



Application Note #2441

Generating ECAM Tables with any Modulus (EM)

This application note describes techniques to generate cam tables with any master period (modulus). For example, one cam table may be a sine wave that repeats every 1000 counts of the master, and a second cam table may be a sine wave that repeats every 1001 counts of the master. Generating the second (1001 count) table is not straight-forward because the cam points must be evenly spaced and the space between each point must be an integer number of counts.

CONTENTS

1	REVIEW OF ECAM COMMANDS.....	2
2	CHOOSING EP AND THE NUMBER OF ET POINTS GIVEN ANY EM.....	3
3	WHEN EM IS NOT DIVISIBLE BY EP (THE ADDITIONAL POINT METHOD)....	5
4	CAM TABLE GENERATOR EXAMPLE.....	6

1 Review of ECAM Commands

This section reviews how to generate a simple cam table (figure 1). A cam table is a graph of slave position versus master position. The graph is specified at certain points (master positions 0, 2000, and 4000 in figure 1) and linear interpolation between those points is used to define the rest of the table.

Although we define the table on a certain window of master positions (0 to 4000 in figure 1), the relationship holds for all master positions (the table looks the same from master positions 4000 to 8000, 8000 to 12000 etc.). This window is called the modulus.

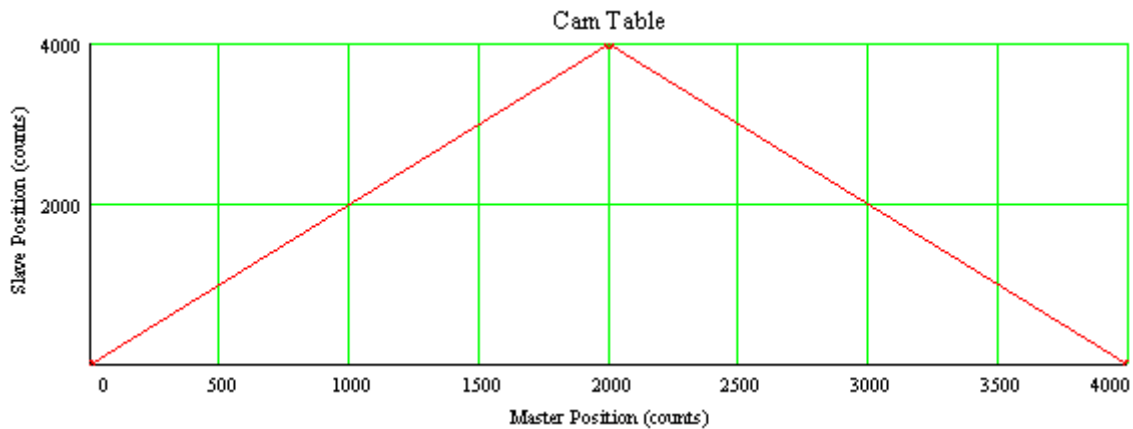


Figure 1. Simple three-point ECAM table

Table 1 reviews the commands necessary to run a cam table.

Table 1. Important ECAM Commands

Command	Description
EA	Selects master axis
EB	Starts camming
EG	Engage slave axis
EM	Master modulus
EP	Master interval
ET	Table of slave positions

The cam table itself is defined by three commands: EM, EP, and ET. EM specifies the master modulus (4000 in figure 1), EP specifies the distance between each point (2000 in figure 1), and ET specifies the slave position at each point. The ratio of EM to EP specifies how many ET points are necessary. The number of intervals N is $EM / EP = 4000 / 2000 = 2$ and the number of points P (3 in figure 1) is one more than the number of intervals:

$$P = N + 1 \quad (1)$$

where P is the number of points and N is the number of intervals.

