

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AKIŞ ESNASINDA SIVILARDA TEMASSIZ ÖLÇÜM

MÜCAHİT ERDEM
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2023

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**AKIŞ ESNASINDA SIVILARDA TEMASSIZ
ÖLÇÜM**

MÜCAHİT ERDEM
YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
PROF. DR. FUAD ALIEW

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**NON-CONTACT MEASUREMENT OF
LIQUIDS DURING FLOW**

MÜCAHİT ERDEM

**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE**

DEPARTMENT OF ELECTRONICS ENGINEERING

THESIS SUPERVISOR

PROF. DR. FUAD ALIEW

GEBZE

2023



YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 10/07/2023 tarih ve 2023/08 sayılı kararıyla oluşturulan jüri tarafından 19/07/2023 tarihinde tez savunma sınavı yapılan Mücahit Erdem'in tez çalışması Elektronik Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : PROF. DR. FUAD ALIEW

ÜYE

: DR. ÖĞR. ÜYESİ İHSAN ÇİÇEK

ÜYE

: DR. ÖĞR. ÜYESİ BÜLENT İRFANOĞLU

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Sıvı miktarının ölçülmesi ve takibi genellikle gıda endüstrisinde çok önemlidir. Çünkü sıvıların toplanması, ölçülmesi ve paketlenmesi esnasında hijyen konusunda çok hassas olunması gerekmektedir. Su üretim tesisleri ve süt sağım çiftlikleri bu konuda en dikkat edilmesi gereken üretim alanlarıdır. Bu üretim tesisleri direkt olarak son kullanıcıya hitap etmektedir. Son kullanıcıya hitap eden endüstrilerde en çok dikkat edilen konuların başında ise hijyen gelir. Bu yüzden sistemde kullanılan ölçü aletlerinin de mümkün olduğu kadar sade ve sıvıya temasının az olması gerekmektedir.

Bu çalışmada, her türlü sıvının içerisindeki yabancı madde tespiti, sıvının miktarı, iletkenliği ve sıcaklığının ölçümünü yapmak mümkündür. Boru içerisinden geçen sıvı miktarı mili litre cinsinden hesaplanmaktadır. Bu devre, kullanıcı tarafından kalibre edilecek ve ölçülen miktarlar ekrandan okunacak şekilde bir kutu içerisine gömülmüş şekilde hazırlanmıştır. Gerçek ortam testlerini karşılaştırabilmek ve projenin başarısını test edebilmek adına süt çiftliklerinin zorlu koşullarında cihazımızın testlerini gerçekleştirdik. Bu testler gerçekleştirilirken tam otomasyonlu bir sağım odası kurularak, sistemin ölçüm testleri sağım odasında ve gerçek sağım hayvanlarında yapılmıştır. Bu testler sırasında sıvının miktarı, sıcaklığı, iletkenliği ve yabancı madde tespitleri üzerine testler gerçekleştirilmiştir. Farklı ortamlarda gerçekleştirilen bu testler neticesinde sıcaklık, iletkenlik, yabancı madde tespiti ve sıvı miktarı ölçümünde yüksek doğruluk sağlanmıştır.

Anahtar Kelimeler: Sıvı Ölçümü, Sıvılarda Yabancı Madde Tespiti, Temassız Ölçüm Sistemleri, Sıvı Takibi.

SUMMARY

Measuring and monitoring the amount of liquid is often very important in the food industry. Because it is necessary to be very sensitive about hygiene during the collection, measurement and packaging of liquids. Water production facilities and milking farms are the production areas that need the most attention in this regard. These production facilities directly address the end user. Hygiene is one of the most important issues in end-user industries. Therefore, the measuring instruments used in the system should be as plain as possible and have less contact with the liquid.

In this study, it is possible to detect foreign matter in all kinds of liquids, and to measure the amount, conductivity and temperature of the liquid. The amount of liquid passing through the pipe is calculated in milliliters. This circuit is prepared to be calibrated by the user and embedded in a case so that the measured quantities can be read from the screen. In order to compare the real environment tests and test the success of the project, we carried out the tests of our device in the harsh conditions of dairy farms. While performing these tests, a fully automated milking parlor was established and the measurement tests of the system were carried out in the milking parlor and real milking animals. During these tests, tests were carried out on the amount, temperature, conductivity and foreign matter detection of the liquid. As a result of these tests carried out in different environments, high accuracy was achieved in the measurement of temperature, conductivity, foreign matter detection and liquid amount.

Key Words: Liquid Measurement, Foreign Matter Detection In Liquids, Non-contact Measurement Systems, Liquid Trekkinge.

TEŞEKKÜR

Yüksek lisans eğitimimde ve akademik hayatımda desteğini ve yardımlarını hiçbir zaman esirgemeyip, çalışma emeğini esirgemedi bilgisi ile yön göstererek bu çalışmanın oluşmasının yolunu açan danışmanım Prof. Dr. Fuad ALIEW'e en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iv
SUMMARY	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
SİMGELER VE KISALTMALAR DİZİNİ	xii
ŞEKİLLER DİZİNİ	xiii
TABLolar DİZİNİ	xv
1. GİRİŞ	1
1.1. Akışkanlar ve Genel Özellikleri	1
1.2. Akışkan Ölçüm Yöntemleri	1
1.2.1. Hacimsel Akış Ölçerler	2
1.2.1.1. Helisel Vidalı Akış Ölçer	3
1.2.1.2. Döner Lop Akış Ölçer	3
1.2.1.3. Türbin Akış Ölçer	4
1.2.1.4. Çark Sistemli Akış Ölçer	5
1.2.1.5. Diskli Sayaçlar	6
1.2.2. Kütlesel Akış Ölçer	7
1.2.2.1. Kuvvetler Dengesi ile Kütlesel Akış Ölçer	7
1.2.2.2. Değişken Alanlı Akış Ölçer	8
1.2.2.3. Pitot Boru ile Hız Ölçümü	9
1.2.2.4. Anemometre (Gaz Ölçer)	9
1.2.2.5. Ventürimetre ile Hız Ölçer	10
1.2.2.6. Orifis ile Akış Ölçer	11

1.2.2.7. Nozul Akış Ölçer	11
1.2.3. Elektronik Akış Ölçerler	12
1.2.3.1. Ultrasonik Akış Ölçer	12
1.2.3.2. Elektro Manyetik Akış Ölçerler	13
2. SIVI MİKTARININ ÖLÇÜLMESİ	15
2.1. Sıvı Ölçümünün Önemi	15
2.2. Sıvı Ölçüm Yöntemleri	16
2.2.1. İki Fazlı Akış Ölçümü	16
2.2.1.1. Ayrıştırıcı Tabanlı Ölçüm	16
2.2.1.2. Karışık Akış Ölçümü:	16
2.2.2. Çift Fazlı Pulsatil Akış Ölçümü	17
2.2.2.1. Lazer Doppler ile Akış Ölçümü	17
2.2.2.2. Parçacık Görüntülemeli Hız Ölçüm Yöntemi	18
3. SIVI AKIŞININ IŞIK ŞİDDETİ İLE ÖLÇÜLMESİ	20
3.1. Optik Yöntem	20
3.1.1. Sıvı Miktar Ölçüm Yönteminin Tasarımı	20
4. ELEKTRONİK DEVRE TASARIMI VE ÖZELLİKLERİ	22
4.1. Tasarım Özellikleri	22
4.1.1. Mikro Denetleyici	23
4.1.2. Sıcaklık Sensörü	24
4.1.3. Haberleşme Modülü	25
4.2. Elektronik Devre Tasarımı	27
5. YAZILIM ALGORİTMASI	32
5.1. Sıvı Varlığı Fonksiyonu	33
5.2. Sıvı Akışı Hız Fonksiyonu	33
5.3. Sıvı Miktarı Hesaplaması	34
5.3.1. ADC Ortalama Hesaplaması	34

5.4. Buton Fonksiyonu	36
5.5. Sıcaklık Ölçüm Fonksiyonu	36
5.6. Yabancı Madde Tespit Fonksiyonu	36
5.6.1. Sensör Okuma Fonksiyonu	36
5.7. Display Ekran Fonksiyonu	37
5.7.1. “Seg Value Calculate” Fonksiyonu	37
5.7.2. “GosterOutput” Fonksiyonu	38
5.7.3. “Seg_Goster” Fonksiyonu	38
5.8. Bluetooth ve Wifi Haberleşme Fonksiyonu	38
5.8.1. WIFI_AT SEND Fonksiyonu	38
5.8.2. “WIFI AT RESPONSE” Fonksiyonu	39
5.8.3. Bluetooth Haberleşme Fonksiyonu	39
5.8.4. BLE AT RESPONSE Fonksiyonu	39
5.8.5. BLE Transmit Fonksiyonu	39
5.8.6. WIFI Transmit Fonksiyonu.	40
5.8.7. Callback Fonksiyonu	40
5.9. İletkenlik	40
6. KATI MODEL TASARIMI	41
7. TEST VE SONUÇLAR	44
7.1. Sağım Simülasyonu Test Düzeneği	44
7.2. Sağım Simülasyonu Test Çalışmaları	45
7.3. Sağım hane Testleri	46
7.3.1. Test Ortamı	46
7.3.2. Sağım Hane Testleri ve Sonuçları	48
7.4. Sonuç ve Öneriler	51
KAYNAKLAR	54

ÖZGEÇMİŞ	58
EKLER	59

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler ve : Açıklamalar

Kısaltmalar

ADC	: Analog Dijital Çevirici
EEPROM	: Elektronik Silinebilir Programlanabilir Salt Okunur Bellek
FRAM	: Üzerine Yazılabilir Kalıcı Bellek
I/O	: Giriş / Çıkış
ID	: Kimlik Bilgisi
IP	: İnternet Protokol
LED	: Işık Yayan Diyot
mm	: Milimetre
PWM	: Darbe Genişliği Modülasyonu
RAM	: Rastgele Erişim Belleği
sn	: Saniye
GTÜ	: Gebze Teknik Üniversitesi
RF	: Radyo Frekansı

ŞEKİLLER DİZİNİ

<u>Sekil No:</u>	<u>Sayfa</u>
1.1: Helisel vidalı akış ölçer.	3
1.2: Döner lop akış ölçer.	4
1.3: Türbinmetre.	4
1.4: Türbinmetre devir ölçümü.	5
1.5: Çark sistemli akış ölçer.	6
1.6: Diskli sayaçlar.	6
1.7: Kütleli akış ölçer.	7
1.8: Değişken alanlı akış ölçer.	8
1.9: Pitot boru.	9
1.10: Anemometre.	10
1.11: Ventürimetre.	10
1.12: Orifis levha akış ölçümü.	11
1.13: Nozul akış ölçümü.	11
1.14: Ultrasonik akış ölçer.	13
1.15: Manyetik alan akım ölçümü.	13
2.1: Ultrasonik yöntemle akışkan ölçümü.	18
2.2: Parçacık görüntülemeli ölçüm.	19
4.1: Elektronik tasarım blok diyagramı.	22
4.2: Üst kartın ön yüzü.	27
4.3: Üst kartın arka yüzü.	27
4.4: Alt kartın ön yüzü.	28
4.5: Alt kartın arka yüzü.	28
4.6: Infrared led ve alıcıların konumu.	28
4.7: Seven segment display.	29
4.8: Arayüz programı bağlantı sekmesi.	30
4.9: Sıvı ölçüm takip ekranı.	30
4.10: Geçmişe yönelik ölçüm görüntüleme ve analiz ekranı.	31
5.1: Yazılım algoritması blok diyagramı.	32

5.2: Doluluk oranı adc değerleri grafiği.	35
6.1: Sıvı ölçer tasarımı.	42
6.2: Sıvı ölçer cihazın bir kesiti.	42
7.1: Sıvı ölçer cihazın ilk prototipi.	44
7.2: Sıvı ölçer cihaz masaüstü test düzeneği.	44
7.3: Test düzeneğine prototip montajı.	45
7.4: Sağmal inek simülasyonu	45
7.5: Güğüme silikon meme montajı.	45
7.6: Sıvı ölçerlerin sağım hanede montajı.	46
7.7: Sağım hane test düzeneği	47
7.8: Cihaz montajı ve tartı düzeneği.	47
7.9: Birinci test verileri grafiği.	48
7.10: Birinci test verileri.	48
7.11: İkinci test verileri grafiği.	49
7.12: İkinci test verileri.	49
7.13: İkinci test verilerinin arayüz ile görüntülenmesi.	50

TABLolar DİZİNİ

<u>Tablo No:</u>	<u>Sayfa</u>
4.1: Mikroişlemci özellikleri.	23
4.2: Devam.	24
4.3: Sıcaklık sensörü özellikleri.	24
4.4: Haberleşme modülü özellikleri.	25
5.1: Doluluk oranı adc değerleri.	35
7.1: Sıvı ölçer test verileri.	51

1. GİRİŞ

1.1. Akışkanlar ve Genel Özellikleri

Akış, bir akışkanın bir hat boyunca sürekli hareketidir. Canlılarda solunum sistemi, suyun dereden akışı, petrolün bir boru hattından akması, damarlarımızdaki kan akışı ve sütün boru içerisinde taşınması akış sistemlerine birer örnektir [1]. Akış sistemlerinin araştırılması ve incelenmesi eski Yunan bilimci Archimedes'e kadar dayanmaktadır. Tarih boyunca akışkanların özellikleri ve ölçümü hakkında birçok çalışma mevcuttur [5].

Akış kavramı akışkanlar mekaniği açısından düşünüldüğünde bütün maddeler katı veya akışkan madde olarak ikiye ayrılabilir [6]. Akışkanların en temel özelliklerinden biri viskoziteleridir. Viskozite, bir akışkanın akmaya karşı gösterdiği dirençtir. Bütün akışkanların belirli bir viskozitesi vardır. Yüksek viskoziteye sahip akışkanlara (örneğin bal) kalın akışkanlar, düşük viskoziteye sahip (örneğin su) ince akışkanlar denir [2]. Sıvı akışkanlar için bilinmesi gereken ikinci önemli konu ise yüzey gerilimidir. Sıvılar gergin bir yüzey üzerinde kalıyormuş gibi bir davranış sergiler. Bunun nedeni sıvı moleküllerin birbirini çekme isteğinden kaynaklanmaktadır. Sıvı merkezinde bulunan moleküller, diğer yönlerde eşit olarak çekilirken, yüzeyde bulunan diğer moleküller sadece merkeze doğru çekilirler. Çünkü diğer yöndeki molekülleri merkeze doğru çekecek bir kuvvet yoktur. Yüzeyde bulunan moleküllere uygulanan bu kuvvet sebebiyle bir gerilim oluşur ve oluşan bu gerilime yüzey gerilimi denir [2], [4].

1.2. Akışkan Ölçüm Yöntemleri

Akış hızı ve debisinin ölçülmesi, bilimsel çalışmalarda, günlük hayatımızda ve sanayini alanında yaygın olarak kullanılmaktadır. Laboratuvar ortamında gerçekleştirilen deneyler, sulama ve havalandırma sistemleri, petrol ve doğalgaz boru hatları, otomotiv sektöründeki yakıt tüketim sistemleri, sıvı ölçümleri gibi önemli veriler akış ölçüm prensiplerinden faydalanmaktadır [7], [9]. Akış ölçümü gerçekleştirmenin dayandırıldığı farklı prensipler bulunmaktadır ve bu prensiplere

dayalı birçok akış ölçer tasarlanmıştır. Akışlar; akış hızı, akışın sürekliliği, akıştaki faz sayısına göre sınıflandırılabilir.

Akış hızlarına göre ise laminar (düzgün) akış ve türbülanslı (girdaplı) akış olarak ikiye ayrılırlar [7]. Düzgün ve girdaplı akış ile ilgili birçok çalışma mevcuttur. Düzgün akış rejiminden, girdaplı akışa geçiş ilk defa 1889 yılında Osborne Reynolds tarafından gerçekleştirilen boya deneyi ile gözlemlenmiştir [8]. Bir akışın düzgün veya girdaplı bir akış olup olmadığı Reynolds katsayısı ile belirlenebilir. Reynolds katsayısı yükseldiğinde düzgün akış rejimine sahip boru içerisindeki akışlarda girdaplar oluşmaya başlamakta ve düzgün akış rejiminden çıkmaktadır [4].

Akışkanlar sürekliliğine göre ise pulsatil ve sabit akış olarak ikiye ayrılabilir [13]. Kalbin kan pompalaması veya bir sütün sağımında oluşan akış pulsatil akışa birer örnektir. Sabit akış ise değişken olmayan bir devirle pompalanan sıvı olarak kabul edilir [15].

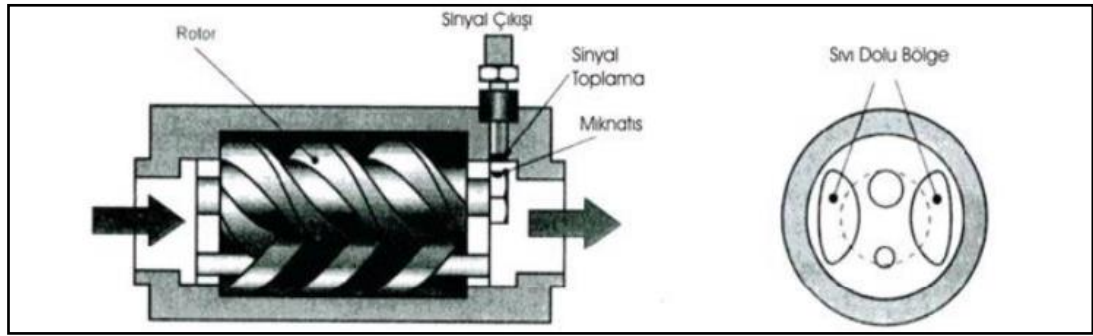
Akışkan ayrıca çoklu ve tekli akışkan olarak ikiye ayrılabilir. Akış içerisinde iki veya üç fazda (katı-gaz, gaz-sıvı, katı-sıvı-gaz) bir akışkan varsa bu akışkana çok fazlı akışkan denilmektedir [10]. Sağım esnasında borudan geçen süt çift fazlı bir akışkandır. Süt sağımı vakumla yapıldığından, akışkan hem süt hem de havadan oluşmaktadır. Dolayısıyla sağım esnasındaki akış çift fazlı bir akışkandır. Tek fazlı akışkan ise, akışkanın sadece tek fazda (gaz veya sıvı) olduğu durumlara denir [13], [15]. Sıvı ölçümünde sektörde kullanılan akış ölçerler; kütleli, hacimsel ve elektronik akış ölçerler olarak üç grupta sınıflandırılabilir.

1.2.1. Hacimsel Akış Ölçerler

Hacimsel akış ölçerler; helisel vidalı akış ölçerler, döner lop akış ölçerler, türbinli akış ölçerler, çark sistemli akış ölçer, diskli sayaçlar olmak üzere beş farklı grupta toplanabilir [1], [2], [3], [9], [11].

1.2.1.1. Helisel Vidalı Akış Ölçer

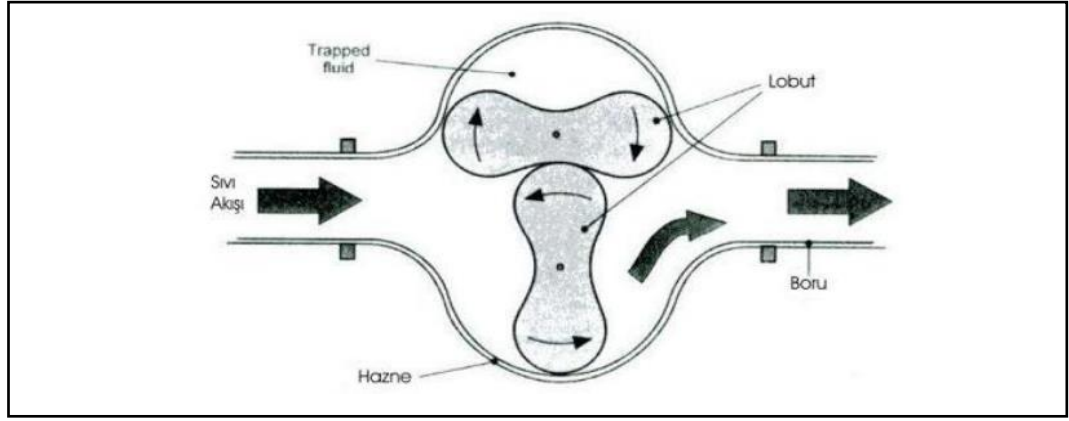
Helisel vidalı akış ölçer sistemi, sıvı akışkan geçişi ile beraber dönme hareketi yapmaktadır. Helisel dişliler arasında dolan sıvı çıkış tarafına doğru hareket etmesi ile dişlilerin dönmesini sağlamaktadır. Helisel dişlilerin adımı ve helisel dişlilerin aralarındaki hacim değeri bilindiği için, her tam bir dönüşte helisin kapasitesi kadar sıvı çıkışı olacaktır. Şaft üzerine yapıştırılan bir mıknatıs ve stator üzerindeki hissedici ile toplam devir sayısı hesaplanır. Bir devirde geçen akışkanın hacmi ile çarpılarak toplam akış miktarı ölçülmektedir. [1], [3], [9], [11], [12].



Şekil 1.1: Helisel vidalı akış ölçer.

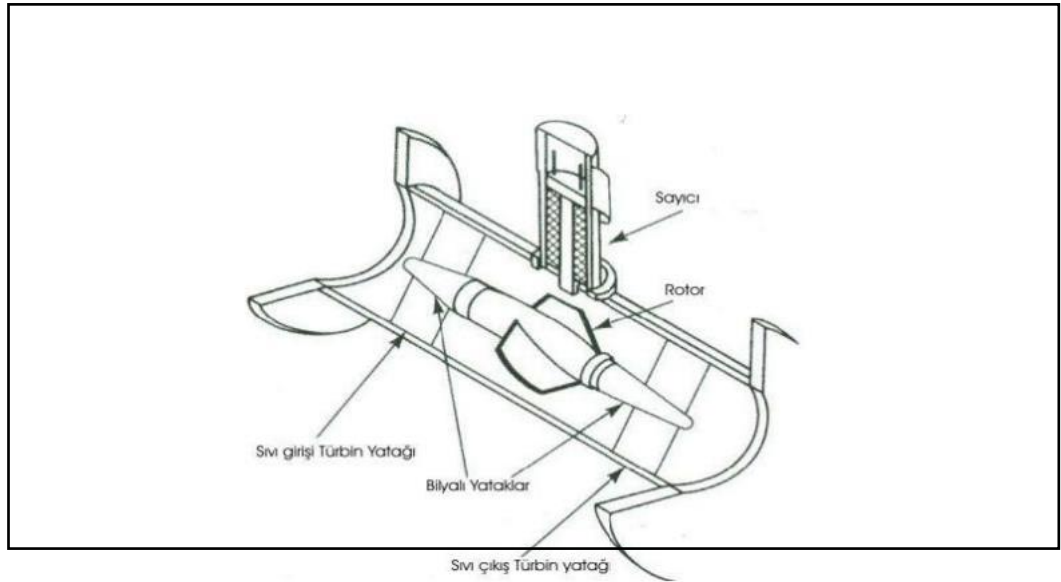
1.2.1.2. Döner Lop Akış Ölçer

Döner lop akış ölçer sistemi, iki adet lop saat yönünde dönmektedir. Akış halindeki sıvı yukarıdaki ve aşağıdaki boşluklara dolmaktadır ve bu sıvı lop ile dış kap arasında sıkışarak saat yönünde itilmektedir. Her bir devirde gönderilen akışkanın hacmi belirli bir miktardadır. Bir sayıcı vasıtası ile lopların yaptığı tur sayısı hesaplanarak, akışkanın toplam miktarı hesaplanmaktadır. Çıkış devri lopların herhangi birine bağlanan bir numaratör, optik algılayıcı veya elektriksel bir dönüştürücü ile ölçülebilir. [1], [3], [9], [11], [14].



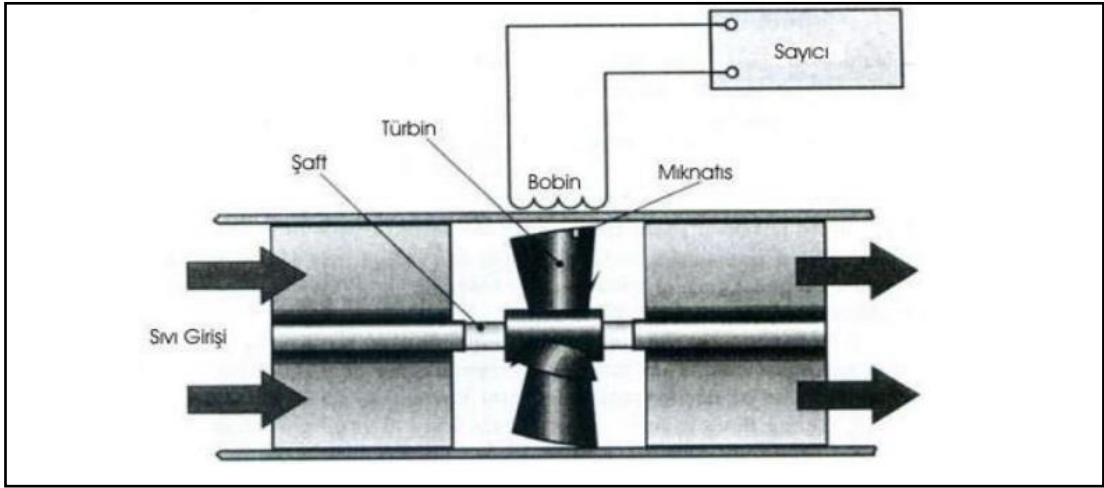
Şekil 1.2: Döner lop akış ölçer.

1.2.1.3. Türbin Akış Ölçer



Şekil 1.3: Türbinmetre.

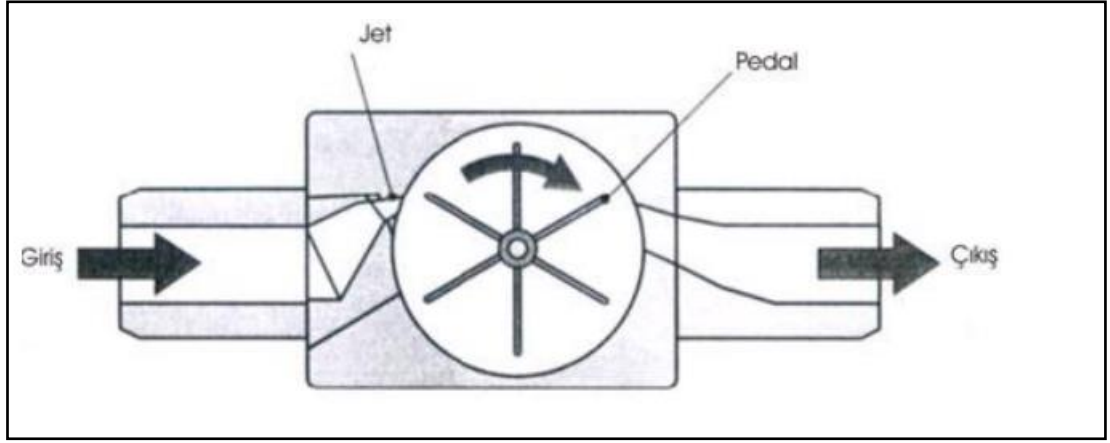
Türbinmetre akışkanın hem hacmini hem de hızını ölçmektedir. Akışkan türbine çarparak hızı ile doğru orantılı olarak kanatçıkları çevirmektedir. Dönen türbin kendisine bağlı mili çevirmektedir. Türbin kanatlarının birine yerleştirilen bobin ile devir sayısı belirlenir. Dönüş hareketi mekanik ve elektriksel sistemler vasıtası ile elektrik sinyaline dönüştürülerek miktar ölçümü yapılmaktadır. Türbinmetreler yüksek hassasiyete sahip olmalarına rağmen türbin giriş ve çıkışında basınç düşümüne neden olması ve maliyeti yüksek olduğu için tercih edilmez. Ayrıca düşük akış oranlarında ölçüm hassasiyetinde bozulmalar olduğu için verimli bir ölçüm yapamayabilirler [1], [2], [3], [9], [11], [14].



Şekil 1.4: Türbinmetre devir ölçümü.

1.2.1.4. Çark Sistemli Akış Ölçer

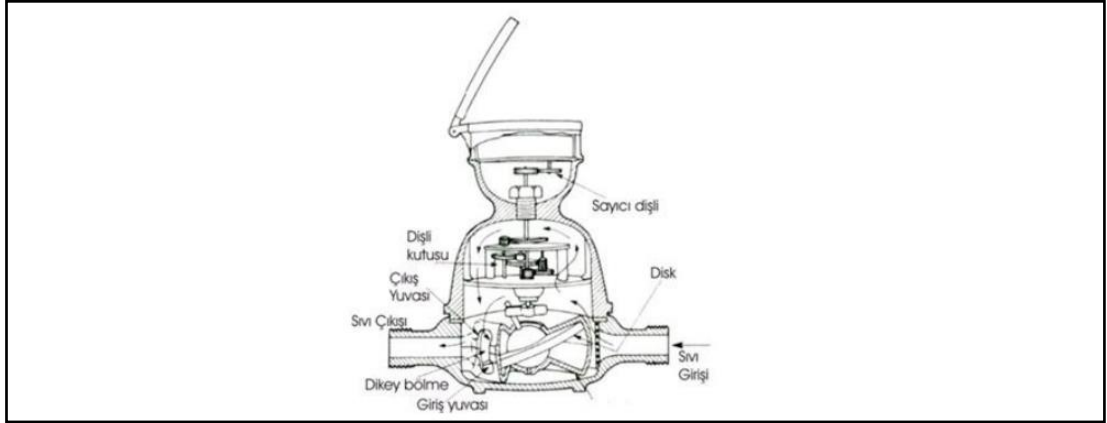
Boru içerisinden akmakta olan sıvı bir nozul vasıtasıyla hızlandırılarak çarkı döndürmektedir. Sıvı akışı ile orantılı dönen çark mekanik veya elektriksel yöntemlerle hacimsel miktarı belirleyebilmektedir. Bu sistemde de nozuldan kaynaklı bir basınç düşümü olmaktadır [1], [2], [3], [9], [11], [12].



