

# پهپاد آردوینو

افسانه تربتی

—

مبانی رباتیک

—

استاد: محمد زارع

تیر ۱۴۰۳

## چکیده

این پروژه ، یک هواپیمای بدون سرنشین (UAV) است که به طور کامل با استفاده از Arduino ساخته شده است. هدف اصلی این پروژه جمع‌آوری داده‌ها در مناطقی است که دسترسی به آن‌ها برای انسان‌ها دشوار یا خطرناک است، یا جمع‌آوری داده‌های آب و هوایی. این هواپیما قادر به جمع‌آوری داده‌هایی مانند دما، رطوبت، تابش (اشعه ایکس و گاما)، ارتفاع، مختصات GPS ، ژيروسکوپ و نیروی G است.





## THE PROCESS

---

### مقدمه

پروژه UAV Arduino ، یک هواپیمای بدون سرنشین (UAV) است که به طور کامل با استفاده از Arduino ساخته شده است. این پروژه توسط Gorceag Victor برای فارغ التحصیلی از دانشگاه طراحی و اجرا شده است. هدف اصلی این پروژه، ساخت یک هواپیمای بدون سرنشین خانگی با استفاده از قطعات و ابزارهای ساده و مقرون به صرفه است.

این هواپیما قادر به جمع آوری داده های مختلفی از جمله دما، رطوبت، تابش (اشعه ایکس و گاما)، ارتفاع، مختصات GPS ، ژيروسکوپ و نیروی G است. این داده ها می توانند در مناطقی که دسترسی به آنها برای انسان ها دشوار یا خطرناک است، جمع آوری شوند. همچنین، این هواپیما می تواند برای جمع آوری داده های آب و هوایی نیز مورد استفاده قرار گیرد.

برای ساخت این هواپیما، از قطعاتی مانند Arduino UNO ، سنسورهای مختلف، واحد اندازه گیری اینرسی (IMU) ، GPS ، و شمارنده گایگر استفاده شده است. فرآیند ساخت شامل طراحی و ساخت مدارهای چاپی (PCB) خانگی، مونتاژ قطعات، و آزمایش موتورهای براشلس بوده است. این پروژه نشان دهنده توانایی ساخت یک سیستم پیچیده با استفاده از ابزارها و تکنولوژی های ساده و در دسترس است.

---

### ۱. مواد و قطعات مورد نیاز:

- میکروکنترلر **Arduino UNO**
- واحد اندازه گیری اینرسی (IMU) (۶ درجه آزادی)
- bmp180
- **GPS ATtiny**
- **ژیروسکوپ MPU6050 Axx**
- جی پی اس
- سنسور رطوبت دما **DHT**
- شمارنده گایگر برای سنسور تشعشع هسته ای **mSv**

### ۲. مراحل اجرا:

#### ۱. طراحی و ساخت مدار:

- استفاده از **Arduino UNO** و چندین **Atmega328p**
- طراحی مدار با استفاده از نرم افزار **EAGLE**
- ساخت **PCB** به صورت دستی با استفاده از فرآیند انتقال کاغذ و اتو کردن.

#### ۲. مونتاژ قطعات:

- استفاده از سنسورهای مختلف مانند ژيروسکوپ و شتاب سنج، **GPS**، سنسور دما و رطوبت، و سنسور فشار.
- مونتاژ قطعات روی **PCB** و اتصال آن‌ها به **Arduino**

#### ۳. ساخت بدنه هواپیما:

- استفاده از فوم عایق و سیم داغ برای برش قطعات.
- سنباده کاری دستی برای صاف و زیبا کردن قطعات.

#### ۴. تست و تنظیمات:

- تست موتورهای براشلس.
- تنظیمات و کالیبراسیون سنسورها و سیستم‌های کنترلی.

