

1)Ne kadar uğraşmamda doğru cevaba ulaştıramadım. Asıl yapmak isteğim koordinatlarından tüm hedefler sırasıyla 1li permütasyon,2li permütasyon... gibi çıkartıp bu targetların arabanın pozisyonu ile sıra sıra işlemlere sokulup en son en yakını yaptırmaktı ancak permütasyon için kullandığım itertools metodunda permütasyonlar grup halinde saklanıyor yani len(itertools desem) sonucu4,12,24,24 şeklinde ve işlemlerimde bundan dolayı index hatası yüzünden devam edemedim

2)Değerleri verip onları reg içerisine atadım aslında 4 bit olayından dolayı yanlış olmasını bekledim çünkü 4 bit max 15 değerini tutabilirim amacım deneyim sonucu görmek ama çalıştı. Hatta denemeden sonra değerleri yuvarlayarak tutmayı düşünmüş Tüm incelemede doğru şeklini veyahut bu doğruysa nasıl oldu açıklarsanız sevinirim

3)Amaç: A ve b değerlerini birebirlerine yyy dizisi içerisinde birbirlerine atamaktır

Xxxxx fonksiyonu 3 parametre alır yyy,a,b

Zzz'ye geçici olarak yyy'nin a dizisindeki değeri verildi.Değer takastan önce zzz de korunur.

Yyy[a] = yyy[b] yye eşitlenir

Yyy[b] = zzz eşitlenir zzz yani bu da yyy[b] = yyy[a](nın ilk değeri)

Yani a dizi ile b dizi yer değişmiştir.

son dizi ile *yyy son haline kavuşturulmuştur

4)Amaç:yyy dizinin minimum değerini bulabilmek

Xxxxx 2 parametre alır yyy,n

Zzz yyy'nin ilk değeri olarak atanır

Döngü başladığında aslında yyy nin ilk değeri ile 1. Karşılaştırılır eğer küçükse yeni değer zzz ye geçer ve bu şekil dizi bir tane kadar devam eder en son zzz dizi içerisindeki en küçük değerdir

5-6)NESNE YÖNELİMLİ PROGRAMLAMA (NPY)

Npy de her bir object belirli sınıfın

veya altsınıfın(subclass) metotlarını(method) veya prosedürlerini(procedure) ve veri değişkenlerini barındıran birer örneğidir. Sınıf belli tür bir nesnenin metotlarının ve değişkenlerinin şablon tanımıdır. Yani bir nesne bir sınıfın özel bir örneğidir. Sınıf içerisinde oluşturulan birim nesnedir.

Sınıftan nesne üretilip değişiklik yapılmak istendiğinde tüm programda değişiklik yapmak gerekmez, sadece oluşturulan nesnenin sınıf içinde değişiklik yapılması yeterlidir.

Oluşturulan nesneler birbirinden bağımsız olduğu için bilgi gizleme olanağı artar.

Sınıf nedir

Sınıfta tanımlanan metot ve değişkenlere sınıfın üyeleri denir. Metotlar ise , bir görevi yerine getiren alt programlardır. Sınıf soyut bir kavramdır doğrudan kullanılmaz nesne oluşturulup kullanılabilir

Nesne Nedir?

İçinde veri saklayan ve bu veriler üzerinde işlem yapacak olan metotlar bulunduran bileşenlerdir. Nesneler her uygulamada tekrar kullanılabilir. Nesne oluşturduğumuzda hafızada yer kaplar.

Nesne Yönelimli Programlamada 4 temel özellik vardır. Bu özelliklerden birini sağlamayan programlama dili nesne yönelimli programlama dili olarak sayılmaz.

- 1.) Soyutlama (Abstraction)
- 2.) Kapsülleme (Encapsulation)
- 3.) Miras Alma (Inheritance)
- 4.) Çok biçimlilik (Polymorphism)

Encapsulation

Davranış ve özellikler sınıfta soyutlanarak kapsülendir. Kapsülleme ile hangi özellik ve davranışın dışarıya sunulup sunulmayacağını belirleriz.

