

<b>Procedimiento 5</b>	<b>CÉLULA FLEXIBLE</b>	<b>19-5-2006</b>
Corrección del FB190 programa puesto de Visión		<b>Puesto 6</b>

## Objetivo

Corregir el módulo FB190 del programa de la cámara de Visión F150 para proporcionar al PLC correctamente los datos medidos por ésta.

## Ámbito de aplicación

Con los datos obtenidos, permitir al PLC la toma de decisiones al respecto, indicando qué piezas son buenas, y qué defecto tienen las malas.

## Modificación:

### **Célula flexible. Puesto 6 (visión). FB 190.**

#### **Parte del segmento 10 ('leemos los datos que devuelve la cámara')**

```

1  L  #Periferia.OUTPUT_AREA_CAMARA.PARTE_DECIMAL_DATO1
2  BTD                // Esta operación no modifica ACU2
3  DTR                // Esta operación no modifica ACU2
4  L  1.000000e+003
5  /R
6  L  #Periferia.OUTPUT_AREA_CAMARA.PARTE_ENTERA1_DATO1
7  BTD                // Esta operación no modifica ACU2
8  DTR                // Esta operación no modifica ACU2
9  +R
10 L  #Periferia.OUTPUT_AREA_CAMARA.PARTE_ENTERA2_DATO1
11 SLD 19             // Esta operación no modifica ACU2
12 OD
13 T  #AuxiliarREAL
14 T  #RESULTADO_0

```

## Explicación:

### ***Acondicionamiento de la parte decimal***

1. Cargamos en ACU1 la parte decimal del dato que envía la cámara.
- 2,3. Transformamos el valor recibido desde BCD (recordar que debe realizarse el ajuste en el menú de la cámara para que envíe los datos en BCD y no en binario) hasta REAL, pasando por DoubleWord (no hay una operación directa). Notar que estas operaciones actúan sobre ACU1 sin variar el valor de ACU2.
4. Cargamos 1000 en ACU1, con lo que el valor decimal (de tres dígitos) pasa a ACU2.
5. Dividimos ACU2/ACU1 (con lo que el valor decimal pasa a la derecha del punto decimal, por ejemplo '374' pasa a '0.374').

### ***Acondicionamiento de la parte entera***

6. Cargamos en ACU1 la parte entera del dato que envía la cámara, con lo que la parte decimal (REAL) pasa a ACU2.

7,8. Transformamos la parte entera de ACU1 de BCD a REAL, de modo al final la parte entera (ACU1) y la decimal (ACU2) tienen el mismo formato (REAL).

*Obtención del valor absoluto del dato (sin signo)*

9. Sumamos la parte entera (está a la izquierda del punto decimal) y la parte decimal (está a la derecha del punto decimal).

***Incorporación del signo y obtención del valor completo del dato***

10. Cargamos en ACU1 el signo del dato, con formato de Palabra y con dos posibles valores: 0 si el signo es positivo y f000 si el signo es negativo. El valor del dato sin signo pasa a ACU2.

11. Transformamos el formato del signo de Palabra a Doble Palabra y realizamos un desplazamiento de 19 posiciones hacia la izquierda, de modo que el bit más a la izquierda queda un 1 si el signo es negativo y 0 si es positivo. El resto de los bits son siempre 0. Como esta operación no modifica el valor del ACU2, en ACU2 todavía tenemos el valor del dato sin signo.

12. Realizamos una OR bit a bit entre las dobles palabras del signo (ACU1) y dato sin signo (ACU2). Esto significa que el bit más a la izquierda, que en formato REAL es el signo, se tomará de ACU1 y el resto de los bits (que en ACU1 son 0) se cogerán de ACU2 (dato sin signo).

#### **Autores:**

Moisés Pérez

Ignacio Latorre