

Concept

The following figure illustrates how Pulse and Phase Width Modulation of both *ENABLE* (on/off) and *PHASE* (+/- polarity) can be optimized to control DC brush motor position.

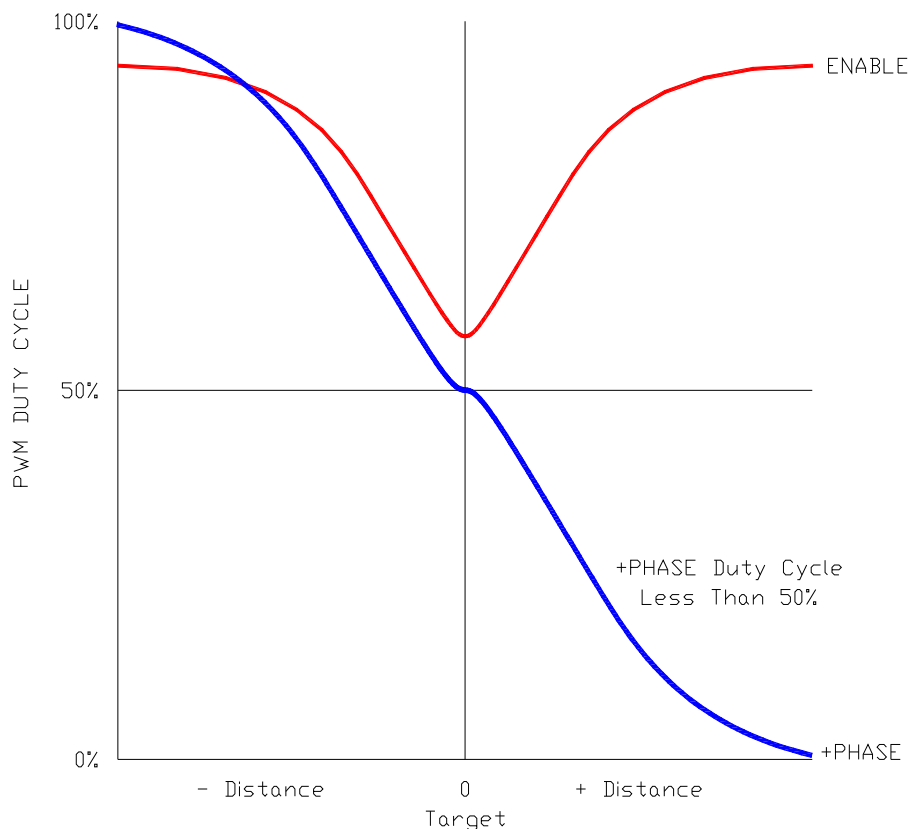


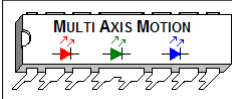
Figure - PWM Duty Cycle VS Target Location

Project Goals

The hypothesis of this project is that it should be possible to combine both *Pulse* and *Phase* modulation to provide measurably more efficient motor control than either method used by itself. The goal of this project is to determine the best methods for combining both *Pulse* and *Phase* modulation for brush DC motor control. Which method uses the least amount of power to hold a position? Which method provides smoother motion to and from the position? What is the best way to combine both methods to minimize the power required to hold a position and provide mobility?

Conclusion

Observations from the experiments described in this document clearly showed that a combination of *ENABLE* and *PHASE* control provides better position control than either does by itself for the brushed DC motor used in the experiments.



Experiment 1 - Pulse Width Modulation

This experiment uses a dsPIC30F4011 micro controller, 14.000 MHz crystal oscillator, Allegro A3953 Full Bridge Motor Controller, Pittman brushed DC gear motor, and 12V PC power supply.

