Moog-Animatics Class 4 SmartMotor Laboratory Project for SJSU ME 190 Course





**San Jose State University**

**Department of Mechanical Engineering**

**ME 190 Mechatronic System Design**

**Dr. Furman**

**Project Created By:**

Joshua Cobain

Ryan Reardon

**Project Sponsorship and Support By:**

Moog-Animatics

Table of Contents

[1 Executive Summary 3](#_Toc405843976)

[2 Introduction 4](#_Toc405843977)

[3 Background 4](#_Toc405843978)

[4 Lab One – Introduction to the SmartMotor 6](#_Toc405843979)

[4.1 Equipment Needed 6](#_Toc405843980)

[4.2 Objectives 6](#_Toc405843981)

[4.3 Method 6](#_Toc405843982)

[5 Lab Two – Advanced SmartMotor Programming and Controls 7](#_Toc405843983)

[5.1 Equipment Needed 9](#_Toc405843984)

[5.2 Objectives 9](#_Toc405843985)

[5.3 Method 9](#_Toc405843986)

[6 Discussion and Future Direction 9](#_Toc405843987)

[6.1 What Was Learned Through Creating These Lab Exercises 9](#_Toc405843988)

[6.2 Proposed Future Direction 9](#_Toc405843989)

[7 Conclusions 10](#_Toc405843990)

[8 References 11](#_Toc405843991)

[9 Appendix A: Project Bill of Materials 12](#_Toc405843992)

[10 Appendix B: Moog-Animatics SM2316D Motor Specifications 13](#_Toc405843993)

[11 Appendix B (Con’t.) 14](#_Toc405843994)

[12 Appendix C: Example PID Controller Program for SmartMotor 15](#_Toc405843995)

[13 Appendix D: Lab One – Introduction to the SmartMotor Lab Manual 17](#_Toc405843996)

[14 Appendix E:Lab Two – Advanced SmartMotor Programming and Controls Lab Manual 32](#_Toc405843997)

[15 Team Evaluation – Ryan Reardon 37](#_Toc405843998)

[16 Team Evaluation – Josh Cobain 38](#_Toc405843999)

# Executive Summary

This laboratory project for ME190 is aimed at exposing engineering students to Moog-Animatics Class 4 SmartMotors and educating them on the benefits SmartMotors provide for motion control applications. The first homework assigned to the student is to download the drivers and the SmartMotor Interface (SMI). The first lab is to get the students familiar with the SMI software by utilizing commands from SmartMotor Playground, BCD controller, and simple IO programming in junction with the SmartMotor. The second part of the SmartMotors lab contains PID creation and tuning using a small mass with a pulley system. Using a PID controller in the SmartMotor with the pulley system will allow students to get a hands approach to physically monitor the control feedback loop rather than theoretically derived.

This project is funded and supported by Moog-Animatics. They will be donating the required cabling and accessories (See Appendix A) to compliment the SmartMotors that they have previously donated to San Jose State University’s mechanical engineering department.

# Introduction

SmartMotors have many advantages over other motors. The advantages include ease of use, sizing/packaging, and advanced motion control that will be experienced in the second part of this lab. SmartMotors are highly programmable in addition to advanced motion control, also the SmartMotors have an integrated servo motor system that is integrated with a motor, an encoder, an amplifier, a controller, RS232communication, and IOs. Another way SmartMotors are more advance is due to the added option of adding additional axes quickly and without hassle, minimized cabling and fewer components are a few reasons for SmartMotors continued use in packaging applications.

This is a laboratory is for future ME190 students to get familiar with Moog-Animatics Class 4 SmartMotors. This lab is designed to be a broken up into two parts. The first part of the lab is primarily basic programming and commands, and the second part is more advanced programming using PID tuning and control systems.

# Background

The use of Moog-Animatics SmartMotors for high precision motion control can provide engineers with an advanced solution that is easy to implement. Generally, engineers turn to using conventional motion control systems. These systems are often separate components consisting of separate motion controllers, drive amplifiers, motors, and feedback devices. This causes multiple problems for an engineering as using separate components means higher system costs, higher rate of failure, slower time-to-market, and increased difficulty to troubleshoot or diagnose. This is due to the highly increased complexity of the system by using multiple components.

The Moog-Animatics SmartMotor eliminates these complexities and difficulties by providing an all-in-one motion control solution. A Moog-Animatics SmartMotor incorporates the motion controller, drive amplifier, motor and feedback devices into a single package, creating an integrated servo system. This integrated servo system reduces costs as separate components do not need to be purchased. Because separate components do not need to be sized and selected, the overall system’s time-to-market is greatly decreased as motion control can be controlled by the SmartMotor itself. The use of SmartMotors increases the system’s simplicity, increasing the reliability of the system and decreasing diagnostic and field repair costs. This lab works to provide engineering students with an in-depth introduction and general product knowledge of SmartMotors. By exposing students to SmartMotors, they can be learn about a commercially available motion control solution. This will give them motion control knowledge that can be used in their professional careers and can lead to them choosing Moog-Animatics products for future applications.

The objective of this project was to create a two part laboratory for San Jose State University’s ME190 Mechatronics Design course using the Moog-Animatics Class 4 SmartMotors. The pre-lab to the first lab has the students download the SMI software from Animatics website along with the drivers for the RS232-to-USB. Once these are downloaded, the students may begin the first lab. The first lab has a total of 12 exercises ranging from opening the SmartMotor Playground to programming the SmartMotor to go to 4 different positions with the BCD controller. The first few exercises are getting familiar with the SMI software and troubleshooting some problems associated with drivers and connecting to the SmartMotor. The first lab is a primarily basic commands and getting use to the SMI software.

The second part of the lab has advanced programming to give the students hands-on experience with PID creation and tuning. PID stands for proportional-integral-derivative, PID is a closed-loop feedback controller that is widely used in industrial control systems. Implementing the PID controller in the second part of the SmartMotors lab will challenge the students to apply their skills. This advanced lab gives the students a hands-on approach allowing changes to the PID to be physically monitored rather than theoretically derived.



Overall this lab is designed to get ME190 students familiar with Moog-Animatics Class 4 SmartMotors and programming. As the students go through this two part lab to discover what SmartMotors are and the advantages of using SmartMotors over conventional motors.

# Lab One – Introduction to the SmartMotor

The first lab is an introduction to SmartMotors and the goal is to get comfortable using the SMI software for the following advanced lab. The introduction to the Moog-Animatics Class 4 SmartMotor and getting familiar with the SmartMotor Interface SMI. The first few exercises involve finding the motor in the SMI software and troubleshooting problems associated with drivers and connection. The other exercises include getting familiar with simple commands to using the BCD controller with the SmartMotor to go between positions using the toggle switches.

## Equipment Needed

The equipment needed to complete the first part of this lab is the following:

* SmartMotor Interface (SMI) software
* RS232-to-USB drivers
* Moog-Animatics Class 4 Integrated Servo System
* CBLSM-DEMO1 Communication / Power Cable
* 24VDC Power Supply
* RS232-to-USB Adapter (RS232485TUSB)

## Objectives

After completing this lab, students should be able to complete and understand the following concepts:

* Basic SmartMotor programming
* Troubleshooting drivers and SMI
* Getting familiar with SMI software

## Method

The method behind the introduction lab is to build the students’ confidence using the SMI software and basic SmartMotor programming to prepare for the advanced lab. The introduction lab contains downloading and troubleshooting the SMI software and drivers for the RS232-to-USB. The introduction lab also includes position analysis and programming the SmartMotor to go to 2 different positions using the BCD controller. The Introduction lab prepares the students for the advanced lab and shows the advantages that SmartMotors have compared to other motor systems.

# Lab Two – Advanced SmartMotor Programming and Controls

The Advanced SmartMotor Programming and Control lab assignment will expose students to the advance programming functions of SmartMotors as well as give them experience with tuning a PID controller. Both tuning of the SmartMotor’s PID controller as well as the creation and tuning of a scripted PID controller created by the student will be covered in this lab. Students will use a mass attached to a pulley on the motor to simulate a load (See Figure 1). Students will then need to tune PID controller and observe the effects the values they chose have on the response of the system. Then, students will create their own PID controller in a SmartMotor program, and tune the controller to give the best response.

