LENTARK

PWM-CORE

PP504F0A-02W30

FEATURES

- Adjustable 3 Output: P1, P2, P3
- Accurate Phase, Frequency and Duty Cycle.
- Frequency Range: 2 Hz – 500 KHz
  - Frequency Tolerance(Max) : ±%0,2
  - Frequency Stability (80ºC) : 100 ppm
- Tunable Period Range: 500,00ms – 120,00s
- Phase Range: 0,00º – 360,00º
- Duty Cycle Range: 0,00% - 100,00%
- Easy communication: USART (Rx, Tx)
- Automatic Recording Capability To Internal Memory.
- Wide Input Supply Voltage Operating Range: 5V-24V
- Low Output Voltage Swing: 0-125 µV

APPLICATIONS

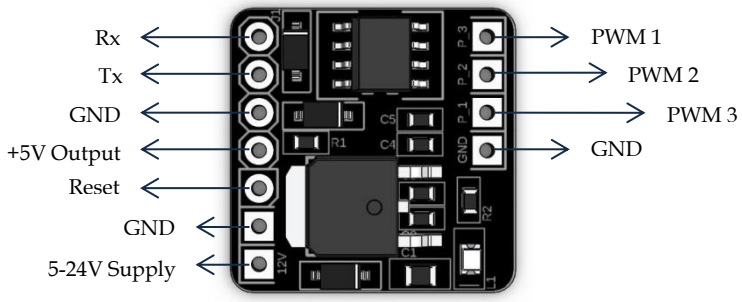
- P & N Channel MOSFET, IGBT Switching.
- H Bridge & Half Bridge Drivers.
- Switching Amplifiers.
- 3-Phase Applications.
- Motor Control.
- R,L And/Or C Load Control.
- Hobby.

GENERAL DESCRIPTION

PP504F0A-02W30 is a Pwm core with 5-24V input voltage and three channel signal output. The amplitudes of the signal outputs are at 5V level and the frequency, phase and duty cycle characteristics of all can be defined separately. It has a maximum tolerance of ±0.2% in the 2 Hz – 500 KHz band and can also be calibrated for applications in lower frequency bands.

The communication is quite convenient and simple. All control operations are provided with the codes to be sent to the Rx pin of PP504F0A-02W30. The **Protocols** are shown together with the examples below. The device keeps all signal configurations in memory. Thus, it does not need reconfiguration in every use.

DESCRIPTION WITH IMAGE



ELECTRICAL SPECIFICATIONS

Π Pushing the device to operate above the "Max." listed in the table below may cause the device to overheat and to take up permanent damage. It is inconclusive that the device will function beyond the operating limits as set out in this technical document. Prolonged exposure to work under "maximum" rating conditions may affect device reliability.

Table 1: *Electrical Specifications*

Conditions: Unless Otherwise Noted, $T_O = +25^oC$ and $9V \leq V_{IN} \leq 18V$.						
Parameters	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Condition**
Input						
Input Voltage	V_{IN}	5	12	24	V	DC
Input Current [No Load]	I_{IN}	5,26 6,51	9,29 11,23	9,48 11,37	mA	$f = 1\ \ Hz$ $f = 500\ KHZ$
Output						
Output Voltage, High	$V_{OUT,HIGH}$	4,200	5	5,100	V	$V_{IN} = 12V$
Output Voltage, Low	$V_{OUT,LOW}$	0	0	0,615	V	
Output Resistance, High	$R_{OUT,HIGH}$	–	–	200	Ω	
Output Resistance, Low	$R_{OUT,LOW}$	75	–	–	Ω	
Total Output Current*	I_{OT}	–	30	45	mA	
Switching						
Rise Time	t_R	–	15	32	ns	C_L = 0 pF
			30	65		C_L = 50 pF
Fall Time	t_F	–	15	30	ns	C_L = 0 pF
			30	60		C_L = 50 pF
Output Power Dissapation	W_{PD}	–	100	720	mW	**Note 1**

* **:** It refers to the total current that can be drawn from the signal outputs.

Note1 : It refers to the power that can be consumed in the sum of the signal outputs.

