EET Senior Design Project

IMS Dynamic Display

Submitted to

Professor Robert Weissbach, PhD
Professor William Lin, PhD
Electrical Engineering Technology Program
Engineering & Technology Department

by

John Tyler Boggess
Michael Lee Elkins

Apr. 29, 2019

IMS Dynamic Display

Abstract

The Indianapolis Motor Speedway Museum has a display transmission that is still actively used in IndyCar today. The museum wants to incorporate this transmission into an interactive display, so guests of all ages can see internal gears spin as well as see the transmission shift between its gears. This project includes mechanical and electrical engineering technology students working together to mount wire motors and sensors. The transmission will have a user-friendly interface allowing the guests to change gears and to turn the transmission on and off. The transmission will be driven by a 24V DC motor and uses a 24V DC linear actuator to rotate a barrel cam to change the position of the forks, allowing the gears to get shifted up and down. A metal enclosure houses the electrical components that provide power and control to the system. The outcome of this project is a failsafe and robust system that will operate within the IMS Museum while being continually updated.

Table of Contents

Revision History

Version	Date	Contributors	Description
1.0	23, Oct. 2018	John Boggess Michael Elkins	Initial Version
2.0	25, Oct. 2018	Michael Elkins	Student Peer Review changes added
3.0	27, Nov. 2018	Michael Elkins	Reformatted to match recommended style
4.0	13, Dec. 2018	John Boggess Michael Elkins	Final Version – Senior Design Phase I
5.0	04, Mar. 2019	Michael Elkins John Boggess	Polished Draft – Senior Design Phase II
6.0	04, Apr. 2019	Michael Elkins John Boggess	Final Draft – Senior Design Phase II
7.0	30, Apr. 2019	John Boggess Michael Elkins	Final Paper Submitted to University Library

IMS Dynamic Display

Introduction

Xtrac, Inc., has supplied a cutaway gearbox to the Indianapolis Motor Speedway (IMS) Museum. The IMS wants to make the gearbox display interactive by having the gears shift up and down at the press of a button. We will create a panel to allow users to approach the gearbox display, turn the transmission mechanism, and shift gears. The mechanical and electronic components that operate the gearbox will be enclosed as much as possible to create an appealing museum display.

A transmission is an essential part to a complete vehicle, but how well can people comprehend how a transmission works if they don't see it in working condition? A transmission allows a vehicle to shift with gears as a motor spins faster or slower. Using an engine and transmission in a race setting requires higher torque to maintain higher speeds. A transmission by itself requires some form of electrically powered input to operate independently from a racing engine. The IMS has an IndyCar transmission that has been donated by a company named Xtrac, Inc. The IMS has contacted the engineering department at IUPUI to get students to make their transmission interactive.

The IMS wants an interactive display that shows how the transmission turns and shifts through the gears and allows people of all ages to learn. The project covers 2 semesters because the Electrical Engineering Technology (EET) curriculum has the design process in the fall semester and implementation in the spring semester. Since a transmission has mostly a mechanical aspect, a team of Mechanical Engineering Technology (MET) students are paired with a team of EET students. Any mounting and specification requirements are a part of the MET students' project requirements. Details regarding electrical power, circuit design, and electromechanical integration will be generated by the EET students.

Problem Statement

The IMS Dynamic Display will have a user-friendly interface allowing students, mentors, lecturers, and guests to control all actions of the transmission. The interior gears of the transmission are going to be rotated with a 24V DC permanent magnet motor. A 24V DC linear actuator will be used to rotate a barrel cam that shifts forks back and forth to shift throughout the gears. The user interface (UI) will be robust allowing for a large amount of wear and tear to the electrical buttons as well as the mechanical parts.

System Overview

Having a display transmission is useful but is not at its full potential because no gears are turning around, and nothing can be viewed in a real-time setting. Having a transmission that will turn the interior gears and shift through the positions will be educational and useful for audiences of all ages. Allowing the audience to see the difference between first neutral and 6th gear can be mesmerizing at a young age.

Referenced Documents

Table 1: Referenced Documents

Title	Document Reference Number	Comment
Project Proposal and Specifications	PPS001 rev. 1	Submitted 10/05/18
User Interface Design	UID001 rev. 1	Submitted 12/05/18
Control Box Schematic	CBS001 rev. 2	Submitted 03/28/18
Final Project Proposal	FPP001 rev. 1	Submitted 01/31/19
CCW PLC Code	Ladder Logic	Phase I Complete

Project Design Decisions

The purpose of this project is getting the transmission gears to move and to increment and decrement by electronic control. A 24V DC motor will be mounted to the input shaft of the transmission that allows the inner gears to spin. A 24V DC linear actuator will increment or decrement the gear position from neutral through 6th gear and back down again. The permanent magnet DC motor in the actuator will be controlled using PLC ladder logic to allow the cylinder to extend/retract from a specific point. A Selector Switch provides power, Start and Stop pushbuttons control the gear motor, and 2 yellow pushbuttons control shifting.

Inputs/Outputs

The primary means of control for this project will be PLC ladder logic. Our design includes 5 inputs in its current state. The user will have access to 4 of these inputs: Start, Stop, Shift Up, and Shift Down (Figure. 2). The other input will be an analog signal that will communicate automatically to the PLC to determine actuator position. A Selector Switch will be placed between the incoming AC voltage and the control box components (Figure. 2). Five outputs are currently in the design (Figure. 3). The gear motor is connected via a relay to the PLC, the linear actuator is connected through 4 relays to the PLC (Figure. 4), and 2 lights are connected to indicate system status (On/Off).

Behavior

The Selector Switch will provide power to the control box. The Start pushbutton will turn the gear motor, while the Stop pushbutton will be used to halt any movement on the display (Figure 5 – Rung 1). Shift Up and Shift Down both communicate through the PLC to the linear actuator to turn the permanent magnet DC motor clockwise (CW) or counterclockwise (CCW) (Figure 8 – Rungs 23-24). Two yellow pushbuttons will contain the shifting inputs. The analog signal from a potentiometer will provide positional feedback of the linear actuator's shaft to the PLC. The internal ladder logic will move the actuator automatically based on the shaft position (Figure 6 – Rungs 9, 13; Figure 7 – Rung 19). The ladder logic will also prevent actuator movement based on the gear position of the transmission. The system will not be able to attempt shifting beyond 6th gear or below neutral gear (Figure 6 – Rung 10). The gear change operation requires 2 different outputs to control, so the ladder logic will include latched bits and timers to prevent a short.

User Interface (UI)

The UI for this display will be a neatly laid out panel. The Start, Stop, and shifting pushbuttons will be placed close together, since they will be used most often. Indicator lights will be placed at the bottom of the UI to indicate the gear motor's state (On/Off). A proposed labeling for the UI can be seen in Figure 9.

Safety

Safety features will be included into the programming, such as: if the motor is turned on and left alone, a timer will end and automatically shut down the gear motor (Figure 5 – Rung 1). Another issue that could occur is motor(s) stalling during operation. This stalling could burn up either motor if not addressed. Our ladder logic includes latches and timers to avoid shorts and prevent an operation that could cause burn up (Figure. 6 – Rung 13; Figure. 7 – Rungs 14, 19-

Commented [PK1]: I won't mark this going forward. Basically, the capitalization is inconsistent. Two paragraphs above, they are "start," "stop," etc. Either is fine, but be consistent in how you capitalize…and be consistent Selector Switch, Stop, Start….selector switch, stop, start…. Also, buttons or pushbuttons?

Commented [PK2]: A couple of things…. Inconsistency in the use of two, five, 2, 5, etc. *However*, when starting a sentence with a number, always spell it out (or adjust the sentence so that it doesn't start with the number). You can be inconsistent in this case (Two outputs are fine for 5 seconds….)

Commented [PK3]: Where are the references to Figure 1-4?

20) A major safety feature in the system will be the placement of the control box, which will be underneath the display and out of reach to users. Fuses are in place throughout the control box circuitry to account for excessive current. Special time-delay fuses will be used on the gear motor to account for inrush current at startup (Figure 4).

Construction

The control box will be an 18” x 18” x 6” metal enclosure. Two DIN rails will hold all mounted components placed within the control box enclosure. Wireway will be used to conceal internal wiring (Figure. 1).