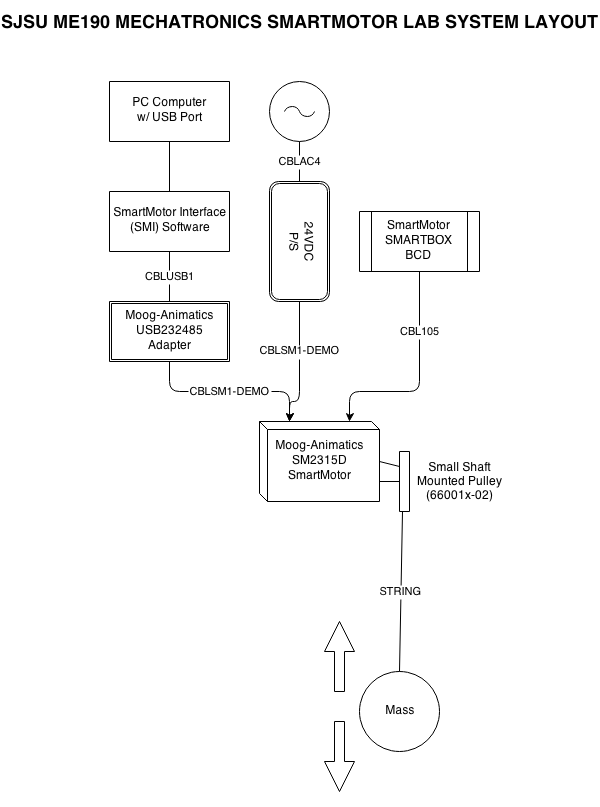


Figure 1: Layout of system for SmartMotor Lab Two

## Equipment Needed

The equipment needed to complete this lab is the following:

* Moog-Animatics Class 4 Integrated Servo System
* CBLSM-DEMO1 Communication / Power Cable
* 24VDC Power Supply
* RS232-to-USB Adapter (RS232485TUSB)
* Large and Small Pulleys
* A mass (< 1 kg)
* SmartMotor Interface (SMI) Software

## Objectives

After completing this lab, students should be able to complete and understand the following concepts:

1. Advanced SmartMotor programming techniques
2. Programming of a scripted PID controller
3. PID Tuning
4. Effects of PID components on system response

## Method

The reasoning behind this advanced lab exercise is to give students with a more in-depth experience with the SmartMotor. Due to the class’s concentration on system controls and the usefulness of the PID controller design, students are challenged in this lab to apply their skills. These skills will include SmartMotor programming abilities and product knowledge gained in the introductory lab as well as theories and principles taught throughout the semester-long class.

# Discussion and Future Direction

## What Was Learned Through Creating These Lab Exercises

While creating this two part lab exposed Josh Cobain to SmartMotors for the first time. Josh got a lot of insight why SmartMotors are used in certain application compared to other motor systems. SmartMotors have high precision motion controllers that can provide advanced solutions that are easy to implement. SmartMotors provide an all-in-one solution, A Moog-Animatics SmartMotor incorporates the motion controller, drive amplifier, motor and feedback devices into a single package, creating an integrated servo system.

## Proposed Future Direction

The lab exercises created through this project can be expanded and improved to better serve the class and fit closer to class lesson plans. These labs are to be supported in the long term by Moog-Animatics. Some possible directions classwork with SmartMotors can take include, but are not limited to the following:

1. Coordinated Motion between 2+ SmartMotors
2. Communication between a third-party MCU or PSoC using serial communication protocols
3. SmartMotor Sizing based on application requirements and desired motion profiles

Future additions to these lab exercises will increase student knowledge of the SmartMotor product line. It will also show the advantages of the all-in-one form factor that the SmartMotor exploits.

# Conclusions

This lab is for future ME190 students to get hands on experience with Moog-Animatics Class 4 SmartMotor that are used in industrial applications such as packaging, manufacturing, and robotics. Using PID controllers exposes the students to closed loop feedback controllers that is widely used in industrial control systems. The first lab introduces the students to the SMI software and basic programming. The second lab contains advance programming with PID creation and tuning with a system of pulleys. This lab proposal covers basic to advance programming and PID feedback loop control.

# References

Moog-Animatics 2008 Institute Training Manual

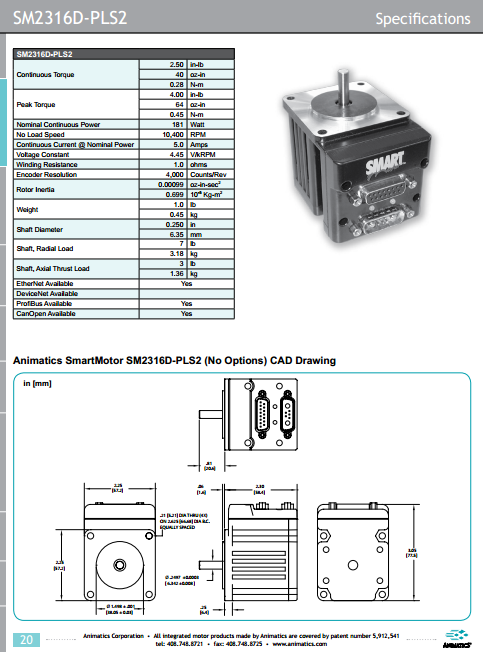
Project documentation video and demonstration video:

<https://www.youtube.com/watch?v=o5zpyyb6gjM>

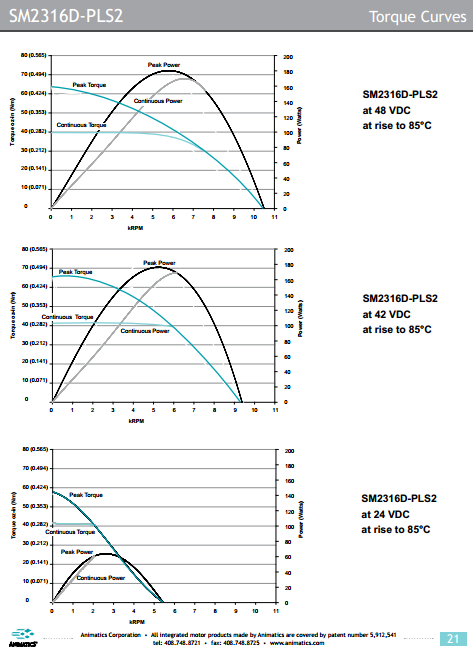
# Appendix A: Project Bill of Materials

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Created By:** | Reardon, Ryan |  |  |  |  | |  |  |  |
| **Date:** | 10/3/2014 |  |  |  |  | |  |  |  |
| **REV:** | 1 |  |  |  |  | |  |  |  |
| **SJSU ME190 Mechatronics Lab Station BOM** | | | | | | | | | |
| **Item Number** | **Part Number** | **Description** | | | | **Cost** | | **Qty.** | **Ext.** |
| 1 | PWR116V | 24VDC, 2.5A Power Supply | | | | $ 18.93 | | 1 | $ 18.93 |
| 2 | CBLAC1 | 6FT AC POWER CORD | | | | $ 3.68 | | 1 | $ 3.68 |
| 3 | CBLSM1-DEMO | TRAINING RS232 COMM/PWR CABLE | | | | $ 89.88 | | 1 | $ 89.88 |
| 4 | SMARTBOX BCD | 4-SWITCH TESTBOX | | | | $ 100.89 | | 1 | $ 100.89 |
| 6 | 660015-02 | VLACT 40T PULLEY - PLASTIC | | | | $ 5.52 | | 1 | $ 5.52 |
| 7 | 660016-02 | VLACT 72T PULLEY - PLASTIC | | | | $ 4.23 | | 1 | $ 4.23 |
| 8 | KITUSB232-485 | USB-TO-SERIAL CONVERTER | | | | $ 55.67 | | 1 | $ 55.67 |
| System Case | | | | | | | | | |
| 5 | PCASE4 | PELICAN DEMO CASE | | | | $ 95.21 | | 5 | $ 476.05 |
|  |  |  |  |  |  | Total/Station | | | $ 278.80 |
|  |  |  |  |  |  | Number of Stations | | | 15 |
|  |  |  |  |  |  | Total Cost | | | $ 4,657.99 |

# Appendix B: Moog-Animatics SM2316D Motor Specifications



# Appendix B (Con’t.)



# Appendix C: Example PID Controller Program for SmartMotor

'==========================================================================

'--- ME 190 MOOG-ANIMATICS DAY TWO ADVANCED LAB

'--- PID POSITION CONTROL DEMO PROGRAM

'--- CREATED BY: REARDON, RYAN (RR) ---

'--- DATE CREATED: 11/17/2014 ---

'--- PROG. REV #: 0

'|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

'| |

'--- PROGRAM DESCRIPTION ---

' This program is an example for the day two advanced SmartMotor lab.

