

TEDAŞ - MLZ/2024-081

**TÜRKİYE ELEKTRİK DAĞITIM A.Ş.
STRATEJİ GELİŞTİRME DAİRESİ BAŞKANLIĞI**

HABERLEŞME (HEAD-END) YAZILIMI TEKNİK ŞARTNAMESİ

Aralık – 2024

İÇİNDEKİLER

GİRİŞ	ii
1. GENEL	1
1.1. Konu ve Kapsam	1
1.2. Standartlar ve Dokümanlar	2
1.3. Tanımlar ve Kısaltmalar	2
1.4. Yönetmelikler ve Tebliğler	3
2. GENEL ÖZELLİKLER	4
3. YAZILIM FONKSİYONLARI	5
3.1. Kullanıcı Arayüzü ve Raporlama	5
3.2. Modemden Alınacak Kimlik Bilgileri	6
3.3. Acil Durum Bildirimleri	7
3.4. Sayaç Verilerinin Okunması	7
3.5. Enerji Analizörü Verilerinin Okunması	8
4. HABERLEŞME VE GÜVENLİK	10
4.1. Haberleşme	10
4.2. Haberleşme Güvenliği	11
4.3. Yazılımın Güvenliği	12
5. DİĞER SİSTEMLERLE BİRLİKTE ÇALIŞABİLİRLİK	13
6. DENEYLER	14
Ek-A MASS Protokolü ve Ek-BEk-B Mass Entegrasyon Dokümanı (Standart API).....	15

GİRİŞ

Dağıtım şebekesinde kullanılacak Haberleşme (Head-End) yazılımına ait bu şartname, yazılımın teknik özelliklerini, yazılım fonksiyonlarını ve haberleşme protokolünü tanımlamaktadır. Bu şartname, asgari şartların belirlendiği bir teknik şartname değildir. Opsiyonel özellikler ve ekleriyle birlikte bir bütünlük oluşturmaktadır. Bu nedenle şartnamede verilen özellikleri değiştirecek veya yeni özellikler ekleyecek herhangi bir ilave şartname olmaksızın kullanılacaktır.

TASLAK

HABERLEŞME (HEAD-END) YAZILIMI TEKNİK ŞARTNAMESİ

1. GENEL

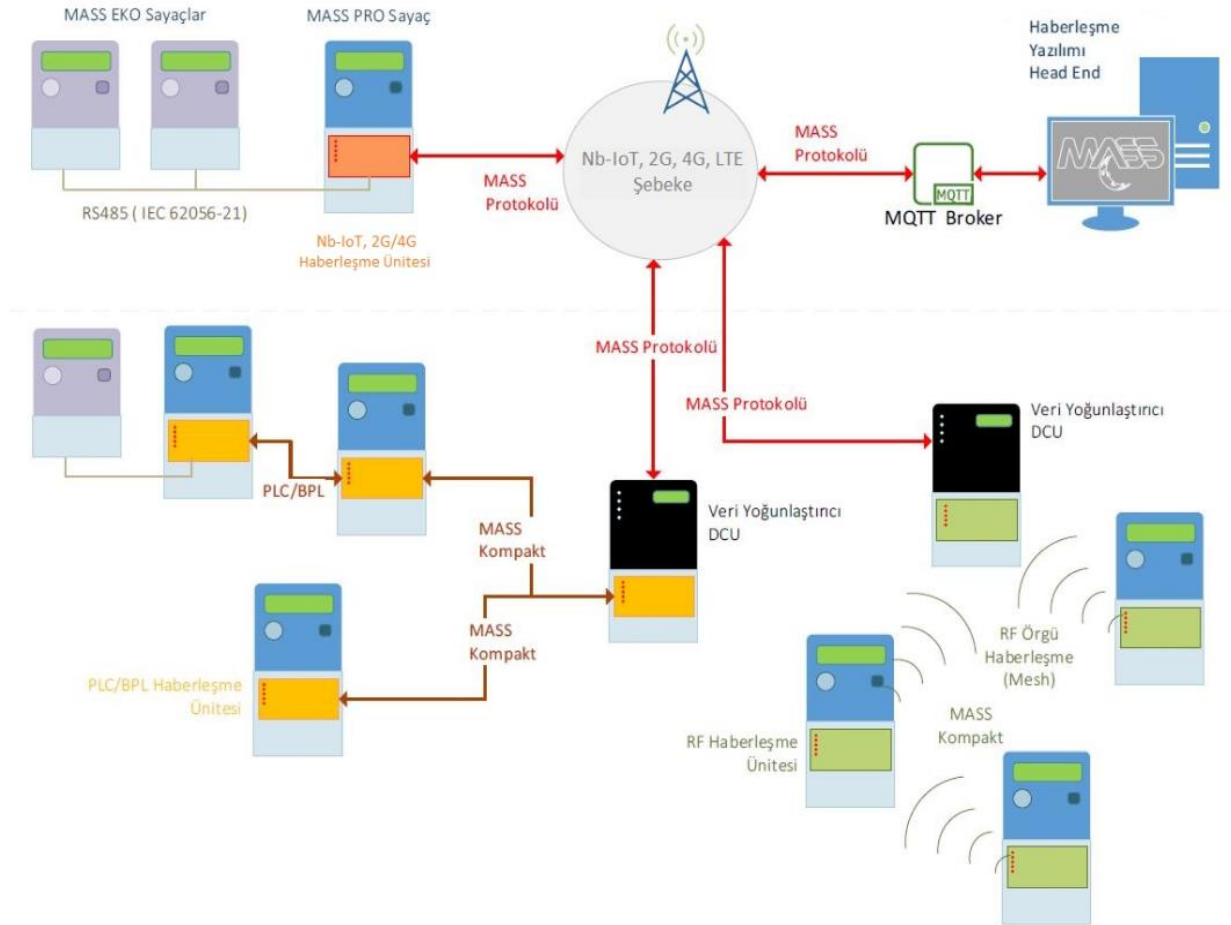
1.1. Konu ve Kapsam

Haberleşme (Head-End) Yazılımı Teknik Şartnamesi; Elektrik Piyasası Ölçüm Sistemleri Yönetmeliği kapsamında yer alan dağıtım sisteminde kullanılacak haberleşme yazılımının özelliklerinin belirlenmesi amacıyla hazırlanmıştır.

Bu şartname dağıtım sisteminde kullanılacak Haberleşme (Head-End) Yazılımının haberleşme ve güvenlik özellikleri, yazılım fonksiyonları ve haberleşme protokolünü kapsar.

Bu şartnamede bundan sonra Haberleşme (Head-End) Yazılımı “Head-End”, haberleşme ünitesi ve modüler modem, “modem” olarak tanımlanacaktır.

Resim 1 – MASS Prensip Şeması



1.2. Standartlar ve Dokümanlar

Bu şartname ve eklerinde aksi belirtilmedikçe; Head-End aşağıdaki Türk Standartları Enstitüsü (TSE), Avrupa Elektroteknik Standart Komitesi (EN) ve Uluslararası Elektroteknik Komisyonu (IEC) standartlarının en son baskılarına uygun olacak ve standartlarda yer alan kriterleri sağlayacaktır. Aşağıdaki tabloda yer almayan ancak teknik şartnamenin ilerleyen bölümlerinde atıfta bulunulan standartların da yürürlükteki en son baskıları esas alınacaktır.

Çizelge 1 - Standartlar ve Dokümanlar

Standart Numarası (TS)	Uluslararası Standart Numarası (IEC, EN, ISO)	Standart Adı
TS EN 62052-21	IEC 62052-21	Elektrik ölçme donanımı (a.a.) - Genel kurallar, deneyler ve deney şartları - Bölüm 21: Tarife ve yük kontrol donanımı
TS EN 62056-6-1	IEC 62056-6-1	Elektrik ölçüm veri değişimi - DLMS / COSEM paketi - Bölüm 6-1: Nesne Tanımlama Sistemi (OBIS)
TS EN 62056-6-2	IEC 62056-6-2	Elektrik Ölçme veri alışverişi - DLMS / COSEM suite - Bölüm 6-2: COSEM arabirim sınıfları
	ISO 8601	Date and time format

1.3. Tanımlar ve Kısaltmalar

MASS	: Milli Akıllı Sayaç Sistemleri
GSM	: Global System for Mobile (Mobil İletişim İçin Küresel Sistem)
Nb-IOT	: Narrow Band Internet of vThings
PLC	: Power line Communication (also known as power-line carrier)
RF	: Radio frequency (Radyo frekansı)
AKM	: Ana Kontrol Merkezi
CRM	: Customer Relationship Management (Müşteri İlişkileri Yönetimi)
MDM	: Meter Data Management (Sayaç Veri Yönetimi)
OMS	: Outage Management System (Kesinti Yönetim Sistemi)
SCADA	: Supervisory Control and Data Acquisition (Merkezi Denetleme Kontrol ve Veri Toplama)
SOA	: Service Oriented Architecture (Servis Odaklı Mimari)
ID	: Identification Number (Kullanıcı Numarası)
MQTT	: Message Queuing Telemetry Transport
TCP/IP	: Transmission Control Protocol/Internet Protocol (TCP/IP Protokol Takımı)
OBIS	: Object Identification System (Nesne Tanımlama Sistemi)
WAN	: Wide Area Network (Geniş Alan Ağı)
LAN	: Local Area Network (Yerel Alan Ağı)
IMEI	: International Mobile Equipment Identity

UDP	: User Datagram Protocol
SMTP	: Simple Mail Transfer Protokol
HTTP	: Hyper Text Transfer Protocol / Hiper Metin Transfer Protokolü
FTP	: File Transfer Protocol / Dosya Transfer Protokolü
DLMS	: Dynamic Lane Merge Systems
SSH	: Secure Shell (Güvenli Kabuk)
SSL	: Secure Sockets Layer (Güvenli Giriş Katmanı)
VEE	: Validation, Editing, Estimation
DCU	: Data Concentrator Unit (Veri Toplama Birimi)
WFM	: Work Force Management (İş Gücü Yönetimi)
IAM	: Identity and Access Management (Kimlik ve Erişim Yöntemi)
HU	: Haberleşme Ünitesi
MM	: Modüler Modem
Birlikte	Farklı marka Head-End yazılımı, modem, sayaç ve uç birimlerin
Çalışabilirlik	: birbirleriyle ve Head-End yazılımının MDM, CRM ve OMS gibi diğer
İlkesi	sistemlerle uyumlu olarak çalışması

1.4. Yönetmelikler ve Tebliğler

Head-End'in yazılım fonksiyonları, haberleşme ve güvenlik kriterlerinde;

- Elektrik Piyasası Ölçüm Sistemleri Yönetmeliği,
- Enerji Sektöründe Siber Güvenlik Yetkinlik Modeli Yönetmeliğinin

yürürlükteki en son baskılarının ilgili hükümlerine uyulacaktır.

2. GENEL ÖZELLİKLER

- (1) Head-End sahada bulunan modemler ile çift yönlü haberleşmeyi sağlayacak, sayaç ve uç birimlerden gelen verileri işleyip veri tabanına kaydedecek ve bu verileri daha sonra kullanabilmek amacıyla istenilen süre boyunca hazır bulunduracaktır. Bu amaçla, modem ile modeme bağlı sayaç ve uç birim bilgilerini tarihsel olarak saklayabilecek, işleyebilecek ve sistem performansını raporlayabilecektir.
- (2) Head End'in iki temel işlevi bulunacaktır.
 - Sahadaki modemlerin merkezden yönetilmesi, sayaç ve uç birim verilerinin merkezden alınması ve değiştirilebilir parametrelerinin programlanması (haberleşme, okuma ve programlama gibi tüm işlemler) (Ek-A)
 - Diğer sistemler (MDM, OMS, CRM, vb.) ile veri iletişimi(Ek-B)
- (3) Bu iki temel işlev dışındaki ve bu şartnamede yer almayan analiz, raporlama, yorumlama, alarm, bildirim, veri doğrulama değiştirme ve tahminleme(VEE) gibi tüm diğer yazılım gerekleri bu şartname kapsamı dışında tutulacaktır.
- (4) Head-End; yedeklilik, bütünsellik, yapısalılık, güvenlik ve güvenilebilirlik özelliklerine sahip ve artan ihtiyaçlara göre genişletilebilir bir yapıda olacaktır.
- (5) Head-End, veriyi işleyerek kaydedecek ve modemden gelen esas veri üzerinde herhangi bir değişiklik yapmayacaktır.
- (6) Head-End; modem, sayaç ve diğer uç birimlerden alınan verileri doğrulama işlemine tabi tutacaktır. Doğrulamada; bütünlük doğrulaması, kimlik doğrulaması, format yönünden uygunluk kontrolleri yapılacaktır. Bu işlem tanımlanan adette tekrarlandığı halde doğrulama yapılamamış ise LOG kaydı oluşturulacaktır.
- (7) Head-End, veri sıkıştırma algoritmalarının kullanımına izin verecek bir yapıda olacaktır.
- (8) Head-End, birden fazla işletim sistemini destekleyecek ve Microsoft, Oracle gibi bir veritabanı sunucusu ile çalışacak yapıda olacaktır.
- (9) Head End, modemleri ve bu modemlere bağlı sayaç ve uç birimlerin envanterini saklayacaktır.

3. YAZILIM FONKSİYONLARI

(10) Head End, önceden tanımlanmış periyotlar ve yönergelere göre sayaçlar ve/veya uç birimler üzerinde istenilen işlemleri gerçekleştirebilecektir.

(11) Head-End en az aşağıdaki temel fonksiyonları yerine getirecektir.

- Uzaktan sayaç veya uç birim okuma,
- Uzaktan sayaç açma/kesme,
- Sayaçların ve uç birimlerin uzaktan programlanması
- Sayaçlardan ve uç birimlerden merkeze gelen alarm bilgilerinin(FF, GF gibi) okunması
- Sayaçlar ve uç birimler için tanımlanmış yönergelerle paket verilerinin okunması
- Modemlerin uzaktan programlanması
- Modemlerin parametrelerinin değiştirilmesi ve yönergelerin yüklenmesi
- Modemlerin gömülü yazılımının (firmware) güncelleme işlemleri
- Haberleşme ile ilgili hata raporlama (sayaç veya uç birim hatası ve/veya varsa veri toplayıcı(DCU) ile diğer iletişim ekipmanı hataları)

(12) Head-End yukarıda tanımlanan temel fonksiyonlarla beraber aşağıdaki diğer fonksiyonları da yerine getirecektir.

- Head-End aynı iş emirlerine ilişkin yanlış komut oluşturulmasına izin vermeyecektir.
- Oluşturulan iş emirleri düşük, orta ve yüksek öncelik seviyeleri ile tanımlanabilecektir.
- Head-End, okuma verilerinin eksik olması durumunda okuma işleminin başarısız olduğu sayaçlar ve diğer uç birimler üzerinde önceden tanımlanmış aralıklarla okuma işlemini istenen sayıda tekrar edecektir.
- Yapılan tüm işlemlerin sonucu başarılı ya da başarısız olarak kaydedilecek, başarısız işlemler için çeşitli aksiyonlar tanımlanabilecektir.
- Sistem envanteri amaçlı (modemler, sayaçlar ve diğer uç birimler vb.) raporlar üretilebilecektir.
- Haberleşme sistemi dahilindeki çevrim içi (online) ve çevrimdışı (offline) olan modemler için sorgulama yapılabilecektir. Belirlenen bir zaman dilimindeki çevrim içi ve çevrim dışı süreleri ile başarılı ve başarısız haberleşme adetleri sorgulanabilecektir.
- Diğer sistemlerden gelen iş emirleri için öncelikler belirlenebilecek, bu öncelik sırasına uygun olarak gelen emirleri işleyebilecek ve cevaplarını döndürebilecektir.

3.1. Kullanıcı Arayüzü ve Raporlama

(13) Head-End, web tarayıcı tabanlı olacak ve sistemin kolayca yönetilmesine imkan verecek bir arayüze sahip olacaktır.

(14) Head-End kullanıcı arayüzü, kullanıcılar için yetkilendirmeye göre müdahale imkanı tanıyacaktır. Kullanıcı hesaplarının kurulumu, uygun yetkilendirilmiş yönetici rolündeki kullanıcı tarafından kullanıcı dostu bir web tabanlı arayüz aracılığıyla yapılacaktır.

- (15) Head End, İngilizce ve Türkçe dil arayüzüne sahip olacak ve menüler, çevrimiçi yardım dosyası ve eğitim materyallerini içerecektir. Diller arasında geçiş, kullanıcı oturum açma ekranında ayarlanarak gerçekleştirilecektir.
- (16) Head-End, kullanıcı arayüzü, fonksiyonların ve yönergelerin kullanımında kullanıcı hatasını önleyecek kontrol ve düzeltme özelliklerine sahip olacaktır.
- (17) Head-End, sisteme bağlı modemleri ve bu modemlere bağlı sayaç ve uç birimlere ait envanter ile bunlara bağlı arıza performans bilgilerinin istenildiğinde kullanıcı arayüzünde gösterecek ve raporlayacaktır.
- (18) Head-End, iletişim performansının kalitesine dair rapor sunacaktır.
- (19) Head-End, başarımlar için en az aşağıdaki performans göstergelerini saklayacak, istenildiğinde kullanıcı arayüzünde gösterecek ve raporlayacaktır. Başarım raporu, her gün için yüksek/düşük sıcaklıkları ve günlük yağış miktarını gösterecek şekilde sunulacaktır.
- Gerçekleştirilen görev sayısı
 - Başarısız görev sayısı
 - Başarılı görev sayısı
 - Saat başına gerçekleştirilen maksimum görev sayısı
 - Planlı okuma görevleri için ortalama yürütme süresi
- (20) Head-End, modemlere atanmış yönergeleri ve alarm bilgilerini saklayacak, istenildiğinde kullanıcı arayüzünde gösterecek ve raporlayacaktır.
- (21) Tüm raporlar ekran, yazıcı, Excel, Xml, Csv, Html, Pdf, Text ortamlarında alınabilecek ve istenildiğinde periyodik olarak e-posta ile gönderilecektir.

3.2. Modemden Alınacak Kimlik Bilgileri

- (22) Head-End, modemden en az aşağıdaki kimlik bilgilerini istenildiğinde alacak, saklayacak ve işleyecektir.
- Flag Kodu (Rumuz)
 - Seri numarası
 - Model
 - Imei
 - Cihaz Tipi (MM, HU, DCU vb.)
 - Haberleşme Tipi (2G/4G, Nb-IoT, PLC, RF vb.)
 - Üretim Tarihi
 - Protokol Versiyonu
 - Cihaz Versiyonu
 - Sunucu bilgileri
 - Haberleşme Arayüzleri
 - Seri portlar
 - Giriş Çıkış Arayüzleri
 - Haberleşme Ön Ayarları (MQTT Prefix Ayarları)

(23) Head-End, modemden en az aşağıdaki performans bilgilerini istenildiğinde alacak, saklayacak ve işleyecektir.

- Son üç aylık sinyal seviyesi bilgileri (kullanılıyorsa)
- Son üç aylık erişim bilgileri

3.3. Acil Durum Bildirimleri

(24) Modemlerin zaman kritik işlemlerini Head-End'e iletmeleri için ayrılmış bir MQTT kanalı bulunacaktır. Enerji kesilmesi gibi zaman kritik olan işlemler bu kanal vasıtasıyla öncelikli olarak Head-End'e ulaştırılacak ve işlemler önceliklendirilecektir.

(25) Acil durum bildirimlerinin işleyişi Ek-A'da tanımlanan Alarm/Bildirim fonksiyonuna uygun olacaktır.

3.4. Sayaç Verilerinin Okunması

(26) Head-End, sayaç ve/veya uç birimlerle ilgili olarak en az aşağıdaki kimlik bilgilerini istenildiğinde alacak, saklayacak ve işleyecektir.

- Flag Kodu
- Seri numarası
- Sayaç nesli
- Sayaç versiyon no
- Modeli
- Model Kodu
- Üretim tarihi (model yılı)
- Cihaz Tipi (Elektrik, su, gaz vb.)

(27) Head-End, sayaç ve/veya uç birimlerle ilgili olarak en az aşağıdaki tahakkuka esas bilgileri istenildiğinde alacak, saklayacak ve işleyecektir.

- Toplam enerji endeksleri (aktif, endüktif ve kapasitif reaktif değerler)
- Çift yönlü sayaçlar için veri toplam enerji endeksleri (aktif, endüktif ve kapasitif reaktif değerler)
- Tarife dilimleri için değerler (T1-T4)
- Çift yönlü sayaçlar için veri tarife dilimleri için değerler (T1-T4)
- En yüksek talep gücü ve tarihi
- Çift yönlü sayaçlar için veri en yüksek talep gücü ve tarihi

(28) Head-End, sayaç ve/veya uç birimlerle ilgili olarak en az aşağıdaki arıza, hata ve alarm bilgilerini istenildiğinde alacak, saklayacak ve işleyecektir.

- Gövde kapağı açılma uyarısı
- Klemens kapağı açılma uyarısı
- Kalibrasyon tarihi
- Tarife değişiklik tarih-saati

- Zayıf pil uyarısı
- Enerji kesinti adetleri ve tarihleri
- Gerilim ve akım uyarı adetleri ve tarihleri
- Manyetik müdahale uyarı adetleri ve tarihleri
- FF kodları
- GF kodları (varsa)
- Röle Durumu
- Sayısal giriş durumu
- Yük profili periyodu
- Demant ölçüm periyodu
- Yük değeri kontrol periyodu
- Yük kontrol değeri
- İleri geri saat durumu ve tarihleri
- En son readout okuma tarihi
- En son yük profili okuma tarihi

(29) Head-End, sayaçta bulunan yük profillerini alacak, en az üç ay saklayacak ve işleyecektir.

3.5. Enerji Analizörü Verilerinin Okunması

(30) Head-End, enerji analizörleriyle ilgili olarak en az aşağıdaki kimlik bilgilerini istenildiğinde alacak, saklayacak ve işleyecektir.

- Firma Kodu
- Ürün Modeli
- Ürün Seri No
- Program Versiyon
- Donanım Versiyon
- Ortalama Değer Periyodu
- RMS Ortalama Değer Periyodu
- Akım Trafosu Oranları
- Zaman (UTC)
- Cihaz ID
- Baud Rate Ayarı
- Pil Durumu
- Aktif Fider Sayısı

(31) Head-End, enerji analizörleriyle ilgili olarak en az aşağıdaki ölçüm bilgilerini istenildiğinde alacak, saklayacak ve işleyecektir.

- Elektriksel parametreler
- Maksimum ortalama değerler
- Enerji ölçümleri tüketim (+)
- Enerji ölçümleri üretim (-)
- Harmonikler (THB)
- Akım harmonikleri
- Gerilim harmonikleri

- Alarm parametre kayıtları
- Anlık Kesinti Kayıtları
- Geriye Dönük THB kayıtları
- Kesinti kayıtları
- Enerji yük profili kayıtları
- Parametre yük profili kayıtları

(32) Head-End, enerji analizörleriyle ilgili olarak en az aşağıdaki arıza, hata ve alarm bilgilerini istenildiğinde alacak, saklayacak ve işleyecektir.

- Yüksek Gerilim Alarmları
- Nötr Yüksek Gerilim Alarmı
- Düşük Gerilim Alarmları
- Yüksek Akım Alarmları
- Nötr Yüksek Akım Alarmı
- Düşük Akım Alarmları
- Düşük Güç Faktörü Alarmları
- Faz Sırası Alarmları

TASLAK

4. HABERLEŞME VE GÜVENLİK

- (33) Head End yazılımı, haberleşme merkezi sunucusundan modeme (pull) ve modemden haberleşme merkezi sunucusuna (push) veri akışını destekleyecektir.
- (34) Head-End, sayaçlar ve modemlerin zaman senkronizasyonunu merkezi olarak yapılabilmesini sağlayacaktır.
- (35) Head-End, sayaçlar üzerindeki bilgilerden tercih edilenlerin (sadece tüketim bilgileri, sadece yük profili bilgileri, sadece aktif-reaktif tüketim değerleri, sadece olay kayıtları vb. gibi) transfer edilebilmesini sağlayacaktır.
- (36) Head-End, sayaçlardaki değiştirilebilir parametrelerin programlamasını ve modem üzerinde izin verilen yazılımsal güncellemeler, tanımlamalar ve parametre değişikliklerinin yapılabilmesini sağlayacaktır.
- (37) Head-End, aynı modem üzerinden birden çok aynı veya farklı marka model sayaç ile haberleşmeyi sağlayacak ve istenilen verileri alabilecektir.

