yazı tipi, metin, beyaz, tipografi içeren bir resim

Açıklama otomatik olarak oluşturuldu**MÜHENDİSLİK FAKÜLTESİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

**EEM 419 MİKROİŞLEMCİLER**

**Dersin Sorumlusu**

Doç. Dr. Rıfat HACIOĞLU

**Proje Adı**

GPS ve Mikrodenetleyici Tabanlı Sistem ile Konumlar Arası Otonom Hareket Eden İKA Tasarımı

**Grup No:** I-20

**Projeyi Yapanlar**

190106106037 Rahime Yaşar

190106106039 Busenur Toprak

207106106014 Ahmed Nasser

|  |  |
| --- | --- |
| **Uygulama Lideri:** | Rahime Yaşar |

|  |  |  |
| --- | --- | --- |
|  | **Talimat Veren** | **Talimat Alan** |
| **Malzeme Temini** | Rahime Yaşar | Ahmed Nasser  Busenur Toprak |
| **Düzenek Hazırlanma** | Ahmed Nasser | Busenur Toprak  Rahime Yaşar |
| **Donanım** | Busenur Toprak | Rahime Yaşar  Ahmed Nasser |
| **Yazılım** | Rahime Yaşar | Busenur Toprak  Ahmed Nasser |
| **Sonuç Kaydetme** | Ahmet Nasser | Rahime Yaşar  Busenur Toprak |
| **Raporlama** | Busenur Toprak | Ahmed Nasser  Rahime Yaşar |

**Teslim Tarihi**

11.01.2024

**I. Projenin Konusu**

Son yıllarda insansız araçlar araştırmacılar için oldukça popüler bir konu haline gelmiştir ve hala bu araçlar araştırmacılar ve mühendisler tarafından geliştirilmektedir. Tasarlanacak insansız araç projelerinde yapması gereken en önemli özelliklerden bir tanesi verilecek görevi yardım almadan gerçekleştirebilecek bütün kabiliyetlere sahip olmasıdır. Tasarlanan projede de dışardan yardım almadan otonom bir şekilde çalışan insansız kara aracı (İKA) yapılarak bu aracın bulunduğu herhangi bir konumdan, belirtilen varış noktalarına gitmesi üzerinde çalışılmıştır. Proje yazılım ve donanım olarak iki aşamadan oluşmaktadır. Tasarlanan sistemin donanım ve yazılım bölümlerinin entegrasyonu sağlanarak İKA'nın çeşitli görevleri yerine getirmesi sağlanmıştır. Bu görevlerden en önemlisi İKA'nın harita verileri üzerinden belirlenen hedeflere otonom bir şekilde ulaşabilmesidir. Bu görevi, GPS ve elektronik pusula sensörlerden alınan verilerin entegrasyonu ve bu verilerin kullanılmasıyla gerçekleşmiştir. Böylece oluşturulan otonom navigasyon sistemi, İKA'nın çevresini anlayıp daha sonra varış noktasına güvenli bir şekilde yönlendirilmesi sağlamıştır. Aynı zamanda İKA'nın denetim ve karar alma süreçleri, bir mikrodenetleyici tabanlı sistem tarafından yönetilecek şekilde bir tasarım oluşturulmuştur. Mikrodenetleyici, GPS ve pusula sensöründen gelen verileri işleyerek, kararlar alması sağlamış ve motorları yönlendirerek aracın hareketini kontrol etmiştir. Mikrodenetleyiciye yüklenen yazılım aracılığıyla gerçekleştirilecek ve belirlenen hedeflere güvenli bir şekilde ulaşmak için gerekli navigasyon stratejilerini içermesi sağlanmıştır. Tasarlanan projede, düşük maliyetli bir GPS modülü ve elektronik pusula sensörü gibi uygun fiyatlı donanım bileşenlerinin kullanılarak, proje maliyetinin düşürülerek benzer uygulamaların daha erişilebilir olması sağlanmıştır.

**II. Projenin Amacı**

Tasarlanan sistemin amaç ve hedefleri arasında;

1. Otonom bir şekilde hareket eden bir İKA tasarımının gerçekleştirilmesi,
2. Tasarlanan İKA'nın, GPS ve pusula sensörleri eklenerek belirlenen varış noktalarına güvenli bir şekilde ve herhangi bir manuel müdahale olmadan ulaşabilmesi,
3. Aynı zamanda sisteme kamera sensörü entegre edilerek istenilen konumlar arasında otonom bir şekilde çalışan keşif aracı olarak kullanılması,
4. Tasarlanan keşif aracının istenen konumlardan kamera görüntüleri alması sağlamak,
5. Alınan görüntülerin bilgisayar ortamında toplanarak ilerde yapılacak çalışmalar için bir veri seti oluşturulması,
6. Düşük maliyetli ve etkili bir otonom navigasyon sistemini başarıyla gerçekleştirmesi,
7. Takım çalışma kültürünün benimsenmesi ve etkili bir grup çalışması gerçekleştirilmesi,

yer almaktadır. Bu amaçlar ve hedefler doğrultusunda sistem tasarlanmış ve ilerde yapılacak çalışmalar için bir zemin oluşturulmuştur.

