```
/*
                TELECOMMANDE.ino

  Brochage pour l'emetteur et le recpteur MODULE NRF24L04
  Broches utilisees pour le MEGA 2560
         MISO -> 50             * MOSI -> 51         * SCK -> 52
         CE -> 48               * CSN -> 49
         GND -> GND             * VCC -> 3
             (Volts du regulateur)

  Broches utilisees pour le UNO
         MISO -> 12           * MOSI -> 11           * SCK -> 13
         CE -> 8              * CSN -> 7
         GND -> GND           * VCC -> 3.3v
             (Volts du regulateur)
*/

// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::                                          LIBRAIRIE                                                       ::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// -------- LIBRAIRIE EMETTEUR RECEPTEUR RADIO --------
// MIRF PERMETTANT LE CONTROLE DU MODULE NRF24L04
#include <SPI.h> // Pour la gestion du port SPI
#include <Mirf.h> // Pour la gestion de la communication
#include <nRF24L01.h> // Pour les definitions des registres du nRF24L01
#include <MirfHardwareSpiDriver.h> // Pour la communication SPI hardware
// ::                                                                                                                    ::
// ::::::                                                                                                            ::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::




// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::                          DENOMINATION DES BROCHES DIGITAL/ANALOGIQUE                                     ::::::::::
```

```
// ::::::::::                                                                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// -------- RACCORDEMENT DES COMPOSANTS ET CIRCUITS ELECTRONIQUES AUX BROCHES DE L'ARDUINO --------

// -------- RACCORDEMENT DU JOYSTICK DE DROITE AUX BROCHE ANALOGIQUES. --------
int SENSOR_PIN_X_D = 0;
int SENSOR_PIN_Y_D = 2;
int SENSOR_PIN_D_Switch = 1;
// -------- RACCORDEMENT DU JOYSTICK DE GAUCHE AUX BROCHE ANALOGIQUES. --------
int SENSOR_PIN_X_G = 3;
int SENSOR_PIN_Y_G = 5;
int SENSOR_PIN_G_Switch = 4;

// -------- RACCORDEMENT DES LEDS AUX BROCHES NUMERIQUES. --------
const int LED_ROUGE = 2;
const int LED_VERTE = 3;
const int LED_BLEU = 4;

// -------- DECLARATION DES BROCHES POUR LE BOUTONS --------
const int Broche_bouton1 = 5; // bleu haut
const int Broche_bouton2 = 6;  // orange bas
const int Broche_bouton3 = 9; // vert gauche
const int Broche_bouton4 = 10;  // jaune droite
// ::                                                                                                                                            ::
// ::::::                                                                                                                                    ::::::
// ::::::::::                                                                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::




// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                                                  ::::::::::
// ::::::::::                                          DEFINITION DES STRUCTURES                                                               ::::::::::
// ::::::::::                                                                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ------- STRUCTURE POUR STOCKER LES VALEURS DU JOYSTICK -------
typedef struct {
  int X_G;
```

```
  int Y_G;
  int SW_G;
  int X_D;
  int Y_D;
  int SW_D;
  byte bouton_Haut;
  byte bouton_Bas;
  byte bouton_Gauche;
  byte bouton_Droite;
} MaStructure;
MaStructure Joystick_original; // VALEUR ANALOGIQUES DE 0 A 1023
MaStructure Joystick;  // VALEURS CONVERTI POUR LE CONTROLE PWM DES MOTEURS 0 A 255
byte taille_message = sizeof(MaStructure);  // 12 octet (valeurS de type int) + 4 octet (valeurS de type byte)
// ::                                                                                                                  ::
// ::::::                                                                                                          ::::::
// ::::::::::                                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::



// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                  ::::::::::
// ::::::::::                                    CREATION D'UN CHRONOMETRE                                     ::::::::::
// ::::::::::                                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ETAT D'ACTIVATION DE LA TEMPORISATION
//int tempoActive = 0;
// TEMPS A L'ACTIVATION DE LA TEMPORISATION
unsigned long tempoDepart = 0;
// INTERVAL DE TEMPS (milliseconds)
const long interval = 2000;
// ::                                                                                                                  ::
// ::::::                                                                                                          ::::::
// ::::::::::                                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::                                          PARAMETRAGE                                                     ::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ------- VARIATION VALEUR ENCADRANT LE POINT MORT -------
// JOYSTICK GAUCHE
int  Seuil_Mini_X_G;
int  Seuil_Maxi_X_G;
int  Seuil_Mini_Y_G;
int  Seuil_Maxi_Y_G;
// JOYSTICK DROIT
int Seuil_Mini_X_D;
int Seuil_Maxi_X_D;
int Seuil_Mini_Y_D;
int Seuil_Maxi_Y_D;
//
int Compteur = 0;
// ::                                                                                                                    ::
// ::::::                                                                                                            ::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```



