8-2021

Design and Simulation of a Supervisory Control System for Hybrid Manufacturing

Michael Buckley
mbuckle6@vols.utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Michael Buckley entitled "Design and Simulation of a Supervisory Control System for Hybrid Manufacturing." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

William R. Hamel, Major Professor

We have read this thesis and recommend its acceptance:

William R. Hamel, Tony L. Schmitz, Bradley H. Jared

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Design and Simulation of a Supervisory Control System for Hybrid Manufacturing

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Michael Buckley

August 2021

To my parents Dan and Leslie for always enabling my ambitions,

my brother Sean for keeping me sane,

and my friends for supporting and carrying me through everything.

Acknowledgments

I would like to thank my advisory committee of Dr.'s Bill Hamel, Bradley Jared and Tony Schmitz for reviewing this thesis and aiding in the development of this system. My involvement in this project is due to Dr. Caleb Rucker taking me on as an undergraduate research assistant during my junior year of my undergraduate career. He answered my questions about transitioning to graduate school, as well as any question I ever had about something I didn't quite understand. He also passed me on to Dr. Hamel to be his graduate research assistant, to better help me obtain funding for my degree. Dr. Hamel has been a wonderful graduate mentor and my job assignment from him to be in charge of supervisory control has helped me find a career path that I want to pursue. The research teams of all of these professors have all been resourceful in their own ways for various jobs I have been tasked with.

I also want to thank my parents Dan and Leslie Buckley for always enabling me to follow whatever path I have chosen in my life. Their unyielding support has led me to everything I've ever achieved. My brother Sean Buckley has always been someone I can turn to to keep my mental health in check. My close friends, and roommates, have always been there for me whenever I've needed it. I don't know where I would be if it weren't for everyone's positive influence on me, especially during the difficult stages of my life. There are too many people to thank individually, but they all know who they are and the impact they've had.

Abstract

The research teams of Dr. Bill Hamel, Dr. Bradley Jared and Dr. Tony Schmitz were tasked by the Office of Naval Research to create a hybrid manufacturing process for a reduced scale model of a naval ship propeller. The base structure of the propeller is created using Wire Arc Additive Manufacturing (WAAM), which is then scanned to compare created geometry to desired geometry. The propeller is then machined down to match the desired geometry. This process is iterated upon until the final product meets design tolerances. Due to the complex nature and numerous industrial machines used in the process, it is desirable to create a control system for Supervisory Control and Data Acquisition (SCADA). This supervisory control system is necessary in order to ensure safe operations and logging of system data to document successful trials.

The goal of this thesis is to outline the design and simulation of a supervisory control system for this hybrid manufacturing cell. The design and implementation is focused on a simulation of the control of relevant boolean states of the system. This is accomplished through a Human Machine Interface (HMI) created in LabVIEW accompanied by appropriate data flow diagrams, models and communication specifications between machines. The creation of a digital twin of this hybrid manufacturing system was successful and useful in the implementation of physical components.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction, Background & Motivation

1.1 Hybrid Manufacturing

Additive manufacturing is a continuously growing industry due to its customization potential for prototyping and final parts [3]. Subtractive manufacturing, otherwise known as machining parts out of stock material, is equally as important. This project's hybrid process utilizes iterative additive and subtractive manufacturing with robotic arc welding and 5-axis machining, respectively. Traditionally, naval propellers are created by casting [19]. This practice was more useful when propellers were more similar to one another, which changed once more emphasis on hydrodynamic performance was placed. Creating a cast for a part that is not going to be repeatedly manufactured is unnecessarily costly. Since these propellers are sometimes 6 feet in diameter, traditional propeller casts take a long time to cool. The goal of utilizing hybrid manufacturing for this process is to expedite the manufacturing process, save on cost and reduce material waste.

On the additive manufacturing side, robotic WAAM is used to create a scaled down version of propeller geometry. Before any machining can be performed on the result, it must be scanned to create a 3D mesh of created geometry to compare to a known Computer Aided Design (CAD) file of desired part geometry. This step is necessary due to thermal expansion, weld spatter, limited resolution and low accuracy of WAAM [7]. Weld paths can be optimized to make parts closer to a desired shape, but there will always be areas that are over-built or under-built [6]. Once the surface representation of the printed part is known, the machining of the part can begin. A CAD file of desired geometry is used as reference for Computer Numeric Control (CNC) machining paths, used to remove excess material and create a desirable surface finish. The machined part must then be scanned again for another comparison to desired geometry. This process is repeated as many times as necessary to create a satisfactory result. An example of a propeller created using this technology is shown in Figure 1.1.

1.2 Supervisory Control

The hybrid manufacturing process described in this work utilizes multiple industrial machines, including but not limited to: robotics, welding equipment, 3D scanners and CNC machines. There is not one dedicated communication protocol to enable all of these different components to communicate with one another, so it is desired to create a supervisory control system for this process. The goal of the resulting system is Supervisory Control and Data Acquisition (SCADA). The supervisory control component is in charge of enabling machines to perform operations in a designated order to create a part, and the data acquisition component is in charge of appropriate data logging of relevant machine messages and process data while the system is in use. This technology can be used to create any reference part to varying success, but the primary function of this system is to create a naval propeller.

The planned implementation of this system utilizes a dedicated supervisory control PC running LabVIEW as a Human Machine Interface (HMI), connected to a Programmable Logic Controller (PLC) that is in turn connected to the appropriate components of the manufacturing system. The biggest challenge with this implementation is ensuring compatibility to send and receive communications from the supervisory control PC to the PLC and beyond to all the industrial machines that will be running. At some points of use, such as the loading of parts, various systems will run synchronously. However, primary operations of printing, scanning and machining will not run synchronously to each other. These machines will be utilizing digital input and output in order to communicate with the

Figure 1.1: Printed propeller test 1' diameter

control system, which will be sending enabling signals based on created boolean logic. Some machines will be sending large data files that need to be properly logged for other stations to access. It is crucial that this system only allows actions to be performed in a designated order to ensure quality results and to avoid any operational conflicts.

1.3 Data Acquisition

It is important that all data, as well as any error messages, are properly logged in order to improve on process quality and speed [10]. These data sets are different based on which subsystems in the overall cell are used. For robotic welding, it is desirable to have logs of weld current and voltage during the creation of the propeller in order to optimize process parameters [6]. The scanning subsystem creates 3D CAD scans of the propeller geometry either after printing or machining. These scan files are necessary know where to add or remove material. These files are the largest that need to be stored, and also need to be accessed by multiple subsystems to complete the manufacturing process. The machining subsystem keeps logs of machine code that are run, which need to adapt to geometry differences that occur with every test ran.

1.4 The Boolean State Control Problem

When implementing a supervisory or state control system, there are various schema that can be implemented depending on preferred modelling techniques. For implementation of control using a PLC, two possible modelling styles are Petri nets and (relay) ladder logic programming.

1.4.1 Petri Nets

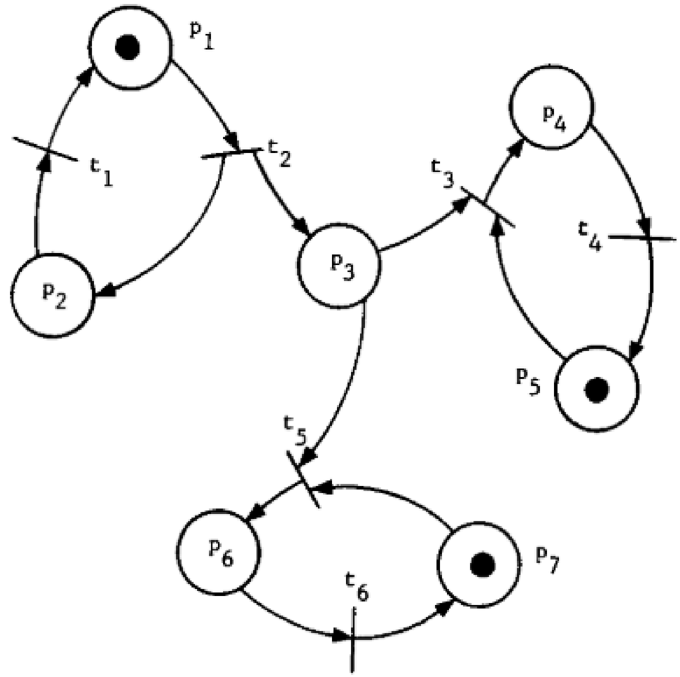
A Petri net is an abstract model of state information flow. A Petri net consists of 3 major components. These node components are circles, also called places, bars, also called transitions, and markings. A Petri net is a graphical representation of the states of a system and the transitions between them. Places are connected by transitions, and a place can only "activate" a transition if it is marked. This activation will move a place's marking to any subsequent places connected by the transition [13]. An example of a basic Petri net firing can be seen in Figure 1.2. Place 1 is marked, which enables transition 2 to be activated. Since transition 2 is connected to places 2 and 3, this activation will move the marking from place 1 to both places 2 and 3, which can be seen in the differences between Figures 1.2a and 1.2b.

A Petri net is able to be extended and adapted in order to model more complicated processes. This extension enables them to be useful in a wide array of system modelling. The extension most relevant to this supervisory control system is the modelling of discrete-event systems. This system type can also be compared to a finite-state machine, a system that can be represented by a certain number of states and the transitions between them [8]. The logic of more complex transitions involves more modelling to ensure order, especially making sure that marks aren't unnecessarily copied by transitions that send a mark to more than one place. The possibility for marking duplication can potentially cause errors with systems being enabled when they are not supposed to be.

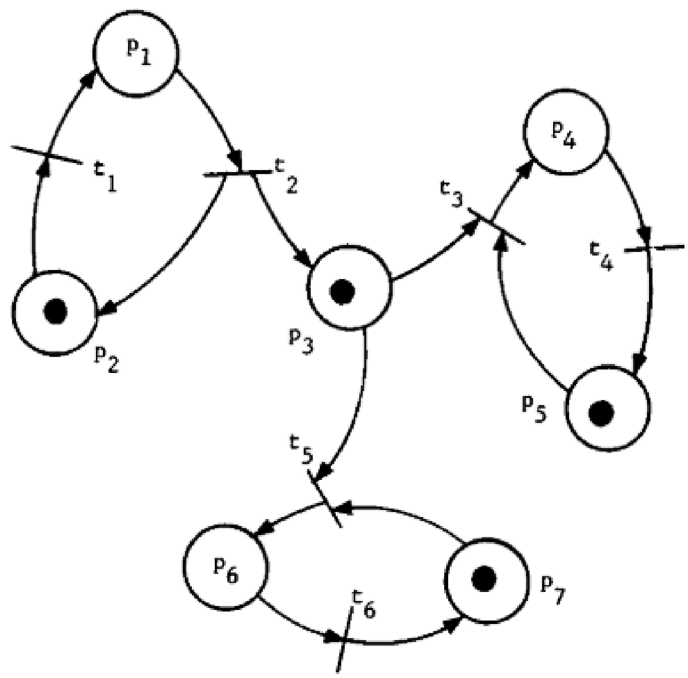
1.4.2 Ladder Logic Programming & Boolean Algebra

While Petri nets are a good modelling technique for discrete event systems, there are not readily available controllers based upon them [18]. Typical control structures use PLCs, which use their own graphical language: ladder logic. The main difference between ladder logic and Petri nets is that ladder logic doesn't fully capture sequential, asynchronous and concurrent events that need to be controlled [22]. This is due to the difference in representation of system actions. While a Petri net is most similar to a type of block diagram, with direct lines between each state, ladder logic programming uses more explicit boolean logic to transition between system states. It is harder to denote order in a ladder logic controlled process due to the way programs execute.

Boolean logic implementation in ladder logic consists of markers on a horizontal line to denote typical programming logic gates such as: AND, OR and NOT. These markers also have to take into account the way a component behaves when triggered. A normally open (NO) component will activate when supplied with an appropriate power signal, while a



(a) Petri net before firing



(b) Petri net after firing transition 2

Figure 1.2: Petri net transition firing [13]

normally closed (NC) component will be active when not powered. In other words, a normally open component requires a state to be on, or true, while a normally closed component requires a state to be off, or false. A NC component corresponds to a logical NOT gate. Ladder logic programming executes its logic gates from left to right and from top to bottom, just like reading. The program runs continuously, checking for input changes that affect the appropriate outputs.

A model of a discrete event system for this hybrid manufacturing cell was created in ladder logic, shown in Figure 1.3. All vertical parallel lines represent a NO component, requiring a signal to execute, while the vertical parallel lines connected by a diagonal represent a NC component. Line 7 corresponds to a typical logic AND statement, requiring all 3 inputs to enable the task at the right side of the line. Lines 6 though 6.3 all are logic equations to enable the task at the end of line 6, but the vertical junction corresponds to a logic OR statement. This means that only one of the 4 sub-lines before the OR junction need to be true, as well as the NO component at the end of line 6. In other words, it is a 4 line OR statement coupled with an AND statement. These are basic examples of how ladder logic and boolean logic function. The advantage to modelling boolean control in ladder logic is that it allows a control scheme to be directly created from the system state model.

1.5 Similar Supervisory Control Systems & Structures

There are various modelling techniques and control paradigms that can work for any system. Part of scheme selection depends on desired results, budget and interfacing software. A dynamic system consisting of interacting discrete and continuous components is considered to be a hybrid system [16, 5]. This description is an accurate definition for the hybrid manufacturing system laid out in this thesis. There are various discrete components of equipment that are enabled or disabled, as well as the continuous processes of printing, scanning and machining that need to be monitored and controlled.

Shuang et al. created a multi-level model for a hybrid manufacturing system utilizing Matlab and Arena [16]. Arena is a simulation software that primarily uses discrete event theory. Matlab is a programming software that was used to communicate to the system

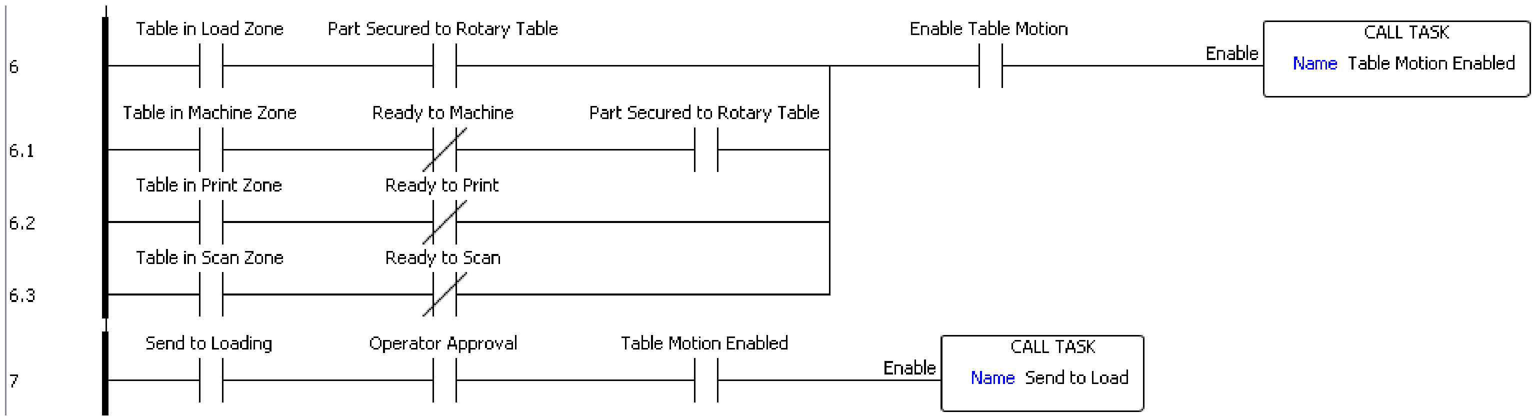
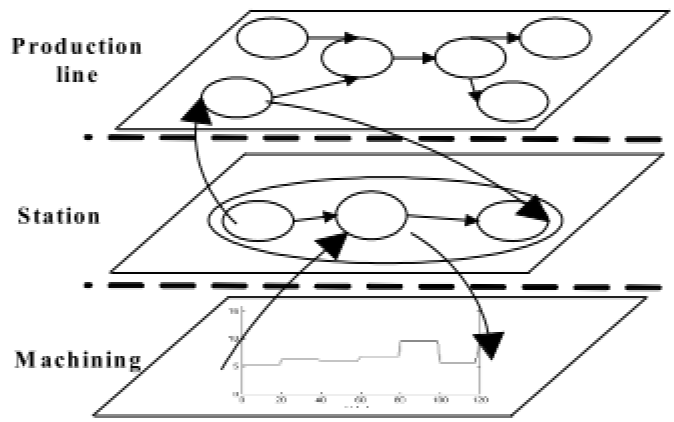


Figure 1.3: Ladder logic example code from this system

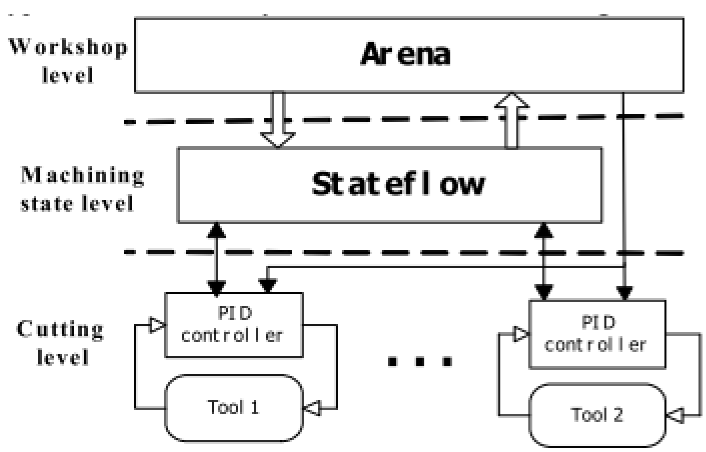
devices. The main structure of his group's system was a 3 level model designed to reveal microscopic continuous system characteristics as well as global discrete features. The 3 levels can be seen in Figure 1.4a and 1.4b. The workshop model was created in Arena in order to simulate the discrete events and transitions of various workshop stations. The machining state level was modelled in Stateflow, which was used to simulate the various machining states of different machine tools. This level of the hierarchy is primarily concerned with the continuous processing time of machining, while the cutting level below is in charge of the machining dynamics utilizing Proportional Integral Derivative (PID) and Proportional Integral (PI) controllers in Simulink. The main structure of this system utilized control selections made in Arena, which are then transmitted to Matlab in order to control the process. This system is effective in explaining the importance of integrating various programs for control of different levels of the process, which will need to be done for the hybrid manufacturing cell.

Koutsoukos et al. discussed in detail the creation of a supervisory control structure using various modelling techniques as well as the advantages and disadvantages of them [5]. Their base model of a hybrid control system is shown in Figure 1.5. In this model, the continuous processes being controlled, as well as their controllers, are represented by the plant box. These continuous parts of the system are typically represented by differential equations. The controller block of this model represents a discrete decision process that is described by the creation of a discrete event system. The intermediate interface block allows these continuous and discrete processes to communicate and interact with each each other. This coupling of discrete and continuous events and controllers is what makes this a hybrid control system. Depending on system complexity, this decoupling of different events can be convenient for mathematical hybrid system modelling. As systems become more complicated, like the one being modelled in this thesis, this modelling technique is most helpful to study the system properties rather than attempt to develop strategies for control, especially if the system does not have a natural separation of the continuous and discrete parts.

The caveat of this modelling technique is that it does not address continuous control problems due to the assumption that continuous control actions are a subset of the plant in Figure 1.5. This model also does not account for jumps in the time-dependent states



(a) Manufacturing system structure



(b) Manufacturing system architecture

Figure 1.4: Hybrid manufacturing system structure created by Shuang et al. [16]

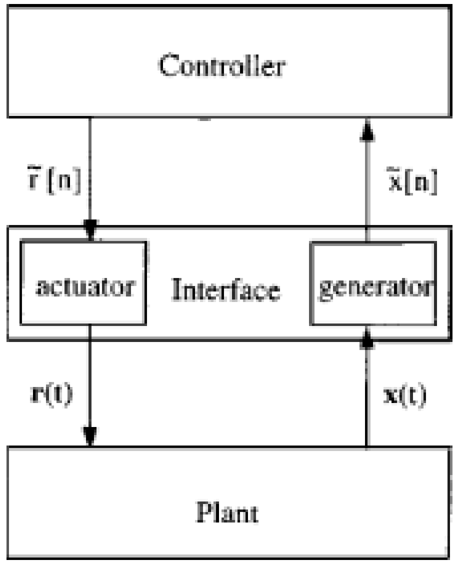


Figure 1.5: Koutsoukos' model of a hybrid control system [5]

when certain state variables are discontinuously reset. This action would correspond to an emergency stop or error sequence causing a shutdown. Since all system control and operation should stop in this case, this discontinuous reset ideally would not be an issue if the system is able to resume at a known state. Koutsoukos et al. describes a spectrum of the mathematical paradigms used to model hybrid control systems [5]. One end of the spectrum contains equation based models that include system discontinuities such as switchings and jumps. These system models typically are used in order to extend continuous systems to study the traditional control problems of stability, robustness and optimal control. The other end of the spectrum contains computer science based models that show how real-time embedded systems behave, a modelling technique called hybrid automata. In general, there are various ways to approximate a continuous system plant; this approximation depends on system complexity and desired model characteristics. This work focused on the general case of control system modelling rather than a specific system.

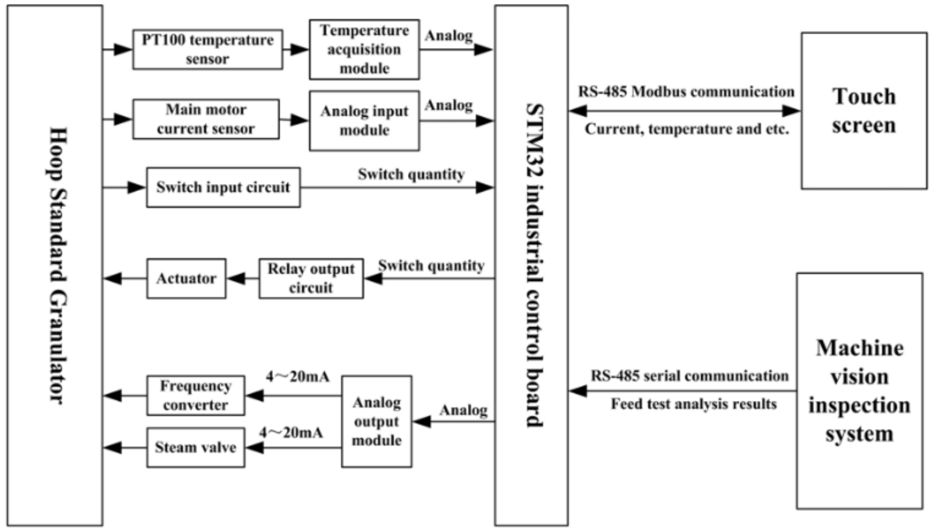
Yang et al. created a machine vision based supervisory control system with the purpose of controlling a hoop granulator for feed production [21]. The machine vision component of this system was used in order to detect various surface pits, cracks and defects of feed particles. The hybrid manufacturing cell laid out in this thesis also uses a machine vision component, but not for real-time error detection. The scanning station of the hybrid manufacturing cell is used to create 3D model geometry to feed data into other stations that run after the scan is complete. While there are differences in how the machine vision components of these systems are used, the modelling techniques used are applicable to the creation of any control system. Yang first created a mathematic model of the hoop granulator control system, which was used for control structure. Yang also created a data flow structure which is useful to show how various system components connect to one another. This data flow structure is shown in Figure 1.6a, while the control board that was used is shown in Figure 1.6b. The data flow diagram is a very helpful to transition between system simulation and system building. A data flow diagram for the hybrid manufacturing system was created for this exact purpose. Yang also created a software flow chart that shows the decision process of the system while it is in use, as shown in Figure 1.7. This flow chart structure is useful for showing the process sequence and for knowing what needs to be controlled or observed at certain times. While Yang's control system is able to follow a moderately linear flow chart, the state diagram created for the hybrid manufacturing cell is more complicated due to the various different stations used. Yang's work is an excellent example of modelling techniques that can be used to transition between a system model and a physical system.

1.6 Discrete Event Systems and Initial Modelling

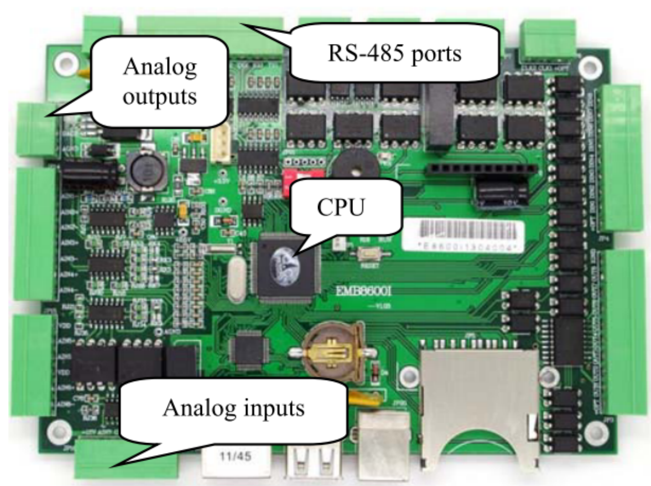
Regardless of the choice of Petri nets, boolean ladder logic, or other modelling techniques, the system being modelled is a Discrete Event System (DES). A DES can be considered as many finite state machines, an event driven system that describes the state transitions of a system. These state transitions can correspond to certain operations running, machines being active and other boolean values that describe the system [20]. This model is not explicitly concerned with continuous overlapping states or processes since only one primary station will be in use at a time during the beginning phases of this work cell. At certain stages, some equipment will be collaborating to perform a transition, but a majority of operations being performed will be by a single subsystem. Typically, a finite state machine will describe one component or subsystem of an assembly. For example, a DES would be the overall hybrid manufacturing cell while a finite state machine would describe the states that one of the robots can be in: turned on/off, loaded, working, waiting for instructions, and so on. A DES is essentially a grouping of many finite state machines.

1.7 Programmable Logic Controllers

There are different control processers that can be used to create control systems for simple or complicated processes, but one of the most widely used is a PLC, a computer based device that executes ladder logic programs [1]. They can be used for a variety of applications, from small scale control to industrial environments [4]. They are useful for their implementation of boolean ladder logic and how they can easily implement control based on a mathematic model. Netto et al. created a comparison chart between using a PLC versus other control boards for a generic control system problem, shown in Figure 1.8 [9]. PLCs are one of the



(a) System data flow



(b) Control Board

Figure 1.6: Data model and control board used by Yang et al. [21]

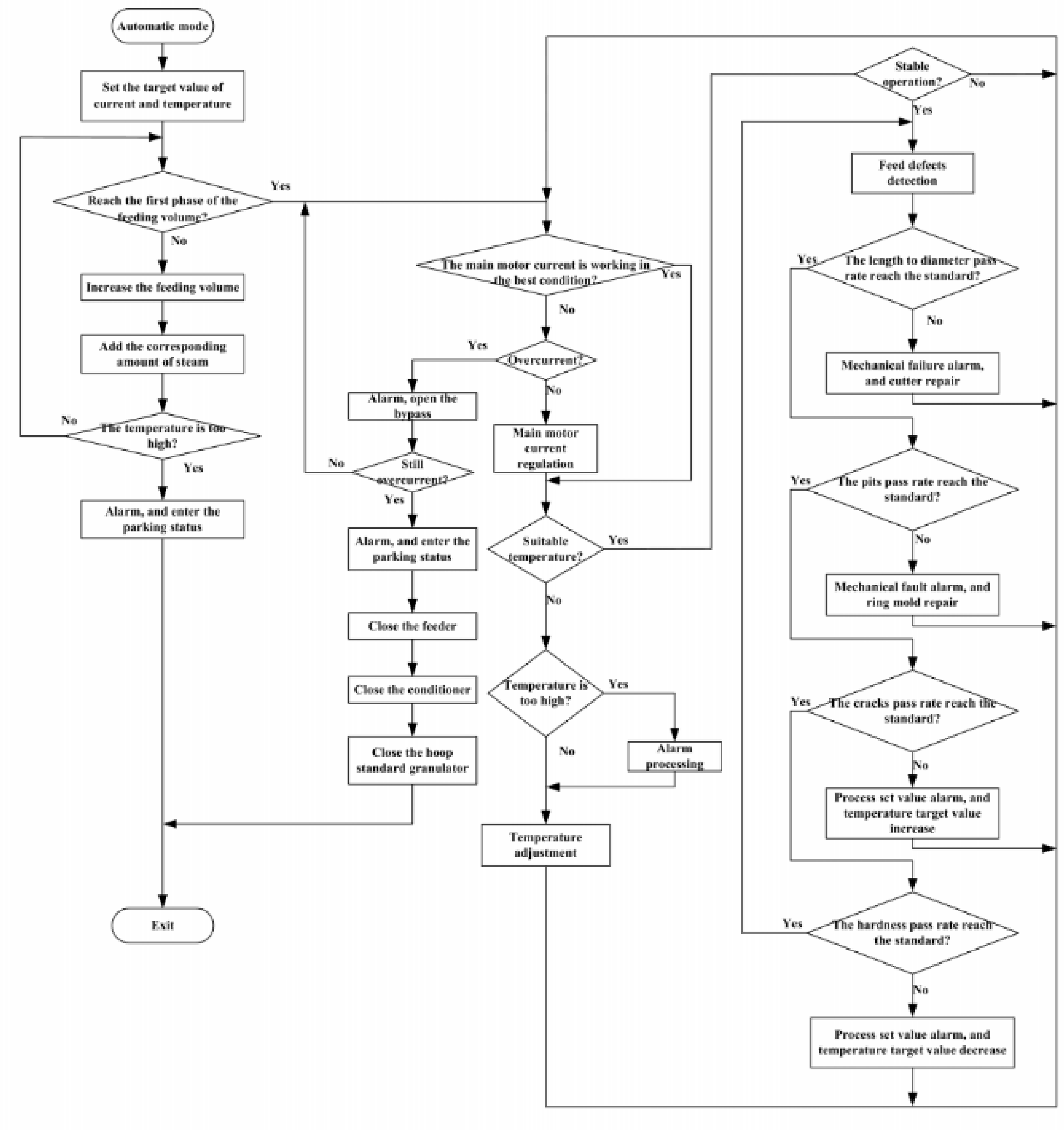


Figure 1.7: Yang's software production flow chart[21]

most widely used control structures due to their overall quality and application capacity being better than most other options, especially with cost and customization options. For this system, it was decided that a PLC would be a slave to the supervisory control PC for its ability to actuate numerous components and have its many inputs and outputs routed from the same location with the option to add more communication modules if necessary.

