Specialty Integration Experiment

Design Report

Student Name : Josephine Oduatyo Eyituoyo Odusanya

Student ID : 9200353007

Major: Robotics Engineering

Grade: 2020

Teacher : Jin Zhe

Date : July 15, 2023

Automation and Electrical Engineering School

Robotic Simulation Workstation Design Based on RobotStudio

Abstract

The purpose of this paper is to document the steps taken to create the simulation of three ABB robots while highlighting information about industrial robots and its uses. Industrial robots are becoming commonplace in various industries such as manuafacturing, aerospace and healthcare. It is important to continue development and research in this field thus urshuring innovation into society. The ABB companys robot studio software is where our virtual simulation will take place. It is important to note the software can also be used for offline maual programming.

Keywords: robot; ABB; RobotStudio; simulation;

1. Introduction

The use of industrial robots is on the rise, ranging from the conventional to the unconventional. In the dynamic landscape of modern manufacturing, the integration of advanced technologies has welcomed a new era of efficiency, precision, and adaptability. This transformation is lead by the fusion of intelligent automation and intricate design has redefined the possibilities of production processes. We can see the world is entering a new era, innorvation being its foreword. Industrial robots, according to the ISO, is an "automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or fixed to a mobile platform for use in automation applications in an industrial environment." The purpose of this report is to document the design and implementation of not one but three ABB industrial robots, i.e. :

Figure 1 IRB 1410

Figure 2 IRB 2600

Figure 3 IRB 1600

These days, we find that industrial robots are becoming commonplace in various industries such as manuafacturing, aerospace and healthcare. It is important to continue development and research in this field thus urshuring innovation into society. This innovation can be found in industries such as logistics, inspection and maintenance, and manufacturing.

Examples of industrial automation applications [7]:

- Arc welding
- Spot welding
- Materials handling
- Machine tending
- Painting
- Picking, packing, palletizing
- Assembly
- Mechanical cutting, grinding, deburring and polishing
- Gluing, adhesive sealing and spraying materials

Figure 4 Automated Assembly Plant

Thankfully, the above applications of industrial robots can first be achieved using robotics softwares and although many industrial robotics platforms exist, ABB RobotStudio remains the most popular with its easy-to-understand interface and powerful simulation capabilities. This may also be due to ABB industrial robot manufacturers extesnive experience in all things industrial robots [3]. RobotStudio is a PC application for modelling, offline programming, and simulation of robot cells [1]. It gives the much needed benefit such as reducing the risk of any damage to the robot during physical implementation.

With this software, we are introduced to RAPID Programming, a high-level programming language used to program ABB Industrial Robots. This Python and C-style language makes the code implementation easy to do and understand. A few notable in-built functions allow the robot to move to pre-set locations [4]. For example:

-MoveL (moves the robot up and down)

-MoveJ (moves the joints of the robot above the predefined target) [1]

This research paper delves into the relationship between industrial robots, the ABB RobotStudio environment, and programming. We will explore the platform and the design assignment given. It is with this analysis and experimentation, we aim to unearth insights that not only enrich academic discourse but also potentially offer tangible guidance to industries seeking to maximize the potential of robotic systems.

2. Design Assignment

Design a robotic simulation workstation in RobotStudio. There are several rough casts (more than three) in the station.

Basic task:

Step 1. Robot A places the rough casts on the workbench.

Step 2. Robot B finishes the machining process.

Step 3. Robot A takes the finished products from the workbench and transports them to the conveyor.

Optional task:

Step 4. Robot C takes the finished products from the end of the conveyor and stacks them up.

3. Simulation Procedure

Our design is made up of different feature that are necessary to accomplish the design assignment. To explain the procedure, we will go through each system, the robots needed to run the system and the other components that work together to ake up the station logic.

