DANGER

All connections are NOT hot plug capable. Please turn off power before removing or plugging in ANY plug!!!

BPU-FCXU

Documentation A.1

IB Ostendorff

Table of content

List of figures

List of tables

Abbreviations

BBU .... Base Board Unit

BPU .... Bus Protection Unit

BPU-FCXU .... Bus Protection Unit with FPGA/CPLD Control eXpansion Unit

CPLD .... Complex Programmable Logic Device

FCXU .... FPGA Control eXpansion Unit

FPGA .... Field Programmable Gate Array

GOLDi .... Grid of Online Lab Devices Ilmenau

JTAG .... Joint Test Action Group

Explanations

Symbols mark especially important information.

DANGER

Please read these sections with extreme attention, to avoid any danger for human beings and the machine.

ATTENTION

Read these sections carefully to avoid problems while using the device.

INFORMATION

Read this section for additional information and hints.

1. Overview

This documentation describes the BPU-FCXU.

BPU-FCXU stands for “Bus Protection Unit with FPGA/CPLD Control eXpansion”. This is a FPGA/CPLD-based control unit for the *Grid of Online Lab Devices Ilmenau (GOLDi)*. It consists of the following parts:

- Base Board Unit (BBU) with a CAN Bus Interface (CBI),
- as well as the FPGA/CPLD Control eXpansion Unit (FCXU).

All boards are designed for academic and research use as part of the *GOLDi* infrastructure and NOT for any industrial usage.

For feedback, as well as ideas and comments please send an email to goldi@ib-ostendorff.de.

1.1. How to read

This document is for users as well as administrators working with the BPU-FCXU. Therefore not all information is equally relevant for all users.

While users should have a short look at section 1 they should especially read section 3 and section 5. The others section might be skipped.

Administrators should be familiar with the complete document.

1.2. BBU

The BBU is used to interface the FCXU to the *GOLDi* infrastructure. It connects the plugged in eXpansion boards, FCXU in this case, via CAN to *GOLDi*. The FCXU has to be plugged in the top eXpansion port of the BBU. See Figure 4 for details.

Figure 1: BBU v1_11 (with CBI fitted and empty eXpansion slot)

1.3. FCXU

The FCXU is designed to act as a control unit in *GOLDi* infrastructure. It has the standard 85x30mm format to plug onto a BBU.

The FCXU Board version 1.00 contains a CPLD from Altera. The type is MAX V (5M570ZT100C5N). It runs on either its internal frequency generator or an external frequency supplied by the BBU. The I/O voltage is 3,3V.

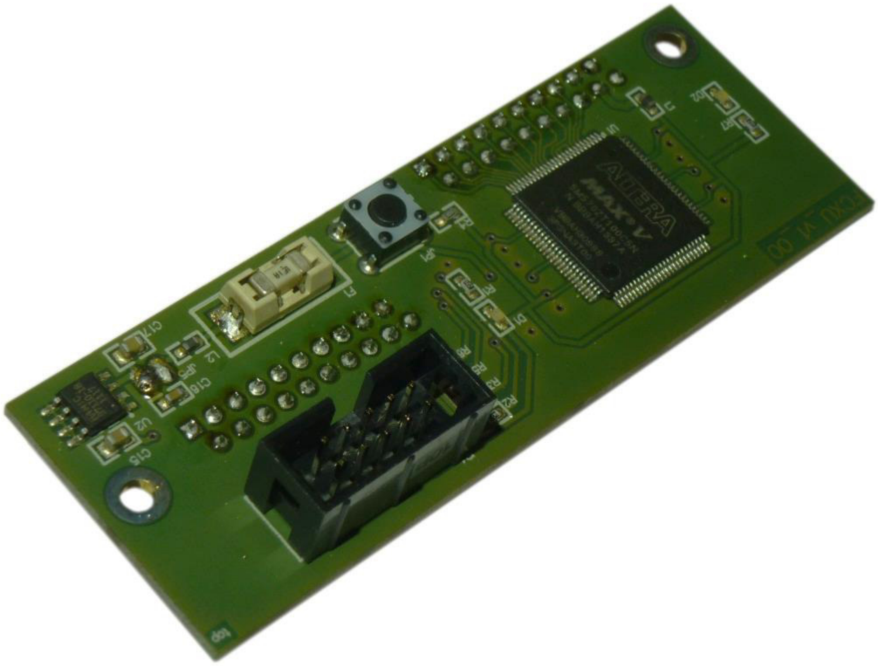

Figure 2: FCXU v1_00 (top side)

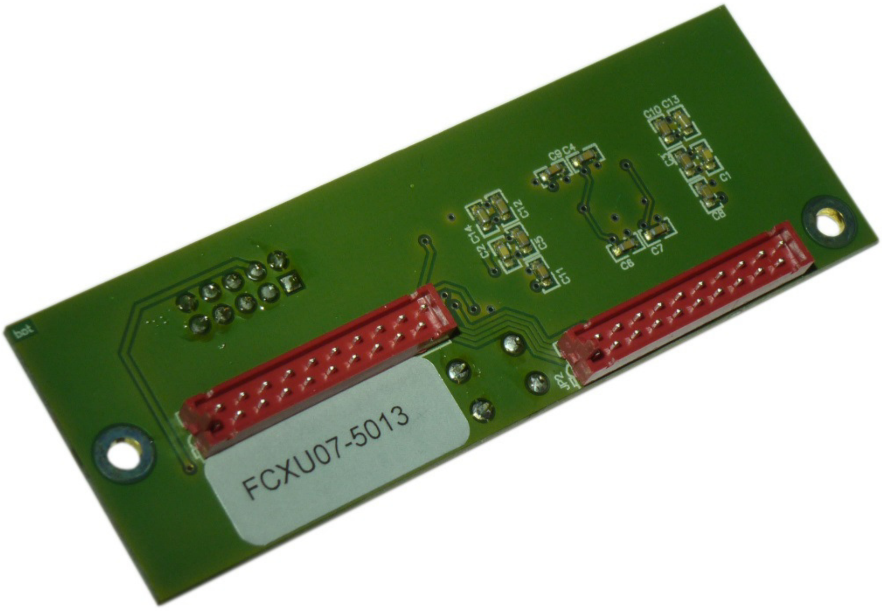

Figure 3: FCXU v1_00 (bottom side)

1.4. BPU-FCXU

The combination of BBU and FCXU together with the right firmware creates the BPU-FCXU.

Figure 4: BPU-FCXU assembly (with CBI and FCXU fitted)

2. Operation

The functions described in this section only relate to the combination of FCXU and BBU with the newest firmware for FCXU-BPU.

2.1. Operation of BBU

The BBU supplies the following functions relevant for the user.

2.1.1. LEDs

The four yellow LEDs on the front side of the BBU have the following functions:

LED	mode	function
1	off	no connection to *GOLDi*
	flashing	firmware running
2	off	MCXU is powered off
	on	MCXU is powered on
3		*GOLDi* bus data traffic
4		*GOLDi* bus control message traffic

Table 1: Function of BBU LEDs

Figure 5: BBU LEDs

The LEDs 1..4 are numbered from left to right.

