



# Arm Mikroişlemci yapısı ve Arm Mikroişlemci ile yapılan bir uygulamanın tanıtımı

Çağdaş ELCAN 11014602

## MESLEKİ TERMİNOLOJİ 1

Yrd. Doç. Dr. Umut Engin AYTEN



## **1. ARM İŞLEMCİLER**

1.1 ARM İşlemcilerin Kısa Tarihi

1.2 ARM İşlemci Mimarisi

1.3 ARM çekirdeği kullanan Firma ve ürünleri

1.4 ARM Programlamada kullanılan derleyiciler

1.5 Neden ARM işlemciler

1.6 STM32F4 Discovery Kiti

1.7 STM32F4 Discovery GPIO uygulaması



## 2. I.M.U (DENGİ MODÜLÜ )

2.1 Accelerometer (İvme Sensörü) Tanıtımı

2.2 Gyroscope (Açı Sensörü ) Tanıtımı

2.3 STM32F4 Discovery Denge Modülü uygulaması

## ARM İşlemcilerin Kısa Tarihi

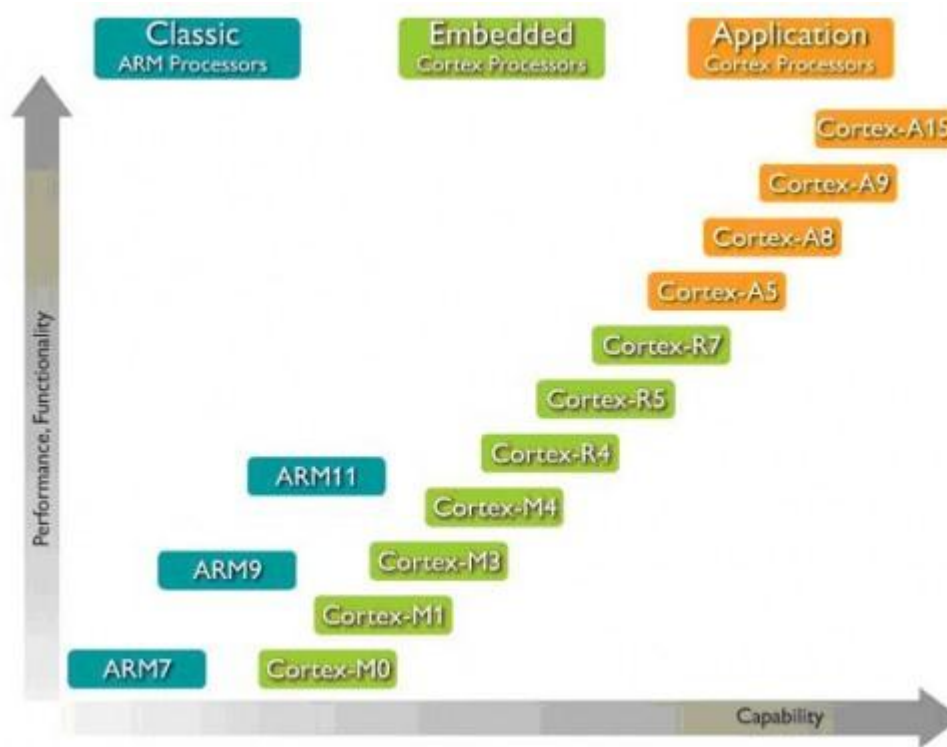
- ❖ ARM Holding, ilk olarak 1978 yılında İngiltere **Acorn Computers** ismi altında kurulmuştur.
- ❖ 1983 yılında **Roger Wilson ve Steve Furber** liderliğinde geliştirilmiş işlemci mimari üzerine çalışmaya başlanılmıştır.
- ❖ 1985 yılında dünyanın ilk **RISC** işlemcisini geliştirmiştir.
- ❖ 1987 senesinde ise düşük maliyetli pc'ler için ilk RISC ARM işlemcileri geliştirmeye başlamıştır.
- ❖ 1990 yılında ilk mimari tamamlanmış ve **Apple** firmasına ilk ARM çekirdek lisansı satılmıştır.

- ❖ 1993 yılında ise Texas Instruments firmasına lisans satmış ve yine aynı yıl ARM 7 çekirdeklerini tanıtmıştır.

## ARM İşlemci Mimarisi

- ❖ **ARM** "**A**dvanced **R**ISC **M**achines" cümlesinin kısaltılmış hali olup, "Gelişmiş RISC Makineleri" anlamına gelir.
- ❖ **RISC** kısaltmasının açılımı ise "indirgenmiş komut setli bilgisayar" anlamına gelen "**R**educed **I**nstruction **S**et **C**omputer" demektir.
- ❖ En kısa anlatım ile **ARM** kısa komut setlerinden oluşan gelişmiş bir mikro kontrol mimarisidir.

## Arm Mimari Ailesi :



ARM mimarisinde 3 temel grup bulunmaktadır:

- ❖ Klasik ARM işlemciler: ARM7, ARM9, ARM11
- ❖ Gömülü sistemlere yönelik ARM işlemciler: Cortex-M0, Cortex-M0, Cortex-M0, Cortex-M0, Cortex-M0
- ❖ Uygulama seviyesindeki ARM işlemciler: Cortex-A5, Cortex-A8, Cortex-A9, Cortex-A15

## Klasik ARM işlemciler:

- ❖ Bu gruptaki işlemciler daha çok **mikrodenetleyicilerin çekirdeklerini** oluşturmaktadırlar.
- ❖ Motor kontrolü, sinyal işleme gibi donanım seviyesine yakın işlemlerde kullanılmaktadır.
- ❖ Klasik ARM çekirdekleri en eski ARM ürünlerindendir ve kullanımları giderek azalmaktadır.
- ❖ Bunların yerine Cortex M ve R serisi ARM çekirdeklerinin kullanımı tavsiye edilmektedir.



