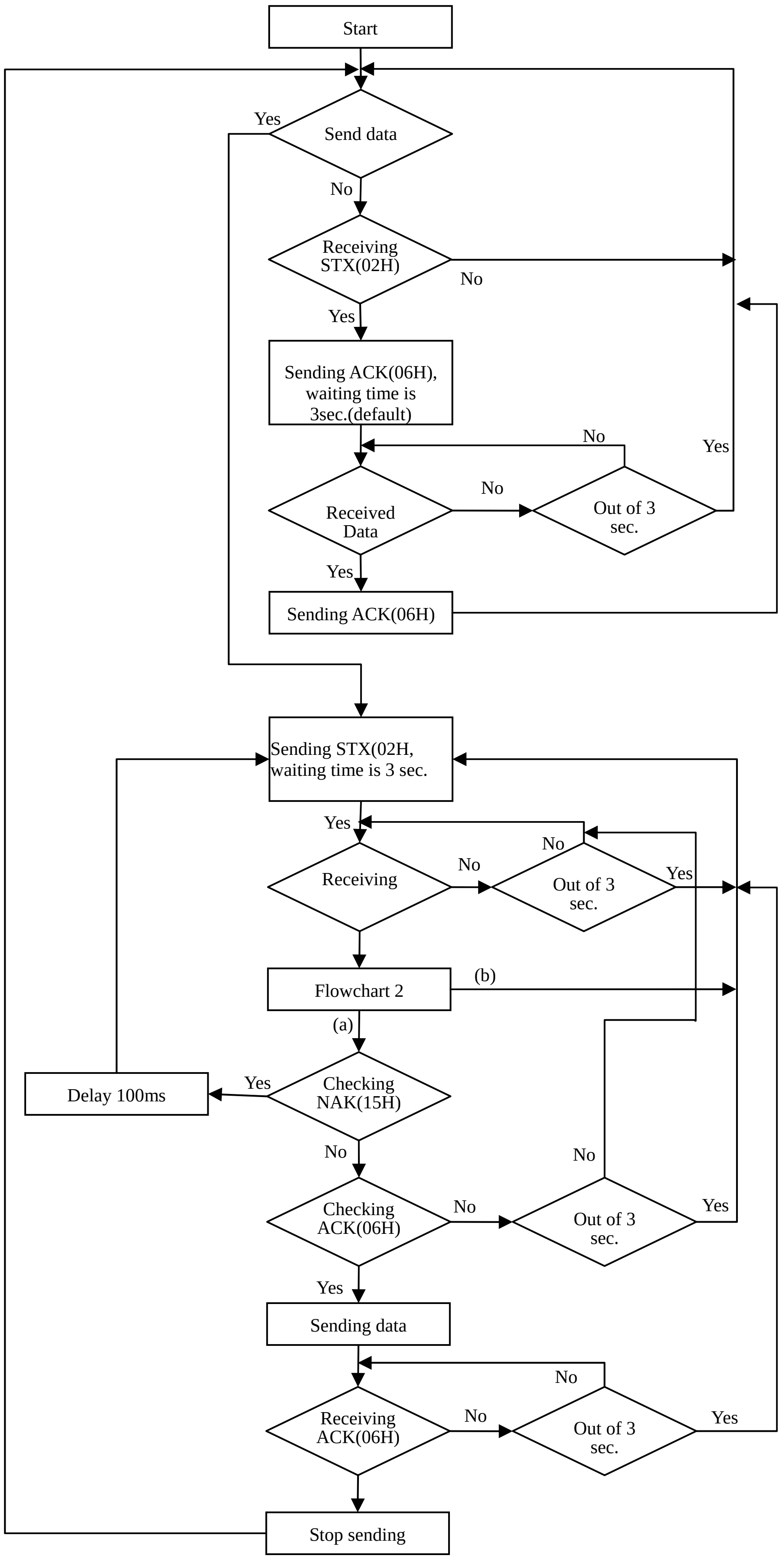
RS232	
Master	Slave
TX(#)	RX(#)
RX(#)	TX(#)
GND(#)	GND(#)