Şekil 1.5: Çark sistemli akış ölçer.

1.2.1.5. Diskli Sayaçlar

Diskli sayaçlar ev, ofis, işyeri gibi yerlerde su ölçümünde kullanılan klasik sayaçlardır. Sayacın içerisinde bulunan disk yalpalı bir dönüş yapmaktadır. Bir tam dönüşte belirli bir hacime sahip sıvıyı girişten alıp çıkışa göndermektedir. Sıvının akış hızına veya debisine göre diskin bu hareketi artmakta veya azalmaktadır. Diskin bu hareketi ile oluşan her bir salınım akışkan miktarınca dişli çarklarda işlem görür. Dişli çarkların diş oranları birim zamandaki diskin iletmış olduğu hacim miktarını göstermektedir [1].



Şekil 1.6: Diskli sayaçlar.

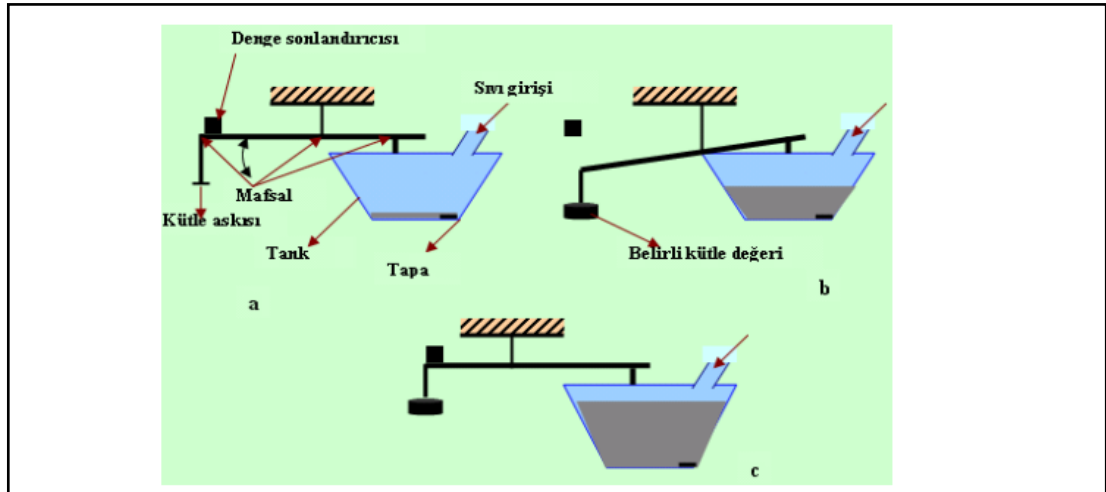
1.2.2. Kütlesel Akış Ölçer

Kütlesel akış ölçerler; kuvvetler dengesi ile kütlesel akış ölçer, değişken alanlı akış ölçer, pitotboru, anemometre, ventüri boru, orifis akış ölçer, nozul akış ölçer olmak üzere yedi farklı başlıkta incelenebilir.

1.2.2.1. Kuvvetler Dengesi ile Kütlesel Akış Ölçer

Bu tip ölçümlerde kullanılan en basit yöntem akışkan miktarını direkt ölçmektir. Bir sokak sütçüsünün 1 kg diye verdiği süt (1L süt #1kg süt) yanlış bir ölçüm de olsa kütlesel bir ölçümdür.

Aşağıdaki şekilde yer çekimi kuvvetiyle ortasından asılarak dengelenmiş bir çubuğun bir tarafına sıvı miktarının ölçülmesi için gereken ağırlık, diğer tarafına ise ölçülecek sıvının kabı konulmaktadır. Kabin içerisinde sıvı doldukça, ölçülmek istenen ağırlık çubuğun diğer tarafındaki ağırlığın değerine yaklaştıkça çubuk dengelenecek ve bir sensöre temas ederek kütlesel olarak akış işlemini sonlandıracaktır. Sıvının bulunduğu kap içerisinde akışkan seviyesi azalırsa sistemin dengesi bozulacaktır ve sensör, bir değer okumayacağı için akış işlemini başlatacaktır [1], [2], [3], [9], [11], [14], [16].

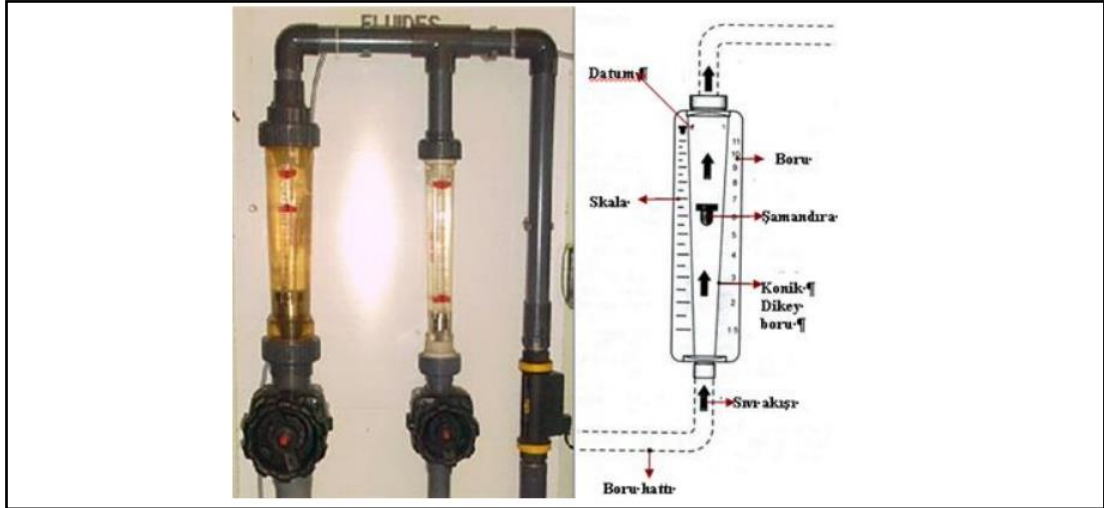


Şekil 1.7: Kütlesel akış ölçer.

1.2.2.2. Değişken Alanlı Akış Ölçer

Değişken alanlı akış ölçer, eşit bir şekilde bölünmüş cam bir tüp ve içerisine yerleştirilmiş şamandıradan oluşan bir düzendir. Akışkanın cam tüp içerisinde bulunan şamandıranın yaptığı basınçla şamandırayı yukarı kaldırır. Basınçla yukarı kalkan şamandıranın bulunduğu yükseklikteki skala değeri, cam tüp ile şamandıra arasından geçen akışkan miktarını, birim zamanda geçen kütle miktarı (g/sn) olarak göstermektedir [1], [2], [3], [9], [11], [14].

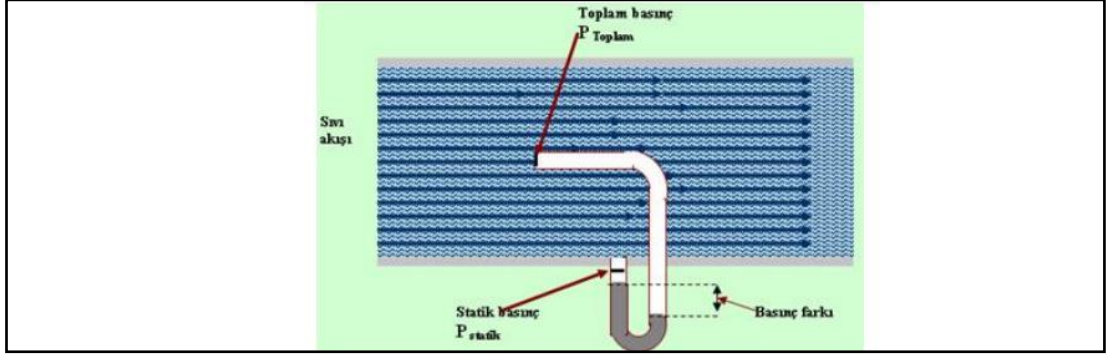
Akışkan ölçümünün doğru bir şekilde gerçekleştirilebilmesi için ölçüm esnasında şamandıra ile yüzeyler arasında herhangi bir sürtünme olmaması veya sıvı geçişini engelleyecek dirençler göstermemesi gerekmektedir.



Şekil 1.8: Değişken alanlı akış ölçer.

1.2.2.3. Pitot Boru ile Hız Ölçümü

Pitot boru, boru merkezine, boru ağzı akış yönüne ters gelecek şekilde yerleştirilir. Pitot borunun diğer ucu da boru yüzeyine yerleştirilmiş bir U manometre düzeneğidir. Boru merkezinde toplam basınç değerini, boru yüzeyinde ise statik basınç değerini ölçüp manometrede fark basıncını göstermektedir.

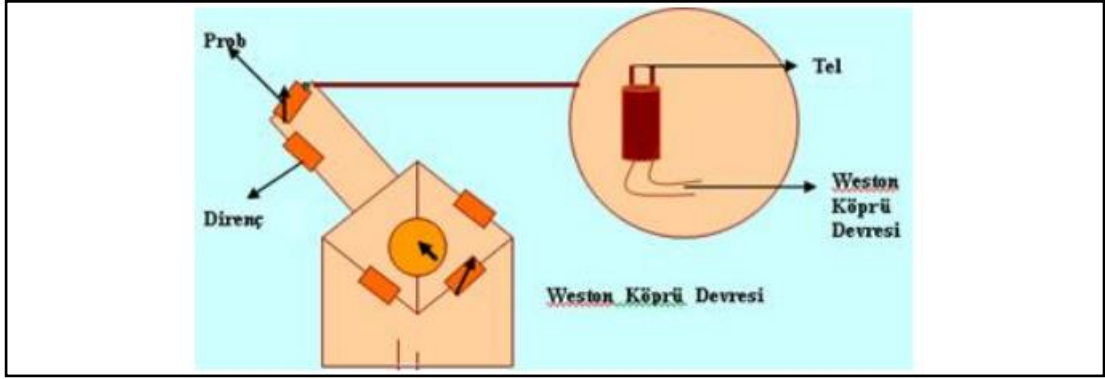


Şekil 1.9: Pitot boru.

Pitot boru ile basınç farkı bulunur. Bulunan basınç farkından akışkan hızı bulunabilmektedir. Boru kesitide bilindiğine göre boru içerisinden geçen akışkan miktarı bulunmaktadır.

1.2.2.4. Anemometre (Gaz Ölçer)

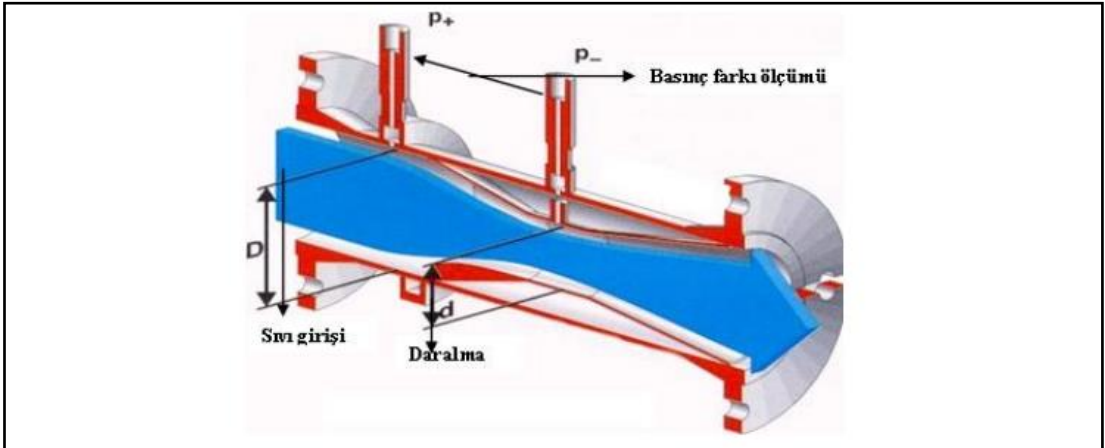
Anemometreler gaz ölçümleri için kullanılmaktadır. Çalışma prensibi iletkenin sıcaklık ile direnç değişikliğine dayanmaktadır. Weston köprü devresine iletkenin bağlanması ile kullanılır. Anemometreler pitot boru ile aynı görevi yapmalarına rağmen pitot borulara göre çok daha pahalı bir sistemdir. Ancak anemometre kullanılan telin özelliğine göre hassas ölçüm değerleri verir ve geniş bir ölçüm aralığına sahiptir [1], [2], [3], [9], [11], [14], [16].



Şekil 1.10: Anemometre.

1.2.2.5. Ventürimetre ile Hız Ölçer

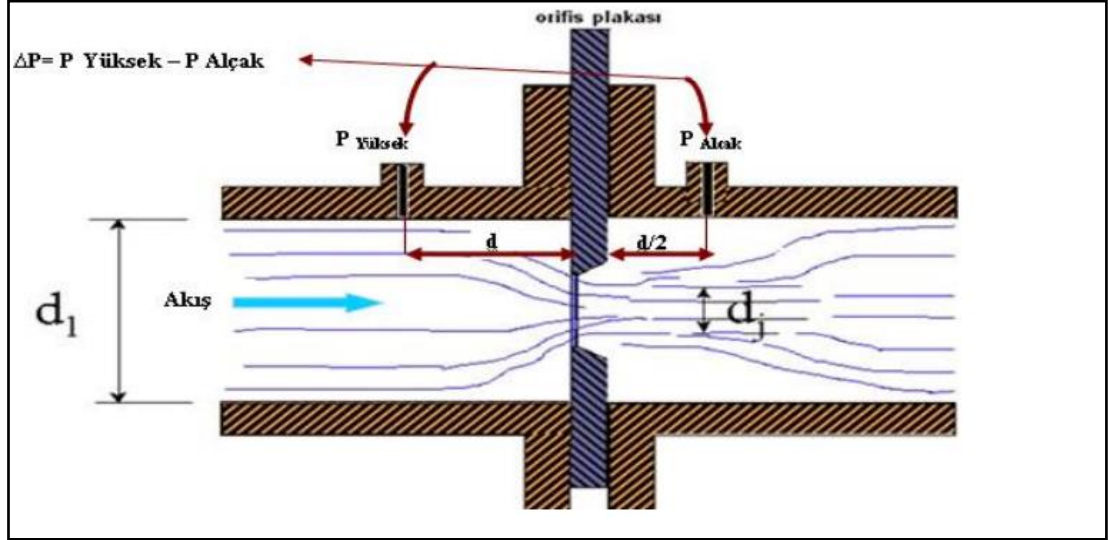
Ventürimetre, akışkanın geçtiği boruya bağlanır. Akışkan boru içerisinde ventüri boruya geldiğinde daraltılmış bir kesit alanına girer. Ventüri boruda bu daralma girişte ve çıkışta hafif eğimler ile sağlanır. Bu daralma sıvı akışının akış özelliğini değiştirmez. Buradaki amaç akışkanı hızlandırmak ve akışkan basıncını düşürmektir. Ventürimetre ile ölçüm işlemi, geniş ve dar noktadaki basınç farkından faydalanarak, akışkanın hızı ölçülmektedir [1], [2], [3], [9], [11], [12].



Şekil 1.11: Ventürimetre.

1.2.2.6. Orifis ile Akış Ölçer

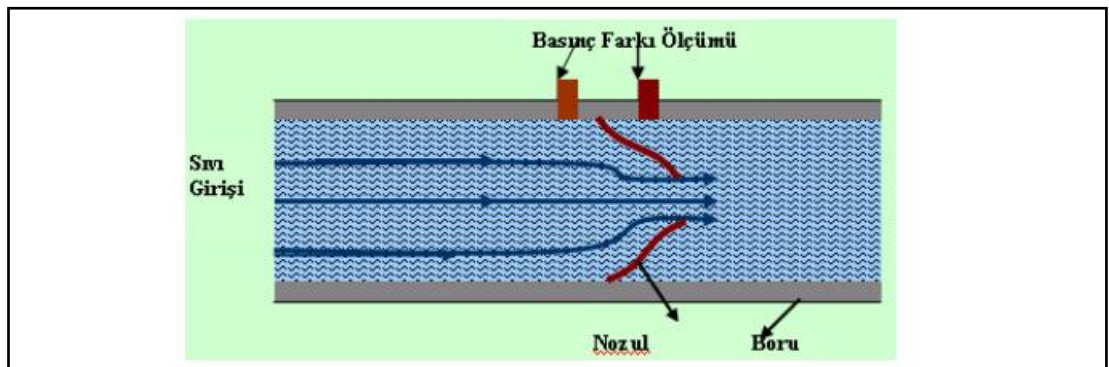
Boru içerisine içi delik bir levha ile akış engellenerek daha küçük bir delikten geçmesi sağlanan bir akış ölçer sistemidir. Giriş noktasında yüksek basınç çıkış noktasında ise alçak basınç oluşturulur. Bu basınç farkından yararlanarak akış hızı ölçülmektedir [1], [2], [3], [9], [10], [12], [16].



Şekil 1.12: Orifis levha akış ölçümü.

1.2.2.7. Nozul Akış Ölçer

Orifis ile akış ölçme yöntemine çok benzemektedir. Akışkanın giriş ve çıkış anında basınç farkı oluşturulur ve oluşturulan bu basınç farkı arasındaki ölçüm ile akışkan hızı belirlenmektedir [1].



Şekil 1.13: Nozul akış ölçümü.

1.2.3. Elektronik Akış Ölçerler

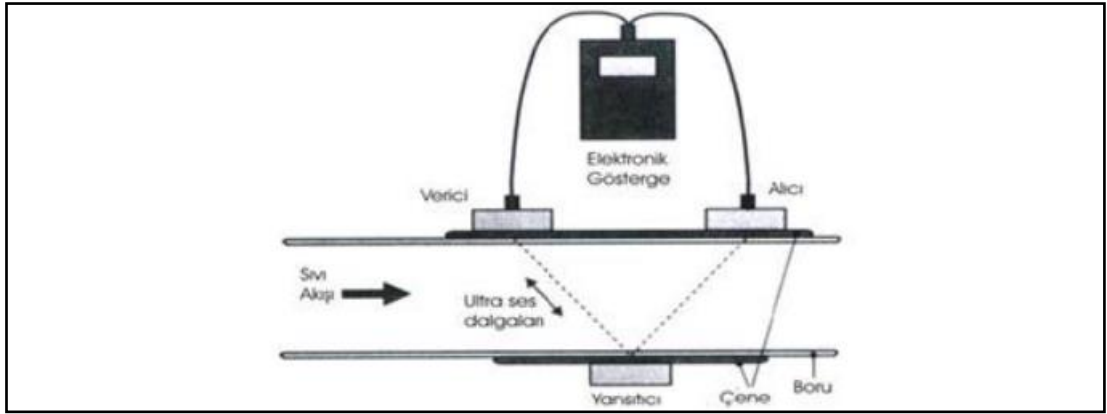
Elektronik akış ölçerler, temassız bir şekilde temel fizik prensiplerini kullanarak ölçüm yapmaya yarayan sistemlerdir. Basit mekanik tasarımına sahip olmaları, basınç kaybı olmaması, kurulumlarının ve kullanımının basit olması, maliyetler olarak diğer ölçüm yöntemlerinden daha avantajlı olması, içerisinde hareketli parça bulundurmamaları, uzun ömürlü olmaları ve yıpranmalarının az olması, yüksek hassasiyet ve doğruluk gibi etkenler elektronik akış ölçerlerin en büyük avantajlarıdır. Diğer akış ölçerlere göre dış etkenlerden daha fazla etkilenmeleri ise elektronik akış ölçerlerin en büyük dezavantajlarıdır [5].

1.2.3.1. Ultrasonik Akış Ölçer

Temel prensibi Avusturyalı Fizikçi Christian Doppler'in çalışmaları ile oluşmuştur. Christian Doppler, hareket eden ses kaynağındaki frekans veya dalga boylarında kaymalar oluştuğunu gözlemlemiştir. Bu gözlemlere dayanan Ultrasonik Doppler geliştirilmiş ve akış ölçer olarak kullanılmaya başlanmıştır [5], [6], [17], [19], [20].

Ultrasonik akış ölçerlerde, ses dalgaları boru içerisine açılı bir şekilde gönderilir ve bu dalgalar yansıtıcıdan yansıtılarak gönderilme açısına uygun bir toplama açısı ile alıcıda toplanır. Boru içerisinden sıvının akmasıyla vericiden gönderilen ses dalgaları ile alıcının algılanması arasında geçen süre değişimi sıvı akışıyla ilgilidir. Alıcı ve verici arasındaki değişim süresi akış hızıyla lineer bir değişim gösterir. Bu değişim göstergede akış hızı olarak kalibre edilir.

Ultrasonik akış ölçerler pahalı olmasına karşılık hassas ve kullanışlı bir cihaz olması, akışı çift yönlü ölçebilmesi, herhangi bir basınç düşümüne sebep olmaması bir avantajdır. Özellikle açık alandaki doğal su kaynakların debilerinin ölçülmesinde kullanılan ultrasonik akış ölçerler, tam dolu ve düzgün akış olması koşulu ile kapalı kanallardaki bütün akışkanların ölçümü ve damarlardaki kan akışının ölçümü gibi, birçok farklı yerde de kullanılmaktadır [21]. Ultrasonik akış ölçerler ile akış içerisindeki kayma hızları belirlenebilmektedir. Bu sayede kayma gerilimlerinin oranları yardımı ile sıvının viskozitesi hakkında da bilgi elde edilebilmektedir [22].

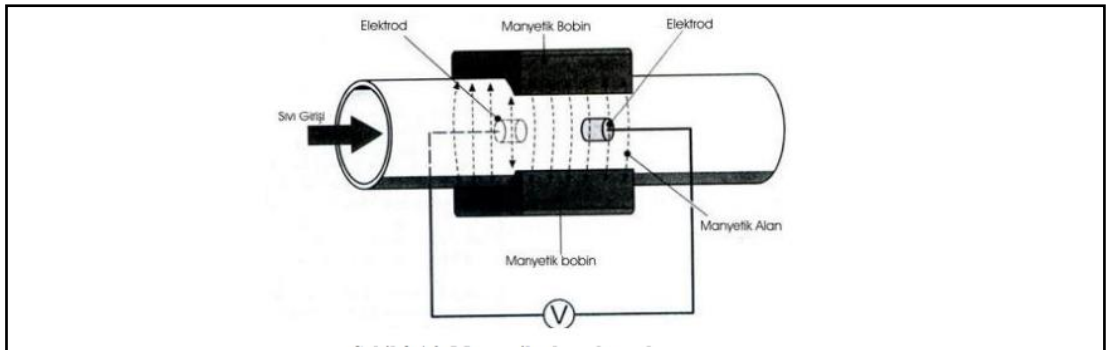


Şekil 1.14: Ultrasonik akış ölçer.

1.2.3.2. Elektro Manyetik Akış Ölçerler

Temel prensibi Michael Faraday'ın manyetik indüksiyon prensibi çalışmalarına dayanmaktadır. Faraday'ın bu kanuna göre manyetik alanda hareketli bir iletken üzerinde elektriksel bir gerilim oluşur ve bu gerilim akışkan hızı ile doğru orantılıdır.

Elektro manyetik akış ölçerlerde sıvıyı iletken olarak kullanırlar. Elektro manyetik akış ölçere seri bağlı bobinler enerjilendiği zaman, sıvı etrafında akım etkisi ile bir manyetik alan oluşturur. Oluşan manyetik alana ve sıvı akışına dik olacak şekilde iki iletken yerleştirilir. Boru içerisindeki sıvı akışkanın hız değişimi, manyetik alanda etki yaparak değişimi elektrotlara iletir, elektrotlar da ölçü aletine sinyal gönderir. Ölçü aletinin almış olduğu sinyalle göndermiş olduğu değişim, akış hızının değişimi ile lineer bir değişim gösterir.



Şekil 1.15: Manyetik alan akım ölçümü.

Bu bölümde, farklı akışkan ölçüm yöntemleri ile literatürde tasarlanmış mekanik ve elektriksel akış ölçerler hakkında genel bilgiler verilmiştir. Bundan sonraki bölümde ise sıvının ölçümü için kullanılan yöntemler anlatılmıştır.

2. SIVI MİKTARININ ÖLÇÜLMESİ

2.1. Sıvı Ölçümünün Önemi

Sıvılarda miktar, sıcaklık, iletkenlik gibi ölçüm değerleri, çeşitli endüstriyel, ticari ve evsel uygulamalarda önemli bir rol oynayan bir ölçüm türüdür. Sıvıların doğru bir şekilde ölçülmesi, ürün kalitesinin kontrol edilmesi, üretim süreçlerinin yönetilmesi ve ürünlerin standartlarının belirlenmesi için önemlidir.

Sıvı ölçümü, çeşitli ölçüm cihazları kullanılarak gerçekleştirilir. Bu cihazlar arasında akış metreler, debimetreler, seviye ölçerler ve hacim ölçerler bulunur. Bu cihazlar, sıvıların akış hızı, debisi, seviyesi ve hacimlerini ölçmek için kullanılır.

Sıvı ölçümü, endüstriyel ve ticari uygulamalarda yaygın olarak kullanılır. Örneğin, kimya endüstrisinde, sıvıların doğru miktarlarda ölçülmesi, ürün kalitesinin kontrol edilmesi ve üretim süreçlerinin optimize edilmesi için önemlidir. Gıda endüstrisinde, sıvıların doğru miktarlarda ölçülmesi, ürünlerin standartlarının belirlenmesi ve üretim süreçlerinin yönetilmesi için gereklidir. Benzer şekilde, petrol ve gaz endüstrisinde de sıvı ölçümü, petrol ve gazın çıkarılması, taşınması ve depolanması için hayati bir öneme sahiptir. Ticari alanlarda sıvının temizliği, verimli olarak değerlendirilmesi gibi konularda ölçümün doğru ve eksiksiz olarak yapılması önem arz ediyor.

Sıvı ölçümü, evsel uygulamalarda da yaygın olarak kullanılır. Örneğin, evlerde su tüketimini ölçmek için su sayaçları kullanılır. Bu sayaçlar, su faturalarının hesaplanması ve su tüketiminin takibi için kullanılır.

Sonuç olarak, sıvı ölçümü, endüstriyel, ticari ve evsel uygulamalarda hayati bir rol oynayan bir ölçüm türüdür. Sıvıların doğru bir şekilde ölçülmesi, ürün kalitesinin kontrol edilmesi ve üretim süreçlerinin yönetilmesi için gereklidir. Günümüz ekonomik koşulları sıvıların üretiminde rekabet oluşturmuş ve kaliteli üretim zorunluluğu ön plana çıkmıştır. Endüstrinin her alanında olduğu gibi bu alanda da temel amaç yüksek karlılıktır. Yüksek karlılığın sağlanması içinde, üretilen sıvı gıdanın temiz, kaliteli ve verimli üretiminin sağlanması ve kayıt altına alınması ihtiyaç haline gelmiştir. Hayatımızın her alanında kullandığımız suyu ve yüksek miktarda tüketilmesinden dolayı ölçümlerinin tam ve hatasız yapılması, her üretim bandındaki verimin hesaplanması, hangi su şişesine hangi tarihte hangi değerlerdeki suyun

doldurulduğuna kadar kayıt altına alınma zorunluluğu veya süt sağım hanelerinde sağılan sütün miktarının tam olarak ölçülmesi, hatta her sağım sonrasında alınan süt miktarının belirlenmesi, hayvan ve gıda sağlığının takibi ve ekonomik açıdan verimliliğin artırılması gerekliliğinden ötürü sıvılarda miktar ölçüm işlemini zorunlu kılmıştır.

2.2. Sıvı Ölçüm Yöntemleri

2.2.1. İki Fazlı Akış Ölçümü

İki fazlı akış ölçümü, sıvı ve gazın birlikte akışının olduğu durumlarda hem sıvı hem de gaz fazının akış hızını ölçmek için kullanılan bir yöntemdir. Bu durum, özellikle petrol, gaz ve kimya endüstrilerinde karşılaşılan bir durumdur.

İki fazlı akış ölçümü, doğru ve güvenilir bir şekilde gerçekleştirildiğinde, sıvı ve gaz fazlarının ayrı ayrı ölçülmesi mümkün olmayan durumlarda önemli bir ölçüm yöntemidir. Bu yöntem, endüstrideki proses kontrolü ve kalite kontrolü gibi birçok uygulamada kullanılır.

İki fazlı akış ölçümü, iki farklı yöntem kullanılarak gerçekleştirilebilir.

2.2.1.1. Ayırıştırıcı Tabanlı Ölçüm

Bu yöntemde, sıvı ve gaz fazları ayırıştırılarak her bir fazın ayrı ayrı ölçülmesi sağlanır. Bu amaçla, akışkanlar öncelikle bir ayırıştırıcıya yönlendirilir ve burada sıvı ve gaz fazları ayırıştırılır. Daha sonra, her iki fazın ayrı ayrı ölçülmesi için özel sensörler kullanılır.

