T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YENİ BİR KAOS TABANLI RASGELE SAYI ÜRETECİ KULLANAN BANKA ŞİFREMATİK CİHAZI TASARIMI VE UYGULAMASI

YÜKSEK LİSANS TEZİ

Serkan AKKAYA

Enstitü Anabilim Dalı	**:**	**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**
Enstitü Bilim Dalı	**:**	**ELEKTRİK**
Tez Danışmanı	**:**	**Doç. Dr. İhsan PEHLİVAN**

Mayıs 2016

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YENİ BİR KAOS TABANLI RASGELE SAYI ÜRETECİ KULLANAN BANKA ŞİFREMATİK CİHAZI TASARIMI VE UYGULAMASI

YÜKSEK LİSANS TEZİ

Serkan AKKAYA

Enstitü Anabilim Dalı	:	**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**
Enstitü Bilim Dalı	:	**ELEKTRİK**

Bu tez 31/05/2016 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

Doç. Dr.	**Doç. Dr.**	**Yrd. Doç. Dr.**
İhsan PEHLİVAN	**Ahmet ZENGİN**	**Enver ÇAVUŞ**
Jüri Başkanı	**Üye**	**Üye**

BEYAN

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

Serkan AKKAYA

31.05.2016

TEŞEKKÜR

Tez çalışmam süresince bana her türlü desteği veren değerli hocam Doç. Dr. İhsan PEHLİVAN'a, bu tezin ortaya çıkmasında değerli tecrübelerini benimle paylaşan Yrd. Doç. Dr. Akif AKGÜL'e teşekkürlerimi sunarım.

Maddi ve manevi olarak desteklerini esirgemeyen aileme, biricik oğlum Kerem Ensar'a, tez çalışmam boyunca bana destek veren kıymettar arkadaşım Emre Kırkaya'ya ve emeği geçen tüm arkadaşlarıma ayrıca teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

ADC : Analog Digital Converter
AES : Advanced Encryption Standard
Android : Mobil işletim sistemi
AP : Akgül - Pehlivan
ARM : Acorn RISC Machine
ASIC : Application Specific Integrated Circuits
Asp.Net : Web uygulama gelişim teknolojisi
C : Yazılımsal programlama dili
CAN : Controller Area Network
CCM : Cache Controller and Memory
DAC : Digital Analog Converter
DES : Data Encryption Standard
DGKK : Doğrusal Geri beslemeli Kayan Kaydedici
DMIPS : Dhrystone Millions of Instructions Per Second
DSP : Digital Signal Processors
erfc : The Complementary Error Function
E-posta : Elektronik posta
FIPS : Federal Information Processing Standard
FPGA : Field Programmable Gate Array
FPU : Floating Point Unit
H_a : Alternatif hipotez
H_0 : Sıfır hipotezi
I^2C : Inter-Integrated Circuit
ID : Identity (Kimlik)
IDEA : International Data Encryption Algorithm
IEEE : The Institute of Electrical and Electronical Engineers

IEEE-754 : IEEE Kayan noktalı sayı formatı

IOS : Mobil işletim sistemi

IP : Internet Protocol

I / O : Input / Output

j : Kesir bitlerinin sayısı

J : i. L-bit bloğun onluk sayı sistemindeki değeri

JTAG : Joint Test Action Group

K : Bağımsızlık katsayısı

k : RK algoritmasında hesaplanan değişken

Kbit : Kilobit

KB : Kilobyte

Keylogger : Klavye girdi kaydedicisi

L : Üniversal testinde her bir bloğun uzunluğu

LCD : Liquid Crystal Display

LFRS : Linear Feedback Shift Register

M : Bit dizisinde belirli sayıdaki bitlerinden oluşan blok

m : Örtüşen şablon eşleştirme testinde özel blokların bit sayısı

MAC : Media Access Control

Malware : Zararlı yazılım

MATLAB : Matrix Laboratory

Mbit : Megabit

n : Bit dizisinin uzunluğu

NIST : National Institute of Standards and Technology

o_i : Gözlemlenen frekans

P-değeri : NIST-800-22 testinde rasgelelik ölçütü

Phishing : Balık avı (Yemleme)

Php : Hypertext Preprocessor

PLL : Phase Locked Loop (Faz Kilitlemeli Döngü)

Q : İkili matris derece testinde sütun sayısı

R : Direnç değeri

RK4 : Dördüncü dereceden Runge-Kutta algoritması

RK5 : Beşinci dereceden Runge-Kutta algoritması

RL : Direnç ve Bobinden oluşan Devre

RLC : Direnç, Bobin ve Kondansatörden oluşan Devre

RSA : Ronald, Shamir, Adleman

RSÜ : Rasgele Sayı Üreteci

Screenlogger : Ekran görüntüsü kaydedicisi

SEA : Scalable Encryption Algorithm

sign : İşaret biti

S_n : Normalizasyon işleminden elde edilen değer

sn : Saniye

S_{obs} : Gözlemlenen değer

Spam : Yığın mesaj (Talep olmaksızın gönderilen toplu ileti)

SPI : Serial Peripheral Interface Bus

SRAM : Static Random Access Memory

SSL : Güvenli Soket Katmanı

SWD : Serial Wire Debug

TEA : Tiny Encryption Standard

UART : Universal Asynchronous Receiver / Transmitter

USB : Universal Serial Port

USART : Universal Synchronous / Asynchronous Receiver / Transmitter

V : Gerilim

v : Onluk sayı değeri

V(obs) : Bit osilasyon sayısı

Vi : En uzun 1 dizisinin akış frekansı

Vpn : Virtual private network

Web : World Wide Web

XOR : Exclusive Or (Özel Veya)

XTEA : eXtended Tiny Encryption Standard

y_λ : Algoritma ilk değeri

$y_{\lambda+1}$: Algoritma sonraki değeri

γ : Sistem parametresi

Δh : Algoritma adım miktarı

ε : Bit dizisi

ε'	: Artırım dizisi
εi	: Bit dizisinin i. elemanı
λ	: Öz değerler
λ_σ	: Algoritma parametreleri
μ	: Beklenen değer
μs	: Mikro Saniye
ξ	: Rasgele yürüyüşlerde ziyaret edilen durumların toplam sayısı
ξ_σ	: Algoritma parametreleri
π	: Bit dizisindeki 1 değerlerinin sayısı
σ^2	: Varyans
χ^2	: Ki-kare dağılımı
τ	: Test için gerekli parametre şartı

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Kaos, Kriptoloji, Kaotik Sistemler, Kaos Tabanlı Şifreleme, Rasgele Sayı Üreteci, İstatistiksel Rasgelelik Testleri, NIST Rasgelelik Testi, Şifrematik, İnternet Bankacılığında Güvenlik.

Bu tez çalışmasında, uluslararası en üst standart olan istatiksel NIST-800-22 rasgelelik testlerinden başarı ile geçirilen kaos tabanlı yeni bir Rasgele Sayı Üreteci(RSÜ) yardımı ile şifre üretme algoritması geliştirmek, donanım tabanlı şifrematik uygulaması yapmak ve örnek bir bankacılık sistemi arayüz programı üzerinde test etmek amaçlanmıştır.

Tezin ilk aşamasında, rasgele sayı üretici tarafından oluşturulan bitler vasıtasıyla yeni bir şifre üretme algoritması geliştirilmiştir. Bu algoritma için gerekli olan dönüşüm tablosu hazırlanmıştır. Daha sonra bu algoritma ile öncelikle bilgisayar ortamında şifreler üretilmiştir. Ardından aynı şifrelerin elektronik devre ortamında gerçeklenmesi için, gerekli bitlerin bilgisayar ortamında kodlanması amaçlanmıştır. Sonrasında C dilinde mikrodenetleyicinin yazılımsal olarak programlanması yapılmıştır. Daha sonra bu programın mikrodenteleyici belleğine yüklenerek; LCD panel üzerinde şifrelerin görüntülenmesi amaçlanmıştır. Son olarak şifrematik cihazınca üretilen şifrelerin test edilmesi maksadıyla, örnek bir bankacılık sistemi kullanıcı arayüz programı tasarımı gerçekleştirilmiştir.

Sonuç olarak; geliştirilen şifrematik cihazının yeni ve özgün özellikleri şunlardır: geliştirilen kaos-tabanlı donanımsal şifrematik cihazının literatürde benzerinin görülmemesi, dinamik yapısı karmaşık ve rasgeleliği yüksek bir kaotik sistem içermesi, NIST-800-22 rasgelelik testlerinden geçmiş özgün bir RSÜ bitlerinden üretilen şifrelere sahip olması, kolay bir şifre üretme algoritma yapısına sahip olması, donanım olarak kolay gerçeklenebilir olması, örnek bir bankacılık sistemi kullanıcı arayüz programında etkinliğinin test edilmiş olmasıdır.

THE DESIGN AND APPLICATION OF BANK AUTHENTICATOR DEVICE WITH A NOVEL CHAOS BASED RANDOM NUMBER GENERATOR

SUMMARY

Keywords: Chaos, Cryptology, Chaotic Systems, Chaos Based Encryption, Random Number Generator, Statistical Randomness Tests, NIST Randomness Test, Authenticator, Safety in Internet Banking.

This thesis study aims to develop a password generation algorithm with the help of a new chaos based Random Number Generator (RNG) that has successfully passed statistical NIST-800-22 randomness tests which are the internationally highest standard as well as to conduct a hardware based Authenticator application and to test it on a sample banking system interface program.

For the initial step of the thesis, a new password generation algorithm was developed via the bits generated by random number generator. Conversion table required for this algorithm was prepared. Following this, primarily passwords were generated on computer through this algorithm. Then, necessary bits were aimed to be coded on computer so as to realize the same passwords on electronic environment. Next, software programming of the microcontroller was carried out in C language. As the following step, this program was uploaded to the memory of the microcontroller and thus visualization of passwords on the LCD panel was purposed. Finally, in order to test passwords generated by authenticator device, a sample internet banking user interface program was designed.

In conclusion; the new and unique features of the authenticator device developed here are: there is no other study in the literature to match this newly-developed chaos-based hardware authenticator device; it includes a chaotic system with a complex dynamic structure and high randomness; it contains passwords generated by unique RNG bits that have passed NIST-800-22 randomness tests; it contains an easy password generation algorithm structure; it is easy to be realized in terms of hardware; its efficiency has been tested on a sample banking system interface program.

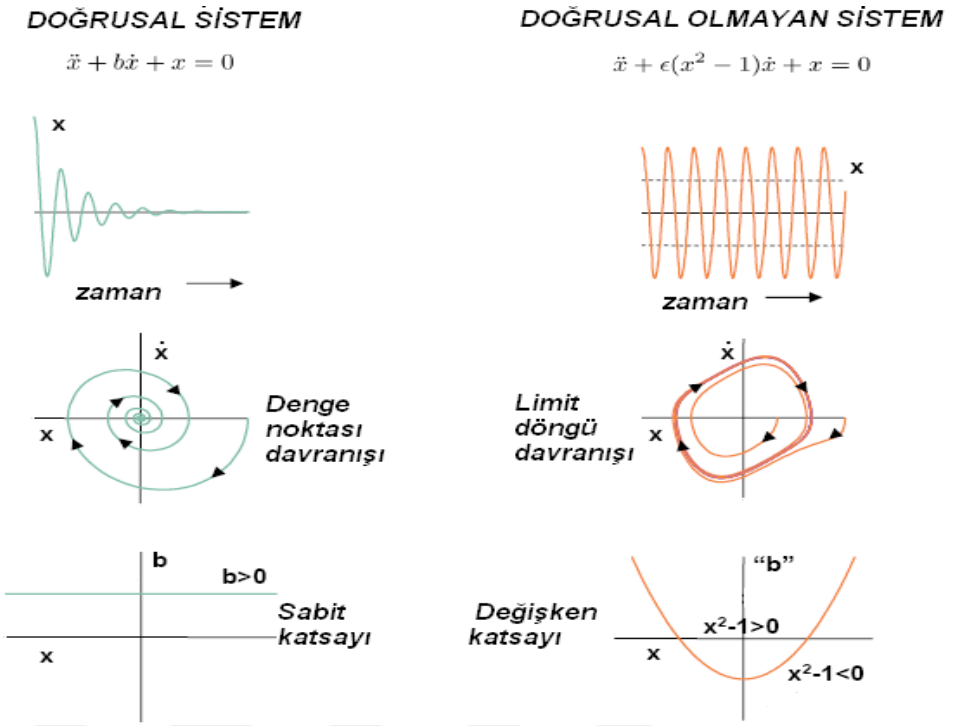
BÖLÜM 1. GİRİŞ

Fizikçiler uzun zamanlar boyunca, dinamik sistemlerin geçici olaylardan sonraki salınımlı davranışlarını tanımlamak için periyodik çözümlerin yeterli olduğuna inandılar. Ondokuzuncu yüzyılın sonlarında, 1892 yılında Fransız matematikçi Henri Poincare yeni ufuklar açan bir araştırma [1] ile basit dinamik kuralların çok karmaşık kararlı-hal davranışlarına yol açabileceğini, zamana göre değişimi Hamilton denklemleri ile yönlendirilen mekanik sistemlerin karmaşık davranışlar gösterebileceğini keşfetti. Ayrıca Poincare, şimdi kaotik yörünge denilen çok karmaşık yörüngelerin mümkün olduğunu ve başlangıç şartlarına hassas bağlılık gibi kaotik dinamiklerin çok önemli özelliklerini gösterdi.

Bilimsel olarak "kaos" terimi, rastgele gözüken olayların içinde var olan ve bu olayların temelini oluşturan bir birbirine bağlılıktan söz eder. Kaos bilimi, gizli biçim düzenleri, ince farklar, nesnelerin "duyarlılığı" ve tahmin edilemeyenin yeniye nasıl yol açtığına dair "kurallar" üzerine odaklanır. Kısaca kaos, düzensizliğin düzeni şeklinde tanımlanan doğrusal olmayan bir bilim dalıdır [2].

Gerçek hayatta fiziksel sistemlerin çoğu, sistem değişikliklerinin belli bir bölgedeki değişimi için doğrusal davranış gösterir. Ancak bu değişkenlerin doğrusal bölgenin dışındaki değişimi, sistemin doğrusal olmayan davranış göstermesine neden olur. Kaosun ve kaotik işaretlerin başlıca özelliklerini şöyle sırlayabiliriz:

- Başlangıç şartına olan hassasiyeti,
- Sınırsız sayıda değişik periyodik salınımlar içermesi,
- Genliği ve frekansı tespit edilemeyen, ancak sınırlı bir alanda değişen işaretler içermesi
- Gürültü benzeri güç spektrumlarına sahip olması.



Şekil 1.1. Doğrusal sistem - denge noktası davranışı ve doğrusal olmayan sistem - limit döngü davranışı [2].

Kaotik sistemlerin araştırılması ve uygulanmasına yönelik bilimsel ve endüstriyel alanlarda önemli çalışmalar gerçekleştirilmektedir. Mühendisliğin pek çok alanında kaotik sistemlerin varlığının ortaya çıkarılması, bu konuda yapılan yoğun çalışmalar ve yaşanan gelişmeler kaotik sistemlerin birçok uygulama alanında kullanılabileceğini göstermiştir. Bu uygulama alanlarına biyomedikal [3-5], kuantum elektroniği [6-8], bulanık mantık [9], güç elektroniği [10], biyokimya [11], kontrol [12, 13], fizik [14, 15], optimizasyon [16], mekatronik [17], gibi alanlar örnek olarak verilebilir. Bu alanların yanısıra özellikle şifreleme çalışmalarında da kullanılmaya başlanılmıştır. Bunun en önemli nedeni ise; kaotik işaretlerin geniş bantlı, gürültü benzeri, önceden tahmin edilmesi zor ve periyodik olmayan özelliklere sahip olması ve şifrelenen veriler üzerindeki karıştırma ve yayılmayı önemli ölçüde arttırmasıdır. Şifrelenmiş verilerin karmaşıklık ve hassasiyet düzeyinin yüksekliği ile, şifreleme algoritmalarının yapısı gibi etkenler şifrelemede en önemli unsurlardır. Standart şifreleme algoritmalarına alternatif olarak, kaos tabanlı şifreleme algoritmalarıyla yapılan çalışmalar son zamanlarda artış göstermiştir.

Günümüzde güvenli haberleşme ve kriptoloji alanlarında, kaotik sistemlerin analog ve sayısal tabanlı donanımlar ile gerçeklenerek kaotik üreteçler oluşturulması konusunda birçok çalışma yapılmıştır [18-19]. Özellikle kaos tabanlı mühendislik uygulamalarında kullanılması gereken en temel yapılardan birisi kaotik işareti üreten bir kaos sinyal üretecidir. Kaos üreteçleri, donanımsal olarak analog veya sayısal tabanlı olmak üzere iki farklı şekilde gerçekleştirilebilmektedir. Analog kaotik üreteç yapıları kullanan sistemlerin sıcaklık ve kullanım ömrü ile değerleri değişmektedir. Bu nedenle sayısal devre tabanlı kaotik üreteçler genel olarak analog yapılı kaotik üreteçlerden daha avantajlıdır [20]. Bu problem için en iyi çözüm sayısal donanım kullanarak kaotik üreteçlerin gerçekleştirilmesidir. Sayısal devre tabanlı kaotik üreteçler literatürde Sayısal İşaret İşlemciler (Digital Signal Processors (DSPs)) [21], Uygulamaya Özel Tümleşik Devreler (Application Specific Integrated Circuits (ASIC)) [22] ve Alan Programlanabilir Kapı Dizileri (Field Programmable Gate Array (FPGA)) [23] gibi farklı entegre yapılarla gerçeklenebilmektedir. ASIC tabanlı kaotik üreteçlerden diğer sayısal tabanlı eşdeğerlerine göre daha yüksek performans elde edilmektedir. Ancak ASIC tabanlı uygulamalar esnek bir yapıya sahip olmamakla birlikte bu sistemlerin ilk tasarım ve test maliyeti oldukça yüksektir. Ayrıca ASIC ile yapılan tasarımların maliyetinin düşürülmesi için önemli miktarda üretim yapılmalıdır. ASIC tabanlı seri üretim aşamalarında yapılacak küçük bir hata oldukça yüksek maliyet ve uzun zaman kaybına da neden olmaktadır. DSP çipleri ise kompleks matematiksel işlemleri gerçekleştirebilmek için optimize edilmiş yapılardır. Bu çipler, işlemleri sıralı (sequential) olarak gerçekleştirmektedir. Sürekli zamanlı kaotik sistemler genelde karakteristik olarak en az üç diferansiyel denklemden oluşmaktadır. Bu diferansiyel denklemlerin ayrık zamanlı yöntemlerle çözümlerinin mikroişlemci veya DSP tabanlı sistemler tarafından sıralı bir şekilde gerçekleştirilmesinde yüksek frekanslı işlemciler çevrim süresini kısaltmaktadır. FPGA çipleri ise paralel işlem yapabilmekle birlikte, tasarım ve test maliyetleri ASIC tabanlı uygulamalara göre daha düşüktür. Ayrıca kaotik devreler tekrar programlanabilir veya yeniden yapılandırılabilir sistemler içerisinde gerçeklenmeye uygundur. Bu sayede kaotik sistemler parametre değişimlerine göre farklı formda işaret üretebilmektedir [24].

Kaotik sistemler, ayrık zamanlı veya sürekli zamanlı olarak sınıflandırılabildiği gibi, içerdiği denklem sayısına göre de sınıflandırılabilir. Sistemin boyutu, yani içerdiği diferansiyel denklem sayısı arttıkça, denklemlerdeki parametrelerin ve başlangıç değerlerinin sayısı da artmaktadır. Parametre ve başlangıç değeri sayısının artması, şifre çözme işlemi sırasında bilinmesi gereken, bilinmeyen sayılarını da artırmaktadır. Şifreleme çalışmalarında ne kadar fazla bilinmeyen olursa, üçüncü kişiler tarafından şifreli verilerin çözülmesi de, o derece zor olacaktır. Kaos tabanlı yöntemlerle şifrelenmiş bir veriyi çözebilmek için, kullanılan kaotik sistemi, kaotik sistemdeki tüm denklemler, parametre ve başlangıç değerlerini bilmek gerekmektedir. Ayrıca, şifre çözme esnasında yapılan her hangi bir hata durumunda da, şifreli verinin çözümü mümkün olmayacaktır. Kaotik sistemlerin belirtilen hassasiyet özellikleri, şifreleme biliminde kaotik dinamiklerin tercih edilme oranını arttırmaktadır.

Sürekli zamanlı kaotik sistemler, Euler, Heun, RK4, RK5 gibi numerik analiz algoritmalarıyla ayrık zamanlı hale getirilerek, bilgisayar veya sayısal işlemciler gibi gerçek zamanlı sayısal ortamlarda, farklı uygulamalar için kullanılmaktadır. Sürekli zamanlı kaotik sistemlerden ayrıklaştırılarak elde edilen veriler, farklı şifreleme yöntemleri yardımıyla şifreleme çalışmalarında da kullanılabilmektedir.

Bazı çalışmalarda şifreleme çalışmaları kaotik sistemlerden rasgele sayılarak üretilerek, bu sayıların anahtarlar olarak kullanılmasıyla yapılmıştır. Üretilen sayıların rasgeleliği, şifreleme uygulamalarındaki güvenilirliği doğrudan etkilemektedir. Rasgele sayı üreteçlerinde kaos tabanlı olmayan yöntemlerle yapılan şifreleme çalışmalarda, karıştırma ve yayılmayı sağlamak en önemli problemdir. Kaotik sistemler bu özellikleri iyi sağladıklarından dolayı, kaos tabanlı şifreleme ön plana çıkmaktadır. Bu sebeple literatürde rasgele sayıların üretilmesine yönelik kaos tabanlı olan birçok çalışma bulunmaktadır.

Wieczorek ve arkadaşları, FPGA ile çift kararlı flip-flop kullanarak 50 MHz çalışma frekanslı ve 5 Mbit/s bit üretim hızında RSÜ tasarımı yaparak istatistiksel testlere tabi tutarak başarılı sonuçlar elde etmişlerdir [25]. Fischer ve arkadaşları 1 Mbit/s bit

üretim hızında, PLL tabanlı osilatörü FPGA kullanarak gerçekleştirmişler ve NIST testlerinden başarılı sonuçlar elde etmişlerdir [26]. István ve arkadaşları ise yine FPGA tabanlı, 50 MHz çalışma frekanslı klasik jitter osilatör yöntemi ile rasgele sayı üretimi gerçekleştirerek, NIST testlerinden başarılı sonuçlar elde etmişlerdir [27]. Akgül ve arkadaşları 2016 yılında Yeni Bir Kaotik Sistem Tabanlı RSÜ tasarımını NIST testlerinden geçirerek başarılı bir şekilde şifreleme uygulamalarında kullanmışlardır [28,29].

Kaos tabanlı şifrelemelerde genellikle literatürde var olan sistemler kullanılmaktadır. Yeni kaotik sistemler tasarlayarak yapılan şifreleme işlemleri ile güvenlik düzeyi arttırılmış olacaktır. Çünkü, şifreli verileri çözmek isteyen kişilerin, öncelikle yeni olan kaotik sistemi (denklemleri, tüm parametreleri ve başlangıç değerlerini) bulmaları gerekmektedir.

Şifrelemenin en temel unsurlarından birisi kullanılan anahtarlardır. Bu anahtarların üretilmesi ve saklanması en önemli problemlerden birisidir. Kaotik sistemler çok karmaşık dinamik özellikler gösterdiklerinden dolayı, rasgele anahtar üretiminde ön plana çıkmaktadırlar. Kriptolojik uygulamalarda kullanılan Rasgele Sayı Üreteçlerinin (RSÜ) ürettiği sayıların rasgeleliği, şifreleme uygulamalarının güvenliğini doğrudan etkilediklerinden, kriptolojik uygulamalar için kritik öneme sahiptirler. Son zamanlarda kaotik sistemler kullanılarak yapılan RSÜ tasarımlarında artışlar meydana gelmiştir. Üretilen rasgele sayılar, uluslararası en üst standartlar olan FIPS-140-1 ve NIST-800-22 gibi rasgelelik testlerinden geçtikten sonra şifreleme uygulamalarında kullanılabilmektedir. Öte yandan kaos tabanlı olan ve kaos tabanlı olmayan bazı şifreleme yöntemleri gerçek ortam uygulamalarında hız ve bellek açısından sorun olduğu için kullanılamamakta ve sınırlı alanlarda şifreleme çalışmaları yapılabilmektedir.