4.1. Haberleşme

- (38) Head-End, standart internet protokollerini (TCP / IP / UDP / POP3 / SMTP / HTTP(S) / (S)FTP / MQTT(S) vb.) ve bağlantılarını destekleyecek yapıda olacaktır.
- (39) Head-End, MASS protokolü haricinde, DLMS / MODBUS / MBUS / WMBUS / IEC 62056-21 protokollerini de destekleyecektir.
- (40) Head-End, şimdiki veya gelecekteki sayaç veya uç birim verilerine şeffaf veya yönergeli okuma (anlık ve periyodik) metodu ile modem üzerinden ulaşacaktır.
- (41) Head-End, markadan ve protokolden bağımsız sayaç veya diğer uç birimlerle haberleşme metodlarını kullanarak çift yönlü haberleşmeyi destekleyecektir.
- (42) Şeffaf yapıda Head-End, okumak istediği sayacı veya uç birimi geniş alan ağıyla (WAN) anlık olarak sayaç veya uçbirimin protokolü aracılığıyla modeme bağlı seri haberleşme portunun soket bağlantısını yöneterek okuyacaktır.
- (43) Seri haberleşmede kullanılacak soket port bilgileri MASS protokolü üzerinden değiştirilebilecektir. Protokol ve port bilgileri Head-End tarafında bulunacaktır.

Çizelge 2 - Şeffaf Yapı Örneği

LOCAL AREA NETWORK	Mesaj Yönü	WIDE AREA NETWORK
/?! [CR][LF]	<	/?! [CR][LF]
/AEL4<3>AEL.MF.14 [CR][LF]	>	/AEL4<3>AEL.MF.14 [CR][LF]
[ACK] 041 [CR][LF]	<	[ACK] 041 [CR][LF]
[SOH] R2 [STX] 0.0.0() [ETX]P	<	[SOH] R2 [STX] 0.0.0() [ETX]P
[STX] 0.0.0(123456789) [ETX]0	>	[STX] 0.0.0(123456789) [ETX]0
[SOH] R2 [STX] 0.9.2() [ETX][<	[SOH] R2 [STX] 0.9.2() [ETX][

[STX] 0.9.2(13-12-12) [ETX];	>	[STX] 0.9.2(13-12-12) [ETX];
[SOH] B0 [ETX]q	<	[SOH] B0 [ETX]q
<p>Not 1: Yukarıdaki örnekte görülebileceği gibi sayaç veya uç birim bilgileri MASS protokolünden bağımsız olarak sayaç veya uç birim protokolünün Head-End'den yönetilmesi ile sisteme aktarılmış olur.</p> <p>Not 2: Bu sistem sayaç veya uçbirimin protokolden bağımsız anlık okunmasına olanak sağlarken geniş alan network'ünü sayaç haberleşme ünitesi haberleşme esnasında meşgul ettiği için özel durumlarda kullanılır periyodik okumalarda tercih edilmez.</p>		

- (44) Head-End yönergeli okuma yapısında anlık veya periyodik olarak okumak istediği sayaç veya uç birim için JSON yapısı ile (Javascript Object Notation) yönergeler oluşturulmasını sağlayacaktır.
- (45) MASS Protokolünde yer alan standart yönergeler vasıtasıyla sayaç veya uç birimin anlık veya periyodik olarak okunmasına olanak sağlanacaktır. Programlanmış komut dizileri yapı örnekleri Ek-A'da verilen MASS Haberleşme Protokolünde gösterilmiştir.
- (46) Head-End tarafından yönergeli okuma yapısında baudrate değişimi, framing ayarları, veri transferi, bekleme, gelen verinin parametreye atanması gibi işlemler, işlem tipi içinde modeme tanımlanabilecektir. Bu yapının detayları MASS Protokolü dokümantasyonunda yer almaktadır (Ek-A).
- (47) İstenen veri modemden Head-End'e geldiğinde, Head-End; sayaç flag-seri numarası ile modem flag-seri numarası bilgisini ve okumuş olduğu veri bütününe sisteme kayıt edecektir.
- (48) Head-End, modemlerin ön tanımlı tanıtım fonksiyonu (identification) bilgisini kullanarak sisteme bağlanmasını sağlayacaktır.
- (49) Sistemdeki taşıyıcı protokol MQTT(S) olacaktır. Güvenlik tanımlamaları, MQTT broker sunucularının tanımlanması ve gerekli diğer kriterler MASS Protokolü (Ek-A) dökümanında yer almaktadır.
- (50) Head-End, MQTT(s)'nin sağlanamadığı özel durumlarda soket (TCP/IP, UDP) haberleşme protokollerinin de kullanılmasını destekleyecektir. Bu haberleşme protokolünün detayları Ek-A'da verilen MASS Protokolü dokümantasyonunda yer almaktadır.
- (51) Modemlere bağlı sayaç veya uç birimlerin envanter bilgisi Head-End'de yer alan ve uygulama örneği Ek-B'de verilen Web API ile sorgulanabilir yapıda olacaktır.

4.2. Haberleşme Güvenliği

- (52) Head-End'in veri aktarım güvenliği genel olarak özel ağlarla (APN) güvenli bir şekilde sağlanacaktır. Bunun dışındaki istenilen özel durumlarda Head End ile modemler arasındaki veri aktarımındaki güvenlik, protokol bazlı güvenlik uygulamasıyla sağlanacaktır. Bu durumlarda MQTT protokolünün üzerine TLS güvenlik katmanı eklenmiş versiyonu olan MQTT(s) protokolü kullanılacaktır.
- (53) MQTT(s) haberleşmesi ile uçtan uca şifrelenen (encrypted) veri ancak Head-End tarafından şifresinin çözülmesi (decrypted) ile anlamlı ve anlaşılır hale getirilebilecektir.

- (54) Head-End, modem gömülü yazılım güncellemeleri, uzaktan kapama ve açma komutları, uzaktan yapılan tarih ve saat ayarlamaları, sayaç programlama işlemleri gibi kritik işlemlerde MQTT(s) protokolünü kullanabilecektir.
- (55) Head-End şifreli ve şifresiz haberleşmeler için modemlerin ve sunucuların farklı portları kullanılacaktır.
- (56) Veri iletim güvenliğinin MQTT(s) dışında sağlanması talep edildiği durumlarda modemlerden Head-End'e doğru iletilecek verilerde dışarıdan müdahale yapılmasını önlemek için veriye eşlik eden bir anahtar (TOKEN) kullanılacaktır. Bu anahtarın üretimi ve üretilen anahtarın doğrulanması işlemi için ilave olarak kurulan IAM (Identity and Access Management) sistemi ve sunucusu kullanılacaktır.

Not: IAM sistemi kullanılması durumunda haberleşme modülü güvenliği istenilen verileri iletmeden önce bu sunucuya bağlanarak kendi öz bilgileri ve IMEI bilgileri ile birlikte bir anahtar (TOKEN) oluşturulmasını ister. IAM sistemi bu bilgilere özel anahtarı oluşturup haberleşme modülüne gönderir. Haberleşme modülü ise bu anahtarı alıp veriyi şifreleyerek Head-End yazılımına iletir. Head-End kendine gelen Anahtar+Veri katarını inceleyip aldığı anahtarı IAM sunucusuna bağlanarak doğrular. Doğrulanana anahtara eşlik eden veri Head-End sisteminde işlenir.

4.3. Yazılımın Güvenliği

- (57) Head-End'in tüm bileşenlerinde, gömülü ve basit parola kullanımı engellenecektir.
- (58) Head-End "Güvenli Geliştirme ve Kodlama Standartları"na uygun olacaktır. Geliştirilen kodlar, güvenlik açıklarına karşı düzenli olarak statik ve dinamik analiz araçları ile kontrol edilmeden yazılıma dahil edilmeyecektir.
- (59) Head-End'de üçüncü taraf kodlar kullanılması durumunda, bu kodlar herhangi bir arka kapı (backdoor) açığı oluşturmayacak ve güvenlik açıkları açısından kapsamlı bir inceleme yapılmadan yazılıma dahil edilmeyecektir.
- (60) Kaynak kodu kontrol sistemleri kullanılarak yazılımda yapılan tüm değişiklikler izlenerek yazılımın güncel ve güvenilir olması sağlanacaktır.
- (61) Head-End'in güncellemelerinde geri alma (rollback) özelliği sağlanacak, sorunlu bir güncelleme durumunda önceki güvenli sürüme dönülebilecektir.
- (62) Head-End'deki tüm kayıtlar ve loglar güvenli bir şekilde saklanacak ve yalnızca yetkili kullanıcılar tarafından erişilebilir olacaktır.
- (63) Head-End'de yapılan tüm işlemler için yetki seviyesi tanımlanacaktır. Bu yetkiler kullanıcı bazlı, sınırsız veya süreli olarak verilebilecektir. Sayacı okumak, programlamak, açma kesme komutu icra etmek, modem değişkenlerini programlamak ve yazılım yönetimi için de farklı seviyelerde yetkiler tanımlanabilecektir. Yetkisiz ve izinsiz erişimler engellenecektir.

5. DİĞER SİSTEMLERLE BİRLİKTE ÇALIŞABİLİRLİK

- (64) Head-End; farklı marka modem, sayaç ve diğer uç birimlerle birlikte çalışabilirlik ilkesine uygun olarak çalışacaktır. Ayrıca Head-End; MDM, CRM ve OMS gibi sistemlerle bu ilkeye uygun olarak haberleşme ve veri akışını sağlayacaktır.
- (65) Head-End; MDM, CRM ve OMS ile entegrasyon için standart arayüzleri sağlayacaktır. Head-End ayrıca saha kurulumlarının yönetilmesi için kullanılan WFM gibi uygulamalarla da birlikte çalışabilecektir. Bu ve benzeri uygulamalar Head-End ile EK-B de yer alan API formatında veri alabilecek, gerekli olduğu durumlarda saha cihazlarına istenilen işlerin yapılabilmesi için Head-End'e görev atayabilecektir.
- (66) Head-End elektrik dağıtımında kullanılan diğer sistemlerin en az aşağıdaki ana başlıklardaki görevlerini yerine getirecektir.
- Sayaç endeks bilgilerinin aktarılması
 - Diğer kayıtlı sayaç bilgilerinin aktarılması
 - Sayaç okuma iş emri oluşturulması
 - Kayıtlı yük profili bilgilerinin aktarılması
 - Yük profili okuma iş emri oluşturulması
 - Sayaç tarife değiştirme iş emri oluşturulması
 - Sayaçlardan gelen alarm bilgilerinin işlenmesi
 - Sayaçlardan gelen kesinti bilgilerinin işlenmesi
 - Sayaç enerji kesme/verme komut işlemleri
 - Sayaç parametre değiştirme iş emirlerinin oluşturulması
 - Modem parametre değiştirme işlemleri
 - Modem ayar bilgilerinin alınması
 - Modem arıza bilgilerinin alınması ve resetlenmesi

6. DENEYLER

(67) Head-End ařađıdaki yazılım testlerinden başarıyla gemiř olacaktır.

- Birlikte alıřılabilirlik Testleri (Saya ve u birimlerle entegrasyon)
- Yazılım Simülasyon Testleri (Uygulamalarla entegrasyon)
- Yazılım Güvenlik Testleri (Fuzz, Linting, Unit testleri)

TASLAK

Ek-A MASS Protokolü**MİLLİ AKILLI SAYAÇ SİSTEMLERİ (MASS)
HABERLEŞME PROTOKOLÜ**

1	Protokol Tanıtımı.....	4
1.1	Protokol Genel Yapısı	4
1.2	Veri Tipleri	7
2	Tanımlar ve Kısaltmalar	8
3	ASCII/Hex/Sayısal Dönüşüm Tablosu.....	10
4	MQTT	12
4.1	MQTT Tanımı	12
4.2	Kullanılacak Versiyon.....	12
4.3	Hizmet Kalitesi Tasarımı (QoS).....	12
4.4	Topicler.....	12
5	TCP/IP.....	14
5.1	Senkron Haberleşme	14
5.2	Asenkron Haberleşme	14
6	REST API.....	15
7	Header.....	16
7.1	Başlık Formatı.....	16
8	CRON.....	19
8.1	CRON Tanımı.....	19
9	Fonksiyonlar	22
9.1	ACK.....	22
9.1.1	Çalışma zamanı.....	22
9.1.2	Çalışma şekli	22
9.1.3	Mesaj Formatları	23
9.2	Identification.....	24
9.2.1	Çalışma zamanı.....	24
9.2.2	Çalışma şekli	24
9.2.3	Mesaj Formatları	25
9.3	Heartbeat.....	35
9.3.1	Çalışma zamanı.....	35
9.3.2	Çalışma şekli	35

9.3.3	Mesaj Formatları	36
9.4	Log	37
9.4.1	Çalışma zamanı.....	37
9.4.2	Çalışma şekli	37
9.4.3	Mesaj Formatları	37
9.4.4	Log tipleri	39
9.5	Alarm	40
9.5.1	Çalışma Zamanı	40
9.5.2	Çalışma Şekli.....	40
9.5.3	Mesaj Formatları	41
9.5.4	Alarm Tipleri	41
9.6	Configuration	44
9.6.1	Çalışma zamanı.....	44
9.6.2	Çalışma şekli	44
9.6.3	Mesaj Formatları	45
9.6.4	Başarılı işlem bildirimi.....	54
9.7	Reset	55
9.7.1	Çalışma zamanı.....	55
9.7.2	Çalışma şekli	55
9.7.3	Mesaj Formatları	56
9.8	Firmware Güncelleme	57
9.8.1	Çalışma zamanı.....	57
9.8.2	Çalışma şekli	57
9.8.3	Mesaj Formatları	58
9.9	Takvimsel İşlemler	59
9.9.1	Takvimsel Okuma Listesi.....	59
9.9.2	Takvimsel Okuma Çıkarma	62
9.9.3	Takvimsel Readout.....	64
9.9.4	Takvimsel Custom Profil.....	66
9.10	Aydınlatma Açma Kapama.....	69
9.10.1	Çalışma Zamanı	69
9.10.2	Çalışma Şekli.....	69
9.10.3	Mesaj Formatları	70
10	Yönergeler.....	72
10.1	Genel Bilgiler.....	72
10.2	Yönerge Listesi	85
10.3	Yönerge Ekle	88
10.4	Yönerge Sil.....	92

10.5	Örnek Yönerge Ekle (OBIS Listesi)]	94
10.6	Örnek Yönerge Ekle (Readout)	99
10.7	Örnek Yönerge Ekle (Profil)	101
10.8	Örnek Yönerge Ekle (Obis Yazma)	106
10.9	Yönerge Çalıştırma (Obis Listesi)	110
10.10	Yönerge Çalıştırma (Readout)	113
10.11	Yönerge Çalıştırma (Yük Profili)	116
10.12	Yönerge Çalıştırma (Obis Yazma)	119
10.13	Yönerge Çalıştırma (Su Sayacı)	121
11	Örnek Ölçüm Cihazı Mesajları	124
11.1	Uzun Okuma Paketi	124
11.2	Programlama Modunda Okuma	133

1 Protokol Tanıtımı

Milli Akıllı Sayaç Sistemleri (MASS) Head-End haberleşme protokolü, haberleşme ünitesi ile haberleşme ünitesine bağlı sayaç ve uç birimlerin haberleşmesini sağlamak amacıyla tasarlanmış bir protokoldür.

Head-End haberleşme protokolü **pull** (haberleşme merkezi sunucusundan haberleşme ünitesine) ve **push** (haberleşme ünitesinden haberleşme merkezi sunucusuna) mesajları desteklemektedir. Mesajlarda bir **header** yapısı bulunmaktadır.

1.1 Protokol Genel Yapısı

Protokolde haberleşme **JSON** veri yapısında sağlanacaktır.

MQTT

TCP/IP

REST API

JSON Veri

{...}

Örnek Mesajlar

ASCII (Json)

Hex

```
{"device":{"flag":"XYZ","serialNumber":"ABCXYZ001"},"function":"identification","requestMethod":"async","referenceId":"ABC123XYZ"}
```

ASCII (Json)

Hex

```
7B 22 64 65 76 69 63 65 22 3A 7B 22 66 6C 61
67 22 3A 22 58 59 5A 22 2C 22 73 65 72 69 61
6C 4E 75 6D 62 65 72 22 3A 22 41 42 43 58 59
5A 30 30 31 22 7D 2C 22 66 75 6E 63 74 69 6F
6E 22 3A 22 69 64 65 6E 74 69 66 69 63 61 74
69 6F 6E 22 2C 22 72 65 71 75 65 73 74 4D 65
74 68 6F 64 22 3A 22 61 73 79 6E 63 22 2C 22
73 65 71 75 65 6E 63 65 49 64 22 3A 22 41 42
43 31 32 33 58 59 5A 22 7D
```

MQTT TCP/IP REST API

TCP/IP haberleşme metodunda, haberleşme ünitelerinin ve Head-End yazılımının gelen paketleri doğru bir şekilde almalarını sağlamak amacıyla **JSON** öncesi **#** karakteri ve sonrası **\$** karakteri olacaktır.

Başlangıç Karakteri	JSON Veri	Bitiş karakteri
#	{...}	\$

ASCII (Json) Hex

```
#{"device":{"flag":"XYZ","serialNumber":"ABCXYZ001"},"function":"identification","requestMethod":"async","referenceId":"ABC123XYZ"}$
```

ASCII (Json) Hex

```
23 7B 22 64 65 76 69 63 65 22 3A
7B 22 66 6C 61 67 22 3A 22 58 59 5A 22 2C 22
73 65 72 69 61 6C 4E 75 6D 62 65 72 22 3A 22
41 42 43 58 59 5A 30 30 31 22 7D 2C 22 66 75
6E 63 74 69 6F 6E 22 3A 22 69 64 65 6E 74 69
66 69 63 61 74 69 6F 6E 22 2C 22 72 65 71 75
65 73 74 4D 65 74 68 6F 64 22 3A 22 61 73 79
6E 63 22 2C 22 73 65 71 75 65 6E 63 65 49 64
22 3A 22 41 42 43 31 32 33 58 59 5A 22 7D 24
```

MQTT TCP/IP REST API

JSON Veri

{...}

ASCII (Json)

Hex

```
{"device":{"flag":"XYZ","serialNumber":"ABCXYZ001"},"function":"identification","requestMethod":"async","referenceId":"ABC123XYZ"}
```

ASCII (Json)

Hex

```
7B 22 64 65 76 69 63 65 22 3A 7B 22 66 6C 61  
67 22 3A 22 58 59 5A 22 2C 22 73 65 72 69 61  
6C 4E 75 6D 62 65 72 22 3A 22 41 42 43 58 59  
5A 30 30 31 22 7D 2C 22 66 75 6E 63 74 69 6F  
6E 22 3A 22 69 64 65 6E 74 69 66 69 63 61 74  
69 6F 6E 22 2C 22 72 65 71 75 65 73 74 4D 65  
74 68 6F 64 22 3A 22 61 73 79 6E 63 22 2C 22  
73 65 71 75 65 6E 63 65 49 64 22 3A 22 41 42  
43 31 32 33 58 59 5A 22 7D
```

⚠ ÖNEMLİ

JSON mesajının içeriğinin sıralaması önemli değildir. **JSON** içerisindeki veriler rastgele sıralamada gelebilirler.

1.2 Veri Tipleri

Veri Tipi	Açıklama	Örnek
bool	Doğru/yanlış mantıksal ifadesi, true ya da false olabilir	true
string	Uzunluk kısıtlaması olmayan metin	"ABC..."
string(n)	Maksimum n karakter uzunluğunda metin	n:3 için "ABC"
byte	0 ila 255 aralığında tam sayı	123
int	-2^{31} (-2,147,483,648) ila $2^{31}-1$ (2,147,483,647) aralığında tam sayı	123
decimal	$-10^{38} + 1$ ila $10^{38} - 1$ aralığında rasyonel sayı	1.23
date	ISO 8601 standartında tarih formatı. (YYYY-MM-DD)	2021-07-30
datetime	ISO 8601 standartında tarih ve zaman formatı. (YYYY-MM-DD HH:mm:ss)	2021-07-30 16:41:58
time	ISO 8601 standartında zaman formatı. (HH:mm:ss)	16:41:58
time2	ISO 8601 standartında saniye hariç zaman formatı. (HH:mm)	16:41
array[x]	x veri tipinde öğeleri içeren dizi	x:string için ["a","b","c"]
array[x](n)	x veri tipinde, maksimum n adet öğe içeren dizi	x:int ve n:4 için [2,5,8,11]

2 Tanımlar ve Kısaltmalar

MASS Haberleşme Protokolü belgelerinde kullanılan tanımlar ve kısaltmaların açıklamaları aşağıdaki tabloda yer almaktadır.

Tanım veya Kısaltma	İngilizce	Açıklama
Head-End Yazılımı	Head-End	Head-End Haberleşme Yazılımı
Haberleşme Ünitesi	communication unit	Merkezin sayaç ve uçbirimlerden veri alabilmesini sağlayan haberleşme ünitesi, modüler modem gibi cihazlardır.
Protokol	protocol	Merkez ile haberleşme üniteleri arasındaki veri alışverişini düzenleyen kurallar bütünü.
Ölçüm Cihazı	meter	Elektrik, su, doğalgaz vb. kaynakların kullanım miktarını ölçen cihazlar.
JSON	JSON	Bilgi alışverişinde yaygın kullanılan bir veri yapısı.
Pull	pull	Head-End yazılımı tarafından haberleşme ünitesine yapılan talep ile mesaj gönderimi.
Push	push	Haberleşme ünitesinin herhangi bir talep olmadan Head-End yazılımına mesaj gönderimi.
OBIS	Object Identification System	IEC-62056-6-1 standartlarında ölçüm cihazlarının verileri tutmak için kullandığı tanımlama sistemi
İletişim Tipi	communication type	Yapılan veri alışverişinin -iletişimin- zaman yönünden tipidir. Senkron (eşzamanlı) veya asenkron (eşzamanlı olmayan) olabilir.

Tanım veya Kısaltma	İngilizce	Açıklama
ICCID	Integrated Circuit Card Identification	Entegre Devre Kartı Tanımlayıcısı. Bir SIM kart, benzersiz seri numarasını içerir.
IMSI	International Mobile Subscriber Identity	Uluslararası Mobil Abone Kimliği. Bu numara, SIM kartın çalıştığı şebeke olan bireysel operatör şebekesini tanımlar. Şebeke sağlayıcısı ile SIM kartın iletişim kurabilmesini sağlar.
DLMS	Device Language Message Specification	DLMS User Association tarafından oluşturulan ve geliştirilen akıllı ölçüm cihazlarının haberleşmesinde kullanılan standartlar bütünüdür.
MODBUS	MODBUS	Modicon tarafından elektronik cihazlarda kullanılmak için geliştirilen bir seri haberleşme protokolüdür.
MBUS	Metering Bus	M-Bus Metering Bus kelimelerinden üretilen bir kısaltmadır. Ölçüm cihazlarının haberleşmesi için geliştirilmiş bir standarttır.
WMBUS	Wireless M-Bus	Wireless M-Bus, ölçüm cihazı okuma işlemi için kablosuz haberleşme yapısı ve M-BUS haberleşme protokolünün birleştirilmesi ile oluşmuş haberleşme sistemidir.
MQTT	Minimum Message Queuing Telemetry Transport	MQTT, makineden makineye iletişim (M2M) için kullanılan, standart tabanlı bir mesajlaşma protokolü veya kurallar dizisidir.