**III. Mikroişlemci Tabanlı Sistem Tasarımı, Donanım**

“GPS ve Mikrodenetleyici Tabanlı Sistem ile Konumlar Arası Otonom Hareket Eden İKA Tasarımı” projesi iki aşamadan oluşmaktadır. Bu aşamaların biri donanım bölümüdür. Donanım bölümde kullanılan mikrodenetleyiciye Şekil 1’deyer verilmiştir. GPS modülü Şekil 2’de yer almaktadır. Pusula sensörü Şekil 3’te bulunmaktadır. Tasarımda kullanılan kamera modülüne Şekil 4’te yer verilmektedir. Konum bilgisinin göstermesi beklenen LCD ekrana Şekil 5’te yer verilmektedir. Engelleri algılan ultrasonik sensör Şekil 6’da gösterilmektedir. Şekil 7’de birimler arası bağlantı sağlayan jumper kablolara yer verilmiştir. Şekil 8’de ise araba kiti bulunmaktadır. Enerji kaynağına Şekil 9’da yer verilmiştir. Kullanılan malzemelere teker teker yer verilmiş ve bu malzemelerin görevlerinden bahsedilmiştir.

|  |  |
| --- | --- |
| Arduino UNO R3 Satın Al 540,00 TL | Robotzade.com  Şekil 1: Mikrodenetleyici | İçerisinde ATmega328 mikrodenetleyici bulunan Arduino Uno; konum bilgisinin doğruluğunu ve güvenilirliğini artırmak için kullanılan sensörlerden gelen verilerin entegrasyonunu yönetecektir. |
| GY-NEO6MV2 GPS Modülü | SAMM Market  Şekil 2: GPS modülü | GPS modülü uydunun ne kadar uzakta olduğu ve uydu sinyallerinin gelme süresini hesaplayarak aracın o anki konum bilgisinin belirlenmesi amacı ile kullanılmaktadır. |
| HMC5883L 3 Eksenli Pusula Sensörü - Manyetometre - GY-271 Uygun Fiyatıyla  Satın Al - Direnc.net®  Şekil 3: Pusula Sensörü | Manyetik alan ölçerek yönleri sisteme belirtmek için ve navigasyondan yararlanmak için pusula sensörü kullanılmaktadır. |
| Arduino Kamera Modülü OV7670 PIC Kamera Modülü Fiyatı  Şekil 4: Kamera modülü | İKA’nın gideceği konum veya gittiği konumdan görüntüler alınarak çevrenin incelenmesi amaçlı kamera modülü kullanılmaktadır. |
| Nokia 5110 Ekranı - 84x48 Grafik LCD Arduino  Şekil 5: LED ekran | LCD ekran ile aracın mevcut konum bilgilerinin kolaylıkla okunması amacı ile kullanılmaktadır. |
| Arduino HC-SR04 Ultrasonik mesafe sensörü - Prototip Elektronik  Şekil 6: Ultrasonik Sensör | Ultrasonik sensör ses dalgalarını kullanarak mesafeyi ölçen sensör çeşididir. Aracın konuma ilerleme durumunda önüne çıkabilecek nesneleri algılayarak engeli aşması ve yoluna devam etmesi için kullanılmaktadır. |
| Jumper Kablo Erkek Erkek 40Pin (20cm) : Amazon.com.tr: Bilgisayar  Şekil 7: Jumper kablolar | Giriş, çıkış ve bellek bölümlerinde yer alan ekipmanların birbiri ile iletişim kurabilmesi, verilerin aktarılması için jumper kablolar yol olarak kullanılmaktadır. |
| 2WD Robot Araba Kiti Uygun Fiyatıyla Satın Al - Direnc.net®  Şekil 8: 2WD Robot Araba Kiti | Robot araba kiti ile aracın motor ve sensör bağlantılarını derleyerek İKA sisteminin model görüntüsü elde edilmek amaçlanmıştır. |
| Varta 2022 Superlife 9V Pil Fiyatı - Pilburada.com  Şekil 9: Pil | Projede kullanılan ekipmanların güç ihtiyacını karşılamak için kullanılmıştır. Kullanılan modülleri 5V’luk güç kaynağı desteklenmediği için 12V tercih edilmiştir. |

Tasarlanan sistemden malzemeler belirlendikten sonra hangi malzemenin hangi birimde kullanılacağı belirlenmiştir. Bu birimlerin blok şemasını Şekil 10’de yer verilmiştir.

metin, elektronik mühendisliği, elektronik donanım, devre içeren bir resim

Açıklama otomatik olarak oluşturulduŞekil 10. Tasarlanan sistemin birim blok şeması

Tasarlanan projenin birimleri;

**Giriş Birimi:** Gerçekleştirilen proje kapsamında giriş birimleri olarak ultrasonik sensör, pusula sensörü, kamera modülü ve GPS modülü kullanılmıştır. İKA ultrasonik sensör kapsamında engelleri fark edip engeli aşarak yoluna devam etmesini sağlamaktadır. Giriş biriminde kullanılan ultrasonik sensör aynı zamanda kesme işlemini gerçekleştirilmek için kullanılmaktadır. Kesme, herhangi bir zamanda gerçekleşerek ana program akışını durdurarak kesme sırasında gerçekleşmesi gereken program çalışarak engel aşıldığı takdirde ana program kaldığı yerden devam edecek şekilde algoritmik düzen oluşturulmuştur. Pusula sensörü sayesinde aracın yönleri bulmada yardımcı olan bir sistem tasarlanması yapılmıştır ve pusula ile yönünü bulan aracın istenilen hedef koordinata doğru otonom olarak harekete geçmesi sağlanmıştır. GPS modülü de mevcut koordinat bilgisini elde edecek şekilde tasarıma eklenmiştir. Kamera modülü de istenilen konumdan görüntü alınabilmesi hedeflenmiş olup sisteme entegre edilmesi sağlanmıştır.

**Merkezi İşlem Birimi:** Mikrodenetleyici, konum bilgisinin doğruluğunu ve güvenilirliğini artırmak için kullanılan sensörlerden gelen verilerin entegrasyonunu yönetmesi sağlanmıştır. Aynı zamanda mikrodenetleyici, bu sinyalleri yorumlayan, hesaplamaları gerçekleştiren ve kullanıcının kesin koordinatlarını türeten algoritmaların yürütülmesinden sorumlu tutulmuştur. Girişlerden gelen verilerin adreslenmesi sağlayarak programının düzgün çalışması sağlamaktadır. Aynı zamanda mikrodenetleyici her bir girişten gelen verinin hangi çıkışa adresleyeceğini ve ALU birimi sayesinde hızlı bir şekilde gerçekleşmesi gereken matematiksel işlemleri gerçekleştirdiği görülmedir. Kontrol merkez biriminde giriş ve çıkış birimleri ve ALU birimdeki verilerin kontrolleri sağlanmış olup sistemin kararlı ve doğrusal bir şekilde çalışmasını sağlamıştır.