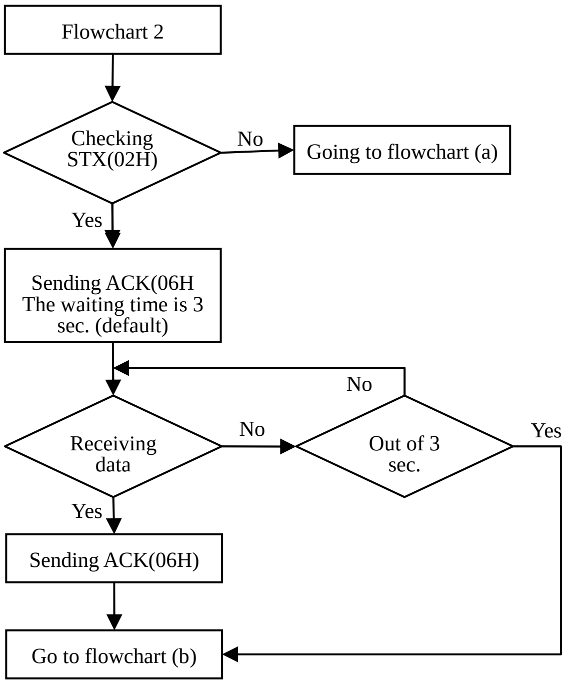
RS485(4W)	
Master	Slaver
TX+(#)	RX+(#)
TX-(#)	RX-(#)
RX+(#)	TX+(#)
RX-(#)	TX-(#)
GND(#)	GND(#)

Precaution: # means decide by PLC or controller.

The flowchart of communication as following:

Flowchart 1





Precaution:

Flowchart 2 is available for slave but master, STX is asking signal for communication, ACK is feedback signal, NAK is busy signal.

There are two data formats, one is for MB and the other is for MW:

For MB command		
Off set (byte)	Format	Description
0	0x02	The operating sign to MB
1 2	0x## 0x##	Address (Low byte) Bit Address (High byte) For example:MB12=>1*16+2=18, is 0x12 and 0x00
3	0x00(or 0x01)	The data of MB address. (This is Bit, so only has 0 or 1)
4，5	0x10，0x03	Stop sign
6	0x##	checksum ,xor from 0 byte to fifth byte.

For MW command		
Offset(byte)	Format	Description
0	0x01	The operating sign to MW
1 2	0x## 0x##	Address (Low byte) Bit Address (High byte) If there is a 0x10 include in address, and insert a 0x10, the byte will move to next position. For

		example: 0x10, 0x04 will become 0x10,0x10,0x04
3	0x##	Sending byte (The byte has to be even, duo to operating for word). If byte is 0x10 to insert a 0x10, the byte will move to next position
4~4+n-1	0x##(L) 0x##(H) 0x##(L)…	The data of initial address for corresponding address for 1,2 byte, n is byte of data, if data include 0x10 and then insert a 0x10, the sending byte number without change, n=n+1, and so on…
4+n，4+n+1	0x10，0x03	End sign
4+n+2	0x##	checksum，Xor check-up and bytes in the front

Below is a example for observation process of communication. If Master has a 0x0a in MW3,

According to this protocol, master will communicate with slave immediately, and slave will instead the 0x0a in MW3, the procedure as following:

Master sending STX(0x02h).

Slave receive STX(0x02h) from master, and sending a ACK(0x06h) to master.

Master received ACK(0x06h) from slave

Master sending 0x01,0x03,0x00,0x02,0x0a,0x00,0x10,0x03,0x19, as below table:

Offset(byte)	Format	Description
0	0x01	The operating sign for MW
1 2	0x03 0x00	Address(Low byte) Bit Address (High byte)
3	0x02	Sending byte (The byte has to be even, duo to MW3 is two byte).
4，5	0x0a，0x00	MW3's content is 0x0a，0x00
6，7	0x10，0x03	End sign
8	0x19	checksum，0x01^0x03^0x00^0x02^0x0a^0x00^0x10^0x03=0x19

Slave received data from master and then sending ACK(0x06h).

Master receive ACK(0x06h) from slave.

When finishing communication, master sending revised data of MW to slave, and slave change the MW which the same as master. At this time, master and slave keep the same data in the
same address.