Hardware

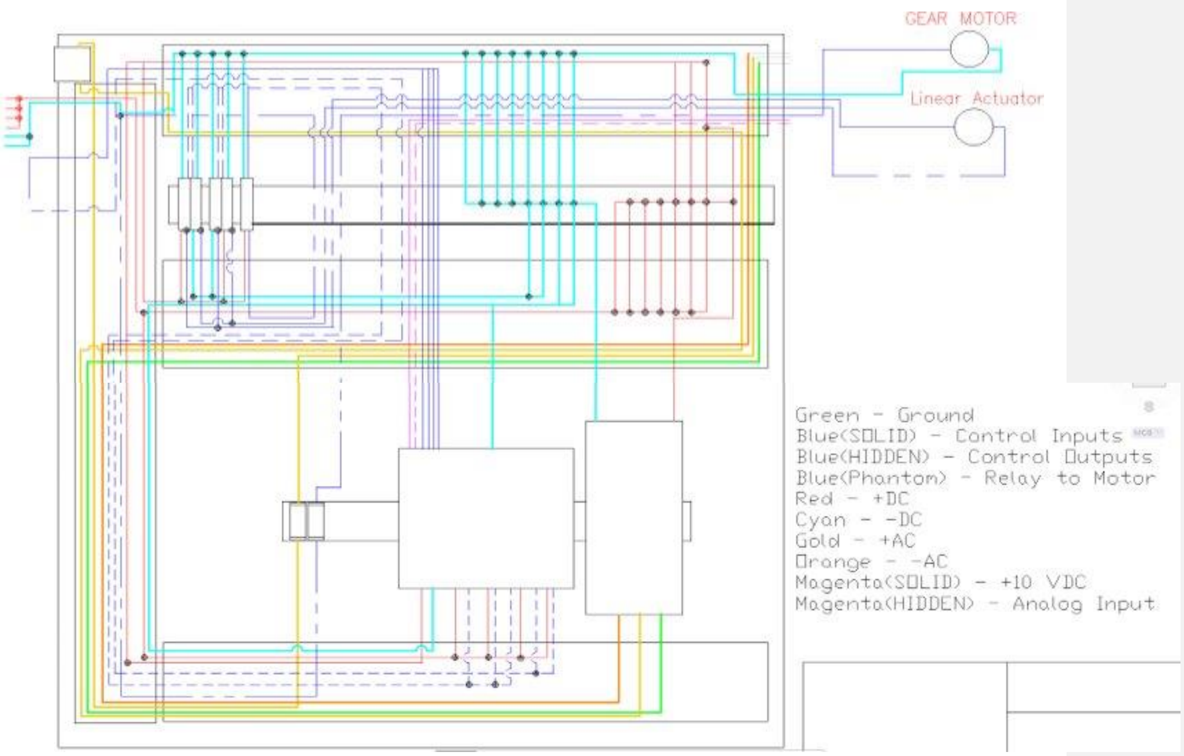


Figure 1: Control Box Wiring Layout

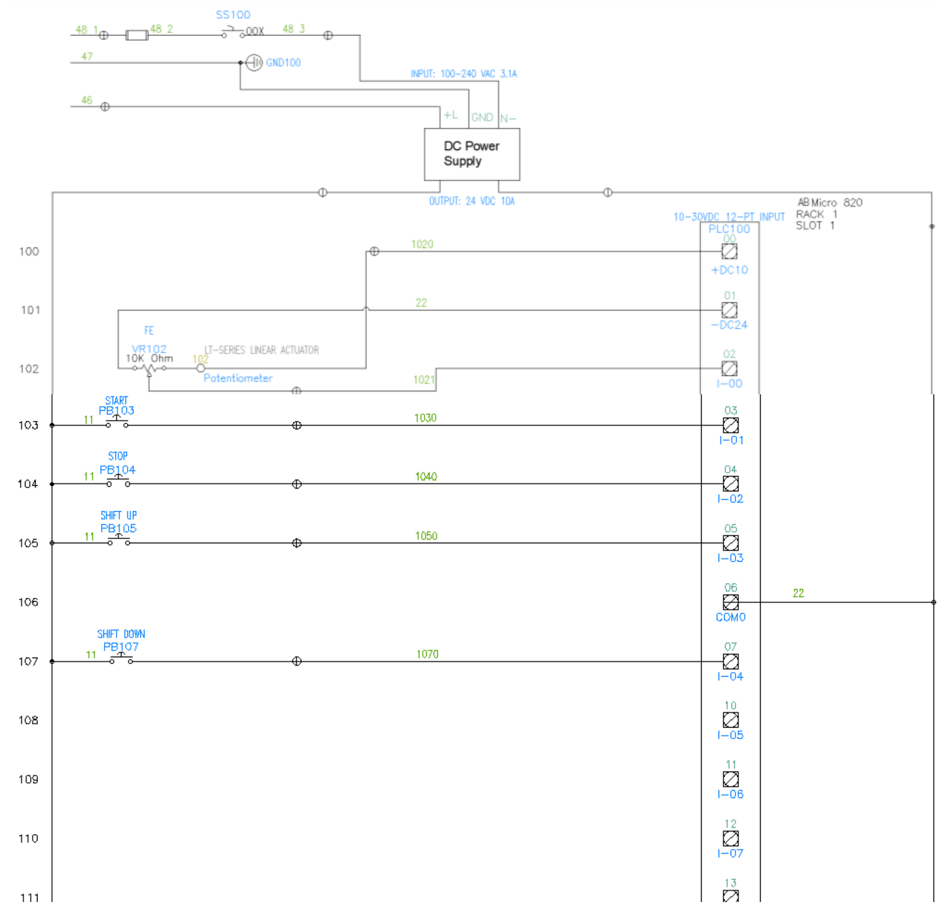


Figure 2: Control Box Schematic – Power and Inputs

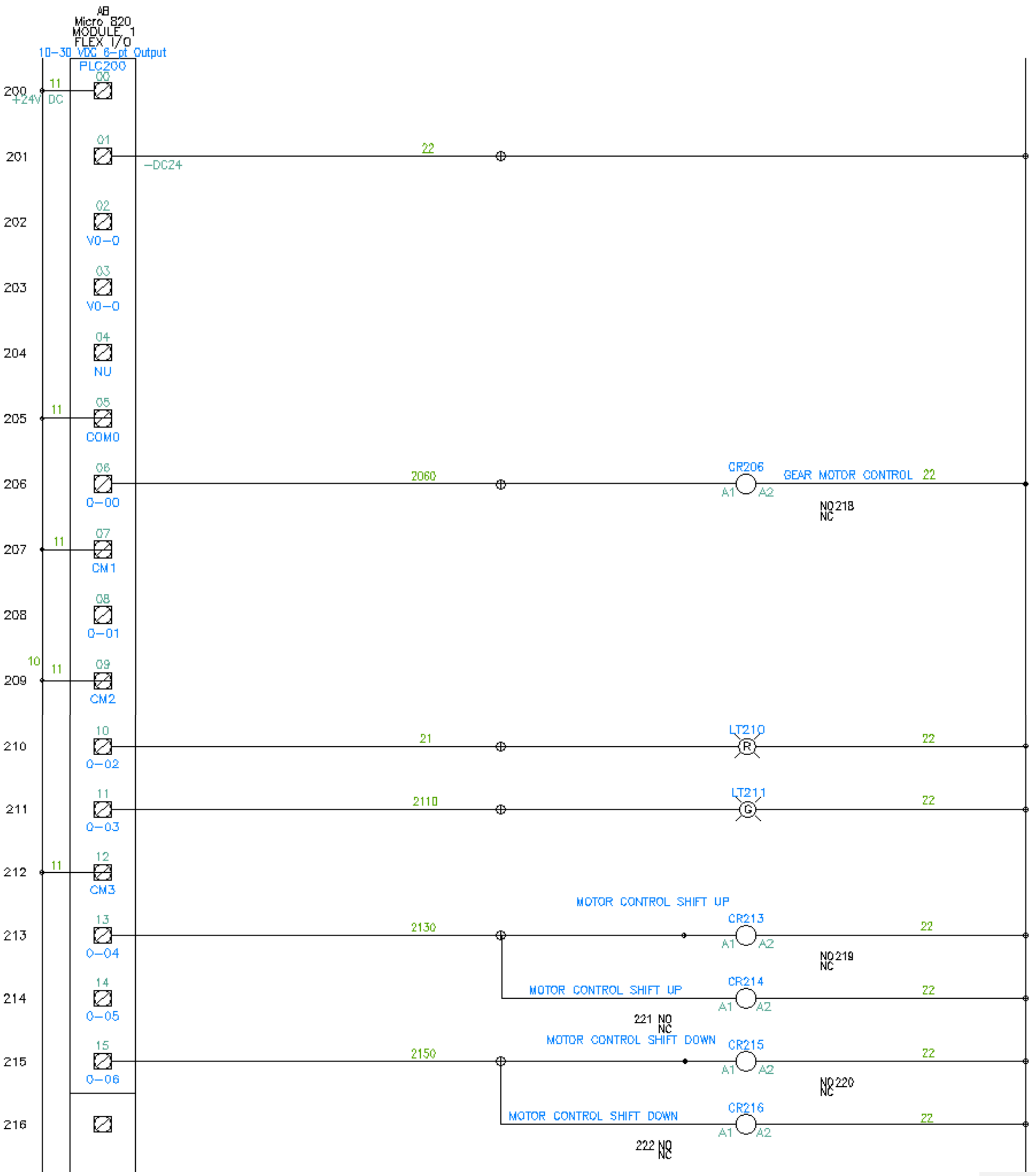


Figure 3: Control Box Schematic - Outputs

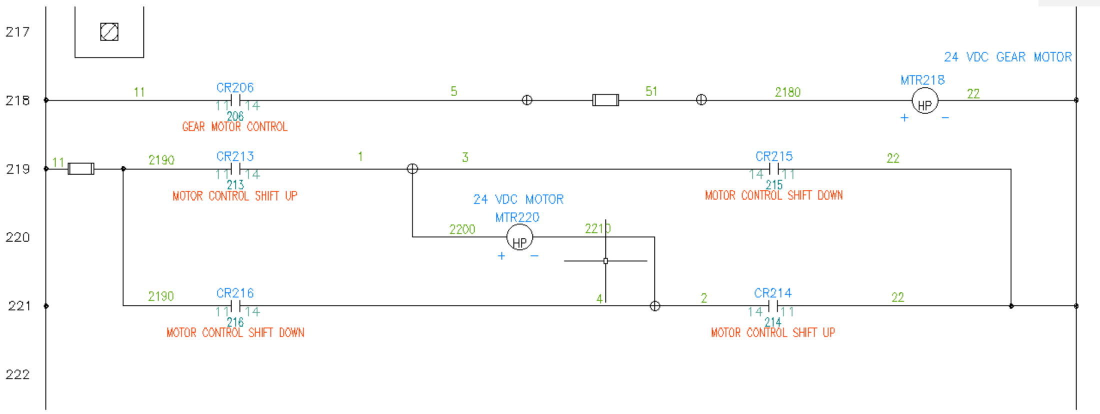


Figure 4: Control Box Schematic – Relay to Motor Connections

Software - Connected Components Workbench Software (CCW)