Günümüzde tekonolojinin hızla gelişmesiyle beraber internet bankacılığının kullanımı da çok ciddi oranda artış göstermiştir. Bu artış internet bankacılığında gizlilik ve güvenlik gibi konuların önemini de arttırmıştır. Çoğunlukla mobil yada web üzerinden kullanılan internet banakacılığında güvenlik, genel olarak kullanıcının

cep telefonu ile sağlanmaktadır. Mobil bankacılıkta kullanıcının gsm numarası, cep telefonu MAC bilgisi, sim kart ID bilgisi gibi çeşitli referans değerler baz alınarak mobil bankacılık uygulamasının cep telefonuna kurulması ile internet bankacılığının kullanımı mümkün olmaktadır. Web bankacılığında ise kullanıcının sms ile almış olduğu tek kullanımlık şifre ile banka sunucularına erişim sağlanmakta ve internet bankacılığının kullanımı gerçekleşmektedir. Mobil bankacılıkta kullanılmakta olan akıllı cep telefonları spam, olta saldırıları, zararlı yazılımlar (malware), elektronik takip, elektronik dinleme gibi saldırılara açık olduğundan, uzaktan yönetilebilir hale gelebilmekte ve çeşitli riskler ile karşı karşıya kalabilmektedir. Web bankacılığındaki sms uygulmasında da kullanıcının yine akıllı cep telefonuna sahip olması halinde, benzer risklerin ortaya çıkması söz konusu olmaktadır. Bu sebeple şifrematik cihazları, farklı düzey bir güvenlik uygulaması olarak bankalar tarafından müşterilerine sunulmaktadır.

1.1. Tezin Amacı

Bu tezin amaçları; yeni bir kaotik sistemi kullanan bir algoritma ile rasgele anahtar dizilerinin üretilmesi, bu dizilerin NIST rasgelelik testlerinden geçirilmesi, üretilen bitleri kullanan şifre üretme algoritmasının geliştirilmesi, yeni şifre üretme algoritmasını kullanan ARM tabanlı donanımsal şifrematik tasarımının yapılması ve gerçekleştirilmesi, tasarlanan örnek bir internet bankacılığı kullanıcı arayüzünde deneme uygulamalarının gerçekleştirilmesidir.

Bu amaçları gerçekleştirmek için öncelikle, literatüre Akgül ve Pehlivan tarafından yeni sunulan bir kaotik sistemi [30] (bundan sonra “AP Kaotik Sistemi” olarak bahsedilecektir) kullanan Rasgele Sayı Üretici (RSÜ) tarafından üretilen bitler, uluslararası en üst düzey rasgelelik testleri olan NIST-800-22 testlerinden geçirilmiştir. Daha sonra üretilen bu bitlerin farklı dizilimleriyle şifre üretme algoritması geliştirilmiştir. Bu algoritma ile önce bilgisayar ortamında, ardından ARM tabanlı mikrodenetleyici üzerinde şifrematik tasarımı yapılmıştır. Donanımsal prototip şifrematik cihazı tarafından üretilen, sayı ve harf içeren şifreler, LCD panel üzerinde yansıtılmaktadır. Son olarak üretilen şifreler, internet bankacılığı kullanıcı

arayüzünü temsil eden bir ekranda test edilerek, pratik hayattaki uygulanabilirliği gösterilmiştir.

1.2. Tezde İzlenecek Yol

Bu tezin tamamı beş bölüme ayrılmıştır. Birinci bölümde Kaos Bilimi ve tanımı, kaosun kullanım alanları, ikinci bölümde kaos ve kaotik sistemler, rasgele sayı üreteçleri ve testleri, gömülü sistemler, şifreleme teknikleri, internet bankacılığında güvenlik ve kullanıcı arayüz ekranları gibi temel kavramlar açıklanmıştır.

Üçüncü bölümde, AP Kaotik Sistemi'nin denge nokta analizi, faz portre analizi, Lyapunov üstel analizi gibi analizlerine değinilmiş, devre tasarımlarına yer verilmiştir. Ardından yeni kaotik sistemin Runga Kutta 4 (RK4) algoritması ile ayrıştırılarak çözülmesi sağlanmış, sonrasında rasgele sayı üretim algoritmasından bahsedilmiştir. Son olarak bu algoritma tarafından üretilen rasgele sayıların NIST istatiksel testleri ve sonuçları paylaşılmıştır.

Dördüncü bölümde Rasgele Sayı Üretici tarafından oluşturulan bitlerden yola çıkılarak şifreleme algoritması geliştirilmiştir. Bu algoritma ile Matlab programında şifreler üretilmişir. Ardından aynı şifrelerin donanımsal olarak elektronik devre ile üretilebilmesi için, gerekli bitlerin kodlanarak ARM tabanlı mikrodenetleyici içerisine gömülmesi amaçlanmıştır. Gömülü bu bitlerin, mikrodenetleyici tarafından çözümlenerek bilgisayar ortamı ile aynı şifrelerin üretilmesini sağlamak için, C dilinde mikrodenetleyicinin programlanması yapılmıştır. Sonrasında bu program mikrodenetleyiciye yüklenmiş, ardından LCD panel üzerinde şifre görüntülemesi yapması sağlanmıştır. Bilgisayar ortamında üretilen şifreler ise belli bir veri tabanına taşınarak, hazırlanan internet bankacılığı kullanıcı arayüzü benzetim programında, şifrematik cihazının testleri yapılmıştır.

Son bölümde ise: bu tez çalışmasında gerçekleştirilen kaos tabanlı donanımsal rasgele sayı üreteci ile üretilen şifreler yerine; kaotik sistemin hassas başlangıç şartları ve parametre değerleri değiştirilerek, farklı müşteri ihtiyaçları için farklı

cihazların kullanımını temsilen, farklı şifreler elde edilebileceğine değinilmiş, birden fazla cihazda kullanımına dair öneriler sunulmuş, değerlendirmeler yapılmıştır. Daha sonra literatürdeki diğer benzer çalışmalar ile karşılaştırmalar yapılmış, bu tezdeki şifre üretme algoritmasının başka alanlarda nasıl kullanılabileceğine değinilmiştir.

BÖLÜM 2. TEMEL KAVRAMLAR

Bu bölümde, kaos tabanlı rasgele sayı üretecinden elde edilen bitlerden tasarlanacak olan şifrematik cihaz tasarımı için gerekli olan bazı temel bilgilere yer verilmiştir. İlk olarak kaos ve kaotik sistemlerden bahsedilmiştir. Sonrasında rasgele sayı üreteçleri ve istastiksel rasgelelik testleri hakkında bilgi verilmiştir. Ardından cihaz tasarımı için kullanılacak olan mikrodenetleyici için gömülü sistemler hakkında genel bilgilendirme yapılmış, mikrodenetleyicinin programlanması için gerekli olan yazılım diline değinilmiştir. Daha sonra şifreleme teknikleri konusu ele alınmıştır. son kısımda da internet bankacılığında güvenlik ve kullanıcı arayüz ekranlarına dair tanımlamalara yer verilmiştir.

2.1. Kaos ve Kaotik Sistemler

Kaotik sistemler, ilk olarak bir matematikçi ve meteorolog olan Edward Norton Lorenz tarafından keşfedilmiştir. Lorenz, 1963 yılında meteorolog olarak çalışırken üç değişkenli bir sistemde başlangıç şartlarındaki çok küçük değişikliklerin belirli bir süre sonunda öngörülemez sonuçlar doğurabileceğini göstermiştir [31]. Literatürde kaos kelimesi ilk olarak 1975 yılında T. Y. Li ve J. A. Yorke tarafından "Üç periyot kaos anlamına gelir" isimli makalede kullanılmıştır [32]. Rössler 1976 yılında, yeni bir yedi terimli ikinci dereceden doğrusal olmayan ve Lorenz sisteminden daha basit olan bir kaotik sistem önermiştir [33]. Yine Rössler 1979 yılında, daha önceden 1975 yılında bulduğu sistemden daha basit bir kaotik sistem geliştirmiştir [34]. 1986 yılında Leon Chua çok basit bir devre yapısına sahip olan otonom bir kaotik devre geliştirmiştir. Chua devresi olarak isimlendirilen devre, basit bir yapıya sahip olmasına rağmen üç değişkenli kaotik sistemlerin elektronik olarak gerçeklenmesi ve kaos olayının açıklanması için örnek bir model devre olmuştur [35].

Literatürde, geliştirilen ve üzerinde bilimsel çalışmalar yapılan kaotik sistemlere örnek olarak Lorenz, Chua, Sprott, Rössler, Rabinovich, Rikitake, Burke-Shaw ve Chen sistemleri verilebilir [24].

Kaotik yapılı sistemler doğrusal yapıda olmayan sistemlerdir. Genel olarak bir sistemin matematiksel modeli durum denklemleri ile tanımlanır.

$$x = f_i\,(x_1, x_2, x_3 \ldots . x_n\ , t),\ \ x(0) = x_0,\ \ i = 1,2,3 \ldots n \qquad (2.1)$$

Şayet f_i fonksiyonlarının hepsi x_i değişkenlerine göre doğrusal ise sistem doğrusal olur ve durum denklemleri matris formunda basitçe ifade edilebilir. Bu durumda sistem, sürekli hal cevabı olarak bir denge noktası davranışı (kararlı veya kararsız) gösterir. Eğer herhangi bir f_i fonksiyonu doğrusal olmayan kısım içeriyorsa, bu sistem doğrusal olmayan sistem olarak adlandırılır. Bu durumda sistemin durum denklemleri matris formunda ifade edilemez. Sistemin sürekli hal cevabı, genelde limit döngü veya denge noktası davranışı gösterir.

2.1.1. Kaotik sistemlerin kullanım alanları

Günümüzde kaotik tabanlı sistemler, sağlık sistemlerinde kalp, karaciğer böbrek ve diğer organların ve sistemlerinin dinamiğine ilişkin bilgilerin toplanılmasında; güvenlik sistemlerinde iris, parmak izi tanıma konularında; gökbiliminde yıldızların, gezegenlerin ve Güneş Sistemindeki uydu ve kuyruklu yıldızların hareketini modellemede; meteoroloji alanında hava durumu tahminlerinde; haberleşme sistemlerinde güvenli bilgi aktarımı ve pek çok alanda kullanılmaktadır.

Dünyanın nonlineer sistemler halinde yaratıldığı göz önünde bulundurulursa, kaotik sistemlerin başlıca kullanım alanları aşağıdaki sıralanabilir [2]:

- Yapay zekâ
- Sanal ağ çözümlerinde
- Görüntü şifreleme tekniklerinde

- Sağlık sistemlerinde
- Savunma sistemlerinde (haberleşme)
- Kriptolojide
- Optik biliminde
- Güvenlik sistemlerinde

2.2. Rasgele Sayı Üreteçleri ve İstatiksel Rasgelelik Testleri

Günümüzde, bilişim dünyasındaki gelişmeler ve bilgi güvenliğine duyulan ihtiyaç rasgele olarak üretilen güçlü şifre üreteçlerinin oluşturulmasını zorunluluk haline getirmiştir. Öte yandan rasgele olarak üretilen bu sayıların gerçekten rasgele olup-olmadığına dair ölçütler konusu gelişim göstermiş ve istatiksel rasgelelik testleri adıyla anılan çeşitli testlere ihtiyaç duyulmuştur. Bu kısımda rasgele sayı üreteçleri ve istatiksel rasgelelik testlerine değinilecektir.

2.2.1. Rasgele sayı üreteçleri

Rasgele sayılar, önceden tahmin edilmesi güç olan yâda bir sonraki teriminin ne olacağı bilinme olasılığı çok düşük olan sayılardır. Diğer bir ifade ile; bir fonksiyonun bir yâda birden fazla teriminin bilinmesi ile birlikte, sonraki terimleri arasındaki ilişkisinin formül haline getirilmesi güç olan sayılardır.

Rasgele sayı üretçleri genel olarak Sözde RSÜ ve Gerçek RSÜ'ler olarak iki bölüm altında incelenmektedirler [36]. Sözde RSÜ'ler yazılımsal olarak gerçekleştirilirken, Gerçek RSÜ'ler donanımsal olarak gerçekleştirilmektedirler. Gerçek RSÜler kendi içerisinde analog tabanlı RSÜ'ler ve sayısal tabanlı RSÜ'ler olarak iki başlık altında incelenebilir. Sürekli zamanlı kaotik RSÜ'ler analog tabanlı RSÜ'ler içerisinde ele alınırken, ayrık zamanlı kaotik RSÜ'ler sayısal tabanlı RSÜ'ler içerisinde ele alınmaktadırlar.

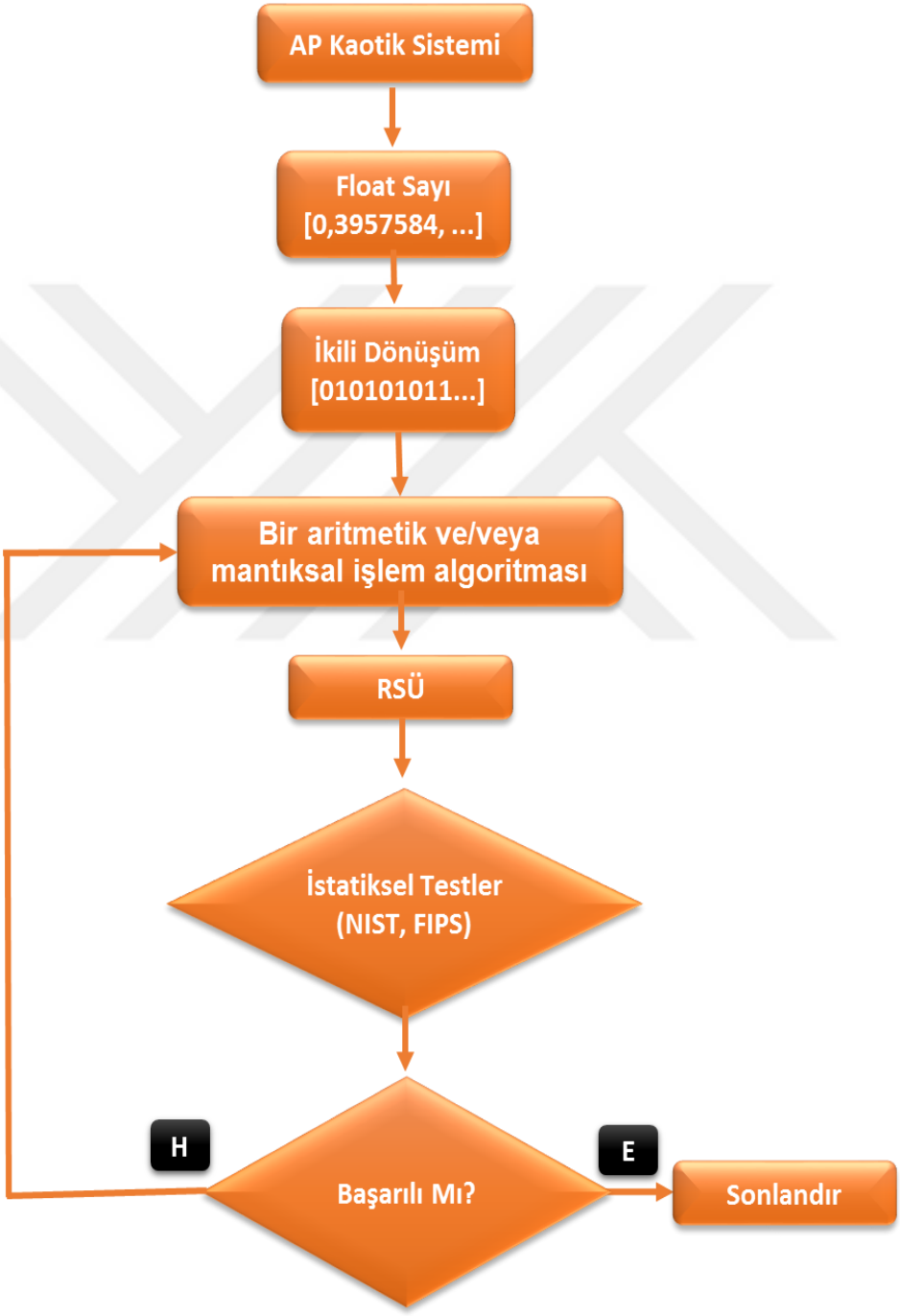
Rasgele sayılar hayatın birçok alanında farklı şekillerde karşımıza çıkmaktadır. Günlük yaşamda internet bankacılığında cep telefonumuza gelen kısa mesajlarda,

internet bankacılığı şifrematik uygulmalarında, e-posta hesaplarının aktifleştirilmesinde, birçok sitenin kullanıcı ekranlarına erişim esnasındaki güvenlik kodlarında, matematik alanında denklem çözme ve olasılık hesaplamalarında, otomobillerin araç kumandalarında, ağ güvenlik sistemlerinde, radar cihazlarında, bilgisayar oyunlarında ve birçok alanda çeşitli uygulamaları bulunmaktadır.

Rasgele sayı üretiminde başlıca üç farklı teknik bilinmektedir; bir gürültü kaynağının güçlendirilmesi, çift osilatör yapısı ve kaotik tabanlı RSÜ'lerdir. Bir gürültü kaynağının güçlendirilmesi; fiziksel olarak bir gürültü kaynağının oluşturulması, gürültünün kuvvetlendirilmesi ve gürültü dalga şeklinin değerlendirilerek bit dizisi şekline dönüştürülmesi ile elde edilir. Çift osilatör yapısı ise; biri hızlı diğeri daha yavaş serbest çalışan iki osilatörden elde edilen verilerin çeşitli şekillerde kıyaslanarak değerlendirilip, bit dizisi haline getirilmesi ile elde edilir. Kaotik tabanlı RSÜ'ler ise; osilatör davranışı sergileyen sistemin durum değişkenlerinin, çeşitili işlemlere tabi tutularak bit dizisine dönüştürülmesi ile elde edilir. Kaotik tabanlı sistemlerdeki bu durum değişkenlerinin bir ya da birkaç tanesi ile birlikte rasgele sayı üreteci tasarımının yapılması, geriye dönük sistem denkleminin tahmin edilmesini hemen hemen imkânsız hale getirmektedir. Bu sebeple RSÜ tasarımında kullanılmayan bu durum değişkenleri, kaotik tabanlı rasgele sayı üreteci tasarımlarına kilit özellik katmaktadır.

Kaotik sistemler ise rasgele sayı üreteç devrelerinin tasarımında, başlangıç şatlarına hasssas bağımlılıkları, doğrusal olmayan limit döngü davranışları, değer kümesindeki çok küçük değişimlere sahip olmalarıyla beraber kararlı bir yapı sergilemeleri şifreleme biliminde kendisini ön plana çıkartmaktadır. Kaotik sistemlerin belli bir küme içerisinde sağlamış olduğu hassas çıktılar, float sayılar olarak gözlenmektedir. Rasgele sayı üretimi için üretilen bu float türdeki sayıların öncelikle ikili sayı dizilerine dönüştürülmesi gerekmektedir. İkili sayı dizilerine dönüştürülen sayıların istatistiksel rasgelelik testlerine tabi tutulmadan önce, tasarlanacak bir aritmetik ve/veya mantıksal işlem algoritmasından geçirilmesi gerekmektedir. Bu işlem algoritmalarının uygulanması sonucu elde edilen sayıların rasgelelik seviyesini belirlemek için uluslararası kabul görmüş NIST-800-22 veya FIPS-140-1 gibi

istatistiksel testlerden geçirilmesi gerekmektedir [29]. Test sonucu başarılı olmayan aritmetik ve/veya mantıksal işlem algoritması tasarımlarının üzerinde testleri geçene kadar düzeltme/iyileştirme işlemleri uygulanmalıdır. Şekil 2.1.'de kaotik bir sistemden nasıl rasgele sayı üretimi gerçekleştirileceği anlatılmıştır.



Şekil 2.1. Kaotik sistemler ile rasgele sayı üretimi

2.2.2. İstatiksel rasgelelik testleri

2.2.2.1. FIPS-140-1 testi

FIPS-140-1 testi dört testten oluşmaktadır. RSÜ'nin çıkışından alınan ve genel olarak ikili sayı sisteminde 20 Kbit'lik bit dizisi dört teste tabi tutulmaktadır. Bit dizinin rasgele kabul edilebilmesi için tanımlı dört testten geçmesi gerekmektedir. Bu testler Monobit, Poker, Koşu ve Uzun Koşu testleridir.

Monobit Testi (Monobit Test): Bu testin amacı, RSÜ tarafından üretilen bit dizisindeki *'0'* ve*'1'* dağılım oranının rasgele bir diziden beklendiği gibi olup olmadığını tespit etmektir. Testin başarılı olabilmesi için 20 Kbit'lik bir dizideki *'1'* sayısının $9654<n<10346$ aralığında olması gerekmektedir. Eğer dizideki *'1'* sayısını ifade eden n belirtilen aralıklar içerisinde ise test başarılıdır. Aksi takdirde bit dizisi için monobit testi başarısız sayılmaktadır [37].

Poker Testi (Poker Test): Bu testte, 'k' bit dizisinin uzunluğunu belirtmek üzere, $k \geq 5*2m$ olacak şekilde, üst üste çakışmayan m bitlik parçalara ayrılmaktadır. Burada *i.* parça n_i diye adlandırılmaktadır. Rasgele bir bit dizisinden beklenen, k uzunluklu bir bit dizisinde tüm m bitlik blok parçalarının aynı sayıda birbirini tekrar etmesidir. Test için aşağıda verilen eşitlik (Denklem 2.2) kullanılmaktadır [38].

$$X = \frac{2^m}{k}\left(\sum_{i}^{2^m} n_i^2\right) - k \qquad (2.2)$$

Poker testinde bit dizisinin başarılı sayılabilmesi için m bitlik blok parçalarının birbirini tekrar etme sayısı olan X değerinin, $k=20000$ ve $m=4$ için, $1.03 < X < 57.4$ aralığında olması gerekmektedir.

Koşu Testi (Run Test): Bu testte RSÜ'den elde edilen bit dizisinin başarılı sayılabilmesi için, bit dizisinde ardı ardına gelen *'1'* ve *'0'*"lardan oluşan çeşitli

uzunluktaki bit blok sayısının Tablo 2.1.'de belirtildiği aralıklar içerisinde olması beklenmektedir. Burada *x* bit dizisinin koşu uzunluğunu göstermektedir. Eğer bit blok uzunluğu 6 bitten daha fazla ise bu uzun bloklar 6 bit olarak kabul edilmekte ve blok sayısı arttırılmaktadır [39].

Tablo 2.1. Koşu testi için blok uzunluklarına göre blok sayıları [39].

Blok Uzunluğu	Blok sayısı aralığı
1	2267≤x≤2733
2	1079≤x≤1421
3	502≤x≤748
4	223≤x≤402
5	90≤x≤223
6 ve 6+	90≤x≤223

Uzun Koşu Testi (Long Runs Test): Bu testte amaç, 34 veya daha fazla ardışık *'0'* veya *'1'* değerlerinin olup olmadığının tespit edilmesidir. Uzun koşu testinin başarılı kabul edilebilmesi için 20 Kbit uzunluğundaki bit dizisinin içerisindeki ardışık *'0'* veya *'1'* değerlerinin uzunluklarının 34'ten küçük olması beklenmektedir. Bu şart sağlanamadığı takdirde bit dizisi testi başarısız olmaktadır [40].