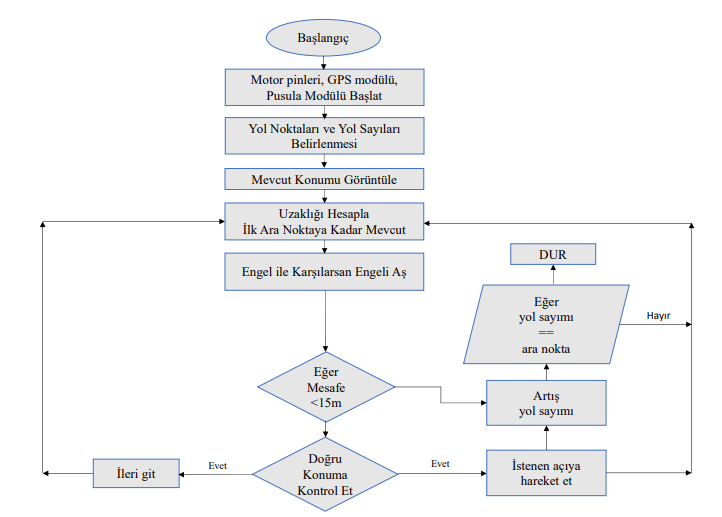
**Çıkış Birimi:** Tasarlanan projede çıkış birimi olarak yapılacak araç hareketini sağlayacak motorlar kullanılmıştır. Aynı zamanda kullanılacak LCD ekranda ise konum bilgisini gösterilmesi sağlanmıştır. Kamera sensörü ile görüntü alımı gerçekleştirilmiştir.

**Yollar:** Giriş birimi, merkezi işlem ve çıkış biriminde kullanılan elemanların birbiriyle iletim sağlaması için kullanılacak elemanlardır. Bu birimler arası iletimi gerçekleştirmek için jumper kablolar kullanılmıştır.

Bu birimlerin birbirine entegrasyonu sonucunda projenin donanım kısmı tamamlanmış ve yazılım bölümüne geçilmiştir.

**IV. Proje Yazılım**

Projenin yazılım bölümünde sırasıyla GPS modülü, Pusula sensörü, Kamera sensörü ,Nokia 5110 Ekran, motorlar ve ultrasonik sensör bağlantıları gerçekleştirildikten sonra tanımlı pinle belirlenmiş ve bu pinlere göre algoritmik düzen oluşturulmuştur. Algoritmik düzen oluşturulurken alt programlardan ve kesmelerden yardım alınmıştır. Programın çalışma mantığı sırasıyla motor ,GPS ve pusula sensörlerin programa tanımlanarak çalıştırılması daha sonra bu alınan veriler ile aracın konumunu belirlenmesi ve istenilen konuma olan uzaklığının belirlenmesi, aracın istenilen konuma giderken engel ile karşılaşırsa bu engeli aşması ve istenilen konuma hareket etmesi şeklinde ilerlemektedir. Araç istenilen konuma geldiğinde durması ve hareketi kesmesi isterler arasındadır. Projenin akış diyagramına Şekil 11’de yer verilmiştir.



Şekil 11: Yazılım bölümü akış diyagramı.

Proje yazılımı Arduino IDE ortamında derlenerek USB aracılığı ile yazılan kodlar mikrodenetleyiciye aktarılmıştır. Proje yazılımında kesme ve alt programlar kullanılmıştır. İlk olarak pusula için Adafruit\_HMC5883\_U kütüphanesi, Adafruit\_Sensor ve GPS için TinyGPSPlusPlus kütüphanesi tanımlanmıştır. LCD ekran içinde Adafruit\_GFX ve Adafruit\_PCD8544 kütüphaneleri tanımlanmıştır. Pusula sensöründen gelen veriler X, Y, Z, koordinat düzleminde gelmektedir. GPS modülünde gelen veriler ise enlem ve boylam olarak iki ayrı veri olarak gelmektedir. Algoritma çalıştırıldıktan sonra hem GPS hemde pusula sensör verileri serial monitörde de gözükmüş ve Arduino Uno enerji bağlandığı zamanda gerçek zamanlı olarak çalıştığı gözlemlenmiştir. Kullanılan kod satırlarına Şekil 12’de yer verilmiştir. Şekil 13’de serial monitör görüntüsüne yer verilmektedir.

|  |  |
| --- | --- |
|  |  |
|  |  |

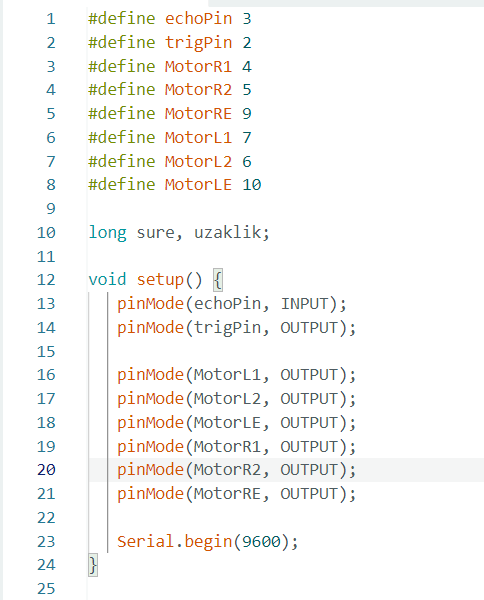
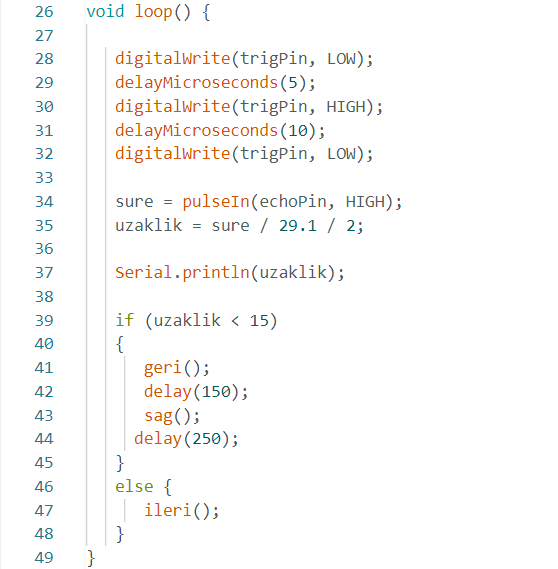
Şekil 12: GPS ve pusula verilerini LCD ekrana aktaran kod satırları.

metin, yazı tipi, ekran görüntüsü, makbuz içeren bir resim

Açıklama otomatik olarak oluşturulduŞekil 13: GPS ve pusula verilerinin serial monitör çıktısında teset edilmesi.