2.2.1.2. Karışık Akış Ölçümü:

Bu yöntemde, sıvı ve gaz fazları ayrılmadan birlikte ölçülür. Bunun için, sıvı ve gazın özelliklerini hesaba katan bir modele dayanan bir sensör kullanılır. Bu sensörler, sıvı ve gaz fazlarının birbirine oranına ve akış hızına bağlı olarak çeşitli parametreleri ölçerler. Bu yöntem, ayırıştırıcı tabanlı ölçüme göre daha ekonomiktir, ancak daha az hassastır.

2.2.2. Çift Fazlı Pulsatil Akış Ölçümü

Çift fazlı pulsatil akış, bir boru içinde iki farklı fazın (genellikle sıvı ve gaz) bir arada ilerlerken, akışkanların dalgalı bir şekilde hareket ettiği bir akış türüdür. Bu tür akışlar, genellikle düşük Reynolds sayılarına (akışkanın viskozitesinin hızına oranı) sahip olan sistemlerde görülür ve sıvı ve gaz fazlarının dinamikleri arasındaki karşılıklı etkileşimlerden kaynaklanır.

Çift fazlı pulsatil akış, bir dizi faktör tarafından etkilenir. Bunlar arasında boru geometrisi, akışkanların fiziksel özellikleri, akış hızı, basınç düşümü, sıvı-gaz oranı ve sıcaklık gibi faktörler yer alır. Bu faktörler, akışın dalgalanma frekansı, dalga boyu, dalga yüksekliği ve dalga formu gibi parametrelerin belirlenmesinde rol oynar.

Bu tür akışlar, birçok endüstriyel uygulamada karşılaşılan bir durumdur. Örneğin, petrol endüstrisinde, petrol ve gazın bir arada akışı sırasında çift fazlı pulsatil akış oluşabilir. Ayrıca, çift fazlı pulsatil akış, tıbbi uygulamalarda da önemli bir role sahiptir. Örneğin, kalp atışlarını taklit eden pulsatil akış simülasyonları, tıbbi cihazların test edilmesinde kullanılabilir.

Bu tez kapsamında sıvıların hareketleri, endüstriyel üretim alanları ve üretim maliyetleri göz önünde bulundurulduğunda en öne çıkan yöntem çift fazlı pulsatil akış olmaktadır.

2.2.2.1. Lazer Doppler ile Akış Ölçümü

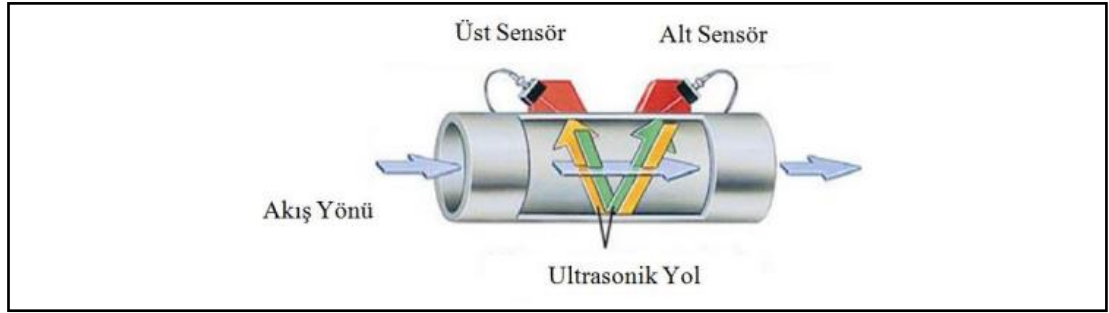
Doppler etkisi, bir dalga kaynağının hareket ettiği bir gözlemciye yaklaştığında veya uzaklaştığında, dalga frekansında bir değişiklik olmasıdır. Bu etki, ses dalgaları, elektromanyetik dalgalar ve diğer dalga türleri için geçerlidir.

Lazer Doppler Anemometresi (LDA), Doppler etkisini kullanarak sıvı veya gaz akış hızlarını ölçmek için kullanılan bir optik ölçüm tekniğidir. Bu teknik, akışkan içinde hareket eden partikülleri lazer ışınıyla aydınlatarak, partiküllerin hareketine bağlı olarak yansıyan ışının frekansındaki değişiklikleri ölçer.

LDA'da, lazer ışını, bir mercek sistemi aracılığıyla akışkanın içine odaklanır. Lazer ışını, akışkan içinde hareket eden partikülleri aydınlatır ve partiküllerin hareketine bağlı olarak yansıyan ışın, fotoğraf detektörüne geri yansıtılır. Yansıyan ışın, geleneksel bir ışık kaynağından farklı olarak, Doppler etkisine tabidir. Bu

nedenle, fotoğraf detektörü tarafından alınan sinyalin frekansı, partikül hareketine bağlı olarak değişir. Bu değişiklik, partiküllerin hızı ve yönü hakkında bilgi sağlar.

LDA, sıvı veya gaz akış hızlarını hassas bir şekilde ölçebilen güçlü bir optik ölçüm tekniğidir. Endüstriyel uygulamalar, tıp, havacılık, nükleer enerji ve araştırma laboratuvarları gibi birçok alanda kullanılmaktadır.



Şekil 2.1: Ultrasonik yöntemle akışkan ölçümü.

2.2.2.2. Parçacık Görüntülemeli Hız Ölçüm Yöntemi

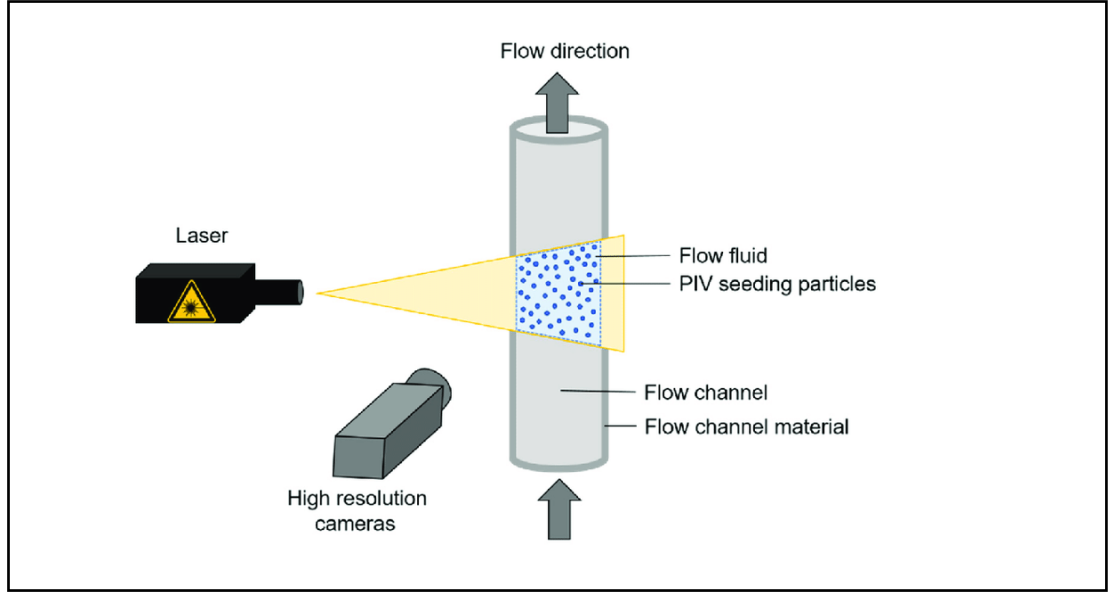
Parçacık görüntülemeli hız ölçümü (PIV), bir lazer ışını tarafından aydınlatılan bir akışkan içindeki parçacıkların hareketini takip ederek akış hızlarının ölçülmesini sağlayan bir optik ölçüm tekniğidir.

Parçacık görüntülemeli hız ölçümü, akışkan içindeki partikülleri aydınlatmak için lazer ışını kullanır. Lazer ışını, bir mercek sistemi aracılığıyla akışkanın içine odaklanır ve parçacıkları aydınlatarak bir kareye yansıtır. Bir kamera, bu kareyi alır ve parçacıkların hareketlerini takip eder. Ardından, bir sonraki kareye geçilir ve aynı işlem tekrarlanır. İki kare arasındaki parçacık hareketleri, akış hızını ve yönünü belirlemek için analiz edilir.

Parçacık görüntülemeli hız ölçümünün en büyük avantajı, akışkanın herhangi bir bölgesindeki akış hızlarının ölçülebilmesidir. Ayrıca, bu teknik, akışkanın özellikle hassas ve ince bir bölgesindeki hız değişikliklerini ve akışkan içindeki dönüşümleri de tespit edebilir.

Parçacık görüntülemeli hız ölçümü, akışkan mekaniği, aerodinamik, biyofizik, tıp ve diğer birçok alanda kullanılan güçlü bir araştırma aracıdır. Özellikle, turbülans

ve çift fazlı akışlar gibi karmaşık akış problemlerinin incelenmesinde etkili bir yöntemdir.



Şekil 2.2: Parçacık görüntülemeli ölçüm.

3. SIVI AKIŞININ IŞIK ŞİDDETİ İLE ÖLÇÜLMESİ

3.1. Optik Yöntem

Sıvı ölçümlerinde iletkenlik kullanılarak bir çalışma gerçekleştirilebilir. Ancak iletkenlikle yapılacak bir ölçümün sıvının geçiş hızına ve sıvının içerisinde bulunan havaya bağlı olarak hassasiyet oranı düştüğünden dolayı bu çalışmada optik yöntem kullanma kararı alınmıştır. Ayrıca sıvı ölçüm sistemlerinde her zaman boru içerisinden geçen sıvı lineer bir akışa sahip olmayabilir. Bazı sıvılar çift fazlı akış özelliğine sahip olmasından dolayı klasik ölçüm sistemleri ile sağım esnasında bütün sıvıların ölçümünün yapılması mümkün olmamaktadır. Akış esnasında sıvının miktarının, hızının ve yükseklik değerinin bilinmesi ile bir hesaplama yapılması gerekmektedir. Aşağıda bu hesaplama için kullanılan sıvı hacmi ve sıvı akış hız formülleri bulunmaktadır. Hesaplama formülü ve formülün oluşturulması oluşturulan yazılım algoritması içerisinde bulunmaktadır. Aşağıda verilen formüllerde “r” yarıçapı, “h” yüksekliği, “x” mesafeyi ve “t” zamanı ifade etmektedir.

$$\text{Sıvının Hacmi} = \pi r^2 h \quad (3.1)$$

$$\text{Sıvının Hızı} = x/t \quad (3.2)$$

3.1.1. Sıvı Miktar Ölçüm Yönteminin Tasarımı

Sıvı ölçer üzerinde karşılıklı olarak 2x5 alıcı ve verici infrared LED bulunmaktadır. Bu LED’ler sıvının miktarını ve içerisindeki hava oranını tespit etmektedir. Verici LED’den çıkan kızılötesi ışınlar alıcı LED’e ulaşmaktadır. Bu ışınlar alıcı LED’de geçtiği yere göre bir değer oluşturmaktadır. Bu ışınlar sadece hava ve sadece sıvıdan geçtiklerinde farklı değerler vermektedir. Sıvı ve hava beraber olduğu zaman ise ışınlar sıvıdaki hava miktarına göre, sadece hava ve sadece sıvıda

geçtiklerindeki değerlerin arasında bir değer alıcılarda okunacaktır ve bu sayede sıvının içerisindeki hava miktarı belirlenebilecektir. Örneğin; sadece hava dolu bir akışta bu infrared LED'lerden okunan değer -ışık kırılımı minimal düzeyde olacağından dolayı maksimum olacaktır. Sadece sıvı dolu bir akışta bu infrared LED'lerden okunan değer -ışık kırılımı maksimal değerde olacağından dolayı minimum olacaktır. Hava ve sıvı karışımı olan bir akışta ise ışık kırılımlarının sıvıdaki hava oranına göre değişkenlik göstereceğinden dolayı okunan değer maksimum ile minimum arasında bir değer olacaktır. Buradan yola çıkarak okunan değere göre sıvının içerisindeki hava miktarı bulunabilecektir. Bu infrared LED'ler bu işi sadece karşılıklı olarak değil çapraz şekillerde de yaparak sıvının içerisindeki hava miktarını algılanan ışık şiddeti değeri ile daha da hassas bir şekilde bulmamıza olanak sağlayacaktır.

Bu çalışmamızda infrared sensörleri kullanılarak sıvının miktarı ölçülmektedir. Bu da sıvının miktarının oluşturulan algoritma hesabı ile birlikte daha hassas ölçülmesini sağlayacaktır. Sıvının, cihazın içinden geçeceği ilk anındaki ve son anındaki iletkenlikleri ölçülerek hem sıcaklıkları ölçülecek hem de ölçüm değerinin hassasiyetini arttıracaktır.

Çalışmamızda hassasiyeti sağlayabilmek için ayrıca dikkat ettiğimiz bir başka konu ise infrared alıcı ve verici sensörlerinin açısız olarak yerleştirilmesidir. Karşılıklı olarak yerleştirilen infrared alıcı ve verici sensörlerin görüş açılarından dolayı birbirlerini etkilemesi problemidir. Sensörlerin birbirlerine olan uzaklığını hassas bir şekilde hesaplayarak ve aralarındaki etkileşimi azaltmak için 15 derecelik yayılım yapabilen ayarlanabilir LED kullanılarak çözülmüştür. Infrared LED'lerin karşılıklı konulması gerekliliğinden dolayı bazı problemler ortaya çıkmıştır. Sıvının cihazın içinden geçtiği yerin infrared sensörlerin çalışmasını etkileyecek şekilde olmasından kaynaklı problemi, geçtiği yerin şeffaf bir malzemeden yapılarak çözülmüştür. Bu sensörlerin karşılıklı olabilmesi için ve sıvının da aralarından geçmesini sağlayacak şekilde baskılı devre kartı tasarımının gerçekleştirilmesi problemi ise, karşılıklı iki baskılı devre kartının bağlantı pinleri kullanılarak sabitlenmesiyle çözülmüştür.

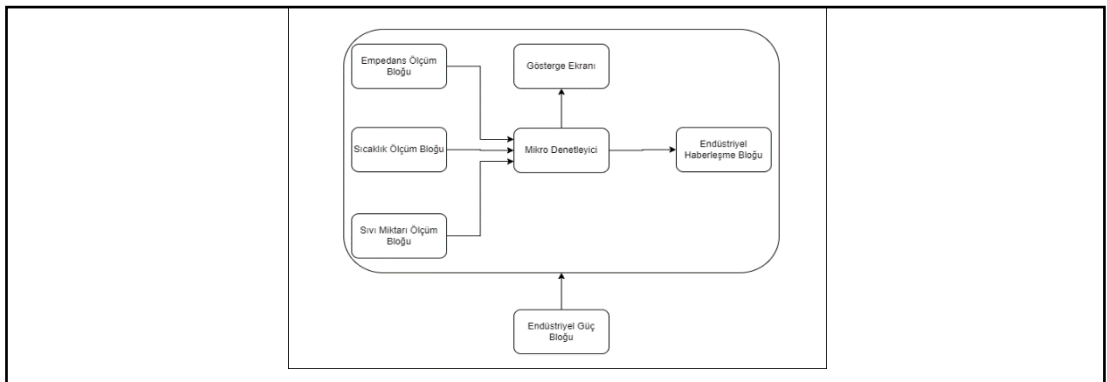
Ayrıca çalışmamızda renk sensörü kullanılarak sıvı içerisinde yabancı bir sıvı veya madde var ise tespit edilmektedir. Tasarımın diğer ayrıntıları ve konumlandırılma görselleri elektronik devre tasarımı kısmında bulunmaktadır.

4. ELEKTRONİK DEVRE TASARIMI VE ÖZELLİKLERİ

Elektronik kart tasarımında çalışmalarımızı gerçekleştirebilmek için analog çıkışları olan ve yüksek hızlı bir işlemci ihtiyacı bulunmaktadır. Ayrıca çalışmamızda iletkenlik, renk sensörü, haberleşme modülü gibi çevresel birimler olduğu için bu ihtiyaçları karşılayacak bir işlemci gerekmektedir. Bu sebeple STMicroelectronics firmasına ait STM32H723VET6 işlemcisi seçilmiştir. STMicroelectronics'e ait bu işlemcinin seçilmesinin bir başka sebebi ise işlemcinin fiyat olarak uygunluğu ve programlanmasının kolay olmasıdır.

4.1. Tasarım Özellikleri

Şekil 18'de tasarımımızın blok diyagramı bulunmaktadır. Cihazımız 24V ile çalışmaktadır ve endüstriye güç bloğu cihazımızın içerisinde bulunan diğer bölümler için gerekli voltaj değerlerini sağlamak için çeşitli regülatörlerden oluşmaktadır. Cihazımızda ölçümleri yapmak için ise empedans ölçüm bloğu, sıcaklık ölçüm bloğu ve sıvı miktarı ölçüm bloğu bulunmaktadır. Sıvı miktarını veya cihazdaki arıza bildirimlerini göstermek için gösterge ekranı, cihazın bir sunucu ya da veri toplama aygıtı ile haberleşmesini sağlamak içinde bir haberleşme bloğu bulunmaktadır. Ayrıca bu işlemlerin gerçekleştirilebilmesi için STMicroelectronics'e ait bir mikro denetleyici vardır.



Şekil 4.1: Elektronik tasarım blok diyagramı.

4.1.1. Mikro Denetleyici

STM32 serisi, ST Microelectronics firmasının ürettiği 32 bitlik mikrodenetleyici ailesidir. Ailenin bütün üyeleri ARM – Cortex – M serisi işlemci çekirdeklerine sahiptir. Projemizde ise bu ailenin alt gruplarından H7 serisi kullanılmıştır. STM32H7 serisi, STM32 serisini yüksek performanslı serilerindendir.

STM32H723VET6, Arm Cortex-M7 işlemci mimarisi kullanılarak tasarlanmıştır ve yüksek performanslı gömülü sistemlerde kullanılmak üzere tasarlanmıştır. STM32H723VET6, 550 MHz'lik bir işlemci hızına sahiptir ve çift çekirdekli olarak çalışır. Ayrıca, 1MB Flash bellek ve 512KB SRAM belleği bulunur. Bu mikrodenetleyici, çeşitli endüstriyel ve otomasyon uygulamalarında kullanılmak üzere tasarlanmıştır.

STM32H723VET6, bir dizi önemli özellik sunar. Bu özellikler arasında 16-bit ADC (analog-dijital dönüştürücü) ve 12-bit DAC (dijital-analog dönüştürücü) gibi yüksek kaliteli analog ölçümler yapabilme yeteneği, Ethernet, USB, ve UART gibi iletişim arabirimleri, ve güvenli bir sistem tasarımı için donanım şifrelemesi ve imza doğrulama işlevleri sayılabilir.

STM32H723VET6, STM32CubeMX yazılımı tarafından desteklenir ve birçok geliştirme platformuyla uyumludur. Bu mikrodenetleyici, endüstriyel otomasyon, medikal cihazlar, güç elektroniği ve akıllı bina uygulamaları gibi birçok alanda kullanılabilir.

Tablo 4.1: Mikroişlemci özellikleri.

Mikrodenetleyici	STM32F723VET6
Mimarisi	ARM Cortex-M7
Maksimum Frekans	216 MHz
Flaş Bellek	512 KB
SRAM Bellek	384 KB
DMA Kanalları	2x16 kanal
ADC Çözünürlüğü	12-bit
ADC Kanalları	24 kanal
DAC Çözünürlüğü	12-bit

Tablo 4.2: Devam.

USART Kanalları	8 kanal
DAC Kanalları	2 kanal
SPI Kanalları	6 kanal
I2C Kanalları	4 kanal
USB Versiyonu	USB 2.0
GPIO Pini Sayısı	169 pin
Çalışma Sıcaklığı	-40°C ile 85°C arası
Besleme Gerilimi	1.7V- 3.6V

4.1.2. Sıcaklık Sensörü

TMP117, yüksek hassasiyetli, düşük güç tüketimli bir dijital sıcaklık sensörüdür. TMP117, kalibrasyon olmadan -20 °C ile 50 °C sıcaklık aralığında 0,0078 °C çözünürlük ve $\pm 0,1$ °C'ye kadar doğrulukla 16 bitlik bir sıcaklık sonucu sağlar. TMP117, I2C ve SMBus™ uyumlu, programlanabilir uyarı işlevine sahip arayüze sahiptir ve cihaz, tek bir veri yolunda dört adede kadar cihazı destekleyebilir. Cihaz programlama için entegre EEPROM, genel kullanım için ek bir 48 bit bellekle içermektedir. TMP117, 1.7 V ile 5.5 V arasında çalışır ve tipik olarak 3.5 μ A tüketir. Bu düşük güç tüketimi sayesinde kendi kendine ısınmanın önüne geçerek doğruluğu yüksek seviyede tutabilmektedir.

Tablo 4.3: Sıcaklık sensörü özellikleri.

İletişim Arayüzü	I2C
Sıcaklık Ölçme Aralığı	-55°C ile 150°C arası
Hassasiyet	± 0.1 °C'ye kadar
Çözünürlük	16-bit
Güç Tüketimi	Düşük güç tüketimi, ultra düşük güç modları
Çalışma Gerilimi	1.8V ile 5.5V arası

4.1.3. Haberleşme Modülü

ESP32 modülleri, düşük güçlü sensör ağlarından ses gibi en zorlu görevlere kadar çok çeşitli uygulamaları hedefleyen iki güçlü, genel Wi-Fi, Bluetooth ve Bluetooth LE içeren bir mikrodenetleyicidir.

ESP32, 2 çekirdekli Tensilica LX6 mikrodenetleyiciye sahiptir ve hızı 240 MHz'e kadar çıkabilir. ESP32, entegre dahili flash bellek ile birlikte gelir ve ayrıca harici flash bellek ve SRAM desteği de sağlar. Modül, 34 adet programlanabilir GPIO pini ve çeşitli analog giriş/çıkış pinleri ile birlikte gelir. ESP32, bir USB bağlantı noktasına sahiptir ve bu port aracılığıyla programlama ve seri haberleşme yapılabilir. Modül, düşük güç tüketimi modları da dahil olmak üzere birçok özelliği destekler ve bu nedenle pil ile çalışan cihazlar için idealdir. ESP32, çeşitli anten seçenekleriyle birlikte sunulur ve farklı uygulama gereksinimlerine uyacak şekilde yapılandırılabilir.

ESP32 modülü, Wi-Fi ve Bluetooth özelliklerinin yanı sıra güçlü bir işlemci ve entegre dahili bellek gibi birçok özellik sağlar. Bu, modülü birçok farklı uygulama için uygun hale getirir ve prototipler yapmak veya nihai ürünler tasarlamak için çok kullanışlıdır.

Tablo 4.4: Haberleşme modülü özellikleri.

İşlemci	2 çekirdekli Tensilica LX6 mikro kontrolcü, 240 MHz'e kadar hız
ROM	448 KB
SRAM	520 KB
SPI Flash	16 Mbit
I/O	26
ADC/DAC	12-bit ADC, 8-bit DAC
Çalışma Voltajı	3.0 – 3.6 V
Çalışma Sıcaklığı	-40 – 105 °C
İletişim Arayüzleri	Wi-Fi 802.11 b/g/n, Bluetooth 4.2 (BLE), USB 2.0

4.1.4. LM358DR

LM358DR, çift kanallı genel amaçlı kullanıma uygun opamplardır. LM358DR ile LED alıcı devresinin tasarımı gerçekleştirilmiştir. Bu devre tasarımında LED alıcısından gelen değer ile belirlenen bir BIAS değeri ile karşılaştırarak bir çıkış değeri elde edilmesini sağlar. Bu sayede belirlenen eşik değerin altında bir değer geldiği durumlarda oluşabilecek kararsız durumlar engellenmektedir.

4.1.5. 555 Devresi

LMC555 entegresi kullandık LMC 555 Entegresi ile sıvının iletkenliğini ölçebilmek için devre tasarımı gerçekleştirdik. Bu devre tasarımında çıkış sinyali işlemcinin TIMER bacağına bağlanarak ölçümler gerçekleştirilmiştir.

4.1.6. FRAM

Sıvı ölçüm bilgilerinin hafıza da tutulması ve yedeklenmesi için konulmuştur. Haberleşmenin olmadığı veya koptuğu anlarda sonrasında verileri alabilmek için kullanılmaktadır.

4.1.7. Seven Segment Display

Sıvı ölçüm miktarının, ölçüm sırasında oluşan hatalar ve ölçümler ile belirlenen uyarıların gösterilmesi için cihaz üzerinde seven segment display eklenmiştir.

4.1.8. Diğer Devre Blokları ve Komponentler

Sıvı ölçümünü gerçekleştirmek için Infrared LED ve fotodiyotlar kullanılmıştır. SFH4545 infrared LED'leri ile gönderilen ışığın karşı tarafa yerleştirilen VEMD2020X01 fotodiyotları ile algılayacak şekilde tasarım gerçekleştirilmiştir. Bu LEDler ve fotodiyotlardan 10'ar adet baskı devre kartı üzerine karşılıklı olarak yerleştirilmiştir. Yerleşimler birbirini en az etkileyecek şekilde ve boru çapına uygun olarak yerleştirilmiştir. Ayrıca dar açılı LED'ler seçilerek bunun önüne geçilmesi sağlanmıştır. SFH4545 infrared LED'leri yüksek verimliliğe sahip ve yüksek oranda

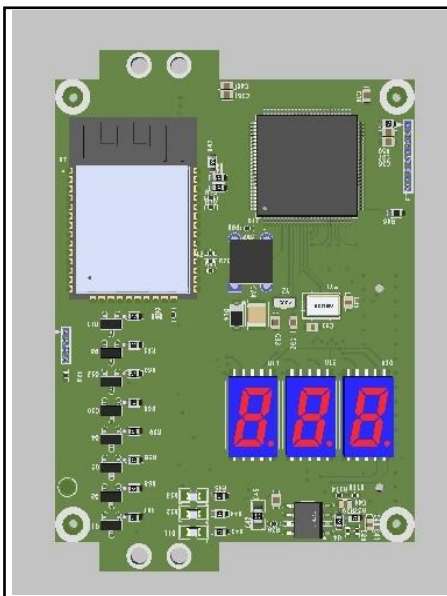
güç vermektedir. Bu infrared LED’lerin yüksek verimliliğe sahip olmasının bir diğer avantajı ise az ısınmasıdır. Çünkü kart üzerindeki ısının sütun sıcaklığını ölçebilmemiz için minimum seviyede tutulması gerekmektedir.

Ayrıca kartımız üzerinde bir RGB LED ve RGB fotodiyot bulunmaktadır. RGB kullanmamızın en büyük nedeni farklı sıvılardaki maddeleri algılayabilmek için farklı ışık renklerini kullanmamız gerektiğindendir. Ayrıca RGB LED’lerin renk karışımı ile oluştuğu için frekans değerleri tek renk LED’lerden farklıdır ve bu bizim tasarımıımızda doğru algılamak için bir avantajdır.

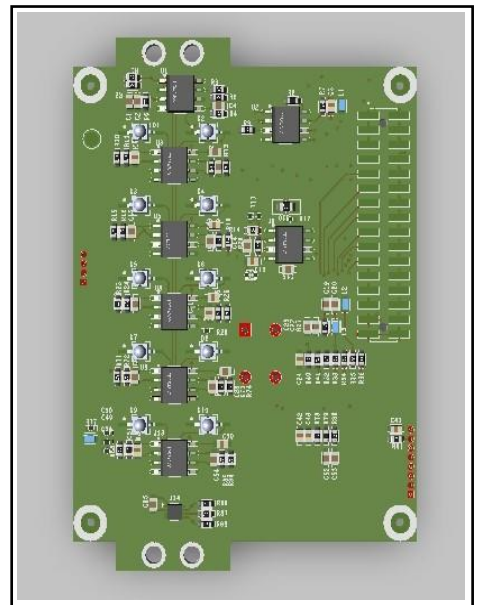
SFH4545 infrared LED’i 950 nm dalga boyuna ve 10 derece yayılım açısına sahiptir. VEMD2020X01 750nm ile 1050 nm dalga boyuna sahip ışınları algılamaktadır. 830 nm ile 950 nm dalga boyu arasında gün ışığı engelleme filtresi, 30 derece bir açısı var.

4.2. Elektronik Devre Tasarımı

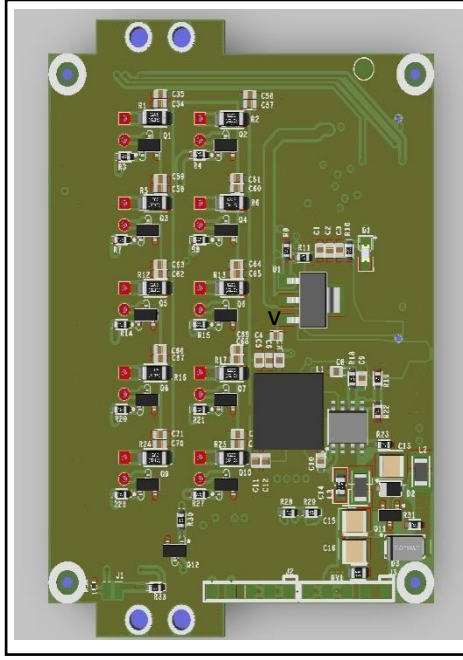
Aşağıda sıvı ölçer tasarımıımızı oluşturan karşılıklı iki elektronik kartın görüntüleri bulunmaktadır. Elektronik devre kartları dörder katmanlıdır. Üst kart sıvı ölçerimizin ana kartını oluşturmaktadır ve içerisinde işlemci, 555 devresi, ESP32 modülü ve infrared alıcılar bulundurmaktadır. Alt kart ise tasarımıımızda güç bloğunu barındırmaktadır. Ayrıca sıvı ölçümünde kullandığımız infrared LED’lerimizde bu kartımızda bulunmaktadır.



