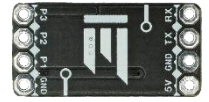
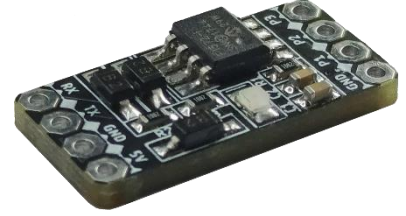


GENEL ÖZELLİKLER

- Π Programlanabilir 3 Pwm Çıkışı: P1, P2, P3
- Π Hassas Frekans ve Çalışma Oranı Kalibrasyonu.
- Π Frekans Aralığı: 4 Hz - 500 KHz
 - Frekans Toleransı (Maks.): $\pm 0,2$
 - Frekans Kararlılığı (80°C): 100 ppm
- Π Kalibre Edilebilir Periyot Aralığı: 250,00ms - 120,00s
- Π Çalışma Oranı Aralığı: %0,00 - %100,00
- Π Kolay İletişim: UART (Rx, Tx)
- Π Programlanabilir Baud Rate: 9600 / 19200 / 57600 / 115200
- Π Dâhili Hafızaya Kayıt Yeteneği.
- Π Giriş Besleme Gerilimi: 5V
- Π Düşük Voltaj Salınımı: 0 - 125 μ V



UYGULAMA ALANLARI

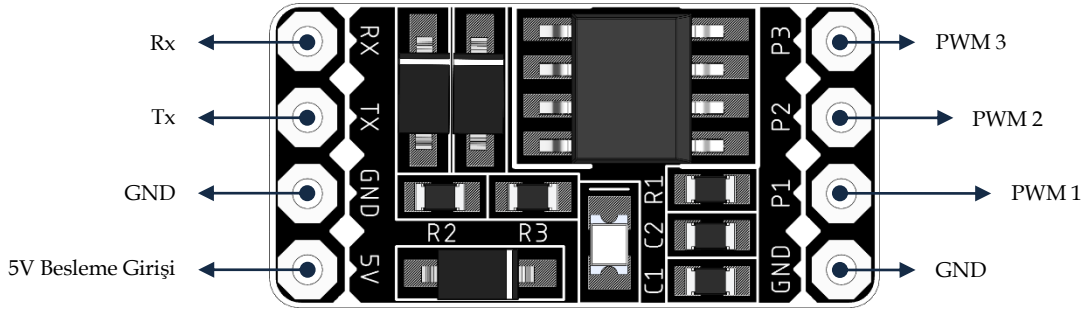
- Π MOSFET, IGBT Tetikleme Uygulamaları.
- Π Aydınlatma Uygulamaları.
- Π Motor Kontrol Uygulamaları.
- Π Ses Uygulamaları.
- Π 16 Bit PWM.
- Π R, L ve/veya C yük Kontrol Uygulamaları.
- Π Hobi Uygulamaları.

GENEL AÇIKLAMA

PS504F0A-02W30, 5V giriş gerilimine ve üç kanal sinyal çıkışına sahip bir PWM çekirdeğidir. PWM kanallarının **frekans, çalışma oranı özellikleri ayrı ayrı tanımlanabilir**. Sinyal çıkışlarının genlikleri 5V seviyesindedir. 4 Hz - 500 KHz bandında en fazla $\pm 0,2$ toleransa ve ayrıca daha düşük frekans bantlarındaki çalışmalar için kalibre edilebilme özelliğine sahiptir.

İletişim mekanizması oldukça kullanışlı ve basittir. Kontrol işlemlerinin tamamı PS504F0A-02W30'nun Rx pinine gelecek olan kodlarla sağlanır. İletişim hızı programlanabilir. **İletişim kuralları** aşağıda örneklerle birlikte gösterilmiştir. Cihaz tüm sinyal yapılandırmalarını dahili hafızasına kaydedebilmektedir. Böylece, cihazın her yeniden başlatılması durumundaki istenilen başlangıç değerleri belirlenebilir.

PİN ÇIKIŞLARI



ELEKTRİKSEL KARAKTERİSTİKLERİ

Π Aşağıdaki tabloda “En Fazla” olarak belirtilen seviyelerin üstünde cihazı çalışmaya zorlamak cihazın ısınmasına ve kalıcı hasar almasına sebep olabilir. Cihazın, bu teknik belgede belirtilen çalışma sınırlarının dışında işlevsel çalışması düşünülemez. Uzun süre “En Fazla” derecelendirme koşullarında çalışmaya maruz kalma cihaz güvenilirliğini etkileyebilir.