Another example below, the address and data include 0x10, please notice the change from data format. Now, if we have 0x10 in MW16 in slave, according to this protocol, slave will communicate with master immediately, and master will instead the 0x10 in data of MW16, the procedure as following:

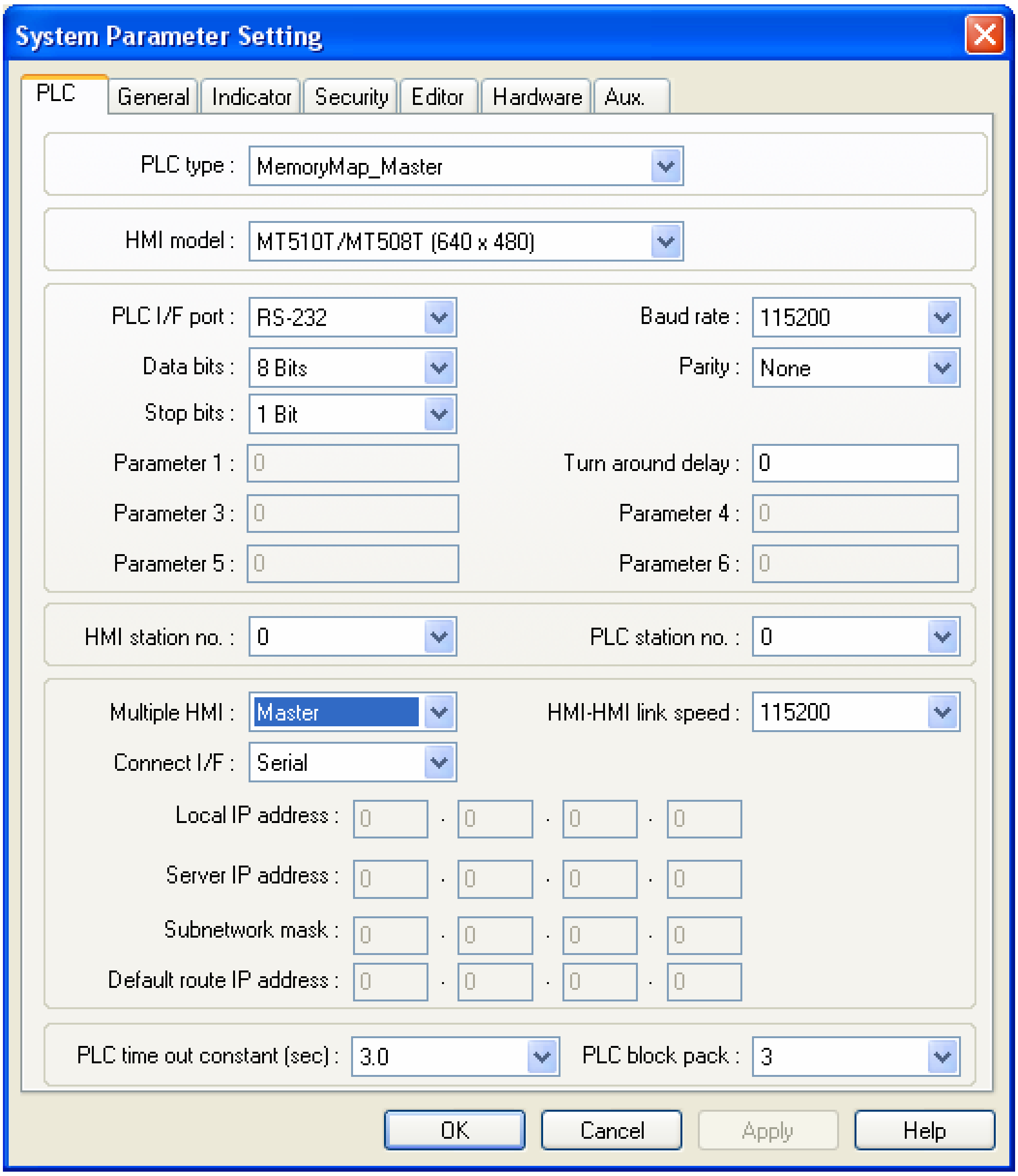
Slave sending STX(0x02h)
Master receive STX(0x02h) from slave, and sending ACK(0x06h) to master
Slave receive ACK(0x06h) from master
Slave sending data 0x01,0x10,0x10,0x00,0x02,0x10,0x10,0x00,0x10,0x03,0x10 as below table:

