MULTIPLE MOTORS CONTROL THROUGH CAN CLM-301

Description

Module used to control two SBS motors (step by step, for the horizontal and vertical movement of the knife) and a brushless (torque, for the turn of the knife) through CAN communication, being able the use of different working ways (asynchronous, synchronous, pulse counting) as well as the control of acceleration and deceleration ramps.

- Allows horizontal synchronized movement of several equipments.
- Allows knife vertical position measurement to control the penetration of it.
- Has inputs and outputs for general purpose

Application

Its principal application is found in the longitudinal cut of heavy machinery in the paper industry: CLM = Motorized, Longitudinal, Cut

Additional Data

✓ Synchronism horizontal movement between top and low knife.
✓ Knifes turning speed control through PMSM motor (permanent magnet synchronous motor).
✓ Modules addressing through micro-switch.
✓ SW adapted for graphical visualization of records and change of version through RS232.
✓ Inputs with supply for extern detectors and differentials encoders.
✓ Outputs for devices control
✓ Knife penetration auto-adjustment through potentiometer reference.
✓ Alarms and devices state signals with led diodes.

Common data

To govern a longitudinal arm, the CLM-301 module has to control:

✓ PMSM motor for knife turn.
✓ SBS motors for horizontal and vertical movement of the knife.
✓ Potentiometer for knife penetration reference.
✓ Four stages
  1: photoelectric cell for knife diameter calculation.
  2: general purpose.
✓ Two outputs:
  1: Acts on the electro-valve that presses the top knife against the low one.
  2: general purpose.

<table>
<thead>
<tr>
<th>Dimensions(mm)</th>
<th>Weight (Kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>Width</td>
</tr>
<tr>
<td>Total necessary area</td>
<td>155</td>
</tr>
<tr>
<td>Case</td>
<td>135</td>
</tr>
</tbody>
</table>
The CLM-301 control is done through the OPENCAN, hanging from a CANMASTER as many modules as necessary. Each CLM-301 uses 4 gates of the CAN controller, must not overload the line. Each module addressing is done with a micro-switch according to selected code, beginning with the switch of less weight (CODE1).

According to standard OPENCAN, the gate definition comes given by…

<table>
<thead>
<tr>
<th>ID</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPENCAN Standard + switch</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>bytes</td>
</tr>
</tbody>
</table>

…being switch the addressing to each CLM-301 module.

The gates addresses and the frames composition of the CLM-301 module seen from the CODESYS programming language is the following:

Each CLM-301 module has a glazed window on the upper side of its case for leds visualization. These leds show all devices state, according to following table code.

<table>
<thead>
<tr>
<th>LED</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WD</td>
<td>(Watch dog) flashing every 1 second</td>
<td></td>
</tr>
<tr>
<td>CAN</td>
<td>Communication</td>
<td>FLASHING</td>
</tr>
<tr>
<td>RS-232</td>
<td>Reception/Transmission</td>
<td></td>
</tr>
<tr>
<td>PMSM</td>
<td>motor.</td>
<td>OFF=STOP ON=START FLASHING=ALARM (*)</td>
</tr>
<tr>
<td>Horizontal</td>
<td>mov. SBS motor</td>
<td>OFF=STOP ON=START FLASHING=ALARM</td>
</tr>
<tr>
<td>Vertical</td>
<td>mov. SBS motor</td>
<td>OFF=LOW ON=MIDDLE FLASHING=HIGH</td>
</tr>
</tbody>
</table>

(*) The alarm indicated by the LED4 can have different meanings for a PMSM or a SBS motor. See them detailed in the part of this document that refers to the above mentioned motor.
The control of this motor is done in closed speed loop. Just the first time that the motor starts (after a reset of the CPU), the motor turns in open loop until it finds the position of the “0” angle of the stator poles. Once the motor is positioned it will turn taking as a reference the “0” position of the calculated angle. The configuration values of the PID loops, alarms... are customized by default, being able to modify them from the CODESYS programming language. The control records and the motor visualization are located on the “processmap” and depend on the ID (Switch number).