Örneğin İnsan sınıfında yemek alışkanlığı bizi ilgilendirmiyorsa bunu kapalı (private) yapıp gizleriz. Ancak isim soy isim gibi bilgiler bizi ilgilendirdiği için bunlar açık bırakılır. Bu olaya bilgi saklama yani kapsülleme denilmektedir. Bilgi saklama erişim belirteçleri (public, private, protected) ile gerçekleştirilir.

public : herkesin kullanabileceği özellik ve davranışlardır

private : sadece kendi sınıfında kullanılabilen özellik ve davranışlardır

protected : sınıf içinde miras alınan alt sınıflar tarafından kullanılır..

Inheritance

: Sınıflar birbirinden türeyebilir. Alt sınıf üst sınıfın özelliklerini alabilir.

Single Inheritance (Tekli Kalıtım): Alt sınıf tek bir üst sınıfın tüm özelliklerini taşır.

- Multiple Inheritance (Çoklu Kalıtım): Bir alt sınıf birden fazla üst sınıfın tüm özelliklerini taşır.

- Multilevel Inheritance (Çok Seviyeli Kalıtım): Bir sınıfın alt sınıfı oluşturulduktan sonra bu alt sınıfın da bir alt sınıfının oluşturulmasına denir.

- Hierarchical Inheritance (Hiyerarşik Kalıtım): Bir üst sınıfın birden fazla alt sınıfa base class (temel sınıf)'lik yapmasına denir.

- Hybrid Inheritance (Melez Kalıtım): Öbür Inheritance türlerinin 2 veya daha fazlasını barındıran Inheritance türüdür.

Polymorphism

Alt sınıflar üst sınıfın gösterdiği davranışları göstermek zorunda değildir. Alt sınıfların farklı davranışları göstermesine Çok biçimlilik denilmektedir.

Abstraction

Bir sınıfta davranış ve özelliklerin tanımlanmasına soyutlama diyoruz.

7)Görüntü işleme

Pillow

Pillow, Python'da bulunan bir görüntü işleme kütüphanesidir. Yeniden boyutlandırma, filtre ekleme ve daha fazlası gibi görüntüleri işlemek için birçok prosedür sağlar. Python'da bulunan en iyi görüntü işleme yöntemlerinden biridir.

Pillow'u yüklemek için bu komutu kullanın: `pip install pillow`

NumPy

Görüntüleri değiştirmek için de kullanılabilir. NumPy, görüntüleri RGB değerlerine göre özelleştirmenizi ve işlemenizi sağlar. NumPy kullanarak, görüntülerin RGB değerlerini kolayca değiştirebiliriz.

NumPy nasıl kurulur: `pip install numpy`

OpenCV

Günümüzde, çoğunlukla yüz algılama ve tanıma, nesne algılama ve daha fazlası gibi bilgisayarla görme görevlerinde kullanılmaktadır.

OpenCV'yi: `pip install opencv-python`

SimpleCV

SimpleCV ayrıca yüz algılama ve tanıma, nesne algılama ve daha fazlası gibi bilgisayarla görme uygulamaları oluşturmak için açık kaynaklı bir pakettir. Bilgisayarla görme görevlerini daha basit hale getirir. Paketi kullanarak çok sayıda yüksek güçlü bilgisayar görme aracına erişebilirsiniz

8) RGB nedir, HSV nedir, aralarındaki farklar nelerdir?

RGB:red green blue üç temel renkten oluşarak tüm renkleri elde edilebilecek sistemdir. Renklerin yoğunluğu 0-255 arasındadır.

HSC:Hue,Saturation,ve Value terimleri ile rengi tanımlar.RGB de renklerin karışımı kullanılmasına karşın HSV de renk, doygunluk ve parlaklık değerleri kullanılır

Hue: Renk özünün sayısal değeri

Saturation: Rengin doygunluğunu ifade eden sayısal değer

Value: Parlaklığı ifade eden sayısal değer

9) Struct yapısındaki bir değişkenin elemanları kullanılmak istendiğinde hangi durumlarda '.', hangi durumlarda '->' kullanılır?

