Using PMAC2’s Position Compare Circuit for Generalized PWM/PFM Output

The hardware circuits on any PMAC2-style axis-interface channel can be used to create a very flexible modulated digital-output signal with the ability to modulate either the pulse-width or the pulse frequency (or both) at will. This technique creates a powerful universal digital modulator, with more flexibility than either PMAC2’s built-in pulse-width-modulation (PWM) or pulse-frequency-modulation (PFM) circuitry.

The process involves several steps, each first given in summary, then explained in detail. In summary, the steps are:

1. Use the channel’s pulse-frequency-modulation circuitry to create a pulse train of fixed frequency.
2. Feed this pulse train into the channel’s encoder input to create a running counter value.
3. Use the auto-incrementing position-compare circuitry to create the output signal, with the auto-increment register determining the pulse frequency and the difference between the two compare registers determining the pulse width.

Creating the Fixed Pulse Train

To create the fixed pulse train that drives the whole process, the C sub-channel for the channel must be configured for PFM output. (Note that the output of this circuit is not the end signal trying to achieve; it is just an intermediate step in the process.) This involves several steps:

1. Select PFM-format output rather than PWM-format output.
2. Set the PFM clock frequency.
3. Set the PFM pulse width.
4. Set the PFM command value for the pulse frequency.

These differ slightly in their implementation depending on which channel of which type of PMAC2.

Non-Turbo PMAC2 Main Servo Channel

When using one of the main servo interface channels 1 – 8 on a PMAC2 with a regular (non-Turbo) CPU, use the following instructions for setting up the PFM signal and frequency.

1. Select PFM-format output for sub-channel C of Channel n by setting I9n6 to 2 or 3 (2 if sub-channels A and B are PWM format, 3 if sub-channels A and B are DAC format). Make sure I9n7 is 0 or 1 so these outputs are not inverted and I9n8 is 0 so the direction output alone is not inverted.
2. Select the PFM clock frequency with I903 if using Channel 1 – 4, or with I907 if using Channel 5 – 8. The pulse output frequency can range from one-fifth of this frequency to one-sixteen-millionth. The default setting of 9.83 MHz for the PFM clock frequency is suitable for almost all applications.
3. Select the PFM pulse width, expressed in PFM clock cycles, with I904 if using Channel 1 – 4, or with I908 if using Channel 5 – 8. The pulse must be at least two SCLK (encoder sample clock) cycles wide, or the encoder input circuits will not be able to detect it (the default SCLK frequency is also 9.83 MHz, and few people change this). The minimum PFM period (corresponding to the maximum PFM frequency) is two times the pulse width. If other channels in this set of four are being used for PWM output, this variable also sets the PWM deadtime, so in this case, there may not be complete independence in setting this value. Despite these constraints, usually there is a wide range of acceptable settings for this variable.

Note:

All of the I-Variable settings in Steps 1 – 3 can be made once during the application setup and stored in flash memory, so that these settings will be executed automatically at every succeeding power-up/reset.

4. Set the PFM output frequency by writing to the C sub-channel output register. This frequency needs
to be high enough to get the sub-cycle resolution needed at the highest resulting compare output frequency. For example, to get 1% resolution on a 4 kHz compare output frequency, this PFM output frequency must be at least 400 kHz.

Writing to the output register is done with an M-Variable. The suggested M-Variables for these registers are:

\[
\begin{align*}
\text{M107} & \rightarrow \ Y:\$C004,8,16,S & \text{; Channel 1 PFM output value} \\
\text{M207} & \rightarrow \ Y:\$C00C,8,16,S & \text{; Channel 2 PFM output value} \\
\text{M307} & \rightarrow \ Y:\$C014,8,16,S & \text{; Channel 3 PFM output value} \\
\text{M407} & \rightarrow \ Y:\$C01C,8,16,S & \text{; Channel 4 PFM output value} \\
\text{M507} & \rightarrow \ Y:\$C024,8,16,S & \text{; Channel 5 PFM output value} \\
\text{M607} & \rightarrow \ Y:\$C02C,8,16,S & \text{; Channel 6 PFM output value} \\
\text{M707} & \rightarrow \ Y:\$C034,8,16,S & \text{; Channel 7 PFM output value} \\
\text{M807} & \rightarrow \ Y:\$C03C,8,16,S & \text{; Channel 8 PFM output value}
\end{align*}
\]

With these definitions, the output frequency can be expressed as:

\[
\text{Output\_frequency} = \frac{\text{PFM\_clock\_frequency} \times \text{PFM\_output\_value}}{65,536}
\]

For example, with a PFM clock frequency of 9.83 MHz and a PFM output command of 512, the output frequency would be 9.83MHz/128 = 76.8 kHz.

