SIEMENS

# Learn-/Training Document

Siemens Automation Cooperates with Education (SCE) | From NX MCD V12/TIA Portal V15.0

**DigitalTwin@Education Module 150-003**
Enhancements and Optimizations of an Automation Program for a 3D Model

siemens.com/sce

SIEMENS
Global Industry Partner of WorldSkills International

worldskills

## Matching SCE trainer packages for this Learn-/Training Document

**SIMATIC STEP 7 Software for Training (inkl. PLCSIM Advanced)**

- **SIMATIC STEP 7 Professional V15.0 - Single License**
  Order no.: 6ES7822-1AA05-4YA5
- **SIMATIC STEP 7 Professional V15.0 - Classroom License for 6 Users**
  Order no.: 6ES7822-1BA05-4YA5
- **SIMATIC STEP 7 Professional V15.0 - Upgrade License for 6 Users**
  Order no.: 6ES7822-1AA05-4YE5
- **SIMATIC STEP 7 Professional V15.0 - Student License for 20 Users**
  Order no.: 6ES7822-1AC05-4YA5

**SIMATIC WinCC Engineering/Runtime Advanced software in the TIA Portal**

- **SIMATIC WinCC Advanced V15.0 - Classroom License for 6 Users**
  6AV2102-0AA05-0AS5
- **Upgrade SIMATIC WinCC Advanced V15.0 - Classroom License for 6 Users**
  6AV2102-4AA05-0AS5
- **SIMATIC WinCC Advanced V15.0 - Student License for 20 Users**
  6AV2102-0AA05-0AS7

**NX V12.0 Educational Bundle (Schools, universities, not for in-company training centers)**

- **Contact**: academics.plm@siemens.com

## Additional information regarding SCE
siemens.com/sce

## Information regarding use
The SCE Learn-/Training Document for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational and R&D institutions. Siemens does not guarantee the contents.

This document is only to be used for initial training on Siemens products/systems. This means it can be copied in whole or in part and given to trainees/students for use within the scope of their training/course of study. Disseminating or duplicating this document and sharing its content is permitted within public training and advanced training facilities for training purposes or as part of a course of study.

Exceptions require written consent from Siemens. Send all related requests to scesupportfinder.i-ia@siemens.com.

We wish to thank the HS Darmstadt, particularly Mr. Heiko Webert and Mr. Prof. Dr.-Ing. Stephan Simons, and all other persons involved for their support in the preparation of this SCE learning/training document.

# Table of contents

# List of figures

# List of tables

# Enhancements and Optimizations of an Automation Program for a 3D Model

## 1 Goal

In this module, expansion options and optimizations of the automation program explained and created in Module 1 and Module 2 are presented. This module also includes a discussion of possible solutions.

The changes relate exclusively to the automation program in this module. Adaptation of the dynamic 3D model is not considered in detail here.

## 2 Requirement

The requirements from Module 1 and Module 2 are still valid.

You should have knowledge from the **Basics of PLC Programming in the TIA Portal**, especially about the programming language **SCL**.

Since the PLC is simulated in this workshop using S7-PLCSIM Advanced, there are no hardware components for the controller in this module.

A further requirement is the completion of the first two modules of this training series. Otherwise, it is difficult to follow the extensions and optimizations described here.

# 3 Required hardware and software

The following components are required for this module:

1. **Engineering Station**: Requirements include hardware and operating system (for additional information: see Readme on the TIA Portal Installation DVDs, and in the NX software package)
2. **SIMATIC STEP 7 Professional software in TIA Portal** – V15.0 or higher
3. **SIMATIC WinCC Runtime Advanced software in TIA Portal** – V15.0 or higher
4. **SIMATIC S7-PLCSIM Advanced software** – V2.0 or higher
5. **NX software with Mechatronics Concept Designer add-on** – V12.0 or higher

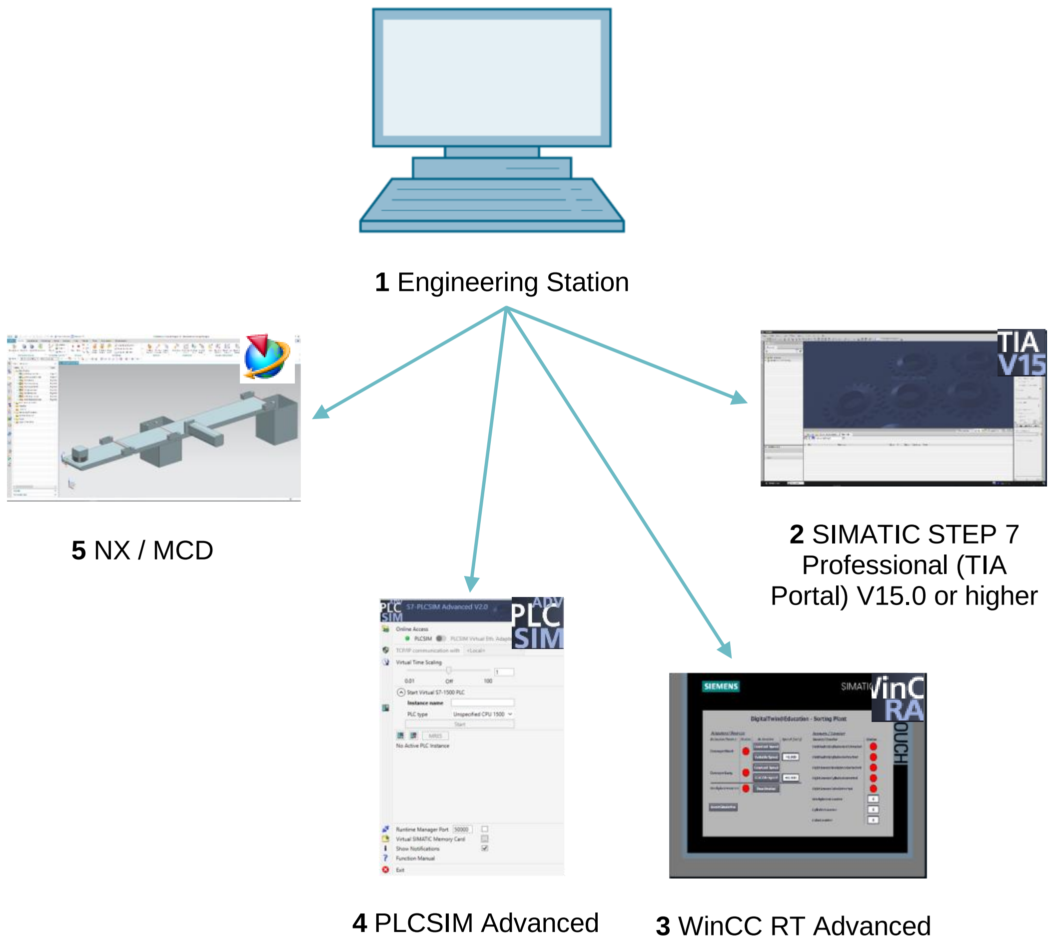


Figure 1: Overview of required software and hardware components for this module

As you can see from Figure 1, the engineering station is the only hardware component of the system. The remaining components are based exclusively on software.