2.2.2.2. NIST-800-22 testi

Uluslararası düzeyde kabul görmüş olan testlerden bir diğeri ise NIST-800-22 testidir. NIST-800-22 testi, FIPS-140-1 testine göre hem test sayısı açısından daha fazla test içermekte, hem de bit dizileri daha güçlü bir şekilde testlere tabi tutulmaktadır. FIPS-140-1 testinde başarılı olabilen bir bit dizisi NIST-800-22 testinden başarısız olabilmektedir. Bu nedenle NIST-800-22 testi daha güvenilir bir test olarak tercih edilmektedir. NIST-800-22 testi 16 farklı test barındırmaktadır. Bit dizisinin başarılı sayılabilmesi için 16 testin tamamından başarıyla geçmesi gerekmektedir. Bu testler aşağıda verilmektedir [41]. NIST-800-22 testinde, test

edilecek rasgele bit dizisinin bazı parametreleri dışarıdan belirlenmektedir. Bu testlerde en önemli parametrelerden birisi olan *P-değeri* teste tabi tutulan rasgele dizilerin rasgeleliğinin bir ölçütü olarak kabul edilmektedir. *P-değeri* gerçekten rasgele bir dizi için 1'e yakın, bunun tersi durumunda ise *P-değeri* 0'a yakın olmaktadır.

- Frekans Testi (The Frequency Test)
- Bir Blok içerisinde Frekans Testi (Frequency Test within a Block)
- Akış Testi (The Runs Test)
- Bir Blok içerisinde En Uzun Birler Akış Testi (Tests for the Longest-Run-of-Ones in a Block)
- İkili Matris Derece Testi (The Binary Matrix Rank Test)
- Ayrık Fourier Dönüşüm Testi (The Discrete Fourier Transform (Spectral) Test)
- Örtüşmeyen Şablon Eşleştirme Testi (The Non-overlapping Template Matching Test)
- Örtüşen Şablon Eşleştirme Testi (The Overlapping Template Matching Test)
- Maurer'in "Evrensel İstatistik" Testi (Maurer's "Universal Statistical" Test)
- Doğrusal Karmaşıklık Testi (The Linear Complexity Test)
- Seri Testi (The Serial Test)
- Yaklaşık Entropi Testi (The Aproximate Entropy Test)
- Birikimli Toplamlar Testi (The Cumulative Sums Test)
- Rasgele Gezinimler Testi (The Random Excursions Test)
- Rasgele Gezinimler Değişken Testi (The Random Excursions Variant Test)

NIST-800-22 testlerinde bulunan testler rasgele üreteçler tarafından üretilen verilerin rasgelelik ölçüsünün belirlenebilmesi amacıyla geliştirilmiş istatistiksel testlerdir. Bu testlerin her biri rasgele olduğu varsayılan verilerin, farklı istatistikî yöntemlerle incelenerek hipotezin karara bağlanmasını sağlamaktadır. Aşağıda bu testler anlatılmaktadır.

İstatistiksel hipotez testleri, rasgele sayı üreteçlerinin ürettiği sayı dizilerine uygulanan istatistiksel testlerin sonuçlarını yorumlamak amacıyla kullanılmaktadır. Hipotez testlerinde öncelikle bir sıfır hipotezi (H_0) öne sürülür. Bu hipotezin tersi ise alternatif hipotez (H_a) olarak adlandırılmaktadır. H_0 hipotezi, üzerinde istatistiksel testler yapılan verilerin rasgele olduğu, H_a hipotezi ise üzerinde istatistiksel testler yapılan verilerin rasgele olmadığı anlamına gelmektedir [42].

Her istatistiksel testte, öne sürülen hipotez için bir test istatistik *P-değeri* hesaplanır. Bu değere göre hipotez kabul edilmekte veya reddedilmektedir. Her istatistiksel test için bir α önem seviyesi (significance level) belirlenmektedir. *P-değeri* $\geq \alpha$ ise H_0 hipotezi kabul edilmekte; diğer bir ifadeyle üzerinde rasgelelik testi yapılan veriler rasgele olduğu anlamına gelmektedir. *P-değeri* $< \alpha$ ise H_0 hipotezi reddedilir. Bir diğer deyişle verilerin rasgele olmadığı anlamına gelmektedir. Genellikle önem seviyesi değeri $0.01<\alpha<0.001$ aralığındadır. Eğer $\alpha=0.01$ için *P-değeri* $\geq \alpha$ ise H_0 hipotezi kabul edilerek üzerinde rasgelelik testi yapılan verilerin %99 doğrulukta rasgele olduğu kararına varılmaktadır. Eğer $\alpha=0.001$ için *P-değeri* $\geq \alpha$ ise H_0 hipotezi kabul edilerek üzerinde rasgelelik testi yapılan verilerin %99.9 doğrulukta rasgele olduğu kararı verilmektedir. Yapılan bu tez çalışmasında *P-değeri=0.001* olarak alınmaktadır [24].

NIST-800-22 testlerinde bulunan ve verilerin rasgelelik ölçüsünün belirlenebilmesi amacıyla geliştirilmiş 16 istatistiksel test aşağıda verilmektedir.

- Frekans Testi (Frequency Test): Bu testin amacı bit dizisindeki *'1'* ve *'0'* dengesini incelemektir. Burada $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots\ldots \varepsilon_n$ olmak üzere üretilen bit dizisi, n bit dizisinin uzunluğu, $S_n = X_1 + X_2 + \ldots\ldots\ldots X_n$ olmak üzere bit dizisinin *'0'* ve *'1'* değerlerinin $X_i = 2\varepsilon_i - 1$ dönüşümü kullanılarak *-1* ve *+1* değerlerine normalize edilmek suretiyle bit dizisinin toplanması ile elde edilen değerdir.

  Sonuç olarak *P-değeri* $\geq$ *0.001* olduğundan verilerin rasgele olduğu ve frekans testinden geçtiği söylenebilir.

– Blok Frekansı Testi (Block Frequency Test): Frekans testi tüm bit dizisini incelerken, blok frekansı testi ise bit dizisini *M* bitlik bloklara ayırarak blok içerisindeki *'1'* oranını incelemektedir. Rasgele üretilen *M* bitlik bloklardaki *'1'* oranının *M/2* olması beklenmektedir. Blok uzunluğu *M=1* olarak alındığında blok frekans testi, frekans testi ile aynı işlevi görmektedir. Testin geçerli sonuçlar üretebilmesi için veri bit uzunluğu en az *n=100* ve blok uzunluğu *M=20* olmalıdır.

  Sonuç olarak *P-değeri>0,001* olduğundan H_0 hipotezi kabul edilir veya diğer bir ifade ile gözlemlenen veri dizileri rasgele olarak kabul edilmektedir [43].

– Akış Testi (Runs Test): Dizideki 0 ve 1 bloklarının osilasyonunun değişimini belirlemektedir. Bu şekilde *'0'* ve *'1'* değerleri arasındaki değişimlerin yavaş veya hızlı olması hakkında fikir edinilebilmektedir [44]. Testin yapılabilmesi için $\tau = 2/\sqrt{n}$ olmak üzere aşağıdaki eşitliğin (Denklem 2.3) sağlanması gerekmektedir.

$$\left|\pi - \frac{1}{2}\right| \geq \tau \tag{2.3}$$

  Örneğin bit dizisi $\varepsilon=\varepsilon_1, \varepsilon_2,\ldots\ldots, \varepsilon_{10}$=1001101011 olsun.
  n=10 ve *π=6/10=3/5* bulunur ve $\tau = \frac{2}{\sqrt{10}} = 0.63$ olarak hesaplanmaktadır. Buradan, *|π-1/2|=|0,6-0,5|=0.1* değeri elde edilmektedir. Denklem (2.3)'e göre *0.1≥0.63* şartı sağlanamadığından akış testi yapılamaz.

– En Uzun Akış Testi (Longest Run Test): Testin amacı *M* bitlik blokların içerisinde en uzun *'1'* akışını incelemektir. Test edilecek *n* bitlik dizi *M* adet bloğa bölünmekte ve her bir blok içerisindeki en uzun ardışık birlerin akışına bakılmaktadır. En uzun birlerin beklenen uzunlukta olması, en uzun sıfırların da düzensizliğini ve beklenen uzunlukta olduğunu göstermektedir. Elde edilen sonuçların frekansları beklenen değer frekansları ile

karşılaştırılmaktadır. Testte *n* dizi uzunluğuna göre önerilen *M* blok uzunluğu değerleri Tablo 2.2.'te görülmektedir.

Tablo 2.2. Dizi uzunluğuna göre önerilen blok uzunluğu [43].

- Minimum *n*	- *M*
- 128	- 8
- 6272	- 128
- 750.000	- 10^4

P-değeri ≥ 0.001 olduğundan dizi rasgele olarak kabul edilmektedir [43].

– İkili Matris Derece Testi (The Binary Matrix Rank Test): Bu testte, *n* bit uzunluklu diziler *M* bit uzunluğundaki bloklara bölünür ve bu bloklar bir satırı belirtecek şekilde kullanılarak bir matris oluşturulur. Bu matrisin derecesi hesaplanarak bloklar arasındaki doğrusal bağımlılığın olup olmadığı incelenmektedir. Bu test için *M* satır sayısı ve *Q* sütun sayısı olmak üzere $M=Q=32$ olarak sabitlenmektedir. Test istatistiği referans dağılımı olarak χ^2 dağılımı kullanılmaktadır. Meydana gelecek matris sayısı $N=|n/MQ|$ şeklinde hesaplanmaktadır. Oluşturulan matrislerden kalan bit sayıları ihmal edilmektedir [41].

– Ayrık Fourier Dönüşüm Testi (The Discrete Fourier Transform (Spectral) Test): Literatürde spektral test olarak da isimlendirilen bu test, dizinin tepe yükseklikliklerine odaklanmaktadır. Bu testte amaç, dizinin periyodikliğinin incelenmesidir. Bu amaçla *d* gözlemlenen ve beklenen %95 eşik değerinin üstündeki frekans bileşenlerinin gözlemlenen ve beklenen sayıları arasındaki standart farklılığı ve $x_i=2\varepsilon-1$ göstermek üzere önce $X=x_1, x_2, \ldots . x_n$ dönüşümü yapılmaktadır [41].

- Örtüşmeyen Şablon Eşleştirme Testi (The Non-overlapping Template Matching Test): Bu testte, rasgele sayı üretecinin ürettiği *n* bitlik dizideki *m* bitlik blokların içerisinde, periyodik olmayan önceden belirlenmiş örnek dizinin bulunma sıklığının tespit edilmesi ve incelenmesi amaçlanmaktadır. Seçilen özel blokların tekrar edilmesi durumunda, gözlemlenen bloktan sonraki ilk bitten aramaya devam edilmektedir. Eğer belirlenen *m* bitlik özel bloklar bulunmaz ise pencere bir bit kaydırılarak arama işlemine devam edilir [41].

- Örtüşen Şablon Eşleştirme Testi (The Overlapping Template Matching Test): Bu testte amaç üretilen *n* bitlik dizi içerisindeki *m* bitlik blokların içerisinde, periyodik olmayan önceden belirlenmiş örnek dizinin bulunma sıklığının tespit edilmesi ve incelenmesidir. Bu testin örtüşmeyen şablon eşleştirme testinden farkı, bu testte eğer önceden belirlenen şablon tespit edilmiş ise arama işlemine bir bit sonra devam edilmesidir. Eğer belirlenen *m* bitlik özel bloklar bulunmaz ise pencere bir bit kaydırılarak arama işlemine devam edilir.

  Sonuç olarak $P \geq 0.001$ olduğunda dizi rasgele kabul edilmektedir [41].

- Maurer "Evrensel İstatistik" Testi (Maurer's "Universal Statistical" Test): Bu test 1992 yılında Princeton Üniversitesi bilgisayar bilimleri bölümünde bulunan U. Maurer tarafından geliştirilmiştir. Test rasgele dizinin veri kaybı olmadan sıkıştırılabilirliğine odaklanmaktadır. Ayrıca sunulan çalışmada bu testin kriptografik uygulamada gizli anahtar kaynağı için bir kalite ölçütü olduğu belirtilmektedir [45].

  Üniversal testte *L* her bir bloğun uzunluğu, *Q* başlangıç bölümü ve $K=[n/L]-Q$ olmak üzere test bölümünü ifade etmektedir. *QxL*-bit ve *KxL*-bit değerlerinin kalan bitleri atılmakta ve testte kullanılmamaktadır.

- Doğrusal Karmaşıklık Testi (The Linear Complexity Test): Bu testte, rasgele bit dizisinin Doğrusal Geri beslemeli Kayan Kaydedici (DGKK) (Linear Feedback Shift Register) (LFRS) uzunluğuna bakılarak dizinin kompleksliğinin incelenmesi amaçlanmaktadır. Dizi içerisinde DGKK uzunluğunun yüksek olması dizinin daha rasgele olduğunu göstermektedir [41].

- Seri Testi (The Serial Test): Bu testte *n* bit dizisinin uzunluğu, *m* her bir bloktaki bitlerin uzunluğu olmak üzere verilen dizideki her *m* bit örneğin dizideki diğer *m* bit örnekler ile aynı değişimi ve tekdüzelilik (uniformity) seviyesini incelemektedir. Eğer seri test için $m=1$ olursa frekans testi ile aynı işlevi görmektedir. Bu testte *P-değeri-1* ve *P-değeri-2* olmak üzere iki test sonucu elde edilmektedir [43].

- Yaklaşık Entropi Testi (The Aproximate Entropy Test): Bu testte amaç, seri testinde olduğu gibi tüm muhtemel örtüşen *m* bitlik örnek dizinin frekansının incelenmesidir. Test, rasgele bir dizi için beklenen frekansın, iki ardışık veya bitişik uzunluktaki (*m* ve $m+1$) örtüşen blokların frekanslarını karşılaştırmaktadır [24].

- Birikimli Toplamlar Testi (The Cumulative Sums Test): Bu testte amaç, rasgele bir dizi için kümülâtif toplamın beklenen davranışı için kısmi alt blokların kümülâtif toplamının çok büyük veya çok küçük olup olmadığının belirlenmesidir. Bu amaçla dizi öncelikli olarak $X_i=2\varepsilon_i-1$ dönüşümü kullanılarak giriş dizisi -1 ve $+1$ değerlerine normalize edilmekte ve yeni X_i dizisi elde edilmektedir. Rasgele bir dizide sonuçların sıfıra yakın çıkması beklenmektedir. Birikimli toplamlar testi için örneğin $\varepsilon=\varepsilon_1,\varepsilon_2,.....,\varepsilon_{10}=1011010111$ olmak üzere bir bit dizisi verilmektedir. Buradan normalize işlemi yapıldığında elde edilen yeni dizi $X=1, -1, 1, 1, -1, 1, -1, 1, 1, 1$ olmaktadır. Bu testin uygulanmasında ileri ve geri yönlü olmak üzere iki farklı metot kullanılmaktadır. Test aşamasında 0 seçilirse test ileri yönlü ve 1

seçilirse test geri yönlü çalışmakta ve bu metotların çalışması Tablo 2.3.'te verilmektedir [41].

Tablo 2.3. Test için ileri ve geri yönlü metotların uygulanması [41].

Metot 1=0 (İleri yönlü)	Metot 2=1 (Geri yönlü)
$S_1=X_1$	$S_1=X_n$
$S_2=X_1+X_2$	$S_2=X_n+X_{n-1}$
$S_3=X_1+X_2+X_3$	$S_3=X_n+X_{n-1}+X_{n-2}$
:	:
$S_k=X_1+X_2+X_3\cdots+X_k$	$S_k=X_n+X_{n-1}+X_{n-2}\cdots+X_{n-k+1}$
:	:
$S_n=X_1+X_2+X_3\cdots+X_k\cdots+X_n$	$S_n=X_n+X_{n-1}+X_{n-2}\cdots+X_{n-k+1}\cdots+X_1$

– Rasgele Gezinimler Testi (The Random Excursions Test): Bu testte amaç, birikimli toplam rasgele yürüyüşünde *K* adet döngünün sayısının belirlenmesidir. Birikimli toplam rasgele yürüyüşü, *'0'* ve *'1'* değerlerinden oluşan rasgele dizinin $X_i=2\varepsilon_i-1$ dönüşümü ile uygun *-1* ve *+1* normalize edilmiş dizisi elde edildikten sonra kısmi toplamlarının hesaplanması ile elde edilmektedir. Bu test -4, -3, -2, -1 ve +1, +2, +3, +4 olmak üzere sekiz *P-değeri*nin hesaplandığı serisel bir testtir [24].

– Rasgele Gezinimler Değişken Testi (The Random Excursions Variant Test): Bu testte amaç, birikimli toplam rasgele yürüyüşte belirli durumların toplam meydana gelme sayısının incelenmesidir. Bu test, rasgele bir yürüyüşte çeşitli durumlar için ziyaretin beklenen sayıdaki sapmalarını belirlemektedir. Bu testte -9, -8, -7, -6, -5, -4, -3, -2, -1 ve +1, +2, +3, +4, +5, +6, +7, +8, +9 olmak üzere on sekiz *P-değeri*nin hesaplandığı serisel bir testtir.

Bu test için örneğin $\varepsilon = \varepsilon_1,\varepsilon_2,.....,\varepsilon_{10}=0110110101$ olmak üzere bir bit dizisi verilmektedir. Buradan $n=10$ ve $X_i=2\varepsilon_i-1$ dönüşümü kullanılarak giriş dizisi -1 ve +1 değerlerine normalize edilmektedir. Yeni X_i dizisi $X=-1, 1, 1, -1, 1, 1, -1, 1, -1, 1$ olmaktadır. Bu test için verilen örneğe göre birikimli toplamlar testinde verildiği gibi uygulama metodu için 1 seçilerek ileri yönlü metot uygulanırsa S_i kısmi toplam olmak üzere $S_1=-1$, $S_2=0$, $S_3=1$, $S_4=0$, $S_5=1$, $S_6=2$, $S_7=1$, $S_8=2$, $S_9=1$, $S_{10}=2$ değerleri elde edilmektedir. Buradan $S=\{-1, 0, 1, 0, 1, 2, 1, 2, 1, 2\}$ olmaktadır. S' kümesi S kümesinin başına ve sonuna 0 elemanlarının eklenmesi ile oluşturulmaktadır. Sonuç olarak $S'=\{0, -1, 0, 1, 0, 1, 2, 1, 2, 1, 2, 0\}$ kümesi elde edilmektedir. Rasgele dizideki döngü sayısını ifade eden J değeri, S' kümesindeki ilk sıfır elemanı hariç kümedeki sıfır elemanlarının sayısı ve ξ ise testte bütün rasgele yürüyüşler süresince ziyaret edilen durumların toplam sayısını temsil etmektedir. Bu bölümde verilen örnek için: $\xi(-1)=1$, $\xi(1)=4$, $\xi(2)=3$ ve diğerleri $\xi(x)=0$ olmaktadır. Her bir döngü için ve her bir sıfır olmayan durum değeri için x $-9\leq x\leq -1$ ve $9\leq x\leq 1$ değerlerini alabilmekte ve her bir döngü içerisinde her bir x değerinin frekansı hesaplanmaktadır. Sonraki aşamada ise *P-değeri*nin hesaplanabilmesi amacıyla her bir $\xi(x)$ değeri için (18 $\xi(x)$ değeri ile) 18 tane *P-değeri* aşağıdaki eşitlik (Denklem 2.4) kullanılarak hesaplanmaktadır.

$$\mathrm{P-value} = \mathrm{erfc}\left(\frac{|\xi(x) - J|}{\sqrt{2J(4|x| - 2)}}\right) \tag{2.4}$$

Yukarıda elde edilen değerler kullanılarak sadece $x=1$ durumu için;

$$\mathrm{P-value} = \mathrm{erfc}\left(\frac{|4 - 3|}{\sqrt{23(4|1| - 2)}}\right) = 0.683 \tag{2.5}$$

Olarak elde edilmektedir. Buradan $P\text{-değeri}=0.502\geq 0,001$ olduğundan dizi rasgele kabul edilmektedir [43].

2.3. Gömülü Sistemler

Belirli bir fonksiyonu yerine getirmek üzere tasarlanmış yazılım ve donanım kombinasyonudur. İçinde bulunduğu sisteme karar verme yetisini kazandırmak üzere, çeşitli donanımlar vasıtasıyla yazılımsal olarak programlanıp, bu program neticesinde sistemin üreteceği çıktıyı çeşitli şekillerde (ses, görüntü, gerilim vb…) dış birimlere aktaran sistemlerdir. Günlük yaşantımızda kullandığımız eşyalarımızın hemen hemen hepsinde bu sistemleri görmek mümkündür. Bilgisayar, yazıcı, tarayıcı, hesap makinesi, cep telefonu, televizyon, fotoğraf makinesi, bulaşık makinesi, buzdolabı, elektronik oyuncaklar, araçlar vb… alanlarda sıkça kullanılmaktadırlar.

Gömülü sistemlerdeki yazılımlar genel olarak gerçek zamanlı çalışırlar. Tasarım olarak uzun soluklu çalışmak üzere ve hata yapmayacak şekilde programlanırlar. Ancak kullanılan sistemin kalitesine bağlı olmakla beraber çevresel koşullardan etkilenmeleri mümkündür. Sistemin çevresel koşullardan etkilenip, kilitlenme durumlarına karşın genelde reset tarzında bir düğme ile ilk programlandığı yapısına geri dönmeleri sağlanabilir. Düşük maliyet, düşük enerji tüketimi, çoğunlukla taşınabilir yapıda olmaları ve az yer kaplayan hacimleri nedeniyle elektronik dünyasında sıkça tercih edilmektedirler.

2.3.1. Mikrodenetleyici ve özellikleri

ARM mimarisi (Acorn RISC Machine) RICS tabanlı bir işlemci mimarisidir. İlk olarak ARM dizaynı 1983 yılında Acorn Computers Ltd tarafından geliştirildi. 32 ve 64 bit düzeyinde işlem yapabilme yeteniğine sahip olan bu işlemciler düşük güç tüketimi, yüksek performans ve düşük maliyetli olmaları nedeniyle en fazla tercih edilen mikroişlemcilerdendir.

Günümüzde ARM işlemci ailesi yeryüzündeki 32-bit gömülü işlemcilerin büyük bir kısmını oluşturmaktadır. Bugüne kadar 40 milyarın üzeri cihazda ARM tabanlı

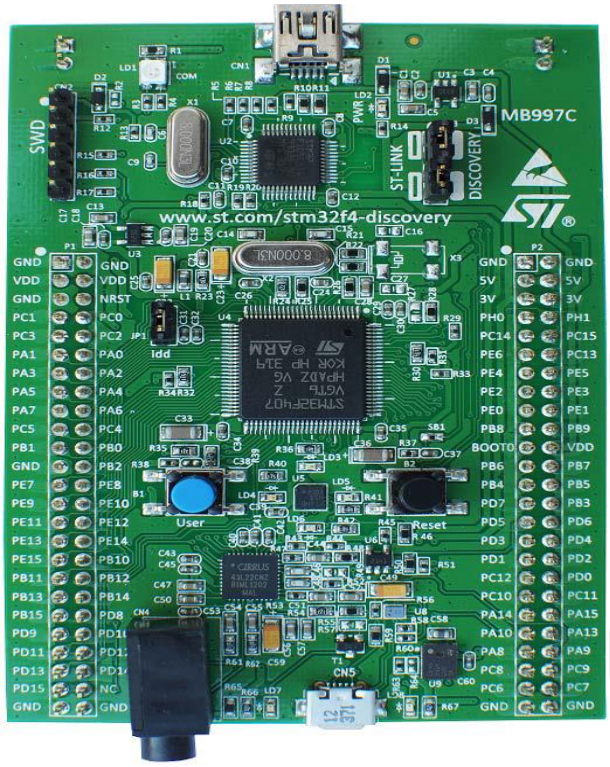
mikroişlemci kullanılmıştır ve kullanılmaya devam edilmektedir. ARM işlemciler akıllı cep telefonlarının % 95'inde, dijital kameraların % 80'inde ve bütün elektronik cihazların yaklaşık % 35'inde kullanılmaktadır [46].

