Bölüm 11
ADC ve DAC

Doç. Dr. Fatih Evran
Elektrik ve Elektronik Mühendisliği
Ofis: 421 Mühendislik Binası
Düzce Üniversitesi
Email: fatihevran@duzce.edu.tr

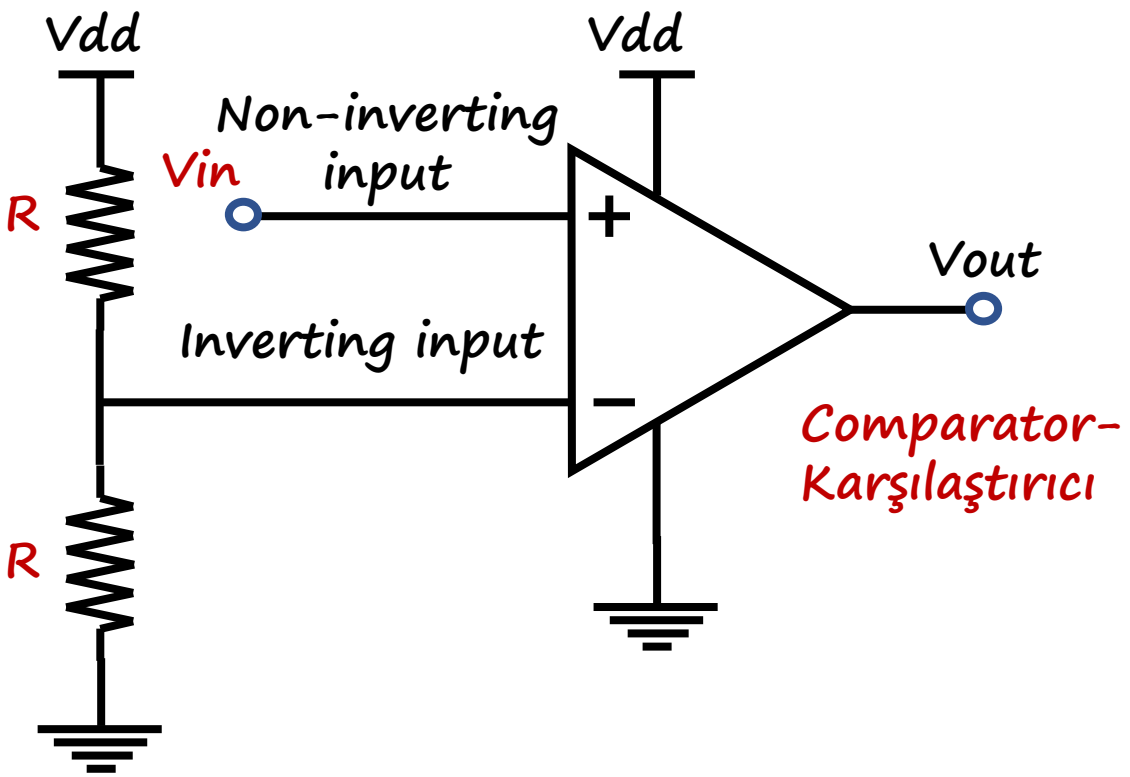
ADC ve DAC

- Analog-Dijital Dönüştürücü (ADC): Analog değeri (gerilim veya akım) ikili sayısal sisteme dönüştürür.
- Dijital-Analog Dönüştürücü (DAC): İkili sayısal sistemdeki değeri analog gerilime dönüştürür.
- ADC ve DAC uygulamaları:
    A. Kablosuz haberleşme
    B. Çevresel izleme
    C. Sayısal filtre
    D. Veri kaydı

Tanımlamalar

- Örnekleme periyodu (sample period): ADC için her dönüşüm arasındaki zamandır
- Referans gerilim (Vref): Analog sinyal 0 ve Vref arasında, ya da Vref- ve Vref+ arasında değişir
- Çözünürlük (Resolution): Veri dönüştürme için kullanılan bit sayısıdır (8 bit, 10 bit, 12 bit, 16 bit gibi)
- Dönüştürme süresi (Conversion Time): ADC için, analogdan dijitale dönüşüm için geçen süredir.