Bir yapı ve yapı değişkeni oluşturulduktan sonra, yapı içinde yer alan elemana yapı değişkeni yolu ulaşmak için `=>yapıdeğişkeni.adı`

```
struct yap *yp, yd1, yd2;
yp = &yd1;
yp->id = 21;
yp->cd = 'A';
yp = &yd2;
yp->id = 35;
yp->cd = 'B';
```

```
struct yap yd1, yd2;
yd1.id = 165;
yd1.cd = 'K';
yd2.id = 384;
yd2.cd = 'T';
```

-> işlemi * yani pointerla yapı içinde yer alan elemana ulaşmak için kullanılır.

10)BİT MASKELEME NEDİR

Bit maskeleme ,verileri char/int/float olarak işlemek yerine, bit olarak işlem yöntemidir. Belirli türdeki verileri kompakt ve verimli bir şekilde depolamak ve işlemek faydalı bir araçtır.

Literatürde 1 için doğru, true, high, set gibi adlar kullanılırken 0 için yanlış, false, low, reset gibi adlar kullanılır.

NOT a	Sonuç, giriş değerinin tersidir. Giriş değeri 1 ise sonuç 0, 0 ise sonuç 1'dir.
a AND b	Her iki giriş değeri de 1 ise, sonuç 1'dir, aksi takdirde 0'dır.
a OR b	Giriş değerlerinden biri 1 ise, sonuç 1'dir, aksi takdirde 0'dır.
a XOR b	Giriş değerleri birbirinden farklı ise, sonuç 1, aynı ise sonuç 0'dır.

n'nci Bitin Değerini 1 Yapma

Bir değişkenin n'nci bitinin değerini 1 yapmak için bu değişken 2^n ile OR mantıksal işlemine tabi tutulur.

OR (|)

int n = 3;

int degışken = 0b01000010;

Degışken |= 1 << n;

n'nci Bitin Deęerini 0 Yapma

Bir deęiřkenin n'nci bitinin deęerini 0 yapmak için bu deęiřken, 2^n 'in mantıksal tersi (NOT) ile mantıksal AND işlemine tabi tutulur.

AND(&)

Bir deęiřkenin n'nci bitinin deęerini 0 yapmak için bu deęiřken, 2^n 'in mantıksal tersi (NOT) ile mantıksal AND işlemine tabi tutulur.

int n = 3;

int degışken = 0b01001010;

degışken &= ~(1 << n); // SIRALAMADAKİ 3. BİTİ 1 İLE DEĞİŞTİRİR

// 01001010 AND 11110111 == 01000010

n'nci Bitin Deęerini Ters Çevirme

Bir deęiřkenin n'nci bitinin deęerini ters çevirmek (1 ise 0, 0 ise 1) için bu deęiřken, 2^n ile XOR işlemine tabi tutulur. Toggle işlemlerinde kullanılır.

XOR (^)

int n = 3;

int degisken = 0b01000010;

degisken ^= 1 << n;

// 01000010 XOR 00001000 == 01001010

11) Mikrodenetleyici nedir ve nasıl çalışır?

Mikrodenetleyici, günlük olarak kullandığımız elektronik aletlerin çoğunu control eden gömülü bilgisayar çipidir. Bölümü bir sistemdeki belirli bir işlemi yönetmek için tasarlanmış kompakt bir entegre devredir.

Tek bir çip üzerinde yer alan bir bilgisayar benzetmesinde bulunulabilir. ok sayıda genel amaçlı giriş/çıkış pini (GPIO) içerir. GPIO pinleri, bir giriş veya çıkış durumuna göre yazılımlarla yapılandırılabilir. GPIO pinleri bir giriş durumuna göre yapılandırıldığında, genellikle sensörleri veya harici sinyalleri okumak için kullanılırlar. Çıkış durumuna göre yapılandırılan GPIO pinleri, LED'ler veya motorlar gibi harici cihazları, harici güç elektronięi yoluyla, genellikle dolaylı olarak çalıştırabilir.

Birçok gömülü sistemin analog sinyaller üreten sensörleri okuması gerekir. İşlemciler 0 ve 1'lerden oluşan dijital verilere işlem yapabildiğinden, okunan analog sinyallere işlem yapabilmek için Analog-Dijital dönüřtürücüye (ADC) ihtiyaç duyulur.