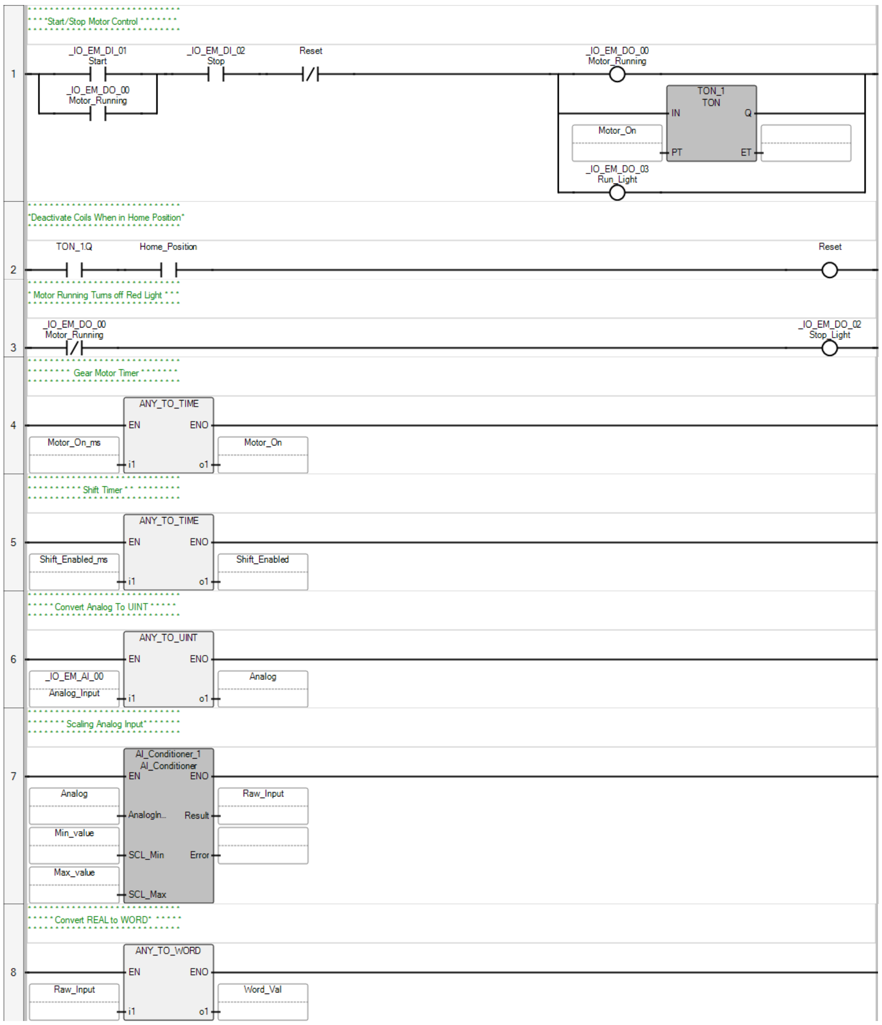


Figure 5: Rungs 1 - 8

IMS Dynamic Display

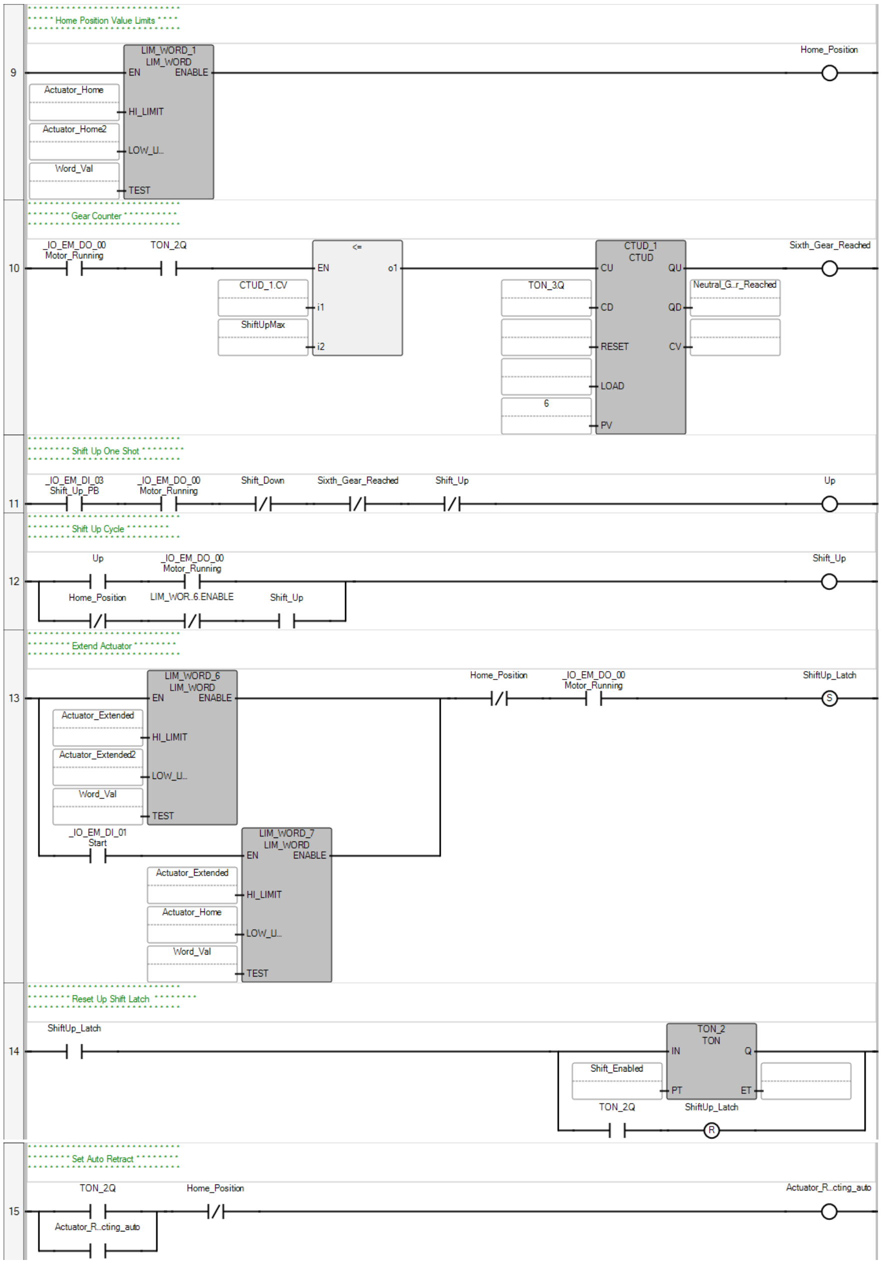


Figure 6: Rungs 9 - 15

IMS Dynamic Display

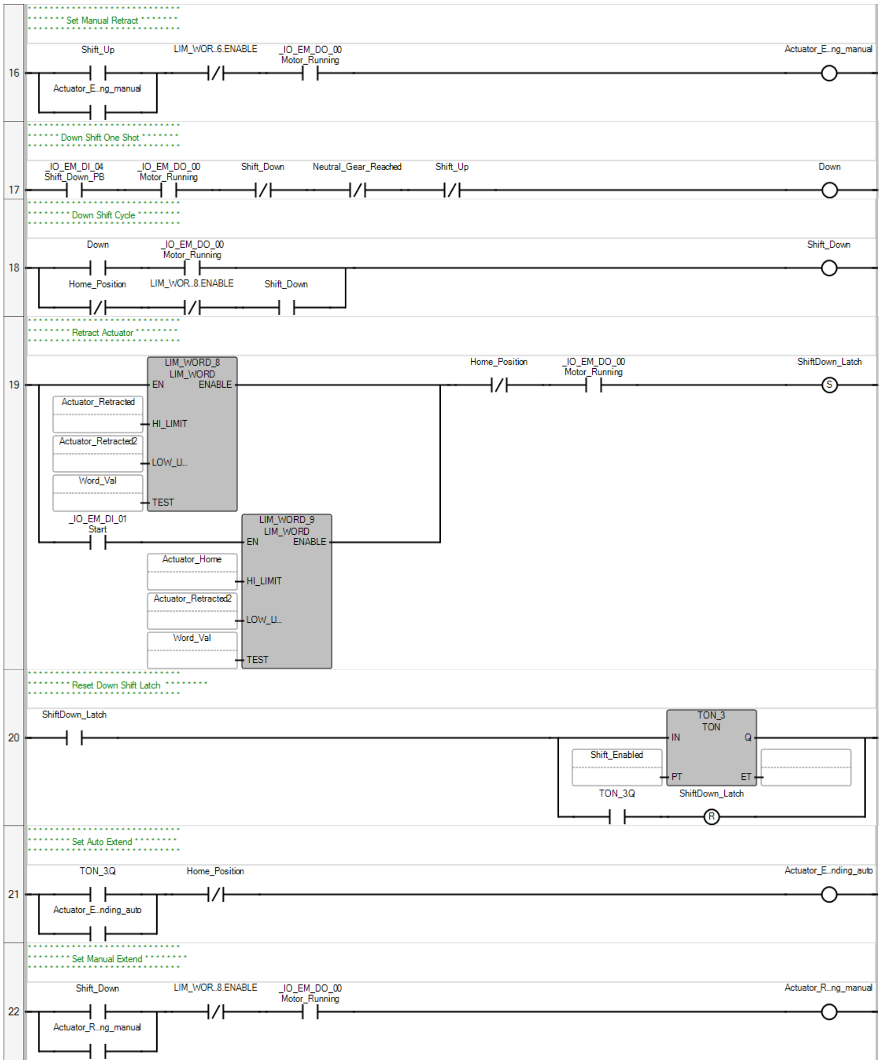


Figure 7: Rungs 16 - 22

Figure 8: Rungs 23 - 24

Interface

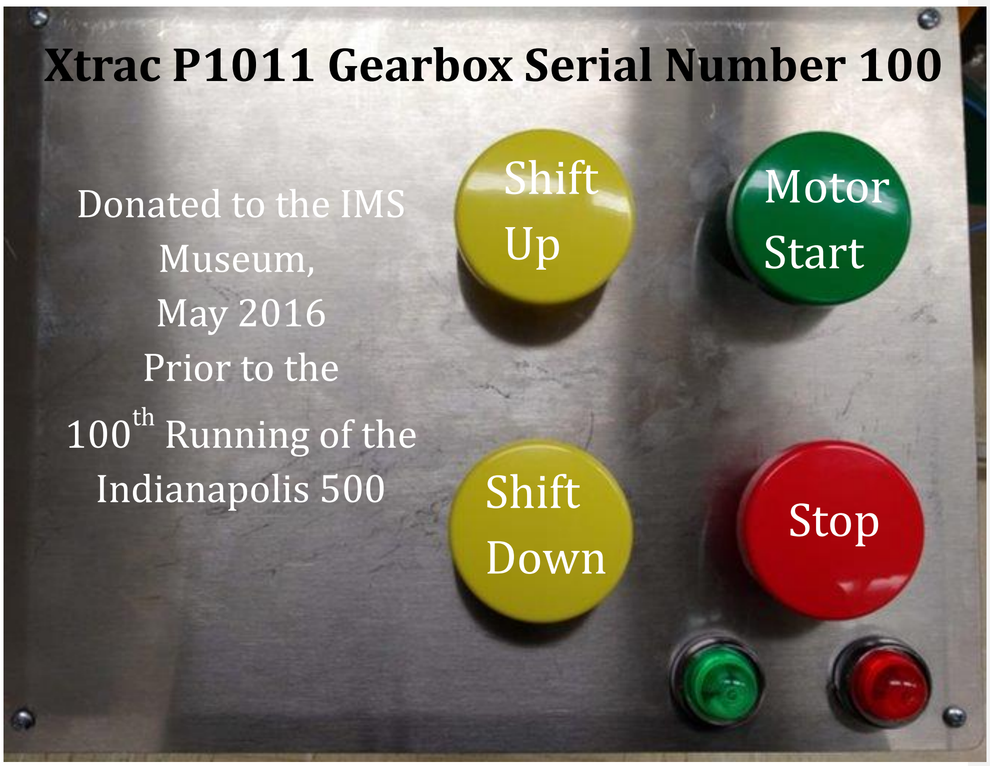


Figure 9: User Interface

Setup and Operation

The power cord will be plugged into the wall. The Selector Switch will be turned CW to provide power to the 24V DC power supply in the control box. After a few seconds, all systems should have power and be ready to be operated. The Start pushbutton should be pressed first, which will turn the gear motor. While the internal mechanism of the transmission is turning, the yellow pushbuttons can be pressed to shift up or down. The engaged gears will be seen through the display case, and the system should respond accurately based on the current gear position. After a set run time, the gear motor will turn off. The Stop pushbutton can be pressed at any time to halt the gear motor and reset the system timer. The display will be powered down by turning the Selector Switch CCW while the transmission's internal mechanism is stationary.

Testing

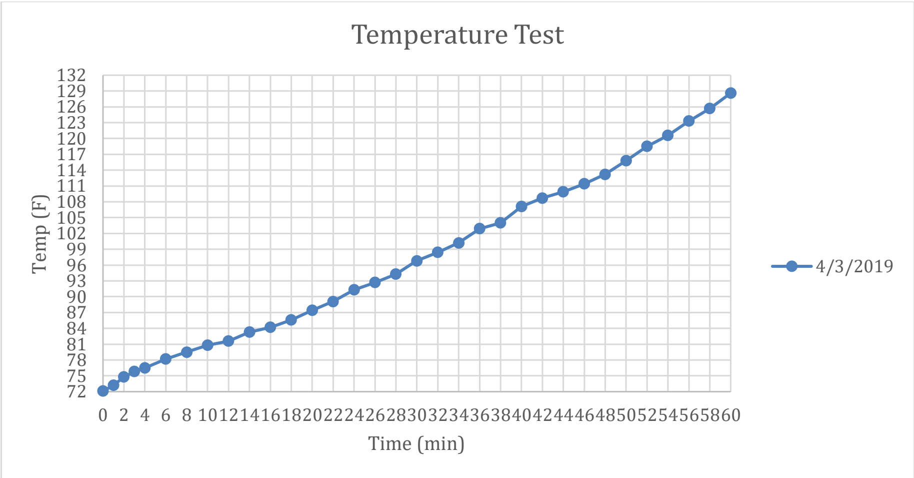


Figure 10: Gear Motor Surface Temperature

	Current from Power Supply (A)		
	Shift Down	Home	Shift Up
	2.22	2.60	3.12
	2.23	2.74	3.03