If a finer resolution is needed in setting the PFM output frequency, treat the output register as a 24-bit value (e.g. \text{M107} \rightarrow \ Y:\$C004,0,24,S). In this case, the output frequency can be expressed as:

\[
\text{Output\_frequency} = \frac{\text{PFM\_clock\_frequency} \times \text{PFM\_output\_value}}{16,777,216}
\]

This value will need to be set again every time a sequence is started for a particular signal configuration (see below).

**Non-Turbo PMAC2 Supplemental Servo Channel**

When using one of the supplemental interface channels 1* or 2* on the JHW handwheel port of a PMAC2 with a regular (non-Turbo) CPU, use the following instructions for setting up the PFM signal and frequency.

**Note:**

Revision A and older of the DSPGATE2 IC that drive the JHW port on the PMAC2 only provides PFM output on Channel 1*. Revision B or newer of the DSPGATE2 IC must be used to get PFM output on Channel 2*.

1. Select PFM-format output for sub-channel C of Channel n* by setting bit 23 of X:$C095 to 1 for Channel 1*, or by setting bit 23 of X:$C09D to 1 for Channel 2*. This is done with an M-Variable (there is no I-Variable for this function for the supplemental channels). For example, the following M-Variable definitions could be used:

\[
\begin{align*}
\text{M991} & \rightarrow \ X:\$C095,23,1 & \text{; Channel 1* C output format control} \\
\text{M992} & \rightarrow \ X:\$C09D,23,1 & \text{; Channel 2* C output format control}
\end{align*}
\]

After every-power-up/reset, the M-Variable needs to be set to one. Usually this is done in a power-up PLC program.

2. Select the PFM clock frequency with I993. The pulse output frequency can range from one-fifth of this frequency to one-sixteenth millionth. The default setting of 9.83 MHz for the PFM clock frequency is suitable for almost all applications.
3. Select the PFM pulse width, expressed in PFM clock cycles, with I994. The pulse must be at least two SCLK (encoder sample clock) cycles wide, or the encoder input circuits will not be able to detect it (the default SCLK frequency is also 9.83 MHz, and few people change this). The minimum PFM period (corresponding to the maximum PFM frequency) is two times the pulse width. If the other channel in this set of two is being used for PWM output, this variable also sets the PWM deadtime, so in this case, there may not be complete independence in setting this value. Despite these constraints, usually there is a wide range of acceptable settings for this variable.

**Note:**
The I-Variable settings in Steps 2 and 3 can be made once during the application setup and stored in flash memory, so that these settings will be executed automatically at every succeeding power-up/reset.

4. Set the PFM output frequency by writing to the C sub-channel output register. This frequency must be high enough so that the sub-cycle resolution needed is obtained at the highest resulting compare output frequency. For example if 1% resolution is needed on a 4 kHz compare output frequency, this PFM output frequency must be at least 400 kHz.

Usually, writing to the output register is done with an M-Variable. Sample M-Variables for these registers are:

```plaintext
M997->Y: $C094, 8, 16, S ; Channel 1* PFM output value
M998->Y: $C09C, 8, 16, S ; Channel 2* PFM output value
```

With these definitions, the output frequency can be expressed as:

```
Output_frequency = PFM_clock_frequency * PFM_output_value / 65,536
```

**Example:** With a PFM clock frequency of 9.83 MHz and a PFM output command of 512, the output frequency would be 9.83MHz/128 = 76.8 kHz.

If finer resolution is needed in setting the PFM output frequency, treat the output register as a 24-bit value (e.g. `M997->Y: $C094, 0, 24, S`). In this case, the output frequency can be expressed as:

```
Output_frequency = PFM_clock_frequency * PFM_output_value / 16,777,216
```

This value must be set again every time a sequence for a particular signal configuration is started (see below).