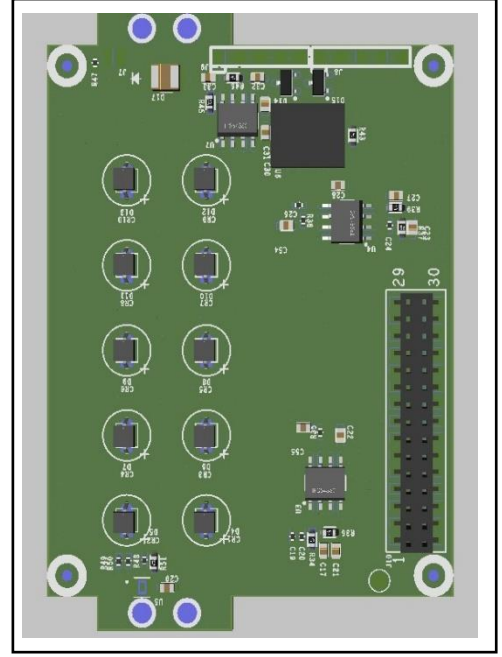
Şekil 4.2: Üst kartın ön yüzü



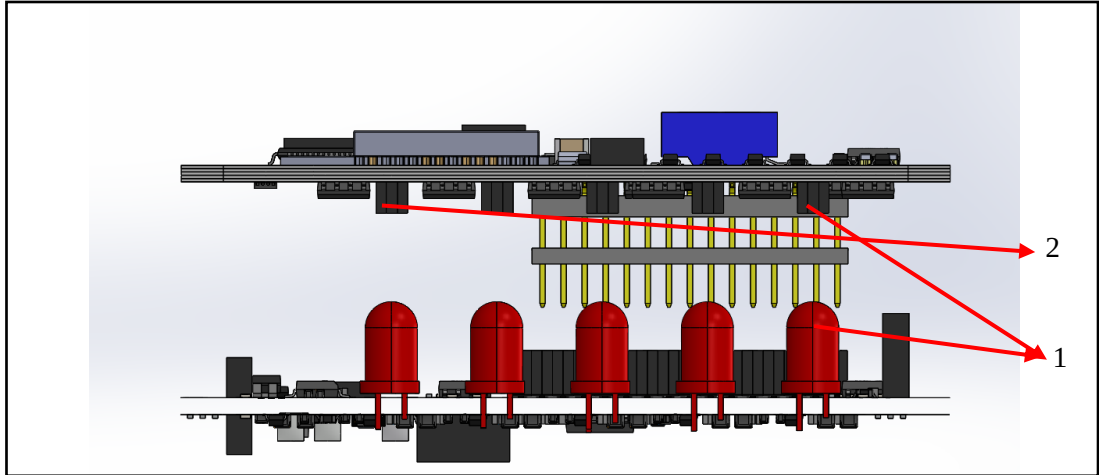
Şekil 4.3: Üst kartın arka yüzü



Şekil 4.4: Alt kartın ön yüzü.



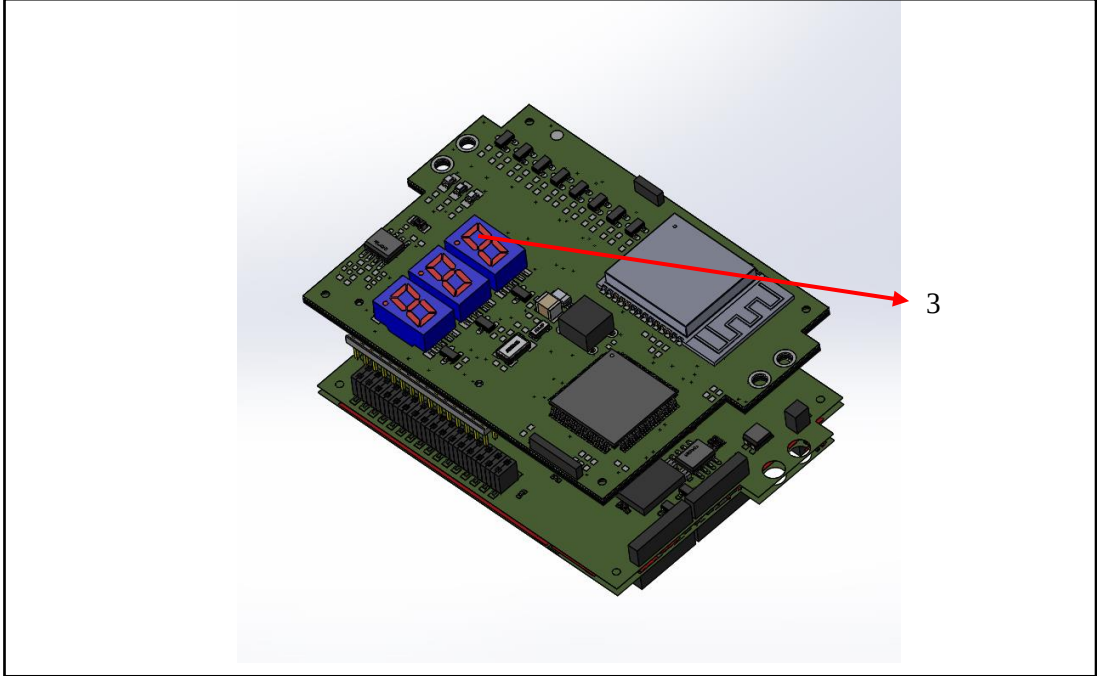
Şekil 4.5: Alt kartın arka yüzü.



Şekil 4.6: Infrared led ve alıcıların konumu.

- 1- Karşılıklıklıklı şekilde bulunan infrared alıcı ve verici sensörler, sütun akış hızını ve miktarını bulmamıza olanak sağlamaktadır. Bu sensörler olmadan sütun miktarı hakkında bilgi sahibi olamayız.

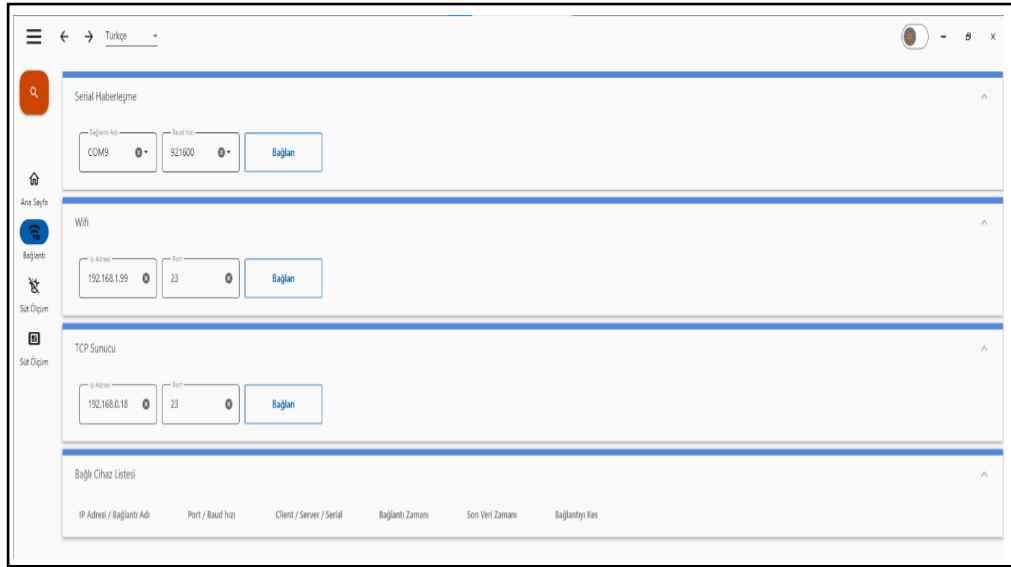
- 2- Sütteki kan takibi renk sensörü ile yapılmaktadır. Bu kısım olmadan sütteki kan takibi yapılamayacaktır



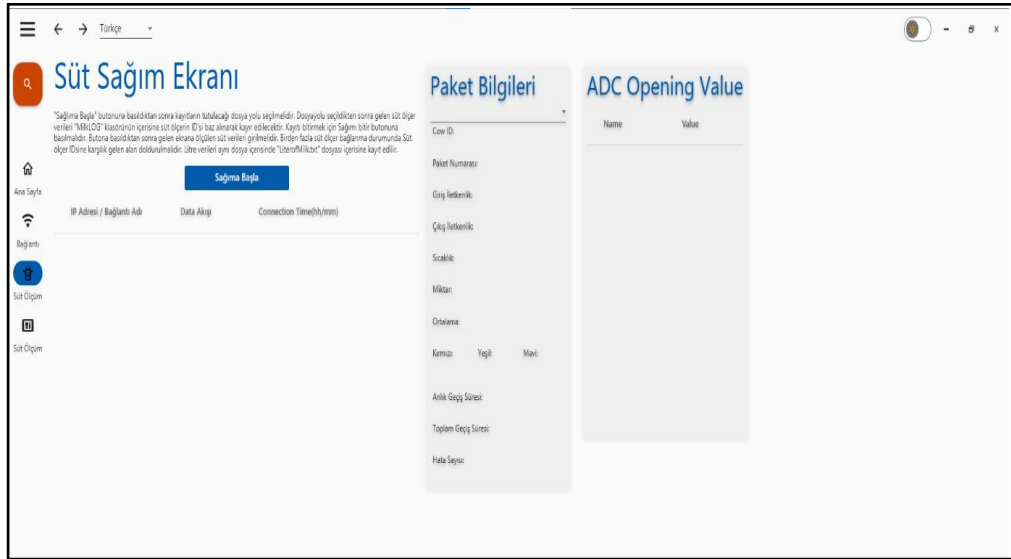
Şekil 4.7: Seven segment display.

- 3- Sütün miktarını gösteren kısımdır. Bu kısım, rakam göstergesi (7 segment display)dir.

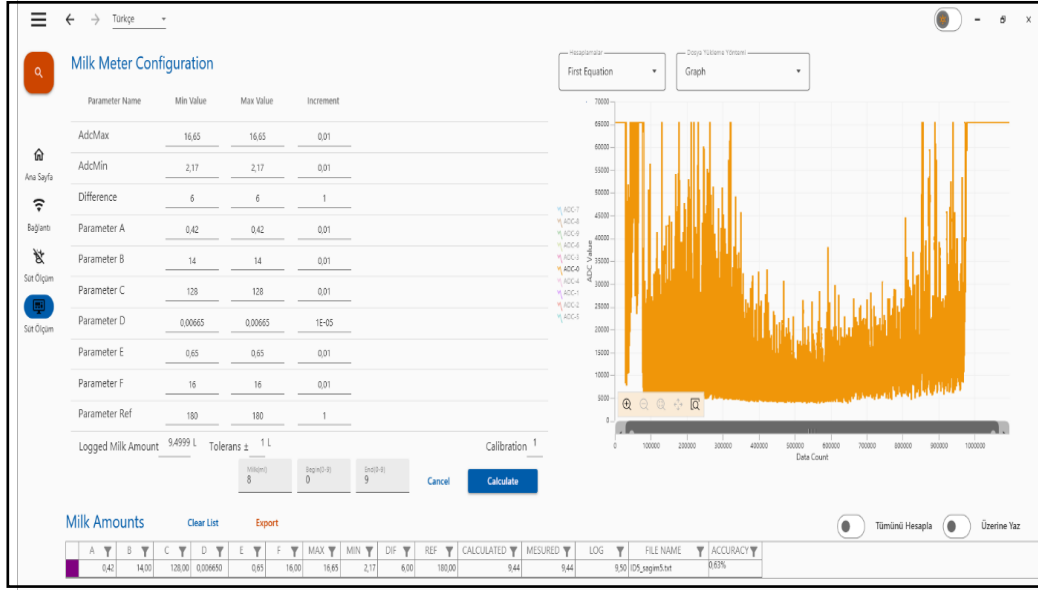
Aşağıda sıvı ölçümü esnasında verileri anlık olarak görebildiğimiz ve daha önceki ölçüm kayıtlarına ulaşabildiğimiz bir arayüzümüz bulunmaktadır. Cihazımız bu arayüz ile haberleşerek arayüze ölçüm değerleri dışında ölçümle ilgili diğer bilgileri de göndererek bir eşleştirme yapmaktadır. Bu sayede ölçümlerde birden fazla cihaz kullanıldığında hangi cihaz ile ölçüm yapıldığı bilinebilmektedir. Ayrıca arayüzümüzde eski verilerle karşılaştırma, verileri inceleme ve bu veriler ile analizler gerçekleştirilebilmektedir.



Şekil 4.8: Arayüz programı bağlantı sekmesi.

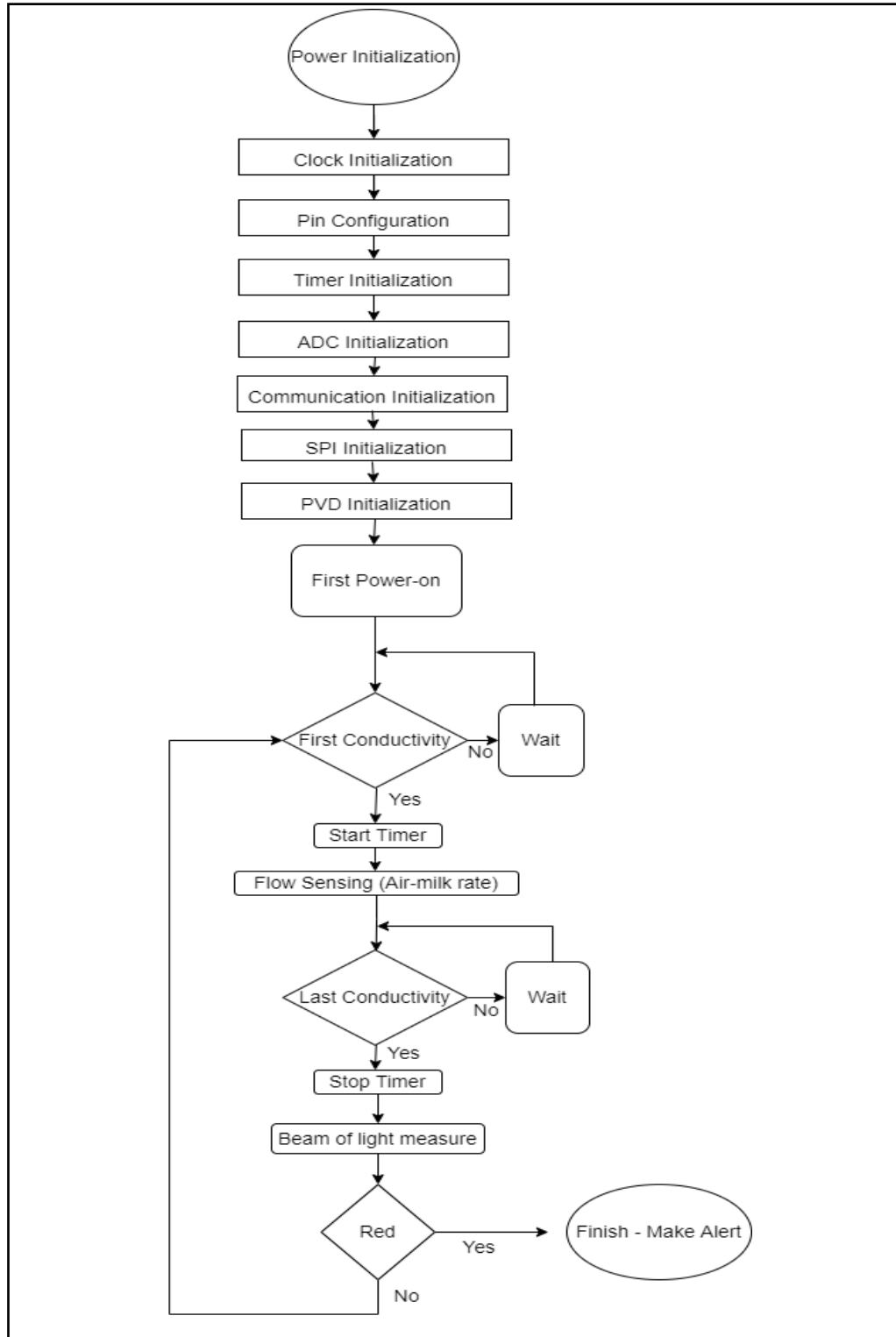


Şekil 4.9: Sıvı ölçüm takip ekranı.



Şekil 4.10: Geçmişe yönelik ölçüm görüntüleme ve analiz ekranı.

5. YAZILIM ALGORİTMASI



Şekil 5.1: Yazılım algoritması blok diyagramı.

Yukarıdaki şekilde yazılım algoritmasının blok diyagramı bulunmaktadır. Yazılım algoritmasında ekler kısmında verilen while döngüsü içerisindeki fonksiyonlar sırayla ve sürekli çağırılarak çalıştırılmaktadır. Bu fonksiyonlar sırasıyla, sıvı miktarı ölçme fonksiyonu, buton işlemleri fonksiyonu, sıcaklık ölçüm fonksiyonu, yabancı madde algılama fonksiyonu, ekrana miktar ve uyarıları yazdırma fonksiyonu, wifi veya bluetooth ile haberleşme fonksiyonlarıdır. Yazılım algoritması dahilindeki tüm fonksiyonların kod parçacıkları ekler kısmında verilmiştir.

5.1. Sıvı Varlığı Fonksiyonu

Sıvı miktarı ölçümünde ilk önce sıvı akış hızı bulunmaktadır. Bu yüzden ilk önce girişte sıvı varlığının tespit edilmesi gerekmektedir. Sıvı varlığı ADC verileri üzerinden belirlenir. Eğer sıvı yok ise kod tüm veri ve sayaçları sıfırlar. Eğer sıvı var ise TIMER fonksiyonu içerisinde bulunan sıvı miktarı hesaplaması kullanılır.

5.2. Sıvı Akışı Hız Fonksiyonu

Sıvı akış hızını bulurken cihaz üzerindeki pimler ve infrared ledler aracılığı ile hesaplama yapılır. Sıvı akış hızına ve akışkan şekline bağlı olarak bazı farklı durumlar gerçekleşmektedir.

- 1- Girişteki ilk led ve çıkıştaki son ledten ölçüm aldığımız durum (40 mm)
- 2- Girişte pime değdiği ve çıkışta pime değdiği durum (67 mm)
- 3- Girişte pime çıkışta lede değdiği durum (51 mm)
- 4- Girişte lede çıkışta pime deddiği durum (56mm)

Sıvı akış hızını hesaplarken yukarıdaki dört farklı durum göz önüne alınarak hesaplama yapılır. Sıvı akış hızı hesaplamasında ilk önce girişteki sıvının pim veya ledten hangisi ile temas ettiği belirlenir. Sonrasında girişte sıvı olduğu ancak çıkışta sıvı olmadığı tespit edilerek zaman hesaplaması için arka planda çalışan TIMER fonksiyonuna sıvı geldiği bilgisi gönderilir ve zaman sayacı başlatılır.

Yazılım sürekli bir while döngüsü içerisinde döndürülerek çıkışta ledde ya da çıkış iletkenlik piminde sıvı teması olana kadar döngü çalışmaya devam eder. Çıkış piminde ya da çıkışta bulunan son ledlerde sıvı olduğu tespit edildiğinde çıkışa sıvı

geldiği bilgisi TIMER fonksiyonuna gönderilerek zaman sayacı durdurulur ve sayaçta elde edilen süre örnek sayısına bölünür. Örnek sayısı kodumuzda 10.000 adettir.

Hız- mesafe ($x=v*t$) formülü uygulanarak, TIMER fonksiyonunda bulduğumuz süre ile giriş ve çıkışta temas ettiği mesafeler bilindiği için formülde mesafe metreye çevrilerek yazılır ve bu işlem ile sıvının ortalama akış hızı bulunur. Yukarıda dört farklı durum içinde, durumların yanında milimetre olarak mesafe bilgileri yazılmıştır. Bu dört farklı durum matematiksel ifadeler 3, 4, 5 ve 6 numaralı denklemlerle aşağıda gösterilmiştir.

$$x_1 = v \times t \quad (5.1)$$

$$x_2 = v \times t \quad (5.2)$$

$$x_3 = v \times t \quad (5.3)$$

$$x_4 = v \times t \quad (5.4)$$

5.3. Sıvı Miktarı Hesaplaması

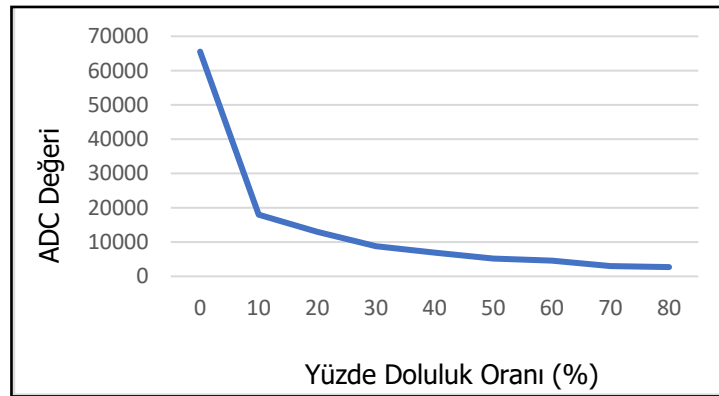
ADC ortalama fonksiyonundan yararlanarak sıvı ağırlığı hesaplanır. Ortalama ADC değerleri hesaplanır. Bulunan ADC yüzdeleri toplanarak ADC sayısına bölünür. Bu sayede tüm ADC değerlerinin ortalaması bulunur ve bu ADC ortalaması ile çıkan yüzde bize sıvı hacminin yüzdesini vermektedir. Bu hacim yüzdesi boru kesitinin hacmi ile orantılanarak o anda borudan geçen sıvı miktarı bulunur ve bu şekilde boru kesiti üzerinden sıvı parçacığının ağırlığı hesaplanır ve üst üste eklenerek sıvı geçişi tamamlandığında tüm geçişin ağırlık miktarı hesaplanmış olur.

5.3.1. ADC Ortalama Hesaplaması

ADC değerlerinin ortalama hesabı yapılması ve yüzde doluluk oranlarının belirlenmesi için bir fonksiyon elde ettik. Bu fonksiyonu elde edebilmek için cihaz üzerindeki sıvı borusunun kesiti alınarak içerisine 1,2,3,4,5,6,7 ve 8 ml sıvılar konulmuştur. Bu sıvıların yüzde doluluk oranları belirlenerek bir ADC değerlerine bakılmıştır ve aşağıdaki veriler ve grafik elde edilmiştir.

Tablo 5.1: Doluluk oranı adc değerleri.

%	ml	ADC_VAL[7]	ADC_VAL[8]
0	0	65535	65535
12.5	1	18000	20700
25	2	13000	13000
37.5	3	8800	8800
50	4	6900	7000
62.5	5	5200	5200
75	6	4600	4600
87.5	7	3000	2900
100	8	2700	2600



Şekil 5.2: Doluluk oranı adc değerleri grafiği.

Bu grafiğe göre hesaplama fonksiyonumuzu oluşturmak için tek bir formül bulamadığımız için 2.170 ile 16.650 değerleri arası için bir formül, 16.650 ile 60.000 değerleri arasında ise başka bir formül kullanılmıştır.

Gelen değer 2170 ile 16650 arasında ise;

$$0,42 \times (x^2 - 14) \times (x + 128) \quad (5.5)$$

16650 ile 60000 arasında ise;

$$0,00665 \times (x^2 - 0,65) \times (x + 16) \quad (5.6)$$

Bu denklemlerde; x değeri okunan ADC değeridir.

Okunan ADC değeri 0 ile 2170 arasında ise sıvı geçişi %100, 60000 ile 65000 arasında ise sıvı geçişi %0 kabul edilmektedir.

İşlem kolaylığı içinde ADC değerleri 1000 ile bölünerek hesaplamalara dahil edilmiştir ve fonksiyonun sabitleri bu değerlere göre belirlenmiştir.

5.4. Buton Fonksiyonu

Buton cihazın göstergelerini sıfırlamak veya cihaza reset atabilmek için konulmuştur. Butona basıldığında sayaç süreyi saymaya başlar. Eğer 3 saniye boyunca butona basılırsa cihaz ekrandaki göstergeleri sıfırlar. Butona 7 saniye boyunca basıldığında ise cihaz resetlenir.

5.5. Sıcaklık Ölçüm Fonksiyonu

Cihaza elektrik geldiği anda sıcaklık fonksiyonu kendisini başlatmaktadır. Sonrasında her beş saniye de bir defa sıcaklık değerini vermektedir. Bu fonksiyon sıcaklık sınırları belirlenerek bu sınırların dışına çıkıldığında ekran üzerinden alarm vermektedir.

5.6. Yabancı Madde Tespit Fonksiyonu

RGB ledi yakmak için ilk önce yeşil ve mavi renkleri aktif edilir. Sonrasında ise tam beyaz rengi elde edebilmek için kırmızı rengi %50 duty cycle ile sürülmesi gerekmektedir. RGB renk ile farklı madde tespiti için rgb renginin tam beyaz yanması gerekmektedir. Yabancı madde tespiti içinde renk okuma sensörü fonksiyonu çalıştırılır.

5.6.1. Sensör Okuma Fonksiyonu

RGB alıcısı ile rgb ledden gelen değerler okunur. 0 ile 65535 arasında bu değerler değişmektedir. Eğer beyaz renk dışında herhangi bir rengin baskın olması durumunda fonksiyon yabancı madde olabilir uyarısı vermektedir. Baskın olan rengin değeri ve eğer çalışılan yerde belirlenmiş bir yabancı madde rengi var ise bu maddenin

de tespiti hakkında yorum yapılabilir. Örneğin; sağım hanelerdeki süt sağım esnasında hayvan memelerinden kan gelme durumunda RGB alıcısı 10.000 değerinden yüksek bir kırmızı değeri gelirse bu sağım esnasında süte kan karışmış olabilir uyarısı verilebilir.

5.7. Display Ekran Fonksiyonu

Projemizde cihaz üzerinde uyarıları ve miktarı göstermek için yedi segmentli ekranlar kullanılmıştır. Yedi segmentli ekranlara süt miktarını yazabilmek için aşağıdaki fonksiyonlar kullanılmıştır. Üç adet yedi segmentli ekran bulunmaktadır. Bu yedi segmentli ekranlar, sayaç ile sürekli aç kapat yapılarak sırayla sürülmektedir.

Yazılımımızda bulunan “Seg_Value_Calculate” fonksiyonu ilk olarak ekrana ne yazdırılacağını kontrol etmektedir. Eğer sayı yazdırılacaksa hesaplama yaparak ondalık ve kusüratın ekranda nasıl gösterileceğini hesaplar. “Seg_Goster” fonksiyonu ise hem yazılacak sayıyı hem de bu sayının nasıl yazılacağını belirler. Bu fonksiyonların kodları ekler kısmındadır.

5.7.1. “Seg Value Calculate” Fonksiyonu

Çalışmamızda tasarladığımız cihazda üç adet yedi segmentli ekran bulunmaktadır. Yani değer olarak maksimum üç hane gösterme imkânı bulunmaktadır. Bunun için değer net anlaşılması ve gösterilmesi gerekmektedir. Bu durumda mililitre ve litre cinsinden bu değerleri göstermek için bu hesaplama fonksiyonuna ihtiyaç duyulmaktadır.

Bu fonksiyonda eğer değerimiz 999 mililitreden düşükse ekrana direkt yazıyor. Eğer 999’dan büyük ise sayıyı 10 ile bölünüyor ve sayıya 999’dan “büyük mü?” diye tekrar kontrol ediliyor. Bu işlemi sayı 999’dan küçük olana kadar tekrarlanıyor. Bu işlem kaç defa yapıldı ise o kadar ondalık olarak nokta kaydırılıyor. Örneğin 26538 değerindeki sıvıyı göstermek için değer iki defa 10’a bölünmesi gerekmektedir ve değeri 265,38 değeri bulunur. Bu sayı 999’dan küçüktür ve işlemi sonlandırılır. İki defa bölündüğü için nokta iki defa sağa kaydırılır ve değer 26.538 litre olarak bulunur. Bu değer yedi segmentli ekranda 26.5 litre olarak görünecektir.

5.7.2. “GosterOutput” Fonksiyonu

Bu fonksiyon yedi segmentli ekranda harf veya rakamın nasıl yazılacağını belirler. İlk olarak fonksiyonumuzda yedi segmentli ekranda bulunan ledlerin sürülebilmesi için ayarlama yapılmıştır. Sonrasında ise kullanılan her harf ve rakamın yazımı için yedi segmentli ekranda hangi ledlerin yanacağı ile ilgili ayrı ayrı çıkış olayları kurgulanmıştır. Tüm çıkış olaylarının yanında hangi çıkış olayının hangi rakam veya harfi yazacağı yorum satırı olarak belirtilmektedir.

5.7.3. “Seg_Goster” Fonksiyonu

“Gosterge output” fonksiyonundan gelen bilgiye göre hangi yedi segmentli ekranda rakam veya harf yazılacaksa sadece yazılacak olan yedi segmentli ekranın enerjilendirilmesini sağlar. Diğer ekranlardaki değeri korumak için onların enerjilerini keser. Bu işlem çok hızlı gerçekleştirilerek insan gözüyle algılanmayacak şekilde kurgulanmıştır ve kullanım esnasında herhangi bir problem yaşanmamaktadır.

