Application Examples with PLCopen Motion Control

PLCopen has specified a number of specifications for Motion Control. Based on the defined function blocks two typical application examples are presented here. For details on the six specifications for Motion Control refer to the website www.PLCopen.org under TC2 Function Blocks.

Example 1: Label machine

Application description

The task is to place a label at a particular position on a product.The application has two drives, one to feed the product via a conveyor belt, the other to feed the labels and to place the labels on the products. The labeling process is triggered by a position detection sensor. From the detection of the product to the start of the label movement there is a delay depending on the velocity of the conveyor, the position of the sensor and the position of the label on the product.

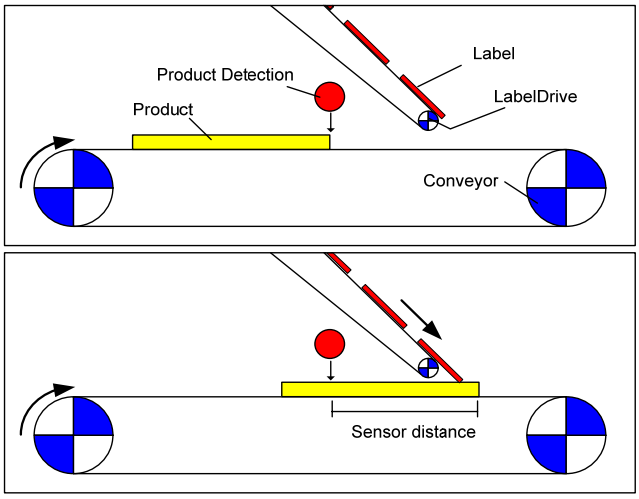

Figure 1: Labeling machine

Programming example

This example shows a way to solve this task.
Both axes move with the same velocity setpoint. The delay for TON is calculated from the sensor distance and the velocity. After a labeling step the LabelDrive stops again and waits for the next trigger, while the conveyor continuously moves.

Possible Improvements

Although this principle is working, there are some possibilities to improve the functionalities and performance to achieve faster and more precise machines. Ways to do this are:

- Compensate for the drift of the label position as a result of the sum of incremental errors.
- A fast touch-probe input to detect the start position of the product more precisely
- A MC_CamIn or MC_GearInPos function to synchronize the label and product position in order to position the label more exact on the product. The conveyor should be the Master axis and the LabelDrive the Slave axis. In this way a mismatch caused by acceleration of the conveyor during labeling can be avoided.
- If the product is smaller than the sensor distance a kind of FIFO for product tracking can be necessary. See for instance the FIFO function block as specified in Part 3 – User Guidelines.

