



Application Note #2510

Programming multiple ECAM profiles on Accelera-series Controllers

Introduction:

Galil Motion Control offers many modes of motion. These different modes define the coordination and movement of one or more axes in different ways. By choosing a mode of motion that most easily defines the axis' movement, programming motion can be made easier and more understandable.

For instance, one mode of motion offered by Galil Motion Control is Electronic CAM, or ECAM. This mode of motion mimics the behavior of a mechanical cam by driving a slave axis, along a predetermined profile, based on the position of a master axis. ECAM is well suited for applications that require one axis to follow a repeatable, non linear profile, coordinated to the position of another. This motion would be much more difficult to program if done in the default "point to point" mode of motion, which allows the user to program the positions of individual axes to specific points.

Certain applications wish to take ECAM one step further and change the slave axis' profile between ECAM cycles. This is also something that is easily accommodated by Galil Motion Controllers by implementing a mode of motion called "Dual-ECAM." Prior to the Accelera series of Galil controllers, Dual-ECAM was implemented with the addition of special firmware. However, by taking advantage of the superior processing power and communication speed offered from the Accelera series controllers, the implementation of Dual-ECAM can be done easily with DMC code, and on a standard controller. This lessens the cost and complexity of an application requiring Dual-ECAM, by eliminating the need for custom firmware to provide this function.

Modes of Motion:

Dual-ECAM is not the only mode of motion offered by Galil. The default mode in which to program motion into a Galil controller is called "Point to Point." This mode allows the programmer to fully and individually control the position of each axis. Point to Point motion works by taking the commanded position of an axis and based on its current position, speed, acceleration and deceleration settings, generates a trapezoidal motion profile for the axis to follow in order to reach that command position. An offshoot of Point to Point motion is "Vector Mode." Vector mode coordinates the motion of 2 or 3 axes and develops a coordinated, trapezoidal motion profile, based on current and desired positions of the axes, which are defined in a 2 axis plane. The respective trapezoidal profiles for each axis are then built much in the same manner as in Point to Point motion, considering vector speeds and accelerations, but will command the motors to simultaneously complete their respective moves within the plane. Another mode of

motion offered by Galil is “Electronic Gearing.” Electronic gearing accommodates a linear position relationship between 2 or more axes. Mimicking the function of a mechanical gear system, this mode of motion drives a slave axis to a position proportional to the position of a master axis. Lastly if the coordinated relationship between the position of the master and slave axes is not linear, the position of the slave axis can be defined, based on the position of the master axis, with a mode of motion called “ECAM.” A slightly detailed description of ECAM can be seen below. However for complete and specific details as to how to use these different modes of motion, please see the controller’s users manual and command reference.
<http://www.galilmc.com/literature/manuals.html>

ECAM Description:

As mentioned above, ECAM coordinates a nonlinear relationship between a master and a slave axis. The master axis provides a reference position, which the slave axis will follow. This slave axis will reach predefined points, creating a profile, based on the location of the master. This is done by building a table of positions which define the motion profile that the slave will follow. As the master axis passes through user defined position intervals, the slave will simultaneously reach the respective slave positions assigned by the table. This behavior is illustrated below in Figure 1, where the slave axis’ position, graphed along the vertical axis, is driven to the locations in Table 1, as the master axis passes through each position interval, graphed along the horizontal axis.

