



**INSTITUT UNIVERSITAIRE
DU GOLFE DE GUINEE
INSTITUT SUPERIEUR DES
TECHNOLOGIES AVANCEES**



Commande éclairage et contrôle température sur smartphone avec Arduino

May 23, 2024

Les Participants

Exposé encadré par Monsieur

① KAMGANG Pierre

Professeur chez **IUG-ISTA**.

Exposé réalisé par

① Yokwe Raoul Giresse

② Bidias Abang Donald

③ Tchuenkam Djoko Arnold Steve

④ Fotie Serges Magloire

Étudiants chez **IUG-ISTA** en classe de **IIA1S**.

Table de matières

1 Introduction

- Contexte
- Objectif
- Description de l'exposé
- Avantages
- Inconvénients

2 Mise en oeuvre

- Solutions matérielles
- Solutions logicielles
- Schéma synoptique
- Organigramme
- Schéma électrique
- Fonctionnement

3 Conclusion

4 Liens utiles

Introduction

Contexte

La domotique est un excellent moyen pour contrôler ses dépenses énergétiques, que ce soit pour les systèmes de chauffage, l'éclairage ou bien d'autres. Les nouvelles technologies offrent des appareils toujours plus précis pour, superviser, automatiser, commander à distance l'extinction des lumières, la fermeture des volets, etc Aussi, les logiciels facilitent beaucoup de tâches quotidiennes de l'être humain où il est devenu indépendant par rapport au contrôle des équipements utilisés.

Introduction

Objectif

Dans le cadre de notre première année en cycle **BTS** en Informatique Industrielle et Automatismes à **ISTA**, il nous est proposé, de mettre en pratique nos connaissances et nos compétences professionnelles pour montrer une face de l'Arduino en domotique.

Introduction

Description de l'exposé

Les Smartphones sont adoptés par le grand public pour des fins journalières, avec le développement des applications et les moyens de connectivité tels que le WIFI, l'infrarouge et le Bluetooth, le Smartphone devient une télé commande universelle pour les équipements électrique et électronique. Ayant une passion commune pour la programmation sur Arduino et la domotique. Étant pour le moment dans un exposé universitaire, en phase d'initiation, limité en moyens et en temps, nous avons décidé de restreindre notre travail, à développer dans un premier temps une application Android et un programme fonctionnant avec une carte Arduino UNO afin de contrôler un climatiseur, commander une lampe via Bluetooth, ensuite on passe à la réalisation.

Introduction

Avantages

- ① Le confort de vie;
- ② Sécurité;
- ③ Gain de temps;
- ④ Économie d'argent et d'énergie;

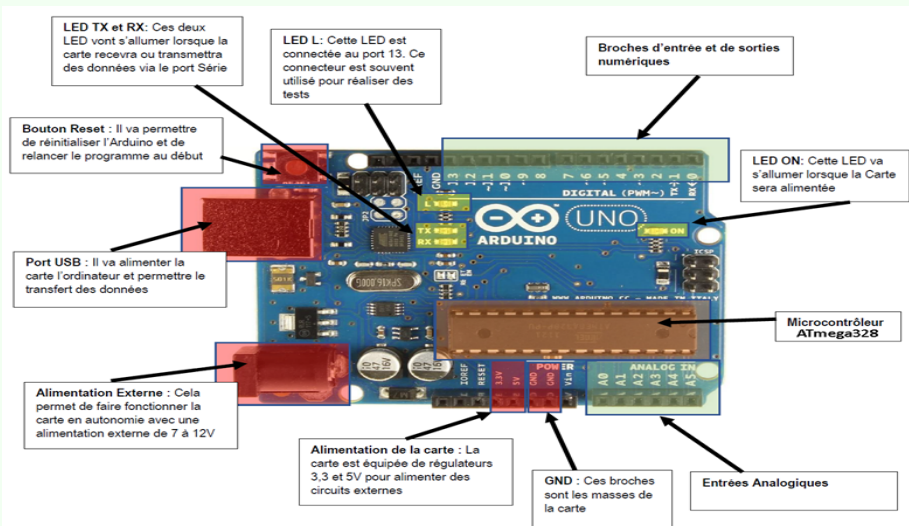
Introduction

Inconvénients

- ❶ Le prix des équipements domotiques reste assez élevé;
- ❷ La durée de vie des équipements domotiques est limitée;
- ❸ Risques de piratage;