Bu tezdeki cihaz tasarımı ve uygulaması ARM Serisi Cortex-M4 işlemcisi içeren STM32F407VG isimli deney kartı ile gerçekleştirildi. Bu kart ve işlemcisine ait özellikler ise aşağıdaki gibidir:

- 168 MHz, 32-bit Cortex™-M4 çekirdekli
- FPU (Floating Point Unit)
- Hafıza koruma ünitesi
- 210 DMIPS/1.25 DMIPS/Mhz (Dhrystone 2.1) ve DSP komutları
- 1 MB Flash memory
- 192 + 4 KB SRAM
- 64 KB CCM
- Paralel LCD ara yüzü, 8080/6800 modları
- Clock, reset, güç kaynağı yönetimi
- 1.8 V – 3.6 V arası uygulamaya yönelik güç kaynağı
- BOR4-26 Mhz kristal osilatör
- 2 x 12-bit D/A dönüştürücü
- 17 Adet timer
- SWD / JTAG bağlantı arayüzleri
- Harici kesme özelleğine sahip 140 adet I/O portu
- I2C, USART, UART, SPI, CAN, Ethernet haberleşme bağlantısı
- USB 2.0 Bağlantı
- 5V Besleme gerilimi
- 3V – 5V Çıkış gerilimi
- Dijital mikrofon, kullanıcı butonu ve ledleri
- RNG
- ADC, DAC
- USB mini-B aparatı

– 3.5 mm kulaklık bağlantı aparatı

Şekil 2.2.'de ise kartın üst taraftan görüntüsü yer almaktadır.



Şekil 2.2. STM32F407VG kartı

2.3.2. Yazılım dili ve programlama

C programlama dili günümüzdeki en yaygın programlama dillerinden biridir. Günümüzde neredeyse tüm işletim sistemlerinin % 95'lere varan oranında kullanılmaktadır. Gömülü sistemlerde, sürücü yazılımlarında, çeşitili analiz programlarında ve hız gereken yerlerde oldukça yaygın kullanıma sahip olan orta seviyeli bir programlama dilidir.

Mikrodenetleyicinin programlanması için *"MikroC"* derleyicisi kullanılmıştır. *MikroC*, pek çok donanım için geliştirmiş olduğu geniş kütüphanesi sebebiyle sıkça tercih edilmektedir. Ayrıca programlamayı kolaylaştıran pek çok arayüz araçlarına sahip olması, *MikroC*'nin tercih edilme oranını arttırmaktadır.

Programlamanın tamamlanmasıyla birlikte yapılan yazılım, "*mikroProg*" isimli bir uygulama ile mikrodenetleyicinin içerisine yüklenmektedir.

2.4. Kriptoloji

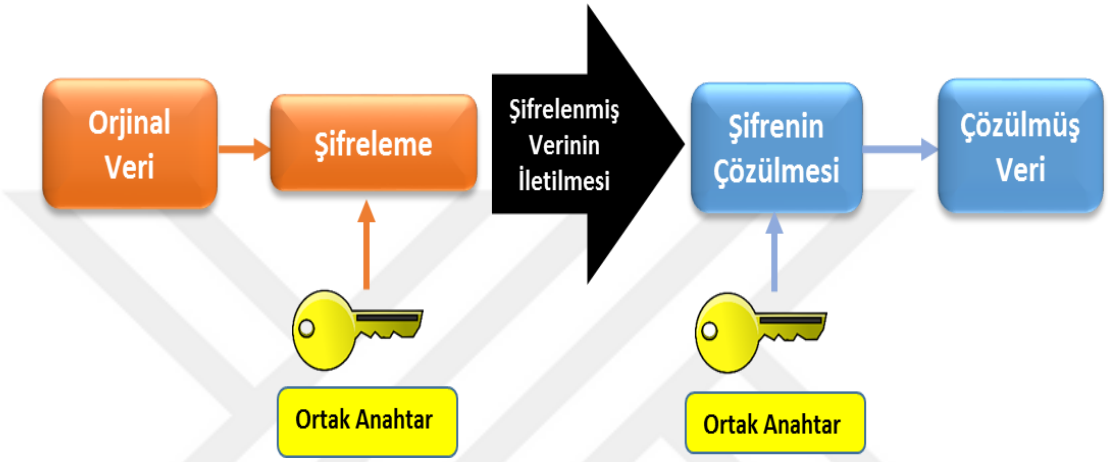
Kriptoloji kelime olarak, eski Yunanca'da gizli dünya anlamına gelen "kryptos" ve neden-sonuç anlamına gelen "logos" kelimelerinden gelmektedir. Esası gizlilik ilkesine dayanan kriptoloji, bir şifreleme bilimidir. Kriptografi ve kriptoanaliz olmak üzere iki ana dala ayrılmaktadır. Kriptografi bir verinin gönderici tarafta şifrelenmesiyle birlikte alıcı tarafta çözülerek tekrar ana veri haline getiren bilim dalıdır. Kriptoanaliz ise bir şifreleme algoritmasının kuvvetli ve zayıf yönlerini ortaya koyan analiz çalışmaları üzerine yoğunlaşır.

Kriptoloji, çeşitli verilerin belli bir sisteme göre şifrelenmesi, bu verilerin güvenlikli bir ortamda alıcıya iletilmesi ve iletilen verilerin karşı tarafta deşifre edilmesidir. Haberleşme, yapay zeka, görüntü şifreleme teknikleri, savunma sistemleri ve güvenlik sistemleri gibi bir çok alanda kriptoloji biliminin uygulamasını görmek mümkündür.

2.4.1. Simetrik anahtarlı şifreleme yöntemi

Simetrik anahtarlı şifreleme yönteminde, şifreleme ve şifre çözme algoritmalarında aynı anahtarlar kullanılmaktadır. Algoritmada kullanılan anahtarlar mutlaka gizli tutulmalıdır. Anahtar gizli olması gerektiğinden dolayı simetrik anahtarlı şifreleme yöntemi, gizli anahtarlı şifreleme olarakta ifade edilmektedir. Simetrik anahtarlı şifreleme yöntemi için genel blok diyagram Şekil 2.3.'de gösterildiği gibidir [47]. Şekilden de görüldüğü üzere orjinal veri bir şifreleme algoritması ile şifrelenmekte ve şifrelenen veri şifre çözme algoritması ile çözülerek tekrar orjinal veri elde edilmektedir. Dikkat edilirse iki işlem içinde ortak bir anahtar kullanılmaktadır. Şifreli veriyi çözmek isteyen kişi ortak anahtarı mutlaka bilmek ve üçüncü kişilerden korumak için gizlemek zorundadır. Günümüzde yaygın olarak kullanılan simetrik

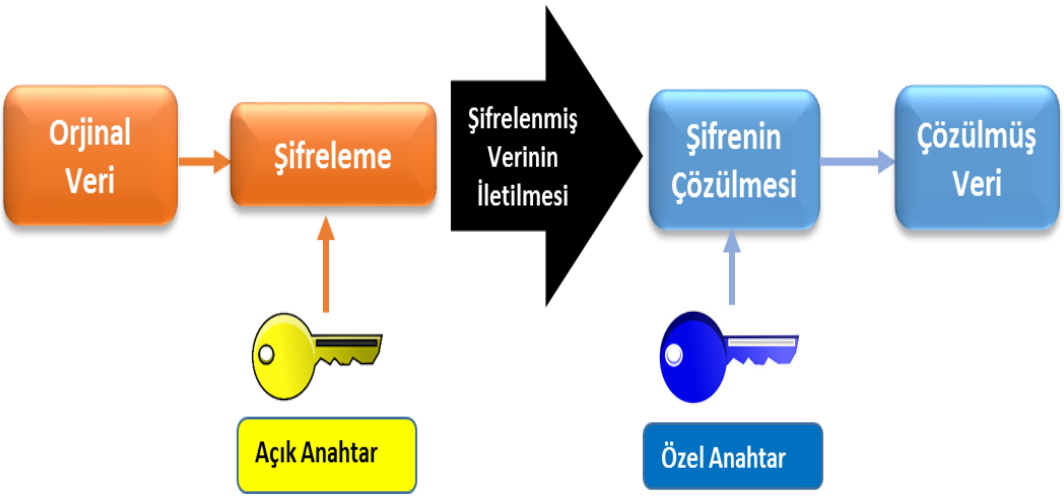
şifreleme algoritmaları, asimetrik şifreleme algoritmalarına nazaran hızlıdırlar, donanımla gerçeklemeleri kolaydır, fakat şifreli verilere karşı gerçekleştirilen saldırılara karşı daha az dayanıklıdır. Ayrıca anahtar dağıtımı, bütünlük ve kimlik denetimi gibi gereksinimler konusunda zayıftırlar [29]. AES, TEA, SEA, DES, Blowfish, IDEA ve RC4 gibi algoritmalar simetrik anahtarlı yöntemlere örnek olarak verilebilir.



Şekil 2.3. Simetrik anahtarlı şifreleme (Gizli anahtarlı şifreleme)

2.4.2. Asimetrik anahtarlı şifreleme yöntemi

Açık anahtarlı şifreleme yöntemi olarakta bilinen asimetrik anahtarlı şifreleme yönteminde şifreleme ve şifre çözme işlemleri simetrik anahtarlı şifreleme yönteminde olduğu gibi ortak değildir. Asimetrik anahtarlı şifrelemede açık ve özel olarak iki tür anahtar bulunmaktadır. Şekil 2.4.'de görüldüğü gibi şifreleme işleminini gerçekleştiren anahtar açık anahtar, şifre çözme işlemini gerçekleştiren anahtar ise özel anahtar olarak isimlendirilebilir [47]. Açık anahtara sahip olan kişi veriyi sadece şifreleyebilir, çözemez. Özel anahtarı olan kişiler şifreli verileri çözüp, okuyabilirler. Şifre çözme işleminde kişiye özel anahtar bulunduğu için şifreleme işleminde kullanılan anahtarın açık olması, simetrik anahtarlı şifreleme yönteminde olduğu gibi herkes tarafından bilinmesi problem değildir [48].



Şekil 2.4. Asimetrik şifreleme (Açık anahtarlı şifreleme)

Asimetrik anahtarda birden fazla kullanıcı varsa her şahsın kendine ait bir özel anahtarı vardır, bu anahtar sadece o kişi içindir ve verilerin güvenliği için o anahtarın gizli tutulması gerekmektedir [49]. Asimetrik anahtarlı şifreleme yöntemi, simetrik anahtarlı şifreleme yöntemine göre daha dayanıklıdır, güvenli ve kırılması zor algoritmalardır. Fakat hız açısından karşılaştırıldıklarında asimetrik algoritmalar simetrik algoritmalara göre çok daha yavaştırlar (örneğin 1500 kat kadar). Ayrıca simetrik anahtarlı şifreleme yöntemindeki gizlilik, anahtar yönetimi, bütünlük, kimlik denetimi gibi problemler asimetrik anahtarlı şifreleme yöntemi ile giderilebilir. Şifreleme algoritmalarında güvenlik genelde anahtar uzunluklarına bağlıdır ve bu yüzden seçilen anahtar uzunluğu şifrelenmek istenen veri türüne uygun olmalıdır. Asimetrik algoritmalar bazı durumlarda anahtar uzunluk açısından kullanıma uygun değildir. Simetrik algoritmalarda istenilen her türlü veri gerçekleştirilen tasarıma göre kolaylıkla şifrelenebilmektedir. Asimetrik anahtarlı şifreleme yöntemine RSA, ECC, DSA ve Elgamal algoritmaları örnek olarak verilebilir [29].

2.5. Şifrelemede Güvenlik Prensipleri

Şifreleme işleminde güvenliğin bir çok boyutu olmasına karşın, temel olarak dört prensipten söz edilebilir: gizlilik, bütünlük, kimlik denetimi, inkar edememe [50].

2.5.1. Gizlilik

Bilginin istenilmeyen kişiler tarafından elde edilememesidir. Fiziksel korumalardan yazılımsal algoritmalara kadar birçok yöntem ile verinin güvenliği sağlanabilir.

2.5.2. Bütünlük

Gönderilen bilgi ile alınan bilgi içeriğinin tamamen aynı olmasıdır. Verinin istenilmeyen kişiler tarafından ekleme, silme ve değiştirme gibi işlemlere uğramaksızın alıcıya iletilmesi gerekmektedir.

2.5.3. Kimlik denetimi

Bir iletişim içerisine giren kişilerin, birbirlerinin kimliklerinden emin olma durumlarıdır. Örneğin bir siteye erişmek isteyen kullanıcının, siteye giriş bilgilerinin doğru olması gerektiği gibi; bilgilerini girdiği sitenin de casus bir site olmama durumudur.

2.5.4. İnkar edememe

Bilgiyi alan yada gönderen kişinin, ilgili işlemden haberdar olmadığını iddia edememe halidir. Bu prensipteki amaç; gönderen ve alıcı arasındaki anlaşmazlıkları en aza indirgemektir. E-posta gönderen kişinin karşı taraftan “okundu” bilgisi istemesi bu duruma örnek verilebilir [50].

2.6. İnternet Bankacılığında Güvenlik ve Kullanıcı Arayüz Ekranları

İnternetin yaşamımızın her alanına az yada çok temas etmesi ile birlikte, hayattaki her sürecin internet yardımı ile nasıl hafif bir hale geleceği, insanlığın hep araştırma konusu olmuştur. İlk olarak askeri haberleşmenin temeli olarak kullanılmaya başlanılan internet, bugün hayatımızın hemen her noktasına girmiştir. Önceleri e-posta yollamak, online oyun oynamak, yazılı sohbet etmek gibi aktivitler üzerinden hayatımıza giren internet; şimdilerde alışveriş yapmak, televizyon izlemek, görüntülü konuşma yapmak, yemek tarifi almak, sosyal mesajlaşmayı sağlamak, home-office zemini olmak, uzak duruşma yapmak (vpn bir ağ ile), telekonferans toplantıları yapmak, sanal sınıf eğitimlerine katılmak gibi farklı amaçlar için kullanılmaktadır. İşte internetin bu geniş uygulama sahalarından biri de internet bankacılığıdır. Bankaya erişim süresi, sıra bekleme sıkıntısı, sadece atm yada internetten yapılan işlemler ve hayattaki her sürecin bir otomasyon sistemi içerisine alınarak; işletme maliyetlerinin düşürülmek istenmesi nedeni ile internet bankacılığına olan ihtiyaç olağanüstü bir şekilde artmıştır. İnternet bankacılığına olan bu ihtiyaç güvenlik konusundaki çalışmalara da ivme kazandırmıştır.

Banka müşterilerinin, hesaplarını internet kullanan platformlarda yönetebilmesi için çeşitli arayüz programlarına ihtiyaç duyulmaktadır. Bu platformlar web bankacılığı ve mobil bankacılık olmak üzere iki ana kısıma ayrılmaktadır.

2.6.1. İnternet bankacılığında güvenlik

1998 yılında Türkiye İş Bankası tarafından kullanıma sunulmuş internet bankacılığı, Garanti Bankası ve diger bankaların takibiyle ülkemizde uygulanmaya baslanmıştır. Rutin bankacılık işlemlerini büyük ölçüde kolaylaştıran internet bankacılığı sayesinde müsteriler, mevduat islemleri, yatırım hesabı işlemleri, para transferleri, vergi ve fatura ödemeleri gibi birçok işlemi zahmetsizce yapabilmektedirler [51].

İnternet bankacılığının müşterilere sağladığı kolaylıklar dışında, bankalara da şube sayısını azaltmasına olanak vererek maliyetleri düşürücü bir kolaylık sağlamaktadır.

Fakat bu avantajlarına karşı güvenlik gibi çok önemli bir riski de beraberinde getirmektedir. İnternet bankacılığı dolandırıcılığında en çok kullanılan metotlar arasında phishing, e-posta, keylogger ve screenlogger yöntemleri bulunmaktadır. Dolandırıcılar, bu yöntemleri kullanarak, müşterilerin hesap numara ve şifrelerini bularak, bu hesaplar üzerinde işlem yapabilmektedirler. Bankalar ise bu gibi durumlara karsı güvenlik önlemlerini artırma yolunda her geçen gün yeni çözüm yolları bulmaktadırlar [51]. Öte yandan sim kart klonlama ve gerçek olmayan kimlik ile sim kart temini yapma gibi yasal olmayan metotlar sebebiyle, sms bankacılığında güvenlik açısından yeni çalışmaların yapılmasına ihtiyaç duyulmaktadır.

Kullanıcının son aşama güvenlik kontrolüne gelmeden önce internet bankacılığındaki temel güvenlik düzeylerine değinmek gerekirse; ilk aşamada kullanıcıya verilen kullanıcı adı ve parolanın başka kişiler ile paylaşılmaması önemlidir. Bankalara ait siteler ile kullanıcı tarayıcısı arasındaki bağlantılar SSL VPN adı verilen özel bir ağ ile korunmaktadır. Devlet tarafından bu siteler kontrol edilmekte olup, e-devlet erişiminin internet bankacılığı yoluyla da yapılıyor olması bu duruma bir gösterge olabilmektedir. Bankalar fiziki olarak da sistemlerinin yer aldığı mekanları kartlı giriş-çıkış, video kamera gibi sistemler ile korumaktadırlar. Banka sistemleri kullanıcının giriş yaptığı IP'leri kaydetmekle beraber, hatalı giriş yapılan tarih, zaman ve IP gibi bilgilerini kullanıcının ilk doğru giriş yaptığı anda ekran üzerinde paylaşmaktadır. Ayrıca internet bankacılığı sistemleri, kullanıcın çoklu hatalı giriş yapması durumlarında ise internet bankacılığı hesabını pasif hale getirebilmektedir.

Son aşamada ise kullanıcı mobil bankacılık, web bankaclığına sms ile erişim, web bankacılığına şifrematik ile erişim, web bankacılığı mobil imza ile erişim gibi çeşitli güvenlik düzeyindeki uygulamalar ile internet bankacılığına giriş yapabilmektedir.

2.6.2. Kullanıcı arayüz ekranları

İnternet bankacılığında kullanıcının çeşitli işlemleri yapabilmesi için hesabını yönetebileceği arayüz programlarına ihtiyaç duyulmaktadır. Bu arayüz programları

mobil bankacılıkta genelde IOS ve Android tabanlı uygulamalar ile yönetilirken, web bankacılığında Asp.Net yada Php tabanlı sunucular tarafından yönetilmektedir.

BÖLÜM 3. AP KAOTİK SİSTEMİ ANALİZİ, DEVRE GERÇEKLEMESİ, RASGELE SAYI ÜRETECİ TASARIMI, İSTATİKSEL RASGELELİK TESTLERİ ve SONUÇLARI

3.1. AP Kaotik Sistemi'nin Analizi ve Devre Gerçeklemesi

Doğrusal olmayan davranış sergileyen kaotik sistemler; sınırsız sayıda değişik periyodik salınımlar içerir, genlik ve frekansları da tespit edilemez. Fakat sınırlı bir alanda kaotik işaretler içeren dinamiklere sahiptirler. Dinamik sistemler şimdiki ve geçmiş durumunun yanında olası durumların kümesini de içermektedir. Kaotik sistemler ayrık zamanlı ve sürekli zamanlı kaotik sistemler olarak iki grup olarak incelenebilir. Kaotik sistemler ayrık işaretler olarak incelenirse ayrık zamanlı, sürekli işaretler olarak incelenirse sürekli zamanlı dinamik sistemler olarak adlandırılmaktadır. Sürekli zamanlı kaotik sistemler diferansiyel denklemler kümesinden oluşmaktadır. Ayrık zamanlı kaotik sistemler tek boyutlu ve tek bir denklemden oluşabilirken, sürekli zamanlı kaotik sistemler için üç durum değişkeni içeren en az üç denklem olmalıdır.

Sürekli zamanlı kaotik sistemler genellikle adi diferansiyel denklemler ile ifade edilmektedir. Sürekli zamanlı *n* tane birinci dereceden adi diferansiyel denklem sistemi *i=1, 2, 3, ...,n* olmak üzere aşağıdaki eşitlik (Denklem 3.1) ile verilebilir [29].

$$\left.\begin{aligned} dx^{(i)}/dt &= f_1\left(x^{(i)}, x^{(i+1)}, \ldots . x^{(n)}\right) \\ dx^{(i+1)}/dt &= f_2\left(x^{(i)}, x^{(i+1)}, \ldots . x^{(n)}\right) \\ &\vdots \\ dx^{(n)}/dt &= f_n\left(x^{(i)}, x^{(i+1)}, \ldots . x^{(n)}\right) \end{aligned}\right\} \quad (3.1)$$

Denklemde verilen *x*, *n* boyutlu bir vektördür. Ayrıca $x \in R^m$ durum vektörüdür. x_0 başlangıç durum vektörüdür. *t* ise zamanı ifade etmektedir.

AP Kaotik Sistemi, sürekli zamanlı, 3 boyutlu bir kaotik sistemdir. Sistem Denklem (3.2)'de verildiği gibi 3 ayrı diferansiyel denklemden oluşmaktadır. Sistemde üç adet durum değişkeni (x, y, z), toplam on adet terim ve "a, b, c, d, e, f" olmak üzere altı adet parametre vardır. Sistemin başlangıç şartları x(0)=0, y(0)=0, z(0)=0 iken kaotik özellik göstermektedir. Tüm başlangıç şartlarının "0" olması gerçek ortam uygulamaları için kolaylık sağlamaktadır.

$$\begin{aligned} \dot{x} &= ay - x + zy \\ \dot{y} &= -bxz - cx + yz + d \\ \dot{z} &= e - fxy - x^2 \end{aligned} \tag{3.2}$$

Tipik sistem parametreleri a=2.8, b=0.2, c=1.4, d=1, e=10 ve f=2 olduğu durumda kaotik özellik göstermektedir. Farklı sistem parametrelerinde de farklı kaotik davranışlar elde edilmektedir. Aşağıdaki eşitlikte (Denklem 3.3) kaotik sistemin tipik parametrelerinin yazılmış hali verilmiştir.

$$\begin{aligned} \dot{x} &= 2.8y - x + zy \\ \dot{y} &= -0.2xz - 1.4x + yz + 1 \\ \dot{z} &= 10 - 2xy - x^2 \end{aligned} \tag{3.3}$$

AP Kaotik Sistemi'nde çok fazla parametre olması ve karmaşık sayılardan oluşan denge noktalarına sahip olması şifreleme çalışmalarında kullanıldığında avantaj sağlamaktadır. Kaotik sistemle şifrelenen verilerin çözülebilmesi için, kaotik sistemin ve sistemdeki tüm parametrelerin bilinmesi gerekmekte, parametre sayısı arttıkça da şifreli verilerin çözülmesi zorlaşmaktadır. Karmaşık sayılardan oluşan denge noktalarına sahip sistemlerde bazı kaotik analiz yöntemlerinin (Shilknov metodu gibi) uygulanamaması, şifreleme işleminde veya rasgele şifre üretiminde kullanılan kaotik sistemin bulunmasını zorlaştıracaktır.

Bir sistemin kaotik dinamik davranışlarının analiz edilmesinde denge noktaları, zaman serileri, faz portrelleri, Lyapunov Üstelleri, Çatallaşma Diyagramları gibi temel analiz yöntemleri kullanılmaktadır.

3.1.1. Sistem denge nokta analizi

AP Kaotik Sistemi'nin $F[x(t)]=0$ durumundaki denge noktalarını bulmak için $\dot{x}=0, \dot{y}=0, \dot{z}=0$ alınırsa,

$$\begin{aligned} 0 &= ay - x + zy \\ 0 &= -bxz - cx + yz + d \\ 0 &= e - fxy - x^2 \end{aligned} \tag{3.4}$$

elde edilir. Bu denklem sistemi çözülürse denge noktaları,

$$\begin{aligned} &E_1\ (2.707 + 0.573i, 0.413 - 0.661i, -1.584 + 3.332i) \\ &E_2\ (2.707 - 0.573i, 0.413 + 0.661i, -1.584 - 3.332i) \\ &E_3\ (-2.962 - 0.739i, -0.107 + 0.766i, -3.215 + 3.924i) \\ &E_4\ (-2.962 + 0.739i, -0.107 - 0.766i, -3.215 - 3.924i) \end{aligned} \tag{3.5}$$

olarak bulunur.

Denge noktaları analizi sonucunda; tümü karmaşık sayılardan oluşan E_1, E_2, E_3 ve E_4 denge noktaları elde edilmiştir. Kaotik sistem gerçek sayılardan oluşan denge noktalarına sahip olmadığı için gizli denge noktalı (hidden attractor) kaotik sistem olarak adlandırılır.

