2025/11/30 07:41 1/7 Portail arduino

Portail arduino

tutoriel-diy-portail-motorise-miniature-arduino

Code arduino

portail001.ino

```
* Ce programme controle le fonctionnement d'un petit protail domotisé.
* Si le badge présenté devant le lecteur RFID est bon, alors le
portail s'ouvre.
* Il se referme apres le passage de la voiture (détécté par le capteur
ultrason) ou bien apres 10 secondes d'inactivité.
 * Même scénario si on appuie sur un bouton de la télécommande
infrarouge. Si on souhaite sotir de l'intérieur,
* le capteur ultrason le détecte et ouvre le portail puis le referme 3
secondes après.
 * Le coulissement de la porte est assuré par un servomoteur à rotation
continue.
 * Des interrupteurs de fin de course permettent de stoper le
coulissement du portail.
* Une led est allumée quand le portail est en mouvement puis clignote
quand celui ci est ouvert.
* Enfin un lcd piloté en i2c donne des informations sur l'ouverture et
la fermeture du portail ainsi que sur la validité du badge.
 */
#include <Wire.h> //librairiepour la communication i2c
#include <LiquidCrystal I2C.h> //librairie pour utliser un écran lcd
avec un module i2c
LiquidCrystal I2C lcd(0x27,16,2); //spécification de l'adrese du
module
#include <IRremote.h> //librairie pour la communication infrarouge
#define pin recepteur infra 10 //variable contenant le numéro du pin
ou est coonnecté le recepteur infrarouge
IRrecv monRecepteur infra(pin recepteur infra); //création d'un nom
pour le recepteur connecté au pin 8
decode results message recu; //variable contenant le message recu
par le recepteur infrarouge
#include <Servo.h> //on inclut une librairie pour utiliser le
servomoteur
Servo monServo; //on déclare l'utilisation d'un servomoteur nommé
"monServo"
```

```
#include <SPI.h>
                    //librairie pour la communication SPI entre
l'arduino et le module RFID
#include <RFID.h> //librairie pour utiliser le module RFID
RFID monModuleRFID(9,8); //déclaration des broches de connection du
module RFID
int UID[5]; //tableau pour stocker le numéro d'identification lue par
le lecteur RFID
int badge BLEU[5] = {54,112,133,24,219};  //tableau contenant le
numéro d'identification de mon badge bleu
byte badge lu = 0; //pour savoir si un badge a été lu
byte ouverture porte = 0; //cette variable indique si on peut ou non
ouvrir le portail
unsigned long fermeture defaut = 0; //pour fermer le portail si ila
été ouvert et que aucune voiture ne passeau bout de 3 seconde
#define bouton fin 7 //pin ou est connecté le bouton poussoir de fin
de course quand le portail est fermé
#define bouton debut 4 //pin ou est connecté le bouton poussoir de
debut de course quand le portail est fermé
#define pin servo 3 //pin sur lequel est connecté le servomoteur qui
actionne le portail.
#define pin_ledV 6  //la led verte qui indique que le badge est bon
#define pin_ledR 5  //la led rouge qui indique que le badge est non
valide
#define pin LED portail 2 //led qui cliquote quand le portail est
ouvert
//capteur ultrason
#define pin TRIGGER 12
#define pin ECHO 11
byte E accent[8] = //création d'un tableau contenant le caractère
spécial 'é'
  B00001,
  B00110.
  B00000,
  B01110,
  B10001,
  B11111.
  B10000,
  B01110
};
void setup()
  Serial.begin(9600);
  //portail
  monServo.attach(pin_servo); //on déclare la broche de connection du
servo(digitale 11 PWM)
```

```
monServo.write(98); //onmet le servomoteur en arrêt
  pinMode(bouton_debut, INPUT); //le bouton de debut de course est
configuré en entrée
  pinMode(bouton fin, INPUT); //le bouton de fin de course est
configuré en entrée
  pinMode(pin LED portail, OUTPUT);
 //Module RFID
  SPI.begin(); //on initialise la communication SPI vers lemodule
RFID
  monModuleRFID.init(); //on initialise le module RFID
  pinMode(pin ledV, OUTPUT);
  pinMode(pin ledR, OUTPUT);
 //recepteur infrarouge
  monRecepteur_infra.enableIRIn(); //commande pour activer le module
infrarouge
  monRecepteur infra.blink13(true); //active une led lors de la
recepteion des données
  //ultrason
  pinMode(pin TRIGGER, OUTPUT); //on met le pin trigger en sortie
  pinMode(pin ECHO, INPUT); //on met le pin echo en entré
 //lcd
 Wire.begin(); //initialisation de la communication i2c
 lcd.init(); //initialisation du module lcd
 lcd.backlight(); //activation du rétroéclairage de l'écran
 lcd.createChar(1,E accent); //création d'un caractère spécial pour
faire un e accent
void loop()
  //affichage de la phrase : "Accés vérouillé"(avec l'insértion du
caractère spécial 'é')
  lcd.home();
   lcd.clear();
  lcd.print("Acc");
   lcd.write(1);
  lcd.print("s v");
   lcd.write(1);
   lcd.print("rouill");
   lcd.write(1);
   test badge(); //fonction pour lire le badge RFID présenté