Mise en oeuvre

Solutions matérielles: Carte Arduino Uno



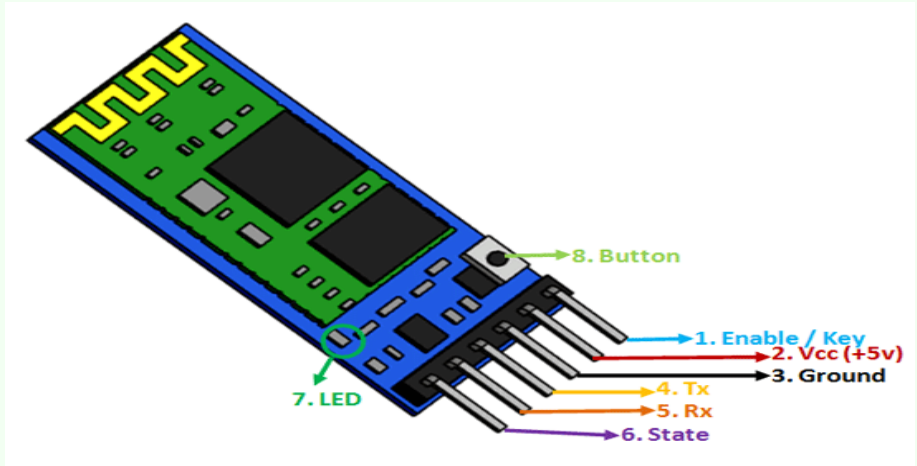
Mise en oeuvre

Solutions matérielles: Carte Arduino Uno

Une carte Arduino est carte électronique équipée d'un micro-contrôleur programmable. Le micro-contrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs.

Mise en oeuvre

Solutions matérielles: Module Bluetooth HC-05



Mise en oeuvre

Solutions matérielles: Module Bluetooth HC-05

Le Bluetooth est un standard de communication sans fil à très courte distance entre les périphériques électroniques. Dans notre projet, nous allons utiliser un module HC-05. Tous les modules Bluetooth HC-05 possèdent la même configuration. Le nom du module est **HC 05** et le code d'appareillage est **1234**. Aussi, une des particularités du module Bluetooth HC-05 est qu'il peut être utilisé en mode esclave ou en mode maître, en d'autres termes, le HC-05 peut être configuré en tant qu'émetteur ou en tant que récepteur, ou même en tant que point d'accès.

Mise en oeuvre

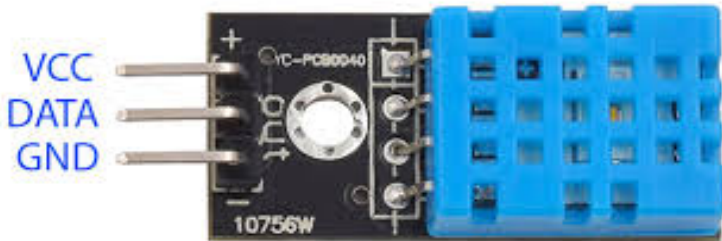
Solutions matérielles: Module Bluetooth HC-05

Le module Bluetooth HC-05 présente 6 broches pour permettre d'établir la connexion :

- ➊ VCC: broche d'alimentation. Typiquement connectée à la broche 5V de l'Arduino.
- ➋ GND: masse. Typiquement connectée à la broche GND de l'Arduino.
- ➌ RX: broche de réception. Typiquement connecté à la broche de transmission (TX) de l'Arduino.
- ➍ TX: broche de transmission. Typiquement connecté à la broche de réception (RX) de l'Arduino.
- ➎ State: sortie pour indiquer l'état du module (en attente, connecté, etc.)
- ➏ Enable: entrée pour activer ou désactiver le module.

Mise en oeuvre

Solutions matérielles: Module DHT11



Mise en oeuvre

Solutions matérielles: Module DHT11

Le capteur DHT11 mesure la température et l'humidité. Son principe de fonctionnement est le suivant:

- ❶ Le DHT11 utilise une technologie capacitive pour mesurer l'humidité relative dans l'air. Il comporte une électrode qui est exposée à l'air ambiant, qui est reliée à un circuit électronique intégré. Lorsque l'air est plus humide, l'électrode est recouverte d'une couche d'eau qui augmente la capacité de l'électrode. Le circuit électronique intégré mesure cette capacité et calcule l'humidité relative en fonction de cette mesure.
- ❷ Pour mesurer la température, le DHT11 utilise un thermistor, qui est un composant qui a une résistance qui varie en fonction de la température. Le circuit électronique intégré mesure la résistance du thermistor et calcule la température en utilisant une courbe de température calibrée.