Offset (byte)	Format	Description
0	0x01	The operating sign to MW
1 2 3	0x10 0x10 0x00	Address(Low byte) Insert 0x10 Bit Address (High byte)
4	0x02	Sending byte (MW10 is two bytes)
5 6 7	0x10 0x10 0x00	0x10 is low byte in MW10 Insert 0x10 0x00 in high byte
8，9	0x10 , 0x03	End sign
10	0x10	checksum , 0x01^0x10^0x10^0x00^0x02^0x10^0x10^0x00^0x10^0x03=0x10

Master receive data from slave and sending ACK(0x06h) to slave.
Slave receive ACK(0x06h) from master.

When finishing communication, slave sending the address and content of MW to master, at this time, master change data of MW for the same as slave, master and slave keep the same data in the same address.

Below is a example for communicating between two HMI via MemoryMap.
First of all, create a new project in EasyBuilder
Edit/System Parameter Setting/PLC



PLC type : MemoryMap_Master

HMI model : MT510T/MT508T (640 x 480)

PLC I/F port : RS-232

Baud rate : 115200

Data bits : 8 Bits

Parity : None

Stop bits : 1 Bit

Parameter 1 : 0

Turn around delay : 0

Parameter 3 : 0

Parameter 4 : 0

Parameter 5 : 0

Parameter 6 : 0

HMI station no. : 0

PLC station no. : 0

Multiple HMI : Master

HMI-HMI link speed : 115200

Connect I/F : Serial

Local IP address : 0 . 0 . 0 . 0

Server IP address : 0 . 0 . 0 . 0

Subnetwork mask : 0 . 0 . 0 . 0

Default route IP address : 0 . 0 . 0 . 0

PLC time out constant (sec) : 3.0

PLC block pack : 3

Precaution:

1. For those two HMI, one is setting Memory_Master and the other is setting Memory_Slave.

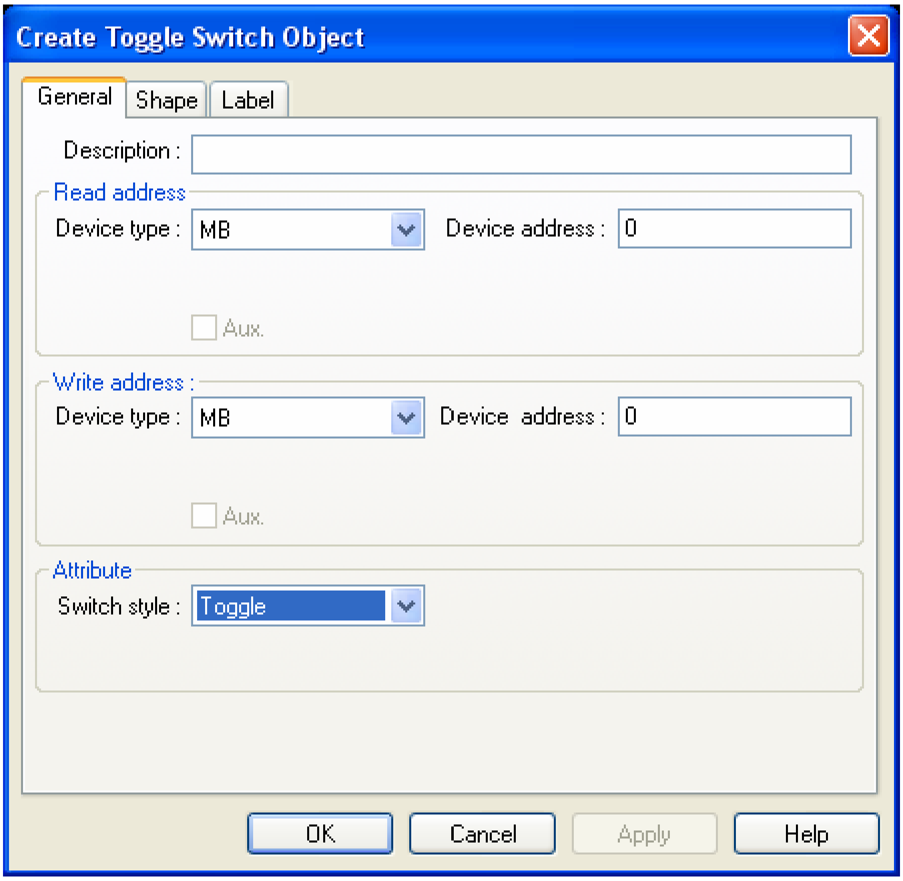
2. [Data bit] has to be 8 bits.

