

NECMETTİN ERBAKAN ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNELERİN İNTERNETİ DERSİ

ÖDEV-I

Ödevin Son Teslim Tarihi: 28 Nisan Pazar, 2024, saat 23:55. Sadece Uzem platformu üzerinden gönderimler kabul edilecektir. Zamanında gönderilmeyen veya başka yollarla gönderilen ödevler değerlendirilmeyecektir.

Ödevle ilişkin herhangi bir açıklama ve revizyon için lütfen <https://uzaktanegitim23bahar.erbakan.edu.tr/> linkteki ders sayfasını takip ediniz.

GİRİŞ

Bu ödevin amacı Nesnelerin İnterneti uygulamaları kapsamında uygulamalarda en çok kullanılan **sensör (sıcaklık ve nem sensörü)**, ayrıca en yaygın kullanılan geliştirme kartlarından olan (**ESP32**) ve bazı giriş ve çıkış cihazları tanımak ve bunları kullanabilmek. Ayrıca sensör bilgilerini cloud'a aktarmak (**Arduino Cloud**) şeklinde olacaktır. Bu noktada gerçekleştireceğiniz ödev yapılan sıcaklık ve nem ölçümünün sonuçlarının 2 farklı ekrana ve clouda aktarılması ayrıca clouddan gönderilen mesajların ise yine karakter LCD modülünde görüntülenebilmesi olacaktır.

Senaryo

Hava sıcaklığı ve nem ölçümlerine bağlı olarak belirli seviyeye kadar ölçülen değerin soğuk diğer değerlerin sıcak, belirli seviyeye kadar ölçülen nem değerinin düşük diğer değerlerin ise yüksek olarak ifade edildiği, yapılan ölçümlerin terminal ekranına basıldığı, LCD modülde görüntülendiği ve aynı zamanda clouda aktarıldığı, sıcak-soğuk, düşük-yüksek mesajlarının ve ayrıca clouddan gönderilen mesajların ise karakter LCD ile kullanıcıyla paylaşıldığı bir uygulama gerçekleştirilecektir. LCD modülde varsa clouddan gelen kullanıcının en son gönderdiği mesaj, sıcaklık ve nem bilgileri 2 sn aralıklarla gösterilmelidir. Yüksek sıcaklık durumlarında ayrıca devrede bulunan bir adet LED yakılacaktır. Bu LED aynı zamanda clouddan da kontrol edilebilecektir. Clouddan açılıp kapatılabilen LED clouddan kapatılsa bile yüksek sıcaklıkta yanık olarak kalacaktır. Düşük

sıcaklık söz konusu olduğunda ise clouddan kapatıldığında ya da açıldığında verilen komuta göre çalışabilecektir. LED'in nem değerine herhangi bir bağımlılığı olmayacaktır.

SPESİFİKASYONLAR

Kullanılacak olan **Simülasyon Ortamı**, **Cloud**, **Sensör**, geliştirme kartı **ESP32** ve **LCD** ekran özelliklerinin detayları aşağıdaki gibidir:

- 1- **Simülasyon Ortamı** olarak wokwi.com platformu kullanılacaktır.
- 2- [ARDUINO Cloud](https://cloud.arduino.cc/) platformu cloud hizmeti için kullanılacaktır.
- 3- Yazılan her bir kod satırı için **yorum satırı** eklenmelidir.
- 4- Sıcaklık ve Nem Sensörü: DHT22 Sensörü kullanılacaktır.
- 5- Geliştirme kartı olarak ESP32 kullanılacaktır.
- 6- Ekran olarak ise I2C özellikli LCD (16x2) LCD modülü kullanılacaktır.
- 7- Sıcaklık değerlerinde 0'ın altında kalan sıcaklıklar için çok düşük, 0 ve 25 arası için normal 25 derece üstü sıcaklıklar için ise yüksek sıcaklık mesajları verilecektir.
- 8- Nem ölçümlerinde 50 ve altı değerlerde nem düşük, diğer değerler için ise yüksek mesajı verilecektir.
- 9- Hem sıcaklık hem de nem ölçümleri için "terminale" bastırılacak format şu şekildedir:

Humidity: 49.00% || Temperature: -40.00°C

- 10- **LCD modülde ve Cloudda** görüntülenecek mesajlar ise şu şekilde olacaktır:

8.1) Örnek Senaryo - 1

2 sn boyunca

				H	U	M	I	D	I	T	Y				
			4	9	.	0	0	-	L	O	W				

2 sn boyunca

		T	E	M	P	E	R	A	T	U	R	E			
-	4	0	.	0	0	-	V	E	R	Y		L	O	W	

Sistemde **interrupt (kesme)** olmadığı düşünülerek **round robin** tekniği ile sıcaklık ve nem değerleri belirli sürelerde görüntülenecektir. Bu sebeple 2sn aralıklarla sıcaklık ve nem değerleri ile ilgili mesajlar LCD ekranda görüntülenecektir. Varsayılan durumda yani **ilk olarak nem bilgisi, ikinci olarak ise sıcaklık bilgisi ve varsa clouddan mesaj** gösterilmelidir.

8.2) Farklı bir senaryoda ise şu şekilde bir durum söz konusu olabilir (cloud'dan kullanıcı mesaj göndermişse):

Terminal çıktısı:

Humidity: 80.00% || Temperature: 80.00°C

2sn boyunca

				H	U	M	I	D	I	T	Y				
			8	0	.	0	0	-	H	I	G	H			

2sn boyunca

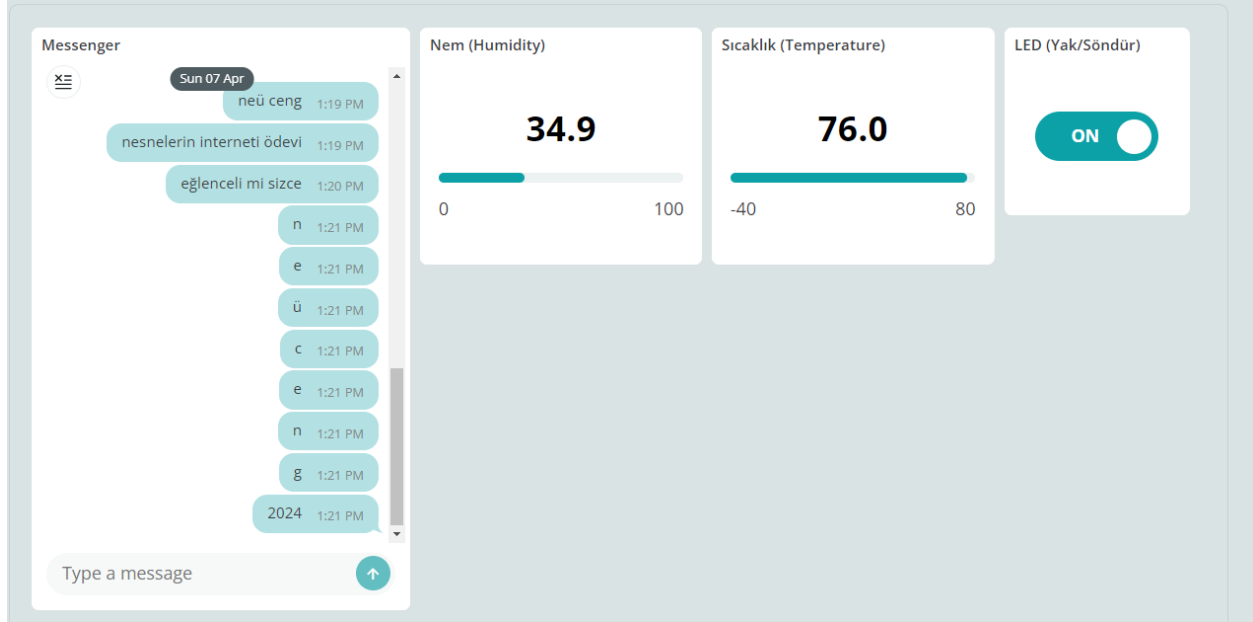
		T	E	M	P	E	R	A	T	U	R	E			
			4	0	.	0	0	-	H	I	G	H			

2sn boyunca

[illegible]

[illegible]

8.4 ARDUINO CLOUD Dashboardda bulunması gereken widget'lar ve ilgili konfigürasyonları aşağıdaki gibi olmalıdır. (Widgetlarda gösterilen değerler farklı olabilir!!! Sadece anlık bir görüntü alınmıştır.)



