

## 6. LENGUAJES DE PROGRAMACIÓN

6.1. Lenguaje AWL: Este incluye una lista de instrucciones que se ejecutan secuencialmente dentro de un ciclo. Una de las principales ventajas que presenta es que cualquier programa creado en FUP o KOP puede ser editado por AWL, no así a la inversa.

6.2. Lenguaje KOP o Ladder: Este lenguaje también llamado lenguaje de escalera permite crear programas con componentes similares a los elementos de un esquema de circuitos. Los programas se dividen en unidades lógicas pequeñas llamadas networks, y el programa se ejecuta segmento a segmento, secuencialmente, y también en un ciclo. Las operaciones se representan mediante símbolos gráficos que incluyen 3 formas básicas:

- Contactos representan condiciones lógicas de “entrada” Ej.: interruptores, botones, condiciones internas, etc.
- Bobinas representan condiciones lógicas de “salida”, actuadores
- Cuadros, representan operaciones adicionales tales como temporizadores, contactores u operaciones aritméticas

Las ventajas de KOP o Ladder son:

- Facilita trabajo de programadores principiantes
- La representación grafica ayudada de la aplicación “estado de programa” colabora a la fácil comprensión del desarrollo del código.
- Se puede editar con AWL

6.3. Lenguaje FUP: Consiste en un diagrama de funciones que permite visualizar las operaciones en forma de cuadros lógicos similares de los de de las puertas lógicas.

El estilo de representación en forma de puertas gráficas se adecua especialmente para observar el flujo del programa

- Se puede editar con AWL o KOP

NOTA: Se puede alternar a cada tipo de código en el menú “View” del programa STEP7-micro/win32.

## 7. PROGRAMACION EN LENGUAJE ESCALERA (ladder)

La programación en ladder o escalera se basa en programar secuencialmente, los procesos. Se programan linealmente en un ciclo que el PLC repetirá luego de leer cada instrucción de arriba a bajo y de izquierda a derecha.

Para comenzar a familiarizarse con el software, se puede comenzar programando una función lógica “Y”. El resultado debería verse tal como el programa que se presenta en la Figura 7.1:

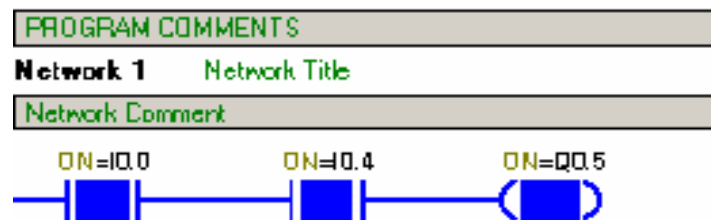


Figura 7.1: Función lógica Y.

Para la configuración de la Fig. 7.1 se debe realizar los siguientes pasos:

1.- Una vez abierto el programa, a la izquierda se tiene el menú de instrucciones de donde se toma el contacto normalmente abierto y se arrastra con el mouse hasta la network 1 y se le asigna una de las señales de entrada de la tabla de arriba. Para este ejemplo, se ha seleccionado la botonera de partir que corresponde a la entrada I0.0; luego se selecciona otra vez el mismo contactor normalmente abierto y se le asigna el selector de dos posiciones que corresponde a la entrada I0.4.

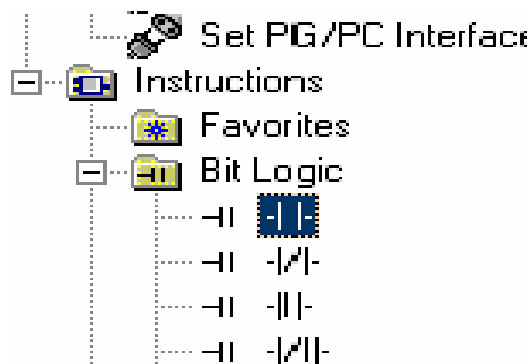


Figura 7.2: Contacto normalmente abierto.

Finalmente se escoge una salida del mismo menú de instrucciones como se muestra en la Figura 7.3. Por ultimo, se le asigna un nombre o dirección del dispositivo actuador ya sea alguna lámpara, contacto, etc. En este caso se usará la luz amarilla del semáforo Q0.5.

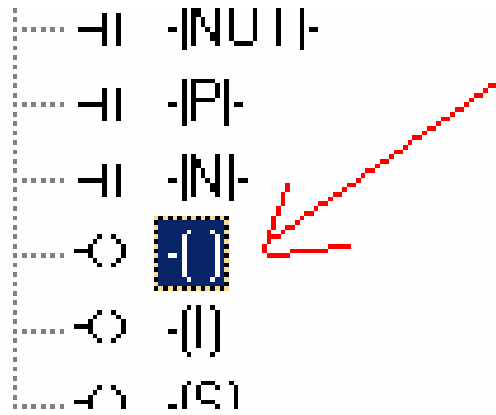


Figura 7.3: Bobina.

#### 7.1. Ejecución de programas



Figura 7.4: Botón download.

En esta etapa, ya se tiene el código diseñado, ahora se le debe volcar al PLC que es lo que tiene que hacer. Para ello se baja el programa a la CPU del PLC, haciendo un clic en el botón “Download” del menú principal de programa. Cuando se realice esta operación aparecerá el cuadro de dialogo de la Figura 7.5:

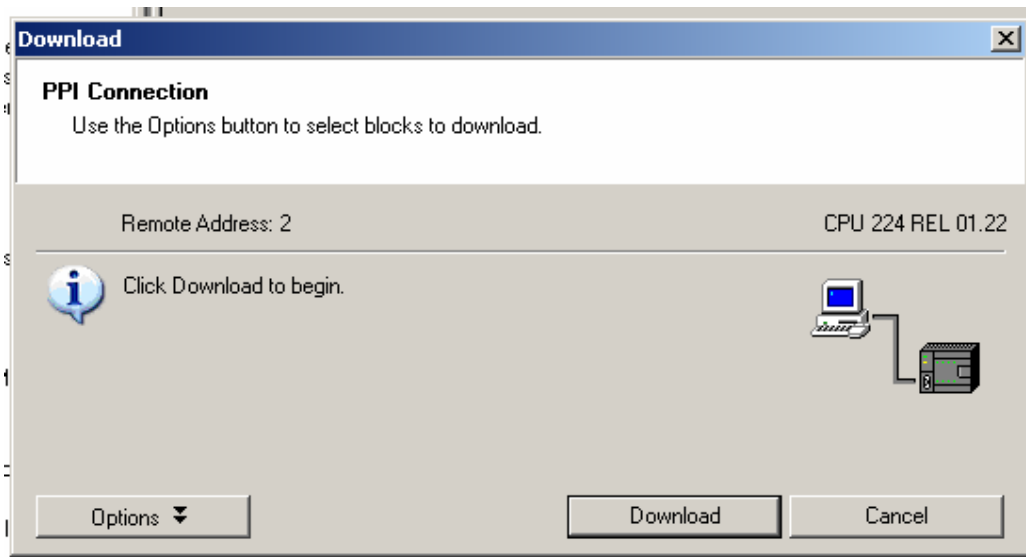


Figura 7.5: Cuadro de descarga.

