



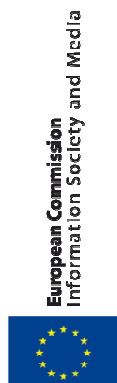
BEMO-COFRA

Brazil-Europe Monitoring and Control Framework

(Project No. 288133)

D5.1.1 Initial infrastructure for distributed control logic

Published by the BEMO-COFRA Consortium
Dissemination Level: Public



Project co-funded by the European Commission within the 7th Framework Programme
and
Conselho Nacional de Desenvolvimento Científico e Tecnológico
Objective ICT-2011-EU-Brazil

Document control page

Document file: D5.1.1 Initial infrastructure for distributed control logic v1.0
Document version: 1.0
Document owner: Juliana Vilela (COMAU)

Work package: WP5 – Distributed Control Logic and Enabling Features
Task: T5.1 – Distributed Control Logic
Deliverable type: P

Document status: approved by the document owner for internal review
 approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Juliana Vilela	24-08-12	ToC.
0.2	Juliana Vilela	05-09-12	Main pictures.
0.3	Juliana Vilela	15-09-12	Chapter 3 & 4.
0.4	Juliana Vilela	18-09-12	Chapter 5.
0.5	Juliana Vilela	19-09-12	Chapter 1 & 2.
0.6	Juliana Vilela	01-10-12	Add text and modifications according to the comments.
1.0	Juliana Vilela	05-10-12	Small modifications according to reviews. Ready to submit.

Internal review history:

Reviewed by	Date	Summary of comments
Markus Taumberger	2012-10-01	
Marcelo Borghetti	2010-10-05	

Legal Notice

The information in this document is subject to change without notice.

The Members of the BEMO-COFRA Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the BEMO-COFRA Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Possible inaccuracies of information are under the responsibility of the project. This report reflects solely the views of its authors. The European Commission is not liable for any use that may be made of the information contained therein.

Index:

1. Executive summary	5
2. Introduction	6
2.1 Background	6
2.2 Purpose, context and scope of this deliverable	7
3. General Line	8
3.1 General Line Layout	8
3.2 Control Logic – Production Flow	9
3.3 Poka-Yoke - Mistake Proofing	9
4. Framing Station	11
4.1 Layout Framing Station.....	11
4.2 Control Logic.....	13
4.3 Flowchart.....	14
4.4 Main Functions.....	17
5. Manufacturing Demonstration	38
5.1 Scenario	38
5.2 Flowchart.....	39
5.3 Ladder Program	40
6. References.....	46

Figures

Figure 1: Devices and network distribution part 1.	8
Figure 2: Devices and network distribution part 2.	9
Figure 3: General view - Framing Station.	11
Figure 4: Frame - Devices in detail.	12
Figure 5: Frame Blocker.	13
Figure 6: Framing Station Control Logic - part 1 and 2.	14
Figure 7: Framing Station Control Logic - part 3 and 4.	15
Figure 8: Framing Station Control Logic - part 5.	16
Figure 9: Function: Check Devices Working.	17
Figure 10: Function: Check if the station is at rest.	18
Figure 11: Function: Check Parity Error.	19
Figure 12: Function: Enable Operation Cycle.	21
Figure 13: Function: Step 0 - Request Start Mute.	23
Figure 14: Function: Step 1 - Skid Move and Positioning.	24
Figure 15: Function: Step 2 - Sensors Reading.	24
Figure 16: Function: Step 3 - Lift Descent.	25
Figure 17: Function: Step 4 - Check Underbody Presence.	26
Figure 18: Function: Step 7 - Close Locks Underbody.	27
Figure 19: Function: Step 8 - Close Frame.	27
Figure 20: Function: Step 9 - Insert work supports frame.	28
Figure 21: Function: Step 10 - Lock Frame.	29
Figure 22: Function: Step 11 - Block Frame.	30
Figure 23: Function: Step 19 - Start Robot.	31
Figure 24: Function: Step 26 - Start Weld.	32
Figure 25: Function: Step 30 - Unblock Frame.	33
Figure 26: Function: Step 31 - Unlock Frame.	33
Figure 27: Function: Step 39 - Open Locks Underbody.	34
Figure 28: Function: Step 41 - Lift Ascent.	35
Figure 29: Function: Step 42 - Sensors Writing.	35
Figure 30: Function: Step 43 - Robot attachment work done.	36
Figure 31: Function: Step 44 - Cycle attachment work done.	37
Figure 32: Demonstration Scenario.	39
Figure 33: Demo Control Logic Flowchart.	40
Figure 34: Demo Ladder Program - part. 1.	42
Figure 35: Demo Ladder program - part. 2.	43

1. Executive summary

The presented deliverable aims to describe the state-of-the-art of control logic used at manufacturing industries. A Body & Welding line is used as example with focus on the framing station. The control logic monitors the devices distributed on the line and controls them, indicating the correct time to act or not, guarantying the correct flow of production.

Normally, when you talk about control logic development the first step is to define the process to be controlled. The focus of this deliverable is the production flow. Independent of what is produced; the industry should control the production flow to guarantee a correct product manufacturing and the achievement of the production goals. With the process in mind, a whole line view allows to plan how many stations and devices will participate in the control logic, allowing the quantity estimation of Programming Logic Controllers (PLC).

A flowchart is created to represent the process dynamic and to define the main inputs and outputs of the control logic. Thus with the whole line control logic mapped, it can be seen how the stations interact with each other. In the next step, the behaviour of the station devices, i.e. the station control logic, has to be defined.

The process to develop the station control logic is similar to the whole line's process. First the station behaviour should be mapped including the devices, which participate in the production flow. A flowchart with the sequence of activities is created and the variables (inputs and outputs) are defined and mapped into the PLCs. Now, it's possible to programme the PLCs with the correct control logic, using a specific programming language, such as Ladder, for example.

In M14 a demonstration of how the BEMO-COFRA framework should interact with a real manufacturing line will be carried out. In this deliverable the process of the Demo control logic development is illustrated.

The work on the control logic will continue to identify which existing sensors can be changed to wireless sensors and, how the control logic needs to be updated to integrate these new devices and the BEMO-COFRA functionalities. The next deliverable is 'D5.1.2 Final infrastructure for distributed control logic', which is planned for M25.

2. Introduction

This deliverable aims to define the state-of-the-art for control logic in the manufacturing industry. The control logic related to the production flow was chosen as base, because no matter which kind of product an industry produces, it is necessary to control the production process to guarantee the correct product manufacturing and the achievement of the company goals. So, a Body & welding line is used as an example to demonstrate a control logic is structured in a production line. A particular focus on the framing station was given, because the whole line control logic is too complex for a detailed discussion.

The control logic is an important part in a production line, as it is responsible for the correct flow of the production, and guarantying the product quality.

The control logic monitors the devices distributed on the line and controls them, indicating their correct time to act or not. The control logic is composed by sensors, actuators and PLC's. The sensors are designated to monitor the devices status and send these information to the PLC. The line control logic, or production flow, programmed inside the PLC, takes these sensors information and provides for the actuators their new activities. Now, the actuators are able to modify the system, keep going with the production flow.

2.1 Background

The BEMO-COFRA project aims to develop an innovative distributed framework which allows networked monitoring and control of large-scale complex systems by integrating heterogeneous smart objects, legacy devices and sub-systems, cooperating to support holistic management and to achieve overall system efficiency with respect to energy and raw materials.

The BEMO-COFRA features a Service oriented Architecture (SoA) and a middleware able to expose smart objects, legacy devices and sub-systems' capabilities by means of web services thus supporting syntactic and semantic interoperability among different technologies coexisting in the overall monitoring and control framework. Wireless Sensor and Actuator Network (WSAN) devices, legacy sub-systems and devices will thus be able to interact and cooperate, orchestrated by a manager in charge of enforcing a distributed logic with the overall monitoring and control network.

BEMO-COFRA reuses the results of the well-reputed Hydra IP and Pobicos STREP and the recently started ebbits IP featuring a Service Oriented Architecture (SOA) and a middleware able to expose smart objects, legacy devices and sub-systems' capabilities by means of web services. Syntactic and semantic interoperability among coexisting technologies in the overall monitoring and control framework is made available.

The integration of heterogeneous smart objects, legacy devices and sub-systems will achieve overall systems' efficiency with respect to energy and raw materials and support holistic management. The BEMO-COFRA project will address both technological aspects and user needs to promote a wider adoption of large-scale networked monitoring and control solutions.

2.2 Purpose, context and scope of this deliverable

With the objective to show the state-of-the-art of a production flow control logic a Body & Welding line is used as an example with the focus on the Framing Station, the most complex station of the whole line.

The deliverable is divided in 6 chapters:

- Chapter 1 – Executive Summary: a brief description of the deliverable’s subject;
- Chapter 2 – Introduction: a brief summary of the deliverable’s subject and context structure;
- Chapter 3 – General line: a general view of the whole Body & welding line control logic, indicating how the control logic system is structured. This chapter contains a brief explanation of the Poka-Yoke methodology normally used for control logic development;
- Chapter 4 – Framing Station: the control logic is explained in detail in this chapter. A general flowchart shows the interaction of the main functions with each function being represented by a black box. The aim of this chapter is to give an idea of how complex the control logic in a real manufacturing line is.
- Chapter 5 – Manufacturing Demonstration: a manufacturing line model has been developed to demonstrate how the BEMO-COFRA framework can interact with a real system. In this chapter the control logic development for the demonstration is presented. All the steps of the demo scenario are discussed, from the flowchart to the ladder program;
- Chapter 6 – References.

3. General Line

The main objective of this chapter is to show how a production line is structured regarding its monitoring and control activities. A Body & Welding line has been used as an example for this and the next chapter, with the focus on a specific station, that was base of this study. (Comau, 2012)

3.1 General Line Layout

Before the structure of the control logic is given, the line layout will be introduced with the most relevant devices of the monitoring and control activities.

The line used as example uses five PLCs for the whole line. The line is split in four groups of stations, each one controlled by a PLC and one PLC dedicated only to the Framing Station, which is considered the most complex station of the line. These PLCs control the production flow, acting on the conveyors, SKIDs and Robots. The welding parameters are controlled by a separated controller, which is not part of this work.

Figures 1 and 2 depict, the line is composed of motors or inverters, cylinders, clamps, sensors, robots and conveyors. The sensors types are mostly proximity, presence and pressure. The PLC receives the information from the sensors, including the sensors linked to the clamps and cylinders, processes that information and sends actions to the actuators.

The operation of the PLC and the status of each device, besides the alarms generated, are shown on a human-machine interface, where the operator can also control the line in manual mode.

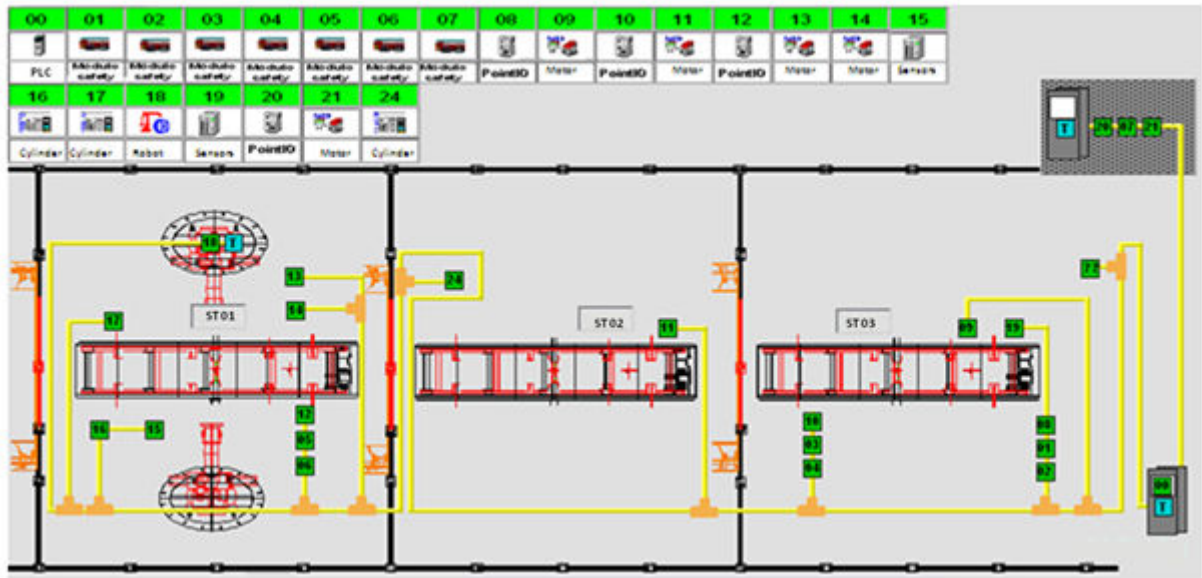


Figure 1: Devices and network distribution part 1.

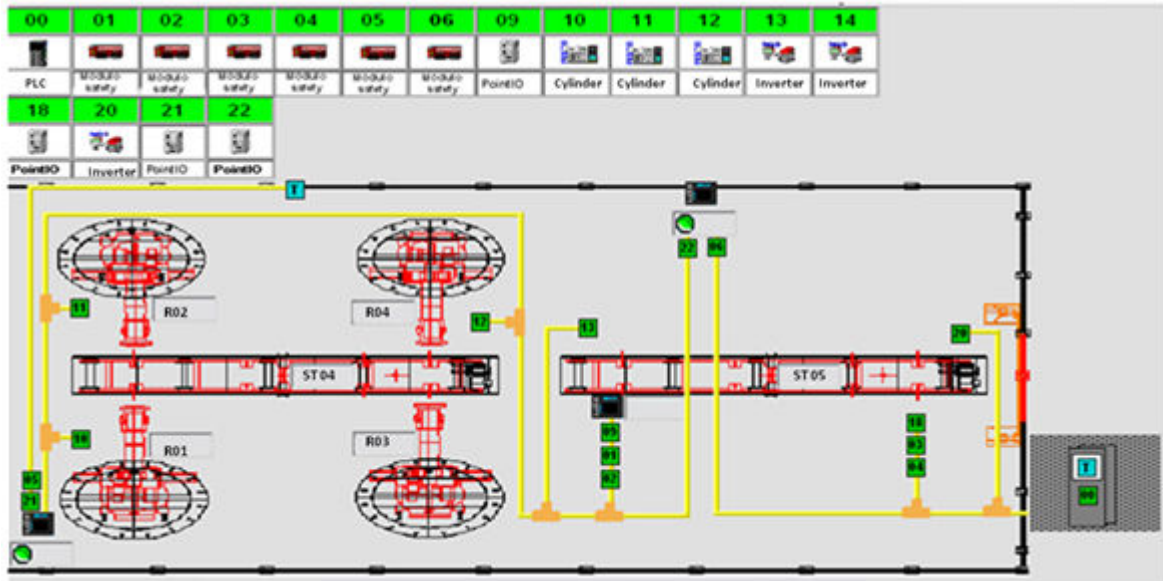


Figure 2: Devices and network distribution part 2.

3.2 Control Logic – Production Flow

The focus of this deliverable is the production flow control logic, which is implemented in every industry shop-floor.

Before the PLC programming, the flow of the production is mapped regarding the correct station sequence, order of activities, devices activation and status. The principal inputs and outputs are highlighted for the correct line running. Then all this production logic is inserted on the PLC, being the control logic with what the PLC will work. Failures must be predicted for the control logic as well, allowing alarms and the line stop.

Normally the production flow control logic is based on the Poka-Yoke methodology; when a failure is detected, an alarm is generated and the whole line stops, waiting the problem being fixed, this form of “mistake-proofing”.

3.3 Poka-Yoke - Mistake Proofing

The term Poka-Yoke was invented in 1960’s by Shiego Shingo. The main idea behind the term Poka-yoke is to design ones process so that the likelihood of mistakes is almost impossible and at least easily detected and corrected. (Wikipedia, 2012)

“Mistake-proofing” means of providing a visual or other signal to indicate a characteristic state. Often referred to as “error-proofing”, poka-yoke is actually the first step in truly error-proofing a system. Error-proofing is a manufacturing technique of preventing errors by designing the manufacturing process, equipment, and tools so that an operation literally cannot be performed incorrectly. (isixsigma,2012)

Devices or procedures associated with Poka-Yoke fall into two main categories: prevention and detection.

A prevention device or process makes it impossible to make a mistake during the procedures. A few examples of a prevention device include window envelopes and electrical plugs. The three point plugs commonly known in Brazil were designed so that they can only fit into the wall socket in one way.

The detection device or process indicates to a user when a mistake has been made so that the user can quickly correct the problem. For example when you leave your key in the car's ignition when opening the door, the car starts beeping.

The application of Poka-Yoke spans over three levels associated with two different categories of Poka-Yoke:

1. Elimination of spills, leaks and losses at the source or prevention of a mistake from being committed;
2. Detection of a loss or mistake as it occurs, allowing the correction before it becomes a problem and;
3. Detection of a loss or mistake after it has occurred, just before it blows up into a major issue.

4. Framing Station

The framing station is the most complex station on a Body & Welding line, having a PLC dedicated only to its activities and control logic.

To understand the framing station complexity first its layout is shown, giving an idea of how many devices need to be monitored and controlled. After that the control logic flowchart depicts the corresponding station running and with each main function being detailed in black box and flowchart formats.

The objective of this chapter is to give a detailed idea of how a station control logic is defined, regarding a Body & Welding line that is composed of about 35 stations, each one with its complexity, and it's necessary main program, that integrate all the stations on the whole production flow. (The deliverable 'D2.1 State of play in production monitoring and control systems' gives a very detailed description of a Body & Welding line and the activities in each station).

4.1 Layout Framing Station

The framing station is basically composed of a chassis, a couple of frames and an opengate, which change according to the car model, four robots, a blocker and a skid. The next figure shows a general view of the Framing Station.

