

Proje Klasör Yapısı

Kaynak	Tarih	Versiyon	İşlem
Zafer Satılmış	2019-05-06	1.0	Döküman oluşturuldu

1.Amaç	2
2.Neden Klasör Yapısı	2
3.Klasör Yapısı	3
3.1.Sub-Repository	3
3.2.Project Repository	4
4.Kaynaklar	8

1.Amaç

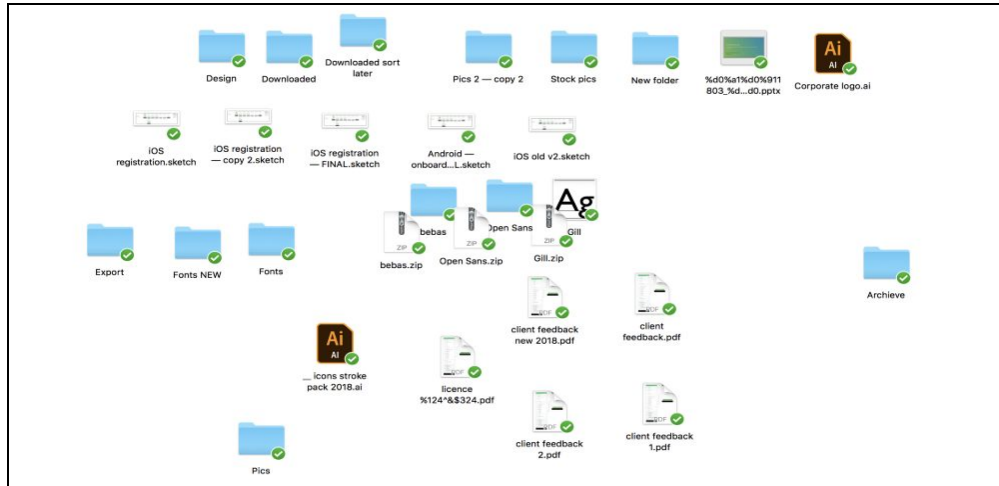
Bu belgenin amacı tüm projelerde kullanılacak kapsamlı klasör yapısını tanıtmaktır. Ekibe yeni katılacak arkadaşlar bu belge ile istenilen dosyaları, kaynak kodları ya da belgeleri hızlıca bulabilir.

2.Neden Klasör Yapısı

Geliştirilen uygulamalar hiçbir zaman “Hello World” uygulaması seviyesinde olmadığı için gelişmiş bir klasör yapısına gereksinim vardır. Gelişmiş klasör yapısının ilk akla gelen faydaları aşağıda sıralanmıştır.

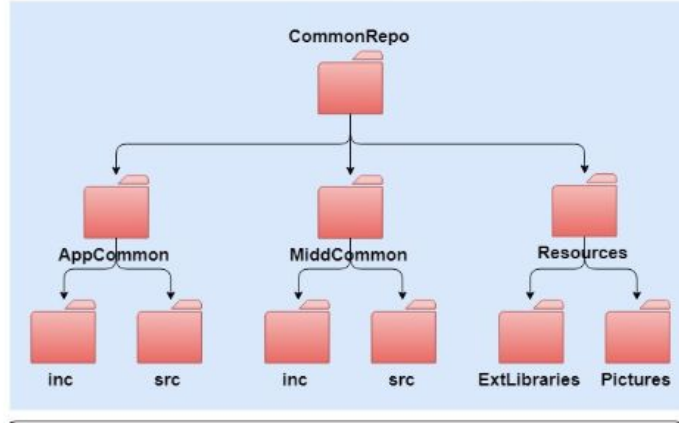
- Dosyaları gruplar ve istenilen dosyaları hızlıca bulmamızı sağlar.
- Eklenecek yeni dosyaya bir neden oluşturur. Eğer dosya klasör yapısındaki klasör gruplarından birine uymuyor ise oluşturulmak istenen dosya geçici ya da o an işi kolaylaştıracak dosya olabilir. Gelip geçici dosyaların projede olmasını engeller.
- Yazılım katmanlarını klasör seviyesinde temsil eder. Bu sayede geliştirici projeye ekleyeceği yeni dosyanın yerini hızlıca belirler.
- Ekibe yeni katılan arkadaşlar projenin yapısını, yazılım katmanlarını ve kaynak kodları hızlıca kavrar.
- Ortak amaç taşıyan kodların birden fazla projede kullanılmasına imkan sağlar. Böylece her projede kullanılan kodlar tekrar tekrar yazılmasına gerek kalmaz,
- Ortak kullanımda olan kaynak kodu dosyasında (.c) bulunan bir bug çözümü otomatik olarak tüm projelerde çözülmüş olur.
- Yazılım geliştiren arkadaşlar diğer proje kodlarını da görerek problem çözümünde referans sağlayacak bilgiler, kod parçaları ya da metotlar bulabilir.

