Saimaa University of Applied Sciences
Faculty of Technology
Mechanical Engineering and Production Technology, Lappeenranta

Elena Taborova

Animation and simulation applications for education

Thesis 2015

Abstract

Elena Taborova
Animation and simulation applications for education, 70 pages, 23 appendices
Saimaa University of Applied Sciences
Bachelor of Engineering, Lappeenranta
Mechanical Engineering and Production Technology
Thesis 2015
Instructors: Lecturers Timo Eloranta and Simo Sinkko, Saimaa University of Applied Sciences.

The main aim of this thesis was to investigate the backgrounds of simulation, animation and modelling. The 3D model of the Roller Conveyors system was created as a case of study in the laboratory at Saimaa University of Applied Sciences. It was designed by using Festo CIROS Studio software. The logic control of the system was programmed in Ladder Diagram language with SIMATIC Manager STEP7, and integrated with virtual Programmable Logic Controller.

3D simulation technology has been adopted successfully in production industry. It has a lot of benefits such as: visualising the on-going process for better understanding, ability to control the system, analysing the results and some other opportunities, which will be described in this thesis. Two programs, based on modelling and simulations, will be compared and then qualified. In conclusion, ideas of improvements of the Roller Conveyor system and possible further applications will be considered.

The evaluation of the results indicates a possibility to continue studying the model by using other programs. It will help to explore the system from other points of view.

Keywords:

- 3D-modelling
- Simulation technology
- Animation
- Programmable Logic Controller
- CIROS Studio
- SIMATIC Manager STEP7
- Programming languages
- Power Hydraulics

Table of contents

Abbreviations

3D – three dimensional

CAD – Computer Aided Design

CCW – Contra Clock Wise

CNC – Computer Numerical Control

CPU – Central Processing Unit

EEPROM – Electrically Erasable Programmable Read Only Memory

EPROM – Erasable Programmable Read Only Memory

FBD – Functional Block Diagram

FMS – Flexible Manufacturing System

IL – Instruction List

I/O – Input and Output

IRL – Industrial Robot Language

LAD or LD – Ladder Diagram

LED – Light-Emitting Diode

NC – Numerical Control

OB – Organazing Block

OS – Operating System

PLC – Programmable Logic Controller

RAM – Random Access Memory

ROM – Read-Only Memory

ST – Structured text

STEP – STandard for the Exchange of Product model data

STL – Statement list

STL – STereoLithography

1 Theory

1.1 Introduction into the modelling and simulation technology

This thesis starts with a research of modelling and simulation processes. The definition of the term Simulation can be explained differently.

Simulation is one of the most powerful analysis tools. It is responsible for the design and operation of the complex processes and systems. (*Introduction to simulation*, Robert E. Shannon, 1992)

The main aim of modelling and simulation processes is to study limitations of the system before testing it in a real model. There is a probability that designers have limited knowledge about the actual system, before implementing it in reality. Experiments with a system allow to study all aspects of the concept, and to explore more facts about the simulated object. Also, visualising the model is an easy way to understand the on-going process in the system for the human being.

The various industries have got wide spread of simulation technology, such as manufacturing industry, computer science, military use, bioscience etc. The main reason of easy adaptation is the close connection between reality and visualisation in 3-dimension universe. It is always a high risk to use a system in reality without testing it in advance. So, the safe choice to avoid risks and faults is to experiment this system by using simulators. The simulation can be run at the same time with implementation in the real time, as well as beforehand. Some common risks, influencing on the production, are time, safety and price. Modelling and simulation technology are designed with goals to reduce cost, to avoid risks and increase rate of success.

As we say, the medal has two sides: besides advantages there exist disadvantages. Of course, simulation has limits, because various applications require suitable interfaces, which might be reasons of complicated data transforming and protocol exchanging. Some examples of advantages and disadvantages are shown in Table 1.

Table 1. Advantages and Disadvantages of simulation technology

	Advantages	Disadvantages
Automation technology	Capability for troubleshooting	No standardized technology
	Ability to get a result of processes which cannot be measured experimentally with current level of technologies	The software itself might be expensive
Operation process	Unnecessary full input data for implementing the model	Many errors while simulation
	Investigation of “What-if” situations	Necessary well-based theories for programming and simulating the system
	Possibility to implement any idea	It is not always accurate simulation of the real-world ideas
General view on technology	Relative cost comparing to testing the system in reality	Staff needs to be trained how to use the software, which also adds expenses
	Modern approach	Unbelief that the system will perform in reality

Typical simulation process

Figure 1.1 shows the steps, required for the simulation process: starting from the problem definition and until the declaration of results.

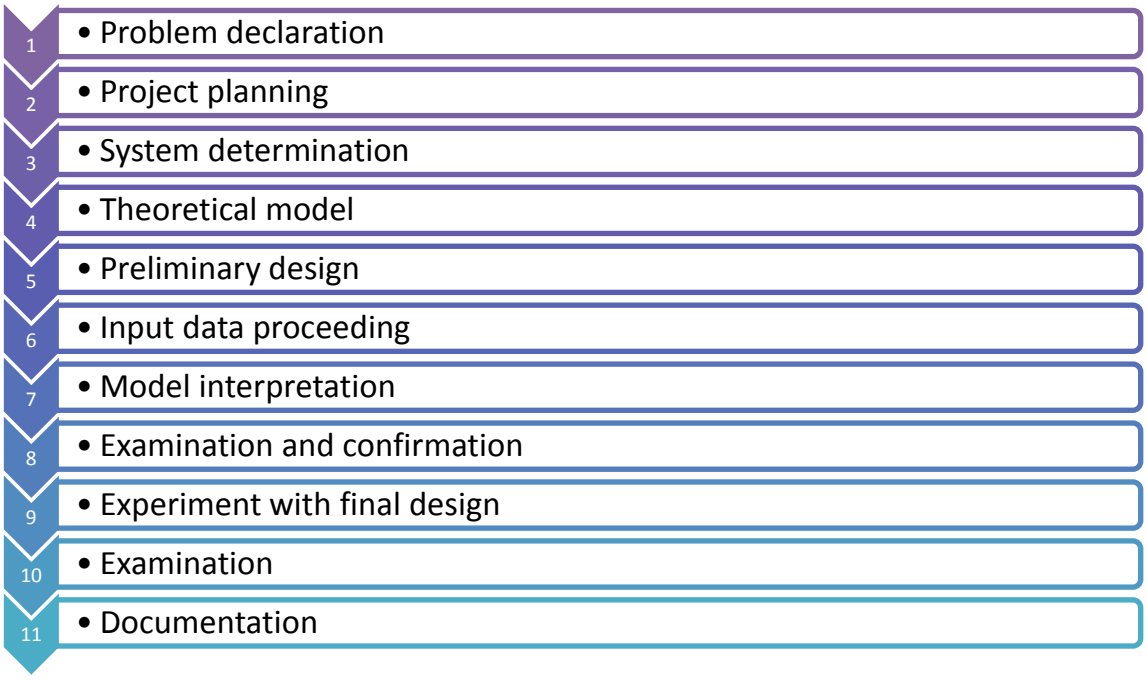


Figure 1.1 Typical simulation process

Online monitoring

An on-line monitoring system is a process with the main functions such as dynamic collecting, interpreting and responding on information related to performances happening in application. Monitoring gathers information of executing computational processes and then classifies it by functionality. It gives users a possibility to choose the information what has to be monitored.

Online monitoring can add proper techniques to increase application reliability. Reliability includes fault tolerance and safety. It helps to detect runtime errors. Also, monitoring is used to extract the data from the system for analysing it on evaluation of performance later on.

Flexible Manufacturing System

Flexible Manufacturing System (FMS) is a system with an amount of flexibility, which allows the system to react on changes. A FMS is a group of numerically-controlled machine tools, interconnected by a central control system. A group of processing stations are attached to each other by means of an automated material handling and storage system. It is called flexible because the system is capable to changes in the product being manufactured. It gives an advantage in quickly changing manufacturing environment with less down time and greater efficiency. The most common applications of FMS are CNC machine tools.

The main advantages of using FMS:

- Higher quality of products
- Higher production rates
- Reducing manufacturing cost
- Greater machine efficiency and reliability
- Saving time for producing
- The production of certain products can be mixed
- Time for changeover is decreased
- The various products can be made at the same time on the system
- A variety of different types of parts can be processed under NC program control at the various workstations

1.2 Components of the animation and simulation system

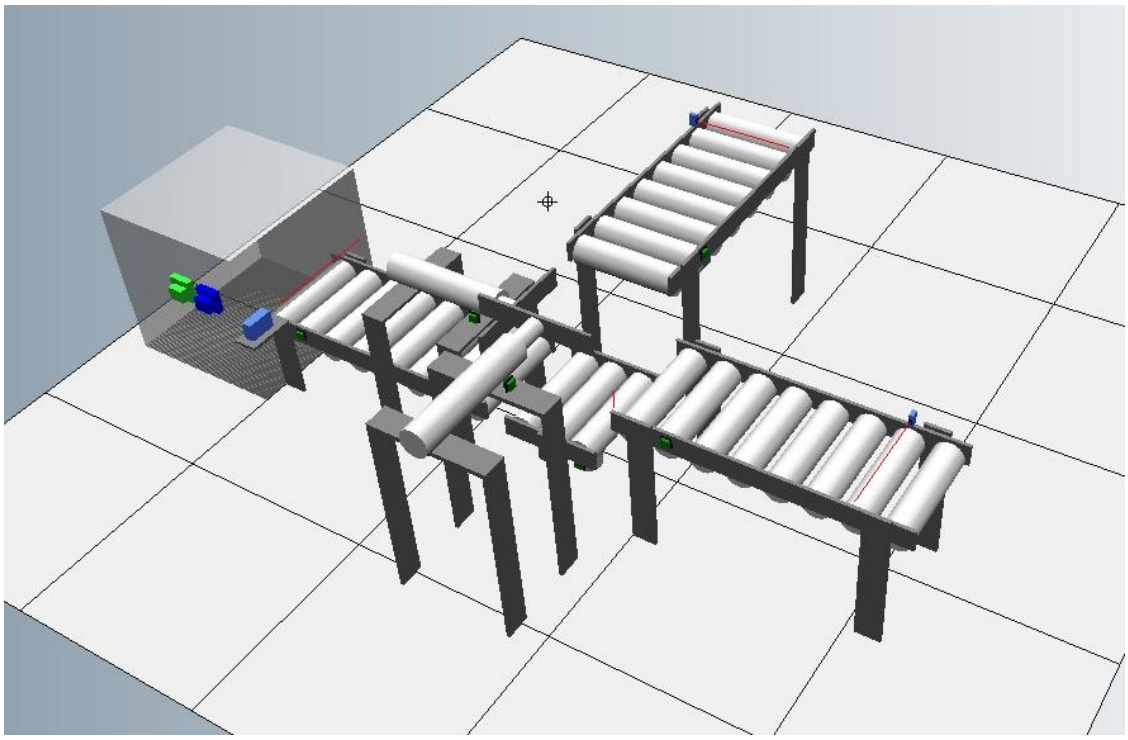


Figure 1.2 Roller Conveyor system

The whole representation of the Roller Conveyor system is shown in Figure 1.2. The constituents and their descriptions of the Roller Conveyor system:

- Replicator Socket, established at the start of Conveyor 1, produces Workpieces
- Two types of Workpiece: blue and green colour templates
- Buttons send information to the Replicator about colour of the Workpiece, which has to be produced
- Colour Detecting Sensor, which detects the colour of the Workpiece, and determines the subsequent path for it. Also this sensor switches Conveyor 1 on
- Three Roller Conveyor Belts, established on different heights
- Lifting Platform, which will supply Workpiece to different conveyors, according to the special features. As well, this platform has to rotate on 90 degrees CCW
- Two Pushing Cylinders which will shift the Workpiece to the certain conveyors

- Reflexing Light Barrier, established on the Lifting Platform, detects the availability of the Workpiece at the end of Conveyor 1
- Two Reflexing Light Barriers, established at the ends of Conveyor 2 and Conveyor 3, detect the Workpieces at the end of conveyors. Send an information about resetting to the initial positions of the Lifting Platform and Pushing Cylinders
- Two invisible Trashcan mechanisms, established at the end of Conveyor 2 and Conveyor 3, remove the Workpieces from the ends of conveyors
- LEDs show which component of the system is working at the moment

A more detailed description of the parts included in the system will be discussed in Chapters 1.3 and 1.5.

1.3 Description of the practical components of Power Hydraulics

Hydraulics is widely used in systems, where a high force and accuracy are required. The reason of using components of Power Hydraulics is their features, which will be described in this chapter.

1.3.1 Typical components of the Power Hydraulic system

- Power supply
    - Hydraulic pump
    - Filters
    - Heaters
    - Coolers
- Hydraulic fluid
- Directional control valve
- Pressure valve
- Flow control valve
- Non-return valves
- Cylinders
- Hydraulic motor

Hydraulic motors and cylinders play a role as actuators. Motors convert the hydraulic energy into mechanical, and generate rotary movements, and cylinders generate translation movements. Flow control valves are used to reduce the speed of the cylinder or the speed of motor.

Some accessories of a Power Hydraulic system:

- Pipes
- Flexible hoses
- Screw fittings
- Pressure and flow gauges
- Quick-release couplings

These accessories are mostly used for transporting hydraulic energy, connecting and clamping components and executing the checking functions.

1.3.2 Advantages and disadvantages of using Power Hydraulics

Table 2. Advantages and Disadvantages of using Power Hydraulics

Advantages	Disadvantages
Capability to transmit large forces by using small components	Sensitive technology to environment (temperature changes, dirt etc.)
Easy start-up under heavy loads	Oil usage causes environment pollution
Large forces can be generated	Efficiency factor is low
Movements' independency of load, because of usage of flow-control valves	Limited energy storage
Relatively good controlling and manip-ulating system	
Accurate control	
Heat dispersion	
High efficiency of energy transmission	
Good speed controlling	
High stability of the system, since oil is almost incompressible	

1.3.3 Typical applications for using Power Hydraulics

Power Hydraulics is widely used in Machinery, Automotive and Heavy Indus-tries.

- Excavators
- Lifting platforms
- Conveying systems
- Production and assembly machines
- Aircrafts
- Marine applications

- Hot and cold rolling mill
- Coke oven pants

Other technologies besides hydraulics which can be used as a control technology for generating forces, movements and signals are:

- Pneumatics
- Mechanics
- Electricity

The main purpose of Hydraulics is a generation of forces and motions by using hydraulic fluids. Any liquid can be used for transferring pressure energy. However, there are characteristics which have to be taken into account before choosing the correct liquid in the certain system. Mostly hydraulic oils are used, because of their special features. The most considerable properties of hydraulic fluid are viscosity and viscosity index. Viscosity defines the ability of the liquid been poured, i.e. resistance to flow. Viscosity index is a value showing the relations between viscosity and temperature characteristics.

Tasks which have to be performed by hydraulic fluids:

- Lubrication of the moving parts
- Corrosion and wearing protection
- Signal transmission
- Cushioning of vibrations generated by pressure jerks

Components for handling the hydraulic fluid:

- Hydraulic fluid
- Filter
- Cooler
- Heater
- Thermometer
- Pressure gauge
- Reservoir

- Filling level indicator

1.3.4 Types of representation

- Hydraulic circuit diagram – description of the functional structure of the system
- Function diagram – graphical representation of changes in status of components in the system
- Function chart – flow chart with divided control sequence into steps
- Schematic drawing of installation – illustration of the position of the certain cylinder in the system
- Displacement-step diagram
- Displacement-time diagram

1.3.5 Hydraulic Cylinders

The main aim of the hydraulic cylinder is to convert the hydraulic energy into mechanical. It generates the linear motion.

Types of Cylinders

1) Single-acting cylinder

The cylinder has only one port where the fluid pressure can be applied. Consequently, the cylinder can perform work only in one direction. The return motion is provided by external force or by a return spring. The common application of using single-acting cylinder is a lifting platform. Figure 1.3 shows the symbol of single-acting cylinder. Figure 1.4 demonstrates the constituents of single-acting cylinder.

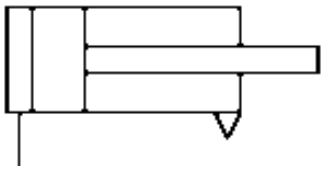


Figure 1.3 Symbolic representation of Single acting cylinder (1)

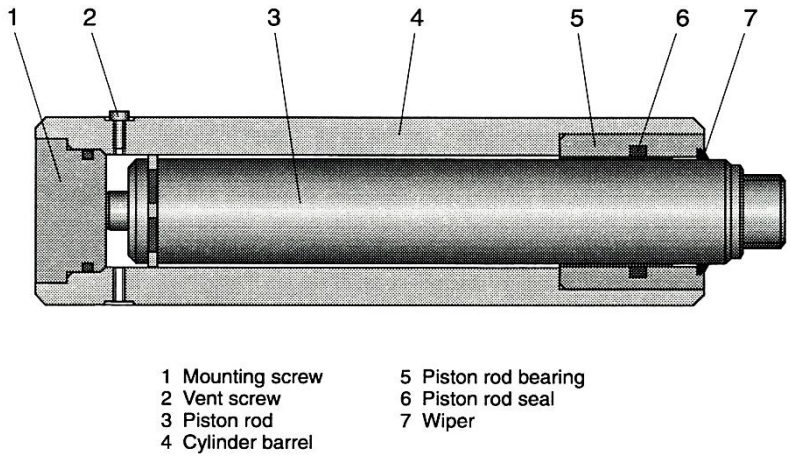


Figure 1.4 Cross-section view of single-acting cylinder (2)

2) Double-acting cylinder

This type of cylinder has two ports for applying fluid pressure. Therefore, it is possible to perform the drive movements in two directions. Figure 1.5 shows the symbol of double-acting cylinder. Figure 1.6 demonstrates the constituents of double-acting cylinder.

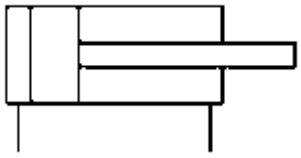


Figure 1.5 Symbolic representation of Double acting cylinder (3)

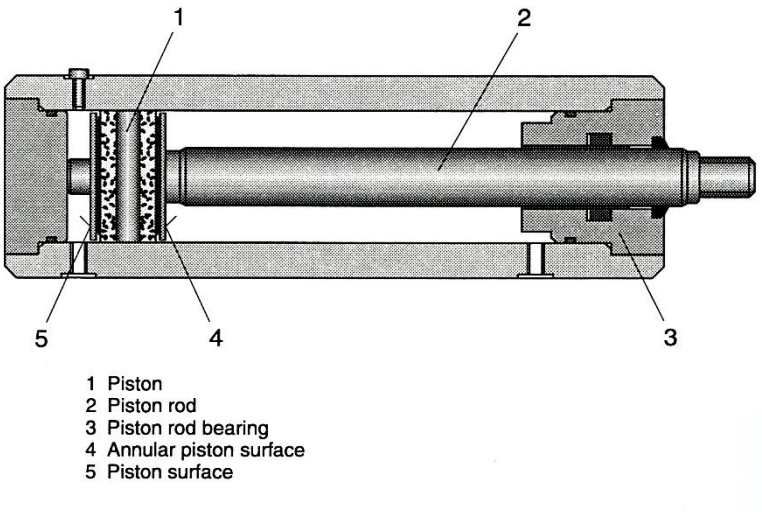


Figure 1.6 Cross-section view of Double acting cylinder (4)

The drive section of a hydraulic system is responsible for the working movements of machine or manufacturing system. Implementation of movements and reproduction of forces are made by contained energy in the hydraulic fluid. This is reached by using components, such as: cylinders and motors. The hydraulic cylinders generate movements for excavator, see Figure 1.7.

Figure 1.7 Application of using the hydraulic cylinders (5)

The main aim of this thesis is to create a 3D-model of the system, which may have hydraulic components. In future, the features of these components can be studied and applied in practice. The hydraulic cylinders can be used as pushing and lifting mechanisms in the Roller Conveyor system. According to advantages of using the hydraulic cylinders, the applying them in the system will be useful and properly.

1.3.6 Conveyor system

A conveyor is responsible for the transportation of objects on the assembly line. The most common conveyor form is the roller conveyor. It consists of the certain quantity of rollers, established perpendicularly to the direction of the path. The rollers are fixed by a frame. The workpiece will be moved as soon as rollers start rotating.

Roller Conveyor System includes three similar conveyor mechanisms. The first conveyor is connected to the replicator and the main aim of that is to supply the Workpiece to the Lifting Platform (see the Figure 1.8). The second conveyor is established along the same axis as the first one, but the height of Conveyor 2 is bigger on 150 mm than the height of Conveyor 1 (see the Figure 1.9). The third conveyor is established perpendicularly to the axis of Conveyor 1 and Conveyor 2, as well, it is higher than Conveyor 2 on 150 mm (see the Figure 1.10). Each Roller Conveyor consists of 8 rollers and miscellaneous mechanism which is called *Conveyor surface*. According to the properties of that mechanism, the Workpiece will be moving along the highest points of the rollers.