1-bit ADC



$$If\ V_{in} = V > \frac{V_{ref}}{2}, V_{out} = V_{dd}$$

$$If\ V_{in} = V < \frac{V_{ref}}{2}, V_{out} = 0$$

N-bit ADC

- N-bit ADC: ADC, bir analog örneği temsil etmek için N-bit kullanır
- Analog girişin genliği Vref' den büyük olamaz
- Giriş = 0 V olduğunda, ADC' nin (ADC kodu) çıkışı 0' dır
- Giriş = Vref olduğunda, ADC' nin (ADC kodu) çıkışı $(2^N - 1)$' dır.
- Herhangi bir analog giriş için ADC kodunu hesaplamak için aşağıdaki denklemi kullanabiliriz:

$$ADC\ Kodu = \left\lfloor \frac{V_{in}}{V_{ref}} \times 2^N \right\rfloor$$

$\lfloor x \rfloor$ is bir merdiven (kat) fonksiyonudur: Sonuç x' ten küçük veya x' e eşit olan en büyük tamsayıdır

Örnek: $\lfloor 8.4 \rfloor = 8$, $\lfloor 3.9 \rfloor = 3$, $\lfloor 10.1 \rfloor = 10$

N-bit ADC Örneği

10 bit bir ADC için referans gerilimin 4 V olduğunu varsayalım

Soru-1: Giriş 3 V ise çıkış ADC kodu nedir?

$$ADC\ Kodu = \left\lfloor \frac{V_{in}}{V_{ref}} \times 2^N \right\rfloor$$

$$= \left\lfloor \frac{3}{4} \times 2^{10} \right\rfloor = 768$$

Soru-2: Giriş 2.17 V ise çıkış ADC kodu nedir?

$$ADC\ Kodu = \left\lfloor \frac{V_{in}}{V_{ref}} \times 2^N \right\rfloor$$

$$= \left\lfloor \frac{2.17}{4} \times 2^{10} \right\rfloor = 555.52 \cong 555$$

ADC Kodundan Gerilime Geçiş

Çıkış ADC kodunu biliyorsak, aşağıdaki denklem giriş gerilimini elde etmemize yardımcı olabilir:

$$V_{in} = \frac{ADC\ Kodu}{2^N} \times V_{ref}$$

Örnek-1: 10 bit bir ADC için referans gerilimi 4 V ve çıkış ADC kodu 555 olarak verildiğine göre, ADC' nin giriş gerilimi nedir?

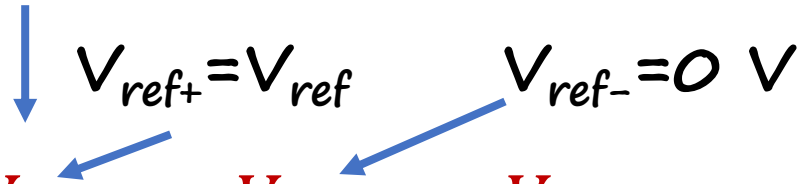
$$V_{in} = \frac{ADC\ Kodu}{2^N} \times V_{ref} = \frac{555}{2^{10}} \times 4\,V \approx 2.168\,V$$

Örnek-2: 10 bit bir ADC için referans gerilimi 4 V ve çıkış ADC kodu 768 olarak verildiğine göre, ADC' nin giriş gerilimi nedir?

$$V_{in} = \frac{ADC\ Kodu}{2^N} \times V_{ref} = \frac{768}{2^{10}} \times 4\,V = 3\,V$$

ADC Çözünürlüğü

- ADC' nin çözünürlüğünün iki tanımı olabilir:
  A. Veri dönüştürme için kullanılan bit sayısı (8 bit, 10 bit, 12 bit, 16 bit gibi)
  B. ADC tarafından ölçülebilen en küçük gerilim değişimi



$$V_{in} = \frac{V_{ref+} - V_{ref-}}{2^N} = \frac{V_{ref}}{2^N}$$

$V_{ref-} = 0$ V ve $V_{ref+} = V_{ref}$ olarak kullanacağız

Örnek: Referans gerilimi 4 V olduğunda 10 bit ADC' nin çözünürlük gerilimini (1 LSB) hesaplayın

