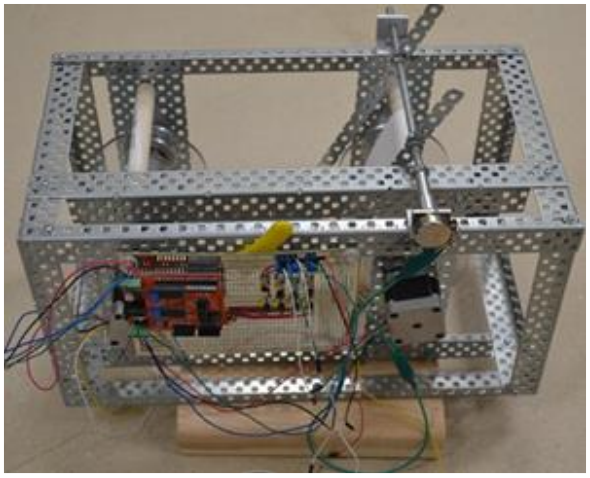
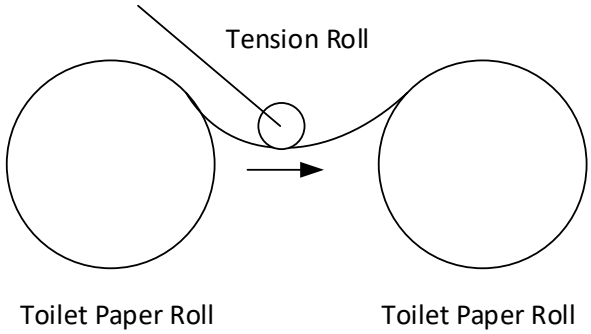
Chapter 23 Tape Rewind – PID+

Toilet Paper Rewind Lab

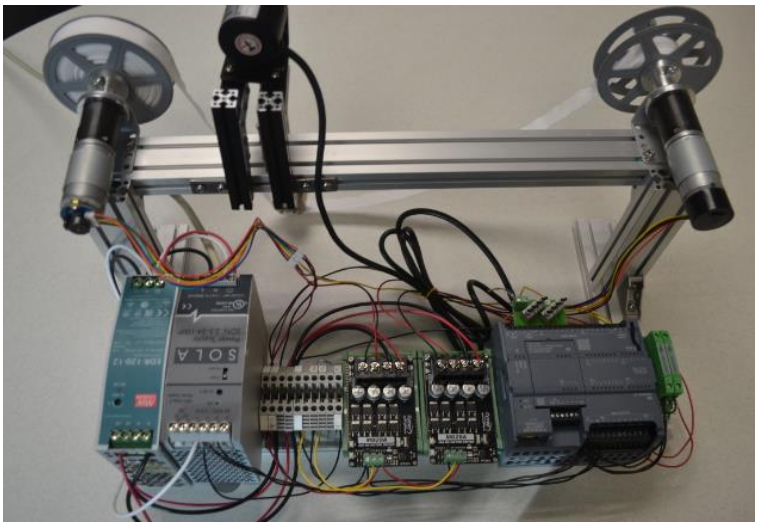
The Tape Rewind Lab began with the idea of rolling up toilet paper. A bad idea. The stuff breaks with the least tug and the system used stepper motors, not bad but controlled only in discrete angle increments. The lab was designed, built and programmed. It worked kinda but not well enough to continue with the design.





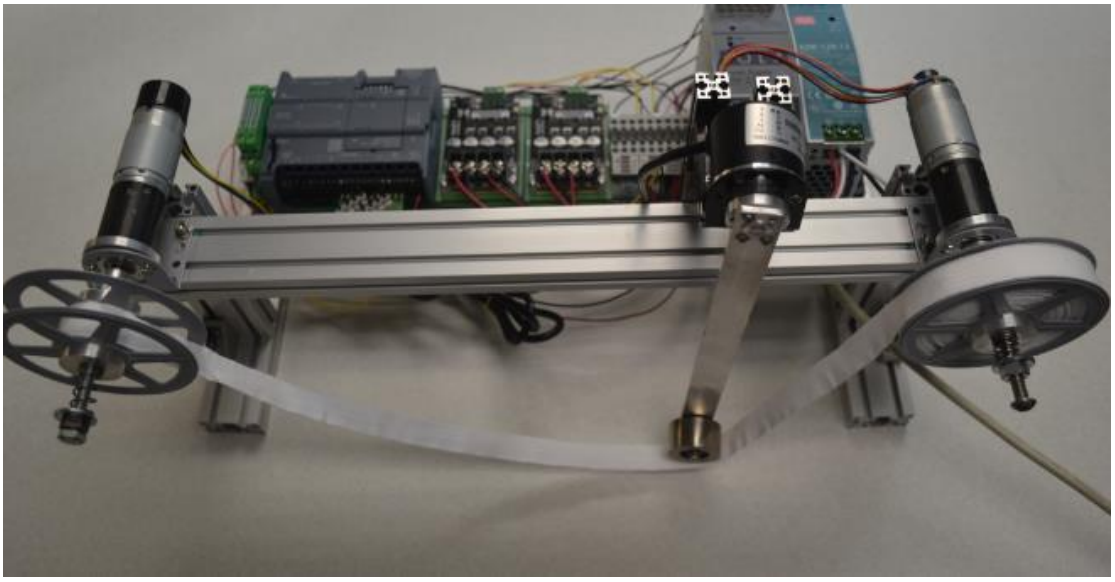
The design of a two-axis motion system with feedback control between the two has been an objective of a lab experience for some time. This lab involved two stepper motors and a PID loop feedback between the two. Tension control is to be maintained. It is anticipated that many rolls of toilet paper will find the bottom of a waste can because of this lab.

While the stepper lab was successful to the point that a MS student received his degree from programming it, the design was not satisfactory. The numerous problems keeping this system operational was enough to cause the construction of the winder shown below.

