



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

İlk gün staj programının oluşturulması, staj boyunca öğrenilecek ve yapılacak şeylerin listesinin çıkartılması.

Program:

Seyrüsefer Yazılımı – 1 gün (Pazartesi)

Raspberry pi + Kamera, Python yazılım dili- 2 gün (Çarşamba- Cuma)

PLC – 2 gün (Salı - Perşembe)

### **Seyrüsefer Yazılımı İçin Kullanılacak Harita Programının Oluşturulması İçin Gereken Harita Kütüphanesinin (GMap.NET) Uygulamaya Eklenmesi**

Windows Forms projesine *GMap.NET.Core.dll* ve *GMap.NET.WindowsForms.dll* başvuru dosyaları eklendikten sonra araç kutusuna da eklenir ve *GMapControl* aracı forma sürüklenerek projeye dahil edilmiş olur.

GMapControl sınıfının bazı özellikleri şunlardır:

- **CanDragMap <bool>**: Fare tuşuyla harita kaydırmayı aktif eder.
- **MarkersEnabled <bool>**: İşaretleri aktif eder.
- **PolygonsEnabled <bool>**: Alanları aktif eder.
- **RoutesEnabled <bool>**: Rotaları aktif eder.
- **MouseWheelZoomType <enum>** : Harita zoom tipini belirler.
- **MinZoom <int>** : Yapılabilecek en az zoom derecesini belirler.
- **MaxZoom <int>** : Yapılabilecek en fazla zoom derecesini belirler.
- **MouseWheelZoomEnabled <bool>**: Fare tekerleğiyle zoomu aktifleştirir.
- **Zoom <int>** : Mevcut zoom derecesini gösterir.
- **Position <PointLatLng>** : Harita merkezinin konumunu belirler.

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 1. PLC'ye Giriş:

İlk öğrenilecek olan şeyler, PLC hakkında genel bir fikir sahibi olmak, PLC'nin nerelerde ve ne için kullanıldığını öğrenmek ve TIA Portal yazılımının nasıl kullanıldığını öğrenmek.

Kullanacağımız PLC Modeli SIEMENS Simatic S7-1200 CPU1214C DC/DC/DC PLC. Bu Modeli S7-1200 Eğitim seti içerisinde kullanacağız.

### 1.1 PLC Nedir?

Programlanabilir lojik kontrolcü, makine kontrolü ve fabrika üretim hatları gibi alanlarda kullanılabilen otomasyon cihazıdır. Bilgisayarlara göre daha fazla giriş ve çıkışı olan bu cihazların en büyük artıları gürültülere, mekanik darbelere ve sıcaklık farklılıklarına dayanıklı olmalarıdır.

PLC'ler, cihazları üreten şirketlere göre farklı işletim sistemlerine sahip olur. Bu işletim sistemlerine kontrol edilecek sisteme uygun programlar yüklenir. Bu programlar da ms düzeyinde girişleri alarak gerçek zamanlıya yakın çıkışlar üretecek şekilde çalışır.

PLC'ler aşağıdaki 4 ana bölümden oluşur.

- Merkezi işlem birimi :

CPU, yazılan programdaki işlemleri gerçekleştiren birimdir.

- Giriş birimi :

Kontrol edilen sistemle ilgili analog girişleri PLC'nin anlayacağı sayısal seviyeye dönüştüren birimdir. Giriş birimi voltaj değerleri DC veya AC olabilir.

- Hafıza birimi :

PLC içinde farklı görevler yapan hafızalar bulunur. Bir kısmı yazılan programı saklarken bir kısmı veri saklar.

- Çıkış birimi :

PLC'de hesaplanan lojik gerilimi ilgili kumanda elemanlarını sürmek amacıyla dönüştüren birimdir. Çıkış röleli veya transistörlü devreden oluşabilir. Transistörlü çıkış

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

us düzeyinde çalışabilirken röleli çıkış ms düzeyinde çalışabilir.

## 1. Raspberry Pi Nedir ? Nasıl Kullanılır ?

Raspberry Pi, üzerinde normal bilgisayarlarda ki gibi USB girişi, Ethernet portu, HDMI çıkışı bulunduran, bilgisayarlardan farklı olarak üzerinde farklı sayılarda GPIO pin bulunan kredi kartı boyutlarında bulunan açık kaynak bir çeşit bilgisayardır. Raspberry pi üzerine bir çok işletim sistemi kurulabilir. (Örn. Windows 10, Linux, Raspbian, Noobs, Ubuntu vs.)

Raspberry Pi kullanabilmek için ilk önce üzerinde kullanmak istediğimiz işletim sistemini bir SD Card'a yüklememiz ve ardından SD Card'ı Raspberry Pi'ye takmamız gerekir.

## 2. Raspberry Pi'nin Python ile kullanılması ;

### 2.1 Python Nedir ?

Python nesne yönelimli programlama, fonksiyonel ya da yapısal programlama gibi birden fazla programlama paradigmasını destekleyen bir programlama dilidir.

#### 2.1.1 Python Öğrenilmesi ve Python'a Giriş :

Python'ın [www.python.org](http://www.python.org) sitesinden indirilip bilgisayara kurulmasının ardından, kullanıcı açısından daha kolay olacak bir düzenleme ve derleme programı seçmeliyiz. Bu hususta benim seçtiğim uygulama Pycharm oldu.

#### 2.1.2 Sayılar, Variables ve Stringler :

Python'da diğer dillerde ( C, C++, C#) olduğu gibi bir variable oluşturulacağı zaman variable türü yazmak gerekmez. Örn. C'de `int x = 5` derken Python'da sadece `x = 5` yazmak yeterlidir. Diğer dillerde olduğu gibi sayılar ve variable'lar kullanılarak 4 işlem gerçekleştirilebilir. Bu işlemler için öncelik sırası bildiğimiz matematikte olduğu gibi öncelikli olarak parantez içi, ardından çarpma ve bölme en son sırada da toplama ve çıkartma işlemleridir. Bunun haricinde her hangi bir ayrı kütüphane gerekmeksizin mod işlemi (%) ve üst alma işlemi (\*\*) yapılabilir.

String oluşturmak için Python da `""` veya `''` işaretleri arasına yazılır. Örn:  
"Bu Python'da bir string" veya `'0123456789'`

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 1.2. Hafıza Birimi

PLC programında kullanılabilecek 3 hafıza alanı bulunuyor. Bunlar aşağıdaki gibidir :

- Giriş Hafızası : Adres atamasında I operatörü kullanılır.
- Çıkış Hafızası : Adres atamasında Q operatörü kullanılır.
- Genel Hafıza : Adres atamasında M operatörü kullanılır.
- Hafızanın en küçük birimi bit olarak ifade edilir.
- 8 bitlik bir alan bayt olarak ifade edilir.
- 2 baytlık bir alan word olarak ifade edilir.
- 8 baytlık bir alan dword olarak ifade edilir.

## 1.3. Kontaklar ve Atamalar

PLC programı yazılırken hafızadaki birimleri kullanabilmek için kontaklar kullanılır. Kontakların birbirine bağlanmasıyla kontrol edilecek sisteme göre sayısal devre kurulur. Bu kontaklar bool tipindedir. Bir bloğun enerjili olmasına yüksek, olmamasına düşük denir.

Kontrol edilecek sisteme göre kullanılabilecek kontak türleri aşağıdaki gibidir:

- **Normalde açık kontak** : Yüksek olduğunda kapalı, düşük olduğunda açık kontaklıdır.
- **Normalde kapalı kontak** : Düşük olduğunda açık, yüksek olduğunda kapalı kontaklıdır.
- **Yukarı kenar algılayan kontak** : Yüksek olduğu an kapalı kontaklıdır.
- **Aşağı kenar algılayan kontak** : Düşük olduğu an kapalı kontaklıdır.