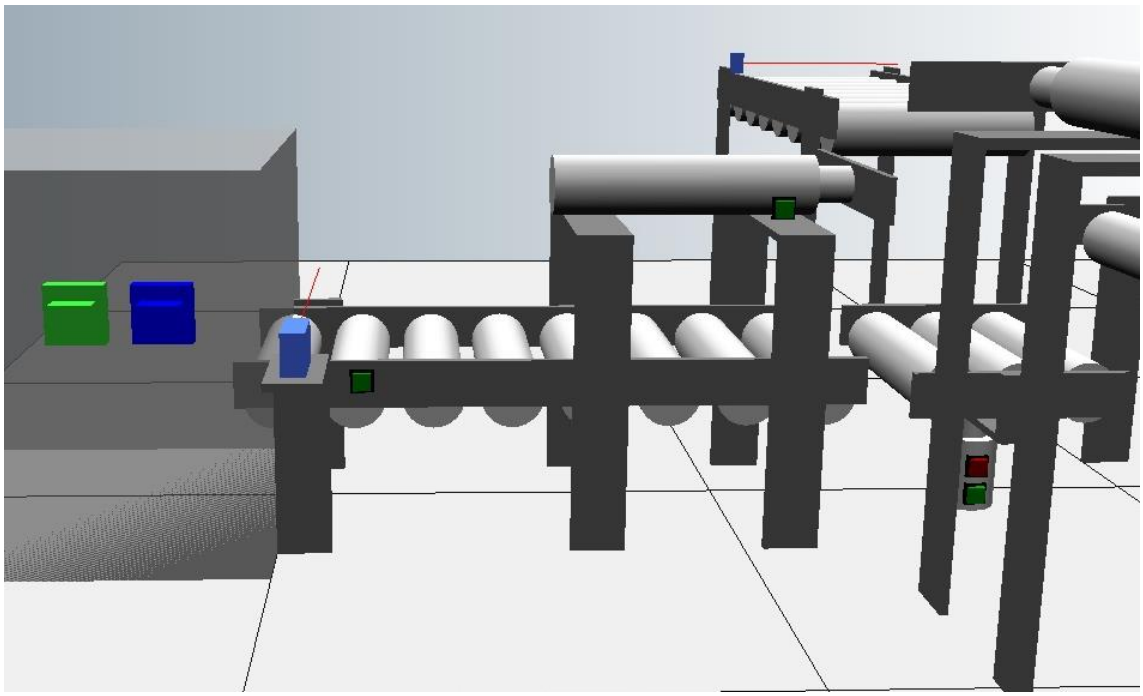


Figure 1.8 Conveyor 1

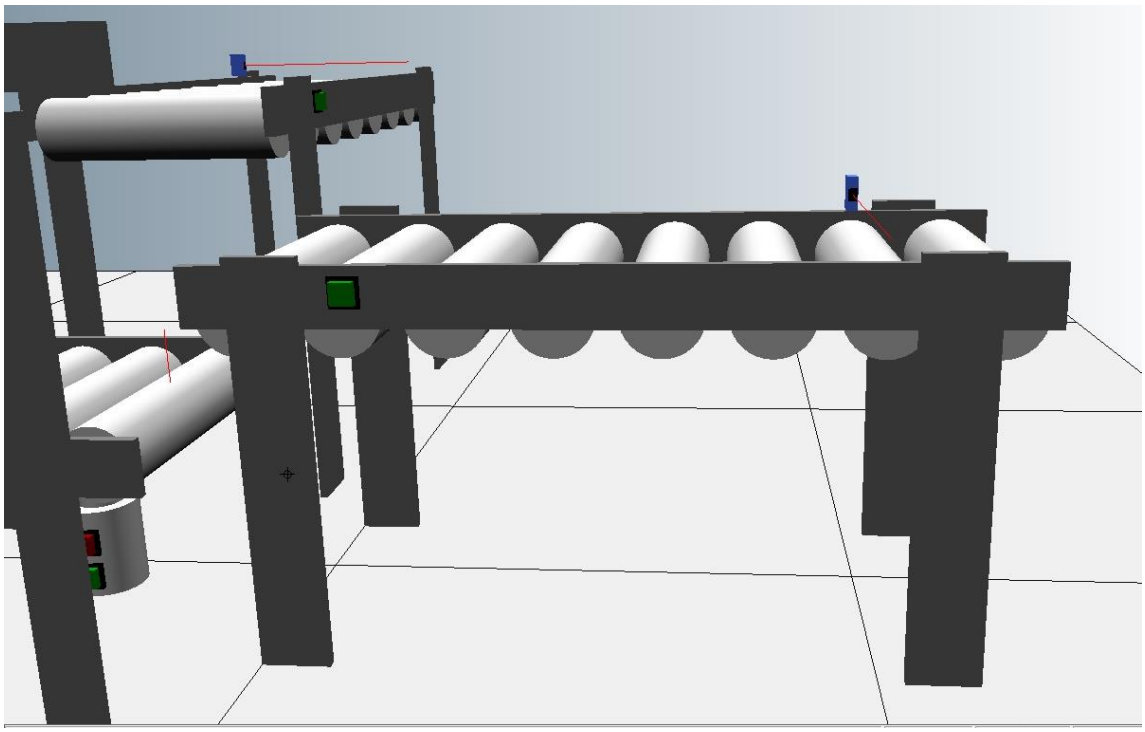


Figure 1.9 Conveyor 2

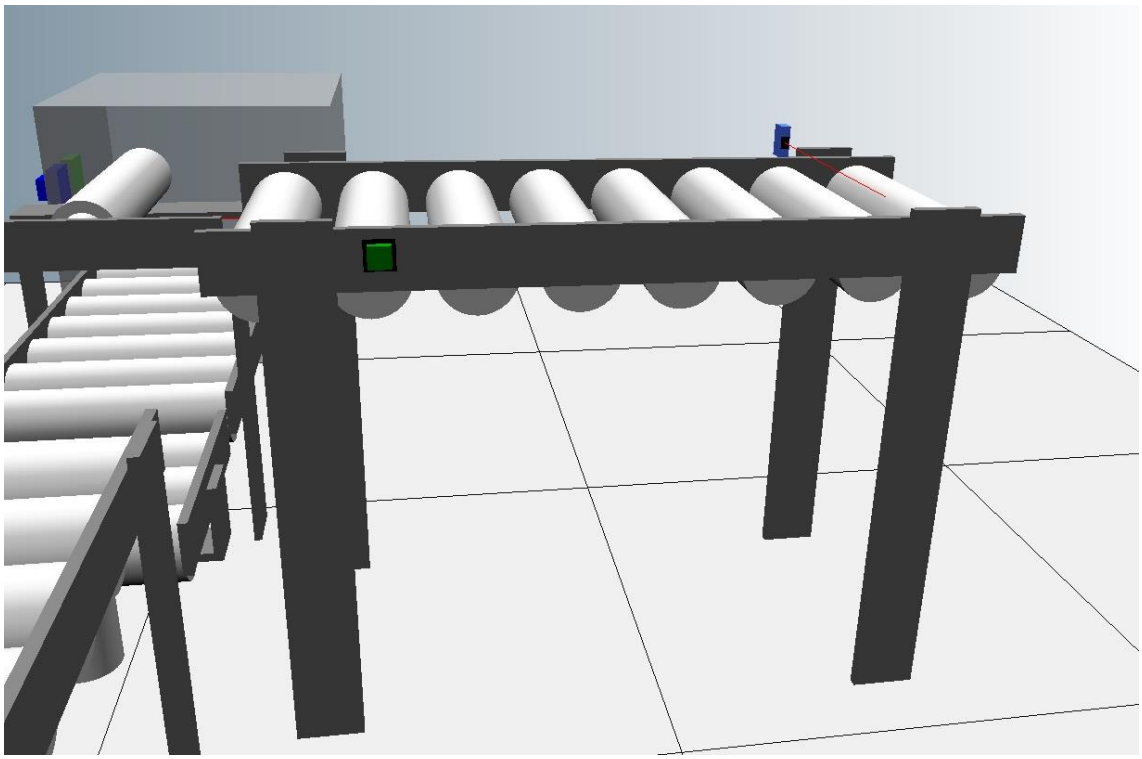


Figure 1.10 Conveyor 3

1.3.7 Lifting Platform

The main idea of the Lifting Platform is supplying an object to other conveyors while on assembly line. It was modelled by using the *Cylinder for translation* and *Cylinder for rotation* mechanisms and individual parts from the roller conveyor. Figure 1.11 shows the appearance of Lifting Platform.

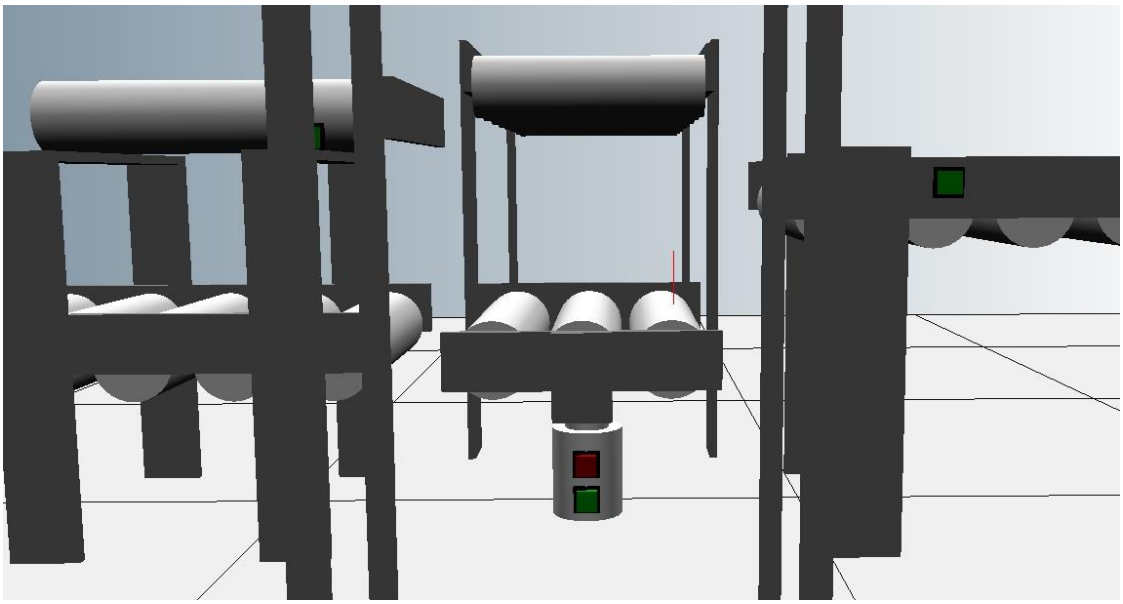


Figure 1.11 Lifting Platform

1.3.8 Pushing cylinders

Since the roller conveyor of the Lifting Platform is a driverless, we need to place an extension cylinder, which will shift the package from the Lifting Platform to the certain conveyor. Pushing Cylinder has a mechanism which is called *Cylinder for translation*. Two Pushing Cylinders are required, because there are 2 conveyors where the Workpiece can be moved. The first Pushing Cylinder is established in front of Conveyor 2 and the second Pushing Cylinder is in front of Conveyor 3. Figure 1.12 and Figure 1.13 represent Pushing Cylinders in the system.

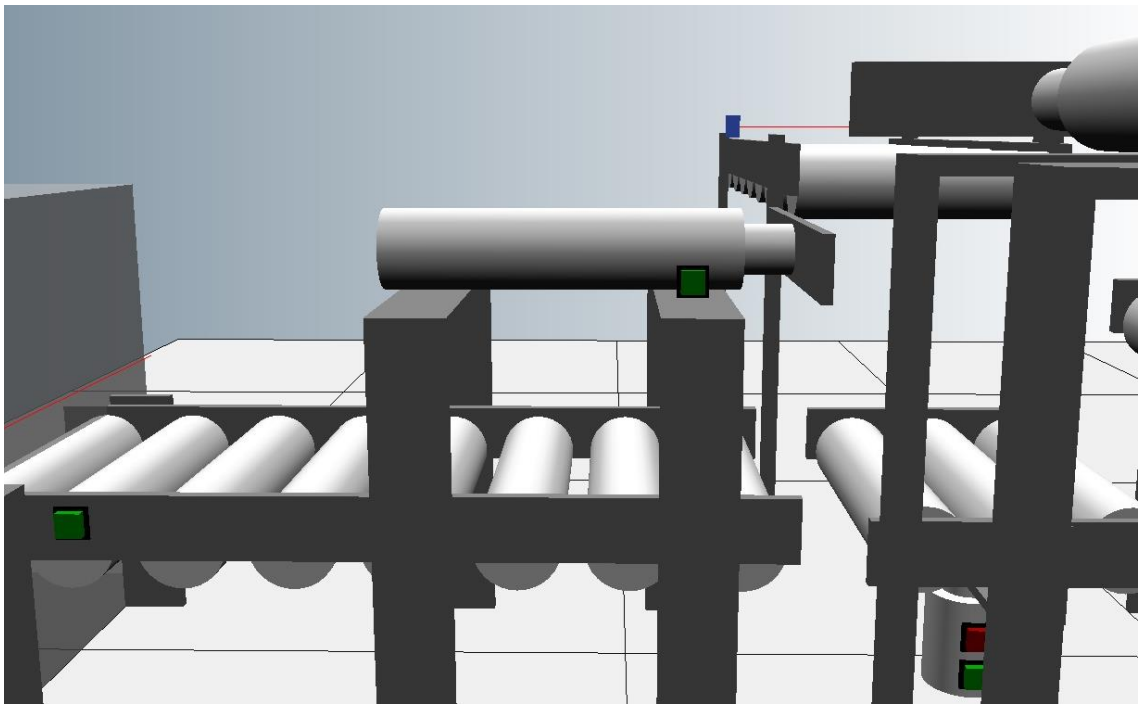


Figure 1.12 Pushing Cylinder 1

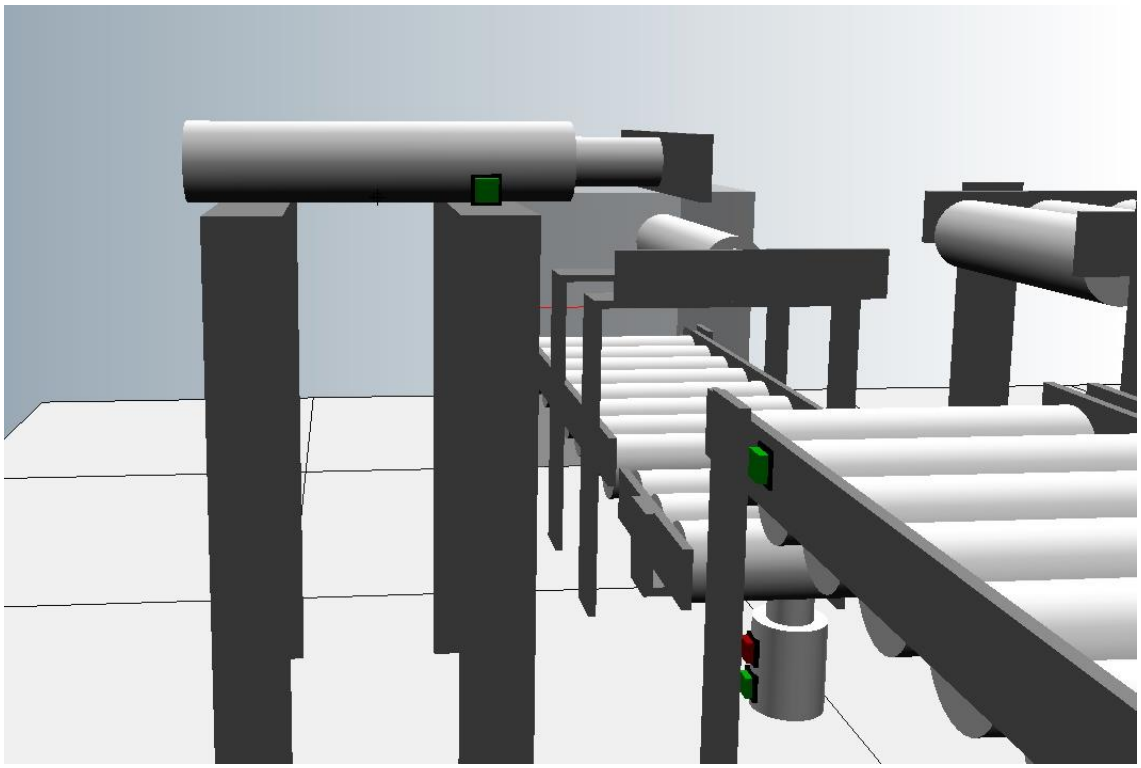


Figure 1.13 Pushing Cylinder 2

1.3.9 Replicator

Replicator Socket is a mechanism which creates new objects according to templates' properties. An object will be created by replicator, whenever the input corresponds to its template. The Workpiece will be posed at the gripper point of the Replicator. Also, there is a possibility to randomize the selection of the templates by establishing the special value "randomize_input" for the property of the number of inputs which is used to create a random object. The value of the property "randomize_range" represents the range of templates, considered for randomized object creation (i.e. 0, 2 means random creation from template0 to template2). Replicator is a pure mechanism without any visualization, so to avoid appearing items out of nothing, thus it is good to design the simple socket for Replicator and assign some material for that.

The Replicator is established at the start of Conveyor 1 in the Roller Conveyor system. The shape of the Replicator is a cube, and the colour assigned for this device is "Smoky glass". Replicator is shown in Figure 1.14.

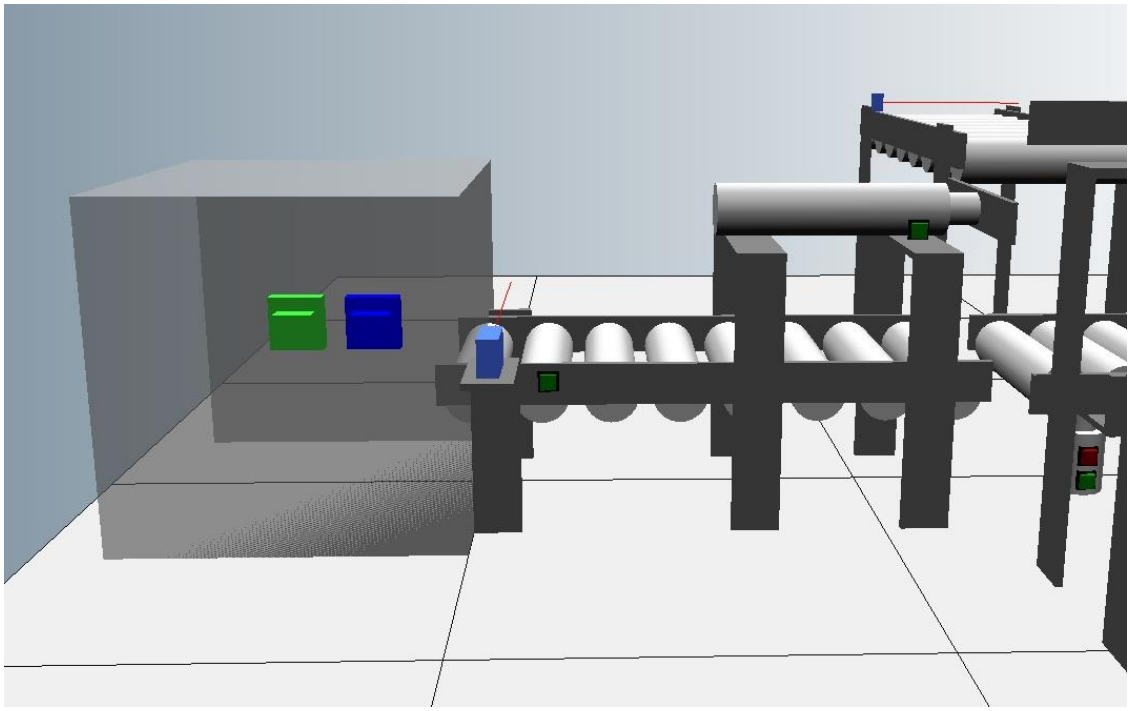


Figure 1.14 Replicator

1.3.10 Two templates of the Workpiece

For the purpose of handling different pallets at the same time, they were distinguished from each other by colour in order to mark which path they should be routed. The blue colour workpiece will be directed to Conveyor 2 (see the Figure 1.15). The green colour workpiece will be transferred to Conveyor 3 (see the Figure 1.16).

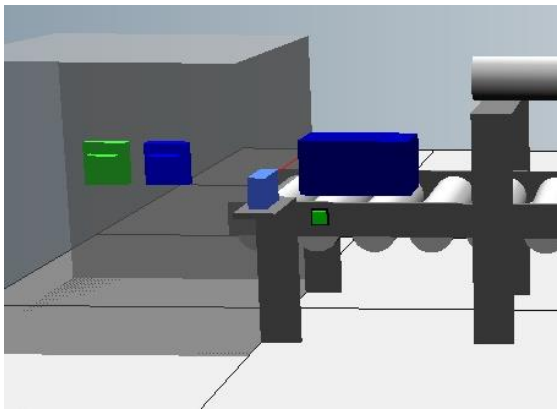


Figure 1.15 Blue Workpiece

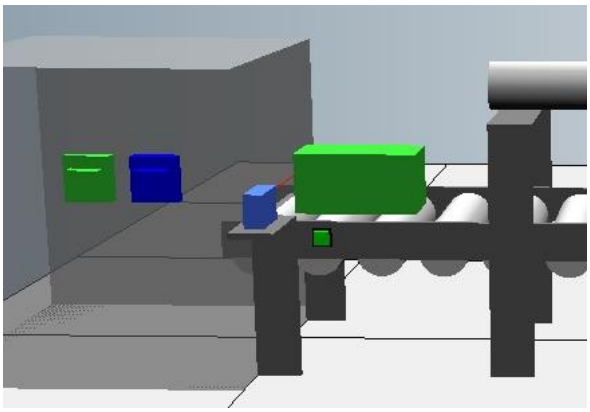


Figure 1.16 Green Workpiece

1.3.11 Trashcan

A property of the Trashcan is removing an object at the runtime. Also, this mechanism can be used as a counterpart of the Replicator. The object will be removed at the place of the gripper point of the Trashcan and when the input of the Trashcan mechanism is set to 1.

There are two Trashcan mechanisms at the end points of conveyors in the system. Workpieces will be removed, when they reach the end positions of conveyors. Trashcan has no visualisation. But we can assign material for the Trashcan, and then model it with geometric primitives, so we will get the object with visualisation. In further improvements, this mechanism can be exchanged on the robot, which will take the workpiece away from the conveyor.

1.3.12 Grip and Gripper Points

Grip Point is an element of an object's construction, which belongs to the section of an object and can be grasped by gripper points. Objects with grip points

can be grasped by robots or transported by the conveyor belt. Robots have to be equipped with gripper points necessarily. Opening and closing gripper are controlled by connections between inputs and outputs of gripper and robot-controller. By easy words, the grip point makes an object “graspable”. One grip point can be grasped by exactly one gripper point and vice versa, except for some extended mechanisms.

For easier visualizing, imagine the human hand. It plays the role of the gripper point. Just like a hand, a gripper enables holding, grasping and handling an object and consequently manipulates it. So, an object will play a role of grip point.

The grip range determines the distance to the origin, where the grip point is active.

