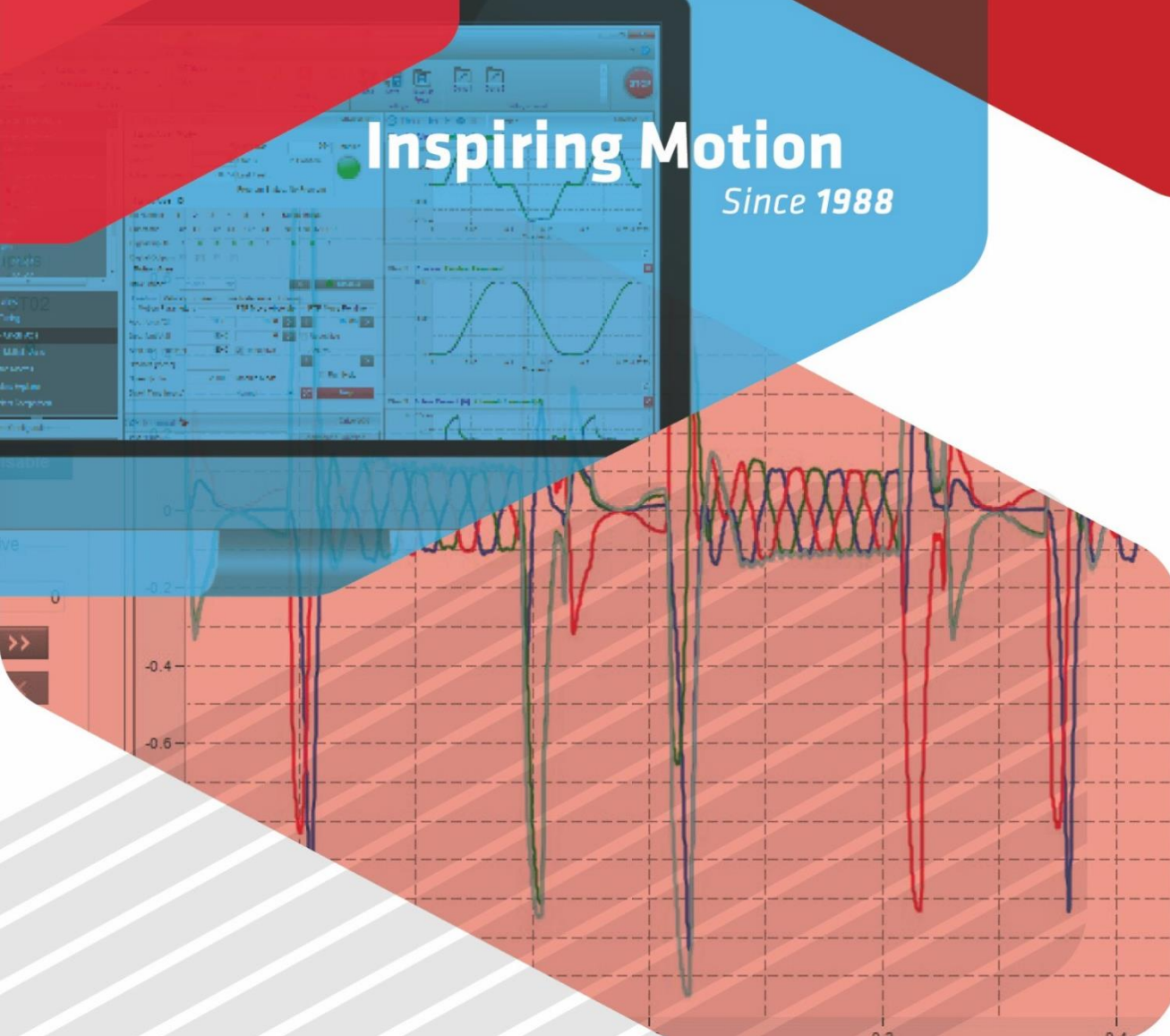


Inspiring Motion
Since 1988



Exercise #7

Drive User Program

Background

Gold servo drives use a communication language that enables the user to:

- Set up the drive.
- Send commands to the drive indicating which functions to perform.
- Query the drive's status.
- Write user programs in the drive language and store them in the drive's non-volatile memory.
- The drive can run the program with minimal or no assistance from the host.

Note: The full set of drive commands is documented in the Command Reference for Gold Line Drives.