Denge noktalarının kararsız olup olmadığını anlamak için sistemin özdeğerlerinin bulunması gerekmektedir. Özdeğerleri bulmak için öncelikle sistemin Jacobian matrisinin alınması gerekmektedir. Sistemin Jacobian matrisi aşağıda (Denklem 3.6) verildiği gibidir.

$$J(x,y,z) = \begin{bmatrix} -1 & a+z & y \\ -bz-c & z & bx+y \\ -fy-2x & -fx & 0 \end{bmatrix} \quad (3.6)$$

E_1 denge noktasına ait özdeğerleri bulmak için; Denklem (3.5)'de bulunan denge noktaları, Denklem (3.6)'deki Jacobian matrisinde yerlerine yazılırsa, aşağıdaki eşitlikte (Denklem 3.7) belirtilen Jacobian matris elde edilir.

$$J(E_1) = \begin{bmatrix} -1 & 1.215+3.332i & 0.413-0.661i \\ -1.083-0.666i & -1.584+3.332i & -0.128-0.776i \\ -6.242+0.175i & -5.4154-1.147i & 0 \end{bmatrix} \quad (3.7)$$

Son olarak , $|\lambda I - J(E_1)| = 0$ denkleminin çözümünden aşağıdaki eşitlikte (Denklem 3.8) belirtilen karakteristik denklem elde edilir.

$$\lambda^3 + (2.584-3.332i)\lambda^2 + (3.341-7.466i)\lambda - (0.280-26.838i) = 0 \quad (3.8)$$

Karakteristik denklemin çözümünden, E_1 denge noktasına ait özdeğerler aşağıdaki gibi bulunur.

$$\begin{aligned} \lambda_1 &= 1.020+2.941i \\ \lambda_2 &= -1.068-2.149i \\ \lambda_3 &= -2.537+2.540i \end{aligned} \quad (3.9)$$

Sistemin kararsızlığı için özdeğerlerden en az birinin reel kısmının pozitif olması gerekir. λ_1'de pozitif olma koşulu sağlandığı için sistemin kararsızlığına ve kaotik olabileceğine işaret etmektedir. E_2, E_3 ve E_4 denge noktaları da Jacobian matrislerinde yerlerine yazılır ve elde edilen karakterisitk denklemler çözülürse, özdeğerler;

E_2 denge noktası için,

$$\begin{aligned}\lambda_1 &= 1.020 - 2.941\mathrm{i}\\ \lambda_2 &= -1.068 + 2.149\mathrm{i}\\ \lambda_3 &= -2.537 - 2.540\mathrm{i}\end{aligned} \tag{3.10}$$

E_3 denge noktası için,

$$\begin{aligned}\lambda_1 &= 1.289 + 3.078\mathrm{i}\\ \lambda_2 &= -1.177 - 1.500\mathrm{i}\\ \lambda_3 &= -4.327 + 2.346\mathrm{i}\end{aligned} \tag{3.11}$$

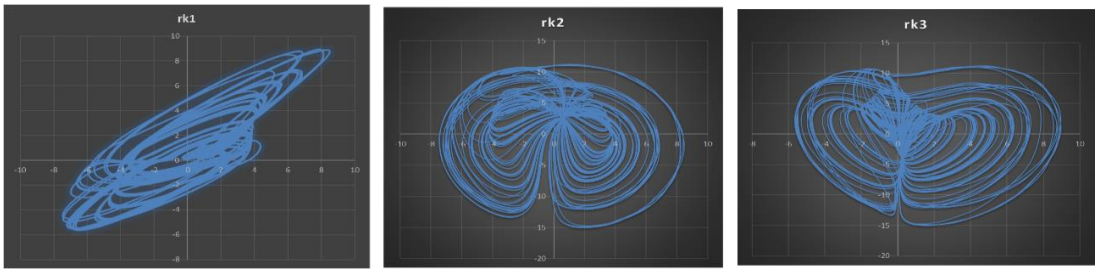
E_4 denge noktası için,

$$\begin{aligned}\lambda_1 &= 1.289 - 3.078\mathrm{i}\\ \lambda_2 &= -1.177 + 1.500\mathrm{i}\\ \lambda_3 &= -4.327 - 2.346\mathrm{i}\end{aligned} \tag{3.12}$$

olarak elde edilir. Sonuçlardan görüldüğü üzere, her bir denge noktası için bir özdeğerin gerçel kısmı pozitiftir ve sistemin kararsızlığını göstermektedir. Sistemin kaotik olduğuna kesin olarak karar verebilmek için Lyapunov ve çatallaşma analizi gibi daha üst seviyede analizler yapılmalıdır.

3.1.2. Faz portre analizi

AP Kaotik Sistemi'nin RK4 algoritması ile Matlab programında sayısal çözümünden elde edilen faz portreleri Şekil 3.1.'de görülmektedir.



Şekil 3.1. AP Kaotik Sisteminin x-y, x-z, y-z faz portreleri

3.1.3. Lyapunov üstelleri ve Lyapunov üstelleri spektrumu analizi

Bir dinamik sistemin davranışının çözümlenmesinde çok önemli bir ölçüt olan Lyapunov üstelleri, sistem hakkında karakteristik bilgiler verir ve aynı zamanda kaotik davranışın da bir ölçüsüdür. Eğer bir dinamik sistemin davranışı başlangıç şartlarına çok duyarlıysa, bu durumda zaman ilerledikçe, faz uzayındaki birbirine yakın yörüngeler hızlıca birbirinden ayrılır [2].

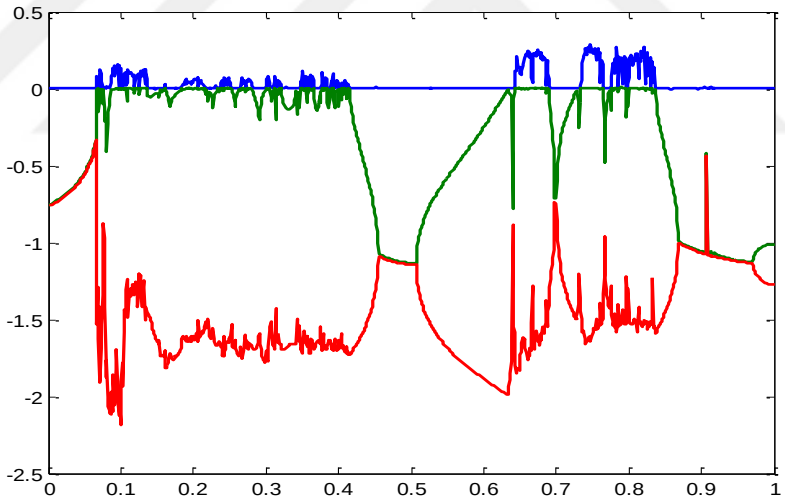
Lyapunov üsteli λ, başlangıç şartlarına olan duyarlılığın bir ölçüsünü verir ve faz uzayı içindeki komşu eğrilerin yerel ayrılma derecelerinin ortalaması olarak tanımlanır. Eğer λ negatif ise farklı başlangıç şartları aynı çıkış değerlerini vermeye meyillidir ve dolayısıyla gelişme kaotik değildir. Eğer λ pozitif ise farklı başlangıç değerleri farklı çıkış değerleri verir, yani hareket kaotiktir.

Dinamik sistemin boyutu kadar Lyapunov Üsteli vardır. Lyapunov üstellerinin toplamı, sıfırdan küçük ise kayıplı bir sistemi, sıfır ise kayıpsız sistemi (Hamiltonian sistem), sıfırdan büyük ise genişleyen bir sistemi tanımlar.

Bir kaotik sistemin temel karakteristiği, başlangıç şartlarına hassas bağımlığıdır. Verilen iki farklı başlangıç durumu birbirine çok yakın bile olsa, bu iki noktada oluşan yörüngeler üstel olarak artan bir ayırımla birbirlerinden uzaklaşırlar. Lyapunov üstelleri, kaotik sistemler için başlangıç durumlarındaki hassas bağımlılığı ölçmek için kullanılır.

Bir dinamik sistem, toplamları sıfırdan küçük olmak üzere, sıfırdan büyük en az bir Lyapunov üsteli içeriyorsa kaotik olarak tanımlanır. Kaotik bir yörüngenin Lyapunov üstelleri, en azından bir pozitif λi 'ye sahiptir. Üç boyutlu bir kaotik sistemin, kaotik davranış gösterdiği durumlarda, sahip olduğu Lyapunov üstelleri için tek mümkün durum (+,0,-) tipidir. Bu durumda λ1>0, λ2=0, ve λ3<0 olmaktadır.

AP Kaotik Sistemi'nin değiştirilen bir parametresinin değerine göre kaotik davranışa sahip olduğu durumlar, Lyapunov üstellerinin işaretlerinden anlaşılabilmektedir. AP Kaotik Sistemi'nin, 'b' parametresinin 0-1 aralığında değişimine göre hesaplanan Lyapunov üstelleri spektrumu Şekil 3.2.'de verildiği gibidir. Sistemin kaotik olması için Lyapunov üstellerine ait işaretlerin, parametrenin belli bir değerinde (+,0,-) olması gerekmektedir. Şekil 3.2.'den de görüldüğü üzere sistem belirli aralıklarda kaotik davranışa sahip olmaktadır [2].



Şekil 3.2. AP Kaotik Sistem için Lyapunov üstelleri spektrumu grafiği (b= 0-1)

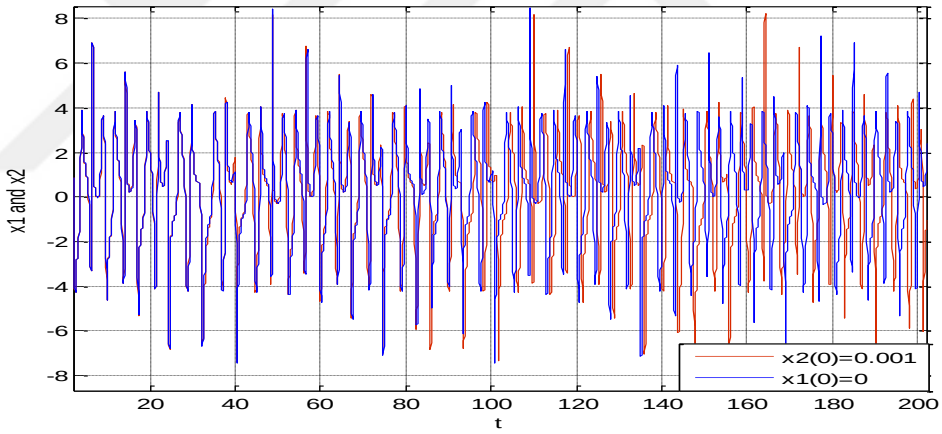
Lyapunov üstellerine ait grafiklerden sistemin kaotiklik boyutu da hesaplanabilmektedir. Örneğin Lyapunov üstelleri spektrumu grafiğinde kaotik olan b = 0.1 noktasından alınan Lyapunov üstelleri değerleri ile (L_1= 0.1403, L_2=0, L_3= -2.1515) aşağıdaki eşitlik (Denklem 3.13) yardımıyla hesaplanan kaotiklik boyutu

$$D_L = j + \frac{1}{|L_{j+1}|}\sum_{i=1}^{j} L_i = 2 + \frac{L_1+L_2}{|L_3|} = 2.0652103183 \quad (3.13)$$

olarak bulunur.

3.1.4. Zaman serisinde başlangıç değerlerine duyarlılık analizi

Sistemin başlangıç şart değerlerindeki çok küçük bir değişiklik sonucu, farklı çıktılar vermesi kaotiklik hakkında önemli ipuçları vermektedir. Şekil 3.3.'de görüldüğü üzere 'x' başlangıç şartı "0" olarak alınmış ve sonucu mavi olan eğri elde edilmiştir. Fakat x 1/1000 değiştirilerek, yani "0.001" yapılarak kırmızı eğri elde edilmiştir. İki eğri Şekil 3.3.'de beraber incelendiğinde çok küçük değişimlerin sistem üzerinde farklı sonuçlar verdiği, yani başlangıç şartlarına çok hassas olduğu görülebilmektedir.



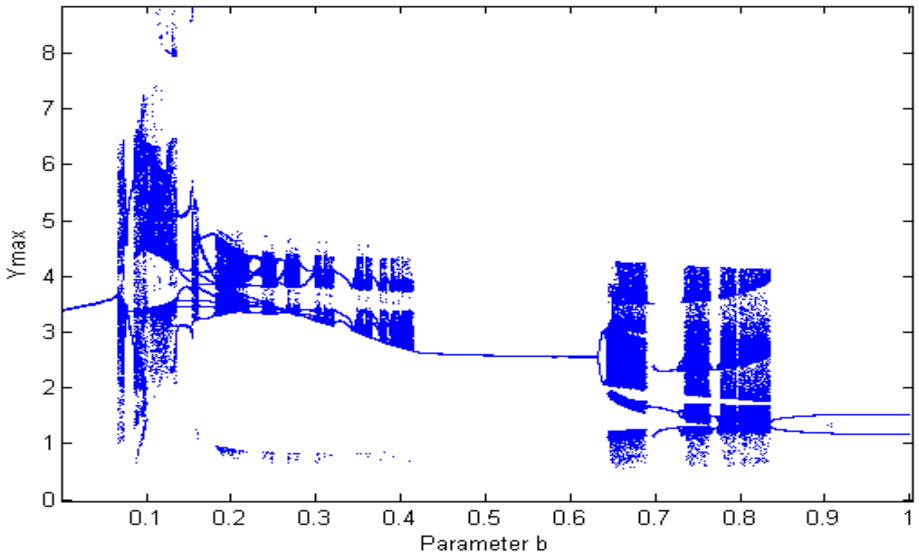
Şekil 3.3. x1(0)=0 ve x2(0)=0.001 için zaman serileri grafiği

3.1.5. Çatallaşma diyagramı analizi

Çatallaşma diyagramı çizdirilen bir sistemin hangi aralıklarda kaotik özellik gösterdiğine karar verilebilir [2].

Bu kısımda b parametresinin değişimine göre çatallaşma diyagramı çizdirilerek analiz yapılmıştır. Lyapunov üstelleri spektrumu analizi ve çatallaşma diyagramı aynı parametre aralıklarında, aynı sonuçları vermelidir. Yani Lyapunov üstellerinin

kaotik davranışı gösterdiği yerlerde, çatallaşma diyagramı da sistemin kaotik davranışa sahip olduğunu göstermelidir. Şekil 3.4.'te b parametresi için 0-1 aralığında çatallaşma diyagramı çizdirilmiştir. Şekil 3.4.'deki çatallaşma diyagramı ile Şekil 3.2.'deki Lyapunov üstelleri spektrumu grafiği karşılaştırıldığında sistemin kaotik davranış gösterdiği parametre değerlerinin aynı olduğu görülebilmektedir.



Şekil 3.4. Çatallaşma diyagramı (b= 0-1)

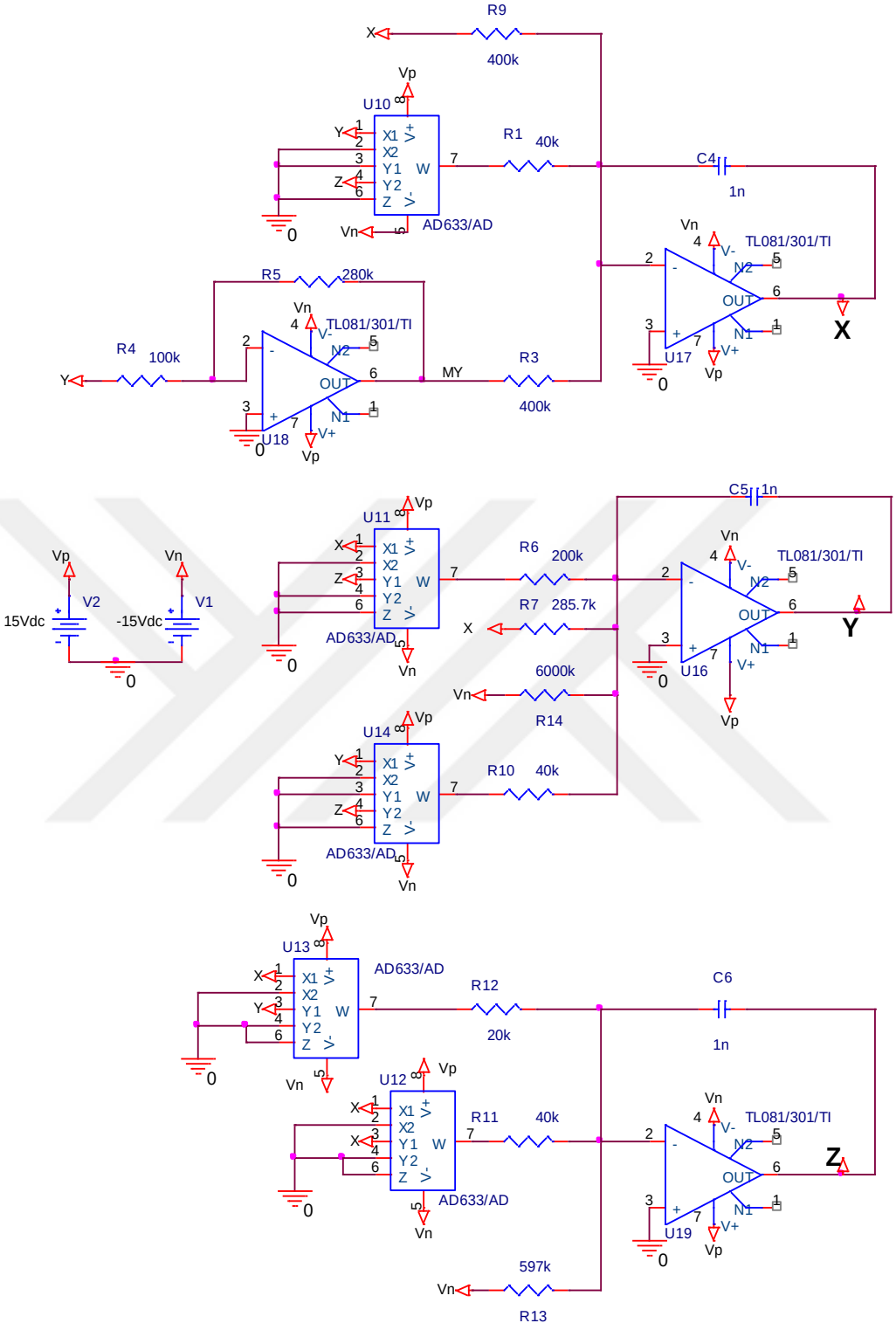
3.1.6. AP Kaotik Sistemi'nin elektronik devre tasarımı ve devre gerçeklemesi

AP Kaotik Sistemi'nin elektronik devre şeması Şekil 3.5.'de görüldüğü gibidir. Elektronik devre elemanları; direnç, opamp, çarpma entegresi, kondansatör gibi temel elemanlardan meydana gelmektedir. Kaotik sistemin tasarımında başlangıç değerleri ve parametreleri a =2.8, b=0.2, c=1.4, d=1, e=10, f=2 ve x(0)=y(0)=z(0)=0 olarak alınmıştır. Başlangıç şartları (x, y, z) "0" olan sistemlerin elektronik devre gerçeklemeleri, başlangıç şartı "0" olmayanlara göre daha kolay gerçekleştirilebilmektedir. AP Kaotik Sistemi, "0" başlangıç özelliğine (x, y, z) sahip olan bir sistem olduğu için gerçekleme daha kolay yapılabilmiştir. Gerçek ortam uygulamaları içinde başlangıç şartlarının "0" olması kolaylık sağlamaktadır. AP Kaotik Sistemi'nin denklemi aşağıda verilmektedir.

$$\begin{aligned}\dot{x} &= 2.8y - x + zy \\ \dot{y} &= -0.2xz - 1.4x + yz + 1 \\ \dot{z} &= 10 - 2xy - x^2\end{aligned} \tag{3.14}$$

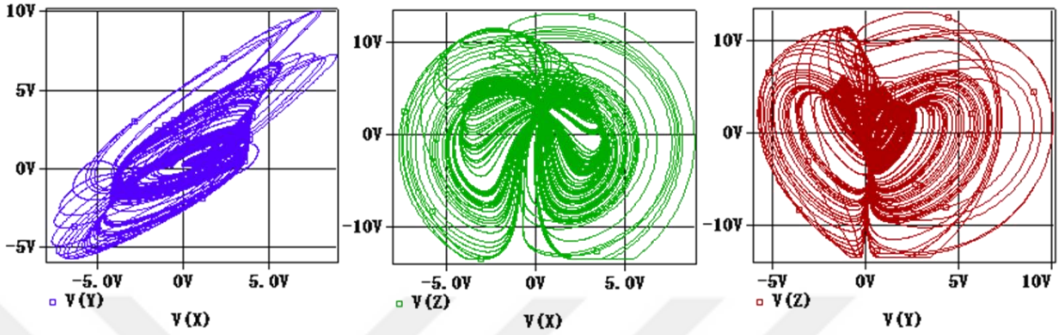
Şekil 3.5.'deki elektronik devre gerçeklemesinde opamp olarak TL081, çarpma entegresi olarak ise AD633 (Analog Devices) kullanılmıştır. Direnç değerleri R1=R10=R11=40K, R3=R9=400K, R4=100K, R5=280K, R6=200K, R7=285.7K, R12=20K, R13=597K, R14=6000K, kondansatör değerleri ise C1=C2=C3=1 nF olarak seçilmiştir. Bu değerler kaotik sistemin diferansiyel denklemlerinden yapılan modelleme sonucu elde edilmiştir.

Kaotik devrelerin elektronik ortamda gerçekleştirilebilmeleri için zamanlama skalalarına ihtiyaç duyulmaktadır. Cuomo ve Oppenheim'in [52] yaptıkları çalışmaya göre zamanlama skalası 2505'dir. Charlesworth, problemin kendi çözümünde olan fiziksel zaman ile (problem zamanı), analog bilgisayarda incelenen çözüm zamanının(hesaplama zamanı) birbirinden çok farklı olabileceğini belirterek zamanlama skalasına duyulan ihtiyacın sebebini açıklamıştır. Charlesworth' a göre t = problem zamanı, τ = hesaplama(devre) zamanı, β = zaman skalalama faktörü olmak üzere $\tau = \beta \times t$ 'dir. Bu tezde de aynı şekilde devre gerçeklemelerinde zaman skalalama faktörü $\beta = 2505$ alınmıştır. Zaman skalalaması uygulanırsa $\frac{1}{\beta.R.C} = 1$ elde edilir.



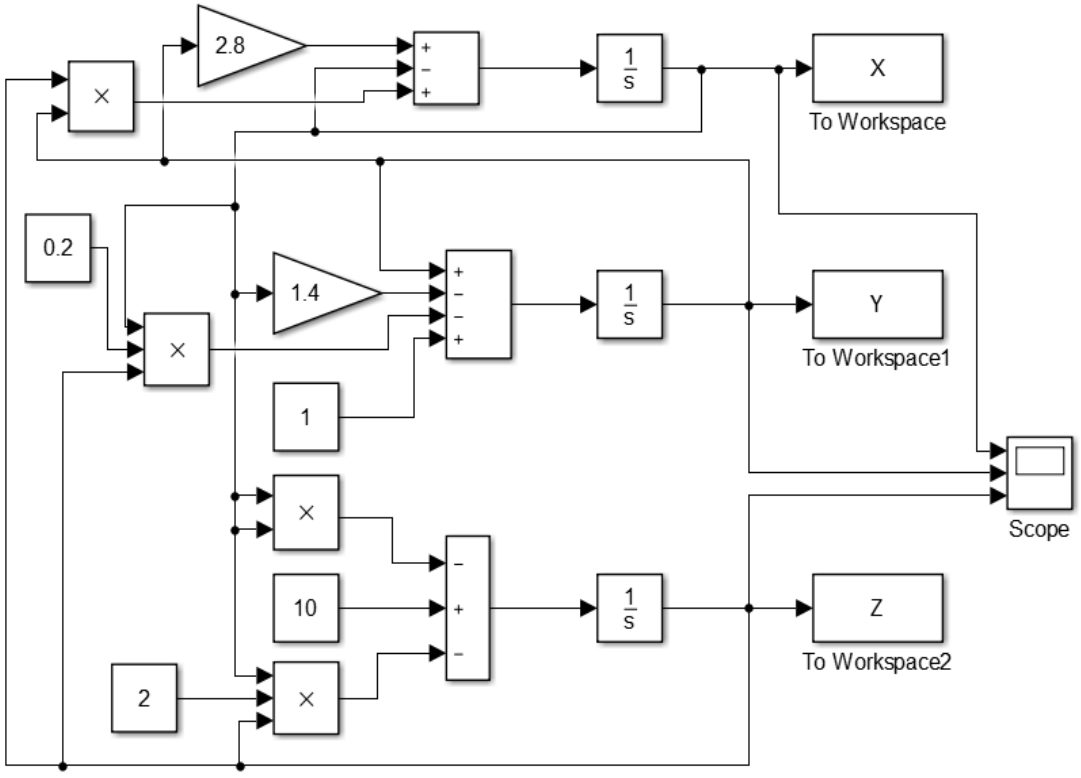
Şekil 3.5. AP Kaotik Sistemi ’nin elektronik devre tasarımı

Gerçekleştirilen simulasyon sonucu AP Kaotik Sistemi'nin faz portre çıktıları Şekil 3.6.'da görüldüğü gibidir. Faz portrelerinden de görüldüğü üzere AP Kaotik Sistemi, gerçek ortam uygulamaları için gerekli olan +15V, -15V aralıklarında olduğu için skale edilmelerine gerek yoktur.



