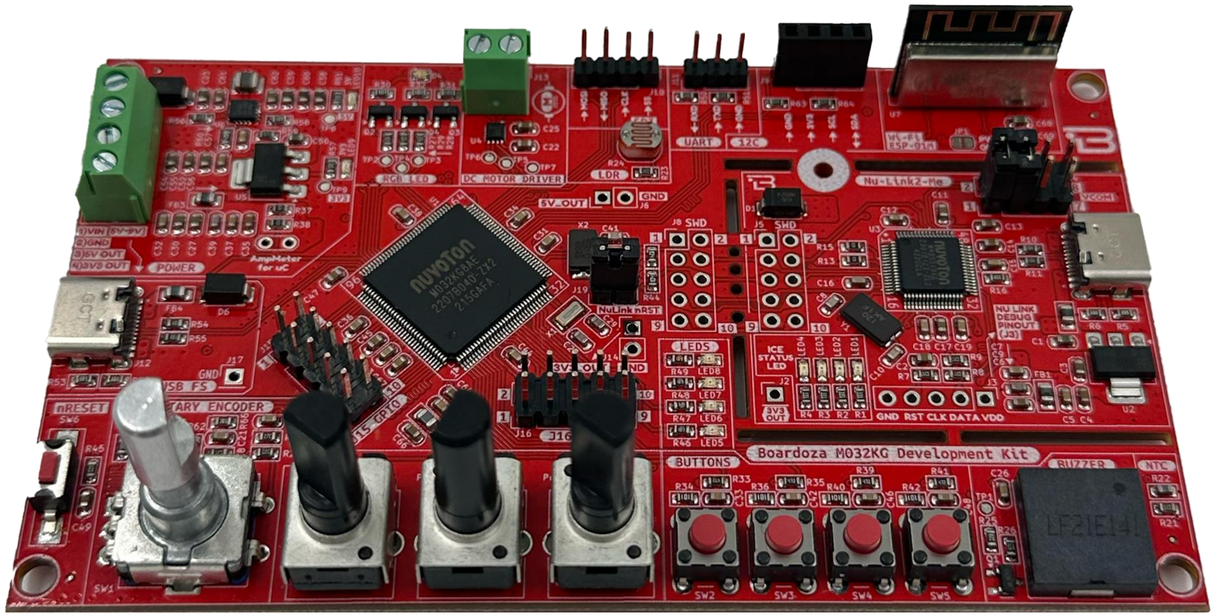
NUVOTON
M032KG BOARDOZA GELİŞTİRME KARTI

ADC – 1 UYGULAMA

Doç. Dr. Barış GÖKÇE

Mart 2025



BOARDOZA M032KG
Development Board

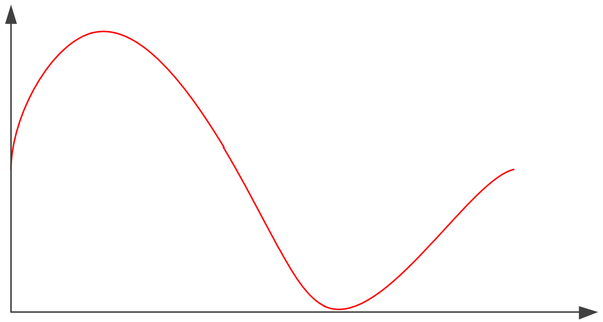
https://github.com/brgokce/NUVOTON-M032KG

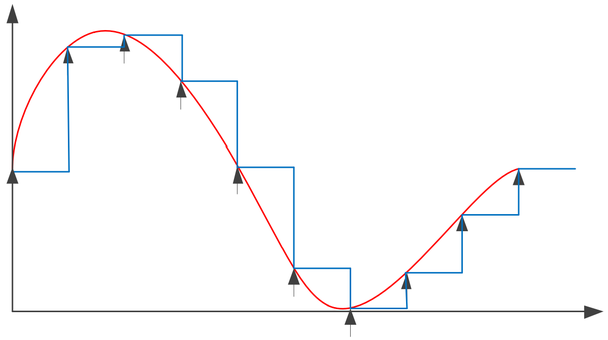
İçerik

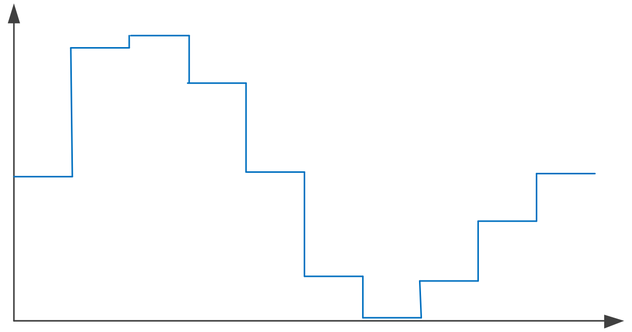
- ADC nedir?
- Nasıl dönüştürülür?
- Özellikler
- Blok Diyagram
- Fonksiyon
- AV_{DD} Örnek kodunu hesaplamak için BandGap'i kullanma

ADC Nedir?

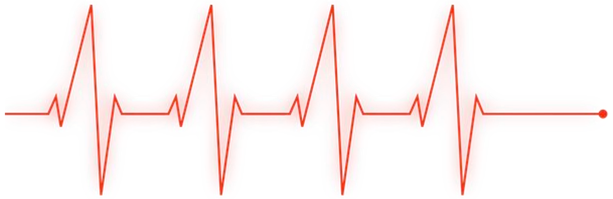
- Analog-dijital dönüştürücü(Analog-to-digital converter), voltaj veya akım gibi analog bir sinyali dijital sinyale dönüştürür.