Bool bir atama yapılacağı zaman atama bloğu kullanılır. Kullanılabilecek atama blokları:

- **Pozitif atama** : Yüksek olduğunda yüksek, Düşük olduğunda düşük atar.
- **Negatif atama** : Düşük olduğunda yüksek, Yüksek olduğunda düşük atar.
- **Set atama** : Yüksek olduğunda durumu değişse bile hep yüksek atar.
- **Reset atama** : Yüksek olduğunda durumu değişse bile hep düşük atar.

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.1.3 List Oluşturma :

Pythonda list oluşturmak için gireceğimiz değerleri köşeli parantez [] içerisine yazarız. Örn.  $x = [10, 12, 14, 16, 18]$

x listinin belirli bir sırasında ki elemana ulaşmak için x ifadesinden sonra tekrar [] konularak o elemana ulaşılır. Örn.  $x[1] = 12$

Oluşturulan bir list başka bir liste eklenebilir. Örn.  $x = [10, 12, 14, 16, 18]$ ,  
 $y = [20, 22, 24]$ ,  $x + y = [10, 12, 14, 16, 18, 20, 22, 24]$

Append fonksiyonu oluşturulan listenin son basamağına yeni bir sayı eklememizi sağlar. Örn.  $x.append(100) = [10, 12, 14, 16, 18, 100]$

Listenin belirli bir kısmını görebilmek için variable isminden sonra [başlangıç noktası: bitiş noktası] eklenir. Örn.  $x[1:3] = [12, 14]$  Aynı şekilde seçilen bu değerler değiştirilebilir. Örn.  $x[1:3] = [0, 0]$   $x = [10, 0, 0, 16, 18]$

### 2.1.4 If elif else :

If elif ve else beyanları bir çok yazılım dilinde olduğu gibi Pythonda da belirli bir şarta göre kodun hareket etmesini sağlamaya yarar. Bu beyanları kullanmak için ilk önce if beyanını yazarız. Örn.

```
x = 20
if x < 21 :
```

Buradan sonra if beyanının altına gelecek kod satırı x 'in 20'den küçük olması durumunda işleme girecektir. Aksi halde o kod satırı atlanılacaktır.

Farklı bir kullanım şeklide şöyledir :

```
name = "Yasin"
if name is "Yasin":
```

elif beyanı ise if ile yapılan beyanın sonuna gelir ve farklı bir şart koşturmak için kullanılır ve zincirlenebilir. Örn.

```
name = "Yasin"
if name is "Yasin":
    #kod satırı
elif name is "Cosgun":
    #kod satırı
elif name is "Yildiz":
    #kod satırı
```

else beyanı ise if veya elif beyanlarından sonra gelir ve önceki beyanlardan hiç birisi gerçekleşmemiş ise gerçekleşir. Örn.

```
name = "Yasin"
if name is "Yasin":
    #kod satırı
else:
    #kod satırı
```

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 2. ZedGraph Nedir

ZedGraph, .NET için hazırlanmış ücretsiz ve açık kaynak kodlu grafik kontrolüdür. 2 boyutlu grafik (çizgi, sütun, pasta grafikleri) çizmek için kullanılır. Grafiği tamamen ve ayrıntılı şekilde düzenlemeyi sağlasa da çoğu ayarlarında varsayılanı olduğu için kullanımı kolaydır.

### 2.1. ZedGraph Kütüphanesinin Uygulamaya Eklenmesi

Windows Forms projesine *ZedGraph.dll* başvuru dosyaları eklendikten sonra araç kutusuna da eklenir ve *ZedGraphControl* aracı forma sürüklenerek bu kütüphane projeye dahil edilmiş olur.

Grafik alanıyla ilgili işlemler için *ZedGraphControl.GraphPane* sınıfı kullanılır. Bu alana ait bazı üye sınıflar şunlardır:

- **Title:** Grafik başlığı ve ilgili parametrelerini tutar.
- **XAxis:** Grafiğin X eksenine ilgili bilgilerini tutar.
- **YAxis:** Grafiğin Y eksenine ilgili bilgilerini tutar.
- **IsEnableHZoom <bool>** : X ekseninde zoom yapmayı aktif eder.
- **IsEnableHPan <bool>** : X ekseninde fareyle alan seçerek zoom yapmayı aktif eder.
- **IsShowHScrollBar <bool>** : X ekseninde hareketi sağlayan barı aktif eder.

### 2.2. ZedGraph Uygulaması – Grafiği Oluşturmak

Aşağıdaki kod parçasıyla projeye sürüklenerek daha önce eklenmiş harita kontrolünü düzenlemek mümkündür:

```
zgc_grafik.GraphPane.Title.Text = "Kontrol ve Otomasyon Mühendisliği";  
zgc_grafik.GraphPane.XAxis.Title.Text = "Zaman";  
zgc_grafik.GraphPane.YAxis.Title.Text = "AGNO";  
double[] xekseni = { 1, 2, 3, 4, 5 };  
double[] yekseni = { 1, 4, 9, 16, 25 };
```

Bu kod parçası derlendikten sonra grafiğin düzenlenmiş başlıklarıyla beraber forma eklendiği görünecektir fakat bu grafiğe herhangi bir veri eklenmediği için grafik ekranı boş görünmektedir.

Grafiğe veri eklemek için *GraphPane* sınıfından *AddCurve* methodu kullanılır. Bu method bu veriyi temsil eden *LineItem* nesnesi döndürür. İsteğe bağlı olarak bu nesnenin referansı saklanabilir. Bu method için aşağıdaki parametreler kullanılır.

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 2. TIA Portal Nedir ?

TIA Portal Siemens'in kullanıcılara sunmuş olduğu PLC'yi bilgisayara tanıtmak, PLC programları yazmak, programları atmak ve atılmış programların canlı bir şekilde incelenmesine olanak sağlayan bir uygulamadır.

### 2.1 TIA Portal Nasıl Kullanılır?

TIA Portal'ı PLC ye tanıtabilmemiz için ilk önce yeni proje açtığımız zaman doğru PLC model ve sürümünü seçmemiz gerekmektedir. Bunun yanı sıra PLC için bir IP adresi atamalı ve bilgisayara da IP adresi elle girilmelidir. Bunun ardından donanımda analog giriş varsa analog giriş PLC'ye eklenip Proje Görünüm ekranına geçilebilir.

Proje Görünüm ekranında sol menüde PLC adının alt menüsünde bulunan Program Blocks menüsünde ki Main[OB1] ekranını açarsak alt alta satırlar ve networklerden oluşturulabilen bir akış şeması görürüz. Bu akış şemasında istenilen yerlerde Normalde Açık giriş, Normalde Kapalı giriş, çıkış gibi elemanları kullanarak devre oluşturabiliriz ve bu devreyi PLC ye atayabiliriz.

### 2.2 TIA Portalda devre oluşturmak:

Bir, normalde açık giriş veya normalde kapalı giriş elemanı oluşturmak için normalde açık veya normalde kapalı giriş sembollerinden birini sürükleyip akış şemasına bırakabilir veya akış şemasında girişi koymak istediğimiz yeri seçip ardından giriş sembolüne tıklayarak girişi akış şemasına taşıyabiliriz. Giriş elemanı hafızada bir bitlik yer tutar ve bu yeri programda biz atayabiliriz. Örn. I0.0. Atadığımız girişe PLC TAGS menüsü altındaki Show all tags kısmından bir isim atayabilir ve bundan sonra o girişi kullanacağımız zaman ismi ile çağırabiliriz.

Normalde açık girişin(I0.0) ucuna bir Q0.0 çıkışı bağlar ve bu programı PLC ye atarsak PLC de I0.0 girişini kullanarak Q0.0 çıkışındaki LED'in yakılıp söndürüldüğünü görebiliriz. Ve Q0.0 gibi bir çıkış birimini logic giriş gibi kullanabilir ve I0.0 girişine paralel Q0.0 normalde açığını bağlayarak mühürleme işlemini gerçekleştirebiliriz.

Haftanın çözülecek problemi: 5 butondan sadece 3 ü basılı olduğunda Q0.0, sadece 4'ü basılı olduğunda Q0.1 çıkışları enerjilensin.

