



**SIVAS CUMHURİYET ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**  
**GÖMÜLÜ SİSTEMLER DERSİ**  
**LABORATUVAR FÖYÜ**



**Bölüm 2**

---

**ADC**  
**(Analog Digital Converter)**

**Dersin Hocası:** Doç. Dr. Ahmet Gürkan YÜKSEK

**Dersin Asistanı:** Arş. Gör. Ahmet Utku ELİK

**İçindekiler**

1. ADC
2. MİKRODENETLEYİCİLERDE ADC
3. STM32 - ADC
4. ARM APP DENEY KİTİ – ADC
5. PRATİK UYGULAMA
6. LABORATUVAR DENEY UYGULAMASI

**Öğrencinin;**

Grup No :

Ad Soyad :

Okul No :

Teslim Tarihi :

## 1. ADC

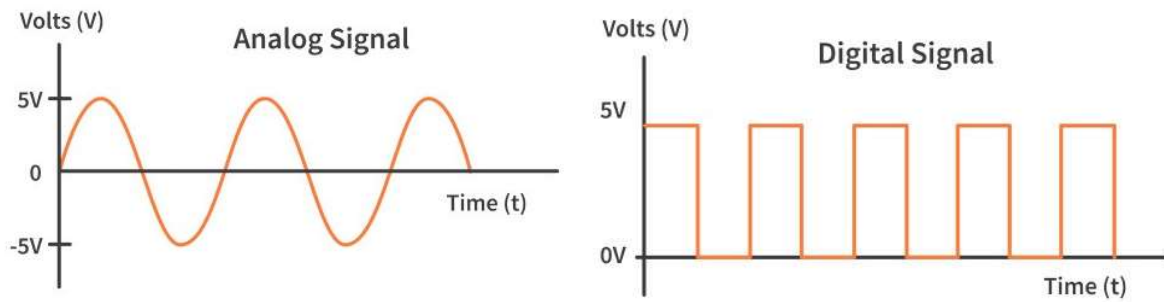
Analog Dijital Dönüştürücü veya ADC, bir analog sinyali ikili koda kodlayarak dijital devrelerin gerçek dünya ile arayüz oluşturmalarını sağlayan bir veri dönüştürücüdür. ADC, mikroişlemci kontrollü devrelerin gerçek dünya ile iletişim kurmasını sağlar. Gerçek dünyada analog sinyaller, çeşitli kaynaklardan ses, ışık, sıcaklık veya hareketi ölçebilen sensörlerden gelen sürekli değişen değerlere sahiptir ve birçok dijital sistem bu tür dönüştürücülerden gelen analog sinyalleri ölçerek çevreleriyle etkileşime girer.



Şekil 1. ADC

ADC'nin neden gerekli olduğunu anlamak için önce analog ve dijital sinyaller arasındaki farkı anlamak gerekmektedir.

Analog sinyaller kısaca sürekli (continuous) sinyaller denir. Yani sinyal zaman ya da uzay gibi sürekli bir değişkenin fonksiyonudur ve belirli bir zaman diliminde sınırsız değer alabilir. Dijital (sayısal) sinyaller kısaca ayırık zamanlı sinyaller denir. Analog sinyalden farklı olarak, bir değerler dizisi mevcuttur ve bu belli aralıklarla sürekli olmayacak şekilde yazılmasıdır. Bu yüzden de analog sinyaller gibi sürekli değildir. Sürekli olan analog sinyallerin belirli zaman aralıkları ile örneklenmesinden (sampling) veya bir sayı dizisi şeklinde sayısallaştırılmasından elde edilmektedir.

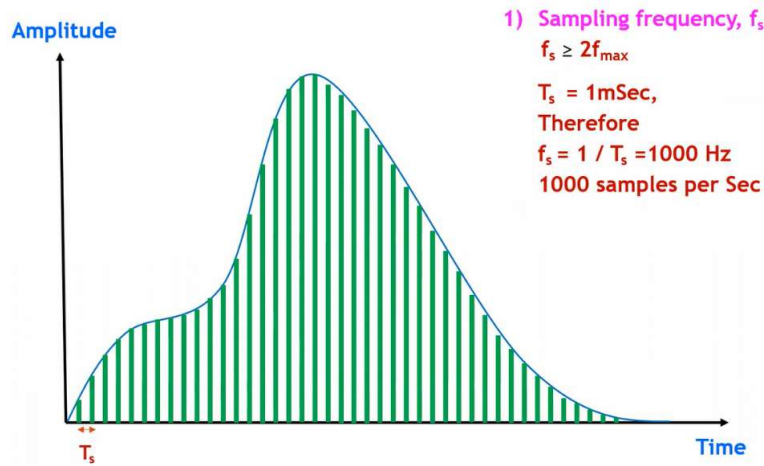


Şekil 2. Analog ve Dijital Sinyal Örneği

ADC, analog dünyadaki sürekli verileri dijital hale getirerek mikrodenetleyicilerin işleyebileceği formatta kullanılmasını sağlar. Temel olarak bir analogdan dijitale dönüştürücü, bir anda analog voltajın anlık görüntüsünü alır ve bu analog voltajı temsil eden bir dijital çıkış kodu üretir. Bir ADC, analog sinyali dijital veriye dönüştürmek için örnekleme (sampling), kuantalama (quantization) ve kodlama (encoding) işlemlerini kullanır.

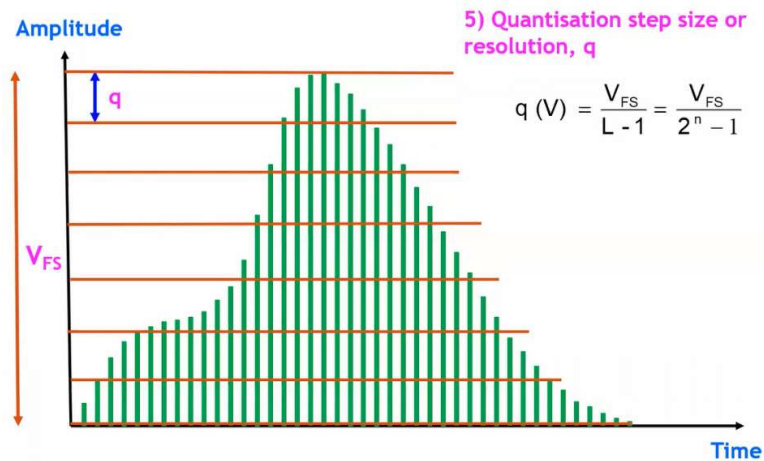
**1.1. Örneklem (Sampling):** Sürekli zaman analog sinyalin bir ayrık zamanlı sinyale dönüştürülmesi için analog işaret değerlerinin belirli zaman aralıklarında örneklenmesi gerekmektedir. Örneklem kısaca analog sinyalin örneklem anındaki genlik değerlerinin kesikli zaman sinyalinin o zaman anındaki genlik değeri olarak atanması işlemi olarak tanımlanabilir. Örneklem sayesinde zamanda sürekli olan analog sinyal zamanda ayrık olan dijital sinyale dönüştürülmektedir. Analog sinyalin doğru şekilde dijitalleştirilebilmesi için, örneklem frekansı en az sinyalin en yüksek frekansının iki katı olmalıdır (Nyquist Teoremi).