<table>
<thead>
<tr>
<th>Switch</th>
<th>Processmap del Codesys (escritura en CLM-301)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PMSM_01_CONTROL AT %QW2572: WORD</td>
</tr>
<tr>
<td>2</td>
<td>PMSM_01_CONSIGNAVELO AT %QW2573: WORD</td>
</tr>
<tr>
<td>3</td>
<td>PMSM_02_CONTROL AT %QW2584: WORD</td>
</tr>
<tr>
<td></td>
<td>PMSM_02_CONSIGNAVELO AT %QW2585: WORD</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>PMSM_03_CONTROL AT %QW2596: WORD</td>
</tr>
<tr>
<td>2</td>
<td>PMSM_03_CONSIGNAVELO AT %QW2597: WORD</td>
</tr>
</tbody>
</table>

The records have the speed “consigna” in encoder impulses/second, which is 2048 pulses/turn. It is possible to accede to the records from the CAN reception and transmission buffer (RXPDO, TXPDO).

The records have the speed “consigna” in encoder impulses/second, which is 2048 pulses/turn. It is possible to accede to the records from the CAN reception and transmission buffer (RXPDO, TXPDO).

### MOTOR PMSM (brushless / torque)

<table>
<thead>
<tr>
<th>Switch</th>
<th>Processmap del Codesys (lectura del CLM-301)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PMSM_01_ESTADO_ALARMA AT %IW2572: WORD</td>
</tr>
<tr>
<td>2</td>
<td>PMSM_01_REALVEL AT %IW2573: WORD</td>
</tr>
<tr>
<td>3</td>
<td>PMSM_02_ESTADO_ALARMA AT %IW2584: WORD</td>
</tr>
<tr>
<td></td>
<td>PMSM_02_REALVEL AT %IW2585: WORD</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>PMSM_03_ESTADO_ALARMA AT %IW2596: WORD</td>
</tr>
<tr>
<td>2</td>
<td>PMSM_03_REALVEL AT %IW2597: WORD</td>
</tr>
</tbody>
</table>

### Códigos alarma PMSM (led 4 display parpadeando)

<table>
<thead>
<tr>
<th>Valor</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sobrecorriente de pico rama R</td>
</tr>
<tr>
<td>2</td>
<td>Sobrecorriente de pico rama S</td>
</tr>
<tr>
<td>3</td>
<td>Sobrecorriente de pico rama T</td>
</tr>
<tr>
<td>4</td>
<td>Sobrecorriente rms rama R</td>
</tr>
<tr>
<td>5</td>
<td>Sobrecorriente rms rama S</td>
</tr>
<tr>
<td>6</td>
<td>Sobrecorriente rms rama T</td>
</tr>
<tr>
<td>7</td>
<td>Alarma módulo de potencia (IGBT)</td>
</tr>
<tr>
<td>8</td>
<td>Alarma sonda temperatura (KTY) ALARM &gt; 95°C</td>
</tr>
<tr>
<td>9</td>
<td>Alarma tensión baja de bus</td>
</tr>
<tr>
<td>10</td>
<td>Alarma error en la comunicación CAN</td>
</tr>
<tr>
<td>11</td>
<td>Alarma conexión antínea encoder</td>
</tr>
</tbody>
</table>
Control loops records configuration

In the CODESYS block, in the MTC-3038 module properties exist the “Editable parameters” flap, where some motor control loops parameters can be modified. They would correspond with slow access record SDO (CMS block of CAN standard), so they are send just once (initial configuration).

