



PRONEM-midi MODBUS RTU İÇİN REGISTER ADRESLERİ

EMKO ELEKTRONİK A.Ş.
Demirtaş Org. San. Bölğ.
Karanfil Sk. No: 6
16369 Bursa / TURKEY

Telefon : +90 224 261 1900
Fax : +90 224 261 1912
Url : www.emkoelektronik.com.tr
e-mail : emko@emkoelektronik.com.tr
exposales@emkoelektronik.com.tr

1. Giriş

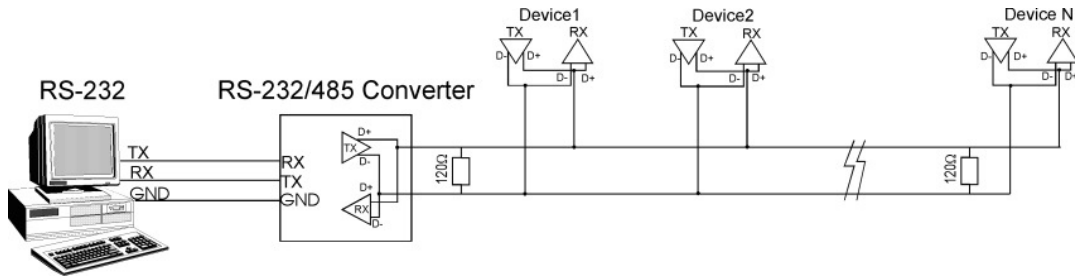
Bu döküman PRONEM-midi cihazı üzerinden Modbus® protokolü kullanılarak RS-232 ve RS-485 haberleşmesinin nasıl yapıldığını tanımlar.
PRONEM-midi cihazında Modbus® RTU protokolü kullanılmıştır.

2. Cihazı Bus'a Bağlama

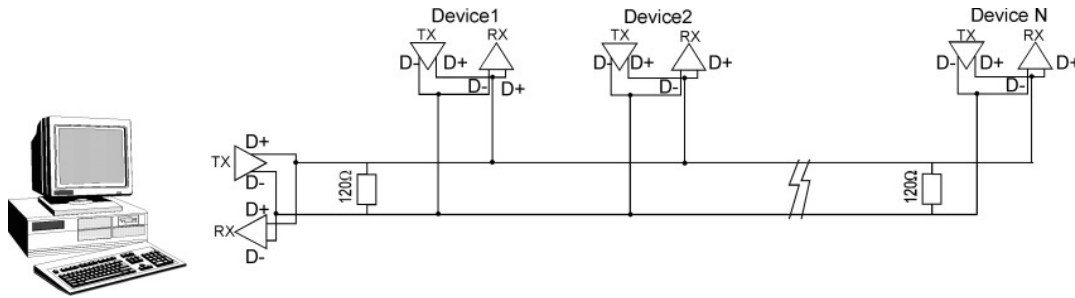
Not : Bus'a bağlanılan her cihazın kendine ait bir adresi olmalıdır.

Tüm cihazların mod (RTU), baudrate, parity ve stop bit ayarları aynı olmalıdır.

RS-485 haberleşmesinde gürültü sebebiyle yaşanabilecek sinyal hatalarına karşı bükümlü kablo kullanımı tavsiye edilir. Uzun mesafelerde yaşanan kablo yansımalarını indirmek için RS-485 sistemleri hat sonlandırmasına ihtiyaç duyarlar. Bu işlem iki adet 120 Ω sonlandırma direnciyle sağlanır. Sonlandırma dirençlerinin biri hattın başında diğeri ise sonunda kullanılmalıdır.

