İçindekiler

[Protokolün Yapısı 2](#_Toc162222173)

[EVSE – EV Bağlantısının Kontrolü (PID\_CONTROL\_PILOT) 4](#_Toc162222174)

[Proximity Pilot Sinyalinin Okunması (PID\_PROXIMITY\_PILOT) 6](#_Toc162222175)

[Control Pilot Pwm Kontrolü (PID\_CP\_PWM) 8](#_Toc162222176)

[Rolenin Kontrolü (PID\_RELAY\_CONTROL) 13](#_Toc162222177)

[Panel Üzerindeki Ledin Kontrolü (PID\_LED\_CONTROL) 17](#_Toc162222178)

[Soket Kilidinin Kontrolü (PID\_LOCKER\_CONTROL) 20](#_Toc162222179)

[Anlık Akım Değerlerinin Okunması (PID\_CURRENT) 23](#_Toc162222180)

[Anlık Voltaj Değerlerinin Okunması (PID\_VOLTAGE) 25](#_Toc162222181)

[Anlık Toplam Güç Değerlerinin Okunması (PID\_POWER) 27](#_Toc162222182)

[Enerji Değerlerinin Okunması (PID\_ENERGY) 29](#_Toc162222183)

[RF ID Kart Uniq ID’nin Okunması (PID\_RFID) 31](#_Toc162222184)

[Sıcaklık Değerlerinin Okunması (PID\_EVSE\_TEMP) 37](#_Toc162222185)

[MCU Board’un Resetlenmesi 40](#_Toc162222186)

[Tanımlamalar 41](#_Toc162222187)

# Protokolün Yapısı

Veri iletişiminde Linux kartı Master olarak davranacaktır. Sorgular Linux Bilgisayar modülü tarafından yapılacak, MCU(Mikrodenetleyici Kontrol Kartı) tarafından cevap verilecektir.

Temel olarak 4 paket türü olacaktır. (Command Type)

GET\_COMMAND : Bilgisayar tarafından bir verinin okunması için gönderilecektir.

GET\_RESPONSE : MCU tarafından, ilgili veri cevap olarak gönderilecektir.

SET\_COMMAND : Bilgisayar tarafından bir verinin değiştirilmesi için gönderilecektir.

SET\_RESPONSE : MCU tarafından ilgili veri set edildikten sonra cevap olarak dönecektir.

**Sabitler** :

**GET\_COMMAND** : ‘G’ (decimal 71)

**GET\_RESPONSE** : ‘g’ (decimal 103)

**SET\_COMMAND** : ‘S’ (decimal 83)

**SET\_RESPONSE** : ’s’ (decimal 115)

Veriler gönderilirken paketin başında **STX** karakteri (decimal 2) ve paketin sonunda **LF** (decimal 10) karakteri olmalıdır. Haberleşme 115200bps 8N1 protokolünde Uart üzerinden gerçekleştirilecektir. 1 veri paketi gönderilirken byte’lar arası 86mikrosaniyenin üstünde bekleme olmamalıdır. Böyle bir durumda MCU kartı paketin sonlandığını kabul edecek ve yanlış veri alınmasına neden olacaktır.

Her bir veri paketi uzunluğu 255 byte’ı aşmamalıdır.

**Parameter ID (PID) :**

İletişimde hangi verinin/parametrenin okunacağı veya set edileceği bu değer ile belirlenecektir. 1 byte’tan oluşacaktır.

**Parameter Data Length (PDL):**

Parametre ye ait datanın uzunluğunu belirtir. Veri protokolü string formatında çalışacağından sabit olarak 3 byte’lık bir alandan oluşacaktır.

Örnek PDL 3 : ‘0’, ’0’, ’3’

Örnek PDL 125 : ‘1’, ‘2’, ‘5’

**Checksum :**

Veri paketinin sonundaki LF(Line Feed:10) karakterinden önceki 3 byte’lık alanda olacaktır. 0 – 255 arası(dahil) değerler alabilir. En basit ifadeyle, paketteki LF karakteri ve ondan önceki 3 byte’lık Checksum verisinden önceki tüm byte’ların toplanması ile(mod 256) elde edilecektir…

unsigned char checksum (unsigned char \*ptr, int size) {

unsigned char checksum\_result = 0;

while (size-- != 0)

checksum\_result += \*ptr++;

return checksum\_result;

}

# EVSE – EV Bağlantısının Kontrolü (PID\_CONTROL\_PILOT)

Şarj kablosu ile araç arasında kablo bağlantısı yapıldıktan sonra kablo üzerindeki CP(Control Pilot) iletkeni vasıtasıyla bir haberleşme gerçekleşmektedir. Olası durumlar aşağıdaki şekildedir.

**State A :** Not Connected

**State B :** EV connected, ready to charge

**State C :** EV charging

**State D :** EV charging, ventilation required

**State E :** Error

**State F :** Unknown error

Yukarıdaki bu state’ler öğrenmek için Bilgisayar tarafından MCU kartına aşağıdaki veri gönderilir.

**GET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

Parameter ID : **‘C’**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |  |  |  |  |  |
|  | **G** | Command Type |  |  |  |  |  |
|  | **C** | Parameter ID (PID) |  |  |  |  |  |
|  | **0** | Parameter Data Length(Digit100) | *PDL : 1 (PDL :0 olursa Parameter Data kısmı atlanacaktır)* | | | | |
|  | **0** | Parameter Data Length(Digit10) |
|  | **1** | Parameter Data Length(Digit1) |
|  | **1** | Parameter Data | *Connector No (İlerisi için aynı cihazda birden fazla konektör kullanma ihtimali için ayrılmıştır. Bu modelde default olarak 1 olacaktır.)* | | | | |
|  |  | **0** | Checksum |  |  |  |  |  |
|  |  | **7** |  |  |  |  |  |
|  |  | **8** |  |  |  |  |  |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |  |  |  |  |

Yukarıdaki veri MCU kartına gönderildikten sonra Parameter Data alanında State leri belirten 1 byte lık bir veri olacak şekilde cevap beklenecektir.

