KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME ÇALIŞMASI

MSP430G2553 MİKRODENETLEYİCİSİ, MPU6050 SENSÖRÜ VE HMC5883L SENSÖRÜ İLE ATIŞ POLİGON SİMÜLASYONU: UART ÜZERİNDEN WİNDOWS FORMS UYGULAMASIYLA İLETİŞİM

OZAN HAMDİ KAPLAN
200207091

DANIŞMAN: PROF. DR. ALİ TANGEL

KOCAELİ , 2024

ÖNSÖZ ve TEŞEKKÜR

Günümüzde mikrodenetleyici ve kablosuz iletişim teknolojisinin hızla gelişmesi, askeri alanlarda ve atış poligonlarında mermi tasarrufu elde eden yenilikçi projelerin hayata geçirilmesine imkan tanımaktadır. Artan mermi maliyetleri ve bütçe kısıtlamaları, silahlı kuvvetler ve sivil atış poligonları için ekonomik ve sürdürülebilir çözümlerin bulunmasını gerektirmektedir.

Bu proje araştırması, MSP430g2553 mikrodenetleyicisi ve HC-05 Bluetooth modülünü kullanarak, mermi tasarrufu elde eden bir atış poligon simülasyonu tasarlamayı amaçlamaktadır. Bu projede, MPU6050 sensöründen alınan ivme ve gyro verileri, HMC5883L sensöründen alınan manyetometre verileri, HC-05 üstünden UART verisiyle bir bilgisayar uygulamasına iletilir ve bu veriler Visual Studio`da açılan Windows Forms uygulamasında görselleştirilir. Uygulama, hedef tahtasında atışların simülasyonunu sağlayarak, kullananların gerçek mermi kullanmadan atış eğitimi yapmasına imkan tanır. Bu biçimde, mermi maliyetlerini düşürmek ve eğitim süreçlerinde tutum sağlamak hedeflenmektedir.

Raporun bu bölümünde, projenin motivasyonlarına ve hedeflerine odaklanılarak, MSP430g2553 mikrodenetleyicisi ve HC-05 Bluetooth modülünün kullanımının neden seçildiği, projenin tasavvur aşamaları ve teknik bilgileri anlatılmaktadır. Ayrıca, simülasyonun mermi tasarrufu sağlayacak biçimde nasıl çalıştığı ve uygulamanın genel yapısı değerlendirilmekte, elde edilen sonuçlar okuyucuya sunulmaktadır.

Sevgili Prof. Dr. Ali Tangel hocama, sınıf arkadaşlarıma, son sınıf arkadaşlarıma ve mezun arkadaşlarıma. Bu uzun sürecin her anında yanımda olan aileme özellikle teşekkür ediyorum. Onların maddi ve manevi destekleri bu noktaya gelmemde etkili oldu. Onların desteği ve güçlü motivasyonu olmasaydı bu süreci başarıyla tamamlayamazdım. İyi ki varsınız.

Hepinize yürekten teşekkürlerimi sunuyorum. Sizlerin katkılarıyla bu çalışma daha anlamlı ve başarılı hale geliyor. Umarım birlikte geçireceğimiz zamanlar gelecekteki projelerde bir araya gelmemize vesile olur. Bunu takdir ediyorum.

Haziran 2024, KOCAELİ OZAN HAMDİ KAPLAN

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR ii
İÇİNDEKİLER iii
ŞEKİLLER DİZİNİ v
TABLOLAR DİZİNİ 6
SİMGELER DİZİNİ VE KISALTMALAR 7
1. GİRİŞ 10
1.1 Gömülü Sistem Nedir Öğrenelim 11
1.2 Tezin Amacı ve Projenin Kapsamı 12
1.3 Çalışmanın Önemi ve Motivasyonu 13
2. Literatür taraması 14
2.1 Teoriler ve Teknolojiler 14
2.1.1 Bluetooth ile Kablosuz Haberleşme 14
2.1.2 I2C (Inter-Integrated Circuit) 15
2.1.3 UART Haberleşmesi 17
2.1.4 Interrupt 19
2.1.5 İvmeölçer ve Jiroskop 20
2.1.6 Manyetometre 22
2.2 Benzer Projeler ve Uygulamalar 23
3. KULLANILAN CİHAZLAR VE PROGRAMLAR 25
3.1 Kullanılan Cihazlar 25
3.1.1 MSP430 Launchpad Geliştirme Aracı 25
3.1.1.1 MSP430G2553 Mikroişlemcisi 26
3.1.2 HC-05 modülü 27
3.1.3 MPU-6050 28
3.1.4 HMC5883L 30
3.2 Kullanılan Programlar 31
3.2.1 Kod Tertipleyen Stüdyo (CCS) Programı 31
3.2.1.1 C Programlama Dili 32
3.2.2 Visual Studio 33
3.2.2.1 Windows Form App 34
3.2.2.2 C# Programlama Dili 35
3.2.3 Altium Designer 36
4. DONANIM 37
4.1 Şematik Tasarımı 37
4.1.1 Güç Kaynağı Şeması 37
4.1.2 Sensör Bağlantı Şeması 39
4.1.3 Ledli Bildirim Şeması 40
4.1.4 Butonların ve Lazer Ledin Bağlantı Şeması 41
4.1.5 MSP430G2553 Bağlantı Şeması 42
4.2 PCB Tasarım 43
4.3 Baskı Devre Yapımı 44

5. YAZILIM 47
5.1 Bluetooth Cihazlarının Yapılandırılması 47
5.2 Mikroişlemci Programlama: Code Composer Studio ile C Kodları 49
5.2.1 Bluetooth İletişimi için UART Yapılandırması 49
5.2.2 MPU-6050 Sensörü Ve HMC5883L için I2C Yapılandırması 49
5.2.3 Buton ve Diğer Bileşenlerin Pin Yapılandırması 50
5.2.4 Fonksiyon İşlemleri 50
5.3 Windows Form App Geliştirme: C# ile Kullanıcı Arayüzü ve Fonksiyon İşlemleri 51
5.3.1 Kullanıcı arayüzü 51
5.3.2 Fonksiyon İşlemleri 52
5.4 Akış Şemaları 54
5.4.1 C kodunun Akış şeması 54
5.4.2 C# kodunun Akış Şeması 56
6. UYGULAMA 57
6.1 MPU6050'nin Kullanımı, ... 57
6.1.1 MPU6050'nin Başlatılması (Mpu6050_WakeUp) 58
6.1.2 İvmeölçer ve Jiroskop Verilerini Okuma (ReadAccelGyro) 59
6.2 HMC5883L'nin Kullanımı 59
6.2.1 HMC5883L'nin Başlatılması (HMC5883L_Init) 60
6.2.2 HMC5883L Verilerinin Okunması (ReadHMC5883L) 61
6.3 Programın Çalışması 62
6.3.1 Silah Bağlantısı ve Hazırlığı 62
6.3.2 Simülasyon ve Sonuçlar 64
7. SONUÇLAR ve ÖNERİLER 66
KAYNAKÇA 68
ÖZGEÇMİŞ 71

ŞEKİLLER DİZİNİ

Şekil 1:1 Mikrodenetleyici İç Yapısı [6] ... 11
Şekil 1:2 Blok Diyagram ... 12
Şekil 2:1 Bluetooth Bağlantıları [10] ... 15
Şekil 2:2 I2C Veri Yolu İletişimi ... 16
Şekil 2:3 Uart Haberleşmesi ... 18
Şekil 2:4 Interrupt Mikrodenetleyici Block Diyagramı [16] ... 20
Şekil 2:5 Duck Hunter ... 24
Şekil 3:1 MSP430 Launchpad Geliştirme Aracı [19] ... 25
Şekil 3:2 MSP430 Launchpad bacak dağılımı ... 26
Şekil 3:3 MSP430G2553 Mimari Yapısı [20] ... 26
Şekil 3:4 HC-05 Modülü [21] ... 28
Şekil 3:5 MPU-6050 modülü [23] ... 29
Şekil 3:6 HMC5883L Modülü [24] ... 31
Şekil 3:7 CCS Programının Görüntüsü. ... 32
Şekil 3:8 CCS Yazılım Geliştirme Akışı [26] ... 32
Şekil 3:9 Visual Studio Programının Görüntüsü ... 34
Şekil 3:10 Visual Studio Yeni Proje Başlatma Görüntüsü ... 36
Şekil 3:11 Altium Designer Programının Görüntüsü ... 36
Şekil 4:1 Güç Kaynağı Şeması ... 37
Şekil 4:2 Sensör Bağlantı Şeması ... 40
Şekil 4:3 Ledli Bildirim Şeması ... 40
Şekil 4:4 Butonların ve Lazer Ledin Bağlantı Şeması ... 41
Şekil 4:5 MSP430G2553 Bağlantı Şeması ... 42
Şekil 4:6 PCB Tasarım ... 44
Şekil 4:7 PCB Ön Yüz ... 44
Şekil 4:8 PCB Arka Yüz ... 45
Şekil 4:9 PCB Baskı ... 45
Şekil 4:10 Kimyasal Karışım Sonrası Devre Deseni ... 45
Şekil 4:11 Baskı Devrenin Silahın İçine Alınması ... 46
Şekil 4:12 Silah Üzerindeki Eklenen Parçalar ve İsimleri ... 46
Şekil 5:1 Bluetooth AT Komutu ile Serial Port Ekranında Yapılandırılması ... 48
Şekil 5:2 Kullanıcı Arayüzü ... 52
Şekil 5:3 C Kodu Akış Şeması ... 55
Şekil 5:4 C# Akış Şeması ... 56
Şekil 6:1 Mpu6050_WakeUp Fonksiyonu ... 58
Şekil 6:2 İvmeölçer Konfigurasyonu Bitleri [32] ... 58
Şekil 6:3 İvmeölçer Konfigurasyonu Range Değerleri [32] ... 58
Şekil 6:4 ReadAccelGyro Fonksiyonu ... 59
Şekil 6:5 İvmeölçer Ölçümleri (X,Y,Z) [32] ... 59
Şekil 6:6 HMC5883L_Init Fonksiyonu ... 60
Şekil 6:7 ReadHMC5883L Fonksiyonu ... 61
Şekil 6:8 Manyetometre Ölçümleri (X,Y,Z) [4] ... 61
Şekil 6:9 Kullanıcı Arayüzü Bluetooth Bağlantısı Yapıldı ... 62
Şekil 6:10 Silah Bluetooth Bağlantısı Yapıldı ... 63
Şekil 6:11 Kalibrasyon Tamamlandı ... 63
Şekil 6:12 Hedefte Vurulan Yer (1) ... 64
Şekil 6:13 Hedefte Vurulan Yer (2) ... 64
Şekil 6:14 Hedefte Vurulan Yer (3) ... 64
Şekil 6:15 Mermi Uyarı Sistemi Devrede ... 65

TABLOLAR DİZİNİ

Tablo 1 MPU6050 Registerları [32] 57
Tablo 2 HMC5883L Registerları [4] 60

SİMGELER DİZİNİ VE KISALTMALAR

MSP	: Microcontroller Subsystem
I2C	: Inter-Integrated Circuit
UART	: Universal Asynchronous Receiver-Transmitter
RX	: Receive
TX	: Transmit
SDA	: Serial Data Line
SCL	: Serial Clock Line
IoT	: Internet of Things
ISM	: Industrial, Scientific, and Medical (radio bands)
AFH	: Adaptive Frequency Hopping
SIG	: Special Interest Group
IBM	: International Business Machines
ADC	: Analog-to-Digital Converter
EEPROM	: Electrically Erasable Programmable Read-Only Memory
ISR	: Interrupt Service Routine
NVIC	: Nested Vectored Interrupt Controller
NMI	: Non-Maskable Interrupt
IRQs	: Interrupt Requests
MEMS	: Micro-Electro-Mechanical Systems
GPS	: Global Positioning System
USB	: Universal Serial Bus
RISC	: Reduced Instruction Set Computer
VCC	: Voltage Common Collector
GND	: Ground
SPI	: Serial Peripheral Interface
DMP	: Digital Motion Processor
CCS	: Code Composer Studio (Texas Instruments IDE)
IDE	: Integrated Development Environment
CI/CD	: Continuous Integration / Continuous Deployment
BOM	: Bill of Materials
PCB	: Printed Circuit Board
PSWD	: Password
USB-TTL	: USB to Transistor-Transistor Logic
AT	: Attention (command prefix used in Hayes command set for modems)
GPIO	: General Purpose Input/Output

MSP430G2553 MİKRODENETLEYİCİSİ, MPU6050 SENSÖRÜ VE HMC5883L SENSÖRÜ İLE ATIŞ POLİGON SİMÜLASYONU: UART ÜZERİNDEN WİNDOWS FORMS UYGULAMASIYLA İLETİŞİM

OZAN HAMDİ KAPLAN

Anahtar Kelimeler: MSP430G2553 Mikrodenetleyicisi, HC-05 Bluetooth Modülü, Kablosuz İletişim, MPU6050 Sensörü, HMC5883L sensörü, UART, Windows Form Uygulaması, Gömülü Sistemler, Atış Poligon Simülasyonu, Veri Görselleştirme.

Özet:
Bu projede, MSP430G2553 mikrodenetleyicisi, HC-05 Bluetooth modülü, MPU6050 sensörü ve HMC5883L sensörü kullanılarak bir atış poligon simülasyonu gerçekleştirilmiştir. Proje kapsamında, mikrodenetleyici tabanlı bir sistem tasarlanarak, MPU6050 sensöründen ve HMC5883L sensöründen alınan ivme, jiroskop ve manyetometre verileri Bluetooth aracılığıyla Windows Forms tabanlı bir uygulamaya aktarılmıştır. Bu veri akışı, kullanıcının bilgisayarda simülasyon üzerinde gerçek zamanlı analiz ve kontrol yapmasına olanak tanımaktadır.

Projenin temel amacı, MSP430 mikrodenetleyicisi ve Bluetooth teknolojisi ile kablosuz veri iletişimini sağlamak ve MPU6050 sensörü ve HMC5883L sensörü ile hareket ve eğim verilerini almak ve iletmektir. Verilerin Windows Forms uygulaması aracılığıyla görselleştirilmesi, kullanıcıya atış poligon simülasyonunun görsel ve etkileşimli bir deneyimini sunar.

Proje sırasında karşılaşılan zorluklardan biri, MPU6050 sensöründen ve HMC5883L sensöründen alınan verilerin doğru ve tutarlı şekilde aktarılmasıdır. UART üzerinden gerçekleştirilen iletişim, veri kaybı riskini azaltmak için uygun baud hızı ayarları ve hata kontrol mekanizmaları ile desteklenmiştir. Ayrıca, kablosuz bağlantı ile ilgili güvenilirlik sorunları, HC-05 Bluetooth modülü için AT komutları kullanılarak çözümlenmiştir.

Bu projede elde edilen deneyimler, gömülü sistemler ve kablosuz iletişim konularında pratik bir uygulama sağlar. Proje sonucunda, mikrodenetleyici tabanlı sistemlerin farklı sensörlerle entegrasyonu ve kablosuz iletişim yoluyla veri aktarımı konularında önemli bilgiler elde edilmiştir. Bu deneyim, gelecekteki projeler ve geliştirmeler için sağlam bir temel oluşturmaktadır.

SHOOTING RANGE SIMULATION WITH MSP430G2553 MICROCONTROLLER, MPU6050 SENSOR AND HMC5883L SENSOR: COMMUNICATION WITH WINDOWS FORMS APPLICATION VIA UART

OZAN HAMDİ KAPLAN

Keywords: MSP430G2553 Microcontroller, HC-05 Bluetooth Module, Wireless Communication, MPU6050 Sensor, HMC5883L Sensor, UART, Windows Form Application, Embedded Systems, Shooting Range Simulation, Data Visualization.

Abstract:
In this project, a shooting range simulation was realized using MSP430G2553 microcontroller, HC-05 Bluetooth module, MPU6050 sensor and HMC5883L sensor. Within the scope of the project, a microcontroller-based system was designed and acceleration, gyroscope and magnetometer data from the MPU6050 sensor and HMC5883L sensor were transferred to a Windows Forms-based application via Bluetooth. This data flow allows the user to perform real-time analysis and control over the simulation on the computer.

The main purpose of the project is to provide wireless data communication with the MSP430 microcontroller and Bluetooth technology and to receive and transmit motion and tilt data with the MPU6050 sensor and HMC5883L sensor. Visualization of the data through the Windows Forms application provides the user with a visual and interactive experience of the shooting range simulation.

One of the challenges encountered during the project is the accurate and consistent transmission of data from the MPU6050 sensor and the HMC5883L sensor. Communication over UART was supported with appropriate baud rate settings and error checking mechanisms to reduce the risk of data loss. In addition, reliability issues related to wireless connectivity were addressed by using AT commands for the HC-05 Bluetooth module.

Translated with DeepL.com (free version)
The experiences gained in this project provide a practical application in embedded systems and wireless communication. As a result of the project, important knowledge has been gained on the integration of microcontroller-based systems with different sensors and data transfer via wireless communication.

1. GİRİŞ

Gömülü sistemler, günümüz teknolojisinin temelini oluşturur ve farklı endüstrilerde geniş bir uygulama yelpazesi sunar. Bu sistemler, mikrodenetleyici tabanlı cihazlar ve çeşitli sensörlerin entegrasyonu ile karmaşık işlevleri yerine getirebilir. Bu tez çalışması, MSP430G2553 mikrodenetleyicisini kullanarak kablosuz Bluetooth iletişimi aracılığıyla veri aktarımını ele alır ve Windows Form App gibi masaüstü uygulamaları ile veri analizini inceler. Ayrıca, kablosuz iletişimde kullanılan HC-05 Bluetooth modülü ile MPU6050 jiroskop ve ivmeölçer sensörünün ve HMC5883L manyetometre sensörünün kombinasyonuna odaklanır.

MSP430G2553, Texas Instruments'ın ürettiği popüler bir mikrodenetleyicidir. Düşük güç tüketimi ve çeşitli çalışma modları ile endüstriyel otomasyon ve tıbbi cihazlar dahil birçok alanda kullanılır. Bu projede MSP430G2553, MPU6050 ve HMC5883L sensörleriyle birleşerek hareket ve yönelim verilerini toplar ve bu verileri kablosuz olarak HC-05 Bluetooth modülü aracılığıyla bir bilgisayara iletir [1].

HC-05, uygun maliyeti ve kolay entegrasyonu sayesinde gömülü sistemlerde sıkça tercih edilen bir Bluetooth modülüdür. Kablosuz iletişim sağlayarak, mikrodenetleyiciden alınan verilerin uzaktan izlenmesine ve analiz edilmesine imkan tanır. Bu tez, HC-05 Bluetooth modülünün nasıl yapılandırıldığını ve veri iletiminde nasıl kullanıldığını detaylandırır [2].