En este punto se presionará el botón de download y el PLC comenzará el proceso de descarga del programa.

### 7.2. Posibles complicaciones

Si se presentan complicaciones en este punto se debe revisar si el PLC está energizado. Si no lo está, se deberá energizar el PLC. Finalmente se reinicia el programa para que el software reconozca el PLC.

### 7.3. Ejecución de un programa

A continuación del punto 7.3, el PLC seguirá en estado STOP, para ejecutar el programa se debe pulsar el botón “run” o “play” (el verde) y luego aceptar el cuadro de dialogo que aparecerá después de presionar el botón de partida ( ver Figura 7.6). Es necesario señalar que en ningún programa se puede cargar en modo “run” por ende para cualquier edición que desee realizar se deberá poner en modo “Stop” el PLC.

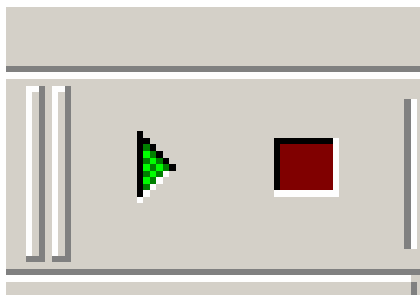


Figura 7.6: Play (verde) y Stop(rojo).

#### 7.4. Estado de programa

Otra aplicación muy útil es el visor de estados de programa que muestra que procesos se está ejecutando. Esta aplicación se puede activar en el menú “Debug” del programa principal, es decir Debug > program status.

#### 7.5. Ejemplos básicos

Una operación lógica recurrente es la operación “o”. Esta se logra con dos contactos en paralelo, que llevan a una misma salida. En el ejemplo de la Figura 7.7, se entrega como alternativa la entrada I0.0 o la entrada I0.6 para dar la salida Q0.0.

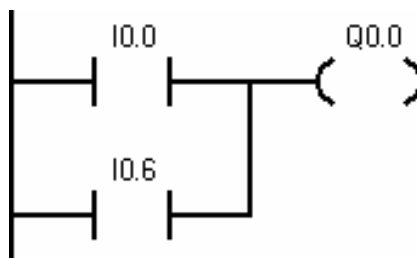


Figura 7.7: Operación lógica O.

Los contadores son elementos básicos, en la Figura 7.8 se muestra un contador hacia abajo o CTD. También existen contadores hacia arriba que son los CTU. Ambos son análogos en su forma de uso. Todos los contadores van desde C1 a C255 sin discriminar si son CTD o CTU o contadores mixtos. Los parámetros del bloque de la Figura 7.8 son: CD es el proceso a contar, LD es el reset y PV es el la cantidad de veces que va a contar el proceso. Luego de que contador cumple con su ciclo, da paso a alguna otra operación, en este ejemplo activa la salida Q1.1 que corresponde al contactor 220 VCA.

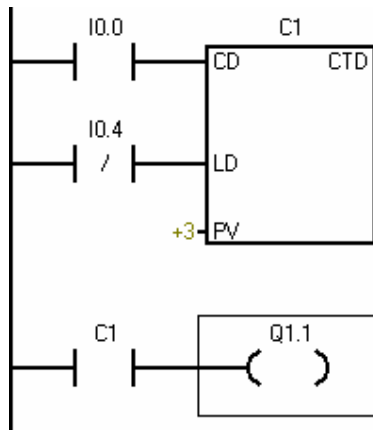


Figura 7.8: Operación de un contador hacia abajo.

Los retenedores o timers funcionan reteniendo o atrasando el funcionamiento de una señal. En el ejemplo de la Figura 7.9 se muestra un TON, que es un retenedor de encendido los parámetros son IN que es la señal que retendrá por un tiempo PT, luego de cumplirse dará paso a otra señal, en el ejemplo: Q0.0 que es la luz de “Operando” que no se encenderá hasta que el contactor I0.0 (botón partir) se presione por un tiempo PT.

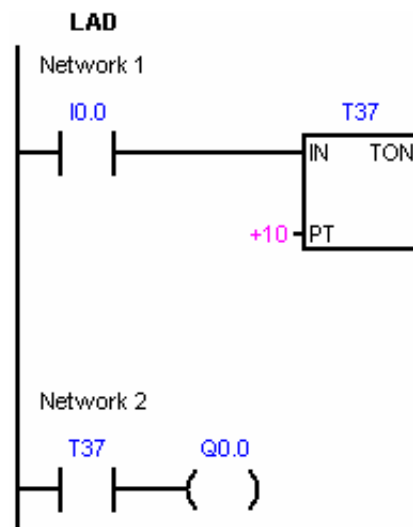


Figura 7.9: Timer de encendido.

Otros timers son TOF y TONR cuyos parámetros son análogos con la salvedad de que TONR retiene la cantidad de tiempo que ha sido usado más allá de lo establecido en el parámetro PT.

Otro punto importante en los timers es su nombre (en ejemplo de arriba T37), es necesario notar que el tipo de timer esta dado por la selección que uno realice en el menú de sus tipos (TON, TOF, TONR) pero no podrán llamarse con el mismo nombre a dos tipos de timers es decir el programa no reconocerá a un TON T37 y otro TOF T37.

Otro punto destacable respecto de los timers son sus resoluciones, en la tabla 5 inferior se puede apreciar con que nombre se deberá llamar a un timer para obtener la resolución deseada, dependiendo de ello, el timer escogido será más rápido o más lento.

Timer Type	Resolution	Maximum Value	Timer Number
TONR	1 ms	32.767 s	T0, T64
	10 ms	327.67 s	T1-T4, T65-T68
	100 ms	3276.7 s	T5-T31, T69-T95
TON, TOF	1 ms	32.767 s	T32, T96
	10 ms	327.67 s	T33-T36, T97-T100
	100 ms	3276.7 s	T37-T63, T101-T255

Tabla 5: Resolución de los timers.

Específicamente, suponga que ha escogido el timer T37 (Figura 7.10). A partir de la tabla 5 se establece que este timer mide en unidades de 100ms, por ende cuando el parámetro PT tiene un valor de 100, como en el ejemplo de la derecha, el tiempo de retardo que dará este timer será de  $t = 100\text{ms} * 100 = 10\text{s}$ . De esta manera dependiendo de la aplicación se deberá escoger el timer más adecuado a la misma.

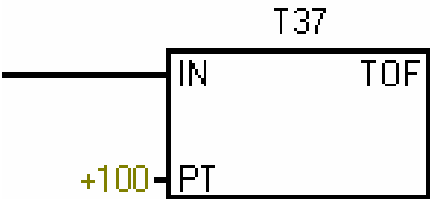


Figura 7.10: Timer de apagado con parámetro 10 segundos.