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.1.5 For Loop ve while loop:

Python'da döngü kurabilmek bir işlemi birden fazla kez yapabilmek için kullanılan yöntemlerden birisi for döngüsüdür.

Örn.

Çıkış:

```
isimler = ["Ahmet", "Mehmet", "Ayşe", "Fatma"]
```

Ahmet

Mehmet

```
for f in isimler:
```

Ayşe

```
    print(f)
```

Fatma

For döngüsü belirli bir sayıda veya bir aralıkta gerçekleşecek ise bu aralığı belirlemek için range() fonksiyonunu kullanırız.

Örn.

Çıkış:

```
For x is range(3):
```

Python

```
    print("Python")
```

Python

Python

range() fonksiyonu range(a, b) şeklinde yazılarak a'dan b'ye sayacak şekilde de ayarlanabilir. Range(a, b, c) şeklinde yazarsak a'dan b'ye c kadar atlayarak sayma işlemi gerçekleştirir.

While döngüsü for döngüsü gibi döngü kurabilmek bir işlemi birden fazla kez yapabilmek için kullanılan yöntemlerden bir diğeridir. While döngüsü sonsuza loop a girebilir. Bunun için döngüden çıkartacak bir tanımlayıcıya ihtiyaç duyar.

Örn.

Çıkış:

```
x = 0
```

```
While x < 3
```

0

```
    print(x)
```

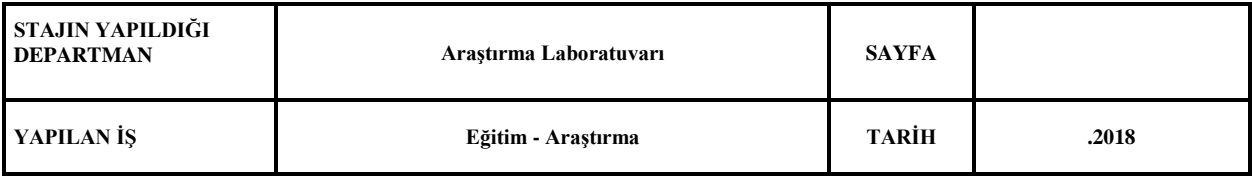
1

```
    x += 1
```

2

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--





Form No: FR-1170; Revizyon Tarihi: 16.11.2016; Revizyon No:00



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.1.6 Break ve Continue :

Break ve continue komutları döngüler için sıklıkla kullanılan istenildiği zaman döngüyü durduran (break) veya döngüyü bir sonraki adımdan devam ettiren (continue) komutlardır.

### 2.1.7 Fonksiyonlar :

Pythonda hazır bulunan fonksiyonlar(print(),range(),len())... vs.) gibi kendi fonksiyonumuzu hazırlayabiliriz. Birden fazla kez kullanılacak bir işlemi, örneğin verilen iki sayının en büyük ortak bölenini almak gibi bir kez fonksiyon şeklinde yazarak programın geri kalanında onu tek satırlık bir kod olarak çağırabiliriz.

Örn.

```
def USDtoTRY(usd):  
    try = usd*4.88  
    print(try)
```

Bir fonksiyondan elde edilecek değeri kullanmak istiyorsak return komutunu kullanabiliriz.  
Örn.

```
def USDtoTRY(usd):  
    try = usd*4.88  
    return try  
Para = USDtoTRY(5) # Para artık 5*4.88 değerini alacak.
```

Fonksiyona verilecek parametreler önceden default olacak şekilde ayarlanabilir ve çağırılabilir. Bir fonksiyon içerisinde oluşturulan variable global sayılmaz ve başka bir fonksiyonda kullanılamaz.

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 3. Seriport üzerinden gelen bilgiyi okuma:

#### 3.1 Arduino'da sanal enlem ve boylam oluşturacak programın yazılması:

Enlem ve boylam ondalık sayılar olduğu için enlem ve boylamı double olan, başlangıçta enlem ve boylama bir değer atayıp ardından o değerleri her bir loopda 0.00001 arttıracak Arduino programı :

*union*

```
{
    double db;
    byte by[4];
} enlem;
union
{
    double db;
    byte by[4];
} boylam;
void setup()
{
    pinMode(A0,INPUT);
    Serial.begin(9600);
    enlem.db = 42.15452316123456789;
    boylam.db = 28.15452316123456789;
}
void loop()
{
    enlem.db = enlem.db + 0.00001;
    boylam.db = boylam.db + 0.00001;
    Serial.write(enlem.by[0]);
    Serial.write(enlem.by[1]);
    Serial.write(enlem.by[2]);
    Serial.write(enlem.by[3]);
    Serial.write(boylam.by[0]);
    Serial.write(boylam.by[1]);
    Serial.write(boylam.by[2]);
    Serial.write(boylam.by[3]);
    delay(5);
}
```

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 2.3 Timerlar :

### 2.3.1 TON (On Delay Timer):

Giriş enerjilendikten sonra Preset Time'da yazılan süre kadar bekleyip ardından enerjiyi çıkışa taşıyan zamanlayıcı.

### 2.3.2 TOF (Off Delay Timer):

Giriş enerjilenir enerjilenmez, çıkışa enerji taşıyan ve giriş enerjili olduğu müddetçe bunu sürdürüp girişten enerji kesilince önceden ayarlanan Preset Time süresince çıkışa enerji sağlamaya devam eden ve Preset Time dolunca çıkıştan enerjiyi kesen zamanlayıcı.

### 2.3.3 TP (Pulse Timer):

Giriş enerjilendikten sonra girişin enerjili olup olmamasına bakmaksızın çıkışa belirli süre enerji gönderen süre bitince enerjiyi kesen zamanlayıcı.

### 2.3.4 TONR (On Delay Timer With Reset):

Sadece girişe enerji verildiği müddetçe Elapsed Time sayan, giriş enerjisi kesilince Elapsed Time'ı olduğu yerde tutan ve Elapsed Time daha önceden ayarlanmış Preset Time a gelince çıkışa enerji veren üzerinde Elapsed Time için Reset girişi bulunan zamanlayıcı.

Haftanın Çözülecek Problemi :

10.0 gibi bir giriş ilk enerjilendiğinde çıkışı enerjilendirecek ikinci enerjilendiğinde çıkışın enerjisini kesecek bu sayede tek bir butondan start/stop yapılabilecek.

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.1.8 Classlar ve Objeler:

En basit şekilde Classlar, belirli bir grup variable ve fonksiyonu bir araya getirdiğimiz yerdir. Python'da class;

*class Enemy:*

şeklinde oluşturulur. Class altında class'a ait fonksiyonlar ve variable'lar olabilir.

Örn.

*Class Enemy:*

*Life = 3*

*def attack(self):*

*print('Attacked')*

*self.life -= 1*

*def checkLife(self):*

*if self.life <= 0:*

*print("Enemy is dead")*

*else:*

*print(str(self.life) + "Life left")*

*# Class içerisinde bir fonksiyonu gerçekleştirebilmemiz için o class'a ait bir obje oluşturmamız gerekir.*

*enemy1 = Enemy()*

*enemy2 = Enemy()*

*enemy1.attack()*

*enemy1.attack()*

*enemy1.checkLife() # Birinci objede yapılan değişiklikler bir başka objeyi etkilemez.*

*enemy2.checkLife()*

Classlar içerisinde bir obje oluşturulur oluşturulmaz gerçekleşmesini istediğimiz komutlar olur. Bu işlem için `__init__` fonksiyonunu kullanılır. Initialize kelimesinden gelir. Class içerisinde normal bir fonksiyon yazılır gibi yazılır. Sadece fonksiyon isminin başında ve sonunda ikişer adet alt çizgi bulunur.

Örn.

