UAV Arduino

(پهپاد آردوینو)

فاطمه صادقی

موسسه آموزش عالی آپادانا – ۱۴۰۰۱۲۰۲۸۰۰۲

—

مبانی رباتیک

—

استاد: دکتر محمد زارع

تیر ۱۴۰۳

چکیده

این پروژه به منظور جمع‌آوری داده‌ها در مناطقی که برای انسان‌ها قابل دسترسی نیستند، مناطق خطرناک یا جمع‌آوری داده‌های آب و هوا طراحی شده است .هواپیما قادر است دما، رطوبت، تشعشعات (اشعه ایکس و گاما) به واحد میلی‌سیوورت(mSv) ، ارتفاع، مختصاتGPS ، ژیرو، نیروی G و دیگر داده‌ها را جمع‌آوری کند .از جنس فوم عایق و سیم گرم است، قطعات هواپیما را برش داده و سپس آن‌ها را بطور دستی تراشیده تا ظاهری زیبا و صاف داشته باشند .در مرحله ساخت مدار، از برد Arduino Uno استفاده شده و از اجزایی مانندMPU6050 Gyro ، سنسور دما و رطوبتDHT ، ژئیگر کانتر برای اندازه‌گیری تشعشعات هسته‌ای، BMPبرای فشار و ارتفاع و سایر اجزا استفاده شده است. سپس مدارها را به‌صورت لایه‌هایی روی هم قرار داده .در نهایت، با استفاده از روش انتقال تصویر با کاغذ گلاسه و چاپر لیزری، مدارهای PCB را ساخته و آن‌ها را به‌طور دستی تمیز کرده تا برای استفاده آماده شوند.این پروژه نشان‌دهنده توانایی‌های Arduino در ساخت هواپیماهای خودساخته و جمع‌آوری داده‌ها است.

THE PROCESS

پهپاد آردوینو:

پهپاد اردوینو (UAV Arduino) یک نوع پهپاد کوچک و قابل برنامه‌ریزی است که از برد کنترلی Arduino برای کنترل و مدیریت عملکردهای پروازی استفاده می‌کند. این پهپاد معمولاً شامل بخش‌های اصلی زیر است:

موتورها: پهپاد اردوینو دارای چهار موتور است که به آنها اجازه می‌دهد تا بالا و پایین حرکت کنند، به سمت‌های مختلف منحرف شوند و تعادل خود را حفظ کنند.

سنسورها: این پهپاد شامل سنسورهای مختلفی مانند ژیروسکوپ، شتاب‌سنج، قطب‌نما و سنسورهای ارتفاع مانند بارومتر است که به آن کمک می‌کند تا جهت‌یابی، تعادل و ارتفاع را کنترل کند.

برد کنترلی: برد کنترلی Arduino به عنوان مغز اصلی پهپاد عمل می‌کند و اطلاعات از سنسورها را دریافت کرده و دستورات پرواز را اجرا می‌کند.

ماژول‌های ارتباطی: پهپاد اردوینو ممکن است دارای ماژول‌های ارتباطی مانند بلوتوث یا وای‌فای باشد که به کاربر این امکان را می‌دهد تا پهپاد را از راه دور کنترل کند یا داده‌های پرواز را بر روی دستگاه خود نمایش دهد.

کاربردهای پهپاد اردوینو شامل، اما محدود به موارد زیر می‌شود:

پروژه‌های هوایی: این پهپاد می‌تواند برای ساخت و طراحی پروژه‌های هوایی خلاقانه و آموزشی مورد استفاده قرار گیرد.

عکاسی هوایی: با نصب دوربین بر روی پهپاد، می‌توان از آن برای عکاسی هوایی و فیلمبرداری هوایی استفاده کرد.

مانیتورینگ محیطی: پهپاد اردوینو می‌تواند برای مانیتورینگ و بررسی مناطق دسترس نیست و یا منطقه‌های خطرناک استفاده شود.

تحقیقات جغرافیایی: در تحقیقات جغرافیایی، پهپاد اردوینو می‌تواند برای جمع‌آوری داده‌های جغرافیایی و تصاویر هوایی استفاده شود.

