

SAVTEK 2018



Bir Açık Kaynak Kodlu Gerçek Zamanlı İşletim Sistemi (FreeRTOS) ile Gömülü Yazılım Geliştirme Çalışmaları

Alper Yazar

Sayısal Tasarım Uzman Mühendisi

ayazar@aselsan.com.tr

Savunma ve Sistem Teknolojileri (SST) Sektör Başkanlığı

ASELSAN

28 Haziran 2018

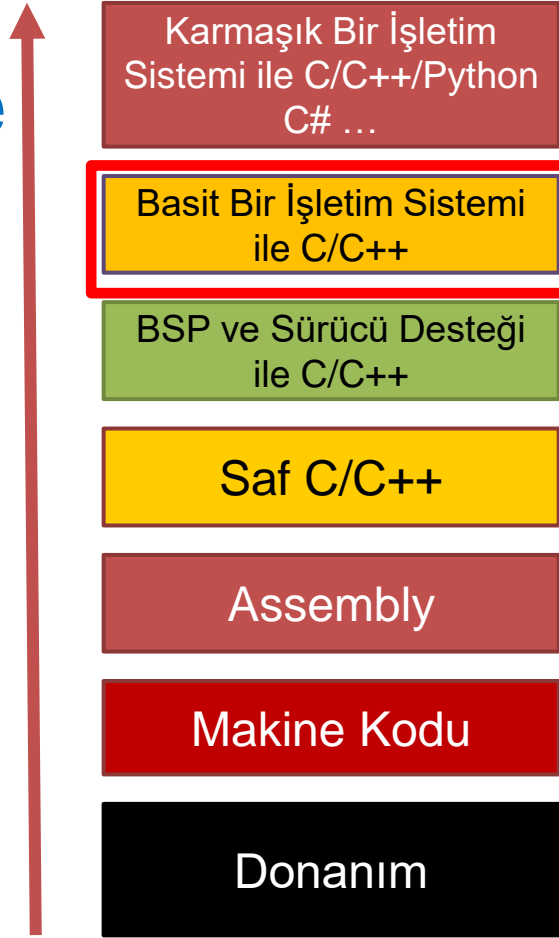
Sunum İeriđi

- Sunum Kapsamı
- Sektördeki İşletim Sistemi Tercihleri
- FreeRTOS Tanıtımı
- FreeRTOS Teknik Bileşenleri
- Artılar ve Eksiler
- Tavsiyeler
- Soru ve Cevap

Çalışma (Sunum) Kapsamı

Yüksek Seviye

Düşük Seviye
(Bare Metal)



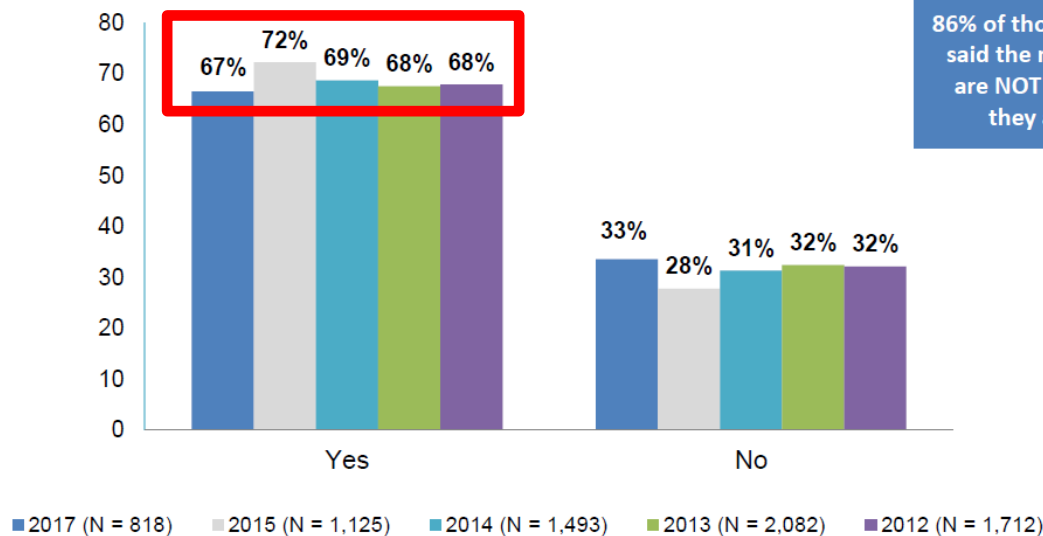
Seyrek ... Sık

Aspencore 2017 Embedded Markets Study



Does your current embedded project use an operating system, RTOS, kernel, software executive, or scheduler of any kind?

