



**T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**OTONOM YÜK TAŞIMA ROBOTU İÇİN BATARYA YÖNETİM SİSTEMİ
DEVRE TASARIMI, YAZILIMI VE BATARYA VERİLERİNİ IOT
TEKNOLOJİSİ İLE İZLEME**

YÜKSEK LİSANS TEZİ

İbrahim Ethem DERE

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektrik-Elektronik Mühendisliği Tezli Yüksek Lisans Programı

OCAK 2025

**T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**OTONOM YÜK TAŞIMA ROBOTU İÇİN BATARYA YÖNETİM SİSTEMİ
DEVRE TASARIMI, YAZILIMI VE BATARYA VERİLERİNİ İOT
TEKNOLOJİSİ İLE İZLEME**

YÜKSEK LİSANS TEZİ

İbrahim Ethem DERE

(21278034035)

ORCID: 0009-0001-5804-0684

**Elektrik-Elektronik Mühendisliği Anabilim Dalı
Elektrik-Elektronik Mühendisliği Tezli Yüksek Lisans Programı**

Danışman: Doç. Dr. Davut ERTEKİN

ORCID: 0000-0003-2234-3453

OCAK 2025



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, Bursa Teknik Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

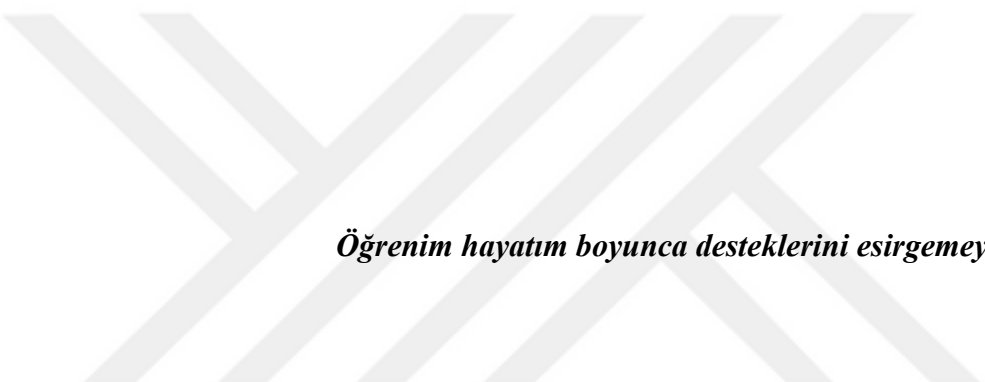
İNTİHAL BEYANI

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı: İbrahim Ethem DERE

İmzası:

[Handwritten signature of İbrahim Ethem DERE]



Öğrenim hayatım boyunca desteklerini esirgemeyen Aileme,

ÖNSÖZ

Otonom Yük Taşıma Robotu İçin Batarya Yönetim Sistemi Devre Tasarımı, Yazılımı ve Batarya Verilerini IoT Teknolojisi ile İzleme adlı tez çalışmamda otonom yük taşıma robotları için geliştirdiğim Batarya Yönetim Sistemi tasarımıyla elde ettiğim bulguları ve geliştirdiğim çözümleri paylaşmaktayım. Çalışmam boyunca projeye yönelik destekleri ve yol gösterici önerileriyle beni yönlendiren danışman hocam Doç. Dr. Davut Ertekin'e teşekkürlerimi sunarım.

Bu çalışmanın hayata geçirilmesinde şirket olanaklarını kullanmama ve deneysel çalışmalar yapmama imkân tanıyan Elfatek Elektronik firmasına ve şirketimde bu süreçte beni destekleyen Süleyman Canan hocama şükranlarımı sunarım. Ayrıca tez çalışmalarım süresince verdikleri destek ve yardımları için değerli çalışma arkadaşlarım Talha Telli ve Muzaffer Yakut 'a teşekkür ederim.

Bu tez süreci boyunca elde ettiğim bilgi ve deneyimlerin hem akademik gelişimime hem de profesyonel kariyerime katkı sağladığını belirtmek isterim. Bu süreçte maddi ve manevi desteklerini her zaman yanımda hissettiğim aileme ve desteğini hiçbir an esirgemeyen değerli eşim Zehra Gül Dere'ye teşekkürlerimi sunarım.

Ocak 2025

İbrahim Ethem DERE
(Elektrik Elektronik Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	viii
KISALTMALAR	x
SEMBOLLER	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xiv
ÖZET.....	xvii
SUMMARY	xix
1. GİRİŞ	1
1.1 Tezin Amacı	2
1.2 Tezin Genel Yapısı.....	3
2. LİTERATÜR ARAŞTIRMASI	4
2.1 Batarya Yönetim Sistemi	4
2.1.1 Pil çeşitleri.....	6
2.1.2 BMS topolojileri	11
2.1.2.1 Merkezi topoloji	12
2.1.2.2 Modüler topoloji	12
2.1.2.3 Dağıtılmış topoloji	13
2.1.3 Batarya yönetim sistemi koruma özellikleri	14
2.1.4 Hücre dengeleme yöntemleri	15
2.1.4.1 Pasif dengeleme	16
2.1.4.2 Aktif dengeleme	17
2.1.5 Durum izleme ve tahmin algoritmaları	18
2.1.5.1 Şarj durumu (SoC) tahmini	21
2.1.5.2 Sağlık durumu (SoH) tahmini	24
2.1.6 BMS için kullanılan entegre devreler (ICs)	26
2.2 Nesnelerin İnterneti (IoT).....	27
3. MATERYAL VE METOTLAR	31
3.1 Tezin Uygulama Kurgusu	31
3.2 Çalışma Yol Haritası Oluşturma	33
3.3 Batarya Yönetim Sistemi Tasarımı Planlaması.....	35
3.3.1 Topoloji seçimi	35
3.3.2 Dengeleme yöntemi seçimi	36
3.3.3 Batarya yönetim sistemi entegresi seçimi	36
3.3.4 Batarya yönetim sistemi ana kontrolcüsü seçimi	38
3.3.5 Batarya yönetim sistemi IoT kontrolcüsü seçimi.....	39
3.3.6 Donanım tasarım programı seçimi	40
3.4 Veri Tabanı Platform Seçimi.....	42
3.5 Bilgisayar ve Telefon Yazılım Platformu Seçimi	43
4. DENEYSEL ÇALIŞMA	45

4.1 Batarya Yönetim Sistemi Tasarımı Deneysel Çalışma	45
4.1.1 Hücre alt gerilim koruması (CUV)	69
4.1.2 Hücre aşırı gerilim koruması (COV)	69
4.1.3 Şarj aşırı akım koruması (OCC).....	70
4.1.4 Deşarj aşırı akım koruması (OCD)	70
4.1.5 Deşarjda kısa devre koruması (SCD).....	70
4.1.6 Şarj korumasında alt sıcaklık (UTC)	71
4.1.7 Deşarj korumasında alt sıcaklık (UTD)	71
4.1.8 Dahili alt limit sıcaklık koruması (UTINT)	72
4.1.9 Dahili aşırı sıcaklık koruması (OTINT).....	72
4.1.10 Şarjda aşırı sıcaklık koruması (OTC).....	72
4.1.11 Deşarjda aşırı sıcaklık koruması (OTD)	72
4.1.12 FET aşırı sıcaklık koruması (OTF)	73
4.1.13 Dengeleme süresi	73
4.1.14 Dengeleme minimum delta değeri	73
4.1.15 Dengeleme minimum hücre voltaj değeri.....	73
4.2 Veritabanı ve IoT Kontrolcü Bağlantısı Deneysel Çalışma	74
4.3 IoT Kontrolcü BMS Veri Akışı Deneysel Çalışma.....	80
4.4 Bilgisayar Yazılımı ve Görsel Tasarım Deneysel Çalışma.....	82
4.5 Telefon Yazılımı ve Görsel Tasarım Deneysel Çalışma.....	101
4.6 Tüm İşlemlerin Uyum İçerisinde Çalışması Deneysel Çalışma.....	103
5. SONUÇ VE ÖNERİLER.....	107
KAYNAKLAR	109
ÖZGEÇMİŞ.....	122

KISALTMALAR

ADC	: Analog Dijital Dönüştürücü (Analog Digital Converter)
BMS	: Batarya Yönetim Sistemi (Battery Management System)
CAN	: Kontrolör Alanı Ağı (Controller Area Network)
CC	: Akım Sayısı (Current Count)
COV	: Hücre aşırı gerilim koruması (Cell Overvoltage Protection)
CUV	: Hücre alt gerilim koruması (Cell Undervoltage Protection)
EA	: Elektrikli Araçlar (EV)
GUI	: Grafik Kullanıcı Arayüzü (Graphical User Interface)
I2C	: Entegre Devreler Arası (Inter-Integrated Circuit)
ICs	: Entegre Devreler (Integrated Circuit)
IoT	: Nesnelerin İnterneti (Internet of Things)
JSON	: Nesne Gösterimi (JavaScript Object Notation)
NTC	: Negatif Sıcaklık Katsayısı (Negative Temperature Coefficient)
OCC	: Şarj aşırı akım koruması (Overcurrent in Charge Protection)
OCD	: Deşarj aşırı akım koruması (Overcurrent in Discharge)
OTC	: Şarjda aşırı sıcaklık koruması (Overtemp in Charge Protection)
OTD	: Deşarjda aşırı sıcaklık koruması (Overtemp in Discharge Protection)
OTF	: FET aşırı sıcaklık koruması (Overtemperature FET Protection)
OTINT	: Dahili aşırı sıcaklık koruması (Internal Overtemperature Protection)
PCB	: Baskılı Devre Kartı (Printed Circuit Board)
SCD	: Deşarjda kısa devre koruması (Short Circuit in Discharge Protection)
SMTP	: Basit Posta Aktarım Protokolü (Simple Mail Transfer Protocol)
SoC	: Şarj Durumu (State of Charge)
SoH	: Sağlık Durumu (State of Health)
SPI	: Seri Çevresel Arayüz (Serial Peripheral Interface)
SSL	: Güvenli Soket Katmanı (Secure Sockets Layer)
UART	: Evrensel Asenkron Alıcı Verici
UTC	: Şarj korumasında alt sıcaklık (Undertemperature in Charge)
UTD	: Deşarj korumasında alt sıcaklık (Undertemperature in Discharge)
UTINT	: Dahili alt limit sıcaklık koruması (Internal Undertemp Protection)

SEMBOLLER

η	: Coulomb verim katsayısını
C_{Th}	: Şarj ve deşarj sırasındaki geçici tepki
C_n	: Bataryanın maksimum enerji depolama kapasitesi
C_{nom}	: Hücrenin nominal kapasitesi
C_{pa}	: Elektrokimyasal polarizasyon kapasitesi
C_{pc}	: Konsantrasyon polarizasyon kapasitesi
C_{total}	: Toplam hücre kapasitesi
E_{cell}	: Hücrenin enerji kapasitesi
I	: Akım
I_L	: Batarya akımı
I_b	: Bataryaya giren veya çıkan akım
$N_{paralel}$	: Paralel bağlı olan hücre sayısı
N_{seri}	: Seri bağlanan hücre sayısı
$Q_{(n)}$	: Nominal kapasite değeri
$Q_{(t)}$	: Bataryanın t anındaki kapasite değeri
R	: Direnç
R_{EOL}	: Bataryanın ömrünün sonunda ölçülen iç direnç
R_s	: Seri direnç
$R_{Termistör}$	: Termistör direnci
R_{Th}	: Polarizasyon direnci
R_{new}	: Bataryanın yeni durumdaki iç direnci
R_{now}	: Bataryanın mevcut iç direnci
R_{pa}	: Elektrokimyasal polarizasyon direnci
R_{pc}	: Konsantrasyon polarizasyon direnci
R_0	: İç direnç
$SoC_{(t)}$	: Belirli bir andaki şarj durumu
$SoC_{(t-1)}$	: Önceki şarj durumu
T	: Sıcaklık
t	: Zaman

U_L	: Batarya terminal gerilimi
U_{Th}	: Kapasitör direnç çifti üzerine düşen gerilim
U_{oc}	: Bataryanın açık devre gerilimi
U_{pa}	: Elektrokimyasal polarizasyon gerilimi
U_{pc}	: Konsantrasyon polarizasyon gerilimi
u_k	: Kontrol sinyali
V	: Gerilim
V_{Ref}	: Referans gerilimi
$V_{Termistör}$	: Termistör gerilimi
V_{nom}	: Hücrenin nominal gerilimi
V_{oc}	: Açık devre gerilimi
V_{total}	: Toplam hücre gerilimi
w_k, v_k	: Gürültü terimleri
w_{k-1}	: İşlem gürültüsü
x_k	: Durum değeri
x_{k-1}	: Bir önceki durum değeri
y_k	: Kesin doğruluğu bilinmeyen değer

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Pil Çeşitleri Karşılaştırılması [21, 31-33].....	7
Çizelge 2.2 : Lityum-iyon pil kimyasallarının karşılaştırılması [24, 36, 44, 53-54].	10
Çizelge 2.3 : Batarya yönetim sistemi entegresi karşılaştırılması [9, 98-112].	26
Çizelge 3.1 : Batarya yönetim sistemi entegre seçim karşılaştırma tablosu.....	37
Çizelge 3.2 : Kontrolcü karşılaştırma tablosu [124-126].	39
Çizelge 3.3 : IoT kontrolcüsü karşılaştırma tablosu [127-131].	40
Çizelge 3.4 : Donanım tasarım programı karşılaştırması [132-136].	41
Çizelge 3.5 : Veritabanı platformu karşılaştırma tablosu [137-142].	42
Çizelge 3.6 : Arayüz platformları karşılaştırması [144-146].....	43
Çizelge 3.7 : Telefon uygulaması tasarlama platform karşılaştırması [147-152].....	44
Çizelge 4.1 : Deşarj akımı zamana bağlı sıcaklık değişimi.	68

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Batarya yönetim sistemi görev diyagramı[16-17].	5
Şekil 2.2 : Pil çeşitlerine göre enerji yoğunluğu.	7
Şekil 2.3 : Pil tipi ve enerji yoğunluğu verimliliği (ağırlık/hacim) [32].	8
Şekil 2.4 : Lityum-iyon batarya ücretleri [55].	11
Şekil 2.5 : Üç farklı BMS topoloji örneği.	11
Şekil 2.6 : Merkezi topoloji örnek diyagram gösterimi.	12
Şekil 2.7 : Modüler topoloji örnek diyagram gösterimi.	13
Şekil 2.8 : Dağıtılmış topoloji örnek diyagram gösterimi.	14
Şekil 2.9 : Örnek batarya hücre dengesizlik durumları [5].	16
Şekil 2.10 : Dengeleme yöntemi çeşitleri [5].	18
Şekil 2.11 : İç direnç eşdeğer devre modeli [94-95].	19
Şekil 2.12 : Thevenin eşdeğer devre modeli [94-95].	20
Şekil 2.13 : Çift polarizasyon eşdeğer devre modeli [94-95].	21
Şekil 2.14 : Şarj durum tahmin yöntemlerinden bazıları [15, 21, 37].	22
Şekil 2.15 : Sağlık durum tahmin yöntemlerinden bazıları [15].	25
Şekil 2.16 : IoT ev uygulamalarının örnek bir diyagramı.	27
Şekil 2.17 : Kontrolcü, çevre birim ve bulut sisteminin temel çalışma diyagramı....	28
Şekil 2.18 : BMS ve IoT özet mimarisi.	29
Şekil 3.1 : AKİBA otonom aracı görseli [121].	31
Şekil 3.2 : Yol haritası diyagram görseli.	34
Şekil 3.3 : Sistemin temel blok diyagramı.	35
Şekil 3.4 : Batarya yönetim sistemi temel yapı diyagramı.	35
Şekil 3.5 : Haberleşme protokolleri güncellenmiş diyagram.	38
Şekil 3.6 : Haberleşme protokolleri güncellenmiş diyagram.	40
Şekil 3.7 : Texas Instruments BQ76952 örnek tasarım [122].	41
Şekil 4.1 : Tüm sistemin genel yapısı görseli.	45
Şekil 4.2 : BQ76952 ve dengeleme yapısı önerilen tasarımın gerçekleşmesi.	46
Şekil 4.3 : Dengeleme tek hücre için örnek.	47
Şekil 4.4 : I2C bağlantı pin header şematik görseli.	48
Şekil 4.5 : Termistör konnektör şematik tasarımı.	48
Şekil 4.6 : Programlama ve haberleşme konnektörü şematik tasarımı.	49
Şekil 4.7 : Batarya yönetim sistemi PCB tasarım görseli.	50
Şekil 4.8 : Batarya yönetim sistemi 3D görseli.	50
Şekil 4.9 : Temel işlemlerin yapılacağı elemanların dizgisi.	51
Şekil 4.10 : STM ve BQ entegresinin haberleşme kontrolü.	54
Şekil 4.11 : Tek bir hücre okuma için oluşturulan kodların bir kısmı.	55
Şekil 4.12 : Tek pil hücresi bağlantısı ve veri okuma işlemi.	55
Şekil 4.13 : Çalışmalar yapılabilmesi için pil hücrelerinden batarya yapımı.	56
Şekil 4.14 : Konnektör ve yalıtkan kağıt ile kaplama.	58
Şekil 4.15 : Oluşturulan bataryanın BMS'e ilk bağlantısı.	59

Şekil 4.16 : Tüm hücre gerilimlerini okuma sorgu işlemleri.....	59
Şekil 4.17 : Tüm hücre gerilimlerini batarya yönetim sistemi tarafından okunması.	60
Şekil 4.18 : Tek bir hücrenin dengeleme işlemi kontrolü.....	60
Şekil 4.19 : Mosfetlerin PCB'ye eklenmiş hali.	61
Şekil 4.20 : Mosfetlerin yazılımsal kontrollerini gerçekleştirilmesi.	62
Şekil 4.21 : Kısa devre testi ve kesim işleminin test cihazıyla incelenmesi.....	63
Şekil 4.22 : Kısa devre testi ve kesim işleminin osiloskopla incelenmesi.	63
Şekil 4.23 : Parametre ayar yazılım yapısının kısa bir kısmı.	64
Şekil 4.24 : Yüksek sıcaklık kesim işleminin osiloskopla incelenmesi.	64
Şekil 4.25 : AKİBA bataryasını test işlemleri için ayarlama.....	65
Şekil 4.26 : Batarya kısa devre test düzeneği.	66
Şekil 4.27 : Uzun süreli deşarj işlemleri test düzeneği.	66
Şekil 4.28 : Deşarj testlerinde çekilen termal kamera görüntüleri.....	67
Şekil 4.29 : Uzun süreli deşarj işlemi görseli.	68
Şekil 4.30 : CUV koruması teknik belge ekran görüntüsü	69
Şekil 4.31 : SCD koruması sabit değerlerinin karşılıkları [155].	71
Şekil 4.32 : ESP32 ve Firebase platform bağlantısı.	74
Şekil 4.33 : Firebase ayarlamalarının yapılması.....	75
Şekil 4.34 : ESP32 internet bağlantısının gerçekleştirilmesi.....	76
Şekil 4.35 : Firebase yapılandırmaları için gerekli yazılımların bir kısmı.	77
Şekil 4.36 : Firebase bağlantısı sağlanması ve seri monitöre bilgi yazdırılması	77
Şekil 4.37 : ESP32 ile Firebaseden veri okuma ve gönderme işlemi.	78
Şekil 4.38 : Tüm batarya verilerinin Firebase ortamına gönderilmesi.	79
Şekil 4.39 : Koruma Parametrelerinin Firebase üzerinden okunması.	79
Şekil 4.40 : BMS ve ESP32 Donanımsal bağlantı.....	80
Şekil 4.41 : STM32 üzerinden UART ile batarya verilerini gönderme.....	81
Şekil 4.42 : STM32'nin gönderdiği verileri okuyan ESP32.	81
Şekil 4.43 : Visual Studio ile yeni proje başlatma.....	82
Şekil 4.44 : Visual Studio genel yapısı ve anlatımlar.....	83
Şekil 4.45 : Firebase Bağlantısı için gerekli eklentinin gerçekleştirilmesi.....	83
Şekil 4.46 : Firebase tanıtılma işlemleri için kütüphane eklemeleri.....	84
Şekil 4.47 : Firebase bilgilerinin programa tanıtılması.	84
Şekil 4.48 : Firebase tarafından okunan değerler için değişken oluşturma işlemi. ...	85
Şekil 4.49 : Firebase hücre-1 değeri okuma ve durum çubuğunda görselleştirme....	85
Şekil 4.50 : Hücre değerinin durum çubuğunda gösterilmesi.	86
Şekil 4.51 : Durum çubuğunu batarya simgesi şekline dönüştürme.....	87
Şekil 4.52 : Firebase veri gönderme fonksiyonu yazılımı.	87
Şekil 4.53 : Firebase platformuna veri gönderme ve alma işlemi.	88
Şekil 4.54 : Mail gönderme için yazılım yapılarının oluşturulması.	88
Şekil 4.55 : Mail gönderme yapısı.	89
Şekil 4.56 : Mail gönderme yazılım temel yapısı.	90
Şekil 4.57 : İki Hücre veri okuma yapısı.	90
Şekil 4.58 : Tüm batarya hücrelerinin eklenmiş hali.	91
Şekil 4.59 : Hücre fark değeri hesaplama.	91
Şekil 4.60 : Görselleştirilmiş fark mV değerleri.....	92
Şekil 4.61 : Batarya SoC ve sıcaklık değer gösterimi.	92
Şekil 4.62 : Görsel geliştirmeler yapılmış SoC değeri.	92
Şekil 4.63 : İşlemlerin birleştirilmiş yapısı görseli.....	93
Şekil 4.64 : Mail ve parametre ayarlar butonları görüntüsü.	93
Şekil 4.65 : Buton aracılığı ile batarya verilerini gönderme işlemi.....	94

Şekil 4.66 : Görsel geliřtirmeler yapılmıř ana ekran.....	94
Şekil 4.67 : Parametre ayar ve bilgilendirme görseli.....	95
Şekil 4.68 : Hesaplamalar ile parametre değeri belirleme.....	96
Şekil 4.69 : Parametre ayarlarının kayıt hesaplamalar ile kayıt iřlemi.....	96
Şekil 4.70 : SCD koruması için seçim yapılabilen bölüm.....	96
Şekil 4.71 : Otomatik sıcaklık uyarısı e-posta gönderimi.....	97
Şekil 4.72 : Hücre gerilim farkı çok olursa otomatik e-posta gönderim iřlemi.....	98
Şekil 4.73 : Parametre ayarlar menüsü tam görünümü.....	98
Şekil 4.74 : Toplu hata kayıt kod yapısı bir kısmı.....	99
Şekil 4.75 : Toplu hata kayıt çıktı örneđi	99
Şekil 4.76 : Şifre ile sisteme giriř yapısının temel görseli.....	100
Şekil 4.77 : Görselleřtirilmiř giriř sayfası.....	100
Şekil 4.78 : Hangi e-posta ile devam edileceđinin seçim menüsü.....	101
Şekil 4.79 : MIT APP programı oluřturma ařamaları.....	102
Şekil 4.80 : Mobil uygulama ile batarya verilerinin takibi.....	103
Şekil 4.81 : BMS, batarya ve ESP32 bađlantısı.....	104
Şekil 4.82 : Telefon ve bilgisayardan verileri izleme.....	104
Şekil 4.83 : Deneysel çalıřma son hali görseli.....	105
Şekil 4.84 : AKİBA ve tüm sistem çalıřma görseli.....	106

OTONOM YÜK TAŞIMA ROBOTU İÇİN BATARYA YÖNETİM SİSTEMİ DEVRE TASARIMI, YAZILIMI VE BATARYA VERİLERİNİ IOT TEKNOLOJİSİ İLE İZLEME

ÖZET

Bu tez çalışması otonom yük taşıma robotları için güvenilir, verimli ve uzaktan izlenebilir bir batarya yönetim sistemi tasarımı hedeflemektedir. Otonom araçlar son yıllarda endüstriyel ve ticari uygulamalarda yaygın olarak kullanılmaya başlanmış ve bu araçların performansı büyük ölçüde batarya sistemlerine bağlı hale gelmiştir. Lityum-iyon bataryaların kullanıldığı bu sistemlerde batarya sağlığının izlenmesi, şarj-deşarj işlemlerinin güvenli bir şekilde yönetilmesi ve hücre dengesinin sağlanması kritik öneme sahiptir. Bu ihtiyaç doğrultusunda geliştirilen batarya yönetim sistemleri bataryaların ömrünü uzatmakta, enerji verimliliğini artırmakta ve güvenli kullanım sağlamaktadır.

Tez çalışmasında otonom yük taşıma araçları için yenilikçi bir batarya yönetim sistemi geliştirilmiştir. Bu sistem hem bataryaların güvenliğini sağlamakta hem de kullanıcıların batarya verilerini anlık olarak izleyebilmesini mümkün kılmaktadır. Batarya yönetim sistemi Texas Instruments firmasının BQ76952 entegresi kullanılarak tasarlanmıştır. Bu entegre sayesinde bataryanın gerilim, akım ve sıcaklık değerleri hassas bir şekilde ölçülmüştür. STM32G431 mikrodenetleyicisi bu verileri işleyerek gerekli dengeleme ve güvenlik işlemlerini gerçekleştirmiştir. Sistem IoT tabanlı bir izleme yapısına sahip olup, ESP32 modülü üzerinden Firebase platformuna bağlanarak bulut tabanlı veri aktarımı sağlanmıştır. Bu sayede kullanıcılar batarya verilerini uzaktan takip edebilmekte ve kritik durumlarda hızlıca müdahale edebilmektedir.

Yazılım geliştirme aşamasında masaüstü uygulama için “Visual Studio”, mobil uygulama için ise “MIT App Inventor” kullanılmıştır. Uygulamalar batarya verilerini kullanıcı dostu bir arayüzde sunmakta, ilerleme çubuğu (Progressbar) ve grafiksel göstergelerle bataryanın şarj durumu, hücre gerilimleri ve sıcaklık değerleri anlaşılır bir şekilde izlenebilmektedir. Ayrıca uygulama üzerinden elde edilen veriler istenildiğinde ek bir çalışma ile bulut ortamında saklanarak uzun dönem analiz yapılmasına olanak tanıyabilir.

Dengeleme, kısa devre koruma ve sıcaklık izleme testlerinde başarılı sonuçlar elde edilmiştir. Termal kamera ile yapılan sıcaklık testlerinde, bataryanın uzun süreli kullanımda dahi aşırı ısınma yapmadığı ve sistemin stabil çalıştığı gözlemlenmiştir. Özellikle dengeleme direncinin sıcaklık artışının 81°C'yi geçmediği ve kritik sıcaklık limitlerinde enerji akışının otomatik olarak kesildiği tespit edilmiştir. Bu çalışma otonom yük taşıma araçları için tasarlanan yenilikçi bir batarya yönetim sistemi sunmaktadır. Sistemin bulut tabanlı izleme ve kontrol özelliği kullanıcıların batarya verilerini anlık olarak takip etmesini sağlamaktadır. Ayrıca standart batarya yönetim sistemlerinde bulunan kritik sıcaklık ve gerilim seviyelerine ulaşıldığında otomatik olarak müdahale edilmesi gibi korumaları da içermesi bataryaların güvenli kullanımını artırmaktadır.

Bu yenilikçi yaklaşım ile otonom sistemlerde batarya yönetimi konusunda önemli bir etken olarak sadece kontrolcünün değil kullanıcının da sisteme uzaktan erişerek gerekli aksiyonları almasına olanak tanır. Bu sayede karar mekanizması sadece kontrolcüye bırakılmayarak ek bir önlem alınabilecek sistem tasarlanmıştır.

Anahtar kelimeler: Batarya yönetim sistemi (BMS), Nesnelerin interneti (IoT), Gerçek zamanlı izleme, Akıllı batarya sistemi



BATTERY MANAGEMENT SYSTEM CIRCUIT DESIGN, SOFTWARE, AND MONITORING BATTERY DATA USING IOT TECHNOLOGY FOR AN AUTONOMOUS LOAD-CARRYING ROBOT

SUMMARY

This thesis aims to design a reliable, efficient, and remotely monitorable battery management system for autonomous load-carrying robots. In recent years, autonomous vehicles have been widely used in industrial and commercial applications, and their performance has become highly dependent on battery systems. In these systems utilizing lithium-ion batteries, monitoring battery health, managing charging and discharging processes safely, and ensuring cell balancing are critically important. Battery management systems developed in line with this need extend the life of batteries, increase energy efficiency, and ensure safe use.

In this thesis, an innovative battery management system has been developed for autonomous load-carrying vehicles. This system both ensures battery safety while allowing users to monitor battery data momentarily. The battery management system is designed using the BQ76952 integrated circuit from Texas Instruments company. Through this integration, the voltage, current, and temperature values of the battery were measured with precision. The STM32G431 microcontroller processed this data and performed the necessary connections and security operations. The system has an IoT-based monitoring structure, and cloud-based data transfer is provided by connecting to the Firebase platform via the ESP32 module. In this way, users can remotely monitor battery data and intervene quickly in critical situations.

During the software development phase, “Visual Studio” was used for the desktop application and “MIT App Inventor” was used for the mobile application. The applications present battery data in a user-friendly interface, and the battery's charge status, cell voltages and temperature values can be monitored in an understandable way with the progress bar and graphical indicators. In addition, the data obtained through the application can be stored in the cloud environment with an additional study when requested, allowing long-term analysis.

Successful results were obtained in balancing, short circuit protection and temperature monitoring tests. In temperature tests conducted with a thermal camera, it was observed that the battery did not overheat even in long-term use and the system operated stably. Specifically, the balancing resistor's temperature increase did not exceed 81°C, and energy flow was automatically interrupted at critical temperature limits. This study presents an innovative battery management system designed for autonomous load-carrying vehicles. The system's cloud-based monitoring and control features allow users to track battery data in real time. In addition, the fact that it includes protections such as automatic intervention when critical temperature and voltage levels are reached in standard battery management systems increases the safe use of batteries.

This innovative approach, which is an important factor in battery management in autonomous systems, allows not only the controller but also the user to remotely access the system and take the necessary actions. In this way, the decision mechanism is not left to the controller alone, and a system was designed that can take additional precautions.

Keywords: Battery Management System (BMS), Internet of Things (IoT), Real-Time Monitoring, Smart Battery System



1. GİRİŞ

Teknolojide söz sahibi olmaya başlamış yapay zekâ teknolojisi son zamanlarda otonom cihazların yaygınlaşmasına önemli bir rol almıştır. Üretilen birçok otonom cihaz batarya sistemleriyle beraber çalışmaktadır. Otonom araç teknolojileri endüstri ve ticarete hızla gelişerek yaygın olarak kullanılmaya başlanmıştır. Bu otonom araçların teknolojileri genellikle elektrikli olarak geliştirilmektedir. Elektrik temelli otonom araç tasarımları bu araçların performanslarının kullanılan batarya sistemlerinin etkinliğine doğrudan bağlı olmasına neden olmaktadır. Otonom cihazların batarya ile çalışma durumu otonom araçların batarya sistemlerinin daha güvenli ve daha verimli şekilde tasarlanması durumunu destekler nitelikte olmuştur. Bunların yanında batarya sistemlerinin yönetim biriminin hem gelişmesini hem de yaygınlığını arttırmıştır [1-3].

Otonom araçların batarya sistemleri güvenlik açısından büyük önem taşımaktadır. Bu sebeple batarya yönetim sistemlerinin gelişimi otonom araçların güvenilir çalışmasını sağlayacak olmasından dolayı otonom araçların temel unsurlardan biri haline gelmiştir. Bu noktalardaki gelişmeler batarya yönetim sistemlerinin önemini gün geçtikçe arttırmaktadır. Özellikle enerji verimliliği ve batarya ömrünün uzatılması gibi konularda yapılan araştırmalar otonom araçları geliştirdiği gibi batarya sistemlerinin de geliştirilmesine katkıda bulunmaktadır [1, 4].

Batarya yönetim sistemleri otonom cihazların güvenilirliği ve kullanım süresi açısından kritik bir role sahiptir. Bu sistemlerin sürekli olarak geliştirilmesi takip edilmesi ve uyarı sistemlerinin efektif şekilde çalışması için yapılan birçok araştırma vardır ve bu geliştirmeler otonom araçlar yaygınlaştıkça gün geçtikçe artarak devam edecektir [1].

Batarya sistemlerinde hücreler arasında üretim farklılıkları nedeniyle gerilim dengesizlikleri oluşabilir. Bu durum bataryaların tam kapasitede kullanılmasını engelleyerek sistem performansını düşürür. Bu gerilim farklılıkları batarya yönetim sistemlerinin gerekliliğini büyük bir ihtiyaç olarak ortaya çıkarmaktadır batarya

yönetim sistemlerinin sunduğu imkanlar olan pasif ve aktif dengeleme yöntemleri sayesinde bu gerilim farklılıkları sorunlarının önüne geçmek mümkündür. Pasif dengeleme hücredeki fazla enerjiyi ısıya dönüştürürken aktif dengeleme enerjiyi düşük enerjili hücrelere transfer eder. Bu yöntemlerin ikisi de bataryaların verimli ve uzun ömürlü çalışmasını sağlarlar [5-6].

Bir Batarya Yönetim Sistemi batarya paketindeki hücrelerin izlenmesi, kontrol edilmesi ve yönetilmesi için kullanılan bir sistemdir. Batarya yönetim sistemleri hücrelerin şarj durumunu, sıcaklığını, gerilimini ve diğer önemli parametrelerini sürekli okuyan bir yapı içerir. Hücre voltajı, akım seviyeleri ve sıcaklık gibi kritik parametreleri izlemesinin yanında güvenlik mekanizmalarının da çalışır hale getirilmesinde aktif olarak rol oynar. Örneğin olumsuz şartlarda oluşabilecek aşırı sıcaklık, yüksek gerilim veya kısa devre durumlarında enerji akışını keserek sistemin güvenliğini sağlar.

1.1 Tezin Amacı

Bu tezin amacı standart bir batarya yönetim sisteminde olması gereken her şeyin yanında batarya yönetim sistemine yenilikçi yetenekler kazandırmaktır. Bu çalışma ile otonom yük taşıma robotları için batarya yönetim sistemlerinin uyarı teknolojileri, bataryaların durum takip işlemleri ve uzaktan kontrol işlemlerinin geliştirilmesi amaçlanmıştır. Bu çalışma ile otonom bir yük taşıma aracının batarya verilerini Nesnelerin interneti (IoT-Internet of Things) teknolojisi ile istenildiği yerden istenildiği zaman izlenmesini ve kontrol edilmesi sağlanacaktır. Kullanıcı isteğine göre şekillenecek olan takip işleminin yapılmasının yanında bir diğer amacı belirlenen güvenlik sınırlarının aşılması durumunda uyarı sistemlerini aktif ederek bu uyarı sistemleri ile kullanıcıya e-posta gönderme işlemiyle bilgilendirme yapılmasıdır.

Bu çalışma otonom yük taşıma araçlarında kullanılan bataryaların yönetimi, verilerinin izlenmesi ve kontrolü üzerine odaklanmaktadır. Tez kapsamında otonom yük taşıma araçları için özgün bir Batarya Yönetim Sistemi tasarımı, veri izleme için kart tasarımı ve bu sistemle uyumlu Batarya Yönetim Sistemi yazılımı, veri izleme yazılımı ve bu yazılım içerisinde kontrol birimleri geliştirilecektir. Otonom yük taşıma araçlarında kullanılan bataryaların güvenliğini ve performansını artırmak için yeni bir yaklaşım sunuldu.

1.2 Tezin Genel Yapısı

Bu çalışmada özellikle elektrikli araçlarda en sık kullanılan lityum iyon bataryaların dengeleme ve güvenlik önlemleri üzerinde durulmuştur. Bu bataryaların hücreler arası dengesizliğini azaltmak için uygun bir batarya yönetim sistemi tasarımı araştırıldı ve geliştirildi.

Batarya yönetim sistemlerinde kullanılan hem kontrolcü entegreler karşılaştırılıp uygun olanın seçileceği hem de batarya yönetim sistemlerinde kullanılan teknik (Aktif, pasif dengeleme) ve topolojilerin (Merkezi Topoloji, Modüler Topoloji, Dağıtılmış Topoloji vb.) karşılaştırılması ile daha detaylı bir sonuç elde edilerek Batarya yönetim sisteminin fiyat performans ölçütü göz önünde bulunduruldu. Bu sayede tasarım kararları daha net bir şekilde verildi. Bu farklı çeşitlerdeki topoloji ve dengeleme sistemlerine detaylı olarak literatür araştırması başlığı altında değinildi [7].

Kısaca bu çalışma standart bir batarya yönetim sisteminin yaptığı görevlerin yanında durumla alakalı bilgilendirmeleri kullanıcıya e-posta olarak bildirildi. Bu sayede uzaktan parametre kontrolleri sağlayarak cihazın güvenlik seviyesini bir derece daha arttırmak hedeflendi. Kullanıcılar otonom araçlarının batarya verilerini tasarlanacak olan bilgisayar programı ve telefon uygulaması ile anlık olarak takip edebilecektir. Bu sayede otonom yük taşıma araçlarında kullanılan bataryaların güvenliğini ve performansını artırmak için yeni bir yaklaşım amaçlandı. Yapılan bu çalışma sayesinde otonom araçlar için modern bir batarya yönetim sistemi tasarımını sunarak güvenli bir batarya yönetimi sağlayan bir çalışma yapıldı.

2. LİTERATÜR ARAŞTIRMASI

Literatür araştırması başlığında tez içeriğinde kullanılan konuların teorik bilgilerine yer verildi. Bu bağlamda pil çeşitleri, batarya yönetim sistemi topolojileri, dengeleme sistemleri ve tahmin algoritmaları gibi temel yapılar anlatıldı. Ayrıca tezin yenilikçi yaklaşım teknolojisi olan nesnelerin interneti (IoT) hakkında temel açıklamalar yapıldı.

2.1 Batarya Yönetim Sistemi

Batarya Yönetim Sistemleri (Battery Management Systems-BMS), bataryaların daha uzun ömürlü olacak bir şekilde kullanılması ve bataryaların güvenliğinin sağlanması için gerekli elektronik sistemlerdir. Bu sistemlerin tasarım yapıları hücre sayısına ve batarya paketinin şekilsel yapısına göre değişiklik göstermektedir.

İyi tasarlanmış bir batarya yönetim sistemi bataryaların uzun ömürlü, güvenli ve verimli bir şekilde kullanılmasını sağlar. Batarya Yönetim Sistemlerinin temel özellikleri arasında hücre voltajı ölçümü, batarya grubu voltaj ölçümü, sıcaklık ölçümü, akım ölçümü, kullanım senaryosuna göre çeşitli koruma devreleri ve hücre voltajı dengelemesi yer almaktadır [6, 8-9].

Batarya Yönetim Sistemleri batarya işlemlerinin güvenliğini ve işlevselliğini korumada çok önemli bir rol oynar. Örneğin aşırı şarj, aşırı deşarj, aşırı sıcaklık ve kısa devre gibi durumlarda yükü veya şarj cihazını kesmesi gereken bu sistem bataryanın sağlam kalması için oldukça önemli bir faktördür [10-11].

Seri bağlı hücreler arasındaki üretim farklılıklarından dolayı, şarj sırasında hücreler farklı voltaj seviyelerine ulaşabilir ve bu durum, bataryanın optimum şekilde kullanılmasını engelleyebilir. Bazı hücreler daha düşük voltaj seviyelerinde kalırken, diğerleri daha yüksek voltajlarda kalır. Bu sorunu çözmek için batarya yönetim sistemleri kullanılır [5-6, 8, 10-13].

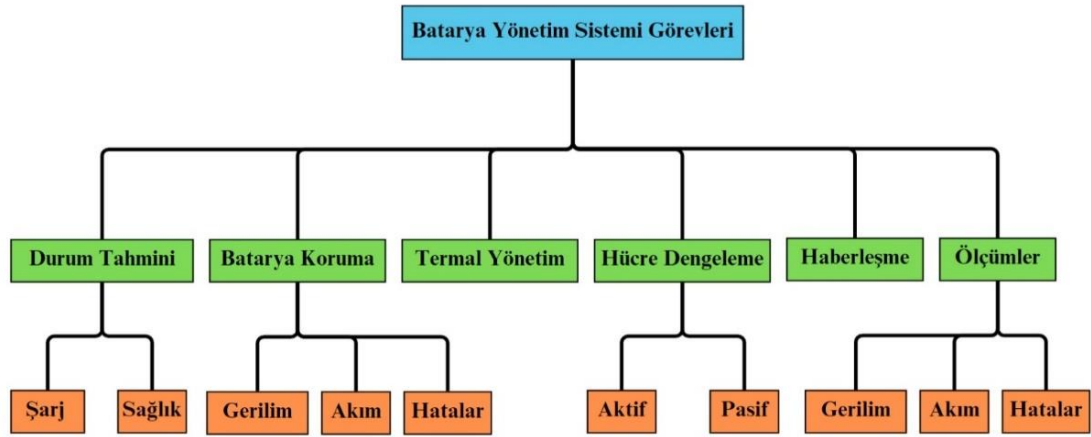
Şarj Durumu (State of Charge-SoC) bataryanın mevcut olarak ne kadar kapasite depolayabildiğini tam şarj kapasitesine göre ifade eder. SoC tahmini için birçok yöntem bulunmaktadır, ilerleyen başlıklarda bu bilgiler detaylandırılacaktır [6, 14-15].

Temel yapıdaki bu görevlerin yani durum tahmini, batarya koruma, termal yönetim, hücre dengeleme, haberleşme ve ölçüm işlerinin özet halinde oluşturulmuş diyagramına Şekil 2.1’de gösterilmiştir [16].

Lityum bazlı bataryalar en küçük elektronik cihazlardan araçlara kadar günümüzde yaygın olarak kullanılmaktadır.

Bunun başlıca nedenleri yüksek çevrim ömrü, kompakt boyutları, yüksek enerji yoğunluğu ve yüksek akımlı uygulamalarda gerekli akımı sağlayabilme yetenekleridir.

Tezin ilerleyen bölümlerinde bu bilgiler detaylandırıldı [9].



Şekil 2.1 : Batarya yönetim sistemi görev diyagramı[16-17].

Batarya yönetim sistemi enerji depolama sistemlerinin güvenli, verimli ve uzun ömürlü bir şekilde çalışmasını sağlamak için birçok önemli görevi yerine getirir. Şekil 2.1’de görüldüğü üzere bu görevler genel olarak durum tahmini, batarya koruma, termal yönetim, hücre dengeleme, haberleşme ve ölçümler başlıkları altında toplanır [16-17].

Kablosuz iletişim modülleri ve bulut bağlantılı sensörlerden yararlanılarak veriler kapsamlı analiz için merkezi platformlara iletilir [1-2, 4]. Bu özet olarak anlatılanları birkaç başlık altında detaylandırmak ve tez içeriğinin kavranmasını kolaylaştıracaktır.

2.1.1 Pil çeşitleri

Elektrikli araçlar üzerinde yapılan çalışmalar büyük bir hızla devam etmektedir ve bu araçların en kritik bileşeni hiç kuşkusuz bataryalardır. Bataryalar bir aracın gücünü hızını ve menzilini doğrudan etkileyen en önemli unsurlardan biri olmasının yanı sıra oldukça maliyetlidir. Bu yüksek maliyetler göz önüne alındığında bataryaların uzun ömürlü olması ve arıza riskinin minimize edilmesi büyük bir önem taşımaktadır [18-20].

Bataryaların kullanım ömrü şarj edilip sonrasında deşarj edildikçe ve bu işlem sürekli yapıldığında kapasitesinin azalması gibi doğal süreçler oluşur. Ayrıca bataryaların hızlı bir şekilde şarj edilebilmesi de kullanıcılar için büyük bir avantaj olarak değerlendirilir. Bu tür beklentiler doğrultusunda elektrikli araçlarda kullanılan batarya teknolojileri yıllar içinde önemli ölçüde gelişmiştir ve gelişmeye devam etmektedir. [1, 8, 17, 21-26].

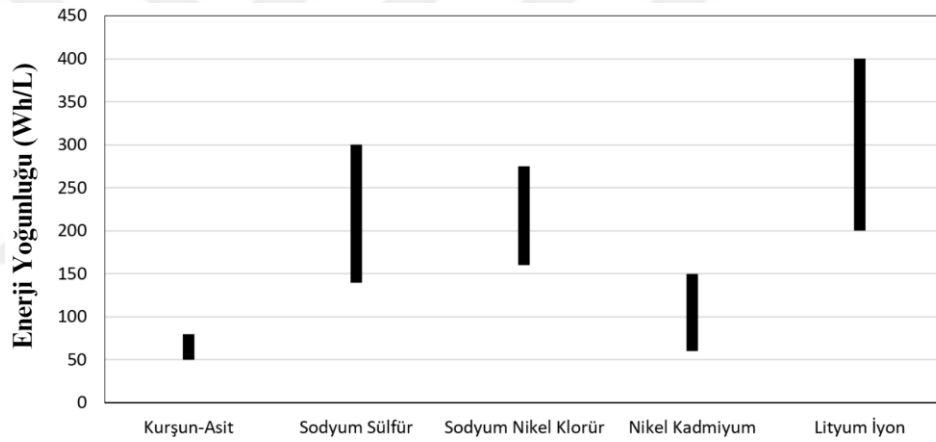
Bataryalar günümüz teknolojisinde önemli bir yer tutan enerji depolama sistemleridir. Elektrikli araçlardan taşınabilir cihazlara enerji depolama sistemlerinden yenilenebilir enerji sistemlerine kadar birçok farklı uygulamada kullanılırlar. Ancak her batarya türü belirli avantajlar ve dezavantajlar sunar. Bu nedenle batarya türlerinin seçimi kullanıldığı uygulamaya ve gereksinimlere göre değişir. Bu alanda yaygın olarak kullanılan birçok batarya türü bulunmaktadır. Bunlar arasında Sodyum Nikel Klorür (NaNiCl), kurşun-asit (Pb-acid), Lityum iyon (Li-ion), Nikel Kadmiyum (NiCd) ve Sodyum Sülfür (NaS) yer alır. Batarya verimliliği, maliyet, güvenlik ve çalışma döngüsü gibi önemli özellikler göz önünde bulundurulduğunda elektrikli araçlar için olumlu bir değerlendirme yapmak zordur [27-31].

Lityum sülfür (Li-S), erimiş tuz (Na-NiCl₂), nikel-metal hidrit (Ni-MH) ve lityum iyon (Li-Ion) olmak üzere dört farklı batarya türü aynı elektrik enerjisi depolama kapasitesine sahiptir [32]. Ayrıca Kurşun-asit, Nikel-Kadmiyum ve nikel-metal hidrit bataryalar detaylı bir şekilde açıklanmış ve bu bataryalar arasındaki karşılaştırmalar yapılmıştır [31]. Lityum iyon bataryalar yüksek verimlilik, yüksek enerji yoğunluğu, yüksek güvenilirlik, yüksek güç yoğunluğu, uzun ömür ve düşük deşarj oranı gibi avantajlar sunar [33]. Çizelge 2.1’de bu batarya çeşitlerinin bir karşılaştırması yapılmıştır. Ayrıca bu karşılaştırılan pil çeşitlerinin enerji yoğunluğunun grafiksel görseli Şekil 2.2’den incelenebilir.

Çizelge 2.1 : Pil Çeşitleri Karşılaştırılması [21, 31-33].

Pil Tipi	Enerji Yoğunluğu (Wh/L)	Nominal Gerilim (V)	Yaşam Döngüsü	Batarya Verimliliği
Kurşun-Asit	50-80	2.0	1500	%82
Sodyum Sülfür	140-300	2.08	5000	%80
Sodyum Nikel Klorür	160-275	-	3000	%84
Nikel Kadmiyum	60-150	1.3	2500	%83
Lityum İyon	200-400	4.3	10000	%95

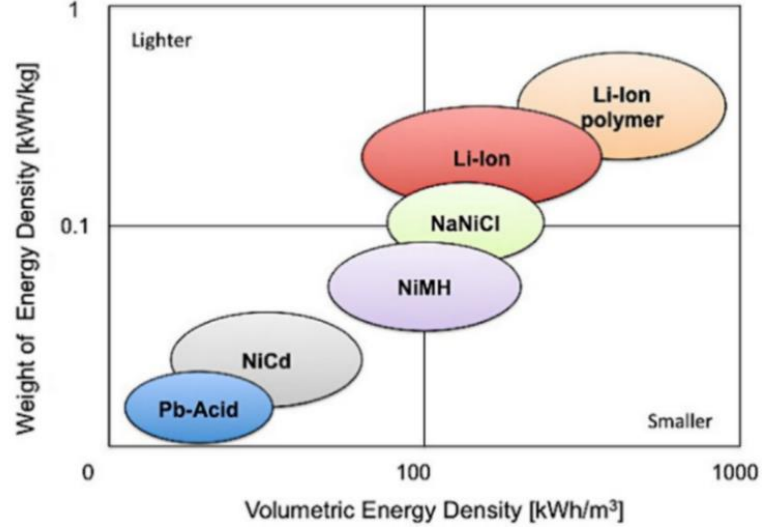
Lityum-iyon bataryalar günümüzde en yaygın kullanılan batarya türlerinden biridir. Elektrikli araçlardan taşınabilir cihazlara kadar geniş bir kullanım alanına sahip olan Lityum-iyon bataryalar yüksek enerji yoğunluğu, uzun ömürleri, hızlı şarj olma kapasiteleri ve düşük bakım gereksinimleri ile öne çıkar.



Şekil 2.2 : Pil çeşitlerine göre enerji yoğunluğu.

Lityum-iyon bataryaların en büyük avantajlarından biri, yüksek enerji yoğunluğuna sahip olmalarıdır. Bu özellik, bataryaların daha küçük boyutlarla daha fazla enerji depolamalarına olanak tanır. Bunun yanında hızlı şarj edilebilir olmaları taşınabilir cihazlar ve elektrikli araçlar gibi uygulamalar için ideal hale getirir. Şekil 2.3'te bu pil çeşitlerinin pil tipi ve enerji yoğunluğu verimliliği (ağırlık/hacim) olarak sınıflandırılması gösterilmiştir.

LiFePO₄ bataryalar lityum-iyon bataryalarının bir türüdür ancak özellikle güvenlik, uzun ömür ve çevre dostu özellikleri ile öne çıkar. Bu bataryalar lityum-iyon teknolojisinin gelişmiş bir versiyonudur ve genellikle daha yüksek güvenlik gerektiren uygulamalarda tercih edilir. LiFePO₄ bataryaların en önemli avantajlarından biri yüksek güvenlik özellikleridir.



Şekil 2.3 : Pil tipi ve enerji yoğunluğu verimliliği (ağırlık/hacim) [32].

Termal ve elektriksel güvenlik açısından diğer lityum-iyon bataryalardan daha iyidir. Ayrıca, bu bataryalar çevre dostu malzemelerle üretilmiştir ve toksik içeriklere sahip değildir. Ancak, LiFePO_4 bataryaların enerji yoğunluğu, diğer lityum-iyon bataryalardan daha düşüktür ve bu da performans açısından bir sınırlama getirir. LiFePO_4 bataryalar, genellikle elektrikli araçlar ve enerji depolama sistemlerinde tercih edilir. Elektrikli araçlar için sağladığı uzun ömür ve güvenlik, bu bataryaların popülerliğini artırmıştır [34-44].

Kurşun-asit (Pb-acid) bataryalar tarihsel olarak en eski batarya türlerinden biri olup hala birçok endüstriyel uygulamada kullanılmaktadır. Özellikle düşük maliyetleri ile bilinir. Ancak bu bataryaların enerji yoğunluğu lityum-iyon bataryalar gibi modern bataryalarla karşılaştırıldığında çok daha düşüktür. Kurşun-asit bataryaların en büyük avantajı düşük maliyetli olmalarıdır.

Bu bataryalar genellikle enerji depolama sistemlerinde ve araçlarda marş sistemleri için kullanılır. Bununla birlikte kurşun-asit bataryaların ömrü lityum-iyon bataryalara kıyasla daha kısa olup aynı zamanda düzenli bakım gerektirirler. Kurşun-asit bataryaların dezavantajları arasında düşük enerji yoğunluğu ve sınırlı ömürleri bulunur. Bu bataryalar genellikle daha büyük ve ağır sistemlerde kullanılır. Bu bağlamda küçük boyuttaki taşınabilir cihazlar için uygun değildir [41, 43, 45-50].

Nikel Kadmiyum (NiCd) bataryalar eski taşınabilir cihazlar ve endüstriyel uygulamalar için yaygın olarak kullanılmıştır. Yüksek akım taşıma kapasitesine sahip olmalarına rağmen çevre üzerindeki olumsuz etkileri nedeniyle günümüzde daha çevre

dostu alternatiflerle yer değiştirilmiştir. NiCd bataryalar, özellikle yüksek akım gereksinimi duyan uygulamalar için uygun olabilir. Bununla birlikte, NiCd bataryalar, çevreye zarar veren kadmiyum içerdiği için çevresel etkiler açısından zararlıdır. Ayrıca, bu bataryaların hafıza etkisi vardır, yani kısmi şarjlarla kullanıldıklarında kapasite kaybı yaşarlar [21, 28, 36].

Sodyum-sülfür (NaS) bataryalar, özellikle büyük ölçekli enerji depolama sistemlerinde kullanılan bataryalardır. Yüksek enerji yoğunluğu ve güvenlik özellikleri ile dikkat çekerler. NaS bataryalar, yüksek enerji yoğunluğu sağlar ve uzun ömürlüdür. Ancak, bu bataryaların çalışabilmesi için yüksek sıcaklık ortamlarına ihtiyaç duyulması, bazı uygulamalarda zorluklar yaratabilir. NaS bataryaların diğer dezavantajı ise karmaşık tasarımlarının olmasıdır, bu nedenle bakım ve kullanım sırasında dikkatli olunması gerekir [42-43, 46-47, 51].

Lityum-iyon bataryalar yüksek çalışma gerilimi, yüksek güç yoğunluğu, hafiflik, yüksek döngü sayısı, düşük çevresel kirlilik ve iyi güvenlik özellikleri sayesinde günümüzde en yaygın kullanılan şarj edilebilir batarya türüdür. Günümüzde lityum-iyon bataryalar cep telefonları gibi kişisel elektronik cihazlardan, elektrikli araçlar gibi yeni gelişen sınıflara kadar çeşitli uygulamalarda kullanılmaktadır. Bu lityum-iyon pillerinde kendi aralarında çeşitleri vardır [20, 52].

Çizelge 2.2' de bu çeşitlerin bazılarının karşılaştırılması gösterilmiştir. Lityum-iyon bataryalar, son yıllarda elektrikli araçlar (EV) ve taşınabilir elektronik cihazlar gibi çeşitli uygulamalarda tercih edilen başlıca enerji depolama çözümleri haline gelmiştir. Bunun temel nedeni lityum-iyon bataryaların yüksek enerji yoğunluğu, uzun ömürleri ve hızlı şarj olabilme kapasiteleri gibi önemli avantajlar sunmasıdır.

Yüksek enerji yoğunluğu sayesinde lityum-iyon bataryalar daha küçük boyutlarda yüksek miktarda enerji depolayabilir bu da elektrikli araçların daha uzun menzil sunmalarını sağlar. Ayrıca hızlı şarj özellikleri sayesinde kullanıcılar bataryalarını kısa sürede şarj edebilir bu da batarya kullanımını daha pratik hale getirir.

Lityum-iyon bataryalar diğer batarya türlerine göre daha uzun bir yaşam döngüsüne sahip olup minimum bakım gereksinimi ile daha düşük maliyetle kullanılabilirler. Bunun yanı sıra çevreye duyarlı tasarımları ve toksik olmayan özellikleri ile lityum-iyon bataryalar çevre dostu bir enerji depolama çözümü sunar.

Çizelge 2.2 : Lityum-iyon pil kimyasallarının karşılaştırılması [24, 36, 44, 53-54].

Kimyasal	Enerji Yoğunluğu (Wh/kg)	Artıları	Eksileri
(LiCoO ₂)	150-200	Yüksek enerji yoğunluğu, hızlı şarj	Düşük termal stabilite, güvenlik endişeleri
(LiFePO ₄)	100-170	Yüksek termal kararlılık, uzun çevrim ömrü	Daha düşük enerji yoğunluğu
(NMC)	200'e kadar	Dengeli performans, özelleştirilebilir özellikler	Daha yüksek maliyet
(NCA)	250-300	Çok yüksek enerji yoğunluğu	Termal yönetim zorlukları

Çizelge 2.2 de görüleceği üzere bu çizelge farklı lityum iyon batarya kimyalarının enerji yoğunluğu, avantajları ve dezavantajlarını özetlemektedir.

LiCoO₂ (lityum kobalt oksit) bataryaları yüksek enerji yoğunluğu ve hızlı şarj avantajı sunarken, düşük termal stabilite ve güvenlik endişeleri taşır.

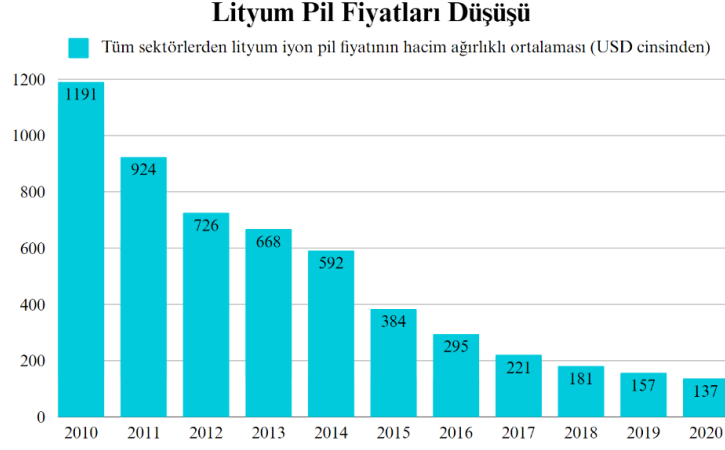
LiFePO₄ (Lityum demir fosfat) yüksek termal kararlılığı ve uzun çevrim ömrü ile güvenilirlik sağlarken, enerji yoğunluğu nispeten düşüktür.

NMC (Lityum Nikel Manganez Kobalt Oksit) bataryaları dengeli performans ve özelleştirilebilir özellikleriyle dikkat çeker, ancak daha yüksek maliyetlidir.

NCA (nikel kobalt alüminyum oksitler) kimyası ise 250-300 Wh/kg gibi çok yüksek enerji yoğunluğu sağlar, ancak termal yönetim zorluklarıyla karşılaşabilir. Bu farklılıklar kullanım amacına göre uygun batarya seçiminde önemli rol oynar.

Tüm bu avantajlar lityum-iyon bataryaların hem taşınabilir cihazlarda hem de elektrikli araçlara kadar varan geniş bir yelpazede tercih edilmesine olanak tanımaktadır.

Elektrikli araçlar için batarya maliyetlerinin oluşturduğu payın büyüklüğü göz önüne alındığında lityum-iyon bataryaların maliyetlerinin zamanla azalması ve teknolojilerinin geliştirilmesi onları daha ekonomik ve erişilebilir hale getirmiştir. Bu ekonomik erişilebilirliklerin zaman içinde değişimine Şekil 2.4'te yer verilmiştir [24-25, 55-57].

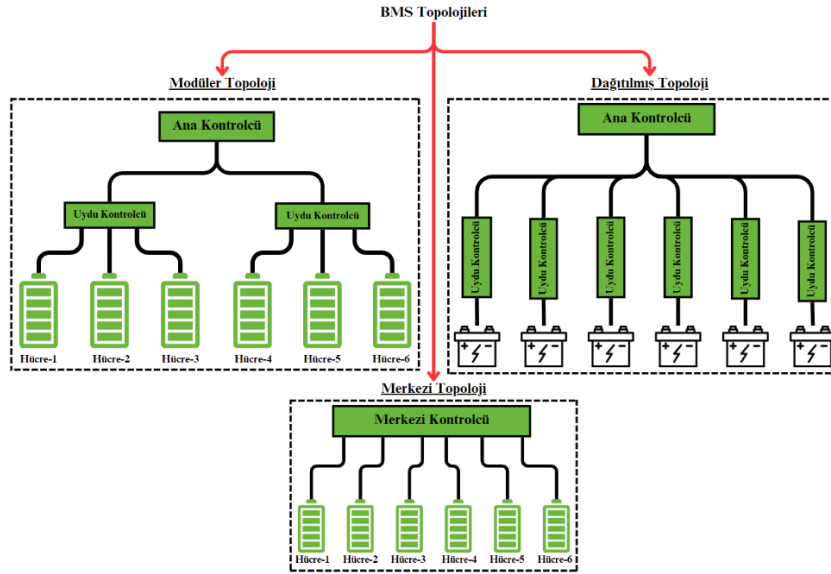


Şekil 2.4 : Lityum-iyon batarya ücretleri [55].

Sonuç olarak, yüksek güvenlik standartları, verimli enerji depolama kapasitesi ve çevre dostu özellikleri ile lityum-iyon bataryalar modern batarya çözümleri arasında en popüler tercih olmaya devam etmektedir. Elektrikli araçlar için farklı batarya modelleri tartışıldı ve lityum-iyon bataryaları daha uygun olduğu bulundu.

2.1.2 BMS topolojileri

Batarya Yönetim Sistemi topolojileri bataryaların güvenli, verimli ve sürdürülebilir bir şekilde yönetilmesinde kritik bir rol oynar. Bu topolojiler iletişim yapıları, maliyet, esneklik ve uygulama alanına göre farklı sınıflandırmalara ayrılır. Genel olarak bu sınıflandırmalar dört ana başlıkta incelenir. Bunlar Merkezi topoloji (Centralized BMS), Modüler topoloji (Modular BMS) ve Dağıtılmış topoloji (Distributed BMS) olarak adlandırılmaktadır. Topoloji çeşitlerine Şekil 2.5'te yer verilmiştir [6, 7, 17].



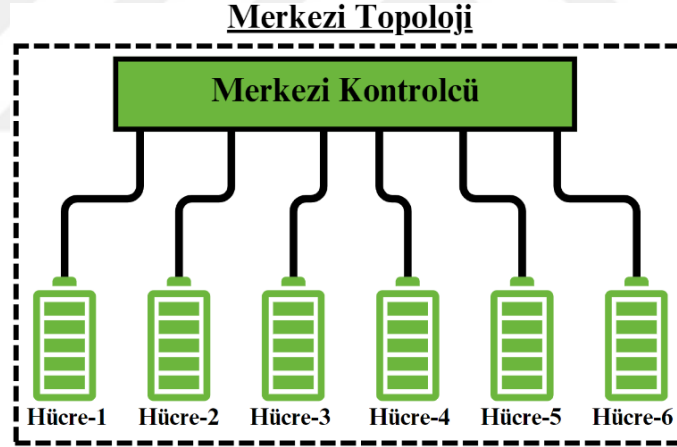
Şekil 2.5 : Üç farklı BMS topoloji örneği.

2.1.2.1 Merkezi topoloji

Bu topoloji çeşidinde batarya yönetim sisteminin tüm görevlerini tek bir kart üstlenmektedir ve tüm işlemler bu kart üzerinden yapılmaktadır. Tek bir kart olduğu için maliyeti düşüktür. Merkezi BMS yapısında batarya sisteminin tüm görevlerini gerçekleştiren tek bir merkezi kontrol birimi bulunur.

Bu yapı tüm işlemler için tek bir baskılı devre kartı (PCB) kullanarak iletişim sağlar ve bu birim batarya voltajlarını izler, sıcaklık ölçümlerini yapar ve hücre dengelemelerini gerçekleştirir. Genellikle batarya çıkışından güç alır ve iletişim bir akıllı devre kartı üzerinden hem dahili hem de harici yapılır.

Ancak merkezi BMS yapısının dezavantajları arasında sistem karmaşıklığının artması hücre sayısının artışıyla birlikte kablo yükünün yükselmesi ve sistemin esneklik eksikliği sayılabilir. Bu nedenle genellikle küçük sistemler ve uygulamalar için uygundur. Bu topoloji çeşidi daha anlaşılır olması için Şekil 2.6'da örnek diyagram hali gösterilmiştir [20, 30, 45, 57-66].



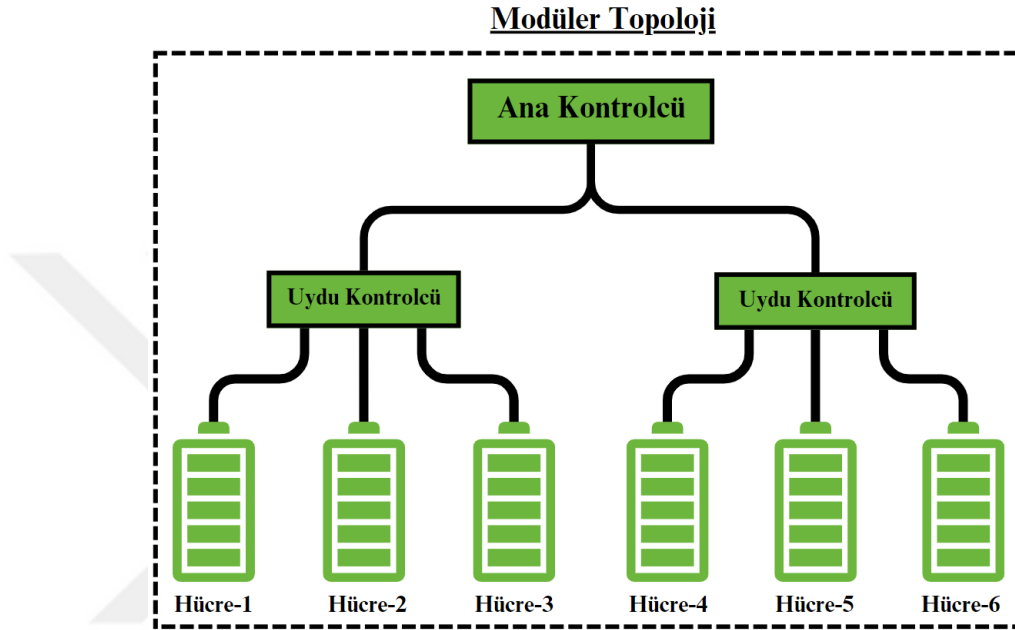
Şekil 2.6 : Merkezi topoloji örnek diyagram gösterimi.

2.1.2.2 Modüler topoloji

Modüler batarya yönetim sistemi hem esneklik hem de maliyet açısından merkezi ve dağıtılmış yapılar arasında bir denge kurar. Bu yapı ana kontrolcü ve uydu kontrolcü alt birimleriyle çalışır.

Ana kontrolcü birden fazla uydu kontrolcüyü yönetir ve hücreler arasında dengeleme işlemlerini gerçekleştirir. Modüler BMS yapıları genellikle 6 ila 18 hücre arasında birden fazla hücreyi yöneten modüller aracılığıyla çalışır.

Bu yapı her bir yapıdaki yükleri azaltmak ve daha esnek sistemler oluşturmak için tasarlanmıştır. Bir diğer avantajı ise hücrelere özel montaj ve modüler bir mimari kullanılarak sistemin kolayca genişletilebilmesidir. Ancak bu yapı donanım açısından daha fazla gereklilik ve yazılım karmaşası gerektirir. Bu gereklilikler modüler yapının maliyet açısından da yükselmesine sebep olur. Bu topoloji çeşidi daha anlaşılır olması için Şekil 2.7’de örnek diyagram hali gösterilmiştir [57-58, 67-70].



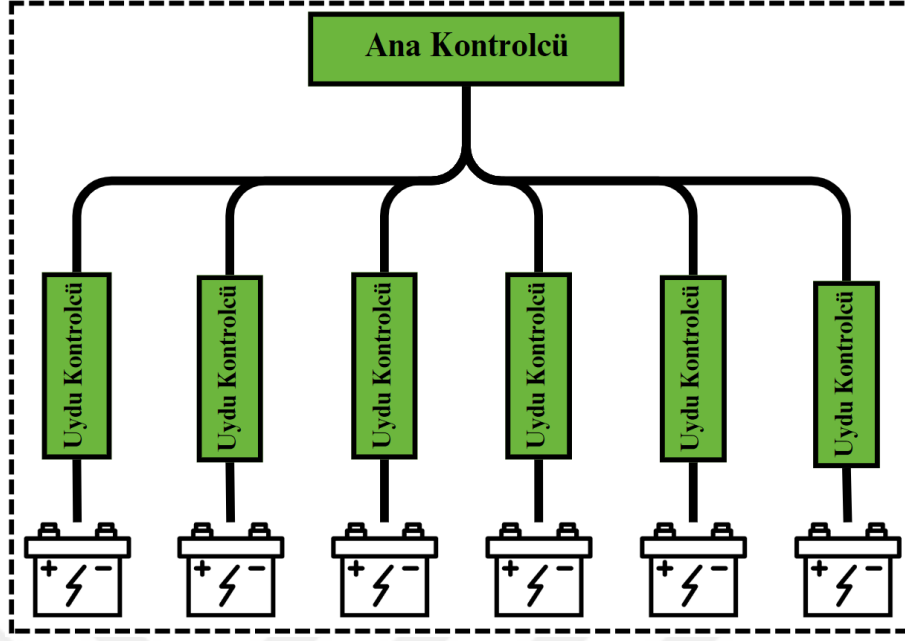
Şekil 2.7 : Modüler topoloji örnek diyagram gösterimi.

2.1.2.3 Dağıtılmış topoloji

Dağıtılmış batarya yönetim yapısında merkezi bir kontrol birimi bulunmaz. Her bir batarya hücresinin izlenmesi için birden fazla bağımsız uydu kontrolcü birimi (ölçüm birimi) kullanılır. Bu uydu kontrolcü birimleri genellikle tek bir iletişim kablosu aracılığıyla birbirine bağlanır ve eğer varsa ana kontrolcüye veri aktarımı gerçekleştirirler. Bu yapı modüler batarya yönetim sisteminden farklı olarak her bir hücre için bağımsız kontrol sağlar.

Bu yapı güvenilirliği artırsa da maliyet açısından daha yüksek bir yapıdır ve karmaşık sistem entegrasyonunu beraberinde getirir. Hücre başına hassas izleme imkânı sunar ve özellikle yüksek hassasiyetli uygulamalar için uygundur. Bu hassas ölçümler beraberinde maliyetin artmasını da beraberinde getirmektedir. Bu topoloji çeşidi daha anlaşılır olması için Şekil 2.8’de örnek diyagram hali gösterilmiştir [20, 57-58, 69, 71-72].

Dağıtılmış Topoloji



Şekil 2.8 : Dağıtılmış topoloji örnek diyagram gösterimi.

Bu incelenen topoloji yapıları uygulama alanları, maliyetler ve sistem güvenliği açısından avantajlar ve dezavantajlar taşır. Doğru BMS topolojisi seçimi genellikle hedef uygulamanın ihtiyaçlarına, maliyet sınırlamalarına ve sistemin esneklik gerekliliklerine göre belirlenir.

2.1.3 Batarya yönetim sistemi koruma özellikleri

Batarya yönetim sistemleri bataryaların güvenli ve verimli bir şekilde çalışmasını sağlamak için çeşitli koruma mekanizmaları içerir. Bu koruma mekanizmaları, bataryaların ömrünü uzatmak ve güvenliğini sağlamak için kritik öneme sahiptir. Bataryayı kısa devre, aşırı voltaj, aşırı akım, aşırı şarj, aşırı sıcaklık ve kaçak akım gibi hatalara karşı korur. Bu korumalar bataryanın güvenliğini sağlamak için kritik öneme sahiptir. Aşırı deşarj koruması (over-discharge protection), aşırı şarj koruması (over-charge protection), ani akım koruması (surge current protection), aşırı ısınma koruması (overheating protection) ve hücre dengeleme (cell balancing) bu korumalara örnek olarak gösterilebilir.

Aşırı deşarj koruması bataryanın belirli bir voltaj seviyesinin altına düşmesini önler. Bu koruma bataryanın kapasitesini ve ömrünü korumak için önemlidir. Aşırı deşarj durumu gerçekleşirse bataryanın kalıcı olarak zarar görmesine ve kapasitesinin süreç içinde azalmasına neden olabilir. Örnek olarak bir pil hücresi için kesme voltajı hücre başına 2.8 volt olduğu varsayıldı. Bu voltajın altına düşüldüğünde batarya yönetim

sistemi devreye girer ve bataryanın daha fazla deşarj olmasını engeller, bu sayede bataryanın ömrünün uzaması sağlanmış olur [38, 40, 73].

Aşırı şarj koruması bataryanın belirli bir voltaj seviyesinin üzerine çıkmasını önler. Aşırı şarj bataryanın aşırı ısınmasına ve hatta patlamasına neden olabilir. Bu koruma ile bataryanın güvenliğini sağlamak batarya sağlığı için kritik öneme sahiptir. Örnek olarak bir pil hücresi için kesme voltajı hücre başına 3.5 volt olduğu varsayıldı. Bu voltajın üzerine çıkıldığında batarya yönetim sistemi şarjı durdurur ve bataryanın aşırı şarj olmasını engeller [22-23, 25, 73-79].

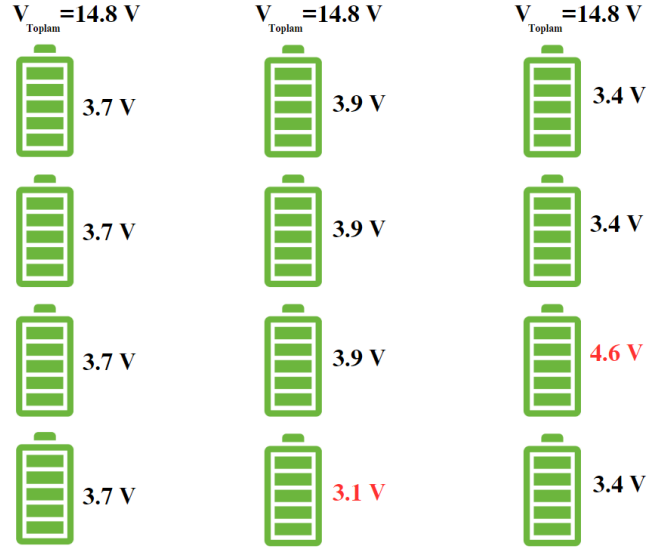
Ani akım koruması bataryanın ani yüksek akım çekişlerinden korunmasını sağlar. Bu tür akımlar bataryanın ve bataryaya bağlı olan cihazların zarar görmesine neden olabilir. Ani akım koruması özellikle otomotiv uygulamalarında önemlidir [11, 80].

Aşırı ısınma koruması bataryanın sıcaklığının belirli bir seviyenin üzerine çıkmasını önler. Aşırı ısınma bataryanın performansını düşürebilir ve güvenlik riskleri oluşturabilir. Bu koruma batarya yönetim sisteminde bulunan sıcaklık sensörleri ve kontrol devreleri kullanılarak sağlanır. Sıcaklık belirli bir seviyeyi aştığında batarya yönetim sistemi ilgili bölümü devreye girer ve bataryanın soğumasını sağlar. Bu koruma ile bataryanın güvenliğini sağlamak olası kaza durumlarında hem sistem hem de canlılar için önemli bir koruma çeşididir [15-17, 42-44].

Hücre dengeleme batarya hücreleri arasındaki voltaj farklarını dengeleyerek bataryanın ömrünü uzatır ve performansını artırır. Bu işlem hücreler arasındaki enerji dengesizliğini giderir ve her bir hücrenin eşit şekilde şarj ve deşarj olmasını sağlar ve bu sayede bataryalardan daha uzun süre faydalanmayı sağlamaktadır [15-17, 81].

2.1.4 Hücre dengeleme yöntemleri

Batarya yönetim sistemi hem şarj hem de deşarj işlemleri sırasında hücreler arasında enerji dengesini korumak için dengeleme görevini yerine getirir. Bu sayede hücreler arasındaki enerji farklılıkları en aza indirilerek bataryanın sağlığı korunmuş olur. Şekil 2.9'da pil hücre dengesizlikleri bir örnek görsel olarak gösterilmektedir. Dengeleme yöntemleri pasif dengeleme ve aktif dengeleme olarak iki kategoriye ayrılır. Pasif dengelemede bir hücredeki fazla enerji dağıtılırken, aktif dengelemede ise çeşitli yöntemlerle yüksek enerjili hücrelerden düşük enerjili hücrelere enerji transferi yapılır. Bu iki ana yöntemden biri kullanılarak hücre voltajları eşitlenir [19, 70, 82-86].



Şekil 2.9 : Örnek batarya hücre dengesizlik durumları [5].

2.1.4.1 Pasif dengeleme

Pasif hücre dengeleme yöntemi aktif hücre dengeleme yöntemine göre daha basit bir yapıya sahiptir. Bu yöntemde fazla enerjiye sahip hücreler bu enerjiyi akım geçirebilme özeliği yüksek olan bir direnç üzerinden ısıya dönüştürerek ve yayarak boşaltır. Bu işlem batarya paketindeki diğer hücrelerin şarj seviyeleriyle eşitlenene kadar devam eder. Ancak bu enerji kaybı bataryanın kullanım süresini olumsuz etkiler. Pasif dengeleme hücreler arasında enerji dağıtımını yapmaz bunun yerine fazla enerji tamamen ısı olarak kaybedilir. Bu ısı ile yayma durumu hücrelerin paralelinde bulunan dirençler aracılığıyla gerçekleştirilir. Bu yöntem düşük maliyetli sistem uygulamaları için uygun olsa da enerji verimliliği oldukça düşüktür ve fazla ısı üretimi nedeniyle soğutma gereksinimi artar [15-17, 87-89].

Pasif hücre dengeleme yöntemi temel olarak sabit bypass direnci ve anahtarlamalı bypass direnci olarak iki farklı topolojiye sahiptir.

Sabit bypass direnci yönteminde her hücrenin paralelinde sabit bir direnç bulunur ve bu direnç fazla enerjiyi sürekli olarak boşaltır. Basit bir tasarıma sahiptir ve maliyet açısından avantajlıdır. Ancak düşük verimlilik ve sürekli enerji kaybı gibi dezavantajlarını beraberinde getirmektedir [5].

Anahtarlamalı bypass direnci yönteminde dirençler hücre gerilim izleme entegreleri tarafından kontrol edilir ve yalnızca gerekli durumlarda devreye girer. Daha az enerji kaybı sağlar ancak kontrol sistemleri sabit bypass yöntemine göre karmaşıktır ve maliyeti artırır [5, 90].

2.1.4.2 Aktif dengeleme

Aktif hücre dengeleme yöntemi bir batarya paketi içindeki seri bağlı hücreler arasında şarj durumundaki dengesizlikleri yani farklılıkları en aza indirmeyi amaçlar. Bu yöntem hücrelerin şarj durumundaki tutarsızlıkları gidermek için pasif dengelemeden farklı olarak yüksek yüklü hücrelerden düşük yüklü hücrelere enerji transferi yapar. Bu transfer genellikle kapasitif veya indüktif yük taşıma yöntemleriyle gerçekleştirilir. Bu sayede fazla olan enerji pasif dengelemede olduğu gibi ısı olarak kaybedilmeden verimli bir şekilde aktarılır ancak bu yöntem devre karmaşıklığını artırır ve maliyeti yükseltir [79, 85, 87, 89].

Aktif dengeleme yöntemleri hücre bypass, hücreden hücreye, hücreden pakete, paketten hücreye ve hücreden pakete-hücreye olmak üzere beş ana alt kategoriye ayrılır [5, 88-89].

Hücre bypass yöntemi ile hücre gerilimleri arasında farklılık olduğunda fazla enerjiye sahip hücreler bypass edilerek diğer hücrelerin dengelemesi beklenir. Bu yöntem uygulaması kolay modülerleştirilebilir ve düşük maliyetlidir. Ancak genellikle şarj veya deşarj sürecinin sonunda kullanılır bu da verimliliğini sınırlamaktadır [5, 89, 91].

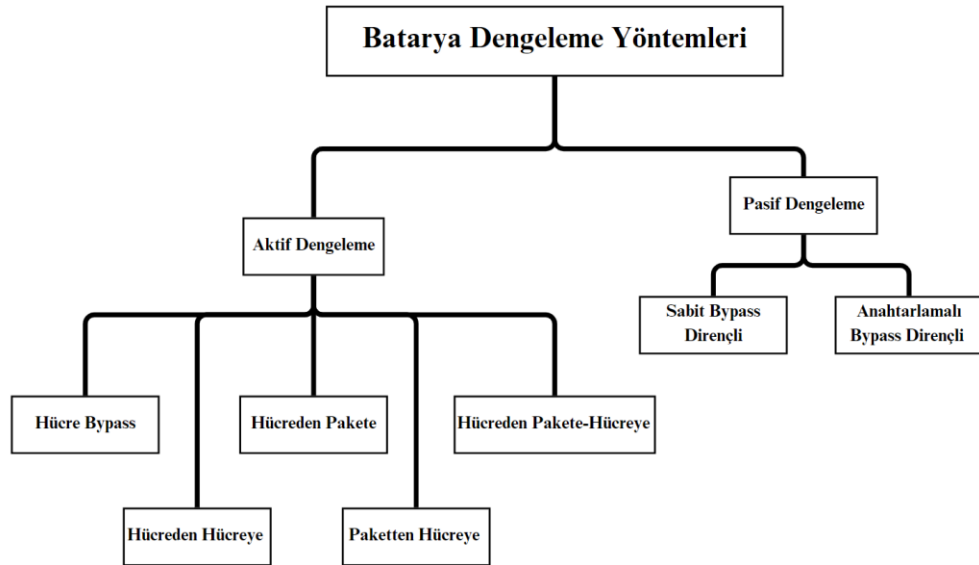
Hücreden hücreye yöntemi ile fazla enerjiye sahip bir hücreden daha az enerjiye sahip komşu hücrelere enerji aktarımı yapılır. Bu yöntemde verimlilik yüksektir ancak kontrol sistemleri karmaşıktır ve dengeleme süresi uzundur [5, 88-89, 92].

Hücreden pakete yönteminde en çok şarj edilmiş hücreden enerji alınarak tüm hücrelere eşit şekilde dağıtılır. Paketten hücreye yönteminde batarya paketi genelinden en düşük şarjlı hücreye enerji transferi yapılır. Bu yöntemler şarj sırasında enerji kaybını minimize eder ve hücrelerin güvenliğini artırır.

Hücreden pakete ve hücreye yöntemi hem hücreden pakete hem de paketten hücreye yöntemlerini birleştirir. Bu yöntemler DC-DC dönüştürücüler gibi karmaşık ve pahalı topolojiler gerektirir. Ancak batarya ömrünü ve güvenliğini artırdığı için avantajlıdır [85, 87, 89-90].

Aktif dengeleme batarya hücrelerinin kullanılabilir kapasitesini maksimize ederken enerji kaybını en aza indirir. Ayrıca batarya hücre ömrünü uzatır ve yüksek döngü sayısına sahip uygulamalarda oldukça idealdir. Ancak bu yöntem ek bileşenlere olan ihtiyaç nedeniyle maliyeti artırır ve dengeleme süresinin uzunluğu gibi dezavantajlar barındırır [5, 86, 89].

Sonuç olarak pasif ve aktif hücre dengeleme yöntemleri enerji yönetimi ve uygulama ihtiyaçlarına göre farklı avantajlar sunar. Pasif hücre dengeleme düşük maliyetli yapısıyla özellikle basit sistemlerde tercih edilirken enerji kaybı ve ısınma gibi dezavantajlara sahiptir. Aktif hücre dengeleme enerji tasarrufu açısından daha üstün bir performans sunar ve daha yüksek güç yüklerini yönetebilir. Bu yöntem daha karmaşık bir tasarım gerektirdiği için maliyetleri de yükseltir. Yani pasif dengeleme düşük maliyet ve basitlik arayan uygulamalar için ideal bir çözüm sunarken aktif dengeleme yüksek enerji verimliliği ve hızlı performans avantajlarıyla öne çıkar. Genellikle daha karmaşık sistemlerde tercih edilir. Hem aktif hem de pasif dengeleme çeşitlerini içeren yapıya Şekil 2.10'da yer verilmiştir [5, 15].



Şekil 2.10 : Dengeleme yöntemi çeşitleri [5].

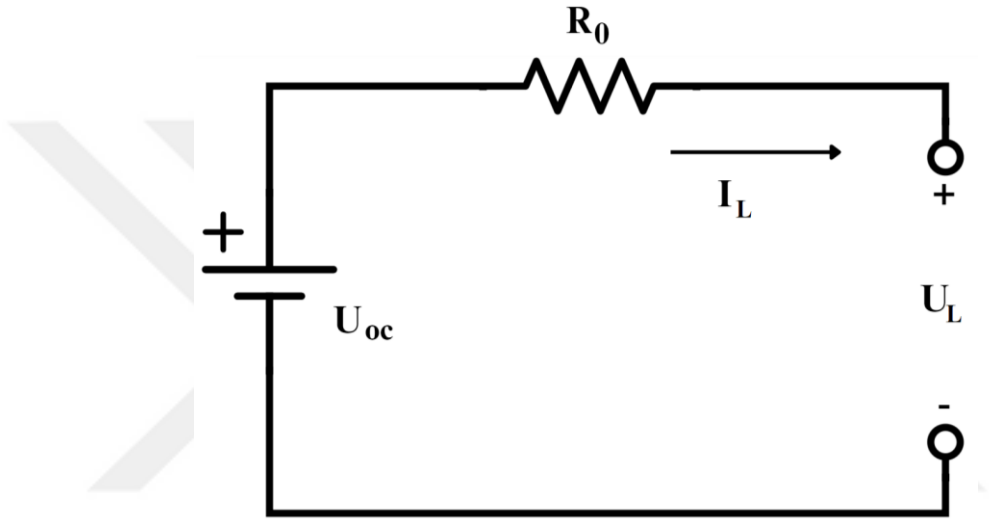
2.1.5 Durum izleme ve tahmin algoritmaları

Batarya yönetim sistemlerinin temel işlevlerinden biri bataryanın şarj durumunu (SoC) ve sağlık durumunu (SoH) tahmin etmektir. SoC bataryanın mevcut şarj seviyesinin tam kapasitesine oranını yani bataryanın örneğin yüzdelik olarak kalan miktarını ifade eder. SoH ise bataryanın zamanla performans düşüşünü yani bataryanın sağlık durumunu gösterir. Başka bir deyişle SoC bataryanın şu anki enerji seviyesini SoH ise bataryanın genel sağlığını ve kalan ömrünü yansıtır [14, 27, 93].

Bataryaların matematiksel analizlerini gerçekleştirebilmek için, dinamik özellikleri ve çalışma prensipleri göz önünde bulundurularak, direnç, kapasitör ve gerilim kaynaklarından oluşan elektriksel eşdeğer devre modelleri geliştirilmiştir. Bu modeller bataryanın şarj durumu (SoC) ve sağlık durumu (SoH) gibi parametrelerin tahmin

edilmesinde yaygın olarak kullanılmaktadır. Şarjdeşarj testleri sonucunda bataryaya ait elektriksel devre parametreleri elde edilir. Bu parametreler sıcaklık, şarj durumu, akım, kapasite ve batarya ömrü gibi değişkenlere bağlı olarak farklılık gösterebilir. Ancak hesaplama sürecini kolaylaştırmak amacıyla bazı parametrelerin sabit olduğu varsayılabilir [27, 94-95].

Bataryaların en temel eşdeğer devresi Şekil 2.11’de gösterilen ve iç direnç modeli olarak adlandırılan devredir. Bu modelde, bataryanın açık devre gerilimi U_{oc} iç direnci R_0 , batarya akımı I_L ve terminal gerilimi U_L olarak tanımlanır.



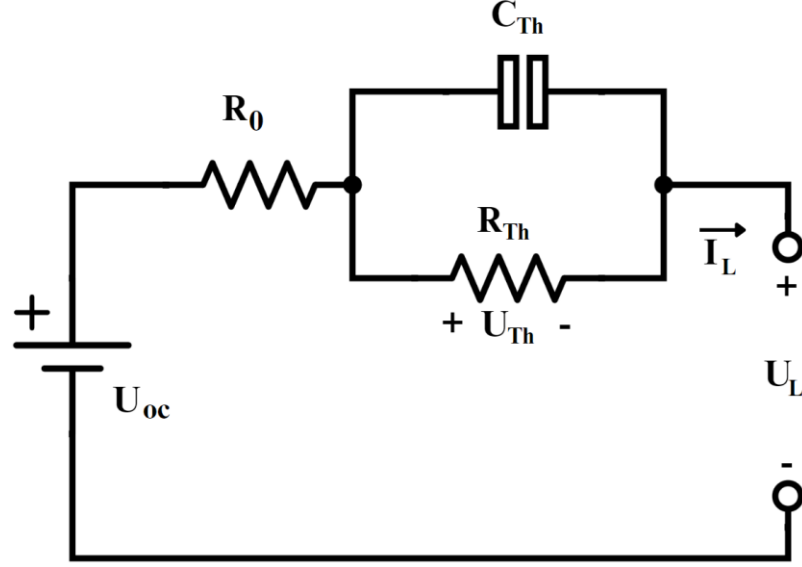
Şekil 2.11 : İç direnç eşdeğer devre modeli [94-95].

Bu modelin matematiksel ifadesi denklem 2.1’deki gibidir.

$$U_L = U_{oc} - I_L R_0 \quad (2.1)$$

Bu basit model bataryanın temel karakteristiklerini açıklamak için yeterlidir. Ancak bataryanın kimyasal yapısı ve dinamik özelliklerinden dolayı bu modelin sonuçları gerçek değerlere kıyasla hatalı olabilir. Bu nedenle daha gerçekçi bir yaklaşım olabilmesi için Thevenin eşdeğer devre modeli geliştirilmiştir [94-95].

Thevenin modeli hazırlanmış devre olan Şekil 2.12’de R_{Th} polarizasyon direnci, I_L batarya akımını, U_{oc} bataryanın açık devre gerilim değeri, C_{Th} şarj vedeşarj sırasındaki geçici tepki, R_0 bataryanın iç direnci, U_{Th} kapasitör direnç çifti üzerine düşen gerilimi ve U_L ise batarya terminal gerilimini ifade eder. Bu eşdeğer devrenin matematiksel modeli denklem 2.2 ve 2.3’te gösterildiği gibi oluşturulabilir.



Şekil 2.12 : Thevenin eşdeğer devre modeli [94-95].

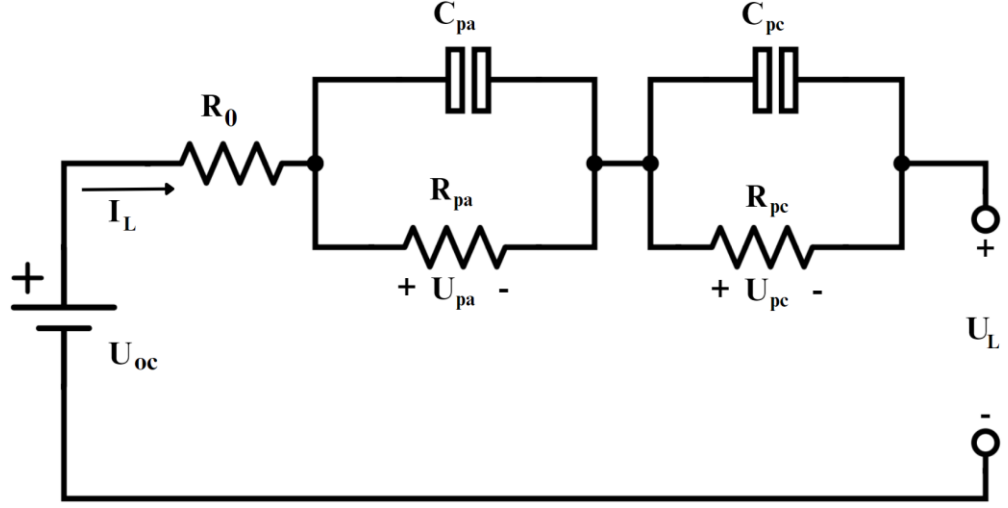
$$\dot{U}_{Th} = -\frac{U_{Th}}{R_{Th}C_{Th}} + \frac{I_L}{C_{Th}} \quad (2.2)$$

$$U_L = U_{oc} - U_{Th} - I_L R_0 \quad (2.3)$$

Lityum iyon bataryalar üzerinde yapılan testler bu tür bataryalarda polarizasyon etkilerinin varlığını ortaya koymuştur. Thevenin modeli bu polarizasyon etkisini kısmen azaltabilir ancak şarj ve deşarj süreçlerinin sonunda meydana gelen elektrokimyasal polarizasyon ve konsantrasyon polarizasyonu arasındaki farkı yeterince iyi açıklayamaz.

Bu açıklayamama durumu Çift Polarizasyon modeli adı verilen ve daha gelişmiş bir eşdeğer devre tasarlanması sonucunu doğurmuştur. Çift Polarizasyon modeli hem elektrokimyasal hem de konsantrasyon polarizasyonlarını dikkate alarak bataryayı gerçeğe en yakın şekilde modellemeyi hedefler. Bu modelin eşdeğer devresine Şekil 2.13'te yer verilmiştir.

Bu modelden oluşturulan denklemlerde yani denklem 2.4, denklem 2.5 ve denklem 2.6'da U_{oc} bataryanın açık devre gerilimini, I_L akımını, R_0 iç direncini ifade eder. Elektrokimyasal polarizasyon direnci R_{pa} , bu karakteristiğin kapasitesi C_{pa} olarak ifade edilir. Ayrıca konsantrasyon polarizasyon direnci R_{pc} ve bu karakteristiğin kapasitesi ise C_{pc} ile gösterilir. Elektrokimyasal polarizasyon gerilimi U_{pa} ve konsantrasyon polarizasyon gerilimi U_{pc} olarak tanımlanır [94-95].



Şekil 2.13 : Çift polarizasyon eşdeğer devre modeli [94-95].

$$\dot{U}_{pa} = -\frac{U_{pa}}{R_{pa}C_{pa}} + \frac{I_L}{C_{pa}} \quad (2.4)$$

$$\dot{U}_{pc} = -\frac{U_{pc}}{R_{pc}C_{pc}} + \frac{I_L}{C_{pc}} \quad (2.5)$$

$$U_L = U_{oc} - U_{pa} - U_{pc} - I_L R_0 \quad (2.6)$$

Çift Polarizasyon eşdeğer devrenin matematiksel modelleri denklem 2.4, denklem 2.5 ve denklem 2.6'da gösterildiği gibi oluşturulabilir. Her ne kadar daha karmaşık modeller geliştirilmişse de Thevenin ve Çift Polarizasyon modelleri SoC kestiriminde en sık kullanılan ve doğruluğu kanıtlanmış yöntemlerdir [94-95].

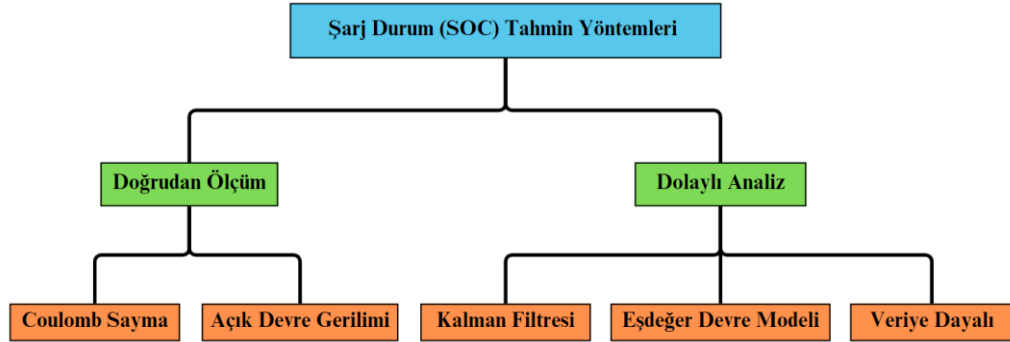
2.1.5.1 Şarj durumu (SoC) tahmini

Bataryanın şarj durumunu tahmin etmek için farklı yöntemler kullanılmaktadır. Bunlardan biri doğrudan ölçüm yöntemi diğeri ise dolaylı analiz yöntemleridir. Bu tahmin yöntemlerinin de çeşitleri vardır. Bu temel yöntemlerin çeşitlerini gösteren örnek yapı Şekil 2.14'te gösterilmiştir. Daha farklı yöntemler de kullanılmaktadır. Bu kısımda birkaç çeşidine yer verildi.

Bataryalar için şarj durumu (SoC) en önemli parametrelerden biridir ve bataryanın performansını değerlendirmede temel bir role sahiptir. SoC farklı yöntemlerle açıklanabilse de genellikle bataryanın o anki kapasite değeri $Q_{(t)}$ ile nominal kapasite değeri $Q_{(n)}$ arasındaki oran bataryanın belirli bir andaki şarj durumunu $SoC_{(t)}$ ifade

eder. Nominal kapasite, bataryanın üretici tarafından belirtilen ve depolayabileceği maksimum şarj miktarını temsil eder. Şarj durumu denklem 2.7'deki gibi ifade edilir [94-95].

$$SoC_{(t)} = \frac{Q_{(t)}}{Q_{(n)}} \quad (2.7)$$



Şekil 2.14 : Şarj durum tahmin yöntemlerinden bazıları [15, 21, 37].

Bu matematiksel ifade bataryanın mevcut şarj seviyesini tam kapasitesine oranlayarak belirler ve enerji yönetiminde önemli bir referans sağlamış olur.

Doğrudan ölçüm yöntemlerinin çeşitleri vardır. En yaygın kullanılan doğrudan ölçüm yöntemleri açık devre gerilimi ve coulomb sayma yöntemidir. Açık devre gerilimi yöntemi bataryanın açık devre gerilimini SoC ile ilişkilendirerek şarj durumunu tahmin eder. Bu yöntem basit ve düşük maliyetli olmasına rağmen bataryanın dinlenme durumunda olması gerektiğinden gerçek zamanlı uygulamalar için çok uygun değildir [14, 27, 93].

Açık devre gerilimi yönteminde batarya belirli bir süre boyunca nominal akım değerinde şarj edilir veya deşarj edilir. Daha sonra batarya şarj devresinden ve yükten ayrılarak bir süre dinlenmeye bırakılır. Bu sürenin ardından bataryanın uçlarındaki gerilim ölçülerek açık devre gerilimi V_{oc} elde edilir. Şarj durumu SoC tahmini için kullanılan temel ifade denklem 2.8'de verilmiştir. Burada örnekleme zamanı t olarak gösterilmiştir [94-95].

$$SoC = \alpha_4 V_{oc}^2 + \alpha_3 t^2 + \alpha_2 V_{oc} + \alpha_1 V_{oc} \quad (2.8)$$

Coulomb sayma yöntemi bataryaya giren ve çıkan akımı entegre ederek şarj durumunu tahmin eder. Bu yöntem basit ve gerçek zamanlı uygulamalar için uygundur.

Fakat başlangıç şarj durumu değerinin bilinmesi gerektiği ve akım sensörlerindeki hataların hesaplamaları SoC tahminini etkileyebileceği bir dezavantaja sahiptir [14, 93, 96].

Ölçülen akımların entegrasyonu sonucunda şarj durumu hesaplanabilir. Bu yöntemin matematiksel ifadesine denklem 2.9'da yer verilmiştir [94-95].

$$SoC_{(t)} = SoC_{(t-1)} - \frac{1}{C_n} \int_0^t \eta I_b dt \quad (2.9)$$

Denklem 2.9'da bataryanın belirli bir andaki şarj durumunu $SoC_{(t)}$, önceki şarj durumu $SoC_{(t-1)}$ üzerinden değerlendirir ayrıca bu denklemde C_n bataryanın maksimum enerji depolama kapasitesini, I_b bataryaya giren veya çıkan akımı ve η ise Coulomb verim katsayısını temsil eder. Verim katsayısı bataryanın enerji dönüşümündeki etkinliği ifade eder ve her batarya türü için farklılık gösterebilir. Bu yöntem doğru akım ölçümü ve hassas hesaplamalar gerektirdiği için güvenilir bir şarj durumu tahmini sunar [94-95].

Bir diğer yöntem dolaylı analiz yöntemidir. Dolaylı analiz yöntemlerinin de çeşitleri vardır. En yaygın kullanılan dolaylı analiz yöntemleri eşdeğer devre modeli, kalman filtresi ve veriye dayalı yöntemleridir. Eşdeğer devre modeli yöntemi bataryayı bir elektrik devresi olarak modelleyerek şarj durumunu tahmin eder. Bu yöntem durum izleme ve tahmin algoritmaları başlığının giriş kısmında detaylı olarak bahsedilmiştir. Bu yöntem bataryanın davranışını yakalayabilme yeteneğine sahip olsa da hesaplama karmaşıklığının yüksek olması dezavantajlarına sahiptir [14-15].

Kalman fitresi yöntemi batarya modelindeki belirsizlikleri ve gürültüyü hesaba katarak şarj durumunu tahmin eder. Bu yöntem bataryanın dinamik davranışını doğru bir şekilde tahmin edebilme yeteneğine sahip olsa da hesaplama karmaşıklığı yüksektir [14, 38, 93].

Bu filtre sistemin matematiksel modeline dayanarak bir önceki durum bilgisi, giriş ve çıkış verileri gibi parametrelerden faydalanır. Kalman filtresi tahmin ve güncelleme olmak üzere iki ana aşamadan oluşur. Tahmin aşamasında sistemin beklenen çıktıları ile hata kovaryansı hesaplanır. Güncelleme aşamasında ise Kalman kazancı olarak adlandırılan bir katsayı optimize edilerek tahmin doğruluğu artırılır. Ayrıca ölçülen değerler ve tahmini sonuçlar karşılaştırılarak hem yeni çıktı hem de hata kovaryansı

Kalman kazancına göre tekrar güncellenir. Bu kalman filtresi yöntemin temel yapısı denklem 2.10'da verilmiştir [94-95].

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (2.10)$$

Denklem 2.11'de bir önceki durum değeri x_{k-1} , kontrol sinyali u_k ve işlem gürültüsü w_{k-1} kullanılarak güncel durum değerini x_k hesaplar.

$$y_k = Cx_k + Du_k + v_k \quad (2.11)$$

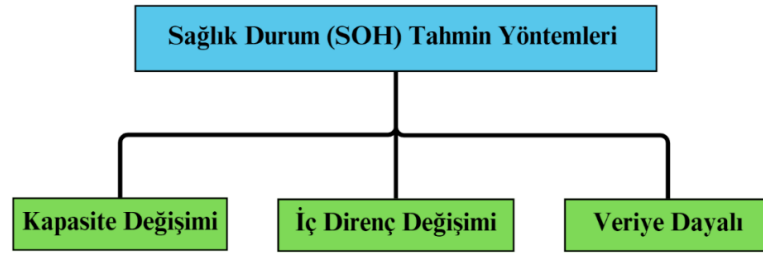
Denklem 2.11'de ölçülen ancak kesin doğruluğu bilinmeyen bir değer y_k , durum değeri x_k ve ölçüm gürültüsü v_k üzerinden güncel durumu ifade eder. Bu denklemlerde w_k ve v_k gürültü terimlerini ifade etmektedir ve genellikle sabit kabul edilir veya etkisiz olarak değerlendirilir. Kalman filtresi hem tahmin hem de güncelleme aşamalarında bu matematiksel yapı ile sistemin durumunu etkili bir şekilde kestirir [94-95].

Veriye dayalı yöntemler batarya verileri kullanarak şarj durumunu tahmin etmek için makine öğrenmesi algoritmalarını kullanır. Yapay sinir ağları bataryanın karmaşık doğrusal olmayan davranışını modellemek için kullanılabilir. Bu yapay sinir ağı yöntemi doğru SoC tahminleri sağlayabilme yeteneğine sahip olsa da büyük miktarda veriye ihtiyaç duyar ve eğitim süreci zaman alıcı olabilir.

Derin Öğrenme yapay sinir ağlarının daha gelişmiş bir formudur ve daha karmaşık batarya davranışlarını modellemek için kullanılabilir. Bu yöntem SoC tahmininde yüksek doğruluk sağlayabilme yeteneğine sahiptir. Fakat çok büyük miktarda veriye ihtiyaç duyar ve eğitim süreci oldukça karmaşık ve hesaplama açısından yoğun olabilir [13-15, 18, 81].

2.1.5.2 Sağlık durum (SoH) tahmini

Batarya sağlık durumunu (SoH) tahmin etmek için en sık kullanılan yöntemler kapasite değişimi, iç direnç değişimi ve veriye dayalı yöntemlerdir. Bu tahmin yöntemlerinin de çeşitleri vardır. Bu temel yöntemlerin çeşitlerini gösteren örnek yapı Şekil 2.15'te gösterilmiştir. Daha farklı yöntemler de kullanılmaktadır burada birkaç çeşidine yer verilmiştir [16, 18].



Şekil 2.15 : Sağlık durum tahmin yöntemlerinden bazıları [15].

Kapasite Değişimi yöntemi batarya sağlık durumunu belirlemek için en yaygın kullanılan göstergedir. Bu yöntem bataryanın tam şarj edilmesi ve ardından belirli bir deşarj akımıyla tamamen boşaltılmasıyla gerçekleştirilir. Boşaltılan kapasite bataryanın başlangıç kapasitesiyle karşılaştırılarak SoH belirlenir [18, 75, 97].

İç direnç değişimi yönteminde bataryanın iç direncindeki artış batarya sağlık durumunun bir göstergesidir. İç direnç bataryanın içindeki iyonların hareketine karşı gösterdiği direnci ifade eder ve bataryalar yaşlandıkça iç dirençleri artar ve bu da performans düşüşüne yol açar. Bu iç direnç değişimiyle bataryaların sağlık durumları incelenebilmektedir [14, 18, 21].

Genellikle SoH tahmini bataryanın iç direncinin ölçülmesiyle gerçekleştirilir. Bataryanın iç direnci kullanım süreciyle birlikte artar. Bu nedenle mevcut bir bataryanın iç direnci yeni bir bataryanın fabrika çıkışı direnci ile karşılaştırılarak SoH değeri tahmin edilebilir. SoH ile batarya iç direnci arasındaki ilişki denklem 2.12’de ifade edilmiştir [94-95].

$$SoH = \frac{R_{EOL} - R_{now}}{R_{EOL} - R_{new}} \times 100 \quad (2.12)$$

Denklem 2.12’de R_{now} bataryanın mevcut iç direncini, R_{new} bataryanın yeni durumdaki iç direncini ve R_{EOL} ise bataryanın ömrünün sonunda ölçülen iç direncini temsil eder. Araştırmalar sonucunda R_{EOL} ile R_{new} arasındaki ilişki denklem 2.13’te olduğu gibi bulunmuştur [94-95].

$$R_{EOL} = 1.6 \times R_{new} \quad (2.13)$$

Veriye dayalı yöntemlerde makine öğrenmesi algoritmaları batarya sağlık durumunu tahmin etmek için batarya verilerini analiz etmek için kullanılabilir. Bu algoritmalar bataryanın voltajı, akımı ve sıcaklığı gibi çeşitli parametrelerdeki değişiklikleri

izleyerek SoH'yi tahmin edebilir. Bu tahminler ile batarya sağlık durumu incelenebilmektedir [34, 75, 97].

2.1.6 BMS için kullanılan entegre devreler (ICs)

Batarya Yönetim Sistemi entegreleri batarya hücrelerinin gerilim, sıcaklık ve akım ölçümlerini gerçekleştiren ve güvenlik fonksiyonları sağlayan entegre devrelerdir. Bu entegre devreler batarya modüllerinin sağlıklı ve verimli bir şekilde çalışmasını destekler ve farklı gereksinimlere göre çeşitli özellikler sunar. Genelde bu yapılan temel okuma ve güvenlik işlemleri sayesinde ana kontrol eden kontrolcüyle sadece haberleşerek ana kontrolcünün iş yükünü azaltabilmektedirler. Bazı popüler batarya yönetim sistemi entegre modelleri ve özellikleri farklı birkaç teknoloji firma markası ile birlikte karşılaştırmalı olarak gösterimine Çizelge 2.3'te yer verilmiştir [9, 22, 35, 55].

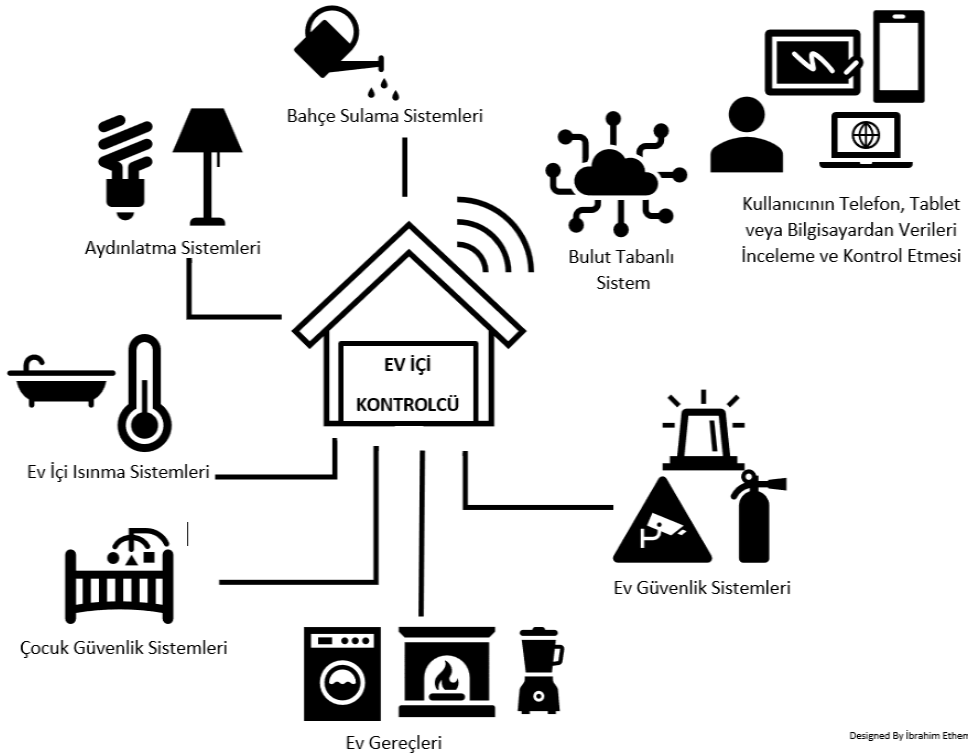
Çizelge 2.3 : Batarya yönetim sistemi entegresi karşılaştırılması [9, 98-112].

Firma	Entegre ismi	Ölçülebilen Hücre	Sıcaklık kanalı
Analog Devices	LTC6801	12	2
Analog Devices	LTC6802-1	12	2
Analog Devices	LTC6803-1	12	2
Analog Devices	LTC6803-2	12	2
Analog Devices	LTC6804-1	12	5
Analog Devices	LTC6811-1	12	5
Maxim IC	MAX11068	12	2
Maxim IC	MAX17843	12	2
Maxim IC	MAX17048	12	0
Renesas	ISL78600	12	4
Renesas	ISL78610	12	4
Texas Instruments	BQ76PL455	16	8
Texas Instruments	BQ76PL536A	6	2
Texas Instruments	BQ76952	16	9
Texas Instruments	BQ76972	16	9

Farklı üreticilere ait entegreler için kolayca karşılaştırma yapılamayabilir çünkü tam anlamıyla karşılaştırma yapmak için detaylı incelemeye ihtiyaç vardır. Daha fazla ve doğru bilgi için entegrelerin kaynaklarının ayrıca incelenmesi gereklidir [98-112].

2.2 Nesnelerin İnterneti (IoT)

Nesnelerin İnterneti (IoT), elektronik cihazların internet aracılığıyla birbirleriyle etkileşim kurmasını ve çevrelerindeki insan veya doğal süreçlerle ilişkilendirilmesini sağlayan bir teknolojidir. İnsanlar hayatları boyunca gerekli ihtiyaçlarını karşılamak için mücadele etmişlerdir. Bu mücadelelerde zaman zaman başarısız olsalar da çalışmaların geneline bakıldığında başarı bir şekilde gelmiştir. Bu mücadeleler insanlara birçok şey öğretmesinin yanı sıra bu bilgiler ışığında hayatlarını kolaylaştıracak ürünler de ortaya çıkarmalarına sebep olmuştur. Yaşamının çok uzun bir bölümü evde geçiren insanlar hayatlarını daha yaşanabilir hale getirebilmek iş hayatlarını daha rahat kontrol edebilir hale getirmek için çalışmalarını her zaman sürdürmeye devam etmişlerdir. Bu gelişim ilk insanlardan günümüze kadar devam etmesine örnek olarak normalde su ihtiyaçları için nehirlerle, ırmaklara giden insanlar evlerinde bu ihtiyacı gidermek için suyu borular aracılığı ile taşımayı ve musluk kullanmak vasıtasıyla evin içine kadar getirmeyi başarmışlardır. IoT, akıllı evlerden sanayi otomasyonuna kadar birçok farklı sektörü kapsayarak günlük yaşamı ve endüstriyel sistemleri dönüştürme potansiyeline sahiptir. Bu teknolojinin kullanıldığı ortamlardan biri olan ev ortamları için temel bir yapı örneğine Şekil 2.16'da yer verilmiştir [97, 113-114].



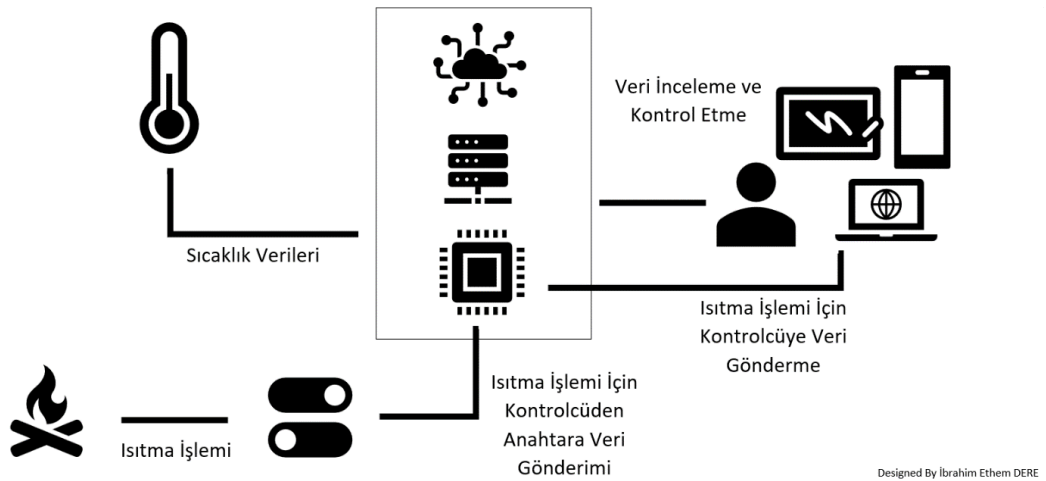
Designed By İbrahim Ethem DERE

Şekil 2.16 : IoT ev uygulamalarının örnek bir diyagramı.

Günümüzde hali hazırda bulunan iş geliştirme uygulamaları her ne kadar yeterli gibi görünse de insanlar çalışmalarına halen son noktayı koymamışlardır. İnsanlar internetin icadıyla birlikte çok hızlı bir gelişim evresine girmiştir. Şu an hemen hemen her insanın duyduğu bir terim olan Nesnelerin İnterneti (IoT-Internet of Things) gelişimi durmaksızın devam etmektedir. Teknolojinin gelişmesi nesnelerin interneti aracılığıyla farklı bir boyut kazanmıştır. Nesnelerin interneti teknolojisi insanların yaşam ve çalışma standartlarını yükseltmek amacıyla akıllı sistemlerin ortaya çıkmasını sağlamıştır [113].

IoT'nin uygulama alanları oldukça çeşitlidir. Akıllı evler ve şehirlerden tarıma, sağlık hizmetlerinden trafik yönetimine kadar geniş bir yelpazede çözümler sunar. IoT'nin gelişimi akıllı cihazların ve bağlı kullanıcıların sayısındaki artış sayesinde hızla devam etmektedir. Akıllı ortamlara örnek olarak akıllı evler, ofisler ve eğitim alanları verilebilir. Bu ortamlar IoT cihazlarının kolay erişilebilirliği sayesinde bireysel kullanıcılar ve şirketler için büyük bir ilgi odağı haline gelmiştir [114-115].

Bu sistemlerin çalışması için kontrolcü, çevre birimler (Sensör vb.) haricinde bulut sistemi üzerinden telefona bildirim gönderme veri kayıt veya cihaz kontrollerini internet üzerinden yapılabilmesi için gereken uygulamalara da ihtiyaç vardır. Bu ihtiyaçlar doğrultusunda birçok platform ortaya çıkmıştır. Bu kontrolcü çevresel birim ve bulut ortamının temel diyagram örneğine Şekil 2.17'de yer verilmiştir [116].



Şekil 2.17 : Kontrolcü, çevre birim ve bulut sisteminin temel çalışma diyagramı.

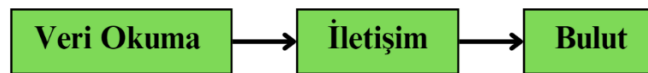
Bu tarz bulut sistemlerini asıl amacı aradaki bağlantıyı kurmak da olsa tek amaçları bu değildir. Bu amaç haricindeki amaçlara örnek olarak müşteriler için ev, iş, endüstriyel, tarım vb. alanlardaki verilerini kullanarak müşterilerin akıllı stratejiler üretmesine

yardımcı olmaktadır. Bu stratejilere örnek olarak bir dağıtım kanalıyla ilgili fazla bilgiyi azaltabilirler ve bu sayede veri depolamadan tasarruf sağlayabilirler. Bir sistemdeki ürün yenileneceğinde yeni ürün seçiminde dayanıklılık, verim, performans vb. konular hakkında üretilen veriler sayesinde karar vermeyi kolaylaştırır [117].

Nesnelerin internetinin kullanım alanı internet bağlantısına dayalı olmasından kaynaklı beraberinde güvenlik sorunlarını da getirmektedir. Bu güvenlik sıkıntılarını gidermek amaçlı çalışmalar yapılmıştır. Örneğin akıllı ev sisteminin güvenlik ve emniyet durumu, ilgili komut dosyasının herhangi bir dış müdahale nedeniyle bozulmasıyla felakete sebep olabilir. Bu yüksek derecede risk faktörü içeren cihazların bulunduğu bir ev sistemi ve iş sistemi durumunda gerçekten kritik olabilir. Bu güvenlik tehditlerinin önüne geçebilmek için farklı çeşitler yöntemler vardır. Kodlayıcı mimarilerini açıklamaya yönelik matematiksel kavramlarda bazılarında sonlu(galois) alan aritmetiği, jeneratör polinomu ve kodlayıcı mimarisi örnek verilebilir [118].

Geleneksel batarya yönetim sistemleri batarya sağlık durumu ve batarya şarj durumu yanlış tahmini ve ölçeklenebilirlik sorunları dahil olmak üzere çeşitli teknik zorluklarla karşılaşmıştır. Standart BMS'lerde pil sağlığı ve sıcaklık yönetimiyle ilgili sensör verilerini edinmek için IoT gibi gelişmiş teknolojilerden yoksundur. Bu da gerçek zamanlı izleme yeteneklerinden yoksun oldukları anlamına gelir. Bu eksiklikleri gidermek için IoT tabanlı BMS (IoT-BMS) çözümleri geliştirilmiştir [27, 77, 97, 119].

BMS ve IoT mimarisiyle oluşturulmuş bir batarya yönetim sistemi üç farklı temel işlem yapısı barındırır. Bunlar veri okuma işlemleri, iletişim işlemleri ve bulut katmanı işlemleridir. Bu işlemleri özet bir diyagramına Şekil 2.18'de yer verilmiştir.



Şekil 2.18 : BMS ve IoT özet mimarisi.

Veri okuma işlemleri pil voltajı, akımı, sıcaklığı ve diğer ilgili parametreleri izlemek için çeşitli sensörlerden oluşur [46, 77].

İletişim işlemleri sensörlerden toplanan verileri daha sonra işleme veya kullanma için bir ağ geçidi görevi üstelenerek buluta iletilir. Bu iletişim için kablosuz iletişim protokolleri yaygın olarak kullanılmaktadır [49].

Bulut katmanı işlemleri istenildiğinde büyük veri depolama ve gelişmiş analiz yetenekleri sağlar. Pil sağlığı ve kalan kullanım ömrü hakkında fikir edinmek için makine öğrenmesi algoritmaları ve tahmine dayalı modeller bulutta uygulanabilir [97, 119].

IoT tabanlı batarya yönetim sistemleri gerilim, akım ve sıcaklık gibi hayati pil parametreleri hakkında gerçek zamanlı veri toplar. Bu veriler pil davranışını anlamak anormallikleri tespit etmek ve zamanında müdahale sağlamak için kullanılabilir bir olanak tanıyabilir [49, 119]. Örneğin güneş ve rüzgâr enerjisi sistemlerinde pil performansını optimize etmek için kullanılabilir. Uzaktan izleme bu sistemlerin verimliliğini ve güvenilirliğini artırmaya yardımcı olur [46, 97, 120].

IoT ve BMS entegrasyonu pil performansını, güvenliğini ve ömrünü optimize ederek elektrikli ürünlerin daha geniş ölçekte kullanılması için güven ortamı sağlanmasında çok iyi bir potansiyele sahiptir. Gelişmiş izleme, veri analizi ve kontrol yetenekleri daha verimli ve güvenilir pil sistemleri için güzel bir yapı oluşturabilmektedir. Bu yapı standartlaşma, siber güvenlik ve yapay zekâ entegrasyonu gibi zorlukların da ele alınmasını gerektirmektedir. Sürekli araştırma ve geliştirme çabaları gelecekte daha da akıllı pil yönetim sistemlerinin ortaya çıkmasına ön ayak olacaktır [71-72, 97].

3. MATERYAL VE METOTLAR

Bu tez çalışmasına başlanırken bir olay kurgusu oluşturulmuştur. Bu olay kurgusu üzerine tüm yol haritaları oluşturulmuş ve çalışmalar yapılmıştır. Materyal ve metotlar başlığı altında öncelikle tezin olay kurgusu anlatılacak ardından bu kurgu üzerine inşa edilmiş tüm materyal ve yöntemlerden bahsedilecektir. Bu materyal metot uygulama çalışmalarına ayrıca detaylı olarak deneysel çalışma başlığı altında değinilecektir.

3.1 Tezin Uygulama Kurgusu

Bu sistem AKİBA otonom yük taşıma robotu için ihtiyaç olma durumundan esinlenilerek gerçekleştirme planı yapıldı. AKİBA aracı Elfatek Elektronik tarafından geliştirilen otonom yük taşıma sistemidir. Bu yük taşıma sistemi fabrikalar ve depolar gibi endüstriyel ortamlarda insanlar ile tam uyumlu bir şekilde çalışabilme yeteneğine sahiptir. AKİBA'nın batarya sistemleri cihazın güvenilir ve kesintisiz çalışması açısından kritik bir öneme sahiptir. Bu çalışma AKİBA otonom aracında kullanılmak üzere bir Batarya Yönetim Sistemi geliştirmeyi ve bu sisteme uzaktan veri Takip-Kontrol özellikleri kazandırmayı hedeflemektedir. AKİBA otonom aracı görseli Şekil 3.1'de gösterilmektedir [121].



Şekil 3.1 : AKİBA otonom aracı görseli [121].

Birçok otonom araçta olduğu gibi AKİBA otonom aracı da batarya sistemleri ile çalışmaktadır. Batarya ile çalışan cihazlarda batarya sadece cihazın çalışması için

yeterli gözükse de bataryanın sağlığı ve cihazın güvenli çalışması ve bunlar gibi etkenlerden dolayı batarya yönetim sistemlerine de ayrıca ihtiyaç vardır.

Bu tez ile otonom yük taşıma robotu olan AKİBA ürünün batarya yönetim sistemini hem yerli olarak tasarlamış olacak hem de yenilikçi bir yaklaşımla ürünün tüm bu batarya verileri bulut ortamına taşınacaktır. Bulut ortamına taşınan bu batarya verileri işlenerek hem bilgisayar hem de telefon üzerinden takibi yapılabilecek şekilde yazılım ve görsel tasarımları gerçekleştirilecektir. Bu tasarım sayesinde hem telefon ile hem de bilgisayar ile anlık takipler dünyanın istenilen yerinde istenilen şekilde anlık şekilde yapılabilecektir. Bu tasarımlara ek olarak e-posta gönderme işlemi eklenecektir. Bu sayede herhangi bir acil durum gerçekleştiğinde e-posta ile kullanıcı bilgilendirilecektir. Sadece acil durum değil eğer kullanıcı o zaman dilimindeki tüm verileri herhangi bir yerde kullanmak üzere isterse değerleri gönder butonu ile belirlenen e-posta adresine butona basıldığı andaki verileri e-posta olarak alabilecektir. Ayrıca kritik rol oynayan batarya yönetim sistemi parametreleri kullanıcı isteklerine değiştirilebilir bir sistem planlamaktadır. Örneğin şarj durumundaki maksimum ulaşması gereken sıcaklık ayarları değiştirilebilir olacak ve kontroller sadece otonom aracın kontrolüne bırakılmamış olacak.

Otonom araçların batarya sistemlerinde verimlilik önemli olduğu kadar güvenlik de önemlidir. Her zaman kararları otonom araç kontrolcüsüne bırakmak doğru olmayabilir. Bazı durumlarda bir insanın uzaktan takip ederek veya kontrol ederek güvenlik önlemi alması gerekebilir. Bu durumlar yöneticinin veya kullanıcının herhangi anormal durumları sezdiğinde takip edilebilecek şekilde olan programları çalıştırıp acil durum butonuna tıklayarak otonom aracı önceden belirlenen ve insan sağlığını etkilemeyecek bir noktada güvenli olarak beklemeye alabilecek bir durum oluşturmak amaçlanmaktadır. Yani aslında herhangi tehlikeli bir durum sezildiğinde batarya sisteminin yüksek sıcaklıklarda yangın çıkarma ihtimali kontrolcülerin dışında kullanıcının da kontrolü yapıp inceleyip cihaz alev alacaksa da önceden belirlenen örneğin fabrikada yangının büyüemeyeceği noktaya uzaktan kontrol edilen acil durum butonuyla hareket etmesi sağlanacak olup tehlikeleri minimuma indirmek hedeflenmektedir.

Bu tarz otonom araçlarda da acil durum butonu cihaz üzerinde bulunmaktadır. Yani cihazı tamamen durdurma işlemi veya başka komutlar yaptırılabilir. Fakat bu durumda cihaza bir insanın yaklaşması gerekmekte olup tam butona basılacağında tehlike arz

eden durum insana sađlıđına ciddi hasarlar bırakabilir. Bu düşünce ile tez çalışması sayesinde otonom aracın sađlıklı ve tehlikeden uzak olması için yenilikçi bir adım daha atılmış olacaktır.

3.2 Çalışma Yol Haritası Oluşturma

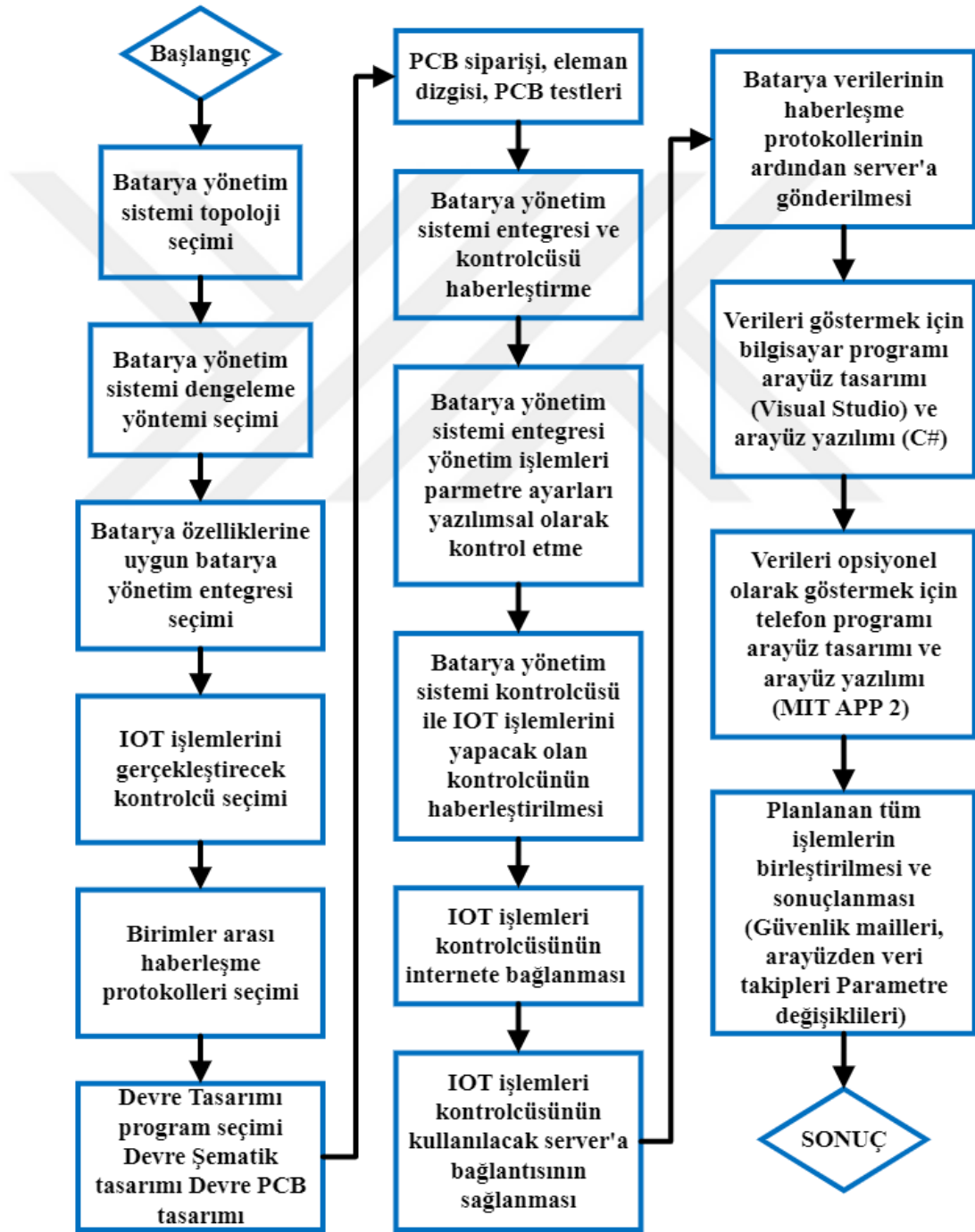
Otonom aracın batarya yönetim sistemi tasarlanırken bir yol haritası çizebilmek adına ve genel yapının oturtulabilmesi için bir yöntem haritası ve yöntem tasarımı ile çalışmalara başlandı. Otonom aracın batarya yönetim sistemi tasarlanırken bir yol haritası çizebilmek adına ve genel yapının oturtulabilmesi için bir yöntem haritası ve yöntem tasarımı ile çalışmalara başlandı.

İlk olarak Batarya Yönetim Sistemi Topolojisi Seçimi gerçekleştirilme planlaması yapılmış ve sistemin temel çalışma yapısı belirlenmiştir. Ardından Batarya Yönetim Sistemi Dengeleme Yöntemi Seçimi adımıyla yani batarya hücreleri arasındaki enerji dengesini sağlamak için kullanılacak yöntem seçilme planı oluşturulmuştur. Bu süreçte batarya özelliklerine uygun Batarya Yönetim Entegresi Seçimi yapılması planlanarak sistemin donanım kısmında kullanılacak olan entegre belirlenmiştir.

Sistemin akıllı veri takibi ve iletişimini sağlamak adına IoT işlemlerini gerçekleştirecek kontrolcü seçimi yapılması akabinde birimler arası haberleşme protokolleri seçimi aşaması planlanmıştır. Ardından devre tasarımı program seçimi, devre şematik tasarımı ve devre PCB tasarımı planlaması gerçekleştirilerek sistemin fiziksel altyapısı planlanacaktır. Bu tasarım sürecinin ardından PCB siparişi eleman dizgisi ve PCB testleri yapılarak donanım kısmı ve donanımın devreye alınması ile batarya yönetim sistemi entegresi ve kontrolcüsü haberleştirme aşaması başlatılacak şekilde planlama yapılmıştır. Bu noktada batarya yönetim sisteminin parametre ayarları kontrol edilmiş ve sistemin yazılımsal entegrasyonu sağlanmıştır.

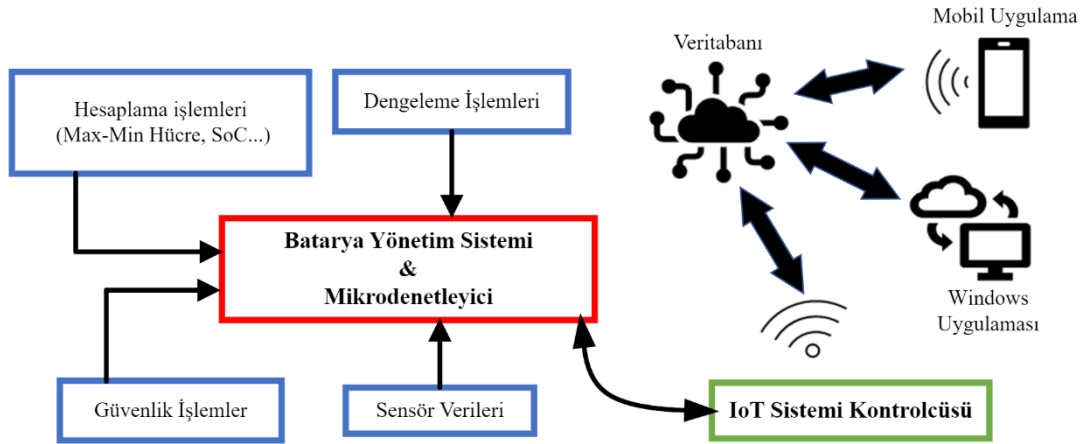
Bir sonraki aşamada batarya yönetim sistemi kontrolcüsü ile IoT sistemleri arasındaki iletişim kurulması ardından IoT işlemleri kontrolcüsünün internete bağlanması ve server'a bağlantısının sağlanması planlandı. Bu sayede veri yönetimi sürecinde batarya verilerinin haberleşme protokolleri aracılığıyla veri tabanına gönderilmesi gerçekleştirilmesi planlandı. Görsel analiz ve takip amacıyla Bilgisayar Program Arayüz Tasarımı ve Yazılımı (Visual Studio ve C#) ile bir yazılım geliştirildi. Ek olarak mobil cihazlar üzerinden de erişim sağlanabilmesi için Telefon Program Arayüz

Tasarımı ve Yazılımı (MIT App Inventor) kullanılarak mobil yazılım hazırlandı. Son olarak sistemin tüm işlemlerinin entegrasyonu sağlandı, güvenlik mail gönderimi, arayüzden veri takipleri ve parametre değişiklikleri ile süreç tamamlandı. Bu yöntem haritası batarya yönetim sisteminin sistematik ve sürdürülebilir bir şekilde planlanmasına olanak sağlamıştır. Bu yol haritasının özet bir diyagramına Şekil 3.2’de yer verilmiştir. Diyagramdan görüleceği üzere süreç aşamalı bir planlama çerçevesinde ilerlemiştir.



Şekil 3.2 : Yol haritası diyagram görseli.

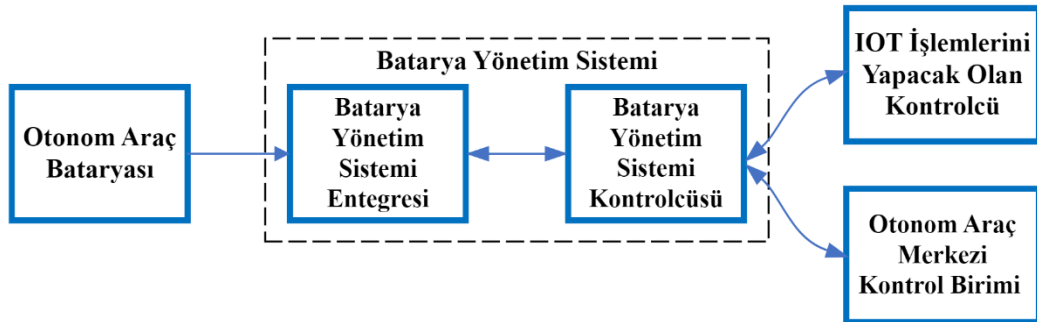
Tüm bu yol haritası çizildikten sonra oluşan ve tasarlanan sistemin kısa bir blok diyagramına Şekil 3.3'te yer verilmiştir.



Şekil 3.3 : Sistemin temel blok diyagramı.

3.3 Batarya Yönetim Sistemi Tasarımı Planlaması

Batarya yönetim sisteminin temel yapısının yol haritasının çizilebilmesi için seçilecek olan materyaller ve yöntemlere karar verilmek üzere planlama yapıldı. Bu seçilecek temel yapıların anlaşılır ve bütünlük sağlayabilmesi adına ön detaysız bir diyagram adımları oluşturuldu. Bu diyagrama Şekil 3.4'de yer verilmiştir.



Şekil 3.4 : Batarya yönetim sistemi temel yapı diyagramı.

3.3.1 Topoloji seçimi

Otonom aracın batarya yönetim sistemi tasarımına uygun bir topoloji ve dengeleme yönteminin seçimi ile başlandı. Bu bağlamda literatür başlığı altında anlatıldığı gibi batarya yönetim sistemi için uygun topolojiler arasında merkezi, modüler ve dağınık topoloji bulunmaktadır. Merkezi topoloji tüm görevlerin tek bir kart üzerinde birleştirilmesiyle maliyeti düşük tutarken modüler topoloji birden fazla elektronik karta bölünmüş işlemleri içerir ve maliyetlidir.

Dağınık topolojide ise her bir batarya hücresine bir adet uydu kontrolcü kart kullanılır ve maliyet yüksektir. Tez kurgusu AKİBA otonom aracı üzerinden ilerlediği için ve maksimum hücre sayısı olarak 16 adet hücre yapısı planlandığından maliyetler ve yeterlilik durumları gözden geçirilmesinin ardından otonom araç için merkezi topoloji tasarımı yapılacak şekilde planlama yapılmıştır.

3.3.2 Dengeleme yöntemi seçimi

Literatür araştırması başlığı altında detaylı incelenebileceği gibi dengeleme yönteminden biri olan pasif dengeleme en düşük hücre gerilim seviyesini belirleyerek diğer hücreleri buna indirir ve maliyet açısından avantajlı bir seçenektir. Aktif dengeleme ise karmaşık devreler içerir ve pahalıdır.

AKİBA otonom aracı için batarya yönetim sistemi tasarımında en uygun seçenek tek bir batarya için bahsi geçen avantajlardan dolayı merkezi topoloji ve pasif dengeleme yöntemi kullanılacak şekilde planlama gerçekleştirildi. Hem merkezi topoloji maliyeti düşük tutmuş olacak hem de pasif dengeleme basit ve maliyet açısından avantajlı bir dengeleme yöntemi ile efektif bir tasarım planlamış olacaktır. Bu kombinasyon otonom aracın batarya yönetim sisteminin etkili ve verimli bir şekilde çalışmasını sağlayacaktır. Özetle araştırmaların gözden geçirilmesinin ardından son karar olarak batarya yönetim sistemini tasarlamaya merkezi topoloji ve pasif dengeleme üzerinden başlanacak şekilde planlama gerçekleştirildi.

3.3.3 Batarya yönetim sistemi entegresi seçimi

Batarya yönetim sistemi birçok işlemi bünyesinde barındırmaktadır. Bu durumda bir kısım işleri batarya yönetim sistemi entegresi olarak adlandırılan kontrolcüler üstlenmektedir.

Otonom araç için batarya yönetim sisteminde hangi batarya yönetim sistemi entegresi kullanılacağı ve bu entegre hangi kontrolcü ile haberleştirilmesi en avantajlı olduğu konusundaki araştırmalar ile süreç başlatıldı. Literatür araştırmaları başlığı altında incelenebileceği üzere batarya verilerini okuyacak, dengeleme sistemlerini gerçekleştirecek ve koruma kontrollerini yapacak batarya yönetim sistemi entegreleri karşılaştırıldı. Bu karşılaştırılması yapılan entegrelerin içlerinden birkaç tanesi seçilerek tekrar bir sınıflandırma yapıldı ve bu seçimler Çizelge 3.1’de karşılaştırıldı. Entegre araştırmalarında piyasadaki en çok tercih edilen ürünler Analog Device ve

Texas Instruments firmalarının ürünleri olduğu gözlemlenmiştir. Bu firmaların bazı ürünleri Çizelge 3.1’de gösterilmiştir. Diğer ürün referans ve çeşitleri literatür araştırması başlığı altında incelenebilir.

Çizelge 3.1 : Batarya yönetim sistemi entegre seçim karşılaştırma tablosu.

Entegre ismi	Ölçülebilen Hücre	Sıcaklık kanalı	Hata Payı	Referans
LTC6803-2	12	2	%25	[107]
LTC6811-1	12	5	1.2mV	[98]
BQ76PL536A	6	2	3 mV	[108]
BQ76952	16	9	< 10 mV	[122]
BQ76972	16	9	±3.0 mV	[101]

Genel hatlarıyla yapılan araştırmalar sonrası batarya yönetim sistemi için maliyet, entegre yetenekleri, güncel olması, stok durumu ve yazılım örnekleri ile birçok kontrolün yapılabildiği entegre olan BQ76952 entegresi kullanılmaya karar verilmiştir.

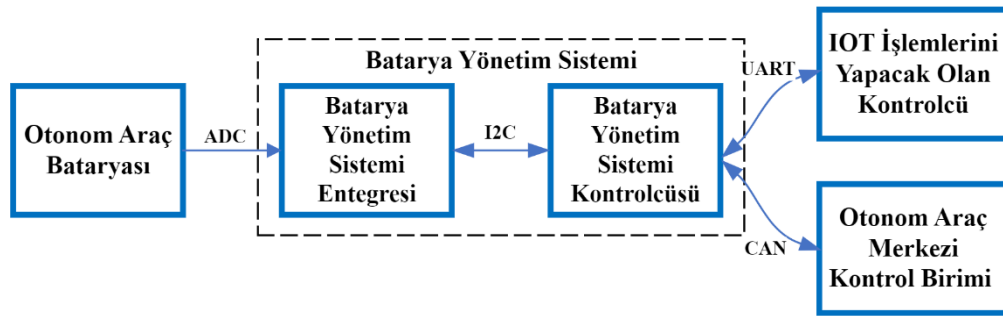
Bu entegre birçok batarya yönetim işlevini karşılayacak fakat bunları yapabilmesi için bu entegreyle haberleşecek ve komutlar göndererek parametre değişiklikleri yapılabilmesi için harici bir kontrolcüye daha ihtiyaç vardır. Bu kontrolcü sadece batarya yönetim sistemi entegresiyle haberleşme ve kontrolünü gerçekleştirmeyecek ayrıca ilerleyen süreçte batarya verilerini server ortamına gönderecek olan IoT kontrolcüsü ile haberleştirilecek şekilde planlama yapıldı. Ayrıca otonom araçta kullanılan CAN haberleşme protokolüne de bu verileri göndermekle sorumlu olan kontrolcü olacak şekilde planlama yapıldı.

Kontrol işlemlerini yapacak olan mikrodenetleyici seçiminden önce tüm birimlerin hangi haberleşme protokollerini kullanacağı ile ilgili ön planlama yapmak üzere araştırma yapıldı. Batarya yönetim sistemi tasarımında haberleşme protokolleri cihazlar arasında veri iletimini sağlayarak sistemin işlevselliğini büyük ölçüde etkiler. Bu protokoller arasında özellikle I2C, CAN, UART ve SPI oldukça yaygın olarak kullanılmaktadır. I2C düşük hızlı seri bir haberleşme protokolü olup, genellikle sensörlerin ve diğer entegre devrelerin kontrolünde tercih edilir. CAN endüstriyel ve otomotiv uygulamalarında kullanılan yüksek hızlı ve güvenilir bir protokoldür. Özellikle araç içi iletişim sistemlerinde yaygın olarak kullanılır. UART bilgisayarlar ve mikrodenetleyiciler arasında seri haberleşme sağlayan temel bir protokoldür. SPI

ise genellikle yüksek hızlı veri transferi gerektiren uygulamalarda tercih edilir ve birden fazla cihazın haberleşmesini sağlar [123]. Bu haberleşme protokollerinin doğru seçilmesi batarya yönetim sisteminin performansı, güvenilirliği ve genel işlevselliği açısından son derece önemli olacağından planlamalar bu özveriyle gerçekleştirildi.

Bu araştırmalar sonucunda seçilen Texas Instrument markasının BQ76952 entegresinin kullanılacağı kararlaştırılmasının ardından haberleşme protokolleri belli olmuş oldu. Bu bağlamda batarya yönetim sistemi ana kontrolcüsünün BQ76952 entegresi ile haberleşmesinin I2C olması ve bu kontrolcünün otonom araç içerisinde haberleşmesi için araç için kullanılan CAN haberleşmesinin kullanılması planlanmıştır. Son olarak da IoT işlemlerini yapacak olan kontrolcü ile haberleşmenin de UART haberleşme protokolü ile yapılması planlanmıştır.

Yani bu sistemde planlanan ana kontrolcü batarya yönetim sistemi entegresiyle I2C haberleşme protokolüyle haberleşecek şekilde planlama yapıldı. Aynı zamanda IoT işlemlerinin gerçekleştirilmesi için UART haberleşme protokolünü kullanabilecek bir yapı oluşturulması planlandı. Bu yapı diyagramının güncellenmiş haline Şekil 3.5'te yer verilmiştir. Ayrıca batarya yönetim sistemi entegresinin analog dijital dönüştürücü (ADC) kullanarak batarya verilerini okuyacağı bilgi de diyagrama eklenmiştir.



Şekil 3.5 : Haberleşme protokolleri güncellenmiş diyagram.

3.3.4 Batarya yönetim sistemi ana kontrolcüsü seçimi

Seçilecek olan ana kontrolcünün özellikleri bir önceki başlık altında değinildiği gibi hem I2C hem UART hem de CAN haberleşme protokollerini destekleyebilecek özelliklere sahip olması etkenleri göz önünde bulunduruldu. Bu bağlamda kontrolcü araştırmaları yoğunlaştırıldı. Birçok mikrodenetleyici üretici firması bulunmaktadır. Bu firmalar arasından en yaygın olan firmalardan olan STMicroelectronics, Atmel Corporation ve Microchip firmaları kullanılacak olan özellikleri açısından karşılaştırılma yapılmıştır. Ayrıca bu firmalardan seçilen bir mikrodenetleyici ile

karşılaştırma biraz daha anlaşılır olması sağlanmıştır. Bu karşılaştırmalara Çizelge 3.2’de yer verilmiştir.

Çizelge 3.2 : Kontrolcü karşılaştırma tablosu [124-126].

Özellik	STM32G431	PIC18F4550	ATmega328P
Stok	Aktif	Aktif	Önerilmiyor
Çekirdek	32-bit Arm	8-bit PIC	8-bit AVR
CPU Frekansı	170 Mhz	40 MHz	16 MHz
Haberleşme	ADC, USART, SPI, I ² C, CAN	ADC, USART, SPI, I ² C	ADC, USART, SPI
Platform	STM32CubeIDE	Atmel Studio	MPLAB X IDE
Canlı Veri İzleme	Var	Yok	Yok

Yapılan detaylı incelemeler ve ihtiyaç analizleri sonucunda STM32G431 serisi mikrodnetleyicinin projede kullanılmasına karar verildi. Bu seriyi tercih etme sebebi yüksek işlem gücü, hızlı veri işleme kapasitesine sahip olmasıdır. Ayrıca geniş haberleşme protokolü desteği (ADC, USART, SPI, I²C ve CAN) ile proje gereksinimlerini tam anlamıyla karşılamaktadır.

STM32CubeIDE geliştirme platformu sayesinde kullanıcı dostu bir yazılım ortamı sunarak geliştirme sürecini hızlandırmakta olması da bu kontrolcünün avantajlarından. STM32CubeIDE geliştirme platformu canlı veri izleme (Live expression) özelliği ile projeye yüksek oranda erişebilirlik ve esneklik kazandırmaktadır. Tüm bu özellikler doğrultusunda STM32G431’in tercih edilmesi planlandı.

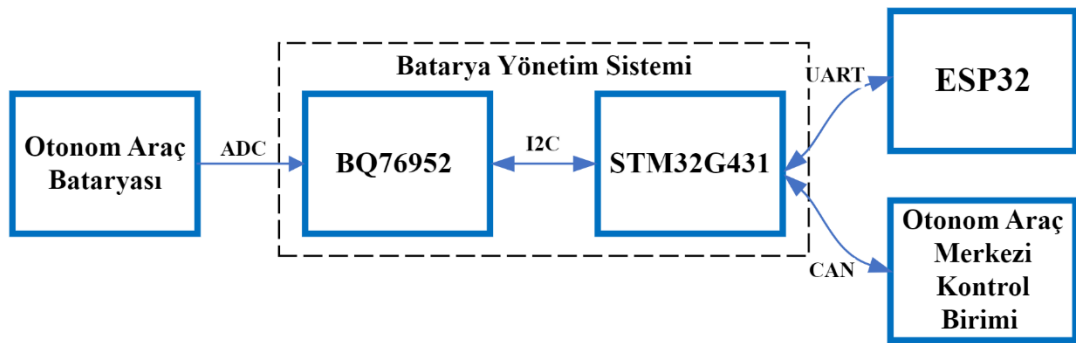
3.3.5 Batarya yönetim sistemi IoT kontrolcüsü seçimi

Mikrodnetleyici seçimi ardından IoT sistem yapıları araştırıldı. Dokümanlar ve internet siteleri araştırması sonucunda STM ile kablosuz olarak internet erişimine sahip olması birkaç yöntemin olduğu sonucuna varıldı. Bunlara çizelge 3.3’te yer verilmiştir [127]. Bu araştırmalar neticesinde ESP32’nin düşük maliyeti, geniş topluluk desteği ve kablosuz iletişim yetenekleri gibi özellikler tercih sebebi oldu. Özetle Çizelge 3.3’te gösterilen özellikler nedeniyle tercih edildi. Bu ESP32’nin seçimi tezin başarısını artırmaya yönelik stratejik bir adım oldu ve projenin gereksinimlerini en iyi şekilde karşılamayı hedefledi.

Çizelge 3.3 : IoT kontrolcüsü karşılaştırma tablosu [127-131].

Özellik	ATWINC1500	WF200	ESP8266	ESP32
Stok	Orta	Orta	Yüksek oranda	Her zaman
Maliyet	Orta	Düşük	Çok Düşük	Çok Düşük
Hız	Hızlı	Hızlı	Hızlı	Hızlı
Boyut	Küçük	Küçük	Küçük	Küçük
Haberleşme	SPI	SDIO, SPI	UART, SDIO	UART, SPI, SDIO
Yazılım Desteği	Kısıtlı	Var	Var	Çok fazla

Örneğin STM32'nin batarya yönetim sistemi çevresel birimleriyle elde ettiği batarya verilerini ESP32'ye göndermesinin ardından ESP32 bu verileri veri tabanına iletimini üstlenmektedir. Bu veri tabanındaki veriler daha sonra bilgisayar ve telefon programları üzerinden takip edilebilecek şekilde planlama yapıldı. Ayrıca bilgisayardan bataryanın parametre ayarları kontrol edilebilecek şekilde yazılım planlandı. Burada en büyük avantaj ESP32'nin geliştirme topluluklarının yazılım ve örnek desteğini çok fazla olmasıdır. Bu seçilen tüm kontrolcülerin diyagramda tekrar düzenlenerek gösterilmiş haline Şekil 3.6'da yer verilmiştir.



Şekil 3.6 : Haberleşme protokolleri güncellenmiş diyagram.

3.3.6 Donanım tasarım programı seçimi

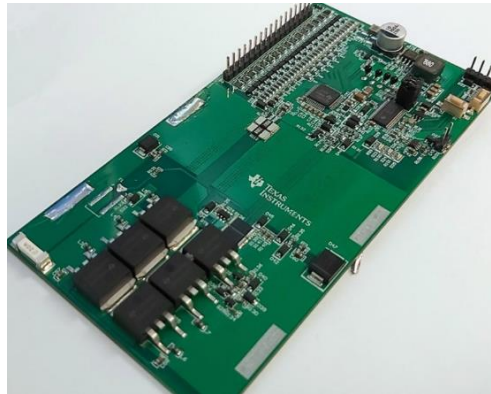
Projede kullanılacak olan temel ve görevleri en fazla olan entegre ve kontrolcülerin belirlenmesinin ardından somut olarak projenin ortaya çıkmasında en önemli rollerden birine sahip olan donanım tasarımı ve yazılım kısmına adım atmak üzere donanım tasarım programlar karşılaştırmaları yapıldı. Bu karşılaştırmalara Çizelge 3.4'te yer verilmiştir.

Çizelge 3.4 : Donanım tasarım programı karşılaştırması [132-136].

Özellik	KiCad	Altium	EAGLE	Proteus
Lisans	Ücretsiz Açık Kaynaklı	Abonelik lisans gerekir	Abonelik lisans gerekir	Abonelik lisans gerekir
Kullanım Kolaylığı	Kullanıcı Dostu	Profesyonel Arayüz	Kullanıcı Dostu	Kullanıcı Dostu
Simülasyon	Var	Var	Var	Var
Şematik Tasarım-PCB Tasarım- 3D Görüntüleme	Var	Var	Var	Var
Topluluk Desteği	Çok geniş kullanıcı topluluğu	Profesyonel kullanıcı tabanı	Geniş kullanıcı topluluğu	Geniş kullanıcı topluluğu

Karşılaştırma çizelgesinin ardından tasarımın KiCad üzerinden yapılması planlanmıştır. KiCad donanım tasarım programları arasında tercih edilmesinin ana nedenleri arasında ücretsiz ve açık kaynaklı olması kullanıcı dostu arayüzü, geniş özellik yelpazesi ve kapsamlı topluluk desteği bulunmaktadır. Bu özellikler KiCad'i tez çalışmasında kullanmak için ideal bir seçim sağlamıştır.

Tasarım programının seçilmesinin ardından donanım tasarımı gerçekleştirmek üzere araştırmalara başlandı. Tasarıma başlarken kullanılacak olan BQ76952 entegresinin dokümanları incelendi. Bu dokümanlar içerisinde tasarım veren ve ayrıca örnek tasarım dokümanları incelendi [122]. Bu dokümanlar aracılığıyla tasarıma kademe kademe başlandı. İlk etapta herhangi bir tasarım hatası oluşmaması adına örnek tasarım üzerinden tasarım planlamaları gerçekleştirildi. Şekil 3.7'de Texas Instruments firmasının örnek tasarımının görseline yer verilmiştir.



Şekil 3.7 : Texas Instruments BQ76952 örnek tasarım [122].

3.4 Veri Tabanı Platform Seçimi

Batarya yönetim sisteminin işlenen verilerini veri tabanına iletecek olan ESP32 internet bağlantı yapısı araştırıldı. Bu bağlantının ardından internet veri tabanı platformlarından hangisinin tercih edilmesinin avantajlı olacağı noktasındaki araştırmalar yapıldı. İnternet ortamında birçok veri tabanı platformu bulunmaktadır. Birkaç tane veri tabanı platformu araştırıldı ve bu platformalar Çizelge 3.5'te karşılaştırıldı.

Çizelge 3.5 : Veritabanı platformu karşılaştırma tablosu [137-142].

Özellik	Türü	Özellik
Firebase	Realtime NoSQL	Kolay entegrasyon, çapraz platform desteği, gerçek zamanlı senkronizasyon.
Microsoft Azure	SQL, NoSQL	Küresel dağıtım, farklı veri modelleri
Amazon DynamoDB	NoSQL	AWS ekosistemiyle güçlü entegrasyon, ölçeklenebilirlik, düşük gecikme.
MongoDB	NoSQL	Esnek sorgulama, çapraz platform desteği, veri replikasyonu ve yedekleme.
Couchbase	NoSQL	Yüksek performans, çevrimdışı çalışma desteği, JSON tabanlı veri saklama.
Supabase	SQL	Firebase'e benzer özellikler, açık kaynak olması, PostgreSQL tabanlı güçlü SQL desteği.

SQL ve NoSQL veri tabanları, dijital verileri depolamak ve yönetmek için kullanılan iki farklı yaklaşımdır. SQL veri tabanları, ilişkisel veri tabanları olarak da bilinir verileri tablolar halinde düzenler ve bu tablolar arasındaki ilişkileri tanımlar. SQL bu veri tabanlarını sorgulamak ve yönetmek için kullanılan standart bir dildir. NoSQL veri tabanları ilişkisel olmayan veri tabanlarıdır ve daha esnek bir yapı sunarlar. Verileri tablolar yerine değer, grafik veya geniş sütun gibi farklı biçimlerde depolayabilirler. Bu esneklik NoSQL veri tabanlarını büyük veri, sosyal medya verileri veya IoT verileri gibi yapılandırılmamış verileri depolamak için daha uygun hale getirir [143]. Araştırmalar sonucunda gerçek zamanlı veri yönetimi, anlık senkronizasyon gereksinimleri ve Google ekosistemine entegre edilebilirliği gibi güçlü avantajları nedeniyle Firebase tercih edilmiştir. Firebase kolay entegre edilebilir yapısı, çapraz platform desteği ve kullanıcı dostu arayüzüyle projede veri yönetim sistemine ideal bir çözüm sunmaktadır.

Özellikle daha önce yapılan çalışmalar incelendiğinde Firebase kullanıldığı görüldü. Yapısal olarak topluluk desteklerinin çok olduğu görüldü. Bu nedenlerle Firebase proje gereksinimlerini karşılayacak en uygun seçenek olarak belirlenmiştir. Firebase tez sürecinde ESP32 mikrodenetleyicisi ile gönderilen verilerin anlık olarak işlenmesini ve izlenmesini mümkün kılması planlandı.

3.5 Bilgisayar ve Telefon Yazılım Platformu Seçimi

Bilgisayar arayüz tasarlama programları araştırılmaya başlandı. Bu araştırmalar sonucunda daha doğru kararlar verebilmek için araştırmalar sonucu için bir çizelge oluşturuldu. Bu çizelgede bilgisayar arayüz tasarım programları karşılaştırıldı. Bu karşılaştırmalara Çizelge 3.6'da yer verilmiştir.

Çizelge 3.6 : Arayüz platformları karşılaştırması [144-146].

Özellik	Visual Studio	Qt Creator	Electron
Desteklediği Diller	C#, Visual Basic, C++	C++, QML, Python	HTML, CSS, JavaScript
Platformlar	Windows	Windows, macOS, Linux	Windows, macOS, Linux
GUI Tasarımı	Evet	Evet	Evet (Web)
Kolaylık	Kolay	Orta	Orta
Geliştirme Ortamı	Gelişmiş	Orta	Orta
Lisans	Ticari / Ücretsiz	Ticari / Ücretsiz	Ücretsiz
Kullanım Alanı	Çok geniştir, özellikle Windows uygulamaları için tercih edilir.	C++ ağırlıklı projeler ve çapraz platform uygulamaları için uygundur.	Web tabanlı masaüstü uygulamaları geliştirmek için yaygın olarak kullanılır.

Araştırmalar sonucunda Visual Studio tasarım ve yazılım platformunun Windows sistem uygulamaları geliştirmesi yapmak için genellikle en iyi seçenek olduğu görülmüş oldu. Grafik kullanıcı arayüzü (GUI) tasarımı, geliştirme ortamı üstünlüğü ve geniş kullanım alanı gibi avantajlara sahip olması tercih edilmesinde önemli etken

olmuştur. Ayrıca daha önce yapılan Visual Studio ile Firebase çalışmalarının olması da bu sistemin tercih edilmesinde önemli rol oynamıştır.

Ardından opsiyonel olarak batarya hücre gerilimlerinin telefondan da görülebilmesi için tasarlanacak olan platformun seçimine geçildi. İnternet araştırmalarında birçok telefon uygulaması tasarlama platformu bulunmaktadır. Birkaç tane telefon uygulaması tasarlama platformu araştırılmıştır ve bu platformlar Çizelge 3.7’de karşılaştırılmıştır.

Çizelge 3.7 : Telefon uygulaması tasarlama platform karşılaştırması [147-152].

Platform	Lisans	Platformlar	Özellikler
MIT App Inventor	C#, Visual Basic, C++	C++, QML, Python	HTML, CSS, JavaScript
Thunkable	Windows	Windows, macOS, Linux	Windows, macOS, Linux
Android Studio	Evet	Evet	Evet (Web)
Kodular	Kolay	Orta	Orta
Bubble	Gelişmiş	Orta	Orta

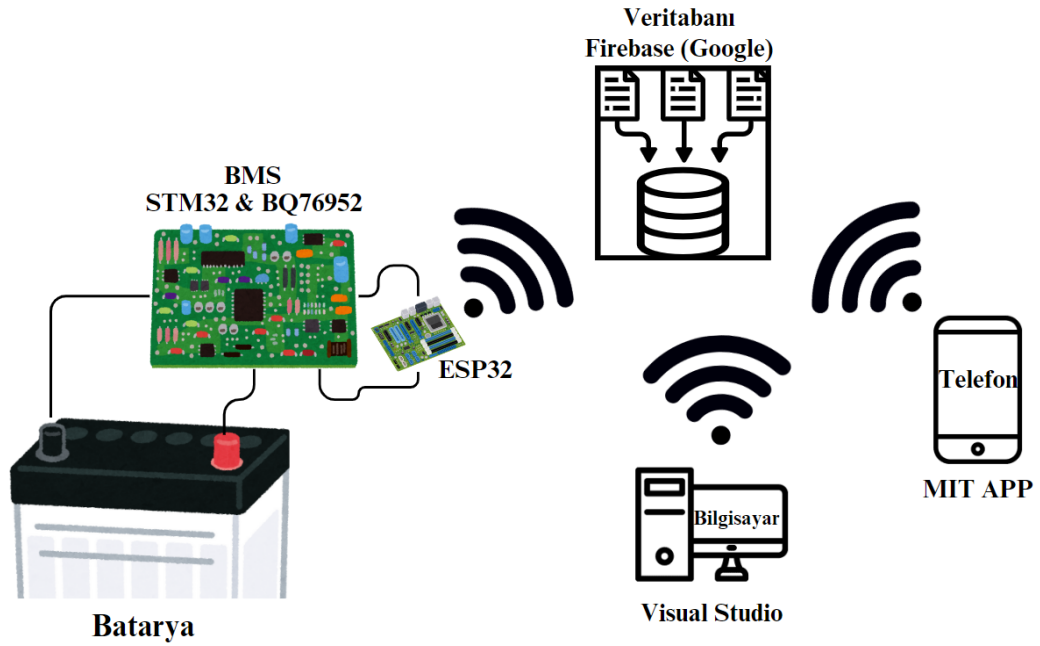
MIT App Inventor özellikle uygulama geliştirme sürecini basitleştiren ve kullanıcı dostu bir platform dikkat çekmiştir. Eğitim amaçlı ve başlangıç seviyesindeki kullanıcılar için ideal olduğu gözlemlenmiştir, çünkü görsel programlama arayüzü sayesinde karmaşık kodlama bilgisi gerektirmeksizin mobil uygulamalar geliştirmeyi mümkün kılmaktadır. Sürükle ve bırak yöntemiyle çalışan bir sistem sunar. Bu da kullanıcıların hızlıca projeler oluşturmalarına olanak tanır.

MIT App Inventor'ın avantajlarından biri de geniş bir kullanıcı kitlesine hitap etmesi olduğu görüldü. Prototip amaçlı uygulamalar için kullanılırken kolay öğrenilebilir yapısı ile uygulama geliştirme sürecini çok kolaylaştırır hale getirdiği gözlemlenmiştir. Diğer gelişmiş uygulama geliştirme platformlarına kıyasla daha basit yapısı ve eğitim amaçlı sunduğu içerik MIT App Inventor'ı öne çıkaran faktörler arasında yer aldığı görülmüştür. Ayrıca daha önce yapılan Firebase platformuyla olan çalışma örnekleri MIT APP platformunun seçiminde etken olmuştur.

4. DENEYSEL ÇALIŞMA

Bu başlık altında tüm tez sürecinde yapılmış olan deneysel çalışmalar anlatıldı. Batarya yönetim sistemi tasarımı ve üretim süreci anlatımı gerçekleştirildi. Ardından batarya yönetim sisteminin temel testleri ve güvenlik önlem testleri gerçekleştirme durumu anlatıldı. Daha sonra batarya yönetim sistemi verilerinin haberleşme protokolleriyle alınıp ardından veri tabanına gönderecek olan ESP32'nin internet bağlantısı ve Firebase bağlantısı açıklandı. Daha sonra masaüstü ve telefon uygulamalarının gerçekleştirilme aşamaları detaylandırıldı. Son olarak tüm sistemin çalışma yapısı gösterildi.

Materyal ve yöntem başlığı altında anlatılan tüm yapılar gerçekleştirilmiştir. Bu kurulacak olan yapının temel bir görseline Şekil 4.1'de yer verilmiştir.



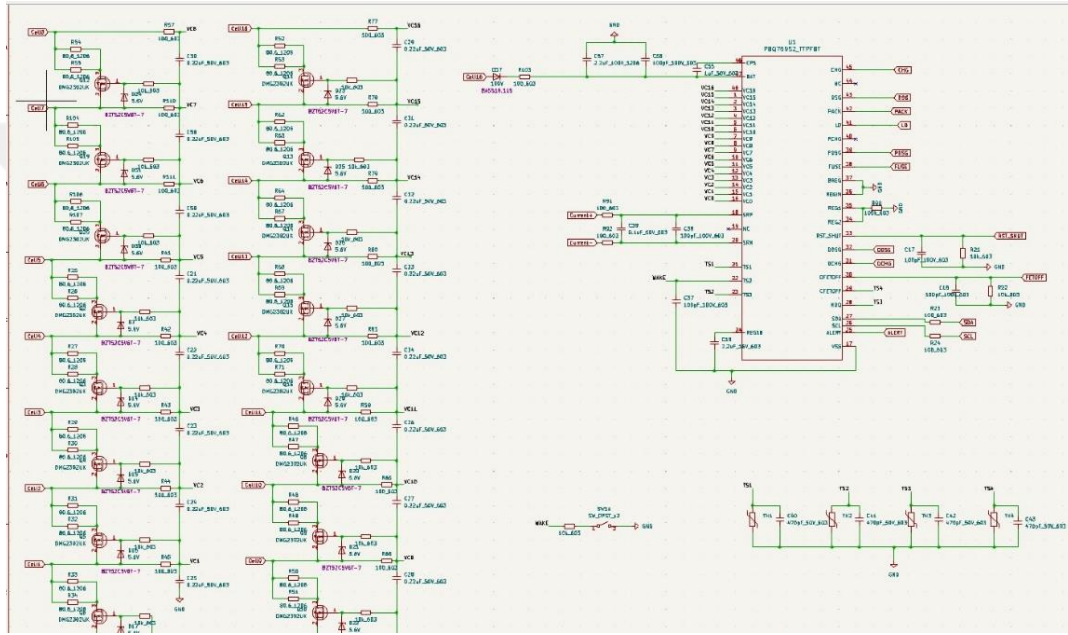
Şekil 4.1 : Tüm sistemin genel yapısı görseli.

4.1 Batarya Yönetim Sistemi Tasarımı Deneysel Çalışma

Batarya yönetim sisteminin tasarımına materyal ve yöntemler başlığı altında bahsedildiği gibi BQ76952 entegresinin dokümanları incelendi. Bu dokümanlardaki

örnek tasarımlar incelenerek öncelikle devrenin şematik tasarımı yapıldı. Bu şematik tasarımın küçük bir kısmının görseline Şekil 4.2’de yer verilmiştir.

Bu tasarım planlanan yapıda olduğu gibi merkezi topoloji, pasif dengeleme ve BQ76952 entegresiyle yapılmıştır. Bu tasarımda ayrıca ana kontrolcü olan STM32G431 olacak şekilde fakat ESP32 haricen konnektörler ile bağlanacak şekilde planlama yapıldı ve yöntem başlığında bahsedildiği gibi tasarım KiCad tasarım platformu üzerinden gerçekleştirildi. Her bölümün şematik tasarımı görsel olarak eklenmedi. Örnek şematiklere 4.2, 4.3, 4.4, 4.5 ve 4.6’daki şekillerde yer verilmiştir.



Şekil 4.2 : BQ76952 ve dengeleme yapısı önerilen tasarımın gerçekleştirilmesi.

Dengeleme yapısı kurulurken tasarım önerilerinde olduğu gibi mosfet ve direnç yapısı kuruldu. Direnç değerleri yaklaşık 100mA geçebilecek şekilde planlama yapılarak iki adet direnç paralel olarak eklenerek direnç değerleri 80.6 ohm seçildi. İki adet direnç paralel olmasının sebebi akımın paylaşarak daha küçük kılıftaki dirençlerin kullanılmasını sağlamaktır, direnç kılıf değeri 1206 olarak seçilmiştir. Buradaki 100mA değeri hesaplanması denklem 4.1’de görüldüğü üzere ohm kanunu ile hesaplanmaktadır.

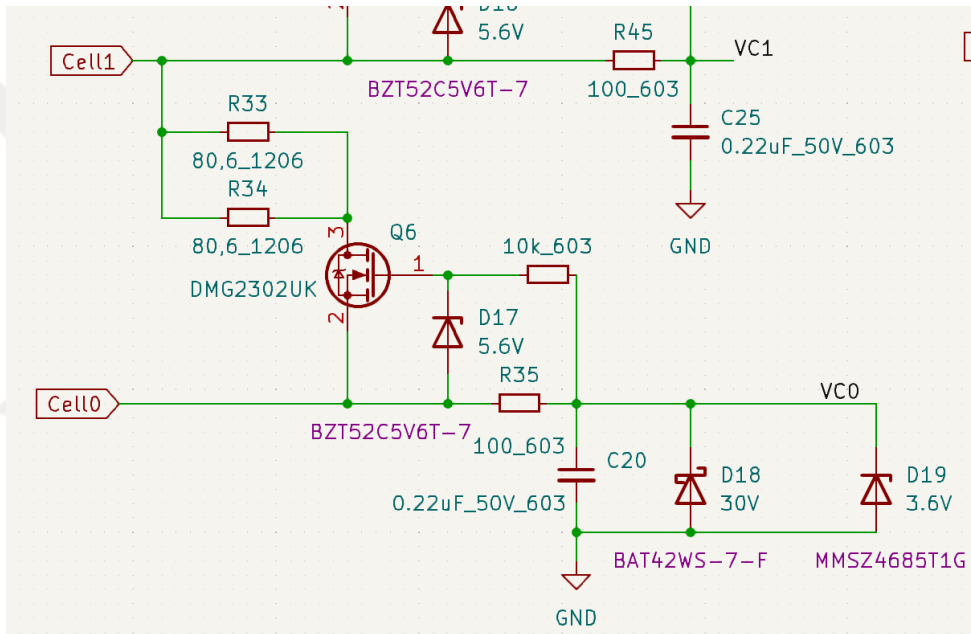
$$V = I \times R \quad (4.1)$$

Denklem 4.1’de görüldüğü üzere V gerilimi, I akımı ve R direnç değerini ifade eder. Bu direnç değeriyle beraber tasarlanmış olan mosfetli yapıya Şekil 4.3’te yer verilmiştir.

Pil gerilimi 4.3V olacak şekilde işlem 4.2’de görüldüğü gibi dirençler üstünden yaklaşık 1A değerlerinde bir akım geçecek şekilde hesaplandı.

Burada direnç değeri 80.6 fakat paralel iki direnç olduğu için eşdeğer direnç değeri 40.3 olmaktadır. İşlem 4.2’de görüldüğü gibi akım değeri yaklaşık 100mA olacak şekilde hesaplanarak direnç kılıf değeri de bu şekilde belirlenmiştir. Örneğin maksimum pil gerilimi 4.3V olacak şekilde bu hesaplamanın sonucu bahsedildiği üzere yaklaşık 100mA hesaplandı.

$$4.3 = I \times (40.3) \rightarrow I = \frac{4.3}{40.3} = 0,107 \text{ A} \quad (4.2)$$



Şekil 4.3 : Dengeleme tek hücre için örnek.

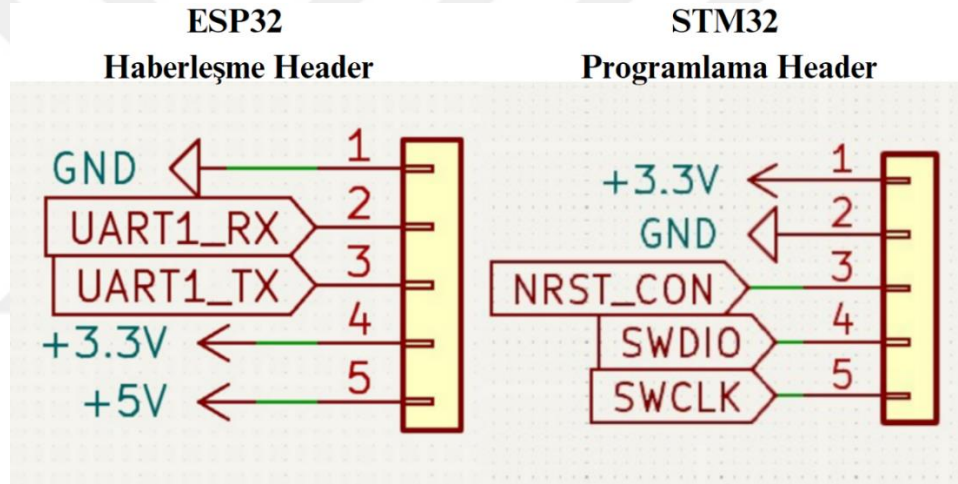
Tasarımda ek olarak batarya yönetim sistemi kontrolcüsü STM32G431 şematik tasarımı, güç giriş çıkış, ön şarj (precharge), ön deşarj (predischarge), ters bağlantı koruması şematik tasarımı, batarya yönetim sistemi kontrolcüsü ve IoT kontrolcüsünün besleme için (3.3V → STM32, 5V → ESP32) gerekli şematik tasarımı, Otonom araç ile haberleşmesi için gerekli olan CAN haberleşmesi şematik tasarımı eklenmiştir.

BQ76952 ve STM32G431 birbirleriyle I2C ile haberleşeceğinde prototip devre üzerinde bağlantının kolayca sağlanması gerektiğinde iptal edilip testlere tabi tutulması için pin header olarak adlandırılan konnektör şematik tasarımı gerçekleştirilmiştir.

Devrede bulunmayacak olan IoT kontrolcüsünün haricen devreye bağlanıp STM32G431 ile haberleşmesini sağlayacak olan konnektör çıkışının şematik tasarımı da eklenerek prototip olarak IoT işlemlerinin istendiğinde yapılacağı bir sistem kurulması hedeflenmiştir.

Yani aslında prototip devre tasarımında IoT kontrolcüsü olan ESP32'nin standartta devre üzerine eklenmesi değil haricen bağlantının yapılması şeklinde planlama yapılmıştır.

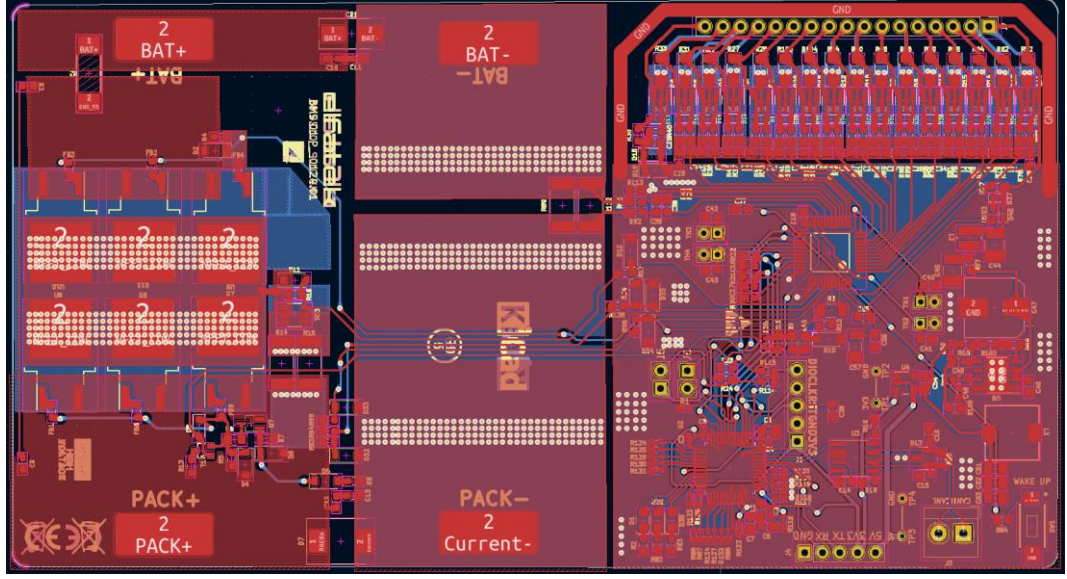
Ayrıca yazılımı ile BQ76952 entegresiyle haberleşecek ve parametre ayarları gönderecek olan STM32G431 mikrodenetleyicisinin programlayıcı aracılığıyla yazılımın yüklenmesinin kolay olması adına programlayıcı konnektörü şematik tasarımı gerçekleştirilmiştir. Bu şematik görsellerine Şekil 4.6'da yer verilmiştir.



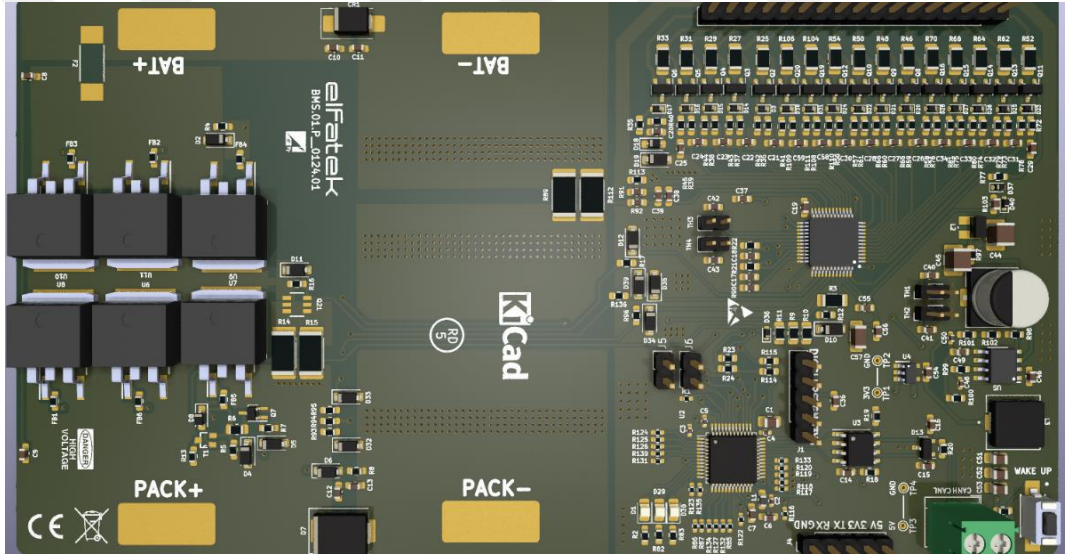
Şekil 4.6 : Programlama ve haberleşme konnektörü şematik tasarımı.

Şematik tasarımın bitirilmesinin ardından PCB tasarımı işlemlerine başlandı. Prototip bu PCB tasarımında herhangi bir hatayı minimize etmek için BQ76952 entegresinin dokümanı ve tasarım dosyalarındaki yerleşimlerinden destek alındı. Herhangi bir sorun olacaksa da tasarım kaynaklı bir sorun olmaması adına prototip olacak bu devre için böyle bir yöntem izlenmiştir.

Batarya yönetim sisteminin PCB tasarım görseline ve 3D görsellerine Şekil 4.7'de ve Şekil 4.8'de yer verilmiştir. PCB yapısı BQ76952 örnek tasarımlarından yararlanılmasının sebebi aslında ana hedef olan IoT işlemlerine yoğunlaşılacağına donanımsal hataların minimumda tutulmasını sağlamakla ilgilidir, bu sebeple bu şekilde bir yol izlenmiştir.



Şekil 4.7 : Batarya yönetim sistemi PCB tasarım görseli.



Şekil 4.8 : Batarya yönetim sistemi 3D görseli.

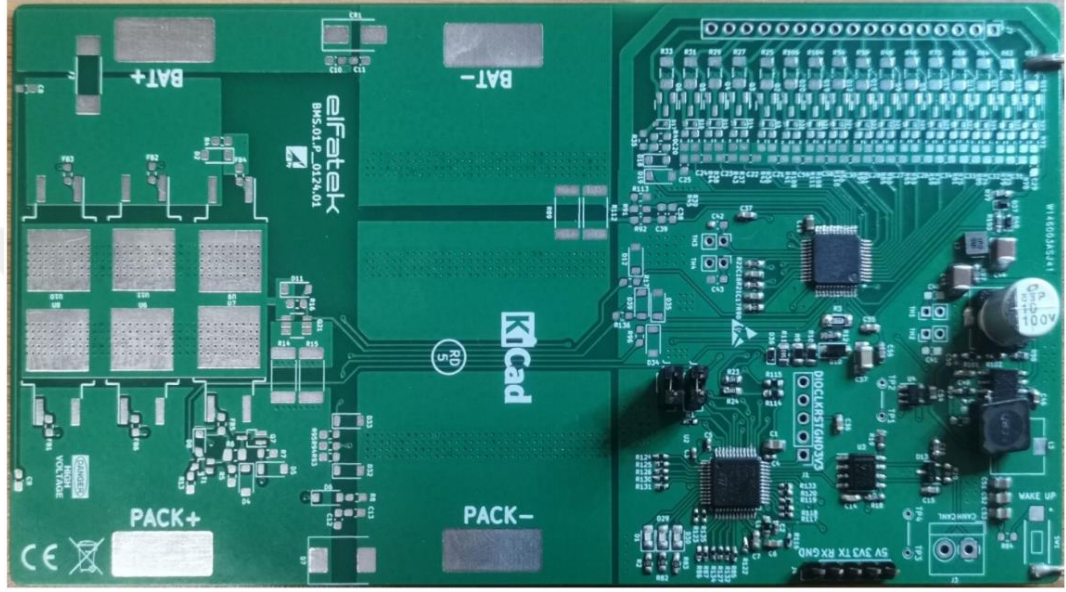
PCB tasarımları tamamlandıktan sonra PCB üretimi için tedarikçiler üzerinden siparişi verildi. Ve bu PCB siparişinin ardından batarya yönetim sisteminde kullanılacak olan komponentlerin siparişleri de işleme alınarak sürecin hızlanması sağlandı.

Bu siparişlerin teslim edilmesinde geçecek olan süre zarfında batarya yönetim sisteminin çalışması için gerekli komutlar ve haberleşme durumları dokümanlardan araştırılarak ürünlerin gelmesinin ve dizgisinin yapılmasının ardından süreci hızlandırmak amaçlı çalışmalara başlanmıştır.

Bu araştırmalar ve yapılar kurulması araştırmaları süresinde geçen süreç devam ederken bir yandan da batarya yönetim sistemi PCB siparişlerinin teslimatı gerçekleşti ve bu PCB testlerine hızlıca başlandı.

Devrenin tesliminin ardından Batarya yönetim sisteminde bulunan batarya yönetim entegresi ve batarya yönetim kontrolcüsünün enerjisini verecek olan güç dönüştürücüsü hattı dizilerek testlere başlandı.

Yani aslında ilk etapta 5V ve 3.3V çıkışı sağlayacak olan elemanların dizgisi yapılarak çıkışta doğru gerilimleri verip vermediği kontrol edildi. Şekil 4.9’da bu dizgisi yapılan devreye yer verilmiştir.



Şekil 4.9 : Temel işlemlerin yapılacağı elemanların dizgisi.

Batarya yönetim sistemi ana kontrolcüsü olacak olan STM32G431’in sorunsuz şekilde çalışıp çalışmadığı durumu kontrol edilmek üzere ilk testleri gerçekleştirildi. Bir led yanıp sönmeye üzerine kurulu olan ufak bir test üzerinden gidildi ve bu led yanması devam ederken ilerleyen süreçlerde çok kullanılacak olan hata ayıklama (debug) işlemi için de ayrıca test edilerek sorunsuz olacak şekilde çalıştığı görülmüştür.

Ayrıca Batarya Yönetim Sistemi entegresi olan BQ76952 ile haberleşme işlemlerinin sağlıklı bir şekilde gerçekleştirilmesi gereklidir. Bu haberleşme sayesinde BQ76952 entegresi bataryadaki tüm hücrelerin değerleri, bağlantısı yapılan tüm sensörlerden gelen sıcaklık değerleri ve parametre belirlemeleri gerçekleştirilmiş olacaktır.

Örneğin batarya yönetim sisteminin hangi sıcaklıkta, hangi gerilim seviyesinde veya ne kadarlık bir kısa devre süresinin ardından enerji aktarımının kesime girmesi gerektiği durumlarını ayarlayıp kontrol edilebilecek bir sistem olması sağlanmış olacaktır.

BQ76952 entegresi batarya yönetim sistemlerinde kullanılan çok hücreli pil paketlerinin gerilim, akım ve sıcaklık ölçümlerini yaparak güvenli ve verimli şarj ve deşarj işlemlerini sağlayan bir entegredir ve bu işlemleri birkaç dizi işlem ardından sayısal veri olarak haberleşme hattından gönderebilmektedir.

BQ76952 Entegresi Analog-Dijital Dönüştürücü (ADC) ham verilerini gerilim cinsine dönüştürme işlemlerini yapmaktadır. ADC sayıları belirli kazanç ve offset değerleri kullanılarak gerilim değerlerine dönüştürülür.

Örneğin, bir batarya hücresinin ADC verisi 2500 olarak ölçülmüş olsun.

$$V = (\text{ADC Ham veri} \times \text{Kazanç}) + \text{Offset} \quad (4.3)$$

Denklem 4.3'te görüldüğü üzere V gerilimi ifade etmektedir. Kazanç değeri 0.00125V/ADC varsayılın ve offset değeri 0.5V olacak şekilde düşünölsün.

$$V = (2000 \times 0.00125) + 0.5 = 2.5 + 0.5 = 3.0 \text{ V} \quad (4.4)$$

İşlem 4.4'teki hesaplama ADC tarafından ölçölen ham verinin gerilim değerine nasıl dönüştüröldüğünü gösterir. Kazanç ADC'nin hassasiyetini belirtir ve offset ise sistemdeki sabit bir hatayı düzeltir.

BQ76952 Entegresi ADC ile sadece gerilim ölçömleri deęil ayrıca sistemden geęen akım ölçömlerini de yapabilmektedir. BQ76952 Akım ölçömleri direnç ve kazanç faktörlerini kullanarak hesaplamaktadır.

Denklem 4.5'te göröldüğü üzere CC Akım Sayısı (Current Count) anlamına gelir, I akımı ifade eder ve R_sense sensör direncini ifade eder. BQ76952 gibi batarya yönetim entegrelerinde akım ölçümü sırasında ADC (Analog-Dijital Dönüştürücü) tarafından dijital sayılara dönüştürölen akım ölçömleri kullanılır. ADC tarafından okunan bu ham sayılar belirli bir kazanç (CC kazancı) ve sensör direnci (R_sense) kullanılarak geręek akım değerlerine dönüştürölür.

$$I = \frac{\text{CC Ham Veri}}{R_{\text{sense}}} \times \text{CC Kazancı} \quad (4.5)$$

Örneğin Current Count Ham veri değeri 1500, Current Count Kazancı değeri 7.47 ve R_sense 1 mΩ olacak şekilde bir hesaplama işlemi denklem 4.6'da gösterilmiştir.

$$I = \frac{1500}{0.001} \times 7.47 = 11215.2 \text{ mA} = 11.2152 \text{ A} \quad (4.6)$$

BQ76952 Entegresi ADC ile ayrıca sıcaklık değerlerini okuma işlemlerini de gerçekleştirmektedir. Bu işlemi de yine gerilim hesabında yaptığı gibi kazanç ve offset gibi değerlere bağlı olarak belirlemektedir. Denklem 4.7 ADC Ham Sayısının gerilime dönüşümü formülasyonunu göstermektedir. Ayrıca sıcaklık hesabında termistör direnci hesaplanması gereklidir bu da denklem 4.8’de gösterilmiştir. Son olarak elde edilen değerlerden sıcaklık değerini elde etme denklemine de denklem 4.9’da yer verilmiştir. NTC (Negative Temperature Coefficient) termistörler, sıcaklık arttıkça dirençleri düşen sıcaklık sensörleridir. Bu direnç sıcaklık ilişkisi doğrusal değildir. Bu nedenle Steinhart-Hart denklemi [153] gibi daha karmaşık bir matematiksel model gerekir. Steinhart-Hart Denklemine denklem 4.9’da yer verilmiştir.

$$V_{\text{termistör}} = \frac{\text{ADC} \times V_{\text{ref}}}{\text{ADC Çözünürlüğü}} \quad (4.7)$$

$$R_{\text{termistör}} = R_s \times \left(\frac{V_{\text{ref}}}{V_{\text{termistör}}} - 1 \right) \quad (4.8)$$

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln \left(\frac{R_{\text{termistör}}}{R_s} \right) \quad (4.9)$$

Örneğin ADC Ham veri değeri 1500, termistör B değeri 3435 (B25/85 değeri), referans gerilim (V_{ref}) 3.3V, ADC çözünürlüğü 12-bit (4096), seri direnç (R_s) 10 k Ω ve termistör direnci 10 k Ω (25°C = 298.15 K) olduğunu bir sistem için hesaplamalara yer verilmiştir.

$$V_{\text{termistör}} = \frac{1500 \times 3.3}{4096} = 1.209 \text{ V} \quad (4.10)$$

$$R_{\text{termistör}} = 10 \times \left(\frac{3.3}{1.209} - 1 \right) = 17290 \text{ } \Omega \quad (4.11)$$

$$\frac{1}{T} = \frac{1}{298.15} + \frac{1}{3435} \ln \left(\frac{17290}{10000} \right) = 0.003515 \quad (4.12)$$

$$T_{\text{Kelvin}} \approx 284.44 \text{ K} \quad \rightarrow \quad \approx 11.29 \text{ } ^\circ\text{C}$$

Burada öncelikle 4.10 işleminde ADC ham sayısının gerilime dönüşümü işlemi yapılmış ardından 4.11 işleminde termistör direncinin hesaplanması işlemi yapılmıştır. Daha sonra bu hesaplamalar sıcaklığın hesaplanması yönteminde olan Steinhart-Hart Denklemi yani 4.12 deki işlemde yerine koyularak sıcaklık değeri elde edilmiştir.

Sonuç olarak BQ76952 Entegresi bir dizi işlem ve hesaplamaların ardından batarya yönetim sistemi için önemli olan bilgileri okumak ve bu verileri göndermekle görevlidir. Ve ayrıca bu entegrenin STM ile haberleşmesi oldukça önemlidir. Batarya yönetim sistemi entegresi ile STM haberleşmesi için ilk adımda I2C haberleşmesinin STM üzerinden aktif edilmesi ve PCB üzerinden doğru pinlerin bağlantısının yapıldığından emin olunarak başlandı.

Ardından I2C Haberleşmesi olup olmadığı STM üzerinden BQ76952 entegresine komut gönderip cevap alınarak çalışır olduğu deneme işlemi ile gerçekleştirildi. Bu sorgu kod yapısının bir bölümüne Şekil 4.10'da yer verilmiştir.

```
if (HAL_I2C_IsDeviceReady(&hi2c1, 0x10, 10, 100) == HAL_OK)
{
    // Eğer cihaz hazırsa:

    // 9 kez GPIO pinlerini değiştir (LED'leri yanıp söndür)
    for (int i = 0; i < 9; i++)
    {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_2); // GPIOA PIN_2'nin durumunu ters çevir
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_1); // GPIOA PIN_1'in durumunu ters çevir
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_0); // GPIOA PIN_0'in durumunu ters çevir
        HAL_Delay(100); // 100 ms bekle
    }
    // Döngü tamamlandıktan sonra tüm pinleri açık duruma getir
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, SET); // GPIOA PIN_2'yi Set et
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, SET); // GPIOA PIN_1'i Set et
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, SET); // GPIOA PIN_0'ı Set et
}
```

Şekil 4.10 : STM ve BQ entegresinin haberleşme kontrolü.

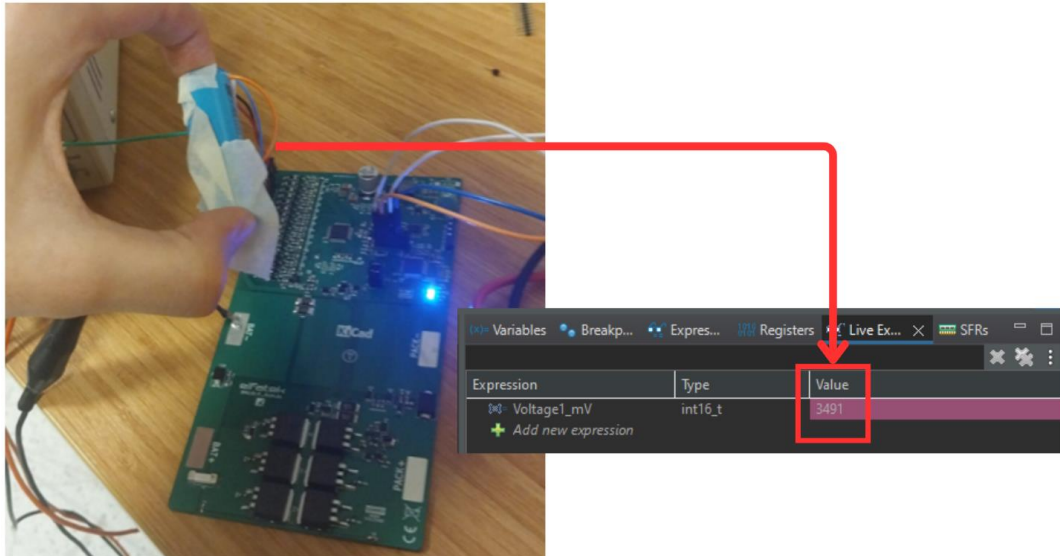
Haberleşmenin başarılı olup olmadığı STM32'nin GPIO pinlerine bağlı LED'lerin yanıp sönmesiyle görsel olarak gösterilir. Örneğin cihazın doğru bir şekilde bağlanıp bağlanmadığını kontrol etmek için HAL_I2C_IsDeviceReady fonksiyonu kullanılır. Haberleşme başarılı olduğunda LED'ler yanıp sönmeye duruma kullanıcıya bildirir ve ardından sürekli açık duruma getirilerek bağlantının aktif olduğunu gösterir. Bu sayede görsel olarak bağlantı kontrolleri gerçekleştirilmiş olmuştur. Bu test uygulanarak ledlerin yanık kalma durumu sonrasında I2C haberleşmesinin başarılı şekilde gerçekleştiği gözlemlenmiştir.

BQ entegresiyle iletişimin sağlanmasının ardından tek bir pil hücresinden okuma işlemi yapmak adına tasarımdaki gerekli konnektörlere bir adet pil bağlantısı gerçekleştirildi. Bu bağlantının ardından STM üzerinden I2C aracılığıyla batarya hücre gerilimini öğrenme komutu gönderildi. Bu kod yapısının kısa bir parçası Şekil 4.11’de gösterilmiştir.

```
main.c BQ76952.c
539 // ***** BQ769x2 Okuma Komutları *****
540
541 uint16_t BQ769x2_ReadVoltage(uint8_t command)
542 // Bu fonksiyon, belirli bir hücrenin voltajını okumak için kullanılır.
543 {
544     // RX_data, global bir değişkendir ve veri okuma işlemlerinde kullanılır.
545     DirectCommands(command, 0x00, R);
546     if (command >= Cell1Voltage && command <= Cell16Voltage) { // İlgili komut adresine göre veri okunmasını sağlar.
547
548         // Eğer komut adresi hücre voltaj aralığındaysa (Hücre 1'den Hücre 16'ya kadar, adresler: 0x14 - 0x32)
549
550         return (RX_data[1] * 256 + RX_data[0]); // Hücre voltajını mV (miliVolt) olarak döndürür.
551     } else {
552         return 10 * (RX_data[1] * 256 + RX_data[0]); // Voltaja 0.01V biriminden mV'ye çevirerek döndürür.
553     }
554 }
555
```

Şekil 4.11 : Tek bir hücre okuma için oluşturulan kodların bir kısmı.

Daha sonra STMCUBEIDE programı üzerinde debug yapılarak live expression sekmesi üzerinden bir adet pil okuma işlemi gerçekleştirilmiş oldu. Bu bağlantı ve okuma işlemi görseline Şekil 4.12’de yer verilmiştir.



Şekil 4.12 : Tek pil hücresi bağlantısı ve veri okuma işlemi.

Bu tek pil hücresinin gerilim okuma işlemi tamamlanmasının sonrasında daha fazla adette hücre okuması yapılması planlaması yapıldı. Burada her pil hücresi ayrı ayrı takmak uygun olmayacağından bataryayı simüle edecek olan bir sistem oluşturuldu. Burada 16 Adet lityum iyon pil seri bağlanarak bir batarya elde edilmesi planlandı.

Batarya oluşturulurken ilk etapta her hücreden batarya dengeleme sistemine bağlanacak olan kablolar her hücre düğüm notasına bağlantısı gerçekleştirildi. Bu gerçekleştirilen işlemlerin bir aşamasına Şekil 4.13'te yer verilmiştir.



Şekil 4.13 : Çalışmalar yapılabilmesi için pil hücrelerinden batarya yapımı.

Buradaki yapılan işlem aslında bir bataryaya doğrudan bağlantı işlemi öncesi risk durumu daha düşük olan bir sistem kurmaktır. Simülasyon bataryası özellikleri batarya yönetim sistemi entegresinin yeteneklerini tamamen kullanmak adına 16 seri hücre olacak şekilde yapılmıştır. Burada oluşturulan batarya sonrasında testler ayrıca kurgusu oluşturulan AKİBA bataryasında da uygulanmıştır. Bu AKİBA aracının bataryası 7 seri 16 paralel olacak şekilde tasarlanmıştır.

Bu iki batarya içerisinde de Molicel tarafından üretilen INR18650-M35A model lityum-iyon hücrelerden oluşmaktadır. Bu hücrelerin nominal kapasitesi 3500 mAh (3.5 Ah), nominal gerilimi 3.6 V, enerji kapasitesi 12.5 Wh, maksimum sürekli deşarj akımı 10 A, hücre ağırlığı 48 g ve enerji yoğunluğu 250 Wh/kg olarak ürün belgelerinde verilmektedir [154].

Buradaki bilgiler kullanılarak bir pil hücresi ve batarya yönetim sistemi için kullanılacak olan bataryaların enerji kapasiteleri hesaplandı. Dokümandaki verilen hücre özellikleri ve enerji kapasitesi hesaplama formülü kullanılarak bu hesaplamalar gerçekleştirildi.

$$E_{\text{cell}} = V_{\text{nom}} \times C_{\text{nom}} \quad (4.13)$$
$$E_{\text{cell}} = 3.6 \text{ V} \times 3.5 \text{ Ah} = 12.6 \text{ Wh}$$

Denklem 4.13'te E_{cell} hücrenin enerji kapasitesini, V_{nom} hücrenin nominal gerilimini ve C_{nom} nominal hücre kapasitesini ifade etmektedir. Buradaki hesaplama sonucunda her bir hücrenin enerji kapasitesi 12.6 Wh olarak hesaplandı. Bu hesaplama işlemleri

batarya yönetim sistemi testlerinde kullanılacak olan bataryalar için de gerçekleştirildi. Burada öncelikle 16 seri 1 paralel olacak olan büyük bir bataryayı simüle eden batarya için hesaplamalar yapıldı.

$$V_{\text{total}} = N_{\text{seri}} \times V_{\text{nom}} \quad (4.14)$$

$$V_{\text{total}} = 16 \times 3.6 \text{ V} = 57.6 \text{ V}$$

$$E_{\text{total}} = V_{\text{total}} \times C_{\text{total}} \quad (4.15)$$

$$E_{\text{total}} = 57.6 \text{ V} \times 3.5 \text{ Ah} = 201.6 \text{ Wh}$$

Denklem 4.14'te V_{nom} nominal hücre gerilimini, N_{seri} seri bağlanan hücre sayısını ve V_{total} toplam gerilimi ifade etmektedir. Burada toplam gerilim 57.6 V olarak hesaplandı. Denklem 4.15'te toplam kapasite hesaplanırken E_{total} bataryanın toplam enerji kapasitesini ve C_{total} toplam hücre kapasitesini ifade etmektedir. Bu hesaplama işleminde paralel hücre olmadığı için yani sadece seri bağlantı olduğundan kapasite değişmez. Yani sadece seri bağlı hücrelerden oluşan bir batarya olduğu için toplam kapasite bir hücre kapasite değerine eşit olacak şekilde hesaplandı. Burada kullanılan pil özellikleriyle oluşturulan 16 seri batarya paketini toplam enerji kapasitesi 201.6 Wh olarak hesaplandı. Bu hesaplama ardından AKİBA otonom aracında standartta kullanılan batarya olan 7 seri 16 paralel olan batarya için de hesaplama işlemleri yapıldı.

$$V_{\text{total}} = N_{\text{seri}} \times V_{\text{nom}} \quad (4.16)$$

$$V_{\text{total}} = 7 \times 3.6 \text{ V} = 25.2 \text{ V}$$

Denklem 4.16'da V_{total} toplam gerilimi, N_{seri} seri bağlanan hücre sayısını ve V_{nom} nominal hücre gerilimini ifade etmektedir. Burada toplam gerilim 25.2 V olarak hesaplandı.

$$C_{\text{total}} = N_{\text{paralel}} \times C_{\text{nom}} \quad (4.17)$$

$$C_{\text{total}} = 16 \times 3.5 \text{ Ah} = 56 \text{ Ah}$$

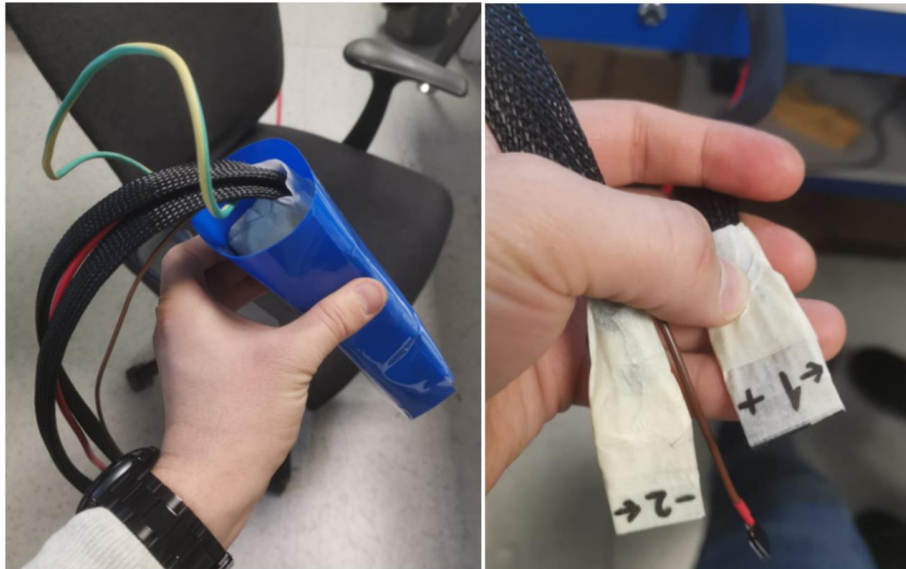
Denklem 4.17'de toplam kapasite hesaplanırken C_{nom} nominal hücre kapasitesini, N_{paralel} paralel bağlı olan hücre sayısını ve C_{total} toplam hücre kapasitesini ifade

etmektedir. Burada 16 paralel yapısından dolayı toplam kapasite 56 Ah olarak hesaplandı.

$$E_{\text{total}} = V_{\text{total}} \times C_{\text{total}}$$
$$E_{\text{total}} = 25.2 \text{ V} \times 56 \text{ Ah} \approx 1.42 \text{ kWh} \quad (4.18)$$

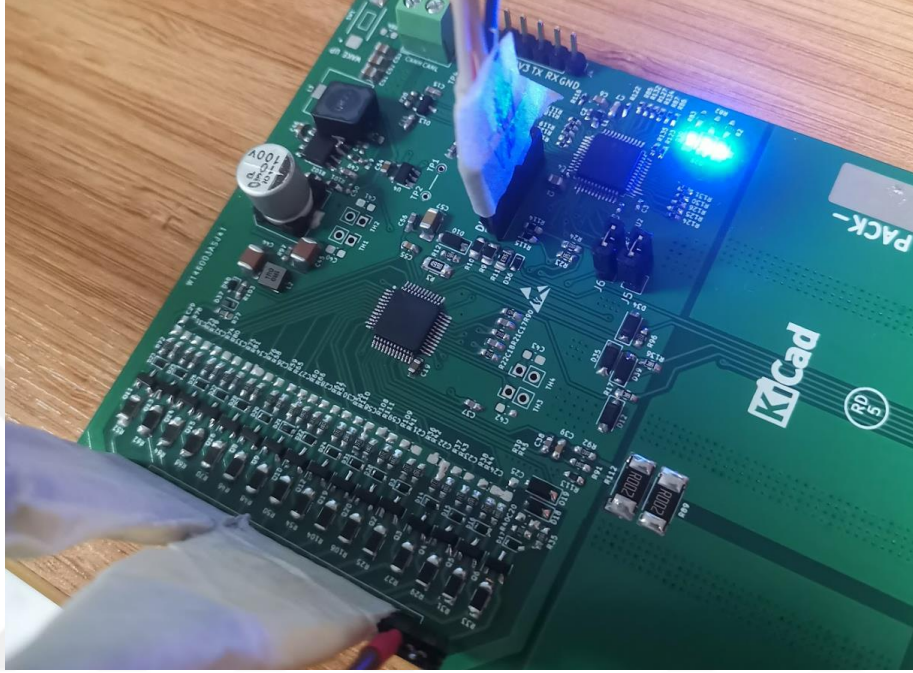
Denklem 4.18’de toplam kapasite hesaplanırken E_{total} toplam kapasiteyi ve C_{total} toplam kapasiteyi ifade etmektedir. Bu hesaplama işleminde paralel hücre olmadığı için yani sadece seri bağlantı olduğundan kapasite değişmez. Yani sadece seri bağlı hücrelerden oluşan bir batarya olduğu için toplam kapasite bir hücre kapasite değerine eşit olacak şekilde hesaplandı. Burada kullanılan pil özellikleriyle oluşturulan 16 seri batarya paketinin toplam enerji kapasitesi 201.6 Wh olarak hesaplandı.

Bu hesaplamalar ardından batarya yönetim sistemine bağlantılarında olası kısa devrelerin olmasının önüne geçebilmek için her hücreden çıkarılan kablolar bir dışı pin header bağlantısı yapılarak yalıtkan malzemelerle korunmaları sağlandı. Bu koruma konnektör bağlantılarının gerçekleştirilmesine Şekil 4.14’te yer verilmiştir. Ayrıca hücre bağlantılarının yanlış yapılması ihtimaline karşı konnektörler numaralandırılarak olası hatalar önlenmesi hedeflenmiştir. Ardından tüm batarya yalıtkan plastik ile kapatılarak herhangi dış etkenlerden ötürü kısa devre oluşmasının önüne geçilmesi planlanmıştır. Ayrıca ana güç çıkışı için bataryanın başı ve sonundan kalın bir kablo ile çıkış alınarak ilerleyen süreçlerde bataryanın güç hattı kontrol edilmek üzere planlama yapılmıştır.



Şekil 4.14 : Konnektör ve yalıtkan kağıt ile kaplama.

Batarya hücrelerinin konnektör çıkışları ayarlandıktan sonra herhangi bir sıkıntı oluşmamasına adına tüm hücreler multimetre ile doğruluğu kontrol edildi. Bu kontrollerin ardından batarya hücreleri kontrollü bir şekilde batarya yönetim sistemine bağlantısı gerçekleştirildi. Bağlantının ilk yapıldığı görsele Şekil 4.15'te yer verilmiştir.



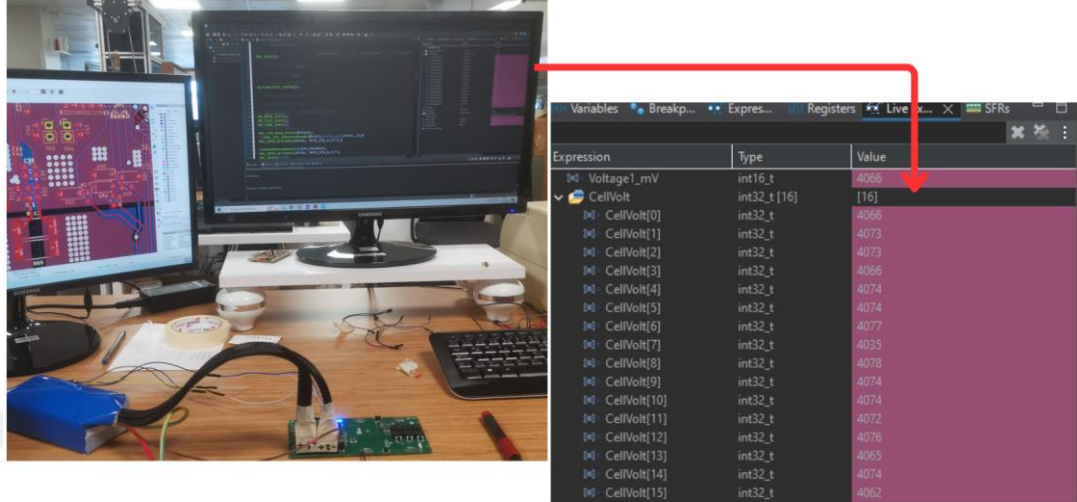
Şekil 4.15 : Oluşturulan bataryanın BMS'e ilk bağlantısı.

Bu bağlantı ile simülasyon için hazırlanan 16S bataryanın her bir hücresinin verileri gerekli parametrelerin BQ76952 entegresine sorgu atıldı. Bu sorgu yazılımlarının bir kısmına Şekil 4.16'da yer verilmiştir.

```
539 // ***** BQ769x2 Okuma Komutları *****
540
541 uint16_t BQ769x2_ReadVoltage(uint8_t command)
542 // Bu fonksiyon, belirli bir hücrenin voltajını okumak için kullanılır.
543 {
544     // RX_data, global bir değişkendir ve veri okuma işlemlerinde kullanılır.
545     DirectCommands(command, 0x00, R);
546     if (command >= Cell1Voltage && command <= Cell16Voltage) { // İlgili komut adresine göre veri okunmasını sağlar.
547
548         // Eğer komut adresi hücre voltaj aralığındaysa (Hücre 1'den Hücre 16'ya kadar, adresler: 0x14 - 0x32)
549
550         return (RX_data[1] * 256 + RX_data[0]); // Hücre voltajını mV (miliVolt) olarak döndürür.
551     } else {
552         return 10 * (RX_data[1] * 256 + RX_data[0]); // Voltajı 0.01V biriminden mV'ye çevirerek döndürür.
553     }
554 }
555
556 void BQ769x2_ReadAllVoltages()
557 // Tüm hücre voltajlarını oku
558 {
559     total_voltage = 0;
560     int cellvoltageholder = Cell1Voltage; // İlk hücre voltajı adresini tutar (örneğin, 0x14).
561     for (int x = 0; x < 16; x++) { // Tüm hücrelerin voltajlarını okumak için döngü başlatılır.
562
563         // Mevcut hücre voltajını okur ve CellVoltage dizisine kaydeder.
564         CellVoltage[x] = BQ769x2_ReadVoltage(cellvoltageholder); // Mevcut hücre voltajını okur ve CellVoltage dizisine kaydeder
565
566         total_voltage += BQ769x2_ReadVoltage(cellvoltageholder); // Hücre voltajını toplam voltaj değişkenine ekler.
567
568         cellvoltageholder = cellvoltageholder + 2; // Bir sonraki hücre voltaj adresine geçmek için adresi 2 artırır.
569     }
570     total_cell_voltage = total_voltage; // Toplam hücre voltajı değeri kaydedilir.
571 }
572
```

Şekil 4.16 : Tüm hücre gerilimlerini okuma sorgu işlemleri.

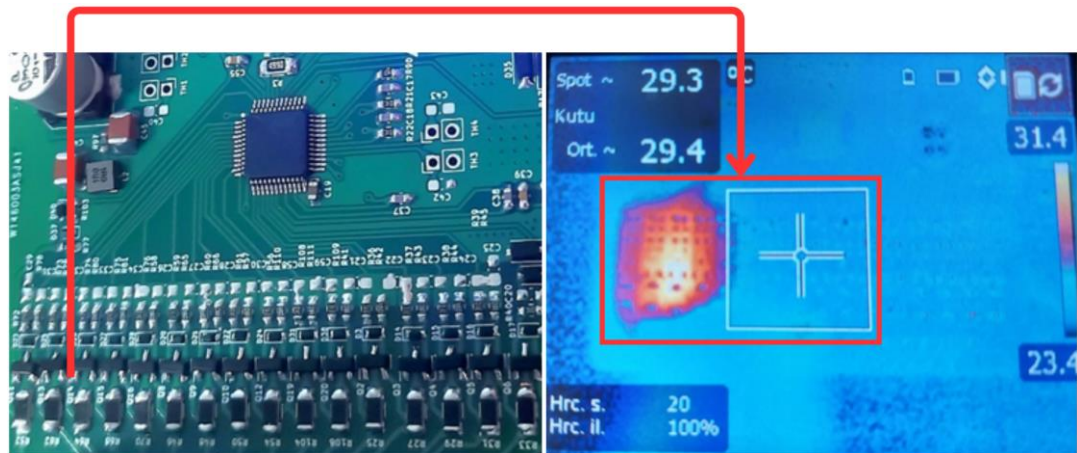
Sorgu işlemlerini yapıldığı yazılımın yüklenmesinin ardından tüm bu batarya verileri STMCUBEIDE platformunun live expression kısmından görüntülenebildi. Bu görüntülemelere Şekil 4.17’de yer verilmiştir.



Şekil 4.17 : Tüm hücre gerilimlerini batarya yönetim sistemi tarafından okunması.

Tüm bu yapılar aslında batarya yönetim sisteminin başarılı bir şekilde çalıştığının ve bundan sonraki aşamaların daha hızlı bir şekilde ilerlemesine önayak oluşturmuş ve bu sayede bütün sistemin temelleri atılmış olmuştur.

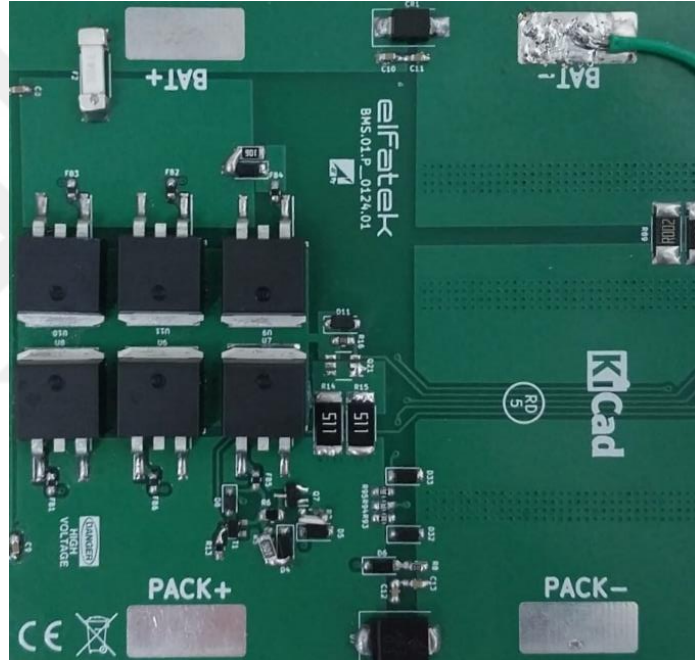
Bundan sonraki aşama olarak batarya yönetim sisteminin temel mantık çalışmalarından olan dengeleme işlemi test edilmek üzere yazılımda güncellemeler yapıldı. Bu işlem için termal kamera kullanılarak ve ilk olarak sadece bir hücrenin dengeleme işlemini gerçekleştirip gerçekleştirmediği üzerinden testler yapıldı. Dengeleme işlemini gösterildiği yapıya Şekil 4.18’de yer verilmiştir.



Şekil 4.18 : Tek bir hücrenin dengeleme işlemi kontrolü.

Buradaki dengeleme işleminin çalışması aslında ilerleyen zamanlarda yazılıma eklenen parametreler sayesinde en yüksek mV değerine sahip olan bataryanın enerjisinin dengeleme dirençleri üzerinden ısıya dönüştürülerek harcanması ve sürekli tüm batarya hücrelerinin gerilim seviyelerinin eşit veya çok yakın tutulabilmesinin çalışacağının göstergesi olmuş oldu.

Bu aşamanın ardından batarya yönetim sisteminin Mosfetlerle ilgili kısımları test edilmeye başlandı. Bu kısımda fetler aslında batarya yönetim sisteminden yük kısmına veya batarya şarj olurken batarya yönüne doğru akacak olan enerjinin kontrol edilmesi için kullanılmaktadır. Şekil 4.19’da bu mosfetlerin dizgisinin yapılı halinin görseline yer verilmiştir.



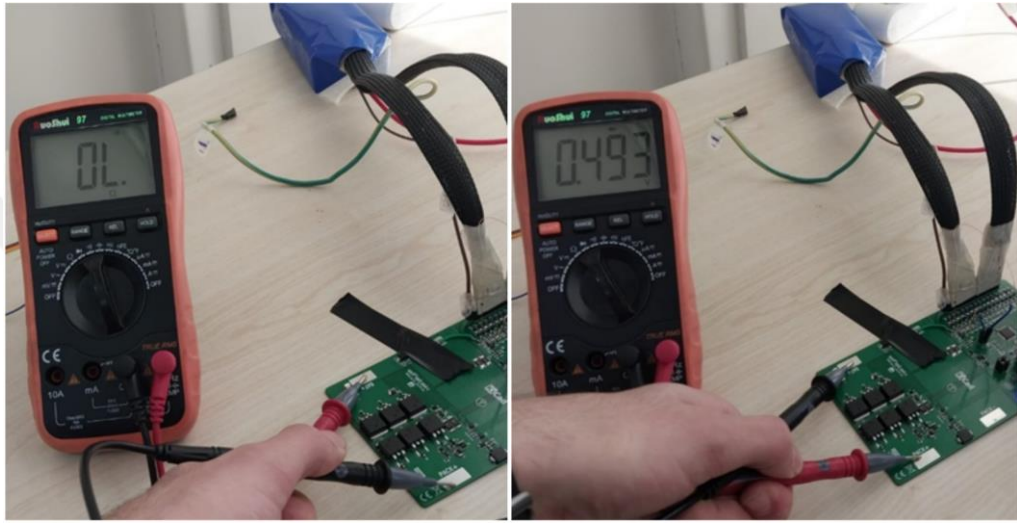
Şekil 4.19 : Mosfetlerin PCB’ye eklenmiş hali.

Bu fetler herhangi bir sıkıntı olmadığı sürece enerji akışına izin veren fakat sistemde herhangi bir sorun olduğu bilgisi batarya yönetim sistemine bildirildiği anda enerji akışını durdurarak sistemin çok büyük sorunlar açmasına ve bu sebepten oluşabilecek can ve mal kayıplarının önüne geçmesini sağlar. Bu sebeple bu kısım aslında diğer bölümlerinde olduğu gibi oldukça önemli bir kısımdır.

Bu fetler sayesinde eğer kullanıcı isterse uzaktan batarya yönetim sistemine erişerek herhangi ekstra bir aşamaya uğramadan tüm sistemin enerjisini keserek oluşabilecek sorunların çok önceden önüne geçebilir. Burada aslında birçok güvenlik önlemlerini batarya yönetim sistemi kendi içerisinde halledebilmektedir.

Bu durum aslında otonom aracında öngöremediği herhangi acil bir durumda bu cihazların kontrollerini yapan kişiler tarafından ekstra bir güvenlik önlemi olarak cihazın maksimum güvenlik seviyelerine ulaşmasına olanak sağlamış olacaktır

Bu fetler test edilirken multimetre yardımıyla kontrol edilmiştir. Önce kapalıyken bağlantı testinde herhangi bir şey yokken fetler yazılımsal parametreler ayarlanmasının ardından bağlantı testinde batarya ve yük tarafı birbirine bağlı olduğu görülmüştür. Bu test yapısının görseline Şekil 4.20’de yer verilmiştir

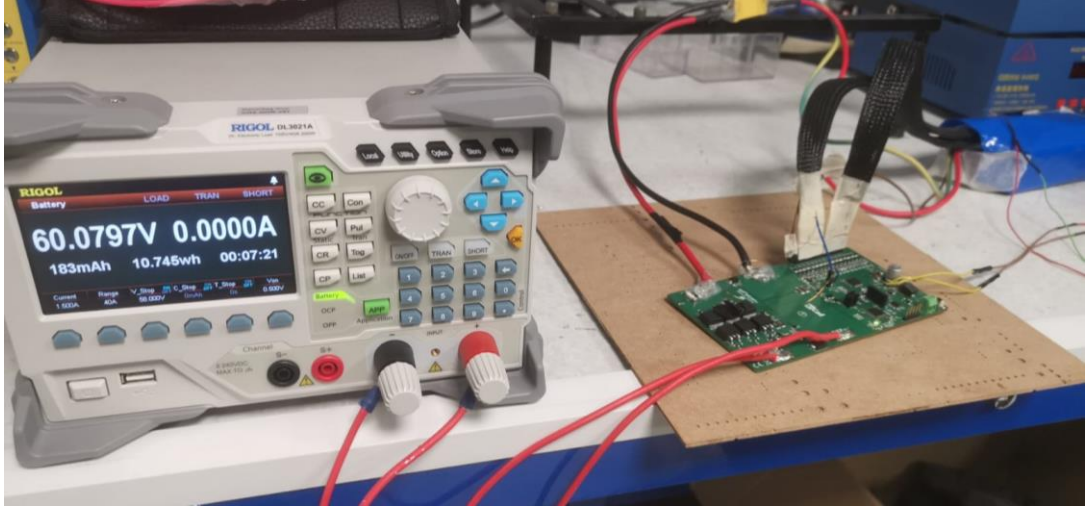


Şekil 4.20 : Mosfetlerin yazılımsal kontrollerini gerçekleştirilmesi.

Bu testler sayesinde ilerleyen aşamalardaki koruma işlemlerini başarılı bir şekilde yapacağının göstergesi oluşmuş oldu. Bu işlem hem şarj modunda hem de deşarj modunda sorunsuz şekilde gerçekleştirildi ve bu durumlarda da hata olmadığı gözlemlendi.

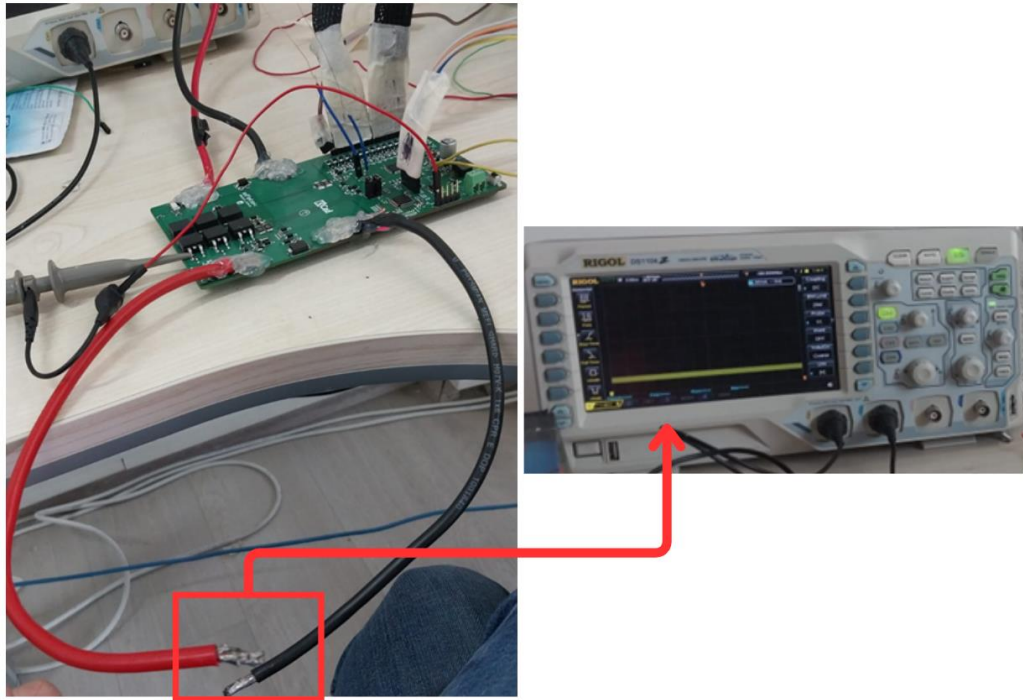
Bu fetlerin açılıp kapanmasının sorunsuz şekilde çalıştığı görüldükten sonra kısa devre korumasının düzgün çalışıp çalışmadığının kontrolü için kısa devre parametreleri yazılım ile aktif edildi. Bu kısa devre sistem koruması işlemleri batarya test cihazı üzerinde de test edildi. Bataryaya yük olarak bağlı olan bu cihazda kontrollü şekilde kısa devre yapılarak enerji akışını kestiği detaylı bir şekilde gözlemlendi. Bu test işleminin görseline Şekil 4.21’de yer verilmiştir.

Daha sonra bu testin gerçek bir kaza ihtimali testi ile yük tarafındaki artı ve eksi uçları çok ani bir şekilde değiştirilip bırakıldı. Yani kısa devreyi simüle edebilmek için çok hızlı bir şekilde batarya paketinin çıkışındaki güç kabloları BMS sonrası kontrollü bir şekilde birbirine değiştirildi.



Şekil 4.21 : Kısa devre testi ve kesim işleminin test cihazıyla incelenmesi.

Bu durumda batarya yönetim sistemi hemen sistemi korumaya alacak şekilde tüm fetleri kapadığı görüldü. Yani herhangi bir kısa devre tehlikesinde batarya yönetim sistemi bunu algılayarak olası tehlikelerin önüne geçebilmiş oldu. Bu test işlemi ayrıca fetlerin gate ucuna osiloskop bağlanarak sıfıra çekildiği görülmesiyle de pekiştirilmiş oldu. Bu test işlemlerinin görseline Şekil 4.22’de yer verilmiştir.



Şekil 4.22 : Kısa devre testi ve kesim işleminin osiloskopla incelenmesi.

Batarya yönetim sistemindeki güvenlik korumalarından biri de sıcaklıktır. Herhangi bir sıcaklık artışı durumunda enerji akışını durdurma görevi batarya yönetim sistemine aittir.

Bu sıcaklık kesim işlemlerinin çalışabilmesi için yazılımda gerekli parametreler güncellendi ve testleri başlatıldı. Öncelikle daha önceden bağlantısı yapılan termistörden sıcaklık okumaları yapıldı ve hata yapılarının oluşması için gerekli parametre ayarları hazırlandı. Ayrıca BQ entegresinin tanınması için hangi termistörün sıcaklığı baz alınarak işleme alınacağı yapısı bu entegreye tanımlama parametreleri eklendi. Bu parametre ayarlarının yazılım kısmının kısa bir yapısına Şekil 4.23'te yer verilmiştir.

```
void BQ769x2_Init() {
    /*BQ76952 nin config ayarlarının gerçekleştirilebilmesi için özel bir komut gönderilir.
    * Bu config komutunu alan BQ76952 artık istenilen parametrelerce programlanabilir.
    * (AT komutlarına benzer bir yapı.)*/

    CommandSubcommands(SET_CFGUPDATE);

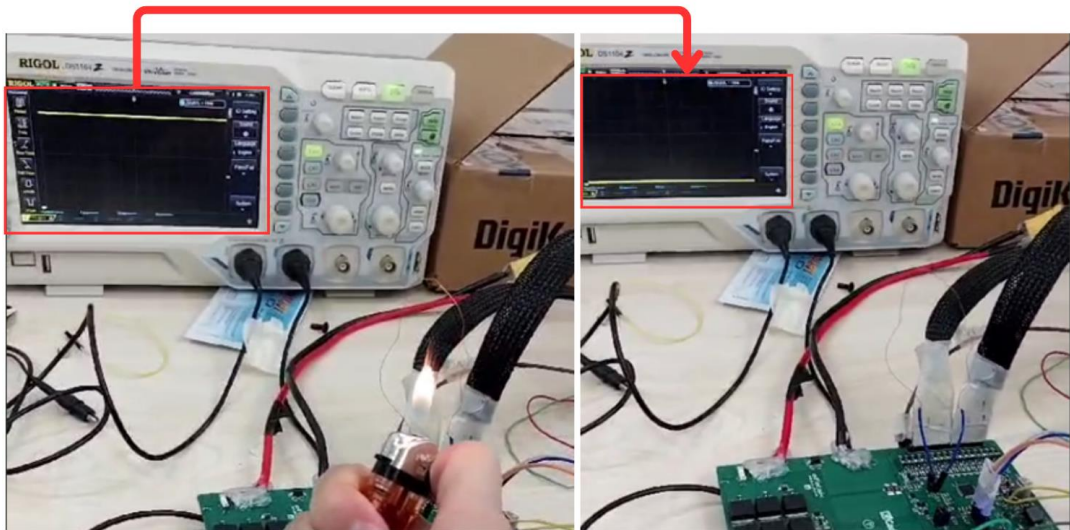
    /* BQ769x2 nin içerisinde bulunan dahili sıcaklık sensörünün config ayarı ile
    BQ769x2 nin iç sıcaklık değerinin belirli bir eşiği geçmesi durumunda sistemin korumaya alması sağlandı.
    */
    BQ769x2_SetRegister(DACConfiguration, 0xD, 1);

    /*Hücrelerin hangi sıcaklıktan sonra korumaya girmesi gerektiği ayarlanmıştır.*/
    BQ769x2_SetRegister(MaxCellTemp, 0x3C, 1);

    /*BQ769x2 nin hani pinlerinin termistör olarak ayarlanacağı ve hangi termistörün
    * hangi sıcaklık değerini baz alacağı ayarlanmıştır.*/
    BQ769x2_SetRegister(TS1Config, 0x0B, 1);
    BQ769x2_SetRegister(TS3Config, 0x07, 1);
    BQ769x2_SetRegister(HDQPinConfig, 0x0F, 1);
    BQ769x2_SetRegister(CFETOFFPinConfig, 0x0F, 1);
}
```

Şekil 4.23 : Parametre ayar yazılım yapısının kısa bir kısmı.

Termistör sıcaklığı belirlenen limit sıcaklık değerinin üstüne çıktığı anda mosfetlerdeki gate pini sıfıra çekilerek mosfeterin kapandığı ve enerji akışının durduğu gözlemlendi. Dolayısıyla sistemdeki olası çok büyük problemlerin önüne bu şekilde geçilmiş oldu. İlerleyen aşamalarda bu otonom araç için hayati bir öneme sahip olacaktır. Bu test işleminin görseline Şekil 4.24'te yer verilmiştir.



Şekil 4.24 : Yüksek sıcaklık kesim işleminin osiloskopla incelenmesi.

Tüm bu işlemlerin ardından simülasyon için kullanılan batarya değil asıl otonom araca bağlanacak ve içerisinde bulunacak olan batarya ile test işlemlerine devam edildi. Öncelikle bataryanın hücrelerinden batarya yönetim sistemine test için bağlanacak olan kablolar isimlendirildi ve batarya yönetim sistemine sorunsuz şekilde bağlantısının olabilmesi için dişi header'a bağlantısı aynı simülasyon bataryasında olduğu gibi lehimleme işlemleri gerçekleştirildi. Bu işlemlerin görseline Şekil 4.25'te yer verilmiştir.



Şekil 4.25 : AKİBA bataryasını test işlemleri için ayarlama.

Bu gerçekte kullanılacak olan bataryanın olası hata durumlarından geri dönülemez sıkıntılar oluşturabileceği göz önünde bulundurularak batarya koruma testleri için düzenek kuruldu. Bu düzenekte otonom araç bataryası yanmaz toz içerisine gömülerek testler yapılırken olası bir yangının önüne geçilmesi hedeflenmiştir.

Batarya yönetim sistemi RF bir kumanda ve kontrolcüsü aracılığıyla kısa devre testine tabi tutulmak üzere ilk teste başlanmıştır. RF kumanda kullanılmasının sebebi olası tehlikeli bir durum olmasına karşın uzaktan kontrolle yük kısmının kısa devre edilebilmesidir. Bir yandan da bilgisayar ile bu sistem takip edilerek kumanda ile kısa devre komutu verildiğinde enerji akışını kesip kesmediği kontrol edildi.

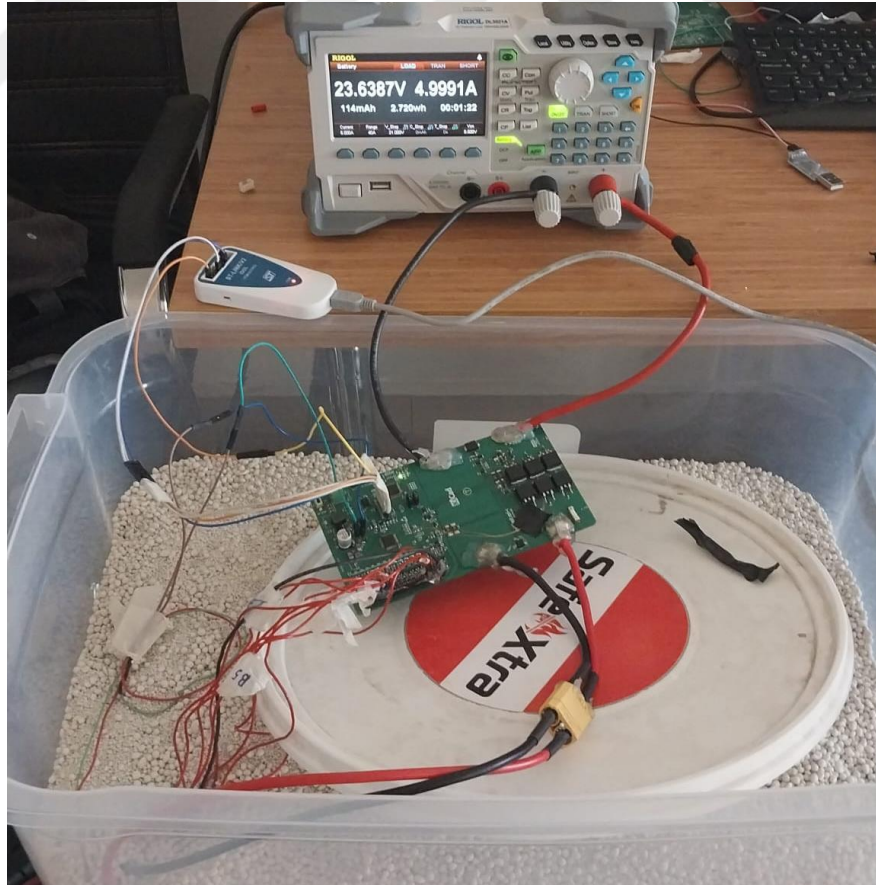
Bu düzeneğin hazırlanmasının temel sebebi yani uzaktan kontroller ve yanmaz kum içerisinde bu test işlemlerinin gerçekleştirilmesinin temel sebebi olası yanlış yapılan işlemlerin veya BMS sisteminde oluşacak olan hatalarda ortamı ve canlıları riske atmamak üzere bu şekilde bir planlama yapılmıştır.

Bu düzenek ile test yapıldığında kumanda kontrolcüsü ile yük tarafı kabloları birbirine bağlantısı yapılmış ve anında enerji akışının kesildiği gözlemlenmiştir. Bu test işleminin görseline Şekil 4.26'da yer verilmiştir.



Şekil 4.26 : Batarya kısa devre test düzeneği.

Sorunsuz şekilde kendini korumaya alan bu sistemin uzun süreli deşarj işlemleri için batarya test cihazı bağlantısı ile test düzeneği kurulmuştur. Bu test düzeneğinin görseline Şekil 4.27’de yer verilmiştir.

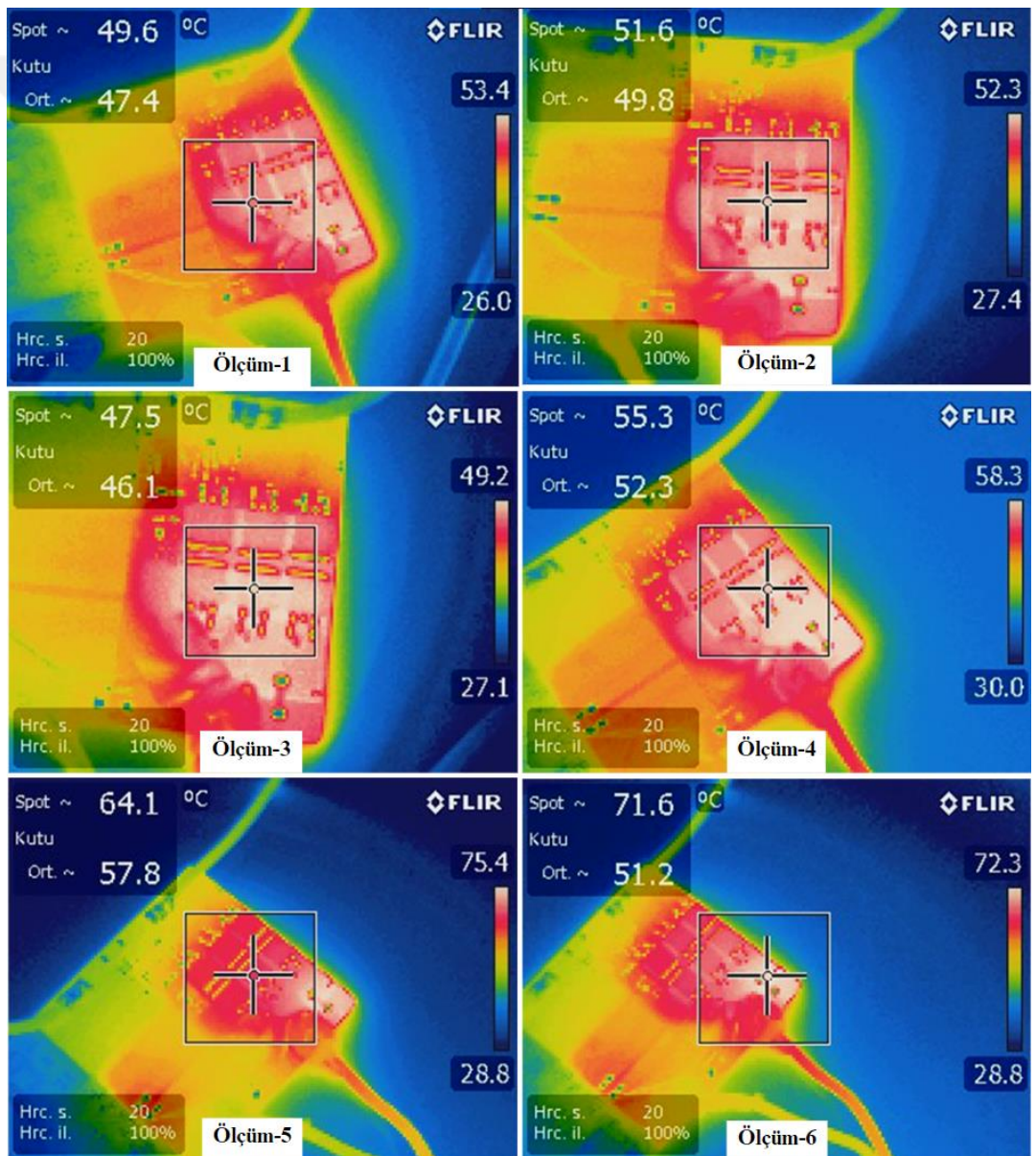


Şekil 4.27 : Uzun süreli deşarj işlemleri test düzeneği.

Bu test düzeneği ile batarya yönetim sistemi farklı akım değerleri ve farklı sürelerde teste tabi tutularak ne kadar ısındığı ve sıkıntı oluşturabilecek herhangi bir durum olup olmadığı gözlemlenerek bu gözlemler not edilmiştir.

Bu gözlemlerdeki sonuçlar için termal kamera kullanılmış ve her durum için gerçekleştirilmiş işlemlerin termal kamera görüntülerine toplu şekilde Şekil 4.28’de yer verilmiştir.

Not edilen bu değerler düzenli bir yapı için çizelge 4.1’de sonuç durumları açıklanmıştır. Sonuç olarak batarya yönetim sisteminin sıcaklıklarının sorun teşkil etmeyecek yani güvenli sınırlar içerisinde çalıştığı gözlemlenmiştir.

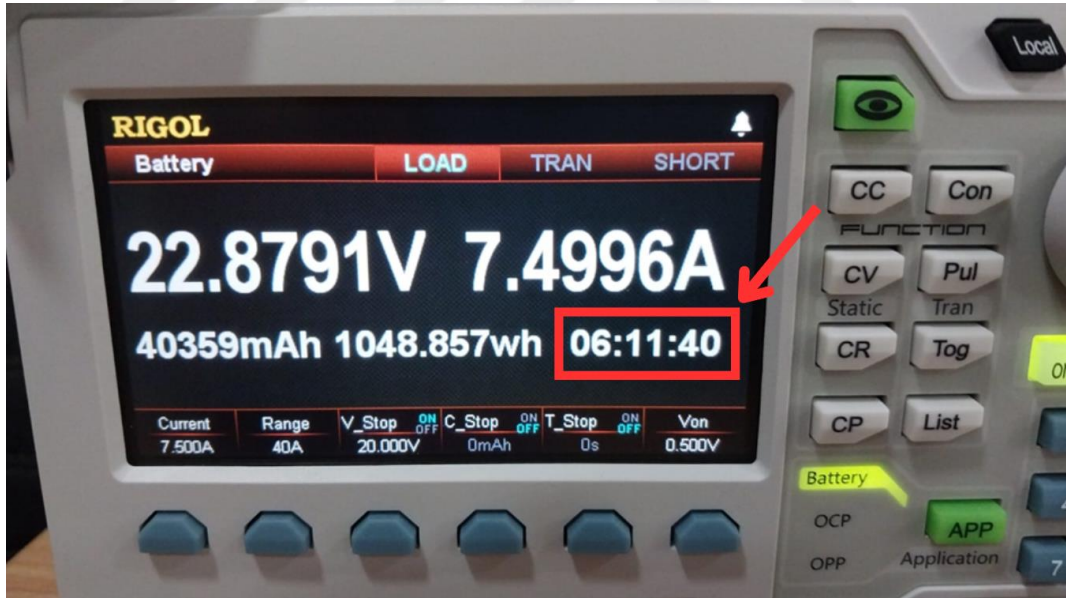


Şekil 4.28 : Deşarj testlerinde çekilen termal kamera görüntüleri.

Çizelge 4.1 : Deşarj akımı zamana bağlı sıcaklık değişimi.

Ölçüm No	Zaman	Akım	Maksimum Sıcaklık
1	0 dk - 15 dk	20A	53.4 °C
2	15 dk - 30 dk	22A	52.3 °C
3	30 dk - 60 dk	18A-22A	49.2 °C
4	60 dk - 75 dk	22A	58.3 °C
5	75 dk - 105 dk	25A	75.4 °C
6	105 dk - 150 dk	25A	72.3 °C

Sıcaklık değer testlerinden de başarıyla geçen batarya yönetim sistemi son olarak uzun süreli deşarj işlemine tabi tutuldu. Bu bağlamda 7.5A yük üzerinde yaklaşık 6 saat boyunca deşarj işlemi gerçekleştirildi. Batarya yönetim sistemi bu testten de başarılı bir şekilde geçmiştir. Test görseline Şekil 4.29’da yer verilmiştir. Ayrıca Otonom aracın bataryası bağlıyken balans işlemleri testleri de yapılarak sıcaklık durumu gözlemlenmiştir. Bu test sonucunda balans dirençlerinin uzun testlerin ardından 81°C olduğu görülmüştür ve istenilen aralıkta kalarak bu testten de geçmiştir.



Şekil 4.29 : Uzun süreli deşarj işlemi görseli.

Batarya yönetim sistemi işlemlerini yapan BQ76952 entegresi kendi bünyesinde birçok koruma yapısına sahiptir. Bu koruma yapılarından bazıları tezin bu bölümünde açıklanacaktır. İlerleyen zamanlarda bu koruma yapılarının parametreleri IoT işlemleri ile kontrol edilebilecek şekilde yenilikçi bir yaklaşım sunacaktır.

Buradan sonraki koruma işlemleri başlıklarında kısa kısa bilgilere yer verilmiştir. Tüm bu korumaların detay bilgilerine BQ76952 entegresini teknik referans kılavuzundan ulaşılabilir [155]. Birkaç tane korumanın teknik dokümandaki ayarlamalarının ekran görüntülerine yer verilmiştir, tüm anlatılan korumalar için resim eklenmemiştir.

4.1.1 Hücre alt gerilim koruması (CUV)

Hücre alt gerilim koruması (Cell Undervoltage Protection-CUV) hücre voltajı bir belirli eşiğin altına düştüğünde koruma sağlayan bir mekanizmadır. BQ76952 cihazında CUV eşik değeri 1.012 V ile 4.048 V arasında ayarlanabilir ve her adımda 50.6 mV'luk bir artış söz konusudur. Bu bilginin görsel ekran görüntüsüne Şekil 4.30'da yer verilmiştir.

13.6.1.1 Protections:CUV:Threshold

Class	Subclass	Name	Type	Min	Max	Default	Unit
Protections	CUV	Threshold	U1	20	80	50	50.6mV

Şekil 4.30 : CUV koruması teknik belge ekran görüntüsü

CUV cihazın hücre voltajını hassas bir şekilde izleyip belirtilen eşiğin altına düştüğünde koruma mekanizmasını devreye sokmasını sağlar. Minimum Eşik Değeri 1.012 V olan bu değer cihazın en düşük voltaj seviyesini belirler. Hücre voltajı 1.012 V'nin altına düştüğünde CUV koruması devreye girer ve hata mesajı tetiklenir. Maksimum Eşik Değeri 4.048 V olan bu değer cihazın en yüksek voltaj sınırıdır. 4.048 V'nin altına inildiğinde CUV koruması çalışır. Bu voltaj seviyesine bağlı olarak, batarya sağlığı için düşük voltajdan kaynaklanabilecek hasarları önlemek mümkündür. Örneğin eğer eşik değeri 1.062 V olarak ayarlandıysa hücre voltajı bu değer altına düştüğünde CUV koruması tetiklenir. Bu hassas ayarlamalar sayesinde bataryalar düşük voltaj seviyelerinde güvenle korunur ve zarar görmeleri engellenir [155].

4.1.2 Hücre aşırı gerilim koruması (COV)

Hücre aşırı gerilim koruması (Cell Overvoltage Protection-COV) hücre voltajı belirli bir eşiği aştığında aşırı voltajdan kaynaklanan zararları önlemek amacıyla çalışan bir koruma mekanizmasıdır. BQ76952'de COV eşik değeri, 1.012 V ile 5.566 V arasında ayarlanabilir ve yine her adımda 50.6 mV'luk bir artış vardır.

Örneğin eğer eşik değeri 4.098 V olarak ayarlandıysa hücre voltajı 4.098 V'yi geçtiğinde aşırı voltaj uyarısı devreye girer [155].

4.1.3 Şarj aşırı akım koruması (OCC)

Şarj aşırı akım koruması (Overcurrent in Charge Protection- OCC) eşik değeri 4 mV ile 124 mV arasında 2 mV'luk adımlarla ayarlanabilir. Bu cihazın şarj akımını aşırı yükselmeye karşı hassas bir şekilde tespit etmesine olanak tanır.

Bu değerlerin şarj devresindeki akıma dönüştürülmesi şönt direncinin değerine bağlıdır. Örneğin şönt direnci 1 mΩ olarak seçilirse 124 mV'luk bir eşik değeri yaklaşık olarak 124 A şarj akımına denk gelir. Bu nedenle cihazın kullanılacağı sistemin gereksinimlerine göre şönt direncinin dikkatle seçilmesi gerekmektedir.

OCC koruması aşırı şarj akımlarını engelleyerek bataryaların ve diğer devre elemanlarının hasar görmesini önler. Bu özellik sistemin güvenliğini artırmak için kritik öneme sahiptir [155].

4.1.4 Deşarj aşırı akım koruması (OCD)

Deşarj aşırı akım koruması (Overcurrent in Discharge-OCD) batarya çıkışındaki aşırı akımı izleyen ve bu akım belirli bir eşiği aştığında koruma sağlayan bir mekanizmadır. BQ76952 cihazında OCD eşik değeri 4 mV ile 200 mV arasında ayarlanabilir ve her adımda 2 mV'luk bir artış yapılabilir. Bu çıkış akımını hassas bir şekilde izlemeyi ve aşırı akım durumu tespit edildiğinde cihazın korunmasını sağlamayı mümkün kılar. Buradaki korumanın çalışma mantığı OCC korumasında olduğu gibidir [155].

4.1.5 Deşarjda kısa devre koruması (SCD)

Deşarjda kısa devre koruması (Short Circuit in Discharge Protection-SCD) eşik değeri kısa devre olaylarını hassas bir şekilde izlemeye olanak tanır. Bu eşik çıkıştaki SRN-SRP pinleri arasındaki voltaj farkını belirler ve cihazın kısa devreyi tespit etmesine yardımcı olur. Örneğin, 350 mV'luk bir eşik değeri seçildiğinde, bu değeri 1 mΩ'luk bir şönt direnç (SR) ile ilişkilendirilirse kısa devreyi tespit etmek için kullanılan akım 4.19'daki denklemdeki gibi hesaplanabilir. Örneğin gerilim $V = 350 \text{ mV}$ ve şönt direnç $R = 1 \text{ m}\Omega$ olduğunda akım hesabı işlem 4.20'de olduğu gibi hesaplanabilir.

$$V = I \times R \quad (4.19)$$

$$I = \frac{350 \text{ mV}}{1 \text{ m}\Omega} = 350 \text{ A} \quad (4.20)$$

Yani çıkıştaki voltaj 350 mV'a ulaştığında sistemde 350 A'lık bir akım olduğunu kabul eder ve kısa devreyi tespit eder. Bu kısa devre tespiti için kullanılan eşik değerinin çıkış akımını belirlemede önemli bir rol oynadığını gösterir. Bu şekilde kısa devre olaylarına müdahale etmek ve güvenliği sağlamak için doğru eşik değerleri belirlenmiş olur.

Bu koruma sistemlerinde sadece sabit değerler kullanılabilir. Bu değerlere teknik dokümandan alınan görsel olan Şekil 4.31'den ulaşılabilir [155].

13.6.7.1 Protections:SCD:Threshold

Class	Subclass	Name	Type	Min	Max	Default	Unit
Protections	SCD	Threshold	U1	0	15	0	—

Description: This parameter sets the Short Circuit in Discharge Protection threshold for the sense resistor voltage.

- 0 = 10 mV
- 1 = 20 mV
- 2 = 40 mV
- 3 = 60 mV
- 4 = 80 mV
- 5 = 100 mV
- 6 = 125 mV
- 7 = 150 mV
- 8 = 175 mV
- 9 = 200 mV
- 10 = 250 mV
- 11 = 300 mV
- 12 = 350 mV
- 13 = 400 mV
- 14 = 450 mV
- 15 = 500 mV

Şekil 4.31 : SCD koruması sabit değerlerinin karşılıkları [155].

4.1.6 Şarj korumasında alt sıcaklık (UTC)

Şarj korumasında alt sıcaklık (Undertemperature in Charge-UTC) batarya hücresinin sıcaklığı belirli bir UTC eşik değeri altına düştüğünde bir uyarı veya hata sinyali üretir. Bu eşik -40°C ile 120°C arasında 1°C adımlarla programlanabilir. Eğer hücre sıcaklığı bu eşik değeri aşarsa cihaz UTC korumasını tetikler ve bir uyarı sinyali gönderir [155].

4.1.7 Deşarj korumasında alt sıcaklık (UTD)

Deşarj korumasında alt sıcaklık (Undertemperature in Discharge-UTD) koruması hücre sıcaklığı belirli bir UTD eşik değeri altına düştüğünde bir uyarı veya hata tetikler. UTD eşik değeri -40°C ile 120°C arasında 1°C adımlarla programlanabilir.

Burada aynı UTC korumasındaki mantıkta olduğu gibi sıcaklık bu eşik değerin altına düştüğünde UTD koruması devreye girer ve bir uyarı sinyali gönderir [155].

4.1.8 Dahili alt limit sıcaklık koruması (UTINT)

Dahili alt limit sıcaklık koruması (Internal Undertemperature Protection-UTINT) dahili sıcaklık yani entegrenin kendi sıcaklığı belirli bir UTINT eşik değeri altına düştüğünde bir uyarı veya hata tetikler. UTINT eşik değeri -40°C ile 120°C arasında 1°C adımlarla programlanabilir. Sıcaklık bu eşik değerinin altına düştüğünde UTINT koruması devreye girer ve bir uyarı sinyali gönderir [155].

4.1.9 Dahili aşırı sıcaklık koruması (OTINT)

Dahili aşırı sıcaklık koruması (Internal Overtemperature Protection-OTINT) dahili sıcaklık yani entegrenin kendi sıcaklığı belirli bir OTINT eşik değeri üzerine çıktığında veya ona eşit olduğunda bir uyarı veya hata tetikler. OTINT eşik değeri aralığı -40°C ile 120°C 1°C adımlarla programlanabilir. Aynı UTINT korumasında olduğu gibi sıcaklık bu eşik değerinin üzerine çıktığında OTINT koruması devreye girer ve bir uyarı sinyali gönderir [155].

4.1.10 Şarjda aşırı sıcaklık koruması (OTC)

Şarjda aşırı sıcaklık koruması (Overtemperature in Charge Protection-OTC) hücre sıcaklığı belirli bir OTC eşik değeri üzerine çıktığında veya ona eşit olduğunda bir uyarı veya hata tetikler. OTC eşik değeri -40°C ile 120°C arasında 1°C adımlarla programlanabilir. Sıcaklık bu eşik değerinin üzerine çıktığında OTC koruması devreye girer ve bir uyarı sinyali gönderir [155].

4.1.11 Deşarjda aşırı sıcaklık koruması (OTD)

Deşarjda aşırı sıcaklık koruması (Overtemperature in Discharge Protection-OTD) hücre sıcaklığı belirli bir OTD eşik değeri üzerine çıktığında veya ona eşit olduğunda bir uyarı veya hata tetikler. OTD eşik değeri -40°C ile 120°C arasında 1°C adımlarla programlanabilir. Sıcaklık bu eşik değerinin üzerine çıktığında OTD koruması devreye girer ve bir uyarı sinyali gönderir [155].

4.1.12 FET aşırı sıcaklık koruması (OTF)

FET aşırı sıcaklık koruması (Overtemperature FET Protection-OTF) FET sıcaklığı belirli bir OTF eşik değeri üzerine çıktığında veya ona eşit olduğunda bir uyarı veya hata tetikler. OTF eşik değeri 0°C ile 150°C arasında 1°C adımlarla programlanabilir. FET sıcaklığı bu eşik değerinin üzerine çıktığında OTF koruması devreye girer ve bir uyarı sinyali gönderir [155].

4.1.13 Dengeleme süresi

Dengeleme süresi (Cell Balance Interval) dengeleme işlemine izin veren zaman aralığıdır. Bu ayar dengeleme işleminin ne kadar süreyle devam edeceğini belirler. Dengeleme algoritması bu süre sonunda hangi hücrenin dengeleme yapılacağını tekrar hesaplar. Bir hücre seçildiğinde dengeleme bu hücre üzerinde belirtilen süre boyunca devam eder ancak dengeleme koşulları engellenirse işlem durur. Bir hücreyi dengeleme komutu verildiğinde dengeleme işlemi belirtilen süre boyunca aktif kalır ve host, komutun süresi bitmeden tekrar gönderilirse, dengeleme süresi sıfırlanır ve işlem devam eder. Süresi 1 saniye ile 255 saniye aralığında değiştirilebilir [155].

4.1.14 Dengeleme minimum delta değeri

Dengeleme minimum delta değeri (Cell Balance Min Delta (Charge)) şarj sırasında maksimum ve minimum hücre voltajları arasındaki farkın belirli bir değeri aşması gerektiğini belirler. Bu fark belirtilen değerden büyük olduğunda otomatik hücre dengeleme işlemi başlar. Değer aralığı 0 ile 255 mV aralığında değiştirilebilir [155].

4.1.15 Dengeleme minimum hücre voltaj değeri

Dengeleme minimum hücre voltaj değeri (Cell Balance Min Cell Voltage Charge) şarj sırasında dengeleme işlemine başlanabilmesi için gereken minimum hücre voltajı eşik değeridir. Şarj esnasında hücre voltajı bu değer altına düşerse otomatik dengeleme devre dışı bırakılır. Değer aralığı 0 ile 5000 mV aralığında değiştirilebilir [155].

Tüm bu koruma işlemleri BQ entegresinin koruma yöntemlerinden sadece birkaçıdır. Daha detaylı bilgi için teknik belge incelenebilir [155]. Batarya yönetim sistemin ana görevlerini yerine getirebildiği testler sonrasında başarıyla doğrulanmıştır. Bu aşamadan sonra asıl yenilikçi yaklaşımlar sunan IoT işlemleri aşamasına geçilmesi planlanarak çalışmalara başlanmıştır.

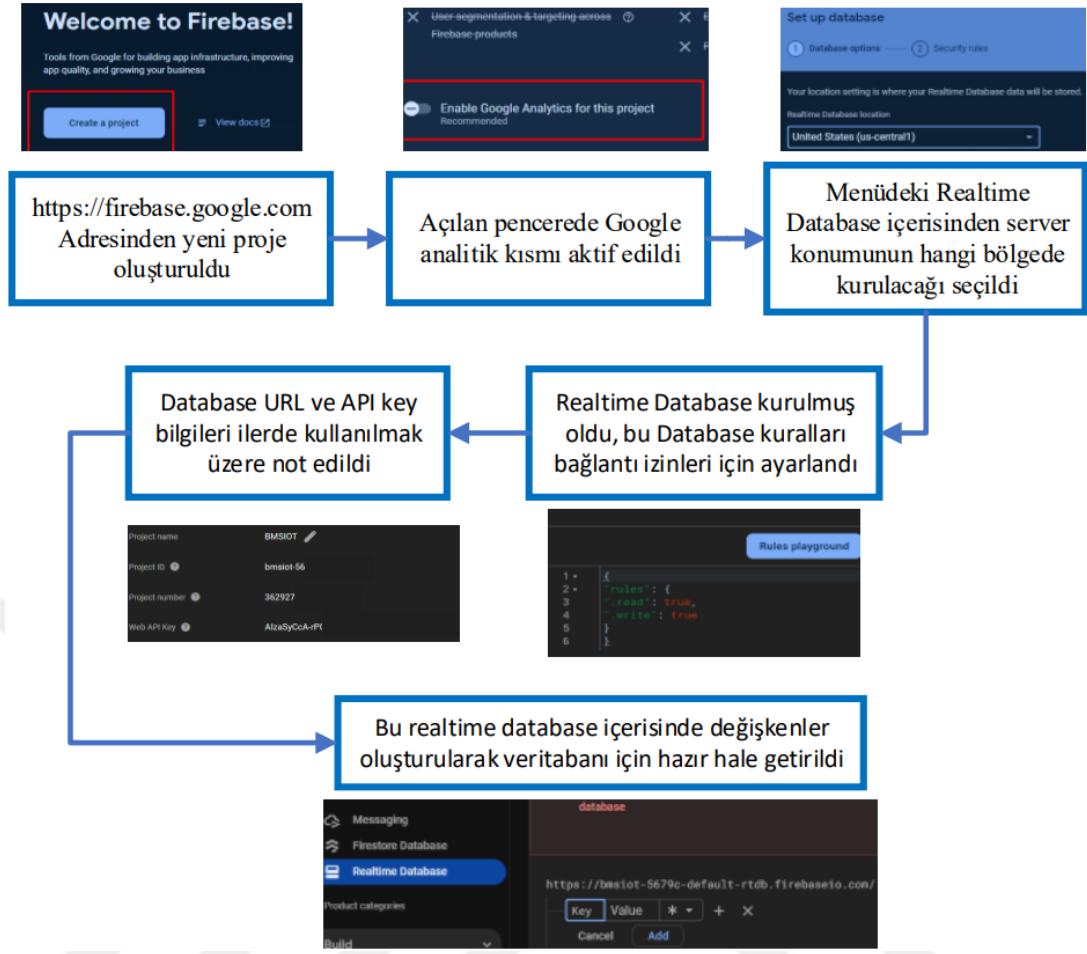
4.2 Veritabanı ve IoT Kontrolcü Bağlantısı Deneysel Çalışma

Tezde materyal ve yöntemler başlığı altında bahsedildiği gibi veri tabanı işlemleri için Firebase platformu ve verileri Firebase ortamına gönderecek olan IoT işlemlerini halledecek olan kontrolcü olarak da ESP32 kullanılacaktır. Bu işlemleri gerçekleştiren ESP32 için oluşturulmuş geliştirme kartından yararlanıldı. Bu ESP32 geliştirme Batarya yönetim sistemi ana kontrolcünden aldığı verileri veri tabanı platformu olan Firebase'e gönderecek şekilde ayarlanacaktır. ESP32 ve Firebase platform logosuna Şekil 4.32'de yer verilmiştir.



Şekil 4.32 : ESP32 ve Firebase platform bağlantısı.

Buradaki ilk etapta Firebase platformunun ayarlamaları ve ESP32 için gerekli olabilecek izin durumları işlemleri gerçekleştirildi. Adım adım işlemler gerçekleştirildi. Bu işlemlerde öncelikle Firebase web adresinden yeni proje oluşturuldu. Ardından açılan pencerede Google analitik kısmı aktif edildi. Daha sonra Menüdeki Realtime Database menüsü içerisinde server konumunun hangi bölgede kurulacağı seçildi. Bu kurulmuş olan Realtime Database kuralları bağlantı izinleri için ayarlandı, yani ESP32 ve diğer izleme araçlarının Firebase'de bulunan verileri okuması için gerekli olan kurallar aktif edildi. Daha sonra Database URL ve APIkey bilgileri ileride kullanılmak üzere not edildi. Bu bilgileri hem ESP32 hem de izleme uygulamalarında Firebase bağlantısında kullanılacak olan bilgileri içermektedir. Son olarak bu oluşturulan Realtime Database içerisinde değişkenler oluşturularak veri tabanı için hazır hale getirildi. Tüm bu işlemler sonucunda Firebase platformu veri tabanı olarak kullanılması için ayarlanmış oldu. Tüm bu yapılan işlemlerin küçük resimler ve metin kutularıyla anlatılmış olan diyagramına Şekil 4.33'te yer verilmiştir.



Şekil 4.33 : Firebase ayarlamalarının yapılması.

ESP32 üzerinden verilerin veri tabanına gönderilmesi için internet bağlantısının olması gerekmesi üzerine araştırmalar bu yönde yoğunlaştırıldı. ESP32 wifi bağlantısının sağlanması için gerekli yazılımlar oluşturuldu. Yazılan kod içerisine internet bağlantısının doğru gerçekleştiğini anlamak üzere yazılım içerisine seri monitöre yazdırılacak bir bilgi yazısı eklendi bu yazı ile bağlantı sağlandığında IP adresiyle birlikte seri monitöre bilgi yazdırılması sağlanmış oldu.

ESP32 wifi bağlantısının yazılımı test edildi herhangi bir problem olmadan wifi bağlantısı gerçekleştirildi. Bu test işlemleri ve yazılım koduna Şekil 4.34'te yer verilmiştir. Bağlantı gerçekleşmesi ve IP adresinin alındığı bilgi bölümü sarı ile işaretlenmiştir.

Bu aşamadan sonra ESP32'nin Firebase platformuna giriş işlemi yapması ve verilerin anlık olarak gönderilmesi işlemi yapmakla sorumlu olacaktır. Bu aşamaların gerçekleşmesi için hangi aşamaların gerçekleşmesi gerektiği adım adım araştırıldı.

```
1 #include <WiFi.h> // WiFi kütüphanesi, ESP32 için gerekli
2
3 // Wifi ağ bilgileri
4 #define WIFI_SSID "ETHEM" // Wifi ağı ismi (SSID)
5 #define WIFI_PASSWORD "12345678" // Wifi ağı şifresi
6
7 void setup() {
8   Serial.begin(115200); // Seri haberleşme başlatılır, baud rate 115200 olarak ayarlandı
9   WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Belirtilen SSID ve şifre ile Wifi ağına bağlanmayı başlatır
10  Serial.print("Ağa bağlanıyor"); // Bağlantı durumu seri port üzerinden kullanıcıya bildirilir
11  delay(3000); // İlk deneme için kısa bir bekleme süresi
12
13  // Wifi ağına bağlanma durumu kontrol edilir
14  while (WiFi.status() != WL_CONNECTED) { // Bağlantı başarılı olana kadar döngü çalışır
15    Serial.print("."); // Her denemede bir nokta yazdırılır
16    delay(300); // Kısa bir bekleme
17  }
18
19  // Wifi ağına başarıyla bağlandığında seri port üzerinden bilgi verilir
20  Serial.println();
21  Serial.print("Bağlandı. IP Adresi: "); // Başarılı bağlantı mesajı
22  Serial.println(WiFi.localIP()); // Cihazın atanmış olan IP adresi yazdırılır
23 }
24
25 void loop() {
26
27 }
28
```

Output Serial Monitor x

Message (Enter to send message to 'ESP32 Dev Module' on 'COM8')

Ağa bağlanıyor
Bağlandı. IP Adresi: 192.168.43.253

Şekil 4.34 : ESP32 internet bağlantısının gerçekleştirilmesi.

Wifi ağına bağlanma kodunda Şekil 4.34'te görüleceği üzere kodda önceden belirlenmiş Wifi SSID (ağ adı) ve şifre bilgileri kullanılarak ESP32'nin ağa bağlanması sağlanmıştır. Bağlantının başarılı bir şekilde kurulmasının ardından cihazın aldığı IP adresi seri port üzerinden görüntülenmiştir. Bu süreç ESP32'nin internete bağlanabilirliğini doğrulamak ve iletişim altyapısının işlevselliğini test etmek için temel bir adım olarak tasarlanmıştır. Bu işlemin gerçekleştirilmesi ile IoT projelerinde temel bir gereklilik olan ağ bağlantısı sağlama başlangıç olarak ilk adımı atmıştır.

İnternet bağlantısının ardından ESP32 Firebase bağlantısı için gerekli yazılım yapıları oluşturuldu. Şekil 4.35'te Firebase platformu kurulduğunda not edilen URL ve APIkey değerleri atamalarının yapılması ve Firebase bağlantısının sağlanması için gerekli olan temel yapıdaki yazılımların bir kısmı eklenmiş hali gösterilmektedir. Şekil 4.35'te görüldüğü üzere Wifi bağlantısı kurulumu için wifi SSID (ağ adı) ve şifre bilgileri tanımlanmış ve Firebase ile iletişim kurmak için API anahtarı ve veri tabanı URL'si yapılandırılmıştır.

Firestore bağlantısının gerçekleştirildiğine dair bilgilendirme aynı wifi bağlantısında yapıldığı gibi seri monitöre yazdırılacak şekilde yazılım eklentileri yapılmıştır. Bu yapılan işleme Şekil 4.35'te yer verilmiştir.

```

1 #include <WiFi.h> // WiFi kütüphanesi, ESP32'nin WiFi bağlantısı için gerekli
2 #include <Firebase_ESP_Client.h> // Firebase etkileşimi için gerekli kütüphane
3 #include "addons/TokenHelper.h" // Firebase token yönetimi için yardımcı kütüphane
4 #include "addons/RTDBHelper.h" // Firebase gerçek zamanlı veritabanı işlemleri için yardımcı kütüphane
5
6 // Zamanlayıcılar ve sabitler
7 unsigned long previousMillis = 0;
8 const long interval = 10000; // ESP32'nin veri gönderme ve alma işlemleri için zaman aralığı
9
10 // WiFi ağ bilgileri
11 #define WIFI_SSID "ETHEM" // WiFi ağı ismi (SSID)
12 #define WIFI_PASSWORD "12345678" // WiFi ağı şifresi
13
14 // Firebase bağlantı bilgileri
15 #define API_KEY "AIzaSyCCA-rPQMae7NjgQra9" // Firebase API anahtarı
16 #define DATABASE_URL "https://bmsiot" // Firebase gerçek zamanlı veritabanı URL'si
17
18 // Seri haberleşme pin tanımları (opsiyonel, projeye özgü olabilir)
19 #define TXD_PIN (GPIO_NUM_1)
20 #define RXD_PIN (GPIO_NUM_3)
21
22 float Bat_Pack_Soc_value, Bat_Pack_Temp_value; // Batarya durumu ve sıcaklık değerleri
23
24 // Firebase ile veri aktarımı için gerekli nesneler
25 FirebaseData firebaseData;
26 FirebaseJson json; // Veri nesnesi
27 FirebaseData fbdo; // Firebase veritabanı nesnesi
28 FirebaseAuth auth; // Firebase kimlik doğrulama nesnesi
29 FirebaseConfig config; // Firebase yapılandırma nesnesi
30
31 unsigned long sendDataPrevMillis = 0; // Veri gönderim zamanı kontrolü
32 unsigned long getDataPrevMillis = 0; // Veri alma zamanı kontrolü
33 bool signupOK = false; // Firebase'e başarılı bağlantı durumu

```

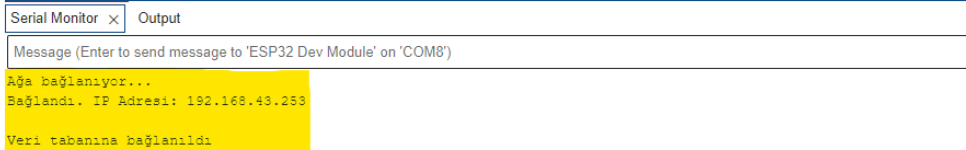
Şekil 4.35 : Firebase yapılandırmaları için gerekli yazılımların bir kısmı.

Şekil 4.36'da görüldüğü üzere Firebase.signUp() fonksiyonu ile Firebase sunucularına bağlantı sağlanmıştır. Verilerin belirli zaman aralıklarıyla gönderimi, millis() fonksiyonu ile kontrol edilerek sistemin zaman yönetimi optimize edilmiştir.

```

45 void setup() {
46   Serial.begin(115200); // Seri haberleşme başlatılır, baud rate 115200 olarak ayarlandı
47   WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Belirtilen SSID ve şifre ile WiFi ağına bağlanmayı başlatır
48   Serial.print("Ağa bağlanıyor"); // Bağlantı durumu seri port üzerinden kullanıcıya bildirilir
49   delay(3000); // İlk deneme için kısa bir bekleme süresi
50
51   // WiFi ağına bağlanma durumu kontrol edilir
52   while (WiFi.status() != WL_CONNECTED) { // Bağlantı başarılı olana kadar döngü çalışır
53     Serial.print("."); // Her denemede bir nokta yazdırılır
54     delay(300); // Kısa bir bekleme
55   }
56
57   // WiFi ağına başarıyla bağlandıığında seri port üzerinden bilgi verilir
58   Serial.println();
59   Serial.print("Bağlandı. IP Adresi: "); // Başarılı bağlantı mesajı
60   Serial.println(WiFi.localIP()); // Cihaza atanmış olan IP adresi yazdırılır
61   Serial.println();
62
63   // Firebase yapılandırması ve bağlantı kontrolü
64   config.api_key = API_KEY;
65   config.database_url = DATABASE_URL;
66
67   if (Firebase.signUp(&config, &auth, "", "")) {
68     Serial.println("Veri tabanına bağlanıldı");
69     signupOK = true; // Bağlantı başarılı
70   } else {
71     Serial.printf("%s\n", config.signer.signupError.message.c_str()); // Hata mesajı
72   }
73
74   config.token_status_callback = tokenStatusCallback; // Token durum geri çağırma işlevi
75   Firebase.begin(&config, &auth); // Firebase başlatılır
76   Firebase.reconnectWiFi(true); // WiFi bağlantısı yeniden bağlanabilir
77 }
78

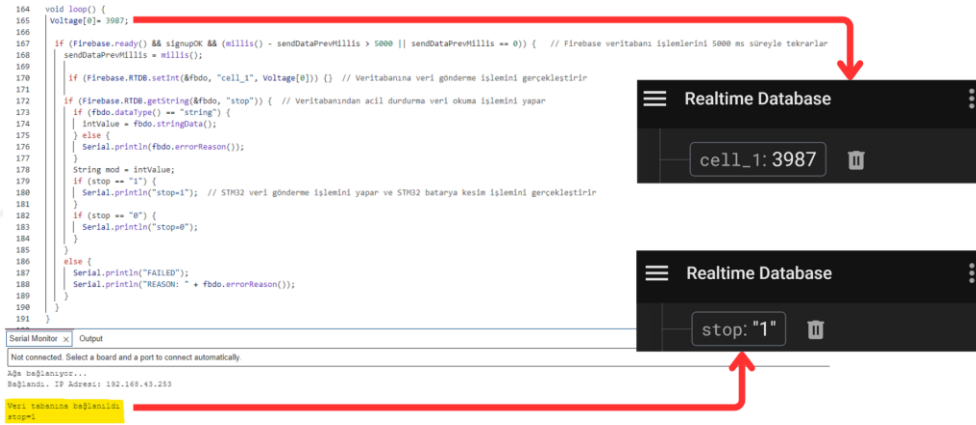
```



Şekil 4.36 : Firebase bağlantısı sağlanması ve seri monitöre bilgi yazdırılması.

ESP32 aslında sadece veri gönderme işlemi yapmayacak ayrıca veri izleme ve kontrol arayüz programlarından gönderilen parametre değişiklik ayarlarını da okuyabilecektir. Bu parametre ayarlarından kasıt batarya yönetim sistemi tasarımı başlığı altında gösterilen BQ76952 entegresinin koruma yeteneklerini parametre ayar değişikliklerinden bahsedilmektedir. Parametre ayarlarının haricinde cihazın uzaktan batarya bağlantısının kesilmesini istediğinde bu kesim işleminin geldiğini veri tabanı üzerinden okumasının gerçekleştirilmesi gereklidir.

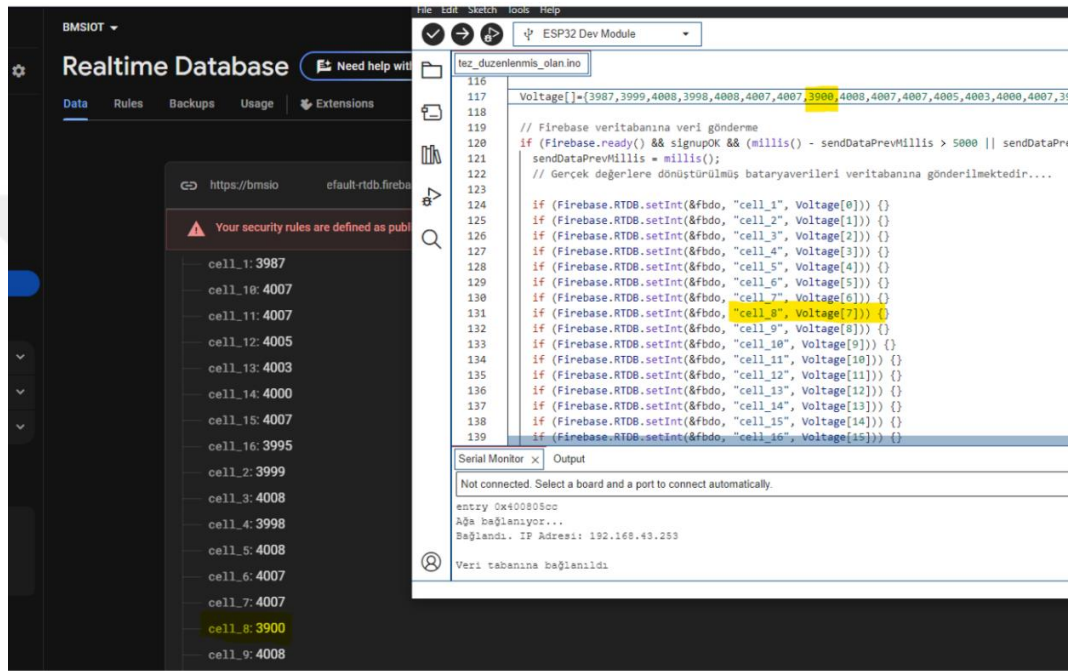
Hem veri gönderme hem de veri okuma işlemleri için örnek yapılar oluşturulmuştur. Bu yapılarda rastgele bir değer olarak hücre-1 değeri 3987 mV belirlenmiştir. Bu değer ESP32'nin çalıştığı bir göstergesi olması için rastgele verilmiştir. Bu hücre Firebase platformunda cell_1 değişkeni olarak adlandırılmıştır. Ayrıca durdurma işlemi örneği için stop adında bir değişken oluşturulmuştur. Bu yapının yazılımda eklenmesi gereken kısımları eklenmiştir. Bu yapıya Şekil 4.37'de yer verilmiştir.



Şekil 4.37 : ESP32 ile Firebaseden veri okuma ve gönderme işlemi.

Şekil 4.37'de görüldüğü üzere Firebase veri tabanında veri yazma işlemi için Firebase.RTDB.setInt() fonksiyonu kullanılarak verileri kaydedilmiştir veri okuma işlemi için ise Firebase.RTDB.getString() fonksiyonu kullanılarak parametreler alınmış ve sistemin çalışma modu bu verilere göre güncellenmiştir. Verilerin belirli zaman aralıklarıyla gönderimi, millis() fonksiyonu ile kontrol edilerek sistemin zaman yönetimi optimize edilmiştir. Bağlantı sırasında oluşabilecek hataların fark edebilmek için fbdo.errorReason() fonksiyonu ile hata nedenleri detaylı şekilde konsola yazdırılmış ve sistemin güvenilirliği artırılmıştır. Sonuç olarak ESP32 kullanılarak Firebase Realtime Database ile başarılı bir haberleşme sağlanmış, ölçülen veriler güvenli bir şekilde veri tabanında saklanmış ve sistemin uzaktan kontrolü mümkün hale getirilmiştir.

Bu çalışma ilerleyen sürecin en büyük temellerini atarak ilerleyen süreçteki yapılacak olan tüm işlemlerin içerisinde bu yöntemler kullanılarak işlemler gerçekleştirilecektir. Bu yapılan işlemlerin ardından bir hücre için yapılan işlemler diğer 15 hücre içinde tekrarlanarak hem ESP32 yazılımına hem Firebase platformuna değişken olarak eklenmiştir. Bu yapılan işlemde batarya verileri bir dizi içerisinde rastgele değerler ile atama yapılarak verilerin doğruluğu kontrol edilmiştir. Bu yapıyı görseline Şekil 4.38’de yer verilmiştir.



Şekil 4.38 : Tüm batarya verilerinin Firebase ortamına gönderilmesi.

Bu işlemlerin aynısı batarya yönetim sistemi başlığı altındaki koruma parametrelerinin kontrollerinin yapılabilmesi için de eklenmiştir ve okumaları gerçekleştirilmiştir. Bu yapının görseline Şekil 4.39’da yer verilmiştir.

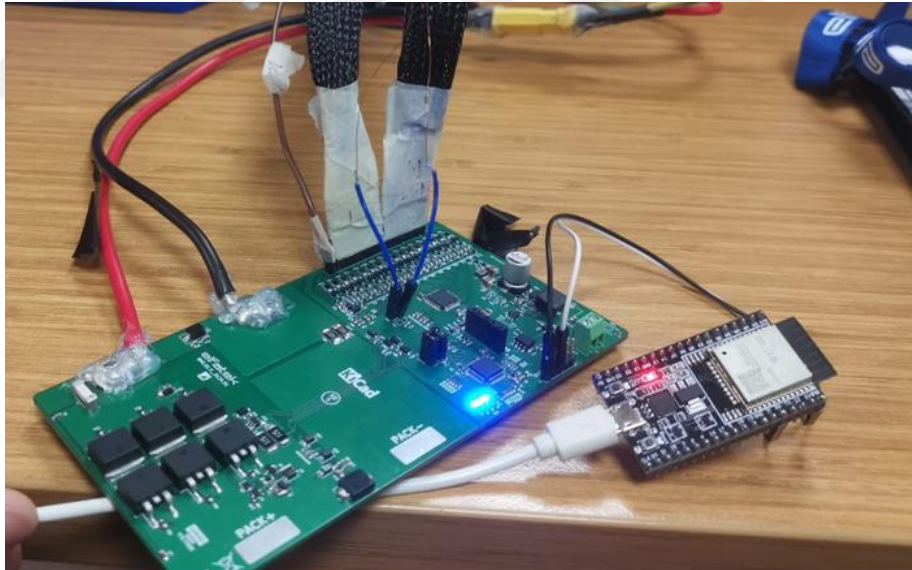


Şekil 4.39 : Koruma Parametrelerinin Firebase üzerinden okunması.

Bu sayede hem rastgele belirlenmiş olan batarya hücrelerinin tüm verilerinin Firebase platformuna iletilmesi sağlanmış oldu hem ileride kullanılacak olan tüm batarya koruma parametrelerinin Firebase platformu üzerinden okuması gerçekleştirilmiş oldu. Bu da planlanan sistemin büyük bir kısmının başarıyla tamamlandığının göstergesi olmuş oldu. Bu sayede kullanıcı tüm batarya verilerini tasarlanacak olan arayüz programlarından takip edebilecek ve batarya parametre ayarlarını uzaktan kontrol ederek değiştirebilir şekilde yeteneklere sahip olacaktır.

4.3 IoT Kontrolcü BMS Veri Akışı Deneysel Çalışma

ESP32 veri paketlerini gönderebiliyor alabiliyor fakat bu işlemi rastgele değerler üzerinden gerçekleştirebiliyor şekilde uygulama gerçekleştirilmişti. Artık bu verilerin tamamen gerçek değerler üzerinden devam etmesi için gerekli çalışmalar başlatıldı. Öncelikle batarya yönetim sistemi devresi ve ESP32 donanımsal olarak yani haberleşmenin sağlanabileceği şekilde donanımsal bağlantı gerçekleştirildi bu bağlantı görseline Şekil 4.40'ta yer verilmiştir.



Şekil 4.40 : BMS ve ESP32 Donanımsal bağlantı.

Değerler mV olarak gösterilmesi planlandığından tek bir veri paketi ile UART haberleşmesi üzerinden göndermek mümkün değildir. Bu yüzden veriler parçalanarak gönderilmesi planlanmıştır. Güvenli veri iletimi için mesaj içeriğinin de iki adet başlangıç bayrak atması gerçekleştirilmiştir. Bu bayrak atamaları aslında verinin geldiğini gördükten sonra ilk etapta veri paketini almak için iki başlangıç bayrak atmasını kontrol eder ardından tüm değerleri kabul eder. Bu yapının ilk etapta STM32

kontrolcüsü tarafındaki veri gönderme yazılımları eklendi. Bu yazılımın kısa bir yapısına Şekil 4.41’de yer verilmiştir.

```

661
662 void UART_Send_Data(void) {
663     uint8_t buf = 2;
664     V_Send[0] = 0x55; // STM32 veri gönderme bayrak ataması
665     V_Send[1] = 0x55; // STM32 veri gönderme bayrak ataması
666
667     uint8_t Adres_Val = Cell1Voltage;
668     //RX_data is global var
669     for (int i = 0; i < 16; i++) {
670
671         DirectCommands(Adres_Val, 0x00, R);
672         V_Send[buf++] = RX_data[0];
673         V_Send[buf++] = RX_data[1];
674         Adres_Val += 2;
675     }

```

Şekil 4.41 : STM32 üzerinden UART ile batarya verilerini gönderme.

Bu yapılan işlemin karşılığı olarak ESP32 kontrolcüsünde de bu işlemleri okuyacak yapılar eklendi. STM32’nin gönderdiği verileri okuyan yazılımın kısa bir yapısına Şekil 4.42’de yer verilmiştir.

Ayrıca batarya sıcaklık değerini gönderen STM32 verileri yine ESP32 içerisinde santigrat derece olacak şekilde çevirme işlemleri sonrası Firebase platformuna göndermektedir. SoC değeri ana yoğunlaşılacak konu olmadığı için SoC hesabı net yapılmadan herhangi bir değer gönderilecek şekilde planlama yapılmıştır. ESP32’nin verileri STM32 kontrolcüsünün gönderdiği değerleri okuyup ardından Firebase platformuna gönderdiği yazılımın bir kısmına Şekil 4.42’de yer verilmiştir.

```

79 void loop() {
80     // STM32 üzerinden gönderilen veri alınıp işlenir
81     // Veri gönderiminde bayrak kullanılmıştır
82     // Hatalı veri işlemek için ilk iki verinin 0x55 0x55 şeklinde gelmesi kontrol edilir
83     // Eğer bu iki veri 0x55 ise geri kalan verilerin alınıp işlenmesini sağlar
84
85     if (Serial.available() > 0) {
86         for (int i = 0; i < 36; i++) {
87             Lt_Buffer[i] = Serial.read();
88
89             if (Lt_Buffer[0] == 0x55 && check_counter == 0 && D_check_count == 0) { // ilk veri 0x55 mi evetse devam et
90                 Big_Buffer[0] = Lt_Buffer[0];
91                 check_counter = 1;
92             } else if (Lt_Buffer[0] == 0x55 && check_counter == 1 && D_check_count == 0 && Big_Buffer[0] == 0x55) { // ikinci veri 0x55 mi evetse devam et
93                 Big_Buffer[1] = Lt_Buffer[0];
94                 check_counter = 2;
95                 D_check_count = 1;
96             } else if (D_check_count == 1 && check_counter >= 2 && Lt_Buffer[0] != 0x55) { // kontrolleri yap ve verileri almaya başla
97                 Big_Buffer[check_counter++] = Lt_Buffer[0];
98             }
99             if (check_counter >= 36) { // Tüm veriler alındıktan sonra kontrolleri sıfırla
100                 check_counter = 0;
101                 D_check_count = 0;
102             }
103         }
104         for (int i = 0; i < 32; i++) {
105             Raw[i] = Big_Buffer[i + 2]; // Batarya verileri alındıktan sonra parçalı veriler ataması yapılır
106         }
107     }
108
109     for (int i = 0; i < 16; i++) {
110         Voltage[i] = Raw[buf + 1] * 256 + Raw[buf]; // Parçalı veriler gerçek gerilim değerine dönüştürülecek şekilde hesaplama işleminden geçirilir
111         buf += 2;
112     }
113
114     buf = 0;
115     Bat_Pack_Temp_value = (0.1 * (float)(Big_Buffer[35] * 256 + Big_Buffer[34])) - 273.15; // Batarya sıcaklığı dönüştürme hesaplaması
116 }

```

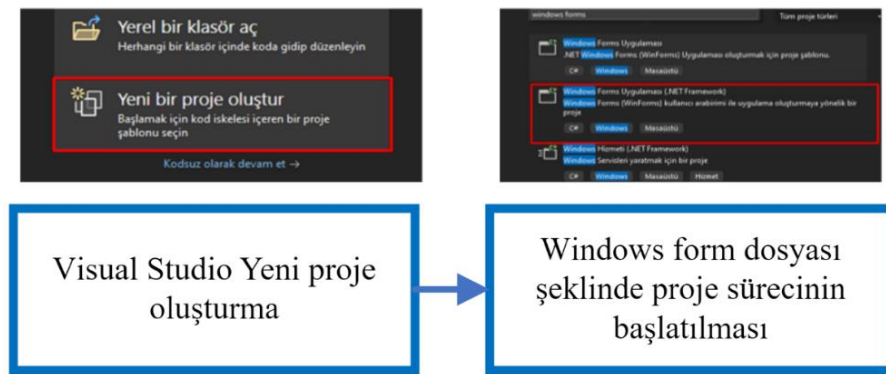
Şekil 4.42 : STM32’nin gönderdiği verileri okuyan ESP32.

Sonuç olarak ESP32 mikrodenetleyicisi kullanılarak STM32 cihazından gelen verilerin UART protokolü üzerinden alınıp Firebase Realtime Database'e gönderilmesi sağlanmıştır. ESP32 ilk olarak Wifi ağına bağlanarak Firebase ile entegrasyon için gerekli olan bağlantıyı kurdu. Bu süreçte API anahtarı ve veri tabanı URL'si kullanılarak Firebase yapılandırması gerçekleştirildi. STM32 cihazı üzerinden gelen veriler UART protokolü ile ESP32'ye iletilmiş ve başlangıç bayrağı kontrolü ile doğrulandıktan sonra tampon bellekte saklandı.

Bu veriler bataryaya ait sıcaklık ve hücre voltajı gibi parametrelerin hesaplanmasında kullanıldı. Elde edilen bu veriler Firebase'e kaydedilerek uzaktan erişilebilir hale getirildi. Firebase ile veri paylaşımı sırasında bağlantı durumu sürekli kontrol edilir ve olası hatalar tespit edilerek sistemin güvenilirliği artırılmış oldu. Bu sistem sayesinde STM32 ile ESP32 arasında başarılı bir veri iletişimi sağlanmış ve IoT tabanlı bir çözüm geliştirilmiştir. Artık sıra bu yapıları izleme ve kontrol işlemlerini yapacak olan yazılımlar ve tasarımlardadır.

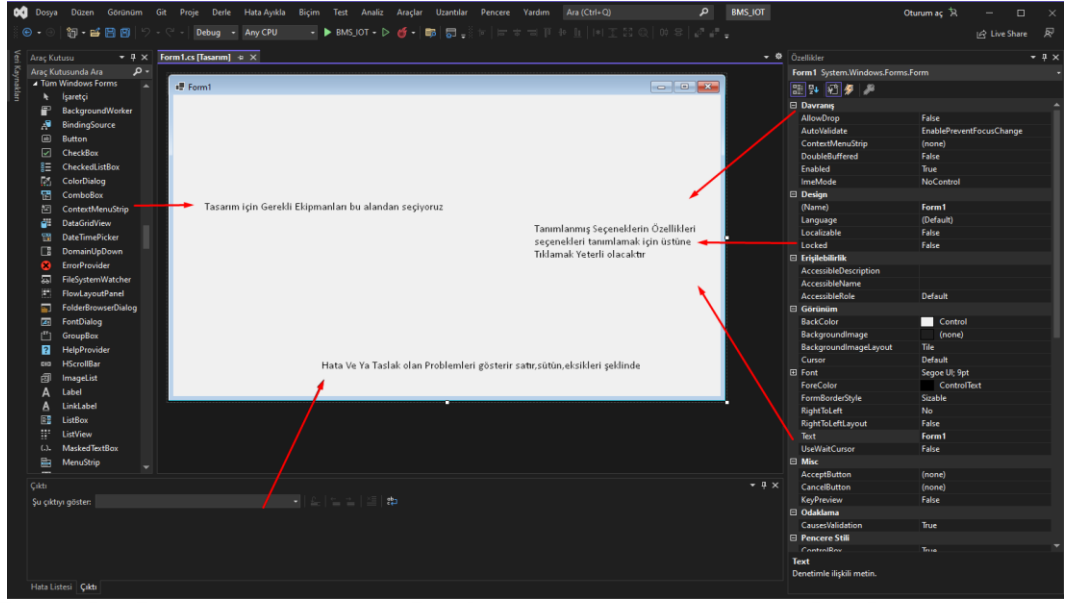
4.4 Bilgisayar Yazılımı ve Görsel Tasarım Deneysel Çalışma

Bilgisayar yazılım işlemleri materyal ve yöntemler başlığı altında anlatıldığı gibi Visual Studio platformu kullanılacaktır. Arayüz tasarım programına Visual Studio programı içerisinde yeni form projesi oluşturulacak şekilde işlemlere başlandı. Bu adımların görseline Şekil 4.43'te yer verilmiştir.



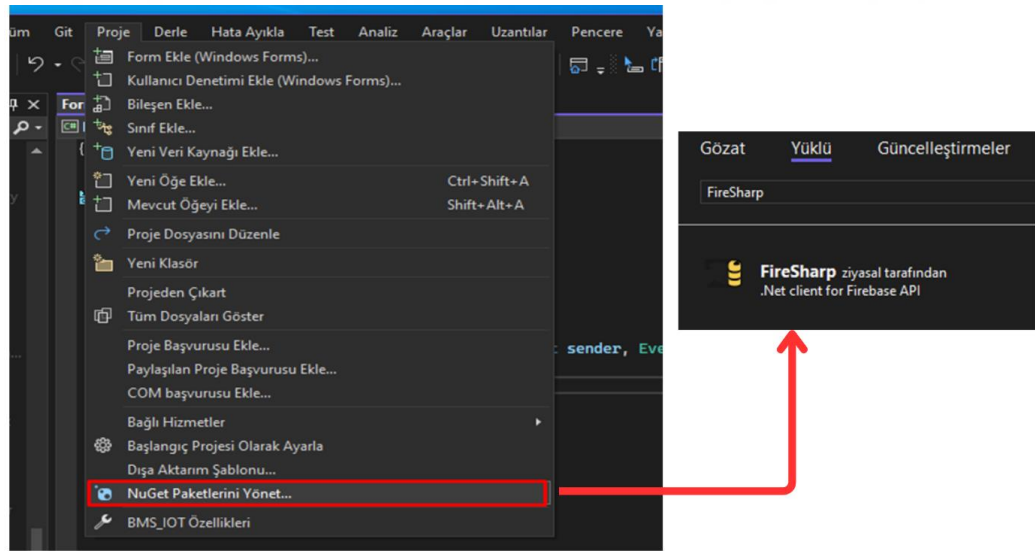
Şekil 4.43 : Visual Studio ile yeni proje başlatma.

Başlatılan bu proje içerisinde çalışmalara başlanmıştır. Bu program arayüzündeki kullanılacak olan yapılardan bazıları Şekil 4.44'te açıklanmıştır. Burada genel olarak arayüzün sol kısmında projede kullanılacak araçları ve sağ tarafından proje detayları ve eklenen araçların ayarlarının değiştirilebileceği alan mevcuttur.



Şekil 4.44 : Visual Studio genel yapısı ve anlatımlar.

Visual Studio programının Firebase platformuna bağlanması gereklidir. Çünkü tüm batarya verilerini ve kontrol işlemlerini bu şekilde gerçekleştirilecektir. Burada programın Firebase tanıyabilmesi için NuGet paketlerinden FireSharp eklentisi Visual Studio programına eklentisi gerçekleştirildi. Bu yapının kurulmasının görseline Şekil 4.45'te yer verilmiştir.



Şekil 4.45 : Firebase Bağlantısı için gerekli eklentinin gerçekleştirilmesi.

Bu eklenti sonrası işlemlerin gerçekleştirilebilmesi için gerekli yazılım kütüphane atamaları gerçekleştirildi. Bu atamalar Firebase bağlantı işlemlerinin internet bağlantısı sonrasında gerçekleştirilebilmesi için şart olan kütüphane yapılarıdır. Bu yapılara Şekil 4.46'da yer verilmiştir.

```
// Proje içinde kullanılan gerekli kütüphaneleri dahil ediyoruz
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net.Mail;
using System.Text;
using System.Threading.Tasks;
using BMS_IOT;
using FireSharp;
using System.Windows.Forms;
using FireSharp.Config;
using FireSharp.Interfaces;
using FireSharp.Response;
```

Şekil 4.46 : Firebase tanıtılma işlemleri için kütüphane eklemeleri.

Daha önce Firebase platformu oluşturulurken kaydedilen URL ve APIkey bilgileri aynı ESP32’de olduğu gibi Firebase platformunu tanıtılacak şekilde gerekli yazılım kodları eklenmiştir. Bu yapı örneğine Şekil 4.47’de yer verilmiştir.

```
// Firebase yapılandırma bilgileri tanımlama
IFirebaseConfig config = new FirebaseConfig()
{
    // Firebase Real-Time Database'in temel URL'sini belirtiyoruz
    BasePath = "https://bmsiot-56.firebaseio.com/",
    // Firebase ile iletişim kurmak için kullanılan kimlik doğrulama anahtarı
    AuthSecret = "OtCUd0ufkwka0slc",
};
IFirebaseClient client;
```

Şekil 4.47 : Firebase bilgilerinin programa tanıtılması.

Firebase platformunun bağlantısı için gerekli işlemlerin gerçekleştirilmesinin ardından herhangi bir değişken değerinin okunabilmesi için sınıf yapıları oluşturuldu. Bu sayede Firebase platformundan gelen değerleri program içinde kullanılabilecek halde bir değişkende saklanması sağlanmış olacaktır.

Bu yapının kurulması örneğine Şekil 4.48’de yer verilmiştir. Bu şekilde örnek olarak hücre-1 değeri için Firebase’de cell_1 olarak adlandırılan değer için oluşturulmuştur.

Bu oluşturulan cell_1 değeri ilerde program içeriğinde yapılması planlanan tüm işlemleri gerçekleştirmek için batarya verisini veri tabanından aldıktan sonra kaydedeceği değişken olmuş olacaktır.

```

namespace BMS_IOT
{
    22 başvuru
    internal class Class1
    {
        // Hücre voltajını saklamak için kod yapısı
        2 başvuru
        public string cell_1 { get; set; }
    }
}

```

Şekil 4.48 : Firebase tarafından okunan değerler için değişken oluşturma işlemi.

Batarya yönetim sisteminde her bir batarya hücresini gösterebilmek için bir yöntem düşünüldü ve her bilgisayar programında genelde yüklenme ekranında çıkan durum çubuğu (progressbar) yapısının batarya doluluk oranını gösterebilecek bir araç olduğu düşünülerek bu yapı denenmeye başlandı. Öncelikle hücre-1 değerinin okunabilmesi için gerekli olan yazılım yapıları eklendi. Bu yapılar sayesinde hücre-1 değeri cell_1 değişkenine kaydedilmiş olacaktır. Bu yapıların bir görseline Şekil 4.49'da yer verilmiştir.

```

private async void Form1_Load(object sender, EventArgs e)
{
    // Firebase istemcisini oluşturmak için config parametresini kullan
    client = new FireSharp.FirebaseClient(config);
    if (client == null) // Eğer bağlantı başarısız olursa kullanıcıyı bilgilendir
    {
        MessageBox.Show("Bağlantı hatası, İnternetinizi kontrol edin..");
    }
    while (true)
    {
        // Firebase'den veriyi asenkron bir şekilde alma
        FirebaseResponse response = await client.GetAsync("");

        // Gelen yanıtı, tanımlı Class1 türünde bir nesneye dönüştürme
        Class1 result = response.ResultAs<Class1>();

        // "cell_1" verisini al
        // ve string türünde temizle (gerekirse çift tırnakları kaldır)
        string c1 = result.cell_1?.Trim('"') ?? string.Empty;

        // Alınan string veriyi integer'a çevir
        int cc1 = Convert.ToInt32(c1);

        // ProgressBar'ı yapılandırma
        cell1_progressBar.Minimum = 0; // Minimum değer (örneğin 0 mV)
        cell1_progressBar.Maximum = 5000; // Maksimum değer (örneğin 5000 mV)
        cell1_progressBar.Step = 1; // Adım boyutu
        cell1_progressBar.Value = cc1; // ProgressBar'ın güncel değerini ayarla
    }
}

```

Şekil 4.49 : Firebase hücre-1 değeri okuma ve durum çubuğunda görselleştirme.

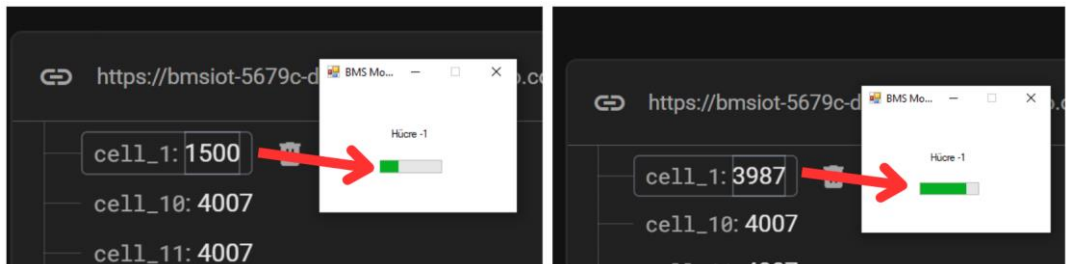
Buradaki yazılım yapısında Firebase Realtime Database ile bağlantı kurularak sensör verileri gerçek zamanlı olarak alınmış ve bir ProgressBar üzerinde görselleştirilme işlemlerini içermektedir. İlk adımda config parametresi kullanılarak Firebase istemcisi (FireSharp.FirebaseClient) yapılandırılmıştır. Bu yapılandırma Firebase veri tabanının URL'sini ve API anahtarını içermektedir. Eğer bağlantı başarısız olursa kullanıcıya bir hata mesajı gösterilerek sistemin durumu bildirilecek şekilde yazılım oluşturulmuştur.

Firestore'den sürekli veri almak için yani veri değişimlerini kaçırmamak için while(true) ifadesiyle bir döngü başlatılmıştır. Bu döngü, veri tabanı ile sürekli bir iletişim sağlar. Firestore veri tabanından veri çekmek için client.GetAsync("") metodu kullanılmıştır. Bu yöntemle verinin kök adresi boş bırakılmış (""), tüm veriler alınmıştır. Gelen veri, Firestore'den alınan Nesne Gösterimi (JSON) yapısına uygun olacak şekilde Class1 türündeki bir nesneye dönüştürülmüştür.

Alınan verinin değişken adı olan "cell_1" verisi, çift tırnak karakterlerinden arındırılarak temizlenmiş ve ardından string türünden int türüne dönüştürülmüştür. Bunun sebebi bazen veritabanındaki veriler çift tırnak var şekilde kaydedilmektedir.

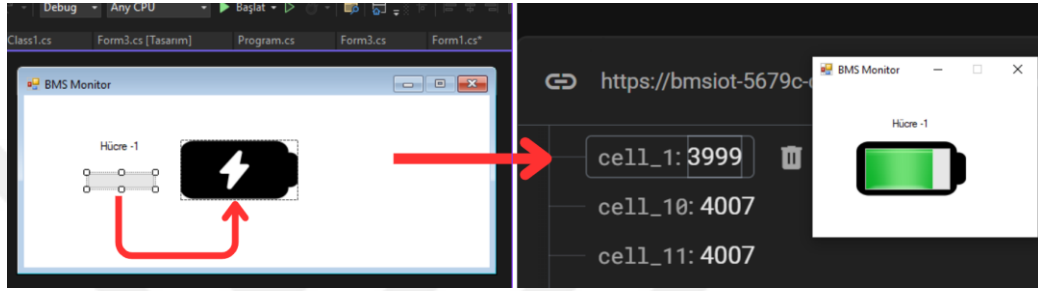
Sayısal değere dönüştürülen (cc1) durum çubuğunda görselleştirilmek üzere kullanılmıştır. Durum çubuğu için minimum (0 mV) ve maksimum (5000 mV) değerler tanımlanmış bu sayede batarya hücresinin doluluk oranı görselleştirilmiş olacaktır. Alınan veriler durum çubuğunun Value parametre ayarları özelliği üzerinden güncellenmiş ve böylece kullanıcıya hücre voltajı gibi değerlerin görsel olarak izlenebilmesi sağlanmıştır.

Sonuç olarak bu yapılar ana hedeflerden biri olan Firestore ile sürekli bağlantı kurarak sensör verilerini etkili bir şekilde almış ve bu verileri gerçek zamanlı olarak kullanıcı arayüzünde görüntülemiştir. Buradaki yapılan işlemlerin gerçekleştirilmesi Şekil 4.50'de gösterilmiştir.



Şekil 4.50 : Hücre değerinin durum çubuğunda gösterilmesi.

Burada Firebase'deki değerler elle değiştirilerek hücre görsel yapısında değişiklik olup olmadığı gözlemlenmiştir. Bu yapının değişmesiyle istenen görselleştirme ve veri okuma işlemlerinin sağlıklı bir şekilde gerçekleştirildiği gözlemlenmiş oldu. Bunun ardından bu durum çubuğunun dış hattını batarya sembolüyle kaplayarak görsel yapının daha anlaşılır olması hedeflenmiştir. Öncelikle bir batarya hücresi sembolü bulunmuş ve bu uygulamaya eklenmiştir. Ardından bu durum çubuğu o batarya simgesinin içerisine sıkıştırılacak şekilde birleştirilmiştir. Bu yapının görseline Şekil 4.51'de yer verilmiştir.



Şekil 4.51 : Durum çubuğunu batarya simgesi şekline dönüştürme.

Bu işlemlerin tamamlanmasının ardından bir diğer önemli iş olan Firebase platformuna veri gönderme işlemleridir. Bu durum ilerleyen zamanlarda acil durdurma veya parametre işlemlerini değiştirdiğinde Firebase gönderecek ve ESP32 bu verileri okuyacak ardından bu verileri ana kontrolcü olan STM32'ye iletecek ve bu sayede uzaktan kontroller gerçekleştirilmiş olacaktır.

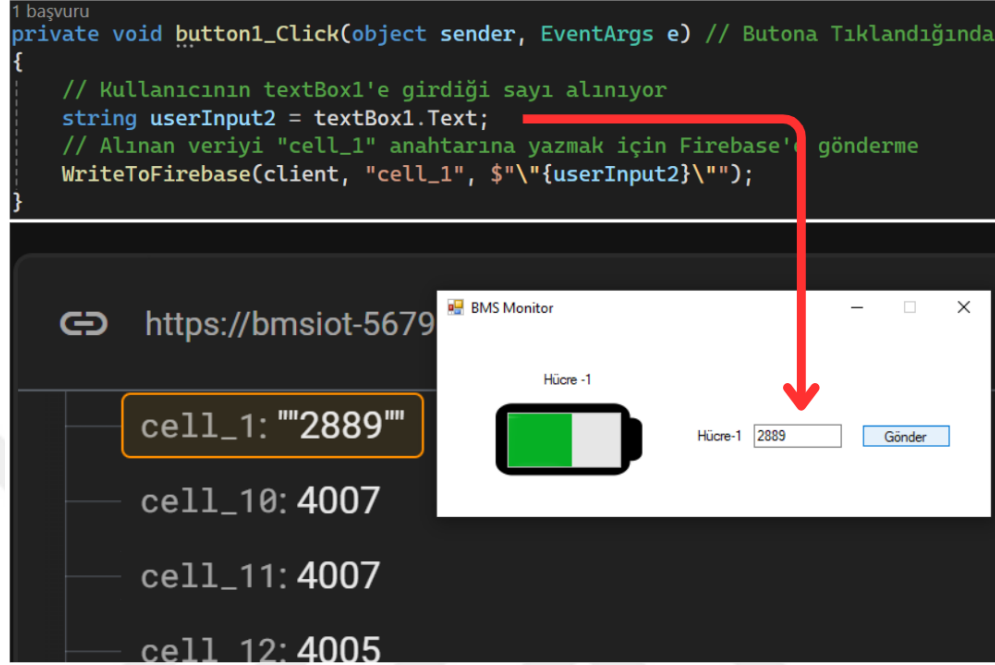
Bu işlemlerin yapılandırılması için bir dizi kod yapısı eklenmiştir. Öncelikle veriyi Firebase platformuna gönderecek olan fonksiyon kod yapısı oluşturulmuştur. Bu fonksiyon yapısının görseline Şekil 4.52'de yer verilmiştir.

```
static void WriteToFirebase(IFirebaseClient client, string nodeName, string data)
{
    var setResponse = client.Set(nodeName, data);
}
```

Şekil 4.52 : Firebase veri gönderme fonksiyonu yazılımı.

Burada veri gönderme işlemi için öncelikle bir metin kutusu aracı ve buton aracı eklendi. Buradaki hedef metin kutusu içerisine yazılan değeri Firebase ortamına gönderme işlemi sağlamaktır. Daha anlaşılır şekilde söylemek gerekirse kullanıcı butona bastığında metin kutusu içerisindeki değer Firebase platformuna gönderilecek şekilde tasarlanmıştır. Bu yapı kurulurken Firebase'de güncellenecek değer olarak yine cell_1 değeri belirlenmiştir. Yani metin kutusuna girilen ve buton aracılığıyla

gönderilen bu değer hücre değerinde değişime sebep olacaktır. Bu sayede hem veri gönderim hem veri okuma işlemi gerçekleştirilmiş olacaktır. Bu kod yapısı eklentisi ve deneme görseline Şekil 4.53'te yer verilmiştir.



Şekil 4.53 : Firebase platformuna veri gönderme ve alma işlemi.

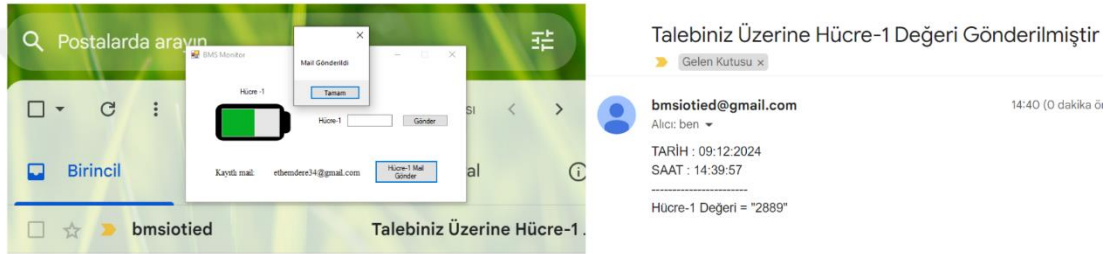
Bu veri gönderme işlemlerinin de tamamlanmasının ardından temel iki özellik gerçekleştirilmiş oldu. Bu sistemin aynı zamanda bir uyarı yapısı oluşturulması da planlanmıştı. Yani herhangi bir istenmeyen durum için kullanıcı mail adresine bilgilendirmelerde bulunarak gözden kaçabilecek hataların önün geçme planlandı. Bu bağlamda sistemin mail gönderebilme yetenekleri eklenmesi için çalışmalara başlandı.

Bu işlemler için öncelikle adres adında bir değişken oluşturuldu bu hem sınıf yapısına hem de Firebase platformuna eklendi. Aynı işlemler tekrar cell_1 de olduğu gibi program için eklendi. Şekil 4.54'te bu eklentilerin bir görseline yer verilmiştir.



Şekil 4.54 : Mail gönderme için yazılım yapılarının oluşturulması.

Ana oluşturulması gereken yapıların oluşturulması ardından e-posta gönderme detay yazılımına başlandı. Burada ilk etapta cell_1 değişkenine atanmış olan hücre-1 değeri e-posta olarak gönderme işlemi yapılması planlandı. Tasarıma eklenti olarak öncelikle hangi e-posta adresinin kullanıldığını gösterebilmek amacıyla Firebase tarafından adres değeri öncelikle okundu ve bir metne dönüştürülerek arayüze eklentisi gerçekleştirildi. Ayrıca bir e-posta gönderme butonu eklenmiştir, bu buton sayesinde hücre-1 değerinin o anki değerini gönderecek şekilde düzenlenmiştir. Burada ayrıca e-posta gönderildiğinde kullanıcıya e-posta gönderildi şeklinde bir bildirim oluşturacak şekilde yapı oluşturuldu. Şekil 4.55'te bu yapıya ve e-posta içeriğine yer verilmiştir. Burada ayrıca e-posta içeriğine gönderilen saat ve tarih eklendi.



Şekil 4.55 : Mail gönderme yapısı.

Yazılım yapısına kısaca bahsedilecektir ilerleyen zamanlardaki yapılan tüm mail gönderme işlemleri bu şekilde bir yapıyla gönderileceği bilinmelidir. Her eklenti için tekrar mail yapı yazılımı gösterilmeyecektir. Yazılım yapısını inceleyecek olursak Firebase Realtime Database ile bağlantı kurularak veriler alınmış ve bu veriler button9_Click fonksiyonu ile kullanıcı bir butona tıkladığında bir e-posta yoluyla iletilmektedir. Bağlantı başarılı olduğunda veri tabanından alınan veriler Class1 türünde bir nesneye dönüştürülür ve veriler arasından e-posta adresi (adres) ve Hücre1 Değeri (cell_1) gibi bilgiler çekilir.

E-posta içeriği sensör verileri ve sistemin çalıştığı tarihle saat bilgilerini içerecek şekilde düzenlenmiştir. Gönderici adresi alıcı adresi ve e-posta konusu tanımlandıktan sonra e-posta içeriği "MailMessage" nesnesi ile oluşturulmuştur. Basit Posta Aktarım Protokolü (SMTP) istemcisi Gmail sunucusu kullanılarak yapılandırılmış ve güvenli bağlantı için güvenli giriş katmanı (SSL) desteği etkinleştirilmiştir. Kullanıcı kimlik doğrulaması tamamlandıktan sonra e-posta gönderim işlemi gerçekleştirilmiş ve "Mail Gönderildi" mesajı iletilmiştir. Bu sayede sistem Firebase'den alınan verileri kullanarak e-posta gönderimi sağlamaktadır. Bu yazılım yapılarına Şekil 4.56'da yer verilmiştir.

```

FirebaseResponse response = await client.GetAsync("");
Class1 result = response.ResultAs<Class1>();

string parola = result.parola;
string adres = result.adres;
string mesaj = "Sisteme giriş yapmak için şifreniz = ";

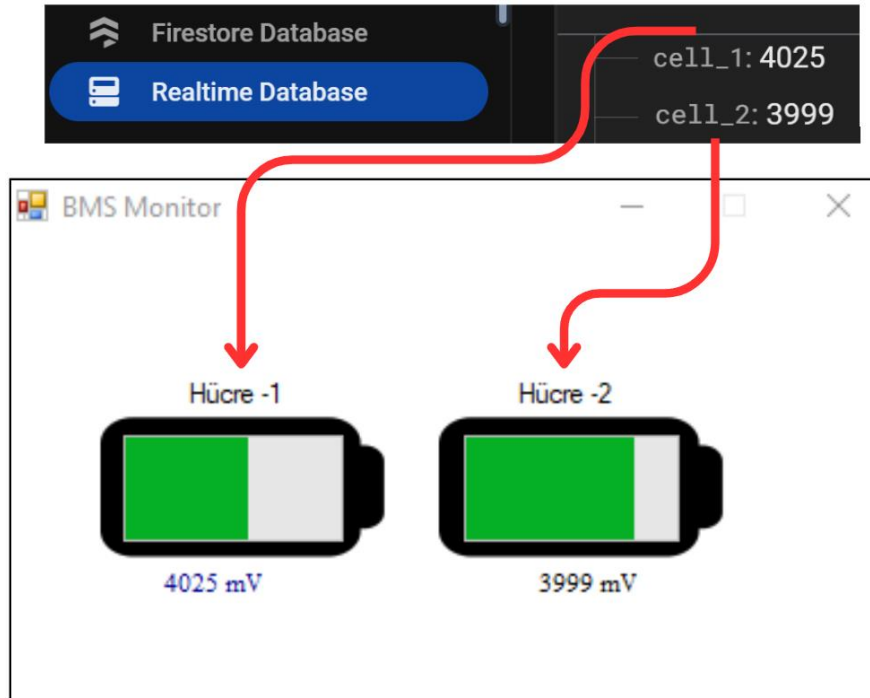
MailMessage ePosta = new MailMessage();
ePosta.From = new MailAddress("bmsiotied@gmail.com");
ePosta.Subject = ("Talebiniz Üzerine Şifreniz Gönderilmiştir");
ePosta.To.Add(adres);
ePosta.Body = mesaj + parola;
SmtpClient smtp = new SmtpClient();
smtp.Credentials = new System.Net.NetworkCredential("bmsiotied@gmail.com", "pszu thap swpc gqfa");
smtp.Host = "smtp.gmail.com";
smtp.EnableSsl = true;
smtp.Port = 587;
smtp.Send(ePosta);
MessageBox.Show("En son kayıtlı mail adresinize şifreniz gönderildi");

```

Şekil 4.56 : Mail gönderme yazılım temel yapısı.

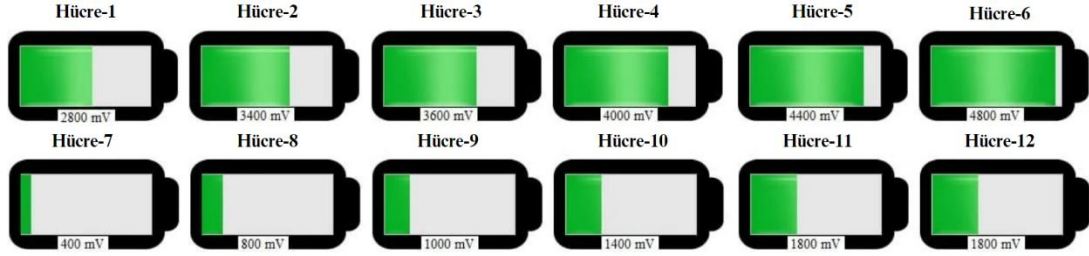
Bu yapıların tümü başarıyla sağlandıktan sonra artık sistem veri tabanından verileri okuyabilir, veri tabanına verileri kaydedebilir ve e-posta gönderebilir şekilde istenen yapılardaki yeteneklere sahip olmuştur. Artık tüm bu sistemin her veri için uygulama aşamasına geçilmiş oldu. Burada öncelikle batarya verilerini tamamlamak adına hücre-2 değeri verisi için de aynı işlemler uygulanmıştır.

Bu işlemler yapılırken ayrıca kullanıcı batarya verilerini daha net anlayabilmesi adına veri tabanından okunan mV değerleri hücre görsellerinin altına oluşturuldu. Bu yapının görseline Şekil 4.57’de yer verilmiştir.



Şekil 4.57 : İki Hücre veri okuma yapısı.

Bu işlemlerin ardından bu iki hücre için yapılan tüm işlemlerin hepsi diğer tüm batarya hücreleri için de gerçekleştirildi. Veri tabanından batarya verileri rastgele değerler girilerek yapının doğruluğu sağlandı. Bu yapı görseli Şekil 4.58’de incelenebilir.



Şekil 4.58 : Tüm batarya hücrelerinin eklenmiş hali.

Şekil 4.58’de görüleceği üzere planlanan asıl işlemlerden birisi daha tamamlanmış oldu. Tüm bu hücreler için aynı yazılım yapıları kuruldu ve verilerin okunması sağlanmış oldu. Bu işlem sonrasında batarya verilerinde önemli olan birkaç özellik daha eklendi. Örneğin bu hücreler arası gerilim farklılıklarının fazla olmaması batarya yönetim sistemlerinin temel görevlerinden biridir.

Bu hücrelerden en büyük mV değerine sahip batarya hücresiyle en küçük batarya hücre mV değeri arasındaki fark ifadesi bu sistem için önemli bir veridir. Bu verinin oluşturulması için yazılım yapıları kuruldu ve ayrıca arayüz programında gösterilebilecek şekilde yapılandırıldı. Bu yapının kurulumuna Şekil 4.59’da yer verilmiştir. Bu iki hücre arasındaki fark mV değeri delta mV değeri olarak adlandırıldı.

```
int[] cells = { cc1, cc2, cc3, cc4, cc5, cc6, cc7, cc8,
               cc9, cc10, cc11, cc12, cc13, cc14, cc15, cc16 };
int max = cells[0];
int min = cells[0];

foreach (int value in cells)
{
    if (value > max)
    {
        max = value;
    }

    if (value < min)
    {
        min = value;
    }
}
```

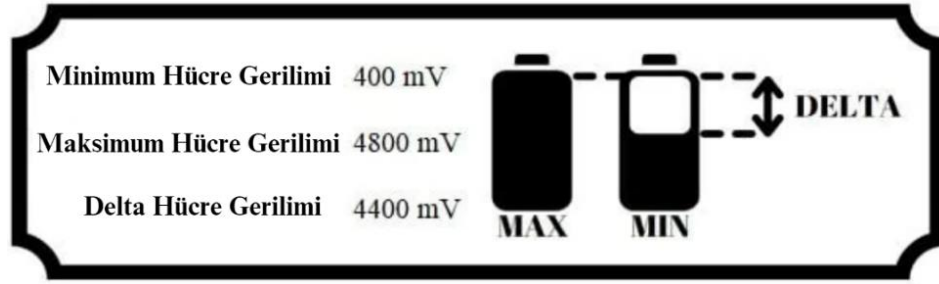
Minimum Hücre Gerilimi 400 mV

Maksimum Hücre Gerilimi 4800 mV

Delta Hücre Gerilimi 4400 mV

Şekil 4.59 : Hücre fark değeri hesaplama.

Bu işlemin kullanıcılar tarafından daha anlaşılır olabilmesi için batarya sembolleriyle bu fark değerini ifadesi görselleştirildi. Bu görselleştirilmiş yapı Şekil 4.60’ta gösterilmiştir. Bugörselleştirilen yapı sayesinde kullanıcı doğrudan verilerle baktığında verileri ne anlama geldiğini anlayabilmesi hedeflenmiştir.



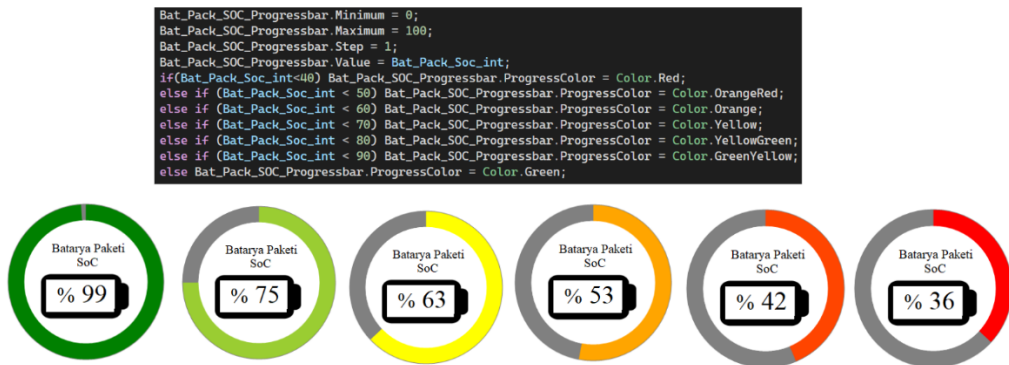
Şekil 4.60 : Görselleştirilmiş fark mV değerleri.

Batarya yönetim sistemi için önemli olan diğer iki husus şarj durumu (SoC) ve sıcaklık durumudur. Bu iki yapının da bu arayüzde olması çok önemli olan veri çeşitlerindendir. Bu yapılarda aynı batarya hücre sistemleri için uygulanan yöntemlerde olduğu gibi veri değerleri veri tabanından okunarak işleme alındı. Bu yapının görseline Şekil 4.61’de yer verilmiştir.



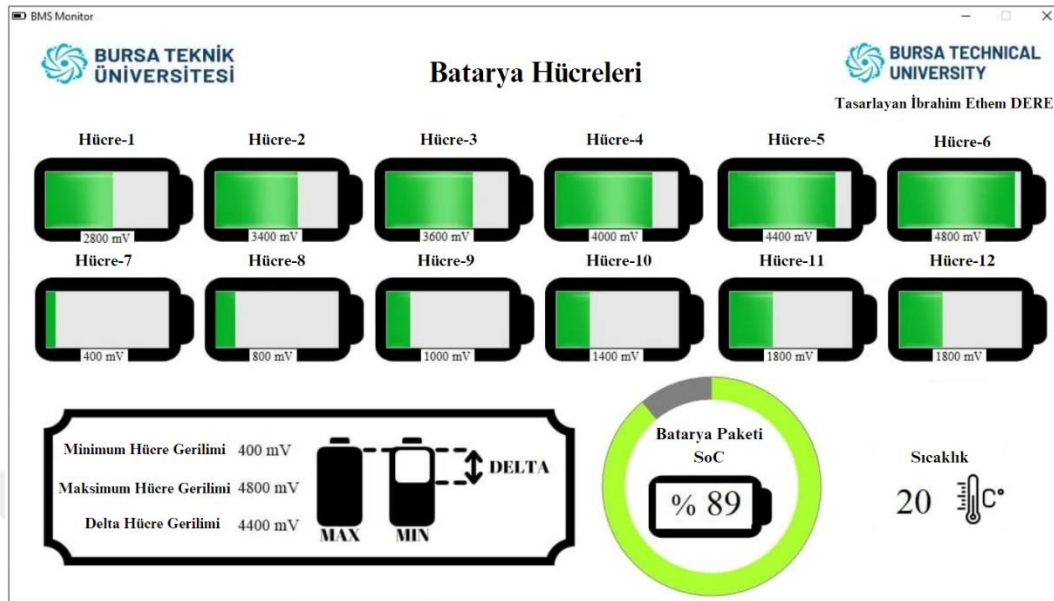
Şekil 4.61 : Batarya SoC ve sıcaklık değer gösterimi.

Bu yapıların oluşturulmasında ayrıca SoC değerinin değişmesiyle görsel olarak renk değiştirmesi durumu da eklenmiştir. Bu yapı görseline kod bloklarına Şekil 4.62’de yer verilmiştir. Buradaki mantık aslında batarya SoC değerinin belli aralıklara geldiğinde farklı renklere dönüşmesidir.



Şekil 4.62 : Görsel geliştirmeler yapılmış SoC değeri.

Bu yapılarında kurulmasının ardından tüm sistem birleştirildi ve yeni bir arayüz ekranı oluşturulmuş oldu bu arayüze Şekil 4.63'te yer verilmiştir.



Şekil 4.63 : İşlemlerin birleştirilmiş yapısı görseli.

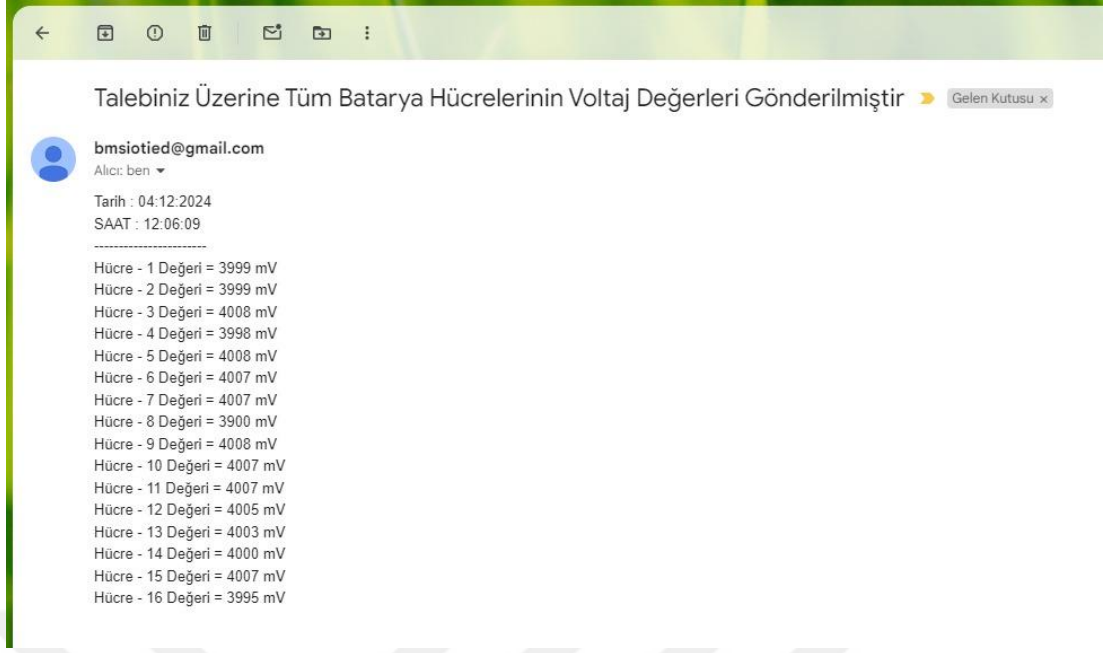
Bu oluşturulan yapı iki farklı bilgisayar üzerinden çalıştırıldı ve sistemin birden fazla bilgisayar üzerinde de çalışabilirliği gözlemlenmiş oldu.

Bu yapı sonrasında kullanıcının istediği anda tüm batarya verilerine ulaşabilmesi için dört farklı buton yapısı eklendi. Bu buton yapıları sayesinde kullanıcı istediği zaman tüm batarya hücre verilerine, maksimum-minimum-delta hücre verilerini ve batarya SoC-sıcaklık verini sisteme kayıtlı e-posta adresine gönderebilecek şekilde özgür bırakılmıştır. Ayrıca parametre ayarları menüsü eklenmiştir kullanıcı bu butona tıkladığında yeni bir pencere açılacak ve burada ayarlanan tüm batarya parametrelerini değiştirebilir şekilde yeteneklere sahip olacaktır. Bu buton yapılarına Şekil 4.64'te yer verilmiştir.



Şekil 4.64 : Mail ve parametre ayarlar butonları görüntüsü.

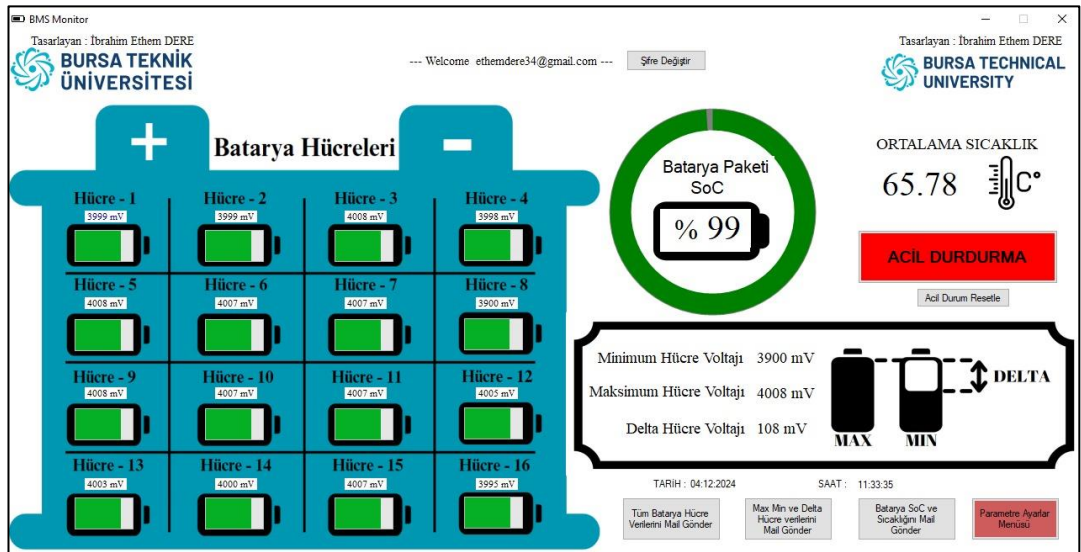
Örnek olarak sadece bir e-posta gönderme butonun çalışma görseli gösterilecektir diğerleri de aynı mantık da çalışmaktadır. Bu yapılardan batarya hücre verileri gönderme butonuna tıkladığında kayıtlı e-posta adresine doğrudan verilerin iletildiği gözlemlenmiştir. Bu e-posta iletisinin görseline Şekil 4.65'te yer verilmiştir.



Şekil 4.65 : Buton aracılığı ile batarya verilerini gönderme işlemi.

Ayrıca ana ekrana acil durdurma ve bu acil durdurma işleminin resetlenmesi için acil durum resetleme butonu eklendi. Bu yapı aslında ESP32 başlığı altında anlatılan “stop” değişkeninin değerini değiştirerek, gerekli aksiyonların alınmasını sağlayacaktır. Bu ana sayfa yapısına ayrıca şifre değiştirme menüsü de eklenmiştir ve bazı görselleştirme çalışmaları yapılmıştır.

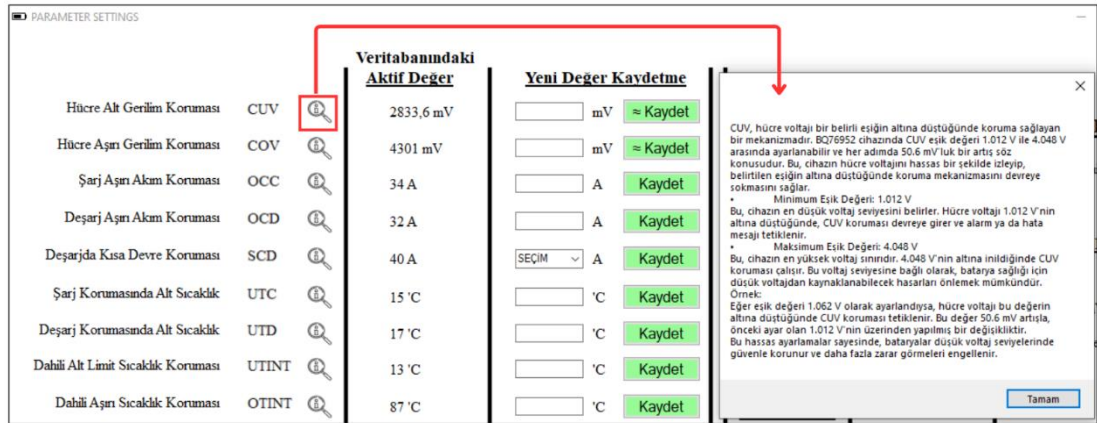
Bu çalışmalara Şekil 4.66’da yer verilmiştir. Burada tüm yapılan işlemler daha önce yapılan işlemlere benzer şekilde gerçekleştirilmiştir. Her detay konu bütünlüğünü bozmamak adına gösterilmemiştir.



Şekil 4.66 : Görsel geliştirmeler yapılmış ana ekran.

Daha sonra parametre ayarlar menüsü oluşturulmaya başlanmıştır. Burada yine aynı şekilde tüm yapılan işlemler daha önce yapılan işlemlere benzer şekilde gerçekleştirilmiştir. Her detay konu bütünlüğünü bozmamak adına gösterilmemiştir.

Öncelikle batarya yönetim sistemi tasarımı başlığı altında anlatılan tüm koruma parametreleri metinsel şekilde eklendi. Ardından bu yapılar daha önce oluşturulmuş olan veritabanındaki tüm koruma parametrelili bir metin yazısına ataması yapılarak kayıtlı parametre ayarları okuması ve sistemde önceki ayarın ne olduğunun gösterilmesi işlemleri yapıldı. Bu yapılara ayrıca yeni değer girme kısmı eklenerek kullanıcının bu parametre ayarlarını değiştireceği değeri gireceği alanların oluşturulma işlemi gerçekleştirildi. Burada ayrıca bilgi baloncuğu eklenmiştir ve buna tıklandığında o parametrenin ne iş yaptığıyla alakalı bilgi veren bir metin açma işlemi gerçekleştirildi. Bu bilgilendirme mesaj ekranı her parametre ayarı için eklenmiştir. Bu yapı örneğine ve CUV bilgilendirme mesaj ekranına Şekil 4.67’de yer verilmiştir.



Şekil 4.67 : Parametre ayar ve bilgilendirme görseli.

Burada parametre ayarları değiştirme işlemleri için örnek bir yazılım yapısı anlatılacaktır, diğer tüm parametre ayarları da aynı işlemleri içerecek şekilde eklentileri yapılmıştır. Örneğin CUV daha önceki başlıklar altında anlatıldığı üzere belli adım sayılarına sahiptir ve her değer dorudan işlenememektedir bu sebeple kullanıcının yazdığı değeri bir dizi hesaplama işleminden geçirildikten sonra Firebase platformuna gönderecek şekilde tasarlanmıştır. Buradaki kod yapılarına Şekil 4.68’de örnek gösterilmiştir. Bu verilerde hem okuma yaptıktan sonra mV cinsine dönüştürmek için hem de kullanıcının girdiği veriyi tekrar kontrolcünün içeriğine uygun olacak şekilde hesaplama işlemlerinden sonra veri tabanına gönderilmektedir.

Mesela CUV yapısı tekrar hatırlanacak olursa CUV eşik değeri 1.012 V ile 4.048 V arasında ayarlanabilir ve her adımda 50.6 mV'luk bir artış söz konusudur. Bu da demek olur ki örneğin işleme alınacak değer 35 olduğunda 1171 mV olarak ayarlanmış olmaktadır. Bu sebeple kullanıcının istediği değere yakın bir değere göre bu adım oranına göre yuvarlanacağı için yaklaşık eşit sembolü eklenmiştir. Ayrıca bu işlemler de yazılım içerisine eklenmiştir. Bu yazılım yapılarına Şekil 4.68'de yer verilmiştir.

Veri Okuma sonrası hesaplama

```

int PCUV_int = Convert.ToInt32(PCUV);
double PCUV_double = Convert.ToDouble(PCUV_int);
PCUV_double = PCUV_double * (50.6);
PCUV = Convert.ToString(PCUV_double);

//*****

int PCOV_int = Convert.ToInt32(PCOV);
double PCOV_double = Convert.ToDouble(PCOV_int);
PCOV_double = PCOV_double * (50.6);
PCOV = Convert.ToString(PCOV_double);

```

Veri gönderme öncesi hesaplama

```

//yayınla
private void button1_Click(object sender, EventArgs e)
{
    int PCUV_int = Convert.ToInt32(CUV_textBox.Text);
    double PCUV_double = Convert.ToDouble(PCUV_int);
    PCUV_double = PCUV_double / (50.6);
    PCUV_int = Convert.ToInt32(PCUV_double);
    string PCUV = Convert.ToString(PCUV_int);

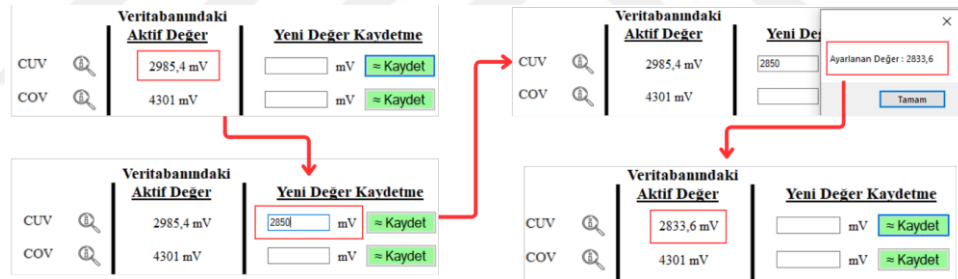
    PCUV_double = Convert.ToDouble(PCUV_int);
    PCUV_double = PCUV_double * (50.6);

    System.Windows.MessageBox.Show("Ayarlanan Değer : " + PCUV_double);
    //string PCUV = CUV_textBox.Text?.Trim('\n') ?? string.Empty;
    WriteToFirebase(client, "CUV", $"{PCUV}");
    CUV_textBox.Text = "";
}

```

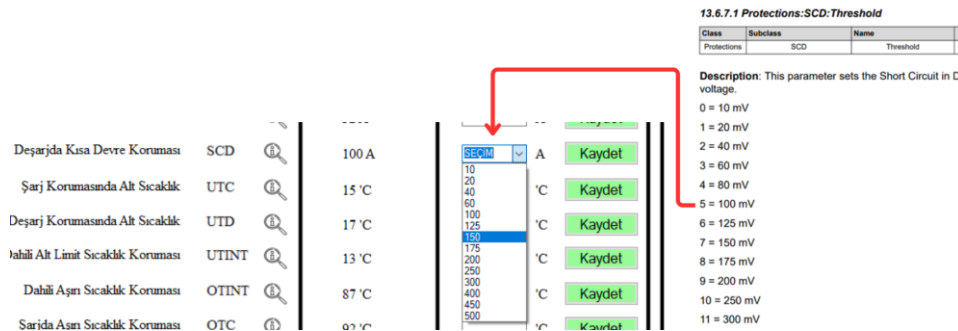
Şekil 4.68 : Hesaplamalar ile parametre değeri belirleme.

Buradaki yapı daha net anlaşılması için deneysel çalışma içerisinde her aşamanın nasıl değiştiğini göstere yapıya Şekil 4.69'da yer verilmiştir.



Şekil 4.69 : Parametre ayarlarının kayıt hesaplamalar ile kayıt işlemi.

Bir başka örnek olarak da deşarjda kısa devre koruması (SCD) örnek verilebilir. Nu yapıda sadece belirli seçenekler kullanılabildiğinden burada seçim menüsü eklenerek işlem yapılmaktadır. Bu örnek yapıya Şekil 4.70'te yer verilmiştir.

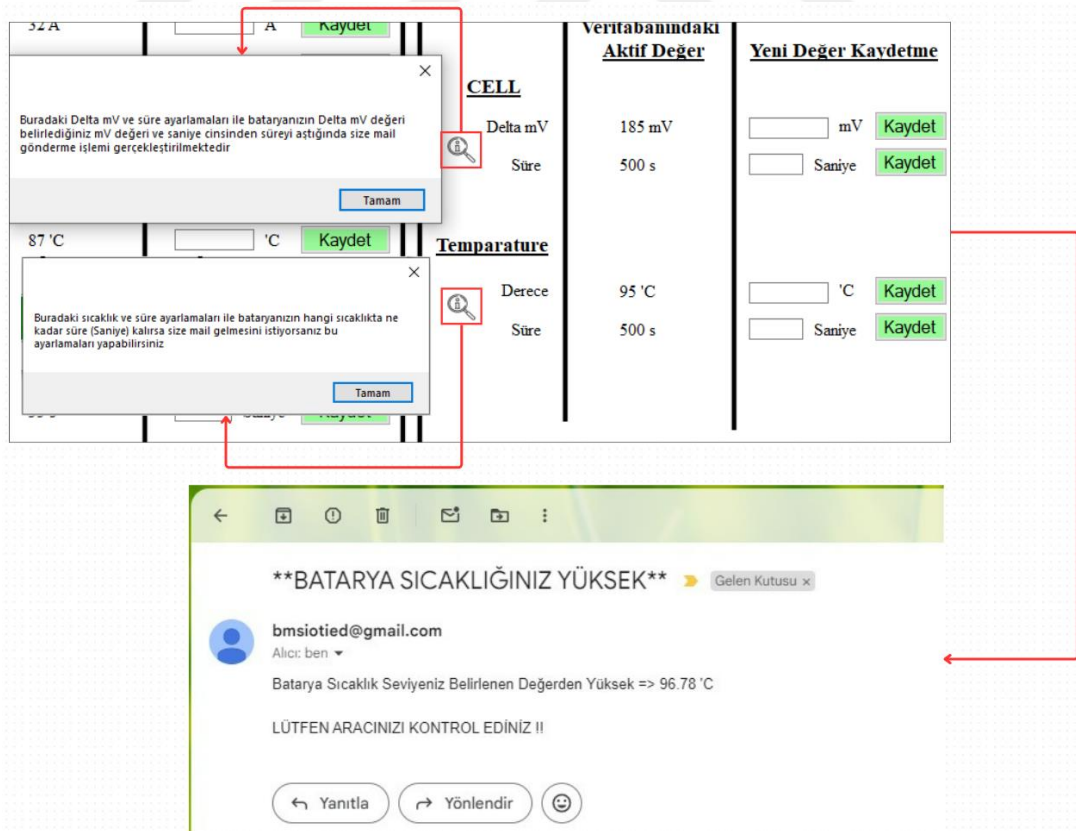


Şekil 4.70 : SCD koruması için seçim yapılabilen bölüm.

Bu işlemlerin ardından batarya yönetim sisteminin otomatik e-posta gönderme yapıları oluşturuldu. Bu konu batarya yönetim sistemi için önemli olan konulardan birisidir. Çünkü IoT ile çalışan bu sistemin dışardan parametre kontrol ayarları yapılabildiği gibi bazı uyarı verilmesi gereken durumlarda da bilgilendirilmesi gerekmektedir.

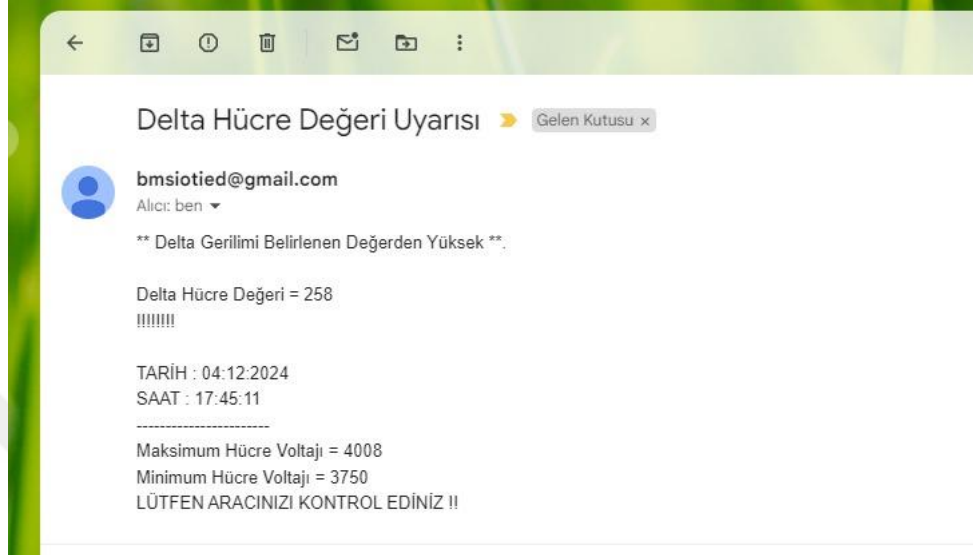
Bu durumlar göz önünde bulundurularak otomatik bir e-posta gönderme yapısı oluşturulmuştur. Bu yapı için iki adet parametre eklenmiştir. Bunlardan biri en önemli durumlardan olan sıcaklık değeridir ve diğeri de batarya maksimum ve minimum hücreleri arasındaki fark olan delta değeridir. Bu yapıda bir süre ve bir değer ataması yapılan bir sistem kurulmuştur. Bu yapıda yine bilgilendirme kutucukları eklenmiştir.

Buradaki işlemde örneğin belirlenen sıcaklık ve süre sonrasında sistem belirlenen değerin üzerinde belirlenen süreden fazla kalırsa bilgilendirme e-postası göndermektedir. Örneğin 95 derece ve 500 saniye belirlenmiş bir sistemde batarya sıcaklığı 500 saniye boyunca 95 derece üzerinden seyrederse otomatik olarak uyarı e-postası gönderilmektedir. Bu yapıya ve e-posta içeriğinin olduğu görsele Şekil 4.71’de yer verilmiştir.



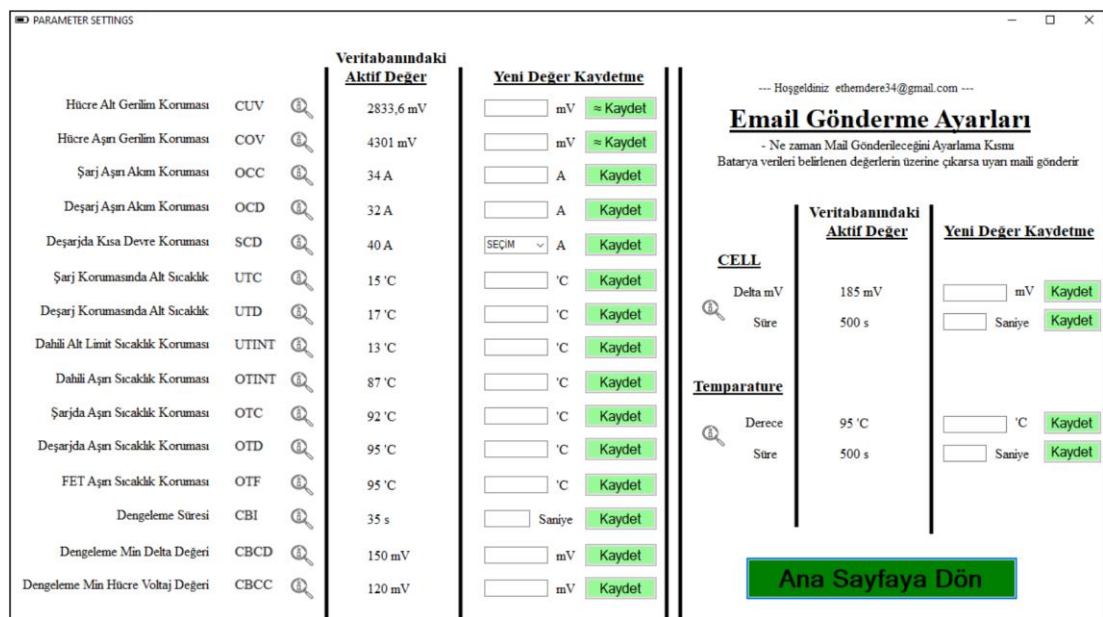
Şekil 4.71 : Otomatik sıcaklık uyarısı e-posta gönderimi.

Bu işlemlerin aynısı delta hücre gerilimi için de uygulanmıştır. Belirlenen fark değeri ve sürenin üzerinde hücre dengesizliği fazla olursa e-posta göndererek kullanıcıya bir hata olabileceği şeklinde bir uyarı e-postası gönderilmektedir. Bu yapının da görseline Şekil 4.72’de yer verilmiştir.



Şekil 4.72 : Hücre gerilim farkı çok olursa otomatik e-posta gönderim işlemi.

Bu yapılan tüm işlemler yani parametre ayarlar menüsü içerisindeki tüm koruma ayarları ve e-posta gönderme ayarları tek bir yapı içerisinde birleştirildi. Yani artık kullanıcı parametre ayarlar kısmına girdiğinde anlatılan tüm işlemleri gerçekleştirebilecek şekilde bir yeteneğe sahip olacaktır. Bu yapı görseline Şekil 4.73’de yer verilmiştir.



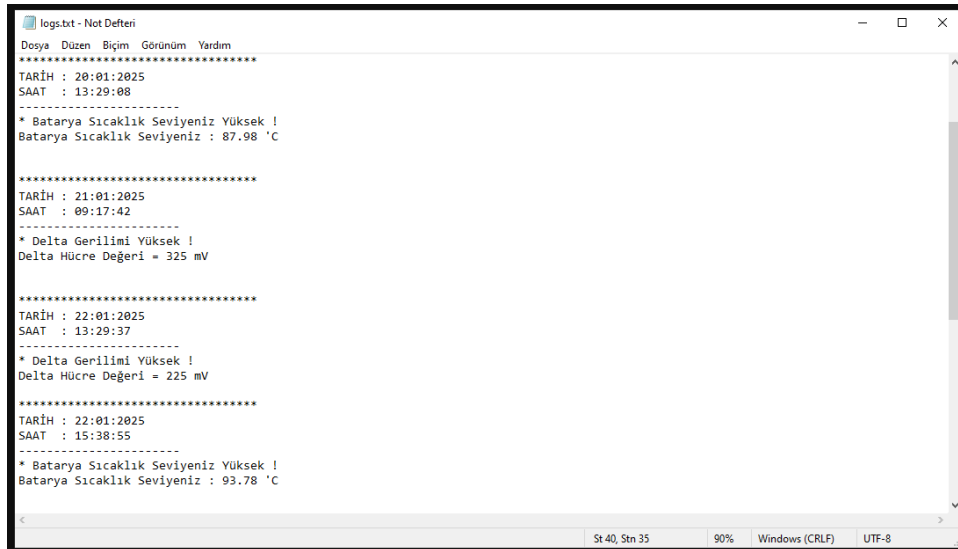
Şekil 4.73 : Parametre ayarlar menüsü tam görünümü.

Bu uygulamada ek olarak meydana gelen olayların ve hata mesajlarının kayıt altına alınmasını sağlamak amacıyla bir log yazma mekanizması geliştirildi. “WriteLog” isimli metod bir log mesajını alarak uygulamanın çalıştığı dizinde logs.txt adlı bir dosyaya yazmaktadır. Bu yapının yazılımsal kod yapısının bir kısmına Şekil 4.74’te yer verilmiştir.

```
2 başvuru
public void WriteLog(string logMessage)
{
    try
    {
        string filePath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "logs.txt");
        using (StreamWriter writer = new StreamWriter(filePath, true))
        {
            writer.WriteLine();
            writer.WriteLine(logMessage);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Hata: {ex.Message}");
    }
}
```

Şekil 4.74 : Toplu hata kayıt kod yapısı bir kısmı

Log dosyasının yol bilgisi uygulamanın çalıştığı dizini veren klasör olacak şekilde AppDomain.CurrentDomain.BaseDirectory kullanılarak oluşturuldu. Ayrıca hata yönetimi için bir “try-catch” bloğu kullanılmış ve yazma işlemi sırasında bir hata meydana geldiğinde kullanıcı bilgilendirilecek şekilde mekanizma kuruldu. Bu yöntem ile uygulamanın hata izleme ve olay kaydı süreçlerinde etkili bir çözüm sunarak sistemin güvenilirliğini artırmak hedeflendi. Yani kullanıcı hangi günlerde, saatlerde ve ne sıklıkla hata mesajlarını aldığını görebilecek bir yapı oluşturuldu. Bu yapının çıktı görseline Şekil 4.75’te yer verilmiştir.



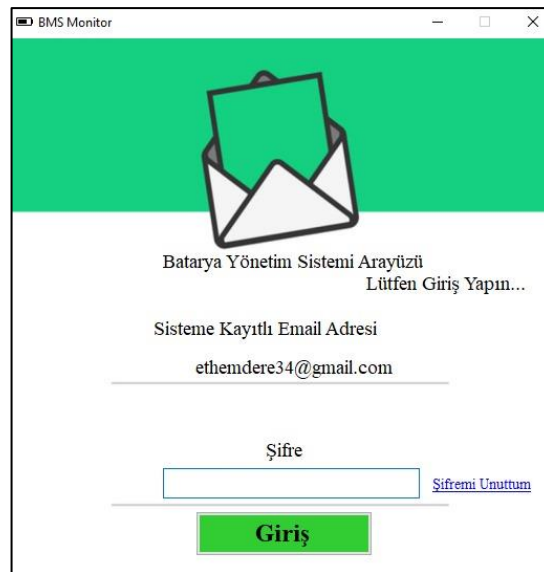
Şekil 4.75 : Toplu hata kayıt çıktı örneği

Bu sistemin artık e-posta gönderme veri okuyup veri yazma işlemlerinden sonra bu yapıların güvenli bir şekilde kullanılabilmesi için program çalıştığında ilk etapta bir şifre istemesi ve bu şifre ile programa giriş yapması mümkün olması için bir yapı oluşturuldu. Burada eğer şifre doğruysa sisteme giriş yapılması eğer yanlışsa kullanıcıya şifrenin yanlış olduğunu gösteren bir yapı oluşturuldu. Ayrıca kullanıcı şifreyi unutma durumu olursa şifremi unuttum bağlantı yazısı eklendi ve bu sayede bu kısma tıklandığında en son kayıtlı olan e-posta adresine şifre gönderilmiştir. Tüm sistemin yazılım yapıları tez konu bütünlüğü bozulmamak amacıyla eklenmemiştir. Temel yapıdaki oluşturulan bu sisteme Şekil 4.76’da yer verilmiştir.



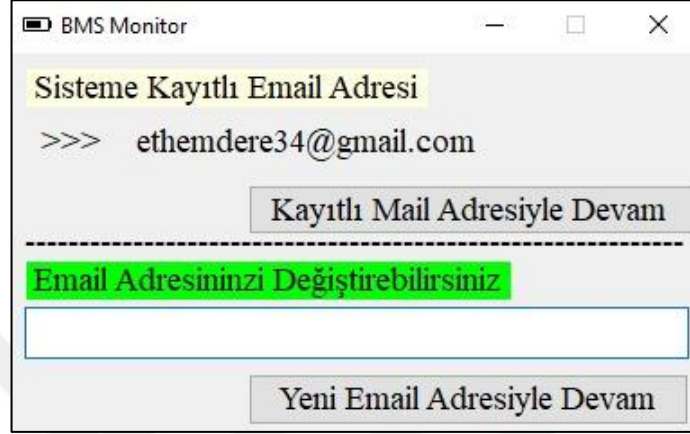
Şekil 4.76 : Şifre ile sisteme giriş yapısının temel görseli.

Bu yapı sonrasında temel yapılmak istenen gerçekleştirildikten sonra bu yapıda en son kayıtlı e-posta adresini gösteren ve daha çok göze hitap eden kullanıcı dostu düzenli bir yapı oluşturulmuştur. Bu yapı Şekil 4.77’den incelenebilir.



Şekil 4.77 : Görselleştirilmiş giriş sayfası.

Kullanıcının daha sonra e-posta adresini değiştirebilme ihtiyacı göz önünde bulundurularak sisteme şifre ile giriş yapılmasının ardından yeni açılan bir pencere yapısı oluşturuldu. Bu yapıda kullanıcıya eski e-posta ile devam etme veya e-posta değişikliği yapma gibi imkanlar sunulmuştur. Yazılım yapıları tamamen buna göre oluşturulmuştur. Bu yapılması istenen yapı görseline Şekil 4.78’de yer verilmiştir.



Şekil 4.78 : Hangi e-posta ile devam edileceğinin seçim menüsü.

4.5 Telefon Yazılımı ve Görsel Tasarım Deneysel Çalışma

Telefon yazılımı ayarlanırken sadece izleme işlemlerine opsiyon olarak eklenmiştir. Buradaki uygulama tasarımında materyal yöntemlerde bahsedildiği gibi MIT APP platformu kullanılmıştır. Mobil uygulamanın geliştirilme sürecindeki temel adımları kullanılan araçlar ve entegrasyonları adım adım gerçekleştirilmiştir. Çalışmada MIT İlk olarak, MIT App Inventor platformuna erişilerek yeni bir proje oluşturulmuştur. Platformun sunduğu görsel programlama ortamı sayesinde kullanıcı arayüzü bileşenleri kolaylıkla eklenmiş ve düzenlenmiştir. Tasarım aşamasında uygulamanın temel işlevlerini yerine getirecek ekranlar ve bileşenler belirlenmiştir.

Uygulamaya gerçek zamanlı veri yönetimi ve depolama yetenekleri kazandırmak amacıyla Firebase veri tabanı entegre edilmiştir. Firebase konsolunda yeni bir proje oluşturularak uygulamaya ait verilerin depolanacağı veri tabanı bilgileri daha önce Firebase platformundan alınan bilgilerle eklentiler yapılmıştır. Yani MIT App Inventor üzerindeki FirebaseDB bileşeni kullanılarak uygulama ile veri tabanı arasında bağlantı kurulmuştur. Bu sayede uygulama üzerinden daha önceden ESP32’nin veri tabanına gönderdiği veriler örneğin batarya verileri Firebase veri tabanından uygulama üzerinden görüntülenme işlemi planlaması yapılmıştır.

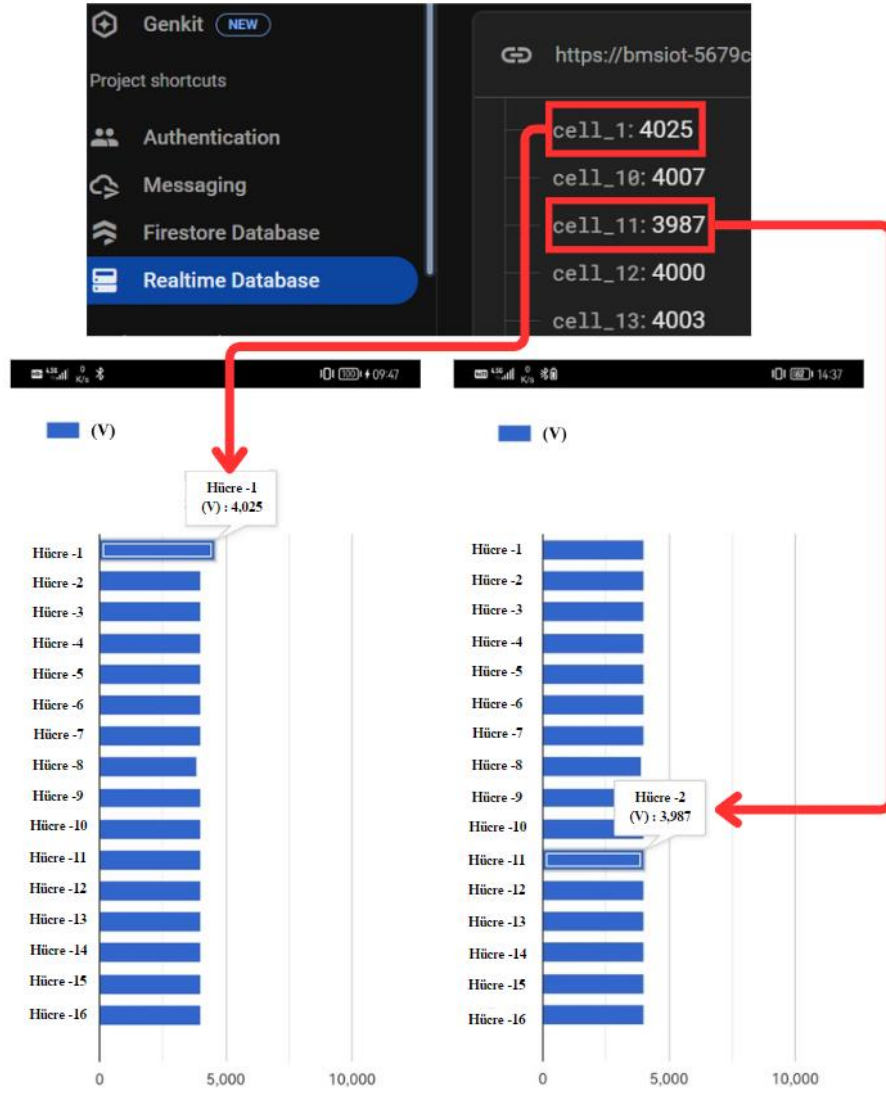
Firestore veri tabanından alınan veriler, uygulama üzerinde görsel olarak sunulmak üzere işlenmiştir. Bu işlemde, MIT App Inventor'ın sağladığı grafik ve çizim bileşenleri kullanılmıştır. Elde edilen veriler kullanıcı dostu grafikler ve tablolar aracılığıyla kullanıcının mobil uygulamayı kullanarak batarya verilerinin izlenebilir mekanizması sunulmuştur. Tüm bu işlemlerin adım adım gerçekleştirilmesi Şekil 4.79'da gösterilmiştir.



Şekil 4.79 : MIT APP programı oluşturma aşamaları.

Bu Şekil 4.79'da yer verilen adım adım uygulama işlemlerini her detayına tezi konu bütünlüğünü bozmamak adına özet bilgiler verilerek anlatılmıştır. Tüm bu adım adım uygulanan işlemlerin ardından mobil uygulama tüm batarya verilerini gerekli yazılımsal işlemleri gerçekleştirdikten sonra tüm batarya verilerini bir grafikte gösterecek şekilde tasarım yapılmıştır. Bu grafikte öğrenilmek istenen hücre değerine

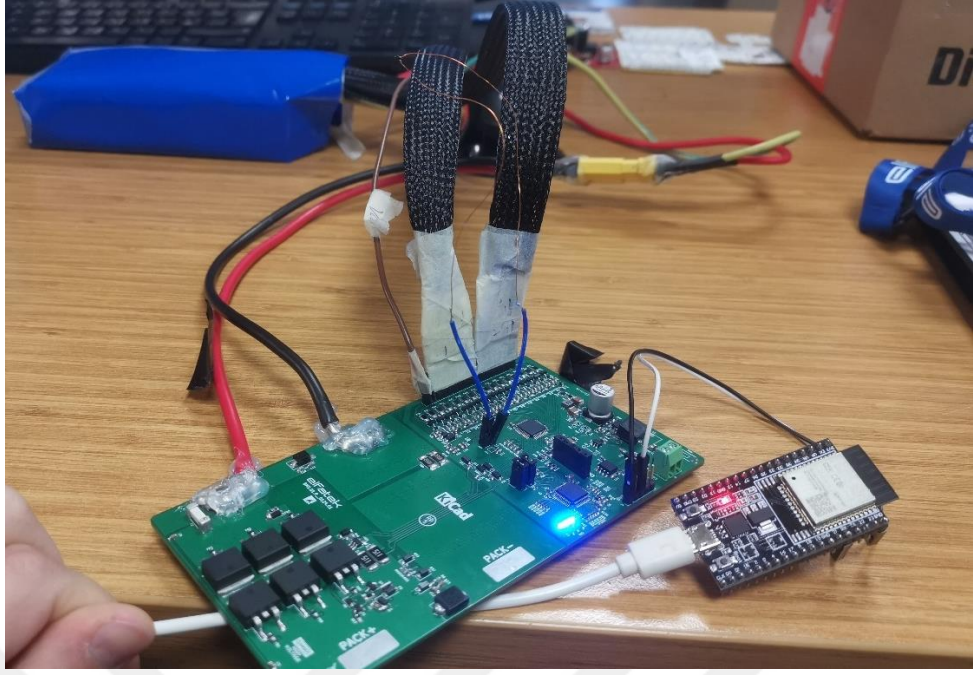
tıklandığında bataryanın o hücre değerini Firebase platformunda kaydedildiği gibi mV cinsinden gösterim yapısı eklenmiştir. Bu yapının örnek bir görseline Şekil 4.80’de yer verilmiştir. Şekil 4.80’de veritabanındaki veriler mobil uygulama ekran görüntüsünde görüleceği üzere kullanıcıya bilgi verme işlemini gerçekleştirmektedir.



Şekil 4.80 : Mobil uygulama ile batarya verilerinin takibi.

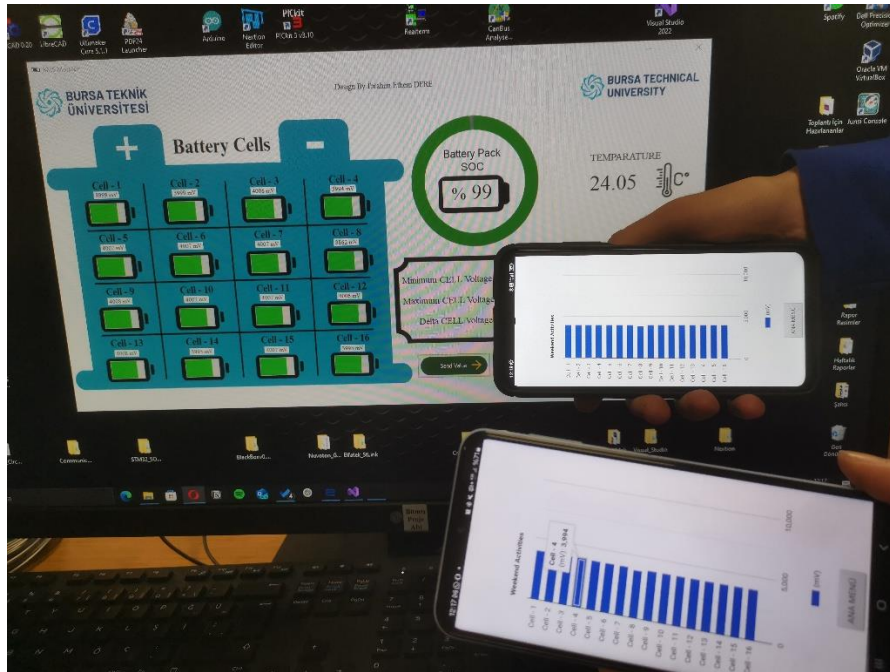
4.6 Tüm İşlemlerin Uyum İçerisinde Çalışması Deneyisel Çalışma

Yapılması planlanan tüm işlemlerin yapılmasının ardından artık sistemin genel yapısının toplu şekilde çalıştırarak deneysel çalışma sonuçlandırma hedeflenmiştir. Burada öncelikle batarya yönetim sistemine batarya bağlanmış ve ardından ESP32 ile bağlantısı gerçekleştirilmiştir. Daha sonra ESP32’nin internete bağlanarak veri alışverişini sağlanmıştır. Bu bağlantı görseline Şekil 4.81’de yer verilmiştir.

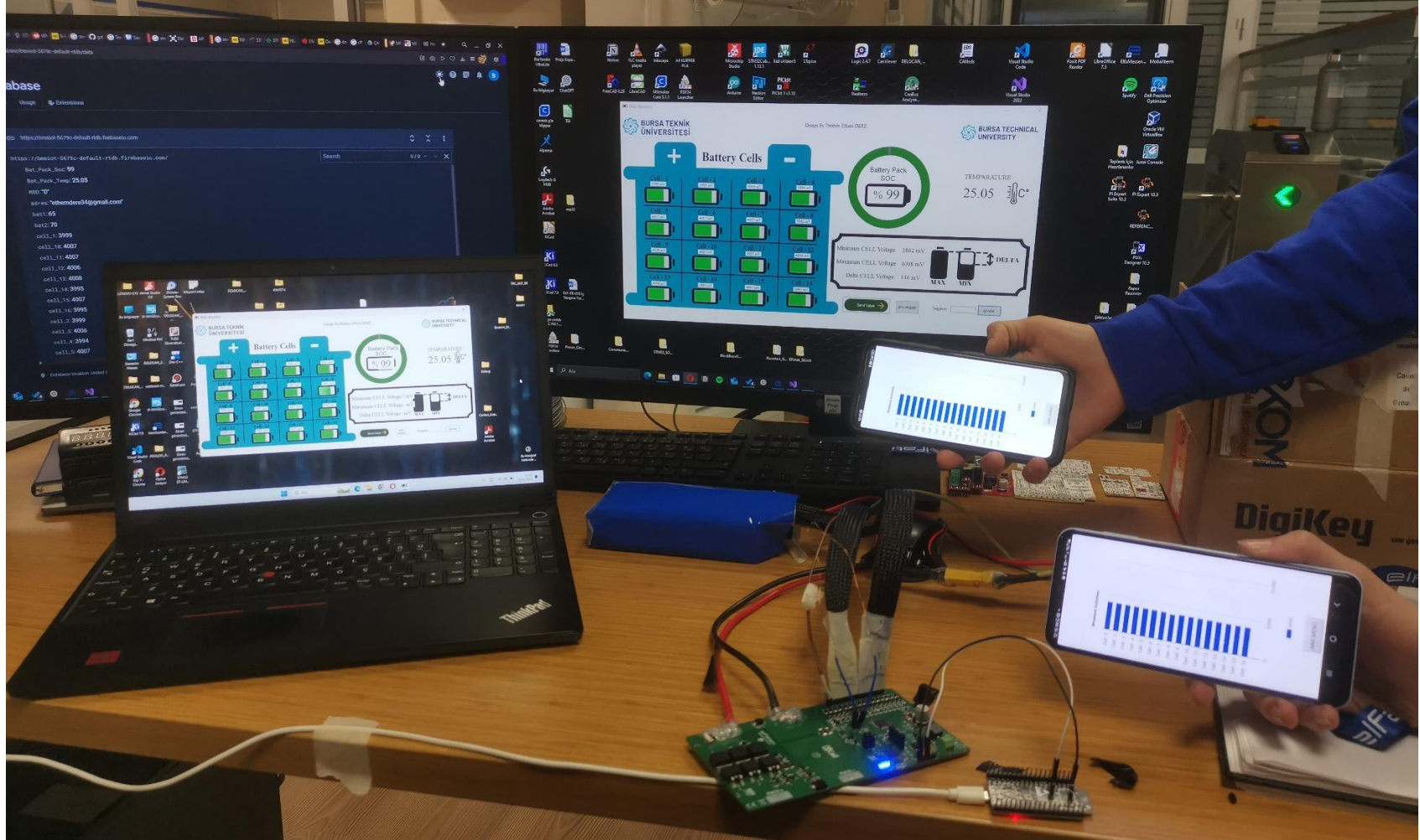


Şekil 4.81 : BMS, batarya ve ESP32 bağlantısı.

Burada ilk etapta batarya verileri masa başında iki telefon ve bir bilgisayardan izlenmiştir, bu görsel Şekil 4.82’den incelenebilir. Ardından iki farklı bilgisayar ve iki farklı telefon üzerinden izlemeleri yapılmıştır. Bu oluşturulan sistem görsellerine Şekil 4.83’te yer verilmiştir. Ardından bu sistem ilerde AKİBA aracında kullanılabileceğini gösteren bir deneysel görsele de Şekil 4.84’te yer verilmiştir. Bu yapılar sonrasında tez hedeflediği işlemleri başarılı bir şekilde gerçekleştirmiştir.



Şekil 4.82 : Telefon ve bilgisayardan verileri izleme.



Şekil 4.83 : Deneysel çalışma son hali görseli.



Şekil 4.84 : AKİBA ve tüm sistem çalışma görseli.

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında otonom yük taşıyan bir robot için tasarlanan Batarya Yönetim Sistemi (BMS) devresi, yazılımı ve IoT teknolojisi temelinde batarya verilerinin etkin bir şekilde izlenmesi ele alınmıştır. Gerçekleştirilen sistem batarya durumunu izleme, koruma algoritmaları sayesinde güvenliği sağlama ve verilerin uzaktan erişimle gözlemlenebilmesine olanak tanıyan bir altyapı sunmuştur. Test sonuçları sistemin belirlenen hedefleri başarıyla karşıladığını ve endüstriyel uygulamalar için kullanılabilecek bir çözüm sunduğunu ortaya koymuştur.

Bu çalışmada geliştirilen BMS batarya gerilimi, akımı ve sıcaklık değerlerini hassas bir şekilde izleyebilmiş ve IoT entegrasyonu sayesinde bu verileri gerçek zamanlı olarak uzak sunuculara aktarmıştır. Sistem batarya güvenliğini optimize etmek amacıyla koruma algoritmalarıyla donatılmış böylece hem batarya ömrü uzatılmış hem de kullanım esnasındaki güvenilirlik artırılmıştır. Özellikle IoT tabanlı izleme sistemi kullanıcıların batarya durumunu anlık olarak takip etmelerini sağlayarak sistem performansını daha geniş bir ölçekte değerlendirme imkânı sunmuştur.

Yapılan testlerde tasarımın yüksek doğrulukta veri sağladığı ve beklenen performans kriterlerini karşıladığı görülmüştür. Ancak bu çalışmanın ileride daha geniş kapsamlı ve etkin bir hale getirilebilmesi adına bazı geliştirmeler yapılabilir. Tasarımın geliştirilmesi için donanım tarafında BQ76972 gibi daha hassas ölçüm yapabilen bir entegre kullanılması önerilebilir. Bu entegre mevcut sistemdeki ölçüm hassasiyetini daha da artırarak kritik batarya parametrelerinin daha doğru bir şekilde izlenmesine olanak tanıyacaktır. Ayrıca PCB tasarımının dört katlı bir yapı ve iki kat bakır kullanılarak daha kompakt hale getirilmesi, devrenin fiziksel boyutlarını küçültecek ve üretim sürecini optimize edecektir. Bu değişiklikler sistemin taşınabilirliğini ve endüstriyel uyumluluğunu artıracaktır.

Yazılım tarafında ise batarya ölçüm verilerinin uzun vadeli analizler için kaydedilebileceği bir veri kayıt sistemi eklenmesi önerilebilir.

Bu geçmiş verilere dayalı analizler yaparak batarya performansını değerlendirme ve sistem hatalarını tespit etme süreçlerini kolaylaştıracaktır. IoT iletişiminde güvenliği

artırmak için şifreleme ve kimlik doğrulama protokollerinin kullanılması da önemli bir adım olacaktır bu da sisteme eklenebilir. Bu özellikle sunucu veri aktarımı sırasında meydana gelebilecek güvenlik açıklarını en aza indirmek için faydalı olacaktır.

Sistemin farklı çevresel koşullarda (örneğin aşırı sıcaklık, titreşim gibi) daha geniş kapsamlı testlerden geçirilmesi güvenilirlik ve dayanıklılıkla ilgili ilave veri sağlayacaktır. Bu testler yoğunlaştırılabilir. Bunun yanı sıra tasarımın farklı batarya teknolojileriyle uyumlu hale getirilmesi endüstriyel uygulamalarda sistemin kullanım alanını genişletebilir. Ayrıca tasarlanan devrenin elektromanyetik dayanım testleri ile sistemin dış etken faktörlerden etkilenmesi minimuma indirgenecek şekilde tasarım ve planlama yapılabilir.

IoT tabanlı sistemin verimliliğini artırmak amacıyla sunucu tarafında daha gelişmiş analiz ve görüntüleme yazılımlarının geliştirilmesi önerilebilir. Bu sayede kullanıcıların sistemden daha fazla bilgi edinmesine ve verimliliklerini artırmasına olanak tanıyabilir.

Sonuç olarak, bu tez çalışması otonom yük taşıyan robotlar için etkili bir batarya yönetim sistemi sunmuştur. Çalışmanın sunulan öneriler doğrultusunda daha da geliştirilmesiyle performansının artırılması ve endüstriyel uygulamalarda daha geniş bir alanda kullanılması mümkün olabilir.

KAYNAKLAR

- [1] **S. Krishnakumar et al.**, (2022) “IoT-based Battery Management System for E-Vehicles,” in *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, Oct. 2022, pp. 428–434. doi: 10.1109/ICOSEC54921.2022.9952052.
- [2] **K. Kumari.K.S, L. Chitra, J. M. Abraham, N. Joseph, and Y. K. T.K**, (2023) “The Role of IOT & AI in Battery Management of Electric Vehicles,” in *2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, IEEE, May 2023, pp. 1–7. doi: 10.1109/IConSCEPT57958.2023.10170275.
- [3] **K. I. V. V, A. B. K, and L. ST**, (2024) “Smart Battery Management System for Electric Vehicles Using IoT,” in *2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, IEEE, Mar. 2024, pp. 720–725. doi: 10.1109/ICICV62344.2024.00120.
- [4] **L. S. B, A. Sangari, E. K, K. V, J. A. Sheeba, and D. Sivamani**, (2022) “IoT-Based Battery Monitoring System for Electric Vehicle,” in *2022 IEEE International Conference on Current Development in Engineering and Technology (CCET)*, IEEE, Dec. 2022, pp. 1–5. doi: 10.1109/CCET56606.2022.10080389.
- [5] **Z. B. Omariba, L. Zhang, and D. Sun**, (2019) “Review of Battery Cell Balancing Methodologies for Optimizing Battery Pack Performance in Electric Vehicles,” *IEEE Access*, vol. 7, pp. 129335–129352, 2019, doi: 10.1109/ACCESS.2019.2940090.
- [6] **M. Manas, R. Yadav, and R. K. Dubey**, (2023) “Designing a battery Management system for electric vehicles: A congregated approach,” *J Energy Storage*, vol. 74, p. 109439, Dec. 2023, doi: 10.1016/j.est.2023.109439.
- [7] **A. Hauser and R. Kuhn**, (2015) “High-voltage battery management systems (BMS) for electric vehicles,” in *Advances in Battery Technologies for Electric Vehicles*, Elsevier, 2015, pp. 265–282. doi: 10.1016/B978-1-78242-377-5.00011-X.
- [8] **M. Darwish, S. Ioannou, A. Janbey, H. Amreiz, and C. C. Marouchos**, (2021) “Review of Battery Management Systems,” in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/ICECCME52200.2021.9590884.
- [9] **M. Lelie et al.**, “Battery Management System Hardware Concepts: An Overview, (2018) ”*Applied Sciences*, vol. 8, no. 4, p. 534, Mar. 2018, doi: 10.3390/app8040534.

- [10] **B. Li, Y. Fu, S. Shang, Z. Li, J. Zhao, and B. Wang**, (2021) “Research on Functional Safety of Battery Management System (BMS) for Electric Vehicles,” in *2021 International Conference on Intelligent Computing, Automation and Applications (ICAA)*, IEEE, Jun. 2021, pp. 267–270. doi: 10.1109/ICAA53760.2021.00055.
- [11] **A. Chinnusamy, D. Dsa, and S. N. Banavath**, (2024) “Intelligent Battery Protection System for Electric Vehicle Applications,” in *2024 IEEE 18th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*, IEEE, Jun. 2024, pp. 1–5. doi: 10.1109/CPE-POWERENG60842.2024.10604351.
- [12] **M. Kumar, V. K. Yadav, K. Mathuriya, and A. K. Verma**, (2022) “A Brief Review on Cell Balancing for Li-ion Battery Pack (BMS),” in *2022 IEEE 10th Power India International Conference (PIICON)*, IEEE, Nov. 2022, pp. 1–6. doi: 10.1109/PIICON56320.2022.10045109.
- [13] **M. Naguib, P. Kollmeyer, and A. Emadi**, (2021) “Lithium-Ion Battery Pack Robust State of Charge Estimation, Cell Inconsistency, and Balancing: Review,” *IEEE Access*, vol. 9, pp. 50570–50582, 2021, doi: 10.1109/ACCESS.2021.3068776.
- [14] **O. Demirci, S. Taskin, E. Schaltz, and B. Acar Demirci**, (2024) “Review of battery state estimation methods for electric vehicles - Part I: SOC estimation,” *J Energy Storage*, vol. 87, p. 111435, May 2024, doi: 10.1016/j.est.2024.111435.
- [15] **R. R. Kumar, C. Bharatiraja, K. Udhayakumar, S. Devakirubakaran, K. S. Sekar, and L. Mihet-Popa**, (2023) “Advances in Batteries, Battery Modeling, Battery Management System, Battery Thermal Management, SOC, SOH, and Charge/Discharge Characteristics in EV Applications,” *IEEE Access*, vol. 11, pp. 105761–105809, 2023, doi: 10.1109/ACCESS.2023.3318121.
- [16] **R. Eskandari, P. Venugopal, and G. Rietveld**, (2022) “Advanced Battery Management Systems with Integrated Battery Electronics,” in *2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC)*, IEEE, Sep. 2022, pp. 55–61. doi: 10.1109/PEMC51159.2022.9962868.
- [17] **H. Gabbar, A. Othman, and M. Abdussami**, (2021) “Review of Battery Management Systems (BMS) Development and Industrial Standards,” *Technologies (Basel)*, vol. 9, no. 2, p. 28, Apr. 2021, doi: 10.3390/technologies9020028.
- [18] **P. Rahmani et al.**, (2025) “Driving the future: A comprehensive review of automotive battery management system technologies, and future trends,” *J Power Sources*, vol. 629, p. 235827, Feb. 2025, doi: 10.1016/j.jpowsour.2024.235827.
- [19] **Z. Gong et al.**, (2020) “Distributed Control of Active Cell Balancing and Low-Voltage Bus Regulation in Electric Vehicles Using Hierarchical Model-Predictive Control,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10464–10473, Dec. 2020, doi: 10.1109/TIE.2019.2956396.

- [20] **R. Zhang, J. Wu, R. Wang, R. Yan, Y. Zhu, and X. He**, (2019) “A Novel Battery Management System Architecture Based on an Isolated Power/Data Multiplexing Transmission Bus,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 5979–5991, Aug. 2019, doi: 10.1109/TIE.2018.2873143.
- [21] **S. Shete, P. Jog, R. K. Kumawat, and D. K. Palwalia**, (2021) “Battery Management System for SOC Estimation of Lithium-Ion Battery in Electric Vehicle: A Review,” in *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, IEEE, Dec. 2021, pp. 1–4. doi: 10.1109/ICRAIE52900.2021.9703752.
- [22] **H.-C. Chen, S.-S. Li, S.-L. Wu, and C.-Y. Lee**, (2021) “Design of a Modular Battery Management System for Electric Motorcycle,” *Energies (Basel)*, vol. 14, no. 12, p. 3532, Jun. 2021, doi: 10.3390/en14123532.
- [23] **A. C. R and A. Ghosh**, (2021) “Battery Management System in Electric Vehicle,” in *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, IEEE, Jan. 2021, pp. 1–6. doi: 10.1109/ICNTE51185.2021.9487762.
- [24] **S. Thangavel, D. Mohanraj, T. Girijaprasanna, S. Raju, C. Dhanamjayulu, and S. M. Muyeen**, (2023) “A Comprehensive Review on Electric Vehicle: Battery Management System, Charging Station, Traction Motors,” *IEEE Access*, vol. 11, pp. 20994–21019, 2023, doi: 10.1109/ACCESS.2023.3250221.
- [25] **B. Devi and V. S. Kumar**, (2022) “Lithium-ion Battery Management System: A review,” in *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, IEEE, Dec. 2022, pp. 1–6. doi: 10.1109/ICPECTS56089.2022.10047494.
- [26] **J. Vijaychandra and Ł. Knypiński**, (2024) “A Comprehensive Review on Challenges and Possible Solutions of Battery Management Systems in Electric Vehicles,” in *2024 Progress in Applied Electrical Engineering (PAEE)*, IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/PAEE63906.2024.10701438.
- [27] **M. Jiang et al.**, (2024) “Advances in battery state estimation of battery management system in electric vehicles,” *J Power Sources*, vol. 612, p. 234781, Aug. 2024, doi: 10.1016/j.jpowsour.2024.234781.
- [28] **F. Ongaro, S. Saggini, and P. Mattavelli**, (2012) “Li-Ion Battery-Supercapacitor Hybrid Storage System for a Long Lifetime, Photovoltaic-Based Wireless Sensor Network,” *IEEE Trans Power Electron*, vol. 27, no. 9, pp. 3944–3952, Sep. 2012, doi: 10.1109/TPEL.2012.2189022.
- [29] **A. Manenti, A. Abba, A. Merati, S. M. Savaresi, and A. Geraci**, (2011) “A New BMS Architecture Based on Cell Redundancy,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4314–4322, Sep. 2011, doi: 10.1109/TIE.2010.2095398.

- [30] **Z. Zhang, Y.-Y. Cai, Y. Zhang, D.-J. Gu, and Y.-F. Liu**, (2016) “A Distributed Architecture Based on Microbank Modules With Self-Reconfiguration Control to Improve the Energy Efficiency in the Battery Energy Storage System,” *IEEE Trans Power Electron*, vol. 31, no. 1, pp. 304–317, Jan. 2016, doi: 10.1109/TPEL.2015.2406773.
- [31] **C. Iclodean, B. Varga, N. Burnete, D. Cimerdean, and B. Jurchiş**, (2017) “Comparison of Different Battery Types for Electric Vehicles,” *IOP Conf Ser Mater Sci Eng*, vol. 252, no. 1, p. 012058, Oct. 2017, doi: 10.1088/1757-899X/252/1/012058.
- [32] **S. Manzetti and F. Mariasiu**, (2015) “Electric vehicle battery technologies: From present state to future systems,” *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 1004–1012, Nov. 2015, doi: 10.1016/j.rser.2015.07.010.
- [33] **M. Umair Ali, S. Hussain Nengroo, M. Adil Khan, K. Zeb, M. Ahmad Kamran, and H.-J. Kim**, (2018) “A Real-Time Simulink Interfaced Fast-Charging Methodology of Lithium-Ion Batteries under Temperature Feedback with Fuzzy Logic Control,” *Energies (Basel)*, vol. 11, no. 5, p. 1122, May 2018, doi: 10.3390/en11051122.
- [34] **M. Ghalkhani and S. Habibi**, (2022) “Review of the Li-Ion Battery, Thermal Management, and AI-Based Battery Management System for EV Application,” *Energies (Basel)*, vol. 16, no. 1, p. 185, Dec. 2022, doi: 10.3390/en16010185.
- [35] **R. Pakdel, M. Yavarinasab, M. R. Zibad, and M. R. Almohaddesn**, (2022) “Design and Implementation of Lithium Battery Management System for Electric Vehicles,” in *2022 9th Iranian Conference on Renewable Energy & Distributed Generation (ICREDG)*, IEEE, Feb. 2022, pp. 1–6. doi: 10.1109/ICREDG54199.2022.9804549.
- [36] **M. A. Hannan, Md. M. Hoque, A. Hussain, Y. Yusof, and P. J. Ker**, (2018) “State-of-the-Art and Energy Management System of Lithium-Ion Batteries in Electric Vehicle Applications: Issues and Recommendations,” *IEEE Access*, vol. 6, pp. 19362–19378, Mar. 2018, doi: 10.1109/ACCESS.2018.2817655.
- [37] **R. Xiong, J. Cao, Q. Yu, H. He, and F. Sun**, (2018) “Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles,” *IEEE Access*, vol. 6, pp. 1832–1843, Dec. 2018, doi: 10.1109/ACCESS.2017.2780258.
- [38] **K. W. E. Cheng, B. P. Divakar, H. Wu, K. Ding, and H. F. Ho**, (2011) “Battery-Management System (BMS) and SOC Development for Electrical Vehicles,” *IEEE Trans Veh Technol*, vol. 60, no. 1, pp. 76–88, Jan. 2011, doi: 10.1109/TVT.2010.2089647.
- [39] **A. K. M. A. Habib, M. K. Hasan, G. F. Issa, D. Singh, S. Islam, and T. M. Ghazal**, (2023) “Lithium-Ion Battery Management System for Electric Vehicles: Constraints, Challenges, and Recommendations,” *Batteries*, vol. 9, no. 3, p. 152, Feb. 2023, doi: 10.3390/batteries9030152.

- [40] **J. Zhang, L. Zhang, F. Sun, and Z. Wang,** (2018) “An Overview on Thermal Safety Issues of Lithium-ion Batteries for Electric Vehicle Application,” *IEEE Access*, vol. 6, pp. 23848–23863, May 2018, doi: 10.1109/ACCESS.2018.2824838.
- [41] **R. Balamurugan, P. RamKumar, B. Singaraj, C. Vasudev, K. K. Kumar, and M. W. Iruthayarajan,** (2022) “A Survey on various Electric Vehicle batteries – Battery power management and performance monitoring system,” in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, IEEE, Feb. 2022, pp. 1497–1503. doi: 10.1109/ICAIS53314.2022.9743050.
- [42] **M. Rouholamini et al.,** (2022) “A Review of Modeling, Management, and Applications of Grid-Connected Li-Ion Battery Storage Systems,” *IEEE Trans Smart Grid*, vol. 13, no. 6, pp. 4505–4524, Nov. 2022, doi: 10.1109/TSG.2022.3188598.
- [43] **S. A. Hasib et al.,** (2021) “A Comprehensive Review of Available Battery Datasets, RUL Prediction Approaches, and Advanced Battery Management,” *IEEE Access*, vol. 9, pp. 86166–86193, 2021, doi: 10.1109/ACCESS.2021.3089032.
- [44] **A. Ria and P. Dini,** (2024) “A Compact Overview on Li-Ion Batteries Characteristics and Battery Management Systems Integration for Automotive Applications,” *Energies (Basel)*, vol. 17, no. 23, p. 5992, Nov. 2024, doi: 10.3390/en17235992.
- [45] **M. S. Ramkumar et al.,** (2022) “Review on Li-Ion Battery with Battery Management System in Electrical Vehicle,” *Advances in Materials Science and Engineering*, vol. 2022, pp. 1–8, May 2022, doi: 10.1155/2022/3379574.
- [46] **S. ceven, E. Kucukkulahli, A. Albayrak, and Y. Bicen,** (2021) “Development of IoT Based Battery Management System,” in *2021 2nd International Informatics and Software Engineering Conference (IISEC)*, IEEE, Dec. 2021, pp. 1–5. doi: 10.1109/IISEC54230.2021.9672346.
- [47] **S. Bhat and Sudharshana,** (2024) “Battery Management System for Electrical Devices: A Review,” in *2024 5th International Conference on Image Processing and Capsule Networks (ICIPCN)*, IEEE, Jul. 2024, pp. 881–886. doi: 10.1109/ICIPCN63822.2024.00152.
- [48] **A. T. Elsayed, C. R. Lashway, and O. A. Mohammed,** (2015) “Advanced Battery Management and Diagnostic System for Smart Grid Infrastructure,” *IEEE Trans Smart Grid*, vol. 7, no. 2, pp. 1–1, Mar. 2015, doi: 10.1109/TSG.2015.2418677.
- [49] **S. Halder, S. Mondal, A. Mondal, and R. Banerjee,** (2020) “Battery Management System Using State of Charge Estimation: An IOT Based Approach,” in *2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTE)*, IEEE, Feb. 2020, pp. 1–5. doi: 10.1109/NCETSTE48365.2020.9119945.

- [50] **V. Rukkumani, T. Anitha, P. A. Evangilin, P. Booja Aniruti, and P. Deepthiga**, (2023) “IoT-based Battery Health Monitoring System for Electric Vehicle,” in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, IEEE, Mar. 2023, pp. 634–638. doi: 10.1109/ICEARS56392.2023.10085388.
- [51] **M. Asaad, F. Ahmad, M. S. Alam, and Y. Rafat**, (2018) “IoT Enabled Monitoring of an Optimized Electric Vehicle’s Battery System,” *Mobile Networks and Applications*, vol. 23, no. 4, pp. 994–1005, Aug. 2018, doi: 10.1007/s11036-017-0957-z.
- [52] **Y.-D. Yang, K.-Y. Hu, and C.-H. Tsai**, (2020) “Digital Battery Management Design for Point-of-Load Applications With Cell Balancing,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6365–6375, Aug. 2020, doi: 10.1109/TIE.2019.2940001.
- [53] **S. Kocsis Szürke, M. Szabó, S. Szalai, and S. Fischer**, (2024) “Deformation Analysis of Different Lithium Battery Designs Using the DIC Technique,” *Energies (Basel)*, vol. 17, no. 2, p. 323, Jan. 2024, doi: 10.3390/en17020323.
- [54] **U. Salman, S. Belaish, Z. Ji, D. Huang, N. Zheng, and B. Xu**, (2022) “Comparing the economic value of lithium-ion battery technologies in the nine wholesale electricity markets in North America,” *iEnergy*, vol. 1, no. 3, pp. 363–373, Sep. 2022, doi: 10.23919/IEN.2022.0044.
- [55] **B. Emanet and I. Kiyak**, (2021) “Performance Effective Battery Management System(BMS) Design of Mini Electric Vehicles,” in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, Oct. 2021, pp. 744–749. doi: 10.1109/ISMSIT52890.2021.9604678.
- [56] **P. Aruna and P. V Vasan**, (2019) “Review on Energy Management System of Electric Vehicles,” in *2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC)*, IEEE, Aug. 2019, pp. 371–374. doi: 10.1109/ICPEDC47771.2019.9036689.
- [57] **X. Huang, X. Sui, D.-I. Stroe, and R. Teodorescu**, (2019) “A Review of Management Architectures and Balancing Strategies in Smart Batteries,” in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, Oct. 2019, pp. 5909–5914. doi: 10.1109/IECON.2019.8926687.
- [58] **A. Popp, H. Fechtner, B. Schmuelling, S. Kremzow-Tennie, T. Scholz, and F. Pautzke**, (2014) “Battery Management Systems Topologies: Applications : Implications of different voltage levels,” in *2021 IEEE 4th International Conference on Power and Energy Applications (ICPEA)*, IEEE, Oct. 2021, pp. 43–50. doi: 10.1109/ICPEA52760.2021.9639285.
- [59] **M. M. Wenger, R. Filimon, V. R. H. Lorentz, and M. Marz**, (2014) “A robust contactless capacitive communication link for high power battery systems,” in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, IEEE, Jun. 2014, pp. 1766–1772. doi: 10.1109/ISIE.2014.6864882.

- [60] **S. Mishra, S. C. Swain, and R. K. Samantaray**, (2021) “A Review on Battery Management system and its Application in Electric vehicle,” in *2021 International Conference on Advances in Computing and Communications (ICACC)*, IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/ICACC-202152719.2021.9708114.
- [61] **D. W. Gao**, (2015) “Interfacing Between an ESS and a Microgrid,” in *Energy Storage for Sustainable Microgrid*, Elsevier, 2015, pp. 79–121. doi: 10.1016/B978-0-12-803374-6.00003-2.
- [62] **S. Yu. Luchkin, K. Romanyuk, M. Ivanov, and A. L. Kholkin**, (2015) “Li transport in fresh and aged LiMn₂O₄ cathodes via electrochemical strain microscopy,” *J Appl Phys*, vol. 118, no. 7, Aug. 2015, doi: 10.1063/1.4927816.
- [63] **S. Gold**, (1997) “A PSPICE macromodel for lithium-ion batteries,” in *The Twelfth Annual Battery Conference on Applications and Advances*, IEEE, pp. 215–222. doi: 10.1109/BCAA.1997.574106.
- [64] **L. Schärtel, B. Reick, M. Pfeil, and R. Stetter**, (2022) “Analysis and Synthesis of Architectures for Automotive Battery Management Systems,” *Applied Sciences*, vol. 12, no. 21, p. 10756, Oct. 2022, doi: 10.3390/app122110756.
- [65] **P. Messier, F.-A. LeBel, J. Rouleau, and J. P. F. Trovao**, (2018) “Multi-Cell Emulation for Battery Management System Validation,” in *2018 IEEE Vehicle Power and Propulsion Conference (VPPC)*, IEEE, Aug. 2018, pp. 1–6. doi: 10.1109/VPPC.2018.8604959.
- [66] **H. Dai, B. Jiang, X. Hu, X. Lin, X. Wei, and M. Pecht**, (2021) “Advanced battery management strategies for a sustainable energy future: Multilayer design concepts and research trends,” *Renewable and Sustainable Energy Reviews*, vol. 138, p. 110480, Mar. 2021, doi: 10.1016/j.rser.2020.110480.
- [67] **A. Samanta and S. S. Williamson**, (2021) “A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges,” Aug. 16, 2021. doi: 10.20944/preprints202108.0334.v1.
- [68] **V. S. R. Kosuru and A. Kavasseri Venkitaraman**, (2023) “A Smart Battery Management System for Electric Vehicles Using Deep Learning-Based Sensor Fault Detection,” *World Electric Vehicle Journal*, vol. 14, no. 4, p. 101, Apr. 2023, doi: 10.3390/wevj14040101.
- [69] **T. Faika, T. Kim, and M. Khan**, (2018) “An Internet of Things (IoT)-Based Network for Dispersed and Decentralized Wireless Battery Management Systems,” in *2018 IEEE Transportation Electrification Conference and Expo (ITEC)*, IEEE, Jun. 2018, pp. 1060–1064. doi: 10.1109/ITEC.2018.8450161.
- [70] **B. Yildirim, M. A. Elgendy, A. N. Smith, and V. Pickert**, (2023) “Efficiency Optimized Power-Sharing Algorithm for Modular Battery Energy Storage Systems,” *IEEE Transactions on Industrial Electronics*, vol. 70, no. 11, pp. 11299–11309, Nov. 2023, doi: 10.1109/TIE.2022.3229311.

- [71] **G. Le Gall, N. Montavont, and G. Z. Papadopoulos**, (2022) “IoT Network Management within the Electric Vehicle Battery Management System,” *J Signal Process Syst*, vol. 94, no. 1, pp. 27–44, Jan. 2022, doi: 10.1007/s11265-021-01670-2.
- [72] **K. Chen, L. Luo, W. Lei, P. Lv, and L. Zhang**, (2024) “Design and implementation of online battery monitoring and management system based on the internet of things,” *Front Energy Res*, vol. 12, Sep. 2024, doi: 10.3389/fenrg.2024.1454398.
- [73] **B. G. Carkhuff, P. A. Demirev, and R. Srinivasan**, (2018) “Impedance-Based Battery Management System for Safety Monitoring of Lithium-Ion Batteries,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6497–6504, Aug. 2018, doi: 10.1109/TIE.2017.2786199.
- [74] **X. Wang et al.**, (2022) “A Review of the Power Battery Thermal Management System with Different Cooling, Heating and Coupling System,” *Energies (Basel)*, vol. 15, no. 6, p. 1963, Mar. 2022, doi: 10.3390/en15061963.
- [75] **L. Zhou et al.**, (2023) “State Estimation Models of Lithium-Ion Batteries for Battery Management System: Status, Challenges, and Future Trends,” *Batteries*, vol. 9, no. 2, p. 131, Feb. 2023, doi: 10.3390/batteries9020131.
- [76] **S. B and P. Pradeepa**, (2022) “Review on Battery Management System in EV,” in *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, IEEE, Jul. 2022, pp. 1–4. doi: 10.1109/ICICCSP53532.2022.9862367.
- [77] **T. Shaikh, S. Kadam, S. Sonawane, P. Narkar, P. Mhatre, and M. Muthu**, (2023) “Implementation of IoT-based Real-time Monitoring of Battery Management System for Electric Vehicles,” in *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Jun. 2023, pp. 32–39. doi: 10.1109/ICCES57224.2023.10192869.
- [78] **R. Manivannan**, (2024) “Research on IoT-based hybrid electrical vehicles energy management systems using machine learning-based algorithm,” *Sustainable Computing: Informatics and Systems*, vol. 41, p. 100943, Jan. 2024, doi: 10.1016/j.suscom.2023.100943.
- [79] **T. Duraisamy and D. Kaliyaperumal**, (2020) “Active cell balancing for electric vehicle battery management system,” *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 11, no. 2, p. 571, Jun. 2020, doi: 10.11591/ijpeds.v11.i2.pp571-579.
- [80] **S. Singh, V. More, and R. Batheri**, (2022) “Driving Electric Vehicles Into the Future With Battery Management Systems,” *IEEE Engineering Management Review*, vol. 50, no. 3, pp. 157–161, Sep. 2022, doi: 10.1109/EMR.2022.3194655.
- [81] **D. Shi et al.**, (2023) “Cloud-Based Artificial Intelligence Framework for Battery Management System,” *Energies (Basel)*, vol. 16, no. 11, p. 4403, May 2023, doi: 10.3390/en16114403.

- [82] **R. D. Anderson, R. Zane, G. Plett, D. Maksimovic, K. Smith, and M. S. Trimboli**, (2017) “Life Balancing – A Better Way to Balance Large Batteries,” Mar. 2017. doi: 10.4271/2017-01-1210.
- [83] **J.-C. Lin**, (2017) “Development of a New Battery Management System with an Independent Balance Module for Electrical Motorcycles,” *Energies (Basel)*, vol. 10, no. 9, p. 1289, Aug. 2017, doi: 10.3390/en10091289.
- [84] **V. C. Valchev, P. V. Yankov, and D. D. Stefanov**, (2018) “Improvement on LiFePO₄ Cell Balancing Algorithm,” *TEM Journal*, pp. 19–24, Feb. 2018, doi: 10.18421/TEM71-03.
- [85] **F. Baronti, R. Roncella, and R. Saletti**, (2014) “Performance comparison of active balancing techniques for lithium-ion batteries,” *J Power Sources*, vol. 267, pp. 603–609, Dec. 2014, doi: 10.1016/j.jpowsour.2014.05.007.
- [86] **C. Piao, Z. Wang, J. Cao, W. Zhang, and S. Lu**, (2015) “Lithium-Ion Battery Cell-Balancing Algorithm for Battery Management System Based on Real-Time Outlier Detection,” *Math Probl Eng*, vol. 2015, pp. 1–12, 2015, doi: 10.1155/2015/168529.
- [87] **W. Diao, N. Xue, V. Bhattacharjee, J. Jiang, O. Karabasoglu, and M. Pecht**, (2017) “Active battery cell equalization based on residual available energy maximization,” *Appl Energy*, vol. 210, pp. 690–698, Jan. 2018, doi: 10.1016/j.apenergy.2017.07.137.
- [88] **J. Gallardo-Lozano, E. Romero-Cadaval, M. I. Milanes-Montero, and M. A. Guerrero-Martinez**, (2013) “Battery equalization active methods,” *J Power Sources*, vol. 246, pp. 934–949, Jan. 2014, doi: 10.1016/j.jpowsour.2013.08.026.
- [89] **Z. Zhao, H. Hu, Z. He, H. H.-C. Iu, P. Davari, and F. Blaabjerg**, (2023) “Power Electronics-Based Safety Enhancement Technologies for Lithium-Ion Batteries: An Overview From Battery Management Perspective,” *IEEE Trans Power Electron*, vol. 38, no. 7, pp. 8922–8955, Jul. 2023, doi: 10.1109/TPEL.2023.3265278.
- [90] **J. Gallardo-Lozano, E. Romero-Cadaval, M. I. Milanes-Montero, and M. A. Guerrero-Martinez**, (2015) “A novel active battery equalization control with on-line unhealthy cell detection and cell change decision,” *J Power Sources*, vol. 299, pp. 356–370, Dec. 2015, doi: 10.1016/j.jpowsour.2015.09.005.
- [91] **M. Caspar and S. Hohmann**, (2014) “Optimal Cell Balancing with Model-based Cascade Control by Duty Cycle Adaption,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 10311–10318, 2014, doi: 10.3182/20140824-6-ZA-1003.01613.
- [92] **K.-Y. Oh and B. I. Epureanu**, (2017) “A phenomenological force model of Li-ion battery packs for enhanced performance and health management,” *J Power Sources*, vol. 365, pp. 220–229, Oct. 2017, doi: 10.1016/j.jpowsour.2017.08.058.

- [93] **M.-J. Kim, S.-H. Chae, and Y.-K. Moon**, (2020) “Adaptive Battery State-of-Charge Estimation Method for Electric Vehicle Battery Management System,” in *2020 International SoC Design Conference (ISOCC)*, IEEE, Oct. 2020, pp. 288–289. doi: 10.1109/ISOCC50952.2020.9332950.
- [94] **E. İ. Tezde and H. İ. Okumuş**, (2018) “Batarya Modelleri ve Şarj Durumu (SoC) Belirleme Battery Models and State of Charge (SoC) Determination,” 2018.
- [95] **G. SEVİM and N. ALTIN**, (2023) “Lityum Bataryalarda Şarj Durumu (SoC) ve Sağlık Durumu (SoH) Kestirimi,” *Karamanoğlu Mehmetbey Üniversitesi Mühendislik ve Doğa Bilimleri Dergisi*, vol. 5, no. 1, pp. 33–51, Jun. 2023, doi: 10.55213/kmujens.1250621.
- [96] **A. Harakare et al.**, (2022) “Design of Battery Management System for an Autonomous Underwater Vehicle,” in *Oceans Conference Record (IEEE)*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/OCEANSSChennai45887.2022.9775282.
- [97] **G. Krishna, R. Singh, A. Gehlot, and S. V. Akram**, (2024) “An IoT-based predictive model for improved battery management system using advanced LSTM model,” *J Energy Storage*, vol. 101, Nov. 2024, doi: 10.1016/j.est.2024.113694.
- [98] **Url** A. Devices Inc, “LTC6811-1/LTC6811-2 (Rev. C) 12-Cell Battery Stack Monitors.” [Online]. Available: www.analog.com
- [99] **Url** “MAX11068 12-Channel, High-Voltage Sensor, Smart Data-Acquisition Interface.” [Online]. Available: www.maximintegrated.com
- [100] **Url** “MAX17843 - 12-Channel, High-Voltage Smart Sensor Data-Acquisition Interface.” [Online]. Available: www.maximintegrated.com
- [101] **Url** “BQ76972 3-Series to 16-Series High Accuracy Battery Monitor and Protector for Li-Ion, Li-Polymer, and LiFePO 4 Battery Packs,” 2023. [Online]. Available: www.ti.com
- [102] **Url** R. Electronics Corporation, “ISL78600 Multi-Cell Li-Ion Battery Manager Datasheet.”
- [103] **Url** A. Devices, “LTC6802-1 (Rev. B) Multicell Battery Stack Monitor.” [Online]. Available: www.analog.com
- [104] **Url** R. Electronics Corporation, “ISL78610 Multi-Cell Li-Ion Battery Manager.”
- [105] **Url** L. Technology Corporation, “LTC6803-1/LTC6803-3 - Multicell Battery Stack Monitor.” [Online]. Available: <http://www.linear.com/leadfree/>
- [106] **Url** L. Technology Corporation, “LTC6804-1/LTC6804-2 – Multicell Battery Monitors.” [Online]. Available: www.linear.com/LTC6804-1
- [107] **Url** L. Technology Corporation, “LTC6803-2/LTC6803-4 - Multicell Battery Stack Monitor.” [Online]. Available: <http://www.linear.com/leadfree/>
- [108] **Url** “bq76PL536A-Q1 3-to-6 Series Cell Lithium-Ion Battery Monitor and Secondary Protection IC for EV and HEV Applications”.

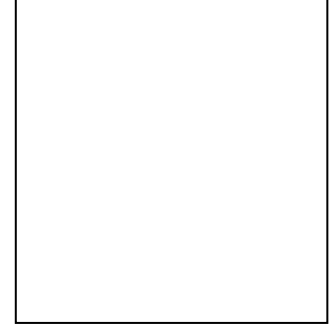
- [109] **Url** “MAX17048/MAX17049 3 μ A 1-Cell/2-Cell Fuel Gauge with ModelGauge.” [Online]. Available: www.maximintegrated.com
- [110] **Url** A. Devices, “LTC6801 (Rev. F) - Independent Multicell Battery Stack Fault Monitor.” [Online]. Available: www.analog.com
- [111] **Url** R. Hwang, “bq76PL455A-Q1 Design Recommendations,” 2016. [Online]. Available: www.ti.com
- [112] **Url** “BQ76952 3-Series to 16-Series High Accuracy Battery Monitor and Protector for Li-Ion, Li-Polymer, and LiFePO₄ Battery Packs,” 2021. [Online]. Available: www.ti.com
- [113] **J. Wang and Y. C. Zhang**, (2021) “A Review on Internet of Things Based Smart Home,” in *2021 International Conference on Culture-oriented Science & Technology (ICCST)*, IEEE, Nov. 2021, pp. 273–277. doi: 10.1109/ICCST53801.2021.00065.
- [114] **L. Da Xu, W. He, and S. Li**, (2014) “Internet of things in industries: A survey,” Nov. 01, 2014, *IEEE Computer Society*. doi: 10.1109/TII.2014.2300753.
- [115] **A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi**, (2014) “Internet of Things for Smart Cities,” *IEEE Internet Things J*, vol. 1, no. 1, pp. 22–32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.
- [116] **J. Jung, K. Kim, and J. Park**, (2019) “Framework of Big data Analysis about IoT-Home-device for supporting a decision making an effective strategy about new product design,” in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, IEEE, Feb. 2019, pp. 582–584. doi: 10.1109/ICAIIIC.2019.8669086.
- [117] **S.-M. Kim, H.-S. Choi, and W.-S. Rhee**, (2015) “IoT home gateway for auto-configuration and management of MQTT devices,” in *2015 IEEE Conference on Wireless Sensors (ICWiSe)*, IEEE, Aug. 2015, pp. 12–17. doi: 10.1109/ICWISE.2015.7380346.
- [118] **I. A. Shah, F. A. Malik, and S. A. Ahmad**, (2016) “Enhancing security in IoT based Home automation using Reed Solomon Codes,” in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, IEEE, Mar. 2016, pp. 1639–1642. doi: 10.1109/WiSPNET.2016.7566417.
- [119] **M. Ismail and R. Ahmed**, (2024) “A Comprehensive Review of Cloud-Based Lithium-Ion Battery Management Systems for Electric Vehicle Applications,” *IEEE Access*, vol. 12, pp. 116259–116273, 2024, doi: 10.1109/ACCESS.2024.3446880.
- [120] **C. S. Cabading, J. Francis Natividad, and R. V. Santiago**, (2022) “Design of A Hybrid Renewable Energy System with IoT Monitoring and Battery Management,” in *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, IEEE, Dec. 2022, pp. 1–6. doi: 10.1109/HNICEM57413.2022.10109524.
- [121] **Url** Akiba, “<https://akiba.com.tr/tr>.”

- [122] **Url** “BQ76952 3-Series to 16-Series High Accuracy Battery Monitor and Protector for Li-Ion, Li-Polymer, and LiFePO₄ Battery Packs,” 2021. [Online]. Available: www.ti.com
- [123] **Url** devreyakan, “Haberleşme Protokolleri,” <https://devreyakan.com/seri-haberlesme-protokolleri/>.
- [124] **Url** “PIC18F2455/2550/4455/4550 Data Sheet”.
- [125] **Url** “ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET.”
- [126] **Url** “This is information on a product in full production. STM32G431x6 STM32G431x8 STM32G431xB,” 2021. [Online]. Available: www.st.com
- [127] **Url** PCB Artists, “https://pcbartists.com/design/embedded/stm32-wifi-options/?srsId=AfmBOOrDjvIK1TGCJp-K_15H15kG9bn7HIs6kwxHbr0xgqWUHbPuyZ_k.”
- [128] **Url** “<https://www.microchip.com/en-us/product/atwinc1500>.”
- [129] **Url** “<https://www.silabs.com/documents/public/data-sheets/wf200-datasheet.pdf>.”
- [130] **Url** “https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf.”
- [131] **Url** “ESP32 Series Datasheet Version 4.7 2.4 GHz Wi-Fi + Bluetooth[®] + Bluetooth LE SoC Including.” [Online]. Available: www.espressif.com
- [132] **Url** “<https://www.kicad.org>.”
- [133] **Url** “<https://www.altium.com>.”
- [134] **Url** “<https://www.autodesk.com/products/eagle/overview?term=1-YEAR&tab=subscription>.”
- [135] **Url** “<https://www.labcenter.com>.”
- [136] **Url** “https://www.cadence.com/en_US/home/tools/pcb-design-and-analysis/orcad.html.”
- [137] **Url** “<https://firebase.google.com>.”
- [138] **Url** “<https://azure.microsoft.com/tr-tr>.”
- [139] **Url** “<https://aws.amazon.com/tr/dynamodb/>.”
- [140] **Url** “<https://www.mongodb.com>.”
- [141] **Url** “<https://www.couchbase.com>.”
- [142] **Url** “<https://supabase.com>.”
- [143] **Url** “<https://www.gtech.com.tr/sql-vs-nosql-farklari-avantaj-ve-dezavantajlari/>.”
- [144] **Url** “<https://visualstudio.microsoft.com/tr/>.”
- [145] **Url** “<https://www.qt.io>.”
- [146] **Url** “<https://www.electronjs.org>.”

- [147] **Url** “<https://alternativeto.net/software/app-inventor-for-android/>.”
- [148] **Url** “<https://appinventor.mit.edu>.”
- [149] **Url** “<https://thinkable.com>.”
- [150] **Url** “<https://developer.android.com/studio?hl=tr>.”
- [151] **Url** “<https://www.kodular.io>.”
- [152] **Url** “<https://bubble.io>.”
- [153] **Url** “https://www-teamwavelength-com.translate.goog/thermistor-basics/?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=wa.”
- [154] **Url** “Molicel Model INR-18650-M35A”.
- [155] **Url** “BQ76952 Technical Reference Manual,” 2022. [Online]. Available: www.ti.com



ÖZGEÇMİŞ



Ad-Soyad : İbrahim Ethem DERE

Doğum Tarihi ve Yeri :

E-posta :

ÖĞRENİM DURUMU:

- **Lisans** : 2021, Bursa Teknik Üniversitesi, Elektrik Elektronik Mühendisliği
- **Yüksek Lisans** : 2024, Bursa Teknik Üniversitesi, Elektrik Elektronik Mühendisliği Anabilim Dalı

MESLEKİ DENEYİM VE ÖDÜLLER:

- Yasa Teknoloji (Donanım Tasarım Mühendisi)
- Elfatek Elektronik (Donanım Tasarım Mühendisi)

TEZDEN TÜRETİLEN ESERLER, SUNUMLAR VE PATENTLER:

- **Dere İ.E. & Ertekin D. & Telli T. & Canan S.** (2024) International Conference On Engineering Technologies – “Development Of A Battery Management System With IoT-Enabled Data Tracking For Autonomous Load Robot”

DİĞER ESERLER, SUNUMLAR VE PATENTLER:

- **Dere İ.E. & Ertekin D.** (2022) Elektrik-Elektronik ve Biyomedikal Mühendisliği Konferansı – “A Control Mechanism for Battery Charging in Electric Vehicles by a Bidirectional DC-DC Converter”
- **Saday A. & Canan S. & Dere İ.E.** (2024) Intelligent Methods In Engineering Sciences – “ Comparison of Data Reduction Algorithms for Real-Time Data Processing in Embedded Systems ”