Mikrodenetleyici içinde okuma için herhangi bir disk bulunmaz.

Çalıştırılacak program çip üzerindeki ROM'a yüklenir.

ROM boyutu çalıştırılacak olan programın boyutunu belirler.

İşletim sistemi mevcut değildir.

Sistemde çalışacak tek program ROM üzerinde bulunur.

Mikroişlemci ve Mikrodenetleyici karşılaştırma	
Mikroişlemci	Mikrodenetleyici
Kişisel bilgisayarlarda kullanılır.	Gömülü sistemlerde kullanılır.
Sadece bir CPU(Merkezi İşlem Birimi)'dan oluşur.	Tamamı tek çipe entegre edilmiş bir CPU, Bellek ve G/Ç portlarından oluşur.
Von Neumann modeline göre imal edilir.	Harvard mimarisine göre imal edilir.
Karmaşık ve pahalıdır. İşlenecek çok sayıda talimat vardır.	Daha ucuz ve basittir. Daha az talimat içerir.
Aynı anda birden fazla görevi gerçekleştirebilir.	Aynı anda sadece tek bir görevi gerçekleştirebilir.
RAM, ROM, G/Ç portları ve zamanlayıcılar haricen eklenebilir ve sayı olarak değişebilir.	RAM, ROM, G/Ç portları ve zamanlayıcılar harici olarak eklenemez. Bu bileşenler bir çip üzerinde birlikte yer alır ve sayısı sabittir.
RAM, ROM ve diğer çevre birimlerine arabirim oluşturmak için harici bir veri yolu kullanır.	RAM, ROM ve diğer çevre birimlerine arabirim oluşturmak için dahili bir kontrol veriyolu kullanır.
Gereken bellek miktarına veya G/Ç port sayısına tasarımcılar karar verebilir.	Bellek miktarı veya G/Ç port sayısı için kullanılan değerler, bir mikrodenetleyiciyi için sınırlı ancak belirli bir görev için uygun bir sistem oluşturur.
Program ve veri depolama sağlamak için harici bellek kullandığından, bellek kısıtlaması yoktur.	Programı depolamak ve yürütmek için çip üzerinde gömülü Flash bellek kullanıldığından, daha kısa bir başlatma süresi ve kodu hızlı bir şekilde yürütme olanağı sağlar. Bellek alanı sınırlıdır.
Harici bellek ve G/Ç port için harici destek, mikroişlemci tabanlı bir sistemi daha ağır ve daha maliyetli hale getirir.	Mikrodenetleyiciler, mikroişlemcilerden daha hafif ve ucuzdur.
Harici cihazlar daha fazla alan gerektirir ve güç tüketimleri daha yüksektir.	Mikrodenetleyici tabanlı bir sistem daha az güç tüketir ve daha az yer kaplar.

Mikrodenetleyici programlama

Mikrodenetleyiciler, Assembly, C, Python ve Javascript programlama dilleri ile programlanabilir.

12) Gömülü yazılımda kullanılan programlama dilleri nelerdir?

Gömülü sistemler özellikleri

Tek Fonksiyonlu: Gömülü bir system genellikle özel bir görevi yapmak için tasarlanmıştır. Yani aynı işi tekrar tekrar yapar .

Sıkı bir şekilde kısıtlanmış

Reaktif ve gerçek zamanlı: Hızlı ve reflektif verileri eksiksiz ve geciktirmeden gerçek zamanlı olarak hesaplamalıdır.

Mikroişlemci veya mikrodenetleyici tabanlı olmalıdır.

Bellek: Yazılımı genellikle ROM'a gömüldüğü için belleğe ihtiyacı vardır.

İşletim sistemleri

Yazılım kodu genellikle C veya C++ programlama dillerinde yazılır, fakat son zamanlarda mikrodenetleyiciler ve gömülü sistemlerde kullanılma amacıyla Python ve JavaScript gibi yüksek seviyeli programlama dillerinin kullanımı yaygınlaşmıştır.

13) Interrupt nedir ve nasıl kullanılır/nasıl çalışır?

İşlemin acil olarak bildirilmesi durumudur. Bu durumlarda işlemci üzerinde işlenmekte olan sürecin bitmesini beklemeyecek acil bir kritik oldukları için bekletilmeden işlenmelidir.