## 7.6. Combinaciones útiles

### 7.6.1. Auto retención

La motivación de considerar la combinación de la Figura 7.11 es que dada la configuración lineal de una entrada y salida, es que la salida se desactivará a los pocos

milisegundos, en cambio si se pone una condición para auto retener o auto sostener el estado deseado no se requiere mantener presionada o ejecutándose alguna señal de entrada. El software nos ofrece tres alternativas para lograr el mismo efecto:

#### 1.- Juego de contactos

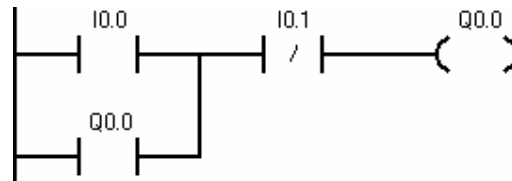


Figura 7.11: Auto retención.

Dado el juego de contactos de la Figura 7.11, la salida Q0.0 que corresponde en nuestro caso a la luz verde de “operando” se encenderá dando paso o habilitando como verdadera la señal de entrada Q0.0 asociada en un “O” lógico como entrada al mismo sistema, de esta manera la auto retención de la salida se hará posible, es decir la luz se mantendrá encendida gracias a si misma.

#### 2.- Set & Reset

La idea en este punto es setear de manera permanente, con un sólo impulso de la entrada, una condición en la salida Q0.0, es decir estamos haciendo lo mismo de arriba pero de una forma menos engorrosa.

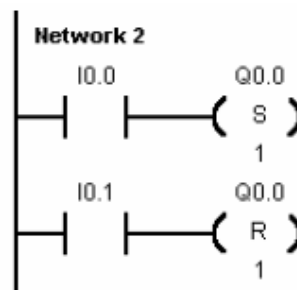


Figura 7.12: comandos SET y RESET.



Las "bobinas" - (S) Poner Q0.0 a "1"  
 - (R) Poner Q0.0 a "0"

se utilizan con frecuencia en PLCs para mantener permanentemente activadas o desactivadas entradas, salidas o marcas cuando se active brevemente (por impulso) un contacto antepuesto.

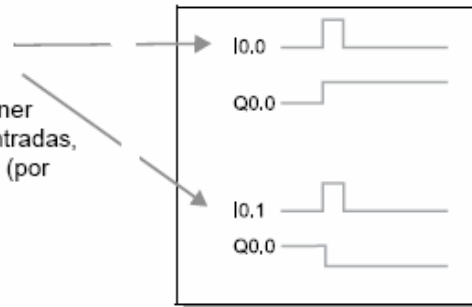


Figura 7.13: Funcionamiento de los comandos SET y RESET.

### 3.- Latches

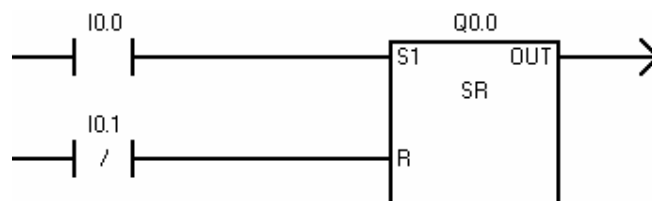
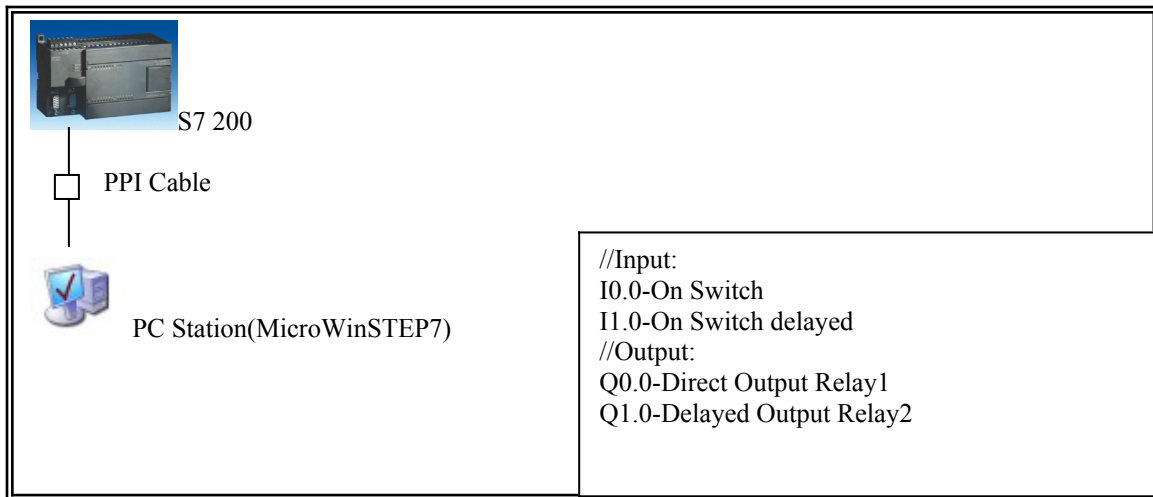


Figura 7.14: Latch de seteo.

Los latches de la Figura 7.14 son comandos o cajas de herramientas que permiten generar el mismo efecto de los puntos anteriores, retener una salida en un estado. En este caso estos comandos se pueden encontrar en el menú "bit logic" y son dos comandos SR (setear instrucción) y RS (resetear instrucción).

## Program-1: Delayed Output using On-delay timer



### Goal:

How to write Direct Output and Delayed Output using LADDER Logic?

### What to do:

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Check the Input I0.0 and I 1.0
- Check the Output Q0.0 and Q1.0
- Switch ON I0.0 and check Q0.0
- Switch ON I1.0 and check Q0.0

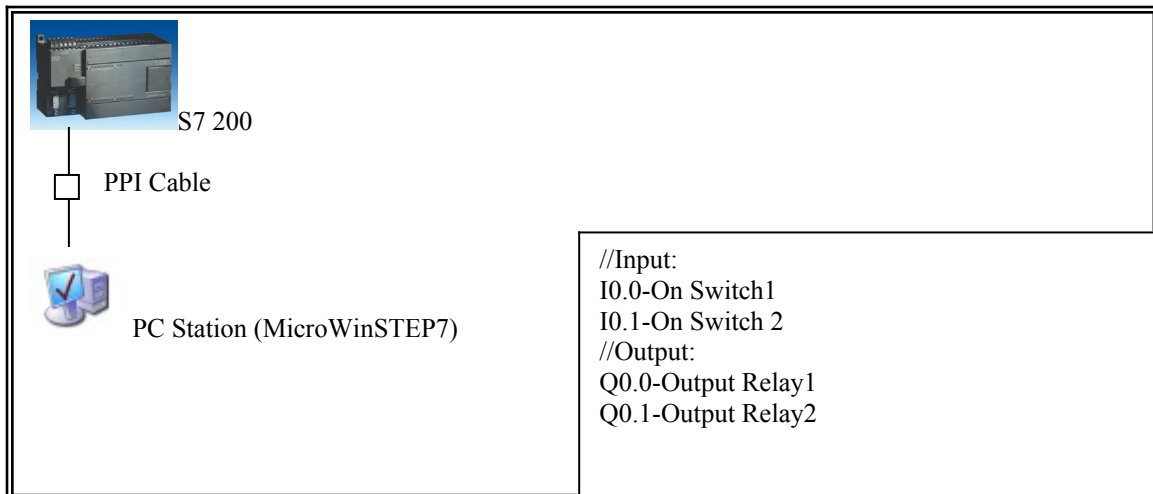
### Result:

- When you switch ON I0.0 the Output directly ON but when you Switch ON I1.0 the Output Q1.0 is delayed by time-delay of 3Sec.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Set time delay 5sec and check
- Set the time delay 0sec and check
- Use OFF-delay timer to implement the above task.

## Program-2: Interlocking Outputs Part1



### Goal:

One output is depending on other output. To implement this interlocking using LADDER logic. Q0.0 can be high when Q1.0 is Low and Vice-versa.

### What to do:

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Check the Input I0.0 and I 0.1
- Check the Output Q0.0 and Q0.1
- Switch ON I0.0 and check Q0.0
- Switch ON I0.1 and check Q0.0
- Switch OFF Either I0.0 or I0.1 and check Outputs.

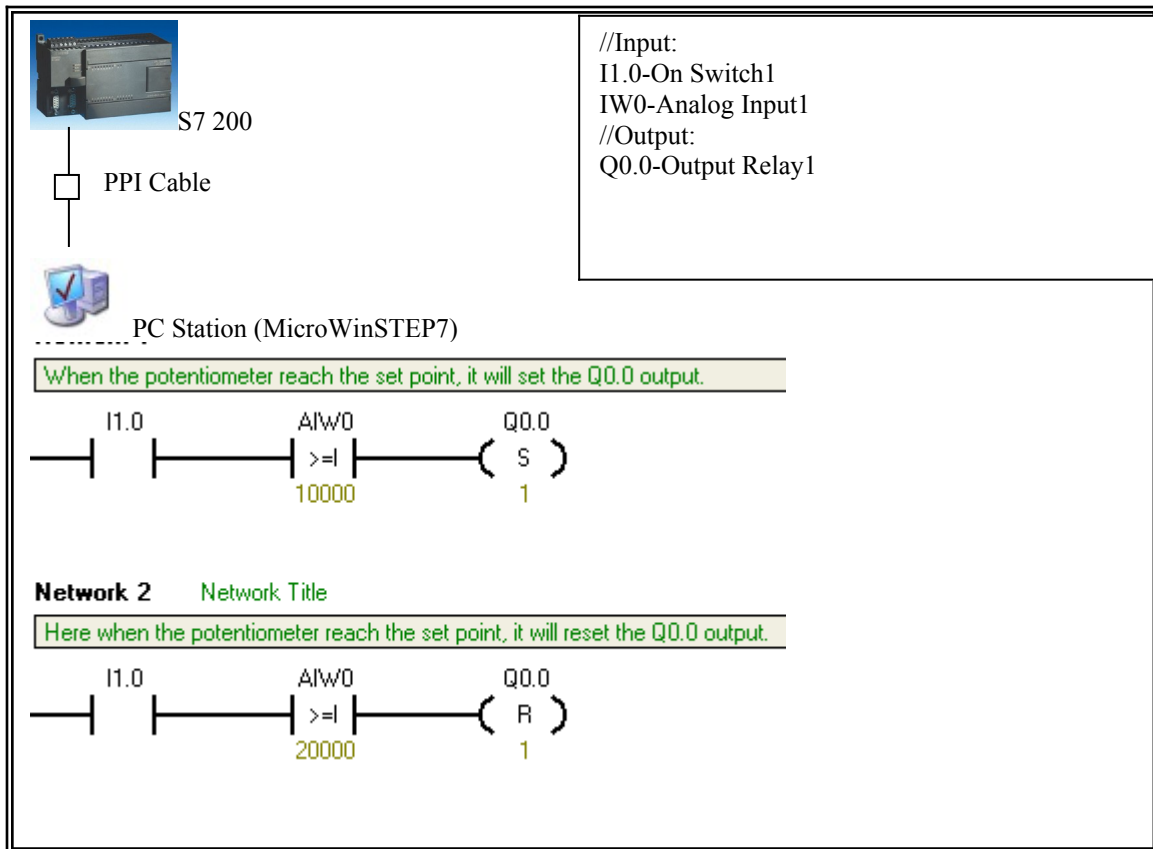
### Result:

- When you switch ON I0.0 the Output Q0.0 is interlocked with Q0.1 but when you Switch ON I0.1 the Output Q0.1 is high. As per interlocking condition is first output goes to Low.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Check when both the inputs are High
- Instead of Output interlocking use input interlock
- Use On-delay timer to interlock.

### Program-3: Interlocking Outputs Part2



#### Goal:

Set or reset Output-relay as per analog "input-value" IW0. Set Q0.0 when input  $\geq 10000$  and reset when  $\geq 20000$ .

#### What to do:

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Check the Input I1.0.
- Check the Output Q0.0
- Switch ON I1.0 and check Q0.0
- Rotate POT clock-wise full and again rotate counter-clock wise until zero position and check Q0.0

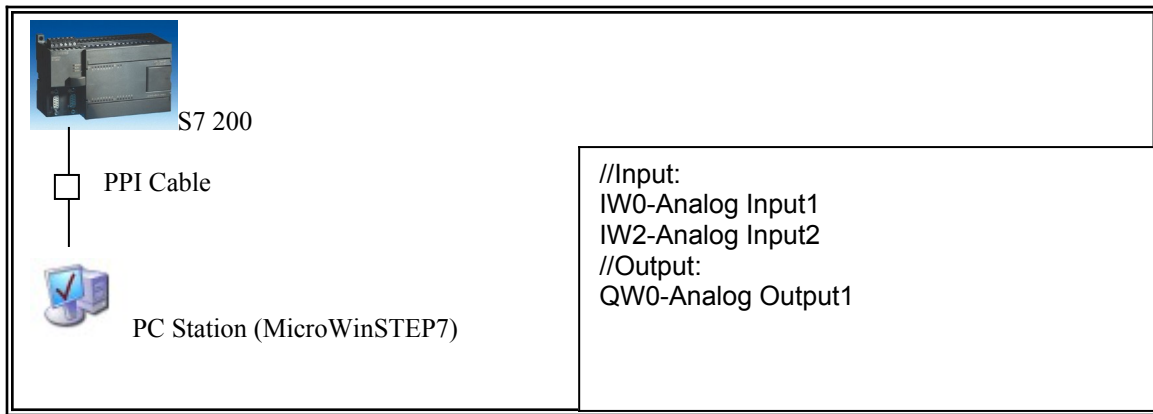
#### Result:

- When you switch ON I1.0 the Output Q0.0 is interlocked with analog input IW0. When the analog value become more than 10000 the output q0.0 become High until the analog value greater than 20000.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Change the analog set point to 15000 and 30000.
- Use <, >, <= and == instruction for set-reset program.