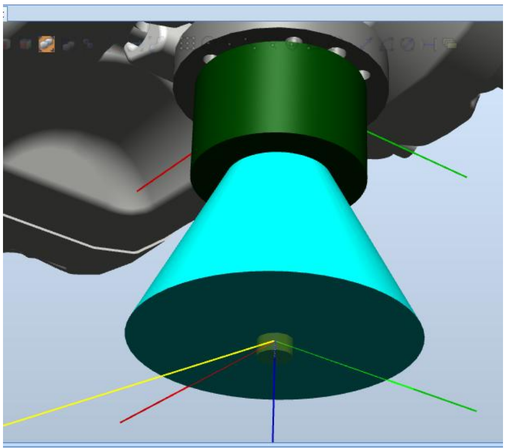

Figure 5 Tool Sucker

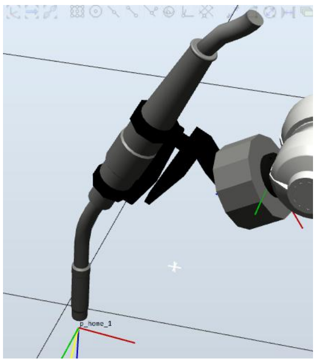

Figure 6 AW GUN

3.1 Workstation

System 1:

The purpose of this system is to control robot 1 to paint the rough cast red. This system is triggered once a rough cast is dropped on this workbench 1. It does this using the AW-Gun attachment. The targets were set at the four corners on the rough cast and a path was created.

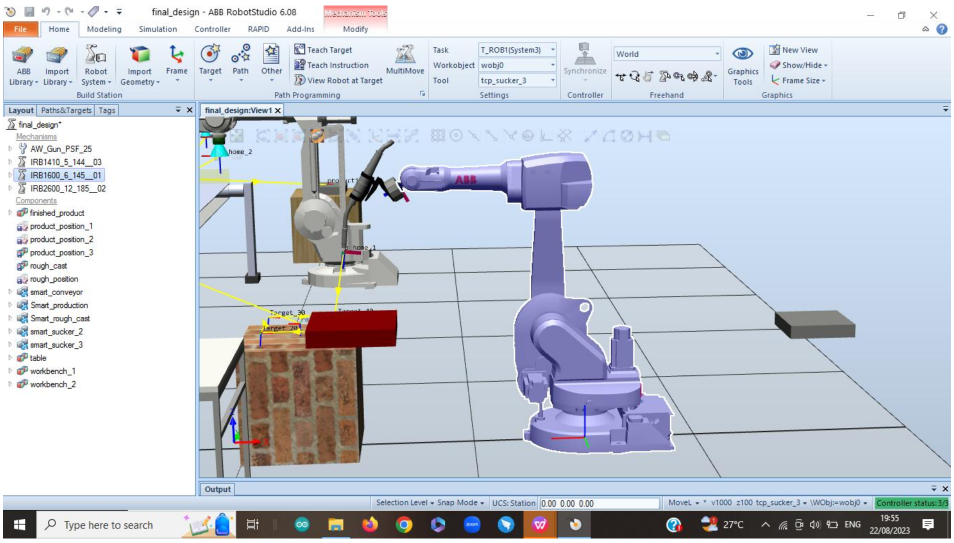

Figure 7 Robot 1 (IRB 1600)

A smart component is used in order to transform the rough cast (grey cast) into the finished product (red cast). We label it *smart_production* and set the parameters.

