

Application Note

C2000 Motor Control SysConfig Tool



Nima Eskandari and Marlyn Rosales

ABSTRACT

The C2000™ Motor Control SysConfig Tool makes it easy to implement motor control functionality into a C2000 software project. The Universal Motor Control Lab (UMCL) along with all the supporting motor control libraries are supported. The tool is based on top of the C2000 SysConfig tool so device resources are presented as well. This application report focuses on the UMCL including its capabilities and features all while showcasing the Motor Control SysConfig Tool. The content presented is based on the F28002x device with a Motor Control SDK version 4.02 and a Code Composer Studio™ (CCS) version 12.2.

Table of Contents

1 Introduction	2
2 Project Import	3
3 Main Settings	4
4 Motor Settings	6
5 Device Resources	9
6 Generated Code	10
7 Summary	11
8 References	11

List of Figures

Figure 1-1. Motor Control SysConfig Tool.....	2
Figure 2-1. UMCL Project Import Within CCS.....	3
Figure 2-2. UMCL Project Structure.....	3
Figure 3-1. Build Levels.....	4
Figure 3-2. Demo vs. Custom Mode.....	4
Figure 3-3. Motor Drive Evaluation Board Selection.....	5
Figure 3-4. Debug Features.....	5
Figure 3-5. Command Interfaces.....	5
Figure 4-1. Control Algorithm.....	6
Figure 4-2. Motor Selection.....	6
Figure 4-3. Motor Features.....	6
Figure 4-4. User Definitions.....	7
Figure 4-5. Calculated Values.....	7
Figure 4-6. Tailored Settings.....	8
Figure 5-1. Device Resources.....	9
Figure 6-1. Generated Files.....	10
Figure 6-2. Reference Files.....	11

Trademarks

C2000™ and Code Composer Studio™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

The Motor Control SysConfig Tool has many benefits:

- **One-Click Addition of Libraries:** When a library is added, the required header, source, and library files are added as well as any predefines needed.
- **Device Resource Management:** The tool provides a great graphical interface that showcases what resources have been utilized and how much more the device has. This is specific to the device being used and its pin-count.
- **All of the SysConfig Benefits:** For a full list of SysConfig benefits, see [Speed Up Development With C2000 Real-Time MCUs Using SysConfig](#).

Specifically with the UMCL, the code size is smaller. Generated code is based on the user settings, so if a control algorithm is not being used or a library is not required, the tool does not include the content/files within the project- effectively reducing the code size. Additionally, all settings required to be changed are housed within one view, instead of having to navigate through multiple files to change a setting.

Figure 1-1 shows the layout of the Motor Control SysConfig tool with the addition of the UMCL. On the left hand side are all of the motor control libraries and system resources. Libraries and or resources used have a green check mark next to the name as well as the amount being used. The middle pane contains all of the main, motor, and device settings for the UMCL. On the right hand side is all of the generated files, reference files, and device pin view. The rest of the application note walks through each of the sections individually.