#### Program-4: Subtract two analog input and display at analog output



#### Goal:

Difference between analog input IW0 and IW2 will be displayed on Analog output QW0.

#### What to do:

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Check the Input IW0 and IW2.
- Check the Output QW0
- Switch ON I1.0 and check analog output QW0.
- Rotate POT1 & 2 and check QW0.

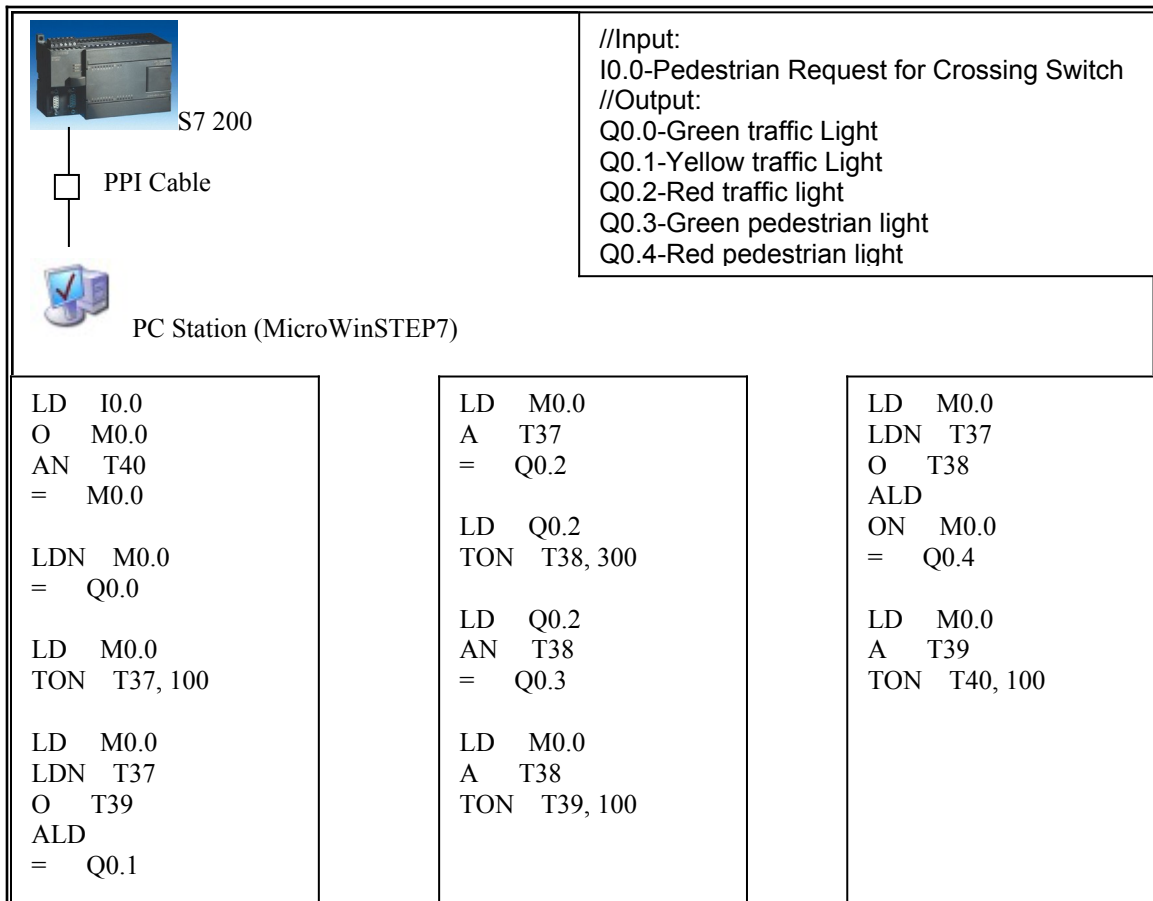
#### Result:

- If you set IW0 at maximum and increase the IW2 value slowly the Output value also will increase and will be equal to  $IW0 - IW2$ .

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Change the analog input IW0 at minimum position and increase IW2.

### Program-5: Traffic light system.



#### Goal:

The program monitors the pedestrian lights (which include a request switch and Walk/Don't Walk light) and traffic light for cars.

**What to do:** -----???

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Check the default state for traffic lights is green for car and red for pedestrian.
- Check when you Switch ON I0.0 program receive a green pedestrian Walk and it cycles the traffic from green, to yellow, to red.
- Traffic light: Yellow light 10 sec, Red 30 sec, Green 30 sec
- Pedestrian light: Green & Red light 10 sec

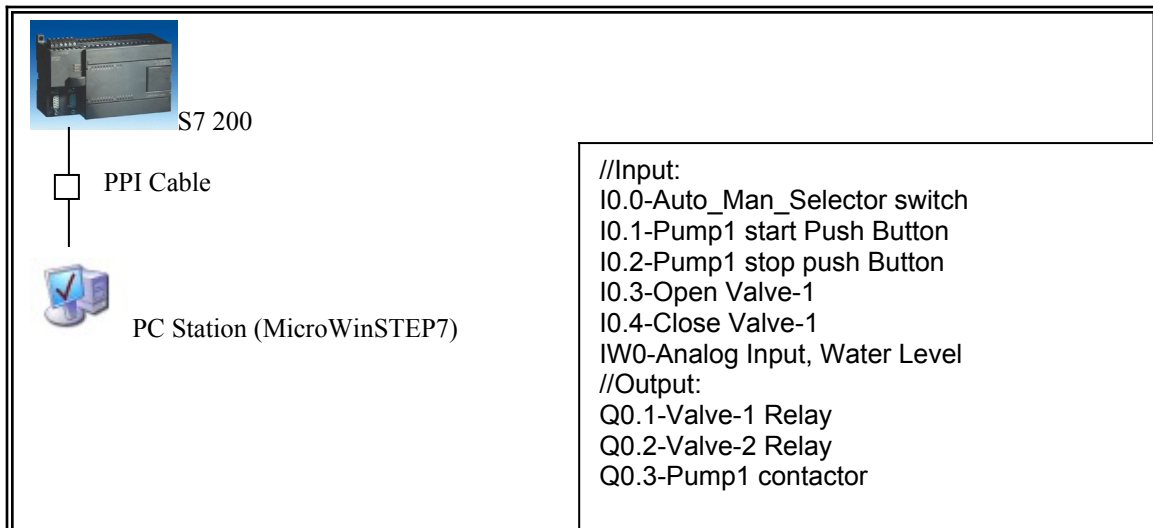
#### Result:

- When you press the I0.0 the traffic signal will follow from Green to yellow to red and pedestrian light is "Green"

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Make new T-junction traffic program
- Make 4-road traffic -crossing program

## Program-6: Water tank level control and level-alarm generation.



### Goal:

In manual mode all the Valves and Pump1 can be start/Stop using individual Start Stop Switch. When I0.0 is low i.e., when in auto mode the Pump1 will start when water level is very low and Valve1 will open at level 25%. Pump1 and Valve2 operate at the same time. Pump1 stop when Level 100% reached. Valve2 also closed as pump1 stops. Valve1 closed only if level is very low (0%).