Parameters	Relays	Solid-State Controls	Microprocessor	Minicomputer	Digital Logic	PLCs
Hardware cost	Low	Equal	Low	High	Average (can be high in small quantities)	Depends on number of controls
Versatility	Low	Low	Yes	Yes		Yes
Troubleshooting and maintainability	Poor	Poor	Poor	Poor	Poor if IC's are soldered	Good
Computer compatible	No	No	Yes	Yes	Yes	Yes
Arithmetic capability	No	No	Yes	Yes	Yes	Yes
Programming cost	(Wiring) High	(Wiring) High	High	High	Low	Low
Reusable	No	No	Yes	Yes		Yes
Space required	Largest	Large	Small	Ok	Fairly compact	Small
Operating speed	Slow	Faster than electro-mechanical relays	Fairly fast	Fairly fast	Fairly fast	Fast

Figure 1.8: Comparison of a PLC to other control systems [9]

Chapter 2

Initial Approach & Modelling

2.1 System Overview & Initial State Model

A digital model of this system was modeled in Octopuz, a 3D CAD environment used for offline robot programming. This software package allows a user to simulate the sequence of actions that a robot will perform based on input commands, which is useful for program testing and development without having to connect to a physical system. The model shown in Figure 2.1 shows the Octopuz model of this system. There are 3 primary stations that perform the operations of printing, scanning, and machining. The printing station consists of a KUKA KR-50 industrial robot outfitted with a Fronius cold metal transfer 2 wire welding torch, and is the additive component of this hybrid manufacturing process. This equipment prints the naval propeller on a designed pallet attached to the 2 axis rotary table that is mounted to the top of the linear track. Print paths for welding are created according to propeller reference geometry and a path planning software algorithm. When printing is complete, the linear track moves the completed part to the scanning station.

The printed part is scanned by a GOM ATOS-Q structured light scanner, which creates a 3D surface representation based on multiple mesh scans that are fitted together by its software package. The purpose of the scanning station is to know how created geometry compares to the desired geometry reference CAD file. This comparison can help optimize the creation of print paths and parameters, and it also gives a base model to use for machining

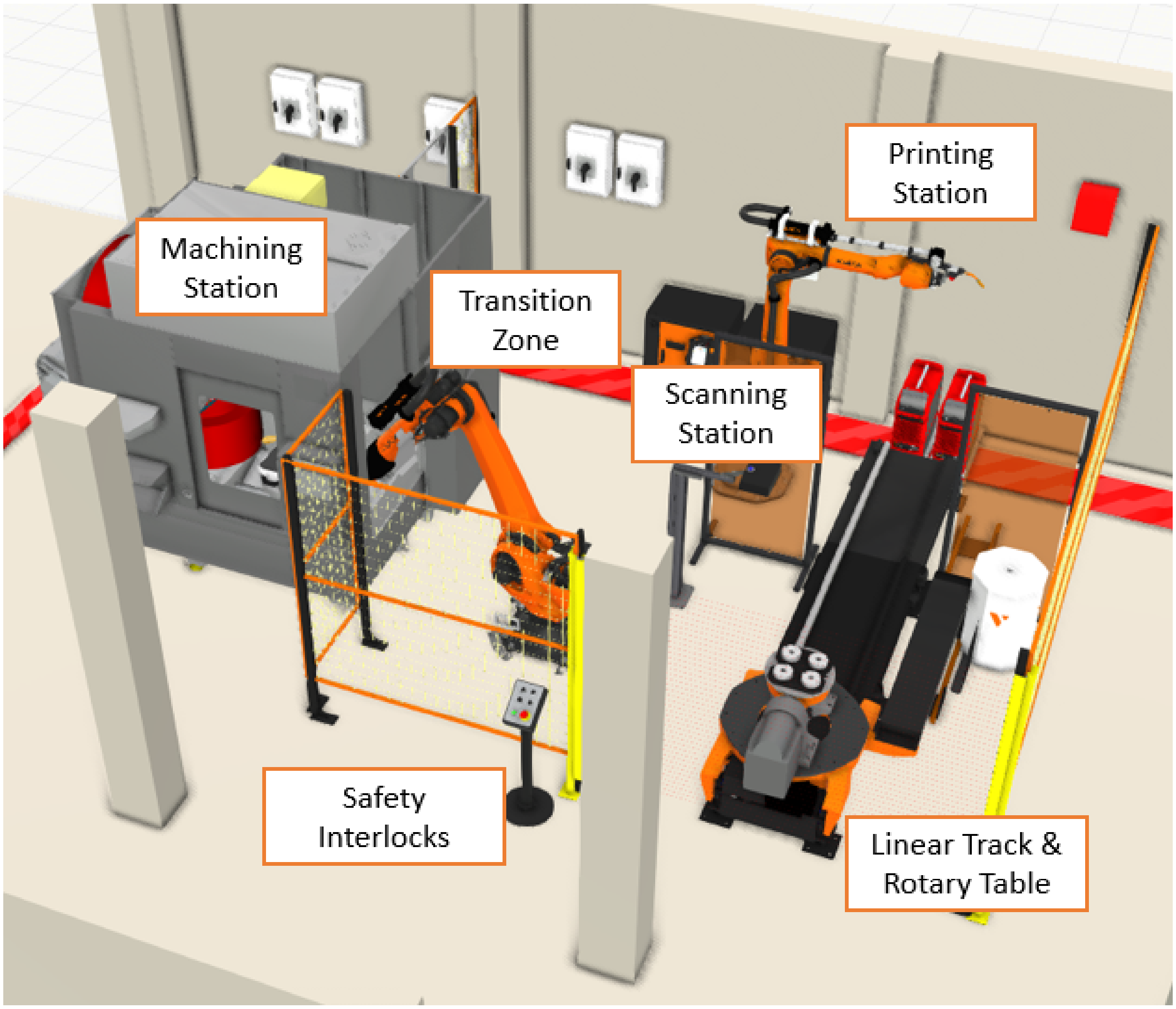


Figure 2.1: Hybrid manufacturing cell model

programming. When a part is created and scanned, the linear track moves the part towards the transition zone.

The transition zone consists of a KUKA KR-250 industrial robot equipped with SCHUNK grippers in order to pick up the pallet from the linear track to move it into the machining station and vice versa. The grippers and pallet are actuated by pneumatic solenoid valves that are controlled by the supervisory control system. Once a part is loaded into the HAAS UMC-750 5-axis CNC machine, the part can be machined to its desired final dimensions. The machining station is the subtractive component of the hybrid manufacturing process. Due to the layering of weld beads that form the part and uncertainty in final weld layer height, resulting prints need to be machined in order to match final geometry. It is important that the final product matches appropriate geometry for hydrodynamic performance. When machining is complete, the transition arm moves the part back to the rotary table in order to do another scanning check for geometric tolerances. It it also possible for this system to work iteratively. This iterative functionality allows the system to print part of the structure, machine it, and then print again and so on. This enables more complex structures, like a propeller, to be more easily machined rather than trying to fully machine a completed structure all at once.

The iterative capabilities of this hybrid manufacturing system allows for continuous changing between the addition or removal of material from a part. Depending on what a scan reveals about the created geometry, it can be either sent back to the printing station or to the machining station. Otherwise, the part can be removed from the system at an opening in the system safety interlocks. The safety barriers of this cell include: a locking access door, fencing with shielding to protect from the bright light of arc welding, and an opening with a light curtain. The purpose of the light curtain is to have an access point that operators can reach through that also functions as a stop if the laser grid is broken while the system is running. Parts are also loaded to and unloaded from the rotary table at this location. It is highly desirable to have a supervisory control system that can properly monitor, log and control these processes for simplicity, efficiency and safety.

Based on an overview structure of all of the primary operations of this cell, a state diagram was created for this system. These primary systems have numerous sub-states

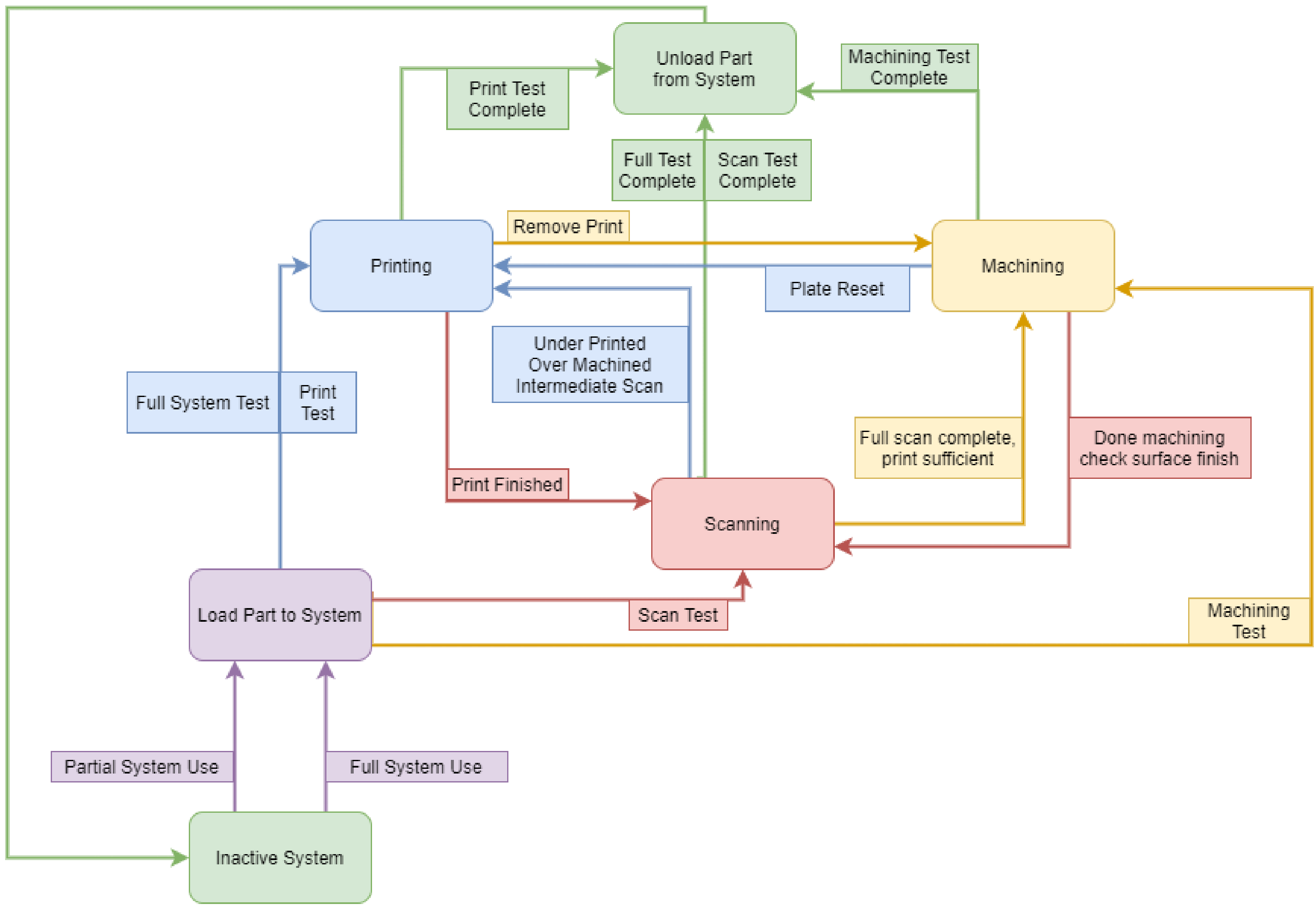


Figure 2.2: Hybrid manufacturing cell state diagram of subsystems

and checks that need to be read and controlled for specific operations. The overall system diagram is shown in Figure 2.2 and describes what actions are necessary to transition between subsystems, in a simplified overview. This model assumes that the system will always initialize from being inactive, where operations begin with the loading of a part or part pallet into the system to be sent to the various stations. This model is primarily concerned with the main three system operations of the system and specific actions that denote a transition between them, rather than all of the involved details of running a single operation.

2.2 System Boolean Algebra

In order to create a more involved model of the system, it is helpful to list and model the system booleans that need to be true or false for an action to occur. Mathematic set theory can be used to describe the combination of boolean variables that describe the overall state of a DES based on a list of boolean system values [14]. Set theory is a good way to define the logic of a DES, especially as a starting point. Before work was performed to create a DES model, the preliminary state diagram was used to create event set notation that describes the different operations of the cell and their booleans. These system booleans and states are shown in the following tables, where a boolean value is a true or false statement describing the system and a state is a certain combination of booleans that describe the system in more detail. Note the symbols used in boolean algebra: ! denotes a NOT statement, $\cup$ is an OR statement and $\cap$ is an AND statement, which have the same logical uses in ladder logic. Beginning with the loading system in Tables 2.1 and 2.2, the logic is fairly straightforward. The system states indicate whether or not the positioner on the rotary table is ready to receive a part based mostly on where it is currently located. Boolean C corresponds to the pneumatic grippers being open to clamp to the part pallet, and boolean D is a standard error that could be communication, software or hardware based.

All of the main subsystems follow a similar structure for states. The booleans that enable them are more specific to the process, but the three main states are ready, not ready, or currently performing their designated process. The booleans and states for the printing system are shown in Tables 2.3 and 2.4. Boolean F corresponds to a system operator

Table 2.1: Loading System Booleans

Boolean Value	Boolean Definition
A	Part Positioner in Loading Zone
B	Part Positioner Empty
C	Table Clamps Open
D	System Errors Present

Table 2.2: Loading System States

State	Boolean Algebra
1: Positioner ready to Receive Part	$A \cap B \cap C \cap !D$
2: Not ready to Receive Part	$!A \cup !B \cup !C \cup D$

providing approval for the print process to initialize and run, since this welding process will need to be continuously monitored to ensure safe operation. Possible errors of the print system are: necessity of slag cleanup, welding current or voltage being out of bounds, weld wire jamming in the torch, long arcing creating bad results, excessive thermal buildup or any sort of safety error. These error lists can also be modified to fit the operational needs of the system, since this supervisory control system will be checking various booleans describing the overall state.

The scanning system booleans and states are shown in Tables 2.5 and 2.6. The states are the same structure as the printing states, but the enabling booleans are specific to the scanning equipment. Possible errors that can occur in the scanning system are: undesirable print quality yielding a result that is unable to be scanned, the scan algorithm being insufficient for printed geometry, the part being too hot for the scanner to work properly or a safety error. This system is the least dangerous of the 3 main systems since nothing is being machined or welded, but it is still important to be following safety protocols and ensuring that operations stop if the safety barriers are breached or throw an error.

The booleans and states for the machining system are shown in Tables 2.7 and 2.8. This system depends on both the printing and scanning stations creating sufficient geometry and CAD to use as a basis to match reference geometry and surface finish. The states follow the same structure as the previous systems, with enabling booleans more specific to machining. The errors of machining, aside from safety concerns, are centered around machining dynamics. Possible errors of this system are: breaking of a machining tool, significant chatter during operations, over-machining the part, large pieces breaking off of the part or insufficient coolant causing chip and thermal buildup. For all of the systems of this cell, it is crucial to follow all process safety protocols and to keep track of all the necessary requirements to operate in order to have safe operations and produce quality results. These boolean statements for enabling of states aid in the creation of a control structure in a ladder logic environment due to an ease of transition between the two. This system will also be monitored by at least one operator that is supplying input to the control system, it will not be running without supervision.

Table 2.3: Printing System Booleans

Boolean Value	Boolean Definition
A	Part Loaded
B	Weld Power Supply On
C	Wire Loaded
D	Print Path Created
E	System Errors Present
F	Printing Enabled

Table 2.4: Printing System States

State	Boolean Algebra
1: Ready to Print (Print Complete)	A ∩ B ∩ C ∩ D ∩ !E
2: Not Ready to Print	!A ∪ !B ∪ !C ∪ !D ∪ E
3: Currently Printing	1 ∩ F

Table 2.5: Scanning System Booleans

Boolean Value	Boolean Definition
A	Part Loaded
B	Scanner Power Supply On
C	Scanning Algorithm Created
D	System Errors Present
E	Scanning Enabled

Table 2.6: Scanning System States

State	Boolean Algebra
1: Ready to Scan (Scan Complete)	A ∩ B ∩ C ∩ !D
2: Not Ready to Scan	!A ∪ !B ∪ !C ∪ D
3: Currently Scanning	1 ∩ E

Table 2.7: Machining System Booleans

Boolean Value	Boolean Definition
A	Part Loaded
B	CNC Machine Power On
C	Part Geometry Known
D	Machining Program Created
E	System Errors Present
F	Machining Enabled

Table 2.8: Machining System States

State	Boolean Algebra
1: Ready to Machine (Machining Complete)	$A \cap B \cap C \cap D \cap !E$
2: Not Ready to Machine	$!A \cup !B \cup !C \cup !D \cup E$
3: Currently Machining	$1 \cap F$

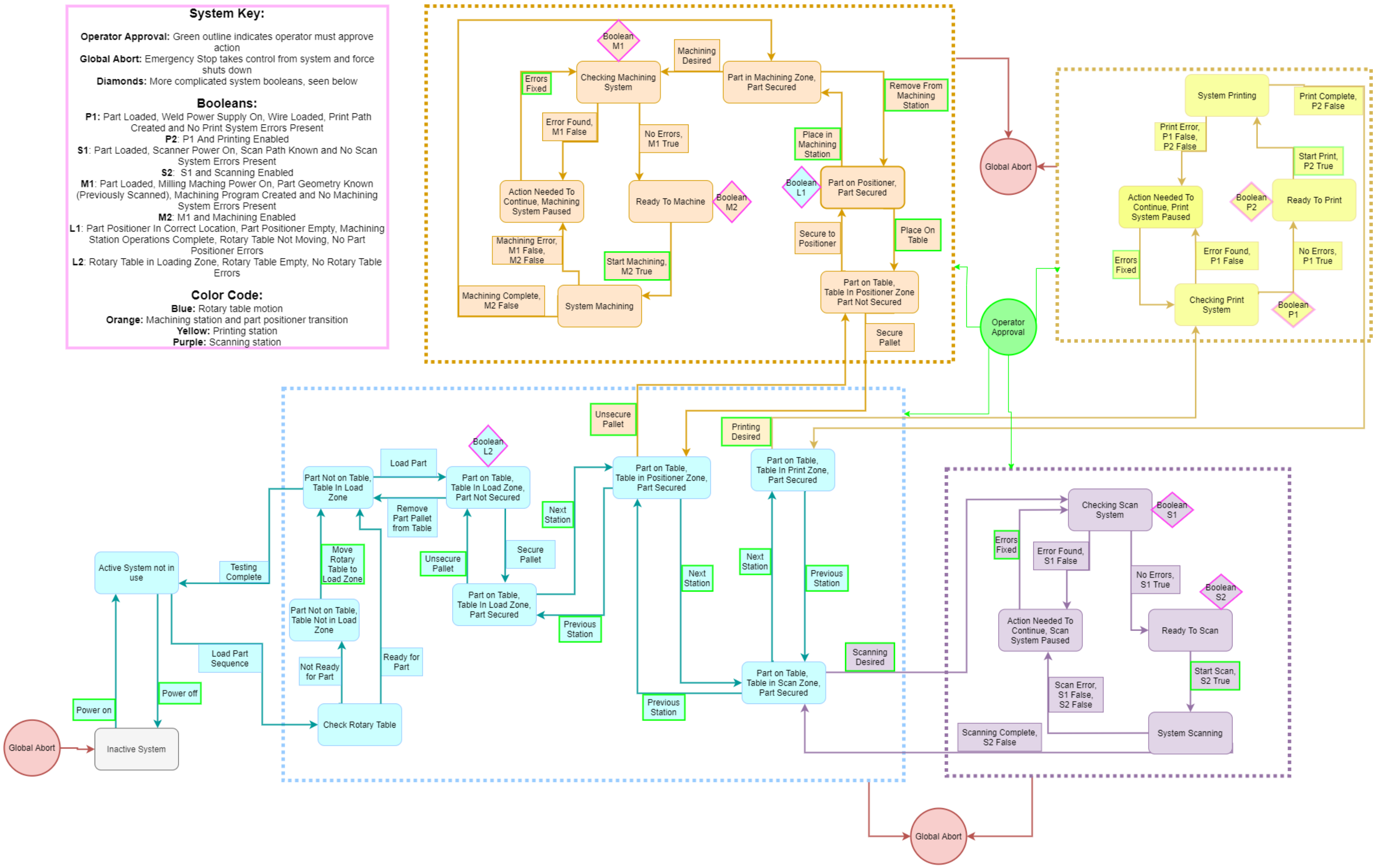


Figure 2.3: Hybrid manufacturing state diagram

2.3 State Diagram

From the overview state diagram model and the mathematic boolean algebra to describe system control, a more in-depth system state diagram was created. The overall diagram can be seen in Figure 2.3, and each component of it is discussed in the following subsections. The boolean key, corresponding to state algebra from the previous tables is shown in Figure 2.4. Operator approval represents any time that the system would need external human input in order to perform or continue an action. This prevents certain operations from running without an external source verifying that the machines are ready to perform that action. Global abort is any situation that would cause a stopping condition to be manually or automatically triggered. These stops can be triggered by an individual system fault, a global error or by an emergency stop button. Only one station should be running at a time but it is important that all motion and processes become disabled if a stop condition occurs, especially if a person is entering the cell behind the safety barriers in order to perform diagnostics or maintenance. Simultaneous processing of different stations will be possible in the future, but it is ideal to set up this system to only run one operation at a time to ensure safety and quality. Since the scanning and printing stations share the linear track and rotary table, synchronous operation can only occur between the machining station and either the printing or scanning station. The machining station is loaded by a robot from a side access door, but it could also be loaded from the front by an operator to machine other parts or to work on testing CAM programs.

2.3.1 Loading System and Rotary Table

The loading and transit system is the key to how all of the different operations are able to hand off workpieces between one another. On top of the linear track is a 2 axis rotary table that allows the part to be rotated in order to orient it in a desirable way. Attached to the rotary table is a removable part pallet that mounts the part to the top of it for secure fastening and easy removal. If any global stop condition occurs, the system will return to an inactive state which is also reached if the system is shut down under normal conditions. When the system is first turned on, or moved from an idle state, the position of the linear

System Key:

Operator Approval: Green outline indicates operator must approve action
Global Abort: Emergency Stop takes control from system and force shuts down
Diamonds: More complicated system booleans, seen below

Booleans:

P1: Part Loaded, Weld Power Supply On, Wire Loaded, Print Path Created and No Print System Errors Present
P2: P1 And Printing Enabled
S1: Part Loaded, Scanner Power On, Scan Path Known and No Scan System Errors Present
S2: S1 and Scanning Enabled
M1: Part Loaded, Milling Maching Power On, Part Geometry Known (Previously Scanned), Machining Program Created and No Machining System Errors Present
M2: M1 and Machining Enabled
L1: Part Positioner In Correct Location, Part Positioner Empty, Machining Station Operations Complete, Rotary Table Not Moving, No Part Positioner Errors
L2: Rotary Table in Loading Zone, Rotary Table Empty, No Rotary Table Errors

Color Code:

Blue: Rotary table motion
Orange: Machining station and part positioner transition
Yellow: Printing station
Purple: Scanning station

Figure 2.4: Key of the system state diagram

track and loading status needs to be validated by an operator in order to begin or continue operations. The control system will also be able to check the position of the rotary table. The section of the state diagram corresponding to the loading zone and rotary table can be seen in Figure 2.5.

The linear track has four different stopping points along its length in order to provide access to different stations. On the first end of the track, the part is in the Loading Zone from Figure 2.1. This state allows operators to reach in beyond a safety laser grid in order to either place or remove a part or pallet on the rotary table. The only actions that are permitted in these states are either securing or unsecuring the part pallet to the rotary table. Motion is only allowed if the table is empty or if a part pallet is secured to the top of it by the pneumatic solenoids that actuate it. Motion is not allowed if the part is not fully secured to the rotary table, or if a system is performing its operations to a workpiece on the rotary table.The next set of states along the length of the table corresponds to the transition zone from Figure 2.1, where the large part positioner robot arm is located. This is the only other part of the motion subsection that allows the actuation of the pneumatics holding the part to the table. The reason for this will be discussed further in the Machining and Part Positioner subsection. Beyond the part positioner zone is the scanning zone. There is no pallet changing that needs to occur for scanning operations to begin here, so it is important that the part stays securely fastened to the rotary table and pallet. On the furthest end of the linear track is the printing zone. There are more safety barriers surrounding this area due to the dangers that arc welding present to the human eye. Thermal cameras are used in order to properly and safely monitor the arc welding process from the outside of these barriers. Just like the scanning system, the part and pallet need to remain securely fastened during all printing operations.

