

**T.C.
MANİSA CELAL BAYAR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**İNTERNET ÜZERİNDEN
AKILLI EV OTOMASYON UYGULAMASI**

Özen SİNE

**Danışman
Doç. Dr. Yücel KOÇYİĞİT**



MANİSA-2019

**Özen
SİNE**

**İNTERNET ÜZERİNDEN
AKILLI EV OTOMASYON UYGULAMASI**

2019

Tez Sırtı Örneği

TEZ ONAYI

Özen SİNE tarafından hazırlanan "**İNTERNET ÜZERİNDEN AKILLI EV OTOMASYON UYGULAMASI**" adlı tez çalışması 03.07.2019 tarihinde aşağıdaki jüri üyeleri önünde Manisa Celal Bayar Üniversitesi Fen Bilimleri Enstitüsü **Elektrik-Elektronik Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak savunulmuş ve **oyçokluğu / oybirliği** ile başarılı olarak kabul edilmiştir.

Danışman

Doç. Dr. Yücel KOÇYİĞİT
Manisa Celal Bayar Üniversitesi

Jüri Üyesi

Prof. Dr. Sezai TAŞKIN
Manisa Celal Bayar Üniversitesi

Jüri Üyesi

Dr. Öğr. Üyesi M. Zübeyir ÜNLÜ
İzmir Yüksek Teknoloji Enstitüsü

TAAHHÜTNAME

Bu tezin Manisa Celal Bayar Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü'nde, akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Özen SİNE



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	I
ŞİMGELER VE KISALTMALAR DİZİNİ	III
ŞEKİLLER DİZİNİ.....	V
TABLO DİZİNİ	VII
TEŞEKKÜR.....	VIII
ÖZET.....	IX
ABSTRACT.....	X
1. GİRİŞ VE LİTERATÜR ÖZETİ	1
2. GENEL BİLGİLER	6
2.1. Tezin Amacı	7
3. MATERYAL VE YÖNTEMLER.....	8
3.1. Materyal.....	8
3.1.1. Kullanılan Elektronik Elemanlar	17
3.1.1.1.MSP432 Mikrokontrolör.....	17
3.1.1.2.Node MCU ESP8266.....	19
3.1.1.3.Servo Motor	20
3.1.1.4.Nextion 3,2" Dokunmatik Ekran	21
3.1.1.5.DHT11 Sıcaklık ve Nem Sensörü.....	21
3.1.1.6.MQ-6 Gaz Sensörü	22
3.1.1.7.MQ-135 Hava Kalite Kontrol Sensörü	24
3.1.1.8.12V DC Fan	25
3.1.1.9.YL-44 Buzzer.....	25
3.1.1.10.Foto Diyot(LDR)	26
3.1.1.11.Toprak Nem Sensörü	27
3.1.1.12.HC-SR501 Hareket Kontrol Sensörü.....	27
3.1.1.13.DC40-1245 Su Pompası.....	28
3.1.1.14.TEC1-12706 Peltier Termoelektrik Sistem	29
3.1.1.15.Power Led ve PWM Girişli Led Sürücü Modülü	30
3.1.1.16. 30 A 12 V DC Güç Kaynağı.....	31
3.1.2. Tasarlanan Ev Maketi	31
3.2. Yöntemler.....	34
3.2.1. Kullanılan Programlar	34
3.2.1.1.Energia Programı	34
3.2.1.2.Arduino Programı	38
3.2.1.3.Nextion Editör Programı.....	42
3.2.1.4.Eagle Programı.....	46
3.2.1.5.Solid Works Programı.....	49
3.2.2. Kullanılan Haberleşme Protokolleri	51
3.2.2.1.MQTT Haberleşme Protokolü	51
3.2.2.2.UART Haberleşme Protokolü.....	55
3.2.3. Kullanılan Kontrol Arayüzü Tasarımları.....	56
3.2.3.1.Adafruit IO Sistemi Kontrol Arayüzü.....	57
3.2.3.2.Nextion Dokunmatik Ekran Kontrol Arayüzü.....	61

3.2.4. Kullanılan Uygulama Programlama Arayüzü(API).....	69
3.2.4.1.Openweather API.....	69
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	72
5. SONUÇ VE ÖNERİLER	75
KAYNAKLAR	77
EKLER.....	79
EK A. (MSP432 ile çevresel birimler arasındaki bağlantı şemaları).....	79
EK B. (MSP432'nin çevresel birimlere çıkış vermesini sağlayan röle kartının devre şeması).....	80
EK C. (MSP432 mikrokontrolörün pin diyagramı)	81
EK D. (MSP432 ile tüm çevresel birimler arasındaki haberleşme kodları)...	82
EK E. (Arduinio'da MQ-135 sensörünün MQTT haberleşme kodları).....	87
EK F. (Arduinio'da DHT-11 sensörünün MQTT haberleşme kodları).....	89
EK G. (Adafruit IO kontrol arayüzü veri gönderim ve alım kodları).....	92
EK H. (Openweather API'dan mevsimsel sıcaklık verisi alım kodları)	96
EK I. (Ev maketinin Solid Works'de çizilen teknik resimleri).....	99
EK İ. (Ev maketi görselleri).....	107
ÖZGEÇMİŞ	109

SİMGELER VE KISALTMALAR DİZİNİ

MQTT	Message Queuing Telemetry Transport
WIFI	Wireless Fidelity
RF	Radio Frequency
ARGE	Araştırma ve Geliştirme
MCU	Microcontroller Unite
WIMAX	Worldwide Interoperability for Microwave Access
IEC	International Electrotechnical Commission
ISO	International Standardization for Organization
IEEE	Instute of Electrical and Electronics Engineers
LAN	Local Area Network
MIMO	Multiple Input Multiple Output
OEM	Original Equipment Manufacturer
SDK	Software Development Kit
LDR	Light Dependent Resistor
PWM	Pulse Width Modulation
CPU	Central Processing Unit
FPU	Fast Processor Up
DSP	Digital Signal Processing
PCB	Printed Circuit Board
I2C	Inter Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
IDE	Integrated Development Environment
HMI	Human Machine Interface
IoT	Internet of Things
RAM	Random Access Memory
TTL	Transistor to Transistor Logic
NTC	Negative Temperature Coefficient
FTDI	Future Technology Devices International
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TCP	Transmission Control Protocol

RX	Receive X
TX	Transmit X
RSS	Real Simple Syndication
IFTTT	If This Then That
API	Application Programming Interface
IT	Information Technology
URL	Uniform Resource Loader



ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. Akıllı ev otomasyonu örneği.....	2
Şekil 3.1. Projede kullanılan elektronik malzemeler arasındaki veri alış verişi .	17
Şekil 3.2. MSP432P401x mikrokontrolör kiti.....	18
Şekil 3.3. NodeMCU ESP8266 pin diyagramı	19
Şekil 3.4. Servo motor boyutları	20
Şekil 3.5. Servo motor çalışma sinyali.....	20
Şekil 3.6. Nextion 3,2” dokunmatik ekran bağlantı şeması	21
Şekil 3.7. DHT-11 görseli	21
Şekil 3.8. DHT-11 pin diyagramı ve devre şeması	22
Şekil 3.9. MQ-6 görseli.....	23
Şekil 3.10. MQ-6 devresi	23
Şekil 3.11. MQ-135 görseli.....	24
Şekil 3.12. MQ-135 devresi	24
Şekil 3.13. Fan görseli.....	25
Şekil 3.14. Buzzer görseli	25
Şekil 3.15. YL-44 buzzer devresi.....	26
Şekil 3.16. LDR görseli.....	26
Şekil 3.17. Toprak nem sensörü görseli	27
Şekil 3.18. Hareket algılama sensörü görseli	28
Şekil 3.19. Su pompası görseli	28
Şekil 3.20. Peltier çalışma mantığı.....	29
Şekil 3.21. Peltier görseli ve termal kamera ile görüntülenmesi.....	29
Şekil 3.22. Power led sürücü kartı	30
Şekil 3.23. Güç kaynağı görseli	31
Şekil 3.24. Solid Works programında tasarlanan ev maketinin çizimleri.....	32
Şekil 3.25. Solid Works programında dış kapı mekanizmasının çizimleri	33
Şekil 3.26. Energia programı tanıtımı 1	35
Şekil 3.27. Energia programı tanıtımı 2	36
Şekil 3.28. Energia programı tanıtımı 3	37
Şekil 3.29. Energia programı tanıtımı 4	37
Şekil 3.30. Energia programı tanıtımı 5	38
Şekil 3.31. Arduino programı tanıtımı 1	39
Şekil 3.32. Arduino programı tanıtımı 2	40
Şekil 3.33. Arduino programı tanıtımı 3	40
Şekil 3.34. Arduino programı tanıtımı 4	41
Şekil 3.35. Arduino programı tanıtımı 5	42
Şekil 3.36. Nextion Editör programı tanıtımı 1	43
Şekil 3.37. Nextion Editör programı tanıtımı 2.....	44
Şekil 3.38. Nextion Editör programı tanıtımı 3.....	44
Şekil 3.39. Nextion Editör programı tanıtımı 4.....	45
Şekil 3.40. Eagle programı tanıtımı 1	46
Şekil 3.41. Eagle programı tanıtımı 2	47
Şekil 3.42. Eagle programı tanıtımı 3	47
Şekil 3.43. Eagle programı tanıtımı 4	48
Şekil 3.44. Solid Works programı tanıtımı 1	50
Şekil 3.45. MQTT kavramları arasındaki ilişki	52
Şekil 3.46. MQ-135’in MQTT ile veri göndermesi	54

Şekil 3.47. DHT-11'in MQTT ile veri göndermesi	54
Şekil 3.48. UART bağlantı şeması	56
Şekil 3.49. Adafruit kontrol arayüz tanıtımı 1	59
Şekil 3.50. Adafruit kontrol arayüz tanıtımı 2	59
Şekil 3.51. Adafruit kontrol arayüz tanıtımı 3	60
Şekil 3.52. Adafruit kontrol arayüz tanıtımı 4	60
Şekil 3.53. Adafruit kontrol arayüz tanıtımı 5	61
Şekil 3.54. Nextion Editör kontrol arayüz tanıtımı 1	62
Şekil 3.55. Nextion Editör kontrol arayüz tanıtımı 2	62
Şekil 3.56. Nextion Editör kontrol arayüz tanıtımı 3	63
Şekil 3.57. Nextion Editör kontrol arayüz tanıtımı 4	63
Şekil 3.58. Nextion Editör kontrol arayüz tanıtımı 5	64
Şekil 3.59. Nextion Editör kontrol arayüz tanıtımı 6	64
Şekil 3.60. Nextion Editör kontrol arayüz tanıtımı 7	65
Şekil 3.61. Nextion Editör kontrol arayüz tanıtımı 8	65
Şekil 3.62. Nextion Editör nihai arayüz tasarımı 1	66
Şekil 3.63. Nextion Editör nihai arayüz tasarımları 2	67
Şekil 3.64. Nextion Editör kontrol arayüz tanıtımı 9	68
Şekil 3.65. Openweather API başlangıç ekranı	70
Şekil 3.66. Openweather API anahtar kod ekranı	70
Şekil 3.67. Openweather API'dan kullanıcıya gönderilen veri bulutu	71
Şekil 4.1. Ev ortamından Adafruit IO'ya veri gönderimi	72
Şekil 4.2. Ev ortamındaki CO ₂ miktarı değişim grafiği	73
Şekil 4.3. Sıcaklık ve nem değişiminin aynı ekran üzerinde görüntülenmesi	73
Şekil 4.4. Energia debug ekranında mevsimsel sıcaklık verilerinin takibi	74
Şekil 4.5. Energia debug ekranında nabız değeri verilerinin takibi	74

TABLO DİZİNİ

	Sayfa
Tablo 3.1. Kontrol edilen parametre ve elektronik eleman ilişkisi	8-16
Tablo 3.2. Openweather API'ya erişim için oluşturulan linkler	71



TEŐEKKÜR

Her konuda beni cesaretlendiren ve alıőmamın her aőamasında bana destek olan sevgili anneme, babama ve ablama yureka teőekkür ederim.

Yüksek lisans eđitimimin baőından sonuna kadar yanımda olan ve bu tez alıőmasında da önemli katkıları bulunan, gerek donanım gerekse yazılım bilgisini benden esirgemeyen deđerli arkadaşım Mert TOKAY'a teőekkür ederim.

Ayrıca bu süreç boyunca her türlü desteđi benden esirgemeyen, tecrübesi ile beni aydınlatan, deđerli öneri, görüş ve yönlendirmeleri için danışman hocam Sayın Do.Dr. Yücel KOYIĐIT'e teőekkürü bor bilirim.

Özen SİNE
Manisa, 2019



ÖZET

Yüksek Lisans

İNTERNET ÜZERİNDEN AKILLI EV OTOMASYON UYGULAMASI

Özen SİNE

Manisa Celal Bayar Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Yücel KOÇYİĞİT

Teknolojinin ilerlemesiyle beraber ev otomasyonu kontrol sistemlerinin kullanımı artarak yaygınlaşmaktadır. Akıllı ev otomasyon sistemleri insanların yaşam kolaylığı ve güvenliğe duydukları ihtiyaçları karşılamakla birlikte enerji sarfiyatı yönünden tasarruf sağlayarak ekonomik açıdan da bir kazanç elde edilmesi imkânı sunmaktadır.

Bu çalışma gelecekteki tam otomasyonlu sistemlere örnek teşkil edebilmek amacıyla ev ortamındaki sıcaklık kontrolünü ve bahçe sulamasını, anlık mevsimsel sıcaklıklara göre ayarlanabilmesini sağlamaktadır. Ayrıca evde yatağa bağımlı bir bireyin aniden bir sağlık problemiyle karşılaşması durumunda bir hasta takip senaryosu da gerçekleştirilmektedir.

Ev otomasyonu sisteminde kontrol edilmek istenen her bir parametreye ait sensör, sistemin konforunu artırması için mikrokontrolörle kablosuz haberleşmektedir. Alıcı ve verici arasındaki kablosuz veri aktarımı ESP8266 modülüyle sağlanmaktadır. Sistemin hem verimli veri işleme hem de gelişmiş düşük güç tüketimi konusunda sorunsuz çalışabilmesi için MSP432 mikrokontrolör tercih edilmiştir.

İnternet üzerinden kullanıcı bilgileri girildikten sonra ev otomasyon sistemine erişim sağlanmaktadır. İnternet arayüz programı ile ev ortamının haberleşebilmesi için MQTT haberleşme protokolü kullanılmaktadır. Evdeki her bir parametre anlık olarak gözlemlenebilmekte ve istenildiği zaman müdahale edilebilmektedir. Ayrıca veriler sürekli olarak kaydedilmekte ve geçmişe dönük olarak kontrolü yapılabilmektedir. Her bir parametrenin istatistikleri grafiksel olarak çizdirilebilmektedir.

Ev ortamını birebir yansıtabilmek ve kontrol edilen parametrelerin rahatça gözlemleyebilmek için tez çalışmasında ev maketi tasarımı yapılmıştır. Bu maket sayesinde istenilen işlemlerin gerçekleşmesinin ne derecede başarılı olduğu görülmüştür.

Bu tez çalışması ile günlük yaşamda ev ortamında gerçekleştirilen bazı işlerin internet üzerinden akıllı cep telefonu, tablet veya Wifi desteği olan TV üzerinden kablosuz olarak ev güvenliği sağlanabilmektedir. Aynı zamanda sulama, ısı, ışık düzeyi, vb. parametreler internet üzerinden periyodik veya isteğe bağlı kontrol edilerek insan hayatını kolaylaştırmaktadır.

Anahtar Kelimeler: (Akıllı Ev Otomasyonu, Kablosuz Haberleşme, İnternet Üzerinden Kontrol)

2019, 109 sayfa

ABSTRACT

Master Licence Thesis

SMART HOME AUTOMATION VIA THE INTERNET

Özen SİNE

Manisa Celal Bayar University
Graduate School of Applied and Natural Sciences
Department of Electric-Electronic Engineering

Supervisor: Assoc. Prof. Dr. Yücel KOÇYIĞIT

Smart Home Automation Systems are becoming increasingly common with development of technology. Smart home automation systems meet the needs of people for ease of life and security, but also save money in terms of energy consumption offers the opportunity to gain an economic point of view.

In this study, temperature control in the home environment and garden irrigation can be adjusted according to the seasonal temperatures in order to set an example for future fully automated systems. In addition, a patient follow-up scenario is performed in case a bed-dependent person suddenly encounters a health problem at home.

The sensor of each parameter that is wanted to be controlled in the home automation system communicates wireless with the microcontroller to increase the comfort of the system. The wireless data transmission between the receiver and transmitter is provided by the ESP8266 module. The MSP432 microcontroller has been chosen to ensure that the system runs smoothly for both efficient data processing and improved low power consumption.

After entering user information over the internet, access to home automation system is provided. MQTT communication protocol is used for the communication of the home environment with the internet interface program.

Each parameter in the house can be observed instantly and intervened at any time. In addition, the data is continuously recorded and retrospectively controlled. The statistics of each parameter can be plotted graphically.

In order to reflect the home environment and observe the controlled parameters comfortably, a home model was designed in the thesis. Thanks to this model, it has been seen how successful the desired transactions are performed.

With this thesis, some works performed in the home environment in daily life can be provided wirelessly over the internet via smart mobile phone, tablet or TV with Wifi support. At the same time irrigation, heat, light level, etc. parameters can be controlled periodically or on demand via the internet to facilitate human life.

Keywords:(Smart Home Automation, Wireless Communication, Control via Internet)

2019, 109 pages

1. GİRİŞ VE LİTERATÜR ÖZETİ

Akıllı ev tanımı, genel olarak herşeyi kontrol edebilen merkezi kontrol sistemine sahip olan ev şeklinde yapılmaktadır. Daha ayrıntılı olarak bahsedecek olursak; akıllı bir ev, çeşitli sistemleri bir arada koordineli bir şekilde kullanarak teknik performansı, yatırımları ve işletim maliyetlerini düşürmeyi, esneklik kazandırmayı en üst seviyeye taşıyan yapıdır[1].

Günümüzde teknolojinin ilerlemesiyle insanların yaşam kalitesi artarken, konfora karşı olan bakış açıları da sürekli olarak değişmektedir. Teknolojinin insanlara sağladığı olanaklarla eskiden zahmet ve zorlukla yapılan bazı işler günümüzde herhangi insan faktörü bulunmayan otomasyon sistemleri ile rahatlıkla gerçekleştirilebilmektedir.

Uzun yıllardan beri otomasyon sistemleri farklı alanlarda kullanılmaktadır. Fakat son yıllarda birçok probleme karşı çözüm sunması ile ev otomasyonu önem kazanmıştır. Yapılması gerekli bazı rutin işlemlerin otomatik olarak gerçekleştirilmesinin sağladığı avantajları dikkate alan birçok firma konu ile ilgili Ar-Ge çalışmaları başlatmış ve bu çalışmalardan başarılı sonuçlar almışlardır. Bu konuda yapılan çalışmalar akıllı ev teknolojisi olarak nitelendirilmektedir. Üzerinde çalışılan bu teknoloji ile insanlara çok geniş ölçüde farklı hizmetler sunabilmesinden dolayı yakın gelecekte daha fazla kullanıcı kitlesine sahip olacağı düşünülmektedir. Akıllı Ev teknolojisi ile insanların, günlük yaşamlarında modern teknolojiler kullanılarak oluşturulan otomasyon sistemleri sayesinde güvenlik, iletişim, konfor, tasarruf, kontrol vb. birçok alanda hizmet almaları mümkündür[2].

Akıllı evlerde çeşitli mekatronik uygulamalar gerçekleştirilebilmektedir. Bunlardan bazıları aydınlatma kontrolü, güvenlik sistemleri, hareket dedektörleri, iklim kontrolü ve benzerleridir. Ayrıca günümüzde her evde ve yanımızda sürekli taşıdığımız akıllı telefonlarda bulunan internet sayesinde akıllı ev uygulamaları son zamanlarda internet üzerinden kontrol edilmeye başlanmıştır.

Akıllı ev otomasyonu üzerine daha önce yapılan çalışmalara bakıldığında, yapılan çalışmada insan yaşamını kolaylaştırma fikri esas alındığı için kullanım kolaylığı ve konfor alanları üzerine diğerlerinden farklı olarak daha fazla yer verilmiş ve daha fazla pratik çözüm elde edilmiştir. Ev maketi üzerindeki dedektörlerden alınan veriler kablosuz olarak MCU(alıcı)'ya iletildiği için kablo maliyeti ortadan kalktığı gibi kablo kalabalığından da ev ortamı arınmıştır. Ayrıca ileride kontrol edilmesi istenen parametre sayısı arttırılmak istenirse herhangi bir tadilata gerek kalmadan yeni dedektörler sisteme eklenebilecektir.

Diğer biri ise ev ortamında veya dışında internet erişimi olması kaydıyla akıllı cep telefonları, tablet, televizyon, bilgisayar, vb. gibi elektronik cihazlarla kolay bir şekilde WEB serveri üzerinden sistem gözlenebilecek, kayıt altına alınabilecek ve kontrol edilebilecek olmasıdır. Bunun için gerekli tek adım, kullanıcı bilgilerini girerek kontrol sayfasına ulaşması gerekmektedir. Ayrıca ev ortamında MCU'ya bağlı bir dokunmatik ekranla da sistem kontrol edilebilecek ve ana kontrol ünitesi görevi görecektir.

İnternet üzerinden akıllı ev otomasyonu uygulamasındaki ana amaç, güvenliğin ve konforun arttırılması, kullanım kolaylığının ve enerji tasarrufunun sağlanmasıdır. Dolayısıyla Şekil 1.1 'de otomasyon sisteminin evin her alanında kullanılabileceği görülmektedir.



Şekil 1.1.Akıllı ev otomasyon örneği

Akıllı ev otomasyon sistemleri, evlerde kullanılan elektrik cihazları ve elektronik cihazların kullanıcı dostu bir arayüzle uyumlu şekilde çalışmasını sağlamaktadır. Bu sistemler konfor ve ergonomi sağlamalarının dışında enerji tasarrufunu da beraberinde getirirler; hatta belirli bir zaman diliminden sonra maliyetlerini amorti etmektedirler. Özellikle yaşlı ve özürlü insanlar için yaşam kalitesini arttırmaktadırlar. Örneğin birçok fiziksel fonksiyondan yoksun biri otomasyon sistemi sayesinde evdeki cihazlarını çalıştırıp durdurabilmekte (açıp kapatabilmekte), gerekli ayarlamaları yapabilmektedir[3].

Akıllı ev otomasyonları ilk olarak 1980'li yıllarda Amerika Birleşik Devletleri'nde ve Japonya'da ışıklandırma ve ısıtma sistemlerinin otomatiğe geçmesi ile başlamıştır. Daha sonra güvenlik, haberleşme ve görüntü sistemlerinin birleşimi ile tam otomatik hale gelmiştir. Kendi içinde bağımsız şekilde bir evdeki birçok değişkeni izleyen, denetleyen ve kaydeden bir otomasyon yönetim sistemi olan akıllı evler, bina içinde ve dışında değişik noktalardaki algılayıcılar ile farklı pozisyonlara sahip birimlerin denetlenmesini sağlayan çıkış noktalarından oluşur[4].

Türkiye'deki ilk uygulama ise 1984 yılında Yeşilköy Havalimanı'nda yapılmıştır. Bu sistem aktif kontrol fonksiyonları olmadığı için yalnızca izlemeye yönelik olarak tasarlanmıştır[5].

İnternet erişimi olduğu sürece akıllı telefonlar aracılığıyla, internet erişiminin olmadığı zamanlarda ise evdeki ana kontrol ünitesinden kontrol sistemine her yerden erişilmektedir. Dolayısıyla evin dışındaki herhangi bir noktadan evdeki klimalar açılabilen; evin ışıkları kapatılabilen; alarm sistemi devreye sokulabilen veya evdeki durumdan(elektrik, bahçe sulaması, ortamın sıcaklığı, .vs.) haberdar olunabilmektedir. Evin sıcaklığı kullanıcı eve dönmeden istenilen derecede ayarlanabilmektedir. Bahçedeki bitkilerin düzenli olarak sulanması sağlanabilmektedir.

Kablolu ve kablosuz olmak üzere ev otomasyon sistemlerinde kullanılan birçok teknoloji vardır. Bu teknolojilerin hepsi farklı özelliklere sahiptir; ilki 1975'de bulunan elektronik aletlerin haberleşmesinde kullanılan en eski haberleşme protokolü X10'dur[6].

İlerleyen teknoloji ile bazı dünya çapında firmalar piyasaya yeni haberleşme protokolleri çıkarmış ama çok azı başarılı olabilmıştır. Günümüzdeki ev otomasyon kontrol sistemleri, kablosuz ağ imkânı sağlayan yapılar sayesinde tüketicinin ilgisini çekeceği düşünülmektedir. Çünkü kablosuz cihaz ve sensörlerin kurulumları pratik ve ücretleri daha düşüktür. Ayrıca kablosuz haberleşme imkânı sunan sistemler kablolu sistemlere göre daha ergonomiktir.

Örneğin Bluetooth teknolojisi, taşınabilir veya sabit cihazların haberleşmesini sağlamak ve ileride kablolu iletişimin yerini almak üzere düşünülen kısa menzilli iletişim teknoloji olarak tasarlanmış ve bu gerçekleşmiştir. Bluetooth teknolojisinin temel özellikleri; sağlamlık, düşük güç ve düşük maliyetli olmasıdır[7].

Wifi'nin sağladığı imkânla evler, kafelerde hatta toplu taşıma araçlarında bile kablosuz haberleşme kullanımı olağan hale gelmiştir. WiMax (4G)'da Wifi'nin yaptığını, bir şehir boyutunda yapmaktadır. Ev otomasyonu haberleşme protokolleri için öneriler diğer endüstrilerden gelmektedir. IEC (International Electrotechnical Commission) ve ISO (International Standardization for Organization) örgütleri ev otomasyonu için dünya çapında bir standart oluşturulması gerektiğini savunmaktadır. Protokol seçimi, hem ağ performansını hem de cihaz maliyetlerini etkilediği için önemlidir[8]. Uluslararası standartlar geliştiren ve onaylayan diğer bir kurum olan IEEE (Institute of Electrical and Electronics Engineers)'nin 802.X LAN haberleşme protokolleri bu enstitünün bilinen en iyi standartlarıdır[9].

IEEE'nin piyasaya sunduğu en son standardı olan 802.11ac, 2014'de tasdiklenmiştir. Piyasada en çok bulunan 802.11n'e göre gönderilen veri boyutu büyük çapta güncelleştirilmiş ve veri gönderim hızı 1,3Gbit/s'e kadar çıkabilmektedir. 802.11n'den farklı olarak yalnızca 5 GHz frekans seviyesinde kullanılabilir. 802.11ac, 802.11n'ye oranla daha iyi ve kullanılabilir bir bant ağ genişliği tanısa da, 2,4 GHz'in geniş dalga boyunu kullanamama dezavantajına sahiptir. Dolayısıyla alıcı ve verici noktaları arasında kalın bir engel olması durumunda 802.11ac kablosuz ağ standardının verimi düşer. 4K çözünürlüğüne sahip videoların daha da insan hayatında yer edinmesiyle 802.11ac'nin kullanım alanı daha da artacaktır.

IEEE 802.11n standardizasyonu, veri gönderimi için MIMO (Multiple Input Multiple Output) alt yapısını kullanır. Böylece büyük boyuttaki veri iletimi gerçekleştirilebilmektedir. Yani benzer uzaklıktaki alıcıya daha hızlı veri gönderilebilir. Haberleşme kanallarının 20 MHz ve 40 MHz'e kadar çıkartılması ve veri iletimine olanak tanıyan anten sayısını dört olacak şekilde yükseltilmesi ile 600 Mbit brüt dosya gönderim hızına erişilebilmektedir. Bu kablosuz ağ standardını destekleyen aygıtlar 2,4 ve 5 GHz bandlarını eş zamanlı olarak kullanabilmektedir. Aynı zamanda her bir kanal 300 Mbit/s'e kadar veri iletim hızına müsaade eder. Söz konusu hıza erişilebilmesi için dört taneye kadar akım beslemesine ihtiyaç vardır. Bu durum da sadece gerekli sayıda alıcı ve gönderici antenin sisteme eklenmesiyle gerçekleştirilebilmektedir.

Kablosuz ağ ile kontrol edilen ev otomasyon sistemleri kullanıcı tarafından yönetilmek istenen her bir parametre için bir sensör ve bir hareket sağlayıcı mekanizmadan oluşur. Bu sensörlerin kablosuz olarak mikrokontrolörle haberleşebildiği, uzaktan takibinin anlık olarak yapılabildiği ve verilerin sürekli olarak kaydının yapılabildiği dinamik ve kompakt bir yapıya sahiptir. Böylece hem kullanıcının konfor ve rahatlığı sağlanırken hem de enerjiden tasarruf etmesi sağlanır. Ev otomasyonu sistemlerinde en fazla tercih edilen kablosuz haberleşme çeşitleri ZigBee ve Wifi'dir.

ZigBee kısa menzilli ve düşük veri transfer hızına sahip kablosuz haberleşme protokol kümesi olarak tanımlanmıştır. ZigBee'nin temel olarak düşük veri transfer oranlarında, düşük maliyetli ve uzun batarya ömrü özelliklerinin barınacağı uygulamalarda kullanılmıştır. ZigBee standardı, yarı iletken endüstrisi, yazılım geliştiriciler, orijinal parça üreticilerden (OEM) oluşan yüzlerce şirketin üyesi olduğu ZigBee Alliance tarafından geliştirilmiştir[10].

2. GENEL BİLGİLER

Günümüzdeki akıllı ev sistemleri genellikle kullanıcının doğrudan veya dolaylı olarak müdahil olduğu kısmi otomasyonlu yapılardır. Gelecek yıllarda teknolojik alt yapının gelişmesiyle bu yapıların tam otomasyonlu sistemlere dönüşerek hiçbir dış faktör (insan müdahalesi) etkisi olmadan gerçekleşebileceğine şüphe yoktur. Çalışmada tasarlanan ev maketindeki deneme odasının sıcaklık kontrolü ve bahçe sulaması, anlık mevsimsel sıcaklıklara göre ayarlanabilmektedir.

Dünya Sağlık Örgütü'nün belirlemiş olduğu oturma odası için belirlemiş olduğu sıcaklık değeri 21 °C' dir. Ancak toplumsal ve yöresel farklılıklardan dolayı bu değer değişebilmektedir. Manisa ili ve çevresinde bu değer 23 °C civarlarındadır. Mevsimsel sıcaklık değerleriyle bu belirlenen değer arasındaki fark arttıkça yakıt ve enerji sarfiyatı artmaktadır. Bu yüzden kullanıcı konforu ve tasarrufu düşünüldüğünde ev ortamının sıcaklığının kararlılığı çok önemlidir. Manisa genelindeki mevsimsel sıcaklık değişimlerine göre ısıtma ve soğutma sisteminin otomatik olarak devreye girmesi kullanıcıya büyük avantajlar getirmektedir. Örneğin daha az yakıt ve elektrik sarfiyatıyla ev sıcaklığı ideal düzeyde tutulabilmekte ve kullanıcının bunun için ekstra bir önlem almasına gerek kalmamaktadır. Mevsimsel sıcaklık değerlerine göre sistem sürekli olarak tepki gösterebilmektedir. Ev maketindeki deneme odasının sıcaklık değeri 23 °C' nin altına düştüğünde Peltier devreye girerek ev ortamını ısıtmakta, 23 °C' nin üstüne çıktığında ise fan devreye girerek ev ortamını soğutmaktadır.

Bitkiler olumsuz çevre koşullarına (besin maddesi eksikliği, su eksikliği, düşük veya yüksek sıcaklık, ultraviyole ışınları, tuzluluk, hastalık ve zararlılar, vb. gibi) maruz kaldıklarında gelişmeleri olumsuz etkilenir. Bu duruma bitkisel stres denir. Stres sonucu bitkilerde ortalama verim kaybı % 65-90'ı bulabilmektedir. Manisa genelinde yetişen bahçe bitkilerinin çoğu (domates, biber, marul, vb.) 35 °C' nin üzerinde sıcaklığa maruz kaldığında strese girmeye başlamaktadır. 45 °C' in üzerindeki sıcaklık değerleri ise öldürücü etkiye sahiptir. Dolayısıyla mevsimsel sıcaklıklar 35 °C' nin üzerine çıktığında kullanıcının herhangi bir zararla karşılaşmaması için bahçe sulaması otomatik olarak devreye girmektedir.

Çalışmada ayrıca *Gerentechnology* (yaşlılar için teknoloji) kavramı kapsamında evde bakıma muhtaç veya sürekli yatağa bağımlı bir bireyin bulunması ve bu bireyin beklenmedik herhangi bir sağlık sorunuyla karşılaşabilme senaryosu üzerinde durulmuştur. Senaryoda bakıma muhtaç bireyin sürekli olarak nabız takibi yapılmaktadır. Bireydeki anormal nabız değişikliklerine göre (aşırı düşmesi veya artması durumunda) sistemin acil sağlık ekibine veya yakınına eposta yoluyla haber verilmektedir. Böylece en kısa sürede hastaya müdahale edilerek can güvenliğinin sağlanması amaçlanmıştır.

Normal sağlıklı bir yetişkin bireyde kalp dakikada 60-100 arasında atmaktadır. Yüksek nabız; kalp yetmezliği, yüksek ateş, tifo, tiroit, guatr gibi pekçok hastalık sonucu gerçekleşebilmektedir. Egzersiz, spor, koşu, depar atma ve zorlamada nabızın yükselmesine neden olmaktadır. Nabızın sürekli olarak dakikada 100 ve üzerinde seyretmesi kalp için yüksek risk oluşturmaktadır.

Düşük nabız; beyin kanaması, tümör, kalp hastalıkları, hormonal dengesizlikler sonucu meydana gelebilmektedir. Yine yaşlılık, doğuştan gelen kalp rahatsızlıkları, bazı mineral eksiklikleri, uyku apnesi ve kullanılan ilaçlarda düşük nabza neden olabilmektedir. Dakikada nabızın sürekli olarak 40 ve altında atması sonucu, kalbin yeterince kan pompalayamamasından dolayı vücudun yeterince oksijenlenememesine neden olduğu için dokular zarar görmektedir. Bu durumda baş dönmesi, baygınlık, terleme, sinir sistemi bozuklukları gerçekleşebilmektedir. Gerekli sağlık müdahalesinin yapılamaması durumunda ölümle bile sonuçlanabilmektedir.

2.1. Tezin Amacı

Amaç, rutin hayatta ev ortamındaki bazı faaliyetlerin, alıcılar ve vericiler (sensörler) aracılığıyla herhangi bir dış faktör (insan, canlı, vb.) karıştırılmadan kullanıcının isteğine göre otomatik gerçekleştirilebilmesidir. Yapılan "İnternet Üzerinden Akıllı Ev Otomasyonu" uygulaması, kullanıcı isteklerini otomatik ya da önceden ayarlanmış şartlara göre gerçekleştirilebildiğinden insan hayatını oldukça yaşanılabilir hale getirerek tam otomasyonlu sistemlere örnek teşkil edebilmesidir.

3. MATERYAL VE YÖNTEMLER

3.1. Materyal

Tasarlanan akıllı ev maketinde kontrol edilecek her parametre için, elektronik sektöründe kullanılan uygun boyutlarda malzemeler araştırılmıştır. İstenilen özellikleri taşıyan elektronik elemanlar, ergonomi ve maliyet açısından en uygun olanları belirlenerek tedarik edilmiştir. Söz konusu elektronik malzemelerin listesi kontrol edecekleri parametreler Tablo 3.1’de gösterilmiştir.

Tablo 3.1. Kontrol edilen parametre ve elektronik eleman ilişkisi

S.N.	Elektronik Eleman	Kontrol Ettiği Parametre	Çalışma Mantığı
1	MSP432 Kontrolör (Denetleyici, Geliştirme Kiti, Alıcı)	Şistemi denetlemektedir. Kontrol edilen her bir parametre sensörünü sürekli olarak denetlemektedir.	<ul style="list-style-type: none">• Sistemde kontrolör-denetleyici olarak görev yapmaktadır.• Her vericiden (sensörden) gelen veriyi yorumlamaktadır.• Programcı-kullanıcı tarafından yazılan algoritmaya göre gerekli çıkışları vererek istenilenler doğrultusunda sistemin tepki vermesini sağlamaktadır.• Yazılım geliştirme seti (SDK) ve zengin araç seti ile kullanımı, kolay geliştirme ortamı sunması büyük avantaj sağlamaktadır.• Kablosuz bağlantı çözümlerini temel alan uygulamalara yüksek hassasiyetli analog ve bellek uzantısı eklemelerine izin vermektedir.
2	NodeMCU ESP8266 ve ESP8266 (Wifi Modülü)	Alıcı ve vericiler arasındaki kablosuz haberleşmeyi sağlamaktadır.	<ul style="list-style-type: none">• Üzerinde ESP 8266 denetleyicisi bulunan zengin pin çıkışı sağlayan usb konektörü bulunan NodeMCU kiti, MSP432’ye bağlantısı UART ile yapılmıştır. Çevre

			<p>parametrelerini kontrol eden sensörlere bağlanan harici ESP8266 denetleyicileri ile kablosuz haberleşmesini sağlayan Wifi modülüdür.</p> <ul style="list-style-type: none"> • İnternet ortamından mevsimsel sıcaklık değerlerinin çekilmesini sağlamaktadır. • Veri akışını kablosuz olarak gerçekleştirerek sistemin verimli bir şekilde çalışmasını sağlamaktadır. • Kablo yığından kullanıcıyı kurtarması oldukça avantaj sağlamaktadır.
3	Servo Motor (Özel Açısal Motor)	Ev maketindeki dış kapının açılıp kapanmasını kontrol edilmesini sağlamaktadır.	<ul style="list-style-type: none"> • Programcı-kullanıcın onay vermesi halinde kapının açılması veya kapanması için uygun açıda dönmektedir. • Açma ve kapama işlemlerini gerçekleştirmek için kapıya yön verecek şekilde montajı yapılmıştır. Bunun için motor miline ek dişliler ve donanımlar bağlanmıştır. • İstenilen açıda dönebilmesi kullanıcıya avantaj sağlar. • Boyutunun küçük olması ergonomi ve montajı açısından büyük avantaj sağlamaktadır.
4	3.2” Nextion Dokunmatik Ekran Ana Kontrol Ünitesi (Kontrol ve Görselleştirme Arayüzü)	Ev maketi üzerine yerleştirilerek her bir parametrenin programcı-kullanıcı tarafından manuel olarak kontrol edilmesini sağlamaktadır.	<ul style="list-style-type: none"> • “Nextion Editör” programı vasıtasıyla kontrol edilecek parametre sayısı ve cinsine göre kullanıcı ekran arayüzü oluşturulmuştur. • Ekran MSP432’ye bağlanarak kontrolör aracılığıyla ev ortamındaki her bir parametrenin kontrol edilmesi sağlanmaktadır.