2.1.2. DIP

The four DIP switches on the front side of the BBU have the following functions:

DIP	mode	
1..4	0000	normal running mode
1..4	1111	firmware update mode
1..4	else	no function defined yet

Table 2: Function of BBU DIP switches

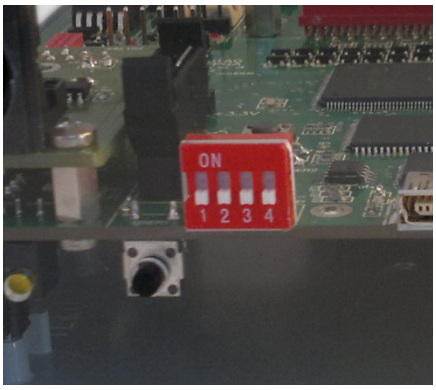

Figure 6: BBU DIP and user button

2.1.3. User Button

The user button currently has no functionality. For the location see Figure 6.

2.1.4. Reset

The reset button resets the complete BPU-FCXU. Press at least 1 second to reset device. It takes at least 3 seconds after releasing the button, until BPU-FCXU has been completely reset and is in a safe state again and LED 1 starts flashing.

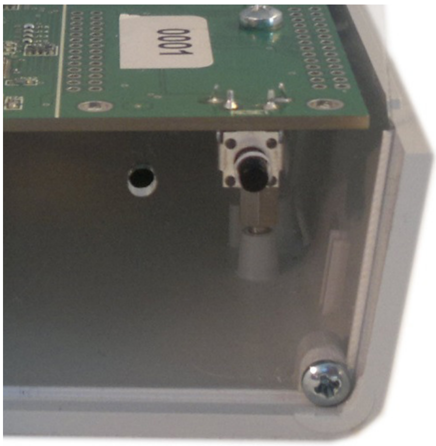

Figure 7: Reset button and power LED (not fitted on this board)

2.2. Operation of FCXU

A UART is used for the communication with the BBU. With this interface all sensor and actuator values will be transmitted to the physical system model. Details about the format can be found in the template.

It is recommended to use the templates for the communication with the BBU.

Make sure you know the right clock frequency supplied by the BBU when creating any design.

The FCXU relies on the correct firmware running on the BBU. This firmware takes care of programming the FPGA/CPLD as well as the data handling and the communication with the *GOLDi* infrastructure and the services running on the *GOLDi* server.

Please make sure to use the correct firmware for the BBU when using the FCXU. The newest firmware can be found at www.ib-ostendorff.de/GOLDI_firmware or have a look at www.tu-ilmenau.de/GOLDI.

3. Template

There are templates for the FCXU Board available that contain all important information to use the board for control tasks within the *GOLDi* infrastructure. The user can concentrate on his control task while leaving the handling of the interface to the functions supplied with the template.

Changing code within the interface functions of the template will not damage any hardware or disrupt the functionality of the *GOLDi* infrastructure, but may result in a non-proper working user design.

It is NOT recommended to change any code in the interface functions.

The user is free to program his control algorithm in any way he wants. This can be as automata (like in the template), as some other VDHL, Verilog, SystemC code or any other language. There are no restrictions to the user.

In most cases edit the <PhyiscalSystemName.vhd> (e.g. Elevator_3Floors.vhd) in the ./Source/ directory. Changes to other files, especially in the directory ./Source/Modules/ are *NOT* necessary and may prevent your design from working correctly.

The template project has been created with Quartus II.

3.1. Directory structure

```
.
├──Binary                                    Directory for the control task.
│  └──<PhysicalSystemName.pof>                  Contains the programming file after synthesis.
│
├──Source                                    Directory with source files.
│  └──Modules                                Directory with predefined modules. Do not change.
│      └──ControlUnit_Adapter_v1_10.vhd         Control adapter unit.
│      └──UART_Transceiver_RX_v1_20.vhd         UART file for receiving data from the BBU.
│      └──UART_Transceiver_TX_v1_20.vhd         UART file for transmitting data to the BBU.
│
│  └──<PhysicalSystemName.qpf>               Project file
│  └──<PhysicalSystemName.qsf>               Project file with constraints.
│  └──<PhysicalSystemName.vhd>               Source file for the control task.
```

Figure 8: Template structure

4. Connections

There are different interfaces available at the FCXU. They are either ‘real’ and connect the FCXU to the BBU, or they are ‘virtual’ and describe how the FCXU can communicate with electro mechanical physical systems connected to the *GOLDi* infrastructure. The ‘virtual’ connections are described in section 5.

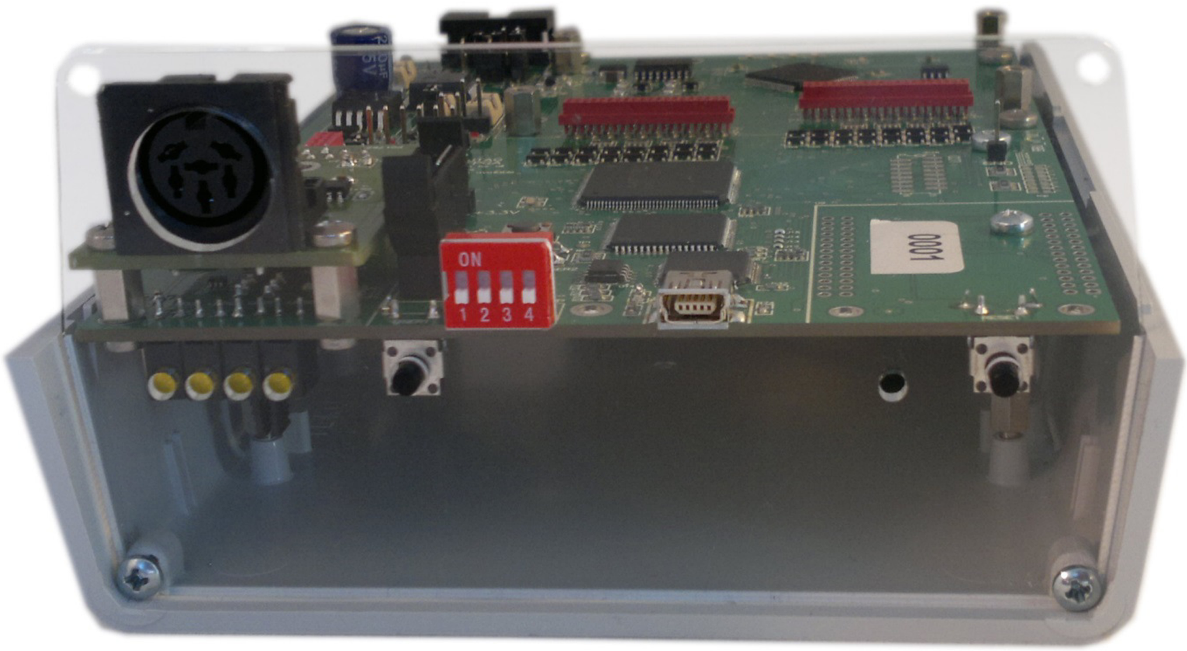

Figure 9: BPU-FCXU front view (with empty eXpansion slot)