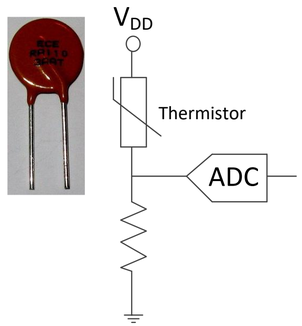


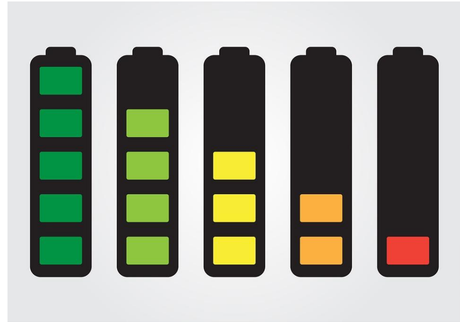




- ADC tıbbi cihazlarda, sıcaklık algılamada, pil voltajı algılamada ve diğer analog dönüşümlerde kullanılır.

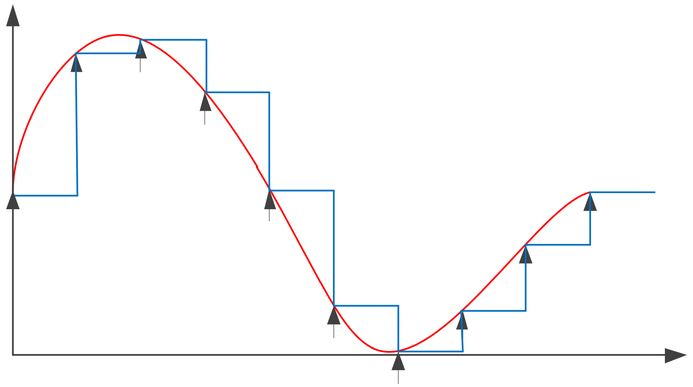


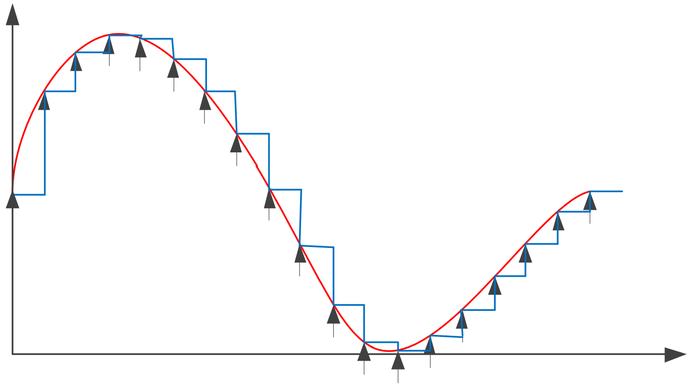


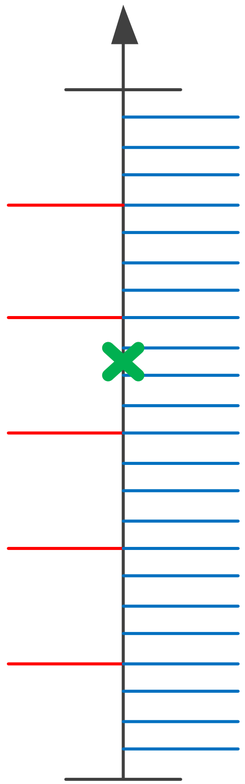


ADC Nedir?

- **Örnekleme oranı:**
  - Giriş analog sinyalini ne sıklıkla dönüştürmelisiniz?
  - Örnekleme oranı ne kadar yüksekse, dijital sinyal o kadar sürekli olur.



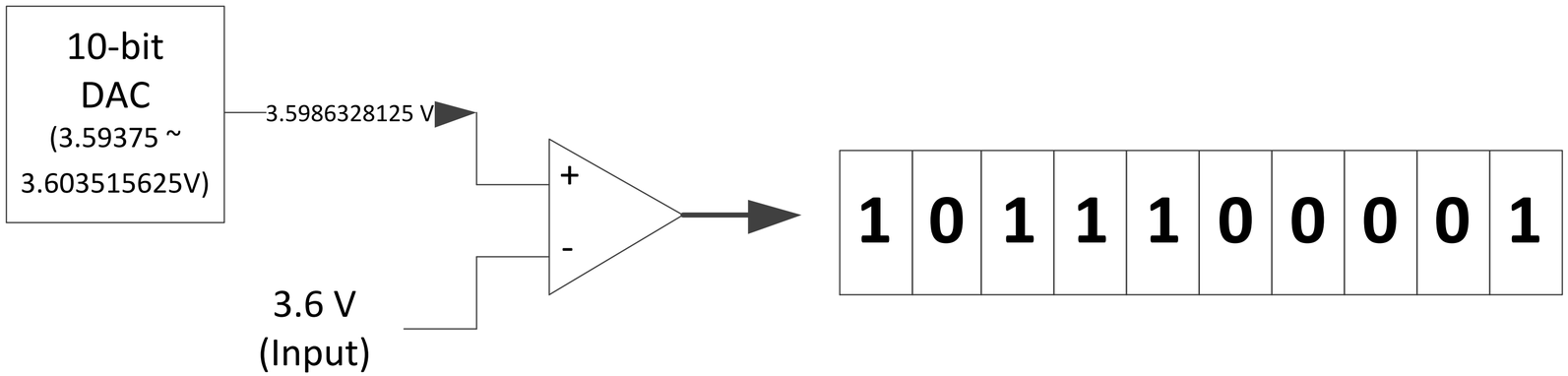




- **Çözünürlük:**
  - Dönüşüm aralığı için dijital değerin bit uzunluğudur.
  - Desteklenen çözünürlük ne kadar yüksekse, dijital sinyal o kadar doğru olur.

Nasıl Dönüştürülür?

- **Ardışık yaklaşım kaydı(Successive approximation register, SAR) ADC**
  - Giriş voltajını içeren bir aralığı art arda daraltmak için bir karşılaştırıcı kullanır.
  - Dönüştürücü, her ardışık adımda giriş voltajını, seçili voltaj aralığının orta noktasını temsil edebilecek dahili bir dijital-analog dönüştürücünün çıkışıyla karşılaştırır.