## ARM Cortex Embedded İşlemciler:

- ❖ Cortex M ve R serileri olarak adlandırılmaktadır.
- ❖ Bu seri gerçek zamanlı ve düşük güç tüketimi gerektiren uygulamalarda kullanılmaktadır.
- ❖ NXP Semiconductors, STMicroelectronics, Texas Instruments, ve Toshiba gibi işlemci üreticileri bu çekirdeğe sahip mikrodeneleyici ürünleri sunmaktadırlar.

## ARM Cortex Uygulama İşlemcileri:

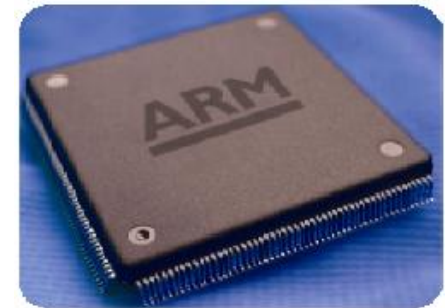
Cortex -A Serisi uygulama işlemcileri :

- ❖ Akıllı telefonlar, tablet bilgisayarlar gibi son kullanıcı ürünlerinin birçoğunda bu serideki çekirdeklere sahip işlemciler bulunmaktadır.
- ❖ Örnek vermek gerekirse iPhone, iPad, Samsung Galaxy Tablet, RIM Playbook gibi ürünlerde Cortex-A serisi ARM tabanlı mikroişlemciler bulunmaktadır.
- ❖ Bu işlemciler GHz frekansında işlem yapabilme kapasitesine sahiptir.

## Pipeline Sistemi :

- ❖ ARM işlemcisinin kalbi olarak nitelendirilebilir.
- ❖ Pipeline işlenecek olan komutları program hafızasından alarak 3 kademeli olarak işlemektedir.
- ❖ Ve bu şekilde diğer işlemcilerde yer alan sırayla komutları işle methodu **tarihe karışarak** 1 makine çevriminde tüm program satırları aynı anda gerçekleştirilecektir.
- ❖ Bu 3 işlem sırasıyla; **Fetch**, **Decode** ve **Execute** işlemleridir.

- 1) **Fetch** aşamasında komut program hafızasından alınır.
  - 2) **Decode** aşamasında işlenecek olan komut çözülür.
  - 3) **Execute** aşamasında ise çözülen komut işlenir.
- ❖ Bu şekilde mikrosaniyeler mertebesinde yapılan işlemler nanosaniyeler mertebesinde gerçekleştirilir.
  - ❖ Ve RTOS denilen gerçek zamanlı işletim sistemleri gerçekleştirilir.



## **ARM Çekirdeği Kullanan Firma ve Ürünler :**

- ❖ Piyasada kullanılan ve maliyet olarak ucuz ve temin edilebilir olanlar ;
- ❖ NXP Firmasının LPC Serisi
- ❖ ST Microelectronic Firmasının STM32F1,STM32F3,STM32F4 Discovery kitleri
- ❖ Texas Instruments Firmasının LM4F serileri

- ❖ ARM işlemcilerde RISC mimarisi kullanılır.
- ❖ RISC mimarisi ve CISC mimarisi nedir ?

### **CISC MİMARİSİ (Complex Instruction Set Computer)**

- ❖ İşlemci kendi üzerinde bulunan minyatür bir yazılımı kullanarak komut setlerini çalıştırır.
- ❖ Bu sayede komut setleri değişik uzunluklarda olabilir ve bütün adresleme modellerini kullanabilir. (16 bit-32 bit)
- ❖ Bunun dezavantajı çalışmak için daha karmaşık bir devre tasarımına ihtiyaç duyulmasıdır.