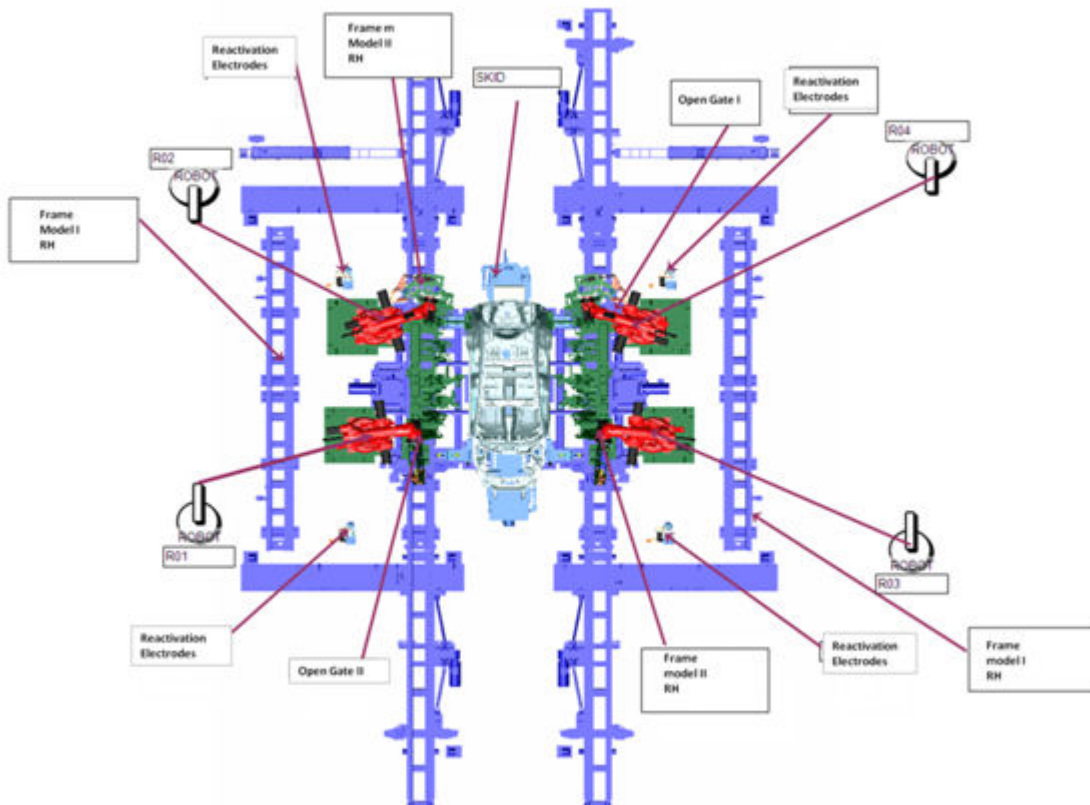


Figure 3: General view - Framing Station.

The complexity of this station attributes to the amount of devices which it is composed of. The next figure shows a frame in detail, highlighting three main devices and their principal clamps, cylinders and switches, which include sensors.

According to the car model which arrives at the station, the frame pair is changed for the right model. When the frames arrive at the station the frame blocker locks them, allowing the clamps to close and cylinders to move and in sequence the welding by the robots.

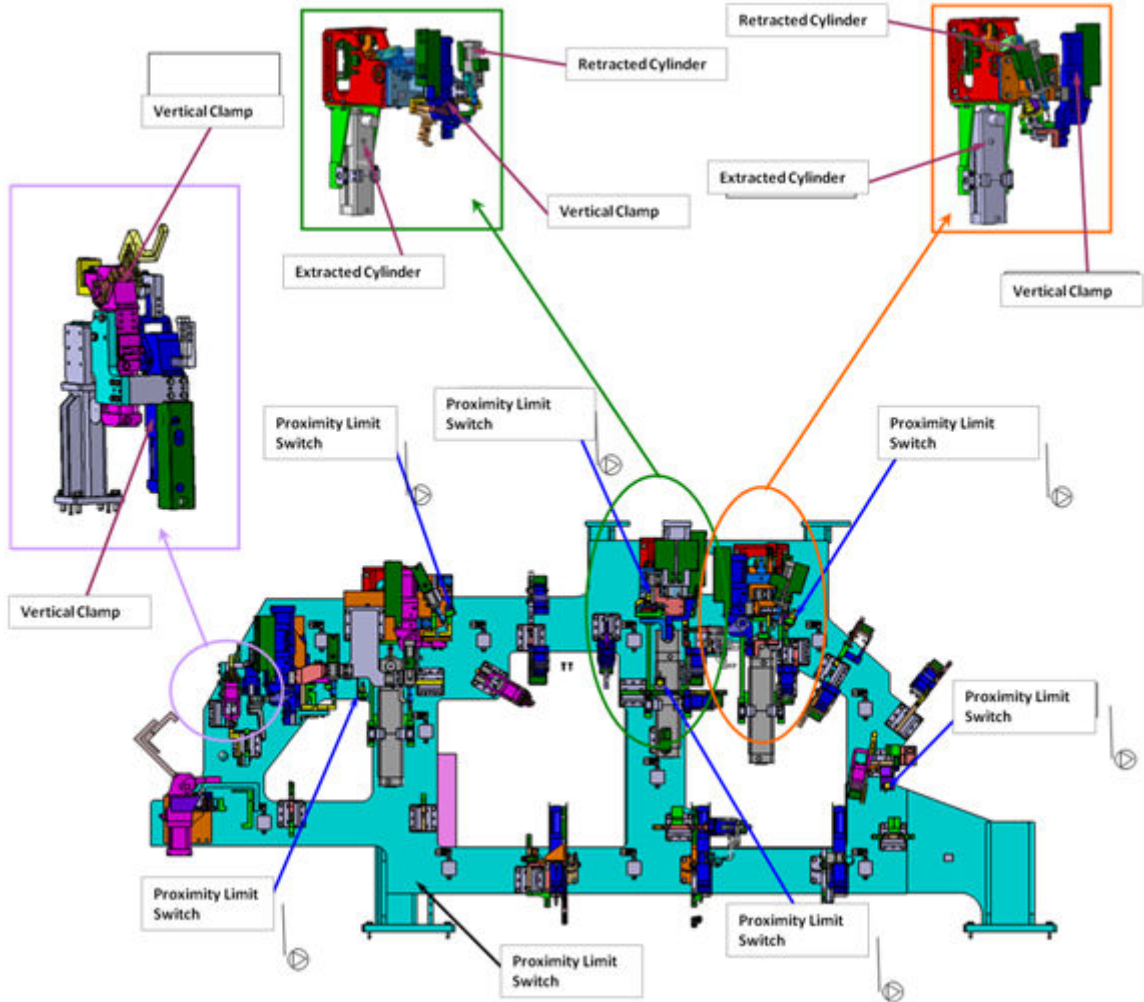


Figure 4: Frame - Devices in detail.

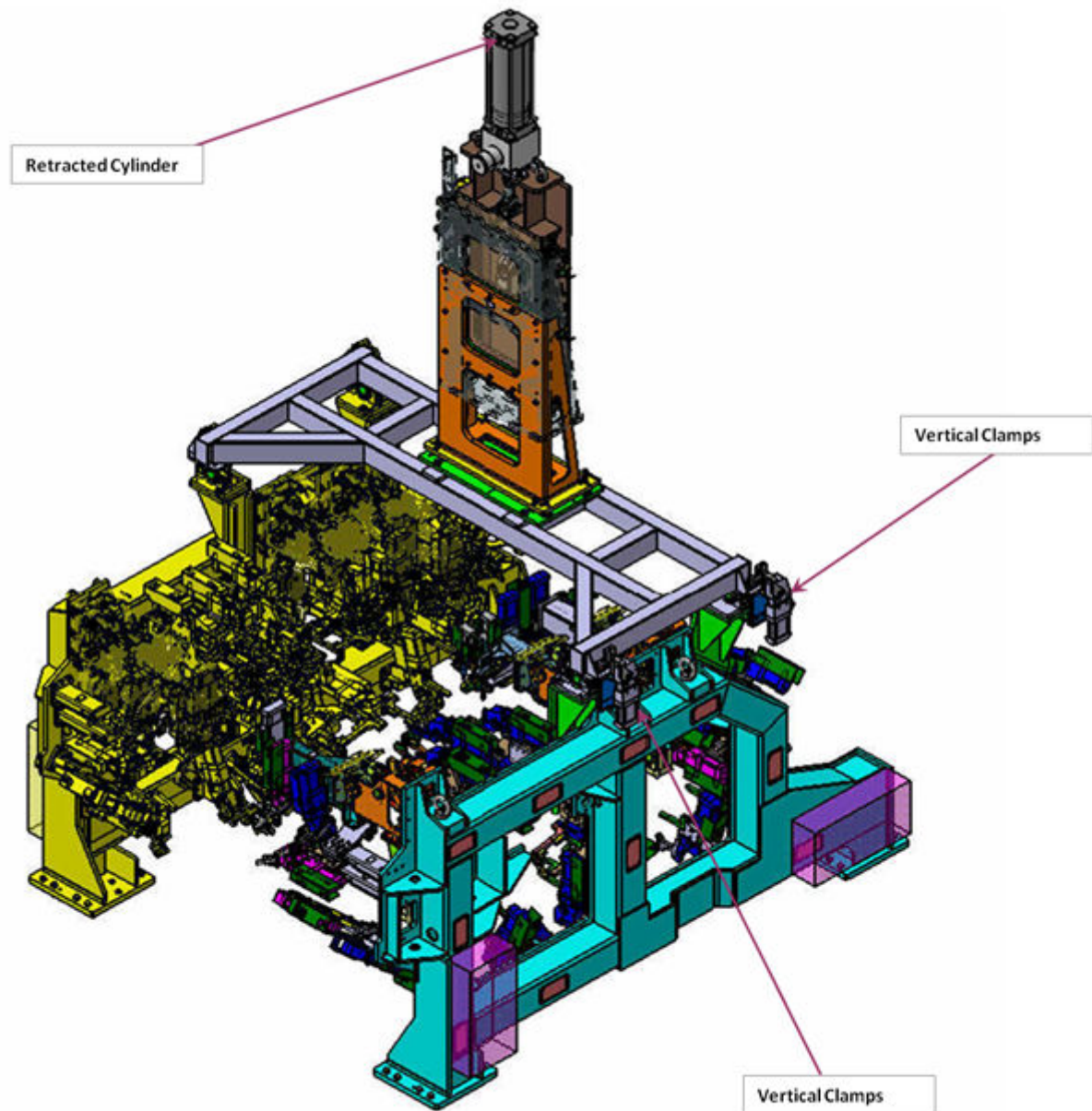


Figure 5: Frame Blocker.

4.2 Control Logic

The focus of this section is summarizing the production flow control logic of the framing station. For the welding activity there is an additional controller, which controls the welding parameters, but this kind of control is out of scope for this deliverable for being too specific of a Body Welding Line. The control logic follows the flowchart presented in the next section 4.3, which represents the production flow of the Framing Station.

In a brief summary the control logic acts in the following way: first the PLC checks if all devices are running ok, if there is any problem the line stops and an alarm is generated. Remember that the control logic is developed according to the Poka-Yoke methodology, so if a problem happens the whole line stops. With all devices running ok, the PLC checks if the station is at rest, because only in that status it is possible to start a new operation cycle. If the station is at rest, the PLC checks if any

errors of parity occurred during the sensors signal transmission, only after that a new operation cycle is enabled. With the start command the skid begins to move to the station with the car body, the side parts come down with the lift. When all the parts are in the station, the body car is locked and blocked, and the robots are authorized to start the welding. After the welding activities the clamps open and the body car is unblocked and unlocked. The lift goes up, and the new sensors states are written. The operation cycle is finished.

4.3 Flowchart

For better readability the following figures show the control logic in flowchart with the main steps and functions. On the framing station PLC this control logic is implemented on Ladder language.

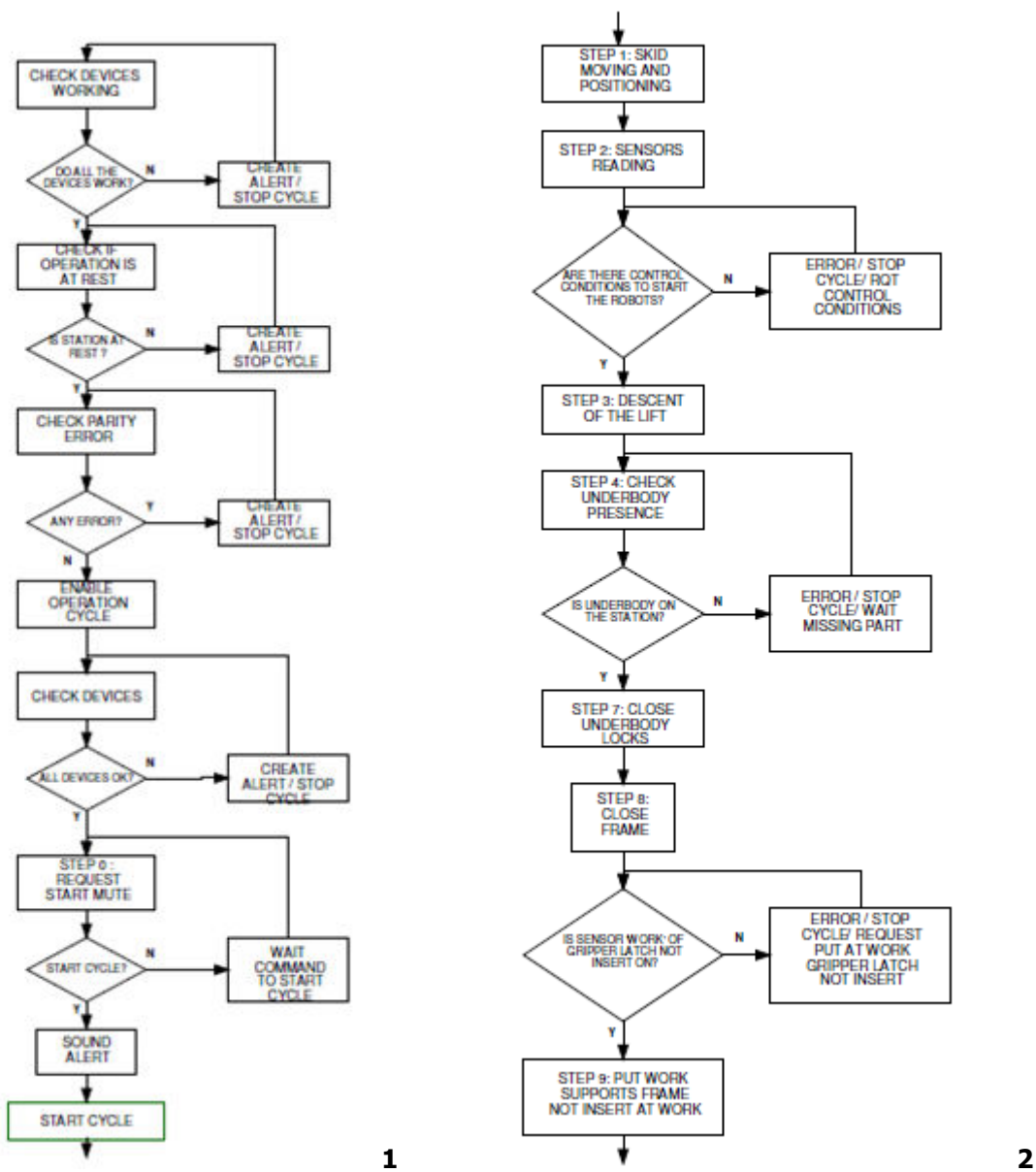


Figure 6: Framing Station Control Logic - part 1 and 2.