Yukarıdaki listeyi daha da uzatabiliriz. Eğer bu gereksinim es geçilir ise kesinlikle zaman içerisinde çalışma ortamımız aşağıdaki gibi olacaktır.

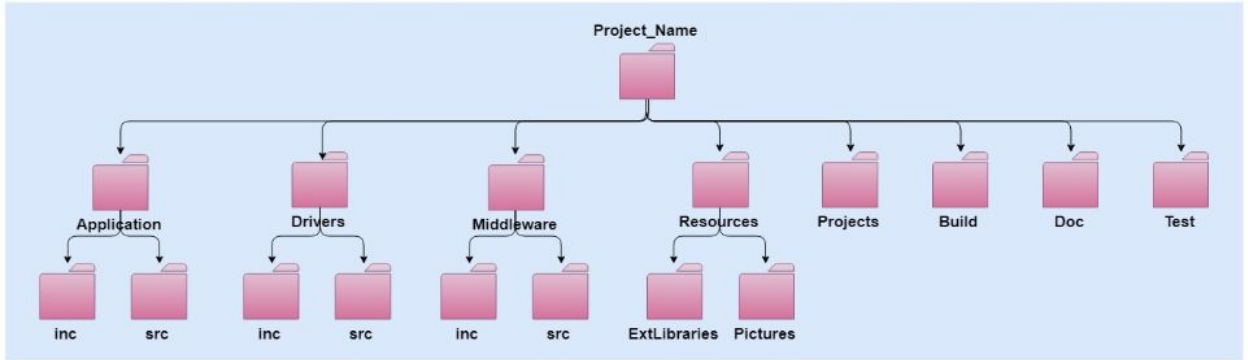


3.Klasör Yapısı

Sub-Repository



Project Repository



Kullanımı hedeflenen klasör yapısının .jpg, .pdf, html formatlarında gösterimini bu belgenin bulunduğu klasörde bulabilirsiniz. Ortak kodlar ve kütüphaneler için Sub-Repository, projeler için ise Project Repository kullanılacaktır. Dolayısı ile sub-repository tüm projelerde kullanılmak üzere birtane, project repository ise her proje için ayrı ayrı oluşturulmalıdır.

