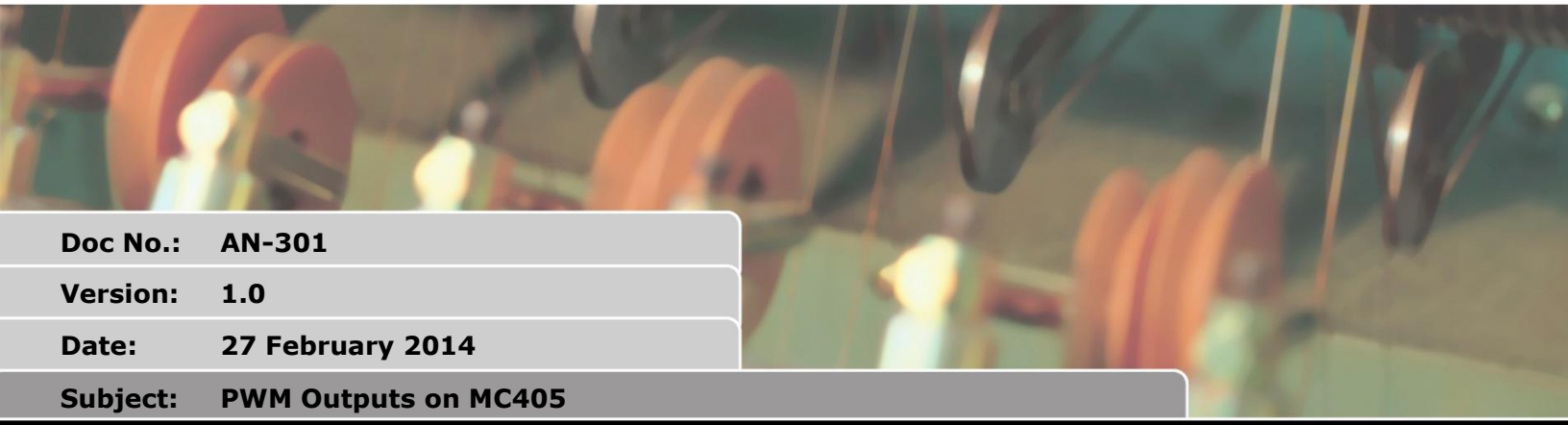
Shannon Way,
Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

187 Northpointe Blvd,
Suite 105
Freeport, PA 16229
United States of America
Tel: +1 724-540-5018
Fax: +1 724-540-5098

Tomson Centre
188 Zhang Yang Rd., B1602
Pudong New Area, Shanghai,
Postal code: 200122
CHINA
Tel/Fax: +86 21 587 97659

SCMC House
16/6 Vishal Nagar
Pimpale Nilakh, Wakad, Pune
PIN 411027
INDIA
Tel: +91 827 506 5441



Doc No.: AN-301

Version: 1.0

Date: 27 February 2014

Subject: PWM Outputs on MC405

APPLICATION NOTE

Introduction

PWM outputs are required for laser control and similar applications. This application note defines the functions available on the MC405:

- Programmable Mark/Space PWM outputs.
- Optional linking of PWM outputs to velocity profile speed of an interpolating axis group.
- Ability to define a PWM output intensity profile which can be linked to distance along an XY profile.
- Gating of PWM signal with sequence of positions on axis.
- Other PWM control options are possible. Contact Trio if you need further functionality.

Controller Configuration:

The PWM functions described here apply to MC405 Motion Coordinator fitted with FPGA version 15 or higher. This FPGA image is available in software version 2.0229 or higher. Note that the PWM facilities of the MC405 differ a little from the PWM features available on the earlier EURO209/EURO205X controller implementations.

If the FPGA version is 14 or lower the FPGA can be updated using “Reprogram FPGA….” option under the “Controller” menu in MPv3.2. The FPGA update files are built into MC405 software 2.0229 but are not loaded automatically.

Note that the MC405 software contains 2 FPGA images for different absolute encoder options. The standard image (0) for SSI and Tamagawa encoder support should be used for PWM functionality. The EnDAT absolute encoder support option does not currently contain PWM functionality.