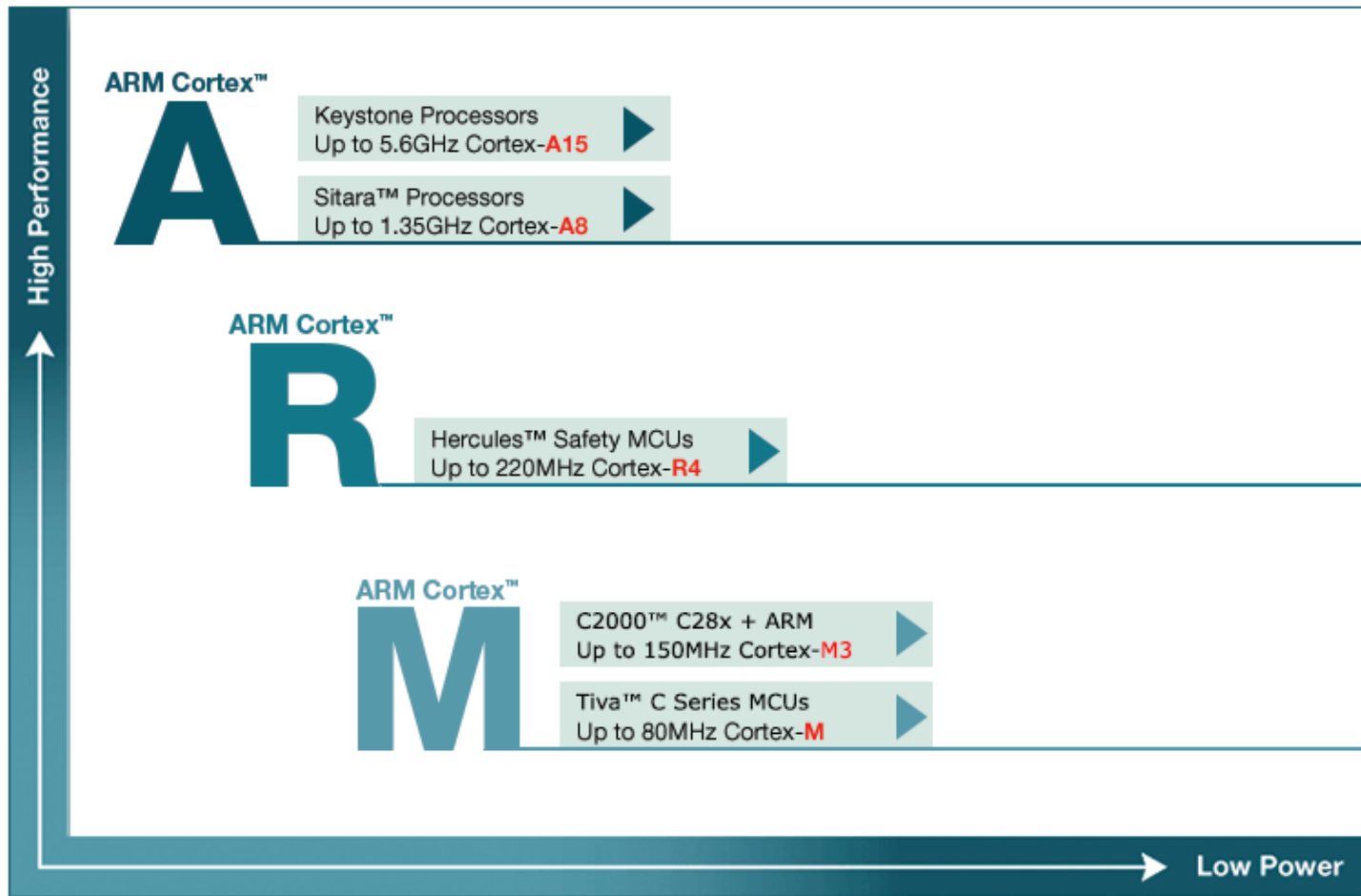
## RISC (Reduced Instruction Set Computer)

- ❖ Daha az komutla çalışır .
- ❖ Tek bir saat döngüsünde işlemini tamamlar.
- ❖ Komutları çok hızlı bir şekilde işler.
- ❖ Basit bir devre tasarımı kullanılmaktadır.

- ❖ Hafıza üzerinde (1,1) ve (1,2) koordinatlarında bulunan 2 sayı çarpılır ve sonuç (1,1) adresine yazılmak istenirse ;
- ❖ CISC : Mult (1,1) ,(1,2) şeklinde arka planda çalışan mini bir yazılım sayesinde işlem gerçekleştirilir.İşlem tek bir komut ile sağlanır.
- ❖ RISC: LOAD A (1,1)  
LOAD B (1,2)  
PROD A,B  
STORE (1,1),A

**HANGİSİ DAHA KOLAY !!!**





## ARM Programlamada kullanılan derleyiciler :

- ❖ Keil uVision (Mdk-Arm)
- ❖ ARM Development Studio
- ❖ Altium, TASKING VX –Toolset
- ❖ Atollic, TrueSTUDIO
- ❖ IAR, EWARM
- ❖ Keil, MDK-ARM
- ❖ Bu derleyiciler piyasada kabul görmüş en iyi derleyicilerdir. "Keil uVision" en profesyonel olan ARM firmasının ticari ürünüdür

## Neden ARM işlemciler ?

- 1) **RISC mimari:** Bu özellik işlemci tasarlamayı, üretmeyi, işlemcide çalışacak kodu üretmeyi kolaylaştırmaktadır.
- 2) **Güç tüketimi:** ARM işlemciler güç tüketimi konusunda çok iyiler, nedeni ise işlemcinin transistör sayısını düşük tutacak şekilde tasarlanmış olmasıdır. Bu özellik mobil aygıtlarda ARM kullanılmasının nedenlerinden biridir.
- 3) **RAM'e erişim:** RAM'e LOAD/STORE ile erişip bütün işlemler yazmaçlar üzerinde yapılır. Dolayısıyla elimizdeki komut seti daha temizdir.

#### 4) İşletim modeli:

- ❖ Intel işlemcilerde bir karar vermek için önce bir karşılaştırma (cmp) yapılır, sonra da bu karşılaştırmının sonucuna göre programın başka bir kısmına atlama yapılır(jump). Bu atlama işlemcinin kafasını karıştırır (bkz: pipelining).
- ❖ ARM işlemcilerde ise her komutun hangi durumda çalıştırılacağı komutun bir parçasıdır. İşlemci o anda bulunan komuta uymayan komutları **atlar**. Bu ise bazı algoritmaların çok daha temiz ve performanslı yazılabilmesini sağlar.

- 5) **Lisans modeli–Model çeşitliliği:** İsteyen bir şirket, ARM lisansı alıp kendi ARM işlemcilerini üretebilir. Bunun sonucu olarak da, piyasada tasarım olarak birbirinden farklı ama aynı şekilde çalışan, fiyatları, performansları, özellikleri farklı birçok ARM işlemci bulunur. Eğer bir mobil aygıt üretecekseniz, sadece Intel'in veya AMD'nin size sunduklarıyla kısıtlı değilsinizdir. Bunun en güzel örneği, **Apple'ın iPad için A4 çipini üretmesidir.**
- 6) **32–Bit:** 32–bit olmak, tek seferde daha fazla işlem yapabilmek demek. ARM ilk işlemciden beri 32–bit. Intel ise önce 4, sonra 8, 16, 32 ve 64 ile gitti. (**Intel** şu an 64, ARM hala 32–bit.) ARM rakipleri(örn:**Microchip** PIC serisi) ise ancak 2000 sonrasında 32–bit işlemcilerini çıkarabildiler.

