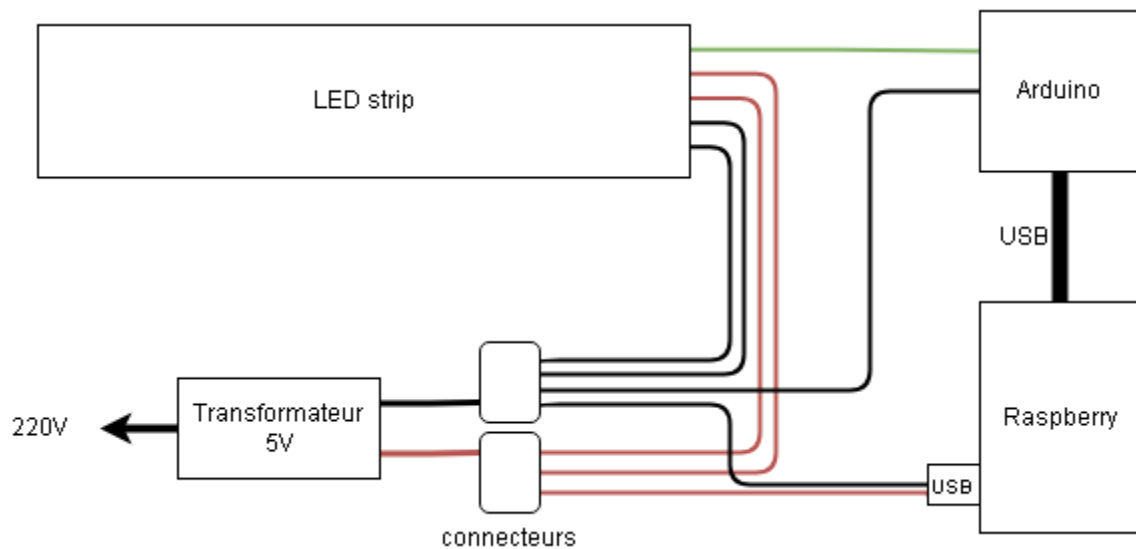
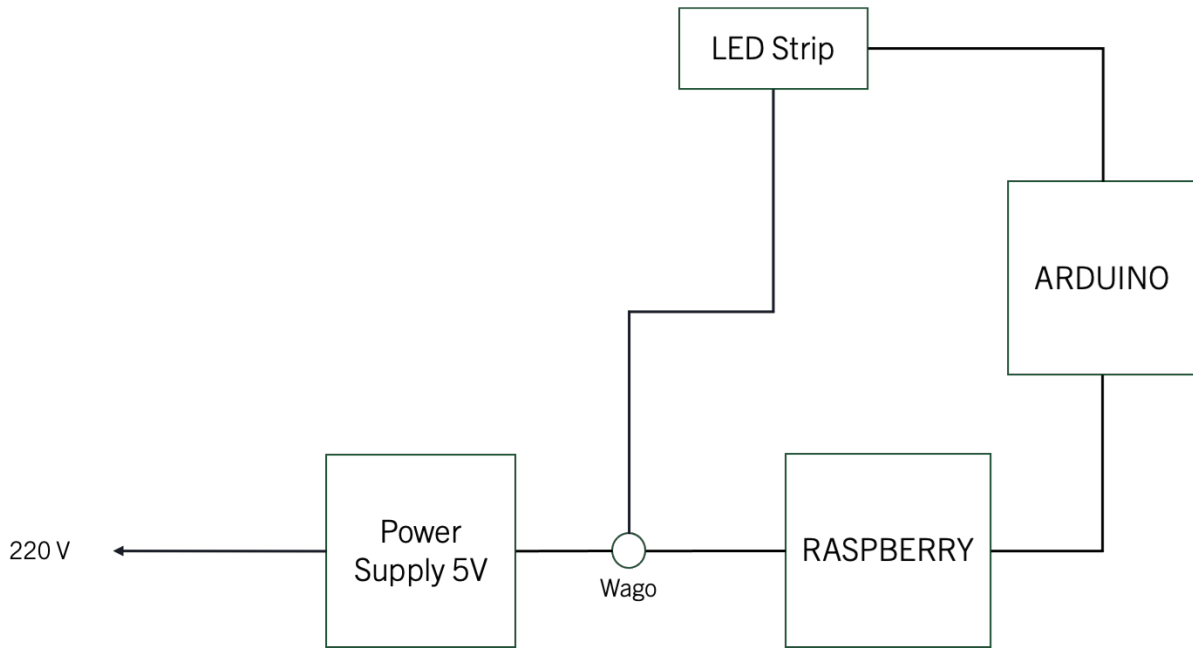


