


AOP S4 – 2023-2024

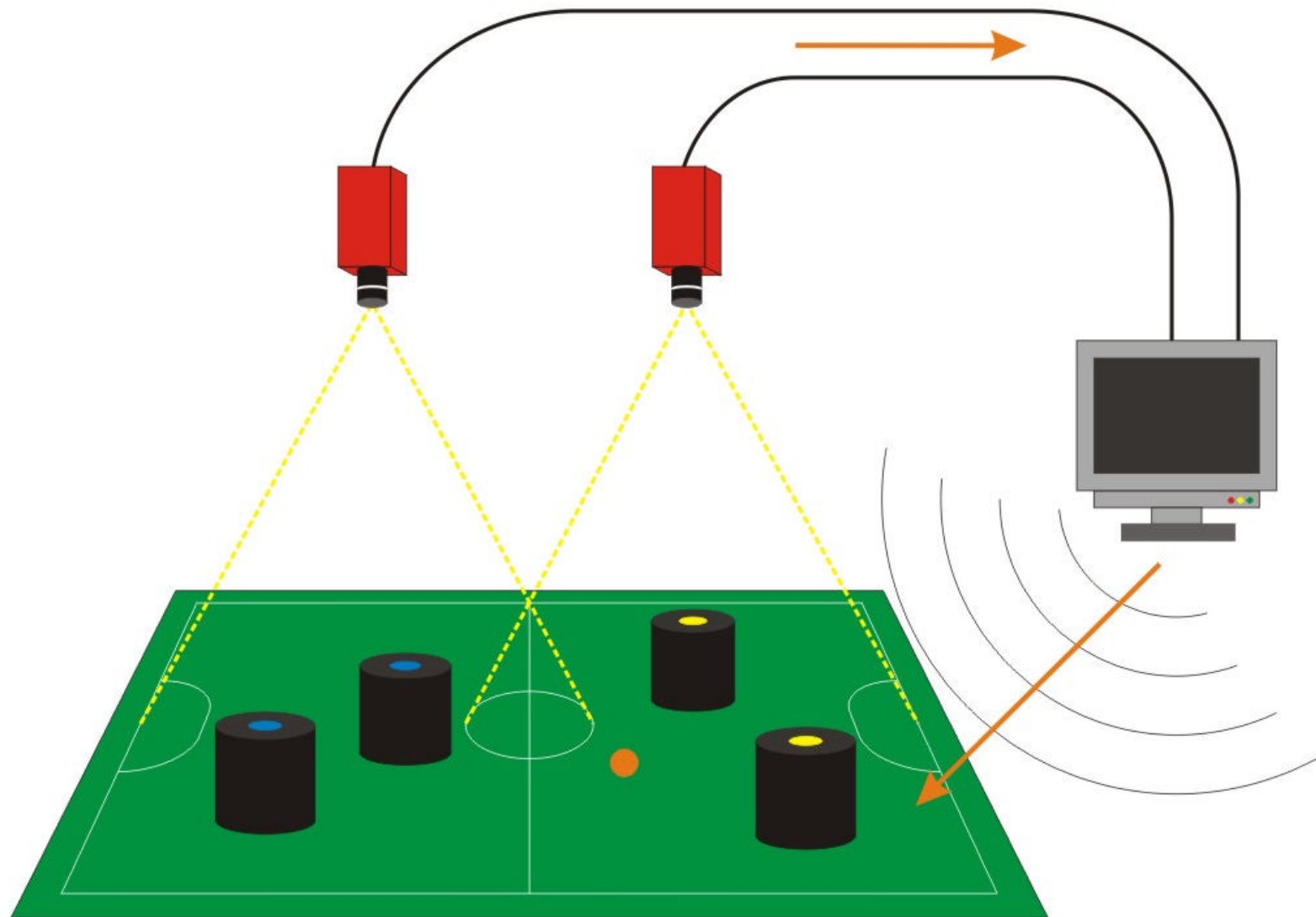


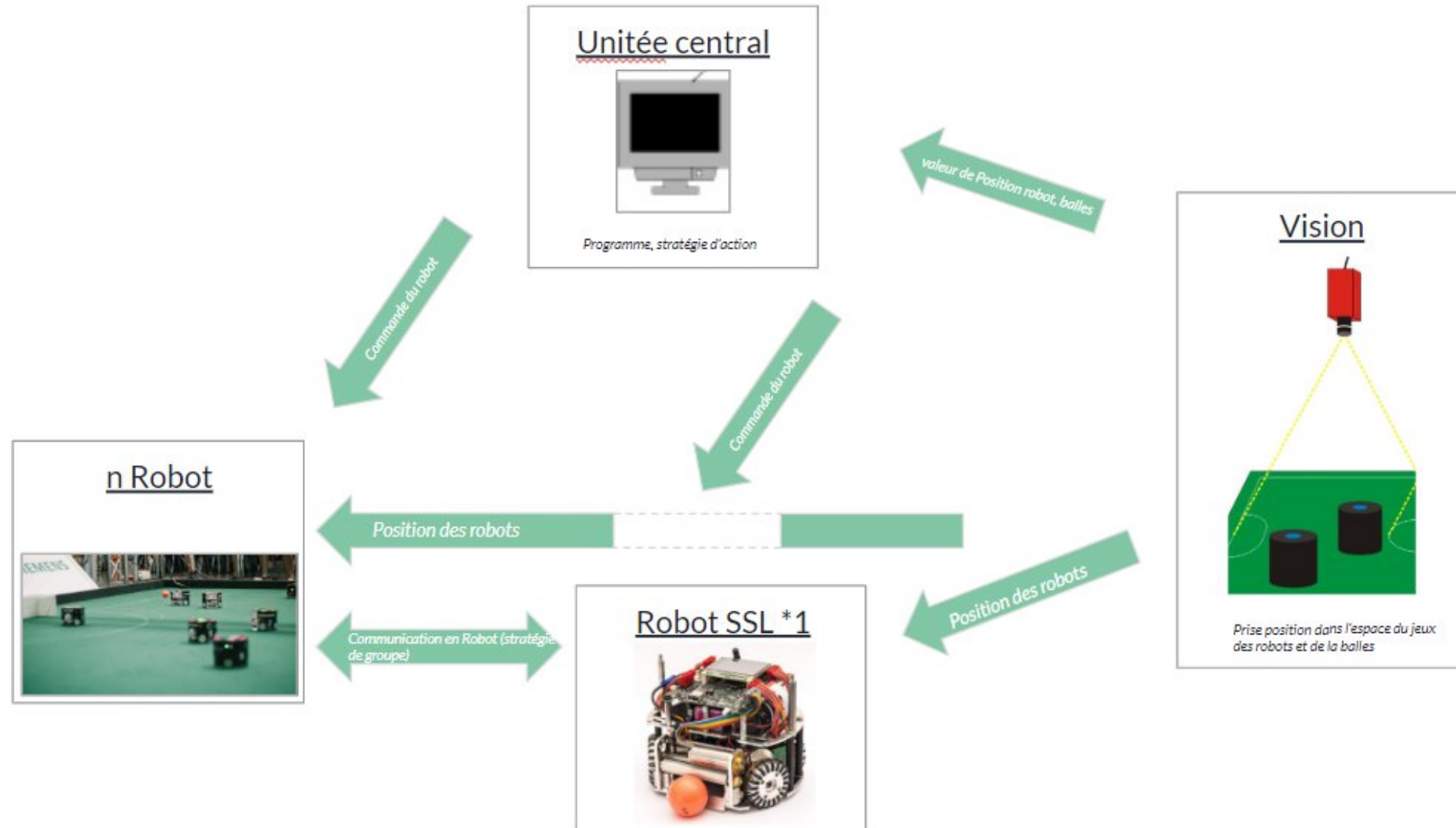
**geii**

**Génie électrique  
et informatique  
industrielle**

**iut**  
de **BORDEAUX**

- Contexte
  - Contexte 2
  - Module NRF24L01
  - Schemas elec
  - Code Arduino
  - Platine de test
- 