**Turbo PMAC2 Main Servo Channel**
When using one of the main servo interface channels 1 – 8 on a Turbo PMAC2, use the following instructions for setting up the PFM signal and frequency.

1. Select PFM-format output for sub-channel C of Servo IC m Channel n by setting I7mn6 to 2 or 3 (2 if sub-channels A & B are PWM format, 3 if sub-channels A and B are DAC format). Make sure I7mn7 is 0 or 1 so these outputs are not inverted, and I7mn8 is 0 so the direction output alone is not inverted.

2. Select the PFM clock frequency for Servo IC m with I7m03. The pulse output frequency can range from one-fifth of this frequency to one-sixteen millionth. The default setting of 9.83 MHz for the PFM clock frequency is suitable for almost all applications.

3. Select the PFM pulse width for Servo IC m, expressed in PFM clock cycles, with I7m04. The pulse must be at least two SCLK (encoder sample clock) cycles wide, or the encoder input circuits will not be able to detect it (the default SCLK frequency is also 9.83 MHz, and few people change this). The minimum PFM period (corresponding to the maximum PFM frequency) is two times the pulse width. If other channels in this set of four are being used for PWM output, this variable also sets the PWM deadtime, so in this case, there may not be complete independence in setting this value. Despite these constraints, usually there is a wide range of acceptable settings for this variable.
Using PMAC2's Position Compare Circuit for Generalized PWM/PFM Output

**Note:**

All of the I-variable settings in Steps 1 – 3 can be made once during the application setup and stored in flash memory, so that these settings will be executed automatically at every succeeding power-up/reset.

4. Set the PFM output frequency by writing to the C sub-channel output register. This frequency must be high enough so that the sub-cycle resolution needed is obtain at the highest resulting compare output frequency. For example, to get 1% resolution on a 4 kHz compare output frequency, this PFM output frequency must be at least 400 kHz.

Usually, writing to the output register is done with an M-Variable. The suggested M-Variables for these registers on a Turbo PMAC2 are:

- **M107→Y:$078004, 8, 16, S**; Channel 1 PFM output value
- **M207→Y:$07800C, 8, 16, S**; Channel 2 PFM output value
- **M307→Y:$078014, 8, 16, S**; Channel 3 PFM output value
- **M407→Y:$07801C, 8, 16, S**; Channel 4 PFM output value
- **M507→Y:$078104, 8, 16, S**; Channel 5 PFM output value
- **M607→Y:$07810C, 8, 16, S**; Channel 6 PFM output value
- **M707→Y:$078104, 8, 16, S**; Channel 7 PFM output value
- **M807→Y:$07810C, 8, 16, S**; Channel 8 PFM output value

(Comparable M-Variable definitions for channels on Acc-24x2 boards can be found in the suggested M-Variable file in the Software Reference manual.)

With these definitions, the output frequency can be expressed as:

\[
\text{Output\_frequency} = \frac{\text{PFM\_clock\_frequency} \times \text{PFM\_output\_value}}{65,536}
\]

**Example:** With a PFM clock frequency of 9.83 MHz and a PFM output command of 512, the output frequency would be 9.83 MHz/128 = 76.8 kHz.

If a finer resolution in setting the PFM output frequency is needed, treat the output register as a 24-bit value (e.g. **M107→Y:$078004, 0, 24, S**). In this case, the output frequency can be expressed as:

\[
\text{Output\_frequency} = \frac{\text{PFM\_clock\_frequency} \times \text{PFM\_output\_value}}{16,777,216}
\]

This value must be set again every time a sequence for a particular signal configuration is started (see below).

### Non-Turbo PMAC2 Supplemental Servo Channel

When using one of the supplemental interface channels 1* or 2* on the JHW handwheel port of a Turbo PMAC2, use the following instructions for setting up the PFM signal and frequency.

**Note:**

Revision A and older of the DSPGATE2 IC that drive the JHW port on the PMAC2 only provide PFM output on Channel 1*. Revision B or newer of the DSPGATE2 IC must be used to get PFM output on Channel 2*.

1. Select PFM-format output for sub-channel C of Channel n* by setting I68n6 to 2 or 3 (2 if the generally unused sub-channels A and B are PWM format, 3 if sub-channels A and B are DAC format). Make sure I68n7 is 0 or 1 so these outputs are not inverted, and I68n8 is 0 so the direction output alone is not inverted.