			<ul style="list-style-type: none"> • Acil veya internet olmadığı zamanlarda ana kontrol ünitesi olarak sistemin çalışmasını sağlamaktadır. • Arayüzünün sade ve kolay anlaşılır olması ergonomi açısından büyük avantaj sağlamaktadır.
5	DHT11 (Sıcaklık ve Nem Sensörü)	Ev maketindeki deneysel odanın sıcaklığının istenilen derecede olup olmadığının tespit edilmesini ve nem miktarının ölçülmesini sağlamaktadır.	<ul style="list-style-type: none"> • Deneysel odanın sürekli olarak sıcaklığını ve nemini ölçerek verinin ESP8266 vasıtasıyla kablosuz olarak kontrolöre iletmektedir. • Kontrolör sıcaklık değerinin kullanıcının istediği seviyede gerçekleşip gerçekleşmediğini denetlemektedir. Eğer; <ul style="list-style-type: none"> ▪ Sıcaklık, istenilen değer altındaysa istenilen sıcaklık değerine ulaşmaya kadar TEC1- 12706 Peltierin devreye girerek çalışmasını sağlar. ▪ Sıcaklık, istenilen değer üstündeyse Peltier devreden çıkar ve sıcaklık değerinin istenilen değere ulaşmaya kadar fanın çalışmasını sağlar. • Kullanıcının isteğine göre sistemin hızlı bir şekilde tepki vermesini sağlamaktadır.
6	MQ-6 Gaz Sensörü (LPG, Propan ve İzobütan)	Ev maketindeki deneysel odada herhangi bir doğalgaz kaçağının olup olmadığının düzenli olarak tespit edilmesini sağlamaktadır.	<ul style="list-style-type: none"> • Deneysel odada sürekli olarak gaz varlığını 300-10000 ppm aralığında ölçerek verinin ESP8266 vasıtasıyla kablosuz ve düzenli olarak kontrolöre

			<p>iletilmesini sağlamaktadır.</p> <ul style="list-style-type: none"> • Kontrolör gaz değerinin programcının belirlediği sınır değerinin üzerinde olup olmadığını denetlemektedir. Eğer; <ul style="list-style-type: none"> ▪ Sınır değerinin üstündeyse buzzer'a sinyal gönderir. Buzzer kullanıcının herhangi bir zarar görmeden haber almasını sağlar. ▪ Kullanıcının can güvenliği için sistemin hızlı bir şekilde tepki vermesini sağlar.
7	MQ-135 Hava Kalite Sensörü (Benzin, Alkol ve Duman)	Ev maketindeki deneysel odada benzin, alkol ve yangın-duman kaçağının olup olmadığını tespit edilmesini sağlamaktadır.	<ul style="list-style-type: none"> • Deneysel odada sürekli olarak CO2, duman varlığını 10-1000 ppm aralığında ölçerek verinin ESP8266 vasıtasıyla kablosuz ve düzenli olarak kontrolöre iletilmesini sağlamaktadır. • Kontrolör gaz değerinin programcının belirlediği sınır değerinin üzerinde olup olmadığını denetlemektedir. Eğer; <ul style="list-style-type: none"> ▪ Sınır değerinin üstündeyse buzzer'a sinyal gönderir. Buzzer kullanıcının herhangi bir zarar görmeden haber almasını sağlar. • Kullanıcının can güvenliği için sistemin hızlı bir şekilde tepki vermesini sağlamaktadır.

8	12V DC Fan (80x80x25mm)	Ev maketindeki deneysel odanın sıcaklığının istenilen derecede tutulabilmesini sağlamaktadır.	<ul style="list-style-type: none"> • Deneysel odanın soğumasını ve sıcak havanın dağılmasını sağlar. • Deneysel odanın sürekli olarak sıcaklığını ölçen DHT11, verinin ESP8266 vasıtasıyla kablosuz olarak kontrolöre iletilmesini sağlamaktadır. • Kontrolör sıcaklık değerinin kullanıcının istediği seviyede olup olmadığını düzenli olarak denetlemektedir. Eğer; <ul style="list-style-type: none"> ▪ Sıcaklık, istenilen değerin üstündeyse fana sinyal göndererek sıcaklığın istenilen değere ulaşana kadar çalışmasını sağlar. ▪ Sıcaklık, istenilen değerin altındaysa Peltierin çalışmasını sağlar. Sıcaklığın istenilen değere ulaşana kadar çalışmasını sağlar. • Kullanıcının isteğine göre sistemin hızlı bir şekilde tepki vermesini sağlamaktadır.
9	YL44 Buzzer (33 x 13 x 12 mm)	Yangın, gaz kaçağı ve hırsızlık durumunda can ve mal güvenliği için kullanıcıyı uyarmaktadır.	<ul style="list-style-type: none"> • Deneysel odada sürekli olarak gaz varlığını tespit eden MQ-6 ve MQ-135 veya herhangi bir hareket olup olmadığını tespit eden HC-SR501, verinin ESP8266 vasıtasıyla kablosuz olarak (HC-SR501 kablolu) kontrolöre iletilmesini sağlamaktadır. • Kontrolör gaz değerinin programcının belirlediği sınır değerinin üzerinde olup

			<p>olmadığını veya herhangi bir hareket olup olmadığını denetlemektedir. Eğer;</p> <ul style="list-style-type: none"> ▪ Sınır değerinin üstündeysen buzzer'a sinyal gönderir. Buzzer sesli ikazla uyararak kullanıcının herhangi bir zarar görmeden haber almasını sağlar. <ul style="list-style-type: none"> • Kullanıcı ev ortamında değil ise ve HC-SR501 tarafından bir hareket algılanırsa buzzer'a sinyal göndermektedir. • Kullanıcının can ve mal güvenliği için sistemin hızlı bir şekilde tepki vermesini sağlamaktadır.
10	LDR Foto Diyot (Optik Sensör)	Bahçe aydınlatmasının devreye girmesi için akşam karanlığından faydalanması sağlanmaktadır.	<ul style="list-style-type: none"> • Gün ışığına LDR tepki göstererek direnç değerinde değişmeye neden olur. • Direnç değerindeki değişimle ev maketinin bahçesinin aydınlanması için Ledler devreye sokulmaktadır. • Kullanım süresi açısından elektrik sarfiyatında tasarruf sağlamaktadır.
11	Toprak Nem Sensörü (Nem Ölçer Problar)	Bahçe sulamasının otomatik olarak yapılabilmesi için toprağın nem oranını kontrol edilmesini sağlamaktadır.	<ul style="list-style-type: none"> • Sensördeki nem ölçer problar ölçüm yapılacak ortama batırılmaktadır. • Toprağa daldırılan probların uçlarında, suyun neden olduğu dirençten bir gerilim indüklemesi meydana gelir. Bu indüklemenin büyüklüğüne göre de nem değeri ölçülebilmektedir.

			<ul style="list-style-type: none"> • Nem miktarının, programcı tarafından belirlenen sınır değerinin altına düşmesi sonucu kontrolöre veri iletilmektedir. • İletilen veri sonucu kontrolör, sulamayı başlatması için su pompasının çalışmasını sağlamaktadır. • Nem miktarı istenilen değere yükselene kadar sulama işlemi devam etmektedir. • Sulama otomasyonunun gerçekleşmesini ve zaman-yakıt açısından verimli ve düzenli bir şekilde çalışmasını sağlamaktadır.
12	HC-SR501PIR (Hareket Algılama Sensörü)	Eve hırsız girme veya beklenmedik bir durumda can ve mal güvenliği açısından kullanıcıyı uyarılmaktadır.	<ul style="list-style-type: none"> • Sensör sürekli olarak deneysel odada herhangi bir hareket olup olmadığını düzenli olarak denetlemektedir.(Kişi, Hayvan, Hırsız, vs.) • Kullanıcı eğer ev ortamında değilse ve HC-SR501 tarafından bir hareket algılanırsa kontrolöre veri iletilmektedir. • Kontrolörde kullanıcıyı uyarmak için buzzer'a sinyal göndermektedir. Buzzer sesli ikazla kullanıcıyı uyararak haber almasını sağlamaktadır. • Kullanıcının can ve mal güvenliği için sistemin hızlı bir şekilde tepki vermesini sağlamaktadır.
13	DC40-1245 Hidrofor (12V DC Su Pompası)	Otomatik bahçe sulamasının kullanıcınnın isteğine göre yapılmasını sağlamaktadır.	<ul style="list-style-type: none"> • Toprağa daldırılan problemlerin uçlarında, suyun neden olduğu dirençten bir gerilim indüklemesi meydana

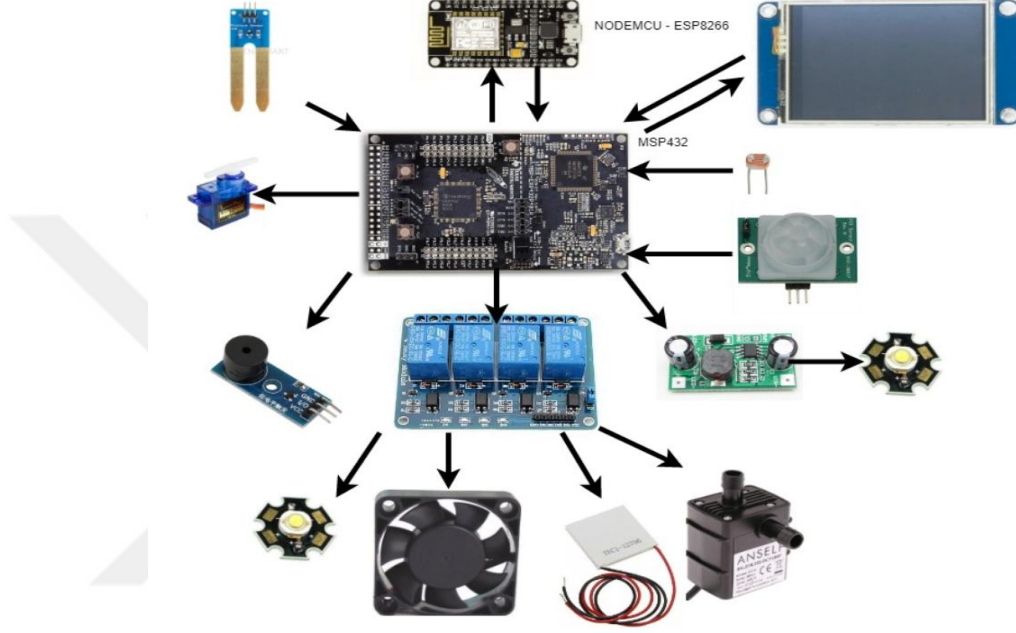
			<p>gelmektedir. Bu indüklemenin büyüklüğüne göre nem değeri ölçülmektedir.</p> <ul style="list-style-type: none"> • Nem miktarının, programcı tarafından belirlenen sınır değerinin altına düşmesi sonucu kontrolöre veri iletilmektedir. • İletilen veri sonucu kontrolör, sulamayı başlatması için DC40-1245 su pompasının çalışmasını sağlamaktadır. • Nem miktarı istenilen değere yükselene kadar su pompası çalışmaktadır. • Sulama otomasyonunun gerçekleşmesini ve zaman-yakıt açısından verimli ve düzenli bir şekilde çalışmasını sağlamaktadır.
14	TEC1-12706 Peltier (Termoelektrik Sistem)	Ev maketindeki deneysel odanın sıcaklığının istenilen derecede tutulabilmesini sağlamaktadır.	<ul style="list-style-type: none"> • Deneysel odanın sürekli olarak sıcaklığını ölçen DHT11, verinin ESP8266 vasıtasıyla kablosuz olarak kontrolöre iletilmesini sağlamaktadır. • Kontrolör sıcaklık değerinin kullanıcının istediği seviyede olup olmadığını düzenli olarak denetlemektedir. Eğer; <ul style="list-style-type: none"> ▪ Sıcaklık, istenilen değerin altındaysa TEC1- 12706 Peltier devreye sokulur. Sıcaklığın istenilen değere ulaşana kadar çalışmasını sağlar. • Kullanıcının isteğine göre sistemin hızlı bir şekilde tepki vermesini sağlamaktadır.

15	1 Watt Power Led ve PWM Led Sürücü Modülü (Aydınlatma Sistemi)	Ev maketindeki deneysel odanın ve bahçenin aydınlatması yapılmaktadır.	<ul style="list-style-type: none"> • Karanlık olduğunda gün ışığı azaldığından LDR'nin direnç değerinde değişmeye neden olmaktadır. • Programcı tarafından belirlenen direnç değerinin altına düştüğünde kontrolöre veri iletilmektedir. • Kontrolörde sinyal geldikten sonra bahçe aydınlatmasını sağlayan ledler devreye girmektedir. • Deneme odasındaki aydınlatma sistemi kullanıcının isteğiyle aktif hale getirilmektedir. • PWM led sürücü modülü sayesinde aydınlatma şiddeti seviyesi ayarlanarak deneme odasında kullanıcın istediği derecede aydınlanma sağlanmaktadır. • Kullanım süresi ve aydınlatma şiddeti açısından elektrik sarfiyatından tasarruf sağlanmaktadır.
16	12V 30A Güç Kaynağı	Ev maketindeki elektronik cihazlar için gerekli olan bütün akım ihtiyacını karşılamaktadır.	<ul style="list-style-type: none"> • Kontrol edilen parametre sayısının fazla olmasından dolayı 30 Amperlik güç kaynağı tercih edilmiştir. • 220 V AC giriş gerilimini, istenirse 12 V DC (30 A) istenirse de 24 V DC (15 A) çıkış gerilimine çevirmektedir. • Yüksek verime sahip olmasından dolayı elektrik sarfiyatından tasarruf sağlamaktadır.

3.1.1. Kullanılan Elektronik Elemanlar

Bu çalışmada akıllı ev uygulaması için kontrol edilecek parametrelere karar verilerek her bir parametre için piyasadaki en uygun sensör araştırılmış ve istenilen özellikleri karşılayabilecek en küçük boyutlu olanlar seçilmiştir. Böylece yapılan ev maketinin boyutları minimum düzeyde tutulmuştur.

Şekil 3.1’de mikrokontrolör ve her bir parametrenin kontrol edilmesini sağlayan elektronik malzemeler (sensörler ve yükler) arasındaki veri alış-verişi gösterilmiştir.



Şekil 3.1. Projede kullanılan elektronik malzemeler arasındaki veri alış-verişi

3.1.1.1. MSP432 Mikrokontrolör

3.1.1.1.1. Genel Bakış

Texas Instruments tarafından piyasaya sürülen SimpleLink MSP EXP432P401R LaunchPad geliştirme kiti, SimpleLink MSP432P401R mikro denetleyicisi için kullanımı kolay bir değerlendirme modülüdür. SimpleLink MSP432 düşük güç + performans Arm 32-bit Cortex-M4F mikrokontrolör (MCU) üzerinde geliştirmeye başlamak için programlama, hata ayıklama ve enerji ölçümleri için bütünlük hata ayıklama probu dahil gereken her şeyi içermektedir.

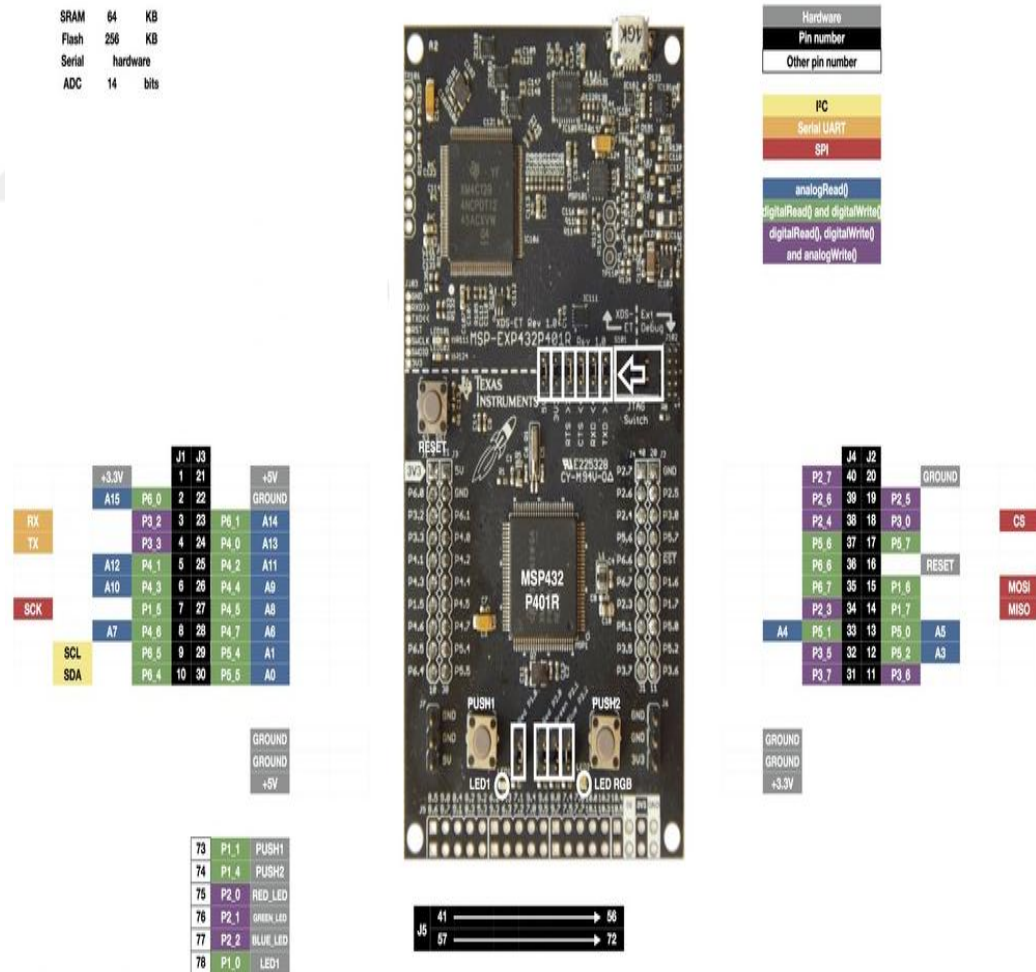
SimpleLink MSP432P401x mikrokontrolörler (MCU'lar), tümleşik 16-bit hassasiyetli ADC ile optimize edilmiş kablosuz ana bilgisayar MCU'larıdır ve 80 μ A / MHz aktif güç ve 660 nA bekleme gücünde FPU ve DSP uzatmalarıyla ultra düşük

güç performansı sunmaktadır. Optimize edilmiş kablosuz bir ana bilgisayar MCU'su olan MSP432P401x, geliştiricilerin SimpleLink kablosuz bağlantı çözümlerini temel alan uygulamalara yüksek hassasiyetli analog ve bellek uzantısı eklemelerine izin verir. Tek çekirdekli yazılım geliştirme seti (SDK) ve zengin araç seti ile kullanımı kolay geliştirme ortamı sunar. SimpleLink platformunun nihai amacı, tasarım gereksinimleri değiştiğinde yüzde 100 kod kullanımının tekrar edilebilmesidir[11].

Çalışmada kullanılan mikrokontrolörün, kontrol edilmesi istenen çevre birimleri ile bağlantıları Eagle programında çizilmiş ve Ek A'da gösterilmiştir.

Programcının yazdığı algoritmaya göre mikrokontrolörün çevresel birimlere çıkış vermesini sağlayan röle kartının Eagle programında çizilen devre şeması Ek B'de gösterilmiştir.

MSP432P401x mikrokontrolörün Eagle programında çizilen pin diyagramı Ek C'de gösterilmiştir.



Şekil 3.2.MSP432P401x mikrokontrolör kiti

3.1.1.2. NodeMCU ESP8266

3.1.1.2.1. Genel Bakış

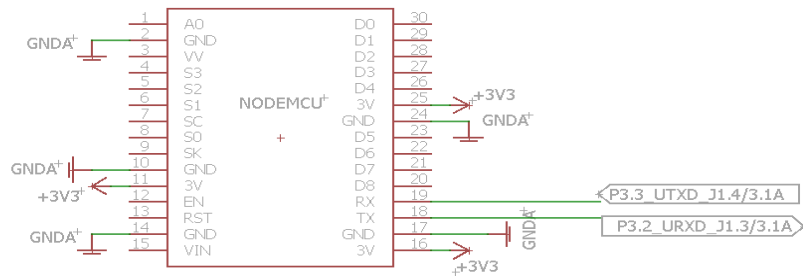
ESP8266, Espressif Systems tarafından tasarlanan mikrokontrolördür. ESP8266'nın kendisi, mevcut mikrokontrolörden WiFi bağlantısına köprü olarak sunulan ve kendi kendine yeten uygulamaları çalıştırabilen, bağımsız bir WiFi ağ çözümüdür. Mikro USB kablosuyla, NodeMCU geliştirme kiti bilgisayara herhangi bir sorun olmadan, aynı Arduino gibi bağlanabilmektedir. Aynı zamanda breadboard uygulamalarında tam olarak uyumludur.

NodeMCU açık kaynaklı IoT platformudur. Espressif Systems'den ESP8266 Wifi SoC üzerinde çalışan ürün yazılımı ve ESP-12 modülüne dayalı donanım içermektedir. “NodeMCU” terimi, geliştirme kiti yerine, yazılım destekli donanımı ifade eder. Yazılım destekli donanım Lua kodlama dilini kullanmaktadır. ELua projesine dayanmakta ve ESP8266 için Espressif Non-OS SDK'yı temel almaktadır. Lua-cjson ve spiff gibi birçok açık kaynaklı proje kullanmaktadır.

Sürüm özellikleri, DevKit v1.0 Breadboard dostu hafif ve küçük boyutludur. 3.3V çalışmakta ve aynı zamanda USB gücü ile de kullanılabilir. 802.11b/g/n kablosuz protokolü kullanır. ESP-12E yongası üzerindeki PCB anteni gömülüdür. PWM, I2C, SPI, UART özelliklerini destekler ve 1 kablolu, 1 analog pine sahiptir. CP2102 USB Seri İletişim arayüz modülünü kullanır. Arduino IDE ile uyumludur. (Uzatma kartı yöneticisi gerekli). Hem Lua'yı (node.js gibi) hem de Arduino C programlama dilini desteklemektedir[12].

3.1.1.2.2. Pin Diyagramı

MSP432 - NODEMCU COM.



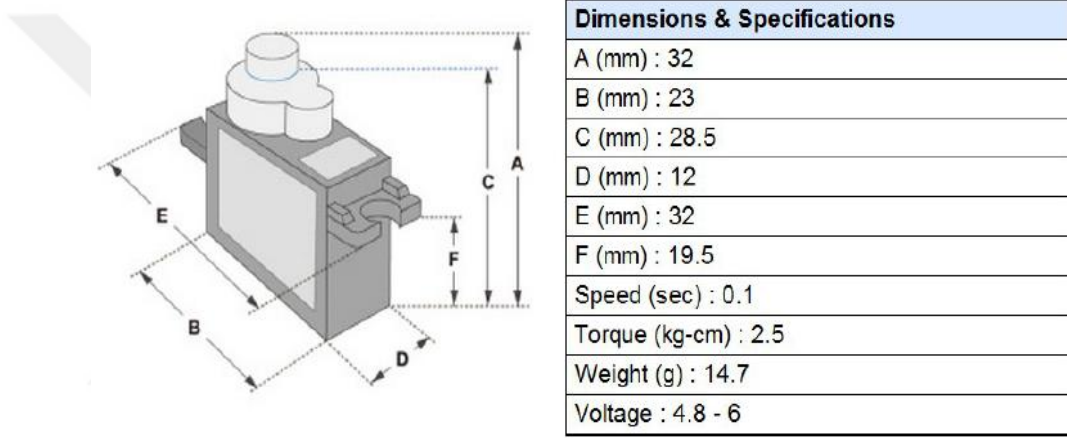
Şekil 3.3. NodeMCU ESP8266 pin diyagramı

3.1.1.3. Servo Motor

3.1.1.3. 1. Genel Bakış

Yüksek çıkış gücüne sahip küçük ve hafiftir. Servo yaklaşık 180 derece dönebilir (her yöne 90°) ve standart tiplerde olduğu gibi çalışmakta ancak onlara göre daha küçüktür. Bu servoyu kontrol etmek için herhangi bir servo kodu, donanımı veya kitaplığı kullanabilmektedir. Geri besleme ve dişli kutusu olan bir motor olmadığı için, özellikle boyutlarının küçük olmasından dolayı büyük bir avantaja sahiptir. 3 tane kol ve donanım ile birlikte kullanılmak istenen sisteme entegre edilebilmektedir[13].

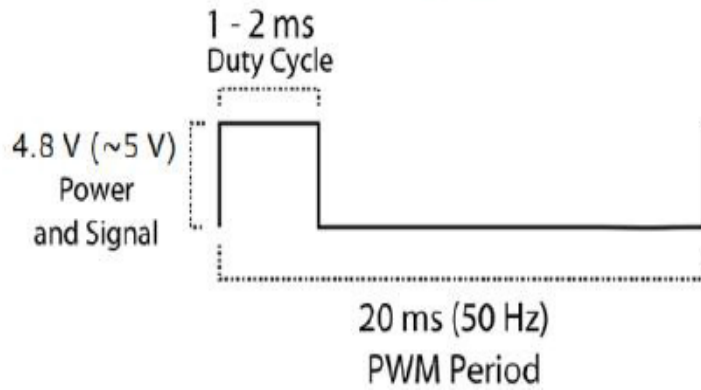
3.1.1.3. 2. Servo Motorun Boyutları



Şekil 3.4.Servo motor boyutları

3.1.1.3. 3. Servo Motorun Çalışma Prensibi

"0"pozisyonu (1,5 ms darbe) ortada, "90"pozisyonu (~ 2ms darbe) tamamen sağa, "-90"pozisyonu (~ 1ms darbe) tamamen sola döner[13].

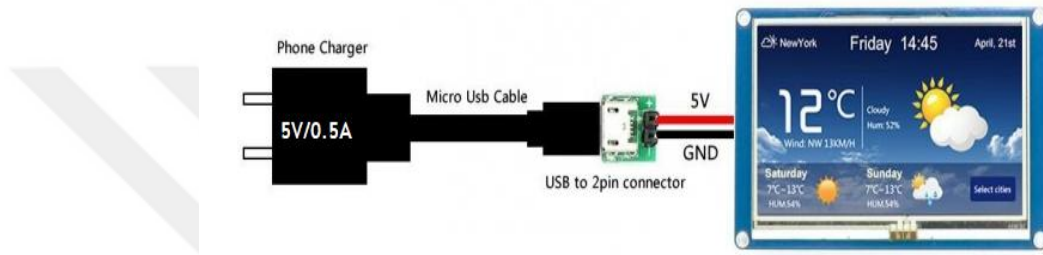


Şekil 3.5.Servo motor çalışma sinyali

3.1.1.4. Nextion 3.2” Dokunmatik Ekran

3.1.1.4. 1. Genel Bakış

Nextion, bir insan ile bir proses, makine, uygulama veya cihaz arasında kontrol ve görselleştirme arayüzü sağlayan kesintisiz bir İnsan Makine Arabirimi (HMI) çözümüdür. Nextion temel olarak IoT veya tüketici elektroniği alanında kullanılmaktadır. Geleneksel LCD ve LED Nixie tüpünü değiştirmek için en iyi çözümdür. Nextion Editor yazılımıyla, kullanıcılar Nextion ekranı için kendi arayüzlerini yaratabilmekte ve tasarlayabilmektedir[14].

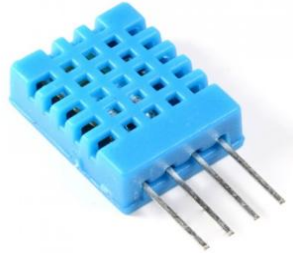


Şekil 3.6.Nextion 3,2” dokunmatik ekran bağlantı şeması

3.1.1.5. DHT11 Sıcaklık ve Nem Sensörü

3.1.1.5. 1. Genel Bakış

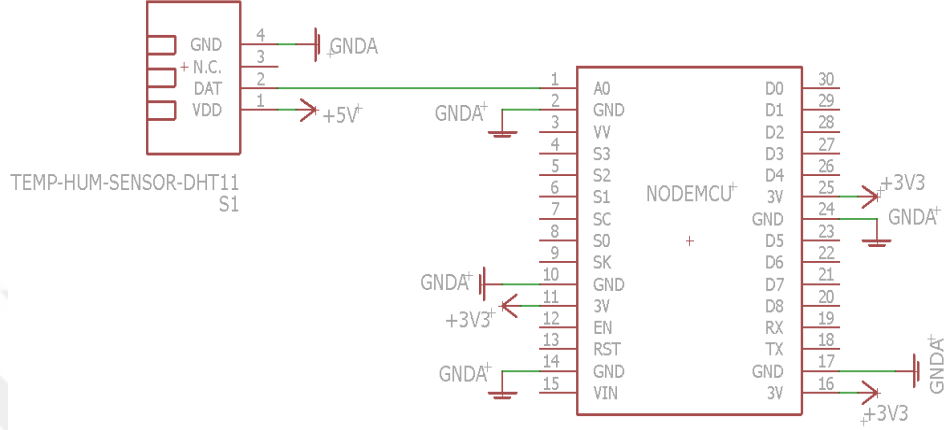
DHT11 dijital sıcaklık ve nem sensörü kompozit (sıcaklık ve nem kalibre edilmiş dijital sinyal çıkışı içeren) bir sensördür. Ürünün yüksek güvenilirlik ve mükemmel uzun süreli stabiliteye sahip olmasını sağlamak için özel dijital modüller toplama teknolojisi; sıcaklık ve nem algılama teknolojisini uygulamaktadır. Sensör, dirençli bir ıslak bileşen algısı ve bir NTC sıcaklık ölçüm cihazı içermekte ve yüksek performanslı 8-bit mikrokontrolör ile bağlanabilmektedir.



Şekil 3.7.DHT11 görseli

3.1.1.5.2. Pin Diyagram

- 1.VDD: güç kaynağı 3,5~ 5,5V DC
- 2.DATA: seri verileri, tek bir veri yolu
3. NC: boş pin
- 4.GND: toprak, negatif güç



Şekil 3.8.DHT11 pin diyagram ve devre şeması

Mikroişlemci ve DHT11 bağlantıları, Şekil 3.8'de gösterildiği gibi tipik uygulama devresidir. Veri, mikroişlemci I / O portlarını çekerek bağlanmaktadır.

3.1.1.6. MQ-6 Gaz Sensörü

3.1.1.6.1. Genel Bakış

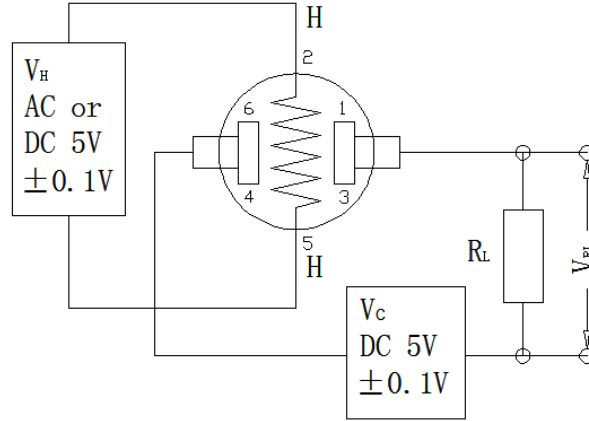
MQ6 LPG Gaz Sensörü havadaki LPG, izobütan ve propan varlığını saptayıp, konsantrasyonunu ölçerek analog voltaj çıkışı olarak veren bir sensör modülüdür. Gaz konsantrasyonu ölçme menzili 300 ppm ile 10000 ppm arasındadır. MQ6 LPG Gaz Sensörü, -100 C ve 500 C arasında çalışabilir ve 5 V besleme ile 150 mA'den daha az akım çeker[16].

Sensörün ısıtma (H) pinlerine 5V bağlamak, sıcaklığın çalışabileceği seviyede tutmak için yeterlidir. A ya da B pinlerine 5V bağlamak sensörün diğer pinlerine analog voltaj sağlamasına neden olur. Çıkış pinleriyle toprak pinleri arasındaki rezistif bir yük ekleyerek sensörün hassasiyetini ayarlamak mümkündür. Rezistif yükü kendi uygulamamız için datasheet içindeki denklemlere göre kalibre etmemiz gerekmektedir, ancak 20 kΩ başlangıç için uygun bir rezistif yüküdür[16].



Şekil 3.9.MQ-6 görseli

Şekil 3.10'daki devre MQ-6'nın temel test devresidir. Sensör iki voltaj girişi gerektirir: ısıtıcı voltajı (V_H) ve devre voltajı (V_C). Sensöre standart çalışma sıcaklığı sağlamak için kullanılan V_His, DC veya AC gücünü benimseyebilmekte, V_{RL} ise R_L'nin yük direncinin voltajını sensör ile seri halde tutmaktadır. V_c, direnç direnci R_L'ye yük tespit voltajını sağlamakta ve DC gücüne sahip olması gerekmektedir.



Şekil 3.10.MQ-6 devresi

3.1.1.6.2. Hassasiyet Ayarı

MQ-6'nın direnç değeri çeşitli türlerde ve çeşitli konsantrasyon gazlarında farklılık göstermektedir. Yani, bu bileşenleri kullanırken, hassasiyet ayarı çok gereklidir. Dedektörü havada 1000 ppm LPG konsantrasyonu için kalibre edilmesi ve yaklaşık 20 K Ω (10 K Ω ile 47 K Ω) arasında yük direnci (R_L) değeri kullanılması gerekmektedir. Doğru ölçüm yaparken, sıcaklık ve nem etkisi göz önünde bulundurularak, gaz dedektörü için uygun alarm noktasının belirlenmesi gerekmektedir.

3.1.1.7. MQ-135 Hava Kalitesi Kontrol Sensörü

3.1.1.7.1. Genel Bakış

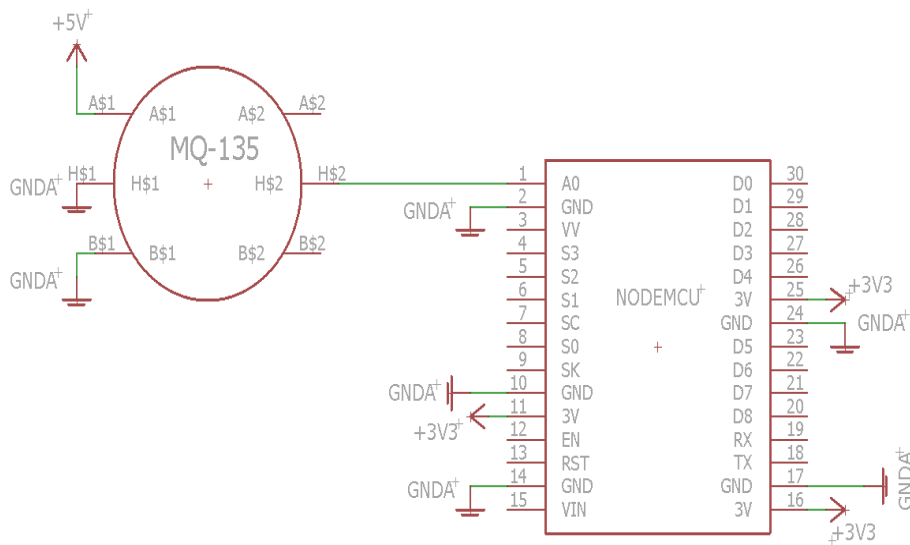
MQ135 Hava Kalitesi Kontrol Sensörü havadaki benzin, alkol ve dumani ölçerek hava kalitesini kontrol eden bir sensördür. Sensörün ısıtıcısı 5V ile çalışmaktadır. Hava kalitesini kontrol eden sistemlerde kullanılan ekipmanlarda sıklıkla kullanılmaktadır[17].



Şekil 3.11.MQ-135 görseli

3.1.1.7.2. Hassasiyet Ayarı

MQ-135'in direnç değeri çeşitli türlerde ve çeşitli konsantrasyon gazlarında farklılık göstermektedir. Dolayısıyla bu bileşenleri kullanırken, hassasiyet ayarı çok gereklidir. Dedektörü havadaki 100 ppm NH₃ veya 50 ppm alkol konsantrasyonu için kalibre edilmesi ve (RRL) yaklaşık 20 K Ω (10 K Ω ila 47 K Ω) olan yük direnci değerini kullanılması gerekmektedir. Doğru ölçüm yaparken, sıcaklık ve nem etkisi göz önünde bulundurularak, gaz dedektörü için uygun alarm noktası belirlenmesi gerekmektedir.



Şekil 3.12.MQ-135 devresi

3.1.1.8. 12V DC Fan

3.1.1.8.1. Genel Bakış

DC Fan, ev ortamını soğutmak için kullanılmış bir ekipmandır. Kullanıcının istediği sıcaklığa göre devreye girip çıkmaktadır. Bu DC fan, uzun servis ömrü için çift hassas bilyalı rulmanlar ile donatılmıştır. İsteğe bağlı montaj konumu yapılabilmektedir. Engellemeye ve ters kutuplamaya karşı korumalı, otomatik yeniden başlatmalıdır. Çalışma sıcaklığı aralığı -10°C ile $+70^{\circ}\text{C}$ 'dir.

Takometre sinyali dönme başına iki darbeye sahiptir. Kilitli rotor sinyali ve otomatik yeniden başlatma özellikleri vardır.



Şekil 3.13.Fan görseli

3.1.1.9. YL-44 Buzzer

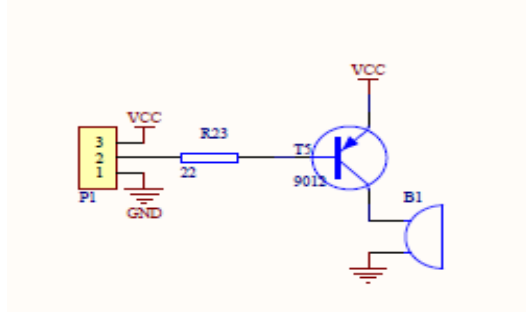
3.1.1.9.1. Genel Bakış

Bu cihaz, sesli 2 kHz frekans aralığında çalışan küçük bir zil modülüdür. Aktif bir zildir, yani harici bir frekans jeneratörüne ihtiyaç duymadan kendi kendine ses çıkarmaktadır. Özel bir PWM kanalına ihtiyaç duymadan Arduino gibi mikrokontrolör kartlarla kolayca kullanılabilir[19].



Şekil 3.14.Buzzer görseli

3.1.1.9.2. Devre Şeması



Şekil 3.15.YL-44 buzzer devresi

3.1.1.10. Foto Diyot(LDR)

3.1.1.10. 1. Genel Bakış

Optik sensör türleri içerisinde akla ilk gelen elektronik elemandır. LDR, ortamdaki ışığın şiddetine göre üzerine düşen direnç değerini ters orantılı olarak ayarlayabilen en basit optik sensör çeşididir[20].

LDR, ortamdaki ışığın şiddetine göre direnç değerini değiştirmesi en önemli özelliğidir. Böylece, ortam ışığını ölçmeye ve ortamda belli bir seviyenin üzerinde ışık olup olmadığını tespit etmeye yarayan bir sensör olarak kullanılabilir. Ortamın ışık şiddeti arttığında LDR'nin direnci düşmekte ve ortam karanlıklaştıkça ise direnci artmaktadır. Ev maketinde bahçe ışıklarının otomatik olarak karanlık ortamda yanması için kullanılmıştır.

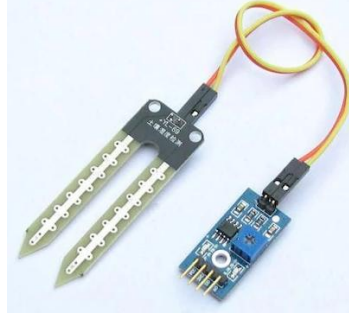


Şekil 3.16.LDR görseli

3.1.1.11. Toprak Nem Sensörü

3.1.1.11. 1. Genel Bakış

Toprak nem sensörü, toprağın içerisindeki nem miktarını veya ufak ölçekte bir sıvının seviyesini ölçmek için kullanabilecek bir sensördür.



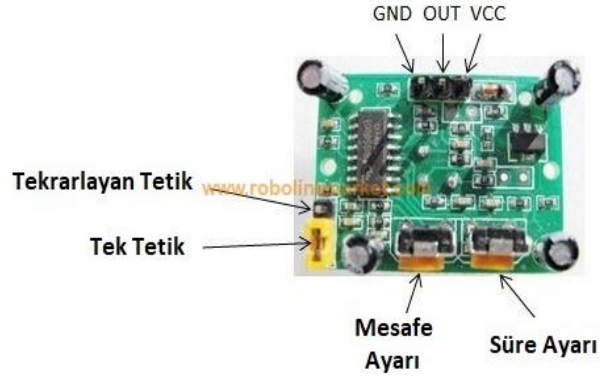
Şekil 3.17. Toprak nem sensörü görseli

Nem ölçer problar ölçüm yapılacak ortama batırılarak kullanılmaktadır. Toprağın veya içine batırılan sıvının meydana getirdiği dirençten dolayı, prob uçları arasında bir gerilim farkı oluşmaktadır. Bu gerilim farkının büyüklüğüne göre de nem miktarı ölçülebilmektedir. Topraktaki nem oranı arttıkça iletkenliği de artmaktadır. Kart üzerinde yer alan trimpot sayesinde hassasiyet ayarı yapılabilmektedir. Arduino veya farklı mikrokontrolörler ile rahatlıkla kullanılabilir[21].

3.1.1.12. HC-SR501 Hareket Algılama Sensörü

3.1.1.12. 1. Genel Bakış

Cisim hareketi algılamak için kullanılan bir modüldür. İçerisinde (Infrared, IR) sensörü vardır. Dijital çıkış veren bir sensördür. Ayrıca bu sensörün güç tüketimi çok düşük olduğu için sadece bu sensörü besleyerek mikrokontrolcüyü uyku moduna almak için de kullanılmaktadır.



Şekil 3.18.Hareket algılama sensörü görseli

HC-SR501, bir nesnenin, bir hayvanın veya bir kişinin hareketini algılamak için pasif kızılötesi ışık algılaması kullanan hareket sensörüdür. HC-SR501 yalnızca 1 veya 0 sonuçları göndermekte, bu nedenle güvenlik alarmları ve otomatik aydınlatma sistemlerinde ağırlıklı olarak kullanılmaktadır[22].