MPU6050, entegre jiroskop ve ivmeölçer sensörüdür. Hareket ve eğim gibi bilgiler sağlayarak robotik ve hareket kontrol sistemlerinde sıkça kullanılır [3]. HMC5883L, manyetometre sensörüdür ve manyetik alanı ölçerek cihazın yönünü belirler [4]. Robotik sistemlerden navigasyon uygulamalarına kadar geniş bir kullanım alanına sahiptir. Bu çalışmada MPU6050 ve HMC5883L sensörlerinden gelen verilerin MSP430G2553 tarafından alınıp işlendiği, ardından HC-05 Bluetooth modülü aracılığıyla Windows Form App'e aktarıldığı açıklanacaktır.

Windows Form App, Microsoft'un masaüstü uygulama geliştirme platformudur. Bu projede, Windows Form App, HC-05 modülü üzerinden gelen sensör verilerini

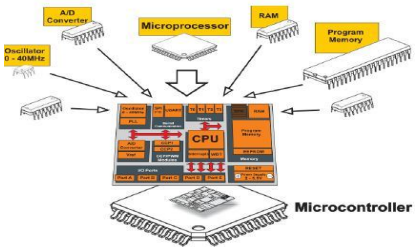
alarak görselleştirme ve analiz için kullanıcı dostu bir arayüz sağlar. Böylece, mikrodenetleyiciden alınan verilerin gerçek zamanlı olarak izlenmesi ve analiz edilmesi mümkün olur.

Bu tez çalışması, MSP430G2553 mikrodenetleyicisi, HC-05 Bluetooth modülü, MPU6050 sensörü ve HMC5883L kullanılarak kablosuz veri iletimini detaylı bir şekilde inceleyecektir. Proje kapsamında, bu bileşenlerin nasıl entegre edildiği ve veri aktarımının Windows Form App üzerinde nasıl işlendiği gösterilecektir. Çalışma, kablosuz veri iletimi ve gömülü sistem tasarımı konusunda pratik bir rehber olmayı hedeflemektedir.

1.1 Gömülü Sistem Nedir Öğrenelim

Gömülü sistem, genellikle belirli bir görevi yerine getirmek üzere tasarlanmış ve genellikle spesifik bir işlevi yerine getirmek amacıyla optimize edilmiş bilgisayar sistemleridir. Bu sistemler, genellikle özel donanım ve yazılım kombinasyonları kullanılarak tasarlanır ve genellikle daha büyük bir bilgisayar sistemi veya ağın bir parçası olarak kullanılmazlar. Gömülü sistemler genellikle daha spesifik ve özelleştirilmiş görevlere odaklanmak üzere optimize edilmiştir ve genellikle daha düşük maliyetli, daha güçlü ve daha enerji verimli olabilirler [5].

Gömülü sistemler, birçok farklı endüstriyel, tıbbi, otomotiv, tüketici elektroniği ve diğer uygulamalarda yaygın olarak kullanılır. Örnek olarak, bir araçtaki motor kontrol sistemleri, bir dijital kamera içindeki görüntü işleme birimi veya bir evdeki termostatın kontrol ünitesi gibi gömülü sistemlere örnek verilebilir. Şekil 1:1 mikrodenetleyicilerin genel iç yapısını göstermektedir.



Şekil 1:1 Mikrodenetleyici İç Yapısı [6]

1.2 Tezin Amacı ve Projenin Kapsamı

Tezin amacı, MSP430G2553 mikrodenetleyicisi ile MPU6050 ve HMC5883L sensörlerinden alınan hareket verilerinin HC-05 Bluetooth modülü üstünden Windows Form uygulamasına nasıl aktarıldığını göstermektir. Bu proje, gömülü sistemler ile kablosuz iletişim arasındaki etkileşimi inceleyerek pratik bir uygulama ortaya koymayı hedefler. Ayrıca, bu çalışmanın hedefleri, Bluetooth tabanlı kablosuz veri iletimini algılamak ve gömülü sistemlerin taşınabilir cihazlarla nasıl çalıştığını göstermektir. Şekil 1:2'de gösterilen blok diyagramı, MSP430 mikrodenetleyicisine bağlı olan butonlar (Buton1, Buton2, Buton3), MPU6050 ve HMC5883L sensörleri, LED ve HC-05 Bluetooth modülünün nasıl bağlandığını detaylandırmaktadır. Bu diyagram, sistemin genel yapısını ve bileşenler arasındaki bağlantıları net bir şekilde ortaya koyar.

Şekil 1:2 Blok Diyagram

Projenin kapsamı, MSP430G2553 mikrodenetleyicisi, HC-05 Bluetooth modülü, MPU6050 sensörü ve HMC5883L ile sınırlıdır. Bu kapsamda, mikrodenetleyici ile sensör arasındaki veri alışverişi, Bluetooth modülü ile kablosuz iletişim ve Windows Form uygulaması aracılığıyla verilerin görselleştirilmesi ayrıntılı olarak ele alınacaktır. Sınırlar ise belirli bir donanım yapısına odaklanarak diğer platformların dışarıda bırakılmasıdır.

1.3 Çalışmanın Önemi ve Motivasyonu

Bu çalışmanın önemi, gömülü sistemler ve kablosuz iletişim alanlarındaki pratik uygulamalara katkı sağlamasından kaynaklanmaktadır. Gömülü sistemler, mikrodenetleyiciler ve sensörler arasındaki etkileşimin kritik olduğu bir alandır ve günümüzde IoT (Nesnelerin İnterneti) ve endüstriyel otomasyon gibi farklı alanlarda yoğun bir biçimde kullanılmaktadır. Bu tür sistemlerde, cihazlar arasındaki kablosuz iletişim ve sensörlerden elde edilen verilerin işlenmesi, çok çeşitli uygulamalar için temel bir gereksinimdir.

Bu proje, MSP430G2553, HC-05, MPU6050 ve HMC5883L gibi bileşenlerin bir arada nasıl çalışabileceğini göstererek, farklı uygulamalar için sağlam ve güvenilir bir temel kurmayı amaçlamaktadır. Proje, mikrodenetleyici tabanlı bir sistemin Bluetooth aracılığıyla sensör verilerini iyi mi işleyebileceğini ve kablosuz iletişimin avantajlarını vurgulamaktadır. Kablosuz iletişim, özellikle uzaktan kontrol, izleme ve veri aktarımı gerektiren uygulamalarda büyük bir öneme sahiptir.

Gömülü sistemlerin endüstriyel otomasyon, akıllı evler, sağlık hizmetleri ve IoT gibi alanlarda artan kullanımı, bu tür teknolojilere olan ilgiyi artırmaktadır. Çalışma, bu tür sistemlerin pratikte nasıl uygulanabileceğini göstererek, bu alanda bilgi ve tecrübe kazanmaya imkan tanır. Proje ayrıca, gömülü sistemlerin güvenli ve enerji verimli bir biçimde çalışmasını sağlamak için doğru bileşen seçiminin ve kablosuz iletişimin önemini vurgular.

Bu bağlamda, çalışmanın önemi, gömülü sistemler ve kablosuz iletişim konularında bilgi sağlamak ve pratik uygulama fırsatları sunmaktır. Bu tür projeler, gelecekteki teknolojik gelişmeler ve inovasyonlar için bir temel oluşturarak, sanayi ve bilimsel niteliği olan araştırmalar için değerli kaynaklar haline gelir. Çalışmanın motivasyonu ise, Bluetooth teknolojisi ve sensör verilerinin gömülü sistemler ile nasıl entegre edildiğini anlamak ve bu bilgiyi pratik bir projede uygulamaktır.

2. LİTERATÜR TARAMASI

2.1 Teoriler ve Teknolojiler

2.1.1 Bluetooth ile Kablosuz Haberleşme

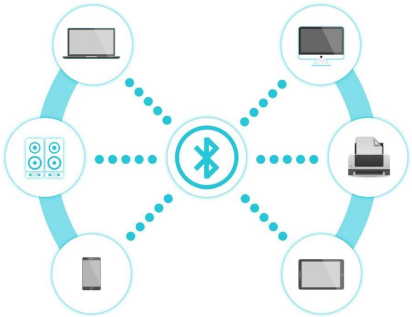
Bluetooth teknolojisi, 1994 senesinde Ericsson tarafınca RS-232 veri kablolarına kablosuz bir alternatif olarak geliştirilmiş ve günümüzde geniş bir kullanım alanına sahiptir [7]. Bluetooth cihazları, 2,4 GHz ile 2,485 GHz arasındaki Sınai, Bilimsel ve Tıbbi Cihaz (ISM) bandında çalışır ve dünya genelinde birçok ülkede lisanssız olarak kullanılabilirler [8]. Bu bandda çalışan birçok aygıt olduğundan, iletişim zaman zaman parazitlere maruz kalabilir, sadece bu problemi hafifletmek için Uyarlanır Frekans Hoplaması (AFH) teknolojisi kullanılır. AFH, yakın çevrede kullanılan frekansları tespit ederek, bu frekanslardan kaçınmayı sağlar.

Frekans atlama özelliği, uzaklık ve parazit şeklinde sorunları en aza indirirken, alıcı-verici maliyetlerini de düşürür. Bluetooth teknolojisi çift yönlü iletişim sunar ve düşük enerji tüketimi ile ucuz olması yardımıyla popülerdir. Bluetooth`un bu avantajları, onun kablosuz iletişim için tercih edilmesinde önemli bir rol oynar.

Bluetooth teknolojisinin geçmişine bakıldığında, 1998'de kısa mesafeli kablosuz iletişim için dünya genelinde ve lisanssız bir standart kurmak amacıyla Ericsson, IBM, Intel, Nokia ve Toshiba'nın Bluetooth Special Interest Group'u (SIG) kurduğu görülür [9]. SIG, bu hızla gelişen değişen teknolojinin yaygın kabul görmesi ve kullanımını teşvik etmek için teknik bilgileri halka açık hale getirdi. 1999'da Bluetooth 1.0A sürümü yayınlandığında, 3Com, Lucent, Microsoft ve Motorola da bu gruba katılarak teknolojiyi geliştirmeye devam ettiler. Günümüzde, Bluetooth 5.2 şeklinde daha ileri sürümler ile daha yüksek iletişim hızı, artırılmış menzil ve daha düşük enerji tüketimi sağlanmıştır.

Bluetooth teknolojisi, sadece kişisel elektronik cihazlarda değil, bununla birlikte akıllı ev teknolojileri, otomasyon sistemleri ve endüstriyel test alanlarında da yaygın olarak kullanılmaktadır. Örneğin, Bluetooth tabanlı kablosuz bağlantı ile üretim

hatları üstündeki sensörlerden veri toplanarak izlenebilir ve anlık ayarlamalar yapılabilir. Bluetooth'un düşük enerji tüketimi ve uyarlanabilir yapısı, MSP430 mikrodenetleyicisi şeklinde düşük kuvvetli cihazlarla beraber kullanılmasını ideal hale getirir. Bu bağlamda, Bluetooth modülü kullanılarak MSP430 tabanlı bir sistemde kablosuz veri aktarımı sağlanabilir ve bu veriler Windows Form App şeklinde bir arayüz ile gerçek zamanlı olarak izlenebilir. Bu tür uygulamalar, endüstriyel test sistemleri, akıllı cihazlar ve diğer gömülü sistem projeleri için büyük potansiyel sunar. Şekil 2:1'de bluetooth bağlantıları gözlemlenmektedir.



Şekil 2:1 Bluetooth Bağlantıları [10]

2.1.2 I2C (Inter-Integrated Circuit)

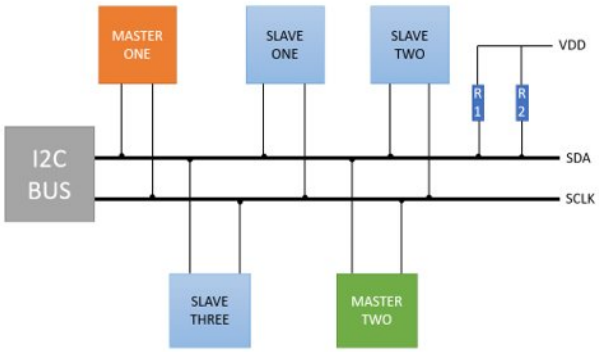
I2C (Inter-Integrated Circuit), düşük hızlı seri bir iletişim protokolüdür. Philips Semiconductor (şimdiki adıyla NXP Semiconductors) tarafınca geliştirilmiştir ve mikrodenetleyiciler, sensörler, bellek çipleri ve diğer elektronik bileşenler arasında veri aktarımı için yaygın olarak kullanılır. I2C`nin popülaritesi, çoklu cihazların aynı fizyolojik hat üzerinde iletişim kurmasına olanak tanıyan rahat ve esnek yapısından kaynaklanır [11].

Özellikleri:

- İki Tel (Two-Wire) Yapısı: I2C, veri aktarımı için sadece iki hat kullanır:
    - SDA (Serial Data Line): Veri iletimi için kullanılır.
    - SCL (Serial Clock Line): Zamanlama ve senkronizasyon için kullanılır.

- Adresleme Sistemi: I2C'de, her cihazın benzersiz bir adresi vardır. Bu adres, bir cihazın ağ üstündeki kimliğini belirler ve ana cihazın belirli bir cihazla iletişim kurmasını sağlar.
- Çoklu Ana Cihaz Desteği: I2C, aynı hatta birden fazla ana cihazın (master) bulunmasını destek sunar. Bu ana cihazlar, hat üzerindeki kontrolü paylaşabilir ve diğer cihazlarla iletişim kurabilir.
- Çoklu Cihaz Desteği: I2C, bir ana cihazın aynı anda birçok cihazla iletişim kurmasına izin verir. Bu, sensörlerin, EEPROM'ların, ADC'lerin ve diğer bileşenlerin aynı veri yolunda beraber çalışmasını sağlar.
- Senkronizasyon ve Sinyalizasyon: I2C, senkronize bir protokol olduğu için SCL hattı, veri iletimini senkronize etmek için kullanılır. Bir cihazın veri göndermek ya da almak için hazır olup olmadığını belirtmek için saat hattını kullanarak sinyal verebilir.

I2C çalışma prensibi, bir ana cihazın veriyi gönderdiği ve birden fazla cihazın bu veriyi alabileceği bir sistemdir [12]. Ana cihaz, bir cihazın adresine veri göndermek için cihaz adresini ve ardından iletilecek veriyi gönderir. Alıcı cihaz, adresin kendisine uygun olup olmadığını kontrol eder ve arkasından veri iletimini gerçekleştirir. Şekil 2:2 ‘de I2C için veri yolu iletişimi gözlemlenmektedir.



Şekil 2:2 I2C Veri Yolu İletişimi [33]

Avantajları

- Basit ve Ucuz: I2C, düşük maliyetli bir iletişim protokolüdür ve karmaşık olmayan yapısı, düşük maliyetli devre tasarımlarına imkan tanır.

- Esnek ve Ölçeklenebilir: I2C, birden fazla cihazı destekler ve yeni cihazların kolayca eklenmesini sağlar.
- İki Tel İletişimi: Diğer seri protokollere göre daha az kablolama gerektirir ve karmaşık olmayan devre tasarımları için idealdir.

Dezavantajları

- Düşük Hız: I2C, daha düşük hızlara sahiptir ve yüksek hızlı uygulamalar için uygun değildir.
- Kısıtlı Mesafe: I2C, kısa mesafeli iletişim için tasarlanmıştır ve uzun mesafeli veri iletimi için uygun değildir.

I2C, gömülü sistem projelerinde sensörler, mikrodenetleyiciler ve diğer bileşenler arasında iletişim kurmak için yaygın olarak kullanılır. Esnekliği ve düşük maliyeti nedeniyle, I2C, elektronik projelerde ve mikrodenetleyici tabanlı sistemlerde tercih edilen bir protokol haline gelmiştir.

2.1.3 UART Haberleşmesi

UART, Universal Asynchronous Receiver/Transmitter`ın kısaltmasıdır ve seri haberleşme için kullanılan yaygın bir iletişim protokolünü temsil eder. Bu protokol, bilgisayarlar, mikrodenetleyiciler, sensörler ve modüller şeklinde çeşitli elektronik cihazlar içinde veri iletimi için kullanılır. UART haberleşmesi, basitliği ve geniş uygulama alanı nedeniyle elektronik ve gömülü sistem projelerinde çoğunlukla tercih edilir [13].

UART, "asenkron" bir protokol olduğundan saat sinyali olmadan çalışır. Bu özelliği, iki cihaz arasında zamanlama senkronizasyonu için özel bir saat sinyaline gereksinim olmadan iletişimi mümkün kılar.

UART haberleşmesi iki temel bileşene sahiptir:

Veri Gönderme (Transmit): Veri, gönderici cihazda seri olarak iletilmek üzere paketlenir. Bu işlemde veri bitleri ardışık olarak alıcı cihaza gönderilir.

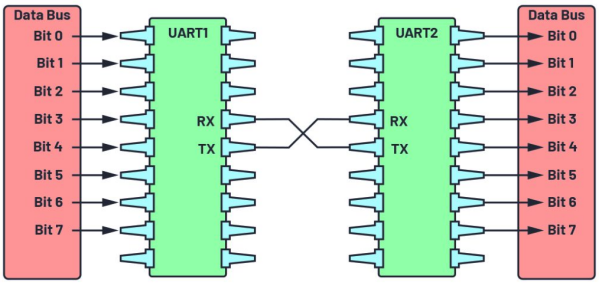
Veri Alma (Receive): Gelen veri, alıcı cihaz tarafınca alınır ve işlenmek üzere bölümlere ayrılır. Alıcı cihaz, gelen veriyi çözerek bilgiye dönüştürür.

UART bağlantısı çoğu zaman iki temel tel kullanır:

TX (Transmit): Veri gönderimini sağlayan hat.

RX (Receive): Veri alımını sağlayan hat.

Şekil 2:3'de Uart haberleşme bağlantıları gözlemlenmektedir.



Şekil 2:3 Uart Haberleşmesi [34]

UART, asenkron olduğundan veri iletimi ve alımı için belirli bir zamanlama gereklidir. Bu nedenle, baud hızı ismi verilen bir terimle ifade edilen, saniyede iletilen bit sayısı ile zamanlama senkronizasyonu sağlanır. İki cihazın doğru bir şekilde iletişim kurabilmesi için, her ikisinin de aynı baud hızında çalışması gerekir. Yaygın baud hızları, 9600, 19200, 38400, 57600, 115200 şeklinde değerlerdir.

UART haberleşmesinin bir diğer önemli özelliği, hata kontrolü ve doğrulama için kullanılan "start" ve "stop" bitleridir. Her veri iletimi bir start bit ile başlar ve bir ya da iki stop bit ile sona erer. Bu bitler, veri iletiminde başlangıç ve bitiş noktalarını belirler, böylece veri aktarımında doğruluk sağlanır. İsteğe bağlı olarak, iletimdeki hataları kontrol etmek için "parite" bitleri de kullanılabilir. Parite, iletilen bitlerin toplamının çift ya da tek olmasıyla doğrulama sağlar.