3 ASCII/Hex/Sayısal Dönüşüm Tablosu

ASCII	HEX	SAYISAL		ASCII	HEX	SAYISAL
NUL	0	0		US	1F	31
SOH	1	1		Space	20	32
STX	2	2		!	21	33
ETX	3	3		"	22	34
EOT	4	4		#	23	35
ENQ	5	5		\$	24	36
ACK	6	6		%	25	37
BEL	7	7		&	26	38
BS	8	8		'	27	39
HT	9	9		(28	40
LF	0A	10)	29	41
VT	0B	11		*	2A	42
FF	0C	12		+	2B	43
CR	0D	13		,	2C	44
SO	0E	14		-	2D	45
SI	0F	15		.	2E	46
DLE	10	16		/	2F	47
DC1	11	17		0	30	48
DC2	12	18		1	31	49
DC3	13	19		2	32	50
DC4	14	20		3	33	51
NAK	15	21		4	34	52
SYN	16	22		5	35	53
ETB	17	23		6	36	54
CAN	18	24		7	37	55
EM	19	25		8	38	56
SUB	1A	26		9	39	57
ESC	1B	27		:	3A	58
FS	1C	28		;	3B	59
GS	1D	29		<	3C	60
RS	1E	30				

TEDAŞ – MLZ/2024-081

ASCII	HEX	SAYISAL		ASCII	HEX	SAYISAL
=	3D	61		[5B	91
>	3E	62		\	5C	92
?	3F	63]	5D	93
@	40	64		^	5E	94
A	41	65		_	5F	95
B	42	66		`	60	96
C	43	67		a	61	97
D	44	68		b	62	98
E	45	69		c	63	99
F	46	70		d	64	100
G	47	71		e	65	101
H	48	72		f	66	102
I	49	73		g	67	103
J	4A	74		h	68	104
K	4B	75		i	69	105
L	4C	76		j	6A	106
M	4D	77		k	6B	107
N	4E	78		l	6C	108
O	4F	79		m	6D	109
P	50	80		n	6E	110
Q	51	81		o	6F	111
R	52	82		p	70	112
S	53	83		q	71	113
T	54	84		r	72	114
U	55	85		s	73	115
V	56	86		t	74	116
W	57	87		u	75	117
X	58	88		v	76	118
Y	59	89		w	77	119
Z	5A	90		x	78	120
				y	79	121
				z	7A	122
				{	7B	123
					7C	124
				}	7D	125
				~	7E	126
				DEL	7F	127

4 MQTT

4.1 MQTT Tanımı

MQTT, makineden makineye iletişim (M2M) için kullanılan, standart tabanlı bir mesajlaşma protokolü veya kurallar dizisidir. Nesnelerin interneti (IoT) cihazları, veri alışverişinde bulunmak için genelde sınırlı bant genişliğine sahip, kısık kaynaklı ağlar kullanmak zorundadır. Protokolün uygulanması basit olduğundan ve IoT verisini kolayca iletebildiğinden, IoT cihazları bu tarz veri iletimi için MQTT'yi kullanır. MQTT, cihazların birbirleri aracılığıyla bulutla ve bulutun cihazla mesajlaşmasını destekler. Bu protokolde istek(request)-yanıt(response) yapısına dayalı HTTP'ye karşıt olarak yayın(publish)-abone(subscriber) yapısında TCP/IP bağlantısı kurulur.

4.2 Kullanılacak Versiyon

MASS Protokolü kapsamında kullanılacak MQTT versiyonu 5.0'dir.

4.3 Hizmet Kalitesi Tasarımı (QoS)

MQTT haberleşme yönteminde mesajların teslim edilmesini güvenceye almak için çeşitli hizmet kalitesi seviyeleri bulunur. MASS protokolünde QoS 0 kullanılacaktır. Ancak ACK paketlerinin gönderimi ile verinin ulaşmış, ulaşmadığı kontrol edilecektir.

QoS 1 ve QoS 2 yöntemlerinde verinin en az 1 kere ya da kesinlikle 1 kere ulaşması garanti edilse de QoS 0 yöntemine göre yavaştır ve daha fazla bant genişliğine ihtiyaç duyar.

4.4 Topicler

Her haberleşme ünitesinin standart olarak abone (subscribe) olacağı ve Head-End yazılımının standart olarak yayın (publish) yapacağı topicler:

Topic	Örnek	Açıklama
<code>{/prefix}/request</code>	<code>/request</code>	Genel amaçlı olarak tüm haberleşme ünitelerine gönderilecek mesajlar için kullanılır.

Topic	Örnek	Açıklama
{/prefix}/request/flagKodu+seriNumarası	/request/ABC1234567890	Flag kodu belirtilen haberleşme ünitelerine mesajlar bu kanal üzerinden iletilecektir.

BİLGİ

MQTT topic'i dinlenirken # ve + işaretleri haberleşme ünitesinde **kullanılmayacaktır**. Örneğin /prefix/# topic'i dinlenmesi durumunda, diğer tüm haberleşme ünitelerine gidecek mesajlarda yakalanacağından, istenmeyen bir yoğunluk oluşturacaktır.

Her haberleşme ünitesinin standart olarak **yayın (publish)** yapacağı topicler:

Topic	Örnek	Açıklama
{/prefix}/response	/response	Genel amaçlı olarak tüm cevaplar için kullanılır.
{/prefix}/alarm	/alarm	Oluşan alarm durumları için kullanılır.

5 TCP/IP

MASS Protokol veri alışveriş yöntemlerinden TCP/IP yönteminde, soket bağlantısı üzerinden haberleşme üniteleri ile iletişim sağlanacaktır. Haberleşmenin senkron ve asenkron olması durumu header'da requestMethod ile paylaşılacaktır. Paylaşılmadığı durumda varsayılan olarak senkron haberleşme yapılacaktır.

5.1 Senkron Haberleşme

Her haberleşme ünitesi en az 1 adet TCP portunu dinleyecek ve bu port üzerinden gelen talepleri aldıktan sonra işleyerek Head-End yazılımına mevcut port üzerinden iletacaktır. Bu bağlantı haberleşme sona erdiğinde Head-End yazılımı tarafından kapatılacaktır.

5.2 Asenkron Haberleşme

Her haberleşme ünitesi en az 1 adet TCP portunu dinleyecek ve bu port üzerinden gelen talepleri aldıktan sonra işleyerek Head-End yazılımı tarafından dinlenen bir sunucu portuna iletacaktır. Bu haberleşme yönteminde mesaj gönderen taraf, mesajı gönderdikten sonra bağlantıyı kapatacaktır.

6 REST API

Uygulama programlama arayüzü (API: Application Programming Interface), bir yazılımın başka bir yazılımda tanımlanmış işlevlerini kullanabilmesi için oluşturulmuş bir tanım bütünüdür.

MASS Protokol veri alışveriş yöntemlerinden REST API yönteminde, Head-End yazılımı ve haberleşme ünitesi birer API arayüzüne sahiptir. Bu API arayüzü sayesinde haberleşme sağlanacaktır.

API arayüzü ile haberleşme esnasında fonksiyonlar birer metot olarak kullanılacaktır.

Rest API haberleşme yönteminde ACK paketleri, response içerisinde direk cevaplanacaktır. HTTP response kodu 200 olan tüm mesajlarda veri olarak ACK gönderilecektir (200:Başarılı operasyon, 405: Geçersiz veri).

7 Header

Haberleşme ünitesinin, Head-End yazılımı ile iletişimi sırasında gönderilen ve alınan tüm paketlerde bir başlık (header) yapısı bulunmaktadır.

7.1 Başlık Formatı

Tüm mesajlarda bulunacak başlık yapısıdır.

MQTT TCP/IP REST API

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "fonksiyonAdi",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": true|false,
  "packageNo": 3,
  "encryption": "AES-128",
  "compression": "base64"
}
```

MQTT TCP/IP REST API

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "fonksiyonAdi",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "requestMethod": "sync|async",
  "encryption": "AES-128",
  "compression": "base64"
}
```

MQTT

TCP/IP

REST API

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "fonksiyonAdi",
  "referenceld": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "encryption": "AES-128",
  "compression": "base64"
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
function	string	Zorunlu	Kullanılan protokol fonksiyonu ismi
referenceld	string	Zorunlu	Head-End yazılımı ile haberleşme ünitesi arasında gerçekleştirilen her bir iletişim için benzersiz olan ID bilgisi. GUID/UUID olarak üretilecektir. (UUID:universally unique identifier, GUID: globally unique identifier)
requestMethod	string	Opsiyonel	TCP/IP haberleşme yönteminde, haberleşmenin işleyiş biçimini belirtir. <code>async</code> ve <code>sync</code> değerlerini alabilir.
streaming	bool	Opsiyonel	Mevcut mesajın sonrasında gönderilmeyi bekleyen başka mesaj olup olmamasıyla alakalı işaret parametresi. Eğer <i>true</i> ise gönderilmeyi bekleyen başka mesaj mevcut, <i>false</i> ise başka mesaj yok. Eğer header içerisinde yok ise <i>false</i> anlamı taşımaktadır.
packageNo	byte	Opsiyonel	Parçalı mesaj gönderimlerinde mevcut paketin sıra numarası. Eğer header içinde yok ise mesajın tek pakette iletildiği anlamına gelmektedir.

encryption	string	Opsiyonel	Gönderilen mesajın <code>request</code> alanı için şifrelemek için kullanılan algoritma bilgisi. Eğer yok ise şifreleme yok anlamına gelmektedir.
compression	string	Opsiyonel	Mesajın <code>request</code> alanı için kullanılan veri sıkıştırma algoritması. Eğer header içinde yok ise veri sıkıştırması yapılmamıştır.

`device` nesnesi

⚠ ÖNEMLİ

`device` nesnesi; MQTT haberleşmede **opsiyoneldir**. Diğer tüm haberleşme yöntemlerinde **zorunludur**.

Parametre	Veri Tipi	Zorunluluk	Açıklama
flag	string(3)	Zorunlu	Haberleşme ünitesinin marka flag kodu.
serialNumber	string(15)	Zorunlu	Haberleşme ünitesinin 15 karakterli seri numarası

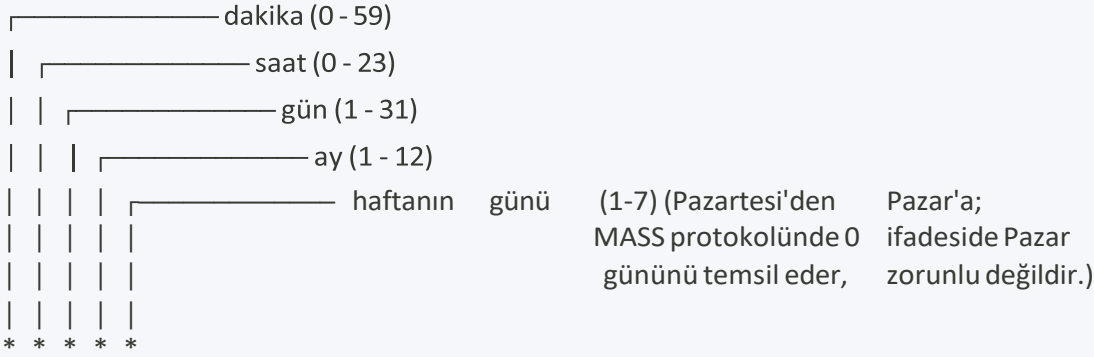
8 CRON

8.1 CRON Tanımı

Bir görevi ilerleyen zamanlarda da kullanmak için tasarlanan boşluklarla ayrılmış 5 ya da 6 dizeden oluşan bir yapıdır. MASS protokolünde 5 dizeli yapı kullanılmaktadır. CRON döngüsü 60 saniyedir.

CRON ifadeleri, MASS protokolünde tek başlarına değil, alarm, takvimli okuma gibi görevlerin içerisinde `period` alanında tanımlanırlar.

CRON Yapısı



Değer Aralıkları

Alan	İzin verilen değer aralığı	İzin verilen özel karakterler
Dakika	0-59	* , - /
Saat	0-23	* , - /
Gün	1-31	* , - / L

Ay	1-12	* , - /
Alan	İzin verilen değer aralığı	İzin verilen özel karakterler
Haftanın günü	0-7	* , - / L

Özel karakterlerin anlamı

Bilgi

Tabloda verilmeyen fakat başka sistemlerde kullanılan özel karakterler MASS protokolünde desteklenmeyecektir.

Karakter	Anlamı	Örnek
*	Bulunduğu alanın tüm değerlerini kapsar.	* * * * * (Her dakika çalışır.)
,	Bulunduğu alanda ve anlamına gelerek değerleri ifade eder.	0,15,30,45 * * * * * (Her saatin 15, 30, 45. dakikaları ile saat başında çalışır.)
-	Bulunduğu alanda verilen iki değer aralığında çalışır.	* * * * 1-5 (Sadece hafta içi çalışır.)
/	Bulunduğu alanda verilen değer, / karakterinden sonra verilen değer periyodunda çalıştırılmasını sağlar.	*/15 * * * * * (Her 15 dakikada bir çalışmayı sağlar. Saatin 15, 30, 45. dakikaları ile saat başında çalışır.)
L	Eğer gün alanında kullanılırsa ayın son günü anlamına gelir. Haftanın günü alanında kullanılırsa, ay içerisinde o güne denk gelen son gün çalışır.	* * * * 1L (Sadece son pazartesi günü çalışır.)

Örnekler

Örnek	Açıklama
* * * * *	CRON döngüsü 60 saniye olduğundan her dakika çalışır.
30 * * * *	Her saatin 30. dakikasında çalışır.
30 12 * * *	Her gün saat 12:30'da çalışır.
0 8-18/2 * * 1-5	Hafta içi saat 8 ile 18 arasında her iki saatte bir (08:00, 10:00, 12:00, 14:00, 16:00, 18:00) çalışır.
0,15,30,45 * * * *	Her saatin 15, 30, 45. dakikaları ile saat başında çalışır.
*/15 * * * *	Her 15 dakikada bir saatin 15, 30, 45. dakikaları ile saat başında çalışır.
* * 1 * *	Her ayın ilk günü çalışır.
* * L * *	Her ayın son günü çalışır.
0 0 1 6-8 *	Sadece yaz aylarında (Haziran, Temmuz, Ağustos) ayın ilk günü saat 00:00'da çalışır.

9 Fonksiyonlar

9.1 ACK

MASS protokolünde bir haberleşmenin her iki ucundaki taraf da karşıdan herhangi bir mesaj aldıklarında bu mesajı başarılı bir şekilde aldıklarına dair bir onaylama mesajı olarak **Acknowledgement** gönderirler. Buna **ACK** denir.

Mesajı gönderen taraf **ACK** alamadığı takdirde mesajın karşı tarafa başarıyla ulaşmadığı kabul edilecektir.

9.1.1 Çalışma zamanı

Head-End yazılımı sunucusu ya da haberleşme ünitesi herhangi bir mesaj aldıktan sonra **ACK** gönderir.

9.1.2 Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir. Çünkü hiçbir zaman haberleşmenin ilk mesajı değildir.

ÖNEMLİ

Alınan bir **ACK** mesajına karşılık **ACK** gönderilmez.

Pull

1. Karşı taraftan başlatılan bir haberleşmenin ardından gönderilir.
2. **ACK** paketini bekleyen taraf, **ACK** paketinin gelmemesi durumunda konfigürasyon paketinde belirlenen `retryInterval` süresi sonrası, `retryCount` adedince tekrar paketi göndermeyi dener.
3. Parçalı paketlerde ara paketlerde gönderilmez. `streaming:false` etiketinden sonra gönderilir.

9.1.3 Mesaj Formatları

Başarılı **ACK** paketi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "ack",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7"
}
```

Başarısız **ACK** paketi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "ack",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "response": {
    "failCode": 525,
    "failDescription": "Sorgulanan ve cevabı gelen ölçüm cihazı seri numarası uyumsuz."
  }
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
response.failCode	int	Zorunlu	Oluşan hatanın protokolde tanımlı kodu.
response.failDescription	string	Opsiyonel	Oluşan hatanın açıklaması.

9.2 Identification

Haberleşme ünitesinin ve üniteye ait tüm özelliklerin, herhangi bir kullanıcı girişi olmadan Head-End yazılımı tarafından otomatik tanınmasını sağlayacak fonksiyondur.

9.2.1 Çalışma zamanı

`Identification` paketi haberleşme ünitesine ilk kez enerji verildiğinde Head-End yazılımına otomatik olarak gönderilir.

Bu paket, Head-End yazılımı tarafından talep edildiğinde de gönderilir.

9.2.2 Çalışma şekli

Push

1. Haberleşme ünitesi kendisini tanımlayıcı bütün bilgileri Head-End yazılımına gönderir.
2. Head-End yazılımı verileri aldıktan sonra `ACK` gönderir.
3. Daha sonra, haberleşme ünitesi üzerinde bulunan "`registered`" alanını `true` değeri olacak şekilde ayarlama fonksiyonu ile değiştirerek bu paketin, enerji kesintileri sonrasında tekrar gönderilmesini engeller.
4. Haberleşme ünitesi, "`registered`": `true` işareti değişikliğini uyguladıktan sonra sunucuya `ACK` gönderir.

Pull

1. Head-End yazılımı, haberleşme ünitesinden `Identification` talep eder.
2. Haberleşme ünitesi Head-End yazılımına, o an haberleşme ünitesinin güncel durumunu içeren bilgileri gönderir.
3. Head-End yazılımı `ACK` gönderir.

🔥 ÖNEMLİ

Pull olarak **Identification** talep edildikten sonra **"registered"** işaretinde bir değişiklik olmaz.

9.2.3 Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

Identification talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "identification",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7"
}
```

Identification cevabı

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "identification",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "response": {
    "registered": true,
    "brand": "Marka1",
    "model": "Model1",
    "protocolVersion": "1.0.0",
    "manufactureDate": "2023-05-23",
    "firmware": "1.01",
    "signal": 13,
    "deviceDate": "2021-06-02 17:19:58",
    "daylightSaving": true,
    "timezone": "+03:00",
    "restartPeriod": 8,
    "networkId": "",
    "servers": [
```

```
{
  "ip": "123.45.68.10",
  "tcpPort": 1234,
  "udpPort": 4567,
  "primary": true
},
{
  "ip": "123.45.68.10",
  "tcpPort": 1234,
  "udpPort": 4567
}
],
"ntp": {
  "server": "",
  "port": 0
},
"ipWhiteList": ["123.45.68.10"],
"retryInterval": 10,
"retryCount": 3,
"maxPackageSize": 1024,
"mqtt": {
  "prefix": ""
},
"communicationInterfaces": [
  {
    "id": 1,
    "type": "gsm",
    "imei": "123456789012345",
    "phoneNumber": "5012345678",
    "ip": "123.45.68.9",
    "port": 3030,
    "apn": {
      "name": "",
      "user": "osos"
    },
    "simId": "",
    "imsi": ""
  },
  {
    "id": 2,
    "type": "wifi",
    "phoneNumber": "2121234567",
    "ip": "123.45.68.10",
    "port": 3030,
    "mac": "",
    "subnet": ""
  }
],
```