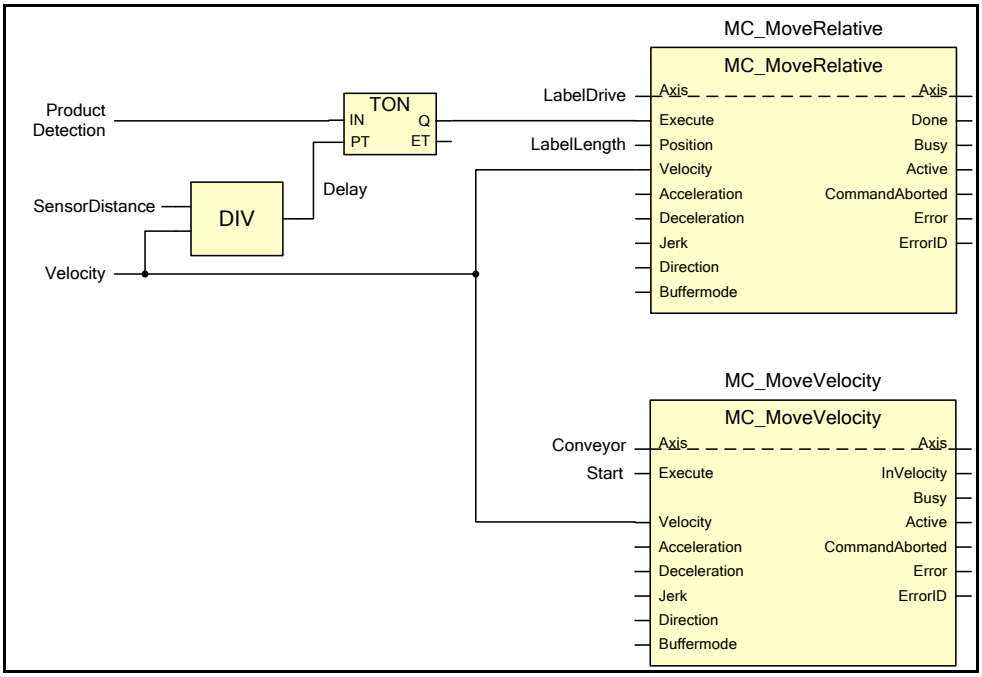

Figure 2: Program example for labeling machine

Example 2: Warehousing

Application description

The purpose of this application is to automatically retrieve goods from a storage cabinet with shelves surface. The goods are stored in pallets that can be retrieved with a fork system.
The warehouse task is to move the fork with three axes to place or take the pallet:

- Axis X moves along the floor;
- Axis Y moves to the needed height;
- Axis Z moves the fork into the shelf to fetch the pallet.

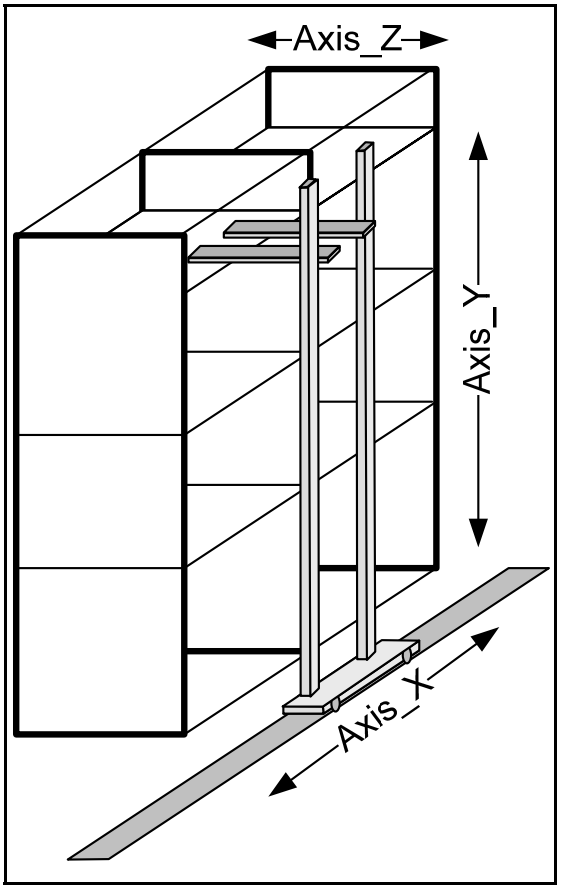

Figure 3: Warehousing example

The sequence is to move the axes X and Y to the requested position. As soon as both axes have reached this position, the Z axis moves into the shelf under the pallet, in this example for 1000 mm. Then the Y axis lifts the pallet for another 100 mm to lift the pallet from the shelf, so it can be moved out of the shelf and to the required position to deliver it.

This example can be implemented in different ways. A straightforward approach is to use Part 1 Function Blocks. Alternatively, XYZ group could be defined in controllers supporting PLCopen Part 4, Coordinated Motion, which can simplify and optimize the movements.

First programming example (using Part 1)

This version could be implemented in the following way by only using function blocks from Part 1.

Timing diagram

The following graphics show the sequence to fetch a pallet from the storage system.