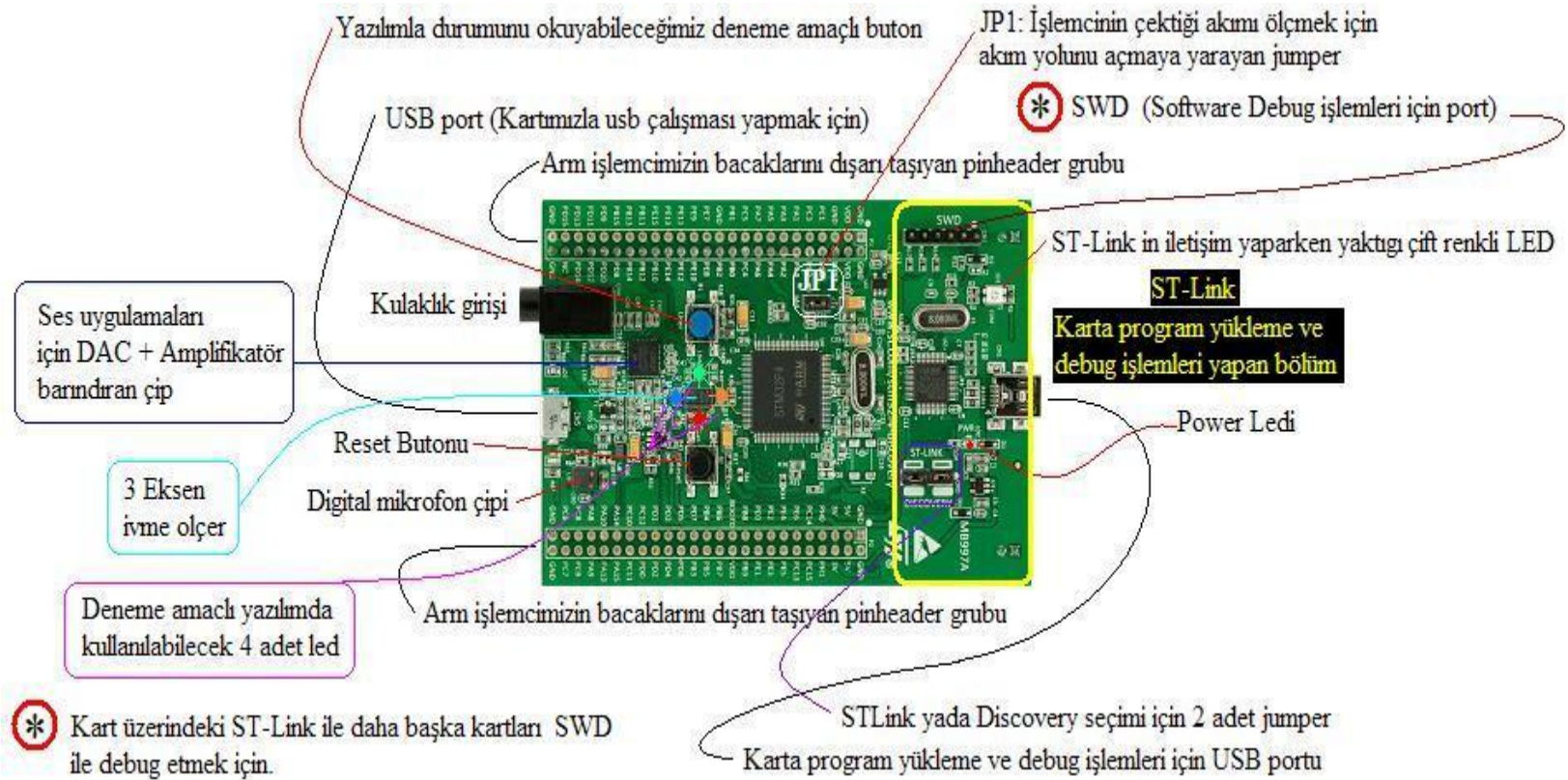
## STM32F4 DISCOVERY KIT :



- ❖ Kit üzerinde **ARM Cortex M4** tabanlı 168 MHz'lik bir mikrodenetleyici bulunur.
- ❖ Üzerinde **FPU**(*Floating Point Unit*) bulunmaktadır. Bunun sayesinde ondalıklı işlemlerin gerçekleştirilmesi çok hızlı bir şekilde gerçekleştirilmektedir.
- ❖ Firma aynı zamanda kit üzerine mikrofon ve ses çıkış jagi da eklemiştir.



## STM32F4 Geliştirme Kiti Donanımı :



- ❖ 1 MB Flash ve 192 KB RAM
- ❖ Dahili ST-LINK/V2 JTAG Debugger
- ❖ USB veya harici kaynaktan doğrudan 5v ile çalışabilme
- ❖ 3 V ve 5 V luk çıkış pinleri
- ❖ 3-eksen dijital ivmeölçer (LIS302DL)
- ❖ Mikrofon (MP45DT02)
- ❖ D sınıfı yükselteçli ses sürücü çipi(CS43L22)
- ❖ Sekiz adet LED



## STM32F4 DISCOVERY GPIO UYGULAMASI :

- ❖ **GPIO** uygulaması normal işlemcilerde kullanılan giriş-çıkış birimi olarak adlandırılır. Bu uygulama kit üzerinde programlama mantığını kavratacaktır.
- ❖ GPIO birimini kullanabilmek için aşağıdaki adımlar izlenmelidir.
- ❖ İlk olarak gpio.h header file dosyasından struct ve enum işlemleri yapılır.
- ❖ Struct işlemini gerçekleştirmek için “typedef struct” denilen kısımlardan tanımlamalar yapılmalıdır. GPIO işlemi için 1 tane struct tanımlanmıştır.
- ❖ GPIO\_InitTypeDef **GPIO\_Config\_ayar**;GPIO Struct tanımladıktan sonra header file içerisinde kullanabileceğimiz ayarlamalar mevcuttur.