The code below implements the cam table in figure 1:

```

EB0          ;'turn cam off
EAY         ;'y axis is the master
EM 0,4000   ;'slave change, master change
EP 2000,0   ;'interval width, starting position

ET[0]=0     ;'define cam table (3 points, 2 intervals)
ET[1]=4000
ET[2]=0

EB1         ;'turn cam on
EG-100     ;'engage slave immediately
JGY=1000   ;'jog the master
BGY        ;'begin motion
EN         ;'end program

```

2 Choosing EP and the Number of ET points given any EM

The maximum number of intervals N_{\max} on a standard controller is 256 (this can be expanded with special firmware). Given this restraint and a desired modulus EM in counts, the question arises: what interval EP and number of intervals N should be chosen so that EP is minimum? The answer is below. Given EM and N_{\max} :

$$EP = \text{ceiling}\left(\frac{EM}{N_{\max}}\right) \quad (2)$$

$$N = \text{ceiling}\left(\frac{EM}{EP}\right) \quad (3)$$

where: N_{\max} is the maximum number of intervals (256 for a standard controller)

EM is the modulus in counts

EP is the distance in counts between each point (the interval)

N is the actual number of intervals used (N is less than or equal to N_{\max})

ceiling() gives the next greatest integer if the argument is non-integer or just returns the argument if it is an integer

The code below shows how to compute the ceiling in the Galil language:

```

IF(@FRAC[input] = 0)          ;'input is integer
  output = input              ;'leave number unchanged
ELSE                           ;'input is decimal
  output = @INT[input] + 1 ;'truncate and add one
ENDIF

```

Figure 2 shows the graphical relationship of EP versus EM and N versus EM for $N_{\max}=256$. For illustrative purposes, figure 3 shows the same relationship if the maximum number of intervals were only 10 ($N_{\max} = 10$). This allows individual points to be seen on the graph.

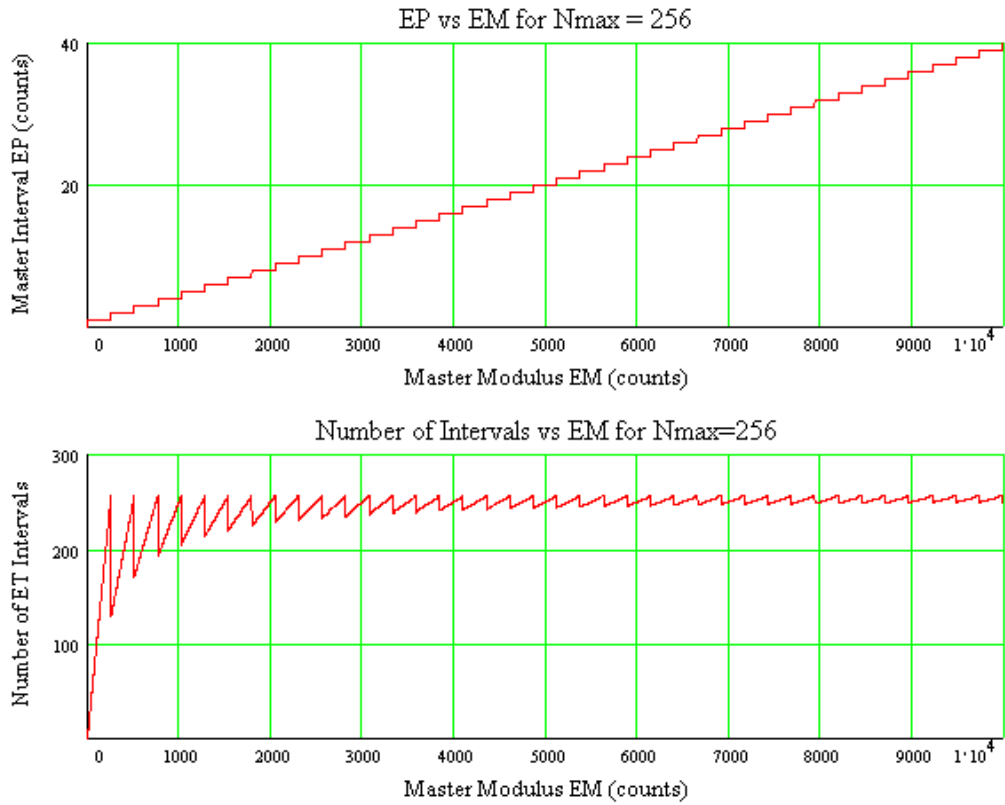


Figure 2. Optimal master interval (EP) and number of intervals for modulus (EM) values 0 to 10000

