

Séance 4.

Au cours de la 4eme séance, on a réglé les quelques petits bugs qu'on a trouvé sur le composant principale, puis intégrer un 2eme joystick.

L'intégration du deuxième joystick s'est avérée être un défi technique, mais après des heures de programmation et de tests rigoureux, nous sommes parvenus à établir une communication fluide entre la carte principale et la manette. Les coordonnées générées par le joystick sont désormais interprétées par le système, permettant ainsi un contrôle précis de l'orientation du canon dans le jeu.

On a utilisé le même code que pour la communication du 1^{er} joystick.

```
#include <SoftwareSerial.h>

#define POTENTIO_X A0
#define POTENTIO_Y A1

#define POTENTIO_X1 A2
#define POTENTIO_Y1 A3

struct Data
{
    uint8_t x1;
    uint8_t x2;
    uint8_t y1;
    uint8_t y2;
};

Data data;
Data lastData;

SoftwareSerial HC12(11, 10); // HC-12 TX Pin, HC-12 RX Pin

void setup() {
    Serial.begin(9600);
    HC12.begin(9600);
    data = (Data) {0, 0, 0, 0};
    lastData = (Data) {0, 0, 0, 0};
}

void loop(){
    data.x1 = (analogRead(POTENTIO_X)/1023.0)*255;
    data.x2 = (analogRead(POTENTIO_Y)/1023.0)*255;
    data.y1 = (analogRead(POTENTIO_X1)/1023.0)*255;
    data.y2 = (analogRead(POTENTIO_Y1)/1023.0)*255;
    if (memcmp(&data, &lastData, sizeof(Data))) {
        HC12.write((byte *)&data, sizeof(Data));
        lastData = data;
        Serial.println(data.y1);
        Serial.println(data.y2);
        delay(500);
    }
}
```