# 4 Theory

With Module 1 from the DigitalTwin@Education Workshop series, you have already put the basic functions of the digital twin into operation. In Module 2, you could create your own automation program and test it on the existing dynamic 3D model. This module also describes and discusses a possible implementation of the automation program. However, further testing of the model will result in various inconsistencies or errors, some of which are explained below.

## 4.1 Error case 1: Incorrect sorting when conveyor belt speed is too low

In the test scenarios from Module 1, the conveyor belts were only moved at a constant speed or a variable speed of 50 %. In this case, no problems with sorting are apparent.

However, if you operate the conveyor belts with a motor speed of 5% or lower, the ejector sorts out both the "Cylinder" workpieces and the "Cube" workpieces into the first container. This phenomenon is illustrated in Figure 2.

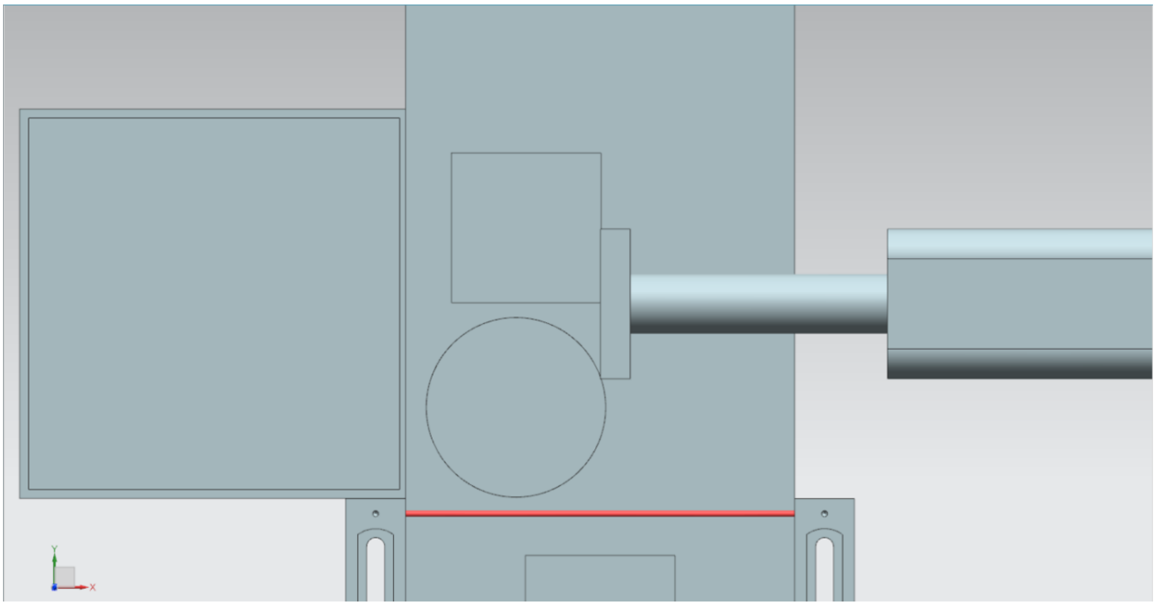

Figure 2: Incorrect sorting when conveyor belt speed is too low

As a result, the second container remains empty and sorting no longer functions properly.

This circumstance is explained by the specified waiting time until the ejection process begins. This is set at a constant 400 ms. To cover this situation correctly, the waiting time would have to be increased accordingly, which is not yet included in the basic function.

For this reason, at a much slower speed, not only the "Cylinder" cylindrical workpiece but also the preceding "Cube" cuboid workpiece is detected by the ejector arm and shoved from the belt.

## 4.2 Error case 2: Incorrect sorting when conveyor belt speed is too high

Another error occurs when the conveyor belts are moved at a variable speed of more than 66% of the maximum motor speed. The "Cylinder" cylindrical workpiece is no longer sufficiently touched by the ejector arm and is therefore not ejected from the belt. As shown in Figure 3, the "Cylinder" is even completely outside the ejection area.

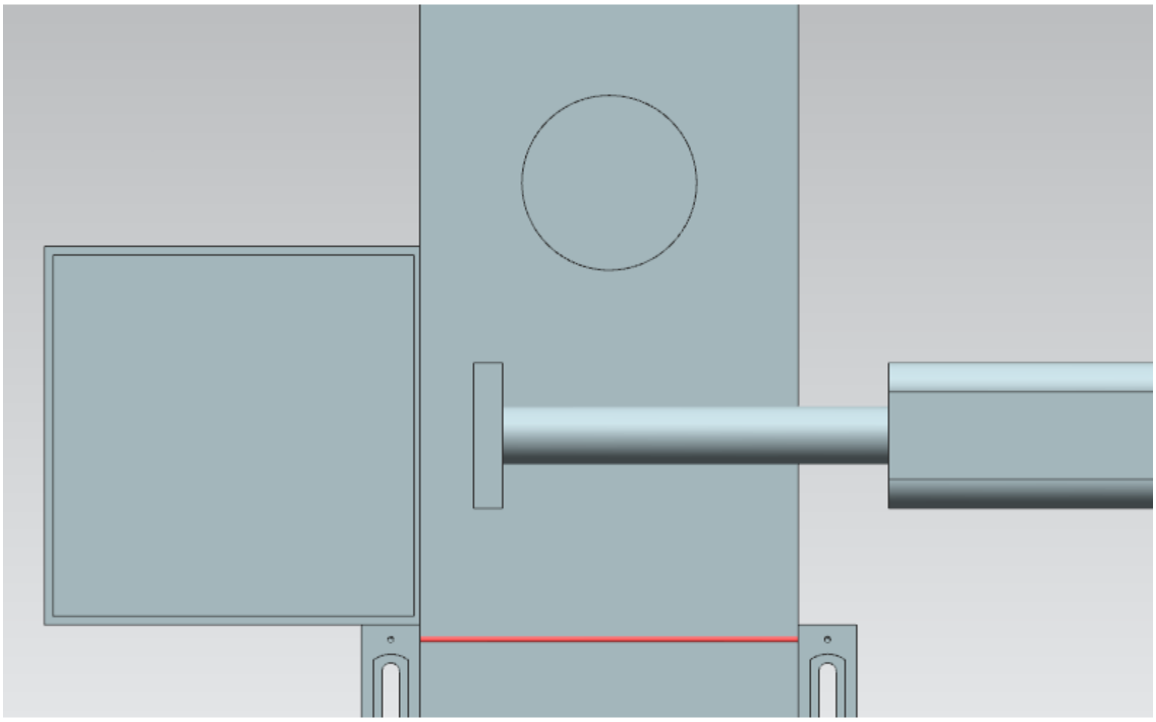

Figure 3: Incorrect sorting when conveyor belt speed is too high

As a result, the first container remains empty and both workpieces are sorted into the second container. Proper sorting consequently fails.