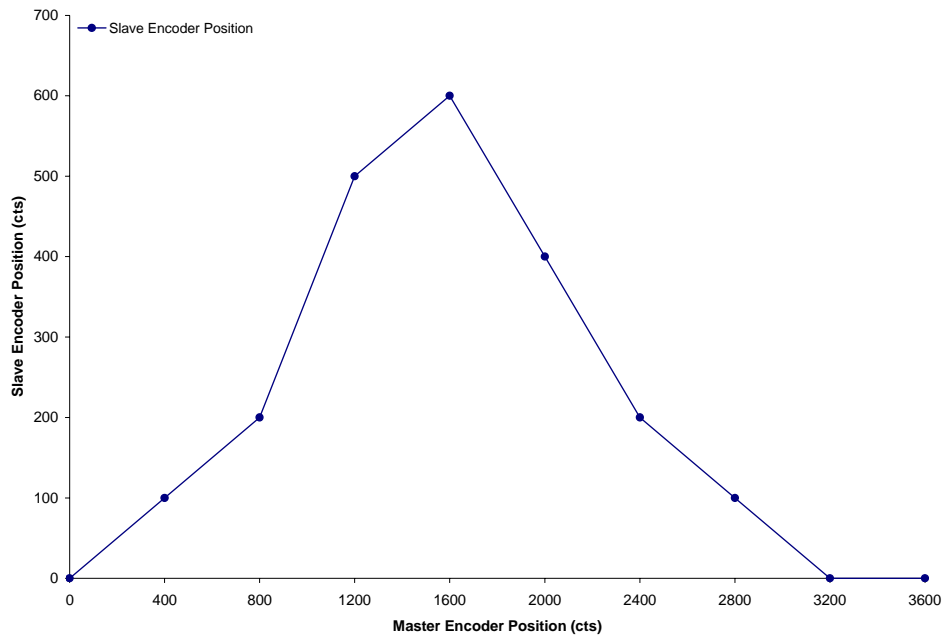


Figure 1: Graph of the ECAM slave position profile vs the master position

| ECAM Point # | Master Axis Position (cts) | Slave Position (cts) |
|--------------|----------------------------|----------------------|
| 1 | 0 | 0 |
| 2 | 400 | 100 |
| 3 | 800 | 200 |
| 4 | 1200 | 500 |
| 5 | 1600 | 600 |
| 6 | 2000 | 400 |
| 7 | 2400 | 200 |
| 8 | 2800 | 100 |
| 9 | 3200 | 0 |
| 10 | 3600 | 0 |

Table 1: Table of slave positions, the corresponding master positions and ECAM point numbers. This particular table corresponds to an ECAM set up that has a modulus of 3600 points, and a master interval of 200.

To program the coordination of 2 axes with Galil's ECAM mode of motion a minimum of 3 parameters have to be specified. These are: ECAM modulus, or the displacement of the master axes that corresponds to one ECAM cycle (EM), the ECAM interval, or displacement of the master axis between each of the slave positions (EP), which are listed in an ECAM table (ET), that dictate the slave profile. For a complete description of ECAM, including different programming parameters and sample code, please see app note 2441: <http://www.galilmc.com/support/appnotes/optima/note2441.pdf> .

The first step to programming in ECAM mode is to build the ECAM table that defines the profile of the slave axis. Next, the position increment of the master that will pass between each slave position must be defined. With the interval and number of ECAM table positions, the modulo can be calculated using simple multiplication. Please see figure 2 and 3 below.

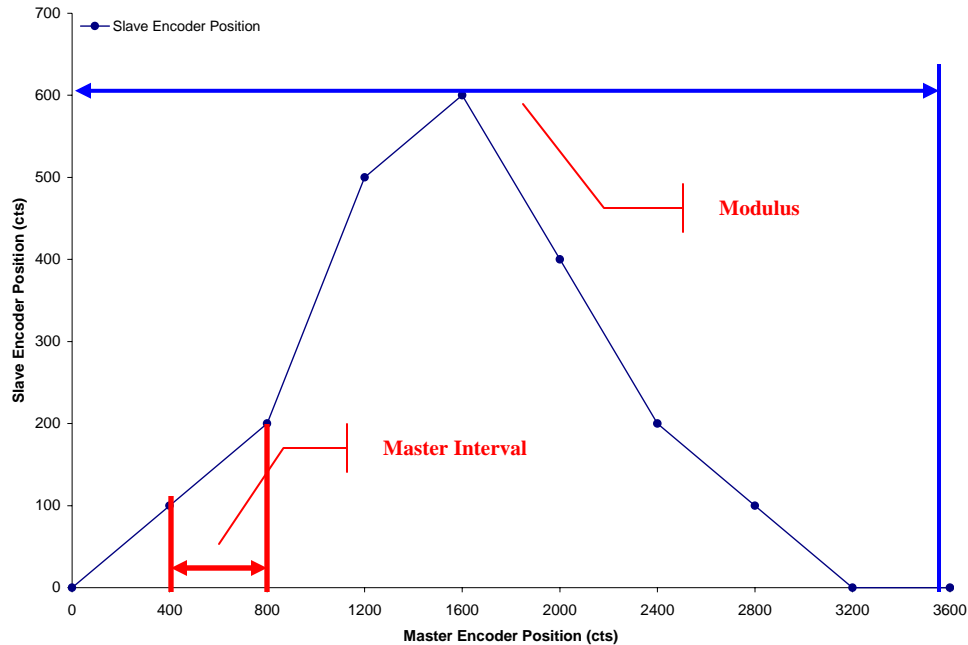


Figure 2: Detailed graph of ECAM values through one cycle of the master modulo.

$$EM = EP * ET_{dim}$$

Where:

EM = Master Modulo

EP = Master Interval

ET_{dim} = ECAM Table Dimension, or Number of ECAM Table Points

Figure 3: Basic equation to calculate master modulo.