**1.1.1. Örneklem Frekansı (Sampling Frequency):** Genelde kolaylık ve uyum nedeniyle örneklem zaman aralıkları sabit seçilmekte ve iki ardışık örnek arasındaki zaman aralığı her zaman eşit olmaktadır. Örneklem frekansı  $f_s$  veya  $f_s$  ile gösterilmektedir ve bir saniye zaman aralığı boyunca alınan örnek sayısını belirtmektedir. Örneklem frekansı Nyquist Örneklem Teoremi gereği sinyal frekansının en az iki katına eşit olmalıdır.



Şekil 3. Örneklem (Sampling)

**1.2. Kuantalama (Quantisation):** Örneklem işleminden sonra ayrık zamanda ifade edilen ve PAM modülasyon şekline sahip sinyalin sahip olduğu genlik değerleri seviyelere ayrılmaktadır.



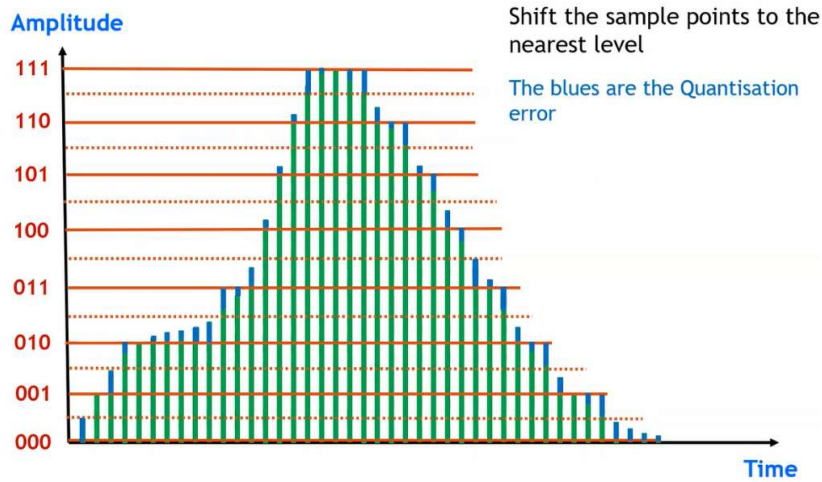
Şekil 4. Kuantalama (Quantisation)

ADC'nin çözünürlüğü (Resolution), kaç farklı dijital değer üretebileceğini belirler ve bit cinsinden ifade edilir (örneğin 8-bit, 10-bit, 12-bit, 16-bit). Çözünürlük arttıkça, sinyalin dijital karşılığı gerçeğe daha yakın olur.

**Tablo 1.** ADC Çözünürlük – Değer Aralığı

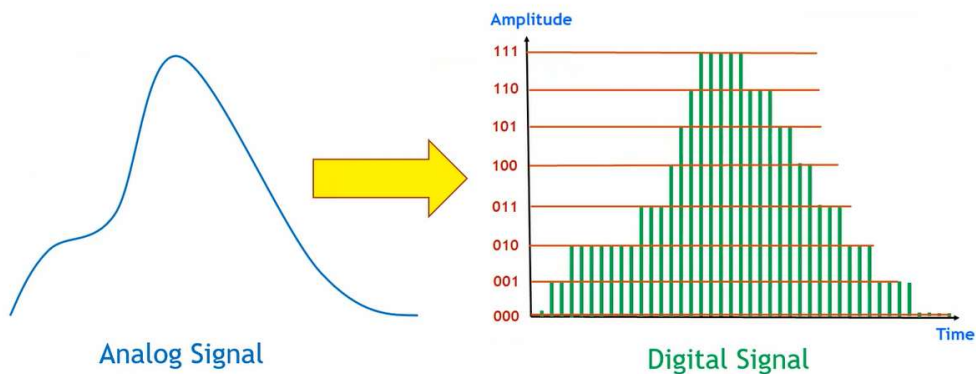
Çözünürlük	Dijital Değer Aralığı
8-Bit	0 - 255
10-Bit	0 – 1023
12-Bit	0 – 4095
14-Bit	0 - 65535

**1.3. Kodlama (Encoding):** Kodlama aşamasında kuantalama seviyeleri karşılık geldiği binary tabanlı sayı değerine dönüştürülmektedir.



**Şekil 5.** Kodlama (Encoding)

Sonuç olarak ADC'ye uygulanan analog sinyal Şekil-6'da gösterildiği üzere dijital sinyallere dönüştürülmektedir.

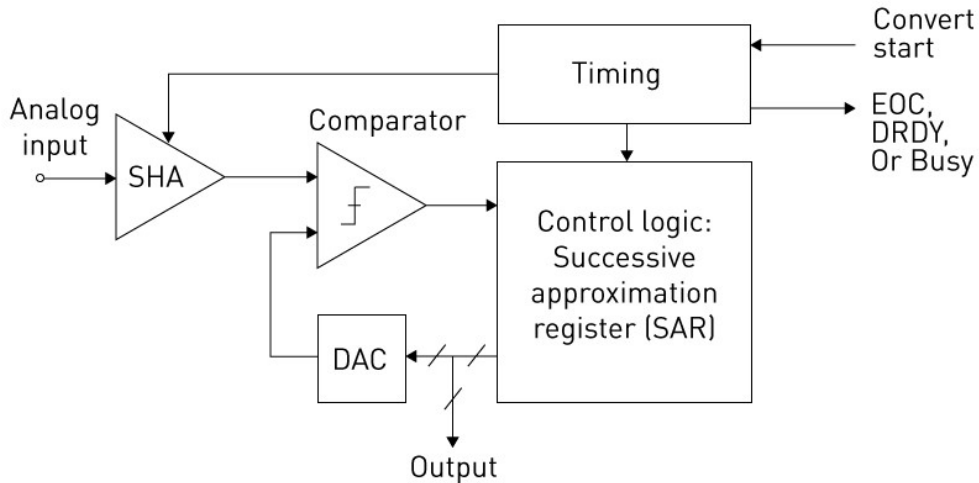


**Şekil 6.** ADC Giriş-Çıkış Sinyalleri

## 1.4. ADC Türleri

### 1.4.1. Successive Approximation Register (SAR) ADC

SAR ADC, mikrodnetleyicilerde yaygın olarak kullanılan, orta-yüksek hızda ve yüksek doğrulukta çalışan özellikle düşük güç tüketimi olan bir ADC türüdür. STM32F407 Discovery kartında da kullanılan ADC türü SAR ADC'dir. ARM mimarisine sahip STM32 mikrodnetleyicisi dışında PIC ve AVR mimarisine sahip mikrodnetleyicilerde de kullanılmaktadır. SAR ADC, ikili arama algoritmasını (Successive Approximation Algorithm) kullanarak giriş sinyalini dijital değere dönüştürmektedir.