3.1.1.13. DC40-1245 12V DC Su Pompası

3.1.1.13. 1. Genel Bakış

Küçük boyutlara sahip su pompası, bilgisayar soğutma sistemi, bahçe sulaması gibi birçok alanda kullanılmaktadır. 11.5 watt gücünde olup 950 mA akım değerine sahip ve 4,5 metreye kadar saatte 480 litreye kadar su basabilmektedir[23].

Yüksek performanslı seramik şaft yapısına sahip olduğundan 30.000 saatten fazla uzun çalışma ömrü vardır. Tamamen su geçirmez tasarımı sayesinde suyla teması herhangi bir sorun teşkil etmemektedir. 35 db'den az düşük gürültüyle 60 ° C' ye kadar 12 V DC gerilimle çalışabilmektedir.



Şekil 3.19.Su pompası görseli

3.1.1.14. TEC1-12706 Peltier Termo Elektrik sistem

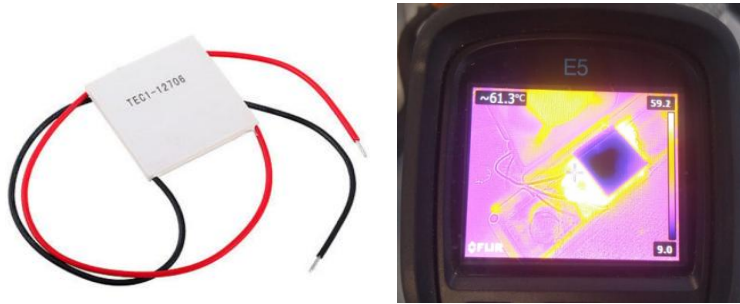
3.1.1.14. 1. Genel Bakış

Peltier, bir termo elektrik sistem elemanıdır. Elektrik enerjisi ile ısı enerjisinin birbirleri arasındaki dönüşümünü sağlamaktadır. Bir termo elektrik modül N ve P tipi yarı iletkenlerden oluşan termo element malzemelerden oluşmaktadır. Bu termo element sistem elektrikselsel olarak seri, termal olarak paralel bağlanarak değişik amaçlar için değişik kapasitelerde termo elektrik modüller elde edilebilmektedir.



Şekil 3.20. Peltier çalışma mantığı

İki ucuna doğru akım verildiğinde N ve P tipindeki elemanlar elektronları bir uçtan diğerine doğru itmekte ve bir yüzde ısınma diğer yüzde soğutma meydana getirmektedir. Böylece termo elektrik modül bir ısı pompası gibi çalışır. Ayrıca termo elektrik modülün iki yüzeyi arasında sıcaklık farkı oluşturulursa termo elektrik modül bir DC akım kaynağı gibi davranarak elektrik üretir. Peltier ise yapısı gereği gerilim uygulandığında bir yüzeyi ısınan diğer yüzeyi ise soğuyan bir elemandır[24].



Şekil 3.21. Peltier görseli ve termal kamera ile görüntülenmesi

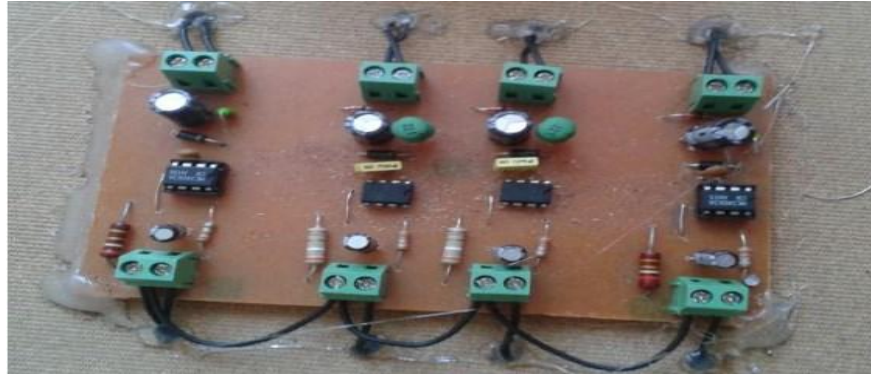
3.1.1.15. Power LED ve PWM Girişli LED Sürücü Modülü

3.1.1.15. 1. Genel Bakış

Yüksek güce sahip (0.5W ve üstü) LED'lere power LED ismi verilmektedir. Bu tip LED'ler elektronik devrelerdeki güç ve durum göstergesi gibi işlerden çok, aydınlatma amaçlı uygulamalarda tercih edilmektedir. Aydınlatmada kullanıldıkları için ışığı odaklamak çoğu için lense sahiptir. Ayrıca bu tip LED'ler yüksek güç tükettikleri için ısınmakta ve bu sebepten dolayı çoğu power LED kendisine ait bir soğutma çözümüne sahiptir[25].

Ev içi ve bahçe aydınlatmasında power LED'ler kullanılmıştır. Ancak power LED'ler aşırı akım çektiği için ısınırlar ve bir süre sonra arızalanmaktadırlar. Bu yüzden Şekil 3.22'de gösterilen Power Led Sürücü Kartı tasarlanmıştır. Tasarlanan sürücü kartı ile ledleri sürebilmek için önemli parametre olan akım sabitlenmiştir. Devrenin girişindeki +12 V, direkt MC34063A DC/DC konverter entegresine gelmektedir. Burada en önemli dikkat edilmesi gereken şey çıkıştaki direnç değeridir. Referans voltajı 1,25 olduğundan çıkış akımı $1,25/3,9 = 0,320$ mA'ye sabitlenir. $I_{peak} = 2 * I_{out}$ değerinde maksimum 1,5A olması gerekmekte ve bu değeri geçmemelidir. 3 numaralı bacağa bağlı 470 pF kondansatör PWM sinyalinin $ton = 470 \text{ pF} / (4 * 10^{-5}) = 11,75$ us ve 14,1 us seviyesine ayarlamaktadır.

PWM girişli led sürücü modülünün çıkış gerilimi, istenilen değerde ayarlanabildiği için sürülen LED'in aydınlatma şiddeti kullanıcının isteğine göre ayarlanabilmektedir. Bu sayede elektrik sarfiyatında tasarruf ve kişiye özel ergonomide avantaj sağlamaktadır.



Şekil 3.22. Power led sürücü kartı

3.1.1.16. 30 A 12 V DC Güç Kaynağı

3.1.1.16. 1. Genel Bakış

Akıllı ev otomasyonu çalışmasında istenilen parametreleri kontrol etmek için kullanılan mikrokontrolör, alıcı-verici ve yük gibi elektronik malzemelerin düzgün çalışabilmesi için gerekli çıkış akımını sağlayan 30 A 12 V DC güç kaynağı tercih edilmiştir. Sistemde kullanılan elektronik malzemelerin sayısının fazla olması ihtiyaç duyulan akım amperajını yükseltmiştir. Özellikle Peltier (8A) gibi yüklerin çektikleri akım değerlerinin büyük olması güç kaynağının 30 Amper olarak seçilmesine etken olmuştur[26].

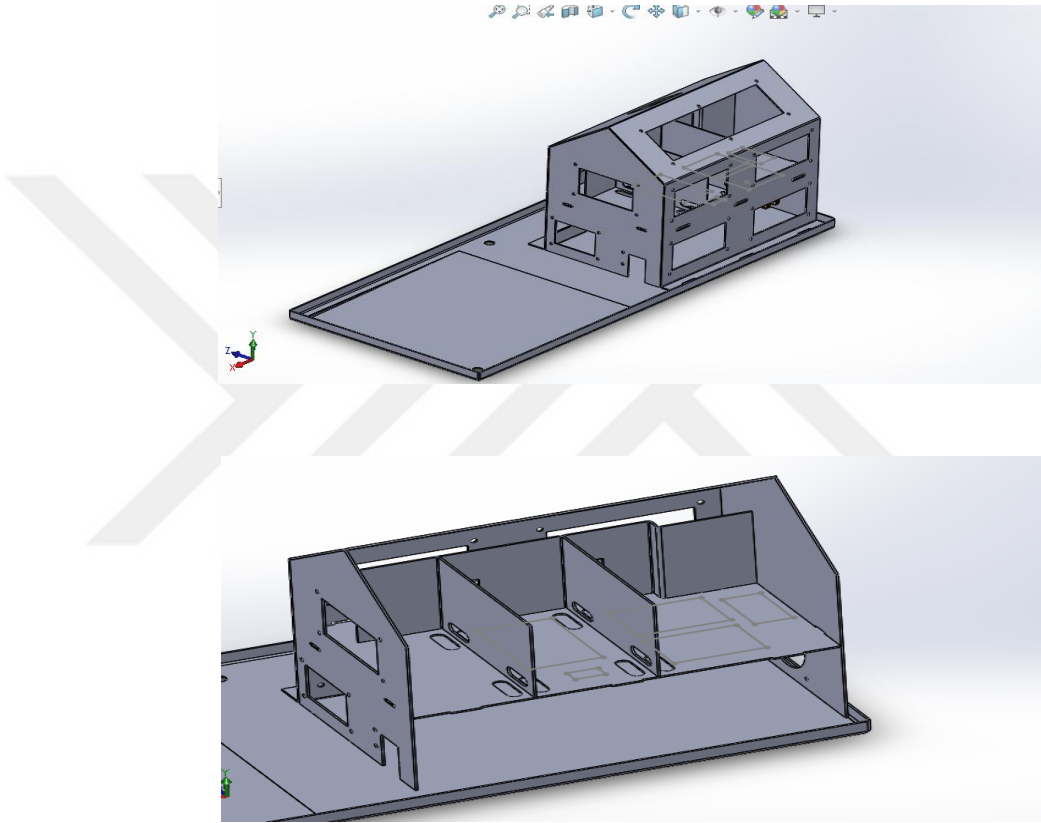


Şekil 3.23. Güç kaynağı görseli

3.1.2. Tasarlanan Ev Maketi

Bu çalışmanın esas işlerliğini ve amaçlanan hedefe ne kadar ulaşılabildiğini göstermek ve oluşturulmuş sistemi doğal ortamında gözlemlemek maksadıyla modern bir ev minyatürize edilerek ev maketi hazırlanmıştır. Ev maketi malzemesi, piyasada kolayca bulunabildiği, montajı rahat yapılabildiği ve CNC tezgâhlarında rahatça işlenebildiği için metal plakalardan imal edilmiştir. Kullanılacak olan ve piyasadan temin edilen en küçük boyuttaki elektronik malzemelere göre evin ölçüleri belirlenmiştir. Bu sayede maketin boyutlarını minimize tutarak hem ağırlıktan hem de ergonomi açısından avantaj sağlanmıştır.

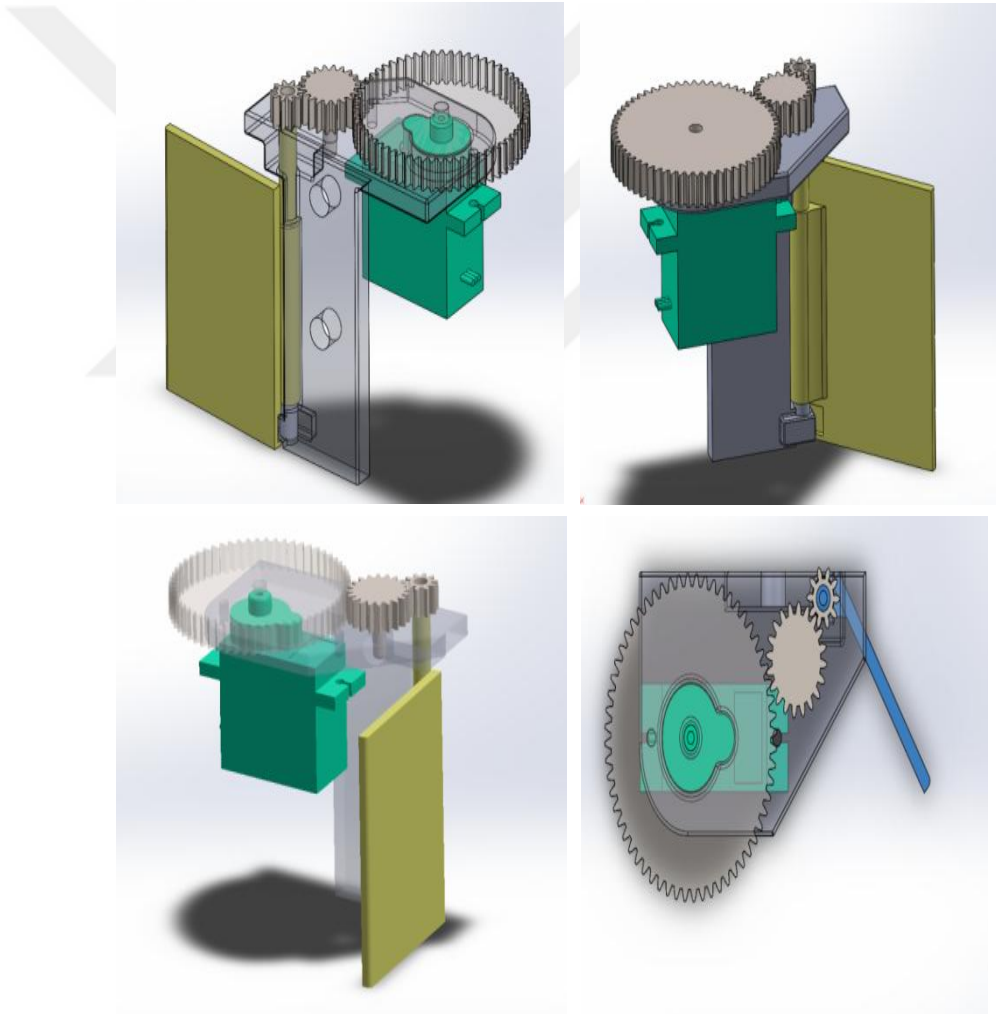
“Solid Works” çizim programında evin tasarım planları belirlenen ölçülere göre hazırlanmıştır. İki katlı olarak tasarlanan evin ilk katı, kontrol edilmesi planlanan parametrelerin ne derecede kontrol edilebildiğini gözlemlemek amacıyla deneme odası olarak tasarlanmıştır. İkinci kat ise 3 odadan oluşmakta olup entegre kartları ve mikrokontrolörü rahatça yerleştirilebilecek şekilde tasarlanmıştır. İki katlı ev maketi, bahçe platformu üzerine yerleştirilmiş ve burada modern bir bahçe sulaması ve aydınlatmasının yapılması planlanmıştır. Ev tasarım çizimleri Şekil 3.24’de gösterilmiştir.



Şekil 3.24.Solid Works programında tasarlanan ev maketinin çizimleri

Çizimleri yapılan ev maketi 9 parçadan oluşmakta olup birbirlerine kolayca montaj yapılacak şekilde tasarlanmıştır. CNC tezgâhında, yapılan çizimlere göre parçalar şekillenerek imal edilmiştir. Daha sonra parçalar estetik bir görünüm kazanması için sprey boya ile boyanmıştır. Ev ortamında sıcaklık ve soğukluk kontrolü yapıldığı için pencereler ısı yalıtımı için fleksi ile kapatılmıştır. “Kullanıcının anahtarı içerde unutması” senaryosuna uygun olarak internet erişimi olan akıllı telefon aracılığı ile kapının açılması sağlanmıştır. Bir diğer senaryoda “kullanıcının evden aceleyle kapıyı çekip çıkması ve kapının kapanmaması”

durumunda kapının otomatik olarak kapanması sağlanmıştır. Bu hareketli mekanizmanın sağlanabilmesi için dış kapı özel olarak tasarlanmıştır. Kapının açılması ve kapanması, istenilen derecede ve yönde hareket ettirilmesi servo motor ve dişli sistemi ile sağlanmıştır. Ayrıca kapıya bir adet mıknatıs ve kapı eşiğine bir adet reed switch monte edilmiştir. Reed switch mıknatısı algılayınca kontaklarını kapatmakta, mıknatıs uzaklaşınca kontağını açmaktadır. Böylece kapının açık mı kapalı mı bilgisi mikrokontrolöre iletilmektedir. Eğer kapı açık konumda 30 sn'den fazla kalırsa kapı kendiliğinden kapanmaktadır. Kapı mekanizmasına ait çizimler Şekil 3.25'de gösterilmiştir. Ev maketinin boyutları ve proje tasarımları "EKLER" kısmında ayrıntılı olarak gösterilmiştir.



Şekil 3.25.Solid Works programında tasarlanan ev maketindeki dış kapı mekanizması çizimleri

3.2. Yöntemler

3.2.1. Kullanılan Programlar

Akıllı ev otomasyonu çalışmasında uygulama ve tez yazım aşamalarında Energia, Arduino, Eagle, Nextion Editör ve Solid Works programları kullanılmıştır.

3.2.1.1. Energia Programı

Energia 2012 yılında Robert Wessels tarafından Wiring ve Arduino yapısını Texas Instruments'a ait MSP430 tabanlı geliştirme kartlarına uyarlamak için gerçekleştirilen bir açık kaynak kodlu geliştirme platformudur. Şu anda birçok TI geliştirme kartını destekleyen popüler bir gömülü sistem geliştirme ara yüzüdür.

Energia, Arduino IDE (Integrated Development Environment) temelli bir geliştirme ortamıdır. Arduino programlama ile Energia arabirimi ve kullanılan kod yazım dizimi ve değişken tanımlamaları birbirine oldukça benzemektedir. Energia ile MSP430 ve 432, CC3200 Launchpadler ile CC3100 Wi-Fi ve Educational BoosterPack ve BoosterPack2'leri programlanabilmektedir. Energia geliştirme ortamı bir kod yazım alanı, mesaj alanı, menüler, metin konsolu, ortak işlevler için düğmeler, araç çubuğu ve bir metin editöründen oluşur. Launchpadler ile iletişim kurar ve onlara program yükleme işini gerçekleştirir.

Energia ile birlikte, LaunchPad etkileşimli nesnelere geliştirmek, çeşitli anahtarlardan veya sensörlerden girdi almak ve çeşitli ışıkları, motorları ve diğer fiziksel çıkışları kontrol etmek mümkündür. Wi-Fi, NFC, Bluetooth, Zigbee ve hücresel gibi çeşitli RF türleri üzerinden iletişimi sağlamak için kablosuz modüller de eklenebilmektedir.

Bu çalışmada Energia programı, MSP432 mikrokontrolörün akıllı ev ortamını denetleyebilmesi ve sistemin planlanan tepkiyi verebilmesi için gerekli algoritmaların yazılıp derlendikten sonra mikrokontrolörün hafızasına yüklenmesini sağlamıştır. Mikrokontrolörün Node MCU ile Nextion 3,2" dokunmatik ekrana veri gönderim-alımı Energia programında yazılan kodlar ile mümkün kılınmıştır. Yazılan algoritmalar "EKLER" bölümünde gösterilmiştir.

3.2.1.1.1. Program Tanıtımı

Enerji programında ekrana gelen pencerenin sağ alt köşesinde kurulu Launchpad ve bağlı olduğu seri port görüntülenmektedir. Yazılan kodları, kontrol etme ve Launchpad'e yükleme ve seri port görüntüleme gibi özellikler aynen Arduino IDE'de olduğu gibi gerçekleştirilmektedir.



```
energia_lcd | Energia 0101F0017
File Edit Sketch Tools Help

energia_lcd
{
  lcd_sicaklik_guncelle(22);
  lcd_nem_guncelle(65);
  lcd_co2_guncelle(250);
}

void LCD_Veri_Alma(void)
{
  if (Serial1.available() > 0)
  {
    lcd_gelen_data[byteSayisi++] = Serial1.read();
    //Serial.println(lcd_gelen_data[byteSayisi-1]);
  }
  else
  {
    byteSayisi = 0;
  }

  if(lcd_gelen_data[0] == 0x65)
  {
    if((lcd_gelen_data[1] == 0x02) && (lcd_gelen_data[2] == 0x02))
    {
      Bahce_State = ! Bahce_State;
      Serial.println("Bahce Isik");
      digitalWrite(40, Bahce_State);
    }
    else if((lcd_gelen_data[1] == 0x01) && (lcd_gelen_data[2] == 0x06))
    {
    }
  }
}

Done uploading
Finished: 96%
Setting PC to entry point.: 96%
Running...
Success

64 LaunchPad w/ msp432 EMT (48MHz) on COM4
```

Şekil 3.26.Energia programı tanıtımı 1

Menülerin altındaki araç çubuğundaki simgelerin açıklamaları şöyledir:

Araç çubuğundaki simgelerin işlevleri:

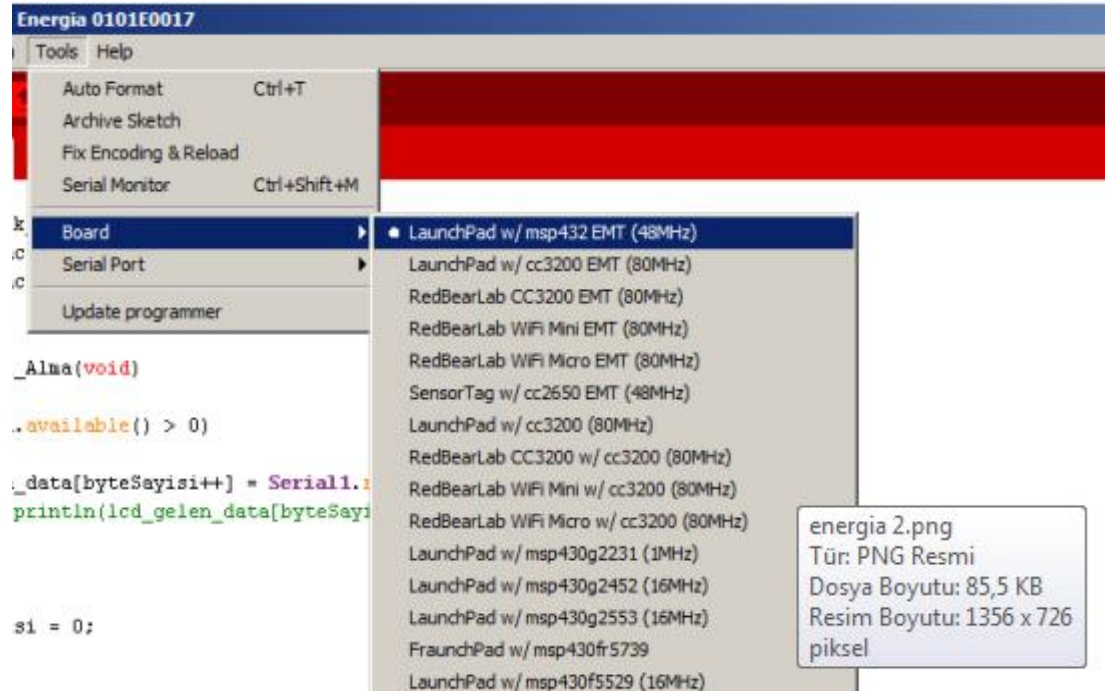
- **Verify:** Kodları derlemekte, hata varsa konsolda mesaj olarak göstermektedir.
- **Upload:** Kodları karta yüklemektedir.
- **New:** Yeni sketch (program dosyası) oluşturmaktadır.
- **Open:** Var olan program dosyasını açmaktadır.
- **Save:** Üzerinde çalışılan program dosyasını kayıt etmektedir.
- **Serial Monitor:** Seri iletişim monitörünü açmaktadır.

Energia’da 5 adet menü bulunur; **File, Edit, Sketch, Tools, Help**. Bunlardan **Tools**, aralarında en çok kullanılan ve elektronik kartın ve portun seçimi, seri monitörü görüntüleme, oluşturulan programı sıkıştırılmış dosya şeklinde kaydetme gibi işlemlerin yapıldığı menüdür.

Tools menüsü altındaki sekmeler:

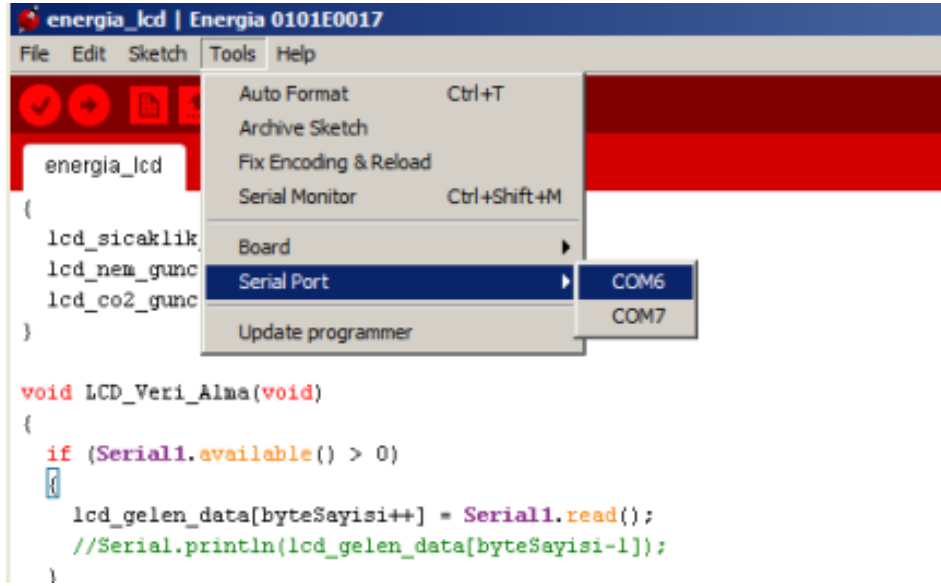
- **Auto Format:** Koddaki girintiler ve boşlukları okunabilirliği arttıracak şekilde ayarlamaktadır.
- **Archive Sketch:** Programı sıkıştırılmış klasör olarak kaydetmektedir.
- **Fix Encoding & Reload:** Programımızda karakter kodlamalarıyla ilgili bir problem varsa bu hataları düzeltmektedir.
- **Serial Monitor:** Seri iletişim monitörünü açar ve buradan gelen verileri görüntülemektedir.
- **Board:** Kullandığımız Launchpad (kart) seçimini yapmaktadır.
- **Serial Port:** Bilgisayarla kart arasında haberleşme amacıyla oluşturulan COM portunun seçimi yapılmaktadır.

Program açıldıktan sonra ilk olarak **Tools > Board** sekmeleri izlenerek mikrokontrolör modelimiz MSP432 seçilmektedir.



Şekil 3.27.Energia programı tanıtımı 2

Model seçildikten sonra ise bilgisayar ile bağlantı noktasını, yani portu, belirlenmesi gerekmektedir. Şekil 3.28'deki resimde olduğu gibi port, **Tools > Serial Port** sekmeleri izlenerek seçilmektedir.



Şekil 3.28.Energia programı tanıtımı 3

Projenin kodları yazılıp algoritmaları oluşturulduktan sonra çalışmanın tamamlanması için araç çubuğundaki **Verify** tuşuna tıklanıp derlenmektedir. Derleme sonrasında herhangi bir hata bulunmadığı takdirde yine araç çubuğundaki **Upload** tuşuna basılarak mikrokontrolöre yazılan kodlar gönderilmektedir.



Şekil 3.29.Energia programı tanıtımı 4

Yükleme tamamlandıktan sonra ekranda “Done Uploading” yazısı görüntülenmektedir.

```
if (message == "65 0 6 0 ffff ffff ffff") {digitalWrite(ledPin,Udal_State); Udal_State = !(Udal_State); }
if (message == "65 0 7 0 ffff ffff ffff") {digitalWrite(ledPin,Udal_State); Udal_State = !(Udal_State); }
if (message == "65 0 8 0 ffff ffff ffff") {digitalWrite(ledPin,Udal_State); Udal_State = !(Udal_State); }
```



Şekil 3.30.Energia programı tanıtımı 5

3.2.1.1.2. Energia Debug Ekranı

Energia programında yazılan senaryoya göre uygulama alanında verilerin nasıl gönderilip alındığı ve sistemin nasıl tepki verdiği debug ekranı vasıtasıyla takip edilebilmektedir. Bu sayede yapılması istenenler rahatlıkla gözlenebilmekte ve istenmeyen bir durum oluştuğunda yazılımsal olarak anında müdahale edilebilmektedir. Openweather API vasıtasıyla internet ortamından sisteme çekilen anlık mevsimsel sıcaklıklarının ve ev ortamında simule edilmiş yatağa bağımlı bir bireyden alınan anlık nabız değerlerinin Energia debug ekranında sürekli olarak takip edilebilmektedir.

3.2.1.2. Arduino Programı

Arduino devre kartları üzerinde dijital ve analog bağlantı noktaları barındıran, programlanabilir devre kartlarıdır.

Arduino'nun açık kaynak kodlu olmasından dolayı her türlü kaynak ve veri-kodlara internet ortamından rahatlıkla ulaşılabilir.

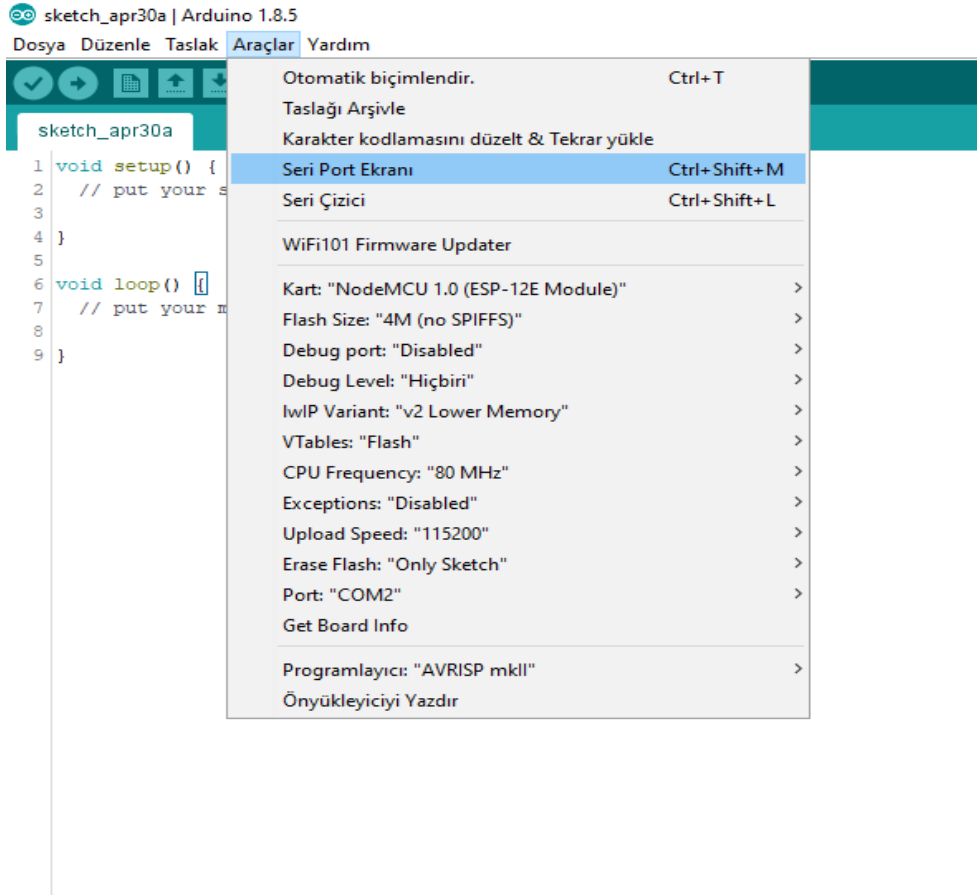
Arduino Programı, çalışmada mikrokontrolör ve sensörlerin kablosuz olarak haberleşebilmesi için kullanılmıştır. MPS432'ye UART yoluyla bağlanan NodeMCU ESP8266 ile parametreyi kontrol eden sensörlere bağlı NodeMCU ESP8266'lar arasındaki kablosuz veri akışı Arduino Programı ile derlenmiştir.

NodeMCU ESP8266 açık kaynak kodlu olduğu için Arduino ile uyumludur. Bu sayede Arduino Programıyla rahatlıkla kontrol edilebilmekte ve istenilen algoritmalar hafızasına yüklenebilmektedir.

3.2.1.2.1. Program Tanıtımı

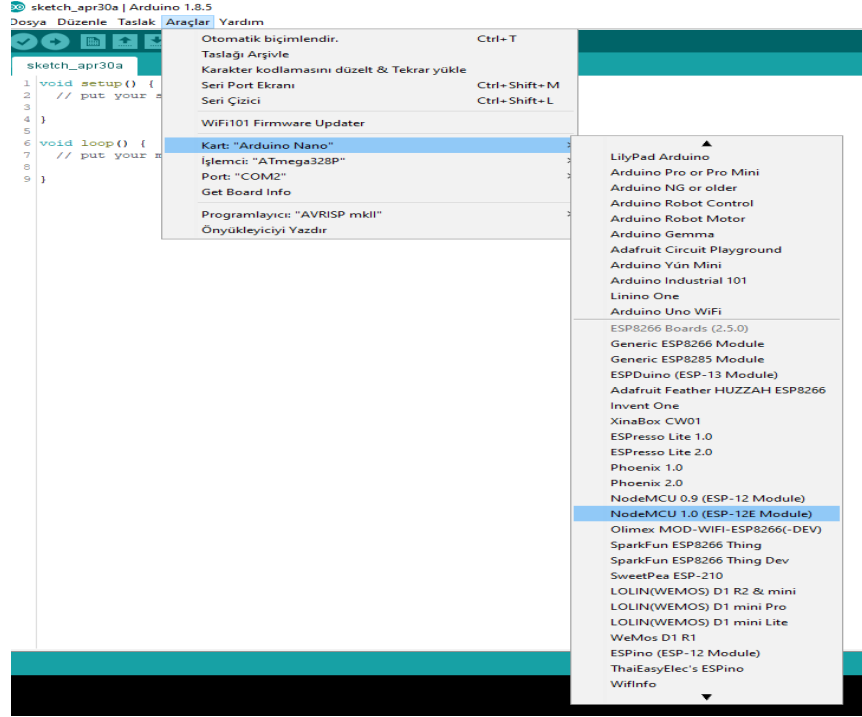
Arduino Bütünleşik Geliştirme Ortamı'nı (IDE) indirmek için Arduino'nun kendi sayfasından en güncel hali indirilebilmektedir. Hatasız bir şekilde kurulumu gerçekleştirdikten sonra IDE başlatılmaktadır.

NodeMCU ESP8266'ya program aktarmak için ilk olarak bilgisayara bağlanılan NodeMCU ESP8266 ve bağlantı portu Arduino Program menüsünden seçilmelidir. Bunun için IDE başlatıldıktan sonra **Tools** menüsü altında **Serial Port** seçeneğinden bağlanılan portun yanında tik işaretinin olup olmadığı kontrol edilmektedir. Eğer tik işareti yok ise tıklanarak işaretlenmektedir. Genelde tek port gözükeceği için burada birden fazla seçeneğe tik konulamamaktadır.



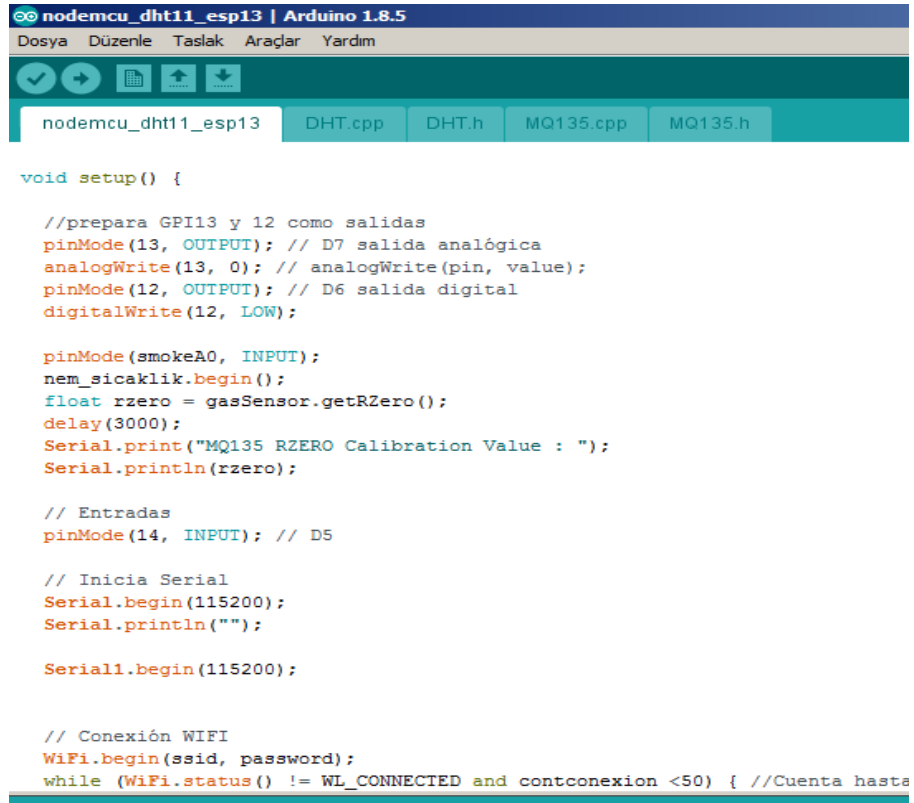
Şekil 3.31. Arduino programı tanıtımı 1

Kullanılan NodeMCU ESP8266 cihazı, yine Tools menüsü altında Board sekmesi üzerinden cihaz modeli olarak seçilmektedir.



Şekil 3.32.Arduino programı tanıtımı 2

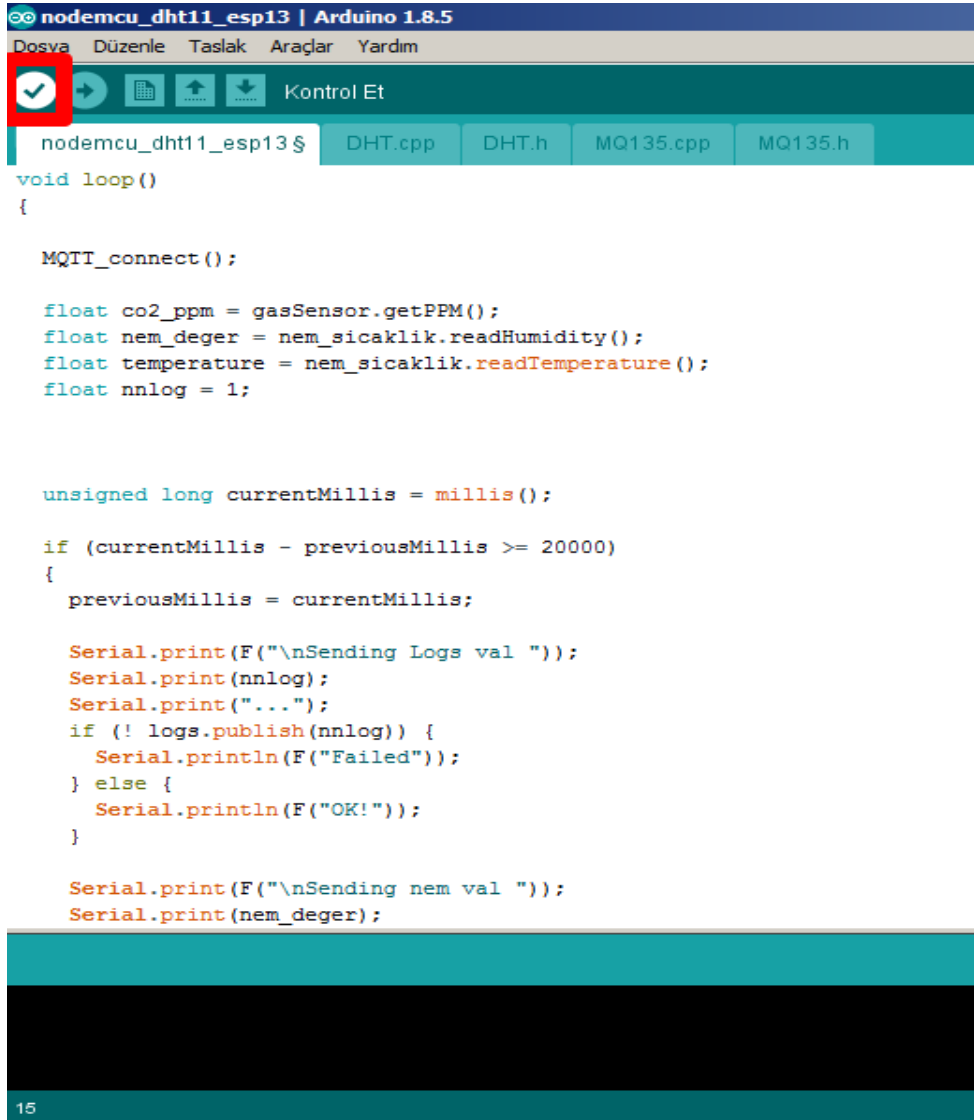
Programda **void setup()** kısmına yazılan fonksiyonlar, kart ilk enerji alıp çalıştığında sadece bir kere çalışmaktadır.