2.3.2 Machining System & Part Positioner

The machining station is where a part will be machined down to accurately produce desired part geometry and surface finish, the portion of the state diagram corresponding to this process is seen in Figure 2.6. Due to the weight of the propeller and the fact that CNC machines need their own enclosed space to properly work, the machining station is loaded

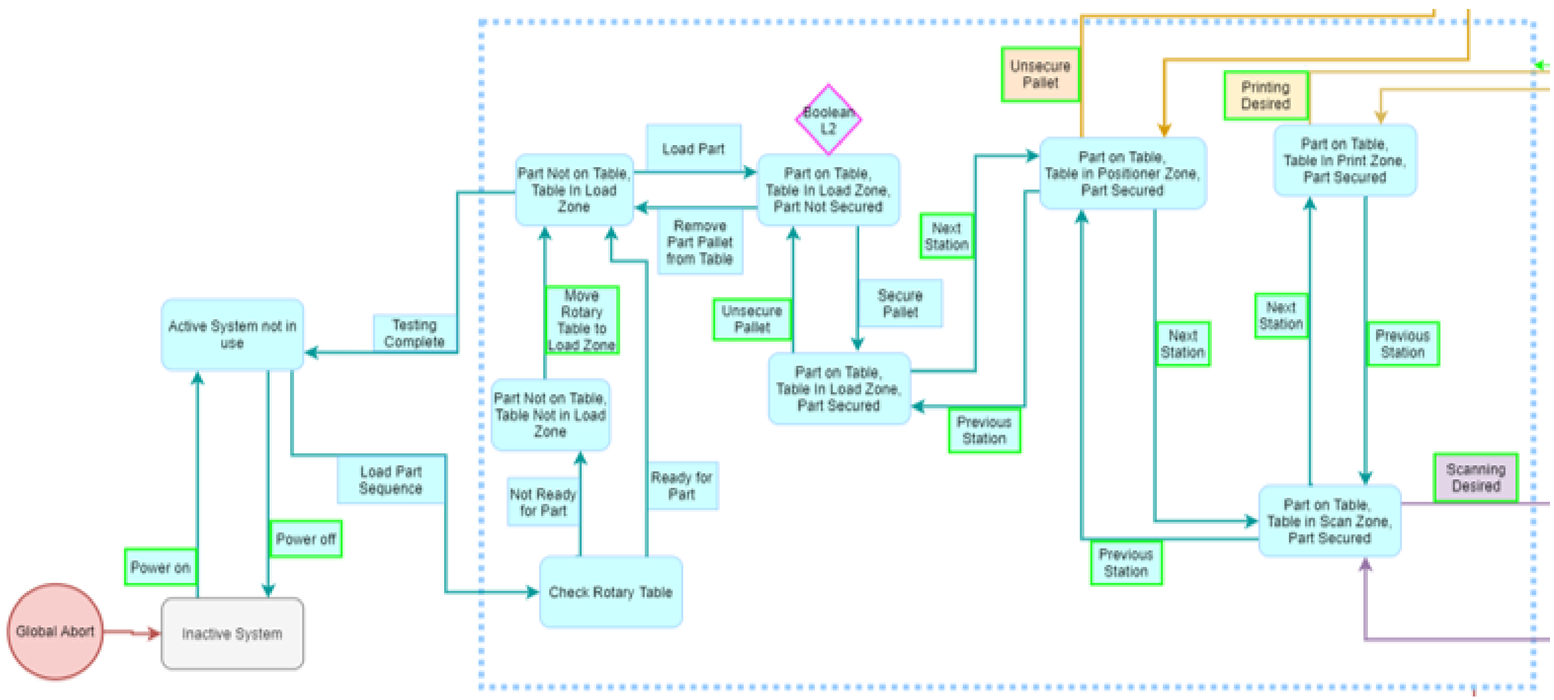


Figure 2.5: Loading and motion subsection of the state diagram

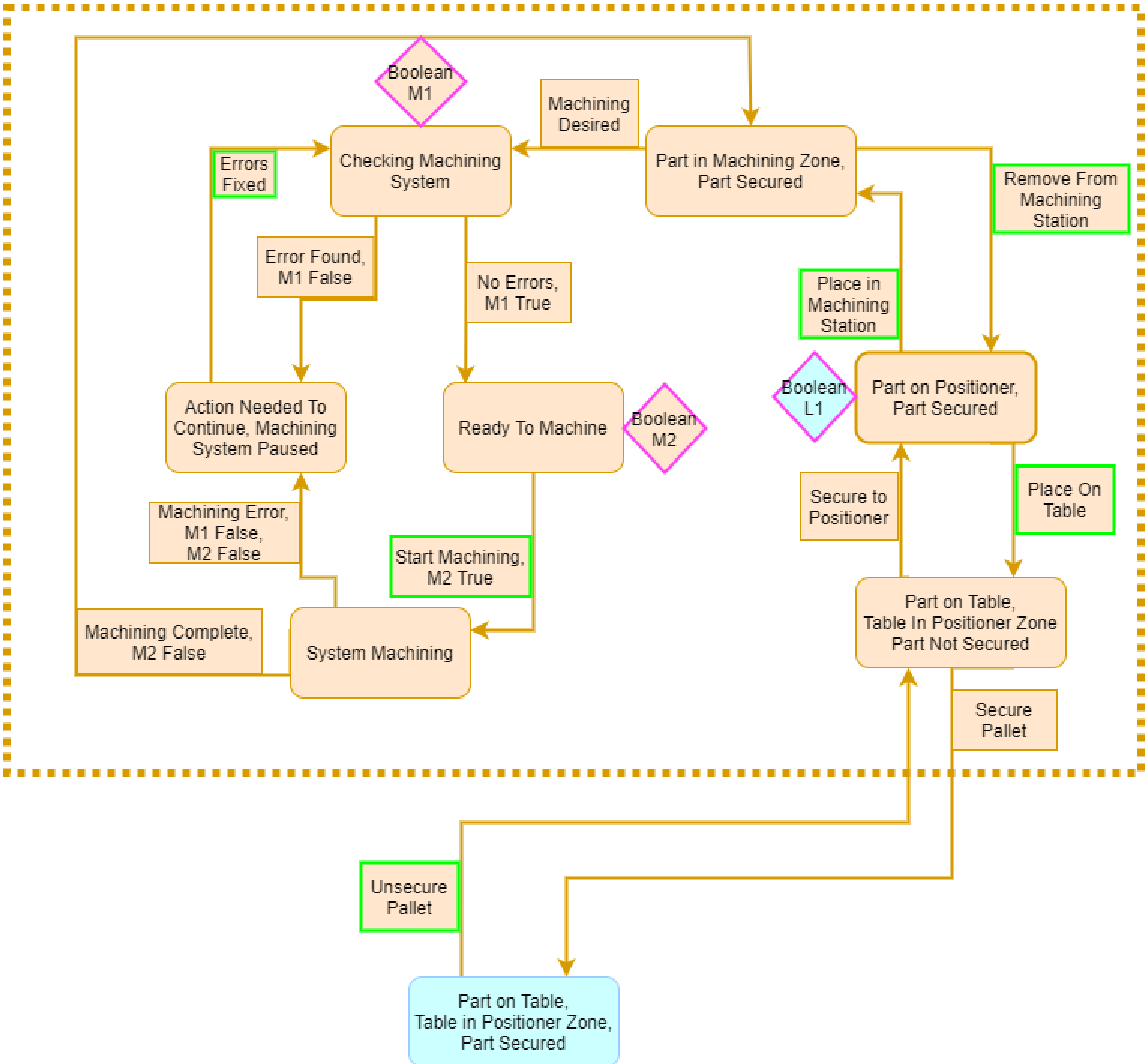


Figure 2.6: Machining and part positioner subsection of the state diagram

using a material handling robot through a side access door rather than through the manual front doors by human personnel. The side door is held open by pneumatic valves, and closed by the same valves and gravity. If machining is desired, the part pallet needs to be unsecured from the rotary table while it is in a fixed location and orientation. Ideally, the part pallet will have motion disabled, and will be resting vertically for easy access by the part positioner.

The part pallet has 2 locations that are able to be clamped on to. The bottom of the pallet is able to clamp to the base of the rotary table, or to the base of the worktable in the 5 axis milling machine. The side of the part pallet has the male side of a pneumatic gripper that is for use by the part positioner only. The KUKA KR-250 is able to clamp its female gripper on the side of the part pallet in order to pick it up and move it between the rotary table and the machining station. Once the part is properly loaded and secured inside the CNC machine, it is able to be machined down to a desired surface finish based on known geometry. Operator approval is needed through various states in this system for machining operations as well as positioner motion and pallet fixturing. The more complicated boolean algebra checks of this system correspond to ensuring that the machining station has all appropriate CNC programs created and that the part scan is manually deemed sufficient to use as a stock model for those programs. This is important in order to ensure that the created machining programs have the appropriate feeds, speeds and paths to create the final part geometry. Since the 5-axis machine is able to reach under overhangs, machining programs need to be more closely checked before use to ensure proper tool access. Multiple propeller blades create numerous overhangs that will need to be machined by this system. If the overhangs are too large on a completed part, tool access will be limited resulting in a more difficult machining process. Over-machining a part due to restricted tool access, or any other reason, would cause excessive damage to the final part. It is possible for pieces of an over-machined part to break off, which is not ideal for final quality and possibly dangerous. The iterative functionality of this cell helps with tool access, enabling the machining station to work on the part throughout various stages of its creation. This helps the machining setup work in a more ideal way, reducing the risk of part damage or tools breaking. The machining paths created for any stage of a part are dependent on sufficient scan geometry being supplied to the system.

2.3.3 Scanning System

The scanning subsystem is the least physically dangerous main operation that is performed in this cell, and its portion of the state diagram is shown in Figure 2.7. The precautions of safe arc welding and proper machining could have much worse results for the final product and operators than the scanning system could. The scanning station can indirectly cause errors in the other systems if a mediocre scan is used as reference geometry for printing or machining paths, so it is important to be sure that the scan is performed properly. The propeller is scanned by a structured light scanner, which is secured to a fixed location next to the linear track. The rotary table will need to rotate the part in front of the camera, while it takes intermittent scans in order to create a full 3D mesh of the geometry. This scan path algorithm will need to be created in accordance with scan targets associated on the part pallet. The scan targets will be taken from the part pallet rather than the printed or machined geometry in order to simplify the process. Taking the reference from a part pallet that doesn't change is better than scanning based on a geometry that will change between uses of the system.

The initial creation of the scan algorithm will take longer than a typical scan to ensure that all sides and angles of a reference propeller are able to be scanned and fit together properly in the created CAD file. Once this algorithm is known, propellers that are scanned will be similar enough in geometry that the scan algorithm will able to plug and play with all propellers after its initial creation. If it turns out that there are holes left in the scans of the propeller, it it simple to add more positions to be scanned throughout the algorithm to fill in the holes. Since a propeller has patterned geometry to it, once a sufficient path for one blade is known, it can be repeated for all the other blades.

In this project, propellers are created at a reduced scale to prove the effectiveness of the process. Larger propellers will need a more thorough algorithm in order to scan all the data of the underside of the blades. Aside from calibration of the scanner and sufficient scan targets, not much needs to be checked in order to enable that the system to scan the

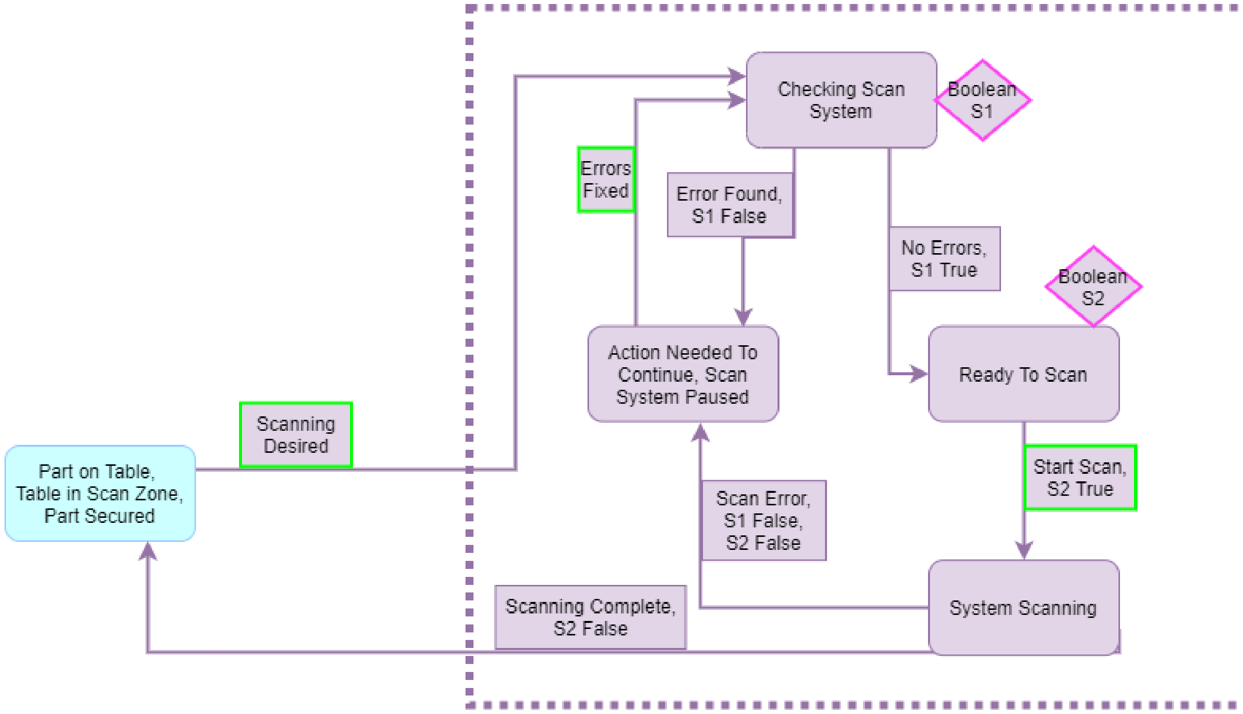


Figure 2.7: Scanning subsection of the state diagram

part, just general safety checks and ensuring that the part pallet does not unfasten from the rotary table. There is a designated metrology computer that is connected to the 3D camera that allows easy real-time monitoring of the results, which an operator can use to check for lapses and shortcomings in the scan algorithm that was used. This station also allows for the exporting of scan files to other systems.

2.3.4 Printing System

The printing system is monitored through the supervisory control PC as well as thermal cameras placed inside the cell. The welding equipment is able to monitor and log the process voltage and current, and thermal cameras allow for visual logging of the process. The printing subsection of the system state diagram is shown in Figure 2.8. Unlike the machining station where all system safety is more-or-less internal to the CNC machine itself, the printing system has more external safety needs that must be met. This is handled by the presence of weld curtains, thermal cameras, and overhead ventilation that are installed around the robot that is fitted with welding equipment. The system booleans that enable printing need to be more closely monitored due to their complexity and variation. Weld wire that is used to deposit material onto the part is a consumable that comes in spools, fastened to the weld power supplies. The amount of wire remaining during printing needs to checked by an operator to reduce the likelihood that a wire spool runs out unexpectedly. If the welding equipment is trying to deposit beads without any remaining wire, the final part quality will be unsatisfactory. It is also possible for the wire to jam inside the feed head, which needs to be monitored by an operator. The wire could also be fed too quickly causing long arcing which affects final geometry and quality. There is also the concern of welding current and voltage, which fluctuates during the welding process. The welding power supply is designed such that welding parameters do not go outside of expected bounds during operation, but they still need to be monitored and logged in order to verify quality results. A change of welding process conditions during the operation can cause various problems. This can cause internal voids that can shorten fatigue life and cause crack propagation. Excessive variation of welding parameters can also affect inter-layer fusion and cause breakages and failures during machining [7]. The welding equipment is able to keep a designated waveform

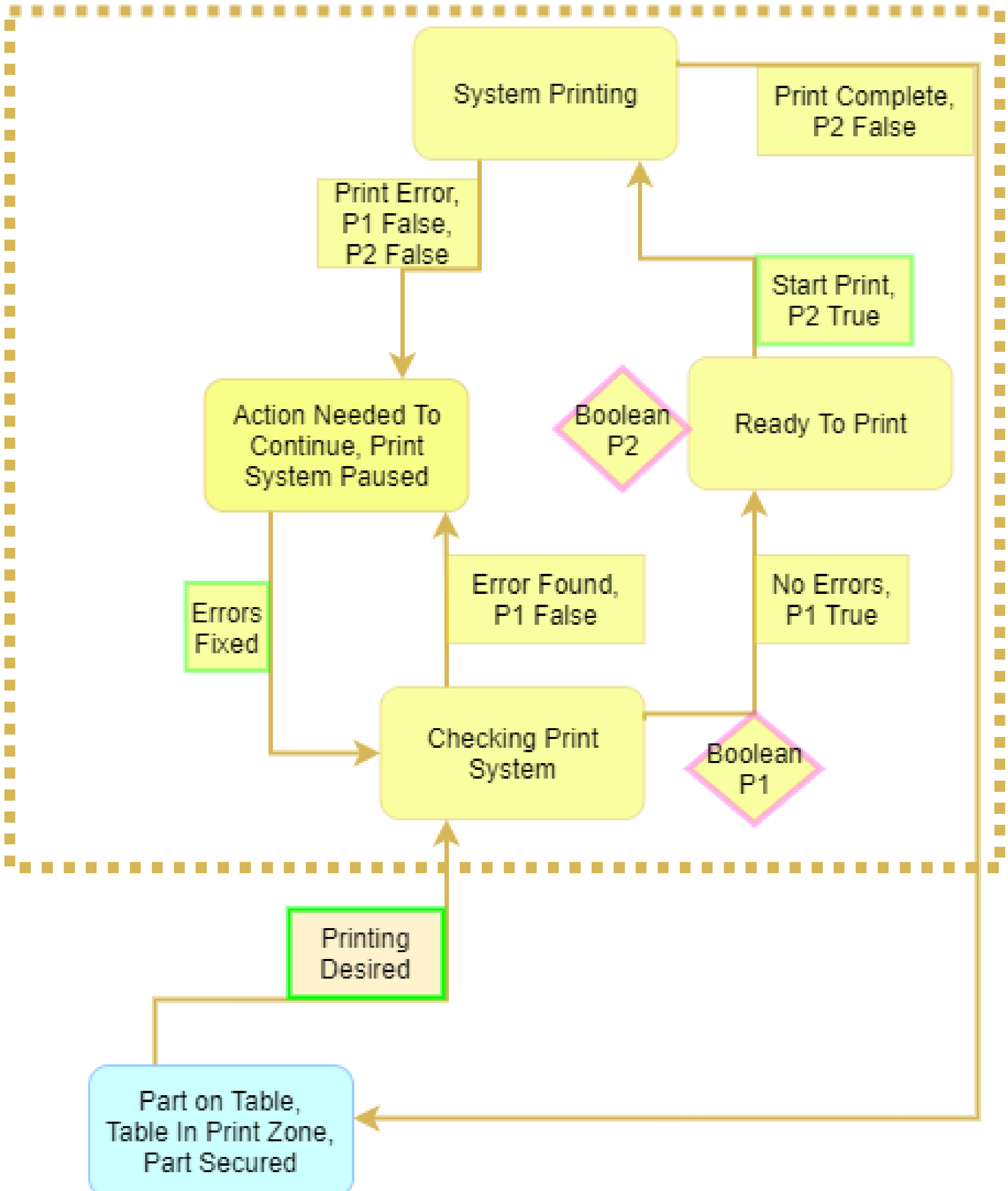


Figure 2.8: Printing subsection of the state diagram

of voltage and current based on pre-defined variables, which should minimize this variation during the process. It is desirable for the final printed part be created with consistent parameters in order to keep uniform quality. Every system of this cell has important factors that need to be considered and properly monitored throughout all operations in order to create desired results in a safe manner.

Chapter 3

Data Flow & Control Simulation

3.1 Data Flow Diagram

It is necessary to document how all of the different systems communicate and connect with one another in order to aid with system integration. The data flow diagram created for this system is shown in Figure 3.1. This diagram works as a list of all of the components of the system, as well as a list of how they are connected to one another and the communication protocols used for either actuation or data transmission. Each component in the diagram serves its own purpose, but the main pieces that primarily control the whole work cell are the supervisory control PC, the master of the system, and the PLC, the primary slave of the system. Both of these primary components communicate back and forth through a Modbus RS-485 cable. Modbus is a serial communication protocol that is particularly helpful with control systems and PLC implementation [17]. Rather than having the PLC be running its own monitoring program synchronously with the supervisory control PC running its own control loop, Modbus enables the creation of tags to the different PLC input and output boards that can be used as an extension of variables that the PC has access to. This also allows the PLC to have its own internal logic to send multiple output signals based on input it receives from the control PC. Control systems have been created using Modbus for a variety of applications [2, 21, 11]. Previous studies used Modbus for real time process control, supervisory control and a control system directly communicating between a PLC

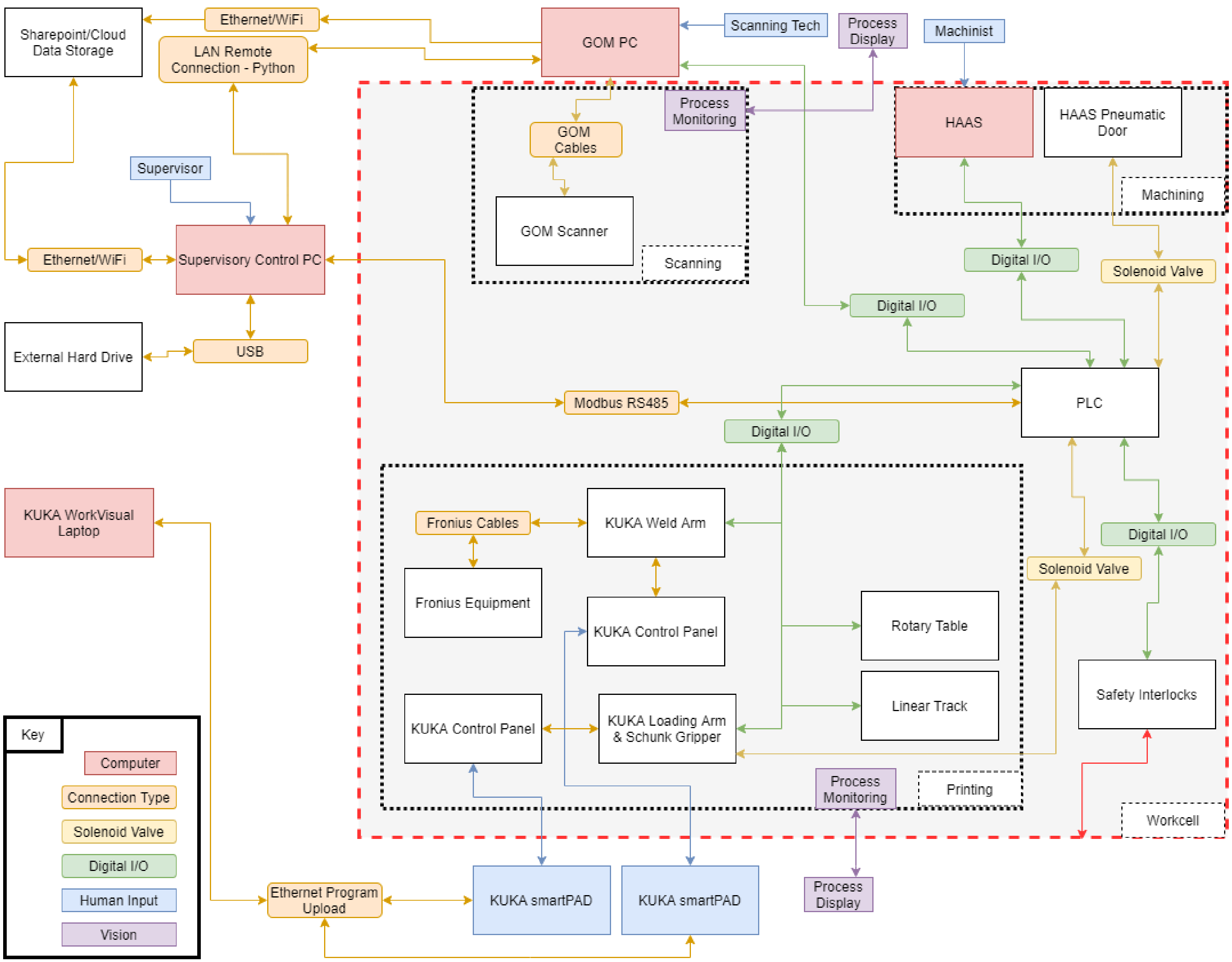


Figure 3.1: System data flow diagram

and LabVIEW. These projects aren't directly related to hybrid manufacturing, but they are a good proof of concept that this technology has been used for similar applications of control.

The supervisory PC and PLC allow most of the operations in the work cell, but there are other communications to consider. Cameras are other key components to this system, especially in the printing station. With the presence of eye-damaging light and weld curtains, the welding process is unable to be directly visually monitored by a system operator. The use of thermal cameras connecting the inside and outside of the cell is crucial to display and monitor the welding process safely. The supervisory control PC is the hub of the work cell where as much relevant operation data needs to be visible or able to be accessed since it is the master of the cell and the only component that bridges the communication gap through the safety barriers. Just like other representations of this system, the data flow diagram is grouped by primary operations that are performed of printing, scanning and machining. This model also better represents communications through the safety barriers of the system. It is important that all actions that are performed on the inside of the safety barriers are able to be viewed safely from the outside of the system.

3.2 LabVIEW

With all of the different components of this system needing to work together in order to create the finished product, it is desired to have an interface that all of these components communicate through at one centralized location outside the safety barriers of the work cell. LabVIEW is a graphical programming language created by National Instruments. This software is very beneficial for data collection and lab monitoring due to its extensive range of modules and functions that can be used out of the box. The weld printing process has been monitored and logged using LabVIEW for various other additive manufacturing research projects with the University of Tennessee Knoxville using robotic WAAM [12, 6, 7, 15]. These projects utilized either Metal Inert Gas (MIG) or Tungsten Inert Gas (TIG) welding for experimental additive manufacturing. Some of these studies were more focused on the CAD to part creation of weld paths, which is the basis of how the propellers are able to be printed. The work done to create algorithms using non-gravity aligned welding is crucial for printing complex geometry such as a naval propeller. All of this previous work is the foundation that was necessary to create weld-based additive processes with KUKA robotics and a LabVIEW interface, the same technology that is used for this hybrid manufacturing system.

LabVIEW is helpful for running and data logging for the weld process, which requires control of asynchronous activities. This asynchronous capability and established use is what influenced the decision to have LabVIEW be the primary software implementation of the supervisory control system on the control PC. The previously created arc welding programs can be adapted for this cell's additive component for monitoring and quality control. The only change in weld process control is the use of different welding equipment, but control of the process through the KUKA control panel is the same. LabVIEW is also able to utilize an object oriented programming schema to create objects for the different cell subsystems and their relevant data. Object oriented programming (OOP) was used in the LabVIEW implementation of this system model. Most of these objects are either a boolean or a group of booleans, but an object oriented structure makes creation more easily adaptable and modular for the addition of equipment and data. LabVIEW's control system toolbox is also able to designate variables based on Modbus tags in order to easily communicate to and from ports on the PLC connected through RS-485 serial connection.

3.2.1 LabVIEW State Machine and HMI

With all data, connections and states laid out; it is possible to create a fully functional state machine in LabVIEW in order to simulate the operations of the work cell without being fully hooked up to equipment. The goal of this implementation is to create a "digital logic twin" of the system that works as a copy of the cell that mimics the physical system and displays relevant data as it is running. The ability to simulate system operations is useful in order to understand work timelines and process order, enabling certain actions to be performed according to what state the system is in. It is important that this state machine is able to run on its own without being hooked up to equipment for safe, intermittent testing. The LabVIEW programming environment works through 2 windows, the front panel and the block diagram. The block diagram is where all of the graphical programming is implemented, and the front panel is how an operator interacts with the program while it is running. The front panel of a LabVIEW program is a HMI since it is the user interface that works as the bridge between the control system and human input. The front panel of the running LabVIEW script displays relevant technical data and enabling switches according to operational needs. The front panel HMI of the created state machine is shown in Figure 3.2. The logic of enabling operations is according to the state diagram that was discussed in Chapter 2, and the code used in the block diagram is shown in Appendix A. The HMI of this system followed the pattern of other system models laid out previously, where it is grouped according to primary system functions.

3.2.2 LabVIEW General Overlay

With this state machine and all of the complicated operations that can be performed in one use of this cell, it is also desirable to have a more generic user interface (UI) display that only shows the status of the system without having any human input to it. Both of these interfaces will be running simultaneously, with the general overlay being able to quickly display generic states of the system at a glance. This allows multiple operators to be able to see the status of production without interrupting another operator who is currently overseeing their system's tasks. It also creates a more streamlined way to display more key system booleans quickly without worrying about some of the more intricate states that are not as useful to always see. This output only general overlay is depicted in Figure 3.3 and is primarily used for casual system monitoring without an input for control or state changing. The main data that is relevant to display for a generic overlay is what main state the system is currently in. These main states are the 3 primary operations of this system of printing, scanning and machining, as well as the transitions between them. There are also dialogue boxes that display text based data more relevant to the inner workings of a state, as well as general system diagnostics to ensure that everything is running as intended.