Figure 8 Smart_production components list

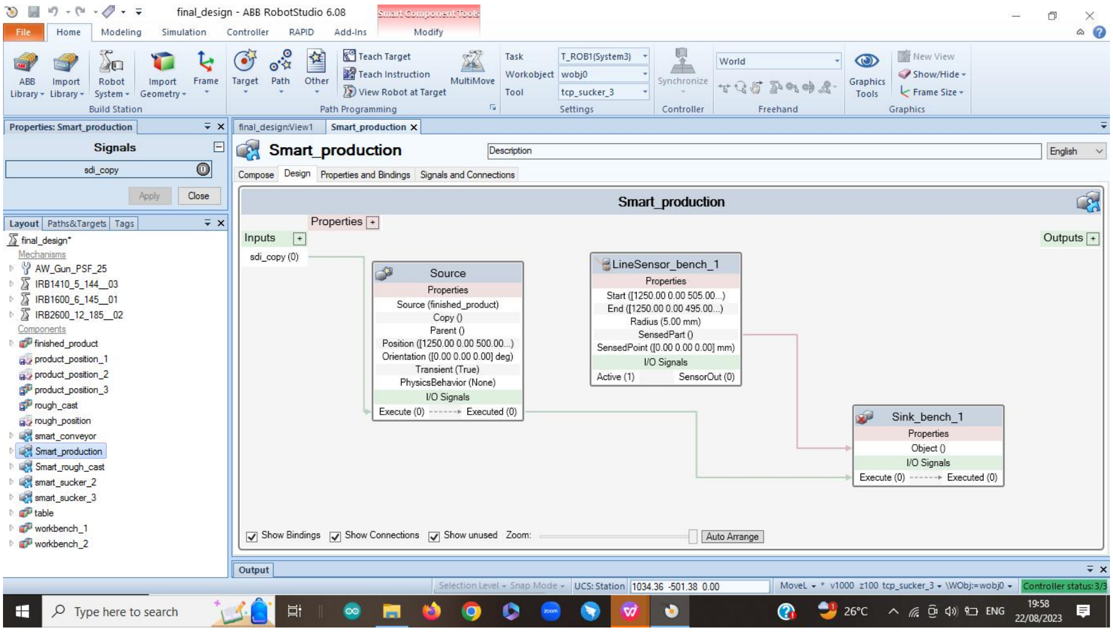

Figure 9 Smart_production logic design

System 2:

The purpose of this system control robot 2 to pick up a *rough_cast* in succession and place it on *workbench_1*. This system is what starts this whole simulation. It does this using the *tcp_sucker_2* attachment that was created. Once *system 1* is triggered and creates the *finished_product*, it is picked up by robot 2 and placed on the conveyor.

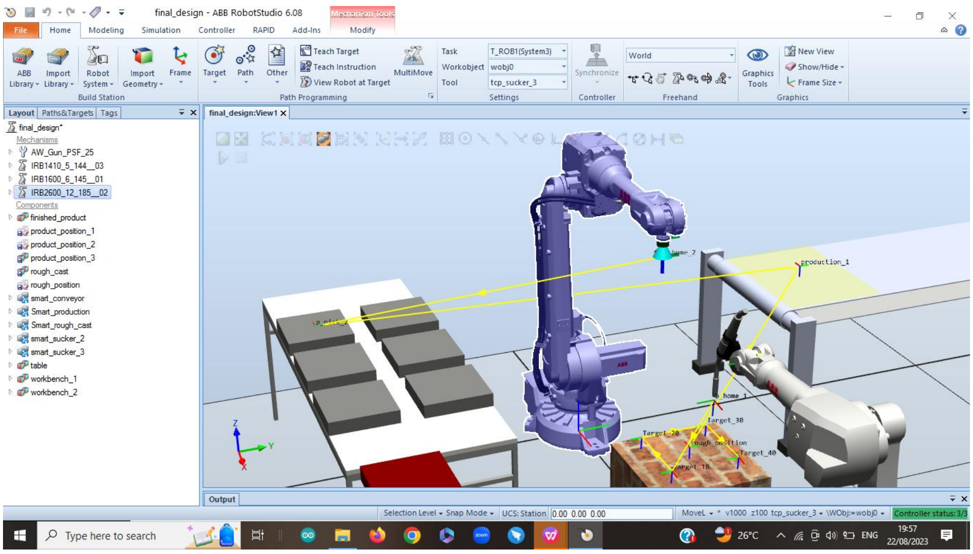

Figure 10 Robot 2 (IRB 2600)

Three smart components are made use of in this system.

First we label our first component *smart_rough_cast*, which simply duplicates the *rough_cast* and positions it on the table.

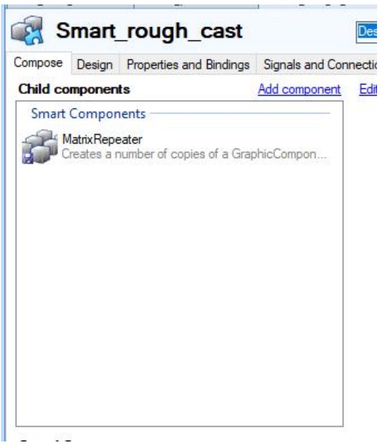

Figure 11 Smart_rough_cast Components list

The *MatrixRepeate*r is the only component meeded where it is set to 3x2 duplication.