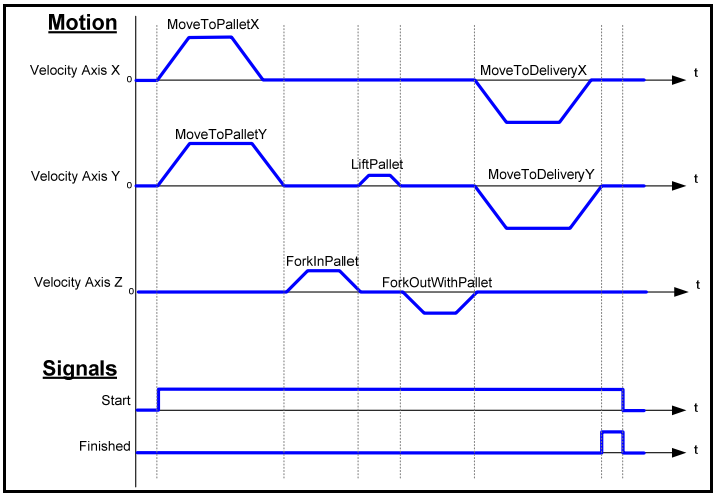

Figure 4: Timing Example

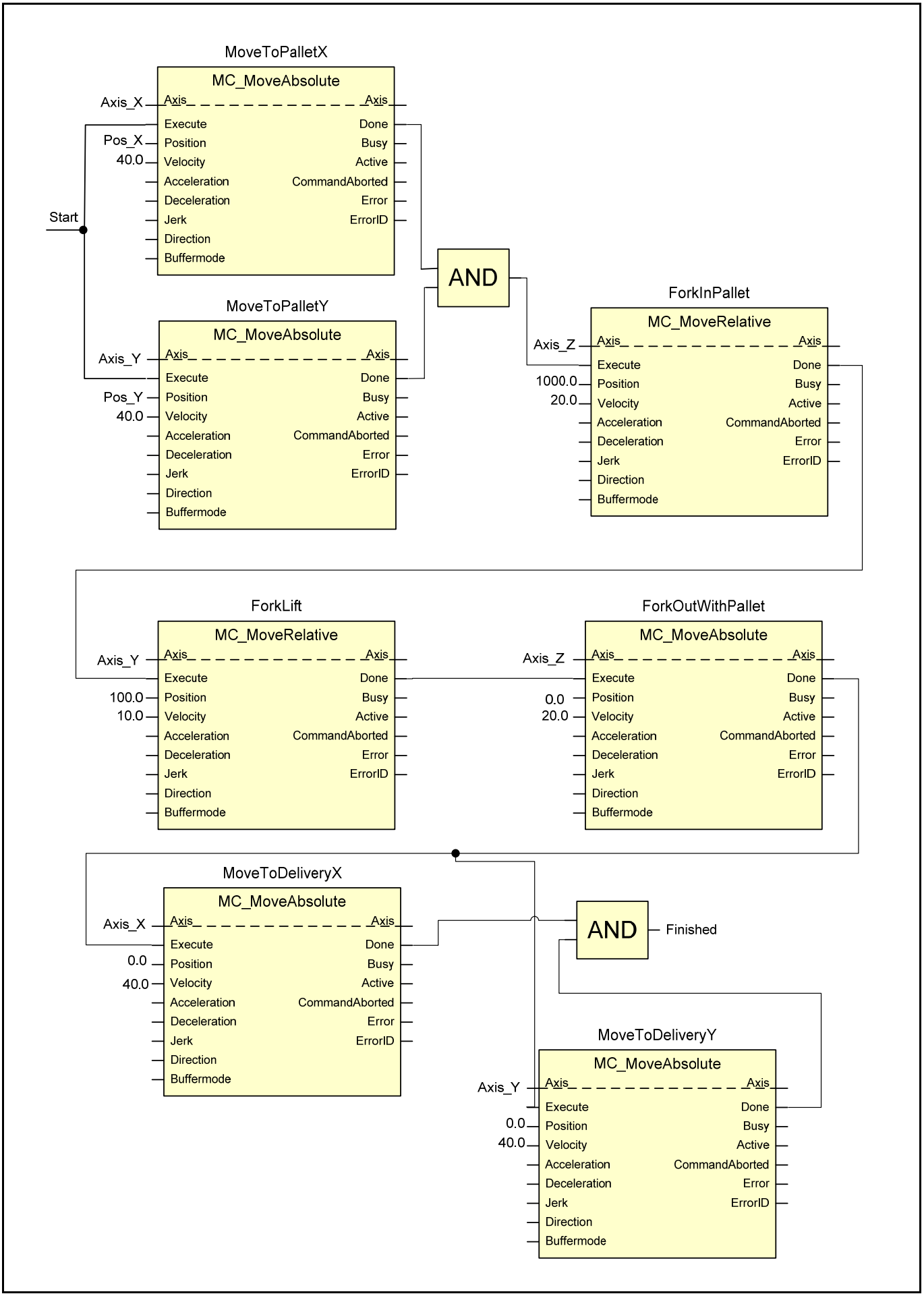

Figure 5: First Program for warehousing example
Note: not all the specified inputs are shown in FBs above.

Second programming example (using Part 4)

This version is implemented using coordinated motion commands from PLCopen Part 4, Coordinated Motion. For information on how to create Axis Groups and enable them for coordinated operation, refer to chapter 4.1 of Part 4 specification. In this example, the group XYZLifter is made up from Axis_X, Axis_Y and Axis_Z. Blending is used to optimize the approach time to the