Figure 3.2: LabVIEW state machine front panel

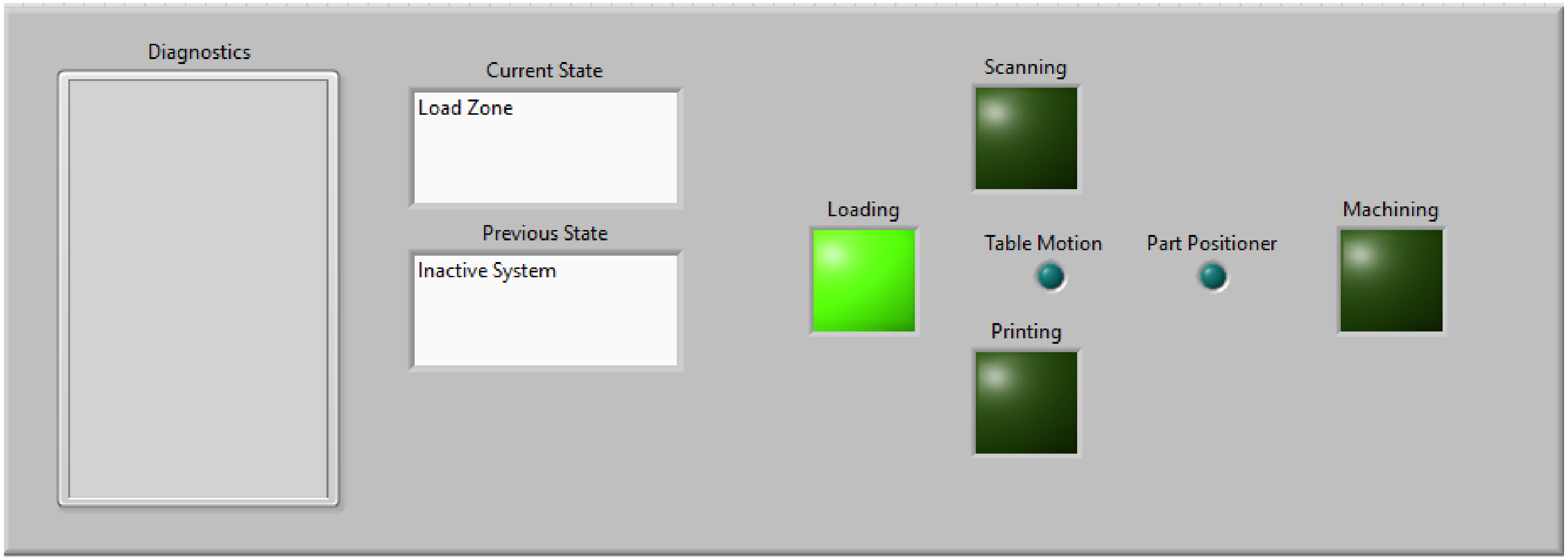


Figure 3.3: LabVIEW general overlay

Chapter 4

Conclusion & Future Work

The goal of the work presented in this thesis was to design and simulate a control system structure for a hybrid manufacturing work cell. There are many different modelling techniques and protocols that can be used to create boolean control structures for a control system, each with their own advantages and disadvantages. Mathematic set theory is a good starting place to define state transition logic based on various state booleans, and can be easily transformed into corresponding ladder logic for direct implementation into a PLC program structure. Due to continuous system operations and desired implementation properties, LabVIEW was chosen as the program used as the master of the control system with the PLC as one of the system slaves. This implementation was successfully created and simulated in LabVIEW as a digital logic twin of the hybrid manufacturing work cell before all of its equipment arrived and was installed in order to have a good baseline to begin system integration. An effective structural implementation enables more ease of testing in order to iterate upon the improvement of the hybrid manufacturing process that is the main focus of several research teams focused on additive manufacturing, metrology and machining.

4.1 Future Physical Implementation

The future of this control system structure is full integration of all system components for monitoring and logging of system data. Equipment will be integrated according to the diagrams previously laid out. LabVIEW also has several toolboxes that are useful for sending and receiving data from different stations in the work cell. Various PLC modules are also able to be used to send and receive a desired number and type of signals from different stations. This will help secure that data is not lost in translation between stations. It will also enable a file organization hierarchy to be created for ease of use. Physical testing of controls using LabVIEW and the PLC ladder logic has been accomplished for components of the safety system as well as the pneumatic solenoid valves for component actuation. This implementation's LabVIEW front panel can be seen in Figure 4.1, while the ladder logic and code side of this system can be seen in Appendix C. The physical PLC used for this setup is shown in Figure 4.2, and is the same PLC that will be run in tandem with the supervisory control PC to monitor the hybrid manufacturing cell.

Figure 4.1: LabVIEW safety system front panel

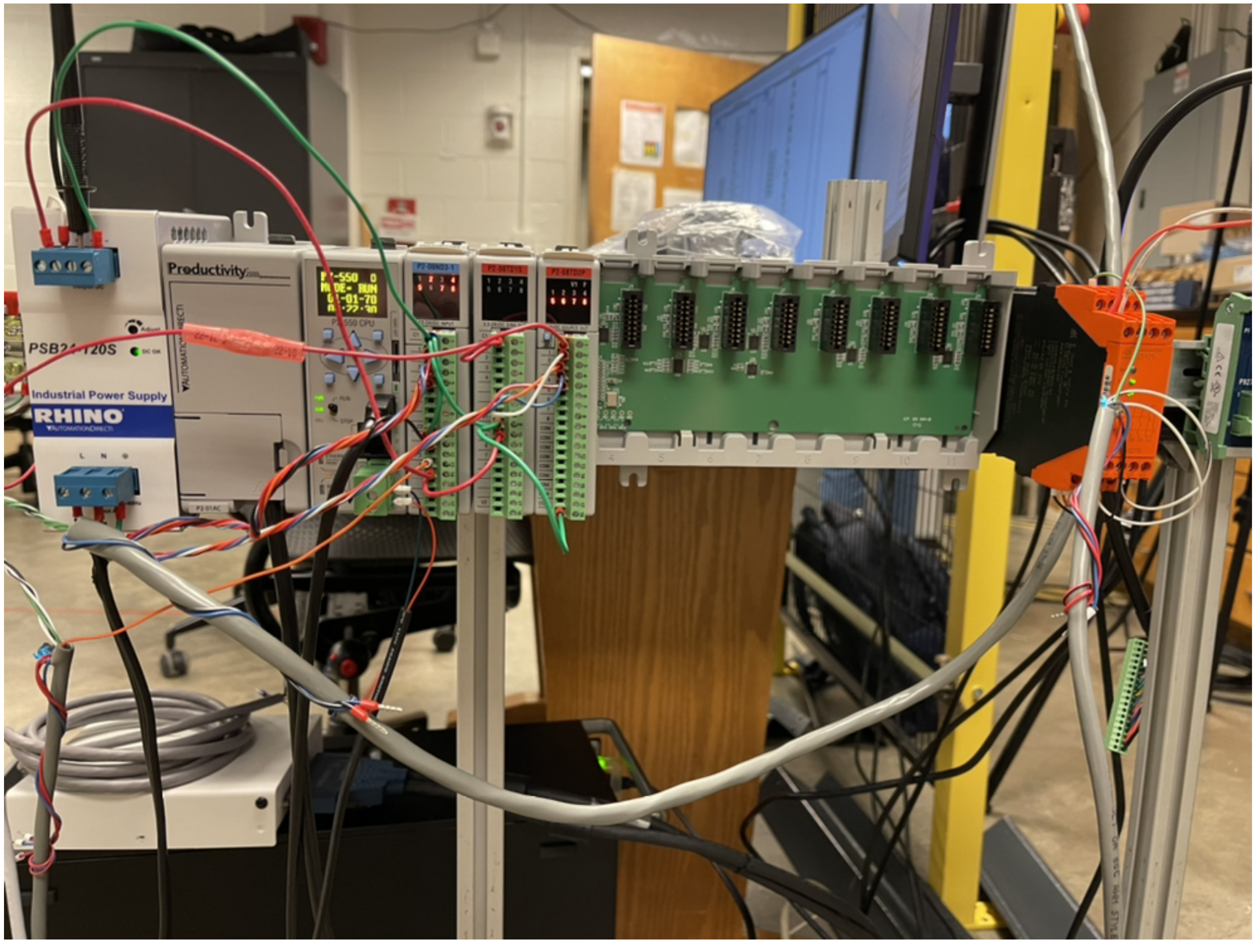


Figure 4.2: The PLC used for this system

Bibliography

[1] Alphonsus, E. R. and Abdullah, M. O. (2016). A Review on the Applications of programmable logic Controllers (PLCs). 13

[2] Ashtekar, A., Parvat, B., No, C. K., and 2013, U. (2013). Application of Modbus to Communicate the PLC and Lab VIEW for Real Time Process Control. *academia.edu*, 1(2):41–45. 40

[3] Carlton, J. (2018). *Marine propellers and propulsion*. 4th edition. 1

[4] Erickson, K. T. (1996). Programmable Logic Controllers. *IEEE Potentials*, 15(1):14–17. 13

[5] Koutsoukos, X., Antsaklis, P., of the . . . , J. S. P., and 2000, U. (2000). Supervisory Control of Hybrid Systems. *Proc. IEEE*, 88(7):1026–1049. ix, 7, 9, 11, 12

[6] Mcneil, J. L., Hamel, W. R., Penney, J., Nycz, A., and Noakes, M. (2019). Framework for CAD to Part of Large Scale Additive Manufacturing of Metal (LSAMM) in Arbitrary Directions. Technical report. 2, 4, 42

[7] McNeil, J. L., Sisco, K., Frederick, C., Massey, M., Carver, K., List, F., Qiu, C., Mader, M., Sundarraj, S., and Babu, S. S. (2020). In-Situ Monitoring for Defect Identification in Nickel Alloy Complex Geometries Fabricated by L-PBF Additive Manufacturing. *Metall. Mater. Trans. A Phys. Metall. Mater. Sci.*, 51(12):6528–6545. 1, 37, 42

[8] Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proc. IEEE*, 77(4):541–580. 5

[9] Netto, R. and Bagri, A. (2013). Programmable Logic Controllers. Technical Report 11. ix, 13, 17

[10] Newton, L. R. (2000). Data-logging in practical science: research and reality. *Int. J. Sci. Educ.*, 22(12):1247–1259. 4

[11] Panchal, P., Patel, A., On, J. B. . I. C., and 2015, U. (2015). PI control of level control system using PLC and LabVIEW based SCADA. *2015 Int. Conf. Ind. Instrumentaion Control*. 40

[12] Penney, J. J. and Hamel, W. R. (2019). USING NON-GRAVITY ALIGNED WELDING IN LARGE SCALE ADDITIVE METALS MANUFACTURING FOR BUILDING COMPLEX PARTS. Technical report. 42

[13] Peterson, J. L. (1977). Petri Nets. *ACM Comput. Surv.*, 9(3):223–252. ix, 5, 6

[14] Ramadge, P. J. and Wonham, W. M. (1986). Modular Supervisory Control of Discrete Event Systems. pages 202–214. 22

[15] Robertson, S., Penney, J., McNeil, J. L., Hamel, W. R., Gandy, D., Frederick, G., and Tatman, J. (2020). Piping and Pressure Vessel Welding Automation through Adaptive Planning and Control. *JOM*, 72(1):526–535. 42

[16] Shuang, L. and Xifan, Y. (2008). Multi-level modeling for hybrid manufacturing systems using arena and MATLAB. In *Proc. - 2008 Int. Work. Model. Simul. Optim. WMSO 2008*, pages 155–159. IEEE Computer Society. ix, 7, 10

[17] Thomas, G. (2008). Introduction to modbus serial and modbus tcp. Technical report. 40

[18] Uzam, M., Jones, A., IEEE, N. A. P. ., and 1996, U. (1996). Conversion of Petri Net Controllers for Manufacturing Systems into Ladder Logic Diagrams. *Proc. 1996 IEEE Conf. Emerg. Technol. Fact. Autom. ETFA '96*. 5

[19] Wong, K. V. and Hernandez, A. (2012). A Review of Additive Manufacturing. *Int. Sch. Res. Netw. ISRN Mech. Eng.*, 2012. 1

[20] Wonham, W. (2013). Supervisory Control of Discrete-Event Systems. *Encycl. Syst. Control*. 13

[21] Yang, X., Wu, J., Zhang, K., Zhang, P., and Xie, C. (2016). Research on Machine Vision Based Networked Supervisory Control System. In *2016 UKACC Int. Conf. Control. UKACC Control 2016*. Institute of Electrical and Electronics Engineers Inc. ix, 12, 14, 15, 40

[22] Zhou, M. and Twiss, E. (1998). Design of Industrial Automated Systems Via Relay Ladder Logic Programming and Petri Nets. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, 28(1):137–150. 5