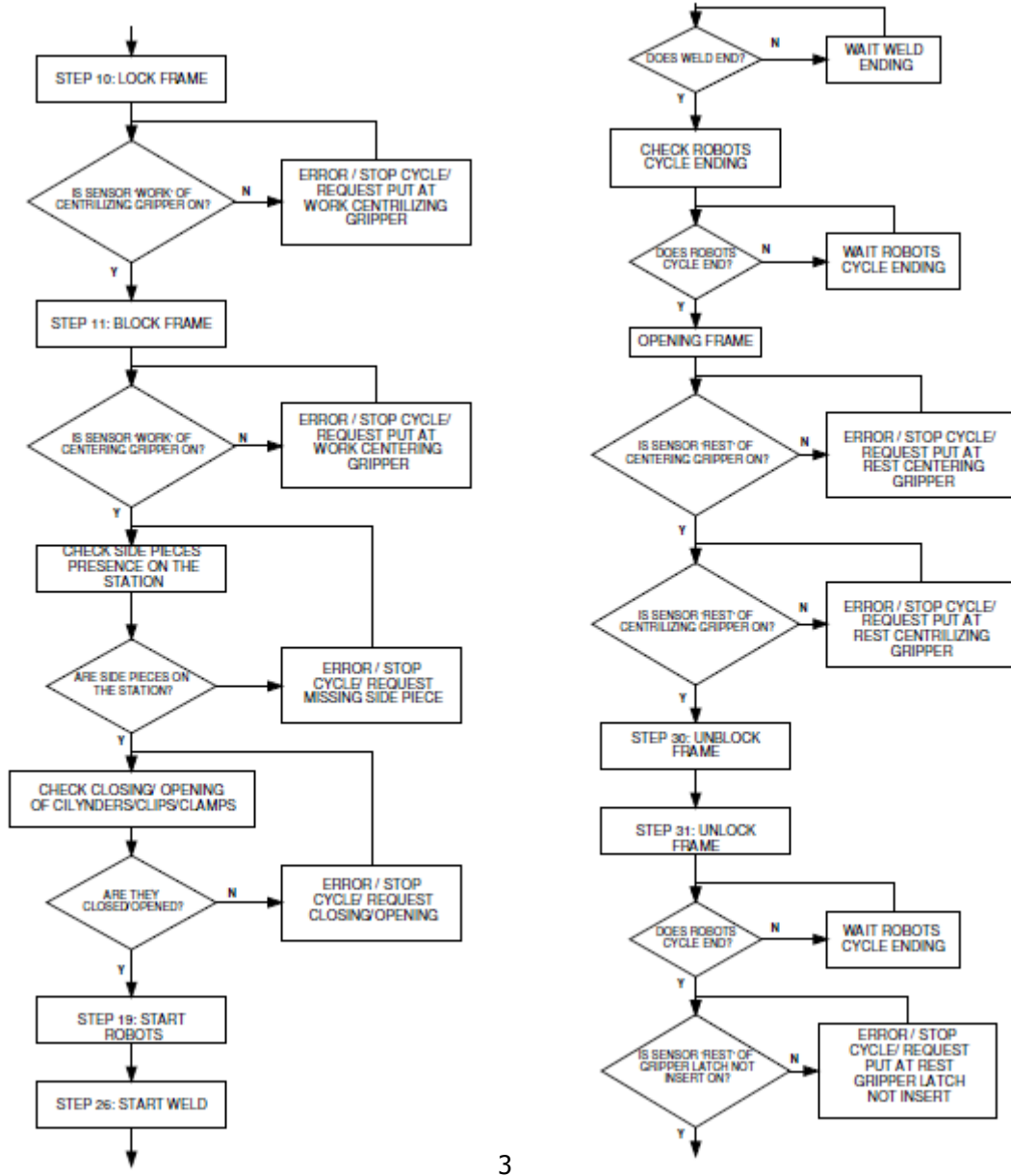
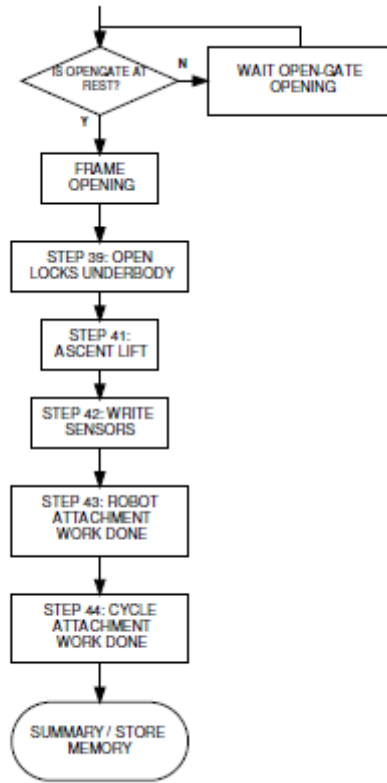


Figure 7: Framing Station Control Logic - part 3 and 4.



5

Figure 8: Framing Station Control Logic - part 5.

4.4 Main Functions

On the flowchart showed at the previous section, each square correspond to a function which is represented bellow in a black box format – inputs and outputs only – and in its own flowchart, that represents its internal control logic.

Check Devices Working

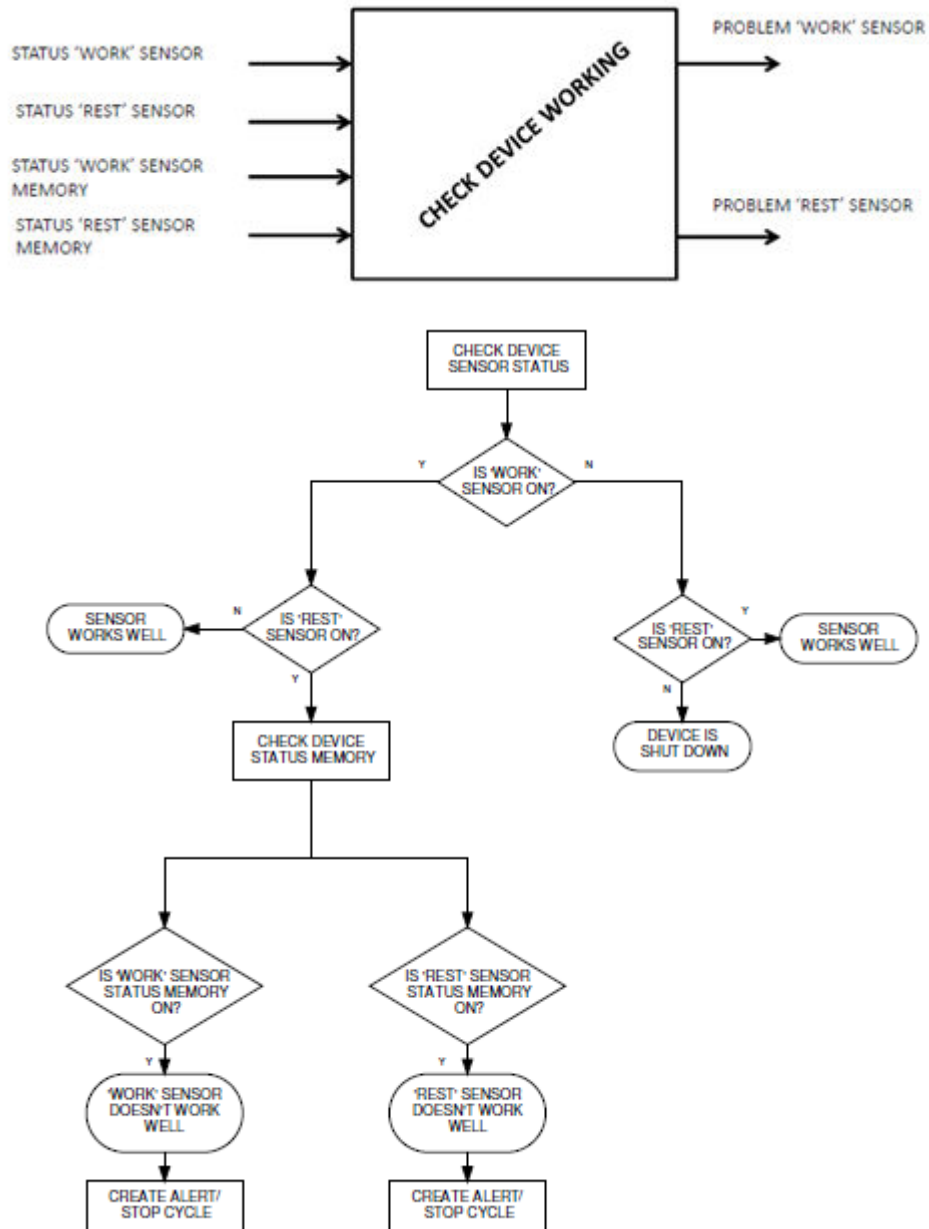


Figure 9: Function: Check Devices Working.

Check if the station is at rest

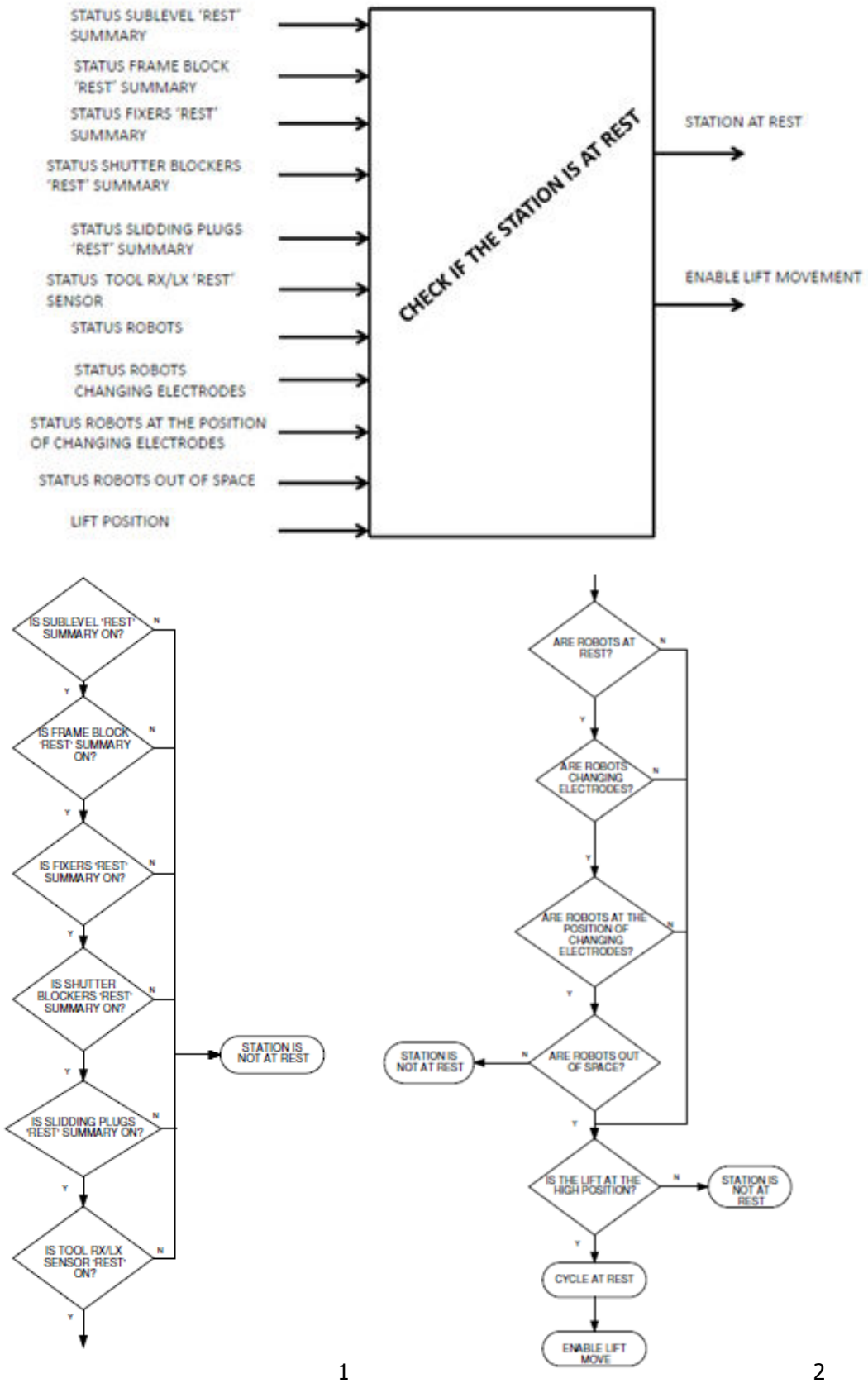


Figure 10: Function: Check if the station is at rest.

Check Parity Error

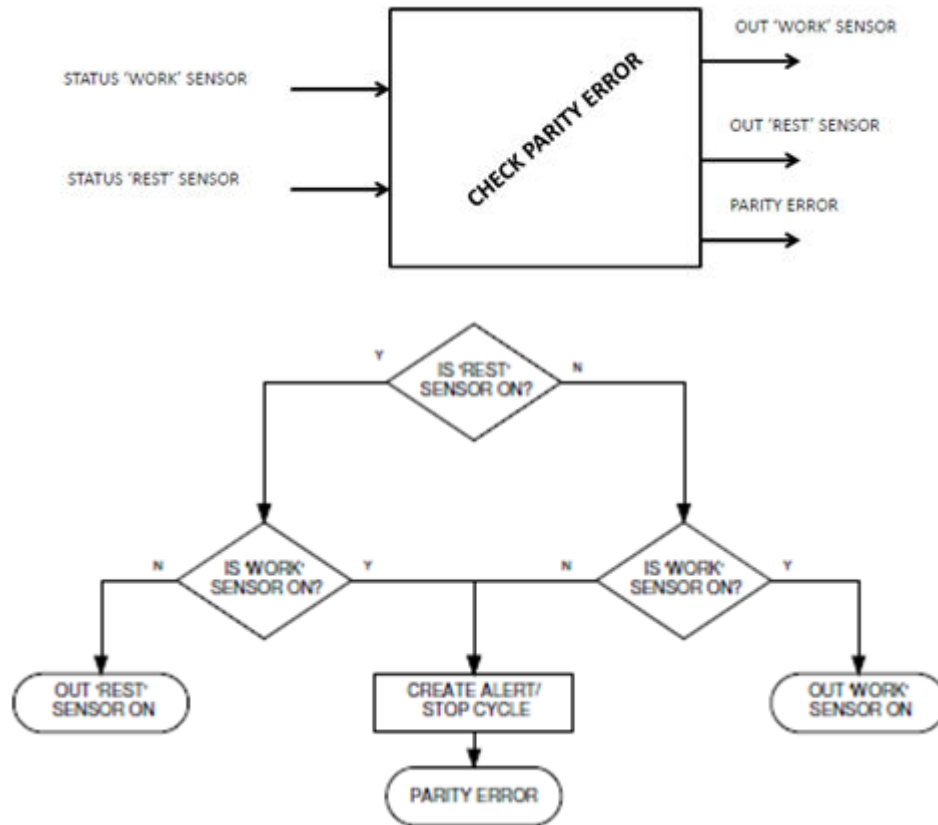
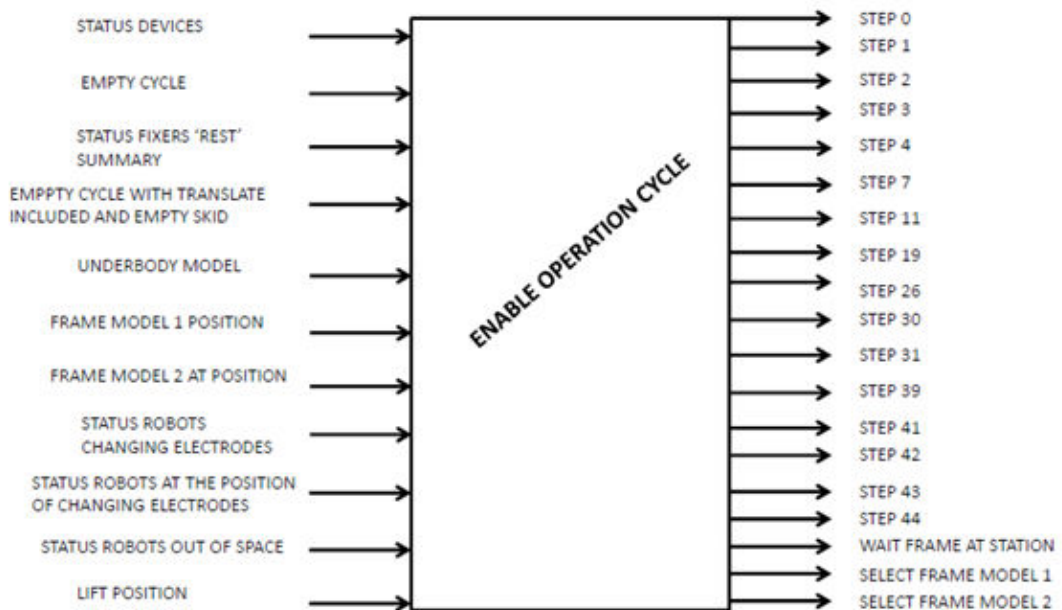
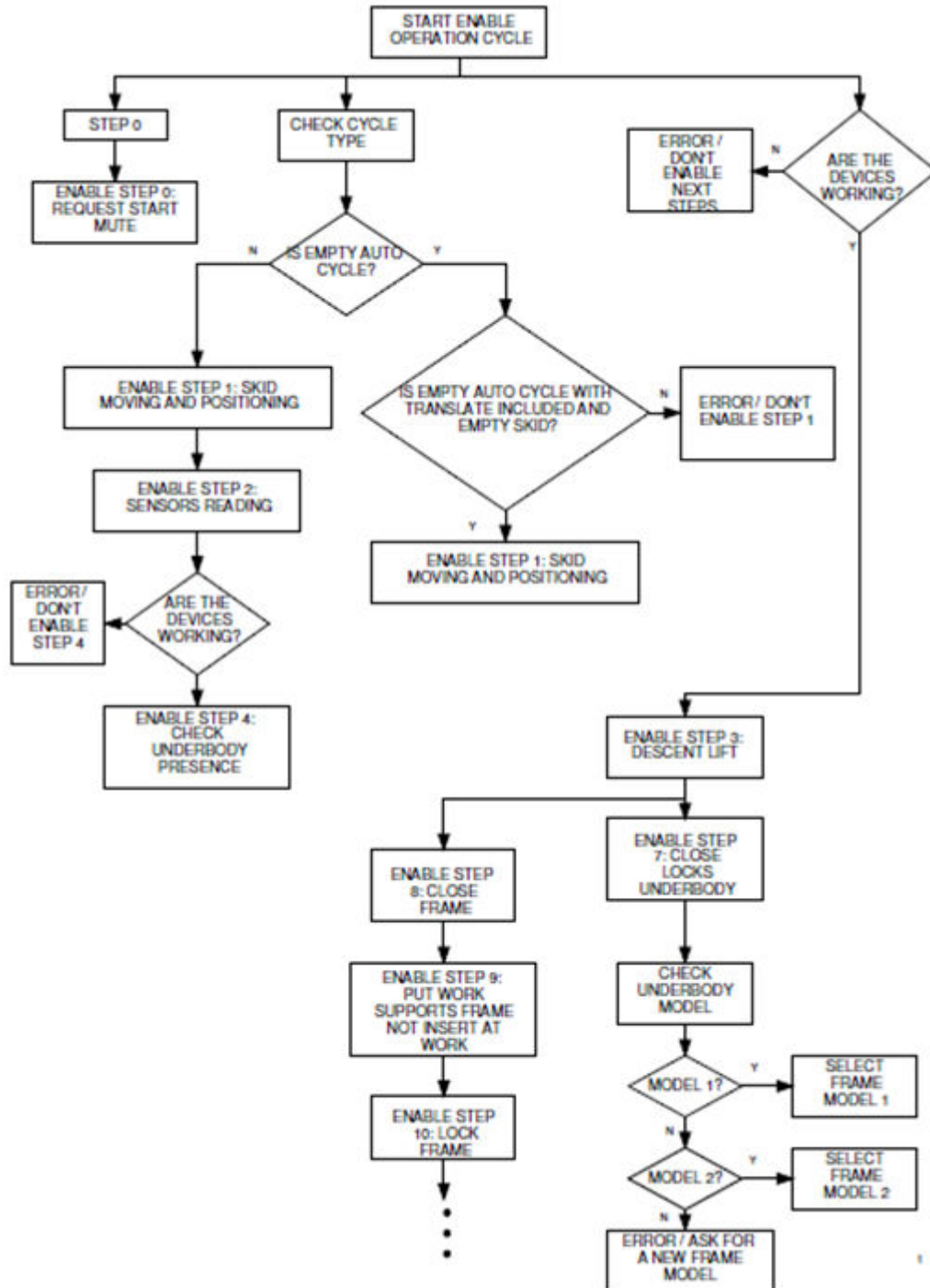


Figure 11: Function: Check Parity Error.