The dsPIC30F4011 can run at up to 120 MHz.

$$F_{osc} = 14 \text{ MHz} * 8 \text{ pll} = 112 \text{ MHz}$$

$$F_{cy} = F_{osc}/4 = 28 \text{ MHz.}$$

PWM frequency of 20 KHz (above normal hearing frequencies) produces a period of 50 μ s. The dsPIC30F4011's PTPER register calculation for this period is

$$PTPER = ((F_{cy} * 1000) / f \text{ KHz}) - 1$$

$$PTPER\{20\text{KHz}\} = ((28 \text{ MHz} * 1000) / 20 \text{ KHz}) - 1 = 1399 \text{ (or } 0x0577)$$

Configure PWM Frequency

Run the following MCLI application and measure the results on the oscilloscope or frequency counter.

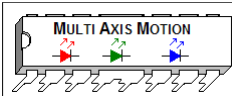
```
; Set the MCU's PWM Registers for 20 KHz @ 50% Duty Cycle  
; PTPER + 1 = PDC for 50% duty cycle  
MOTOR 0x20 PWM1 0x0577 0x0578 0x03 0x03
```

The A3953 output from these values is about 25 μ s ON and 25 μ s OFF. We loose about 4 μ s in switching performed by the A3953.

Test and Record Output

Run the following MCLI application and measure the results on the oscilloscope and digital multi-meter. Start with 0% and record values; uncomment next line, run, record values, etc.

```
MOTOR 0x20 PWM1 0x0577 0x0000 0x03 0x03 ; 0% DC  
; MOTOR 0x20 PWM1 0x0577 0x008B 0x03 0x03 ; 5% DC  
; MOTOR 0x20 PWM1 0x0577 0x0117 0x03 0x03 ; 10% DC  
; MOTOR 0x20 PWM1 0x0577 0x01A3 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x022F 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x02BB 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x0347 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x03D3 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x045F 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x04EB 0x03 0x03 ; 45% DC  
; MOTOR 0x20 PWM1 0x0577 0x0577 0x03 0x03 ; 50% DC  
; MOTOR 0x20 PWM1 0x0577 0x0602 0x03 0x03 ; 55% DC  
; MOTOR 0x20 PWM1 0x0577 0x068E 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x071A 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x07A6 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x0832 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x08BE 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x094A 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x09D6 0x03 0x03  
; MOTOR 0x20 PWM1 0x0577 0x0A62 0x03 0x03 ; 95% DC  
; MOTOR 0x20 PWM1 0x0577 0x0AEE 0x03 0x03 ; 100% DC
```