**GET\_RESPONSE:**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Checksum için Kullanılan Veri Alanı |  | **STX(decimal 2)** | Start of Text (STX) |  |
|  | **g** | Command Type | Get Command |
|  | **C** | Parameter ID (PID) | State of Control Pilot Signal |
|  | **0** | Parameter Data Length(Digit100) | *PDL : 2* |
|  | **0** | Parameter Data Length(Digit10) |
|  | **2** | Parameter Data Length(Digit1) |
|  | **1** | Connector No |  |
|  | **B** | Parameter Data | *A,B,C,D,E veya F değerini almaktadır* |
|  |  | **1** | Checksum |  |
|  |  | **7** |  |
|  |  | **7** |  |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |

Yukarıdaki pakette cevap olarak State durumu ‘B’ gelmiştir. Yani cihaz ile araç arasında bağlantı tespit edilmiştir. (Ev connected)

# Proximity Pilot Sinyalinin Okunması (PID\_PROXIMITY\_PILOT)

Kendi üzerinde şarj kablosu olmayan Soketli Tip (Type 2) AC Şarj cihazlarında kullanıcı, şarj işlemi için kendi kablosunu kullanmaktadır. Kullanılan kablolardaki iletkenin kalınlığına göre bu kablonun akım taşıma kapasitesi 13A, 20A, 32A ve 63A olarak değişiklik göstermektedir.

Soketli Tip AC Şarj Cihazları, soket üzerinde bulunan Proximity Pilot(PP) pini üzerinden, kablonun takılı olup olmadığını ve takılı ise kablonun maximum akım taşıma kapasitesini algılayabilmektedir. Kablolu tip (Tethered Type) şarj cihazlarında Proximity Pilot ucu aktif değildir ve kullanılmamaktadır. Çünkü üretici, kabloyu bütünleşik olarak sunmakta ve cihaz içerisinde maximum şarj akımı zaten bilinmektedir.

Şarj işlemi başlatılırken, tüm hata durumları kontrol edilir ve Control Pilot sinyalinden “State C” alındıktan sonra kablonun takılı olup olmadığı Proximity Pilot sinyali okunarak kontrol edilir ve takılı olan kablonun maximum akım taşıma kapasitesi algılandıktan sonra şarj işlemine devam edilir.

Örneğin, AC Şarj cihazı 32 Amper akım verme kapasitesine sahip olduğu halde, takılan kablo 13 Amperlik bir kablo ise bu durumda araçtan, kablonun maximum kapasitesi kadar(13A) akım çekilmesi talep edilir. (Bu işlem Control Pilot ucundaki PWM duty genişliği ile ayarlanır. (Bknz:PID\_CP\_PWM)

Proximity Pilot Sinyali sadece okunabilir bir değer olduğundan GET\_COMMAND kullanılır.

**GET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

Parameter ID : **‘X’**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX (decimal 2)** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **X** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **1** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **?** | Checksum (Digit 100) |
| 8 |  | **?** | Checksum (Digit 10) |
| 9 |  | **?** | Checksum (Digit 1) |
| 10 |  | **LF (decimal:10)** | Line Feed (Paket Sonu) |

Yukarıdaki veri MCU kartına gönderildikten sonra Parameter Data alanında 1 byte lık bir veri olacak şekilde cevap beklenecektir.

**GET\_RESPONSE:**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **X** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **0** | Parameter Data Length(Digit10) |  |
| 5 |  | **2** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **N** | Parameter Data | Cable Not Plugged |
| 28 |  | **?** | Checksum (Digit 100) |  |
| 29 |  | **?** | Checksum (Digit 10) |  |
| 30 |  | **?** | Checksum (Digit 1) |  |
| 31 |  | **LF** | Line Feed (Paket Sonu) |  |

Yukarıdaki paketteki cevapta Parameter Data olarak ‘N’ gelmiştir. Yani kablo takılı değildir. Fakat burada dikkat edilmesi gereken bir durum vardır; Proximity Pilot sinyalinden okunan veri, kablonun araca bağlanıp bağlanmadığı bilgisini vermez, sadece Şarj Cihazına bağlı olup olmadığı bilgisini verir.

Parameter Data’nın Alabileceği değerler aşağıdaki gibidir.

**‘N’ :** Cable Not Plugged

**‘E’ :** Error

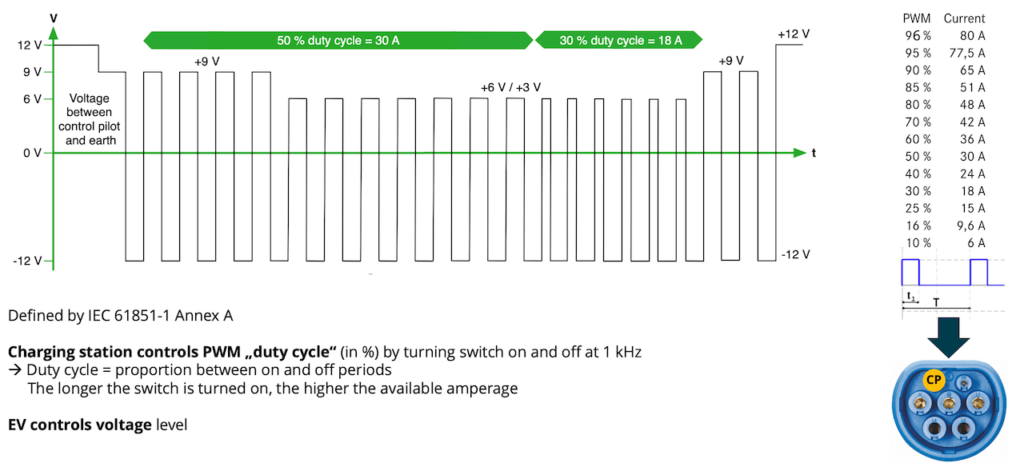
**‘1’ :** Cable Plugged into Charger, Cable Capacity is 13 Amper

**‘2’ :** Cable Plugged into Charger, Cable Capacity is 20 Amper

**‘3’ :** Cable Plugged into Charger, Cable Capacity is 32 Amper

**‘6’ :** Cable Plugged into Charger, Cable Capacity is 63 Amper

# Control Pilot Pwm Kontrolü (PID\_CP\_PWM)



AC Şarj Sisteminde, şarj istasyonu ile EV arasından bağlantı sağlandıktan sonra (B sinyali alındıktan sonra) Control Pilot iletkeni üzerinden bir PWM sinyali uygulanması gerekir. PWM sinyali başlatıldıktan sonra Araç, şarja hazır ise “C” sinyalini gönderir ve role kontakları da kapatılarak şarj işlemi başlatılmış olur. PWM sinyalinin darbe genişliği ise, araca en fazla ne kadar akım çekmesi gerektiği konusunda bilgi verir. AC şarj sistemlerinde şarj akımı minimum 6A ile sınırlıdır. PWM sinyali üretilirken 6Amper ile 32A arası bir değer gönderilecektir.