Sonuç olarak, UART haberleşmesi, basitliği, esnekliği ve geniş uygulama alanıyla öne çıkar. Elektronik cihazlar arasında seri bağlantılar, mikrodenetleyicilerin programlanması, sensör verilerinin iletimi ve diğer birçok uygulamada yaygın olarak

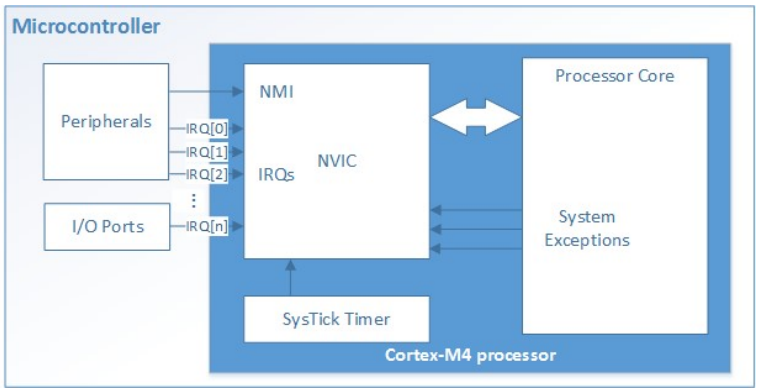
kullanılır. UART, aynı zamanda düşük maliyeti ve düşük güç tüketimiyle de gömülü sistemler için ideal bir çözümdür [14].

2.1.4 Interrupt

"Interrupt" (kesme), mikrodenetleyiciler ve diğer gömülü sistemlerde kullanılan bir konseptdir. Kesmeler, normal program akışını geçici olarak kesen ve bir olay meydana geldiğinde mikrodenetleyiciye veya işlemciye dikkat çekmek için kullanılan bir mekanizmadır. Bir kesme gerçekleştiğinde, işlemci normal çalışmasını durdurur, belirli bir kesme hizmet rutini (interrupt service routine - ISR) çalıştırılır ve ardından normal çalışma durumuna geri dönülür [15].

İşte kesmelerle ilgili bazı temel kavramlar:

- Kesme Kaynağı (Interrupt Source): Kesme, genellikle bir harici olay veya bir iç olay (yani bir zamanlayıcı tarafından belirli bir zaman aralığında tetiklenen bir kesme) tarafından başlatılır.
- Kesme Vektörü (Interrupt Vector): Kesme vektörü, bir kesmenin hangi ISR'yi çalıştıracağını belirten bir adrestir. Mikrodenetleyicinin kesme tablosu içinde bu vektörler bulunur.
- ISR (Interrupt Service Routine): ISR, bir kesme gerçekleştiğinde çalıştırılan özel bir alt programdır. Bu rutin, genellikle belirli bir kesme olayına ilişkin işlemleri gerçekleştirir ve ardından normal program akışına döner.
- Kesme Kontrol Register (Interrupt Control Register): Bir mikrodenetleyici veya işlemci, kesmeleri etkinleştirmek veya devre dışı bırakmak için kullanılan kesme kontrol registerlarına sahiptir.
- Öncelikli Kesme (Priority Interrupt): Bazı sistemlerde, birden çok kesme kaynağı bulunabilir ve bu kesmeler farklı önceliklere sahip olabilir. Öncelikli kesmeler, daha öncelikli bir kesme gerçekleştiğinde diğer kesmeleri geçersiz kılabilir.



Şekil 2:4 Interrupt Mikrodenetleyici Block Diyagramı [16]

Şekil 2:4'te Kesmeler için block diyagramını inceliyoruz. Kesmeler, zamanlama, harici olaylara tepki, düşük güç tüketimi ve gerçek zamanlı sistemler gibi birçok uygulamada önemli bir rol oynar. İşlemcinin normal program akışını bozmadan özel işlemleri gerçekleştirmesine olanak tanıyarak sistemlerin daha etkili ve hızlı çalışmasına yardımcı olurlar.

2.1.5 İvmeölçer ve Jiroskop

İvmeölçer (Accelerometer) ve jiroskop (Gyroscope), hareket ve yönlendirmeyle ilgili verileri elde eden sensörlerdir. Bu sensörler, özellikle mobil cihazlar, drone`lar, oyun kontrolcüleri ve giyilebilir teknoloji şeklinde hareket algılama ve konum belirleme gerektiren birçok uygulamada kullanılır. MSP430 ya da STM32 gibi mikrodenetleyiciler, bu sensörlerden gelen verileri işleyip analiz ederek çeşitli sonuçlar elde edebilir [17].

İvmeölçerler, nesnelerin hızlanmasını (ivme) algılar ve çoğu zaman üç eksende ölçüm yapar: X, Y ve Z. Bu eksenler, cihazın hangi yönde hareket ettiğini ya da yerçekimine karşı hangi konumda bulunduğunu belirlemeye destek olur.

- Çalışma Prensibi: İvmeölçerler, piezoelektrik, kapasitif ya da MEMS (MicroElectroMechanical Systems) teknolojisiyle çalışır. İvme sonucu içindeki kütleler hareket eder ve bu hareket, elektriksel bir sinyale

dönüştürülür. Bu sinyal, mikrodenetleyici tarafınca okunarak hızlanma ya da yerçekimi ivmesi hesaplanır.

- Uygulamalar: İvmeölçerler, cihazların yönünü belirlemek, hareketi algılamak ve cihazın eğimini ölçmek için kullanılır. Örneğin, akıllı telefonlarda ekranın yatay ya da dikey yönünü belirlemek, fitness cihazlarında hareketi algılamak ve drone'larda stabiliteyi sağlamak için yaygın olarak kullanılır.

Jiroskoplar, nesnelerin açısal hızını ölçer ve cihazın dönme hareketini algılar. İvmeölçerler şeklinde jiroskoplar da çoğu zaman üç eksende ölçüm yapar: X, Y ve Z.

- Çalışma Prensibi: Modern jiroskoplar, MEMS teknolojisiyle çalışır. Genellikle içlerinde dönen bir rotor ya da titreşimli bir eleman bulunur. Açısal hızdaki değişiklikler bu elemanlar tarafından algılanır ve elektriksel sinyallere dönüştürülür. Bu sinyaller, mikrodenetleyici tarafından okunarak cihazın dönme hareketi hesaplanır.
- Uygulamalar: Jiroskoplar, hareket algılama, stabilizasyon ve yönlendirme uygulamalarında kullanılır. Drone'lar, robotlar, oyun kontrolcüleri ve araçlarda yönlendirme sistemleri gibi birçok uygulamada jiroskoplar kritik bir rol oynar.

İvmeölçer ve jiroskop birlikte Birçok cihazda, özellikle hareket ve konum algılaması ihtiyaç duyulan sistemlerde, ivmeölçer ve jiroskop beraber kullanılır. İvmeölçer, cihazın yerçekimine karşı konumunu belirlerken, jiroskop açısal hareketi ve dönüşleri algılar. Bu kombinasyon, cihazların yönünü, hareketini ve stabilitesini daha duyarlı bir biçimde kontrol etmeyi sağlar.

Birçok mikrodenetleyici, bu sensörlerden gelen verileri okuma ve işleme kabiliyetine sahiptir. İvmeölçer ve jiroskop verileri, hareket algılama, eğim tespiti, stabilizasyon ve konum belirleme için kullanılabilir. Ayrıca, bu sensörlerin kombinasyonu, hareketli cihazların davranışını ve hareket paternlerini belirlemek için kuvvetli bir araç sağlar.

2.1.6 Manyetometre

Manyetometre, manyetik alanları ölçen ve doğrultu belirleme uygulamalarında kullanılan bir sensördür. Bu sensörler, özellikle mobil cihazlar, drone`lar, otomotiv sistemleri ve çeşitli navigasyon uygulamalarında yaygın olarak kullanılır. MSP430 ya da STM32 benzer biçimde mikrodenetleyiciler, manyetometrelerden gelen verileri işleyerek cihazın yönünü belirlemek için kullanabilir [18].

Manyetometreler, çoğu zaman cihazın yerel manyetik alanını, yani Dünya'nın manyetik alanını ölçer ve bu sayede cihazın manyetik kuzeye göre yönünü belirler. Çoğu manyetometre üç eksende ölçüm yapar: X, Y ve Z eksenleri. Bu eksenler, cihazın uzaydaki oryantasyonunu ve yönünü belirlemeye yardımcı olur.

• Çalışma Prensibi: Manyetometreler, manyetik alanları algılamak için Hall etkisi, fluxgate ya da MEMS (MicroElectroMechanical Systems) teknolojisi kullanır. Hall etkisi manyetometrelerde, manyetik alanın neden olduğu voltaj değişiklikleri algılanır ve bu değişiklikler elektriksel sinyallere dönüştürülür. Fluxgate manyetometrelerde, manyetik alanın etkisiyle bir çekirdeğin mıknatıslanması ve demagnetizasyonu kullanılarak sinyaller elde edilir. MEMS manyetometrelerde ise mikroskobik sensörler manyetik alan değişikliklerini algılar ve bu değişimler elektriksel sinyallere dönüştürülür. Mikrodenetleyiciler bu sinyalleri okuyarak cihazın manyetik alanını ve dolayısıyla yönünü hesaplar[18].
• Uygulamalar: Manyetometreler, çoğu zaman cihazların yönünü belirlemek ve navigasyon uygulamalarında kullanılır. Örneğin, akıllı telefonlarda dijital pusula işlevi sağlamak, drone'larda yönlendirme ve konum belirlemek, otomotiv sistemlerinde GPS ile birlikte yön tayini yapmak için yaygın olarak kullanılır.

Manyetometreler, ivmeölçer ve jiroskoplarla beraber kullanıldığında daha kırılgan ve emin bir doğrultu ve hareket algılama sistemi sağlar. İvmeölçer, cihazın yerçekimine karşı konumunu belirlerken, jiroskop açısal hareketleri ve dönüşleri algılar, manyetometre ise cihazın manyetik kuzeye göre yönünü belirler. Bu kombinasyon, cihazların hareket ve konumunu daha doğru bir biçimde takip etmeyi ve test etmeyi sağlar.

Mikrodenetleyiciler, bu sensörlerden gelen verileri okuyup işleyerek hareket algılama, yön belirleme, stabilizasyon ve navigasyon uygulamalarında kullanabilir. Manyetometre, ivmeölçer ve jiroskop verileri, cihazların uzaydaki hareket ve yönelimini anlamak ve test etmek için güçlü bir araç seti sunar. Bu sensörlerin birleşimi, hareketli cihazların davranışını ve hareket paternlerini belirlemek ve optimize etmek için kritik öneme sahiptir.

2.2 Benzer Projeler ve Uygulamalar

Benzer projeler, kullanıcı etkileşimi ve hareket algılama üzerine odaklanan uygulamalardır. Atari salonlarındaki silah oyunları, atarideki Duck Hunter ve televizyonlar için airmouse gibi projeler, kablosuz iletişim ve hareket algılama teknolojileriyle ilgilidir. Ancak bu projelerin bazı eksik yönleri bulunmaktadır:

- Atari salonlarındaki silah oyunları, kullanıcıların bir tabanca benzeri cihazla ekrana ateş ederek hedefleri vurmaya çalıştığı popüler bir oyun türüdür. Ancak, bu oyunlar genellikle sabit bir ortama bağlıdır ve evde kullanılacak şekilde tasarlanmamıştır. Ayrıca, kullanılan ışık tabancaları ve kızılötesi sensörler, yüksek hassasiyet gerektiren oyunlarda yanıltıcı olabilir ve zamanla kalibrasyon sorunlarına yol açabilir. İvmeölçer ve jiroskop gibi sensörler yerine basit kızılötesi algılama kullanılması, hareket algılamadaki hassasiyeti sınırlayabilir.
- Atari'deki Duck Hunter, klasik atari oyunlarından biridir ve oyuncuların bir av tüfeği kullanarak uçan ördekleri vurmalarını içerir. Bu tür oyunlar, oyuncuların nişan alma doğruluğunu kontrol etmek için ışık sensörleri kullanır, ancak gerçek dünya koşullarında çalışmaya uygun değildir. Örneğin, ortam ışığı değişiklikleri ve ekran parlamaları, ışık sensörlerinin doğru çalışmasını engelleyebilir. Ayrıca, oyun, gerçek dünya hareketlerinin tüm nüanslarını yakalamaz; bu da kullanıcıların daha karmaşık hareketleri gerçekleştirmesine engel olabilir. Şekil 2:5 ‘de oyundan alıntı bir görüntü gözlenmektedir.

Şekil 2:5 Duck Hunter

- Televizyonlar için Airmouse, kullanıcının el hareketlerini algılayan ve bir televizyon veya bilgisayar ekranında imleci kontrol eden bir cihazdır. Airmouse gibi cihazlar, genellikle ivmeölçer ve jiroskop sensörleri içerir, ancak sınırlı hareket doğruluğu ve kalibrasyon sorunlarıyla karşılaşabilirler. Ayrıca, Bluetooth teknolojisi bazen parazitlere maruz kalabilir, bu da imleç kontrolünün dalgalanmasına veya gecikmesine neden olabilir. Bu tür cihazların diğer Bluetooth cihazlarıyla etkileşime girmesi de potansiyel bağlantı sorunlarına yol açabilir.

Bu projeler, kullanıcı etkileşimi ve hareket algılama alanlarında temel bir anlayış gösterse de, modern gömülü sistem projeleri için yeterli esneklik ve doğruluk sunmayabilirler. İvmeölçer ve jiroskop gibi sensörlerin daha karmaşık uygulamalar için doğru şekilde çalışmasını sağlamak için ek kalibrasyon ve optimizasyon gereklidir. Ayrıca, kablosuz iletişimdeki parazit sorunlarının üstesinden gelmek için daha dayanıklı protokoller ve donanım tasarımları gerekebilir. Bu eksiklikler, MSP430G2553, HC-05, MPU6050 ve HMC5883L gibi bileşenlerle daha modern ve kesin projelerin geliştirilmesine olanak sağlamıştır.

3. KULLANILAN CİHAZLAR VE PROGRAMLAR

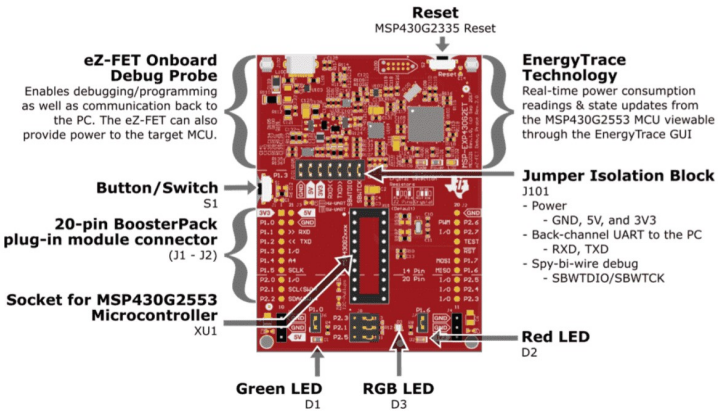
Bitirme tezinin bu kısmında tasarım sırasında kullanılan cihazların özelliklerinden ve bu cihazları kontrol eden programların nasıl çalıştıklarından bahsedilmiştir

3.1 Kullanılan Cihazlar

Bu bitirme tezinde, Bluetooth aracılığıyla kablosuz veri iletimi gerçekleştiren ve MSP430G2553 mikrodenetleyicisini temel alan bir sistemin tasarımı ve uygulanması ele alınmıştır. Bu tasarım kapsamında, MSP430 LaunchPad geliştirme kartı üzerinde yer alan MSP430G2553 mikrodenetleyicisi, HC-05 Bluetooth modülü, MPU6050 ivmeölçer ve jiroskop sensörü ve HMC5883L manyetometre sensörü kullanılarak bir cihaz tasarlanmıştır.

3.1.1 MSP430 Launchpad Geliştirme Aracı

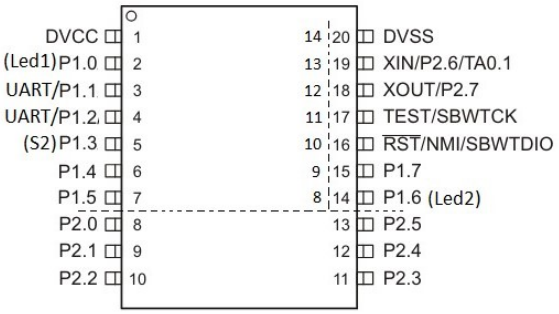
Texas Instruments firmasının geliştirdiği MSP430 Launchpad, basit ve ucuz olmasının yanı sıra, üzerindeki butonlar, LED ışıklar, programlama ve hata ayıklama bölümleri ile kullanıcılarına MSP430G2xx serisi mikroişlemcileri kullanmalarında kolaylık sağlar [1]. USB üzerinden hata ayıklama ve programlama imkanı sunan cihaz Şekil 3:1'de gösterilmiştir.



Şekil 3:1 MSP430 Launchpad Geliştirme Aracı [19]

MSP430 ailesi, ücretsiz yazılım sağlaması, ucuz mikroişlemci ve düşük güç harcaması gibi sebeplerden proje geliştirmek için ideal bir cihazdır. Birçok 16 bitlik

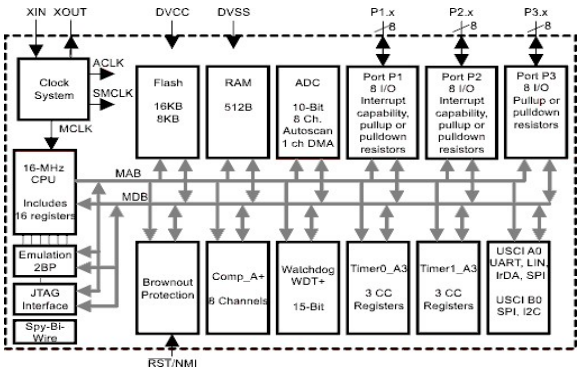
MSP430 mikroişlemcisiyle uyumlu olan bu cihaz 9600 baud hızına kadar Evrensel Eşzamansız Alıcı-Verici (Universal Asynchronous Receiver Transmitter – UART) seri iletişim sağlar [1]. Hem 14 bacaklı hem de 20 bacaklı mikroişlemciler kullanılarak uygulama geliştirilebilir. Cihazın bacak dağılımı Şekil 3:2'de görüldüğü gibidir.



Şekil 3:2 MSP430 Launchpad bacak dağılımı

3.1.1.1 MSP430G2553 Mikroişlemcisi

MSP430 Launchpad geliştirme aracı ile beraber gelen MSP430G2553 mikroişlemcisi 16 bitlik ve 20 bacaklı bir yapıda olup düşük besleme gerilimi aralığı ile beş farklı düşük güç tüketim modu içerir [20]. Bu modlar sayesinde uzun batarya ömrü amaçlanan taşınabilir uygulamalar için idealdir. Sahip olduğu 16 bitlik İndirgenmiş Komut Takımıyla Hesaplama (Reduced Instruction Set Computing – RISC) bilgisayarı, 16 bitlik saklayıcıları ve sabit katsayı üreticileri ile maksimum kod verimliliği sağlarlar. Mikroişlemcinin mimari yapısı Şekil 3:3'teki gibidir.