### What to do:

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Check the Input IW0
- Check the input I0.0, I0.1, I0.2, I0.3, I0.4
- Check the Output Q0.1, Q0.2, Q0.3
- Select mode-manual by switching ON I0.0 and check each valve & pump1 by manual operation.
- Select mode-Auto by switching OFF I0.0 and Simulate IW0 by rotating POT1.
- Monitor the alarm bits M0.0, M0.1 to M0.5

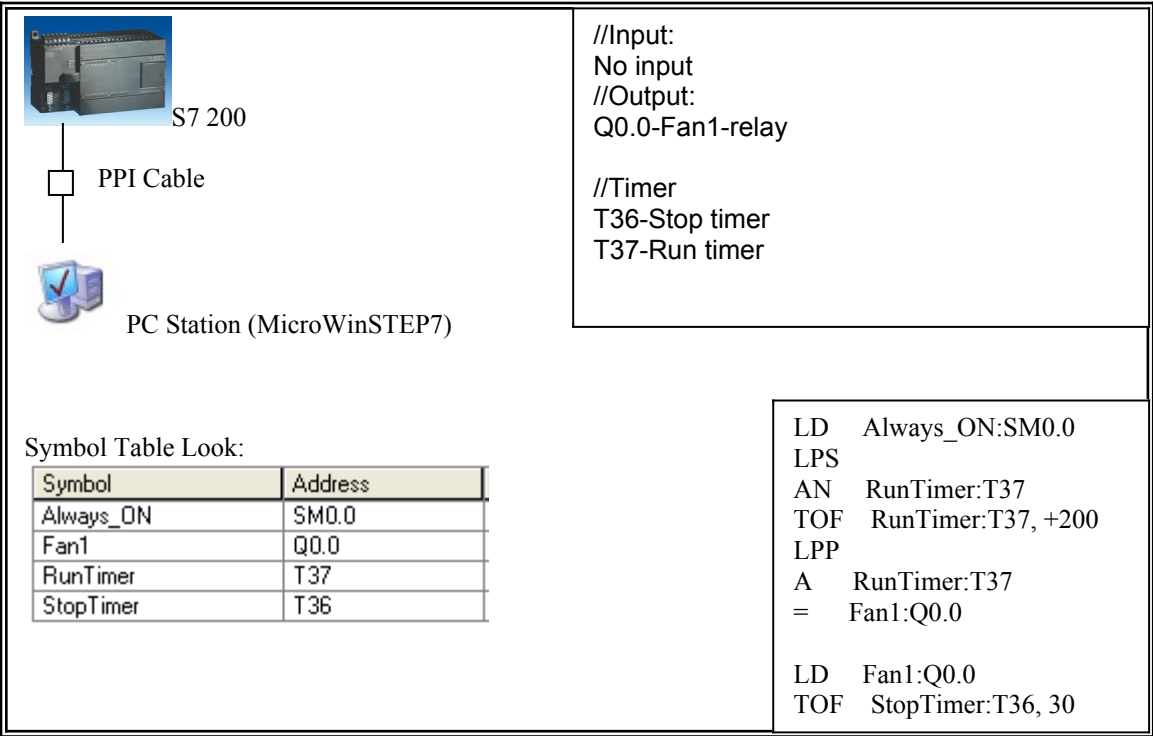
### Result:

- When you select manual mode you can start Pump1 by I0.1 and stop by I0.2 and open valve 1 by I0.3, close valve1 by I0.4. But when you select auto mode the pump1 automatically start and stop as per input signal IW0 that is Water Tank level signal.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Change the analog input IW0 at minimum position and increase IW0 slowly.
- Change the analog input IW0 at Maximum position and Decrease IW0 slowly.
- Add more analog alarm levels at 10% and 90%.
- Use 10% level to open Valve-1

**Program-7: Time based fan control.**



**Goal:** According to Timer T36 and T37 the Fan1 will start and stop.

**What to do:**

Write the program and download it to Siemens S7 200 PLC.


- Check the PLC mode, Run Mode
- Check the Run Timer T37 and Stop Timer T36.
- Check the Output Q0.0

**Result:**

- The Fan1 will continue running as long as T37 is running and Fan1 will be in OFF position until T36 finished.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Change the timer values.
- Use MW10 for Timer input T36 and MW12 for T37
- Change the MW10 and MW12 value in the Status Chart
- Monitor Q0.0 in the status chart