Enable Operation Cycle





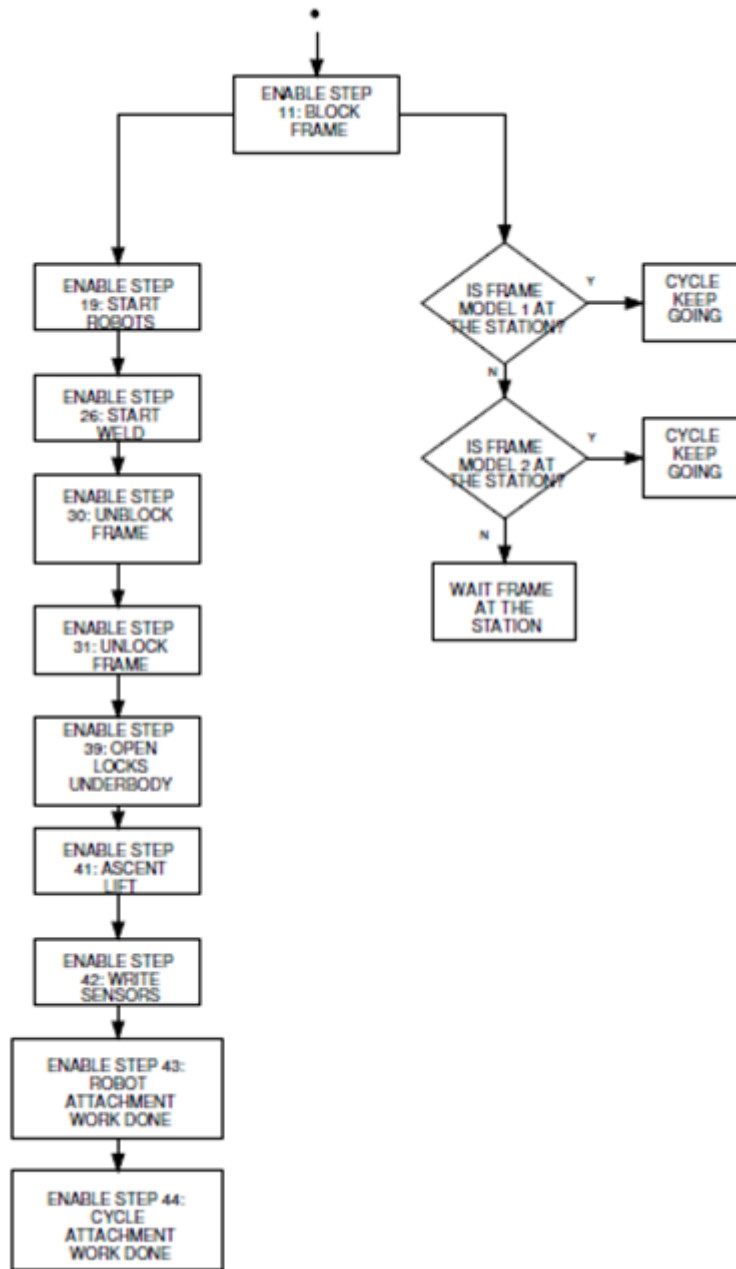
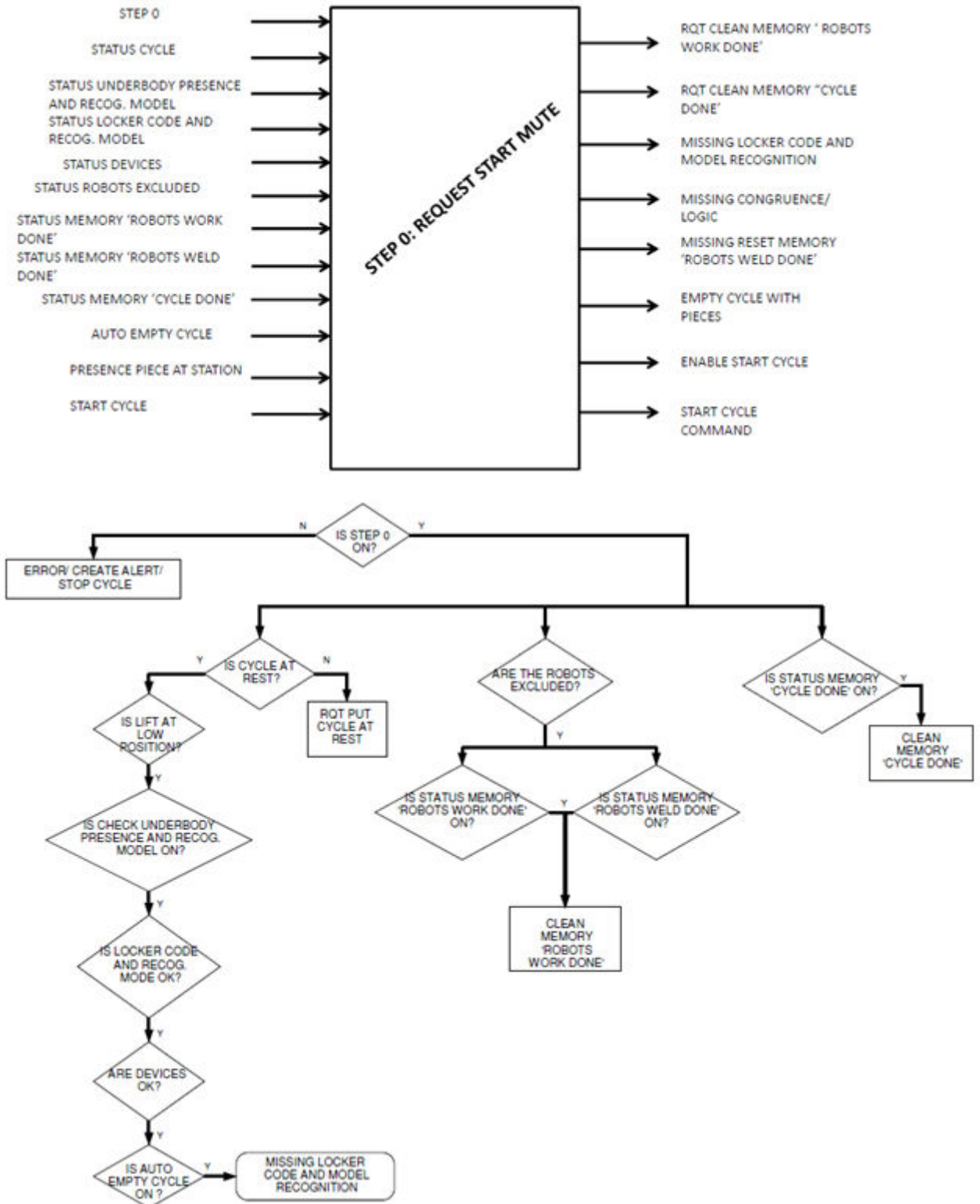


Figure 12: Function: Enable Operation Cycle.

Step 0: Request Start Mute



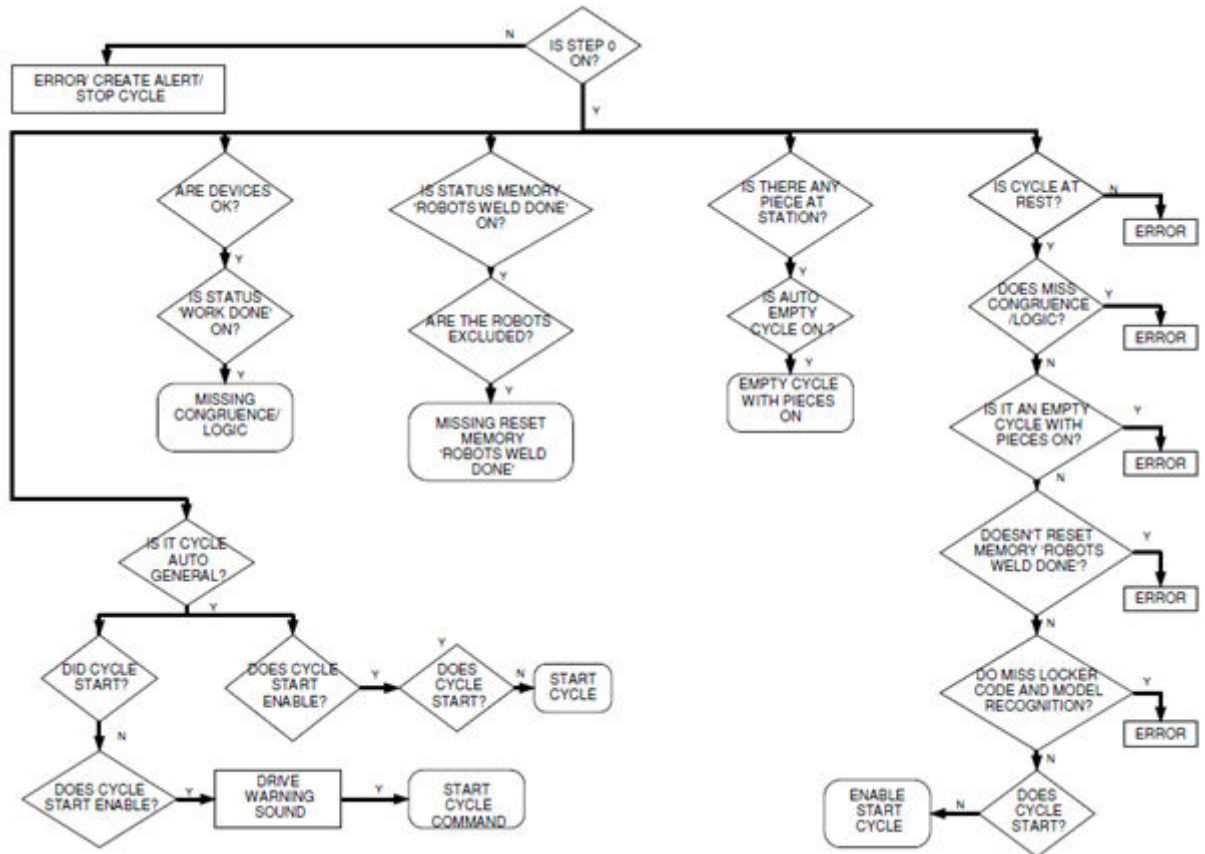
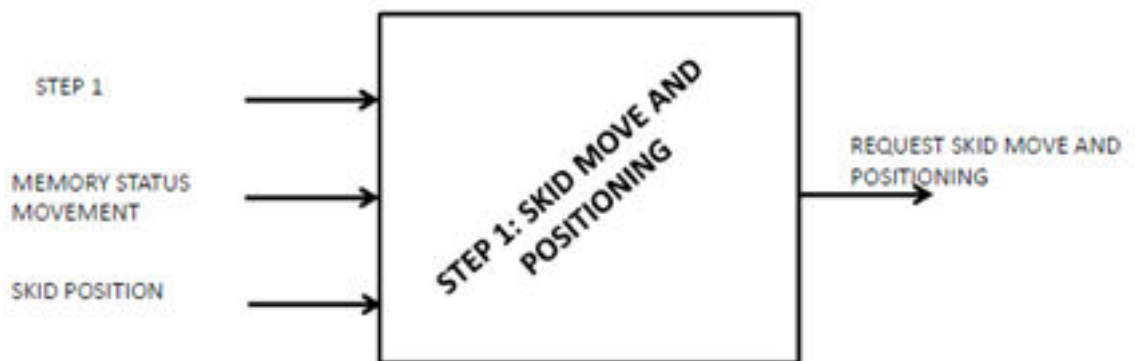


Figure 13: Function: Step 0 - Request Start Mute.

Step 1: Skid Move and Positioning



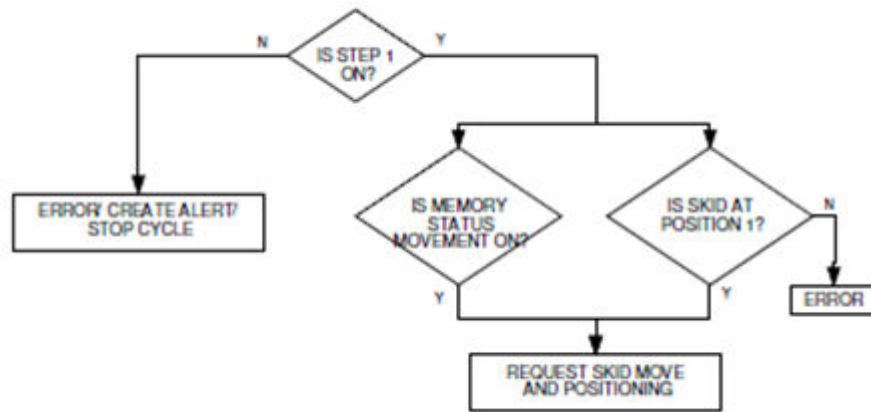


Figure 14: Function: Step 1 - Skid Move and Positioning.

Step 2: Sensors Reading

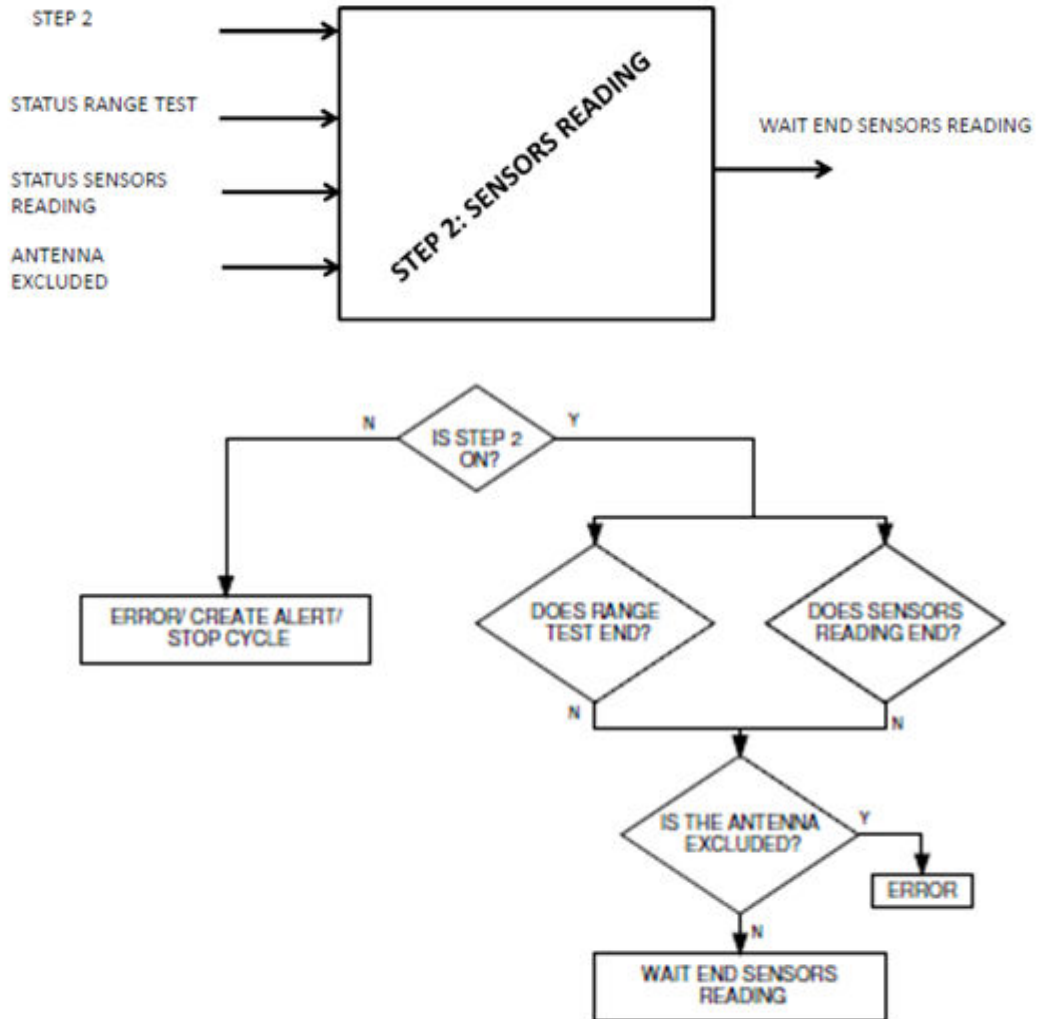


Figure 15: Function: Step 2 - Sensors Reading.