Gripper Point is a component of an object’s construction, which belongs to the section of an object and can grasp the grip points. Gripper points can be attached to a robot or conveyor.

There are two types of gripper connections:

- Dynamical gripping is similar to temporary or permanent grasping an object. A dynamical gripping connection can be established or released during the simulation. For example: grasping and releasing an object with robot’s gripper
- Static gripping is similar to screwing objects together in the real world

Grip points and Gripper points are used to create functional connections between objects. If an object **A** is grasped at the grip point **GP_A** by a gripper point **GP_B** of an object **B**, these objects are functionally connected to each other. Moving of the grasping object **B** is a result of corresponding movement of object **A**. Besides positions, grip and gripper points have an orientation. The orientation is given by Roll, Pitch and Yaw values in the Properties window. This is used to provide a correct alignment between two connected objects.

Common applications of using grip and gripper points:

- The Replicator creates objects on the base of corresponding templates. The orientation and position of these objects are defined by the first grip point: objects will be placed with the first grip point exactly covering the gripper point of the replicator
- The Trashcan uses grip and gripper point mechanisms to remove objects from the workcell
- Active surfaces automatically create a dynamical connection with grip points. An active surface can grasp more than one grip point at the same time. This mechanism is used for palletizing objects or for the conveyor belt mechanism, which automatically grasps an object's grip point and moves the object until the grip point has reached the belt's end

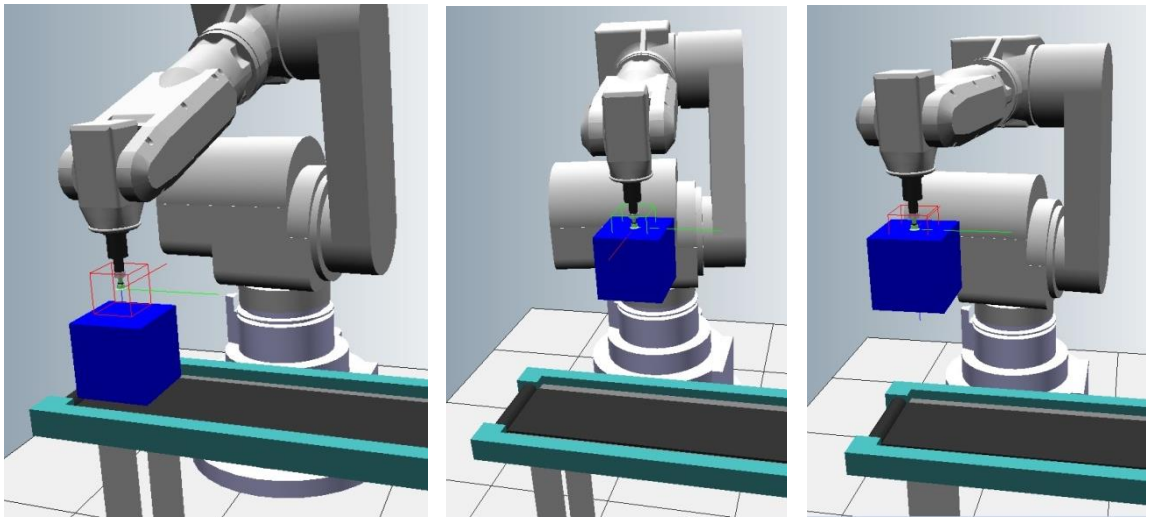


Figure 1.17 Grip and Gripper points of Robot and Cube

Figure 1.17 shows the example of grip and gripper points. On the first picture, there is a gripper point of the Vacuum gripper. On the second picture, there is a grip point of the object "Cube". This grip point is a place where the object has to be grasped. The third picture shows how the gripper point of the robot grasps the place of the grip point of the object. Then the robot will continue motion according to the certain sequences.

1.3.13 Buttons

A button plays a role as a transmitter of information about the colour of workpiece to the replicator, which will reproduce the patterns (see Figure 1.18).

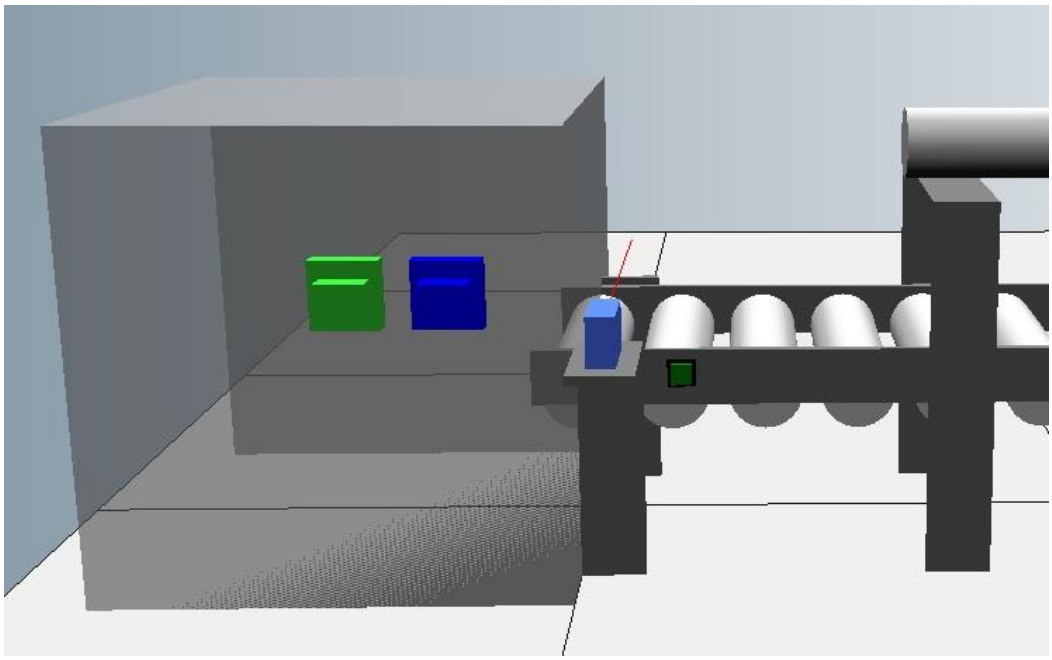
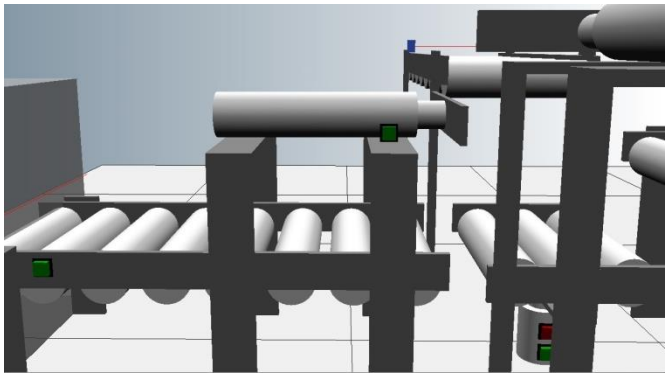
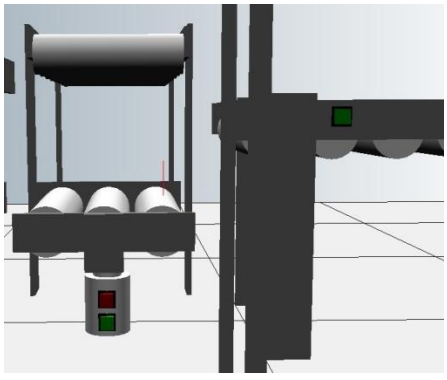


Figure 1.18 Buttons

1.3.14 LEDs

Light-emitting diode (LED) shows which object is in working mode. There are seven LEDs in the Roller Conveyor system (see Figure 1.19). Each LED is related to the certain object and to the certain motion of that object. For example, the Lifting Platform has two LEDs: one of them, the green colour LED, belongs to the translation motion, and the other, the red colour LED, is responsible for the rotation of the Lifting Platform.





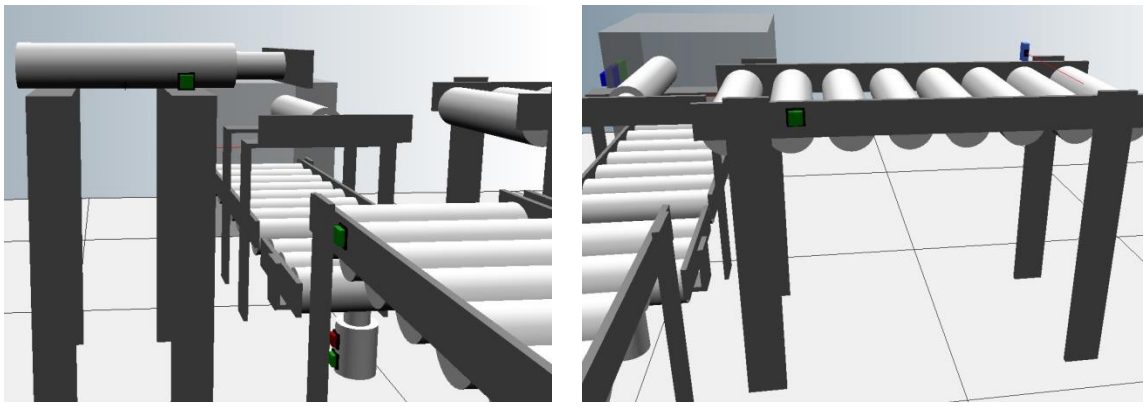


Figure 1.19 LEDs

1.3.15 IO Controller

IO Controller is a device which controls the simulation process. There are three parts of the controller: Inputs, Outputs and Memory. The controller obtains information about on-going processes via Inputs and reacts via Outputs. It can store information in Memory for the further monitoring. In CIROS Studio, assigning indexes to IOs is carried out via Controller. Depending on the amount of required IOs, there are some types of IO Controllers. The smallest one contains 8 bits, the biggest – 256 bits. The number of bits is related to the quantity of IOs. 16 bits Controller has been used, because there are 11 inputs and 10 outputs in the system.

CIROS Studio has such a complicated system concerning converting Inputs into Outputs and vice versa via Controller. For the first perception, it can seem meaningless, but there is one trick. Conveyor 1 has an Input “ConveyorBelt_1 BeltOn” and an Output “ConveyorBelt_1 PartAtEnd”. But from the PLC side of view, concerning the construction of sequences, the conveyor has to be switched, so it is the result of the certain motion. When the Workpiece reaches the end of the conveyor, the information about its arrival has to be used for the consequent motion. IO Controller converts the resultant function into defining function and defining function into resultant function. So, after conversion, the switching conveyor on action will be a result of certain action happened beforehand. And the part at the end action will be an input action which will entail some consequences. Figure 1.20 and Figure 1.21 show the index values of IOs and the relations between inputs and outputs in the system.

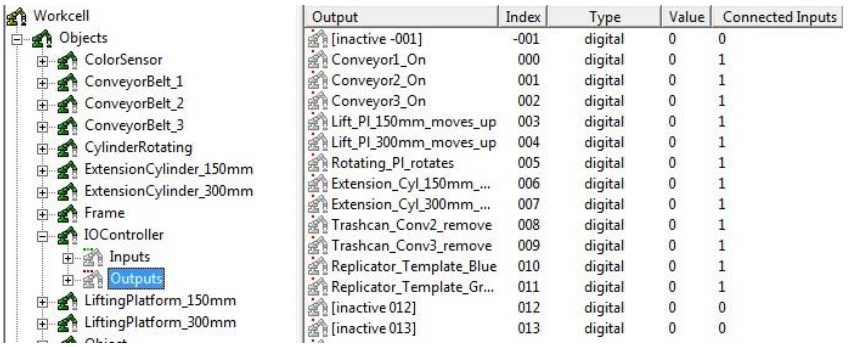


Figure 1.20 Index values of Outputs

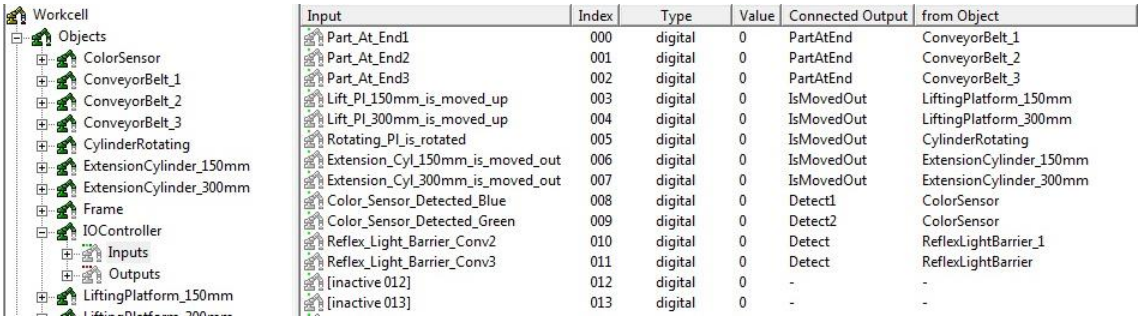


Figure 1.21 Index values of inputs and their connections with outputs

1.3.16 Sensors: Colour Detecting and Reflexing Light Barrier

The Colour Detecting Sensor performs detection of a colour of the Workpiece. Then, according to the defined colour, the system defines the direction of the next motion. This type of a sensor is established at the start of Conveyor 1 (see Figure 1.22). When the Replicator produces the Workpiece, it stands at the position which is reachable for the sensor. It requires less than one second for detecting the colour of the Workpiece. Then the Workpiece starts moving along Conveyor 1. When it reaches the Lifting Platform, it then goes to the certain path.

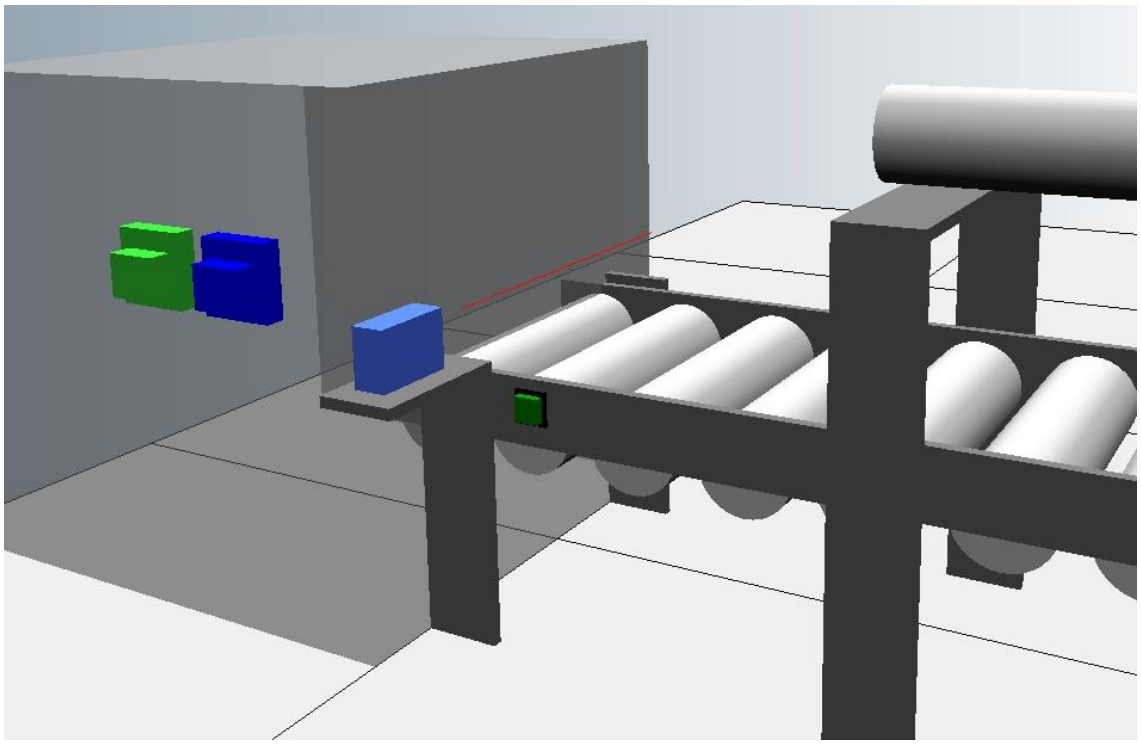


Figure 1.22 Colour Detecting Sensor

Reflexing Light Barrier reacts on the availability of the Workpiece at the certain position. One sensor is established at the end of Conveyor 2, and another is at the end of Conveyor 3 (see Figure 1.23). These sensors will detect the availability of Workpieces at the ends of the conveyors, for consequent removing them from there.

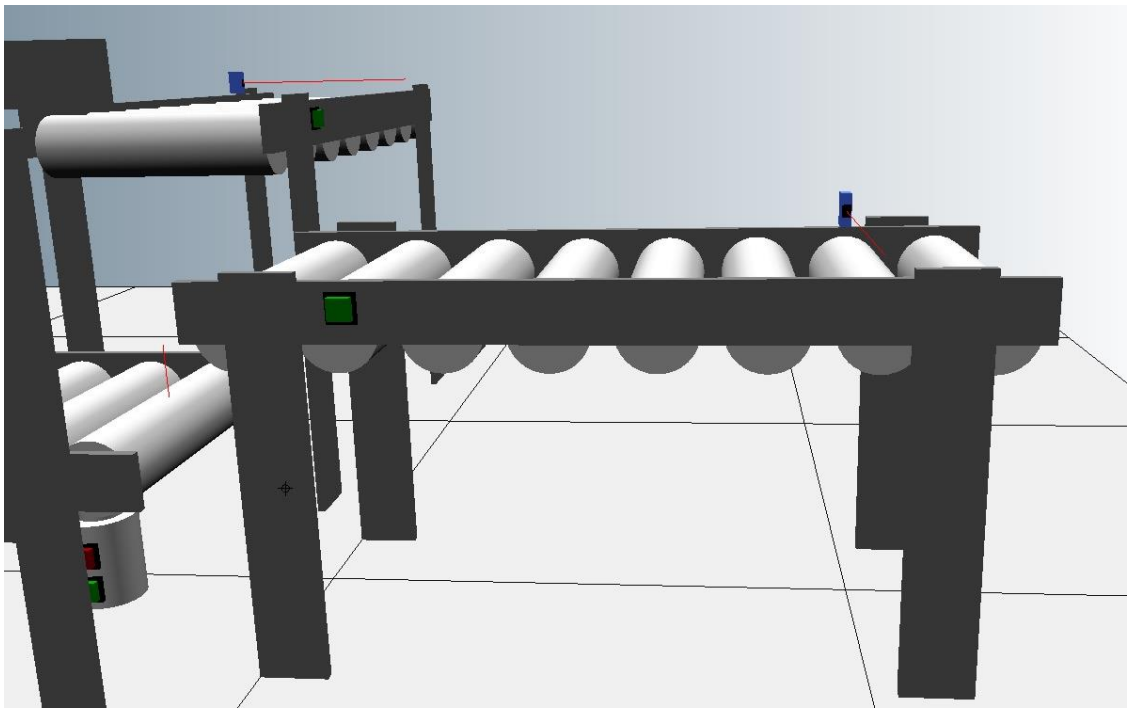


Figure 1.23 Reflexive Light Barriers

More precise information about sensors and their particularities will be discussed in Chapter 1.5.

1.4 Introduction to Automation Studio

Automation is a technology related to the mechanical, electronic and computer-based systems with the main functions to operate and to control production. Saving the labour is the biggest benefit of using automation. The main aims of using Automation technology are decreasing cost and time, but increasing productivity. Other Pros and Cons of Automation technology are presented in Table 3.

Table 3. Advantages and Disadvantages of Automation technology

Advantages	Disadvantages
Increased productivity	High probability of errors
Improved quality of the production	High initial cost, but investments in automation technology will pay you back in future
Increasing speed of production process	Good scope of knowledge and skills of worker for manipulating machines are required
Reducing human labour expenses	Limitations for tasks' performance
Growth of accuracy and precision	High potential of hazards
Safety	
Saves energy and materials	

All of these factors make Automation technology on implementable and attractive alternative to manual methods of manufacturing.

Applications for Automation technology:

- If there is demand for high accuracy and reducing cycle time
- Where human labour has to be replaced in case of inability applying manual work
- If there is dangerous environment for human being to work in

Definition of Automation technology by STEP 7:

The SIMATIC automation system is a wide diapason of coordinated components with unique approach of configuring, data management and data transmission. STEP 7 software increases efficiency in all of automation tasks. STEP 7 is used to configure SIMATIC components, assign parameters and subsequently program them. SIMATIC Manager in STEP 7 is the central tool for manipulating the automation data and required software tools. It keeps all data in a

project folder with a hierarchic structure, stores standard software and user software in libraries. SIMATIC S7 has several types of software: SIMATIC S7-200, SIMATIC S7-300 and SIMATIC S7-400. Each of them has own properties, features and special fields of applications. According to their properties and features, these types of software might be applied in certain applications.

- SIMATIC S7-200 is applicable for the low-end performance range, such as: autonomous solution or in linear-bus network
- SIMATIC S7-300 is suitable for usage in manufacturing industry
- SIMATIC S7-400 is the top-level device with the highest implementation, usable for manufacturing and process industries

The activities implemented by STEP 7:

- Configuration of the hardware: organizing modules in racks, assigning addresses to them and setting the module properties
- Establishing and maintaining communication connections
- Creating the user program for PLC by using programming languages, and then testing the program online by the controller

Many criteria exist for selecting the type of controller. The most considerable criteria are an amount of IOs and the size of the user program. Other points of consideration, such as the response time, the memory and the volume of data have to be taken into account in case of big production lines and factories. The user program includes the user instructions for implementing signals, which are responsible for manipulating the machine. Symbol table, instructions and blocks are used for creating programs. These constituents will be described more precisely later in this chapter and in chapter 1.7.

SIMATIC S7 automation systems are based on a central PLC. The solutions for the tasks are stored in the user memory of CPU represented as instructions. The CPU reads instructions in the certain order, according to hierarchy, and then interprets the content into actions. The CPU in PLC has to be linked to the machine. IO modules, connected to sensors and actuator in the machine are creating this link.

The system will be started, when the programming device is plugged in to the CPU and the user memory has the downloaded program inside. There is a possibility to monitor values of variables and observe the program flow, and it can be modified in future. These diagnostic and observation functions enable to define the location of errors and possible reasons of their presence rapidly. As well, some individual sections can be tested offline beforehand with the special software PLCSIM.

