EOPY Project X

Vibration Analysis

info@eopy.com.tr
+90(232) 4624498

07 EYLÜL 2021

EOPY
Bornova Caddesi İzeltaş Galeria Sitesi no:7/1-I Işıkkent/Bornovaİzmir-TÜRKİYE

CHAPTER 1

Genel Kullanım Kuralları

SCD Manage Komutları ve Açıklamaları :

1- **Set Mode** : SCD uzun süre kapalı kalması halinde açıldığı zaman komutları yerine getirmemektedir. Bunun için uygulamanın ilk çalıştığı zaman SCD'nin Mode selectionu **True** değeri yollanarak tetiklenerek bu sorun giderilmektedir.

2- **Self Test** : Tüm sensörleri test eder ve doğru ise Mobusta belirtilen yerdeki değer 0 olur.

3- **Delete Flash Memory** : Kayıt alındıktan sonra silme işlemini yapmamız gerekir.

4- **Reset SCD** : SCD ' yi bazı durumlarda (Set Mode yapsak da bazen cevap vermemekte) sıfırlamak istediğimizde çağrılacak fonksiyondur.

5- Set Sensör Speed : Sensörün 4 tür hız ayarı olup 0'dan 3'e kadar parametre yollanabilir. 0 en yavaş 3 en hızlıdır. (Burada 4 değeri de yollanabilir fakat o zaman tek eksen kayıt alacağı için bunu projeye eklemedik)

6- Stream With Time : Verilen zaman kadar yayın yapmamızı sağlar burada zamanın dakika mı saat mi olduğunu ise Modbus Serverde Ayarlanan yere bitler yollanarak ayarlanabilir. (Max 15 saniye 0 olarak yollarsa zamanı herhangi bir şey yapmaz)

7- **Stream With No Time :** Kullanıcı koparsa sürekli yayın kesilir . Kullanıcı kapatmak isterse yetkiyi(Modbus Server'da) 0 yapması gerekir

8- **Download Record** : Kaydın PC'ye ya da yazılım çalıştığı yere indirilmesi sağlanır. Burada dosyaların tarihlere göre kayıt alınması ve adlarının öyle yapılması denendi. Güncel projede bu eklenti kaldırıldı. (Eklenmeli)

9- **Download Status** : İndirmede hata ya da eksik olup olmadığını kontrol ediyoruz. Bu koda göre indirene kadar istek yollanması planlandı.

10- **Record** : Kayıt alma işleminin kaç saniyede olacağı parametresi ile kayıt hızı parametresi girilir. Burada kayıt Flasha olur. 6 saniyeyi geçen kayıt yapılması tavsiye edilmez. Uzun verilerin indirme süresi de çok uzun olmaktadır.Maksimum kayıt zamanları sensör hızına göre değişir.

11- **Periodic Live Stream** : Belirli periyotlar arasında canlı yayın yapmasını sağlayan fonksiyondur.

12- **Periodic Record** : Belirli zaman aralıklarında kayıt alıp durması planlanır. Kayıt alıcağı zaman ve hızı parametre olarak yollanır.

13- **Periodic Stream and Record** : Belirli zaman aralığında kayıt belirli zaman aralığında süreli stream yapması sağlanır. Yayın hızı ile kayıt hızı farklı olabilir. Parametreleri kayıt zamanı , canlı yayın zamanı , v, canlı yayın sensör hızı , kayıt hızı şeklindedir.

Modbus Manage Komutları ve Açıklamaları :

1- **Select Status** : Hangi komutun geldiğini değerlendirir ve belirli bir hiyerarşik yapısı vardır . Bölümün sonunda bu hiyerarşik yapı belirtilmiştir. Burada numarası ya da return değeri küçük olan diğer kendinden sonra gelen komutları pasif yapar. Örneğin kayıt alma streamden daha önceliklidir ama bu iki komut aynanda yollanırsa kayıt alınır stream yapılmaz.

2- **RotationalObj** : Burada alınan kayıtları analiz edilmesi ve çıktılarının ve sonuçlarının modbusa ve ekrana verilmesi için oluşturulan bir Class'dır. Detaylarında analiz fonksyonları ve fft gibi şeyler içerir buranın yapısı daha sonraki bölümlerde açıklanacaktır.

3- **Modbus SCD Gate** : Modbustan düzenli verileri okuyarak bu verilerin ve isteklerinin yerine gelmesi ve sonuçlarının Modbusa yazılmasını sağlar. SCD ve Modbus arasında bir köprü olup birbirlerinin çökmemesini sağlarlar.