ADC'nin Özellikleri

- **10-bit doğruluk garantili 12-bit SAR ADC**
- **Analog giriş voltaj aralığı:**
  - 0 ~ V_{REF} (V_{REF} ≦ AV_{DD} ≦ 3.6V), V_{REF} pini varsa
  - 0 ~ AV_{DD} , V_{REF} pini yoksa
- **ADC çalışma voltajı:**
  - AV_{DD} = 1.8V ~ 3.6V
- **Giriş kanalı:**
  - 16 tek uçlu analog giriş kanalı
  - 8 tam diferansiyel analog giriş kanalı
- **2 MSPS'ye kadar örnekleme hızı**

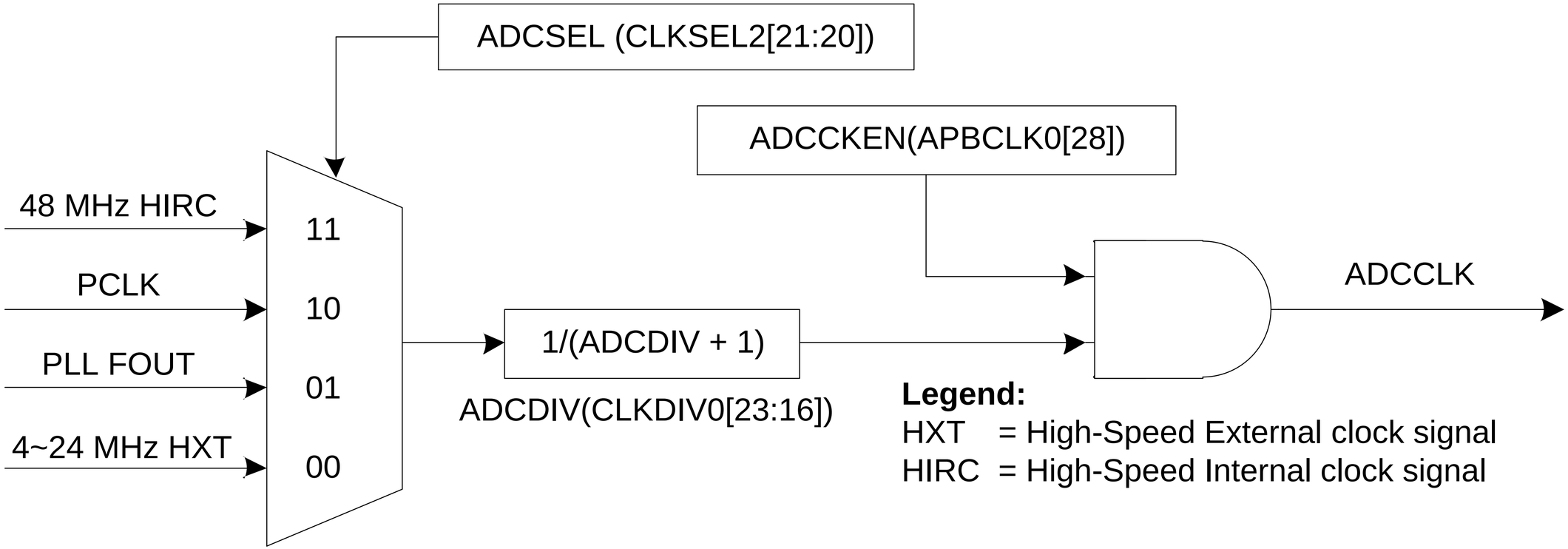
ADC'nin Özellikleri

- **Dört operasyon modu**
  - Tek mod
  - Burst (patlama) modu
  - Tek çevrim tarama modu
  - Sürekli tarama modu
- **Bir A/D şu şekilde başlatılır**
  - ADST register yazılımında 1 yazarak
  - Harici pin (STADC)
  - Timer 0~3 taşma tetikleyicisi
  - PWM tetikleyici

ADC'nin Özellikleri

- **Her dönüşüm sonucu, geçerli ve taşma göstergeleriyle her kanalın veri kaydında tutulur.**
- **Örnekleme süresini uzatma fonksiyonunu destekler. (0~255 ADC clock)**
- **Dahili giriş kaynağını destekler.**
  - Bant aralığı voltajı (V_{BG}) (Max 300 kSPS örnekleme oranı)
- **Karşılaştırma modunda sayaç değeriyle eşleştiğinde kesmeyi üretmeyi destekler.**
- **PDMA Fonksiyonunu destekler**

