

BLM206 Mikroişlemci Sistemleri ARM Uygulamaları

Öğr. Gör. Musa AYDIN

Ar. Gör. Ömer Faruk GÖKSU

Ar. Gör. Okan KARA

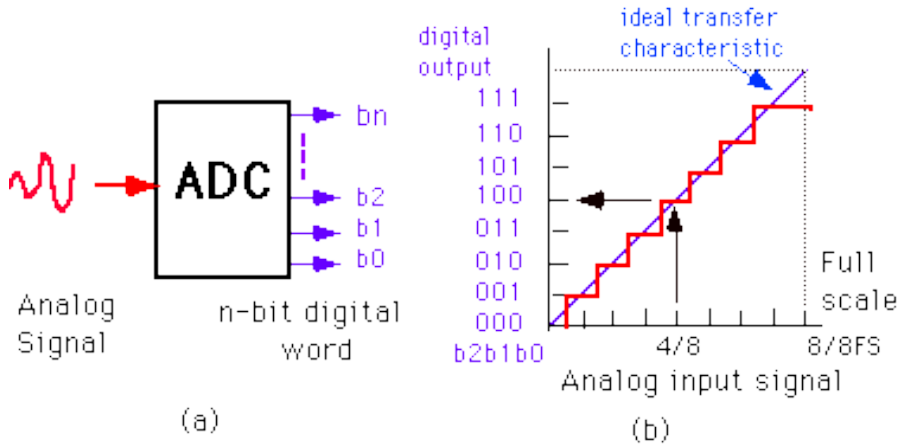
Uygulama 3 – STM32 ADC

Gerekli Malzemeler

- STM32 geliştirme kartı
- USB kablosu
- 3 x Led
- 3 x 560 ohm direnç
- Potansiyometre

Genel Bilgi

Mikrodenetleyicilerde ADC birimi, analog sinyali dijital veriye çevirmeye yaramaktadır. Dijital sistemlerde lojik 0 ve lojik 1 seviyeleri olmasına karşın, analog sinyaller teorik olarak sonsuz değerden oluşmaktadır. Bu değerleri mikrodenetleyicilerin anlayacağı şekilde dönüştürmek için kullanılan çevre birimlerinden biri de ADC birimidir.



ADC birimleri farklı çözünürlükte olabilmektedir. Bu çözünürlük, ADC ölçümünün hassasiyetini belirlemektedir.

- ADC için referans voltaj değerleri belirlenmelidir. Genel değerler, 3.3 V ve 5 V'tur. Referans voltaj değeri, ADC biriminin maksimum ölçüm yapabileceği analog sinyal sınırlarını belirlemektedir.
- Bit sayısı, referans voltaj değerini kaç parçaya böleceğimizi belirlemektedir. 3 bit ADC, referans voltaj değerini 8 eşit parçaya bölmektedir.
- ADC hassasiyeti, dijital değerde 1 birimlik değişme gerçekleşmesi için gereken analog sinyaldeki değişim miktarıdır.

Örnek verirsek, 10 bit çözünürlüğe sahip bir ADC biriminin referans voltajı 3.3 V olsun.

10 bit = 1023 maksimum değer.

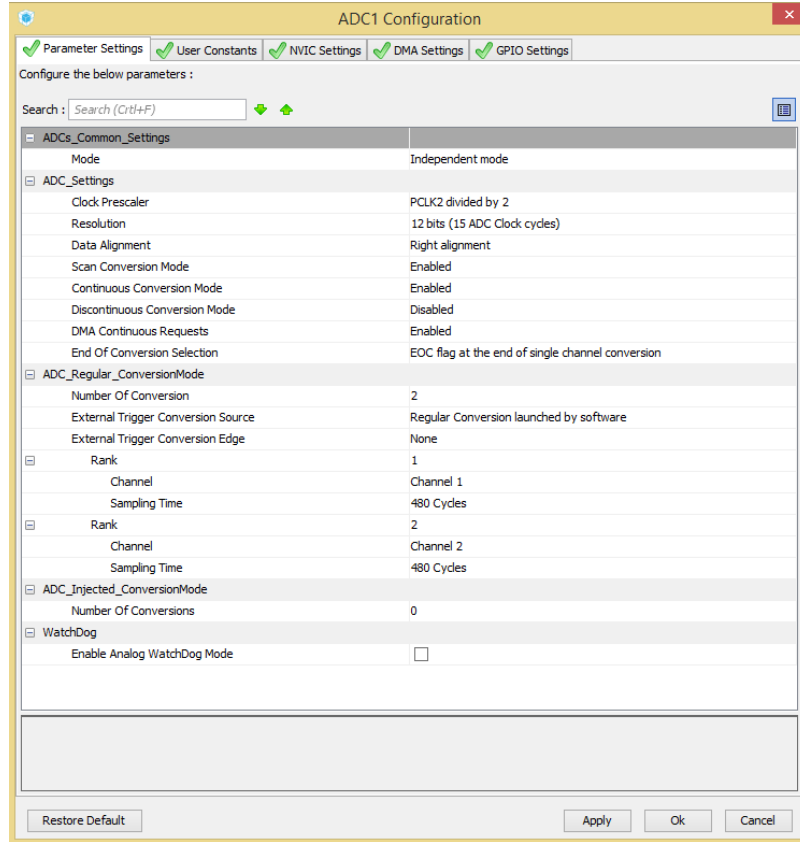
1 LSB= 3.3 V / 1023 = 0,0032258064516129 V hassasiyete sahiptir.