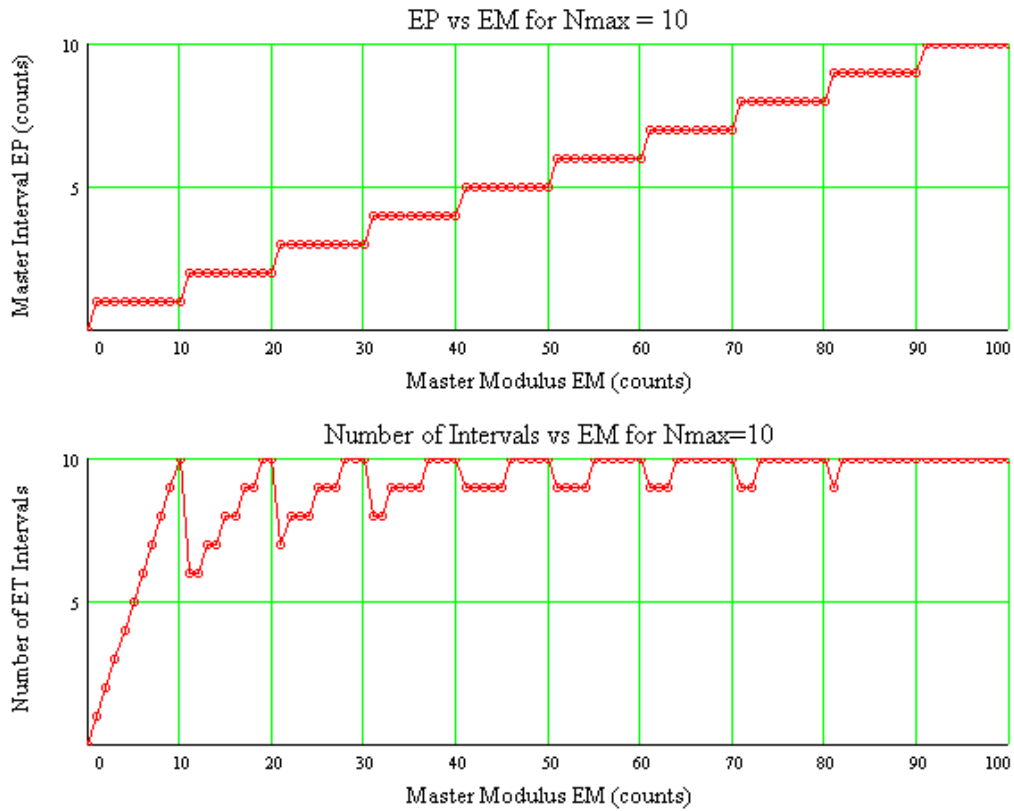


Figure 3. The maximum number of intervals is reduced from 256 to 10 for illustrative purposes.

3 When EM is not Divisible by EP (the Additional Point Method)

For some EM values, it is not possible to find an EP value that divides evenly into EM. For example, if we need EM = 257, equations 2 and 3 give EP = 2 and N = 129. In this case, EP · N = 258 and this is as close to 257 as we can get. These special cases are identified by the following equation (where EP and N are computed via equations 2 and 3):

$$EP \cdot N \neq EM \quad (4)$$

For these cases, we must define an additional point past the modulus such that the line defined by the last two points gives the desired master/slave relationship between the second-to-last point and the modulus. This will be illustrated by an example. Suppose we wish to generate the cam table in figure 4 with only three points.

