



**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

Kurnaz Alarm

Ömer ÇEVİK

**Danışman
Dr. Öğr. Üyesi Alp Arslan BAYRAKÇI**

**Ocak, 2020
Gebze, KOCAELİ**



**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

Kurnaz Alarm

Ömer ÇEVİK

**Danışman
Dr. Öğr. Üyesi Alp Arslan BAYRAKÇI**

**Ocak, 2020
Gebze, KOCAELİ**

Bu çalışma 15/01/2020 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümünde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Dr. Öğr. Üyesi Alp Arslan BAYRAKÇI	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	
Jüri Adı	Prof. Dr. Erkan ZERGEROĞLU	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

ÖNSÖZ

Bu raporun ilk taslaklarının hazırlanmasında emeği geçenlere, raporun son halini almasında yol gösterici olan Sayın Dr. Öğr. Üyesi Alp Arslan BAYRAKÇI hocama ve bu çalışmayı destekleyen Gebze Teknik Üniversitesi'ne teşekkürlerimi sunarım.

Ayrıca bu zamana kadar geçirdiğim süre içerisinde bana sürekli destek veren aileme, dostlarıma ve beni bilgileri ile aydınlatıp akademik anlamda beni dolduran tüm hocalarıma saygı ve sevgilerimi sunarım.

Ocak, 2020

Ömer ÇEVİK

İÇİNDEKİLER

ÖNSÖZ	v
İÇİNDEKİLER.....	vi
ŞEKİL LİSTESİ	vii
KISALTMA LİSTESİ	viii
SEMBOL LİSTESİ.....	ix
ÖZET	x
SUMMARY	xi
1. GİRİŞ	1
2. ARKAPLAN	2
3. METOT	3
4. Buzzer Modülü	4
5. Uzaklık Sensörü Modülü	5
6. LCD Modülü	6
7. Keypad Modülü	7
8. Oyun	8
9. Motorlar	9
KAYNAKLAR.....	10

ŞEKİL LİSTESİ

ŞEKİL 1: Tasarım Planı	3
ŞEKİL 2: Buzzer Modülü	4
ŞEKİL 3: Uzaklık Sensörü Modülü	5
ŞEKİL 4: LCD Modülü	6
ŞEKİL 5: Keypad Modülü	7
ŞEKİL 6: Oyun	8
ŞEKİL 7: Motorlar	9

KISALTMA LİSTESİ

IDE : Integrated Development Environment

LCD : Liquid Crystal Display

ST : Società Generale Semiconduttori & Thomson Semiconducteurs

DC : Direct Current

SEMBOL LİSTESİ

ÖZET

Günümüzde insanlar sabahları uyanmakta zorluk çekmektedirler ve bu sorunu alarmlarla, alarmlı saatlerle çözmeye girişmişlerdir. Ancak normalde kurulumu yapıldıktan sonra bir tuşla durdurulabilen veya ertelenebilen bu alarmlı saatler ihtiyacı karşılamamakla birlikte istenildiği sonuçlar oluşturmamaktadır. Bu nedenle donanımsal olarak daha kurnaz işler çeviren bir alarmlı saate ihtiyaç duyulmaktadır. Bu alarm çalmaya başladığı zaman kaçmaya da başlamakta ve belirli süre aralıklarla melodisi değişmekte ve durdurulabilmesi için üzerinde olan oyunu oynayabilmesi ve bitirebilmesi gerekmektedir. Bu şekilde uyuyan insan uyanmaktan başka çaresi kalmayacaktır.

SUMMARY

Today people have difficulty waking up in the morning and have attempted to solve this problem with alarms and alarm clocks. However, these alarm clocks, which can normally be stopped or postponed with a key after installation, do not meet the need, but do not produce the desired results. For this reason, there is a need for an alarm clock which makes hardware more cunning. When this alarm starts to sound, it starts to run away and its melody changes periodically and it must be able to play and end the game to stop it. In this way, sleeping people will have no choice to wake up.

1. GİRİŞ

Günümüzde insanlar sabahları uyanmakta zorluk çekmektedirler ve bu sorunu alarmlarla, alarmlı saatlerle çözmeye girişmişlerdir. Ancak normalde kurulumu yapıldıktan sonra bir tuşla durdurulabilen veya ertelenebilen bu alarmlı saatler ihtiyacı karşılamamakla birlikte istenildiği sonuçlar oluşturmamaktadır. Bu nedenle alarmların evrimleşmesi yönünde bir girişim ortaya çıkması gerekmektedir. Kurnaz Alarm projesi ile alarmın susturulması kolay olmamakla beraber sabah uyandırmak için bire bir çözüm sunmaktadır. Alarmın çalmaya başladıktan sonra susturulacak veya erteleyebilecek bir tuşunun olmaması, alarm çalmaya başladıktan sonra hareket edip kaçmaya başlaması ve alarmın peşinden koşup yakalanılırsa bile üzerinde başlanılmış olan oyunun tamamlanması koşuluyla ancak susturulabilmesi tamamen amaçlanan uyandırma işlemini doğrular niteliktedir.