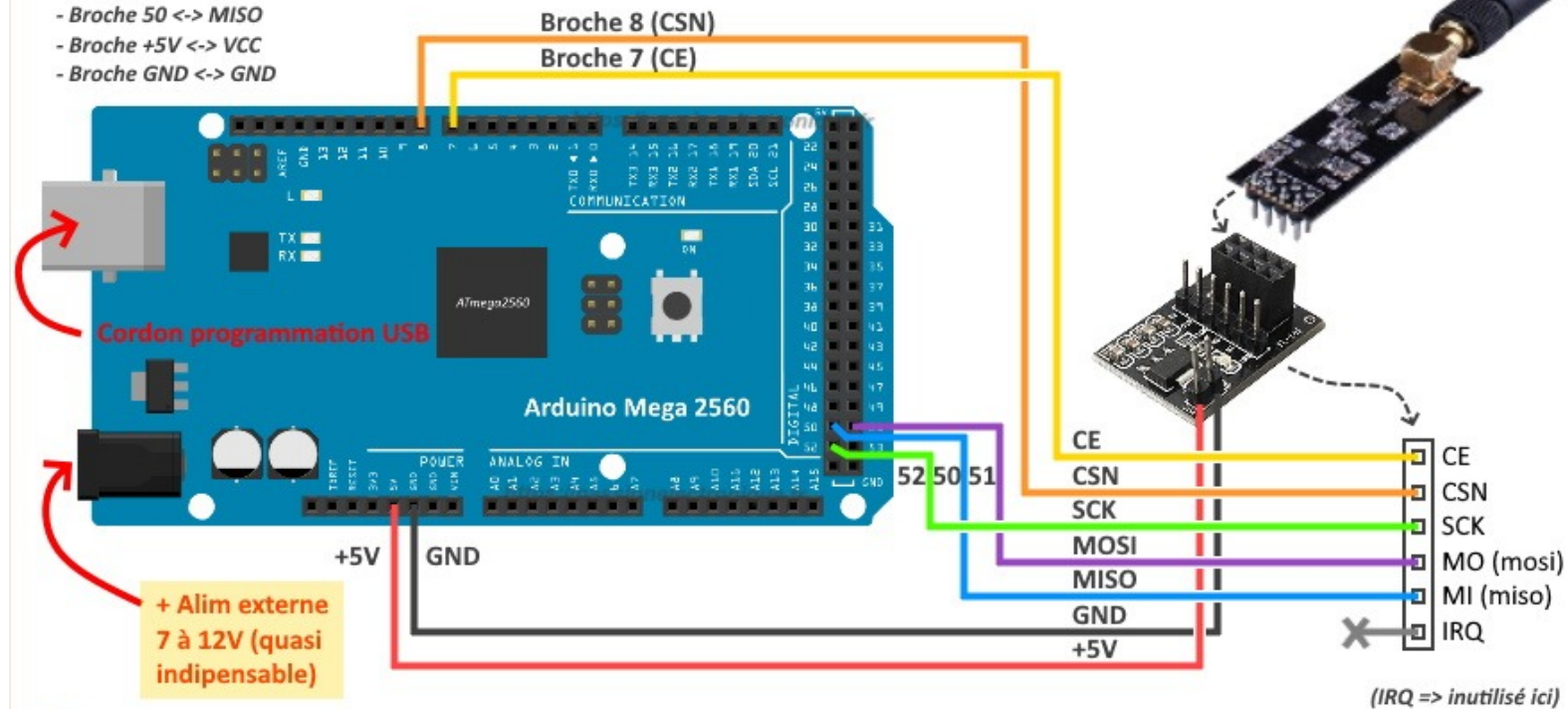


- Communication SPI
- Tension d'alimentation de 1.9V à 3.6V
- Emission à 2.4Ghz
- Consommation de 13.5mA



## Connections Arduino Méga <-> nRF24 PA LNA :

- Broche 7 <-> CE
- Broche 8 <-> CSN
- Broche 52 <-> SCK
- Broche 51 <-> MOSI
- Broche 50 <-> MISO
- Broche +5V <-> VCC
- Broche GND <-> GND