Step 3: Lift Descent

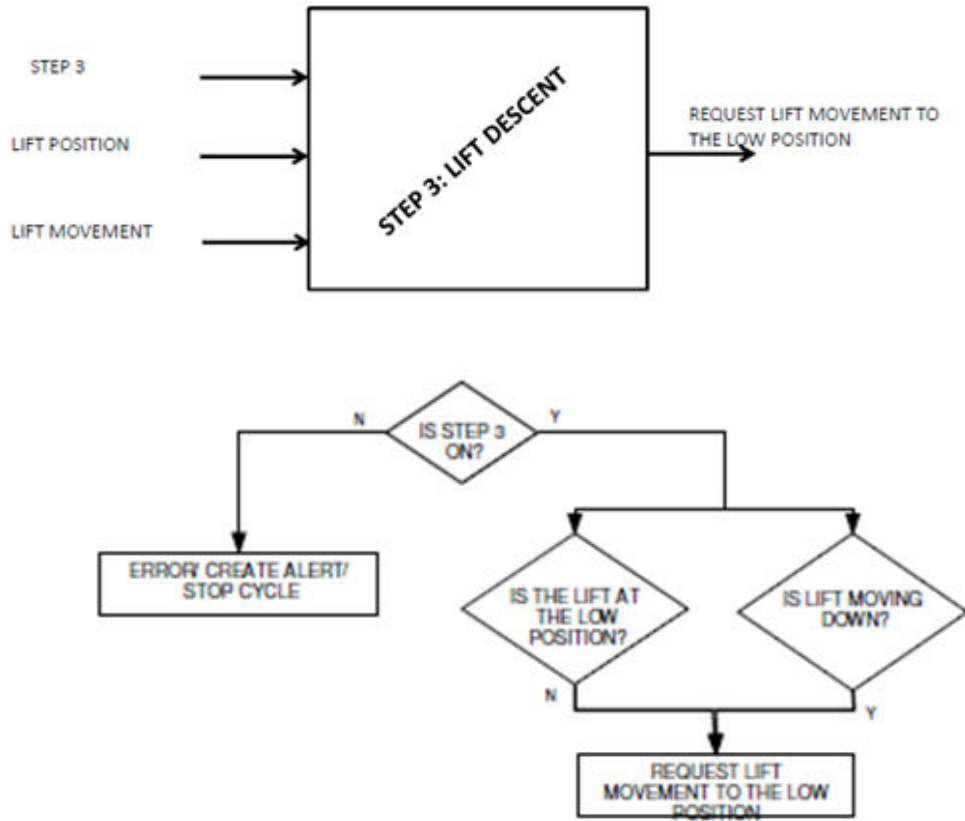
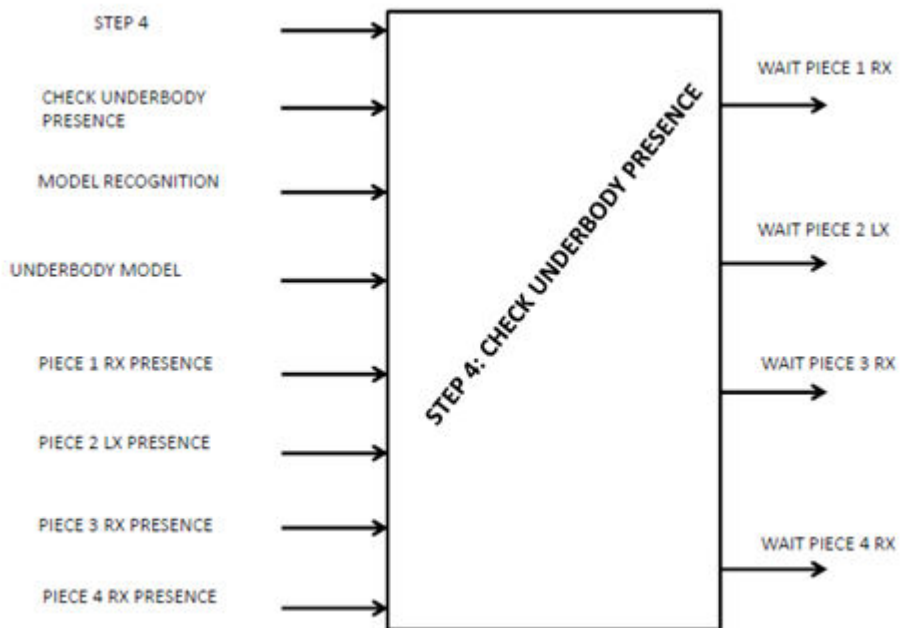


Figure 16: Function: Step 3 - Lift Descent.

Step 4: Check Underbody Presence



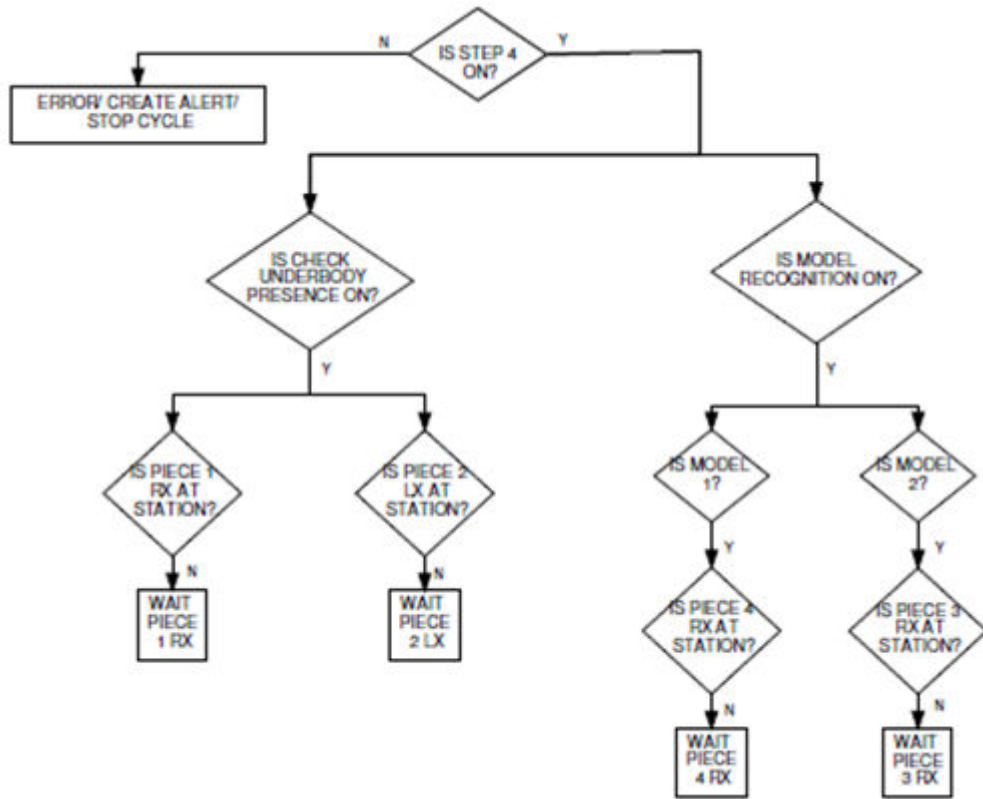
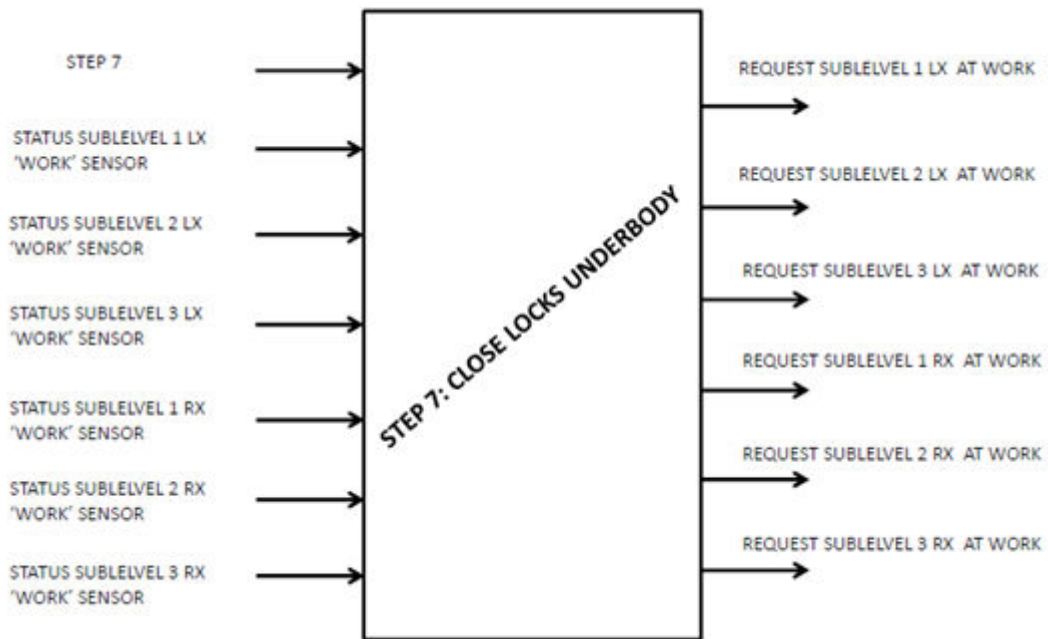


Figure 17: Function: Step 4 - Check Underbody Presence.

Step 7: Close Locks Underbody



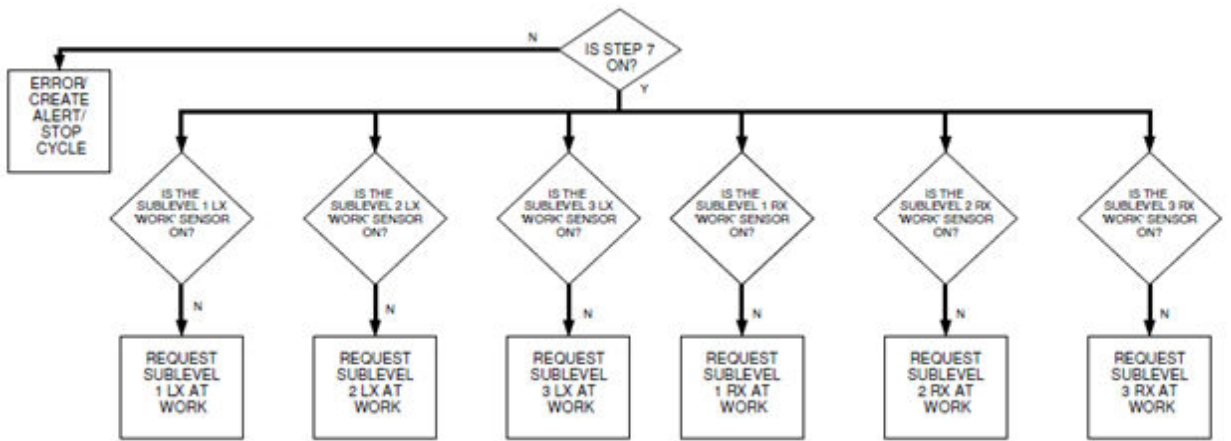


Figure 18: Function: Step 7 - Close Locks Underbody.

Step 8: Close Frame

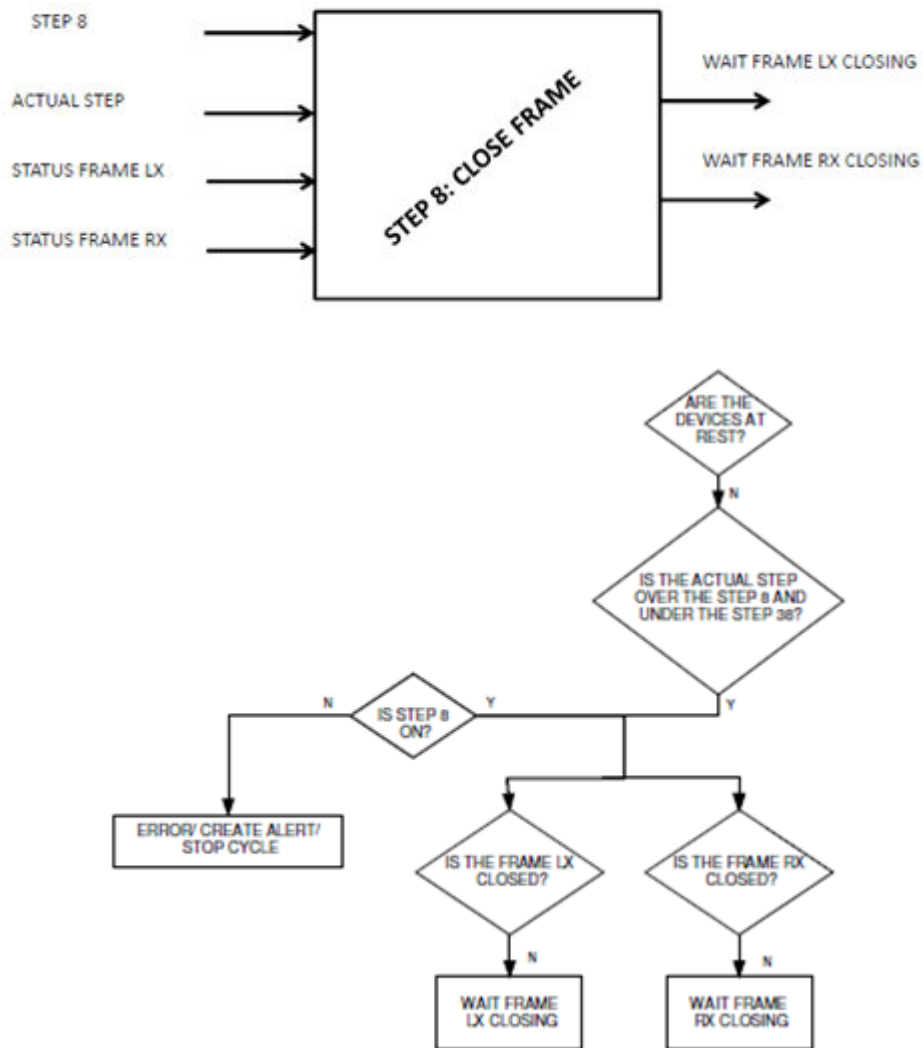


Figure 19: Function: Step 8 - Close Frame.

Step 9: Insert work supports frame

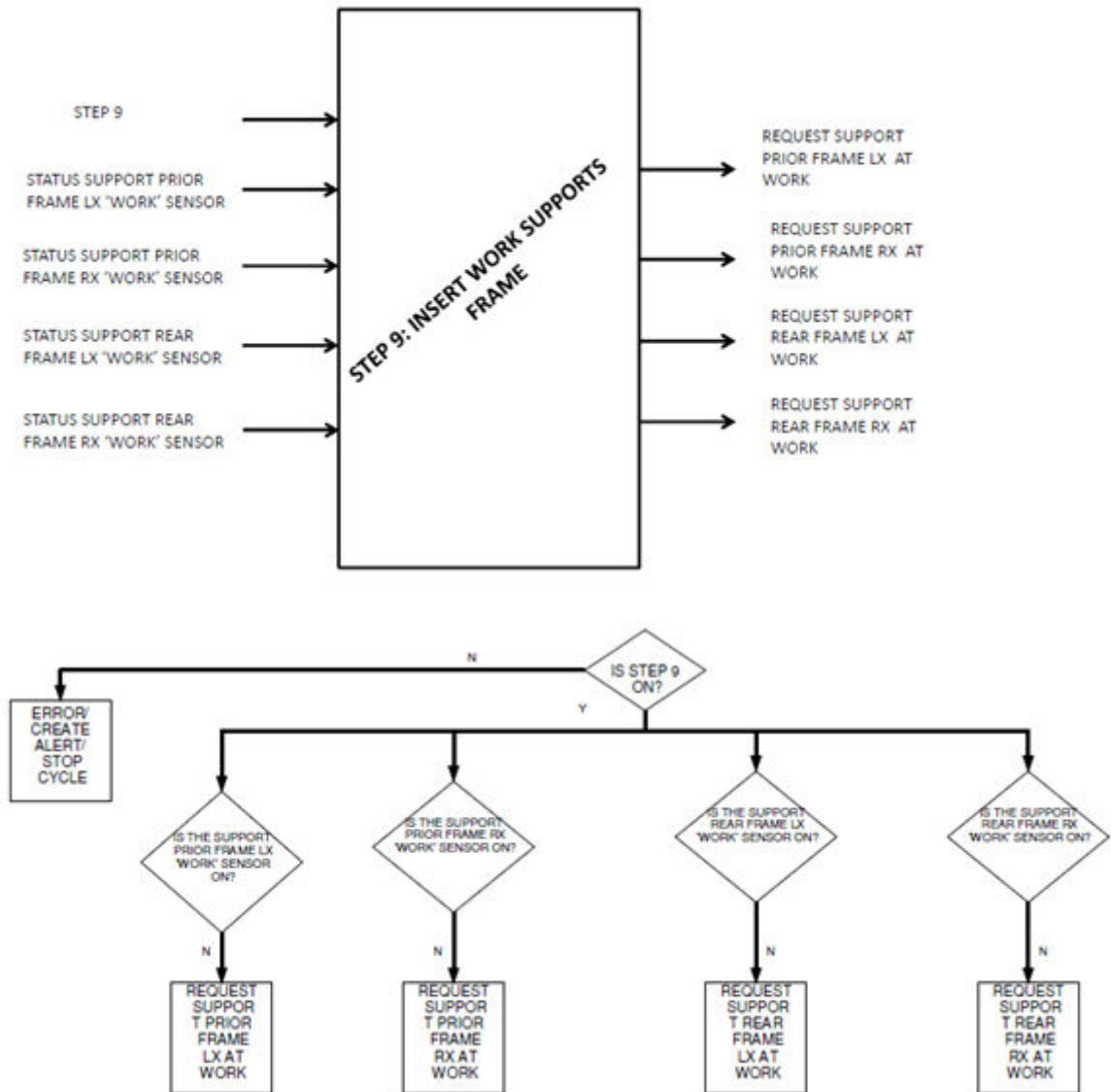


Figure 20: Function: Step 9 - Insert work supports frame.