Şekil 7. SAR ADC Blok Diyagram

Çalışma Adımları;

#### 1. Örnekleme ve Tutma (Sample & Hold)

- Giriş analog sinyali, bir Sample & Hold (S/H) devresi tarafından örneklenir ve dönüşüm süresince sabit tutulur.

#### 2. İlk Tahmin (MSB Denemesi)

- ADC, dijital çıkışın en yüksek bitini (MSB) 1 olarak belirler.
- Dijital-Analog Dönüştürücü (DAC), bu tahmini kullanarak bir analog voltaj üretir.

#### 3. Karşılaştırma (Comparator)

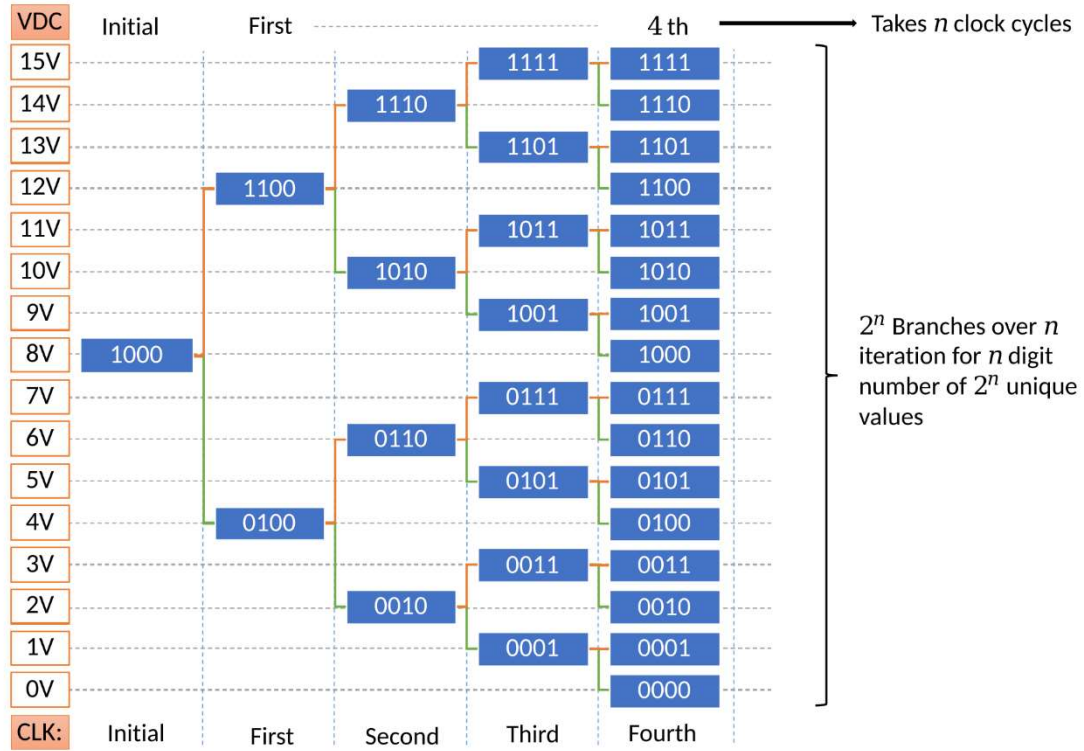
- Üretilen analog voltaj, giriş voltajı ile Comparator (Karşılaştırıcı) tarafından karşılaştırılır.
- Eğer üretilen voltaj giriş voltajından büyükse, MSB 0 yapılır.
- Eğer üretilen voltaj küçükse, MSB 1 olarak kalır.

#### 4. Ardışık Bitlerin Denenmesi (Successive Approximation)

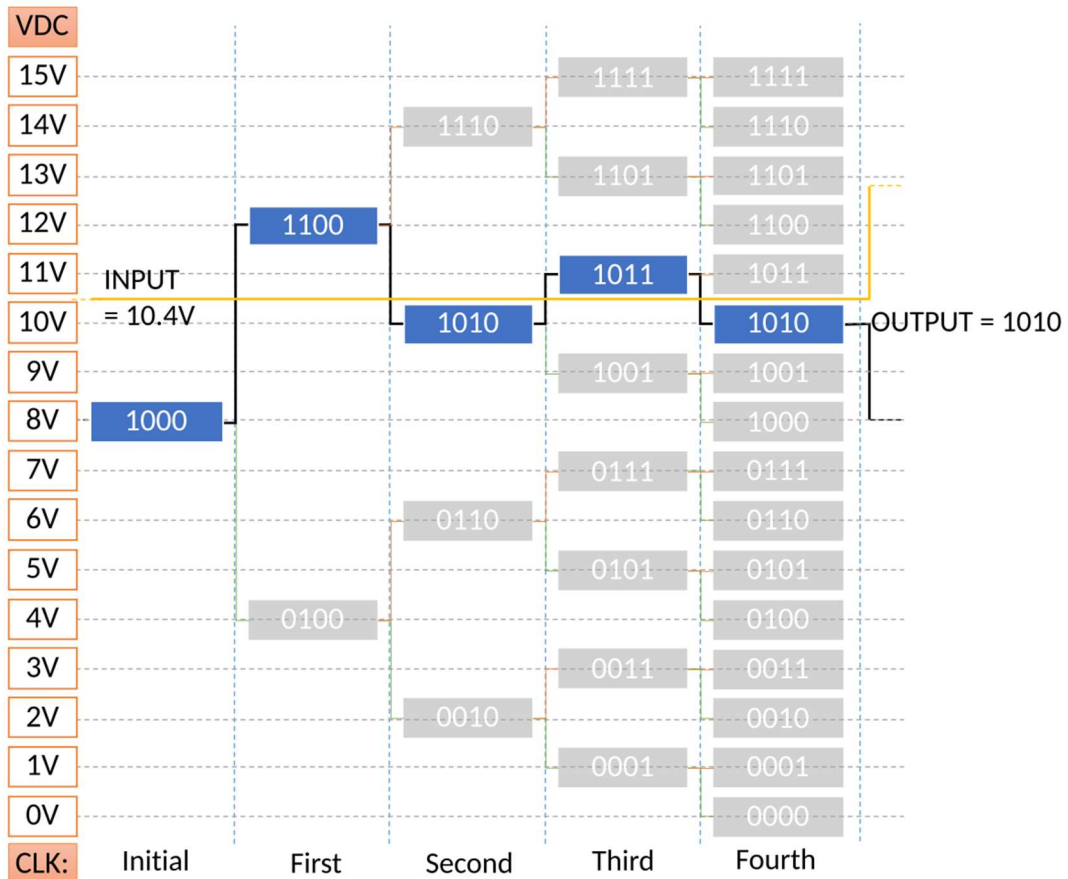
- Bir sonraki en büyük bit 1 olarak ayarlanır.
- DAC, güncellenmiş dijital değeri tekrar analog voltaja dönüştürür ve karşılaştırma işlemi tekrar edilir.
- Bu işlem N-bit ADC için N kez tekrarlanır.