```

"serialPorts": [
  { "id": 1, "type": "rs485", "name": "rs485-1", "port": 7000 },
  { "id": 2, "type": "rs485", "name": "rs485-2", "port": 7001 },
  { "id": 3, "type": "rs232", "name": "rs232", "port": 7002 }
],
"ioInterfaces": [
  { "id": 1, "type": "relay", "name": "relay-1" },
  { "id": 2, "type": "relay", "name": "relay-2" },
  { "id": 3, "type": "dryContact", "name": "dry-1" },
  { "id": 4, "type": "digitalInput", "name": "panoKapagi" },
  { "id": 5, "type": "digitalInput", "name": "digitalInput-2" }
],
"modules": [
  {
    "serialNumber": "",
    "ip": "",
    "ipV6": "",
    "frequency": "",
    "band": "",
    "bitrate": "7E1",
    "meters": []
  }
],
"meters": [
  {
    "protocol": "IEC62056",
    "type": "electricity|water|gas|...",
    "brand": "MKL",
    "serialNumber": "12345678",
    "serialPort": "rs485-1",
    "initBaud": 300,
    "fixBaud": false,
    "frame": "7E1"
  },
  {
    "protocol": "DLMS",
    "type": "electricity|water|gas|...",
    "brand": "EMH",
    "serialNumber": "12345678",
    "serialPort": "rs485-2",
    "initBaud": 2400,
    "fixBaud": true,
    "frame": "8N1"
  }
]
}
}
}

```

Parametre	Veri Tipi	Zorunluluk	Açıklama
registered	bool	Zorunlu	Haberleşme cihazı daha önce kendisini headend sistemine tanıtmış ise true; tanıtamamış ise false
model	string	Zorunlu	Haberleşme cihazının model bilgisi
protocolVersion	string	Zorunlu	Haberleşme ünitesi üzerinde çalışan MASS protokolü versiyon bilgisi
manufactureDate	date	Zorunlu	Haberleşme ünitesinin üretim tarihi
firmware	string	Zorunlu	Haberleşme ünitesinin güncel yazılım versiyonu
signal	int	Zorunlu	Haberleşme ünitesinin güncel GSM sinyal seviyesi
deviceDate	datetime	Zorunlu	Haberleşme ünitesinin güncel zaman bilgisi
daylightSaving	bool	Zorunlu	Haberleşme ünitesinin güncel zaman bilgisi için gün ışığından yararlanma aktif ise true; pasif ise false
timezone	string	Zorunlu	Haberleşme ünitesinin güncel zaman bilgisi için bulunduğu zaman dilimi

Parametre	Veri Tipi	Zorunluluk	Açıklama
restartPeriod	int	Zorunlu	Haberleşme ünitesinin kendisini otomatik olarak restart etmesi için gerekli periyot. Süre saat cinsindedir. Eğer 0 ise bu özellik kapalıdır.
ntp	ntp	Zorunlu	Haberleşme ünitesinin zaman saatini güncellemek için kullanacağı sunucu bilgileri.
ipAddressWhiteList	array [string]	Zorunlu	Haberleşme ünitesine bağlanması için izin verilen ip adresleri. Eğer ip adresi dizisi boş ise tüm adreslerden erişim mümkündür.
retryInterval	int	Zorunlu	Haberleşme ünitesinin mesajı sunucuya göndermek için tekrar yapacağı deneme aralığı. Birimi dakikadır.
retryCount	int	Zorunlu	Haberleşme ünitesinin mesajı sunucuya göndermek için tekrar yapacağı deneme sayısı.
maxPackageSize	int	Opsiyonel	Haberleşme ünitesinin desteklediği maksimum paket boyutu
servers	array [server]	Zorunlu	Haberleşme ünitesinin push olarak bağlanacağı sunucu bilgileridir. Birden fazla sunucuya veri gönderebilir.

Parametre	Veri Tipi	Zorunluluk	Açıklama
communicationInterfaces	array [communicationInterface]	Zorunlu	Haberleşme ünitesi üzerinde bulunan haberleşme arayüzleri listesidir. Nesne tanımı aşağıda açıklanmıştır.
serialPorts	array [serialPort]	Zorunlu	Haberleşme ünitesi üzerinde bulunan seri portları listesidir. Nesne tanımı aşağıda açıklanmıştır.
ioInterfaces	array [ioInterface]	Zorunlu	Haberleşme ünitesi üzerinde bulunan giriş/çıkış arayüzleri listesidir. Nesne tanımı aşağıda açıklanmıştır.
meters	array [meter]	Zorunlu	Haberleşme ünitesine bağlı olan ölçüm cihazı listesidir. Nesne tanımı aşağıda açıklanmıştır.
schedules	array [schedule]	Zorunlu	Haberleşme ünitesi üzerinde tanımlı olan takvimlerin listesidir. Nesne tanımı Takvimsel Okuma Listesi bölümünde açıklanmıştır.

ntp nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
server	string	Zorunlu	Haberleşme ünitesinin zaman saatini güncellemek için kullanacağı sunucu adresi.
port	string	Zorunlu	Haberleşme ünitesinin zaman saatini güncellemek için kullanacağı sunucu port bilgisi.

servers nesnesi

Nesne MQTT ve REST/IP'de **kullanılmayacaktır**. Ancak TCP/IP'de kullanıldığında zorunluluklar aşağıda belirtilmiştir.

Parametre	Veri Tipi	Zorunluluk	Açıklama
ipAddress	string	Zorunlu	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı head-end sunucusunun ip adresi.
tcpPort	int	Zorunlu	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı head-end sunucusunun tcp port bilgisi.
udpPort	int	Zorunlu	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı headend sunucusunun udp port bilgisi.
primary	bool	Zorunlu	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı birincil sunucuyu ifade eder. Tek bir sunucu birincil olabilir.

mqtt nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
prefix	string	Opsiyonel	MQTT topic isimlendirmelerinde, topic adresleri için kullanılacak ön etiket.

communicationInterface nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	int	Zorunlu	Arayüzün id bilgisi. Benzersiz olacak şekilde verilmeli
type	string (enum)	Zorunlu	Arayüzün tipi: GSM, DSL
imei	string	Opsiyonel	(Mevcut ise) Arayüzün IMEI adresi
phoneNumber	string	Opsiyonel	(Mevcut ise) Arayüzün telefon numarası
ipAddress	string	Opsiyonel	(Mevcut ise) Arayüze erişim için IP adresi
apn	apn	Opsiyonel	(Mevcut ise) Arayüzün APN bilgisi
simId	string	Opsiyonel	(Mevcut ise) Arayüzün kullandığı SIM kartın benzersiz seri numarası bilgisi
imsi	string	Opsiyonel	(Mevcut ise) Arayüzün kullandığı SIM kartın abone kimliği bilgisi (IMSI)

apn nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
name	string	Zorunlu	Arayüzün erişim adı bilgisi
user	string	Opsiyonel	Arayüzün erişim adı kullanıcı bilgisi

serialPort nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	int	Zorunlu	Seri portun id bilgisi. Benzersiz olacak şekilde verilmeli
Type	string	Zorunlu	Seri portun tipi: RS-485, RS-232
name	string	Zorunlu	Seri portun ismi. Bu isim Head-End yazılımı tarafından değiştirilebilir
port	string	Zorunlu	Seri porta, TCP (ve eğer destekleniyorsa UDP) üzerinden ulaşmak için kullanılacak haberleşme portu

ioInterface nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	int	Zorunlu	Giriş/çıkış arayüzünün id bilgisi. Benzersiz olacak şekilde verilmeli
type	string	Zorunlu	Giriş/çıkış arayüzünün tipi: relay , dryContact , digitalInput
name	string	Zorunlu	Giriş/çıkış arayüzünün ismi. Bu isim Head-End yazılım tarafından değiştirilebilir

meter nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
protocol	string	Zorunlu	Ölçüm cihazının okunması için kullanılacak birincil protokol. IEC 62056, DLMS, MODBUS, MBUS, WMBUS

type	string	Zorunlu	Ölçüm cihazının ölçüm tipi. <input type="text" value="electricity,"/> <input type="text" value="water,"/> <input type="text" value="gas,"/> <input type="text" value="other"/>
brand	string	Zorunlu	Ölçüm cihazının marka kodu (Meter Flag Code)
serialNumber	string	Zorunlu	Ölçüm cihazının seri numarası
serialPort	string	Zorunlu	Ölçüm cihazının, haberleşme ünitesi üzerinde bağlı olduğu seri portun ismi
initBaud	int	Zorunlu	Ölçüm cihazını okumak için, haberleşme ünitesinin başlangıçta ayarlaması gereken baud-rate değeri
fixBaud	bool	Zorunlu	Ölçüm cihazı eğer sabit bir baud-rate değerine sahip ise <input type="text" value="true,"/> değil ise <input type="text" value="false"/>
frame	string	Zorunlu	Ölçüm cihazını okumak için, haberleşme ünitesinin ayarlaması gereken frame bilgisi. 7N1, 8E1 gibi

Özel Durumlar

- MQTT ile haberleşilmesi durumunda **servers** nesnesi kullanılmayacaktır
- REST/API ile haberleşilmesi durumunda **servers** nesnesi kullanılmayacaktır..

9.3 Heartbeat

Haberleşme ünitesinin, ayarlanan belirli aralıklarla Head-End yazılımına aktif olarak çalıştığına dair bilgi göndermesini sağlayan fonksiyondur.

9.3.1 Çalışma zamanı

Haberleşme ünitesi, ayarlanan belirli aralıklarla gönderim gerçekleştirir.

9.3.2 Çalışma şekli

Push

1. Haberleşme ünitesi, Head-End yazılımına `heartbeat` paketi gönderir.
2. Head-End yazılımı haberleşme ünitesine bu bilgileri aldığına dair `ACK` gönderir.

ÖNEMLİ

Eğer haberleşme ünitesi ACK mesajı almaz ise, `heartbeat` haberleşme yazılımına ulaşmamış kabul edilir. Haberleşme ünitesi üzerinde kayıtlı `retryCount` kadar deneme yapılır.

Pull

Bu fonksiyon için pull olarak çalışma desteklenmemektedir.

9.3.3 Mesaj Formatları

Heartbeat paketi

```
{  
  "device": {  
    "flag": "XYZ",  
    "serialNumber": "0123456789ABCDE"  
  },  
  "function": "heartbeat",  
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",  
  "response": {  
    "signal": 13,  
    "deviceDate": "2021-06-02 17:19:58",  
    "cpuTemp": 17  
  }  
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
signal	int	Zorunlu	Haberleşme ünitesinin güncel GSM sinyal seviyesi
deviceDate	datetime	Opsiyonel	Haberleşme ünitesinin güncel zaman bilgisi
cpuTemp	int	Opsiyonel	Haberleşme ünitesinin sıcaklığı, °C

9.4 Log

Haberleşme ünitesi üzerinde bulunan kayıtlı logları Head-End yazılımı üzerinden talep etmek için kullanılacak fonksiyondur.

9.4.1 Çalışma zamanı

Head-End yazılımı tarafından talep gönderildiğinde çalışır.

9.4.2 Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı tarafından `log` talebi gönderilir.
2. Haberleşme ünitesi, logları Head-End yazılımına gönderir. Eğer gönderim sırasında hata alınırsa `alarm` olarak bildirilir.
3. Head-End yazılımı mesajı aldıktan sonra, haberleşme ünitesine `ACK` paketi gönderir.

9.4.3 Mesaj Formatları

`log` talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "log",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "startDate": "2021-07-21 10:00:00",
    "endDate": "2021-07-22 10:00:00"
  }
}
```

⚠ DİKKAT

Eğer talep içerisinde gönderilen bitiş tarihi, başlangıç tarihinden önce ise haberleşme ünitesi, Head-End yazılımına **ACK** içerisinde hata (524 Sorgu mantık hatası (başlangıç tarihi > bitiş tarihi)) gönderir.

Parametre	Veri Tipi	Zorunluluk	Açıklama
startDate	dateTime	Zorunlu	Log talep edilen başlangıç tarihi
endDate	dateTime	Zorunlu	Log talep edilen bitiş tarihi

log cevabı

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "log",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "response": [
    {
      "incidentCode": 4,
      "description": "Yönerge bulunamadı",
      "date": "2021-06-28 13:55:00",
      "meter": {
        "brand": "EMH",
        "serialNumber": "12345678"
      }
    },
    {
      "incidentCode": 101,
      "description": "Sim kart ip alamadı",
      "date": "2021-06-28 13:55:00"
    }
  ]
}
```

9.4.4 Log tipleri

Tüm alarm tipleri birer log olarak kayıt atılmalıdır. Alarm tipleri dışında;

| Kod | Log |

|----|-----|

| 101 | Sim kart IP alamadı. (SMS atılacak) |

| 102 | MQTT erişilemiyor |

| 103 | ACK alınamayan mesajların dökümü (Günlük) |

| 104 | Erişim durumu dökümü (Günlük) |

Burada tanımlanmayan tüm log tipleri için `incidentCode` alanında LOG yazarak, modem özel log bilgilerini iletebilir.

9.5 Alarm

Haberleşme ünitesinde oluşan alarm bilgilerinin sunucuya anında iletilmesini sağlayan fonksiyondur.

9.5.1 Çalışma Zamanı

Haberleşme ünitesinde tetikleyici bir alarm oluştuğunda otomatik olarak gönderilir.

9.5.2 Çalışma Şekli

Push

1. Haberleşme ünitesi oluşan alarmları Head-End yazılımına gönderir.
2. Head-End yazılımı ise haberleşme ünitesine bu bilgileri aldığına dair ACK gönderir.

ÖNEMLİ

Eğer haberleşme ünitesi **ACK** mesaj almaz ise, uyarı Head-End yazılımına ulaşmamış kabul edilecektir.

Pull

Bu fonksiyon için pull olarak çalışma desteklenmemektedir.

9.5.3 Mesaj Formatları

Alarm Paketi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "ABCXYZ001"
  },
  "function": "alarm",
  "messageStatus": "success|fail",
  "referanceNo": "ABC123XYZ",
  "streaming": true|false,
  "response": [
    {
      "incidentCode": 4,
      "description": "Yönerge bulunamadı",
      "date": "2021-06-28 13:55:00",
      "meter": {
        "brand": "EMH",
        "serialNumber": "12345678"
      }
    },
    {
      "incidentCode": 5,
      "description": "Enerji kesintisi",
      "date": "2021-06-28 13:55:00"
    }
  ]
}
```

9.5.4 Alarm Tipleri

Kod	Alarm	Açıklama	Gönderim Zamanı
1	Disk Dolu	Yeni adımları kaydetmek için yeterli alan olmadığına	Adım kaydetme esnasında
		Yeni yönerge eklemek için yeterli alan olmadığına	Yönerge ekleme esnasında
		Gönderilecek push verileri diski doldurduğunda	Dolduğu anda sunucudan ACK alması durumunda 1 kere

		Loglar diski doldurduğunda	Dolduğu anda sunucudan ACK alması durumunda 1 kere
2	Parametre bulunamadı	Yönerge içerisinde bulunan parametre bulunamadığında gönderilir	Yönerge tetiklenirken
3	Fonksiyon bulunamadı	Protokolde bulunan ancak modem bulunamadığı fonksiyon gelmesi durumu	Modemde fonksiyon bulunamadığında
4	Yönerge bulunamadı	Çalıştırılması istenen yönerge modem üzerinde tanımlı değilse	
5	Enerji kesintisi	5 saniye üzerinde enerji kesintisi olduğu durumda gönderilecek	
6	Sinyal seviyesi düşük	GSM sinyal seviyesi 31 üzerinden 5 ve altına düştüğü an saatte en fazla 1 kere gönderilir.	
7	Kapak açıldı	Modem kapağına müdahale edildiği an gönderilir.	
8	Yazılım dosyası için http/FTP bağlantısı hatası	Adrese erişilemedi	
		Yetkisiz işlem	
		Dosya bulunamadı	
		Dosya bozuk	
9	Yazılım güncelleme başarısız	Dosya başarılı açıldıktan sonra yazılım güncelleme başarısız olursa	
10	Sim kart değişti	Simkart değiştiğinde sunucudan ACK alması durumunda 1 kere	
11	Açılıştaki sayaç okunamadı	Açılıştaki sayaçları tararken, sayacı okuyamaması durumunda	

TEDAŞ – MLZ/2024-081

12	Röle konumu hatası	Zamanlı çalışması gereken röle aktif edilemedi	
		Zamanlı çalışması gereken röle pasif yapılamadı	
13	Yetkisiz erişim denemesi	Hatalı kullanıcı adı ve şifre ile webservis üzerinden erişim denemesi	
		Hatalı kullanıcı adı ve şifre ile MQTT üzerinden erişim denemesi	
		MQTT dışında porta farklı şekilde erişim denemesi	
		İzin verilmeyen IP adreslerinden erişim denemesi	
14	CPU aşırı ısındı	CPU aşırı ısındığında sunucudan ACK alması durumunda 1 kere	
15	Hatalı header	Hatalı modeme ait paket gelmesi durumunda	
16	Desteklenmeyen özellik	Encryption desteklenmiyor	
		Compression desteklenmiyor	
17	NTP üzerinden modem saati güncellenemedi	Günde 1 kere zaman saati güncellenemediği ilk denemede	
18	Streaming hatası	Paketler başarılı birleştirilemediğinde	
		Kayıp paket olduğunda	
19	Takvimli yönerge çalıştırılmadı	Takvimli yönerge çalıştırılmadığında	

9.6 Configuration

Haberleşme ünitesinin ayarlarını değiştirmek için kullanılacak fonksiyondur.

9.6.1 Çalışma zamanı

Head-End yazılımı tarafından talep gönderildiğinde çalışır.

9.6.2 Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı tarafından `configuration` fonksiyon mesajı gönderilir.
2. Haberleşme ünitesi, bu mesajı aldığına dair Head-End yazılımına `ACK` gönderir.
3. Haberleşme ünitesi fonksiyon ile gönderilen ayarları yapar.
4. Ayarların güncelleme işlemi sonrası;
 - a. İşlem başarılı ise Head-End yazılımına `notification` fonksiyonu ile bilgi gönderilir. Bu bilgi paketinde değişen her bir ayar için eski ve yeni ayar değerleri bulunur.
 - b. İşlem kısmen başarılı olur ise başarılı olan parametreler için a maddesinde belirtilen şekilde `notification` gönderilir, hata alan parametreler için ise c maddesinde belirtilen şekilde `alarm` gönderilir.
 - c. İşlem başarısız olur ise Head-End yazılımına `alarm` fonksiyonu ile bu durumu bildirir.

ÖNEMLİ

Paket içerisinde tek bir register güncellemesi yapılabilir. İstek gönderilirken tüm etiketlerin değerleri olmayabilir. Bu durumda etiketleri olan değerler güncellenirken, etiketi olmayanlarda ise bir güncelleme işlemi yapılmaz.

9.6.3 Mesaj Formatları

configuration talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "configuration",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "registered": true,
    "deviceDate": "2021-06-02 17:19:58",
    "daylightSaving": true,
    "timezone": "+03:00",
    "restartPeriod": 8,
    "servers": [
      {
        "ipAddress": "123.45.68.10",
        "tcpPort": 1234,
        "udpPort": 4567
      }
    ],
    "ntp": {
      "server": "",
      "port": 0
    },
    "ipAddressWhiteList": [
      {
        "operation": "add|remove ",
        "ipAddress": ""
      }
    ],
    "retryInterval": 10,
    "retryCount": 3,
    "maxPackageSize": 1024,
    "mqtt": {
      "prefix": "",
      "address": "",
      "userName": "",
      "pwd": ""
    },
    "communicationInterfaces": [
      {
        "id": 1,
        "apn": {
          "name": ""
        }
      }
    ]
  }
}
```

```

    "user": "osos",
    "pwd": ""
  }
},
"serialPorts": [
  { "id": 1, "name": "rs485-1", "port": 7000 },
  { "id": 2, "name": "rs485-2", "port": 7001 },
  { "id": 3, "name": "rs232", "port": 7002 }
],
"ioInterfaces": [
  { "id": 1, "name": "relay-1" },
  { "id": 2, "name": "relay-2" },
  { "id": 3, "name": "dry-1" },
  { "id": 4, "name": "panoKapagi" },
  { "id": 5, "name": "digitalInput-2" }
],
"meters": [
  {
    "operation": "add",
    "meter": {
      "protocol": "IEC62056",
      "type": "electricity|water|gas|...",
      "brand": "MKL",
      "serialNumber": "12345678",
      "serialPort": "rs485-1",
      "initBaud": 300,
      "fixBaud": false,
      "frame": "7E1"
    }
  },
  {
    "operation": "remove",
    "meter": {
      "brand": "MKL",
      "serialNumber": "12345678"
    }
  }
]
}
}

```

Parametre	Veri Tipi	Zorunluluk	Açıklama
registered	bool	Zorunlu	Haberleşme ünitesinden gelen <code>identification</code> fonksiyonuna <code>true</code> olarak cevap dönülmesi durumunda, haberleşme ünitesinden bir daha bu paketi push olarak alamayacaktır
deviceDate	datetime	Opsiyonel	Haberleşme ünitesinin zaman bilgisini manuel olarak güncellemek için kullanılır
daylightSaving	bool	Opsiyonel	Haberleşme ünitesinin güncel zaman bilgisi için gün ışığından yararlanma aktif edilecek ise true; pasif ise false
timezone	string	Opsiyonel	Haberleşme ünitesinin güncel zaman bilgisi için bulunduğu zaman dilimi ayarı
restartPeriod	int	Opsiyonel	Haberleşme ünitesinin kendisini otomatik olarak restart etmesi için gerekli periyot. Süre saat cinsindedir. Eğer <code>0</code> ise bu özellik kapalıdır.

Parametre	Veri Tipi	Zorunluluk	Açıklama
ntp	ntp	Opsiyonel	Haberleşme ünitesinin zaman saatini güncellemek için kullanacağı sunucu bilgileri.
ipAddressWhiteList	array[ipAddressWhiteList]	Opsiyonel	Haberleşme ünitesine bağlanması için izin verilen ip adreslerinin bilgileri.
communicationInterfaces	array[communicationInterface]	Opsiyonel	Haberleşme ünitesi üzerinde bulunan haberleşme arayüzlerinin bilgilerinin güncellenmesi için kullanılacak nesne
serialPorts	array[serialPort]	Opsiyonel	Haberleşme ünitesi üzerinde bulunan seri portların bilgilerinin güncellenmesi için kullanılacak nesne
ioInterfaces	array[ioInterface]	Opsiyonel	Haberleşme ünitesi üzerinde bulunan giriş/çıkış arayüzlerinden bilgisi güncellenecek olan giriş/çıkış arayüzünün isim bilgisi güncellenecek olan arayüzlerin bilgisi.
meters	array [meterSetting]	Zorunlu	Haberleşme ünitesine bağlı olan ölçüm cihazını güncellemek için kullanılır.

ntp nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
server	string	Zorunlu	Haberleşme ünitesinin zaman saatini güncellemek için kullanacağı sunucu adresi.
port	string	Zorunlu	Haberleşme ünitesinin zaman saatini güncellemek için kullanacağı sunucu port bilgisi.

ipAddressWhiteList nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
operation	string	Opsiyonel	Haberleşme ünitesine bağlanması için izin verilen ip adreslerini ekleme veya kaldırmak için yapılacak işlem tipidir. add yeni bir ip adresini listeye ekler. remove mevcut bir ip adresini listeden kaldırır.
ipAddress	string	Opsiyonel	Haberleşme ünitesine bağlanmasına izin verilen veya izni iptal edilen ip adresi bilgisidir.

server nesnesi

Nesne MQTT ve REST/IP'de kullanılmayacaktır. Ancak TCP/IP'de kullanıldığında zorunluluklar aşağıda belirtilmiştir.

Parametre	Veri Tipi	Zorunluluk	Açıklama
ipAddress	string	Opsiyonel	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı headend sunucusunun ip adresi.
tcpPort	int	Opsiyonel	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı headend sunucusunun tcp port bilgisi.
udpPort	int	Opsiyonel	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı headend sunucusunun udp port bilgisi.
primary	bool	Opsiyonel	Haberleşme ünitesinin push işlemleri gerçekleştirmek için bağlanacağı birincil sunucuyu ifade eder. Tek bir sunucu birincil olabilir.

communicationInterface nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	int	Zorunlu	Haberleşme ünitesi üzerinde bulunan haberleşme arayüzlerinden isim bilgisi güncellenecek olan arayüzün id bilgisi.
apn	apn	Zorunlu	Haberleşme ünitesi üzerinde bulunan haberleşme arayüzlerinden bilgisi güncellenecek olan arayüzün apn bilgisi güncellenecek olan arayüzün bilgisi.

apn nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
name	string	Zorunlu	Arayüzün erişim adı bilgisi
user	string	Opsiyonel	Arayüzün erişim kullanıcı bilgisi
pwd	string	Opsiyonel	Arayüzün erişim şifresi

serialPort nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	int	Zorunlu	Haberleşme ünitesi üzerinde bulunan seri portlardan bilgisi güncellenecek olan seri portun id bilgisi.
name	string	Zorunlu	Seri porta verilmek istenen isim
port	string	Zorunlu	Seri porta direk (şeffaf) bağlantı için kullanılacak port bilgisi

ioInterface nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	int	Zorunlu	Haberleşme ünitesi üzerinde bulunan giriş/çıkış arayüzlerinden bilgisi güncellenecek olan giriş/çıkış arayüzünün isim bilgisi güncellenecek olan arayüzün id bilgisi.
name	string	Zorunlu	Haberleşme ünitesi üzerinde bulunan giriş/çıkış arayüzlerinden bilgisi güncellenecek olan giriş/çıkış arayüzünün isim bilgisi güncellenecek olan arayüzün bilgisi.

meterSetting nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
operation	string	Zorunlu	Haberleşme ünitesine, ölçüm cihazı ekleme veya kaldırmak için yapılacak işlem tipidir. add yeni bir ölçüm cihazı listeye ekler. remove mevcut bir ölçüm cihazını listeden kaldırır.
meter	meter	Zorunlu	Eklenecek ya da çıkarılacak ölçüm cihazı bilgileri

meter nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
protocol	string	Zorunlu	Ölçüm cihazının okunması için kullanılacak birincil protokol. IEC 62056, DLMS, MODBUS, MBUS, WMBUS
type	string	Zorunlu	Ölçüm cihazının ölçüm tipi. <input type="text" value="electricity,"/> <input type="text" value="water,"/> <input type="text" value="gas,"/> <input type="text" value="other"/>
brand	string	Zorunlu	Ölçüm cihazının marka kodu (Meter Flag Code)
serialNumber	string	Zorunlu	Ölçüm cihazının seri numarası
serialPort	string	Zorunlu	Ölçüm cihazının, haberleşme ünitesi üzerinde bağlı olduğu seri portun ismi
initBaud	int	Zorunlu	Ölçüm cihazını okumak için, haberleşme ünitesinin başlangıçta ayarlaması gereken baud-rate değeri
fixBaud	bool	Zorunlu	Ölçüm cihazı eğer sabit bir baud-rate değerine sahip ise <input type="text" value="true,"/> değil ise <input type="text" value="false"/>
frame	string	Zorunlu	Ölçüm cihazını okumak için, haberleşme ünitesinin ayarlaması gereken frame bilgisi. 7N1, 8E1 gibi

9.6.4 Başarılı işlem bildirimi

`notification` fonksiyonu ile Head-End yazılımına `referenceId` olarak bildirilir.

```
{
//mesajın öncesi
"response": [
  {
    "type": "alarm|info|danger",
    "level": "critical",
    "incidentCode": 278,
    "description": "cover opened",
    "date": "2021-06-28 13:55:00",
    "meter": {
      "brand": "EMH",
      "serialNumber": "12345678"
    }
  },
  {
    "type": "alarm|info|danger",
    "level": "info",
    "incidentCode": 439,
    "description": "relay removed",
    "date": "2021-06-28 13:55:00"
  }
]
//mesajın sonrası
}
```

9.7 Reset

Haberleşme ünitesini, Head-End yazılımı üzerinden resetlemek için kullanılacak fonksiyondur.

9.7.1 Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde ve/veya haberleşme ünitesi üzerinde bulunan periyodik resetleme zamanı geldiğinde çalışacaktır.

9.7.2 Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

Reset işlemi yalnızca *senkron* olarak çalışmaktadır ve haberleşme ünitesinden ACK alındıktan sonra bağlantı kesilir.