Tablo 1: Elektriksel Karakteristikler.

Koşullar: Aksi Belirtilmedikçe, $T_O = +25^{\circ}C$ ve $V_{IN} = 5V$.						
Parametreler	Sembol	En Az	Norm.	En Fazla	Birim	Test Şartı
Giriş						
Giriş Voltajı	V_{IN}	4,5	5	5,5	V	DC
Giriş Akımı [Yüksüz]	I_{IN}	–	2,24	–	mA	$f = 1 \text{ Hz}$ $f = 500 \text{ KHZ}$
Çıkış						
Yüksek Çıkış Voltajı	$V_{OUT,HIGH}$	4,200	5	5,100	V	$V_{IN} = 5V$
Düşük Çıkış Voltajı	$V_{OUT,LOW}$	0	0	0,615	V	
Yüksek Voltaj Çıkış Direnci	$R_{OUT,HIGH}$	–	–	200	Ω	
Düşük Voltaj Çıkış Direnci	$R_{OUT,LOW}$	75	–	–	Ω	
Toplam Çıkış Akımı*	I_{OT}	–	30	45	mA	
Tetikleme						
Yükselme Zamanı	t_R	–	15	32	ns	$C_L = 0 \text{ pF}$
			30	65		$C_L = 50 \text{ pF}$
Düşme Zamanı	t_F	–	15	30	ns	$C_L = 0 \text{ pF}$
			30	60		$C_L = 50 \text{ pF}$
Çıkış Güç Tüketimi	W_{PD}	–	100	720	mW	Not 1

* : Sinyal çıkışlarından çekilebilecek toplam akımı ifade etmektedir.

Not1 : Sinyal çıkışlarının toplamında tüketilebilecek gücü ifade etmektedir.

Tolerans & Hassasiyet							
Parametreler	Sembol	En Az	Norm.	En Fazla	Birim	Test Şartı	
Frekans Toleransı*		–	0,05	0,2	%	250 ms – 2 µs	
		–	–	5		[yüzde]	120 s – 250 ms
Frekans Hassasiyeti		–	0,0312	–	µs	$T > 2 \mu s$	
Çalışma Oranı Hassasiyeti (0,00 – 100,00)	D	0,01	–	0,01	%	$T > 312 \mu s$	
		0,01	–	0,1		[yüzde]	$T > 31 \mu s$
		0,1	–	1			$T < 31 \mu s$

* : 120 s – 250 ms arasındaki periyot değerleri manuel olarak kalibre edilebilmektedir. Örnekleri görmek için “İletişim Kuralları Hızlı Örnekler” başlığına bakınız.

PIN AÇIKLAMALARI

Pin	Açıklama	Notlar	Bağlantı Çeşitleri
Rx	Herhangi bir MCU'dan veri okumak için kullanılır. (9600 Baud Rate – 8 Bit Buffer)	Herhangi MCU'nun Tx portuna bağlanır.	
Tx	Bu pin, Pwm-Core'un Rx pininin yeni veri okumaya hazır olup olmadığı bilgisini verir. (1: Hazır – 0: Meşgul)	Herhangi MCU'nun Rx ya da Input portuna bağlanır. ($0 \leq R_{1,2} \leq 470 \Omega$)	
GND	GND	–	
5V	5V Besleme Girişi	–	
PWM1	1.Sinyal Çıkışı	Detaylı bilgi için Elektriksel Karakteristikler tablosuna bakınız.	
PWM2	2.Sinyal Çıkışı		
PWM3	3.Sinyal Çıkışı		

Tablo 2: Pin Açıklamaları.

Cihaz PWM kanallarını asenkron olarak çalıştırmaktadır. Bu, PWM kanalları arasında faz farkı aranmaması anlamına gelir. Ancak asenkron çalışma, dinamik olarak değer atama, sürekli (kesinti veya yeniden başlama olmaksızın) PWM sağlama imkânı sağlamaktadır. Bu sayede herhangi bir PWM kanalının parametresinin değiştirilmesi, diğer kanallarda işlem yapılmasını gerektirmez ve yeni değerler ilgili PWM kanalının periyodunun tamamlanması ile birlikte PWM kanalına yazılır.