```
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::                                    FONCTIONNEMENT DES JOYSTICKS                                          ::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

// POSITION JOYSTICK ENVOYE VIA LE JOYPAD ( j'ai inversé l'axes X et Y )
//                                                  (+)
//                                            (Axe des  X)
//                                                  0
//                                                  |
//                                                  |
```

```
//                                                  500  (Seuil_Mini_X)
//                                                   |
//                                  (Seuil_Maxi_Y)  (Point_Mort)   (Seuil_Mini_Y)
//(Axe des  Y)(-) 1023 ------------------550-------------511------------500---------------------- 0  (+)
//                                                   |
//                                                  550  (Seuil_Maxi_X)
//                                                   |
//                                                   |
//                                                  1023
//                                                  (-)
// ::                                                                                                ::
// ::::::                                                                                        ::::::
// ::::::::::                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                  ::::::::::
// ::::::::::                                  CONFIGURATION                                   ::::::::::
// ::::::::::                                                                                  ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
void setup() {
  Serial.begin(9600);
  // ------- RECUPERE LES VALEURS D'ENTREE DU JOYSTICK GAUCHE CONNECTE AUX BROCHES ANALOGIQUES -------
  Serial.println("CALIBRAGE DES JOYSTICKS");
  Serial.print("Joystick.X_G: ");
  Serial.print(analogRead(SENSOR_PIN_X_G));
  Serial.print(" , Joystick.Y_G: ");
  Serial.print(analogRead(SENSOR_PIN_Y_G));
  Serial.print(" , Joystick.SW_G: ");
  Serial.print(analogRead(SENSOR_PIN_G_Switch));
  Serial.print(" ,  Joystick.X_D: ");
  Serial.print( analogRead(SENSOR_PIN_X_D));
  Serial.print(" , Joystick.Y_D: ");
  Serial.print(analogRead(SENSOR_PIN_Y_D));
  Serial.print(" , Joystick.SW_D: ");
  Serial.println(analogRead(SENSOR_PIN_D_Switch));
  Serial.println();
```

```
// ------- MARGE D'ERREUR DES DEUX JOYSTICK AUTOUR DU RETOUR DE FORCE -------
int const value_seuil = 10;


// ------- AUTOCALIBRAGE DES JOYSTICK AVEC MARGE D'ERREUR REPRESENTE PAR LA VARIABLE "value_seuil" -------
// RECUPERE LES VALEURS DE POSITION DU JOYSTICK GAUCHE CONNECTE AUX BROCHES ANALOGIQUES
Seuil_Mini_X_G = analogRead(SENSOR_PIN_X_G) - value_seuil;
Seuil_Maxi_X_G = analogRead(SENSOR_PIN_X_G) + value_seuil;
Seuil_Mini_Y_G = analogRead(SENSOR_PIN_Y_G) - value_seuil;
Seuil_Maxi_Y_G = analogRead(SENSOR_PIN_Y_G) + value_seuil;
// RECUPERE LES VALEURS DE POSITION DU JOYSTICK DROIT CONNECTE AUX BROCHES ANALOGIQUES
Seuil_Mini_X_D = analogRead(SENSOR_PIN_X_D) - value_seuil;
Seuil_Maxi_X_D = analogRead(SENSOR_PIN_X_D) + value_seuil;
Seuil_Mini_Y_D = analogRead(SENSOR_PIN_Y_D) - value_seuil;
Seuil_Maxi_Y_D = analogRead(SENSOR_PIN_Y_D) + value_seuil;

// ------- ACTIVE LES BROCHES DES BOUTONS -------
pinMode(Broche_bouton4, INPUT);
pinMode(Broche_bouton3, INPUT);
pinMode(Broche_bouton2, INPUT);
pinMode(Broche_bouton1, INPUT);