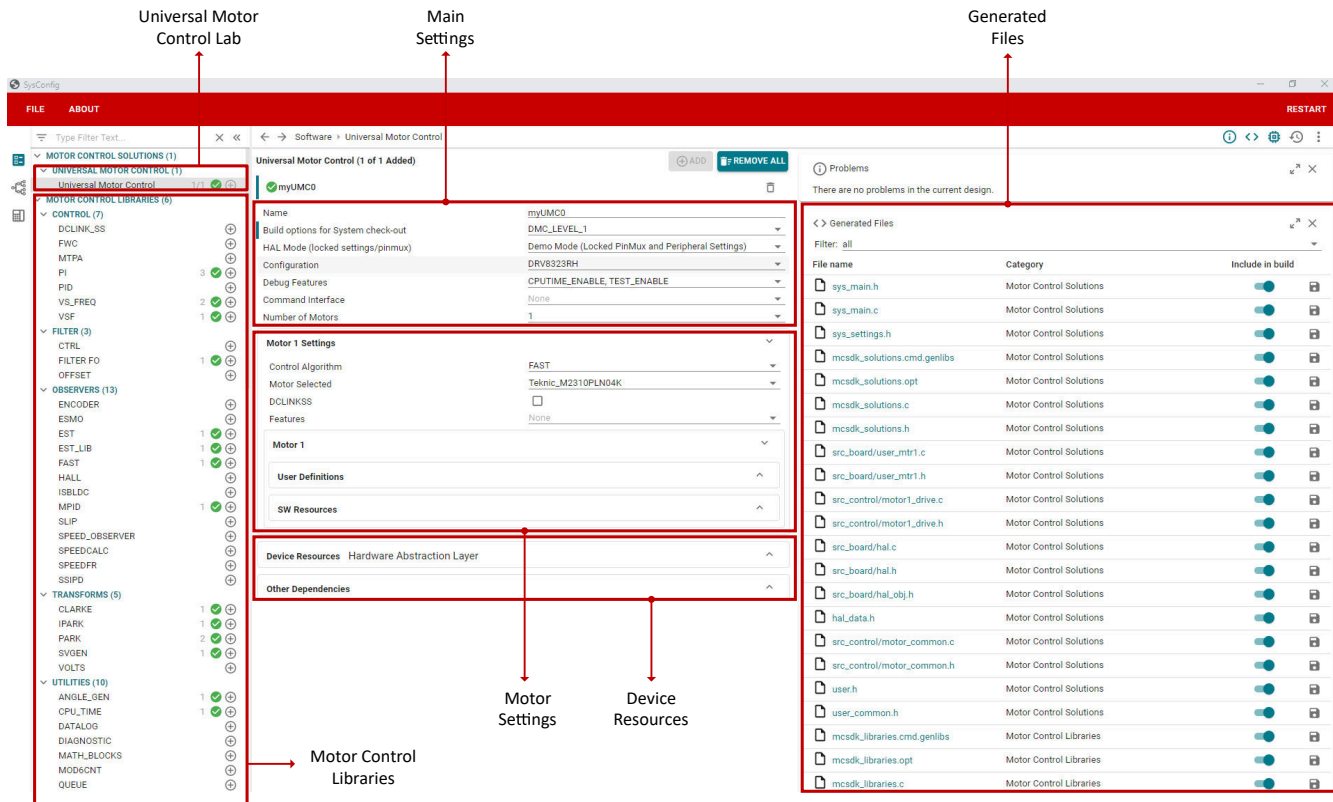


Figure 1-1. Motor Control SysConfig Tool

In order to use the UMCL lab, a launchpad, boosterpack, and motor are required. Follow the instructions in the following document to setup the hardware correctly: [Motor Control SDK Universal Project and Lab](#).

Note

Not all features may be supported in initial releases of this lab. Please refer to the example within the MCSDK for support capabilities offered.

2 Project Import

The UMCL project can be imported into CCS by going to *Project -> Import-> <Path to Project - as seen in the image below>* and clicking on the respective project.

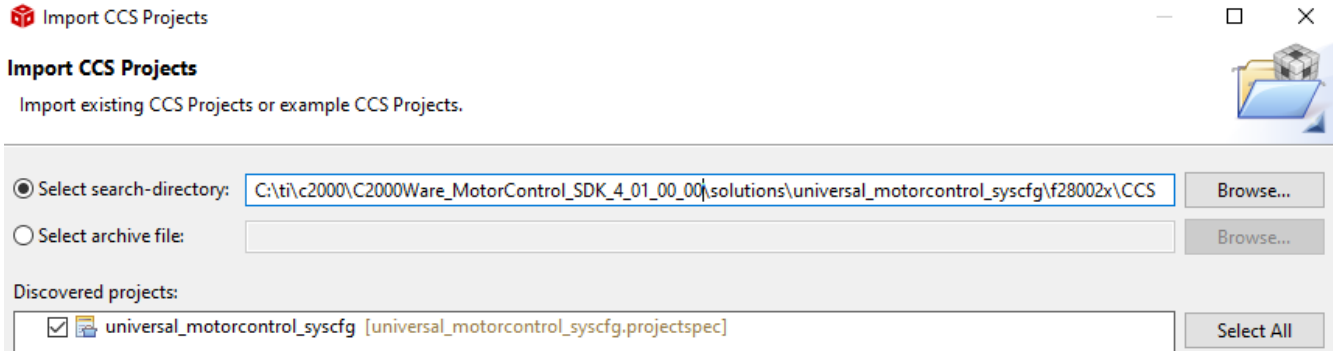


Figure 2-1. UMCL Project Import Within CCS

After importing the project, double click on the 'C2000.syscfg' file in order to open up the Motor Control SysConfig Tool. An instance of the Universal Motor Control Lab is added by default. After building the project (*Project -> Build Project*) the Generated Source folder will contain all of the generated files from the tool. The referenced files will be located within the 'Referenced Source' folder.