One behavior of ECAM worth noting is that the master interval corresponds to absolute position increments. This means that regardless of master axis direction or speed, the slave axis will follow accordingly, essentially coordinating with any speed and direction of the master.

ECAM Application:

An application presented to Galil, that required the use of ECAM, was to provide a motion controller that was to be used in a metal stamping machine. This machine would stamp 2 distinct profiles into a metal strip. The raw metal strip was fed into the machine at a constant rate. As the metal was fed into the machine, a stamping die would then plunge in from above the metal strip to different distances forming the metal to the desired profile. Programming two profiles into one ECAM table would not be possible in this application due to the number of ECAM points necessary to achieve the proper profile resolution. To solve this issue it was decided to separate the two profiles into two

ECAM tables and then use Galil's Dual-ECAM function to switch back and forth between them without stopping the machine. Figure 4 illustrates the profile that will result from the use of Dual-ECAM functionality, following the positions listed in tables 2 and 3.

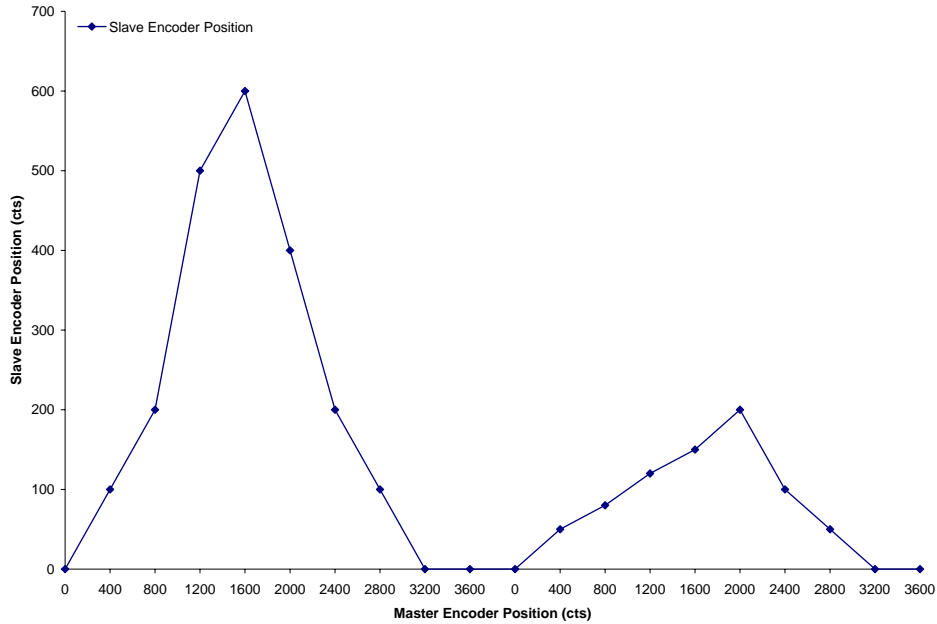


Figure 4: Multiple (dual) ECAM profile using two ECAM tables on the same set of axes. Note the cyclic rollover of the master encoder axis at the modulo of 3600 encoder counts. This illustrates 2 ECAM profiles being applied to the same master/slave axis pair.

| ECAM Point # | Master Axis Position (cts) | Slave Position (cts) |
|--------------|----------------------------|----------------------|
| 1 | 0 | 0 |
| 2 | 400 | 100 |
| 3 | 800 | 200 |
| 4 | 1200 | 500 |
| 5 | 1600 | 600 |
| 6 | 2000 | 400 |
| 7 | 2400 | 200 |
| 8 | 2800 | 100 |
| 9 | 3200 | 0 |
| 10 | 3600 | 0 |