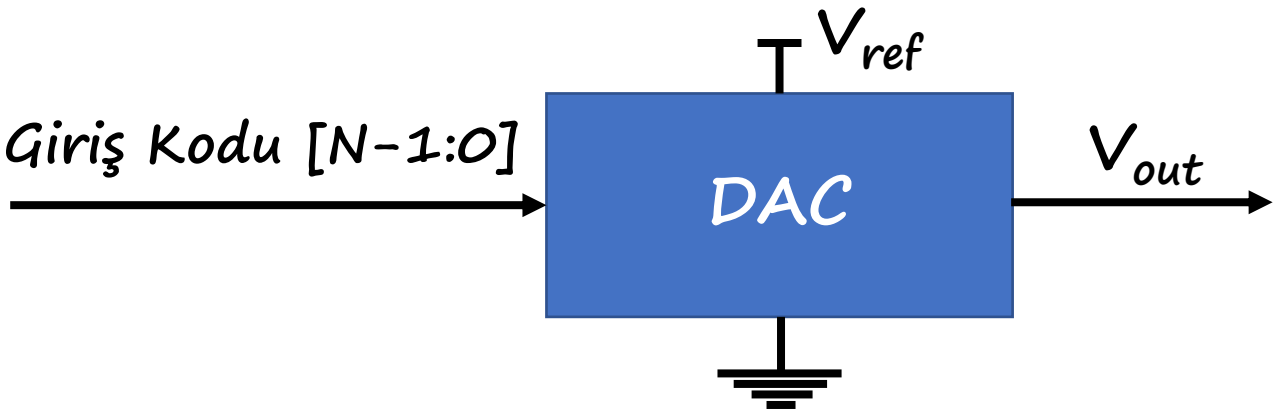
$$V_{in} = \frac{V_{ref}}{2^N} = \frac{4\,V}{2^{10}} = \frac{4\,V}{1024} \approx 3.9\,mV$$

Ticari ADC Entegreler

- 32 bit ADC mevcuttur
- Anahtar parametrelerden biri dönüştürme süresidir: Dönüştürme başlatıldıktan sonra doğru bir sayısal çıkışın üretilmesi ne kadar sürer?
- Dönüştürme süresine göre hızlı/orta/düşük hızlı ailelere ayrılmıştır
- Yüksek çözünürlüklü ADC' de sistem gürültüsü dönüşümü etkilemektedir
- Gürültü seviyesi, çözünürlük geriliminin yarısından az olmalıdır
- 4.1 V referans gerilimine sahip 16 bitlik ADC için çözünürlük gerilimi 4,1 V/216 = 62 µV' dir. Bu nedenle, 16 bit ADC kullanmak için gürültü seviyesinin 31 µV' den küçük olması gerekir

DAC

- DAC: Girişindeki ikili sayısal kodu çıkışında analog gerilime dönüştürür



- $0 \leq V_{out} < V_{ref}$ aralığındaki işaretsiz bir ikili giriş kodu kullanan bir N-bit DAC için:
  A. Giriş kodu = 0 olduğunda, DAC çıkışı 0 V olmaktadır
  B. Giriş kodu= $2^N - 1$ olduğunda, DAC çıkışı V_{ref} olmaktadır
  C. Herhangi bir giriş kodu verildiğinde DAC çıkışını hesaplamak için aşağıdaki denklemi kullanabiliriz:

$$V_{out} = \frac{DAC\ Kodu}{2^N} \times V_{ref}$$

N-bit DAC Örneği

8 bit bir DAC kullandığımızı varsayalım.