Bilgisayarın merkezi işlem biriminin (CPU) hiç hesapta olmayan bir olayın etkisiyle ,normal olarak yapmakta olduğu işi bırakarak, kesmenin gösterdiği olaya geçici olarak yönelmesi için kullanılır

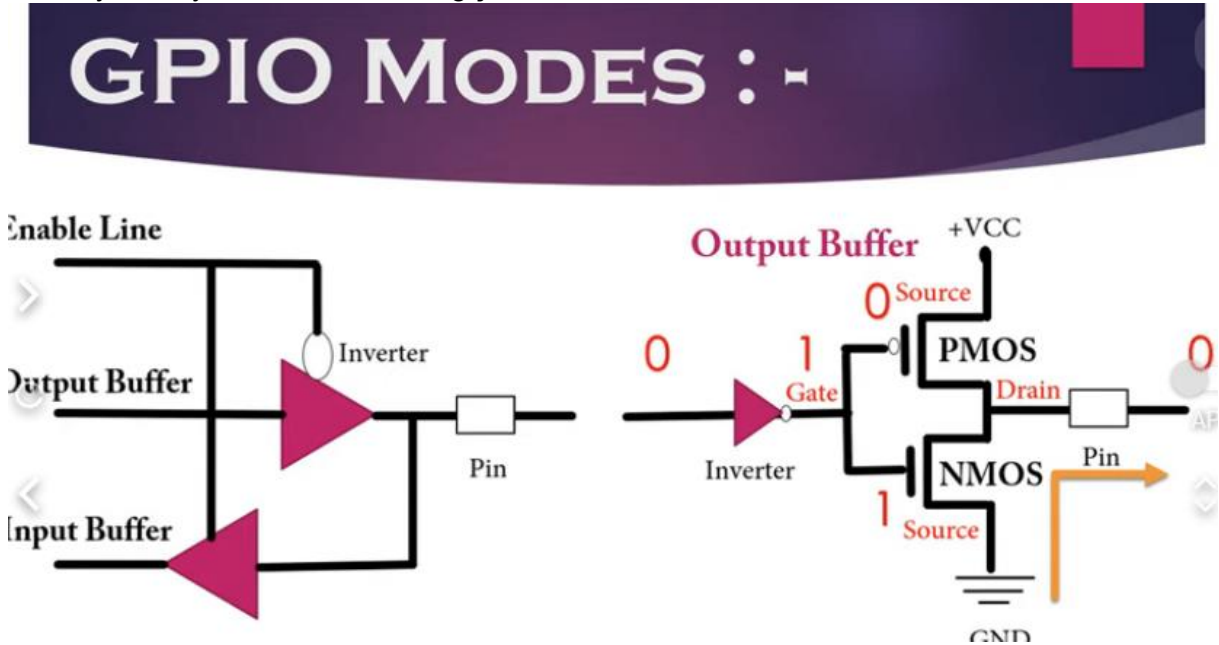
14) GPIO nedir ve nasıl kullanılır?(GENERAL PURPOSE INPUT OUTPUT

PIO pinleri, kullanıcının veri alışverişini yapabileceği ve kontrol edebileceği genel amaçlı dijital giriş/çıkış pinleridir.

Çalışma prensibi ,GPIO pinleri eiki temek modda kullanılabilmesidir: giriş ve çıkış modu. Giriş modunda ,GPIO pinleri dışarıdan gelen sinyalleri okuyabilir. Bu durumda pin genellikle yüksek (1) veya düşük (0) seviyelerdeki gerilimi algılar.

Gpio, bir pini giriş , çıkış ve alternatif modlarla kullanabilmemizi sağlayan bir dahili çevre birimdir.