مسابقات پهپاد: این نوع پهپاد برای شرکت در مسابقات پهپاد و رقابت‌های هوایی نیز قابل استفاده است.

با توجه به این کاربردها، پهپاد اردوینو به عنوان یک ابزار چندمنظوره و کارآمد در زمینه‌های مختلف مورد استفاده قرار می‌گیرد و به عنوان یک ابزار خلاقانه و تحقیقاتی بسیار مناسب است.

سخت افزار:

اردوینو uno

یک پلتفرم سخت افزاری و پردازشی است که به صورت کد باز یا Open-Source طراحی شده است. این پلتفرم بر پایه یک بورد I/O ساده و بر مبنای زبان processing/wiring طراحی شده. بورد Arduino هم برای طراحی مدارات مستقل و هم برای ارتباط با سیستمهای خارجی و نرم افزارها مناسب است.

ماژول IMU

از این ماژول جهت اندازه گیری کم نویز وضعیت جسم در شش جهت و ارتعاشات طولی و زاویه ای به صورت همزمان استفاده می گردد. دلیل انتخاب سنسور های با خروجی آنالوگ برای این ماژول همزمانی سیگنالهای خروجی و عدم تاخیر فاز در سنسورهای ژایرو و شتاب و همچنین افزایش پهنای باند فرکانسی می باشد.

ماژول فشار سنج بارومتریک

ماژول سنسور فشار بارومتریک BMP180 یک سنسور فشار دیجیتالی با دقت بالاست.فشار را در محدوده ی ۳۰۰ تا ۱۱۰۰ hPaاندازه گیری می کند BMP180 .جانشین مناسبی برای BMP085 است که نسل جدیدی از سنسورهای فشار با دقت بالا برای کاربردهای مصرفی به شمار میرود. مصرف انرژی بسیار پایین، این سنسور را برای استفاده در گوشی های همراه ، PDAها ، دستگاه های GPS و تجهیزات فضای باز مناسب کرده است.

ماژول GPS

ماژول gps تراشه‌ای است که دارای پردازنده و آنتن است و سیگنال‌ها را مستقیما از ماهواره‌ها و به‌شکل امواج رادیویی دریافت می‌کند. این ماژول از هر ماهواره اطلاعاتی را دریافت می‌کند و اگر بتواند حداقل چهارتا از ماهواره‌ها را شناسایی کند، می‌تواند به‌صورت دقیق مکان و زمان را محاسبه کند.

شمارشگر گایگر

شمارشگرگایگر (Geiger Counter) تشعشعات هسته ای را با اندازه گیری میزان انتشار تشعشعات یونیزه کننده ی ذرات آلفا،بتا و اشعه ی گاما اندازه گیری می کند. این دستگاه که با نام آشکارساز تشعشع نیز شناخته می شود، از دو جز اصلی تشکیل شده است: مدارات الکترونیک پردازشگر و لوله ی گایگر. لوله ی گایگر با گاز کم فشار نجیبی مانند هلیوم، آرگون یا نئون پرشده است .

کارت حافظه

برای دخیره کردن تمام داده های پرواز در یک کارت حافظه است .

نرم افزار:

مایکروسافت ویندوز ۱۰

آردوینو IDE

ابزار دستی و ماشین الات:

Dremel
برد pcb

۲.مراحل اجرا:

برای ساخت پهپاد آردوینو، مراحل زیر را طی میکنیم:

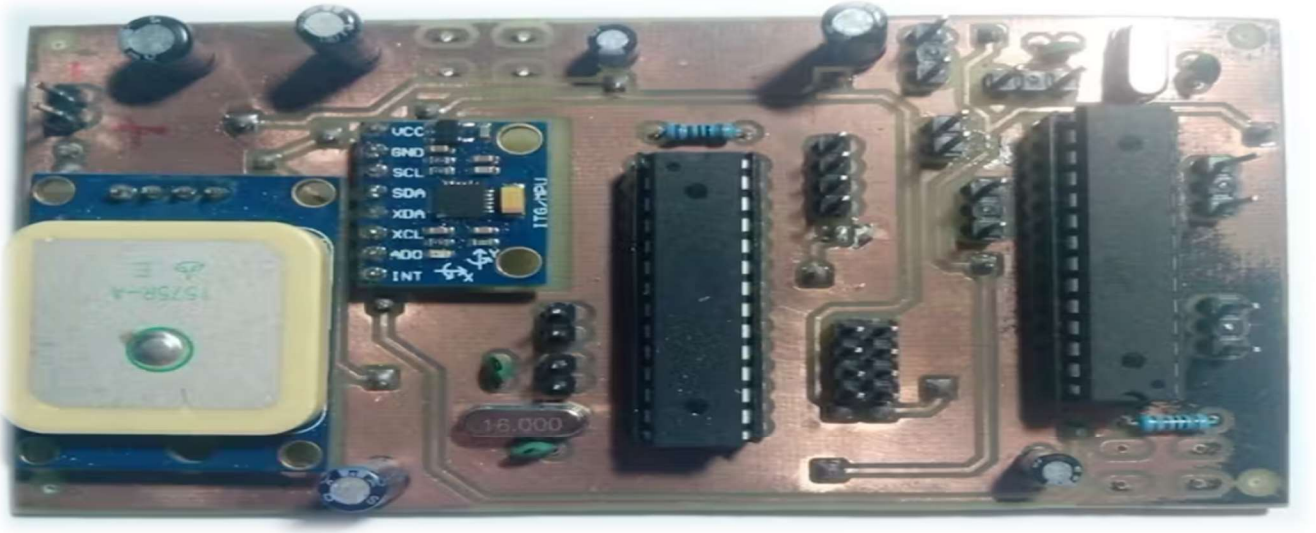
فرایند حکاکی

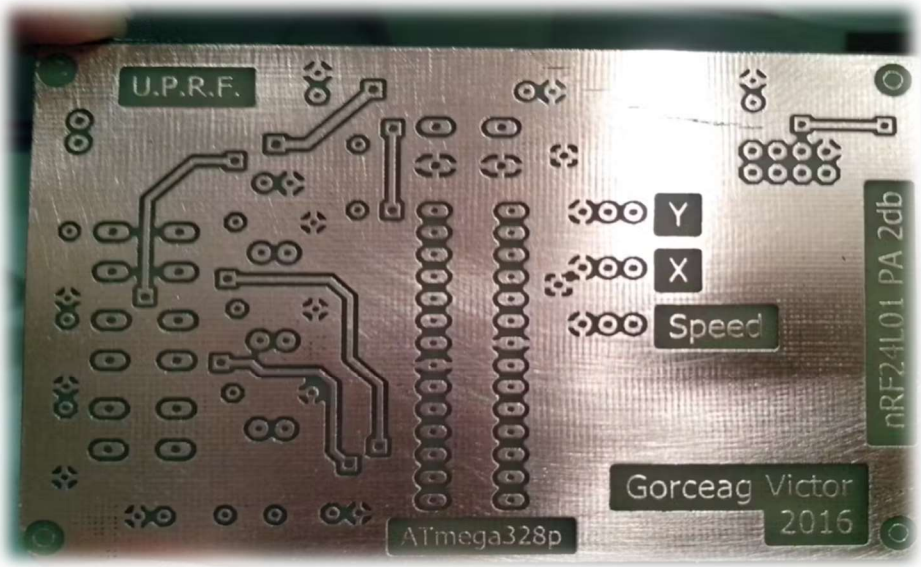
PCBها را با استفاده از نرم افزار EAGLE (نسخه رایگان) با استفاده از میکروکنترلر ATmega328p به عنوان پایه ساخته و قطعات مورد نیاز را اضافه کرده، از جمله دکمه ها، LEDها، رگولاتورها، پین های برای nRF24L01 و غیره.

