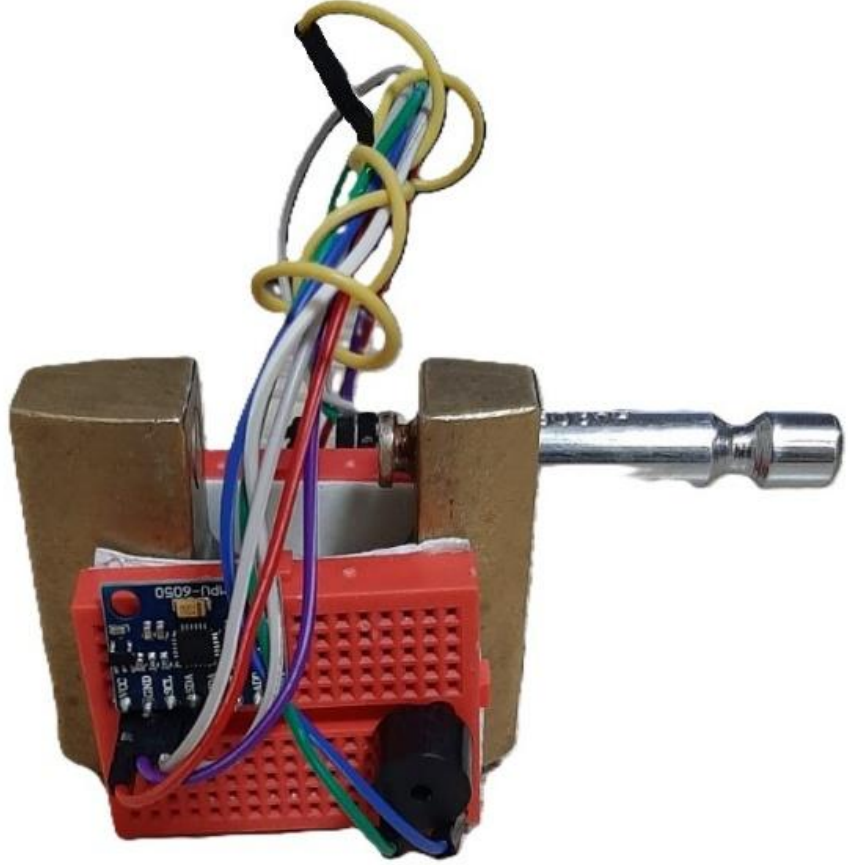


ALARMLI DİSK KİLİDİ



Hazırlayan:

Hasan Hüseyin Turna

İÇİNDEKİLER

Sayfa No:

| | |
|---|---|
| 1.GİRİŞ..... | 3 |
| 1.1 Projenin Amaçları ve Hedefleri | 3 |
| 1.2 Projenin Genel Tasarımı ve İşlevselliği | 3 |
| 1.3 Kullanılan Teknolojiler ve Platformlar | 4 |
| 2.Literatür Taraması | 4 |
| 2.1 Benzer Projelerin İncelenmesi | 4 |
| 2.3 Proje Bağlamında Önemli Teorik Konseptler | 4 |
| 3.Proje Tasarımı | 4 |
| 3.1 Donanım ve Yazılım Gereksinimleri | 4 |
| 3.2 Algoritmaların ve Modüllerin Tasarımı | 4 |
| 4.Uygulama | 4 |
| 4.1 Kullanılan Programlama Dili ve Araçlar | 4 |
| 4.2 Projenin Uygulama Aşamaları | 5 |
| 4.3 Algoritmaların Modülleri ve Tasarımları | 5 |
| Projenin Kaynak Kodu: | 6 |
| 5.Sonuçlar ve Tartışma | 8 |
| Gelişim aşaması : | 8 |
| 6. Gelecek Çalışmalar ve İyileştirmeler | 8 |
| Entegre Batarya Sistemi: | 8 |
| 3D Yazıcı ile Koruyucu Kalıp Entegrasyonu: | 9 |
| Mobil Uygulama Entegrasyonu: | 9 |
| Daha Hassas Sensörler: | 9 |
| Yüksek Sesli Alarm Modülü: | 9 |
| 7.Kaynaklar | 9 |