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>COMMENT</th>
<th>FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current sensor adjust IR</td>
<td>Registers for IR current measurement adjustment.</td>
<td></td>
</tr>
<tr>
<td>Current sensor adjust IS</td>
<td>Registers for IS current measurement adjustment.</td>
<td></td>
</tr>
<tr>
<td>Note: the IT current measurement is done matematically</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear sensor Poten. Adjust</td>
<td>Register for the adjustment of the measurement that the potentiometer gives, which indicates in hundreth the height of the knife</td>
<td>Q12</td>
</tr>
<tr>
<td>Reak current limit</td>
<td>Current peak so that alarm goes off. 1000H=5A.</td>
<td>Q12</td>
</tr>
<tr>
<td>RMS Current limir</td>
<td>RMS Current so that alarm goes off.</td>
<td>Q12</td>
</tr>
<tr>
<td>Maximum speed ramp</td>
<td>Value that a slope generate us if the speed value produces a sudden change or jump.</td>
<td></td>
</tr>
<tr>
<td>Speed filter</td>
<td>Filter in the speed measurement.</td>
<td></td>
</tr>
<tr>
<td>Speed PID</td>
<td>Kp= proportional constant speed loop.</td>
<td></td>
</tr>
<tr>
<td>Speed PID</td>
<td>Ki= integrative constant speed loop.</td>
<td></td>
</tr>
<tr>
<td>Torque PID</td>
<td>Kp= proportional constant speed loop.</td>
<td></td>
</tr>
<tr>
<td>Torque PID</td>
<td>Ki= integrative constant speed loop.</td>
<td></td>
</tr>
<tr>
<td>Magnetizing current reference</td>
<td>Reference of the magnetizing current.</td>
<td></td>
</tr>
</tbody>
</table>

PMSM module configuration record.

Through RS-232 communication with its respective equipment, the addresses in upper graph can tested.
The horizontal movement of the knife is controlled by a SBS motor. This motor can start in synchronous way, starting the upper and lower knife at the same time with the synchronous order. The motor can start doing a torque stopped.

**Working ways**

**Asynchronous**
Starts in an individual way with a ramp from a minimum frequency to a maximum. The time of raise is given in tenths of second.

**Deceleration**
It is only applied in the asynchronous way. Goes from a maximum frequency to a minimum one. The deceleration time is the same as the acceleration time.

**Pulses counting**
The number of pulses that the knives want to be moved has to be indicated, the way itself generates the acceleration and deceleration ramp from the maximum and minimum frequency. If there are not many pulses, it is possible that the maximum frequency is not reached.

**Synchronous**
This way allows starting several motors at the same time (example: pair of knives upper and lower). Firstly the motors that want to be started are kept with the order "start", when a common signal reach to all the equipments “synchronization order” all of them start at the same time. The stop can also be in synchronous way.

**Synchronous way starting process**
- Send to the records control of the motors that want to be started, the value “CONTROL_SBS=0x010Eh”.
- Verify which state record has located the value “ESTADO_SBS=0x010Eh”.
- Send “TRUE” to the system value “MTC3038_SINCRO(BOOL)=TRUE”. Motors start at the same time.
The visualization records and the horizontal movement motor control are located in the "processmap".

MSW shares word with outputs.

It is possible to access to the records from CAN reception and transmission buffer (RXPDO, TXPDO).

Common register to all the modes. When starting it sets to 0 until the number of pulses programmed. MSW shares word with