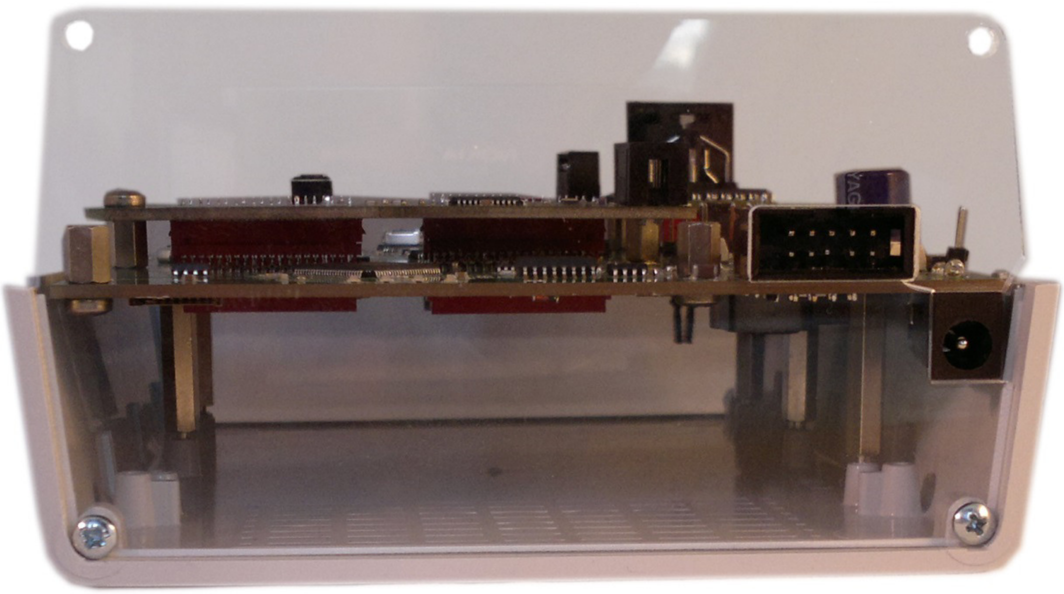

Figure 10: BPU-FCXU back view (with fitted MCXU on top eXpansion slot)

4.1. *GOLDi* bus connector

The *GOLDi* connector is a 6-pin DIN connector that interfaces the BPU-MCXU to the *GOLDi* infrastructure.

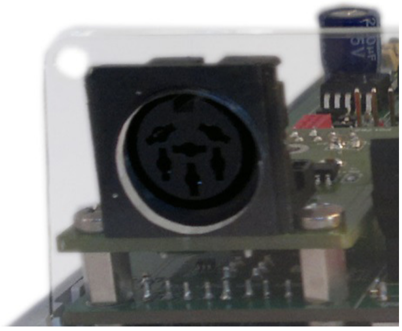

Figure 11: *GOLDi* connector

4.2. Secondary power connector

During regular operation within the *GOLDi* infrastructure the power is supplied via the *GOLDi* connector. In some cases it might be useful to supply the BBU with its own power supply. Use a power supply with 24V DC with a minimum of 1 A. See Figure 13 for correct adapter polarity.

Figure 12: 2.1mm power connector

Figure 13: Adapter polarity

4.3. USB

The BBU has a mini USB connector for maintenance (e.g. firmware updates) and stand-alone use. A FTDI 2232HL is used to enable the reprogramming of the BBU.

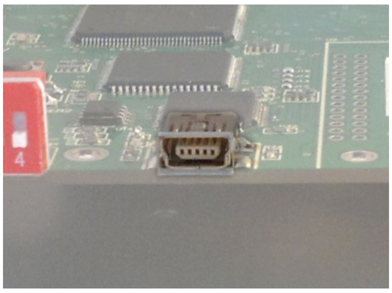

Figure 14: Mini USB connector

Only use the BBU programming tool supplied for any reprogramming to avoid damage to the hardware.

4.4. BBU 10-pin connector

On a BPU-MCXU the 10-pin connector on the back side of the BBU has no functionality.

Figure 15: 10-pin connector on BBU

4.5. FCXU JP1

Reset button. This is for development only.

4.6. FCXU JP2

10 pin connector for JTAG. This is for development only.

4.7. BBU to FCXU internal interface

The internal interface connection to the BBU is as follows.

4.7.1. FCXU JP3

Connection	Pin number	Pin number	Connection
TDO	1	2	TCK
TDI	3	4	TMS
GPIO_00	5	6	GPIO_01
GPIO_02	7	8	GPIO_03
GPIO_04	9	10	GPIO_05
GPIO_06	11	12	GPIO_07
GPIO_08	13	14	GPIO_09
GPIO_10	15	16	GPIO_11
GND	17	18	+3,3V
GND	19	20	Not connected

Table 3: FCXU JP3

4.7.2. FCXU JP4

Connection	Pin number	Pin number	Connection
Not connected	1	2	Not connected
Not connected	3	4	Not connected
Not connected	5	6	Not connected
Not connected	7	8	Not connected
Not connected	9	10	Not connected
Not connected	11	12	Not connected
Not connected	13	14	Not connected
Not connected	15	16	Not connected
GND	17	18	Not connected
GND	19	20	Not connected

Table 4: FCXU JP4

4.7.3. CPLD connections

The GPIOs and other important signals are connected to the CPLD at the following pins.

Connection	Pin number
GPIO_00	21
GPIO_01	20
GPIO_02	19
GPIO_03	18
GPIO_04	17
GPIO_05	16
GPIO_06	15
GPIO_07	14
GPIO_08	12
GPIO_09	7
GPIO_10	6
GPIO_11	5

Table 5: CPLD connections

5. Electro mechanical physical system connection

The virtual external interface connects the microcontroller to different electro mechanical physical systems within the *GOLDi* infrastructure. The actual connection depends on the models and the firmware used.

If you use a different electro mechanical physical system, please contact goldi@ib-ostendorff.de for a template and documentation of this model.

5.1. Elevator (type A – 3 floors)

There are different elevator models available. This one describes the larger model with 3 floors.

The elevator consists of a cage with counterweights, a pit and three floor units, each one containing a pneumatic driven sliding door, call buttons and colored control lamps to indicate the moving direction of the cage. In addition to this there is a control panel, realizing the operating options from inside the cage. In essence, these are selection buttons to choose a floor, an alarm button, an emergency stop and the ability to choose a mode of operation, where the lift is controlled exclusively from outside the cage. The simulated process shows the elevator being brought from a basic position to one of the floors, by operating the control panel or one of the call buttons, and after opening and closing the sliding door being ready for the next sequence: After operation one of the call buttons, indicated by a signal lamp, the cage is brought in a slow-fast-slow-movement, being controlled by mechanical switches depending on the distance, to the chosen floor. The sliding door gets opened and remains open, until the programmed loading time is over. A one way light barrier controls the entrance to prevent, in a real case persons or things that are in the danger zone of the door, from getting hurt. After closing the sliding door, the cage gets moved to the next chosen floor, where the sequence of opening and closing the sliding door occurs in the same manner. A miniature compressor for the pneumatic driven sliding doors is integrated in the model.