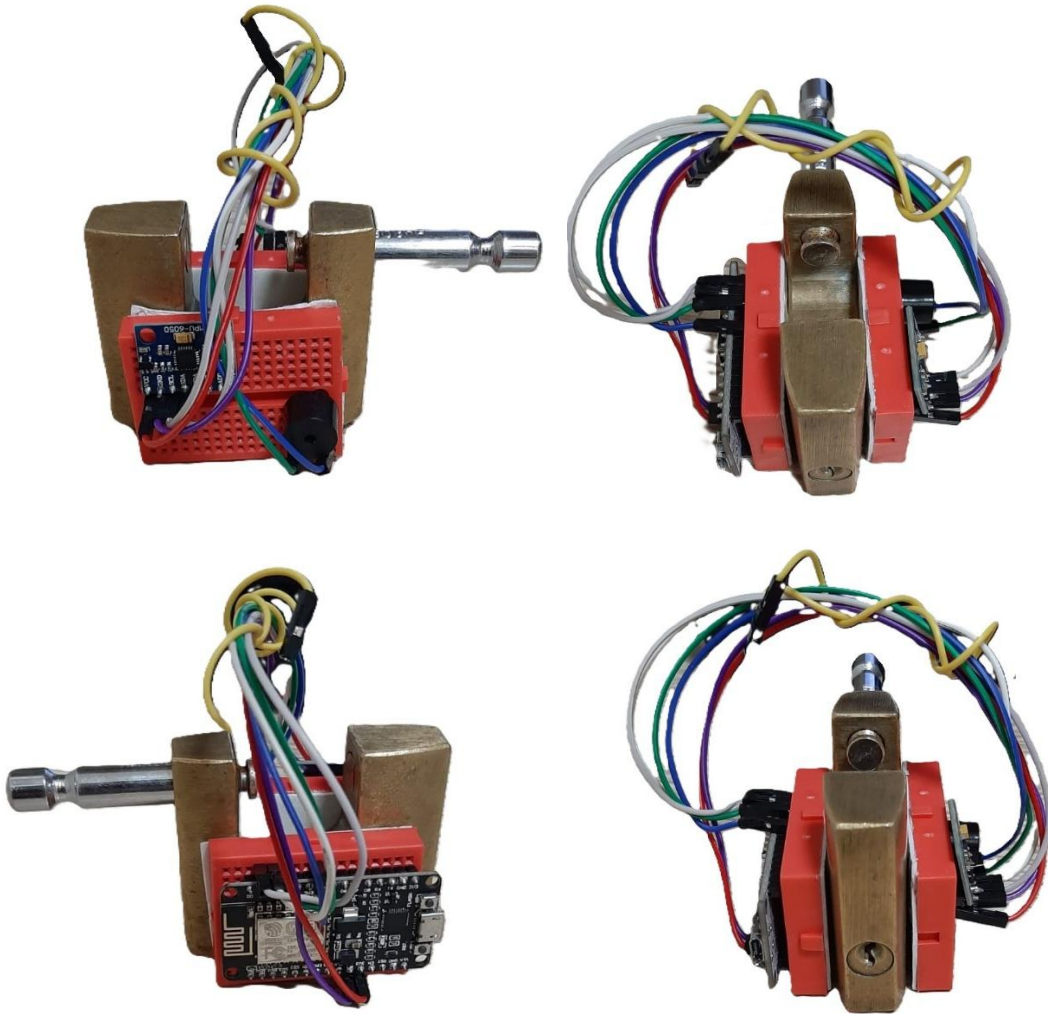
1.GİRİŞ

1.1 Projenin Amaçları ve Hedefleri

Bu proje, motosiklet disk kilidi için bir alarm sistemi geliştirmeyi amaçlamaktadır. Amaç, güvenlik ve maliyet avantajı sunan, dayanıklı bir çözüm oluşturulmasıdır. Geleneksel zincir kilitlerin ağırlığı ve kullanım zorluğu nedeniyle, hafif ve etkili bir alternatif olarak tasarlanmıştır. Proje, titreşim algılayarak alarmı tetikleyen bir sistem oluşturmayı hedefler.