Appendices

A LabVIEW Code

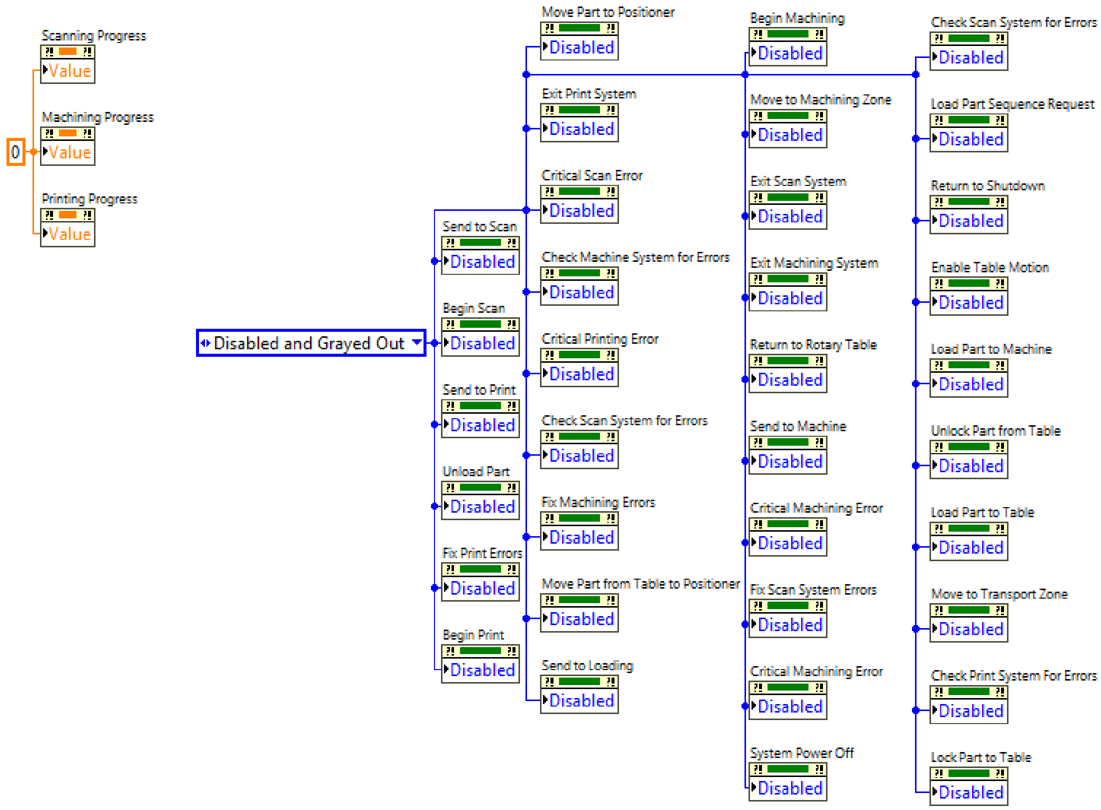


Figure 3: LabVIEW block diagram value initialization

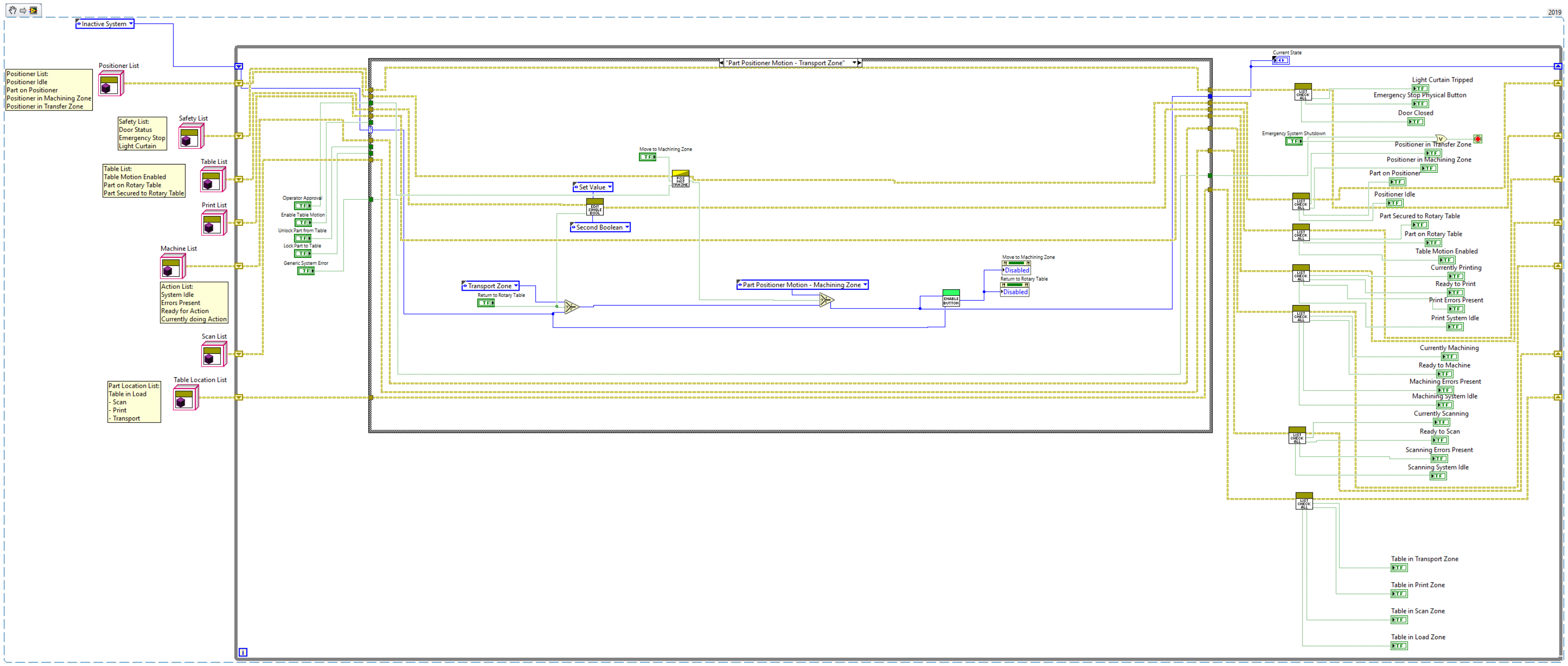


Figure 4: LabVIEW block diagram overview

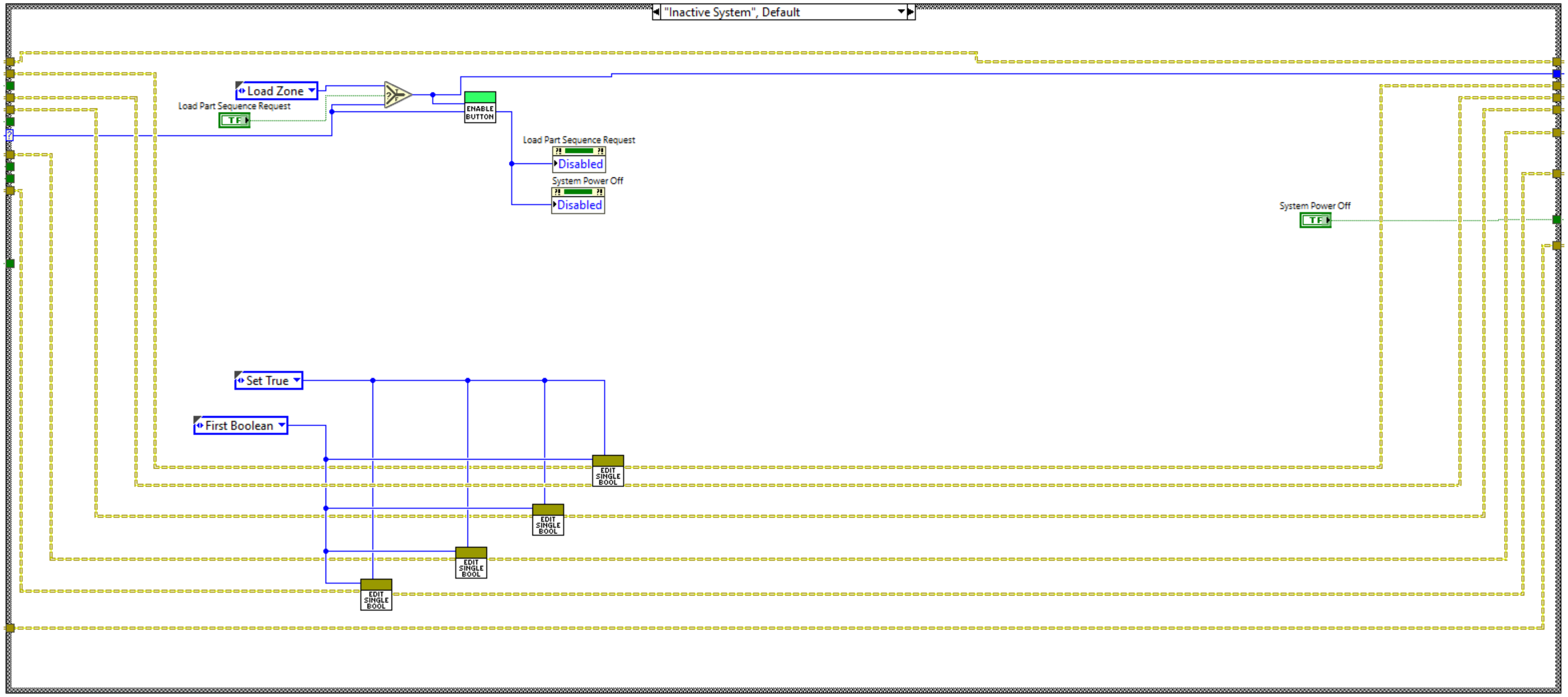


Figure 5: LabVIEW block diagram inactive system

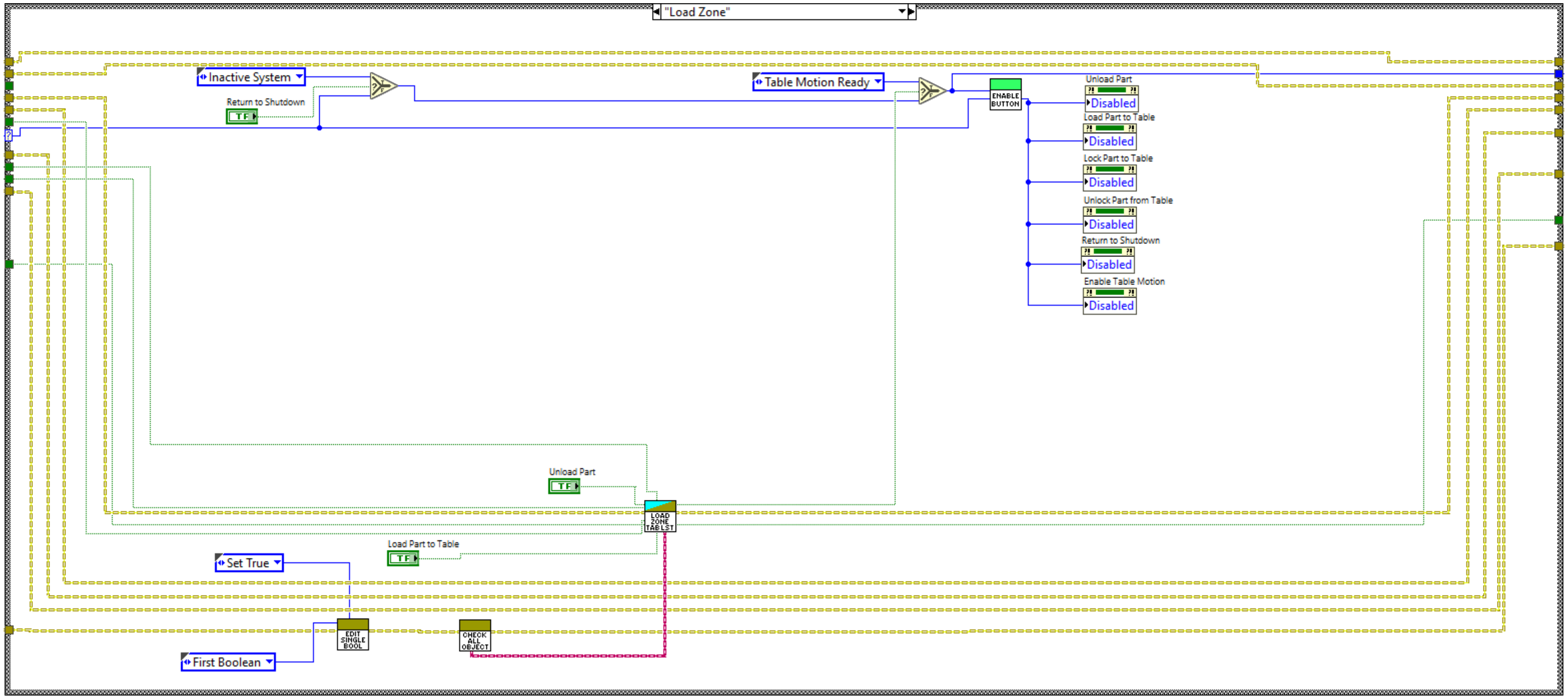


Figure 6: LabVIEW block diagram load zone

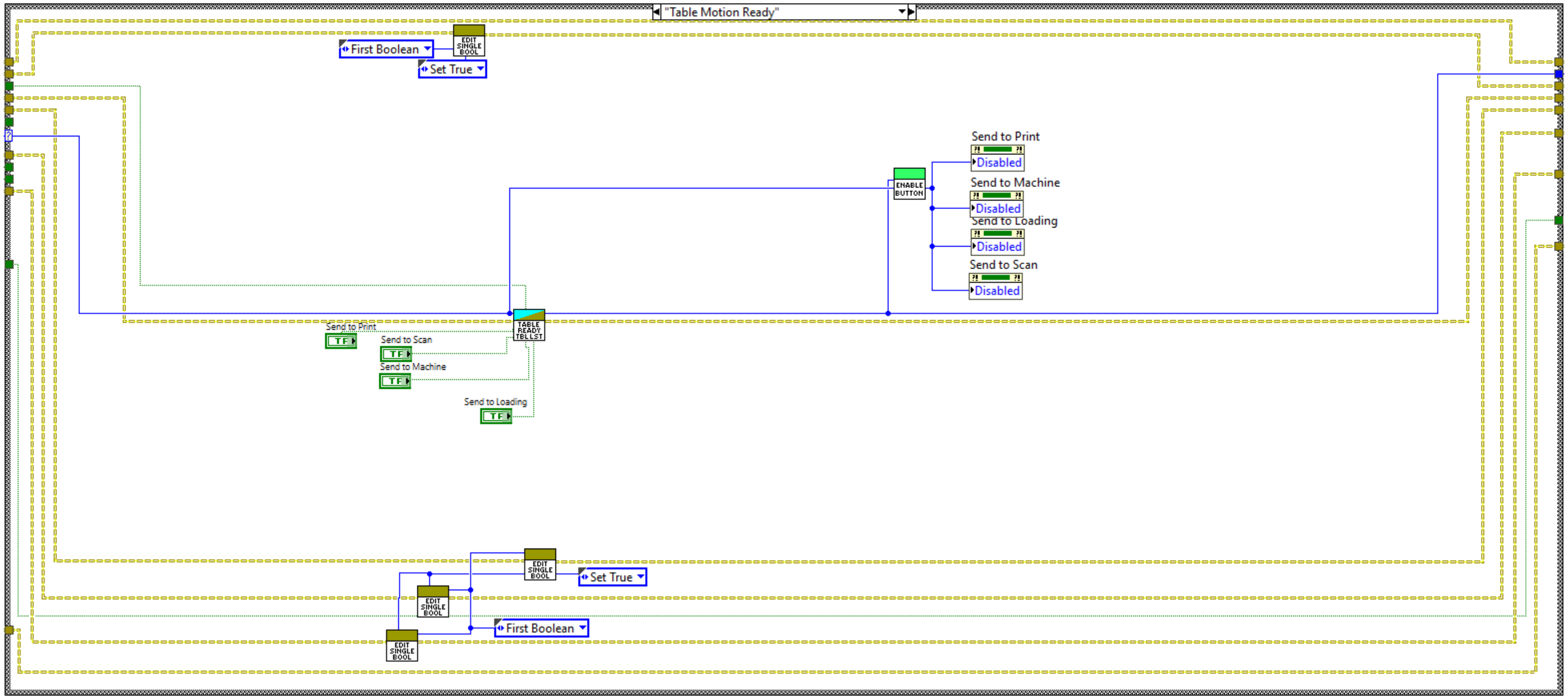


Figure 7: LabVIEW block diagram table motion ready

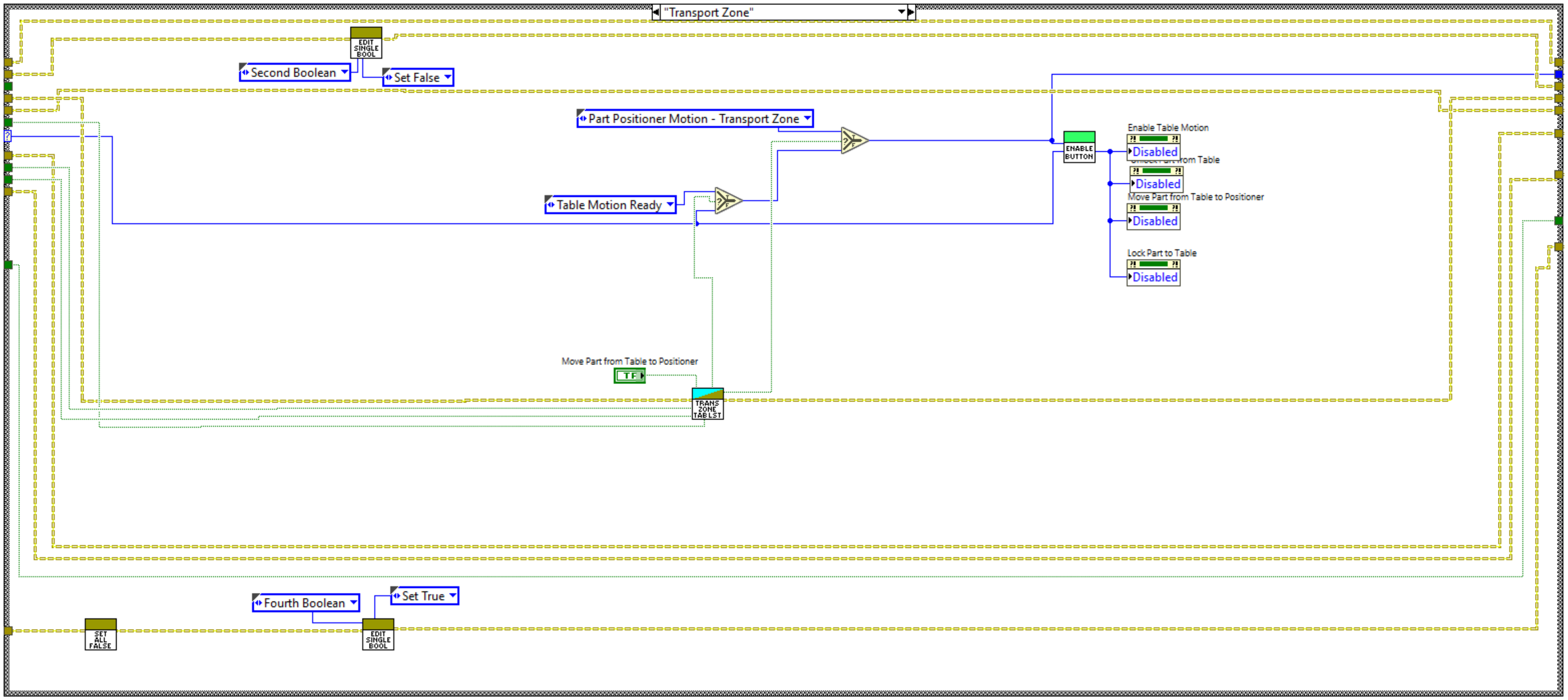


Figure 8: LabVIEW block diagram transport zone

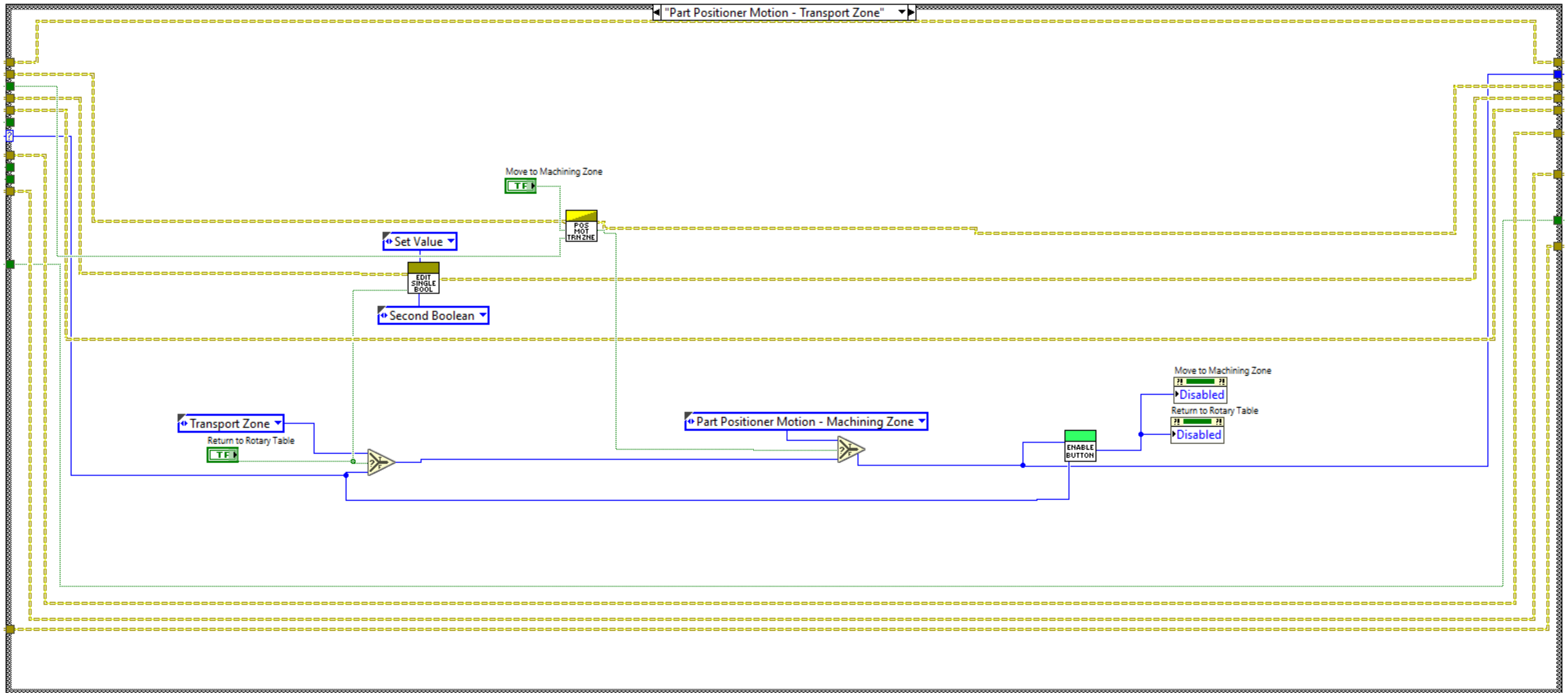


Figure 9: LabVIEW block diagram part positioner in transport zone

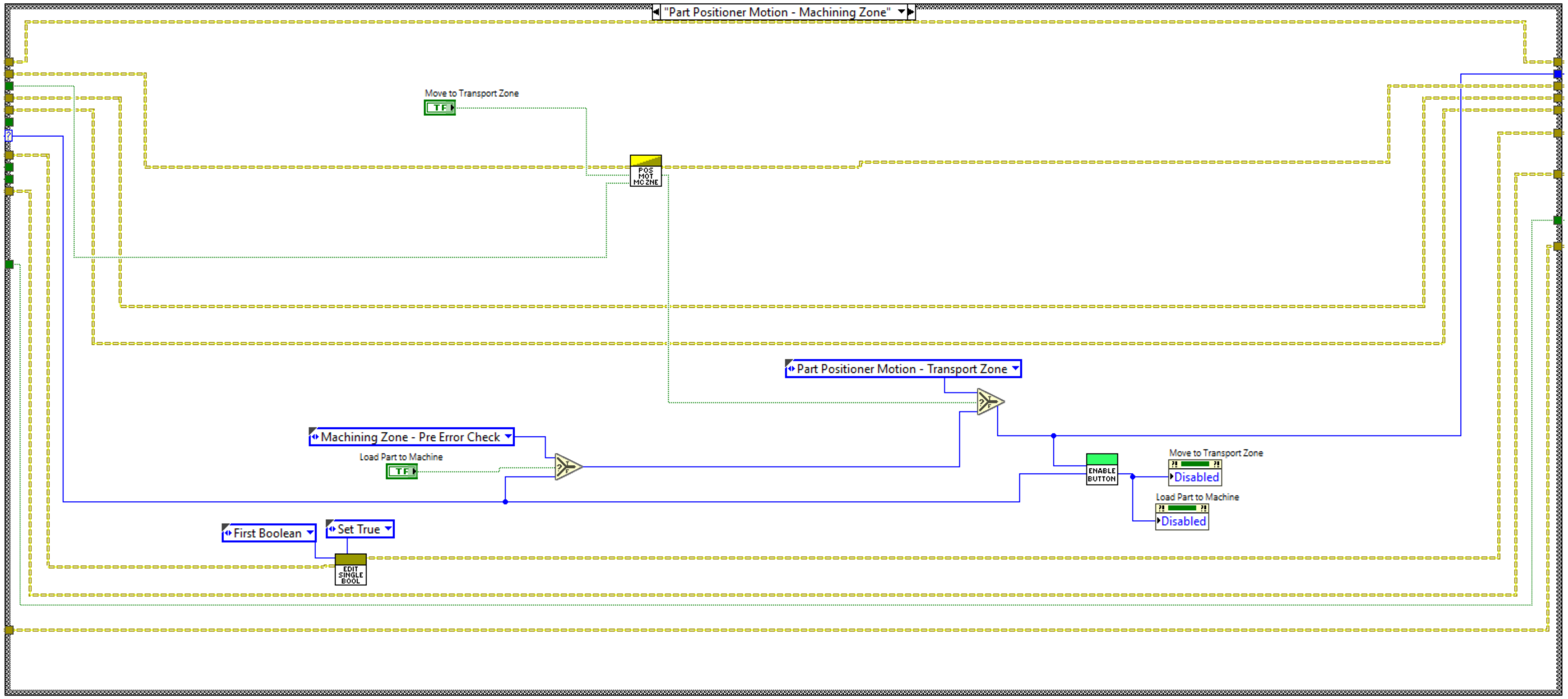


Figure 10: LabVIEW block diagram part positioner in machining zone

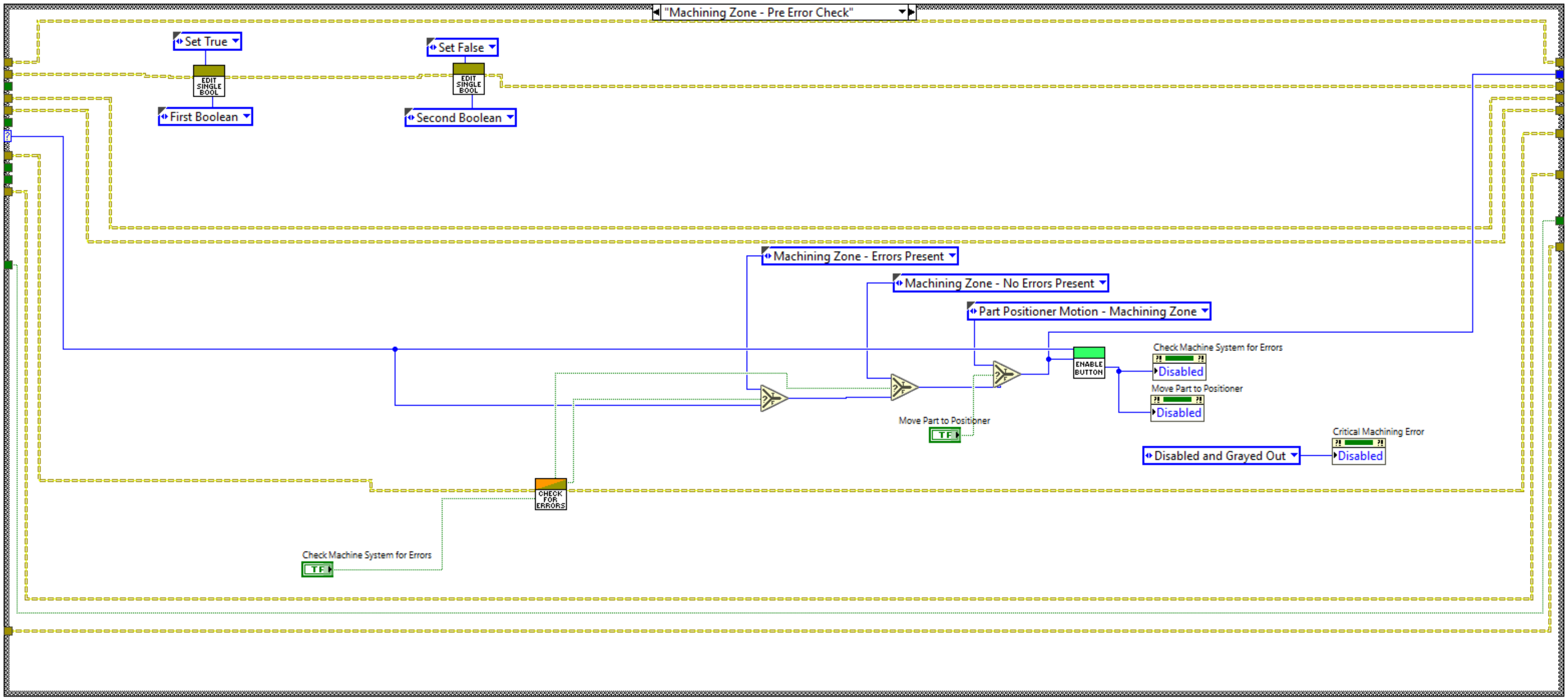


Figure 11: LabVIEW block diagram machining zone before error check

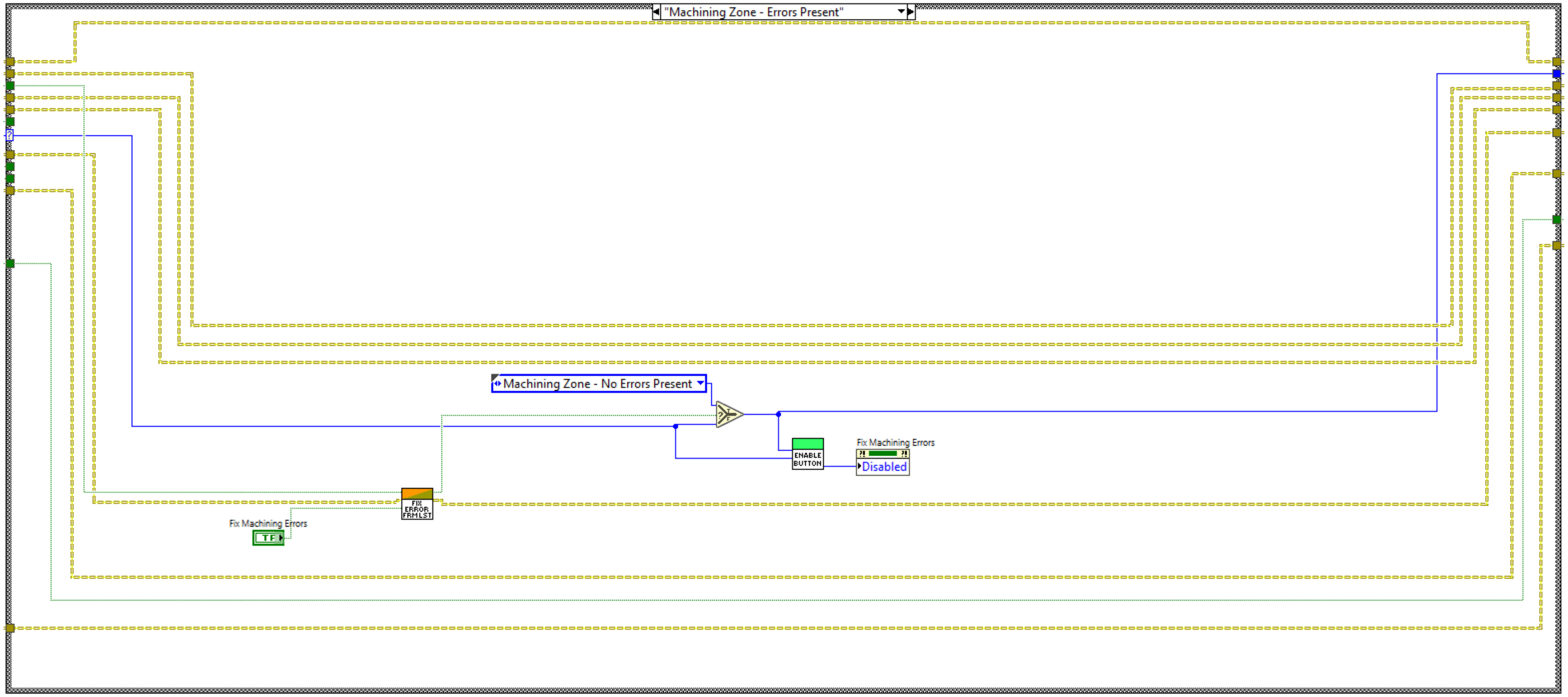


Figure 12: LabVIEW block diagram machining zone errors present

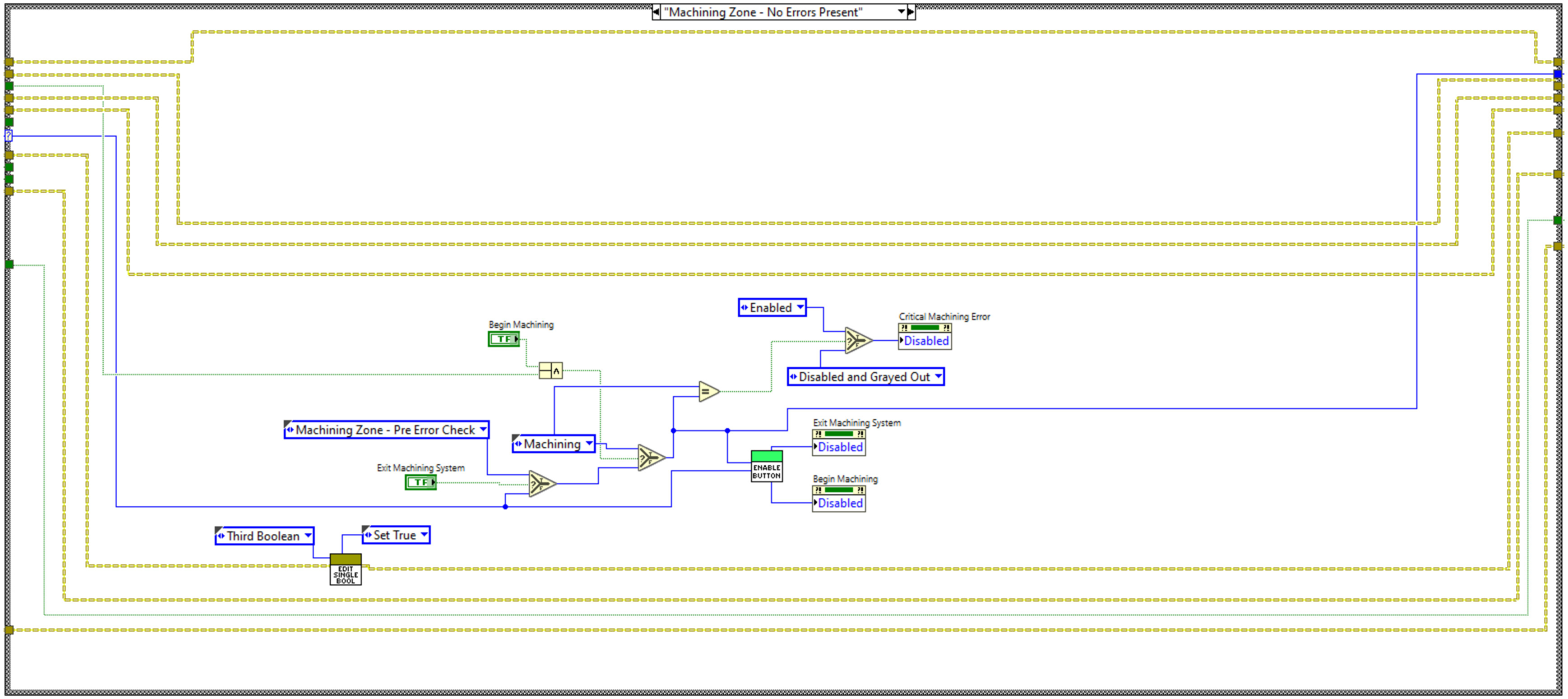


Figure 13: LabVIEW block diagram machining zone no errors present

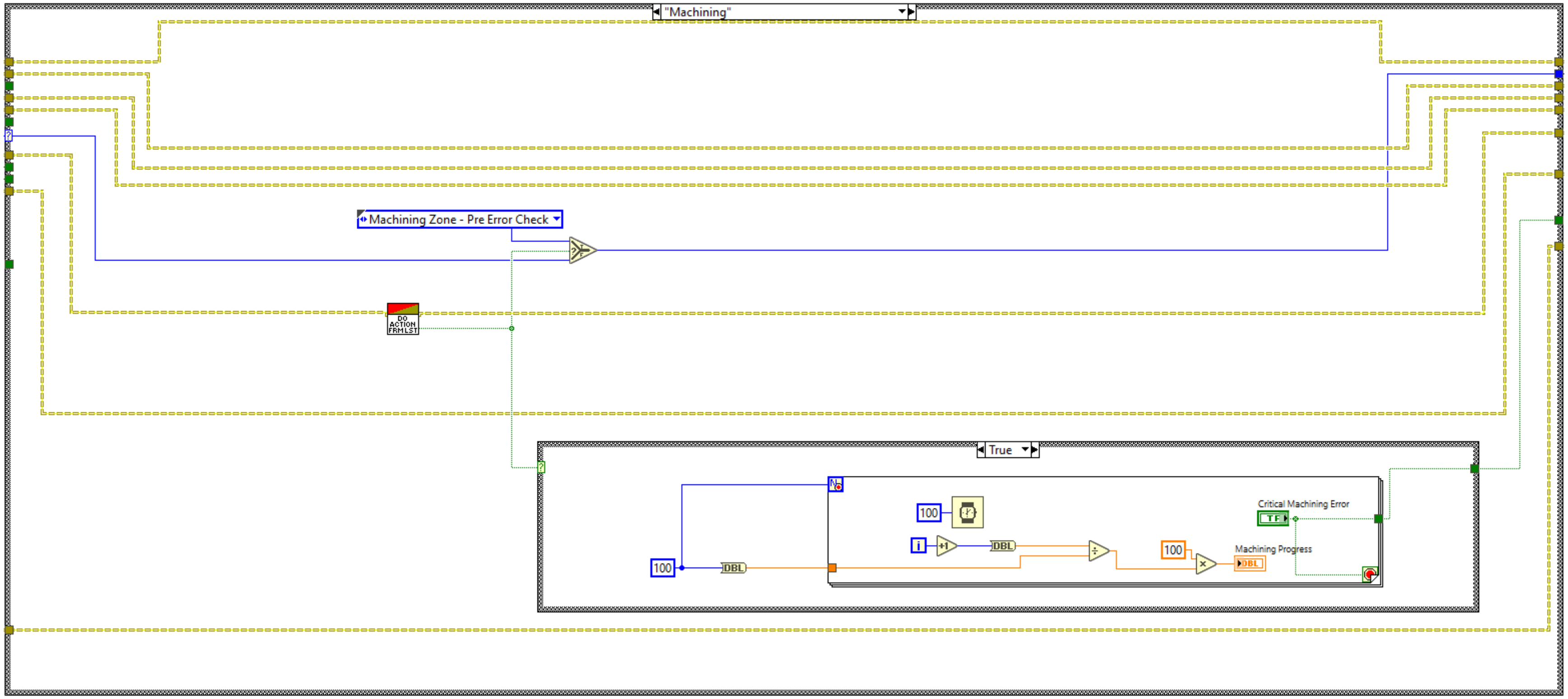


Figure 14: LabVIEW block diagram machining operation

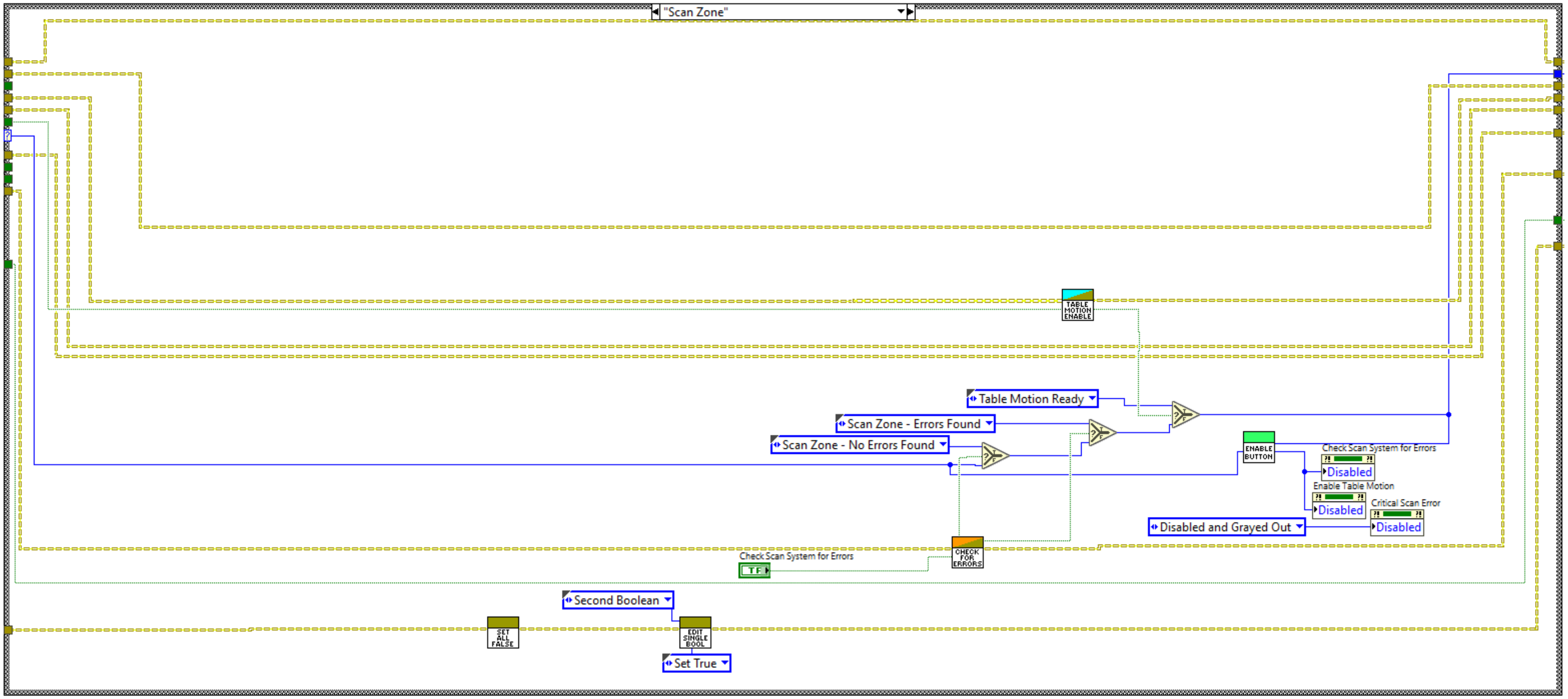


Figure 15: LabVIEW block diagram scan zone before error check

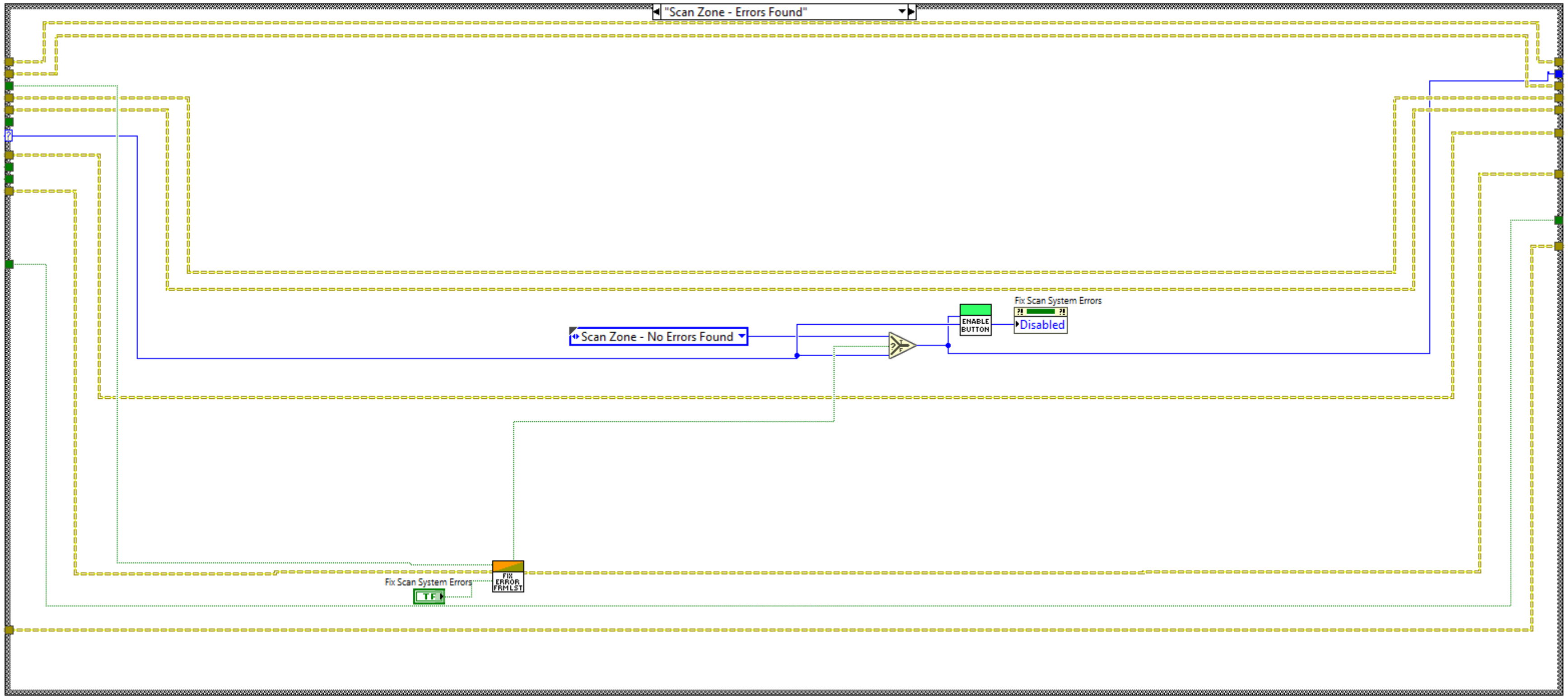


Figure 16: LabVIEW block diagram scanning zone errors present

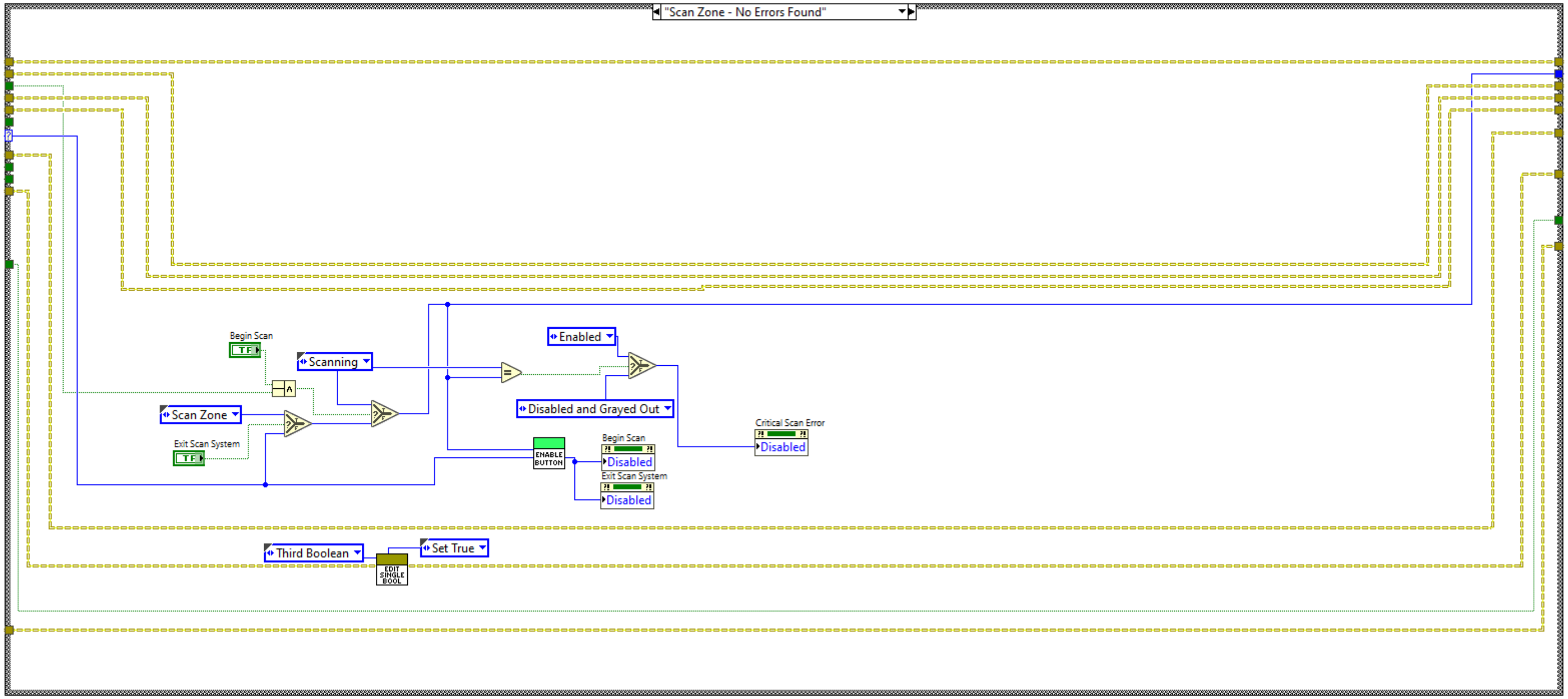


Figure 17: LabVIEW block diagram scanning zone no errors present