Tolerance & Sensitivity						
Parameters	**Sym**	**Min**	**Typ**	**Max**	**Units**	**Condition**
Frequency Tolerance *		–	0,05	0,2	% [percent]	500 ms – 2 µs
		–	–	5		120 s – 500 ms
Frequency Sensitivity		–	0,0312	–	µs	$T > 2$ µs
Duty Cycle Sensitivity (0,00 – 100,00)	D	0,01	–	0,01	% [percent]	$T > 312$ µs
		0,01	–	0,1		$T > 31$ µs
		0,1	–	1		$T < 31$ µs
Phase Angle Sensitivity (0,00 – 360,00)	P	0,01	–	0,01	o [degrees]	$T > 1.12$ ms
		0,01	–	0,1		$T > 112$ us
		0,1	–	1		$T < 112$ us

* **:** The period between 120 s - 500 ms can be calibrated manually. See "**Fast Examples Of Protocols**" to see examples.

PIN CONFIGURATION

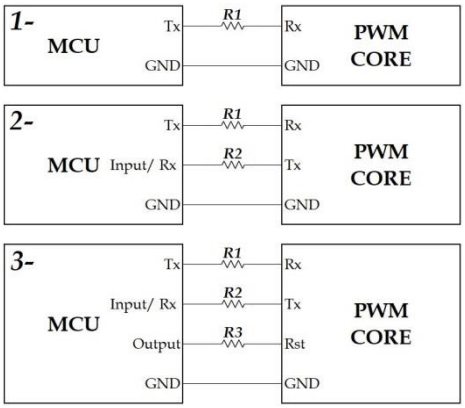
Pin	*Description*	*Notes*	*Types Of Connections*
Rx	Used to read data from any MCU. (9600 Baud Rate – 8 Bit Buffer)	You can terminate the connection after completing the signal configuration.	
Tx	This pin tells whether Pwm-core's Rx pin is ready to read new data. (1: Ready – 0: Busy)	You can connect to any MCU's Rx or Input pin. (0 ≤ R1,2,3 ≤ 470 Ω)	
Rst	Pwm-core's reset pin. (1:Active – 0:Passive)	You can connect to any MCU's Output pin and use Pwm-core as a reset or enable pin.	
+5V	5V Output	Max. 130 mA can be drain.	
12V	5-24V Supply Voltage	Nominal 12V	
GND	GND	–	
PWM1	Signal Output 1	For detailed information, see also **Electrical Characteristics** table.	
PWM2	Signal Output 2		
PWM3	Signal Output 3		

Figure 1: *Typical Connection Schemes*

Table 2: *Pin Configurations*

PROTOCOLS

Standard USART / UART communication rules apply. Make connections to PP504F0A-02W30 as specified in the **Pin Configuration** header. The string/Char* / Char[] variables to be sent from any MCU's Tx PIN must be sent in accordance with the following two rules:

Note: The device keeps all signal configurations in memory. Thus, it does not need reconfiguration in every use.

Rule 1 - Format:

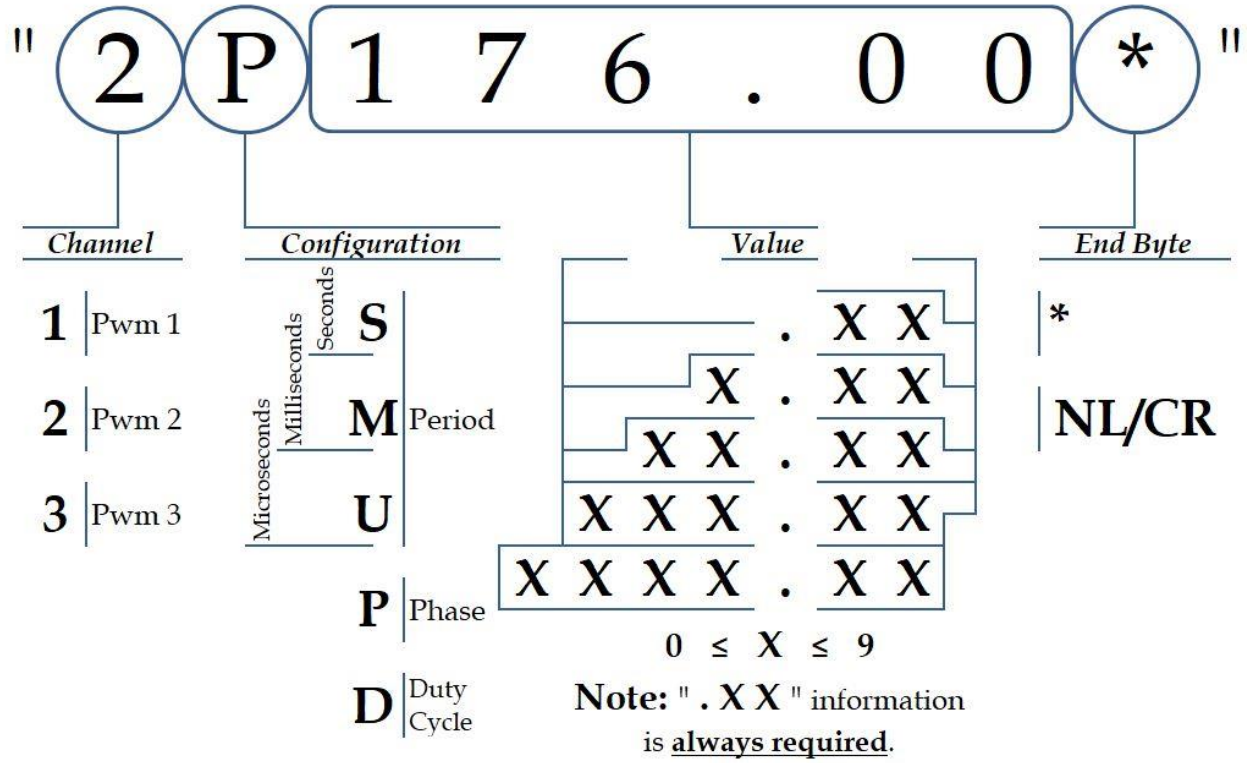


Figure 2: *Format*

Rule 2 – Protocol, Processing Time and Limits:

Protocol Information:

Mode:	Asynchronous
Baud Rate:	9600
Data Polarity:	Active-High
Rx Read Bits:	8 Bits

Limits:

	Lowest Value	Highest Value
S	0.50	120.00
M	0.01	500.00
U	2.00	2000.00
P	0.00	360.00
D	0.00	100.00

Processing_Time:

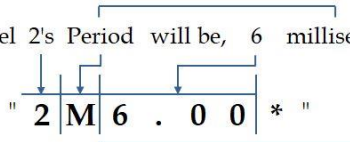
If Pwm-core's Tx[Rx_Rdy] pin is not to be used, Pwm-core's processing time must be taken into consideration in order to obtain the full and complete data sent, and the second data must be kept waiting while the first data is being processed. The processing times specified in **Table 3** show the highest and typical processing times determined. However, since each configuration goes through a different set of mathematical operations, it cannot be guaranteed that it will not exceed the specified processing times.

Configuration	*Typical*	*Max.*
Period	4.260 ms	4.875 ms
Phase	3.325 ms	3.491 ms
Duty Cycle	3.465 ms	3.515 ms

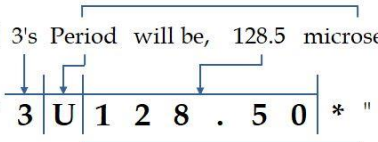
Table 3: *Processing Times*

Fast Examples Of Protocols:

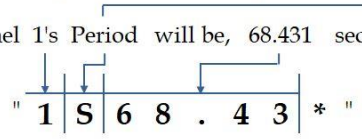
∏ If Channel 2's Period will be, 6 milliseconds;



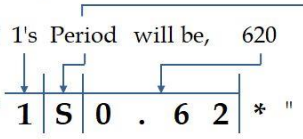
∏ If Channel 3's Period will be, 128.5 microseconds;



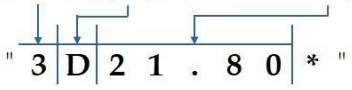
∏ If Channel 1's Period will be, 68.431 seconds;



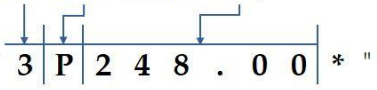
∏ If Channel 1's Period will be, 620 milliseconds;



∏ If Channel 3's Duty Cycle will be %21.8;



∏ If Channel 3's Phase will be 248°;



Examples For Coding:

Example – 1

```
#define Rx_Ready 10 //Any Input Pin
#include <SoftwareSerial.h>
SoftwareSerial _mySerial(10, 11);// RX, TX
//-----------Period Type--------------------------
// S: Seconds | M: Milliseconds | U: Microseconds
uint8_t channel_1_SMU_= 'U';
uint8_t channel_2_SMU_= 'U';
uint8_t channel_3_SMU_= 'U';

//{Integer Part, Floating Part}
//-----------Period-------------------------
uint16_t channel_1_period_[2]= {500, 5};
uint16_t channel_2_period_[2]= {100, 0};
uint16_t channel_3_period_[2]= {15, 38};
//-----------Phase--------------------------
uint16_t channel_1_phase_[2]= {0, 0};
uint16_t channel_2_phase_[2]= {120, 0};
uint16_t channel_3_phase_[2]= {240, 0};
//-----------Duty Cycle---------------------
uint16_t channel_1_duty_[2]= {25, 0};
uint16_t channel_2_duty_[2]= {25, 0};
uint16_t channel_3_duty_[2]= {50, 0};

void setup() {
Serial.begin(9600); while (!Serial) {;}
_mySerial.begin(9600); // Set Baud Rate
pinMode(Rx_Ready, INPUT); // Set Rx_Ready to input
digitalWrite(Rx_Ready, HIGH); // Turn on pull-up resistors
send_configuration(1, channel_1_SMU_, channel_1_period_);
send_configuration(2, channel_2_SMU_, channel_2_period_);
send_configuration(3, channel_3_SMU_, channel_3_period_);
send_configuration(1, 'P', channel_1_phase_);
send_configuration(2, 'P', channel_2_phase_);
send_configuration(3, 'P', channel_3_phase_);
send_configuration(1, 'D', channel_1_duty_);
send_configuration(2, 'D', channel_2_duty_);
send_configuration(3, 'D', channel_3_duty_);

}void loop() {}
void send_configuration (uint8_t _ch, uint8_t _conf,
uint16_t* _value){
char _buffer[12]; // Buffer
sprintf(_buffer,
"%d%c%d.%02d*\n\r",_ch,_conf,_value[0],_value[1] ); //
Prepare the buffer Note: \n\r is not necessary
while(!digitalRead(Rx_Ready)); // Wait for the Pwm-Core to
be ready
_mySerial.write(_buffer); // Send buffer to Pwm-Core
Serial.write(_buffer);}
```

Outputs:
1U500.05*
2U100.00*
3U15.38*
1P0.00*
2P120.00*
3P240.00*
1D25.00*
2D25.00*
3D50.00*

Example - 2

```
#define Rx_Ready 10              //Any Input Pin
#include <SoftwareSerial.h>
SoftwareSerial _mySerial(10, 11);// RX, TX

uint8_t  _channel  = 1;          // 1: Channel 1 | 2: Channel 2 | 3:
Channel 3
uint8_t  _config   = 'D';        // S: Seconds | M: Milliseconds | U:
Microseconds | P: Phase | D: Duty Cycle
uint16_t _period[2]= {2, 0};     // [Integer Part, Floating Part] Ex:
{1245,75}  => if _config == 'U' than Period = 1245.75 microseconds
uint16_t _phase[2] = {0,0};      // [Integer Part, Floating Part] Ex:
{60,0}     => if _config == 'P' than Phase  = 60.00 degree
uint16_t _duty[2]  = {50,0};     // [Integer Part, Floating Part] Ex:
{47,40}    => if _config == 'D' than Duty   = %47.40

void setup() {
  Serial.begin(9600); while (!Serial) {;}
  _mySerial.begin(9600);              // Set Baud Rate

  pinMode(Rx_Ready, INPUT);           // Set Rx_Ready
  digitalWrite(Rx_Ready, HIGH);       // Turn on pull-

  send_configuration(1, 'S', _period);
  send_configuration(2, 'S', _period);
  send_configuration(3, 'S', _period);
  send_configuration(1, 'P', _phase);_phase[0] += 120;
  send_configuration(2, 'P', _phase);_phase[0] += 120;
  send_configuration(3, 'P', _phase);_phase[0] += 0;
  send_configuration(1, 'D', _duty);
  send_configuration(2, 'D', _duty);
  send_configuration(3, 'D', _duty);
}void loop() {}

void send_configuration (uint8_t _ch, uint8_t _conf, uint16_t*
_value){
char buffer[12];      // Buffer
    sprintf(_buffer, "%d%c%d.%02d*\n\r",_ch,_conf,_value[0],_value[1]
);  // Prepare the buffer Note: \n\r is not necessary
while(!digitalRead(Rx_Ready));
 // Wait for the Pwm-Core to be ready
_mySerial.write(_buffer);
 // Send buffer to Pwm-Core
Serial.write(_buffer);}
```