Control Pilot PWM sinyalini set edebilmek(kontrol etmek) için aşağıdaki paket gönderilir.

Parameter ID : **‘G’**

**SET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **S** | Command Type |  |
| 2 |  | **G** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **0** | Parameter Data Length(Digit10) |  |
| 5 |  | **5** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **0** | Current Digit 100 | **Akım Değeri: 032.0 Amper** |
| 8 |  | **3** | Current Digit 10 |
| 9 |  | **2** | Current Digit 1 |
| 10 |  | **0** | Current Digit 0.1 |
| 11 |  | **?** | Checksum (Digit 100) |  |
| 12 |  | **?** | Checksum (Digit 10) |  |
| 13 |  | **?** | Checksum (Digit 1) |  |
| 14 |  | **LF** | Line Feed (Paket Sonu) |  |

PWM sinyalini tamamen durdurmak için, akım değeri “0000” olarak gönderilmesi gerekmektedir.

**SET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

Yukarıdaki paket MCU kartına gönderildikten sonra, işlemin onaylandığına dair aşağıdaki şekilde bir cevap dönecektir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **s** | Command Type |  |
| 2 |  | **G** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **0** | Parameter Data Length(Digit10) |  |
| 5 |  | **6** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **0** | Current Digit 100 | **Akım Değeri: 032.0 Amper** |
| 8 |  | **3** | Current Digit 10 |
| 9 |  | **2** | Current Digit 1 |
| 10 |  | **0** | Current Digit 0.1 |
| 11 |  | **?** | Checksum (Digit 100) |  |
| 12 |  | **?** | Checksum (Digit 10) |  |
| 13 |  | **?** | Checksum (Digit 1) |  |
| 14 |  | **LF** | Line Feed (Paket Sonu) |  |

**GET\_COMMAND :**

Herhangi bir anda, Control Pilot iletkeni üzerindeki PWM sinyalinin değerini öğrenmek için aşağıdaki GET\_COMMAND komutu gönderilir.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **G** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **1** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **?** | Checksum (Digit 100) |
| 8 |  | **?** | Checksum (Digit 10) |
| 9 |  | **?** | Checksum (Digit 1) |
| 10 |  | **LF** | Line Feed (Paket Sonu) |

Yukarıdaki GET komutu gönderildikten sonra PWM sinyalinin hangi değerde çalıştığını gösteren aşağıdaki veri paketi döner. Paket içerisindeki akım değeri “0000” ise, PWM sinyali Stop konumundadır, PWM sinyali üretilmemektedir.

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |
| 1 |  | **g** | Command Type |
| 2 |  | **G** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **5** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **0** | Current Digit 100 |
| 8 |  | **3** | Current Digit 10 |
| 9 |  | **2** | Current Digit 1 |
| 10 |  | **0** | Current Digit 0.1 |
| 11 |  | **?** | Checksum (Digit 100) |
| 12 |  | **?** | Checksum (Digit 10) |
| 13 |  | **?** | Checksum (Digit 1) |
| 14 |  | **LF** | Line Feed (Paket Sonu) |

Örnek cevap paketinde, PWM in aktif olduğu ve 32 Ampere karşılık gelen darbe genişliğinde PWM üretildiğini göstermektedir.

# Rolenin Kontrolü (PID\_RELAY\_CONTROL)

Röleyi kontrol etmek için (‘1’ veya ‘0’) aşağıdaki paket gönderilir.

Parameter ID : **‘R’**

**SET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |  |  |  |
|  |  |  |  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |  |  |  |
|  | **S** | Command Type |  |  |  |
|  | **R** | Parameter ID (PID) |  |  |  |
|  | **0** | Parameter Data Length(Digit100) |  |  |  |
|  | **0** | Parameter Data Length(Digit10) |  |  |  |
|  | **2** | Parameter Data Length(Digit1) |  |  |  |
|  | **1** | Connector No |  |  |  |
|  | **1** | On-Off Control | *Başlatmak için 1, durdurmak için 0* | | |
|  |  | **?** | Checksum (Digit 100) |  |  |  |
|  |  | **?** | Checksum (Digit 10) |  |  |  |
|  |  | **?** | Checksum (Digit 1) |  |  |  |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |  |  |

**SET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

Mcu Kartı, komutu aldıktan sonra aynı komutu Set Response ile geri gönderir. Geri dönen bu cevap rölenin On / Off olduğunu kontrol etmek için değildir. Gelen cevap sadece komutun alındığını gösterir. Rolenin kontaklarının gerçekten kapandığını veya açıldığını öğrenmek için “GET\_COMMAND” gönderilmelidir.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |  |  |  |
|  |  |  |  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |  |  |  |
|  | **s** | Command Type |  |  |  |
|  | **R** | Parameter ID (PID) |  |  |  |
|  | **0** | Parameter Data Length(Digit100) |  |  |  |
|  | **0** | Parameter Data Length(Digit10) |  |  |  |
|  | **2** | Parameter Data Length(Digit1) |  |  |  |
|  | **1** | Connector No |  |  |  |
|  | **1** | On-Off Control | *Gönderilen komutun aynısını geri gönderir* | | |
|  |  | **?** | Checksum (Digit 100) |  |  |  |
|  |  | **?** | Checksum (Digit 10) |  |  |  |
|  |  | **?** | Checksum (Digit 1) |  |  |  |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |  |  |

**GET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

|  |  |
| --- | --- |
| **Örnek Veri** |  |
|  |  |
| **STX(decimal 2)** | Start of Text (STX) |
| **G** | Command Type |
| **R** | Parameter ID (PID) |
| **0** | Parameter Data Length(Digit100) |
| **0** | Parameter Data Length(Digit10) |
| **1** | Parameter Data Length(Digit1) |
| **1** | Connector No |
| **?** | Checksum (Digit 100) |
| **?** | Checksum (Digit 10) |
| **?** | Checksum (Digit 1) |
| **LF (decimal 10)** | Line Feed (Paket Sonu) |

Yukarıdaki “GET” komutu gönderilerek, MCU Boarddan cevap dönmesi beklenir. Rolenin kontakları fiziksel olarak kontrol edilir ve kontakların durumu cevap olarak geri döner. Burada dikkat edilmesi gereken bir konu vardır ki;

Eğer SET\_COMMAND ile röleye bir komut verilirse, rölenin kontaklarının açılması veya kapanması bir miktar süre aldığından(~200 milisaniye) GET\_COMMAND ile rölenin kontakları kontrol edilirken en az 200ms kadar beklenmelidir. Rolenin GET\_COMMAND ile kontaklarının kontrol edilmesi büyük önem arz etmektedir. Eğer Roleye ‘1’ komutu verildiyse ve rölenin durumu okunurken ‘0’ geldiyse(200ms sonra) veya tam tersi durum oluştuysa bu durum rölenin kontaklarının yapıştığına işaret eder. Böyle bir durumda belirli aralıklarla birkaç kez daha durum kontrol edilmeli ve sorun halen devem ediyorsa cihaz HATA DURUMU’na geçirilmelidir.