ADC Saat (Clock) Kaynağı

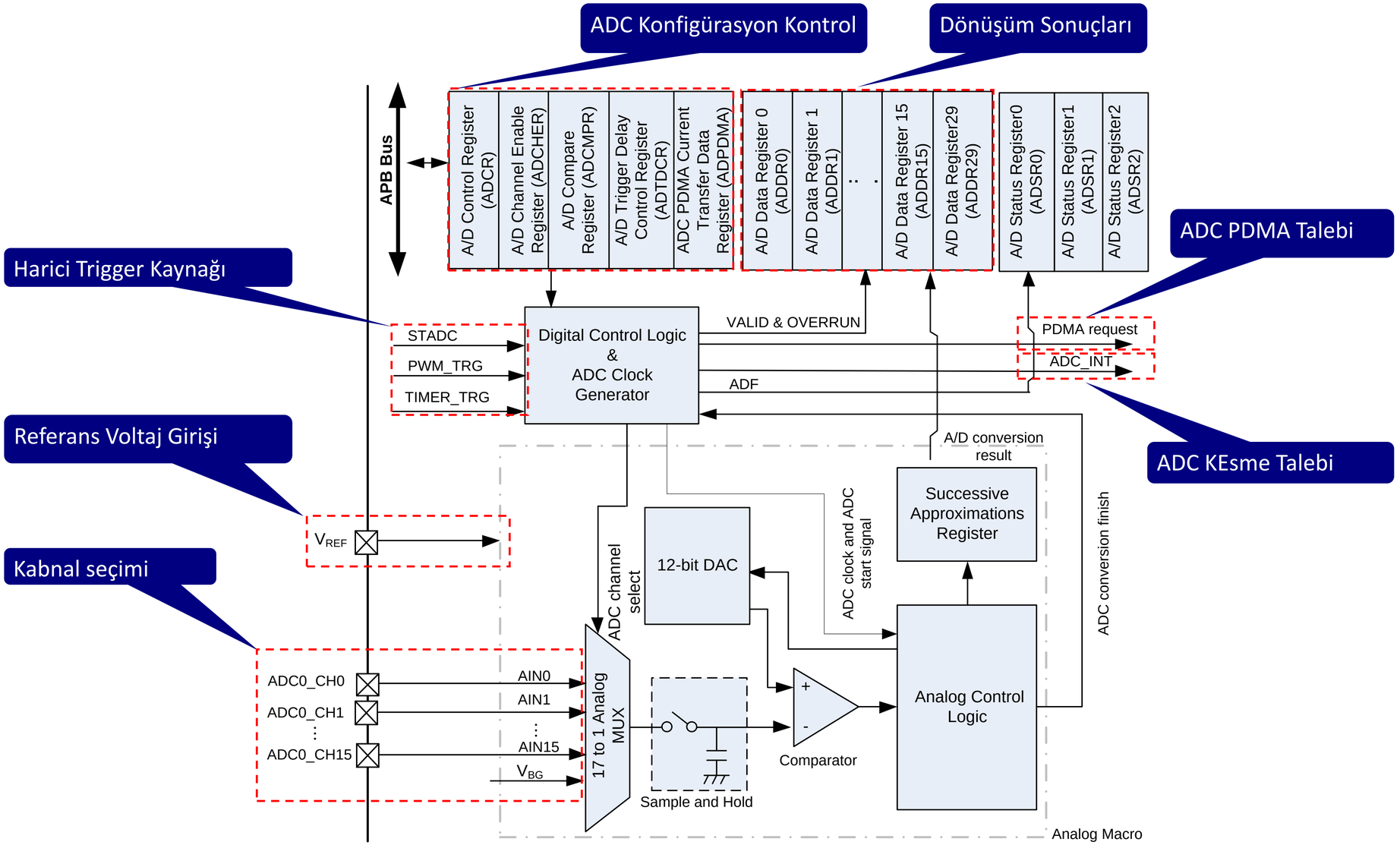


Maksimum ADCCLK frekansı 34 MHz'dir

Maksimum örnekleme hızı 2 MSPS'ye kadardır

ADC saat frekansı = (ADC çevresel saat kaynağı frekansı) / (ADCDIV+1)

ADC Blok Diyagramı



ADC Giriş Kanal Ayarları

- **ADC input pinleri GPIO ile paylaşılır**
- **ADC input pinleri input modu olarak yapılandırılmalıdır.**
- **Kullanıcı kaçak akımı önlemek için dijital giriş yolunu devre dışı bırakılır.**
- **Single-end input modu**
  - Channel 0 ~ Channel 15
- **Diferansiyel input modu**
  - Input voltajı (V_{diff}) = V_{plus} - V_{minus}

Differential Channel	V_{plus}	V_{minus}
0	Channel 0	Channel 1
1	Channel 2	Channel 3
2	Channel 4	Channel 5
3	Channel 6	Channel 7
4	Channel 8	Channel 9
5	Channel 10	Channel 11
6	Channel 12	Channel 13
7	Channel 14	Channel 15

MFP'yi Ayarla ve Dijital Giriş Yolunu Devre Dışı Bırak

◆ GPIO_DISABLE_DIGITAL_PATH

```
#define GPIO_DISABLE_DIGITAL_PATH (  port,
                                     u32PinMask
                                  )
```

Disable I/O Digital Input Path.

Parameters

[in]	**port**	GPIO port. It could be PA, PB, PC, PD, or PF.
[in]	**u32PinMask**	The single or multiple pins of specified GPIO port. It could be BIT0 ~ BIT15 for PA and PB. It could be BIT0 ~ BIT7, and BIT14 for PC. It could be BIT0 ~ BIT3, and BIT15 for PD. It could be BIT0 ~ BIT6, BIT14, and BIT15 for PF.

Returns

None

Disable I/O digital input path of specified GPIO pin.

```
/* Set 0 to input mode */
GPIO_SetMode(PB, BIT0, GPIO_MODE_INPUT);
/* Configure the PB.0 ADC analog input pin */
SYS->GPB_MFPL = (SYS->GPB_MFPL & ~SYS_GPB_MFPL_PB0MFP_Msk) | SYS_GPB_MFPL_PB0MFP_ADC_CH0;
/* Disable the PB.0 digital input path to avoid the leakage current */
GPIO_DISABLE_DIGITAL_PATH(PB, BIT0);
```

ADC Modülünde Güç

◆ ADC_POWER_ON

#define ADC_POWER_ON (adc)

Power on ADC module.

Parameters

[in] **adc** The pointer of the specified ADC module.