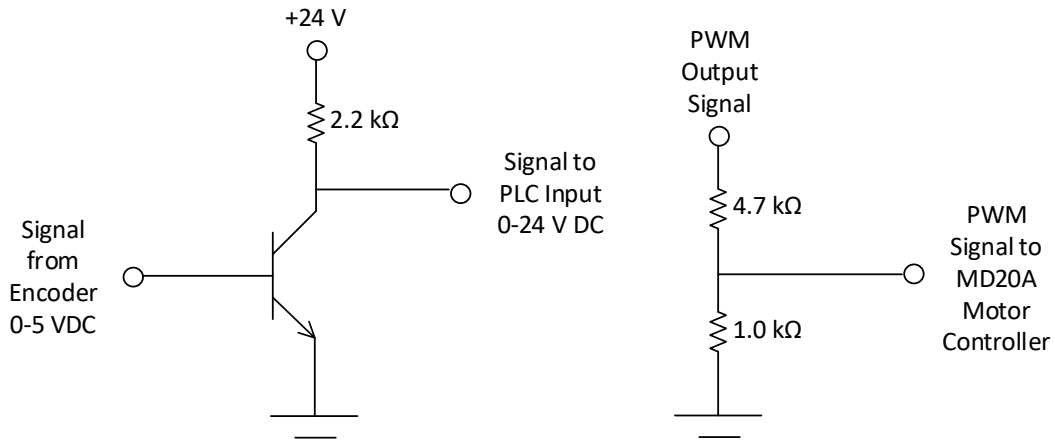


Later Motor Speed Control Design

This design uses two dc gear motors to transfer a cloth tape from one reel to another with a dancer roll in the middle. This design has the advantage of position and speed control of two motors and the PID control of the tension with the dancer roll.



This design was developed before the design in Ch. 21 – Single Axis Gear Motor Speed Control. This diagram uses 24 V inputs instead of the 5 V input for the encoder.



The wiring for the motor from the PLC and motor drive board is shown below:

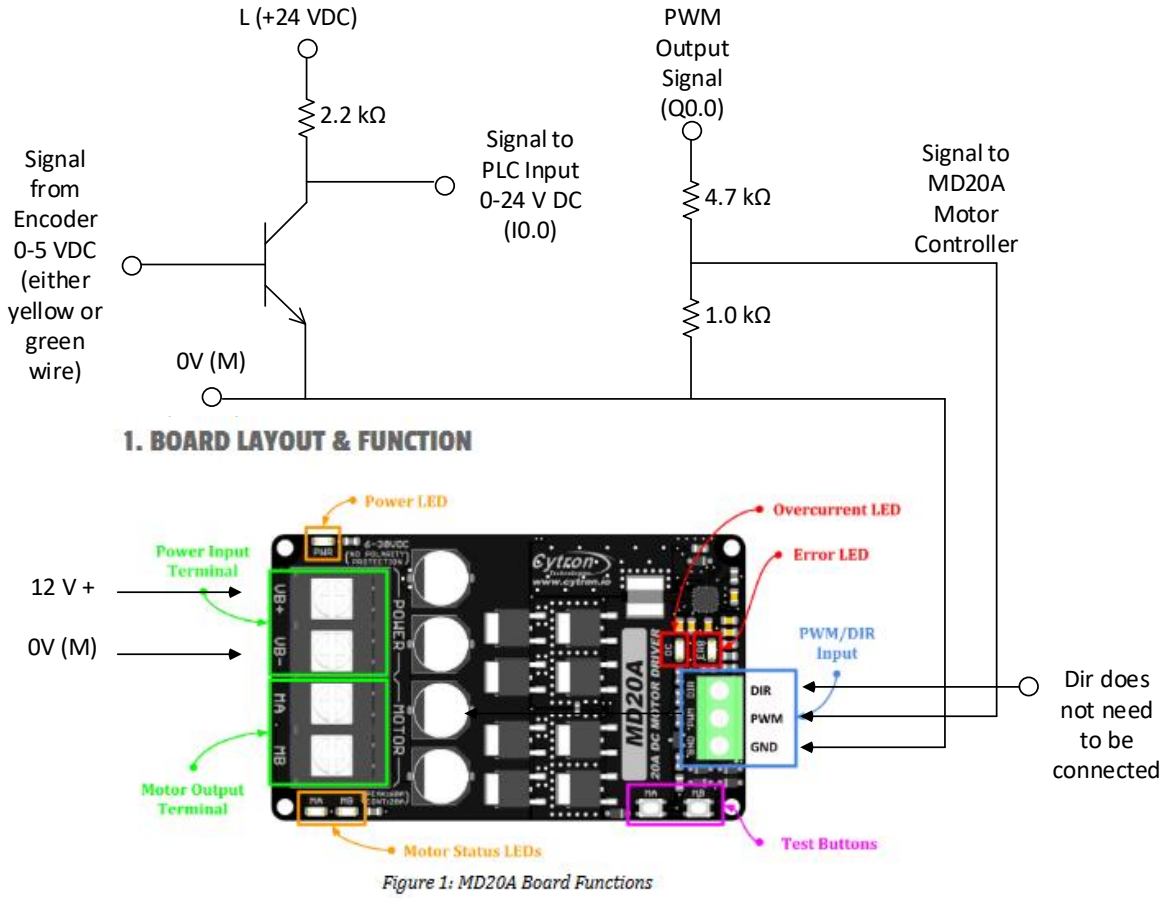


Figure 1: MD20A Board Functions

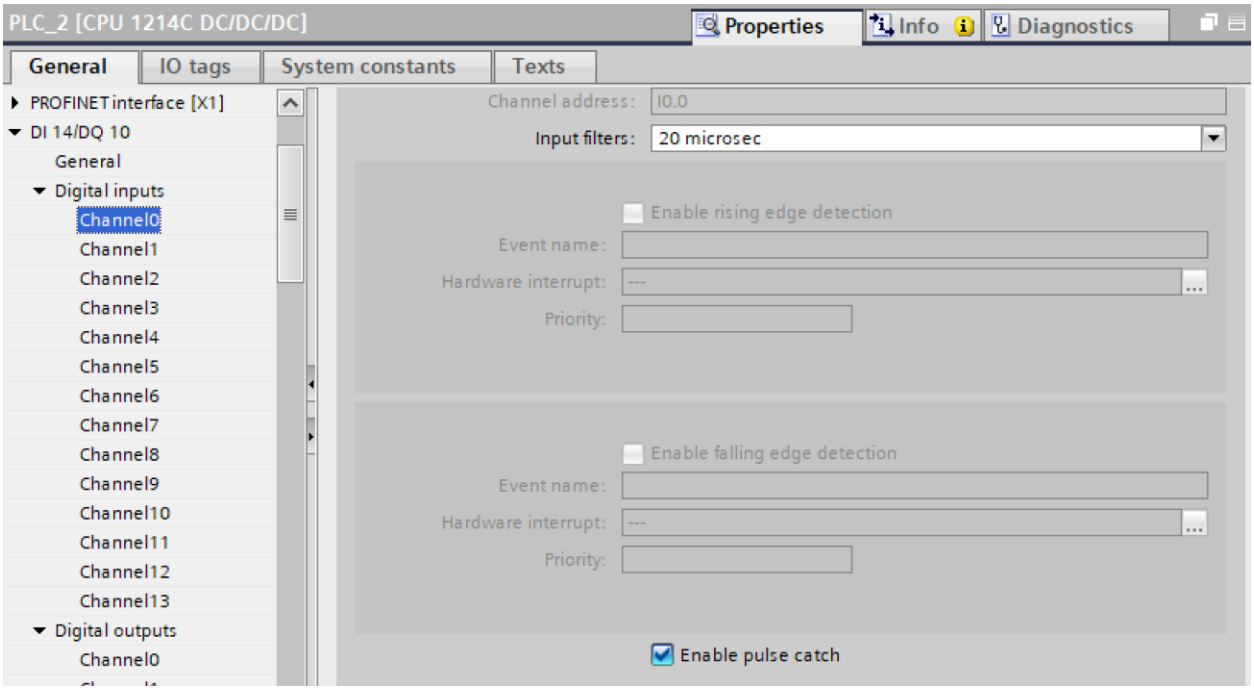
Separate wiring acquires the POT information from the dancer arm. While this lab in all regards is a double of the previous Lab 21, there are several changes. First, the need for position control may be needed and is not added at present. The design of this lab pre-dates information for Lab 21 in that the 5 V interface was not known at that time. From this lab, that information was obtained and that lab was built. Also, this lab has several possible outcomes:

1. The second motor can run as a percent of the first motor and a percent of the POT.
2. The second motor can run independent of the first motor at a set ramp speed.
3. Both motors can run independent and be separately ramped.
4. The second motor can run as a direct follower of the POT only.

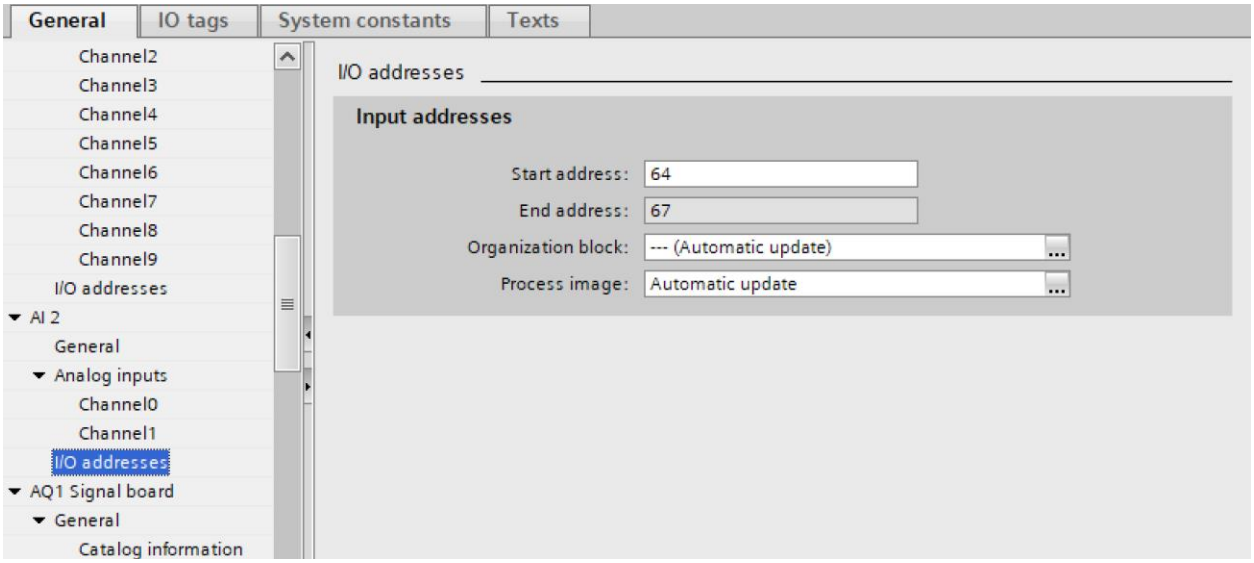
Any of these schemes are acceptable under certain criteria. Any could be used by industry.

Since this is a process, the inputs for starting and stopping the system plus each motor independently have been added. These are I0.6 and I0.7 for the motors.

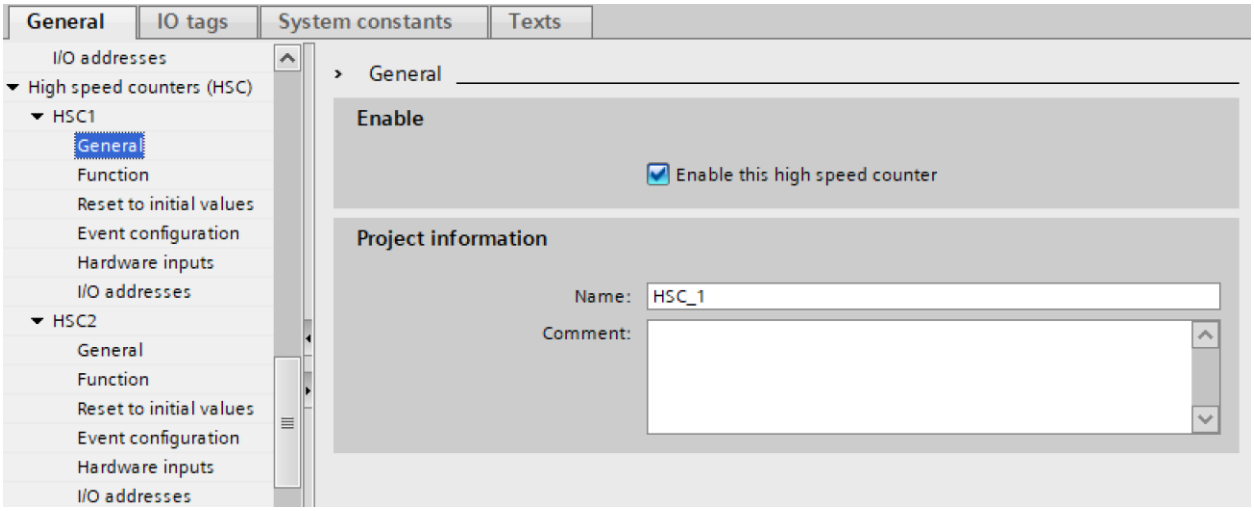
Configuration of Digital Inputs for high-speed pulse input:

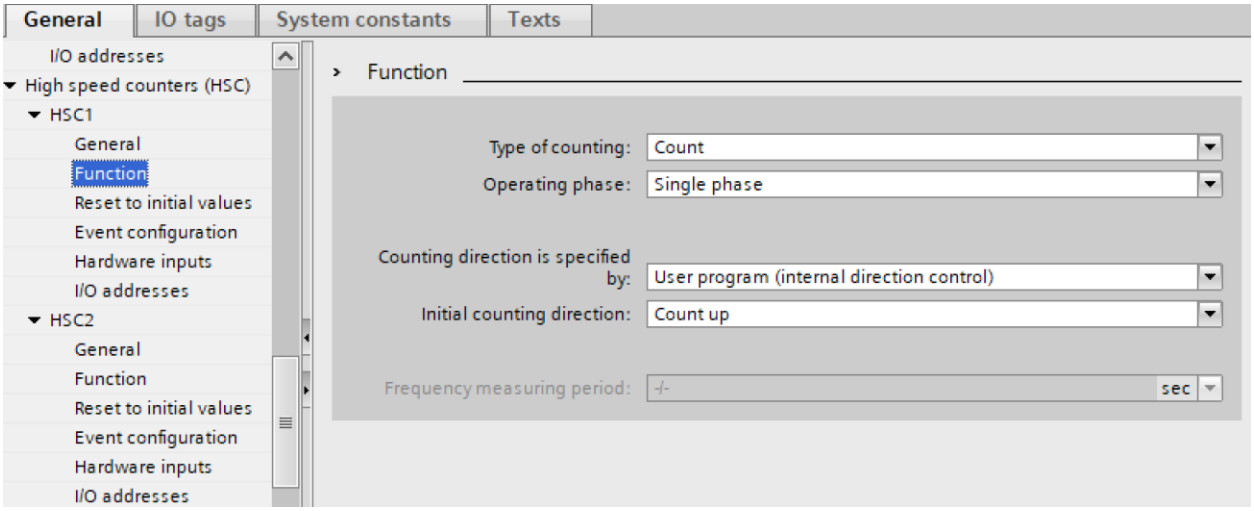


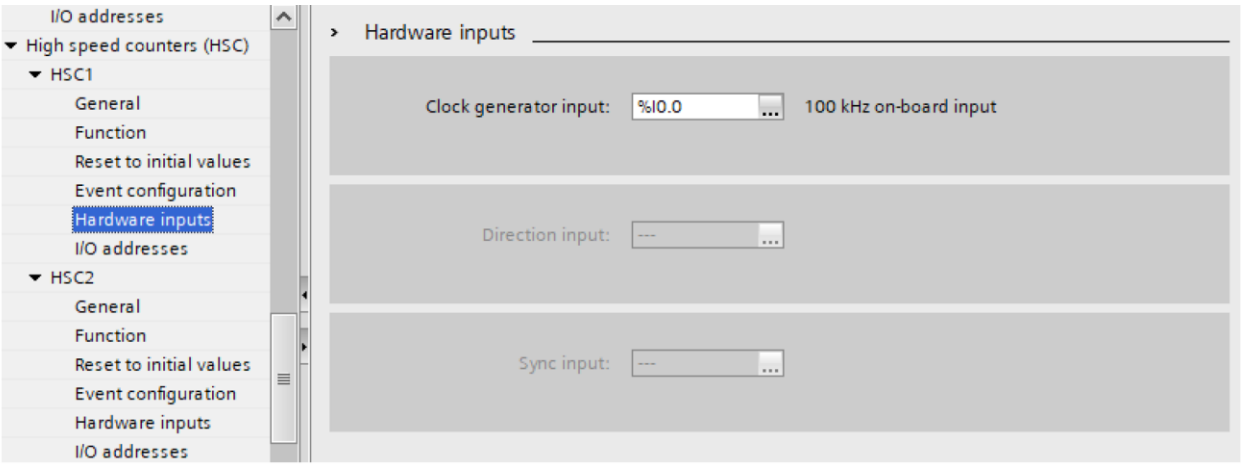
Analog address of the POT when added to the design. The POT is not presently included in the program.



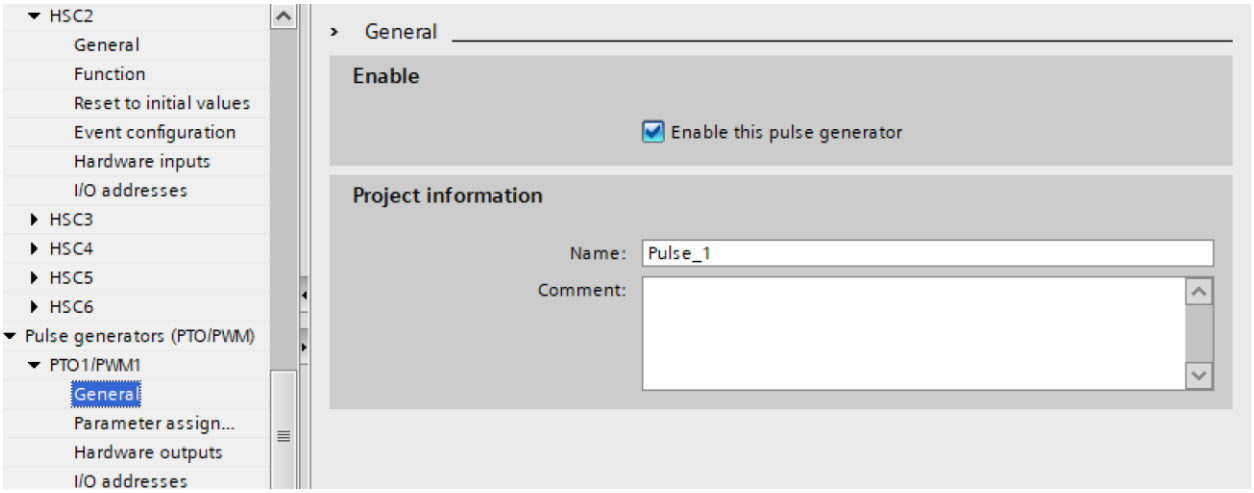
Configuration of the High Speed Counter:

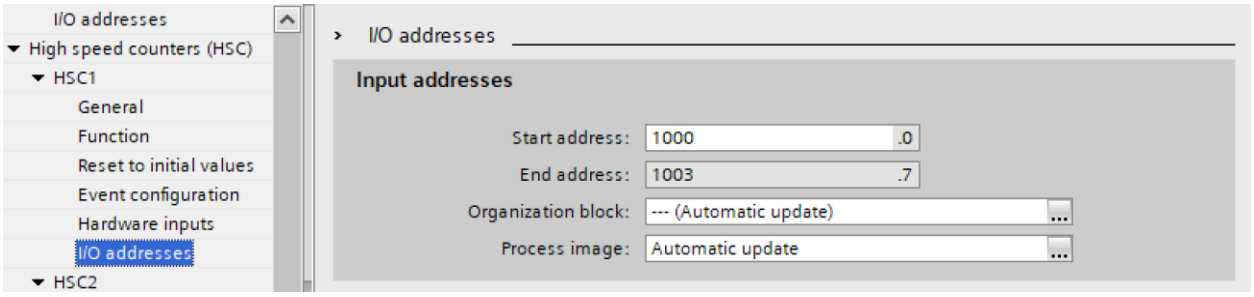


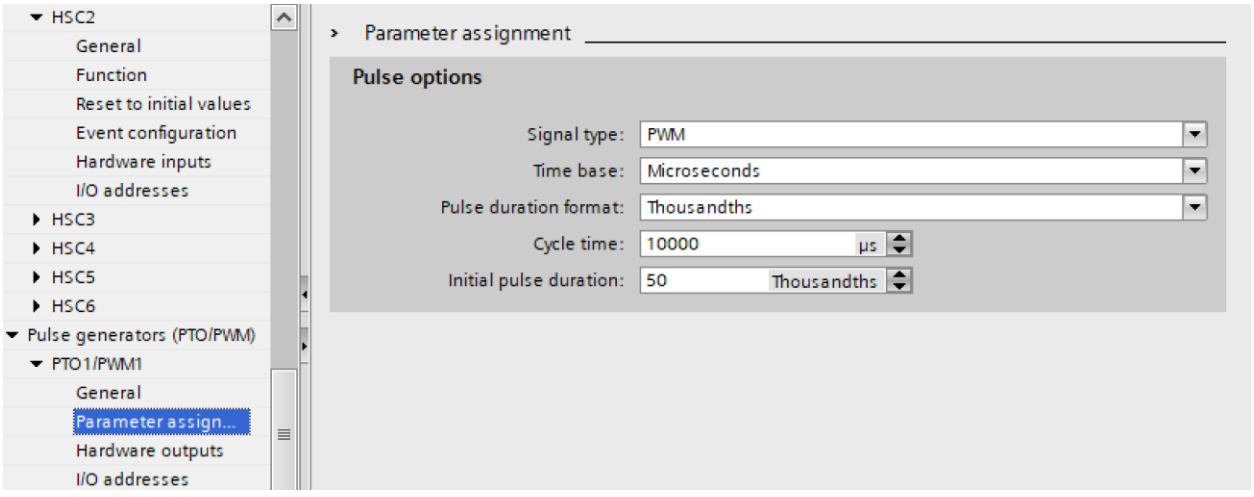


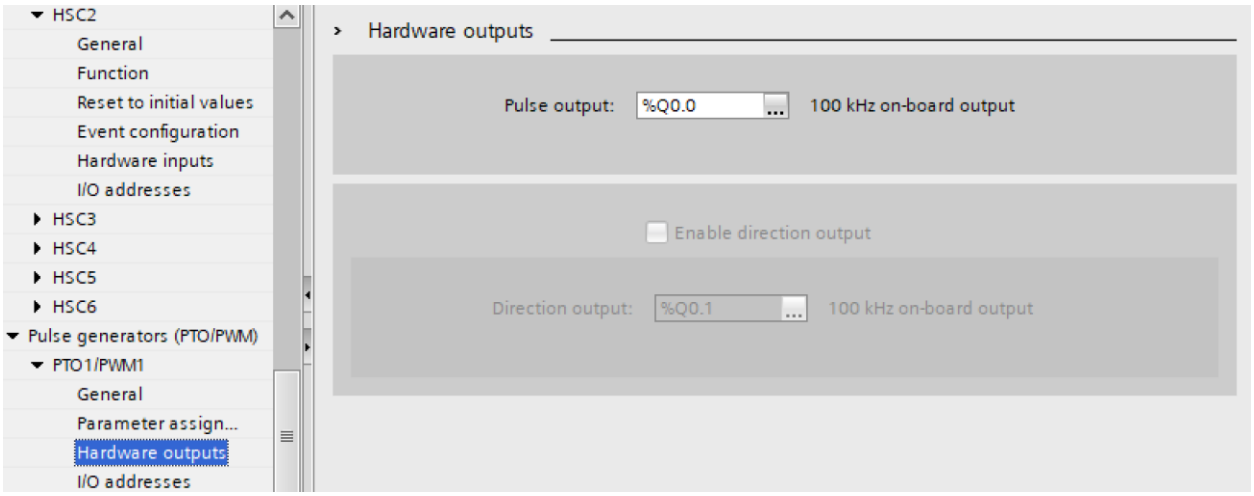


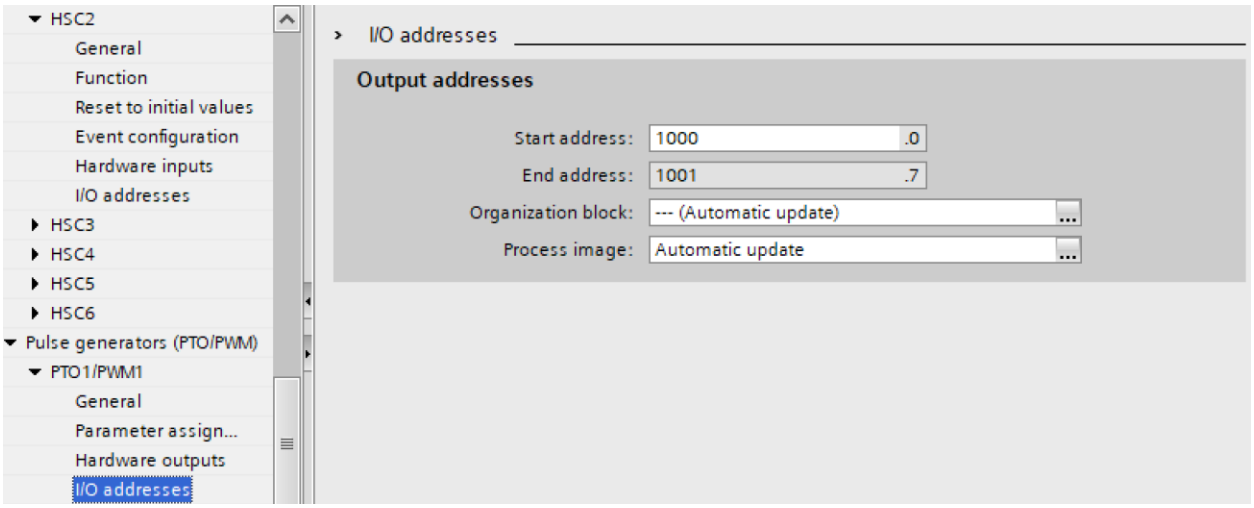
Configuration of the PWM Outputs:

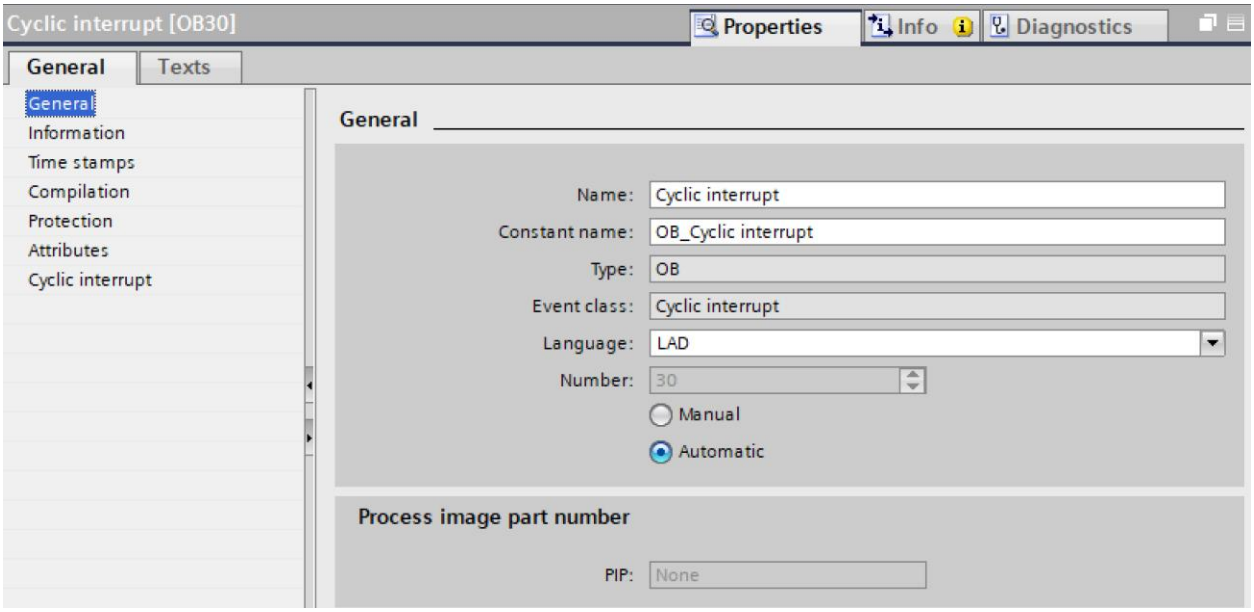


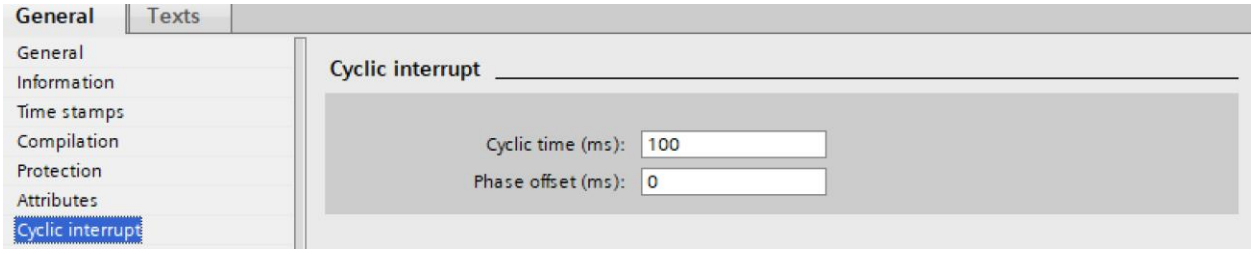




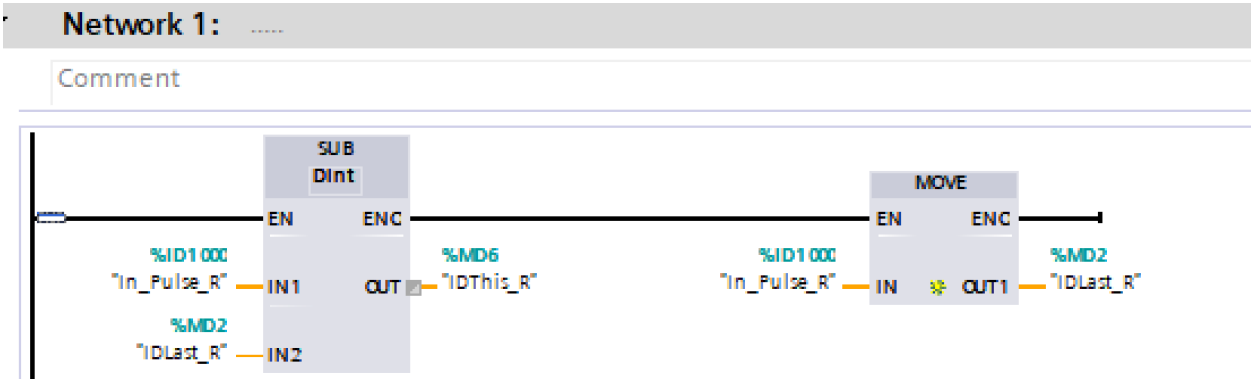

HSC2
General
Function
Reset to initial values
Event configuration
Hardware inputs
I/O addresses
HSC3
HSC4
HSC5
HSC6
Pulse generators (PTO/PWM)
PTO1/PWM1
General
Parameter assign...
Hardware outputs
I/O addresses
Hardware outputs
Pulse output:
%Q0.0
100 kHz on-board output
Enable direction output
Direction output:
%Q0.1
100 kHz on-board output