1. Head-End yazılımı tarafından reset fonksiyon mesajı gönderilir.
2. Haberleşme ünitesi, bu mesajı aldığına dair Head-End yazılımına ACK gönderir.
3. Haberleşme ünitesi resetleme işlemini gerçekleştirir.
 - a. Haberleşme ünitesi, başarılı resetleme sonrası Head-End yazılımına notification fonksiyonu ile bildirim gönderir. Bu gönderimde, notification fonksiyonu kuralları geçerlidir.
 - b. Resetleme başarılı olamaz ise alarm paketi ile Head-End yazılımına durum iletilir.

Zamanlanmış Çalışma

Haberleşme ünitesi, Head-End yazılımına notification fonksiyonu ile bildirim gönderir. Bu gönderimde, notification fonksiyonu kuralları geçerlidir.

9.7.3 Mesaj Formatları

reset talebi

```
{  
  "device": {  
    "flag": "XYZ",  
    "serialNumber": "0123456789ABCDE"  
  },  
  "function": "reset",  
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",  
  "request": {  
    "module": {  
      "ipV6": ""  
    },  
    "factoryDefault": false  
  }  
}
```

9.8 Firmware Güncelleme

Haberleşme ünitesinin yazılımını güncellemek için kullanılacak fonksiyondur.

9.8.1 Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

9.8.2 Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

Firmware güncelleme işlemi yalnızca *senkron* olarak çalışmaktadır ve haberleşme ünitesinden ACK alındıktan sonra bağlantı kesilir.

1. Head-End yazılımı tarafından `firmwareUpdate` fonksiyon mesajı gönderilir.
2. Haberleşme ünitesi, bu mesajı aldığı için Head-End yazılımına ACK gönderir.
3. Haberleşme ünitesi fonksiyonda gönderilen bilgiler ile yazılım dosyalarını indirmeye başlar.
 - a. Eğer güncelleme dosyası başarılı olarak indirilemez ise Head-End yazılımına alarm fonksiyonu ile bu durumu bildirir.
 - b. Eğer güncelleme dosyası başarılı olarak indirilebilirse, yazılım güncelleme işlemi yapmaya başlar.
4. Yazılım güncellemesi yapılırken haberleşme ünitesi üzerinde bulunan diğer ayarların etkilenmemesi gerekir.
5. Yazılım güncelleme işlemi sonrası;
 - a. Güncelleme işlemi başarılı bir şekilde tamamlanmış ise haberleşme yazılımına `notification` fonksiyonu ile bilgi gönderilir.
 - b. Güncelleme işlemi başarısız olur ise haberleşme yazılımına `alarm` fonksiyonu ile durum bildirilir.

9.8.3 Mesaj Formatları

firmwareUpdate talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "firmwareUpdate",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "address": "ftp://[user[:password]@]host[:port]/url-path",
    "dnsIp": "123.45.67.89",
    "module": {
      "ipV6": ""
    }
  }
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
address	string	Zorunlu	RFC-1738 standardına uygun FTP adresi ya da HTTP adresi
dnsIp	string	Opsiyonel	HTTP üzerinden yazılım güncellemesi yapılacağı zaman kullanılacak DNS IP adresi bilgisi
module	module	Opsiyonel	Yazılım güncellemesi yapılacak modüle ait bilgiler

module nesnesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
ipV6	string	Zorunlu	Yazılım güncellemesi yapılacak modüle ait IP V6 bilgisi

9.9 Takvimsel İşlemler

9.9.1 Takvimsel Okuma Listesi

Haberleşme ünitesine tanımlanmış takvimsel okumaları listeleyen komuttur.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesinden `schedule - list` talep edecektir.
2. Haberleşme ünitesi isteği aldığı için [ACK](#) mesajını Head-End yazılımına gönderir.
3. Haberleşme ünitesi, üzerinde tanımlı tüm takvimsel okumaları listeleyip Head-End yazılımına gönderir.
4. Head-End yazılımı mesajı aldığı için [ACK](#) mesajını haberleşme ünitesine gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

list talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "schedule",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "operation": "list",
```

```
"filter": {  
  "id": "Readout12345678",  
  "function": "read"  
}  
}
```

list cevabı

```
{  
  "device": {  
    "flag": "XYZ",  
    "serialNumber": "0123456789ABCDE"  
  },  
  "function": "schedule",  
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",  
  "response": {  
    "schedules": [  
      {  
        "id": "Readout12345678",  
        "function": "execute",  
        "startDate": "2020-06-01 00:00:00",  
        "endDate": "2022-07-01 00:00:00",  
        "period": "0 0 0/6 * * ?",  
        "directive": "ReadoutDirective1"  
      }  
    ]  
  }  
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
filter	object	Zorunlu	Listeleme kriterleri.
id	string	Zorunlu	Listelenmesi istenen takvimsel okumanın id'si. Request'te id girilirse cevapta sadece bir takvim dönebilir
startDate	number	Zorunlu	İlk Takvimsel okumanın başlayacağı tarih
endDate	number	Zorunlu	Son Takvimsel okumanın başlayacağı tarih
period	string	Zorunlu	Takvimsel okumanın tekrarlama periyodu

9.9.2 Takvimsel Okuma Çıkarma

Haberleşme ünitesinden takvimsel okumaları silen komuttur.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

`remove` komutu Head-End yazılımının başlatabileceği bir komuttur. Bu komutu alan haberleşme ünitesi bir [ACK](#) döner.

1. Head-End yazılımı, haberleşme ünitesinden `remove` talep eder.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#)'yı Head-End yazılımına gönderir
3. Haberleşme yazılımı üzerinde başka komut yok ise Head-End yazılımı bağlantıyı kapatmakla yükümlüdür.
4. Haberleşme ünitesi, istenilen işlemi gerçekleştirdikten sonra sonucu bir `notification` olarak Head-End yazılımına gönderir.
5. Head-End yazılımı mesajı aldığına dair [ACK](#) mesajını haberleşme ünitesine gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

remove talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "schedule",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "operation": "remove",
    "filter": {
      "id": "Readout12345678",
      "function": "execute"
    }
  }
}
```

Parametre	Veri Tipi	Açıklama
filter	object	Silme kriterleri.
id	string	Silinmesi istenen takvimsel okumanın id'si. Request'te id girilirse cevapta sadece bir takvim silinir
startDate	number	İlk Takvimsel okumanın başlayacağı tarih
endDate	number	Son Takvimsel okumanın başlayacağı tarih
period	string	Takvimsel okumanın tekrarlama periyodu

9.9.3 Takvimsel Readout

Haberleşme ünitesine takvimsel `readout` okuması tanımlayacak komuttur.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine Takvimsel Readout komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#) mesajını Head-End yazılımına gönderecektir

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanacaktır.

schedule talebi

```

{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "schedule",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "operation": "add",
    "schedules": [
      {
        "id": "Readout12345678",
        "function": "execute",
        "startDate": "2020-06-01 00:00:00",
        "endDate": "2022-07-01 00:00:00",
        "period": "0 0 0/6 * * ?",
        "directive": "ReadoutDirective",
        "parameters": {
          "METERSERIALNUMBER": "12345678"
        }
      }
    ]
  }
}

```

Parametre	Veri Tipi	Zorunluluk	Açıklama
startDate	number	Zorunlu	İlk Takvimsel okumanın başlayacağı tarih
endDate	number	Zorunlu	Son Takvimsel okumanın başlayacağı tarih
period	string	Zorunlu	Takvimsel okumanın tekrarlama periyodu

9.9.4 Takvimsel Custom Profil

Haberleşme ünitesine takvimsel `profil` okuması tanımlayacak komuttur.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine takvimsel profil komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair `ACK` mesajını Head-End yazılımına gönderir.

Mesaj Formatları

Aşağıdaki örnekte görüleceği üzere `custom` profil okumalarında okunması istenen profil aralığının tarih formatları "yy-MM-dd HH:mm;yy-MM-dd HH:mm" şeklinde gönderilebilir. Burada başlangıç ve bitiş tarihleri ";" ile ayrılmıştır. Bazı ölçüm cihazları için "," de kullanılabilir. Haberleşme ünitesinin bu tür ayrımları yapabilmesi gereklidir.

ÖNEMLİ

Yönergeli takvimsel profil okumalarında bitiş tarihi her zaman okuma zamanıdır.

Başlangıç tarihini ise haberleşme ünitesi `traverseHours` değişkeninden faydalanarak bulur. Şöyle ki: `traverseHours` rakamsal bir değerdir. Haberleşme ünitesi okuma anının tarihinden `traverseHours` değeri kadar "saat" öncesini bulup "başlangıç" değeri olarak kullanılmalıdır.

schedule talebi

Örnek 1

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "schedule",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "operation": "add",
    "schedules": [
      {
        "id": "Profil12345678",
        "function": "execute",
        "startDate": "2021-07-01 00:00:00",
        "endDate": "2022-07-02 00:00:00",
        "period": "0 0 0/6 * * ?",
        "traversHours": "4",
        "dateFormat": "yy-MM-dd HH:mm",
        "directive": "ProfileDirective",
        "parameters": {
          "METERSERIALNUMBER": "12345678",
          "STARTDATE": "$TRAVERSDATE",
          "ENDDATE": "$CURRENTDATE"
        }
      }
    ]
  }
}
```

Örnek 2

```

{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "schedule",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "operation": "add",
    "schedules": [
      {
        "id": "Profil12345678",
        "function": "execute",
        "startDate": "2021-07-01 00:00:00",
        "endDate": "2022-07-02 00:00:00",
        "period": "0 0 0/6 * * ?",
        "traversHours": "4",
        "dateFormat": "yy-MM-dd HH:mm",
        "directive": "ProfileDirective",
        "parameters": {
          "METERSERIALNUMBER": "12345678",
          "STARTDATE": "$TRAVERSDATE",
          "ENDDATE": "$CURRENTDATE"
        }
      }
    ]
  }
}

```

Parametre	Veri Tipi	Zorunluluk	Açıklama
startDate	number	Zorunlu	İlk Takvimsel okumanın başlayacağı tarih
endDate	number	Zorunlu	Son Takvimsel okumanın başlayacağı tarih
period	string	Zorunlu	Takvimsel okumanın tekrarlama periyodu
traversHours	number	Opsiyonel	Profil okumasının kaç saat geriden itibaren alınacağı

9.10 Aydınlatma Açma Kapama

Haberleşme ünitesine bağlı ölçüm cihazının sene boyunca hangi zamanlarda enerjisinin kesilip açılacağını haberleşme ünitesine tanımlayan komuttur.

! ÖNEMLİ

Bu komut eğer açma kesme işlemini haberleşme ünitesi kontrol edecekse kullanılır.

Açma kesmeyi ölçüm cihazı kontrol edecekse `write` komutu kullanılır.

9.10.1 Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

9.10.2 Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir. Çünkü hiçbir zaman haberleşmenin ilk mesajı değildir.

Pull

1. Head-End yazılımı, haberleşme ünitesinden `relay` talep edecektir.
2. Haberleşme ünitesi isteği aldığına dair `ACK`'yi Head-End yazılımına gönderecektir.
3. Haberleşme ünitesi, komutta belirtilen kriterlere göre bir takvim oluşturur. Zamanı geldiğinde açma-kesme işlemi yapar.
4. Haberleşme ünitesi, açma-kesme işleminden sonra Head-End yazılımına bildirim gönderebilir.

9.10.3 Mesaj Formatları

Örnek bir `relay` paketi aşağıda verilmiştir.

⚠ ÖNEMLİ

`openTimes` ve `closeTimes` parametreleri aydınlatmayı yöneten rölenin saat kaçta açılıp kaçta kapanacağını belirten byte dizileridir. Aydınlatmaların açma kapanma saatleri değildir.

Bu byte dizilerindeki her byte ikilisi tek bir zamana işaret etmektedir. İlk byte saati ikinci byte dakikayı temsil eder.

- [6 31] = 06:31
- [17 30] = 17:30

`startDate` parametresi ise diğer iki byte dizisindeki ilk ikilinin hangi güne işaret ettiğini belirler. Bu alan zorunlu değildir. `startDate` parametresi olmayan `relay` komutlarında byte dizilerindeki ilk ikili 1 Ocak gününe işaret etmiş sayılır. Bu parametrenin amacı komut boyutu bağlantının izin verdiği miktarı aştığında karışıklık çıkmasını engellemektir. `startDate` sayesinde bir yıllık açma-kapama zamanını 12'ye bölerek aylık olarak gönderebilirsiniz.

⚠ ÖNEMLİ

Yıl kaç olursa olsun hesaplamalara 29 Şubat dahil edilir. Yani bu komut için bir yıldaki gün sayısı her zaman 366'dır.

Başarılı relay paketi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "ABCXYZ001"
  },
  "function": "relay",
  "referenceId": "ABC123XYZ",
  "streaming": false,
  "request": {
    "startDate": "0101",
    "closeTimes": [6 31 6 32 .....],
    "openTimes": [17 30 17 29 .....]
  }
}
```

Parametre	Veri Tipi	Açıklama
startDate	string	açma-kapanma zamanlarının ilk günü
closeTimes	byte array	Röle kapanma zamanı
openTimes	byte array	Röle açılma zamanı

10 Yönergeler

10.1 Genel Bilgiler

Tanım

MASS protokolünde tanımlanan yönergeler (**directive**) haberleşme ünitesi ile ölçüm cihazı arasındaki iletişimi yönetmek içindir. Haberleşme ünitesi, Head-End yazılımı tarafından tanımlanan bu yönergelerdeki adımları (**step**) sıra ile (**order**) uygulayarak ölçüm cihazı ile haberleşecektir. Her adımda yapılacak bir işlem ve bu işlemi gerçekleştirmek için ihtiyaç duyacağı bir parametre (**parameter**) vardır.

Operasyon çeşitleri

setBaud

Haberleşme ünitesi ile sayaç haberleşmesi seri (serial) bir haberleşme olduğu için haberleşme ünitesinin ve sayacın aynı haberleşme hızına sahip olmaları gerekmektedir. Sayacın haberleşme hızı üretim esnasında belirlendiğinden dolayı haberleşme ünitesi iletişim hızını ayarlamalıdır. **setBaud** basamağında yer alan parametrede (**parameter**) bu hızın değeri yer alır. IEC-62056 standardında belirlenen hızlar şunlardır:

300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200

Örnek

```
//mesajın öncesi
{
  "order": 1,
  "operation": "setBaud",
  "parameter": 9600
}
//mesajın sonrası
```

setFraming

Haberleşme ünitesi ile sayaç seri haberleşmesinde bir diğer önemli husus da **framing** dir. Haberleşme ünitesi bu basamakta kendi **framingini** parametredeki değere göre değiştirmelidir.

Aşağıda haberleşme ünitelerin desteklemesi gereken **framing** tipleri verilmiştir:

Data Bits	Parity	Stop Bits
7	E	1
7	E	2
7	O	1
7	O	2
7	N	1
7	N	2
8	E	1
8	E	2
8	O	1
8	O	2
8	N	1
8	N	2
9	E	1
9	E	2
9	O	1
9	O	2
9	N	1
9	N	2

setFraming operasyonunda parameter alanının veri tipi string olarak kullanılacaktır.

Örnek

```
//mesajın öncesi
{
  "order": 2,
  "operation": "setFraming",
  "parameter": "8N1"
}
//mesajın sonrası
```

wait

Haberleşme ünitesi ile sayaç haberleşmesinde sayaç bir komut aldıktan sonra buna vereceği cevabı hazırlaması için bir zamana ihtiyaç duymaktadır. Haberleşme ünitesi bu sürede okuma yapmadan bekler. Bu süre sayaç markasına ve hatta modeline göre değişir. Bu yüzden bu süreyi Head-End yazılımı belirler. Bu basamakta `parameter` olarak değer bekleme süresidir.

ÖNEMLİ

Bekleme süresinin değeri her zaman `milisaniye` birimindedir. 1 sn = 1000 ms

`wait` operasyonunda `parameter` alanının veri tipi `int` olarak kullanılacaktır.

Örnek

```
//mesajın öncesi
{
  "order": 3,
  "operation": "wait",
  "parameter": 200
}
//mesajın sonrası
```

sendData

Bu basamakta haberleşme ünitesinin sayaca göndereceği veri belirtilir. Bu veri `parameter` alanında dizi olarak yer alır. Bu dizi gönderilecek byte'ların sayısal karşılığıdır. Örnek bir tablo aşağıda paylaşılmıştır. Tüm tablo ASCII/Hex/Sayısal dönüşüm tablosundan görülebilir.

ASCII	Hex	Sayısal Değer (Desimal)
NUL	0x00	0
SOH	0x01	1
!	0x21	33
7	0x37	55

🔥 ÖNEMLİ

`parameter` alanında gönderilen dizi içerisinde `string` veri tipinde öge var ise, bu öge bir argümanı işaret etmektedir ve doğrudan sayaca gönderilmeyecektir.

`sendData` operasyonunda `parameter` alanının veri tipi `array` olarak kullanılacaktır.

Örnek 1 (Argüman olmadan gönderim)

Gidecek mesaj;

```
ASCII : /?![CR][LF]
HEX   : 0x2F 0x3F 0x21 0x0A
SAYISAL: 47, 63, 33, 13, 10
```

```
//mesajın öncesi
{
  "order":4,
  "operation": "sendData",
  "parameter": [47, 63, 33, 13, 10]
}
//mesajın sonrası
```

Örnek 2 (Argüman **string** veri tipinde)

sendData basamağında parametre (**parameter**) olarak gelen dizide sayısal olmayan değerler argüman anlamına gelir ve haberleşme ünitesi tarafından gönderilen değer, karakterlerin ASCII sayısal değerleri ile değiştirilir.

- Eğer doğrudan bir okuma komutu geldiyse bu argüman o komut içerisinde yer alır.
- Takvimsel bir okuma ise takvimsel okuma kaydının içerisinde yer alır.

```
//mesajın öncesi
"request":{
  "directive": "ReadoutDirective1",
  "parameters": {
    "METERSERIALNUMBER": "12345678"
  }
}
//mesajın sonrası
```

Gidecek mesaj;

```
ASCII : /?12345678![CR][LF]
HEX   : 0x2F 0x3F 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x21 0x0A
SAYISAL: 47, 63, 49, 50, 51, 52, 53, 54, 55, 56, 33, 13, 10
```

```
//mesajın öncesi
{
  "order": 4,
  "operation": "sendData",
  "parameter": [47, 63, "METERSERIALNUMBER", 33, 13, 10]
}
//mesajın sonrası
```

Örnek 3 (Argüman **array** veri tipinde)

`sendData` basamağında parametre (`parameter`) olarak gelen dizide sayısal olmayan değerler argüman anlamına gelir ve haberleşme ünitesi tarafından gönderilen değer argüman dizisi ile yer değiştirilir.

- Eğer doğrudan bir okuma komutu geldiyse bu argüman o komut içerisinde yer alır.
- Takvimsel bir okuma ise takvimsel okuma kaydının içerisinde yer alır.

```
//mesajın öncesi
"request":{
  "directive": "ReadRegisterDirective",
  "parameters": {
    "SLAVEADDRESS": [17],
    "REGISTER": [0, 107]
  }
}
//mesajın sonrası
```

Gidecek mesaj;

ASCII : Değerlerin ASCII gösterimi uygun olmadığından örnek eklenmemiştir.

HEX : 0x11 0x03 0x00 0x6B 0x00 0x03 0x76 0x87

SAYISAL: 17, 3, 0, 107, 0, 3, 118, 135

```
//mesajın öncesi
{
  "order": 4,
  "operation": "sendData",
  "parameter": ["SLAVEADDRESS", 3, "REGISTER", 0, 3, 118, 135]
}
//mesajın sonrası
```

readData

Haberleşme ünitesi bu basamağa geldiğinde sayaçtan okuma yapmalıdır. Okuduğu değeri ise `parameter` alanındaki isimle kaydedip tüm basamaklar tamamlandığında bu isimle Head-End yazılımına göndermelidir.

`readData` operasyonunda `parameter` alanının veri tipi `string` olarak kullanılacaktır. `dataType` alanı ise `ASCII` veya `HEX` olabilir.

Örnek 1 (`dataType:ASCII`)

Yönergede `readData` adımının kullanımı:

```
//mesajın öncesi
{
  "order": 5,
  "operation": "readData",
  "parameter": "rawData",
  "dataType": "ASCII"
}
//mesajın sonrası
```

Daha sonra haberleşme ünitesi okuma yapıp gelen veriyi `rawData` alanına atadıktan sonra merkeze `ASCII` formatında gönderilmelidir. `ASCII/Hex/Sayısal dönüşüm tablosunda` `HEX` değeri `0x20` (hariç) (`SAYISAL` değeri `32` (hariç)) küçük olan karakterleri merkeze gönderirken köşeli parantez içerisinde `HEX` formatında göndermelidir. Burada sadece `[CR]` `[LF]` (`HEX:0x0D` `0x0A`) (`SAYISAL: 13, 10`) istisna durumdadır. Bu karakterler, `\r\n` şeklinde gönderilecektir. " karakteri ise gönderim sırasında `\` şeklinde gönderilecektir.

```
//mesajın öncesi
"data":{
  "rawData": "[02]0.0.0(23660088)\r\n0.9.2(2021-06- 22)\r\n0.9.1(10:18:42)\r\n1.8.0(000004891.722)\r\n![03]"
}
//mesajın sonrası
```

Okunan veri `rawData` örnektir, gerçek bir sayaç okuması sonucunu göstermez.

Örnek 2 (data`Type:HEX`)

Yönergede readData adımının kullanımı.

```
//mesajın öncesi
{
  "order":5,
  "operation": "readData",
  "parameter": "register",
  "dataType": "HEX"
}
//mesajın sonrası
```

Daha sonra haberleşme ünitesi okuma yapıp gelen veriyi register alanına atadıktan sonra merkeze HEX formatında gönderir.

