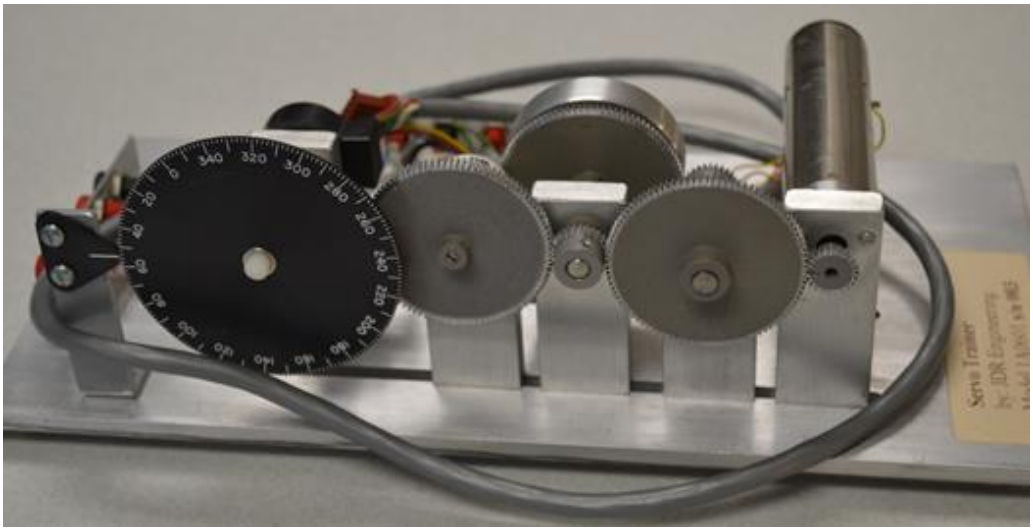
Chapter 21 Gear Motor Speed Control - PID

DC Motor Position Control Lab

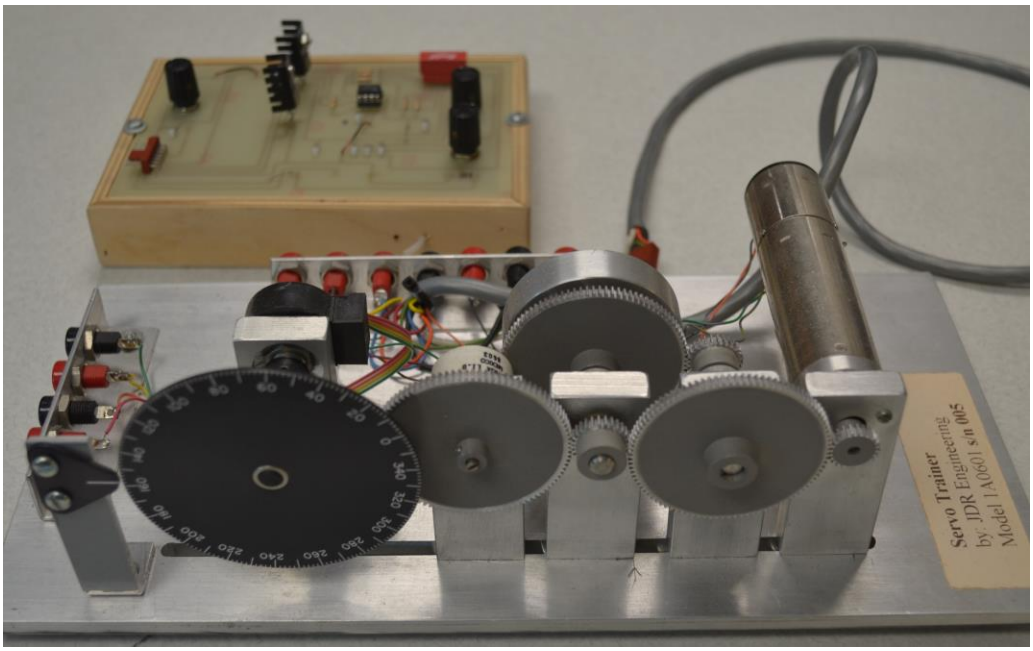
This lab has a history from a number of previous developers. One was the original lab for the Auto Controls course manufactured by a former instructor in EET, John Rich. Its purpose was to control speed and then position using PID loops. The motor at right can be run both forward and reverse and to a position. The lab was the highlight of the AutoControl course by Mr. Rich during his tenure.

Prof. Rich's design is shown here:

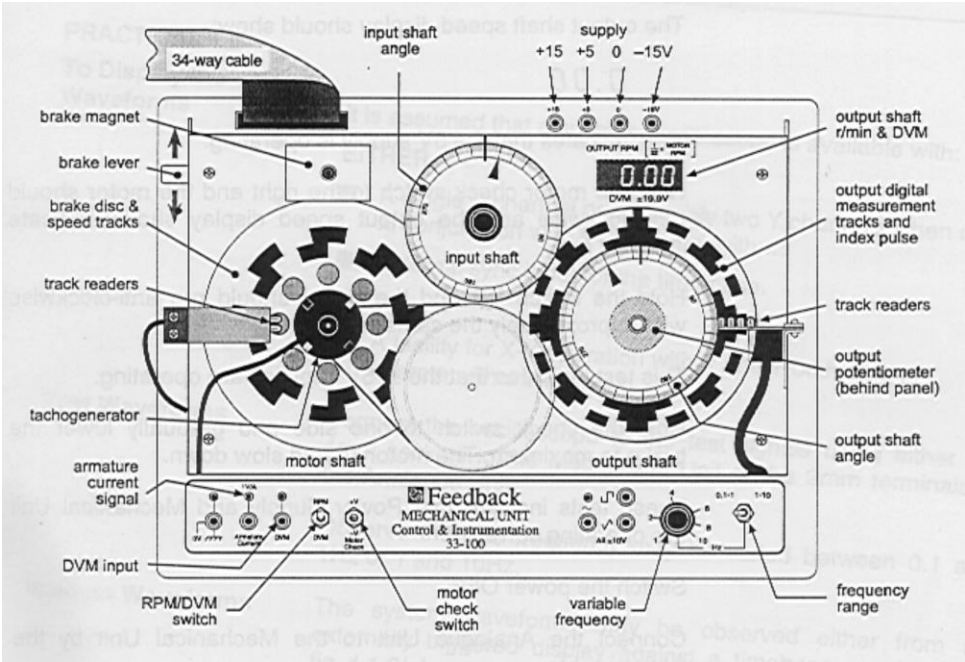


Speed and Position Control of DC Motor

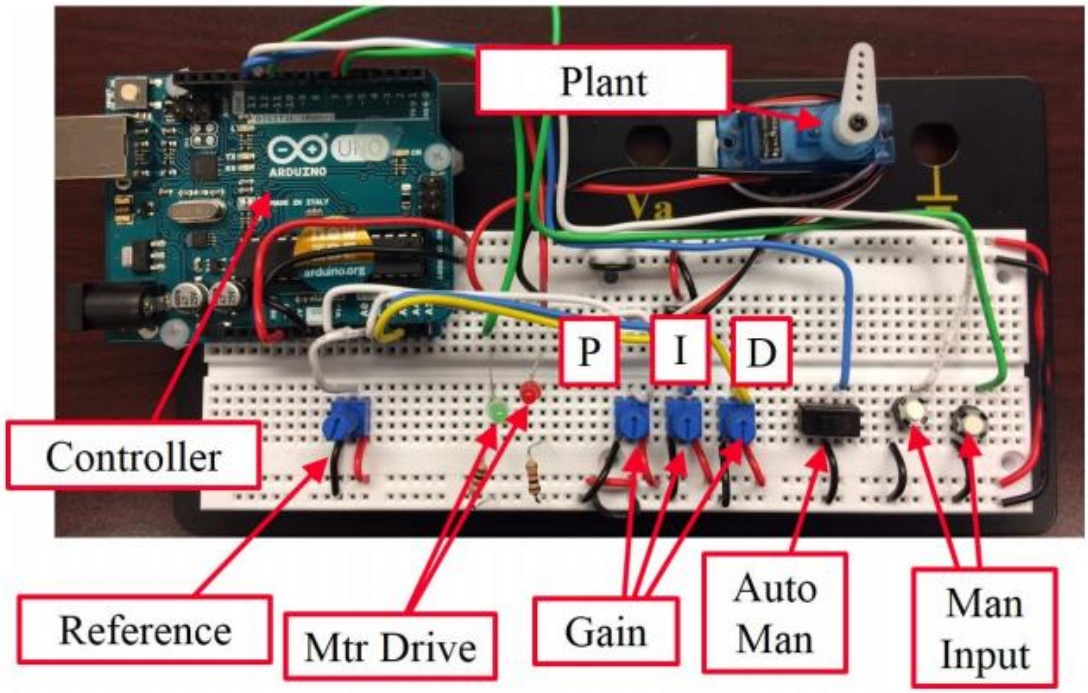
and here.