Table 2: First table of slave positions and the corresponding master positions and ECAM point numbers. This particular table corresponds to an ECAM set up that has a modulus of 3600 points, and a master interval of 200.

| ECAM Point # | Master Axis Position (cts) | Slave Position (cts) |
|--------------|----------------------------|----------------------|
| 1 | 0 | 0 |
| 2 | 400 | 50 |
| 3 | 800 | 80 |
| 4 | 1200 | 120 |
| 5 | 1600 | 150 |
| 6 | 2000 | 200 |
| 7 | 2400 | 100 |
| 8 | 2800 | 50 |
| 9 | 3200 | 0 |
| 10 | 3600 | 0 |

Table 3: Second table of slave positions and the corresponding master positions and ECAM point numbers. The modulo and master interval is the same for this table as the first.

There are additional benefits to switching between two separate ECAM tables beyond expanding the ability to profile with a slave axis. For instance, during the execution of one table, the other is residing in memory which allows the user to edit points without disengaging the ECAM mode. In standard ECAM mode it would be necessary to disengage the ECAM mode prior to adjusting these points. This turned out to be a very valuable feature for the metal stamping application. As the metal changed batches, the “spring back” of the material changed. This required the profile(s) to be changed per metal batch. By using this feature the profiles could be changed without stalling production. Also, if the slave profile(s) dictate(s) it, the modulus can be different between the two tables, and changed on the fly as well. Again, ECAM would have to be disengaged to adjust this setting in standard ECAM mode.

Dual-ECAM Example:

As stated above, on previous generations of Galil controllers, Dual-ECAM functionality was applied with the addition of special firmware. This added cost and complexity to the application. However, by taking advantage of the 100BASE-T communication speed, expanded memory and superior processing power offered from Galil’s Accelera controllers, Dual-ECAM can be applied with DMC code. A generic Dual-ECAM application program can be seen below. The code has been shortened and annotated to include within this text, but can be seen in its entirety in Appendix 1.

The program begins with the clearing of any existing variables and arrays and initializes specific variables necessary for the Dual-ECAM code. V1 is used as a termination variable, which upon changing its value, will execute a termination loop to stop the program in a predefined manner. This is necessary to allow for any command to start properly upon subsequent execution. The variables Master1/2 and Slave1/2 are set to an initial value to initialize the system. But will be the variables used to set the controller modulus, and adjust the master interval accordingly, on the fly.

```

#CAM
DA*,*[];           'DEALLOCATE ARRAYS AND VARIABLES
MG"V1=1 TO DISABLE"; 'TERMINATION VARIABLE INSTRUCTION
V1=0;             'TERMINATION VARIABLE
Master1=4096;     'MASTER ENCODER MODULO TABLE1
Slave1=8000;     'SLAVE MODULO TABLE1
Master2=4096;     'MASTER ENCODER MODULO TABLE2
Slave2=8000;     'SLAVE MODULO TABLE2

```

The following code defines two arrays Etbl1 and Etbl2 and stores an arbitrary slave profile.

```

#ARRAY;
DMEtbl1[10],Etbl2[10]; 'DIMENSIONING TABLES
Etbl1[0]=0;           'BUILDING TABLE FOR INITIAL MOVEMENT
Etbl1[1]=100
Etbl1[2]=200
Etbl1[3]=500
Etbl1[4]=600
Etbl1[5]=500
Etbl1[6]=200
Etbl1[7]=100
Etbl1[8]=0
Etbl1[9]=0

Etbl2[0]=0
Etbl2[1]=50
Etbl2[2]=80
Etbl2[3]=120
Etbl2[4]=150
Etbl2[5]=200
Etbl2[6]=100
Etbl2[7]=50
Etbl2[8]=0
Etbl2[9]=0

```

For this example, the Y axis will function as the master, which will be geared off of a dummy axis Z with a ratio of 1. The Z axis is used to mimic a constant feedback for the master to follow. In an actual system, the Z feedback could be a production line, motor velocity or anything that the ECAM needs to be coordinated with. In an actual application the master axis, or Y axis in this case, can be an axis under Galil control or can simply be feedback from an external source.

```

DP0,0,0;           'SET POSITIONS TO ZERO
SHXYZ;           'CLEAR POSITION ERROR
GAY=CZ;           'USED TO KEEP MASTER UNPROFILED
GRY=1;           'GEAR RATIO FOR UNPROFILED MASTER AXIS
JGZ=10000;BGZ;    'JOG DUMMY AXIS TO DRIVE MASTER ENCODER
WT10;           'SMALL DELAY TO ALLOW MASTER MODULO TO BE NON ZERO