PWM kanallarının mevcut değerlerinin kaydedilmesi komut ile gerçekleşir. Tablo 3'de açıklandığı şekilde, eğer komut dizisinin sonunda '/' karakteri bulunuyor ise gelen komut değeri de dahil olmak üzere tüm mevcut değerlerin kaydı gerçekleşir. Bu sayede cihazın yeniden başlatılması durumunda başlangıç değerleri belirlenebilir.

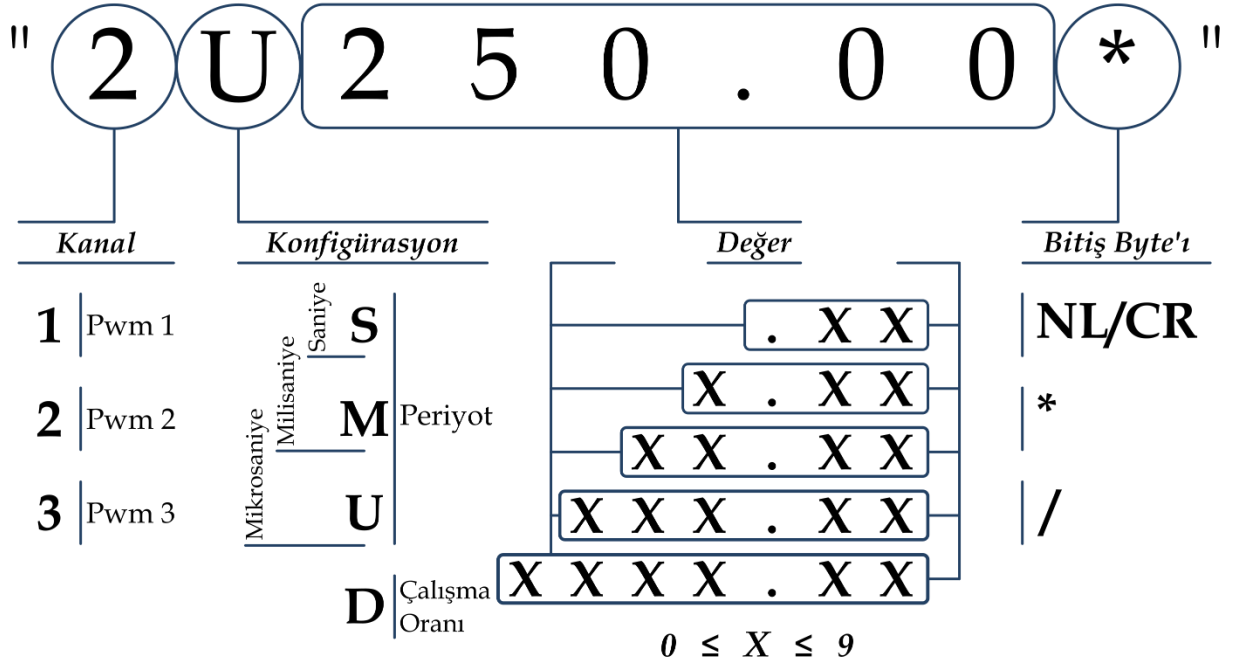
PWM kanalının periyodunun tamamlanması ile birlikte yazılmayı bekleyen değer varken, bu değeri değiştirecek yeni komutların gelmesi durumunda en son gelen komut PWM kanalına yazılır. Örneğin; 2. PWM kanalının periyodu 500 µs ve çalışma oranı %24'dür. Bu kanalın PWM periyodu tamamlanmadan önce kanalın çalışma oranını sırasıyla %24.15 , %24.3 , %24.45 olacak şekilde değiştirecek komutlar gönderilmesi durumunda, periyot tamamlanmadan önce en son gönderilen değer PWM kanalına yazılır. Periyot komutları ve çalışma oranı komutları ile oluşan yeni değerler ayrı ayrı hafızada tutulur. Örneğin; 3. Kanalın periyodunun tamamlanmasından önce bu kanalın hem periyodunu hem de çalışma oranını değiştirecek komutlar gönderilmesi durumunda kanalın periyodu tamamlandığında en son gönderilen çalışma oranı değeri ve en son gönderilen periyot değeri PWM kanalına aynı anda yazılır.

İLETİŞİM KURALLARI

Pwm-Core'un bağlantıları **Pin Açıklamaları** başlığında belirtildiği gibi gerçekleştirilmelidir. Komut gönderecek olan MCU'nun UART iletişim protokolü **Protokol Bilgileri** başlığında verilen bilgilere göre yapılandırılmalı ve gönderilecek komut için, String/Char*/Char[] değişkeni aşağıdaki iki kurala uygun olarak hazırlanmalıdır.