<table>
<thead>
<tr>
<th>Switch</th>
<th>Processmap del Codesys (nomenclature of CLM-301)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SBS_01_HORIZONTAL_CONTROL AT%W2574: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_HORIZONTAL_FRE_MAX  AT%W2574: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_HORIZONTAL_FRE_MIN  AT%W2574: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_HORIZONTAL_RAMPA    AT%W2574: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_HORIZONTAL_REF_PULSO_LSW 24 bits AT%W2576: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_HORIZONTAL_REF_PULSO_MSW AT%W2577: WORD</td>
</tr>
<tr>
<td>2</td>
<td>SBS_02_HORIZONTAL_CONTROL AT%W2586: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_HORIZONTAL_FRE_MAX  AT%W2586: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_HORIZONTAL_FRE_MIN  AT%W2586: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_HORIZONTAL_RAMPA    AT%W2586: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_HORIZONTAL_REF_PULSO_LSW 24 bits AT%W2588: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_HORIZONTAL_REF_PULSO_MSW AT%W2589: WORD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switch</th>
<th>Processmap del Codesys (reading of CLM-301)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SBS_01_HORIZONTAL_REAL_PULSO_LSW 24 bits AT%W2576: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_HORIZONTAL_REAL_PULSO_MSW AT%W2577: WORD</td>
</tr>
<tr>
<td>2</td>
<td>SBS_02_HORIZONTAL_REAL_PULSO_LSW 24 bits AT%W2586: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_HORIZONTAL_REAL_PULSO_MSW AT%W2587: WORD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Hex</th>
<th>Horizontal SBS control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000h</td>
<td>Direction</td>
</tr>
<tr>
<td>1</td>
<td>0002h</td>
<td>Torque 1 (**)</td>
</tr>
<tr>
<td>2</td>
<td>0004h</td>
<td>Torque 2</td>
</tr>
<tr>
<td>3</td>
<td>0008h</td>
<td>Torque with stopped motor</td>
</tr>
<tr>
<td>4</td>
<td>0010h</td>
<td>Asynchronous start</td>
</tr>
<tr>
<td>5</td>
<td>0020h</td>
<td>Asynchronous deceleration</td>
</tr>
<tr>
<td>6</td>
<td>0040h</td>
<td>Asynchronous stop</td>
</tr>
<tr>
<td>7</td>
<td>0080h</td>
<td>Start with pulses</td>
</tr>
<tr>
<td>8</td>
<td>0100h</td>
<td>Synchronous start</td>
</tr>
<tr>
<td>9</td>
<td>0200h</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0400h</td>
<td>SBS synchronous stop</td>
</tr>
<tr>
<td>11</td>
<td>0800h</td>
<td>Stop by pulses</td>
</tr>
<tr>
<td>12</td>
<td>1000h</td>
<td>Reset alarm</td>
</tr>
<tr>
<td>13</td>
<td>2000h</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4000h</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8000h</td>
<td></td>
</tr>
</tbody>
</table>

**(*)** "Multiplexando" control bit 1 and 2 it is obtained...

<table>
<thead>
<tr>
<th>Bit</th>
<th>Hex</th>
<th>Horizontal SBS control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000h</td>
<td>Address</td>
</tr>
<tr>
<td>1</td>
<td>0002h</td>
<td>Torque1</td>
</tr>
<tr>
<td>2</td>
<td>0004h</td>
<td>Torque2</td>
</tr>
<tr>
<td>3</td>
<td>0008h</td>
<td>Torque with motor stopped</td>
</tr>
<tr>
<td>4</td>
<td>0010h</td>
<td>Asynchronous start</td>
</tr>
<tr>
<td>5</td>
<td>0020h</td>
<td>Deceleration</td>
</tr>
<tr>
<td>6</td>
<td>0040h</td>
<td>Asynchronous stop</td>
</tr>
<tr>
<td>7</td>
<td>0080h</td>
<td>Start with pulses</td>
</tr>
<tr>
<td>8</td>
<td>0100h</td>
<td>Synchronous start</td>
</tr>
<tr>
<td>9</td>
<td>0200h</td>
<td>Synchronous turning</td>
</tr>
<tr>
<td>10</td>
<td>0400h</td>
<td>SBS synchronous stop</td>
</tr>
<tr>
<td>11</td>
<td>0800h</td>
<td>Stop by pulses</td>
</tr>
<tr>
<td>12</td>
<td>1000h</td>
<td>Reset alarm</td>
</tr>
<tr>
<td>13</td>
<td>2000h</td>
<td>Power alarm</td>
</tr>
<tr>
<td>14</td>
<td>4000h</td>
<td>CAN alarm</td>
</tr>
</tbody>
</table>

Example bits 15 and 14 of the SBS1 control register:

1. Access motor control
   - Starting order asynchronous to max. torque
   
   \[
   SBS = 0x0016 \ (hex) = 0000 0000 0001 0110 \ (bin) \]
2. Access final frequency
   - Frequency of 5000Hz
   
   \[
   SBS = 0x5388 \ (hex) = 0101 0011 1000 1000 \ (bin) \]
3. Access initial frequency
   - Min. frequency of 50Hz
   
   \[
   SBS = 0x8032 \ (hex) = 1000 0000 0011 0010 \ (bin) \]
4. Access to acceleration slope
   - Acceleration slope 100dsg
   
   \[
   SBS = 0xC064 \ (hex) = 1100 0000 0011 0100 \ (bin) \]
The vertical movement of the knife is controlled by SBS motor. This motor does not have synchronous starting mode. Has automatically photoelectric cell stop, being able to calculate the 0 point of the knife.