' This program is designed to show how to programatically created a

' a PID controller and implement it to a Class 4 Moog-Animatics Smart-

' Motor.

'--------------------------------------------------------------------------

'--- REVISION HISTORY

' REV 0: RR, 11/17/14; Initial Program and template created

'--- ---

'--------------------------------------------------------------------------

'--- MACROS DECLARATION ---

#define MAXTORQ 1023 ' Maximum value for T (This will be the PID limit)

#define MAXPOS 15000 ' Maximum shaft position for end of travel

' MAXPOS will be the Position Setpoint

#define SCALE 1000 ' Used to scale PID values

#define PROP 5 ' Proportional (Kp) Gain

#define INT 0 ' Integrator (Ki) Gain

#define DERV 0 ' Derivative (Kd) Gain

#define DT 50 ' d/dt time difference (mS)

'--------------------------------------------------------------------------

'--- VARIABLE DECLARATION ---

d=0 ' Product of derivative

e=0 ' Value for currrent error (Global)

f=0 ' Value for previous error (Global)

g=0 ' Scratch variable for integrator

i=0 ' Product of integrator

o=0 ' Value of PID Output (For T value)

p=0 ' Product of proportional gain

z=0 ' Value for continue command

'--------------------------------------------------------------------------

'--- MOTOR SETUP ---

UAI ' Set all GPIO as inputs

UBI ' By doing this, hardward travel limits are disabled

UCI

UDI

UEI

UFI

UGI

ZS ' Clear Error Flags

MT ' Set Motor in Torque mode (MT)

T=0 ' T (Torque) value set to zero

OFF ' Turn drive off

AMPS=1023 ' Allow maximum current

WAIT=1000 ' Allow time (1 Sec) for mass to settle

O=0 ' Zero motor position

'--- MAIN ROUTINE ---

C0

PRINT("Moog-Animatics ME190 Advanced Lab",#13)

PRINT("PID-controlled Position Exercise",#13)

PRINT("Type z=55 and enter when ready",#32)

PRINT("To execute program!",#13)

WHILE z!=55

WAIT=100 ' Wait until user enters z=55 to confirm ready

LOOP

WHILE 1

GOSUB4

GOSUB1

GOSUB2

GOSUB3

GOSUB5

GOSUB10

f=e

LOOP

END

C1 ' Proportional Calculator

p=PROP\*e

p=p/SCALE

RETURN

C2 ' Integrator Calculator

g=e\*DT

i=i+g

i=i/SCALE

RETURN

C3 ' Derivative Calculator

d=e-f

d=d\*DERV

d=d/DT

d=d/SCALE

RETURN

C4 ' Error Calculator

e=MAXPOS-@P

RETURN

C5 ' Movement Function

o=p+i

o=o+d

T=o

G

RETURN

C10 ' Data Print Function

PRINT(CLK,#32,@P,#32,e,#32,f,#32,p,#32,i,#32,d,#32,o,#13)

RETURN

# Appendix D: Lab One – Introduction to the SmartMotor Lab Manual

**ME 190 Lab 1 - Introduction to Animatics Smart Motor controller**

**San Jose State University**

**Department of Mechanical Engineering**

**Fall 2014**

**Dr. Burford Furman**

**Josh Cobain**

**Ryan Reardon**

**Learning Objectives:** At the end of this lab you should be able complete and understand the following concepts:

* Getting familiar with the SMI Software interface, features, and functions
* Better understanding of the advantages in using a Moog-Animatics SmartMotor
* Basic SmartMotor programming utilizing inputs from external I/O from the BCD controller

**Pre-Lab tasks:**Please follow these links to download the Smart Motor Interface(SMI) software and drivers to control the animatics smart motors. **Please follow the step by step instructions that they provide for the drivers**.

* **SMI download**: http://www.animatics.com/products/smartmotor/animatics/host-software/smi-smartmotor-interface.html
* **Drivers**: **Download the Install notes.txt and follow the directions** http://www.animatics.com/index.php?option=com\_content&view=article&id=104&Itemid=9&dir=JSROOT%2Ftop+level%2F3.+SMI+Software+And+Drivers/USB+Driver
* **Review Command Overview in the Animatics Institute Training Manual** (pg84-88)

**Introduction**

The use of Moog-Animatics SmartMotors for high precision motion control can provide engineers with an advanced solution that is easy to implement. Generally, engineers turn to using conventional motion control systems. These systems are often separate components consisting of separate motion controllers, drive amplifiers, motors, and feedback devices. This causes multiple problems for an engineering as using separate components means higher system costs, higher rate of failure, slower time-to-market, and increased difficulty to troubleshoot or diagnose. This is due to the highly increased complexity of the system by using multiple components. The Moog-Animatics SmartMotor eliminates the complexity and difficulties by providing an all-in-one motion control solution.

A Moog-Animatics SmartMotor incorporates the motion controller, drive amplifier, motor and feedback devices into a single package, creating an integrated servo system. This integrated servo system reduces costs as separate components do not need to be purchased. Because separate components do not need to be sized and selected, the overall system’s time-to-market is greatly decreased as motion control can be controlled by the SmartMotor itself. The use of SmartMotors increases the system’s simplicity, increasing the reliability of the system and decreasing diagnostic and field repair costs.

This lab works to provide engineering students with an in-depth introduction and general product knowledge of SmartMotors. By exposing students to SmartMotors, they can be learn about a commercially available motion control solution. This will give them motion control knowledge that can be used in their professional careers and can lead to them choosing Moog-Animatics products for future applications.

**Equipment:**

* Moog-Animatics Class 4 Integrated Servo System
* CBLSM-DEMO1 Communications/Power Cable
* 24VDC Power Supply
* RS232-to-USB Adapter (RS232485TUSB)

**Exercises:**

**Exercise 1**: Assemble Class 4 Servo System.

Please refer to the **image below** for clarification of how to assembly the class 4 servo system.

1. Plug USB end into computer, and plug Micro USB end ---> RS232-to-USB adapter.
2. RS232-to-USB adapter ---> CBLSM-DEMO1( USB adapter to 9pin)
3. CBLSM-DEMO1 ---> Smart Motor
4. 24VDC power supply ---> CBLSM-DEMO1(side of 5 pin, that connects to motor)
5. Smart Motor Smart Box BCD ---> Smart Motor

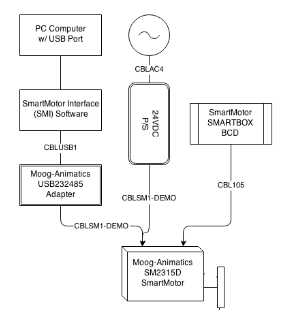


Figure 1: Image showing how the Class 4 Servo System is assembled

**Exercise 2**: Connecting the SmartMotor to the SMI program and troubleshooting

Once in the Class 4 Servo System is assembled correctly. Communicating to the motor is important. Please **open document** called *Class 4 SmartMotor Training Overview*. Go to page 6 for **Serial Communication Set-up** and follow the directions

If problems persist, please follow the **Serial Communication Troubleshooting** on page 8 of *Class 4 SmartMotor Training Overview.*

If problems are still persistent, please re-download the drivers for the SmartMotor and try again.

**Exercise 3:** Motor View

Once the serial communication is setup for the SmartMotor, open the SMI program. To select Motor View, open up the SMI program and follow these directions.

Select Tools --- Select Motor View---Select motor used-- Click Poll.