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Örnek Veri** |  |  |  |  |
|  |  |  |  |  |
| **STX(decimal 2)** | Start of Text (STX) |  |  |  |
| **g** | Command Type |  |  |  |
| **R** | Parameter ID (PID) |  |  |  |
| **0** | Parameter Data Length(Digit100) |  |  |  |
| **0** | Parameter Data Length(Digit10) |  |  |  |
| **2** | Parameter Data Length(Digit1) |  |  |  |
| **1** | Connector No |  |  |  |
| **0** | State of Relay (1 or 0) | *Rolenin devrede olup olmadığını gösterir* | | |
| **?** | Checksum (Digit 100) |  |  |  |
| **?** | Checksum (Digit 10) |  |  |  |
| **?** | Checksum (Digit 1) |  |  |  |
| **LF (decimal 10)** | Line Feed (Paket Sonu) |  |  |  |

# Panel Üzerindeki Ledin Kontrolü (PID\_LED\_CONTROL)

Linux kartı, yapılan işlemlere göre led panel üzerinde çeşitli efektleri kontrol etmesi gerekmektedir. Linux kartı tarafından sadece gerekli State parametresi gönderilmesi yeterlidir. Renklere ve renklerin animasyonunun kontrolü MCU Board tarafından yapılacaktır.

Parameter ID : **‘L’**

**SET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |  |  |  |
|  |  |  |  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |  |  |  |
|  | **S** | Command Type |  |  |  |
|  | **L** | Parameter ID (PID) |  |  |  |
|  | **0** | Parameter Data Length(Digit100) |  |  |  |
|  | **0** | Parameter Data Length(Digit10) |  |  |  |
|  | **2** | Parameter Data Length(Digit1) |  |  |  |
|  | **1** | Connector No |  |  |  |
|  | **1** | **Led State** | *tabloya göre buraya deger verilecektir.* | | |
|  |  | **?** | Checksum (Digit 100) |  |  |  |
|  |  | **?** | Checksum (Digit 10) |  |  |  |
|  |  | **?** | Checksum (Digit 1) |  |  |  |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |  |  |

Led State Tablosu:

|  |  |  |  |
| --- | --- | --- | --- |
| Efekt | Led State Tablosu | Değer  ASCII | Değer  Decimal |
|  |  |  |  |
| Constant Green | Standby | **1** | *49* |
| Pulsate Blue (Slow Flash) | Connecting | **2** | *50* |
| Pulsate Blue (Fast Flash) | RfId Verified | **3** | *51* |
| Breathing Blue | Charging | **4** | *52* |
| Pulsate Red (Fast Flash) | RfId Failed | **5** | *53* |
| Pulsate Red (Slow Flash) | Need Replugging | **6** | *54* |  |
| Constant Red | Fault | **7** | *55* |
| Constant Turquaze | ChargingStopped | **8** | *56* |
| Pulsate Green(Slow Flash) | Waiting Plugging | **;** | *59* |
| Pulsate Turquaze | Device Offline | **=** | *61* |

**SET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

Linux Board tarafından ilgili led state komutu gönderildikten sonra, MCU board set edilen değeri geri gönderir.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |
|  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |
|  | **s** | Command Type |
|  | **L** | Parameter ID (PID) |
|  | **0** | Parameter Data Length(Digit100) |
|  | **0** | Parameter Data Length(Digit10) |
|  | **2** | Parameter Data Length(Digit1) |
|  | **1** | Connector No |
|  | **4** | Led State (Charging) |
|  |  | **?** | Checksum (Digit 100) |
|  |  | **?** | Checksum (Digit 10) |
|  |  | **?** | Checksum (Digit 1) |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |

**GET\_COMMAND :**

Linux Kartı led panelinde halihazırda hangi efektin çalıştığını öğrenmek için aşağıdaki GET\_COMMAND komutu gönderir ve Led State hakkında bilgi alır.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |
|  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |
|  | **G** | Command Type |
|  | **L** | Parameter ID (PID) |
|  | **0** | Parameter Data Length(Digit100) |
|  | **0** | Parameter Data Length(Digit10) |
|  | **1** | Parameter Data Length(Digit1) |
|  | **1** | Connector No |
|  |  | **?** | Checksum (Digit 100) |
|  |  | **?** | Checksum (Digit 10) |
|  |  | **?** | Checksum (Digit 1) |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |

Yukarıdaki “GET” komutu gönderilerek, MCU Boarddan aşağıdaki şekilde cevap dönmesi beklenir.

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |
|  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |
|  | **g** | Command Type |
|  | **L** | Parameter ID (PID) |
|  | **0** | Parameter Data Length(Digit100) |
|  | **0** | Parameter Data Length(Digit10) |
|  | **2** | Parameter Data Length(Digit1) |
|  | **1** | Connector No |
|  | **1** | Led State (Standby) |
|  |  | **?** | Checksum (Digit 100) |
|  |  | **?** | Checksum (Digit 10) |
|  |  | **?** | Checksum (Digit 1) |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |

# Soket Kilidinin Kontrolü (PID\_LOCKER\_CONTROL)

Soketli tip Şarj Cihazlarında soket içerisindeki kilit mekanizmasının kontrolü aşağıdaki şekilde SET\_SOMMAND ile kontrol edilir. Kilidin durumunu öğrenmek için ise GET\_COMMAND komutu kullanılır.