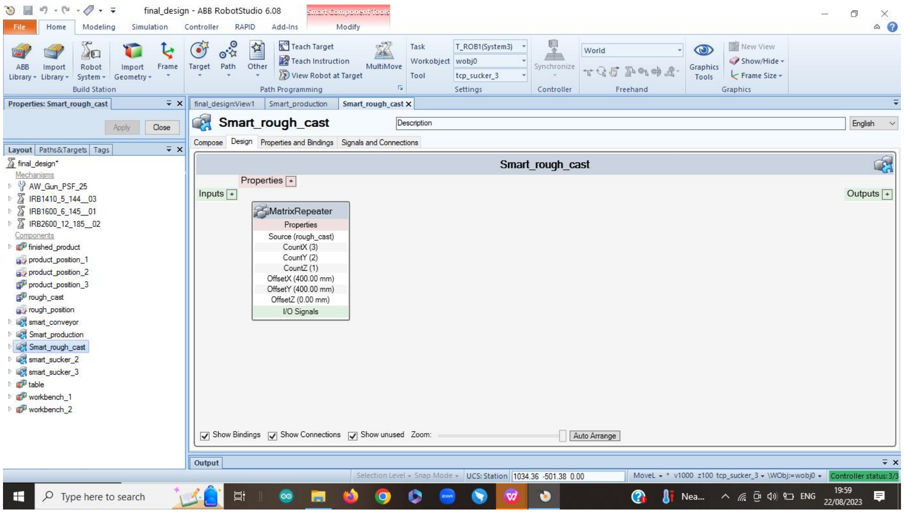

Figure 12 Smart_rough_cast logic design

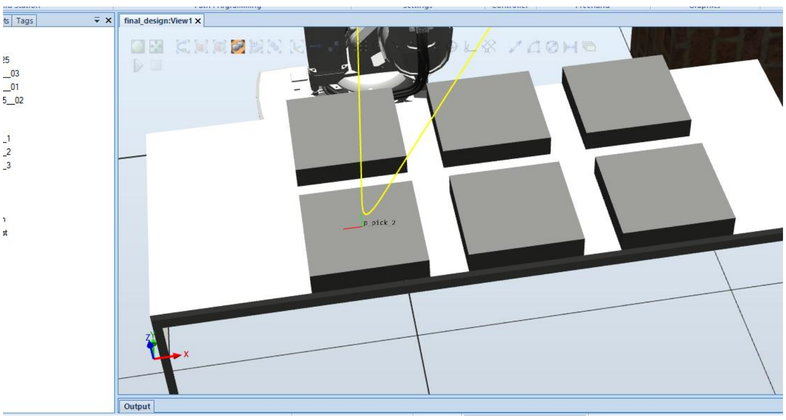

Figure 13 Smart_rough_cast

Next, we label *smart_sucker_2*. This attachment is needed to move the rough casts onto workbench one for processing.

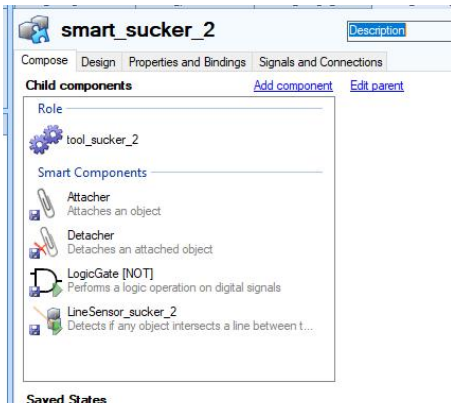

Figure 14 Smart_sucker_2 Components List

We ensure the *tool_sucker_2* is set as the role of this component. The set the following parameters which makes up the following working design.

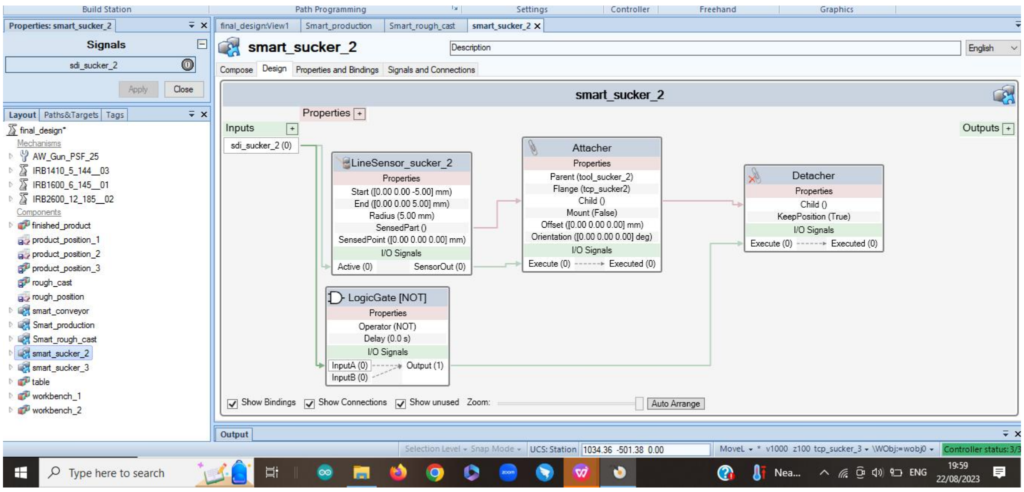

Figure 15 Smart_sucker_2 logic design

Smart_Conveyor: works by moving the finished product from one end to the other end to be picked up by robot 3 and stacked on workbench_2.

