AN113

v1.0

Downloading CAM tables

Introduction

Most Copley digital servo drives support a mode of operation in which a CAM table is used to define the desired motor position. CAM tables are normally downloaded to the drive's flash file system using the CME interface software. The "Camming User Guide" document available in the downloads section of the Copley web site describes this mode of operation and the use of CME to create and download CAM tables.

In some cases it may be necessary to create and download CAM tables outside of the CME software environment. This document describes the CAM table format, and the necessary commands to download CAM tables to the drive.

CAM table format

When CAM tables are entered into CME they consist of pairs of master and slave positions, both in units of encoder counts. Both master and slave positions in the table are relative to the positions at the time the CAM table started.

When CME downloads a CAM table to the servo drive it first must change the format to a series of differences in positions from one row of the table to the next.

For example, if the following table was entered into CME:

Master	Slave
0	0
100	50
200	75
300	85
400	75
500	50
600	0

Then when CME downloaded the table to the drive, it would first subtract each row from the following row to get a table of changes in position for both master and slave. The resulting table which is actually used by the drive would contain the following values:

Master	Slave
100	50
100	25
100	10
100	-10
100	-25
100	-50

Each row in the table downloaded to the drive consists of two 16-bit values; the master position change and the slave position change. Master position changes must always be in the range 1 to 16383. Slave position changes can range from -32768 to 32767.

CAM tables in flash

Copley digital drives implement a simple flash file system which is used to store CVM (Copley Virtual Machine) programs and CAM tables. This is normally the location where CAM tables will be written.

CAM tables are identified by their file names. Up to ten CAM tables can be stored in flash with names that range from CAM0 to CAM9.

The binary serial interface can be used to download a CAM table to the drive's file system. The binary serial interface is defined by applications note number 112. Downloading a file to the file system consists of the following steps:

1. Allocate space in the file system for the new CAM table.

   This step uses sub-command 0 of opcode 20 (see the binary serial interface documentation). For example, to download the CAM table described above we would send the following binary serial command:

   ```
   0x00 0x43 0x03 0x14 0x00 0x00 0x00 0x0c 0x00 0x02
   ```

   The first 4 bytes are the standard binary serial command header. The header includes the checksum (0x43), the number of words of data (0x03) and the opcode number (0x14).

   The first data word after the header (0x0000) gives the sub-command. This sub-command is used to allocate space in the CVM file system.

   The next data word (0x000C) indicates that we will be allocating 12 words of data in the new file.

   The next data word (0x0002) indicates that we will be allocating 2 words of data for the name of the new file.

   The drive response from this command is formatted as follows:

   ```
   0x00 0x5b 0x01 0x00 0x00 0x00
   ```

   The first 4 bytes give the standard command header including a checksum (0x5B), number of words of data (0x01), and error code (0x00). The single word of data gives the file number of the newly allocated file (0x0000).

2. Set the name of the new CAM file.

   This step uses sub-command 1 of opcode 20. The first CAM table should be named CAM0, so to name this table we would send the following command:

   ```
   0x00 0x35 0x05 0x14 0x00 0x01 0x00 0x00 0x00 0x00 0x43 0x41 0x4d
   0x30
   ```

   The first 4 bytes are the header as always. The header indicates that 5 words of additional data will be passed. The first data word (0x0001) gives the sub-command used to write the file name. The second word gives the file number returned from the allocate command (0x0000), the third word gives the offset into the file name (0x0000), and the remaining two words give the file name in ASCII: 0x43 ('C'), 0x41 ('A'), 0x4D ('M'), 0x30 ('0').

3. Download the CAM table data to the file.

   This uses sub-command 2 of opcode 20.

   ```
   0x00 0x42 0x0f 0x14 0x00 0x02 0x00 0x00 0x00 0x00 0x00 0x64 0x00
   0x32 0x00 0x64 0x00 0x19 0x00 0x64 0x00 0x0a 0x00 0x64 0xff 0xf6
   0x00 0x64 0xff 0xe7 0x00 0x64 0xff 0xce
   ```

   As always, the first 4 bytes give the command header. Here we're passing 15 words of data with the command. The first data word is the sub-command (0x0002), the second is the file number (0x0000), the third gives an offset into the file where the data will be written (0x0000), and the remaining 12 give the master and slave increments for the CAM table: (0x0064,0x0032), (0x0064,0x0019), etc.