**Working modes**

**Asynchronous**
Starts in an individual way with a ramp from a minimum frequency to a maximum. The time of raise is given in tenths of second.

**Deceleration**
It is only applied in the asynchronous way. Goes from a maximum frequency to a minimum one. The deceleration time is the same as the acceleration time.

**Pulses counting**
The number of pulses that the knives want to be moved has to be indicated, the way itself generates the acceleration and deceleration ramp from the maximum and minimum frequency. If there are not many pulses, it is possible that the maximum frequency is not reached.

**Photocell**
The motor starts and remains at the minimal frequency, until stop because of photocell detection.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Codesys Processmap (nomenclature in CLM-301)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SBS_01_VERTICAL_CONTROL AT %QW2575: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_VERTICAL_FRE_MAX AT %QW2575: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_VERTICAL_FRE_MIN AT %QW2575: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_VERTICAL_RAMPA AT %QW2575: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_VERTICAL_REF_PULSOS 16 bits AT %QW2579: WORD</td>
</tr>
<tr>
<td>2</td>
<td>SBS_02_VERTICAL_CONTROL AT %QW2587: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_VERTICAL_FRE_MAX AT %QW2587: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_VERTICAL_FRE_MIN AT %QW2587: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_VERTICAL_RAMPA AT %QW2587: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_VERTICAL_REF_PULSOS 16 bits AT %QW2591: WORD</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Switch</td>
<td>Codesys Processmap (Reading of CLM-301)</td>
</tr>
<tr>
<td>1</td>
<td>SBS_01_VERTICAL_ESTADO AT %IW2575: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_01_VERTICAL_REAL_PULSOS 16 bits AT %IW2579: WORD</td>
</tr>
<tr>
<td>2</td>
<td>SBS_02_VERTICAL_ESTADO AT %IW2587: WORD</td>
</tr>
<tr>
<td></td>
<td>SBS_02_VERTICAL_REAL_PULSOS 16 bits AT %IW2591: WORD</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The visualization records and the motor control are located in the processmap.

It is possible to access to the records from CAN reception and transmission buffer (RXPDO, TXPDO).

Common register to all modes.
1.0 08/08/2008

MULTIPLE MOTORS CONTROL THROUGH CAN CLM-301

MONTELEC S.L.

(*a) “Multiplexando” control bit 1 and 2 it is obtained …

SBSx_CONTROL (Bits 2 and 1)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Hex</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000h</td>
<td>Direction</td>
</tr>
<tr>
<td>1</td>
<td>0002h</td>
<td>Torque 1</td>
</tr>
<tr>
<td>2</td>
<td>0004h</td>
<td>Torque 2</td>
</tr>
<tr>
<td>3</td>
<td>0008h</td>
<td>Torque with stopped motor</td>
</tr>
<tr>
<td>4</td>
<td>0010h</td>
<td>Asynchronous start</td>
</tr>
<tr>
<td>5</td>
<td>0020h</td>
<td>Asynchronous deceleration</td>
</tr>
<tr>
<td>6</td>
<td>0040h</td>
<td>Asynchronous stop</td>
</tr>
<tr>
<td>7</td>
<td>0080h</td>
<td>Start with pulses</td>
</tr>
<tr>
<td>8</td>
<td>0100h</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0200h</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0400h</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0800h</td>
<td>Stop by pulses</td>
</tr>
<tr>
<td>12</td>
<td>1000h</td>
<td>Reset alarm</td>
</tr>
<tr>
<td>13</td>
<td>2000h</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4000h</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8000h</td>
<td></td>
</tr>
</tbody>
</table>

(*b) The registers “control”, “max. frequency”, “min. frequency” and slope share the same address in the Codesys. “Multiplexando” the SBS1 control bit 15 and 16 it is obtained…