#### 5. Sonuç Üretimi

- Tüm bitler denendiğinde, en son elde edilen dijital değer çıkışa verilir.



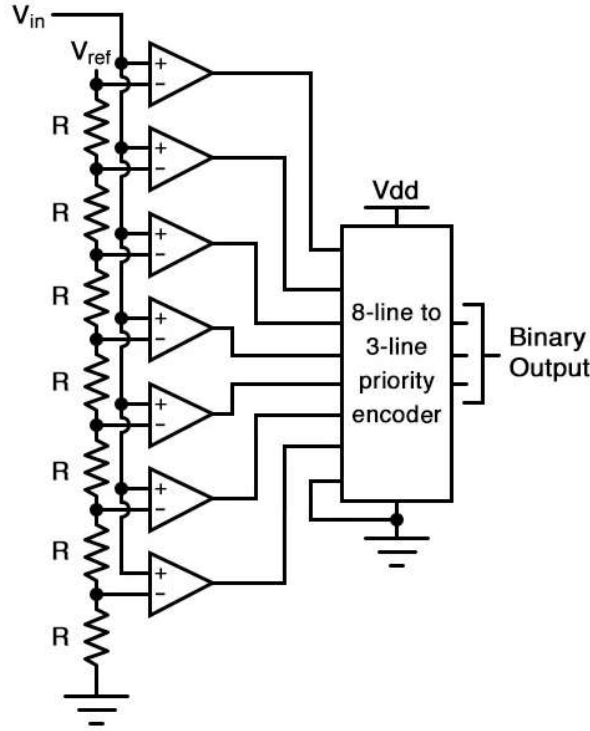
Şekil 8. Değerlerin bir ızgarada düzenlendiği 0V-15V arasındaki giriş voltajlarını ölçen 4 bitlik bir ADC'dir SAR ADC örnek kurulumu.



Şekil 9. 10.4'luk giriş geriliminde SAR ADC örneği.

### 1.4.2. Flash ADC

Flash ADC, en hızlı ADC türlerinden biridir ve çok yüksek hızda analogdan dijital dönüşüm yapabilme yeteneğine sahiptir. Bu yüzden GHz seviyesinde örnekleme hızlarına ulaşabilir ve gerçek zamanlı veri işleme uygulamalarında yaygın olarak kullanılır. Flash ADC, çok sayıda karşılaştırıcı (comparator) ve bir kod çözücü (encoder) kullanarak tek döngüde dönüşüm yapmaktadır. Şekil 10'da örnek bir 3-Bitlik Flash ADC blok diyagramı verilmiştir.



Şekil 10. 3-Bit Flash ADC Blok Diyagramı 1

Çalışma Adımları;

#### 1. Gerim Bölücü

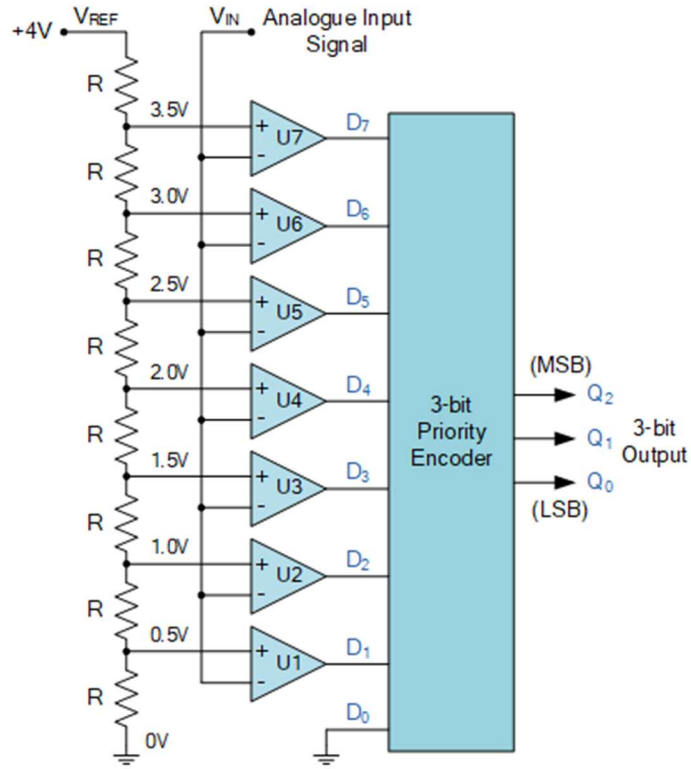
- Referans voltajı  $2^n - 1$  adet karşılaştırıcı için oluşturulan sabit gerilim seviyelerine bölünür.
- Bu gerilim seviyeleri, ADC'nin çözünürlüğüne bağlıdır (örneğin, 3-bit ADC için 7 referans gerilimi kullanılır).

#### 2. Karşılaştırıcı

- Her karşılaştırıcı, giriş voltajını kendisine bağlı referans voltajı ile karşılaştırır.
- Eğer giriş voltajı, belirli bir referans geriliminden yüksekse, karşılaştırıcının çıkışı 1, düşükse 0 olur.