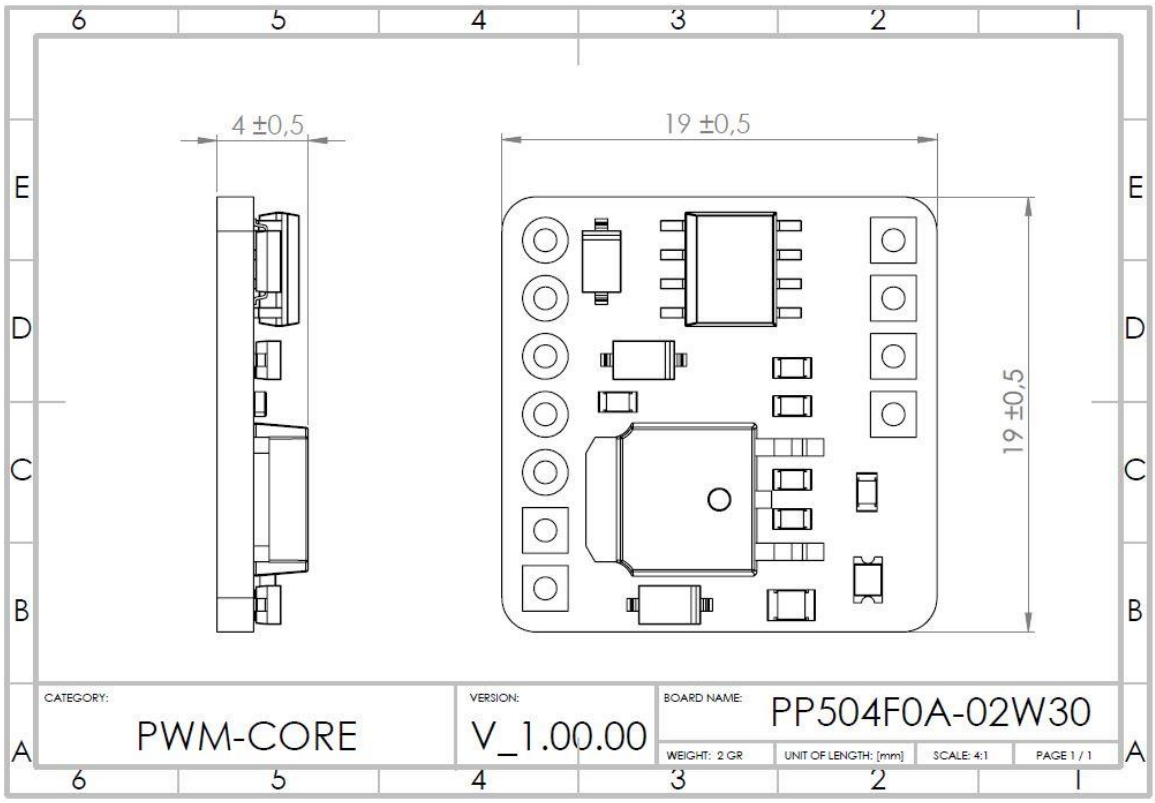
Outputs:
1S2.00*
2S2.00*
3S2.00*
1P0.00*
2P120.00*
3P240.00*
1D50.00*
2D50.00*
3D50.00*

PRODUCT CODE

LA	-	PC	-	P	P	504F	0A	-	02W	30

PRODUCT CATEGORY	SİGNAL TYPE	SİGNAL CONTROL	MAX. FREQUENCY	MAX. INSTANT CURRENT	POWER DIS. OF EACH OUTPUTS	OUTPUTS
MID = MOSFET/ IGBT DRIVER PC = PWM-CORE	P = PWM U = Triangle T = Sawthooth	S = Standart P = + Phase Shift O = + Ofset Match	xyzF = xy* 10^z Hertzs	xA = x Amps	xy = x,y Watts	xy = x Total Out. y Inverted Out.

TECHNICAL DRAWING



CONTACT INFORMATION

Lentark Electronics

Website :www.lentark.com
E-mail :info@lentark.com