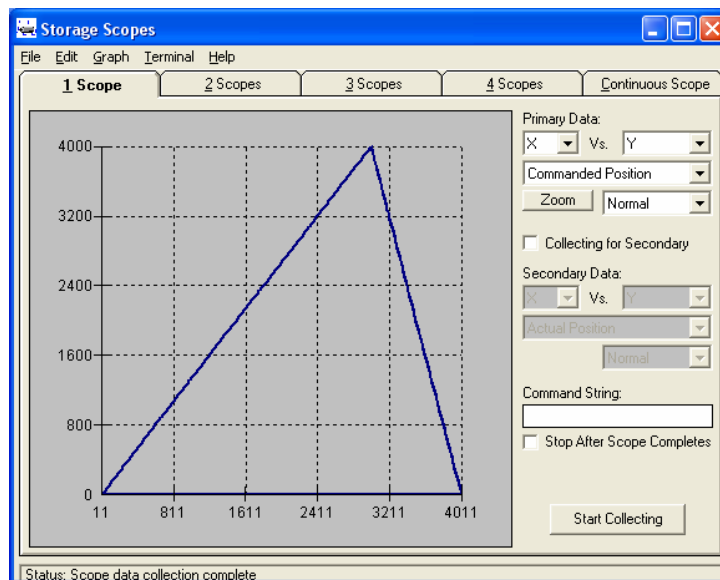


Figure 4. Cam table for which EM (4000) is not divisible by EP (3000)

We choose a modulus EM of 4000 and an interval EP of 3000 (note that this EP does not follow equations 2 and 3 and so is not optimal). To generate the cam in figure 4, we must give points to ET as in figure 5:

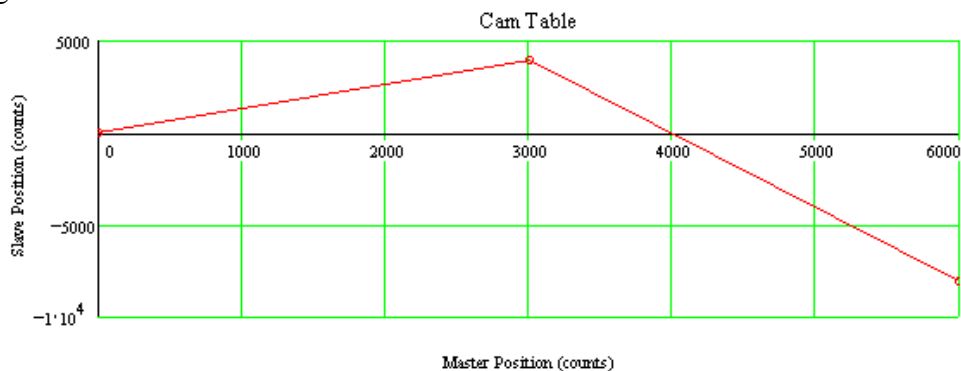


Figure 5. The full three-point cam table that must be given to ET

Since the interval EP is 3000, we know that the third point must be at $x = 6000$. To find it's y position, we must know the equation of a line given two points (x_1, y_1) and (x_2, y_2) :

$$y = mx + b \quad \text{where: } m = \frac{y_2 - y_1}{x_2 - x_1} \text{ and } b = y_1 - mx_1$$

Given that $(x_1, y_1) = (3000, 4000)$ and $(x_2, y_2) = (4000, 0)$, the y position of the last point is as follows:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{0 - 4000}{4000 - 3000} = -4$$

$$b = y_1 - mx_1 = 4000 - (-4)3000 = 16000$$

$$y = mx + b = (-4)6000 + 16000 = -8000$$

The code below implements the cam table in figure 4:

```

EB0           ; 'turn cam off
EAY           ; 'y axis is the master
EM 0,4000    ; 'slave change, master change
EP 3000,0    ; 'interval width, starting position

ET[0]=0      ; 'define cam table (3 points, 2 intervals)
ET[1]=4000
ET[2]=-8000

EB1           ; 'turn cam on
EG-100       ; 'engage slave immediately
JGY=1000     ; 'jog the master
BGY          ; 'begin motion
EN           ; 'end program

