

# VOLATILE

By Mustafa Onur Parlak

- **Değişkenler** ile birlikte kullanılan bir anahtar kelimedir.
- **Volatile** kullanılarak, derleyiciden bağımsız değişken değişimleri yapılabilir. **Yani** değişkenin çeşitli değişimlerine karşı derleyicinin **optimize** yapmasını engeller
  - Derleyicinin optimize yapması,
    - Bellek alanını verimli kullanmayı amaçlar
    - Üretilen akışı sade tutmayı hedefler
  - Sürekli olarak **orijinal değişken adresin-değerin** hatırlanması gereken döngüsel - fonksiyonel durumlarda, stabilizasyon sağlamak için kullanılmaktadır.
    - Örnek olarak,
      - 1) Interrupt Reset Handler fonksiyonu veya
      - 2) Interrupt Service Routine
  - Kullanılan yerler;
    - Çevrebirimlerine ait özel fonksiyon register'larında
    - Interrupt (Handler veya Service Routine'leri ) durumlarında kullanılan global değişkenlerde (veya bufferları threadler arası paylaşıırken)
    - Multitask uygulamalarında

## LÜTFEN AŞAĞIDAKİ YAZILIMI ÇALIŞTIRMAYINIZ

```
int main (void)
{
    int32_t i32Value = 0;
    int32_t *pi32Variable = (int *)MemoryAddress;

    while (1)
    {
        i32Value = *pi32Variable;

        if(i32Value) break;
    }
    return 0;
}

//*****
#define MemoryAddress 0x010101
int main (void)
{
    int32_t i32Value = 0;
    volatile int32_t *pi32vVariable = (int *)MemoryAddress;

    while (1)
    {
        i32Value = *pi32vVariable;

        if(i32Value) break;
    }
    return 0;
}
```

- **'const' vs 'volatile'**

- **Const:** Derleyici tarafından uygulanır ve değişken değerinin değişmeyeceğini söyler. Atama yapılırsa, derleme sırasında hata meydana getirir.
- **Volatile:** Derleme optimizasyonunu engeller ve program dışında da değişkenin değişebileceğini kabul eder. Her tekrarda değişkenin adresindeki değerini okur.

- **Bir Değişkenin volatile + const olma durumu var mı?**

- **Evet,** GPIO register erişimi örnek verilebilir. Genel Amaçlı Giriş/Çıkış durumlarında değer dış faktörler ile bağlantılı olarak değişmesi muhtemeldir. Bu durumda **volatile** anahtar kelimesi bilindiği üzere **değişkenin değerini** her zaman GPIO adresinden okumak ister ve bu şekilde ilgili port ile doğru değer etkileşimi sağlanmış olur.

**Sorun şu ki; pointer bir const değildir.** Bu sebeple program pointerın gösterdiği adresi değiştirebilir.

Bu sebeple **volatile ile const pointer** oluşturmalıyız.

```
/* cpvi32Variable bir 'constant'tır.  
 * Pointerdır  
 * Volatile  
 * Integer 32-bit gösterir  
 */  
int32_t volatile * const cpvi32Variable;
```

- **Volatile pointer olarak kullanılabilir.**