2. Select the PFM clock frequency with I6803. The pulse output frequency can range from one-fifth of this frequency to one-sixteen millionth. The default setting of 9.83 MHz for the PFM clock frequency is suitable for almost all applications.
3. Select the PFM pulse width, expressed in PFM clock cycles, with I6804. The pulse must be at least 2 SCLK (encoder sample clock) cycles wide, or the encoder input circuits will not be able to detect it (the default SCLK frequency is also 9.83 MHz, and few people change this). The minimum PFM period (corresponding to the maximum PFM frequency) is two times the pulse width. If the other channel in this set of 2 is being used for PWM output, this variable also sets the PWM deadtime, so in this case, you may not have complete independence in setting this value. Despite these constraints, there is usually a wide range of acceptable settings for this variable.

**Note:**

All the I-Variable settings in Steps 1 – 3 can be made once during the application setup and stored in flash memory, so that these settings will be executed automatically at every succeeding power-up/reset.

4. Set the PFM output frequency by writing to the C sub-channel output register. This frequency must be high enough so that the sub-cycle resolution needed is obtained at the highest resulting compare output frequency. For example, if 1% resolution on a 4 kHz compare output frequency is needed, this PFM output frequency must be at least 400 kHz.

Writing to the output register is usually done with an M-Variable. Sample M-Variables for these registers are:

- M6807 -> Y: $078414, 8, 16, S
- M6808 -> Y: $07841C, 8, 16, S

With these definitions, the output frequency can be expressed as:

\[
Output\_frequency = \frac{PFM\_clock\_frequency \times PFM\_output\_value}{65,536}
\]

**Example:** With a PFM clock frequency of 9.83 MHz and a PFM output command of 512, the output frequency would be 9.83MHz/128 = 76.8 kHz.

For a finer resolution in setting the PFM output frequency, treat the output register as a 24-bit value (e.g. M6807 -> Y: $078414, 0, 24, S). In this case, the output frequency can be expressed as:

\[
Output\_frequency = \frac{PFM\_clock\_frequency \times PFM\_output\_value}{16,777,216}
\]

This value must be set again every time a sequence for a particular signal configuration is started (see below).

**Feeding the Encoder Channel**

The next step is to feed the pulse train into the encoder circuitry. This can be done with a software switch, or with external loop-back cabling. Following the instructions in this section will cause the encoder count frequency to be equal to the PFM pulse output frequency calculated above.

**Non-Turbo PMAC2 Main Servo Channel**

On a PMAC2 with a regular (non-Turbo) CPU, to feed the pulse train back into the encoder circuitry of the same channel without any cabling, set I9n0 of Channel n to 8. This causes the circuitry to take the internally generated pulse-and-direction signal automatically and puts the decode circuitry in pulse-and-direction mode.

To connect the pulse train to the encoder circuitry with an external cable, wire:

- PULSEm+ (a.k.a. PWMBOTn+) to CHAn+ PULSEm- (a.k.a. PWMBOTn+) to CHAn-DIRm+ (a.k.a. PWMTOPn+) to CHAn+ DIRm- (a.k.a. PWMTOPn+) to CHAn

**Note:**

In this method, the PFM output channel number (m here) does not necessarily have to match the encoder input channel number (n here).

In this method, set I9n0 for Encoder Channel n to 0 for external pulse-and-direction mode.
Non-Turbo PMAC2 Supplemental Servo Channel

On a PMAC2 with a regular (non-Turbo) CPU, to feed the pulse train back into the encoder circuitry of the same supplemental channel without any cabling, set I990 to 8 for Channel 1* or I991 to 8 for Channel 2*. This automatically causes the circuitry to take the internally generated pulse-and-direction signal directly, and puts the decode circuitry in pulse-and-direction mode.

Note:

This technique requires revision B or newer of the DSPGATE2 IC on the PMAC2. Revision A and older require external loop-back cabling, explained below. To connect the pulse train to the encoder circuitry with an external cable, wire:

PULSEm+ (pin 11 or 15 of JHW*) to CHAn+ PULSEm- (pin 12 or 16 of JHW*) to CHAn-DIRm+ (pin 13 or 17 of JHW*) to CHAn+ DIRm- (pin 14 or 18 of JHW*) to CHAn

Some manuals incorrectly have the PULSEm and DIRm lines exchanged in the documentation.