```
//mesajın öncesi
"data":{
  "register":"110306AE415652434049AD"
}
//mesajın sonrası
```

loop

Bu basamak tekrarlanacak işlemler için kullanılacaktır. loop basamağının steps alanı bir çok basamaktan oluşan bir JSON dizisi olacaktır. Bu alt basamaklar döngü adedince tekrarlanacaktır.

Bu döngünün adedini ise parameter alanındaki değer belirleyecektir. parameter alanında bir argüman ismi belirtilecek, okuma komutu bu argümanı sağlayan bir liste gönderecektir. Bu listenin elemanlarının değerine argüman isminin başına * karakteri eklenerek ulaşılabılır.

Haberleşme ünitesi loop altındaki alt stepsleri bu listenin elemanları adedince tekrarlayacaktır.

loop operasyonunda parameter alanının veri tipi array olarak kullanılacaktır.

Örnek 1 (Argüman **array[string]** veri tipinde)

Liste argümanın okuma ya da takvim tanımlama sırasında kullanımı:

```
//mesajın öncesi
"request": {
  "directive": "ReadObisDirective1",
  "parameters": {
    "METERSERIALNUMBER": "12345678",
    "OBIS":["1.8.0","1.8.1","1.8.2","1.8.3","0.9.1","0.9.2","0.0.0"]
  }
}
//mesajın sonrası
```

Yönerge tanımlama sırasında argüman kullanımı:

```
{
  //mesajın öncesi
  "order": 10,
  "operation": "loop",
  "parameter": "OBIS",
  "steps": [
    {
      "order": 1,
      "operation": "sendData",
      "parameter": ["METERSERIALNUMBER"]
    },
    {
      "order": 2,
      "operation": "wait",
      "parameter": 100
    },
    {
      "order": 3,
      "operation": "sendData",
      "parameter": [2, 3, "*OBIS", 40, 41, 4, 5]
    },
    {
      "order": 4,
      "operation": "wait",
      "parameter": 100
    },
    {
      "order": 5,
      "operation": "readData",
      "parameter": "*OBIS"
    }
  ]
}
//mesajın sonrası
}
```

Okuma sonucu dönüşü sonrasında argüman kullanımı:

```
// mesajın öncesi
```

```
"data":{  
  "1.8.0": "00001.258*kwh",  
  "1.8.1": "00000.258*kwh",  
  "1.8.2": "00000.500*kwh",  
  "1.8.3": "00000.500*kwh",  
  "0.9.1": "18:12:56",  
  "0.9.2": "2022-09-28",  
  "0.0.0": "123456789"
```

```
}
```

```
// mesajın sonrası
```

TASAŞ

Örnek 2 (Argüman `array[array]` veri tipinde)

Liste argümanın okuma ya da takvim tanımlama sırasında kullanımı:

```
//mesajın öncesi
"request":{
  "directive": "ReadRegisterDirective1", "parameters": {
    "SLAVEADDRESS": [17],
    "REGISTER": [[0,107],[0,111],[0,121]]
  }
}
//mesajın sonrası
```

Yönerge tanımlama sırasında argüman kullanımı:

```
{
  //mesajın öncesi
  "order": 10,
  "operation": "loop",
  "parameter": "REGISTER",
  "steps": [
    {
      "order": 1,
      "operation": "sendData",
      "parameter": ["SLAVEADDRESS", 3, "*REGISTER", 0, 3, 118, 135]
    },
    {
      "order": 2,
      "operation": "readData",
      "parameter": "*REGISTER"
    }
  ]
}
//mesajın sonrası
```

ÖNEMLİ

Eğer `parameter` içerisinde bulunan argüman `array` ise merkezi yazılıma veri gönderimi sırasında dizi elemanının değerinin hex karşılığı, field isimlendirmesinde kullanılacaktır.

Okuma sonucu dönüşü sonrasında argüman kullanımı:

```
//mesajın öncesi
"data":{
  "006B":"00001.258*kwh",
  "006F":"00000.258*kwh",
  "0079":"00000.500*kwh"
}
//mesajın sonrası
```

calculate

Seri haberleşme sırasında, bazı kontrol karakterlerinin hesaplanması haberleşme ünitesi tarafından yapılacaktır. MASS protokolü kapsamında haberleşme üniteleri `bcc`, `crc-16/modbus`, `hcs`, `fcs` kontrol hesaplamalarını destekleyeceklerdir.

`calculate` operasyonunda `parameter` alanının `veri` tipi `calculate` nesnesi olarak kullanılacaktır. `calculate` nesnesi 3 alan içermektedir.

Parametre	Veri Tipi	Zorunluluk	Açıklama
type	string	Zorunlu	Kontrol hesaplama tipi. <code>bcc</code> , <code>crc-16/modbus</code> , <code>hcs</code> , <code>fcs</code> değerleri olabilir.
argument	string/array	Zorunlu	Kontrol hesabı yapılacak veri
out	string	Zorunlu	Kontrol hesabı yapıldıktan sonra kullanılacak değer için yer tutucu.

```
{
//mesajın öncesi
{
  "order": 1,
  "operation": "calculate",
  "parameter": {
    "type": "hcs",
    "argument": [160, 22, 3, 0, 2, 0, 35],
    "out": "hcs1"
  }
},
{
  "order": 2,
  "operation": "calculate",
  "parameter": {
    "type": "fcs",
    "argument": [160, 22, 3, 0, 2, 0, 35, 82, 162, 94, 230, 231, 0, 196, 1, 193,0, 18, 0, 2],
    "out": "fcs1"
  }
},
{
  "order": 3,
  "operation": "sendData",
  "parameter": [126, 160, 22, 3, 0, 2, 0, 35, "hcs1", 94, 230, 231, 0, 196, 1,193, 0, 18, 0, 2, "fcs1", 126]
}
//mesajın sonrası
}
```

10.2 Yönerge Listesi

Haberleşme ünitesine tanımlanmış yönergeleri listeleyen komuttur.

Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesinden `directive - list` talep eder.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#) mesajını Head-End yazılımına gönderir
3. Haberleşme ünitesi, üzerinde tanımlı tüm yönergeleri listeleyip haberleşme yazılımına gönderir.
4. Head-End yazılımı mesajı aldığına dair [ACK](#) mesajını haberleşme ünitesine gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

list talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "directive",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "operation": "list",
    "filter": {
      "id": "ReadoutDirective"
    }
  }
}
```

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "directive",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "response": {
    "directives": [
      {
        "id": "ReadoutDirective",
        "steps": [
          {
            "order": 1,
            "operation": "setBaud",
            "parameter": 300
          },
          {
            "order": 2,
            "operation": "setFraming",
            "parameter": "7E1"
          },
          {
            "order": 3,
            "operation": "sendData",
            "parameter": [47, 63, 33, 13, 10]
          },
          {
            "order": 4,
            "operation": "wait",
            "parameter": 10
          },
          {
            "order": 5,
            "operation": "readData",
            "parameter": "id"
          }
        ]
      }
    ]
  }
}
```

```

{
  "order" : 6,
  "operation": "sendData",
  "parameter": [6, 48, 53, 48, 13, 10]
},
{
  "order" : 7,
  "operation": "setBaud",
  "parameter": 9600
},
{
  "order" : 8,
  "operation": "wait",
  "parameter": 600
},
{
  "order" : 9,
  "operation": "readData",
  "parameter": "rawData"
}
]
}
]
}
}

```

Parametre	Veri Tipi	Açıklama
filter	object	Listeleme kriterleri.
id	string	Listelenmesi istenen yönergenin id'si. Request'te id girilirse cevapta sadece bir yönerge dönebilir

10.3 Yönerge Ekle

Haberleşme ünitesine bir yönerge tanımlayacak komuttur.

`directive` içerisindeki adımlar iki alan içerir:

- `Operation:` Bu adımda haberleşme ünitesinin yapacağı işin adıdır.
- `Parameter:` Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir.

Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `directive - add` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#) mesajını Head-End yazılımına gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

directive - **add** talebi

```

{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "directive",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "operation": "add",
    "directives": [
      {
        "id": "ReadoutDirective",
        "steps": [
          {
            "order": 1,
            "operation": "setBaud",
            "parameter": 300
          },
          {
            "order": 2,
            "operation": "setFraming",
            "parameter": "7E1"
          },
          {
            "order": 3,
            "operation": "sendData",
            // "parameter": "/?##METERSERIALNUMBER##![0D][0A]",
            "parameter": [47, 63, "METERSERIALNUMBER", 33, 13, 10]
          },
          {
            "order": 4,
            "operation": "wait",
            "parameter": 10
          },
          {
            "order": 5,
            "operation": "readData",
            "parameter": "id"
          },
          {
            "order": 6,
            "operation": "sendData",
            // "parameter": "[06]060[0D][0A]"
            "parameter": [6, 48, 54, 48, 13, 10]
          }
        ]
      }
    ]
  }
}

```

```
{
  "order" : 7,
  "operation": "setBaud",
  "parameter": 19200
},
{
  "order" : 8,
  "operation": "wait",
  "parameter": 600
},
{
  "order" : 9,
  "operation": "readData",
  "parameter": "rawData"
}
]
}
]
```

Parametre	Veri Tipi	Açıklama
id	string	Listelenmesi istenen yönergenin id'si. Request'te id girilirse cevapta sadece bir yönerge dönebilir
operation	string	Bu adımda haberleşme ünitesinin yapacağı işin adıdır
parameter	string	Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir
setBaud	string	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin baud değerinin belirlenmesi
setFraming	string	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin frame değerinin belirlenmesi
sendData	string	Ölçüm cihazına komut gönderilmesi işlemi, gönderilecek veri parameter alanında
readData	string	ölçüm cihazından veri okunması işlemi, parametre değeri ise okunan verinin atanacağı değişken
wait	string	Haberleşme ünitesinin haberleşmeye ara verip beklemede kalması, bekleme miktarı parameter alanında

10.4 Yönerge Sil

Haberleşme ünitesine tanımlanmış yönergelerden belirtilenleri silen komuttur.

Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `directive - add` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#) mesajını Head-End yazılımına gönderir.

Mesaj Formatı

Protokol genel formatı dahilinde haberleşme sağlanır.

list talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "directive",
  "referenceld": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "operation": "remove",
    "filter": {
      "id": "ReadoutDirective1"
    }
  }
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
filter	object	Zorunlu	Listeleme kriterleri.
id	string	Zorunlu	Listelenmesi istenen yönergenin id'si. Request'te id girilirse cevapta sadece bir yönerge dönebilir

TASLAK

10.5 Örnek Yönerge Ekle (OBIS Listesi)]

Haberleşme ünitesine bir yönerge tanımlayacak komuttur.

`directive` içerisindeki adımlar iki alan içerir:

- `operation`: Bu adımda haberleşme ünitesinin yapacağı işin adıdır.
- `parameter`: Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir.
 - Parametrik değer `operationIN` cinsine göre değişir.

Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `directive - add` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair Head-End yazılımına [ACK](#) mesajı gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

- `parameter` alanındaki parametrelerin nasıl kullanılacağı `okuma` sayfasında anlatılmıştı.
- `operation in loop` olduğu durumlarda `parameter` alanı alt yönergelerden oluşur. Bu yönergeler bu yönergeyi kullanan `okuma` komutu içerisinde `OBIS` parametresinin uzunluğu kadar tekrarlanır. bir `okuma` komutu

```
{ "OBIS" : ["1.8.0", "1.8.1", "1.8.2", "1.8.3"] } yani
```

```
{ "OBIS" : [[49, 46, 56, 46, 48], [49, 46, 56, 46, 49], [49, 46, 56, 46, 50], [49, 46, 56, 46, 51]] } şeklinde bir parametre içeriyorsa ve kullandığı directive içerisinde aşağıdaki gibi bir loop varsa bu döngü için OBIS listesinin elemanlarını parametre olarak kullanır.
```

directive-add talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "directive",
  "referenceld": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "operation": "add",
    "directives": [
      {
        "id": "ReadObisDirective",
        "steps": [
          {
            "order": 1,
            "operation": "setBaud",
            "parameter": 300
          },
          {
            "order": 2,
            "operation": "setFraming",
            "parameter": "7E1"
          }
        ]
      }
    ]
  }
}
```



```
"order": 3,  
"operation": "sendData",  
"parameter": [47, 63, "METERSERIALNUMBER", 33, 13, 10]  
},  
{  
"order": 4,  
"operation": "wait",  
"parameter": 100  
},  
{  
"order": 5,  
"operation": "readData",  
"parameter": "id"  
},  
{  
"order": 6,  
"operation": "sendData",  
"parameter": [6, 48, 54, 49, 13, 10]  
},  
{  
"order": 7,  
"operation": "setBaud",  
"parameter": 19200  
},  
{  
"order": 8,  
"operation": "wait",  
"parameter": 500  
},  
{  
"order": 10,  
"operation": "loop",  
"parameter": "OBIS",  
"steps": [  
  {  
    "order": 1,  
    "operation": "sendData",  
    "parameter": [2, 3, "*OBIS", 40, 41, 4, 5]  
  },  
  ],  
}
```

```

    "order": 2,
    "operation": "wait",
    "parameter": 100
  },
  {
    "order": 3,
    "operation": "readData",
    "parameter": "*OBIS"
  },
  {
    "order": 4,
    "operation": "wait",
    "parameter": 100
  }
]
}
]
}
}
}

```

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	string	Zorunlu	Listelenmesi istenen yönergenin id'si. Request'te id girilirse cevapta sadece bir yönerge dönebilir
operation	string	Zorunlu	Bu adımda haberleşme ünitesinin yapacağı işin adıdır
parameter	string	Zorunlu	Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir
setBaud	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin baud değerinin belirlenmesi

setFraming	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin frame değerinin belirlenmesi
sendData	string	Opsiyonel	Ölçüm cihazına komut gönderilmesi işlemi, gönderilecek veri <code>[parameter]</code> alanında
readData	string	Opsiyonel	Ölçüm cihazından veri okunması işlemi, parametre değeri ise okunan verinin atanacağı değişken
wait	string	Opsiyonel	Haberleşme ünitesinin haberleşmeye ara verip beklemede kalması, bekleme miktarı <code>[parameter]</code> alanında
loop	string	Opsiyonel	<code>[steps]</code> alanındaki yönergeler <code>[parameter]</code> içindeki liste adedince tekrarlanmalıdır

10.6 Örnek Yönerge Ekle (Readout)

Haberleşme ünitesine bir yönerge tanımlayacak komuttur.

Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `directive - add` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair Head-End yazılımına [ACK](#) mesajı gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

Aşağıdaki örnekte bir readout okuması yapmak için kullanılacak yönergeler dizisi verilmiştir. `sendData` yönergelerinde `parameter` alanındaki byte dizisi ölçüm cihazına gönderilecek byte dizisine işaret etmektedir. Bu byte dizisinin içerisinde karakter tabanlı veriler dikkatinizi çekecektir. Bunlar bu okuma sırasında kullanılacak parametreleri temsil ediyorlar. Örneğin:

- `[47, 63, "METERSERIALNUMBER", 33, 13, 10]` byte dizisi içerisindeki `METERSERIALNUMBER` okunmak istenen ölçüm cihazının seri numarasıdır. Haberleşme ünitesinin yapması gereken okuma esnasında byte dizisi içerisindeki bu alana `komut` içerisinde yer alan ölçüm cihazı seri numarasını yerleştirmektir. Örnek:
- Ölçüm cihazı seri numarası 12345678 ise gönderilecek byte dizisi şu şekilde olacaktır: `[47 63 49 50 51 52 53 54 55 56 33 13 10]`

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	string	Zorunlu	Listelenmesi istenen yönergenin id'si. Request'te id girilirse cevapta sadece bir yönerge dönebilir
operation	string	Zorunlu	Bu adımda haberleşme ünitesinin yapacağı işin adıdır
parameter	string	Zorunlu	Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir
setBaud	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin baud değerinin belirlenmesi
setFraming	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin frame değerinin belirlenmesi

Parametre	Veri Tipi	Zorunluluk	Açıklama
sendData	string	Opsiyonel	Ölçüm cihazına komut gönderilmesi işlemi, gönderilecek veri <code>parameter</code> alanında
readData	string	Opsiyonel	Ölçüm cihazından veri okunması işlemi, parametre değeri ise okunan verinin atanacağı değişken
wait	string	Opsiyonel	Haberleşme ünitesinin haberleşmeye ara verip beklemede kalması, bekleme miktarı <code>parameter</code> alanında

10.7 Örnek Yönerge Ekle (Profil)

Haberleşme ünitesine bir yönerge tanımlayacak komuttur.

Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `directive - add` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair Head-End yazılımına [ACK](#) mesajı gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

Aşağıdaki örnekte bir profil okuması yapmak için kullanılacak yönergeler dizisi verilmiştir. `sendData` yönergelerinde `parameter` alanındaki byte dizisi ölçüm cihazına gönderilecek byte dizisine işaret etmektedir. Bu byte dizisinin içerisinde karakter tabanlı veriler dikkatinizi çekecektir. Bunlar bu okuma sırasında kullanılacak parametreleri temsil ediyorlar. Örneğin:

- `[47, 63, "METERSERIALNUMBER", 33, 13, 10]` byte dizisi içerisindeki `METERSERIALNUMBER` okunmak istenen ölçüm cihazının seri numarasıdır. Haberleşme ünitesinin yapması gereken okuma esnasında byte dizisi içerisindeki bu alana `komut` içerisinde yer alan ölçüm cihazı seri numarasını yerleştirmektir. Örnek:
 - Ölçüm cihazı seri numarası 12345678 ise gönderilecek byte dizisi şu şekilde olacaktır: `[47 63 49 50 51 52 53 54 55 56 33 13 10]`
 - `STARTDATE` ve `ENDDATE` parametrelerinde ise özel bir durum vardır.

- Anlık okumalarda `STARTDATE` ve `ENDDATE` parametreleri Head-End yazılımından gelecek komut içerisinde yeralacaktır.
- Takvimsel okumalarda ise `STARTDATE` alanı güncel saatten `traversHours` parametresi kadar saat çıkarılarak elde edilecektir. `ENDDATE` için ise okuma esnasında tarih-saat kullanılacaktır.

directive-add talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "directive",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "operation": "add",
    "directives": [
      {
        "id": "ProfileDirective1",
        "steps": [
          {
            "order": 1,
            "operation": "setBaud",
            "parameter": 300
          },
          {
            "order": 2,
            "operation": "setFraming",
            "parameter": "7E1"
          },
          {
            "order": 3,
            "operation": "sendData",
            // "parameter": "/*##METERSERIALNUMBER##![OD][OA]",
            "parameter": [47, 63, "METERSERIALNUMBER", 33, 13, 10]
          },
          {
            "order": 4,
            "operation": "wait",
            "parameter": 10
          }
        ]
      }
    ]
  }
}
```

```

{
  "order" : 5,
  "operation": "readData",
  "parameter": "id"
},
{
  "order" : 6,
  "operation": "sendData",
  //"parameter": "[06]061[0D][0A]"
  "parameter": [6, 48, 54, 49, 13, 10]
},
{
  "order" : 7,
  "operation": "setBaud",
  "parameter": 9600
},
{
  "order" : 8,
  "operation": "wait",
  "parameter": 600
},
{
  "order" : 9,
  "operation": "readData",
  "parameter": "r2"
},
{
  "order" : 10,
  "operation": "sendData",
  //"parameter": "[01]R5[02]P1(21-06-22 00:00;21-06-22 12:22)()[03]?",
  "parameter": [1, 82, 53, 2, 80, 49, 40, "STARTDATE", 59, "ENDDATE", 41, 40, 41, 3, 63]
},
{
  "order" : 11,
  "operation": "readData",
  "parameter": "rawData"
}
]
}
]
}

```


TASLAK

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	string	Zorunlu	Listelenmesi istenen yönergenin id'si. Request'te id girilirse cevapta sadece bir yönerge dönebilir
operation	string	Zorunlu	Bu adımda haberleşme ünitesinin yapacağı işin adıdır
parameter	string	Zorunlu	Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir
setBaud	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin baud değerinin belirlenmesi
setFraming	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin frame değerinin belirlenmesi
sendData	string	Opsiyonel	Ölçüm cihazına komut gönderilmesi işlemi, gönderilecek veri parameter alanında
readData	string	Opsiyonel	Ölçüm cihazından veri okunması işlemi, parametre değeri ise okunan verinin atanacağı değişken
wait	string	Opsiyonel	Haberleşme ünitesinin haberleşmeye ara verip beklemede kalması, bekleme miktarı parameter alanında

10.8 Örnek Yönerge Ekle (Obis Yazma)

Haberleşme ünitesine bir yönerge tanımlayacak komuttur.

`directive` içerisindeki adımlar iki alan içerir:

- `operation`: Bu adımda haberleşme ünitesinin yapacağı işin adıdır.
- `parameter`: Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir.
 - Parametrik değer `operation`ın cinsine göre değişir.

Çalışma Zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma Şekli

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `directive - add` komutu gönderir.
2. Haberleşme ünitesi isteği aldığı için Head-End yazılımına [ACK](#) mesajı gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

`parameter` alanındaki parametrelerin nasıl kullanılacağı sayfasında anlatılmıştı.

directive-add talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "directive",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "request": {
    "operation": "add",
    "directives": [
      {
        "id": "WriteDirective1",
        "steps": [
          {
            "order": 1,
            "operation": "setBaud",
            "parameter": 300
          },
          {
            "order": 2,
            "operation": "setFraming",
            "parameter": "7E1"
          },
          {
            "order": 3,
            "operation": "sendData",
            "parameter": [47, 63, "METERSERIALNUMBER", 33, 13, 10]
          },
          {
            "order": 4,
            "operation": "wait",
            "parameter": 10
          },
          {
            "order": 5,
            "operation": "readData",
            "parameter": "id"
          },
          {
            "order": 6,
            "operation": "sendData",
```

```
"parameter": [6, 48, 54, 49, 13, 10]
},
{
  "order" : 7,
  "operation": "setBaud",
  "parameter": 19200
},
{
  "order" : 8,
  "operation": "wait",
  "parameter": 10
},
{
  "order" : 9,
  "operation": "sendData",
  "parameter": "PASSWORD"
},
{
  "order" : 10,
  "operation": "readData",
  "parameter": "passResponse"
},
{
  "order" : 11,
  "operation": "wait",
  "parameter": 10
},
{
  "order" : 12,
  "operation": "sendData",
  "parameter": "DATA"
},
{
  "order" : 13,
  "operation": "readData",
  "parameter": "obisResponse"
},
{
  "order" : 14,
  "operation": "wait",
  "parameter": 10
},
{
  "order" : 15,
```

```

"operation": "sendData",
// "parameter": "<SOH>B0<ETX><<XOR>"
"parameter": [1 66 48 3 113]
},
]
}
]
}
}
}
}
}
}
}
}
}

```

Parametre	Veri Tipi	Zorunluluk	Açıklama
id	string	Zorunlu	Listelenmesi istenen yönergenin id'si. Request'te id girilirse cevapta sadece bir yönerge dönebilir
operation	string	Zorunlu	Bu adımda haberleşme ünitesinin yapacağı işin adıdır
parameter	string	Zorunlu	Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir
setBaud	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin baud değerinin belirlenmesi
setFraming	string	Opsiyonel	Haberleşme ünitesi-ölçüm cihazı arası haberleşmenin frame değerinin belirlenmesi
sendData	string	Opsiyonel	Ölçüm cihazına komut gönderilmesi işlemi, gönderilecek veri parameter alanında
readData	string	Opsiyonel	Ölçüm cihazından veri okunması işlemi, parametre değeri ise okunan verinin atanacağı değişken
wait	string	Opsiyonel	Haberleşme ünitesinin haberleşmeye ara verip beklemede kalması, bekleme miktarı parameter alanında

10.9 Yönerge Çalıştırma (Obis Listesi)

Haberleşme ünitesinin altındaki ölçüm cihazlarından obis listesi okunmasını sağlayan fonksiyondur.

Gönderilen komut bir `execute` fonksiyonudur.

Yönerge; haberleşme ünitesinin yapması gereken bir işlemin aşamalarının Head-End yazılımı tarafından dikte edilmesi demektir. Obislerden oluşan bir listeyi okumak için haberleşme ünitesi birkaç işlemi ardarda gerçekleştirir.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Push

1. Haberleşme ünitesi `schedule` zamanı gelen ölçüm cihazından veriyi `schedule` içerisindeki belirtilen yönergeleri takip ederek okuduktan sonra Head-End yazılımına bağlanıp veriyi gönderir.
2. Head-End yazılımı okumayı aldığına dair [ACK](#) gönderir

Pull

1. Head-End yazılımı, haberleşme ünitesine `execute` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#)'yı Head-End yazılımına gönderir.
3. Haberleşme ünitesi ilgili ölçüm cihazından veriyi komut içerisindeki yönergeleri takip ederek okuduktan sonra Head-End yazılımına bağlanıp veriyi gönderir.
4. Head-End yazılımı okumayı aldığına dair [ACK](#) gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

Obis List talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "directive": "ReadPackageDirective",
    "parameters": {
      "METERSERIALNUMBER": "12345678",
      "OBIS": ["1.8.0", "1.8.1", "1.8.2", "1.8.3", "0.9.1", "0.9.2", "0.0.0"]
    }
  }
}
```

Obis List cevabi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "response": {
    "directive": "ReadPackageDirective",
    "executionDate": "2021-06-22 11:23:06",
    "data": {
      "id": "/LGZ5\2ZMG405000b.P07",
      "1.8.0": "00001.258*kwh",
      "1.8.1": "00000.258*kwh",
      "1.8.2": "00000.500*kwh",
      "1.8.3": "00000.500*kwh",
      "0.9.1": "18:12:56",
      "0.9.2": "2022-09-28",
      "0.0.0": "123456789"
    }
  }
}
```


Parametre	Veri Tipi	Zorunluluk	Açıklama
executionDate	datetime	Zorunlu	İşleme tarihi
data	string	Zorunlu	Okuma sonucu
id	string	Zorunlu	Ölçüm cihazı
OBIS	json array	Zorunlu	Okuma sonuçları

TASLAK

10.10 Yönerge Çalıştırma (Readout)

Haberleşme ünitesinin altındaki ölçüm cihazlarından `readout` verisi alınmasını sağlayan fonksiyondur.