3. [Multiple HMI] has to be set as [Master].

4. [HMI model] depend on the HMI model.

5. The other setting has to be identical between two HMI.

Adding two objects on window10, a toggle switch as below illustration:



Description :

Read address

Device type : MB

Device address : 0

□ Aux.

Write address :

Device type : MB

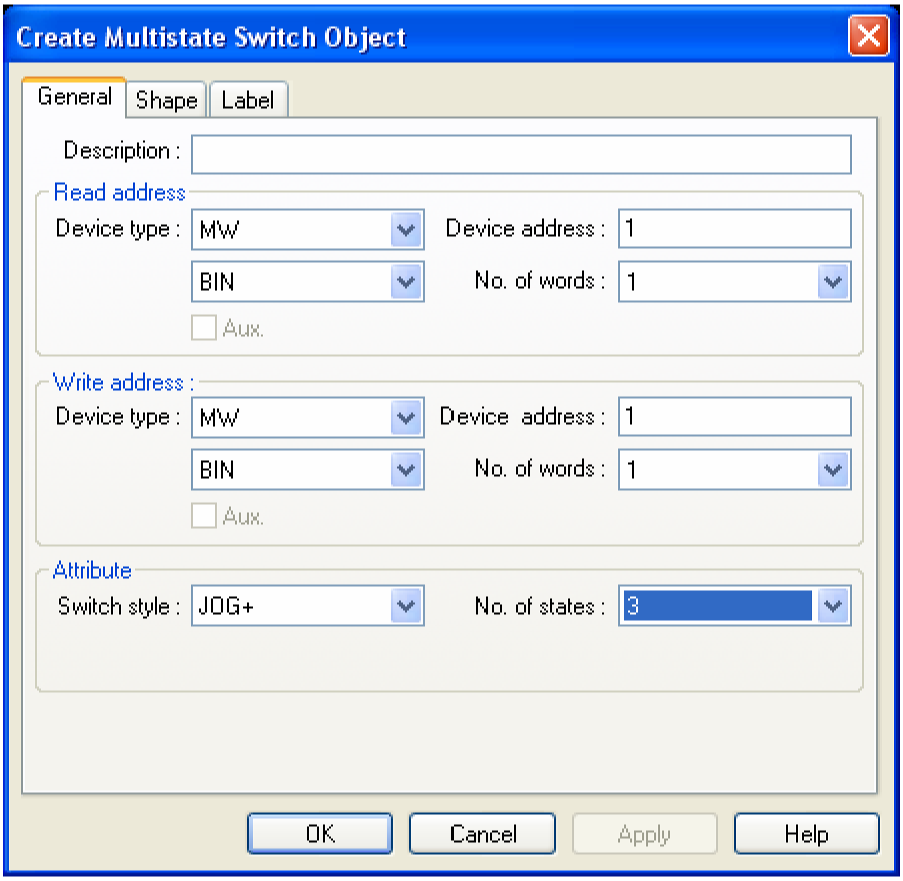
Device address : 0

□ Aux.

Attribute

Switch style : Toggle

A multistate switch object setting as following:



Description :

Read address

Device type : MW

Device address : 1

BIN

No. of words : 1

□ Aux.

Write address :

Device type : MW

Device address : 1

BIN

No. of words : 1

□ Aux.