Axis Configuration:

- The PWM signal is provided as an additional output on an axis using the third differential (Z) output. This uses the “enable / Z” output of the set of three (step/dir/enable). The ATYPE must be set by the programmer to an ATYPE that configures the Z channel as an output: The Step/Direction outputs/A+B inputs can be used in the normal way at the same time. ***NOTE: Software version 2.0229 only supports PWM on ATYPE=35***.

ATYPE	Description
35	Pulse and direction output with PWM output (maintains compatibility with older controllers)
36	Incremental encoder Servo with PWM output (maintains compatibility with older controllers)
43	Pulse and direction output with enable output (Recommended default: **ATYPE AXIS(4)= 43**)
45	Quadrature encoder output with enable output
77	Incremental encoder Servo with enable output
78	Pulse and direction with VFF_GAIN and enable output
84	Quadrature encoder output with VFF_GAIN and enable output

- Each axis of the MC405 has its own PWM circuit which can be programmed separately. The implementation therefore works with axes 0, 1, 2, 3 and 4 of the MC405. The PWM output of any axis can be linked to the speed of moves running on other axes. For example: Axis 4's PWM output can be linked to the velocity profile of interpolated moves running on axes 0, 1 and 2.

Functions:

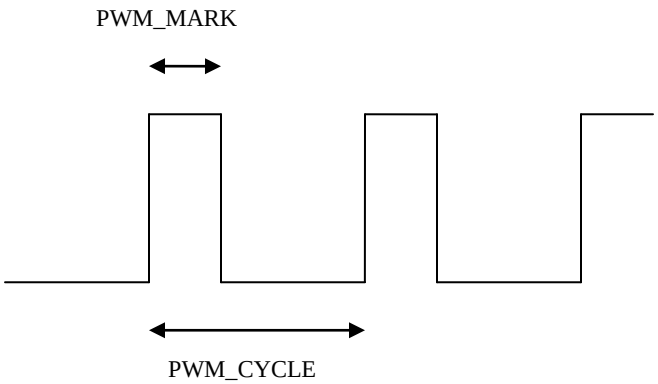
- The PWM output mark/space can be written to via application software program in a simple way
- The PWM mark size can be automatically set via an interpolated path speed
- The PWM mark size can be automatically set via an interpolated path speed AND a defined profile along an XY path.
- The MC405 implementation does feature the hardware gating functions of the earlier EURO205X PWM implementation.

PWM Generation:

The 20MHz clock is used as the base clock for the PWM generation.

PWM_CYCLE Axis parameter defines number of clocks in the combined mark + space period (Maximum value 65535)
PWM_MARK Axis parameter defines the number of ON clocks in the mark (Maximum value 65535)

For example: If PWM_CYCLE=1000, PWM will be generated at 20KHz frequency
If PWM_CYCLE=200, PWM will be generated at 100KHz frequency (but with lower resolution)



PWM Gating, Speed Linking and Profile Linking:

The PWM signal can be controlled and optionally gated with a series of positions. A dedicated 29 bit encoder hardware register is used for the position of the axis. This register can be read using the keyword PWM_ENCODER. When the axis moves the PWM_ENCODER register will change with the MPOS register. The PWM_ENCODER register is independent of the position register. It can be reset separately and is not changed by the DEFPOS and OFFPOS commands.

PWM_CONTROL(0)

The PWM output is switched OFF using this axis function.

PWM_CONTROL(1)

PWM output is switched on. The PWM_MARK value comes from the PWM_MARK axis parameter. NOTE that the WDOG must be ON for PWM generation.

PWM_CONTROL(2, multiplier, offset, limit value, link axis)

PWM output is calculated via VP_SPEED (path velocity of an axis) of an axis and multiplier and will be adjusted on each servo cycle. The value loaded into the PWM_MARK is given by VP_SPEED * multiplier + offset. If the calculated value exceeds the “limit value” the limit value is applied. The VP_SPEED value is the velocity profile speed for the link axis in user units.