Mise en oeuvre

Solutions matérielles: Module DHT11

Les données de température et d'humidité sont ensuite transmises à l'Arduino via un signal numérique sur un seul fil, qui peut être lu par la bibliothèque DHT11 pour obtenir les lectures de température et d'humidité.

Le brochage du DHT11 est:

- ❶ VCC: C'est la broche d'alimentation qui doit être connectée à une source de tension de 3 à 5V pour alimenter le capteur.
- ❷ Data: C'est la broche de communication qui est utilisée pour envoyer les données de température et d'humidité au dispositif connecté.
- ❸ GND: C'est la broche de masse qui doit être connectée à la masse du dispositif connecté pour assurer une bonne connexion électrique.

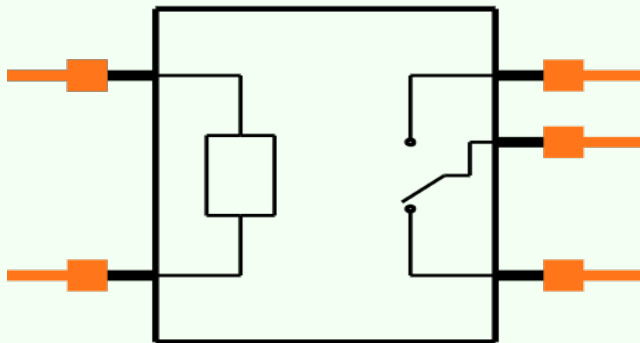
Mise en oeuvre

Solutions matérielles: Module relais 5v



Mise en oeuvre

Solutions matérielles: Module relais 5v



Exemple de symbole

Mise en oeuvre

Solutions matérielles: Module relais 5v

Le module de relais 5V contrôlable qui permet d'isoler la partie commande de la partie puissance. Ici le circuit de commande est la carte Arduino et le circuit de puissance est la lampe 230V, le climatiseur. Son câblage avec Arduino est:

- ➊ Connecter les broches VCC et GND du module relais aux broches 5V et GND de l'Arduino.
- ➋ Connecter la broche S du module relais à une broche de sortie numérique de l'Arduino.
- ➌ Connecter les broches NO et COM du module relais à la charge.

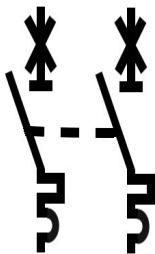
Mise en oeuvre

Solutions matérielles: Disjoncteur Phase-Neutre(DPN)



Mise en oeuvre

Solutions matérielles: Disjoncteur Phase-Neutre(DPN)



Symbole électrique

Mise en oeuvre

Solutions matérielles: Disjoncteur Phase-Neutre(DPN)

Un disjoncteur est un interrupteur électrique à commande automatique conçu pour laisser circuler le courant électrique, et, protéger un circuit électrique et les personnes contre les dommages causés par un courant excessif provenant d'une surcharge, d'un court-circuit ou d'une fuite à la terre (disjoncteur différentiel). Il est également utile pour isoler électriquement le circuit en cas de dépannage.

Mise en oeuvre

Solutions matérielles: Boîte de dérivation étanche



Mise en oeuvre

Solutions matérielles: Boîte de dérivation étanche

Destinées à recevoir et à protéger des éléments de jonction électriques à installer dans les locaux domestiques ou industriels. Dimensions(mm): 100x140x60.

Mise en oeuvre

Solutions matérielles: Lampe LED 230V



Mise en oeuvre

Solutions matérielles: Lampe LED 230V

LED signifie diode électroluminescente. Une lampe LED produit de la lumière en faisant passer le courant électrique à travers un matériau semi-conducteur, la diode, qui émet ensuite la lumière lorsqu'elle est alimentée(principe de l'électroluminescence). Les LED sont économique car avec seulement 7W, on économise 80% d'énergie. Caractéristiques:

- ① En plastique à l'extérieur et châssis en aluminium.
- ② Puissance : 7W.
- ③ Tension : 170-240V.
- ④ Intensité lumineuse : 90 lumen.