*Class Enemy*

*def \_\_init\_\_(self):*

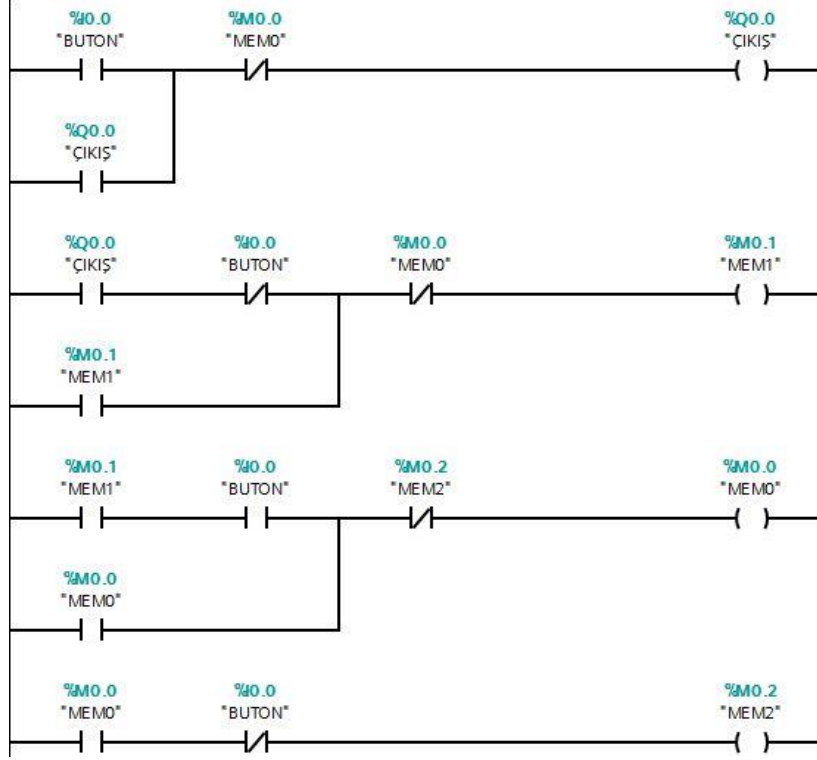
*print("You saw an enemy")*

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### Haftanın Problemi Çözüldü:



1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.1.8 Classlar ve Inheritance:

Classlarda inheritance kalıtım-mirası için kullanılır. Bir Class başka bir Classın kalıtımı-mirası olabilir. Bu koşullarda bir Classdaki variable ve fonksiyonlara genetikte bulunan bir diğer class ulaşabilir.

Örn.

```
class isim():  
    def print_isim(self):  
        print("Yasin")  
class soyisim(isim):  
    def print_soyisim(self):  
        print("Cosgun")
```

```
x = soyisim()  
x.print_isim() # soyisim classına ait obje isim classının fonksiyonunu kullanabilir.  
x.print_soyisim()
```

Bir classın birden fazla Inheritance'ı ve inheritance olan bir class'ında kendi inheritance'ı olabilir.

Örn.

```
class Mario():  
    def move(self):  
        print("Mario is moving")
```

```
class Shroom():  
    def eat_shroom(self):  
        print("Mario is big")
```

```
class BigMario(Mario,Shroom):  
    pass
```

```
bm = BigMario()  
bm.move() # Mario classının fonksiyonu kullanılabilir.  
bm.eat_shroom() # Shroom clasının fonksiyonu kullanılabilir.
```

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 3.2 Seriport Verisini Okuyan Programın Yazılımı:

Bu konuda ilk önce yapılması gereken okumanın hangi türde olacağını kararlaştırmak. C# üzerinden yapılabilecek iki tür method şekli vardır. Senkron ve asenkron method

#### 3.2.1 Asenkron Method:

Asenkron programlama, işin parçalara ayrılıp tüm işlemlerin aynı anda sürdürülmesini sağlar. Asenkron Programlama ile programımız içerisinde yazdığımız bir kod işletilirken, aynı program içerisinde diğer kodlar da işletilebilir. Bu sayede kullanıcı programımızın bir bölümünü kullanırken, başka bir bölümü ile de işlem yapabilir. Asenkron programlama, multi threading ile karıştırılmamalıdır. Asenkron olarak yazacağımız kodlar tek bir thread üzerinde de çalışabilir. Asenkron programlamanın özelliği farklı thread'lerde çalışması değil, verdiğimiz işi parçalara ayırarak programımız üzerinde birden fazla işin aynı anda yürütülmesini sağlamaktır.

#### 3.2.1 Senkron Method:

Senkron method asenkron methoddan farklı olarak bir işlemi gerçekleştirirken bulunduğu thread'i işgal etmeye devam eder ve o işlem bitmeden bir başka işlemi gerçekleştirmez.

#### 3.2.2 Asenkron vs Senkron Method farkı:

Senkron methodla yazılan: *Thread.Sleep(1000);* kodu 1 saniyede işlenir ve bu süre zarfında o threaddeki kod parçacığı başka bir iş gerçekleştiremez. Asenkron methodla yazılan: *await Task.Delay(1000);* kodu gene 1 saniyede işlenir ancak bu 1 saniye içerisinde program farklı işleri gerçekleştirebilir. İşlemin bitmesini beklemez.

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--





STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 2.4 Sayıcılar(Counter) :

### 2.4.1 Yukarı Sayıcı(CTU):

Üzerinde yukarı sayma girişi ve preset girişi olmak üzere ihtiyacımız olan iki girişi bulunur. Başlangıçta 0 değerinden başlayıp daha önceden belirtilmiş preset değere kadar sayan ve set edilen değere gelince enerji çıkışı sağlayan araçtır. Manuel reset edilmez ise sayaç preset değerinin üzerinde saymaya devam eder.

### 2.4.2 Aşağı Sayıcı(CTD):

Üzerinde aşağı sayma girişi ve preset girişi olmak üzere ihtiyacımız olan iki girişi bulunur. Başlangıçta daha önceden belirtilmiş preset değerinden başlar ve 0'a doğru geri sayar. 0' a ulaştığı zaman enerji çıkışı sağlar ve Reset edilmediği takdirde 0'dan sonra aşağı saymaya devam eder.

### 2.4.3 Yukarı Aşağı Sayıcı(CTUD):

Üzerinde diğer iki sayaçtan farklı olarak ihtiyacımız olan 3 adet giriş mevcuttur. Yukarı saydırma, aşağı saydırma ve preset girişleri bulunmaktadır . Bu sayıcı hem yukarı hemde aşağı doğru sayabilir.

Haftanın Problemi: 2-) Sadece TON (On Delay Timer) kullanılarak TOF(Off Delay Timer),TP(Pulse Timer),TONR(On Delay Timer with Reset) gerçekleştirilecek.

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 2.2 Raspberry Pi'nin Hazırlanması:

### 2.2.1 İşletim Sisteminin Yüklenmesi:

Raspberry Pi'ye istediğimiz işletim sistemini yüklemek için <https://www.raspberrypi.org/downloads> sitesinden Noobs işletim sistemini seçip bilgisayarımıza indiriyoruz. Ardından SD Formatter 4.0 ile SD kartımıza format atıyoruz. Formatlanan SD kartımıza indirdiğimiz Noobs işletim sisteminin image dosyasını kopyalıyoruz. SD kartını bilgisayardan çıkartıp Raspberry pi'ye takabiliriz.

### 2.2.2 İşletim Sisteminin Ayarlarının Yapılması:

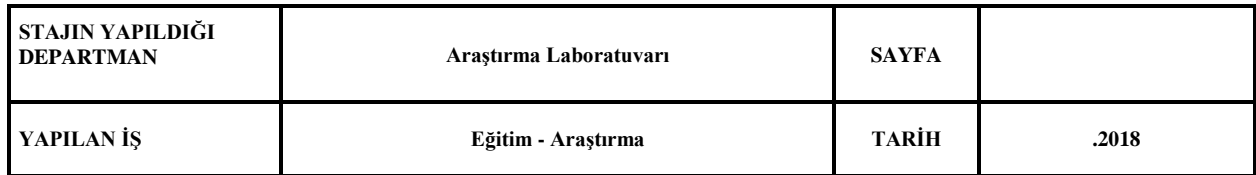
Raspberry Pi ayarlarının yapılandırılması ve istenilen uygun programları indirebilmek için bir internet bağlantısına ihtiyaç duyulur. İnternet bağlantısı; kart üzerinde ki Ethernet portundan kablo ile veya daha yeni modellerde bulunan Wifi ile yapılabilir. Eski modellerde Wifi ile bağlantı yapılmak isteniyorsa Wifi dongle kullanılabilir.