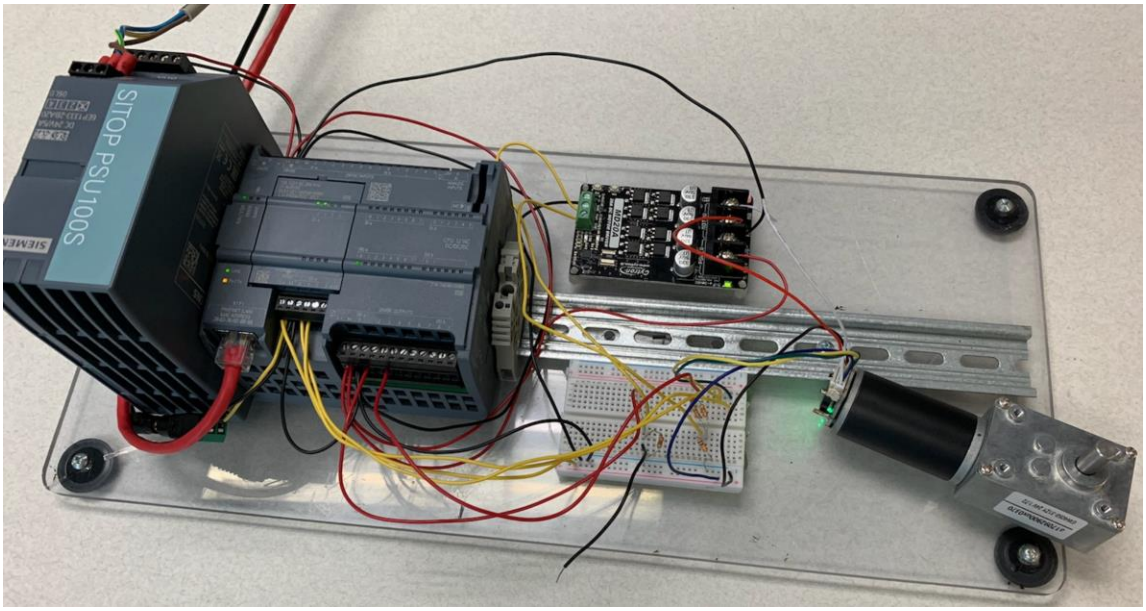
The equipment whose diagram is shown below was used but only by demonstration since the school owned only two of these units. This unit was from the company Feedback. Again, the DC motor was controlled by a PID loop for speed control and a second PID loop for position.



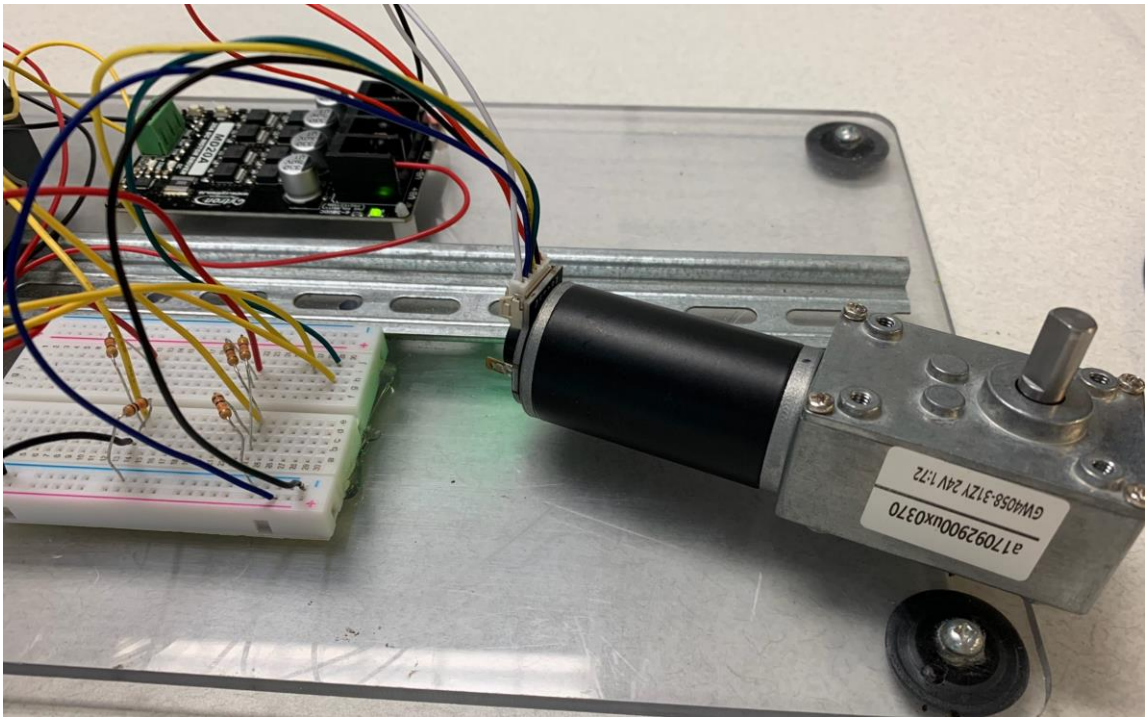
Another was from a paper "A Low-Cost Control System Experiment for Engineering Technology Students" by Dr. Curtis Cohenour Ph.D., P.E. P.E., Ohio University. Dr. Cohenour's implementation is shown here. This design was attempted with limited success.



At present, the PID block controlling a dc gear motor is used to control the DC motor. The control loop is to control the speed of the motor to a setpoint. The controller is a Siemens S7-1200 with encoder inputs and PWM output.



Connections are seen in these figures. The motor connects its encoder logic through the breadboard. The power to the motor is furnished through the controller at top. The power is provided by the 24 V supply at left and a 5 V supply which is not shown.



The motor can be seen here. It is rated at 24 V dc and has a gear reduction of 1:72.

Wiring from Motor and Encoder Connection:

- Red Wire - positive power supply of motor(+)
- White Wire - negative power supply of motor(-)
- Black Wire - negative power supply of encoder(-) (positive and negative power supply of encoder do not allow connect wrong; voltage is 3.3 -5V)
- Blue Wire - positive power supply of encoder(+) (positive and negative power supply of encoder do not allow connect wrong; voltage is 3.3 -5V)
- Yellow Wire - signal feedback (11 signals per motor turns a circle)
- Green Wire - signal feedback (11 signals per motor turns a circle)

The S7 1200 v2.2 CPU in relation to the Quadrature encoder:

The 1200 with 24V DC inputs support up to six High Speed Counters.

a. Up to 3 of the addresses Ia.0 to Ia.5 can be used for Quadrature Mode at 80Khz.
b. Up to 3 of the addresses Ia.6 to Ib.5 can be used for Quadrature Mode at 20Khz.

The DI4 5 VDC signal board with part number 6ES7 221-3AD30-0XB0 is used in this example since the encoder pulse signals are 5V. It supports 160 KHz for Quadrature Mode. The input is "source" type. So, the encoder must support this which means that it is NPN or Open Collector type. The encoder and card work together in this example.

For the S7 1200 v2.2 when configuring the HSC, please select the Input Source as "signal board input". Please note that the option is only available if the signal board has been added in the hardware configuration. Please note that the addresses of the HSC will be the address of signal board (default 4.x). In v4.x S7 1200's, the user selects the address rather than the Input Source.

The dc motor controller is shown below. We used either 12 V for power to the motor or 24 V depending on the rating of the DC motor. This controller is very inexpensive and can accommodate either the 12 V or 24 V varities. It is also used in the tank-over-tank lab in Hybrid Text Ch. 25. It works well with the cheap power supplies (MeanWell or equivalent) for most applications.

Cytron Technologies MD 20A
20 A 6V-30 V DC Motor Driver

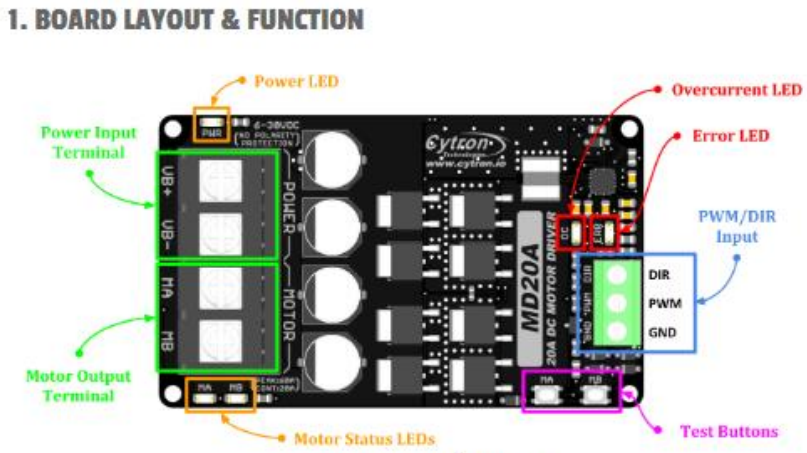
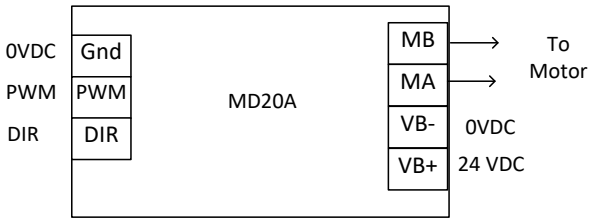
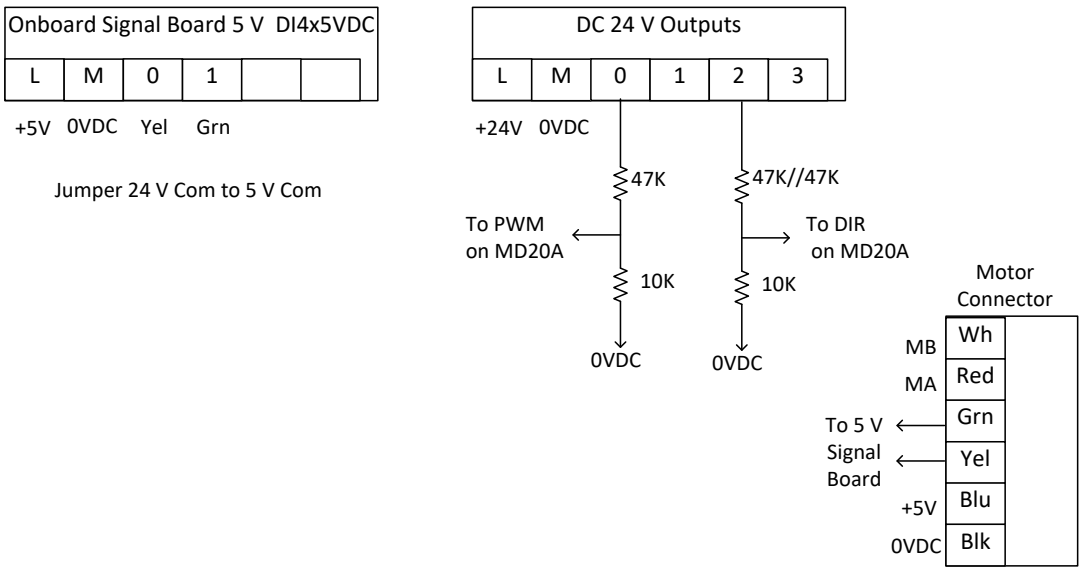