Returns

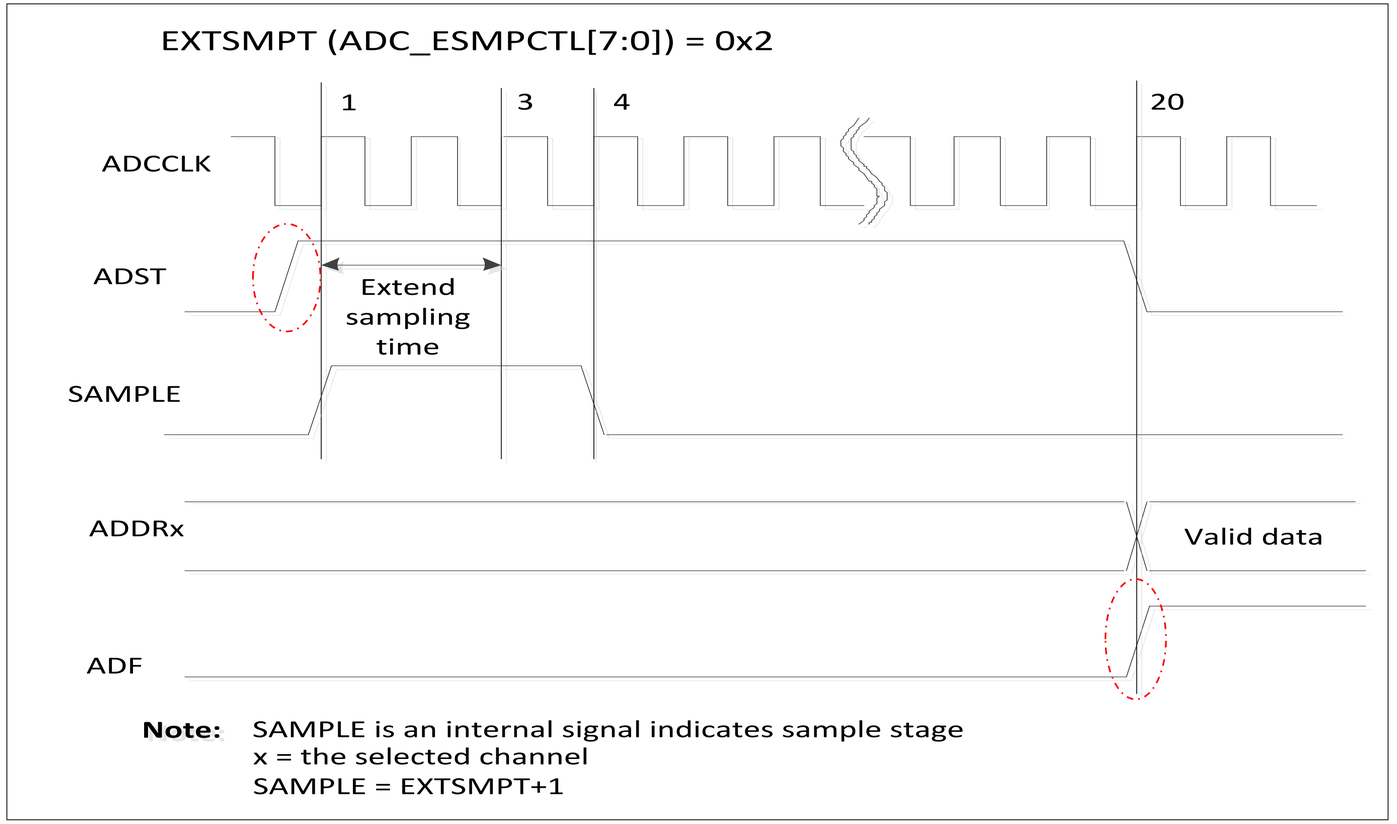
None

Before starting A/D conversion function, ADEN bit (ADCR[0]) should be set to 1.

```
/* Enable ADC converter */
ADC_POWER_ON(ADC);
```

Single Modu

- ADC, belirtilen tek kanalda **yalnızca bir kez** gerçekleştirilecektir.
- Yazılım tek modda birden fazla kanalı (örn. CH4, CH5, CH9) etkinleştirirse, **yalnızca en küçük sayı kanalı** (CH4) dönüştürülecek ve etkinleştirilen diğer kanallar göz ardı edilecektir.



ADC Kanalını Etkinleştir ve ADC Modunu Ayarla

Joy of innovation
nuvoTon

◆ ADC_Open()

```
void ADC_Open ( ADC_T * adc,
                uint32_t u32InputMode,
                uint32_t u32OpMode,
                uint32_t u32ChMask
              )
```

This API configures ADC module to be ready for convert the input from selected channel.

Parameters

`[in]` **adc** The pointer of the specified ADC module

`[in]` **u32InputMode** Decides the ADC analog input mode. Valid values are:

- **ADC_ADCR_DIFFEN_SINGLE_END** :Single-end input mode
- **ADC_ADCR_DIFFEN_DIFFERENTIAL** :Differential input mode

`[in]` **u32OpMode** Decides the ADC operation mode. Valid values are:

- **ADC_ADCR_ADMD_SINGLE** :Single mode.
- **ADC_ADCR_ADMD_BURST** :Burst mode.
- **ADC_ADCR_ADMD_SINGLE_CYCLE** :Single cycle scan mode.
- **ADC_ADCR_ADMD_CONTINUOUS** :Continuous scan mode.

`[in]` **u32ChMask** Channel enable bit. Each bit corresponds to a input channel. Bit 0 is channel 0, bit 1 is channel 1..., bit 15 is channel 15.