Şekil 12’de yer alan GPS modülü için kullanılan kod satırlarında Zonguldak için manyetik deklinasyon eğrisi tanımlanarak GPS ve pusula verisi LCD ekrana gönderilmesi sağlanmıştır.

Ultrasonik sensör için gerekli olan tanımlamalar yapıldıktan sonra engel ile karşılaştığında motorların hareket komutları şekil 14’de yer alan kod satırlarında tanımlanmıştır.

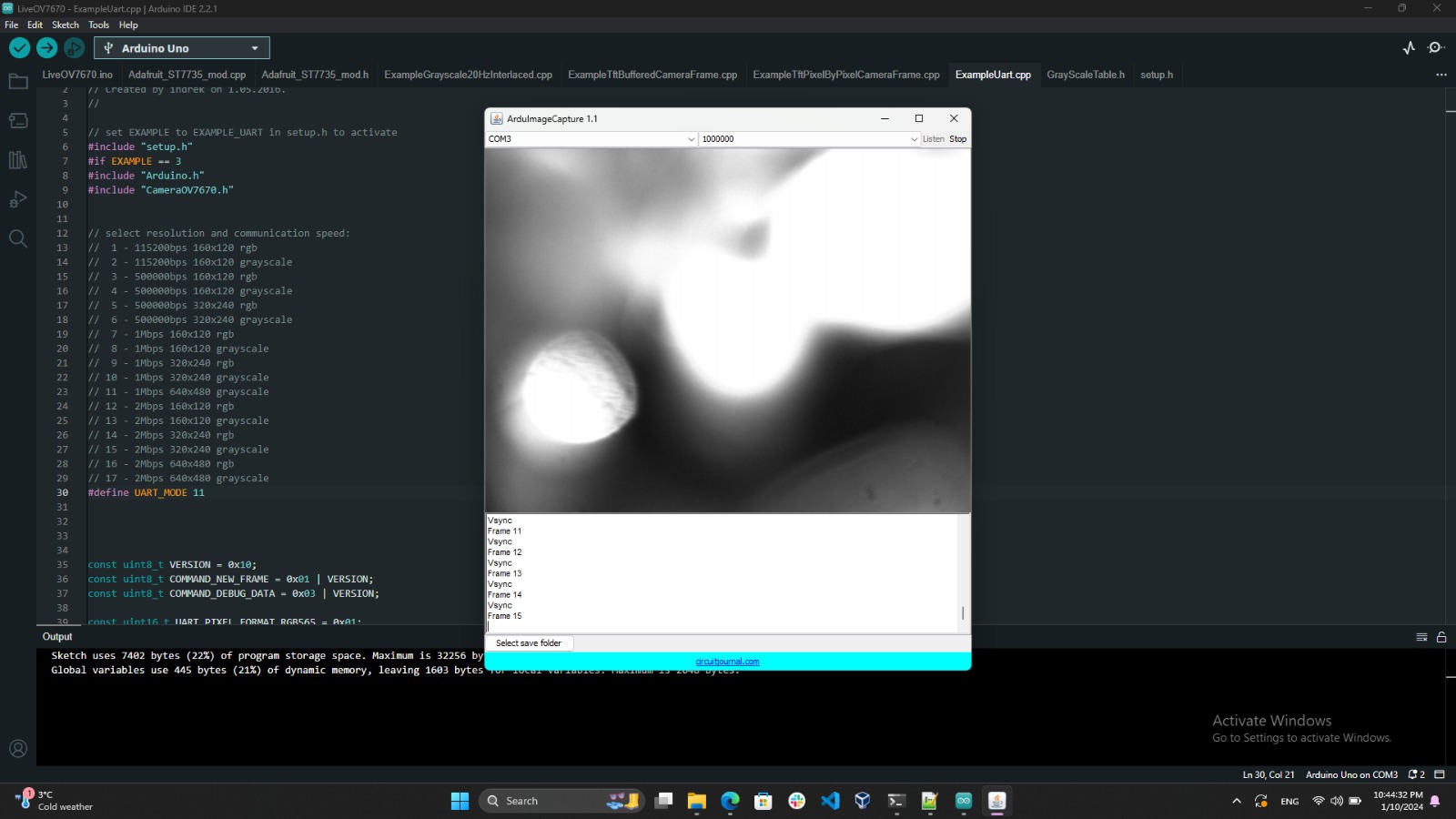
  metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 14: Aracın engel ile karşılaşma durumunda motorların hareket komutları.

Şekil 14’de görülen kod satırlarında öncelikle ultrasonik sensörün ses dalgasına göre mesafeyi ölçmektedir. Burada ultrasonik sensör herhangi bir zamanda engel ile karşılaştığında kesme işleminin gerçekleşmesi için kullanılmaktadır. İstenilen konuma giden araç herhangi bir engelle karşılaşmadığı takdirde ana programda konuma ilerlerken karşına engel çıktığında ultrasonik sensör ve engel ile arasındaki mesafe 15 cm altına düştüğünde ana program durmakta ve kesme alt programı ile birinci motor ve ikinci motorların yönleri kısa süreliğine değiştirilerek aracın engeli aşması sağlanmaktadır. Engel aşıldığı takdirde kesme alt programından çıkıp ana programa tekrar dönmektedir.