Mise en oeuvre

Solutions matérielles: Douille électrique



Mise en oeuvre

Solutions matérielles: Douille électrique

En éclairage, c'est un support en plastique à l'extérieur et châssis en aluminium permettant la fixation du culot d'une lampe électrique.

Mise en oeuvre

Solutions matérielles: prise électrique 2P+T



Mise en oeuvre

Solutions matérielles: prise électrique 2P+T

Une prise électrique est un connecteur permettant de relier des appareils électriques, domestiques ou industriels, au réseau électrique. Elle est directement reliée au tableau électrique par un circuit électrique.

Composition de la prise 3P+T :

- ❶ La borne de phase L, qui alimente en énergie l'appareil branché.
- ❷ La borne neutre N, qui permet au courant de retourner vers le circuit une fois l'appareil alimenté.
- ❸ La borne de terre (fil vert-jaune), associée à une protection différentielle, elle protège l'utilisateur en cas de défaut d'isolement.

Mise en oeuvre

Solutions matérielles: Climatiseur



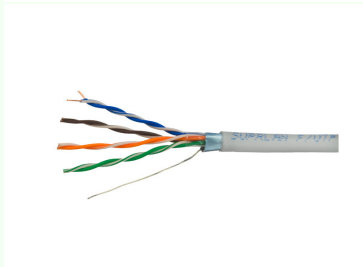
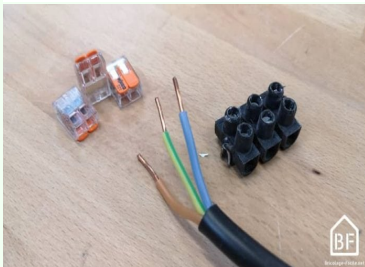
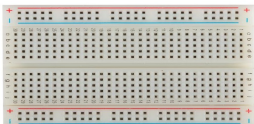
Mise en oeuvre

Solutions matérielles: Climatiseur

Le climatiseur est un équipement indispensable dans de nombreux endroits pour réguler la température et l'humidité de l'air. La régulation de la température est l'un des principaux rôles du climatiseur. Lorsqu'il fait chaud à l'extérieur, l'air chaud peut pénétrer dans un bâtiment, augmentant ainsi la température intérieure. Le climatiseur abaisse la température en absorbant l'air chaud de la pièce et en le refroidissant avant de le faire circuler dans la pièce. Il maintient une température confortable pour les occupants du bâtiment, ce qui est important pour les appareils électriques, électroniques, informatique qui doivent fonctionner dans les environnement à température acceptable.

Mise en oeuvre

Solutions matérielles: Accessoires de raccordement



Mise en oeuvre

Solutions matérielles : Smartphone



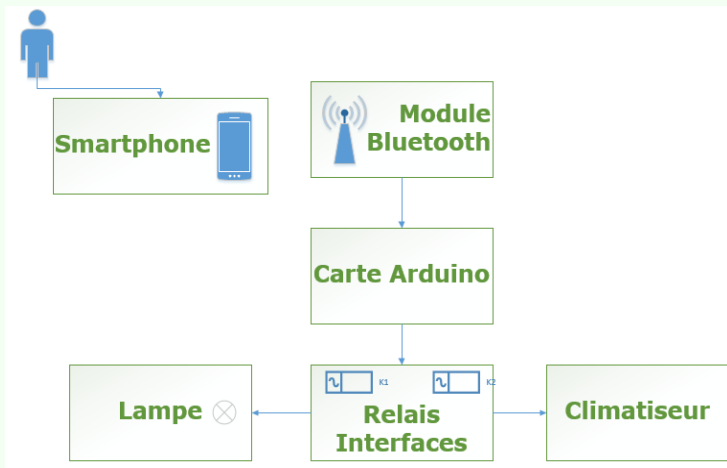
Mise en oeuvre

Solutions matérielles: Smartphone

Sera utile pour la commande et l'acquisition des données.