Step 10: Lock Frame

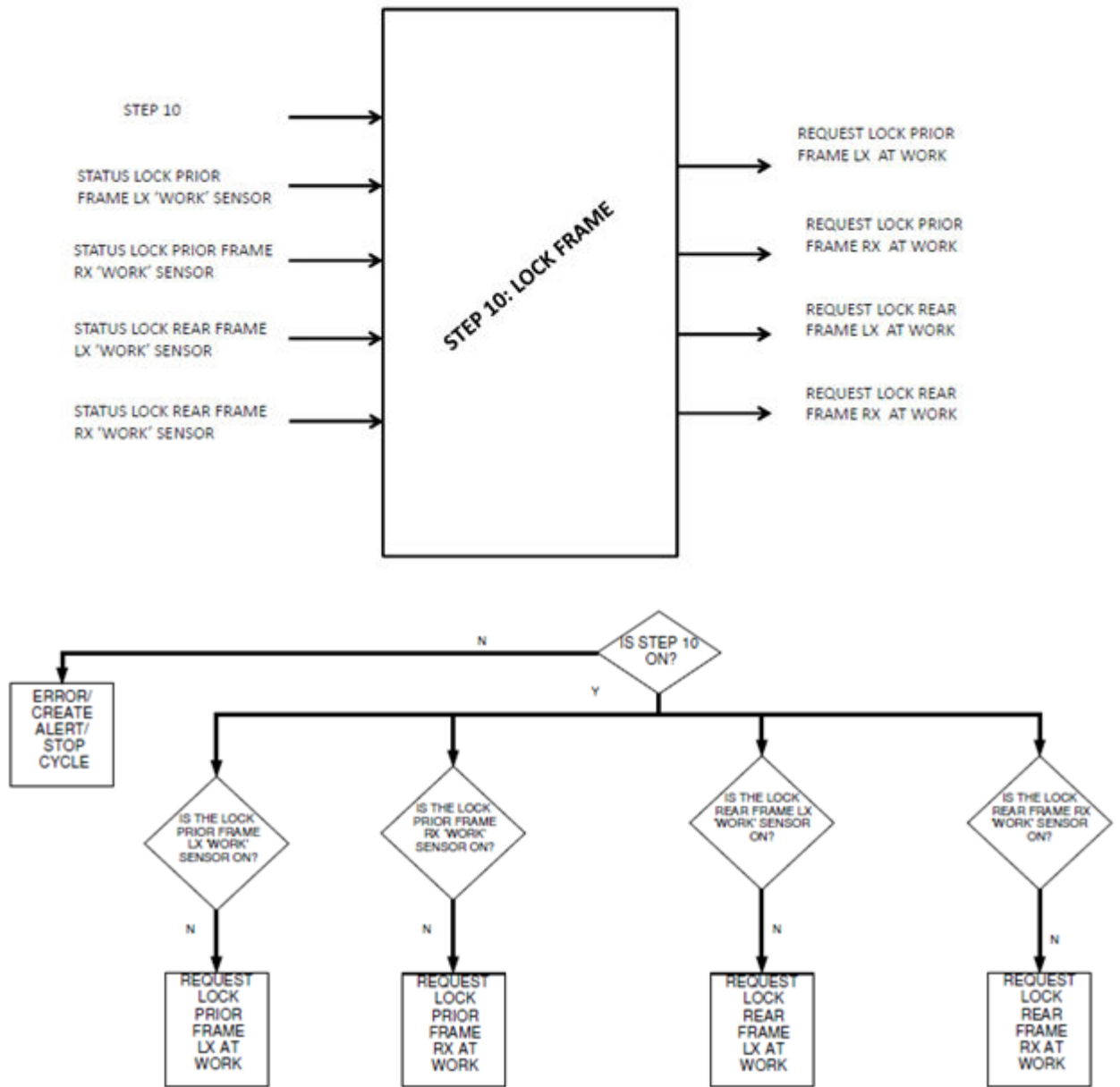


Figure 21: Function: Step 10 - Lock Frame.

Step 11: Block Frame

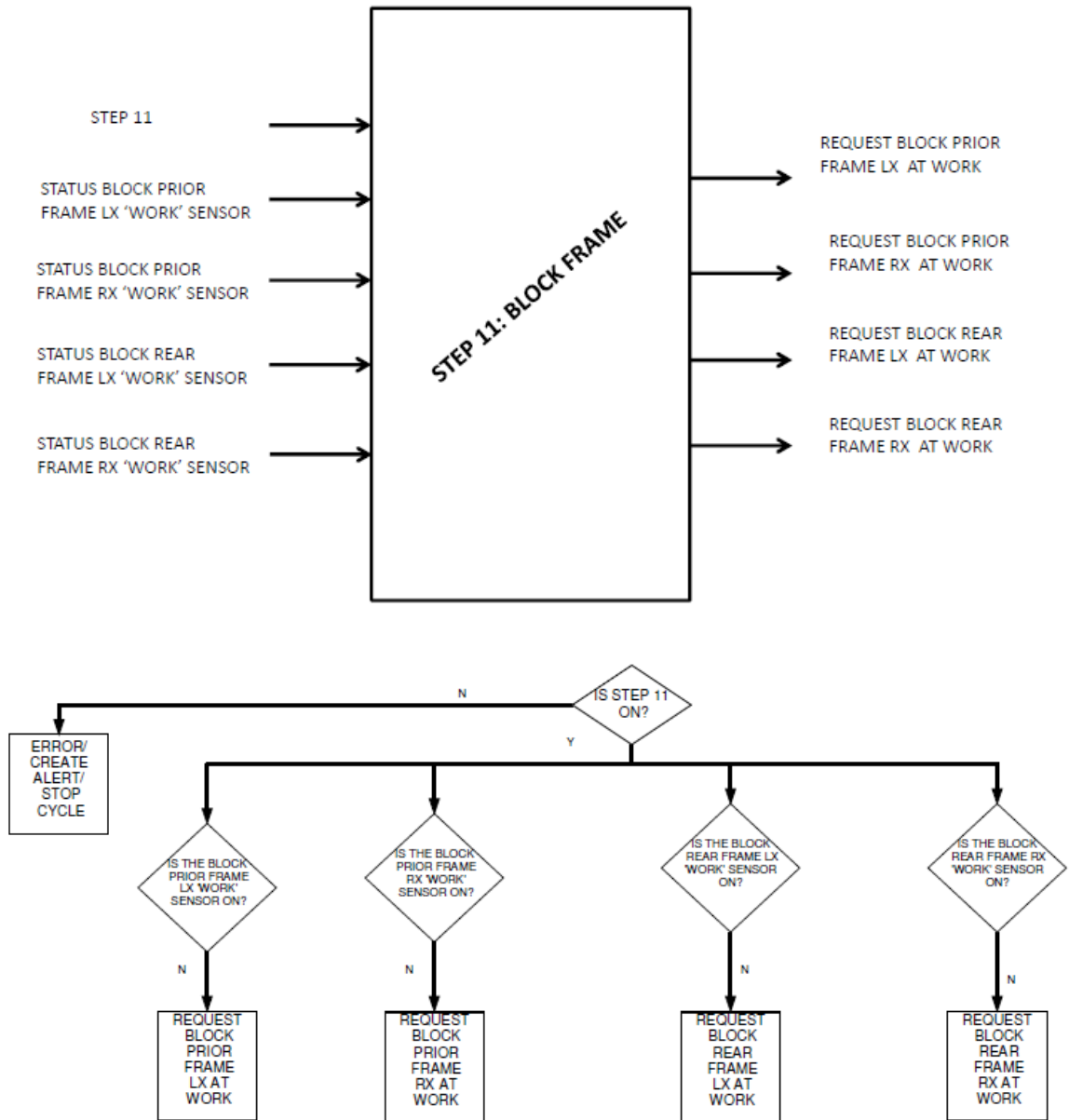


Figure 22: Function: Step 11 - Block Frame.

Step 19: Start Robot

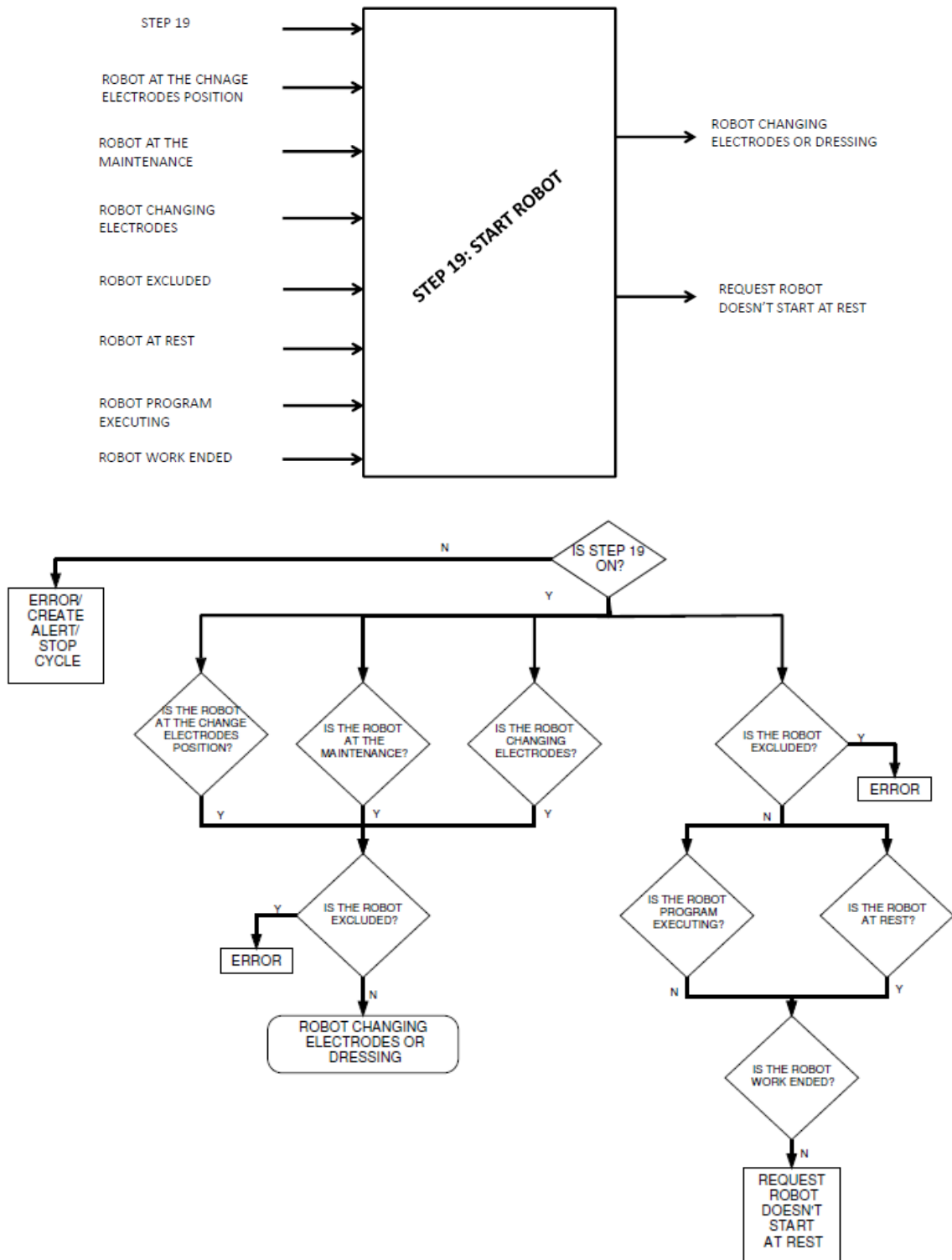


Figure 23: Function: Step 19 - Start Robot.

Step 26: Start Weld

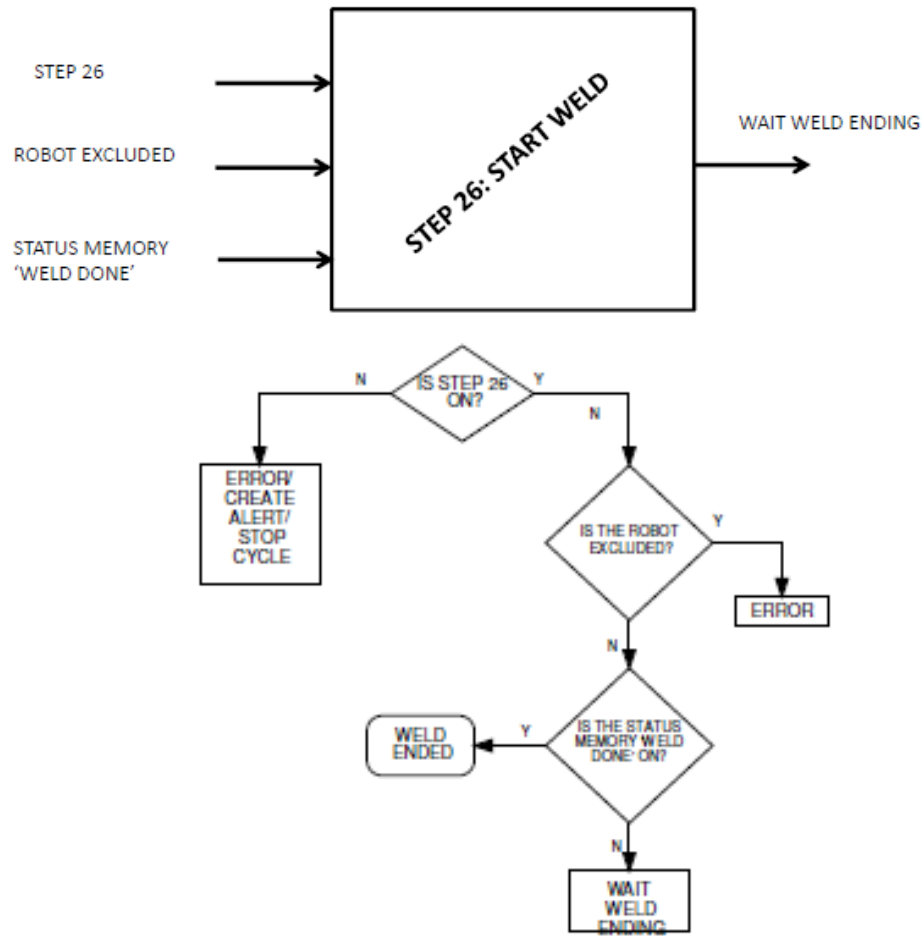
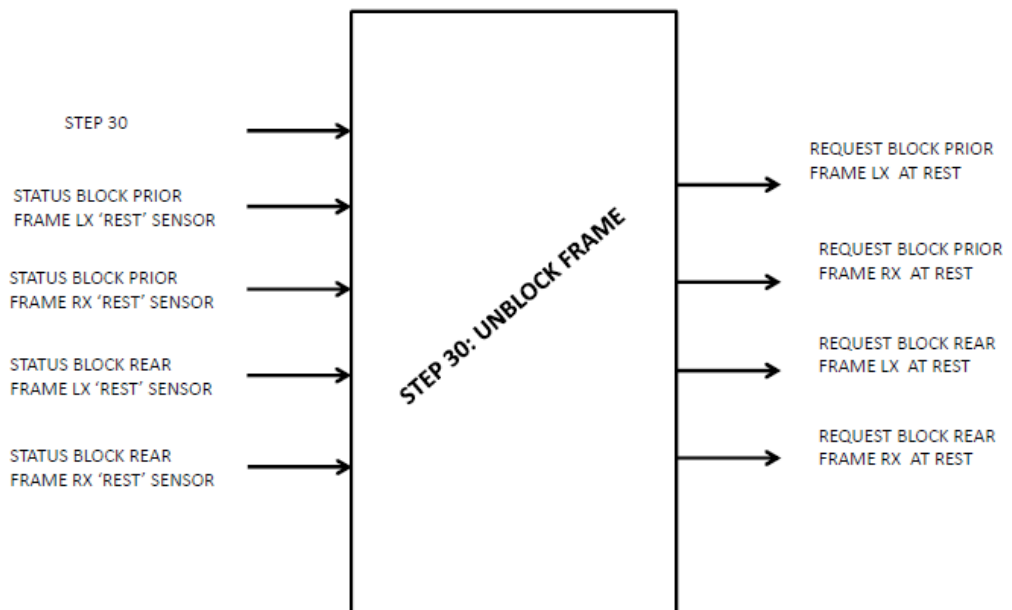


Figure 24: Function: Step 26 - Start Weld.

Step 30: Unblock Frame



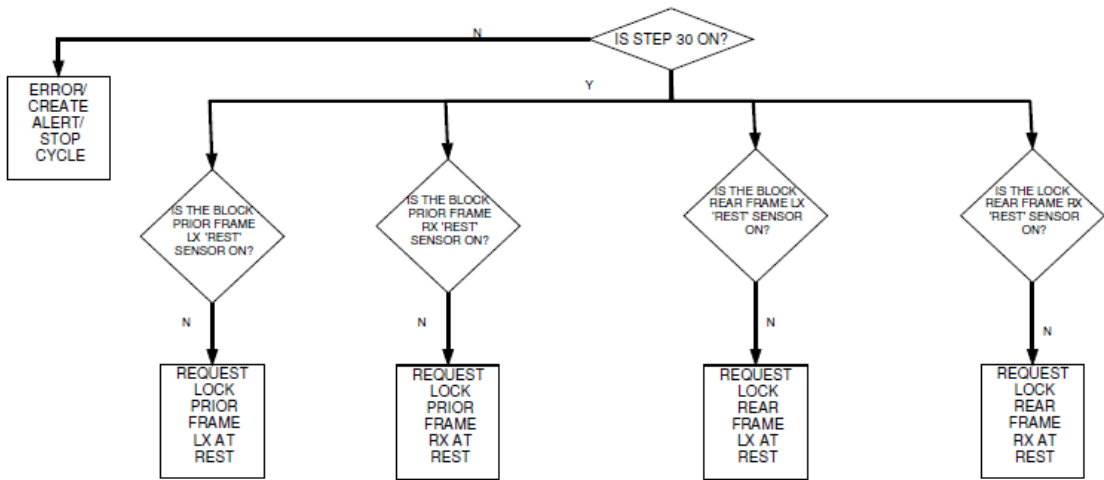


Figure 25: Function: Step 30 - Unblock Frame.

