#### Kodu adım adım açıklayalım:

(Hata aldiysaniz, kutuphane ekli degildir, eklemek icin oncelikle asagidaki islemleri yapin)

**LiquidCrystal\_I2C kütüphanesini Arduino'ya yüklemek için aşağıdaki adımları izleyebilirsiniz:**

1. Arduino IDE'yi açın.
2. Menü çubuğunda "Araçlar" sekmesini seçin.
3. "Kitaplık Yöneticisi"ni seçin. Bu, kütüphane yöneticisini açacaktır.
4. Kütüphane yöneticisinde "LiquidCrystal\_I2C" yazarak arama kutusuna kütüphane adını yazın.
5. Arama sonuçlarında "LiquidCrystal\_I2C by Frank de Brabander" adlı kütüphaneyi bulun ve "Yükle" düğmesine tıklayın.

Yükleme işlemi tamamlandığında, kütüphane Arduino IDE'ye başarıyla yüklenmiş olacaktır.

———————————— Kodlarin aciklanmasi———————————————————

İlk olarak, gerekli kütüphaneler Wire.h ve LiquidCrystal\_I2C.h içe aktarılıyor. Wire kütüphanesi, I2C cihazlarıyla iletişim kurmak için kullanılırken, LiquidCrystal\_I2C kütüphanesi ise LCD ekranı kontrol etmek için kullanılır. LiquidCrystal\_I2C sınıfından bir nesne oluşturuluyor. Bu nesne, LCD'nin I2C adresini (0x27) ve ekranın boyutunu (16 sütun, 2 satır) belirtir.

**potPin** değişkeni, potansiyometrenin okunacağı analog pin numarasını tutmak için tanımlanırken, **fanPin** değişkeni ise fanı kontrol etmek için kullanılacak dijital pin numarasını tutar.

**setup() fonksiyonunda:**

LCD nesnesi üzerinde **begin()** metodu çağrılarak, 16 sütun ve 2 satırlı bir ekran başlatılır.

LCD'nin ilk satırına "Sicaklik: " metni yazdırılır.

**fanPin** belirtilen pin, çıkış olarak yapılandırılır.

**loop() fonksiyonunda:**

**analogRead()** fonksiyonu, **potPin** üzerinden potansiyometreden okunan değeri alır. Bu değer, 0 ile 1023 arasında bir sayıdır.

**map()** fonksiyonu, analog okumayı 0 ile 1023 arasındaki değerden 0 ile 100 arasındaki bir sıcaklık aralığına ölçeklendirir. (Potansiyometrenin bir sıcaklık sensörünü taklit ettiği varsayılır.)

Sıcaklık değeri, LCD ekran üzerinde imleç konumunu ayarlayarak ve **print()** fonksiyonunu kullanarak yazdırılır.

Bir if ifadesi, sıcaklığın 30 dereceden büyük olup olmadığını kontrol eder.

Eğer sıcaklık 30'dan yüksekse, **analogWrite()** fonksiyonu kullanılarak **fanPin** üzerine bağlı fan kontrol edilir ve hızı 150 olarak ayarlanır (**PWM** değeri 0 ile 255 arasında).

Eğer sıcaklık 30 veya daha düşükse, fanın hızı tam hızda çalışacak şekilde **PWM** değeri 255 olarak ayarlanır.

Program, döngünün tekrarlamadan önce 500 milisaniye bekleme yapar.

Bu kodu kullanmak için, Arduino kartınızda uygun şekilde bağlantıların yapıldığından emin olmanız gerekmektedir. Örneğin, **I2C LCD**'nin doğru pinlere bağlı olduğundan, potansiyometrenin ve fanın belirtilen pinlere bağlı olduğundan emin olmalısınız.