Şekil 3.6. OrCAD PSpice programında çizdirilen x-y, x-z, y-z faz portre çıktıları

Şekil 3.7.'de ise AP Kaotik Sistemi'nin Matlab – Simulink programındaki modellemesi yer almaktadır.



Şekil 3.7. AP Kaotik Sistemi'nin Matlab – Simulink programında modellenmesi

3.2. Rasgele Sayı Üreteci (RSÜ) Tasarımı

3.2.1. AP Kaotik Sistemi'nin RSÜ tasarımı için ayrıklaştırılması

Euler, Heun, dördüncü dereceden Runge Kutta (RK4), beşinci dereceden Runge Kutta (RK5), Dormand-Prince gibi nümerik analiz algoritmalarıyla sürekli zamanlı diferansiyel denklem sistemlerinin sayısal çözümleri elde edilip, birçok sayısal uygulamada kullanılabilmektedir [53].

Euler algoritması, sürekli zamanlı sistemleri ayrıklaştırma işlemi için kullanılan en basit yöntemlerden birisidir. Euler algoritması sayısallaştırma için çok sık tercih edilmekte, fakat hassas çözümler yapamamaktadır. Kaotik sistemler hassas olduklarından dolayı Euler algoritmasını kullanmak uygun değildir. Gelişmiş olan diğer bir nümerik analiz algoritması olan Heun algoritması ise, yüksek frekanslı fonksiyonlar için uygun değildir. Bu nedenlerden dolayı kaotik sistemlerin ayrıklaştırılabilmesi için daha hassas olan RK4, RK5, Dormand-Price gibi yöntemlerinin kullanılması gerekmektedir. AP Kaotik Sistemi'nin ayrıklaştırılmış modeli için RK4 algoritması kullanılmıştır. RK4 algoritması oldukça iyi hassas sonuçlar üretmekte, hata oranı oldukça düşük, yazılımsal ve donanımsal olarak RK5 gibi daha gelişmiş nümerik analiz algoritmalarından daha basittir [29]. RK4 algoritması ile ayrıklaştırılmış modeli oluşturulan AP Kaotik Sistemi, RSÜ tasarımı ve diğer sayısal birçok alanda kullanılabilmektedir. AP Kaotik Sistemi'nin ayrıklaştırılması sonucu, kaotik sistemlerden kayan noktalı (floating point) sayı standartında sayılar elde edilerek, rasgele sayı üretimi için ilk basamak tamamlanmış olacaktır. Elde edilen kayan noktalı sayıların gerçek ortam uygulamalarında kullanılabilmeleri için ikili sayı sistemine dönüştürülmeleri gerekmektedir.

3.2.2. RK4 nümerik analiz algoritması

Runge-Kutta veya RK4 olarak isimlendirilen RK4 nümerik analiz algoritmasına ait ifadeler denklem kümesi aşağıda (Denklem 3.15) verildiği gibidir. Denklem (3.15)'te en son satırda verilen $y_{\lambda+1}$ değeri girilen sürekli zamanlı sayının ayrıklaştırılmış halidir. $y_{\lambda+1}$ değerinin bulunabilmesi için öncelikle k_1, k_2, k_3 ve k_4 değerlerinin hesaplanması gerekmektedir. Verilen denklem kümesindeki ilk ifadedeki k_1, Δh kadar aralık sonundaki başlangıç eğimi, k_2, k_3 ve k_4 değeri ise Δh aralığının orta noktasındaki sırasıyla k_1, k_2 ve k_3 değeri kullanılarak hesaplanan eğimdir. Bu şekilde devam edilerek y_λ değeri ve Δh aralık değerleri kullanılarak sistemin bir sonraki değeri olan $y_{\lambda+1}$ değeri sayısal olarak hesaplanmaktadır [54]. Aşağıda Denklem (3.15)'deki denklem kümesinde RK4 nümerik ayrıklaştırma algoritmasında bir adım için, gerekli işlemler sırasıyla verilmiştir.

$$
\begin{aligned}
k_1 &= f(y_\lambda) \\
k_2 &= f\left(y_\lambda + \frac{\Delta h}{2} k_1\right) \\
k_3 &= f\left(y_\lambda + \frac{\Delta h}{2} k_2\right) \\
k_4 &= f(y_\lambda + \Delta h k_3) \\
y_{\lambda+1} &= y_\lambda + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\Delta h
\end{aligned}
\tag{3.15}
$$

3.2.3. AP Kaotik Sistemi'nin RK4 algoritması ile ayrıklaştırılması

Bu kısımda AP Kaotik Sistemi'nin RK4 nümerik çözümyöntemi kullanılarak ayrıklaştırılmış modeli çıkarılmıştır. Denklem (3.16)'da verilen AP Kaotik Sistemi'nin, Denklem (3.17)'de f, g ve ξ fonksiyonlarına göre RK4 algoritması kullanılarak ayrıklaştırılmış matematiksel modeli verilmektedir [29].

$$\begin{aligned}\dot{x} &= f(t,x,y,z) = ay - x + zy \\ \dot{y} &= g(t,x,y,z) = -bxz - cx + yz + d \\ \dot{z} &= \delta(t,x,y,z) = e - fxy - x^2\end{aligned} \tag{3.16}$$

$$\begin{aligned}x(k+1) &= x(k) + \frac{1}{6}\Delta h[k_1(k) + 2k_2(k) + 2k_3(k) + k_4(k)] \\ y(k+1) &= y(k) + \frac{1}{6}\Delta h[\lambda_1(k) + 2\lambda_2(k) + 2\lambda_3(k) + \lambda_4(k)] \\ z(k+1) &= z(k) + \frac{1}{6}\Delta h[\xi_1(k) + 2\xi_2(k) + 2\xi_3(k) + \xi_4(k)]\end{aligned} \tag{3.17}$$

Denklem (3.17)'de bulunan κ, λ, ξ parametreleri, aşağıdaki eşitlikte (Denklem 3.18) verildiği gibi hesaplanmaktadır. Denklem (3.17)'de 1. basamaktaki tüm k parametreleri, Denklem (3.16)'daki kaotik sistemin ilk denklemine ait değerleri, 2. basamaktaki tüm λ parametreleri ikinci denkleme ait değerleri, 3. Basamaktaki tüm ξ parametreleri ise üçüncü denkleme ait değerleri Denklem (3.18)'de verildiği gibi hesaplanmaktadır. Bulunan katsayılar Denklem (3.17)'deki RK4 algoritmasında yerlerine konularak, kaotik sistemin Δh kadar adım sonrası değeri olan ayrıklaştırılmış *x(k+1)*, *y(k+1)* ve *z(k+1)* değerlerini hesaplanmaktadır. Her adım sonunda bulunan *x(k+1)*, *y(k+1)* ve *z(k+1)* değerleri hem hesaplanan o adımda çıkış olarak, hem de bir sonraki adımda başlangıç şartı olarak kullanılmaktadır [29].

$$k_1 = f\big(x(k), y(k), z(k)\big)$$
$$\lambda_1 = g\big(x(k), y(k), z(k)\big)$$
$$\xi_1 = \delta\big(x(k), y(k), z(k)\big)$$

$$k_2 = f(x(k) + \frac{1}{2}\Delta h k_1, y(k) + \frac{1}{2}\Delta h \lambda_1, z(k) + \frac{1}{2}\Delta h \xi_1)$$
$$\lambda_2 = g(x(k) + \frac{1}{2}\Delta h k_1, y(k) + \frac{1}{2}\Delta h \lambda_1, z(k) + \frac{1}{2}\Delta h \xi_1)$$
$$\xi_2 = \delta(x(k) + \frac{1}{2}\Delta h k_1, y(k) + \frac{1}{2}\Delta h \lambda_1, z(k) + \frac{1}{2}\Delta h \xi_1)$$

(3.18)

$$k_3 = f(x(k) + \frac{1}{2}\Delta h k_2, y(k) + \frac{1}{2}\Delta h \lambda_2, z(k) + \frac{1}{2}\Delta h \xi_2)$$
$$\lambda_3 = g(x(k) + \frac{1}{2}\Delta h k_2, y(k) + \frac{1}{2}\Delta h \lambda_2, z(k) + \frac{1}{2}\Delta h \xi_2)$$
$$\xi_3 = \delta(x(k) + \frac{1}{2}\Delta h k_2, y(k) + \frac{1}{2}\Delta h \lambda_2, z(k) + \frac{1}{2}\Delta h \xi_2)$$

$$k_4 = f(x(k) + \Delta h k_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3)$$
$$\lambda_4 = g(x(k) + \Delta h k_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3)$$
$$\xi_4 = \delta(x(k) + \Delta h k_3, y(k) + \Delta h \lambda_3, z(k) + \Delta h \xi_3)$$

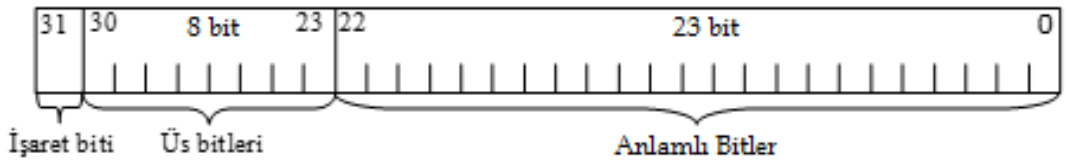
AP Kaotik Sistemi'nin birinci durum değişkeni x için örnek ayrıklaştırılmış ifadeler 10 adım için Tablo 3.1.'de verilmiştir.

Tablo 3.1. AP Kaotik Sisteminin RK4 ile ayrıklaştırma işlemi sonucu elde edilen kayan noktalı sayılar

Adım Sayısı	Ayrıklaştırma Sonucu Elde Edilen Kayan Noktalı Sayılar (x durum değişkeni için)
1	0.003866508205357
2	0.017022109192077
3	0.042062260612412
4	0.082042775172218
5	0.140612988083349
6	0.222137744672527
7	0.331789743765176
8	0.475571084896906
9	0.660185655353679
10	0.892627200833881

3.2.4. AP Kaotik Sistemi ile RSÜ Tasarımı

RK4 nümerik analiz algoritması sonucu ayrıklaştırılan sayıların en hassas ve yüksek miktarda gösterilmesini sağlayan sayı sistemleri, tek duyarlı (32-bit_float) ve çift duyarlı (64-bit_double) olarak bilinen kayan noktalı sayı standartıdır. Bir önceki kısımda kayan noktalı sayı standartında elde edilen sayıların gerçek ortam uygulamalarında kullanılabilmeleri için binary sayı formatına çevrilmeleri gerekmektedir. Şekil 3.8.'de 32-bit tek duyarlı IEEE 754-1985 kayan noktalı sayı standardı gösterilmiştir. Bu standartın en yüksek değerlikli biti, yani en anlamlı olan 31. Bit, işaret biti olarak isimlendirilmektedir. İşaret bitinin değeri eğer "0" ise sayı pozitif, "1" ise sayı negatif olmaktadır. Sayının üstel kısmını belirtmek için, ikinci kısımdaki 8 adet olan üs bitleri kullanılmaktadır. Verilen gösterim tek duyarlı olduğu için üs için kaydırma değeri 2^{8-1}-1'den 127'dir. Son kısım olan üçüncü kısım ise anlamlı bitleri temsil etmektedir. Anlamlı bitler kayan noktalı sayı standardında sayının kesirli yani anlamlı kısmını göstermektedir [55].



Şekil 3.8. 32-bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi

RK4 algoritması ile ayrıklaştırılan sayılar, kayan noktalı sayı standartında yani float sayılar olduğu için rasgele sayı üretimi için ikili sayı sistemine dönüştürülmesi gerekmektedir. Tablo 3.1.'deki, RK4 ile elde edilen sayıların ikili sayı formatına dönüştürülmüş formatı Tablo 3.2.'de verilmiştir. Dikkat edilirse, tüm float sayılar pozitif olduğu için, ikili sayıya dönüştürülmüş formatta da en anlamlı yani 31. bitde "0" sonucu elde edilmiştir.

Tablo 3.2. AP Kaotik Sistemi'nin RK4 ile ayrıklaştırma işlemi sonucu elde edilen ikili sayılar

Adım Sayısı	Ayrıklaştırma Sonucu Elde Edilen Kayan Noktalı Sayılar	İkili Sayı Dönüşümleri (Float to Binary)
1	0.003866508205357	00111011011111010110010100111110
2	0.017022109192077	00111100100010110111000111110011
3	0.042062260612412	00111101001011000100100101111010
4	0.082042775172218	00111101101010000000011000001011
5	0.140612988083349	00111110000011111111110011011010
6	0.222137744672527	00111110011000110111100000010100
7	0.331789743765176	00111110101010011110000001011000
8	0.475571084896906	00111110111100110111111000001110
9	0.660185655353679	00111111001010010000000111101101
10	0.892627200833881	00111111011001001000001100110111

Tablo 3.1. ve Tablo 3.2. de AP kaotik sisteminden örnek olarak 10 adımda üretilen sayıların, RSÜ için çok daha fazla üretilerek istatistiksel testlere tabi tutulması gerekmektedir. FIPS-140-1 testi için ihtiyaç duyulan bit sayısı 20.000 iken, NIST-800-22 testi için ise en az 1.000.000 bitlik bir sayı dizisine ihtiyaç duyulmaktadır.

AP Kaotik Sistemi RSÜ tasarımı için yukarıdaki bölümlerde anlatılan işlemlerin gerçekleştirilmesi gerekmektedir. AP Kaotik Sistemi 3 boyutlu olduğu için RK4 ile ayrıklaştırma sonucu 3 farklı kayan noktalı sayı elde edilecektir. Elde edilen 3 farklı kayan noktalı sayının, ikili sayı formatına dönüşümleri yapılarak RSÜ tasarımı için kullanılabilir. Ayrıklaştırma ve ikili sayı formata çevirme işlemleri sonucunda, kaotik sistemden üretilen her bir sayı sonucunda "0" ve "1" lerden oluşan 32 bitlik bir sayı dizisi elde edilmiş olur. Elde edilen bu sayı dizilerindeki "0" ve "1" ler üzerinde belirli bitlerin seçilmesi veya aritmetik/mantıksal işlem operatörleri ile rasgeleliğin artırılmasına çalışılır. Tablo 3.3.'de, AP Kaotik Sistemibirinci denkleminin ürettiği x durum değişkeninden elde edilen ilk 30 adımdaki sayıların RSÜ için ikili sayı formatına çevrilmiş durumları gösterilmiştir.

Tablo 3.3. AP Kaotik Sistemi 1.denkleminin x değişkenine ait ilk 30 adımdaki ikili sayılar

Üretilen Sayı No	AP Kaotik Sistemi (1.denklem_ilk 30 bitlik Binary Veri)
1	00111011011111010110010100111110
2	00111100100010110111000111110011
3	00111101001011000100100101111010
4	00111101101010000000011000001011
5	00111110000011111111110011011010
6	00111110011000110111100000010100
7	00111110101010011110000001011000
8	00111110111100110111111000001110
9	00111111001010010000000111101101
10	00111111011001001000001100110111
11	00111111100101101111001001110011
12	00111111110000110001100110011001
13	00111111111101101001101011011011
14	01000000000110000010000101110100
15	01000000001101101010101000111100
16	01000000101010010110001000010111
17	01000000110111101101100010010011
18	01000000100000100001011110101101
19	01000000100010001010001110000101
20	01000000100010110011111110100000
21	01000000100010100111111010011010
22	01000000100001110100110011000001
23	01000000100000101001100000010100
24	01000000011110100011100000010011
25	01000000011011101010100011010110
26	01000000011000110000110111010000
27	01000000010101111010011010011101
28	01000000010011001000001110110110
29	01000000010000011001101000101100
30	01000000001101101100111100101000

Tablo 3.3. dikkatli incelendiğinde, üretilen ikili sayılar içerisinde sol tarafta ard arda gelen bol miktarda “0” ve “1” lerin olduğu görülmektedir. Bu yüzden sayıların bit bazında alınması daha uygun olacaktır. Tabloya dikkat edilirse en sağdaki bitleri alarak RSÜ tasarımı yapmak, istatistiksel testler için en uygunu olacaktır. Eğer sadece belirli bitleri seçme yoluyla istatistiksel testlerden başarılı sonuçlar elde edilemiyorsa, ikili sayılar üzerinde çeşitli aritmetik/mantıksal işlem operatörleri kullanılarak rasgelelik arttırılabilir ve testlerden geçmesi mümkün olur.

AP Kaotik Sistemi için, sadece en sağdaki bitler arka arkaya sıralanıp elde edilen 1.000.000'luk bit dizisi ile, gerçekleştirilen NIST-800-22 istatistiksel testlerinden başarılı sonuçlar alınmıştır.

Rasgelelik testlerinin yapılabilmesi için gerekli olan 1.000.000 sayı dizisi için harcanan süre 6290.1 sn olmuştur.

3.2.5. AP Kaotik Sistemi tabanlı RSÜ'nün istatistiksel rasgelelik testleri ve sonuçları

Bu kısımda AP Kaotik Sistemi tabanlı RSÜ ile üretilen rasgele sayı dizileri uluslararası en üst düzey rasgelelik testleri olan NIST-800-22 testleri ile bir milyon bit kullanarak test edilmiş ve sonuçları bu kısımda verilmiştir.

NIST-800-22 testi içerisinde 16 farklı test bulunmaktadır. Random-Excursions ve Random Excursions Variant testlerinden dolayı NIST-800-22 testi için genellikle 1.000.000'luk sayı dizisine ihtiyaç duyulmakadır. AP Kaotik Sistemi ile elde edilen 1.000.000 sayı dizisine yönelik gerçekleştirilen NIST-800-22 testinin sonuçları Tablo 3.4.'de verilmiştir. Random-Excursions testindeki x değişkeni $-4 \leq x \leq -1$ *ve* $4 \leq x \leq 1$ arası değerler alabildiği için test sonucunda 8 adet P-değeri üretilmektedir. 8 testin sonucuda başarılı olarak gerçekleştirilmiştir, fakat tabloda örnek olması için bunlardan sadece birisi (x=-4) verilmiştir. Aynı şekilde Random Excursions Variant testide benzer olarak $-9 \leq x \leq -1$ *ve* $9 \leq x \leq 1$ arasında 18 adet P-değeri üretmektedir. Bu testdeki 18 değerin tümü için başarılı sonuçlar elde edilmiştir ve tabloda sadece x=-9 için olan değer verilmiştir [29].

Test sonuçlarının başarımı için P-değerine bakılmaktadır. Eğer $P\text{-değeri} \geq 0.001$ ise testi gerçekleştirilen sayı dizileri rasgele olarak kabul edilmekte, yani NIST-800-22 testini geçmektedir. Tablo 3.4'deki sonuçlardan da görüldüğü üzere; tüm P değerleri 0.001'den büyük oldukları için tasarlanan RSÜ testi geçmiştir ve şifrematik cihazları, şifreleme uygulamaları gibi üst düzey rasgeleliğin gerektiği uygulamalarda kullanmak için uygundur.

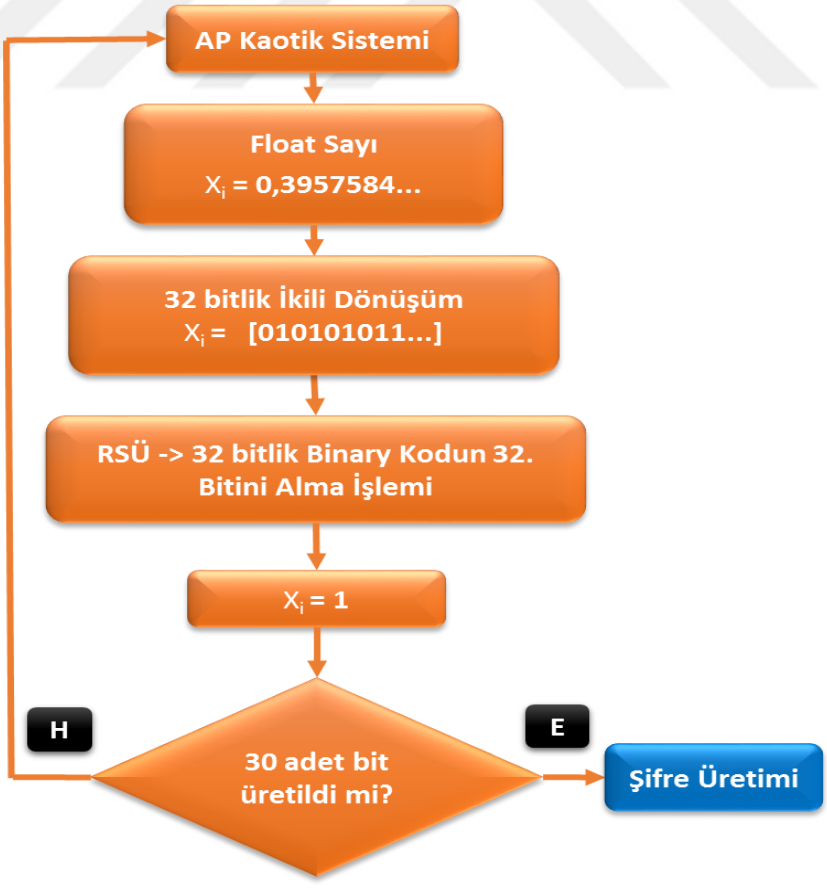
Tablo 3.4. AP Kaotik Sistemi tabanlı RSÜ'nün NIST-800-22 testleri ve Sonuçları

İstatistiksel Testler	P-değeri	Sonuç
Frequency (Monobit) Test	0.5850	Başarılı
Block-Frequency Test	0.4921	Başarılı
Cumulative-Sums Test	0.7486	Başarılı
Runs Test	0.7858	Başarılı
Longest-Run Test	0.5146	Başarılı
Binary Matrix Rank Test	0.8459	Başarılı
Discrete Fourier Transform Test	0.5772	Başarılı
Non-Overlapping Templates Test	0.0114	Başarılı
Overlapping Templates Test	0.1298	Başarılı
Maurer's Universal Statistical Test	0.6092	Başarılı
Approximate Entropy Test	0.0409	Başarılı
Random-Excursions Test	0.9727	Başarılı
Random-Excursions Variant Test	0.4973	Başarılı
Serial Test-1	0.2679	Başarılı
Serial Test-2	0.5038	Başarılı
Linear-Complexity Test	0.0881	Başarılı

BÖLÜM 4. GELİŞTİRİLEN RSÜ TABANLI BANKA ŞİFREMATİK CİHAZ TASARIMI VE KULLANICI ARAYÜZ UYGULAMASI

4.1. RSÜ Tabanlı Bit Seviyesinde Şifre Üretimi

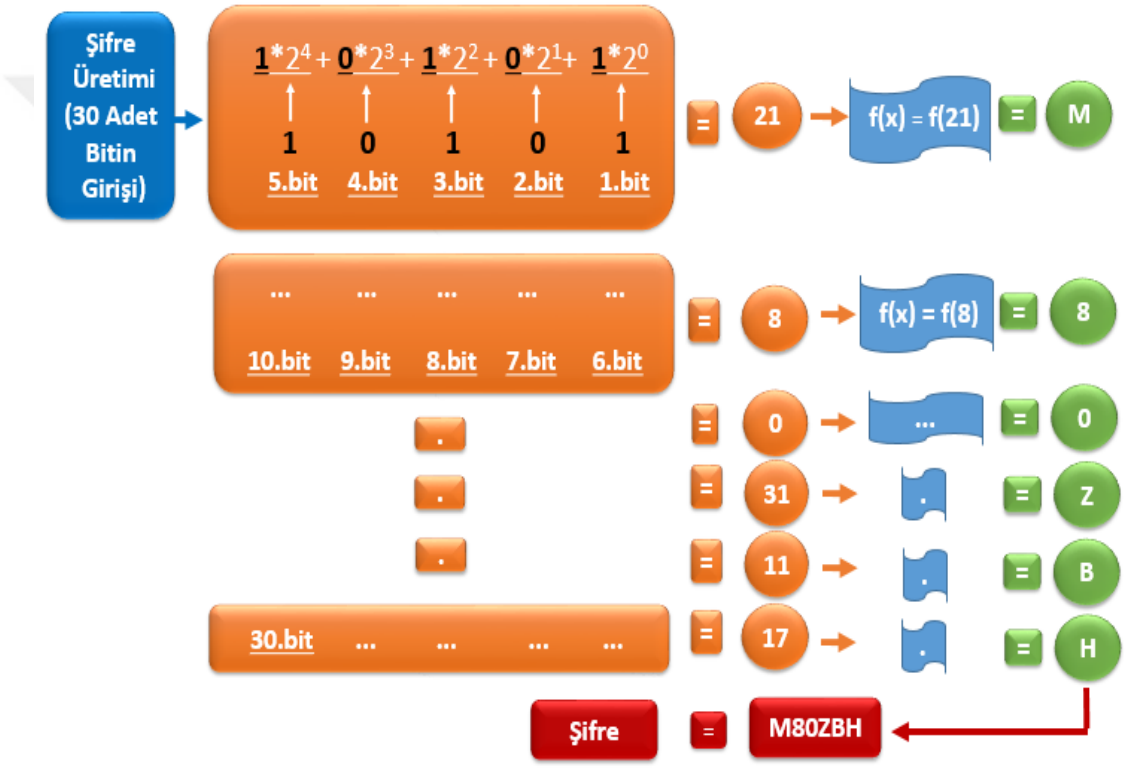
Bu bölümde istatiksel testlerden geçmiş olan bitlerden nasıl şifre üretimi yapılacağı anlatılacaktır. İlk olarak bilgisayar ortamında üretilen şifreler MATLAB programında m-file içerisine yazılan kodlardan elde edilmiştir. Şifre üretimine ait algoritma akış diyagramında bitlerin elde edilme yöntemi Şekil 4.1.'de verildiği gibidir.