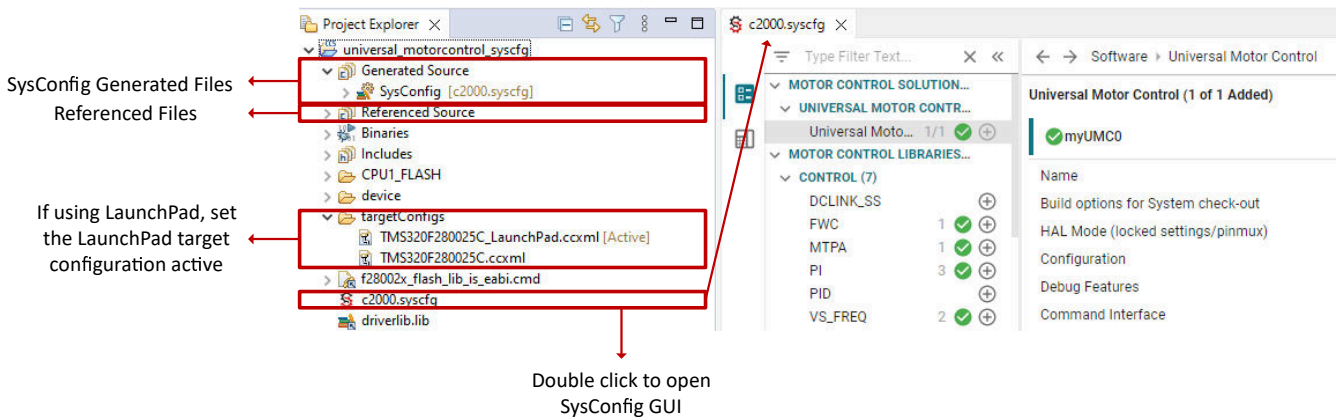


Figure 2-2. UMCL Project Structure

To run the project, click on the Debug button within CCS. To setup the watch expression variables, use the instructions outlined in the [Motor Control SDK Universal Project and Lab](#).

3 Main Settings

All of the general settings for the UMCL are located within the center pane of the Motor Control SysConfig tool. The first is the **build level**. The build levels progresses from one to four, each adding more functionality to the project. For a full explanation on what each of the build levels does, see the [Universal Motor Control Project and Lab User's Guide](#).

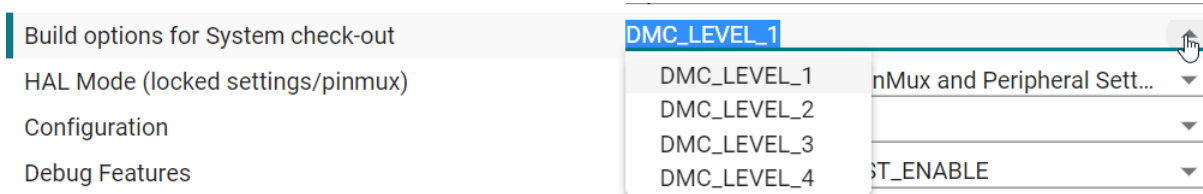


Figure 3-1. Build Levels

The next setting is for the **HAL Mode**. There are two settings for this, 'Demo' and 'Custom'. Within the 'Demo' view, all device resources are locked. This mode is meant to be used for evaluation purposes. All configurations in this mode have been tested. The 'Custom' mode un-locks all of the device resources. Within this mode, settings for the peripherals or device resources can be altered to meet specific hardware/application needs.