#### 3. Kodlayıcı

- Karşılaştırıcılardan gelen dijital sinyaller bir kod çözücüye (Priority Encoder) gönderilir.
- Kod çözücü, en yüksek seviyedeki 1'in bulunduğu konumu belirleyerek dijital çıkışı üretir.



Şekil 11. 3-Bit Flash ADC Blok Diyagramı 2

Örneğin yukarıda verilen referans gerilimi 4V olan 3-bitlik bir Flash ADC için gerilim bölücü devrede 8 ( $2^3$ ) farklı değer elde edilip analog giriş sinyali ile karşılaştırıldıktan sonra 3-bitlik encoder ile kodlanarak dijital sinyale dönüştürülmektedir. Aşağıdaki tabloda farklı analog giriş sinyallerine karşılık elde edilen dijital çıkış değerleri verilmiştir.

Tablo 2. 3-Bit Flash ADC

Analog Giriş Gerilimi (Vin)	Karşılaştırmacı Çıkışı (D7-D0)	Dijital Çıkış (Binary)
0.3V	0000000	000
0.8V	0000001	001
2.2V	0001111	100
3.3V	0111111	110
3.8V	1111111	111

Bu süreci daha iyi anlamak için dosyalar kısmında yer alan Proteus dosyasını veya aşağıda yer alan simülasyon linkini inceleyebilirsiniz.

3-Bit Flash ADC Simülatör: <https://electronics-course.com/flash-adc>

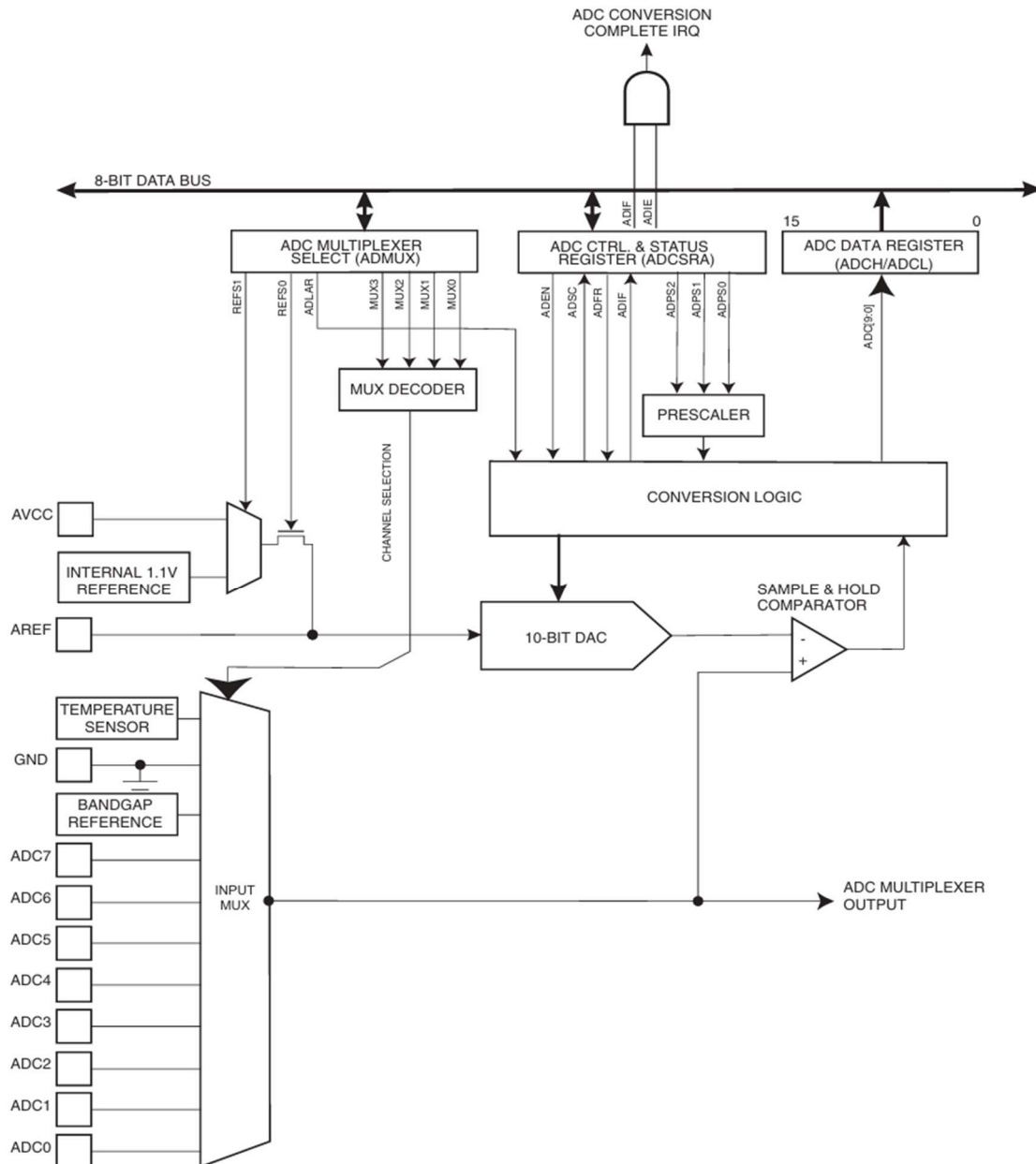


Bahsedilen ADC türlerinin yanısıra çalışma hızı düşük fakat hassasiyeti yüksek olan Sigma - Delta ( $\Sigma$ - $\Delta$ ) ADC türünde medikal cihazlar ve ses işleme uygulamalarında tercih edilmektedir.

## 2. MİKRODENETLEYİCİLERDE ADC

Mikrodenetleyiciler, genellikle analog girişleri doğrudan işleyemez, çünkü mikrodenetleyiciler sayısal (dijital) sistemlerdir. Bu nedenle, Analogdan Dijitale Dönüştürücü (Analog Digital Converter, ADC) çevre birimi, analog sinyalleri dijital formatta işlemeye uygun hale getirmek için kullanılır.

Örneğin 8-Bit AVR mimarisine sahip mikrodenetleyici yapısı içerisinde yer alan ADC çevre biriminin blok diyagramı Şekil 12’de verilmiştir.



Şekil 12. 8-Bit AVR Mimarisine sahip mikrodenetleyicilerdeki ADC yapısı.