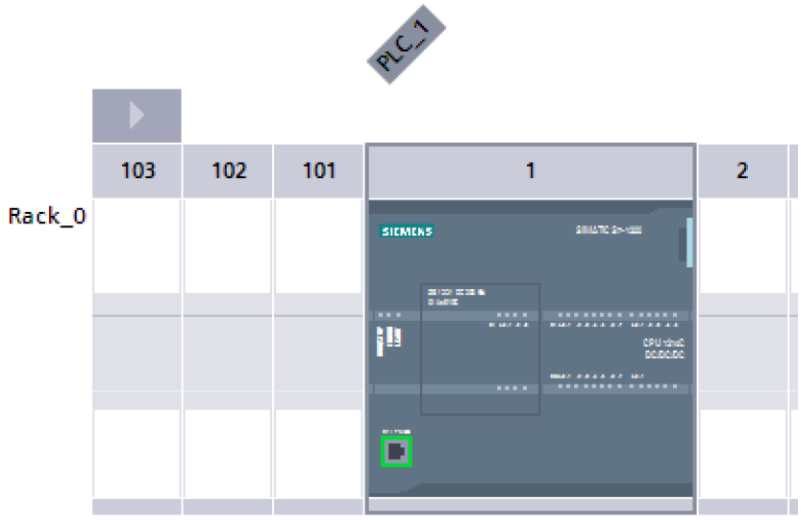
Figure 1: MD20A Board Functions

The wiring diagram shown below gives the complete wiring diagram for the motor shown in the figures above:

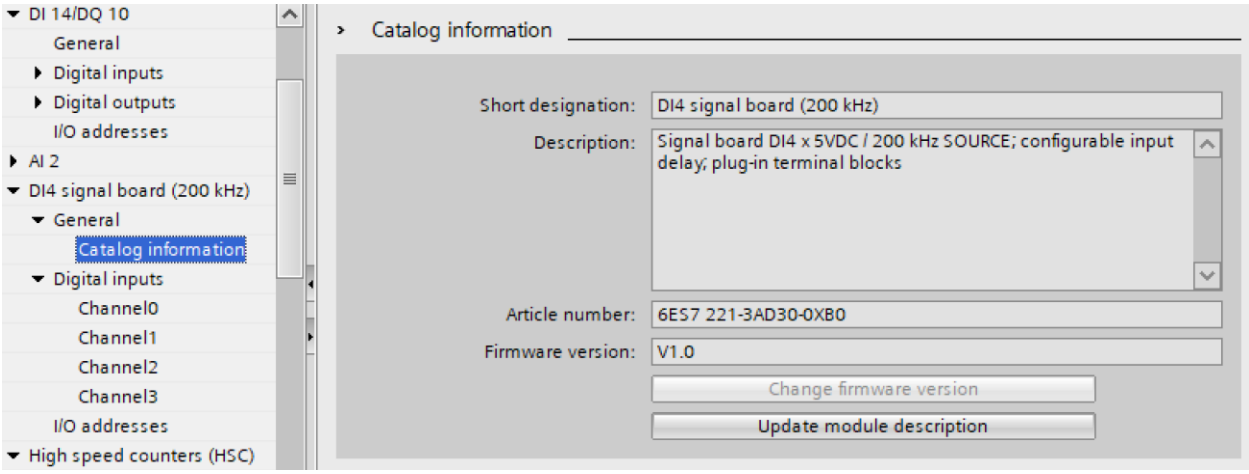




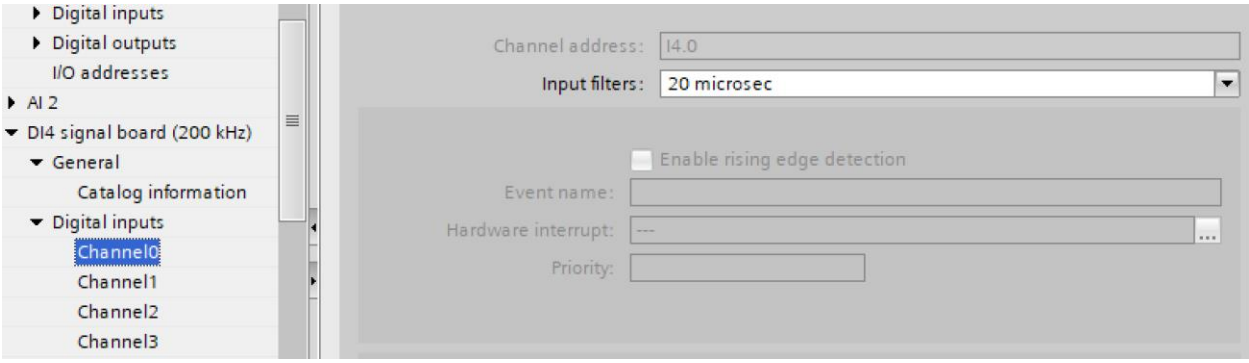
The diagram below shows the Siemens PLC:



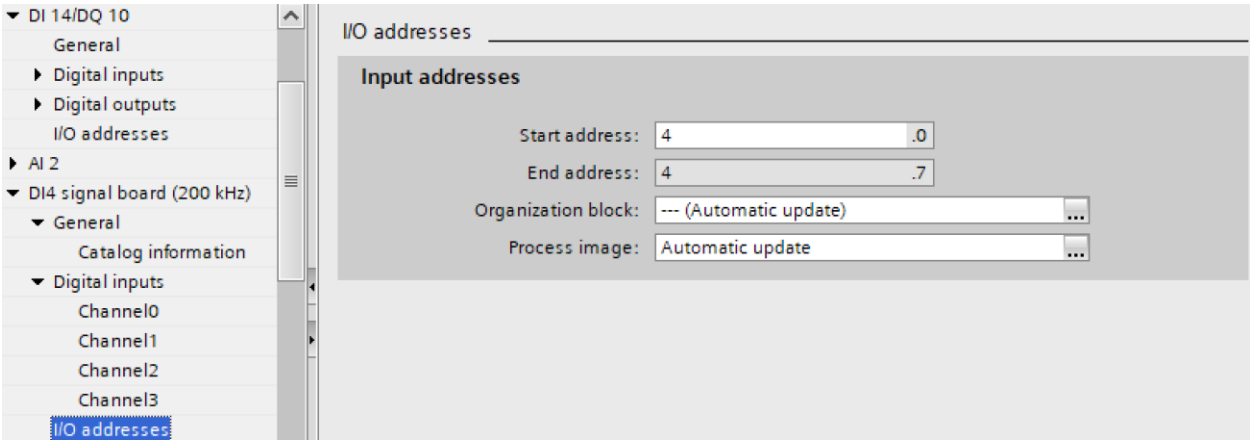
Configuration of the PLC is given in the following configuration tabs:

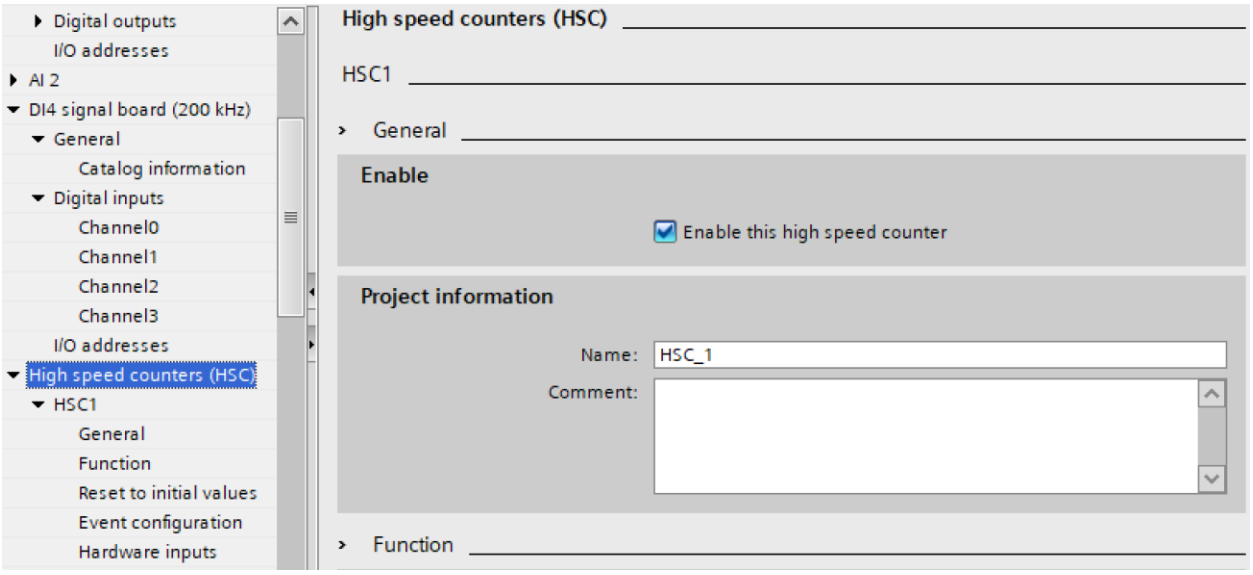


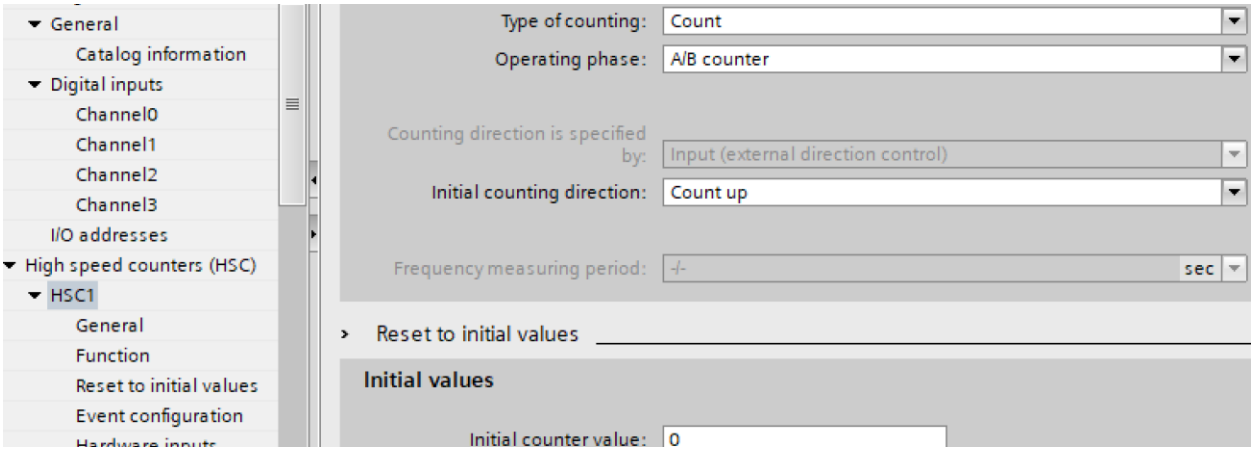
The Signal Board must be changed to allow for the fast pulse inputs:



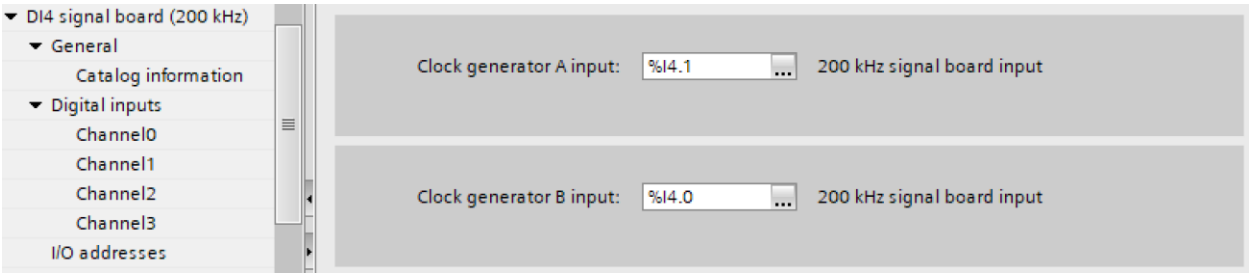
Addressing of the Digital Inputs for the encoder from the Signal Board are shown here:



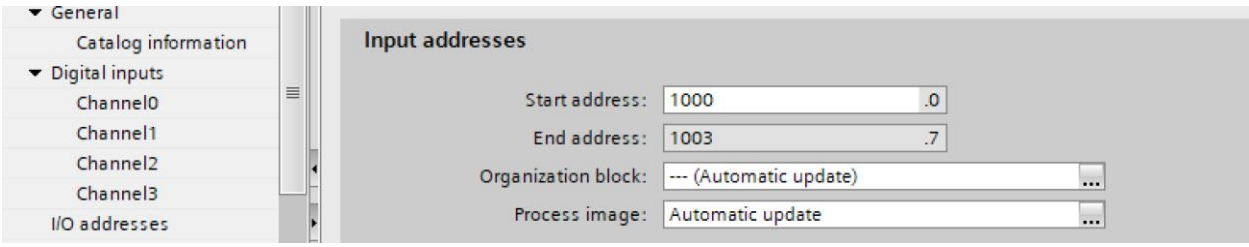




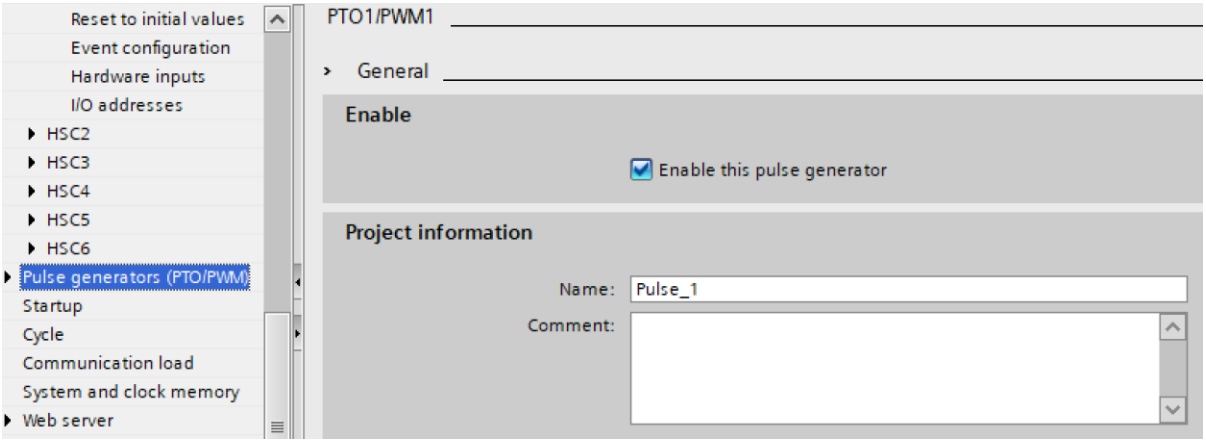
The address of the encoder is configured below:

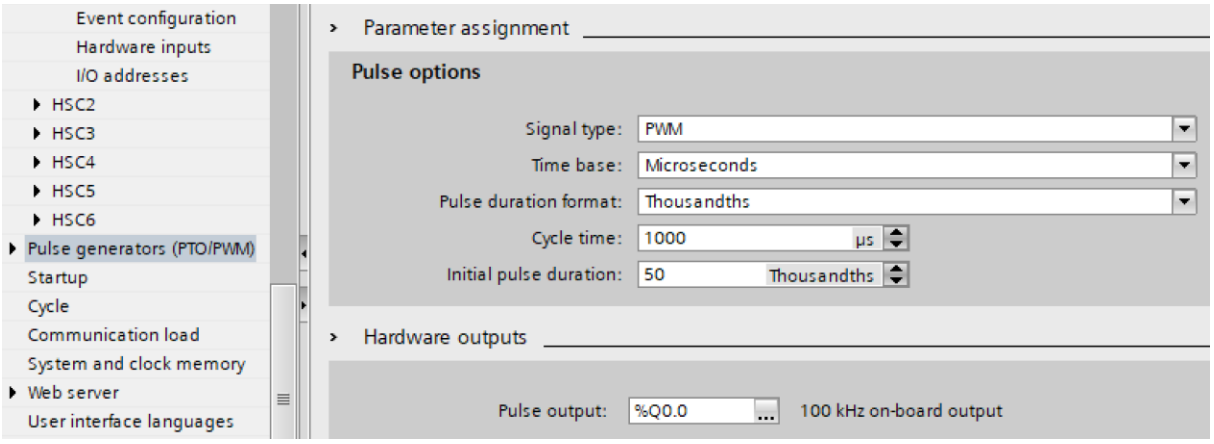


The high-speed input enters the PLC as a pulse in ID1000. This is the address used to calculate the pulses received in the last 100 msec.

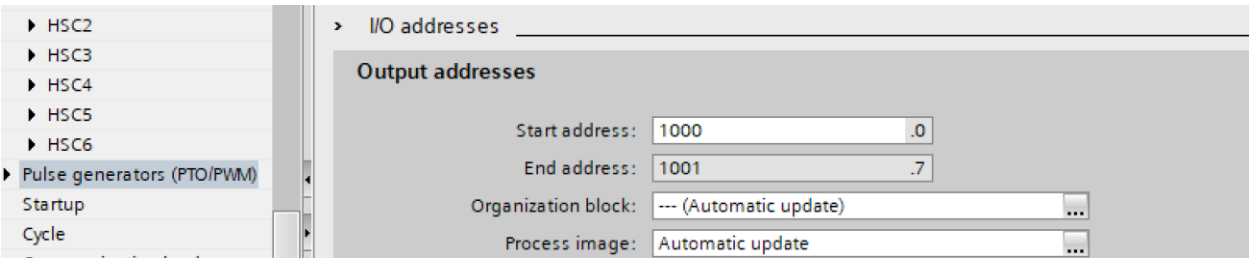


The output PWM is shown here:



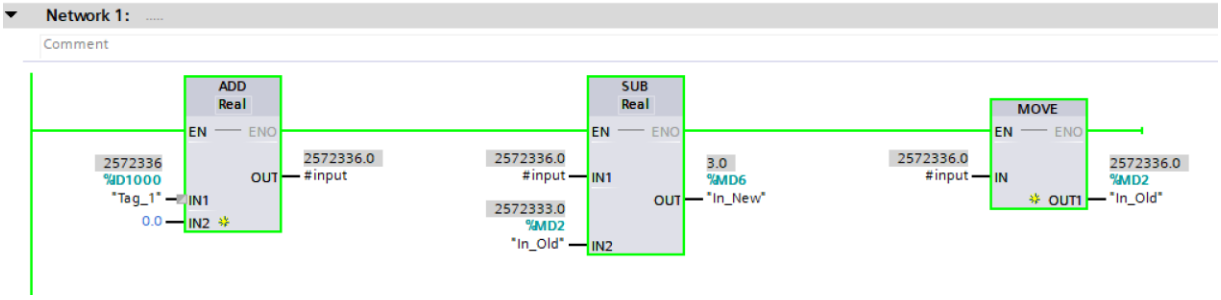


The output configured is a PWM signal that varies from 1 to 1000 microseconds and repeats each millisecond.