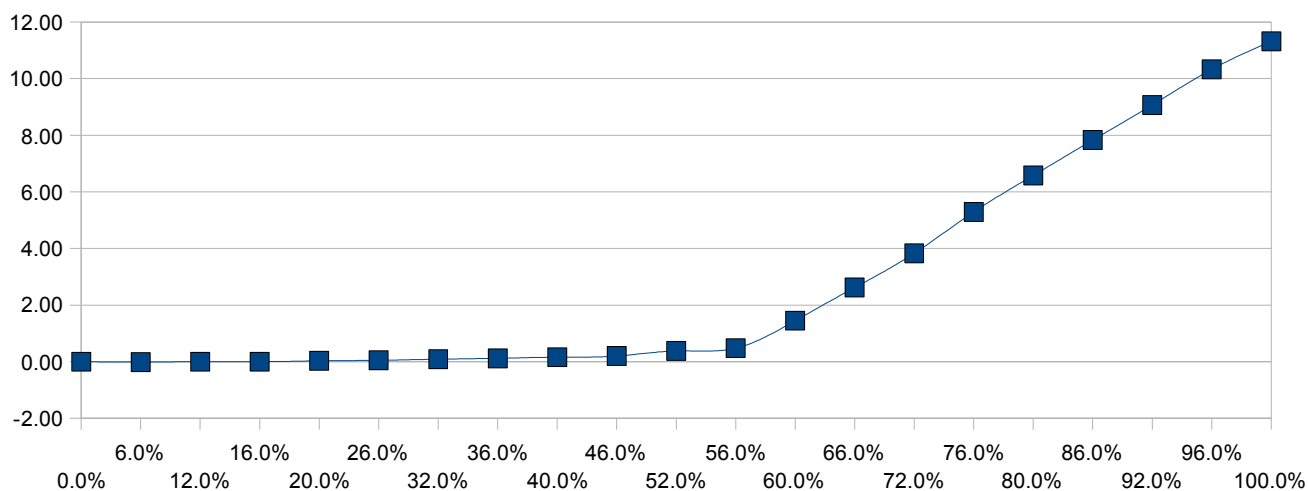


Pulse and Phase Modulation for DC Brush Motor Motion Control

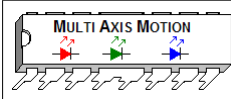
12/23/2007
ajs

PWM Duty Cycle VS Voltage Output

			PDC1	Duty Cycle	Out A	Out B	V[A-B]	
Fcy MHz	28	0	0	0.0%	0.00	0.00	0.00	
Fpwm KHz	20	1	140	6.0%	3.47	3.46	-0.01	
PTPER	1399	2	280	12.0%	2.95	2.95	0.00	
Vbb	12.27	3	420	16.0%	2.89	2.89	0.00	
		4	560	20.0%	3.04	3.07	0.03	
		5	700	26.0%	3.32	3.37	0.05	
		6	839	32.0%	3.67	3.76	0.09	
		7	979	36.0%	4.08	4.20	0.12	
		8	1119	40.0%	4.50	4.66	0.16	
		9	1259	46.0%	4.94	5.14	0.20	
		10	1399	52.0%	5.20	5.58	0.38	
		11	1539	56.0%	5.66	6.14	0.48	
		12	1679	60.0%	5.28	6.73	1.45	<- Rotation
		13	1819	66.0%	4.68	7.30	2.62	
		14	1959	72.0%	4.07	7.90	3.83	
		15	2099	76.0%	3.25	8.54	5.29	
		16	2238	80.0%	2.59	9.17	6.58	
		17	2378	86.0%	1.95	9.78	7.83	
		18	2518	92.0%	1.31	10.38	9.07	
		19	2658	96.0%	0.66	10.99	10.33	
		20	2798	100.0%	0.13	11.45	11.32	Max Speed



PWM Duty Cycle VS Voltage Output



Experiment 2 - Phase Modulation

This experiment uses a dsPIC30F4011 micro controller, 14.000 MHz crystal oscillator, Allegro A3953 Full Bridge Motor Controller, Pittman brushed DC gear motor, and 12V PC power supply.

The dsPIC30F4011 can run at up to 120 MHz.

$$F_{osc} = 14 \text{ MHz} * 8 \text{ pll} = 112 \text{ MHz}$$

$$F_{cy} = F_{osc}/4 = 28 \text{ MHz.}$$

PWM frequency of 20 KHz (above normal hearing frequencies) produces a period of 50 μ s. The dsPIC30F4011's PTPER register calculation for this period is

$$PTPER = ((F_{cy} * 1000) / f \text{ KHz}) - 1$$

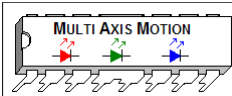
$$PTPER\{20\text{KHz}\} = ((28 \text{ MHz} * 1000) / 20 \text{ KHz}) - 1 = 1399 \text{ (or } 0x0577\text{)}$$

In this experiment, we will use 2 control pins on the A3953: *ENABLE* and *PHASE*. The *ENABLE* pin is Active Low, so it will be set Low for 100% of the cycle. We will vary the Hi/Lo duty cycle on the *PHASE* pin.