Should see that the motor is now online. **What does one rotation equal to in for the position encoder?**

**What does each toggle switch on the SMARTBOX indicate in the I/O section?**

**What Version of motor is used for this lab?**

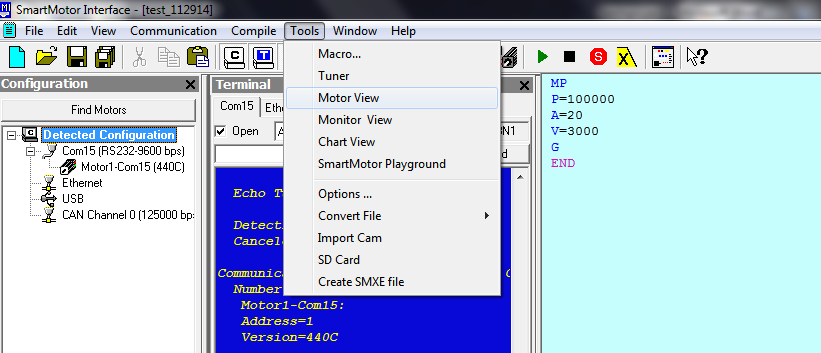
****

Figure 2:Selecting Motor View.

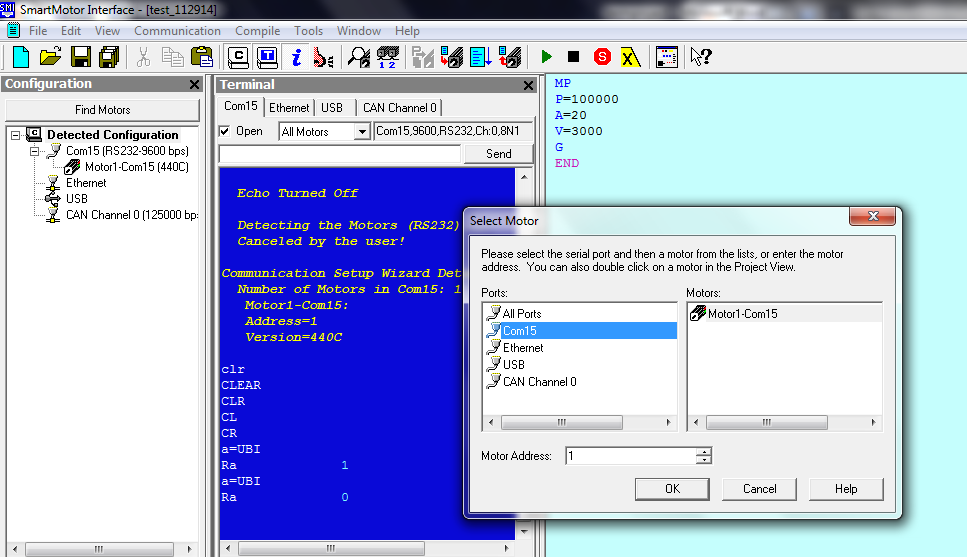


Figure 3: Selecting the motor for Motor View.

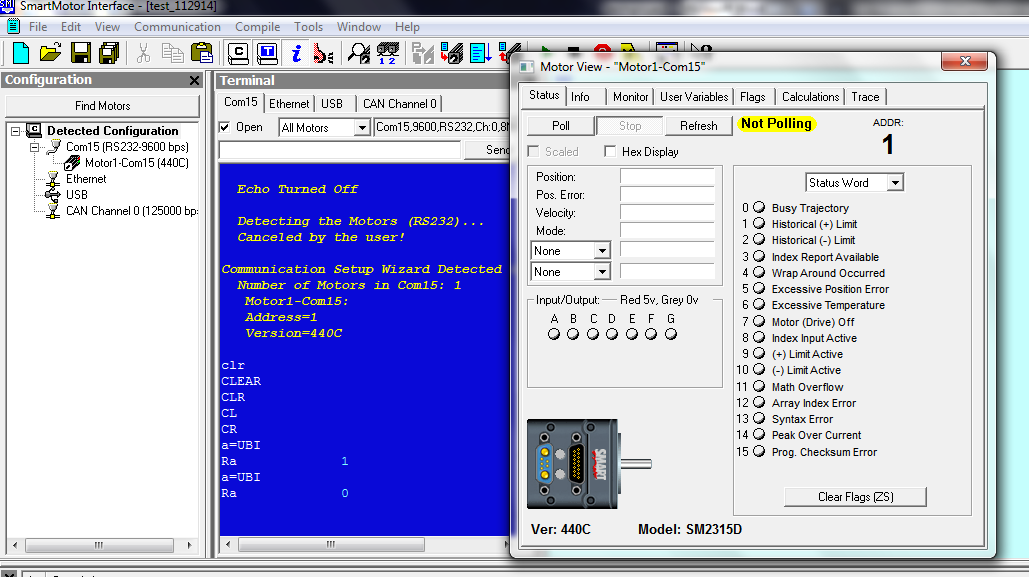
****

Figure 4: Selecting Poll.

**Exercise 4:** Monitor View

Select Tools--Monitor View

Add the motor used for this lab with the green plus on the top of the screen. Once the motor is selected, click poll. Spin the motor with your hand, **what do you see changing?**

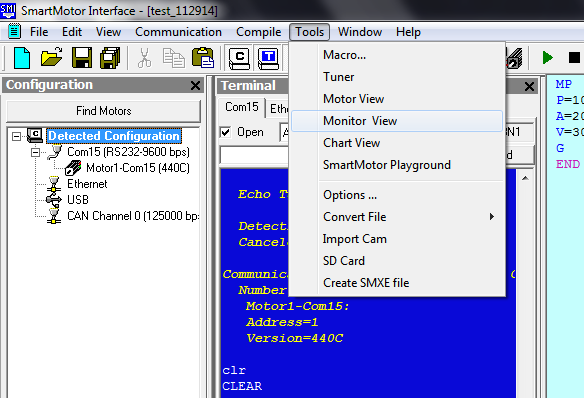
****

Figure 5: Selecting Monitor View

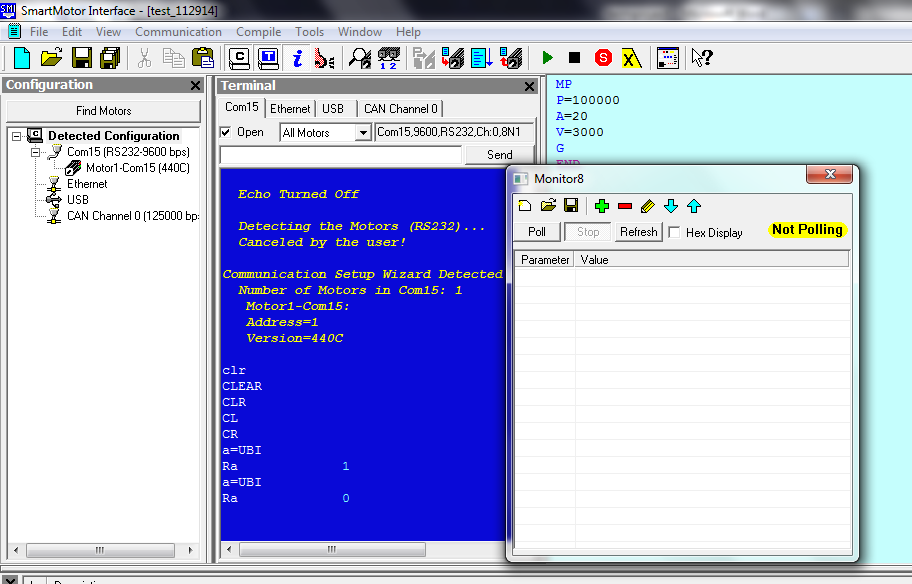


Figure 6: Selecting the Green plus sign to add the motor to the Monitor View.