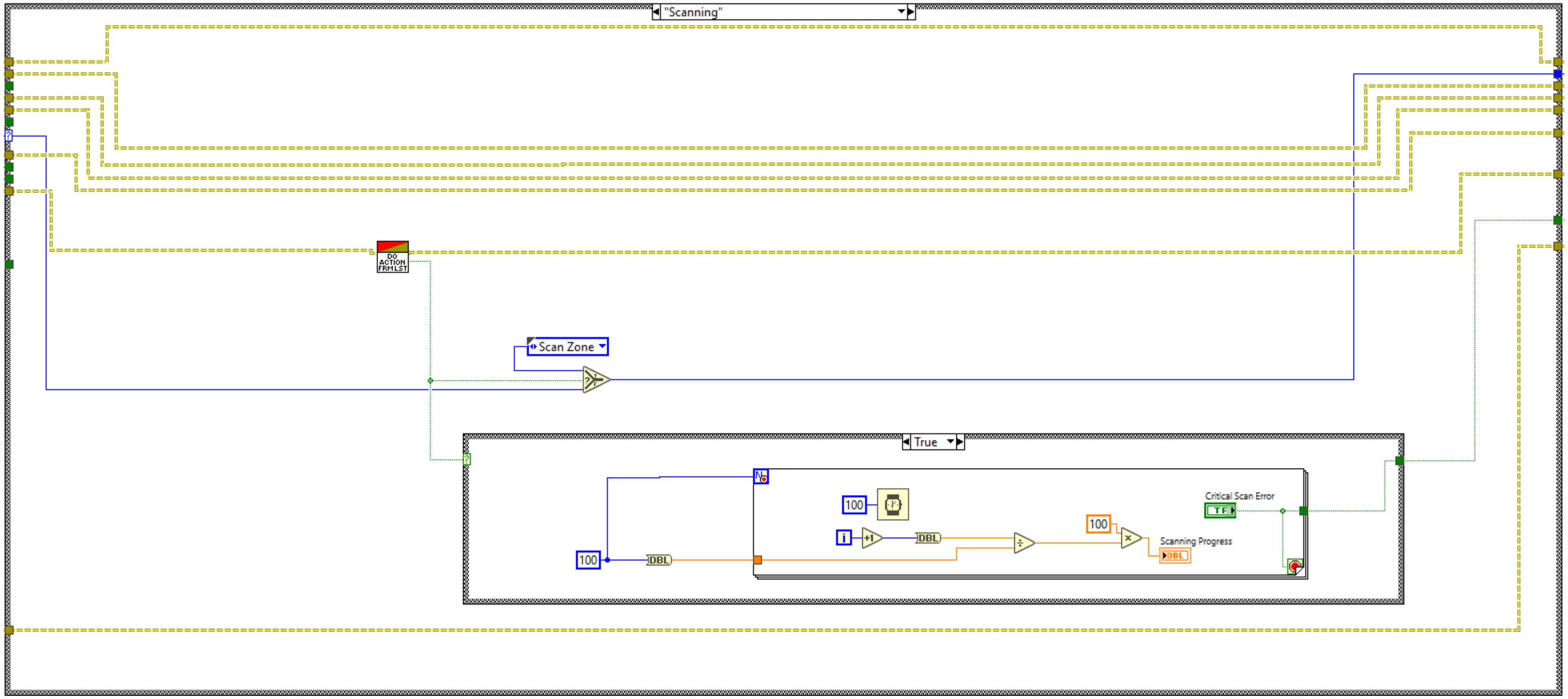


Figure 18: LabVIEW block diagram scanning operation

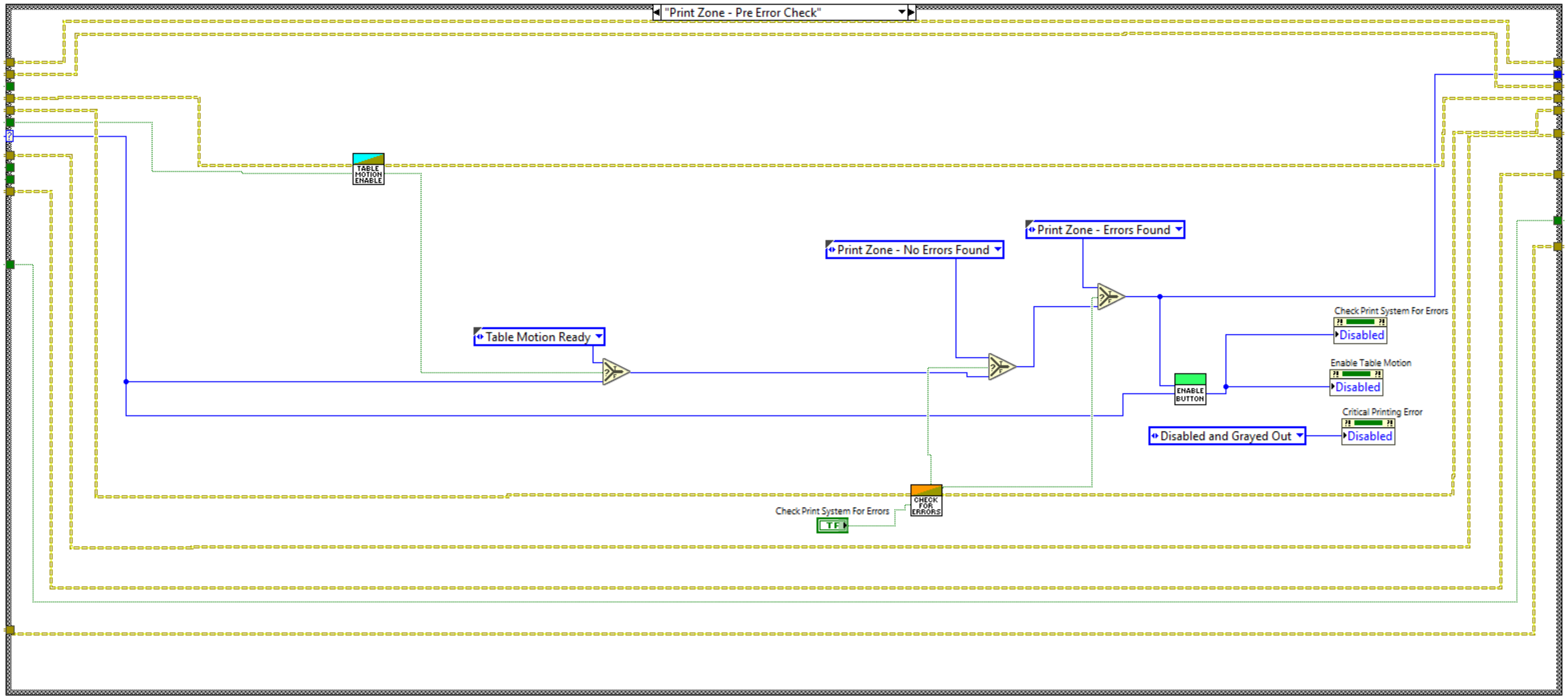


Figure 19: LabVIEW block diagram printing zone before error check

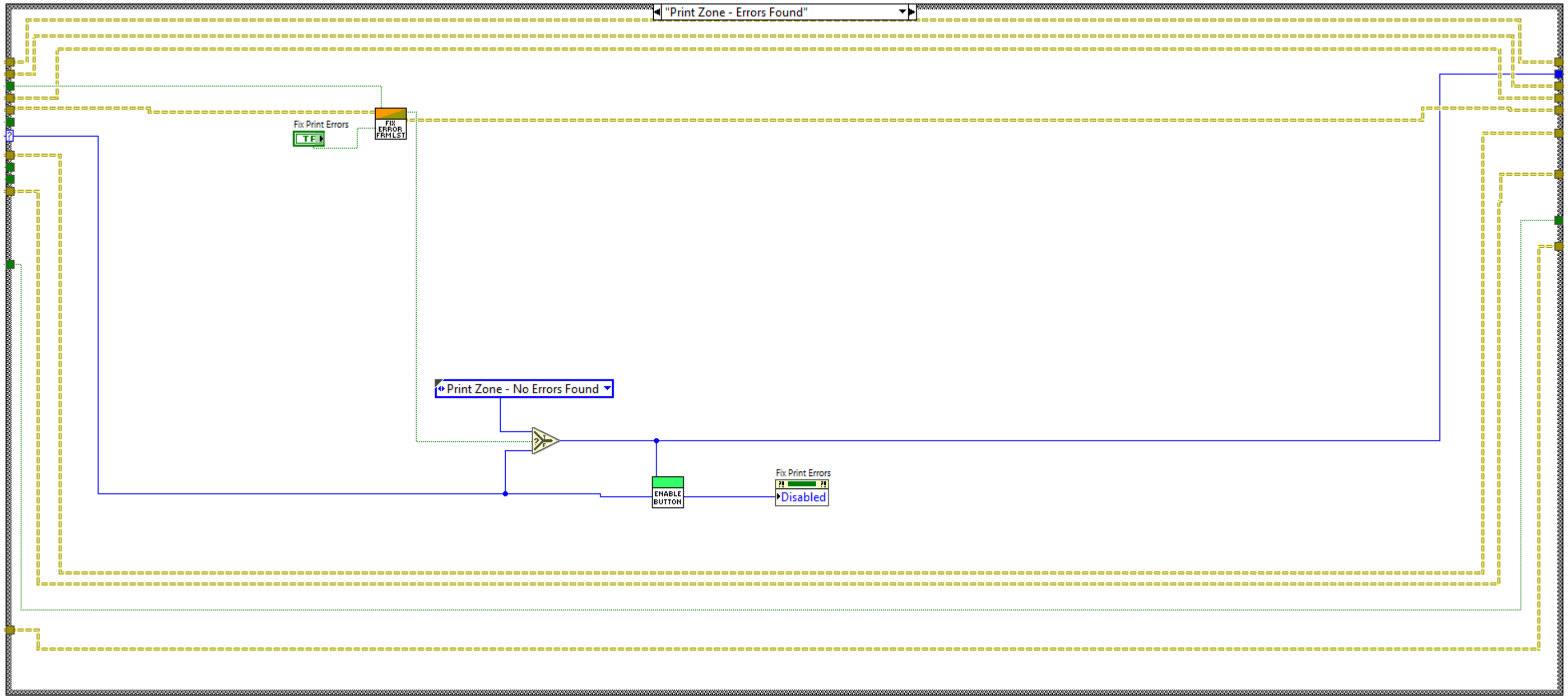


Figure 20: LabVIEW block diagram printing zone errors present

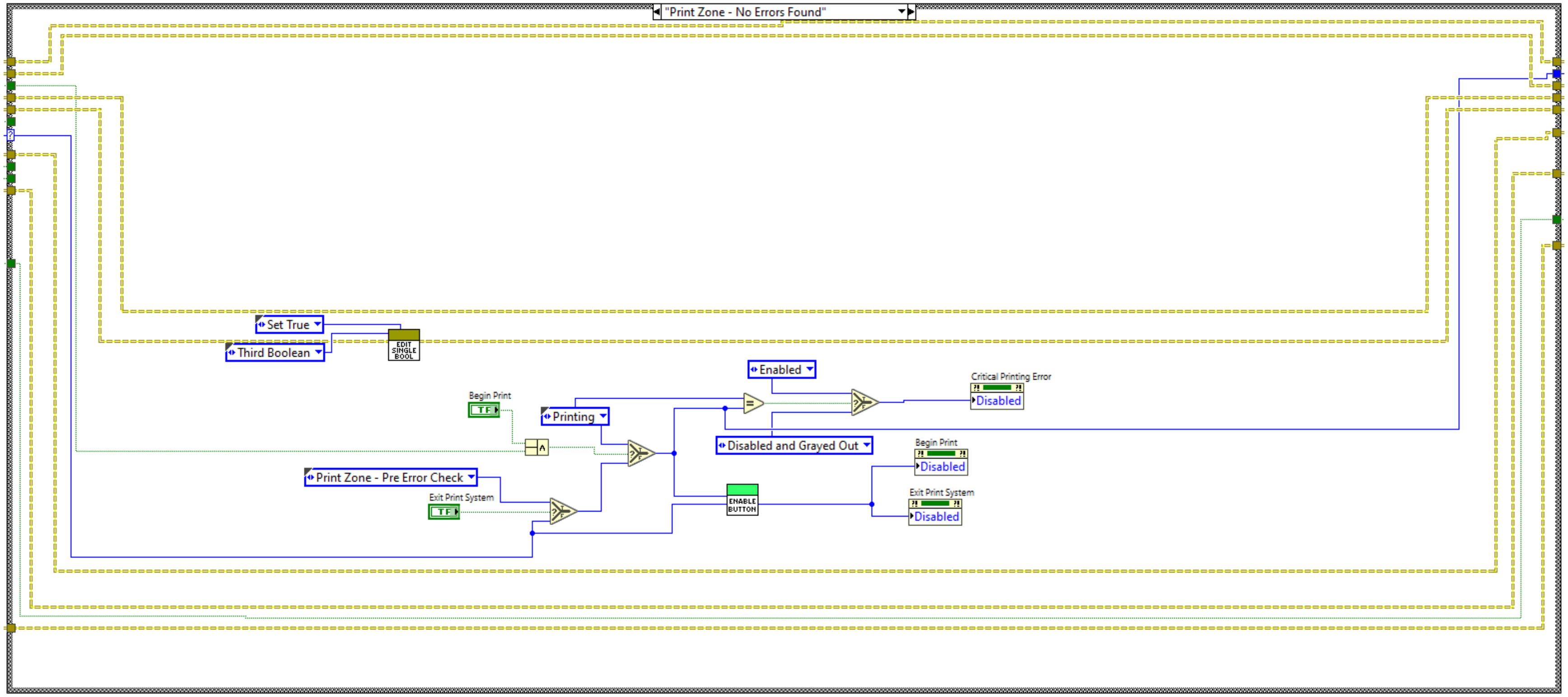


Figure 21: LabVIEW block diagram printing zone no errors present

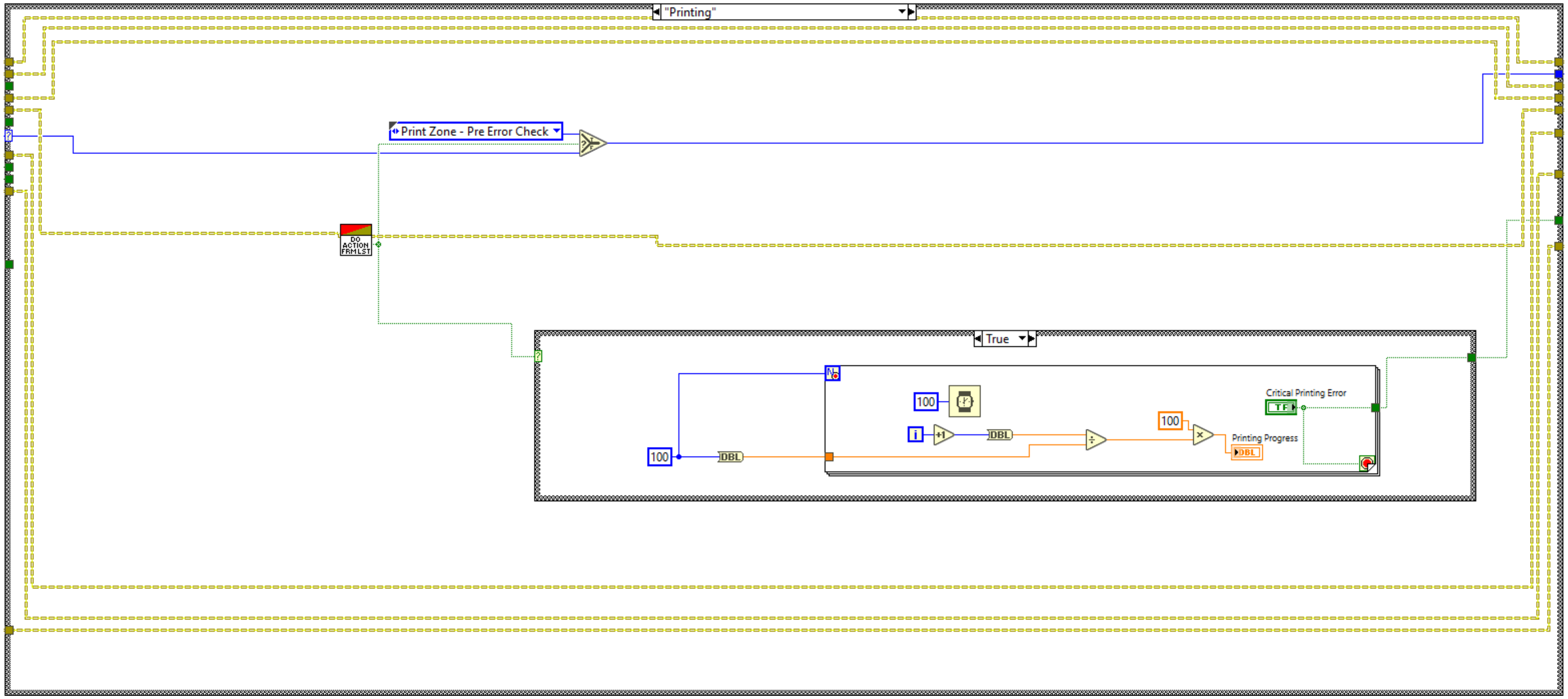


Figure 22: LabVIEW block diagram printing operation

B LabVIEW Object Virtual Instruments

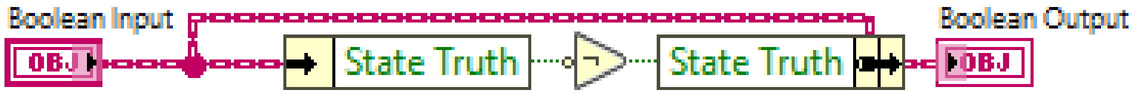


Figure 23: LabVIEW boolean toggler

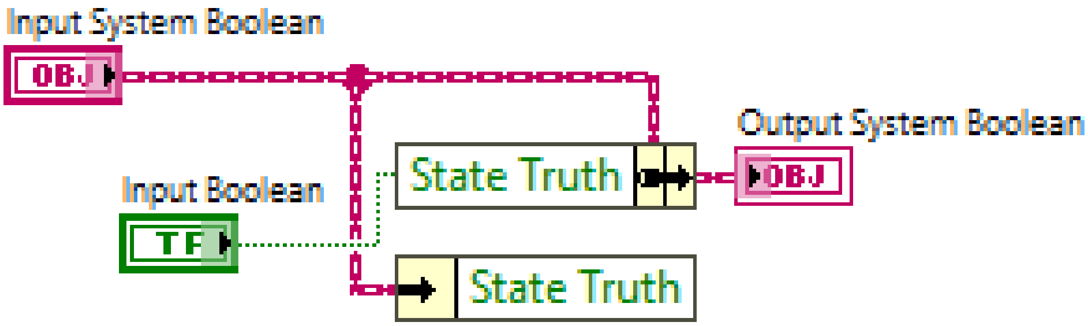


Figure 24: LabVIEW boolean setter

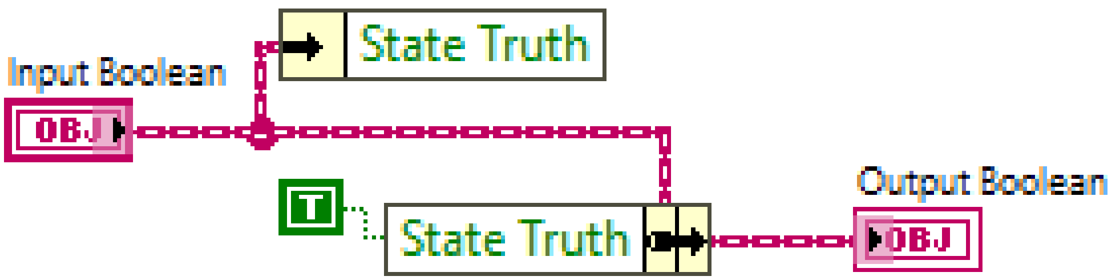


Figure 25: LabVIEW boolean set true

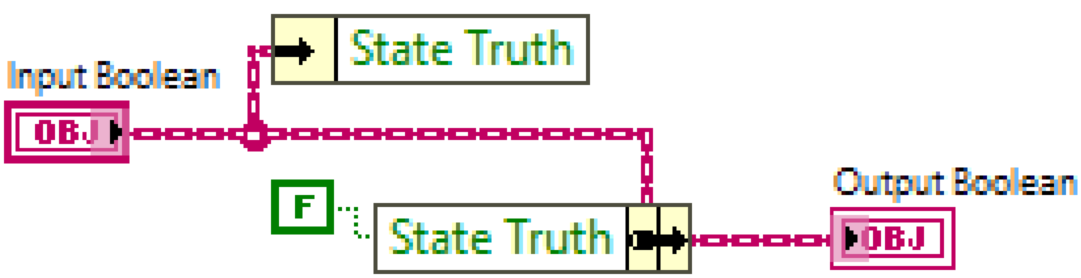


Figure 26: LabVIEW boolean set false

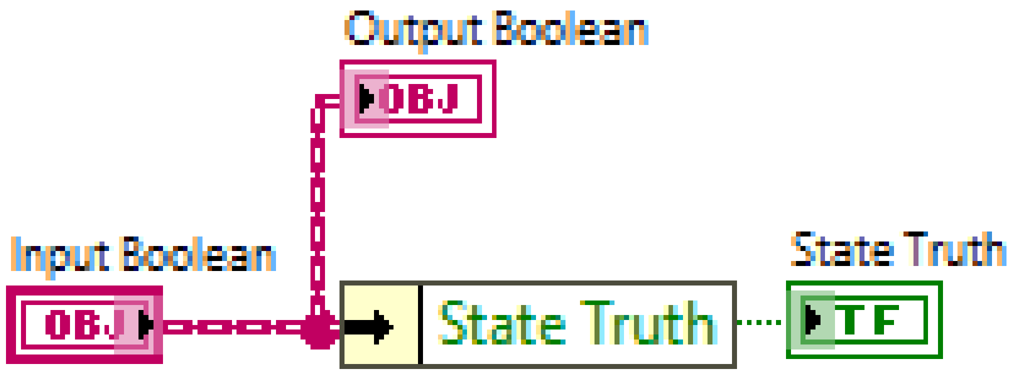


Figure 27: LabVIEW boolean accessor

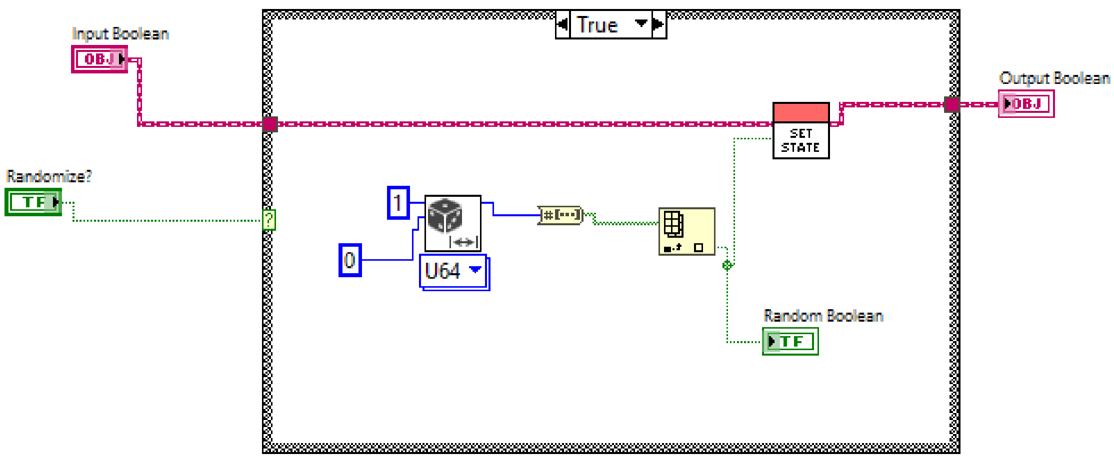


Figure 28: LabVIEW boolean randomizer

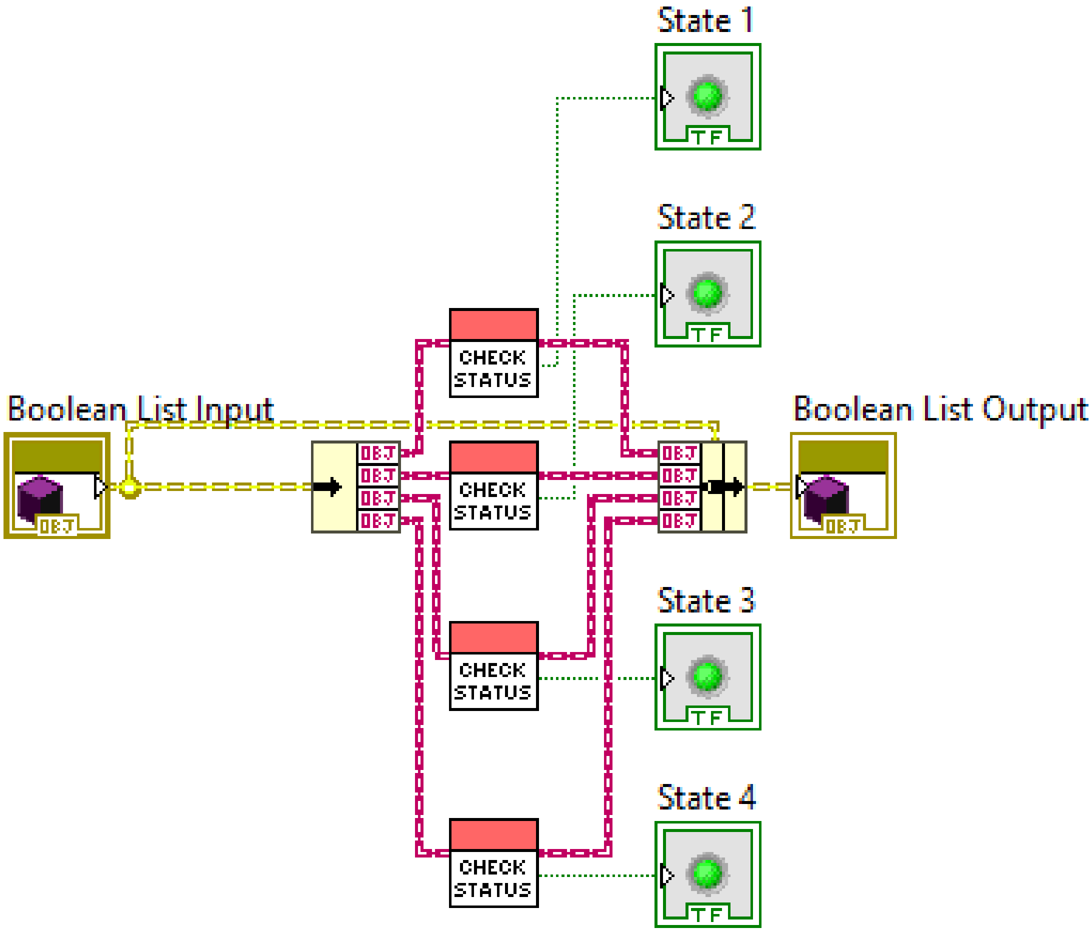


Figure 29: LabVIEW boolean list checker

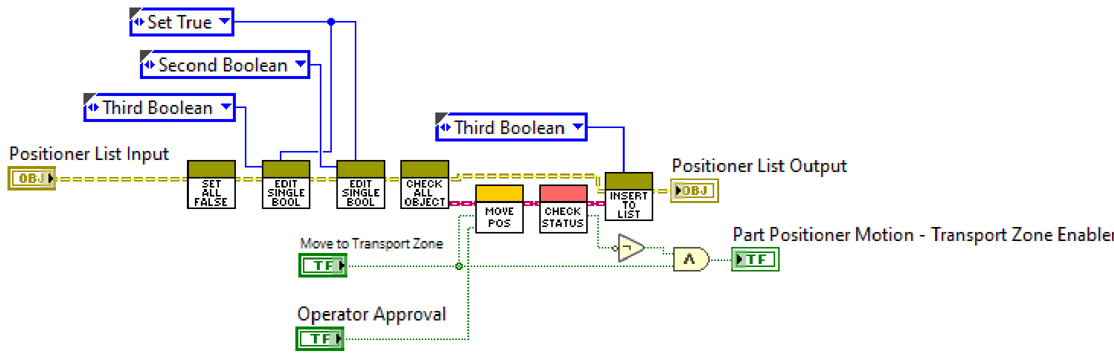


Figure 30: LabVIEW part positioner machining zone

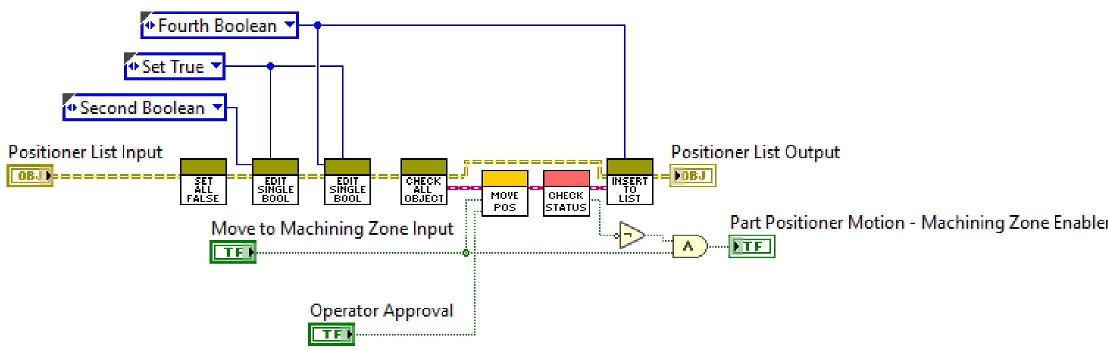


Figure 31: LabVIEW part positioner transport zone

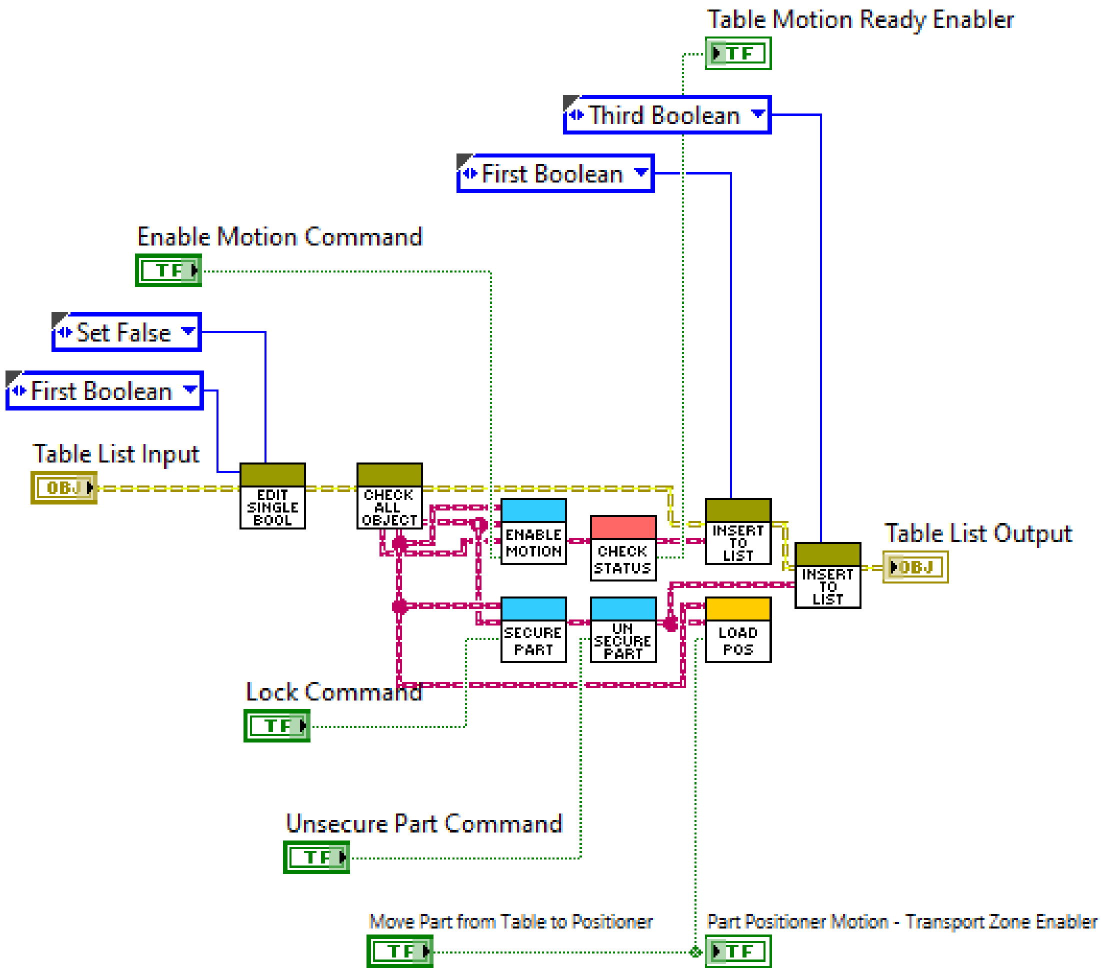


Figure 32: LabVIEW part positioner loading from table

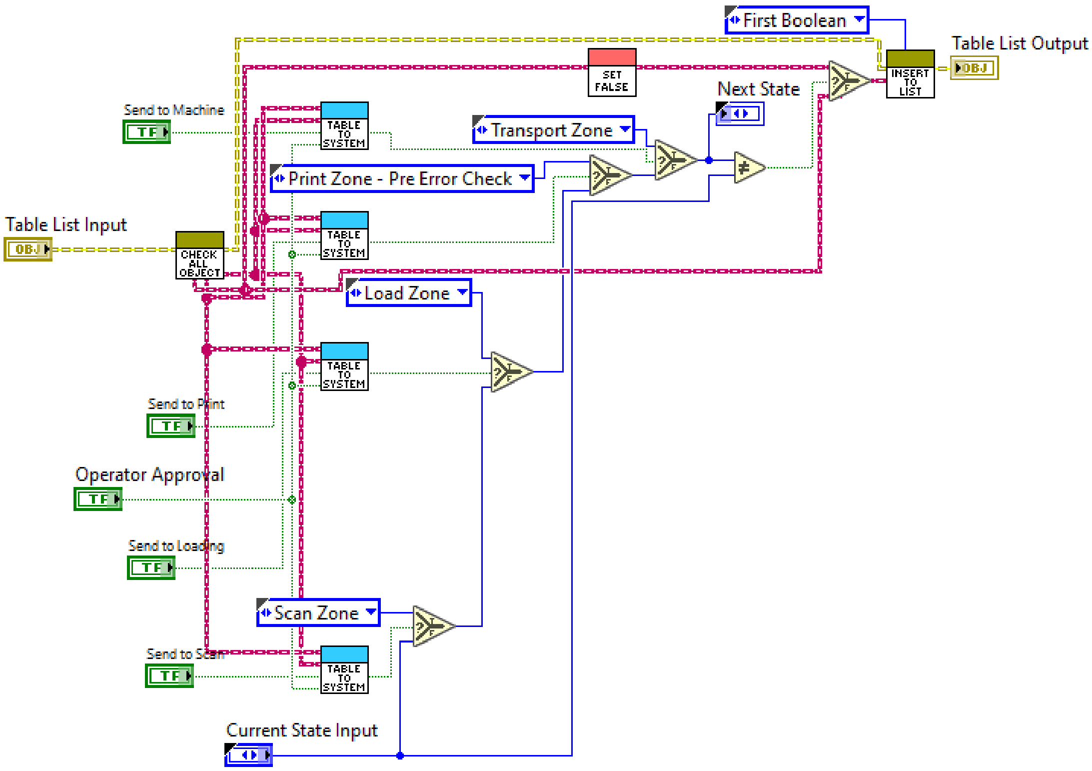


Figure 33: LabVIEW send table to location

