**Programın Amacı**

TCPDeviceTester programının amacını anlamak için program yazarak bir cihazla haberleşmeyle ilgili temel kavramları gözden geçirmemiz gerekir.

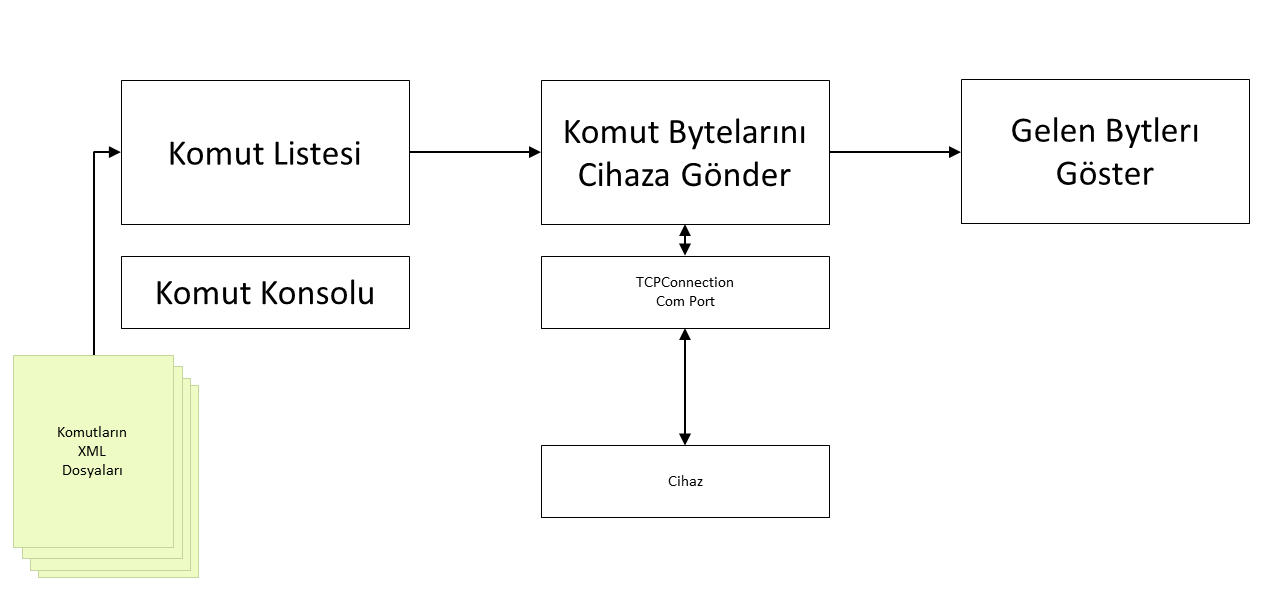
Bir cihazı kontrol etmek için ve cihazdan veri okumak için protokol dediğimiz bir yapı vardır. Bu dökümanın sonunda protokol kavramı biraz daha detaylı açıklanmıştır. Cihazla haberleşmek için cihaza bir byte dizisi göndeririz. Bu byte dizisinin yapısı bir protokoldür. En basit şekliyle başlangıç byte ı, cihaz ID si, komut byte ı, bitiş byte ı ve checksum.

Protokoller cihazdan cihaza değişir. Ayrıca her cihazın kontrol ve veri okuma ile ilgili çok sayıda komutu olabilir.

TCP Device Tester, çok farklı cihazlarla çalışmamız gerektiğinde, bunların komutlarını kaydedip düzenlemeye imkan veren ve bu komutları cihaza gönderip, cihazdan gelen cevapları, hexadesimal, desimal, ASCII vb. değişik formatlarda incelememizi sağlayan bir programdır.

**Bölümleri**

**Algoritmik Akışı**



**Protokol Kavramı**  
Kelime anlamı sözlükte taraflarca kabul edilmiş haberleşme yöntemi şeklinde geçiyor. Cihaz haberleşmesinde protokol bir byte dizisinin tanımıdır. Byte dizisini gönderen ve alan bu tanıma göre byte dizisini yorumladığından, her bir byte ın ne anlama geldiğinde mütabık olduğundan haberleşmeleri mümkündür.  
  