3.Connexion

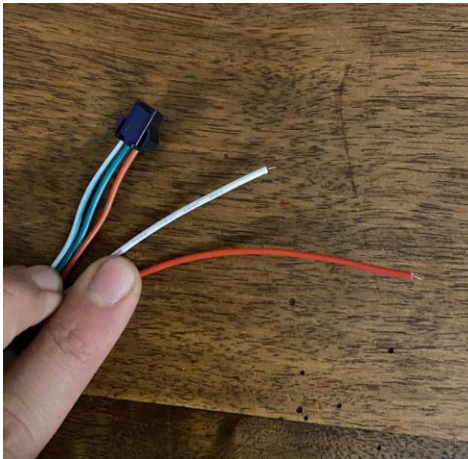
Schémas du circuit :



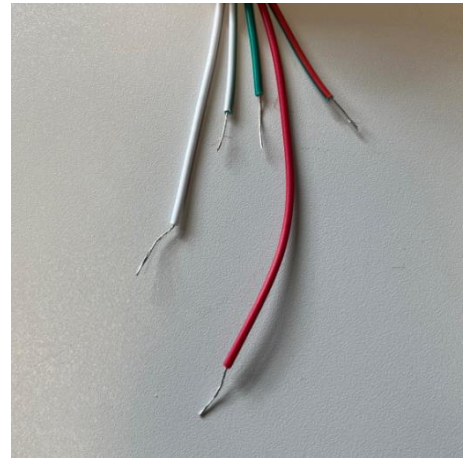
Préparation de la LED

1. Enlever le bout noir sur le fil de la guirlande et l'autre parti des fils de la guirlande les scotcher, nous ne les utiliserons pas.

AVANT



APRÈS



2. Regrouper les fils :
 - Les fils rouges ensemble qui correspondent au 5V
 - Les fils blancs ensemble qui correspondent à la masse
3. Placer les fils dans les wago:
 - 1 wago pour les fils rouges (5V) et ajouter le fils avec du scotch rouge de l'alimentation de la raspberry Pi dans une autre entrée
 - 1 wago pour les fils blancs (la masse) ajouter le fils avec du scotch blanc de l'alimentation de la raspberry Pi dans une autre entrée

PHOTO DE L'ETAPE

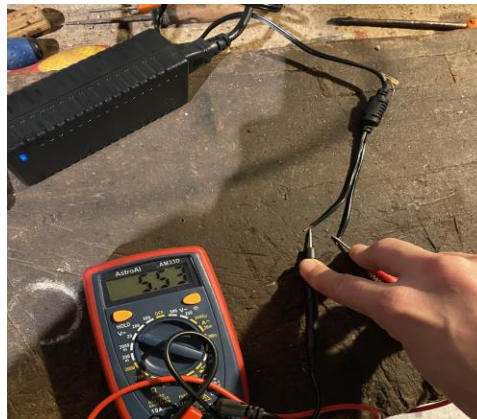
Présentation de l'alimentation 5v

1. Découper l'embout

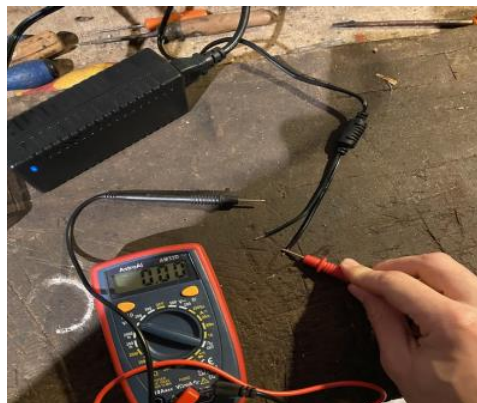


2. Dénuder les fils

- Déterminer l'utilité des fils
Avec un voltmètre déterminer le voltage de chaque fils
- Celui qui a du voltage, mettre un scotch rouge et l'ajouter au wago avec tous les fils rouges