SIMATIC Manager is the central tool in STEP 7 world. Configuration of the hardware of the SIMATIC controller is executed by STEP 7: selecting the module addresses and parameters, and forming the network connections.

Advantages of using STEP 7:

- STEP 7 software runs on a standard PC with the Microsoft Windows OS, which is still widespread OS in a turnover
- STEP 7 has different versions
- STEP 7 is provided in five languages: English, German, French, Italian and Spanish
- STEP 7 requires small capacity of memory in the hard drive
- STEP 7 runs in 32-bit operating system
- STEP 7 allows to edit a project, which is located in a central server, from other workstations
- STEP 7 has an online help service

Programming languages:

STEP 7 allows writing programs for a controller in several languages, such as: Ladder Diagram, Function Block Diagram and Statement List. Depending on the task and its representation, there is a possibility to choose a more suitable programming language. LAD and FBD are excellent to apply for graphical representation of tasks. STL is suitable for execution of complex variables or using indirect addressing. These programming languages allow to operate with binary signals, but as well to convert digital values into other number formats. More information about programming languages will be given in Chapter 1.7.

The user can write a program by using one of the programming languages, and the software will convert it into other types of programming languages. So, the user does not need to write the same program in other languages by himself.

The program is written in a section called “blocks”. It joins the OS and user program. When the certain event occurs, then CPU calls these blocks, which define the starting point of the program in a specific priority of implementation.

User blocks:

- Organization blocks provide the communication between OS and the user program. The main program is located in OB 1. Other OBs, divided into various events, have fixed numbers. The execution of process is handled by priority of blocks
- Function blocks – programmable blocks with assigned parameters, which were given by user, according to its requirements. Parameters can be stored in permanent data blocks
- Functions – used to program cycle automation functions, but without storing data about them
- Data blocks – store the user program data

According to all information which was said above, a conclusion can be made, that the STEP 7 is suitable software for configuring hardware, installation of communications, programming, testing and consequent analysis of results, documentation and declaration of outcome.

1.5 Description of sensors

Sensor – is a device that detects some types of input from the physical environment (for example: temperature, pressure, speed, light, etc.) and converts it into the corresponding output, which can be measured electrically.

Criteria to choose a Sensor:

1) Accuracy – how far the given reading value can be from the true value
2) Range – the maximum distance that the sensor can detect

3) Calibration – comparison between measured values (readings changes with time)
4) Resolution – the smallest tolerance which can be detected by the sensor
5) Environmental condition – limits for temperature, humidity etc.
6) Cost
7) Repeatability - the repeatability of a sensor is a measure of its random accuracy. In general, the more accurate a sensor, the more repeatable it will be.

By considering these criteria we will choose a suitable sensor for the certain situation. There are different types and classifications of sensors according to their properties.

Proximity Sensor detects the availability of a Workpiece without physical contact on the nearby distance. Proximity Sensor emits electromagnetic fields, seeks out changes and the return signals. The most common examples of proximity sensors are inductive, capacity and photoelectric sensors. So, if the metallic part has to be studied, then **Inductive Sensor** is needed. The sensor detects the metal part and the value of sensor will turn to “1”. If there is a plastic part, then the sensor will not detect the part, because the workpiece does not emit magnetic field. So, the value of sensor will turn to “0”. The advantages of the Proximity sensor are a high reliability and long functional life (no physical contact between sensor and sensed object). The operating principle is based on a high frequency oscillator, which creates a field in the close to sensed surface surroundings. The presence of a metallic object can be detected by **Inductive Sensor**. Other material objects can be detected by **Capacitive Sensor**. It is happening in the operating area with causes of a change of the oscillation amplitude. These changes are identified at the starting point and entail a suitable output state of the sensor. The operating distance of the sensor depends on the actuator’s shape, size and the material of an object. A screw placed on the back of the Capacitive Sensor allows regulating the operating distance.

If the Workpiece has to be distinguished according to the different colours, then an **Optical Sensor** will be the best choice for usage. Being more precise, the

Colour Detective Sensor type is the most suitable one. This type of sensor detects the light rays and transfers it into an electronic signal. Colour Sensor detects a colour of the available object and compares that with a number of references of colours, configured in the extended properties of the sensor.

There are two colours, in the system. When the sensor detects the blue colour, then the system determines the certain path for the Workpiece: the blue colour Workpiece has to be transferred to Conveyor 2. If there is a green colour Workpiece, then the system understands that it should be moved to Conveyor 3.

Another type of sensor which was used in the system is **Reflexive Light Barrier**. It reacts on the availability of the Workpiece. It is a proximity switch, so it is contactless and it can react on the presence of the Workpiece at the certain distance. When the sensor detects a Workpiece, then the value of the sensor turns to “1”. And the initial red colour of the Reflexing Light becomes the green. It shows that, the Workpiece has been successfully detected.

This type of sensor is established at the ends of Conveyor 2 and Conveyor 3, and on the Lifting Platform. Reflexive Light Barrier on the Lifting Platform is used instead of non-working Output of Conveyor 1. Reflexing Light Barrier sensors in the ends of conveyors are used for activating the Trashcan mechanism. It sends information, that the Workpiece reached the end of the conveyor, so it needs to be removed.

Another type of sensors is **Mechanical Switch.** This switch can be manipulated manually as well as automatically. It will act as a sensor, when it is operated by different processes, such as: changing temperature, flow, force, current and voltage. If the switch is manipulated manually, then it will play a role as a button. Mechanical Switch has only two positions: on and off. It cannot send other information about an object, which is detected. The part that implements the operating force to the contact is called an **Actuator**.

This type of sensor was used as a button, which sends information to a Replicator about the workpiece’s colour. The user has to press the button by himself for activating the system.

1.6 Sequences of motions

The best representation for better understanding the on-going process is an algorithm. It can easily sort motions into steps, which describe actions, to be performed by the system. To activate the system, the user needs to choose the colour of the Workpiece: blue or green. Then the system will decide by itself the path which has to be passed by the Workpiece. In the end, when the system has implemented all steps, the user has to repeat the actions, thus: to choose the colour again. Figure 1.24 presents the sequences of motions in the algorithm structure.

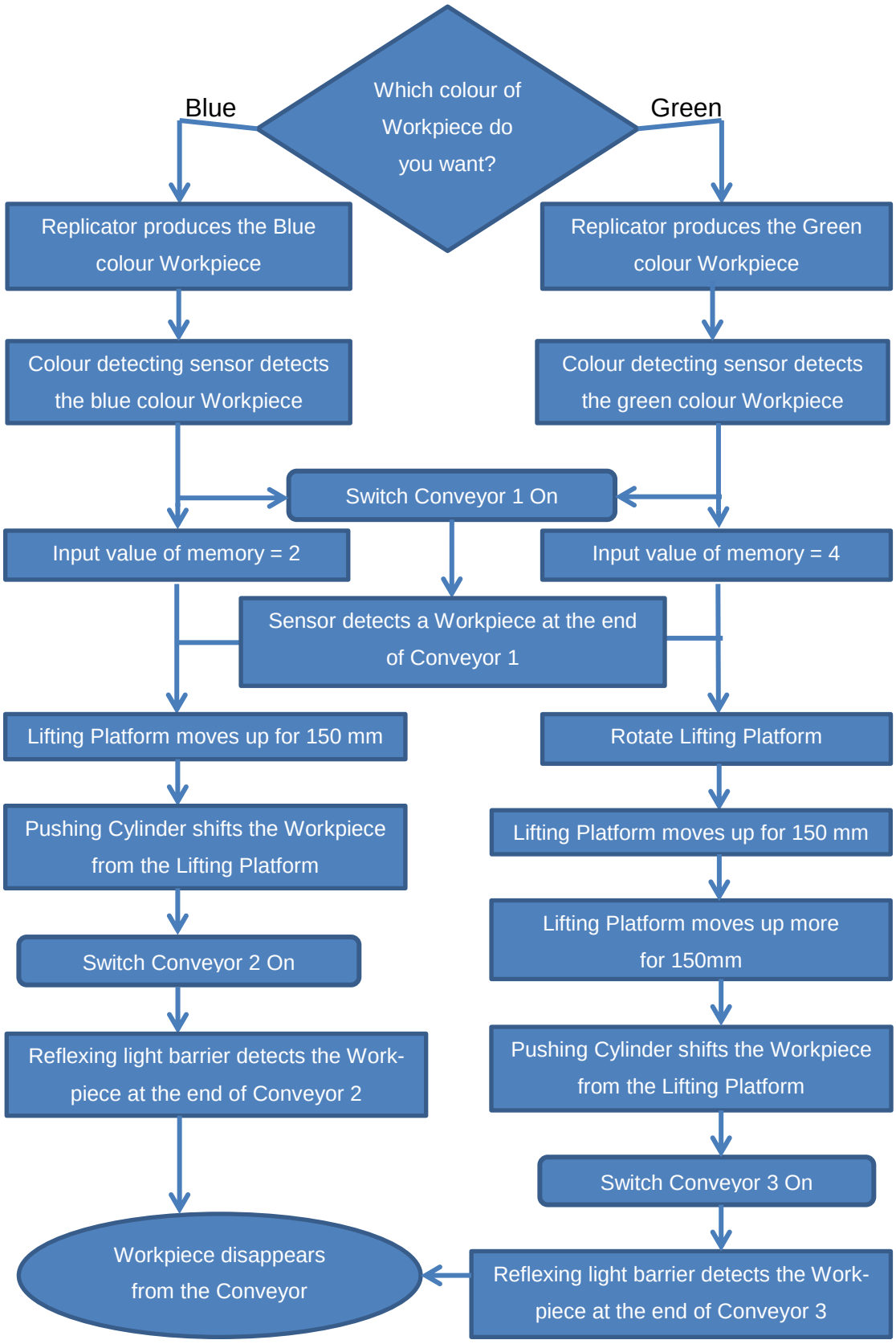


Figure 1.24 Algorithm of motions

1.7 Familiarizing with PLC and its components

This chapter introduces a logic structure based on SIMATIC Manager STEP7, which was inserted into the model. Virtual PLC embedded into the system provides connections between simulation and controller. I/O signals from the system were converted to PLC signals and a program based on LAD language was written to perform the workpiece transportation routes.

Programmable Logic Controller (PLC) is a special computer device which controls machines and processes. PLC includes a Central Processing Unit (CPU), which is indicated to run a program, monitoring different inputs and manipulating outputs by special logic, given by sequences for controlling a system. Also, CPU includes I/O interface modules, which are directly connected to the I/O devices. It uses a programmable memory to store instructions and specific functions that include On/Off controlling, timing, counting, sequencing, arithmetic and data handling. The program controls the PLC so, that, when an input signal from an input device turns ON, then the corresponding response is made. The response normally involves switching on an output signal of output devices. The difference between a PC and PLC is that PLC can handle multiple configurations and carry out control functions.

PLC is usually considered as a heart of the control system. It may be used for controlling a simple task, as well as complicated processes, which can be interconnected with other controllers or host computer through communication network. PLC is flexible to variety of processes. It defines the faults and removes errors quickly.

Input module converts the input signals into the signals understandable for PLC. Output module has the reverse function: it converts the PLC signals into signals understandable for actuators.

There exist hardware, firmware and software in the PLC. Hardware includes the actual technological device: integrated modules, wiring, power supply, circuit boards housing etc. Firmware is a software part installed permanently by PLC manufacturer and stored in ROM or EPROM. It includes basic routines of a sys-

tem, which is applied to start the execution of process after supplying the power to the system. Software is a program written by the user and stored in RAM with ability to be modified.

The cycle time is the time required by the PLC for each implementation of a program, including realization and output of the process image.

The specifications of the cycle process:

- When the program has been accomplished once, it automatically returns to the start. Then the process is repeated.
- As soon as the prior program line has been executed, the status of the inputs is stored in the image table. If the inputs are changed physically during the cycling, the status of the inputs stored in the image table stays constant.
- Outputs have the same way of working as inputs: setting and re-setting IOs do not happen immediately. All outputs will be physically switched in the end of a cycle. It will be realized according to the logic stored in memory.

Interpretation of program lines is carried out via CPU of PLC. The time, required for operating a process, can vary between few microseconds and a few milli-seconds.

The program created for specific applications demands a program memory. Consequently, it can be read by CPU cyclically.

Data types:

There are different types of data which can be used while creating the PLC for the system. For example: BOOL data type has a sequence with the length of 1 bit, BYTE has 8 bits and WORD has 16 bits. Other examples of data type are shown in Figure 1.25.

Keyword	*Data type*	*Range of values*
BOOL	Boolean number	0, 1
SINT	Short integer	0 to 255
INT	Integer	-32 768 to +32 767
DINT	Double integer	-2 147 483 648 to +2 147 483 647
UINT	Unsigned integer	0 to 65 535
REAL	Floating point number	+/-2.9E-39 to +/-3.4E+38
TIME	Time duration	implementation-dependent
STRING	Variable-long string	implementation-dependent
BYTE	Bit sequence 8	no range of values declarable
WORD	Bit sequence 16	no range of values declarable

Figure 1.25 Data types (6)

1.7.1 Pros and Cons of using PLC

Table 4. Advantages and Disadvantages of using PLC

Advantages	Disadvantages
Using Ladder logic for programming	Debugging PLC consumes a lot of time
Withstand a harsh industrial environment: humidity, heat, oscillation etc.	The operating system of PLC is smaller than a PC's OS
Handle much more complicated systems	Required well-skilled work force
Increased reliability: once a program has been written and tasted it can be downloaded to other PLCs	Parts and software from one manufacturer cannot be easily used in combination with parts of another manufacturer
Easy to troubleshoot	
Communication capability: PLC can communicate with other controllers or computer equipment	
Fast response time	
Reliable cost	
Have interface of I/Os already inside the controller	
Size of PLC varies from small to the large facilities, so reasonable space saving feature	
Possibility to be re-programmed	
Easier maintenance and programming of PLC comparing to relays: no great wiring	

Components of the PLC:

- Power supply
- CPU
- Programming device
- IOs modules
- Memory

1.7.2 I/O Configurations

PLCs have two main categories of I/O: fixed and modular. Both of them have the same basic functions. However, these two systems are very different in application use.

- Fixed I/O
    1) Typically for small PLC
    2) Comes in one package with no separate removable unit
    3) The processor and I/O packaged together
    4) Lower in cost, but lack of flexibility

It is designed to perform basic functions. Fixed PLC is compact and has less memory. The proper functioning of all processes depends on the execution of every component. Fixed PLC contains every component in a single unit, which includes CPU, I/O's sections and power supply. The number of inputs and outputs cannot be expanded. This type of PLC is shown in Figure 1.26.

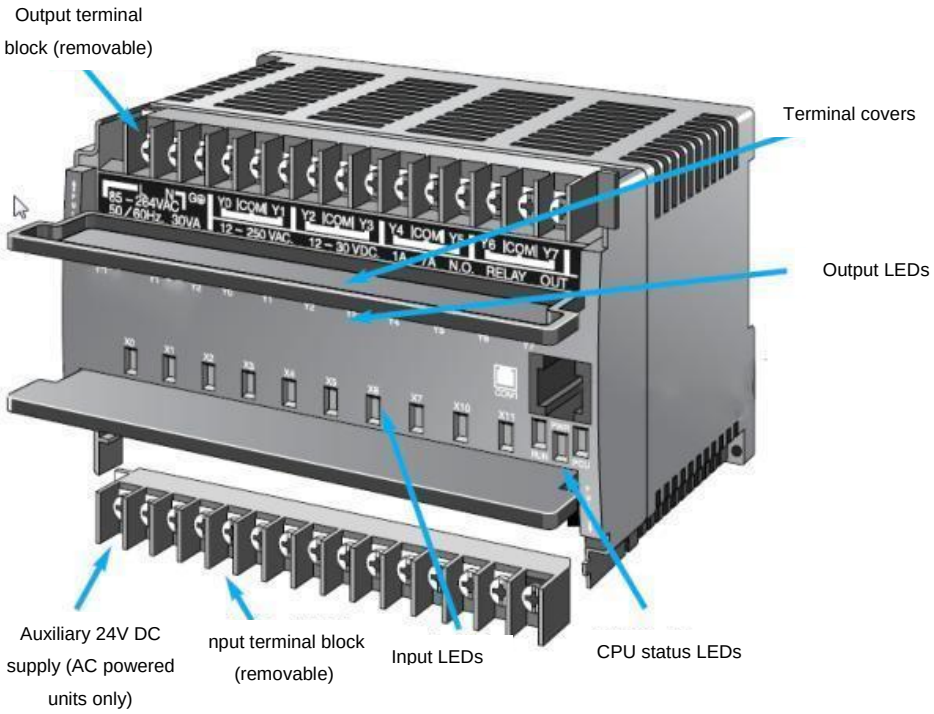


Figure 1.26 Fixed IO PLC type (7)

- Modular I/O

  1) It is divided by compartments into which separate modulus can be plugged in

  2) It increases your option and unit's flexibility

  3) When the module slides into the rack, it makes an electrical connection with series of contacts, called the backplane. The backplane is located at the rear of the rack

Also Modular PLC is known as the Rack-mounted PLC. The main advantage of Modular PLC is that the number of inputs and outputs can be expanded. These modules run along the backplane, and generate communication between CPU and modules. Modular PLC has separate constructions for the CPU, power supply and I/O system. The I/O system contains pluggable sections, so modules can be mixed and matched. Modular PLC is easier for troubleshooting and repairing. Figure 1.27 presents the Modular PLC type.

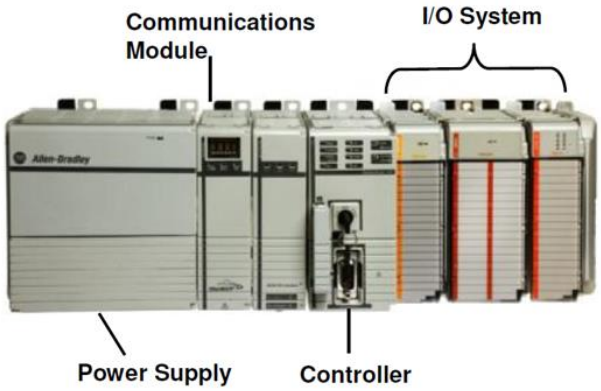


Figure 1.27 Modular IO PLC type (8)

Power Supply:

- Supplies DC power to other modules that plug into the rack
- In large PLC systems, this power supply does not normally supply power to the field devices
- In small and micro PLC systems, the power supply is also used to power field devices

Processor “CPU”:

- It is a brain of the PLC
- Consists of microprocessor for implementing the logic and controlling the communication among the modules
- Designed so the desired circuit can be entered in relay ladder logic form
- The processor accepts input data from various sensing devices, executes the stored user program, and sends appropriate output commands to control devices

I/O Section:

- Input modules
- Output modules

PLC Size Classifications:

- Number of Inputs and Outputs
- Physical size
- Cost

Depending on these features, a suitable PLC for a system can be chosen.

1. Nano PLC:
    - Smallest size PLC
    - Handles up to 16 I/O points
2. Micro PLC:
    - Handles up to 32 I/O points

Other types of CPUs and their features are presented in Table 5.

Table 5. Types of CPUs and their features

Element	CPU 312C	CPU 313C	CPU 313C-2 DP	CPU 313C-2 PtP	CPU 314C-2 DP	CPU 314C-2 PtP
Digital Inputs	10	24	16	16	24	24
Digital Outputs	6	16	16	16	16	16
Analog Inputs	-	4+1	-	-	4+1	4+1
Analog Outputs	-	2	-	-	2	2
Technological functions	2 counters	3 counters	3 counters	3 counters	4 counters, 1 channel for positioning	4 counters, 1 channel for positioning

Usually, PLC can be programmed by using five standard programming languages: FBD, Structured test (ST), LD, Instruction list (IL) and Sequential functional chart (SFC). These languages have different functions and structures.

LAD and FBD are graphic-oriented programming languages: connecting contacts in series and parallel arrangements in LAD, interconnecting AND and OR boxes in FBS. STL is a text-oriented high-level programming language which describes tasks in the form of a list. There is a possibility to convert a program written in LAD or FBD into STL format. Observation of the power flow in the rung is possible in LAD. Displaying logic operations by different colour is possi-

ble in FBD format. Testing the instructions in each line of the program and observing digital and binary values are possible in STL format.

1.7.3 Ladder Logic or Ladder Diagram (LAD)

Ladder diagram is a graphical technic of representation of the several logic elements. It is called “Ladder Logic”, because it looks like ranges on the ladder (see Figure 1.28). This is the most preferred programming method. The Time-diagram shows the actions of variables, how they change during the certain period of time. Ladder Logic method performs the sequential and logic controls.

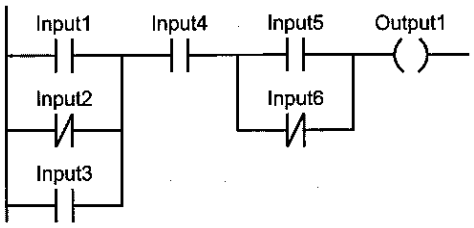


Figure 1.28 Representation of LAD programming language (9)

The various elements and other components are placed along the horizontal lines with rail contacts in the left side and with the coil elements in the right side of the diagram (see Figure 1.29).

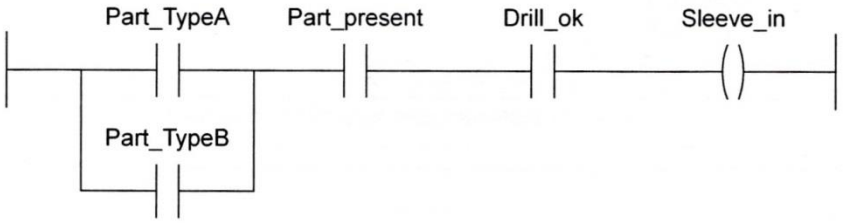


Figure 1.29 Example of sequence written in LAD (10)

Two possibilities for representing the inputs on the logic circuit are normally open (ON) and closed contacts (OFF). The roles of input are played by switches, ON/OFF sensors, relays and similar by functions binary contact devices. The roles of outputs might be played by electrical components, which can be switched ON and OFF by logic, such as: motors, solenoids etc. Some examples

of components, which can be used for programming in LAD language are shown is Figure 1.30.