با استفاده از فرآیند کاغذ انتقالی از پرینتر لیزری و کاغذ گلاسه عکس برای چاپ مدار استفاده کرده، سپس دقیقاً آن را برش داده و همچنین یک PCB مسی دو رو برش داده. بعد از آن با دقت کاغذها را تمیز کرده و قرار داده و آنها را با چسب کاغذ قفل کرده، حدود ۱۰ دقیقه هر دو طرف را با اتو زده، از طرف ها با فشار و دقت فشار زده. پس از ۱۰ دقیقه PCB را در آب قرار داده تا کاغذ مرطوب شود و خیلی آسان بشود که جدا شود.

بسیار دقت کنید/به آرامی کاغذ را جدا کنید، پس از تمیز شدن تمام کاغذ زمان حکاکی است، با استفاده از آب گرم فرآیند را سریعتر انجام دهید و با اسفنج کوچکی که در تصویر نیست با دقت آن را تمیز کنید.

در پایان از یک اسفنج شنی بالا یا یک اسفنج تمیز کننده بشقاب برای پاک کردن تمام جوهر سیاه / رنگ روی PCB استفاده کنید و در نهایت یک PCB خانگی تمیز و آماده برای استفاده بگیرید.





فریم :فریم یا قاب همان ستون فقرات کوادکوپتر است! فریم تمام قسمت های کوادکوپتر ما را در کنار هم نگه میدارد. فریم باید سبک باشد اما از استحکام مناسبی نیز برخوردار باشد. تمام موتور ها، باتری ها و ... بر روی فریم سوار میشوند. برای قسمت فریم، شما میتوانید خودتان این فریم را بسازید یا از فریم های آماده استفاده کنید. توجه داشته باشید که فریم شما باید دارای قسمت های نصب وسایل الکتریکی، چهار بازو برای قرار دادن ملخ ها و چهار محل برای نصب موتور ها باشد. این قاب می تواند از آلومینیوم ، الیاف کربن یا چوب ساخته شود اما ماده ای که بیشتر برای بازوها استفاده می شود آلومینیوم است. آنها نسبتاً سبک ، سفت و سخت و ارزان هستند. اما میتوانند باعث ایجاد خطا در سنسور ها شوند.فیبر کربن خیلی بهتر لرزش موتور ها را جذب میکند و سفت و سخت تر است. اما از همه انواع فریم ها گرانترین نیز هست. فیبر کربن گزینه برتر است ، اما این خیلی به بودجه شخصی شما بستگی دارد.

موتور :موتور ها هوا را با سرعت بالا منتقل میکنند که باعث میشود کوادکوپتر حرکت کند. هر کدام از موتور ها توسط یک ESC یا همان کنترل کننده سرعت، کنترل میشوند. در کوادکوپتر ها از موتور براشلس استفاده میشود. موتور های براشلس تقریبا شبیه به موتور های DC معمولی هستند. اما دارای ویژگی های خاصی هستند که استفاده از آن ها را برای کوادکوپتر ها بسیار مناسب میکند. موتور های مورد استفاده ما باید بتوانند هم در جهت عقربه های ساعت، و هم در خلاف جهت عقربه های ساعت بچرخند.

ESC : کنترل کننده سرعت الکترونیکی، مانند عصب در بدن ما عمل میکند ESC .ها میزان برق موتور ها را تعیین میکند که باعث تعیین جهت سرعت حرکت کوادکوپتر میشود. دستگاهی که وظیفه کنترل سرعت موتورها را بر عهده دارد یک برد کنترل کننده ارزان قیمت است که فقط برای موتور ها استفاده می شود. هر ESC یک ورودی برای یک باتری دارد و دارای یک خروجی موتور با سه فاز است. هنگام خریدESC ، به سطح جریان الکتریکی خروجی آن توجه کنید. پیشنهاد میشود ESC را با حداقل جریان ۱۰ آمپر خریداری کنید.

ملخ :بسته به نوع کواد کوپتری که می سازید، می توانید از ملخ های ۹ تا ۱۰ یا ۱۱ اینچی (برای پروازهای ثابت و عکاسی هوایی) استفاده کنید. حتی میتوانید از ملخ های ۵ اینچی نیز برای کاربرد های ساده استفاده کنید. رچقدر قطر بیشتر باشد، ملخ رانش بیشتری ایجاد میکند و در نتیجه به برق بیشتری احتیاج دارد اما میتواند وزن بیشتری را بلند کند. برای موتور های دور بالا شما باید از ملخ های کوچک استفاده کنید. برای موتور های دور پایین شما به ملخ های بزرگتر نیاز دارید تا بتوانید با سرعت کمتر، کوادکوپتر را در هوا نگه دارید.