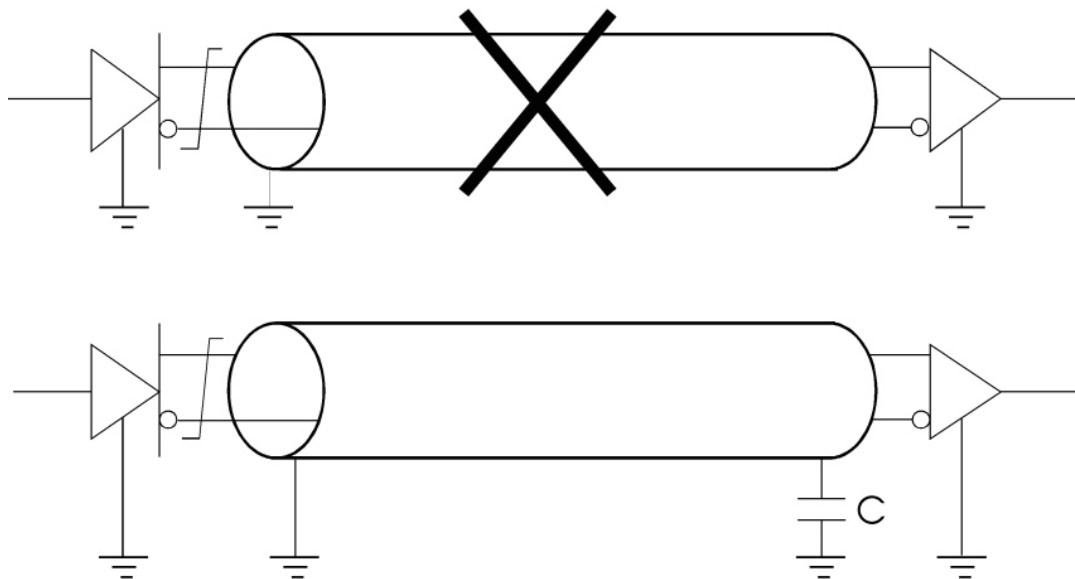


RS-232/485 çevirici.



RS-485 arayüzü.

SHIELDING of the CABLE



Blendaj her iki taraftan toprağa bağlandığı zaman yüksek frekans parazitlerini de engellenmiş olur. Topraklama hattına seri bağlanan C kapasitörü ise sızıntı akımlarını engeller.

3. Modbus® Protokolü

Haberleşme protokolü, sistemdeki tüm cihazlar tarafından bilinen komutları ve data formatlarını kullanır. Modbus®, tüm işlemlerin tek bir sağlayıcı (örn PC) tarafından başlatıldığı bir master(yöneten)-slave(yönetilen) protokolüdür. Mesaj paketleri; cihaz adresi, komut, data ve hataları algılamak için bir checksum içerir. Tüm slave cihazlar sürekli olarak hatta gelen mesajın başlangıcını dinler. Mesaj paketleri tüm slave cihazlar tarafından algılanır fakat sadece sorgudaki adresle eşleşen cihaz cevap verir, diğerleri mesajı görmezden gelir.

3.1.Modbus® İletim Modları

3.1.1.Özellikler

Arayüz	: RS-232 ve RS-485
İletişim Sistemi	: Half Duplex
Senkronizasyon Sistemi	: Start-stop senkronizasyonu
Data Uzunluğu	: 8 bit
Parity	: Yok, tek, çift
Stop Biti	: 1, 2 stop biti
İletim Hızı	: 9600, 19200, 38400, 57600 ve 115200 kbps
İletim Kablosu	: Blendajlı bükümlü kablo
Hata Yakalama Teknikleri	: 1. Parity kontrolü: Yok / Tek / Çift parity 2. Cyclic redundancy checks (CRC)

3.1.2.Fonksiyon Kodları

Fonksiyon Kodu 03 : Read Holding Register

Fonksiyon Kodu 04 : Read Input Registers

Fonksiyon Kodu 06 : Write Single Register

3.2. Modbus® Mesaj Formatı

Modbus® mesajları bir format dahilinde iletilir. Mesajın başlangıcı ve bitişi slave tarafından bilinir.

Slave cihazları başlangıç karakterini algıladıkları anda, mesajdaki adres bölgesini okuyup sorgunun kendilerine mi yapıldığına karar verirler. Aynı zamanda mesajın tamamlandığını da algılayabilirler. Eğer mesaj tamamlanmamışsa hata durumları meydana gelir.

3.2.1. RTU Formatı

Bu modda mesajlar en az 3,5 karakter süresi kadar boşluk gönderilerek başlatılır. Ağdaki cihazlar bu boşluk aralığını bekler. Bu boşluktan sonra cihaz adresi gönderilir. Mesaj alındıktan sonra tüm cihazlar o mesajın kendilerine ait olup olmadığını kontrol eder. Son karakter gönderildikten sonra 3,5 karakter süresi kadar boşluk gönderilerek mesajın bittiği bilgisi verilir. Yeni mesaj bu aralıktan sonra başlar.

Mesaj paketi sürekli olarak gönderilmelidir. Eğer mesaj tamamlanmadan önce 1,5 karakter süresi kadar boşluk oluşursa, cihaz alınan mesajı yok sayar ve bir sonraki byte'ı yeni bir mesajın adres alanı olarak algılar. Eğer 3,5 karakter süresi kadar boşluktan önce yeni bir mesaj başlarsa, yeni mesaj eski mesajın devamı olarak algılanır ve sonrasında CRC hatası oluşur.