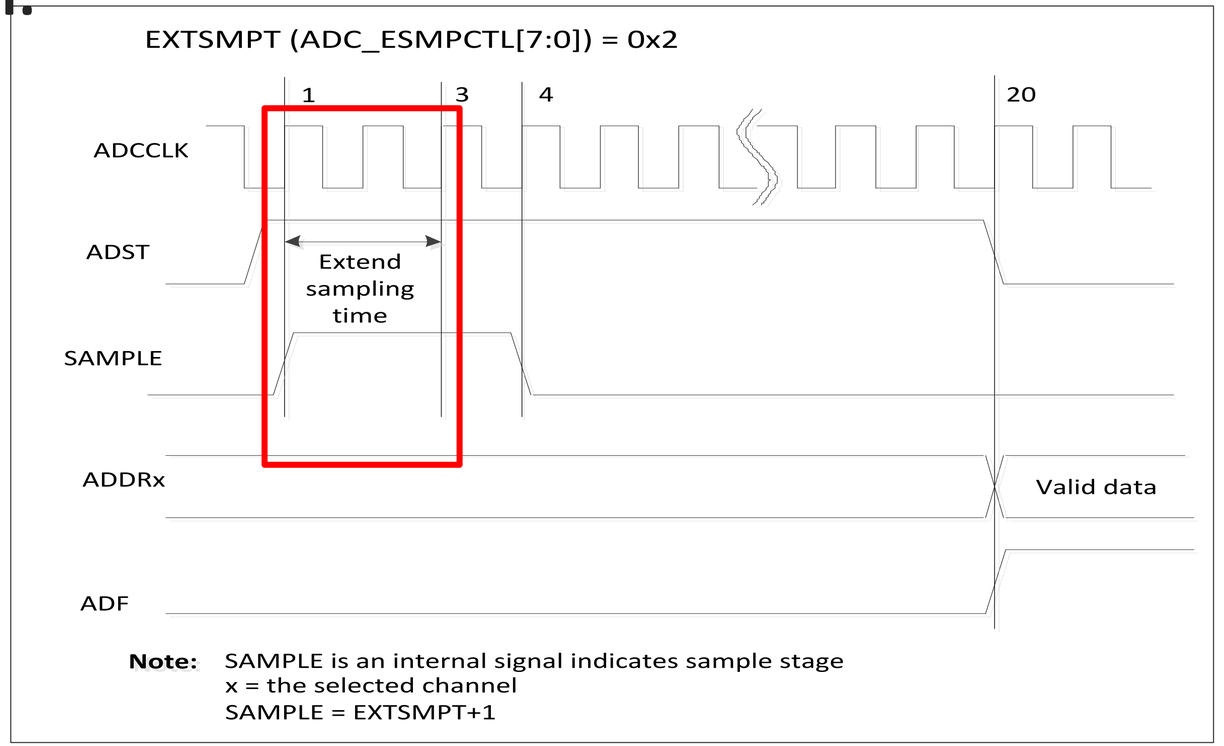
Returns

None

```
/* Set input mode as single-end, Single mode, and select channel 0 */
ADC_Open(ADC, ADC_ADCR_DIFFEN_SINGLE_END, ADC_ADCR_ADMD_SINGLE, BIT0);
```

Örnekleme Süresini Uzat

- Yüksek ADC saat hızında çalışırken, analog kanalda tam şarj süresinin daha uzun olmasına neden olacak ağır yükleme varsa örnekleme süresi yeterli olmayabilir.
- Kullanıcı ayrıca her örnek modülü için EXTSMPT (ADC_ESMPCTL[7:0]) yazarak örnekleme süresini uzatmayı ayarlayabilir. Örnekleme süresini uzatma aralığı 0 ~255 ADC saatidir.



Örnekleme Süresini Uzatmayı Ayarlama

Joy of innovation
nuvoTon

◆ ADC_SetExtendSampleTime()

```
void ADC_SetExtendSampleTime ( ADC_T *  adc,
                               uint32_t  u32ModuleNum,
                               uint32_t  u32ExtendSampleTime
                             )
```

Set ADC extend sample time.

Parameters

[in]	**adc**	The pointer of the specified ADC module.
[in]	**u32ModuleNum**	Decides the sample module number, valid value are 0.
[in]	**u32ExtendSampleTime**	Decides the extend sampling time, the range is from 0~255 ADC clock. Valid value are from 0 to 0xFF.

Returns

None

```
/* Set extend sampling time */
ADC_SetExtendSampleTime(ADC, 0, 2);
```

ADC Dönüşümünü Tetikle

- ADST bitine 1 yaz
- Harici pin (STADC)
- Zamanlayıcı 0~3 taşma tetikleyicisi
- İsteğe bağlı başlatma gecikme süresine sahip PWM tetikleyicisi

ADC Yazılım Tetikleyici

◆ ADC_START_CONV

#define ADC_START_CONV (adc)

Start the A/D conversion.

Parameters

[in] **adc** The pointer of the specified ADC module.

Returns

None

Set ADST bit to 1 to start the A/D conversion.

```
/* Trigger ADC to start A/D conversion */
ADC_START_CONV(ADC);
```

Dönüşüm Verilerini Okuma

◆ ADC_GET_CONVERSION_DATA

```
#define ADC_GET_CONVERSION_DATA (  adc,
                                   u32ChNum
                                )
```

Get conversion data of specified channel.

Parameters

`[in]` **adc** The pointer of the specified ADC module.

`[in]` **u32ChNum** ADC Channel, valid value are from 0 to 15 and 29.

Returns

16-bit data.

Read RSLT bit field to get conversion data.

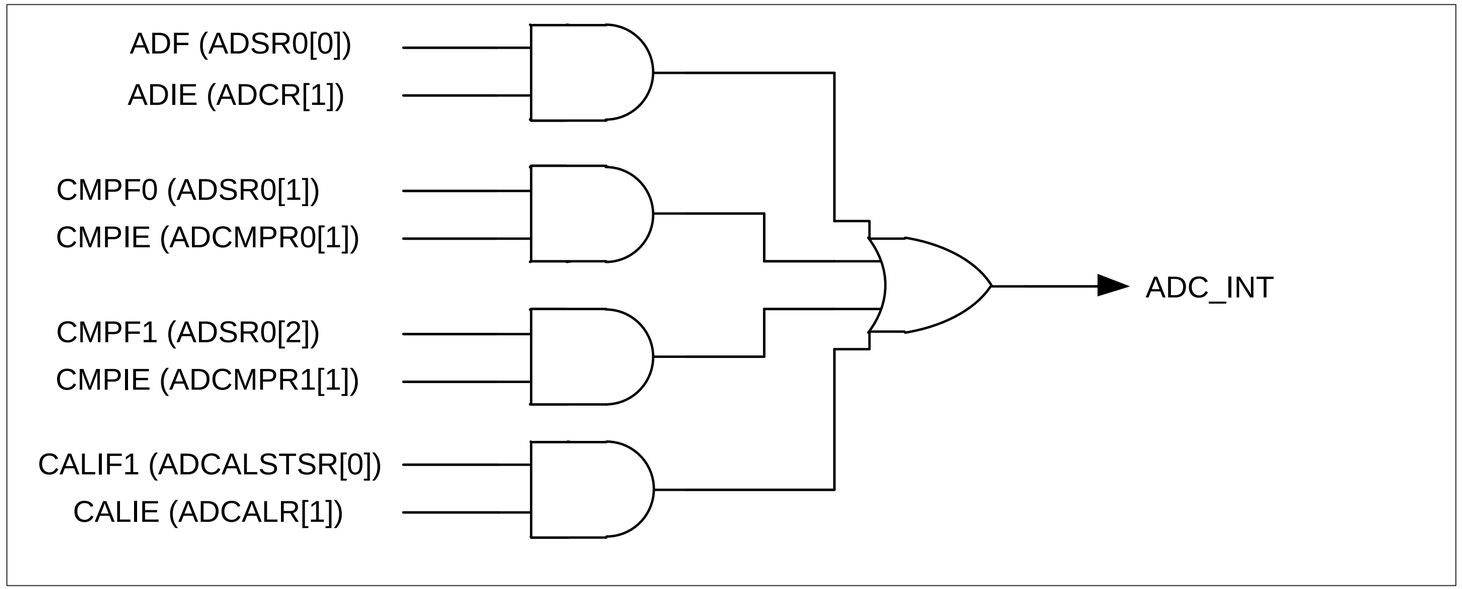
```
/* Get the conversion result of the channel 0 */
i32ConversionData = ADC_GET_CONVERSION_DATA(ADC, 0);
```