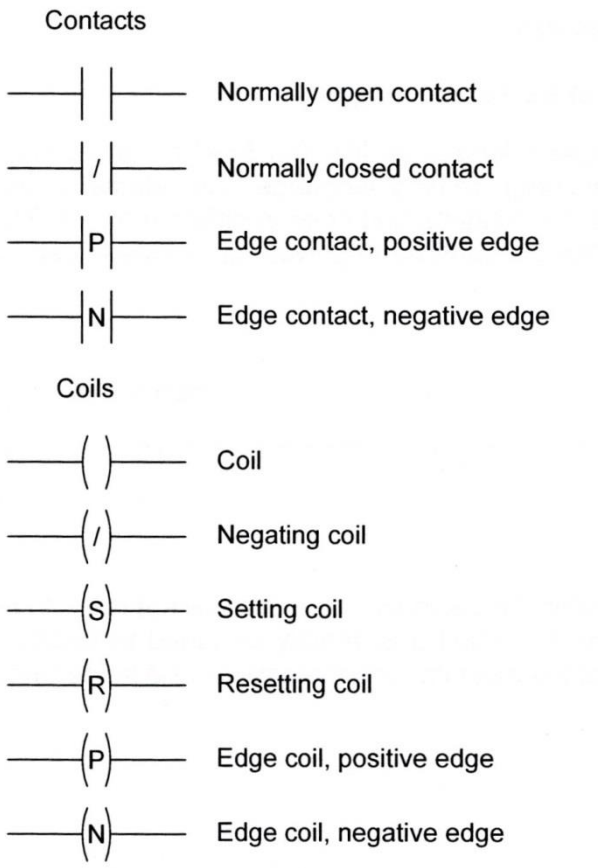


Figure 1.30 Components of sequence for LAD programming language (11)

Operations, such as: AND, OR and NOT can be interpreted in LAD. These functions are shown in Figure 1.31.

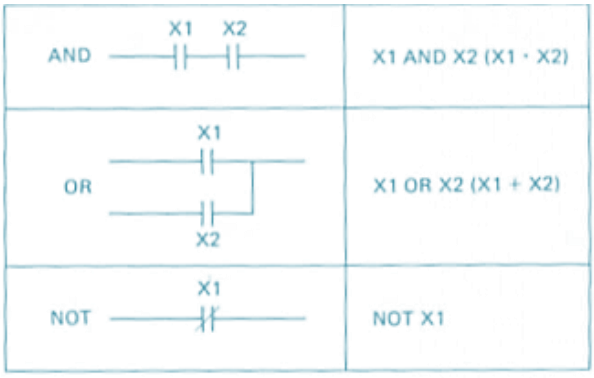


Figure 1.31 Operations in sequence written in LAD (12)

1.7.4 Function Block Diagram (FBD)

Another graphical representation of the program is Function Block Diagram. The function blocks and functions show the interconnection in network.

The various functions are programmed and have each box with a line linking the function to the output. Example of programming in FBD is presented in Figure 1.32.

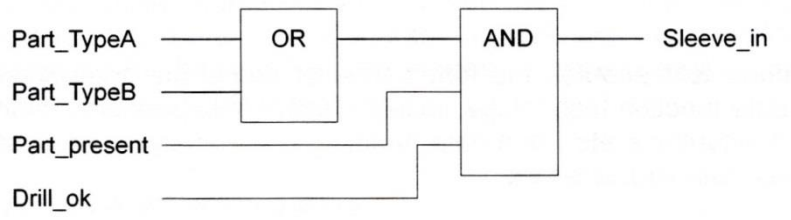


Figure 1.32 Example of sequence written in FBD programming language (13)

1.7.5 Structured text (ST)

It is the high-level programming language with the background of Pascal with text expressions and instructions. Algorithmic problems and data handling are the common applications of ST usage. Figure 1.33 shows the example of programming in ST programming language.

```
Sleeve_in := (Part_TypeA OR Part_TypeB) AND Part_present AND Drill_ok;
```

Figure 1.33 Example of sequence written in ST programming language (14)

1.7.6 Statement List (STL)

Another type of the programming language is STL. It is one of the most difficult languages. One example is given in Figure 1.34.

```
A(
O       Input1          //OR function
ON      Input2
O       Input3
)
A       Input4          //AND function
A(
O       Input
ON      Input
)
=       Output1         //Output assignment
```

Figure 1.34 Example of sequence written in STL programming language (15)

1.7.7 Instruction List (IL)

It is the most difficult programming language, which is used for repetitive actions and functions (see Figure 1.35).

```
LD   Part_TypeA
OR   Part_TypeB
AND  Part_present
AND  Drill_ok
ST   Sleeve_in
```

Figure 1.35 Example of sequence written in IL programming language (16)

1.7.8 Sequential chart

Sequential chart is a graphical representation of the control program, in reason to make it more clearly. The main idea of using the sequential chart is to structure a control program into the separate steps, interconnected to each other by links (see Figure 1.36). A Step includes a number of implementing parts of the control program, which are composed in action blocks. Steps have to have different names, even if the two steps have the same implementing parts. The reason is that the controller stores the information for each step. Steps set and reset values of IOs. The step's name is used for calling the step to perform the certain task and to have an access for step's data.

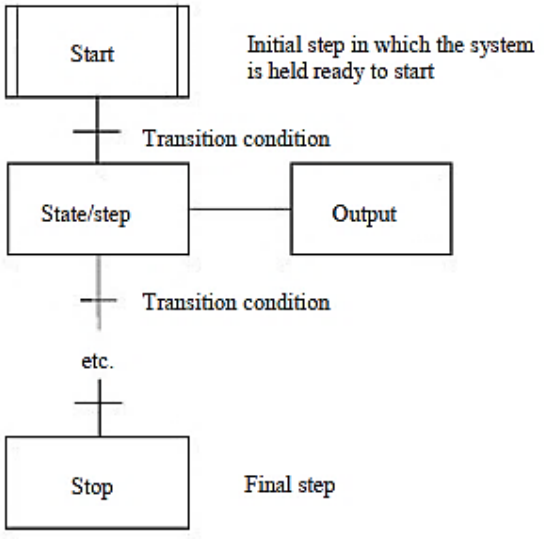


Figure 1.36 Sequential chart (17)

1.7.9 Timers

Timers are represented as software modules with generation of the digital timing. As soon as the input signal was obtained, the timer waits the estimated delay time, before turning ON the output signal. Turning OFF the input signal resets the timer. Delay time is established in the control program according to user's requirements. Representation of timer values is T# and followed by the time characteristics such as: days, hours, minutes, seconds and milliseconds. Example: T#15M34S – wait for 15 minutes and 34 seconds.

Three types of timers:

- Pulse timer

  Short input signal activates the timer. The output signal has fixed duration of implementation by time specifications, written in means of outputs.

- Time on-delay

  Input signal sends a signal to the timer about specific time about waiting for the activating the output. Output will be turned to value 1 after the certain period of time, and will hold this value until the input signal will turn to value 0.

- Time off-delay

  The output signal is turned to value 1 immediately after input activates the timer. When the input value turns to 0, the output value 1 is remained to the estimated duration. When the time is expired, the output value will turn to 0.

1.7.10 Counters

The main aim of using counters is detecting the number of pieces and events. If the production line has many identical parts, which are moving along the conveyor belt, then it is a good choice to use a counter for sorting device.

Two inputs are demanded for the execution of a counter. The first input counts the pulses received by the counter. The second resets the counter value and restarts the counting process. The collected values of counter are saved in memory, in case of applying them to the application in future.

Three different counter modules:

- Count-up counter

  The initial value of this type of counter is 0. This value will be increased by 1 with every positive edge at the input of counter. Simultaneously, the presented value is compared with the previously received value.

- Count-down counter

  This type of counter has already been set by user value. In future, this value will be decreased as soon as the counter receives the specified input. The output of the counter will be 0 until the value of counter is more than 0. When it turns to 0, then the output value will become 1.

- Count-up/down

  It is a combination of the count-up and count-down counters.

1.8 PLC Hardware settings vs. simulation

CPU is a microprocessor that co-ordinates the activities of the PLC system. It performs the program, handling I/O signals and communicates with external device.

1.8.1 Memory

There are various types of memory units. It is the area that holds the operating system and user memory. The operating system is actually a system software that co-ordinates the PLC. Ladder program, Timer and Counter values are stored in the user memory. Depending on the user's need, various types of memory are available for choice:

- Read-Only Memory (ROM) is a permanent memory that can be programmed only once. It is therefore unsuitable, it is least popular as compared with other memory types.
- Random Access Memory (RAM) is a commonly used memory type for storing the user program and data. This type of memory is fast and relatively cost effective. They can be modified and programmed freely. The main disadvantage of RAM is that the program has a capability to be lost after power failure. So, this memory requires a back-up such as battery or accumulator.
- Erasable Programmable Read Only Memory (EPROM) holds data constantly just like ROM. It does not require battery backup. However, its content can be erased by exposing it to ultraviolet light. It is the low cost and fast memory, but unstable. The initial information store in the memory has to be deleted firstly, then, after cooling period, it can be re-programmed. The specific device for erasing process, and the special programming unit for programming are needed. EPROM is used at the last stage of the commissioning. The program firstly stored in RAM will be re-stored in EROM.

- Electrically Erasable Programmable Read Only Memory (EEPROM) combines the access flexibility of RAM and the non-volatility of EPROM in one. Its contents can be erased and reprogrammed electrically, however, to a limited number of times.

Demands for a program memory:

- High efficiency
- Rapid response, without delay, to operations of the CPU
- Safety, to be assure that the program will not be lost, even after the power failure or short circuit
- Possibility to be modified rapidly or to create the new version of a program and then store it in the programming device or PC

1.8.2 Introduction to IRL- programming code

Industrial Robot Language (IRL) is the programming language used to create programs to control simulations.

The following programming languages can be used in CIROS® Robotics for programming robots:

- Mitsubishi MELFA Basic IV robot programming language
- Mitsubishi MRL robot programming language
- Standardized industrial robot language (IRL DIN 66312)

When a program is written in IRL it then requires an additional Position list. Software for industrial robot consists of data objects and lists of instructions. Data and program are usually located in separate sections of the robot controller memory. One can change a data without changing the program and vice versa. For example: one can write different programs using the same position list, or one can adjust positions without changing the programs that it uses.

2 3D-computer graphic software products

Computer Aided Design (CAD) is an important part in industrial automation. It enables engineers to gain a comprehensive preview of the system, helps to improve the quality of the product and optimize the production time. The next section compares two programs for creating 3D models.

Reasons to use CAD:

- It helps to designer to perform the conceptual design of the product with reduced time, required for modelling, analysis and documentation
- It improves the quality of the design itself
- CAD system allows to do a more complex analysis of the design and then consider it with other alternatives
- Graphical representation of a result is much better, than manual drafting: higher definition of drawings and less drafting errors
- There is a possibility to create a manufacturing database: bill of materials table, dimensions of parts and geometrics of the designed target

The general process of design:

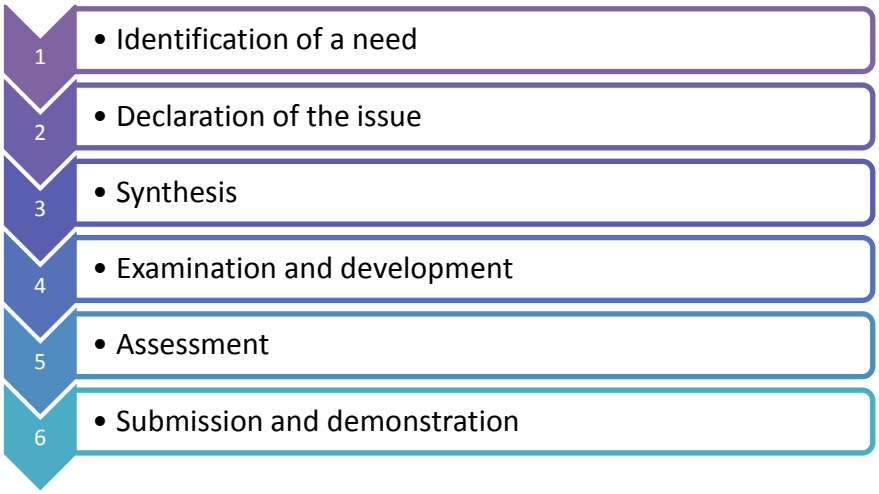


Figure 2.1 Designing process

2.1 Comparison between CIROS Studio and Blender

Evaluation of Festo CIROS Studio will be analysed, based on its performance, during the current chapter. The reason why these programs have to be compared is that they both specialize on animations and simulations.

2.2 Similarities and differences

CIROS Studio and Blender programs have their own features, different of each other, but they also have similarities. Some characteristics are described in Table 6.

Table 6. Comparison between CIROS Studio and Blender programs

CIROS Studio	Blender
CIROS Studio has already ready object for usage	In Blender you have to create every object from the ground
For creating an object, user can use prepared geometric primitives, such as: cube, sphere, cylinder, cone etc.	
An ability to enter PLC program into the workcell + many different languages to write a program for a system	The model can be programmed in languages, such as C, C++ and Python.
In the end, the model looks realistic	
Software is mostly in use for visualizing the processes	The program is used for creating special effect in movies and games
File format can be transferred to another	
Monitoring the results	Animating a human body and face motions
Merging different stations together	Building the big scene with many components

Connection with other similar software, such as: CIROS Mechatronics and Robotics	Leading physical experiments: mechanics, fluid- and aeromechanics, etc.
Recording a video about simulation process	Filming the scene or taking a picture with high quality

2.3 Advantages of using CIROS Studio

- CIROS Studio runs on all PCs based on Windows operating system. It makes the software more allowable for users to establish it on their own PCs, because nowadays, Windows is still a widespread OS in use.
- CIROS Studio enables you to create a detailed plan of an industrial manufacturing workcell, subsequently, test it on reachability of critical position, to develop robots and PLC programs and optimize a cell layout.
- All movements and handling processes can be simulated and checked on collision problems. It is good to test the system before using it in real life: to lead experiments in advance and to study case deeper and find further possible issues.
- Ability of controlling time while testing
- Optimizing the cycle time
- CIROS Studio has a library containing mechanisms, robots, tools, conveyor belts, feeders, controllers etc. This great storage gives a possibility for engineers to create a workcell with variable elements and then to simulate the system according to requirements.
- Importing from CAD systems gives a possibility to create new appearances of objects, which will be used in the certain system and production line. It is possible to do it via the standard data format STEP and STL.

3 Applications for the Roller Conveyor system

Possible future applications and works using CIROS Studio environment will be discussed in this chapter.

The main idea of this thesis was creating a workcell with an educational aim. Thus, students will use this system as a training simulation during the lectures and as a tutorial before writing their theses, if they are interested in the similar topic. The manual, attached to this thesis, will help students to build the workcell from the ground, and consequently, to animate the system with the help of programming.

The Roller Conveyor system can be used in the production line, because it can distinguish different objects and define their final destination. As well, this system can be presented as a middle-connection between two other systems. For example, if the pre-system is used instead of Replicator, which supplies different targets to the conveyor. And, if a subsequent system, such as Robot is established, at the end of conveyors, then the Workpiece will be taken away from the conveyors by the Robot, instead of just disappearing from the final position, as it is happening in the simulation now.

So if the same system is used in real life, then it is good that there is a possibility to test it beforehand, by using the constructed system.

4 Results and future plans

The most interesting part in the system is a Lifting Platform, because it is a complicated object, which includes different mechanisms, such as: two Cylinders for translation and one Cylinder for rotation. Complexity consists of correct connections between the grip and gripper points. The possible development is establishing one more mechanism into the Lifting Platform. This mechanism will work instead of the Pushing cylinders, which will shift the Workpiece from the Lifting Platform. This mechanism is called *Conveyor surface*. Wherein, the Lifting Platform base has been made of cylinders, as conveyors in the system.

As soon as a Conveyor Surface is established in the Lifting Platform, which will transmit the Workpiece to another Conveyor, an additional analog Input and Output for the Lifting Platform will be needed: input Conveyor_On and output Part_At_End. Another possible improvement is establishing a motor into the

Lifting Platform. Then, there is no need for *Cylinder for rotation* mechanism, instead of it there will be a *Motor* mechanism, which can rotate CW and CCW. Once the Pushing Cylinders are removed, the system will look more attractive, because the great Pushing Cylinders cover the view of the system. But it will work only for the 3D model. In real life, it is needed to be taken into account all features of the mechanisms, which might be applied instead of Pushing Cylinders, and make an analysis.

One more possible change in the system is about the Light Detecting Barrier at the Lifting Platform, which plays a role as a detecting device instead of output PartAtEnd1 of Conveyor 1. The case of non-working output of Conveyor 1 has to be studied. What are the reasons for the disabled output? One of the reasons might be an overlay of the conveyor surface on the next object, so, the *Conveyor surface* mechanism is bigger than the conveyor itself. Other reasons need to be considered in the future.

Reflexing Light Barriers at the end of Conveyors 2 and 3 will send info to the consequent station after the Workpieces reach the ends of conveyors: sensors play a role as inputs for the next motions. These sensors are needed to plug an extra station into this system: for example, some robots which will grab the Workpiece from one of the conveyors, and then, they will prolong the motion. With these extra stations we will get rid of the Workpiece, stopped at the end of conveyors.

Hydraulic components can be used in the Roller Conveyor system. It means there will be a possibility to create a circuit diagram, to get a feedback from the hydraulic side of view. The Pushing Cylinders, in the system, will be changed on the hydraulic cylinders in the real-time system. The Lifting Platform will include the hydraulic cylinder, instead of base.

4.1 Difficulties during the work

The main difficulty was learning the software from the ground only with the help of manuals. Other examples of difficulties which were met are presented below:

1) Rotating motion of the Lifting Platform for 90 degrees CCW around Z-axis
2) Lifting Platform has 2 stages where it has to stop, because of different heights of subsequent conveyors (see Figure 5.1)

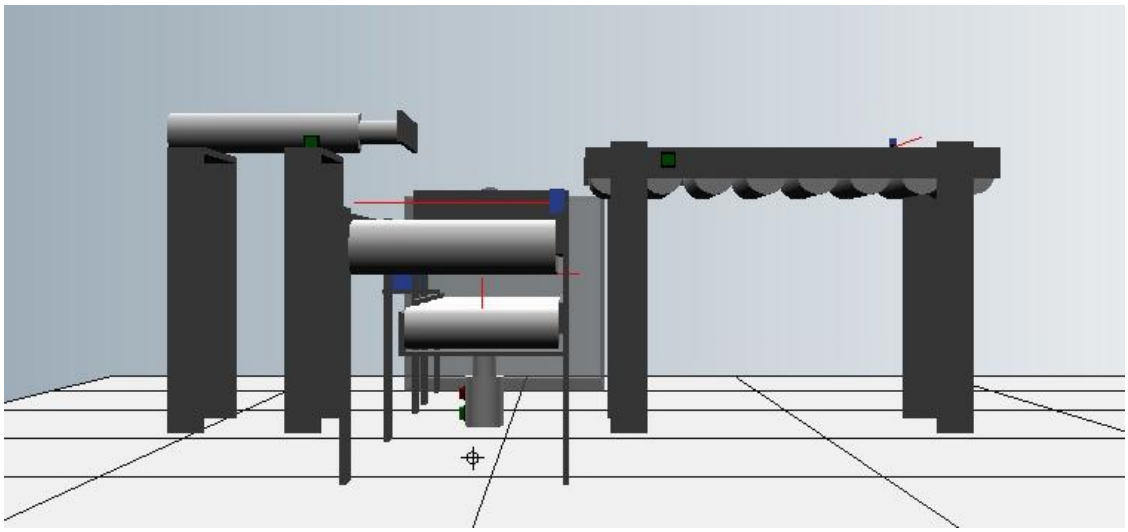


Figure 4.1

3) Configuration of the length of cylinder's piston of the Lifting Platform (see Figure 5.2)

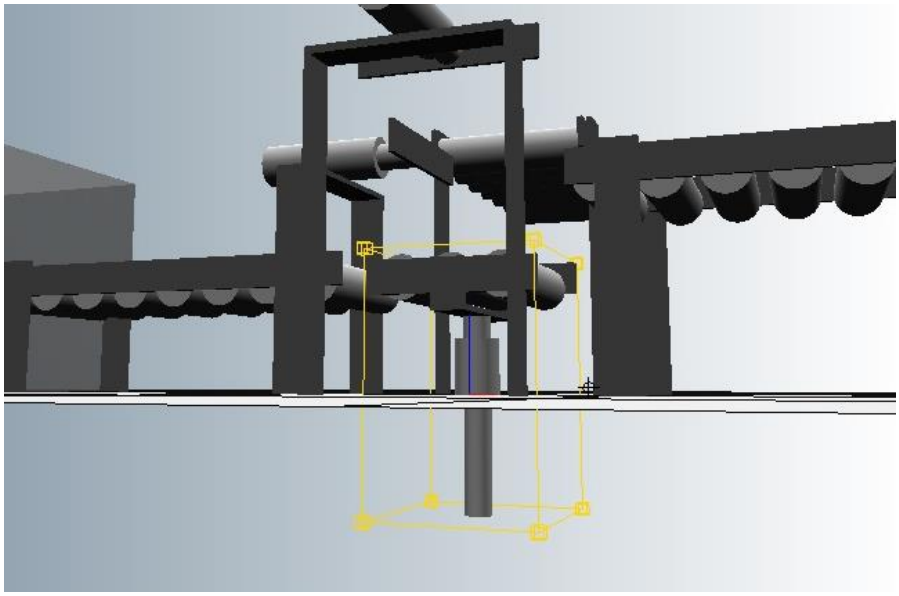


Figure 4.2

4) After connecting the rotating mechanism to the Lifting Platform, it starts rotating around a certain point, but not around the original Z-axis. Lifting

Platform is shifted to the left for a few millimetres during the rotating motion (see Figure 5.3).