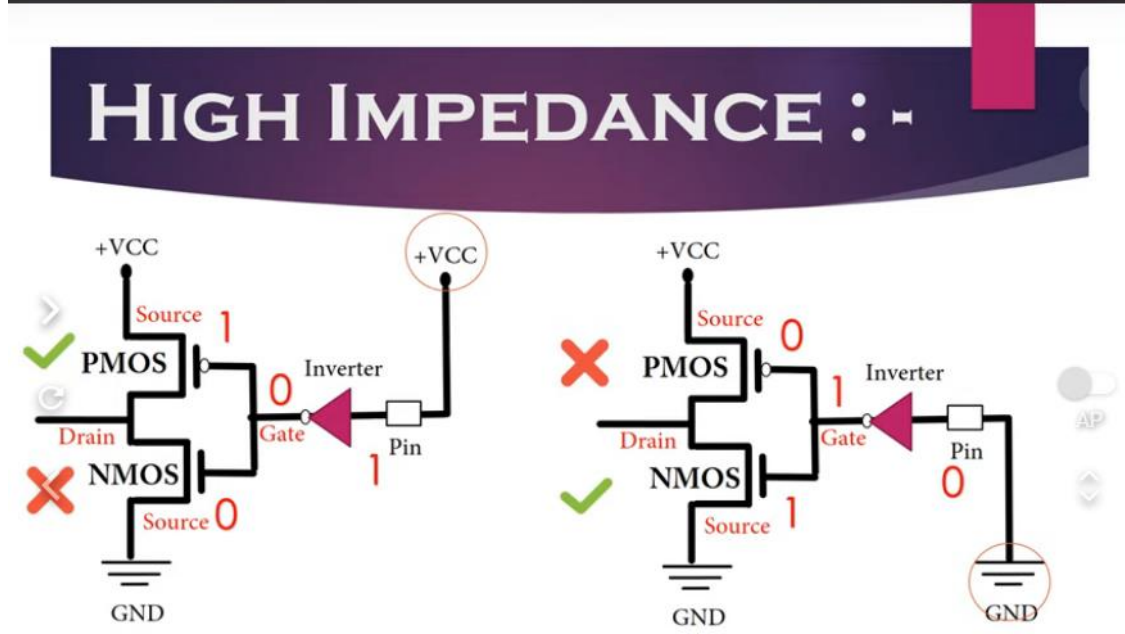
Fonksiyonlar yazılımsal olarak değiştirilebilir.



GPIO pinlerini çalıştırmak ve kontrol etmek için genellikle bir programlama dili veya kütüphane kullanılır. Örneğin, Python programlama dili genellikle Raspberry Pi gibi tek kart bilgisayarlarla kullanılan GPIO kontrolü için yaygın olarak tercih edilen bir dil ve kütüphaneye sahiptir. Bu kütüphane, GPIO pinlerini etkinleştirmek, okumak, yazmak ve durumlarını kontrol etmek için kullanılabilir.

Özetlemek gerekirse, GPIO, dijital sistemlerde genel amaçlı giriş/çıkış sağlayan pinlerdir. Bu pinler, dış bileşenlere bağlanarak veri alışverişi yapılabilir veya çıkış sinyalleri gönderilebilir. GPIO pinlerinin çalışması, giriş ve çıkış modları aracılığıyla gerçekleştirilir ve bir programlama dili veya kütüphane kullanılarak kontrol edilir.

Inputun devresi ise outputun tersi şekildedir



15) Timer nedir ve prescaler ne işe yarar?

Timer, bir mikrodenetleyici veya dijital sistemde zamanlama işlevi sağlayan bir bileşendir. Timer'ın temel işlevi, belirli bir süre veya frekansta bir olayı tetiklemektir. Bu, bir mikrodenetleyicinin belirli bir süre boyunca beklemesini veya belirli aralıklarla bir işlemi gerçekleştirmesini sağlayabilir. . Örneğin, bir mikrodenetleyicinin belirli bir süre sonra bir sensörü okuması veya bir çıkış sinyali üretmesi gerekiyorsa, bir timer kullanılabilir.

Prescaler, bir timer'ın giriş frekansını bölerek zamanlama hassasiyetini ayarlamak için kullanılan bir ölçeklendirme faktörüdür. Bu sistem saat frekansı, timer'ın çalışma frekansını belirler. Ancak bazen bu frekansın doğrudan kullanılması gerekmez. Prescaler, sistem saatini bölerek daha düşük bir frekansta zamanlama yapılmasını sağlar.