// ------- ACTIVE LES BROCHES DES LED -------
pinMode(LED_ROUGE, OUTPUT);
pinMode(LED_VERTE, OUTPUT);
pinMode(LED_BLEU, OUTPUT);
// CLIGNOTEMENT DES LEDS MODE TEST
Serial.println("Controle des leds");
for (int i = 0; i < 5; i++) {
  digitalWrite(LED_ROUGE, HIGH);
  delay(50);
  digitalWrite(LED_ROUGE, LOW);
  delay(50);

  digitalWrite(LED_VERTE, HIGH);
  delay(50);
  digitalWrite(LED_VERTE, LOW);
  delay(50);
```

```
    digitalWrite(LED_BLEU, HIGH);
    delay(50);
    digitalWrite(LED_BLEU, LOW);
  }

  // -------- CONFIGURATION DE L'EMETTEUR RECEPTEUR RADIO --------
  /* Broches utilisees pour le UNO
          MISO -> 12           * MOSI -> 11           * SCK -> 13
          CE -> 8              * CSN -> 7
          GND -> GND           * VCC -> 3.3v
  */
  // Configuration des broches CSN et CE :
  Mirf.cePin = 8; // CE
  Mirf.csnPin = 7; // CSN
  // configuration du SPI : utiliser le port SPI hardware
  Mirf.spi = &MirfHardwareSpi; // utilisation du port SPI hardware
  Mirf.init(); // initialise le module SPI
  // canal et longueur utile doivent etre identiques
  // pour le client et le serveur
  Mirf.channel = 12;
  Mirf.payload = 16; // taille utile de la donnee transmise: 12 octet (valeurs de type int) + 4 octet (valeurs de type byte)
  // Configuration des adresses de reception et d'emission
  Mirf.setRADDR((byte *)"clie1"); // adresse de reception du module (de 5 octets)
  Mirf.setTADDR((byte *)"serv1"); // adresse vers laquelle on transmet (de 5 octets)
  Mirf.config(); // ecriture de la configuration

  // -------- AFFICHE LE MESSAGE DANS LE MONITEUR SERIE --------
  Serial.print("JOYPAD: ");
  Serial.print("initialisation OK, ");
  Serial.println("Envoi des donnees");
}
// ::                                                                                                                                                  ::
// ::::::                                                                                                                                            ::::::
// ::::::::::                                                                                                                                      ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::                                    DEMARRGE DU PROGRAMME                                                 ::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
void loop() {
  // -------- ENVOI LES VALEUR RELATIVE A LA POSITION DES JOYSTICK ET BOUTONS --------
  envoi_Position_Joystick();

  // SI ON RECOIT UN MESSAGE RETOUR
  byte paquet2[32];
  if (Mirf.dataReady()) {
    Mirf.getData((byte *)&paquet2); // Réception du paquet
    Serial.print("On a recu : ");
    for (int i = 0 ; i < Mirf.payload ; i++) {
      Serial.write(paquet2[i]); // write affiche le caractere
    }
    Serial.println(" ");
  }
  delay(100);
}
// ::                                                                                                                  ::
// ::::::                                                                                                          ::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::




// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::                              EMISSION DES DONNEES VIA L'EMETTEUR RADIO                                   ::::::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
void envoi_Position_Joystick(void) {
  // -------- RECUPERE LES VALEURS D'ENTREE DES JOYSTICK CONNECTE AUX BROCHES ANALOGIQUES --------
  Joystick_original.X_G =  analogRead(SENSOR_PIN_X_G);
  Joystick_original.Y_G =  analogRead(SENSOR_PIN_Y_G);
```

```
  Joystick_original.SW_G =  analogRead(SENSOR_PIN_G_Switch);

  Joystick_original.X_D =  analogRead(SENSOR_PIN_X_D);
  Joystick_original.Y_D =  analogRead(SENSOR_PIN_Y_D);
  Joystick_original.SW_D =  analogRead(SENSOR_PIN_D_Switch);