Ctrl+alt+t tuş kombinasyonu ile İşletim sistemi içerisinde kod terminalini açıp `$ sudo apt-get update` koduyla mevcut sistem için var olan güncellemeleri bulabilir ve `$ sudo apt-get upgrade` koduyla güncellemeleri gerçekleştirebiliriz.

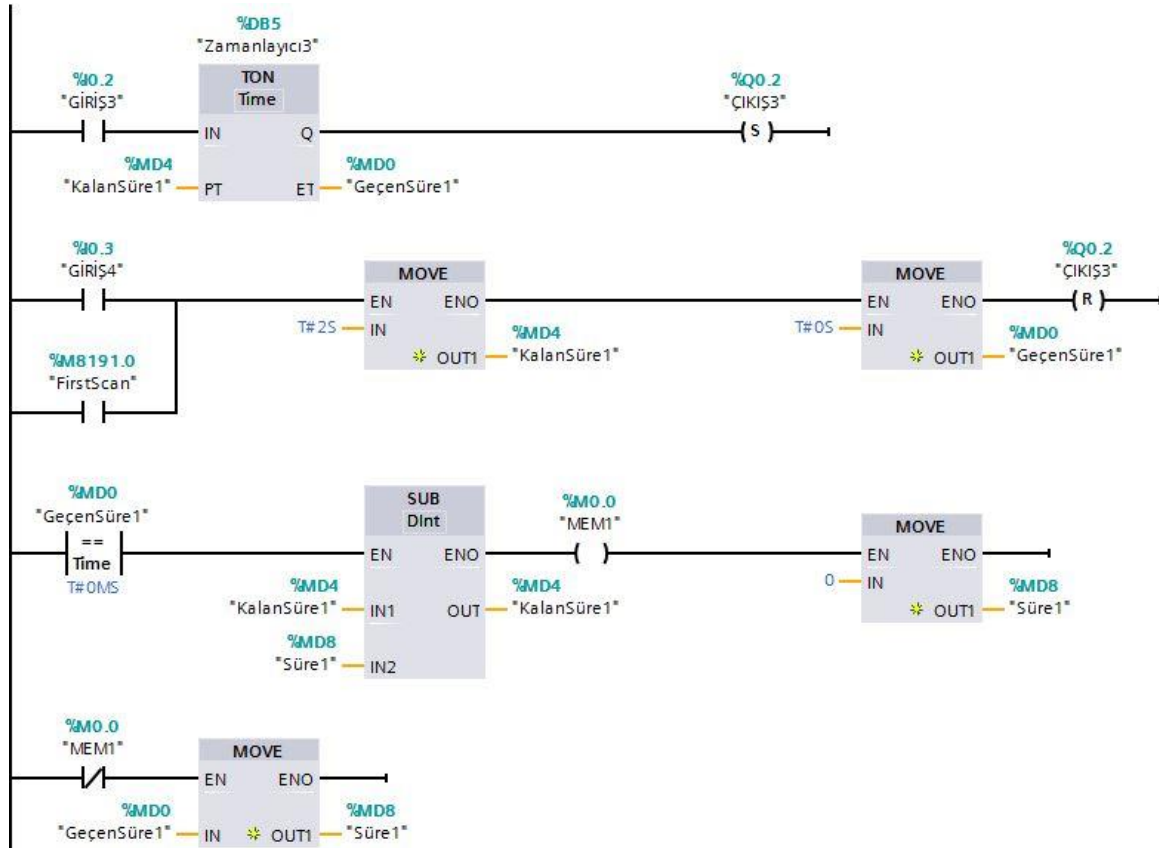
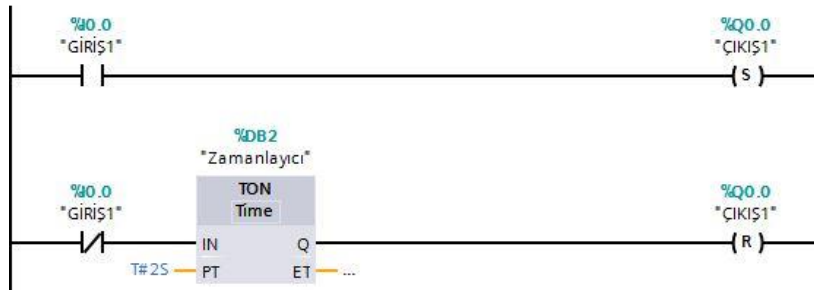
İşletim sisteminde yapılabilecek diğer ayarlar:

```
$ sudo apt-get dist-upgrade
$ sudo apt-get wolfram
$ sudo apt-get purge wolfram
$ sudo apt-get autoremove
$ sudo raspi-config
```

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



**TOF:**



1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.2.3 Raspberry Pi'nin Remote Control uygulamalarıyla kullanılması:

Raspberry Pi cihazını bir remote control uygulamasına bağlayarak bilgisayarımız üzerinden Raspberry Pi cihazına erişebiliriz.

Kod terminalinde `$ sudo raspi-config` yazıp entera basarak raspberry pi ayarlar kısmına ulaşabiliriz. Buradan Interfacing Options sekmesine girerek P2-SSH modülünü Enable duruma getirmemiz gerekmektedir. Ardından Bilgisayarımıza SSH destekli bir remote control uygulaması indirmemiz gerekmektedir. Bilgisayarımızda kurulum işlemi tamamlandıktan sonra uygulamaya sizden istenilen Raspberry Pi adresini girerek bağlanabilirsiniz. Bağlantı öncesi login as : pi Ve password : raspberry default olarak gelen ayarlardır.

Bağlantı problemi yaşanırsa `$ sudo apt-get install tightvncserver` ve `$ sudo apt-get install xrdp` kodları kullanılabilir

### 2.2 Raspberry Pi Terminalinin Detaylı Kullanım Şekilleri:

`$ clear` : Terminal ekranında yazdıklarınızın hepsini temizler.

`$ find / -name ornek.txt` : Sisteminizde ornek.txt isimli dosyayı arar ve bu isimli dosyayı içeren \$ klasörleri listeler.

`$ nano ornek.txt` : Linux Text Editör'de ornek.txt isimli dosyayı açar.

`$ poweroff` : Raspberry'yi kapatır.

`$ raspi-config` : Raspberry konfigürasyon menüsünü açar.

`$ reboot` : Cihazı reboot eder.

`$ cat ornek.txt` : ornek.txt içeriğini gösterir.

`$ cd /abc/xyz` :Terminal ekranında çalıştığınız klasörü /abc/xyz klasörü haline getirir.

`$ cp /home/pi/documents/ornek.txt /home/pi/office/` : ornek.txt dosyasını bulunduğu \$ documents klasöründen office klasörüne kopyalar.

`$ ls` : Bulunduğunuz klasördeki dosyaları listeler.

`$ mkdir ornek_klasor` : "ornek\_klasor" isminde yeni bir klasör oluşturur.

`$ mv ornek.txt /home/pi/office/` : Bulunduğunuz klasördeki ornek.txt dosyayı office klasörüne taşır.

Eğer taşıyacağınız dosya bulunduğunuz klasörde değil ise yukarıda kopyalama örneğinde olduğu gibi öncesinde dosyanın bulunduğu klasörde yazabilirsiniz.

`$ rm ornek.txt` : ornek.txt dosyasını siler.

`$ rmdir ornek_klasor` : ornek\_klasor isimli klasörü siler. (Sadece içi boş ise)

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 3. Python Tkinter ile GUI oluşturma:

#### 3.1 Tkinter Nedir ?

Tkinter, Python yazılım dilinde temel grafik arayüz işlemleri için kullanılan, Python içerisinde ki standart kütüphanelerden birisidir.