Soru 1: Vref = 5 V ve giriş DAC kodu 0x8A ise DAC çıkış gerilimi nedir?

$$V_{out} = \frac{DAC\ Kodu}{2^N} \times V_{ref} = \frac{0x8A}{2^8} \times 5\ V \approx 2.7\ V$$

Soru 2: Vref = 4 V ve çıkış gerilimi = 1.25 V ise giriş DAC kodu nedir?

$$DAC\ Kodu = \frac{V_{out}}{V_{ref}} \times 2^N = \frac{1.25}{4} \times 2^8 = 80 = 0x50$$

Ticari DAC Entegreler

- 32 bit DAC mevcuttur
- Çıkış sinyali gerilim veya akımdır. Akım DAC, gerilime dönüştürmek için harici bir işlemsel amplifikatöre (OPAMP) ihtiyaç duyar
- Anahtar parametrelerden biri yerleşme süresidir (settling time): Giriş kodu değiştikten sonra kararlı bir çıkış gerilimi için gereken süredir

PIC33 µC ADC

- PIC33 µC ADC' nin Özellikleri

  A. PIC33 µC bir yerleşik ADC' ye sahiptir

  B. 10-bit ya da 12-bit çözünürlük

  C. Referans gerilimi Vdd ya da ayrı bir gerilim olabilir

  D. Birden fazla giriş (birden fazla giriş kanalı)

  E. ADC için saat kaynağı, bölünmüş bir F_{OSC} (ana saat kaynağı) veya dahili olarak oluşturulmuş bir saattir (250 ns)

  F. ADC saat periyodu, 10 bit çözünürlük için 76 ns' den veya 12 bit çözünürlük için 118 ns' den az olamaz

Dönüştürme Süresi (Conversion Time)

- Dönüştürme Süresi: ADC için analogdan sayısala dönüştürme için geçen süredir
- Toplam dönüştürme süresi = Örnekleme süresi (sampling time) + Çözünürlük x 1 Tad + 2 Tads

A. 1 Tad = 1 ADC saat periyodu

B. Örnekleme süresi, ADC' nin örnekleme kondansatörünün (Csh) geriliminin dolması için gereken süredir.

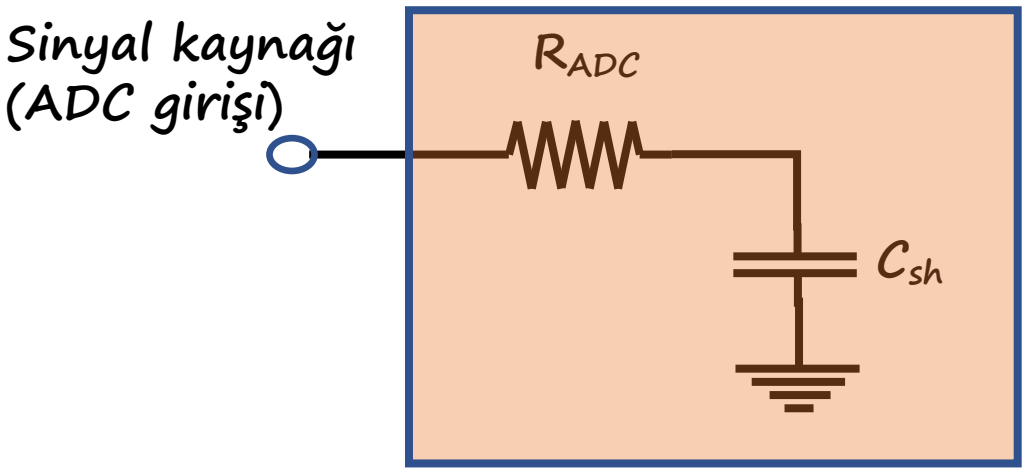
Dönüştürme Süresi (Conversion Time) (devam)

- Örnekleme süresi yapılandırılabilir
  A. Devre özelliklerine göre uygun bir örnekleme zamanı belirlememiz gerekiyor
  B. PIC33 µC için maksimum örnekleme süresi 31 Tad periyodudur

Dönüştürme süresi = Örnekleme süresi + Çözünürlük * 1 Tad + 2 Tad
= 31 Tad + Çözünürlük x 1 Tad +2 Tad