Şekil 3:3 MSP430G2553 Mimari Yapısı [20]

Cihaz içinde yerleşik olarak 16 bitlik sayıcılar, 24 genel amaçlı girdi çıktı bacağı, çok yönlü analog karşılaştırıcı ve 10 bitlik analog sayısal çevirici bulunur [20]. En sık kullanıldığı uygulamalar, ucuz algılayıcılarla analog sinyallerin yakalanıp sayısal değerlere çevrilerek, elde edilen verinin bir ekranda gösterilmesi veya başka bir sisteme gönderilmesi şeklindedir.

3.1.2 HC-05 modülü

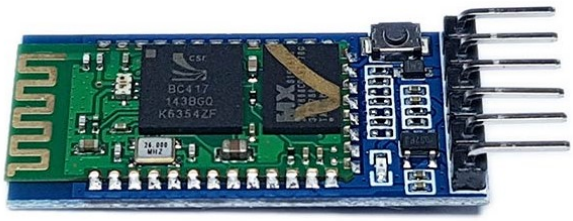
HC-05, Bluetooth özellikli bir seri haberleşme modülüdür. Bu modül, düşük maliyetli ve yaygın olarak kullanılan bir Bluetooth bağlantısı sağlar ve genellikle mikrodenetleyiciler, Arduino ve diğer gömülü sistemlerde kablosuz iletişim amacıyla kullanılır.

İşte HC-05 modülü hakkında bazı temel bilgiler:

- Bluetooth Versiyonu: HC-05, Bluetooth 2.0 standardına dayanmaktadır. Bluetooth 2.0, önceki versiyonlara kıyasla daha iyi hız ve güvenilirlik sunar.
- Çalışma Modları: Master (Anahtarlayıcı) Modu: Diğer Bluetooth cihazlarına bağlanabilir [2].
  - ◊ Slave(Köle) Modu: Diğer cihazlara bağlanabilir ve onların komutlarını kabul edebilir.
  - ◊ Master/Slave (Anahtarlayıcı/Köle) Modu: Hem anahtarlayıcı hem de köle olarak aynı anda çalışabilir.
- Bağlantı Aralığı: HC-05 modülünün tipik bağlantı aralığı, açık alanda yaklaşık 10 metreye kadar olabilir [2].
- Hız ve Veri Transferi: HC-05, seri haberleşme hızını destekleyerek (baud rate) farklı hızlarda veri transferine olanak tanır [2].
- AT Komutları: HC-05 modülü, AT komutları ile yapılandırılabilir. Bu komutlar, modülün çalışma modu, ismi, bağlantı parametreleri gibi çeşitli ayarlarını değiştirmek için kullanılır [2].
- Elektriksel Özellikler: HC-05 genellikle 3.3V besleme gerilimine ihtiyaç duyar. TTL seviyesinde seri haberleşme sinyallerini kullanır [2].

- Pairing ve Eşleştirme: HC-05, diğer Bluetooth cihazları ile eşleşebilir ve güvenli bir bağlantı kurmak için kullanıcıdan bir eşleştirme işlemi talep edebilir [2].
- Giriş/Çıkış Pinleri: HC-05 modülü genellikle VCC (besleme), GND (toprak), TX (veri gönderme), RX (veri alma) gibi temel giriş/çıkış pinlerine sahiptir [2].

HC-05 modülü, genellikle Arduino projelerinde ve diğer gömülü sistem uygulamalarında kablosuz iletişim sağlamak için kullanılır. Projeler arasında veri iletimi, kablosuz kontrol ve Bluetooth üzerinden seri haberleşme gibi uygulamalar için oldukça popülerdir. Modül, kullanımı kolaydır ve geniş bir kullanıcı kitlesi tarafından benimsenmiştir. HC-05 modülü Şekil 3:4'de gösterilmektedir.



Şekil 3:4 HC-05 Modülü [21]

3.1.3 MPU-6050

MPU-6050, MEMS (Micro-Electro-Mechanical Systems) teknolojisi kullanılarak üretilmiş bir ivmeölçer ve jiroskop sensörüdür. İki temel sensörü tek bir entegre devrede birleştirerek, hem ivme ölçümü bununla beraber açısal hız ölçümü sağlayan bu sensör, hareket algılama ve kontrol uygulamalarında yaygın olarak kullanılır [22].

Özellikleri:

- 6 Eksenli Sensör: MPU-6050, üç eksenli ivmeölçer ve üç eksenli jiroskopu içerir. İvmeölçer, doğrusal ivme ölçümü için X, Y ve Z eksenlerinde veriler sağlarken, jiroskop ise açısal hız ölçümü için aynı eksenlerde veriler sunar [3].
- I2C ve SPI Arabirimi: MPU-6050, mikrodenetleyicilerle bağlantı için I2C (Inter-Integrated Circuit) ve SPI (Serial Peripheral Interface) arabirimlerini destekler 23]. I2C çoğu zaman daha yaygın kullanılan bir protokoldür .

- İçsel Dijital Hareket İşlemcisi (DMP): Bu sensör, ivme ve açısal hız verilerini işleyebilen bir Dijital Hareket İşlemcisi içerir [3]. DMP, ham veriyi işleyerek, eğim açısı, yönelim ve hareket algılama gibi hesaplamaları yapabilir.
- Geniş Aralık da Ayarlanabilir Hassasiyet: MPU-6050, farklı uygulama gereksinimlerine müsait çeşitli duyarlılık aralıklarına sahiptir [3]. İvmeölçer için ±2g, ±4g, ±8g ve ±16g seçenekleri, jiroskop için ise ±250, ±500, ±1000 ve ±2000 derece/saniye aralıkları mevcuttur [3].
- Dahili Sıcaklık Sensörü: MPU-6050, çevresel faktörleri izlemek için yerleşik bir ısı sensörüne de sahiptir [3].

MPU-6050, çok çeşitli uygulamalarda kullanılır:

- Hareket Takibi ve Kontrol: Dronlar, insansız hava araçları ve robotlarda konum ve hareket kontrolü sağlamak için kullanılır.
- Oyun ve Eğlence: Hareket tabanlı kontrol cihazlarında ve sanal gerçeklik uygulamalarında hareket algılama için yaygın olarak tercih edilir.
- Fitness ve Sağlık: Adım sayar, kalori ölçer ve diğer fitness cihazlarında kullanıcı hareketlerini izlemek için kullanılır.
- Otomotiv: Araç stabilite kontrol sistemlerinde ve navigasyon uygulamalarında ivme ve jiroskop verileri kullanılır.

MPU-6050, çok yönlü ve kuvvetli bir sensör olarak, hem amatör hem de ustalaşmış projelerde yaygın olarak kullanılmaktadır. I2C protokolü ile mikrodenetleyicilere kolayca bağlanması ve kapsamlı işlevselliği, onu elektronik ve gömülü sistem projelerinde popüler bir seçim haline getirmektedir. MPU-6050 modülü Şekil 3:5'te gösterilmektedir.

Şekil 3:5 MPU-6050 modülü [23]

3.1.4 HMC5883L

HMC5883L, bir manyetik alan sensörüdür ve özellikle manyetik alanın ölçülmesi ve yönünün belirlenmesi için kullanılır. Bu sensör, üç eksenli manyetik alan algılama yeteneğine sahiptir, yani manyetik alanın X, Y ve Z eksenlerindeki bileşenlerini ölçebilir.

Özellikler:

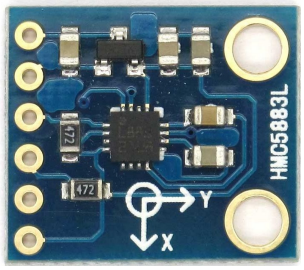
- Üç Eksenli Manyetik Alan Algılama: HMC5883L, manyetik alanın X, Y ve Z eksenlerindeki bileşenlerini ölçebilir, böylece üç boyutlu manyetik alanın tam bir resmini sağlar [4].
- Yüksek Hassasiyet: Sensör, yüksek hassasiyet seviyeleriyle manyetik alanın küçük değişikliklerini dahi hassas bir şekilde algılayabilir [4].
- Düşük Güç Tüketimi: Düşük güç tüketimi, pil ile çalışan cihazlar için idealdir ve uzun pil ömrü sağlar [4].
- Hızlı Örnekleme Hızı: Hızlı örnekleme hızı, gerçek zamanlı manyetik alan değişimlerini hızlı bir şekilde algılayabilir ve işleyebilir [4].
- Entegre Yapı: Entegre bir çip olarak gelir, bu da uygulamaya entegrasyonu kolaylaştırır ve dış bileşenlere olan ihtiyacı azaltır [4].
- Kompakt Boyut: Küçük boyutu sayesinde, birçok farklı cihazda kullanılabilir ve yerden tasarruf sağlar [4].

Kullanım Alanları:

- Elektronik Pusula Sistemleri: Manyetik alanın ölçülmesi ve yönünün belirlenmesi için idealdir, bu nedenle elektronik pusula sistemleri için sıklıkla kullanılır.
- Manyetik Alan Haritalama: Manyetik alanın çeşitli bölgelerdeki dağılımını haritalamak ve analiz etmek için kullanılabilir.
- Manyetik Materyal Tespiti: Manyetik materyallerin tespiti ve izlenmesi için kullanılabilir, örneğin metal tespit cihazlarında.
- Navigasyon Sistemleri: Hareketli cihazlarda (örneğin, dronlar, robotlar, araçlar) kullanılarak navigasyon sistemlerine entegre edilebilir.

- Endüstriyel Uygulamalar: Endüstriyel otomasyon, manyetik alan kontrolü ve izleme gibi birçok endüstriyel uygulamada kullanılabilir.
- Hobi Projeleri: Arduino gibi geliştirme platformlarıyla uyumlu olduğu için hobi projelerinde ve maker topluluklarında sıkça kullanılır.

HMC5883L, geniş bir kullanım yelpazesine sahip çok yönlü bir sensördür ve manyetik alanın ölçülmesi ve izlenmesi gereken birçok farklı uygulamada kullanılabilir. HMC5883L modülü Şekil 3:6'da gösterilmektedir.



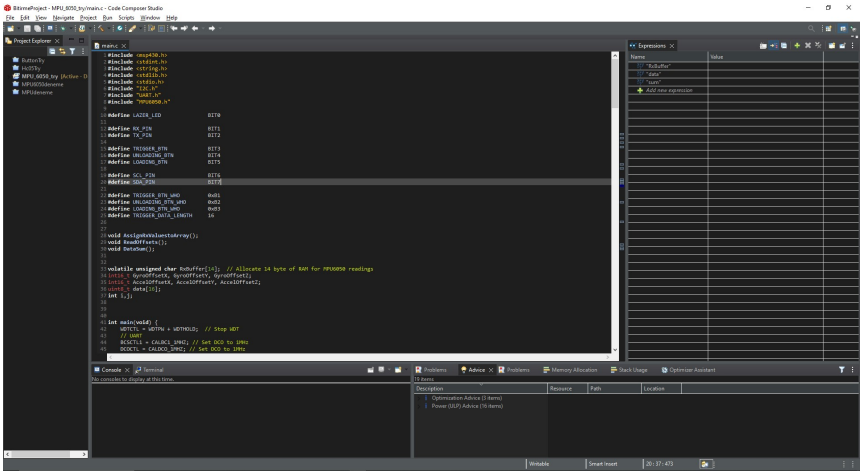
Şekil 3:6 HMC5883L Modülü [24]

3.2 Kullanılan Programlar

Bitirme tezinin bu kısmında tasarımdaki cihazları programlamak, kontrol etmek ve görüntülemek için kullanılan programlardan bahsedilecektir.

3.2.1 Kod Tertipleyen Stüdyo (CCS) Programı

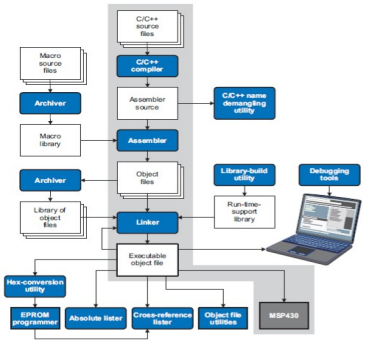
MSP430G2553 mikroişlemcisi programlanırken, Texas Instruments firmasının kendi ürünleri için geliştirdiği Kod Tertipleyen Stüdyo (Code Composer Studio - CCS) Programı kullanılmıştır [25]. Bu program ile MSP430 mikroişlemcilerinde program geliştirme, hata ayıklama ve enerji tüketimini görüntüleme gibi işlemler yapılabilir. Kolay ve anlaşılır ara yüzü kullanıcılara büyük kolaylık sağlamaktadır. Programlama yapıldıktan sonra cihaza yüklemeden önce yapılan hatalar, yazılımla ilgili uyarılar ve öneriler ayrı ayrı gösterilerek kullanıcının mikroişlemciyi en verimli şekilde kullanması sağlanmıştır. Programın bilgisayar ekranındaki görüntüsü Şekil 3.7'deki gibi gösterilmiştir.



Şekil 3:7 CCS Programının Görüntüsü.

3.2.1.1 C Programlama Dili

CCS Programı hem çevirici (assembly) dilinde hem de C programlama dilinde programlama olanağı sunmaktadır. Ancak hem kolaylığı hem de daha modern olması açısından bu bitirme tezinde C programlama dili tercih edilmiştir. CCS içinde bulunan derleyici C kodunu derleyerek çeviriciye gönderir. Çevirici tarafından makine koduna çevrilen yazılım bağ düzenleyicisi aracılığıyla MSP430 üzerine yüklenir. CCS programı içindeki derleyici araçları, yürütme hızını arttırmak, C kodunun boyutunu azaltıp basitleştirmek, komutları ve deyimleri yeniden düzenlemek ve değişkenleri saklayıcılara atamak gibi işlemleri gerçekleştirerek mikroişlemciye yüklenecek yazılımın en iyi duruma gelmesini sağlarlar. Bu yazılım geliştirme akışı Şekil 3.8'deki gibi gösterilmiştir.



Şekil 3:8 CCS Yazılım Geliştirme Akışı [26]

3.2.2 Visual Studio

Visual Studio, Microsoft tarafından geliştirilen kapsamlı bir entegre geliştirme ortamıdır (IDE). Yazılım geliştiricilerinin farklı programlama dilleri ve platformlar için uygulama oluşturmasına imkan tanır. Visual Studio, masaüstü, mobil, internet ve bulut tabanlı uygulamalar dahil olmak üzere geniş bir yelpazede geliştirme desteği sunar [27].

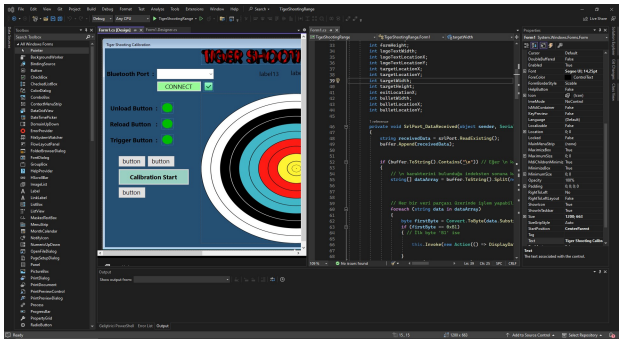
Visual Studio`nun temel özellikleri ve bileşenleri şunlardır:

- Çok Yönlü Programlama Dili Desteği: Visual Studio, C#, Visual Basic, F#, C++, Python, JavaScript, TypeScript gibi birçok popüler programlama dilini destekler [27].
- Proje Türleri ve Şablonlar: IDE, farklı türde projeler kurmak için önceden tanımlanmış şablonlar sunar. Windows Forms ve WPF masaüstü uygulamalarından ASP.NET internet uygulamalarına, Xamarin ile mobil geliştirmeden Unity tabanlı oyunlara kadar çeşitli alanlarda proje kurmak mümkündür [27].
- Derleme ve Derleyiciler: Visual Studio, çeşitli derleyicilerle birlikte gelir ve derleme sürecini kolaylaştırır. Derleme sırasında hataları test eder ve hızlı bir biçimde hata ayıklama imkanı sunar [27].
- Gelişmiş Hata Ayıklama ve Profil Analizi: Visual Studio, hata ayıklama işlemini kolaylaştırmak için araçlar sağlar. Breakpoint'ler, izleme değişkenleri, çağrı yığını analizi gibi özellikler, geliştiricilere kodlarındaki sorunları belirleme ve düzeltme kabiliyeti verir. Profil analizi araçları, uygulamanın performansını ve kaynak kullanımını incelemek için kullanılır [27].
- Kaynak Kontrol ve Takım Çalışması: Visual Studio, Git gibi popüler kaynak test sistemlerini destekler [27]. Bu, geliştiricilerin ekip çalışması yaparak projelerini yönetmelerine, kod incelemeleri yapmalarına ve versiyon kontrolü sağlamalarına imkan tanır.
- Test ve Sürekli Entegrasyon: IDE, yazılım testleri oluşturmak ve yürütmek için entegre kontrol araçları sunar [27]. Ayrıca, devamlı entegrasyon ve

devamlı teslimat (CI/CD) için Azure DevOps ve GitHub gibi hizmetlerle entegre çalışır.

- Geniş Eklenti Ekosistemi: Visual Studio, zengin bir eklenti ve uzantı ekosistemine sahiptir [27]. Geliştiriciler, IDE'nin işlevselliğini artırmak ve özelleştirmek için yüzlerce eklentiye erişebilir.

Visual Studio'nun sunduğu bu özellikler, yazılım geliştiricilerine kuvvetli ve esnek bir platform sağlayarak, farklı türde projeleri süratli ve etkili bir biçimde geliştirmelerine olanak tanır. IDE'nin kullanıcı dostu arayüzü ve kapsamlı belgeleri, hem yeni başlayanlar bununla birlikte deneyimli geliştiriciler için uygun bir geliştirme ortamı sunar. Programın bilgisayar ekranındaki görüntüsü Şekil 3.9'daki gibi gösterilmiştir.



Şekil 3:9 Visual Studio Programının Görüntüsü

3.2.2.1 Windows Form App

Windows Form App, Microsoft'un .NET platformu üzerinde çalışan ve Windows işletim sistemleri için masaüstü uygulamaları geliştirmek amacıyla kullanılan bir programlama çerçevesidir [28]. Kullanıcı dostu grafiksel arayüzler oluşturmak için kontroller, bileşenler ve araçlar sağlayarak, geliştiricilerin zengin özelliklere sahip uygulamalar oluşturmasına yardımcı olur.

Windows Form App, kullanıcı etkileşimi için düğmeler, metin kutuları, etiketler ve daha birçok kontrol içerir. Bu uygulama çerçevesi, karmaşık masaüstü uygulamalarından basit araçlara kadar geniş bir yelpazede projeler geliştirmek için kullanılır. Aynı zamanda, veri bağlama, olay tabanlı programlama ve tasarımcı tabanlı geliştirme gibi özelliklerle kullanıcıların yazılım geliştirme sürecini hızlandırır ve kolaylaştırır [28].