5.8. Bluetooth ve Wifi Haberleşme Fonksiyonu

Bu çalışma kapsamında tasarlanan cihazın wifi veya bluetooth üzerinden haberleşmesini sağlayan ESP32 modülünü fonksiyonları bulunmaktadır. Cihazın haberleşme tercihiye göre aşağıdaki kodda bluetooth fonksiyonu ya da wifi fonksiyonu aktifleştirilmelidir.

5.8.1. WIFI_AT SEND Fonksiyonu

Bu fonksiyon işlemciden ESP modülüne gönderilen AT komutlarını içermektedir. Bu AT komutları ile, ESP modülüne hangi modem ve kablosuz haberleşme cihazına bağlanacağı bilgisi verilir. Bu bilgiler içerisinde ayrıca modülün IP adresi ve modem IP adresi de bulunmaktadır.

5.8.2. “WIFI AT RESPONSE” Fonksiyonu

Bu fonksiyonda ESP modülünden işlemciye AT komutları ile dönüş yapılmaktadır. İlk olarak ESP modülü işlemciye “ready” yazısı göndererek hazır olduğunu bildirir. Ayrıca bu fonksiyonda, wifi bağlantı durumu sorgulaması, başarılı veri iletimi sorgulaması, server durumu, baudrate belirlenmesi, cihaz ile wifi arasındaki bağlantı sorgulaması veya bağlandıktan sonra bağlantı durumundaki değişikliğin takibi gibi sorgulamalar ve ayrıca gönderilecek olan veri boyutunun bilgisi gibi komutları bulunmaktadır.

5.8.3. Bluetooth Haberleşme Fonksiyonu

Bu fonksiyonda işlemciden ESP modülüne gönderilen Bluetooth AT komutları bulunmaktadır. Bu AT komutları ile, ESP modülünde bluetooth ismi belirlenmektedir ve bluetooth yayını açma ve kapama işlemi yapılmaktadır. Gönderilen verinin boyutunu ayarlama komutu da bulunmaktadır. Bluetooth ile maksimum 22 byte veri gönderilebildiği için gönderilecek verinin boyutunun sınırlandırılması gerekmektedir.

5.8.4. BLE AT RESPONSE Fonksiyonu

Bu fonksiyonda ESP modülünden işlemciye AT komutları ile dönüş yapılmaktadır. Bu fonksiyonda, bluetooth bağlantı durumu sorgulaması, başarılı veri iletimi sorgulaması, cihaz ile diğer cihaz arasında başarılı bir bluetooth bağlantısı sorgusu veya bağlandıktan sonra bağlantı durumu sorgulaması ve ayrıca gönderilecek olan verinin boyutunun bilgisi gibi komutları içermektedir.

5.8.5. BLE Transmit Fonksiyonu

Bu fonksiyon ile bluetooth komutları diğer bağlantı yapılan cihaza gönderilmektedir.

5.8.6. WIFI Transmit Fonksiyonu.

Bu fonksiyon ile wifi komutları sunucuya gönderilmektedir.

5.8.7. Callback Fonksiyonu

Bluetooth ve wifi üzerinden bağlantı kurulan cihazdan gelen verileri toplayan kesme fonksiyondur.

5.9. İletkenlik

İletkenlik ölçümleri cihaz üzerinde bulunan pimler aracılığı ile yapılmaktadır. Cihaz üzerinde her iki baskılı devre kartının giriş ve çıkışında 2 adet olmak üzere toplam sekiz adet pim bağlantı noktası bulunmaktadır. Pim bağlantı noktaları Şekil 19, 20, 21 ve 22’de bulunan baskılı devre kartı resimlerinde görünmektedir.

İletkenlik ölçümünde sıvı değiştiği anda devreden gelen sinyal ile frekans değeri ölçülmektedir. Sıvının cinsine ve içerisindeki mineral, bakteri, somatik hücre vb. miktarına göre bu değerler değişmektedir. Burada iletkenlik değerine bakılarak daha önceden bilinen ortalama bir frekans değerinin üzerinde veya altında ise sistem uyarı vermektedir.

İletkenlik hesabı; gelen kare dalga sinyalinin yükselen kenarı geldiğinde, işlemci kendi içerisinde TIMER fonksiyonunu başlatmaktadır. Gelen ikinci yükselen kenarda TIMER fonksiyonu durdurulur. Bu iki değer arasındaki fark alınarak süre bulunur. Buradan zaman frekansa çevrilerek bir değer bulunur. Sonrasında işlemci kendi hızına bağlı olarak bir referans clock değeri belirler ve referans clock değeri bulunan değere bölünerek iletkenlik frekansını hesaplanır.

$$\text{İletkenlik Frekansı} = \frac{\text{Referans Clock}}{\text{TIMER}} \quad (5.7)$$

6. KATI MODEL TASARIMI

Bu çalışmanın başlangıcında prototip aşamasında testlerimizi gerçekleştirmek üzere tasarım 3D printer cihazında üretime uygun olarak yapılmıştır ve endüstriyel testlerden ziyade geliştirme test aşamalarına uygundur. Daha sonrasında bu tasarımlarda karşılaşılan problemlerle katı model tasarımı geliştirilmiş ve endüstriyel kullanıma uygun bir kutu tasarımı gerçekleştirilmiştir.

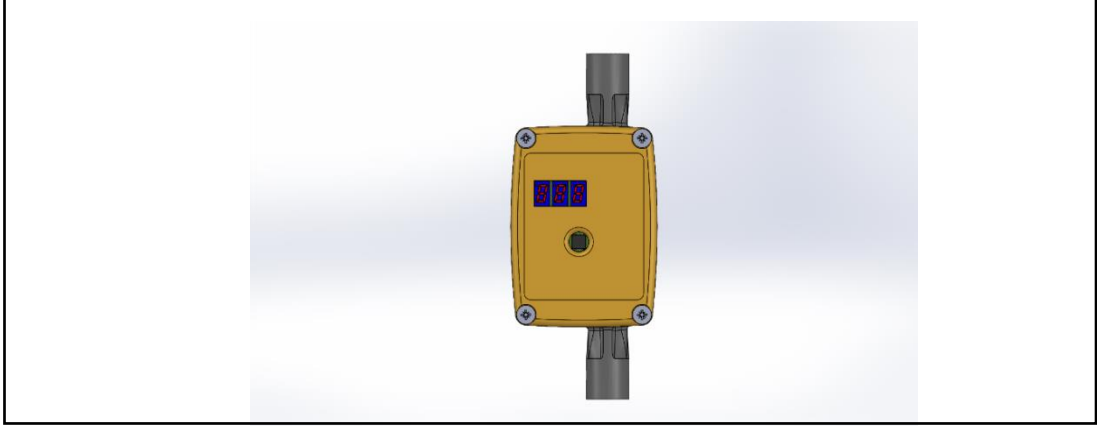
Tasarımımızdaki en önemli nokta cihazda içerisinden geçecek boru kısmının şeffaf olması ve hacminin piyasadaki dolum tesisleri ve sağım haneler gibi endüstriyel tesislerdeki sıvı akış borularına uygun olmasıdır. Boru bağlantı noktalarının da sert plastik parçaya takma ve çıkartma işlemleri gibi işlemlerde zarar görmemesi için destekli ve sert malzemeden yapılması gerekmektedir.

İlk testler için kutu tasarımında üç boyutlu yazıcı malzemesi olarak Polytic Acid (PLA) filament kullanılmıştır. Bu aşamada PLA tercih etme sebebimizin en önemli sebebi organik malzemelerden üretilmesi ve insan sağlığına zararının olmamasıdır. Ayrıca diğer filament çeşitlerine göre yazıcıda basımı daha kolay ve darbelere karşı dayanıklı olmasıdır. Ancak malzeme esnek bir yapıya sahip olduğu için sağım hanelerde sütün basınçla gelmesinden dolayı zamanla sütün geçtiği boruda çatlamalar gerçekleşmiştir. Bu nedenden dolayı sütün geçtiği boruda seri üretimde farklı bir malzeme kullanılmıştır.

Sütün geçtiği borunun malzemesi polikarbon olarak belirlenmiştir. Aynı şekilde bu malzemeyi seçme sebebimizin en başında insan sağlığına uygun olmasıdır. Çünkü bu malzemeler süt ile direkt temas ettikleri için insan sağlığına uygun olması gerekmektedir. Ayrıca bu malzeme sertlik açısından bizim ihtiyacımızı karşılamaktadır. Polikarbon malzemenin şeffaflık ve ışık geçirgenliği açısından da cam ile çok yakın değerlerde olması bizim ölçüm doğruluğumuzun yüksek olması için bir başka önemli husustur. Ayrıca bu malzeme sıvı geçişi esnasında sıvıların boru üzerinde kalmaması ve bir sıvı lekesi bırakmamasıdır. Bu sayede her ölçüm aynı hassasiyet ile yapılabilmektedir.

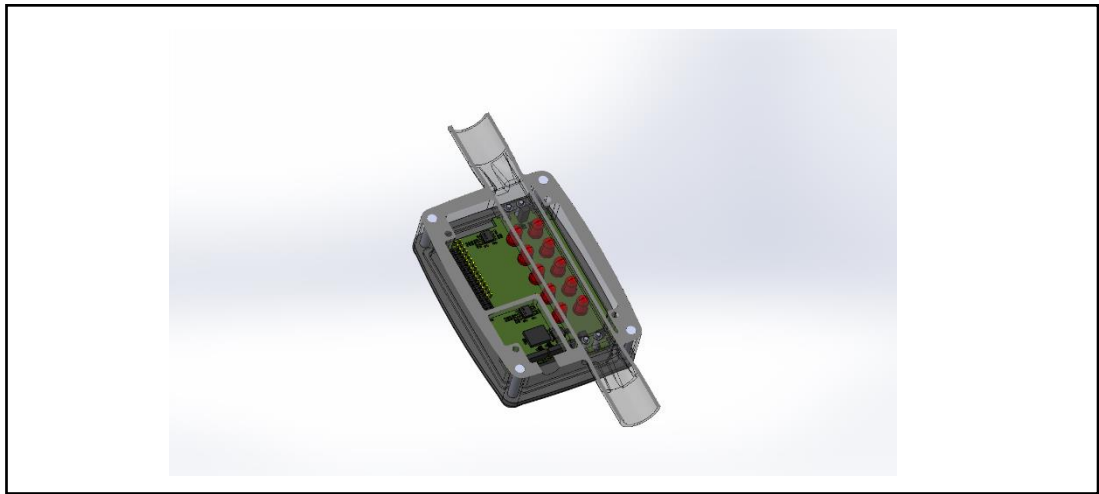
Tasarımımızda dış kaplama ise %15 cam elyaf takviyeli polibütilen tereftalat (PBT) malzeme kullandık. Bu malzemenin tercih edilme sebebi ise polikarbon malzemeye göre daha dayanıklı olması ve bu sayede dışarıdan gelebilecek olan darbelere karşı cihazı daha korunaklı bir hale getirmesidir. Polikarbon malzeme sert

bir malzeme olduđu için polibütilen tereftalat malzemeye göre kırılma ihtimali daha yüksek olacaktır. Bir başka sebepte infrared ledler ile sıvı miktar ölçümü yapıldığı için dış ortamın ışığından cihazın iç kısmını ayırt etmemiz gerekiyordu. Polikarbon malzeme şeffaf olduđu için polibütilen tereftalat kullandık.



Şekil 6.1: Sıvı ölçer tasarımı.

Şekil 29’da cihazımızın prototip olarak katı model tasarımı görünmektedir. Boru çıkıntısı ile cihaz arasında ekstra destekler konularak cihazımızın plastik akış boruları ile montajı esnasında kırılma veya çatlama gibi problemlerin önüne geçilmesi planlanmıştır.



Şekil 6.2: Sıvı ölçer cihazın bir kesiti.

Şekil 30’da cihazımızın bir kesiti bulunmaktadır. Bu kesitte iki kart arasında geçen şeffaf boru görünmektedir. Borunun şeffaf olması ve tam olarak led konumlarının borunun kenarlarına denk gelmesi ölçüm doğruluğunu arttırabilmek için gelen sıvıya karşı dirençli olması ve cihaza sıvı temasını engellemek için malzeme olarak polikarbon tercih edilmiştir.

7. TEST VE SONUÇLAR

7.1. Sağım Simülasyonu Test Düzeneği

Cihazımız birçok çeşit farklı sıvının miktarını ölçebilmekte ve içerisindeki yabancı maddeyi tespit edebilmektedir. Aşağıda masaüstündeki ölçüm testlerimizde su, süt ve yağ ölçümlerini gerçekleştirerek yazılım ve algoritma geliştirmelerini farklı sıvılar için tamamladık. Tasarımımızın tam doğruluğunu ve yüksek miktartlı ölçümlerdeki hassasiyetini test edebilmek ve bu oranı düşürmek için süt sağım simülasyonu kurduk. Süt sağım simülasyonunun kurmak ve bu sağım simülasyonu üzerinde çalışma istememizin ana sebebi diğer sistemlere göre sağım simülasyonunun kurulumunun kolaylığı ve ilk saha testlerimizi gerçekleştirebileceğimiz bir süt üretim çiftliğinin bulunmasıdır.

Süt sağım simülasyonun da bir kompresör ile aşağıda gösterilen süt tankına yaptığımız silikon memelerden süt sağımını gerçekleştiriyoruz. İnek memelerinden sütün kesik kesik gelmesinden dolayı sistemimizin düzgün çalışabilmesi ve süt miktarını doğru ölçebilmesi için sağım sisteminde sütün geçtiği borunun cihaz girmeden önce ve cihazdan çıktıktan sonra S şeklinde montajı yapılmıştır. Bu sayede süt S borusunda birikerek borudan sağım basıncının da etkisiyle mümkün olduğunca dolu geçmesi sağlanmaktadır.



Şekil 7.1: Sıvı ölçer cihazın ilk prototipi.



Şekil 7.2: Sıvı ölçer cihaz masaüstü test düzeneği.



Şekil 7.3: Test düzeneğine prototip montajı.



Şekil 7.4: Sağmal inek simülasyonu.



Şekil 7.5: Güğüme silikon meme montajı.

7.2. Sağım Simülasyonu Test Çalışmaları

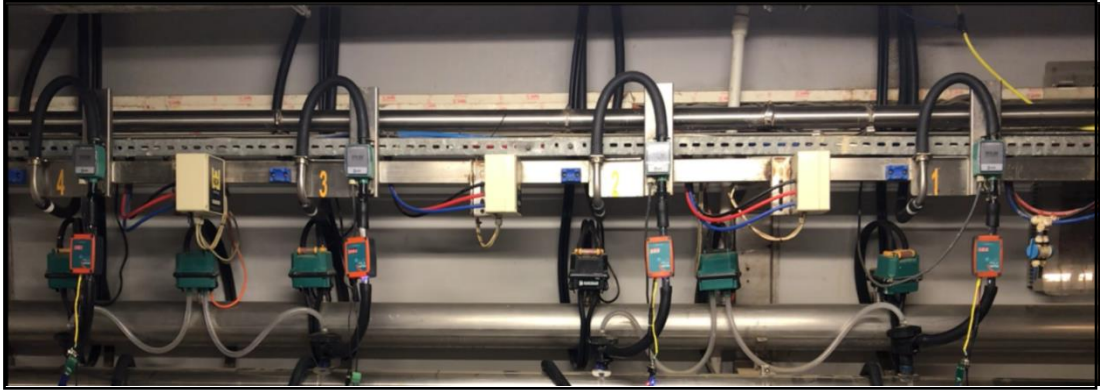
Sıvı ölçer testlerimizin süt sağım simülasyonu ile testleri gerçekleştirilmiştir. Testlerimizi süt tankından 5, 10, 20, 30 ve 50 litre olmak üzere farklı süt miktarlarında gerçekleştirdik. Sağmal ineklerin süt verimliliği ırkına, beslenmesine, bakımına, sağılma sıklığına ve sağılma yöntemi gibi faktörlere bağlıdır. Bu faktörlere bağlı olarak bir sağmal ineğin bir sağımda 50 litreye kadar süt vermesi mümkündür. Bizde testlerimizde bunu göz önünde bulundurarak çalışmalarımızı gerçekleştirdik.

Yapılan test çalışmalarında olumlu sonuçlar aldıktan sonra saha testleri için Çanakkale’de bulunan bir süt çiftliğine gittik ve orada saha testlerini gerçekleştirdik.

7.3. Sağım hane Testleri

7.3.1. Test Ortamı

Sağım hane testleri Çanakkale’de profesyonel olarak süt ve süt ürünleri yapmakta olan bir çiftlikte gerçekleştirilmiştir.



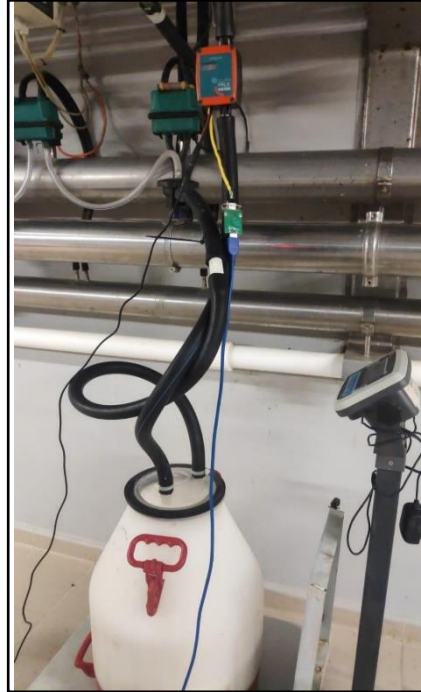
Şekil 7.6: Sıvı ölçerlerin sağım hanede montajı.

Testlerimizi birden fazla cihaz bağlantısı kurarak aynı anda farklı sağımlar gerçekleştirilecek şekilde yapıldı.



Şekil 7.7: Sağım hane test düzeneği.

Test düzeneğimiz ile birden fazla cihazı bilgisayar üzerinde kurduğumuz arayüzde kendi ölçümlerimizin verilerini kaydettik.



Şekil 7.8: Cihaz montajı ve tartı düzeneği.

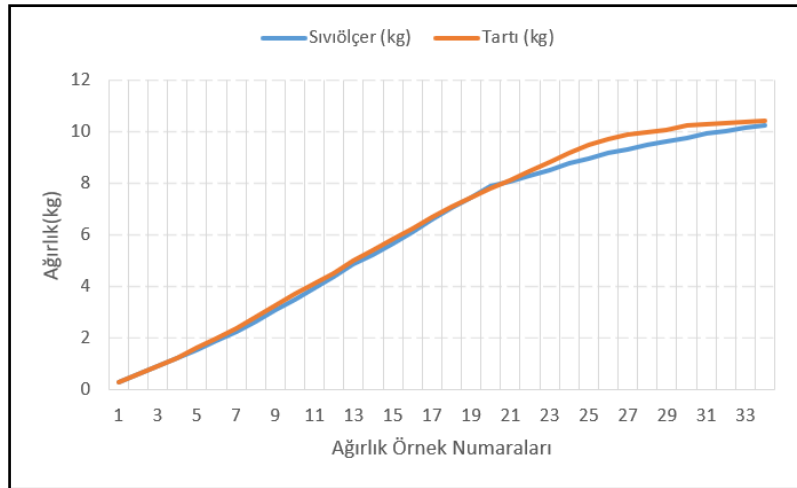
Sağım esnasında ayrıca sütün güğümlere sağımı yapılmıştır. Bu sayede güğümler tartı üzerine konularak anlık olarak süt miktarının takibi gerçekleştirilmiştir. Tartı ile elde ettiğimiz değerleri daha sonra cihazımızın verileri ile karşılaştırarak ölçüm doğruluk payları üzerine hesaplamalarımızı gerçekleştirdik.

7.3.2. Sağım Hane Testleri ve Sonuçları

Sağım hane testleri esnasında yaklaşık 3000 adet sağım gerçekleştirilmiştir ve bu sağım sonuçlarına göre algoritmada iyileştirmeler yapılarak en yüksek doğruluk sağlanmıştır.

Algoritmadaki iyileştirmeler sonucunda elde ettiğimiz bazı ölçüm verileri aşağıdadır.

Süre (sn)	Sıvıölçer (gr)	Tartı (gr)
30	0,313	0,300
40	0,602	0,600
50	0,91	0,950
60	1,22	1,250
70	1,53	1,650
80	1,89	2,000
90	2,27	2,400
100	2,67	2,850
110	3,09	3,300
120	3,52	3,750
130	3,94	4,150
140	4,38	4,550
150	4,88	5,000
160	5,22	5,400
170	5,67	5,850
180	6,12	6,250
190	6,6	6,700
200	7,05	7,100
210	7,47	7,450
220	7,89	7,800
230	8,105	8,150
240	8,325	8,500
250	8,545	8,850
260	8,775	9,200
270	8,995	9,500
280	9,195	9,750
290	9,345	9,900
300	9,495	10,000
310	9,645	10,100
320	9,795	10,250
330	9,945	10,300
340	10,045	10,350
350	10,17	10,400
360	10,245	10,450

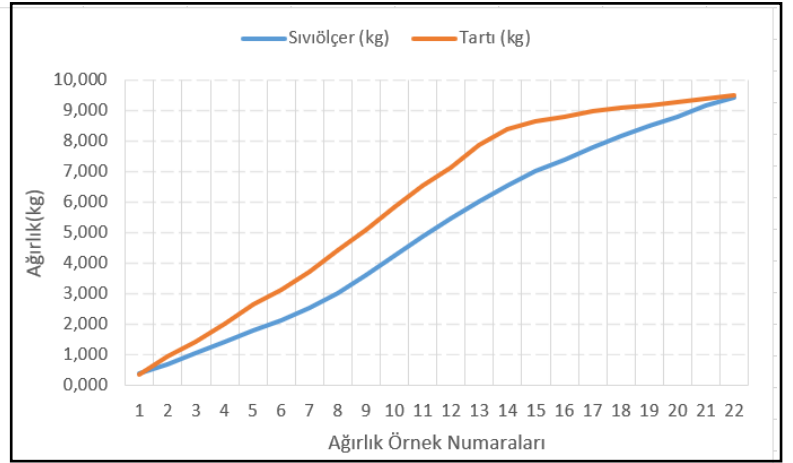


Şekil 7.9: Birinci test verileri grafiği.

Şekil 7.10: Birinci test verileri.

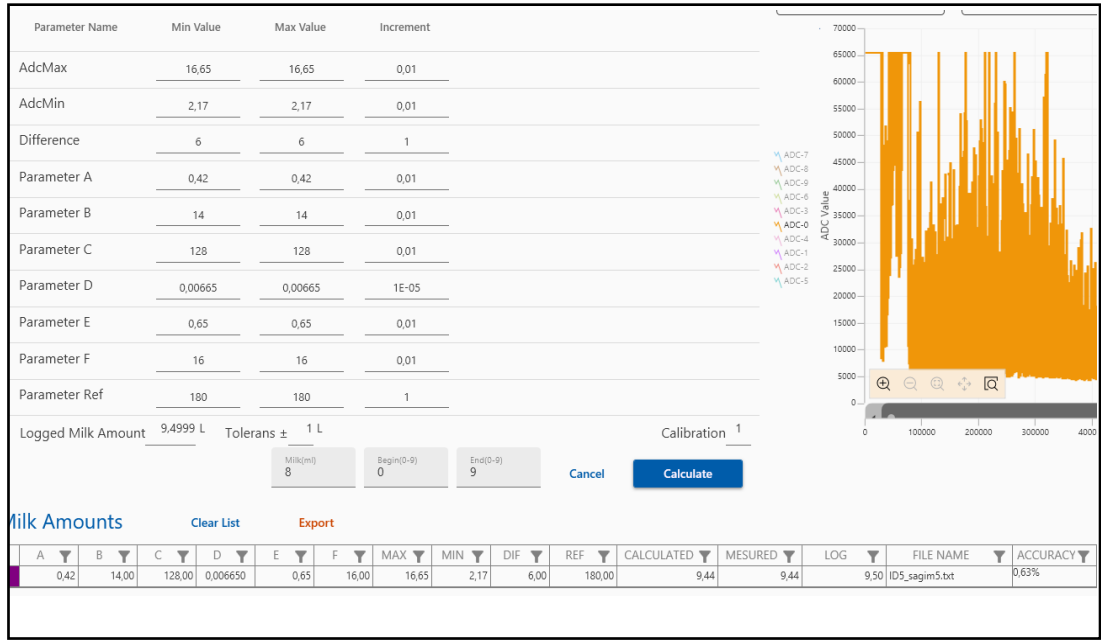
Yukarıdaki bir sağmal inekten elde edilmiş sağım verileridir. Tabloda her 10 saniyelik aralıklarla tartı üzerindeki anlık değerleri ve cihazın okuduğu değerler gösterilmektedir. Bu değerleri anlık olarak karşılaştırdığımızda doğruluk hataları biraz yüksek gibi görünmektedir. Bunun en büyük sebebi cihaz içerisinde geçen süt verisinin haberleşme ile aktarım hızı ve sütün cihazdan geçtikten sonra güğüme dökülmesinden kaynaklı zaman farklılıklarıdır. Sağım tamamen bittiğinde ve son ölçüm değerlerine baktığımızda ise hata oranımızın %1'den düşük olduğu görülmektedir. Bu ölçüm için hata oranımız %0.98'dir. Aşağıda ise bir başka sağım verisi ve bu verilerin arayüz programında analizi bulunmaktadır.

Süre (sn)	Sıvıölçer (gr)	Tartı (gr)
30	0,385	0,350
40	0,704	0,950
50	1,080	1,450
60	1,450	2,050
70	1,800	2,650
80	2,160	3,150
90	2,570	3,750
100	3,050	4,450
110	3,610	5,100
120	4,250	5,850
130	4,890	6,550
140	5,460	7,150
150	6,050	7,900
160	6,550	8,400
170	7,020	8,650
180	7,420	8,800
190	7,830	9,000
200	8,200	9,100
210	8,530	9,200
220	8,810	9,300
230	9,200	9,400
240	9,430	9,500



Şekil 7.11: İkinci test verileri grafiği.

Şekil 7.12: İkinci test verileri.



Şekil 7.13: İkinci test verilerinin arayüz ile görüntülenmesi.

Sağmal inekten yapılan sağımın verileri yukarıdadır ve tartı ile cihazımızın ölçümleri karşılaştırıldığında hata oranı %0,63 çıkmıştır. Ayrıca bu ölçümün log kayıtları arayüz programında incelenebilmektedir. Yukarıda ise bu log kayıtları ile gerçek değer karşılaştırılması ve bu karşılaştırılma sonucunda elde edilen doğruluk oranı görülmektedir. Bu arayüz programındaki yaptığımız karşılaştırmalar bize yapılan test sonuçlarının hızlı bir analizini yapmak ve algoritmamızda kullandığımız miktar hesaplama formülünün geliştirilmesinde çok büyük bir avantajı olmuştur. Arayüz programı sayesinde değerlendirmeler çok hızlı gerçekleştirilmiştir ve kısa zaman içerisinde çok yüksek miktarda testlerin gerçekleştirilmesi ve analizi yapılmıştır.

İki test verisi arasında ve diğer test verileri sonuçlarında grafikleri çizdirildiğinde bazı verilerde ara değerlerin sapmasının yüksek olduğu ve ara değerlerde hata payının yüksek olduğu görülmektedir. Bu hata payı yüksekliğinin sebebi arayüzde log kaydı tutulurken yaşanan gecikmeler ve sıvı miktar artışının bu gecikmelerden dolayı daha yüksek olarak kaydedilmesidir. Ölçümler esnasında aynı değerler anlık ölçülse de işlenmesinde yaşanan gecikmelerden dolayı bu problemle karşılaşmıştır.

Tablo 7.1: Sıvı ölçer test verileri.

Test Numarası	Tartı Değeri	Sıvıölçer Değeri	Hata Oranı
1	10,450	10,245	0.98%
2	9,500	9,430	0.63%
3	14,000	14,044	0,31%
4	12,100	11,940	1,32%
5	8,200	8,352	1,65%
6	13,150	13,180	0,22%
7	13,300	13,495	1,46

Yukarıda ise sağmal hayvan çiftliğinde yapmış olduğumuz testler ve bu testlerin sonuçları bulunmaktadır. Yukarıdaki testlerden iki tanesinin ayrıntılı sonuçları ve grafiği yukarıda bulunmaktadır. Diğer test verileri ve grafikleri ise ekler kısmındadır. Tablodaki testlerde görüldüğü üzere ölçümlerimizde %1 civarında hata payı bulunmaktadır. Tabloda verilen test sonuçlarının ayrıntılı tablo ve grafikleri ekler kısmında bulunmaktadır.

7.4. Sonuç ve Öneriler

Dünya genelinde önüne geçilemeyen hızlı bir nüfus artışı bulunmaktadır. Bu nüfus artışının beraberinde getirdiği gıda ihtiyacındaki artış tarım alanında ve tarım ürünlerinin işlenmesinde endüstrileşme ihtiyacını ortaya çıkarmıştır. Su, yağ, süt gibi ana gıda ürünlerin üretiminden işlenmesine ve hatta paketlenmesine kadar endüstrinin gelişmesi ve verimliliğinin artırılması gerekmektedir. Bu yüzden gıda alanında endüstriyel sisteme geçiş hızla devam etmektedir.