Test and Record PWM Output

Run the following MCLI application and measure the results on the oscilloscope and digital multi-meter. The program exercises the *PHASE* output in 5% increments. Start with 0% and record values; uncomment next line, run, record values, etc.

```
MOTION 0x20 HOME 1
; Set PWM1 to 100% Duty Cycle
; MOTOR 0x20 PWM1 0x0577 0x0AEE 0x03 0x00
; Set the PWM2, PTPER, PDC, Release Break, and Enable MCU 0x20
; MOTOR 0x20 PWM2 0x0577 0x0000 0x03 0x03 ; 0%
; MOTOR 0x20 PWM2 0x0577 0x008B 0x03 0x03 ; 5%
; MOTOR 0x20 PWM2 0x0577 0x0117 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x01A3 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x022F 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x02BB 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x0347 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x03D3 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x045F 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x04EB 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x0577 0x03 0x03 ; 50%
; MOTOR 0x20 PWM2 0x0577 0x0602 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x068E 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x071A 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x07A6 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x0832 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x08BE 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x094A 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x09D6 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x0A62 0x03 0x03
; MOTOR 0x20 PWM2 0x0577 0x0AEE 0x03 0x03 ; 100%
```

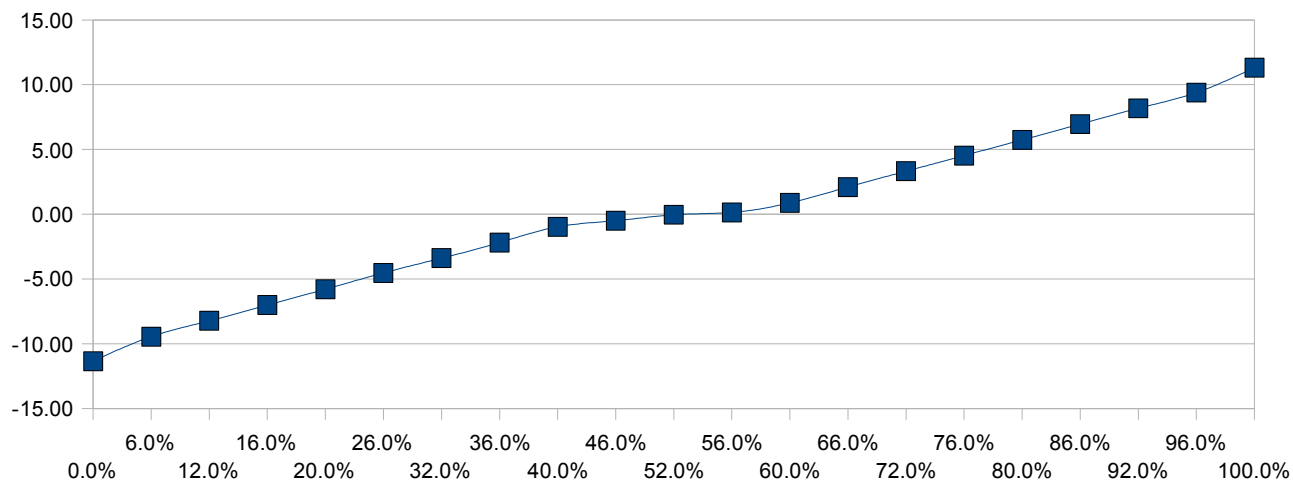


Pulse and Phase Modulation for DC Brush Motor Motion Control

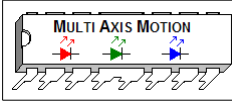
12/23/2007
ajs

Phase Modulation VS Voltage Output

			PDC1	Duty Cycle	Out A	Out B	V[A-B]	
Fcy MHz	28	0	0	0.0%	11.47	0.13	-11.34	
Fpwm KHz	20	1	140	6.0%	10.57	1.12	-9.45	
PTPER	1399	2	280	12.0%	10.00	1.77	-8.23	
Vbb	12.27	3	420	16.0%	9.43	2.42	-7.01	
		4	560	20.0%	8.86	3.07	-5.79	
		5	700	26.0%	8.25	3.72	-4.53	
		6	839	32.0%	7.70	4.31	-3.39	
		7	979	36.0%	7.15	4.96	-2.19	
		8	1119	40.0%	6.57	5.60	-0.97	<- Rotation
		9	1259	46.0%	6.30	5.80	-0.50	
		10	1399	52.0%	6.04	6.00	-0.04	
		11	1539	56.0%	5.89	6.03	0.14	
		12	1679	60.0%	5.65	6.53	0.88	<- Rotation
		13	1819	66.0%	5.00	7.10	2.10	
		14	1959	72.0%	4.36	7.68	3.32	
		15	2099	76.0%	3.72	8.25	4.53	
		16	2238	80.0%	3.08	8.82	5.74	
		17	2378	86.0%	2.43	9.39	6.96	
		18	2518	92.0%	1.79	9.97	8.18	
		19	2658	96.0%	1.14	10.53	9.39	
		20	2798	100.0%	0.13	11.45	11.32	Max Speed