Figure 16: 3-level elevator electro mechanical physical system

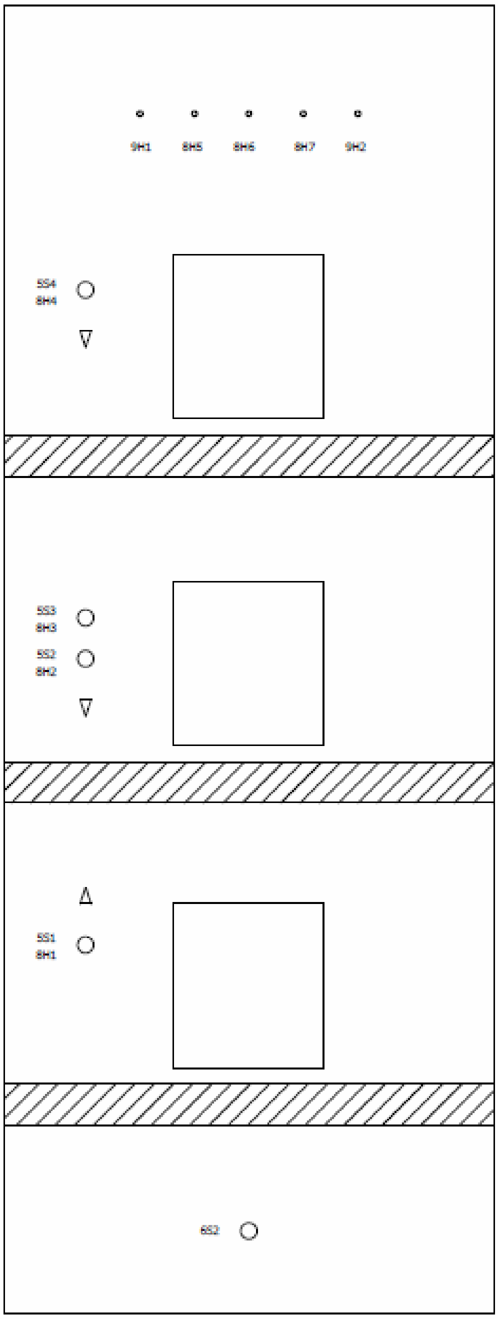

Figure 17: 3-level elevator electro mechanical physical system schematic

Var	Name	Function
x0	not defined yet	cabin pos. ground floor
x1	not defined yet	cabin pos. 1. floor
x2	not defined yet	cabin pos. 1. floor
x3	not defined yet	switchover slow ground floor
x4	not defined yet	switchover slow 1. floor from the bottom
x5	not defined yet	switchover slow 1. floor from the top
x6	not defined yet	switchover slow 2. floor
x7	not defined yet	door ground floor open
x8	not defined yet	door ground floor closed
x9	not defined yet	door 1. floor open
x10	not defined yet	door 1. floor closed
x11	not defined yet	door 2. floor open
x12	not defined yet	door 2. floor closed
x13	not defined yet	light barrier ground floor
x14	not defined yet	light barrier 1. floor
x15	not defined yet	light barrier 2. floor
x16	not defined yet	call button ground floor
x17	not defined yet	call button 1. floor downward
x18	not defined yet	call button 1. floor upstairs
x19	not defined yet	call button 2. floor
x20	not defined yet	call button ground floor (operator panel)
x21	not defined yet	call button 1. floor (operator panel)
x22	not defined yet	call button 2. floor (operator panel)
x23	not defined yet	alert (operator panel)
x24	not defined yet	emergency stop (operator panel)
x25	not defined yet	Simulation overload

Table 6: 3-level elevator sensors

Var	Name	Function
y0	not defined yet	cabin to Z+
y1	not defined yet	cabin to Z-
y2	not defined yet	cabin slow
y3	not defined yet	to open door ground floor
y4	not defined yet	to close door ground floor
y5	not defined yet	to open door 1. floor
y6	not defined yet	to close door 1. floor
y7	not defined yet	to open door 2. floor
y8	not defined yet	to close door 2. floor
y9	not defined yet	call display ground floor
y10	not defined yet	call display 1. floor upstairs
y11	not defined yet	call display 1. floor downward
y12	not defined yet	call display 2. floor
y13	not defined yet	indicator display ground floor
y14	not defined yet	indicator display 1. floor
y15	not defined yet	indicator display 2. floor
y16	not defined yet	drive direction display downward
y17	not defined yet	drive direction display upstairs
y18	not defined yet	call display ground floor LED (operator panel)
y19	not defined yet	call display 1. floor LED (operator panel)
y20	not defined yet	call display 2. floor LED (operator panel)
y21	not defined yet	alert LED (operator panel)
y22	not defined yet	emergency stop LED (operator panel)
y23	not defined yet	overload LED (operator panel)

Table 7: 3-level elevator actuators

5.2. Elevator (4 level)

There are different elevator models available. This one describes the larger model with 4 floors.

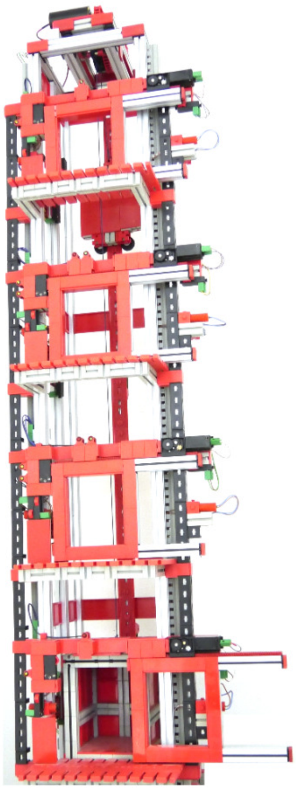

Figure 18: 4-level elevator electro mechanical physical system

Var	Name	Function
x0	elevator_sens_call1_up	call elevator to go up on 1^{st} floor
x1	elevator_sens_call2_down	call elevator to go down on 2^{nd} floor
x2	elevator_sens_call2_up	call elevator to go up on 2^{nd} floor
x3	elevator_sens_call3_down	call elevator to go down on 3^{rd} floor
x4	elevator_sens_call3_up	call elevator to go up on 3^{rd} floor
x5	elevator_sens_call4_down	call elevator to go down on 2^{nd} floor
x6	elevator_sens_cab_F1	cabin is on 1^{st} floor
x7	elevator_sens_cab_F2	cabin is on 2^{nd} floor
x8	elevator_sens_cab_F3	cabin is on 3^{rd} floor
x9	elevator_sens_cab_F4	cabin is on 4^{th} floor
x10	elevator_sens_cab_above_F1	cabin is close above 1^{st} floor
x11	elevator_sens_cab_above_F2	cabin is close above 2^{nd} floor
x12	elevator_sens_cab_above_F3	cabin is close above 3^{rd} floor
x13	elevator_sens_cab_below_F2	cabin is close below 1^{st} floor
x14	elevator_sens_cab_below_F3	cabin is close below 2^{nd} floor
x15	elevator_sens_cab_below_F4	cabin is close below 3^{rd} floor
x16	elevator_sens_cab_F1	1^{st} floor selected in cabin
x17	elevator_sens_cab_F2	2^{nd} floor selected in cabin
x18	elevator_sens_cab_F3	3^{rd} floor selected in cabin
x19	elevator_sens_cab_F4	4^{th} floor selected in cabin
x20	elevator_sens_cab_alarm	alarm triggered
x21	elevator_sens_cab_emerg	emergency triggered
x22	elevator_sens_door_F1_free	door on 1^{st} floor is not blocked
x23	elevator_sens_door_F1_closed	door on 1^{st} floor is closed
x24	elevator_sens_door_F1_opened	door on 1^{st} floor is opened
x25	elevator_sens_door_F2_free	door on 2^{nd} floor is not blocked
x26	elevator_sens_door_F2_closed	door on 2^{nd} floor is closed
x27	elevator_sens_door_F2_opened	door on 2^{nd} floor is opened
x28	elevator_sens_door_F3_free	door on 3^{rd} floor is not blocked
x29	elevator_sens_door_F3_closed	door on 3^{rd} floor is closed
x30	elevator_sens_door_F3_opened	door on 3^{rd} floor is opened
x31	elevator_sens_door_F4_free	door on 4^{th} floor is not blocked
x32	elevator_sens_door_F4_closed	door on 4^{th} floor is closed
x33	elevator_sens_door_F4_opened	door on 4^{th} floor is opened
x34	elevator_sens_overload	overload detected