end positions (Fork Axis_Z do not have to wait completion of Axis_X and Axis_Y movement to enter the pallet). This has to be done with a "TMCornerDistance" method to avoid collision with the shelf (in this particular case, we assume the distance from fork to shelf is bigger than 100). For this cornering to become effective, the 'Busy' output of precedent Function is triggering the 'Execute' of the buffered movement.

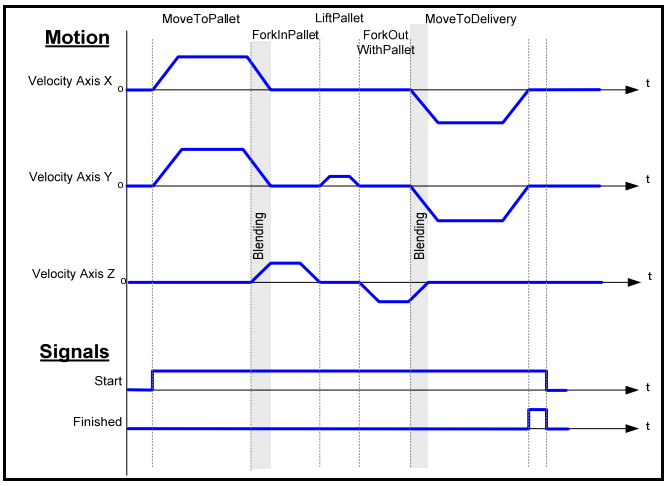

Figure 6: Timing Example

Timing diagram

The following figure shows the sequence with the use of Part 4 function blocks. Blending between movements provides optimization.