### GPIO Konfigurasyon Mod Listesi

GPIO\_Mode\_**IN** / **OUT** / **AF** / **AN**

### **GPIO Çıkış Tipi Listesi**

**GPIO\_OType\_PP**

**GPIO\_OType\_OD**

### **GPIO Çıkış Frekansı Listesi**

**GPIO\_Speed\_2MHz / 25 MHz / 50 MHz / 100 MHz**

### **GPIO Konfigürasyon PullUp/ PullDown Listesi**

**GPIO\_PuPd\_NOPULL**

**GPIO\_PuPd\_UP**

**GPIO\_PuPd\_DOWN**

### **GPIO Pin Tanımlamaları**

**#define GPIO\_Pin\_x** şeklinde yapılmalıdır.

## GPIO Biriminde Kullanılan Fonksiyonlar ve Görevleri :

❖ `void GPIO_DeInit(GPIO_TypeDef* GPIOx);`

Bu fonksiyon belirtilen Portu Default(Baslangıç) ayarına çeker.Burada x değişkeni A,B,C,D,E,F,G,H,I değerlerinden birini alabilir.

Örnek kullanım :`void GPIO_DeInit(GPIOA);`

- ❖ `void GPIO_Init(GPIO_TypeDef* GPIOx, GPIO_InitTypeDef* GPIO_InitStruct);`
- ❖ Bu Fonksiyon GPIO\_InitStruct içersindeki parametrelere göre GPIOx portunu ayarlamaya yarar.
- ❖ GPIO\_TypeDef\* GPIOx buradaki x değişkeni A...I değerlerini alabilir.

GPIO\_InitTypeDef\* GPIO\_InitStruct

bu ifadenin eşdeğeri ise bizim belirlemiş olduğumuz struct ismidir.

GPIO\_InitTypeDef\* GPIO\_InitStruct=&Belirlenen isim

Örnek Kullanım : **void GPIO\_Init(GPIOA, &GPIO\_Config\_Ayar);**

❖ void **GPIO\_PinLockConfig**(GPIO\_TypeDef\* GPIOx, uint16\_t GPIO\_Pin)

Bu Fonksiyon kullanıldığı taktirde yeni bir reset gelene kadar belirtilen porta ait olan pini kitler ve üzerinde işlem yapılmasını engeller.

Örnek Kullanım : **void GPIO\_PinLockConfig(GPIOA, GPIO\_Pin\_12)**

## GPIO Okuma ve Yazma Fonksiyonları :

- ▶ `uint8_t GPIO_ReadInputDataBit(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)`
- ▶ Bu Fonksiyon verilen giriş portundaki pini okumak için kullanılır.
- ▶ GPIOx: x değeri(A..I) olabilir.
- ▶ GPIO\_Pin: Okunacak pini belirler. Burada ise pin seçimi için (0..15) arasında 16 tane seçme hakkı vardır.
- ▶ Örnek Kullanım : `GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_13)`

❖ `uint16_t GPIO_ReadInputData(GPIO_TypeDef* GPIOx)`

Bu Fonksiyon ile Verilen Portun Tamamı okunur.Sadece istenilen pin okunamaz.

Örnek Kullanım : `GPIO_ReadInputData(GPIOA)`

❖ `uint8_t GPIO_ReadOutputDataBit(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)`

Bu Fonksiyon Çıkış olarak tanımlanan Portun ait olan Pinini okur.

Örnek Kullanım : `GPIO_ReadOutputDataBit(GPIOA, GPIO_Pin_5)`

❖ `uint16_t GPIO_ReadOutputData(GPIO_TypeDef* GPIOx)`

Bu fonksiyon çıkış olarak belirtilen Portu okur.

Örnek Kullanım : `GPIO_ReadOutputData(GPIOA)`

❖ `void GPIO_SetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)`

Bu Fonksiyon ile Belirtilen Portun tanımlanan pini lojik olarak “1” yapılır.

Örnek Kullanım : `GPIO_SetBits(GPIOA, GPIO_Pin_11)`

❖ void **GPIO\_ResetBits**(GPIO\_TypeDef\* GPIOx, uint16\_t GPIO\_Pin)

Bu Fonksiyon ile Belirtilen Portun tanımlanan pini lojik olarak “0” yapılır.

Örnek Kullanım : **GPIO\_ResetBits(GPIOA, GPIO\_Pin\_10)**

❖ void **GPIO\_WriteBit**(GPIO\_TypeDef\* GPIOx, uint16\_t GPIO\_Pin, BitAction BitVal)

Bu Fonksiyon ile Belirtilen portun tanımlanan pinine yazma işlemi gerçekleştirilir.

Bu işlem reset de olabilir set de olabilir.Bunu belirleyen **BitAction**, **BitVal** kısmıdır.