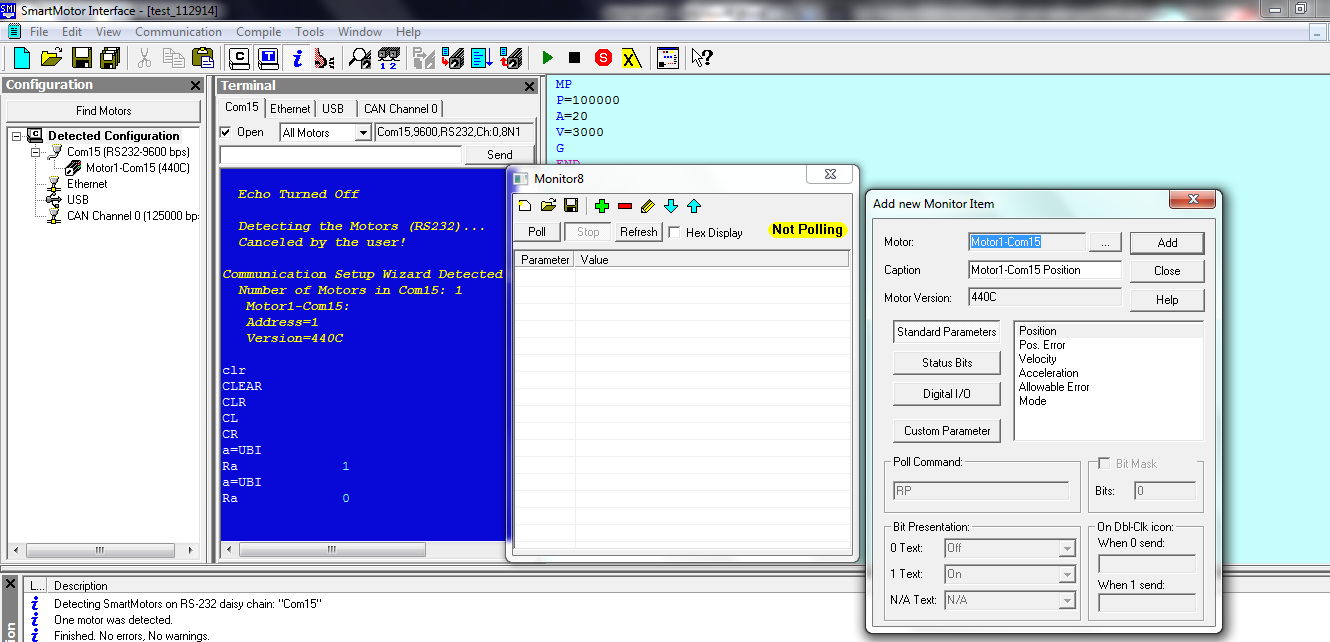


Figure 7: Selecting the motor for the Monitor view, once added select close.

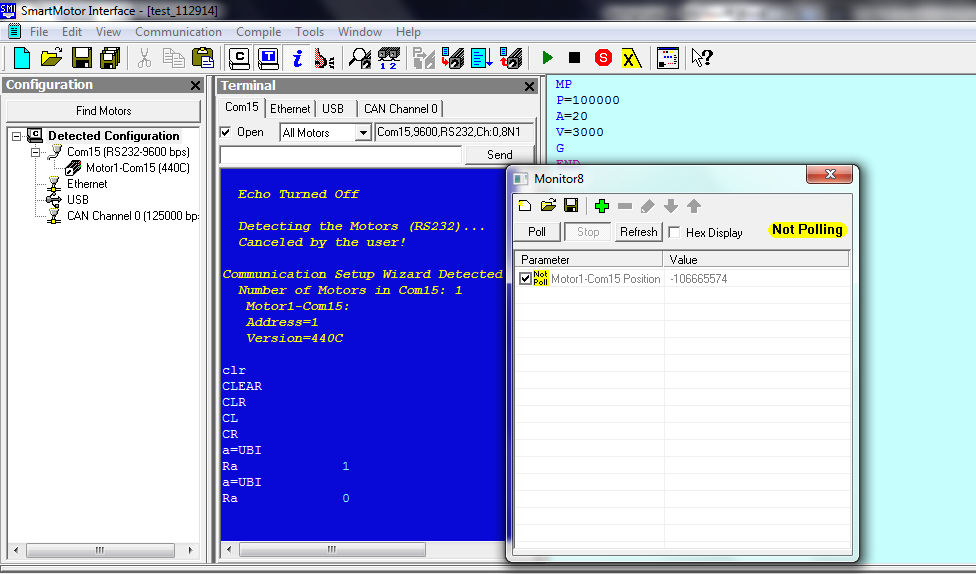


Figure 8: Select Poll in the Monitor window.

**Exercise 5:** SmartMotor Playground

Select Tools--SmartMotor Playground

Once the SmartMotor Playground is open select Connect to connect the motor to the SmartMotor Playground. Once the motor is connect, select the positions tab.

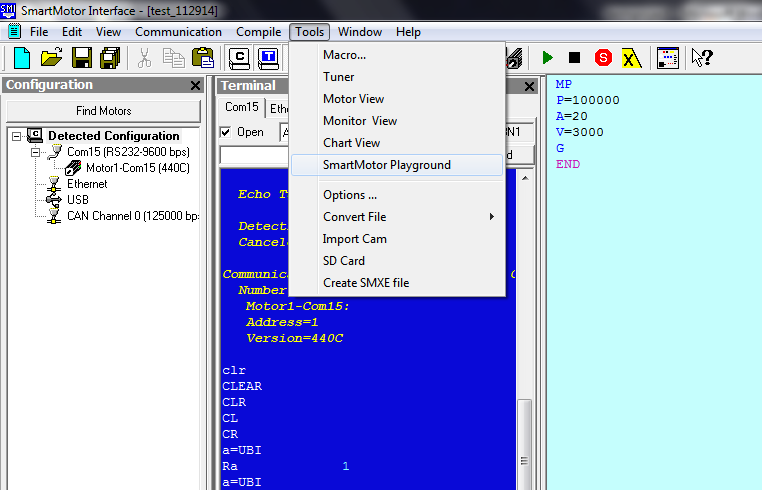


Figure 9: Selecting SmartMotor Playground.

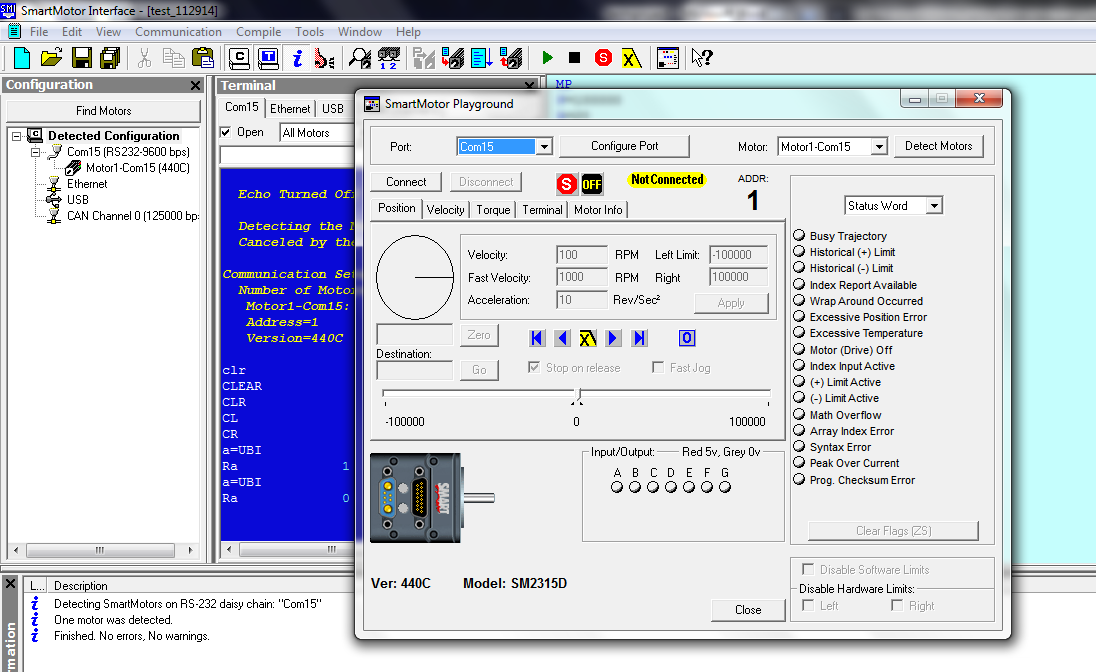


Figure 10: Select Connect in the SmartMotor Playground

Click zero, then type 100000 into Destination:---select Go. **What does the motor do?**

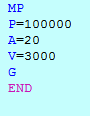
Select the Velocity tab. Drag the slider up or down to 1000 or -1000. **What does the motor do? When 60RPM is typed into max speed, is this close to 1rev/s?**

Select the Torque tab. **Type in 512 for max torque, you'll make the power supply angry. What is the maximum velocity attained with torque at 512?** If the power supply is to shut off, unplug the power supply and wait for fifteen minutes.