Parameter ID : **‘K’**

**SET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |  |  |  |
|  |  |  |  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |  |  |  |
|  | **S** | Command Type |  |  |  |
|  | **K** | Parameter ID (PID) |  |  |  |
|  | **0** | Parameter Data Length(Digit100) |  |  |  |
|  | **0** | Parameter Data Length(Digit10) |  |  |  |
|  | **2** | Parameter Data Length(Digit1) |  |  |  |
|  | **1** | Connector No |  |  |  |
|  | **1** | **Locker State (1:Lock, 0:Unlock)** | *Kilitlemek için 1, kilidi açmak için 0 değeri gönderilir.* | | |
|  |  | **?** | Checksum (Digit 100) |  |  |  |
|  |  | **?** | Checksum (Digit 10) |  |  |  |
|  |  | **?** | Checksum (Digit 1) |  |  |  |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |  |  |

**SET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

Linux Board tarafından ilgili komutu gönderildikten sonra, MCU board set edilen değeri geri gönderir. Kilitleme işlemi bir miktar süre alacağından 300mS ~ 1500ms sonra tekrar GET\_COMMAND ile kilidin gerçekten kilitlendi mi veya açıldı mı durumu kontrol edilmelidir.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |
|  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |
|  | **s** | Command Type |
|  | **K** | Parameter ID (PID) |
|  | **0** | Parameter Data Length(Digit100) |
|  | **0** | Parameter Data Length(Digit10) |
|  | **2** | Parameter Data Length(Digit1) |
|  | **1** | Connector No |
|  | **1** | Locker State (Locked) |
|  |  | **?** | Checksum (Digit 100) |
|  |  | **?** | Checksum (Digit 10) |
|  |  | **?** | Checksum (Digit 1) |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |

**GET\_COMMAND :**

Kilit mekanizmasının kilitli veya açık olup olmadığı bilgisi aşağıdaki şekilde komut gönderilerek öğrenilir.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |
|  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |
|  | **G** | Command Type |
|  | **K** | Parameter ID (PID) |
|  | **0** | Parameter Data Length(Digit100) |
|  | **0** | Parameter Data Length(Digit10) |
|  | **1** | Parameter Data Length(Digit1) |
|  | **1** | Connector No |
|  |  | **?** | Checksum (Digit 100) |
|  |  | **?** | Checksum (Digit 10) |
|  |  | **?** | Checksum (Digit 1) |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |

Yukarıdaki “GET” komutu gönderilerek, MCU Boarddan aşağıdaki şekilde cevap dönmesi beklenir.

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Örnek Veri** |  |
|  |  |  |  |
| **Checksum için Kullanılan Veri Alanı** |  | **STX(decimal 2)** | Start of Text (STX) |
|  | **g** | Command Type |
|  | **K** | Parameter ID (PID) |
|  | **0** | Parameter Data Length(Digit100) |
|  | **0** | Parameter Data Length(Digit10) |
|  | **2** | Parameter Data Length(Digit1) |
|  | **1** | Connector No |
|  | **0** | Locker State (Unlocked) |
|  |  | **?** | Checksum (Digit 100) |
|  |  | **?** | Checksum (Digit 10) |
|  |  | **?** | Checksum (Digit 1) |
|  |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |

# Anlık Akım Değerlerinin Okunması (PID\_CURRENT)

Araç tarafından çeklilen akım değerlerinin okunması için GET\_COMMAND sorgusu yapılır. Bu değerler sadece okunabilir olduğundan SET\_COMMAND kullanılamaz.

Veri paketinde standart olarak 3 faz için de ayrı ayrı alan ayrılmıştır. 1 fazlı modeller için de diğer 2 fazın akım bilgisi de gelecektir, fakat bu alan 1 fazlı modelde dikkate alınmamalıdır.

Parameter ID : **‘I’**

**GET\_COMMAND :**

Cihazdan anlık akım değerlerini öğrenmek için aşağıdaki paket gönderilir.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX (decimal 2)** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **I** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **1** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **?** | Checksum (Digit 100) |
| 8 |  | **?** | Checksum (Digit 10) |
| 9 |  | **?** | Checksum (Digit 1) |
| 10 |  | **LF (decimal:10)** | Line Feed (Paket Sonu) |

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX(decimal 2)** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **I** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **2** | Parameter Data Length(Digit10) |  |
| 5 |  | **2** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **R** | Phase Name (L1) | 1. Faz seperator |
| 8 |  | **0** | Current Digit 100 | Faz 1 den çekilen akım : 16,500 Amper |
| 9 |  | **1** | Current Digit 10 |
| 10 |  | **6** | Current Digit 1 |
| 11 |  | **5** | Current Digit 0.1 |
| 12 |  | **0** | Current Digit 0.01 |
| 13 |  | **0** | Current Digit 0.001 |
| 14 |  | **S** | Phase Name (L2) | 2. Faz seperator |
| 15 |  | **0** | Current Digit 100 | Faz 2 den çekilen akım : 16,485 Amper |
| 16 |  | **1** | Current Digit 10 |
| 17 |  | **6** | Current Digit 1 |
| 18 |  | **4** | Current Digit 0.1 |
| 19 |  | **8** | Current Digit 0.01 |
| 20 |  | **5** | Current Digit 0.001 |
| 21 |  | **T** | Phase Name (L3) | 3. Faz seperator |
| 22 |  | **0** | Current Digit 100 | Faz 3 ten çekilen akım : 16,532 Amper |
| 23 |  | **1** | Current Digit 10 |
| 24 |  | **6** | Current Digit 1 |
| 25 |  | **5** | Current Digit 0.1 |
| 26 |  | **3** | Current Digit 0.01 |
| 27 |  | **2** | Current Digit 0.001 |
| 28 |  | **?** | Checksum (Digit 100) |  |
| 29 |  | **?** | Checksum (Digit 10) |  |
| 30 |  | **?** | Checksum (Digit 1) |  |
| 31 |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |

# Anlık Voltaj Değerlerinin Okunması (PID\_VOLTAGE)

Fazlardaki gerilim değerlerinin okunması için GET\_COMMAND sorgusu yapılır. Bu değerler sadece okunabilir olduğundan SET\_COMMAND kullanılamaz.

Veri paketinde standart olarak 3 faz için de ayrı ayrı alan ayrılmıştır. 1 fazlı modeller için de diğer 2 fazın gerilim bilgisi de gelecektir, fakat bu alan 1 fazlı modelde dikkate alınmamalıdır.