3.2.2. Adres Alanı

RTU modunda adres alanı 8 bittir. Slave cihazın adresi 1 ile 247 arasında olabilir.

3.2.3. Fonksiyon Alanı

RTU modunda fonksiyon alanı 8 bittir. Fonksiyon alanı, slave cihaza ne yapacağını bildirir. Eğer herhangi bir hata yoksa slave aynı fonksiyon kodunu geri döndürür ancak bir hata durumu belirmişse, slave cihazı fonksiyon kodunun en yüksek değerlikli bitini (MSB) lojik 1 yapıp fonksiyon kodunu geri döndürür. Hata kodları açıklanacaktır.

Örneğin master register'ların bir grubunu okumak istediğinde aşağıdaki fonksiyon kodunu gönderecektir:

0000 0011 (Hex 03)

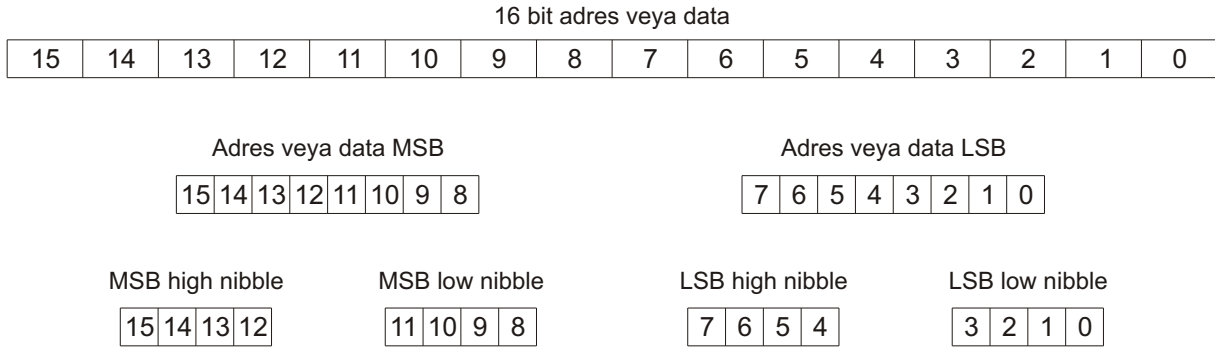
Eğer slave mesajı hatasız alırsa, aynı fonksiyon kodunu geri dönecektir, ancak bir hata durumu belirirse fonksiyon kodu aşağıdaki gibi olacaktır:

1000 0011 (Hex 83)

3.2.4. Data Alanı

Bu alan register adreslerini, kaç byte okunacağını ve okunacak byte'ların sayısını içerir. Örneğin master bir grup registerı okumak istiyor. Data alanı; register adresini, okumaya nereden başlayacağını ve kaç register okunacağını içerir.

3.3. RTU Modu



Not : Cihaza bir data gönderileceği veya okunacağı zaman register adres değerinden 1 çıkarılmalıdır. Örneğin 15 numaralı register adresini okumak istiyorsanız, cihaza register adresi olarak 14 yollamalısınız.

Modbus® ağında RTU modu kullanılarak haberleşme yapılacağı zaman, bir mesajdaki her 8-bit byte iki adet 4-bit hexadecimal karakter içerir. Her mesaj sürekli bir biçimde iletilmelidir.

RTU modunun formatı:

Kodlama Sistemi : 8 bit binary, hexadecimal 0-9,A-F

Hata Kontrolü: Cyclical Redundancy Check (CRC)

3.3.1 CRC Hesaplama

16-bit CRC alanı mesajın sonuna eklenir. CRC, mesaj içeriğine bağlı olarak hesaplanır. Slave cihazı CRC kodunu tekrar hesaplar ve gelen mesajın içerdiği CRC koduyla karşılaştırır. Eğer bu iki değer birbirine eşit değilse bir hata durumu doğar ve slave cihazı gelen mesajı dikkate almaz.

En basit yöntem; hesaplamanın sonunda CRC içeriğinin düşük ve yüksek byte'larının yer değiştirilmesidir. Bu yöntem aşağıda gösterildiği gibidir.

