PMAC Position Registers

The PMAC Executive position window or the online **P** command reports the value of the actual position register plus the position bias register plus the compensation correction register, and if bit 16 of Ix05 is 1 (handwheel offset mode), minus the master position register:

```
M175->X:$002A,16,1          ; Bit 16 of I105
M162->D:$002B               ; #1 Actual position (1/[Ix08*32] cts)
M164->D:$0813               ; #1 Position bias (1/[Ix08*32] cts)
M167->D:$002D               ; #1 Present master ((handwheel) pos (1/[Ix07*32] cts of master or
                            ; (1/[Ix08*32] cts of slaved motor)
M169->D:$0046               ; #1 Compensation correction
```

$$P100 = \frac{(M162 + M164 + M169 - M175 * M167)}{I108 * 32}$$

P100 will report the same value as the **P** online command or the position window in the PMAC Executive program. The addresses given are for Motor #1. For the registers for another motor x, add (x-1)*$3C – (x-1)*60 – to the appropriate motor #1 address.).

```
M161->D:$0028               ; #1 Commanded position (1/[Ix08*32] cts)
```

The motor commanded position registers contain the value in counts where the motor is commanded to move. It is set through **JOG** online commands or axis move commands (**X10**) inside motion programs.

To read this register in counts: `P161 = M161 / (I108*32)`

```
M162->D:$002B               ; #1 Actual position (1/[Ix08*32] cts)
```

The actual position register contains the information read from the feedback sensor after it has been converted properly through the encoder conversion table and extended from a 24-bits register to a 48-bits register.

To read this register in counts: `P162 = M162 / (I108*32)`

```
M163->D:$080B               ; #1 Target (end) position (1/[Ix08*32] cts)
```

This register contains the most recent programmed position and it is called the target position register. If I13>0, PMAC is in segmentation mode and the value of M163 corresponds to the last interpolated point calculated.

To read this register in counts: `P163 = M163 / (I108*32)`

```
M164->D:$0813               ; #1 Position bias (1/[Ix08*32] cts)
```

This register contains the offset specified in the axis definition command `#1->X + <offset>`.

The **{axis}={constant}** online command or the **PSET** motion program command adds the specified offset to the existing M164 offset: `M164 = M164 + <new_offset>`.

To read this register in counts: `P164 = M164 / (I108*32)`

```
M165->L:$081F               ; &1 X-axis target position (engineering units)
```

M165 contains the programmed axis position through a motion program, X10 for example, in engineering units. In addition, it is updated by the **"{axis}={constant}"** online command or the **PSET** motion program command.

```
M166->X:$0033,0,24,S        ; #1 Actual velocity (1/[Ix09*32] cts/cyc)
```

M166 is the actual velocity register. For display purposes, use the Motor filtered actual velocity, M174

To read this register in cts/msec: `P166 = M166 * 8388608 / (I109 * 32 * I10 * (I160+1))`

```
M167->D:$002D               ; #1 Present master ((handwheel) pos (1/[Ix07*32] cts ; of master or
                            ; (1/[Ix08*32] cts of slaved motor)
```

M167 is related to the master/slave relationship set through Ix05 and Ix06. It contains the present number of counts the master.

To read this register in counts:
```
P167 = M167 / (I108*32)
            or
P167 = M167 / (I107*32)
```

```
M169->D:$0046                ; #1 Compensation correction
```

Calculated leadscrew compensation correction according to actual position (M162) and the leadscrew compensation table set through the define comp command.

To read this register in counts:
```
P169 = M169 / (I108*32)
```

```
M172->L:$082B                ; #1 Variable jog position/distance
```
counts)

Contains the distance for the **J=*** command.

Example:
```
M172=2000  J=*               ; Jog to position 2000 encoder counts
M173->Y:$0815,0,24,S         ; #1 Encoder home capture offset (counts)
```

Contains the home offset from the reset/power-on position. Important for the capture/compare features.

Example:
```
If (M117=1)
P103=M103-M173               ; Captured position minus offset endif
M174->Y:$082A,24             ; #1 filtered actual velocity (1/[Ix09*32] cts/servo cycle)
```

These registers contain the actual velocities averaged over the previous 80 real-time interrupt periods (80*[I8+1] servo cycles); good for display purposes.

To read this register in cts/msec:
```
P174 = M174 * 8388608 / (I109 * 32 * I10 *
(I160+1))
```

```
M176->D:$0840                ; #1 following error (1/[Ix08*32] cts)
```

Following error is the difference between motor desired and measured position at any instant. When the motor is open-loop (killed or enabled), following error does not exist and PMAC reports a value of 0.

$$P176 = \frac{(M161 - M162 + M164 = M169 * M167)}{I108 * 32}$$

To read this register in counts:
```
P176 = M176 / (I108*32)
```