Parameter ID : **‘V’**

**GET\_COMMAND :**

Cihazdan anlık akım değerlerini öğrenmek için aşağıdaki paket gönderilir.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX (decimal 2)** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **V** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **1** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **?** | Checksum (Digit 100) |
| 8 |  | **?** | Checksum (Digit 10) |
| 9 |  | **?** | Checksum (Digit 1) |
| 10 |  | **LF (decimal:10)** | Line Feed (Paket Sonu) |

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX(decimal 2)** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **V** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **2** | Parameter Data Length(Digit10) |  |
| 5 |  | **2** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **R** | Phase Name (L1) | 1. Faz seperator |
| 8 |  | **0** | Voltage Digit 1000 | Faz 1 deki gerilim: 230,15 Volt |
| 9 |  | **2** | Voltage Digit 100 |
| 10 |  | **3** | Voltage Digit 10 |
| 11 |  | **0** | Voltage Digit 1 |
| 12 |  | **1** | Voltage Digit 0.1 |
| 13 |  | **5** | Voltage Digit 0.01 |
| 14 |  | **S** | Phase Name (L2) | 2. Faz seperator |
| 15 |  | **0** | Voltage Digit 1000 | Faz 2 deki gerilim: 228,10 Volt |
| 16 |  | **2** | Voltage Digit 100 |
| 17 |  | **2** | Voltage Digit 10 |
| 18 |  | **8** | Voltage Digit 1 |
| 19 |  | **1** | Voltage Digit 0.1 |
| 20 |  | **0** | Voltage Digit 0.01 |
| 21 |  | **T** | Phase Name (L3) | 3. Faz seperator |
| 22 |  | **0** | Voltage Digit 1000 | Faz 2 deki gerilim: 220,00 Volt |
| 23 |  | **2** | Voltage Digit 100 |
| 24 |  | **2** | Voltage Digit 10 |
| 25 |  | **0** | Voltage Digit 1 |
| 26 |  | **0** | Voltage Digit 0.1 |
| 27 |  | **0** | Voltage Digit 0.01 |
| 28 |  | **?** | Checksum (Digit 100) |  |
| 29 |  | **?** | Checksum (Digit 10) |  |
| 30 |  | **?** | Checksum (Digit 1) |  |
| 31 |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |

# Anlık Toplam Güç Değerlerinin Okunması (PID\_POWER)

Araç tarafından çeklilen anlık güç değerlerinin okunması için GET\_COMMAND sorgusu yapılır. Bu değerler sadece okunabilir olduğundan SET\_COMMAND kullanılamaz.

Veri paketinde 7 nolu slottaki Power Type verisi, hangi gücün okunacağını belirtir.

Aktif Güç için(kw) : ‘P’

Reaktif Güç için(kVAR) : ‘R’

Görünür Güç için(kVA) : ‘S’ değeri girilmelidir.

Şuan için sadece aktif güç gerekli olduğundan, diğer paketler daha sonra eklenecektir.

Parameter ID : **‘P’**

**GET\_COMMAND :**

Cihazdan anlık toplam aktif güç değerini öğrenmek için aşağıdaki paket gönderilir.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX (decimal:3)** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **P** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **2** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **P** | Power Type |
| 8 |  | **?** | Checksum (Digit 100) |
| 9 |  | **?** | Checksum (Digit 10) |
| 10 |  | **?** | Checksum (Digit 1) |
| 11 |  | **LF (decimal:10)** | Line Feed (Paket Sonu) |

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX(decimal 2)** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **P** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **0** | Parameter Data Length(Digit10) |  |
| 5 |  | **8** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **P** | Power Type | P: Aktif Güç |
| 8 |  | **0** | Power Digit 100 | Anlık Toplam Güç: 17.450 kW |
| 9 |  | **1** | Power Digit 10 |
| 10 |  | **7** | Power Digit 1 |
| 11 |  | **4** | Power Digit 0.1 |
| 12 |  | **5** | Power Digit 0.01 |
| 13 |  | **0** | Power Digit 0.001 |
| 14 |  | **?** | Checksum (Digit 100) |  |
| 15 |  | **?** | Checksum (Digit 10) |  |
| 16 |  | **?** | Checksum (Digit 1) |  |
| 17 |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |

# Enerji Değerlerinin Okunması (PID\_ENERGY)

Tüketilen enerji değerlerinin okunması için GET\_COMMAND sorgusu yapılır. Bu değerler sadece okunabilir olduğundan SET\_COMMAND kullanılamaz.

Veri paketinde 7 nolu slottaki Energy Type verisi, hangi türünün okunacağını belirtir.

Aktif Enerji için(kWh) : ‘P’

Reaktif Enerji için(kVARh) : ‘R’

Görünür Enerji için(kVAh) : ‘S’ değeri girilmelidir.

Şuan için sadece aktif enerji gerekli olduğundan, diğer paketler daha sonra eklenecektir.

Yalnız enerji değerleri okunurken şuna dikkat edilmelidir. Enerji değeri MCU Boardda sürekli artan şekilde kalıcı hafızada tutulmaktadır. Yani okuma yapıldığında bir sayaç gibi sayacın değeri okunacaktır.

Şarj işleminin ne kadar enerji tükettiğini hesaplamak için, şarj başlangıcındaki okuma değeri ile şarjın sonlandırılması esnasında okunan iki değer arasındaki fark dikkate alınacaktır.

Parameter ID : **‘E’**

**GET\_COMMAND :**

Cihazdan anlık toplam aktif güç değerini öğrenmek için aşağıdaki paket gönderilir.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX (decimal:3)** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **E** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **2** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **P** | Energy Type (Aktif Energy:kWh) |
| 8 |  | **?** | Checksum (Digit 100) |
| 9 |  | **?** | Checksum (Digit 10) |
| 10 |  | **?** | Checksum (Digit 1) |
| 11 |  | **LF (decimal:10)** | Line Feed (Paket Sonu) |

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX(decimal 2)** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **E** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **1** | Parameter Data Length(Digit10) |  |
| 5 |  | **2** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **P** | Energy Type | P: Aktif Energy (kWh) |
| 8 |  | **0** | Energy Digit 1000000 | Sayaçtan Gelen Değer: 12345.678 kWh |
| 9 |  | **0** | Energy Digit 100000 |
| 10 |  | **1** | Energy Digit 10000 |
| 11 |  | **2** | Energy Digit 1000 |
| 12 |  | **3** | Energy Digit 100 |
| 13 |  | **4** | Energy Digit 10 |
| 14 |  | **5** | Energy Digit 1 |
| 15 |  | **6** | Energy Digit 0.1 |
| 16 |  | **7** | Energy Digit 0.01 |
| 17 |  | **8** | Energy Digit 0.001 |
| 18 |  | **?** | Checksum (Digit 100) |  |
| 19 |  | **?** | Checksum (Digit 10) |  |
| 20 |  | **?** | Checksum (Digit 1) |  |
| 21 |  | **LF (decimal 10)** | Line Feed (Paket Sonu) |  |

# RF ID Kart Uniq ID’nin Okunması (PID\_RFID)

RFID kartların türlerine göre 4 byte lık ve 7 byte lık Uniq ID leri bulunmaktadır. MCU kartı, en son okutulan RFID kartı hafızasında tutar ve Linux kartı tarafından talep edildiğinde bu bilgiyi Linux kartına aktarır.