	2.25	2.71	3.12
avg	2.23	2.68	3.09

Figure 11: System Current Test

Gear	Motor RPM	Gear RPM	Gear Motor Current (A)		avg
nuetral	140	0	1.20	1.30	1.25
1st	130	11	1.62	1.60	1.61
2nd	126	15	1.91	1.90	1.91
3rd	122	19	2.17	2.20	2.19
4th	115	24	2.65	2.60	2.63
5th	115	24	2.67	2.70	2.69
6th	115	24	2.72	2.80	2.76

Figure 12: Gear Ratio/Gear Motor Current Test

Analog signal

Min Output Range	Max Output Range	Analog Input Value	Scaled Output Value
0	1000	3031	740
0	1000	3049	744
0	1000	3032	741
0	1000	3031	740
0	1000	2993	731
0	1000	3036	742
0	1000	3218	786
0	1000	3220	787
0	1000	3213	785
0	1000	3210	784
0	1000	3233	788
0	1000	3540	864
0	1000	3549	867
0	1000	3533	863
0	1000	3538	864
0	1000	3546	866

Figure 13: Analog Scaling Test

Conclusion

The IMS is involving the Engineering Technology Department at IUPUI in designing and implementing an interactive display for their transmission. The result should include a working transmission that shifts through all the gears using a lifelike racing simulated steering wheel. The transmission will be in a Lexan box with many safety features. The IMS is allowing us students the opportunity to get involved with the racing community as well as have a high-profile project to add to a resume. Allowing a neat but robust interface that will be used by guests is necessary so that electrical and mechanical parts do not fail inside a museum setting.