As already discussed in Chapter 4.1, the error here is also due to the constantly defined waiting time, from the detection of the "Cylinder" by the light sensor until the start of the ejection process. As a result, the waiting time would have to be considerably shorter than currently specified.

## 4.3 Proposed optimization 1: Sorting out while the conveyor belt is running

Module 2 of the DigitalTwin@Education Workshop series makes it clear that the automation program currently only implements sorting out when the conveyor belt is stationary. This requires stopping the conveyor belts and blocking the generation of new workpieces during the ejection process.

If, however, one considers a real system, sorting processes are usually performed during operation, if possible, without stopping the conveyor belts. It should therefore also be possible in this model to eject cylindrical workpieces with running transport surfaces.

Stopping the conveyor belts and blocking the generation of workpieces is therefore no longer necessary, since congestion on the conveyor belts is no longer to be expected.

# 5 Task

In this module, you should eliminate the errors described in Chapter 4 on the basis of the project you have created in Module 2, and include the optimization proposals in the automation program.

On the one hand, this includes the adjustment of the waiting time depending on the conveyor belt speed specified by the user.

On the other hand, the static sorting method, with stopping and locking the conveyor belts, is to be converted into a dynamic sorting method. This optimization requires a dynamic adaptation of the waiting time to the current conveyor belt speed. Locking the conveyor belts and generating new workpieces is no longer necessary.

# 6 Planning

For this module, you should use your TIA project from Module 2. The descriptions in Chapter 7 refer to the project "**150-001_DigitalTwinAtEducation_TIAP_Basic**", which was already used in Module 1 and explained in detail in Module 2. Accordingly, you should transfer the planned changes from Chapter 5 to your own project. It is advisable to create a copy of your created TIA project for each change so that you can always access your functional basic project.

Both the PLC program and the HMI were created and configured in the **SIMATIC STEP 7 Professional V15.0** software. The PLC is simulated virtually using the **SIMATIC S7-PLCSIM Advanced V2.0** software. The HMI is simulated with the **SIMATIC WinCC Runtime Advanced V15.0** optional package. The virtual PLC and the simulated HMI are connected via the simulated Ethernet interfaces.

The digital twin was created using **Mechatronics Concept Designer V12.0**. The suitably configured signals are already connected to the inputs and outputs of the PLC. Note that changing the dynamic 3D model in MCD in this module is not intended. You can continue to use the MCD project "**150-001_DigitalTwinAtEducation_MCD_dynModel_Signals**" to test your solutions.

When creating new automation programs, also refer to the Programming and Standardization Guidelines, as well as the notes from Chapter 9 for reading and creating UML diagrams.

In Chapter 7, you will find a proposal for a modified automation program in which the errors have been eliminated and the proposed optimization has been incorporated.

# 7 Structured step-by-step instructions

This chapter explains the proposed solution for eliminating the errors from Chapter 4.1 and Chapter 4.2 and describes the implementation of the optimization from Chapter 4.3.

From this, you can get ideas for your own automation program, for example using the activity diagrams shown in this chapter.

You can find the project with the error solutions under the designation "**150-003_DigitalTwinAtEducation_TIAP_Basic_Extended_1**". Optimization is implemented in the project "**150-003_DigitalTwinAtEducation_TIAP_Basic_Extended_2**".

## 7.1 Adaptation of the waiting time for safe sorting out

In order to eliminate the problems described in Chapter 4.1 and Chapter 4.2, changes may only be made in FB "SortingPlantControl".

While the basic procedure for sorting out the cylindrical workpieces remains the same, here only the calculation of the waiting time changes. An activity diagram of the implemented changes is shown in Figure 4.

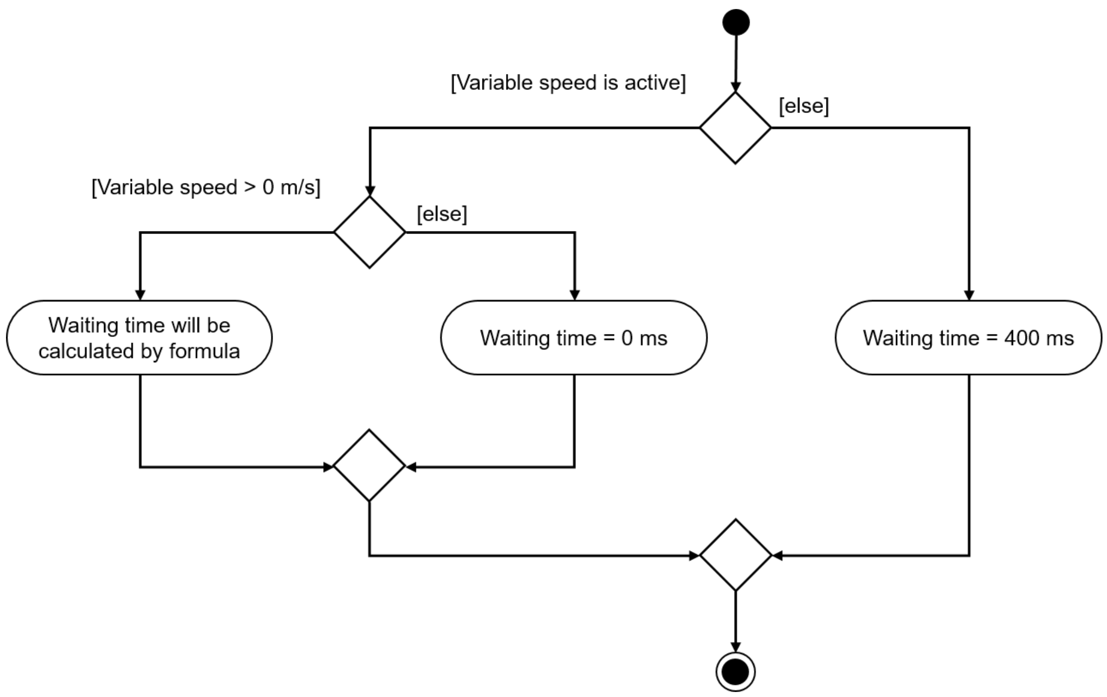


Figure 4: Activity diagram for the dynamic adjustment of the waiting time as a function of the conveyor belt speed.

As long as the "ConveyorLong" conveyor belt is to move at a steady speed, a constant waiting time of 400 ms is assumed. If, on the other hand, the "ConveyorLong" conveyor belt has the variable speed method activated, the waiting time is calculated taking into account the currently set target speed.