   verification badge(); //fonction pour vérifier que le badge
présenté est valide
  test telecommande infra(); //fonction pour savoir si le bouton
"play/pause" de la telecommande infrarouge à été activé
   souhait_sortie(); //si une voiture souhaite sortir par le portail
depuis l'intérieur
```

```
}
void test badge() //on lit le badge RFID présenté
 if(monModuleRFID.isCard()) //Si il y a un badge à lire
   if(monModuleRFID.readCardSerial())
      Serial.print("Le code du badge est : ");
      for(char lecture=0; lecture<=4; lecture++) //on répète 4 fois</pre>
        UID[lecture] = monModuleRFID.serNum[lecture]; //on lit le
numéro d'identification du badge présenté et on le stock dans le
tableau UID
        Serial.print(UID[lecture]);
        Serial.print(".");
      Serial.println("");
      badge lu = 1; //on note que un badge a été lu
   monModuleRFID.halt(); //on stop la communication avec le module
RFID
 }
void verification badge() //fonction pour vérifier que le badge
présenté est valide
  if(UID[0] == badge BLEU[0] \&\& UID[1] == badge BLEU[1] \&\& UID[2] ==
badge BLEU[2] && UID[3] == badge BLEU[3] && UID[4] == badge BLEU[4])
//si le badge est bon(donc si c'est le badge bleu)
   lcd.clear();
   lcd.home();
   lcd.print("Badge valide");
   digitalWrite(pin ledV, HIGH);
   delay(1000);
   digitalWrite(pin_ledV, LOW);
   ouverture portail(); //fonction pour ouvrir le portail
   decision fermeture(); //cette fonction ferme autorise ou non la
fermeture du portail
  else if(badge_lu == 1) //si on a déja lu le badge
   lcd.clear();
   lcd.home();
   lcd.print("Badge non valide");
   digitalWrite(pin ledR, HIGH);
   delay(1000);
   digitalWrite(pin ledR, LOW);
   badge lu = 0;
```

```
void ouverture portail() //fonction pour ouvrir le portail
 lcd.clear();
 while(digitalRead(bouton_debut) != 1) //tant quele portail n'est
pas complètement ouvert
 {
   monServo.write(80); //on ouvre la porte
   digitalWrite(pin LED portail, HIGH);
   lcd.setCursor(0,0);
   lcd.print("Ouverture du");
   lcd.setCursor(0,1);
   lcd.print("portail...");
   fermeture defaut = millis();
 monServo.write(98);
void fermeture portail() //fonction pour fermer le portail
 lcd.clear();
 while(digitalRead(bouton fin) != 1) //tant que le portail n'est pas
complètement fermé
   monServo.write(110); //on ferme la porte
   digitalWrite(pin LED portail, HIGH);
   lcd.setCursor(0,0);
   lcd.print("Fermeture du");
   lcd.setCursor(0,1);
   lcd.print("portail...");
  badge lu = 0;
 UID[0] = 0;
 monServo.write(98);
  digitalWrite(pin LED portail, LOW);
 //affichage de la phrase : "Accés vérouillé" (avec l'insértion du
caractère spécial 'é')
  lcd.home();
 lcd.clear();
 lcd.print("Acc");
 lcd.write(1);
 lcd.print("s v");
 lcd.write(1);
 lcd.print("rouill");
 lcd.write(1);
void test_telecommande_infra() //fonction pour savoir si le bouton
"play/pause" de la telecommande infrarouge à été activé
```

```
if(monRecepteur infra.decode(&message recu))
    monRecepteur infra.resume(); //permet au recepteur de recevoir
de nouveaux messages
    if(message recu.value == 0xFFC23D)
      digitalWrite(pin ledV, HIGH);
      delay(500);
      digitalWrite(pin ledV, LOW);
      ouverture portail(); //fonction pour ouvrir le portail
      decision fermeture(); //cette fonction ferme autorise ou non la
fermeture du portail
    }
 }
float distance ultrason() //fonction pour mersué détécté la présence
d'un passage de voiture
 //on génere une impultion pour le TRIGGER du capteur à ultrason
  digitalWrite(pin TRIGGER, LOW);
  delayMicroseconds(2); //on attend 2 microsecondes
  digitalWrite(pin TRIGGER, HIGH);
 delayMicroseconds(10); //on attend 2 microsecondes
  digitalWrite(pin TRIGGER, LOW);
  float distance = pulseIn(pin ECHO, HIGH)/58.0; //on lit en on
convertit la distance en cm
  return distance;
void decision fermeture() //cette fonction autorise ou non la
fermeture du portail
 while((millis() - fermeture_defaut) < 10000) //tant que cela fait</pre>
moins de 10 secondes que le portail est ouvert
    lcd.home();
    lcd.clear();
    lcd.print("Portail ouvert");
    digitalWrite(pin LED portail, HIGH);
    delay(500);
    digitalWrite(pin LED portail, LOW);
    delay(500);
    if(distance ultrason() < 10) //si une voiture passe</pre>
      fermeture portail();
      break;
  fermeture portail();
```

```
void souhait_sortie() //si on souhaite sortir par le portail de
l'intérieur
{
  if(distance_ultrason() < 10) //si une voiture se présente
  {
    ouverture_portail();
    delay(3000);
    fermeture_portail();
  }
}</pre>
```

From

https://chanterie37.fr/fablab37110/ - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:portail&rev=1671716370

Last update: 2023/01/27 16:08