Recommendations

This project will require further work in coming semesters to bring it to its best possible conclusion. The design of the control box is such that modifications and improvements can be made very quickly. The UI and motor will need to be updated to better fulfill requirements given by Xtrac, Inc., and the IMS Museum. Below is a more detailed list of aspects of the project that:

- Will be changed by future project owners
- Should be updated based on tests and observations
- Can be modified if future project owners deem it useful and/or necessary

a) Necessary Updates
   a. The UI (Figure. 9) will be replaced by a Cosworth Steering Wheel and Paddle Shifters. An existing mounting system designed by a MET team will hold the steering wheel.
   b. Concurrent with the UI is the communication to the Micro 820 Controller. All UI controls will have to be transferred to the steering wheel and must be able to directly communicate with the current program downloaded to the controller.

b) Physical Updates
   a. Currently, the wiring throughout the project does not perfectly match with NEMA standards. As the project continues, wires internal and external to the control box should be swapped with the correct wire colors based on the type of connection (AC, DC, Ground, with load, etc.).
   b. The current 24V DC gear motor can turn the gearbox's internal mechanisms, but it does heat up steadily over time. When 4^{th} gear is reached, current and rpm level out. The greatest concern is that the motor is not strong enough to compensate for the higher torque in 4^{th} - 6^{th} gear. An incident occurred while the transmission was resting in 5^{th} gear which caused the motor to break. This issue has been partially resolved, but a larger motor would be necessary to ensure no breakage in the future.
   c. The current setup of the electromechanical relays connected to the linear actuator is enough for the project to run, but there is a possible concern that a short could occur. Software changes have been made to deal with the issue, but a physical interlock between the sets of relays would further prevent the chance of a short.

c) Optional Updates
   a. The CCW program for this project does not contain any subroutines. If desired, some of the functions of the program could be consolidated into subroutines to simplify the main program. This type of change would require knowledge of the main program structure and functions.
   b. There is a need for increased fault protection in the software. Some possible solutions can be seen in Appendix B.

Commented [PK4]: This just sort of appears…. Need transition to this.

Appendices

Appendix A: Datasheets/Product Documentation

https://literature.rockwellautomation.com/idc/groups/literature/documents/um/2080-um005_-en-e.pdf

http://www.xtrac.com/sectors/motorsport/single-seater/product/77

https://literature.rockwellautomation.com/idc/groups/literature/documents/pp/700-pp020_-en-p.pdf

LT SERIES

Features

- **Self locking Acme screw**
- **Internal, non-adjustable limit switches**
- **IP66 Sealing**
- **25% allowable duty cycle**
- **-25° to 65ºC (-13º to 150ºF) operating temperature**

Specifications

Model #	DC Voltage	Rated Load		Speed at Rated Load		Speed at No Load		Full Load Amps
		Newtons	lbs.	mm/sec	in/sec	mm/sec	in/sec	
Al'*"&"	&'	&'%	',	((	&#(	))	&#,	'#*
Al'*""""	')	&'%	',	('	&#(	))	&#,	&#*
Al*%"&"	&'	')%	*)	"#*	%#.	('	&#(	(#*
Al*%""""	')	')%	*)	'(#+	%#.	('	&#(	'#%
Al&%%"&"	&'	*%%	&&'	&'	%#*	&+	%#+	(#*
Al&%%""""	')	*%%	&&'	&(	%#*	&+	%#+	'#%
Al&*%"&"	&'	,*%	&+.	-	%#(	&%	%#)	(#*
Al&*%""""	')	,*%	&+.	.	%#)	&%	%#)	'#%
Al"*"&"	&'	&%%	"*	+	%#'	-	%#(	(#*
Al"*""""	')	&%%	"*	+#*	%#(	-	%#(	'#%

* = Stroke in millimeters

Installation & Operating Instructions

1. Application

The LT series actuators are intended for industrial use only and should not be used to lift, support, or otherwise transport people or loads over people. The actuator is designed for lifting or pushing loads no more than its load rating, on an intermittent basis, not for applications requiring continuous operation. The actuator is designed for clevis mounting and cannot support side loading of the tubes. The actuator is not suitable for use in areas containing explosive dust, vapors, or gases.

The actuator translating tube is keyed to prevent rotation; however, external torques should not be applied to the actuator, as this could result in un-screwing of the tube from the actuator.

2. Installation

The actuator should be connected with a ¼ inch (6.35 mm) steel clevis pin at each end. If using a threaded fastener, be sure that the screw has an unthreaded shank long enough to extend through the actuator clevis.

3. Wiring

All electrical connections are made to the pigtail cable. Refer to the diagrams following for standard models, with and without potentiometer. End of travel limit switches are internally connected and non-adjustable. Motor reversal requires use of a DPDT -center-off switch.

The optional 10K ohm potentiometer may be excited with up to 24 vdc to provide a proportional feedback voltage for various electronic position indicators or controls.

When supplied with optional suffi x "C" plug, LT series actuators may be plugged directly into Duff-Norton LC series controllers, eliminating the need for additional power supplies or switches. See the LC controller catalog for applications.

For non-stock options, including Third Limit Switch, Signal-sending Limit Switch, and Hall Effect Pulse Generator, contact Duff-Norton for appropriate wiring diagram.

DUFF-NORTON ELECTROMECHANICAL ACTUATOR
LIMITATION OF WARRANTIES, REMEDIES AND DAMAGES

The warranty stated below is given in place of all other warranties, express or implied, of merchantability, fi tness for a particular purpose or otherwise. No promise or affi rmation of fact made by any agent or representative of seller shall constitute a warranty by seller or give rise to any liability or obligation.

Seller warrants that on the date of its delivery to carrier the goods are free from defects in workmanship and materials.

Seller's sole obligation in the event of breach of warranty or contract or for negligence or otherwise with respect to goods sold shall be exclusively limited to repair or replacement, f.o.b. seller's point of shipment, of any parts which seller determines to have been defective or if seller determines that such repair or replacement is not feasible, to a refund of the purchase price upon return of the goods to seller.

⚠ WARNING ⚠

The equipment shown in this catalog is intended for industrial use only and should not be used to lift, support, or otherwise transport people unless you have a written statement from Duff-Norton, which authorizes the specifi c actuator used in your applications as suitable for moving people.

NOTE

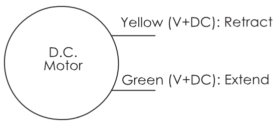
Authors: Approval By: Submission Date:

Version:

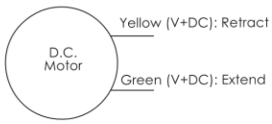
IMS Dynamic Display

Duff-Norton has made every effort to ensure that the information contained in this publication is accurate and reliable. Determining the suitability of our products for specifi c applications is the user's responsibility.

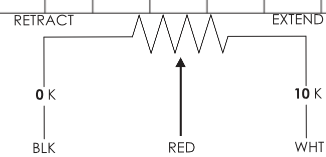
Standard Model



Potentiometer Model



Potentiometer Stroke mm(in.)	50 [1.97]	100 [3.94]	150 [5.91]	200 [7.87]	250 [9.84]	300 [11.81]
POT Travel (% of 10K)	47%	94%	71%	94%	39%	47%



Any action against seller for breach of warranty, negligence or otherwise must be commenced within one year after such cause of action accrues.

No claim against seller for any defect in the goods shall be valid or enforceable unless buyer's written notice thereof is received by seller within one year from the date of shipment.

Seller shall not be liable for any damage, injury or loss arising out of the use of the goods if, prior to such damage, injury or loss, such goods are (1) damaged or misused following seller's delivery to carrier; (2) not maintained, inspected, or used in compliance with applicable law and seller's written instructions and recommendations; or (3) installed, repaired, altered or modifi ed without compliance with such law, instructions or recommendations.

Under no circumstances shall seller be liable for incidental or consequential damages as those terms are defi ned in section 2-715 of the uniform commercial code.

Appendix B: CCW Documentation

Project Controller

This is the body section for a project. This section should contain general information about the project and its properties.

Configuration Micro820

The detailed description of the configuration goes here

Resource Micro820

(* *)

Status: Readable, Modifiable, Deletable

The resource defines 50 variable(s).