**BitVal**:Seçilen bit için yazma işlemi gerçekleştirileceğini belirtir.

**BitAction** değerleri:

**Bit\_RESET**: Ait olan portun pinini temizlemeye yarar.

**Bit\_SET**:Ait olan portun pinini lojik 1 seviyesine getirir.

Örnek Kullanım : **GPIO\_WriteBit(GPIOA,GPIO\_Pin\_9, Bit\_RESET)**

❖ void **GPIO\_ToggleBits**(GPIO\_TypeDef\* GPIOx, uint16\_t GPIO\_Pin)

Bu Fonksiyon ile belirtilen Portun belirtilen pini toggle işlemine tabii tutulur. Toggle işlemi ledin yanıp sönmesi olarak bilinmektedir.

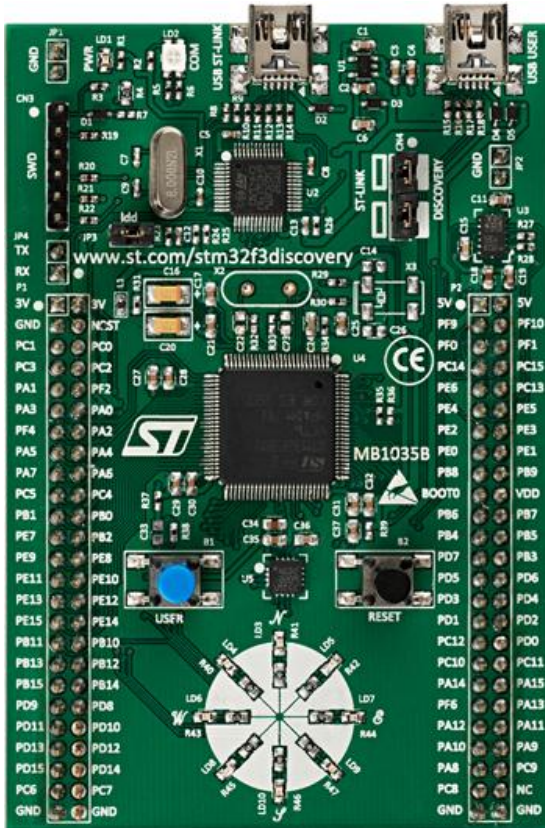
Örnek Kullanım : void **GPIO\_ToggleBits**(**GPIOA,GPIO\_Pin\_12**)

Belirtilen fonksiyonlar sayesinde GPIO uygulaması gerçekleştirilir.

## STM32F3 DISCOVERY DENGİ MODÜLÜ

- ❖ Denge modülü IMU olarak adlandırılmaktadır.
- ❖ Bir denge modülünün ne kadar hassas olduğunu modülün sahip olduğu serbestlik derecesi belirler.
- ❖ Serbestlik derecesi DOF olarak adlandırılır.
- ❖ 3 eksen İvme sensörü +3 eksen Açık sensörü + 3 eksen Magnetometre sensörünün oluşturmuş olduğu sistemin DOF değeri 9 dur.
- ❖ İvme sensörü accelerometer , açı sensörü gyroscope olarak adlandırılır.
- ❖ Denge modülü oluşturulurken sadece ivme sensörü veya açı sensörü kullanılmaz bu sensörden gelen veriler koordinat düzlemine dökülür.Bu koordinat düzlemleri karşılaştırılır.Bu sayede sensörlerden alınan veriler matematiksel analizler sonucu birleştirilir.

STM32F3 DISCOVERY Kit aşağıda gösterildiği gibidir :

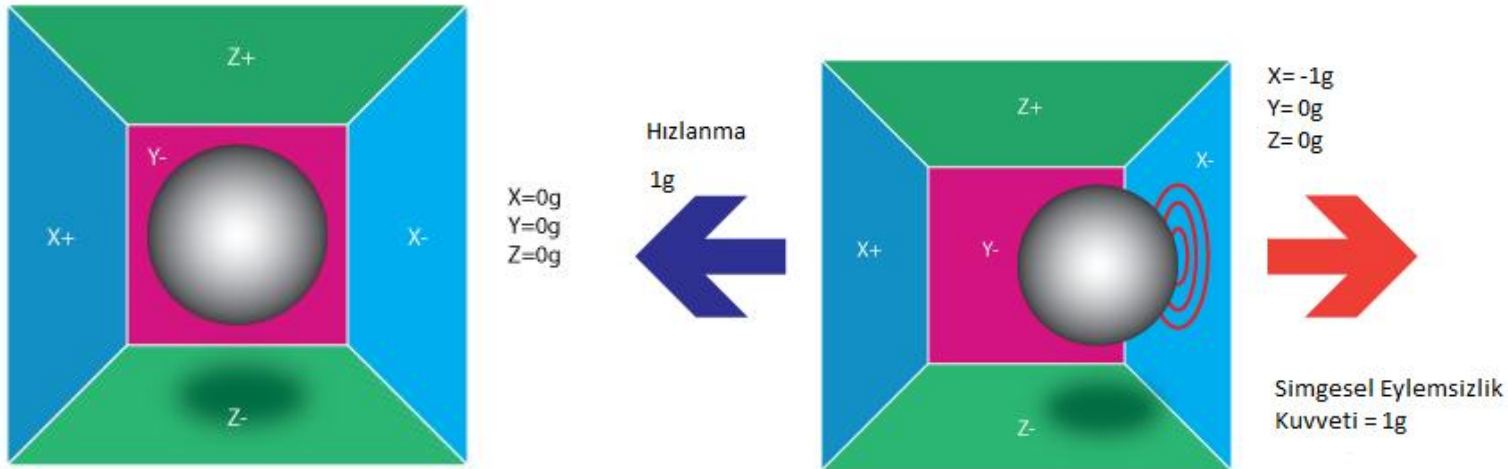


- ❖ L3GD20, ST MEMS motion sensor, 3-axis digital çıkış gyroscope
- ❖ LSM303DLHC, ST MEMS system-in-package featuring a 3D digital linear acceleration sensor and a 3D digital magnetic sensor