Step 31: Unlock Frame

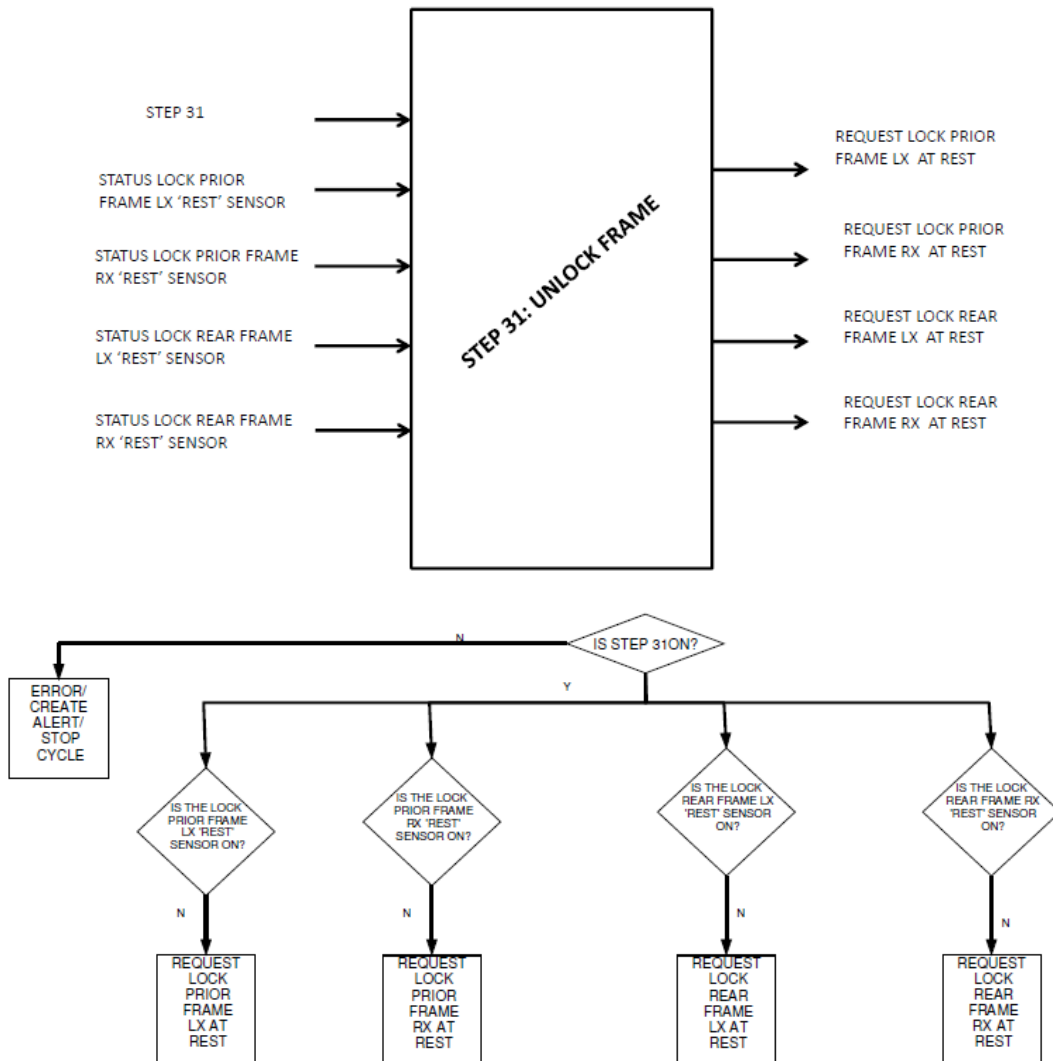


Figure 26: Function: Step 31 - Unlock Frame.

Step 39: Open Locks Underbody

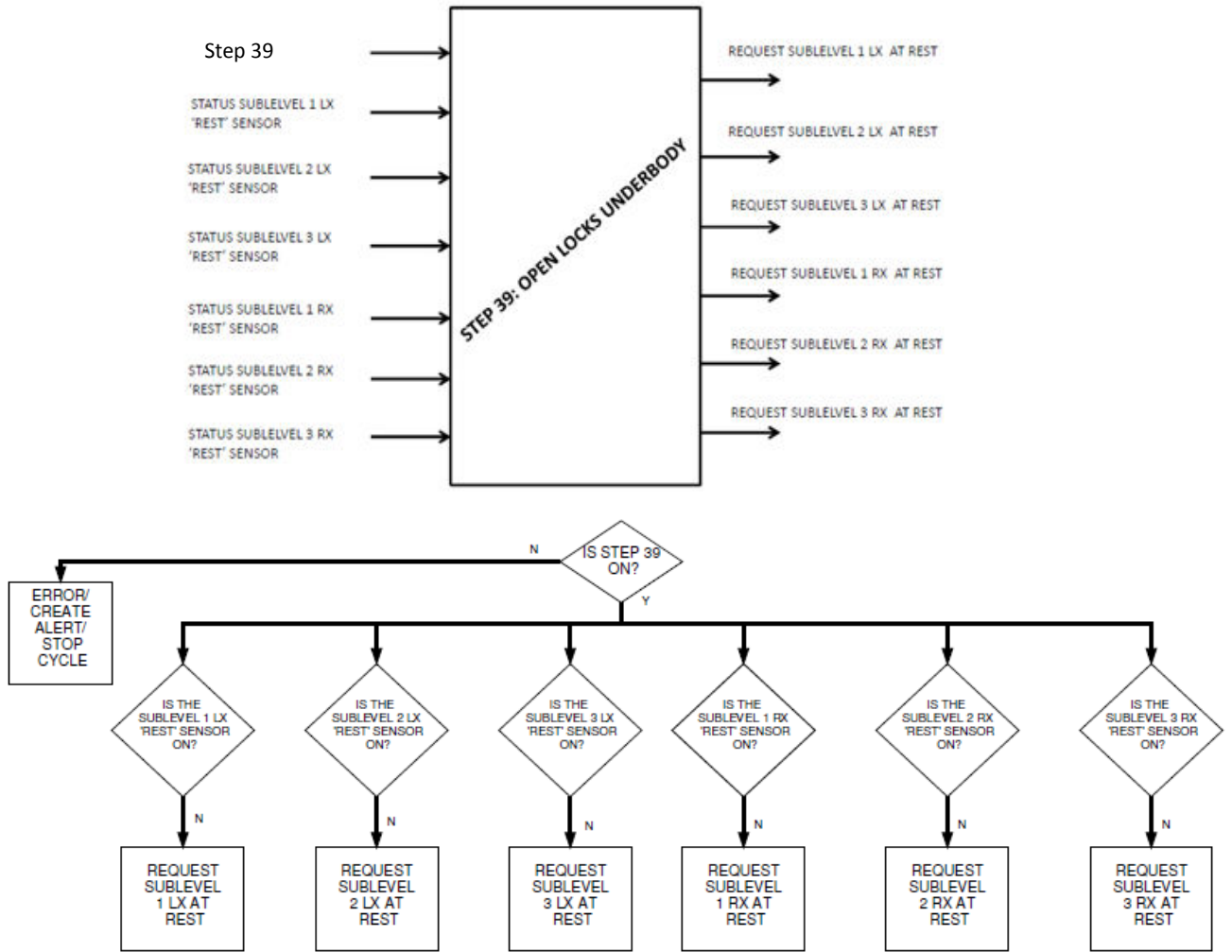


Figure 27: Function: Step 39 - Open Locks Underbody.

Step 41: Lift Ascent



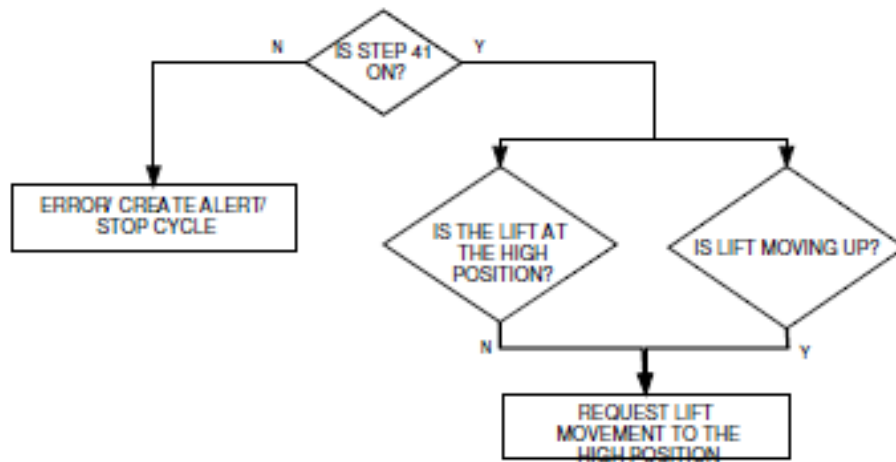


Figure 28: Function: Step 41 - Lift Ascent.

Step 42: Sensors Writing

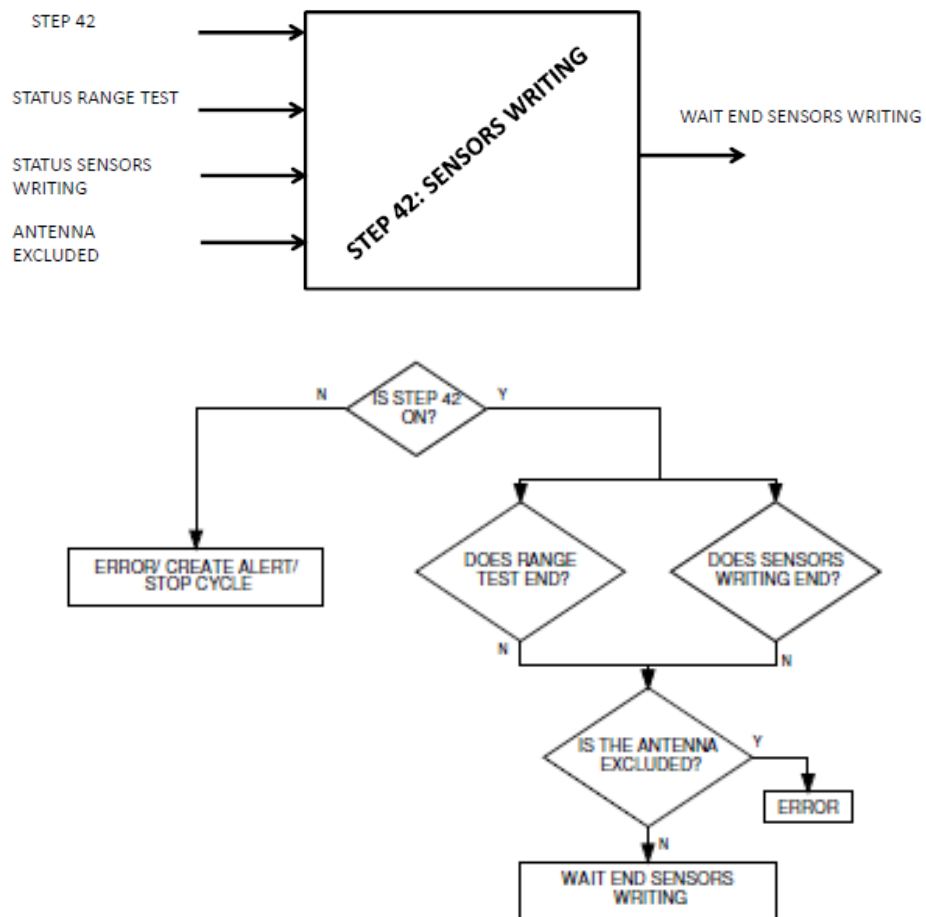


Figure 29: Function: Step 42 - Sensors Writing.

Step 43: Robot attachment work done

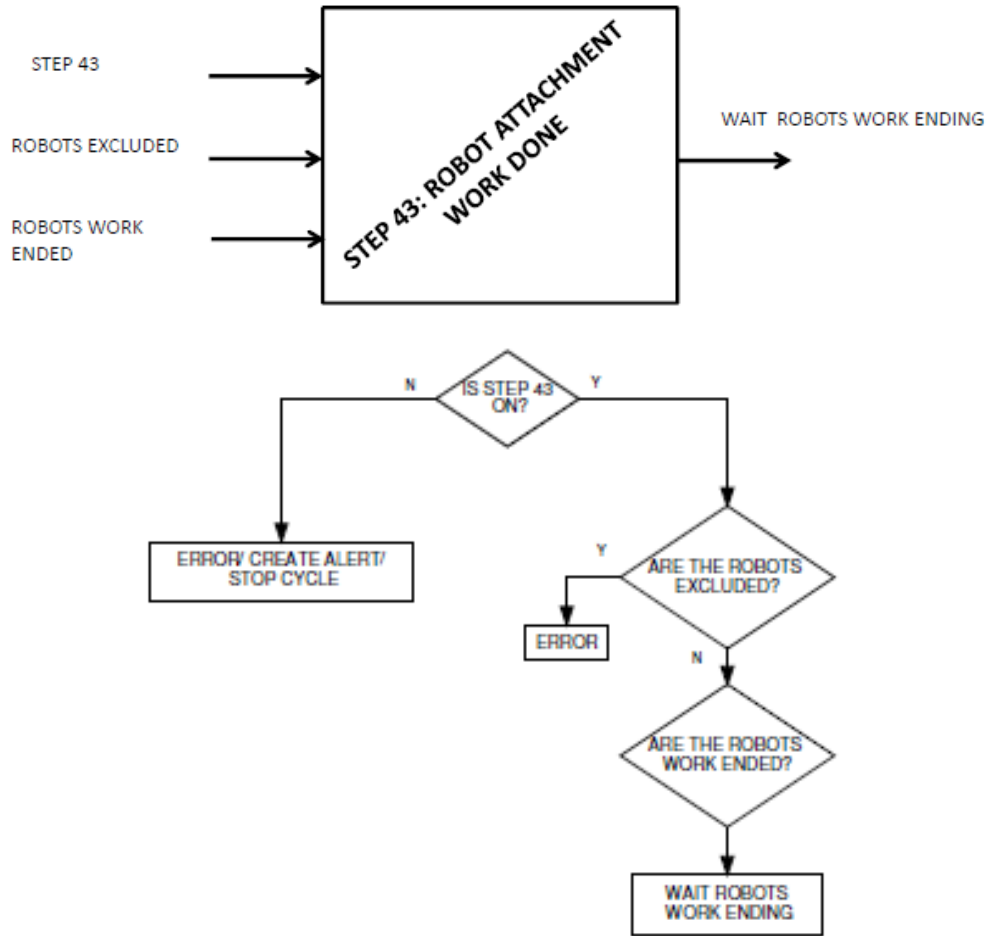
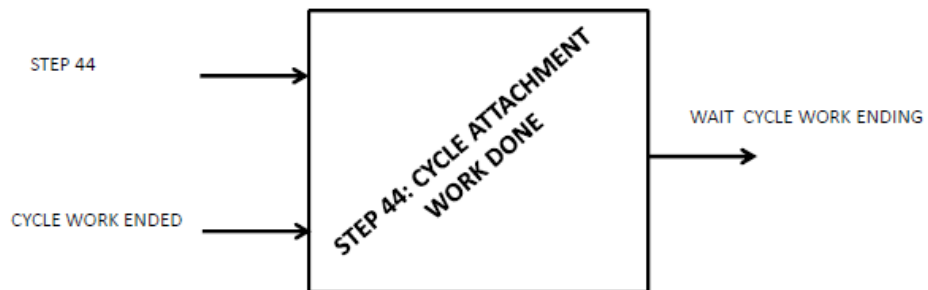


Figure 30: Function: Step 43 - Robot attachment work done.

Step 44: Cycle attachment work done



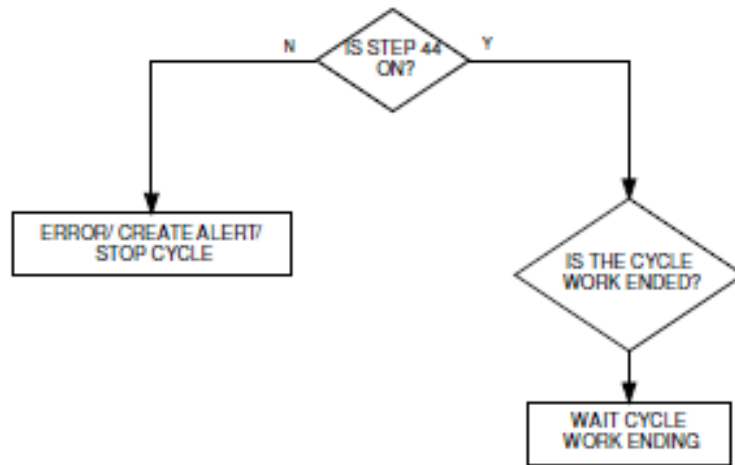


Figure 31: Function: Step 44 - Cycle attachment work done.

5. Manufacturing Demonstration

The aim of this demonstration is to test and show the first achievements of the project and how the BEMO-COFRA can be incorporated in a manufacturing scenario. The demonstration uses a reduced manufacturing model with the main devices.

5.1 Scenario

The scenario describes the structure of the system, which will be controlled, and how this system will behave during the demonstration. Basically, it is the system definition for the control logic development. In this case we are creating a system and not mapping an existing one.