In this method, the PFM output channel number (m here) does not necessarily have to match the encoder input channel number (n here). The pulse train could even come from one of the main servo channels, or an external source.

Set I990 for Channel 1*, or I991 for Channel 2*, to 0 for external pulse-and-direction mode.

Turbo PMAC2 Main Servo Channel

On a Turbo PMAC2, to feed the pulse train back into the encoder circuitry of the same channel without any cabling, set I7mn0 of Servo IC m Channel n to 8. This automatically causes the circuitry to take the internally generated pulse-and-direction signal directly, and puts the decode circuitry in pulse-and-direction mode.

To connect the pulse train to the encoder circuitry with an external cable, wire:

PULSEp+ (a.k.a. PWMBOTn+) to CHAn+ PULSEp- (a.k.a. PWMBOTn+) to CHAnDIRp+ (a.k.a. PWMTOPn+) to CHAn+ DIRp- (a.k.a. PWMTOPn+) to CHAn

Note:

In this method, the PFM output channel number (p here) does not necessarily have to match the encoder input channel number (n here).

In this method, set I7mn0 for Servo IC m Encoder Channel n to 0 for external pulse-and-direction mode.

Turbo PMAC2 Supplemental Servo Channel

On a Turbo PMAC2, to feed the pulse train back into the encoder circuitry of the same supplemental channel without any cabling, set I6810 to 8 for Channel 1* or I6820 to 8 for Channel 2*. This automatically causes the circuitry to take the internally generated pulse-and-direction signal directly, and puts the decode circuitry in pulse-and-direction mode.

Note:

This technique requires revision B or newer of the DSPGATE2 IC on the PMAC2. Revision A and older require external loop-back cabling, explained below.

To connect the pulse train to the encoder circuitry with an external cable, wire:

PULSEm+ (pin 11 or 15 of JHW*) to CHAn+ PULSEm- (pin 12 or 16 of JHW*) to CHAnDIRm+ (pin 13 or 17 of JHW*) to CHAn+ DIRm- (pin 14 or 18 of JHW*) to CHAn

* Some manuals incorrectly have the PULSEm and DIRm lines exchanged in the documentation.
In this method, the PFM output channel number (m here) does not necessarily have to match the encoder input channel number (n here). The pulse train could even come from one of the main servo channels, or an external source. Set 16810 for Channel 1*, or 16820 for Channel 2*, to 0 for external pulse-and-direction mode.

**Setting up the Position Compare Circuits**

With the encoder counter incrementing at a known, steady frequency, the position-compare circuitry can be used to generate variable-width, variable-frequency pulse trains.

The position-compare circuit’s auto-increment register sets the frequency of the position-compare output. The frequency can be calculated as

\[
\text{Compare\_output\_frequency} = \frac{\text{Count\_frequency}}{\text{Auto-increment\_value}}
\]

**Example:** With a count frequency of 400 kHz and an auto increment value of 100, the compare output frequency would be 4 kHz.

The difference between the circuit’s Compare A and Compare B registers sets the pulse width, and therefore the duty cycle. The pulse width in time can be calculated as:

\[
\text{Pulse\_width} = \frac{(\text{Compare\_B} – \text{Compare\_A})}{\text{Count\_frequency}}
\]

If the count frequency is expressed in kHz, the pulse width will be expressed in milliseconds. The duty cycle in percent can be calculated as:

\[
\text{Duty\_cycle} = 100\% \times \frac{(\text{Compare\_B} – \text{Compare\_A})}{\text{Auto-increment\_value}}
\]

**Non-Turbo PMAC2 Main Servo Channel**

To set up the position-compare output value on Channel 1 – 8 of a PMAC2 with a regular non-Turbo CPU, several registers must be used. The suggested M-Variables for these registers on Channel 1 are:

- M101→X:$C001,0,24,S
- M107→Y:$C004,8,16,S
- M108→X:$C007,0,24,S
- M109→X:$C007,0,24,S
- M110→X:$C006,0,24,S
- M111→X:$C005,11
- M112→X:$C005,12

The suggested M-variable definitions for other channels can be found in the Software Reference manual.