12 bit ADC

= 31 Tad + 12 x 1 Tad + 2 Tad
= 45 Tad = 45 saat periyodu



Gerilim Referansları

$$ADC\ Kodu = \frac{V_{in}}{V_{ref}} \times 2^N$$

- Gerilim referansının kararlılığı, yüksek hassasiyetli dönüşümler için kritik öneme sahiptir
- Kolaylık sağlamak için gerilim referansımız olarak Vdd' yi kullanacağız, ancak Vdd dalgalanmaları nedeniyle en az iki bit hassasiyeti kullanmayacağız
- Dönüştürme hassasiyetini artırmak için gerilim referansı bir entegreden alınabilir
    A. Gerilim referansının farklı seçenekleri vardır: 2.048 V, 2.5 V, 3 V, 3. 3 V, 4.096 V, 5 V
    B. PIC33 yalnızca 3. 0 V veya 3. 3 V değerinde bir gerilim referansı kullanabilir
    C. Gerilim referansı için anahtar parametre, sıcaklık çalışma aralığı üzerindeki kararlılıktır. Bunun ADC çözünürlüğünün yarısından az olması gerekiyor

ADC' yi Yapılandırma

```
void configADC1_ManualCH0(uint16_t u16_ch0PositiveMask,
          uint8_t u8_autoSampleTime,  uint8_t u8_use12bit) {
if (u8_autoSampleTime > 31) u8_autoSampleTime=31;
AD1CON1bits.ADON = 0; // turn off ADC
/** Configure the internal ADC **/
AD1CON1 = ADC_CLK_AUTO | ADC_AUTO_SAMPLING_OFF;
if (u8_use12bit)
      AD1CON1bits.AD12B = 1;
      else AD1CON1bits.AD12B = 0;
AD1CON3 = ADC_CONV_CLK_INTERNAL_RC | (u8_autoSampleTime<<8);
AD1CON2 = ADC_VREF_AVDD_AVSS;
AD1CHS0 = ADC_CH0_NEG_SAMPLEA_VREFN | u16_ch0PositiveMask;
AD1CON1bits.ADON = 1; //turn on the ADC
}
```

ADC' yi Yapılandırma (devam)

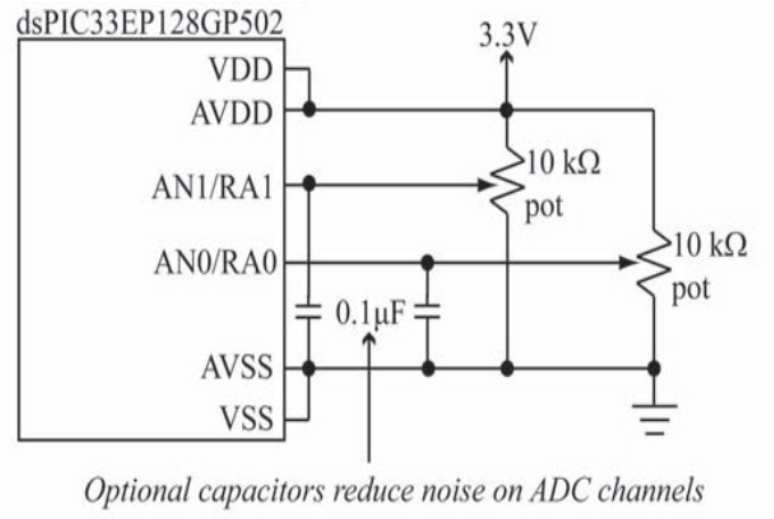
- Program dahili ADC saatini yapılandırır, manuel örnek başlatma/otomatik dönüştürmeyi kullanır
- u16_ChOPositiveMask dönüştürmek için Kanal O' dan ANx girişini seçer
- AVDD ve AVSS referans olarak kullanılır
- u8_autoSampleTime parametresi örnekleme süresini sayarlar
- u8_Use12bits dönüştürmenin 12 bit mi yoksa 10 bit mi yapıldığını belirler