1.2 Projenin Genel Tasarımı ve İşlevselliği

Proje, üç temel bileşen üzerine kuruludur: MPU6050 hareket sensörü, ESP8266 mikrodenetleyicisi ve buzzer. MPU6050 sensörü, disk kilidi üzerindeki en ufak hareketleri algılayarak üç eksenli ivme verilerini sağlar. Bu veriler ESP8266 tarafından işlenir. ESP8266, aynı zamanda kablosuz ağ bağlantısı kurarak Adafruit IO platformuna veri gönderimi ve alarmı tetikleme görevlerini yerine getirir. Sistem, hareket algılandığında hem sesli uyarı (buzzer) hem de veri kaydı (Adafruit IO) olmak üzere iki farklı tepki verir. Bu şekilde, hem anlık uyarı hem de geçmişe yönelik hareket takibi sağlanır.



1.3 Kullanılan Teknolojiler ve Platformlar

Bu projede kullanılan teknolojiler:

- **ESP8266:** Kablosuz bağlantı ve kontrol için kullanılmıştır.
- **MPU6050:** Hareket algılama.
- **Buzzer:** Sesli uyarı.
- **Adafruit IO:** IoT veri kayıt ve görüntüleme.
- **Arduino IDE:** Yazılım geliştirme ortamı.

2.Literatür Taraması

2.1 Benzer Projelerin İncelenmesi

Alarmlı disk kilidi sistemleri genellikle mekanik dayanıklılık ve alarm entegrasi sağlar. Ancak, uygun maliyetli çözümler güvenlik zafiyetleri taşıırken, üyelik tabanlı premium cihazlar yüksek maliyetlidir. Bu proje, IoT teknolojilerini kullanarak etkili bir çözüm sunmayı hedefler.

2.3 Proje Bağlamında Önemli Teorik Konseptler

Hareket algılama, kablosuz iletişim ve IoT entegrasyonu gibi temel konseptler bu projede kullanılmıştır. Bu konseptler, güvenlik uygulamaları için kritik öneme sahiptir.

3.Proje Tasarımı

3.1 Donanım ve Yazılım Gereksinimleri

- Donanım: ESP8266, MPU6050, buzzer, kablolar.
- Yazılım: Arduino IDE, Adafruit IO.

3.2 Algoritmaların ve Modüllerin Tasarımı

Proje, şu algoritmalara dayanır:

- Hareket algılama.
- Veri kayıt ve gönderim.
- Alarm tetikleme.

4.Uygulama

4.1 Kullanılan Programlama Dili ve Araçlar

- **Arduino C:** Kodlama dili.
- **Adafruit IO:** Veri kayıt ve görüntüleme platformu.

4.2 Projenin Uygulama Aşamaları

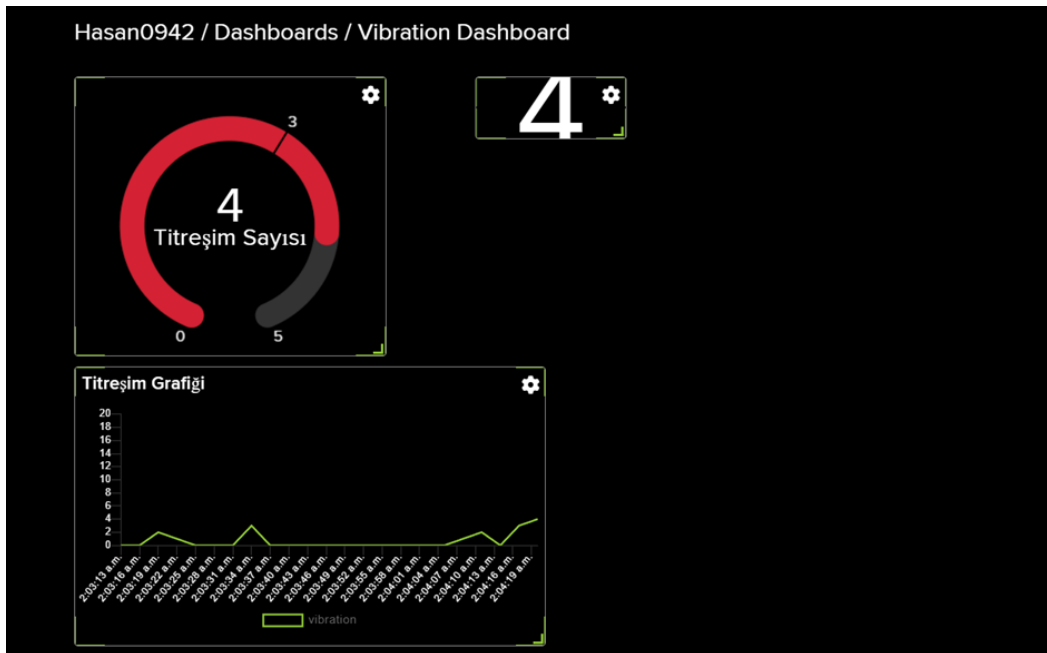
1. Donanım kurulumu.
2. Yazılım geliştirme.
3. Test ve hata ayıklama.

