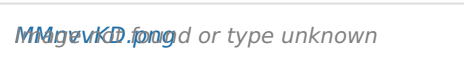
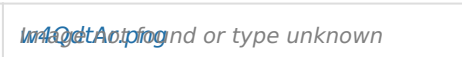
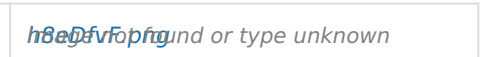


# [exemplo] rumo ao DEEC

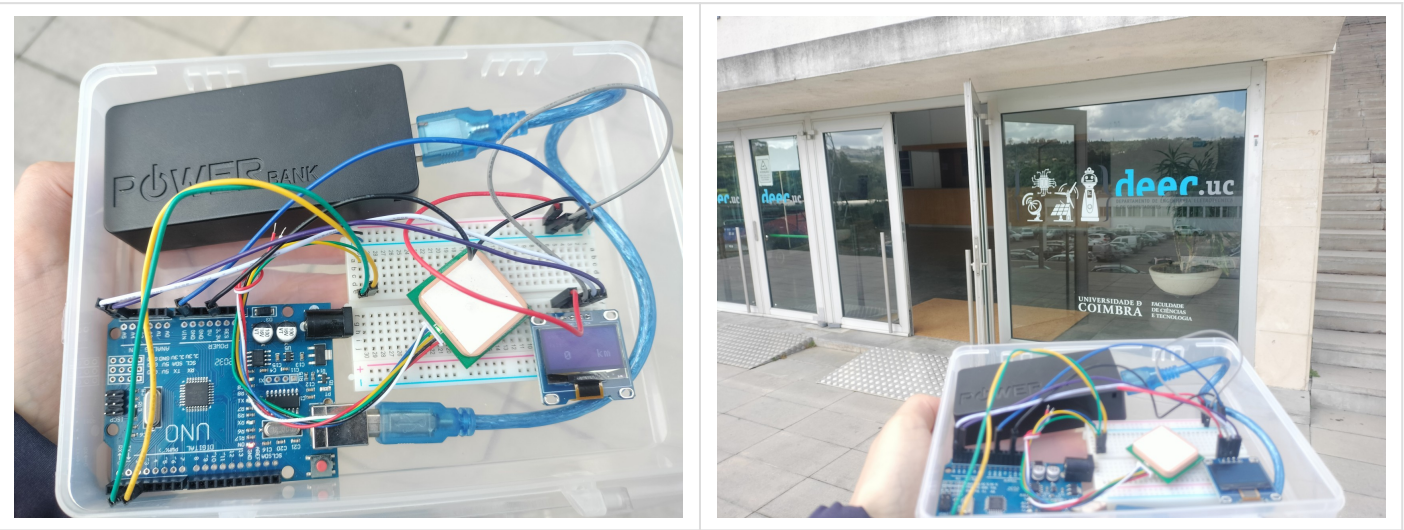
## Objetivo:

Chegar ao DEEC - FCTUC via sistema de posicionamento global com Arduino.

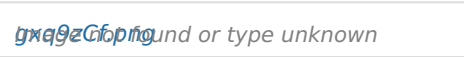

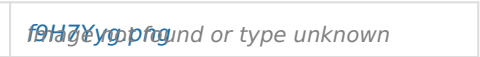
## Requisitos

		
[a]	[b]	[c]