CHAPTER 2

Modbus Server ve Yapısı :

- **Stream Verileri**

Modbusta 0 ile 2 arasında adreslenen registerlarda X , Y , Z verilerinin yayın sırasındaki anlık değerleri saklanmaktadır. Yayın yapılırken kullanıcı modbus üzerinden bunları anlık görmek istiyorsa burayı okumalıdır.

Modbus üzerinde 3 ile 68 arası adreslenen registerlar X , Y , Z'nin 1 saniyedeki değerlerini ve varyanslarının tutulduğu yerdir. Burada 66 adet olmasının sebebi maksimum hızda anlık olarak en fazla 22 verinin gelebilmesindendir. X , Y , Z verileri 16 bitlik int değeridir. Variance değerleri 32 bitlik float değerlerdir.

- **SCD Yönetmek İçin Kullanılan Alanlar**

69 ve 70. Adresler Setting1 ve Setting2 olarak adlandırılmıştır. Setting1 ve Setting2'nin yapısı:

Setting 1							
Sensör Testi	Stream Sensör Hızı	Delete Sensör	Error Code	Stream Time	Süreli Yayın Süresiz Yayın	Yayın Aç/ Kapa	Periyodik Stream Aç/ Kapa
1 Bit	2 Bit	1Bit	4 Bit	4 Bit	1 Bit	1 Bit	1 Bit

Sensör Testi : 1 yollanırsa Sensör Testi Scd 'e başlamış olur.

Stream Sensör Hızı : Streamde sensörün hızını ayarladığımız alandır herhangi bir şey girilmez ise 0 default olarak kabul edilir.

Delete Sensör : 1 yollanırsa SCD'nin Flash hafızasını siler.

Error Code : 4 bitlik alanda 15 hata kodu vardır. Eğer hata kodu alanı 0 ise hata yok demektir. Hata kodlarının açıklaması ve değerleri aşağıda verilmiştir.

Bunların tablosu :

Hata Kodu	Açıklaması
1	Set Mode Sırasında Hata Oluştu
2	Self Test Çalıştırıldığında Hata Oluştu
3	Start Toggle Çalıştırıldığında Hata Oluştu
4	Stop Toggle Çalıştırıldığında Hata Oluştu
5	Reserved
6	Delete Flash Memory Çalıştırıldığında Hata Oluştu
7	SCD Reset Çalıştırıldığında Hata Oluştu
8	Set Sensör Speed Çalıştırıldığında Hata Oluştu
9	Stream With Time Çalıştırıldığında Hata Oluştu
10	Stream With No Time Çalıştırıldığında Hata Oluştu
11	Download Error
12	Download Status Error
13	Data Record Error
14	Periodic Live Stream Error
15	Periodic Record Error

Setting 2							
Yetki	Kayıt Al veya Alma	Periyodik Kayıt ya da Tek - Seferlik Kayıt	Record Time	Record Sensör Speed	Download Status	Periyotların saat dakika seçimi	Modbus Yapılan İşlem Status
1 Bit	1 Bit	1Bit	4 Bit	2 Bit	1 Bit	2 Bit	3 Bit

Setting 1 ve 2 de de 1’er bitlik sonlarında sonraki amaçlar için kullanılabilecek bitler bırakılmıştır.

Yetki : Yetki 1 yapmadan herhangi bir komut yollansa dahi Modbus Gate bunu geçersiz sayar. Her işlemde yetkinin 1'e çekilmesi gerekir. Eğer 0 Olursa Bilgisayar tarafında yönetme işlemi yapılacaktır.

Kayıt Al / Alma : 1 Yollanırsa Kayıt alma aktiftir , kayıt alma aktif edilmek istenmiyorsa 0 yollanır. Default değeri 0'dır.

Periyodik Kayıt / Tek Seferlik Kayıt : Periyodik kayıt alınmak istediğinde 1 yapılmalıdır. Tek seferlik kayıt alınmak istenirse 0 yapılmalıdır.

Record Time : Flash hafızasına kayıt alma zamanını ayarlar.

Record Sensör Speed : Flash hafızasına kayıt alma hızını belirler.

Download Status : Burada indirmenin kopup komadığı , hata olup olmadığını kontrol edilir. İndirme başarılıysa 2 değeri yazılır.(SCD deki değerler ile aynı algoritmamız vardır.)

Periyotların Saat Dakika Seçimi : 00 = Saniye , 01 = Dakika , 10 = Saat , 11 = Gün anlamı taşır. Eski kod versiyonlarında bulunmakta güncel olana eklenmesi gerekmektedir.