```

After the ECAM is initialized, the program loads an array to be used as reference for the ECAM table values. At this point in the code execution, timing becomes critical. Because of the high processing speed of the Accelera processor, the code needs to be slowed in

order to minimize the possibility of race conditions when switching between the two tables. For instance the #w0 loop is used to insure that the ECAM has been engaged before starting the execution of the ECAM table. The #w1 loop is used to again hold code execution of the next ECAM table until the previous cycle has been completed. This is done by stalling the code until half of the previous ECAM table has been executed. After which an ECAM disengagement point (90% of the ECAM table) is set and the code waits for disengagement before proceeding.

```
#CA
EAY;                               'Y AXIS ECAM MASTER
EM Slave1,Master1;                 'SET ECAM MODULOS FOR TABLE 1
EP Master1/10,0;                   'SET ECAM TABLE DELTA
EB1;                               'START ECAM MODE
#L1;                               'LOAD ARRAY1 INTO ECAM TABLE
ET[0]=Etbl1[0]
ET[1]=Etbl1[1]
ET[2]=Etbl1[2]
ET[3]=Etbl1[3]
ET[4]=Etbl1[4]
ET[5]=Etbl1[5]
ET[6]=Etbl1[6]
ET[7]=Etbl1[7]
ET[8]=Etbl1[8]
ET[9]=Etbl1[9]
EG0;                               'ECAM GO
#w0;JP#w0,(_EGX=0);               'WAIT FOR ECAM TO BE ENGAGED
#w1;JP#w1,(_TPY<(Master1*.5));    'SAME AS MF TRIPPOINT
EQ(Master1*.9);                   'DISENGAGE ECAM AT % OF MASTER
#w2;JP#w2,(_EGX=1);               'WAIT FOR ECAM TO DISENGAGE
```

After the ECAM table has been disengaged, the modulus for the proceeding table must be entered. These new modulus could be different as set per the initial programming, or changed by the user during the execution of the previous table. Next, the following ECAM table is loaded and executed in the same manner as above (omitted here but can be seen in full in the appendix).

```
EM Slave2,Master2;                 'SET ECAM MODULOS FOR TABLE2
EB0
EP Master2/10,0;                   'SET ECAM TABLE DELTA
EB1
```

This is the termination loop. If the code reaches this label and the operator does not change the value of the termination variable, V1, to a non-zero value, the code will loop infinitely. If the termination variable *is* set to a non-zero value the current ECAM execution is concluded, the ECAM mode is disengaged, and the dummy axis is stopped (the dummy axis is specific to this code, and not essential for ECAM function). This is necessary to avoid code execution conflict upon the issuing of any subsequent commands.

```
JP#L1,V1=0;                       'LOOP UNLESS OPERATOR SETS V1
EB0;                               'STOP ECAM MODE
STYZ;                              'STOP DUMMY AXIS
EN
```

Included in the ECAM table used in this example are a series of ECAM table entries of 0. This is to provide a dwell in the ECAM function while the new ECAM table is loaded. This dwell is necessary because the processor is occupied with the loading of the new table and would stall ECAM profiling at the last point reached. By including this dwell period the axis can be returned to a predicted and controlled location while stalled during the loading of the new table. The necessary dwell time depends on table size and master encoder velocity. This will vary from application to application. For example, running this code as a single thread on a DMC-4020, a 10-point ECAM table required just 1.95ms to load. With a 100-point ECAM table only 4.9ms was necessary. For the maximum size of 257 ECAM points, only 12.5ms was needed.

Additional Features to Dual-ECAM:

Using this code, changing a point in the ECAM table “on the fly” is as easy as sending a new variable to the reference table. The profile that is generated by the code above can be seen in Figure 4 below.