2. ARKAPLAN

Projede alarının alması ve hareket edebilmesi iin mikroişlemci aracılığına ihtiyaç duyulmaktadır. Araştırmalar sonucunda STM32 Nucleo F401RE mikroişlemci kitine karar verildi ve STM32 mikroişlemci kitinde yer alan pinlerle istenilen modüller bağlanıldı. Gerçek zamanlı saat olarak STM32 içerisinde yer alan RTC modülü kullanıldı.

Bu modüllerden başlıca olarak buzzer modülü STM32 mikroişlemci kitine entegrasyonu sağlanılarak ses ıkartılması yönünde ilk işlemler tamamlandı.

Bir başka modül olarak ileride hareket halinde olacak olan alarının yön değıştirebilir halde olabilmesi iin gerekli olan bir adet uzaklık sensör modülü de gerekli olan araştırmalar ve implementasyonlar sonucunda STM32 mikroişlemci kitine bağlanıldı.

Saatın gösterilmesi ve alarının durdurulabilmesi amacıyla kullanıcıya bir arayüz oluşturulması yönünde LCD 16 x 2 ekran modülü i2c protokolüyle STM32 mikroişlemci kitine bağlanıldı.

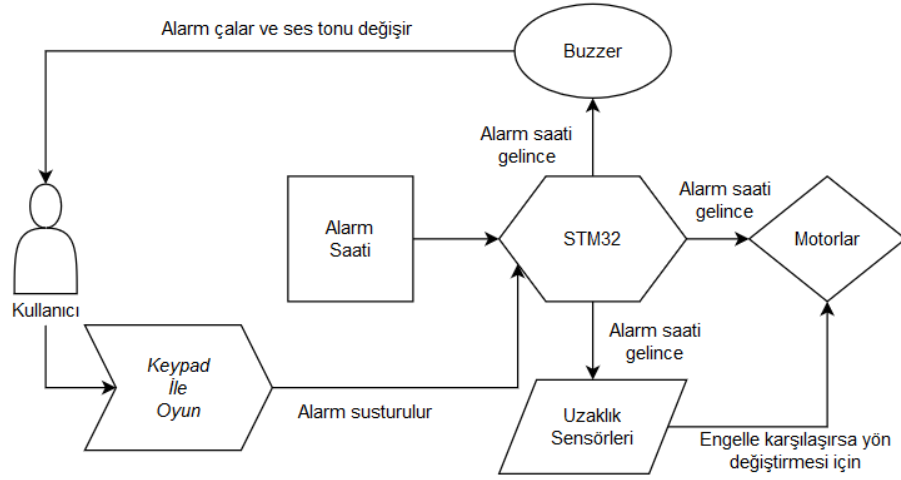
Alarm almaya başladığında LCD ekranda beliren basit bir toplama işlemi oyununun oynanabilmesi iin keypad modülü STM32 mikroişlemci kitine entegre edildi.

Alarmın aldığı anda kaçabilmesi iin gerekli olan iki adet DC motor, DC motor sürücü kartı ve tank paletleri kullanıldı. Mikroişlemci ile motorlar DC motor sürücü kartı ile haberleşmektedir.

Yapılan bütün implementasyonlar Windows 10 işletim sistemi üzerinde, Keil µVision 5 ve STM32CubeMX IDEleri kullanılarak C programlama diliyle gerçekleştirilmiştir.

3. METOT

Bütün modüllerin aynı anda STM32 üzerinde çalışmasından önce modül modül tasarlanan implementasyonlar teker teker STM32 üzerinde çalıştırılması yönünde aşama kaydedildi.