Mikrodenetleyicilerdeki ADC çevre birimi, analog girişlerden aldığı voltajı belirli bir bit çözünürlüğünde dijital değerlere dönüştüren bir donanım modülüdür. ADC modülü, aşağıdaki temel bileşenlerden oluşur;

ADC'nin Ana Bileşenleri;

**1. Analog Çoklayıcı (Multiplexer - MUX)**

- Birden fazla analog girişten (ADC kanallarından) gelen sinyalleri seçmek için kullanılır.
- Tek ADC birimi, birden fazla analog giriş kanalına sahip olabilir.
- Örneğin STM32F407 mikrodenetleyicisinde ADC1, ADC2 ve ADC3 olmak üzere 3 ADC modülü vardır ve her biri 16 kanala sahiptir.

**2. Örnekleme ve Tutma (Sample & Hold - S/H) Devresi**

- Analog sinyal belirli bir anda örneklenir (sampling) ve ADC dönüşüm süresince sabit tutulur.
- Örnekleme süresi, ADC'nin giriş empedansı ve giriş sinyalinin frekansına bağlı olarak belirlenir.
- ADC modülünün örnekleme frekansı (Sampling Rate,  $F_s$ ), Nyquist Teoremine uygun olarak seçilmelidir.

**3. ADC Çekirdeği (ADC Core)**

- Analog sinyali dijital forma dönüştüren temel bileşendir.
- Mikrodenetleyicilerde genellikle Successive Approximation Register (SAR) ADC kullanılır.

**4. Dijital Filtreleme (Opsiyonel)**

- Gürültüyü azaltmak için kullanılan düşük geçiren filtreler bulunabilir.

**5. Veri Kayıtları (ADC Data Registers)**

- Dijital olarak dönüştürülmüş verileri saklayan ve işlemciye aktaran bellek alanlarıdır.
- ADC tamamlandığında sonuç genellikle ADC veri kaydında (ADC\_DR) bulunur.

### 3. STM32 – ADC

STM32F407VG mikrodnetleyicisinde 3 adet bağımsız ADC modülü (ADC1, ADC2, ADC3) bulunmaktadır. Her ADC modülü 16 adede kadar analog giriş kanalını destekler ve 12-bit çözünürlüğe sahiptir. STM32F407 ADC modülü, Successive Approximation Register (SAR) ADC yapısına sahiptir. SAR ADC, ikili arama algoritması (binary search algorithm) ile giriş voltajını dijital değerlere dönüştürür.

#### 3.1. STM32 – ADC Çalışma Modları

STM32F407 mikrodnetleyicisindeki ADC, farklı uygulama gereksinimlerine uygun olarak çeşitli modlarda çalışabilir. Bu modlar, veri toplama sürecini optimize etmek ve mikrodnetleyici kaynaklarını verimli kullanmak için kullanılır.

ADC'nin altı temel çalışma modu vardır;

1. Tek Kanal Tek Dönüşüm Modu (Single Conversion Mode)
2. Sürekli Dönüşüm Modu (Continuous Conversion Mode)
3. Taramalı Dönüşüm Modu (Scan Mode)
4. Kesme (Interrupt) Tabanlı ADC Modu
5. DMA (Direct Memory Access) ile ADC Modu
6. Çift ve Üçlü ADC Modları (Multi-ADC Mode - Dual/Triple Mode)

##### 3.1.1. Tek Kanal Tek Dönüşüm Modu (Single Conversion Mode)

Bu modda, ADC sadece belirlenen tek bir kanaldan analog sinyali okur ve bir kez dönüşüm yapar. Dönüşüm tamamlandıktan sonra ADC otomatik olarak durur ve yeni bir dönüşüm başlatılmaz.

Kullanım Alanları;

- Batarya voltajı ölçümü
- Potansiyometre okuma

##### 3.1.2. Sürekli Dönüşüm Modu (Continuous Conversion Mode)

ADC, her dönüşüm tamamlandığında otomatik olarak yeni bir dönüşüm başlatır. Mikrodnetleyici ADC verilerini sürekli olarak okuyabilir.

Kullanım Alanları;

- Gerçek zamanlı ortam sensörleri (Sıcaklık, gaz, nem vb.)
- Mikrofon verisi okuma

##### 3.1.3. Taramalı Dönüşüm Modu (Scan Mode)

Bu mod, birden fazla ADC kanalını belirli bir sırayla okuyarak tüm verileri kayıt altına alır. Çoklu sensör okuma işlemleri için uygundur.

Kullanım Alanları;

- Birden fazla sıcaklık, ışık veya gaz sensörünü aynı anda okumak
- Voltaj ve akım ölçümü yapan sistemlerde çoklu girişleri işlemek

#### 3.1.4. Kesme (Interrupt) Tabanlı ADC Modu

Bu modda, ADC dönüşümü tamamlandığında bir kesme (interrupt) oluşturur. Mikrodenetleyici sürekli beklemek yerine, kesme geldiğinde işlemi tamamlar. Gerçek zamanlı sistemlerde kullanışlıdır.

Kullanım Alanları;

- Mikrodenetleyicinin başka işlemlerle meşgul olduğu durumlarda ADC ölçümünü verimli hale getirmek
- Gerçek zamanlı sistemlerde mikrodenetleyici yükünü azaltma
- Güç tüketimini azaltarak uyku modunda ADC ölçümü yapmak

#### 3.1.5. DMA ile ADC Modu (Direct Memory Access)

Bu modda, ADC verileri CPU kullanmadan doğrudan RAM'e aktarılır, böylece CPU yükü azalır. Yüksek hız gerektiren işlemler için kullanışlıdır.

Kullanım Alanları;

- Nesnelerin İnterneti (IoT) cihazlarında veri toplama
- Kablosuz sensör ağları

#### 3.1.6. Çift ve Üçlü ADC Modları (Multi-ADC Mode - Dual/Triple Mode)

STM32F407'de birden fazla ADC modülü bulunur (ADC1, ADC2, ADC3). Bu modlar, çok kanallı ve yüksek hızda veri toplama işlemleri için kullanılır. Paralel ADC çalıştırarak veri toplama hızını artırır.

Kullanım Alanları;

- Osiloskop, hızlı veri toplama sistemleri
- Güç analiz sistemleri

### 3.2. STM32 – ADC HAL

HAL (Hardware Abstraction Layer), STMicroelectronics tarafından sağlanan bir üst seviye API'dir ve donanım seviyesindeki işlemleri kolaylaştırır. STM32CubeIDE ortamında ADC yapılandırması ve kullanımı için HAL ADC sürücüsü kullanılır.

HAL ADC Kütüphanesi, konfigürasyon, başlatma, okuma ve kesme/DMA yönetimi için çeşitli fonksiyonlar içerir.