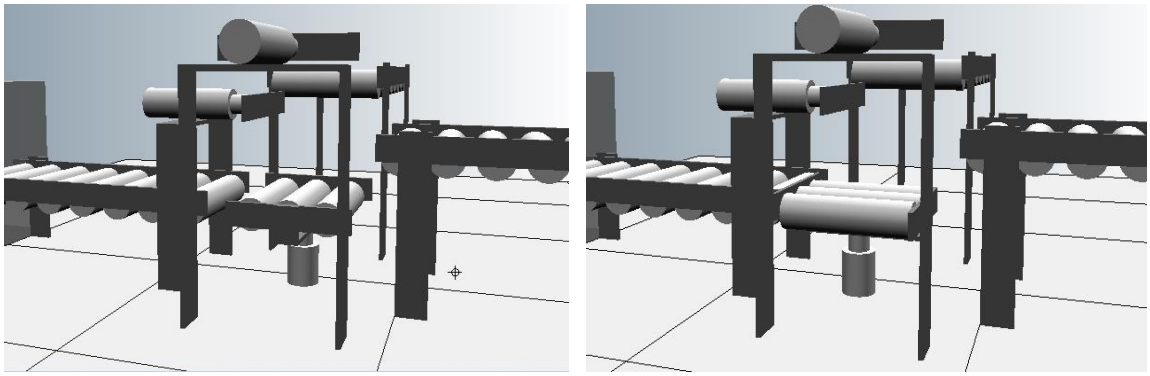


Figure 4.3

5) The workpiece does not want to be rotated on the Lifting Platform (see Figure 5.4).

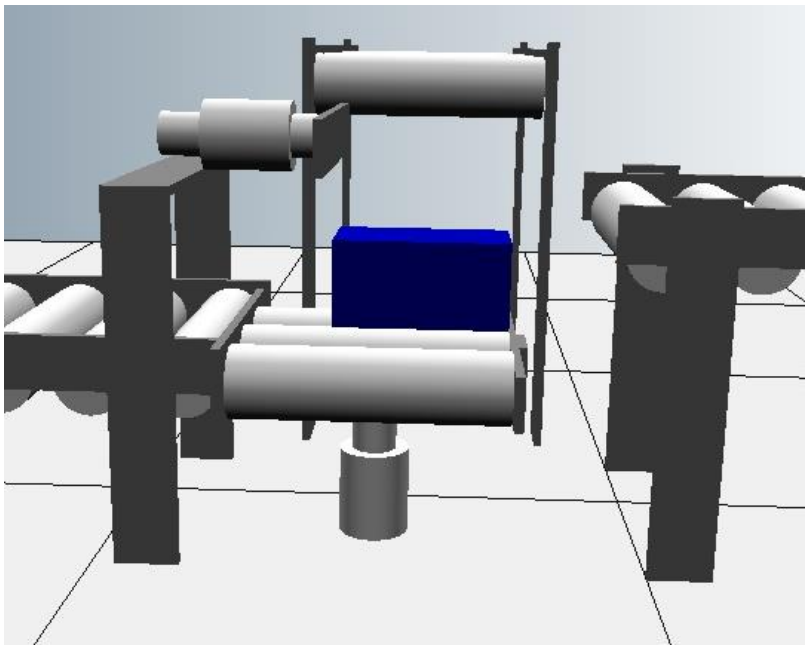


Figure 4.4

6) The piston of the cylinder established in the Lifting Platform moves with the base (see Figure 5.5).

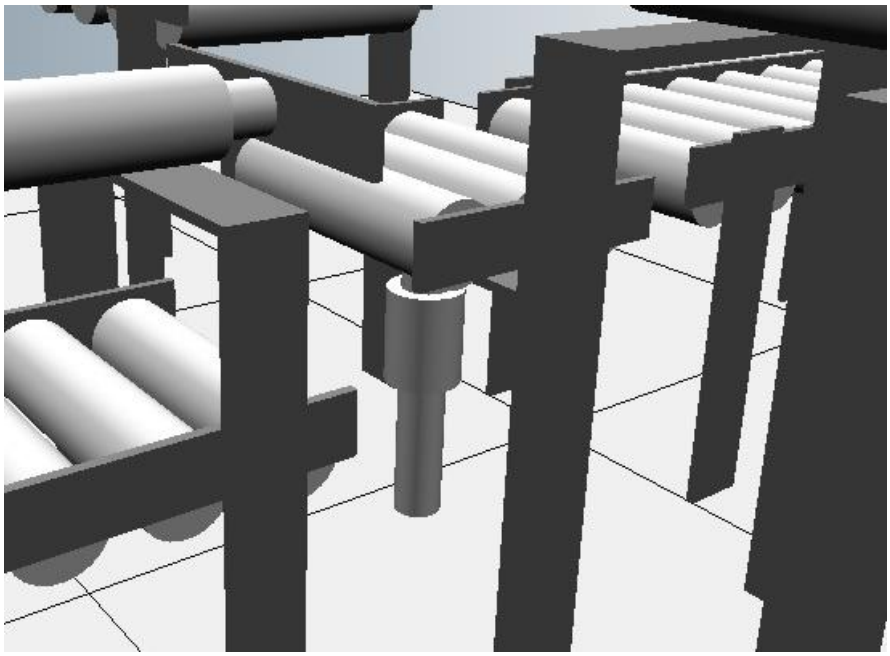


Figure 4.5

7) Output “PartAtEnd1” of Conveyor1 does not react, when Workpiece reaches the end of conveyor

The solutions for these issues are described in Chapter 5.2.

4.2 Solutions

During the design of the objects included in the workcell, some problems were met. The issues were described in Chapter 5.1. Each number of solution matches to the number of issue. The solutions which were found for the issues are described below:

1) There is a possibility to combine mechanisms like cylinders and rotating devices via static grip/gripper points’ connections
2) Adding extra Cylinder Translation mechanism into the Lifting Platform is a solution how to stop the Lifting Platform at 2 stages. This mechanism has to be connected to the static gripper point. Now, there are 2 inputs for the Lifting Platform. The first one is connected to the height of 150 mm; the second one is related to the height of 300 mm.
3) In reality, there has to be a cut hole in the floor for cylinder. But this is the 3D model, so we can skip some decorating details.

4) The reasons which caused these problems are connected to the grip and gripper points. Firstly, make sure that the grip point is established in the centre of a cylinder. Then, the gripper of rotating cylinder has to be set at the original point of the cylinder. Also, the coordinate system has to be changed form World to Section coordinate system
5) Here is the same problem with grip and gripper points. The best solution was to add the acting surface in Pushing Cylinder system, aligned with grip and gripper points' positions. An *Active surface* mechanism is used instead of grip and gripper points.
6) One part from an object can be transferred to another. It is called *Hull*. So the base of the cylinder of Lifting Platform object was transferred into another object, which is called Rotating Cylinder. A rotating mechanism has not visualisation, so it does not influence on the appearance of the Lifting Platform, and does not change the way of acting of Lifting Platform
7) Output PartAtEnd1 does not work, because, the Conveyor surface is laying over the Lifting Platform, so it covers another object of the workcell. That is why one more sensor is established, a light detecting barrier, which detects the availability of a Workpiece on the Lifting Platform. This sensor reacts only on the Replicator produced Workpieces. Even if the Lifting Platform is going through that sensor, it will not react on that motion.

5 Conclusion and analysing results

Finally, the simulation was completed by downloading PLC into the workcell. Everything was working correctly: without errors. Workpieces are detected according to the initial plan: the blue colour Workpiece is moving to Conveyor 2, the green colour Workpiece is going to Conveyor 3. The Colour Detecting Sensor defines the certain path, where the Workpiece has to be transferred.

In general, the work went successfully, however there were some difficulties in the start of the learning process. The most labour-intensive processes were designing the 3D model of the Roller Conveyor system, especially the elements of Lifting Platform, and writing a PLC program for the simulation. It required paying

attention to some details, which have influence on the operation process. As was already said before, the Roller Conveyor system was created in education aim, so it can be used by students for training their skills in 3D-modeling, programming, simulation and animation fields. The manual override, which shows the connections between inputs and outputs, IO table and PLC program for the Roller Conveyor system and hardware settings are attached as the files in appendix. The manual, attached as a file in appendix, will help to the students to create the similar system as the Roller Conveyor system. It includes the detailed steps of creating the workcell.

I wish to express my heartfelt acknowledgment to my supervisors Timo Eloranta and Simo Sinkko. They have been guiding me during the whole process of writing this thesis: from exploring the simulation and animation worlds, and collecting information about Power Hydraulics until the applying obtained information in practice. We went through all difficulties together, and reached good results in the end.

In addition, I am thankful to Professor Dirk Pensky, from Festo Didactic in Germany, for his help with fixing issues in my simulation in CIROS Studio. We have had two online conferences with Dr.Dirk Pensky. He has helped me to fix some issues is the simulation and he has given me some useful hints for the consequent usage of this software. Also, I would like to congratulate Professor Dirk with been appointed to the position of a Head of Software Engineering department, and to wish him all the best and a success in reaching new goals in future.

References

Art systems, 06.2006. Festo FluidSIM 4 Hydraulics. User's Guide. Festo Didactic, Art Systems Software, Eaton Corporation. [Accessed 30 March 2015]

Beth A.Schroeder, 1995. On-Line Monitoring: A Tutorial. Computing Practices. State University of New York. [Accessed 02 April 2015]

Charles McLean and Swee Leong, "The role of Simulation in Strategic Manufacturing". [pdf] National Institute of Standards and Technology. Gaithersburg. Available at: <http://www.mel.nist.gov/msidlibrary/doc/ifip5.pdf>. [Accessed 25 March 2015].

Christine Löffler, 01.2010. Festo CIROS Mechatronics. Manual. Festo Didactic, Germany. [Accessed 27 February 2015]

Concentric. Advantages and Disadvantages of Simulation. [online text] <http://www.concentricabm.com/blog//advantages-and-disadvantages-of-simulation>. [Accessed 22 April 2015].

D.Merkle, B.Schader, M.Thomes, 1998. "Hydraulics: basic level", TP 501 Textbook, by Festo Didactic

EngineersGarage. 02.03.2011. Sensors: Different Types of Sensors. [online text] <http://www.engineersgarage.com/articles/sensors#>. [Accessed 11 March 2015].

E-University. Programmable logic controllers. IL, SFS and ST programming methods. Sequential Function Charts. [online text] <http://www.wisdomjobs.com/e-university/programmable-logic-controllers-tutorial-523/il-sfc-and-st-programming-methods-2446/sequential-function-charts-14695.html>. [Accessed 29 April 2015].

Hans Berger, 2003. “Automating with SIMATIC”, Integrated Automation with SIMATIC S7-300/400, Controllers, Software, Programming, Data Communication, Operator Control and Process Monitoring. Second revised edition

Idaho State University. Continuing Education/Workforce Training. 19.08.2014. Advantages of Modular PLC over a Fixed PLC. [online text] <http://blog.cetrain.isu.edu/blog/bid/353287/Advantages-of-a-Modular-PLC-Over-a-Fixed-PLC>. [Accessed 02 April 2015].

Jerry Banks, 1998. “Handbook of Simulation”, Principles, Methodology, Advances, Applications and Practice [e-book]. Available at: <https://books.google.com/>. Canada

John W. Fowler and Oliver Rose, “Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems”. [pdf] <http://www.thesimguy.com/GC/papers/scsgc-02.pdf>. [Accessed 25 March 2015].

Mikell P.Groover, “Automation, Production systems, and Computer integrated manufacturing”, Prentice-Hall International Editions

R.Bliesener, F.Ebel, C.Löffler, B.Plagemann, H.Regberg, E.v.Terzi and A.Winter, 08.2002, “Programmable logic controllers: basic level”, TP301 Textbook, Festo Didactic

Robert E. Shannon, 1992. “Introduction to Simulation”. [pdf] A winter simulation conference. Available at: http://www.informs-sim.org/wsc92papers/1992_0003.pdf>. [Accessed 25 March 2015].

Siemens SIMATIC, 03.2006. Programming with STEP 7. [pdf] Manual. Siemens, Germany [Accessed 03 February 2015]

U.Karras, 01.2010. Festo CIROS Robotics. [pdf] User’s Guide. Festo Didactic, Germany [Accessed 23 January 2015]

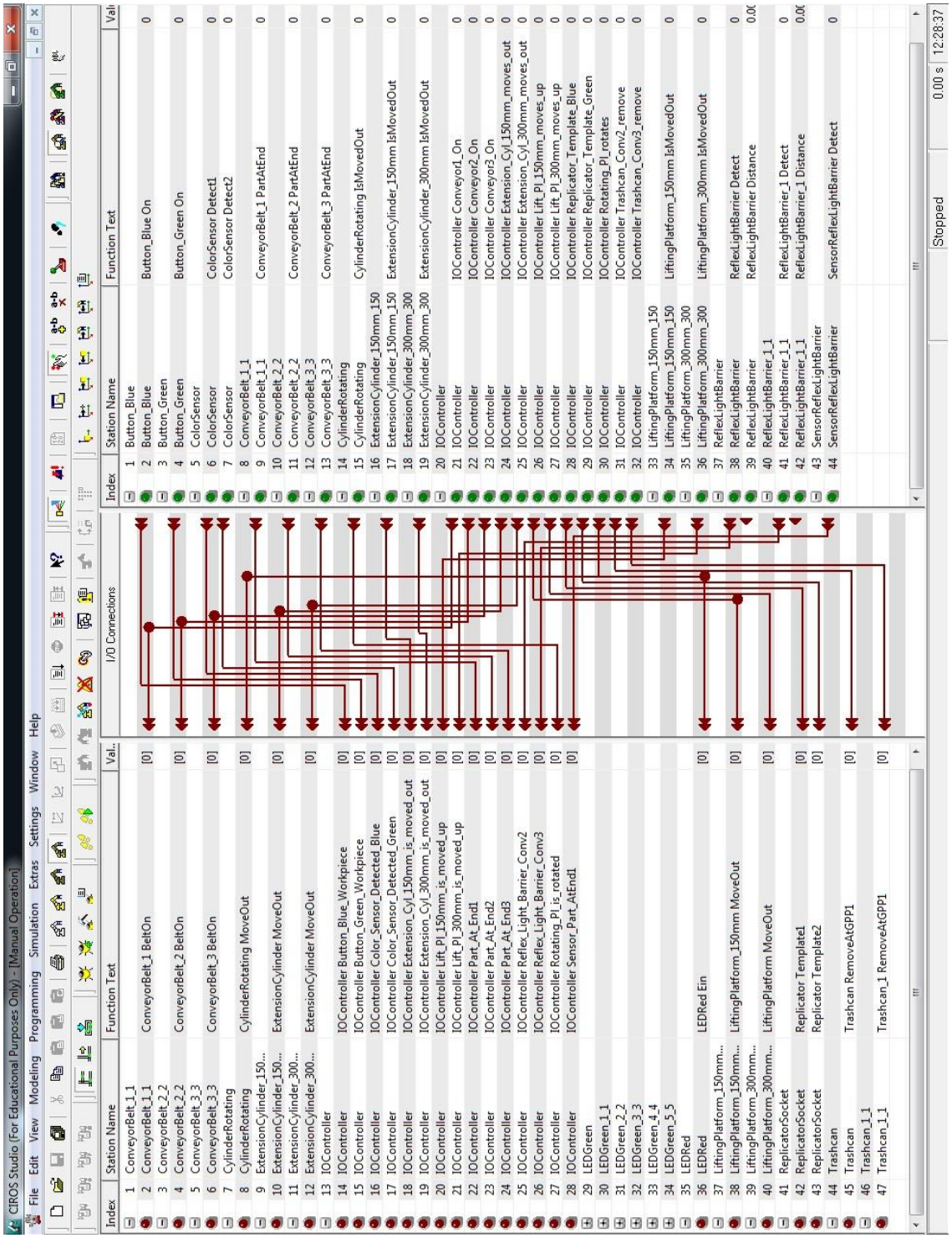
U.Karras, 08.2008. Festo CIROS Studio 1.0. [pdf] User’s Guide. Festo Didactic, Germany. [Accessed 31 January 2015]

Sources of Figures:

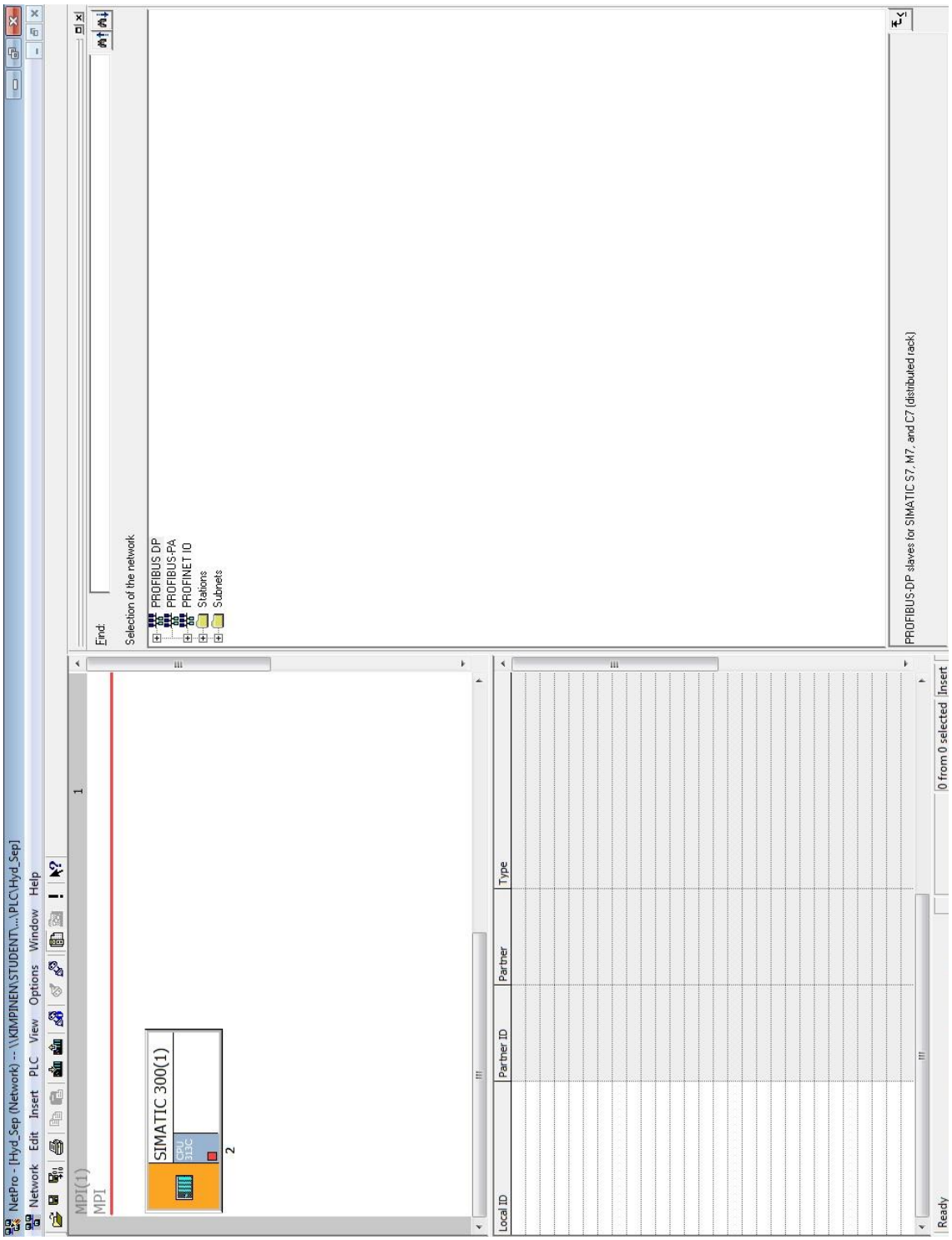
1. Manual of Fluid SIM Hydraulics software, Festo Didactic
2. D.Merkle, B.Schader, M.Thomes, 1998. “Hydraulics: basic level”, TP 501 Textbook, by Festo Didactic; page 228
3. Manual of FluidSIM Hydraulics software, Festo Didactic
4. D.Merkle, B.Schader, M.Thomes, 1998. “Hydraulics: basic level”, TP 501 Textbook, by Festo Didactic; page 228
5. http://agylesafetytrainingghana.com/dir/view-content/28/Excavator-Training.html
6. R.Bliesener, F.Ebel, C.Löffler, B.Plagemann, H.Regberg, E.v.Terzi and A.Winter, 08.2002, “Programmable logic controllers: basic level”, TP301 Textbook, Festo Didactic; page 64
7. http://blog.cetrain.isu.edu/blog/bid/353287/Advantages-of-a-Modular-PLC-Over-a-Fixed-PLC
8. http://blog.cetrain.isu.edu/blog/bid/353287/Advantages-of-a-Modular-PLC-Over-a-Fixed-PLC
9. Hans Berger, 2003. “Automating with SIMATIC”, Integrated Automation with SIMATIC S7-300/400, Controllers, Software, Programming, Data Communication, Operator Control and Process Monitoring. Second revised edition; page 99
10. R.Bliesener, F.Ebel, C.Löffler, B.Plagemann, H.Regberg, E.v.Terzi and A.Winter, 08.2002, “Programmable logic controllers: basic level”, TP301 Textbook, Festo Didactic; page 54
11. R.Bliesener, F.Ebel, C.Löffler, B.Plagemann, H.Regberg, E.v.Terzi and A.Winter, 08.2002, “Programmable logic controllers: basic level”, TP301 Textbook, Festo Didactic; page 90
12. Hans Berger, 2003. “Automating with SIMATIC”, Integrated Automation with SIMATIC S7-300/400, Controllers, Software, Programming, Data Communication, Operator Control and Process Monitoring. Second revised edition; page 659
13. R.Bliesener, F.Ebel, C.Löffler, B.Plagemann, H.Regberg, E.v.Terzi and A.Winter, 08.2002, “Programmable logic controllers: basic level”, TP301 Textbook, Festo Didactic; page 55

14. R.Bliesener, F.Ebel, C.Löffler, B.Plagemann, H.Regberg, E.v.Terzi and A.Winter, 08.2002, “Programmable logic controllers: basic level”, TP301 Textbook, Festo Didactic; page 56
15. Hans Berger, 2003. “Automating with SIMATIC”, Integrated Automation with SIMATIC S7-300/400, Controllers, Software, Programming, Data Communication, Operator Control and Process Monitoring. Second revised edition; page 99
16. R.Bliesener, F.Ebel, C.Löffler, B.Plagemann, H.Regberg, E.v.Terzi and A.Winter, 08.2002, “Programmable logic controllers: basic level”, TP301 Textbook, Festo Didactic; page 55
17. http://www.wisdomjobs.com/e-university/programmable-logic-controllers-tutorial-523/il-sfc-and-st-programming-methods-2446/sequential-function-charts-14695.html

Manual override of designed application



PLC – program and settings



HW Config - SIMATIC 300(1)
Station Edit Insert PLC View Options Window Help
SIMATIC 300(1) (Configuration) -- Hyd_Sep
(0) UR
Slot Module Order number Firmware MPI address I address Q address Comment
1 PS 307 2A 6ES7 307-1BA00-0AA0
2 CPU 313C 6ES7 313-5BE00-0AB0 V1.0 2 0...2 0...1
2.2 DI24/DO16 252...767 252...255
2.3 AI5/AO2 768...783 768...783
2.4 Count
3
4
5
6
7
Find:
Profile: Standard
PROFIBUS DP
PROFIBUS-PA
PROFINET IO
SIMATIC 300
SIMATIC 400
SIMATIC PC Based Control 300/400
SIMATIC PC Station
PROFIBUS-DP slaves for SIMATIC S7, M7, and C7 (distributed rack)
Press F1 to get Help.