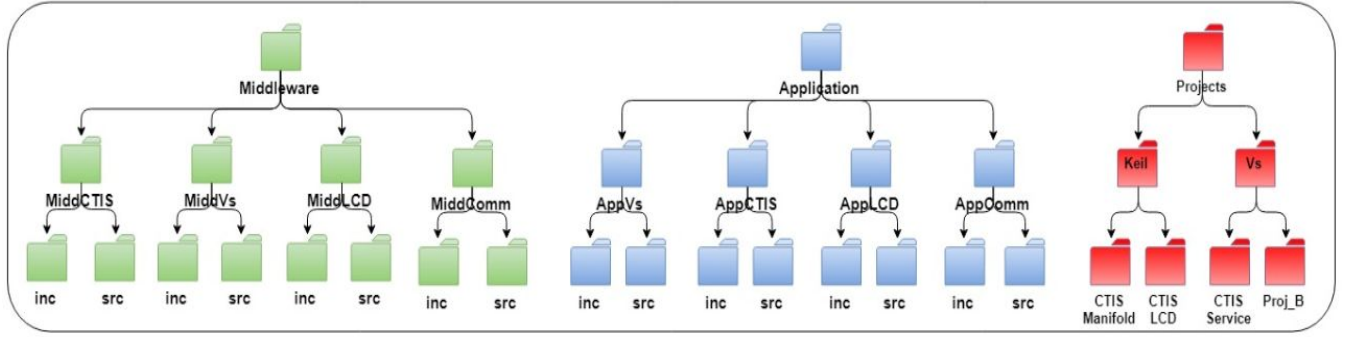
3.1.Sub-Repository

- **AppCommon:** Uygulama seviyesinde ortak kodları içeren dizindir. Örneğin JSON işlerini yapan dosya burada olabilir. Çünkü diğer projelerde kullanılma ihtimali var.
- **MiddCommon:** Middleware seviyesinde ortak kodları içerir. Örneğin zaman bazlı alarm veren bir yapı yada thread işlerini yapan kodlar burada bulunabilir.
- **Resource:** Projelerde ortak kullanılabilecek kaynakları içerir. Örneğin versiyonlama sistemi için gerekli uygulama, resimler ya da kütüphaneler olabilir.

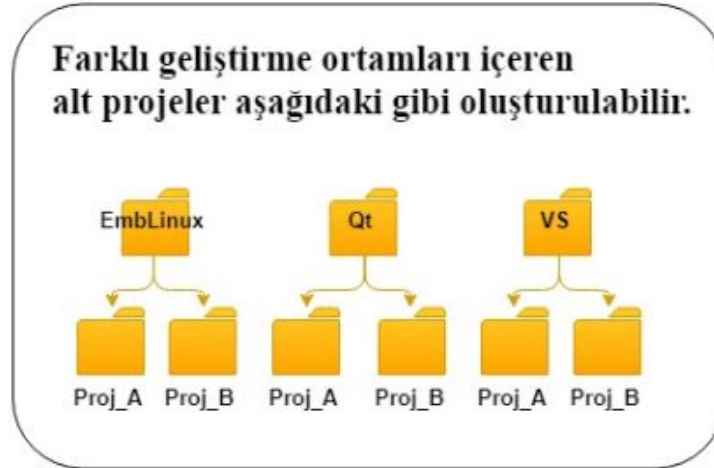
3.2. Project Repository

- **WorkSpace:** Kök dizindir. Tüm projeler bu dizin altında oluşturulur.
- **Project_Name:** Proje isminde oluşturulan dizindir. Projeye ait tüm çalışmalar bu dizin altında bulunur.
- **Project_Name/Application:** Bu dizin altında sadece uygulama seviyesinde kullanılan kaynak(.c) ve başlık(.h) dosyaları bulunur. Yani uygulama katmanı kodlarını burada aramamız gerekir.
- **Project_Name/Drivers:** Projede kullanılan tüm driver dosyalarını içerir. Örneğin kullanılan bir LCD' nin driver burada bulunması gerekir.
- **Project_Name/Middleware:** Middleware tanımı, farklı iki uygulamayı ya da katmanı birbiriyle ilişkilendiren ara yazılım demektir. Bizim tarafımızda application seviyesindeki kodlar ile driver seviyesindeki kodları birbirine bağlayan ya da ara katmanda bulunan kodları içerir. Application seviyesinde direk driver kodlarını çağırmaı engeller. Ayrıca şifreleme işlemleri de middleware katmanına girer. Çünkü uygulama seviyesinde şifreleme işlemleri yapılmaz. Middleware seviyesinde şifrelenir ve application seviyesinde kullanılır.
- **Project_Name/Resource:** Projede kullanılan kaynakları içerir. Örneğin resim dosyaları, ses dosyaları ya da kütüphaneler (static or dynamic library) burada bulunur.
- **Project_Name/Projects:** Projenin oluşturulma dizini bu dizin altındadır. Örneğin Keil ile proje oluşturulacak ise bu dizin altında oluşturulmalıdır. Bu dizin altında kod dosyaları bulunmaz. Örneğin Keil uygulamalarını açmak için gerekli olan .uvprojx uzantılı dosya bulunur. Bu dosya açılarak tüm proje açılmış olur.
- **Project_Name/Build:** Projenin derlenmesi için gerekli toolchain burada bulunur. Bu sayede projenin derleyicisi her zaman elimizin altında bulunmuş olur.
- **Project_Name/Doc:** Proje geliştirilmesinde kullanılan ya da oluşturulan tüm belgeler bu dizin altında toplanır. Örneğin uygulamanın akış diyagramı, proje gereksinimleri, ...
- **Project_Name/Test:** Projeye ait tüm test adımları, test belgeleri, test ortamının oluşturulmasını sağlayan kurulum, ekipman listesi gibi test ile ilgili tüm belgeler, kaynaklar, resimler vb bu dizin altında bulunur.