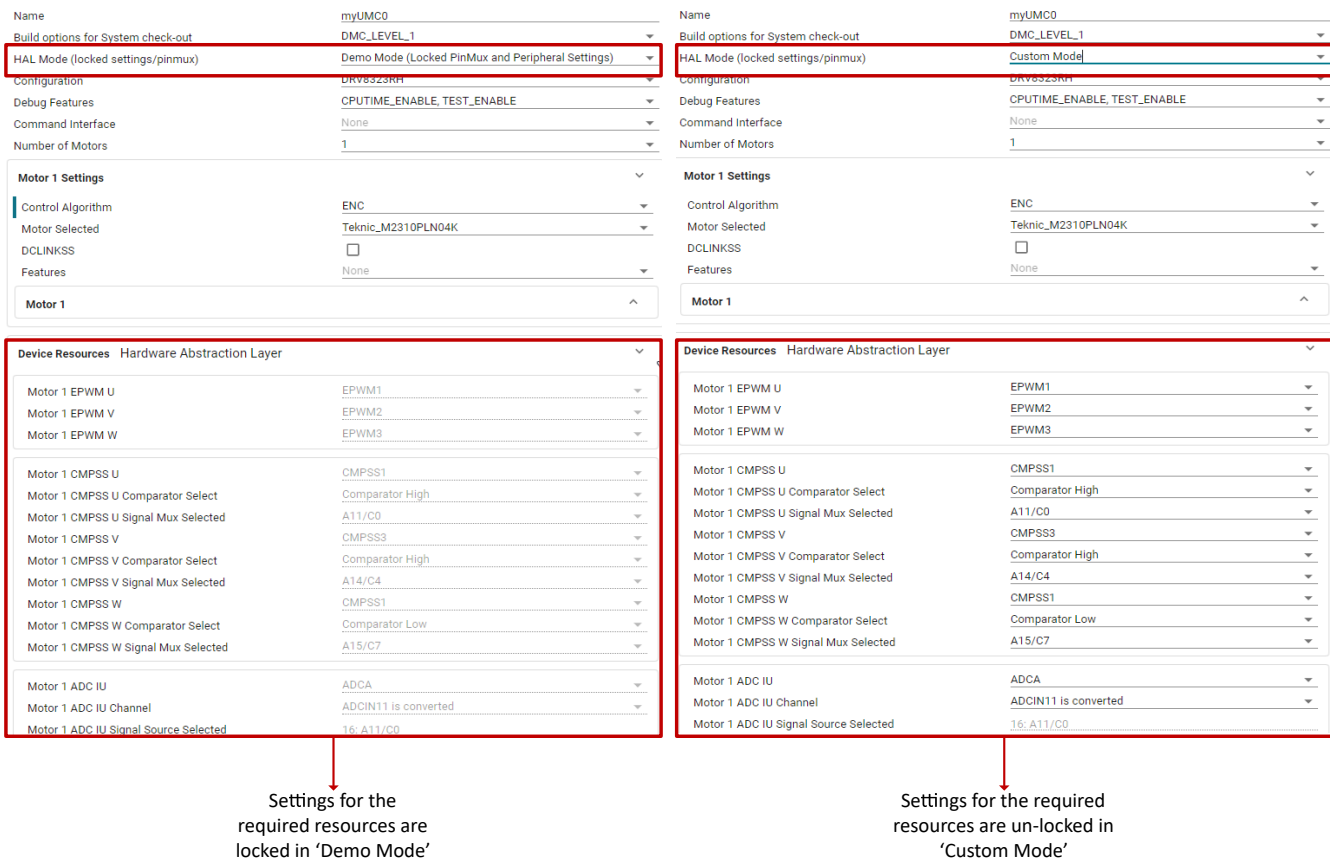


Figure 3-2. Demo vs. Custom Mode

There are multiple motor drive evaluation boards that can be used for the UMCL. The **configuration** option allows the selection of the evaluation boards. If a board is chosen that conflicts with other settings of the lab, errors/warnings are provided by the tool.

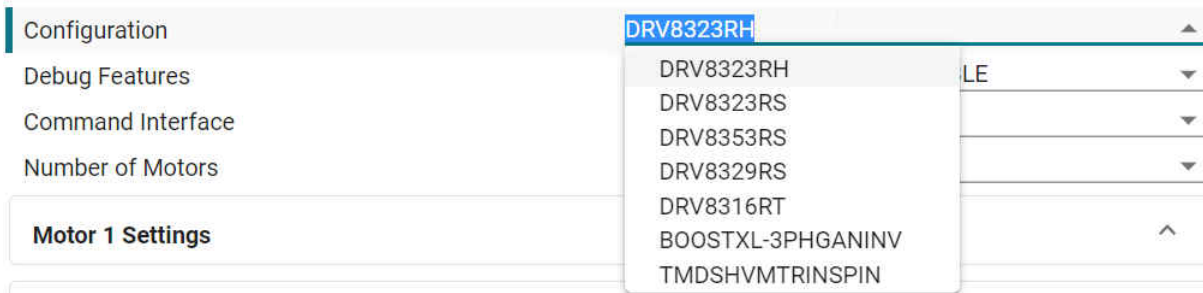


Figure 3-3. Motor Drive Evaluation Board Selection

Next are the **debug features**. These features, once checked, enable the check of different parameters and settings within the project as another means of debugging. Options that are not applicable based on the motor drive evaluation board become disabled. If required libraries or supporting files are needed when a debug feature is added, the tool automatically adds that into the project.