SIMATIC Manager - Hyd_Sep
File Edit Insert PLC View Options Window Help
< No Filter >
Hyd_Sep -- \\KIMPINEN\STUDENT\...\taborova_elena\CIROS Studio\PLC\Hyd_Sep
Hyd_Sep
SIMATIC 300(1)
CPU 313C
S7 Program(1)
Sources
Blocks
SIMATIC 300(1)
MPI(1)

OB1 - <offline>

""

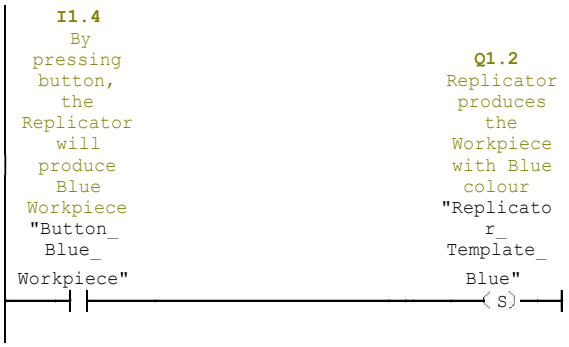
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 04/20/2015 12:11:58 PM
Interface: 04/09/2015 04:16:32 PM

Lengths (block/logic/data): 00358 00228 00020

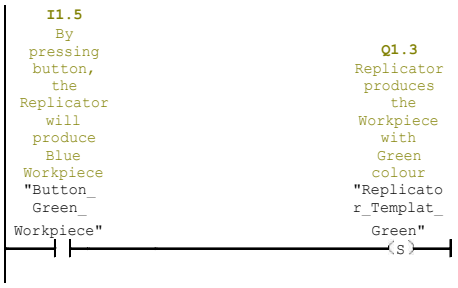
Name	Data Type	Address	Comment
TEMP		0.0	
Default	Array [1..20] Of Byte	0.0	

Block: OB1

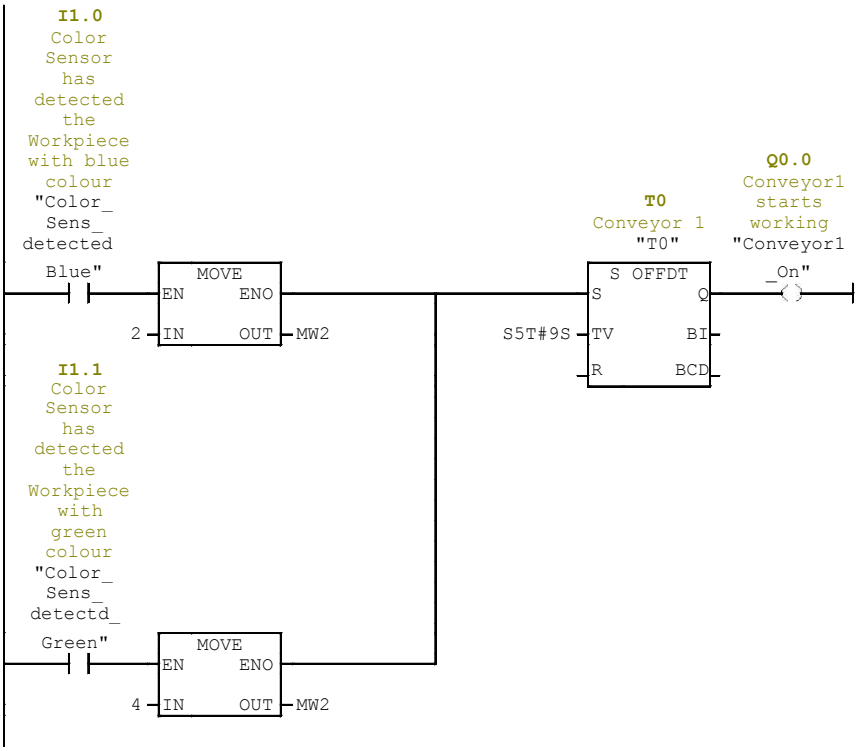
Network: 1 Replicator produces the Workpiece with Blue colour



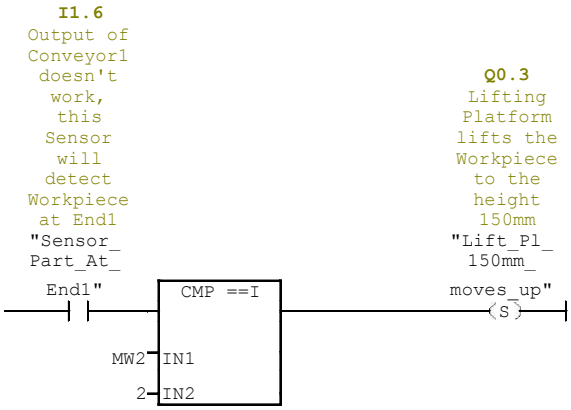
Network: 2 Replicator produces the Workpiece with Green colour



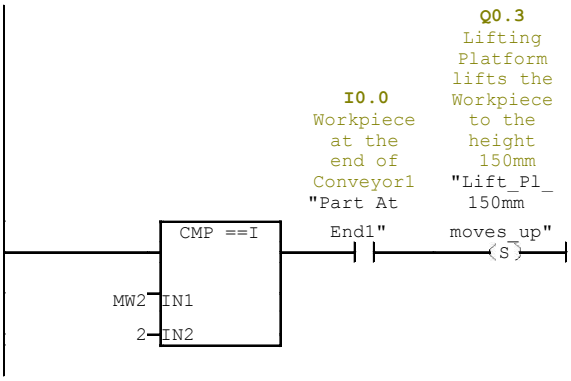
Network: 3 Workpiece is available; Coneveyor 1 starts working



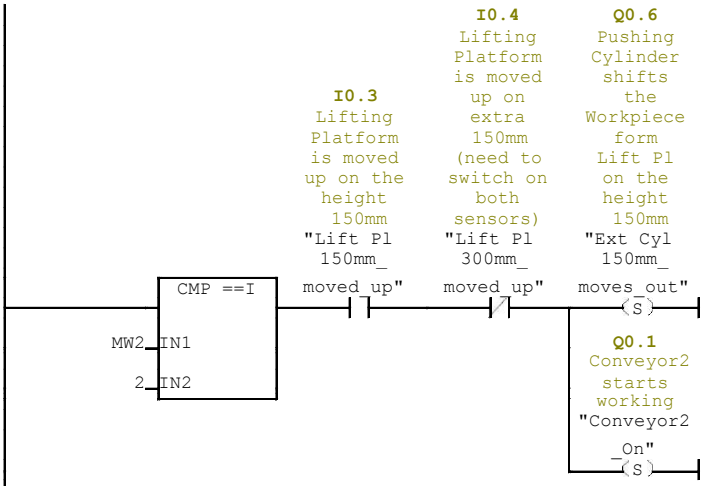
Network: 4 If the Blue colour, then lifts the Workpiece on height 150mm



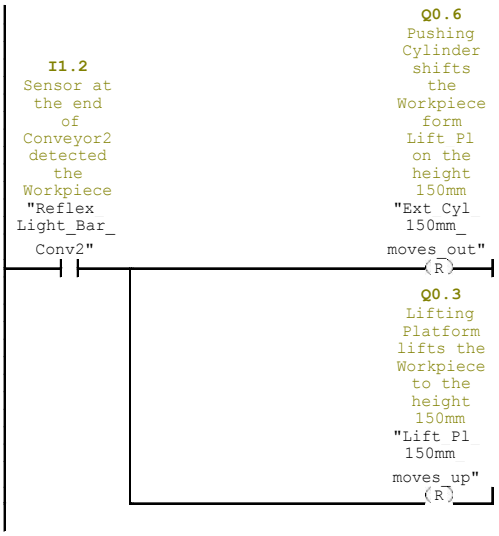
Network: 5 If the Blue colour, then lifts the Workpiece on height 150mm



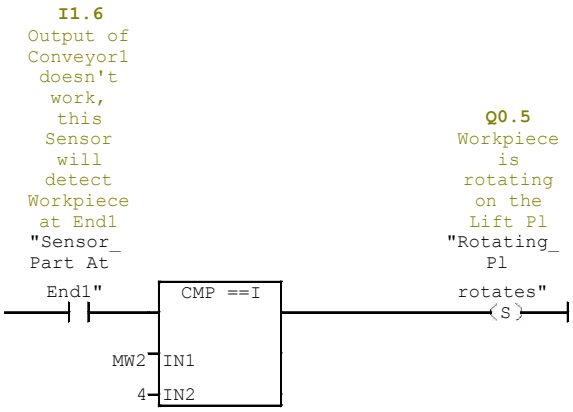
Network: 6 Pushing Cylinder shifts the Workpiece form Lift_Pl on the height



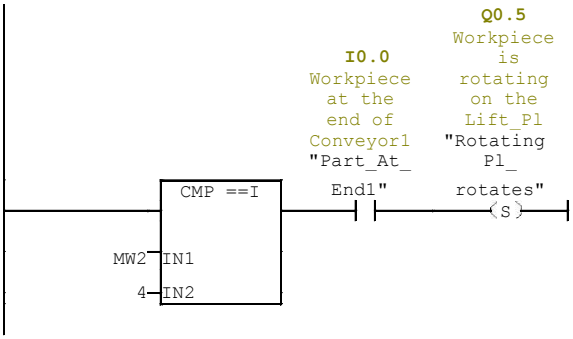
Network: 7 Reset LiftingPl and Pushing Cylinder to the initial position



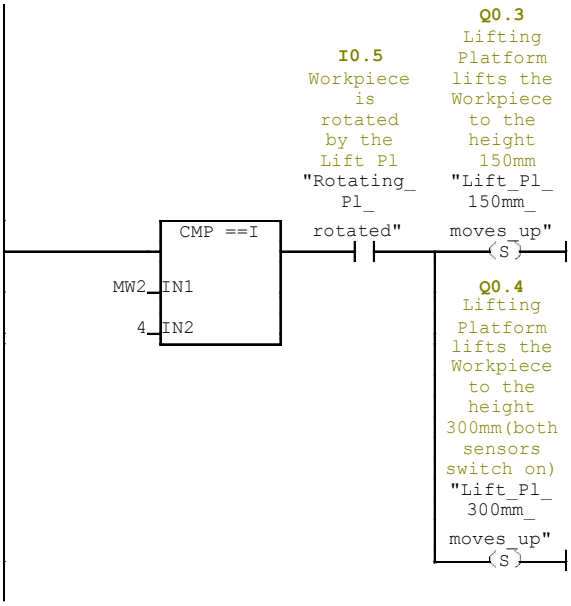
Network: 8 If the Green colour, then rotates the Lifting Platform



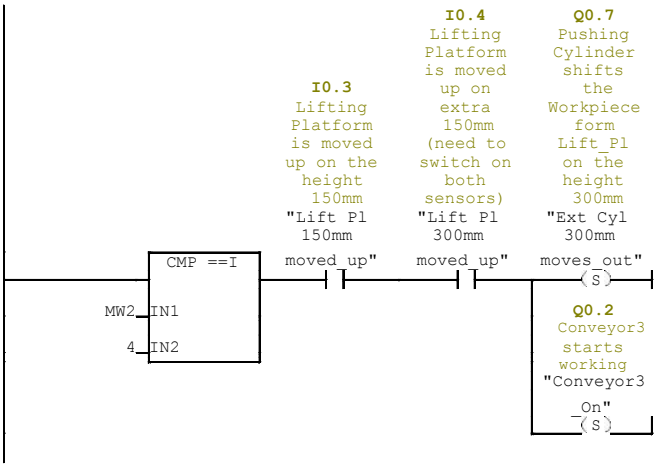
Network: 9 If the Green colour, then rotates the Lifting Platform



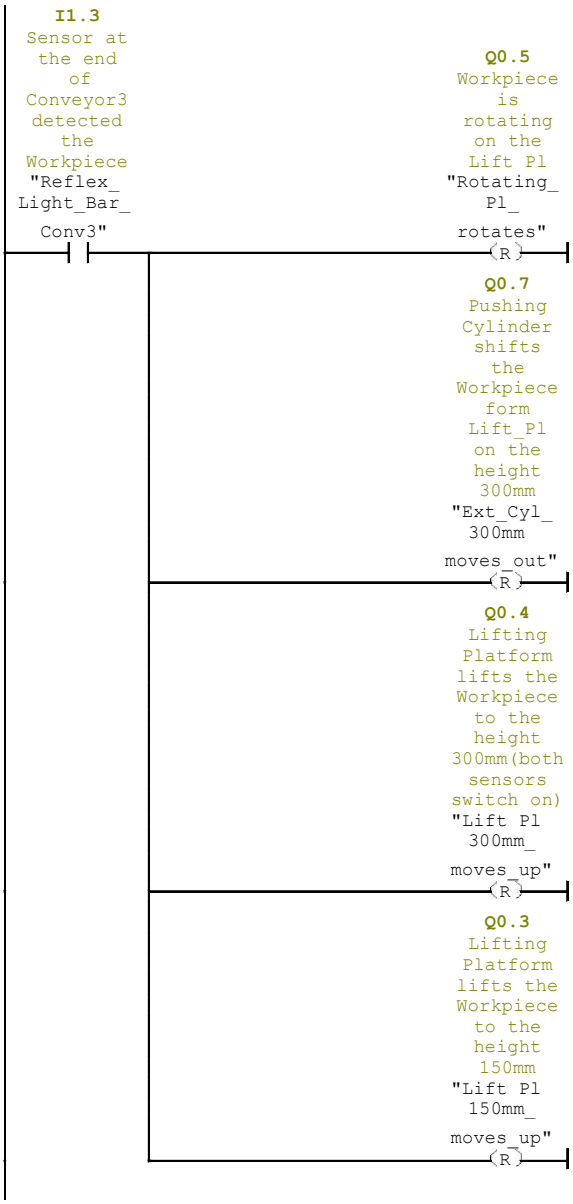
Network: 10 Lifting Platform lifts the Workpiece to the height 300mm



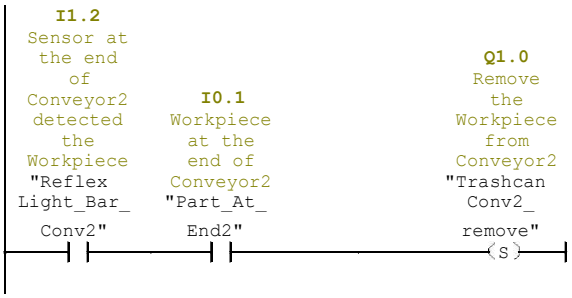
Network: 11 Pushing Cylinder shifts the Workpiece form Lift_Pl on the height



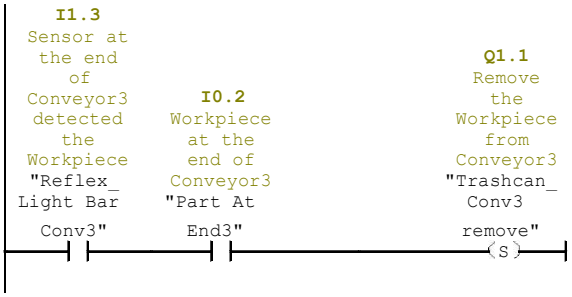
Network: 12 Reset Roting Pl, LiftingPl and PushingCyl to initial position



Network: 13 Remove the Workpiece from Conveyor2



Network: 14 Remove the Workpiece from Conveyor3



I/O – table

Properties of symbol table	
Name:	Symbols
Author:	
Comment:	
Created on	04/09/2015 04:16:32 PM
Last modified on:	04/20/2015 12:08:15 PM
Last filter criterion:	All Symbols
Number of symbols:	28/28
Last Sorting:	Address Ascending

Status	Symbol	Address	Data type	Comment
	Part_At_End1	I 0.0	BOOL	Workpiece at the end of Conveyor1
	Part_At_End2	I 0.1	BOOL	Workpiece at the end of Conveyor2
	Part_At_End3	I 0.2	BOOL	Workpiece at the end of Conveyor3
	Lift_Pl_150mm_moved_up	I 0.3	BOOL	Lifting Platform is moved up on the height 150mm
	Lift_Pl_300mm_moved_up	I 0.4	BOOL	Lifting Platform is moved up on extra 150mm (need to switch on both sensors)
	Rotating_Pl_rotated	I 0.5	BOOL	Workpiece is rotated by the Lift_Pl
	Ext_Cyl_150mm_moved_out	I 0.6	BOOL	Pushing Cylinder has shifted the Workpiece from Lift_Pl on height 150mm
	Ext_Cyl_300mm_moved_out	I 0.7	BOOL	Pushing Cylinder has shifted the Workpiece from Lift_Pl on height 300mm
	Color_Sens_detected_Blue	I 1.0	BOOL	Color Sensor has detected the Workpiece with blue colour
	Color_Sens_detectd_Green	I 1.1	BOOL	Color Sensor has detected the Workpiece with green colour
	Reflex_Light_Bar_Conv2	I 1.2	BOOL	Sensor at the end of Conveyor2 detected the Workpiece
	Reflex_Light_Bar_Conv3	I 1.3	BOOL	Sensor at the end of Conveyor3 detected the Workpiece
	Button_Blue_Workpiece	I 1.4	BOOL	By pressing button, the Replicator will produce Blue Workpiece
	Button_Green_Workpiece	I 1.5	BOOL	By pressing button, the Replicator will produce Blue Workpiece
	Sensor_Part_At_End1	I 1.6	BOOL	Output of Conveyor1 doesn't work, this Sensor will detect Workpiece at End1
	Conveyor1_On	Q 0.0	BOOL	Conveyor1 starts working
	Conveyor2_On	Q 0.1	BOOL	Conveyor2 starts working
	Conveyor3_On	Q 0.2	BOOL	Conveyor3 starts working
	Lift_Pl_150mm_moves_up	Q 0.3	BOOL	Lifting Platform lifts the Workpiece to the height 150mm
	Lift_Pl_300mm_moves_up	Q 0.4	BOOL	Lifting Platform lifts the Workpiece to the height 300mm(both sensors switch on)
	Rotating_Pl_rotates	Q 0.5	BOOL	Workpiece is rotating on the Lift_Pl
	Ext_Cyl_150mm_moves_out	Q 0.6	BOOL	Pushing Cylinder shifts the Workpiece form Lift_Pl on the height 150mm
	Ext_Cyl_300mm_moves_out	Q 0.7	BOOL	Pushing Cylinder shifts the Workpiece form Lift_Pl on the height 300mm
	Trashcan_Conv2_remove	Q 1.0	BOOL	Remove the Workpiece from Conveyor2
	Trashcan_Conv3_remove	Q 1.1	BOOL	Remove the Workpiece from Conveyor3
	Replicator_Template_Blue	Q 1.2	BOOL	Replicator produces the Workpiece with Blue colour
	Replicator_Templat_Green	Q 1.3	BOOL	Replicator produces the Workpiece with Green colour
	T0	T 0	TIMER	Conveyor 1

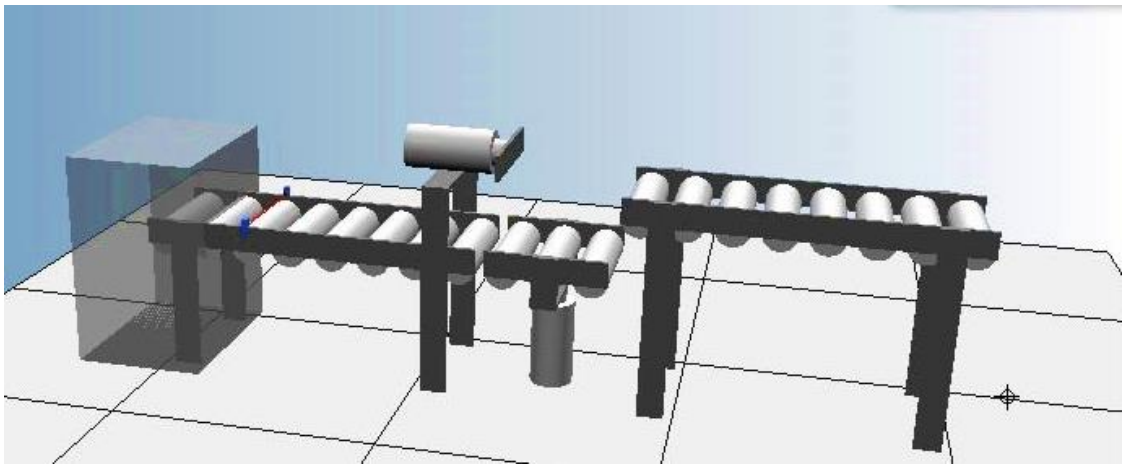
Automation Studio manual

The tutorial familiarizes the user with method of building the workcell by using CIROS Studio. Transporting a workpiece along the conveyors will be considered as an example. It requires basic knowledge about designing the 3D models, controlling system by PLC and ability to write sequences in SIMATIC Manager Step7. The system of the roller conveyors can be used as the preliminary process before robot simulation (or with similar processes stations), which will grab the workpiece form the conveyor belt and continue motion. The 3D model of station was created by the program CIROS Studio, and followed by sequence written in SIMATIC Manager STEP7 program.