- 1- 16-bitlik register'a (CRC Register'ı) FFFF Hex yüklenir.
- 2 - CRC register'ının düşük değerlikli byte'ı ile mesajın ilk 8 biti Özel-VEYA işlemine tabii tutulur. Sonuç CRC register'ına yazılır.
- 3 - CRC register'ı 1 bit sağa kaydırılır (2'ye bölünür), MSB'ye sıfır yazılır.
- 4 - Sağa kaydırılan bit 1 ise, CRC registerı A001 Hex değeri ile ÖZEL-VEYA işlemine tabii tutulur.
- 5 - 8 bit sağa kaydırılınca kadar 3. ve 4. adımlar tekrarlanır. Böylece 1 byte değer işlenir.
- 6 - Tüm byte'lar işlenilene kadar 2. ve 5. adımlar tekrar edilir.
- 7 - CRC register'ının içeriği yüksek değerlikli byte'dan başlanarak mesajın sonuna yerleştirilir.
- 8 - Tamsayı sonucunun yüksek değerlikli byte'ı ile düşük değerlikli byte'ı yer değiştirilir.

Uygulamanın C kodu aşağıda verilmiştir.

```
unsigned int check_sum(unsigned char *buff, char start, char bytes)
{
    Char byte_cnt, bit_cnt; /* loop counters */
    unsigned int crc_reg; /* Result register */
    unsigned int CRCHi, CRCLo; /* Low and high order bytes of the crc */
    /* Set the CRC register to all 1's */ crc_reg = 0xFFFF;
    /* Repeat for each byte of sub string */
    for(byte_cnt=start; byte_cnt<(bytes+start); byte_cnt++)
    {
        crc_reg = crc_reg ^ (Unsigned int)buff[byte_cnt]; /*EXOR CRC &Next Byte*/
        /* Test each bit of the CRC */
        for(bit_cnt=0; bit_cnt<8; bit_cnt++)
        {
            if(crc_reg & 0x0001)
            {
                crc_reg = crc_reg >>1; /* IF LSB=1 EXOR
                CRC with A001H*/
                crc_reg = crc_reg ^ 0Xa001; /* Then shift
                CRC toward LSB */
            }
            else crc_reg = crc_reg >>1; /* ELSE Shift CRC towards LSB */
        }
    }
    CRCLo=crc_reg>>8; /*Swap the low and high order bytes of the crc result*/
    CRCHi=crc_reg<<8;
    crc_reg = CRCLo+CRCHi;
    return crc_reg; /*Final CRC register Result */
}
```

3.4 Modbus İstisnaları

* Eğer slave iletişim hatasından dolayı sorguyu alamazsa, cevap döndürmez . (Timeout Hatası)

* Eğer slave sorguyu alır ancak iletişim hatası algıarsa (parity, CRC), cevap döndürmez. (Timeout Hatası)

* Eğer slave herhangi bir iletişim hatası olmadan sorguyu alır ancak beklenmeyen bir durumla karşılaşır (örneğin master var olmayan bir register okuma talebinde bulunursa), slave istisna yanıtlarından birini gönderecektir.

Hata kodları aşağıdaki gibidir:

01 : ILLEGAL FUNCTION : İstenilen fonksiyon kodu desteklenmiyor.

02 : ILLEGAL DATA ADDRESS : Adres slave için işlenilebilir bir adres değil.

03 : ILLEGAL DATA VALUE : Adres miktarı doğru değil. Miktar parametresi çok geniş ya da sıfır.

05 : ACKNOWLEDGE : İsteğin yerine getirilişi esnasında serverda bazı tanımlanamayan hatalar oluştu.

Eğer bir hata durumu oluşmuş ise slave cihazı fonksiyon kodunun en yüksek değerlikli bitini (MSB) lojik 1 yapıp cevap döndürür.

4. PRONEM-midi MODBUS RTU İÇİN REGISTER ADRESLERİ

4.1. HOLDING REGISTERS (4xxxx)

Parametre	Parametre Tanımı	Adres	Çarpan
Slave adresi	Slave Adresi (1 - 247) (Default = 1)	40001	1
Baud rate	Baud Rate 0 - 9600 baud (Default) 1 - 19200 baud 2 - 38400 baud 3 - 57600 baud 4 - 115200 baud	40002	1
Parity	Parity 0 - YOK (Default) 1 - TEK 2 - ÇİFT	40003	1
Stop biti	Stop Bit 0 - 1 Stop Biti (Default) 1 - 2 Stop Biti	40004	1

Örnek-1 : "Baud rate" parametresinin değerini okumak;

Slave ID : 1
Komut : 3 (Read Holding Register)
Register Adresi : 1
Register Adedi : 1

Slave ID	1
Komut	3
Register Adres MSB	0
Register Adres LSB	1
Register Adedi MSB	0
Register Adedi LSB	1
CRC MSB	213 (desimal)
CRC LSB	202 (desimal)