باتری :بسته به میزان ولتاژ مورد نیاز، میتوانید از باتری های ۲ S، 3S، 4Sیا حتی ۵ Sاستفاده کنید. اما برای استاندارد شما از باتری ۳ S 11.4Vمیتوانید استفاده کنید. اگر سرعت بالا میخواهید میتوانید از باتری های ٤ Sاستفاده کنید.

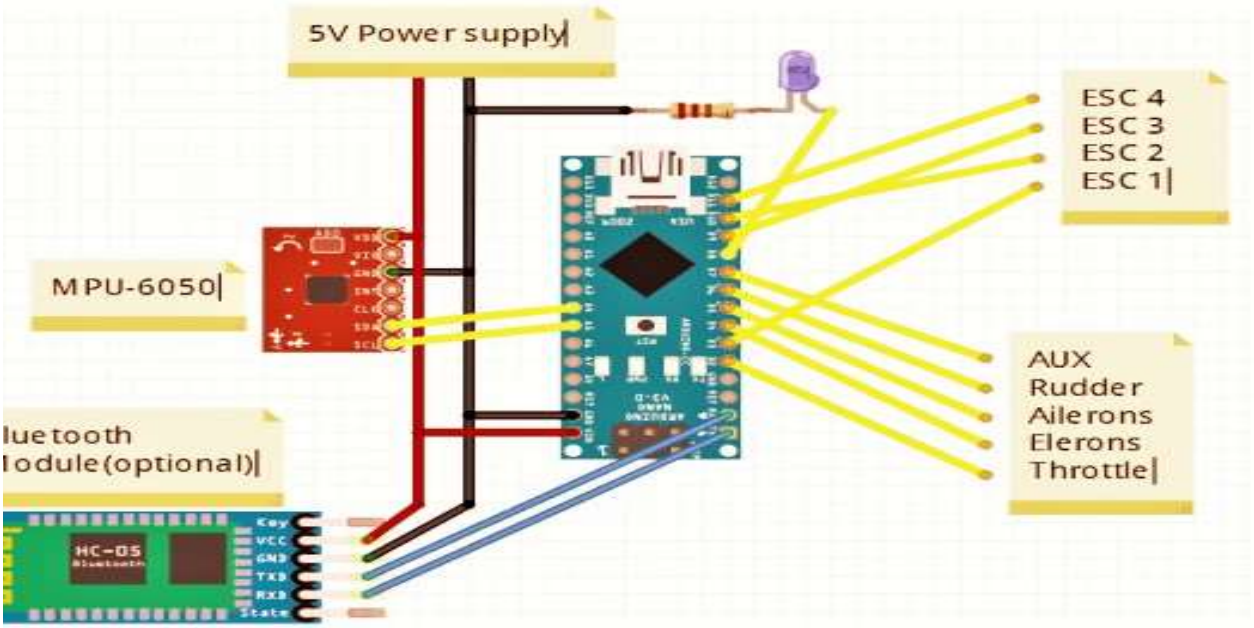
برد آردوینو :از برد آردوینو به عنوان مغز این پروژه استفاده میشود. ا استفاده از برد آردوینو اونو و انتخاب قطعات مناسب، ما یک فلایت کنترل (کنترل کننده پرواز) حرفه ای برای کوادکوپتر خود طراحی میکنیم.

کنترلر RC : دسته کنترل کننده است که برای کنترل کردن کوادکوپتر استفاده میشود. متداول ترین روش برنامه نویسی و کنترل کوادکوپتر استفاده از فرستنده RC است. معمولاً می توانید حالت آکروباتیک (Acrobatic) یا پایدار (Stable) را انتخاب کنید. در حالت آکروباتیک، ژیروسکوپ مقادیر را به پردازشگر میفرستد، در این حالت جوی استیک ها فقط برای کنترل و تنظیم سرعت وجود دارند و اگر آنها را رها کنید، کوادکوپتر در حالت متعادل نگه داشته نمیشود. این حالت برای مبتدیان مناسب نیست زیرا کوادکوپتر به راحتی کج میشود و کنترل آن سخت است.