The following assumptions were made in the example solution provided:

- In order to ensure optimum ejection of the cylindrical body, the workpiece should be transported another 20 mm = 0.02 m after the negative flank of the "Cylinder" light sensor (see Figure 5).
- The running speed **v** in m/s is now used as the reference speed.
- This results in the following formula: $t_{Wait} = \frac{0.02\ m}{v\ [\frac{m}{s}]} x \frac{1000\ ms}{1\ s} = \frac{20}{v}\ ms$

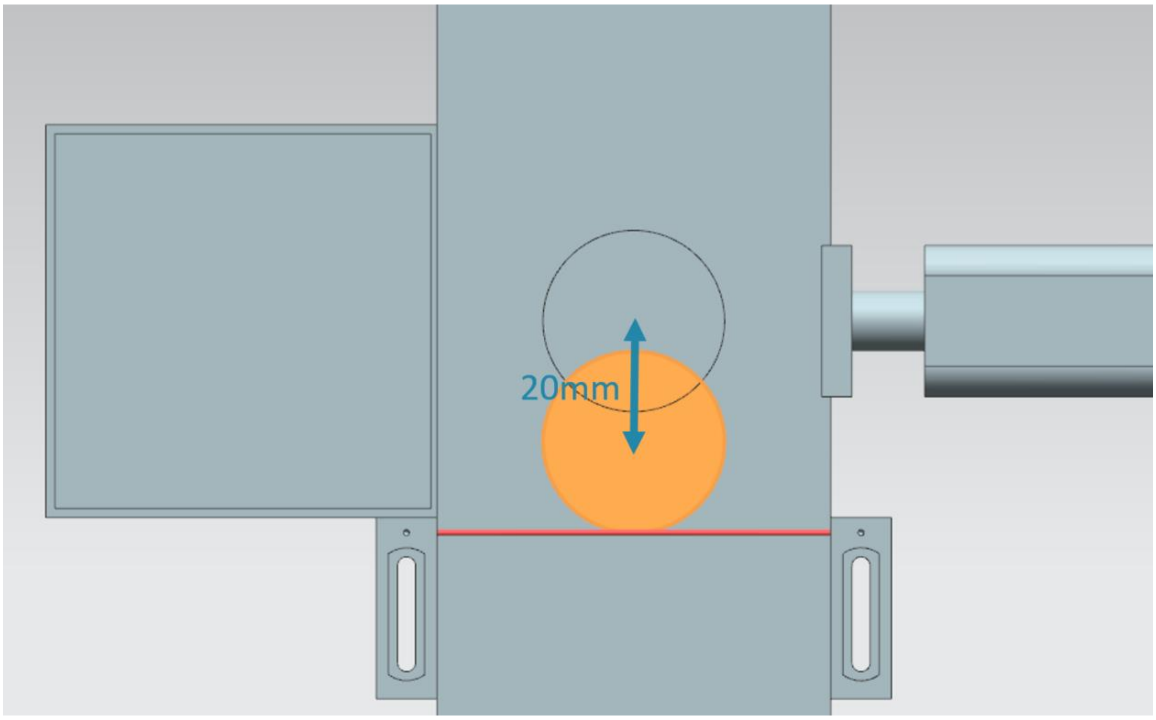


Figure 5: Distance of a cylindrical workpiece from the triggering of the sensor to the ejector arm.

If the user has defined a traverse speed of 0% via the HMI, the calculation formula cannot be applied. In this case, the denominator of the formula would assume the value ZERO, which is not mathematically defined. In this case, a waiting time of 0 ms is specified.

Since the variable speed is provided by the HMI as a percentage value, it must first be converted into the unit m/s. The variable speed is then converted into the unit m/s by the HMI.

This is done in the proposed solution by the following context: $v\left[\frac{m}{s}\right] = \frac{v[\%]}{100\%}\ x v_{max}[\frac{m}{s}]$.

This simple rule of three was implemented in the TIA project using two constants and a temporary tag:

- PERCENT_CONV: Factor defined as constant for the conversion of the percentage velocity value into a floating-point format (= $\frac{1}{100\%}$ = 0.01)
- MOTORSPEED_MAX: The maximum travel speed of the conveyor belts is defined as a constant (= $v_{max}[\frac{m}{s}]$ = 0.15 m/s)

- tempSpeedConvertPercentToM_S: temporary tag for converting the percentage speed value to a value in the unit m/s. The following conversion formula results when looking at the specified rule of three: $tempSpeedConvertPercentToM\_S = Speed\ in\,\% \times$ PERCENT_CONV$x$MOTORSPEED_MAX
  The tag for the expression "Speed in %" is provided in FB "SortingPlantControl" **conveyorLongVarSpeedPercentValue**.

After converting the speed, the waiting time can now be calculated according to the above formula. Two additional constants have been defined as auxiliary parameters in FB "SortingPlantControl".

- POS_LIGHTSENSOR_CYLINDERCENTER: The distance between the light beam of the "Cylinder" light sensor system and the center of the ejector arm (= 20 mm).
- S_TO_MS: Conversion factor s to ms (1 s = 1000 ms)

The result of the calculation is in REAL format. To convert it into a valid time value, however, it must be in DINT format. Therefore, the calculated REAL value is converted into a DINT value using the CONVERT function. In this format, the value can now be assigned to the corresponding input of the ON delay.

A corresponding code chapter from the proposed solution can be found in the Figure 6.

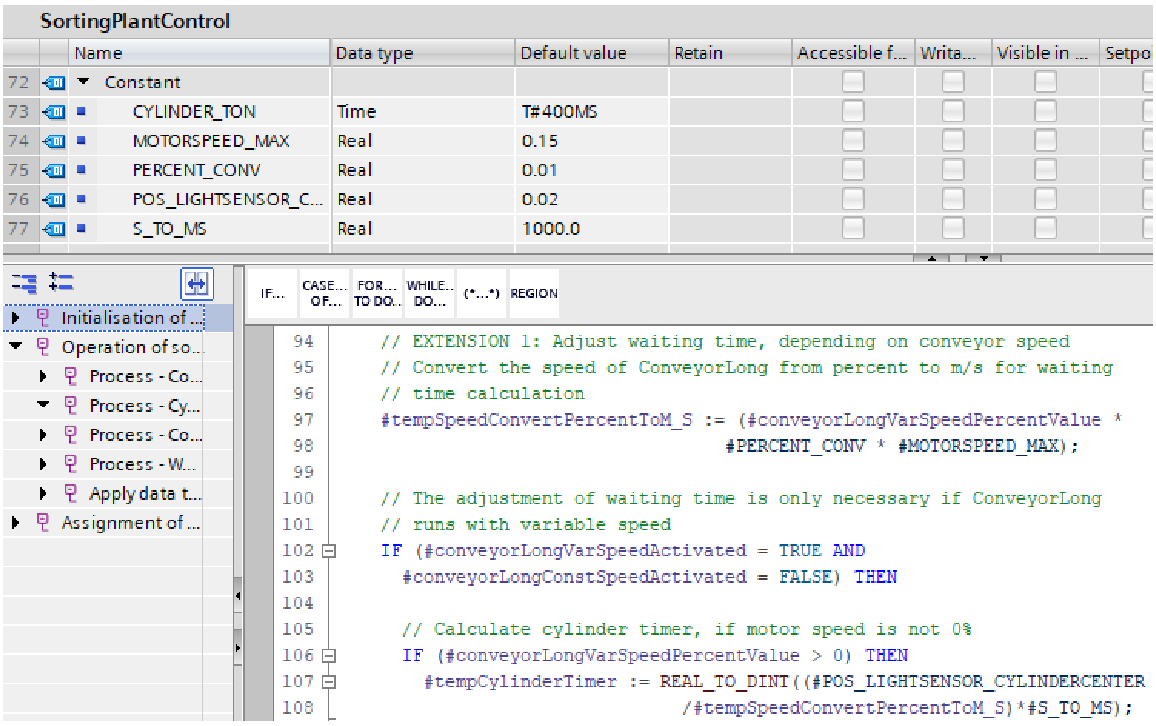


Figure 6: Code excerpt of the changes in FB "SortingPlantControl" for adjusting the waiting time for reliable ejection