CPU 224XP REL 02.00

Program Block

Symbol Table

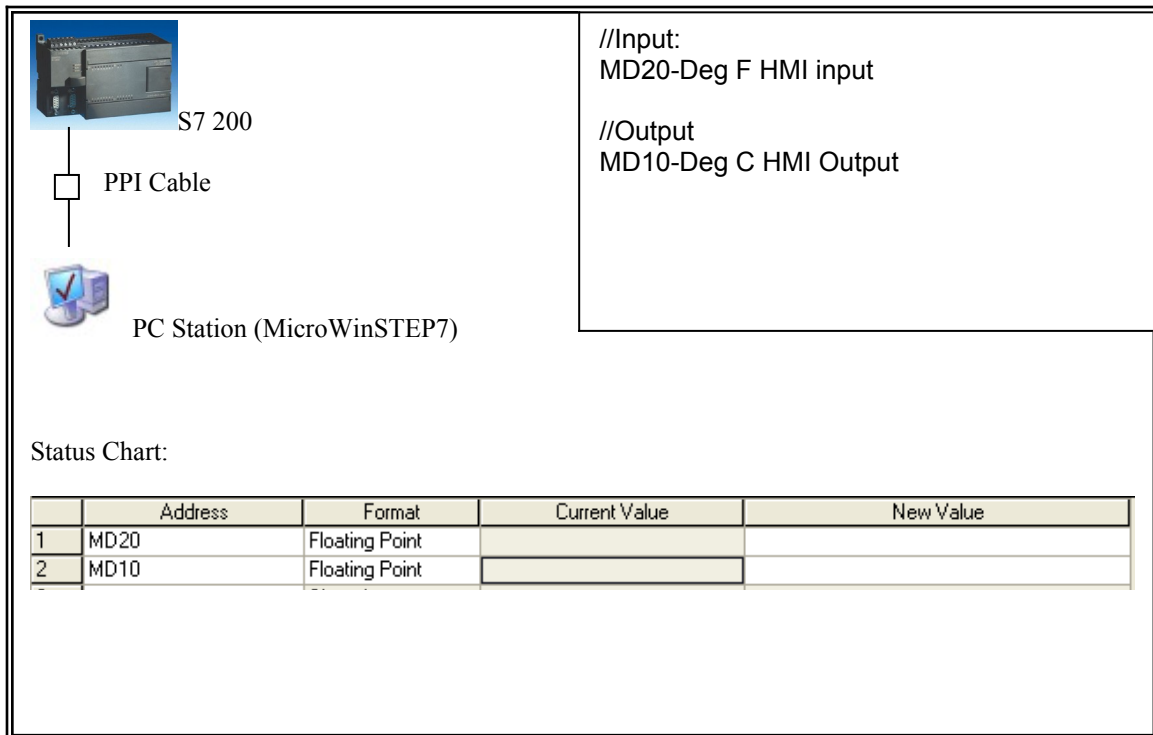
Status Chart

USER1

	Address	Format	Current Value	New Value
1	MD20	Floating Point		
2	MD10	Floating Point		
3		Signed		
4		Signed		
5		Signed		



### Program-8: Temperature C-F conversion.



**Goal:** We need to convert user input temperature in Deg F to Deg C.

#### What to do:

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Open Status Chart.
- Enter MD 20 as Floating point
- Enter MD10 as Floating point
- Change the value of MD20 (Deg F) and check deg C in MD10.

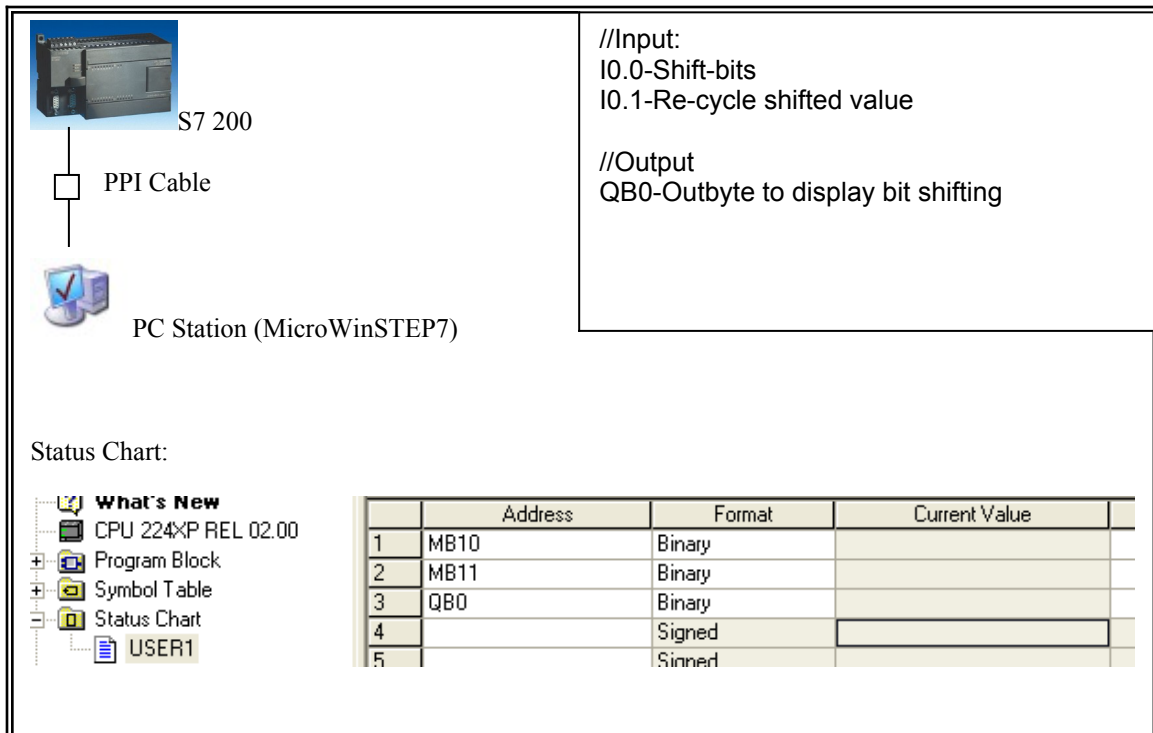
#### Result:

- $C = (F - 32) / 9 * 5$  formula implemented. MD10 is in Deg C which is equivalent to MD 20 in Deg F.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Change the timer values.
- Convert Deg C to Deg F

## Program-9: Shift Instruction



**Goal:** Shift the value in MB10 to the left by one-bit position each time we switch ON I0.0. For recycling the shifted bits I0.1 need to Switch ON and OFF.

**What to do:**

Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Open Status Chart.
- Enter MB10 as binary
- Enter MB11 as binary
- Check the value of MB10 between 1 and 254
- Step-1: Switch ON I0.0 and then OFF
- Step-2: Switch ON I0.1 and then OFF
- Repeat Step1 and 2.