مدار پهپاد با اردوینو:

این پیچیده ترین قسمت در مراحل ساخت کوادکوپتر با آردوینو است. لحیم کاری یک تکنیک بسیار خاص است، بنابراین حتماً این فرآیند را با دقت انجام دهید. در این قسمت به قطعات زیر نیاز داریم:

آردوینو NANO یاUNO
ماژولMPU-6050
مقاومت 330 اهم
LED
ماژول بلوتوثHC-05
برد سوراخدار
سیم نازک



نحوه اتصال ESC ها:

- سیگنال پینD3 – ESC 1
- سیگنال پینD9 – ESC 3
- سیگنال پینD10 – ESC 2
- سیگنال پینD11 – ESC 4

نحوه اتصال ماژول بلوتوث:

- Tx – Rx
- Rx – Tx

نحوه اتصال:MPU-6050

- SDA – A4
- SCL – A5

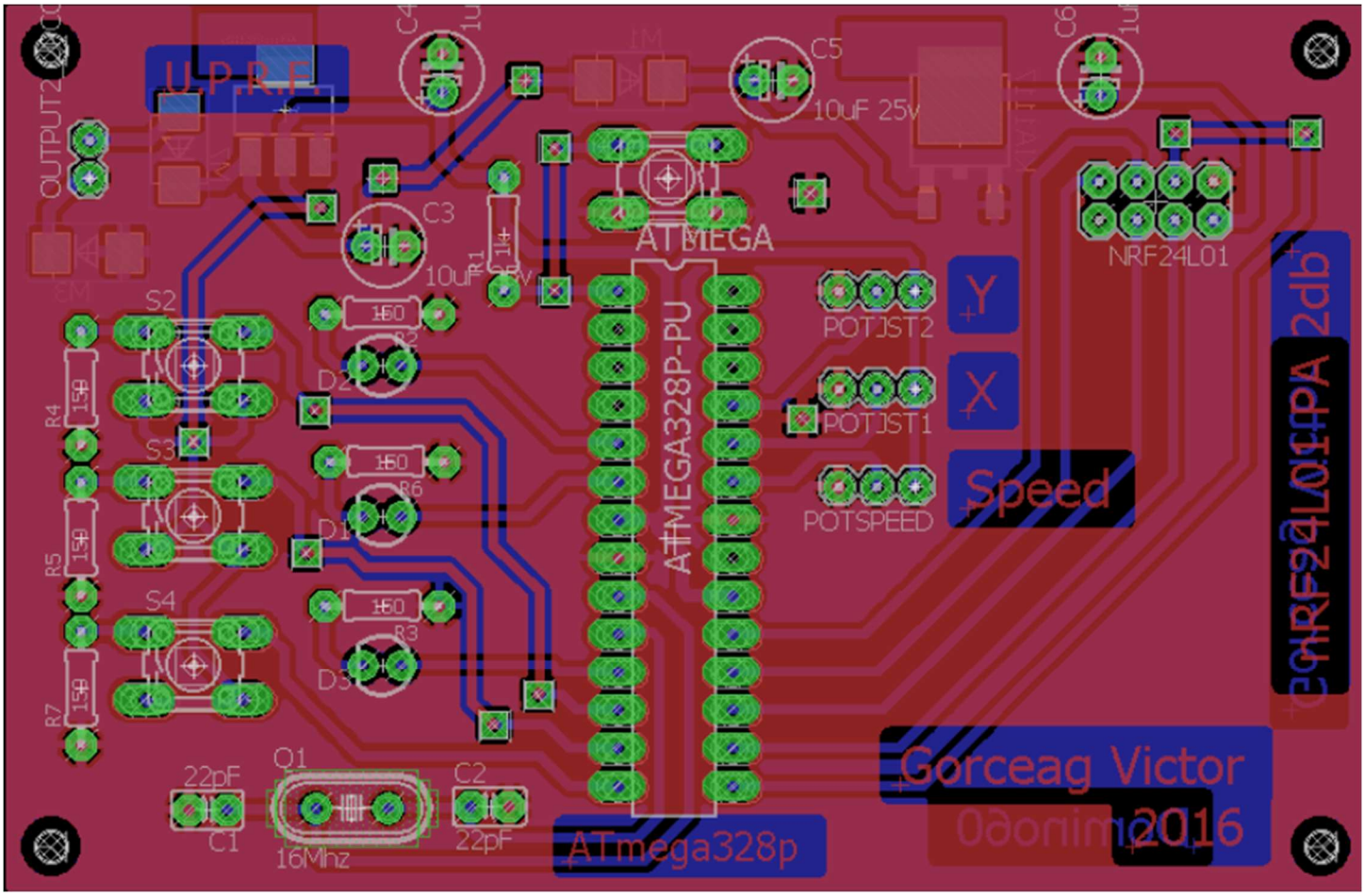
نحوه اتصال:LED

- پایه آند8D – LED

نحوه اتصال گیرنده:

- Throttle – 2
- Elerons – D4
- Ailerons – D5
- Rudder – D6
- AUX 1 – D7

پایه های GND مازول6050-MPU ، ماژول بلوتوث ، گیرنده و ESC را به پین GND آردوینو متصل کنید.



```
/* ©Gorceag Victor / Domino60 2016 / Arduino 1.0.5
   Components/modules

   - Joystick 3 Potentiometers
   - nRF24L01 PA 2db antenna
   - 3x Buttons 3x Leds

*/

//------------------------------------ nRF24
#include <nRF24L01.h>
#include <RF24.h>
#include <RF24_config.h>
#include <SPI.h>


RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;


int dataToSend[6];

//---------------------------------------POT's
int PotSpeed;
int PotY;
int PotX;

long nowP = 0;
long intervalP = 60;

void Potentiometers(){

    if(millis() - nowP >= intervalP){
   PotSpeed = analogRead(A0);
   PotY = analogRead(A1);
   PotX = analogRead(A2);
     nowP = millis();
    }
}

// ********************************************************* LED_BUTTON 1
int led1 = 3;
int button1 = 6;

long now1 = 0;
long interval1 = 1000;
int x1 = 0;
void Button1_Led1(){
   if(millis() - now1 >= interval1){
  int valofbutton1 = digitalRead(button1); //value of the button 0 or 1 (LOW or HIGH)
   if(valofbutton1 == 1){ //if button HIGH
     x1++;  // x+1;  x=1
     if(x1 == 1){  //if x = 1
        digitalWrite(led1, HIGH); //then led HIGH
     }
     if(x1 == 2){  // if we press again x=2
        digitalWrite(led1, LOW); // if x = 2 led LOW
        x1 =0;   // making x = 0 for the next loop
     }
   }
   now1 = millis();
  }
}
```

```