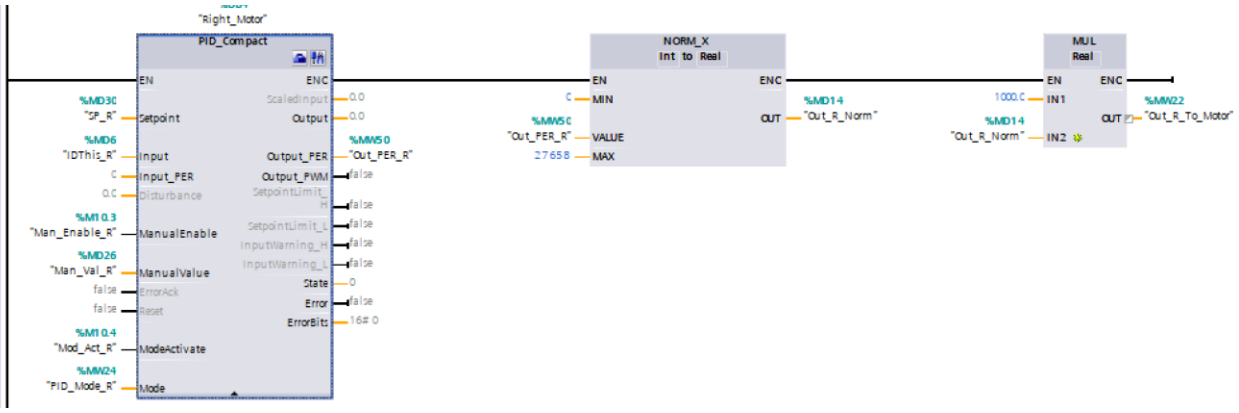

HSC2
General
Function
Reset to initial values
Event configuration
Hardware inputs
I/O addresses
HSC3
HSC4
HSC5
HSC6
Pulse generators (PTO/PWM)
PTO1/PWM1
General
Parameter assign...
Hardware outputs
I/O addresses
I/O addresses
Output addresses
Start address:
1000
.0
End address:
1001
.7
Organization block:
--- (Automatic update)
Process image:
Automatic update

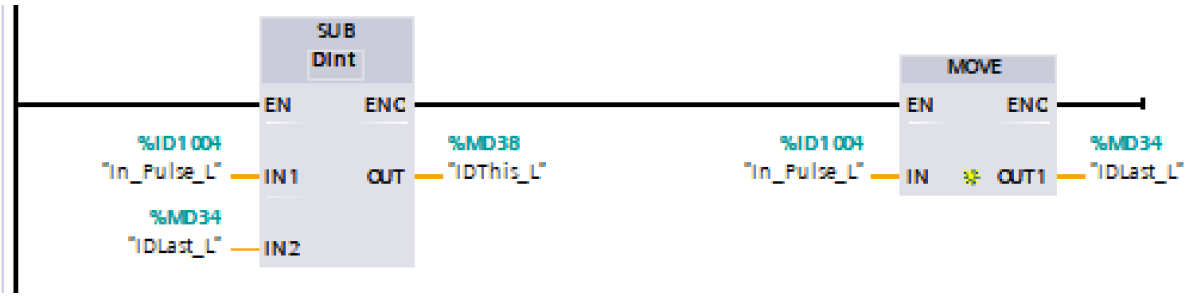


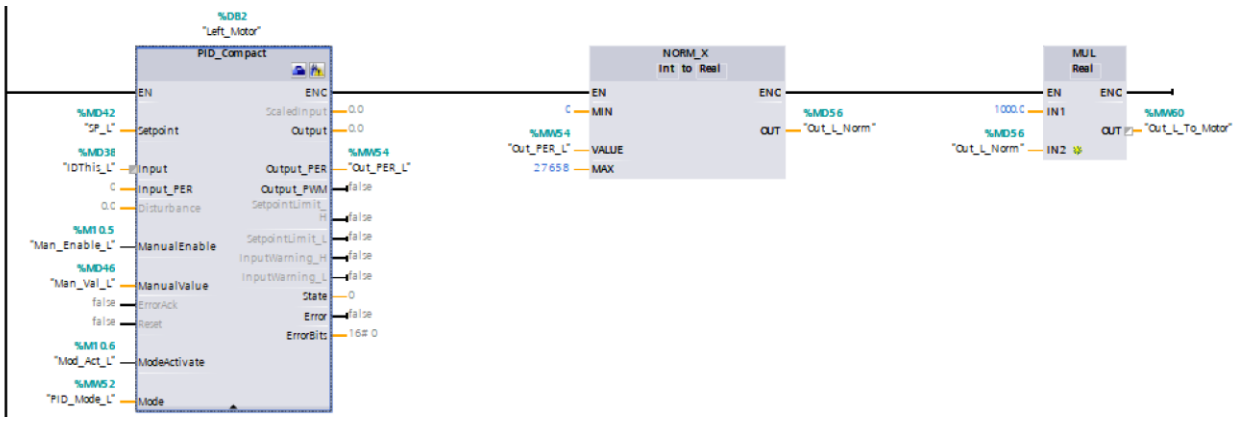


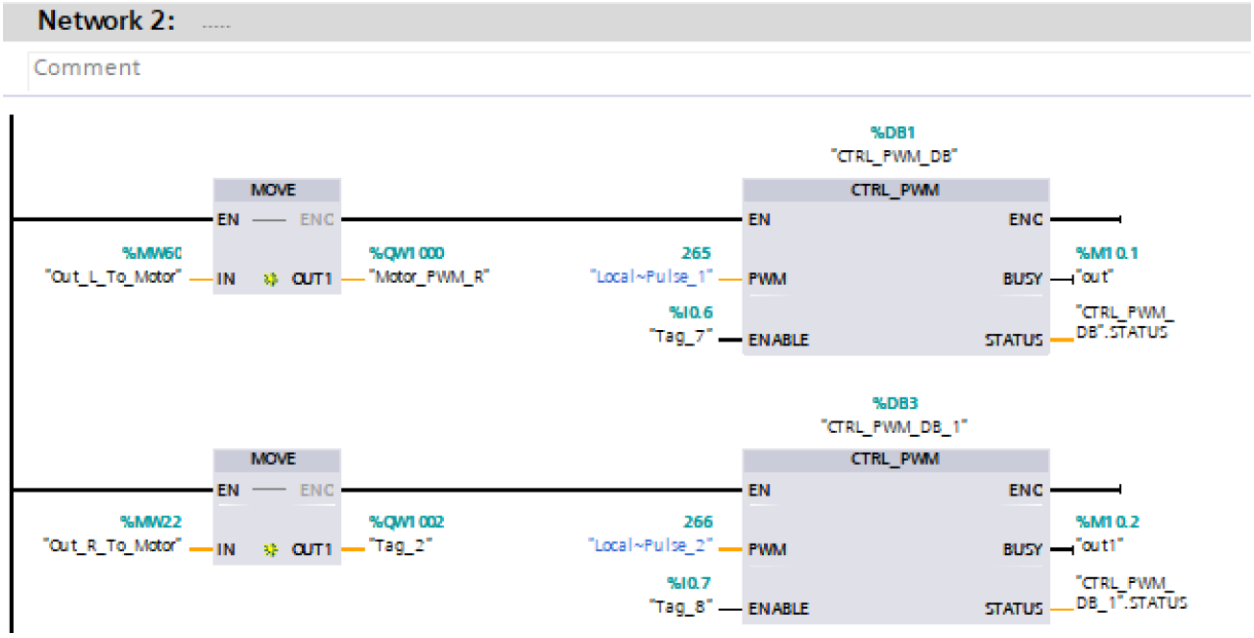
The programming is next. There are many statements to be added to complete the actual program desired. The suggestions listed above may be used as a starting point for your program.