Fairly consistent usage of RTOS, kernels, execs, schedulers over past 5 years



86% of those not using RTOSes, said the main reason RTOSes are NOT used is simply that they are not needed.

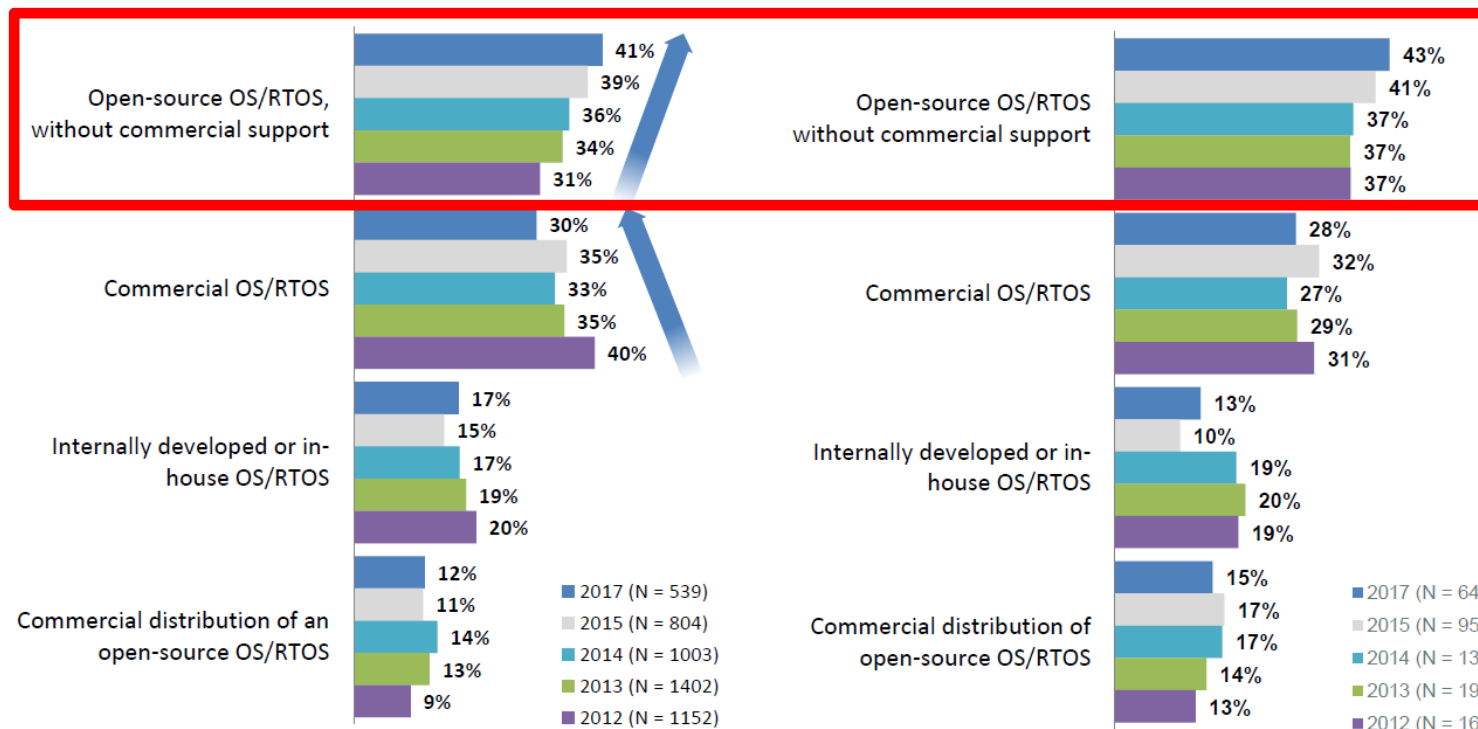
Aspencore 2017 Embedded Markets Study



ASPENCORE

My current embedded project uses:

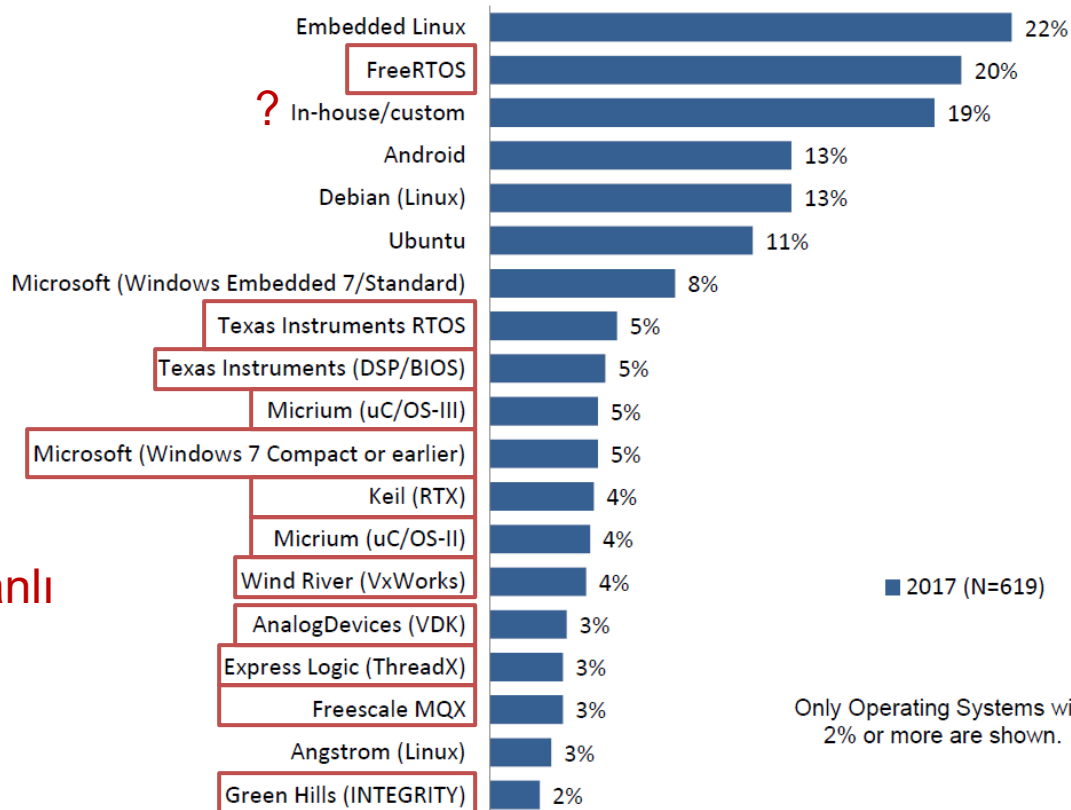
My next embedded project will likely use:



Aspencore 2017 Embedded Markets Study



Please select ALL of the operating systems you are currently using.