Table 8: 4-level elevator sensors

Var	Name	Function
y0	elevator_act_cab_down	move cabin downwards
y1	elevator_act_cab_up	move cabin upwards
y2	elevator_act_cab_slow	move cabin slowly
y3	elevator_act_LED1_up	indicate request for cabin going up on 1st floor
y4	elevator_act_LED1_red	red indicator showing cabin is coming down to 1st floor
y5	elevator_act_LED2_down	indicate request for cabin going down on 2nd floor
y6	elevator_act_LED2_up	indicate request for cabin going up on 3rd floor
y7	elevator_act_LED2_green	green indicator showing cabin is coming up to 2nd floor
y8	elevator_act_LED2_red	red indicator showing cabin is coming down to 2nd floor
y9	elevator_act_LED3_down	indicate request for cabin going down on 3rd floor
y10	elevator_act_LED3_up	indicate request for cabin going up on 3rd floor
y11	elevator_act_LED3_green	green indicator showing cabin is coming up to 3rd floor
y12	elevator_act_LED3_red	red indicator showing cabin is coming down to 3rd floor
y13	elevator_act_LED4_down	indicate request for cabin going down on 4th floor
y14	elevator_act_LED4_green	green indicator showing cabin is coming up to 4th floor
y15	elevator_act_cab_LED_F1	indicate cabin is on 1st floor
y16	elevator_act_cab_LED_F2	indicate cabin is on 2nd floor
y17	elevator_act_cab_LED_F3	indicate cabin is on 3rd floor
y18	elevator_act_cab_LED_F4	indicate cabin is on 4th floor
y19	elevator_act_cab_LED_emerg	indicate emergency button pushed
y20	elevator_act_cab_LED_overlo	indicate overload situation of cabin
y21	elevator_act_door1_open	open door on 1st floor
y22	elevator_act_door1_close	close door on 1st floor
y23	elevator_act_door2_open	open door on 2nd floor
y24	elevator_act_door2_close	close door on 2nd floor
y25	elevator_act_door3_open	open door on 3rd floor
y26	elevator_act_door3_close	close door on 3rd floor
y27	elevator_act_door4_open	open door on 4th floor
y28	elevator_act_door4_close	close door on 4th floor

Table 9: 4-level elevator actuators

5.3. Production Cell

The model process cell simulates a process cell with an integrated circular store, as used for example in industrial metal cutting processes. The process cell consists of a rail leaded carriage with a conveyor belt, two slewing tables with conveyor chains, a vertical milling machine and three conveyor belts. The different means of transport are arranged in a closed circuit. Mechanical switches control the end positions of the moveable conveying units and inductive proximity switches check positions of parts. The simulated process shows a part being brought into the circular store, then being clockwise transported to the vertical milling machine, where it gets machined, and finally being brought out to a discharge station. The unmachined part is laid onto the conveyor belt and is brought to the rail leaded carriage. The carriage brings it to the next conveyor belt. There the part is transported to the slewing table, which executes a 90°-turn (+C-direction) to bring the part to the next conveyor belt. This conveyor belt transports it to the milling machine. The upright housing executes a motion along its –Y-direction to draw near the part. The spindle starts rotating and gets moved along its –Z-direction to intimate a machining sequence. After finishing this sequence, the upright housing and the spindle are brought back to their starting position and the machined part is conveyed to the second slewing table that brings it back again onto the first conveyor belt, ready to be withdrawn.