Endüstrinin en önemli amacı kısa sürede çok sayıda kaliteli ve verimli ürünler üreterek karlılığı en üst seviyede tutmaktır. Bu sebepten dolayı da tarım alanındaki yatırımcılarında tıpkı sanayi sektöründeki gibi tam otomasyonlu sistemler kullanarak işlerini yönetmeyi zorunlu kılmıştır.

Suyun kaynaktan çıkarılıp işlendikten sonra paketlenme süreci, sıvı yağların ürün çeşidine göre işlenerek üründen yağının çıkarılarak paketlenme süreci veya süt sağım çiftliklerinden sağmal hayvanın sütünün sağılmasından sütün işlenerek paketlenme sürecine kadar hijyenik, verimli ve analizlerini yapılarak verimliliğin artırılmasına kadar birçok konuya dikkat edilmesi gerekmektedir.

Kaynaktan çıkarılan suyun mineral oranı, içerdığı yararlı ve zararlı bakterilerin düzenli ölçümü ve miktarı, aynı şekilde sıvıyağlarda ürünün yağına karışabilecek yabancı maddenin tespiti ya da sağım hanelerde süt sağım esnasında sütün miktarı, süt içerisindeki bakteri miktarı, süt sağımında aynı hayvandan alınan süt miktarının takibi sayesinde hayvan verimliliğinin takibi ve sağım esnasında oluşabilecek meme kanaması sonucu süte kan karışımını engellemek gibi tarımsal gıda alanında birçok dikkat edilmesi ve verimin artırılmasına yardımcı olabilecek koşul bulunmaktadır.

Tarım ve gıda alanındaki kazançların artması ve bu alanların endüstrileşmesi ile işlerin büyümesi sonucunda tam otomasyonlu ve hijyenik yöntemler tercih edilmektedir. Bu çalışmada da bu alanlarda kullanılmak üzere bir ölçü aleti tasarımı yapılmış, tasarım bilgileri ve test sonuçları yazılmıştır.

Bu çalışmada tasarım aşamasında birçok debi ölçer araştırılmış ancak farklı sıvı türlerinin doğru ölçülmesi ihtiyacı olduğu için çift fazlı pulsatil akış yöntemine de uygun bir temassız ölçüm tasarımı yapılmıştır. Bu tasarım su ve süt olmak üzere iki farklı sıvı ile test edilmiş hem laboratuvar ortamında hem üretim tesislerinin simülasyonu oluşturularak hem de gerçek saha ortamında testleri gerçekleştirilerek tatmin edici sonuçlar elde edilmiştir.

Bu çalışma kapsamında testler hem süt ölçümünün zorluğu hem de saha şartlarının en zor olduğu sektör olduğu için sağmal hayvan çiftliklerinde gerçekleştirilmiştir. Çalışmamızda yüksek doğruluk elde edebilmek için süt yoğunluğuna uygun olarak bir yazılım algoritması oluşturulmuştur.

Çalışmamızda tasarladığımız ölçü aleti ile temassız sıvı miktar ölçümü mümkündür. Mililitre hassasiyetindeki ölçüm kapları ve hassas tartı ile ölçüm sonuçlarımızı karşılaştırdığımızda hata oranımızın yüzde 1 olarak süt miktarı ölçümünü gerçekleştirdiğini gördük. Su ve yağ gibi diğer sıvıların ölçümü içinde aynı şekilde yazılım algoritması geliştirilmesi ve saha testlerinin gerçekleştirilmesi gerekmektedir. Saha testleri ile birlikte gerçek ölçüm şartlarında elde edilen veriler daha doğru bir sonuç vereceği için yazılım algoritmasında geliştirmeler yapılması gerekebilir. Laboratuvar ortamında bu testler su için gerçekleştirilmiş ve yüksek doğruluklu sonuçlar elde edilmiştir. Aynı çalışmalar gerçekleştirildiğinde ve algoritmada düzeltmeler yapıldığında yüksek doğruluklu sonuçlar almak mümkündür.

Yapılan çalışmalar kapsamında elde edilen sonuçlar tasarlanan sistemin tam otomasyonlu bir hayvan sağım çiftliğine uygun olduğunu göstermektedir. Diğer sıvıların ölçümleri için algoritmalar gerçekleştirilerek arayüz üzerinden ölçülen sıvının

çeşidi seçilerek farklı sıvılar için tek bir cihaz ile ölçüm yapılmasına uygun bir tasarım olarak iyileştirme çalışmaları yapılabilir.

Sıvı ölçü aletinin kullanımı ve temizliğinin kolaylığı, cihaz üzerinden ölçülen tüm verilerin bilgisayar veya telefona kablo veya kablosuz olarak aktarabilme özelliği ve ilerideki çalışmalarla geliştirilmeye açık olması tam otomasyonlu üretim tesisleri için oldukça verimli olacaktır.

KAYNAKLAR

- [1] MEB, (2012), “Kimya Teknolojisi Enstrümantasyon Ders Notları”, Ankara: Milli Eğitim Bakanlığı
- [2] MEB, (2008), “Akışkanlar Mekaniği Ders Notları”, Ankara: Milli Eğitim Bakanlığı.
- [3] MEB, (2010), “Endüstriyel Otomasyon Teknolojilerinde Akış Ölçümü1, Ankara: Milli Eğitim Bakanlığı.
- [4] Web 1, (2008), http://www.bayar.edu.tr/besergil/2_akma_olayi.pdf, (Erişim Tarihi: 20/05/2022).
- [5] Ertaş, E., (2007), "Geçiş Zamanlı Esaslı Bir Ultrasonik Akış Ölçerin Tasarımı ve Uygulaması", Yüksek Lisans Tezi, Gazi Üniversitesi.
- [6] Bilgili, H., (2013), "QCM Akış Ölçer Tasarımı ve Üretimi", Yüksek Lisans Tezi, Başkent Üniversitesi.
- [7] Feng, C. C., Lin, W. T., and Yang, C. T., (2011), "Laminar flow meter with straight glass capillary", Mapan, 26(3), 237-245.
- [8] Bayındır, C., (2013), "Akışkan Görüntülüme Teknikleri", Bilim ve Teknik Dergisi, (543), 67-70.
- [9] Aktacir, A., (2010), "<http://eng.harran.edu.tr/~hbulut/5akis.pdf>", (Erişim Tarihi : 25/05/2022).
- [10] Kaçur, G., (2005), "A Low-Cost Microcontroller Based Gas Flowmeter Using a Specially Designed Parabolic Airway Resistor", Yüksek Lisans Tezi, Boğaziçi Üniversitesi.
- [11] MEB, (2013), “Proseste Akış ve Seviye Ders Notları”, Ankara: Milli Eğitim Bakanlığı.
- [12] Heinz, G., (2016), "www.tuncell.com/userfiles/Flow%20Metering%20Controlling%20Batching%20Monitoring%20and%20Reconrding-OVAL-GEAR-METER.pdf", (Erişim Tarihi : 24/02/2022).
- [13] Yılmaz, S., Arslan, N., Toker, O. and Sedef, H., (2010), "In vitro pulsatile flow waveform simulation and comparison using AC pump and proportional pneumatic valve", 15th National Biomedical Engineering Meeting (BIYOMUT),

1-5, İstanbul, Türkiye, 21-24 Nisan.

- [14] Haydar, E., Çıtlak, A., ve Yıldız, Ş., (2006), "Pervaneli Tip Akış Ölçerlerin Diferansiyel Basınç Ölçer Prensipleri ile Debi Ölçümünde Kullanılabilirliği", Tesisat Mühendisliği Dergisi, 5(1), 63-68.
- [15] Nishimura, F.R., Kawashima, S.K., and Kagawa, T.T., (2008), "Analysis of laminar flow meter with flute-type cross section laminar elements", 7th International Asia Simulation Conference, 1110-1114 10-12 October.
- [16] Ergelen, C., (1998), "Volimetrik Esaslı Bir Akış Ölçerin Dizaynı İmalı ve Kalibrasyonu", Yüksek Lisans Tezi, Gazi Üniversitesi.
- [17] Akyazı, H., Güzel, İ., Çataklı, T. ve Karadurmuş, E., (2015), "Açık Kanallarda ve Akarsularda Ultrasonik Yöntemlerle Debi Ölçülmesi", Hitit Üniversitesi.
- [18] İskender, İ., Emre, E., ve Naci, G., (2007), "Tasarım ve Uygulaması Yapılan Bir Ultrasonik Akış Ölçerin Kalibrasyonu", Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 22(4), 759-767.
- [19] Demirci, A., (2008), "Taşınabilir, Düşük Güçlü, 10MHZ Sürekli Dalga Ultrasonik Dopler Kan Akış Ölçer", Yüksek Lisans Tezi, Gazi Üniversitesi.
- [20] Ardıçlıoğlu, M., Gemici, E., ve Özden, S., (2011), "Doğal Akarsularda Debinin Belirlenmesi", Tarım Bilimleri Araştırma Dergisi, 4 (2), 73-77.
- [21] Göncü, S., (2016),
<http://traglor.cu.edu.tr/objects/objectFile/ohm1qSE5-2342013-58.pdf> , (Erişim Tarihi : 7 Mayıs 2016),.
- [22] MUNSON Bruce R., YOUNG Donald F., OKIISHI Theodore H., HUEBSCH Wade W. M., (2010), "Fundamentals of Fluid Mechanics", John Wiley and Sons Inc..
- [23] Öz, H., Meierhofer, S. R., Ströbel, U. and Ammon, C., (2010), "Comparison of the Vacuum Dynamics of Conventional and Quarter Individual Milking Systems", Journal of Agricultural Sciences, 16(3), 162-168.
- [24] Tekin, A.B., Değirmencioğlu, A., (2010), "Tarımsal Bilişim: İleri Tarım Teknolojileri", Akademik Bilişim Konferansı, 9(1), 351-359 Muğla, Türkiye, 10-12 Şubat.
- [25] Artmann, R., (1997), "Sensor systems for milking robots", Computers and electronics in agriculture, 17(1), 9-40.

- [26] Takeda, Y., (1999), "Ultrasonic Doppler Method for Velocity Profile Measurement in Fluid Dynamics and Fluid Engineering", *Experiments in fluids*, 26(3), 177- 178.
- [27] Çobanoğlu, B., Şentürk, A.İ., (2015), "Kalibrasyon Laboratuvarları İçin Sayısal Sıcaklık-Nem Ölçer Kayıt Cihazının Tasarım Ve Gerçekleştirimi", *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 19(3), 371-375.
- [28] Doruk, A., Buluş, H. N., Moralar, A., Çelen, S., Uzun, E., (2015), "Fiziksel Programlama Platformları Kullanarak Elektrikli Taşıtların Hız, Batarya ve Sıcaklık Verilerinin Takibi", *Electronic Journal of Vocational Colleges*, 5(4), 48-56.
- [29] Shokouhian, M., Morling, R., and Kale, I., (2016), "Interference Resilient Sigma Delta-Based Pulse Oximeter", *IEEE Transactions On Biomedical Circuits and Systems*, 8(1), 623-631.
- [30] Türkyılmaz, M.K., (2005), "Süt Sığırcılık İşletmelerinde Sağım Robotu Kullanımı", *Erciyes Üniversitesi Veteriner Fakültesi Dergisi*, 4(1), 61-65.
- [31] Mortensen J., (2007), "Polymer coated QCM sensors for detection of the mix gases", *NATBAS 4th Semester Report in Chemistry made by group 7, Roskilde University*.
- [32] Michalzik M., Wilke R., Buttgenbach S., (2005), "Miniaturized QCM-based flow system for immunosensor application in liquid", *Sensors and Actuators B* 111–112, 410–415,.
- [33] Spahr, S.L., Maltz, E., (1997), "Herd management for robot milking", *Computers and electronics in Agriculture*, 17(1), 53-62.
- [34] Artmann, R., (1997), "Sensor systems for milking robots", *Computers and electronics in agriculture*, 17(1), 19-40.
- [35] Nedret, A. ve Tülay, B., (1997), "Mastitis İle Mücadele Programının Uygulaması Üzerinde Çalışmalar", *Türkiye Bilimsel ve Teknik Araştırma Kurumu*, 989.
- [36] Özer, A. S., (2014), "Seyyar Süt Sağım Makinasının Gömülü Sistem İle Otomasyonu", *Yüksek Lisans Tezi, Selçuk Üniversitesi*.
- [42] Demir, C., Gönüloğlu, E., ve Ülger, P., (2012), "Merkezi Köy Süt Sağım Tesislerinin Uygulanabilirliği", *Tarım Makinaları Bilimi Dergisi*, 8(1), 13-18.
- [43] Alıç, D., Yener, M., (2006), "Süt Sığırcılığı İşletmelerinde Robotlu Sağım

Sistemi", Tarım Bilimleri Dergisi, 12(4), 369-380.

ÖZGEÇMİŞ

Mücahit ERDEM, 2013 yılında başladığı Gebze Teknik Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümünü 2018 yılında başarıyla tamamladıktan sonra 2020 yılında yüksek lisans eğitimine Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalında başladı. 2018 yılından beri farklı sektörlerde hizmet veren teknoloji şirketlerinde görev aldım. Şu anda IOT tabanlı teknolojiler üreten bir firmada iş hayatıma devam ediyorum.

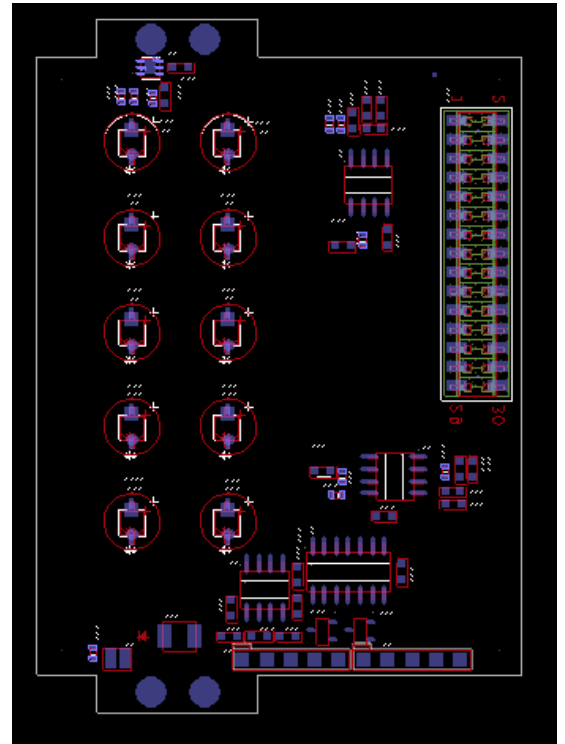
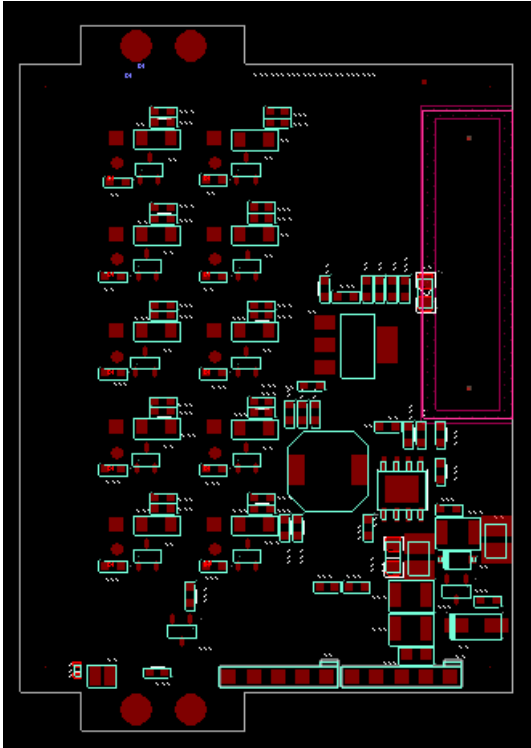
EKLER

Ek-A Tez Çalışması Kapsamında Yapılan Yayınlar

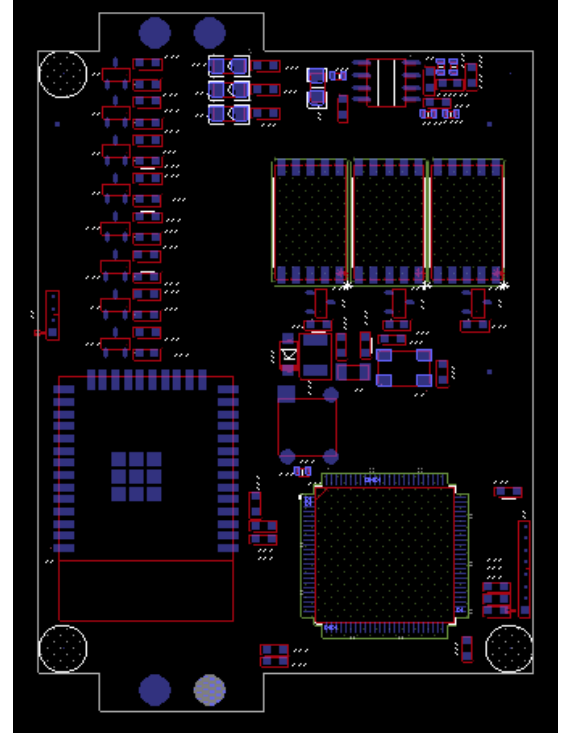
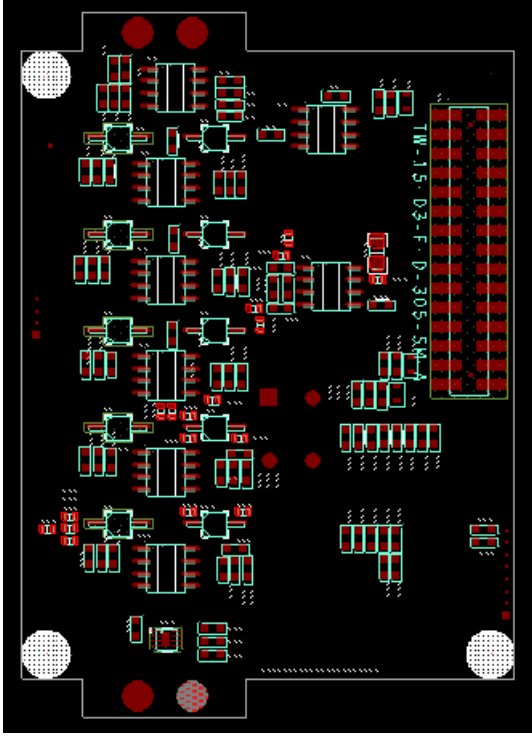
Erdem M., (2023), “Akış esnasında sıvılarda temassız ölçüm”, 7. Lisansüstü Araştırmalar Sempozyumu, Gebze, Türkiye 31 Mayıs – 01 Haziran.

Ek-B Sıvı Ölçer Baskı Devre Şeması

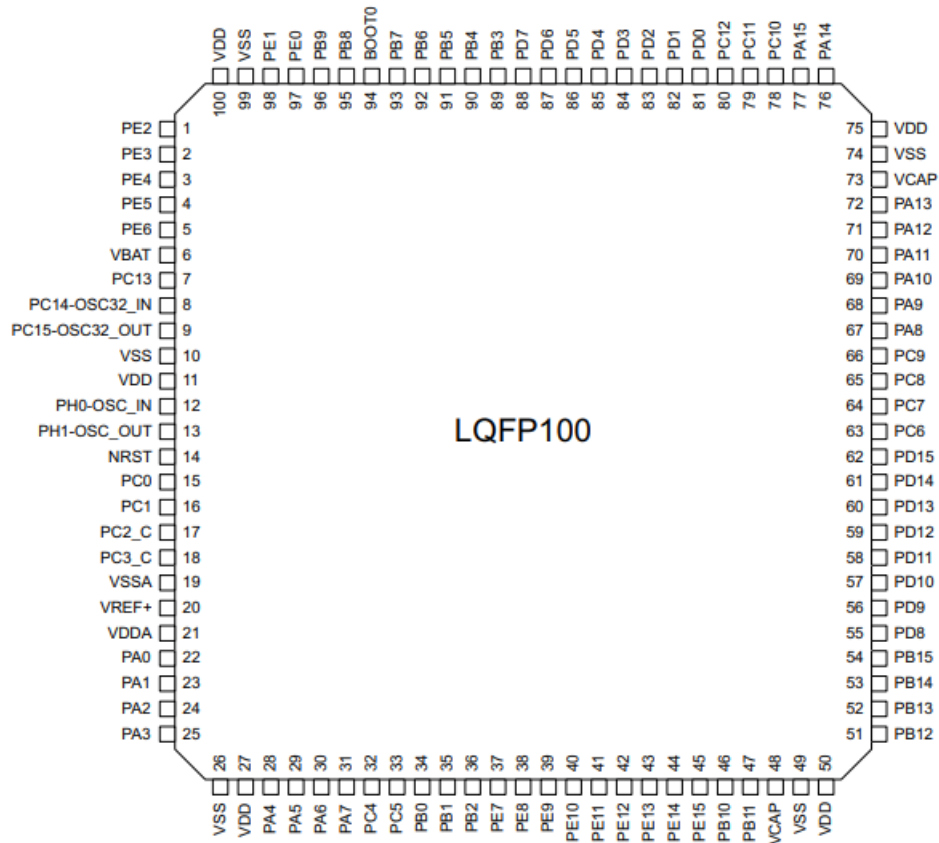
Güç katmanı Baskı Devre Şeması



İşlemci katmanı Baskı Devre Şeması



Ek-C STM32H723VET6 Bağlantı Bacakları



EK-D Yazılım Kod Parçaları

Yazılım Döngüsü

```
while (1)

{

    Milk_Quantity_Calc(); // Sıvı miktar fonksiyonu

    Device_Reset();      // Buton fonksiyonu

    Temperature_Calc();  // Sıcaklık fonksiyonu

    RGB_Calc();          // Renk fonksiyonu

    SevenSegment();      // Display fonksiyonu

    ESP_Connection();    // Haberleşme fonksiyonu

}
```

Süt Varlığı Yazılımı

```
void Milk_Quantity_Calc(void)

{

    Milk_Speed_Calc();

    if(((FrequencyGiris < HighValue) && (FrequencyCikis < HighValue) &&
(ADC_VAL[7] > (ADC_Ust_Num7-UstHysteresis))

                                && (ADC_VAL[8] >
(ADC_Ust_Num8-UstHysteresis))

                                && (ADC_VAL[2] >
(ADC_Ust_Num7-UstHysteresis))

                                && (ADC_VAL[5] >
(ADC_Ust_Num8-UstHysteresis))
```

```

&& (ADC_VAL[9] >
(ADC_Ust_Num7-UstHysteresis))

&& (ADC_VAL[3] >
(ADC_Ust_Num7-UstHysteresis))

&& (ADC_VAL[4] >
(ADC_Ust_Num7-UstHysteresis))

&& (ADC_VAL[6] >
(ADC_Ust_Num8-UstHysteresis))

&& (ADC_VAL[0] >
(ADC_Ust_Num8-UstHysteresis))

&& (ADC_VAL[1] >
(ADC_Ust_Num8-UstHysteresis))) && FLAG.LedCalibrationFlag)

{

//Eğer akan bir sıvı yok ise kod bu kısma girer.

FLAG.SpeedCalculateStartFlag=1;

FLAG.SpeedCalculateStopFlag=0;

if(FLAG.MakeSureFlag!=3)

{

FLAG.MakeSureFlag = 1;

}

Milk_Check();

if(FLAG.MakeSureFlag==1)

{

COUNTER.MakeSureCounter++;

if(COUNTER.MakeSureCounter>=100)

{

```

```

    COUNTER.MakeSureCounter = 0;

    FLAG.MakeSureFlag = 2;

}

}

if(FLAG.MakeSureFlag == 2)

{

    FLAG.RGBLedsOk=0;

    COUNTER.BloodCounter=0;

    if(FLAG.LedCloseFlag == 3)

    {

        FLAG.LedCloseFlag=1;

    }

    FLAG.TotalTimeFlag=0;

    FLAG.SpeedTimerFlag2=0;

    FLAG.MilkFinishedFlag=0;

    MilkSpeed = 0;

    TotalTime=0;

    SpeedTimer = 0;

    Log_Tx_Data_Average[0]=0;

    Log_Tx_Data_Average[1]=0;

    RGB_BLUE_DISABLE;

    RGB_GREEN_DISABLE;

    RGB_RED_DISABLE;

    redBuffer = 0;

```

```

        greenBuffer = 0;

        blueBuffer = 0;

        MilkDataIndexToplam = 0;

        FLAG.MilkWhere = 0;

        COUNTER.MakeSureLastADC = 0;

    }

}

if((FrequencyGiris>HighValue||FrequencyCikis>HighValue||((ADC_VAL[2]<(ADC
_Ust_Num2-UstHysteresis))&&(ADC_VAL[2]>(ADC_Alt_Num2+AltHysteresis)))

        ||((ADC_VAL[5]<(ADC_Ust_Num5-
UstHysteresis))&&(ADC_VAL[5]>(ADC_Alt_Num5+AltHysteresis)))

        ||((ADC_VAL[7]<(ADC_Ust_Num7-
UstHysteresis))&&(ADC_VAL[7]>(ADC_Alt_Num7+AltHysteresis)))

        ||((ADC_VAL[8]<(ADC_Ust_Num8-
UstHysteresis))&&(ADC_VAL[8]>(ADC_Alt_Num8+AltHysteresis)))

        ||((ADC_VAL[3]<(ADC_Ust_Num3-
UstHysteresis))&&(ADC_VAL[3]>(ADC_Alt_Num3+AltHysteresis)))

        ||((ADC_VAL[4]<(ADC_Ust_Num4-
UstHysteresis))&&(ADC_VAL[4]>(ADC_Alt_Num4+AltHysteresis)))

        ||((ADC_VAL[9]<(ADC_Ust_Num9-
UstHysteresis))&&(ADC_VAL[9]>(ADC_Alt_Num9+AltHysteresis)))

        ||((ADC_VAL[0]<(ADC_Ust_Num0-
UstHysteresis))&&(ADC_VAL[0]>(ADC_Alt_Num0+AltHysteresis)))

        ||((ADC_VAL[1]<(ADC_Ust_Num1-
UstHysteresis))&&(ADC_VAL[1]>(ADC_Alt_Num1+AltHysteresis)))

```

```

                                                                    ||((ADC_VAL[6]<(ADC_Ust_Num6-
UstHysteresis))&&(ADC_VAL[6]>(ADC_Alt_Num6+AltHysteresis)))) &&
FLAG.LedCalibrationFlag)

    {

        FLAG.MilkFinishedFlag=1;

        FLAG.FirstMilkData = 1;

        FLAG.MakeSureFlag = 0;

        COUNTER.MakeSureCounter = 0;

        FLAG.TotalTimeFlag = 1;

        if(((ADC_VAL[7]<(ADC_Ust_Num7-
UstHysteresis))&&(ADC_VAL[7]>ADC_Alt_Num7))||((ADC_VAL[8]<(ADC_Ust
_Num8-UstHysteresis))&&(ADC_VAL[8]>ADC_Alt_Num8)))

            {

                if(FLAG.SpeedTimerFlag2!=2)

                    {

                        FLAG.SpeedTimerFlag2 = 1;

                    }

            }

            FLAG.RGBLedsOk=1;

            FLAG.LedCloseFlag = 3;

            COUNTER.LedCloseCounter = 0;

            OpenIRLed();

        }

    }

```

Sıvı Akışı Hız Yazılımı

```
void Milk_Speed_Calc(void)
{
    if(!MilkSpeed)
    {
        if(FLAG.SpeedCalculateStartFlag && FLAG.FirstMilkData)
        {
            if(FLAG.LedCloseFlag!=2)
            {
                OpenIRLed();
            }

            if((ADC_VAL[7]<(ADC_Ust_Num7-
1000))||ADC_VAL[8]<(ADC_Ust_Num8-1000)) ||
FrequencyGiris>HighValue)//Giriste süt var ve

            {

                if(FrequencyGiris>HighValue)
                {
                    FLAG.MilkWhere = 1;
                }else
                {
                    FLAG.MilkWhere = 0;
                }

                if(ADC_VAL[2]>(ADC_Ust_Num2-
1000))||ADC_VAL[5]>(ADC_Ust_Num5-1000))//Çikista süt yoksa

                {
```

```

        if(!TotalTime)

            FLAG.SpeedTimerFlag = 1;

        }

    }

    if((ADC_VAL[2]<(ADC_Ust_Num2-
1000)||ADC_VAL[5]<(ADC_Ust_Num5-
1000)||FrequencyCikis>HighValue)&&(FLAG.SpeedTimerFlag)&&(FLAG.LedCali
brationFlag)&&(FLAG.SpeedCalculateStartFlag)&&(!FLAG.SpeedCalculateStopFl
ag))

    {

        FLAG.SpeedCalculateStopFlag = 1;

        FLAG.SpeedTimerFlag = 0;

        FLAG.SpeedCalculateStartFlag = 0;

        if(((ADC_VAL[2]<(ADC_Ust_Num2-
UstHysteresis))||(ADC_VAL[5]<(ADC_Ust_Num5-UstHysteresis))) &&
!FLAG.MilkWhere)

        {

            MilkSpeed = FiveLedDistance / (SpeedTimer/1000); //1 saniyede kaç m
süt geçtiği (metre / saniye)

        }else if(((ADC_VAL[2]<(ADC_Ust_Num2-
UstHysteresis))||(ADC_VAL[5]<(ADC_Ust_Num5-UstHysteresis))) &&
FLAG.MilkWhere)

        {

            MilkSpeed = 0.051 / (SpeedTimer/1000); //1 saniyede kaç m süt geçtiği
(metre / saniye) (+0.011)

        }else if((FrequencyCikis>HighValue) && !FLAG.MilkWhere)