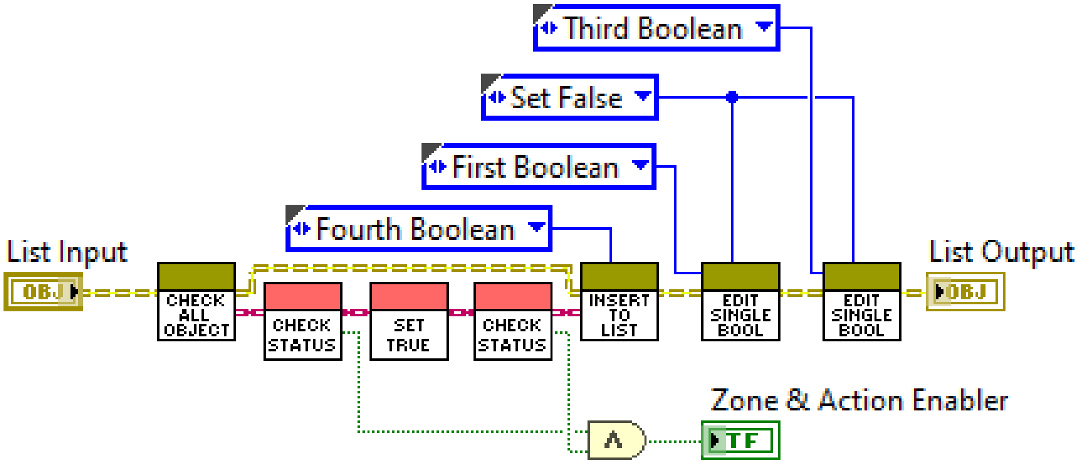


Figure 34: LabVIEW do action from list

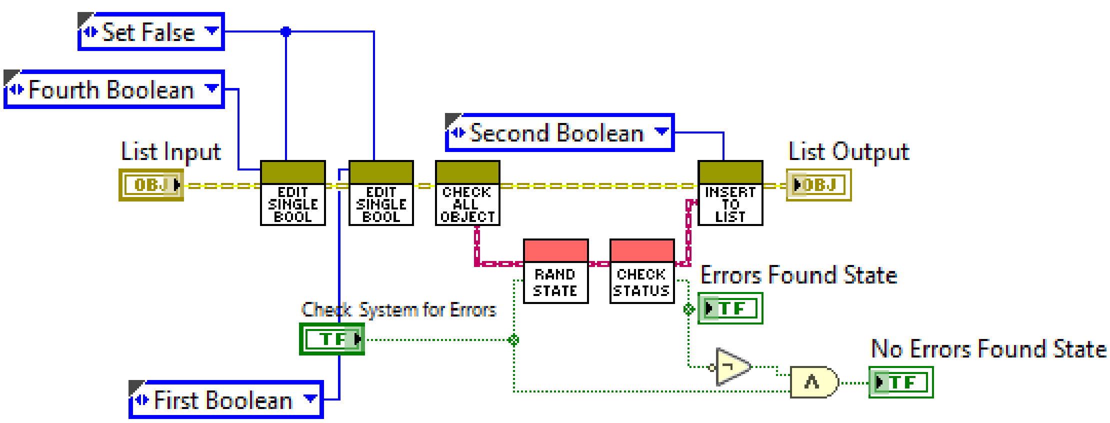


Figure 35: LabVIEW check system for errors

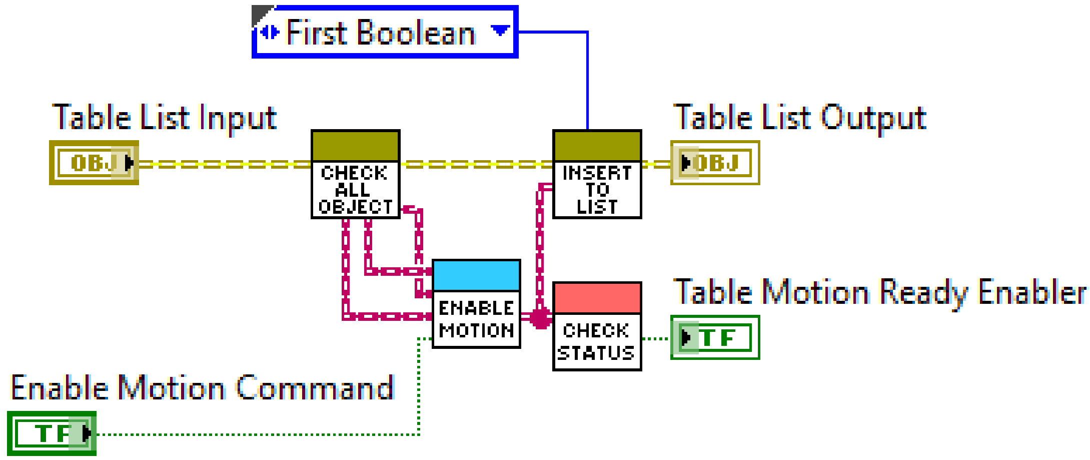


Figure 36: LabVIEW enable table motion

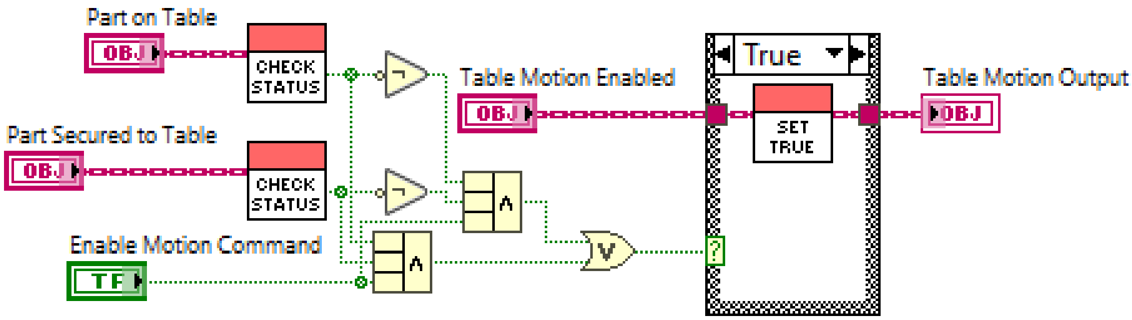


Figure 37: LabVIEW enable motion

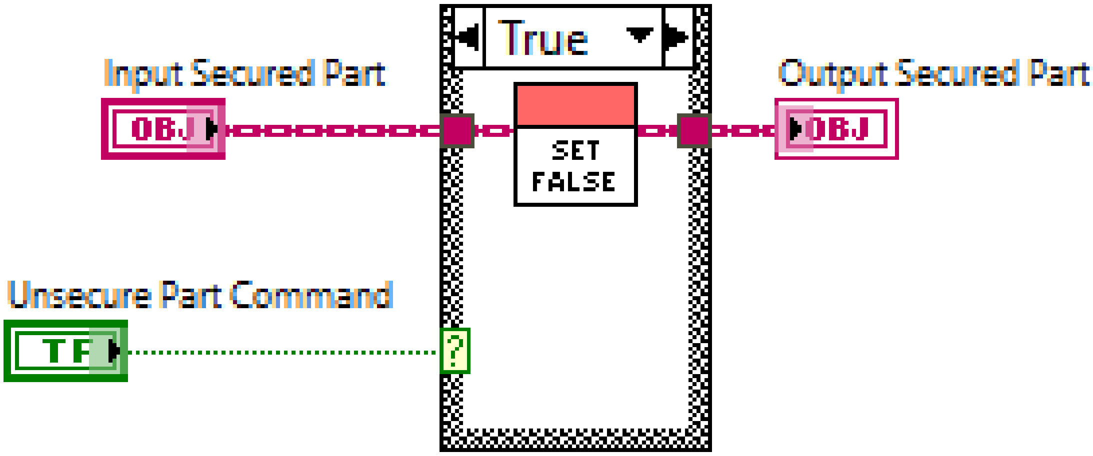


Figure 38: LabVIEW unsecure part

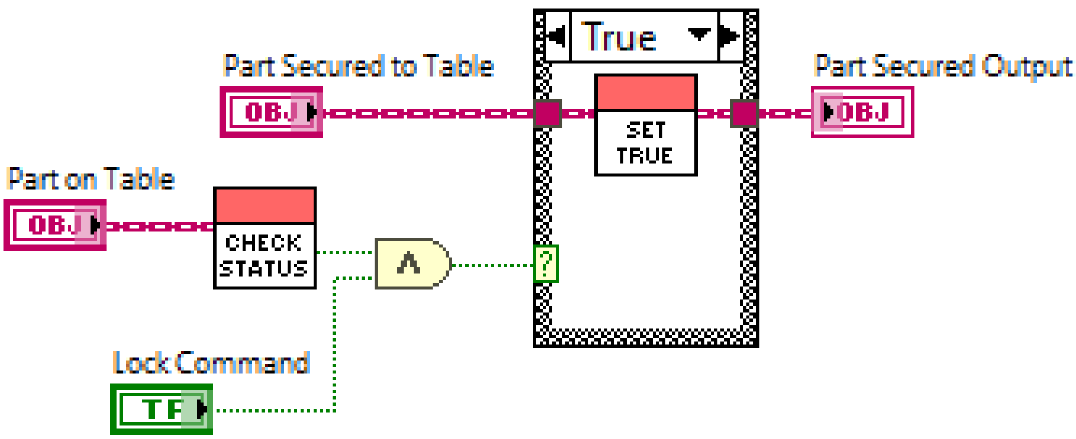


Figure 39: LabVIEW secure part

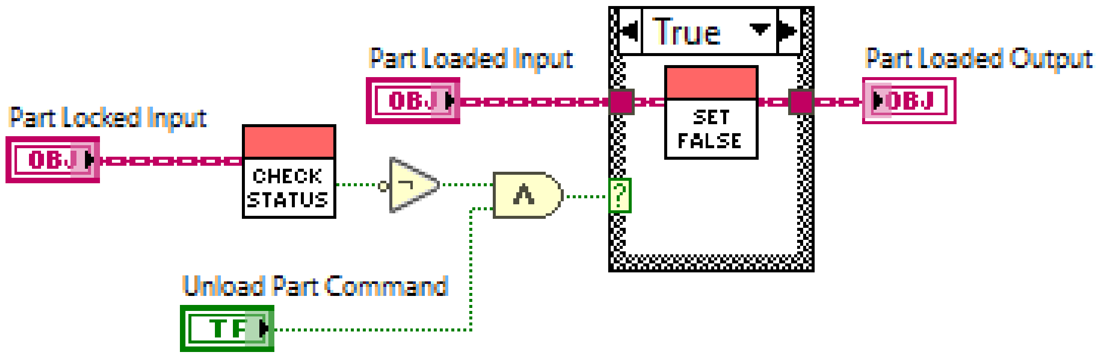


Figure 40: LabVIEW unload part

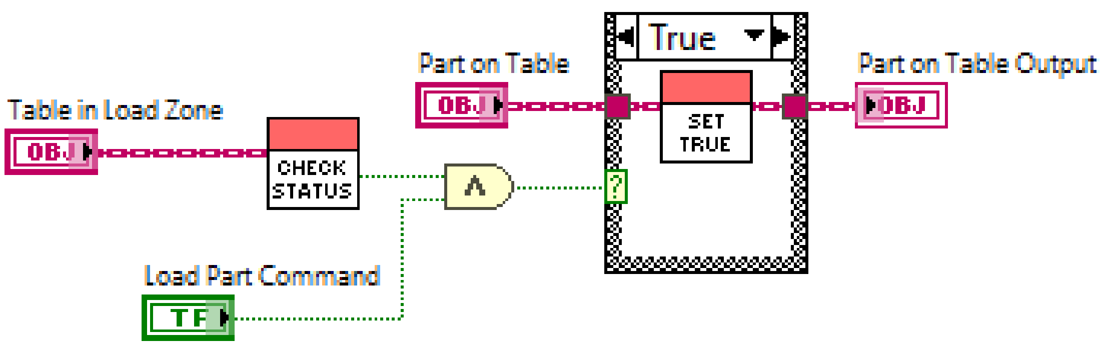


Figure 41: LabVIEW load part

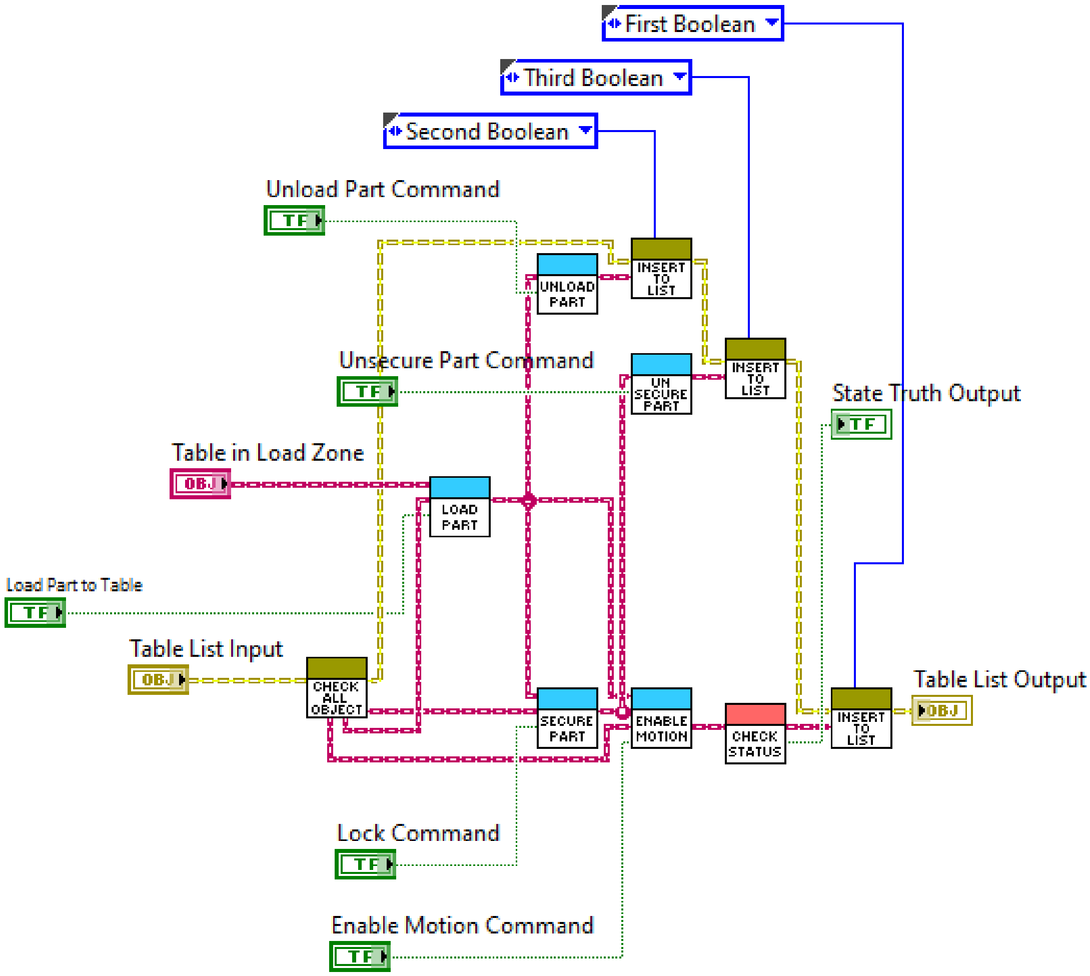


Figure 42: LabVIEW loading and unloading

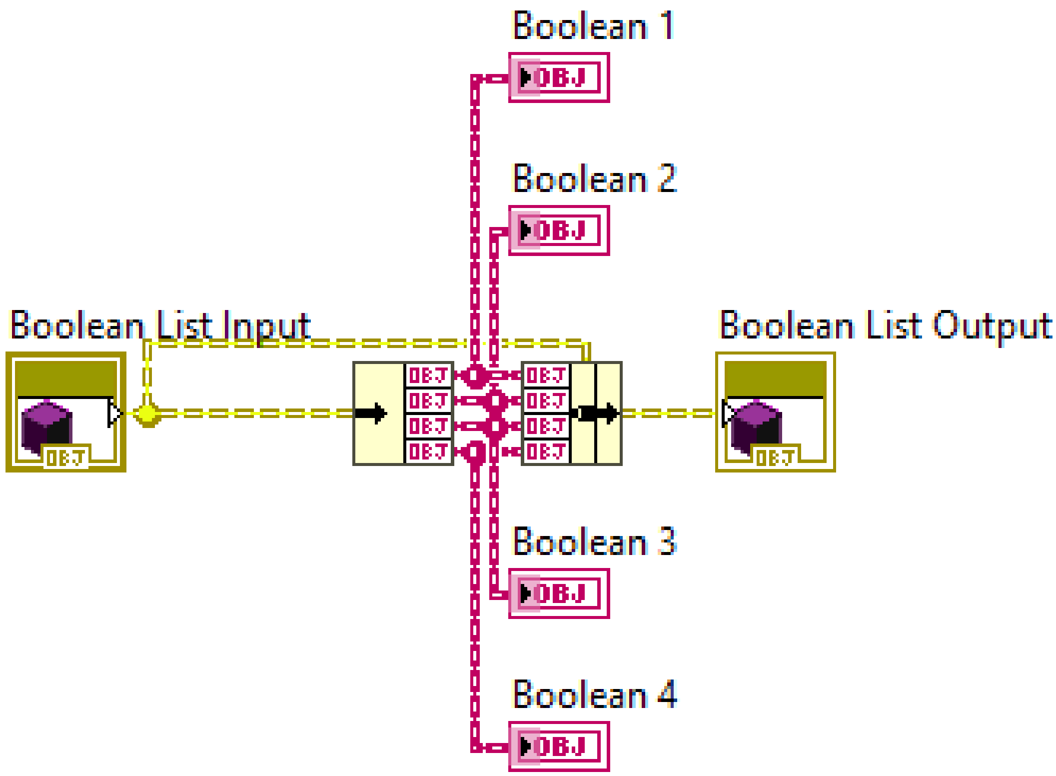


Figure 43: LabVIEW check all objects

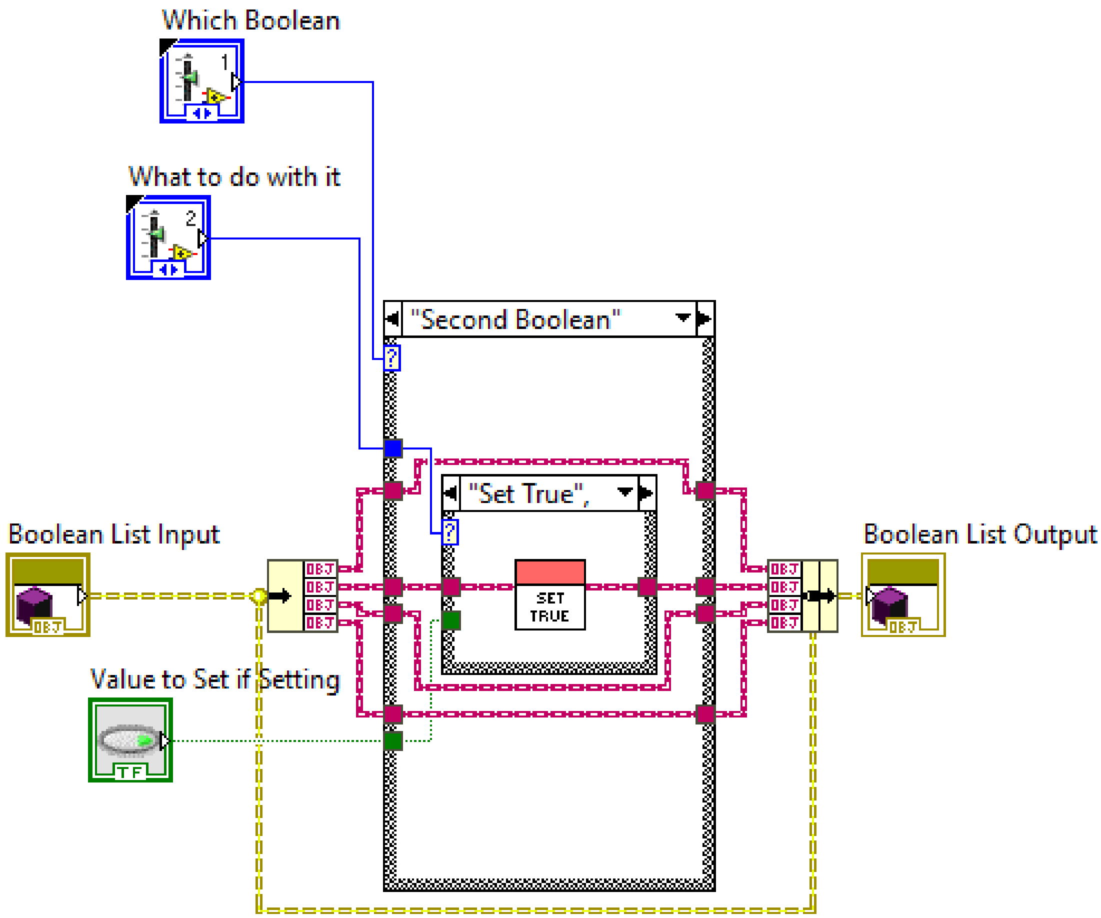


Figure 44: LabVIEW index action

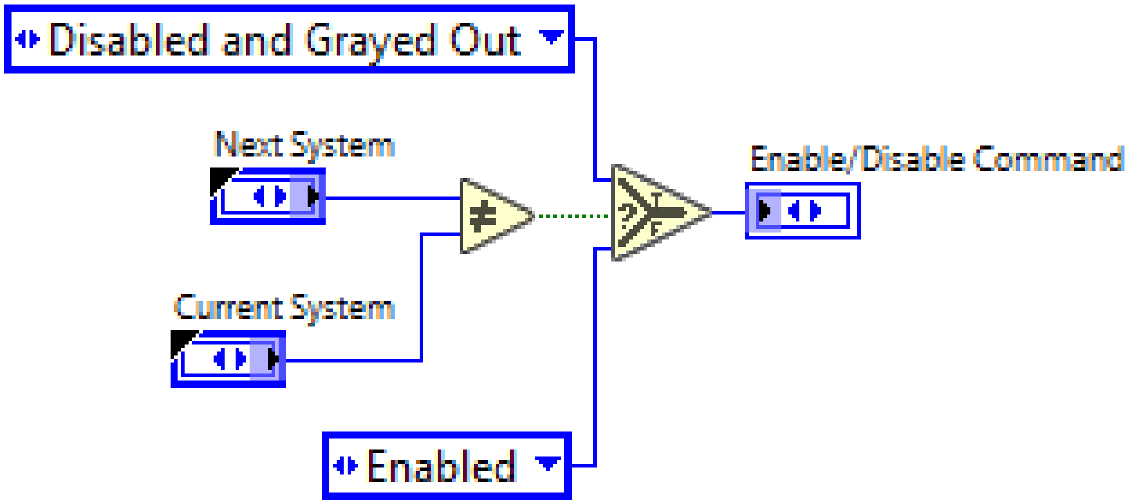


Figure 45: LabVIEW enable button

C Physical Component Implementation

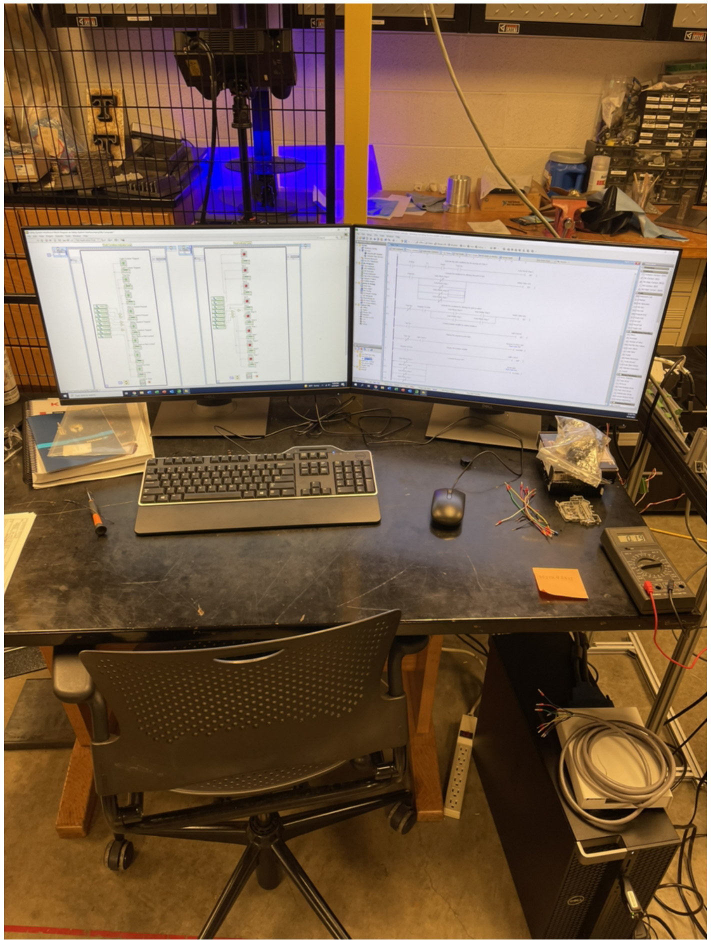


Figure 46: The supervisory control PC that is the master of this cell

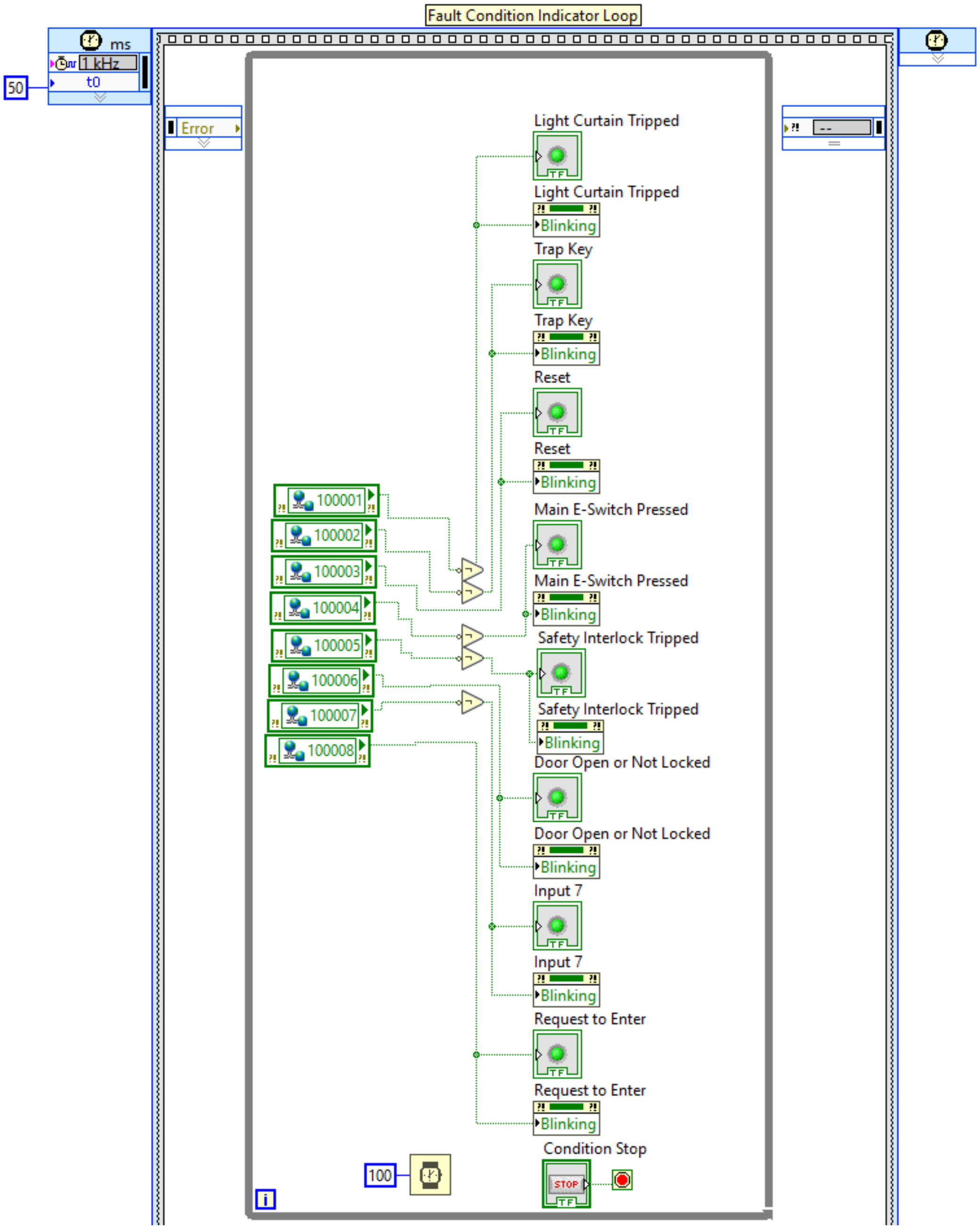


Figure 47: The fault condition indicator component of the safety interlock LabVIEW block diagram

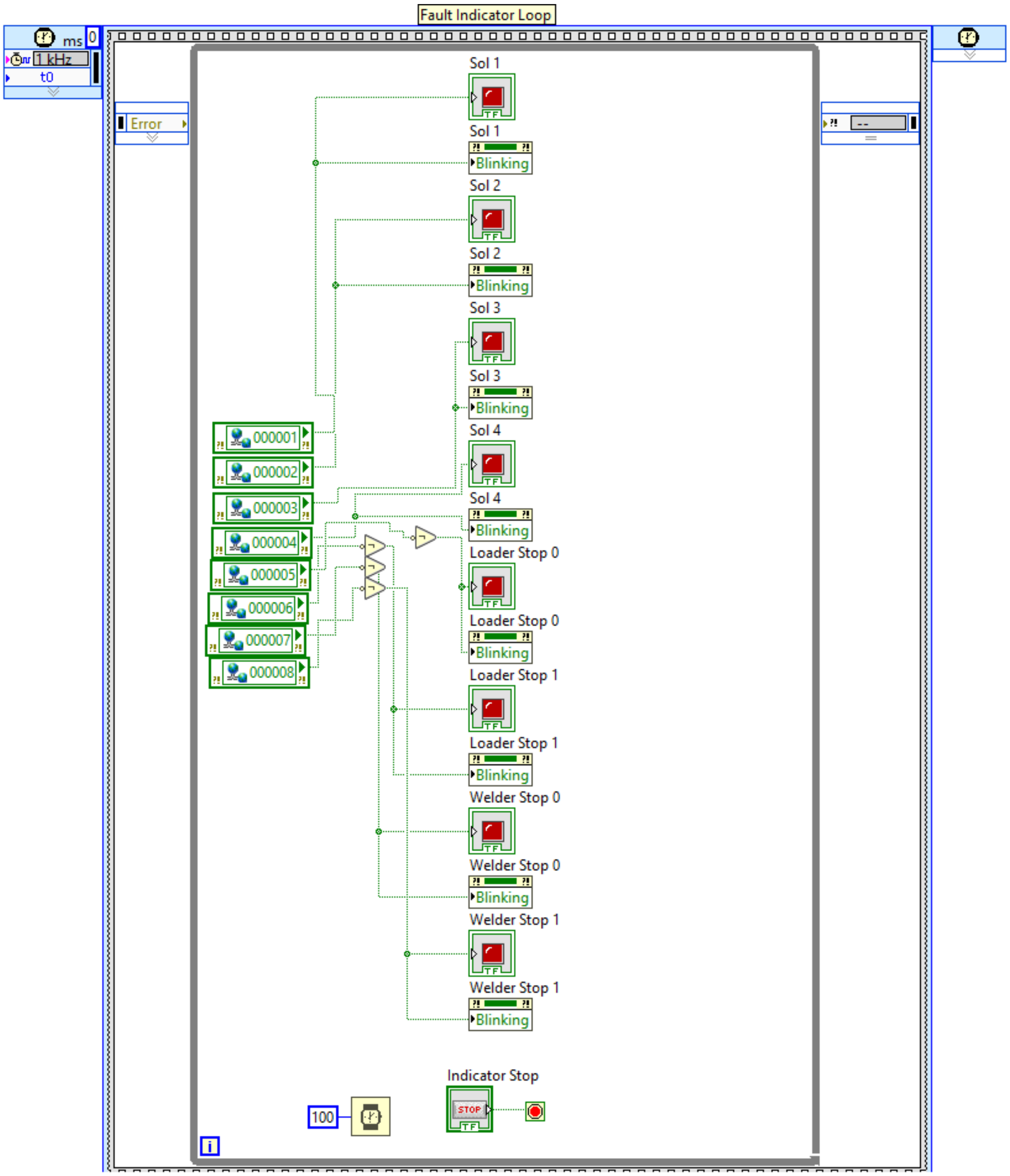


Figure 48: The fault indicator component of the safety interlock LabVIEW block diagram

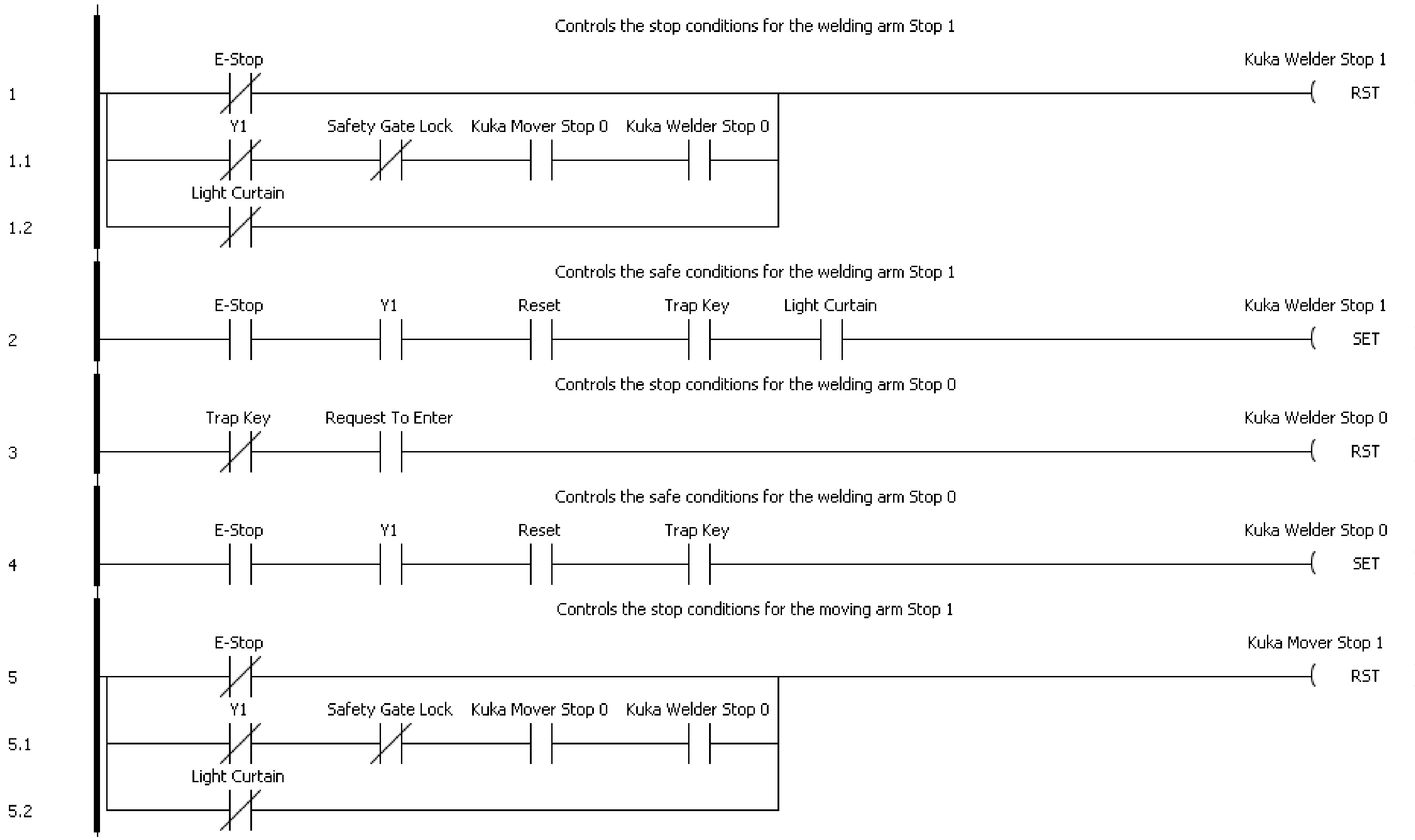


Figure 49: The first section of the ladder logic code for the safety interlocks

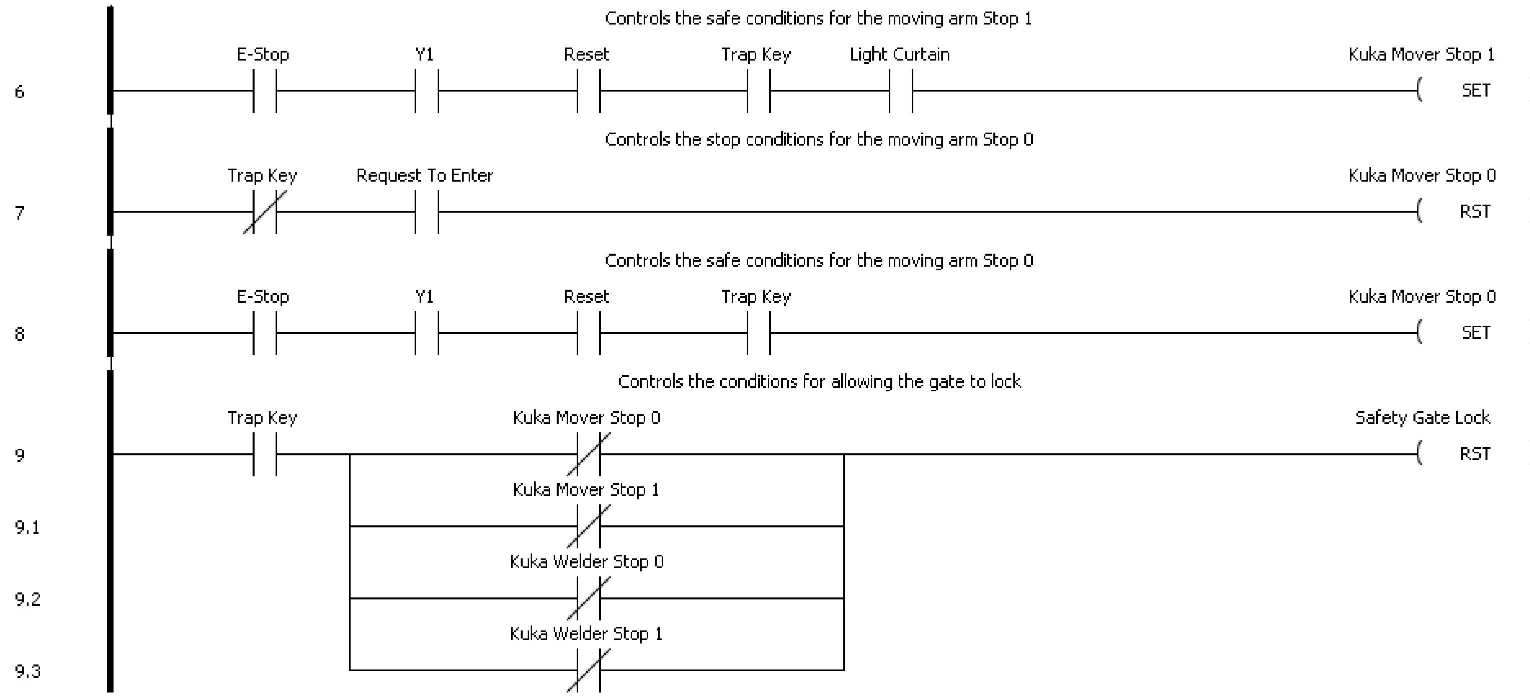