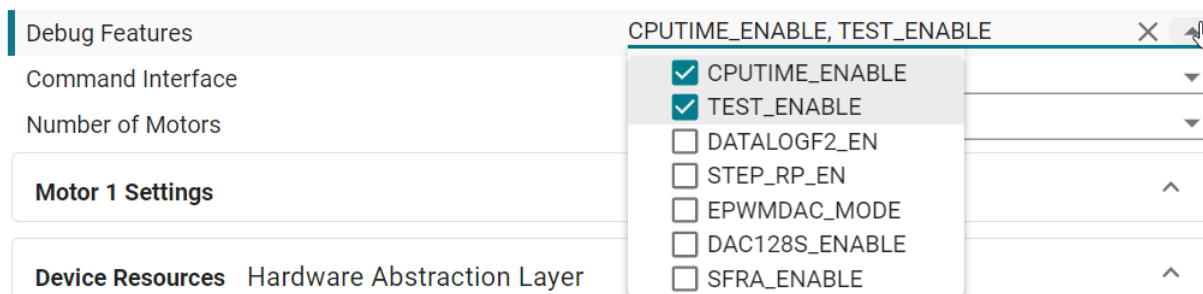


Figure 3-4. Debug Features

There are multiple **command interfaces** that can be used to change the reference speed of the motor or start/stop the motor. These options are listed within the command interface setting. Similar to the debug features, if a command interface is added and requires supporting libraries or resources, the tool automatically adds them.

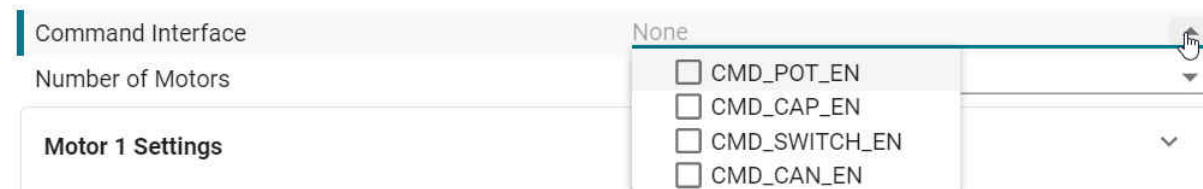


Figure 3-5. Command Interfaces

Lastly, the number of motors can also be selected to fit application needs. Note, the number of motors can be constricted due to device resources, especially on lower pin count devices.

4 Motor Settings

Right underneath the main settings for the UMCL are the motor specific settings. The first option is to choose a **control algorithm**. There are multiple control algorithms and some can be used in conjunction with others. If an option is not supported, the tool throws an error or warning describing the incorrect configuration.

Motor 1 Settings

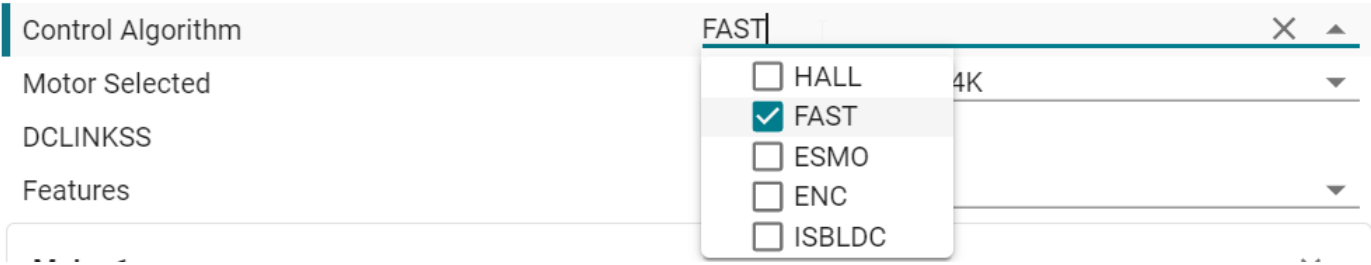


Figure 4-1. Control Algorithm

Next is the **selection of a motor**. There are only a few supported motors in the lab. The tool already contains default parameters for each of the motor options provided and locks the parameters in place. If a custom motor is desired, select the 'Custom' option. This option unlocks the motor parameters. The parameters can be changed to fit the criteria of the custom motor.