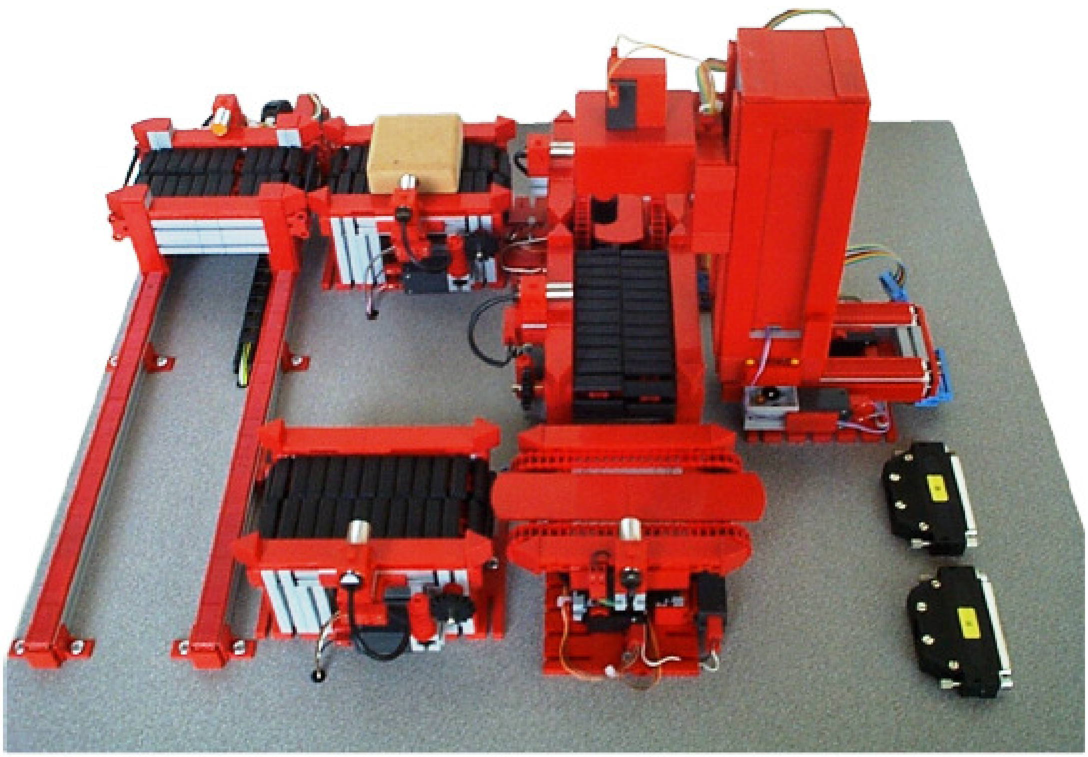

Figure 19: Production cell electro mechanical physical system

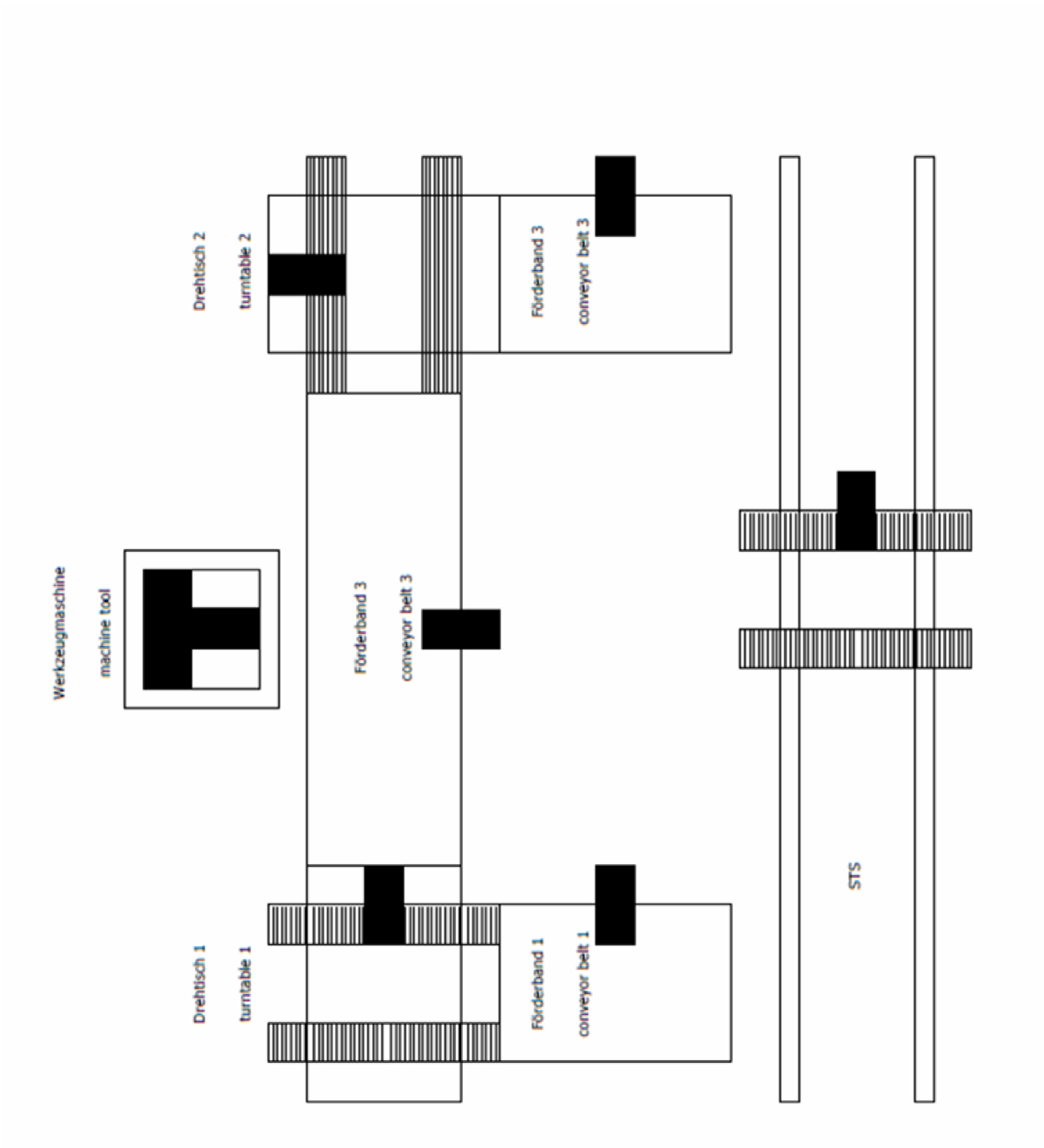

Figure 20: Production cell electro mechanical physical system schematic

Var	Name	Function
x0	not yet defined	Transport_table_in_line_with_conveyor_belt_3_LOW_ACTIVE
x1	not yet defined	Transport_table_in_line_with_conveyor_belt_1_LOW_ACTIVE
x2	not yet defined	Transport_table_workpiece_available
x3	not yet defined	Conveyor_belt_1_workpiece_available
x4	not yet defined	Turntable_1_in_line_with_conveyor_belt_1_LOW_ACTIVE
x5	not yet defined	Turntable_1_in_line_with_conveyor_belt_2_LOW_ACTIVE
x6	not yet defined	Turntable_1_workpiece_available
x7	not yet defined	Conveyor_belt_2_workpiece_available
x8	not yet defined	Turntable_2_in_line_with_conveyor_belt_2_LOW_ACTIVE
x9	not yet defined	Turntable_2_in_line_with_conveyor_belt_3_LOW_ACTIVE
x10	not yet defined	Turntable_2_workpiece_available
x11	not yet defined	Conveyor_belt_3_workpiece_available
x12	not yet defined	Milling_machine_away_from_conveyor_belt_2_LOW_ACTIVE
x13	not yet defined	Milling_machine_at_conveyor_belt_2_LOW_ACTIVE
x14	not yet defined	Milling_head_is_up_LOW_ACTIVE
x15	not yet defined	Milling_head_is_down_LOW_ACTIVE
x16	not yet defined	Emergency_Stop

Table 10: Production cell sensors

Var	Name	Function
y0	not yet defined	Transport_table_move_to_conveyor_belt_3
y1	not yet defined	Transport_table_move_to_conveyor_belt_1
y2	not yet defined	Transport_table_drive_conveyor_belt_similar_to_conveyor_belt_1
y3	not yet defined	Transport_table_drive_conveyor_belt_similar_to_conveyor_belt_3
y4	not yet defined	Conveyor_belt_1_drive_belt
y5	not yet defined	Turntable_1_rotate_to_conveyor_belt_1
y6	not yet defined	Turntable_1_rotate_to_conveyor_belt_2
y7	not yet defined	Turntable_1_drive_belt
y8	not yet defined	Conveyor_belt_2_drive_belt
y9	not yet defined	Turntable_2_rotate_to_conveyor_belt_2
y10	not yet defined	Turntable_2_rotate_to_conveyor_belt_3
y11	not yet defined	Turntable_2_drive_belt
y12	not yet defined	Conveyor_belt_3_drive_belt
y13	not yet defined	Milling_machine_retreat_from_conveyor_belt_2
y14	not yet defined	Milling_machine_approach_conveyor_belt_2
y15	not yet defined	Milling_head_rise
y16	not yet defined	Milling_head_lower
y17	not yet defined	Milling_head_drive_head

Table 11: Production cell actuators

5.4. 3-axis portal

The model 3-axis-portal simulates a stationary used handling robot with an orthogonal work space used for passing on work pieces to processing or sorting unit, as used e. g. in factories being automated in a large degree. The model consists of the portal robot that is able to move in three linear directions and an electromagnetic gripper, fit to be moved in Z-direction, a piece store and a discharge station. The end positions of the several moving parts are each recognized by software end position switches. The simulated process shows metal work pieces being withdrawn from the store by the electromagnetic gripper, being moved to the discharge station and there getting put down: At the beginning of the sequence the moving parts of the robot execute a reference tour, in order to equalize the incremental distance measuring systems of the X- and Y-axes with their real positions. The reference tour is done with the gripper being in its upper end position. After this, the gripper is moved in X- and Y-direction until it has reached its demanded position above the piece store. The gripper moves in –Z-direction until it touches the work piece. The magnet gets switched on and the part is attached to the gripper. The gripper moves in +Z-direction until it has again reached its upper end position. Following this, it executes a movement in X- and Y-direction until it has reached a position above the discharge station. Having reached this state, the gripper again moves in –Z-direction until the attached piece is put on the discharge station, where it gets recognized by an inductive proximity switch. The work piece is withdrawn by deactivating the electromagnetic gripper. In order to save the whole unit from being damaged by moving out of the allowed work space, caused by a mistake in programming the control unit, the linear directions are supplementary equipped with hardware end position switches, which cause an immediately stop of the corresponding axis in case of being actuated. The electronic concept of the model only enables a restarting of the robot by executing a movement towards the work space.