**Exercise 6:** Programming, MP

Open a new document by selecting File --- New. A new window should appear on the right side. In this Editor window is where the code is downloaded to the motor.

Type in this simple code to make the SmartMotor move to the declare position.



**Figure 11:**  Simple code to make the motor move to the position P using MP(Mode Position). A is acceleration and the V is velocity. The G is reference to Go and END, ends the program once at the destination.

Once the code is typed in the editor window, Right click in the Editor window--- select Compile & Transmit SMX file, or press f5--- Select the motor used on the right side, double click the motor--- Click Yes --- OK---Run.

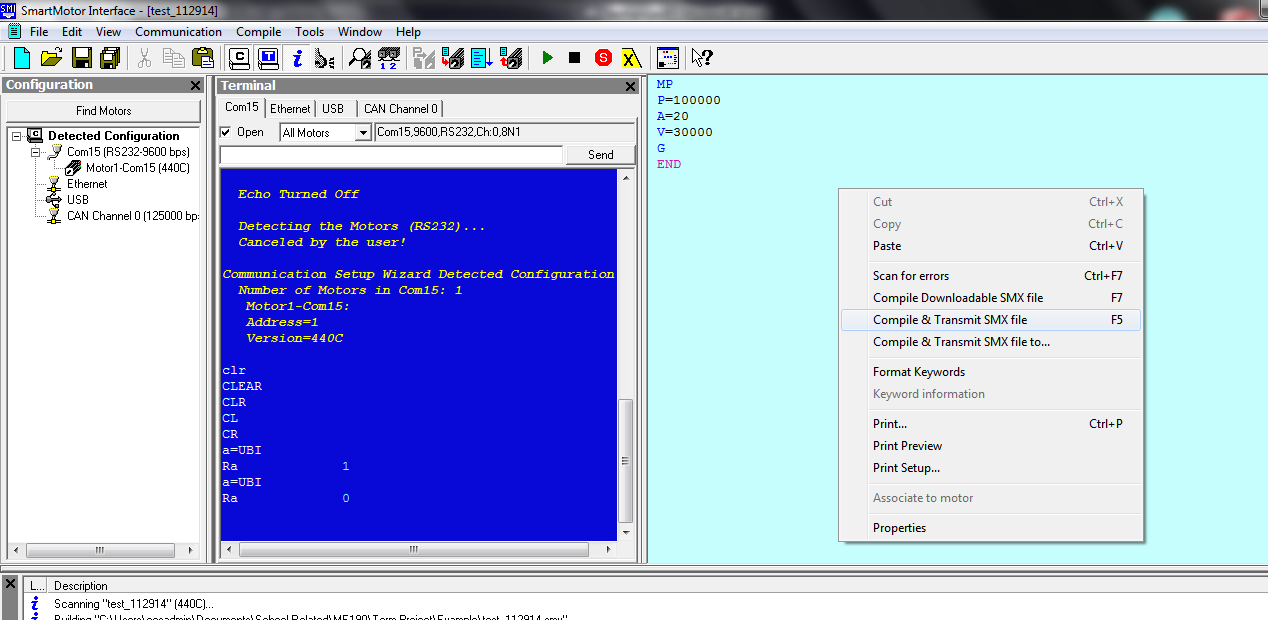


Figure 12: Right click over Editor Window and click on Compile & Transmit SMX file, or press f5

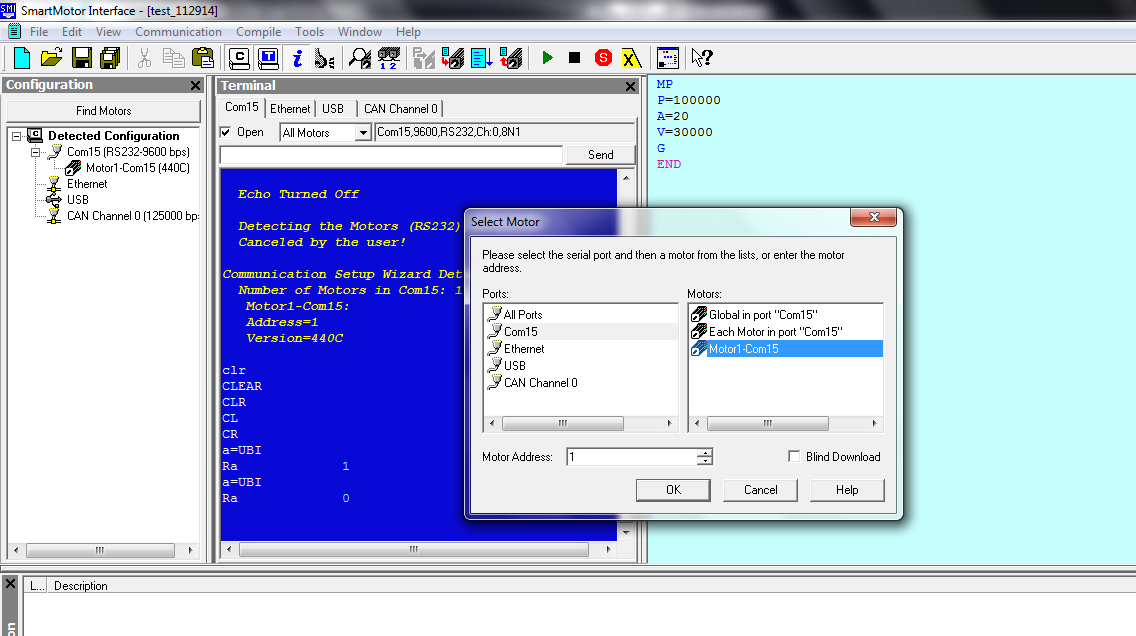


Figure 13: Select motor used on the right hand side.

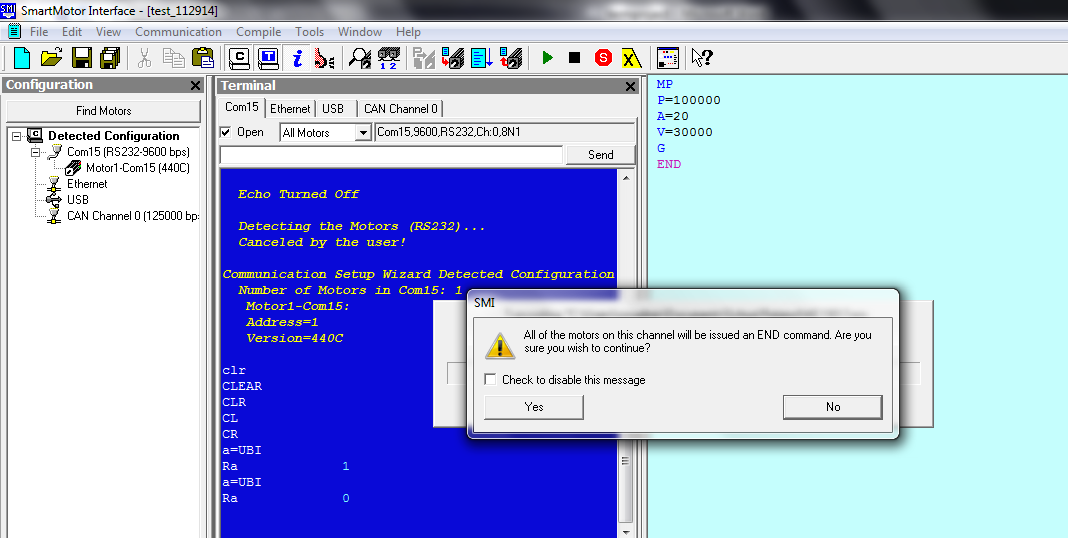


Figure 14: Select Yes to all of the motors on this channel.