Eğer proje birden fazla uygulamadan oluşuyor ise (Thermostat board, modem board, Service Yazılımı) bu uygulamalara ait kodlar ve diğer kaynaklar da /Project_Name dizini altında ilgili alt dizinde olmalıdır. Örnek olarak aşağıdaki resimler incelenebilir.



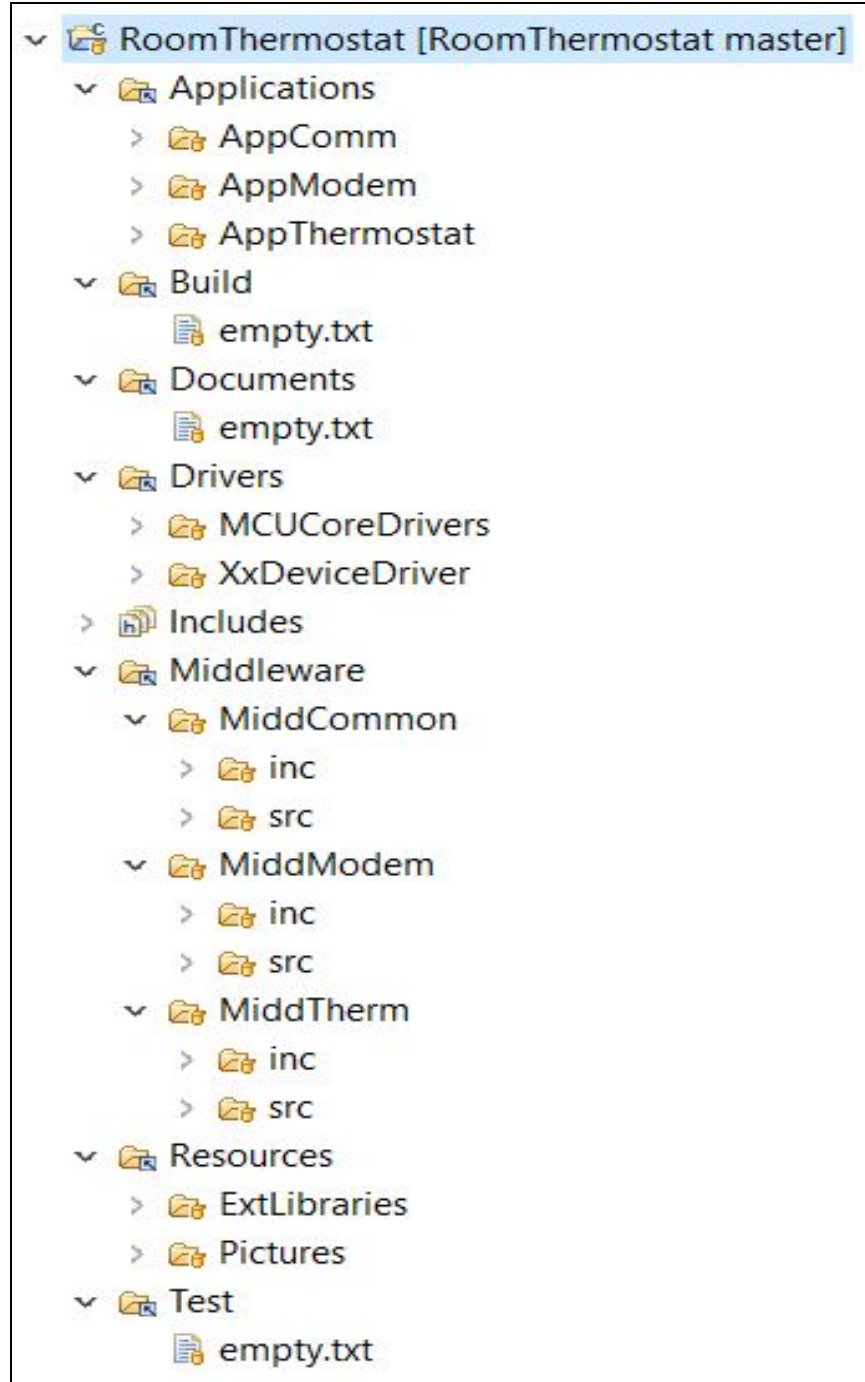
Resimde görüleceği üzere Proje 3 farklı uygulamadan oluşmaktadır. Thermostat, VS(visual studio), Modem uygulamaları. Ayrıca bunlar arasında ortak kullanılan kaynakları için oluşturulan alt dizinler görülmektedir.(Comm). Oluşturulacak tüm projeler bu örnek referans alınarak oluşturulmalıdır.