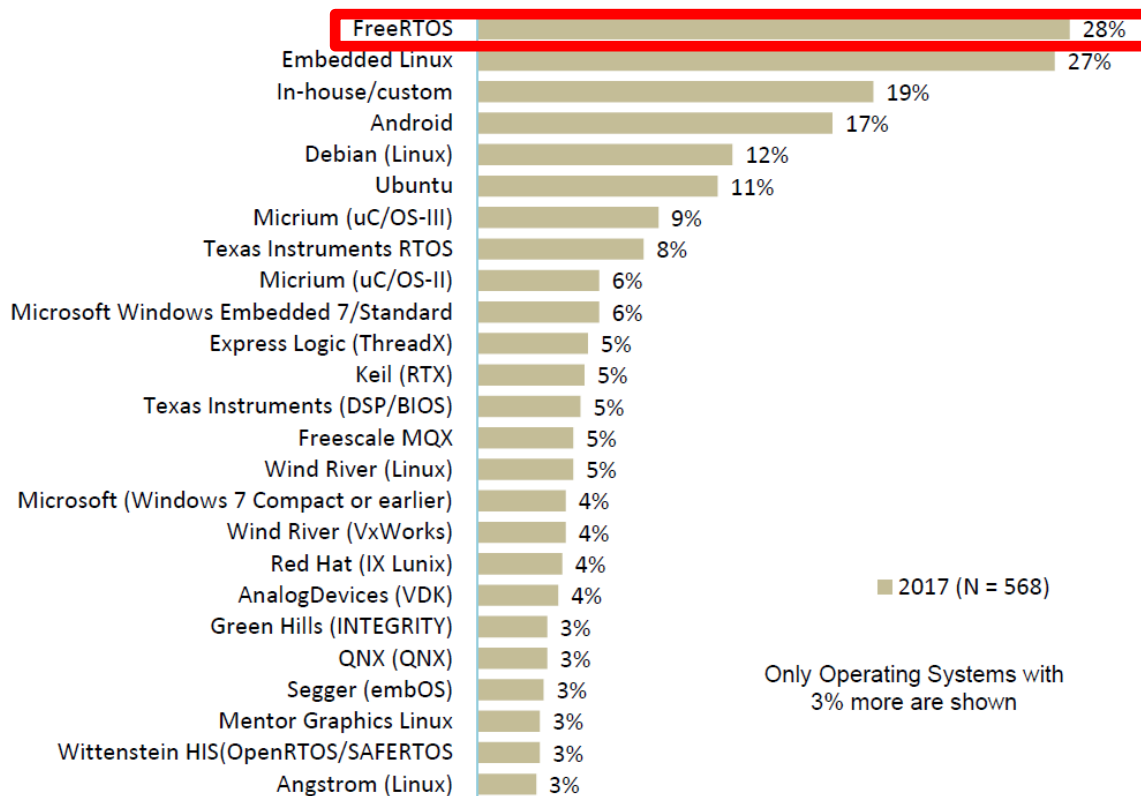
Gerçek Zamanlı

Base: Currently using an operating system

Aspencore 2017 Embedded Markets Study



Please select ALL of the operating systems you are considering using in the next 12 months.



Only Operating Systems with 3% more are shown

Base: Those who are considering an operating system in any project in the next 12 months



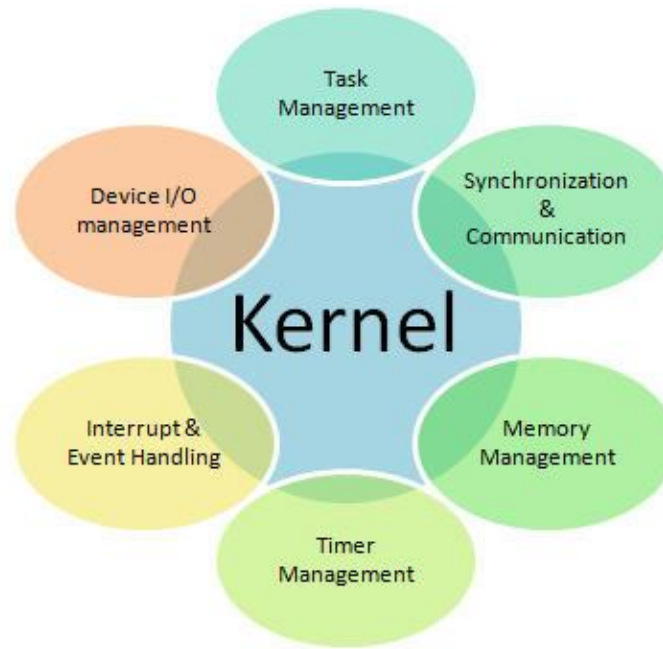
FreeRTOS

- 2003 yılından itibaren geliştirilen, güncel sürümü v10 olan açık kaynak kodlu, ücretsiz bir gerçek zamanlı işletim sistemidir.
- Yakın zamanda Amazon AWS tarafından desteklenmeye başlanmış (IoT platformu için) ve «Amazon FreeRTOS» adı ile sunulmaktadır.
- MIT lisansı ile sunulmaktadır. Kaynak kodunda kapalı değişiklikler yapılabilir.
- Düşük güçlü mikrokontrolcülerden yüksek performanslı işlemcilere kadar 35'ten farklı işlemci ailesini desteklemektedir.