  // -------- CONVERTI LES VALEURS X ET Y (0 A 1023) DES JOYSTICKS EN VALEUR COMPRISE ENTRE 0 ET 255 --------
  // DONNEES DIRECTEMENT EXPLOITABLE PAR LE SYSTEME ROBOTISE POUR LA VARIATION DE VITESSE DES MOTEURS DC
  Joystick.X_G = convert_value_joypad(Joystick_original.X_G, Seuil_Mini_X_G, Seuil_Maxi_X_G);
  Joystick.Y_G = convert_value_joypad(Joystick_original.Y_G, Seuil_Mini_Y_G, Seuil_Maxi_Y_G);
  Joystick.X_D = convert_value_joypad(Joystick_original.X_D, Seuil_Mini_X_D, Seuil_Maxi_X_D);
  Joystick.Y_D = convert_value_joypad(Joystick_original.Y_D, Seuil_Mini_Y_D, Seuil_Maxi_Y_D);

  // -------- LIT L'ETAT DES BOUTONS --------
  Joystick.bouton_Haut = digitalRead(Broche_bouton1);  // bleu haut
  Joystick.bouton_Bas = digitalRead(Broche_bouton2);  // orange bas
  Joystick.bouton_Gauche = digitalRead(Broche_bouton3);  // vert gauche
  Joystick.bouton_Droite = digitalRead(Broche_bouton4);  // jaune droite


  // -------- ENVOIE DES DONNEES AVEC LA LIBRAIRIE MIRF VIA EMETTEUR NRF24L01 --------
  Mirf.send((byte *) &Joystick);
  // On boucle tant que le message n'a pas ete envoye
  while (Mirf.isSending()) {
    // on attend que le message soit envoye
  }
  // -------- AFFICHE LES DONNEES DANS LE MONITEUR SERIE --------
  Serial.print("Message envoye: ");
  Serial.print("Joystick.X_G: ");
  Serial.print(Joystick.X_G);
  Serial.print(" , Joystick.Y_G: ");
  Serial.print(Joystick.Y_G);
  Serial.print(" , Joystick.SW_G: ");
  Serial.print(Joystick.SW_G);
  Serial.print(" ,  Joystick.X_D: ");
  Serial.print( Joystick.X_D);
  Serial.print(" , Joystick.Y_D: ");
  Serial.print(Joystick.Y_D);
  Serial.print(" , Joystick.SW_D: ");
```

```
  Serial.print(Joystick.SW_D);
  Serial.print(" , Bouton haut: ");
  Serial.print(Joystick.bouton_Haut);
  Serial.print(" , Bouton bas: ");
  Serial.print(Joystick.bouton_Bas);
  Serial.print(" , Bouton gauche: ");
  Serial.print(Joystick.bouton_Gauche);
  Serial.print(" , Bouton droite: ");
  Serial.println(Joystick.bouton_Droite);
}
// ::                                                                                                                  ::
// ::::::                                                                                                            ::::::
// ::::::::::                                                                                                      ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
// ::::::::::                                                                                                      ::::::::::
// ::::::::::                                  CONVERTI LES VALEURS DE JOYSTICKS                                   ::::::::::
// ::::::::::        valeurs broches analogiques(0 à 1023) convertis en valeur digital PWM (0 et 255)              ::::::::::
// ::::::::::                                                                                                      ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
int convert_value_joypad(int position_Joystick, int seuil_mini, int seuil_maxi) {
  int position_joy ;

  // -------- CALCUL DE LA POSITION DU JOYSTICK SI INFERIEUR AU POINT MORT OU SEUIL MINIMUM --------
  if (position_Joystick < seuil_mini) {
    position_joy = -(position_Joystick - seuil_mini) / 2;
    if (position_joy > 255) {
      position_joy = 255;
    }
    return position_joy;
  }

  // -------- CALCUL DE LA POSITION DU JOYSTICK SI SUPERIEUR AU POINT MORT OU SEUIL MAXIMUM --------
  else if (position_Joystick > seuil_maxi) {
    int position_joy = (position_Joystick - seuil_maxi) / 2;
```

```
    if (position_joy > 255) {
      position_joy = 255;
    }
    return -position_joy;
  }
  else {
    position_joy = 0;
    return position_joy;
  }
}
// ::                                                                                                                          ::
// ::::::                                                                                                                  ::::::
// ::::::::::                                                                                                          ::::::::::
// ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```