Adım 1: Arduino Cloud

1. Arduino Cloud üzerinden **ücretsiz bir hesap** oluşturun.
2. IoT projesi için yeni bir **“thing”** oluşturun.
3. Bu sayfada üç bölüm var. “Variables” bizim projemizde kullanacağımız değişkenleri ifade eder. “Device” kısmından kullanılacak cihaz tanımlaması ve yönetimi yapılır. “Network” kısmından ağ ayarları gerçekleştirir.

Proje için dört adet variable oluşturun:

4. **İsim:** sıcaklık, **Tür:** Temperature sensor (°C), **İzinler:** Read Only
5. **İsim:** nem, **Tür:** Relative humidity, **İzinler:** Read Only
6. **İsim:** mesaj, **Tür:** Character string, **İzinler:** Read & Write
7. **İsim:** led, **Tür:** Boolean, **İzinler:** Read & Write
8. Ardından, **“device”** ayarlamalarına geçin ve şu adımları takip ederek cihazı seçin:
 1. Add device > Third party device > ESP32 > DOIT ESP32 DEVKIT V1
9. Bu adımları tamamladıktan sonra **“secret key”** ve **“device id”** bilgilerini kopyalayın ve not defterine kaydedin!
10. Son olarak, **network ayarlarını** yapılandırın. Wi-fi name olarak **“Wokwi-GUEST”** yazın ve password kısmını boş bırakın. Secret key kutucuğuna kopyaladığınız secret key'i ekleyin.

Adım 2: Arduino Cloud Dashboard

1. Sol menüden **Dashboard**'ı seçin.
2. **“CREATE DASHBOARD”** butonuna tıklayarak yeni bir dashboard oluşturun.
3. Düzenleme bölümüne geçin ve Add butonuna tıklayarak şu adımları takip edin:

Things > proje_ismi > Tüm variable'ları seçin > **“Create widgets”** diyerek tamamlayın.

Adım 3: Wokwi Simülasyonu

Kodları kendi cloudunuza göre düzenlemek için `thingsProperties.h` dosyasını açın ve aşağıdaki kısımları kendinize göre değiştirin.

- `DEVICE_LOGIN_NAME` “kaydettiğiniz_device_id”
- `DEVICE_KEY` “kaydettiğiniz_secret_key”

Adım 4: Projeyi Çalıştırma

Artık simülasyonu başlatabilirsiniz. Simülasyonun clouda bağlanması birkaç dakika sürebilir. Simülasyon önce Wi-Fi'a bağlanacak ardından cloud ile bağlantı kuracaktır.

Dashboardda oluşturduğunuz widgetlardan simülasyondaki ledi ve lcd yi kontrol edebilir, dht sensörden gelen verileri de ilgili widgetlarda görebilirsiniz.

Karşılaşılabileceğiniz en önemli sorun clouda bağlanmanın geç olması, bu durumu lütfen göz önünde bulundurunuz uygulamalarınızda.