```

4 Cam Table Generator Example

This section provides a working example of a cam table generator that creates a sin wave of any period (modulus EM) that uses the techniques from sections 2 and 3. Again, for illustrative purposes, we restrict the maximum number of intervals N_{\max} to 10. Figure 6 shows the cam table generated when EM is 100 and figure 7 shows the results for EM = 101. For both figures, the interval EP and the number of intervals N are chosen via equations 2 and 3 (with $N_{\max} = 10$).

Note how the “additional point method” from section 3 is used in figure 7 since $EP \cdot N = 11 \cdot 10 = 110$ is not equal to the modulus $EM = 101$.

The code below generates figure 7. To generate figure 6, change the variable *period* to 100.

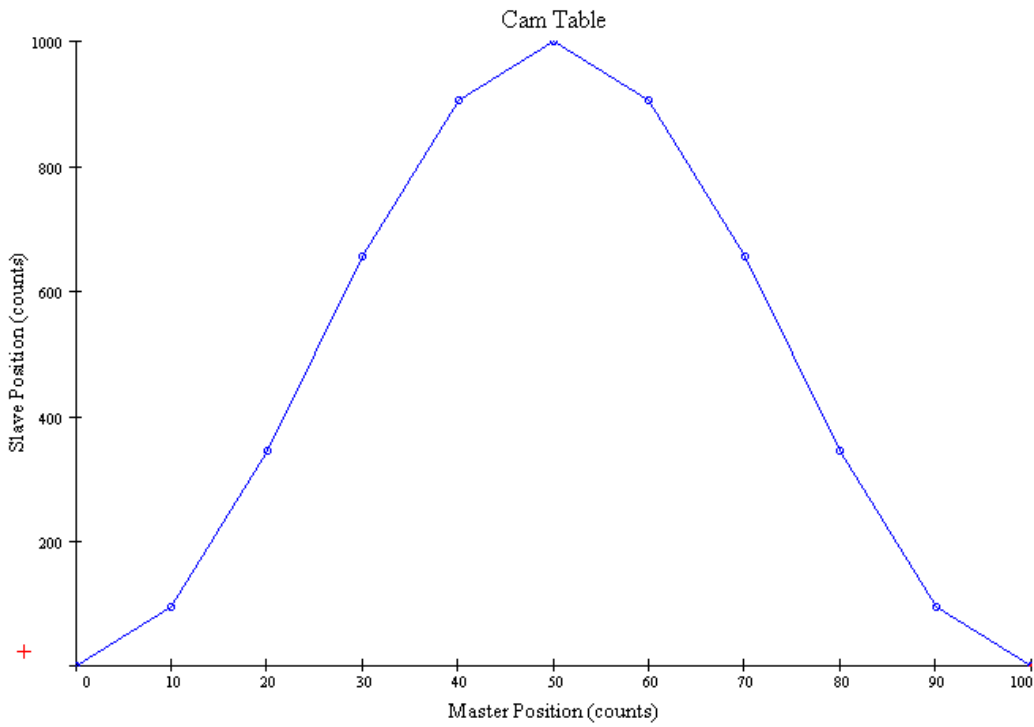


Figure 6. Cam table for which EM (100) is divisible by EP (10). The number of intervals N = 10.