Tasarlanan sisteme son olarak kamera modülü entegrasyonu sağlanmış ve görüntü alabilmek için gerekli algoritmik düzen oluşturmuştur. Kamera modülü algoritması eklenirken LiveOV7670, Setup, GrayScaleTable, OV7670REGISERDEFINITIONS ve BufferedCameraOV7670 kütüphaneleri eklenerek ana program altında çalışan alt programlar olarak kütüphaneler eklenmiş ve yazılım düzeni tamamlanmıştır. Şekil 15’de kamera kodlarının çalışma çıktısı yer almaktadır.



Şekil 15:Kamera modülü yazılım çıktısı

**V. Proje Uygulama Sonuçları**

İlk olarak dc motorlar, motor sürücü, ultrasonik sensör ve araba platformu birleştirilerek Şekil 16’te görülen piller yardımı ile beslemesi yapılmıştır.

|  |  |
| --- | --- |
| Elektrik kabloları, Araba parçası, elektronik mühendisliği, araba lastiği içeren bir resim  Açıklama otomatik olarak oluşturuldu |  |

Şekil 16: Motor bağlantıları ve araç platformunun üstten ve önden görünümü.

Şekil 16’te yer alan bağlantılara mikrodenetleyici entegre edilerek motorların ve ultrasonik sensörün istenilen düzeyde çalışması için testler gerçekleştirilmiştir. Şekil 17’da pusula, GPS, LCD ekran bağlantıları yer almaktadır.

elektronik donanım, kablo, Elektrik kabloları, elektronik mühendisliği içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 17: Pusula ve GPS verilerinin ekrana aktarılması test çalışması.

Şekil 17’da mikrodenetleyiciye uygun kodlar yüklenerek GPS ve pusula verileri LED ekrana yansıtılmıştır.

Şekil 18’de mikrodenetleyiciye gerekli kodlar yüklenerek kamera modülünün test edilmesi sağlanmıştır.

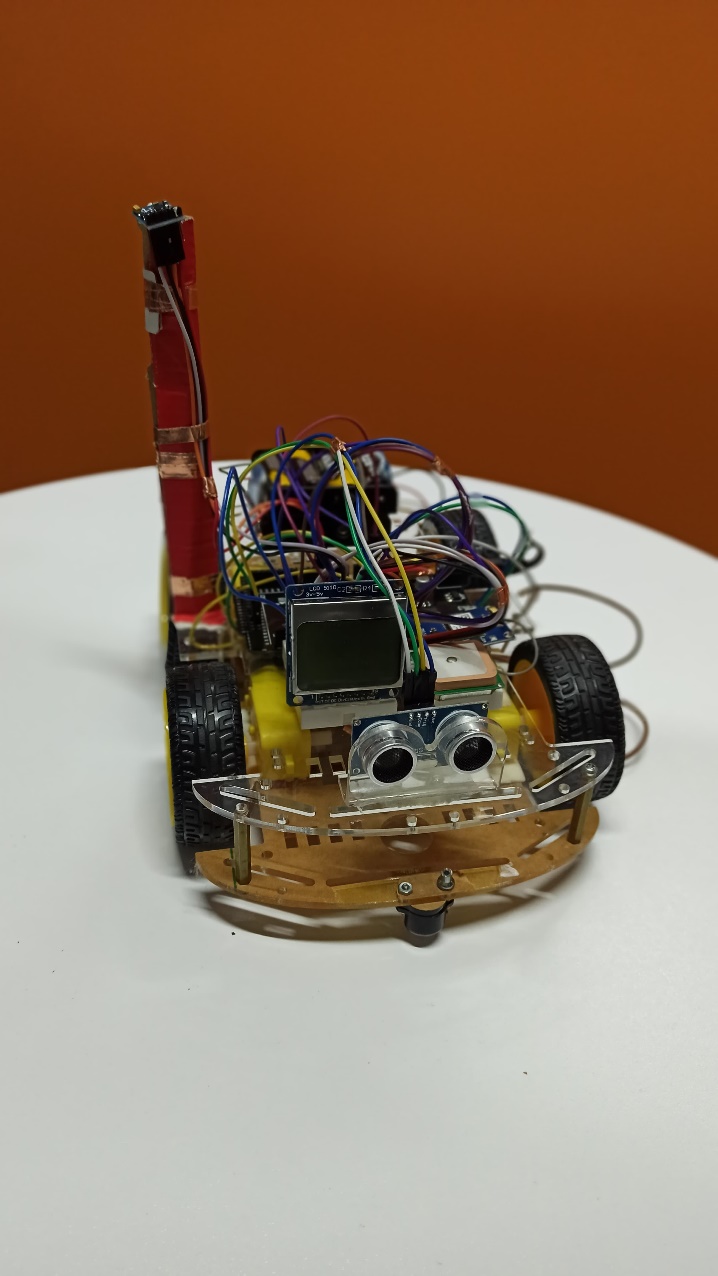
Elektrik kabloları, kablo, elektronik mühendisliği, elektronik donanım içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, bilgisayar, elektronik donanım, elektronik cihaz içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 18:Kamera modülünün düzeneğinin kurulması ve test edilmesi

Şekil 16, Şekil 17 ve Şekil 18’da gerçekleşen test işlemlerinden sonra kontrolü sağlanan bileşenler Şekil 19’de görüldüğü gibi bir bütün olarak araba platformuna entegre edilmiştir.