thinksProperties.h dosyasının içeriği:

```
#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

const char DEVICE_LOGIN_NAME[] = "16e630c3-3869-47db-a821-75a0a2e5c86f";//Sample

const char SSID[] = "Wokwi-GUEST"; // Network SSID (name)
const char PASS[] = ""; // Network password (use for WPA, or use
as key for WEP)
const char DEVICE_KEY[] = "jZBnGZPM4Gz5aFCznqItzr7gG"; // Secret
device//Sample password

void onMesajChange();
void onLedChange();

String mesaj;
CloudTemperatureSensor sicaklik;
CloudRelativeHumidity nem;
bool led;

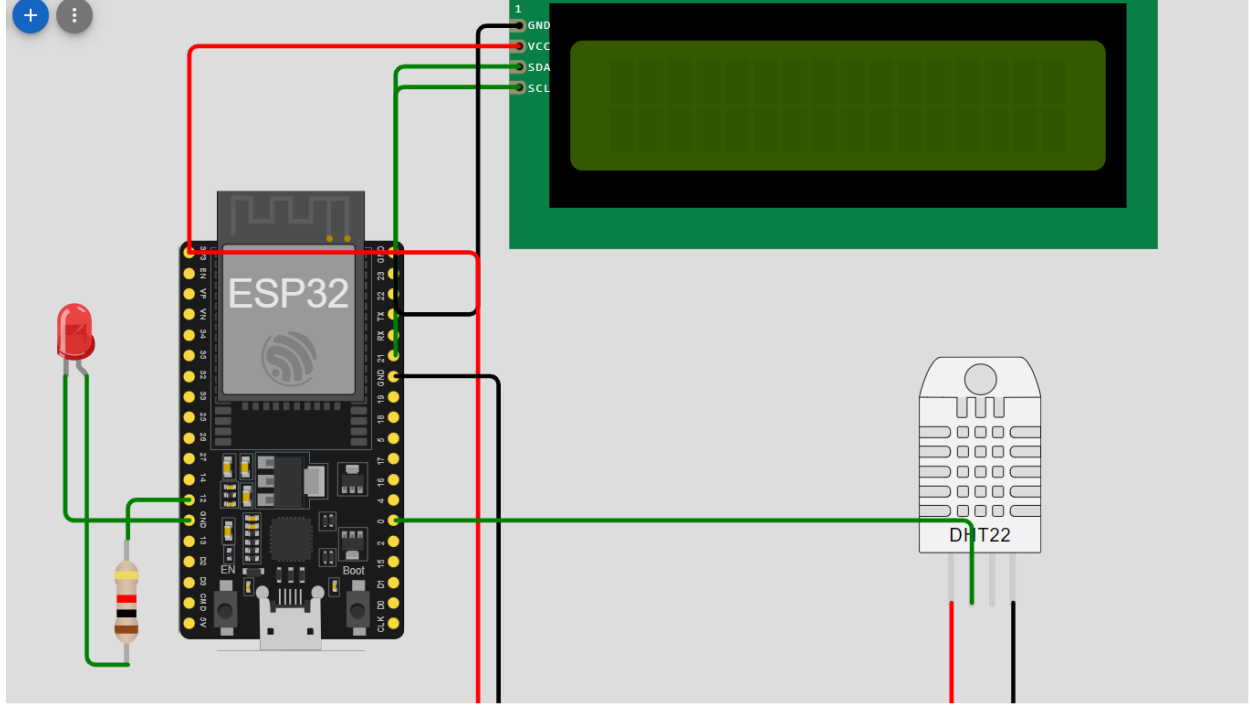
void initProperties(){

    ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
    ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
    ArduinoCloud.addProperty(mesaj, READWRITE, ON_CHANGE, onMesajChange);
    ArduinoCloud.addProperty(sicaklik, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(nem, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(led, READWRITE, ON_CHANGE, onLedChange);

}

WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
```

10. İlgili devre için aşağıdaki bağlantıları gerçekleştiriniz?



ÖDEV İLE İLGİLİ RAPOR ve KOD PAYLAŞIMI

Ödev gönderiminiz yazmış olduğunuz wokwi platformundaki kodunuzu içermelidir. Kod dosyaları: **sketch.ino** , **diagram.json** ve diğer kütüphaneler (örneğin **thinkProperties.h**) dosyasını ve bir de Readme dosyasında eklemiş olduğunuz kütüphane isimlerini, projenizi çalıştırabilmek için tüm gerekli bilgileri bir klasöre koyup sıkıştırarak göndermeniz beklenmektedir. Ayrıca projenize ait wokwi linkini mutlaka **READme** dosyanıza ekleyiniz. **Read.me** dosyanıza projenin adım adım nasıl çalıştırılması gerektiğini yazınız. Değerlendirme boyunca silmeyiniz. Projelerinizi adı-soyadı şeklinde isimlendiriniz. Gönderilen dosyanın ismi ise **OgrenciADI-SOYADI.zip** şeklinde olmalıdır.