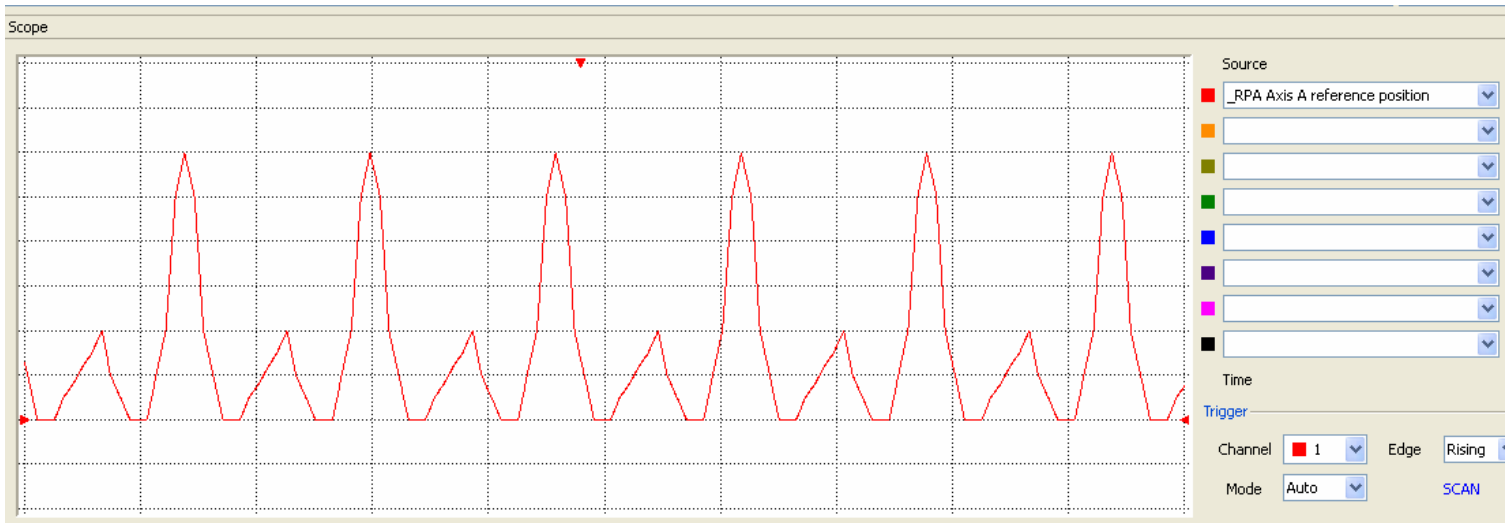


Figure 4: Plot of the ECAM slave profile generated by the code provided in above example.

If it is desired to change the motion profile of the slave axis on the fly, a point within the ECAM table will have to be edited, (although more than 1 can be edited at a time, only 1 will be provided for example). This is done by simply assigning a new variable in the reference array via the command window. During the next cycle, the Dual-ECAM program will update the array, including the edited ECAM point, thus editing the ECAM profile. This will all occur without the need for stopping the motion and the code. Figure 5 below is a plot of the ECAM profile as a point is changed on the fly. It can be noted that the motion has not been stopped, or paused in any way and the only commands issued were a new ECAM reference table point (upper right command window).