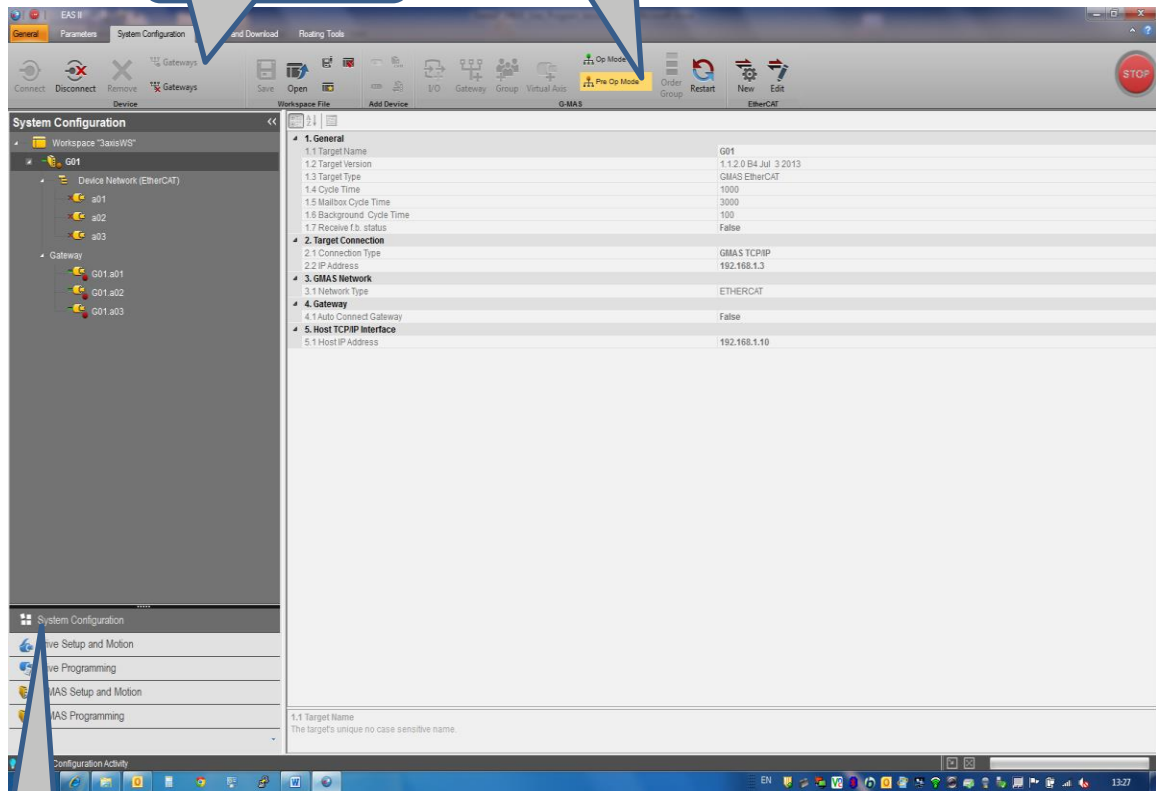
List of parameters for this demo			
Set according to screenshots		Parameter	Value
		GMAS IP	192.168.1.3
		Computer IP	192.168.1.XXX