Bir Dönüşüm Başlatma, Sonuç Alma:

```
uint16_t convertADC1(void) {
    uint8_t u8_wdtState;
    sz_lastTimeoutError = "convertADC1()";
    u8_wdtState = _SWDTEN;                    // save WDT state
    _SWDTEN = 1;                              // enable WDT since we block
    SET_SAMP_BIT_ADC1();                      // start sampling
    NOP();                                    // takes one clock to clear previous
                                              // DONE flag, delay before checking.
    WAIT_UNTIL_CONVERSION_COMPLETE_ADC1();    // wait for conversion to finish
    _SWDTEN = u8_wdtState;                    // restore WDT
    sz_lastTimeoutError = NULL;               // reset error message
    return(ADC1BUF0);
}
```

Bu programda, ADC' nin örneklemeyi başlatması istenir, örnekleme tamamlandıktan sonra ADC dönüştürmesi başlar ve dönüştürme bittiğinde bir durum biti birlenir (lojik '1').

ADC' yi Test Etme



Optional capacitors reduce noise on ADC channels

```
int main (void) {
 //Store ADC output code (12 bits)
        uint16_t  u16_pot1, u16_pot2;
//Store input analog voltage of ADC
        float   f_pot1, f_pot2;
//DEFAULT_BAUDRATE macro is 230,400
        configBasic(HELLO_MSG);
// Use AN0 as ADC input
        CONFIG_RA0_AS_ANALOG();
// Use AN1 as ADC input
        CONFIG_RA1_AS_ANALOG();
  while (1) {.....}
}
```

ADC' yi Test Etme (devam)