## İvme Sensörü :

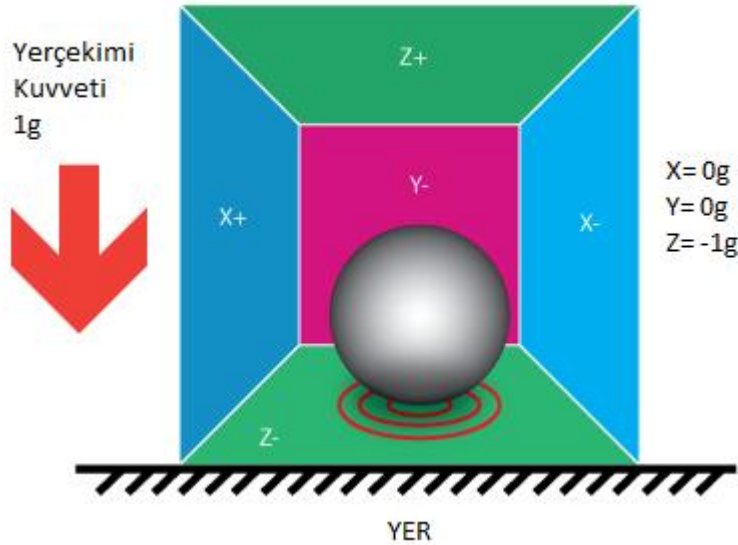
- ❖ İvme Sensörleri üzerlerine düşen statik(yerçekimi) veya dinamik (aniden hızlanma veya durma) ivmeyi ölçmektedirler.
- ❖ Sensörden aldığımız değer  $m/s^2$  veya yer çekimi (g) türünden ifade edilebilir.
- ❖ Uygulamalarda genelde yerçekimi türünden ifade edilmektedir.
- ❖ Eğer uzayda veya herhangi bir çekim alanının kapsamında değilseniz sensör üzerine 1g lik bir yerçekimi kuvveti etki etmektedir.
- ❖ Bu da bilindiği gibi yaklaşık olarak  $9.8m/s^2$  dir ve dünyadaki bulunulan noktaya göre değişiklik göstermektedir.
- ❖ Sensör sürekli olarak yer çekimi etkisi altında kaldığından eğim ölçer (örneğin yeni nesil akıllı cep telefonlarında kullanılmaktadır) .

Aşağıda bu sensörlerin nasıl çalıştığı anlatılmıştır.



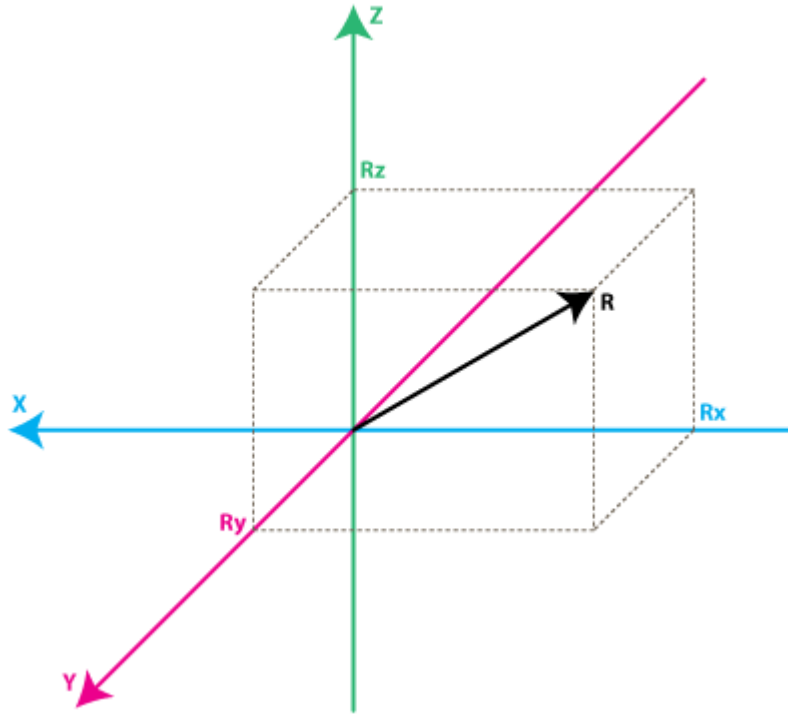
ivme sensörü temsilen kürenin  
uzaydaki görünüşü