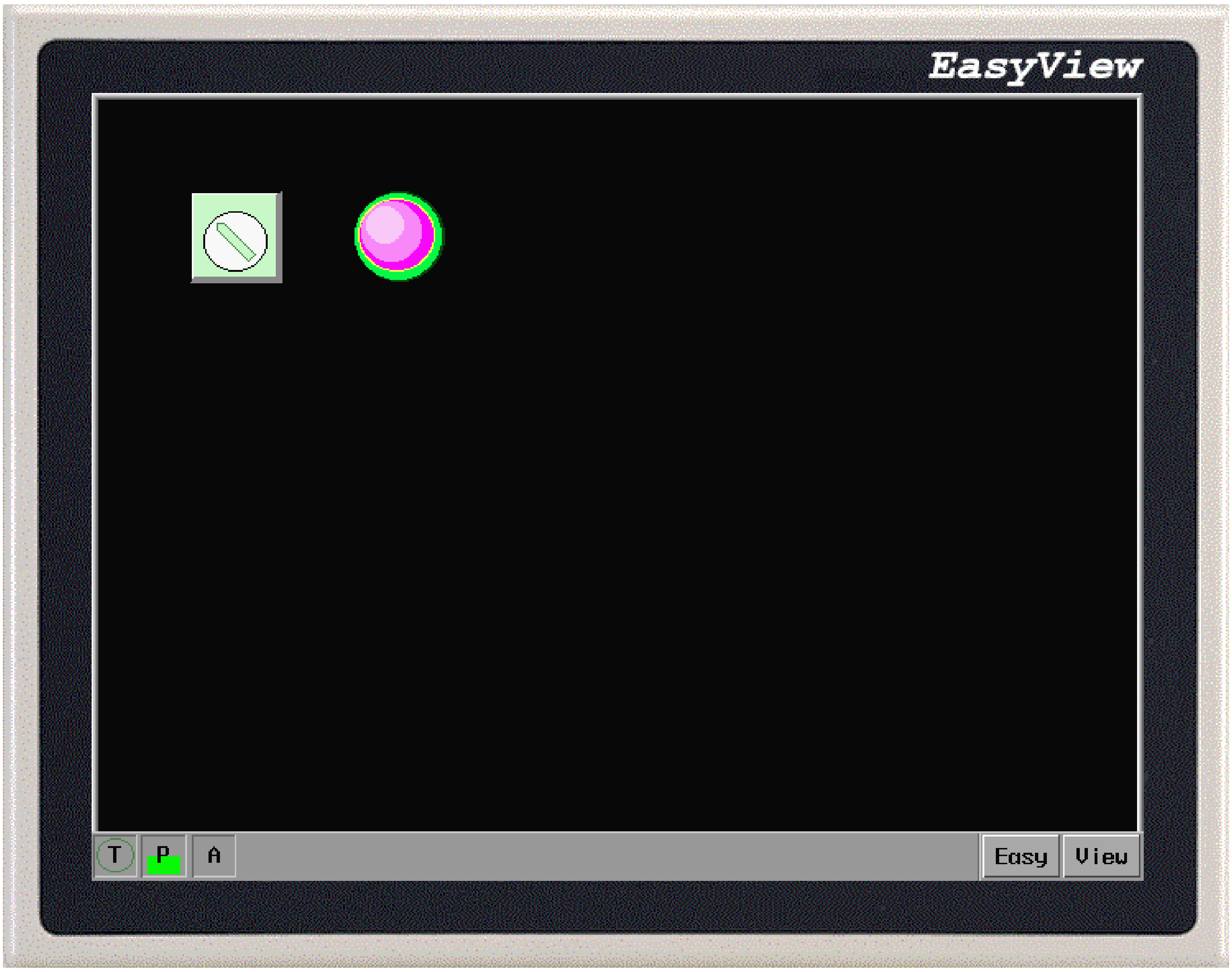
Attribute

Switch style : JOG+

No. of states : 3

[Save],[Compile],[Download]
Change parameter in [System Parameter Setting]/[PLC] and download to another HMI.

The HMI display as following:



Trying to touch the screen, the other HMI will follow the action as current HMI.

The communicating way is the same as above-mentioned. The point is to keep the same data in the same register.