Not: Cihaz tüm sinyal yapılandırmalarını hafızasında tutabilmektedir. Böylece her kullanımda tekrar yapılandırmaya ihtiyaç duymamaktadır. Bk. **Tablo 3**.

Kural 1 - Format:



Not: ".XX" bilgisi her zaman gereklidir.

Şekil 2: İletişim Kuralları.

Tablo 3: Bitiş Baytlarının Açıklanması.

Bitiş Baytları	Açıklama	Örnekler
** veya NL/CR	Komut tamamlama baytıdır. Gönderilen komut işlenir ve ilgili PWM kanalının periyodunu tamamlamasından sonra PWM kanalına yeni değer yazılır.	1M1.23* 2D74.11*
/'	Komut tamamlama ve kaydetme baytıdır. Gönderilen komut işlenir, tüm değerler hafızaya kaydedilir ve ilgili PWM kanalının periyodunu tamamlamasından sonra PWM kanalına yeni değer yazılır.	1U100.10/ 2D47.25/ 3U720.00/

Tablo 4: Baud Rate Programlama için Gerekli Komutların Gösterilmesi.

Konfigürasyon	Açıklama	Değerler
B	Cihaz varsayılan olarak 9600 Baud Rate hızındadır. Yüksek iletişim hızlarında çalışma istediğinde Baud Rate programlamak için bu konfigürasyon kullanılır.	1B4.00/ 9600 Baud Rate [Varsayılan] 1B3.00/ 19200 Baud Rate 1B2.00/ 57600 Baud Rate 1B1.00/ 115200 Baud Rate

Kural 2 - Protokol, İşlem Süresi ve Sınırlar:

Protokol Bilgileri:

Mod: Asenkron
Baud Rate: 9600 / 19200 / 57600 / 115200
Veri Polaritesi: Aktif-Yüksek
Rx Alış Bitleri: 8 Bit

Sınırlar:

	En Düşük Değer	En Yüksek Değer
S	0 . 5 0	1 2 0 . 0 0
M	0 . 0 1	5 0 0 . 0 0
U	2 . 0 0	2 0 0 0 . 0 0
D	0 . 0 0	1 0 0 . 0 0

İşlem Süresi:

Eğer Pwm-Core'un Tx[Rx_Rdy] pini **kullanılmayacak** ise gönderilen verinin tam ve eksiksiz alınabilmesi için mutlaka Pwm-Core'un işlem süresi göz önünde bulundurulmalıdır, ilk gönderilen verinin işlemleri sürerken ikinci veri bekletilmelidir. **Tablo 5'**de belirtilen süreler, belirlenen en yüksek ve tipik işlem sürelerini göstermektedir. Ancak her konfigürasyon farklı bir dizi matematiksel işlemden geçtiği için belirtilen işlem sürelerinin dışına çıkmayacağı garanti edilemez.

Konfigürasyon	Tipik	En Fazla
Periyot	420 µs	770 µs
Çalışma Oranı	230 µs	280 µs
+ Kayıt Komutu	+ 3,25 ms	+ 3,35 ms

Tablo 5: İşlem Süreleri.

İletişim Kuralları Hızlı Örnekler:

II 1. Kanalda, Periyot 6 milisaniye olacaksa;

" 1 | M | 6 . 0 0 | * "

II 2. Kanalda, Periyot 68.431 saniye olacaksa;

" 2 | S | 6 8 . 4 3 | * "

II 3. Kanalda, Periyot 2 milisaniye olacaksa;

" 3 | M | 2 . 0 0 | * "

II 1. Kanalda, Çalışma Oranı %21.8 olacaksa;

" 1 | D | 2 1 . 8 0 | * "

II 2. Kanalda, Çalışma Oranı %11.1 olacaksa;

" 2 | D | 1 1 . 1 1 | * "

II 3. Kanalda, Çalışma Oranı %77.05 olacaksa;

" 3 | D | 7 7 . 0 5 | * "

II 1. Kanalda, Periyot 128.5 mikrosaniye olacaksa;

" 1 | U | 1 2 8 . 5 0 | * "

II 2. Kanalda, Periyot 620 milisaniye olacaksa;

" 2 | S | 0 . 6 2 | * "

II 3. Kanalda, Periyot 774.05 mikrosaniye olacaksa;

" 3 | U | 7 7 4 . 0 5 | * "

II 1. Kanalda, Çalışma Oranı %47.05 olacaksa;