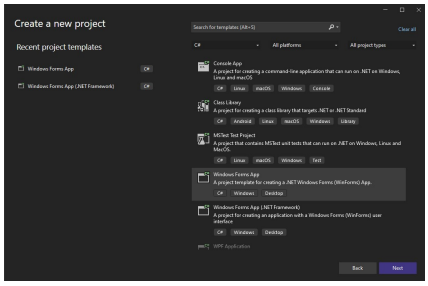
3.2.2.2 C# Programlama Dili

C# ise .NET platformu için tasarlanmış, nesne yönelimli bir programlama dilidir. Visual Studio ve C# bir arada kullanılarak çeşitli uygulamalar ve yazılımlar geliştirilebilir. Visual Studio, programlama sürecinde geniş araç ve özellikler sunar ve çeşitli programlama dilleri içinde çalışmayı destekler [29].

Visual Studio, C# programlama dilini destekleyerek kullanıcıların farklı uygulama türlerini geliştirmesini sağlar. C# dilinin modernliği ve kolaylığı nedeniyle, Visual Studio ile C# kullanılarak uygulama geliştirme, yaygın bir yaklaşımdır. Visual Studio`da, C# kodu derleyici tarafından derlenir ve daha sonra .NET platformunda çalıştırılacak bir ara dil haline getirilir. Bu ara dil, C# kodunun çeşitli platformlarda ve ortam koşullarında çalıştırılabilmesine imkan tanır.

Visual Studio'da C# ile çalışırken, kod yazma, derleme, hata ayıklama, kontrol etme ve dağıtma gibi aşamalar için kapsamlı araçlar mevcuttur. Derleyici, C# kodunu derleyerek, uygulamanın çalıştırılmaya hazır hale gelmesini sağlar. Çevirici, derlenen kodu daha düşük seviyeli bir forma dönüştürür ve bağ düzenleyicisi, gerekli bağlama işlemlerini yaparak uygulamanın düzgün çalışmasını sağlar.

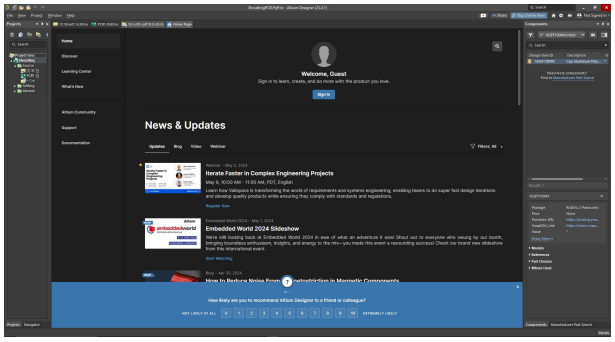
Visual Studio ve C#, kullanıcı dostu bir ortamda kuvvetli ve esnek uygulamalar geliştirmeyi sağlar. Bu entegre geliştirme ortamı, geniş bir uygulama yelpazesi için kullanılabilir ve C# programlama dilinin avantajlarını en üst düzeyde kullanma olanağı sunar. Programın bilgisayar ekranındaki görüntüsü Şekil 3.10’daki gibi gösterilmiştir [29].



Şekil 3:10 Visual Studio Yeni Proje Başlatma Görüntüsü

3.2.3 Altium Designer

Altium Designer, elektronik mühendisleri ve devre kartı tasarımcıları için kapsamlı bir entegre geliştirme platformudur. Şematik tasarım ve baskı devre kartı (PCB) tasarımını aynı ortamda birleştirir. Şematik tasarımda, mühendisler elektronik bileşenleri yerleştirir ve elektriksel bağlantıları tanımlar. Baskı devre kartı tasarımında, bileşenlerin fizyolojik yerleşimi ve elektriksel yollar belirlenir. Altium Designer, otomatik yerleştirme ve yönlendirme (auto-routing) gibi özelliklerle PCB tasarımını kolaylaştırır [30]. Yazılım, elektriksel kontroller ve simülasyon özellikleriyle hataları saptamaya ve çözmeye destek olur . Proje yönetimi için de araçlar sunan Altium Designer, devre kartının üretimi için Gerber dosyaları ve parça listeleri (BOM) şeklinde belgeler oluşturur. Yazılım, kuvvetli bileşen kütüphanesi ve kullanıcı dostu arayüzüyle elektronik tasarım sürecini kolaylaştırır ve hızlandırır. Programın bilgisayar ekranındaki görüntüsü Şekil 3.11'deki gibi gösterilmiştir.



Şekil 3:11 Altium Designer Programının Görüntüsü

4. DONANIM

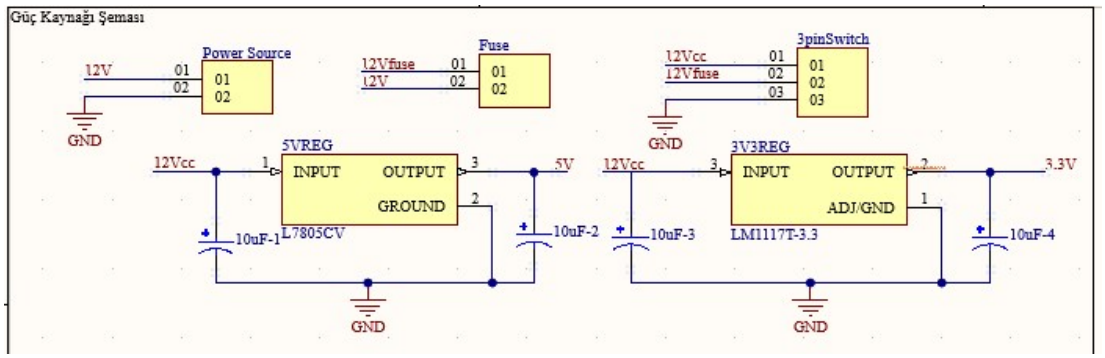
Bu bölümde, elektronik devre tasarımının temel adımları incelenecektir. Bu adımlar, devre şematiğinin Altium Designer kullanılarak oluşturulmasından, baskı devre kartının (PCB) tasarımına ve nihayetinde bakır plakaya baskıya kadar uzanır. Her aşama, projenin işlevselliği ve güvenilirliği için kritik öneme sahiptir. İyi tasarlanmış bir şematik, bileşenlerin ve bağlantıların doğru bir şekilde düzenlenmesini sağlar. PCB tasarımı, devrenin fiziksel yapısını oluştururken, baskı işlemi tasarımı gerçek bir devre kartına dönüştürür. Bu giriş paragrafında, donanım tasarımının bu temel aşamalarına dair genel bir bakış sunulacaktır.

4.1 Şematik Tasarımı

Bu bölümünde, projenin önemli bileşenlerinin ve alt sistemlerinin nasıl bağlantılı olduğunu gösteren devre şemaları sunulacaktır.

4.1.1 Güç Kaynağı Şeması

Bu şemada, projede kullanılan ana güç kaynağının bileşenleri ve bağlantıları gösterilecektir. Güç kaynağı, projenin tüm elektriksel ihtiyaçlarını karşılamak için regülatörler ve kondansatörlerle birlikte tasarlanmıştır. Şema, giriş voltajının nasıl regüle edildiğini ve farklı voltaj seviyelerinde çıkış sağlandığını gösterecektir. Şekil 4:1'de güç kaynağı şeması gösterilmiştir.



Şekil 4:1 Güç Kaynağı Şeması

Şemada yer alan ana bileşenler ve görevleri şunlardır:

- İkili Header: Voltaj girişi için kullanılan 2`li header, projenin harici güç kaynağına bağlanmasını sağlar. Bu bağlantı çoğu zaman bir güç adaptörü ya da benzeri bir kaynaktan gelir.
- Cam Sigorta: Cam sigorta, elektrik akımını korumak için kullanılır. Aşırı akım ya da kısa devre durumunda devreyi keserek bileşenleri korur. Sigorta, giriş gücünün güvenliğini sağlamak için önemli bir bileşendir.
- 3 Pin Switch: Güç kaynağını açmak ya da kapatmak için kullanılan 3 pinli bir anahtardır. Bu anahtar, kullanıcıya güç kaynağını kontrol etme imkânı sağlar. Açık konumda, elektrik akımının devreye geçmesine izin verir; kapalı konumda, akımı keser.
- L7805CV: L7805CV, çıkış voltajını 5V'a regüle eden bir lineer regülatördür. Bu regülatör, giriş voltajının 7V ile 35V içinde olmasına izin verir ve sabit bir 5V çıkış sağlar. Şemada bu regülatör, mikrodenetleyici ve diğer bileşenler için sabit bir voltaj sağlamak için kullanılır.
- LM1117T-3.3: LM1117T-3.3, çıkış voltajını 3.3V'a regüle eden başka bir lineer regülatördür. 5V'tan gelen akımı alır ve stabil bir 3.3V çıkış sağlar. Bu çıkış, düşük voltajlı bileşenler için kullanılır.
- 10uF Kapasitörler: Kapasitörler, elektriksel dalgalanmaları ve gürültüyü azaltmak için kullanılır. Şemada, giriş ve çıkış voltajlarında stabiliteyi artırmak için kullanılır. L7805CV ve LM1117T-3.3 regülatörlerinin giriş ve çıkışlarına yerleştirilen kapasitörler, elektriksel dalgalanmaları önlemeye destek olur.

Bu bileşenlerin birleşimi, projenin istikrarlı bir güç kaynağına haiz olmasını sağlar ve bileşenlerin güvenilir bir biçimde çalışmasını temin eder. Güç kaynağı şeması, projenin temelini oluşturur ve doğru bir şekilde tasarlanması, sistemin düzgün çalışması için kritik öneme sahiptir.

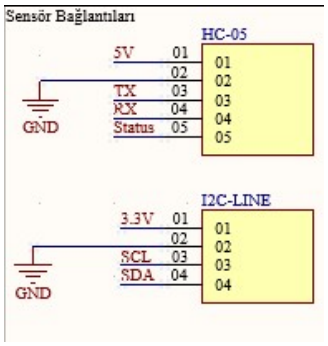
4.1.2 Sensör Bağlantı Şeması

Sensör Bağlantı Şeması, HC-05 Bluetooth modülü ve MPU6050 ve HMC5883L sensörlerinin mikrodenetleyiciye doğru şekilde bağlanmasını gösterir. Bu bağlantılar, kablosuz iletişim ve sensör verilerinin alınması için kritik öneme sahiptir.

Şemada yer alan ana bileşenler ve bağlantılar şunlardır:

- HC-05 Bluetooth Modülü: HC-05, UART üzerinden veri iletişimi sağlayan bir Bluetooth modülüdür. MSP430G2553 gibi mikrodenetleyicilerle kablosuz iletişim kurmak için kullanılır. Modülün bağlantıları şunlardır:
- VCC: 5V veya 3.3V güç kaynağına bağlanır. Projede kullanılan regülatörlerin çıkışlarından uygun voltaj sağlar.
- GND: Toprak bağlantısı için kullanılır. Mikrodenetleyici ve diğer bileşenlerle ortak bir toprak noktasını paylaşır.
- TX: Veri iletimi için kullanılır. Mikrodenetleyicinin RX pinine bağlanır, böylece mikrodenetleyici modülden gelen verileri alabilir.
- RX: Veri alımı için kullanılır. Mikrodenetleyicinin TX pinine bağlanır, böylece mikrodenetleyici modüle veri gönderebilir.
- I2C-LINE: MPU6050 ve HMC5883L, I2C (Inter-Integrated Circuit) protokolünü kullanarak mikrodenetleyici ile iletişim kurar. Sensörün bağlantıları şunlardır:
- SCL: I2C seri saat hattı için kullanılır. Mikrodenetleyicinin I2C saat (SCL) pinine bağlanır.
- SDA: I2C seri veri hattı için kullanılır. Mikrodenetleyicinin I2C veri (SDA) pinine bağlanır.

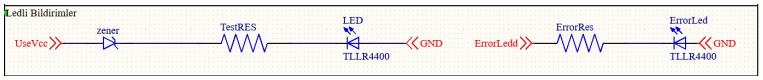
Bu bileşenlerin doğru şekilde bağlanması, sistemin güvenilir çalışmasını sağlar. HC-05 Bluetooth modülü ile kablosuz iletişim kurulur ve MPU6050 sensöründen hareket ve jiroskop verileri alınır. Bu bağlantılar, projenin temel iletişim ve sensör verisi işleme işlevlerini destekler. Şekil 4:2'de sensör bağlantı şeması gösterilmiştir.



Şekil 4:2 Sensör Bağlantı Şeması

4.1.3 Ledli Bildirim Şeması

LED'li Bildirim Şeması, elektronik devrelerde belirli durumları görsel olarak göstermek için kullanılan bir devre tasarımıdır. Bu şema, gelen voltajın doğru olup olmadığını ve MPU6050 sensör kartının çalışıp çalışmadığını göstermek için iki LED içerir. Şekil 4:3'te ledli bildirim şeması gösterilmiştir.



Şekil 4:3 Ledli Bildirim Şeması

İlk LED, gelen voltajın kontrolünü sağlamak için bir zener diyot ile bağlantılıdır. Zener diyotlar, belirli bir voltaj değerini tutmak için kullanılır ve voltaj regülasyonunu sağlar. Zener diyotu ve LED'in doğru bir şekilde çalışması, gelen voltajın 5V olup olmadığını kontrol etmek için önemlidir.

Birinci LED:

- Zener Diyot: Zener diyot, belirli bir voltaj değerini tutmak için kullanılır. Bu durumda, gelen voltajın 5V olup olmadığını kontrol eder. Zener diyotunun kırılma voltajı 5V'dir. Voltaj 5V'un üzerine çıktığında, zener diyot akımı yönlendirir ve bu durum LED'in yanmasına neden olur. Bu, hatalı veya aşırı voltajı göstermek için kullanılır.

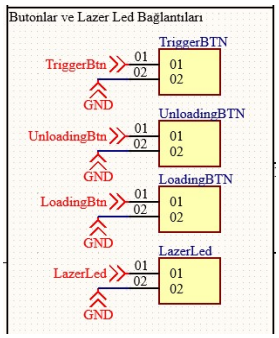
- LED: LED, gelen voltajın 5V olup olmadığını kontrol etmek için kullanılır. Eğer voltaj 5V veya daha yüksekse, LED yanar ve hata olduğunu belirtir.

İkinci LED, MPU6050 kartının kullanım durumunu kontrol etmek için kullanılır. Eğer sensör kartı çalışıyorsa, bu LED yanar ve sensörün kullanıldığını gösterir.

Bu şema, devredeki farklı durumları görsel olarak kontrol etmeyi sağlar. İlk LED, voltajın 5V olup olmadığını doğrularken, ikinci LED, MPU6050 sensör kartının kullanımda olup olmadığını gösterir. Bu tür bir bildirim şeması, devrenin doğru çalışmasını sağlamak ve olası sorunları erken tespit etmek için faydalıdır.

4.1.4 Butonların ve Lazer Ledin Bağlantı Şeması

Buton Bağlantı Şeması ve Lazer LED şeması, sistemde kullanıcı etkileşimini sağlayan bileşenlerin nasıl bağlandığını gösterir. Bu şemalarda, butonlar ve lazer LED'in MSP430G2553 mikrodenetleyicisi veya ilgili diğer bileşenlerle nasıl bağlantılı olduğu detaylandırılır. Şekil 4:4'te butonların ve lazer ledin bağlantı şeması gösterilmiştir.



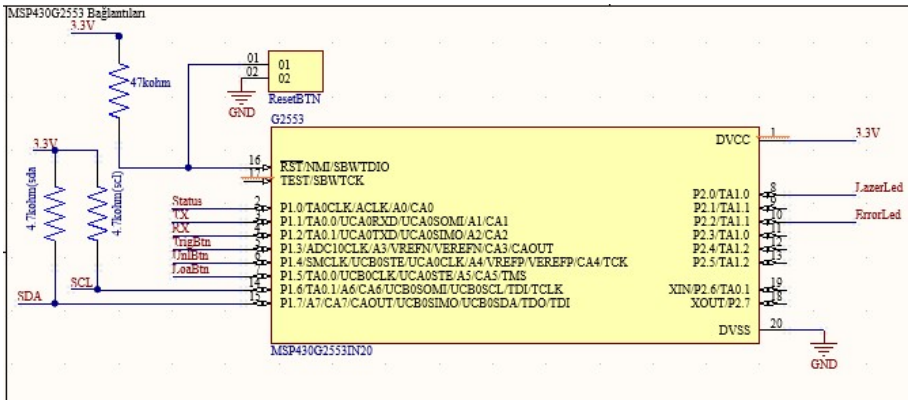
Şekil 4:4 Butonların ve Lazer Ledin Bağlantı Şeması

Butonlar ve Lazer Led Bağlantıları, sistemin kullanıcı etkileşimi ve işlevlerini sağlayan bileşenlerdir. Her bir butonun ve lazer LED'in farklı bir amacı vardır ve belirli eylemleri tetikler.

- Trigger Butonu, silah üzerindeki tetik görevi görür. Bu butona basıldığında, MPU6050 sensöründen alınan ivme ve jiroskop verileri, UART ile Windows Form App uygulamasına gönderilir.
- Unloading Butonu, şarjör boşaltma işlemini simüle eder. Basıldığında, belirli bir işlevi tetikler veya bir eylemi başlatır. Örneğin, oyun ortamında cephane değişimi veya karakter eylemi gibi işlemleri kontrol edebilir.
- Loading Butonu, şarjör doldurma işlemini simüle etmek için kullanılır. Butona basıldığında, sistemde belirli bir eylemi tetikler. Genellikle cephane ekleme veya benzer bir işlemi temsil eder.
- Lazer Led,Kullanıcı, lazer LED'i kullanarak hedefin merkezini belirleyebilir. Bu, sistemdeki diğer bileşenlerin doğru hizalanmasını sağlar ve kullanıcı etkileşimini kolaylaştırır. Özellikle silah simülasyonları veya hedef oyunlarında, lazer LED, doğru nişan alma ve hedef belirleme için kritik bir rol oynar.

4.1.5 MSP430G2553 Bağlantı Şeması

MSP430G2553 Bağlantı Şeması, mikrodenetleyicinin çeşitli bileşenlerle bağlantısını detaylandırır. Bu şema, butonların, sensörlerin ve LED'lerin MSP430G2553 ile nasıl etkileşime girdiğini gösterir. Şema, her bir bileşenin MSP430G2553'ün hangi pinlerine bağlandığını ve bu bağlantıların elektronik sistemin genel çalışmasını nasıl etkilediğini açıklar. Şekil 4:5’te MSP430G2553 bağlantı şeması gösterilmiştir.