Modbus Yapılan İşlem Status : Bir işlem bitmeden yeni bir işlem gelmemelidir.Gelirse cevapsız kalacaktır. İşlemin bitip bitmediğini buradan modbus client kontrol ederek istekte bulunur. 000 ise statü boştur. Yeni komut yollanabilir.

CHAPTER 3

Bazı Önemli Notlar :

1- ***Modbus Tcp Server*** 502 portunda çalışmakta olup Linux'ta (Debian piOS) port yönlendirmenin otomatik yapılabilmesi için aşağıdaki komutların tek seferlik olarak girilmesi gerekmektedir :

(Nerede oluşturulacağının önemi yok ama tavisye olarak usr/pi/ de oluşturma önerilir)

Sudo nano ourScript.sh

Burada script dosyası oluşacaktır içerisine :

```
#!/bin/bash

sudo iptables -t nat -A PREROUTING -p tcp --dport 502 -j REDIRECT --to-ports 5020
port yönlendirme
```

komutu yazılır ve ctrl + x tuşuna basılarak kaydetmek istiyor musunuz gelir . Burada kaydedin.
Daha sonra bunun executable dosya yapmak için

Sudo chmod u+x ourScript.sh

Komutunu çalıştırın.

Sudo crontab -e

Bu komut sonrasinda bize bir not defteri seçmemizi isteyecektir bunun herhangi bir önemi yoktur. (Nano kullandık nanonun seçilmesi önerilir.)

Seçtikten sonra crontab açılacaktır en alt kısma

@reboot (scriptimizinYolu)

Bunu crontabda en alta yazıp kaydedin yeniden başlattığınızda otomatik bu komutu çalıştıracaktır.

2- ***Statu Kodlarındaki Hiyerarşiler*** şu şekildedir :

Status 0 : Kapalı Komuta Hazır
Status 1 : Kayıt Alınacak ve Periyodik Olacak
Status 2 : Kayıt Alınacak
Status 3: Periyodik Stream Ve Record çalışıyor
Status 4: Sensör testi
Status 5: Delete Sensör Çalışmakta
Status 6 : Yayın Açık ve Periyodik
Status 7 : Yayın Açık ve Tek seferlik

Hiyerarşi :
Status 1 > 2> 3 > 4 > 5 > 6 > 7

1 'deki komut ile 3'teki komut aynanda yollanırsa 1 çalışır.
7. Komut ile 3. Komut aynanda yollanırsa 3 çalışır vb.

3- HMI tasarımında yararlı olabilecek ayrımlar :

HMI kısmı tasarlanırken kolaylık olması adına komutlar şu şekilde ayrılmıştır :

1 Bitlik Toggle Komutları :

Periodic Stream
Süreli Süresiz Yayın
Delete Sensör
Sensör Testi
Periodic Aç Kapa
Record Aç Kapa
Stream Aç Kapa
Yetki

Sadece Display Yapılacak Kısımlar :

Error Code 4 Bit
Status 3 Bit
Download Status 2 Bit

Hem İnput Hem Output Olanlar :

Stream Time 4 Bit
Stream Sensör Speed 2 Bit
Period Delay Time 4 Bit
Record Sensör Speed 2 Bit
Record Time 4 Bit

4- Ip Adreslerinin Alınması için yapılan değişiklikler

Modbus Serverin içerisinde pymodbus library'si kullanılarak server oluşturulmuştur. Serverda bağlanan kişilerin Ip Adreslerini alabilmemiz ve bağlantının koptuğunu anlayabilmemiz için bu kütüphanelerde değişiklik yapılmıştır. Bunun için bu işlemleri versiyonlarını kontrol ederek belirttiğimiz dosyaları otomatik değiştirilen bir kod yazılmıştır. Bu kodun 1 kere çalıştırılması yeterlidir.

5- SCD'den alınan verilerin anlamlandırılması ile ilgili kısım :

SCD üzerinden gelen veriler Tarihlerine göre otomatik klasör oluşturularak kaydedilecektir. SCD içerisindeki gelen byte dizileri int değerlerine dönüştürülerek kaydedilmektedir. Bu datalar üzerinde Bosch firması ile de iletişim kurmamıza rağmen anlamlandırmada bazı sıkıntılar vardır. Her 56 Pakette (Bu 148 adet X , Y , Z verisine denk gelmektedir) bir kayma yaşanmaktadır. Başlangıçtaki ilk paket header olmakla beraber 4-8. Bitleri arasında toplam kaç paket gelmesi gerektiği yazmaktadır . İndirmedeki % dilimi de buna göre çalışmaktadır.