FreeRTOS

- 6KB – 12KB arası hafıza izi, düşük hafızaya sahip platformlarda kullanımına imkan sağlamaktadır.
- FreeRTOS'un web sitesinden erişilebilen iyi hazırlanmış eğitim ve referans kaynaklar vardır. Her işlemci ailesi için örnek proje bulunmaktadır. **FreeRTOS, tek başına çalışılarak 1-2 hafta içerisinde öğrenilebilir.**
- Geliştiriciler ve topluluk tarafından forum üzerinden yeterli destek sağlanmaktadır. Ücretli destek için **OpenRTOS** düşünülebilir.

Temel İşletim Sistemi ve FreeRTOS Bileşenleri



Temel İşletim Sistemi ve FreeRTOS Bileşenleri

- Temel iki bileşen: **Scheduler** (Zamanlayıcı) ve **IPC** (Inter-Process Communication)
- Scheduler: **pre-emption** ve **time slicing** özellikleri konfigüre edilebiliyor.
- IPC: **queue**, **mutex**, **semaphore** yapıları sunuluyor.
- **Software Timers**: Periyodik veya tek seferlik zamanlı işlerin yapılması için kullanılabilir.
- **Heap Management**: Farklı senaryolar için optimize edilmiş 5 farklı heap yönetimi yöntemi sunulmaktadır.

Artılar ve Eksiler





Ekip Çalışmasını Kolaylaştırması

Farklı geliştiricilerin yazdığı kodların entegrasyonu, «bare metal» yazılımlara kıyasla FreeRTOS mekanizmaları (Scheduler, IPC vb.) ile daha kolay entegre edilmiştir.



Kısıtlı Donanım Sayısı ile Geliştirme Yapmanın Kolaylaşması

Platformlar (kartlar) arasındaki kod taşımayı kolaylaştırması ve Windows'ta çalışan emülatör bulunması.



Donanımdan Soyutlanabilme

ISR kodlarının kısmi olarak FreeRTOS tarafından sağlanması, «bare metal» yazılımcıları için «yeteri kadar» soyutlanma imkanı sunar.



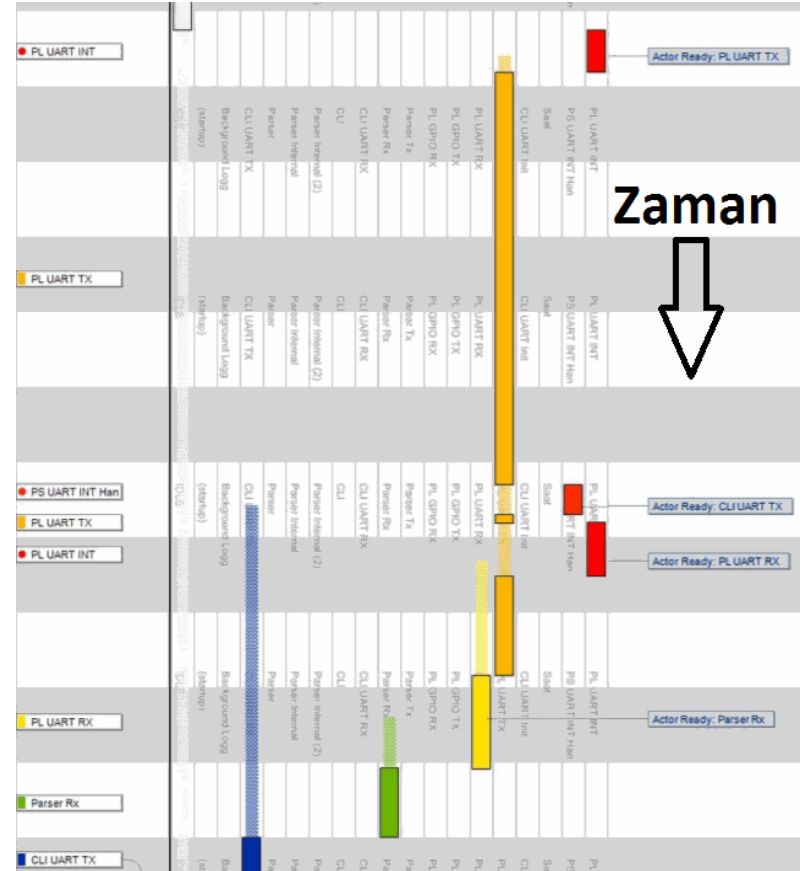
Yazılımın Alt Parçalara Daha Kolay Bölünmesi