Phase Modulation VS Voltage Output



Experiment 3 - Proportional Control with PWM

This experiment uses a dsPIC30F4011 micro controller, 14.000 MHz crystal oscillator, Allegro A3953 Full Bridge Motor Controller, Pittman brushed DC gear motor, and 12V PC power supply.

The dsPIC30F4011 can run at up to 120 MHz.

$$F_{osc} = 14 \text{ MHz} * 8 \text{ pll} = 112 \text{ MHz}$$

$$F_{cy} = F_{osc}/4 = 28 \text{ MHz.}$$

PWM frequency of 20 KHz (above normal hearing frequencies) produces a period of 50 μ s. The dsPIC30F4011's PTPER register calculation for this period is

$$PTPER = ((F_{cy} * 1000) / f \text{ KHz}) - 1$$

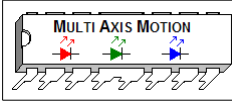
$$PTPER\{20\text{KHz}\} = ((28 \text{ MHz} * 1000) / 20 \text{ KHz}) - 1 = 1399 \text{ (or } 0x0577)$$

This experiment uses Pulse Width Modulation on the A3953 *ENABLE* pin and uses the *PHASE* pin for directional control. For simplicity, we will use the *Proportional* control for position. If the motor is within +/- 1 sector of the target, then we will cut the power to minimum. When +/- 1 to 4 sectors we will use increased power proportional to the distance from the target. At +/- 5 sectors, we will use 100% power to try to return to the target position.

For this experiment, 1 sector = 1/4 degree of rotation. Our gear motor has a 127:1 gear ratio, so it really won't move until there is more than about 60% duty cycle.

Observations

I tested various proportional methods. If I reduce power below about 55% duty cycle, there is no oscillation of phase due to bouncing back and forth across the target. At above 60%, the motor will oscillate at about 72 ms period (an audible vibration). The process loop is executing at 33 μ s; so the vibration is a function of the motor speed and the minimal sector size.



Experiment 4 - Proportional Control with Phase Modulation

This experiment uses a dsPIC30F4011 micro controller, 14.000 MHz crystal oscillator, Allegro A3953 Full Bridge Motor Controller, Pittman brushed DC gear motor, and 12V PC power supply.

The dsPIC30F4011 can run at up to 120 MHz.

$$F_{osc} = 14 \text{ MHz} * 8 \text{ pll} = 112 \text{ MHz}$$

$$F_{cy} = F_{osc}/4 = 28 \text{ MHz.}$$

PWM frequency of 20 KHz (above normal hearing frequencies) produces a period of 50 μ s. The dsPIC30F4011's PTPER register calculation for this period is

$$PTPER = ((F_{cy} * 1000) / f \text{ KHz}) - 1$$

$$PTPER\{20\text{KHz}\} = ((28 \text{ MHz} * 1000) / 20 \text{ KHz}) - 1 = 1399 \text{ (or } 0x0577)$$

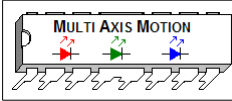
This experiment uses Phase Modulation on the A3953 *PHASE* pin and uses the *ENABLE* pin for on/off full power control. For simplicity, we will use the following *Proportional* control for position. If the motor is within +/- 1 sector of the target, then we will set the *PHASE* to (+/-)50/50% equal duty cycle. When +/- 1 to 4 sectors we will use change the *PHASE* duty cycle proportionally with respect to polarity and the distance from the target. At +/- 5 sectors, we will use 100% + or - polarity to try to return to the target position.