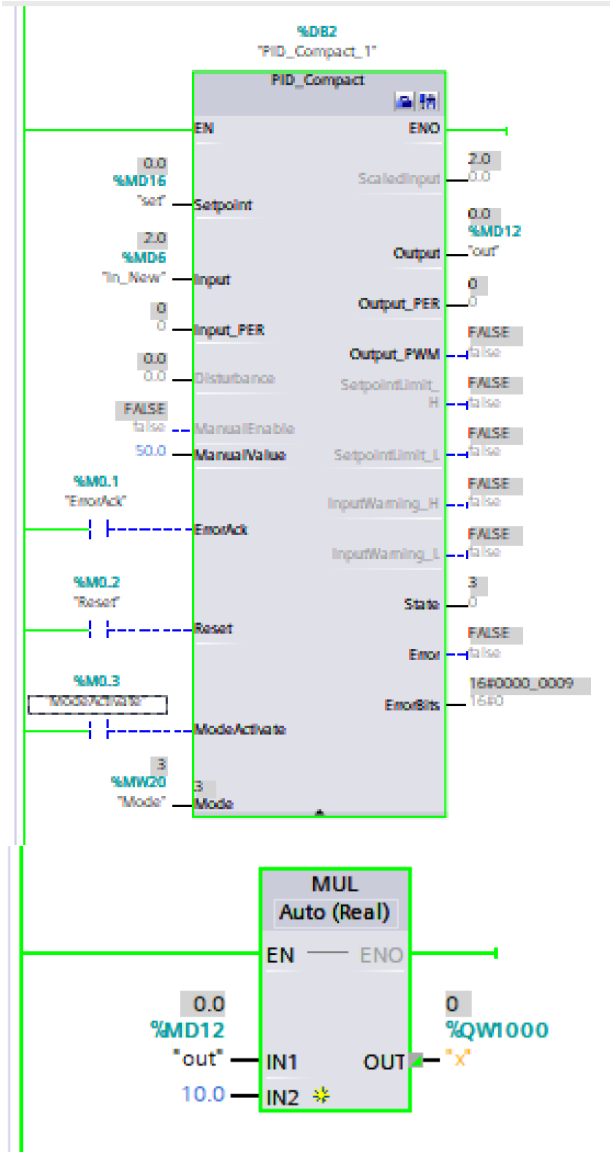


The program is in a cyclic interrupt program that is executed each 100 ms. It consists of a rung to get the input count for the last 100 msec followed by the PID block and finally the output to the PWM block.

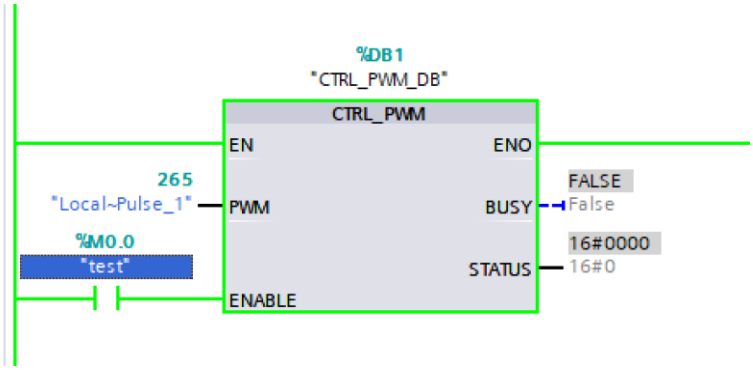
The choice of 100 ms is chosen due to the observation that with 11 pulses per revolution of the motor, we only receive about 125 pulses in 100 ms. If we were to choose 10 ms as our cycle time, we would only expect 12 pulses per scan at full speed. We need more accuracy than 1/12. If one were to spend more for a more sophisticated motor, more pulses would give greater accuracy and the scan could be reduced.



The PID block is shown next. The values for the numeric inputs are found in the Watch Table. Boolean variables can be set directly by modifying the input in online mode.



OB1: The OB1 block contains only the PWM input block. The 'test' bit is to be set before the PWM is allowed.



The following is a watch table built to display the variables for the PID block. The video gives additional detail of the PID block and uses this watch table to vary the speed.. Set the setpoint via the input 'set'.

Project10 ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Watch and force tables ▸ Watch table_1

	Name	Address	Display format	Monitor value	Modify value	Comment
1	"x"	%QW1000	DEC+/-	743	100	
2	"Tag_1"	%ID1000	DEC+/-	1189089		
3		%Q0.2	Bool	FALSE	FALSE	
4	"set"	%MD16	Floating-poin...	125.0	125.0	
5	"Mode"	%MW20	DEC+/-	3	3	
6		<Add new>				

The commission of the PID block allows the auto tuning of the PID variables. With this option, the following PID parameters are generated and the output below.

PID Parameters

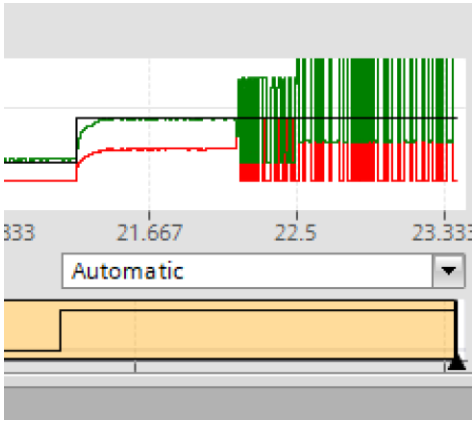
Enable manual entry

Proportional gain:	2.557061E-1	
Integral action time:	1.367988	s
Derivative action time:	3.059973E-1	s
Derivative delay coefficient:	0.1	
Proportional action weighting:	0.58	
Derivative action weighting:	0.0	
Sampling time of PID algorithm:	9.99991E-2	s

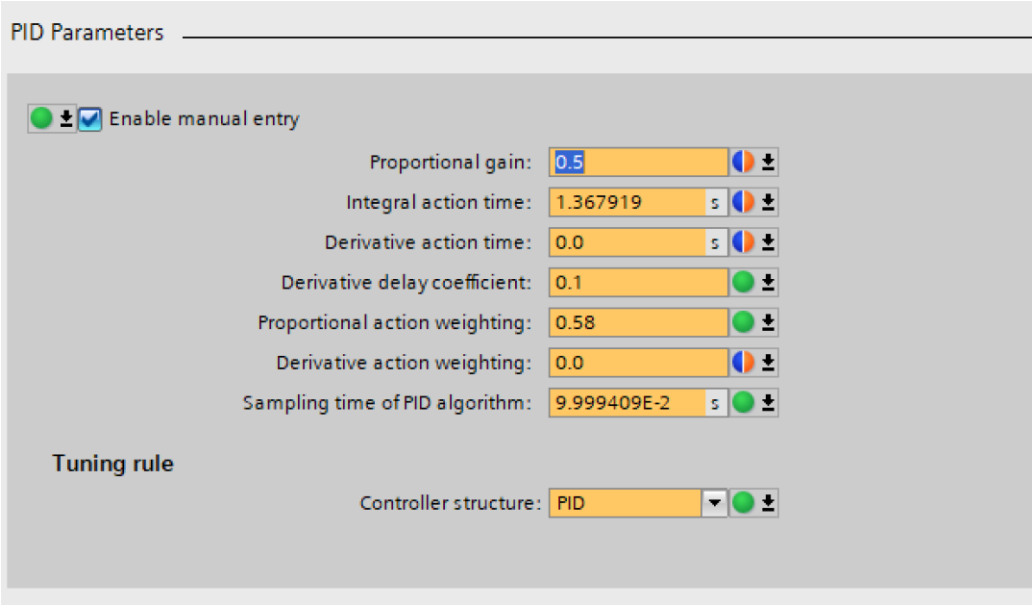
Tuning rule

Controller structure: PID

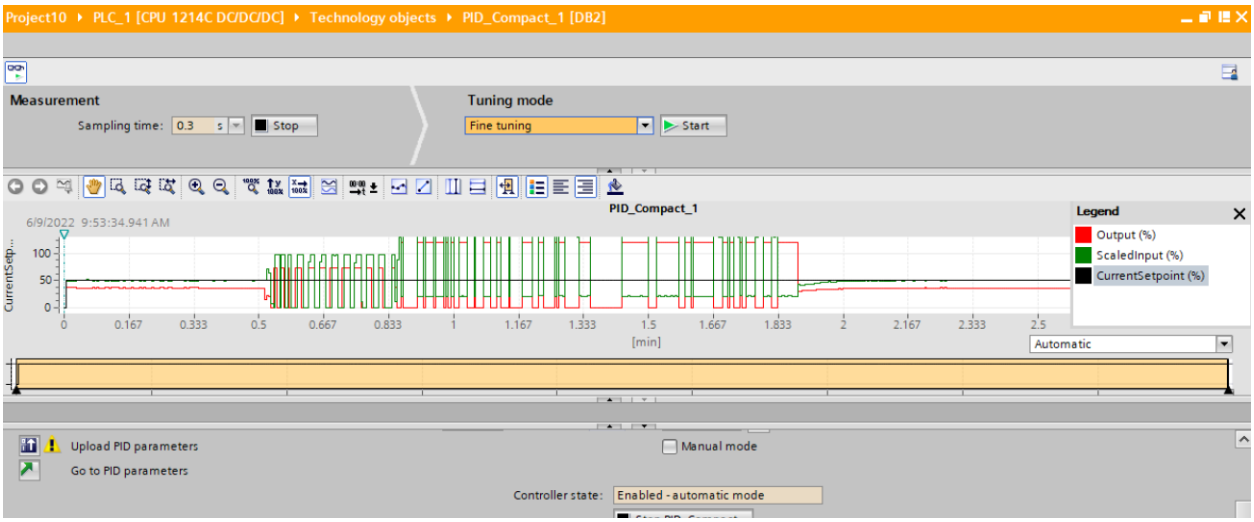
The motor oscillates. This demonstrates the instability of the tuning process. This is a good example of instability.