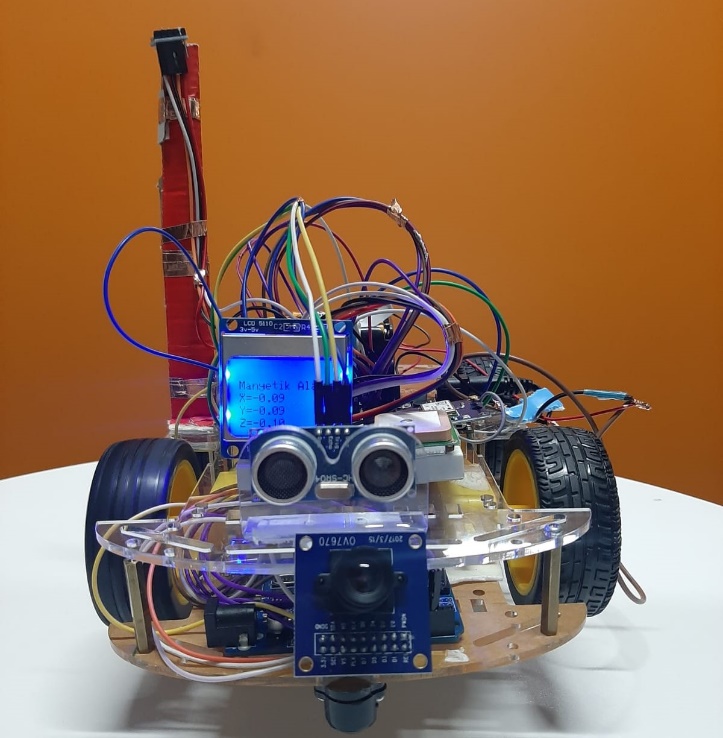
 otomat, oyuncak, iç mekan içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 19: Modüllerin araca entegre edilme aşaması.

Şekil 19’de yer alan pusula, GPS, ultrasonik sensör ve LED ekranın mikrodenetleyici ile bağlantısı sağlanarak test aşaması gerçekleştirilmiştir.

Şekil 20’de projenin tüm modüllerinin entegre edilmiş haldeki görüntüsü yer almaktadır.



Şekil 20: Aracın son halinin önden görünümü.

**VI. Yorumlar**

Projemizde kullanılan donanım malzemeleri ve bu malzemelerinin algoritma düzenleri öğrenilmiş oldu. Tasarlanan sistem yazılım ve donanım olarak iki ayrı bölümde incelenip bu bölümlerinin birbiri ile entegrasyonu sağlandı. Tasarlanan sistemde mikrodenetleyicinin kullanılmasının önemi pekiştirilerek çalışma mantığı anlaşıldı. Tasarlanan sistemde ATmega328 mikrodenetleyici kullanılarak istenilen sonuca ulaşılamadığından ve bu mikrodenetleyicinin sistem için yetersiz olduğundan dolayı ATmega2560 mikrodenetleyicisi kullanıldı. Sistemin ana amacı olan otonom bir şekilde çalışan İKA tasarlanmış ve çalıştırılmış olup keşif için kullanılacak kamera sensöründe sadece bilgisayar ile bağlı iken görüntü alınarak USB bağlantı kesildikten sonra görüntü alınamamıştır. Sistem tasarımında takım çalışması kültürü benimsenmiş olup planlanan iş paketleri paylaşılarak proje birlikte tamamlanmıştır. Tasarlanan sistemin zorlu arazilerde kullanılacak olup savunma, ileri konumlandırma ve haritalandırma gibi birçok alanda da faydalı olacağı öngörülmektedir. Ayrıca sistemde geliştirilen düşük maliyetli otonom navigasyon sisteminin bu gibi projelere entegre edilerek maliyetin düşürülmesi sağlanabilir. Tasarlanan proje göz önüne alındığında geliştirilme açık ve potansiyeli olan bir teknolojik gelişim olduğu görünmektedir. Örneğin sürekli ve düzenli görüntü alınabilmesi için sisteme bluetooth modülü eklenerek sistem daha geliştirilebilir. Daha sonra Çin’de gösterime sunulan görünmezlik pelerinde kullanılan malzemeler araca entegrasyonu sağlandığın görünmez bir keşif aracı olarak savunma sanayi gibi alanlarda kullanılabilir.

**VII. Ekler**

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_HMC5883\_U.h>

#include <TinyGPS++.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_PCD8544.h>

#define SCREEN\_WIDTH 84

#define SCREEN\_HEIGHT 48

#define echoPin 28

#define trigPin 24

#define MotorR1 3

#define MotorR2 4

#define MotorRE 2

#define MotorL1 5

#define MotorL2 6

#define MotorLE 7

Adafruit\_HMC5883\_Unified mag = Adafruit\_HMC5883\_Unified(12345);

TinyGPSPlus gps;

Adafruit\_PCD8544 display = Adafruit\_PCD8544(8, 9, 10, 11, 12);

long sure, uzaklik;

float headingDegrees;

void setup(void) {

  Serial.begin(9600);

  Serial.println(F("GPS ve Pusula Entegrasyonu"));

  display.begin();

  display.setContrast(60);

  if (!mag.begin()) {

    Serial.println(F("HMC5883 algılanamadı... Bağlantıları kontrol edin!"));

    while (1);

  }

  displaySensorDetails();

}

void loop(void) {

  // GPS verilerini oku

  while (Serial.available() > 0) {

    if (gps.encode(Serial.read())) {

      displayGPSInfo();

    }

  }

  sensors\_event\_t event;

  mag.getEvent(&event);

  displayCompassInfo(event);

  ileri();

  delay(5000);

  stopMotors();

  delay(0);

  sag();

  delay(750);

  stopMotors();

  ileri();

  delay(10000);

  stopMotors();

  solaDon();

  delay(750);

  stopMotors();

  ileri();

  delay(5000);

  stopMotors();

}

void displaySensorDetails(void) {

  sensor\_t sensor;

  mag.getSensor(&sensor);

  Serial.println("------------------------------------");

  Serial.print  ("Pusula Sensörü:       "); Serial.println(sensor.name);

  Serial.print  ("Pusula Sürücü Ver:    "); Serial.println(sensor.version);

  Serial.print  ("Pusula Benzersiz ID:   "); Serial.println(sensor.sensor\_id);

  Serial.print  ("Pusula Max Değer:    "); Serial.print(sensor.max\_value); Serial.println(" uT");

  Serial.print  ("Pusula Min Değer:    "); Serial.print(sensor.min\_value); Serial.println(" uT");