To set up the compare circuitry for a given frequency and pulse width, a section of PMAC code such as the following (in either a PLC or motion program, or with on-line commands) can be used. In this code,

- P110 is the auto-increment value necessary to produce the desired frequency
- P109 is difference between A and B necessary to produce the desired pulse width
- P108 is the phase of the output compared to the present counter value (0 < P108 < P110)
- P107 is the PFM output value necessary to produce the desired counter frequency

```
M107=0 ; Freeze the encoder counter during setup
M12=0 ; Prepare to set output initial state off
M11=0 ; Force the initial state of the compare output
M10=P110 ; Set the auto-increment value for frequency
M08=M101+P108 ; Set first ON point
M09=M108+P109–P110 ; Set the first OFF point (minus one cycle)
M07=P107 ; Start PFM frequency to generate compare pulses
```

Because the B value is incremented when the A compare position is reached (and vice-versa), initially it is set one cycle below where its first real comparison will be.

In this case, the compare pulses are brought out on the J8 (JEQU) position-compare port on the EQUn line for the Channel n compare circuit.
Non-Turbo PMAC2 Supplemental Servo Channel

To set up the position-compare output value on supplemental channel 1* or 2* of a PMAC2 with a regular non-Turbo CPU, several registers must be used. Sample M-Variables for Channel 1* registers are:

- **M901**→**X:**$C091,0,24,S ; Encoder 1* counter value
- **M907**→**Y:**$C094,8,16,S ; PF M 1* output value (from above)
- **M908**→**Y:**$C097,0,24,S ; Compare 1* A position
- **M909**→**X:**$C097,0,24,S ; Compare 1* B position
- **M910**→**X:**$C096,0,24,S ; Compare 1* auto-increment value
- **M911**→**X:**$C095,11 ; Compare 1* initial-state write enable
- **M912**→**X:**$C095,12 ; Compare 1* initial-state

The matching registers for Channel 2* are eight numerical addresses higher.

To set up the compare circuitry for a given frequency and pulse width, a section of PMAC code such as the following (in either a PLC or motion program, or with on-line commands) can be used. In this code:

- **P910** is the auto-increment value necessary to produce the desired frequency
- **P909** is difference between A and B necessary to produce the desired pulse width
- **P908** is the phase of the output compared to the present counter value (0 < P908 < P910)
- **P907** is the PFM output value necessary to produce the desired counter frequency

```
M907=0 ; Freeze the encoder counter during setup
M912=0 ; Prepare to set output initial state off
M911=0 ; Force the initial state of the compare output
M910=P910 ; Set the auto-increment value for frequency
M908=M901+P908 ; Set first ON point
M909=M908+P909-P910 M907=P907 ; Set the first OFF point (minus one cycle)
M107=0 ; Start PFM frequency to generate compare pulses
```

Because the B value is incremented when the A compare position is reached (and vice-versa), initially it is set one cycle below where its first real comparison will be.

In this case, the position-compare lines are alternate uses of general-purpose I/O lines on the J2 (JTHW) multiplexer port. **EQU1** is brought out on the DAT4 line, pin 11 of J2. **EQU2** is brought out on the DAT5 line, pin 13 of J2. With this alternate use of these pins, none of the multiplexed I/O accessories for the JTHW port can be supported.

A few registers must be set up after every power-up/reset to convert these pins to alternate use. Suggested M-variables for these registers are:

```
M61→Y:$E800,4 ; Buffer direction control for DAT0 – DAT7, ISA/PCI boards
M61→Y:$E802,0 ; Buffer direction control for DAT0 – DAT7, VME boards
M64→Y:$C086,0,8 ; Alternate use control for DAT0 – DAT7
```

To set up these pins for alternate use and the buffer for output, the following code could be used:

```
M64=M64&207 ; Clear bits 4 (value 16) and 5 (value 32) for alt use of DAT4 & DAT5 (255–48=207)
M61=1 ; Set byte-wide buffer’s direction to output
```