```

```

{
    MilkSpeed = (0.056) / (SpeedTimer/1000); //1 saniyede kaç m süt geçtiği
(metre / saniye) (+0.016)

    }else if((FrequencyCikis>HighValue) && FLAG.MilkWhere)

    {

        MilkSpeed = (0.067) / (SpeedTimer/1000); //1 saniyede kaç m süt geçtiği
(metre / saniye) (+0.027)

    }

    if(MilkSpeed<0.1)

    {

        MilkSpeed = 0.1;

    }else if(MilkSpeed>0.5)

    {

        MilkSpeed = 0.5;

    }

    }

}

}

```

Sıvı Miktarı Hesaplaması

```

if(FLAG.MilkFinishedFlag)

{

    Average_Calc_Func2(7,8); // 7 ve 8. Adc değerleri kullanılarak oradan
geçen sütün ortalaması bulunur

    Average_Calc_Func2(9,6); // 9v 6

```

```

Average_Calc_Func2(3,0); // 3 ve 0

Average_Calc_Func2(4,1); //4 ve1

Average_Calc_Func2(2,5); //2 ve 5

MilkDataIndexToplam++;

MilkDataIndexReference2++;

if(MilkDataIndexReference2 > (MilkDataIndexReference-1))
{
    AverageTop = Average1 + Average2;

    AverageTop = AverageTop / 10; //ADC Sayisi

    AverageTop /= MilkDataIndexReference;

    Hacim_New = AverageTop / 100.0;

    Hacim_New *= Hacim;

    Segments_ValueTest += Hacim_New;

    /*if(Segments_ValueTest >= 13000)
    {
        Segments_ValueTest = 14100;
    }*/

    Segments_Value = Segments_ValueTest;

    SegmentsValueFloat = ((double)Segments_ValueTest) / 1000.0;

    Average1=0;

    Average2=0;

    MilkDataIndexReference2 = 0;

}

```

ADC Ortalama Hesaplaması

```
void Average_Calc_Func2(int FirstIndex,int SecondIndex)
{
    ADCArray[FirstIndex] = ADC_VAL[FirstIndex] / 1000.0;
    ADCArray[SecondIndex] = ADC_VAL[SecondIndex] / 1000.0;
    if(ADCArray[FirstIndex] <= 16.65 && (ADCArray[FirstIndex] > 2.17))
    {
        AverageADCArray[FirstIndex] = (0.42 * (ADCArray[FirstIndex]) *
(ADCArray[FirstIndex])) - (14.0 * (ADCArray[FirstIndex])) + 128.0;
    }else if((ADCArray[FirstIndex]) > 16.65 && (ADCArray[FirstIndex]) <
((ADC_Ust_Num7/1000.0)-6.0))
    {
        AverageADCArray[FirstIndex] = (0.00665 * (ADCArray[FirstIndex]) *
(ADCArray[FirstIndex])) - (0.65 * (ADCArray[FirstIndex])) + 16;
    }else if(ADCArray[FirstIndex] >= ((ADC_Ust_Num7/1000.0)-6.0))
    {
        AverageADCArray[FirstIndex] = 0.0;
    }else
    {
        AverageADCArray[FirstIndex] = 100.0;
    }
    Average1+=AverageADCArray[FirstIndex];

    if((ADCArray[SecondIndex]) <= 16.65 && ((ADCArray[SecondIndex]) >
2.17))
```

```
{
```

```
        AverageADCArray[SecondIndex] = (0.42 * ((ADCArray[SecondIndex])) *  
(ADCArray[SecondIndex])) - (14.0 * (ADCArray[SecondIndex])) + 128.0;  
  
        }else if((ADCArray[SecondIndex]) > 16.65 && (ADCArray[SecondIndex])  
< ((ADC_Ust_Num7/1000.0)-6.0))  
  
        {  
  
            AverageADCArray[SecondIndex] = (0.00665 * (ADCArray[SecondIndex])  
* (ADCArray[SecondIndex])) - (0.65 * (ADCArray[SecondIndex])) + 16;  
  
            }else if((ADCArray[SecondIndex]) >= ((ADC_Ust_Num7/1000.0)-6.0))  
  
            {  
  
                AverageADCArray[SecondIndex] = 0.0;  
  
            }else  
  
            {  
  
                AverageADCArray[SecondIndex] = 100.0;  
  
            }  
  
            Average2+=AverageADCArray[SecondIndex];  
  
        }
```

Buton Fonksiyonu

```
ButtonPressed = HAL_GPIO_ReadPin(GPIOE,GPIO_PIN_3);  
  
if(ButtonPressed)  
  
{  
  
    ButtonPressedCounter++;  
  
    ResetWithButton++;  
  
}else  
  
{
```

```
ButtonPressedCounter=0;
```

```
ResetWithButton=0;
```

```
}
```

```
if(ButtonPressedCounter>300000)//3 sn {
```

```
    ButtonPressedCounter=0;
```

```
    Segments_Value=0;
```

```
    SegmentsValueFloat=0;
```

```
    gosterge1=0;
```

```
    gosterge2=0;
```

```
    gosterge3=0;
```

```
    FLAG.index_Point_Flag=0;
```

```
}
```

```
if(ResetWithButton>700000)//7sn
```

```
{
```

```
    ResetWithButton=0;
```

```
    Segments_Value=0;
```

```
    SegmentsValueFloat=0;
```

```
    gosterge1=0;
```

```
    gosterge2=0;
```

```
    gosterge3=0;
```

```
    FLAG.index_Point_Flag=0;
```

```
    NVIC_SystemReset();
```

```
}
```

Sıcaklık Ölçüm Fonksiyonu

```
void Temperature_Calc(void)
{
    TempReadValue++;
    if(TempReadValue>50000)
    {
        TempReadValue = 0;
        FLAG.TempReadFlag = 1;
        MilkTemperature = TMP117_get_Temperature(hi2c2);
        if(MilkTemperature > TemperatureThreshold)
        {
            No_Value = 20;
        }
    }else
    {
        FLAG.TempReadFlag = 0;
    }
}
```

Yabancı Madde Tespit Fonksiyonu

```
void RGB_Calc(void)
{
    //RGB Ledlerini yöneten kod kısmı
```

```
if(FLAG.RGBLedIsOk)
```

```
{  
  
    RGB_GREEN_ENABLE;  
  
    RGB_BLUE_ENABLE;  
  
    COUNTER.RgbSayac++;  
  
    if(COUNTER.RgbSayac==100)  
  
    {  
  
        RGB_RED_ENABLE;  
  
        ColourSensor_Read();  
  
    }else if(COUNTER.RgbSayac==200)  
  
    {  
  
        RGB_RED_DISABLE;  
  
    }else if(COUNTER.RgbSayac==400)//azalttikca yesil ve mavi degeri  
azaliliyor kirmizi degeri artiyor  
  
    {  
  
        COUNTER.RgbSayac=0;  
  
    }  
  
}  
  
}
```

Sensör Okuma Fonksiyonu

```
void ColourSensor_Read(void)//Renk sensoru  
  
{  
  
    HAL_I2C_Mem_Read(&hi2c1,(0x38<<1),0x50,1,&bufferrgb[0],6,100);  
  
    redBuffer=(bufferrgb[1]<<8)|bufferrgb[0];  
  
}
```

```
greenBuffer=(bufferrgb[3]<<8)|bufferrgb[2];
```

```
blueBuffer=(bufferrgb[5]<<8)|bufferrgb[4];
```

```
if(FLAG.BloodFlag)
```

```
{
```

```
    if(redBuffer>RedThreshold)
```

```
    {
```

```
        COUNTER.BloodCounter++;
```

```
        if(COUNTER.BloodCounter>250)
```

```
        {
```

```
            COUNTER.BloodCounter=0;
```

```
            //No_Value = 19;
```

```
        }
```

```
    }
```

```
}
```

```
}
```

Display Ekran Fonksiyonu

```
void SevenSegment(void)
```

```
{
```

```
    //Ekrani kapatıp açmak için yazılan kod kısmı
```

```
    COUNTER.segment_sayac++;
```

```
    Seg_Value_Calculate();
```

```
    if(COUNTER.segment_sayac==10)
```

```
    {
```

```
        Seg_Goster1();
```

```
}else if(COUNTER.segment_sayac==20)
```

```
{  
    Seg_Goster2();  
}else if(COUNTER.segment_sayac==30)  
{  
    COUNTER.segment_sayac=0;  
    Seg_Goster3();  
}  
}
```

Seg Value Calculate Fonksiyonu

```
void Seg_Value_Calculate(void)//yedi segmentli ekran yazdirma hesabi  
{  
    if(!No_Value)  
    {  
        if((uint32_t) Segments_Value<=999999)  
        {  
            if ((uint32_t) Segments_Value > 999)  
            {  
                FLAG.index_Point_Flag = 1;  
                New_Segments_Value=(uint32_t) Segments_Value;  
                New_Segments_Value /= 10;  
                if(New_Segments_Value > 999)  
                {  
                    FLAG.index_Point_Flag = 2;
```

```
New_Segments_Value2 = New_Segments_Value;
```

```
New_Segments_Value2 /= 10;

if (New_Segments_Value2 > 999)
{
    FLAG.index_Point_Flag = 3;

    gosterge1 = New_Segments_Value2 / 1000;
    gosterge2 = (New_Segments_Value2 % 1000) / 100;
    gosterge3 = (New_Segments_Value2 % 100) / 10;
}else
{
    gosterge1 = (uint32_t) Segments_Value / 10000;
    gosterge2 = ((uint32_t) Segments_Value % 10000) / 1000;
    gosterge3 = ((uint32_t) Segments_Value % 1000) / 100;
}

}else
{
    gosterge1 = (uint32_t) Segments_Value / 1000;
    gosterge2 = ((uint32_t) Segments_Value % 1000) / 100;
    gosterge3 = ((uint32_t) Segments_Value % 100) / 10;
}

}else
{
    if(!FLAG.ElektrikFlag)
    {
```

```

gosterge1 = (uint32_t) Segments_Value / 100;

gosterge2 = ((uint32_t) Segments_Value % 100) / 10;

gosterge3 = (uint32_t) Segments_Value % 10;

}else

{

FLAG.ElektrikFlag=0;

Segments_Value = 100*gosterge1+10*gosterge2+gosterge3;

if(FLAG.index_Point_Flag==1)

{

Segments_Value = Segments_Value*10;

}else if(FLAG.index_Point_Flag==2)

{

Segments_Value = Segments_Value*100;

}else if(FLAG.index_Point_Flag==3)

{

Segments_Value = Segments_Value*1000;

}

}

}

}else

{

Segments_Value=0;

SegmentsValueFloat=0;

FLAG.index_Point_Flag = 0;

```

```
}
```

```
}else  
{  
    switch(No_Value)  
    {  
  
        case 1://btc (Bluetooth connected)  
  
            gosterge1=12;gosterge2=13;gosterge3=14;  
  
            break;  
  
        case 2://btd (Bluetooth disconnected)  
  
            gosterge1=12;gosterge2=13;gosterge3=15;  
  
            break;  
  
        case 3://On  
  
            gosterge1=0;gosterge2=18;gosterge3=20;  
  
            break;  
  
        case 4://Off  
  
            gosterge1=0;gosterge2=19;gosterge3=19;  
  
            break;  
  
        case 5://-5-  
  
            gosterge1=20;gosterge2=5;gosterge3=20;  
  
            break;  
  
        case 6://-4-  
  
            gosterge1=20;gosterge2=4;gosterge3=20;  
  
            break;  
  
        case 7://-3-
```

```
gosterge1=20;gosterge2=3;gosterge3=20;
```

```
break;
```

```
case 8://-2-
```

```
gosterge1=20;gosterge2=2;gosterge3=20;
```

```
break;
```

```
case 9://-1-
```

```
gosterge1=20;gosterge2=1;gosterge3=20;
```

```
break;
```

```
case 10://-0-
```

```
gosterge1=20;gosterge2=0;gosterge3=20;
```

```
break;
```

```
case 11://-9-
```

```
gosterge1=20;gosterge2=9;gosterge3=20;
```

```
break;
```

```
case 12://-8-
```

```
gosterge1=20;gosterge2=8;gosterge3=20;
```

```
break;
```

```
case 13://-7-
```

```
gosterge1=20;gosterge2=7;gosterge3=20;
```

```
break;
```

```
case 14://-6-
```

```
gosterge1=20;gosterge2=6;gosterge3=20;
```

```
break;
```

```
case 15://btL (Bootloader)
```

```
gosterge1=12;gosterge2=13;gosterge3=21;
```

```
break;
```

```
case 16://Str (Start)
```

```
gosterge1=22;gosterge2=13;gosterge3=11;
```

```
break;
```

```
case 17://PrS (Press)
```

```
gosterge1=23;gosterge2=11;gosterge3=22;
```

```
break;
```

```
case 18://HOP
```

```
gosterge1=24;gosterge2=0;gosterge3=23;
```

```
break;
```

```
case 19://bLd (Blood)
```

```
gosterge1=12;gosterge2=21;gosterge3=15;
```

```
break;
```

```
case 20://Hot
```

```
gosterge1=24;gosterge2=17;gosterge3=13;
```

```
break;
```

```
}
```

```
}
```

```
}
```

GosterOutput Fonksiyonu

```
void GostergeOutput(uint8_t Value) //ekrandaki harf ya da sayi fonksiyonu
```

```
{
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_3,GPIO_PIN_SET);//SEVEN
```

SEGMENT 3

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_4,GPIO_PIN_SET); //SEVEN
```

SEGMENT 2

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_7,GPIO_PIN_SET); //SEVEN
```

SEGMENT 1

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_RESET); //Üst çizgi
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_RESET); //Orta  
çizgi
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_RESET); //Alt çizgi
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_RESET); //Sol üst  
çizgi
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_RESET); //Sol alt  
çizgi
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_RESET); //Sag üst  
çizgi
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_RESET); //Sag alt  
çizgi
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_11,GPIO_PIN_RESET);
```

//Noktalar

```
switch(Value)
```

```
{
```

```
case 0 ://0
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_  
WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP  
IO_PIN_5,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_10,GPIO_PIN_SET);
```

```
    break;
```

```
    case 1://1
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
        break;
```

```
    case 2 ://2
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);
```

```
        break;
```

```
    case 3://3
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
        break;
```

```
    case 4 ://4
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_WritePi
n(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GPIO_PIN
_1,GPIO_PIN_SET);
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
        break;
```

case 5://5

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

case 6://6

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

case 7://7

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

case 8://8

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_5,GPIO_PIN_SET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_9,GPIO_PIN_SET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

```
case 9://9
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_5,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

```
case 10://E
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);
```

```
break;
```

```
case 11: //r
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);
```

```
break;
```

```
case 12://b
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

case 13://t

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);
```

```
        break;
```

case 14://c

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_9,GPIO_PIN_SET);
```

```
        break;
```

case 15://d

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,G
PIO_PIN_1,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_WritePi
n(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);
```

```
        break;
```

case 16://u

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_10,GPIO_PIN_SET);
```

```
        break;
```

case 17://o

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_1,GPIO_PIN_SET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

```
case 18://n
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_10,GPIO_PIN_SET);
```

```
break;
```

```
case 19://f
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_2,GPIO_PIN_SET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_SET);
```

```
break;
```

```
case 20://bos
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_RESET);HAL_GPIO
_WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,G
PIO_PIN_9,GPIO_PIN_RESET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_RESET);HAL_GPIO
_WritePin(GPIOD,GPIO_PIN_8,GPIO_PIN_RESET);HAL_GPIO_WritePin(GPIO
D,GPIO_PIN_6,GPIO_PIN_RESET);
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_RESET);HAL_GPI
O_WritePin(GPIOD,GPIO_PIN_11,GPIO_PIN_RESET);
```

```
break;
```

```
case 21://L
```

```
HAL_GPIO_WritePin(GPIOD,GPIO_PIN_9,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_8,GPIO_PIN_SET);
```

```
break;
```

```
case 22://S
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_9,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
    break;
```

```
case 23://P
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_5,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_8,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);
```

```
    break;
```

```
case 24://H
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_1,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);HAL_GPIO_WritePin(GPIOD,GP
IO_PIN_8,GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_6,GPIO_PIN_SET);HAL_GPIO_
WritePin(GPIOD,GPIO_PIN_10,GPIO_PIN_SET);
```

```
    break;
```

```
    }
```

```
}
```

Seg_Goster Fonksiyonu

```
void Seg_Goster1(void)

{

    GostergeOutput(gosterge1);

    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_3,GPIO_PIN_RESET);

    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_7,GPIO_PIN_SET);

    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_4,GPIO_PIN_SET);

    if ((FLAG.index_Point_Flag == 1)&&(No_Value==0))

    {

        GPIOD->ODR ^= 0x0800;

    }

}

void Seg_Goster2(void)

{

    GostergeOutput(gosterge2);

    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_4,GPIO_PIN_RESET);

    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_7,GPIO_PIN_SET);

    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_3,GPIO_PIN_SET);

    if ((FLAG.index_Point_Flag == 2)&&(No_Value==0))

    {

        GPIOD->ODR ^= 0x0800;

    }

}
```

```
void Seg_Goster3(void)
```

```
{  
  
    GostergeOutput(gosterge3);  
  
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_7,GPIO_PIN_RESET);  
  
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_4,GPIO_PIN_SET);  
  
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_3,GPIO_PIN_SET);  
  
    if ((FLAG.index_Point_Flag == 3)&&(No_Value==0))  
  
    {  
  
        GPIOD->ODR ^= 0x0800;  
  
    }  
  
}
```

Bluetooth ve Wifi Haberleşme Fonksiyonu

```
void ESP_Connection(void)  
  
{  
  
    COUNTER.EspCounter++;  
  
    if(COUNTER.EspCounter>=20000)  
  
    {  
  
        COUNTER.EspCounter = 0;  
  
        WIFI_AT_Response();  
  
        //BLE_AT_Response();  
  
    }  
  
}
```

WIFI_AT SEND Fonksiyonu

```
void WIFI_AT_Send(void)
{
    switch(receive_queue_wifi.step_wifi)
    {
    case 0:
        if(!receive_queue_wifi.data_sended_wifi)
        {
            if(!FLAG.ExternalBaudRateFlag)
            {
                WIFI_Transmit("AT+RESTORE\r\n");
            }else
            {
                WIFI_Transmit("AT\r\n");
            }
            receive_queue_wifi.data_sended_wifi = 1;
        }
        break;
    case 1:
        if(!receive_queue_wifi.data_sended_wifi)
        {
            WIFI_Transmit("AT+CIPMUX=1\r\n");
            receive_queue_wifi.data_sended_wifi = 1;
        }
    }
```

```
}
```

```
break;
```

```
case 2:
```

```
if(!receive_queue_wifi.data_sended_wifi)
```

```
{
```

```
WIFI_Transmit("ATE0\r\n");
```

```
receive_queue_wifi.data_sended_wifi = 1;
```

```
}
```

```
break;
```

```
case 3:
```

```
if(!receive_queue_wifi.data_sended_wifi)
```

```
{
```

```
WIFI_Transmit("AT+RST\r\n");
```

```
receive_queue_wifi.data_sended_wifi = 1;
```

```
}
```

```
break;
```

```
case 4:
```

```
if(!receive_queue_wifi.data_sended_wifi)
```

```
{
```

```
WIFI_Transmit("AT+CWMODE=3\r\n");
```

```
receive_queue_wifi.data_sended_wifi = 1;
```

```
}
```

```
break;
```

```
case 5:
```

```

if(!receive_queue_wifi.data_sended_wifi)

{

    WIFI_Transmit("AT+UART_DEF=115200,8,1,0,1\r\n");

    receive_queue_wifi.data_sended_wifi = 1;

}

break;

case 6:

if(!receive_queue_wifi.data_sended_wifi)

{

    WIFI_Transmit("AT+CIPSTA=\"192.168.1.15\"\r\n");

    receive_queue_wifi.data_sended_wifi = 1;

}

break;

case 7:

if(!receive_queue_wifi.data_sended_wifi)

{

WIFI_Transmit("AT+CWJAP=\"MODEM_ADI\", \"MODEM_SIFRE\"\r\n");

    receive_queue_wifi.data_sended_wifi = 1;

}

break;

case 8:

if(!receive_queue_wifi.data_sended_wifi)

{

    WIFI_Transmit("AT+CIPSTART=\"TCP\", \"192.168.1.52\", 23\r\n");

```

```
receive_queue_wifi.data_sended_wifi = 1;
```

```
}  
  
break;  
  
case 9:  
  
if(!receive_queue_wifi.data_sended_wifi)  
  
{  
  
if(ConnectionInfo>=1000)//30 saniyede bir bagli mi degil mi kontrolü  
  
{  
  
ConnectionInfo = 0;  
  
WIFI_Transmit("AT+CIPSTATUS\r\n");  
  
receive_queue_wifi.data_sended_wifi = 1;  
  
}  
  
}  
  
break;  
  
}  
  
}
```

WIFI AT RESPONSE Fonksiyonu

```
void WIFI_AT_Response(void)  
  
{  
  
if(ConnectionInfoFlag)  
  
{  
  
ConnectionInfo++;  
  
}  
  
}
```

```
if((strstr((char*)Rx_data, "ready"))&&(receive_queue_wifi.step_wifi!=99))
```

```
{  
  
    receive_queue_wifi.data_sended_wifi = 0;  
  
    memset(Rx_data,0,255);  
  
    myindex=0;  
  
}  
  
else if(strstr((char*)Rx_data, "+CIPSTATUS"))  
  
{  
  
    if(strstr((char*)Rx_data, "STATUS:2"))  
  
    {  
  
        receive_queue_wifi.data_sended_wifi = 0;  
  
        memset(Rx_data,0,255);  
  
        myindex=0;  
  
    }  
  
    else if(strstr((char*)Rx_data, "STATUS:3"))  
  
    {  
  
        receive_queue_wifi.data_sended_wifi = 0;  
  
        memset(Rx_data,0,255);  
  
        myindex=0;  
  
    }  
  
    else if(strstr((char*)Rx_data, "STATUS:0"))  
  
    {  
  
        receive_queue_wifi.step_wifi = 0;  
  
        receive_queue_wifi.data_sended_wifi = 0;  
  
    }  
  
}
```

```
memset(Rx_data,0,255);
```

```
myindex=0;
```

```
Wifi_Status=0;
```

```
No_Value=2;
```

```
}
```

```
else if(strstr((char*)Rx_data, "STATUS:4"))
```

```
{
```

```
receive_queue_wifi.step_wifi = 0;
```

```
receive_queue_wifi.data_sended_wifi = 0;
```

```
memset(Rx_data,0,255);
```

```
myindex=0;
```

```
Wifi_Status=0;
```

```
No_Value=2;
```

```
}
```

```
else if(strstr((char*)Rx_data, "STATUS:5"))
```

```
{
```

```
receive_queue_wifi.step_wifi = 0;
```

```
receive_queue_wifi.data_sended_wifi = 0;
```

```
memset(Rx_data,0,255);
```

```
myindex=0;
```

```
Wifi_Status=0;
```

```
No_Value=2;
```

```
}
```

```
else if(strstr((char*)Rx_data, "ERROR"))
```

```

{
    receive_queue_wifi.step_wifi = 0;

    receive_queue_wifi.data_sended_wifi = 0;

    memset(Rx_data,0,255);

    myindex=0;

    Wifi_Status=0;

    No_Value=2;

}

}

else if(strstr((char*)Rx_data, "SEND FAIL"))

{

    //FirstConnection = 2;

    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_10,GPIO_PIN_RESET);

    receive_queue_wifi.step_wifi = 0;

    receive_queue_wifi.data_sended_wifi = 0;

    memset(Rx_data,0,255);

    myindex=0;

    FirstConnection=0;

    No_Value=2;

}

else if(strstr((char*)Rx_data, "ERROR"))

{

    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_10,GPIO_PIN_RESET);

    memset(Rx_data,0,255);

```

```
myindex=0;
```

```
    receive_queue_wifi.step_wifi = 0;

    receive_queue_wifi.data_sended_wifi = 0;

    FirstConnection=0;

}

else if(strstr((char*)Rx_data, "CLOSED"))

{

    //FirstConnection = 2;

    FirstConnection=0;

    Wifi_Status=0;

    No_Value=2;

    ConnectionInfoFlag=0;

    receive_queue_wifi.step_wifi = 0;

    receive_queue_wifi.data_sended_wifi = 0;

    memset(Rx_data,0,255);

    myindex=0;

}

else if(strstr((char*)Rx_data,
"+RESTORE")&&(receive_queue_wifi.step_wifi ==
0)||(receive_queue_wifi.step_wifi == 99))

{

    receive_queue_wifi.step_wifi = 99;

    if(strstr((char*)Rx_data, "ready"))

    {

        receive_queue_wifi.step_wifi = 1;
```

```
receive_queue_wifi.data_sended_wifi = 0;
```

```
    memset(Rx_data,0,255);

    myindex=0;

}

else if(strstr((char*)Rx_data, "ERROR"))

{

    receive_queue_wifi.step_wifi = WIFI_ERROR;

}

}

else if(strstr((char*)Rx_data, "AT\r\n") && receive_queue_wifi.step_wifi == 0)

{

    receive_queue_wifi.step_wifi = 1;

    receive_queue_wifi.data_sended_wifi = 0;

    memset(Rx_data,0,255);

    myindex=0;

    if(strstr((char*)Rx_data, "ERROR"))

    {

        receive_queue_wifi.step_wifi = WIFI_ERROR;

    }

}

else if(strstr((char*)Rx_data, "+CIPMUX") && receive_queue_wifi.step_wifi == 1)

{

    if(strstr((char*)Rx_data, "OK"))
```

```
{
```

```
    receive_queue_wifi.step_wifi = 2;
```

```
    receive_queue_wifi.data_sended_wifi = 0;
```

```
    memset(Rx_data,0,255);
```

```
    myindex=0;
```

```
}
```

```
else if(strstr((char*)Rx_data, "ERROR"))
```

```
{
```

```
    receive_queue_wifi.step_wifi = WIFI_ERROR;
```

```
}
```

```
}
```

```
else if(strstr((char*)Rx_data, "ATE0") && receive_queue_wifi.step_wifi ==
```

2)

```
{
```

```
    if(strstr((char*)Rx_data, "OK"))
```

```
{
```

```
    receive_queue_wifi.step_wifi = 3;
```

```
    receive_queue_wifi.data_sended_wifi = 0;
```

```
    memset(Rx_data,0,255);
```

```
    myindex=0;
```

```
}
```

```
else if(strstr((char*)Rx_data, "ERROR"))
```

```
{
```

```
    receive_queue_wifi.step_wifi = WIFI_ERROR;
```

```
}
```

```
}
```

```
    else if(strstr((char*)Rx_data, "+RST") && ((receive_queue_wifi.step_wifi == 3)||
(receive_queue_wifi.step_wifi == 99)))

    {

        receive_queue_wifi.step_wifi = 99;

        if(strstr((char*)Rx_data, "OK"))

        {

            receive_queue_wifi.step_wifi = 4;

            receive_queue_wifi.data_sended_wifi = 0;

            memset(Rx_data,0,255);

            myindex=0;

        }

        else if(strstr((char*)Rx_data, "ERROR"))

        {

            receive_queue_wifi.step_wifi = WIFI_ERROR;

        }

    }

    else if(strstr((char*)Rx_data, "+CWMODE=") &&
receive_queue_wifi.step_wifi == 4)

    {

        if(strstr((char*)Rx_data, "OK"))

        {

            if(!FLAG.ExternalBaudRateFlag)

            {

                receive_queue_wifi.step_wifi = 5;
```

```

    }else

    {

        receive_queue_wifi.step_wifi = 6;

    }

    receive_queue_wifi.data_sended_wifi = 0;

    memset(Rx_data,0,255);

    myindex=0;

}

else if(strstr((char*)Rx_data, "ERROR"))

{

    receive_queue_wifi.step_wifi = WIFI_ERROR;

}

}

else if(strstr((char*)Rx_data, "+UART_DEF=") &&
receive_queue_wifi.step_wifi == 5)

{

    if(strstr((char*)Rx_data, "OK"))

    {

        FLAG.ExternalBaudRateFlag=1;

        writeFRAM_signedInteger(1,256);//dummy
writeFRAM_signedInteger(FLAG.ExternalBaudRateFlag,BAUDRATE_ADDRESS)
;

        HAL_NVIC_SystemReset();

    }

    else if(strstr((char*)Rx_data, "ERROR"))

```

```

    {

        receive_queue_wifi.step_wifi = WIFI_ERROR;

    }

}

else if(strstr((char*)Rx_data, "+CIPSTA=") &&
receive_queue_wifi.step_wifi == 6)

{

    if(strstr((char*)Rx_data, "OK"))

    {

        receive_queue_wifi.step_wifi = 7;

        receive_queue_wifi.data_sended_wifi = 0;

        memset(Rx_data,0,255);

        myindex=0;

    }

    else if(strstr((char*)Rx_data, "ERROR"))

    {

        receive_queue_wifi.step_wifi = WIFI_ERROR;

    }

}

else if(strstr((char*)Rx_data, "+CWJAP") && receive_queue_wifi
== 7)

{

    if(strstr((char*)Rx_data, "WIFI GOT IP"))

    {

        if(strstr((char*)Rx_data, "OK"))

```

```

{
    receive_queue_wifi.step_wifi = 8;

    receive_queue_wifi.data_sended_wifi = 0;

    memset(Rx_data,0,255);

    myindex=0;

