19/04/22

On delay timer

Off delay timer

Pulse timer

Retentive timer /totalizing timer

Timer No

000-255

**Timer base**

1ms timer

10ms timer

100ms timer

1second timer

1min timer

***Timer Syntex***

**On delay timer**

TIM 000 #10 TIM represents 100ms timer

#10 represents delay time

#XX XX represents max value “65535”

000 represents timer number

**Totalizing timer**

TTIM 001 #20

***COUNTER***

1. UP COUNTER
2. REVERSIBLE COUNTER/UP-DOWN COUNTER

COUNTER NO (000 – 255)

CNT 000 #2

CNTR 001 #3

***JUMP***

CJ &012………63

JME &012

***SUBROUTINE***

SBS 0123

SBN 0123

SRET

20/04/23

SET(Latch) & RESET(unlatch)

*Set syntax:* SET address(W0.00)

*RESET syntax:* RSET address(W0.00)

***Functional Block Diagram***

Graphical language for

Programmable logic controller

This defines the function b/w

i/p & o/p variables.

***if (condition) then***

***:=statement;***

***end\_if;***

**if else**

**if(condition) then**

**:=Statement1;**

**Else**

**:=statement2;**

**End\_if;**

**Boolean-> 0/1 True/flase On/Off**

**Integer-> 16bit memory /1 word**

**int -> -32768 to +32767**

**uint-> 0 to 65535**

**dint-> double integer(32bit memory)**

**dint-> -2147483648 to +2147483647**

**udint-> o to 42949667295**

**TASK:**

**SW1 🡪 ON**

**Motor 🡪 ON**

**SW2 🡪 ON**

**Motor 🡪 OFF**

**IF SWITCH1 = TRUE THEN**

**MOTOR := TRUE;**

**END\_IF;**

**IF SWITCH2 = TRUE THEN**

**MOTOR := FALSE;**

**END\_IF;**

***IF & ELSIF***

IF CONDITION THEN

:=Statement1;

ELSIF CONDITION THEN   
 := Statement2;

ELSE

:= Statement3;

END\_IF;

IF VALUE = 1 THEN

MOTOR3 := TRUE;

MOTOR4 := FALSE;

ELSIF VALUE = 2 THEN

MOTOR3 := FALSE;

MOTOR4 := TRUE;

ELSE

MOTOR3 := FALSE;

MOTOR4 := FALSE;

END\_IF;

***DATA REGISTER***

Data register address: D0,D1,D2……..D3999

🡪Single data register carry 16bit of data memory.

***MOVE OPERATION***

SYNTEX: MOV S D where S-> Source, D->Destination

XFER &5 S D &5 defines no.of data is to be moved

***ARITHMETIC OPERATION:***(INTEGER)

ADD: + D20 D21 D22 //D20+D21=D22

SUB: - D24 D25 D26 //D24-D25=D26

Note: The result value should not exceed the integer

Range values.

MUL: \* D27 D28 D29 //D27\*D28=D29

DIV: / D30 D31 D32 // D30/D31=D32

Where D32 stores quotient

(next successive address of OUTPUT) 🡪

D33 stores Remainder

21/4/23

***INTEGER TO FLOAT CONVERSION***

FLT 16bit integer to 32 bit float conversion

*Float arithmetic operation*

Addition : +F

Subtraction : -F

Multiplication : \*F

Division : /F

***FLOAT to INTEGER Conversion***

FIX 32bit Float to 16bit Integer conversion.

In some other PLC’s : INT

***COMPARISION:***

In industry,we use “analog level meter” to measure

the water level of the tank.

**<, >, <=, >=, =**

***COMMUNICATION PROTOCALS***

RS232 🡪 Recomended Standard

Communication 🡪 Serial & Parallel

* Full duplex & Half duplex modes.

Half duplex 🡪 we can either send or receive data .but not

(HD) At a time will not happen

Ex: Walky-Talky

Full duplex 🡪 At a time both operation like sending

(FD) Receiving will take place.

Ex: Cell-Phone,etc.

RS485(HD,FD),RS422(HD,FD),

Modbus,profibus,profinet,hostlink

🡪uploading and downloading take place with

these communication Protocols.