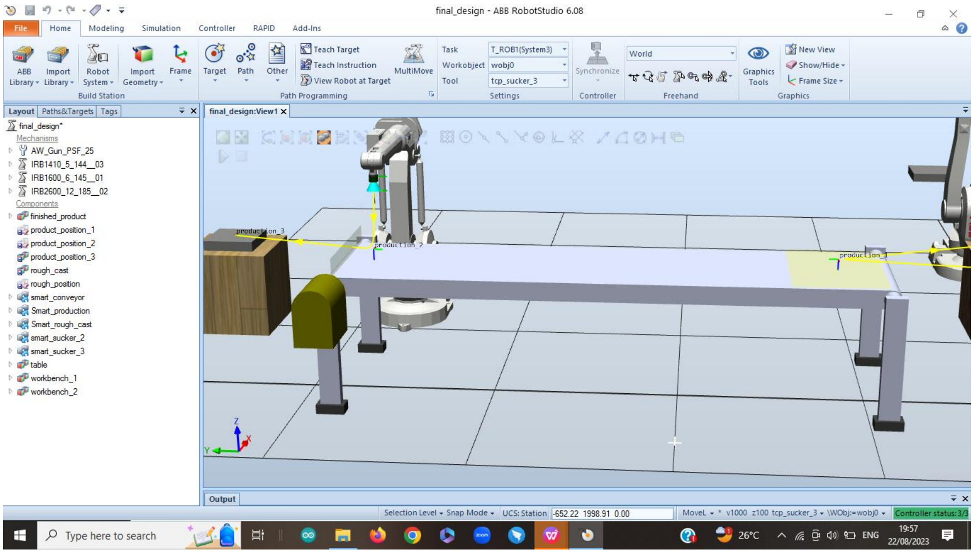

Figure 16 FlatBelt_ServoConveyor

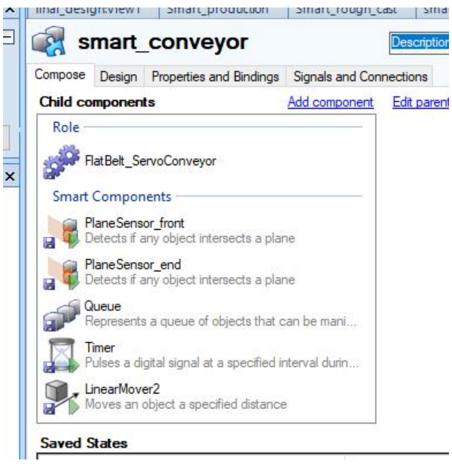

Figure 17 Smart_conveyor components list

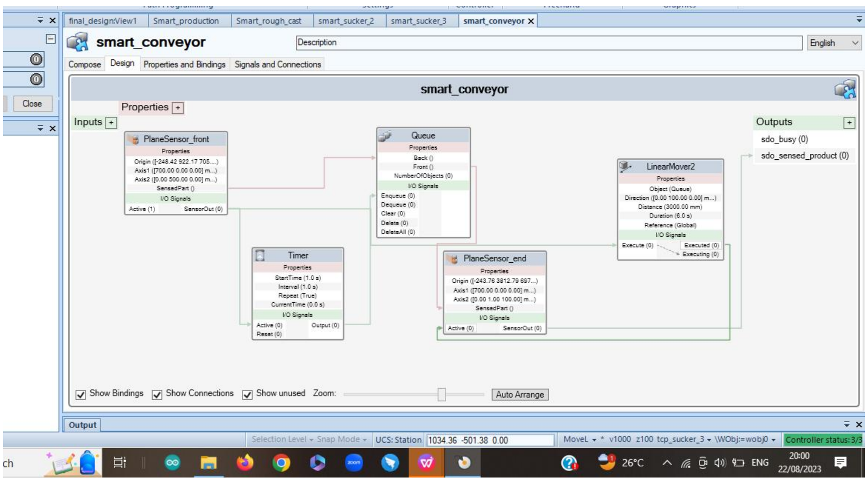

Figure 18 Smart_conveyor component logic design

System 3:

This system is meant to control robot 3 to pick the finished_product at the other end of the conveyor and stack it on workbench_2.