  Serial.print  ("Pusula Çözünürlük:   "); Serial.print(sensor.resolution); Serial.println(" uT");

  Serial.println("------------------------------------");

  Serial.println("");

  delay(500);

}

void displayGPSInfo() {

  display.clearDisplay();

Serial.print(F("Konum: "));

  if (gps.location.isValid()) {

    Serial.print(gps.location.lat(), 6);

    Serial.print(F(","));

    Serial.print(gps.location.lng(), 6);

  } else {

    Serial.print(F("GEÇERSİZ"));

  }

  Serial.print(F("  Tarih/Zaman: "));

  if (gps.date.isValid()) {

    Serial.print(gps.date.month());

    Serial.print(F("/"));

    Serial.print(gps.date.day());

    Serial.print(F("/"));

    Serial.print(gps.date.year());

  } else {

    Serial.print(F("GEÇERSİZ"));

  }

  Serial.print(F(" "));

  if (gps.time.isValid()) {

    if (gps.time.hour() < 10) Serial.print(F("0"));

    Serial.print(gps.time.hour());

    Serial.print(F(":"));

    if (gps.time.minute() < 10) Serial.print(F("0"));

    Serial.print(gps.time.minute());

    Serial.print(F(":"));

    if (gps.time.second() < 10) Serial.print(F("0"));

    Serial.print(gps.time.second());

    Serial.print(F("."));

    if (gps.time.centisecond() < 10) Serial.print(F("0"));

    Serial.print(gps.time.centisecond());

  } else {

    Serial.print(F("GEÇERSİZ"));

  }

  Serial.println();

  display.setTextSize(0.5);  // Yazı boyutunu küçült

  display.setCursor(0, 0);

  display.print(F("Lat: "));

  display.println(gps.location.lat(), 6);

  display.print(F("Lon: "));

  display.println(gps.location.lng(), 6);

  display.display();

}

void displayCompassInfo(sensors\_event\_t event) {

  display.clearDisplay();

  Serial.print(F("Manyetik Alan (uT): X=")); Serial.print(event.magnetic.x); Serial.print(F(" Y=")); Serial.print(event.magnetic.y); Serial.print(F(" Z=")); Serial.print(event.magnetic.z); Serial.println(F(" uT"));

  float heading = atan2(event.magnetic.y, event.magnetic.x);

  heading += declinationAngle();

  if (heading < 0) heading += 2 \* PI;

  if (heading > 2 \* PI) heading -= 2 \* PI;

  headingDegrees = heading \* 180 / PI;

    Serial.print(F("Başlık (derece): ")); Serial.println(headingDegrees);

  display.setTextSize(0.5);

  display.setCursor(0, 10);

  display.print(F("Manyetik Alan :"));

  display.setCursor(0, 20);

  display.print(F("X=")); display.println(event.magnetic.x);

  display.setCursor(0, 30);

  display.print(F("Y=")); display.println(event.magnetic.y);

  display.setCursor(0, 40);

  display.print(F("Z=")); display.println(event.magnetic.z);

  display.setCursor(0, 50);

  display.print(F("Konum Derece: "));

  display.print(headingDegrees);

  display.display();

}

float declinationAngle() {

  return 0.22;

}

void ileri() {

  digitalWrite(MotorR1, HIGH);

  digitalWrite(MotorR2, LOW);

  analogWrite(MotorRE, 150);

  digitalWrite(MotorL1, HIGH);

  digitalWrite(MotorL2, LOW);

  analogWrite(MotorLE, 150);

}

void sag() {

  digitalWrite(MotorR1, HIGH);

  digitalWrite(MotorR2, LOW);

  analogWrite(MotorRE, 0);

  digitalWrite(MotorL1, HIGH);

  digitalWrite(MotorL2, LOW);

  analogWrite(MotorLE, 150);

}

void solaDon() {

  digitalWrite(MotorR1, HIGH);

  digitalWrite(MotorR2, LOW);

  analogWrite(MotorRE, 150);

  digitalWrite(MotorL1, LOW);

  digitalWrite(MotorL2, HIGH);

  analogWrite(MotorLE, 0);

}

void stopMotors() {

  digitalWrite(MotorR1, LOW);

  digitalWrite(MotorR2, LOW);

  analogWrite(MotorRE, 0);

  digitalWrite(MotorL1, LOW);

  digitalWrite(MotorL2, LOW);

  analogWrite(MotorLE, 0);

}

#include "setup.h"

void setup() {

  CLKPR = 0x80;

  CLKPR = 0;

initializeScreenAndCamera();

}

void loop() {

  processFrame();

}

#ifndef LIVEOV7670\_SETUP\_H

#define LIVEOV7670\_SETUP\_H

#define EXAMPLE 3

void initializeScreenAndCamera();

void processFrame();

#endif

#ifndef \_ADAFRUIT\_ST7735\_MOD\_H\_

#define \_ADAFRUIT\_ST7735\_MOD\_H\_

#if ARDUINO >= 100

 #include "Arduino.h"

 #include "Print.h"

#else

 #include "WProgram.h"

#endif

#include <Adafruit\_GFX.h>

#if defined(\_\_SAM3X8E\_\_)

  #include <include/pio.h>

  #define PROGMEM

  #define pgm\_read\_byte(addr) (\*(const unsigned char \*)(addr))

  #define pgm\_read\_word(addr) (\*(const unsigned short \*)(addr))

  typedef unsigned char prog\_uchar;

#elif defined(\_\_AVR\_\_)

  #include <avr/pgmspace.h>

#elif defined(ESP8266)

  #include <pgmspace.h>

#endif

#if defined(\_\_SAM3X8E\_\_)

    #undef \_\_FlashStringHelper::F(string\_literal)