When retesting, the conveyor belt should always stop so that the workpiece is at the center of the ejector arm. The sorting process should then be reliably executed, as can be seen from the examples in Figure 7 and Figure 8.

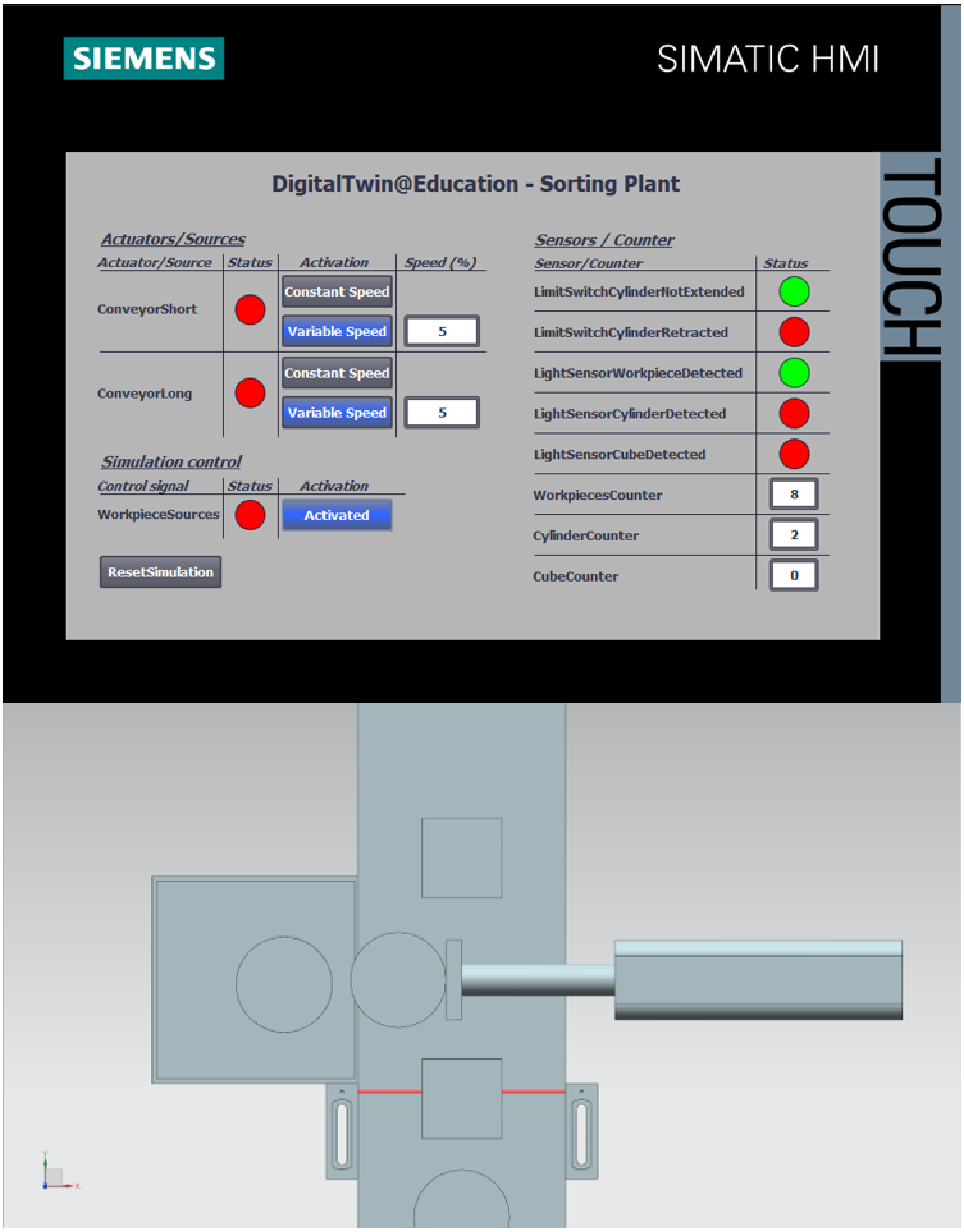


Figure 7: Ejection behavior after adjustment of waiting time for low conveyor belt speeds