Net bir örnek üzerinden gidelim. Bir lambayı yakıp söndüren basit bir devremiz olsun. Bu devre COM port üzerinden bilgisayara bağlı olsun.   
  
1) Y gönderince lamba yansın, S gönderince sönsün. Bu bir byte tan oluşan bir protokol olmuş olur.   
  
2) Örneği geliştirelim. Lambanın renkli yanma özelliği olsun. YK:kırmızı yan,YY:yeşil yan olsun. Bu durumda iki byte tan oluşan komut + parametre şeklinde bir protokol tanımlamış oluruz.  
  
3) Cihaza bağlı 4 lamba olsun. Aynı anda hepsini birden yakmak isteyelim. Şöyle olabilir: YK YY YS YM. Komutların sırası lambaların sırası ile aynı olsun. Bu durumda burada birinci kırmızı, ikinci yeşil, üçüncü sarı, dördüncü mavi yanar. Gördüğünüz gibi bu protokolde sırayı referans aldık. Ama bir adım ilerisini düşünüp, ilerde dörtten fazla lamba olacağını ve işlerin karışacağını düşünerek şöyle de yapabiliriz.  
  
4) 1YK 2YY 4YM. Dikkat ederseniz üçüncü lambayı içermedik. Çünkü onun durumunu değiştirmek istemedik. Ve onu etkilemeden dördüncü lambayı kontrol edebildik. Böylece daha gelişmiş bir protokol oldu.  
  
**Gerçek Protokol Örnekleri**  
**CIF Protokolü**  
Cihaz haberleşmesinde en çok kullanılan protokollerden biridir.   
Protokol şu şekildedir: STX=123,address,command,ETX=125,checksum.   
Somut örnek: 123,65,49,125,76. (not: Bu CPI marka HPA cihazının statüs bilgisini verir)  
STX başlangıç ETX bitiş byte ı anlamına gelir.   
  
Adres Kavramı:  
Burada protokolü tam anlamak için hızlıca adres kavramından bahsedelim. Bir sistemde birden fazla cihaz olduğunda, bunlar aynı kabloya bağlı olduğunda, istenen cihaza komutu ulaştırmak için ayırdedici parametre adrestir. Komut belirtilen adresteki cihaza ulaşır.   
  
Checksum Kavramı:  
Daha sonra derinlemesine inleceyeceğimiz checksum ı burada protokolü anlatırken gerekli olduğundan hızlıca açıklayalım.   
  
Checksum gönderilen byteların karşı tarafa doğru olarak ulaştığını kontrol etme amaçlı bir yöntemdir. Buradaki örnekte 123,65,49,125 byteları üzerinde yapılan bir işlem sonucu 76 oluşur. 76 checksum dır. Bytelar karşı tarafa ulaştığında karşı tarafta aynı işlemi yapacak ve 76 yı bulamazsa gelen bytelarda bir problem olduğunu anlayacaktır.  
  
Checksum tipleri ve hesaplama yöntemlerini ilgili kısımda inceleyeceğiz.  
  
Dolayısı ile CIF protokolünü bir tabloda özetlersek:

|  |  |  |  |
| --- | --- | --- | --- |
| **TANIM** | **ASCII** | **HEX** | **Ondalık** |
| STX, Başlangıç Byte (START BYTE) | { | 7B | 123 |
| Adres (ADDRESS BYTE) | A | 41 | 65 |
| Komut (COMMAND BYTE) | 1 | 31 | 49 |
| ETX,Bitiş Byte ı (STOP BYTE) | } | 7D | 125 |
| CHECKSUM | L | 4C | 76 |

**Başka Bir Protokol**  
Biraz daha karmaşık sık kullanılan başka bir protokol şu şekildedir.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Protokol | STX | Address | Komut Uzunluğu | Komut | Cheksum Byte1 | CheckSum Byte 2 |
| Örnek | 1 | 2 | 01 | 07 | 224 | 110 |

Görüldüğü gibi burada bitiş byte ı yok. Onun yerine komut uzunluğu kavramı var. Verilen komut uzunluğunda göre byteları sayarak komutun sonucu buluyorsunuz. CIF protokolünde tek byte ken, bu protokolde checksum 2 byte tan oluşuyor. Checksum kısmında bu iki byte ın nasıl elde edildiğini göreceğiz.