- Celui qui n'a pas de voltage, mettre un scotch blanc et l'ajouter au wago avec tous les fils blancs

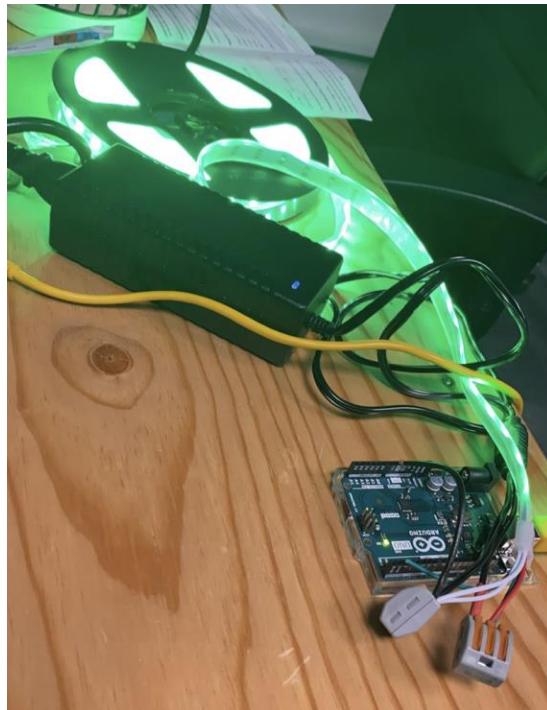


Connexion du matériel :

Arduino-LED

1. Ajout d'un fil électrique qui lie le wago où sont regroupés les masses et la masse de l'arduino (GND).
2. On utilise le fil vert restant de la guirlande que l'on branche sur le pin 2 de l'Arduino.

Ce que vous devez obtenir :



Raspberry-LED

1. On utilise le câble de l'Arduino utiliser pour téléverser le code
2. Brancher le câble de l'Arduino sur le Raspberry en utilisant le port USB.

Analyse des codes :

Code Python – Raspberry

Le dossier leds-strips à implanter dans la Raspberry. Explorons alors les documents présents.

- **FICHER MAIN.PY** : Il permet de tout lier et envoie la commande de localisation à l'Arduino.
- **FICHER CONFIG.JSON** : Ce fichier contient toutes les configurations nécessaires. C'est le seul fichier qui est à modifier.

```
{  
  
  "API_KEY": "UHWAFM-NWVP4R-5UXGQ5-4QIN",  
  "serial": {  
  
    "port": "/dev/arduino/leds-strip",  
    "baudrate": 9600  
  },  
  "location": {  
    "name": "Toulouse",  
    "latitude": "43.611660",  
    "longitude": "1.429760"  
  },  
  "criteria": {  
    "culmination": 15  
  },  
  "satellites": [  
    {  
      "name": "YAM-2",  
      "NORAD_ID": 48911,  
      "color": [245, 164, 66]  
    },  
    {  
      "name": "YAM-3",  
      "NORAD_ID": 48915,  
      "color": [66, 176, 245]  
    }  
  ]  
}
```

Explication du code:

« Location » : à remplacer par la position voulue.

« Culmination » : permet de paramétrer si on veut que la led s'allume que lorsque le satellite passe exactement au-dessus ou sinon qu'elle degré d'écart on accepte. Les 15°C sont calculés et déterminés comme ceci :



« NORAD_ID » : est à trouver sur internet selon le satellite que l'on veut observer et par conséquent changer aussi le nom.

- **FICHIER DOCKERFILE** : il s'agit du fichier nécessaire à la création de l'image.
- **FICHIER DOCKER-COMPOSE.YML** : il permet la création du conteneur.
- **FICHIER MAKEFILE** : Il contient les instructions pour les commandes du terminal et la commande pour lancer la création de l'image c'est make run.