## [exemplo] rumo ao DEEC



## Set-up

		
Arduino compatível UNO	módulo GPS	ecrã OLED
3.3V	V	VCC
GND		GND

-----	G		
2	-----	R	
3	-----	T	
A4	-----		SDA
A5	-----		SDL

## Programa

```
#include <SoftwareSerial.h>
#include <TinyGPSPlus.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// OLED display TWI address

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET      -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address;
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

SoftwareSerial gpsSerial(2, 3); // RX, TX
TinyGPSPlus gps;
bool selector = false;

void setup() {
  // inicializa ecrã
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
  display.clearDisplay();
  display.display();

  // imprime linha de texto
  display.setTextSize(1);
```

```

display.setTextColor(WHITE);
display.setCursor(27, 30);
display.print(F("..."));

// actualiza ecrã
display.display();

gpsSerial.begin(9600);
Serial.begin(115200);
}

void loop() {

    while (gpsSerial.available() > 0){
        if (gps.encode(gpsSerial.read()))
            displayInfo();
    }

    if (millis() > 5000 && gps.charsProcessed() < 10)
    {
        Serial.println(F("No GPS detected: check wiring."));
        while(true);
    }
}

void displayInfo() {
    bool chalk_a = true;
    bool chalk_b = true;
    selector = false;
    Serial.print(F("Location: "));

    if (gps.location.isValid()) {
        Serial.print(gps.location.lat(), 12);
        Serial.print(F(", "));
        Serial.print(gps.location.lng(), 12);

        if (chalk_a) {
            chalk_a = false;
        }
    }
}

```

```
} else {
    Serial.print(F("INVALID"));
}

Serial.print(F("  Date/Time: "));
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("INVALID"));
}

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());

    if (chalk_b) {
        chalk_b = false;
    }
} else {
    Serial.print(F("INVALID"));
}

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
```

```

if (!chalk_a) {
    static const double DEECFCTUC_LAT = 40.18639388156963, DEECFCTUC_LON = -8.416520235329802;
    unsigned long distanceKmToDEEC =
    (unsigned long)TinyGPSPlus::distanceBetween(
        gps.location.lat(),
        gps.location.lng(),
        DEECFCTUC_LAT,
        DEECFCTUC_LON);

    if (distanceKmToDEEC > 1000) {
        distanceKmToDEEC = distanceKmToDEEC / 1000;
        selector = true;
    }

    Serial.println(distanceKmToDEEC);

    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(30, 35);
    display.print(distanceKmToDEEC);
    display.setCursor(90, 35);
    if (selector) {
        display.print(F("km"));
    } else {
        display.print(F("m"));
    }

    display.setTextSize(1);
    display.setTextColor(WHITE);
} else {
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(40, 35);
    display.print(F("..."));
    display.setCursor(90, 35);
    if (selector) {
        display.print(F("km"));
    } else {
        display.print(F("m"));
    }
}

```

```

    display.setTextSize(1);
    display.setTextColor(WHITE);
}

display.setCursor(5, 5);
display.print(F("agora"));
display.setCursor(5, 20);
display.print(F("faltam"));
display.setCursor(5, 55);
display.print(F("para chegar ao DEEC"));

if (!chalk_b) {

    display.setCursor(50, 5);
    display.print(gps.time.hour());
    display.setCursor(63, 5);
    display.print(":");
    display.setCursor(70, 5);
    display.print(gps.time.minute());
    display.setCursor(83, 5);

}
else {
    display.setCursor(50, 5);
    display.print(F("00"));
    display.setCursor(63, 5);
    display.print(":");
    display.setCursor(70, 5);
    display.print(F("00"));
    display.setCursor(83, 5);

}
display.display();
Serial.println();
}

```

## Outros exemplos

<a href="#">MMnvvKD.png</a>	<a href="#">w4QdtAr.png</a>	<a href="#">h8eDfvF.png</a>
Image not found or type unknown	Image not found or type unknown	Image not found or type unknown
[a]	[b]	[c]

[a] DEEC - FCTUC. Kit Arduino Uno > Instalação. url:  
<https://kb.deec.uc.pt/books/deec/page/instalacao-irK> (acedido em 10/04/2024).

[b] DEEC - FCTUC. Kit Arduino Uno > [exemplo] módulo GPS. url:  
<https://kb.deec.uc.pt/books/deec/page/exemplo-modulo-gps> (acedido em 10/04/2024).

[c] DEEC - FCTUC. Kit Arduino Uno > [exemplo] I2C OLED. url:  
<https://kb.deec.uc.pt/books/deec/page/exemplo-i2c-oled> (acedido em 10/04/2024).