Variable _IO_EM_DO_00

(* *)
Direction: VarDirectlyRepresented
Alias: Motor_Running
Data type: BOOL
Attribute: Read/Write
Direct variable (Channel): %QX0.0.0

Variable _IO_EM_DO_01

(* *)
Direction: VarDirectlyRepresented
Data type: BOOL
Attribute: Read/Write
Direct variable (Channel): %QX0.0.1

Variable _IO_EM_DO_02

(* *)
Direction: VarDirectlyRepresented
Alias: Stop_Light
Data type: BOOL
Attribute: Read/Write
Direct variable (Channel): %QX0.0.2

Variable _IO_EM_DO_03

(* *)
Direction: VarDirectlyRepresented
Alias: Run_Light
Data type: BOOL
Attribute: Read/Write
Direct variable (Channel): %QX0.0.3

Variable _IO_EM_DO_04

(* *)
Direction: VarDirectlyRepresented
Alias: Rotate_CW
Data type: BOOL
Attribute: Read/Write
Direct variable (Channel): %QX0.0.4

Variable _IO_EM_DO_05

(* *)
Direction: VarDirectlyRepresented
Data type: BOOL
Attribute: Read/Write
Direct variable (Channel): %QX0.0.5

Variable _IO_EM_DO_06

(* *)
Direction: VarDirectlyRepresented
Alias: Rotate_CCW
Data type: BOOL
Attribute: Read/Write
Direct variable (Channel): %QX0.0.6

Variable _IO_EM_DI_00

(* *)
Direction: VarDirectlyRepresented
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.0

Variable _IO_EM_DI_01

(* *)
Direction: VarDirectlyRepresented
Alias: Start
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.1

Variable _IO_EM_DI_02

(* *)
Direction: VarDirectlyRepresented
Alias: Stop
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.2

Variable _IO_EM_DI_03

(* *)
Direction: VarDirectlyRepresented
Alias: Shift_Up_PB
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.3

Variable _IO_EM_DI_04

(* *)
Direction: VarDirectlyRepresented
Alias: Shift_Down_PB
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.4

Variable _IO_EM_DI_05

(* *)
Direction: VarDirectlyRepresented
Alias: Actuator_Home_LS
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.5

Variable _IO_EM_DI_06

(* *)
Direction: VarDirectlyRepresented
Alias: Actuator_Extended_LS
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.6

Variable _IO_EM_DI_07

(* *)
Direction: VarDirectlyRepresented
Alias: Actuator_Retracted_LS
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.7

Variable _IO_EM_DI_08

(* *)
Direction: VarDirectlyRepresented
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.8

Variable _IO_EM_DI_09

(* *)
Direction: VarDirectlyRepresented
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.9

Variable _IO_EM_DI_10

(* *)
Direction: VarDirectlyRepresented
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.10

Variable _IO_EM_DI_11

(* *)
Direction: VarDirectlyRepresented
Data type: BOOL
Attribute: Read
Direct variable (Channel): %IX0.1.11

Variable _IO_EM_AI_00

(* *)
Direction: VarDirectlyRepresented
Alias: Analog_Input
Data type: WORD
Attribute: Read
Direct variable (Channel): %IW0.2.0

Variable _IO_EM_AI_01

(* *)
Direction: VarDirectlyRepresented
Data type: WORD
Attribute: Read
Direct variable (Channel): %IW0.2.1

Variable _IO_EM_AI_02

(* *)
Direction: VarDirectlyRepresented
Data type: WORD
Attribute: Read
Direct variable (Channel): %IW0.2.2

Variable _IO_EM_AI_03

(* *)
Direction: VarDirectlyRepresented

Data type: WORD
Attribute: Read
Direct variable (Channel): %IW0.2.3

Variable _IO_EM_AO_00

(* *)
Direction: VarDirectlyRepresented
Data type: WORD
Attribute: Read/Write
Direct variable (Channel): %QW0.3.0

Variable Motor_On

(* *)
Direction: Var
Data type: TIME
Attribute: Read/Write

Variable Analog

(* *)
Direction: Var
Data type: UINT
Attribute: Read/Write

Variable ShiftDown_Latch

(* Hold timer while shifting down *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable ShiftUpMax

(* *)
Direction: Var
Data type: DINT
Attribute: Read/Write

Variable ShiftUp_Latch

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable __SYSVA_CYCLECNT

(* Cycle counter *)
Direction: VarGlobal
Data type: DINT

Attribute: Read

Variable __SYSVA_CYCLEDATE

(* Timestamp of the beginning of the cycle in milliseconds (ms) *)
Direction: VarGlobal
Data type: TIME
Attribute: Read

Variable __SYSVA_KVBPERR

(* Kernel variable binding producing error (production error) *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read

Variable __SYSVA_KVBCERR

(* Kernel variable binding consuming error (consumption error) *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read/Write

Variable __SYSVA_RESNAME

(* Resource name (max length=255) *)
Direction: VarGlobal
Data type: STRING
Attribute: Read

Variable __SYSVA_SCANCNT

(* Input scan counter *)
Direction: VarGlobal
Data type: DINT
Attribute: Read

Variable __SYSVA_TCYCYCTIME

(* Programmed cycle time *)
Direction: VarGlobal
Data type: TIME
Attribute: Read/Write

Variable __SYSVA_TCYCURRENT

(* Current cycle time *)
Direction: VarGlobal
Data type: TIME
Attribute: Read

Variable __SYSVA_TCYMAXIMUM

(* Maximum cycle time since last start *)
Direction: VarGlobal
Data type: TIME
Attribute: Read

Variable __SYSVA_TCYOVERFLOW

(* Number of cycle overflows *)
Direction: VarGlobal
Data type: DINT
Attribute: Read

Variable __SYSVA_RESMODE

(* Resource execution mode *)
Direction: VarGlobal
Data type: SINT
Attribute: Read

Variable __SYSVA_CCEXEC

(* Execute one cycle when application is in cycle to cycle mode *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read/Write

Variable __SYSVA_REMOTE

(* Remote status *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read

Variable __SYSVA_SUSPEND_ID

(* Last Suspend ID *)
Direction: VarGlobal
Data type: UINT
Attribute: Read

Variable __SYSVA_TCYWDG

(* Software Watchdog *)
Direction: VarGlobal
Data type: UDINT
Attribute: Read/Write

Variable __SYSVA_MAJ_ERR_HALT

(* Major Error Halted status *)
Direction: VarGlobal

IMS Dynamic Display

Data type: BOOL
Attribute: Read

Variable __SYSVA_ABORT_CYCLE

(* Aborting Cycle *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read

Variable __SYSVA_FIRST_SCAN

(* First scan bit *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read

Variable __SYSVA_USER_DATA_LOST

(* User data lost *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read/Write