Eğer Baud rate parametresinin değeri 9600kbps ise, cihaz aşağıdaki cevabı döndürecektir:

Slave ID	1
Komut	3
Byte Adedi	2
Data MSB	0
Data LSB	0
CRC MSB	184 (desimal)
CRC LSB	68 (desimal)

Byte Adedi = 2*(Okunmak istenilen register adedi)

Örnek-2 : "Baud rate" parametresini değiştirme;

Slave ID : 1
Komut : 6 (Write Single Register)
Register Adresi : 1
Data : 4

Slave ID	1
Komut	6
Register Adres MSB	0
Register Adres LSB	1
Data MSB	0
Data LSB	4
CRC MSB	217 (desimal)
CRC LSB	201 (desimal)

Registere istenilen değer yazıldıktan sonra slave cihazı sorgu paketini aynen geri döndürür. Cevap aşağıdaki gibi olacaktır:

Slave ID	1
Komut	6
Register Adres MSB	0
Register Adres LSB	1
Data MSB	0
Data LSB	4
CRC MSB	217 (decimal)
CRC LSB	201 (decimal)

Byte Adedi = 2*(Okunmak istenilen register adedi)

Değiştirilen parametrelerin aktif olabilmesi için cihazın enerjisinin kesilip tekrar verilmesi gerekmektedir!

4.2. INPUT REGISTERS (3xxxx)

Açıklama	Adres	Format	Çarpan
Sıcaklık	30001	INT 16	0,01
Bağıl Nem	30002	INT 16	0,1
Cihaz Revizyonu	30003	INT 16	1

INT 16 : signed int, 16 bit data

Ölçüm değerlerini okuma;

Örnek-1 : Sıcaklık değerini okuma

Slave ID : 1
Komut : 4 (Read Input Register)
Register Adresi : 0
Register Adedi : 1

Slave ID	1
Komut	4
Register Adres MSB	0
Register Adres LSB	0
Register Adedi MSB	0
Register Adedi LSB	1
CRC MSB	49 (desimal)
CRC LSB	202 (desimal)

Eğer sıcaklık değeri 2764 ise, cihaz aşağıdaki cevabı döndürecektir:

Slave ID	1
Komut	4
Byte Adedi	2
Data MSB	10 (desimal)
Data LSB	204 (desimal)
CRC MSB	191 (desimal)
CRC LSB	197 (desimal)

Byte Adedi = 2*(Okunmak istenilen register adedi)

Alınan sıcaklık parametesinin değeri 0,01 ile çarpılmalıdır!

Örneğin;

$2764 * 0,01 = 27,64^{\circ}\text{C}$

Örnek-2 : Bağlı Nem değerini okuma

Slave ID : 1
Komut : 4 (Read Input Register)
Register Adresi : 1
Register Adedi : 1

Slave ID	1
Komut	4
Register Adres MSB	0
Register Adres LSB	1
Register Adedi MSB	0
Register Adedi LSB	1
CRC MSB	96 (desimal)
CRC LSB	10 (desimal)

Eğer bağlı nem değeri 417 ise, cihaz aşağıdaki cevabı döndürecektir:

Slave ID	1
Komut	4
Byte Adedi	2
Data MSB	1
Data LSB	161 (desimal)
CRC MSB	121 (desimal)
CRC LSB	24 (desimal)

Byte Adedi = 2*(Okunmak istenilen register adedi)

Alınan bağlı nem parametresinin değeri 0,1 ile çarpılmalıdır!

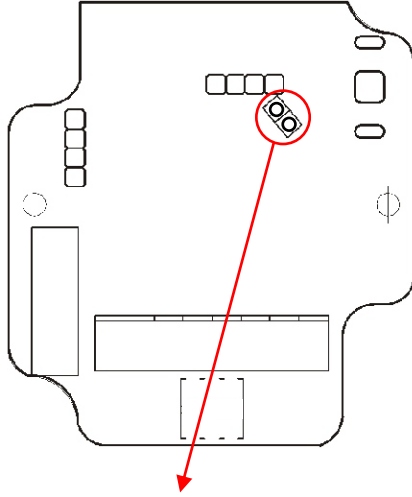
Örneğin;

$417 \times 0,1 = \%41,7$

5. Default Parametreler

Default parametreler;

Slave adresi : 1
Baud rate : 9600 kbps
Parity : Yok
Stop biti : 1 bit



Eğer default parametrelere dönmek isterseniz, yukarıda yeri belirtilen, kartın üzerindeki padleri en az 1 saniye kısa devre etmeniz yeterli olacaktır!