" 1 | D | 4 7 . 0 5 | * "

II 2. Kanalda, Çalışma Oranı %0 olacaksa;

" 2 | D | 0 . 0 0 | * "

II 3. Kanalda, Çalışma Oranı %100 olacaksa;

" 3 | D | 1 0 0 . 0 0 | * "

Kodlama İçin Örnek:

```
//PS504F0A-02W30
#define Rx_Ready 10 //Any Input Pin
#include <SoftwareSerial.h>
SoftwareSerial _mySerial(10, 11); // RX, TX

//-----Baud Rate-----
//Baud Rate: 4 -> 9600 | 3 -> 19200 | 2 -> 57600 | 1 -> 115200
uint16_t cong_Baud [2]= {4, 0};
//-----Period Type-----
// S: Seconds | M: Milliseconds | U: Microseconds
uint8_t channel_1_SMU = 'M';
uint8_t channel_2_SMU = 'M';
uint8_t channel_3_SMU = 'U';
//-----Period-----
//{Integer Part, Floating Part}
//Exp: {245,75} => if _config == 'U' than Period = 245.75 microseconds
uint16_t channel_1_period [2]= {5, 0};
uint16_t channel_2_period [2]= {5, 0};
uint16_t channel_3_period [2]= {500, 0};
//-----Duty Cycle-----
uint16_t channel_1_duty [2]= {50, 0};
uint16_t channel_2_duty [2]= {50, 0};
uint16_t channel_3_duty [2]= {0, 0};

void setup() {
  Serial.begin(9600); while (!Serial) {}
  _mySerial.begin(9600); // Set Baud Rate
  pinMode(Rx_Ready, INPUT_PULLUP); // Set Rx_Ready to input & Turn on pull-up resistors
  //send_configuration(1, 'B', cong_Baud, '/');
  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, 'D', channel_1_duty, '*');
  send_configuration(2, 'D', channel_2_duty, '*');
  send_configuration(3, 'D', channel_3_duty, '/');
}

void loop() {

  /*-----*Channel 1*-----
  //0.1% Duty each Cycle
  if((channel_1_duty [1]+10) >=90) {channel_1_duty [1] = 0;
   if(channel_1_duty [0]++ >=99) channel_1_duty [0] = 0;
  }
  send_configuration(1, 'D', channel_1_duty, '*');
  while(!digitalRead(Rx_Ready)); // Wait for the Pwm-Core to be ready

  //10 micro seconds each cycle.
  if(channel_1_SMU == 'U'){
  if((channel_1_period [0]+10) >=2000 && channel_1_SMU == 'U') {channel_1_period [0] = 2;channel_1_SMU = 'M';}}

  if(channel_1_SMU == 'M'){
  if(channel_1_period [1]++ >=99) {channel_1_period [1] = 0;
  if(channel_1_period [0]++ >=9 ) {channel_1_period [0] = 500;channel_1_SMU = 'U';}}
  }
  send_configuration(1, channel_1_SMU, channel_1_period, '*'); */

  /*-----*Channel 3*-----
  //0.1% Duty each Cycle
  if(channel_3_duty [0]++ >=99) channel_3_duty [0] = 0;
  send_configuration(3, 'D', channel_3_duty, '*');
  while(!digitalRead(Rx_Ready)); // Wait for the Pwm-Core to be ready

  //10 micro seconds each cycle.
  if(channel_3_SMU == 'U'){
  if((channel_3_period [0]+10) >=2000 && channel_3_SMU == 'U') {channel_3_period [0] = 2;channel_3_SMU = 'M';}}

  if(channel_3_SMU == 'M'){
  if(channel_3_period [1]++ >=99) {channel_3_period [1] = 0;
  if(channel_3_period [0]++ >=9 ) {channel_3_period [0] = 500;channel_3_SMU = 'U';}}
  }
  send_configuration(3, channel_3_SMU, channel_3_period, '*'); */

  if (Serial.available()) {int a = Serial.read();_mySerial.write(a);Serial.write(a);}
}

void send_configuration (uint8_t_ch, uint8_t_conf, uint16_t* _value, uint8_t_end)
{char _buffer[12]; // Buffer
// Prepare the buffer Note: \n\r is not necessary
sprintf( _buffer, "%d%c%d.%02d%c\n\r", ch, conf, value[0], value[1], end);
while(!digitalRead(Rx_Ready)); delayMicroseconds(50); // Wait for the Pwm-Core to be ready
_mySerial.write( _buffer); // Send buffer to Pwm-Core
Serial.write( _buffer);
}

// End of CODE
// Outputs:
// 1M5.00*
// 2M5.00*
// 3U500.00*
// 1D50.00*
// 2D50.00*
// 3D0.00/
```