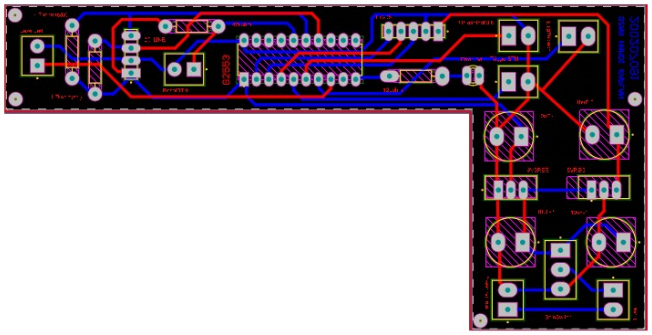


Şekil 4:5 MSP430G2553 Bağlantı Şeması

- RST Pini: Mikrodenetleyicinin reset pinidir ve genellikle 47k Ohm direnç ile 3.3V'a çekilir. Bu direnç, mikrodenetleyicinin stabil çalışmasını sağlar ve yanlışlıkla resetlenmesini önler.
- Reset Butonu: RST pinine bağlıdır ve mikrodenetleyiciyi manuel olarak resetlemek için kullanılır. Kullanıcı, sistemde bir sorun olduğunda veya yeniden başlatma gerektiğinde bu butonu kullanarak mikrodenetleyiciyi sıfırlayabilir.
- Sensörler: MSP430G2553, MPU6050 ve HMC5883L sensörlerle bağlanır. Bu sensörlerin veri hatları (örneğin, I2C için SDA ve SCL) mikrodenetleyicinin uygun pinlerine bağlanır. Pull-up dirençler, SCL ve SDA hatlarını yüksek seviyeye çeker ve bu sayede hatlarda bulunan aktif sürücüler olmadığında (örneğin, bir cihazın veri göndermediği durumda), hatların mantıksal seviyesini korur. 4.7k dirençler sıklıkla kullanılır çünkü I2C standartlarında önerilen bir değerdir ve genellikle güvenilir bir iletişim sağlar. Bu dirençler, özellikle birden fazla cihazın aynı I2C veri hattını paylaştığı çoklu cihaz sistemlerinde önemlidir. Çünkü bir cihaz veri göndermediğinde, diğer cihazlar SCL ve SDA hatlarını pull-up dirençler aracılığıyla yüksek seviyeye çekebilirler, böylece iletişim hattının doğru çalışmasını sağlarlar.
- Butonlar: Butonlar, genellikle mikrodenetleyicinin giriş pinlerine bağlanır. Pull-up veya pull-down dirençleri, butonun açma veya kapama durumunu sabit bir şekilde algılamak için kullanılır, ancak bu dirençler doğrudan şemada gösterilmez.

4.2 PCB Tasarım

Bu bölüm, devre şemasındaki bileşenlerin fiziksel yerleşimini ve bağlantılarını içerir. Tasarım, bileşenlerin boyutlarına, devre şemasındaki bağlantıların fiziksel olarak gerçekleştirilebilirliğine ve PCB'nin kullanılacağı ortama göre optimize edilir. Şekil 4:6'da PCB tasarımı gösterilmiştir.

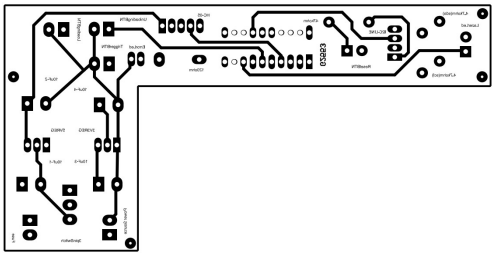


Şekil 4:6 PCB Tasarım

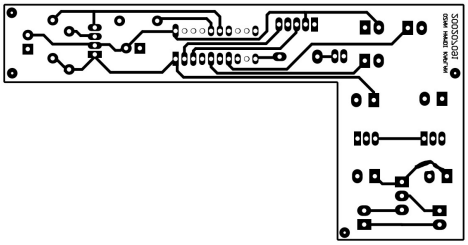
4.3 Baskı Devre Yapımı

Baskı devre üretimi, elektronik cihazların içindeki devrelerin düzenli ve etkili bir şekilde oluşturulmasını sağlayan önemli bir süreçtir. Bu bölümde, baskı devre yapımının adımlarını içerir.

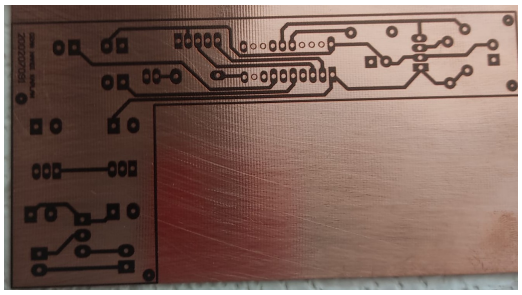
İlk adım, gerçek boyutlarda bir baskı kağıdına çıktı almakla başlar. Bu çıktı, son devrenin tasarımını temsil eder ve üzerinde yer alan izler, bileşenlerin konumları ve bağlantı noktaları gibi detayları içerir. Ardından, bir ütü yardımıyla temizlenmiş bakır plakaya bu baskı yapılır. Bu işlem, baskıyı bakır plakaya aktararak devre elemanlarının yerlerini belirlemeyi sağlar. PCB baskısı için PCB ön ve arka yüz görüntüleri sırasıyla Şekil 4:7 ve Şekil 4:8'de gösterilmiştir. Şekil 4:9'da Bakır plaka üzerine yapılan devre baskısı gösterilmektedir.



Şekil 4:7 PCB Ön Yüz

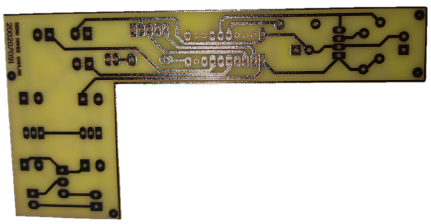


Şekil 4:8 PCB Arka Yüz



Şekil 4:9 PCB Baskı

Daha sonra, doğru oranlarda hazırlanan perhidrol ve tuz ruhu çözeltisi içine bakır plaka atılır. Bu kimyasal karışım, baskıyı oluşturan bakır yolların dışındaki fazla bakırı çözerek istenilen devre desenini ortaya çıkarır. Bu desen Şekil 4:10'da gösterimiştir. Kalan bakır yolların üzerindeki mürekkepler, aseton yardımıyla temizlenir ve böylece net ve temiz bir devre deseni elde edilir.



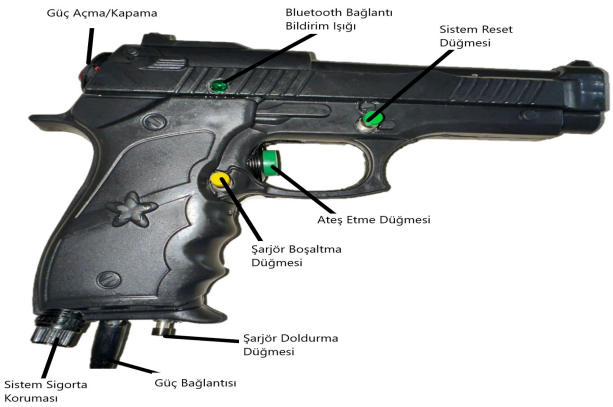
Şekil 4:10 Kimyasal Karışım Sonrası Devre Deseni

Via delikleri, devrenin farklı katmanları arasında iletişim sağlayan deliklerdir ve bunlar 1mm matkap ucu ile açılır. Çift taraflı devrenin tamamlanması için gerekli alanlarda kısa bir bakır tel bağlantısı ile iletim sağlanır. Bu adım, devre elemanları arasındaki elektriksel bağlantıyı güçlendirir ve devrenin sağlam bir şekilde çalışmasını sağlar.

Son olarak, kullanılan bileşenlerin lehimlenmesi yapılır. Bu adım, devreye yerleştirilen parçaların kalıcı ve güvenilir bir şekilde bağlanmasını sağlar. Tüm bileşenlerin yerleştirilmesi ve lehimlenmesinin ardından, devre silahın içine uygun şekilde yerleştirilir ve montaj işlemi tamamlanmış olur. Şekil 4:11'de devrenin yerleştirilmiş ve bitmiş hali gösterilmiştir.

Şekil 4:11 Baskı Devrenin Silahın İçine Alınması

Bu adımların dikkatlice ve doğru bir şekilde uygulanması, işlevsel ve güvenilir bir baskı devrenin oluşturulmasını sağlar. Elektronik projelerde kullanılan baskı devrelerinin bu titiz süreci, istikrarlı ve sağlam çalışma prensibini garanti eder. Silah üzerindeki eklenen parçalar ve isimleri Şekil 4:12'de gösterilmektedir.



Şekil 4:12 Silah Üzerindeki Eklenen Parçalar ve İsimleri

5. YAZILIM

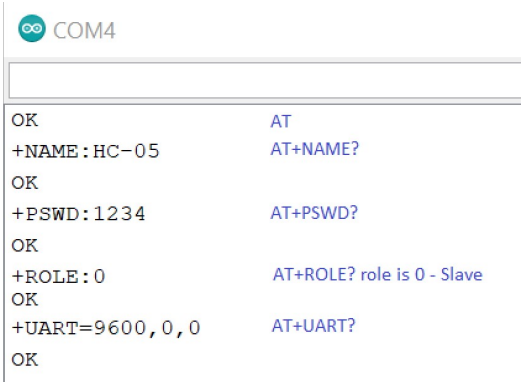
Bu bölüm, projenin yazılım bileşenlerini ve yapılandırmalarını inceleyecektir. Öncelikle, MSP430 mikroişlemcisinin Code Composer Studio'da nasıl programlandığı ve HC-05 Bluetooth modülü ile UART üzerinden bağlantının nasıl sağlandığına bakacağız. Ayrıca, MPU6050 ve HMC5883L sensörlerinden I2C aracılığıyla veri toplama sürecini ve mikroişlemci ile sensör arasındaki etkileşimi ele alacağız. Son olarak, mikroişlemciye bağlı butonlar, LED'ler ve diğer bileşenlerin proje için nasıl yapılandırıldığı ve bu bileşenlerin projenin genel işlevselliğine nasıl katkı sağladığı anlatılacaktır.

Diğer bir başlıkta, Windows Form App'te Bluetooth ile alınan verilerin nasıl işlendiği ve görselleştirildiği incelenecektir. Bu bölümde, mikroişlemciden gelen verilerin bilgisayar ortamında nasıl analiz edildiği ve uygulama arayüzünde nasıl sunulduğu üzerinde durulacaktır.

5.1 Bluetooth Cihazlarının Yapılandırılması

Bu bölümde, HC-05 Bluetooth modülünün yapılandırılması ele alınmaktadır. HC-05 modülü, MSP430 mikroişlemcisi ile kablosuz iletişim sağlamak için kullanılır. Modül, AT komutlarıyla yapılandırılır ve burada belirlenen parametreler arasında modülün adı (name), baud hızı (UART) ve şifre (PSWD) yer alır.

HC-05 Bluetooth modülünü yapılandırmak için USB-TTL dönüştürücüsü ve Arduino IDE'nin seri port ekranı kullanılır. Modülü yapılandırmak için AT komutları ile modülün adı, baud hızı ve şifresi gibi parametreler belirlenir. Bu işlemlerin ardından modül, MSP430 mikroişlemcisi ile Bluetooth iletişimi sağlayacak şekilde hazır hale getirilir. Şekil 5:1’de yapılan işlemler gösterilmiştir.



Şekil 5:1 Bluetooth AT Komutu ile Serial Port Ekranında Yapılandırılması

AT komutları, Bluetooth modüllerinin (örneğin, HC05) yapılandırılması ve kontrol edilmesi için kullanılan standart komutlardır. HC05 modülü, AT komutlarına yanıt vererek yapılandırma işlemlerinin yapılmasını sağlar. Aşağıda, AT komutlarının bazı temel kullanımları açıklanmaktadır:

- AT Komutu: Bluetooth modülünün çalışıp çalışmadığını kontrol etmek için kullanılan temel komuttur. "AT" gönderildiğinde, modül "OK" ile yanıt verirse, modülün düzgün çalıştığını gösterir [2]. Bu komut, Bluetooth modülü ile doğru bir şekilde iletişim kurulduğunu doğrulamak için kullanılır.
- AT+NAME? Komutu: Bluetooth modülünün mevcut adını sorgulamak için kullanılır [2]. Modülün adı, Bluetooth üzerinden başka cihazlar tarafından görülecek olan isimdir. Bu komut kullanılarak, modülün adı öğrenilebilir.
- AT+PSWD? Komutu: Bluetooth modülünün mevcut parolasını sorgulamak için kullanılır [2]. Parola, cihazların birbirine bağlanmasını kontrol eden güvenlik ölçüsüdür. Bu komut, modülün mevcut parolasını öğrenmek için kullanılır.
- AT+ROLE? Komutu: Bluetooth modülünün mevcut rolünü sorgulamak için kullanılır [2]. HC05 modülü, ana cihaz (master) veya bağımlı cihaz (slave) olarak çalışabilir. Bu komut, modülün rolünü belirler.
- AT+UART? Komutu: Bluetooth modülünün mevcut UART ayarlarını sorgulamak için kullanılır [2]. Bu komut, modülün baud hızı, dur bitleri ve parite ayarlarını içerir.

5.2 Mikroişlemci Programlama: Code Composer Studio ile C Kodları

Bu bölümde, MSP430 mikrodenetleyicisini Code Composer Studio (CCS) kullanarak programlamanın temelleri ele alınmaktadır. Mikroişlemci, Bluetooth ve sensör bileşenlerinin yapılandırılması ve etkileşimi C dili kullanılarak gerçekleştirilmektedir. Bu kodlarla, Bluetooth iletişimi, sensör verisi toplama ve butonların kullanımı gibi temel işlevler içerir.

5.2.1 Bluetooth İletişimi için UART Yapılandırması

Bluetooth iletişimi için MSP430 ile HC-05 modülü arasındaki UART protokolü yapılandırılır. UART ayarlarının yapılması için gerekli fonksiyonlar ve işlemler aşağıda belirtilmiştir:

- Setup_UART(): UART yapılandırmasını yapar ve iletişim için gerekli baud hızını ayarlar.
- uart_send_hex(uint8_t data): 1 byte'lık veriyi ASCII formatında gönderir. Bu, Bluetooth modülü aracılığıyla verilerin iletilmesi için kullanılır.
- uart_send_char(unsigned char c): Tek bir karakteri UART üzerinden gönderir.
- UART_Send_End_Of_Data(): Veri iletiminin sonunu belirtmek için satır sonu karakteri gönderir.

5.2.2 MPU-6050 Sensörü Ve HMC5883L için I2C Yapılandırması

MPU6050 ve HMC5883L, I2C protokolünü kullanarak mikroişlemci ile iletişim kurar [31]. Bu bölümde, sensör yapılandırması ve I2C üzerinden veri alışverişi için kullanılan temel fonksiyonlar açıklanmaktadır:

- Setup_TX(unsigned char Dev_ID): I2C üzerinden veri iletimi için yapılandırma yapar.
- Setup_RX(unsigned char Dev_ID): I2C üzerinden veri alma işlemi için yapılandırma yapar.
- Transmit(unsigned char Reg_ADD, unsigned char Reg_DAT, int txByte_count): Belirtilen adres ve veri miktarını I2C üzerinden iletir.

- TransmitOne(unsigned char Reg_ADD): Belirli bir adrese I2C üzerinden tek bir byte gönderir.
- Receive(int rxByte_count): I2C üzerinden belirli sayıda byte veri alır.

5.2.3 Buton ve Diğer Bileşenlerin Pin Yapılandırması

MSP430 mikroişlemcisinin giriş/çıkış (GPIO) pinlerinin doğru şekilde yapılandırılması, çeşitli sensörler, butonlar ve LED'ler için hayati öneme sahiptir. Bu bölümde, GPIO pinlerinin yapılandırılması için kullanılan fonksiyonlar ve ayarlar açıklanmaktadır:

- P1SEL ve P1SEL2: Pinlerin fonksiyonlarını belirler.
- P1DIR: Pinlerin giriş veya çıkış modunda olup olmadığını ayarlar.
- P1REN: Pull-up ve pull-down direnç ayarlarını yapar.
- P1OUT: Pinlerin varsayılan çıkış durumunu belirler.
- P2DIR, P2SEL, P2SEL2: Diğer çıkış pinlerinin ayarlarını yapar.

5.2.4 Fonksiyon İşlemleri

Bu bölümde C kodundaki fonksiyonların temel işlevleri ve işlemleri açıklanmıştır.

- System_Init() ve GPIO_Init(): Bu fonksiyonlar, temel donanım ayarlarını yaparak sistem hazırlığını sağlar.
- CheckBluetoothConnection(): Bluetooth bağlantısının varlığını kontrol eder. Bu fonksiyon, bağlantının varlığını belirleyerek, bağlı ise true değeri döndürür.
- Mpu6050_WakeUp(): MPU6050 sensörünü uyandırır. Sensörün başlangıç durumunu ayarlar ve belirli bir başlangıç konfigürasyonu sağlar.
- HMC5883L_Init(): HMC5883L manyetometresinin çalışma modunu ve konfigürasyonunu ayarlar.
- Mpu6050_States_OK(): MPU6050 sensörünün çalışma durumunu kontrol eder ve bir hata durumunda uygun bir aksiyon alabilir.
- ReadAccelGyro() ve ReadHMC5883L(): İlgili sensörlerden gelen veriyi okur ve uygun işleme için hazırlar.

- AssignRxValuestoArrayMPU() ve AssignRxValuestoArrayHMC(): Okunan sensörlerden gelen verileri uygun bir veri dizisine atar ve işleme hazır hale getirir.
- CheckSum(): Bu fonksiyon, veri bütünlüğünü kontrol etmek için bir kontrol toplamı hesaplar ve veri dizisine ekler.
- Handle_Buttons(): Buton durumlarını kontrol eder ve Bluetooth aracılığıyla ilgili bilgileri gönderir.

5.3 Windows Form App Geliştirme: C# ile Kullanıcı Arayüzü ve Fonksiyon İşlemleri

Bu bölümde, Windows Form App kullanılarak MSP430'dan gelen verilerin C# dilinde nasıl işlendiği incelenir. Windows Form App, Bluetooth üzerinden mikroişlemciden gelen verileri alır ve analiz eder. Bu kısımda, C# kullanılarak Windows Form App'te yapılan veri işleme ve görselleştirme süreçleri açıklanmaktadır.

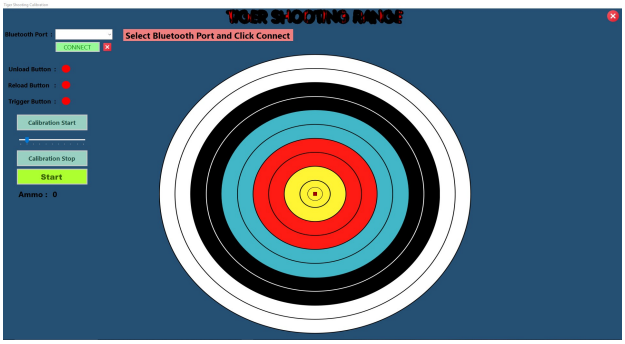
5.3.1 Kullanıcı arayüzü

Bu uygulama, bir atış menzili simülasyonu için kullanıcı arayüzü sağlar. Aşağıda kullanıcı arayüzünün ana bileşenleri ve işlevleri açıklanmıştır:

- Bluetooth Bağlantısı: Kullanıcı, Bluetooth üzerinden bir cihaza bağlanmak için COM portunu seçer ve ardından "Bağlan" butonuna tıklar.
- Kalibrasyon Başlatma: Atış menzilini kalibre etmek için "Kalibrasyon Başlat" butonuna tıklanır. Bu işlem, sensörlerin başlangıç konumlarını ayarlar ve atış noktasını belirler.
- Atış Modu Başlatma: Kalibrasyon tamamlandıktan sonra, "Başlat" butonuna tıklanarak atış modu başlatılır. Bu modda, sensörlerin algıladığı verilere dayanarak atışları simüle edilir.
- Mermi Sayacı: Atış modunda, kullanıcının kaç mermi kaldığını gösterir. Mermi sayısı, her atış yapıldığında azalır ve mermi bittiğinde kullanıcıya uyarı verilir.
- Çıkış: Uygulamadan çıkmak için sağ üst köşedeki "Çıkış" butonuna tıklanır.