The figure below shows how the devices are integrated in the demonstrator. The devices used are:

- Robot
- Robot Controllers
- PLC
- Conveyor belt:
 - Railway
 - RFID reader
 - Arduino
 - Arduino mega
- Server
- Plug wise
- Accelerometer
- Camera (iVision)

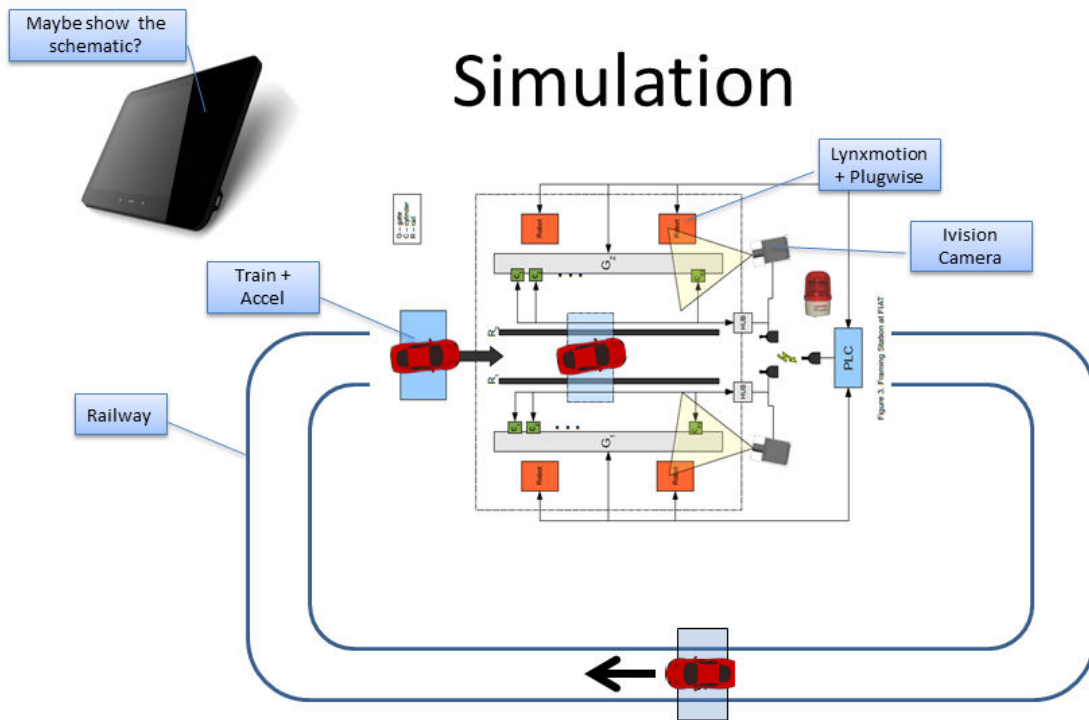


Figure 32: Demonstration Scenario.

Summarized, the skid (train) enters the station, the camera confirms its position, then the robots start the simulation of the welding activities, and when it finishes the skid leaves the station, coming back to the rest station, waiting for a new start command. This sequence of activities is shown in the next section 5.2 in a flowchart.

5.2 Flowchart

After understand how the system behaves, the follow step is to put it in a sequence of activities, inputs and outputs, which can be easily translate to a control logic for the PLC. The PLC reads a control logic program, normally in Ladder language, top down from left to right, and cycling. So to make easier the PLC correct running the system behaviour should be thinking in steps like a flowchart and the probable errors and failures should be inside this flowchart as well.

The following flowchart represents the sequence of activities on the demo scenario. This flowchart is a summary of the principal functions discussed in the previous chapter, section 4.3. It summarizes the framing station production flow control logic in a simple and reduced sequence of activities, which can be showcased with the demonstrator.

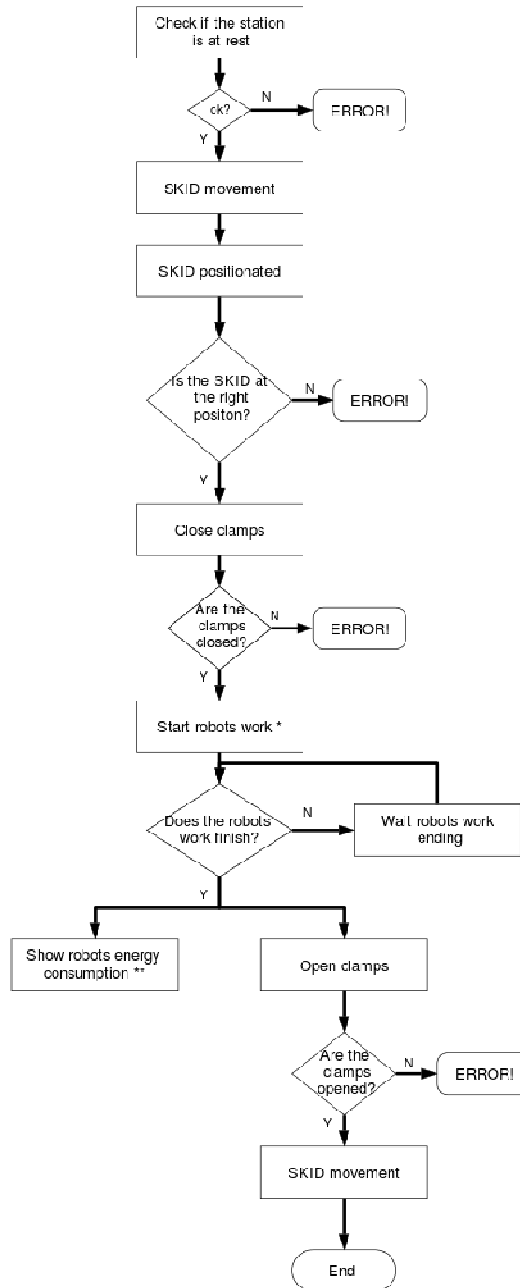


Figure 33: Demo Control Logic Flowchart.

5.3 Ladder Program

The Ladder program is the last step of implementing control logic. After the mapping of the production flow, the devices that will be monitored and/or controlled and the creation of the flowchart, the PLC is ready to receive the control logic program.

A common language for this program is the Ladder language, which is based in contacts normally open or closed. The inputs, sensors, are represented as contacts normally open or closed and the outputs, actuators, are represented as coils. When the inputs or outputs are connected in sequence is the same as an AND command, and when they are connect in parallel is the same as an OR command. At the Ladder language also is possible implement counters, timers, any special block function for your program.

The program can be described as the following steps:

1. Check if the station status is "at rest";
2. If the StartDemo is activated, but the station is not "at rest", Alarm1 is generated warning that the station is already working;
3. After the start with the station "at rest", the skid starts moving and station status changes to "at work";
4. A timer counts the necessary time to the skid getting to the welding position;
5. When the time is elapsed the skid stops;
6. Check if the skid is at the right position, if it is not at the right position Alarm2 is generated;
7. If the skid is at the right position and the clamps are open, send a signal to close the clamps;
8. If the skid is at the welding position and the clamps return a confirmation that they are closed, the body car is considered "fixed";
9. If the command to close the clamps is sent and the body car status is yet "released", Alarm3 is generated warning about a problem at the clamps;
10. With the body car fixed, a signal is sent to the robots to start their work;
11. Wait for the robots finishing their work and if the body car is fixed, send a signal to stop the robots;
12. After the robots finish their work and with the body car fixed and the robots stopped, the command is to open the clamps
13. With the command to open the clamps and the confirmation that they are opened, the body car status goes to "released";
14. If a command to open the clamps was sent and the body car status is yet "fixed", Alarm4 is generated;
15. If the body car is released, the robots have finished their work and, the skid starts moving to the "start position";
16. After the time elapsed for the skid reaching to the start position, a command is sent to stop the skid.

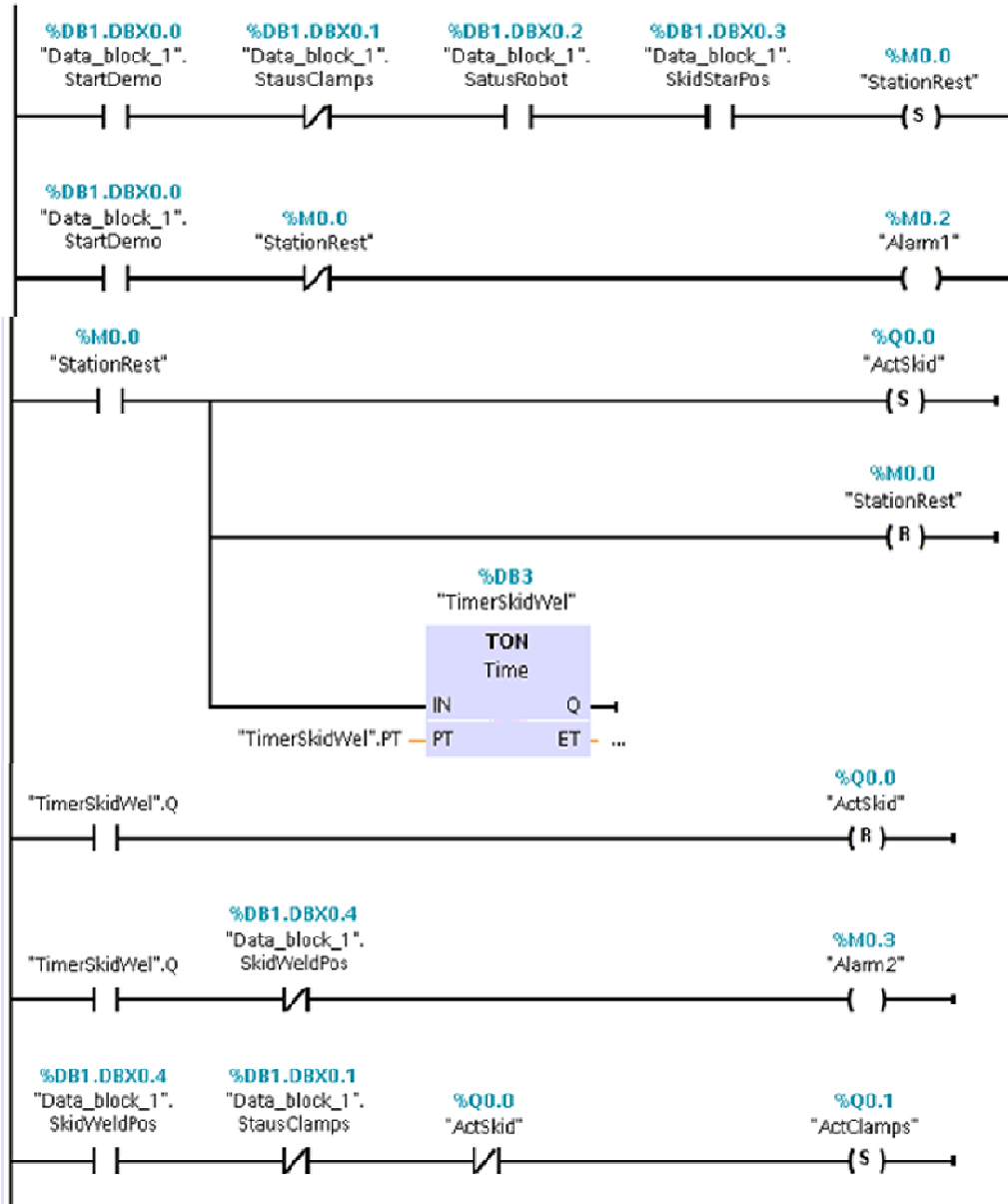


Figure 34: Demo Ladder Program - part. 1.

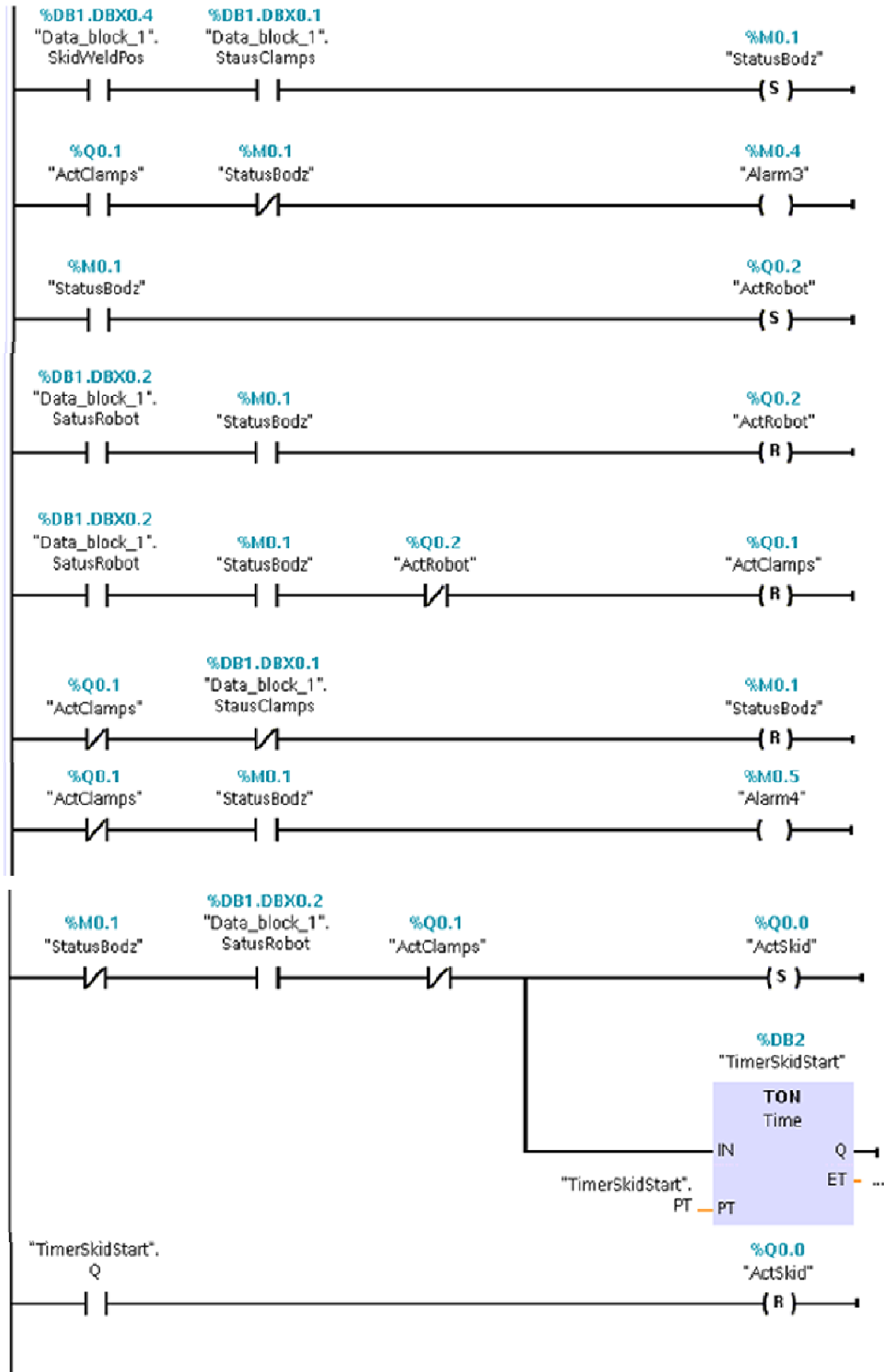


Figure 35: Demo Ladder program - part. 2.

Following an explanation about the program variables, and their role in the control logic:

The Actuators change the system depending on their values.

- ActSkid: controls the movement of the skid
 - True: the skid starts (and continues) moving
 - False: the skid stops moving
- ActClamp: controls the movement of the clamps
 - True: The clamps are locked (thus, system is ready for welding)
 - False: The clamps are opened
- ActRobot: controls the movement of the welder
 - True: the robot does the welding
 - False: the robot stays idle (no welding)

The next group of variables is composed by inputs, basically sensors:

- StartDemo: the input that will start the operating cycle.
- SkidStartPos: a sensor that determines whether the Skid is in the start position (true) or not (false)
- SkidWeldPos: a sensor that determines whether the Skid is in the welding position (true) or not (false)
- StatusClamp: a confirmation variable (possibly a sensor) that informs about the status of the clamps.
 - True: the clamps are properly locked
 - False: the clamps are not properly locked (default)
- StatusRobot: a confirmation variable (possibly a sensor) that informs about the status of the robot/welder.
 - True: robot has finished the welding process (and thus, should be stopped)
 - False: robot has not finished the welding process (and thus, shouldn't be interrupted)

The follow variables are internal of the program, they are auxiliary variables;

- StationRest: a verification variable. When input "StartDemo" is activated (it becomes true), then "StatusClamps" must be false, "StatusRobots" must be true and "SkidStartPos" must be true. When all the conditions are met, the value of this variable becomes true.
- StatusBodz: verification variable that the system is ready for welding. When "SkidWeldPos" is true and "StatusClamps" is true (i.e. skid is in the welding position and clamps are properly locked), then the value of this variable will be true.
- Alarm1: when input "StartDemo" is activated (it becomes true), but "StationRest" is still false (i.e. "StartDemo" is activated when the system is not in rest condition), then alarm1 will be raised.

- Alarm2: after the skid start moving from the initial position, we use a timer to determine how long it must keep moving. Once the timer is up, the skid will stop. If the timer is already up, but the value of the variable "SkidWeldPos" is still false (i.e. the the skid already stops before reaching the correct position), then alarm2 will be raised.
- Alarm3: when clamps are locked but "StatusBodz" is still false (i.e. clamps are already locked while system is not ready for the welding process), then alarm3 will be raised.
- Alarm4: when clamps have already been opened but "StatusBodz" is still true (i.e. clamps are already opened while the system is still expecting for the welding process), then alarm4 will be raised.

6. References

(Comau, 2012) Body & Welding knowledge

(Wikipedia, 2012) <http://en.wikipedia.org/wiki/Poka-yoke>

(isixsigma, 2012) <http://www.isixsigma.com/dictionary/poka-yoke/>