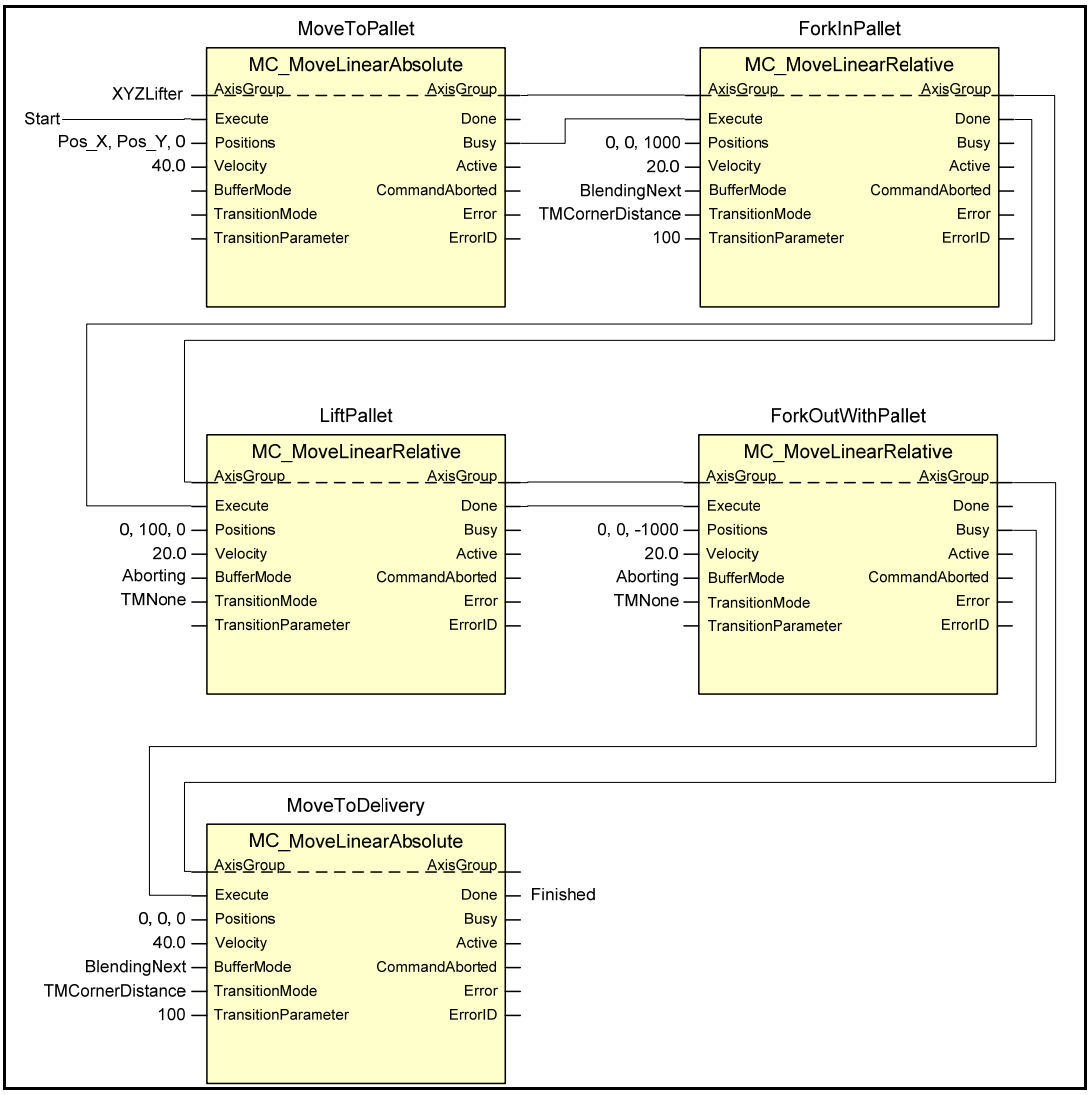

Figure 7: Second program example for warehousing

PLCopen, as an organization active in industrial control, is creating a higher efficiency in your application software development. With results like Motion Control Library, Safety, XML specification, Communication with OPC UA, Reusability Level and Conformity Level. PLCopen made solid contributions to the community, extending the hardware independence from the software code, as well as reusability of the code and coupling to external software tools. Based on IEC 61131-3, the only global standard for industrial control programming, we help people with different backgrounds and skills to create different elements of a program during different stages of the software lifecycle. Yet all pieces adhere to a common structure and work together harmoniously.

More information on www.PLCopen.org