Not: Uygulamayı geliştirirken kullandığınız her bir sensör, geliştirme kartı ve diğer yardımcı devre elemanlarının neden, nasıl kullanıldığına dair ayrıntılı incelemelerde bulununuz.

Referans kaynaklardan kullandığınız zaman mutlaka referans göstermelisiniz raporunuzda!)

Uygulamayı gerçekleştirirken farkında olmanız gereken bazı önemli sorular (Bunları raporlamanız gerekmektedir.):

Soru 1- ESP32 teknik özellikleri nelerdir, kaç girişten oluşmaktadır, her bir giriş ne işe yaramaktadır?

Soru-2- DHT22 Sensörü hangi pinleri barındırmaktadır. Bu pinler ne işe yaramaktadır? DHT22 sensörüne ait bir datasheet var mıdır? Varsa içindeki spesifikasyonlar nelerdir?

Soru-3- ESP32 ve DHT22 sensörüne ait hangi pinler hangi pinlerle bağlantı kurmalıdır, neden?

Soru 4- LCD (I2C, 16x2) modülünde I2C özelliği bize ne sağlamaktadır. I2C özelliği olmayan LCD modüller ile arasındaki fark nedir? 16x2 ne anlama gelmektedir.

Soru-5- Round Robin algoritması nedir, ne için kullanılmaktadır? Gömülü sistem projelerinde kullanımı uygun mudur?

Soru-6 ESP32 ve ARDUINO UNO arasında ne gibi farklar vardır? ARDUINO UNO ile bu proje yapsam başka hangi parçalara ihtiyacım olurdu?

Ödevinizi yaparken aşağıdaki linklerden faydalanabilirsiniz. Referans kaynaklar (Bunlar dışında da kullandığınızı veya bunları kullandığınız zaman da mutlaka referans göstermelisiniz raporunuzda)

https://gulsumturk.com/arduino_iot_project/

<https://www.kompent.com/dht22a-am2302-sicaklik-ve-nem-sensoru-183>

<https://arduinodestek.com/arduino-uno-teknik-ozellikleri-nelerdir/>

Kopya Çekme Durumu

Kopya çekilmesine karşın sıfır tolerans politikamız vardır. Kopya çekenler üniversite yönetmeliğine göre cezalandırılacaktır.

Kopya Politikası: Öğrenciler/Gruplar kavramları kendi aralarında veya öğretim elemanı veya yardımcıları ile tartışabilirler. Ancak asıl işin yapılması söz konusu olduğunda, bunun sadece öğrenci/grup tarafından yapılması gerekir. Çözümünüzü yazmaya veya yazmaya başladığınızda, yalnız çalışmalısınız. Başka bir deyişle, doğrudan bir başkasından metin kopyalıyorsanız - ister dosyaları kopyalıyor, ister başka birinin notlarından yazıyor ya da onlar dikte ederken yazıyorsanız - kopya çekiyorsunuz demektir (daha kesin olmak gerekirse, intihal yapıyorsunuz anlamına gelir). Bu, kaynağın bir sınıf arkadaşı, eski bir öğrenci, bir web sitesi, çöpte bulunan bir program listesi veya herhangi bir şey olup olmadığına bakılmaksızın doğrudur. Ayrıca, programın küçük bir bölümünde bile intihal yapmak kopya çekmek anlamına gelir. Ayrıca, yazmadığınız bir kodla başlayıp, kendi kodunuz gibi görünecek şekilde değiştirmekte kopya çekmektir. Başkasının aldatmasına yardım etmek de aldatma sayılır. Programınızı açıkta bırakmak veya oturumu kapatmadan bir bilgisayardan ayrılmak, böylece programlarınızı kopyalamaya açık bırakmak, duruma göre kopya teşkil edebilir. Sonuç olarak, sizi kesinlikle kopya çekme suçlamalarına açık bırakacağından, başkalarının programlarınızı kopyalamasını önlemeye her zaman özen göstermelisiniz. Kopya çekilme durumunu belirlemek için otomatik araçlarımız var. Kopya çeken taraflar disiplin cezasına çarptırılacaktır.