4.3 Algoritmaların Modülleri ve Tasarımları

Kod, MPU6050 sensöründen gelen verilere dayalı olarak alarmı tetikler ve Adafruit IO'ya veri kayıt eder.

```
Output Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'C
02:01:32.930 -> Son 5 Titreşimden Aktif Olanlar: 0
02:01:32.471 -> Son 5 Titreşimden Aktif Olanlar: 0
02:01:32.585 -> Son 5 Titreşimden Aktif Olanlar: 0
02:01:32.686 -> Son 5 Titreşimden Aktif Olanlar: 0
02:01:32.804 -> Son 5 Titreşimden Aktif Olanlar: 0
02:01:32.904 -> Son 5 Titreşimden Aktif Olanlar: 1
02:01:33.038 -> Son 5 Titreşimden Aktif Olanlar: 1
02:01:33.119 -> Son 5 Titreşimden Aktif Olanlar: 2
02:01:33.253 -> Son 5 Titreşimden Aktif Olanlar: 2
02:01:33.352 -> Son 5 Titreşimden Aktif Olanlar: 3
02:01:34.406 -> Son 5 Titreşimden Aktif Olanlar: 2
```

Serial Monitor üzerinden alınan anlık veriler.



Adafruit IO'nun feed'i üzerinden dashboard'da anlık verilerin görselleştirilmesi.

Projenin Kaynak Kodu:

```
32 void setup() {
33     Wire.begin();
34     mpu.initialize();
35
36     pinMode(buzzerPin, OUTPUT);
37     digitalWrite(buzzerPin, LOW);
38
39     Serial.begin(115200);
40     Serial.println("Sistem baslatiliyor...");
41
42     // MPU6050'nin doğru çalışıp çalışmadığını kontrol et
43     if (!mpu.testConnection()) {
44         Serial.println("MPU6050 bağlantısı basarisiz!");
45         while (1);
46     }
47
48     // Adafruit IO bağlantısını başlat
49     io.connect();
50
51     // Bağlantı durumunu kontrol et
52     while (io.status() < AIO_CONNECTED) {
53         Serial.print(".");
54         delay(500);
55     }
56     Serial.println("\nAdafruit IO'ya baglandi!");
57 }
58
```