Eğer proje farklı ortamlarda geliştirilen birden fazla uygulamadan oluşuyor ise yukarıdaki örnekteki gibi **Project_Name/Projects** dizini altında gerekli alt dizinler oluşturulmalıdır.

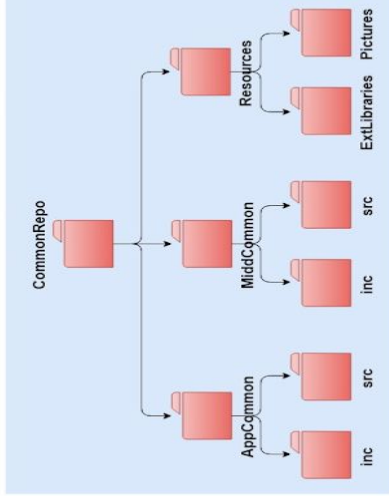
Aşağıdaki sayfada yukarıdaki plana uygun olarak oluşturulmuş bir proje örneği görülmektedir.

Eclipse ortamında oluşturulan yapı aşağıdaki gibi olacaktır. Burada kladörler link olarak projeye eklenmiştir. Proje tıpkı yukarıda yazdığı gibi **RoomThermostat\Projects\EclipseWin\Thermostat** dizini altında oluşturulmuştur. Bu dizin altında sadece Eclipse kurulum dosyaları bulunmakta. Projeyi indirmek için Projeler sekmesine ya da github bakabilirsiniz.