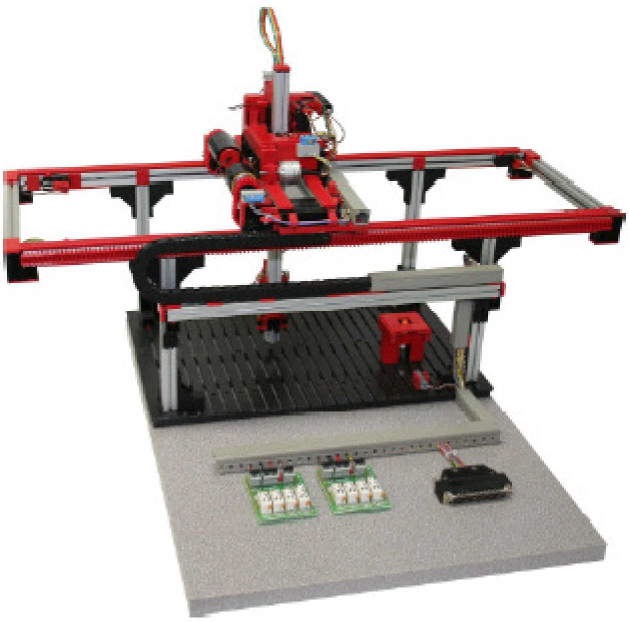

Figure 21: 3-axis portal mechanical physical system

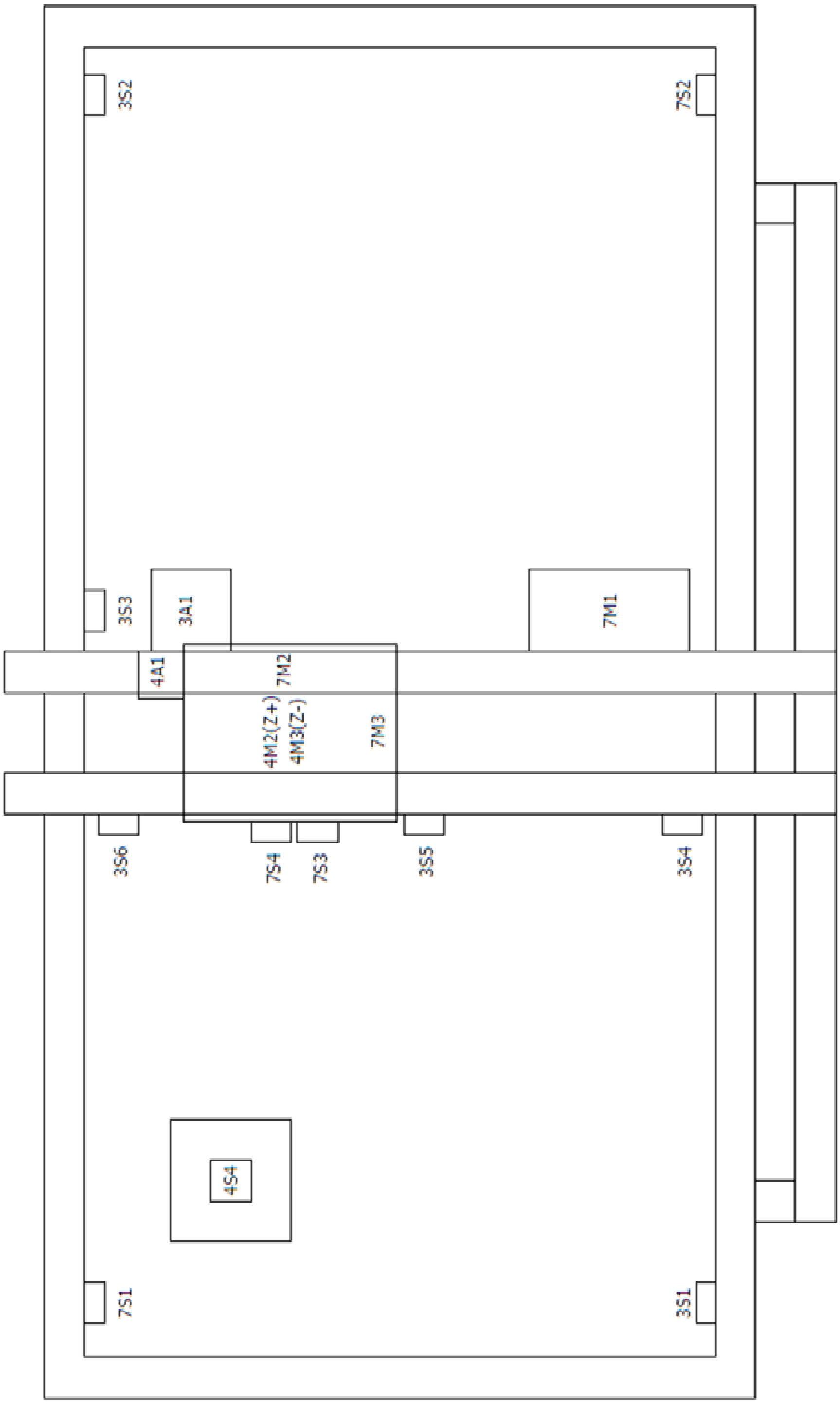

Figure 22: 3-axis portal mechanical physical system schematic

Var	Name	Function
x0	not yet defined	x_axis_at_position_x_minus
x1	not yet defined	x_axis_at_position_x_plus
x2	not yet defined	x_axis_reference_position
x3	not yet defined	y_axis_at_position_y_minus
x4	not yet defined	y_axis_at_position_y_plus
x5	not yet defined	y_axis_reference_position
x6	not yet defined	z_axis_at_position_z_plus
x7	not yet defined	z_axis_at_position_z_minus
x8	not yet defined	proximity_switch
x9	not yet defined	start_button
x10	not yet defined	abs_position_x_15
x11	not yet defined	abs_position_x_14
x12	not yet defined	abs_position_x_13
x13	not yet defined	abs_position_x_12
x14	not yet defined	abs_position_x_11
x15	not yet defined	abs_position_x_10
x16	not yet defined	abs_position_x_09
x17	not yet defined	abs_position_x_08
x18	not yet defined	abs_position_x_07
x19	not yet defined	abs_position_x_06
x20	not yet defined	abs_position_x_05
x21	not yet defined	abs_position_x_04
x22	not yet defined	abs_position_x_03
x23	not yet defined	abs_position_x_02
x24	not yet defined	abs_position_x_01
x25	not yet defined	abs_position_x_00
x26	not yet defined	abs_position_y_15
x27	not yet defined	abs_position_y_14
x28	not yet defined	abs_position_y_13
x29	not yet defined	abs_position_y_12
x30	not yet defined	abs_position_y_11
x31	not yet defined	abs_position_y_10
x32	not yet defined	abs_position_y_09
x33	not yet defined	abs_position_y_08
x34	not yet defined	abs_position_y_07
x35	not yet defined	abs_position_y_06
x36	not yet defined	abs_position_y_05
x37	not yet defined	abs_position_y_04
x38	not yet defined	abs_position_y_03
x39	not yet defined	abs_position_y_02
x40	not yet defined	abs_position_y_01
x41	not yet defined	abs_position_y_00