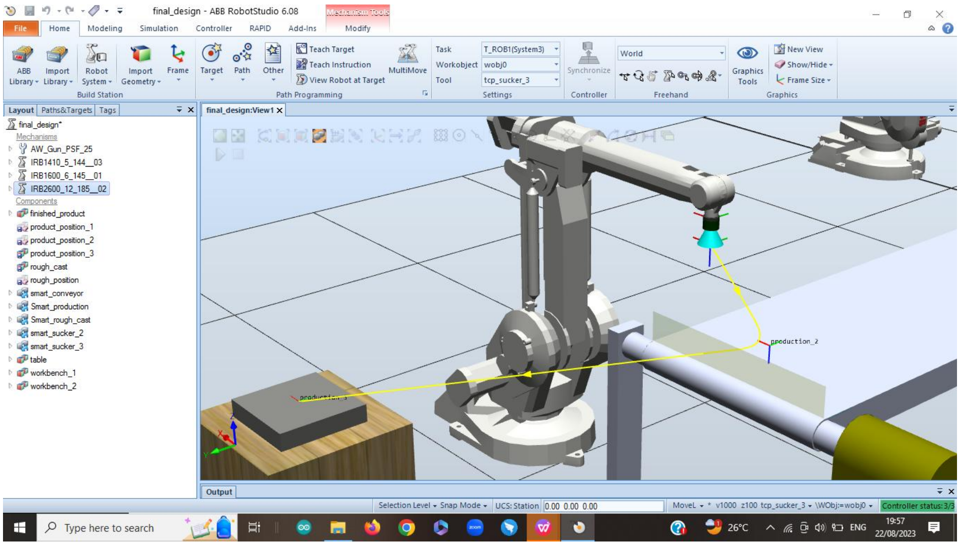

Figure 19 Robot (IRB 1410)

The smart component smart_sucker_3 makes an appearance here, which follows the same design as seen in *Figures 14 and 15.*