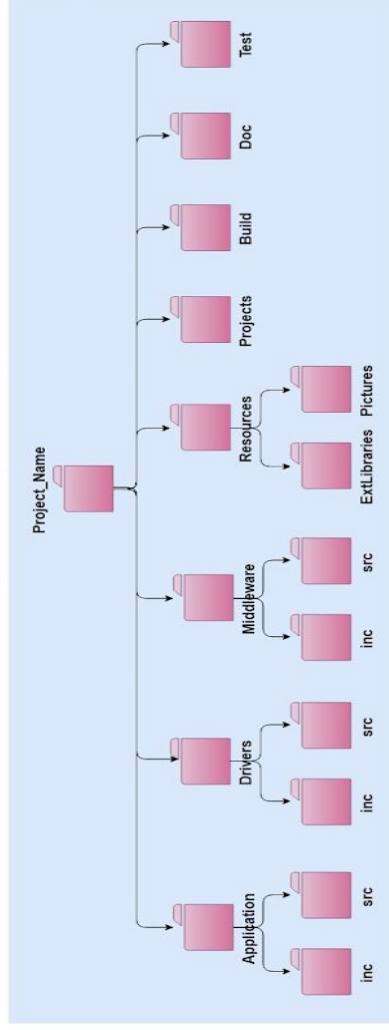
Geliştirme Ortamı Klasör Yapısı

Sub-Repository



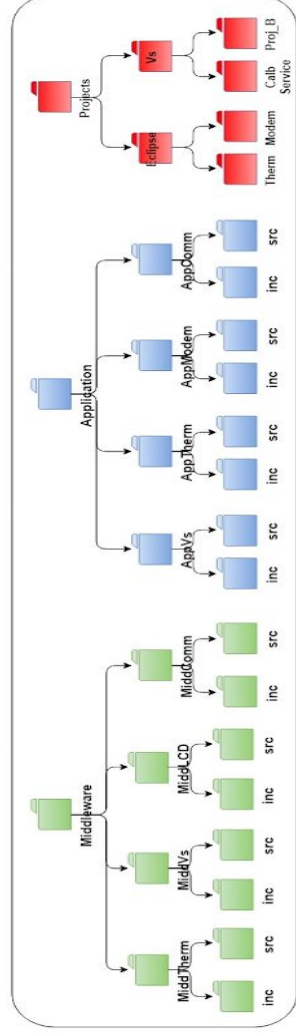
Projelerde ortak kullanılan kaynaklar CommonRepo klasörü altında toplanır. Bu alanın ayrı bir repository vardır. CommonRepo diğer projelere sub-repo olarak eklenir.

Project Repository

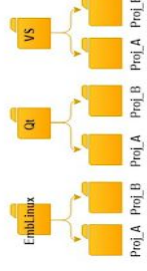


Bu alan ilgili projenin ana repository alanıdır. Her projenin kendisine ait bir repository mevcuttur. CommonRepo bu alanın sub-repo olarak tanımlanmalıdır. Projeye özgü gelişmelerin hepsi ilgili klasörün altında toplanmalıdır. CommonRepo kaynakları projelere link verilerek eklenmelidir.

Eğer bir projede birden fazla proje var ise o projeler geliştirme ortamları isminde oluşturulacak klasör altında oluşturulmalıdır. Örneğin kontrol bords, lcd bords ve Visual Studio(VS) uygulamasında oluşan Oda Termostat-Modem projisinin tüm alt uygulamaları aynı repository toplanmalıdır. Bu örneğe göre Middleware, Application ve Proje klasörleri aşağıdaki gibi olacaktır. Faydaları: tüm projenin bir anda toplanması, projede olanların farklı uygulamaların son versiyonlarını hızlıca kullanabilmesi, uygulama bağlantılarının iki tarafta da kolaylıkla erişilmesi (örneğin mesaj paketleri içerikleri), belgelerin ve test adımlarının/sonuçlarının ortak alanda ana projeye hitap edilebilecek şekilde saklanması



Farklı geliştirme ortamları içeren alt projeler aşağıdaki gibi oluşturulabilir:



4.Kaynaklar

1. <https://www.linkedin.com/pulse/project-directory-structure-abhishek-arora>
2. <https://hiltmon.com/blog/2013/07/03/a-simple-c-plus-plus-project-structure/>
3. <https://www.embeddedrelated.com/showarticle/628.php>
4. <https://uxdesign.cc/a-systematic-approach-for-managing-project-folder-structures-4e2e553cad00>
5. <https://git-scm.com/download/gui/windows>