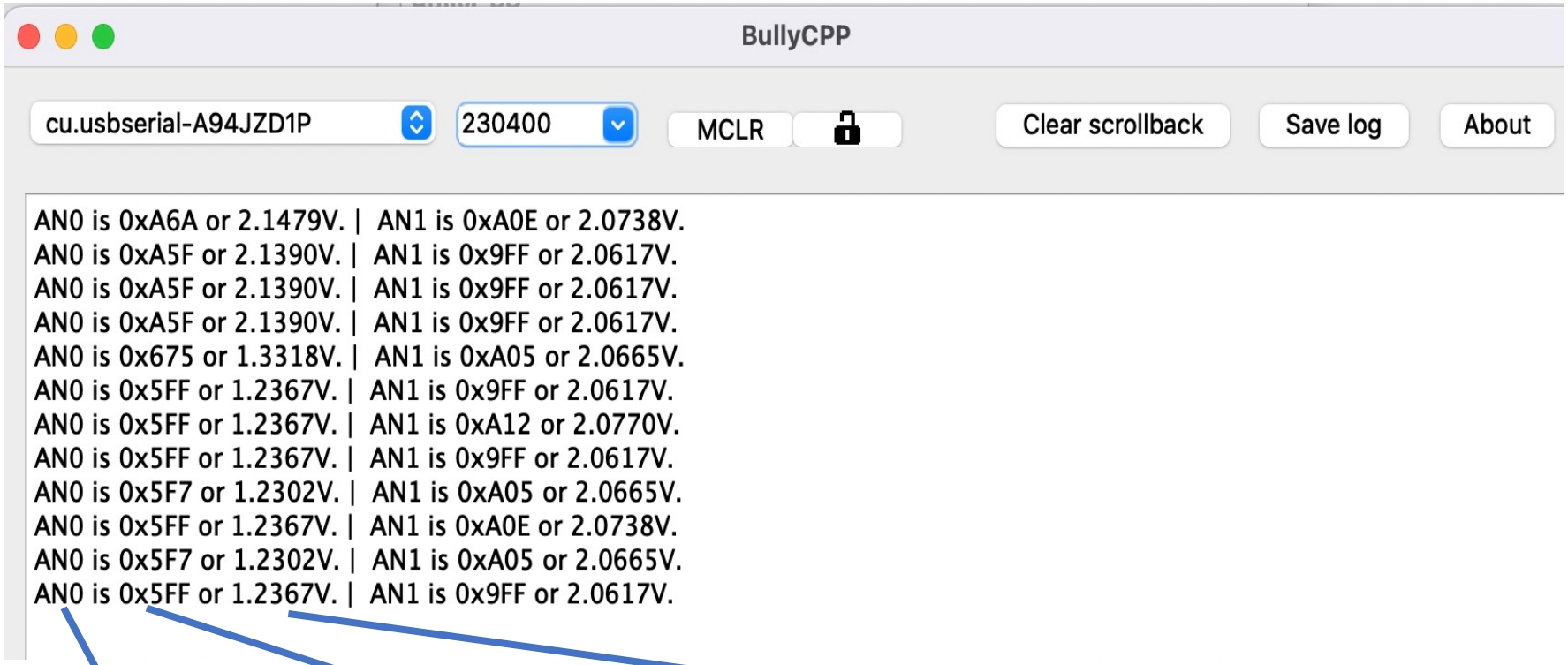
Use Channel 0 as ADC input

Sampling time=31 T_{AD}

0: ADC resolution =10 bits
1: ADC resolution =12 bits

```
while (1) {
    configADC1_ManualCH0(RA0_AN, 31, 1 );
    u16_pot1 = convertADC1();    // Get ADC output code
    configADC1_ManualCH0(RA1_AN, 31, 1 );
    u16_pot2 = convertADC1();    // Get ADC output code
    // Convert ADC output code to ADC input value
    f_pot1 = 3.30 / ADC_NSTEPS * u16_pot1;
    f_pot2 = 3.30 / ADC_NSTEPS * u16_pot2;
    printf("AN0 is 0x%0X or %1.4fV. |  AN1 is 0x%0X or
    %1.4fV.\n",    \
    u16_pot1, (double) f_pot1, u16_pot2, (double) f_pot2 );
    DELAY_MS(1500); // Delay certain so that we do not flood the UART
 } //endof while()
```

Program Çıkışı



ADC kanalı ADC çıkış kodu ADC giriş gerilimi

Sensörler

- Sensör, gerçek dünya niceliğini bir gerilime eşleyen bazı görevlere sahiptir. Üretilen gerilim fonksiyonu doğrusal veya doğrusal olmayabilir
- Doğrusal bir fonksiyon aşağıdakilerle karakterize edilir:

Gerilim = katsayı * gerçek dünya niceliği + ofset

Sensör girişi sıfır olduğunda çıkış gerilimi

Sensörler-Örnek1

Bir LM60 sıcaklık sensörü, her 1°C için 6.25 mV üretir ve 424 mV DC ofsetine sahiptir

Soru: Vref = 3.3V ve 10 bit bir ADC için, -10°C sıcaklık değerinde üretilen ADC kodu değeri nedir?

Çözüm:

Adım 1: Sıcaklık gerilime dönüştürülür

Gerilim (mV) = 6.25 mV * Tsantigrat + 424 mV
= 6.25 mV * (-10) + 424 mV
= 0.3615 V

Adım 2: Gerilim ADC koduna dönüştürülür

$$ADC\ Kodu = \frac{V_{in}}{V_{ref}} \times 2^N = \frac{0.3615}{3.3} \times 2^{10} = 112.17 \cong 112$$

Sensörler-Örnek2

Freescale MPX5050 basınç sensörü, 0 kPa' da 0.2 V çıkış vermektedir ve 90 mV/kPa hassasiyete sahiptir

Soru: Vref = 3.3 V olan 10 bit ADC, 420 ADC kod değerini vermektedir. Basınç sensörü tarafından algılanan basınç nedir?

Çözüm:

Adım 1: ADC kodu girişi gerilimine dönüştürülür

$$V_{in} = \frac{ADC\ Kodu}{2^N} \times V_{ref} = \frac{420}{2^{10}} \times 3.3\ V \approx 1.354V$$

Adım 2: Gerilim değeri basınca dönüştürülür

Basınç (kPa) = (gerilim (mV) – 200 mV) / 90 mV

= (1354 mV – 200 mV) / 90 mV = 12.8 kPa