1. Système de propulsion

```
#include<Servo.h>
int value = 0;
Servo firstESC, secondESC;
void setup() {
 firstESC.attach(9);
 Serial.begin(9600);
}

void loop() {
 firstESC.write(value);
 if(Serial.available())
   value = Serial.parseInt();
```

Système de contrôle de vol

```
#include <Wire.h>
#include <MPU6050.h>
MPU6050 mpu;
void setup (){
 Wire.beggin();
 mpu.initialize();
 Serial.begin(9600);
}
void loop(){
  Vector3f gyro = mpu.getRotation();
  Vector3f accel = mpu.getAcceleration();
  Serial.print("Gyro (deg/s): ");
  Serial.print(gyro.x);
  Serial.print(", ");
  Serial.print(gyro.y);
  Serial.print(", ");
  Serial.print(gyro.z);
  Serial.print(", ");
  Serial.print("Accel (m/s2): ");
  Serial.print(accel.x);
  Serial.print(", ");
  Serial.print(accel.y);
  Serial.print(", ");
  Serial.print(accel.z);
  Serial.println();
  delay(100);
}
```

2. **Système de navigation**

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
const int RX_PIN = 2;
const int TX_PIN = 3;
SoftwareSerial gpsSerial(RX_PIN, TX_PIN);
TinyGPSPlus gps;
void setup (){
 serial.beggin(9600);
 gpsSerial.begin(9600);
}
void loop(){
  while (gpsSerial.available() > 0){
    char c = gpsSerial.read();
    if(gps.encode(c)) {
      if(gps.location.isValid()){
  Serial.print("Latitude");
  Serial.print(gps.location.lat(), 6);
  Serial.print(", longitude: ");
  Serial.print(gps.location.lng(), 6);
      }
    }
  }
}
```

3. **Système de commande**

Emetteur

```
#include <SoftwareSerial.h>
SoftwareSerial communicationSerial(2, 3);
void setup() {
  Serial.begin(9600);
  communicationSerial.begin(9600);
}
void loop() {
  if (Serial.available()) {
    char dataToSend = Serial.read();
    communicationSerial.write(dataToSend);
  }
}
```

Recepteur

```
#include <SoftwareSerial.h>
SoftwareSerial communicationSerial(2, 3);
void setup() {
  Serial.begin(9600);
  communicationSerial.begin(9600);
```

```
}
void loop() {
  if (communicationSerial.available()) {
    char receivedChar = communicationSerial.read();
  }
}
```