Figure 50: The second section of the ladder logic code for the safety interlocks

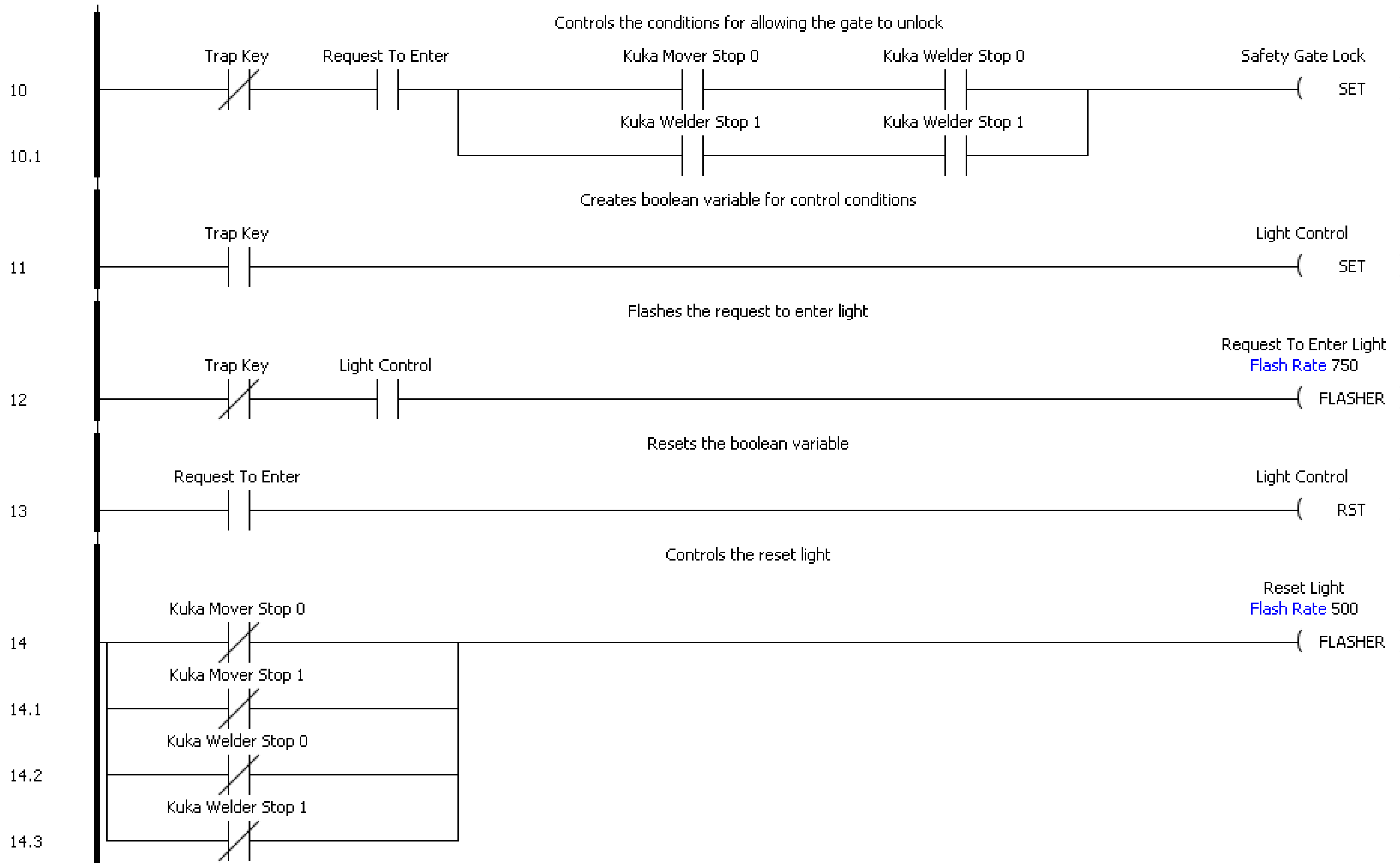


Figure 51: The third section of the ladder logic code for the safety interlocks

Vita

Michael John Buckley was born and raised in Brentwood, Tennessee by his parents Dan and Leslie along with his older brother Sean. Exposed and interested in engineering from a young age, he went to the University of Tennessee Knoxville in pursuit of a mechanical engineering undergraduate degree. He had always wanted to find a way to mesh mechanical engineering with computer programming, either through a minor or a job, but it never quite worked out. Beginning in his junior year, he began to work as an undergraduate research assistant to Dr. Caleb Rucker in the field of soft-robotics, which he continued to do until he completed his undergraduate degree.

Michael had not originally planned to attend graduate school until learned he could finish a master's program in one year at the University of Tennessee. He chose to stay in the field of mechanical engineering for his master's degree, and chose a focal area of robotics and controls. In the pursuit of his master's degree, he was taken on as a graduate research assistant under Dr. Bill Hamel in a lab space focused on industrial robotics and additive manufacturing through arc welding. This research and work creating a supervisory control system spurred an interest in the field of robotic controls, where he plans to work.