PWM_CONTROL(3, GT/LT, ON/OFF, table start, table end)

This axis function loads a sequence of positions into the gating FIFO. The GT/LT value should be set to 1 for a GT comparison, 0 for a LT comparison. The ON/OFF bit is toggled as the positions are loaded automatically by the function. The FIFO is up to 512 positions deep, and can be loaded with one or more PWM_CONTROL(3,… functions. Longer sequences of positions can be loaded as the position buffer becomes available.

PWM_CONTROL(4)

The FIFO can be emptied using the PWM_CONTROL(4) function.

PWM_CONTROL(5)

The FIFO gating is switched on using the PWM_CONTROL(5) function.

PWM_CONTROL(6)

The PWM_ENCODER register can be cleared to zero using the PWM_CONTROL(6) function.

PWM_CONTROL(7)

Reserved.

PWM_CONTROL(8, multiplier, offset, limit value, velocity link axis, amplitude link axis)

PWM output is calculated via VP_SPEED (path velocity of an axis) of an axis and multiplier and will be adjusted on each servo cycle. The value loaded into the PWM_MARK is given by

VP_SPEED velocity link axis * DPOS amplitude link axis * multiplier + offset.

If this calculated value exceeds the “limit value” the limit value is applied. The VP_SPEED value is the velocity profile speed for the velocity link axis.

The DPOS of the amplitude link axis can be set using a CAMBOX profile so that the amplitude varies.
This is achieved using 3 stages:

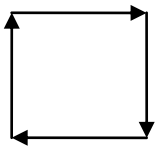
1. The CONNPATH(base XY) move type can be used to “connect” a virtual axis to an XY path.
2. A CAMBOX profile is then linked to the virtual axis.
3. The PWM_CONTROL(8,…) function then links to the CAMBOX profile

The PWM_CONTROL function is an axis function and can be directed to an axis using the AXIS() modifier or the BASE() function.

Example 1:

Using the PWM_CONTROL(2,…) automatic link to velocity mode:

XY Profile

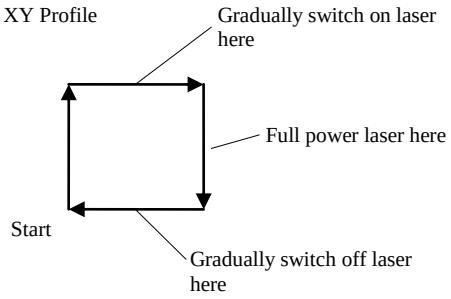


```
BASE(0,1)
SPEED=5000
ACC(50000)
MERGE=ON
MOVE(10000,0)
MOVE(0,-10000)
MOVE(-10000,0)
MOVE(0,10000)

' This program sets the PWM output to a minimum of 50 and maximum of 500.  At a speed of
2000 the PWM output will be 250:

BASE(4)
WDOG=OFF
WA(10)
ATYPE=35
WDOG=ON
PWM_CYCLE=1000' sets 20khz PWM
link=0'         link to axes 0 and 1
mult=0.1'     if speed=2000: pwm=2000*0.1=200
PWM_CONTROL(2,mult,50,500,link)
```

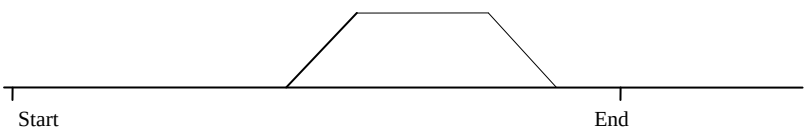
‘ Note that above speed=4500 the PWM will be limited to 500, by the limit setting

Example 2:

Using the PWM_CONTROL(8,…) automatic link to velocity mode with profile:



Suppose we want to control the laser power during the movement cycle:



Such a profile could be entered as a CAM profile using:

```
start=1000
TABLE(start,0,0,0,0,1000,1000,1000,0,0)
```