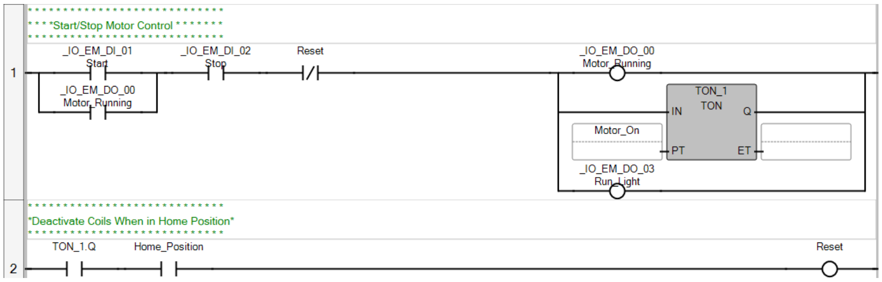
Variable __SYSVA_POWERUP_BIT

(* Power-up bit *)
Direction: VarGlobal
Data type: BOOL
Attribute: Read

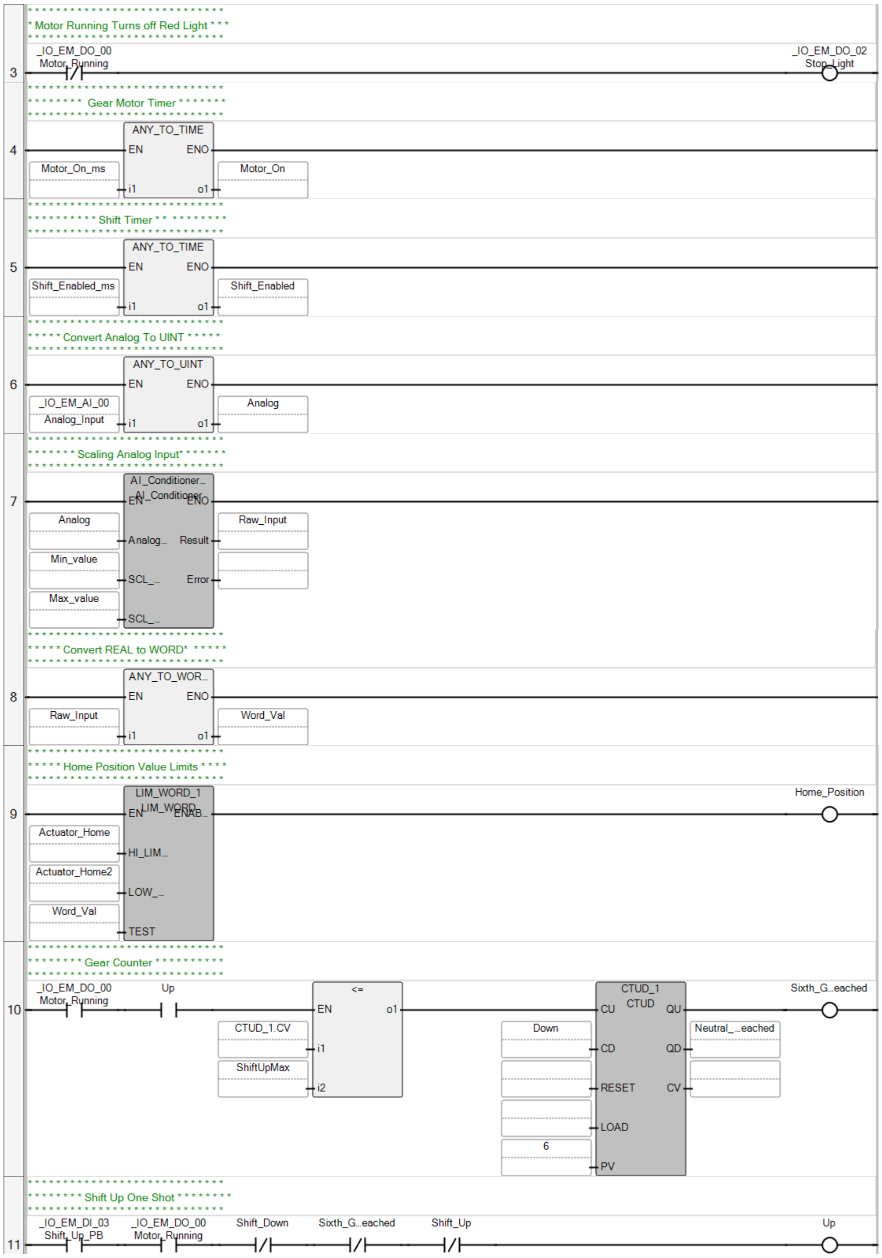
Variable __SYSVA_PROJ_INCOMPLETE

(* Project Incomplete *)
Direction: VarGlobal
Data type: UDINT
Attribute: Read

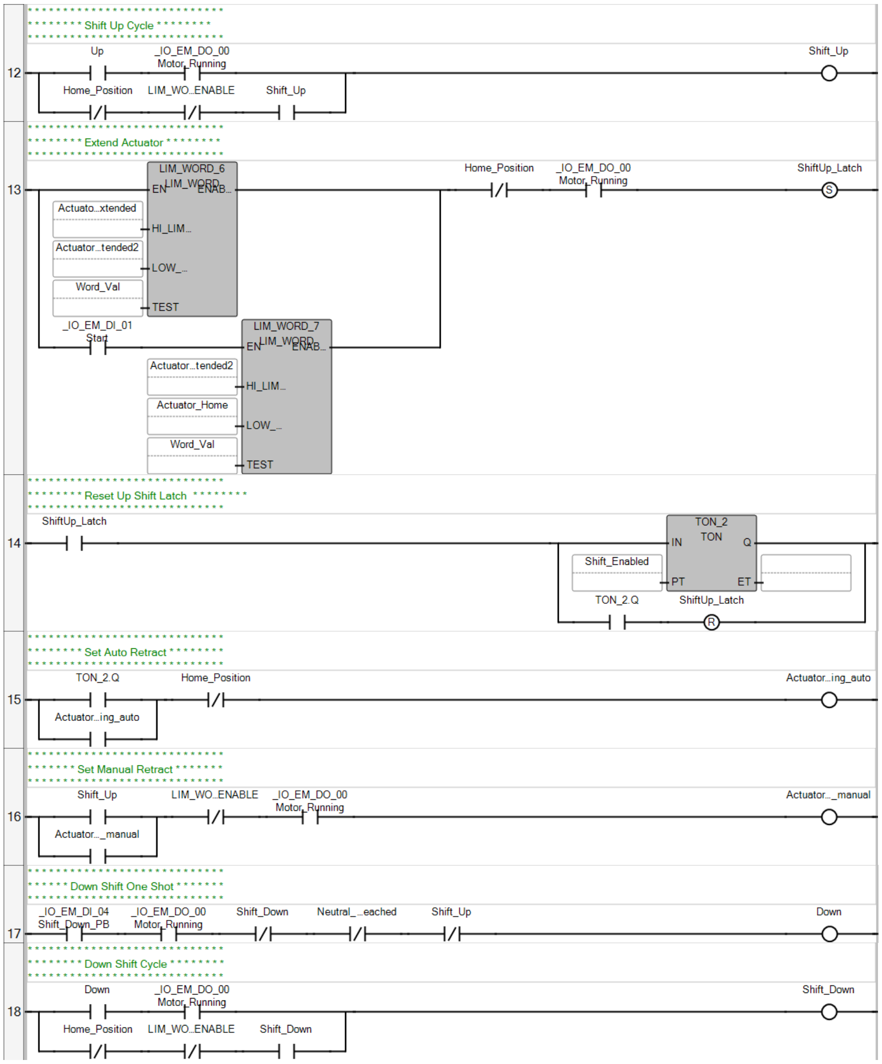
Controller.Micro820.Micro820.Prog1

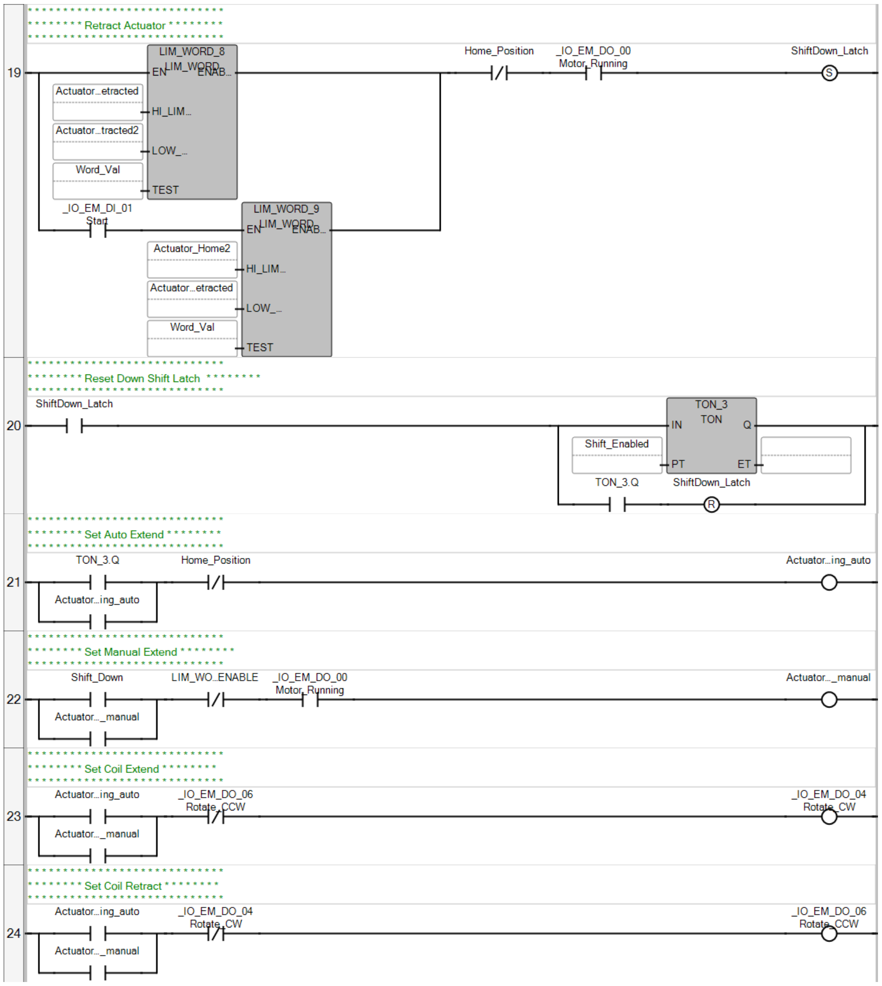


IMS Dynamic Display



IMS Dynamic Display





POU Prog1

The POU defines 45 variable(s).