Exercise #7: Drive User Program

1. Starting New Project.

3. Gateways

2. Pre Op Mode



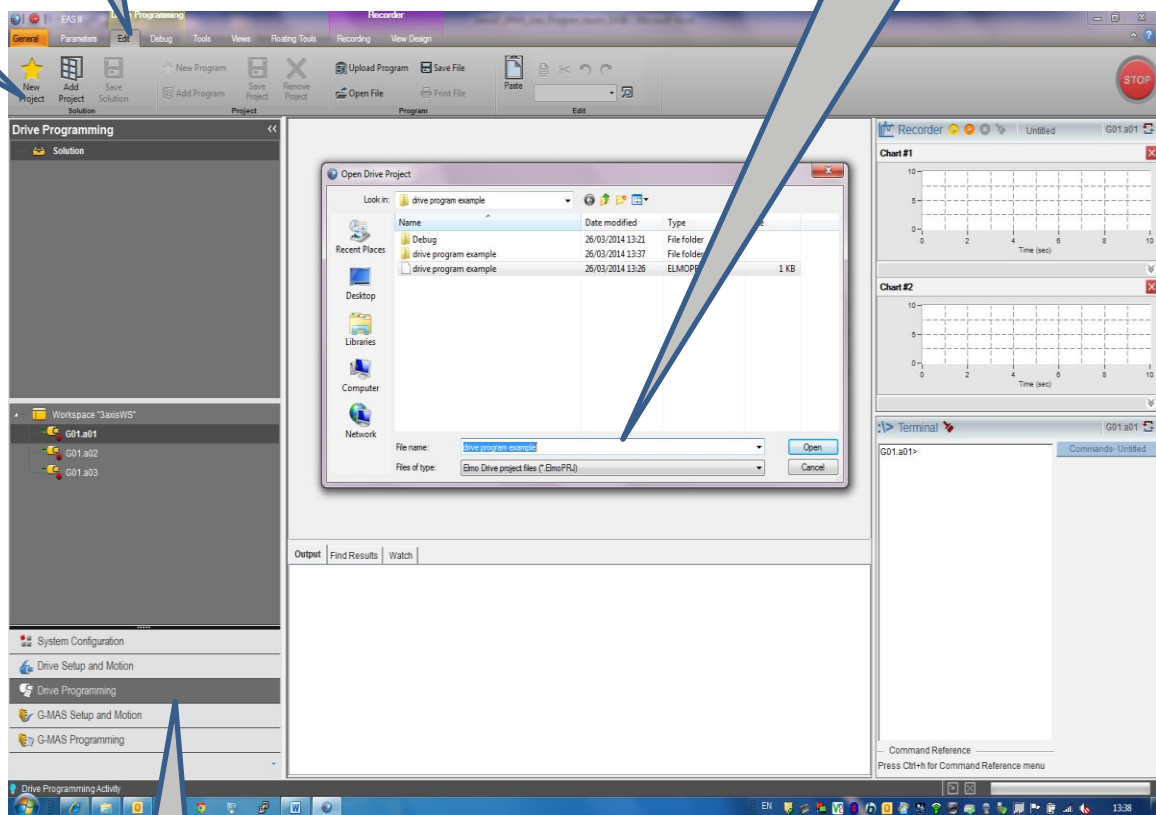
1. System Configuration

Exercise #7: Drive User Program

5. Edit

6. New Project

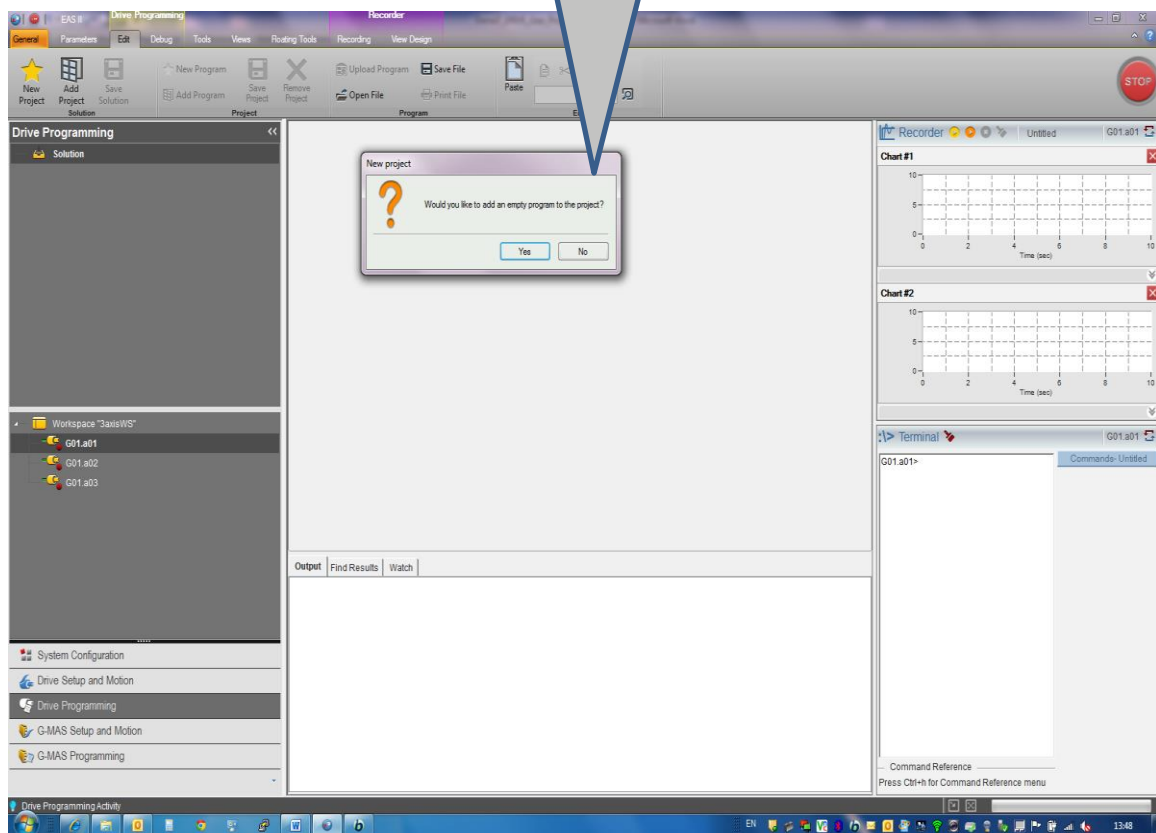
7. Name the Project



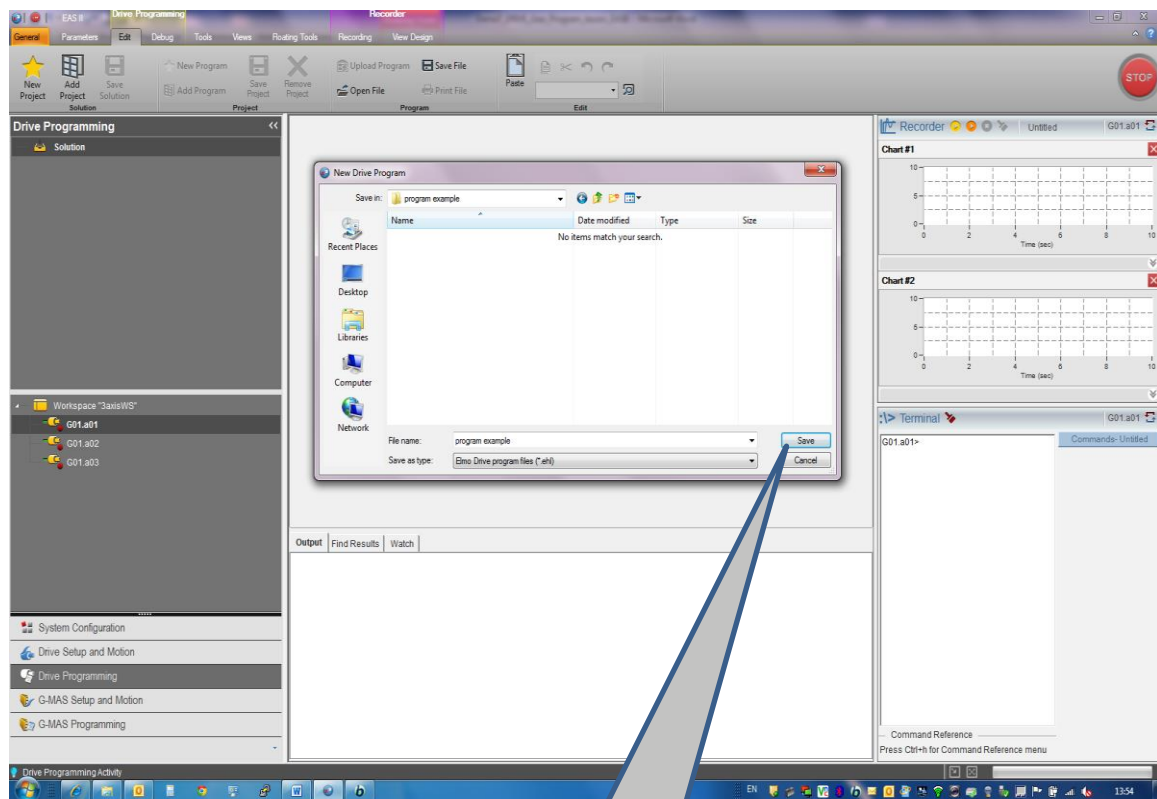
4. Drive Programming

Exercise #7: Drive User Program

8. Accept and add a new program



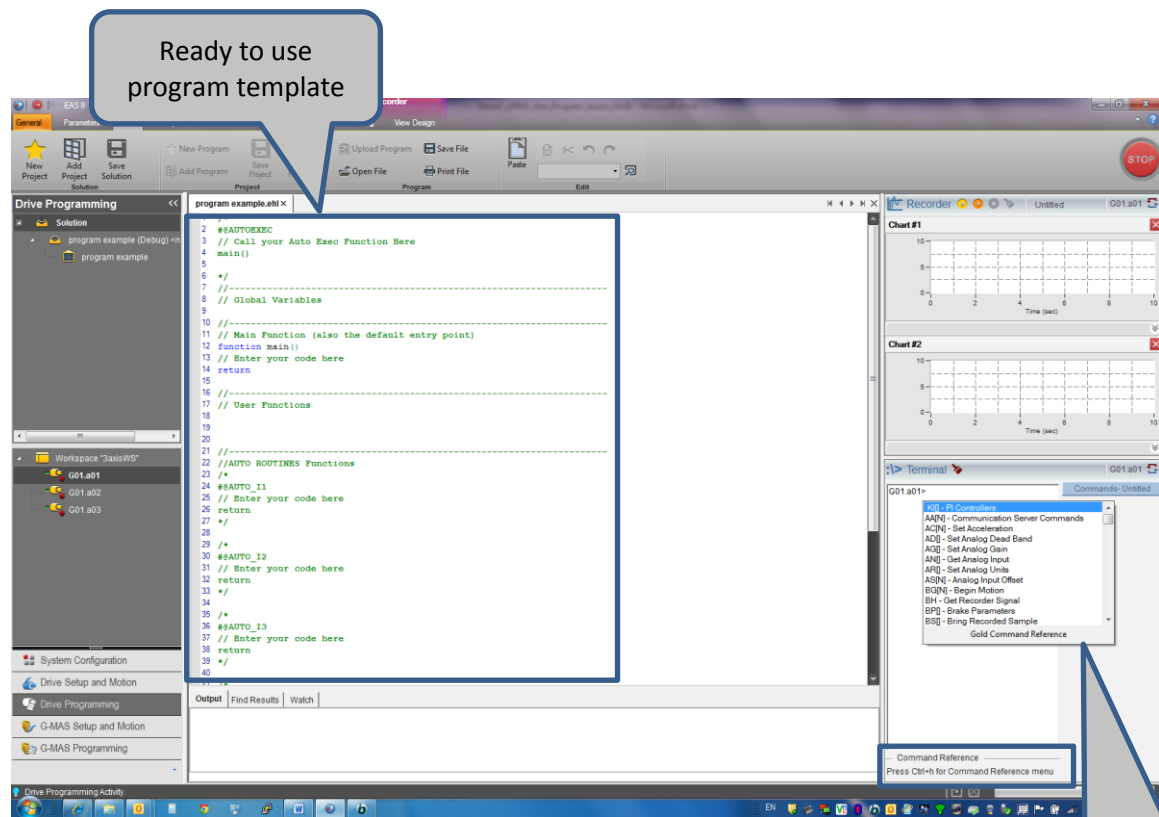
Exercise #7: Drive User Program



9. Type a new program name and press save

Exercise #7: Drive User Program

2. Introducing the program tools.



Pressing Ctrl+h in the Terminal area opens the command reference menu

Note: You can find more commands in our Command reference Manual

3. Write the program code.

*First delete the template code,
Then paste the sample code in its place:*

```
##START // START label

P2P_Rel(100000,50000) //Call to function with two variables (distance & speed)

function P2P_Rel(int Distance, int Speed) // Start function

if (UM!=5) // Ensure motor is in position mode
    MO=0    // Disable motor
end

if (MO==0) // Change the motor to position mode and enable the motor
    UM=5    // Set position mode
    MO=1    // Enable motor
end

while(1) // Endless loop
    PR= Distance // Set relative target position
    SP= Speed    // Set motion speed
    BG    // begin motion
    wait(5000) //delay of 5 sec
end

return;
```