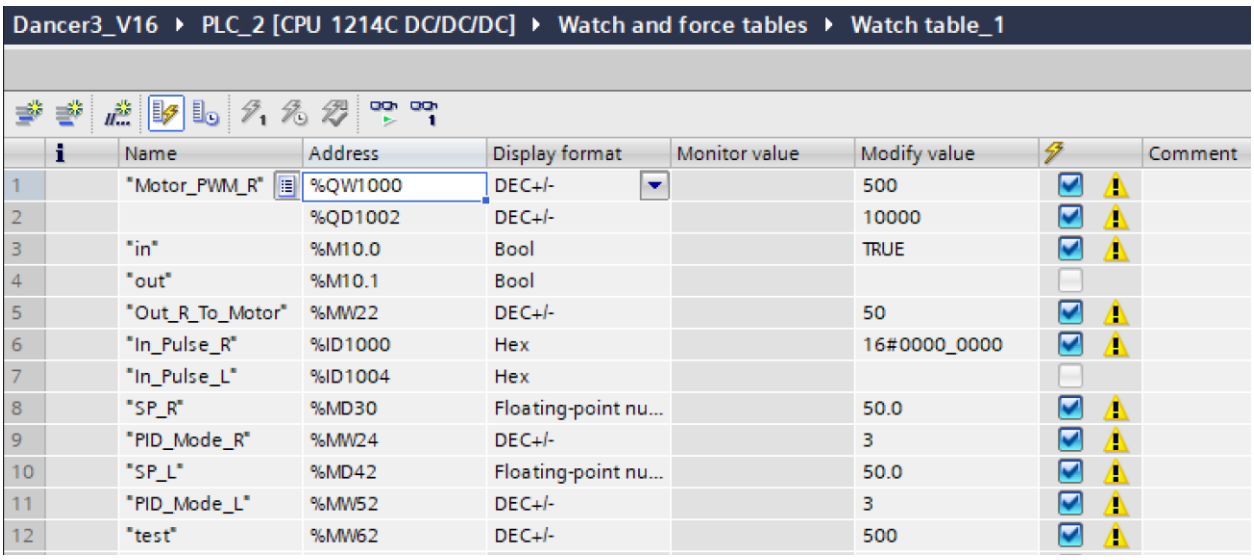

"Right_Motor"
PID_Compact
EN
ENC
%MD30
"SP_R"
Setpoint
ScaledInput
Output
%MD6
"IDThis_R"
Input
Output_PER
%MW50
"Out_PER_R"
Input_PER
Output_PWM
Disturbance
SetpointLimit_H
%M10.3
"Man_Enable_R"
ManualEnable
SetpointLimit_L
InputWarning_H
%MD26
"Man_Val_R"
ManualValue
InputWarning_L
ErrorAck
State
Reset
Error
ErrorBits
%M10.4
"Mod_Act_R"
ModeActivate
%MW24
"PID_Mode_R"
Mode
NORM_X
Int to Real
MIN
%MW58
"Out_PER_R"
VALUE
27658
MAX
%MD14
"Out_R_Norm"
MUL
Real
IN1
IN2
%MW22
"Out_R_To_Motor"


SUB
DInt
EN
ENC
%ID1004
"In_Pulse_L"
IN1
OUT
%MD38
"IDThis_L"
%MD34
"IDLast_L"
IN2
MOVE
IN
OUT1


%DB2
"Left_Motor"
PID_Compact
EN
ENC
%MD42
"SP_L"
Setpoint
ScaledInput
Output
%MD38
"IDThis_L"
Input
Output_PER
%MW54
"Out_PER_L"
Input_PER
Output_PWM
Disturbance
SetpointLimit_H
%M10.5
"Man_Enable_L"
ManualEnable
SetpointLimit_L
InputWarning_H
%MD46
"Man_Val_L"
ManualValue
InputWarning_L
ErrorAck
State
Reset
Error
ErrorBits
%M10.6
"Mod_Act_L"
ModeActivate
%MW52
"PID_Mode_L"
Mode
NORM_X
Int to Real
MIN
%MW54
"Out_PER_L"
VALUE
27658
MAX
%MD56
"Out_L_Norm"
MUL
Real
IN1
IN2
%MW60
"Out_L_To_Motor"



The watch table below was used in troubleshooting the proram above. There may be additions necessary as you complete the program.



	Name	Address	Display format	Monitor value	Modify value	Comment
1	"Motor_PWM_R"	%QW1000	DEC+/-		500	
2		%QD1002	DEC+/-		10000	
3	"in"	%M10.0	Bool		TRUE	
4	"out"	%M10.1	Bool			
5	"Out_R_To_Motor"	%MW22	DEC+/-		50	
6	"In_Pulse_R"	%ID1000	Hex		16#0000_0000	
7	"In_Pulse_L"	%ID1004	Hex			
8	"SP_R"	%MD30	Floating-point nu...		50.0	
9	"PID_Mode_R"	%MW24	DEC+/-		3	
10	"SP_L"	%MD42	Floating-point nu...		50.0	
11	"PID_Mode_L"	%MW52	DEC+/-		3	
12	"test"	%MW62	DEC+/-		500	