- Hata ve Bilgilendirme Penceresi: Uygulama, kullanıcıya çeşitli durumlarda bilgi ve uyarı mesajları gösterir. Örneğin, eksik mermi, yanlış sıra gibi durumlarda uyarı verir.

Bu bileşenlerin bir araya gelmesiyle, kullanıcıya gerçek bir atış menzili deneyimi sunulur. Uygulama, Bluetooth üzerinden cihaza bağlanarak gerçek zamanlı veri alışverişi yapabilir ve kullanıcıya interaktif bir deneyim sağlar. Şekil 5:2'de kullanıcı arayüzü gösterilmiştir.



Şekil 5:2 Kullanıcı Arayüzü

5.3.2 Fonksiyon İşlemleri

Bu bölümde C# kodundaki fonksiyonların temel işlevleri ve işlemleri açıklanmıştır.

- Form1_Load: Form yüklendiğinde çağrılan bu fonksiyon, kullanıcı arayüzü kontrollerinin konumlarını ve boyutlarını ayarlar, kalibrasyon ve başlatma butonlarını devre dışı bırakır ve mevcut seri portları listeleyerek kullanıcıya sunar.
    - Kontrol konumlarını ve boyutlarını ayarlar.
    - Kalibrasyon ve başlatma butonlarını devre dışı bırakır.
    - Mevcut seri portları alır ve bluetoothCMB bileşenine ekler.
- SrlPort_DataReceived: Seri porttan veri alındığında tetiklenen bu olay işleyici, alınan veriyi tamponda biriktirir ve verinin tamamı alındığında işleme başlar.
    - Gelen veriyi buffer adlı StringBuilder'a ekler.

    - Tam veri paketi alındığında (\n karakteri ile sonlandığında), veriyi parçalar ve ilk byte'a göre uygun işlemi gerçekleştirir (DisplayDataPark_event, BulletFree, BulletFull)
- BulletFull: Merminin dolu olduğunu belirten işlemleri gerçekleştiren fonksiyondur.
    - Duruma bağlı olarak kullanıcıya uyarılar gösterir veya mermi sayacını günceller.
    - reloadGreenPB öğesini görünür yapar ve durumu günceller.
- BulletFree: Merminin boş olduğunu belirten işlemleri gerçekleştiren fonksiyondur.
    - Duruma bağlı olarak kullanıcıya uyarılar gösterir veya mermi sayacını sıfırlar.
    - unLoadGreenPB öğesini görünür yapar ve durumu günceller.
- DisplayDataPark_event: Alınan veri paketini işleyen ve form üzerindeki kontrolleri güncelleyen fonksiyondur.
    - Veri paketi doğruluğunu kontrol eder.
    - Sensör verilerini alır ve hesaplamalar yapar.
    - Mermi konumunu hesaplar ve günceller.
    - Mermi sayacını günceller ve gerekli uyarıları gösterir.
- SetControlLocationsAndSizes: Form üzerindeki kontrollerin konumlarını ve boyutlarını ayarlayan fonksiyondur.
    - Formun genişlik ve yüksekliğine göre kontrol konumlarını ve boyutlarını dinamik olarak hesaplar ve ayarlar.
- ConnectBtn_Click: Bağlan butonuna tıklanıldığında seri port bağlantısını gerçekleştiren fonksiyondur.
    - Seçilen seri portu açar ve gerekli ayarları yapar.
    - Bağlantı başarılı olduğunda, ilgili kontrolleri günceller ve veri alımı için olayı abone eder.
- bluetoothCMB_SelectedIndexChanged: Bluetooth combobox öğesinde seçim değiştiğinde bağlan butonunu etkinleştiren fonksiyondur.
    - ConnectBtn butonunu etkinleştirir.
- Form1_SizeChanged: Formun boyutu değiştiğinde kontrol konumlarını ve boyutlarını yeniden ayarlayan fonksiyondur.

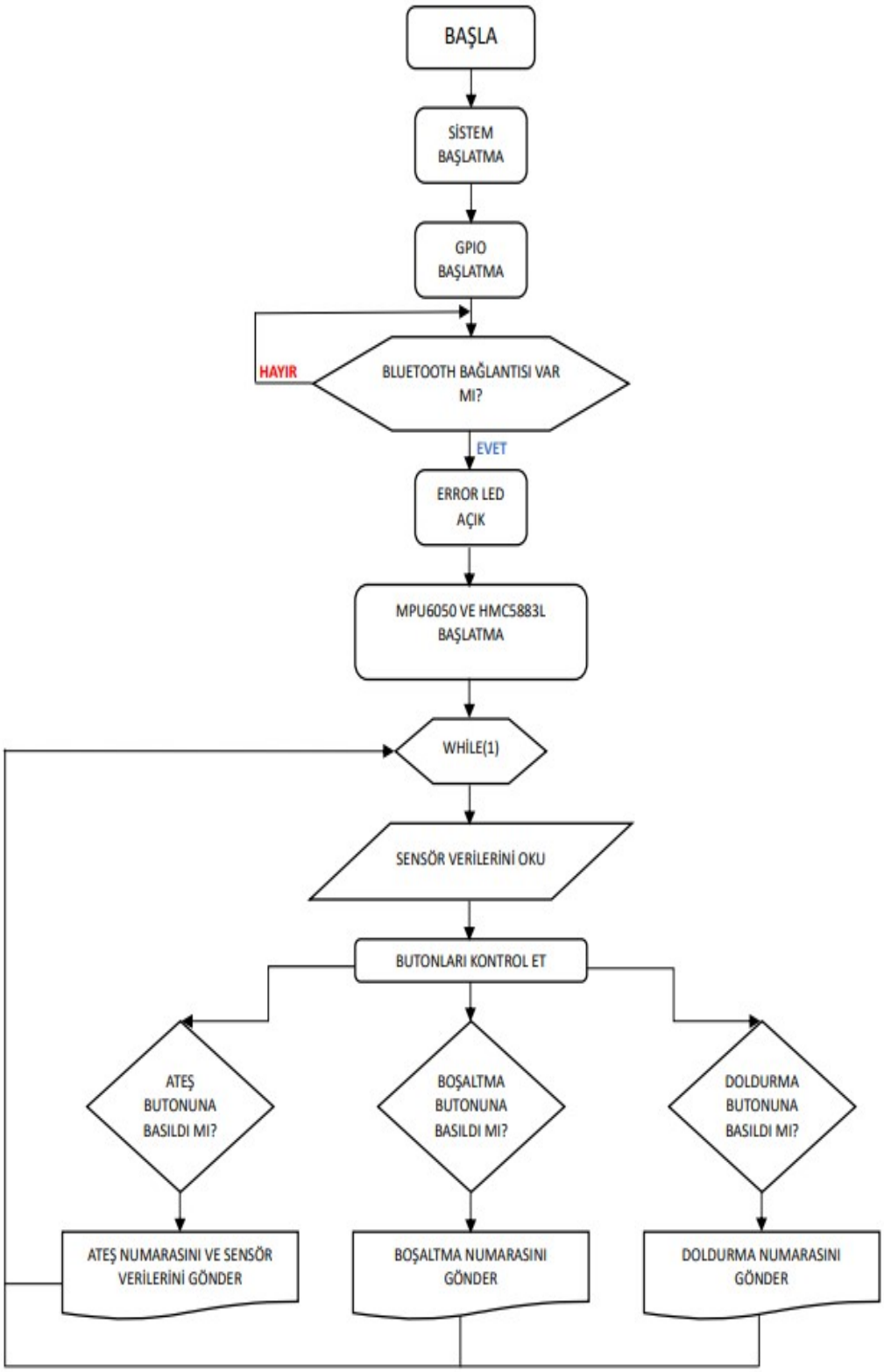
    - SetControlLocationsAndSizes fonksiyonunu çağırarak kontrollerin boyutlarını ve konumlarını yeniden hesaplar.
- calibrateStartBTN_Click: Kalibrasyon başlatma butonuna tıklanıldığında kalibrasyon işlemlerini başlatan fonksiyondur.
    - Görsel öğeleri günceller ve durumu değiştirir.
- calibrateOkBTN_Click: Kalibrasyon onay butonuna tıklanıldığında başlatma butonunu etkinleştiren fonksiyondur.
    - StartBtn butonunu etkinleştirir ve kalibrasyon başlatma butonunu devre dışı bırakır.
- StartBtn_Click: Başlat butonuna tıklanıldığında atış modunu başlatan fonksiyondur.
    - Kalibrasyon onay butonunu devre dışı bırakır ve durumu değiştirir.
- exitPB_Click: Çıkış butonuna tıklanıldığında uygulamayı kapatan fonksiyondur.
    - Uygulamayı kapatır

5.4 Akış Şemaları

Akış şemaları, bir programın veya algoritmanın mantıksal akışını görsel olarak temsil eder ve bu sayede hem geliştiriciler hem de diğer paydaşlar için sürecin anlaşılmasını kolaylaştırır. Bu bölümde, C ve C# programlama dilleri için örnek kodların akış şemalarını inceleyeceğiz.

5.4.1 C kodunun Akış şeması

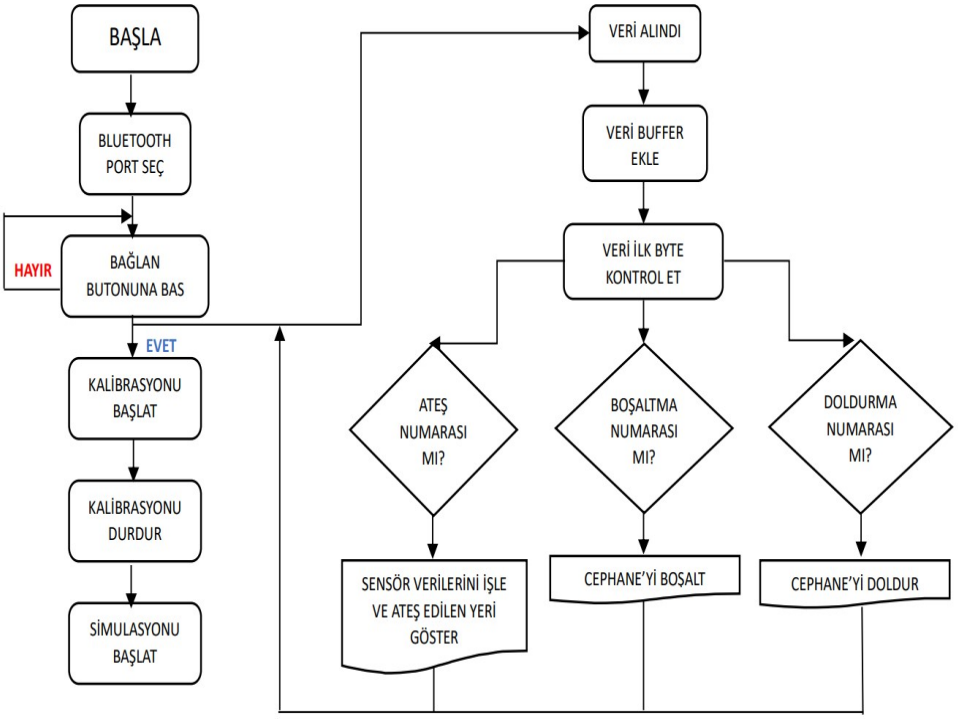
Bu alt bölümde, belirli bir C kod örneğinin akış şemasını oluşturarak, kodun işleyiş mantığını adım adım Şekil 5:3‘de görsel olarak inceleyeceğiz.



Şekil 5:3 C Kodu Akış Şeması

5.4.2 C# kodunun Akış Şeması

Bu bölümde, bir C# programının akış şemasını detaylandırarak, kodun nasıl bir mantık sırasına göre işlediğini ve hangi adımlardan geçtiğini Şekil 5:4‘de görsel olarak inceleyeceğiz.



Şekil 5:4 C# Akış Şeması

6. UYGULAMA

Bu bölümde, MPU6050 ivmeölçer ve jiroskop sensörü ile HMC5883L manyetometre sensörünün nasıl kullanıldığını ve programın nasıl çalıştığını detaylandıracağız. Sensörlerin özelliklerini, başlatma ve veri alma fonksiyonlarını açıklayarak, uygulamanın genel işleyişini adım adım inceleyeceğiz. Bu sensörler, hareket ve manyetik alan ölçümleri yaparak çeşitli uygulamalarda kullanılabilir, bu nedenle doğru yapılandırma ve veri alma süreçleri kritik öneme sahiptir.

6.1 MPU6050'nin Kullanımı,

MPU6050'nin kullanımı, başlatılması ve verilerin okunması gibi temel adımları içerir. Başlatma işlemi genellikle cihazın başlangıç durumuna getirilmesini sağlar, böylece doğru ölçümler alınabilir. Ardından, ivmeölçer ve jiroskop verileri okunarak, cihazın konumunu, hızını ve hareketini belirlemek mümkün olur. Bu başlık altında, MPU6050'nin başlatılması ve ivmeölçer ile jiroskop verilerinin okunması gibi temel işlemler ele alınacaktır. Tablo 1'de, MPU6050'nin önemli registerları ve bu registerların adresleri verilmiştir.

Tablo 1 MPU6050 Registerları [32]

REGİSTER İSMİ	REGİSTER ADRESİ (HEX)	TANIM
MPU6050_ADDR	0X68	MPU6050 Sensörünün I2C Adresi
PWR_MGMT_1	0X6B	Güç Yönetim Registerı
SMPLRT_DIV	0X19	Örnekleme Hızı Bölücü Registerı
ACCEL_CONFIG	0X1C	İvmeölçer Konfigürasyon Registerı
ACCEL_XOUT_H	0X3B	İvmeölçer X Eksen Yüksek Baytı

6.1.1 MPU6050'nin Başlatılması (Mpu6050_WakeUp)

Mpu6050_WakeUp fonksiyonu, MPU6050 sensörünü uyandırmak ve gerekli ayarları yapmak için kullanılır. Bu fonksiyon üç temel registera veri yazar. Şekil 6:1'de bu fonksiyon gösterilmektedir.

```
void Mpu6050_WakeUp()
{

    Setup_TX(MPU6050_ADDR);
    Transmit(PWR_MGMT_1, 0x00 , NUM_BYTES_TX);
    while (UCB0CTL1 & UCTXSTP);

    Setup_TX(MPU6050_ADDR);
    Transmit(SMPLRT_DIV, 0x07,1);
    while (UCB0CTL1 & UCTXSTP);

    Setup_TX(MPU6050_ADDR);
    Transmit(ACCEL_CONFIG, 0x00 , 1);
    while (UCB0CTL1 & UCTXSTP);

    Setup_TX(MPU6050_ADDR);
    Transmit(GYRO_CONFIG, 0x00,1);
    while (UCB0CTL1 & UCTXSTP);

}
```

Şekil 6:1 Mpu6050_WakeUp Fonksiyonu

- PWR_MGMT_1 registerına 0x00 yazmak: Sensörü uyandırır [32]. Bu register başlangıçta 0x40 gibi bir değer alır ve sensörü uyku modunda tutar. 0x00 yazarak sensörü aktif hale getiririz.
- SMPLRT_DIV registerına 0x07 yazmak: Örnekleme hızını ayarlar [32]. 0x07 değeri, 1 kHz örnekleme hızı için kullanılabilir.
- ACCEL_CONFIG registerına 0x00 yazmak: İvmeölçerin tam ölçek aralığını ±2g olarak ayarlar (default). Şekil 6:2 ve Şekil 6:3'te bu ayarın cihazın belirli özellikleri üzerindeki etkisi detaylı bir şekilde incelenmiştir.

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

Şekil 6:2 İvmeölçer Konfigurasyonu Bitleri [32]

AFS_SEL	Full Scale Range
0	± 2*g*
1	± 4*g*
2	± 8*g*
3	± 16*g*

Şekil 6:3 İvmeölçer Konfigurasyonu Range Değerleri [32]

6.1.2 İvmeölçer ve Jiroskop Verilerini Okuma (ReadAccelGyro)

ReadAccelGyro fonksiyonu, MPU6050 sensöründen ivmeölçer ve jiroskop verilerini okumak için kullanılır. Bu fonksiyon, verilerin okunacağı registerı belirtir ve ardından belirli bir miktar veri alır. Şekil 6:4'te bu fonksiyon gösterilmektedir.

```
void ReadAccelGyro()
{

    Setup_TX(MPU6050_ADDR);
    TransmitOne(ACCEL_XOUT_H);
    while (UCB0CTL1 & UCTXSTP);

    Setup_RX(MPU6050_ADDR);
    Receive(NUM_BYTES_RX);
    while (UCB0CTL1 & UCTXSTP);

}
```

Şekil 6:4 ReadAccelGyro Fonksiyonu

- ACCEL_XOUT_H registerından ivmeölçerin X ekseni verisini okumak için bu registera erişilir [32]. Buradan itibaren 6 bayt okunur (X, Y, Z ivmeölçer verileri). Şekil 6:5'te bu register içeriği ve işlevi detaylı bir şekilde gösterilmektedir.

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

Şekil 6:5 İvmeölçer Ölçümleri (X,Y,Z) [32]

6.2 HMC5883L'nin Kullanımı

HMC5883L'nin kullanımı da MPU6050 gibi temel adımları içerir. Başlatma işlemi genellikle cihazın başlangıç durumuna getirilmesini sağlar, böylece doğru ölçümler alınabilir. Ardından, manyetik alan verileri okunarak, cihazın bulunduğu yön

belirlenebilir. Bu başlık altında, HMC5883L'nin başlatılması ve verilerin okunması gibi temel işlemler ele alınacaktır.

Tablo 2'de, HMC5883L'nin önemli registerları ve bu registerların adresleri verilmiştir. Bu registerlar, sensörün yapılandırılması ve ölçümlerin alınması için kullanılır.