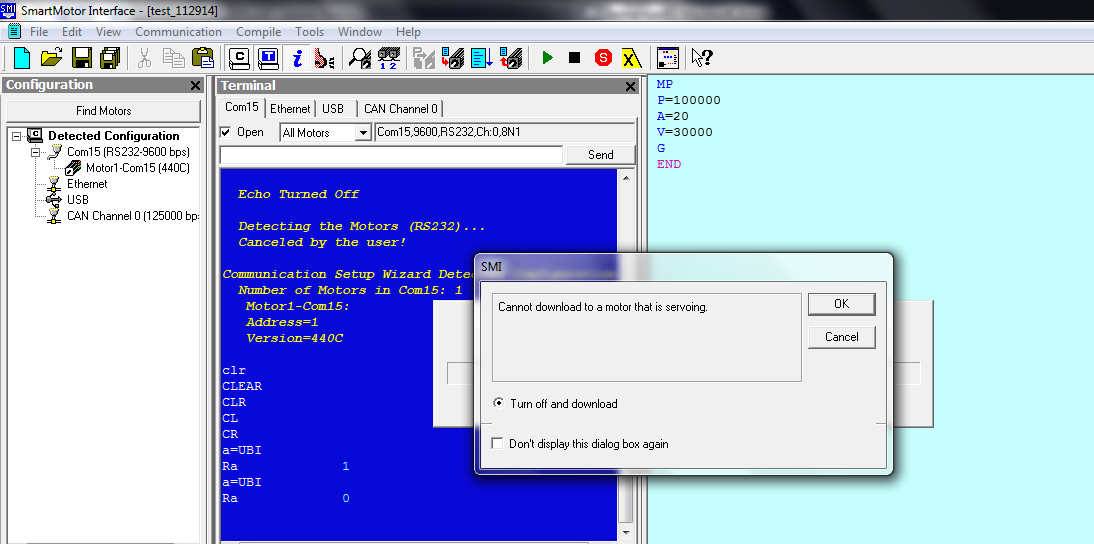
****

Figure 15: Select OK, then press RUN

**What does the motor do once the program is downloaded?**

Once this program is execute, declare V=30000. **Is the RPM with respect to the shaft output or the encoder?**

**Exercise 7:** Programming, MV

Open a new document and put in the following code. To upload the program to the SmartMotor, follow the above steps in Exercise 6.

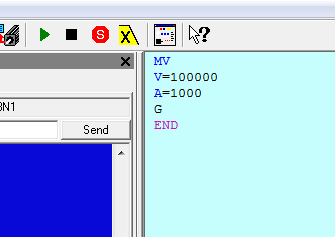


Figure 16: Mode Velocity

When downloaded to SmartMotor**, what does the SmartMotor do?**

**Exercise 8:** Programming, MT

Open a new document and put in the following code.

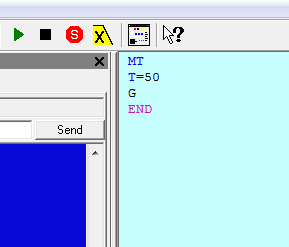


Figure 17: Mode Torque

**What happens when toggle switch 8 on the BCD is flipped?** Now put toggle switch 8 back to original position and GO on the BCD is flipped**, what happens when GO is flipped on?**

**Exercise 9:** Going between positions using TWAIT

Create a new document and put in the following code. Use Monitor View to determine if the SmartMotor spins in both directions.

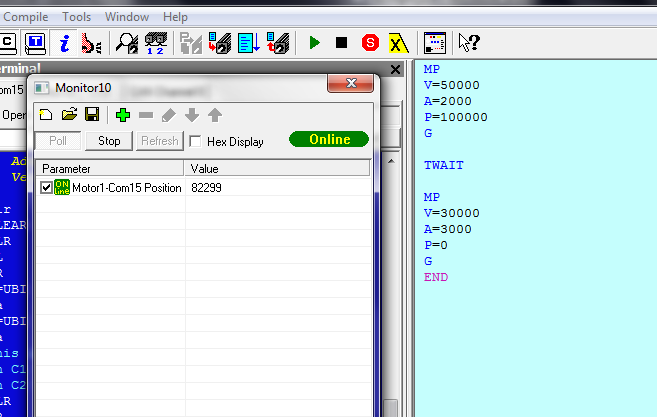


Figure 18: Example of two positions using the TWAIT function between different commands. Use monitor view to keep track of position.

**Exercise 10:** IOs of the BCD controller

Open up the SmartMotor Playground and see what Toggle switch corresponds to the red dots(i.e. ABCDEFG)

In the terminal window of SM,. Type inthe following commands seen in figure 19

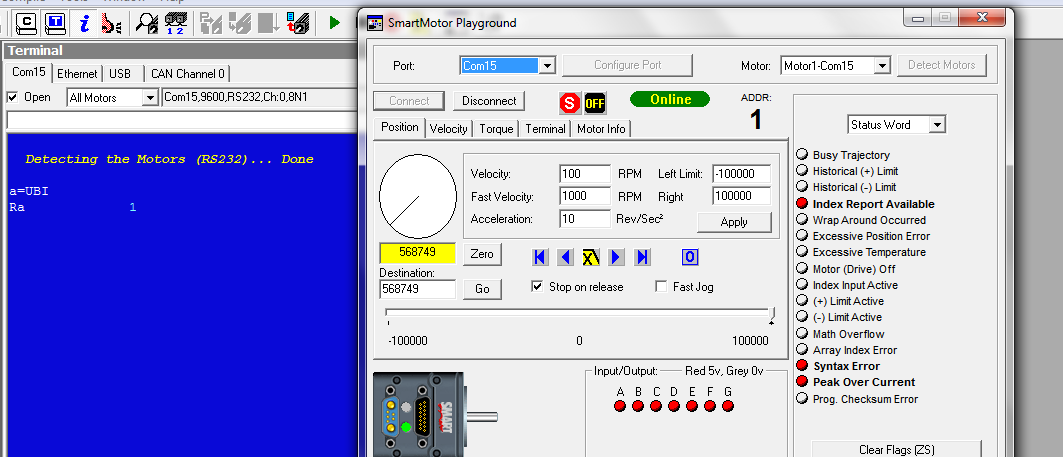


Figure 19: In the terminal screen in blue, type: a=UBI, then type Ra. This Ra commands reports what the current value for B under IOs. The current position of B is valued at 1.

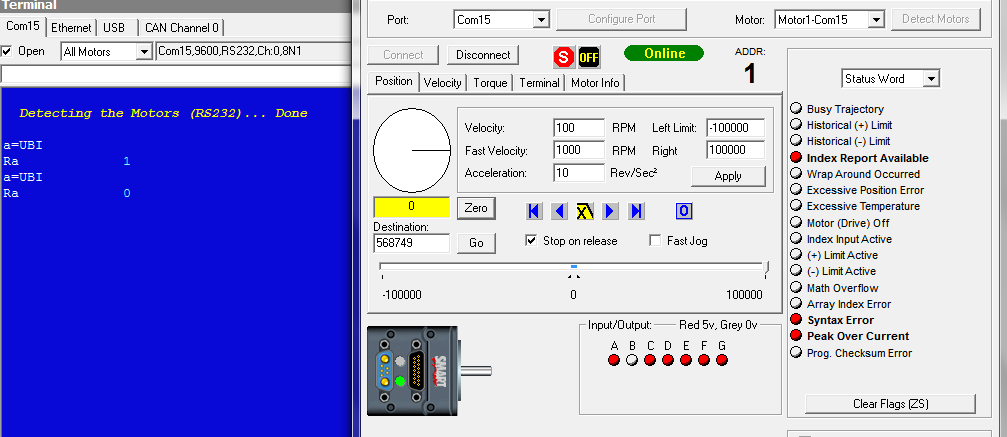


Figure 20: Declare what a is again by typing, a=UBI, then type Ra. When the B is flipped to a different position, the reporting value is now 0.

**Is each toggle switch on the BCD controller 1 or 0? (**Hint: set UAI or UCI equal to a constant like above in the Terminal window, and use Ra or Rc to report the value(1 or 0) in the terminal window)

**Exercise 11:** Using IOs to go between positions

Type in the following code.

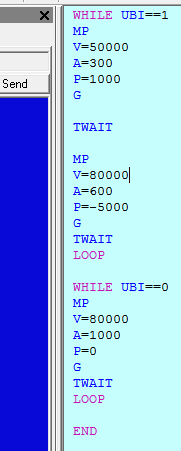


Figure 21: Example of how to use the BCD controller with toggle switch B on the BCD controller.

When the program is uploaded to the SmartMotor, **what does the SmartMotor do when the program is executed? Play with the toggle switches and report what happens.**

**Exercise 12: Programming for 4 different positions**

Now program the Smart Motor to do the following:

* Go to 4 different positions using the BCD controllers 4 toggle switches.
* Has to go to that position by flipping the GO toggle switch and the toggle switch designated for that specific position.
* Must be able to go to that position multiple times using the BCD controller.