CAM tables in RAM

In some applications CAM tables need to be constantly updated during normal operation. In this case it is preferable to store the CAM tables in RAM rather then download them to flash memory.

Up to 16 CAM tables can be loaded into RAM at once. The block of RAM used to store the CAM tables starts with an index which is used by the drive to locate each table in RAM. The size of this index is 2 words for every CAM table that's being stored. For each stored table, the first word in the index gives the offset to the start of the table, and the second word in the index gives the size of the table in words.

For example, if 3 CAM tables were being stored in RAM, and the length of the three CAM tables was 20 words, 30 words and 20 words, then the contents of RAM would be as follows:

Memory location	Contents	Description
0	6	Location of first CAM table
1	20	Length of first CAM table
2	26	Location of second CAM table
3	30	Length of second CAM table
4	56	Location of third CAM table
5	20	Length of third CAM table
6 – 25	…	CAM table 1
26 – 55	…	CAM table 2
56 – 75	…	CAM table 3

The location where the CAM table is stored is the drive's trace buffer. The following steps must be taken to download a CAM table to this RAM area.

1. Allocate space in RAM

   Before a CAM table can be downloaded to the trace buffer area, a section of the trace buffer needs to be reserved for this use. To do this one would use sub-command 20 of opcode 15 (trace command).

   For example, to download the example CAM table described at the beginning of this applications note into RAM we would need to allocate 14 words of data from the trace buffer. 12 words are used to store the CAM table itself, and the remaining 2 words are used to store the index for a single CAM table.

   ```
   0x00 0x4d 0x02 0x0f 0x00 0x14 0x00 0x0e
   ```

   The first 4 words give the standard header used with all binary serial commands. These words include the Checksum (0x4D), number of additional bytes of data passed (0x02), and opcode (0x0F). The first word of data gives the sub-command used to allocate memory in the trace buffer (0x0014). The second word of data gives the number of words to reserve (0x000E).

2. Download the index and CAM table

   After reserving space in the trace buffer we can download the index and CAM table(s) by using sub-command 22 of the trace opcode:

   ```
   0x00 0x5c 0x10 0x0f 0x00 0x16 0x00 0x00 0x00 0x02 0x00 0x0c 0x00
   0x64 0x00 0x32 0x00 0x64 0x00 0x19 0x00 0x64 0x00 0x0a 0x00 0x64
   0xff 0xf6 0x00 0x64 0xff 0xe7 0x00 0x64 0xff 0xce
   ```

   The first 4 bytes give the header. Here we are passing an additional 16 words of data (0x10) and sending opcode 15 (0x0F). The first word of data passed (0x0016) gives the sub-command used to write data to the trace buffer. The second word of data (0x0000) gives the offset into the trace buffer where the remaining data should be stored.

   The next two words hold the index. The first gives the offset of the table (0x0002), and the second gives the length of the table (0x000C).

   The final 12 words of data give the CAM table data: (0x0064,0x0032), (0x0064,0x0019), …

Compressed CAM tables

It's frequently the case that the master increment in a CAM table is always the same value. This is true in the example CAM table used in this document where the master increment in each row of the table is equal to 100.

For such cases, it's possible to reduce the size of the CAM table by only sending the master increment value once. This reduces the amount of flash or RAM memory used to store the table, and also reduces the amount of data that needs to be send to the drive when the CAM table is updated.

The first word of a compressed CAM table should hold the master increment in bits 0-13, and bit 14 should be set to indicate that the table is compressed. The remaining words of the table should hold the slave position increments.

For example, the following CAM table is 12 words long in the normal format:

Master	Slave
100	50
100	25
100	10
100	-10
100	-25
100	-50

When passed to the drive in the normal format, the table data would take the following values in hex:

```
0x0064,0x0032,0x0064,0x0019,0x0064,0x000A,0x0064,0xFFF6,
0x0064,0xFFE7,0x0064,0xFFCE
```

Since all the master increments in the table are the same, it can be stored in the compressed format to save space. In that case, the data would be formatted as follows:

```
0x4064,0x0032,0x0019,0x000A,0xFFF6,0xFFE7,0xFFCE
```

Here bit 14 of the first word is set, and the lower 13 bits hold the master increment used in every row of the table. The remainder of the table consists only of slave increments. This reduces the size of the table from 12 words to 7 words.

Revision History

Date	Version	Revision
3/13/2012	1.0	Initial release