Tablo 2 HMC5883L Registerları [4]

REGİSTER İSMİ	REGİSTER ADRESİ (HEX)	TANIM
HMC5883_ADDR	0X1E	HMC5883 Sensörünün I2C Adresi
CONFİG_REG_A	0X00	Konfigürasyon Registerı A
CONFİG_REG_B	0X01	Konfigürasyon Registerı B
MODE_REG	0X02	Mod Registerı
DATA_OUT_X_MSB	0X03	Veri Çıkış X MSB Registerı

6.2.1 HMC5883L'nin Başlatılması (HMC5883L_Init)

HMC5883L_Init fonksiyonu, HMC5883L sensörünü başlatmak ve gerekli ayarları yapmak için kullanılır. Bu fonksiyon üç temel registera veri yazarak sensörü yapılandırır. Şekil 6:6'da fonksiyonun kodu gösterilmiştir.

```
void HMC5883L_Init()
{

    Setup_TX(HMC5883_ADDR);
    Transmit(CONFİG_REG_A, 0x70, NUM_BYTES_TX);
    while (UCB0CTL1 & UCTXSTP);

    Setup_TX(HMC5883_ADDR);
    Transmit(CONFİG_REG_B, 0xA0, NUM_BYTES_TX);
    while (UCB0CTL1 & UCTXSTP);

    Setup_TX(HMC5883_ADDR);
    Transmit(MODE_REG, 0x00, NUM_BYTES_TX);
    while (UCB0CTL1 & UCTXSTP);

}
```

Şekil 6:6 HMC5883L_Init Fonksiyonu

- CONFIG_REG_A registerına 0x70 yazmak: Sensörü çalışma moduna geçirir ve veri çıkış hızını ayarlar [4].

- CONFIG_REG_B registerına 0xA0 yazmak: Ölçüm aralığını ayarlar ve diğer bazı konfigürasyonları yapar [4].
- MODE_REG registerına 0x00 yazmak: Sensörün çalışma modunu belirler. Burada, sürekli ölçüm modunu seçiyoruz [4].

6.2.2 HMC5883L Verilerinin Okunması (ReadHMC5883L)

ReadHMC5883L fonksiyonu, HMC5883L sensöründen manyetik alan verilerini okumak için kullanılır. Bu fonksiyon, verilerin okunacağı registerı belirtir ve ardından belirli bir miktar veri alır. Şekil 6:7'de fonksiyonun kodu gösterilmiştir.

```
void ReadHMC5883L() {

    Setup_TX(HMC5883_ADDR);

    while (UCB0CTL1 & UCTXSTP);
    TransmitOne(DATA_OUT_X_MSB);
    while (UCB0CTL1 & UCTXSTP);

    Setup_RX(HMC5883_ADDR);
    Receive(NUM_BYTES_RX);
    while (UCB0CTL1 & UCTXSTP);

}
```

Şekil 6:7 ReadHMC5883L Fonksiyonu

- DATA_OUT_X_MSB X ekseninin en yüksek byte'ını okumak için bu registera erişilir [16]. Buradan itibaren 6 bayt okunur (X, Y, Z manyetometre verileri). Şekil 6:8'de bu register içeriği ve işlevi detaylı bir şekilde gösterilmektedir.

03	Data Output X MSB Register	Read
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read
08	Data Output Y LSB Register	Read

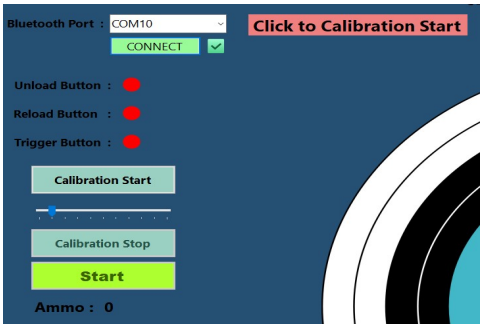
Şekil 6:8 Manyetometre Ölçümleri (X,Y,Z) [4]

6.3 Programın Çalışması

Program, bir mikrodenetleyici (MSP430) ve sensörlerin yanı sıra bir bilgisayar arayüzü üzerinden kontrol edilebilen bir uygulama içermektedir. Mikrodenetleyici, MPU6050 ve HMC5883L gibi sensörler aracılığıyla hareket ve manyetik alan verilerini okur. Bu veriler, atış yapılacak hedefin doğru hizalanması ve atışın hassasiyetinin belirlenmesi için kullanılır. Bluetooth bağlantısı aracılığıyla bilgisayar arayüzü, kullanıcıya atış poligonunu kontrol etme ve izleme imkanı sağlar. Kullanıcı, atışı başlatmak ve duraklatmak için butonları kullanabilir, ayrıca atış yapmadan önce poligonu kalibre etme işlemlerini gerçekleştirebilir. Sensör verileri, kullanıcı arayüzünde gerçek zamanlı olarak görselleştirilir ve atışın doğruluğunu ve hassasiyetini kontrol etmek için kullanılır. Programın çalışması şu adımlarla gerçekleşir: Kullanıcı, Bluetooth bağlantısı üzerinden bilgisayar arayüzüne erişir ve atış poligonunu kontrol etmek için gerekli komutları gönderir. Mikrodenetleyici, sensörler aracılığıyla hedefin konumunu ve atışın yapılacağı açıyı belirler. Kullanıcı, bilgisayar arayüzü üzerinden atış yapar ve sensör verileri doğrultusunda atışın başarısını izler. Bu süreç, kullanıcının atış becerisini geliştirmesi ve poligonun doğruluğunu test etmesi için interaktif bir ortam sağlar.

6.3.1 Silah Bağlantısı ve Hazırlığı

Silahın kullanıma hazır hale getirilmesi, bir dizi adımı içerir. İlk olarak, Bluetooth aracılığıyla silah bağlantısı sağlanır ve bağlantı başarıyla gerçekleştirildiğinde, bildirim ışığı aktive edilir. Bu adımlar, Şekil 6:9 ve Şekil 6:10'da görsel olarak gösterilmiştir.

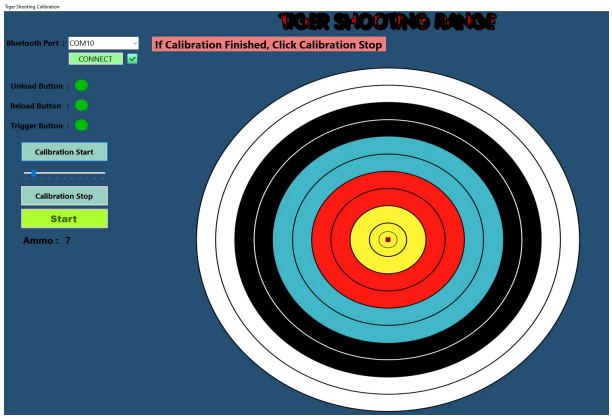


Şekil 6:9 Kullanıcı Arayüzü Bluetooth Bağlantısı Yapıldı

Şekil 6:10 Silah Bluetooth Bağlantısı Yapıldı

Kalibrasyon işlemi, silahın doğruluğunu ve hassasiyetini sağlamak için kritik bir adımdır. Kalibrasyon başlatma düğmesine tıklanarak bu süreç başlatılır. Ardından, şarjör boşaltma düğmesine basılarak silahta bulunan mermiler boşaltılır ve şarjör doldurma düğmesine basılarak 7 mermi yüklenir. Son olarak, ateş düğmesine basılarak kalibrasyon hassasiyeti ayarlanır.

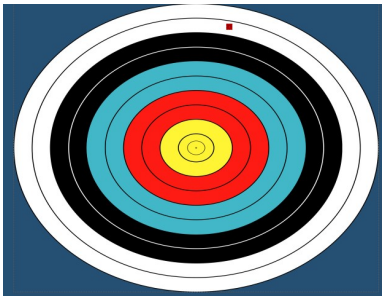
Tüm bu kalibrasyon adımlarının tamamlanmasının ardından, silah simülasyon için hazır hale gelir. Bu durum, Şekil 6:11'de kalibrasyon adımlarının tamamlandığı son durum gösterilmektedir.



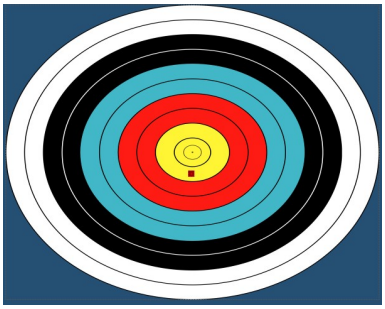
Şekil 6:11 Kalibrasyon Tamamlandı

6.3.2 Simülasyon ve Sonuçlar

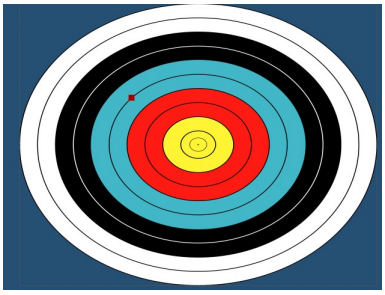
Kalibrasyon tamamlandıktan sonra simülasyon başlatıldı. Ateş edilen yerler incelenerek hedefte vurulan yerler belirlendi. Şekil 6:12, Şekil 6:13 ve Şekil 6:14 ‘te bu sonuçlar görselleştirilmiştir.



Şekil 6:12 Hedefte Vurulan Yer (1)

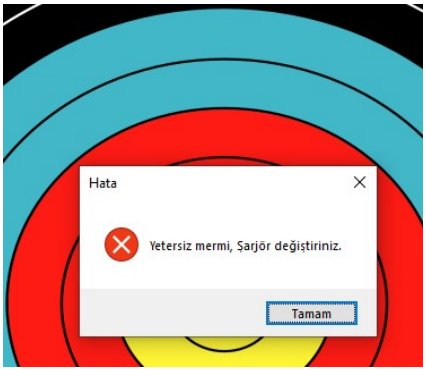


Şekil 6:13 Hedefte Vurulan Yer (2)



Şekil 6:14 Hedefte Vurulan Yer (3)

Mermi miktarının tükenmesi durumunda uyarı sistemi devreye girdi. Şekil 6:15'te uyarı detayları görülmektedir



Şekil 6:15 Mermi Uyarı Sistemi Devrede

7. SONUÇLAR VE ÖNERİLER

Sonuçlar, projemizin hedefleri doğrultusunda gerçekleştirdiğimiz çalışmalar ve testler sonucunda elde edilen bulgular aşağıda detaylandırılmıştır:

- Şematik Tasarım Testleri: Bölüm 4.1'de sunulan şematik tasarımda belirtilen elektriksel değerler, multimetre ile test edilerek doğrulanmıştır. Ölçülen değerlerin beklentilere uygun olduğu gözlemlenmiştir.
- Breadboard Üzerindeki Testler: Yazılım kodlarının doğruluğunu teyit etmek amacıyla breadboard üzerinde devre kurulmuş ve testler gerçekleştirilmiştir. Bu testler sonucunda, sistemin beklenen şekilde çalıştığı görülmüştür.
- MSP430G2553 Testleri: Bölüm 4.3'te açıklanan devre tasarımı aşamalarını tamamladıktan sonra, MSP430G2553 mikrodenetleyicisi devreye entegre edilmiştir. Uygulama üzerinden alınan değerler kontrol edilerek, doğru sonuçlar elde edilmiştir.
- Mekan Değişikliği Testleri: Sistemin farklı mekanlarda çalışabilirliğini test etmek amacıyla çeşitli mekân değişiklikleri yapılmıştır. Bu testler sonucunda, sistemin her mekânda başarılı bir şekilde çalıştığı görülmüştür.
- Hata Testleri: Bilinçli olarak oluşturulan hata senaryoları ile sistemin hata tespit ve tepki verme yetenekleri test edilmiştir. Hata durumunda, sistemin LED uyarı mekanizması başarılı bir şekilde çalışmıştır.
- Hassasiyet Testleri: Sensör hassasiyet değerleri incelenmiş ve gerekli ayarlamalar yapıldığında istenilen doğrulukta sonuçlar elde edilmiştir.

Öneriler, projede elde edilen deneyimler ve sonuçlar doğrultusunda, gelecekte yapılabilecek geliştirmeler ve iyileştirmeler aşağıda sunulmuştur:

- 3 Boyutlu Tasarım: Donanım sisteminin daha düzenli ve derli toplu olması için üç boyutlu tasarım teknikleri kullanılarak muhafaza edilebilir.
- Mekanik Tasarım: Daha gerçekçi bir hissiyat sağlamak amacıyla mekanik tasarım unsurları eklenebilir. Bu, kullanıcı deneyimini önemli ölçüde iyileştirecektir.
- Çevresel Algılamalar: Ateş edildikten sonraki vurulan yeri daha doğru hesaplamak için uzaklık ve hava koşulları gibi çevresel faktörler algılanarak sisteme entegre edilebilir.

- Görüntü İşleme: Hedefte vurulan yerin daha net tespiti için görüntü işleme teknikleri kullanılabilir. Bu sayede, sistemin doğruluğu ve güvenilirliği artırılabilir.
- Kablosuz Güç Kaynağı: Sistemin taşınabilirliğini ve kullanım kolaylığını artırmak amacıyla, güç kaynağı olarak batarya kullanılarak tamamen kablosuz bir sistem kurulabilir.

Bu sonuçlar ve öneriler, projenin mevcut durumunu değerlendirerek, gelecekte yapılabilecek iyileştirmeler ve genişletmeler için bir yol haritası sunmaktadır. Elde edilen veriler ve deneyimler, mikrodenetleyici tabanlı sistemlerin geliştirilmesi konusunda önemli bir referans kaynağı oluşturacaktır.

KAYNAKÇA

[1] Texas Instruments, 2015. MSP-EXP430G2 LaunchPad Evaluation Kit User's Guide (Rev. F) (Nisan 2024 tarihinde ziyaret edilmiştir.)

[2] https://www.alldatasheet.com/datasheet-pdf/pdf/3042/MOTOROLA/HC05.html (Nisan 2024 tarihinde ziyaret edilmiştir.)

[3] https://www.alldatasheet.com/datasheet-pdf/pdf/517744/ETC1/MPU-6050.html (Nisan 2024 tarihinde ziyaret edilmiştir.)

[4] https://www.alldatasheet.com/datasheet-pdf/pdf/428791/HONEYWELL/HMC5883L-TR.html (Nisan 2024 tarihinde ziyaret edilmiştir.)

[5] https://tr.wikipedia.org/wiki/G%C3%B6m%C3%BCl%C3%BC_sistem (Nisan 2024 tarihinde ziyaret edilmiştir.)

[6] https://maker.robotistan.com/mikrodenetleyici-mikroislemci/ (Nisan 2024 tarihinde ziyaret edilmiştir.)

[7] SIG, [Alıntı Tarihi: 3 Mayıs 2015], http://www.bluetooth.com (Nisan 2024 tarihinde ziyaret edilmiştir.)

[8] Haartsen, J. C., and Mattisson, S., (2000). Bluetooth-a new low-power radio interface providing short-range connectivity. Proceedings of the IEEE, 88(10), 1651- 1661. (Nisan 2024 tarihinde ziyaret edilmiştir.)

[9] Bisdikian, C., (2001). An overview of the Bluetooth wireless technology. IEEE Commun Mag, 39(12), 86-94. (Nisan 2024 tarihinde ziyaret edilmiştir.)

[10] https://humansofglobe.com/iot-smart-city-homes/ (Nisan 2024 tarihinde ziyaret edilmiştir.)

[11] https://medium.com/@cenkysn/i2c-haberle%C5%9Fme-protokol%C3%BC-a309e6c17516 (Nisan 2024 tarihinde ziyaret edilmiştir.)

[12] https://www.robocombo.com/blog/icerik/i2c-haberlesme-protokolu-nedir-arduino-ile-nasil-calisir (Nisan 2024 tarihinde ziyaret edilmiştir.)

[13] https://devreyakan.com/seri-haberlesme-protokolleri/ (Nisan 2024 tarihinde ziyaret edilmiştir.)

[14] https://www.hakankaya.kim/blog/uart-temelleri-uart-nedir-calisma-yapisi-ve-seri-haberlesme/ (Nisan 2024 tarihinde ziyaret edilmiştir.)

[15] https://medium.com/@nubika/interrupt-kesme-nedir-20847341c224 (Nisan 2024 tarihinde ziyaret edilmiştir.)
[16] https://filderbaer.wordpress.com/ (Nisan 2024 tarihinde ziyaret edilmiştir.)
[17] https://dergipark.org.tr/en/download/article-file/2033383 (Nisan 2024 tarihinde ziyaret edilmiştir.)
[18] https://rgb-elektronik.com/manyetometre-magnetometer/ (Nisan 2024 tarihinde ziyaret edilmiştir.)
[19] https://www.robolinkmarket.com/msp-exp430g2et-launchpad-gelistirme-kiti (Nisan 2024 tarihinde ziyaret edilmiştir.)
[20] Texas Instruments, 2013. MSP430G2x53-mixed signal microcontroller (Rev.J) (Nisan 2024 tarihinde ziyaret edilmiştir.)
[21] https://gorsuelektronik.com/hc05-arduino-bluetooth-modul (Nisan 2024 tarihinde ziyaret edilmiştir.)
[22] https://www.robotistan.com/mpu6050-6-eksen-ivme-ve-gyro-sensoru-6-dof-3-axis-accelerometer-and-gyros (Nisan 2024 tarihinde ziyaret edilmiştir.)
[23] https://www.hepsiburada.com/alkatronik-mpu6050-6-eksen-imu-ivme-olcer-sensor-modulu-gy-521-mpu-6050-gyro-pm-HBC00000FDMBZ (Nisan 2024 tarihinde ziyaret edilmiştir.)
[24] https://reefwing.medium.com/connecting-the-duinotech-3-axis-compass-to-an-arduino-b13c28d7d936 (Nisan 2024 tarihinde ziyaret edilmiştir.)
[25] Texas Instruments, 2015. Code Composer Studio™ v6.1 for MSP430™ User's Guide (Rev. AH) (Nisan 2024 tarihinde ziyaret edilmiştir.)
[26] Texas Instruments, 2014. MSP430 Optimizing C/C++ Compiler v 4.4 User's Guide (Rev. J) (Nisan 2024 tarihinde ziyaret edilmiştir.)
[27] https://tr.wikipedia.org/wiki/Microsoft_Visual_Studio (Nisan 2024 tarihinde ziyaret edilmiştir.)
[28] https://learn.microsoft.com/tr-tr/dotnet/visual-basic/developing-apps/windows-forms/ (Nisan 2024 tarihinde ziyaret edilmiştir.)
[29] https://medium.com/@berkekurnaz/c-ile-nesne-y%C3%B6nelimli-programlama-26997a747d0c (Nisan 2024 tarihinde ziyaret edilmiştir.)
[30] https://devreyakan.com/altium-designerda-sematik-devre-tasarimi/ (Nisan 2024 tarihinde ziyaret edilmiştir.)

[31] https://github.com/mangpo/MSP430-ADXL345/blob/master/msp430f5529/accelerometer.c (Nisan 2024 tarihinde ziyaret edilmiştir.)

[32] https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf (Nisan 2024 tarihinde ziyaret edilmiştir.)

[33] https://www.design-reuse.com/articles/54776/i2c-interface-timing-specifications-and-constraints.html (Nisan 2024 tarihinde ziyaret edilmiştir.)

[34] https://www.megaplc.com.tr/uart-usart-haberlesme-protokolu-nedir/ (Nisan 2024 tarihinde ziyaret edilmiştir.)

ÖZGEÇMİŞ

2000 yılında Ankara'da doğdu. İlk, orta ve lise öğrenimini Ankara'da tamamladı. 2019 yılında Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü Lisans öğrenimine başladı. 2024 yılında mezun olma durumundadır.