Figure 4-2. Motor Selection

There are multiple **motor features** supported. Each of these features enhances the performance of the motor control. To test the different features select the desired features.

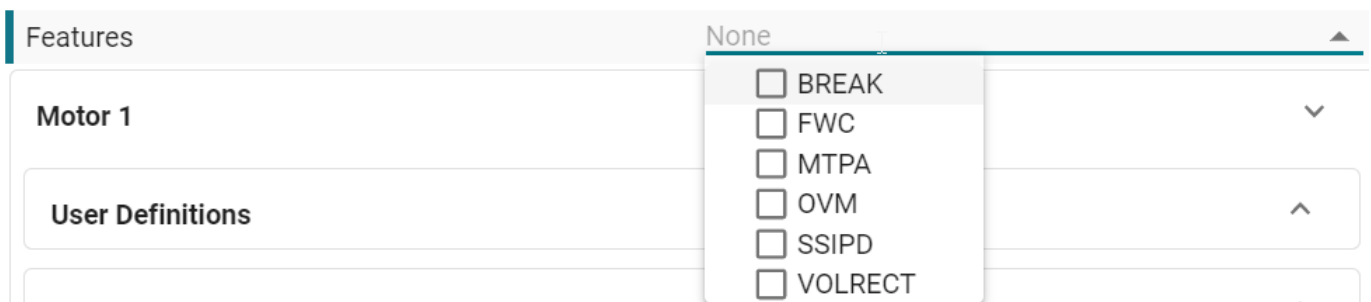


Figure 4-3. Motor Features

User Definitions are grouped together. These are settings that can be changed and control multiple parts of the control loop and functionality of the lab. Some of the grouped settings include 'Offset Parameters', 'Fault Protection Parameters', 'Timing Parameters', and 'Timing Parameters'.

Offset Parameters		^
Fault Protection Parameters		^
Timing Parameters		^
Motor Specs		v
Type	MOTOR_TYPE_PM	
Number of Pole Pairs	4	
Rr [Ohm]	0	
Rs [Ohm]	0.393955578	
Ls d [H]	0.000190442806	
Ls q [H]	0.000190442806	
Rated Flux [VpHz]	0.0399353318	
RES EST Current [A]	1.5	
IND EST Current [A]	-1	

Figure 4-4. User Definitions

Depending on the settings being inputted into the tool, the specific settings that require calculations are automatically calculated by the tool. This includes any conversions of units as well.

Motor 1		v
User Definitions		v
DC Nominal Bus Voltage [V]	48	
Number of Current Sensors	3	
Number of Voltage Sensors	3	
Voltage Filter Pole [Hz]	680.4839	
Voltage Filter Pole [RPS]	4275.6064	
Max Negative ID REF Current [A]	-2	
Direct Voltage Scale Factor	0.95	
Max Vs Mag [PU]	0.576	
Vs Ref Mag [PU]	0.528	
ADC Maximum Input Voltage [V]	3.3	
Voltage Sample Resistor [kohm]	4.99	
Voltage Divide Resistor 1 [kohm]	82	
Voltage Divide Resistor 2 [kohm]	0	
Voltage Divide Resistor 3 [kohm]	0	
ADC Full Scale Voltage [V]	57.5285	
Current Sample Resistor [kohm]	0.007	

Calculated Values Based on User Inputs

Figure 4-5. Calculated Values

The presented settings also changed based on the control algorithm and motor selected. This reduces the amount of configurable options to be only the ones that pertain to the specific setup of the application.