Şekil 4.1. Bitlerin elde edilmesi

İlk aşama olarak AP Kaotik Sistemi RK4 algoritmasıyla ayrıklaştırılarak çözümlenmiştir. Elde edilen float (virgüllü sayı) yapıdaki bu sayılardan bir adetinin 32 bitlik dönüşümü yapılmıştır. Daha sonra elde edilen bu sayının 32. biti kayıt altına alınmıştır. Bu işlem 30 çevrim boyunca tekrarlanarak toplamda 30 adet bitin elde edilmesi sağlanmıştır.

Elde edilen bitlerin şifreye dönüştürülmesi amacıyla oluşturulan akış diyagramı Şekil 4.2.'de görüldüğü gibidir.



Şekil 4.2. Şifre üretimi akış diyagramı

Kayıt altına alınan 30 adet bitin, her beş adedi bir araya getirilerek ikilik tabanda bir sayı elde edilmiştir. Daha sonra bu sayı onluk tabana çevrilmiştir. Onluk tabandaki her sayı değeri için bir karakter dönüşümü yapılmıştır. Toplamda oluşan 32 çeşit değer için, sayı ve harflerden oluşan 32 iki adet karakter dönüşümü yapılmıştır. Şifre hanelerinin onluk tabandaki değerlerinin karakter karşılığı için kullanılan dönüşüm tablosu Tablo 4.1.'de verilmektedir.

Tablo 4.1. Dönüşüm tablosu

Onluk Tabandaki Değer	Bit Karşılığı	Karakter Karşılığı
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	G
17	10001	H
18	10010	J
19	10011	K
20	10100	L
21	10101	M
22	10110	N
23	10111	O
24	11000	P
25	11001	R
26	11010	S
27	11011	T
28	11100	U
29	11101	V
30	11110	Y
31	11111	Z

Böylece, dönüşümü tamamlanan beş adet sayının karakter karşılıkları bir araya getirilerek şifre üretimi tamamlanmaktadır.

4.1.1. RSÜ ile oluşturulan şifrelerin bilgisayar ortamında kodlanması

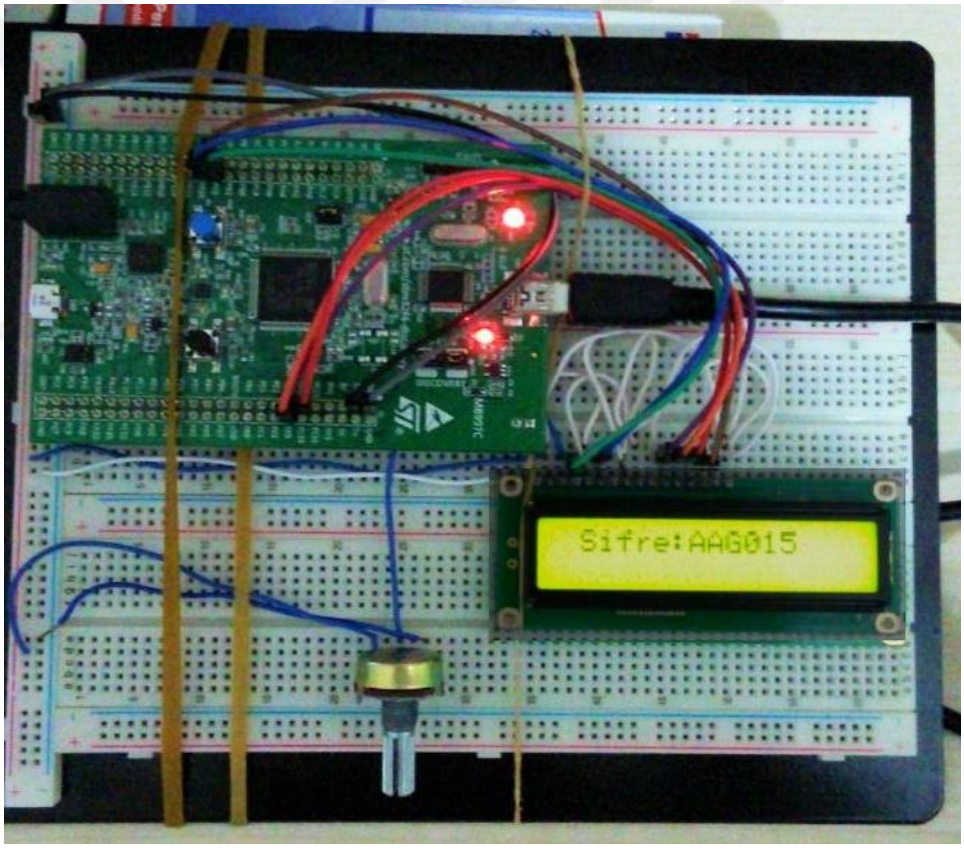
RSÜ ile üretilen bitlerin bilgisayar ortamından elektronik cihaza aktarımı sırasında, bitlerin minimum hafıza alanı kaplayacak şekilde kodlanması gerekmektedir. Aksi takdirde 30.000 adet şifre üretimi için tasarlanan şifrematik cihazı için, ihtiyaç duyulan 900.000 adet bitin mikrodenetleyici içerisinde saklanması mümkün olmayacaktır. Bu sebeple her şifreyi oluşturan 30 adet bitin onluk tabana dönüştürülerek mikrodenetleyici içerisine gömülmesi amaçlanmıştır. Böylece her şifrenin 32 bitlik hücrelerde (mikrodenetleyici verilerin 8-16-32-64 bit düzeyinde yazılmasına olanak sağlıyor) saklanması mümkün olacak ki; toplamda yaklaşık 130 KB'lık bir alan 30.000 adet şifrenin saklanması için yeterli olacaktır. İlk 10 adet şifre için yapılan örnek dönüşümün onluk tabandaki karşılığı Tablo 4.2.'de verilmektedir.

Tablo 4.2. Şifrelerin kodlanması

Şifre Numarası	Şifre Bitleri	Onluk Tabandaki Değeri
1	01010000111110011100101001000	339638600
2	00001011110110001111010101 1001	49691993
3	00010001000111111111100110101	71827253
4	11111011011010011010010101 1000	1054501208
5	11001001111111000111011111 0000	847191536
6	11000000111011100011010101 1000	809209176
7	11110101010010110111111011000	1028841432
8	10111011011011000110011001 1011	786110875
9	11111011100001110100000110 1001	1054986345
10	01000111001111110110010010 1101	298834221

4.1.2. Banka şifrematik cihaz tasarımı

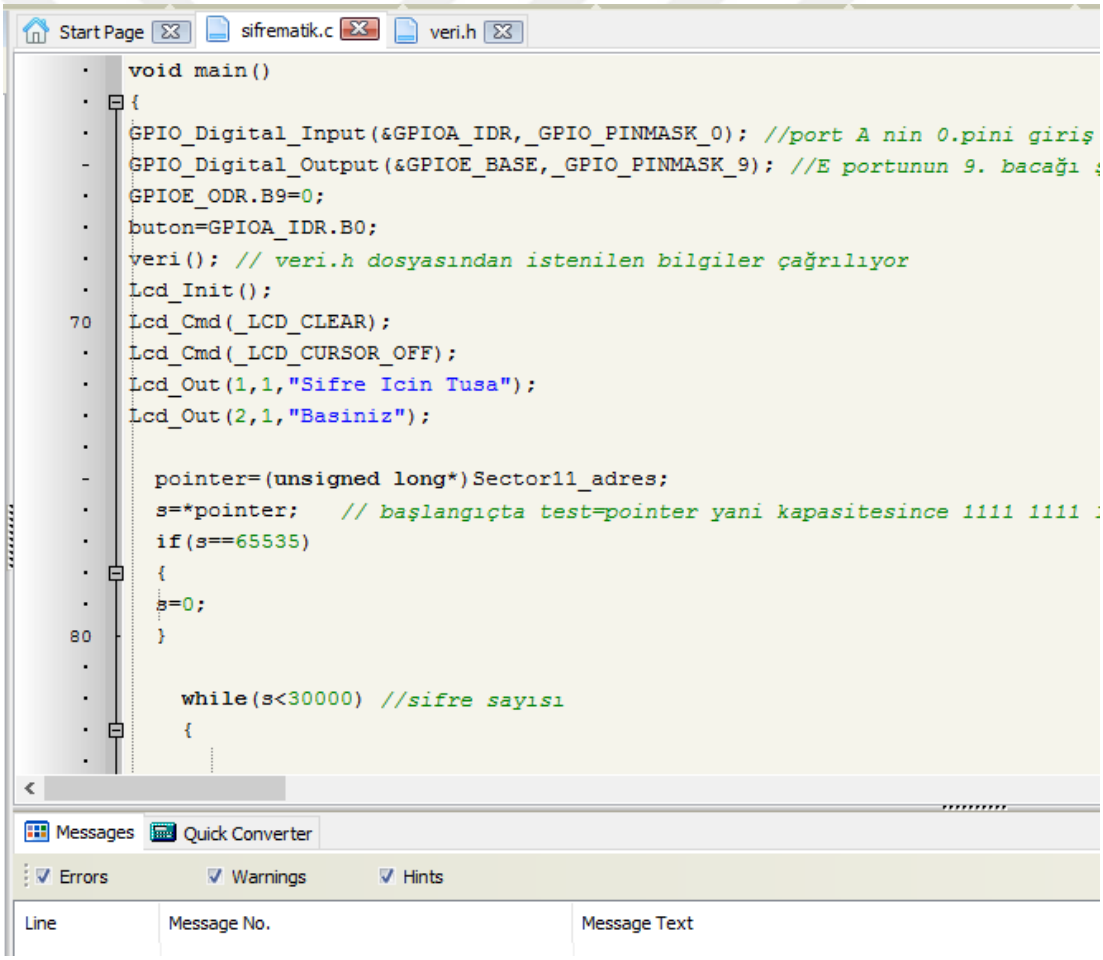
Şifrematik cihaz tasarımının elektronik olarak gerçeklenmesi için ARM mimarisine sahip Cortex-M4 işlemcili STM32F407VG discovery kartı kullanılmıştır. Şifrelerin görüntülenmesini sağlamak için 2x16 LCD ekran kullanılmıştır. LCD ekran parlaklığını ayarlamak için de 1K ohm pots kullanılmıştır. Cihaz tarafından üretilen her bir şifrelenin ekranda gösterimini sağlamak için kart üzerindeki buton kullanılmaktadır. Besleme gerilimi 5V olarak ayarlanmıştır. Şekil 4.3.'de tasarlanan ve gerçeklenen şifrematik cihazının elektronik devre yapısı görülmektedir.



Şekil 4.3. Tasarlanan ve gerçeklenen şifrematik cihazı

4.1.3. Mikrodenetleyici ile şifre üretilmesi ve şifrenin lcd ekran üzerine yansıtılması

Şifrematik cihazının yazılımsal olarak programlanması için *"MikroC"* derleyicisi kullanılmıştır. Öncelikle şifre kümesinin saklanması için alt program oluşturulmuş ve kodlanmış şifrelerin bu alana yazılması sağlanmıştır. Sonrasında karakter dönüşüm tablosu için alt fonksiyon tanımlanarak onluk tabandaki şifre hanelerinin karaktere dönüşmesi sağlanmıştır. Kodlanmış bitlerin "AND" kapısından geçirilmesi ile elde edilen bitlerden şifre üretimi yapılmıştır. Şekil 4.4.' de programlamaya dair bir ekran kesiti verilmiştir.



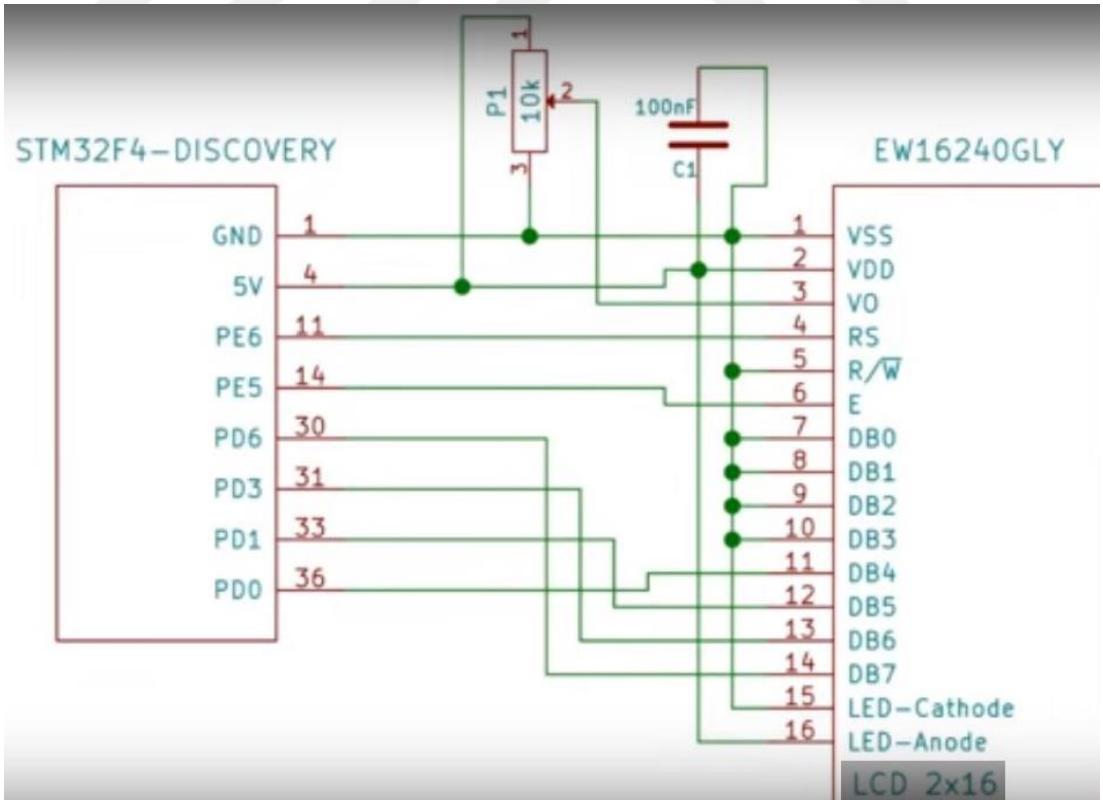
Şekil 4.4. Şifrematik cihazı programlama ekranından bir kesit

Üretilen şifrenin ekran üzerinde yansıtılabilmesi için öncelikle LCD bağlantılarının yapılması ve LCD bağlantı noktalarının programda tanımlanması gerekmektedir. LCD bağlantı tanımlamaları Şekil 4.5.'deki gibidir.

```
#include <veri.h>
#define buton GPIOD_ODR.B12 //butonun port d ye baglantısı tanımlanıyor.12.port.
char AA;
int cevir(int K)
{ ... }
// lcd bağlantı tanımlaması
sbit LCD_RS at GPIOE_ODR.B8;
sbit LCD_EN at GPIOE_ODR.B10;
sbit LCD_D4 at GPIOE_ODR.B4;
sbit LCD_D5 at GPIOE_ODR.B5;
sbit LCD_D6 at GPIOE_ODR.B6;
sbit LCD_D7 at GPIOE_ODR.B7;
// LCD tanımlama sonu
```

Şekil 4.5. Lcd ekran bağlantı tanımlamaları

Şekil 4.6.'da ise LCD panelin fiziki bağlantı şeması yer almaktadır.



Şekil 4.6. Lcd bağlantı şeması

4.2. Kullanıcı Arayüz Tasarımı ve Uygulaması

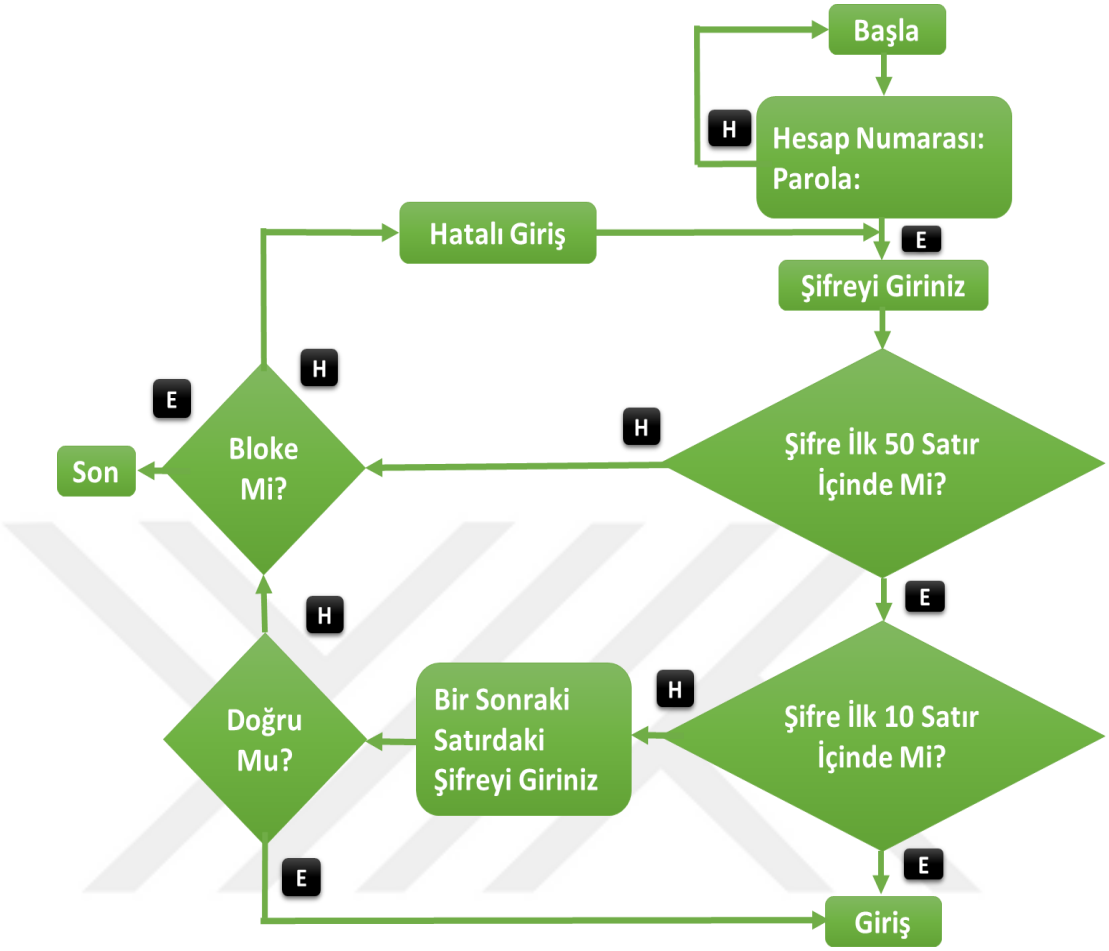
İnternet bankacılığında, kullanıcıların çeşitli işlemleri yapabilmesi için, hesaplarını yönetebileceği arayüz programlara ihtiyaç duyulmaktadır. Bu arayüz programları mobil bankacılıkta genelde IOS ve Android tabanlı uygulamalar ile yönetilirken, web bankacılığında Asp.Net yada Php tabanlı sunucular tarafından yönetilmektedir.

4.2.1. Şifre giriş ekranı tasarımı

Bankacılık işlemlerine güvenli erişimi sağlayabilmek amaçlı tasarlanan şifrematik cihazının test edilebilmesi için internet bankacılığı giriş ekranını temsil eden bir kullanıcı arayüz programı yapılmıştır. Bu arayüz programı belli bir algoritmaya göre çalışmakta olup, kullanıcının doğru giriş yapması durumunda ana sayfa erişimine izin vermektedir. Hatalı giriş yapılması durumunda ise; program çeşitli yönlendirmeler ile tekrar giriş ekranını açmaktadır.

4.2.1.1. İnternet bankacılığı ana giriş ekranı akış diyagramı

İnternet bankacılığı arayüz ekranı, kullanıcının şifrematik cihazı tarafından ürettiği şifreyi programa doğru giriş yapması durumunda, internet bankacılığı ana sayfasını temsilen tasarlanan benzetim sayfasına yönlendirme yapmaktadır. Şifrematik cihazınca üretilen bu şifrelerin tamamı belirli bir sıraya göre bankanın sistemlerinde daha önceden kayıt altında tutulmaktadır. Sistem bu şifrelerin belli bir sıraya göre girilmesi durumunda erişime izin vermektedir. Kullanıcının, cihaz tarafından üretilen ilk 10 şifreden herhangi birisini programa girmesi halinde ana sayfa doğrudan açılmaktadır. Şayet ilk 10 şifre dışındaki 11-50 aralığındaki bir şifre girilirse; program bu aşamada güvenlik amaçlı bir sonraki şifrenin de kullanıcı tarafından girilmesini isteyerek ana sayfa erişimine izin vermektedir. İlk 50 adet şifre dışındaki herhangi şifre, programa girildiğinde; program hatalı giriş uyarısı vererek erişime izin vermemektedir. Ana giriş ekranına dair oluşturulan algoritma Şekil 4.7.'deki gibidir.



Şekil 4.7. İnternet bankacılığı ana giriş sayfasını temsil eden arayüze ait akış diyagramı

4.2.1.2. Hatalı kullanım-resetleme akış diyagramı

Kullanıcının şifrematik cihazını hatalı kullanması yada kullanıcı hesabını bloke etmesi durumunda kullanıcı arayüz programının sıfırlanması gerekmektedir.