#### ۵. نرم افزار و برنامه نویسی:

- استفاده از **Arduino IDE** برای برنامه نویسی.
  - ذخیره داده‌ها در کارت **SD** به عنوان جعبه سیاه.
-

```

/* @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:

#### ۱. Potentiometers():

- این تابع مقادیر سه پتانسیومتر را می‌خواند و آن‌ها را در متغیرهای `PotSpeed` ، `PotY` و `PotX` ذخیره می‌کند. این مقادیر هر ۶۰ میلی‌ثانیه یکبار به‌روزرسانی می‌شوند.

#### ۲. Button1\_Led1():

- این تابع وضعیت دکمه `button1` را بررسی می‌کند. اگر دکمه فشار داده شود، مقدار `x1` افزایش می‌یابد و چراغ `led1` روشن یا خاموش می‌شود. اگر `x1` برابر با ۲ شود، دوباره به ۰ بازنشانی می‌شود.

#### ۳. Button1\_Led2():

- این تابع مشابه تابع قبلی است، اما برای دکمه `button2` و چراغ `led2` عمل می‌کند.

#### ۴. Button1\_Led3():

- این تابع نیز مشابه دو تابع قبلی است، اما برای دکمه `button3` و چراغ `led3` عمل می‌کند.

---

۵: `setup()` .

- این تابع برای تنظیمات اولیه استفاده می‌شود. در اینجا، ارتباط رادیویی با استفاده از ماژول `nRF24L01` تنظیم می‌شود و پین‌های ورودی و خروجی برای دکمه‌ها و چراغ‌ها تعریف می‌شوند.

۶: `loop()` .

- این تابع به صورت مداوم اجرا می‌شود و توابع `Button1\_Led1` ، `Button1\_Led2` ، `Button1\_Led3` و `Potentiometers` را فراخوانی می‌کند. همچنین، مقادیر خوانده شده از پتانسیومترها و وضعیت دکمه‌ها را از طریق ارتباط رادیویی ارسال می‌کند.

---

## نتیجه گیری

پروژه UAV Arduino ، نمونه‌ای برجسته از خلاقیت و نوآوری در استفاده از تکنولوژی‌های ساده و مقرون‌به‌صرفه برای ساخت یک سیستم پیچیده است. این پروژه نشان می‌دهد که چگونه می‌توان با استفاده از قطعات و ابزارهای در دسترس، یک هواپیمای بدون سرنشین (UAV) کاملاً کاربردی ساخت.

یکی از نوآوری‌های این پروژه، استفاده از Arduino UNO به عنوان مغز متفکر سیستم است. این انتخاب به دلیل سادگی و قابلیت‌های گسترده Arduino در کنترل سنسورها و ماژول‌های مختلف انجام شده است. همچنین، طراحی و ساخت مدارهای چاپی (PCB) خانگی با استفاده از فرآیندهای دستی و ابزارهای ساده، نشان‌دهنده خلاقیت و مهارت‌های فنی بالای سازنده است<sup>1</sup>.

فواید این پروژه در یادگیری رباتیک و Arduino بسیار زیاد است. این پروژه به دانشجویان و علاقه‌مندان به رباتیک و الکترونیک نشان می‌دهد که چگونه می‌توان با استفاده از منابع محدود و ابزارهای ساده، پروژه‌های پیچیده و کاربردی را اجرا کرد. همچنین، این پروژه فرصتی برای یادگیری و تمرین مهارت‌های مختلف از جمله طراحی مدار، برنامه‌نویسی، و کار با سنسورها و ماژول‌های مختلف فراهم می‌کند.

به طور کلی، پروژه UAV Arduino یک مثال عالی از خلاقیت و نوآوری در استفاده از تکنولوژی‌های ساده برای حل مسائل پیچیده است و می‌تواند الهام‌بخش بسیاری از علاقه‌مندان به رباتیک و الکترونیک باشد<sup>1</sup>.

منبع

UAV Arduino

<https://www.hackster.io/Domino60/uav-arduino-6432b5>

---