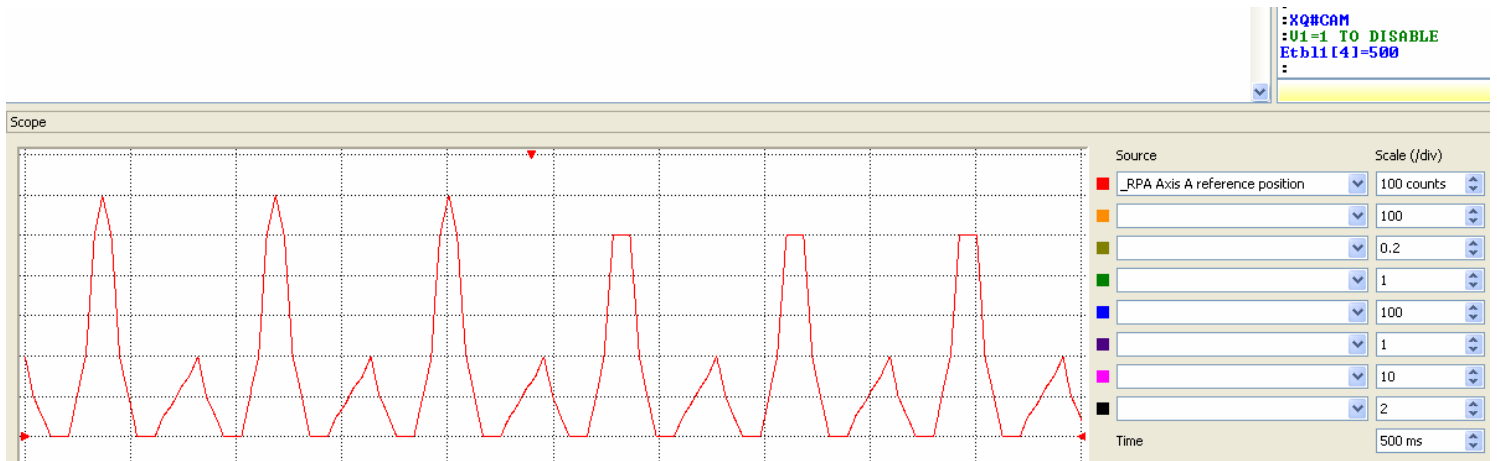


Figure 5: Plot of the ECAM slave profile generated by the code provided in the above example, as the ECAM table point, Etb11[4], is edited on the fly.

In much of the same manner the ECAM modulo can be changed on the fly using this code. Seen below, in Figure 6, by assigning a different value to the modulo variable (Master1/2, Slave1/2) the modulo can be changed on the fly upon the following execution of the ECAM. With the provided code, the master interval will also be updated automatically.

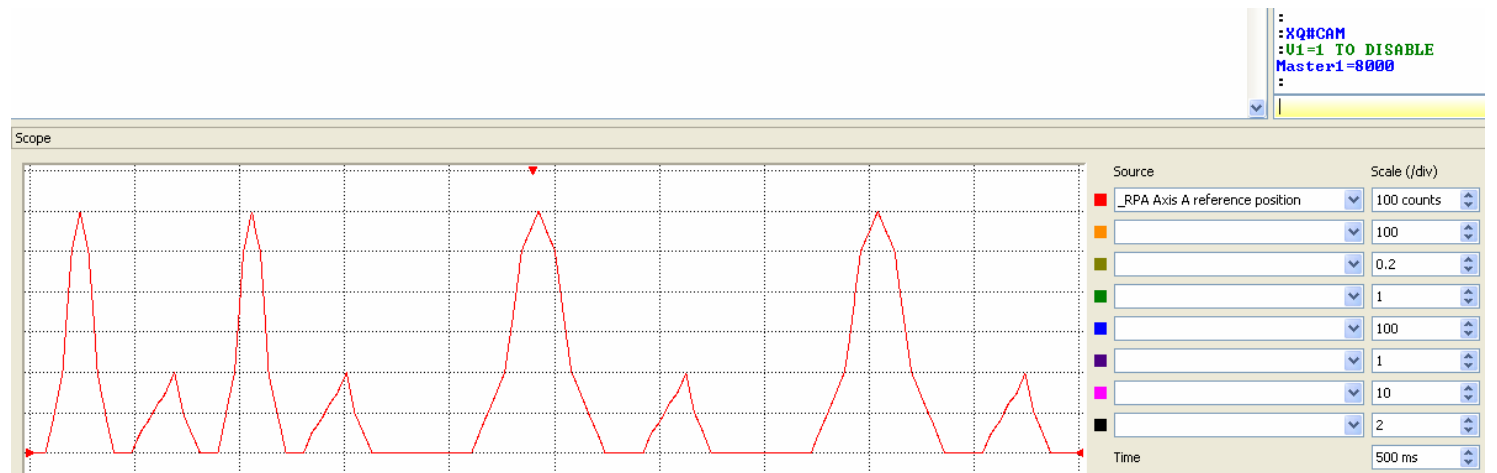


Figure 5: Plot of the ECAM slave profile generated by the code provided in the above example, as the modulus of table 1, Master1, is edited on the fly.

Conclusion:

The processing power and the communication speed of the 100Base-T Ethernet connection on the DMC-40x0 Accelera series controllers allows for complex multi-profile ECAM motion without the need for custom firmware. The example program illustrates this function and could be edited to apply to many different applications. For additional information please contact Galil Motion Control. An additional application example that has benefited with the use of Dual-ECAM can be found in Appendix II.