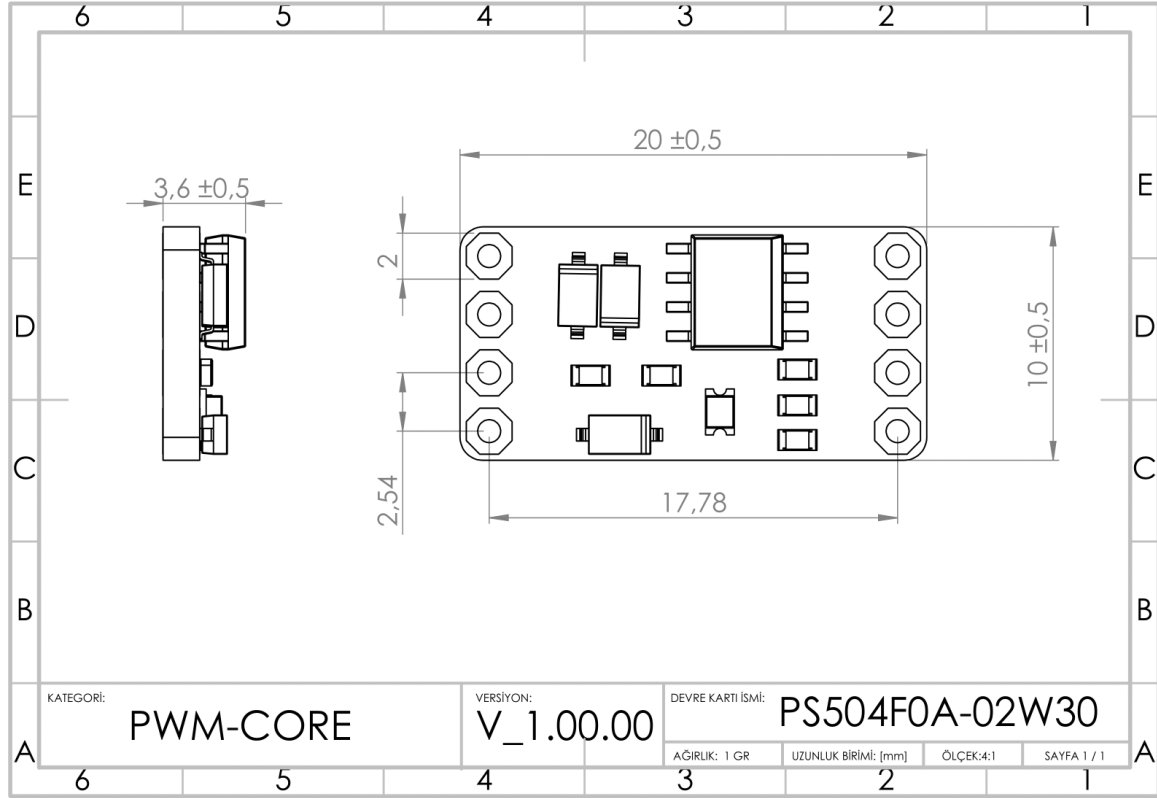
ÜRÜN KODU

PC	-	P	S	504F	0A	-	02W	30	ÇIKIŞLAR $xy = x : 3$ Toplam & $y : 0$ Terslenmiş Çıkış.
									GÜÇ TÜKETİMİ $xyW = x,y : 0.2$ Watt Toplam Çıkış Gücü.
									ANLIK AKIM $x A = x : 0$ Amper Anlık Akım Her Bir Çıkış İçin.
									MAKS. FREKANS $xyzF = xy*10^z : 500$ KHz Maksimum Frekans.
									SİNYAL KONTROLÜ S : Frekans, Çalışma Oranı Kontrolü. P : Frekans, Çalışma Oranı, Faz Açısı Kontrolü.
									SİNYAL TİPİ P : PWM SPWM : Sinüzoidal PWM

KATEGORİ

GD : Gate Sürücüler
PC : Pwm-Çekirdeği
MM : Mini-Multimetre
DD : DC-DC Çevirici
MIS : Çeşitli

TEKNİK ÇİZİM



İLETİŞİM BİLGİLERİ

Lentark Elektronik

Web Sitesi : www.lentark.com

E-posta : info@lentark.com