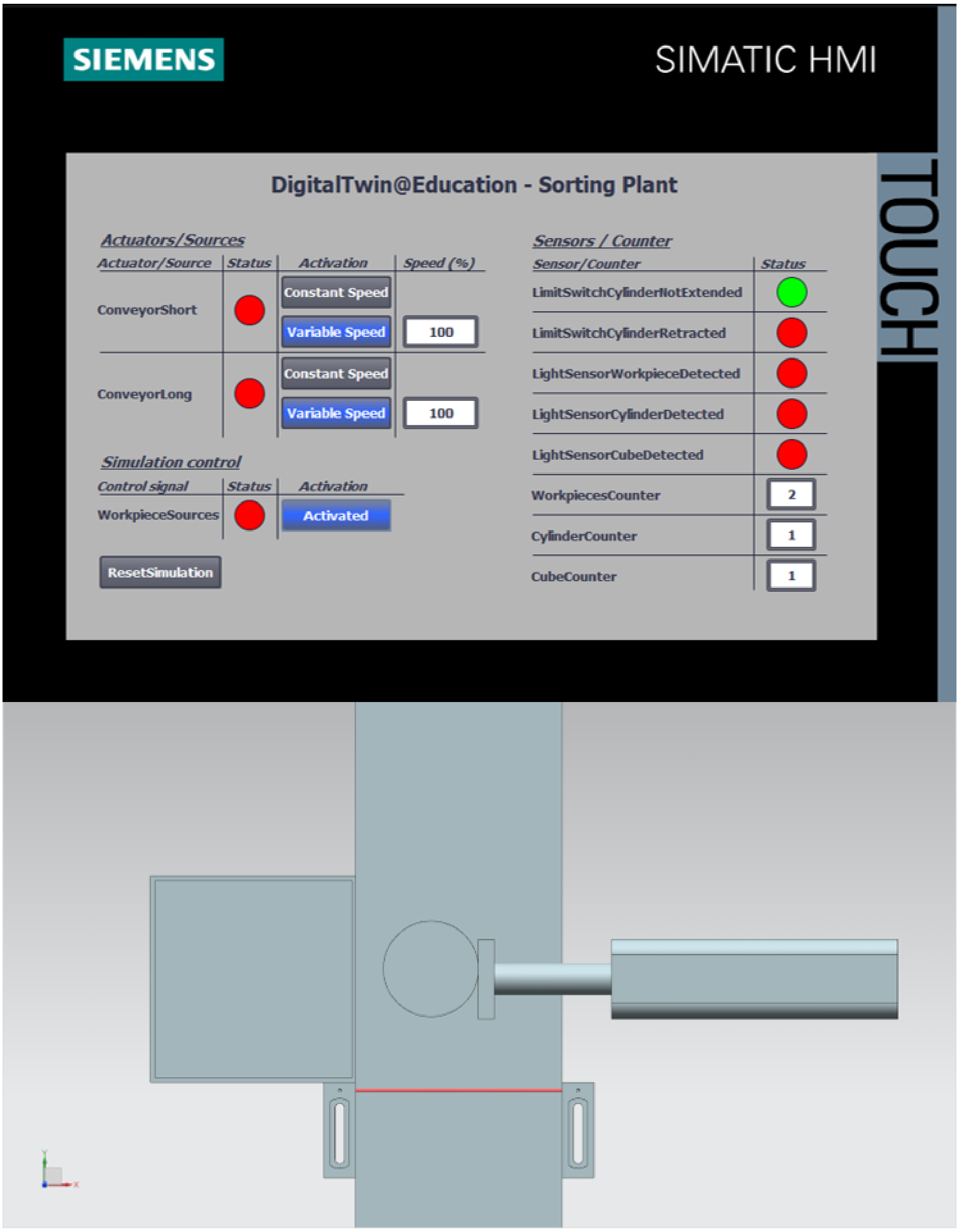


Figure 8: Ejection behavior after adjustment of waiting time for high conveyor belt speeds

However, it can be seen from Figure 8 that the workpiece is not ejected exactly at the center of the ejection cylinder. The reason for this is the friction between the conveyor belt surface and the cylindrical workpiece. Abrupt stopping of the transport surface causes the cylindrical workpiece to slip slightly in the travel direction before it finally comes to a standstill with some offset. However, the automation program has no effect on this behavior. If necessary, the friction parameters can be adjusted in the 3D model within the NX MCD tool.

## 7.2 Ejection while the belt is running

In this case, only changes in FB "SortingPlantControl" are to be made in order to implement ejection with the conveyor belts running, in the same way as described in Chapter 4.3.

Again, the basic procedure for sorting out the cylindrical workpieces remains the same. However, the calculation of the waiting time, as illustrated in Figure 9, must be changed.

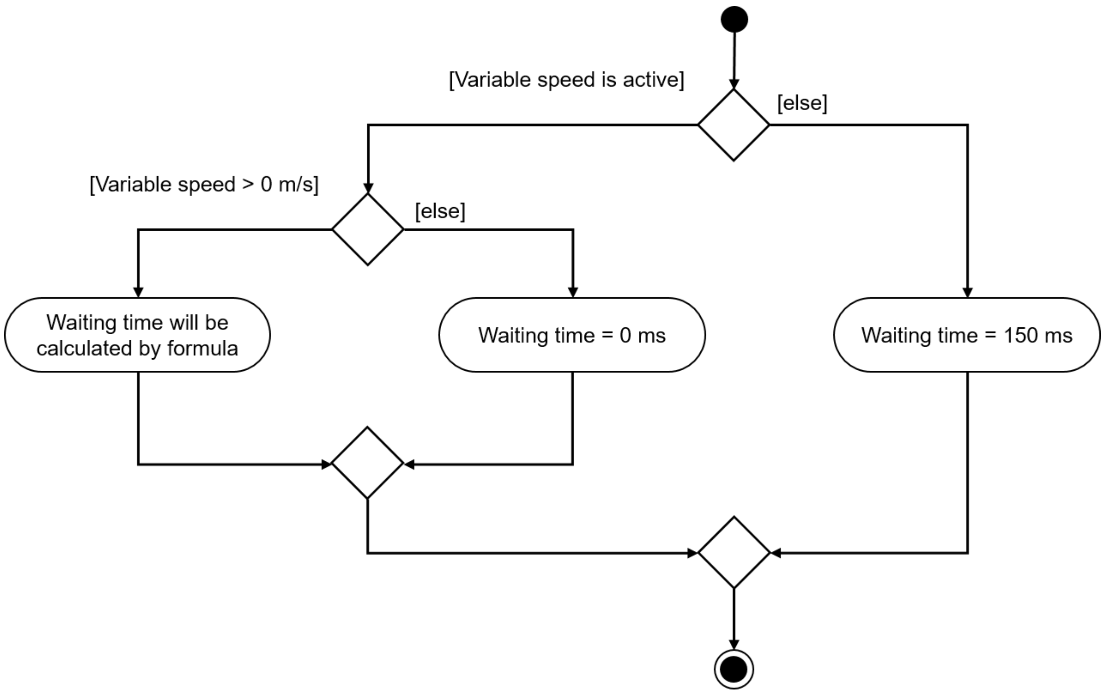


Figure 9: Activity diagram for the adjustment of the waiting time calculation for ejection while the belt is running

The waiting time at variable belt speed is determined by the following assumptions:

- In order to eject the cylindrical body while the belt is running, the workpiece should be transported 7.5 mm = 0.0075 m further to the front edge of the ejection arm after the negative edge of the "Cylinder" light sensor, see Figure 10.
- The currently valid speed **v** in m/s is used as the reference speed.
- This results in the following formula: $t_{Wait} = \frac{0.0075\ m}{v\ [\frac{m}{s}]} x \frac{1000\ ms}{1\ s} = \frac{7.5}{v}\ ms$