STM32 genel ADC özellikleri;

- Analog sinyal giriş aralığı = 0-3.6 V
- Referans Voltaj değeri = 2.4 V – 3.6 V
- DMA
- Interrupt



CubeMX ile yeni proje oluşturduktan sonra, ADC birimini aktif ediyoruz. İstedğimiz girişleri seçtikten sonra, configuration sekmesinden ADC birimine ait özellikler penceresini açıyoruz.

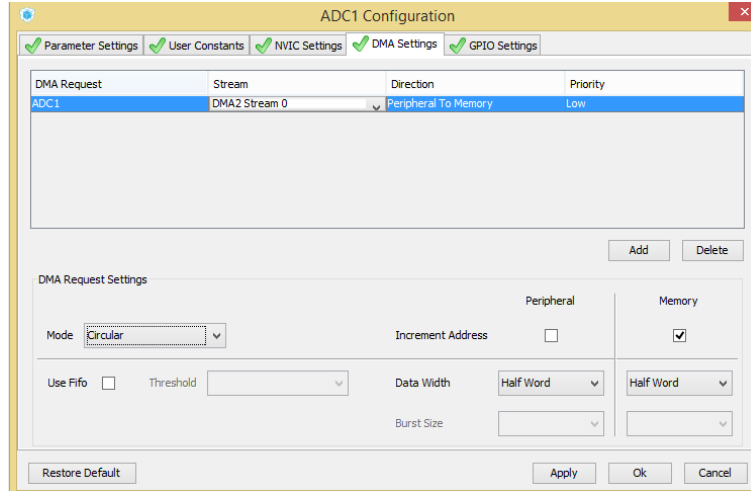


Burada karşımıza değiştirebilir özellikler çıkıyor. Bunlardan aktif ettiğimiz özelliklere kısaca değinirsek,

- Scan conversion: Birden fazla giriş seçildiği zaman, sırasıyla girişlerin hepsinde analog-dijital çevrim yapılacaktır.

- Continuous Conversion: Bu seçenek aktif edildiğinde, Analog-dijital çevrim işlemi durmadan sürekli tekrarlanacaktır. Bu seçenek aktif edilmezse, seçilen kanallar sırası ile çevrildikten sonra bir daha çevrim işlemi yapılmayacaktır.
- DMA request: Çevrimlerin DMA birimi yardımı ile yapılacağı belirlenmektedir. (DMA-Direct Memory Access)

Bu ayarlar yapıldıktan sonra,



ADC ayarlarından çıkmadan, DMA settings sekmesine gelerek, DMA birimini aktif ediyoruz. Ayarlarımızı şekilde görüldüğü gibi tamamlıyoruz. Circular seçeneği, DMA biriminin sürekli çevrim yapmasını sağlamaktadır.

Ayarları yaptıktan sonra, kodumuzu oluşturuyoruz ve Keil programına geçiş yapıyoruz.

```
/* USER CODE BEGIN PV */
/* Private variables -----
uint16_t adc_olcum[2];
/* USER CODE END PV */
```

Keil programı açıldıktan sonra, ADC ölçümlerinin saklanacağı bir değişken oluşturuyoruz. 2 kanal için, 2 elemanlı bir dizi tanımlaması yapılmıştır.

```
83 /* Initialize all configured peripherals */
84 MX_GPIO_Init();
85 MX_DMA_Init();
86 MX_ADC1_Init();
87
88 /* USER CODE BEGIN 2 */
89 HAL_ADC_Start_DMA(&hadc1, (uint32_t*) adc_olcum[0], 2);
90 /* USER CODE END 2 */
91
92 /* Infinite loop */
93 /* USER CODE BEGIN WHILE */
94 while (1)
95 {
96 /* USER CODE END WHILE */
97
98 /* USER CODE BEGIN 3 */
99
100 }
101 /* USER CODE END 3 */
102
103 }
104
```

ADC kütüphanesinden, HAL_ADC_Start_DMA fonksiyonu kullanılarak ADC çevrimi başlatılır. Bundan sonra, seçilen kanallar otomatik olarak çevrilerek adc_olcum dizisi içine kaydedilecektir.

Uygulama

Potansiyometreden okunan değere göre, ledler yanacaktır. Voltaj değeri 0-1 V arasında ise 1 led, 1-2 V arasında ise 2 led, 2 V ve üstünde ise 3 led yanacaktır.