«Task» ve IPC yapıları sayesinde problem, ufak parçalara daha kolay bölünebilmektedir.



Hata Ayıklamanın Zorlaşması

«Bare metal» ile karşılaştırıldığı zaman standart hata ayıklama (debug) yöntemleri yetersiz kalabilmektedir. Bunun için «FreeRTOS Aware Debugger» çözümleri kullanılabilir.



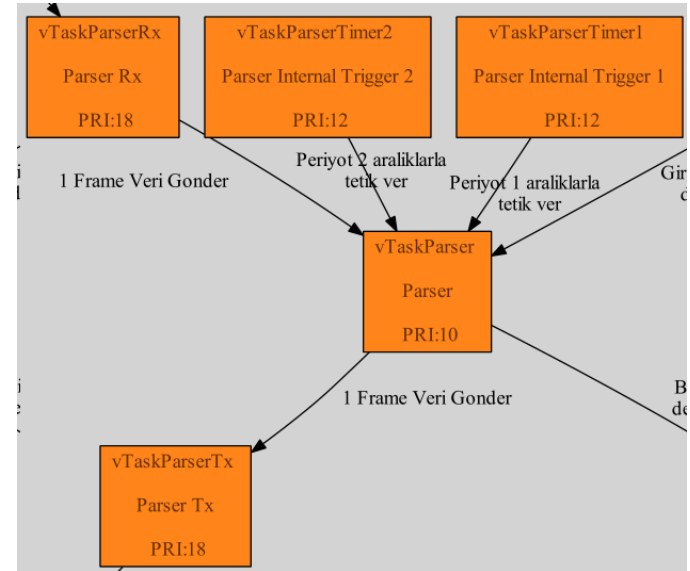


FreeRTOS ile Çalışmanın Öğrenilmesi

- İşletim sistemi ile çalışmanın,
- Multi-tasking kodlamada dikkat edilmesi gereken noktaların (kaynak paylaşımı vb.) öğrenilmesi gerekmektedir.

```
int main(void)
{
    /* ilklendirmeler */

    while(1)
    {
        /* Tum işlemler */
    }
}
```



«Super Loop» Yaklaşımı

FreeRTOS «Task» İlişkileri

Tavsiyeler

- FreeRTOS web sitesindeki eğitici dokümanlar ve referans dokümanı geliştirmeye başlamadan önce okunmalıdır.
- Çalışılacak işlemci (platform) için hazırlanmış örnek proje FreeRTOS sitesinden indirilip, onun üzerinden gidilmelidir.
- Kodlamaya başlamadan önce problemin işletim sisteminin çalıştıracağı «task»'lara nasıl bölüneceği, IPC mekanizmaları ve «task» öncelikleri kurgulanmalıdır.
- Xilinx ürünleri için SDK'da FreeRTOS projesinin «bare metal» proje üzerine elle eklenmesini önermekteyiz.

Teşekkürler

Sorularınız



SAVTEK 2018



Bir Açık Kaynak Kodlu Gerçek Zamanlı İşletim Sistemi (FreeRTOS) ile Gömülü Yazılım Geliştirme Çalışmaları

Alper Yazar

Sayısal Tasarım Uzman Mühendisi

ayazar@aselsan.com.tr

Savunma ve Sistem Teknolojileri (SST) Sektör Başkanlığı

ASELSAN

28 Haziran 2018