Son kısım footer olarak geçmekte ve o da analiz edilebilmektedir. X , Y , Z verilerin anlamlandırılmasında 56'lı paketler şeklinde yorumlanmaya gidilerek büyük ölçüde doğru fakat kaymış bilgi elde etmekteyiz. Burada ilk başladığımızdan itibaren her 56 pakette 1 artacak şekilde bir sayacımız bulunmaktadır. Bu sayaç tek sayı olduğu durumda 5. Bit sona eklenmektedir. Bu sayede kaybolan 1 bilgi kayıp edilmemiş olur. Bazı durumlarda ise (Bu hem tek hem de çift olduğu durumlarda geçerli bu sayaç için) 19. Biti kontrol edilerek ona göre bir algoritma çıkarılmıştır. Bunlar kendi çıkarımlarla bulabildiğimiz en optimum çözümdür.

Kodu örneklemek gerekirse :

```
counter = 1
flag = True
for i in value:
    a =bytearray(i)
    try:
        if counter % 57== 0:
            flag = not flag
            versionFlag = True
            counter = 1
        if flag :
            first(a , counter)
        elif not flag:

            if a[19] >225 :
                a.append(255)
            else:
                a.append(0)
            second(a , counter)
    except :
        print(a)
        print(len(a))
        print(counter)
        break
    counter +=1
```

Counter Paket sayacını ifade eder 1'den başlama sebebi ilk paketin header olmasıdır. 19.byte'ın kontrolü de burada gözükmektedir. Burada bazı sorunlar da her 56 paketten sonra gelen ilk pakette normal rutin bir anlamlandırmaya sahip olmamasıdır bunun da örneği :

```
if  counter % 56 == 1:
    if  not versionFlag:
        f.write((str(s16floatfactor(a[10:12],0.1))+ " " + str(s16floatfactor(a[12:14],0.1)) + " " + str(s16floatfactor(a[14:16],0.1))))
        f.write("\n")
        f.write((str(s16floatfactor(a[16:18],0.1))+ " " + str(s16floatfactor(a[18:20],0.1))))
    else:
        f.write(str( str(s16floatfactor(a[4:6],0.1)) + " " + str(s16floatfactor(a[6:8],0.1))    +" "+str(s16floatfactor(a[8:10],0.1))))
        f.write("\n")
        f.write((str(s16floatfactor(a[10:12],0.1))+ " " + str(s16floatfactor(a[12:14],0.1)) + " " + str(s16floatfactor(a[14:16],0.1))))
        f.write("\n")
        f.write(str(str(s16floatfactor(a[16:18],0.1))+ " " + str(s16floatfactor(a[18:20],0.1))))
elif counter == 56:
```

Bu kısımda 56'lık paketin içerisinde ilk anlamlı veri 10-12. Bytlarda başlamaktadır. Normalde bu çift olduğu durumlarda 4-6 , tek olduğu durumlarda 5-7'den başlamaktadır.

6- FFT ve Peak Durumlarının Analizi :

FFT formu uygulamak için Y değerlerinin yollanması gerekmektedir. Burada dikkat edilmesi gereken fft bize oluşan sinüs dalgaların Euler Formülüne göre bir matris döndürmektedir. Biz frekansları alabilmek için ax.gca () fonksiyonunu kullanmaktayız.

Burada peak seviyelerinin tespiti için peakutils kütüphanesi kullanılmaktadır. Belirlediğimiz aralıkta peaki aramak için ise şu formülü kullanmaktayız :

Frekans Artış Oranımız = Toplam X Eksenindeki Değer sayımız / Maximum Frekans Değeri

Buradan istediğimiz frekansın hangi x değerleri arasında olduğu ise :

İstediğimiz Başlangıç Frekansı // Frekans Artış Değeri

Not : '//' Yanlış kullanılmamıştır . Python'da integer'a yuvarlanarak bölme yapmamızı sağlar. Burada başlangıçta istediğimiz frekansın ilk alacağı değeri bulmuş oluruz. Daha sonra bitmesini istediğimiz frekansın da x değeri bulunarak buradan sub array alınarak peak fonksiyonuna gönderilir. Burada tolerans aralıklarını peakutils kütüphanesini kullanarak rahatça verebilmekteyiz. Gerekli açıklamalar ve daha fazla detay için aşağıdaki link yararlı olacaktır :

Complete Reference — PeakUtils 1.3.3 documentation