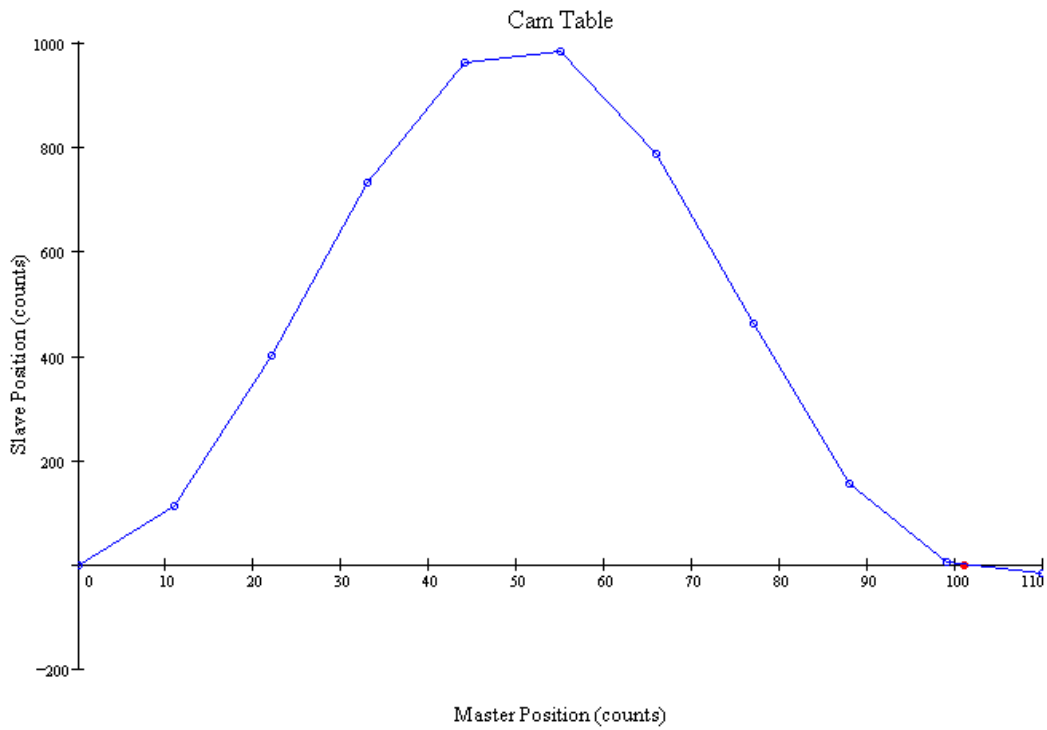


Figure 7. Cam table for which EM (101) is NOT divisible by EP (11). The number of intervals N = 10. Note how the “additional point method” is used to define the slope of the last interval.

```

#AUTO
amplitud = 1000 ;'slave distance in counts
period = 101 ;'master period in counts (EM)
'THE FOLLOWING NUMBER SHOULD BE CHANGED TO 256 FOR REAL APPLICATIONS
Nmax = 10 ;'maximum intervals (256 for standard fw)

input = period / Nmax ;'compute EP
JS#Ceiling
ep = output

input = period / ep ;'compute number of ET intervals
JS#Ceiling
intervls = output

DM A[257] ;'define array to hold cam table
EB0 ;'turn cam off
EAY ;'y is master
EM 0, period ;'slave change, master change
EP ep,0 ;'interval width, starting position

i = 0
#Array ;'fill the array with the sin wave
A[i] = amplitud / 2 * (1 - @COS[(360 / period) * i * ep])
ET[i] = A[i]
i = i + 1
JP #Array, i < intervls + 1

'adjust last point if necessary
IF(ep * intervls <> period)
m = -A[intervls - 1] / (period - (ep * (intervls - 1))) ;'slope
b = -m * period ;'y intercept
A[intervls] = m * ep * intervls + b ;'set new value
ET[intervls] = A[intervls]
ENDIF

EB1 ;'turn cam on
EGX=-100 ;'engage slave immediately
JGY=100 ;'jog y axis
BGY ;'begin motion
EN

#Ceiling ;'computes next highest integer (takes input. answer is in output)
IF(@FRAC[input] = 0) ;'input is integer
output = input ;'leave number unchanged
ELSE ;'input is decimal
output = @INT[input] + 1 ;'truncate and add one
ENDIF
EN

#Print ;'prints the cam table for debugging
j = 0
#Loop
ET[j] = ?
j = j + 1
JP#Loop, j <= intervls
EN

```