

SISTEMA DECIMAL CODIFICADO EN BINARIO

Hasta el momento hemos visto que la información que almacena el PLC en sus espacios de memoria puede adoptar dos valores. Un bit de memoria que se encuentra apagado se le asocia el valor 0 mientras que un bit de memoria que se encuentra prendido se le asocia el valor 1.

Lo que podemos preguntarnos con esto es cómo hace el PLC para almacenar **valores numéricos mayores a 1**. Si quiero que el PLC guarde la información del valor de temperatura en 35 (grados) o el valor de presión en 4 (bar) o un tiempo de 24 (segundos) ¿Cómo logra guardar en memoria todos esos números si solo dispone de dos, el 0 y el 1?

En nuestro sistema de numeración, el **decimal**, disponemos de **diez dígitos** para representar todos los números. Los diez dígitos son 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Con estos dígitos evidentemente podemos cubrir cantidades que van de 0 a 9. ¿Qué sucede si al 9 le queremos agregar una unidad más? Claramente un solo dígito no nos alcanza para esto así que debemos incluir un segundo dígito (al cual le damos el nombre de decena).

Por ende, para pasar del 9 al siguiente número podemos hacer simplemente $9 + 1 = 10$, o también podemos decir que “volvemos el 9 a 0 y le agregamos un 1 a la izquierda en el siguiente dígito”

O sea 9 -> 0 con un 1 a su izquierda -> 10

Esta forma de decir que pasamos de 9 a 10 puede resultar muy poco práctica, pero es la base con la que el PLC puede aumentar las cantidades por encima de 1 (recordemos que solo dispone de 0 y 1 por lo que no puede guardar el número 2 salvo que empiece a agregar más dígitos).

Armemos unas tablas que nos permitan ir aumentando las cantidades que puede guardar el PLC empezando desde 0.

Evidentemente al principio tenemos:

Dígito decimal	Dígito binario
0	0
1	1

Los números 0 y 1 del decimal los podemos representar directamente con los números 0 y 1 del sistema binario que usa el PLC. Si queremos representar en número 2 decimal, necesitamos la ayuda de un segundo dígito binario, lo cual nos quedaría así



Agregamos otro dígito binario

Dígito decimal	Dígito binario
0	0
1	1
2	???

Dígito decimal	Dígito binario	Dígito binario
0	0	0
1	0	1
2	1	0

Con un nuevo dígito binario ya podemos aumentar la cantidad decimal. “volvemos el 1 a 0 y le agregamos un 1 a la izquierda en el siguiente dígito” (lo mismo que dijimos en el sistema decimal).

O sea 1 -> 0 con un 1 a su izquierda -> 10

Esto quiere decir que para representar una cantidad superior a 1 el PLC va a requerir como mínimo **dos espacios de memoria**, el primer espacio representa los valores 0 y 1 del decimal y el segundo espacio permite seguir representando decimales pasando del 1, o sea nos permite llegar al 2 y al 3.

Dígito decimal	Dígito binario	Dígito binario
0	0	0
1	0	1
2	1	0
3	1	1

Entonces nuestros primeros números binarios que podemos representar quedarían

00 para el 0
01 para el 1
10 para el 2
11 para el 3

Podemos continuar de esta manera y representar más números decimales. Con un nuevo dígito binario ubicado a la izquierda podemos abarcar más números decimales hasta llegar al 7, de esta manera.

Dígito decimal	Dígito binario	Dígito binario	Dígito binario
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Ya solo nos quedaría poder representar el 8 y el 9 para cubrir todos los dígitos decimales, con lo cual necesitamos un cuarto dígito binario.

Dígito decimal	Dígito binario	Dígito binario	Dígito binario	Dígito binario
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

De esto podemos concluir que, por cada dígito decimal, el PLC requiere **un grupo de cuatro bits** prendidos y apagados según las combinaciones que vemos arriba para hacer la representación.

Si queremos representar la cantidad 5 en binario, según lo que dice la tabla, sería.

Primer bit (todo a la derecha) = 1

Segundo bit (el siguiente a la izquierda) = 0

Tercer bit = 1

Cuarto bit = 0

Por ende 5 en decimal sería 0101 en binario.

¿Qué haríamos si queremos representar un número más grande que el 9? En el sistema decimal sabemos que necesitamos un nuevo dígito. En el sistema binario lo que podemos hacer es agregar un **nuevo grupo de cuatro bits** y usarlo para el nuevo dígito que estamos agregando en decimal.

Si queremos representar el 94 en binario, podemos tomar un grupo para el 9 y otro para el 4.

9 en decimal = 1001 en binario.

4 en decimal = 0100 en binario.

Por ende, usaríamos 8 bits de memoria del PLC que, unidos, quedarían 10010100.

El manual del fabricante del PLC nos debería informar cuál es la cantidad máxima en decimal que se puede representar de forma binaria. El PLC Izumi del laboratorio logra representar hasta el 9999 ya que posee cuatro grupos de cuatro bits (16 bits en total) para destinar cada valor numérico que debe guardar en memoria.

REGLAS PRÁCTICAS PARA PASAR ENTRE EL SISTEMA CODIFICADO EN BINARIO Y EL DECIMAL

Si resulta complicado recordar la tabla de pasaje entre decimal y binario, podemos hacer los siguientes cálculos.

Partiendo del número binario, para llegar al decimal se debe multiplicar cada dígito binario por 8, 4, 2 y 1 respectivamente y sumar los resultados.

Ejemplos:

$$1001 = 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 8 + 0 + 0 + 1 = 9$$

$$1001 = 9$$

$$0110 = 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 0 + 4 + 2 + 0 = 6$$

$$0110 = 6$$

Partiendo del número decimal, para llegar al binario se debe dividir sucesivamente por 2 e ir anotando los restos de cada cuenta y el resultado final.

Ejemplos:

$$7 / 2 \text{ (resultado 3, resto 1)}$$

$$3 / 2 \text{ (resultado 1, resto 1)}$$

$$1 / 2 \text{ (resultado 0, resto 1)}$$

$$7 = 0111$$

$$8 / 2 \text{ (resultado 4, resto 0)}$$

$$4 / 2 \text{ (resultado 2, resto 0)}$$

$$2 / 2 \text{ (resultado 1, resto 0)}$$

$$8 = 1000$$