#### 3.2 Tkinter ile Pencere Oluşturmak:

```
import Tkinter  
root = Tk()
```

Yukarıdaki kod parçası Tkinter ile pencere oluşturmamıza yetmektedir. Ancak pencere açılır ver açılır açılmaz kapanır. Bu işlem çok hızlı gerçekleştiği için kullanıcı pencereyi farkedemez. Pencerenin ekranda kalıcı olmasını sağlamak için bir loop oluşturmamız gerekir.

```
import Tkinter  
root = Tk()  
root.mainloop()
```

İşimizi görecektir.

Pencerede en basitinden buton oluşturabilmek için Button( ) fonksiyonunu kullanırız. Fonksiyon parantezi içerisinde ilk girilen bilgi butonun pencerenin neresinde olacağını belirten index bilgisidir. İkinci girilen bilgi text = ile birlikte yazılır ve buton üzerinde ki text'i belirtmemizi sağlar. Üçüncü girilen bilgi fg = ile girilir ve buton üzerinde ki text rengini belirtmemizi sağlar.

ÖRN.

```
Button1 = Button(topFrame, text = "Button1 ", fg = "red")  
Button2 = Button(bottomFrame, text = "Button2 ", fg = "green")
```

Örnekte ki kodda Button1 pencerenin üst kısmında bulunur, üzerinde kırmızı renkte Button1 yazar. Button2 ise pencerenin alt kısmında bulunur ve üzerinde yeşil renkte Button2 yazar.

Butonları pencere üzerinde görüntülemek için pack() fonksiyonunu kullanırız.

ÖRN.

```
Button1.pack()  
Button2.pack()
```

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 2.5 Analog Giriş ve Çıkış

### 2.4.1 Analog Giriş ve Çıkış Nedir? Dijital Giriş ve Çıkıştan Farkları Nelerdir?

Dijital giriş ve çıkış 1 ve 0 olarak tanımlanan, enerjinin olması veya olmaması durumudur. Analog giriş ve çıkış ise, belirli bir uygulanan gerilim veya akımın, belirli bir aralıkta oransal olabilmesidir. Örneğin 0-10VDC demek, 0 ile 10VDC arasında gerilim uygulamak demektir. Plc'lerde giriş ve çıkış portları vardır, bunlar Plc nin modeline ve bizim ihtiyacımıza göre çeşitli sayılarda bulunabilirler. Plc'lerin üzerinde doğrudan bulunan giriş ve çıkışlar dijitaldir. Analog kontrol sağlamak için Plc'lere Analog modül bağlanır. Birlikte olan modelleride mevcuttur ama genelde analog modül haricen kullanılır. Örneğin PLC'nin giriş tarafındaki portundan birine + -24VDC uygulanırsa, ilgili numaralı girişi enerjilendirmiş ve 1 yapmış olunur, veya aktif. Enerji kesilirse 0 yapmış olunur ve ilgili giriş açık devre olur. Bu girişlere uygulanan gerilimlerle, Plc'nin hafızasına programladığınız mantık algoritması dahilinde de, istenilen çıkışlar aktif etmiş olunur.

Analogta farklı olarak gerilim veya akım belirli bir aralık içerisinde olabileceğinden, tam açık tam kapalı değilse, ara konumlarda da kontrol sağlanabilir. Örneğin 0-10V DC çıkışlı bir basınç sensörünün çıkışını bir PLC nin analog girişine bağlayarak, sensörden gelen değerleri alıp, yorumlanabilir ve ilgili kontrol cihazına yine analog olarak geri besleme yapılabilir. Burada PLC nin hafızasına yazılan programlar doğrultusunda enerjilenen girişlere göre çıkış almanız sağlanır.

Haftanın Çözülecek Problemleri:

1-) Bir fabrika bantında 5 adet ürün sayan 5 adet ürün geçtikten sonra bandı durduran ve 5 adet ürünün alındığı işareti gelene kadar enerjilendirmeyen devreyi oluşturunuz.

2-) Analog girişte alınan veriyle çıkışın, giriş 5.1 V'un üzerinde iken enerjilenmesini 4.9 V'un altında iken enerjisini kaybetmesini sağlayan devreyi oluşturunuz.

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.3 Raspberry Pi ve Python:

Raspberry Pi GPIO portlarını python ile kullanabilmek için 1k direnç ve 1 led ile ledi raspberry pi üzerinden yakıp söndürmeyi deneyeceğiz. Bunun için raspberry pi üzerinde ki GPIO portlarından Ground(GND) olanlardan birini boardun en sol dikey kısmına yerleştirip yerleştirdiğimiz yerin altına 1k direncimizi yerleştireceğiz. Direncimizin bir ayağı GND bağlanmış olacak ve diğer ayağını boardun ortasında bir yere yerleştirip, yerleştirdiğimiz yerin yatay iletken hizasına LED'in eksi bacağını bağlayacağız. Ledin artı bacağını bağladığımız yere de GPIO pinlerinden uygun olanlardan kullanmak istediğimiz yere bağlayacağız.

Raspberry Pi komut terminalini açıp GPIO pin kütüphanesini indirmemiz gerek.Bu işlem bir kereliğe mahsus olup her seferinde yapmamız gerekmez.

```
$ cd Desktop/  
$ sudo apt-get install python-rpi.gpio
```

İşlemler tamamlandığı zaman python için gerekli GPIO kütüphanesinin indirilme işlemi tamamlanmış olacak. Bundan sonra python kodunu yazacağımız belgenin oluşturulması gerek.

```
$ nano ledexample.py
```

Enter'a bastığımız anda .py uzantılı klasörümüzün içerisine girmiş bulunmaktayız.

```
import Rpi.GPIO as GPIO  
import time
```

```
GPIO.setmode(GPIO.BCM)  
GPIO.setup(18, GPIO.OUT)  
GPIO.output(18 ,GPIO.HIGH)  
Time.sleep(5)  
GPIO.output(18, GPIO.LOW)  
GPIO.cleanup()
```

Bulunduğumuz klasörden çıkarken kaydedip çıkıyoruz. Tekrar komut terminaline geldiğimiz zaman

```
$ python ledexample.py
```

Kodunu girerek yazdığımız kodu çalıştırıyoruz ve LED'imiz 5 saniye yandıktan sonra sönüyor.

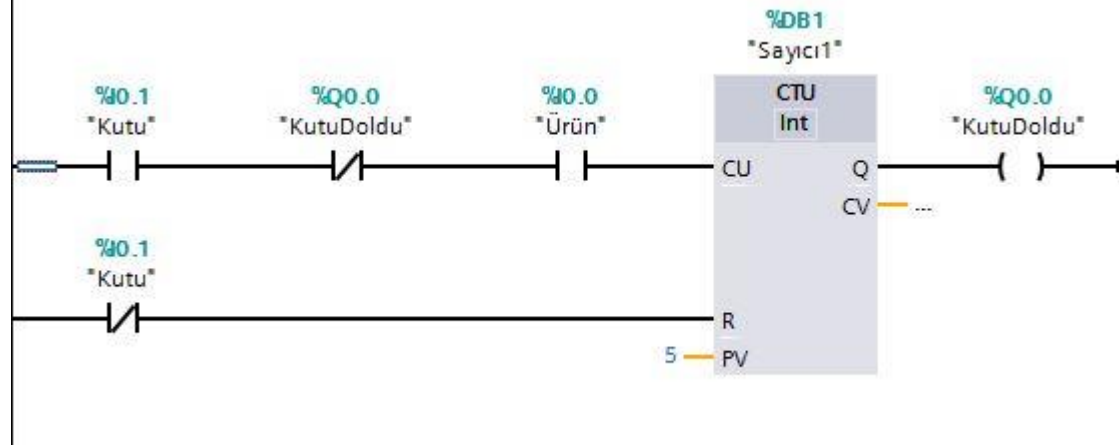
1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