Dahili Referans Gerilmi

- Bant aralığı voltaj referansı (VBG), dahili sabit referans voltajıdır (CH29).
- Kullanıcı pil gücü algılama uygulaması için, VBG'yi ADC giriş kanalı olarak kullanabilir, böylece kullanıcı ADC sonucunu aşağıdaki formülle AVDD'yi hesaplamak için dönüştürebilir.
- $AV_{DD} = ((2 \text{ ^ } N) / R) * V_{BG}$
  - N: ADC çözünürlüğü
  - R: A/D dönüşüm sonucu
  - V_{BG}: bant aralığı voltajı
- Örneğin : N = 12, R = 1638 ve V_{BG} = 1.2 V
  - $AV_{DD} = ((2 \text{ ^ } 12) / 1638) * 1.2 = (4096 / 1638) * 1.2 = 3.0$ V

Kesme Kaynakları

- **Dört kesme kaynağı**
  - ADF: ADC çalışma modunun tamamlanması
  - CMPF0: ADC comparator0 tarafından izlenen koşul eşleşmesi
  - CMPF1: ADC comparator1 tarafından eşlenen koşul eşleşmesi
  - CALIF1 : Kalibrasyon modu tarafından izlenen koşul eşleşmesi



BSP'de ADC Örnek Kodu

- **\SampleCode\StdDriver**
  - ADC_SingleMode
  - ADC_BurstMode
  - ADC_SingleCycleScanMode
  - ADC_ContinuousScanMode
  - ADC_ResultMonitor
  - ADC_2MSPS_ContinuousScanMode
  - ADC_1411kSPS_ContinuousScanMode
  - ADC_BandGap
  - ADC_BandGapCalculateAVDD
  - ADC_SwTrg_Trigger
  - ADC_STADC_Trigger
  - ADC_Timer_Trigger
  - ADC_PWM_Trigger
  - ADC_PDMA_PWM_Trigger
  - ADC_ADINT_Trigger

Örnek Kod

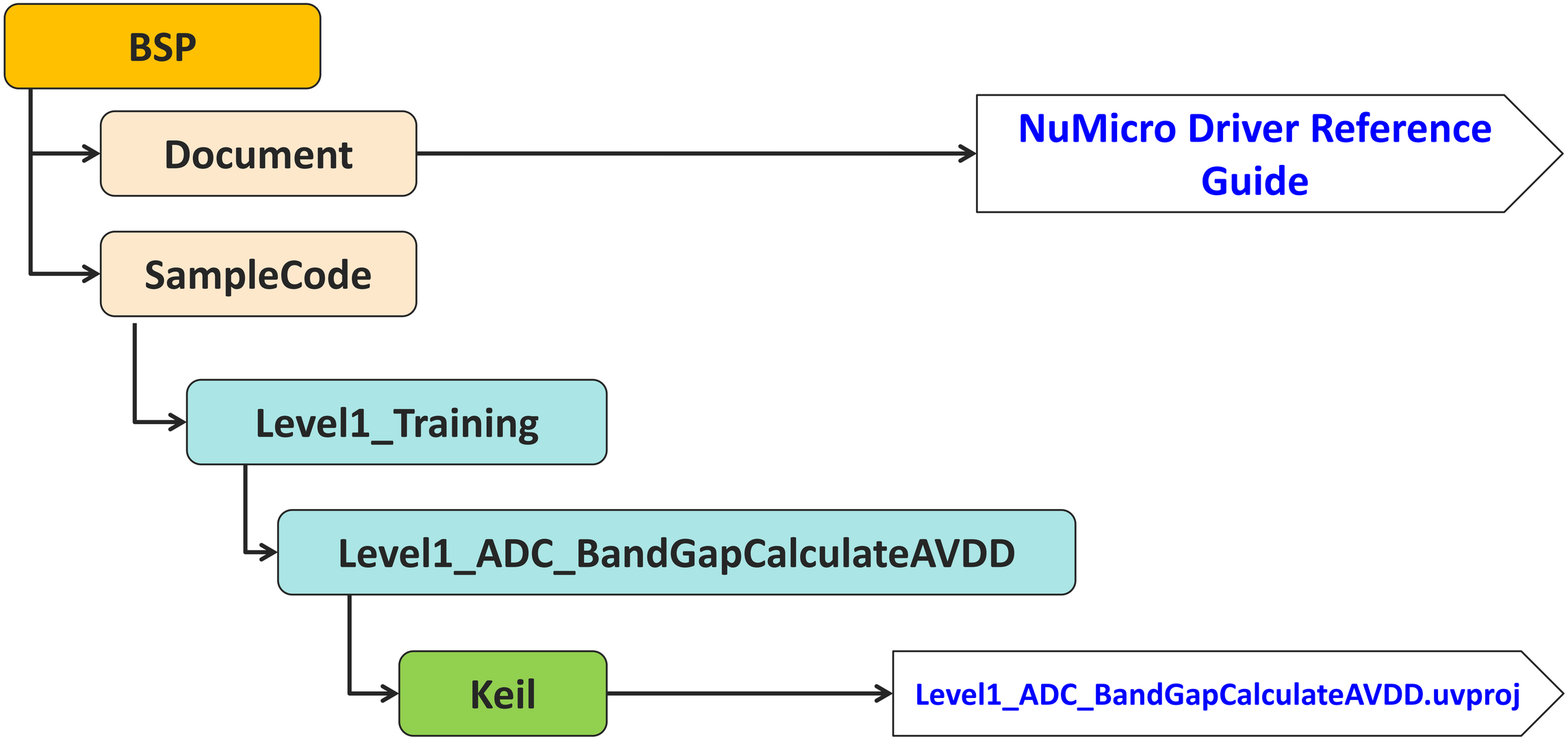
AVDD'yi hesaplamak için BandGap'i kullanın