Table 12: 3-axis portal sensors

Var	Name	Function
y0	not yet defined	x_axis_to_x_minus
y1	not yet defined	x_axis_to_x_plus
y2	not yet defined	y_axis_to_y_minus
y3	not yet defined	y_axis_to_y_plus
y4	not yet defined	z_axis_to_z_plus
y5	not yet defined	z_axis_to_z_minus
y6	not yet defined	magnet

Table 13: 3-axis portal actuators

5.5. Water level control

Coming soon…

5.6. Storage warehouse

The High Level Storage Warehouse simulates an automatically working high-level-storage system as used for example in many industrial branches. The model consists of a rack, being divided up in 5 x 10 storage places, a warehouse operating device, being portable in X-direction, and two charge / discharge stations. A cage being portable in Z-direction and including a telescopic palette carrier, that is portable in Y-direction, is attached to the warehouse operating device. The simulated process shows palettes being stored and withdrawn from the high-level storage: In case of one charge station being occupied by a palette, the telescopic palette carrier moves to the station and takes over the palette. This is recognized by a reflection light switch. Following this, the warehouse operating device brings the palette to the intended storage place in an optimized manner by moving in X- and Z-direction at the same time. Occupying a storage place is recognized by software. In order to enable a quick movement to the storage place on the one hand and a safe lay-in-movement on the other hand, the horizontal rack positions are equipped with advanced mechanical switches that allow retarding the warehouse operating device before reaching the intended position. Withdrawing palettes occurs in the same manner, done in inverse chronological order. In cause of bolting the Y-axes against the two others moving the palette carrier in X- and Z-direction is only possible, if the palette carrier is in its middle position. Moreover, the X-axes is equipped with a hardware end position switch to prevent the whole warehouse system from fatal mistakes in using the conveyor or programming the control unit. The High-level-storage warehouse is fit to be combined with further modules and standard models in order to automate the periphery of the warehouse.

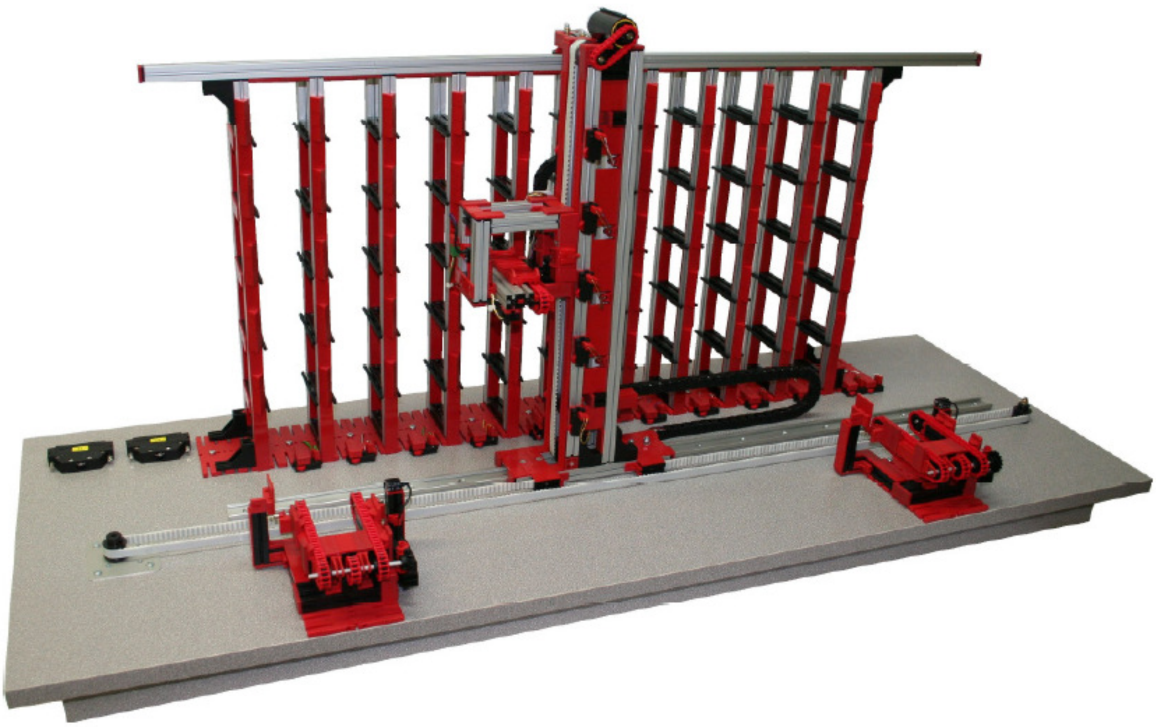

Figure 23: Storage warehouse electro mechanical physical system

Var	Name	Function
x0	not yet defined	
x1	not yet defined	
x2	not yet defined	
x3	not yet defined	
x4	not yet defined	
x5	not yet defined	
x6	not yet defined	
x7	not yet defined	
x8	not yet defined	
x9	not yet defined	
x10	not yet defined	
x11	not yet defined	
x12	not yet defined	
x13	not yet defined	
x14	not yet defined	
x15	not yet defined	
x16	not yet defined	
x17	not yet defined	
x18	not yet defined	
x19	not yet defined	
x20	not yet defined	
x21	not yet defined	
x22	not yet defined	
x23	not yet defined	
x24	not yet defined	
x25	not yet defined	
x26	not yet defined	
x27	not yet defined	
x28	not yet defined	
x29	not yet defined	
x30	not yet defined	
x31	not yet defined	
x32	not yet defined	
x33	not yet defined	
x34	not yet defined	

Table 14: Storage warehouse sensors

Var	Name	Function
y0	not yet defined	
y1	not yet defined	
y2	not yet defined	
y3	not yet defined	
y4	not yet defined	
y5	not yet defined	
y6	not yet defined	
y7	not yet defined	
y8	not yet defined	
y9	not yet defined	
y10	not yet defined	
y11	not yet defined	
y12	not yet defined	
y13	not yet defined	
y14	not yet defined	
y15	not yet defined	
y16	not yet defined	
y17	not yet defined	
y18	not yet defined	
y19	not yet defined	
y20	not yet defined	
y21	not yet defined	
y22	not yet defined	
y23	not yet defined	
y24	not yet defined	
y25	not yet defined	
y26	not yet defined	
y27	not yet defined	
y28	not yet defined	

Table 15: Storage warehouse actuators

6. Revisions

A.0	06.08.2014	First version of documentation.
A.1	26.01.2015	Minor corrections. Added pictures and more model descriptions.