sketch_dec11a_copy_20241211022605.ino

```
1  #include <Wire.h>
2  #include <ESP8266WiFi.h>
3  #include "AdafruitIO_WiFi.h"
4  #include <MPU6050.h>
5
6  // Adafruit IO kimlik bilgileri
7  #define IO_USERNAME "Hasan0942" // Adafruit IO kullanıcı adı
8  #define IO_KEY "aio_WPDZ86W6XoMoV072TP0o2We35B5V" // Adafruit IO anahtarı
9
10 // Wi-Fi bilgileri
11 #define WIFI_SSID "Galaxy A507087" // Wi-Fi ağ adı
12 #define WIFI_PASS "hasan1234" // Wi-Fi şifresi
13
14 // Adafruit IO Wi-Fi istemcisi
15 AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
16
17 // Feed tanımı
18 AdafruitIO_Feed *vibrationFeed = io.feed("vibration");
19
20 MPU6050 mpu;
21
22 const int buzzerPin = D3; // Buzzer'ın bağlı olduğu pin
23
24 const int threshold = 3; // Son 5 okumadan en az 3'ü titreşim ise alarm çalışacak
25 uint8_t vibrationHistory = 0; // Son 5 titreşim okumasını tutan bit maskesi
26 const float accelThreshold = 1.5; // İvme eşik değeri (g cinsinden)
27
28 unsigned long previousMillis = 0; // Son veri gönderim zamanını saklar
29 const unsigned long interval = 3000; // Adafruit IO'ya veri gönderim aralığı (ms)
30 unsigned long currentMillis = millis();
31
```

```

59 void loop() {
60     io.run(); // Adafruit IO'yu çalıştır
61
62     // MPU6050'den ivme verilerini oku
63     int16_t ax, ay, az;
64     mpu.getAcceleration(&ax, &ay, &az);
65
66     // İvme değerlerini g cinsine dönüştür
67     float accelX = ax / 16384.0;
68     float accelY = ay / 16384.0;
69     float accelZ = az / 16384.0;
70
71     // Toplam ivme değişimini hesapla
72     float totalAccel = sqrt(accelX * accelX + accelY * accelY + accelZ * accelZ);
73
74     // Titreşim algılama
75     int sensorValue = (totalAccel > accelThreshold) ? 1 : 0;
76
77     // Bit maskesini kaydır ve yeni değeri ekle
78     vibrationHistory = (vibrationHistory << 1) | sensorValue;
79
80     // Son 5 bitin titreşim sayısını kontrol et
81     int vibrationCount = __builtin_popcount(vibrationHistory & 0x1F); // Son 5 biti kontrol et
82
83     // Seri monitöre yazdır
84     Serial.print("Son 5 Titreşimden Aktif Olanlar: ");
85     Serial.println(vibrationCount);
86
87     // Alarm kontrolü
88     if (vibrationCount >= threshold) {
89         alarmCalistir();
90     }

```

```

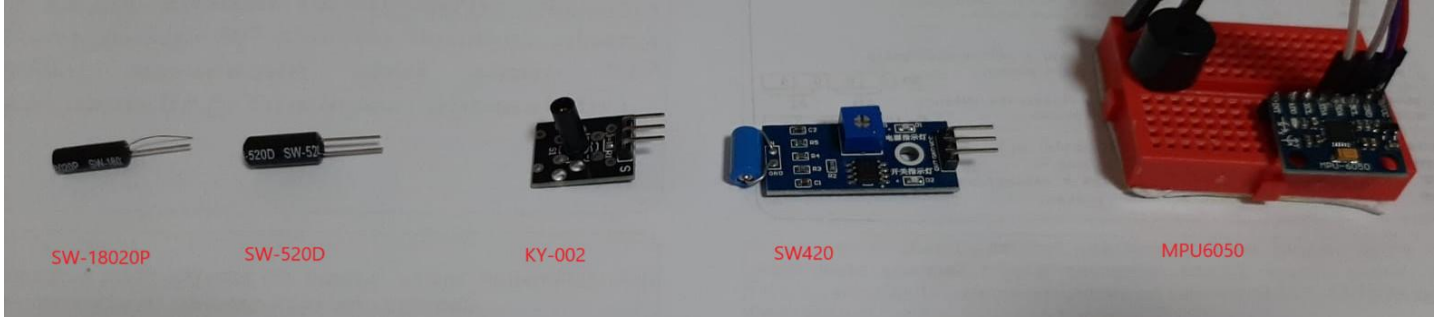
91
92     // Veri gönderim kontrolü
93     currentMillis = millis();
94     if (currentMillis - previousMillis >= interval) {
95         previousMillis = currentMillis;
96
97         // Adafruit IO'ya veri gönder
98         vibrationFeed->save(vibrationCount);
99     }
100 }
101
102 void alarmCalistir() {
103     for (int i = 0; i < 2; i++) {
104         digitalWrite(buzzerPin, HIGH);
105         delay(100);
106         digitalWrite(buzzerPin, LOW);
107         delay(100);
108     }
109
110     digitalWrite(buzzerPin, HIGH);
111     delay(500);
112     digitalWrite(buzzerPin, LOW);
113 }
114

```

5.Sonuçlar ve Tartışma

Proje, motosiklet güvenliği konusunda etkili bir çözüm sunmuştur. IoT tabanlı entegrasyon, sistemi daha fonksiyonel hale getirmiştir. Görülen başlıca zorluklar, kablolu bağlantı sorunları ve donanım entegrasyonunda ortaya çıkmıştır.