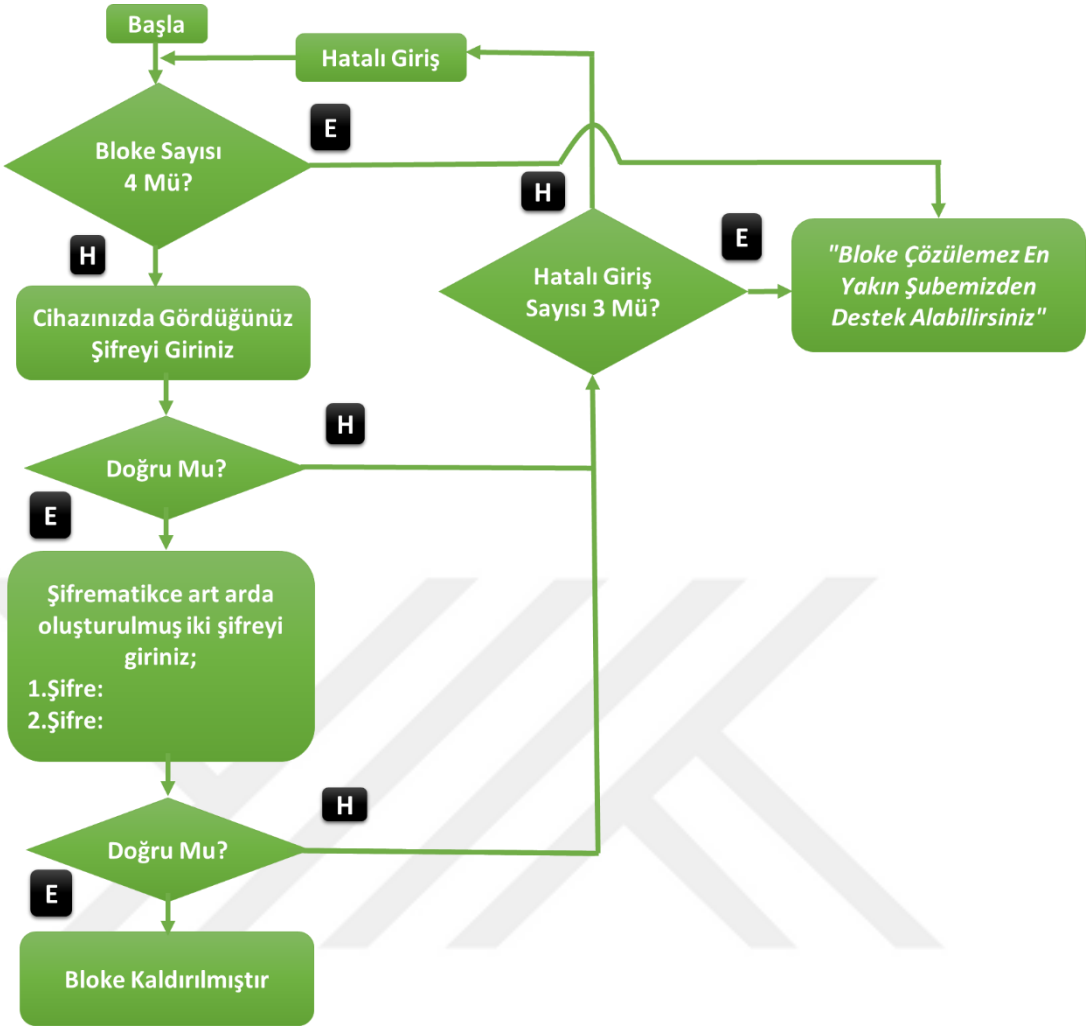
Kullanıcının şifrematik cihazı ile üretilmiş şifreleri kullanmayarak boş geçmesi durumunda arayüz programına girilecek şifrelerde program hata verecektir. Örneğin şifrematik cihazına 100 defa basılması ile elde edilecek şifreler kullanılmaz ise web arayüzü ile cihaz arasındaki senkronizasyon bozulacaktır. Şifrematik cihazı, kullanıcı arayüz programının beklemiş olduğu aralıktaki değeri çoktan geçmiş olacağı için erişime izin vermeyecektir. Web arayüzünün şifrematik cihazı ile tekrar senkronize olabilmesi için resetleme işlemi yapılması gerekmektedir.

Kullanıcının internet bankacılığı ana giriş sayfasındaki hesabına erişimi esnasında, şifreyi üç defa hatalı girmesinden dolayı da web arayüzü kilitlenecektir (bloke olacaktır). Yine bu durumu ortadan kaldırmak için resetleme işleminin yapılması gerekmektedir.

Tasarlanan akış diyagramı gereği hatalı kullanım-resetleme işlemi en fazla üç defa yapılabilecektir. Bu sayı dört olduğunda sistem hata verecek ve *"Bloke Çözülemez, En Yakın Şubemizden Destek Alabilirsiniz"* şeklinde kullanıcıyı uyaracaktır. Bloke kaldırma işlemi esnasında kullanıcının bu ekranda üç defa hatalı giriş yapması durumunda sistem yine hata verecek ve *"Bloke Çözülemez, En Yakın Şubemizden Destek Alabilirsiniz"* şeklinde kullanıcıyı uyaracaktır.

Bloke kaldırma işlemi için, kullanıcının ilk aşamada şifrematik cihazında gördüğü ilk değeri girmesi gerekmektedir. Şayet bu değer yanlış olursa sistem hata verecektir. Doğru olması durumunda ise; sistem art arda oluşurulmuş ikinci ve üçüncü şifreyi isteyecektir. Bu değerlerin yanlış girilmesi durumunda sistem yine hata verecektir. Doğru girilmesi durumunda ise *"Bloke Kaldırılmıştır"* uyarısı veren sistem kullanıcıyı ana sayfaya yönlendirecektir.

Şifrematik cihazı ile web arayüzü arasındaki senkronizasyonu sağlamak amaçlı tasarlanan hatalı kullanım-resetleme işlemi Şekil 4.8.'deki algoritma ile gerçekleştirilmektedir.



Şekil 4.8. Hatalı kullanım-resetleme akış diyagramı

4.2.2. Web ortamında arayüz uygulaması

Şifrematik cihazının kullanımını test etmek amaçlı tasarlanan örnek bir internet bankacılığı giriş sayfası Şekil 4.9.'deki gibidir. Bu web sayfası Asp.NET tabanlı sunucuda geliştirilmiş olup, eğitim amaçlı olarak yapılmıştır. Örnek bir web sayfasının çalışma algoritması Bölüm 4.2.1.'de verildiği gibidir.

Şekil 4.9. Örnek bir internet bankacılığı giriş sayfası

Şekil 4.10.'da ise şifrematik cihazınca üretilen şifrenin girilmesi istendiği ekran gösterilmektedir.

Şekil 4.10. Şifrematik cihazınca üretilen şifrenin giriş yapıldığı ekran

Şekil 4.11.'de bloke işlem menüsüne ait örnek bir ekran gösterilmektedir.

Şekil 4.11. Bloke işlem menüsüne ait örnek bir ekran

Şekil 4.12.'de başarı ile giriş yapılmış örnek bir internet bankacılığı hesap ekranı gösterilmektedir.

Şekil 4.12. Örnek bir internet bankacılığı hesap ekranı

BÖLÜM 5. SONUÇLAR ve ÖNERİLER

Sunulan Tez çalışmasında, uluslararası en üst standart olan NIST-800-22 istatistiksel testinden başarıyla geçmiş olan AP Kaotik Sistemi tabanlı RSÜ kullanılmıştır. Bu RSÜ yardımı ile yeni bir şifre üretme algoritması geliştirilmiş olup, fiziksel ve sanal ortamda uygulaması gerçekleştirilmiştir.

Tez çalışmasının ilk kısmında AP Kaotik Sisteminin analizleri yapılmıştır. Daha sonra NIST testlerine tabi tutulmuş RSÜ yardımıyla yeni bir şifre üretme algoritması geliştirilmiştir. Bu algoritmanın bilgisayar ve mikrodenetleyici ortamında gerçeklenmesi sağlanmıştır. Literatürde şifre üretme üzerine yapılan "*ARM İşlemciler İle Tek Kullanımlık Şifre Uygulamasının Gerçekleştirilmesi"* [50] isimli çalışma incelendiğinde, uluslararası en üst standart olan NIST-800-22 testlerinin yapılmadığı, üretilen şifreler için özetleme fonksiyonlarından faydalanıldığı görülmüştür. Tez çalışmasında kullanılmış olan STM32F407VG mikrodenetleyicisi aynı zamanda gerçek rasgele sayı üreteci birimine sahiptir. Bu kart üzerindeki RNG birimi tarafında üretilen sayıların sadece FIPS PUB 140-2 (2001 October 10) testinden %99 başarı oranı ile geçtiği görülmüştür [56]. Bu tezdeki AP Kaotik Sistemi tabanlı RSÜ yardımıyla oluşturulan şifre üretme algoritmasında kullanılan bitlerin, NIST testlerinden geçirilmesinden ötürü, şifrematik cihazınca üretilen şifreler gerçek rasgeleliğe sahiptir. Bu nedenle bu tezde yapılan tasarım, internet bankacılığına giriş işlemlerinde yüksek güvenlik düzeyine ulaşmayı sağlayacaktır. Şifrematik cihazınca üretilen 30.000 adet şifrenin birbirleri arasında benzerlik testleri yapıldığında; % 99,9966 oranında farklı oldukları görülmüştür. Şifre üretme algoritmasının kullanım alanları göze alındığında, tez çalışmasının şifrematik cihazı üzerine yapılmış az sayıdaki araştırmaya kaynak olacak nitelikte olduğu görülmüştür.

Sunulan tez çalışmasının son aşamasında, şifrematik cihazının gerçek hayatta uygulanabilirliğini test etmek amacıyla örnek bir kullanıcı arayüz programı yapılmıştır. Bu arayüzü oluşturabilmek için öncelikle bir akış diyagramı oluşturulmuş, sonrasında ekran arayüzleri tasarlanmıştır. Daha sonra programın yazılımsal olarak gerçekleştirilmesi sağlanmıştır.

Kaos tabanlı sistemlerin sınırsız sayıda değişik periyodik salınımlar içermesi, genlik ve frekansının tespit edilememesi, sınırlı bir alanda değişen işaretler içermesi ve başlangıç şartlarına hassas bağımlılığı gibi genel özellikleri göze alındığında; AP Kaotik Sisteminin sadece başlangıç şartlarını değiştirerek yeni rasgele sayı üretimlerinin yapılabileceği, yeterince farklı sayıda kullanıcı için farklı şifrelere sahip başka şifrematik cihazlarının elde edilebileceği görülmektedir.

Cihazın tasarım maliyetleri ele alındığında, yaklaşık 8 adet 9V pil tutarında, orta düzey bir maliyet ile üretimi mümkün olacaktır. Toplu üretim yada entegre tasarımların göz önünde bulunduruluduğu düşünülürse, çok daha düşük maliyetler ile şifrematik cihazının üretilmesi söz konusu olacaktır. Bankaların bu cihazı tahsis etmesindeki amacının güvenlik ve müşteri memnuniyeti olduğu göz önünde tutulursa, cihazın maliyet paylaşımlı tedarik süreçleri ile müşteriyle buluşturulması mümkündür. Ayrıca bu tasarım FPGA, DSP ve farklı mikrodenetleyiciler üzerinde de gerçekleştirilebilir.

Son zamanlarda geliştirilen güvenlik uygulamalarından biri de mobil cihazlar üzerindeki şifrematik yazılımlarıdır. Bu yazılımlar şifrematik cihazları üzerindeki algoritmaların aynılarını mobil ortamda gerçekleştirmektedirler. Bu durum göze alındığında kaos tabanlı RSÜ yardımı ile geliştirilen bu tasarımın mobil cihazlar üzerinde gerçekleştirilmesi mümkün olmaktadır. Ayrıca sms bankacılığında kullanıcının cep telefonuna gönderilen tek kullanımlık şifreler de belli bir fonksiyona göre üretilmekte ve kullanıcıya iletilmektedir. Kaos tabanlı RSÜ yardımı ile geliştirilen bu tasarımın yine bu fonksiyonlar gibi kullanılarak şifre üretmesi mümkündür.

Teknolojinin hızla gelişmesiyle beraber, mobil çözümlerin en etkili tasarım alanı olan cep telefonlarının gizlilik ve güvenlik açısından casus yazılımlara açık olması, mobil bankacılık ve internet bankacılığının çeşitli riskler altında olduğunu doğrular niteliktedir. Ayrıca müşteri ile banka arasındaki köprünün, üçüncü kişi olarak gsm operatörlerince kurulması da güvenlik zaafiyetlerine neden olmaktadır. Belirtilen çeşitli riskler göze önünde tutulduğunda; şifrematik cihazının bireysel kullanıcı ve küçük ölçekli işletmelerden ziyade, orta ve büyük ölçekli şirketlerce kullanılmasının önemli bir güvenlik düzeyi oluşturacağı düşünülmektedir.

KAYNAKLAR

[1] Holmes PJ., Poincare celestial mechanics, dynamical-systems theory and "chaos", Phys. Rep., 1990;193(3):138-163.

[2] Pehlivan I., Yeni Kaotik Sistemler: Elektronik Devre Gerçeklemeleri, Senkronizasyon ve Güvenli Haberleşme Uygulamaları, Sakarya Üniv. Doktora Tezi, 2007.

[3] Zhengxing, H., Wei, D., Huilong, D., Haomin, L., Similarity measure between patient traces for clinical pathway analysis: problem, method, and applications. IEEE J. of Biomedical and Health Inf., 18(1):4-14, 2014.

[4] Xiong, A., Zhao, X., Han, J., Liu, G., Application of the chaos theory in the analysis of EMG on patients with facial paralysis. In Robot Intelligence Tech. and App., Springer, 274:805-819, 2014.

[5] Ching, C., Chun, L., Shyan, L., Yen, C., Cheng, C., A chaotic theorectical approach to ECG-based identity recognition. IEEE Comp. Intelligence Mag., 9(1):53-63, 2014.

[6] Yanhua, H., Spencer, PS., Shore, KA., wideband chaos with time-delay concealment in vertical-cavity surface-emitting lasers with optical feedback and injection. IEEE J. of Quantum Electr., 50(4):236-242, 2014.

[7] Zexin, K., Jiang, S., Lin, M., Yanhui, Q., Shuisheng, J., Multimode synchronization of chaotic semiconductor ring laser and its potential in chaos communication. IEEE J. of Quantum Electr., 50(3):148-157, 2014.

[8] Li, H., Hu, Y., Observer-based synchronization for laser systems. IEEE J. of Quantum Electr., 50(5):372-378, 2014.

[9] U, SH., Kang, HS., Kim, YT., Hyun, CH., Park, M., Fuzzy adaptive modular design of uncertain chaotic duffing oscillators. Int. J. of Control Automation and Sys., 12(1):188-194, 2014.

[10] Natarajan, S., Rajasekar, N., An FPGA chaos-based PWM technique combined with simple passive filter for effective EMI spectral peak reduction in DC-DC converter. Advances in Power Electr., 1-11, 2014.

[11] Lones, MA., Fuente, LA., Turner, AP., Caves, LSD., Stepney, S., Smith, SL., Tyrrell, AM., Artificial biochemical networks: evolving dynamical systems to control dynamical systems. IEEE Trans. on Evolutionary Comp., 18(2):145-166, 2014.

[12] Jin, L., Mei, J., LI, L., Chaos control of parametric driven Duffing oscillators. Applied Physics Lett., 104(13):1011-1015, 2014.

[13] Wan, L., Luo, XS., Zeng, SY., Zhang, B., Global exponential stabilization for chaotic brushless DC motors with a single input. Springer Nonlinear Dyn., 77(1-2):209-212, 2014.

[14] Arashar, A., Singh, R., Panigrahi, PK., Muralidhar, K., Chaotic flow in an aortic aneurysm. J. of Applied Phys., 113(21):1-14, 2013.

[15] Pignolet, A., Ferroelectrics chaotic memory. Nature Phys., 10(1):9-11, 2014.

[16] Hemmati, M., Amjady, N., Ehsan, M., System modeling and optimization for islanded micro-grid using multi-cross learning-based chaotic differential evolution algorithm. Int. J. of Elect. Power and Energy Syst., Elsevier, 56:349-360, 2014.

[17] Pomares, J., Perea, I., Torres, F., Dynamic visual servoing with chaos control for redundant robots. IEEE Trans. on Mechatronics, 19(2):423-431, 2014.

[18] Akizawa, Y., Yamazaki, T., Uchida, A., Harayama, T., Sunada, S., Arai, K., Yoshimura, K., Davis, P., Fast RNG with bandwidth-enhanced chaotic semiconductor lasers at 8 times 50Gb/s. IEEE Photonics Tech. Lett., 24(12):1042-1044, 2012.

[19] Uyaroğlu, Y., Pehlivan, I., Nonlinear Sprott94 case a chaotic equation: synchronization and masking communication applications. Computers and Elect. Eng., Elsevier, 36:1093-1100, 2010.

[20] Eroğlu, C., Implementation of synchronized chaotic systems by FPGA. Graduate Sch. of Eng. and Sci. of Izmir Inst. of Tech., Izmir, Turkey, 2007.

[21] Kharel, R., Busawon, K., Aggoune, W., Ghassemloy, Z., Implementation of a secure digital chaotic communication scheme on a DSP board. 7th Int. Symp. on Comm. Syst. Networks and Digital Signal Process., 212-216, 2010.

[22] Yiwei, Z., Zexiang, L., Xinjian, Z., A chaos-based image encryption ASIC using reconfigurable logic. IEEE Asia Pacific Conf. on Circuits and Syst., 1782-1785, 2008.

[23] Paolo, A., Sebastiano, DF., Luigi, F., Mattia, F., Luca, P., Guido, V., Reactive navigation through multiscroll systems: from theory to real-time implementation. Autonomous Robots, Springer, 25(1-2):123-146, 2008.

[24] Koyuncu, İ., Kriptolojik Uygulamalar İçin Fpga Tabanlı Yeni Kaotik Osilatörlerin ve Gerçek Rasgele Sayı Üreteçlerinin Tasarımı Ve Gerçeklenmesi. Doktora Tezi, Sakarya Üniversitesi, 2014.

[25] Wieczorek, PZ., Golofit, K., Dual-metastability time-competitive TRNG. IEEE Trans. on Circuits and Syst., 61(1):134-145, 2014.

[26] Fischer, V., Drutavosky, M., Simka, M., Bochard, N., High performance TRNG in sltera stratix FPLDs.Field Program. Logic and App., Springer, 555–564, 2004.

[27] Istvan, H., Suciu, A., Cret, O., FPGA based TRNG using automatic calibration. Intelligent Comp. Comm. and Proc., IEEE 5th Int. Conf. on ICCP, 373-376, 2009.

[28] Akif Akgul, Haris Calgan, Ismail Koyuncu, Ihsan Pehlivan, Ayhan Istanbullu, "Chaos-based engineering applications with a 3D chaotic system without equilibrium points", Nonlinear Dynamics, Vol 84, Issue 2, 481-495, April 2016.

[29] Akgül, A., Yeni Kaotik Sistemler İle Rasgele Sayı Üreteci Tasarımı ve Çoklu-Ortam Verilerinin Yüksek Güvenlikli Şifrelenmesi. Doktora Tezi, Sakarya Üniversitesi, 2015..

[30] Akif Akgül, İhsan Pehlivan, "A New Three-Dimensional Chaotic System Without Equilibrium Points, Its Dynamical Analyses And Electronic Circuit Application", Technical Gazette, Volume 23, No: 1, February 2016.

[31] Lorenz, EN., Deterministic nonperiodic flow. J. of the Atmospheric Sci., 20:130-141, 1963.

[32] LI, TY., Yorke, JA., Period three implies chaos. The American Math. Monthly, 82(10):985–992, 1975.

[33] Rössler, OE., An equation for continuous chaos. Physics Lett., 57(5):397-398, 1976.

[34] Rössler, OE., Continuous chaos-four prototype equations. Annals of the New York Academy of Sci., 316:376-392, 1979.

[35] Matsumoto, T., Chua, LO., Tanama, S., Simplest chaotic nonautonomous circuit. Physical Rev., 30:1155-1157, 1984.

[36] Saritaş, E., Karataş, S., Her yönüyle FPGA ve VHDL. Palme Yayıncılık, Ankara, 2013.

[37] Demirkol, A., Kaotik osilatör girişli ADC tabanlı rasgele sayı üreteci. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, 2007.

[38] Güven, P., Otonom olmayan kaotik sistemlerde rasgele sayı üretiminin incelenmesi. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, 2006.

[39] Federal information processing standards publication, Security requirements for cryptographic modules. FIPS PUB 140-1, 1994. http://csrc.nist.gov/publications/fips/fips1401.htm, Erişim Tarihi: 06.06.2014.

[40] http://www.csm.ornl.gov/~dunigan/fips140.txt, Erişim Tarihi: 26.05.2014.

[41] A statistical test suite for random and pseudo RNGs for cryptographic applications. National institute of stand. and tech.,NIST-800-22, 2001. http://csrc.nist.gov/publications/ nistpubs/800-22/sp-800-22-051501.pdf, Erişim Tarihi: 06.06.2014.

[42] Yayık, A., Kutlu, Y., Improving PNRG using artificial neural networks. IEEE 21st Signal Processing and Comm. App. Conf., 1-4, 2013.

[43] Avaroğlu, E., Donanım tabanlı rasgele sayı üretecinin gerçekleştirilmesi. Doktora Tezi, Fırat Üniversitesi, 2014.

[44] Kriptografiye giriş ders notları. Uygulamalı Matematik Enstitüsü Kript. Böl., ODTÜ, 2004.

[45] Maurer, UM., A universal statistical test for random bit generators. J. of Cryptology, 5(2):89-105, 1992.

[46] Koç, S., Dal M. Ali 2013. MikroC ile ARM Programlama, İstanbul.

[47] Alvarez, G., LI, S., Some Basic Cryptographic Requirements For Chaos-Based Cryptosystems. International Journal of Bifurcation and Chaos, 16(8): 2129–2151, 2006.

[48] Akgul, A., Yüksek güvenlikli kızılötesi iletişim uygulaması. Yüksek Lisans Tezi, Sakarya Üniversitesi, 2011.

[49] PRO-G, bilişim güvenliği, sürüm 1.1, Pro-G Bilişim Güvenliği ve Araştırma Ltd., http://www.pro-g.com.tr/whitepapers/bilisim-guvenligi-v1.pdf, 2003.

[50] Türk Ö., ARM işlemciler ile tek kullanımlık şifre uygulamasının gerçekleştirilmesi. Yüksek Lisans Tezi, Fırat Üniversitesi, 2011.

[51] Adıgüzel C. G., Güvenlik endişesinin internet bankaciliği kullanimina etkisi ve vakifbank müşterilerine yönelik bir araştirma. Yüksek Lisans Tezi, Gazi Üniversitesi, 2009.

[52] Cuomo KM., Oppenheim AV., Circuit Implementation of Synchronized Chaos with applications to Communication, Phys. Rev. Lett., 1993;71:65-68.

[53] Şahin, I., A 32-bit floating-point module design for 3D graphic transformations. Sci. Research and Ess., 5(20):3070-3081, 2010.

[54] Nien, HH., Huang, CK., Changchien, SK., Shieh, HW., Chen, CT., Tuan, YY., Digital color image encoding and decoding using a novel chaotic random generator. Chaos, Solitons & Fract., 32(3):1070-1080, 2007.

[55] Büyüksaraçoğlu, F., Buluş, E., Sözde rastsal sayı üretiminin kriptografik açıdan incelenmesi. TMMOB Elektrik Müh. Odası IV. İletişim Tekn. Ul. Semp., Adana, 2009.

[56] STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx advanced ARM®-based 32-bit MCUs. Reference manual.www.st.com.

ÖZGEÇMİŞ

Serkan AKKAYA, 13.06.1988 tarihinde Şişli'de doğdu. İlköğrenimini İstanbul'da tamamladı. 2005 yılında Kartal Hacı Hatice Bayraktar Lisesi, Fen Bilimleri Bölümünden mezun oldu. 2006 yılında başladığı Sakarya Üniversitesi Elektrik-Elektronik Mühendisliği bölümünü 2010 yılında tamamladı. 2010–2011 yılları arasında Yedek Subay olarak askerlik görevini ifa etti. 2011 yılında, Haberleşme ve Telekomünikasyon Sektöründe başladığı iş hayatına aynı alanda devam etmektedir. Evli ve bir çocuk babasıdır.