#### HAL\_ADC\_Start (ADC\_HandleTypeDef \* hadc)

**Metot Açıklaması** : Bu metot çağrıldıktan sonra ADC çevre birimini başlatır, giriş voltajını okur ve dijital değere dönüştürmeye başlar.

**Parametreler** : Belirtilen ADC için yapılandırma bilgilerini içeren handle (tanıtıcı) pointer değişken.

**Dönüş Değeri** : HAL\_StatusTypeDef veri tipinde durum değişkeni.

#### HAL\_ADC\_PollForConversion (ADC\_HandleTypeDef \* hadc, uint32\_t Timeout)

**Metot Açıklaması** : Bu metot ADC dönüşümünün tamamlanmasını beklemek için kullanılır.

**Parametreler** : Belirtilen ADC için yapılandırma bilgilerini içeren handle (tanıtıcı) pointer ve ADC dönüşümünün tamamlanmasını beklemek için maksimum süre Timeout değişkenleri.

**Dönüş Değeri** : HAL\_StatusTypeDef veri tipinde durum değişkeni.

#### HAL\_ADC\_GetValue (ADC\_HandleTypeDef \* hadc)

**Metot Açıklaması** : ADC, giriş voltajını dijital değere dönüştürdükten sonra, bu değer ADC veri kaydedicisine (ADC\_DR) yazılır. Bu metot ADC\_DR kaydedicisindeki bu değeri döndürür.

**Parametreler** : Belirtilen ADC için yapılandırma bilgilerini içeren handle (tanıtıcı) pointer değişken.

**Dönüş Değeri** : uint32\_t veri tipinde ADC dönüşüm sonucu.

#### HAL\_ADC\_Stop (ADC\_HandleTypeDef \* hadc)

**Metot Açıklaması** : Bu metot çağrıldıktan sonra ADC dönüşümünü durdurur ve ADC çevre birimini kapatır.

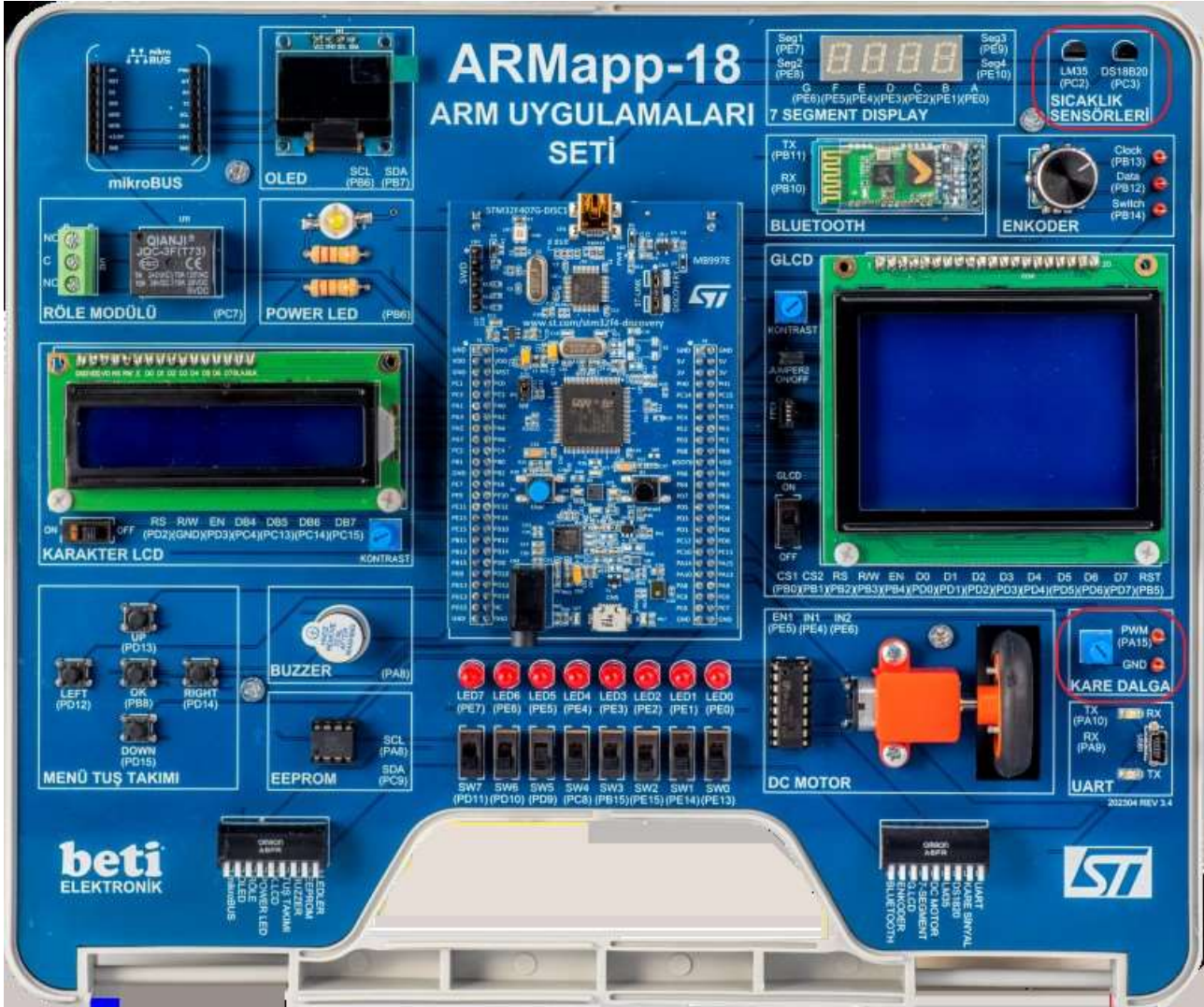
**Parametreler** : Belirtilen ADC için yapılandırma bilgilerini içeren handle (tanıtıcı) pointer değişken.