SBSx_CONTROL (Bits 15 and 14)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Hex</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000h</td>
<td>Address</td>
</tr>
<tr>
<td>1</td>
<td>0002h</td>
<td>Torque1</td>
</tr>
<tr>
<td>2</td>
<td>0004h</td>
<td>Torque2</td>
</tr>
<tr>
<td>3</td>
<td>0008h</td>
<td>Torque with motor stopped</td>
</tr>
<tr>
<td>4</td>
<td>0010h</td>
<td>Asynchronous start</td>
</tr>
<tr>
<td>5</td>
<td>0020h</td>
<td>Deceleration</td>
</tr>
<tr>
<td>6</td>
<td>0040h</td>
<td>Asynchronous stop</td>
</tr>
<tr>
<td>7</td>
<td>0080h</td>
<td>Start with pulses</td>
</tr>
<tr>
<td>8</td>
<td>0100h</td>
<td>Start with photocell</td>
</tr>
<tr>
<td>9</td>
<td>0200h</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0400h</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0800h</td>
<td>Stop by pulses</td>
</tr>
<tr>
<td>12</td>
<td>1000h</td>
<td>Reset alarm</td>
</tr>
<tr>
<td>13</td>
<td>2000h</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4000h</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8000h</td>
<td>CAN alarm</td>
</tr>
</tbody>
</table>

Procedure to measure the knife diameter.

As the work with a photocell is really slow this procedure is done to speed up the work. First of all a quick approximation is done in asynchronous way.

1. Raise the knife in asynchronous way and be looking at the input of the photoelectric cell until it detects it, once introduced stop. (Point2)
2. Lower the knife in asynchronous way until the photocell stops detecting and stop. (Point3)
3. Start the knife in photocell way. This way the knife will stop as soon as it detects the photocell, giving a precise adjustment of the same one. (Point4)

\[ \text{Procedure to measure the knife diameter.} \]

\[ \text{As the work with a photocell is really slow this procedure is done to speed up the work. First of all a quick approximation is done in asynchronous way.} \]

\[ \text{1. Raise the knife in asynchronous way and be looking at the input of the photoelectric cell until it detects it, once introduced stop. (Point2)} \]

\[ \text{2. Lower the knife in asynchronous way until the photocell stops detecting and stop. (Point3)} \]

\[ \text{3. Start the knife in photocell way. This way the knife will stop as soon as it detects the photocell, giving a precise adjustment of the same one. (Point4)} \]
The module has four inputs: one for the photocell and three of general purpose (IN2-IN3-IN4). In the machines where the vertical movement is manual, the inputs IN3-IN4 will be assigned to the raising or descending movement of the knife IN3=KNIFE UP and IN4=KNIFE DOWN.

The state of the inputs is in the high part of each of the “CODESYS” registers.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Codesys processmap (CLM-301 writing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INPUTS_01 AT %IW2576: WORD</td>
</tr>
<tr>
<td>2</td>
<td>INPUTS_02 AT %IW2588: WORD</td>
</tr>
<tr>
<td>3</td>
<td>INPUTS_03 AT %IW2600: WORD</td>
</tr>
<tr>
<td>4</td>
<td>INPUTS_04 AT %IW2612: WORD</td>
</tr>
<tr>
<td>5</td>
<td>INPUTS_05 AT %IW2624: WORD</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

General Prog.  
Photocell

The module has two outputs: one for the electro-valve and the other for general purpose.

The state of the outputs is in the high part of each of the “CODESYS” registers.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Codesys processmap (CLM-301 writing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OUTPUTS_01 AT %IW2576: WORD</td>
</tr>
<tr>
<td>2</td>
<td>OUTPUTS_02 AT %IW2588: WORD</td>
</tr>
<tr>
<td>3</td>
<td>OUTPUTS_03 AT %IW2600: WORD</td>
</tr>
<tr>
<td>4</td>
<td>OUTPUTS_04 AT %IW2612: WORD</td>
</tr>
<tr>
<td>5</td>
<td>OUTPUTS_05 AT %IW2624: WORD</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

General Prog.  
Electro-valve

It is possible to accede the registers from the CAN reception and transmission buffers.