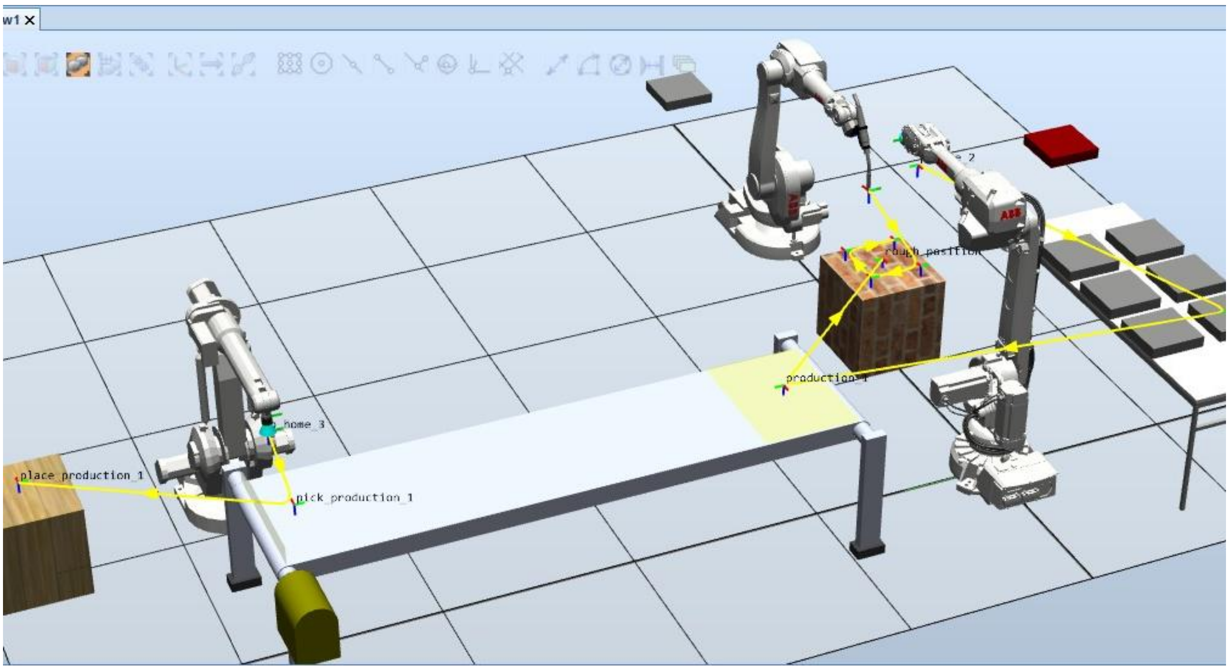

Figure 20 Completed workstation

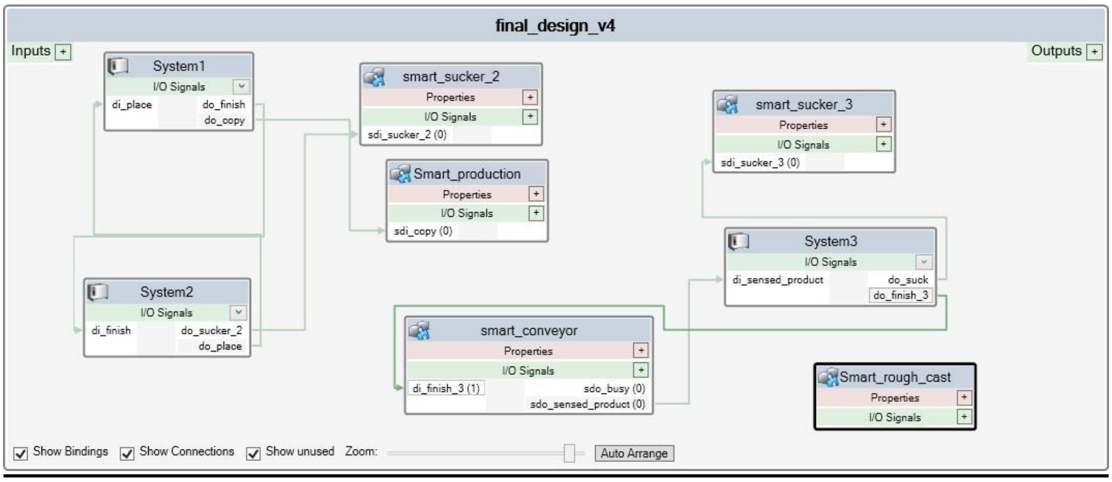

Figure 21 Station logic

3.2 Rapid Program Design

System 1 Rapid Program Design:

```
PROC main()
        !Add your code here
        MoveJ p_home_1, v1000, fine, AW_Gun;
        FOR ii FROM 0 TO 5 DO
            Reset do_copy; !=0
            Reset do_finish; !=0
            WaitTime 0.5;

            WaitUntil di_place=1; !triggers painting
            MoveJ Target_10,v200,z10,AW_GUN;
            MoveJ Target_20,v200,z10,AW_GUN;
            MoveJ Target_20,v200,z10,AW_GUN;
            MoveJ Target_40,v200,z10,AW_GUN;
            !MoveL Target_10,v200,z100,AW_GUN\WObj:=wobj0;
            WaitTime 0.5;

            Set do_copy;
            WaitTime 1;

            MoveJ p_home_1,v1000,fine,AW_GUN;
            Set do_finish;
            WaitTime 1;
        ENDFOR
    ENDPROC
```

System 2 Rapid Program Design:

```
PROC main()
            !Add your code here
             MoveJ p_home_2, v1000, fine, tcp_sucker2;
            FOR xx FROM 0 TO 2 DO
                  FOR yy FROM 0 TO 1 DO
                        Reset do_sucker_2; !=0
                        Reset do_place; !=0

                        MoveJ Offs(p_pick_2, 400*xx, 400*yy, 200), v1000, fine,
tcp_sucker2;
                        MoveL Offs(p_pick_2, 400*xx, 400*yy, 0), v200, fine,
tcp_sucker2;
                        Set do_sucker_2; !picks up rough_cast
                        WaitTime 0.5;

                        MoveL Offs(p_pick_2, 400*xx, 400*yy, 200), v200, fine,
tcp_sucker2;
                        MoveJ Offs(rough_position, 0, 0, 200), v1000, fine, tcp_sucker2;
                        MoveL rough_position, v200, fine, tcp_sucker2;
                        Reset do_sucker_2; !places rough_cast on workbench 1
                        WaitTime 0.5;

                        MoveJ p_home_2, v1000, fine, tcp_sucker2;
                        Set do_place; !triggers system 1
                        WaitTime 0.5;
                        Reset do_place;

                        WaitUntil di_finish=1; ! sends a signal once smart_production is
done

                        MoveJ Offs(rough_position, 0, 0, 200), v1000, fine, tcp_sucker2;
                        MoveL rough_position, v200, fine, tcp_sucker2;
                        Set do_sucker_2; !picks up finished_product
                        WaitTime 0.5;

                        MoveL Offs(rough_position, 0, 0, 200), v200, fine, tcp_sucker2;
                        MoveJ Offs(production_1, 0, 0, 200), v1000, fine, tcp_sucker2;
                        MoveL production_1, v200, fine, tcp_sucker2;
                        Reset do_sucker_2; !places finished product on conveyor
                        WaitTime 0.5;

                        MoveL Offs(production_1, 0, 0, 200), v200, fine, tcp_sucker2;
                        MoveJ p_home_2, v1000, fine, tcp_sucker2;
```

```
        ENDFOR
    ENDFOR
    WaitTime 10;
ENDPROC
```