Speed Force [Hz]	30
Acceleration Start [Hzps]	10
Acceleration Max [Hzps]	20
Speed FS [Hz]	3
Encoder Position Max	3999
Encoder Position Offset	668
Rs Online Wait Time [sec]	300
Rs Online Work Time [sec]	120

Motor specifications hide/show
depending on the Control
Algorithm chosen

Figure 4-6. Tailored Settings

5 Device Resources

The **Device Resources** section contains all of the required device resources for the UMCL. Additionally, the panel on the left hand side can also show the resources used. Upon clicking any of the device resources, the configured parameters are shown. More device resources can be added by simply clicking the '+' icon next to the resource desired. This allows for an easy integration of further capabilities.

The screenshot displays the configuration interface for device resources. On the left, a list of resources is categorized into SYSTEM (18), ANALOG (4), and CONTROL (5). Each resource has a status indicator (checkmark and plus icon) and a count. On the right, the 'Motor 1' configuration is shown, including parameters like Control Algorithm (ENC), Motor Selected (Teknic_M2310PLN04K), and DCLINKSS. Below this, the 'Device Resources' view for Motor 1 is expanded, showing a Hardware Abstraction Layer and a list of other dependencies such as Motor 1 EPWM U, V, W, and various GPIO resources.

System resources utilized based upon user settings

Device Resources view shows allocated resources

Figure 5-1. Device Resources

6 Generated Code

One of the key benefits of using SysConfig is the auto-generated code. The right-hand side of the tool showcases all of the files SysConfig generates based on the settings chosen. Upon changing a setting within the tool, the files that are modified change icons to signify a modification made. This provides an easy way to identify which files are modified for certain functionality. If you desire not to use one or all of the generated files, toggle the 'Include in build' switch to remove the generated files from being included in the build.

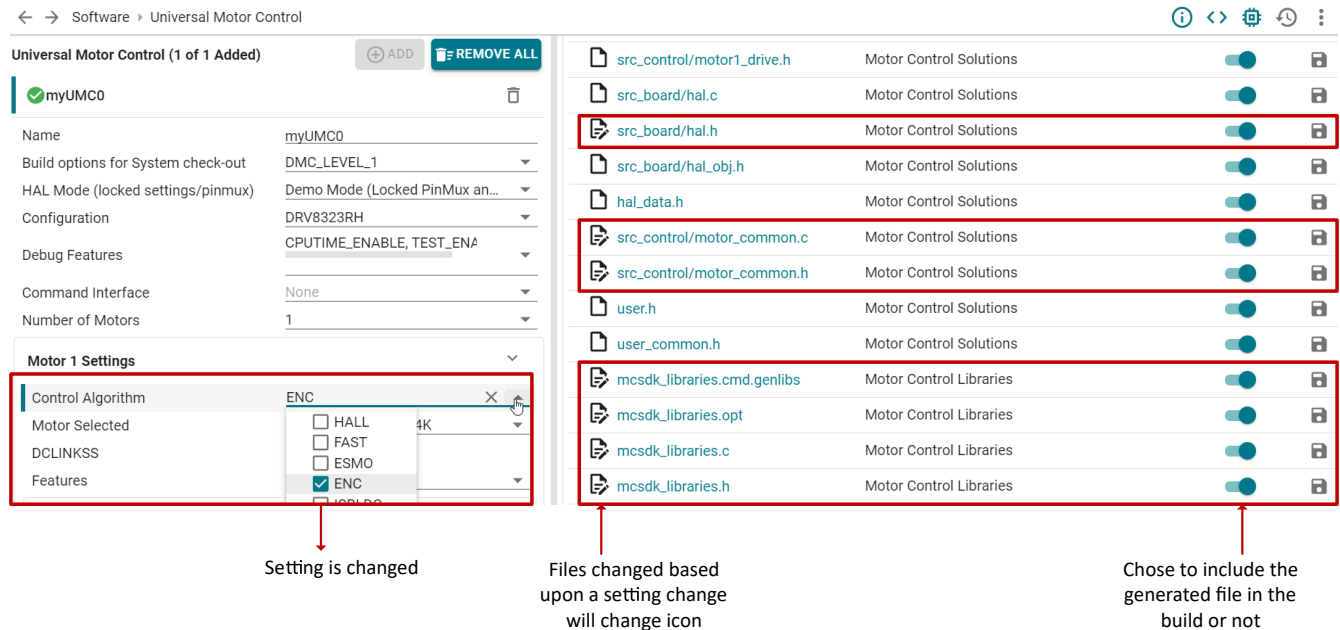


Figure 6-1. Generated Files

The ".genlibs" files generated by SysConfig includes all of the paths to any ".lib" files that the project requires. These files are automatically pulled in when the project is compiled to maintain proper linking and usage of the libraries. Below is an example entry located within the ".genlibs" file of the motor control sdk libraries file (mcsdk_libraries.cmd.genlibs).

```
-l"/libraries/observers/est_lib/lib/fast_full_lib.lib"
```