Fakat yeni bir okuma işleminden önce Linux tarafından MCU Board’a bir kez SET komutu gönderilerek, hafızasındaki UniqID ‘nin silinmesi talep edilir. Daha sonra bir kart tespit edildiyse ve okuma işlemi tamamlandıysa yine MCU board’a SET komutu ile Silme komutu gönderilir.

Parameter ID : **‘N’**

**SET\_COMMAND :**

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |
| 1 |  | **S** | Command Type |
| 2 |  | **N** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **2** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **R** | Reset Komutu(Sil) |
| 8 |  | **?** | Checksum (Digit 100) |
| 9 |  | **?** | Checksum (Digit 10) |
| 10 |  | **?** | Checksum (Digit 1) |
| 11 |  | **LF** | Line Feed (Paket Sonu) |

7 nolu slota ‘R’ (Sil))verisi eklenerek gönderilen paketten sonra MCU Board hafızasında en son okunan kart id sini siler ve aşağıdaki şekilde cevap döner.

**SET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

Yukarıdaki paket MCU kartına gönderildikten sonra, işlemin onaylandığına dair aşağıdaki şekilde bir cevap dönecektir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |
| 1 |  | **s** | Command Type |
| 2 |  | **N** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **2** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **R** | Reset Komutu |
| 8 |  | **?** | Checksum (Digit 100) |
| 9 |  | **?** | Checksum (Digit 10) |
| 10 |  | **?** | Checksum (Digit 1) |
| 11 |  | **LF** | Line Feed (Paket Sonu) |

**GET\_COMMAND :**

Hafızadai en son okunan kartın Uniq ID sini okumak için ise aşağıdaki GET komutu kullanılır.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **N** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **1** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **?** | Checksum (Digit 100) |
| 8 |  | **?** | Checksum (Digit 10) |
| 9 |  | **?** | Checksum (Digit 1) |
| 10 |  | **LF** | Line Feed (Paket Sonu) |

Yukarıdaki GET komutu gönderildikten sonra en son okunan kartın “Uniq ID” uzunluğuna göre farklı uzunluklarda cevap döner.

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

4 byte lık bir Uniq Id ‘ye sahip kart okutulduğunda dönen cevap aşağıdaki şekildedir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **N** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **1** | Parameter Data Length(Digit10) |  |
| 5 |  | **1** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **0** | Nof Bytes to Read (Uid Length\*2) | Digit 10 |
| 8 |  | **8** | Digit 1 |
| 9 |  | **3** | 34 (Card Id 1. Byte Hexadecimal) | **CARD ID : 0x34F5A07E** |
| 10 |  | **4** |
| 11 |  | **F** | F5 (Card Id 2. Byte Hexadecimal) |
| 12 |  | **5** |
| 13 |  | **A** | A0 (Card Id 3. Byte Hexadecimal) |
| 14 |  | **0** |
| 15 |  | **7** | 7E (Card Id 4. Byte Hexadecimal) |
| 16 |  | **E** |
| 17 |  | **?** | Checksum (Digit 100) |  |
| 18 |  | **?** | Checksum (Digit 10) |  |
| 19 |  | **?** | Checksum (Digit 1) |  |
| 20 |  | **LF** | Line Feed (Paket Sonu) |  |

Yukarıdaki paket incelendiğinde PDL: 11 byte olarak gözükmekte ve kartın Uniq Id sini elde etmek için okunması gereken byte sayısı 08 olarak gözükmektedir(7 ve 8 nolu slot ). Kartın hexadecimal Uniq ID ‘si ASCII değerlere çevrildiğinden, okunacak byte sayısı kartın id’sinin 2 katı olarak gözükmektedir.

Ardından 9 nolu slottan başlayarak karakterler okunmalı ve kartın Uniq ID’si hexadecimal olarak **34F5A07E** şekinde elde edilmelidir.

Eğer kartın Uniq ID’si 7 byte’tan oluşuyorsa aşağıdaki şekilde bir cevap gelecektir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **N** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **1** | Parameter Data Length(Digit10) |  |
| 5 |  | **7** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **1** | Nof Bytes to Read (Uid Length\*2) | Digit 10 |
| 8 |  | **4** | Digit 1 |
| 9 |  | **2** | 23 (Card Id 1. Byte Hexadecimal) | **CARD ID : 0x23456789ABCDEF** |
| 10 |  | **3** |
| 11 |  | **4** | 45 (Card Id 2. Byte Hexadecimal) |
| 12 |  | **5** |
| 13 |  | **6** | 67 (Card Id 3. Byte Hexadecimal) |
| 14 |  | **7** |
| 15 |  | **8** | 89 (Card Id 4. Byte Hexadecimal) |
| 16 |  | **9** |
| 17 |  | **A** | AB (Card Id 5. Byte Hexadecimal) |
| 18 |  | **B** |
| 19 |  | **C** | CD (Card Id 6. Byte Hexadecimal) |
| 20 |  | **D** |
| 21 |  | **E** | EF (Card Id 7. Byte Hexadecimal) |
| 22 |  | **F** |
| 23 |  | **?** | Checksum (Digit 100) |  |
| 24 |  | **?** | Checksum (Digit 10) |  |
| 25 |  | **?** | Checksum (Digit 1) |  |
| 26 |  | **LF** | Line Feed (Paket Sonu) |  |

Bunun dışında, SET komutu ile MCU boardda en son okunan kartın id’si silindikten sonra hiçbir kart okutulmadıysa, PDL:3 gelecek ve 7,8 nolu slotlar” (Nof Bytes to Read) : “00” olarak gelecektir. Bu durumda henüz kart okutulmadığı anlaşılacaktır.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **N** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **0** | Parameter Data Length(Digit10) |  |
| 5 |  | **3** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **0** | Nof Bytes to Read (Uid Length\*2) | Digit 10 |
| 8 |  | **0** | Digit 1 |
| 9 |  | **?** | Checksum (Digit 100) |  |
| 10 |  | **?** | Checksum (Digit 10) |  |
| 11 |  | **?** | Checksum (Digit 1) |  |
| 12 |  | **LF** | Line Feed (Paket Sonu) |  |

# Sıcaklık Değerlerinin Okunması (PID\_EVSE\_TEMP)

Şarj cihazı üzerinde 3 farklı noktada sıcaklık sensörü bulunmaktadır.