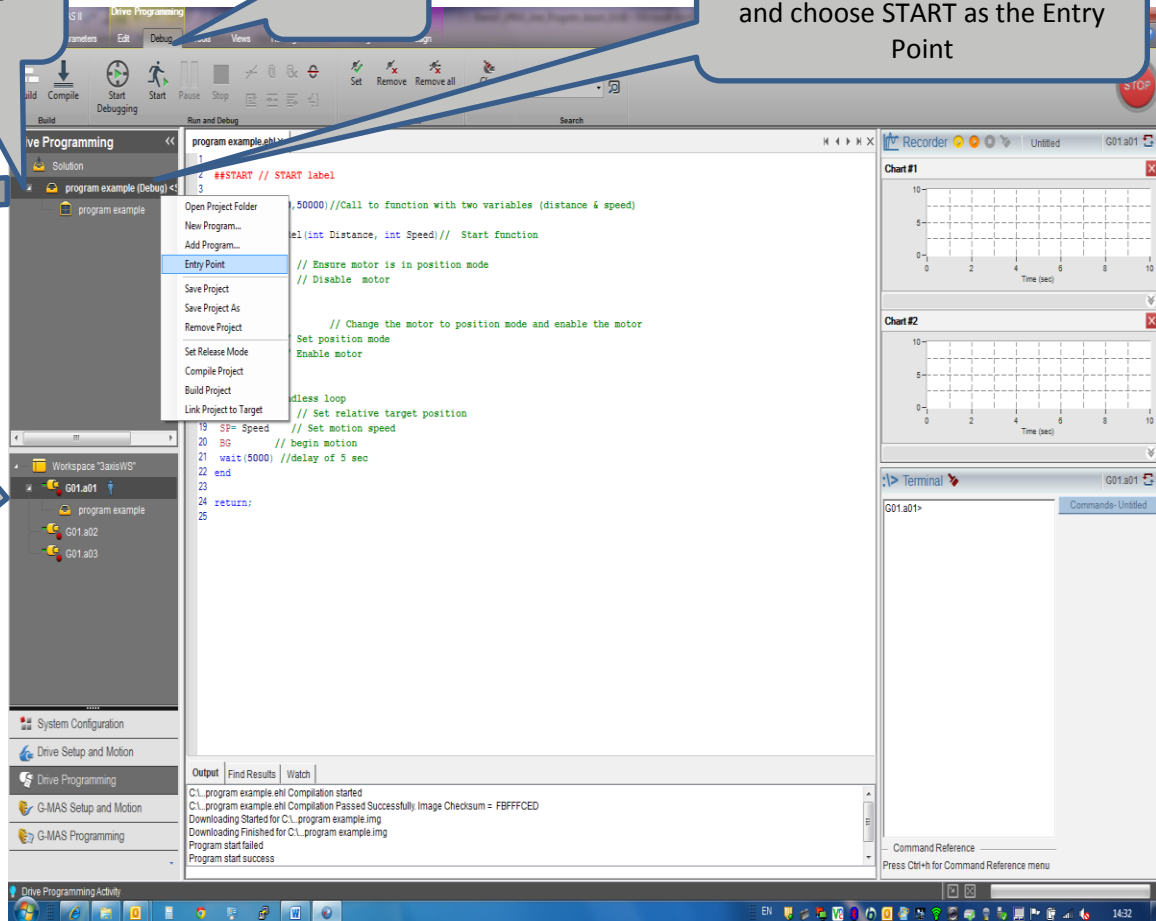
Exercise #7: Drive User Program

4. Link project to target and set Entry Point.

2. Drag the project to target drive

1. Debug

3. Right click on project name and choose START as the Entry Point



5. Compile, build and start the program

The screenshot displays the Elmo Drive Programming software interface. The main window shows a C-like program in 'program example.ehl'. The code includes a function call to 'F2F' with parameters 100000 and 50000, followed by a function definition for 'F2F' that sets motor parameters and enters a while loop. The left sidebar shows a project tree with 'program example' and its sub-items. The bottom status bar shows the output of the compilation process.

```
1 //START // START label
2
3 F2F(100000,50000)//Call to function with two variables (distance & speed)
4
5 function F2F(int Distance, int Speed)// Start function
6
7 if (TM1=5) // Enable motor is in position mode
8 NO=0 // Disable
9 end
10
11 if (NO=0) // Change the motor position mode and enable the motor
12 TM=5 // Set position mode
13 NO=1 // Enable motor
14 end
15
16 while(1) // Endless loop
17 SP= Distance // Set relative target position
18 SP= Speed // Set motion speed
19 BG // begin motion
20 wait(5000) //delay of 5 sec
21 end
22
23 return;
```

Output: C:\program example.ehl Compilation started
C:\program example.ehl Compilation Passed Successfully Image Checksum = FBFFFCED

Terminal: G01.a01
Commands: Unlabeled

3. Start, the motor will turn 100,000 cnt, wait 5 and repeat again (endless loop)

1. Compile, to find errors in code

2. Build, convert to machine language and download the program to the drive