Code Arduino – Arduino

- **FICHER LED-STRIPS** : il permet de commander l'éclairage de la led à la position rechercher.

```

led-strips
#include <avr/wdt.h>
#include <Console.h>
#include <FastLED.h>
#include <stdio.h>
#include <string.h>

#define NUM_LEDS 150
#define DATA_PIN 2
#define CLOCK_PIN 13

CRGB leds[NUM_LEDS];
String input_buffer;
String command[3];

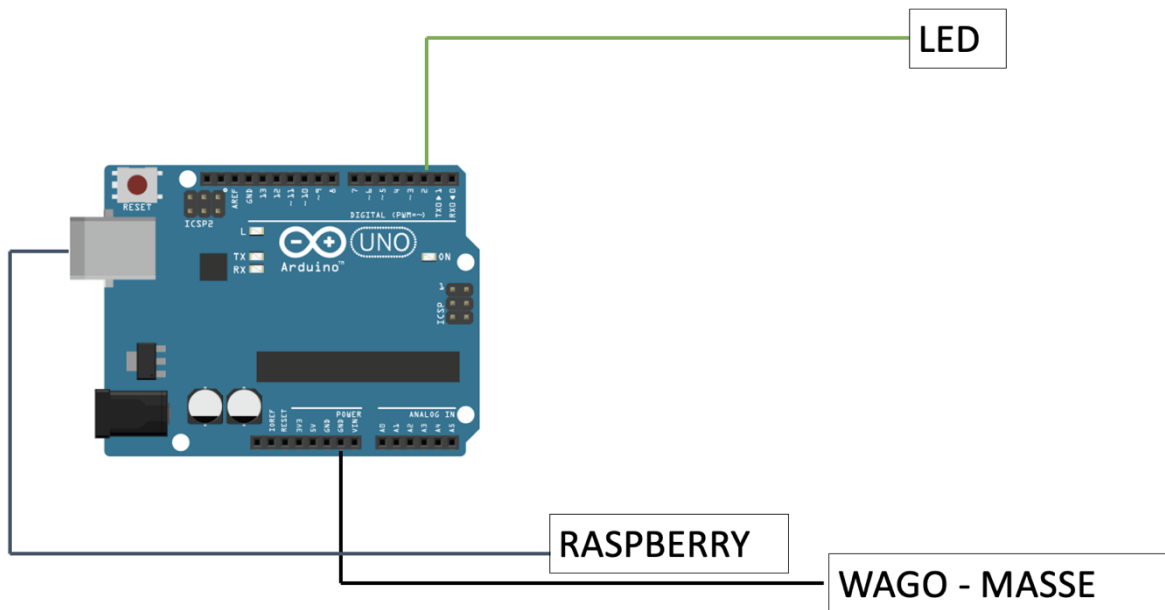
void setup()
{
  wdt_enable(WDTO_4S);
  Serial.begin(9600);
  FastLED.addLeds<WS2812B, DATA_PIN, RGB>(leds, NUM_LEDS);
  Serial.println("Welcome to LEDs STRIP v2.0");
  Serial.println(".. by Pierre");
  Serial.println("\nEnter command: ");
  wdt_reset();
}

void setAllColor(CRGB Color) {
  for(int i = 0; i < NUM_LEDS; i++) {
    leds[i] = Color;
  }
  FastLED.show();
}

int StringSplit(String sInput, char cDelim, String sParams[], int iMaxParams)
{
  int iParamCount = 0;
  int iPosDelim, iPosStart = 0;

```

Schema connection de l'arduino



Groundtrack API

Cette API demande les prochains passages d'un satellite donné au-dessus de vous, pour cela utilisez Skyfield pour prédire les passages, et l'API GP de Celestrak pour obtenir des données TLE mises à jour.

Les paramètres de cette API sont :

- lat : la latitude (obligatoire)
- lon : la longitude (obligatoire)
- limit : le nombre de prochains passages autorisés
- days : nombre de jours pour calculer les passages en avance
- visible_only c'est du binaire donc c'est soit true ou false, et filtre les passes par les passes visibles uniquement

Un exemple de code afin d'utiliser cette API :

```
GET /passes/25544?lat=-34.9112212&lon=-57.9372988&limit=1 HTTP/1.1 Accept:
*/*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: satellites.fly.dev
User-Agent: HTTPie/1.0.3
```

En gras sont les paramètres à rentrer, nos données.