For this experiment, 1 sector = 1/4 degree of rotation. Our gear motor has a 127:1 gear ratio, so it really won't move until *PHASE* is more than about 60% either polarity.

Observations

I tested various proportional methods and experienced essentially the same results as using PWM control of the *ENABLE* pin for power and using the *PHASE* for directional control only. If I reduce power below about 55% duty cycle, there is no oscillation of phase due to bouncing back and forth across the target. At above 60%, the motor will oscillate at about 72 ms period (an audible vibration). The process loop is executing at 33 μ s; so the vibration is a function of the motor speed and the minimal sector size.

Ok, if I look at the voltage output charts, it is easy to see that they are almost identical for a given polarity. The *PHASE* chart is 1/2 scale in the vertical axis for Voltage and the left side is negative. Essentially the negative *PHASE* cancels the *ENABLE*'s positive PWM cycle; so strictly measuring the *ENABLE* voltage showed the left side of the 50% duty cycle to be null. It is null because with *ENABLE*, we are only measuring the positive polarity!



Experiment 5 - Enable and Phase Modulation

This experiment uses a dsPIC30F4011 micro controller, 14.000 MHz crystal oscillator, Allegro A3953 Full Bridge Motor Controller, Pittman brushed DC gear motor, and 12V PC power supply.

The dsPIC30F4011 can run at up to 120 MHz.

$$F_{osc} = 14 \text{ MHz} * 8 \text{ pll} = 112 \text{ MHz}$$

$$F_{cy} = F_{osc}/4 = 28 \text{ MHz.}$$

PWM frequency of 20 KHz (above normal hearing frequencies) produces a period of 50 μ s. The dsPIC30F4011's PTPER register calculation for this period is

$$PTPER = ((F_{cy} * 1000) / f \text{ KHz}) - 1$$

$$PTPER\{20\text{KHz}\} = ((28 \text{ MHz} * 1000) / 20 \text{ KHz}) - 1 = 1399 \text{ (or } 0x0577)$$

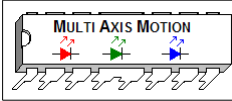
This experiment uses Phase Modulation on the A3953 *PHASE* pin and uses the *ENABLE* pin for on/off full power control. For simplicity, we will use the following *Proportional* control for position. If the motor is within +/- 1 sector of the target, then we will set the *PHASE* to (+/-)50/50% equal duty cycle. When +/- 1 to 4 sectors we will use change the *PHASE* duty cycle proportionally with respect to polarity and the distance from the target. At +/- 5 sectors, we will use 100% + or - polarity to try to return to the target position.

For this experiment, 1 sector = 1/4 degree of rotation. Our gear motor has a 127:1 gear ratio, so it really won't move until *ENABLE* and *PHASE* produces about 0.2V above or below 6V depending on polarity with respect to direction of rotation.

Observations

This experiment's results convinced me that the best approach to motor position control for the brushed DC motor used in this experiment is a combination of PWM on *ENABLE* and *PHASE*. I tried various proportional control methods and observed that matching the duty cycle between the *ENABLE* and *PHASE* works OK. The resultant output voltage can be easily maintained at 6V for position locking without oscillation.

The diagram on the first page of this document is a reasonable representation of PWM Duty Cycle VS Distance from Target Position. In this experiment, the endpoints are 100% duty cycle for *ENABLE* and the target position is 50%.



Glossary

The terminology in this glossary is defined as it is used in the context of this project document. In other words, don't quote these definitions for other works.

ENABLE - The On/Off switching control input to a motor controller. *ENABLE* is pulsed at varying frequencies and On/Off duty cycles to control average power.

PWM - In this document, *PWM* may refer to either Pulse Width Modulation or Phase Width Modulation. However, if the context is not specified as either Pulse or Phase, then *PWM* refers to the more common *Pulse Width Modulation*.