Embedded system, belirli bir görevi yerine getirmek için özelleştirilmiş donanım ve yazılım barındıran sistem.

Sadece offise ortamında değil, kontrol edemediğimi her koşulda çalışabilmesi için tasarlanmıştır.

Özelleştirilmiş görevler yerine getirmesi gerektiğinden daha maliyet performans oranı düşük donanıma ve %100 yararlanılabilir bir yazılıma ihtiyaç duyar.

Donanım spesifik bir şematik ve tasarıma sahiptir.

Embedded sistem durumların kontrolü, data gönderilmesi ve alınması, dış cihazların controlü gibi parçalardan oluşur. Kullanıcı arayüzlü değildir genelde. Debug, diagnostik ve geliştirme süreleri web veya pc programlarına göre zordur.

Embedded sistemin türleri

Microcontroller(MCUs)

System on Chip(SoC)

Application-specific integrated circuit(ASICs)

Field programmable gate array(FPGAs)

Endiannes big endian, little endian, bi-endian (switching )

Data Alignment

Donanımla iç içe bir çalışma yapıldığından hatanın nereden kayanaklandığının anlaşılabilmesi gerkmekte.

Hardware deki bir elemanın bozulması veya yanlış çalışması software’i etkileyebilir ve hatanıın kök nedeni araçtırılmak zorunda kalınabilir. Yazılımda revizyon yapıldığında eski donanım karşılamayabilir.

Yazılım donanıma bağlı olarak çalışır.

Donanım ve memory hatalarına karşı cyclic redundancy check (CRC) kullanılır.

High temperature, low temperature, high humidity, vibration, dust, and other

environmental factors can significantly affect the performance

Embedded sistemde yapılması gerekenler

Over time, the complexity of embedded systems increased and developers faced the

limitations of C, the most notable being error-prone resource management and a lack of

high-level abstractions. The development of complex applications in C requires a lot of

effort and time.

At the same time, C++ was evolving, gaining new features and adopting programming

techniques that make it the best choice for developers of modern embedded systems. These

new features and techniques are as follows:

You don't pay for what you don't use.

Object-oriented programming to time the code complexity.

**Resource acquisition is initialization** (**RAII**).

Exceptions.

A powerful standard library.

Threads and memory model as part of the language specification.

Detay kodların bilinmesi istenmediği durumlarda kullanılan bazı yazılımları C de yazmak daha zahmetli olduğundan object oriented programlamanın C++ ta olması işleri kolaylaştırıyor.

**A build system**: The system you use to write the code

**A target system**: The system your code is going to be run on

Ubuntu 18.04 as the build system

Raspberry Pi as the target system