Kutunun -X yönünde  
1g ile hızlanması



Sensör 1g ile -Z yönünde hızlandırılırsa

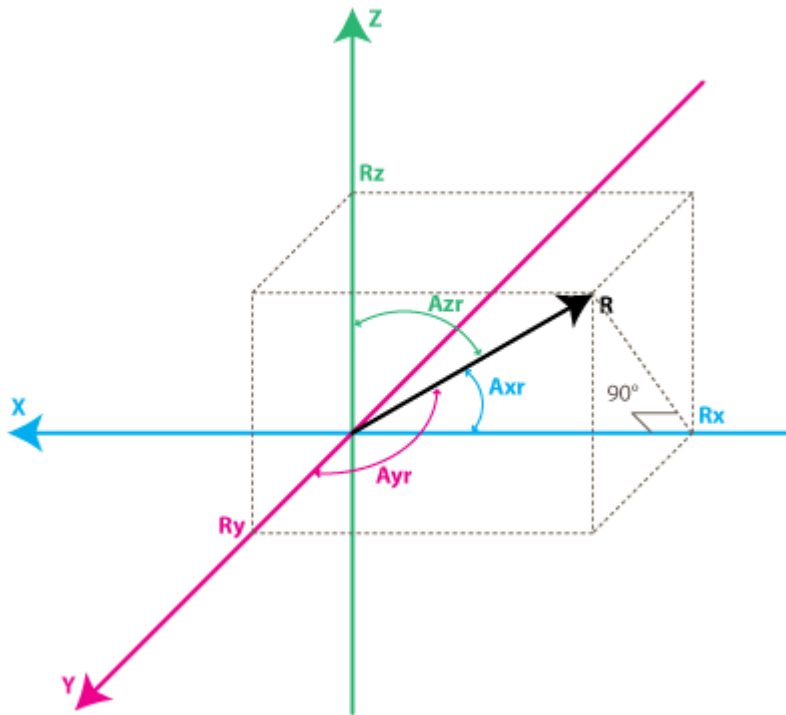
- ❖ İvmeölçerlerde benzer bir şekilde çalışmaktadır.
- ❖ Yüzeyleri basınca veya manyetik alana tepki verecek şekilde yapılmaktadır .
- ❖ İvmeölçerde bu tepkiyi ölçerek bir değer vermektedir.
- ❖ Sensörün yer yüzü ile yaptığı açı değiştiğinde sensörün eksenlerine uygulanan kuvvette değişecektir.



- ❖ İvme sensörünü temsilen kutu-koordinat sistemi modeli vardır.
- ❖ Bu şekilde 3eksendeki  $R_x, R_y, R_z$  vektörlerinin bileşimi olan  $R$  vektörünün eksenlerle yaptığı açı bulunabilir.

$$R^2 = R_x^2 + R_y^2 + R_z^2$$





$$\cos(A_{xr}) = R_x / R$$

$$A_{xr} = \arccos(R_x / R)$$

$$\cos(A_{yr}) = R_y / R$$

$$A_{yr} = \arccos(R_y / R)$$

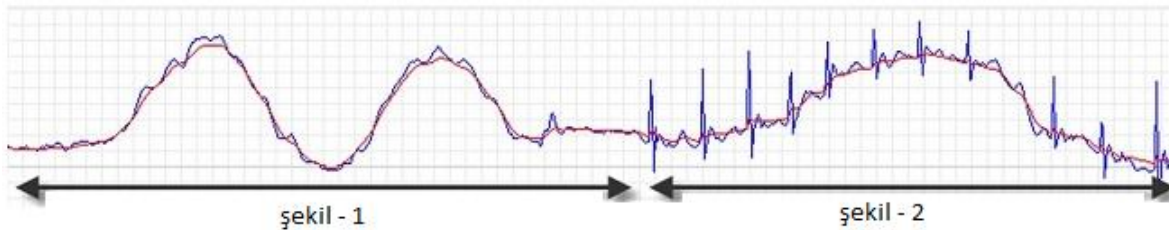
$$\cos(A_{zr}) = R_z / R$$

$$A_{zr} = \arccos(R_z / R)$$

## Açı Sensörü :

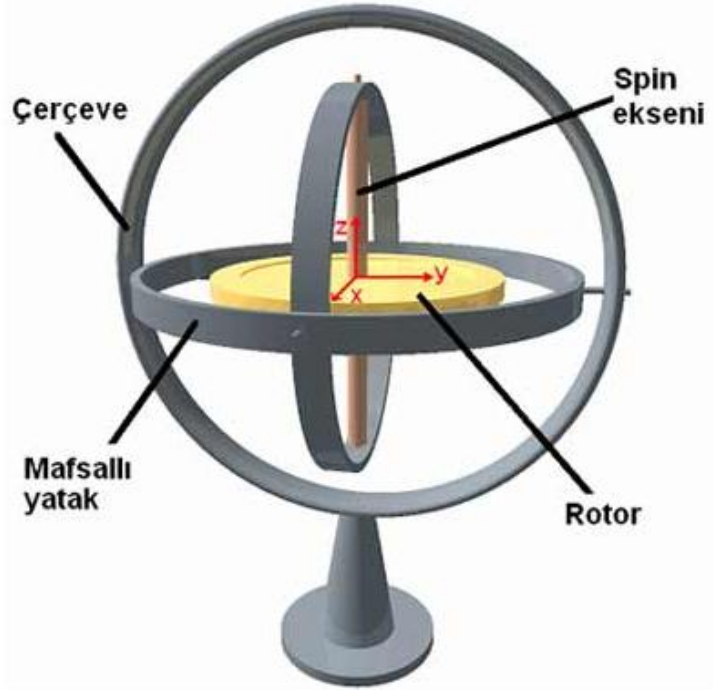
- ❖ Gyroscope olarak adlandırılır.
- ❖ Bu sensör ile istenen eksen yönlerindeki açısal hız değişimi ölçülür.
- ❖ Birimi derece / sn dir.
- ❖ Denge modülünde kullanılmaktadır.
- ❖ Accelerometer dan çıkan sonuçlar gyroscope ile bir çeşit filtreye tabii tutulmaktadır.

- ❖ Mavi sinyal : İvme ölçerden gelen sinyal
- ❖ Kırmızı sinyal : İvme ölçer çıkışının açılış ile filtrelendiğini gösteren sinyal



Şekil – 1; sadece titreşim gürültüsü eklenmediği durumdaki normal ivme sensörü çıkışını gösterir.

Şekil – 2 ; normal sinyal üzerine titreşim gürültüsü eklenmiş olan durumu göstermektedir.



Temsili Gyroscope şekli



DİNLEDİĞİNİZ İÇİN TEŞEKKÜR EDERİM.

**SORULAR**

## **Mesleki Terminoloji 1**