-Librairie NRF24  
-Arduino IDE  
-2 cartes arduino / 2 Module NRF24L

## Emetteur

Send.ino

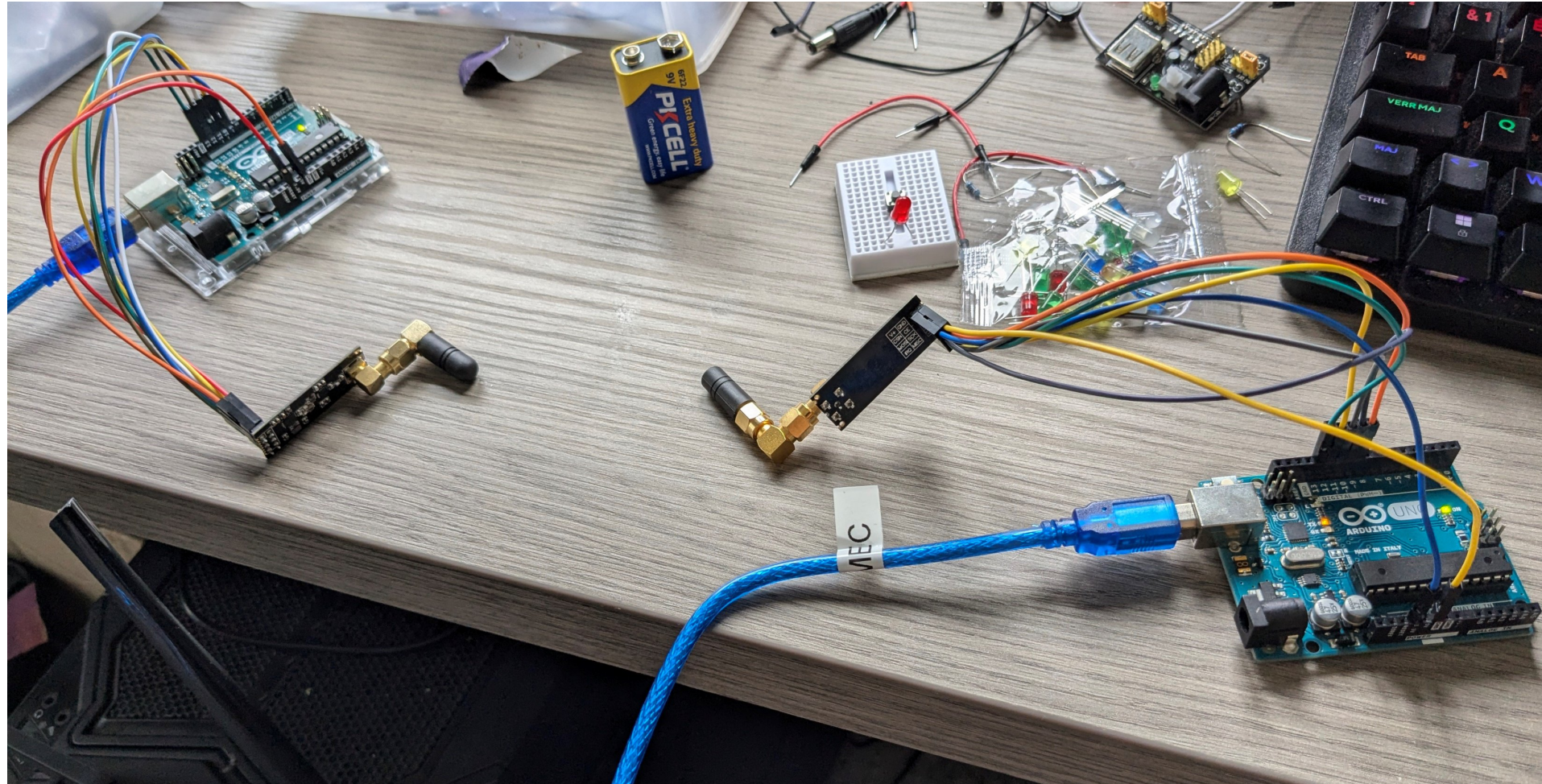
```
1 *recepteur*
2 #include <SPI.h>
3 #include <nRF24L01.h>
4 #include <RF24.h>
5
6 #define PIN_CE 9
7 #define PIN_CSN 10
8
9 RF24 radio(PIN_CE, PIN_CSN);
10 int dataToSend = 42;
11
12 void setup() {
13     radio.begin();
14     radio.openWritingPipe(0xF0F0F0E1LL); // ID unique pour l'émetteur
15 }
16
17 void loop() {
18     radio.write(&dataToSend, sizeof(dataToSend));
19     delay(1000);
20 }
```

## Réception

receive.ino

```
1 *recepteur*
2 #include <SPI.h>
3 #include <nRF24L01.h>
4 #include <RF24.h>
5
6 #define PIN_CE 9
7 #define PIN_CSN 10
8
9 RF24 radio(PIN_CE, PIN_CSN);
10 int receivedData;
11
12 void setup() {
13     radio.begin();
14     radio.openReadingPipe(1, 0xF0F0F0E1LL); // ID unique pour l'émetteur
15     radio.startListening();
16     Serial.begin(9600);
17 }
18
19 void loop() {
20     if (radio.available()) {
21         radio.read(&receivedData, sizeof(receivedData));
22         Serial.print("Données reçues: ");
23         Serial.println(receivedData);
24     }
25 }
```







# Conclusion