(Hint: Look at the command overview in the Animatics Institute Training Manual(pg84-88))

# Appendix E:Lab Two – Advanced SmartMotor Programming and Controls Lab Manual

**Moog-Animatics SmartMotor Day 2 Advanced Lab**

**San Jose State University**

**Department of Mechanical Engineering**

**Fall 2014**

**Dr. Burford Furman**

**Josh Cobain**

**Ryan Reardon**

**Introduction**

The Moog-Animatics SmartMotor provides an all-in-one motion control solution. It provides users with a simple solution by handling all advanced commutation, drive-stage, and communication interfacing. Although it allows for an initial system that is easy to interface and give basic control, the SmartMotor also provides advanced programming and control capabilities.

**Interesting and Useful Resource Links**

1. Moog-Animatics Website: <http://www.animatics.com>
2. Example of creating a scripted PID controller in C-language programming: <http://www.element14.com/community/servlet/JiveServlet/download/108711-132593/How%20To%20Write%20PID%20control%20algorithm%20using%20C%20language%20_%20TipsWell.pdf>

**Lab Objectives**

After completing this lab, students should be able to complete and understand the following concepts:

1. Advanced SmartMotor programming techniques
2. Programming of a scripted PID controller
3. PID Tuning
4. Effects of PID components on system response

**Equipment Needed**

1. Moog-Animatics Class 4 Integrated Servo System
2. CBLSM-DEMO1 Communication / Power Cable
3. 24VDC Power Supply
4. RS232-to-USB Adapter (RS232485TUSB)
5. Large and Small Pulleys
6. A mass (< 1 kg)
7. SmartMotor Interface (SMI) Software

**Procedure**

*Closed-Loop PID Tuning (Exercise 1)*

1. In this exercise, use the chart view utility to plot the position error and velocity of the motor.
2. Attached the large pulley to the motor shaft, and attach a mass to the pulley using the supplied string.
3. Clamp the motor at the edge of a work bench, so the pulley and mass hang off the side of the work bench (See Appendix A).
4. Place the motor in Position Mode and zero the motors position with the mass at the bottom of its travel. Then determine the maximum motor position where the motor will be at the top of its travel.
5. Set the motor’s PID settings to mimic a Proportional controller. This can be done by setting the integral and derivative gains to zero.
6. Set the velocity and acceleration values and command the motor to go to the maximum position value that was previously determined.
7. Observe the motor’s response in the chart view window. Stop the chart record and copy the data and save the data to an excel file.
8. Repeat steps 5-7 adjusting the P, I, and D values and observe how it effects the response of the motor system. Include the values used for a P, PI, and PID controller in your report.

*Programming and Creating a PID-based Position Controller (Exercise 2)*

1. In this exercise, use the chart view utility to plot the position error and velocity of the motor.
2. Attached the large pulley to the motor shaft, and attach a mass to the pulley using the supplied string.
3. Clamp the motor at the edge of a work bench, so the pulley and mass hang off the side of the work bench (See Appendix).

Create a program in SMI that will drive the motor’s position to the maximum position determined in the previous exercise. This program will need to place the motor in mode Torque (MT). The input of the controller is to be the error between the current position and the target position. The output of the controller is to be a torque value (T= ±1024). Test and optimize your program to provide the best response. Repeat using the smaller pulley. Include the program script and the values used to obtain the best response in your report.

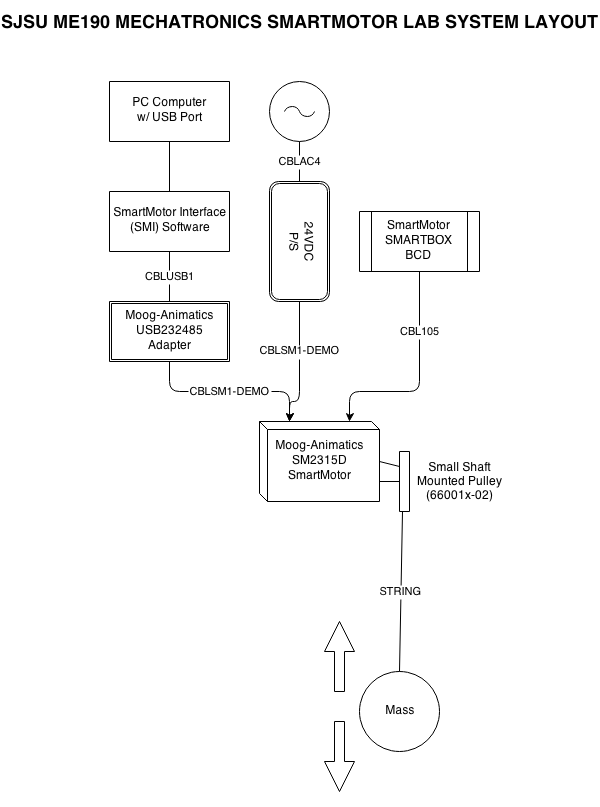
**Questions and Ideas to Consider**

1. How did the proportional gain effect the response of your system? What effect does the integrator have? What effect does the derivative term have?
2. How did the response of your PID program compare to the Closed-Loop position mode response?

**Important Details to Include in Report**

1. Summary of the experiment (including work from lab 1)
2. Observations of effects that different values for the PID controller in the SmartMotor had on the system
3. Observations on the system response from the scripted PID controller
4. Any source code used

**Appendix A: System Layout and Connection Diagram**



# Team Evaluation – Ryan Reardon

|  |  |  |
| --- | --- | --- |
|  | Name of Team Member | |
| Criteria | Ryan Reardon | Josh Cobain |
| **1. Attending meetings** | 2 | 2 |
| Comments/Justification: | Both made time to meet and collaborate on project | |
| **2. Contribution to idea generation** | 2 | 2 |
| Comments/Justification: | Both created individual manuals that creatively met project requirements | |
| **3. Contribution to decision making** | 2 | 2 |
| Comments/Justification: | Collaborated for decision making | |
| **4. Communication skills** | 2 | 2 |
| Comments/Justification: | Both showed competent understand of project components | |
| **5. Assuming fair share of work** | 2 | 2 |
| Comments/Justification: | Both contributed equally to final project | |
| **6. Completion of work assumed** | 2 | 2 |
| Comments/Justification: | Both met project deadlines | |
| **7. Team spirit and respect for others** | 2 | 2 |
| Comments/Justification: | Both were respectful and supportive of others | |
| **8. Negotiation and conflict resolution skills** | 2 | 2 |
| Comments/Justification: | No issues were observed | |
| **9. Time management** | 2 | 2 |
| Comments/Justification: | All deadlines were met | |
| **10. Dependability and preparedness** | 2 | 2 |
| Comments/Justification: | Work from both were of quality | |
| **Totals** = | 20 | 20 |

|  |  |
| --- | --- |
| Name | Roles and Responsibilities of Team Member |
| Ryan Reardon | Project Lead / Main Report / Day Two Manual / Presentation / Moog-Animatics Contact / Project Video |
| Josh Cobain | Main Report / Day One Manual / Presentation / Project Video |

Print name of evaluator: **Ryan Reardon** Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 12/8/14

# Team Evaluation – Josh Cobain

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Name of Team Member | | | | |
| Criteria | Josh Cobain | Ryan Reardon |  |  |  |
| **1. Attending meetings** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **2. Contribution to idea generation** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **3. Contribution to decision making** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **4. Communication skills** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **5. Assuming fair share of work** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **6. Completion of work assumed** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **7. Team spirit and respect for others** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **8. Negotiation and conflict resolution skills** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **9. Time management** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **10. Dependability and preparedness** | 2 | 2 |  |  |  |
| Comments/Justification: |  | | | | |
| **Totals** = | 20 | 20 |  |  |  |

|  |  |
| --- | --- |
| Name | Roles and Responsibilities of Team Member |
| Joshua Cobain | Pre-lab homework, lab 1-basic SMI programming, Lab report, and presentation |
| Ryan | Lab 2-advanced PID creation and tuning, Lab report, and presentation. |

Print name of evaluator: Joshua Cobain Signature: *Joshua G. Cobain* Date: 12/8/2014