**Dönüş Değeri** : HAL\_StatusTypeDef veri tipinde durum değişkeni.

#### 4. ARM APP DENEY KİTİ – GPIO

Laboratuvar uygulamalarında kullanılan ARM APP Deney Kiti içerisinde ADC çevre biriminin kullanımı için birçok komponent bulunmaktadır. Bunlar;

- Potansiyometre (PC5)
- LM35 Sıcaklık Sensörü (PC2)
- DS18B20 Sıcaklık Sensörü (PC3)



Şekil 13. ARM APP Deney Kiti

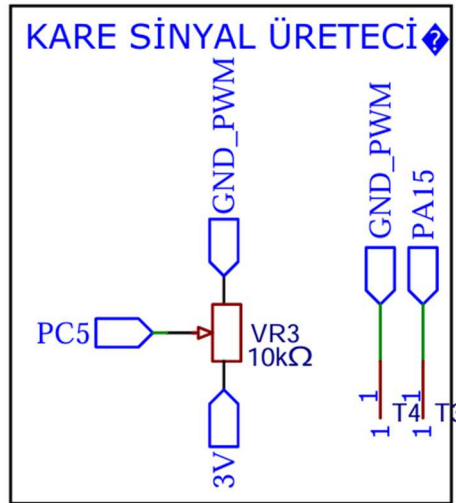
ADC çevre birimi mikrodenetleyici uygulamalarında genellikle sensör verilerini ölçmek ve anlamlandırmak için kullanılır. ARM APP Deney Kiti içerisinde bulunan STM32F407VGT6 mikrodenetleyicisi için 0 – 3.3V aralığındaki analog bir sinyal 12-Bit ADC için 0 – 4095 aralığında dijital sinyale karşılık gelmektedir. Elde edilen bu dijital değer çeşitli matematiksel işlemler uygulanarak örneğin Gerilim (V) veya sıcaklık (°C) gibi anlamlı bilgi elde edilir.

## 5. PRATİK UYGULAMA

Laboratuvar uygulamasına başlamadan önce ADC çevre birimi ile ilgili bir pratik uygulama yapalım. Öncelikle bir Gömülü Sistem uygulamasına başlamadan önce yapılması gereken mikrodnetleyici ile uygulamada kullanılacak olan bileşenin donanımsal bağlantılarının yapılması ve kontrol edilmesi gerekmektedir. Ders kapsamında kullanılan deney kitinde mikrodnetleyici ve bileşenler PCB kart üzerinde donanımsal olarak bağlı bulunduğu için kullanıcıyı kablolama ve bağlantı işlemlerinden soyutlayarak doğrudan Gömülü Sistem Yazılımı geliştirmesine imkan tanımaktadır.

Gömülü Yazılım geliştirmeye başlarken uygulamada kullanılacak olan bileşenlerin mikrodnetleyiciye bağlı bulunduğu pinlerin yapılandırılması gerekmektedir. Yapılandırma işlemi laboratuvar uygulamalarında kod geliştirme için kullanılacak olan STM32CubeIDE içerisinde yer alan CubeMX modülü ile yapılmaktadır. Örneğin ARM APP Deney Kitinde bulunan Potansiyometre Devresi üzerinde değişken analog sinyal okuma uygulamasını geliştirelim.

STM32CubeIDE’de proje oluşturduktan sonra gelen CubeMX modülü ile pin yapılandırmalarını gerçekleştirebilmek için Potansiyometre Devresinin mikrodnetleyicideki bağlı bulunduğu pinlerin Port ve Numaralarını ARM APP Deney Kiti Şematik Tasarım dosyasına bakarak belirlenmelidir. Şekil 14’de ilgili devre şeması gösterilmiştir.



Şekil 14. ARM APP Deney Kiti Potansiyometre Devre Şeması

Verilen devre şemasına göre GPIOC portunun 5 numaralı pinini CubeMX modülü üzerinden ADC1\_IN15 olarak yapılandırılması gerekmektedir. ADC parametreleri ise aşağıdaki şekilde belirlenmelidir;

- **Clock Prescaler** : PCLK2 divided by 4
- **Resolution** : 12 bits (15 ADC Clock cycles)
- **Continuous Conversion Mode** : Enabled
- **End of Conversion Selection** : EOC flag at the end of single channel conversion

CubeMX üzerinde pin yapılandırması bittikten sonra “Generate Code” ikonuna tıklayarak kod geliştirme arayüzünü açıp aşağıda verilen kodu yazıp “Debug/Run” ikonuna tıklayıp yazılımı geliştirme kartına yüklüyoruz.

```

#include "main.h"

ADC_HandleTypeDef hadc1;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);

uint16_t adc_value;

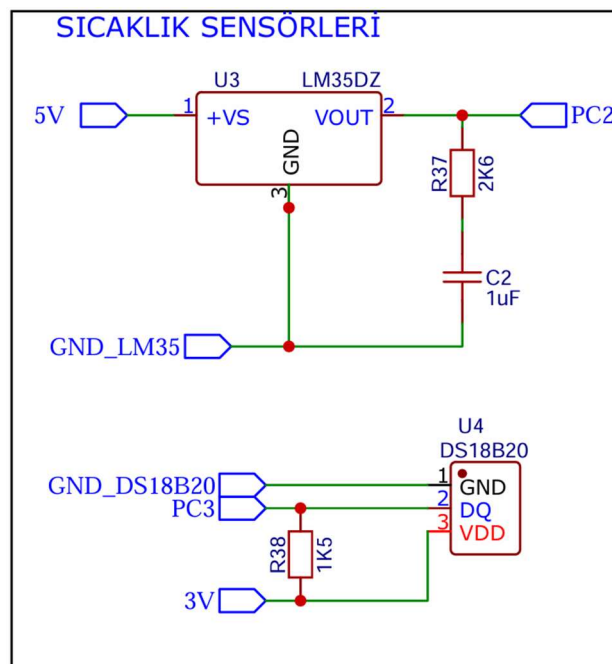
void Read_ADC()
{
    HAL_ADC_Start(&hadc1);
    if(HAL_ADC_PollForConversion(&hadc1, 100000) == HAL_OK)
    {
        adc_value = HAL_ADC_GetValue(&hadc1);
    }
    HAL_ADC_Stop(&hadc1);
}

int main(void){
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_ADC1_Init();

    while (1){
        Read_ADC();
    }
}

```

ADC çevre birimi üzerinden dijital giriş uygulaması geliştirmek için ARM App Deney Kiti üzerinde yer alan Şekil 15’de verilen LM35 ve DS18B20 Sıcaklık Sensör devre şemaları kullanılarak uygulama geliştirilebilir.



Şekil 15. ARM APP Deney Kiti LM35/DS18B20 Sıcaklık Sensör Devre Şeması



## **6. LABORATUVAR DENEY UYGULAMASI**

Uygulama isterleri ders esnasında paylaşılacaktır.

## 7. KAYNAKLAR

- [1] [https://en.wikipedia.org/wiki/Successive-approximation\\_ADC](https://en.wikipedia.org/wiki/Successive-approximation_ADC)
- [2] <https://circuitdigest.com/article/how-does-successive-approximation-sar-adc-work-and-where-is-it-best-used>
- [3] <https://deepbluembedded.com/stm32-adc-tutorial-complete-guide-with-examples/>
- [4] <https://electronics-course.com/flash-adc>