// ********************************************************** LED_BUTTON 2
int led2 = 4;
int button2 = 7;

long now2 = 0;
long interval2 = 1000;
int x2 = 0;
void Button1_Led2(){
   if(millis() - now2 >= interval2){
  int valofbutton2 = digitalRead(button2); //value of the button 0 or 1 (LOW or HIGH)
   if(valofbutton2 == 1){ //if button HIGH
     x2++;  // x+1;  x=1
     if(x2 == 1){  //if x = 1
        digitalWrite(led2, HIGH); //then led HIGH
     }
if(x2 == 2){  // if we press again x=2
        digitalWrite(led2, LOW); // if x = 2 led LOW
        x2 =0;   // making x = 0 for the next loop
     }
   }
   now2 = millis();
  }
}

// ********************************************************** LED_BUTTON 3
int led3 = 5;
int button3 = 8;

long now3 = 0;
long interval3 = 1000;
int x3 = 0;
void Button1_Led3(){
   if(millis() - now3 >= interval3){
  int valofbutton3 = digitalRead(button3); //value of the button 0 or 1 (LOW or HIGH)
   if(valofbutton3 == 1){ //if button HIGH
     x3++;  // x+1;  x=1
     if(x3 == 1){  //if x = 1
        digitalWrite(led3, HIGH); //then led HIGH
     }
     if(x3 == 2){  // if we press again x=2
        digitalWrite(led3, LOW); // if x = 2 led LOW
        x3 =0;   // making x = 0 for the next loop
     }
   }
   now3 = millis();
  }
}

// ----------------------------------------SETUP
void setup() {
  Serial.begin(9600);

   radio.begin();
  radio.setAutoAck(false);
  radio.setChannel(108); //108 - 2.508 Ghz //0-124 2.4gHz-2.5gHz
  radio.setDataRate(RF24_250KBPS);
  radio.setPALevel(RF24_PA_MAX);
  radio.openWritingPipe(pipe);

  pinMode(led1, OUTPUT);
  pinMode(button1, INPUT);
  pinMode(led2, OUTPUT);
pinMode(button3, INPUT);
  pinMode(led3, OUTPUT);
  pinMode(button3, INPUT);
}
```

```
// ---------------------------------------LOOP
void loop() {

  Button1_Led1();
  Button1_Led2();
  Button1_Led3();
  Potentiometers();

    dataToSend[0] = PotSpeed;
    dataToSend[1] = PotY;
    dataToSend[2] = PotX;
    dataToSend[3] = x1;
    dataToSend[4] = x2;
    dataToSend[5] = x3;
    radio.write(dataToSend, sizeof(dataToSend));
}
```