The ".opt" files includes all paths for any required header files. The content of these files is auto-generated based on the added motor control libraries. Below is an example entry located within the ".opt" file of the motor control sdk libraries file (mcsdk_libraries.opt).

```
-I"C:/ti/c2000/C2000Ware_MotorControl_SDK_4_01_00_00/libraries/control/pi/include"
```

Lastly, any reference files used are showcased within the 'Reference File' pane. These reference files are added to the project based on the settings selected. The image below shows the "fwc.c" file being added to the project when the "fwc" motor feature is added.

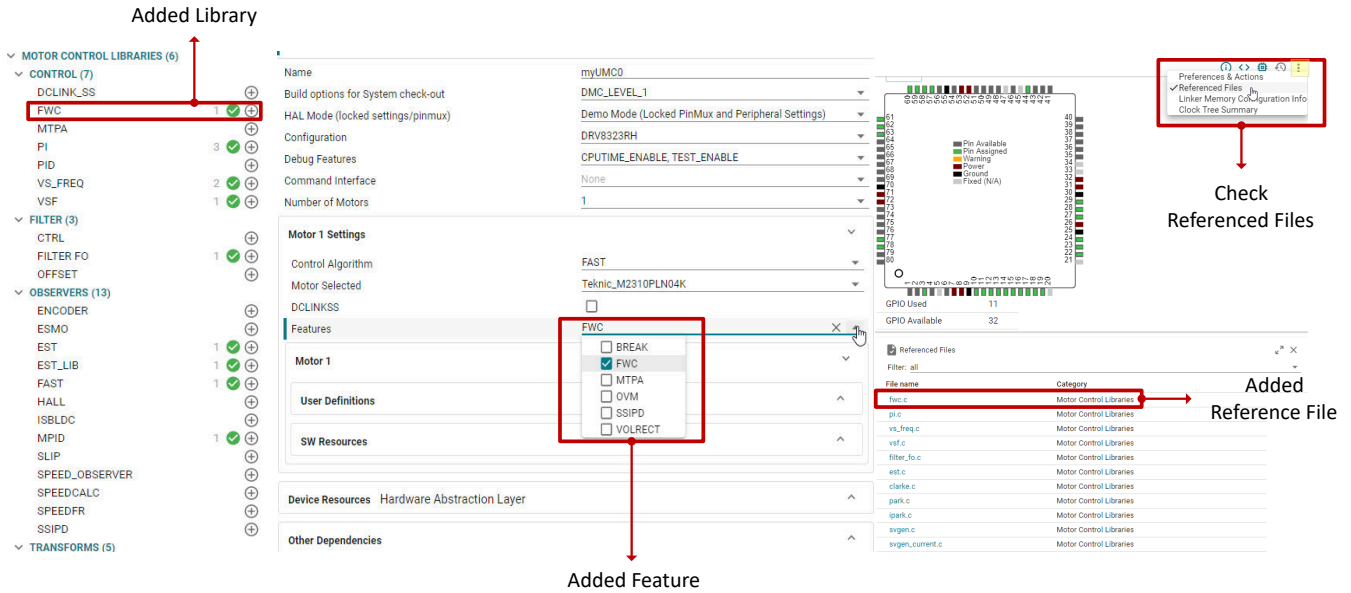


Figure 6-2. Reference Files

7 Summary

This document outlines the different components to the Motor Control SysConfig tool, in particular the UMCL. Through the use of SysConfig, the complexity of adding libraries, modifying parameters, and managing device resources is greatly decreased. Integration of this tool within a motor control project provides the same benefits listed.

8 References

- Texas Instruments: [Motor Control SDK Universal Project and Lab](#)
- Texas Instruments: [Speed Up Development With C2000 Real-Time MCUs Using SysConfig](#)
- Texas Instruments: [C2000 SysConfig](#)
- [SysConfig Video Series](#)
- [C2000WARE-MOTORCONTROL-SDK](#)
- [CCSTUDIO](#)

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2023, Texas Instruments Incorporated