System 3 Rapid Program Design:

```
PROC main()
    !Add your code here
    Set do_finish_3;   !send signal to trigger conveyor
    WaitUntil di_sensed_product = 1; !wait until plane_sensor_end senses to start
    MoveJ p_home_3, v1000, fine, tcp_sucker_3;

    FOR zz FROM 0 TO 5 DO
        Reset do_suck; !=0

        MoveJ Offs(pick_production_1,0,0,200),v1000,fine,tcp_sucker_3;
        MoveL Offs(pick_production_1,0,0,0),v200,fine,tcp_sucker_3;
        Set do_suck; !picks product
        WaitTime 0.5;

        MoveL Offs(pick_production_1,0,0,200),v200,fine,tcp_sucker_3;
        MoveJ Offs(place_production_1,0,0,50*zz+200),v1000,fine,tcp_sucker_3;
        MoveL Offs(place_production_1,0,0,50*zz),v200,fine,tcp_sucker_3;
        Reset do_suck; !places product on workbench_2
        Reset do_finish_3; ! turns off conveyor
        WaitTime 0.5;
        MoveL Offs(place_production_1,0,0,50*zz+200),v1000,fine,tcp_sucker_3;

        MoveJ p_home_3, v1000, fine, tcp_sucker_3;
        Set do_finish_3; ! =1, turns on plane_sensor_front
        WaitUntil di_sensed_product = 1; !triggers code loop

    ENDFOR
ENDPROC
```

4 . Conclusion

At their core, robots still do the mundane, monotonous and time-consuming tasks employees don't enjoy while providing invaluable data to improve operations. Without sensors and other recent advancements, none of this would be possible.

If we consider the future, Artificial intelligence (AI), of course, is a key part of the conversation. AI will reshape the capabilities of robotic technology, but that's still at least a decade in the future. In the meantime, we have these industrial machines that are making a clear impact in various industries- especially manufacturing [8].

In industrial robotics, success is about exponentials. It's riding exponential technology curves in related industries to robots in order to drive adoption. It's hard to drive technology innovation within an industry-but when you adopt technologies from "outside" industries, opportunities are limitless. The robotics industry is just beginning.

References

[1] Operating manual - RobotStudio. Copyright 2008-2018 3HAC032104-001 Revision: X, pp 13.

[2] Zhang Ruishu et al, The Status and Development of Industrial Robots. 2018, IOP Conf. Ser.: Mater. Sci. Eng. 423 012051

[3] W. Shen, Research on virtual simulation design of ABB robot welding operation based on Robotstudio [J]. 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 894-897, doi: 10.1109/ICAICA50127.2020.9182551.

[4] Shawn Dietrich, "Introduction to ABB Robot Programming Language" https://control.com/technical-articles/introduction-to-abb-robot-programming-language/

[5] "What is an industrial robot. Industrial robot definition", https://robotnik.eu/what-is-an-industrial-robot-industrial-robot-definition/#:~:text=An%20industrial%20robot%20is%20one,and%20processes%20revolve%20around%20them

[6] "Ten Popular Industrial Robot Applications", https://www.jabil.com/blog/ten-popular-industrial-robot-applications.html

[7] "Manufacturing Applications of Automation and Robotics", https://www.britannica.com/technology/automation/Manufacturing-applications-of-automation-and-robotics