توضیحات توابع:

Void setup():این کد یک برنامه Arduino است که از یک ماژول رادیویی RF24 برای ارسال و دریافت داده استفاده می‌کند. این باز کردن و آماده‌سازی ماژول رادیویی RF24 را به منظور ارسال و دریافت داده‌ها انجام می‌دهد. همچنین پورت‌ها و پین‌های ورودی و خروجی را برای کنترلLED ها و دکمه‌ها تنظیم می‌کند.

loop() :کار تابع بالا این است که ابتدا وضعیت دکمه‌ها وLED ها را بررسی کند و سپس مقادیر خروجی پتانسیومترها را در آرایه `dataToSend`قرار دهد. سپس این آرایه را با استفاده از ماژول رادیویی RF24 به ماژول دیگری ارسال می‌کند. به این ترتیب، داده‌هایی که توسط پتانسیومترها اندازه‌گیری شده‌اند، به ماژول دیگر منتقل می‌شوند.

Void potentiometers(): این تابع یک تابع با نام "Potentiometers" است که به صورت دوره‌ای اجرا می‌شود. این تابع از تابع millis() برای بررسی گذشت زمان استفاده می‌کند و اگر زمان مشخص شده (intervalP) از زمان قبلی گذشته باشد، مقادیر پتانسیومترها را از ورودی‌های آنالوگ خوانده و در متغیرهایPotSpeed ، PotY وPotX ذخیره می‌کند. سپس مقدار فعلی زمان را به عنوان nowP ذخیره می‌کند تا برای بار دیگر اجرا شدن تابع، زمان قبلی را مشخص کند.

Void button1_led1,2,3(): وضعیت دیجیتال دکمه (button1) را خوانده و در متغیر valofbutton1 ذخیره می‌کند. سپس اگروضعیت دکمه (1) HIGH باشد، متغیر x1 را افزایش داده و اگر x1 برابر ۱ باشد، وضعیت پین LED (led1) را بهHIGH تغییرمی‌دهد. اگر x1 برابر ۲ شود، وضعیت پین LED را به LOW تغییر می‌دهد و مقدار x1 را صفر می‌کند تا برای بار دیگر اجرا شدن تابع، آماده باشد. سپس مقدار فعلی زمان را به عنوان now1 ذخیره می‌کند.

منابع:

https://www.hackster.io/Domino60/uav-arduino-6432b5#story
https://projecthub.arduino.cc
https://www.arduino.cc
https://copilot.microsoft.com
https://irenx.ir/arduino/arduino-project/arduino-quadcopter/