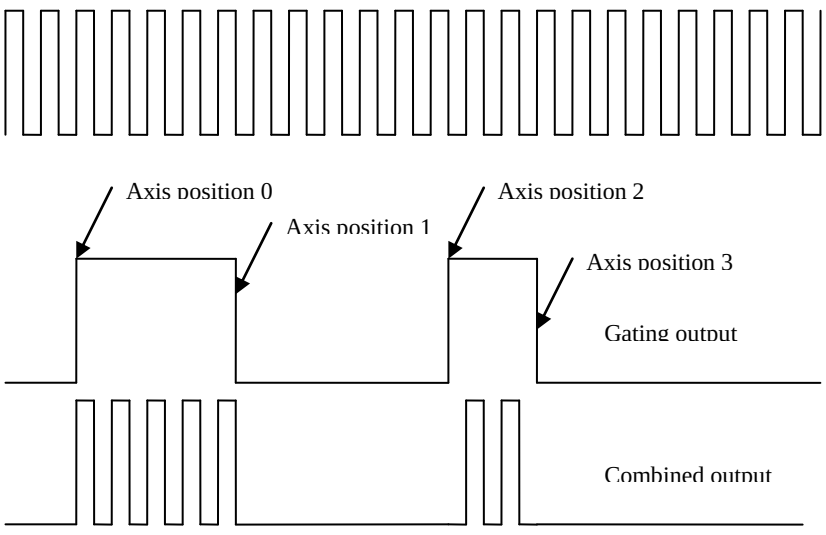
Note that the cam profile can be any complex shape and have any number of points.

Stage 1: Link a virtual axis to the XY path:

```
CONNPATH(0) AXIS(8)

BASE(4)
WDOG=OFF
WA(10)
ATYPE=35
WDOG=ON
PWM_CYCLE=1000' sets 20khz PWM
link=0'         link to axes 0 and 1
mult=1/10000'     if speed=5000: pwm=5000*(1/10000) * DPOS = 500
PWM_CONTROL(8,mult,50,500,link,9)

' Note how the CAMBOX links to a length of the box outline path of 40000:

TABLE(1000,0,0,0,0,1000,1000,1000,0,0)

BASE(0,1)
SPEED=10000
ACC(50000)
MERGE=ON
WHILE TRUE
    CAMBOX(1000,1008,1,40000,8) AXIS(9)
    MOVE(10000,0)
    MOVE(0,-10000)
    MOVE(-10000,0)
    MOVE(0,10000)
    WAIT IDLE
    WA(10)
    DEFPOS(0) AXIS(8)
WEND
```

Example 3:

Axis Gating Example:



```
WDOG=OFF
BASE(0)
PWM_CONTROL(0)
ATYPE=35 ' atype for stepper output with PWM
FE_LIMIT=500000000 ' dpos and mpos may not match
PWM_CYCLE=1000 ' 20Khz PWM output
PWM_MARK=0 ' switch off for now
UNITS=16 '
PWM_CONTROL(1)
DEFPOS(0)
PWM_CONTROL(6)
TABLE(1000,110,115,195,220)
TABLE(2000,220,195,115,110)
PWM_CONTROL(4)
PWM_CONTROL(5)' switch on gating function
PWM_MARK=400
WDOG=ON

REPEAT
    PWM_CONTROL(3,1,ON,1000,1003)' load positions to gate
    MOVEABS(300)
    WAIT IDLE
    PWM_CONTROL(3,0,ON,2000,2003)' same pattern in reverse
    MOVEABS(0)
    WAIT IDLE
UNTIL FALSE
```

The axis switch positions should be a sequence of axis positions.

For example, suppose the axis starts at position 100 and the laser should be switch ON at position 110 and OFF at 115, then ON at 195 and OFF at 220:

The positions are put into TABLE memory at locations 1000 to 1003:

TABLE(1000,110,115,195,220)

PWM_CONTROL(3,1,ON,1000,1003)

This command loads the 4 positions into the gating hardware and sets the comparison logic to GT. As the axis moves past the positions the gating output is toggled ON then OFF.

Note: After changing the ATYPE to 35 the PWM_ENCODER register will often have a value. It can be cleared to zero using the PWM_CONTROL(6) function.

If the PWM_MARK value equals or exceeds PWM_CYCLE, the output will be continuously on.