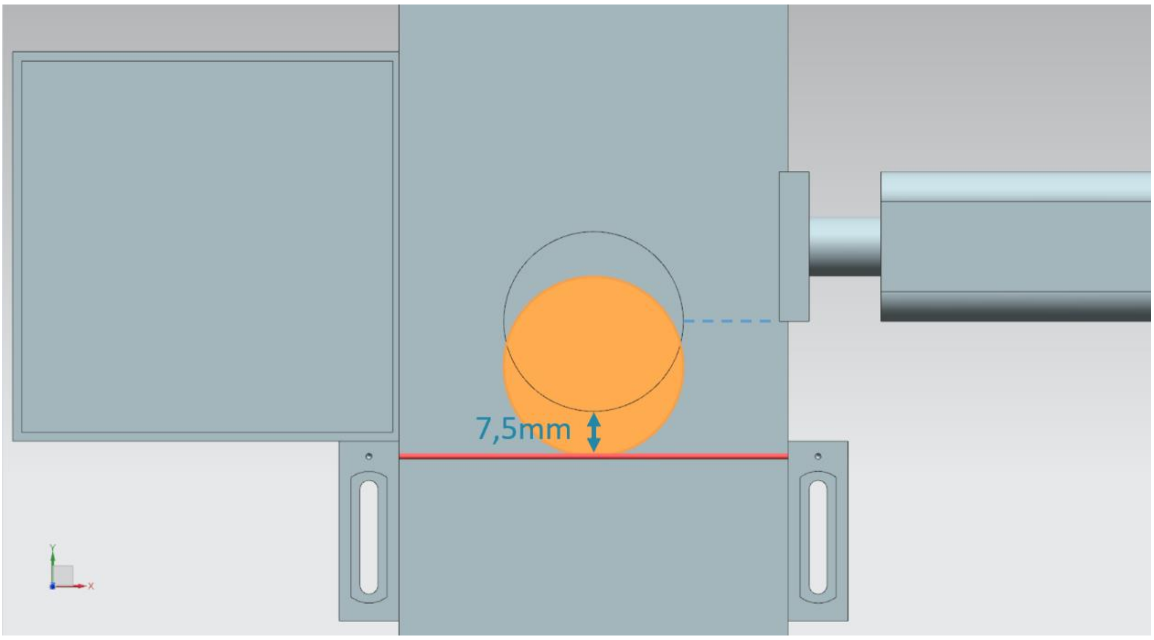


Figure 10: Distance of a cylindrical workpiece from the triggering of the sensor to the edge of the ejector arm

However, if the user has defined a travel speed of 0% via the HMI, a constant waiting time of 0 ms must be specified without calculation in order to prevent a division by ZERO in the formula.

When moving at constant speed, the previously constant waiting time should now be adjusted according to the above formula. Since the constant speed is fixed at 50 mm/s, this results in a new waiting time of

$$t_{Wait} = \frac{0.0075\ m}{0.05\ [\frac{m}{s}]} x \frac{1000\ ms}{1\ s} = \frac{7.5}{50}\ ms = 150\ ms$$

As already mentioned in Chapter 7.1, it is recommended that the percentage value of the variable speed first be converted to the unit m/s. The rule of three described in Chapter 7.1 was used as well with this extension. Again, the two constants and the temporary tag are used:

- PERCENT_CONV: Factor defined as a constant for converting the percentage speed value into a floating-point format (= $\frac{1}{100\%}$ = 0.01)
- MOTORSPEED_MAX: The maximum travel speed of the conveyor belts is defined as a constant (= $v_{max}[\frac{m}{s}]$ = 0.15 m/s)

- tempSpeedConvertPercentToM_S: temporary tag for converting the percentage speed value to a value in the unit m/s. The following conversion formula results when looking at the specified rule of three:

  $tempSpeedConvertPercentToM_S = \ Speed\ in\ \% \times \text{PERCENT\_CONV}x\text{MOTORSPEED\_MAX}$

  The tag for the "Speed in %" expression in the provided FB "SortingPlantControl" is **conveyor LongVarSpeedPercentValue**.

The waiting time can now be calculated with this representation of the specified variable speed. Two further constants have been defined as auxiliary parameters for this purpose:

- POS_LIGHTSENSOR_CYLINDEREDGE: The distance between the light beam of the "Cylinder" light sensor system and the front edge of the ejector arm (= 7.5 mm)
- S_TO_MS: Conversion factor s to ms (1 s = 1000 ms)

This can now be used for the calculation of $t_{Wait}$ using the formula above. However, the result must first be converted from REAL to DINT using the CONVERT function. Only then is the assignment to the corresponding input of the ON delay possible.

Figure 11 shows a code chapter from the proposed solution for adapting the waiting time.

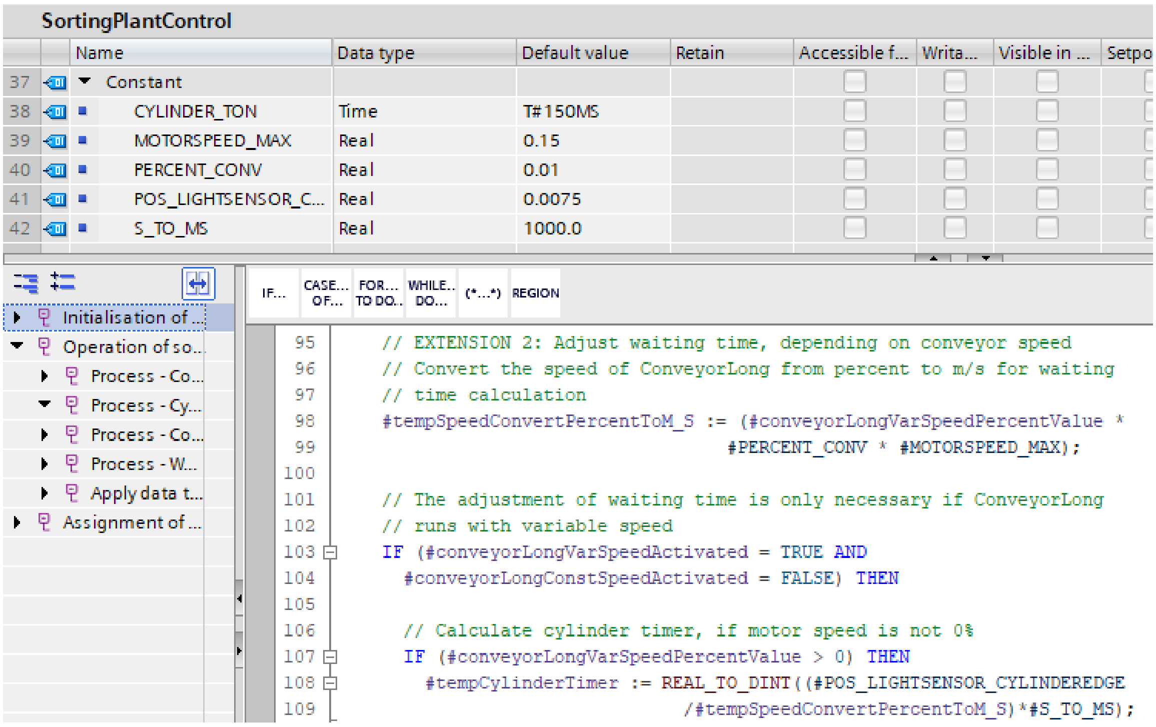


Figure 11: Code excerpt of the changes in FB "SortingPlantControl" for adjusting the waiting time for ejection while the belt is running.

Since the stopping and locking of the conveyor belts for ejection is no longer required for this extension, corresponding locks have been removed from the program and the conveyor belts can be controlled normally. This is shown in the activity diagram in Figure 12.

