PHY_52064_EP

StepMania duel

Léo & Espérance & Gabriel

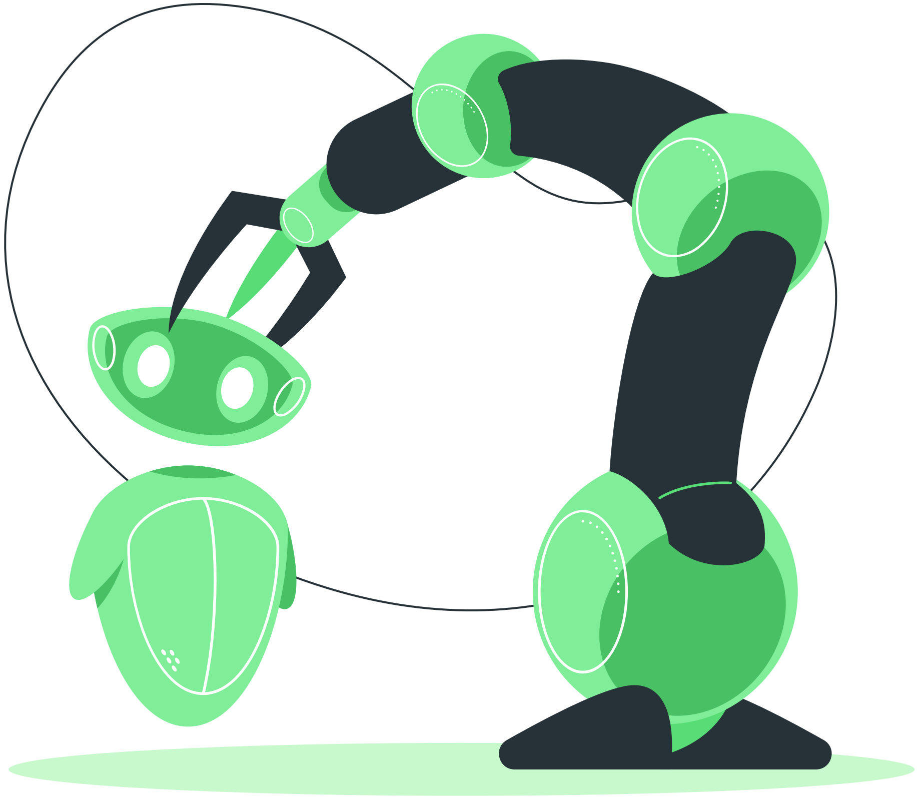


Table of Contents

1 Introduction

2 One player application

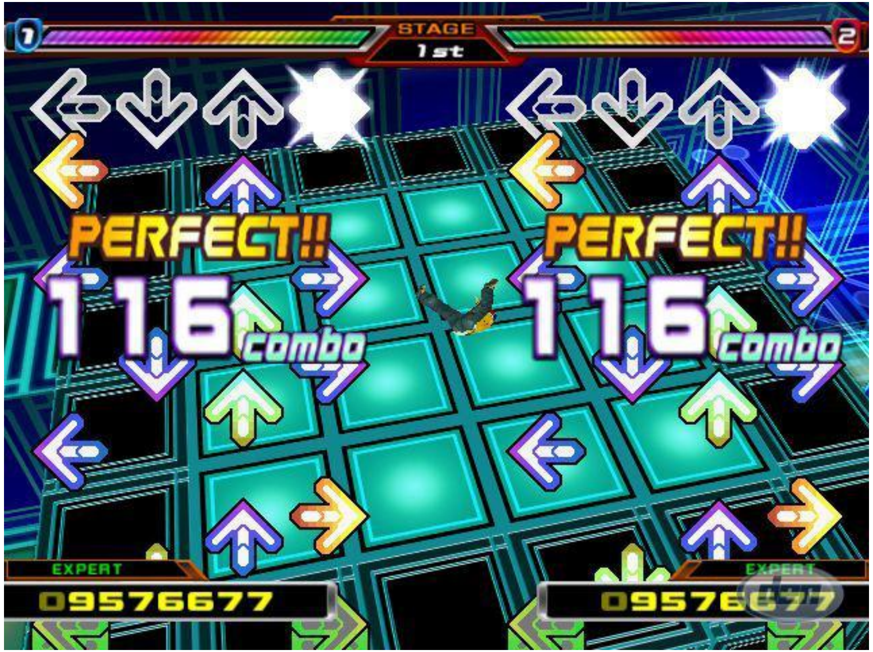
3 Extension for two players

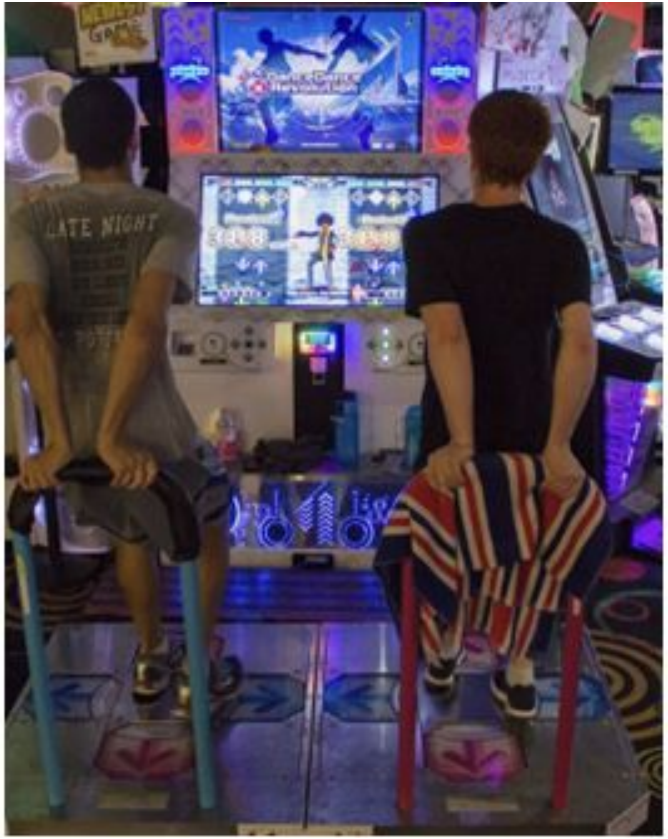
4 A few details about the code

Introduction

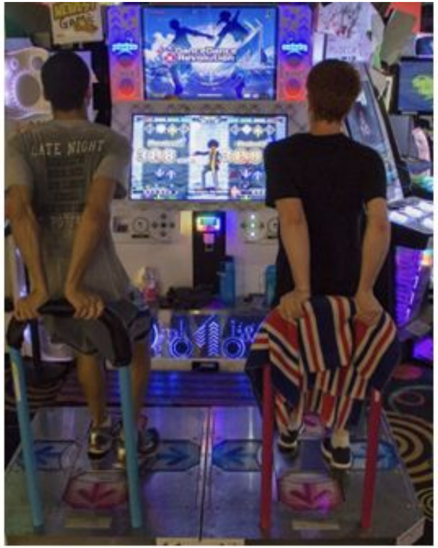
Project goals, materials,
application architecture

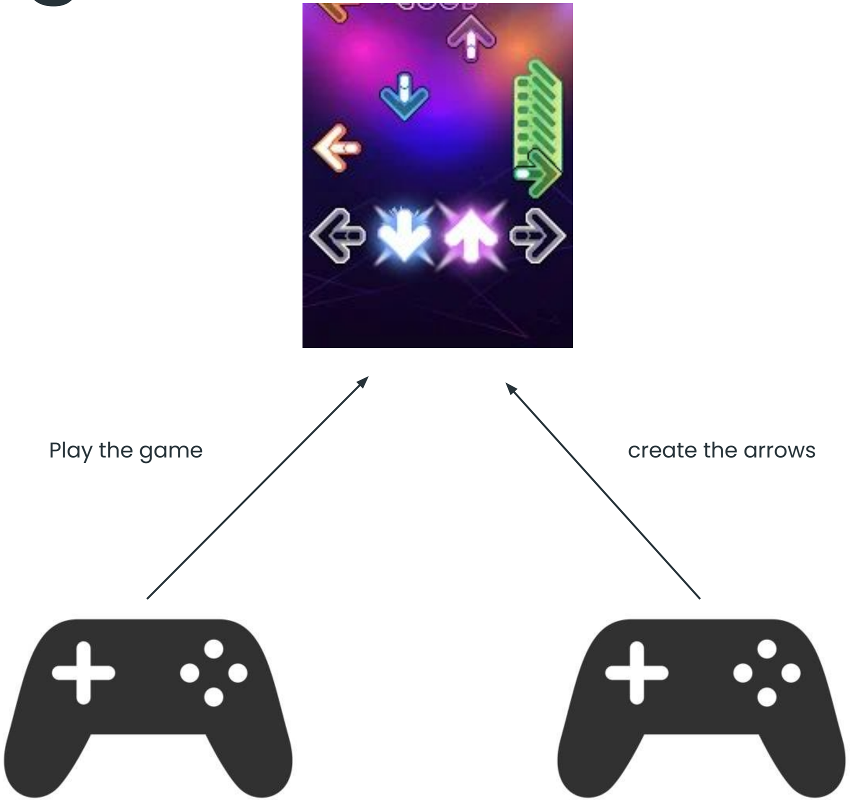
What is Stepmania ?





How about making a game





Play the game

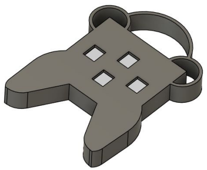
create the arrows

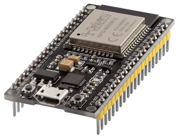
Player 1

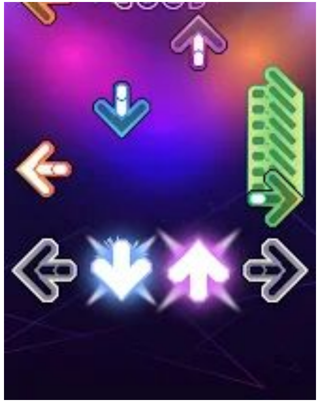
Player 2

Hardware

➔ Wireless controller using an **ESP32** microcontroller with **LED buttons** and a **3D case**





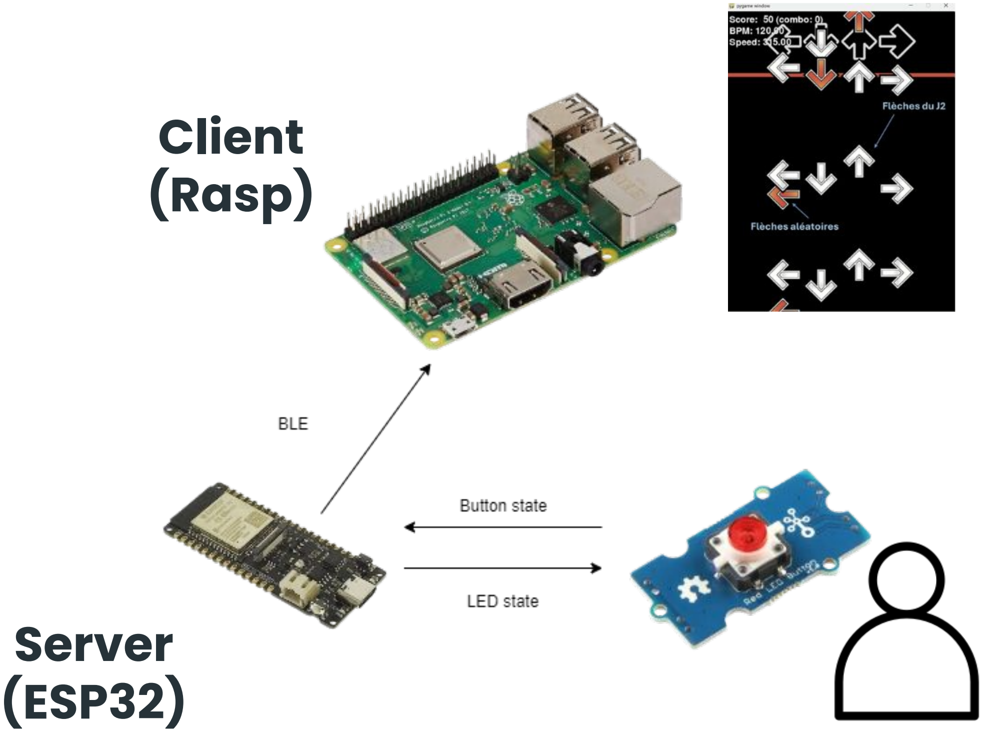


➔ pygame application running on a **Raspbery Pi**

One player application

Latency analysis of different protocols,
Wireless time synchronization,
interrupt based game events,
implementation details

Si latence < tolerance ?



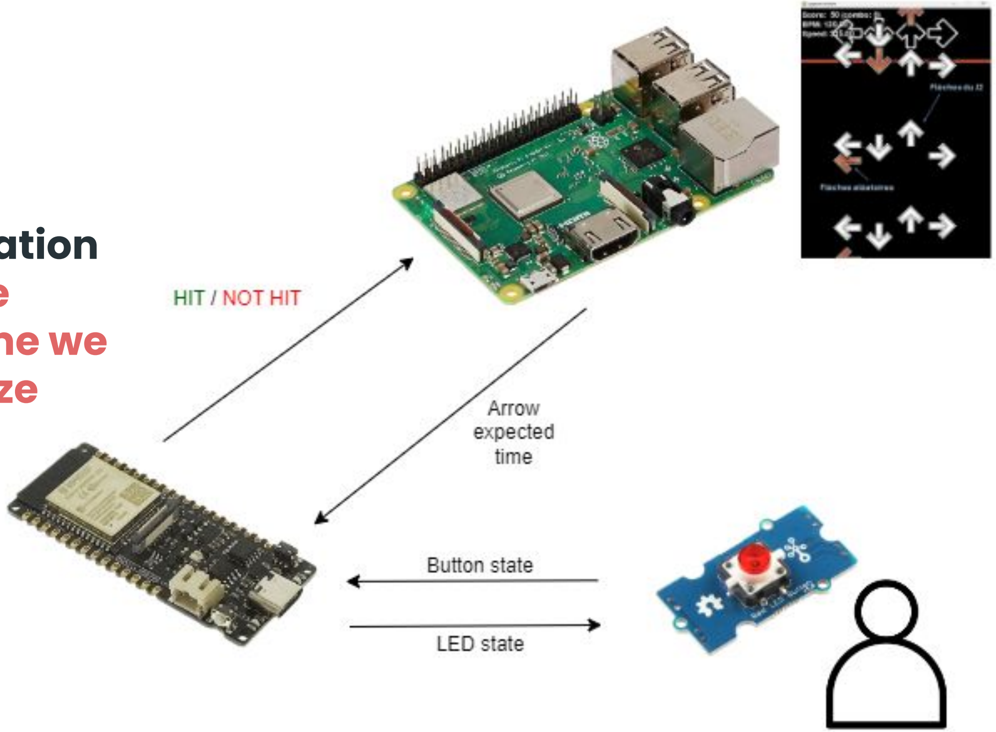
pygame window

Score: 50 (combo: 0)
BPM: 120.00
Speed: 315.00

Si latence › tolerance ?

Two-way communication

***for ESP to decide move time ≶ arrow time we need to synchronize clocks!**

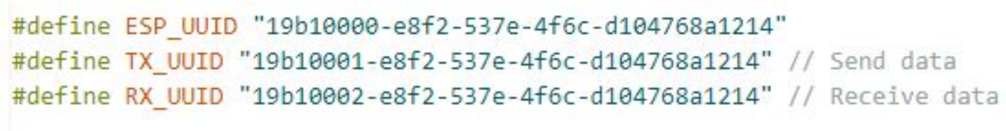


Communication protocols

	MQTT	Bluetooth Classic	Bluetooth Low Energy (**GATT notification**)
Power consumption		≅1W	≅0.01W - 0.5 W
Data rate		2Mbps - 3Mbps	500kbps - 1Mbps
Latency	≅150ms - 500ms	≅100ms	≅6ms

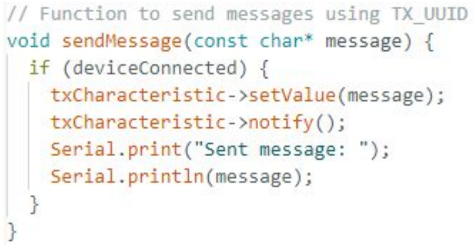
*Assez pour tourner le jeu à 120 fps!!

Implementation



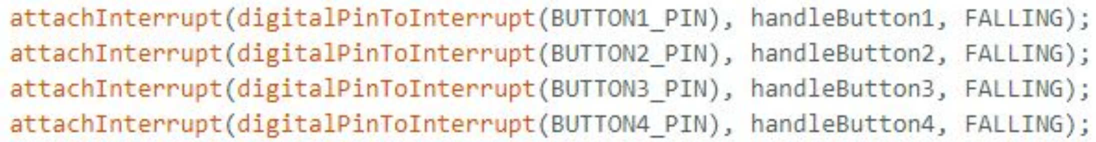
```
#define ESP_UUID "19b10000-e8f2-537e-4f6c-d104768a1214"
#define TX_UUID "19b10001-e8f2-537e-4f6c-d104768a1214" // Send data
#define RX_UUID "19b10002-e8f2-537e-4f6c-d104768a1214" // Receive data
```

➔ Configuration of the BLE Server



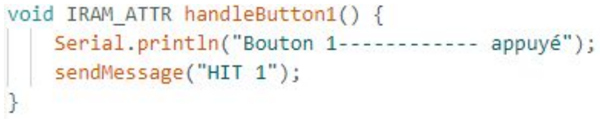
```
// Function to send messages using TX_UUID
void sendMessage(const char* message) {
  if (deviceConnected) {
    txCharacteristic->setValue(message);
    txCharacteristic->notify();
    Serial.print("Sent message: ");
    Serial.println(message);
  }
}
```

➔ Send a message



```
attachInterrupt(digitalPinToInterrupt(BUTTON1_PIN), handleButton1, FALLING);
attachInterrupt(digitalPinToInterrupt(BUTTON2_PIN), handleButton2, FALLING);
attachInterrupt(digitalPinToInterrupt(BUTTON3_PIN), handleButton3, FALLING);
attachInterrupt(digitalPinToInterrupt(BUTTON4_PIN), handleButton4, FALLING);
```

➔ management of interrupts associated with each button

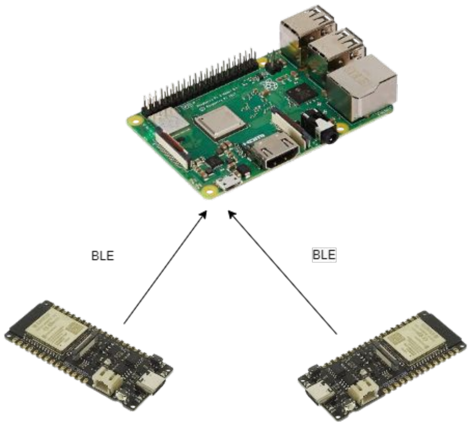


```
void IRAM_ATTR handleButton1() {
    Serial.println("Bouton 1----------- appuyé");
    sendMessage("HIT 1");
}
```

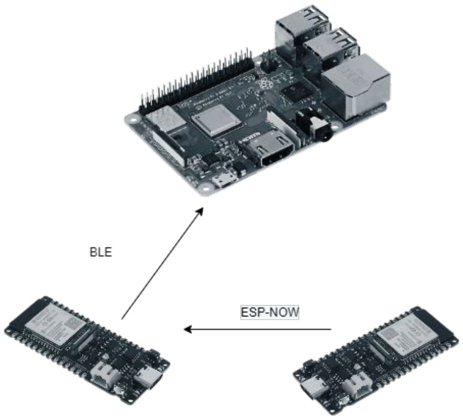
Extension: adding a second player

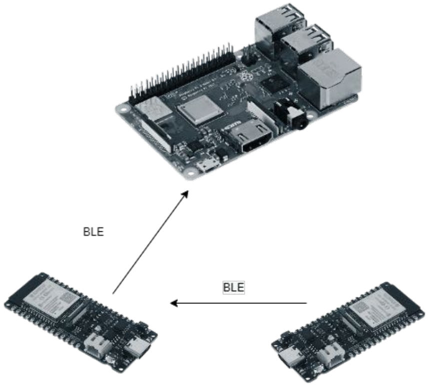
Different architectures and protocols tested, difficulties faced and axes of improvement

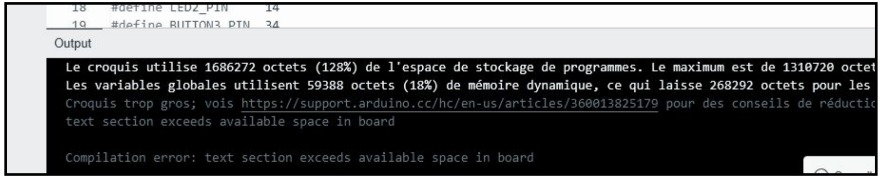
Propositions



- ➔ **Problem** : the Raspberry could not maintain both connections simultaneously
- ➔ Possible **reasons** :
  - ◆ awkward implementation on the ESP32
  - ◆ Implementation on Raspberry code and libraries





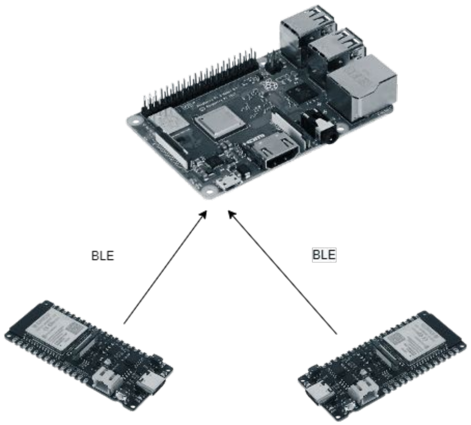


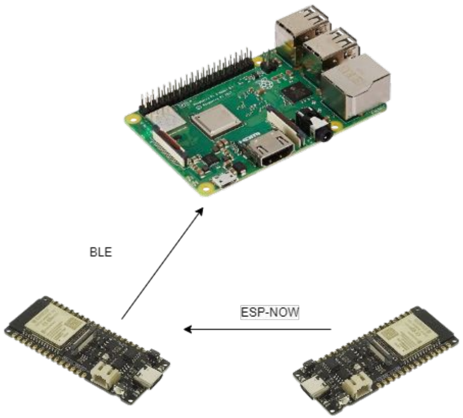
```
#define LED2_PIN     14
#define BUTTON3_PIN  34
```

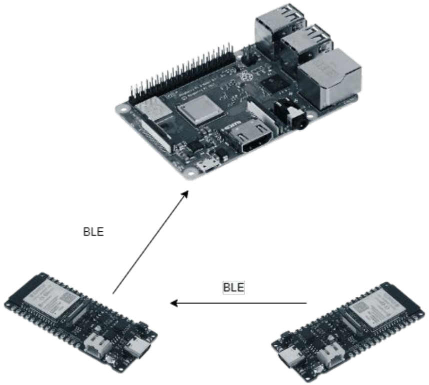
Output

```
Le croquis utilise 1686272 octets (128%) de l'espace de stockage de programmes. Le maximum est de 1310720 octet
Les variables globales utilisent 59388 octets (18%) de mémoire dynamique, ce qui laisse 268292 octets pour les
Croquis trop gros; vois https://support.arduino.cc/hc/en-us/articles/360013825179 pour des conseils de réductic
text section exceeds available space in board

Compilation error: text section exceeds available space in board
```

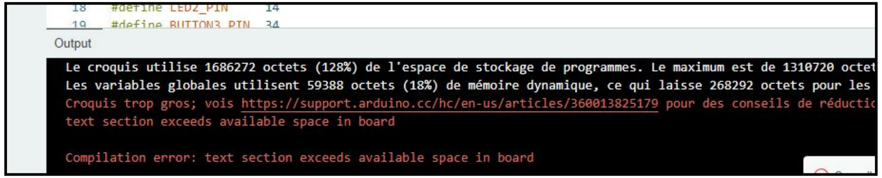
Propositions







- ➔ **ESP now** : Layer 2 wireless communication protocol designed and optimised for ESP 32
- ➔ **Problems** :
  - ◆ MAC address different from that of the BLE
  - ◆ Memory limit

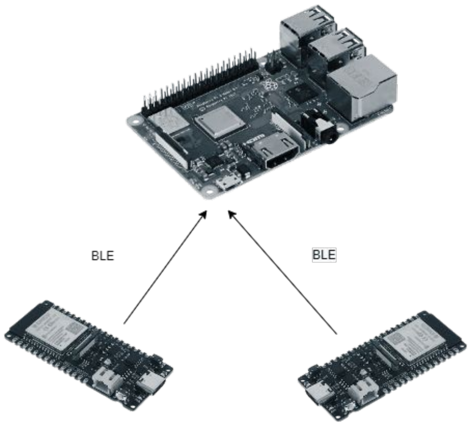


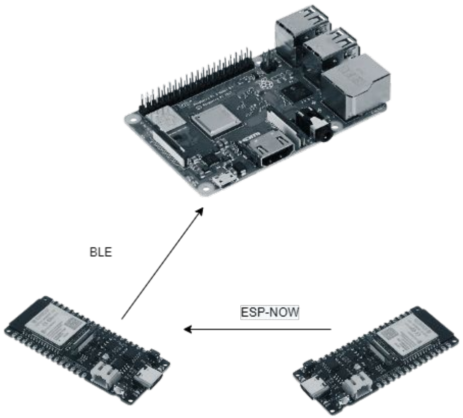
```
#define LED2_PIN     14
#define BUTTON3_PIN  34
```

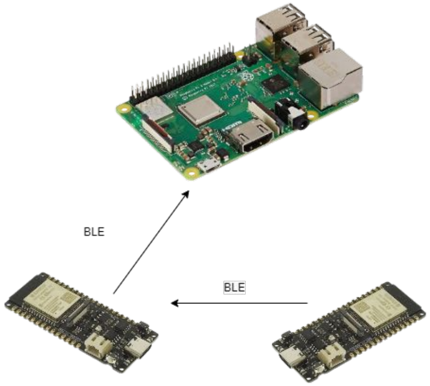
Output

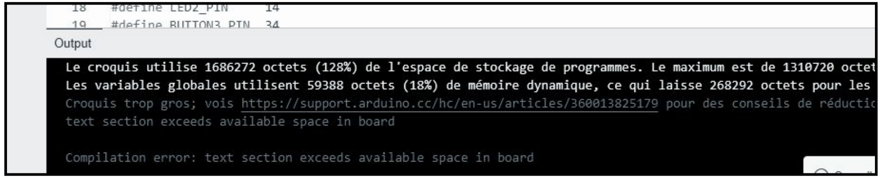
```
Le croquis utilise 1686272 octets (128%) de l'espace de stockage de programmes. Le maximum est de 1310720 octet
Les variables globales utilisent 59388 octets (18%) de mémoire dynamique, ce qui laisse 268292 octets pour les
Croquis trop gros; vois https://support.arduino.cc/hc/en-us/articles/360013825179 pour des conseils de réductic
text section exceeds available space in board

Compilation error: text section exceeds available space in board
```

Propositions









```
#define LED2_PIN     14
#define BUTTON3_PIN  34
```

```
Le croquis utilise 1686272 octets (128%) de l'espace de stockage de programmes. Le maximum est de 1310720 octet
Les variables globales utilisent 59388 octets (18%) de mémoire dynamique, ce qui laisse 268292 octets pour les
Croquis trop gros; vois https://support.arduino.cc/hc/en-us/articles/360013825179 pour des conseils de réductio
text section exceeds available space in board

Compilation error: text section exceeds available space in board
```

A few details about the software

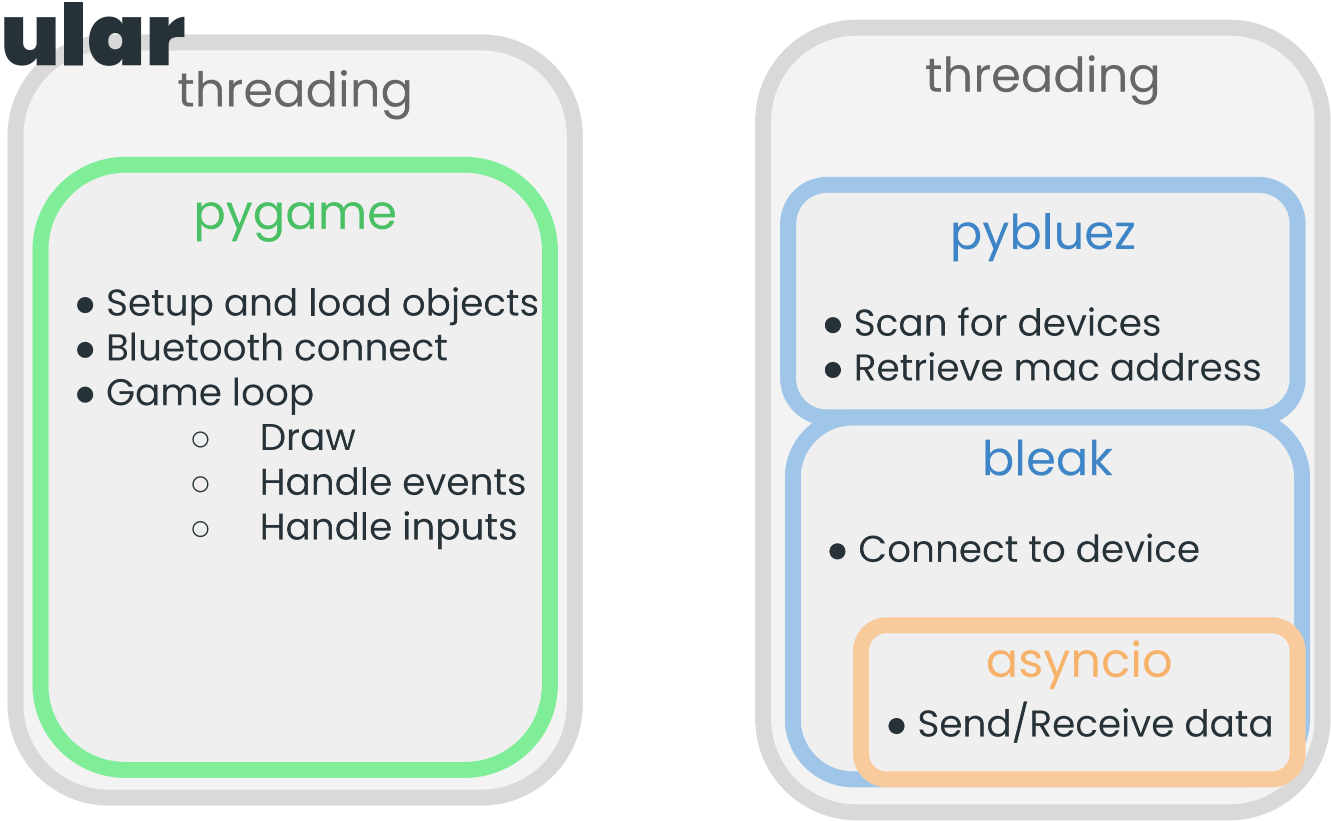
python3 and arduino specific implementations

A game engine

+ Powerful yet simple to use
+ “Event” handling
+ Ability to interact with audio, sprites,…

- Slow…

A bunch of libraries, modular



threading

pygame

- Setup and load objects
- Bluetooth connect
- Game loop
  - ○ Draw
  - ○ Handle events
  - ○ Handle inputs

threading

pybluez

- Scan for devices
- Retrieve mac address

bleak

- Connect to device

asyncio

- Send/Receive data

StepMania duel

It’s demo time!

Thank you for your attention!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**