Variable Start

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

IMS Dynamic Display

Variable Motor_On

(* *)
Direction: Var
Data type: TIME
Attribute: Read/Write

Variable CTUD_1

(* *)
Direction: Var
Data type: CTUD
Attribute: Read/Write

Variable Motor_On_ms

(* *)
Direction: Var
Data type: DINT
Attribute: Read/Write

Variable Sixth_Gear_Reached

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Neutral_Gear_Reached

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Shift_Up

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Shift_Down

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Up

(* *)
Direction: Var

Data type: BOOL
Attribute: Read/Write

Variable Down

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Actuator_Extending_auto

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Actuator_Extending_manual

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Actuator_Retracting_auto

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Actuator_Retracting_manual

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Raw_Input

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Variable Actuator_Extended

(* *)
Direction: Var
Data type: WORD
Attribute: Read/Write

Variable Actuator_Retracted

(* *)
Direction: Var
Data type: WORD
Attribute: Read/Write

Variable Actuator_Home

(* *)
Direction: Var
Data type: WORD
Attribute: Read/Write

Variable Home_Position

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Shift_Enabled_ms

(* *)
Direction: Var
Data type: INT
Attribute: Read/Write

Variable Shift_Enabled

(* *)
Direction: Var
Data type: TIME
Attribute: Read/Write

Variable Max_value

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Variable Min_value

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Variable Actuator_Home2

(* *)
Direction: Var

Data type: WORD
Attribute: Read/Write

Variable LIM_WORD_1

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Variable Word_Val

(* *)
Direction: Var
Data type: WORD
Attribute: Read/Write

Variable LIM_WORD_2

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Variable Actuator_Extended2

(* *)
Direction: Var
Data type: WORD
Attribute: Read/Write

Variable Actuator_Retracted2

(* *)
Direction: Var
Data type: WORD
Attribute: Read/Write

Variable LIM_WORD_3

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Variable AI_Conditioner_1

(* *)
Direction: Var
Data type: AI_Conditioner
Attribute: Read/Write

Variable LIM_WORD_4

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Variable LIM_WORD_5

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Variable TON_1

(* *)
Direction: Var
Data type: TON
Attribute: Read/Write

Variable TON_2

(* *)
Direction: Var
Data type: TON
Attribute: Read/Write

Variable TON_3

(* *)
Direction: Var
Data type: TON
Attribute: Read/Write

Variable LIM_WORD_7

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Variable LIM_WORD_8

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Variable LIM_WORD_9

(* *)
Direction: Var

Data type: LIM_WORD
Attribute: Read/Write

Variable Reset

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable Input_Min_value

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Variable Input_Max_value

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Variable Output_Min

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Variable Output_Max

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Variable LIM_WORD_6

(* *)
Direction: Var
Data type: LIM_WORD
Attribute: Read/Write

Controller.Micro820.Micro820.AI_Conditioner

```
Value_4mA := 0.0;
Value_20mA := 4095.0;
AnalogInput_Real := Any_To_Real(AnalogInput);

IF AnalogInput_Real > Value_4mA AND AnalogInput_Real < Value_20mA THEN
```

```
        Error := FALSE;
ELSE
        Error := TRUE;
END_IF;
SCALE_AI(AnalogInput_Real,Value_4mA, Value_20mA, SCL_Min, SCL_Max);
Result := Scale_AI.Output;
```

POU AI_Conditioner

The POU defines 9 variable(s).

Variable AnalogInput

(* *)
Direction: VarInput
Data type: UINT
Attribute: Read

Variable SCL_Min

(* *)
Direction: VarInput
Data type: REAL
Attribute: Read

Variable SCL_Max

(* *)
Direction: VarInput
Data type: REAL
Attribute: Read

Variable Result

(* *)
Direction: VarOutput
Data type: REAL
Attribute: Write

Variable Error

(* *)
Direction: VarOutput
Data type: BOOL
Attribute: Write

Variable Value_4mA

(* *)
Direction: Var

IMS Dynamic Display

Data type: REAL
Attribute: Read/Write

Variable Value_20mA

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

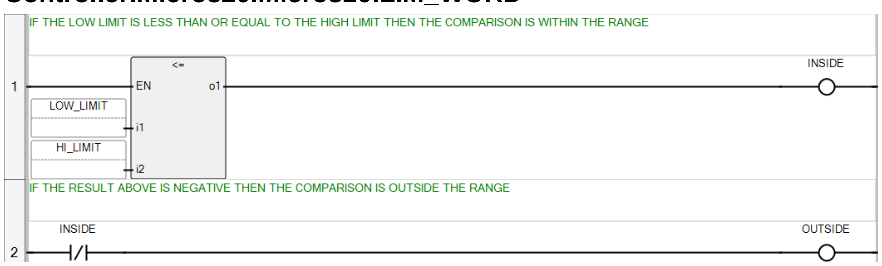
Variable SCALE_AI

(* *)
Direction: Var
Data type: SCALER
Attribute: Read/Write

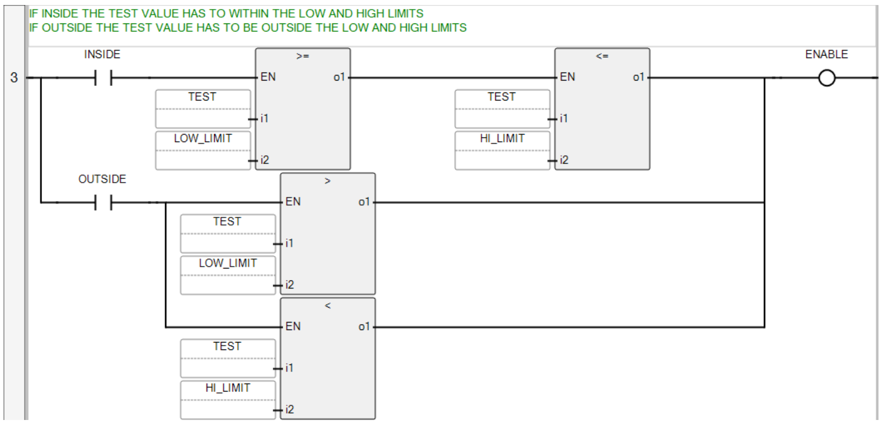
Variable AnalogInput_Real

(* *)
Direction: Var
Data type: REAL
Attribute: Read/Write

Controller.Micro820.Micro820.LIM_WORD



IMS Dynamic Display



POU LIM_WORD

The POU defines 6 variable(s).

Variable HI_LIMIT

(* *)
Direction: VarInput
Data type: WORD
Attribute: Read

Variable LOW_LIMIT

(* *)
Direction: VarInput
Data type: WORD
Attribute: Read

Variable TEST

(* *)
Direction: VarInput
Data type: WORD
Attribute: Read

Variable ENABLE

(* *)
Direction: VarOutput
Data type: BOOL
Attribute: Write

Variable INSIDE

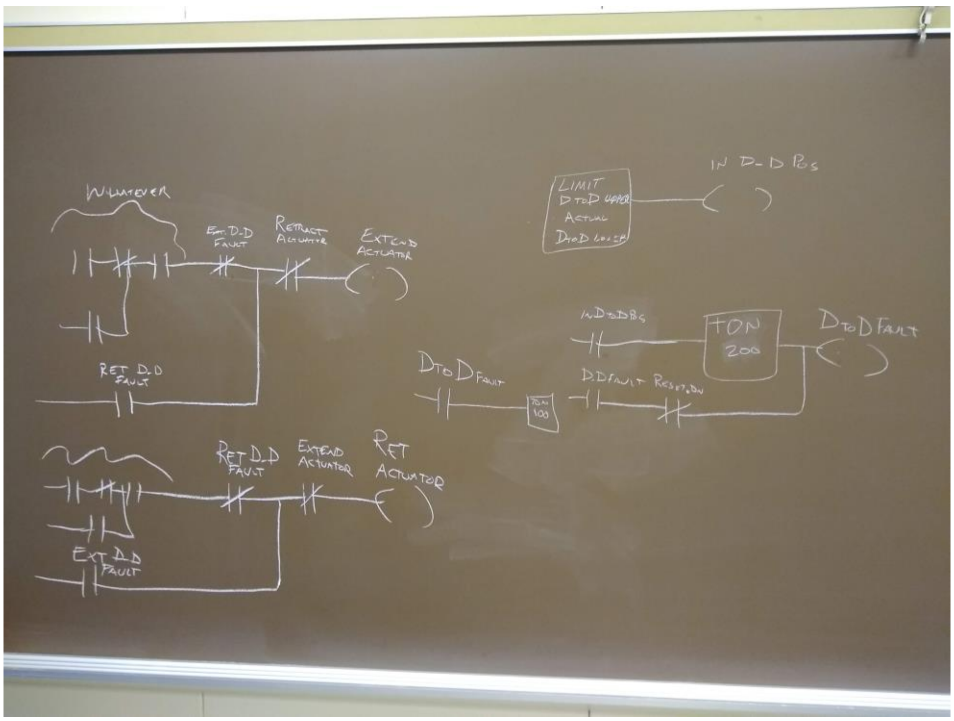
(* *)

IMS Dynamic Display

Direction: Var
Data type: BOOL
Attribute: Read/Write

Variable OUTSIDE

(* *)
Direction: Var
Data type: BOOL
Attribute: Read/Write



Appendix C: Resources

http://www.xtrac.com/the_company.php