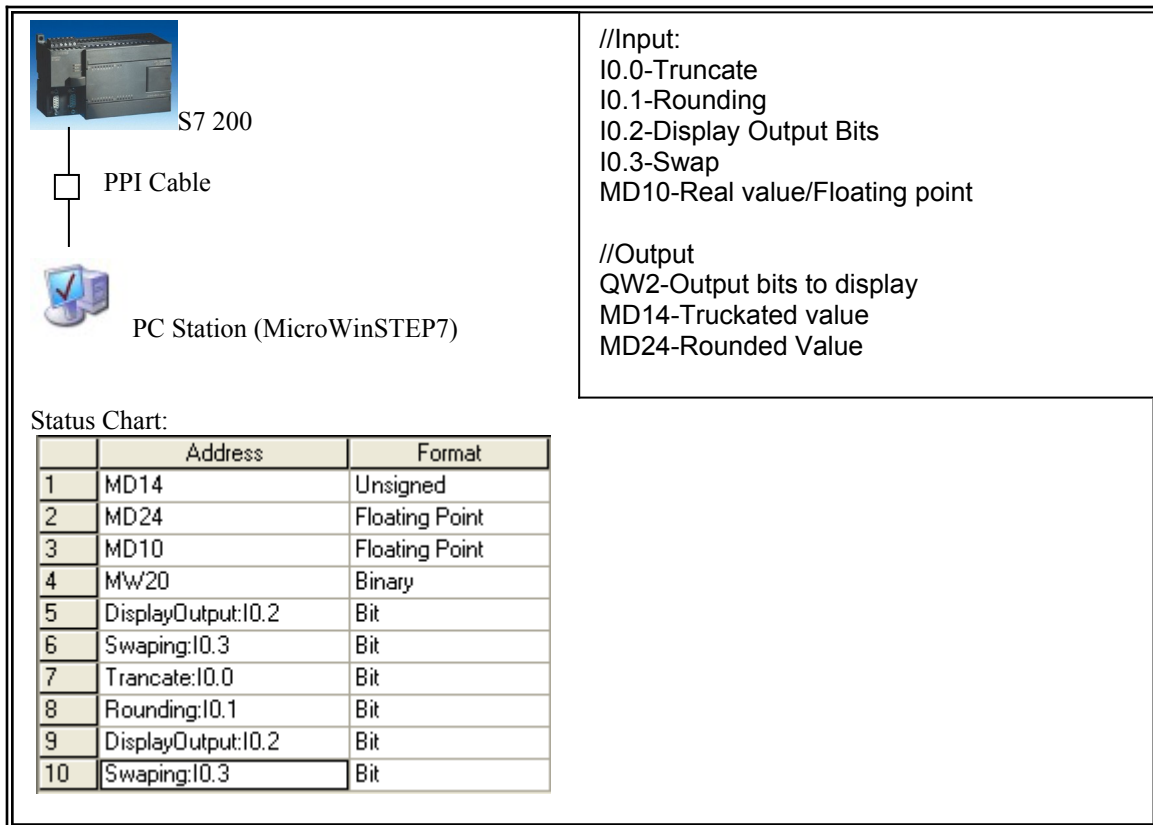
**Result:**

- Resulting bit structure will move left every time you switch ON I0.0 by one bit.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Change the number of Bit (N) to be shifted left to 2.
- Use Shift-right Instruction for same program.

## Program-10: Rounding and Swapping



**Goal:** Differentiate Rounding and Truncate instruction and Use of Swap Instruction

**What to do:** Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode
- Open Status Chart.
- Enter address as per above Status Chart
- Enter 101.6 to MD10, Enter 2#0101 to MW20
- Switch ON I0.0 to Truncate & check MD14
- Switch ON I0.1 to Round & check MD24
- Switch ON I0.2 to Display Bit Structure at Q2.0 to Q3.7
- Switch ON I0.3 to Swap the MW20

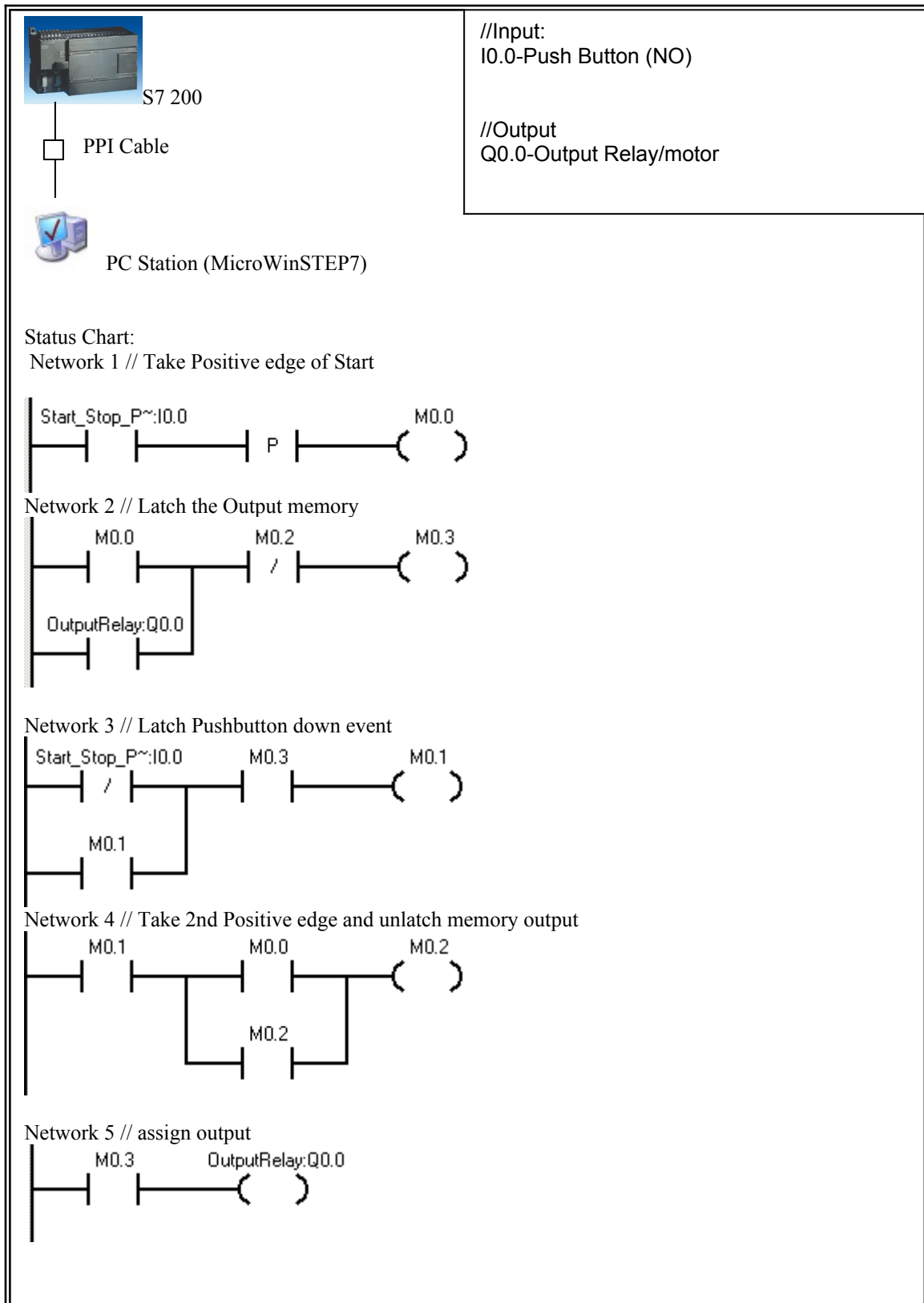
**Result:**

- Truncated value 101 at MD14
- Rounded value 102 at MD24
- LEDs of QW2 are swapping from lower byte to higher and Vice versa.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Change the MD10 value and check MD14, MD24 when I0.0 and I0.1 are Switched ON.
- Change the value of MW20 and check swapping.


## Program-11: Single pushbutton Start and Stop



**Goal:** Start and Stop motor (Q0.0) using single NO push button (I0.0)

**What to do:** Write the program and download it to Siemens S7 200 PLC.

- Check the PLC mode, Run Mode

- 
- - Enter address as per above Status Chart
  - Press Start Button & check output
  - Press again the same button and check output.

### Result:

- When you press the pushbutton (I0.0) first time the motor Q0.0 will start. If you release the button nothing will happen to motor. If you press the button second time the motor will stop.

**Note:** When downloading the program please check PC address, PLC address, baud rate etc correctly.

- Make same program using different logic instructions other than SET-RESET
- Use Set-Reset to do the same job.