}

}

else if(strstr((char*)Rx_data, "WIFI CONNECTED"))
{
    if(strstr((char*)Rx_data, "OK"))
    {
        receive_queue_wifi.step_wifi = 8;

        receive_queue_wifi.data_sended_wifi = 0;

        memset(Rx_data,0,255);

        myindex=0;

    }

}

else if(strstr((char*)Rx_data, "ERROR"))
{
    receive_queue_wifi.step_wifi = WIFI_ERROR;

}

}

else if(strstr((char*)Rx_data, "+CIPSTART=\"TCP\"") &&
receive_queue_wifi.step_wifi == 8)
{

```

```
if(strstr((char*)Rx_data, "CONNECT"))
```

```
{
```

```
if(strstr((char*)Rx_data, "OK"))
```

```
{
```

```
receive_queue_wifi.step_wifi = 9;
```

```
receive_queue_wifi.data_sended_wifi = 0;
```

```
memset(Rx_data,0,255);
```

```
myindex=0;
```

```
Wifi_Status=1;
```

```
ConnectionInfoFlag = 1;
```

```
No_Value=1;
```

```
}
```

```
}
```

```
else if(strstr((char*)Rx_data, "ERROR"))
```

```
{
```

```
receive_queue_wifi.step_wifi = WIFI_ERROR;
```

```
}
```

```
}else if(strstr((char*)Rx_data, "WIFI GOT IP"))
```

```
{
```

```
receive_queue_wifi.step_wifi = 1;
```

```
receive_queue_wifi.data_sended_wifi = 0;
```

```
memset(Rx_data,0,255);
```

```
myindex=0;
```

```
}else if((strstr((char*)Rx_data, "WIFI DISCONNECT")) && Wifi_Status)
```

```
{
```

```
    HAL_GPIO_WritePin(GPIOA,GPIO_PIN_10,GPIO_PIN_RESET);
```

```
    memset(Rx_data,0,255);
```

```
    myindex=0;
```

```
    receive_queue_wifi.step_wifi = 0;
```

```
    receive_queue_wifi.data_sended_wifi = 0;
```

```
    FirstConnection=0;
```

```
    Wifi_Status=0;
```

```
    ConnectionInfoFlag=0;
```

```
    No_Value=2;
```

```
}else
```

```
{
```

```
    if(((!receive_queue_wifi.data_sended_wifi)|| (receive_queue.ready_to_write))
```

```
    {
```

```
        if(strstr((char*)Rx_data, "+IPD"))
```

```
        {
```

```
            //receive_queue_wifi.step = 9;
```

```
            receive_queue_wifi.data_sended_wifi = 0;
```

```
            memset(Rx_data,0,255);
```

```
            myindex=0;
```

```
        }
```

```
    }
```

```
}
```

```
    if(((!receive_queue_wifi.data_sended_wifi) && (receive_queue_wifi.step_wifi  
!= 99))
```

```
{
```

```
    WIFI_AT_Send();
```

```
}
```

```
}
```

Bluetooth Haberleşme Fonksiyonu

```
void BLE_AT_Send(void)
```

```
{
```

```
    switch(receive_queue.step)
```

```
    {
```

```
    case 1:
```

```
        if(!receive_queue.data_sended)
```

```
        {
```

```
            BLE_Transmit("AT+BLEINIT=2\r\n");
```

```
            receive_queue.data_sended = 1;
```

```
        }
```

```
        break;
```

```
    case 2:
```

```
        if(!receive_queue.data_sended)
```

```
        {
```

```
            BLE_Transmit("AT+BLENNAME=\"MILKMETER1\"\r\n");
```

```
            receive_queue.data_sended = 1;
```

```
        }
```

```
        break;
```

```
    case 3:
```

```

if(!receive_queue.data_sended)

{

    BLE_Transmit("AT+BLEGATTSSRVCRE\r\n");

    receive_queue.data_sended = 1;

}

break;

case 4:

if(!receive_queue.data_sended)

{

    BLE_Transmit("AT+BLEGATTSSRVSTART\r\n");

    receive_queue.data_sended = 1;

}

break;

case 5:

if(!receive_queue.data_sended)

{

    BLE_Transmit("AT+BLEADVSTART\r\n");

    receive_queue.data_sended = 1;

}

break;

case 6:

if(!receive_queue.data_sended)

{

    BLE_Transmit("AT+BLEGATTCHAR?\r\n");

```

```

        receive_queue.data_sended = 1;
    }

    break;

case 7:

    if(!receive_queue.data_sended)

    {

        BLE_Transmit("AT+BLEADDR?\r\n");

        receive_queue.data_sended = 1;

    }

    break;

case 8:

    BLE_Status = 1;

    //BLE_Transmit("AT+BLEADDR?\r\n");

    //receive_queue.data_sended = 1;

    break;

case 9:

    if(!receive_queue.data_sended)

    {

        BLE_Transmit("AT+BLECFGMTU=0,20\r\n");

        receive_queue.data_sended = 1;

    }

    break;

case 10:

    //if(!receive_queue.data_sended)

```

```
{
```

```
    BLE_Transmit("AT+BLEADVSTART\r\n");//Tekrar baglanilmasi için,  
tekrardan baglanti kurulmasi gerekiyor
```

```
    receive_queue.data_sended = 1;
```

```
    BLE_Status = 2;
```

```
}
```

```
break;
```

```
case 12:
```

```
    FLAG.index_Point_Flag=0;
```

```
    gosterge1=10;//E
```

```
    gosterge2=0;//0
```

```
    gosterge3=2;//2
```

```
    break;
```

```
case 15:
```

```
    if(!receive_queue.data_sended)
```

```
    {
```

```
        //BLE_Transmit("AT+BLEGATTSIND=0,1,7,20\r\n");
```

```
        receive_queue.data_sended = 1;
```

```
    }
```

```
    break;
```

```
case 16:
```

```
    if(!receive_queue.data_sended)
```

```
    {
```

```
        receive_queue.data_sended = 1;
```

```
    }
```

```
break;
```

```
case 17:
```

```
    DataSended=1;
```

```
    break;
```

```
    }
```

```
}
```

BLE AT RESPONSE Fonkiyonu

```
void BLE_AT_Response(void)
```

```
{
```

```
    if(strstr((char*)Rx_data, "ready") && receive_queue.step == 0)
```

```
    {
```

```
        memset(Rx_data,0,255);
```

```
        myindex=0;
```

```
        receive_queue.step = 1;
```

```
        receive_queue.data_sended = 0;
```

```
    }
```

```
    else if(strstr((char*)Rx_data, "+RST"))
```

```
    {
```

```
        if(strstr((char*)Rx_data, "OK"))
```

```
        {
```

```
            if(strstr((char*)Rx_data, "ready"))
```

```
            {
```

```
                receive_queue.step = BLE_SERVER;
```

```
            }
```

```
}
```

```
else if(strstr((char*)Rx_data, "ERROR"))
{
    receive_queue.step = BLE_ERROR;
}
}

else if(strstr((char*)Rx_data, "+BLEINIT=2") && receive_queue.step == 1)
{
    if(strstr((char*)Rx_data, "OK"))
    {
        receive_queue.step = 2;
        receive_queue.data_sended = 0;
        memset(Rx_data,0,255);
        myindex=0;
    }

    else if(strstr((char*)Rx_data, "ERROR"))
    {
        receive_queue.step = BLE_ERROR;
    }
}

else if(strstr((char*)Rx_data, "+BLENAME") && receive_queue.step == 2)
{
    if(strstr((char*)Rx_data, "OK"))
    {
```

```

receive_queue.step = 3;

receive_queue.data_sended = 0;

memset(Rx_data,0,255);

myindex=0;
}

else if(strstr((char*)Rx_data, "ERROR"))
{
    receive_queue.step = BLE_ERROR;
}
}

else if(strstr((char*)Rx_data, "+BLEGATTSSRVCRE") &&
receive_queue.step == 3)
{
    if(strstr((char*)Rx_data, "OK"))
    {
        receive_queue.step = 4;

        receive_queue.data_sended = 0;

        memset(Rx_data,0,255);

        myindex=0;
    }

    else if(strstr((char*)Rx_data, "ERROR"))
    {
        receive_queue.step = BLE_ERROR;
    }
}
}

```

```

        else if(strstr((char*)Rx_data, "+BLEGATTSSRVSTART") &&
receive_queue.step == 4)

        {

            if(strstr((char*)Rx_data, "OK"))

            {

                receive_queue.step = 5;

                receive_queue.data_sended = 0;

                memset(Rx_data,0,255);

                myindex=0;

            }

            else if(strstr((char*)Rx_data, "ERROR"))

            {

                receive_queue.step = BLE_ERROR;

            }

        }

        else if(strstr((char*)Rx_data, "+BLEADVSTART") && (receive_queue.step
== 5 || receive_queue.step == 10))

        {

            if(strstr((char*)Rx_data, "OK"))

            {

                receive_queue.step = 8;

                receive_queue.data_sended = 0;

                memset(Rx_data,0,255);

                myindex=0;

            }

        }

```

```

else if(strstr((char*)Rx_data, "ERROR"))

{

    receive_queue.step = BLE_ERROR;

}

}

else if(strstr((char*)Rx_data, "+BLEGATTCHAR?") &&
receive_queue.step == 6)

{

    if(strstr((char*)Rx_data, "OK"))

    {

        //receive_queue.step = BLE_START;

    }

    else if(strstr((char*)Rx_data, "ERROR"))

    {

        receive_queue.step = BLE_ERROR;

    }

}

else if(strstr((char*)Rx_data, "+BLEADDR?") && receive_queue.step == 7)

{

    if(strstr((char*)Rx_data, "+BLEADDR:"))

    {

        if(strstr((char*)Rx_data, "OK"))

        {

        }

    }

}

```

```

else if(strstr((char*)Rx_data, "ERROR"))

{

    receive_queue.step = BLE_ERROR;

}

}else

{

    if((!receive_queue.data_sended)|| (receive_queue.ready_to_write))

    {

        if(strstr((char*)Rx_data, "+BLECONN:")||strstr((char*)Rx_data,
"+BLECONNPARAM:"))

        {

            memset(Rx_data,0,255);

            myindex=0;

            receive_queue.step = BLE_CONNECTED;

            receive_queue.data_sended = 0;

        }

        else if(strstr((char*)Rx_data, "+BLEDISCONN:"))

        {

            FirstConnection=0;

            receive_queue.data_sended = 0;

            receive_queue.step = 0;

            receive_queue.data_sended = 0;

            memset(Rx_data,0,255);

            myindex=0;

            BLE_Transmit("AT+BLEADVSTART\r\n");

```

```

    }

    else if(strstr((char*)Rx_data, "+READ:"))

    {

        receive_queue.data_sended = 0;

        memset(Rx_data,0,255);

        myindex=0;

    }

    else if(strstr((char*)Rx_data, ">"))

    {

        //receive_queue.ready_to_write = 0;

        //receive_queue.step = 16;

        receive_queue.ready_to_write = 1;

        receive_queue.data_sended = 1;

        memset(Rx_data,0,255);

        myindex=0;

    }

    else if(strstr((char*)Rx_data, "+WRITE:"))

    {

        if(strstr((char*)Rx_data,"1,7,1,2,")//STX - indicator

        {

            receive_queue.ready_to_write = 1;

            No_Value=1;

            receive_queue.step = 15;

            receive_queue.data_sended = 0;

```

```

        memset(Rx_data,0,255);

        myindex=0;

HAL_GPIO_WritePin(GPIOA,GPIO_PIN_9,GPIO_PIN_SET);//STATUS LED 2

        BLE_Transmit("AT+BLEGATTSIND=0,1,7,20\r\n");

    }

}

else if(strstr((char*)Rx_data, "+BLECFGMTU:"))

{

    //receive_queue.step = 9;

    BLE_Transmit("AT+BLECFGMTU=0,20\r\n");

    receive_queue.data_sended = 0;

    memset(Rx_data,0,255);

    myindex=0;

}

else if(strstr((char*)Rx_data, "+BLECFGMTU?"))

{

    receive_queue.data_sended = 0;

    memset(Rx_data,0,255);

    myindex=0;

}

else if(strstr((char*)Rx_data, "OK"))

{

    //receive_queue.step = BLE_DATA_SENDED;

}

```

```

        else if(strstr((char*)Rx_data, "ERROR"))
        {
            //receive_queue.step = BLE_ERROR;

        }
    }

    if(!receive_queue.data_sended)
    {
        BLE_AT_Send();
    }
}

```

BLE Transmit Fonksiyonu

```

void BLE_Transmit(unsigned char* data)
{
    if(data == NULL)
    {
        return;
    }

    //data = (uint8_t*) receive_queue.write_data;

    HAL_UART_DMAStop(&huart7);

    HAL_UART_Transmit_DMA(&huart7,(uint8_t*)data,strlen(data));
}

```

WIFI Transmit Fonksiyonu

Bu fonksiyon ile wifi komutları sunucuya gönderilmektedir.

```
void WIFI_Transmit(unsigned char* data)
{
    if(data == NULL)
    {
        return;
    }

    //data = (uint8_t*) receive_queue.write_data;

    //UART_DE_ENABLE;

    HAL_UART_DMAStop(&huart7);

    HAL_UART_Transmit_DMA(&huart7,(uint8_t*)data,strlen(data));
}
```

Callback Fonksiyonu

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if(huart->Instance == UART7)
    {
        Rx_data[myindex++] = myData[0];

        /*if(myData[0]=='>')
        {
            receive_queue.ready_to_write = 1;
        }*/
    }
}
```

```

        HAL_UART_Receive_IT(&huart7,(uint8_t* ) myData,1);

    }

}

/*void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart,
uint16_t Size)

{

    if(huart->Instance == UART7)

    {

        HAL_UARTEx_ReceiveToIdle_DMA(&huart7, Rx_data, 255);

        __HAL_DMA_DISABLE_IT(&hdma_uart7_rx,DMA_IT_HT);

    }

}*/

```

İletkenlik Fonksiyonu

```

void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)//İletkenlik

{

    if((htim->Channel == HAL_TIM_ACTIVE_CHANNEL_3) || (htim-
>Channel == HAL_TIM_ACTIVE_CHANNEL_4))

    {

        if(FLAG.IletkenlikGirisFlag==0)

        {

            if(htim->Channel == HAL_TIM_ACTIVE_CHANNEL_3)

            {

                IC_ValueGiris1 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_3);

```

```

        }else

        {

            IC_ValueGiris1 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_4);

        }

        FLAG.IletkenlikGirisFlag = 1;

    }

    else if(FLAG.IletkenlikGirisFlag == 1)

    {

        if(htim->Channel == HAL_TIM_ACTIVE_CHANNEL_3)

        {

            IC_ValueGiris2 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_3);

        }else

        {

            IC_ValueGiris2 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_4);

        }

        if(IC_ValueGiris2 > IC_ValueGiris1)

        {

            DifferenceGiris = IC_ValueGiris2-IC_ValueGiris1;

        }else if(IC_ValueGiris2 < IC_ValueGiris1)

        {

            DifferenceGiris = ((0xffffffff - IC_ValueGiris1) + IC_ValueGiris2);

        }

    }

```

```

else

{

    Error_Handler();

}

float refClock = TIMCLOCK/(PRESCALAR);

if(DifferenceGiris<110&&DifferenceGiris>0)

FrequencyGiris = (refClock/DifferenceGiris);

if(FrequencyGiris>HighValue)

{

    //FLAG.TotalTimeFlag=1;

    //FLAG.LedCloseFlag=1;

    OpenIRLed();

    FLAG.MilkTimeSayacFlag=1;

    FLAG.SpeedCalculateStartFlag=0;

    COUNTER.LedCloseCounter=0;

    if(FrequencyCikis<HighValue)

    {

        FLAG.MilkTransitTimeFlag = 0;

        MilkTransitTime = 0;

    }

}

}else

{

    FLAG.MilkTimeSayacFlag = 0;

}

}

```

```

        FLAG.IletkenlikGirisFlag = 0;

    }

}

else if((htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1) || (htim-
>Channel == HAL_TIM_ACTIVE_CHANNEL_2))

{

    if(FLAG.IletkenlikCikisFlag==0)

    {

        if(htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1)

        {

            IC_ValueCikis1 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_1);

        }else

        {

            IC_ValueCikis1 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_2);

        }

        FLAG.IletkenlikCikisFlag = 1;

    }

    else if(FLAG.IletkenlikCikisFlag == 1)

    {

        if(htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1)

        {

            IC_ValueCikis2 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_1);

        }else

```

```

{
    IC_ValueCikis2 = HAL_TIM_ReadCapturedValue(&htim1,
TIM_CHANNEL_2);

}

if(IC_ValueCikis2 > IC_ValueCikis1)

{
    DifferenceCikis = IC_ValueCikis2-IC_ValueCikis1;
}
else if(IC_ValueCikis2 < IC_ValueCikis1)

{
    DifferenceCikis = ((0xffffffff-IC_ValueCikis1)+IC_ValueCikis2);
}

else

{ Error_Handler();

}

float refClock2 = TIMCLOCK/(PRESCALAR);

if(DifferenceCikis<110&&DifferenceCikis>0)

FrequencyCikis = refClock2/DifferenceCikis;

if(FrequencyCikis>HighValue)

{ if(!FLAG.MilkTransitTimeFlag)

{ FLAG.MilkTimeSayacFlag=0;

MilkTransitTime = MilkTransitTimeTemp / 10000;

//MilkSpeed = PinDistance / MilkTransitTime;// metre/saniye - mm/ms

MilkFrequency = FrequencyGiris;

if(MilkFrequencyLast < MilkFrequency)

{

```

```

        MilkFrequencyLast = MilkFrequency;

        MilkFrequencyBolum = MilkFrequencyLast / 255;

        if(MilkFrequencyBolum>0)

        {

            MilkFrequencyKalan = MilkFrequencyLast -
(MilkFrequencyBolum*255);

            writeFRAM_signedInteger(MilkFrequencyBolum,MILK_FREQUENCY_AD
DRESS_BOLUM);

            writeFRAM_signedInteger(MilkFrequencyKalan,MILK_FREQUENCY_ADD
RESS_KALAN);

        }

        else

        {

            writeFRAM_signedInteger(MilkFrequencyLast,MILK_FREQUENCY_ADDR
ESS);

        }

        /*MilkFiveLedTime = MilkTransitTime * (FiveLedDistance /
PinDistance);

        SampleNumber = MilkFiveLedTime * 1000 ; //mm basina örnek sayısı*/

        FLAG.MilkTransitTimeFlag = 1;

    }

    MilkTransitTimeTemp = 0;

}

FLAG.IletkenlikCikisFlag = 0;

}

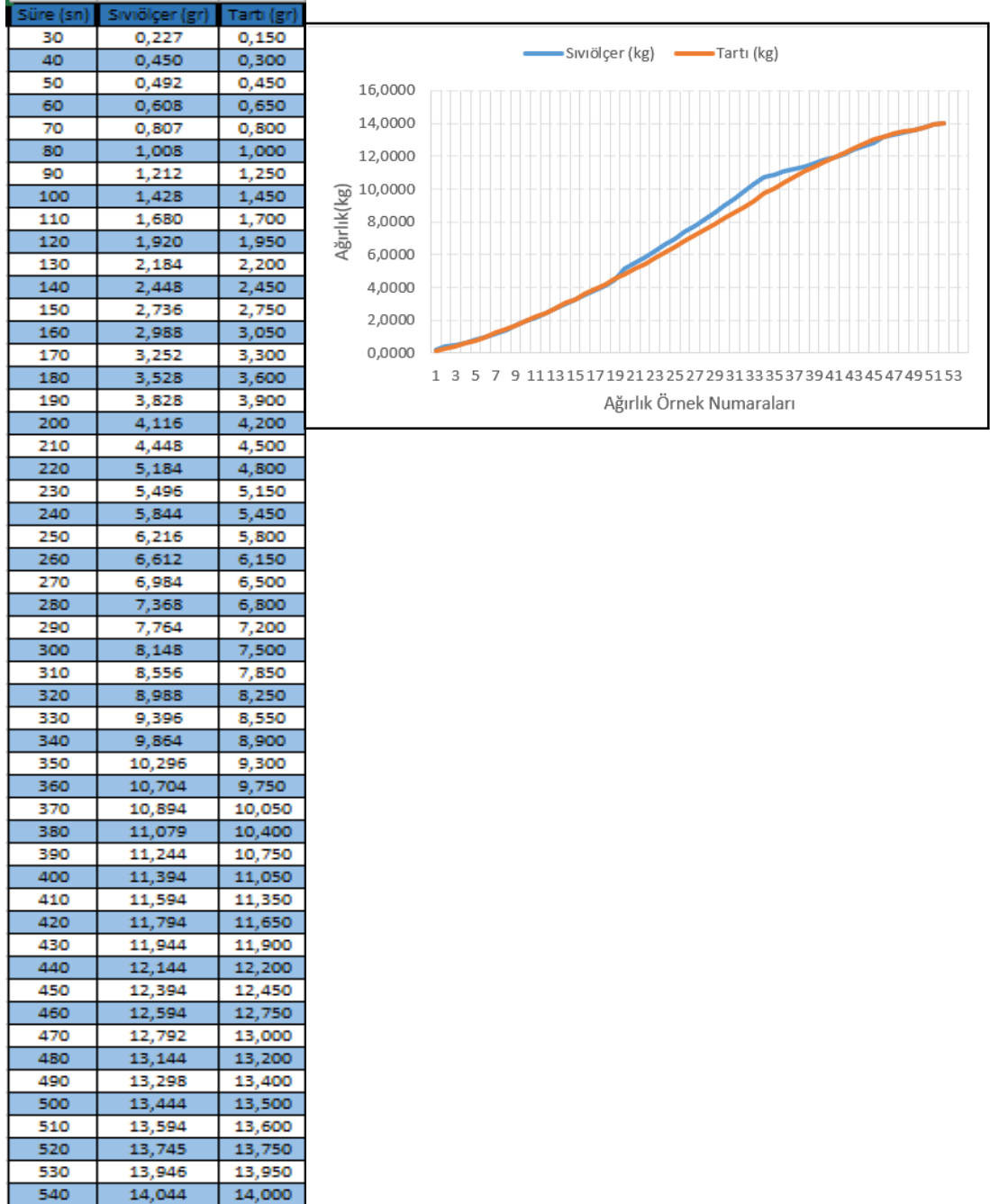
}

```

EK-E Test Verileri

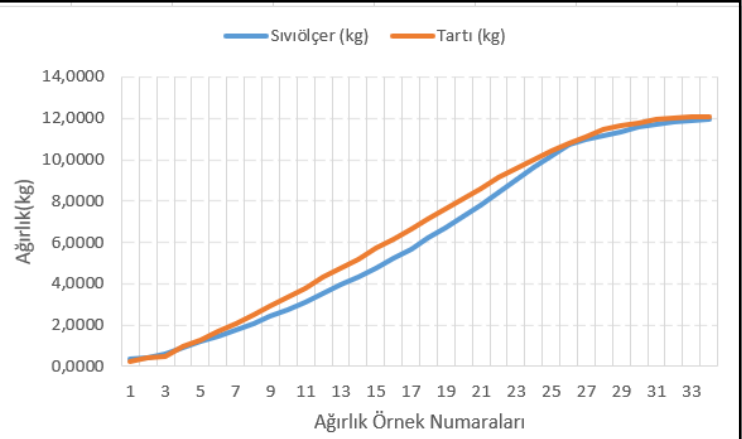
İlk iki test verisi daha önce test sonuçları başlığı altında verilmiştir. Bu bölümde daha fazla test verisi paylaşılması için üçüncü testten başlanılmıştır.

Test 3 Verileri ve Grafiği



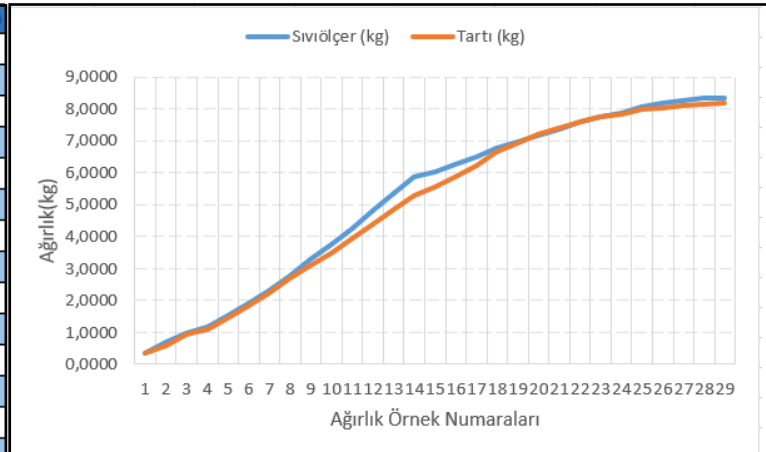
Test 4 Verileri ve Grafiği

Süre (sn)	Sıvıölçer (gr)	Tartı (gr)
30	0,36	0,250
40	0,4068	0,400
50	0,5976	0,500
60	0,9372	0,950
120	1,2	1,300
130	1,476	1,700
140	1,776	2,050
150	2,1	2,500
160	2,424	2,900
170	2,772	3,350
180	3,132	3,800
190	3,528	4,300
200	3,936	4,750
210	4,344	5,200
220	4,776	5,750
230	5,232	6,150
240	5,7	6,650
250	6,192	7,150
260	6,72	7,600
270	7,26	8,100
280	7,812	8,600
290	8,436	9,150
300	9,048	9,550
310	9,612	10,000
320	10,176	10,400
330	10,74	10,800
340	10,96	11,100
350	11,18	11,450
360	11,365	11,650
370	11,565	11,800
380	11,715	11,950
390	11,815	12,000
400	11,915	12,050
410	11,94	12,100



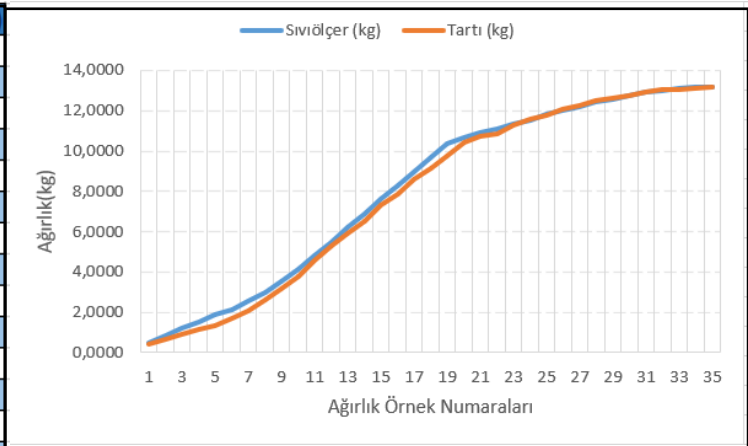
Test 5 Verileri ve Grafiği

Süre (sn)	Sıvıölçer (gr)	Tartı (gr)
30	0,359	0,350
40	0,687	0,600
50	0,9984	0,950
60	1,164	1,100
70	1,512	1,450
80	1,922	1,850
90	2,328	2,250
100	2,796	2,700
110	3,276	3,100
120	3,768	3,500
130	4,284	3,950
140	4,836	4,400
150	5,388	4,850
160	5,868	5,300
170	6,053	5,550
180	6,283	5,900
190	6,516	6,250
200	6,766	6,650
210	6,983	6,950
220	7,188	7,200
230	7,378	7,400
240	7,588	7,600
250	7,748	7,750
260	7,893	7,850
270	8,058	8,000
280	8,178	8,050
290	8,283	8,100
300	8,343	8,150
310	8,352	8,200



Test 6 Verileri ve Grafiği

Süre (sn)	Sıvıölçer (gr)	Tartı (gr)
30	0,5016	0,400
40	0,8748	0,650
50	1,236	0,900
60	1,548	1,150
70	1,86	1,350
80	2,16	1,700
90	2,56	2,100
100	3	2,600
110	3,54	3,150
120	4,152	3,800
130	4,836	4,600
140	5,496	5,300
150	6,228	5,900
160	6,868	6,550
170	7,596	7,300
180	8,28	7,850
190	8,988	8,600
200	9,708	9,150
210	10,37	9,750
220	10,65	10,450
230	10,89	10,750
240	11,105	10,850
250	11,355	11,300
260	11,55	11,600
270	11,805	11,750
280	12,005	12,050
290	12,208	12,250
300	12,466	12,500
310	12,568	12,650
320	12,755	12,750
330	12,905	12,900
340	13,002	13,050
350	13,105	13,050
360	13,156	13,100
370	13,18	13,150



Test 7 Verileri ve Grafiği

Süre (sn)	Sıvölçer (gr)	Tartı (gr)
30	0,448	0,400
40	0,717	0,700
50	1,147	0,900
60	1,356	1,200
70	1,628	1,550
80	1,998	2,150
90	2,364	2,550
100	2,748	3,050
110	3,204	3,700
120	3,684	4,350
130	4,200	4,900
140	4,764	5,450
150	5,340	6,150
160	5,928	6,700
170	6,516	7,300
180	7,128	7,900
190	7,740	8,500
200	8,380	9,150
210	8,976	9,700
220	9,688	10,250
230	10,198	10,750
240	10,736	11,150
250	11,266	11,500
260	11,784	11,900
270	12,240	12,200
280	12,720	12,550
290	12,920	12,800
300	13,070	12,900
310	13,170	13,050
320	13,320	13,150
330	13,420	13,200
340	13,495	13,300