ŞEKİL 1: Tasarım Planı

4 Buzzer Modülü

Buzzer modülünü kullanmak için STMCubeMX IDE'si ve Keil μ Vision 5 IDE'si aracılığıyla C kodunda timerlar kullanılarak belirli frekanslarla tonun başlatılıp durdurularak melodi elde edilmesi sağlandı. Bu tonun hızlılığı ve yavaşlığına göre farklı melodiler oluşturulması elde edildi.

```
int melody = 100;
void buzzerModule(void)
{
    int index = 0;
    float freq = 0;

    while(freq != -1)
    {
        freq = ToneFreq[index++];
        startTone(freq);
        HAL_Delay(melody);
        stopTone();
        HAL_Delay(melody);
    }
    index = 0;
    freq = 0;
}
```

Şekil 2: Buzzer Modülü

5 Uzaklık Sensörü Modülü

Uzaklık sensörü, belirlenmiş trigger pinini set ve reset ederken echo pininden okunan değerlerle ses hızını çarpıp mesafe ölçmek için kullanılır. Bu sensör hareket halindeki alarmin motorlarının durup yön değiştirebilmesi için gereklidir ve sayısı bir adettir.

```

void ultraSonicModule()
{
    uint32_t numTicks;
    HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_RESET);
    usDelay(3);

    HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_SET);
    usDelay(10);
    HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_RESET);

    while (HAL_GPIO_ReadPin(ECHO_GPIO_Port, ECHO_Pin) == GPIO_PIN_RESET);

    numTicks = 0;
    while (HAL_GPIO_ReadPin(ECHO_GPIO_Port, ECHO_Pin) == GPIO_PIN_SET)
    {
        ++numTicks;
        usDelay(2); //2.8usec
    };

    distance = (numTicks + 0.0f)*2.8f*speedOfSound;

    if (distance < 10)
    {
        // Stop Motors. Turn the motor. Start motors.
        sprintf(uartBuf, "Distance is smaller than 10 cm !\r\n");
    } else
    {
        sprintf(uartBuf, "Distance (cm) = %.1f\r\n", distance);
    }
    HAL_UART_Transmit(&huart2, (uint8_t *)uartBuf, strlen(uartBuf), 100);
    HAL_Delay(150);
}

```

Şekil 3: Uzaklık Sensörü Modülü

6 LCD Modülü

LCD ekranı kullanabilmek için i2c protokolüyle ekrana belirlenen karakterler yazdırılabilmektedir. 16 x 2 LCD ekran için yapılan implementasyonlarda ilk satıra yazmak için 0x80, ikinci satıra yazmak için 0xc0 komutları ile ayarlamalar yapılabilmektedir.


```

void lcd_send_cmd (char cmd)
{
    char data_u, data_l;
    uint8_t data_t[4];
    data_u = (cmd&0xf0);
    data_l = ((cmd<<4)&0xf0);
    data_t[0] = data_u|0x0C; //en=1, rs=0
    data_t[1] = data_u|0x08; //en=0, rs=0
    data_t[2] = data_l|0x0C; //en=1, rs=0
    data_t[3] = data_l|0x08; //en=0, rs=0
    HAL_I2C_Master_Transmit (&hi2c1, SLAVE_ADDRESS_LCD, (uint8_t *) data_t, 4, 100);
}

void lcd_send_data (char data)
{
    char data_u, data_l;
    uint8_t data_t[4];
    data_u = (data&0xf0);
    data_l = ((data<<4)&0xf0);
    data_t[0] = data_u|0x0D; //en=1, rs=0
    data_t[1] = data_u|0x09; //en=0, rs=0
    data_t[2] = data_l|0x0D; //en=1, rs=0
    data_t[3] = data_l|0x09; //en=0, rs=0
    HAL_I2C_Master_Transmit (&hi2c1, SLAVE_ADDRESS_LCD, (uint8_t *) data_t, 4, 100);
}

void lcd_init (void)
{
    lcd_send_cmd (0x02);
    lcd_send_cmd (0x28);
    lcd_send_cmd (0x0c);
    lcd_send_cmd (0x80);
}

void lcd_send_string (char *str)
{
    while (*str) lcd_send_data (*str++);
}

```

Şekil 4: LCD Modülü

7 Keypad Modülü

Keypad modülü ile alarm çaldığı zaman LCD ekranda beliren oyunun oynanması sağlanmaktadır. 4 x 4 keypad için yapılan implementasyonda her bir sütun için girdi olarak kullanıcı belirlemekte ve her bir sütunun hangi satırda olduğunu kod içerisinde her satıra set ve reset ederek bulunabilmektedir.

```

HAL_GPIO_WritePin (KEYPAD_R1_GPIO_Port, KEYPAD_R1_Pin, GPIO_PIN_RESET); //Pull the R1 low
HAL_GPIO_WritePin (KEYPAD_R2_GPIO_Port, KEYPAD_R2_Pin, GPIO_PIN_SET); // Pull the R2 High
HAL_GPIO_WritePin (KEYPAD_R3_GPIO_Port, KEYPAD_R3_Pin, GPIO_PIN_SET); // Pull the R3 High
HAL_GPIO_WritePin (KEYPAD_R4_GPIO_Port, KEYPAD_R4_Pin, GPIO_PIN_SET); // Pull the R4 High

for (int i = 0; i < 65535; ++i);
if (!(HAL_GPIO_ReadPin (KEYPAD_C1_GPIO_Port, KEYPAD_C1_Pin))) // if the Col 1 is low
{
    gameUser[usrIndex] = '2';
    if(gameUser[usrIndex]-48 + game[usrIndex-6] == 9)
    {
        ++winner;
    }
    ++usrIndex;
    ++counter;
}