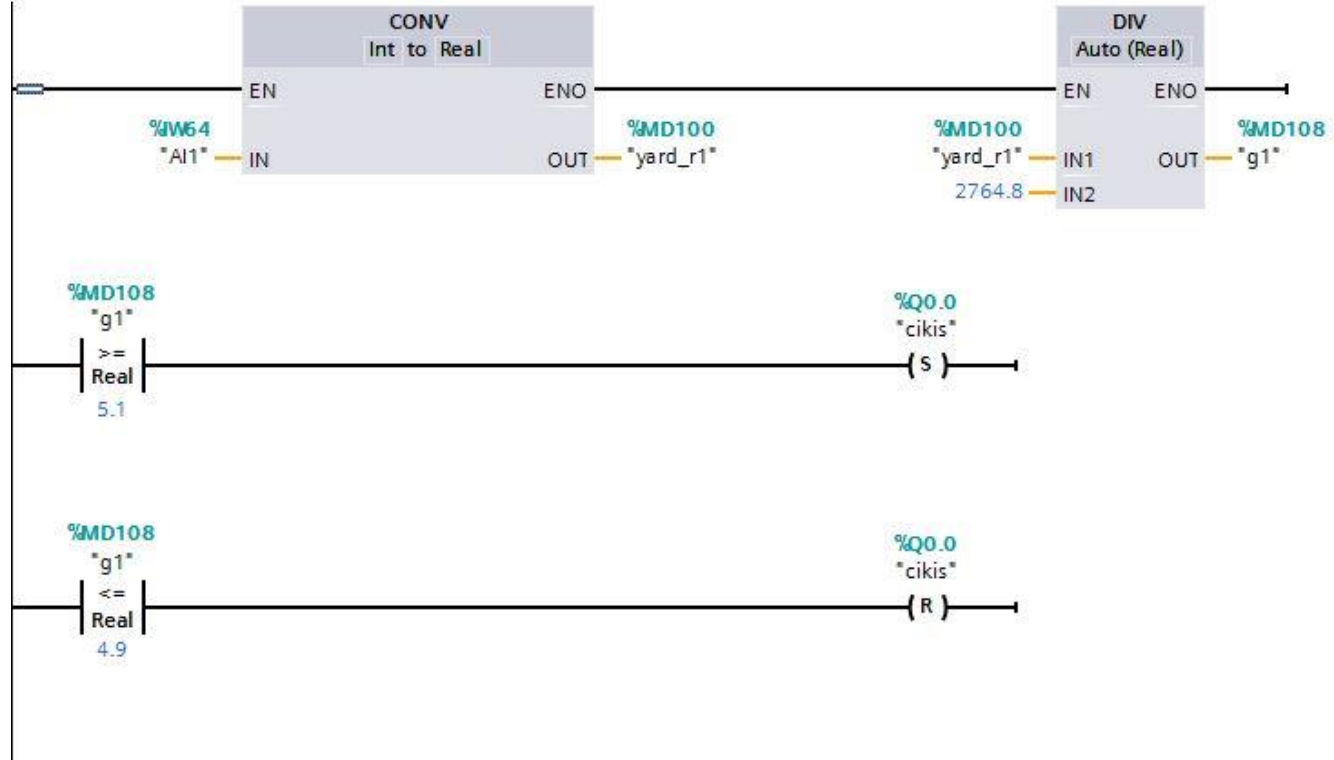
STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

Haftanın Problemleri Çözümleri:

1-)



2-)



1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--





STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 2.3.1 Raspberry Pi ve Python Kullanarak Uzaklık Sensörü çalıştırmak:

Raspberry pi komut terminalinde masaüstüne ulaşıyoruz.

```
$ cd Desktop/
```

Kodu yazacağımız dosyayı oluşturuyoruz.

```
$ nano distancesensor.py
```

Oluşturduğumuz dosyada kodumuzu yazmaya başlıyoruz:

```
import Rpi.GPIO as GPIO
```

```
import time
```

```
GPIO.setmode(GPIO.BCM)
```

```
TRIG = 4
```

```
ECHO = 18
```

```
GPIO.setup(TRIG, GPIO.OUT)
```

```
GPIO.setup(ECHO,GPIO,IN)
```

```
GPIO.output(TRIG, TRUE)
```

```
Time.sleep(0.0001)
```

```
GPIO.output(TRIG,False)
```

```
While GPIO.input(ECHO)== False:
```

```
    Start = time.time()
```

```
While GPIO.input(ECHO)== True:
```

```
    End = time.time()
```

```
Sig_time= end-start
```

```
Distance = sig_time/0.000058 # 0.000058 CM cinsinden mesafey görebilmek için.
```

```
Print('Distance: {} cm'.format(distance))
```

```
GPIO.cleanup
```

Kodu tamamladıktan sonra kaydedip çıkıyoruz ve çalıştırmak için komut terminaline aşağıdaki kodu giriyoruz:

```
$ python distancesensor.py
```

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### 3.2.1 Tkinter ile Widget-Label oluşturmak:

Widget oluşturmak için Label() fonksiyonuna ihtiyaç duyarız. Fonksiyonun ilk indisi hangi pencerede oluşturmak istediğimiz bilgisidir. İkinci indis text bilgisidir. Üçüncü indis arka plan rengine ait bilgisidir ve son olarak 4. indis ise text rengi bilgisidir.

ÖRN.

```
From tkinter import *
Root = Tk()
One = Label(root, text = "One", bg = "red",fg = "black")
One.pack()
Two = Label(root, text = "Two", bg = "blue",fg = "white")
Two.pack(fill = X) # fill = X, X eksenini boyunca doldur demek
Three = Label(root, text = "Three", bg = "yellow",fg = "red")
Three.pack(side = LEFT, fill = Y) # side hangi yönde olacağını söyler
fill Y eksenini boyunca doldur demek

Root.mainloop()
```

### 3.2.1 Tkinter ile Grid oluşturmak:

Grid oluşturmak için oluşturduğumuz label etiketlerini pack() fonksiyonu ile değil grid() fonksiyonu ile pencerede görünür yaparız.

ÖRN.

```
From tkinter import *

Root = Tk()

Label1 = Label(root, text = "Name")
Label2 = Label(root, text = "Password")
Entry1 = Entry(root)
Entry2 = Entry(root)
Label1.grid(row = 0)
Label2.grid(row = 1)
Entry1.grid(row = 0, column = 1)
Entry2.grid(row = 1, column = 1)

Root.mainloop()
```

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

## 2.5 Encoder Nedir?

Encoder (Sinyal Üretici); bir milin dönme ya da ötelenme hareketine karşılık, sayısal(dijital) bir elektrik sinyali (kare ya da sincos dalga) üreten elektromekanik bir cihazdır. Şaftlarının Çalışma şekillerine göre; dönel olarak çalışan Encoderlar (Rotary Encoder) ve doğrusal olarak çalışan (Linear Encoder) lar olmak üzere ikiye ayrılırlar. Encoder Şaftının, tahrik şaftına Bağlantı Şekillerine göre; Milli Tip (Shaft Encoder) ve Delik Milli Tip (Hollow Shaft Encoder) olmak üzere ikiye ayrılırlar. Sinyallerin Oluşturulmasında Kullanılan Algılama(Sensör) Teknolojisine göre; Optik Algılamalı ve Manyetik Algılamalı Tipler olmak üzere ikiye ayrılırlar. Bulundukları Gerçek Pozisyonlarının her adımda Belirlenip Belirlenememe özelliklerine göre; Artımsal (Incremental Encoder) ve Mutlak (Absolute Encoder) olmak üzere ikiye ayrılırlar.

### 2.5.1 Incremental Encoder:

Dönen mil için sürekli kare sinyal üreterek bulunduğu konumu bilgilendirmede ve sayma işlemlerinde kullanılır. Bu tip encoderların mutlak encoderlardan farkı enerjilendikleri andan itibaren bulunduğu konumu 0 olarak kabul eder; ve bu değer in üstüne işlem yapar. Bu tip encoderlar endüstriyel robotlar, Cnc Tezgahlarda, Antenler, Medikal cihazlar, Paketleme makinaları Deri işleme makinaları, şişeleme makinalarında vb alanlarda kullanılır.