Gönderilen komut bir `execute` fonksiyonudur.

Yönerge; haberleşme ünitesinin yapması gereken bir işlemin aşamalarının Head-End yazılımı tarafından dikte edilmesi demektir. Bir `readout` okuması yapmak için haberleşme ünitesi birkaç işlemi ardarda gerçekleştirir

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Push

1. Haberleşme ünitesi `schedule` zamanı gelen ölçüm cihazından veriyi `schedule` içerisindeki belirtilen yönergeleri takip ederek okuduktan sonra Head-End yazılımına bağlanıp veriyi gönderir.
2. Head-End yazılımı okumayı aldığına dair [ACK](#) gönderir

Pull

1. Head-End yazılımı, haberleşme ünitesine `execute` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#)'yı Head-End yazılımına gönderir.
3. Haberleşme ünitesi ilgili ölçüm cihazından veriyi komut içerisindeki yönergeleri takip ederek okuduktan sonra Head-End yazılımına bağlanıp veriyi gönderir.
4. Head-End yazılımı okumayı aldığına dair [ACK](#) gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

readout talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "directive": "ReadoutDirective1",
    "parameters": {
      "METERSERIALNUMBER": "12345678"
    }
  }
}
```

readout cevabı

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "response": {
    "directive": "ReadoutDirective1",
    "executionDate": "2021-06-22 11:23:06",
    "data": {
      "id": "/LGZ5\2ZMG405000b.P07",
      "rawData": "0.0.0(23660088)\r\n0.9.2(2021-06-22)\r\n0.9.1(10:18:42)\r\n1.8.0(0000004891.722)\r\n....."
    }
  }
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
rawData	string	Zorunlu	Ölçüm cihazından okunan verinin işlenmemiş ham hali
executionDate	datetime	Zorunlu	İşleme tarihi
data	string	Zorunlu	Okuma sonucu
id	string	Zorunlu	Ölçüm cihazı

10.11 Yönerge Çalıştırma (Yük Profili)

Haberleşme ünitesinin altındaki ölçüm cihazlarından `profile` verisini yönergeli bir şekilde alacak fonksiyondur.

Gönderilen komut bir `execute` fonksiyonudur.

Yönerge; haberleşme ünitesinin yapması gereken bir işlemin aşamalarının Head-End yazılımı tarafından dikte edilmesi demektir. Bir `profile` okuması yapmak için haberleşme ünitesi birkaç işlemi ardarda gerçekleştirir.

İşlem adımları `steps` dizisi içerisinde tanımlanır. `steps` içerisindeki adımlar iki alan içerir:

- `operation`: Bu adımda haberleşme ünitesinin yapacağı işin adıdır.
- `parameter`: Yukarıdaki işi yaparken haberleşme ünitesinin kullanacağı parametrik değerdir.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

Bu fonksiyon yalnızca *asen kron* olarak çalışır.

1. Head-End yazılımı, haberleşme ünitesine `execute` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair `ACK` yı Head-End yazılımına gönderir.
3. Head-End yazılımı üzerinde başka komut yok ise Head-End yazılımı bağlantıyı kapatmakla yükümlüdür.
4. Haberleşme ünitesi ilgili ölçüm cihazından veriyi komut içerisindeki yönergeleri takip ederek okuduktan sonra Head-End yazılımına bağlanıp veriyi gönderir.
5. Head-End yazılımı okumayı aldığına dair `ACK` gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

profile talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "directive": "ProfileDirective1",
    "parameters": {
      "METERSERIALNUMBER": "12345678",
      "startDate": "2021-06-22 00:00:00",
      "endDate": "2021-06-22 12:05:00"
    }
  }
}
```

profile cevabı

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "response": {
    "identification": "/LGZ5\2ZMG405000b.P07",
    "executionDate": "2021-06-22 11:23:06",
    "data": {
      "directive": "ProfileDirective1",
      "id": "/LGZ5\2ZMG405000b.P07",
      "r2": "R5()",
      "rawData": "(2021-07-13)(02:00)(000018.036*kWh)(000000.000*kWh)(000001.540*kVArh)(000005.933*kVArh)(000000.000*kVArh)(000000.000*kVArh)(232V)(001.0A)(0.83)(233V)(001.5A)(0.67)(232V)(001.5A)(0.56)\r\n(2021-07-13)(02:15)(000018.162*kWh)(000000.000*kWh)(000001.540*kVArh)(000006.126*kVArh)(000000.000*kVArh)(000000.000*kVArh)(231V)(001.0A)(0.84)(234V)(001.5A)(0.50)(232V)(001.5A)(0.56)\r\n(2021-07-
```

```

13)(02:30)(000018.275*kWh)(000000.000*kWh)(000001.540*kVArh)(000006.323*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(232V)(001.0A)(0.84)(235V)(001.7A)(0.24)(233V)(001.5A)(0.57)\r\n(2021-07-
13)(02:45)(000018.423*kWh)(000000.000*kWh)(000001.540*kVArh)(000006.516*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(233V)(001.0A)(0.83)(234V)(002.0A)(0.70)(234V)(001.5A)(0.57)\r\n(2021-07-
13)(03:00)(000018.577*kWh)(000000.000*kWh)(000001.540*kVArh)(000006.697*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(232V)(000.8A)(0.55)(235V)(002.0A)(0.48)(234V)(001.5A)(0.56)\r\n(2021-07-
13)(03:15)(000018.700*kWh)(000000.000*kWh)(000001.540*kVArh)(000006.895*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(233V)(000.8A)(0.55)(235V)(001.7A)(0.24)(234V)(001.5A)(0.56)\r\n(2021-07-
13)(03:30)(000018.816*kWh)(000000.000*kWh)(000001.540*kVArh)(000007.094*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(233V)(001.0A)(0.83)(235V)(001.7A)(0.24)(236V)(001.5A)(0.56)\r\n(2021-07-
13)(03:45)(000019.405*kWh)(000000.000*kWh)(000001.540*kVArh)(000007.270*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(232V)(010.2A)(0.99)(236V)(010.9A)(0.99)(234V)(011.1A)(0.99)\r\n(2021-07-
13)(04:00)(000020.681*kWh)(000000.000*kWh)(000001.540*kVArh)(000007.463*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(230V)(000.8A)(0.55)(234V)(002.1A)(0.49)(231V)(001.5A)(0.57)\r\n(2021-07-
13)(04:15)(000020.831*kWh)(000000.000*kWh)(000001.540*kVArh)(000007.651*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(229V)(001.0A)(0.83)(232V)(001.7A)(0.25)(232V)(001.5A)(0.57)\r\n(2021-07-
13)(04:30)(000020.943*kWh)(000000.000*kWh)(000001.540*kVArh)(000007.846*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(231V)(001.0A)(0.84)(232V)(001.6A)(0.25)(232V)(001.5A)(0.57)\r\n(2021-07-
13)(04:45)(000021.076*kWh)(000000.000*kWh)(000001.540*kVArh)(000008.022*kVArh)(000000.000*kVArh)(0000
00.000*kVArh)(231V)(001.0A)(0.84)(232V)(001.8A)(0.35)(232V)(001.5A)(0.58)\r\n"
}
}
}

```

Parametre	Veri Tipi	Açıklama
identification	string	Ölçüm cihazının kimlik bilgisi
data	string	Okuma sonucu
iD	string	Yönergelerde yer alan değişken ismi
r2	string	Yönergelerde yer alan değişken ismi
rawData	string	Yönergelerde yer alan değişken ismi

10.12 Yönerge Çalıştırma (Obis Yazma)

Haberleşme ünitesinin altındaki ölçüm cihazlarına veri kaydedilmesi için kullanılan fonksiyondur.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Ölçüm cihazlarında: Haberleşme ünitesi altındaki ölçüm cihazının bir `register`'ı değiştirilmek istendiğinde bu registerın `OBIS` anahtarı ile birlikte gönderilir.

Push

Bu fonksiyon için push olarak çalışma desteklenmemektedir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `execute` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair `ACK` mesajını Head-End yazılımına gönderir
3. Head-End yazılımı üzerinde başka komut yok ise Head-End yazılımı bağlantıyı kapatmakla yükümlüdür.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

execute talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "directive": "WriteDirective1",
    "parameters": {
      "METERSERIALNUMBER": "12345678",
      "PASSWORD": "12345678",
      "DATA": "0.9.2(21-07-28)"
    }
  }
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
directive	string	Zorunlu	Komut ismi
password	string	Zorunlu	Ölçüm cihazının şifresi
data	string	Zorunlu	Veri yazılmak istenen register adresi ve yazılacak değer

10.13 Yönerge Çalıştırma (Su Sayacı)

Haberleşme ünitesinin altındaki ölçüm cihazlarından su sayaçları verisi alınmasını sağlayan fonksiyondur.

Gönderilen komut bir `execute` fonksiyonudur.

Yönerge; haberleşme ünitesinin yapması gereken bir işlemin aşamalarının Head-End yazılımı tarafından dikte edilmesi demektir. Bir `wmbus_water` okuması yapmak için haberleşme ünitesi birkaç işlemi ardarda gerçekleştirir.

Çalışma zamanı

Head-End yazılımı tarafından komut gönderildiğinde çalışır.

Çalışma şekli

Push

1. Haberleşme ünitesi `schedule` zamanı gelen ölçüm cihazından veriyi `schedule` içerisindeki belirtilen yönergeleri takip ederek okuduktan sonra Head-End yazılımına bağlanıp veriyi gönderir.
2. Head-End yazılımı okumayı aldığına dair [ACK](#) gönderir.

Pull

1. Head-End yazılımı, haberleşme ünitesine `execute` komutu gönderir.
2. Haberleşme ünitesi isteği aldığına dair [ACK](#)'yı Head-End yazılımına gönderir.
3. Haberleşme ünitesi ilgili ölçüm cihazından veriyi komut içerisindeki yönergeleri takip ederek okuduktan sonra Head-End yazılımına bağlanıp veriyi gönderir.
4. Head-End yazılımı okumayı aldığına dair [ACK](#) gönderir.

Mesaj Formatları

Protokol genel formatı dahilinde haberleşme sağlanır.

wmbus_water talebi

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "request": {
    "directive": "wmbus_water",
    "parameters": {
      "METERSERIALNUMBER": "12345678"
    }
  }
}
```

wmbus_water cevabı

```
{
  "device": {
    "flag": "XYZ",
    "serialNumber": "0123456789ABCDE"
  },
  "function": "execute",
  "referenceId": "3d0b242e-1866-4a41-a8ca-1372f1b34ab7",
  "streaming": false,
  "response": {
    "directive": "wmbus_water",
    "executionDate": "2021-06-22 11:23:06",
    "data": {
      "rawData":
"550003D72A23442C0B26631801740784007AB600000004150200000004FD01E80393E7000009FD1F012F00090CE19ADA"
    }
  }
}
```

Parametre	Veri Tipi	Zorunluluk	Açıklama
rawData	string	Zorunlu	Ölçüm cihazından okunan verinin işlenmemiş hali
executionDate	datetime	Zorunlu	İşleme tarihi
data	string	Zorunlu	İşlem sonucu

id	string	Zorunlu	Ölçüm cihazı
rawData	string	Zorunlu	Ham veri

TASLAK

11 Örnek Ölçüm Cihazı Mesajları

11.1 Uzun Okuma Paketi

```

- (15:23:07.326) =====
- (15:23:07.342) Serial port opened
- (15:23:07.357) =====
- (15:23:07.373) Baud rate is changed to 300
- (15:23:07.388) -> /?! [CR][LF]
- (15:23:07.607) /BYL6<2>BGZ(BT10.LP-R1) [CR][LF]
- (15:23:08.529) -> [ACK] 060 [CR][LF]
- (15:23:08.795) Baud rate is changed to 19200
- (15:23:08.810) Serial port opened
- (15:23:08.841) =====
- (15:23:08.873) Serial port opened
- (15:23:08.920) =====
- (15:23:08.920) Baud rate is changed to 19200
- (15:23:09.248) [STX] 0.0.0(40000331) [CR][LF] 0.2.0(V01.00) [CR][LF] 0.9.1- (15:22:56) [CR][LF] 0.9.2(21-05-
08) [CR][LF] 0.9.5(6) [CR][LF] 1.8.0(000021.278*kWh) [CR][LF] 1.8.1(000015.015*kWh) [CR][LF]
1.8.2(000002.084*kWh) [CR][LF] 1.8.3(000003.001*kWh) [CR][LF] 1.8.4(000001.178*kWh) [CR][LF]
1.6.0(000.000*kW)(21-05-01,00:00) [CR][LF] 96.1.3(21-03-19) [CR][LF] 96.2.5(21-03-19) [CR][LF] 32.7.0(230.8)
[CR][LF] 52.7.0(000.0) [CR][LF] 72.7.0(000.0) [CR][LF] 31.7.0(000.0) [CR][LF] 51.7.0(000.0) [CR][LF]
71.7.0(000.0) [CR][LF] 14.7.0(50.0) [CR][LF] 1.6.0*1(000.024*kW)(21-04-01,14:14) [CR][LF]
1.6.0*2(001.008*kW)(21-03-31,10:24) [CR][LF] 1.6.0*3(000.000*kW)(30-05-01,00:00) [CR][LF]
1.6.0*4(000.000*kW)(30-04-01,00:00) [CR][LF] 1.6.0*5(000.000*kW)(22-01-01,00:00) [CR][LF]
1.6.0*6(000.000*kW)(21-11-01,00:00) [CR][LF] 1.6.0*7(000.000*kW)(21-09-01,00:00) [CR][LF]
1.6.0*8(000.000*kW)(21-08-01,00:00) [CR][LF] 1.6.0*9(000.000*kW)(21-06-01,00:00) [CR][LF]
1.6.0*10(000.000*kW)(21-04-01,00:00) [CR][LF] 1.6.0*11(000.000*kW)(21-02-01,00:00) [CR][LF]
1.6.0*12(000.000*kW)(21-12-01,00:00) [CR][LF] 96.71*1(21-04-01,00:00)(20) [CR][LF] 96.71*2(30-06-01,00:00)(02)
[CR][LF] 96.71*3(30-05-01,00:00)(06) [CR][LF] 96.71*4(30-04-01,00:00)(01) [CR][LF] 96.71*5(22-01-01,00:00)(07)
[CR][LF] 96.71*6(21-11-01,00:00)(01) [CR][LF] 96.71*7(21-09-01,00:00)(01) [CR][LF] 96.71*8(21-08-01,00:00)(01)
[CR][LF] 96.71*9(21-06-01,00:00)(01) [CR][LF] 96.71*10(21-04-01,00:00)(01) [CR][LF] 96.71*11(21-02-
01,00:00)(01) [CR][LF] 96.71*12(21-12-01,00:00)(01) [CR][LF] 96.50(05001000110014000150030904309999)
[CR][LF] 96.51(0600170022009999999999999999999999) [CR][LF] 96.52(0600170022009999999999999999999999)
[CR][LF] 96.60(12345678) [CR][LF] 96.61(12340000) [CR][LF] 96.62(12340000) [CR][LF] 96.2.2*1(21-03-31,10:02)
[CR][LF] 96.2.2*2(21-03-31,10:00) [CR][LF] 96.2.2*3(21-03-31,10:00) [CR][LF] 96.2.2*4(21-03-31,09:58) [CR][LF]
96.2.2*5(21-03-31,09:57) [CR][LF] 96.2.2*6(21-03-31,09:57) [CR][LF] 96.2.2*7(21-03-31,09:55) [CR][LF]
96.2.2*8(21-03-31,09:53) [CR][LF] 96.2.2*9(21-03-31,09:52) [CR][LF] 96.2.2*10(21-03-31,09:50) [CR][LF]
1.8.1*1(000015.015*kWh) [CR][LF] 1.8.1*2(000015.010*kWh) [CR][LF] 1.8.1*3(000014.983*kWh) [CR][LF]
1.8.1*4(000014.983*kWh) [CR][LF] 1.8.1*5(000014.983*kWh) [CR][LF] 1.8.1*6(000014.983*kWh) [CR][LF]
1.8.1*7(000014.983*kWh) [CR][LF] 1.8.1*8(000014.983*kWh) [CR][LF] 1.8.1*9(000014.983*kWh) [CR][LF]
1.8.1*10(000014.983*kWh) [CR][LF] 1.8.1*11(000014.983*kWh) [CR][LF] 1.8.1*12(000014.983*kWh) [CR][LF]
1.8.2*1(000002.084*kWh) [CR][LF] 1.8.2*2(000002.084*kWh) [CR][LF] 1.8.2*3(000001.832*kWh) [CR][LF]
1.8.2*4(000001.832*kWh) [CR][LF] 1.8.2*5(000001.832*kWh) [CR][LF] 1.8.2*6(000001.832*kWh) [CR][LF]

```

1.8.2*7(000001.832*kWh) [CR][LF] 1.8.2*8(000001.832*kWh) [CR][LF] 1.8.2*9(000001.832*kWh) [CR][LF]
 1.8.2*10(000001.832*kWh) [CR][LF] 1.8.2*11(000001.832*kWh) [CR][LF] 1.8.2*12(000001.832*kWh) [CR][LF]
 1.8.3*1(000003.001*kWh) [CR][LF] 1.8.3*2(000003.001*kWh) [CR][LF] 1.8.3*3(000003.001*kWh) [CR][LF]
 1.8.3*4(000003.001*kWh) [CR][LF] 1.8.3*5(000003.001*kWh) [CR][LF] 1.8.3*6(000002.960*kWh) [CR][LF]
 1.8.3*7(000002.960*kWh) [CR][LF] 1.8.3*8(000002.960*kWh) [CR][LF] 1.8.3*9(000002.960*kWh) [CR][LF]
 1.8.3*10(000002.960*kWh) [CR][LF] 1.8.3*11(000002.960*kWh) [CR][LF] 1.8.3*12(000002.960*kWh) [CR][LF]
 1.8.4*1(000001.178*kWh) [CR][LF] 1.8.4*2(000001.172*kWh) [CR][LF] 1.8.4*3(000001.172*kWh) [CR][LF]
 1.8.4*4(000001.172*kWh) [CR][LF] 1.8.4*5(000001.172*kWh) [CR][LF] 1.8.4*6(000001.172*kWh) [CR][LF]
 1.8.4*7(000001.172*kWh) [CR][LF] 1.8.4*8(000001.172*kWh) [CR][LF] 1.8.4*9(000001.172*kWh) [CR][LF]
 1.8.4*10(000001.172*kWh) [CR][LF] 1.8.4*11(000001.172*kWh) [CR][LF] 1.8.4*12(000001.172*kWh) [CR][LF]
 0.8.0(15*min) [CR][LF] 0.8.4(60*min) [CR][LF] 96.6.1(1) [CR][LF] 96.71(21-05-01,00:00)(01) [CR][LF] 96.70(21-03-
 23,17:14) [CR][LF] 96.2.2(21-03-31,10:03) [CR][LF] 96.90.0(0) [CR][LF] 96.90.1(+01:00,21-03-28,03:00;21-10-
 31,04:00) [CR][LF] 96.90.2(+01:00,22-03-27,03:00;22-10-30,04:00) [CR][LF] 96.90.3(+01:00,23-03-26,03:00;23-10-
 29,04:00) [CR][LF] 96.90.4(+01:00,24-03-31,03:00;24-10-27,04:00) [CR][LF] 96.90.5(+01:00,25-03-30,03:00;25-10-
 26,04:00) [CR][LF] 96.90.6(+01:00,26-03-29,03:00;26-10-25,04:00) [CR][LF] 96.90.7(+01:00,27-03-28,03:00;27-10-
 31,04:00) [CR][LF] 96.90.8(+01:00,28-03-26,03:00;28-10-29,04:00) [CR][LF] 96.90.9(+01:00,29-03-25,03:00;29-10-
 28,04:00) [CR][LF] 96.90.10(+01:00,30-03-31,03:00;30-10-27,04:00) [CR][LF] 96.90.11(+01:00,31-03-30,03:00;31-
 10-26,04:00) [CR][LF] 96.90.12(+01:00,32-03-28,03:00;32-10-31,04:00) [CR][LF] 96.7.4(12) [CR][LF] 96.77.4*1(21-
 03-26,14:49;21-04-01,00:02) [CR][LF] 96.77.4*2(21-03-24,11:58;21-03-24,13:40) [CR][LF] 96.77.4*3(21-03-
 24,11:47;21-03-24,11:52) [CR][LF] 96.77.4*4(21-03-24,11:40;21-03-24,11:45) [CR][LF] 96.77.4*5(21-03-
 24,11:33;21-03-24,11:38) [CR][LF] 96.77.4*6(21-03-24,11:25;21-03-24,11:31) [CR][LF] 96.77.4*7(21-03-
 24,11:17;21-03-24,11:23) [CR][LF] 96.77.4*8(21-03-24,11:09;21-03-24,11:14) [CR][LF] 96.77.4*9(21-03-
 24,10:29;21-03-24,11:05) [CR][LF] 96.77.4*10(21-03-24,10:19;21-03-24,10:26) [CR][LF] 96.7.5(16) [CR][LF]
 96.77.5*1(22-02-01,00:18;22-02-01,00:27) [CR][LF] 96.77.5*2(22-02-01,00:07;22-02-01,00:13) [CR][LF]
 96.77.5*3(22-01-01,00:13;22-02-01,00:07) [CR][LF] 96.77.5*4(21-12-31,23:59;22-01-01,00:10) [CR][LF]
 96.77.5*5(21-12-31,23:59;22-01-01,00:02) [CR][LF] 96.77.5*6(21-12-31,23:58;22-01-01,00:02) [CR][LF]
 96.77.5*7(21-03-26,14:36;21-03-26,14:41) [CR][LF] 96.77.5*8(21-03-26,14:27;21-03-26,14:34) [CR][LF]
 96.77.5*9(21-03-26,14:10;21-03-26,14:15) [CR][LF] 96.77.5*10(21-03-26,13:49;21-03-26,14:08) [CR][LF]
 96.7.6(25)(00005*min) [CR][LF] 96.77.6*1(21-04-01,00:10;21-04-01,00:10) [CR][LF] 96.77.6*2(21-04-01,00:07;21-
 04-01,00:07) [CR][LF] 96.77.6*3(21-04-01,00:05;21-04-01,00:05) [CR][LF] 96.77.6*4(21-04-01,00:05;21-04-
 01,00:05) [CR][LF] 96.77.6*5(21-04-01,00:04;21-04-01,00:04) [CR][LF] 96.77.6*6(21-04-01,00:04;21-04-01,00:04)
 [CR][LF] 96.77.6*7(21-04-01,00:04;21-04-01,00:04) [CR][LF] 96.77.6*8(21-04-01,00:04;21-04-01,00:04) [CR][LF]
 96.77.6*9(21-04-01,00:03;21-04-01,00:03) [CR][LF] 96.77.6*10(21-04-01,00:02;21-04-01,00:02) [CR][LF]
 96.7.0(0102) [CR][LF] 96.77.0*1(21-05-07,15:07;21-05-07,15:26) [CR][LF] 96.77.0*2(21-04-21,16:35;21-05-
 07,14:15) [CR][LF] 96.77.0*3(21-04-08,13:32;21-04-15,16:32) [CR][LF] 96.77.0*4(21-04-07,17:17;21-04-08,13:30)
 [CR][LF] 96.77.0*5(21-04-07,11:19;21-04-07,17:16) [CR][LF] 96.77.0*6(21-04-05,11:26;21-04-06,11:16) [CR][LF]
 96.77.0*7(21-04-05,09:32;21-04-05,09:45) [CR][LF] 96.77.0*8(21-04-05,09:05;21-04-05,09:15) [CR][LF]
 96.77.0*9(21-04-02,17:12;21-04-05,08:47) [CR][LF] 96.77.0*10(21-04-02,16:57;21-04-02,17:11) [CR][LF]
 96.77.0*11(21-04-02,16:33;21-04-02,16:45) [CR][LF] 96.77.0*12(21-04-02,16:01;21-04-02,16:19) [CR][LF]
 96.77.0*13(21-04-02,15:19;21-04-02,15:27) [CR][LF] 96.77.0*14(21-04-02,14:48;21-04-02,15:04) [CR][LF]
 96.77.0*15(21-04-02,14:11;21-04-02,14:35) [CR][LF] 96.77.0*16(21-04-02,11:56;21-04-02,13:26) [CR][LF]
 96.77.0*17(21-04-02,11:27;21-04-02,11:42) [CR][LF] 96.77.0*18(21-04-02,11:10;21-04-02,11:23) [CR][LF]
 96.77.0*19(21-04-02,10:25;21-04-02,11:01) [CR][LF] 96.77.0*20(21-04-02,09:51;21-04-02,10:08) [CR][LF]
 96.77.0*21(21-04-02,09:34;21-04-02,09:39) [CR][LF] 96.77.0*22(21-04-02,08:46;21-04-02,09:10) [CR][LF]
 96.77.0*23(21-04-01,17:28;21-04-02,08:42) [CR][LF] 96.77.0*24(21-04-01,17:09;21-04-01,17:20) [CR][LF]
 96.77.0*25(21-04-01,16:42;21-04-01,16:56) [CR][LF] 96.77.0*26(21-04-01,16:32;21-04-01,16:35) [CR][LF]
 96.77.0*27(21-04-01,16:28;21-04-01,16:31) [CR][LF] 96.77.0*28(21-04-01,16:15;21-04-01,16:18) [CR][LF]
 96.77.0*29(21-04-01,16:04;21-04-01,16:08) [CR][LF] 96.77.0*30(21-04-01,14:49;21-04-01,15:55) [CR][LF]
 96.77.0*31(21-04-01,14:19;21-04-01,14:31) [CR][LF] 96.77.0*32(21-04-01,13:55;21-04-01,14:02) [CR][LF]
 96.77.0*33(21-04-01,13:37;21-04-01,13:47) [CR][LF] 96.77.0*34(21-04-01,11:27;21-04-01,11:58) [CR][LF]
 96.77.0*35(21-04-01,11:12;21-04-01,11:18) [CR][LF] 96.77.0*36(21-04-01,11:02;21-04-01,11:09) [CR][LF]
 96.77.0*37(21-04-01,10:13;21-04-01,10:19) [CR][LF] 96.77.0*38(21-04-01,09:59;21-04-01,10:08) [CR][LF]
 96.77.0*39(21-04-01,09:42;21-04-01,09:56) [CR][LF] 96.77.0*40(21-04-01,09:29;21-04-01,09:36) [CR][LF]