```

Şekil 5: Keypad Modülü

8 Oyun

Alarmin durdurulabilmesi için bilgisayar tarafından belirlenmiş dokuz rakamla kullanıcının gireceği dokuz rakamın birbirlerine karşılık gelen sayısal değerlerin toplamının dokuz olması gerekmektedir. Oyun amaç olarak basitçe bu işlemi gütmektedir.

```

void playGame(void)
{
    char gamePC[30];
    lcd_send_cmd(0x80);
    sprintf(gamePC,"PC : %d%d%d%d%d%d%d", game[0], game[1], game[2], game[3], game[4], game[5], game[6], game[7], game[8]);
    lcd_send_string(gamePC);
    lcd_send_cmd(0xc0);
    lcd_send_string(gameUser);

    while(gameContinue)
    {
        while(counter < 9)
        {
            ultraSonicModule();
            buzzerModule();
            lcd_send_cmd(0xc0);
            lcd_send_string(gameUser);
        }
        if(winner == 9)
        {
            alarmRing = false;
            gameContinue = false;
            HAL_GPIO_DeInit(Button_GPIO_Port,Button_Pin);
        }
        else{
            for (int i = 6; i < 16; ++i)
                gameUser[i] = ' ';
            counter = 0;
            winner = 0;
            usrIndex = 6;
        }
    }
}

```

Şekil 6: Oyun

9 Motorlar

Alarm çalmaya başladığı anda ekranda beliren oyunla birlikte kaçmaya çalışması gerekmektedir. Bu kaçış DC motorlara 12 volt gerilim verilerek motorların paletleri ileri doğru sürmesiyle gerçekleşmektedir. Eğer alarm karşısında bir engelle karşılaşırsa uzaklı sensörüyle haberleşen motorlar durur ve başka bir yöne dönerek hareketine devam etmektedir.

```
#include "Motors.h"

void startForwardMotors(void)
{
    HAL_GPIO_WritePin(MOTOR_1_1_GPIO_Port,MOTOR_1_1_Pin,GPIO_PIN_SET);
    HAL_GPIO_WritePin(MOTOR_1_2_GPIO_Port,MOTOR_1_2_Pin,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(MOTOR_2_1_GPIO_Port,MOTOR_2_1_Pin,GPIO_PIN_SET);
    HAL_GPIO_WritePin(MOTOR_2_2_GPIO_Port,MOTOR_2_2_Pin,GPIO_PIN_RESET);
}

void stopMotors(void)
{
    HAL_GPIO_WritePin(MOTOR_1_1_GPIO_Port,MOTOR_1_1_Pin,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(MOTOR_1_2_GPIO_Port,MOTOR_1_2_Pin,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(MOTOR_2_1_GPIO_Port,MOTOR_2_1_Pin,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(MOTOR_2_2_GPIO_Port,MOTOR_2_2_Pin,GPIO_PIN_RESET);
}

void turnMotors(void)
{
    stopMotors();
    HAL_Delay(600);

    HAL_GPIO_WritePin(MOTOR_1_1_GPIO_Port,MOTOR_1_1_Pin,GPIO_PIN_SET);
    HAL_GPIO_WritePin(MOTOR_1_2_GPIO_Port,MOTOR_1_2_Pin,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(MOTOR_2_1_GPIO_Port,MOTOR_2_1_Pin,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(MOTOR_2_2_GPIO_Port,MOTOR_2_2_Pin,GPIO_PIN_SET);

    HAL_Delay(600);
    startForwardMotors();
}
```

Şekil 7: Motorlar

KAYNAKLAR

- [1] <https://en.wikipedia.org/wiki/STMicroelectronics>
- [2] <https://github.com/MYaqoobEmbedded/STM32-Tutorials>
- [3] <https://www.draw.io>
- [4] <https://www.controllerstech.com/use-4x4-keypad-with-stm32>