Turbo PMAC2 Main Servo Channel

To set up the position-compare output value on Channel 1 – 8 of a Turbo PMAC2, several registers must be used. The suggested M-Variables for these registers on Channel 1 are:

```
M101→X:$078001,0,24,S ; Encoder counter value
M107→Y:$078004,8,16,S ; PFM output value (from above)
M108→Y:$078007,0,24,S ; Compare A position
M109→X:$078007,0,24,S ; Compare B position
M110→X:$078006,0,24,S ; Compare auto-increment value
M111→X:$078005,11 ; Compare initial-state write enable
M112→X:$078005,12 ; Compare initial-state
```
The suggested M-variable definitions for other channels can be found in the Software Reference manual. To set up the compare circuitry for a given frequency and pulse width, a section of PMAC code such as the following (in either a PLC or motion program, or with on-line commands) can be used. In this code:

- P110 is the auto-increment value necessary to produce the desired frequency
- P109 is difference between A and B necessary to produce the desired pulse width
- P108 is the phase of the output compared to the present counter value (0 < P108 < P110)
- P107 is the PFM output value necessary to produce the desired counter frequency

```
M107=0 ; Freeze the encoder counter during setup
M112=0 ; Prepare to set output initial state off
M110=P110 ; Force the initial state of the compare output
M108=M101+P108 ; Set first ON point
M109=M108+P109-P110 ; Set first OFF point (minus one cycle)
M107=P107 ; Start PFM frequency to generate compare pulses
```

Because the B value is incremented when the A compare position is reached (and vice-versa), initially it is set one cycle below where its first real comparison will be.

In this case, the compare pulses are brought out on the J8 (JEQU) position-compare port on the EQUn line for the Channel n compare circuit.

**Turbo PMAC2 Supplemental Servo Channel**

To set up the position-compare output value on supplemental channel 1* or 2* of a Turbo PMAC2, several registers must be used. Sample M-Variables for Channel 1* registers are:

```
M6801->X:$078411,0,24,S ; Encoder 1* counter value
M6807->Y:$078414,8,16,S ; PFM 1* output value (from above)
M6808->Y:$078417,0,24,S ; Compare 1* A position
M6809->X:$078417,0,24,S ; Compare 1* B position
M6810->X:$078416,0,24,S ; Compare 1* auto-increment value
M6811->X:$078415,11 ; Compare 1* initial-state enable
M6812->X:$078415,12 ; Compare 1* initial-state
```

The matching registers for Channel 2* are eight numerical addresses higher.

To set up the compare circuitry for a given frequency and pulse width, a section of PMAC code such as the following (in either a PLC or motion program, or with on-line commands) can be used. In this code:

- P6810 is the auto-increment value necessary to produce the desired frequency
- P6809 is difference between A and B necessary to produce the desired pulse width
- P6808 is the phase of the output compared to the present counter value (0 < P6808 < P6810)
- P6807 is the PFM output value necessary to produce the desired counter frequency

```
M6807=0 ; Freeze the encoder counter during setup
M6812=0 ; Prepare to set output initial state off
M6811=0 ; Force the initial state of the compare output
M6810=P6810 ; Set the auto-increment value for frequency
M6808=M6801+P6808 ; Set first ON point
M6809=M6808+P6809-P6810 ; Set first OFF point (minus 1 cycle)
M6807=P6807 ; Start PFM frequency to generate compare pulses
```

Because the B value is incremented when the A compare position is reached (and vice-versa), it is initially set one cycle below where its first real comparison will be.

In this case, the position-compare lines are alternate uses of general-purpose I/O lines on the J2 (JTHW) multiplexer port. EQU1* is brought out on the DAT4 line, pin 11 of J2. EQU2* is brought out on the DAT5 line, pin 13 of J2. With this alternate use of these pins, none of the multiplexed I/O accessories for the JTHW port can be supported.
A few registers must be set up after every power-up/reset to convert these pins to alternate use. Suggested M-Variables for these registers are:

M61→Y:$070000, 4
; Buffer direction control for DAT0 – DAT7,
; ISA/PCI boards

M61→Y:$070002, 0
; Buffer direction control for DAT0 – DAT7,
; VME boards

M64→Y:$078406, 0, 8
; Alternate use control for DAT0 – DAT7

To set up these pins for alternate use and the buffer for output, the following code could be used:

M64=M64&207
; Clear bits 4 (value 16) and 5 (value 32) for
; alt use of DAT4 & DAT5 (255−48=207)

M61=1
; Set byte-wide buffer’s direction to output