The scope includes:

- Description of the main idea of motion
- Designing the objects in CIROS Studio
- IO Controller description
- Information about IOs
- Controlling the system with manual override and with PLC program written

The Conveyor System is shown in the Picture 1.



Picture 1

An idea of the system is transporting the workpiece along the conveyors. The lifting platform, which is established between the conveyors, pushes the workpiece up to the height of conveyor belt 2. The pushing cylinder will shift the

workpiece from the lifting platform to conveyor belt 2. When the workpiece has crossed the second conveyor belt and stopped at the end of that, it needs to be removed. This action can be performed by a robot or by the trashcan mechanism.

Components of the system:

- Two roller conveyor belts

If there is an object above the conveyor belt, then the object is moved along the conveyor's surface. The conveyor is laid aboard the X-axis in the system. It has two input values and one output. The input *Belt_Reverse* is not needed in the system, because there is only one direction of moving the workpiece. The primary input and output values are estimated as 0. To activate the conveyor – set the value to 1. The input values tell about switching the conveyor ON and OFF. The output tells about an availability of an object at the end of the conveyor. The conveyor belt has a gripper point, so when there is an object with free grip point above the conveyor, this object will be moved along the conveyor surface.

Input values of both conveyors are: *Conveyor1_On* and *Conveyor2_On*

Output values are: *PartAtEnd_1* and *PartAtEnd_2*

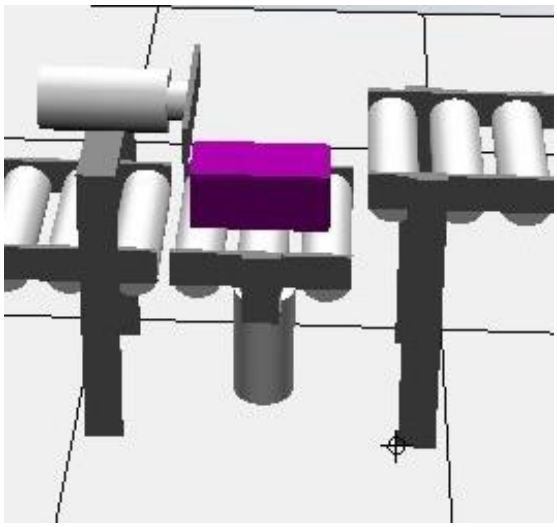
The difference of each input and each output is an index value. The index value is needed for writing the PLC program for the system. They will be assigned to the objects when the IO Controller is inserted into the workcell. When the manual operation mode is used, the index values are not needed.

The roller conveyor is made of cylinders and has a mechanism *conveyor surface*. The conveyor surface of Conveyor 1 has the length along X-axis equaled to 900 mm, and along Y-axis equaled to 250 mm. But the conveyor surface of Conveyor 2 has the length of 550 mm. It means that the workpiece will stop at the position, when it passes 550 mm of the conveyor's length. However, this value can be changed. The speed of cylinders' rotation is 100 mm/s. The cylinders have to be designed in new section *Rollers*. The conveyor surface mechanism also needs to be in the same section as the cylinders. The radius of cylin-

ders is 40 mm, and the length is 250 mm. Place cylinders 100 mm far from each other. Then, move all cylinders and the conveyor surface to the height of 40 mm along the z-axis. Conveyor 2 is higher than Conveyor 1 for 150 mm. Frames are used for better representation as a real model of the conveyors. They fix the cylinders and support the conveyors. To make frames, firstly, you need to create a new section *Frame* in the object. Then, create a box with size of 780x10x60 mm. It will play a role as an edge frame for Conveyor 1. To create other frames, you can copy the first frame and then change the dimensions, direction and locations.

- One lifting platform

The lifting platform (see Picture 2) is made of cylinders with the same diameters and lengths as the conveyor belt has. Also, there is a cylinder which is responsible for moving up and down the lifting platform. It has a mechanism *Cylinder for translation*. The cylinder for translation mechanism has 0 mm value for the lower axis limit and 150 mm – for the upper axis limit. The upper axis limit has the same value as the value of Conveyor 2's height. These limits are responsible for the length of the piston which will be moved out. This cylinder has a base and a piston, and it is established perpendicular to the conveyor part of the lifting platform. The piston has the radius 30 mm and height 230 mm. The base has the radius 50 mm and height 200 mm. Piston has two gripper points: CylinderT_GPPauto and CylinderT_GPPman. Base has a grip point as CylinderT_GP. Gripper grasps an object that has free grip point in the grip range. A grip point, in this case, will be the workpiece. Moving up the workpiece motion is provided by the griper point of the lifting cylinder. The position of the griper point has to be adapted to the height of the lifting platform. CylinderT_GPPauto has to be set to 3400 mm by the Z-axis. Frames are used for the lifting platform. Create them by the same way as for the conveyor belt.



Picture 2

- One Pushing Cylinder

The pushing cylinder (see Picture2) has a piston and a base. The piston is connected to the frame, which will contact the side of workpiece and then move it from the lifting platform. The frame has the dimensions 60x290x10 mm. It has to be created in the section *Piston* of the pushing cylinder object and has to be connected to the end of the piston. The pushing cylinder has a mechanism *Cylinder for translation*. The lower axis limit is 0 mm, and the upper – 150 mm. The radius of piston's cylinder is 30 mm and height 230 mm. The base cylinder has the radius 50 mm and height 200 mm. The base has a grip point CylinderT_GP_1. The piston contents two gripper points: CylindeT_GPPauto_1 and CylinderT_GPman_1. CylindeT_GPPauto_1 has to be adapted to the position 50:0:350 and with pitch -90 degrees. Now, the cylinder will detect the workpiece.

- Workpiece

The workpiece can have different design and dimensions. It is visualized as a *Box* in the system. Box was assigned with purple colour for better representation. Firstly, create a new *Object* in the workcell and put the new *Section*. Use the *Geometric primitives* to create a box. To change the dimension and colour of the workpiece, go to the properties of the object. The box has the dimensions: 200x100x100 mm in the system. Then, place the workpiece at the start of the conveyor belt. The workpiece has to have the grip point, for been grasped

by another object, for example by conveyor belt. Establish the grip point at the position 100:50:0.

- Replicator

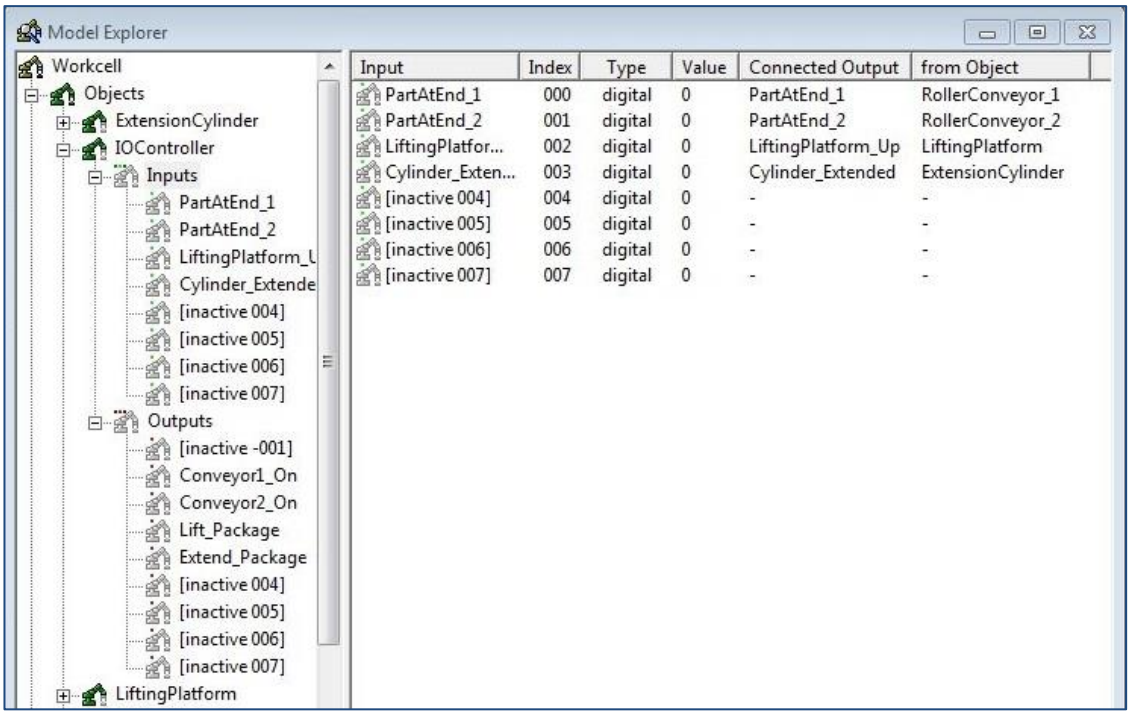
The replicator is responsible for producing the workpieces. Initially it has no visualization, so for better representation and not appearing the workpiece from nowhere, it is better to design a box and assign the material. The material is *Smoky glass*. The replicator is established at the start of the first conveyor. The workpiece will be produced and put at the gripper point of the replicator. The replicator has a mechanism *Replicator socket*. An object *Box* has to be moved to the object *Replicator*. It will play a role as a pattern of the object which has to be produced.

- Reflexive Light Barrier sensor

This sensor will detect a workpiece and send information to Conveyor 1 that it has to be switched on. When the sensor detects the workpiece the input value of the sensor is set to 1.

- IO Controller

IO Controller with 8 I/O's is used for controlling system. The system is controlled by SPS-S7-Simulator and uses sequences written in SIMATIC Manager Step7. The decision about a type of IO Controller is made according to the number of inputs and outputs in the system. Picture 3 presents the index values of IOs and the relations between inputs and outputs of some objects in the IO Controller.



Picture 3

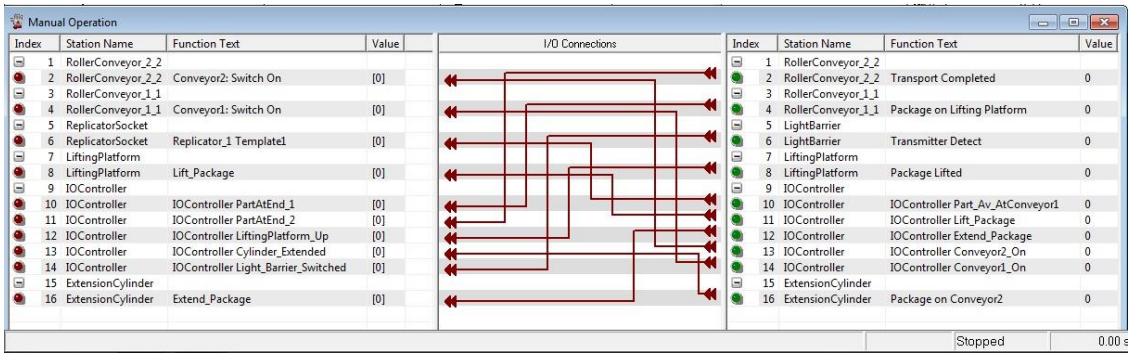
The system is controlled by STEP 7, so to load the program into the workcell, follow:

Programming – S7 Program Manager – IO Controller – Load Program

As well, the simulation can be manually operated:

Modeling – Manual Operation

Relations between IO Controller and IOs of objects are presented in Picture 5.

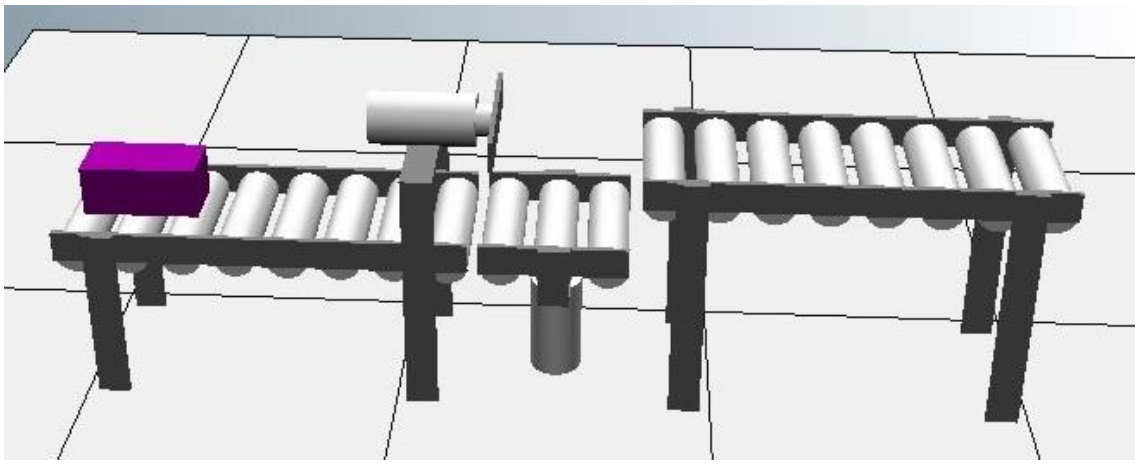


Picture 4

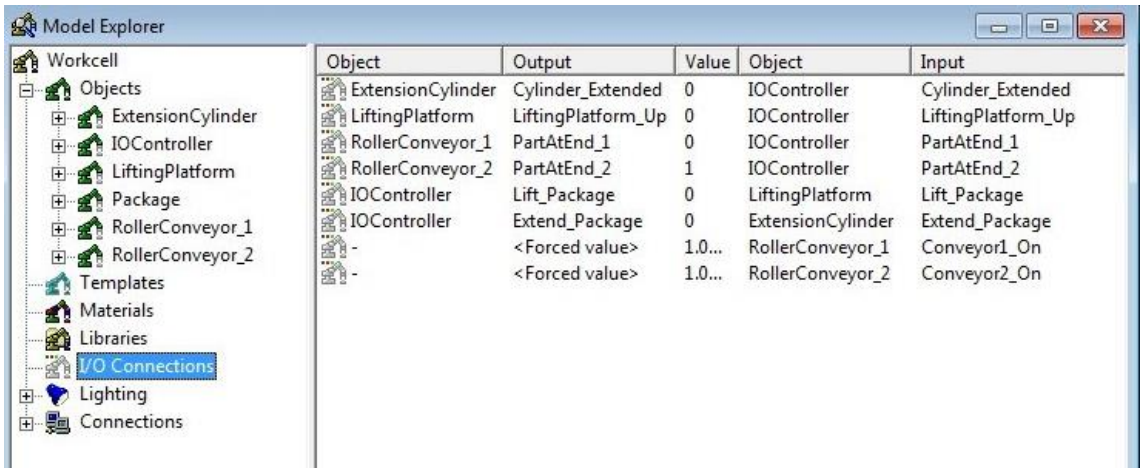
Algorithm of the motions:

Firstly, the workpiece has to pass Conveyor 1 and at the end of the conveyor it will be detected. Then, the workpiece goes to Lifting Platform, which will push it up. The next stage is leaded by Pushing Cylinder, which will shift the workpiece to Conveyor 2. After, the workpiece is moving along Conveyor 2. Finally, the workpiece is detected at the end of Conveyor 2. Now, it can be removed by Trashcan mechanism, the subsequent station which includes the robot or other constituents. As soon as the PLC program has been downloaded into the work-cell, the simulation can be started.

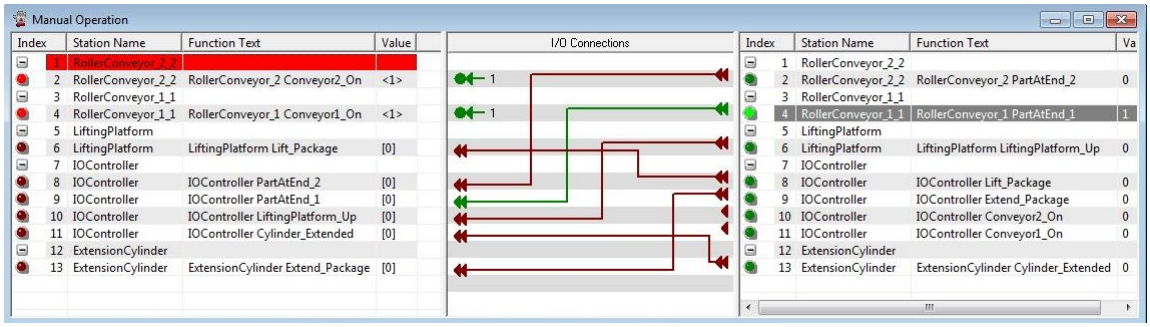
The old version of the system had not Replicator and Reflexive Light Barrier sensor (see Picture 5). And it had two forced values for the conveyors. An abil-ity to switch the conveyor on automatically appeared when the sensor was es-tablished, because it plays a role of pre-action. The relations between IOs of object and IO Controller can be seen in Picture 6 and Picture 7.



Picture 5



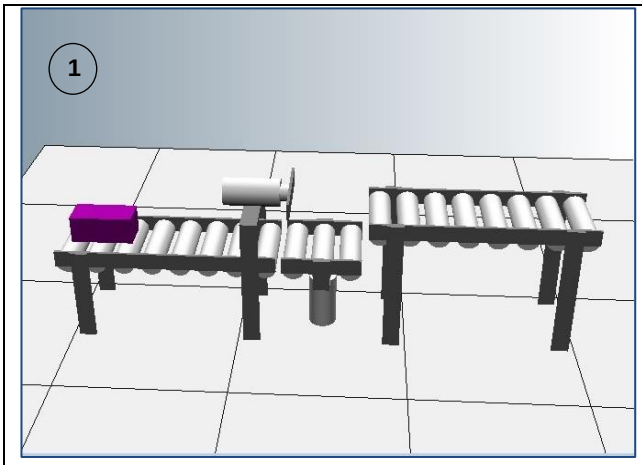
Picture 6

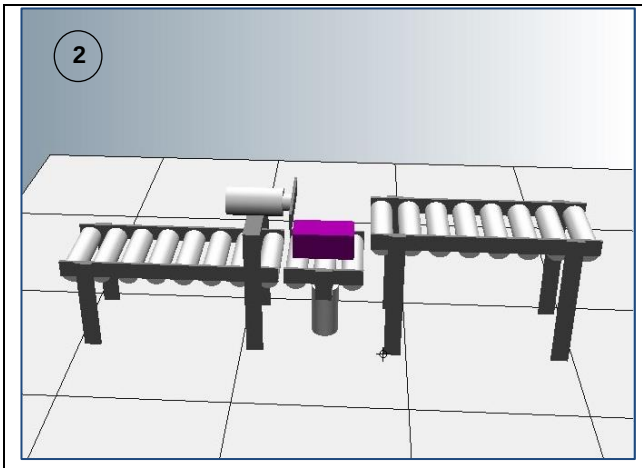
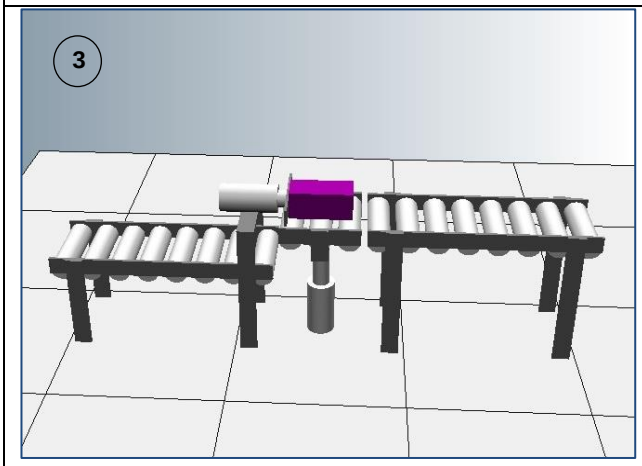
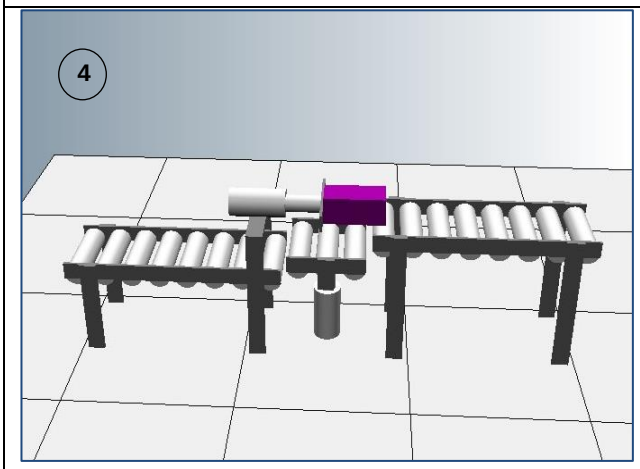


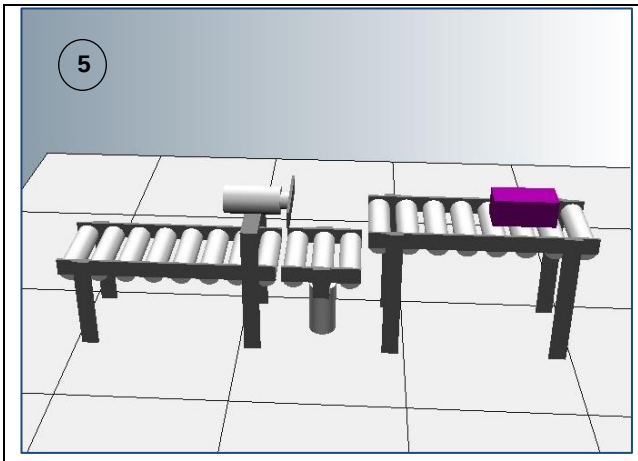
Picture 7

The old version of the sequences of motions in the system is presented in Table 1.

Table 1

	Output Conveyor1_On = 1

	Input PartAtEnd_1= 1 Output Lift_Package = 1
	Input LiftingPlatform_Up = 1 Output Extend_Package = 1
	Input Cylinder_Extended= 1 Output Conveyor2_On = 1

	Input PartAtEnd_2= 1 Output Conveyor2_On = 0 Output Conveyor1_On = 0 Output Lift_Package = 0 Ouput Extend_Package = 0