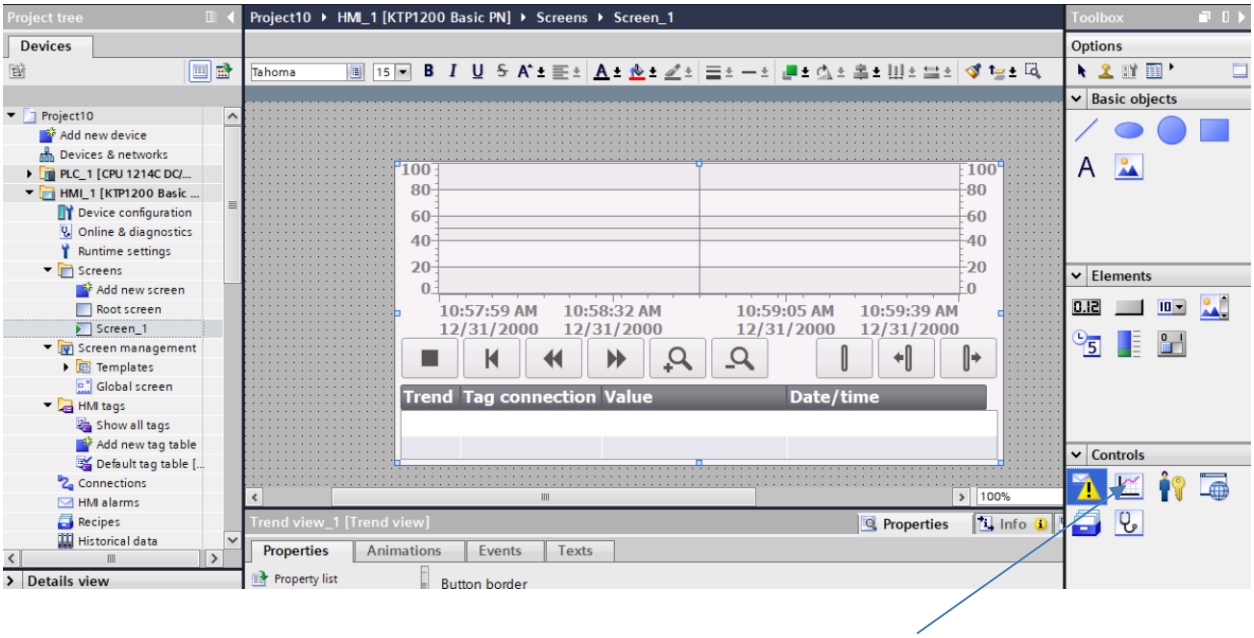
With the change shown below, instability is gone. Move a ‘0’ into the Derivative action time variable and stability.



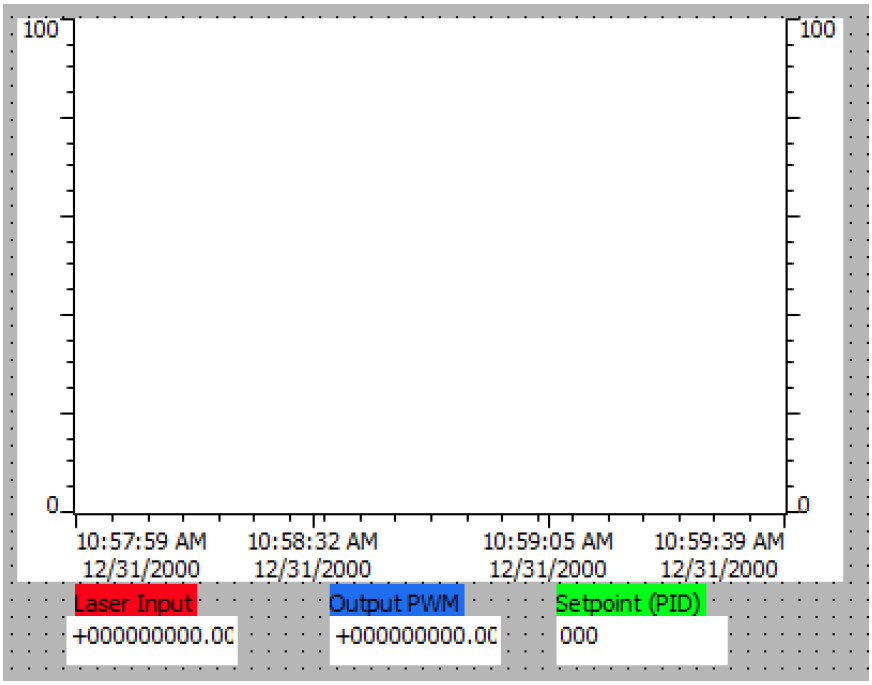
Now, the process of finding the best P, I, and D for a best response is required.



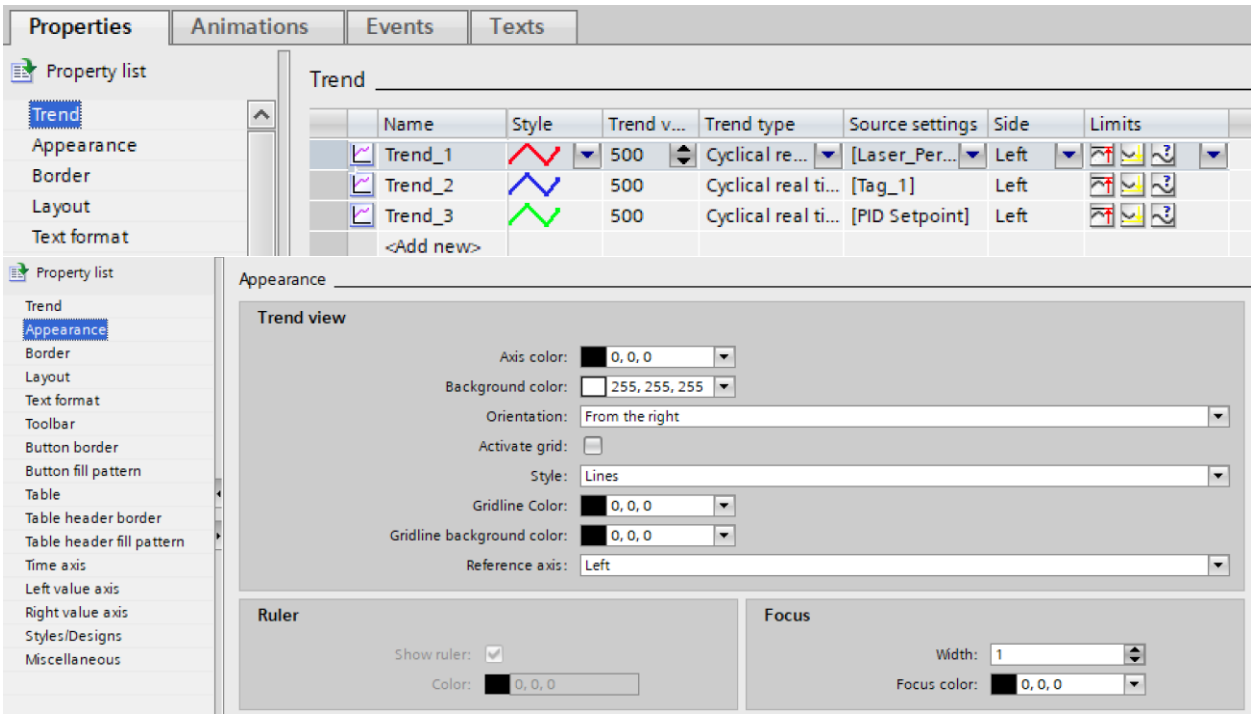
The best display of the response to the PID loop is not the chart above but rather the display below on a screen of an HMI display. Following is a step-by-step set of instructions to configure the display similar to the Ball-in-Tube and other PID labs. These instructions give a complete set of instructions for configuration of a historical data trend.

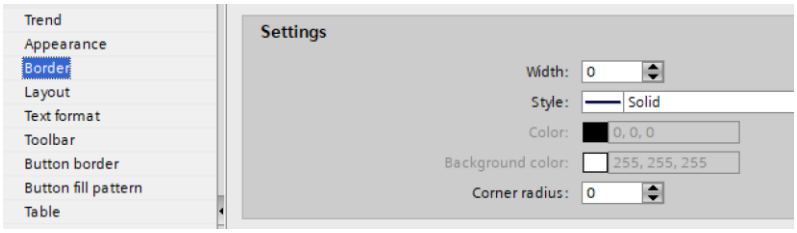


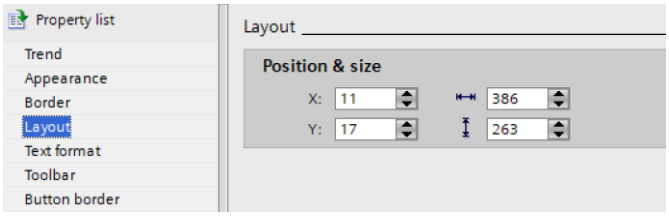
The Trend View under Controls is chosen. The picture above is very different than what is configured below:

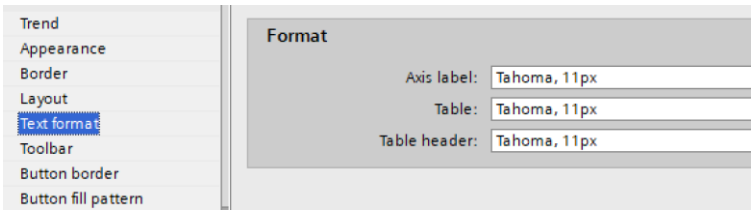


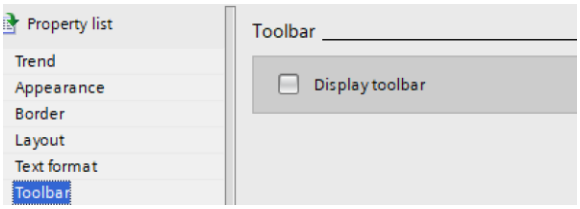
The following set of screens give the detail for set-up of the historical data trend:

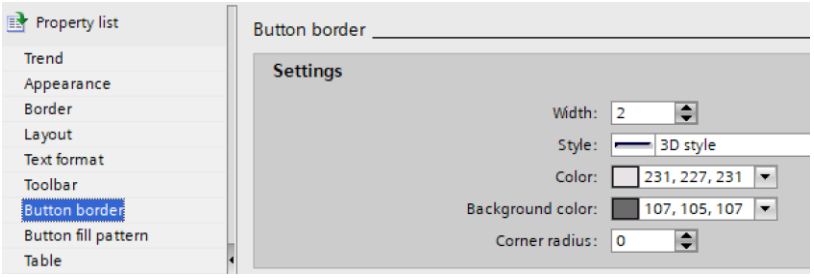

Properties
Animations
Events
Texts
Property list
Trend
Appearance
Border
Layout
Text format
Trend
Name
Style
Trend v...
Trend type
Source settings
Side
Limits
Trend_1
500
Cyclical re...
[Laser_Per...
Left
Trend_2
500
Cyclical real ti...
[Tag_1]
Left
Trend_3
500
Cyclical real ti...
[PID Setpoint]
Left
<Add new>
Property list
Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Table header border
Table header fill pattern
Time axis
Left value axis
Right value axis
Styles/Designs
Miscellaneous
Appearance
Trend view
Axis color:
0, 0, 0
Background color:
255, 255, 255
Orientation:
From the right
Activate grid:
Style:
Lines
Gridline Color:
0, 0, 0
Gridline background color:
0, 0, 0
Reference axis:
Left
Ruler
Show ruler:
Color:
0, 0, 0
Focus
Width:
1
Focus color:
0, 0, 0

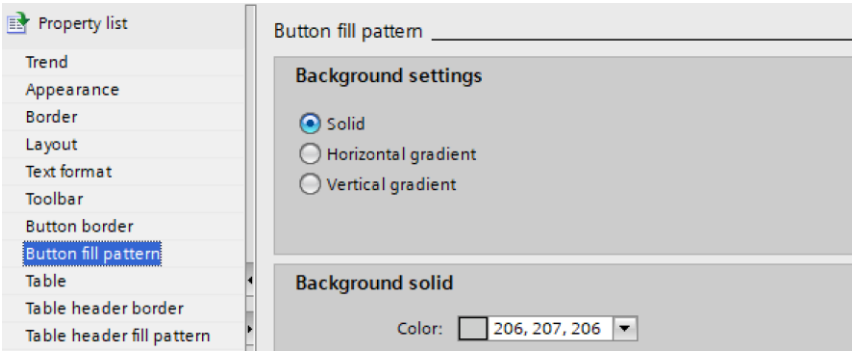

Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Settings
Width:
0
Style:
Solid
Color:
0, 0, 0
Background color:
255, 255, 255
Corner radius:
0

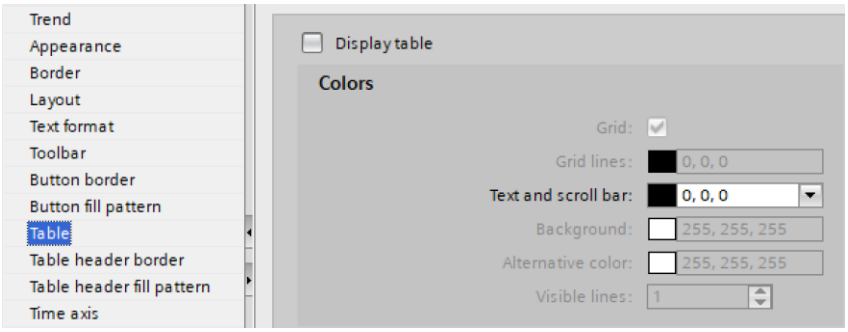

Property list
Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Layout
Position & size
X:
11
386
Y:
17
263

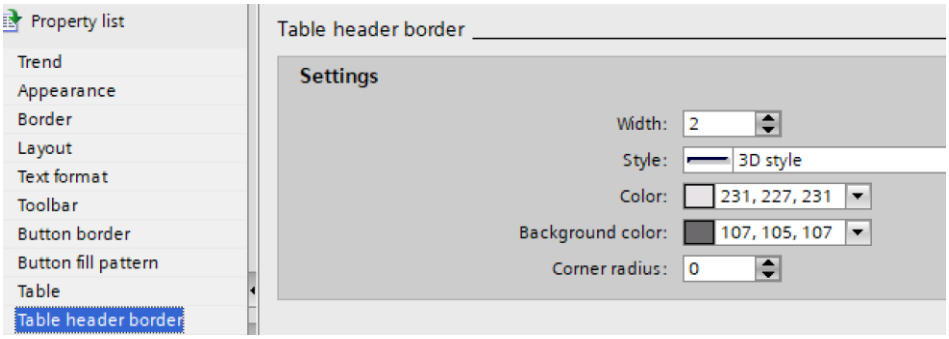

Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Format
Axis label:
Tahoma, 11px
Table:
Tahoma, 11px
Table header:
Tahoma, 11px

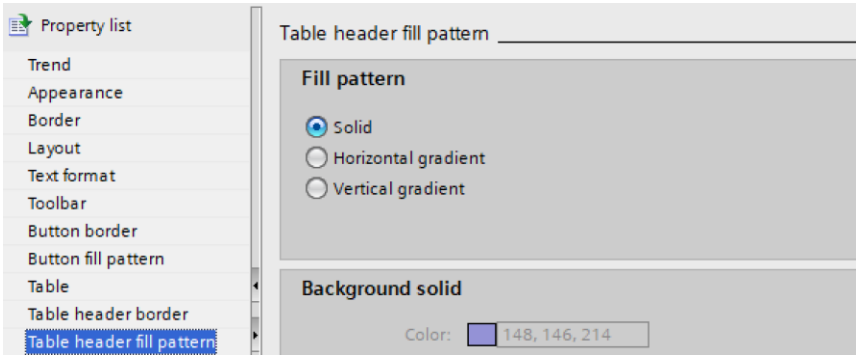

Property list
Trend
Appearance
Border
Layout
Text format
Toolbar
Toolbar
Display toolbar

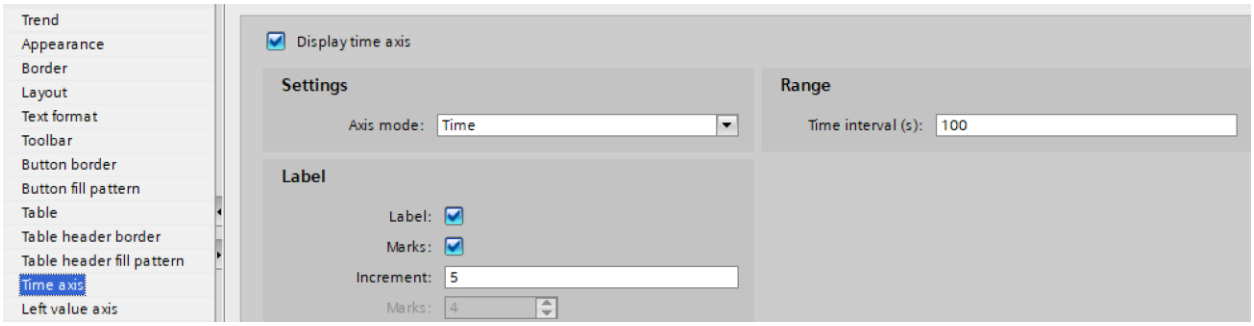

Property list
Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Button border
Settings
Width: 2
Style: 3D style
Color: 231, 227, 231
Background color: 107, 105, 107
Corner radius: 0

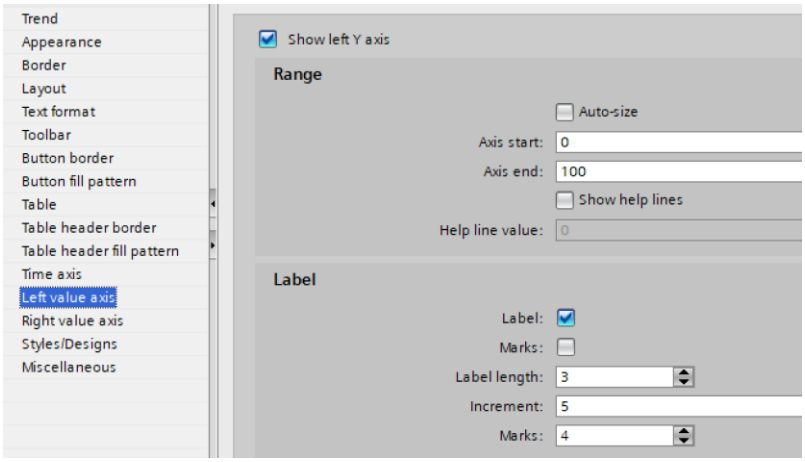

Property list
Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Table header border
Table header fill pattern
Button fill pattern
Background settings
Solid
Horizontal gradient
Vertical gradient
Background solid
Color: 206, 207, 206