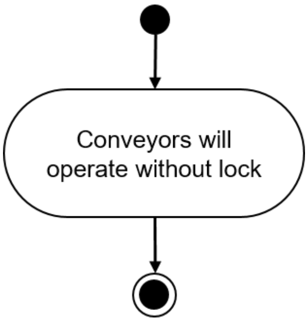


Figure 12: Activity diagram for the conveyor belts within FB "SortingPlantControl" for sorting out while the belt is running

Figure 13 shows the activity diagram for controlling the generation of workpieces after implementation of this extension. It can be recognized that the generation of new workpieces can be activated at any time. Any restrictions from previous modules are no longer needed with this extension.

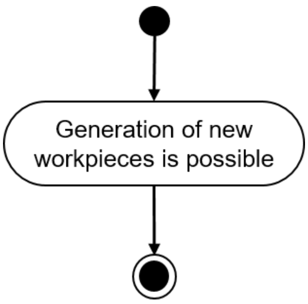


Figure 13: Activity diagram for the generation of workpieces within FB "SortingPlantControl" for sorting while the belt is running

Testing this extension shows that it is now possible to eject and sort the workpieces without stopping the transport surfaces. This is demonstrated, for example, in Figure 14.

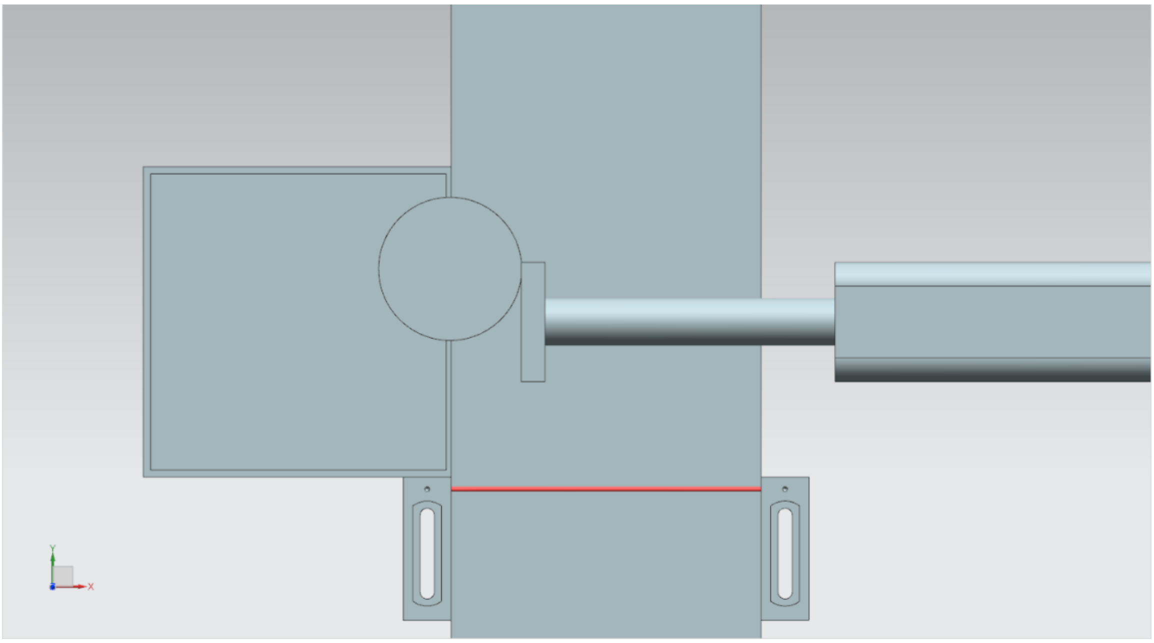

Figure 14: Ejection behavior after adjustment of the waiting time for ejection while the belt is running

Further optimizations and extensions on the part of the automation program are conceivable:

- When generating new workpieces, attention must be paid that no new bodies are produced if the variable speed is less than 5%. Otherwise, the workpieces would stack on top of each other because they are not transported quickly enough from the point of generation. This would make the sorting process considerably more difficult.
- When moving at speeds greater than 50%, it is no longer possible to reliably sort them out while the belt is running. Here, it is conceivable to slow down or limit the speed of the transport surface by the automation program during the ejection process to guarantee correct ejection. This involves further changes. For example, the user should be informed via HMI that the speed must be reduced in order to maintain the functionality of the ejection process. Input of a new variable speed should be blocked in this case.

Basically, this module has shown that optimizations and extensions of the automation program can be verified effectively and quickly with the help of a digital twin.

Pure automation activities of this workshop series have been worked through with this module. In the next module, you will perform the first steps with the CAD tools of NX MCD and create the static model of this training series completely independently.

# 8 Checklist – step-by-step instructions

The following checklist helps students/trainees to independently check whether all steps of the step-by-step instructions have been carefully completed and enables them to successfully complete the module on their own.

| No. | Description | Checked |
|---|---|---|
| 1 | For extension 1, a copy of the TIA project from Module 2 was successfully created and renamed. | |
| 2 | Errors in the automation program were eliminated based on the descriptions in Chapter 4.1, Chapter 4.2 and Chapter 7.1. | |
| 3 | The implemented changes were tested. | |
| 4 | For extension 2, a copy of the TIA project from Module 2 was successfully created and sensibly renamed. | |
| 5 | You optimized your own automation program based on the descriptions in Chapter 4.3 and Chapter 7.2. | |
| 6 | The implemented optimization was tested. | |

Table 1: Checklist of "Extensions and optimizations of an automation program for a 3D model".

# 9 Additional information

You can find additional information as an orientation aid for initial and advanced training, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software/firmware, at the following link:

**Preview "Additional information" – In preparation**

Here are some interesting links:

[1] support.industry.siemens.com/cs/document/90885040/programming-guideline-for-s7-1200-s7-1500?dti=0&lc=en-US

[2] support.industry.siemens.com/cs/document/109756737/guide-to-standardization?dti=0&lc=en-US

[3] omg.org/spec/UML/2.5.1/PDF

[4] geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/

[5] geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/

## Further Information

Siemens Automation Cooperates with Education
**siemens.com/sce**

SCE Learn-/Training Documents
**siemens.com/sce/documents**

SCE Trainer Packages
**siemens.com/sce/tp**

SCE Contact Partners
**siemens.com/sce/contact**

Digital Enterprise
**siemens.com/digital-enterprise**

Totally Integrated Automation (TIA)
**siemens.com/tia**

TIA Portal
**siemens.com/tia-portal**

TIA Selection Tool
**siemens.com/tia/tia-selection-tool**

SIMATIC Controller
**siemens.com/controller**

SIMATIC Technical Documentation
**siemens.com/simatic-docu**

Industry Online Support
**support.industry.siemens.com**

Product catalogue and online ordering system Industry Mall
**mall.industry.siemens.com**

Siemens
Digital Industries, FA
P.O. Box 4848
90026 Nuremberg
Germany



**siemens.com/sce**