Şekil 3.33.Arduino programı tanıtımı 3

Kullanılan giriş/çıkış pinlerini, seri port konfigürasyonunu vb. ayarlar bu kısımdan yapılmaktadır. **Void loop()** kısmında ise, setup fonksiyonundaki komutlar çalıştıktan sonra kartın enerjisi kesilene kadar sürekli çalışacak olan fonksiyonlar bulunmaktadır.

Program yazıldıktan sonra karta yükleme yapılmak istendiğinde, öncelikle “**Verify**” seçeneğine tıklanmaktadır. Program, yazılan kodu öncelikle bilgisayarda bir klasöre kaydedilmesini istemekte, daha sonra da yazılan kodu derlemekte ve herhangi bir hata bulursa bu hatayı kullanıcıya bildirmektedir. Algoritmada herhangi bir hata olduğu zaman derleme engellenmekte ve uyarıya sebep olan hatalar gösterilmektedir.



```
nodemcu_dht11_esp13 | Arduino 1.8.5
Dosya Düzenle Taslak Araçlar Yardım
Kontrol Et
nodemcu_dht11_esp13$ DHT.cpp DHT.h MQ135.cpp MQ135.h
void loop()
{
    MQTT_connect();

    float co2_ppm = gasSensor.getPPM();
    float nem_deger = nem_sicaklik.readHumidity();
    float temperature = nem_sicaklik.readTemperature();
    float nnlog = 1;

    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= 20000)
    {
        previousMillis = currentMillis;

        Serial.print(F("\nSending Logs val "));
        Serial.print(nnlog);
        Serial.print("...");
        if (! logs.publish(nnlog)) {
            Serial.println(F("Failed"));
        } else {
            Serial.println(F("OK!"));
        }

        Serial.print(F("\nSending nem val "));
        Serial.print(nem_deger);
    }
}
```

Şekil 3.34.Arduino programı tanıtımı 4

Eğer yazılan algoritmada bir hata yoksa USB ile aracılığıyla, “**Upload**” seçeneğine tıklanarak kod NodeMCU'ya aktarılmaktadır. Yükleme sorunsuzca tamamlandığında program ekranında "Yükleme tamamlandı" yazısı görünmektedir.

```
34 }
35
36 void loop()
37 {
```

Şekil 3.35. Arduino programı tanıtımı 5

Arduino Programı ile yazılmış ve NodeMCU ESP8266'lara yüklenmiş olan bütün kodlar, "EKLER" bölümünde açıklamalarıyla birlikte verilmiştir.

3.2.1.3. Nextion Editör Programı

Nextion Editör programı, Nextion firmasının dokunmatik ekranlarının arayüz tasarımlarını oluşturabilmeleri için aynı firma tarafından geliştirilmiştir.

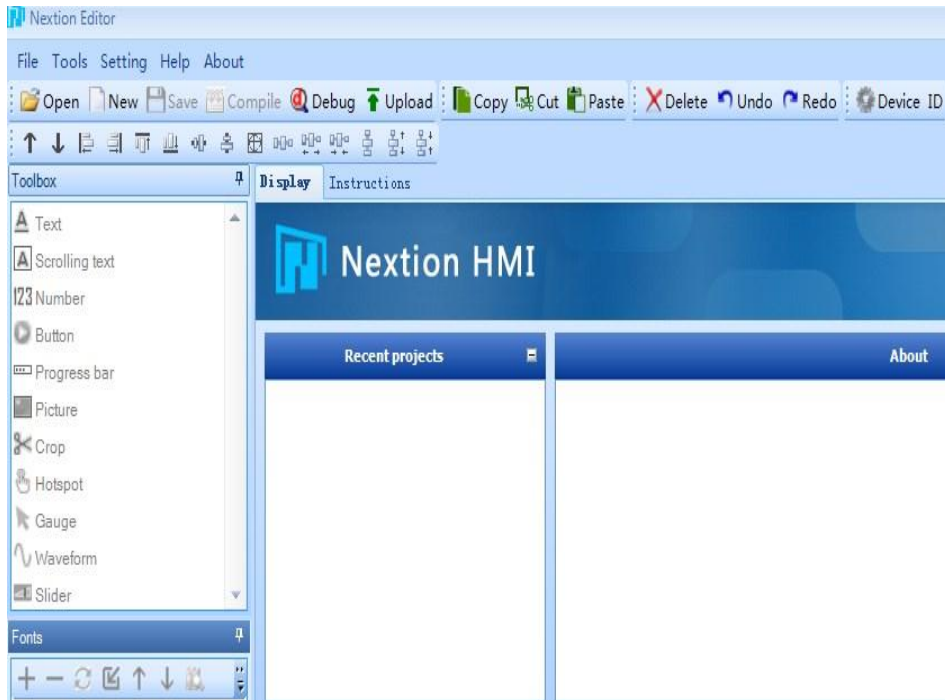
Nextion markasının ürettiği dokunmatik ekranlar (LCD) diğer grafik ve karakterlere sahip LCD' lere göre farklı bir mantık ile çalışmaktadır. Bu farklılık, Nextion dokunmatik ekranın ayrı olarak programlanabilmesinden kaynaklanır. Sadece Nextion dokunmatik ekranların tasarlanabilmesi için hazırlanan ve firmanın ürettiği özel bir arayüz uygulaması ile LCD üzerinde istenen tasarım arayüzü rahatlıkla oluşturabilir ve kişiselleştirebilir. Bu özellik sayesinde aynı zamanda geliştiriciye platformdan bağımsız bir şekilde tasarım imkânı sunar. MSP, Arduino, PIC vb. kartlar **Nextion Editör** uygulaması üzerinden geliştirilen arayüzler ile rahatlıkla kullanılabilir. Mikrokontrolörde olması gereken tek ihtiyaç UART bağlantı erişiminin olmasıdır. Çünkü dokunmatik ekran, kullanılacak mikrokontrolörler veya mikroişlemciler ile UART üzerinden haberleşmektedir. Mikroişlemci UART hattı üzerinden veri göndererek LCD üzerindeki değerleri değiştirebilir ya da ekran dokunmalarını UART üzerinden algılayabilmektedir.

Dokunmatik ekran olarak çalışmada 3,2 inç ve 4 MB hafızaya sahip modeli kullanılmıştır. 4 MB hafıza arayüz için yeterli gelmediği durumda ekran üzerinde bulunan SD kart slotu kullanılarak arayüz çalıştırabilmektedir. SD kart kullanılmak istenmediğinde basit bir FTDI dönüştürücü ile ekran programlanabilmektedir.

3.2.1.3.1. Program Tanıtımı

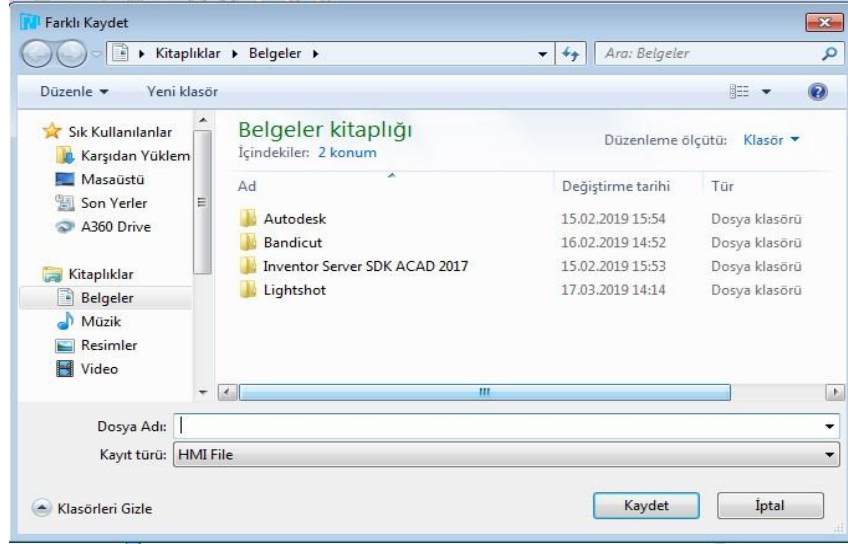
Nextion Editör Programına, Nextion'un kendi sayfasından son sürümü güncel olarak erişilip indirilebilmektedir. Kurulum işlemi gerçekleştirdikten sonra Nextion Editör Programı başlatılmaktadır.

Program açıldıktan sonra tasarım arayüz penceresi görünmektedir. Arayüz çok fazla fonksiyona sahip olmadığından sade ve kolay anlaşılmalıdır. Birçok programda olduğu gibi “File”, “Tool”, “Help” gibi menüleri vardır. Sol tarafta ise “Toolbox” menüsü bulunmaktadır. Üst tarafta yeni proje açmak ve var olan dosyaları açmak için “New” ve “Open” düğmeleri mevcuttur. Burada diğer tasarım arayüzlerinden farklı olarak “Compile“, “Debug” ve “Upload” düğmeleri mevcuttur. Bu düğmeler, oluşturulan ekran arayüzünü sırası ile derlemek, simule etmek ve Nextion Ekranına yüklemek içindir. Simule etme özelliğinin olması özellikle normal OLED ekranlar ile yapılan tasarımlara kıyasla işi oldukça hızlandıran bir özelliktir. Oluşturulan tasarım arayüzü derlendikten sonra “Debug” butonuna basarak Nextion ekrana yüklemeyen bilgisayar üzerinde nasıl görüldüğüne bakılabilmekte ve etkileşime geçildiğinde (tıklama, kaydırma vb.) ekrana hangi verilerin geleceğini simule etme imkânı sağlamaktadır.



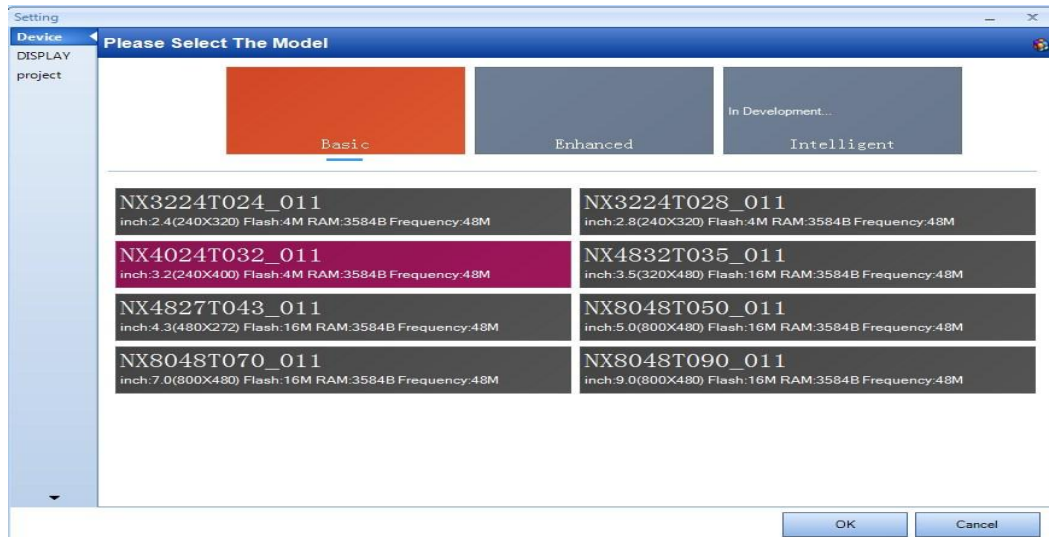
Şekil 3.36.Nextion Editör programı tanıtımı 1

Yeni proje oluşturmak için “New” butonuna basılmakta ve oluşturulan veya oluşturulacak projenin “*.HMI” uzantılı dosyasınının nereye kayıt edileceği seçilmektedir.



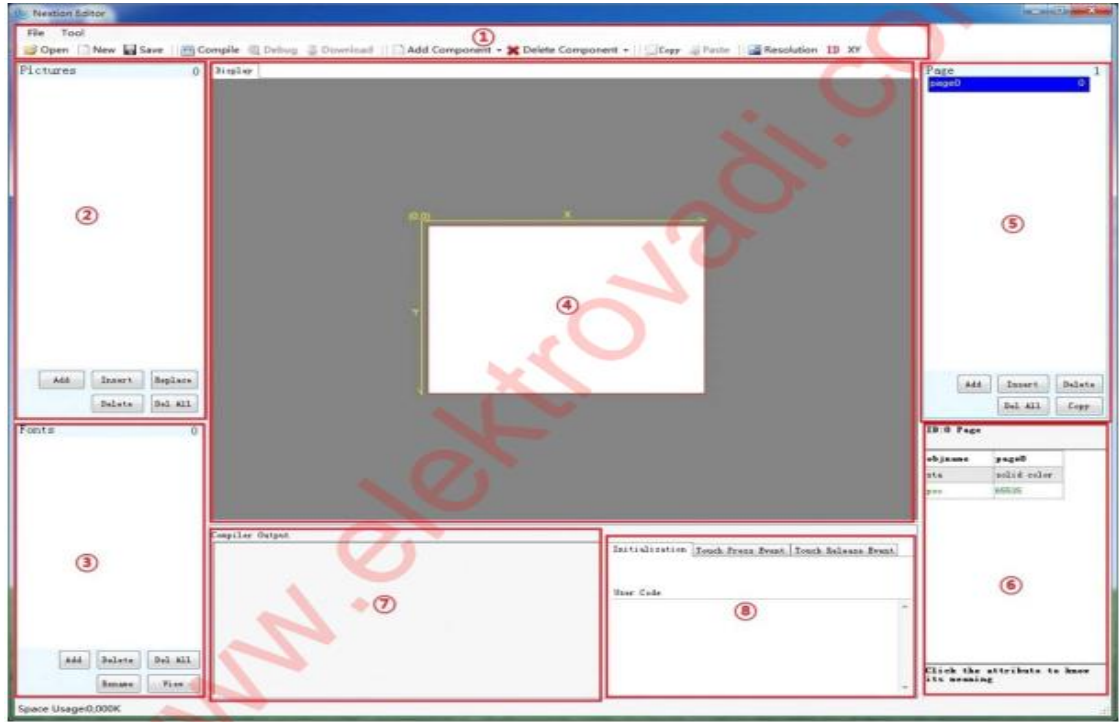
Şekil 3.37.Nextion Editör programı tanıtımı 2

Dosya kayıt yeri belirlendikten sonra Nextion Ekran modelinin seçildiği ekran görüntülenmektedir. Burada kullanılan ekran boyutlarının doğru seçilmesi gerekmektedir. Yanlış seçilmesi durumunda oluşturulan tasarım dokunmatik ekrana yüklenmemektedir. Çalışmada kullanılan modül **3,2 inç ve 240*400** boyutlarında olduğu için ilgili ekran seçeneği seçilmekte ve dolayısıyla Nextion Editör tasarım ekranının boyutları bu ebatta olmaktadır.



Şekil 3.38.Nextion Editör programı tanıtımı 3

Tasarım ekranındaki herbir arayüz penceresine ait bilgiler aşağıdaki resimde gösterilmiştir.



Şekil 3.39.Nextion Editör programı tanıtımı 4

1.Anasayfa

2.Resim Kütüphanesi: Çalışmada kullanılacak olan resimler buradan yüklenmektedir.

3. Font Kütüphanesi: Çalışmada kullanılacak yazı fontları belirlenmektedir.

4. Ekran Alanı

5. Sayfa Alanı: Çalışmada kullanılacak sayfalar buradan eklenmektedir.

6. Özellik Düzenleme Alanı: Seçilen bileşenin özellikleri düzenlenmektedir.

7. Derleyici Çıkışı: Eğer bir hata meydana gelirse burada gösterilmektedir.

8.Etkinlik Alanı: Seçilen bileşenin çalıştırması istenen kısa kodlar buraya yazılmaktadır.

Nextion Editör programından bu konu başlığı altında genel hatlarıyla bahsedilmiş olup, çalışmada ana kontrol ünitesi olarak kullanılan 3,2 inç dokunmatik ekran arayüzünün tasarlanma aşamaları ayrıntılı olarak “Kontrol” bölümünde “Nextion Editör Arayüz Tasarımı” kısmında anlatılmıştır.

3.2.1.4. Eagle Programı

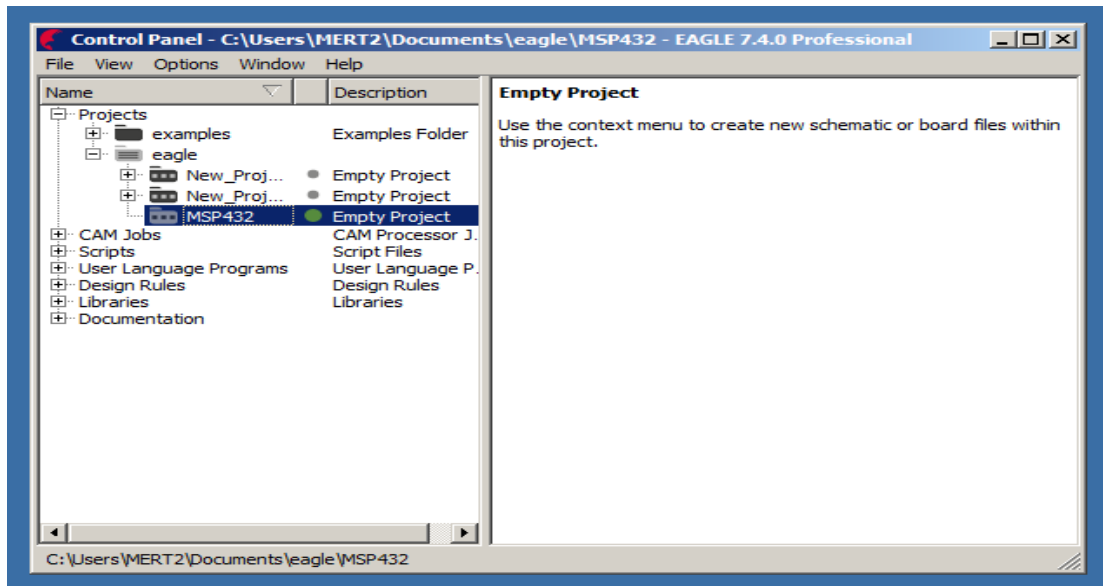
Eagle programı PCB (printed circuit board) çizim programıdır. Üç ana modülden oluşmaktadır; Çizim editörü (Devre), şema editörü(PCB), Auto Route (Düzenleme) modülü. Tüm bu modüller, tek bir arayüz üzerinden kontrol edildiğinden modüller arası geçişte dönüştürme işlemi gerek olmaması büyük bir avantajdır.

Eagle, Proteus veya Multisim gibi devre çizim programlarında olduğu gibi simulasyon içermemektedir. Ancak içerdiği komponent sayısı ile diğer programlara daha çok avantaj sunmaktadır. Eagle'ın çok geniş bir kütüphanesinin olmasının yanı sıra dışarıdan da kütüphane eklenebilmektedir. Yani Eagle içerisinde bulunmayan bir elemanın kütüphanesini oluşturarak ya da dışarıdan bularak program içerisine dahil edebilmektedir.

3.2.1.4.1. Program Tanıtımı

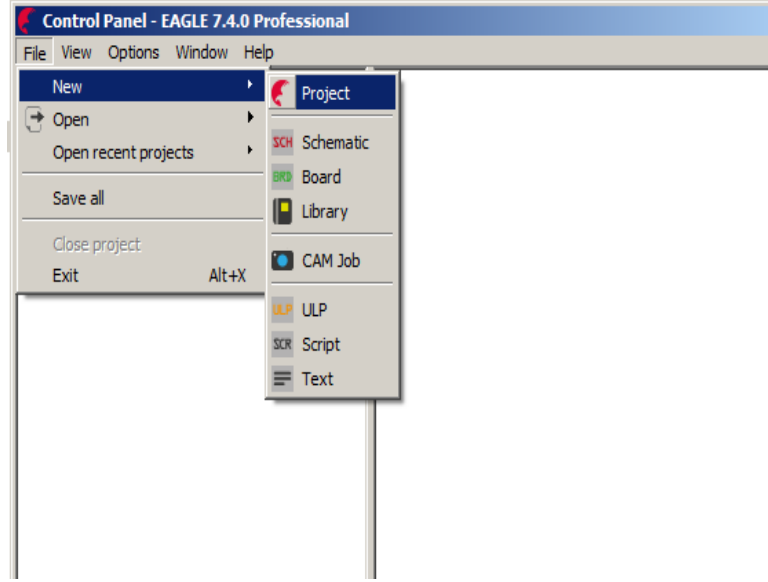
Eagle ücretsiz sürümünde 10×10' luk bir alanda çalışmaya izin vermektedir. 10 binlerce komponent desteği ile bütün devreleri profesyonel bir şekilde çizmeye imkân sağlamaktadır. Eagle'ın son sürümü kendi anasayfasından indirilebilmektedir.

Eagle'ın ilk açılış arayüzü Şekil 3.40'daki gibidir. Bu arayüzden var olan projeler görülebilmekte ve projelerin şematik ve PCB kısımları arasında geçiş yapılabilmektedir. Ya da üstteki **File** menüsünden yeni proje oluşturabilmektedir.



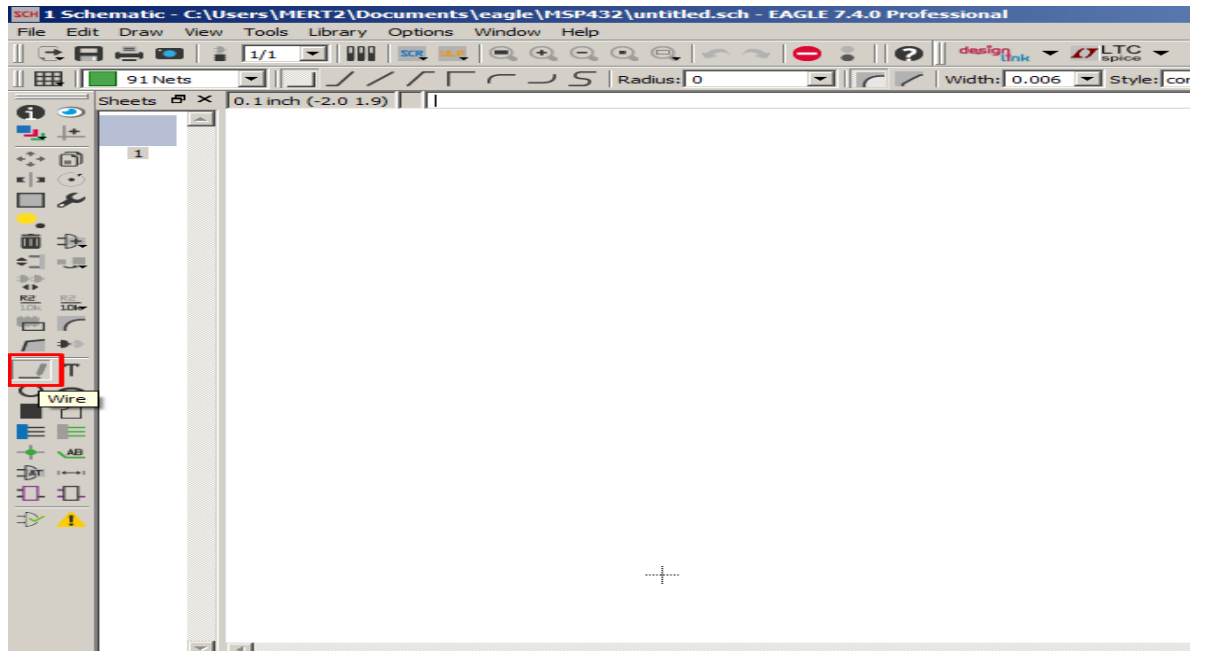
Şekil 3.40.Eagle programı tanıtımı 1

Eagle' ın Şekil 3.41'de gösterilen şematik ekranını açmak için var olan bir proje sağ tıklanır ya da **“File”** menüsünden şematik ekranı seçilir. Şematik ekrana elemanlar soldaki menü üzerinde **“Add Part”** menüsü altından eklenmektedir.



Şekil 3.41.Eagle programı tanıtımı 2

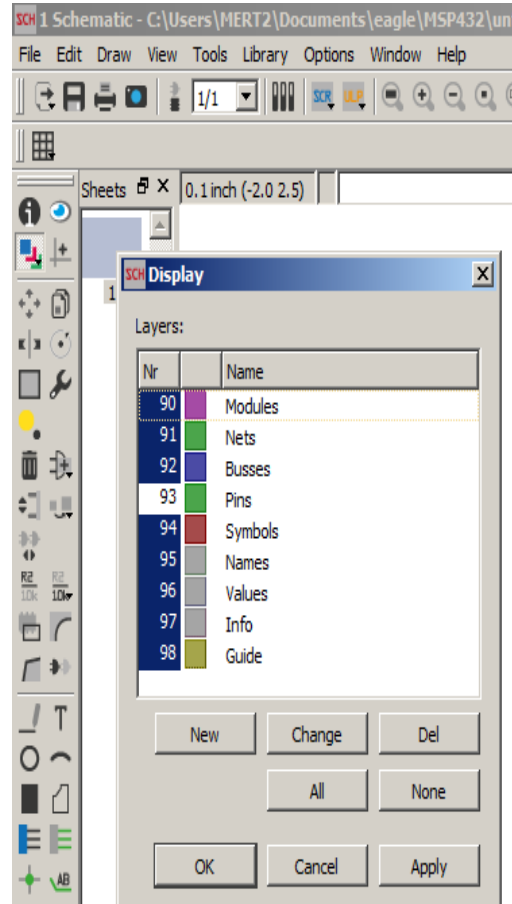
Elektronik yolların çizilebilmesi için ise yine soldaki menü üzerinden **“Wire”** aracı kullanılmaktadır. Ancak bu araç ile yollar birleştirilememekte ve atlama şeklinde gerçekleştirilmektedir. Birleştirebilmek için **“junction”** aracı kullanılmaktadır.



Şekil 3.42.Eagle programı tanıtımı 3

Aynı devre üzerinden PCB tasarımına geçmek için üstteki menüden “**Switch Board**” seçeneğine tıklanmaktadır. Devrenin PCB çizimine geçildikten sonra elemanlar solda karışık bir şekilde gelmektedir. Bunları yerleştirmek ve düzenlemek kullanıcının isteğine bağlıdır. Bu iş için bir Eagle modülü olan Auto-Router kullanılmaktadır. Eagle size baskı devresi için birçok katman desteğini sorunsuzca sağlamaktadır.

Devrenin şematik ve PCB çizimi bittikten sonra çıktı için hazır hale getirilmesi gerekmektedir. Bunun için PCB’deki gereksiz komponent resimleri ve isimlerinin kaldırılması gerekmektedir. Bu ayar PCB ekranında araçlar menüsündeki “**Layer Settings**” kısmından yapılmaktadır. PCB görüntüsü gereksiz bileşenlerden arındırılmış ve çıktıya hazır hale getirildikten sonra dikkat edilmesi gereken nokta devredeki katmanlardır. Eğer devre üzerinde 2 veya daha fazla katman var ise her katmanı ayrı olarak gösterip çıktısının alınması gerekmektedir.



Şekil 3.43.Eagle programı tanıtımı 4

Çalışmada Eagle Programı, MSP432 mikrokontrolörün, NodeMCU ESP8266 modulünün ve kullanılan elektronik cihazların devre şemalarını ve kablosuz haberleşme şemalarını gösterebilmek için kullanılmıştır.

3.2.1.5. SolidWorks Programı

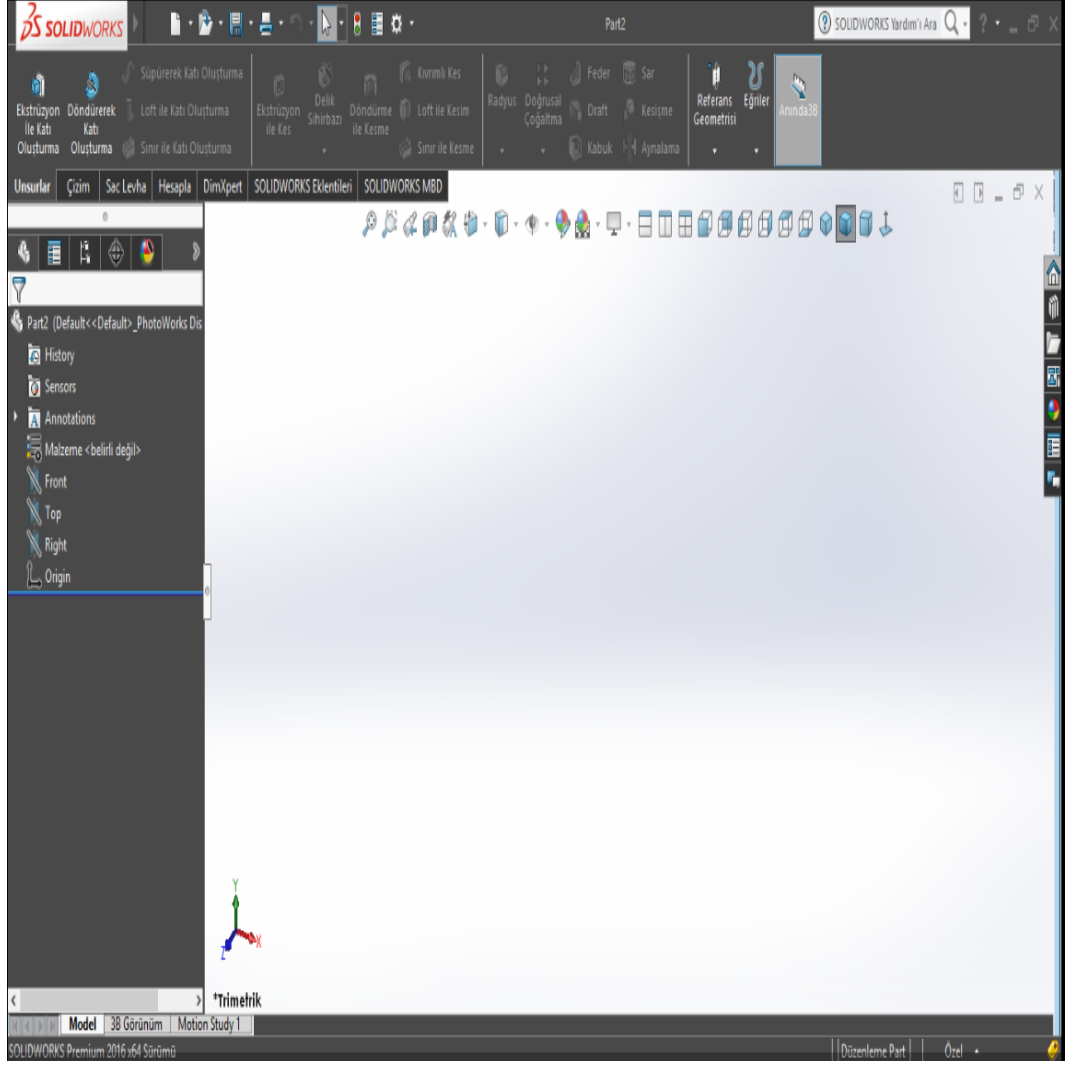
SolidWorks, bir CAD programıdır; her türlü makine, ürün ve tesis tasarımında kullanıcının hızlı ve kolay bir şekilde çizim yapmasına olanak tanımaktadır. İmalat kısmında başarılı ve kolaylıklar sağlayan bir programdır. Talaşlı imalat, kaynaklı imalat, plastik enjeksiyon ve döküm yöntemi ile imalat gibi çoğu yöntemeye uygun tasarım yapmaya olanak vermektedir. Online kütüphanelerle tasarım yapmak daha da kolay hale getirmektedir.

SolidWorks diğer CAD programları gibi parametrik özellik tabanlı modelleme teknolojisine sahiptir. SolidWorks kullanarak ölçülü ve vektör tabanlı üç boyutlu çizimler yapılabilmektedir. Esnek yapısı sebebiyle yapılan çalışmaları pek çok program üstünde kullanabilmektedir. Yapılan çizimler üzerinden teknik veriler alabilmek mümkündür. Montaj modülü yardımı ile ayrı ayrı yapılan çalışmaları bir araya getirebilmektedir. Yapılan parçalar üstünde statik, darbe gibi testler yapabilmekte ve çok detaylı analiz sonuçlarına ulaşabilmektedir. Yapılan çalışmalar animasyon eklentisi ile hareketlendirebilmektedir.

3.2.1.5.1. Program Tanıtımı

SolidWorks Programı, lisanslı bir program olup kendi anasayfasından kolayca indirilebilmektedir. **“Sketch”** komutu ile 3D hale getirilecek olan parçanın profili çizilebilmektedir. Daha sonra ise yapılan çizim farklı yöntemler ile 3D hale getirilebilmektedir. Bu yöntemlere sac levha, yüzey, profil ve unsur gibi seçenekler örnek verilebilmektedir.

Çizimi yapılan parçalar birden fazla olduğu durumda program çizilen parçaların bir araya getirilmesinde kullanıcıya yardımcı olarak parçaların montajının yapılmasına olanak vermektedir.



Şekil 3.44.Solid Works programı tanıtımı 1

Montajı tamamlanan bir ürün, üretim için hazır olması durumunda teknik resme ihtiyacı doğmaktadır. Bu aşamada da program sayesinde, çizimi yapılan ürünün teknik resmi oluşturulabilmektedir. Ürünün analizi de SolidWorks simülasyon paketi ile gerçekleştirebilmektedir.

Tasarımı tamamlanan bir ürünün nasıl görüldüğüne Render görüntüsü elde edilerek bakılabilmektedir.

SolidWorks programı, çalışmadaki ev maketinin istenilen boyutlarda çizilebilmesini dolayısıyla da CNC tezgâhında sac levhaların işlenebilmesini sağlamıştır.

3.2.2. Kullanılan Haberleşme Protokolleri

Akıllı ev otomasyonu uygulamasında sistemin kontrolü, hem internet aracılığıyla Web Server üzerinden hem de ev ortamında bulunan Nextion 3,2 inç dokunmatik ekran aracılığıyla yapılmaktadır. Web Servere sistem üzerinden internet aracılığıyla sürekli veri gönderilmekte ve burada kayıt altına alınmaktadır. Kullanıcının istekleri de yine internet aracılığıyla evdeki kontrol sistemine veri olarak iletilmekte ve sistemin tepki vermesi sağlanmaktadır. Bu veri alışverişi **MQTT (Message Queuing Telemetry Transport) Protokolü** sayesinde gerçekleşmektedir.

İnternet üzerinden MQTT Protokolüyle veya Nextion dokunmatik ekran üzerinden kullanıcının manuel girişiyle gelen veriler kontrol sistemine dolayısıyla mikrokontrolöre (MSP432) gelerek yazılan algoritmaya göre yorumlanmaktadır. Daha sonra **UART (Universal Asynchronous Receiver Transmitter) Protokolü** sayesinde Node MCU ESP8266 modülü ile haberleşmektedir. Kontrol edilmesi istenen parametre sensörüne; ESP8266 bağlı ise bu modülden **WiFi (Wireless Fidelity) Protokolü** ile kablosuz olarak, ESP8266 bağlı değilse **UART** Protokolü ile kablolu olarak veri gönderilmektedir. Böylece sistem kullanıcının isteğine göre tepki verebilecektir.

Bu bölümde MQTT ve UART Protokolleri detaylı olarak anlatılmış olup WiFi Protokolüne “Genel Bilgiler” kısmında ayrıntılı olarak anlatıldığı için bu başlık altında değinilmemiştir.

3.2.2.1. MQTT Haberleşme Protokolü

MQTT (Message Queuing Telemetry Transport), yayınlama ve abone olma mantığına dayanan telemetri (kablosuz ya da sabit bir ağ aracılığıyla cihazların uzaktan izlenebilmesi ve denetlenmesi) mesajlaşma protokolüdür. TCP/IP üzerinden mesaj yayınlayıp almayı sağlamayan oldukça hafif bir protokoldür. Hafif protokol olması diğer benzer protokollerden ayıran en önemli özelliklerinden biridir.

Bu haberleşme trafiğini kontrol eden yöneticiye **Broker**, mesaj yayınına **Publish** ve bu mesaj yayınına abone olanlara **Subscribe** denmektedir. Şekil 3.45' de bu kavramlar arasındaki ilişki gösterilmiştir.



Şekil 3.45.MQTT kavramları arasındaki ilişki

MQTT de asenkron bir haberleşme kullanılmaktadır. Mesajı yayınlayan ve mesaja abone olanlar arasında veriler asenkron (eş-zamansız) olarak taşınmaktadır. Şekil 3.45'deki görselde sıcaklık verileri (Publish) haberleşme trafiğini kontrol eden yöneticiye (Broker) gönderilmektedir. Broker bu verileri abone (Subscribe) online olduğu anda iletmektedir.

MQTT kısaca buluttan kontrol ve gözlenmeye ihtiyacı olan küçük cihazlar için network oluşturmaya imkân sağlayan protokoldür. MQTT, nadiren rastlanılan bant genişliği kısıtlamalarından ya da güvensiz bağlantılardan dolayı oluşan gecikmelerin yaşandığı kablosuz ağlar için iyi bir seçimdir. Abone olan birimden broker (aracı) birimine bağlantı yapılmakta ve broker, abone birim online olduğunda bu mesajı ona tekrar iletmektedir.

Yayınlayan birimden broker birimine olan bağlantı ise kesilmesi gerekmekte ve broker birimi, bağlantıyı kapattıktan sonra yayıncıdan aldığı komutu abone birime iletmektedir.

MQTT diğer haberleşme protokollerine göre daha basit bir yapıya sahip ve kolayca projelere entegre edilebilmektedir. Minimum kaynak tüketimi sayesinde özellikle M2M (Machine-to-machine) haberleşmesinde kullanılmaktadır. Bu da MQTT'yi IoT (Internet of Things) projeleri için vazgeçilmez bir mesajlaşma protokolü haline getirmektedir.

MQTT Genel Özellikleri;

- Asenkron (eş-zamansız) çalışan bir protokoldür.
- Güvenlik olarak SSL / TLS desteklemektedir.
- Minimum kaynak kullanımında bulunmaktadır.
- Broker üzerinden haberleşme temeline dayanmaktadır.
- Bilgiler MQTT protokolü üzerinden çok hızlı bir şekilde iletilebilmektedir. (ms düzeyinde bir haberleşme)
- TCP/IP nin kullanıldığı Windows, Linux, MacOS, Android ve iOS işletim sistemlerinde çalışmaktadır.

IoT (Nesnelerin İnterneti), çeşitli haberleşme protokolleri sayesinde birbirleri ile haberleşen ve birbirine bağlanarak, bilgi paylaşarak akıllı bir ağ oluşturmuş cihazlar sistemidir.

IoT ile hayatımızda birçok şey kolay hale gelmektedir. Daha da akıllı hale gelen evlerde elektronik cihazlar nesnelerin interneti ile birlikte farklı cihazlar ile haberleşip belirlenen kurallara göre (bazı durumlarda kendi yapay zekâsını kullanarak) hayatı büyük ölçüde kolaylaştırmaktadır.

Şekil 3.46 ve Şekil 3.47'de Arduino programında kodları yazılan MQ-135 hava kalite ve DHT-11 nem-sıcaklık sensörlerinin MQTT haberleşme protokolü ile Adafruit IO'ya nasıl veri gönderildiği gösterilmiştir.

```

NODEMCU_CO2 MQ135.cpp MQ135.h
1 #include <ESP8266WiFi.h>
2 #include <Adafruit_MQTT.h>
3 #include <Adafruit_MQTT_Client.h>
4 #include "MQ135.h"
5
6 /***** WiFi Access Point *****/
7
8 #define WLAN_SSID "MERIKSK35"
9 #define WLAN_PASS "mert1488191235"
10
11 /***** Adafruit.io Setup *****/
12
13 #define AIO_SERVER "io.adafruit.com"
14 #define AIO_SERVERPORT 1883 // u
15 #define AIO_USERNAME "ozensine"
16 #define AIO_KEY "adb4a25d18e4a44ba3e14799b1"
17
18 /***** Global State *****/
19
20 WiFiClient client; // MQTT server bağlanmak için cla
21 Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_S
22
23
24

```

CO2 Gönderiliyor 262.99...OK!
CO2 Gönderiliyor 258.53...OK!
CO2 Gönderiliyor 256.32...OK!
CO2 Gönderiliyor 254.12...OK!
CO2 Gönderiliyor 251.94...OK!
CO2 Gönderiliyor 247.62...OK!
CO2 Gönderiliyor 244.41...OK!
CO2 Gönderiliyor 240.18...OK!
CO2 Gönderiliyor 237.05...OK!
CO2 Gönderiliyor 233.95...OK!
CO2 Gönderiliyor 229.86...OK!
CO2 Gönderiliyor 226.82...OK!

Yükleme tamamlandı. Otomatik Kaydırma

Şekil 3.46.MQ-135'in MQTT ile veri göndermesi

```

NODEMCU_NEM_SICAK | Arduino 1.8.5
Dosya Düzenle Taslak Araçlar Yardım
NODEMCU_NEM_SICAK DHT.cpp DHT.h MQ135.cpp MQ135.h
151 }
152 mqtt.processPackets(500); // bu bizim 'gelen abonelik paketlerini bekle ve
153
154 // MQTT bağlantısını canlı tutmak için sunucuya ping atma
155
156 if(! mqtt.ping())
157 {
158 mqtt.disconnect();
159 }
160
161 COM15
162
163
164 .....
165 WiFi bağlantısı Kuruldu
166 192.168.1.24
167 Connecting to MQTT... MQTT Connected!
168
169 Sıcaklık Gönderiliyor nan...OK!
170
171 Nem Gönderiliyor nan...OK!
172
173 Sıcaklık Gönderiliyor 30.00...OK!
174
175 Nem Gönderiliyor 35.00...OK!

```

Yükle Otomatik Kaydırma

[67%]
[100%]

Şekil 3.47.DHT-11'in MQTT ile veri göndermesi

3.2.2.2. UART Haberleşme Protokolü

UART (Universal Asynchronous Receiver Transmitter), bilgisayar ile mikrokontrolör veya mikrokontrolör ile çevre birimler arasında haberleşmeyi sağlayan haberleşme protokolüdür. Asenkron olarak çalıştığı için herhangi bir "clock" ihtiyacı duymamaktadır.

UART iletişim protokolü bir seri iletişim protokolüdür. Diğer I2C veya SPI ile iki mikrokontrolör yada entegre birbiri ile haberleşme imkanı sunarken UART'da farklı olarak mikrokontrolör - bilgisayar ile ya da bilgisayar-bilgisayar haberleşmesi sağlamaktadır.

UART Genel Özellikleri;

- Sistem maliyetsizdir sadece 2 kablo ve GND ile bağlantı kurulmaktadır. (Kablosuz iletişimde GND kullanılmamaktadır.)
- RF verici -alıcı, ethernet, internet gibi sistemlerde ve PC – PIC ya da PC – PC bağlantılarında kullanılmaktadır.
- Gerekli tedbirler ile I2C veya SPI gibi seri iletişim protokollerinin aksine uzun mesafelerde kullanılabilir.
- İletişim esnasında herhangi bir saat darbesine ihtiyaç yoktur.
- Yüksek hız istenmeyen uygulamalarda kullanılabilir. Büyük veri gönderimlerinde zaman almaktadır.
- Diğer seri iletişim protokolleri gibi aynı hat üzerine birden fazla ünite bağlanamamaktadır. Herhangi bir şekilde master-slave ilişkisi yoktur iletişim sadece iki ünite arasında gerçekleşmektedir. Sadece 1 alıcı ve 1 verici ile sistem çalışmaktadır.
- Haberleşme sağlayan alıcı ve verici aynı hızda haberleşmek zorundadır.
- Seri iletişim protokolü olduğundan bütün veri parçalanıp seri olarak gönderilmelidir. Tek seferde veri gönderim boyutu en fazla 9 bit ile sınırlıdır.

UART haberleşmesini gerçekleştirirken ilk olarak **baudrate (veri taşıma hızı)** ayarlanması gerekmektedir. Veri taşıma hızı çok çeşitli aralıklarda olabilmekte ancak piyasada yaygın olarak kullanılan baudrate'ler 4800, 9600, 57600, 115200,

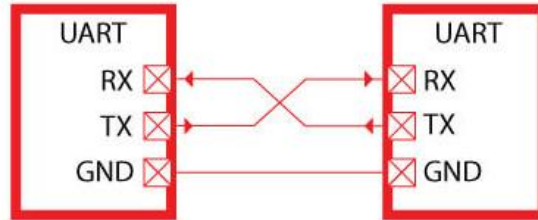
230400 ve 921600 hızlarına sahiptir. 921600, hızlı işlem gerektiren işlerde kullanıldığı için pek tercih edilmemektedir.

Çalışmada kullanılan MSP432 ile NodeMCU ESP8266 haberleşmesini sağlayan UART bağlantısı 115200 baudrate hızında yapılmaktadır. Dolayısıyla bu haberleşmede saniyede yaklaşık olarak 115200 byte veri iletimi sağlanmaktadır.

Haberleşme işlemi bir başlangıç bitinden sonra data bitleri, ardından parity biti ve son olarak da bitiş biti gönderilerek tamamlanmaktadır. Bu işlem sırasında data uzunluğu ve parity biti opsiyonel olarak değişkenlik gösterebilmektedir.

Bu haberleşme tipinin kullanılabilmesi için alıcı ve vericinin veri taşıma hızlarının (baudrate) aynı olması gerekmektedir. Bunun sebebi ise aktarım sırasında oluşabilecek hataları minimuma indirebilmektir. Hata paylarının tolere edilebilir seviyede olması durumunda herhangi bir sorun teşkil etmemektedir.

UART haberleşmesinin yapılabilmesi için MSP432 mikrokontrolöründeki ve NodeMCU ESP8266 modülünde daha önceden tanımlanmış olan UART pinleri kullanılmaktadır. Şekil 3.48’de gösterildiği gibi NodeMCU modülündeki RX-TX pinleri ile MSP432’nin RX-TX pinleri ile ters olarak bağlandıktan sonra haberleşme sağlanmaktadır.



Şekil 3.48. UART bağlantı şeması

3.2.3. Kullanılan Kontrol Arayüzü Tasarımları

Akıllı ev otomasyonunda kontrol sistemi hem internet üzerinden Web Server aracılığıyla hem de ev ortamında bulunan dokunmatik ekran ile yönetilebilmektedir. Kullanıcının ev otomasyon kontrol sistemini rahat bir şekilde yönetebilmesi amacıyla her iki yönteme ait kontrol arayüzleri özel olarak tasarlanmıştır. Bu sayede kullanıcının ev otomasyonunu aktif bir şekilde yönlendirebilmesi ve anlık olarak sistemi sürekli takip edebilmesi sağlanmıştır.

Web Server üzerinden MQTT Protokolüyle kontrol sistemin haberleşmesi için **Adafruit IO Sistemi** kullanılmış ve kontrol arayüzü tasarlanmıştır. Adafruit Kontrol Arayüzü sayesinde sistem her an internet üzerinden izlenebilmekte, müdahale edilebilmekte ve ev ortamından çekilen veriler kayıt altına alınabilmektedir. Kayıt altına alınan verilere istenildiği zaman ulaşılabilen ve kaydedilen verilerin periyodik olarak grafiksel istatistikleri izlenebilmektedir.

Nextion 3,2 inç dokunmatik ekran aracılığıyla UART Protokolüyle kontrol sisteminin haberleşmesi için daha öne "Kullanılan Programlar" bölümünde anlatılan **Nextion Editör** programı ile ekran arayüzü tasarlanmıştır. Bu ekran ile kullanıcı, internet erişimi olmadığı zamanlar da bile otomasyon sistemini kontrol edebilecektir.

3.2.3.1. Adafruit IO Sistemi Kontrol Arayüzü

Adafruit Endüstri, birçok elektronik ürünleri, bileşenleri ve aksesuarları tasarlayan ve üreten açık kaynaklı bir donanım şirkettir. Adafruit IO, bir bulut hizmeti olup internet üzerinden bağlanılarak veri alma ve veri depolama olanağı sağlamaktadır.

Bu çalışmada internet üzerinden otomasyon sisteminin yönetimi bu kontrol arayüzü ile sağlanmaktadır. Adafruit IO, projenin verilerini görüntülemek, cevaplamak ve etkileşimde bulunmak için tasarlanmış bir platformdur. Ayrıca, kayıt altına alınan verileri gizli ve güvenli bir şekilde tutmaktadır. (Bu verileri asla başka platformlarda yayınlamamakta ve satmamaktadır.)

Veri kaydı için veya mikrokontrolör ile web üzerinden iletişim kurmak için çok sayıda servis bulunmakta, ancak bu hizmetler genellikle çok karmaşık veya kullanımları kolay değildir. Adafruit IO, kullanımı gayet kolay ve anlaşılır, arayüz ekranı ise sade bir yapıya sahiptir. Kontrol arayüzü tasarımı, oldukça eğlencelidir.

Adafruit IO, birden fazla veri beslemesini işleyebilmekte ve görselleştirebilmektedir. Ya bir hava kalite sensöründen gelen veri ya da bir sıcaklık-nem sensöründen gelen veri görüntülenmekte ve ekrandaki bir butonla odadaki klimayı açarak istenilen şartların oluşması sağlanmaktadır.

Adafruit IO'ya entegre edilmiş, gösterge panoları ile verilerin grafiklendirilmesine, ölçülmesine, kaydedilmesine ve görüntülenmesine izin vermektedir. Yapılan panolara dünyanın her yerinden görüntüleyebilme olanağı sunmaktadır.

Verileri kontrol etmek ve sistem tepkisini gözlemlenebilmesi için Adafruit IO içindeki tetikleyiciler kullanabilmektedir. Sistem çevrimdışı olduğunda kullanıcıya e-posta göndermesi için tetikleyiciler yapılandırılabilmekte ve örneğin sıcaklık algılayıcısının çok ısınmasından haberdar olunabilmektedir.

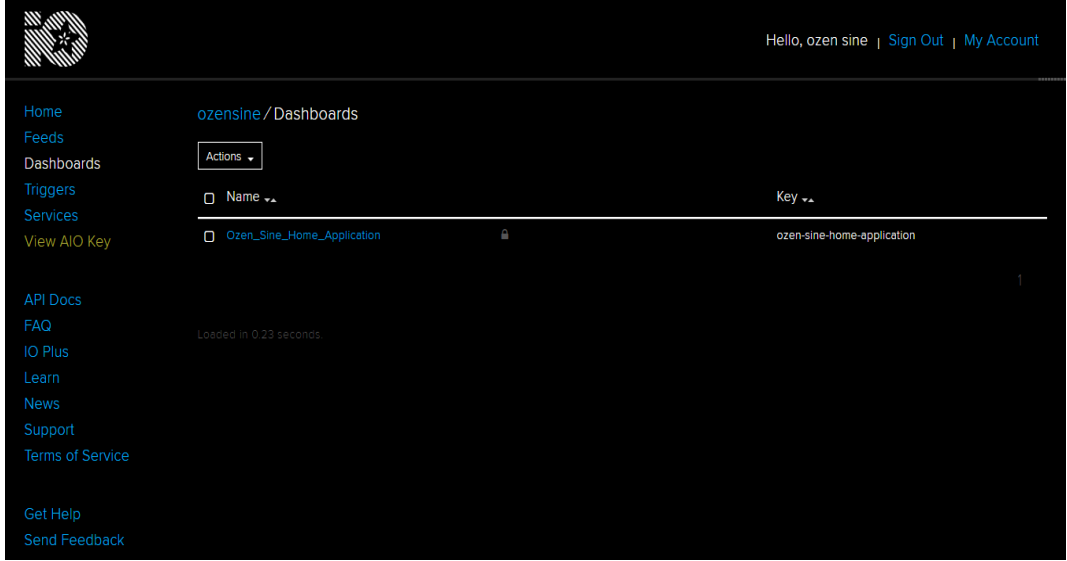
Yapılabilecekler arasında bir e-postaya yanıt verilmesi, trend tweet'lerin görüntülenmesi veya ev yolundayken evin ışıkları açabilmesi yer almaktadır. Sensörlerin yüzlerce web servisine bağlanabilmesi için geniş yelpazedeki yazılım ürünlerini birbirine entegre edilmesini sağlayan web tabanlı uygulamalar olan IFTTT ve Zapier ile entegrasyonu yapılmıştır.

3.2.3.1.1. Adafruit IO Sistemi Kontrol Arayüzünün Tasarımı

Akıllı ev otomasyonu MQTT protokolüyle internet üzerinden kontrol edilebilmesi için Adafruit IO kullanılarak bir IoT düzeni hazırlanmıştır. Adafruit IO sisteminin kontrol arayüzünü tasarlanması için ilk olarak Adafruit IO'ya kayıt olunması gerekmektedir. Daha sonra Adafruit anasayfasından gerekli işlemlerin tamamlanarak kayıt olunması ve sisteme giriş yapılması gerekmektedir.

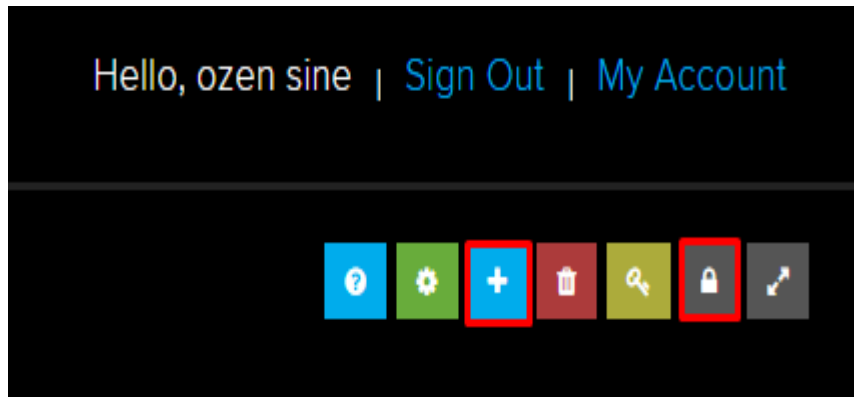
Giriş yapıldıktan sonra "Dashboard" tasarım arayüz ekranı ile kontrol paneli istenildiği gibi özelleştirebilmektedir.

İlk olarak sağ tarafta bulunan "**Create Dashboard**" tıklanarak yeni bir proje başlatılabilmekte ve projeye isim verilebilmektedir. Çalışmadaki uygulama "**Ozen_Smart_Home_Application**" olarak kaydedilmiştir.



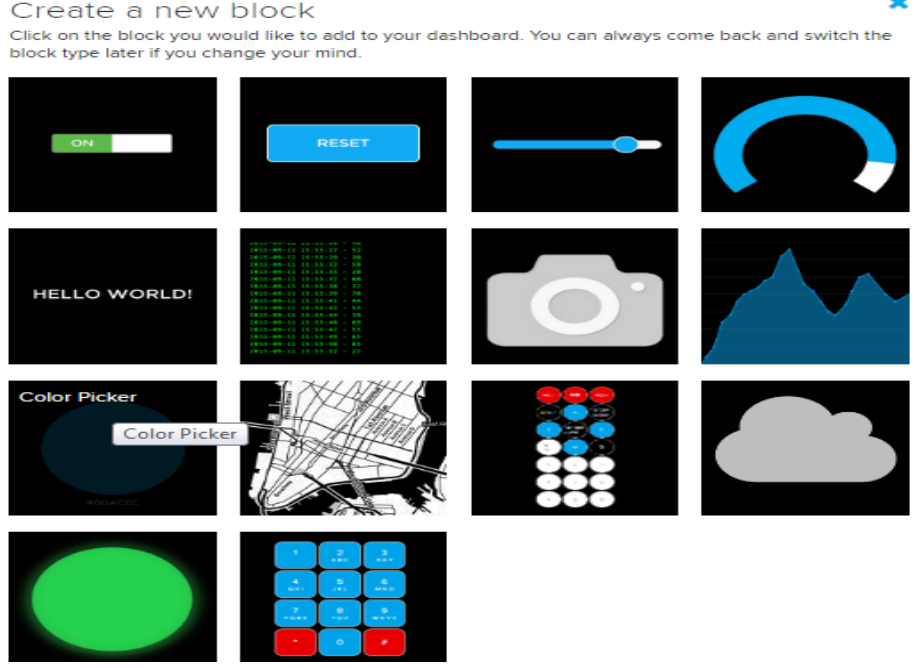
Şekil 3.49. Adafruit kontrol arayüzü tanıtımı 1

Projenin adı kaydedildikten sonra yukarıdaki gibi bir sayfa ile karşılaşılmaktadır. Böylece IO Paneli oluşturularak kullanıcının kendi isteğine göre özelleştirebilmesi sağlanmıştır. Kontrol panelinin dışarıdan erişilebilir duruma getirilebilmesi için Şekil 3.50'de sağ tarafta bulunan "**Kilit**" simgesine tıklanarak, onaylanması gerekmektedir.



Şekil 3.50. Adafruit kontrol arayüzü tanıtımı 2

Kontrol paneli ekranının sağ tarafında bulunan "**artı**" simgesine tıklanarak gerekli bütün araçlar eklenebilmektedir. Ev otomasyon sisteminde kontrol edilmesi istenen her bir parametre için buradan araç eklemesi yapılmıştır. Şekil 3.51'deki ekran görüntüsünde bütün araçlar gösterilmiştir. Standart veya daha ileri projeler için yeterli araç bulunmaktadır.



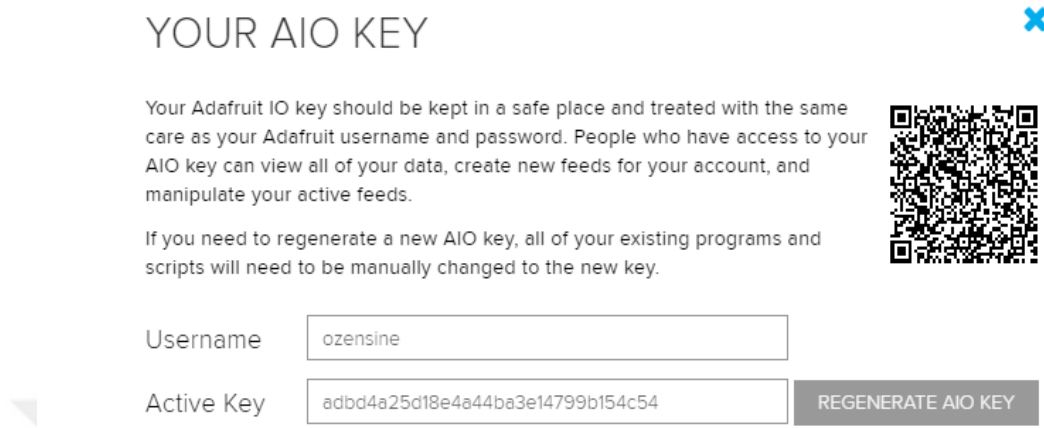
Şekil 3.51. Adafuit kontrol arayüzü tanıtımı 3

Araç kutusundan ev otomasyon sisteminde kullanacağımız açma-kapama ve kaydırma butonları ile kalibre ekranı gibi gerekli tüm araçlar seçilmiştir. Seçilen araçlar kullanıma hazır halde IoT paneline kopyalandıktan sonra Adafuit kontrol paneli Şekil 3.52’ de gösterildiği gibi son halini almaktadır. Böylece akıllı ev otomasyon sisteminin internet üzerinden kontrol edilebilmesi için Adafuit IO kontrol arayüzü tasarlanmıştır.



Şekil 3.52. Adafuit kontrol arayüzü tanıtımı 4

Adafruit IO ile oluşturulan panele, sadece kullanıcı tarafından erişilebilmesi için Şekil 3.53' te görülen ekranın sağ tarafındaki "**anahtar**" simgesine tıklanarak bir AIO anahtarı alınabilmektedir.



Şekil 3.53. Adafruit kontrol arayüzü tanıtımı 5

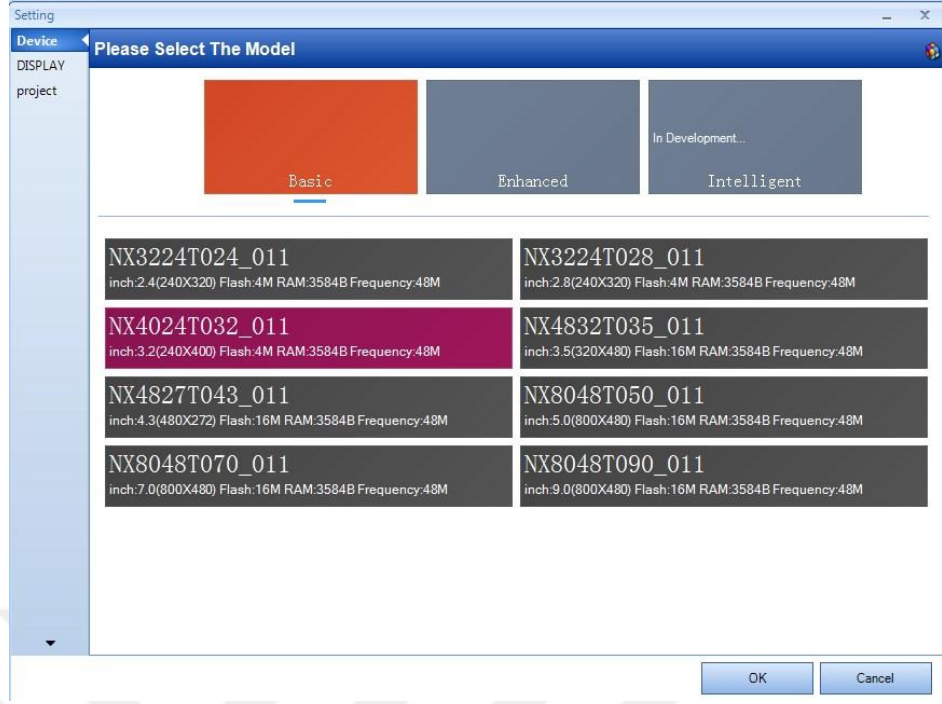
3.2.3.2. Nextion Dokunmatik Ekran Kontrol Arayüzü

Nextion dokunmatik ekranı, akıllı ev otomasyon uygulamasında ana kontrol ünitesi olarak kullanılmıştır. Bu sayede kullanıcıya internet erişiminin olmadığı durumlarda manuel olarak ev ortamından sisteme sürekli erişebilmesi açısından büyük avantaj sağlamaktadır.

Ekranın tasarlanması için Nextion Editör programı kullanılmış ve ayrıntılı olarak "Kullanılan Programlar" bölümünde anlatılmıştır. Bu başlık altında dokunmatik ekran kontrol arayüzünün nasıl tasarlandığı üzerinde durulmuştur.

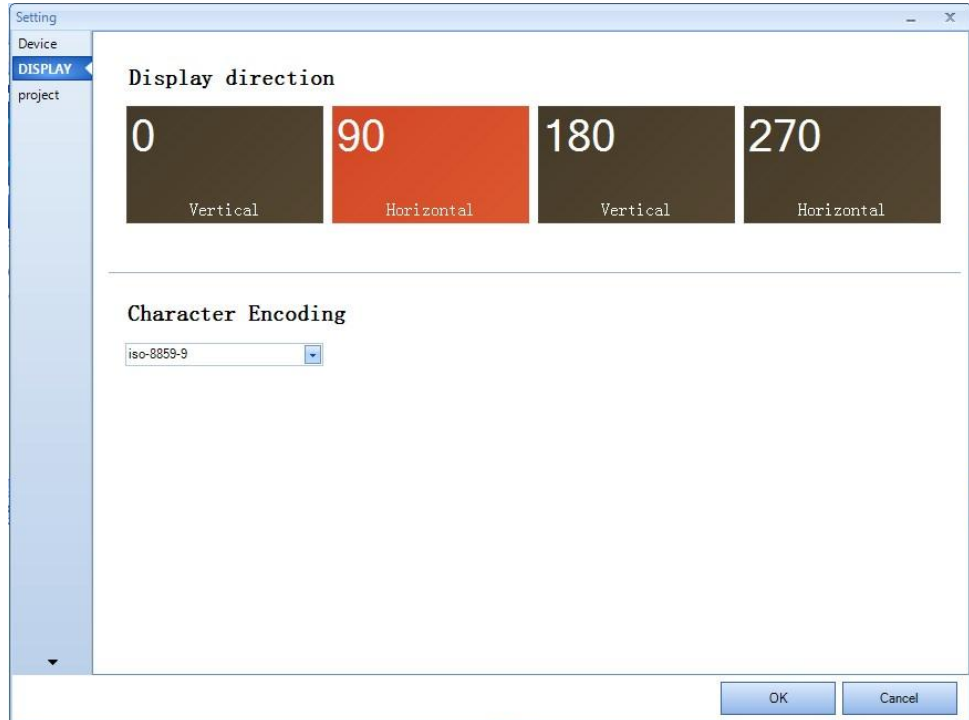
3.2.3.2.1. Nextion Dokunmatik Ekran Kontrol Arayüzünün Tasarımı

İlk olarak Nextion Editör'de yeni bir proje başlatabilmek için Nextion modelinin seçilmesi gerekmektedir. Çalışmada 3,2 inç ekran kullanıldığı için **NX4024K032_011 modeli** seçilmiştir. Kullandığımız model "**basic**" türünde olduğu için ekran türü değiştirilmemiştir(Şekil 3.54).



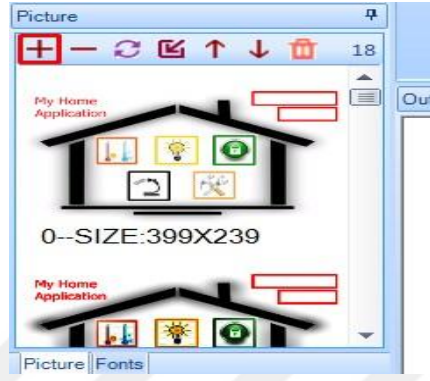
Şekil 3.54.Nextion Editör kontrol arayüzü tanıtımı 1

Daha sonra sol taraftaki “**Display**” sekmesinden ekranın nasıl ve hangi yönde kullanılacağı seçilmektedir (Şekil 3.55). Ekran yatay kullanılacağı için “**90**” seçeneği onaylandıktan sonra programın tasarım ekranına geçilmektedir.



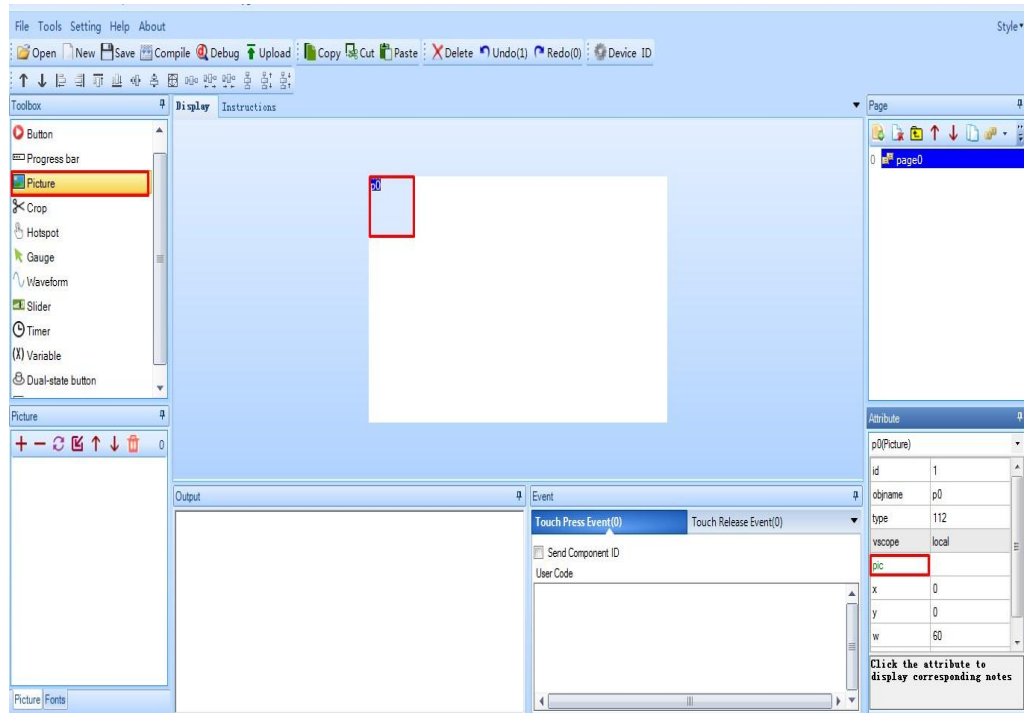
Şekil 3.55.Nextion Editör kontrol arayüzü tanıtımı 2

Tasarım ekranının sol köşesinde bulunan **“Picture”** kısmından **“+”** simgesine tıklanarak resim eklenebilmektedir. Eklenen resimlerin boyutlarına dikkat edilmesi gerekmektedir. Ekran arka duvar kâğıdı olarak ekleyeceğimiz resimlerin boyutları, ekran boyutlarıyla birebir aynı olmalıdır. Kullanılan ekran 240x400 olduğu için eklenen arka duvar resimleri de bu boyuta göre düzenlenmiştir.



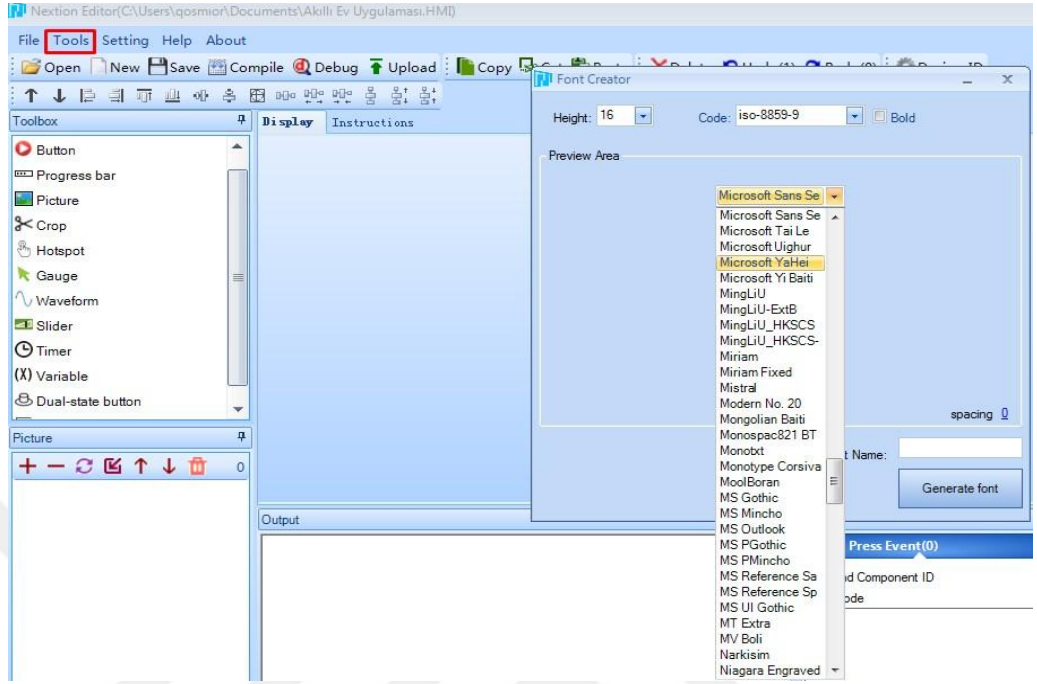
Şekil 3.56.Nextion Editör kontrol arayüzü tanıtımı 3

Resimler, program arayüzüne eklendikten sonra tasarım ekranının sol tarafındaki **“Toolbox”** kısmından Picture’a tıklanarak ekran üzerine **“p0”** bağlayıcı bileşen olarak atanmaktadır (Şekil 3.57). Daha sonra ekranın sağ tarafındaki **“Attribute”** kısmından **“pic”** değişkenine tıklanarak eklenen resmin bu değişkene ataması yapılmaktadır.



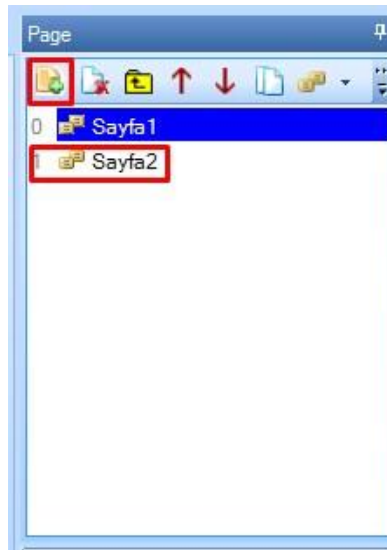
Şekil 3.57.Nextion Editör kontrol arayüzü tanıtımı 4

Tasarım ekranına font eklenebilmesi için menüdeki **Tools > Font Creator** adımları izlenmesi gerekmektedir(Şekil 3.58).



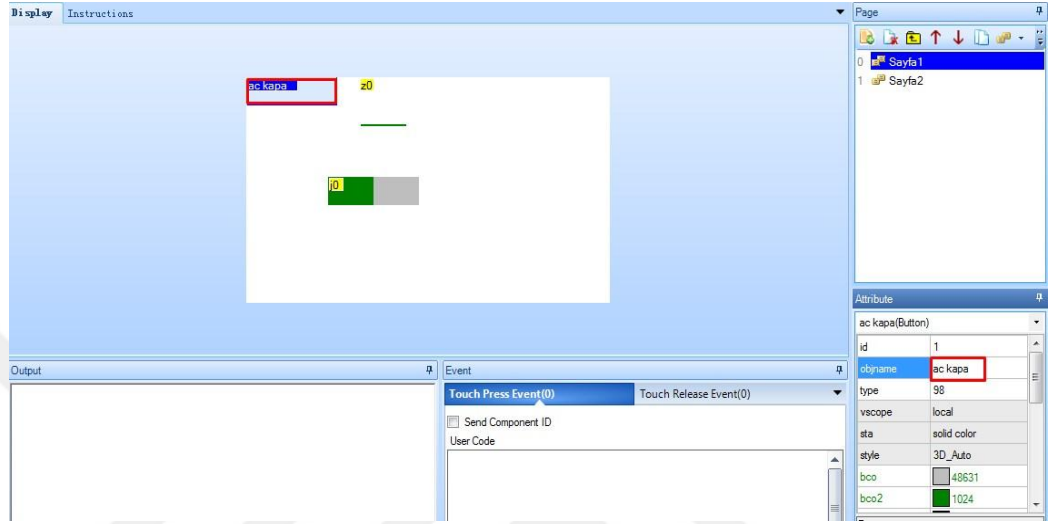
Şekil 3.58.Nextion Editör kontrol arayüzü tanıtımı 5

Ekran dokunulduğunda gerçekleşmesi istenen her senaryo için tasarım ekranına sayfa eklemesi gerçekleştirilmektedir. Bunun için sağ taraftaki **“Page”** kısmındaki **“+ işaretli sayfa”** simgesine tıklanarak ekrana sayfa eklenebilmektedir (Şekil 3.59).



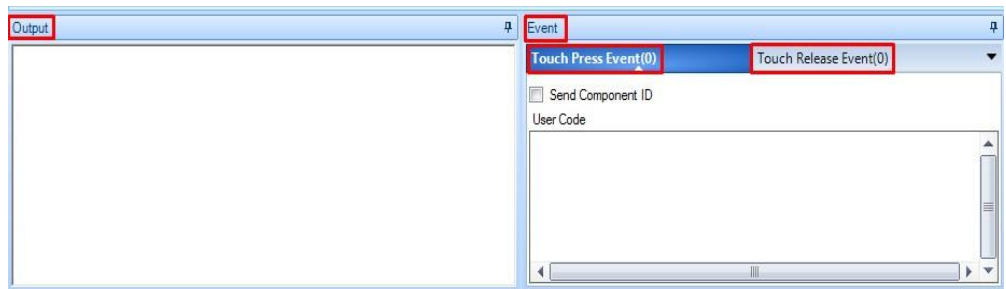
Şekil 3.59.Nextion Editör kontrol arayüzü tanıtımı 6

Kullanabilmek için tasarım ekranına buton, slider, label ve progress bar vs. gibi nesnelere sol taraftaki Toolbox'dan eklenebilmektedir (Şekil 3.58). Tüm nesnelere adlarının yazıldığı sağ taraftaki Attribute kısmında “objname” değişkeni bulunmaktadır. Nesne adları kod yazımında önemli olduğu için herhangi bir problemle karşılaşılmasını için uygun isimlerin verilmelidir. Ayrıca her nesne bulunduğu sayfada kendine özgü bir kimlik numarasına sahiptir.



Şekil 3.60.Nextion Editör kontrol arayüzü tanıtımı 7

Şekil 3.61'de görülen tasarım ekranının alt kısmında bulunan pencereler (**Output**, **Event**), sayfaları gösteren ve senaryolar gerçekleşirken kullanıcının kodları yazabileceği alanlardır. Ama ekranın çalışması için kod yazmak zorunlu olmayıp, gerektiğinde kullanıcı tarafından ekleme yapılabilmektedir. **Event** penceresi buton gibi iki şartı sağlayan nesnelere kullanılıyorsa önem arz etmektedir. Örneğin “**Touch Press Event**” sekmesi butona basıldığı anda gerçekleşecek senaryoları temsil etmekte ve bu andaki kodları göstermektedir. “**Touch Release Event**” sekmesi ise butonun bırakıldığı andaki gerçekleşecek senaryoları temsil etmekte ve bu andaki kodları göstermektedir.

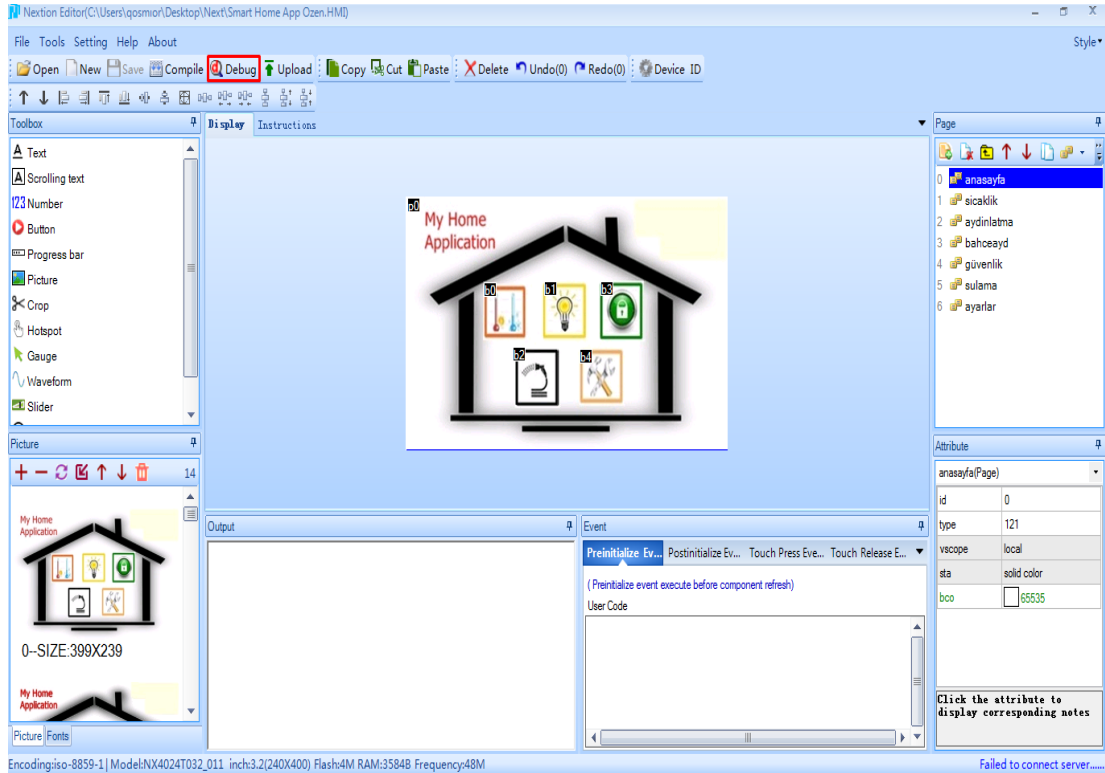


Şekil 3.61.Nextion Editör kontrol arayüzü tanıtımı 8

Arka plan resimleri üzerinde basılmak istenen kısımlara butonlar tanımlanması gerekmekte ve butonlara basıldığı zaman bir diğer sayfaya yollanarak açma kapama olgusu yaratılmaktadır. Butona basıldığı zaman diğer sayfaya yönlendirme işlemi butonun **Touch Release Event** bölümüne yazılan “page Sayfa No” komutu ile gerçekleştirilmektedir. Örneğin 0 numaralı sayfadan 1 numaralı sayfaya geçilebilmesi için “page 1” komutu kullanılmaktadır.

Tasarım ekranına nesne eklemesi, örneğin buton eklemesi şu şekilde yapılmaktadır: Programda butonlar kare şeklinde **default** olarak yer almaktadır. Nextion Editör programında nesnelere tasarımcının isteğine göre estetik bir görünüm kazandırılabilir. Butona basılmadan önce ve basıldıktan sonra olmak üzere iki farklı arka plan resmi seçilmektedir. **Attribute** kısmından basılmadan önceki resim “pic0” basıldıktan sonraki resim “pic1” değişkenine atanmaktadır. Bu sayede butona basıldığında görsel bir değişim elde edilmekte ve butona estetik bir görünüm kazandırılmaktadır. Gerekli olan bütün nesnelere aynı adımlar tekrarlanarak tasarım ekranına eklenebilmektedir.

Tamamlanan tasarım ekranının test edilebilmesi için “**debug**” üzerine tıklanarak ekran simüle edilmektedir. Şekil 3.62 ve Şekil 3.63’te dokunmatik ekranın Nextion Editör programındaki nihai arayüz tasarımları gösterilmiştir.



Şekil 3.62. Nextion Editör nihai arayüz tasarımı 1

My Home
Application



Salon Sıcaklığı : C

Manuel Kontrol

Oto Kontrol

İstenilen Sıcaklık : 18 C

Nem Oranı : % CO2 Miktarı :

SALON IŞIĞI OFF

Ayd. Sev. : %000

Bahçe Aydınlatması

Bahçe Aydınlatması

Manuel : OFF

Otomatik : OFF

Sağlık-Güvenlik

Nabız değeri: (dakikada) OFF Buzzer

Hareket algılama sensörü: OFF (Hırsız alarmı)

Yangın sensörü: OFF

Bahçe Sulaması

Manuel : OFF

Otomatik : Nem Oranı 000%

Mevsimsel Sıcaklık Takibi

Otomatik Mod: OFF

Manisa ili sıcaklık değeri: C

Durum:

Oda Sıcaklığı: C

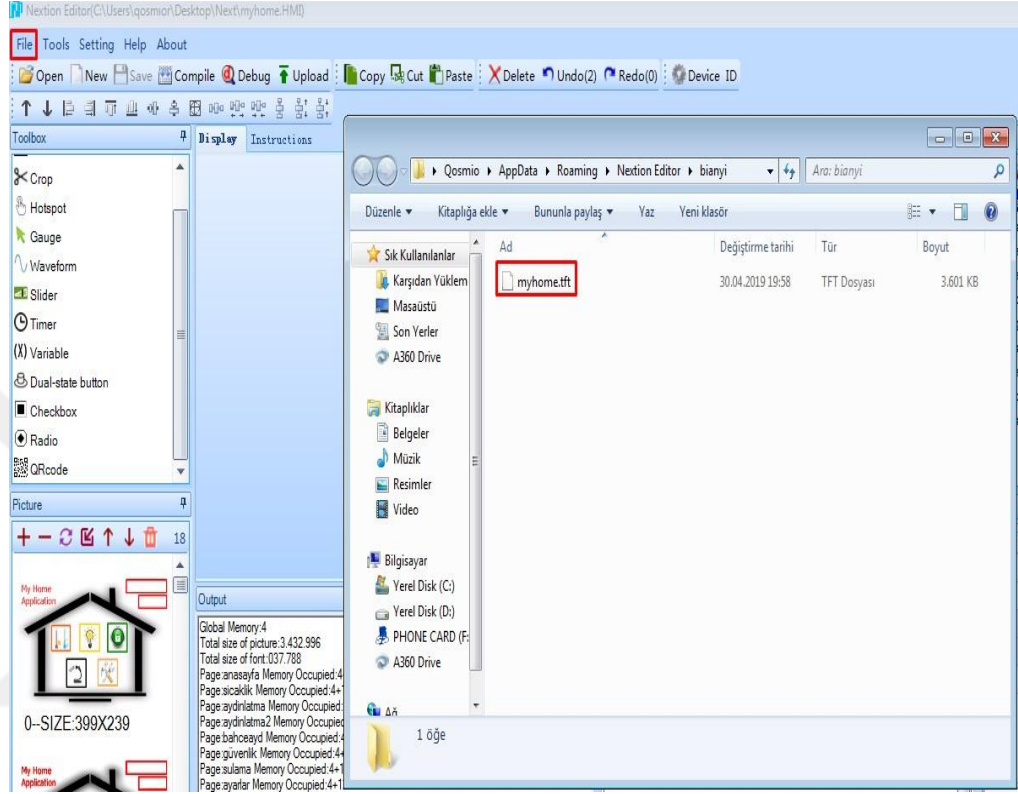
Ayarlar

Ekran Parlaklığı : 000 %

WiFi Sensör Haberleşmesi : ON

Şekil 3.63. NextionEditör nihai arayüz tasarımları 2

Son olarak yapılan ekran tasarımının Nextion ekranına yüklenmesi, ya SD aktarımı ile ya da UART bağlantısıyla gerçekleştirilebilmektedir. Mikro SD kart ile yüklenebilmesi için **File > Open Build Filder** seçildikten sonra “.tft” dosyası oluşturulmaktadır (Şekil 3.64).



Şekil 3.64. Nextion Editör kontrol arayüzü tanıtımı 9

İlk olarak oluşturulan dosya, mikro SD karta aktarıldıktan sonra Nextion ekranın arka tarafındaki SD kart yuvasına yerleştirilmektedir. Daha sonra ekran, power adaptörü ile beslenmekte ve mikro SD içeriği ekrana yüklenmektedir. Yükleme tamamlandıktan sonra ekranda “Update Succeeded!” yazısı görünmektedir. Dikkat edilmesi gereken husus oluşturulan “.tft” dosyası SD karta yüklenmeden önce kartın FAT32 olarak formatlanmış olması gerekmektedir.

İkinci olarak UART yoluyla yapılan tasarım, dokunmatik ekrana aktarabilmesi için yardımcı bir RX-TX bağlantılarının yapılabileceği elektronik karta ihtiyaç vardır (Arduinio vs. gibi). Ekran ile Arduinio arasındaki bağlantı şekli, Ground pinleri birbirleriyle doğrudan RX-TX bağlantıları ise birbirlerine ters gelecek şekilde gerçekleştirilmektedir. Daha sonra Nextion Editör programı üzerinden “Upload” a tıklandıktan sonra tasarım, dokunmatik ekrana yüklenmektedir.

3.2.4. Kullanılan Uygulama Programlama Arayüzü(API)

Çalışmada Manisa ilinin mevsimsel sıcaklık değerlerini internet ortamından çekebilmek için Weather API'sı olan "Openweather map" sitesi kullanılmıştır.

API (Application Programming Interface), bir uygulamaya ait yeteneklerin, başka bir uygulamada da kullanılabilmesi için, yeteneklerini paylaşan uygulamanın sağladığı arayüzdür.

3.2.4.1. Openweather API

Openweather, 2014 yılında veri ve uydu görüntüsü işleme alanlarında çalışan bir grup mühendis ve büyük veri işleme uzmanları tarafından kurulan küçük bir IT (Information Technology) şirkettir.

Hava durumu verileri, uydu görüntüleri ve diğer çevresel veriler hakkında geniş veri tabanı ağını kullanıcılara çok basit ve hızlı bir şekilde erişime sunarak kullanabilmelerini sağlamaktadır.

Openweather API'sını bireysel ve küçük işletme firmalarında dâhil olduğu 1,6 milyon geliştirici kullanmaktadır. Bugün itibariyle yaklaşık 0,5 PB (petabayt) kullanılabilir geçmişe dönük, güncel ve tahmini hava durumu ve uydu görüntüsü olmak üzere geniş veri tabanını kullanıcılara sunmaktadır.

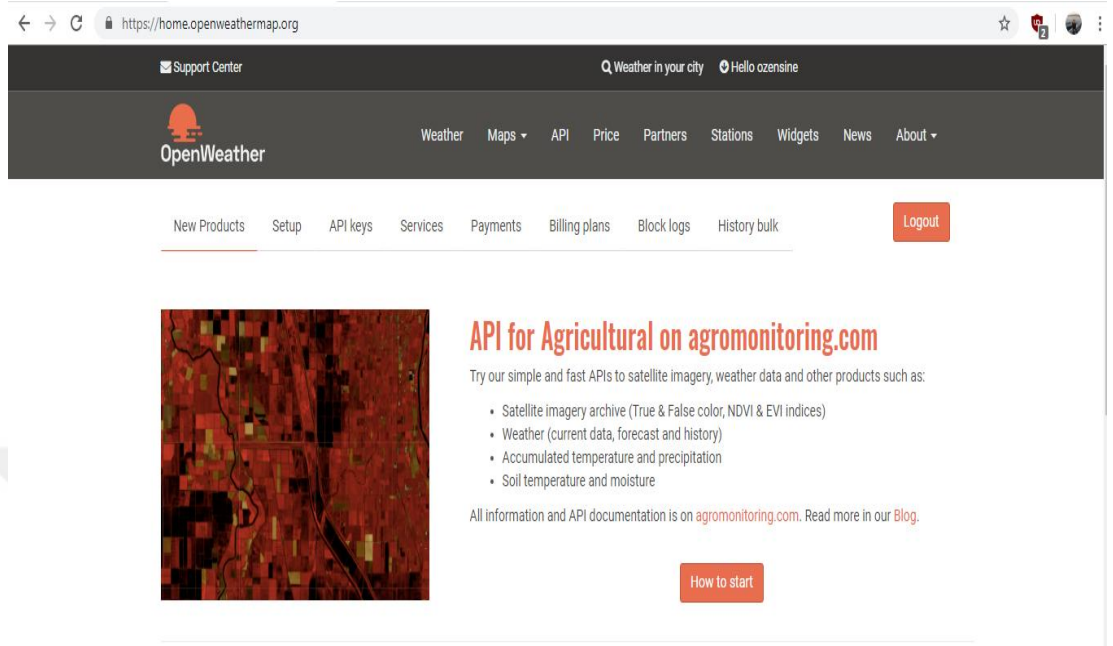
Tasarlanan ev maketindeki deneme odasının sıcaklık kontrolü ve bahçe sulaması, anlık mevsimsel sıcaklıklara göre ayarlanabilmektedir. Sıcaklık verilerinin internet ortamından çekilebilmesi için Openweather API veri tabanı kullanılmıştır. Mikrokontrolere sıcaklık verilerinin iletimi NodeMCU ESP8266 modülü sayesinde internet erişimiyle sağlanmaktadır.

3.2.4.1.1. Openweather Map İnternet Sitesi Tanıtımı

Çalışmada ESP8266 denetleyicisi kullanılarak OpenWeather map sitesinden hava durumu bilgileri çekilip ev ortamındaki Nextion dokunmatik ekranına aktarılması sağlanmaktadır. Ekran üzerinden sıcaklık takibi yapılabilmekte ve mevsimsel sıcaklık değerinin 23 °C' nin altına düşmesi veya üstüne çıkması durumlarında ısıtma ve soğutma sisteminleri otomatik olarak devreye girerek ev ortamı sıcaklık değerinde kararlılık sağlanmaktadır.

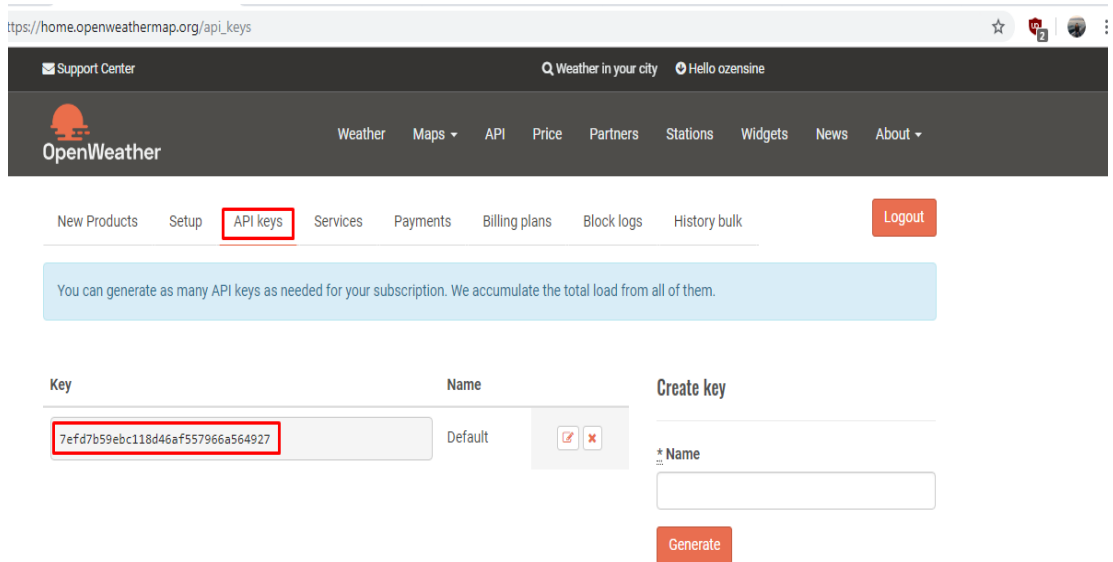
İlk olarak sistemdeki sıcaklık verilerine erişebilmek için Openweather map'in kendi internet sayfası üzerinden sisteme kayıt olunması gerekmektedir. Hesap

oluşturulduktan sonra, Şekil 3.65’de gösterilen birkaç sekme içeren bir gösterge tablosu sunulmaktadır.



Şekil 3.65. Openweather API başlangıç ekranı

Sekmelerden API Anahtarları sekmesi seçildikten sonra sistem kullanıcıya özel tanımlanmış bir kod üretmektedir. Bu kod, siteden bilgi alınabilmesi için gereken bir anahtardır. Bu anahtar yazılan programda kullanılır.(Şekil 3.66).



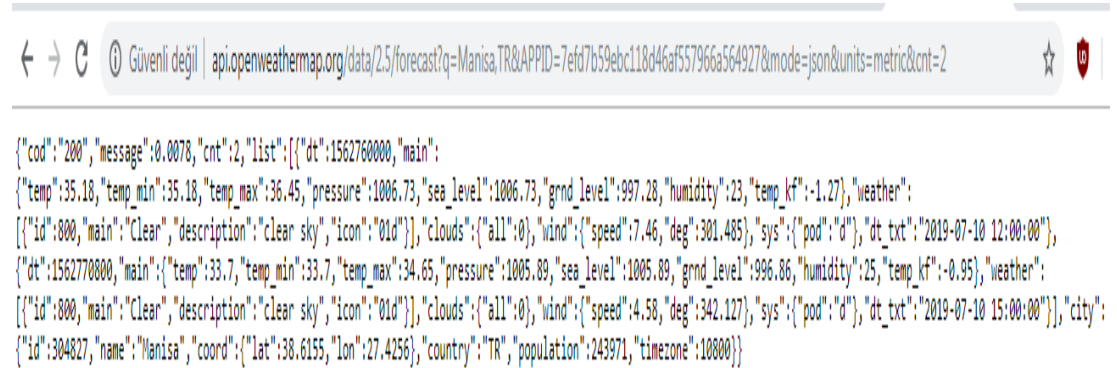
Şekil 3.66. Openweather API anahtar kod ekranı

İstenilen konumdaki hava durumu bilgilerini alabilmek için, konum bilgileri ve kullanıcıya özel tanımlanmış API anahtarı süslü parantez içindeki kısımlara girildikten sonra oluşturulan URL'ler (Uniform Resource Loader) Tablo 3.2' de gösterilmektedir.

Tablo 3.2. Openweather API'ya erişim için oluşturulan linkler

Durum	Oluşturulan API Linki
Konum Bilgileri Girilmeden Önce	<code>http://api.openweathermap.org/data/2.5/forecast?q={your city}, {your country code}&APPID={your API Key}&mode=json&units=metric&cnt=2</code>
Konum Bilgileri Girildikten Sonra	<code>http://api.openweathermap.org/data/2.5/forecast?q=Manisa,TR &APPID=7efd7b59ebc118d46af557966a564927&mode=json&units=metric&cnt=2</code>

Oluşturulan URL internet tarayıcısının link kısmına yapıştırıldığında Manisa iline ait yerel hava durumu tahminleriyle alakalı veri bulutuna erişilebildiği görülmektedir (Şekil 3.67).



Şekil 3.67. Openweather API'dan kullanıcıya gönderilen veri bulutu

Akıllı ev ortamına; Openweather API'sından önce MSP432 mikrokontrolöre daha sonra da dokunmatik ekrana Manisa iline ait mevsimsel sıcaklık değerleri NodeMCU ESP8266 modülü vasıtasıyla aktarılabilir. Mevsimsel sıcaklık veri akış trafiği yazılan programla düzenlenmektedir. Yazılan algoritmalar "EKLER" bölümünde gösterilmiştir.

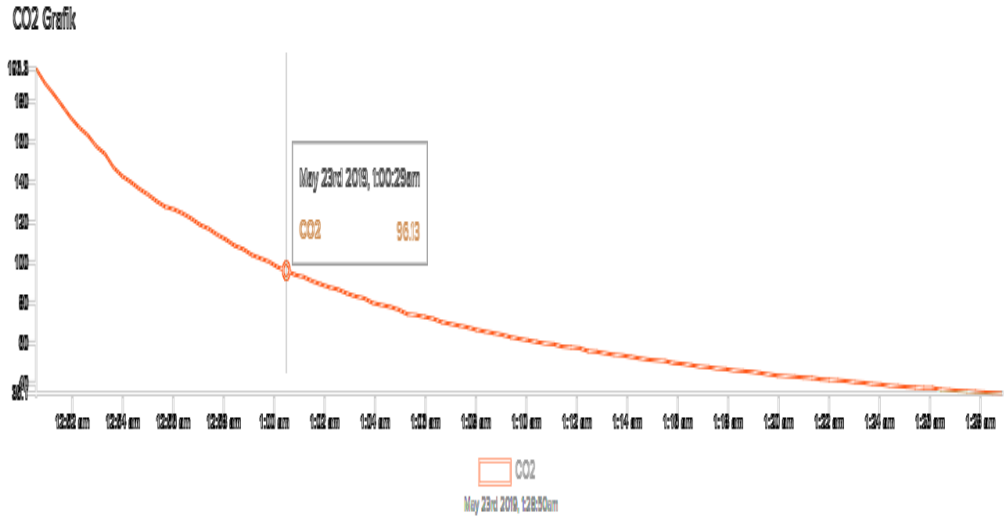
4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Akıllı ev otomasyonu çalışması kapsamında sistemin uygulanabilirliği ev maketi ortamında araştırılmıştır. MQTT Protokolü sayesinde ev ortamındaki verilerin düzenli olarak Adafruit IO' da depolanması sağlanmıştır. Şekil 4.1'de sistemden anlık olarak çekilen veriler görülebilmektedir. Bu kapsamda sistem geçmişe dönük ve sürekli güncel bir şekilde rahatlıkla takip edilebilmektedir.

Created at	Value	Location
2019/05/23 1:28:50am	35.12	x
2019/05/23 1:28:29am	35.39	x
2019/05/23 1:28:09am	35.65	x
2019/05/23 1:27:48am	35.92	x
2019/05/23 1:27:28am	36.19	x
2019/05/23 1:27:07am	36.46	x
2019/05/23 1:26:47am	36.73	x
2019/05/23 1:26:26am	37.01	x
2019/05/23 1:26:06am	37.83	x
2019/05/23 1:25:45am	37.83	x
2019/05/23 1:25:25am	38.11	x
2019/05/23 1:25:05am	38.39	x
2019/05/23 1:24:44am	38.68	x

Şekil 4.1.Ev ortamından Adafruit IO'ya veri gönderimi

Aynı zamanda ev ortamından gelen her bir parametreye ait veri Adafruit IO'da grafiksel olarak Şekil 4.2'deki gibi çizdirilebilmektedir. Örneğin evdeki sıcaklık değişimi an ve an rahatlıkla gözlenebilmekte ve analizi yapılabilmektedir. Grafikler istenilen tarihlerde ve periyotlarda oluşturulabilmektedir. Böylece kullanıcı sistem kontrolünde aktif bir role bürünebilmekte ve kullanım konforu açısından avantaj sağlanmaktadır.



Şekil 4.2.Ev ortamındaki CO₂ miktarı değişim grafiği

Birden fazla parametreye ait veri değişimleri aynı ekran üzerinde Şekil 4.3'deki gibi eş zamanlı olarak takip edilebilmektedir. Grafiklerin eşik değerleri kullanıcının isteğine veya gönderilen verinin minimum-maksimum değerine göre otomatik olarak ayarlanabilmektedir.



Şekil 4.3.Sıcaklık ve nem değişiminin aynı ekran üzerinde görüntülenmesi

Openweather API vasıtasıyla internet ortamından sisteme çekilen anlık mevsimsel sıcaklıklarının ve ev ortamında simule edilmiş yatağa bağımlı bir bireyden alınan anlık nabız değerlerinin Energia debug ekranında nasıl takip edilebildiği Şekil 4.4 ve Şekil 4.5’de gösterilmiştir.

```
SICAKLIK 23 IDEAL - ISITICI:OFF, HAVALANDIRMA:OFF, SULAMA POMPA:OFF
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 23
, Manisa ili Sıcaklık Degeri: 28
, Manisa ili Sıcaklık Degeri: 30
SICAKLIK 23'un USTUNDE - ISITICI:OFF, HAVALANDIRMA:ON, SULAMA POMPA:OFF
, Manisa ili Sıcaklık Degeri: 33
, Manisa ili Sıcaklık Degeri: 35
, Manisa ili Sıcaklık Degeri: 37
SICAKLIK 35'in USTUNDE - ISITICI:OFF, HAVALANDIRMA:ON, SULAMA POMPA:ON
, Manisa ili Sıcaklık Degeri: 36
, Manisa ili Sıcaklık Degeri: 36
SICAKLIK 35'in ALTINDA - ISITICI:OFF, HAVALANDIRMA:ON, SULAMA POMPA:OFF
, Manisa ili Sıcaklık Degeri: 35
, Manisa ili Sıcaklık Degeri: 33
, Manisa ili Sıcaklık Degeri: 32
, Manisa ili Sıcaklık Degeri: 31
, Manisa ili Sıcaklık Degeri: 31
, Manisa ili Sıcaklık Degeri: 31
, Manisa ili Sıcaklık Degeri: 29
, Manisa ili Sıcaklık Degeri: 28
, Manisa ili Sıcaklık Degeri: 27
, Manisa ili Sıcaklık Degeri: 27
, Manisa ili Sıcaklık Degeri: 27
```

Şekil 4.4.Energia debug ekranında mevsimsel sıcaklık verilerinin takibi

```
COM5
Nabiz Degeri: 46
Nabiz Degeri: 39
Nabiz Degeri: 37
NABIZ 40'in ALTINDA - ALARM:ON - MAIL ATILDI
Nabiz Degeri: 38
NABIZ IDEAL
Nabiz Degeri: 45
Nabiz Degeri: 51
Nabiz Degeri: 59
Nabiz Degeri: 65
Nabiz Degeri: 68
Nabiz Degeri: 74
Nabiz Degeri: 76
Nabiz Degeri: 80
Nabiz Degeri: 83
Nabiz Degeri: 84
Nabiz Degeri: 86
Nabiz Degeri: 91
Nabiz Degeri: 94
Nabiz Degeri: 98
Nabiz Degeri: 100
Nabiz Degeri: 105
NABIZ 100'un USTUNDE - ALARM:ON - MAIL ATILDI
Nabiz Degeri: 110
Nabiz Degeri: 115
NABIZ IDEAL
Nabiz Degeri: 99
Nabiz Degeri: 95
Nabiz Degeri: 89
Nabiz Degeri: 84
Nabiz Degeri: 79
Nabiz Degeri: 75
Nabiz Degeri: 74
Nabiz Degeri: 74
Nabiz Degeri: 79
Nabiz Degeri: 83
Nabiz Degeri: 78
Nabiz Degeri: 84
Nabiz Degeri: 88
Nabiz Degeri: 94
Nabiz Degeri: 91
Nabiz Degeri: 83
```

Şekil 4.5.Energia debug ekranında nabız değeri verilerinin takibi

5. SONUÇ VE ÖNERİLER

Yapılan ev maketi uygulamasında çağın gerektirdiği teknolojik ve modern alt yapının kullanıcıya sağladığı imkânlar incelenmiş ve inşa edilen sistemin gerçek bir evde uygulanabilirliği kanıtlanmıştır. Ev maketinde oluşturulan deneysel odaya kontrol edilmesi planlanan her bir parametre için sensör yerleştirilmiş ve hedeflenen haberleşme protokolleri sorunsuzca gerçekleştirilmiştir. Kompakt bir yapıda oluşturulan akıllı ev kontrol sisteminde kullanıcının ev ortamına istediği zaman aktif olarak müdahale edebilmesi amaçlanmıştır.

Sistemin internet üzerinden kontrolünde internet sağlayıcısı firmanın indirme/yükleme hızları önem arz etmektedir. Bu hızların kararlılığı, web serveri üzerinden akıllı ev kontrol sistemine erişimde gecikmelere neden olabileceği ve sistemin verimini düşürebileceği tespit edilmiştir. Gelecek yıllarda internet alanındaki geniş bant performansını iyileştirecek gelişmeler ve çalışmalar neticesinde bu sorunun ortadan kalkacağı düşünülmektedir.

Daha önce yapılmış akıllı ev uygulamalarında sistemin verimini, kullanılan mikrokontrolörün (denetleyicinin) belirlediği tespit edilmiştir. O yüzden modern bir ev modelini uyguladığımız ev maketinde sistemin verimini maksimumda tutabilmek için Texas Instruments'ın piyasaya yeni sürmüş olduğu düşük güç tüketimi ve ARM Cortex 32 Bit mimari tabanlı performansı ile öne çıkan MSP432 mikrokontrolörü kullanılmıştır. Hem verimli veri işleme hem de gelişmiş düşük güç tüketimi konusunda büyük önem taşımaktadır.

Ev otomasyonu sisteminde ev ortamında bulunan sensörlerin mikrokontrolör ile kablosuz haberleşmesi hedeflenmiş ve başarılmıştır. Bunun için oldukça ekonomik ve kullanışlı ESP8266 Wifi Modülü kullanılmıştır. ESP8266 üzerinde dâhili anten bulunmakta ve bu sayede ortamdaki Wifi ağına rahatlıkla bağlanabilmekte, veri paketleri alıp gönderilebilmektedir. Sistemin kablosuz haberleşmesi sayesinde kablo masrafı olmadığı gibi kablo kalabalığından da ev ortamı arındırılmış olacak ve böylece kablo montajı için duvar kazısı, sabitlenmesi ve sıva, vb. gibi işlerde gerekmeyecektir.

Çalışmada, Nextion dokunmatik ekran, akıllı ev otomasyon uygulamasında ana kontrol ünitesi olarak kullanılmış ve bu sayede kullanıcıya internet erişiminin olmadığı durumlarda ev ortamından sisteme sürekli erişebilmesi açısından büyük

avantaj sağlanmıştır. Bu sayede sistemde herhangi bir sorun meydana geldiğinde kullanıcı tarafından rahatlıkla çözümlenmektedir.

Adafruit Kontrol Arayüzü sayesinde sistem her an internet üzerinden izlenebilmekte, sisteme müdahale edilebilmekte ve ev ortamından çekilen veriler kayıt altına alınabilmektedir. Kayıt altına alınan verilere istenildiği zaman ulaşılabilen ve kaydedilen verilerin periyodik olarak grafiksel istatistikleri izlenebilmektedir. Böylece sisteme ait verilere daha sonra rahatlıkla ulaşılabilir.

Akıllı ev sistemleri ile günümüzde kontrol edilmesi istenen parametreler kısmi olarak yapılabilmektedir. Çalışmada hem farkındalık yaratmak hem de geleceğe yönelik hiçbir dış faktör müdahalesi gerektirmeyen tam otomasyon sistemlerine örnek teşkil etmesi için ev ortamındaki sıcaklık ve bahçe sulama parametreleri anlık mevsimsel sıcaklıklara göre kontrol edilebilmesi sağlanmıştır. Ayrıca yine kullanıcı konforu düşünülerek bakıma muhtaç veya yatağa bağımlı bir bireyin herhangi bir sağlık sorunu ile karşılaşabilme senaryosu üzerinde durulmuş ve bu durumun en az kayıpla atlatabilmesi sağlanmıştır. Bu senaryo, bireyin nabız değerlerinin sürekli olarak takibinin yapılarak normal değerinin altına düşmesi veya üstüne çıkması durumunda acil sağlık ekibine ve bireyin yakınına haber verilmesiyle gerçekleştirilmiştir.

Akıllı Ev Otomasyon Uygulaması amaçlanan hedefte gerçekleştirilebilmiş, beklenen sonuçlara ulaşılabilmiş ve gerçek bir ev ortamı simule edilebilmiştir. Ancak yukarıda da değinildiği gibi çağın en ileri teknolojilerine sahip cihazlar kullanılmasına rağmen bazı noktalarda verim konusunda sıkıntılar yaşanmıştır. Teknolojinin ilerlemesiyle ve insan hayatını kolaylaştıracak yeni orjinal fikirlerin bulunmasıyla akıllı ev otomasyonları daha da iyileşecek ve mevcut sorunlara çözümler bulunacaktır. Kullanıcının isteklerini şahsi olarak girmesi yerine biyomedikal cihazlarla isteklerinin tespiti ve sistemin otomatik olarak bu taleplere cevap vermesi ilerleyen yıllarda yeni araştırma konusu olabilir. Örneğin kullanıcının vücut ısısının akıllı kol saatiyle ölçülüp ihtiyaç duyduğu ısıya göre bulunduğu oda sıcaklığının ayarlanması gibi. Günümüzde kısmi olarak yapılabilen akıllı ev otomasyonlarının gelecek yıllarda tam otomasyonlu sistemlere dönüşeceğine hiç şüphe yoktur.

KAYNAKLAR

- [1]Berardi, U. ve Amirhosein,G.,What is an intelligent building? Analysis of recent interpretations from an internationalperspective, Architectural Science Review, Ekim 2015, 59, 338-357.
- [2]Gedikpınar,M.veCAVAŞ,M.,Pic16F84 Mikrodenetleyici ile Akıllı Ev Otomasyonu, Otomasyon Dergisi, Şubat2005,153.
- [2]Barış Yüksekaya,A.Alper Kayalar,M.Bilgehan Tosun,M.Kaan Özcan and A.Ziya Alkar, Gsm,Internet and Speech Controlled Wireless Interactive, IEEE, 2006, 837-843.
- [3]Tosunoğlu,S.O., Akıllı Ev Sistemlerinde Merkezi Veri Toplama ve Cihaz Yönetimi, Gebze Yüksek Teknoloji Enstitüsü, Mühendislik ve Fen Bilimleri Enstitüsü, Gebze, 2009,49s.(Yüksek Lisans Tezi)
- [4]Sheikh Izzal Azid, Sushil Kumar, Analysis and Performance of a Low Cost SMS Based Home Security System, International Journal of Smart Home, Temmuz 2011, 5, 15-24.
- [5]Murat Eğrikavuk,Bina Otomasyon Sistemlerinde Yenilikler,Türk Tesisat Mühendisleri Derneği Eğitim Toplantıları,Mart 2004.
- [6]Khuswinder Gill,Shuang-Hua Yang,Fang Yao,and Xin Lu,A ZigBee-Based Home Automation System,IEEE Transactions on Consumer Electronics,55,May 2009, 430-433.
- [7]Çetin Kamil,Home Automation,Dokuz Eylül Üniversitesi,Graduate School Of Natural And Applied Sciences, İzmir, 2007, 12s.(Yüksek Lisans Tezi)
- [8]Wacks,K.P.,International Development of Home Automation Standards,Home and Building Automation Consultant,June 4,1992.
- [9]Üstkan S.,Ağ Standartları,<http://profkan.blogspot.com/>,February 27,2009, 21.05.2019.
- [10]Uğuz,S. ,KILIÇ, B.,Şişeci, M.,Akıllı Ev Otomasyonu Sistemlerinde ZigBee Tabanlı Ağ Uygulamaları,Mehmet Akif Ersoy Üniversitesi,Kasım 2013,7s.
- [11]Datasheet, MSP432P401R and MSP432P401M SimpleLink Mixed-Signal Microcontrollers, Texas Instruments, Slas826g, Mart 2015, Yeniden Düzenleme Eylül 2017, 198s.
- [12]Datasheet, ESP8266 NodeMCU WiFi Devkit User Manuel v1.2, Handson Technology, 2017, 22s.
- [13]Datasheet, Servomotor SG90, Tower Pro, 2014, Yeniden Düzenleme Haziran 2016, 1s.
- [14]Datasheet, Nextion 3,2" dokunmatik ekran, https://nextion.itead.cc/resources/datasheets/nx4024t032_011/, 21.05.2019.
- [15]Datasheet, DHT 11 Humidity and Temperature Sensor, D-Robotics, 2010, Yeniden Düzenleme Ocak 2015, 10s.
- [16]Datasheet, Flammable Gas Sensor(MQ-6) Manual v1.3, Zhengzhou Winsen ElectronicsTechnology Co. Ltd, 2014, 7s.
- [17]Datasheet, Air Quality Gas Sensor(MQ-135) Manual v1.4, Zhengzhou Winsen ElectronicsTechnology Co. Ltd, 2015, 7s.
- [18]Datasheet, DC Fans 12-24 V, Minebea Company, 2015, 1s.
- [19]Datasheet, YL-44 Buzzer, <https://www.openimpulse.com/blog/products-page/product-category/active-buzzer-module-and-cable/>, 21.05.2019.
- [20]Datasheet, Light Dependent Resistör, Sunroom Technologies, 3190, 2008, 4s.
- [21]Datasheet, Toprak Nem Sensörü, <http://www.elektrohobim.com/index.php/arduino/sensorler/toprak-nem-olcer/> 21.05.2019.

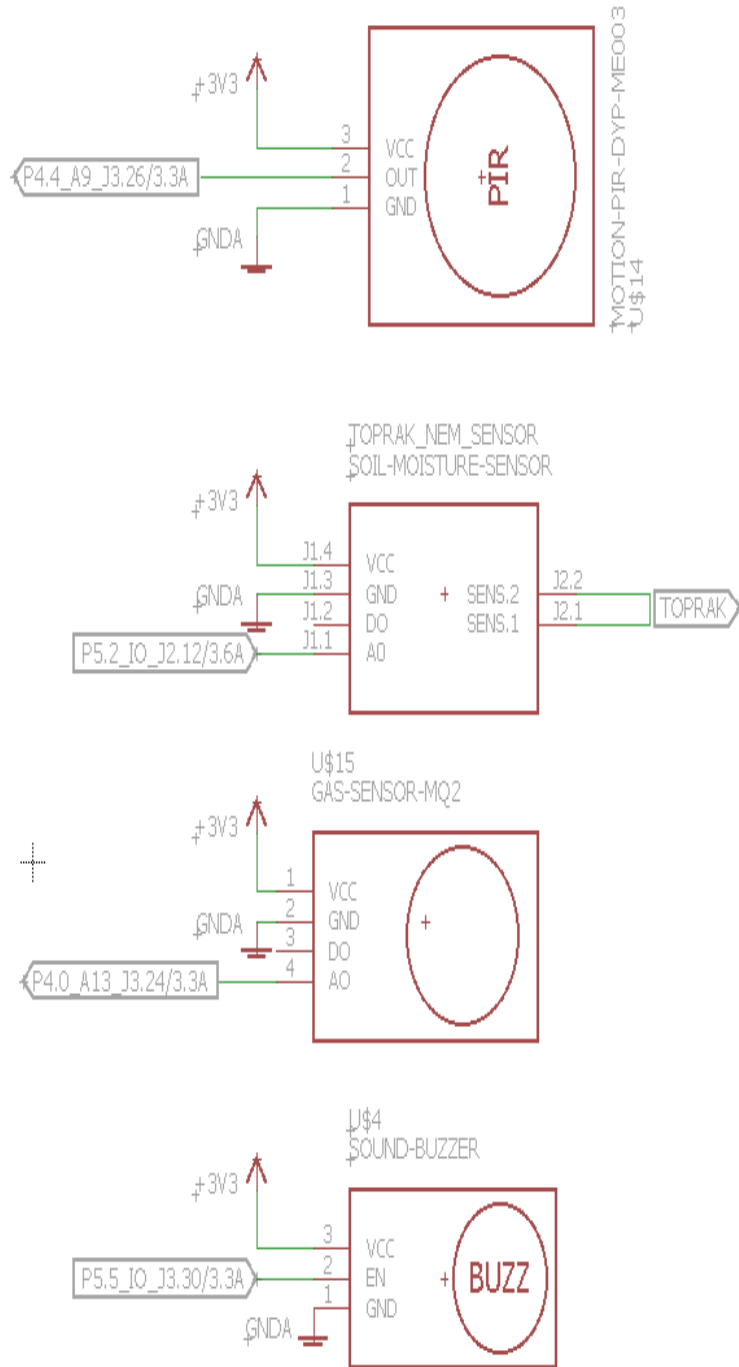
- [22]Datasheet, HC-SR501 Pır Motion Detector, MPJA, 2013, 3s.
- [23]Datasheet, 12V DC Water Pump with Brushless Motor DC40-1245, Brushless, 2011, 3s.
- [24]Datasheet, Thermoelectric Cooler, Hebei I.T. (Shanghai) Co. Ltd, 2004, Yeniden Düzenleme Mayıs 2017, 3s.
- [25]Datasheet, 1 Watt High Power Led, Multicomp, 2012, 5s.
- [26]Datasheet, 360 Watt Power Supply, Zhuhai Jinbo Kechuang Electronics Co. Ltd, 2017, 2s.



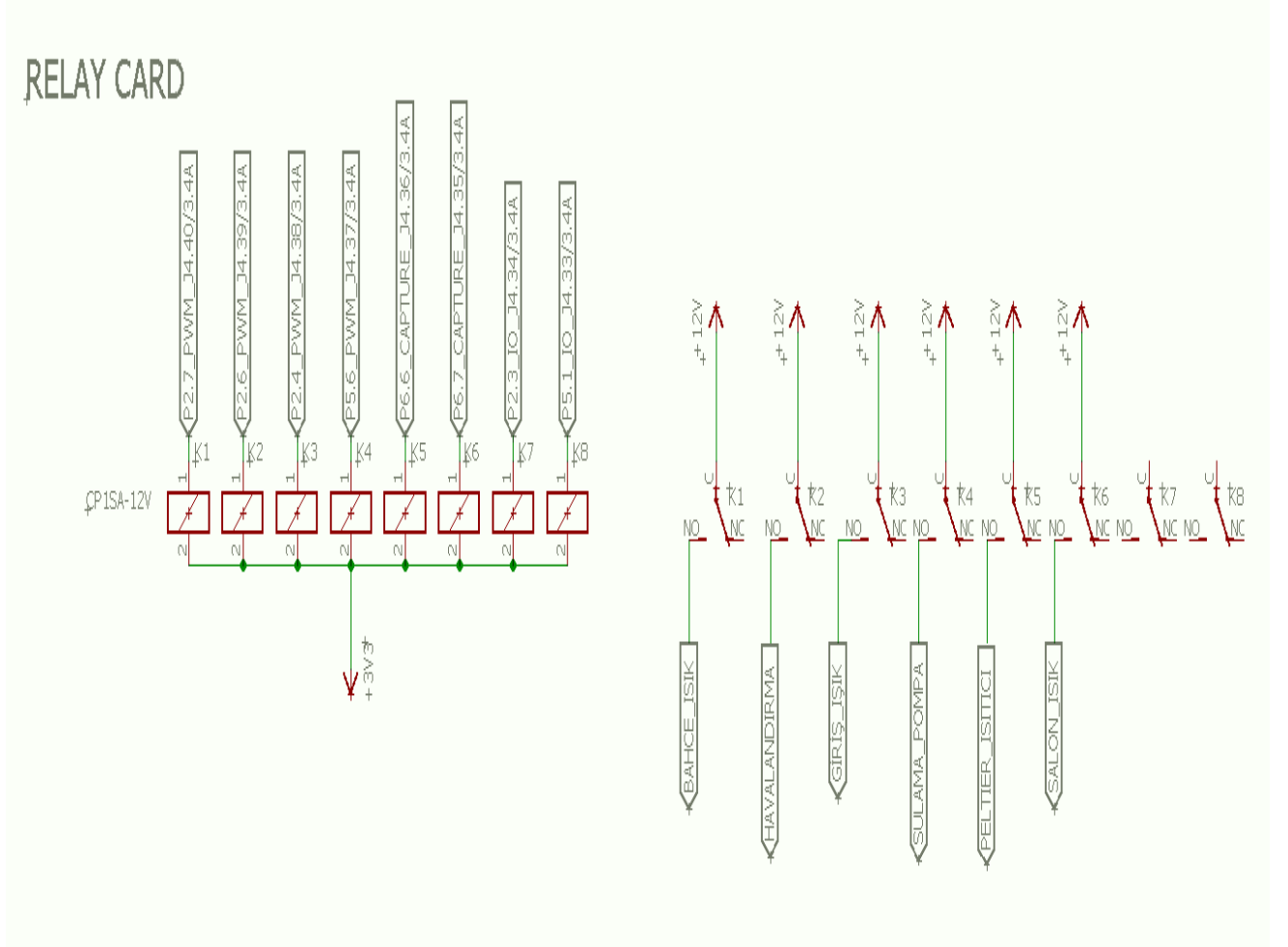
EKLER

EK A.(MSP432 ile çevresel birimler arasındaki bağlantı şemaları.)

MSP432 - ÇEVRESEL BİRİMLER



EK B.(MSP432'nin çevresel birimlere çıkış vermesini sağlayan röle kartının devre şeması.)

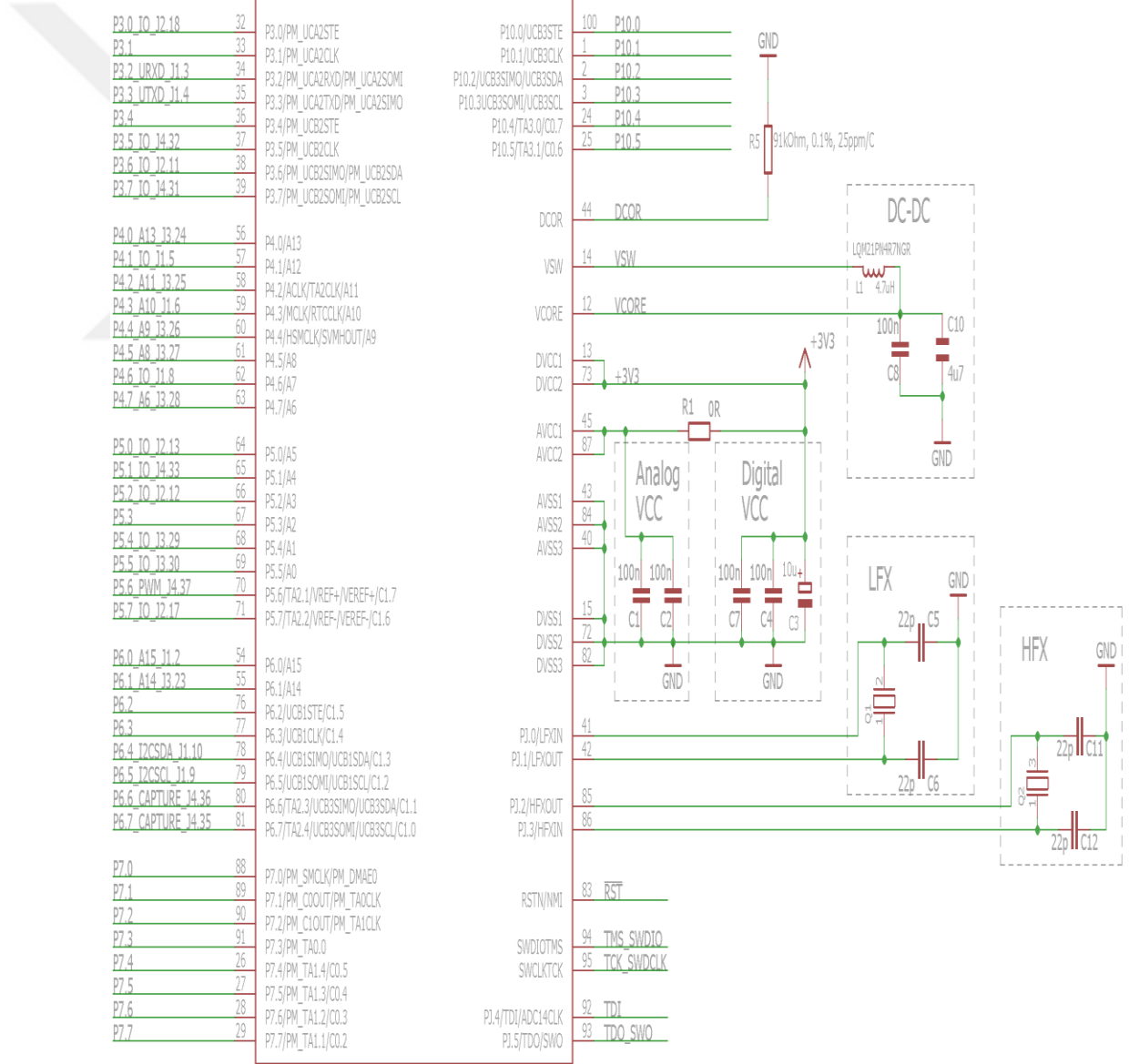


EK C.(MSP432 mikrokontrolörün pin diyagramı.)

MSP1

P1.0_LED1	4	P1.0/UCA0STE	30	P8.0	
P1.1_BUTTON1	5	P1.1/UCACLK	31	P8.1	
P1.2_BCLUART_RXD	6	P1.2/UCARXD/UCASOMI	46	P8.2	
P1.3_BCLUART_TXD	7	P1.3/UCATXD/UCASIMO	P8.2/TA3.2/A23	47	P8.3
P1.4_BUTTON2	8	P1.4/UCB0STE	P8.3/TA3CLK/A22	48	P8.4
P1.5_SPICLK_J1.7	9	P1.5/UCB0CLK	P8.4/A21	49	P8.5
P1.6_SPIMISO_J2.15	10	P1.6/UCB0SIMO/UCB0SDA	P8.5/A20	50	P8.6
P1.7_SPIMISO_J2.14	11	P1.7/UCB0SOMI/UCB0SCL	P8.6/A19	51	P8.7
			P8.7/A18		
P2.0_RGBLED_RED	16	P2.0/PM_UCA1STE	P9.0/A17	52	P9.0
P2.1_RGBLED_GREEN	17	P2.1/PM_UCA1CLK	P9.1/A16	53	P9.1
P2.2_RGBLED_BLUE	18	P2.2/PM_UCA1RXD/PM_UCA1SOMI	P9.2/TA3.3	74	P9.2
P2.3_IO_J4.34	19	P2.3/PM_UCA1TXD/PM_UCA1SIMO	P9.3/TA3.4	75	P9.3
P2.4_PWM_J4.38	20	P2.4/PM_TAO.1	P9.4/UCA3STE	96	P9.4
P2.5_PWM_J2.19	21	P2.5/PM_TAO.2	P9.5/UCA3CLK	97	P9.5
P2.6_PWM_J4.39	22	P2.6/PM_TAO.3	P9.6/UCB3RXD/UCB3SOMI	98	P9.6
P2.7_PWM_J4.40	23	P2.7/PM_TAO.4	P9.7/UCB3TXD/UCB3SIMO	99	P9.7
P3.0_IO_J2.18	32	P3.0/PM_UCA2STE	P10.0/UCB3STE	100	P10.0
P3.1	33	P3.1/PM_UCA2CLK	P10.1/UCB3CLK	1	P10.1
P3.2_URXD_J1.3	34	P3.2/PM_UCA2RXD/PM_UCA2SOMI	P10.2/UCB3SIMO/UCB3SDA	2	P10.2
P3.3_UTXD_J1.4	35	P3.3/PM_UCA2TXD/PM_UCA2SIMO	P10.3/UCB3SOMI/UCB3SCL	3	P10.3
P3.4	36	P3.4/PM_UCB2STE	P10.4/TA3.0/CO.7	24	P10.4
P3.5_IO_J4.32	37	P3.5/PM_UCB2CLK	P10.5/TA3.1/CO.6	25	P10.5
P3.6_IO_J2.11	38	P3.6/PM_UCB2SIMO/PM_UCB2SDA			
P3.7_IO_J4.31	39	P3.7/PM_UCB2SOMI/PM_UCB2SCL			
P4.0_A13_J3.24	56	P4.0/A13			
P4.1_IO_J1.5	57	P4.1/A12			
P4.2_A11_J3.25	58	P4.2/ACLK/TA2CLK/A11			
P4.3_A10_J1.6	59	P4.3/MCLK/RTCCLK/A10			
P4.4_A9_J3.26	60	P4.4/HSMCLK/SVMHOUT/A9			
P4.5_A8_J3.27	61	P4.5/A8			
P4.6_IO_J1.8	62	P4.6/A7			
P4.7_A6_J3.28	63	P4.7/A6			
P5.0_IO_J2.13	64	P5.0/A5			
P5.1_IO_J4.33	65	P5.1/A4			
P5.2_IO_J2.12	66	P5.2/A3			
P5.3	67	P5.3/A2			
P5.4_IO_J3.29	68	P5.4/A1			
P5.5_IO_J3.30	69	P5.5/A0			
P5.6_PWM_J4.37	70	P5.6/TA2.1/VREF+/VREF-/C1.7			
P5.7_IO_J2.17	71	P5.7/TA2.2/VREF-/VREF-/C1.6			
P6.0_A15_J1.2	54	P6.0/A15			
P6.1_A14_J3.23	55	P6.1/A14			
P6.2	76	P6.2/UCB1STE/C1.5			
P6.3	77	P6.3/UCB1CLK/C1.4			
P6.4_I2CSDA_J1.10	78	P6.4/UCB1SIMO/UCB1SDA/C1.3			
P6.5_I2CSCL_J1.9	79	P6.5/UCB1SOMI/UCB1SCL/C1.2			
P6.6_CAPTURE_J4.36	80	P6.6/TA2.3/UCB3SIMO/UCB3SDA/C1.1			
P6.7_CAPTURE_J4.35	81	P6.7/TA2.4/UCB3SOMI/UCB3SCL/C1.0			
P7.0	88	P7.0/PM_SMLK/PM_DMAED			
P7.1	89	P7.1/PM_COOUT/PM_TAOCLK			
P7.2	90	P7.2/PM_C1OUT/PM_TA1CLK			
P7.3	91	P7.3/PM_TAO.0			
P7.4	26	P7.4/PM_TA1.4/CO.5			
P7.5	27	P7.5/PM_TA1.3/CO.4			
P7.6	28	P7.6/PM_TA1.2/CO.3			
P7.7	29	P7.7/PM_TA1.1/CO.2			
			P1.0/LFXIN	41	
			P1.1/LFXOUT	42	
			P1.2/HFXOUT	85	
			P1.3/HFXIN	86	
			RSTN/NMI	83	RST
			SWDIO/TMS	94	TMS_SWDIO
			SWCLK/TCK	95	TCK_SWDCLK
			TDI	92	TDI
			TDO/SWO	93	TDO_SWO

Target MCU:
MSP432P401R



MSP432

EK D.(MSP432 ile tüm çevresel birimler arasındaki haberleşme kodları)

```
1  #include "Timer.h"
2  #include <Servo.h>
3  #include <Nextion.h>
4
5  // SW_VER: 2.4
6
7  //////////////// PIN TANIMLAMALARI ////////////////
8  const int RELAY_0 = 40;
9  const int RELAY_1 = 39;
10 const int RELAY_2 = 38;
11 const int RELAY_3 = 37;
12 const int RELAY_4 = 36;
13 const int RELAY_5 = 35;
14 const int RELAY_6 = 34;
15
16 const int BUZZER = 30;
17
18 const int PWM_LED = 31; //2 nolu pin
19 const int PIR_PIN = 26; // P4_4
20 const int servoPin= 23; // P6_1
21 int nemdeger = 0;
22 int Toprak_Nem_Sensor = 12; // P5_2
23 int LDR = 5; // P4_1 ADC
24
25 //////////////// VARIABLE ////////////////
26 int PIR_timer_value;
27 int PIR_Deger;
28
29 int relay_seg_gelen = 0;
30 int relay_value_gelen = 0;
31
32 int pwm_deger_gelen = 0;
33
34 int Oda_Sicaklik_Deger = 0;
35 int Oda_Sicaklik_Set = 30; // Default 30
36 bool Oto_Fan_Calisti = false;
37
38 uint16_t LPG_Degeri = 0;
39 uint16_t CO2_Degeri_Gelen = 0;
40 uint8_t Ev_Sicaklik_Degeri_Gelen = 0;
41 uint8_t Ev_Nem_Degeri_Gelen = 0;
42 uint8_t Salon_Sicaklik_Degeri_Gelen = 0;
43
44
45 bool alarm = false;
46 bool CO2_alarm = false;
47 bool LPG_alarm = false;
48 int alarm_timer = 0;
49 bool kapi_acik_status = false;
50
51 int servoAngle = 0; // servo position in degrees
52
53 int Toprak_Nem_Deger = 0;
54
55 char Gelen_Data[4]; // Uart Buffer - Wifi'dan gelen veriler için.
56
57 Timer myTimer; // Timer Değişkeni
58 Servo servo; // Servo Değişkeni
59
60
61
62 void setup()
63 {
64     Serial1.begin(115200); // WIFI NODEMCU ile Haberleşme
65     Serial.begin(9600); // EKTRAN
66
67     servo.attach(servoPin); // Servo motor 9 numarali pine baglandi.
68     pinMode(PWM_LED, OUTPUT); // LED Pin tanımlaması = ÇIKIS
69     pinMode(BUZZER, OUTPUT); // BUZZER Pin tanımlaması = ÇIKIS
70     pinMode(PIR_PIN, INPUT); // PIR Pin tanımlaması = GIRIS
71     pinMode(Toprak_Nem_Sensor, INPUT); // Toprak_Nem_Sensör Pin tanımlaması = GIRIS
72     pinMode(LDR, INPUT); // LDR Pin tanımlaması = GIRIS
```



```

74 pinMode(RELAY_0, OUTPUT); // RELAY Pin tanımlaması = ÇIKIS
75 pinMode(RELAY_1, OUTPUT); // RELAY Pin tanımlaması = ÇIKIS
76 pinMode(RELAY_2, OUTPUT); // RELAY Pin tanımlaması = ÇIKIS
77 pinMode(RELAY_3, OUTPUT); // RELAY Pin tanımlaması = ÇIKIS
78 pinMode(RELAY_4, OUTPUT); // RELAY Pin tanımlaması = ÇIKIS
79 pinMode(RELAY_5, OUTPUT); // RELAY Pin tanımlaması = ÇIKIS
80 pinMode(RELAY_6, OUTPUT); // RELAY Pin tanımlaması = ÇIKIS
81
82
83
84 digitalWrite(RELAY_0, LOW); // Röleleri Default state'e getirdim.
85 digitalWrite(RELAY_1, LOW);
86 digitalWrite(RELAY_2, LOW);
87 digitalWrite(RELAY_3, LOW);
88 digitalWrite(RELAY_4, LOW);
89 digitalWrite(RELAY_5, LOW);
90 digitalWrite(RELAY_6, LOW);
91
92 digitalWrite(BUZZER, LOW); // Buzzer Init Değerine Ayarlandı.
93 digitalWrite(PWM_LED, LOW); // Init Değerine Ayarlandı.
94
95
96 myTimer.begin(timerFunction, 500); // TIMER Kurulumu
97
98 myTimer.start(); // TIMER Başlat
99 }
100
101 void loop()
102 {
103     while (Serial1.available())
104     {
105         Serial1.readBytes(Gelen_Data,4); // Wifi'den gelen dataları alınıyor.
106         MQTT_Process(); // NodeMCU'dan gelen datalar burada işleniyor.
107     }
108
109     Buffer_Clear(); // NodeMCU'dan gelen datanın buffer'ını temizliyoruz.

```

```

111 PIR(); // PIR sensör ölçümü.
112
113 Toprak_Nem_Olcumu(); // Toprak Nem Sensör ölçümü.
114
115 Process(); // Genel alarmlar ve işlevler burada işletiliyor.
116
117 Nextion.LCD_Process(); // Ekran ile haberleşme kısımları burada işleniyor.
118 }
119
120
121 void MQTT_Process(void)
122 {
123     switch (Gelen_Data[0])
124     {
125         case 0xA0:
126             if(Gelen_Data[1] == 0x01)
127             {
128                 relay_cont(0, HIGH);
129             }
130             else
131             {
132                 relay_cont(0, LOW);
133             }
134             break;
135
136         case 0xA1:
137             if(Gelen_Data[1] == 0x01)
138             {
139                 relay_cont(1, HIGH);
140             }
141             else
142             {
143                 relay_cont(1, LOW);
144             }
145             break;

```

```

147 case 0xA2: // Sokak Kapı - Servo
148     if(Gelen_Data[1] == 0x01)
149     {
150         if(kapi_acik_status == false) // Kapının açık olma durumuna göre kontrol
151         {
152             kapi_ac();
153             kapi_acik_status = true;
154         }
155     }
156     else
157     {
158         if(kapi_acik_status == true) // Kapının kapalı olma durumuna göre kontrol
159         {
160             kapi_kapa();
161             kapi_acik_status = false;
162         }
163     }
164     break;
165
166 case 0xA3: // Salon PWM LED
167     pwm_ledd(Gelen_Data[1]);
168     break;
169 case 0xB0: // Ev Sicaklik
170     Ev_Sicaklik_Degeri_Gelen = Gelen_Data[1];
171     break;
172
173 case 0xB1: // Salon Sicaklik
174     Salon_Sicaklik_Degeri_Gelen = Gelen_Data[1];
175     break;
176
177 case 0xB2: // Ev Nem
178     Ev_Nem_Degeri_Gelen = Gelen_Data[1];
179     break;
180
181 case 0xB3: // CO2 Degeri
182     CO2_Degeri_Gelen = ((8 << (uint16_t)Gelen_Data[1]) | Gelen_Data[2]);
183     break;

```

```

186 default:
187
188     break;
189
190 }
191
192 }
193
194
195 void pwm_ledd(int pwm_deger)
196 {
197     int pwm_degeri = 0;
198     pwm_degeri = (255 * (100-pwm_deger)) / 100;
199     analogWrite(PWM_LED, pwm_degeri);
200     // delay(10);
201 }
202
203 void relay_cont(int relay_seg, int value)
204 {
205     switch(relay_seg)
206     {
207     case 0:
208         digitalWrite(RELAY_0, value); // BAHCE ISIK
209         break;
210     case 1:
211         digitalWrite(RELAY_1, value); // HAVALANDIRMA
212         break;
213     case 2:
214         digitalWrite(RELAY_2, value); // GIRIS ISIK
215         break;
216     case 3:
217         digitalWrite(RELAY_3, value); // Sulama Pompa
218         break;
219     case 4:
220         digitalWrite(RELAY_4, value); // PELTIER Isitici
221         break;
222     case 5:
223         digitalWrite(RELAY_5, value); // SALON ISIK

```

```

224     break;
225     case 6:
226         digitalWrite(RELAY_6, value);
227         break;
228     }
229 }
230
231 void kapi_ac(void)
232 {
233     servo.write(180);    // Turn SG90 servo Left to 45 degrees
234 }
235 void kapi_kapa(void)
236 {
237     servo.write(0);    // Turn SG90 servo Left to 45 degrees
238 }
239
240 void Buffer_Clear(void)
241 {
242     Gelen_Data[0] = 0x00;
243     Gelen_Data[1] = 0x00;
244     Gelen_Data[2] = 0x00;
245     Gelen_Data[3] = 0x00;
246 }
247
248
249 void PIR(void)
250 {
251     PIR_Deger = digitalRead(PIR_PIN); //Pir'in deęerini okuyoruz ve bunu deęer deęiřkenine atıyoruz.
252     if (PIR_Deger == HIGH)
253     {
254         PIR_timer_value = 0;
255         relay_cont(2, HIGH);
256     }
257 }

```

```

259 void Toprak_Nem_Olcumu(void)
260 {
261     Toprak_Nem_Deger = analogRead(Toprak_Nem_Sensor);
262 }
263
264
265 void Process(void)
266 {
267     if(Toprak_Nem_Deger > 120)
268     {
269         relay_cont(3, HIGH);
270     }
271     else
272     {
273         relay_cont(3, LOW);
274     }
275
276     if(LDR > 120)
277     {
278         relay_cont(0, HIGH);
279     }
280     else
281     {
282         relay_cont(0, LOW);
283     }
284
285     if(Oda_Sicaklik_Deger >= Oda_Sicaklik_Set)
286     {
287         Oto_Fan_Calisti = true;
288         relay_cont(1, HIGH); // Havalendirma açılır.
289         relay_cont(4, LOW); // Isıtıcı kapanır.
290     }
291     else if(Oda_Sicaklik_Deger < (Oda_Sicaklik_Set - 5))
292     {
293         Oto_Fan_Calisti = false;
294         relay_cont(4, HIGH); // Isıtıcı açılır.
295         relay_cont(1, LOW); // Havalendirma kapanır.
296     }

```

```

298     if(CO2_Degeri_Gelen > 500) { CO2_alarm = true; }
299     else { CO2_alarm = false; }
300
301     if(LPG_Degeri > 500) { LPG_alarm = true; }
302     else { LPG_alarm = false; }
303
304
305 }
306
307 void timerFunction(void)
308 {
309     if(PIR_Deger == LOW) // Otomatik 5 sn içinde kapının kapanmayı sağlayan kısım
310     {
311         if(PIR_timer_value < 5)
312         {
313             if(++PIR_timer_value >= 5)
314             {
315                 relay_cont(2, LOW);
316             }
317         }
318     }
319
320     if((alarm == true) || (CO2_alarm == true) || (LPG_alarm == true)) // BUZZER ALARM
321     {
322         if(alarm_timer == 2)
323         {
324             digitalWrite(BUZZER, HIGH);
325         }
326         else if(alarm_timer >= 4)
327         {
328             digitalWrite(BUZZER, LOW);
329         }
330     }
331     else
332     {
333         digitalWrite(BUZZER, LOW);
334     }
335 }

```


EK E.(Arduinio'da MQ-135 sensörünün MQTT haberleşme kodları)

```
1 #include <ESP8266WiFi.h>
2 #include <Adafruit_MQTT.h>
3 #include <Adafruit_MQTT_Client.h>
4 #include "MQ135.h"
5
6 /***** WiFi Access Point *****/
7
8 #define WLAN_SSID      "ozen123"
9 #define WLAN_PASS      "ozen123456"
10
11 /***** Adafruit.io Setup *****/
12
13 #define AIO_SERVER      "io.adafruit.com"
14 #define AIO_SERVERPORT  1883          // use 8883 for SSL
15 #define AIO_USERNAME    "ozensine"
16 #define AIO_KEY         "adbd4a25d18e4a44ba3e14799b154c54"
17
18 /***** Global State *****/
19
20 WiFiClient client; // MQTT server bağlanmak için class oluşturuyoruz.
21
22 // Wifi ye ve server'a bağlanıyoruz.
23 Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
24
25 /***** MQ 135 Setup *****/
26 #define RZERO 206.85 // Define RZERO Calibration Value
27
28 int smokeA0 = A0;
29 MQ135 gasSensor = MQ135(smokeA0);
30
31 /***** Feeds *****/
32 Adafruit_MQTT_Publish co2 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/co2");
33
34 /-----VARIABLES GLOBALES-----
35 int contconexion = 0;
36 unsigned long previousMillis = 0;
37 char charPulsador [15];
38 String strPulsador;
39 String strPulsadorUltimo;
40 bool Connected = false;
41
42 /-----
43
44 void MQTT_connect();
45
46 void setup()
47 {
48     // INIT Serial
49     Serial.begin(115200);
50     delay(10);
51
52     pinMode(smokeA0, INPUT); // smokeA0 Giriş olarak ayarlandı.
53     float rzero = gasSensor.getRZero(); // MQ135'i kalibre ediyoruz.
54     delay(2000);
55     Serial.print("MQ135 RZERO Calibration Value : ");
56     Serial.println(rzero);
57
58     delay(500);
59     WiFi.disconnect();
60     delay(2000);
61
62     WiFi.begin(WLAN_SSID, WLAN_PASS);
63     // Wifi ad ve şifreyi girer.
64     while (WiFi.status() != WL_CONNECTED and contconexion < 50)
65     { // Bağlanamıyorsa 50'ye kadar dener ve iptal eder.
66         {
67             ++contconexion;
68             delay(500);
69             Serial.print(".");
70         }
71     }
72 }
```

```

71     if (contconexion <50)
72     {
73         // Sabit ip ile kullanmak
74         /*
75         IPAddress ip(192,168,1,105);
76         IPAddress gateway(192,168,1,1);
77         IPAddress subnet(255,255,255,0);
78         WiFi.config(ip, gateway, subnet);
79         */
80         Serial.println("WiFi bağlantısı Kuruldu");
81         Serial.println(WiFi.localIP());
82         Connected = true;
83     }
84     else
85     {
86         Connected = false;
87         Serial.println("");
88         Serial.println("WIFI Bağlantı Hatası");
89     }
90 }
91
92 //-----LOOP-----
93 void loop()
94 {
95     if(Connected == true)
96     {
97         MQTT_connect();
98
99         float co2_ppm = gasSensor.getPPM();

```

```

103     if (currentMillis - previousMillis >= 20000) // 20sn'de bir veri gönderir.
104     {
105         previousMillis = currentMillis;
106
107         Serial.print(F("\nCO2 Gönderiliyor "));
108         Serial.print(co2_ppm);
109         Serial.print("...");
110         if (! co2.publish(co2_ppm)) {
111             Serial.println(F("Failed"));
112         } else {
113             Serial.println(F("OK!"));
114         }
115     }
116
117     mqtt.processPackets(500);
118     // bu bizim 'gelen abonelik paketlerini bekle ve geri arama em' meşgul subloop'umuz
119
120     // MQTT bağlantısını canlı tutmak için sunucuya ping atma
121
122     if(! mqtt.ping())
123     {
124         mqtt.disconnect();
125     }
126 }
127
128
129 // MQTT sunucusuna gerektiği gibi bağlanma ve yeniden bağlanma işlevi.
130 // Döngü işlevinde çağrılmalıdır ve bağlanırsa özen gösterecektir.
131 void MQTT_connect()
132 {
133     int8_t ret;
134
135     // Stop if already connected.
136     if (mqtt.connected())
137     {
138         return;
139     }

```

```

141     Serial.print("Connecting to MQTT... ");
142
143     uint8_t retries = 3;
144     while ((ret = mqtt.connect()) != 0) { // Connect bağlı için 0 döndürür
145         Serial.println(mqtt.connectErrorString(ret));
146         Serial.println("Retrying MQTT connection in 10 seconds...");
147         mqtt.disconnect();
148         delay(10000); // wait 10 seconds
149         retries--;
150         if (retries == 0)
151         {
152             retries = 3;
153             // while (1); // WDT'nin Sıfırlamasını Bekle
154         }
155     }
156     Serial.println("MQTT Connected!");
157 }
158
159

```

EK F.(Arduinio'da DHT-11 sensörünün MQTT haberleşme kodları)

```
1 #include <ESP8266WiFi.h>
2 #include <Adafruit_MQTT.h>
3 #include <Adafruit_MQTT_Client.h>
4 #include "DHT.h"
5
6 /***** WiFi Access Point *****/
7
8 #define WLAN_SSID      "ozen123"
9 #define WLAN_PASS      "ozen123456"
10
11 /***** Adafruit.io Setup *****/
12
13 #define AIO_SERVER      "io.adafruit.com"
14 #define AIO_SERVERPORT  1883           // use 8883 for SSL
15 #define AIO_USERNAME    "ozensine"
16 #define AIO_KEY         "adbd4a25d18e4a44ba3e14799b154c54"
17
18 /***** Global State *****/
19
20 WiFiClient client; // MQTT server bağlanmak için class oluşturuyoruz.
21
22 Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
23 // Wifi ye ve server'a bağlanıyoruz.
24 /***** Feeds *****/
25
26 // Setup a feed called 'photocell' for publishing.
27 // Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
28 Adafruit_MQTT_Publish sicaklik = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Sicaklik");
29 Adafruit_MQTT_Publish nem = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/nem");
30
31 /***** MQ 135 Setup *****/
32 #define DHTPIN 5
33 #define DHTTYPE DHT11
34
35 DHT nem_sicaklik(DHTPIN, DHTTYPE);
36
37 //-----VARIABLES GLOBALES-----
38 int contconexion = 0;
39 unsigned long previousMillis = 0;
40 char charPulsador [15];
41 String strPulsador;
42 String strPulsadorUltimo;
43 bool Connected = false;
44
45 //-----
46
47
48 void MQTT_connect();
49
50 void setup()
51 {
52   // INIT Serial
53   Serial.begin(115200);
54   delay(10);
55
56   pinMode(14, INPUT); // D5
57   nem_sicaklik.begin();
58
59   delay(1000);
60   WiFi.disconnect();
61   delay(2000);
62
63   WiFi.begin(WLAN_SSID, WLAN_PASS);
64   // Wifi ad ve şifreyi girer.
65   while (WiFi.status() != WL_CONNECTED and contconexion < 50)
66     // Bağlanamıyorsa 50'ye kadar dener ve iptal eder.
67     {
68       ++contconexion;
69       delay(500);
70       Serial.print(".");
71     }
```



```

72   if (contconexion <50)
73   {
74       // Sabit ip ile kullanmak
75       /*
76       IPAddress ip(192,168,1,105);
77       IPAddress gateway(192,168,1,1);
78       IPAddress subnet(255,255,255,0);
79       WiFi.config(ip, gateway, subnet);
80       */
81       Serial.println("");
82       Serial.println("WiFi bağlantısı Kuruldu");
83       Serial.println(WiFi.localIP());
84       Connected = true;
85   }
86   else
87   {
88       Connected = false;
89       Serial.println("");
90       Serial.println("WIFI Bağlantı Hatası");
91   }
92 }
93
94 //-----LOOP-----
95 void loop()
96 {
97     if(Connected == true)
98     {
99         MQTT_connect();
100
101         float nem_deger = nem_sicaklik.readHumidity();
102         float temperature = nem_sicaklik.readTemperature();
103         // float nnlog = 1;
104
105         unsigned long currentMillis = millis();
106
107         if (currentMillis - previousMillis >= 20000)
108         {
109             previousMillis = currentMillis;
110
111             Serial.print(F("\nSicaklık Gönderiliyor "));
112             Serial.print(temperature);
113             Serial.print("...");
114             if (! sicaklik.publish(temperature)) {
115                 Serial.println(F("Failed"));
116             } else {
117                 Serial.println(F("OK!"));
118             }
119
120             Serial.print(F("\nNem Gönderiliyor "));
121             Serial.print(nem_deger);
122             Serial.print("...");
123             if (! nem.publish(nem_deger)) {
124                 Serial.println(F("Failed"));
125             } else {
126                 Serial.println(F("OK!"));
127             }
128         }
129         mqtt.processPackets(500);
130         // bu bizim 'gelen abonelik paketlerini bekle ve geri arama em' meşgul subloop'umuz
131
132         // MQTT bağlantısını canlı tutmak için sunucuya ping atma
133
134         if(! mqtt.ping())
135         {
136             mqtt.disconnect();
137         }
138     }
139 }

```



```

141 // MQTT sunucusuna gerektiği gibi bağlanma ve yeniden bağlanma işlevi.
142 // Döngü işlevinde çağrılmalıdır ve bağlanırsa özen gösterecektir.
143 void MQTT_connect()
144 {
145     int8_t ret;
146
147     // Stop if already connected.
148     if (mqtt.connected())
149     {
150         return;
151     }
152
153     Serial.print("Connecting to MQTT... ");
154
155     uint8_t retries = 3;
156     while ((ret = mqtt.connect()) != 0) { // Connect bağlı için 0 döndürür
157         Serial.println(mqtt.connectErrorString(ret));
158         Serial.println("Retrying MQTT connection in 10 seconds...");
159         mqtt.disconnect();
160         delay(10000); // wait 10 seconds
161         retries--;
162         if (retries == 0)
163         {
164             retries = 3;
165             // while (1); // WDT'nin Sıfırlamasını Bekle
166         }
167     }
168     Serial.println("MQTT Connected!");
169 }
170
171

```

EK G. (Adafruit IO kontrol arayüzü veri gönderim ve alım kodları)

```
1 #include <ESP8266WiFi.h>
2 #include <WiFiClient.h>
3 #include "Adafruit_MQTT.h"
4 #include "Adafruit_MQTT_Client.h"
5
6 /***** WiFi Access Point *****/
7
8 #define WLAN_SSID      "ozen123"
9 #define WLAN_PASS      "ozen123456"
10
11 /***** Adafruit.io Setup *****/
12
13 #define AIO_SERVER      "io.adafruit.com"
14 #define AIO_SERVERPORT 1883          // use 8883 for SSL
15 #define AIO_USERNAME   "ozensine"
16 #define AIO_KEY        "adbd4a25d18e4a44ba3e14799b154c54"
17
18 /***** Global State *****/
19
20 byte msp432_buffer[] = {0x00, 0x00, 0x00, 0x00};
21
22 /***** Global State (you don't need to change this!) *****/
23
24 // MQTT server bağlanmak için class oluşturuyoruz.
25 WiFiClient client;
26
27 // Wifi ye ve server'a bağlanıyoruz.
28 Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
29
30 /***** Feeds *****/
31
32 // Setup a feed called 'onoff' for subscribing to changes to the button
33 Adafruit_MQTT_Subscribe salon_isik = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/salon", MQTT_QOS_1);
34 Adafruit_MQTT_Subscribe Bahce_isik = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Bahce_isik", MQTT_QOS_1);
35 Adafruit_MQTT_Subscribe Havalandirma = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/havalandirma", MQTT_QOS_1);
36 Adafruit_MQTT_Subscribe Kapi = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/kapi", MQTT_QOS_1);
37
38 Adafruit_MQTT_Subscribe Salon_Sicaklik = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Salon_Sicaklik", MQTT_QOS_1);
39 Adafruit_MQTT_Subscribe Sicaklik = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Sicaklik", MQTT_QOS_1);
40 Adafruit_MQTT_Subscribe Nem = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/nem", MQTT_QOS_1);
41 Adafruit_MQTT_Subscribe CO2 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/co2", MQTT_QOS_1);
42
43 /***** Sketch Code *****/
44
45 void Bahce_Isik_Callback(char *data, uint16_t len)
46 {
47     msp432_buffer[0] = 0xA0;
48     Serial.print("Bahce Isik Degeri= ");
49     Serial.println(data);
50
51     String message = String(data);
52     message.trim();
53
54     if (message == "ON")
55     {
56         msp432_buffer[1] = 0x01;
57     }
58     else if (message == "OFF")
59     {
60         msp432_buffer[1] = 0x00;
61     }
62     Serial1.write(msp432_buffer, 2);
63 }
64
65
66 void Havalandirma_Callback(char *data, uint16_t len)
67 {
68     msp432_buffer[0] = 0xA1;
69     Serial.print("Havalandirma Degeri= ");
70     Serial.println(data);
71
72     String message = String(data);
73     message.trim();
```

```

75     if (message == "ON")
76     {
77         msp432_buffer[1] = 0x01;
78     }
79     else if (message == "OFF")
80     {
81         msp432_buffer[1] = 0x00;
82     }
83     Serial1.write(msp432_buffer, 2);
84 }
85
86 void Kapi_Callback(char *data, uint16_t len)
87 {
88     msp432_buffer[0] = 0xA2;
89     Serial.print("Kapi Degeri= ");
90     Serial.println(data);
91
92     String message = String(data);
93     message.trim();
94
95     if (message == "ON")
96     {
97         msp432_buffer[1] = 0x01;
98     }
99     else if (message == "OFF")
100    {
101        msp432_buffer[1] = 0x00;
102    }
103    Serial1.write(msp432_buffer, 2);
104 }
105
106 void Salon_Isik_Callback(char *data, uint16_t len)
107 {
108     msp432_buffer[0] = 0xA3;
109     Serial.print("Salon Isik Degeri= ");
110     Serial.println(data);
111
112     String message = String(data);
113     message.trim();
114     msp432_buffer[1] = message.toInt();
115     Serial1.write(msp432_buffer, 2);
116 }
117
118 void Ev_Sicaklik_Callback(char *data, uint16_t len)
119 {
120     msp432_buffer[0] = 0xB0;
121     Serial.print("Ev Sicaklik deęeri alındı: ");
122     Serial.println(data);
123
124     String message = String(data);
125     message.trim();
126     msp432_buffer[1] = message.toInt();
127     Serial1.write(msp432_buffer, 2);
128 }
129
130 void Salon_Sicaklik_Callback(char *data, uint16_t len)
131 {
132     msp432_buffer[0] = 0xB1;
133     Serial.print("Salon Sicaklik deęeri alındı: ");
134     Serial.println(data);
135
136     String message = String(data);
137     message.trim();
138     msp432_buffer[1] = message.toInt();
139     Serial1.write(msp432_buffer, 2);
140 }
141
142 void Ev_Nem_Callback(char *data, uint16_t len)
143 {
144     msp432_buffer[0] = 0xB2;
145     Serial.print("Ev Nem deęeri alındı: ");
146     Serial.println(data);

```

```

148     String message = String(data);
149     message.trim();
150     msp432_buffer[1] = message.toInt();
151     Serial1.write(msp432_buffer, 2);
152 }
153
154 void Ev_CO2_Callback(char *data, uint16_t len)
155 {
156     msp432_buffer[0] = 0xB3;
157     Serial.print("Ev CO2 deęeri alındı: ");
158     Serial.println(data);
159
160     String message = String(data);
161     message.trim();
162     msp432_buffer[1] = message.toInt();
163     Serial1.write(msp432_buffer, 3); // 3 byte gnderilecek.
164 }
165
166
167 //-----VARIABLES GLOBALES-----
168 int contconexion = 0;
169 unsigned long previousMillis = 0;
170 char charPulsador [15];
171 String strPulsador;
172 String strPulsadorUltimo;
173 bool Connected = false;
174
175 //-----
176
177 void setup() {
178
179     //prepara GPI13 y 12 como salidas
180     pinMode(13, OUTPUT); // D7 salida analgica
181     analogWrite(13, 0); // analogWrite(pin, value);
182     pinMode(12, OUTPUT); // D6 salida digital
183     digitalWrite(12, LOW);
184
185     // Inicia Serial
186     Serial.begin(115200);
187     Serial.println("");
188
189     Serial1.begin(9600);
190
191     WiFi.begin(WLAN_SSID, WLAN_PASS); // WIFI Baęlantısı // Wifi ad ve Őifreyi girer.
192     while (WiFi.status() != WL_CONNECTED and contconexion <50) // Baęlanamıyorsa 50'ye kadar dener ve iptal eder.
193     {
194         ++contconexion;
195         delay(500);
196         Serial.print(".");
197     }
198     if (contconexion <50)
199     {
200         // Sabit ip ile kullanmak
201         /*
202         IPAddress ip(192,168,1,156);
203         IPAddress gateway(192,168,1,1);
204         IPAddress subnet(255,255,255,0);
205         WiFi.config(ip, gateway, subnet);
206         */
207         Serial.println("WiFi baęlantısı Kuruldu");
208         Serial.println(WiFi.localIP());
209         Connected = true;
210     }
211     else
212     {
213         Connected = false;
214         Serial.println("WIFI Baęlantı Hatası");
215     }
216
217     salon_isik.setCallback(Salon_Isik_Callback);
218     Bahce_isik.setCallback(Bahce_Isik_Callback);
219     Havalandirma.setCallback(Havalandirma_Callback);
220     Kapi.setCallback(Kapi_Callback);

```



```

222     Salon_Sicaklik.setCallback(Salon_Sicaklik_Callback);
223     Sicaklik.setCallback(Ev_Sicaklik_Callback);
224     Nem.setCallback(Ev_Nem_Callback);
225     CO2.setCallback(Ev_CO2_Callback);
226
227     mqtt.subscribe(&salon_isik);
228     mqtt.subscribe(&Bahce_isik);
229     mqtt.subscribe(&Havalandirma);
230     mqtt.subscribe(&Kapi);
231
232     mqtt.subscribe(&Salon_Sicaklik);
233     mqtt.subscribe(&Sicaklik);
234     mqtt.subscribe(&Nem);
235     mqtt.subscribe(&CO2);
236 }
237
238 //-----LOOP-----
239 void loop()
240 {
241     if(Connected == true)
242     {
243         MQTT_connect();
244
245         unsigned long currentMillis = millis();
246
247         if (currentMillis - previousMillis >= 5000)
248         {
249             previousMillis = currentMillis;
250
251             // Serial1.write(msp432_buffer, 8);
252             // Serial.write(msp432_buffer, 8);
253         }
254
255         mqtt.processPackets(500); // bu bizim 'gelen abonelik paketlerini bekle ve geri arama em' meşgul subloop'umuz
256
257         if(! mqtt.ping()) // MQTT bağlantısını canlı tutmak için sunucuya ping atma
258         {
259             mqtt.disconnect();
260         }
261     }
262 }
263
264 // MQTT sunucusuna gerektiği gibi bağlanma ve yeniden bağlanma işlevi.
265 // Döngü işlevinde çağrılmalıdır ve bağlanırsa özen gösterecektir.
266 void MQTT_connect()
267 {
268     int8_t ret;
269
270     if (mqtt.connected()) // Stop if already connected.
271     {
272         return;
273     }
274
275     Serial.print("Connecting to MQTT... ");
276
277     uint8_t retries = 3;
278     while ((ret = mqtt.connect()) != 0) // Connect bağlı için 0 döndürür
279     {
280         Serial.println(mqtt.connectErrorString(ret));
281         Serial.println("Retrying MQTT connection in 10 seconds...");
282         mqtt.disconnect();
283         delay(10000); // wait 10 seconds
284         retries--;
285         if (retries == 0)
286         {
287             retries = 3;
288             // while (1); // WDT'nin Sıfırlamasını Bekle
289         }
290     }
291     Serial.println("MQTT Connected!");
292 }

```

EK H. (Openweather API'dan mevsimsel sıcaklık verisi alım kodları)

```
1 #include <ArduinoJson.h>
2 #include <ESP8266WiFi.h>
3 #include <WiFiClient.h>
4
5 char ssid[] = "MERTKSK35"; // WIFI detayları
6 char pass[] = "mert1488191235";
7
8 WiFiClient client;
9
10 const char server[] = "api.openweathermap.org"; // OpenWeatherTheMap sitesi
11
12 String nameOfCity = "Manisa,TR"; // Şehir İsmi, Ülke Kodu
13
14 String apiKey = "7efd7b59ebc118d46af557966a564927"; // Siteden alınan Kullanıcı adı ile bütünlük API KEY
15
16 String text;
17
18 int jsonend = 0;
19 boolean startJson = false;
20 int status = WL_IDLE_STATUS;
21
22 #define JSON_BUFF_DIMENSION 2500
23
24 unsigned long lastConnectionTime = 10 * 60 * 1000; // Sunucuya en son bağlandığınızda, milisaniye cinsinden
25 const unsigned long postInterval = 10 * 60 * 1000; // 10 dakikalık kayıt aralığı (10L * 1000L; test için 10 saniye gecikme)
26
27 void setup()
28 {
29   Serial.begin(9600);
30
31   text.reserve(JSON_BUFF_DIMENSION);
32
33   WiFi.begin(ssid,pass);
34   Serial.println("WIFI Bağlanıyor...");
35   while (WiFi.status() != WL_CONNECTED)
36   {
37     delay(500);
38     Serial.print(".");
39   }
40   Serial.println("WIFI Bağlandı");
41   printWiFiStatus();
42 }
43
44 void loop()
45 {
46   //10 dk'lık aralıklarla sorgu çekiliyor.
47   if (millis() - lastConnectionTime > postInterval)
48   {
49     lastConnectionTime = millis();
50     makeHttpRequest();
51
52     Serial.print(tempNow, 16); // Şuanki Manisa İlının Sıcaklık Değeri
53     Serial.print(humidityNow, 16); // Şuanki Manisa İlının Nem Değeri
54     Serial.print(weatherNow, 16); // Şuanki Manisa İlının Hava Durumu
55
56     Serial.print(tempLater, 16); // Bir Sonraki Tahmin Manisa İlının Sıcaklık Değeri
57     Serial.print(humidityLater, 16); // Bir Sonraki Tahmin Manisa İlının Nem Değeri
58     Serial.print(weatherLater, 16); // Bir Sonraki Tahmin Manisa İlının Hava Durumu
59   }
60 }
61
62 void printWiFiStatus() // WIFI Bilgilerini LOG'a yazdırma
63 {
64   Serial.print("SSID: ");
65   Serial.println(WiFi.SSID());
66
67   IPAddress ip = WiFi.localIP();
68   Serial.print("IP Address: ");
69   Serial.println(ip);
70
71   long rssi = WiFi.RSSI();
72   Serial.print("signal strength (RSSI):");
73 }
```

```

74 Serial.print(rssi);
75 Serial.println(" dB");
76 }
77
78 void makeHttpRequest() // OWM'den veri istemek için
79 {
80 // müşterinin sunucuya bağlantı kurmasına izin vermek için yeni bir istek göndermeden önce herhangi bir bağlantıyı kapat
81 client.stop();
82
83 // başarılı bir bağlantı varsa:
84 if (client.connect(server, 80)) // VERİ çekmek için gerekli olan HTTP bilgileri(Sorgu) gönderilir.
85 {
86 client.println("GET /data/2.5/forecast?q=" + nameOfCity + "&APPID=" + apiKey + "&mode=json&units=metric&cnt=2 HTTP/1.1");
87 client.println("Host: api.openweathermap.org");
88 client.println("User-Agent: ArduinoWiFi/1.1");
89 client.println("Connection: close");
90 client.println();
91
92 unsigned long timeout = millis();
93 while (client.available() == 0) {
94 if (millis() - timeout > 5000)
95 {
96 Serial.println(">>> Client Timeout Oldu !");
97 client.stop();
98 return;
99 }
100 }
101
102 char c = 0;
103 while (client.available())
104 {
105 c = client.read();
106
107 if (c == '{')
108 {
109 startJson = true;
110 jsonend++;
111 }
112 if (c == '}')
113 {
114 jsonend--;
115 }
116 if (startJson == true)
117 {
118 text += c;
119 }
120 if (jsonend == 0 && startJson == true)
121 {
122 parseJson(text.c_str());
123 text = "";
124 startJson = false;
125 }
126 }
127 }
128 else
129 {
130 Serial.println("Connection Failed");
131 return;
132 }
133 }
134
135 void parseJson(const char * jsonString) // OWM'den alınan json verilerini ayrıştırmak için
136 {
137 const size_t bufferSize = 2*JSON_ARRAY_SIZE(1) + JSON_ARRAY_SIZE(2) + 4*JSON_OBJECT_SIZE(1)
138 + 3*JSON_OBJECT_SIZE(2) + 3*JSON_OBJECT_SIZE(4) + JSON_OBJECT_SIZE(5) + 2*JSON_OBJECT_SIZE(7) + 2*JSON_OBJECT_SIZE(8) + 720;
139 DynamicJsonBuffer jsonBuffer(bufferSize);
140
141 JsonObject& root = jsonBuffer.parseObject(jsonString);
142 if (!root.success()) {
143 Serial.println("ParseObject() Failed");
144 return;

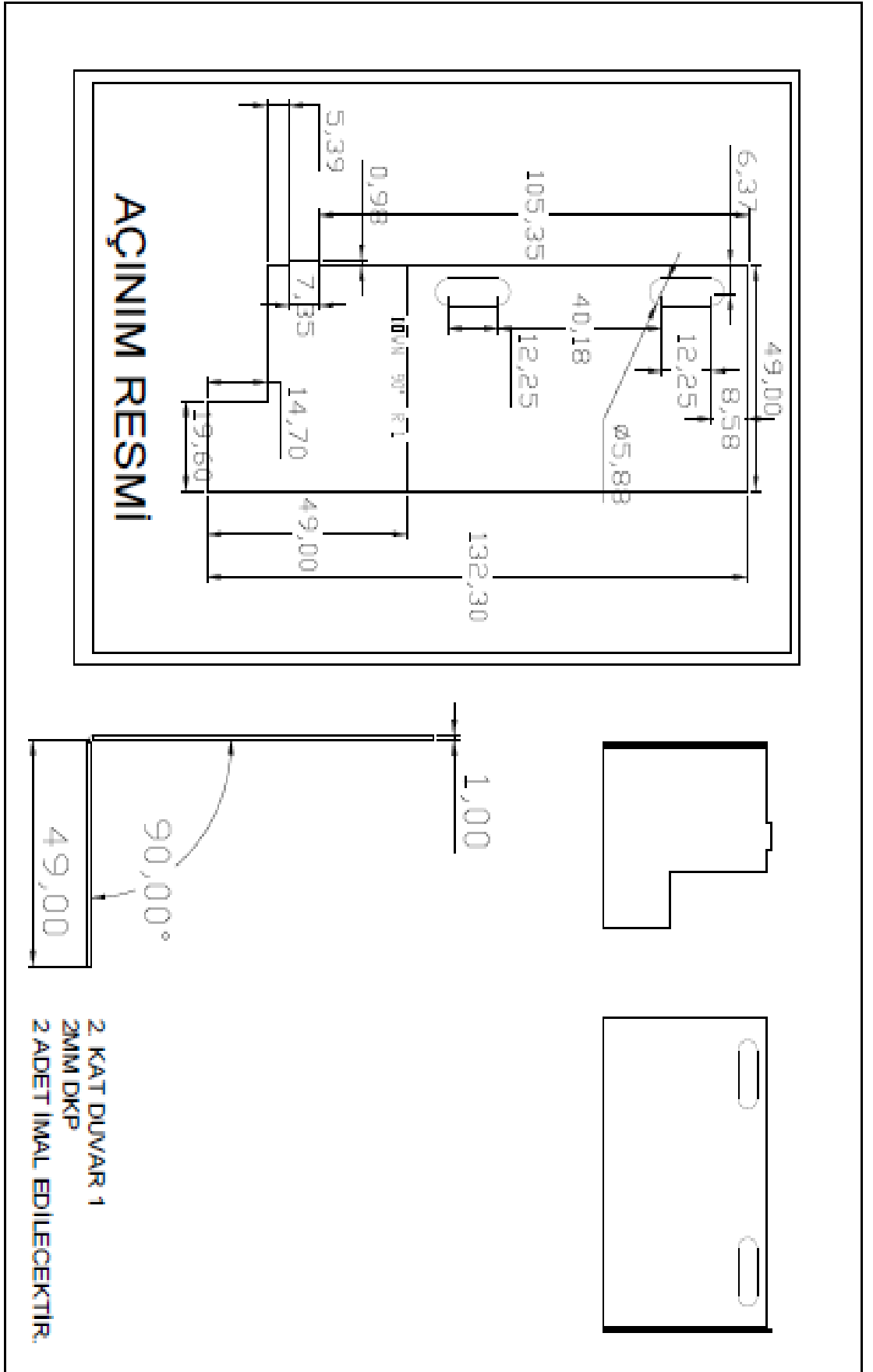
```

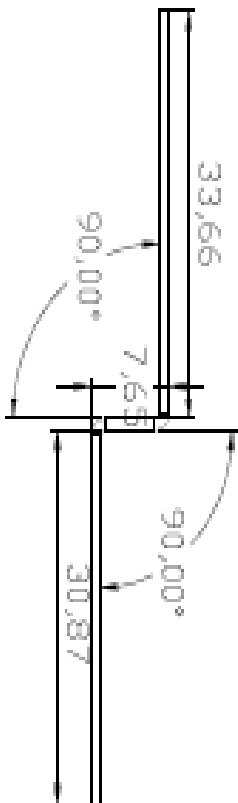
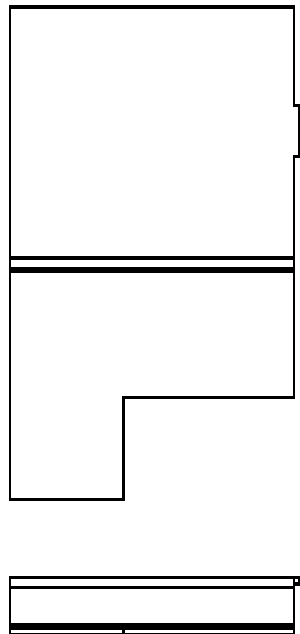
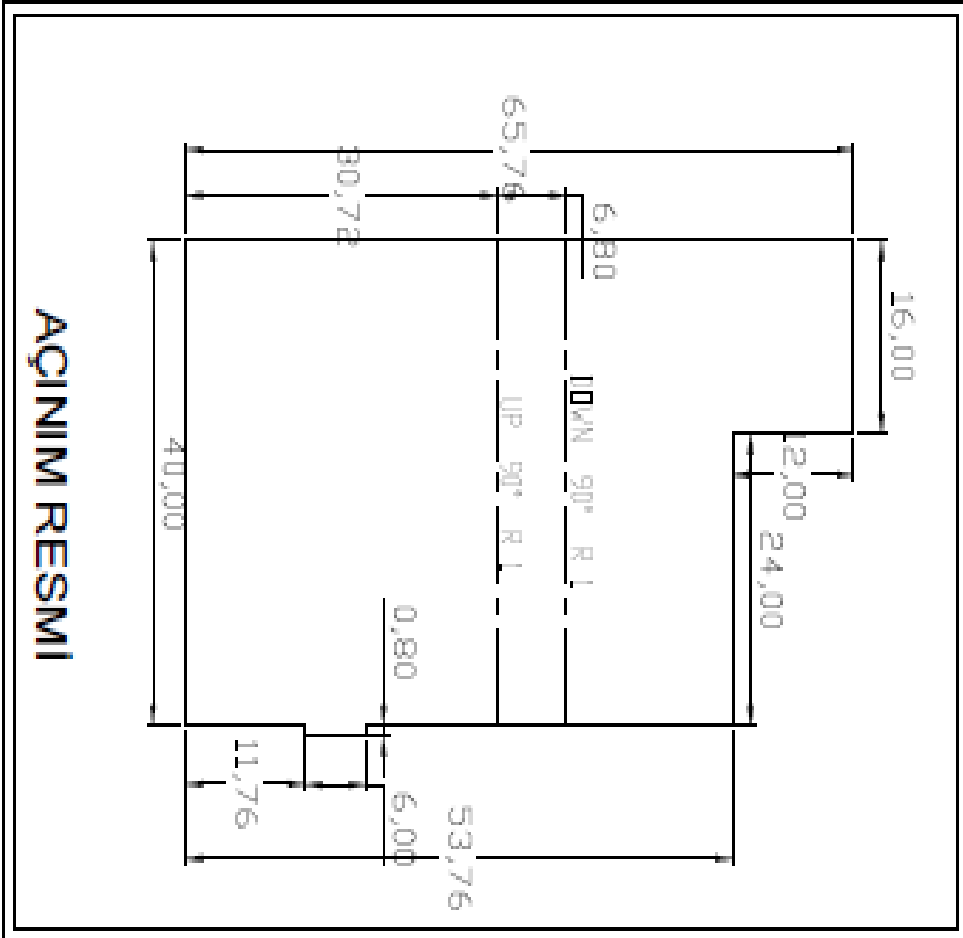
```

145     }
146
147     JSONArray& list = root["list"];
148     JSONObject& nowT = list[0];
149     JSONObject& later = list[1];
150
151     String city = root["city"]["name"];
152
153     float tempNow = nowT["main"]["temp"];
154     float humidityNow = nowT["main"]["humidity"];
155     String weatherNow = nowT["weather"][0]["description"];
156
157     float tempLater = later["main"]["temp"];
158     float humidityLater = later["main"]["humidity"];
159     String weatherLater = later["weather"][0]["description"];
160
161     // Dört ana hava olasılığını denetleme
162     diffDataAction(weatherNow, weatherLater, "clear");
163     diffDataAction(weatherNow, weatherLater, "rain");
164     diffDataAction(weatherNow, weatherLater, "snow");
165     diffDataAction(weatherNow, weatherLater, "hail");
166
167     Serial.println();
168 }
169
170 void diffDataAction(String nowT, String later, String weatherType)
171 {
172     int indexNow = nowT.indexOf(weatherType);
173     int indexLater = later.indexOf(weatherType);
174     if (weatherType == "rain")
175     {
176         if (indexNow == -1 && indexLater != -1)
177         {
178             Serial.println("Hava " + weatherType + " Yağışlı olacak. Öngörülen Deger " + later);
179         }
180     }
181
182     else if (weatherType == "snow")
183     {
184         if (indexNow == -1 && indexLater != -1)
185         {
186             Serial.println("Hava " + weatherType + " Yağışlı olacak. Öngörülen Deger " + later);
187         }
188     }
189     else if (weatherType == "hail")
190     {
191         if (indexNow == -1 && indexLater != -1)
192         {
193             Serial.println("Hava " + weatherType + " Yağışlı olacak. Öngörülen Degered " + later);
194         }
195     }
196     else
197     {
198         if (indexNow == -1 && indexLater != -1)
199         {
200             Serial.println("Hava Güneşli olacak. Öngörülen Deger " + later);
201         }
202     }
203 }
204

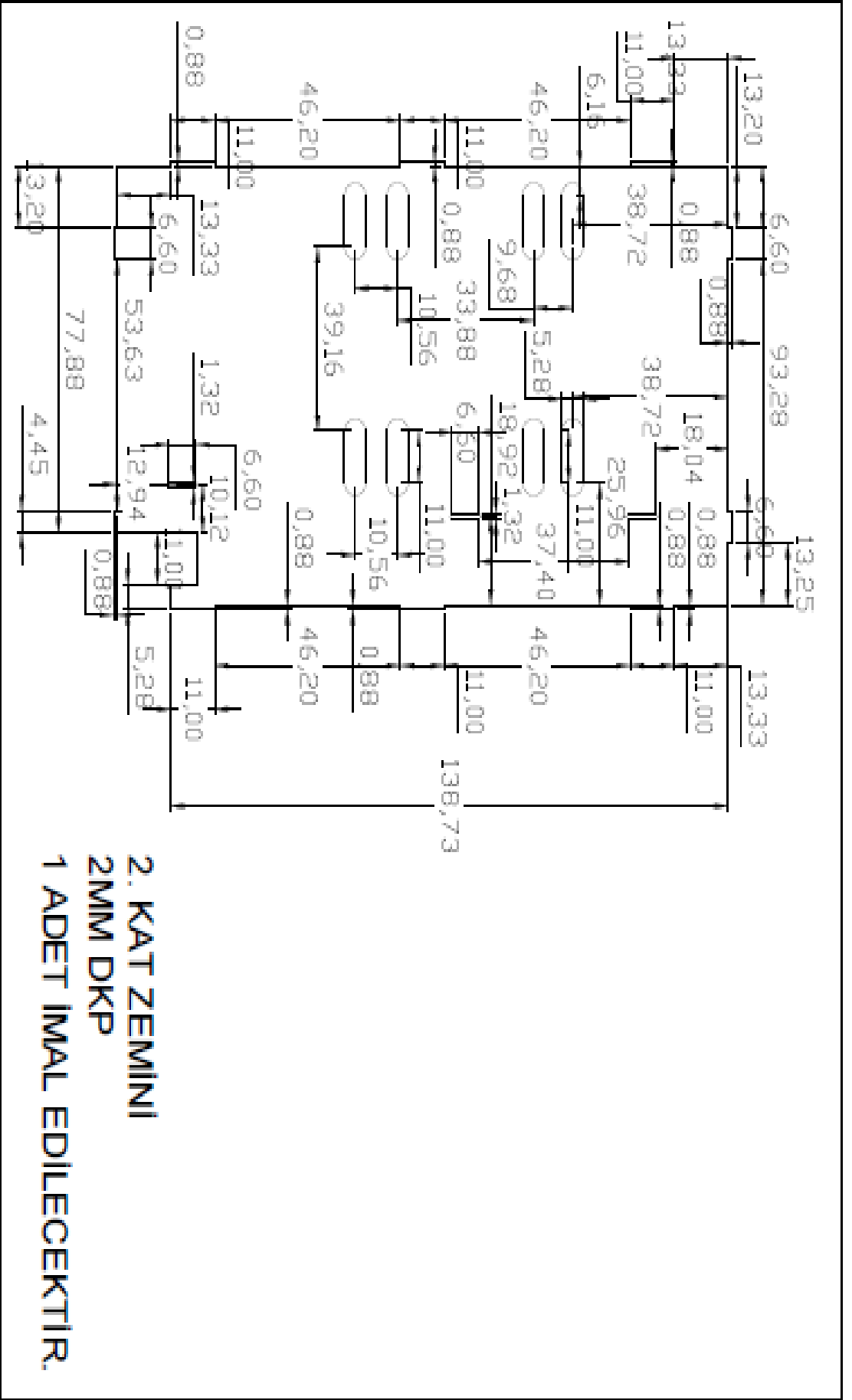
```

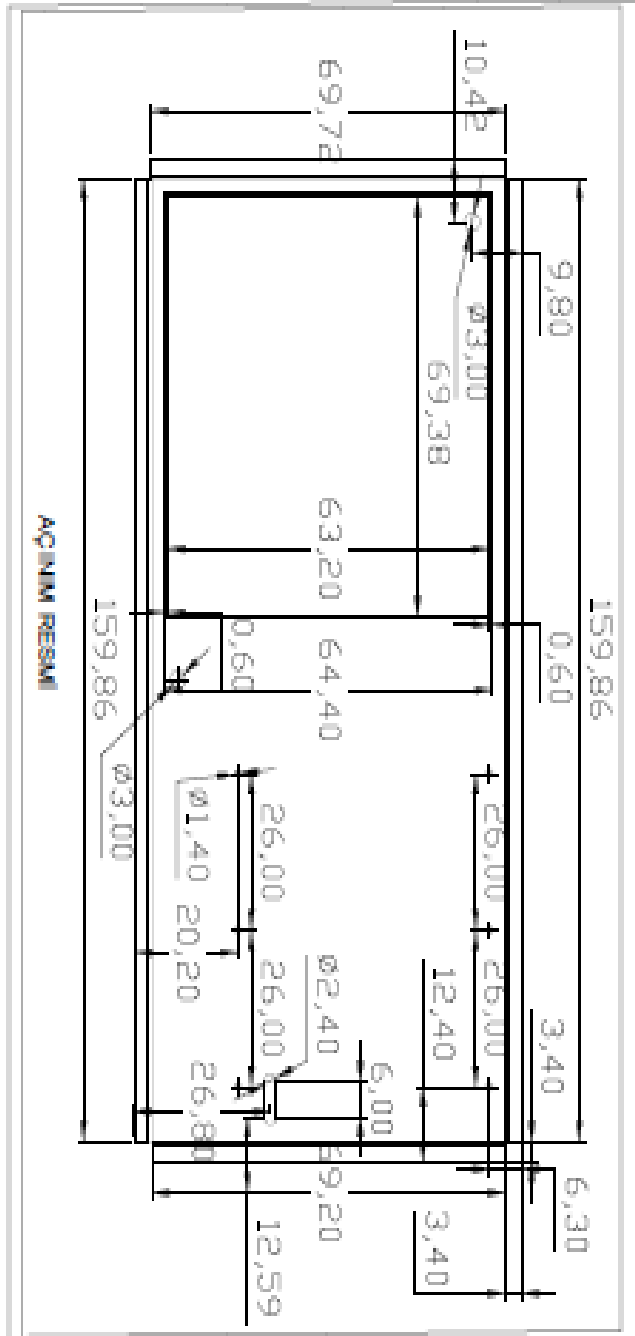

EK I. (Ev maketinin Solid Works’de çizilen teknik resimleri ve boyut ölçüleri)



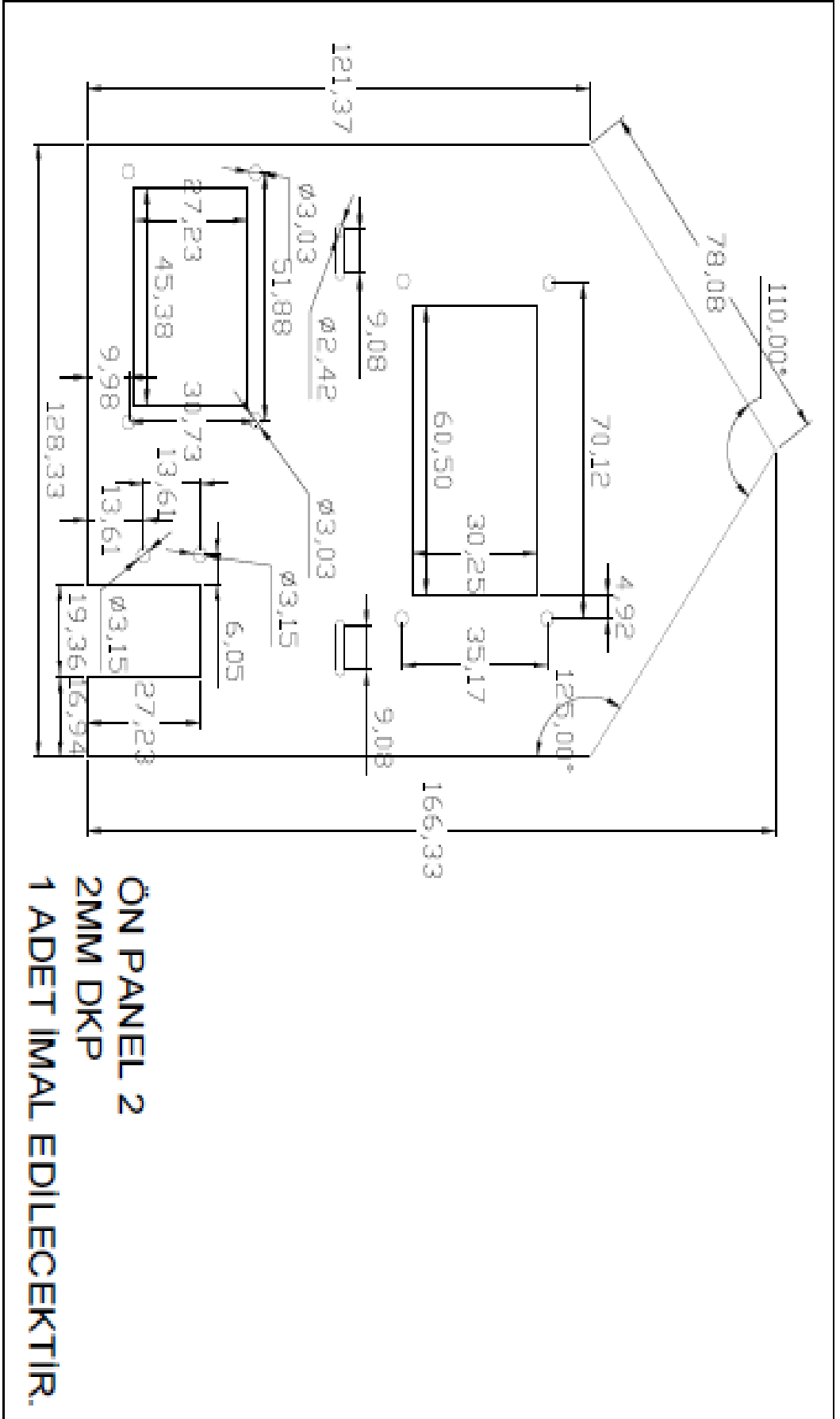


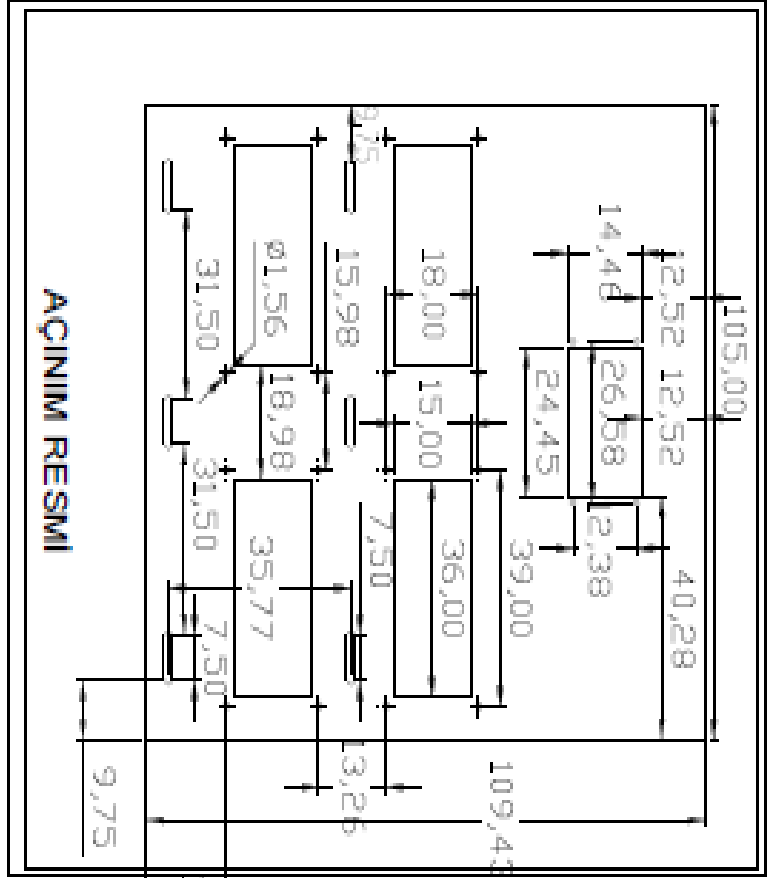
2. KAT DUVAR 2
2MM DKP
1 ADET İMAL EDİLECEKTİR



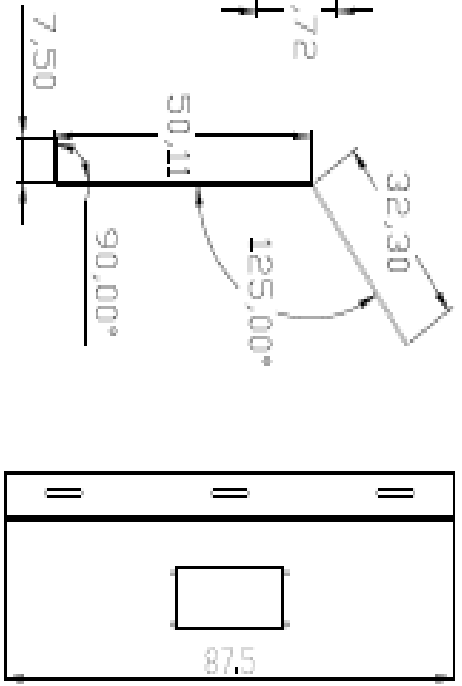
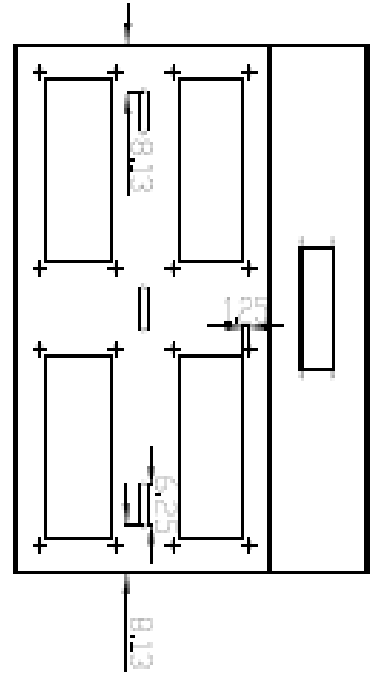


**1.KAT ZEMİNİ
1 ADET İMAL EDİLECEKTİR
İÇERİDEKİ BOKÖM YAPILMA YACAKTIR.**

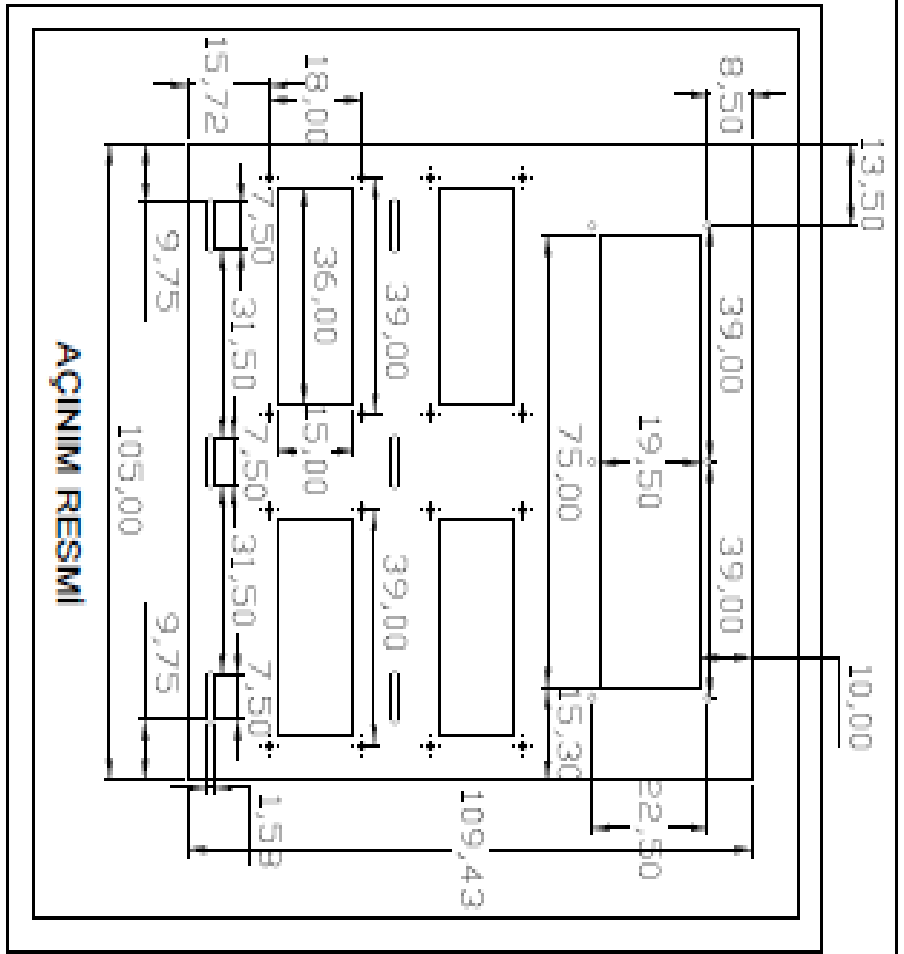




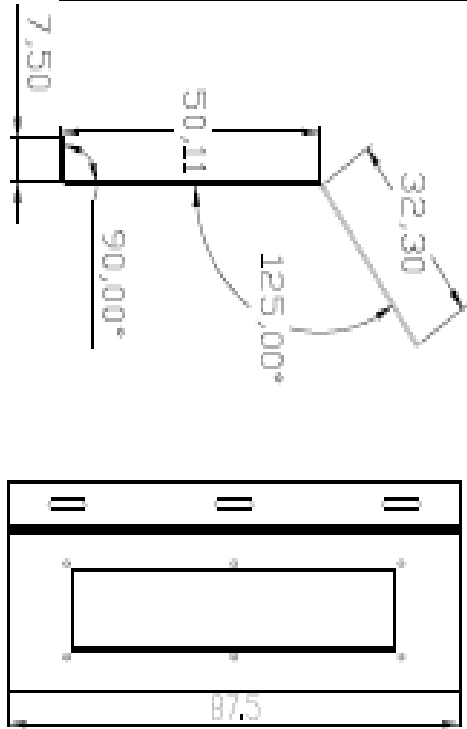
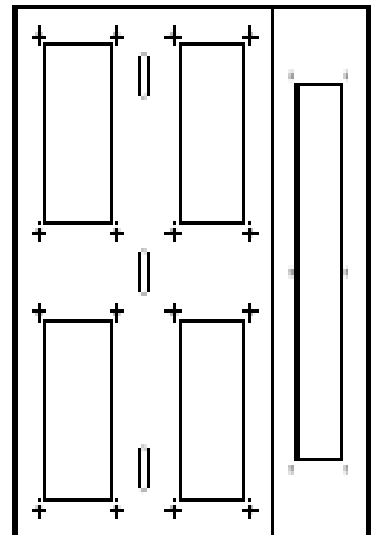
ACINIM RESMI



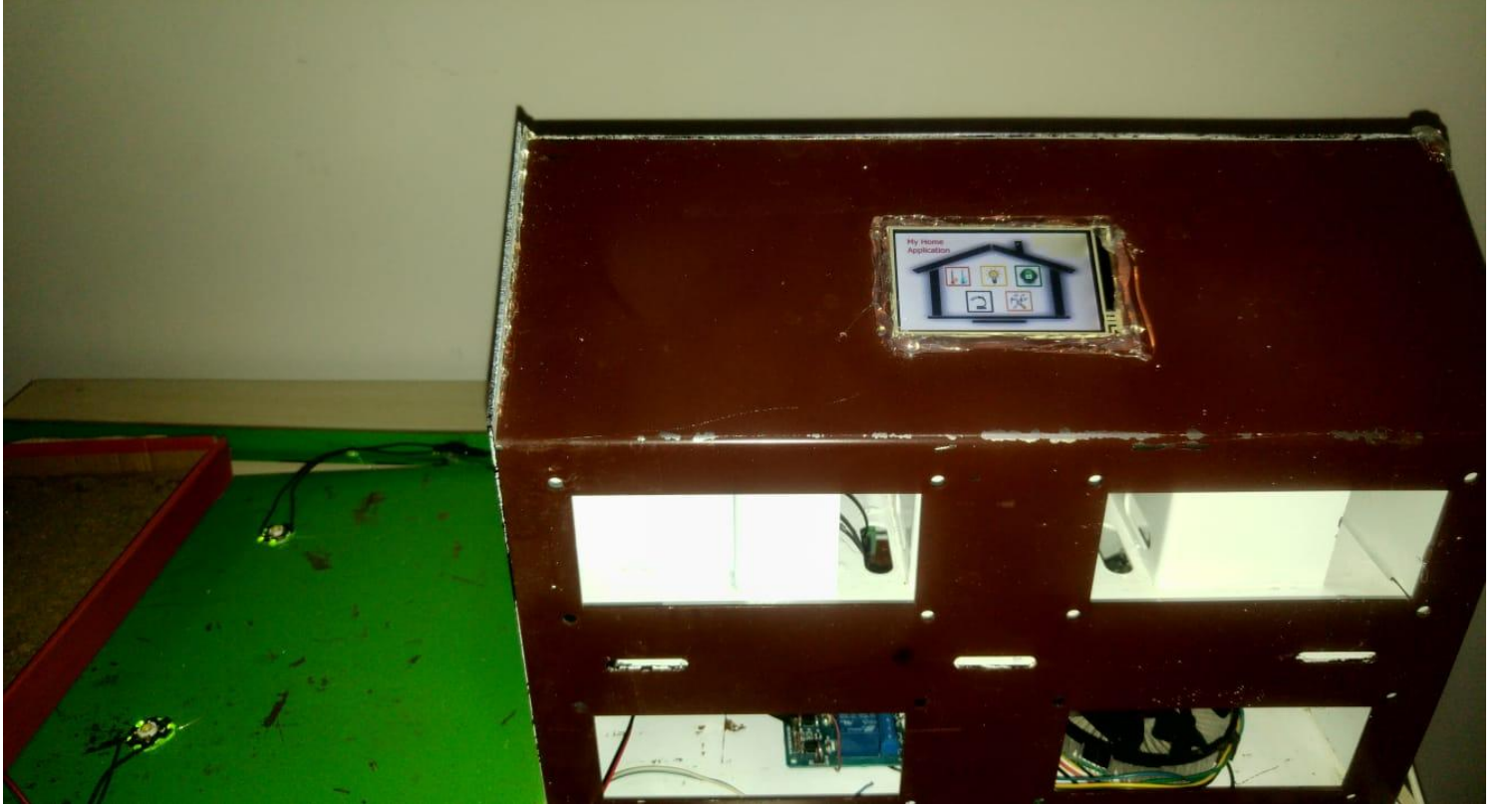
SABIT ÇATI YAN DUVARI 1
 2MM DKP
 1 ADET İMAL EDİLECEKTİR.



SABİT ÇATI YAN DUVARI 2
 2MM DKP
 1 ADET İMAL EDİLECEKTİR.



EK İ. (Ev maketi görselleri)





ÖZGEÇMİŞ

Adı Soyadı : Özen SİNE
Doğum Yeri ve Yılı : Gördes, 1992
Medeni Hali : Bekâr
Yabancı Dili : İngilizce
E-posta : ozensine@icloud.com

Eğitim Durumu

Lise : Demirci Anadolu Lisesi, 2006-2007
Gördes Anadolu Lisesi, 2007-2010
Lisans Bölümü, 2010-2011 : Dumlupınar Üniversitesi, Elektrik Elektronik Mühendisliği
Celal Bayar Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, 2011-2014
Yüksek Lisans Bölümü, 2014-(halen) : Celal Bayar Üniversitesi, Elektrik Elektronik Mühendisliği

Mesleki Deneyim

Kurum bilgisi : Bosch Termoteknik A.Ş. , 2015-2016
Kurum bilgisi : Manisa Büyükşehir Belediyesi, 2016-(halen)

Yayınları