AV_{DD} Örnek Kodunu Hesaplamak için BandGap'i Kullanın

- V_{REF} = 3072 mV olduğunda V_{BG} ADC dönüşüm sonucu (BuiltInData) okumak için **FMC_ReadBandGap** kullanın.
- V_{BG} ADC dönüşüm sonucunu (ConversionData) geçerli V_{REF} = AV_{DD} voltajındayken alın.
- Mevcut AV_{DD} voltajını hesaplamak için AV_{DD} = 3072 * (BuiltInData / ConversionData) formülünü kullanın.

```
+-------------------------------------------------------------------------+
|   ADC clock source -> PCLK1  = 48 MHz                                   |
|   ADC clock divider          = 2                                        |
|   ADC clock                  = 48 MHz / 2 = 24 MHz                      |
|   ADC extended sampling time = 71                                       |
|   ADC conversion time = 17 + ADC extended sampling time = 88            |
|   ADC conversion rate = 24 MHz / 88 = 272.7 ksps                        |
+-------------------------------------------------------------------------+
 Press any key to start the test

      AVdd          i32BuiltInData
   ---------- = -------------------------
      3072         i32ConversionData

AVdd =  3072 * i32BuiltInData / i32ConversionData

Built-in band-gap A/D conversion result: 0x652 (1618)
Conversion result of Band-gap:           0x653 (1619)

AVdd = 3072 * 1618 / 1619 = 3070 mV
```

Örnek-Yol


BSP
Document
NuMicro Driver Reference Guide
SampleCode
Level1_Training
Level1_ADC_BandGapCalculateAVDD
Keil
Level1_ADC_BandGapCalculateAVDD.uvproj

AV_{DD} Örnek Kodunu Hesaplamak için BandGap'i Kullanın

```
int32_t main(void)
{
      /* Init System, IP clock and multi-function I/O. */
      SYS_Init();
      /* Init UART0 for printf */
      UART0_Init();
      printf("\nSystem clock rate: %d Hz", SystemCoreClock);

      /* ADC function test */
      ADC_FunctionTest();

      /* Disable ADC IP clock */
      CLK_DisableModuleClock(ADC_MODULE);
      /* Disable External Interrupt */
      NVIC_DisableIRQ(ADC_IRQn);
      printf("Exit ADC sample code\n");
      while(1);
}
```

AV_{DD} Örnek Kodunu Hesaplamak için BandGap'i Kullanın

```
void SYS_Init(void)
{
      ……
      /* Enable module clock */
      CLK_EnableModuleClock(ADC_MODULE);

      /* Switch ADC clock source to HIRC, set divider to 2, ADC clock is 48/2 MHz */
      CLK_SetModuleClock(ADC_MODULE, CLK_CLKSEL2_ADCSEL_PCLK1, CLK_CLKDIV0_ADC(2));
      ……
}
```

AV_{DD} Örnek Kodunu Hesaplamak için BandGap'i Kullanın

```
void ADC_FunctionTest()
{
    ……
    /* Enable ADC converter */
    ADC_POWER_ON(ADC);
    /* Set input mode as single-end, Single mode, and select channel 29 (band-gap voltage) */
    ADC_Open(ADC, ADC_ADCR_DIFFEN_SINGLE_END, ADC_ADCR_ADMD_SINGLE, BIT29);

    /* To sample band-gap precisely, the ADC capacitor must be charged at least 3 us for charging the ADC capacitor ( Cin )*/
    /* Sampling time = extended sampling time + 1 */
    /* 1/24000000 * (Sampling time) = 3 us */
    ADC_SetExtendSampleTime(ADC, 0, 71);

    /* Clear the A/D interrupt flag for safe */
    ADC_CLR_INT_FLAG(ADC, ADC_ADF_INT);
    /* Enable the sample module interrupt.  */
    ADC_ENABLE_INT(ADC, ADC_ADF_INT);   // Enable sample module A/D interrupt.
    NVIC_EnableIRQ(ADC_IRQn);
    ……
```

ADC conversion time = 17 + ADC extended sampling time = 88

ADC conversion rate = 24 MHz / 88 = 272.7 kSPS

AV_{DD} Örnek Kodunu Hesaplamak için BandGap'i Kullanın

```
    /* Reset the ADC interrupt indicator and trigger sample module to start A/D conversion */
    g_u32AdcIntFlag = 0;
    ADC_START_CONV(ADC);

    /* Wait ADC conversion done */
    while(g_u32AdcIntFlag == 0);
    /* Disable the A/D interrupt */
    ADC_DISABLE_INT(ADC, ADC_ADF_INT);

    /* Get the conversion result of the channel 29 */
    i32ConversionData = ADC_GET_CONVERSION_DATA(ADC, 29);
    /* Enable FMC ISP function to read built-in band-gap A/D conversion result*/
    SYS_UnlockReg();
    FMC_Open();
    i32BuiltInData = FMC_ReadBandGap();

    ……

    printf("AVdd = 3072 * %d / %d = %d mV \n\n", i32BuiltInData, i32ConversionData, 3072*i32BuiltInData/i32ConversionData);
}
```

Joy of innovation
nuvoTon

SORULAR

Teşekkürler

Dr. Barış GÖKÇE
www.nuvoton.com