### 2.5.1 Absolute Encoder:

Encoder a güç verildiği anda dönen mile göre dijital bit dizileri şeklinde birbirine benzemeyen çıkışlar vererek konumu belirler. enerji kesilse dahi en son ki konumunda kalır.Ve tekrar enerjilendiğinde kaldığı konumdan işlemine devam eder.

Haftanın Problemi: Incremental Encoder'dan gelen veriden Konum ve hız elde etmeye yarayan devre oluşturulacak.

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

**Python'da çalışan, C# Seyrüsefer yazılımında ki seri port üzerinden koordinat okuma programının yazılması :**

```
import serial
import struct
from time import sleep
ser = serial.Serial('COM8', baudrate = 9600)
enlem = bytes(4)

while True:

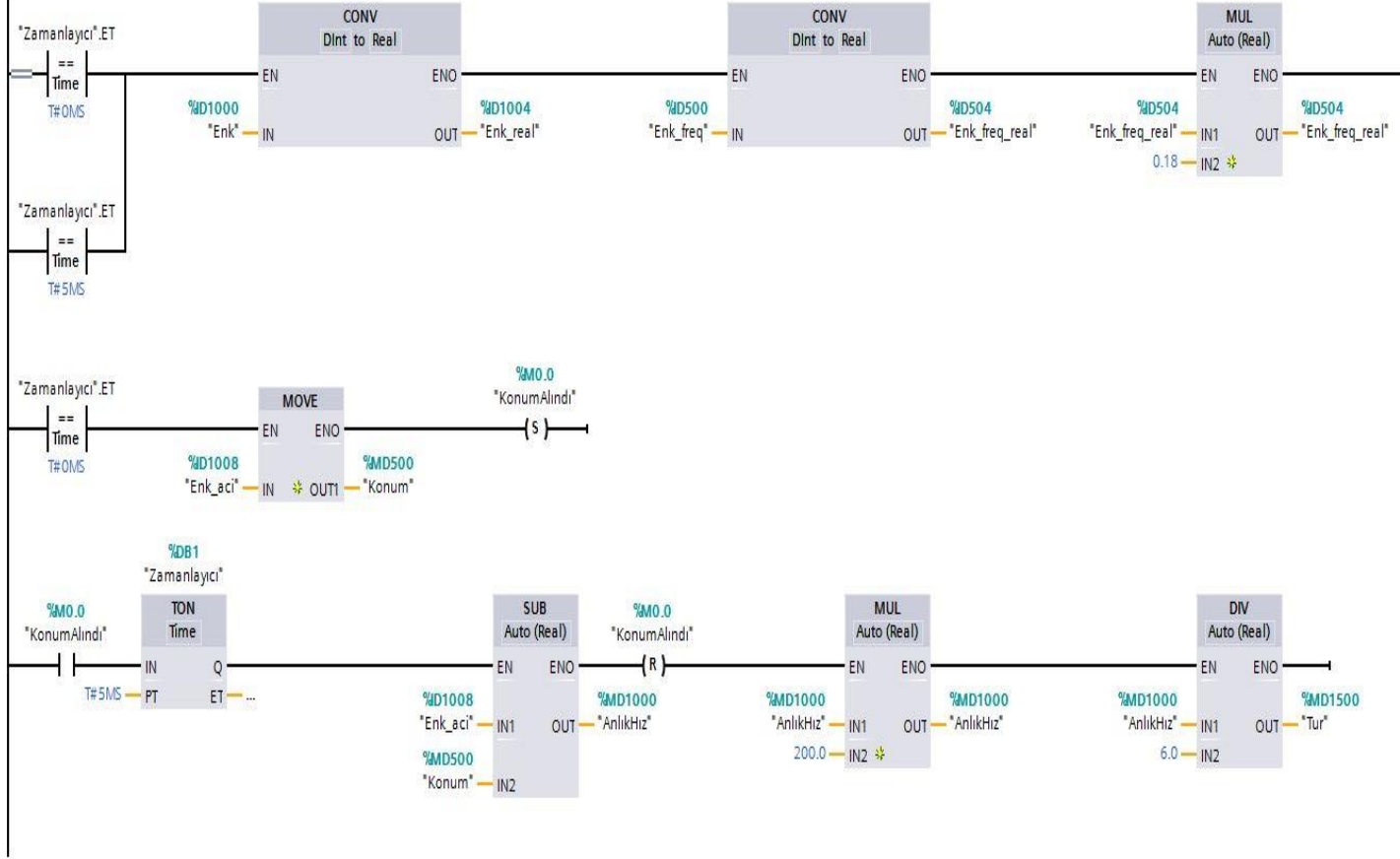
    veri = ser.read(8)
    enlemdata = veri[:4]
    boylamdata= veri[4:]
    enlem = struct.unpack('f', enlemdata)
    boylam = struct.unpack('f', boylamdata )
    print(enlem ),
    print(boylam)
```

1.1	STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
-----	--------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

### Haftanın Problemi Çözümü:



### Python -Raspberry Pi IMU okuma programı

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

```
import smbus
import math
import time
```

```
power_mgmt_1 = 0x6b # Güç yönetim register'leri
power_mgmt_2 = 0x6c
def read_byte(adr):
    return bus.read_byte_data(address, adr)
def read_word(adr):
    high = bus.read_byte_data(address, adr)
    low = bus.read_byte_data(address, adr+1)
    val = (high << 8) + low
    return val
def read_word_2c(adr):
    val = read_word(adr)
    if (val >= 0x8000):
        return -((65535 - val) + 1)
    else:
        return val
def dist(a,b):
    return math.sqrt((a*a)+(b*b))
def get_y_rotation(x,y,z):
    radians = math.atan2(x, dist(y,z))
    return -math.degrees(radians)
def get_x_rotation(x,y,z):
    radians = math.atan2(y, dist(x,z))
    return math.degrees(radians)
bus = smbus.SMBus(1)
address = 0x68 #MPU6050 I2C adresi
bus.write_byte_data(address, power_mgmt_1, 0)
while True:
    time.sleep(0.1)
    gyro_xout = read_word_2c(0x43) #Jiroskop register'lerini oku
    gyro_yout = read_word_2c(0x45)
    gyro_zout = read_word_2c(0x47)
    print "Jiroskop X : ", gyro_xout, " ölçekli : ", (gyro_xout / 131)
    print "Jiroskop Y : ", gyro_yout, " ölçekli : ", (gyro_yout / 131)
    print "Jiroskop Z : ", gyro_zout, " ölçekli : ", (gyro_zout / 131)
```

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--



STAJIN YAPILDIĞI DEPARTMAN	Araştırma Laboratuvarı	SAYFA	
YAPILAN İŞ	Eğitim - Araştırma	TARİH	.2018

```
accel_xout = read_word_2c(0x3b) #lvmeolcer register'larini oku
accel_yout = read_word_2c(0x3d)
accel_zout = read_word_2c(0x3f)
accel_xout_scaled = accel_xout / 16384.0
accel_yout_scaled = accel_yout / 16384.0
accel_zout_scaled = accel_zout / 16384.0
print "lvmeolcer X: ", accel_xout, " olcekli: ", accel_xout_scaled
print "lvmeolcer Y: ", accel_yout, " olcekli: ", accel_yout_scaled
print "lvmeolcer Z: ", accel_zout, " olcekli: ", accel_zout_scaled
print "X dondurme: ", get_x_rotation(accel_xout_scaled, accel_yout_scaled, accel_zout_scaled)
print "Y dondurme: ", get_y_rotation(accel_xout_scaled, accel_yout_scaled, accel_zout_scaled)
time.sleep(0.5)
```

1.1 STAJDAN SORUMLU MÜHENDİS	2 Ünvan Ad ve Soyad	Doktor Öğretim Üyesi Türker Türker	Firma İmza Kaşe	
------------------------------------	------------------------	---------------------------------------	-----------------------	--