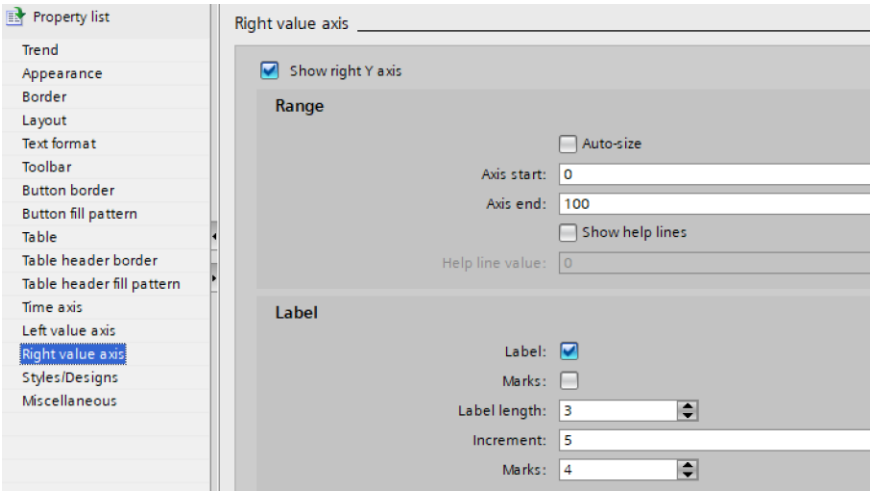

Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Table header border
Table header fill pattern
Time axis
Display table
Colors
Grid:
Grid lines: 0, 0, 0
Text and scroll bar: 0, 0, 0
Background: 255, 255, 255
Alternative color: 255, 255, 255
Visible lines: 1

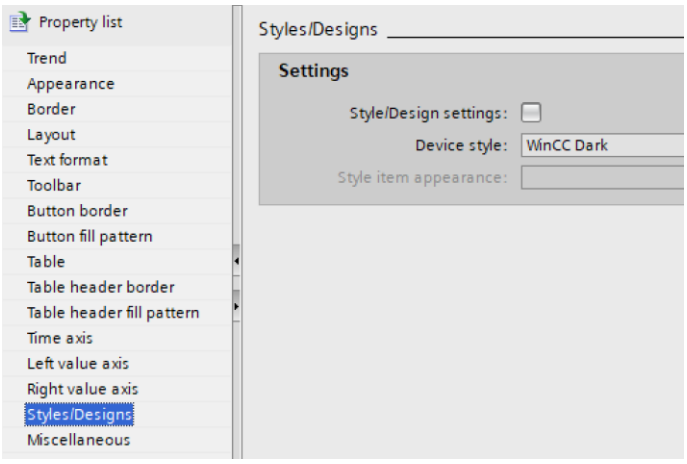

Property list
Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Table header border
Table header border
Settings
Width: 2
Style: 3D style
Color: 231, 227, 231
Background color: 107, 105, 107
Corner radius: 0

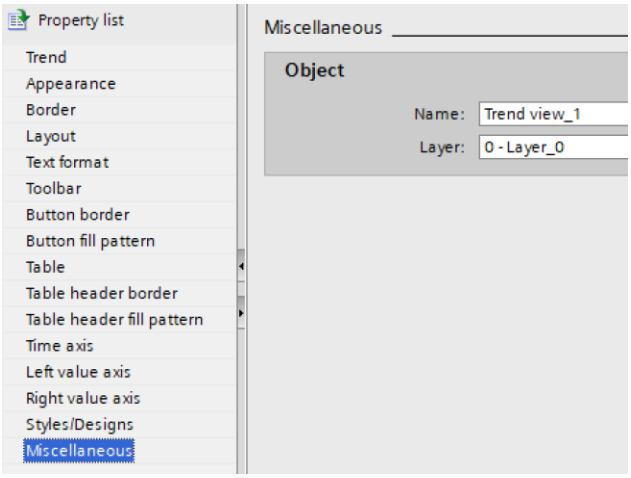

Property list
Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Table header border
Table header fill pattern
Table header fill pattern
Fill pattern
Solid
Horizontal gradient
Vertical gradient
Background solid
Color: 148, 146, 214


Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Table header border
Table header fill pattern
Time axis
Left value axis
Display time axis
Settings
Axis mode: Time
Range
Time interval (s): 100
Label
Label:
Marks:
Increment: 5
Marks: 4


Trend
Appearance
Border
Layout
Text format
Toolbar
Button border
Button fill pattern
Table
Table header border
Table header fill pattern
Time axis
Left value axis
Right value axis
Styles/Designs
Miscellaneous
Show left Y axis
Range
Auto-size
Axis start: 0
Axis end: 100
Show help lines
Help line value: 0
Label
Label:
Marks:
Label length: 3
Increment: 5
Marks: 4



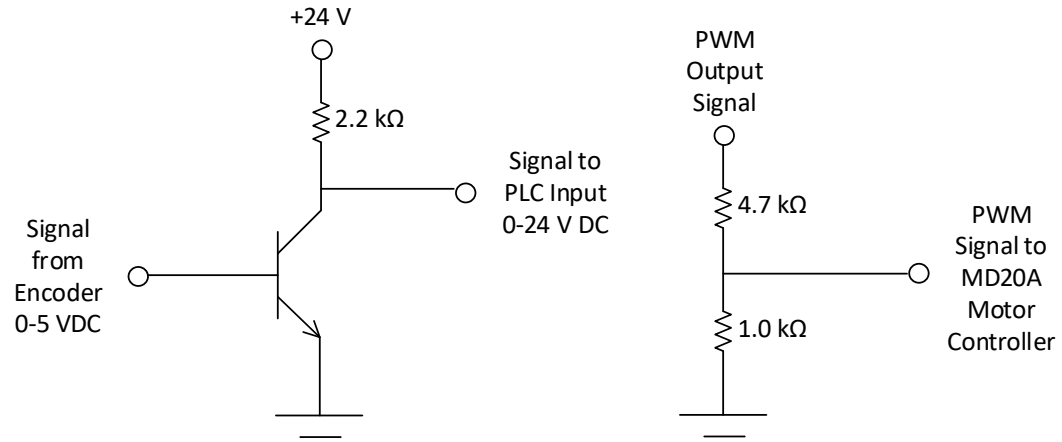




This lab would consist of building this motor and controller and replicate the results here. To continue, add a second PID block to control to a position. This was the original result of the lab from Prof. Rich. To continue, use the hardware to configure the motion application from the motion control labs in Ch. 7 of the Hybrid Text.

Appendix I

A second interface between the PLC and the motor and the motor controller is shown in the next diagrams. This diagram is an earlier wiring diagram that 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:

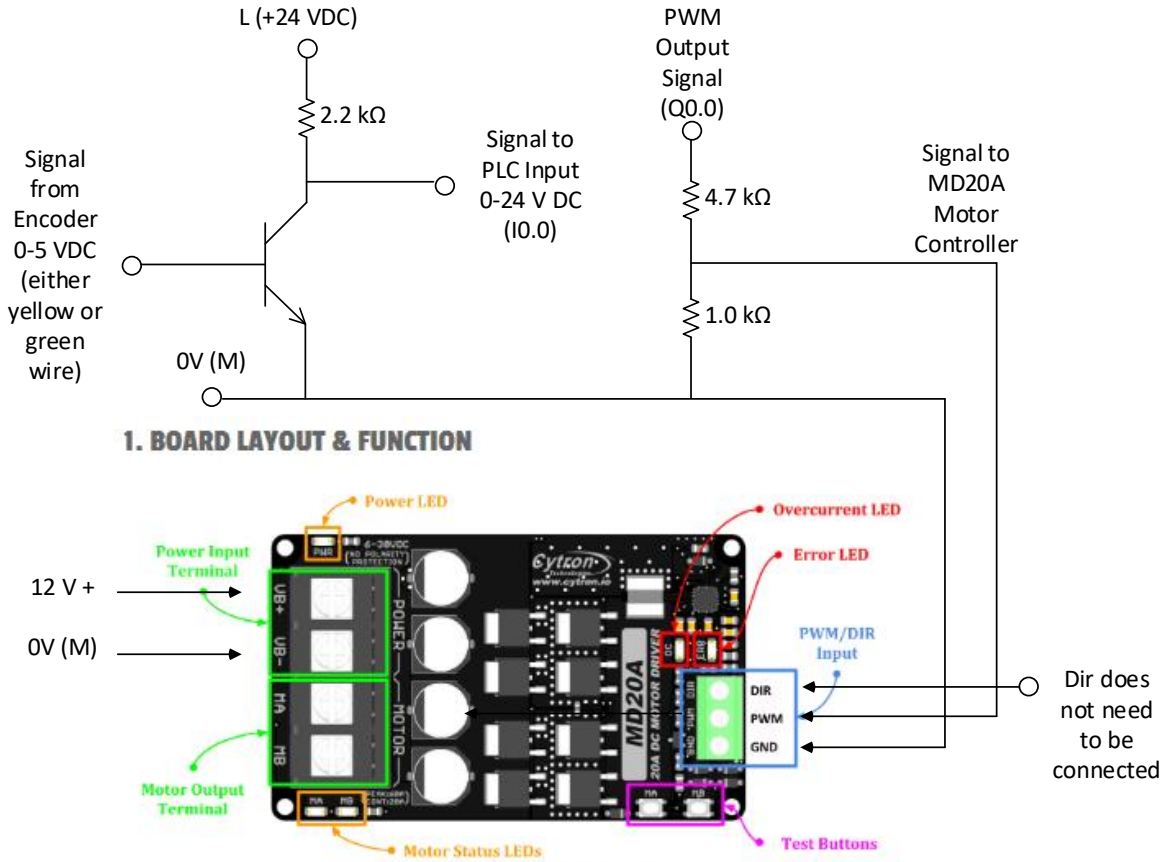


Figure 1: MD20A Board Functions

Appendix II

Parts for this lab include:

1. S7-1214 or 1215 DCDCDC Siemens PLC
2. DI4 5 VDC signal board
3. Cytron Technologies MD 20A - 20 A 6V-30 V DC Motor Driver
4. ALITOVE 5V 3A 15W AC 100V~240V to DC Power Supply Adapter Converter
5. Uxcell DC 24V 111RPM 16Kg.cm Self-Locking Worm Gear Motor with Encoder/Cable or equal

Appendix III

Two very good Simulink labs were created at U of Michigan with the DC motor. These can be done with the motor characteristics of the motor selected by you for the lab and the results compared between the actual working PID controller and the Simulink control output: