

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**MODBUS ENDÜSTRİYEL PROTOKOL TABANLI  
DİZEL JENERATÖR SÜRÜCÜ KARTI TASARIMI VE  
İNSAN MAKİNE ARAYÜZÜ TASARIMI**

**YÜKSEK LİSANS TEZİ**

**Motuma ABAFOĞİ**

**Enstitü Anabilim Dalı : ELEKTRİK ELEKTRONİK  
MÜHENDİSLİĞİ**

**Tez Danışmanı : Yrd. Doç. Dr. Metin VARAN**

**Eylül 2016**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**MODBUS ENDÜSTRİYEL PROTOKOL TABANLI  
DİZEL JENERATÖR SÜRÜCÜ KARTI TASARIMI VE  
İNSAN MAKİNE ARAYÜZÜ TASARIMI**

**YÜKSEK LİSANS TEZİ**

**Motuma ABAFOĞİ**

**Enstitü Anabilim Dalı : ELEKTRİK ELEKTRONİK  
MÜHENDİSLİĞİ**

**Bu tez 21.09.2016 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.**

**Yrd. Doç. Dr.  
Metin VARAN  
Jüri Başkanı**

**Prof. Dr.  
Abdullah FERİKOĞLU  
Üye**

**Doç. Dr.  
Ali ÖZTÜRK  
Üye**

## **BEYAN**

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Motuma ABAFOĞİ

21.09.2016

## TEŐEKKÜR

İlk önce bu tez çalışmasının tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Yrd. Doç. Dr. Metin VARAN'a teşekkürlerimi sunarım.

Ayrıca, Nero Endüstri Genel Müdürü Makine Mühendisi Alican ÖKCÜN Bey'e bu tez çalışmasının yapılmasında değerli destekleri için teşekkür etmeyi bir borç bilirim.

Son olarak, çalışmanın maddi açıdan desteklenmesine olanak sağlayan Sakarya Üniversitesi Bilimsel Araştırma Projeleri (BAP) Komisyon Başkanlığına teşekkür ederim.

# İÇİNDEKİLER

TEŞEKKÜR .....	i
İÇİNDEKİLER .....	ii
SİMGELER VE KISALTMALAR LİSTESİ .....	iv
ŞEKİLLER LİSTESİ .....	v
TABLolar LİSTESİ .....	vii
ÖZET .....	viii
SUMMARY .....	ix
BÖLÜM 1.	
GİRİŞ .....	1
1.1. Modbus Temelleri .....	1
1.2. Jeneratör Seti Kontrolörleri .....	2
BÖLÜM 2.	
MODBUS TCP PROTOKOLÜ .....	5
2.1. Modbus Protokolü Versiyonları .....	5
2.2. Modbus TCP Protokolü Temelleri .....	6
2.3. Modbus Mesajlaşma Fonksiyon Kodları .....	8
2.4. Modbus TCP Avantajları .....	8
BÖLÜM 3.	
UYGULAMA YAZILIMLARI VE CİHAZLARI .....	10
3.1. Qt C++ Yazılımı .....	10
3.1.1. Qt Creator .....	11
3.2. CodeWarrior .....	15
3.2.1. Proje türleri .....	15

3.2.2. Proje oluřturma .....	16
3.2.2.1. alıřma tezgahı bařlatma .....	16
3.2.2.2. Kinetis hedef kurulu projesinin oluřturulması .....	17
3.2.2.3. Projeleri hata ayıklaması .....	20
3.3. Freescale Kinetis .....	21
3.3.1. Kinetis K ailesi mikrodenetleyici birimleri .....	23
3.4. TWR-K60D100M .....	23
3.4.1. TWR-K60D100M donanım tanımlaması .....	26
3.4.2. Sistem g¼c¼ .....	27
3.4.3. Hata ayıklama arabirimi .....	27
BÖLÜM 4.	
MODBUS İLE DİZEL JENERATÖR KONTROLÜ .....	29
4.1. Dizel Jeneratörlerin Paralel Senkronizasyonu .....	29
4.2. Modbus İle Birden Fazla Dizel Jeneratör Kontrolü .....	30
BÖLÜM 5.	
MODBUS TCP CLİENT VE SERVER UYGULAMASI .....	33
5.1. MQX .....	33
5.2. Genel Jeneratör Kontrol Sistemleri .....	34
5.3. Modbus Protokol Kütüphanesi .....	36
5.4. MQX Üzerinde Modbus TCP İstemci Uygulaması .....	41
5.5. MQX Üzerinde Modbus TCP Sunucu Uygulaması .....	43
BÖLÜM 6.	
SONUÇ .....	46
KAYNAKLAR .....	48
ÖZGEÇMİŐ .....	51

## **SİMGELER VE KISALTMALAR LİSTESİ**

ADU	: Application Data Unit
MBAP	: Modbus Application Protocol
MQX	: Message Queue eXecutive
PDU	: Protocol Data Unit
RTCS	: Real Time Communication Suite
RTOS	: Real Time Operating System
RTU	: Remote Terminal Unit
TCP	: Transmission Control Protocol
UDP	: User Datagram Protocol

## ŞEKİLLER LİSTESİ

Şekil 1.1. Modbus haberleşmede Master-Slave ilişkisi .....	2
Şekil 1.2. Örnek bir kontrol paneli kullanıcı arayüzü .....	3
Şekil 2.1. Genel Modbus çerçevesi .....	6
Şekil 2.2. Modbus TCP veri paketin oluşturulması .....	7
Şekil 2.3. TCP/IP üzerinde Modbus istek / yanıt .....	7
Şekil 3.1. Qt Creator karşılama sayfası .....	11
Şekil 3.2. Qt Creator yeni proje sihirbazı .....	12
Şekil 3.3. Tasarım modundaki Qt Creator görünümü .....	13
Şekil 3.4. Düzenleme modundaki Qt Creator görünümü .....	14
Şekil 3.5. Program sonucu .....	15
Şekil 3.6. Workspace Launcher iletişim kutusu .....	16
Şekil 3.7. Yeni proje sihirbazı (MCU projesi) .....	17
Şekil 3.8. Yeni proje sihirbazı (Hedef işlemci seçimi) .....	18
Şekil 3.9. Yeni proje sihirbazı (Bağlantı türü) .....	18
Şekil 3.10. Yeni proje sihirbazı (Dil ve derleme araçları seçimi) .....	19
Şekil 3.11. Yeni projesi sihirbazı (Hızlı uygulama geliştirme) .....	20
Şekil 3.12. Debug Konfigürasyonu.....	21
Şekil 3.13. TWR-K60D100M .....	24
Şekil 3.14. TWR-K60D100M MCU modülü ön tarafı .....	25
Şekil 3.15. TWR-K60D100M MCU modülü arka tarafı .....	26
Şekil 3.16. TWR-K60D100M blok diyagramı .....	26
Şekil 4.1. Ortak yola bağlı paralel jeneratörler .....	30
Şekil 4.2. İstemci ve sunucu Modbus TCP / IP bağlantısı emülasyonu .....	31
Şekil 4.3. Tasarlanan denetleyici programı ve jeneratör kontrol panelleri arasında Ethernet üzerinden Modbus bağlantısı .....	32
Şekil 5.1. RTCS katmanları ve protokolleri .....	34



Şekil 5.2. Freescale TWR-SER modülü .....	36
Şekil 5.3. Libmodbus Kütüphanesi .....	37
Şekil 5.4. Modifiye edilen Modbus kütüphanesi .....	37
Şekil 5.5. MQX RTOS üzerinde Modbus istemci uygulaması .....	43
Şekil 5.6. MQX RTOS üzerinde Modbus sunucu uygulaması .....	45

## TABLolar LİSTESİ

Tablo 1.2. Modbus veri tipleri .....	2
Tablo 2.1. Modbus fonksiyon kodları .....	8
Tablo 3.1. Qt çerçevesi .....	10
Tablo 3.2. Kinetis ailesi ürünleri .....	22

## ÖZET

Anahtar kelimeler: Dizel jeneratör, Gerçek zamanlı kontrol, Jeneratör Set Kontrol, Modbus, Qt

Dizel jeneratörler, yüksek güvenilirlik ve düşük yakıt kullanımından dolayı birçok sektörde büyük ölçüde kullanılmaktadır. Ani kesinti durumunda yedek güç kaynakları olarak kullanılmakta ve üretim işlemlerinin devam etmesini sağlamaktadır. Jeneratör grubu (genset) denetleyicisi jeneratörlerin düzgün işleyişini izleme ve kontrol etme imkanı sağlamaktadır. Mevcut jeneratör kontrolörlerinden bazıları motorun uzaktan başlatma ve durdurma, jeneratörün güç ölçümlerini okuma ve ayarlanabilir parametreleri modifiye etme gibi işlevleri sağlamaktadır.

Günümüzde, Modbus TCP PLC ve I/O cihazları gibi araçlar arasında iletişim bağlantısının kurulması için endüstriyel uygulamalarda en yaygın olarak kullanılan ağ protokolünden biri olmuştur. Neredeyse tüm iletişim ortamı üzerinde çalıştırma yeteneğinin olması, mevcut veya yeni sanayi ortamlarında Modbus TCP bağlantısının kurulmasını kolaylaştırmaktadır.

Bu çalışmada, dizel jeneratörlerin kontrolü ve izlenmesi için bir jeneratör denetleyicisi Freescale Kinetis K60D100M mikroişlemcisi üzerine dizayn edilmiştir. İki dizel jeneratörün parametrelerini gerçek zamanlı olarak alıp ona göre doğru kontrol önlemleri alınmıştır. Qt C++ yazılımını kullanarak Modbus TCP protokolü bağlantısı ile verimli çalışan bir interaktif kullanıcı arayüzü tasarlanmıştır.

# **MODBUS INDUSTRIAL PROTOCOL BASED DIESEL GENERATOR SET CONTROLLER DESIGN AND USER INTERFACE DESIGN**

## **SUMMARY**

Keywords: Diesel generator, Real-time control, Generator Set Controller, Modbus, Qt

Diesel generators are prominent for their high reliability and low use of fuel which makes them largely applicable to industries. They are also used as standby power supplies in case of sudden interruption thus aiding in the continuation of manufacturing operations. A generator set (genset) controller allows the user to monitor and control the proper functioning of these generators. Some of the existing genset controllers provide functionalities such as remote starting and stopping of the engine, reading the generator's power measurements, modifying adjustable parameters and many more.

Nowadays, Modbus TCP has become one of the most widely used network protocol in industrial applications for establishing connections amongst products such as PLCs, I/O devices etc. Its ability to run over nearly all communication media makes the establishment of Modbus TCP connection on an existing or new industrial floor quite easy.

In this thesis, a generator set controller has been designed for monitoring and controlling of diesel generators by using the Freescale Kinetis K60 processor. Different generator parameters of two diesel generators have been acquired in real time and proper control measures have been taken accordingly. Qt C++ software is employed in the design of an interactive user interface that works efficiently with the established Modbus TCP protocol connection.

## **BÖLÜM 1. GİRİŞ**

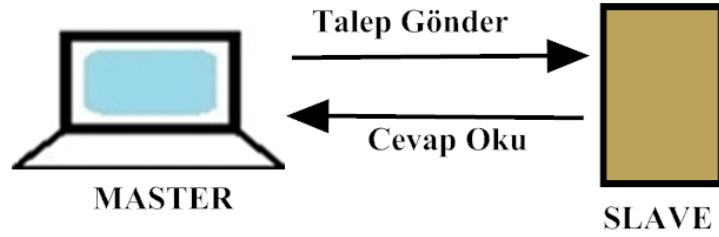
Dizel jeneratörü bir set olarak dizel motoru, kontrol sistemleri ve kesicilerden oluşmaktadır. Çeşitli sanayiler dizel jeneratörleri ana güç veya yedek güç kaynakları olarak kullanılmaktadır [1]. Şu anda, uygun bir güç sistemi yönetimine ihtiyaç duyulması kontrol sistemlerinin gelişmesine neden olmuştur. Çok sayıda işlevleri sağlayan jeneratör kontrol setleri geliştirilmiş durumdadır. Kullanıcı uzaktan bu jeneratörlerin işlemlerini kontrol edebilmektedir.

Modbus, CAN, PROFIBUS, PROFINET, IEC protokolleri vb. dahil olmak üzere şu anda endüstriyel otomasyon için kullanılan birçok iletişim protokolleri bulunmaktadır [2, 3, 4]. Jeneratör kontrolörlerinde kullanılmakta olan çeşitli iletişim protokollerinden, Modbus protokolü yaygın olarak kullanılmaktadır.

### **1.1. Modbus Temelleri**

1979 yılında geliştirilen Modbus sanayi protokolün yürürlüğe girmesinden sonra otomasyon elektronik ekipmanları arasındaki iletişim çok daha kolay hale gelmiştir. Bir seri RTU, TCP / IP üzerinde veri transferi, ve kullanıcı datagram protokolü (UDP) uygulanabilmektedir. Günümüzde birçok cihaz farklı iletişim ağları üzerinden, basit, güvenilir ve verimli bir iletişim için bu protokolü kullanılmaktadır.

Modbus TCP protokolü, genellikle bir SCADA sistemi entegrasyonu için kullanılmaktadır. Ayrıca, PLC ve sensörler ile endüstriyel ekipman kontrol etmek için de kullanılmaktadır [5]. Master ve slave mimarisi kullanılmaktadır. Master iletişimi başlatmakta ve slave ise masterin isteğine göre cevap vermektedir. Bu durumda Modbus protokolü cihazlar arasındaki iletişim esnasında mastera tam kontrol vermektedir.



Şekil 1.1. Modbus haberleşmede Master-Slave ilişkisi [6]

Modbus birinci tabaka uygulama veri birimi ya da Application Data Unit (ADU) olan bir veri tabakalı takımı içermektedir. Her ADU için bir protokol veri birimi ya da Protocol Data Unit (PDU) vardır. Üç uygulama veri birimi şunlardır: uzak terminal birimi (RTU), ASCII ve TCP / IP.

TCP / IP’de Modbus istekleri için özel bağlantılar ve tanımlayıcılar daha sık kullanılmaktadır. RTU iletişim için kompakt ikili gösterimi kullanıldığı gibi ASCII’de ASCII karakterleri kullanılmaktadır. Modbus veri tipleri aşağıdaki Şekil 1.2.’de gösterilmiştir.

Tablo 1.1. Modbus veri tipleri [7]

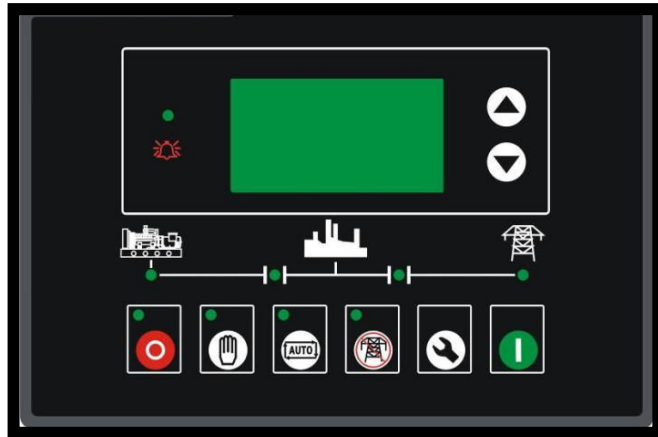
Ana Tablo	Nesne Türü	Erişim Türü
Ayrık Giriş	1 bit	Sadece Okunabilir
Bobin	1 bit	Okunabilir-Yazılabilir
Giriş Registerı	16-bit	Sadece Okunabilir
Tutucu (Holding) Registerı	16-bit	Okunabilir-Yazılabilir

## 1.2. Jeneratör Seti Kontrolörleri

Birçok gelişmiş makinenin verimli kontrol ve düzgün işleyişini izlemek için kullanıcıya sağlayan bir kullanıcı arabirimi gerektirmektedir. Jeneratör gibi elektrikli

makineler aşırı ısınma, yavaşlama veya hızlanma gibi çeşitli nedenlerden dolayı elektrik sinyalleri üretmekte ve bununla makinenin performans kontrolü işlenebilmektedir.

Modern jeneratörlerde çeşitli değişiklikleri algılamak için sensörler geliştirilmiş ve makinenin düzgün yönetimi için sensörlerden gelen girişi mikroişlemcisi olan bir elektronik aygıt ve bir kontrol paneli üzerinden kontrol edilebilmektedir. Bu geri bildirimler ısı seviyesini gösteren sıcaklık ölçümleri, düşük hız, yüksek hız, düşük veya yüksek yağ basıncı altında veya üstünde olabilmektedir. Mikroişlemci durumu işlemekte ve daha sonra da buna göre performansı düzenleyen etkili önlemler almaktadır. İçinde gömülü mikroişlemci veya mikrodenetleyici sayesinde kontrol panelleri giderek kritik hale gelmiştir. Bu kontrolörü sensörden gelen girişi değerlendirmek ve bazı kontrol kurallarına göre cevap verecek şekilde programlanmıştır.



Şekil 1.2. Örnek bir kontrol paneli kullanıcı arayüzü [8]

İki veya daha fazla jeneratör ortak çalışırken güç sağlamak için jeneratörlerin paralel senkronizasyon olması gereklidir. Paralel senkronizasyon bir jeneratörün farklı parametrelerine uyan karşılığını yada bus çubuğunu bulma işlemidir. Bu işlem, birkaç küçük birimlerin paralel olarak faaliyet gösterdiği ticari santrallerde fabrikanın ihtiyacını karşılamak için yaygın kullanılmaktadır. Bu, hizmet sürekliliğini sağlayarak güç kaynağını daha güvenilir yapmaktadır.

Bugüne kadar, Modbus protokolüne dayanarak geliştirilen kontrol sistemleri ile ilgili birçok çalışmalar bulunmaktadır [9, 10, 11]. Derwin Suhartono [12] yerel ağ ve internet üzerinden Modbus protokolüne dayanan çoklu jeneratör setleri izleyebilir bir uygulama tasarımı başarmıştır. Bu tasarım kullanıcıya sadece bir arayüz kullanarak birden fazla jeneratörler uzaktan kontrol ve izleme olanağı sağlamaktadır.

Öte yandan, ShiBen Zhu [13] trafo ölçüm ekipmanı kontrol etmek ve veriyi Microsoft Access icinde kaydetmek için kullanılabilecek bir monitory sistem tasarımı için Modbus Protokolü kullanmıştır. STM32F103RBT6 (32 bitlik yüksek performanslı, ARM tabanlı mikrodenetleyici) transformatör özelliklerini test etmek için kullanılmıştır. VC6.0 yazılımı ve Microsoft Access bu kontrol sistemini programlamak için kullanılan yazılımlardır.

Bölüm 2'de, Modbus protokolü üzerinde genel bilgiler verilir, Modbus TCP özellikleri belirtilmektedir. Bölüm 3 uygulamada kullanılan yazılımları ve mikrodenetleyicileri açıklamaktadır. Bölüm 4 ise birden fazla dizel jeneratör kontrolü için Modbus protokolü kullanımı dahil olmak üzere dizel jeneratörlerin paralel senkronizasyonu hakkında ayrıntılı bilgi vermektedir. Bölüm 5 jeneratör denetleyicisi istemci (client) ve sunucu (server) sürümü nasıl uygulanacağına dair ayrıntıları açıklamaktadır. Tüm süreci özetleyen akış şeması da bu bölümde verilmiştir.



## **BÖLÜM 2. MODBUS TCP PROTOKOLÜ**

Modbus protokolü denetleyicilerin birbiri ile haberleşmesi için kullanılan yaygın bir haberleşme protokolüdür. Kendine özel mesaj gönderme ve alma çerçeve biçimi dahil olmak üzere hata algılama mekanizmaları vardır. Tüm Modbus protokolü özellikli cihazlar kendi ileti alanları için benzer bir biçim kullanmaktadır. Modbus protokolünde kullanılan verilerin dört çeşiti vardır; Bobin, Ayrık Giriş, Giriş Registeri ve Tutucu Registeri bulunmaktadır [14, 15].

Endüstriyel ortamda Modbus protokolünün kullanılma nedenleri arasında şunlar vardır;

- Kolay kurulum ve bakım
- Endüstriyel uygulamalar göz önüne alınarak geliştirilmesi
- Açık olarak yayınlanması
- Üreticiler konusunda çok fazla sınırlama koymadan ham bit ve kelimeleri (words) gönderebilmesi

### **2.1. Modbus Protokolü Versiyonları**

Modbus protokolünün farklı versiyonlarından bazıları aşağıda listelenmiştir [16].

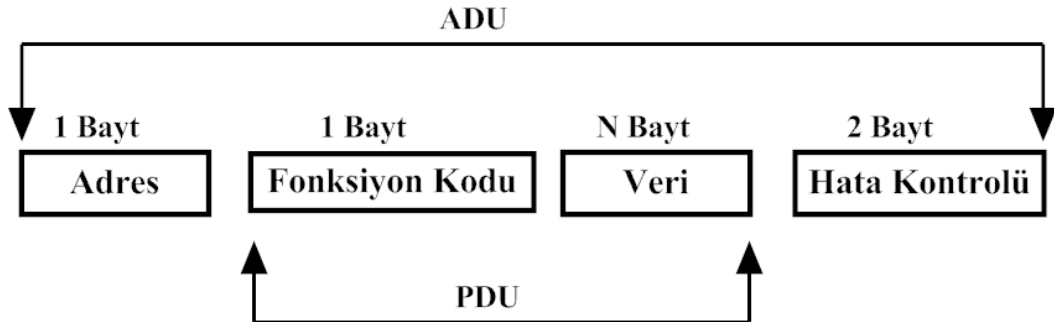
- Modbus ASCII : iletişim için ASCII karakterleri kullanır. Her mesajın başlangıcı ":" veya altıgen 3A karakteri kullanılarak belirtilmektedir. Her mesajı sonlandırmak için ise 0D ve 0A karakterleri kullanılmaktadır.
- Modbus RTU (Remote Terminal Unit) ya da Uzak Terminal Ünitesi : Bu protokol RS-232, RS-485 veya herhangi bir benzer fiziksel arayüze sahip

olan bir seri hat üstünde kullanılmaktadır. Her bayt bir başlangıç biti ve bir dur biti kullanılarak 8 bitlik dizi şeklinde gönderilmektedir.

- Modbus TCP : TCP / IP protokolleri kullanarak Modbus mesajlaşma kullanımını amaçlamaktadır. Genellikle port 502 üzerinden olmakla birlikte TCP/IP ağları üzerinden iletişim yapılmaktadır.
- Modbus UDP : Genelde kullanılan TCP protokolünden daha az ek yük oluşturur ama TCP'ye göre daha az güvenilirdir.

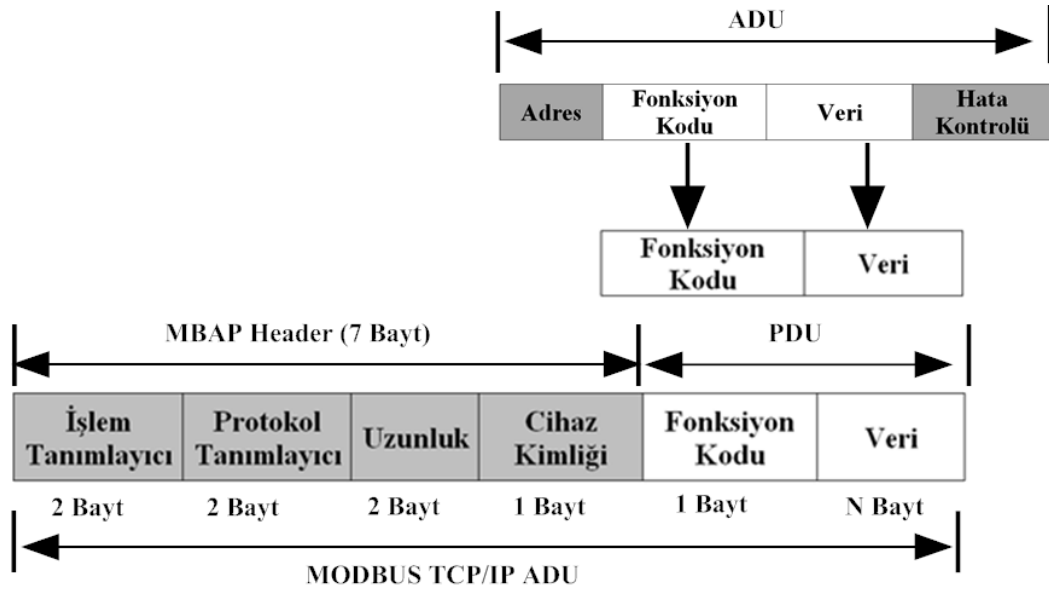
## 2.2. Modbus TCP Protokolü Temelleri

Modbus protokolünde işlem türü ve ana veriyi içeren Protokol Veri Birimi yada Protocol Data Unit (PDU) alt iletişim tabakası tanımlanmaktadır. Ancak, bazı ağlar Uygulama Bilgileri Biriminde ya da Application Data Unit (ADU)'da ekstra alan içerebilmektedir.



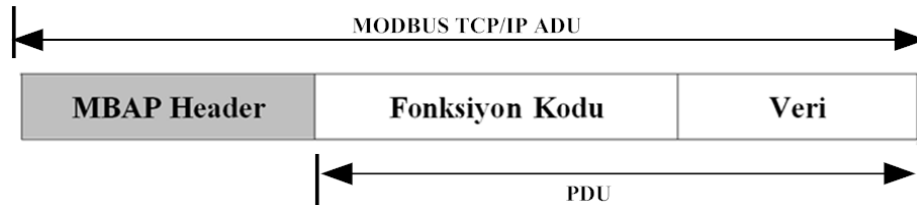
Şekil 2.1. Genel Modbus çerçevesi [14]

Modbus TCP / IP, farklı cihazlar arasında Modbus mesaj yapısını gerçekleştirmek için TCP / IP'yi ve Ethernet ağını kullanmaktadır. Diğer bir deyişle TCP / IP ağ standardı, Modbus protokolü ve bir de Ethernet ağını içermektedir. Modbus TCP/IP mesajı Ethernet TCP/IP ambalajında çerçeveli bir Modbus iletişimidir.



Şekil 2.2. Modbus TCP veri paketin oluşturulması [15]

Modbus TCP / IP ağı üzerinden mesaj gönderim sırası aşağıdaki Şekil 2.3.'teki gibi görünmektedir. TCP/IP Protokolünde Modbus ADU'yu tanımlamak için özel bir MBAP (MODBUS Uygulama Protokolü) başlığı kullanılmaktadır.



Şekil 2.3. TCP/IP üzerinde Modbus istek / yanıt [14]

Modbus TCP/IP Uygulama Veri Birimi (ADU) yukarıdaki Şekil 2.2.'de gösterildiği gibi işlem tanımlayıcı, protokol tanımlayıcı, uzunluk alanı ve birim tanımlayıcı içeren bir 7 baytlık başlığa sahiptir. 7 bayt uzunluğundaki MBAP aşağıdaki alanları içermektedir [15];

- İşlem Tanımlayıcı (Transaction Identifier: 2 Bayt): Sunucuya gönderilen her isteği tanımlamak için client tarafından ayarlanmaktadır. Sunucu client'a bu baytı geri göndermektedir.

- Protokol Tanımlayıcı (Protocol Identifier: 2 Bayt): İstemci tarafından ayarlanır ve her zaman sıfırdır.
- Uzunluk (Length: 2 Bayt): Mesajdaki kendinden sonra gelen bayt sayısını belirler ve birim tanımlayıcıyı, fonksiyon kodunu ve veri alanlarını kapsamaktadır.
- Cihaz Kimliği ya da Tanımlayıcı (Unit Identifier: 1 Bayt): Client'a bağlı bir uzak sunucu / slave'i belirlemek amacıyla client tarafından ayarlanmaktadır.

### 2.3. Modbus Mesajlaşma Fonksiyon Kodları

PDU içinde fonksiyon kodu alanı yapılması gereken görev türünü slave'e bildiren 1 baytı içermektedir. Fonksiyon kodları 1-255 arasında değişir, ancak bu kodların hepsi şu anda kullanımda değildir. Aşağıdaki Tablo 2.1. Modbus protokolü haberleşmede kullanılan fonksiyon kodlarından bazılarını göstermektedir.

Tablo 2.1. Modbus fonksiyon kodları [15]

Modbus Fonksiyon Kodları	İsim
01	Bobin Durumu Oku
02	Giriş Durumu Oku
03	Tutucu Registerı Oku
04	Giriş Registerı Oku
05	Tek Bobin Yaz
06	Tek Register Yaz
16	Birden Fazla Register Yaz

### 2.4. Modbus TCP Avantajları

Modbus TCP / IP protokolü aşağıda özetlendiği gibi birçok avantaja sahiptir;

- Ölçeklenebilirlik: Modbus TCP'yi kullanan cihazların sayısı tek anahtarlı Ethernet ağında iletişim kurabilen 10000 (on bin) ya da daha fazlasına uzanabilmektedir.
- Kolay Geliştirme: Ağa yeni bir istasyon eklenmesi sırasında karmaşık yapılandırma araçlarının kullanımını önlemektedir.

## BÖLÜM 3. UYGULAMA YAZILIMLARI VE CİHAZLARI

### 3.1. Qt C++ Yazılımı

Qt çoğunlukla kod temeline önemli bir değişiklik gerektirmeden çeşitli yazılım ve donanım platformları üzerinde çalıştırılması mümkün yazılımları geliştirmek için kullanılan bir çapraz-platform uygulama yazılımıdır.

Qt; dosyalar, süreçler, ağ ve veritabanı erişimi gibi çeşitli programlama yönlerine uygulanabilir portatif tesisleri kapsayacak şekilde genişletilmiştir. Qt'yi kullanmanın bazı avantajları şunlardır;

- Aynı kod farklı bir platformda derlenebilir.
- Açık kaynak lisansı kullanılabilirliği.

Tablo 3.1. Qt Çerçevesi [17]

Qt SDK		
Sınıf Kütüphaneleri		Geliştirme Araçları
- Core	- XML	- Qt Creator
- GUI	- Multimedia	- GUI Designer
- WebKit	- Database	- I18n Tools
- Graphic View	- Network	- Help System
- Scripting	- Unit Tests	- Build Tool
- OpenGL	- Benchmarking	

Aşağıda listelendiği gibi, Qt birçok platform için mevcuttur;

- Gömülü Linux
- Mac OS X
- Windows
- Linux/X11
- Windows CE
- MeeGo vb.

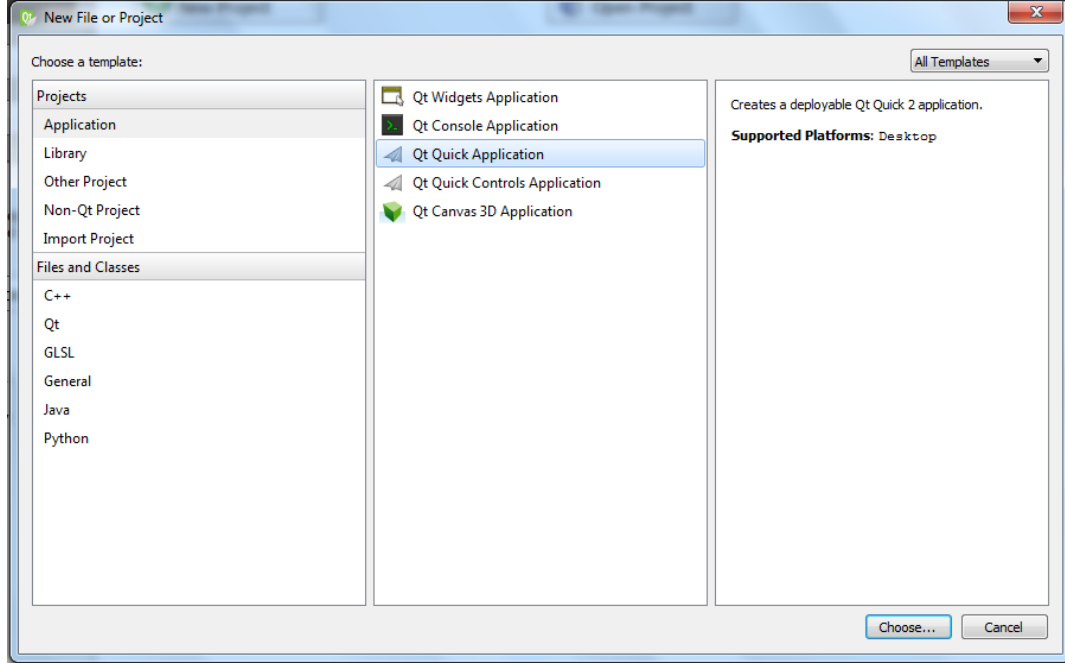
### 3.1.1. Qt Creator

Qt Creator; Qt için varsayılan IDE'dir. MAC, Windows vb. gibi sık sık çoğunlukla kullanılan tüm masaüstü platformlarında kullanılabilir. İnteraktif bir kullanıcı arayüzüne sahip olmakla birlikte aynı zamanda C++ programlama dili ile yazılmış kodları derleyebilmektedir [18]. Qt Creator karşılama sayfası aşağıdaki Şekil 3.1.'de verilmiştir.



Şekil 3.1. Qt Creator karşılama sayfası

Yeni bir proje oluşturmak amacıyla " File → New File or Project " kullanılabilir. Qt Creator; masaüstü ya da mobil uygulamaları oluşturmak için olarak sağlamaktadır. Yaygın olarak kullanılan proje türleri aşağıdaki Şekil 3.2.'de verilmiştir.

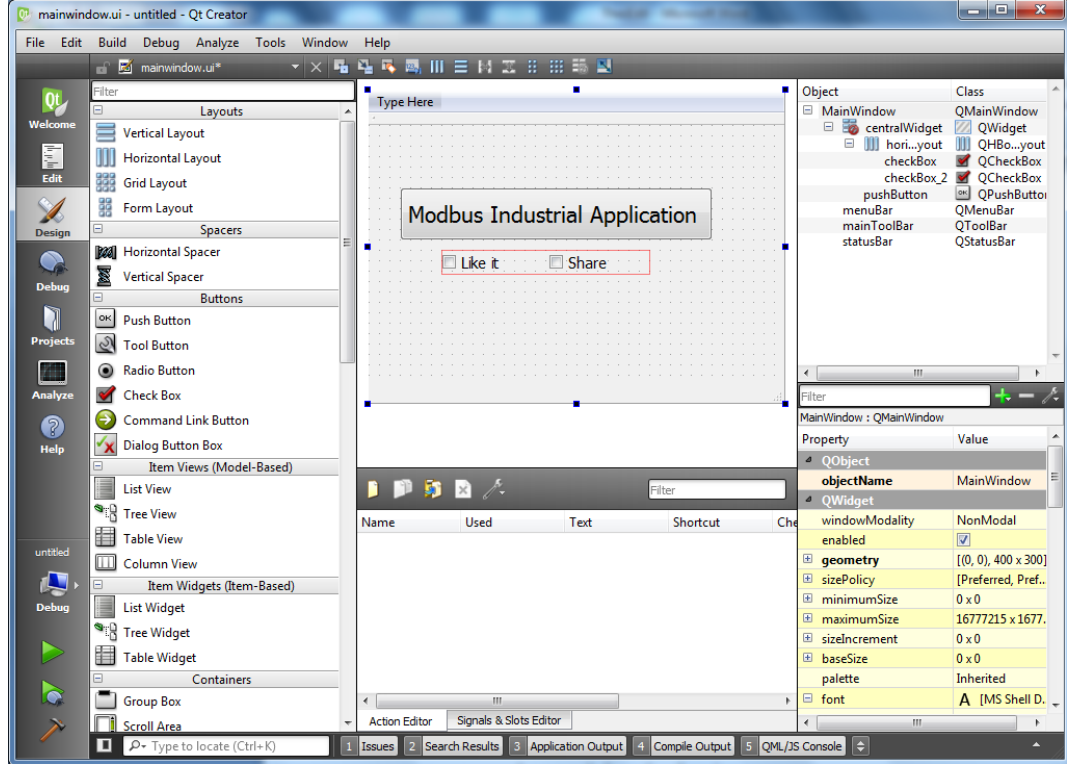


Şekil 3.2. Qt Creator yeni proje sihirbazı

- Qt Widgets: Bir masaüstü uygulaması oluşturmak için bir Qt Designer tabanlı pencere içermektedir.
- Qt Console Application: Bir taslak uygulaması ile bir C++ dosyasını oluşturmaktadır.
- Other Project /Empty qmake Project: Herhangi bir dosya olmadan boş bir proje oluşturulmaktadır. Hiçbir varsayılan sınıflar olmadan kullanıcı, sıfırdan C++ kodu yazmak zorundadır.

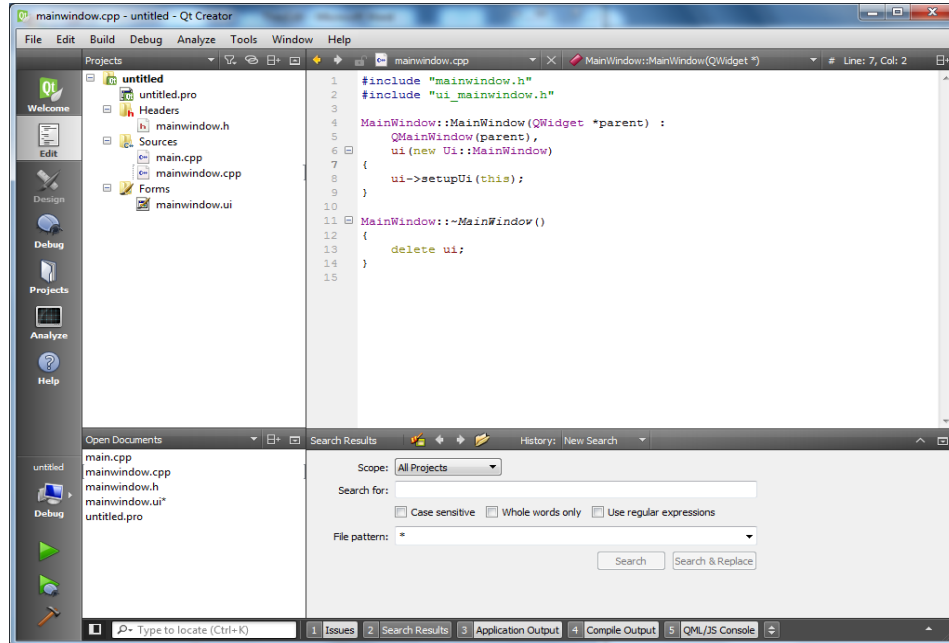
Design Mode (tasarım modu) uygulamaya farklı düğmeler ve işlevleri sürüklemek ve koymak için kullanılabilir. Aşağıdaki Şekil 3.3. tasarım modundaki Qt Creator'ü göstermektedir.





Şekil 3.3. Tasarım modundaki Qt Creator görünümü

Edit Mode (Düzenleme modu) Qt Creator editör penceresinde açılabilir proje dosyalarını göstermektedir. Editör penceresi birçok işlevleri sağlamaktadır. Bunlar arasında kod tamamlama ve hızlı düzeltme seçenekleri vardır. Aşağıdaki Şekil 3.4. düzenleme modundaki Qt Creator'ü göstermektedir.



Şekil 3.4. Düzenleme modundaki Qt Creator görünümü

Qt tarafından sağlanan işlevselliği ile temel C++ kodu, grafik kullanıcı arayüzü (GUI) uygulamaları oluşturmak için kullanılabilir. "Modbus TCP Application" görüntülemek için örnek bir Qt C++ kodu aşağıda verilmiştir.

```
#include <QApplication>
#include <QLabel>
int main(int argc, char *argv[])
{
    QApplication MyApp(argc, argv);
    QLabel *label = new QLabel(" Modbus TCP Application ");
    label->show();
    return MyApp.exec();
}
```

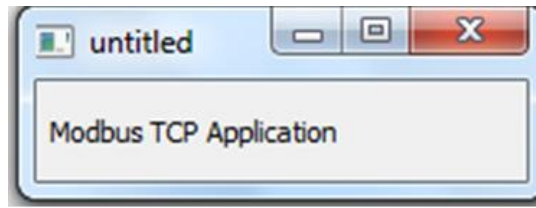
İlk 2 satır sadece sınıfların yani QApplication ve QLabel'ın tanımlarıdır. Tüm Qt sınıflarının aynı isimde olmak üzere başlık dosyaları vardır.

- Uygulama genelindeki kaynakları yönetmek için QApplication nesnesi 5. satırdaki kod ile oluşturulmaktadır. QApplication yapıcı argümanları kendi

başına bir komut satırı desteklemesi için argc ve argv argümanlarına ihtiyacı vardır.

- "Modbus TCP Application" gösteren QLabel 6. satır çalıştırıldığında oluşturulmaktadır.
- 7. satır çalıştırıldığında etiket görünür hale gelmektedir.
- 8. Satır çalıştırıldığında ise olay döngüsüne girmektedir.

Bu basit programın sonucu aşağıdaki Şekil 3.5.'te verilmiştir.



Şekil 3.5. Program sonucu

### 3.2. CodeWarrior

CodeWarrior geliştirme ortamı en karmaşık gömülü uygulamaların geliştirilmesini hızlandırmak için birçok görsel ve otomatik çerçeve sunan bir IDE'dir [19]. Freescale Kinetis gibi çeşitli mikrodenetleyiciler ve mikroişlemciler üzerine programlama ve hata ayıklama uygulamaları yapmak amacıyla Freescale Semiconductor tarafından geliştirilmiştir.

#### 3.2.1. Proje türleri

Yeni CodeWarrior projeleri 'New Project Wizard' kullanılarak oluşturulabilmektedir. Bu adımda dosyalar ve ayarlar; build ve launch konfigürasyonu halinde gruplandırılmaktadır. Aşağıdakiler oluşturulabilen projeler arasında yer almaktadır;

- MQX-Lite Projeleri: Çok hafif MQX çekirdeğe dayalı oluşturulan projelerdir.
- Bareboard Projeleri: Bir hedef tahtası doğrudan konuşlanmış projeler; Linux işletim sistemi olmadan oluşturulmaktadır.

- Linux Projeleri: Linux işletim sistemine dayalı uygulamalar oluşturmak için kullanılmaktadır.

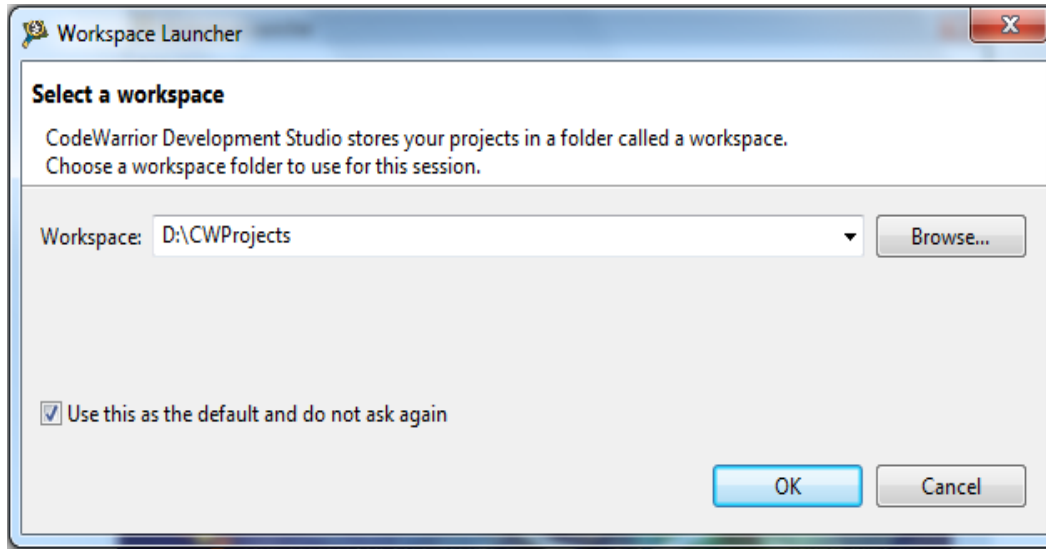
### 3.2.2. Proje oluşturma

Freescall Coldfire ve Kinetis gibi farklı işlemciler için yeni Bare Kurulu ve Linux / uClinux Uygulama projeleri oluşturmak için kullanılan adımlar aşağıda verilmiştir.

#### 3.2.2.1. Çalışma tezgahı başlatma

Aşağıdaki adımlar herhangi bir uygulamayı geliştirmek amacıyla CodeWarrior IDE'yi başlatmak için kullanılmaktadır.

- Adım 1: CodeWarrior yazılımının sürümü açıldığında bir çalışma alanının seçilmesini istemektedir (Şekil 3.6.).



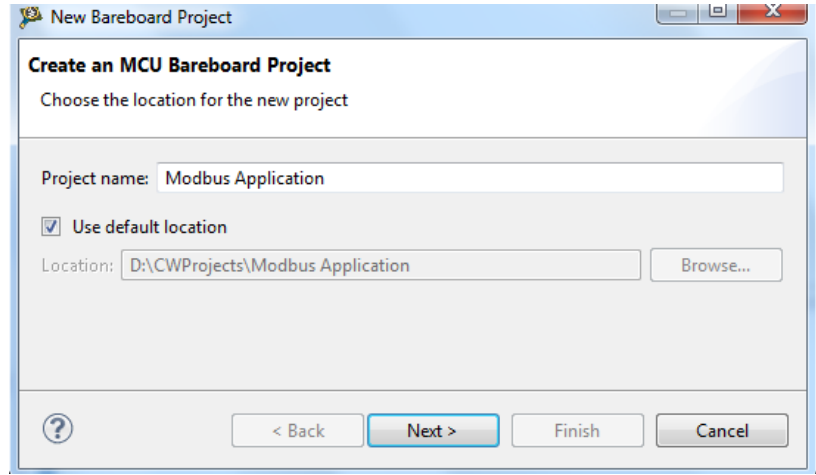
Şekil 3.6. Workspace Launcher iletişim kutusu

- Adım 2: Amaçlanan projeyi başka bir yere kaydetmek için Browse'ı tıklayın ve bir çalışma alanı seçin.
- Adım 3: OK düğmesine tıkladığında Hoşgeldiniz sayfası görünmektedir.

### 3.2.2.2. Kinetis hedef kurulu projesinin oluşturulması

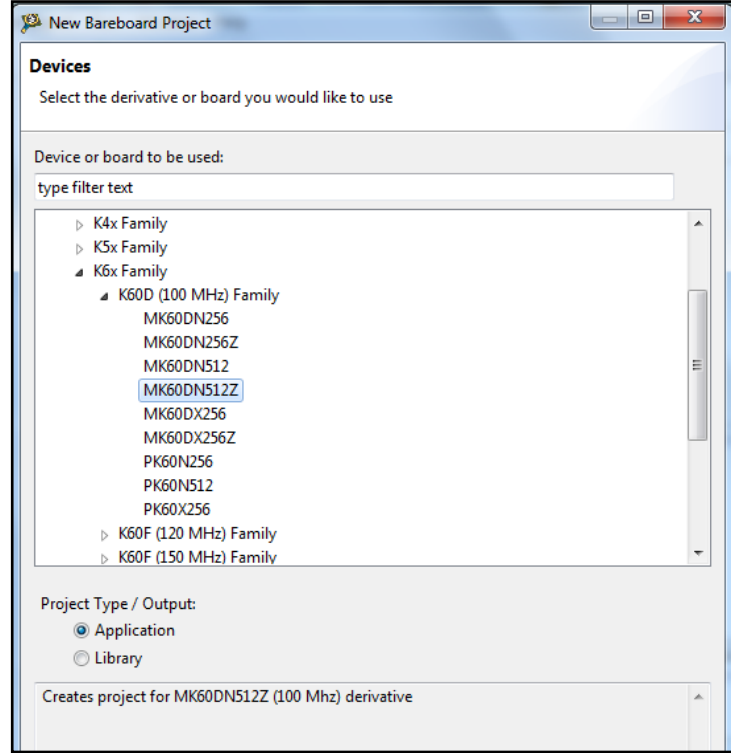
Yeni bir bareboard kullanılarak Freescale Kinetis üzerine yeni bir hedef karta proje oluşturmak için:

- Adım 1: Yukarıda tarif edildiği gibi CodeWarrior uygulama programını başlatın.
- Adım 2: File / New Menu'den Bareboard Project'ı seçin. MCU bareboard proje oluşturma sayfası açılmaktadır.
- Adım 3: Projeye bir ad belirtin (Şekil 3.7.).



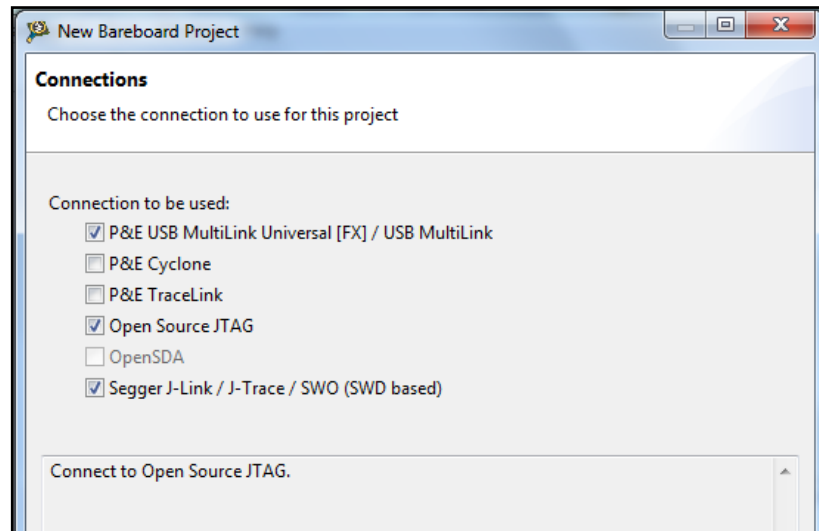
Şekil 3.7. Yeni proje sihirbazı (MCU projesi)

- Adım 4: Cihazlar seçim sayfasına gidin ve seçenekleri uzun bir liste içinden işlemciyi seçmek için Next'ı seçin (Şekil 3.8.). Kinetis K Serisi için örneğin K6x sonra MK60DN512Z en sonunda da K60D (100 MHz) ailesini seçin.



Şekil 3.8. Yeni proje sihirbazı (Hedef işlemci seçimi)

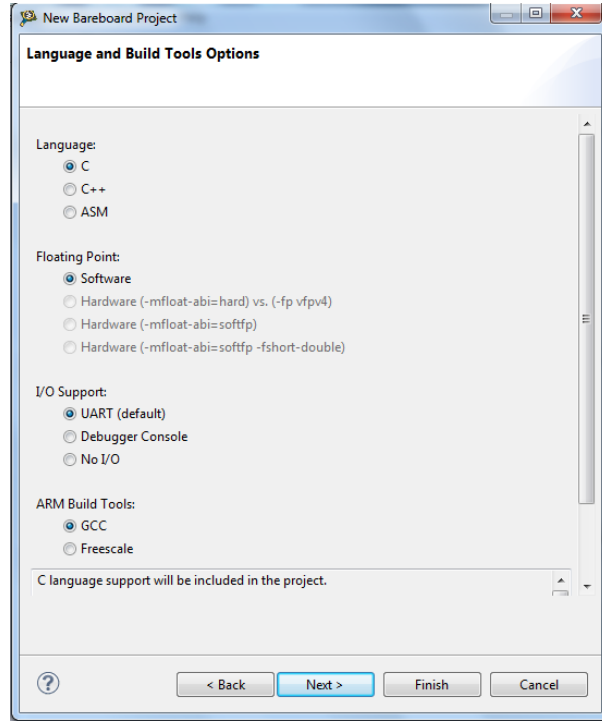
- Adım 5: Connections (bağlantılar) sayfası Next seçildikten sonra görüntülenmektedir (Şekil 3.9.).



Şekil 3.9. Yeni proje sihirbazı (Bağlantı türü)

Onboard açık kaynak JTAG kullanmak için OSJTAG kutusunu işaretleyin. Harici J-Link Segger de hedef tahtasına program yazmak için kullanılabilir. Birden fazla seçenek yapılabilir.

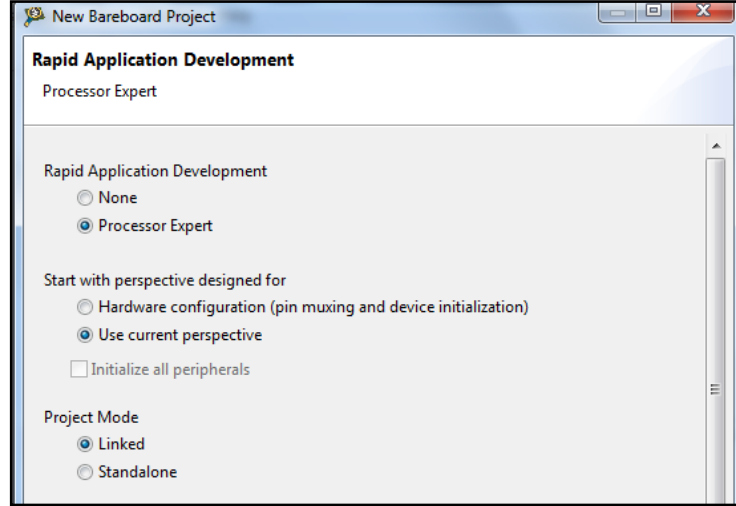
- Adım 7: Programlamak için kullanılan dil de bir sonraki pencereden seçilebilir (Şekil 3.10.).



Şekil 3.10. Yeni proje sihirbazı (Dil ve derleme araçları seçimi)

UART gibi bir I/O ve derleyici seçeneği bu pencereden işaretlenebilir.

- Adım 8: Bu adımda Şekil 3.11.'de gösterildiği gibi başlangıç koduna gerek olup olmadığı belirtilmektedir.



Şekil 3.11. Yeni proje sihirbazı (Hızlı uygulama geliştirme)

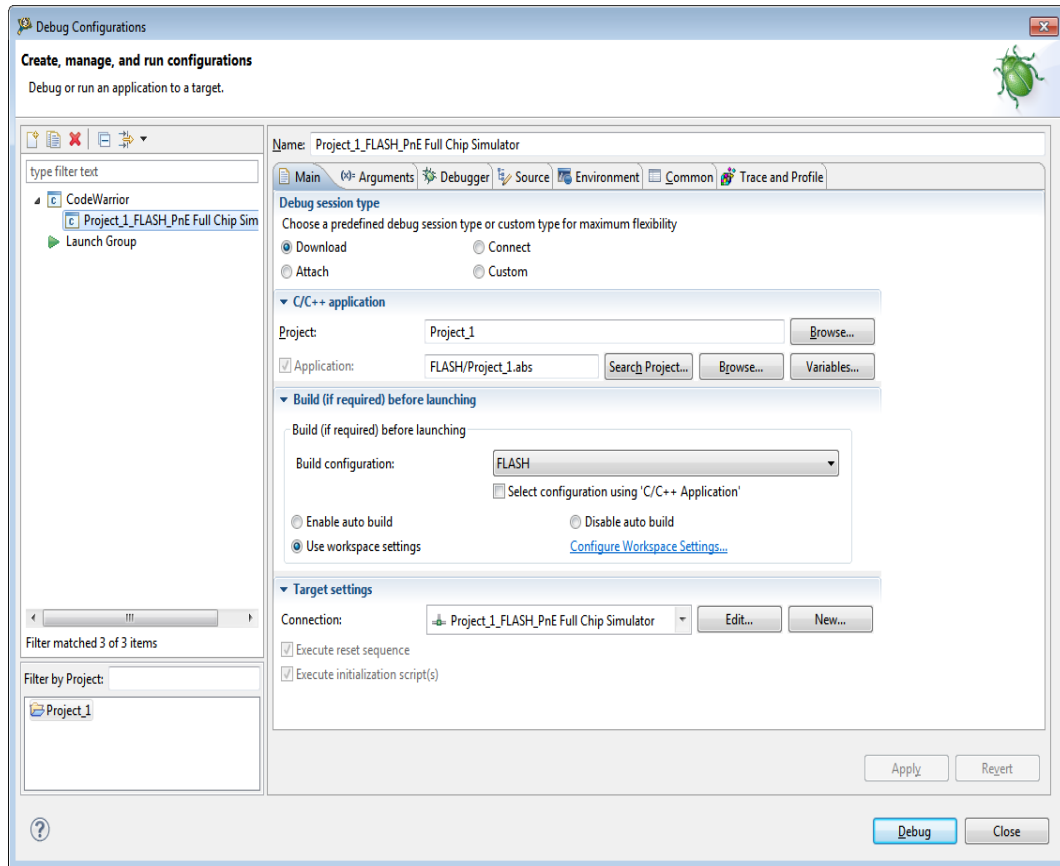
### 3.2.2.3. Proje hata ayıklaması

CodeWarrior IDE aşağıdaki ayıklayıcı fırlatma yapılandırmalarını oluşturmaktadır:

- Debug Konfigürasyonu
- Release Konfigürasyonu

Hata ayıklamak için CW projesinin ana menüsünden Run / Debug Configurations'ı seçin. Hata Ayıklama Konfigürasyonu ayarları iletişim kutusu aşağıdaki Şekil 3.12.'de gösterilmiştir.





Şekil 3.12. Debug konfigürasyonu

### 3.3. Freescale Kinetis

Freescale ağırlıklı olarak ARM teknolojisine dayalı en önemli gömülü sistemler kontrol solüsyonu sağlayıcılarından biridir. Freescale mikrodenetleyici birimleri ve dijital ağ işlemciler aşağıdaki avantajları sağlamaktadır;

- Yüksek performans
- Güç verimliliği vb.

Freescale Coldfire, Freescale Kinetis ve Freescale Dragonball Freescale ürünleri arasındadır.

Tablo 3.2. Kinetis ailesi ürünleri [20]

Kinetis İşlemci Tipi	Özellikler
Kinetis E-Serisi	- 5V / Sağlam - Cortex-M0+/M4 MCU
Kinetis EA-Serisi	- Otomotiv - Cortex-M0+ MCU
Kinetis K-Serisi	- Performans ve entegrasyon - Cortex-M4 MCU
Kinetis L-Serisi	- Ultra-düşük güç - Cortex-M0+ MCU
Kinetis M-Serisi	- Ölçüm bilimi - Cortex-M0+ MCU
Kinetis MİNİ-Serisi	- Küçük çip ölçekli paketler - Dünyanın en küçük ARM tabanlı MCU
Kinetis V-Serisi	- Gerçek zaman kontrol sistemleri - Motor ve güç dönüşümü - Cortex-M0+/M4/M7 MCU
Kinetis W-Serisi	- Kablosuz bağlantı - Cortex-M0+/M4 MCU

Tüm Kinetis ailelerinin ortak özellikleri şunlardır [21];

- 1.25 DMIPS / MHz sağlayan ARM Cortex-M4
- Yüksek hızlı 16-bitlik ADC.
- 32 KB flaştan 1 MB flaşa veya 128 KB RAM'dan 8 KB RAM'a kadar ölçeklenebilir hafıza.
- 12-bitlik DAC.
- Periferik aktivite ve uzatılmış pil ömrü için çok düşük güçlü çalışma modu
- UART gibi seri arabirimleri
- Bağımsız flaş bankalar
- Bellek içeriklerini doğrulamak için kullanılan CRC
- 5 V toleranslı genel amaçlı giriş ve çıkış

### 3.3.1. Kinetis K ailesi mikrodenetleyici birimleri

Kinetis K Ailesi Mikrodenetleyici Birimleri ölçeklenebilir performans ve bağlantı dahil olmak üzere birçok özellik sunmak için üretilmiştir.

Kinetis K-Serisi denetleyiciler aşağıda listelenen özellikler de bulunmaktadır [21];

- Geniş ölçeklenebilirliği ve uyumluluğu
- 50-180 MHz
- 32KB-2MBFlaş
- 256 KB SRAM
- Noktalı sayı işlem birimi
- Güvenlik, analog ve seri arabirimi

### 3.4. TWR-K60D100M

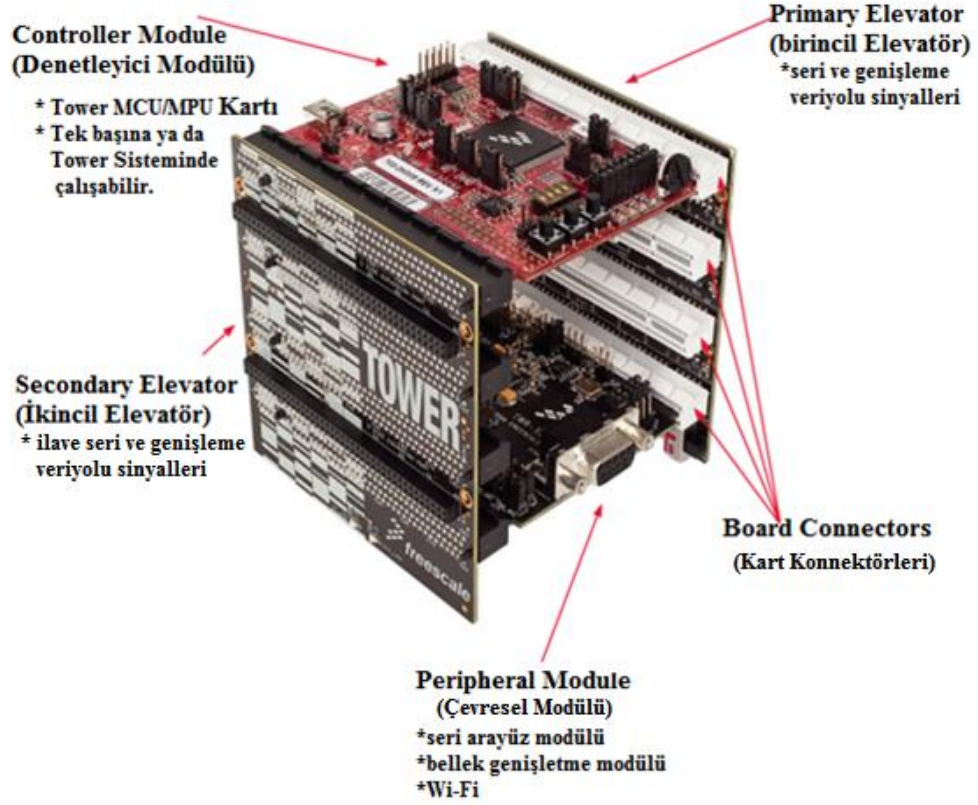
TWR-K60D100M Freescale Tower Sistemi ile uyumlu çalışabilen ileri tower denetleyici modülüdür. K10, K20 ve K60 aile MCU'lar gibi Kinetis K serisi işlemcilerinin değerlendirilmesi bu tower sistemi üzerinde yapılabilmektedir. TWR-K60D100M özellikleri şunlardır [22];

- Düşük güç tüketimi
- ARM cortex-M4 ile tam hızlı OTG denetleyicisi
- 10/100 Mbps Ethernet MAC.

Tek başına bir cihaz olarak kullanılmadığında, TWR-K60D100M aşağıdaki cihazlar ile birleştirilmektedir [22];

- Tower Asansör Modülleri (TWR-ELEV)
- Tower Seri Modülü (TWR-SER).

Aynı zamanda çeşitli uygulamalarda platformları geliştirmek için diğer çevresel tower modüllerine entegre edilebilmektedir. Freescale Tower Sistemi aşağıdaki 3.13. şekilde gösterilmiştir.



Şekil 3.13. TWR-K60D100M [22]

TWR-K60D100M kiti şunları içermektedir;

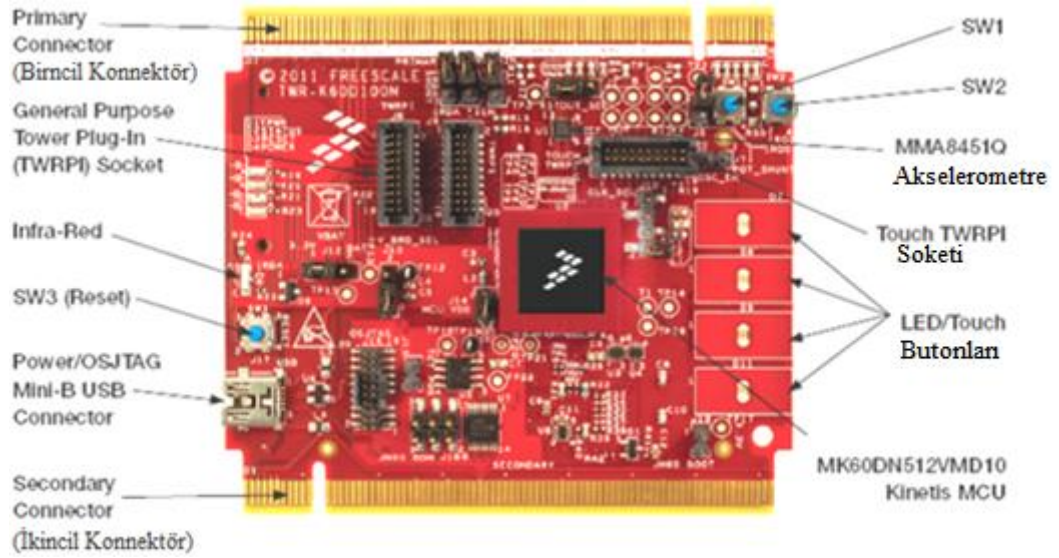
- TWR-K60D100M MCU modülü
- TWR-ELEV - hem birincil hem de ikincil asansör modülü
- TWR-SER - Ethernet, RS232, RS485 vb. içeren seri modülü

TWR-K60D100M tower MCU Modülü özellikleri aşağıdaki gibi özetlenmiştir [22];

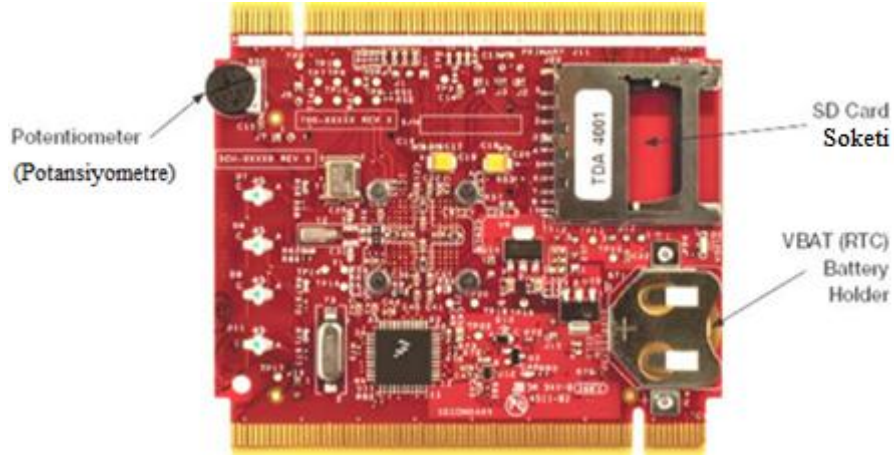
- Tower uyumlu mikroişlemci modülü
- MK60DN512VMD10
- Tower Plug-in dokunmatik soketi
- Genel amaçlı Tower Plug-in (TWRPI) soketi

- Sanal seri port ile On-board JTAG hata ayıklama devresi (OSJTAG)
- Üç eksenli akselerometre (MMA78451Q)
- Dört kullanıcı kontrollü LED
- Dört kapasitif dokunmatik ped
- İki kullanıcı buton anahtarı
- Potansiyometre
- 20mm lityum pil için Pil Tutucu (örneğin 2032, 2025)
- SD Kart yuvası
- 3.3V veya 1.8V çalışma gerilimi
- Düşük güç ölçümleri için izolasyon

Aşağıdaki Şekil 3.14. ve 3.15. TWR-K60D100M MCU modülünün ön ve arka tarafını göstermektedir.



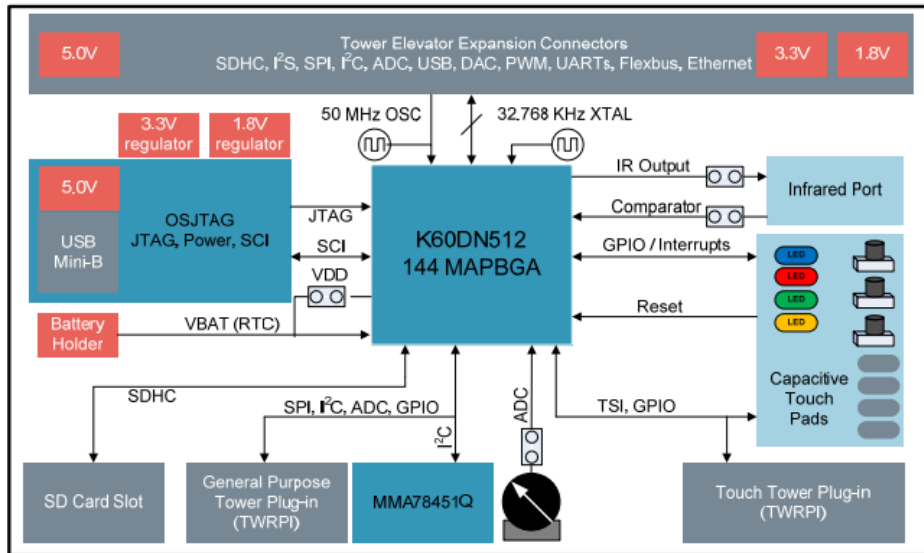
Şekil 3.14. TWR-K60D100M MCU modülü ön tarafı [22]



Şekil 3.15. TWR-K60D100M MCU modülü arka tarafı [22]

### 3.4.1. TWR-K60D100M donanım tanımlaması

TWR-K60D100M MK60DN512VMD10 ARM Cortex-M4 tabanlı bir mikroişlemcidir. USB konnektör ile giriş güç kaynağı ve harici JTAG hata ayıklama arayüzü sağlamaktadır. Ayrıca onboard hata ayıklama devresi olan açık kaynak JTAG'ı (OSJTAG) da içermektedir. Aşağıdaki Şekil 3.16. TWR-K60D100M blok diyagramını göstermektedir.



Şekil 3.16. TWR-K60D100M blok diyagramı [22]

TWR-K60D100M modülü MK60DN512VMD10 özelliklerine sahiptir. Temel özellikleri aşağıda listelenmiştir [22]:

- 32-bit ARM Cortex-M4 ile DSP talimatları
- 100MHz maksimum çekirdek çalışma frekansı
- 144 MAPBGA, 13mm x 13mm, 1.0mm saha paketi
- 1.71V - 3.6V çalışma voltajı
- 512 KB flaş
- 128 KB statik RAM
- PLL ve FLL çalışma modları
- 16-bit SAR ADC
- 12-bit DAC
- Programlanabilir voltaj referans
- SPI, I2C, UART, CAN, I2S
- Hata ayıklama arayüzleri: JTAG, cJTAG, SWD
- 10/100 Mbps Ethernet MAC vb.

### 3.4.2. Sistem gücü

Bağımsız bir modül olarak çalıştığında TWR-K60D100M modülü için güç kaynağı USB konektörü kullanılarak elde edilebilmektedir . Kinetis K60 Gerçek Zamanlı Saat (RTC) modülü için iki çalışma modu bulunmaktadır [22]:

- Sistem güç çekim modu
- Sistem kapanma modu: RTC gücü VBAT'tan elde etmektedir.

### 3.4.3. Hata ayıklama arabirimi

Hedef işlemci üzerine bir program dağıtmak için kullanılan hata ayıklama arayüz seçenekleri şunlardır [22]:

- On-board Açık Kaynak JTAG (OSJTAG) devresi: K60D100M üzerindeki on-board açık kaynak JTAG (OSJTAG) devresi bir JTAG hata ayıklama arabirimini sağlamaktadır.

- Dış Cortex hata ayıklama ve ETM konektörü: Kinetis K60 cihazda mevcut olan SWD, JTAG, ETM vb. iz sinyallerine erişebilen 20 pinli bir konektördür.



## **BÖLÜM 4. MODBUS İLE DİZEL JENERATÖR KONTROLÜ**

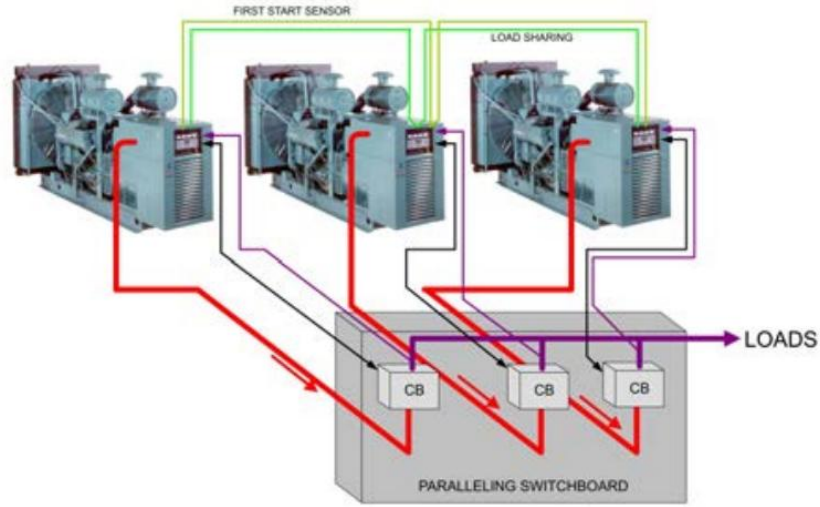
### **4.1. Dizel Jeneratörlerin Paralel Senkronizasyonu**

Sanayilerde kullanılan jeneratörlerin paralel bağlanmasına ilişkin çalışmalar, günümüzde yoğun olarak kullanılan elektronik cihazların sayıca artması nedeniyle kullanılmakta olan teknoloji sistemlerin daha dinamik olması gerekliliğini ortaya koymuştur.

Jeneratörlerin paralel bağlanma işlemini gerçekleştirebilmek için aşağıda verilen şartların sağlanması gerekmektedir [23, 24, 25].

- Eşit frekanslı
- Aynı faz sıraları
- Eşit gerilimli ve benzer gerilim dalga şekilleri
- Eşit faz açıları olmalıdır.

Günümüzde, bir jeneratörün genel ayarlarını izlemek ve kontrol etmek için jeneratör denetleyicileri kullanarak paralel jeneratörleri çalıştırmak çok daha basit hale gelmiştir. Her jeneratör fazını genel sisteme senkronize etmek çok önemlidir. Aşağıdaki Şekil 4.1. paralel jeneratörler arasındaki bağlantıyı göstermektedir.



Şekil 4.1. Ortak yola bağlı paralel jeneratörler [26]

Paralel jeneratörlerin avantajları şunlardır;

- Geliştirilmiş güvenilirlik
- Geliştirilmiş performans
- Kolay hizmet
- Az maliyetli

Bizim durumumuzda, jeneratör denetleyicisi aynı bara paralel olarak bağlanan diğer jeneratörleri kontrol edebilen ve izleme kabiliyetine sahip olan bir master ya da duruma göre parametrelerini ayarlayabilen bir slave olabilmektedir. Bu ağ içinde herhangi bir düğüme Ethernet Modbus TCP / IP bağlantısı aracılığıyla da işleme ve aktarma bilgileri iletilebilmektedir.

#### 4.2. Modbus ile Birden Fazla Dizel Jeneratör Kontrolü

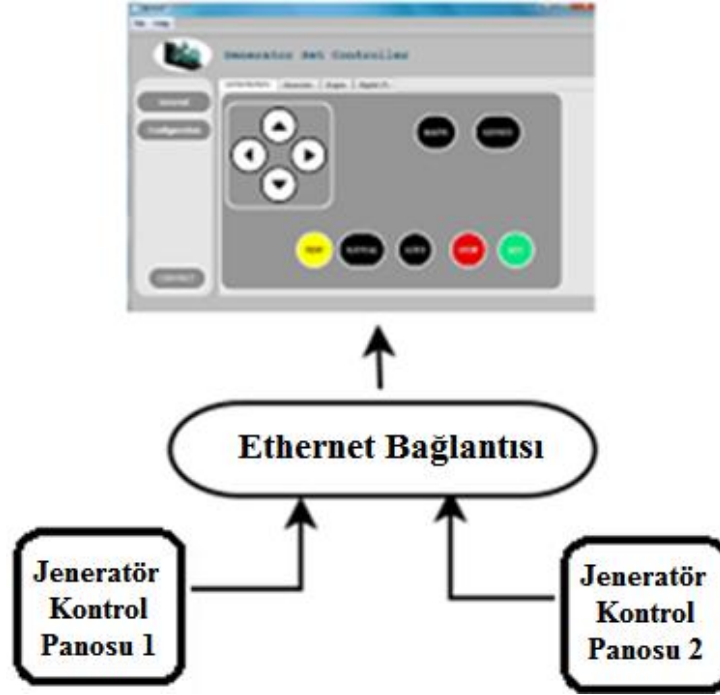
Modbus TCP / IP elektronik cihazlar arasında bir iletişim oluşturmak için sağlam bir protokoldür. Dizel jeneratör denetleyicisi tasarımında ilk adım, gerçek kontrol panellerini taklit eden Modbus TCP protokolünü kullanan bir sanal slave / sunucu programı oluşturmaktır. Bu şekilde kendisine bizim master/client programını

bağlayarak okuma / yazma kayıtları gibi tüm işlemlerinin uygulaması Şekil 4.2.'de gösterilmiştir.



Şekil 4.2. İstemci ve sunucu Modbus TCP / IP bağlantısı emülasyonu

Aşağıdaki Şekil 4.3.'te iki jeneratör kontrol paneli ile Qt C++ kullanarak tasarlanan Modbus TCP / IP jeneratör kontrol ve senkronizasyon programı arasındaki Ethernet bağlantısı anlatılmaktadır. Bu çalışmada tasarlanan program ve kontrol paneli arasındaki Modbus TCP bağlantısı üzerinden jeneratörlerin motor hızı, yakıt seviyesi, yağ sıcaklığı ve LN gerilimleri vb. gibi jeneratörlerin gerçek zamanlı durumu elde edilebilmekte ve uzaktan jeneratör kontrolü sağlanılabilmektedir.



Sekil 4.3. Tasarlanan denetleyici programı ve jeneratör kontrol panelleri arasında Ethernet üzerinden Modbus bağlantısı

Tasarlanan arayüzünün bazı özellikleri şunlardır;

- Veritabanına veya cihaza kaydedilebilme ya da yazılabilmektedir.
- Dosya veya cihazdan okunabilmektedir.

## **BÖLÜM 5. MODBUS TCP CLİENT VE SERVER UYGULAMASI**

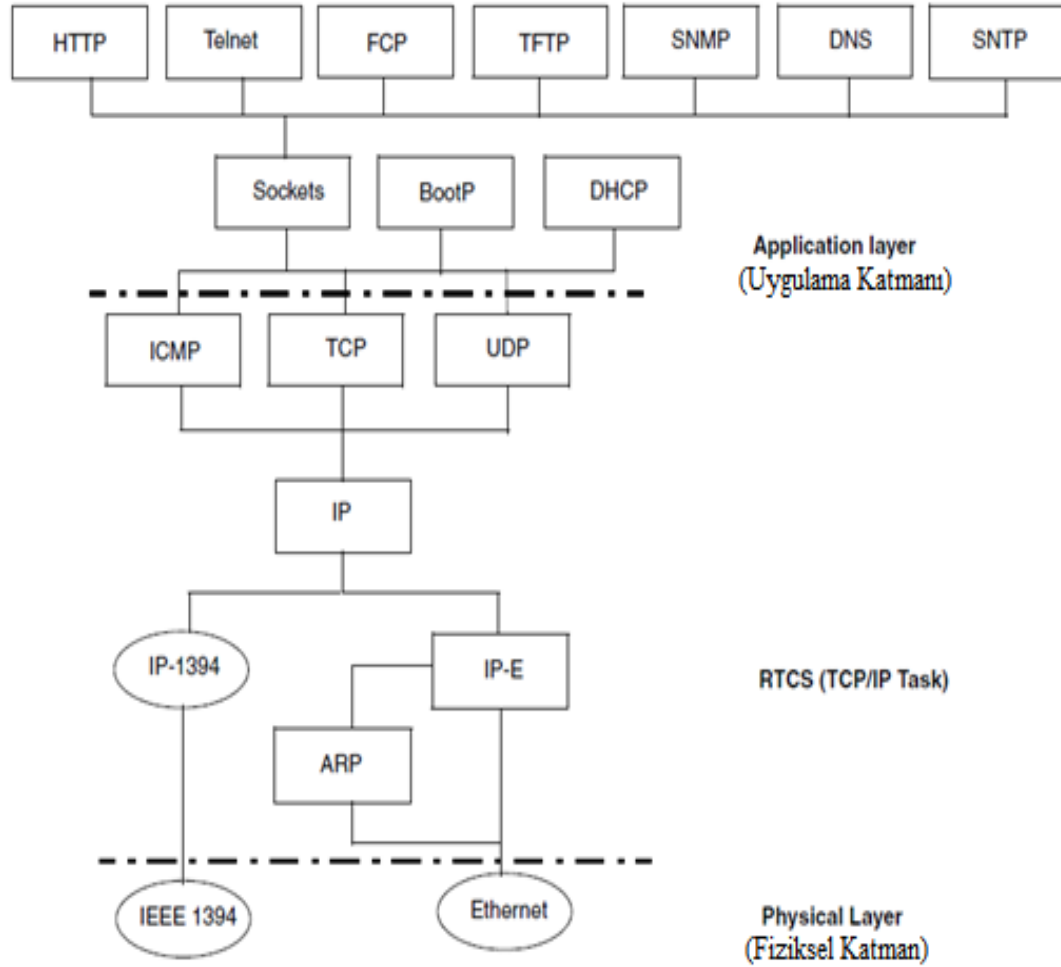
### **5.1. MQX**

MQX RTOS öncelikli zamanlama ve hızlı kesme tepkisi ile geliştirilen gerçek zamanlı işletim sistemidir. Bir Modbus TCP istemci / sunucu Freescale Kinetis K-Serisi İşlemcisi (K-60) üzerinde uygulanan gerçek zamanlı sistemler için MQX RTOS kullanılmıştır. MQX yazılım kütüphanesi olarak herhangi bir projeye entegre edilebilen RTCS ağ yığını, MFS dosyaları sistemi ve USB host / aygıt sürücülerini gibi çekirdek sistem bileşenlerini içermektedir.

MQX yazılım kütüphaneleri aşağıdakileri içermektedir [27, 28, 29, 30,31];

- RTCS ağ yığını
- Kabuk arabirimi kütüphanesi
- USB (Sunucu ve Aygıt) sürücülerini
- MS-DOS Dosya Sistemi Kütüphanesi (MFS)

Real-Time TCP / IP Communication Suite (RTCS) HTTP, Telnet, FTP ve DHCP gibi uygulamaları desteklemek için farklı protokoller oluşturan MQX RTOS çalıştırmak için optimize edilen gömülü Ethernet yığınıdır. Aşağıdaki Şekil 5.1. RTCS'nin içerdiği farklı katmanları ve protokolleri göstermektedir.



Şekil 5.1. RTCS katmanları ve protokolleri [27]

## 5.2. Genel Jeneratör Kontrol Sistemleri

Jeneratör kontrol sistemleri genellikle Otomatik Şebeke Arıza Kontrol Modülleri, Kesici Kontrolörleri, Alternatör İzleme Modülleri, Motor Kontrol ve İzleme Modülleri ve Seri Veri Arayüzler Modülleri içeren akıllı elektronik kontrolörleri kapsamaktadır. Bazı jeneratör kontrol sistemlerinde EKÜ (Elektronik Kontrol Ünitesi) ile donatılmış motorlar ile haberleşmek için RS485 ve CANBUS arabirimi bulunmaktadır. Basit jeneratör kontrol sistemlerinde sadece bazıları veya bir tanesi bulunurken kompleks Jeneratör Kontrol Sistemlerinde ise tüm modüller bulunabilmektedir [32].

- Otomatik Şebeke Arıza Kontrol Modülleri: Bu işlem kesintisiz olarak gerilimler, frekans ve faz gücü gibi şebeke parametrelerini izleyebilen elektronik devresinin bir türüdür. İstenirse bunlara seçilen sınırlar programlanabilmektedir. Parametreler belirlenmiş sınırlar dahilinde olmadığında, Şebeke Arıza Kontrol Modülü, örneğin şebeke arıza tespit edilmesi üzerine, kontrolör vasıtasıyla motoru başlatabilmektedir. Ayarlanmış olduğu programlama rutinlerine göre, Şebeke Arıza Kontrol Modülü aynı zamanda otomatik olarak şebeke devre kesicisini kontrol edebilmektedir.
- Kesici Kontrolörü: Bu şebeke ve jeneratörlerde devre kesicilerin durumunu izleyebilen elektronik kart veya elektronik devrenin bir türüdür. Devre kesiciler otomatik olarak şebeke veya jeneratörü yüke bağlamaktadır. Sigortaları içerebilir ve devre kesici bobinleri ve röleleri korumak için tasarlanmıştır. Bir geri besleme mesajı bir devre kesici başarısızlığı oluşması halinde kullanıcıyı bilgilendirmek için otomatik olarak oluşturulabilmektedir.
- Alternatör İzleme Modülleri: Bu modül Alternatör parametrelerini izlemektedir. Bu Şebeke Arıza Kontrol Modülüne benzer temel işlevleri gerçekleştirmektedir. Bu modül Alternatör İzleme Modülü tarafından yapılan ölçümler ile ayarlanan parametreleri (örneğin aşırı akım veya aşırı gerilim) karşılaştırmaktadır. Bu karşılaştırmanın sonucunda duruma göre alarmlar ve uyarılar üretecektir veya kontrollü bir şekilde Alternatör Devre Kesiciyi çalıştırmaktadır.
- Motor Kontrol ve İzleme Modülleri: Bu sistemin çekirdeğidir. Güvenli ve otomatik bir şekilde motoru kontrol etmek için gerekli tüm temel fonksiyonları kapsamaktadır. İç mikrodenetleyici kullanıcı istekleri (yerel butonlar veya uzaktan kumandalar), tüm girişler ve yardımcı iç modüller isteklerini sürekli birleştirmektedir. Her saniye yüzlerce kez yapılan ölçümlerle jeneratörün dahili ayarlarını karşılaştırmaktadır. Bu sistemin verimli bir şekilde çalışması için çıkış bağlantıları ve bağlantı noktalarında komutlar sunulmaktadır. Bir motor kontrol ünitesi veya engine control unit (ECU) aşadaki gibi işlevsel bloklar içerebilmektedir [33].

- Güç Kaynağı - dijital ve analog (analog sensörler için güç)
  - MPU - Mikroişlemci ve hafıza (genellikle Flaş ve RAM)
  - İletişim Linki - (örneğin CAN-Bus)
  - Sayısal Girişler
  - Frekans Girişler
  - Analog Girişler - sensörlerden gelen geri besleme sinyalleri
  - Anahtar Çıkışlar
  - PWM Çıkışlar
  - Frekans Çıkışlar
- Seri Veri Arayüz Modülü : Bu Modbus protokolü üzerinden, bir bilgisayar kullanarak Jeneratör Kontrol Sistemi ile iletişim kurmasını sağlayan RS485 endüstriyel arabirimi olabilmektedir. Aynı zamanda uyumlu motorları bağlamak için tasarlanmış bir CANBUS arayüzü içerebilmektedir. Bugün daha az popüler olsa da bir MODEM kullanıldığında, RS232 seri arabirimi de kullanılmaktadır. Aşağıdaki Şekil 5.2.'de gösterildiği gibi bu çalışmada kullanılan TWR K60D100M modulündeki TWR SER farklı iletişim arabirimleri sağlamaktadır.



Şekil 5.2. Freescale TWR-SER modülü [34]

### 5.3. Modbus Protokol Kütüphanesi

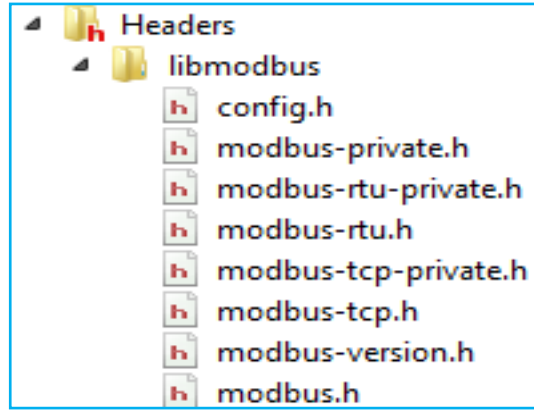
Şu anda kullanımda olan farklı Modbus kütüphaneleri arasında olan Libmodbus Modbus Protokolü veri göndermek ve almak için kullanılan ücretsiz bir yazılım



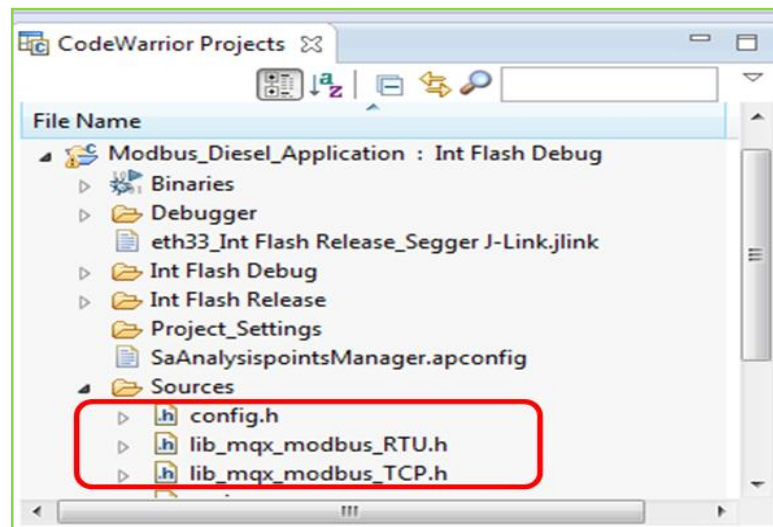
kütüphanesidir. C dilinde yazılmış ve Linux, Mac OS X, FreeBSD, QNX ve Win32 için edinilebilmektedir. Seri RTU ve Ethernet TCP iletişimini desteklemektedir [35].

MQX gerçek zamanlı işletim sistemi kullanıldığı için libmodbus kütüphane işlevlerine değişiklik yapılmıştır. Günümüzde birçok jeneratör kontrol sistemlerinde yaygın olarak kullanılan üç Modbus Protokol fonksiyonları şunlardır;

- Birden fazla register oku (Fonksiyon kodu: 03)
- Tek bir register yaz (Fonksiyon kodu: 06)
- Birden fazla register yaz (Fonksiyon kodu: 16)



Şekil 5.3. Libmodbus Kütüphanesi



Şekil 5.4. Modifiye edilen Modbus kütüphanesi

Tutucu registerleri okumak için modifiye edilen libmodbus C kodu kütüphanesi aşağıda verilmiştir. (İstemci tarafı)

```

Static int read_registers (uint32_t locate,int slave,int function, int addr, int
nb,uint16_t *dest)
{
int rc=0;
int req_length;
uint8_t req[_MIN_REQ_LENGTH];
uint8_t rsp[MAX_MESSAGE_LENGTH];
uint32_t error,option,retval;
option = TRUE;
retval = setsockopt (locate, SOL_TCP, OPT_RECEIVE_NOWAIT, &option,
sizeof(option));
if (nb > MODBUS_MAX_READ_REGISTERS) {
//ERROR Too many registers requested!!
return -1;
}
req_length = modbus_tcp_build_request_basis(slave,function, addr, nb, req);
rc = send(locate, req, req_length, 0);
if(rc==req_length){
int offset;
int i,jj;
option = FALSE;
retval = setsockopt(locate, SOL_TCP, OPT_RECEIVE_NOWAIT, &option,
sizeof(option));
if (retval !=RTCS_OK) {
fputs("[MODBUS_CLIENT]: Fatal Error: Unable to set socket options.",
stderr);
_task_block();
}
rc = recv(locate,rsp, MAX_MESSAGE_LENGTH,0);
if (rc == RTCS_ERROR) {

```

```

        error = RTCS_geterror(locate);
        // Close TCPClient if TCPServer close first.
        if (error == RTCSERR_TCP_CONN_CLOSING) {
            shutdown(locate,FLAG_ABORT_CONNECTION);
            printf("\n[MODBUS_CLIENT]: Closed.");
            return -1;
            // Close TCPClient if TCPServer abort.
        }
    else if (error == RTCSERR_TCP_CONN_RESET) {
        shutdown(locate,FLAG_ABORT_CONNECTION);
        printf("\n[MODBUS_CLIENT]: Connection reset by peer.");
        printf("\n[MODBUS_CLIENT]: Closed.");
        return -1;
    }
    else {
        printf("\n[MODBUS_CLIENT]: Failed to recv, CODE = 0x%x.", error);
        _task_block();
    }
}

jj = modbus_tcp_check_confirmation(req, rsp, rc);
if (jj == -1)
    return -1;
offset = _MODBUS_TCP_HEADER_LENGTH;
for (i = 0; i < nb; i++) {
    /* shift reg high byte to temp OR with low byte */
    dest[i] = (rsp[offset + 2 + (i << 1)] << 8) | rsp[offset + 3 + (i << 1)];
}
}
else{
    rc=-1;
}
memset(rsp, 0, sizeof(rsp));

```

```

return rc;
}

```

İstemcinin talebini yanıtlamak için sunucu tarafından kullanılan kısmi modifiye libmodbus C kodu kütüphanesi aşağıda verilmiştir;

```

static int reply (uint32_t locate,uint16_t *tab_reg, uint16_t nb_reg,uint8_t *req,
uint8_t req_length, uint8_t _UNIT_IDENTIFIER_TCP)
{
uint8_t slave = req[_MODBUS_TCP_UNIT_IDENTIFIER_TCP];
uint8_t function = req[_MODBUS_TCP_FUNCTION];
uint16_t address = (req[_MODBUS_TCP_FUNCTION + 1] << 8) +
req[_MODBUS_TCP_FUNCTION + 2];
uint16_t nb = (req[_MODBUS_TCP_FUNCTION + 3] << 8) +
req[_MODBUS_TCP_FUNCTION + 4];
uint8_t rsp[_MODBUS_TCP_MAX_ADU_LENGTH];
uint8_t rsp_length = 0;
uint8_t rp=0;
if (slave != _UNIT_IDENTIFIER_TCP &&
slave != _BROADCAST_ADDRESS_TCP) {
return rp;
}
if (address + nb > nb_reg) {
rsp_length = response_exception(slave, function,
MODBUS_EXCEPTION_ILLEGAL_DATA_ADDRESS, rsp);
} else {
req_length -= _MODBUS_TCP_CHECKSUM_LENGTH;
if (function == _FC_READ_HOLDING_REGISTERS) {
uint16_t i;
rp=1;
rsp_length = build_response_basis(slave, function, rsp);
rsp[rsp_length++] = nb << 1;
}
}
}

```

```

    for (i = address; i < address + nb; i++) {
        rsp[rsp_length++] = tab_reg[i] >> 8;
        rsp[rsp_length++] = tab_reg[i] & 0xFF;
    }
} else {
    uint16_t i, j;
    rp=2;
    for (i = address, j = 6; i < address + nb; i++, j += 2) {
        /* 6 and 7 = first value */
        tab_reg[i] = (req[_MODBUS_TCP_FUNCTION + j] << 8) +
            req[_MODBUS_TCP_FUNCTION + j + 1];
    }
    rsp_length = build_response_basis(slave, function, rsp);
    /* 4 to copy the address (2) and the no. of registers */
    memcpy(rsp + rsp_length, req + rsp_length, 4);
    rsp_length += 4;
}
}
send_msg(locate,rsp, rsp_length);
return rp;
}

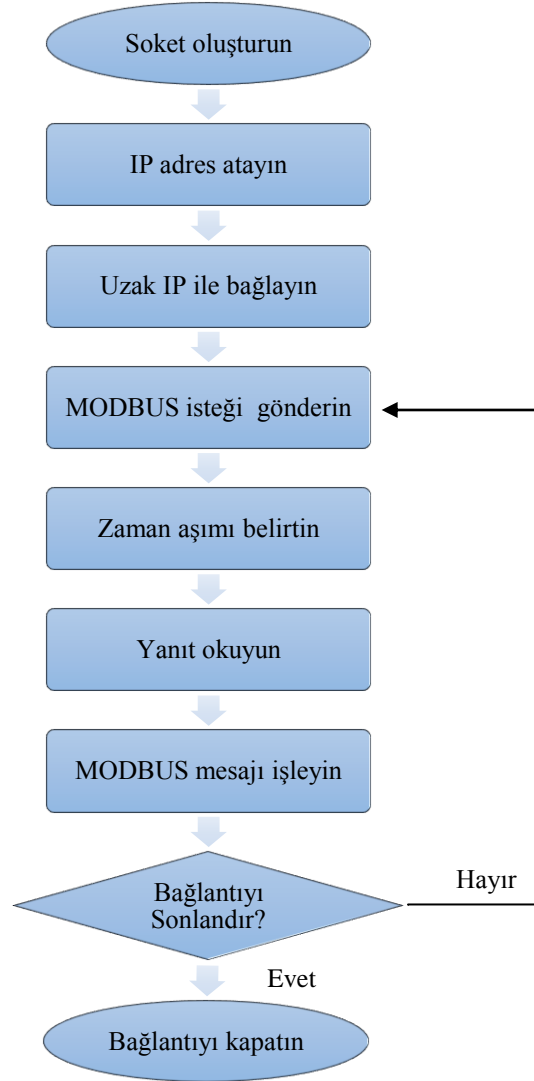
```

#### 5.4. MQX Üzerinde Modbus TCP İstemci Uygulaması

CodeWarrior 10.6 versiyonu K-60 işlemci üzerine Modbus TCP istemci ve sunucu uygulamasını programlamak için kullanılmıştır. Temel bir Modbus TCP istemcisi aşağıdaki gibi uygulanmaktadır [36]:

- Adım 1: socket () işlevini kullanarak bir ilişkisiz soket oluşturun.
  - socket (AF\_INET, SOCK\_STREAM, 0);
- Adım 2: bind () işlevini kullanarak bir soket için yerel soket adresi atayın.
  - bind(sock, &addr, sizeof(addr));

- Adım 3: connect() kullanarak istenilen sunucuya bir bağlantı noktası (genellikle 502) bir TCP bağlantısı kurun.
  - connect(socket\_handle, &RemoteIPAddr, sizeof(RemoteIPAddr));
- Adım 4: send() kullanarak iletilecek olan 6 bayt Modbus TCP öneki içeren bir MODBUS isteği hazırlayın.
- Adım 5: Bir yanıt aynı TCP bağlantısı üzerinde görüntülenmesini bekleyin. Select() fonksiyonunu kullanarak bir zaman aşımı çalıştırın.
- Adım 6: Gelen baytı okumak için recv () işlevini kullanın.
  - Tüm Modbus TCP ile ilgili fonksiyonları gönderme/alma döngüsü içinden çağırılmaktadır.
- Adım 7: Başka bir iletişim bu özel hedefe beklenmiyorsa TCP bağlantısını kapatın.



Şekil 5.5. MQX RTOS üzerinde Modbus istemci uygulaması

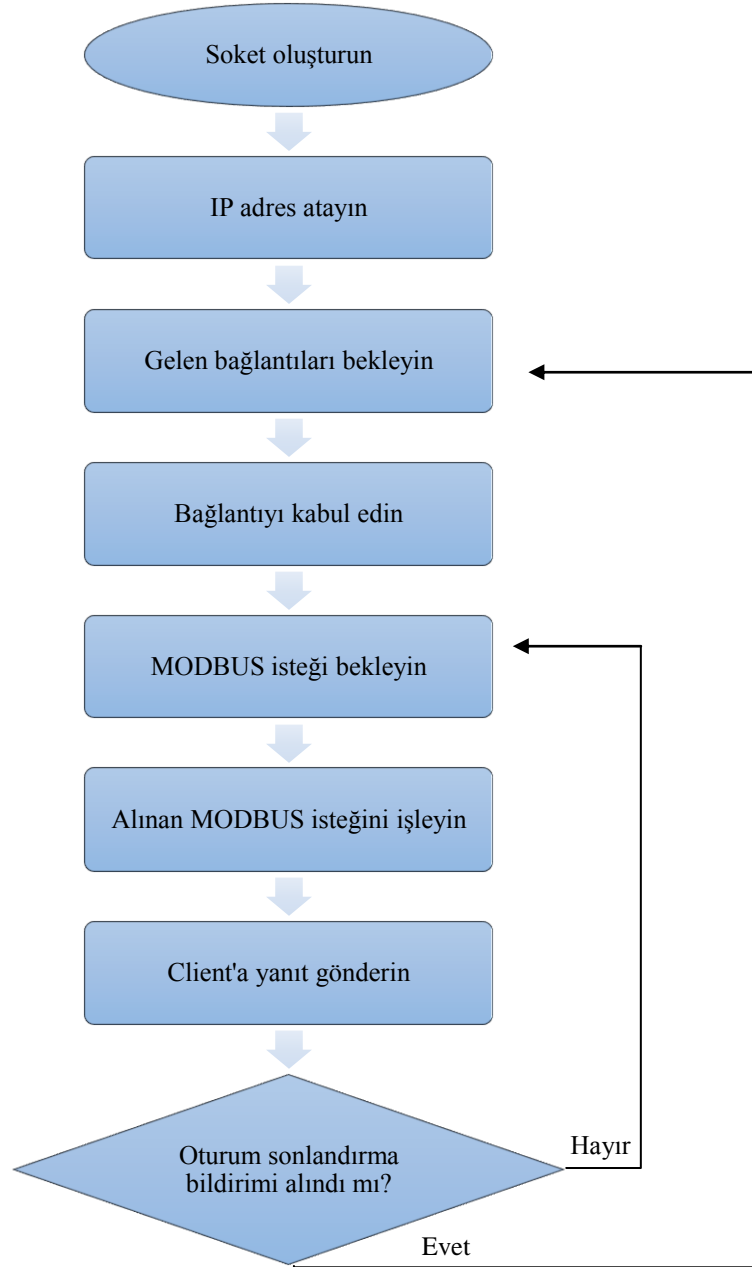
## 5.5. MQX Üzerinde Modbus TCP Sunucu Uygulaması

Bir Modbus TCP sunucu uygulaması aynı anda birden çok istemcileri desteklemek zorundadır. Bazı bir Modbus TCP Sunucusu aşağıdaki gibi yapılmaktadır [36];

- Adım 1: socket () işlevini kullanarak bir ilişkisiz soketi oluşturun.
  - socket\_handle = socket (AF\_INET, SOCK\_STREAM, 0);
- Adım 2: bind () işlevini kullanarak bir soket için yerel soket adresi atayın.
  - bind (socket\_handle & server\_socket\_addr, sizeof (server\_socket\_addr));
- Adım 3: listen() fonksiyonunu kullanarak bir TCP portuna (genellikle 502) gelen bağlantıları bekleyin.

- listen(socket\_handle, 0);
- Gelen bağlantı kabul edilirse yeni bir soketi oluşturun
- Adım 4: bağlantıyı işlemek için accept() işlevini kullanın
  - new\_socket\_handle = accept(socket\_handle & client\_socket\_addr & socket\_addr\_len);
- Adım 5: sonsuz bir döngü içinde aşağıdakileri yapın:
  - Gelen baytı okumak için recv () işlevini kullanın.
  - Cevap için Modbus TCP ön eki oluşturun.
  - Bağlantı iletimi için tek bir tampon olarak, Modbus TCP ön eki dahil send() işlevini kullanarak yanıtı gönderin.
  - Bir sonraki gelen mesaj için bekleyin.
  - Sonsuz döngü prizden okuyun ve yanıtlayın
  - Tüm Modbus ile ilgili fonksiyonlar (Alma-Gönderme) bu döngüden çağrılmaktadır.
  - recv (new\_socket\_handle, buff, RX\_BUFFER\_SIZE, 0);
  - send (DestAddress, data, dataLength);
  - Eğer istemci bağlantıyı sonlandırırorsa bağlantıyı kapatın.





Őekil 5.6. MQX RTOS üzerinde Modbus sunucu uygulaması

## BÖLÜM 6. SONUÇ

Modbus TCP / IP bir endüstriyel Ethernet-TCP / IP ağı üzerinden uygulanan iletişim protokolüdür. Jeneratörlerin paralel olarak çalışması bir sistemin çıkış gerilim dalga formunu başka bir sistemin çıkış gerilim dalga formu ile eşleştirilerek elde edilmektedir. Bu çalışmanın sonucunda uzaktan kontrol ve gerçek zamanlı olarak dizel jeneratörlerin çalışmasını kontrol eden program aracılığıyla izlenilmekte ve kontrol edilebilmektedir.

Modbus TCP / IP protokolünü kullanan K-60 Freescale Kinetis işlemcisini kullanarak bir jeneratör sürücü kartı geliştirilmiştir. CodeWarrior 10.6 versiyonu sunucu / istemci uygulamaları yapmak için kullanılmıştır. Bu Modbus TCP jeneratör denetleyicisi hem istemci hem de sunucu olarak çalışmaktadır. Qt C++ yazılımı ile tasarlanan uygulamayı kullanarak jeneratör denetleyicisi sürücü kartından jeneratörlerin durumu ile ilgili bilgi uzaktan izlenerek alınmaktadır.

Bir istemci olarak, bir Modbus isteği gönderirken cihaz kimliği (seri haberleşme halinde bir slave kimliği olan birim tanımlayıcı), işlem kimliği, fonksiyon türü (okuma / yazma) gibi gerekli tüm bilgileri içermektedir.

Sunucu aldığı bu mesaj ile ilgili tüm bilgileri çıkartarak bu isteği işlemektedir. Modbus kütüphane destek fonksiyonları istemci ve sunucu için istek ve cevap oluşturmak için kullanılmıştır.

Tasarlanan jeneratör kontrol sisteminin bazı özellikleri şunlardır;

- Veri depolamak için bellek tahsisi
- İstemci ve sunucu arasında otomatik geçiş
- Tam donanımlı kütüphane

İstemci modunda, tasarlanan kontrolör birden fazla jeneratöre istekleri gönderebilmekte ve yanıt elde edebilmektedir. Edinilen parametrelerden bazıları şunlardır;

- Ortam sıcaklığı
- Motor devri
- Batarya voltajı
- Jeneratör frekansı
- Jeneratör toplam güç faktörü
- Yağ basıncı
- Motor sıcaklığı
- Yakıt seviyesi vb.

Modbus TCP sunucu durumunda, uzaktan çeşitli parametrelere ayar yapma imkanının yanı sıra mevcut durumu da elde edilebilmektedir.

## KAYNAKLAR

- [1] <http://www.electricaleasy.com/2015/10/diesel-power-generator-and-power-plant.html>, Erişim Tarihi: 08.07.2016.
- [2] Lantronix, “Modbus Protocol User Guide”, June 2013.
- [3] Johannes Kasberger, “Comparison of Modbus over TCP/IP and PROFINET”, 2011.
- [4] James Powell, “Profibus and Modbus: A Comparison”, Siemens Oct 2013.
- [5] MOXA, “Industrial Protocols User’s Guide”, Fifth edition, June 2013.
- [6] <http://www.ni.com/white-paper/7675/en/>, Erişim Tarihi: 03.08.2016.
- [7] <https://www.medialon.com/product/medialon-flexio-modbus-tcp/>, Erişim Tarihi: 08.06.2016.
- [8] [http://www.pacifictoolcompany.com/control-panels-c-13\\_143\\_147/](http://www.pacifictoolcompany.com/control-panels-c-13_143_147/), Erişim Tarihi: 08.07.2016.
- [9] PENG Dao-gang, ZHANG Hao, YANG Li, LI Hui, “Design and Realization of Modbus Protocol Based on Embedded Linux System”, International Conference on Embedded Software and Systems Symposia (ICCESS), 2008.
- [10] Jinfeng Li, Shun Cao, “Remote Monitoring and Management System of CNG Flow based on Modbus RTU Protocol”, International Journal of Online Engineering (IJOE), 2014.
- [11] Daogang Peng, Hao Zhang, Jiannian Weng, “Design and development of Modbus/RTU master monitoring system based on embedded PowerPC platform”, IEEE International Symposium on Industrial Electronics, 2009.
- [12] Derwin Suhartono, Aryan Wibowo, Setiady Wiguna, Robby Saleh, "Developing Controller Area Network Management Application Based on Modbus in Multi Generator Set Controller through Local Network and Internet", International Conference on Advances Science and Contemporary Engineering (ICASCE), 2012.

- [13] Shiben Zhu, Xiaoping Zhang, "Design of Monitoring Control System Based on MODBUS for Transformer Measurement Equipment", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE), 2013.
- [14] Technical Document, "MODBUS Messaging on TCP/IP Implementation Guide V1.0b", October 24, 2006.
- [15] Technical Reference, "Introduction to Modbus TCP/IP", ACROMAG INCORPORATED, 2005.
- [16] [https://en.wikipedia.org/wiki/Modbus#Protocol\\_versions](https://en.wikipedia.org/wiki/Modbus#Protocol_versions), Erişim Tarihi: 07.03.2016.
- [17] <https://www.qt.io/>, Erişim Tarihi: 07.05.2016.
- [18] Blanchette, Mark Summerfield, C++ GUI Programming with Qt 3, 2004.
- [19] Technical Document, "CodeWarrior Development Studio for Microcontrollers V10.x Targeting Manual", 2014.
- [20] <http://www.nxp.com/products/microcontrollers-and-processors/arm-processors/kinetis-cortex-m-mcus:KINETIS>, Erişim Tarihi: 07.06.2016.
- [21] Freescale Semiconductor, "K60 Sub-Family Data Sheet", 2011.
- [22] TWR-K60D100M Tower Module User's Manual Rev. 1.1, Aug 28, 2011.
- [23] İbrahim Sefa, Ramazan Bayındır, İlhan Garip, Sertaç Bayhan Ve İlhami Çolak, "Generatörlerin Paralel Bağlanma Şartlarından Frekans Ve Faz Açısı Eşitliğinin Mikrodenetleyici Tabanlı Olarak Sağlanması", J. Fac. Eng. Arch. Gazi Univ. Vol 25, No 1, 39-48, 2010.
- [24] <http://www.aksapowergen.com/pdf/other/Parallel+Generators+Synchronizati on.pdf>, Erişim Tarihi: 06.03.2016.
- [25] Uzma Amin, Ghulam Ahmad, SumbalZahoor, FarihaDurrani, "Implementation of Parallel Synchronization Method of Generators for Power & Cost Saving in University of Gujrat", Energy and Power Engineering, 317-332, 2014.
- [26] [https://powersuite.cummins.com/PS5/PS5Content/SiteContent/en/Binary\\_Ass et/pdf/Commercial/technical/T-016.pdf](https://powersuite.cummins.com/PS5/PS5Content/SiteContent/en/Binary_Ass et/pdf/Commercial/technical/T-016.pdf), Erişim Tarihi: 04.04.2016.
- [27] Antonio Ramos, Luis Garabito, "Using MQX Libraries", 2009.

- [28] Michal Princ, “Implementing an IEEE 1588 V2Node on the Kinetis K60 Using the Freescale MQX IEEE 1588 Communication Library”, 2011.
- [29] Carlos Musich, Alí Piña, and Carlos Casillas, “Remote Monitoring Solution Using MQX and Kinetis”, 2013.
- [30] Freescale MQX™ RTOS User's Guide, Rev. 14, 2015.
- [31] Freescale Semiconductor, “Getting Started with Freescale MQX™ RTOS”, 2011.
- [32] <https://bernini-design.com/generator-control-systems/>, Erişim Tarihi: 07.03.2016.
- [33] <http://www.ni.com/white-paper/3064/en/>, Erişim Tarihi: 08.09.2016.
- [34] <http://www.nxp.com/products/software-and-tools/hardware-development-tools/tower-development-boards/peripheral-modules/serial-usb-ethernet-cans232-485-tower-system-module:TWR-SER>, Erişim Tarihi: 09.03.2016.
- [35] <http://libmodbus.org/>, Erişim Tarihi: 06.06.2016.
- [36] Andy Swales, “Open MODBUS /TCP Specification”, Schneider Electric, March 1999.

## **ÖZGEÇMİŞ**

Motuma Abafogi, 20.08.1988'de Etiyopya'nın Metu şehrinde doğdu. İlk, orta ve lise eğitimini Etiyopya'daki Jimma şehrinde tamamladı. 2005 yılında Jimma Hazırlık Lisesi'nden mezun oldu. 2005 yılında başladığı Jimma Üniversitesi Elektrik Mühendisliği Bölümü'nü 2009 yılında bitirdi. 2012 yılından itibaren Sakarya Üniversitesi Elektrik Elektronik Mühendisliği Bölümü'nde yüksek lisans eğitimini görmektedir.