Bunlar :

* Mikrodenetleyici Kartı
* Enerji Analizör Kartı
* Role Kartı

Mobil uygulama üzerinde, “Mikrodenetleyici Kartı” na ait sıcaklık bilgisi gösterilecektir. Sıcaklık bilgilerini okumak için GET\_COMMAND sorgusu yapılır.

Veri paketinde 7 nolu slottaki Temperature Type verisi, hangi karttaki sıcaklık bilgisinin okunacağını belirtir.

Mikrodenetleyici Kartı İçin : ‘M’

Enerji Analizör Kartı İçin : ‘E’

Role Kartı İçin : ‘R’ değeri girilmelidir.

Parameter ID : **‘T’**

**GET\_COMMAND :**

Örnek olarak Mikrodenetleyici Kartının sıcaklık bilgisini okumak için aşağıdaki paket gönderilir.

**Sorgu : Bilgisayar -> MCU Board**

|  |  |  |  |
| --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |
|  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |
| 1 |  | **G** | Command Type |
| 2 |  | **T** | Parameter ID (PID) |
| 3 |  | **0** | Parameter Data Length(Digit100) |
| 4 |  | **0** | Parameter Data Length(Digit10) |
| 5 |  | **2** | Parameter Data Length(Digit1) |
| 6 |  | **1** | Connector No |
| 7 |  | **M** | Temperature Type |
| 8 |  | **?** | Checksum (Digit 100) |
| 9 |  | **?** | Checksum (Digit 10) |
| 10 |  | **?** | Checksum (Digit 1) |
| 11 |  | **LF** | Line Feed (Paket Sonu) |

**GET\_RESPONSE :**

**Cevap : MCU Board -> Bilgisayar**

Yukarıda, Mikrodenetleyici Kartına ait olan sıcaklığı öğrenmek için gerekli olan Get Komutu gönderildikten sonra aşağıdaki şekilde bir cevap döner.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **T** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **0** | Parameter Data Length(Digit10) |  |
| 5 |  | **7** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **M** | Temperature Type |  |
| 8 |  | **+** | Temperature Sign( minus or plus) | Sıcaklık : + 45.7 C |
| 9 |  | **0** | Temperature Digit 100 |
| 9 |  | **4** | Temperature Digit 10 |
| 10 |  | **5** | Temperature Digit 1 |
| 11 |  | **7** | Temperature Digit 0.1 |
| 12 |  | **?** | Checksum (Digit 100) |  |
| 13 |  | **?** | Checksum (Digit 10) |  |
| 14 |  | **?** | Checksum (Digit 1) |  |
| 15 |  | **LF** | Line Feed (Paket Sonu) |  |

Gelen cevapta 7 nolu slotta ‘M’ verisi, sıcaklık bilgisinin Mikrodenetleyici kartına ait olduğunu gösterir. 8 Nolu slotta ise sıcaklık değerinin işaretini verir (artı veya eksi). Gelen pakette sıcaklıkğın + 45.7 C derece olduğu anlaşılmaktadır.

Eğer sıcaklık sensöründe bir sorun olursa aşağıdaki gibi bir hata mesajı dönecektir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Slot No** |  | **Örnek Veri** |  |  |
|  |  |  |  |  |
| 0 |  | **STX** | Start of Text (STX) |  |
| 1 |  | **g** | Command Type |  |
| 2 |  | **T** | Parameter ID (PID) |  |
| 3 |  | **0** | Parameter Data Length(Digit100) |  |
| 4 |  | **0** | Parameter Data Length(Digit10) |  |
| 5 |  | **7** | Parameter Data Length(Digit1) |  |
| 6 |  | **1** | Connector No |  |
| 7 |  | **M** | Temperature Type |  |
| 8 |  | **E** | Temperature Sign( minus or plus) | Hata Mesajı |
| 9 |  | **R** | Temperature Digit 100 |
| 9 |  | **R** | Temperature Digit 10 |
| 10 |  | **O** | Temperature Digit 1 |
| 11 |  | **R** | Temperature Digit 0.1 |
| 12 |  | **?** | Checksum (Digit 100) |  |
| 13 |  | **?** | Checksum (Digit 10) |  |
| 14 |  | **?** | Checksum (Digit 1) |  |
| 15 |  | **LF** | Line Feed (Paket Sonu) |  |

NOT : Role kartındaki sıcaklık değeri daha sonra, yüksek ısı durumunda şarj akımını azaltmak ve gerekirse şarj akımını tamamen durdurmak için kullanılacaktır.

# MCU Board’un Resetlenmesi

Linux karttaki Alwinner işlemcinin PE10 (Port E 10) bacağı MCU Board’daki mikrodenetleyicinin reset bacağına bağlıdır. Port E 10’u 100ms kadar SET(1) edip daha sonra RESET(0) yaparak mikrodenetleyici kartı resetlenmiş olur. Yalnız Linux boardu boot edilir edilmez bu pinin ‘0’ konumuna alınması unutulmamalıdır.

gpio-test.64 w e 10 1 // Port E 10 u set et

delayMs(100) // 100 milisaniye bekle

gpio-test.64 w e 10 0 // Port E 10 u reset et

# Tanımlamalar

// Command Type Definitions

**typedef** **enum**{

*GET\_COMMAND* ='G',

*GET\_RESPONSE* ='g',

*SET\_COMMAND* ='S',

*SET\_RESPONSE* ='s'

}COMMAND\_TYPES\_e;

// PID List

**#define** PID\_CONTROL\_PILOT ('C')

**#define** PID\_PROXIMITY\_PILOT('X')

**#define** PID\_RELAY\_CONTROL ('R')

**#define** PID\_CP\_PWM\_CONTROL ('G')

**#define** PID\_LOCKER\_CONTROL ('K')

**#define** PID\_LED\_CONTROL ('L')

**#define** PID\_CURRENT ('I')

**#define** PID\_VOLTAGE ('V')

**#define** PID\_POWER ('P')

**#define** PID\_ENERGY ('E')

**#define** PID\_EVSE\_TEMP ('T')

**#define** PID\_RFID ('N')