    #define F(string\_literal) string\_literal

#endif

#define INITR\_GREENTAB 0x0

#define INITR\_REDTAB   0x1

#define INITR\_BLACKTAB   0x2

#define INITR\_18GREENTAB    INITR\_GREENTAB

#define INITR\_18REDTAB      INITR\_REDTAB

#define INITR\_18BLACKTAB    INITR\_BLACKTAB

#define INITR\_144GREENTAB   0x1

#define ST7735\_TFTWIDTH  128

#define ST7735\_TFTHEIGHT\_144 128

#define ST7735\_TFTHEIGHT\_18  160

#define ST7735\_NOP     0x00

#define ST7735\_SWRESET 0x01

#define ST7735\_RDDID   0x04

#define ST7735\_RDDST   0x09

#define ST7735\_SLPIN   0x10

#define ST7735\_SLPOUT  0x11

#define ST7735\_PTLON   0x12

#define ST7735\_NORON   0x13

#define ST7735\_INVOFF  0x20

#define ST7735\_INVON   0x21

#define ST7735\_DISPOFF 0x28

#define ST7735\_DISPON  0x29

#define ST7735\_CASET   0x2A

#define ST7735\_RASET   0x2B

#define ST7735\_RAMWR   0x2C

#define ST7735\_RAMRD   0x2E

#define ST7735\_PTLAR   0x30

#define ST7735\_COLMOD  0x3A

#define ST7735\_MADCTL  0x36

#define ST7735\_FRMCTR1 0xB1

#define ST7735\_FRMCTR2 0xB2

#define ST7735\_FRMCTR3 0xB3

#define ST7735\_INVCTR  0xB4

#define ST7735\_DISSET5 0xB6

#define ST7735\_PWCTR1  0xC0

#define ST7735\_PWCTR2  0xC1

#define ST7735\_PWCTR3  0xC2

#define ST7735\_PWCTR4  0xC3

#define ST7735\_PWCTR5  0xC4

#define ST7735\_VMCTR1  0xC5

#define ST7735\_RDID1   0xDA

#define ST7735\_RDID2   0xDB

#define ST7735\_RDID3   0xDC

#define ST7735\_RDID4   0xDD

#define ST7735\_PWCTR6  0xFC

#define ST7735\_GMCTRP1 0xE0

#define ST7735\_GMCTRN1 0xE1

#define ST7735\_BLACK   0x0000

#define ST7735\_BLUE    0x001F

#define ST7735\_RED     0xF800

#define ST7735\_GREEN   0x07E0

#define ST7735\_CYAN    0x07FF

#define ST7735\_MAGENTA 0xF81F

#define ST7735\_YELLOW  0xFFE0

#define ST7735\_WHITE   0xFFFF

class Adafruit\_ST7735\_mod : public Adafruit\_GFX {

 public:

  Adafruit\_ST7735\_mod(int8\_t CS, int8\_t RS, int8\_t SID, int8\_t SCLK, int8\_t RST = -1);

  Adafruit\_ST7735\_mod(int8\_t CS, int8\_t RS, int8\_t RST = -1);

  void     initB(void),                             // for ST7735B displays

           initR(uint8\_t options = INITR\_GREENTAB), // for ST7735R

           setAddrWindow(uint8\_t x0, uint8\_t y0, uint8\_t x1, uint8\_t y1),

           startAddrWindow(uint8\_t x0, uint8\_t y0, uint8\_t x1, uint8\_t y1),

           endAddrWindow(),

           pushColor(uint16\_t color),

           fillScreen(uint16\_t color),

           drawPixel(int16\_t x, int16\_t y, uint16\_t color),

           drawFastVLine(int16\_t x, int16\_t y, int16\_t h, uint16\_t color),

           drawFastHLine(int16\_t x, int16\_t y, int16\_t w, uint16\_t color),

           fillRect(int16\_t x, int16\_t y, int16\_t w, int16\_t h, uint16\_t color),

           setRotation(uint8\_t r),

           invertDisplay(boolean i);

  uint16\_t Color565(uint8\_t r, uint8\_t g, uint8\_t b);

private:

  uint8\_t  tabcolor;

  void     spiwrite(uint8\_t),

           writecommand(uint8\_t c),

           writedata(uint8\_t d),

           commandList(const uint8\_t \*addr),

           commonInit(const uint8\_t \*cmdList);

//uint8\_t  spiread(void);

  boolean  hwSPI;

#if defined(\_\_AVR\_\_) || defined(CORE\_TEENSY)

  volatile uint8\_t \*dataport, \*clkport, \*csport, \*rsport;

  uint8\_t  \_cs, \_rs, \_rst, \_sid, \_sclk,

           datapinmask, clkpinmask, cspinmask, rspinmask,

           colstart, rowstart; // some displays need this changed

#elif defined(\_\_arm\_\_)

  volatile RwReg  \*dataport, \*clkport, \*csport, \*rsport;

  uint32\_t  \_cs, \_rs, \_sid, \_sclk,

            datapinmask, clkpinmask, cspinmask, rspinmask,

            colstart, rowstart; // some displays need this changed

  int32\_t   \_rst;  // Must use signed type since a -1 sentinel is assigned.

#endif

};

#endif

#ifndef \_BUFFEREDCAMERAOV7670\_80x120\_10HZ\_GRAYSCALE\_H

#define \_BUFFEREDCAMERAOV7670\_80x120\_10HZ\_GRAYSCALE\_H

#include "BufferedCameraOV7670.h"

class BufferedCameraOV7670\_80x120\_10hz\_Grayscale : public BufferedCameraOV7670<uint8\_t, 80, uint8\_t, 80, uint8\_t, 120> {

public:

    BufferedCameraOV7670\_80x120\_10hz\_Grayscale() : BufferedCameraOV7670(Resolution::RESOLUTION\_QQVGA\_160x120, CameraOV7670::PIXEL\_YUV422, 0) {};

    void readLine() override;

};

void BufferedCameraOV7670\_80x120\_10hz\_Grayscale::readLine() {

  pixelBuffer.writeBufferPadding = 0;

  waitForPixelClockLow();

  asm volatile("nop");

  asm volatile("nop");

  asm volatile("nop");

  for (uint8\_t i = 0; i<80; i++) {

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    asm volatile("nop");

    readPixelByte(pixelBuffer.readBuffer[i]);

  }

}

#endif