96.77.0*41(21-04-01,08:58;21-04-01,09:24) [CR][LF] 96.77.0*42(21-03-31,17:51;21-04-01,08:40) [CR][LF]
 96.77.0*43(21-03-31,16:42;21-03-31,17:05) [CR][LF] 96.77.0*44(21-03-31,15:25;21-03-31,16:10) [CR][LF]
 96.77.0*45(21-03-31,14:42;21-03-31,14:48) [CR][LF] 96.77.0*46(21-03-31,13:28;21-03-31,13:35) [CR][LF]
 96.77.0*47(21-03-31,11:24;21-03-31,11:39) [CR][LF] 96.77.0*48(21-03-31,10:34;21-03-31,10:54) [CR][LF]
 96.77.0*49(21-03-31,10:23;21-03-31,10:29) [CR][LF] 96.77.0*50(21-03-31,10:04;21-03-31,10:12) [CR][LF]
 96.77.0*51(21-03-30,17:25;21-03-31,09:29) [CR][LF] 96.77.0*52(21-03-30,14:35;21-03-30,17:22) [CR][LF]
 96.77.0*53(21-03-30,14:25;21-03-30,14:35) [CR][LF] 96.77.0*54(21-03-30,10:56;21-03-30,11:01) [CR][LF]
 96.77.0*55(21-03-30,10:47;21-03-30,10:51) [CR][LF] 96.77.0*56(21-03-30,10:34;21-03-30,10:43) [CR][LF]
 96.77.0*57(21-03-29,17:35;21-03-30,10:22) [CR][LF] 96.77.0*58(21-03-29,16:46;21-03-29,17:47) [CR][LF]
 96.77.0*59(21-03-29,15:09;21-03-29,15:09) [CR][LF] 96.77.0*60(22-01-01,00:03;22-01-01,00:07) [CR][LF]
 96.77.0*61(21-03-29,13:37;21-03-29,13:41) [CR][LF] 96.77.0*62(21-03-29,11:12;21-03-29,13:33) [CR][LF]
 96.77.0*63(21-03-29,10:25;21-03-29,11:12) [CR][LF] 96.77.0*64(21-03-29,10:13;21-03-29,10:21) [CR][LF]
 96.77.0*65(22-02-03,17:28;22-02-03,17:34) [CR][LF] 96.77.0*66(22-02-01,00:30;22-02-03,16:28) [CR][LF]
 96.77.0*67(22-01-01,00:02;22-01-01,00:39) [CR][LF] 96.77.0*68(21-03-26,14:42;21-03-26,14:49) [CR][LF]
 96.77.0*69(21-03-26,12:01;21-03-26,13:33) [CR][LF] 96.77.0*70(21-03-26,11:44;21-03-26,11:47) [CR][LF]
 96.77.0*71(21-03-26,09:59;21-03-26,10:36) [CR][LF] 96.77.0*72(21-03-25,16:33;21-03-26,09:43) [CR][LF]
 96.77.0*73(30-03-25,15:02;30-03-25,16:04) [CR][LF] 96.77.0*74(30-03-25,14:04;30-03-25,14:43) [CR][LF]
 96.77.0*75(22-04-01,02:43;22-04-01,02:43) [CR][LF] 96.77.0*76(22-04-01,00:44;22-04-01,02:15) [CR][LF]
 96.77.0*77(22-04-01,00:22;22-04-01,00:26) [CR][LF] 96.77.0*78(22-02-01,00:03;22-02-01,00:05) [CR][LF]
 96.77.0*79(22-01-01,00:10;22-02-01,00:00) [CR][LF] 96.77.0*80(22-01-01,00:03;22-01-01,00:07) [CR][LF]
 96.77.0*81(21-03-24,17:35;21-03-25,08:58) [CR][LF] 96.77.0*82(21-03-24,17:23;21-03-24,17:28) [CR][LF]
 96.77.0*83(21-03-24,17:00;21-03-24,17:14) [CR][LF] 96.77.0*84(21-03-24,22:04;00-04-00,09:28) [CR][LF]
 96.77.0*85(21-03-24,14:29;00-08-00,09:20) [CR][LF] 96.77.0*86(21-03-24,12:08;21-03-24,13:40) [CR][LF]
 96.77.0*87(21-03-24,09:42;21-03-24,09:45) [CR][LF] 96.77.0*88(21-03-23,17:32;21-03-24,08:53) [CR][LF]
 96.77.0*89(21-03-23,15:27;21-03-23,17:31) [CR][LF] 96.77.0*90(21-03-23,14:50;21-03-23,14:55) [CR][LF]
 96.77.0*91(21-03-23,11:08;21-03-23,11:11) [CR][LF] 96.77.0*92(21-03-23,10:28;21-03-23,11:08) [CR][LF]
 96.77.0*93(21-03-23,09:09;21-03-23,09:47) [CR][LF] 96.77.0*94(21-03-22,17:15;21-03-23,08:50) [CR][LF]
 96.77.0*95(21-03-22,16:42;21-03-22,17:01) [CR][LF] 96.77.0*96(21-03-22,15:30;21-03-22,15:57) [CR][LF]
 96.77.0*97(21-03-22,15:18;21-03-22,15:26) [CR][LF] 96.77.0*98(21-03-22,14:16;21-03-22,14:33) [CR][LF]
 96.77.0*99(21-03-22,11:58;21-03-22,14:10) [CR][LF] 96.7.00(0201) [CR][LF] 96.77.00*1(21-05-07,17:21;21-05-07,17:21) [CR][LF]
 96.77.00*2(21-05-07,15:29;21-05-07,15:29) [CR][LF] 96.77.00*3(21-04-02,17:11;21-04-02,17:11) [CR][LF]
 96.77.00*4(21-04-01,16:36;21-04-01,16:39) [CR][LF] 96.77.00*5(21-04-01,14:13;21-04-01,14:13) [CR][LF]
 96.77.00*6(21-04-01,14:10;21-04-01,14:12) [CR][LF] 96.77.00*7(21-03-31,10:17;21-03-31,10:18) [CR][LF]
 96.77.00*8(21-03-31,10:16;21-03-31,10:17) [CR][LF] 96.77.00*9(21-03-31,10:12;21-03-31,10:15) [CR][LF]
 96.77.00*10(21-03-31,10:12;21-03-31,10:12) [CR][LF] 96.77.00*11(21-03-30,10:28;21-03-30,10:28) [CR][LF]
 96.77.00*12(21-03-30,10:24;21-03-30,10:25) [CR][LF] 96.77.00*13(21-03-30,10:24;21-03-30,10:24) [CR][LF]
 96.77.00*14(21-03-30,10:23;21-03-30,10:23) [CR][LF] 96.77.00*15(21-03-30,10:23;21-03-30,10:23) [CR][LF]
 96.77.00*16(21-03-30,10:23;21-03-30,10:23) [CR][LF] 96.77.00*17(21-03-30,10:23;21-03-30,10:23) [CR][LF]
 96.77.00*18(21-03-30,10:23;21-03-30,10:23) [CR][LF] 96.77.00*19(21-03-29,17:49;21-03-29,17:49) [CR][LF]
 96.77.00*20(21-03-29,17:47;21-03-29,17:48) [CR][LF] 96.77.00*21(21-03-29,16:11;21-03-29,16:13) [CR][LF]
 96.77.00*22(21-03-29,13:49;21-03-29,13:49) [CR][LF] 96.77.00*23(21-03-29,13:44;21-03-29,13:46) [CR][LF]
 96.77.00*24(21-03-29,13:36;21-03-29,13:36) [CR][LF] 96.77.00*25(22-02-01,00:30;22-02-01,00:30) [CR][LF]
 96.77.00*26(22-02-01,00:28;22-02-01,00:30) [CR][LF] 96.77.00*27(22-02-01,00:27;22-02-01,00:27) [CR][LF]
 96.77.00*28(22-02-01,00:25;22-02-01,00:26) [CR][LF] 96.77.00*29(22-02-01,00:17;22-02-01,00:18) [CR][LF]
 96.77.00*30(22-02-01,00:15;22-02-01,00:16) [CR][LF] 96.77.00*31(22-02-01,00:10;22-02-01,00:12) [CR][LF]
 96.77.00*32(22-01-01,00:13;22-01-01,00:14) [CR][LF] 96.77.00*33(22-01-01,00:13;22-01-01,00:13) [CR][LF]
 96.77.00*34(22-01-01,00:12;22-01-01,00:12) [CR][LF] 96.77.00*35(22-01-01,00:11;22-01-01,00:12) [CR][LF]
 96.77.00*36(22-01-01,00:08;22-01-01,00:10) [CR][LF] 96.77.00*37(22-01-01,00:41;22-01-01,00:43) [CR][LF]
 96.77.00*38(22-01-01,00:02;22-01-01,00:02) [CR][LF] 96.77.00*39(22-01-01,00:06;22-01-01,00:08) [CR][LF]
 96.77.00*40(22-01-01,00:02;22-01-01,00:02) [CR][LF] 96.77.00*41(21-04-01,00:06;21-04-01,00:07) [CR][LF]
 96.77.00*42(21-04-01,00:01;21-04-01,00:02) [CR][LF] 96.77.00*43(21-03-26,14:41;21-03-26,14:41) [CR][LF]
 96.77.00*44(21-03-26,14:35;21-03-26,14:36) [CR][LF] 96.77.00*45(21-03-26,14:33;21-03-26,14:34) [CR][LF]
 96.77.00*46(21-03-26,14:27;21-03-26,14:27) [CR][LF] 96.77.00*47(21-03-26,14:24;21-03-26,14:34) [CR][LF]

00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*55(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*56(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*57(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*58(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*59(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*60(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*61(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*62(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*63(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*64(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*65(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*66(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*67(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*68(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*69(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*70(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*71(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*72(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*73(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*74(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*75(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*76(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1[8][8]*77(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*78(00-00-00,00:00;[8]00-00-00,00:00) [CR][LF] 96.77.1*79(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96[8].77.1*80(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*81(00-00-00,0[8]0:00;00-00-00,00:00) [CR][LF] 96.77.1*82(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.7700,00:00) [CR][LF] [8][8]96.77.1*84(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*85(00-00-00[8],00:00;00-00-00,00:00) [CR][LF] 96.77.1*86(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*87(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*88(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*89(00-00-00,00:00;00-00-00,00:00)-00,00:00) [CR][LF] [8][8]96.77.1*98(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.1*99(00-00-00[8],00:00;00-00-00,00:00) [CR][LF] 96.7.10(0104) [CR][LF] 96.77.10*1(21-03-30,10:[8]23;21-03-30,10:23) [CR][LF] 96.77.10*2(22-01-01,00:07;22-01-01,00:10) [CR][8][LF] 96.77.10*3(21-03-24,09:59;21-03-24,10:00) [CR][LF] 96.77.10*4(21-03-24,09:57;[LF] 96.77.10*12([8][8]21-03-23,13:47;21-03-23,13:47) [CR][LF] 96.77.10*13(21-03-23,13:46;21-[8]03-23,13:46) [CR][LF] 96.77.10*14(21-03-23,13:46;21-03-23,13:46) [CR][LF] 96.7[8]7.10*15(21-03-23,13:46;21-03-23,13:46) [CR][LF] 96.77.10*16(21-03-23,1[8]3:46;21-03-23,13:46) [CR][LF] 96.77.10*17(21-03-23,13:45;21-03-23,13:45) [CR][LF] 96.727(21-03-23,13:43;21-03-23,1[8][8]3:43) [CR][LF] 96.77.10*28(21-03-23,13:42;21-03-23,13:42) [CR][LF] 96.77.10*29[8](21-03-23,13:42;21-03-23,13:42) [CR][LF] 96.77.10*30(21-03-23,13:42;21[8]-03-23,13:42) [CR][LF] 96.77.10*31(21-03-23,13:42;21-03-23,13:42) [CR][LF] 96.[8]77.10*32(21-03-23,13:41;21-03-23,13:41) [CR][LF] 96.77.10*33(21-03-23,13:41;21-03-23,13:39) [CR][LF] [8][8]96.77.10*42(21-03-23,13:39;21-03-23,13:39) [CR][LF] 96.77.10*43(21-03-[8]23,11:58;21-03-23,11:59) [CR][LF] 96.77.10*44(21-03-23,11:58;21-03-23,[8]11:58) [CR][LF] 96.77.10*45(21-03-23,11:58;21-03-23,11:58) [CR][LF] 96.77.10*4[8]6(21-03-23,11:57;21-03-23,11:57) [CR][LF] 96.77.10*47(21-03-23,11:57;21-03-23,1-03-23,11:54;21-03-23,11:54[8][8]) [CR][LF] 96.77.10*56(21-03-23,11:53;21-03-23,11:54) [CR][LF] 96.77.10*57(21-[8]03-23,11:53;21-03-23,11:53) [CR][LF] 96.77.10*58(21-03-23,11:53;21-03-[8]23,11:53) [CR][LF] 96.77.10*59(21-03-23,11:53;21-03-23,11:53) [CR][LF] 96.77.1[8]0*60(21-03-23,11:52;21-03-23,11:52) [CR][LF] 96.77.10*61(21-03-23,11:51;21-03-*71(21-03-23,11:45;21-03-23,[8][8]11:45) [CR][LF] 96.77.10*72(21-03-23,11:45;21-03-23,11:45) [CR][LF] 96.77.10*7[8]3(21-03-23,11:45;21-03-23,11:45) [CR][LF] 96.77.10*74(21-03-23,11:44;2[8]1-03-23,11:44) [CR][LF] 96.77.10*75(21-03-23,11:44;21-03-23,11:44) [CR][LF] 96[8].77.10*76(21-03-23,11:44;21-03-23,11:44) [CR][LF] 96.77.10*77(21-03-23,11:44;2[LF] 96.77.10*84(2[8][8]1-03-23,11:39;21-03-23,11:39) [CR][LF] 96.77.10*85(21-03-23,11:38;21-0[8]3-23,11:38) [CR][LF] 96.77.10*86(21-03-23,11:38;21-03-23,11:38) [CR][LF] 96.77[8].10*87(21-03-23,11:38;21-03-23,11:38) [CR][LF] 96.77.10*88(21-03-23,11[8]:38;21-03-23,11:38) [CR][LF] 96.77.10*89(21-03-23,11:38;21-03-23,11:38) [CR][LF] 96.775-07,14:15;00-00-00,00:00) [CR][LF] [8][8]96.77.2*2(22-02-01,00:08;22-02-01,00:12) [CR][LF] 96.77.2*3(21-03-23,1[8]5:12;21-03-23,15:17) [CR][LF] 96.77.2*4(21-03-23,10:22;21-03-23,10:26)[8] [CR][LF] 96.77.2*5(21-03-22,11:22;21-03-22,11:25) [CR][LF] 96.77.2*6(00-00-00[8],00:00;00-00-00,00:00) [CR][LF] 96.77.2*7(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.700,00:00;00-00-00,00:00) [CR][LF] [8][8]96.77.2*17(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*18(00-00-00[8],00:00;00-00-00,00:00) [CR][LF] 96.77.2*19(00-00-00,00:00;00-00-00,00:[8]00) [CR][LF] 96.77.2*20(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*21(00-[8]00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*22(00-00-00,00:00;00-00-00,00:00):00) [CR][LF] 96.77[8][8].2*30(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*31(00-00-00,00:0[8]0;00-00-00,00:00) [CR][LF] 96.77.2*32(00-00-00,00:00;00-00-00,00:00) [CR][LF] [8]96.77.2*33(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*34(00-00-00[8],00:00;00-00-00,00:00) [CR][LF] 96.77.2*35(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96..2*38(00-00-00,00:00;00-00-0[8][8]0,00:00) [CR][LF] 96.77.2*39(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*4[8]0(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*41(00-00-00,00:00;00[8]-00-00,00:00) [CR][LF] 96.77.2*42(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.7[8]7.2*43(00-00-00,00:00;00-00-00,00:00) [CR][LF] 96.77.2*44(00-00-00,00:00;00-00[LF] 96.77.2*45(00-00-00,00:00;00-00-


```

23,14:04) [CR][LF] 96.77.30*66(21-03-23,14:04;21-03-23,14:04) [CR][LF] 96.77.30*67(21-03-23,14:04;21-03-
23,14:04) [CR][LF] 96.77.30*68(21-03-23,14:03;21-03-23,14:03) [CR][LF] 96.77.30*69(21-03-23,14:03;21-03-
23,14:03) [CR][LF] 96.77.30*70(21-03-23,14:03;21-03-23,14:03) [CR][LF] 96.77.30*71(21-03-23,14:02;21-03-
23,14:02) [CR][LF] 96.77.30*72(21-03-23,14:02;21-03-23,14:02) [CR][LF] 96.77.30*73(21-03-23,14:02;21-03-
23,14:02) [CR][LF] 96.77.30*74(21-03-23,14:02;21-03-23,14:02) [CR][LF] 96.77.30*75(21-03-23,14:02;21-03-
23,14:02) [CR][LF] 96.77.30*76(21-03-23,14:01;21-03-23,14:01) [CR][LF] 96.77.30*77(21-03-23,14:01;21-03-
23,14:01) [CR][LF] 96.77.30*78(21-03-23,14:01;21-03-23,14:01) [CR][LF] 96.77.30*79(21-03-23,14:00;21-03-
23,14:01) [CR][LF] 96.77.30*80(21-03-23,14:00;21-03-23,14:00) [CR][LF] 96.77.30*81(21-03-23,14:00;21-03-
23,14:00) [CR][LF] 96.77.30*82(21-03-23,14:00;21-03-23,14:00) [CR][LF] 96.77.30*83(21-03-23,13:59;21-03-
23,13:59) [CR][LF] 96.77.30*84(21-03-23,13:59;21-03-23,13:59) [CR][LF] 96.77.30*85(21-03-23,13:59;21-03-
23,13:59) [CR][LF] 96.77.30*86(21-03-23,13:58;21-03-23,13:59) [CR][LF] 96.77.30*87(21-03-23,13:58;21-03-
23,13:58) [CR][LF] 96.77.30*88(21-03-23,13:58;21-03-23,13:58) [CR][LF] 96.77.30*89(21-03-23,13:58;21-03-
23,13:58) [CR][LF] 96.77.30*90(21-03-23,13:58;21-03-23,13:58) [CR][LF] 96.77.30*91(21-03-23,13:57;21-03-
23,13:57) [CR][LF] 96.77.30*92(21-03-23,13:57;21-03-23,13:57) [CR][LF] 96.77.30*93(21-03-23,13:57;21-03-
23,13:57) [CR][LF] 96.77.30*94(21-03-23,13:56;21-03-23,13:56) [CR][LF] 96.77.30*95(21-03-23,13:56;21-03-
23,13:56) [CR][LF] 96.77.30*96(21-03-23,13:56;21-03-23,13:56) [CR][LF] 96.77.30*97(21-03-23,13:56;21-03-
23,13:56) [CR][LF] 96.77.30*98(21-03-23,13:55;21-03-23,13:55) [CR][LF] 96.77.30*99(21-03-23,13:55;21-03-
23,13:55) [CR][LF] ! [CR][LF] [ETX]h

```

- (15:24:40.182) Serial port closed

- (15:24:40.292) =====

11.2 Programlama Modunda Okuma

```

- (15:40:02.027) =====
- (15:40:02.059) Serial port opened
- (15:40:02.074) =====
- (15:40:02.090) -> /?! [CR][LF]
- (15:40:02.121) Baud rate is changed to 300
- (15:40:02.340) /BYL6<2>BGZ(BT10.LP-R1) [CR][LF]
- (15:40:03.262) -> [ACK]
- (15:40:03.277) -> 061 [CR][LF]
- (15:40:03.527) Baud rate is changed to 19200
- (15:40:03.996) [SOH] P0 [STX] (40000331) [ETX]e
- (15:40:04.074) -> [SOH] R2 [STX] 0.0.0(00) [ETX]1
- (15:40:04.121) [STX] 0.0.0(40000331) [ETX]7
- (15:40:04.137) -> [SOH] B0 [ETX]1
- (15:40:04.215) Serial port closed
- (15:40:04.230) =====
- (15:40:04.262) Serial port opened
- (15:40:04.277) =====
- (15:40:04.293) Baud rate is changed to 300
- (15:40:27.792) Serial port closed
- (15:40:27.839) =====

```

Ek-B Mass Entegrasyon Dokümanı (Standart API)

Sunulan uygulama entegrasyon dokümanında en az aşağıdaki fonksiyonlar yer alacaktır.

- Meter
 - Create (POST)
 - Delete (DELETE)
 - Get (GET)
 - AddCommunicationDevice (PUT)
 - RemoveCommunicationDevice (PUT)++
- CommunicationDevice
 - Create (POST)
 - Delete (DELETE)
 - Get (GET)
- General Orders (execute directive)
 - MonitorOrder (GET)
 - GetOrder (GET)
 - GetOrders (GET)
- Meter Commands
 - GetDirectives (GET)
 - ExecuteDirective (POST)
- Communication Device Commands
 - Configure (POST)
 - CreateReadSchedule (POST)
 - DeleteReadSchedule (DELETE)
 - GetReadSchedules (GET)
 - UpdateFirmware (POST)
 - Reset (POST)
- Meter Reading Data
 - LastEndex (GET)
 - EndexesByDate (GET)
 - WarningsByDate (GET)
 - OutagesByDate (GET)
 - ProfiesByDate (GET)
- CommunicationDeviceData
 - DeviceInformation (GET)
 - WarningsByDate (GET)
 - OutagesByDate (GET)