Appendix

I: Complete Dual-ECAM DMC Code

```
#CAM
DA*,*[];           'DEALLOCATE ARRAYS AND VARIABLES
MG"V1=1 TO DISABLE"; 'TERMINATION VARIABLE INSTRUCTION
V1=0;             'ENABLE TERMINATION VARIABLE
Master=4096;      'MASTER ENCODER MODULO
Slave=8000;       'SLAVE MODULO
#ARRAY;
DMEtbl1[10],Etbl2[10]; 'DIMENSIONING TABLES
Etbl1[0]=0;       'BUILDING TABLE FOR INITIAL MOVEMENT
Etbl1[1]=100
Etbl1[2]=200
Etbl1[3]=500
Etbl1[4]=600
Etbl1[5]=500
Etbl1[6]=200
Etbl1[7]=100
Etbl1[8]=0
Etbl1[9]=0

Etbl2[0]=0
Etbl2[1]=50
Etbl2[2]=80
Etbl2[3]=120
Etbl2[4]=150
Etbl2[5]=200
Etbl2[6]=100
Etbl2[7]=50
Etbl2[8]=0
Etbl2[9]=0

DP0,0,0;         'SET POSITIONS TO ZERO
SHXYZ;          'CLEAR POSITION ERROR
GAY=CZ;         'USED TO KEEP MASTER UNPROFILED
GRY=1;         'GEAR RATIO FOR UNPROFILED MASTER AXIS
JGZ=1000;BGZ;   'JOG DUMMY AXIS TO DRIVE MASTER ENCODER
WT10;          'SMALL DELAY TO ALLOW MASTER MODULO TO BE NON
ZERO
#CA
EAY;           'Y AXIS ECAM MASTER
EM Slave,Master; 'SET ECAM MODULOS
EP Master/10,0; 'SET ECAM TABLE DELTA
EB1;         'START ECAM MODE
#L1;        'LOAD ARRAY1 INTO ECAM TABLE
ET[0]=Etbl1[0]
ET[1]=Etbl1[1]
ET[2]=Etbl1[2]
ET[3]=Etbl1[3]
ET[4]=Etbl1[4]
ET[5]=Etbl1[5]
ET[6]=Etbl1[6]
ET[7]=Etbl1[7]
ET[8]=Etbl1[8]
```

Appendix

```
ET[9]=Etbl1[9]
EG0;                                'ECAM GO
#W0;JP#W0,(_EGX=0);                'WAIT FOR ECAM TO BE ENGAGED
#W1;JP#W1,(_TPY<(Master*.5));      'SAME AS MF TRIPPOINT
EQ(Master*.9);                      'DISENGAGE ECAM AT % OF MASTER
#W2;JP#W2,(_EGX=1);                'WAIT FOR ECAM TO DISENGAGE

#L2;                                'LOAD ARRAY2 INTO ECAM TABLE
ET[0]=Etbl2[0]
ET[1]=Etbl2[1]
ET[2]=Etbl2[2]
ET[3]=Etbl2[3]
ET[4]=Etbl2[4]
ET[5]=Etbl2[5]
ET[6]=Etbl2[6]
ET[7]=Etbl2[7]
ET[8]=Etbl2[8]
ET[9]=Etbl2[9]
EG0
#W3;JP#W3,(_EGX=0)
#W4;JP#W4,(_TPY<(Master*.5))
EQ(Master*.9)
#W5;JP#W5,(_EGX=1)

JP#L1,V1=0;                          'LOOP UNLESS OPERATOR SETS V1
EB0;                                  'STOP ECAM MODE
STYZ;                                 'STOP DUMMY AXIS
```

Appendix

II: Foodservice nozzle application:

One application that utilized the dual-ECAM mode of motion came from the foodservice industry. A machine was designed to control a fluid stream from a nozzle to apply topping to food products. The food products pass underneath the nozzle on a conveyer. The controller varies the fluid flow by adjusting a servo valve. The ECAM master data is an encoder on the conveyer motor. Not only does the output of the nozzle need to vary based on conveyer speed, but upstream circuits detect and calculate the physical length of the food product. This variability requires that the ECAM modulo be flexible in some way.

By taking advantage of the capabilities of dual-ECAM, there is no need to halt the process to re-arm the ECAM motion with a new modulo value. This adjustable modulo capability increased throughput of the machine dramatically.