Gelişim aşaması :



Projeyi geliştirirken SW-18020P, SW-520D, KY-002, SW-420 ve MPU6050 sensörleri sırasıyla denendi. Bu kadar farklı sensörün test edilmesinin sebebi, projenin gerektirdiği hassasiyeti ve doğruluğu yakalamaktır. Yapılan denemelerde yaylı sensörlerin farklı pozisyonlarda stabil çalışmadığı ve takılarak yanlış veriler ürettiği gözlemlendi. Son olarak, MPU6050'nin en stabil çalışan ve doğru değerler veren sensör olduğuna karar verilerek projede kullanılmasına karar verilmiştir.

6. Gelecek Çalışmalar ve İyileştirmeler

Bu proje, motosiklet güvenliği için etkili bir çözüm sunarken, gelecekte daha da geliştirilerek kullanıcı deneyimini ve sistemin dayanıklılığını artırmayı hedeflemektedir. Bu doğrultuda, aşağıdaki çalışmalar planlanmaktadır:

Entegre Batarya Sistemi:

- Projenin mevcut aşamasında sistem, harici bir güç kaynağına bağlı olarak çalışmaktadır. Gelecek aşamalarda, sistemin bağımsız ve taşınabilir hale gelmesi için entegre bir batarya sistemi geliştirilecektir. Bu sistem, motosikletin kendi bataryasından güç alma zorunluluğunu ortadan kaldıracak ve sistemi tamamen bağımsız hale getirecektir. Şarj edilebilir ve uzun ömürlü batarya seçenekleri değerlendirilerek sistemin pratik kullanımı sağlanacaktır.

3D Yazıcı ile Koruyucu Kalıp Entegrasyonu:

- Geliştirilen tüm elektronik bileşenleri (sensörler, mikrodenetleyici, batarya vb.) dış etkenlerden (darbe, su, toz vb.) korumak amacıyla 3D yazıcı teknolojisi kullanılarak özel bir kalıp tasarlanacaktır. Bu kalıp, disk kilidinin mekanik yapısına tam olarak entegre edilecek ve hem sistemi koruyacak hem de estetik bir görünüm sağlayacaktır. Kalıp, dayanıklı ve hafif malzemelerden üretilerek uzun ömürlü ve güvenilir bir koruma sağlanacaktır.

Mobil Uygulama Entegrasyonu:

- Mobil uygulama entegrasyonu ile kullanıcılar, alarm sistemini uzaktan yönetebilecek ve anlık bildirimler alabileceklerdir. Uygulama üzerinden alarmı aktif/pasif etme, titreşim hassasiyetini ayarlama, veri geçmişini görüntüleme gibi özellikler sunulacaktır.

Daha Hassas Sensörler:

- Daha hassas sensörler kullanılarak yanlış alarm olasılığı azaltılacak ve daha doğru hareket algılama sağlanacaktır. Bu sayede, gereksiz alarmlar önlenerek sistemin güvenilirliği artırılacaktır.

Yüksek Sesli Alarm Modülü:

- Daha yüksek ve dikkat çekici bir ses düzeyi kullanılacaktır. Bu sayede, hırsızlık girişimleri anında fark edilecek ve caydırıcılık artırılacaktır. Alarm modülü, yüksek ses çıkışı sağlayacak şekilde optimize edilecektir.

7.Kaynaklar

- Arduino IDE Resmi Belgeleri.
- MPU-6050 Datasheet.
- IoT Uygulamalarında ESP8266 Kullanımı.
- <https://maker.robotistan.com/deprem-alarmi-yapimi/>