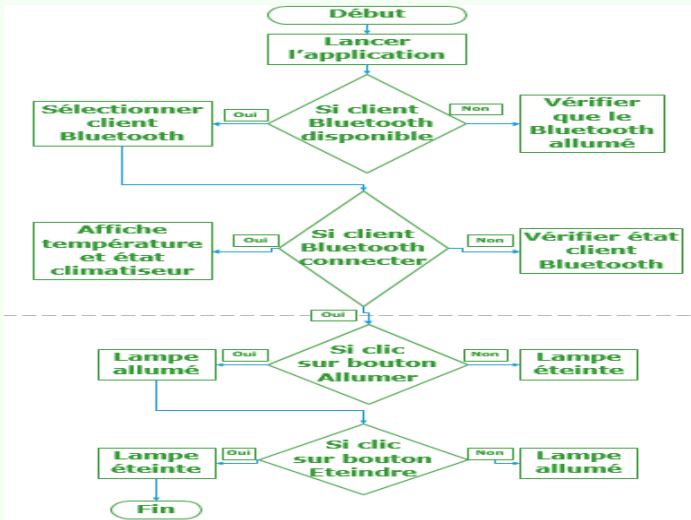
Mise en oeuvre

Schéma synoptique



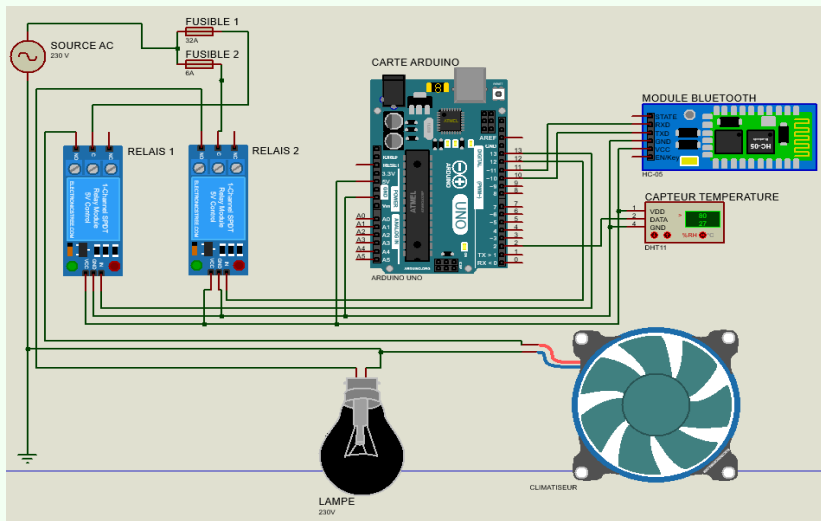
Mise en oeuvre

Organigramme



Mise en oeuvre

Schéma électrique



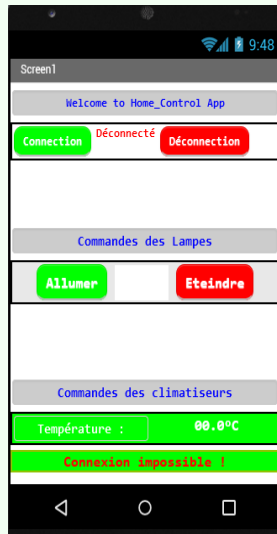
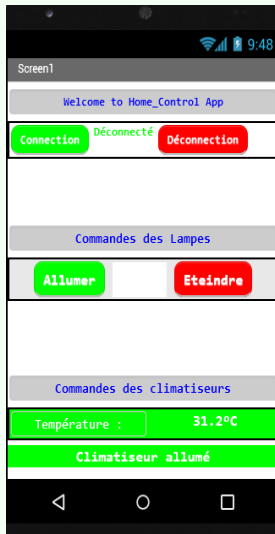
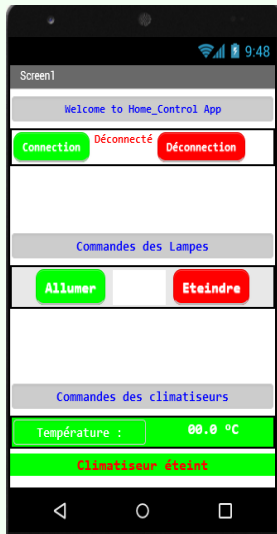
Mise en oeuvre

Fonctionnement

Des la mise sous tension du système, la lampe 230V est éteinte, le module DHT11 transmet les données relatives à la température de la salle au module Arduino, qui peut être visible sur le moniteur série de l'IDE Arduino, aussi le climatiseur peut être en marche ou en arrêt car sa mise en route dépend de la température de la salle. Par contre le module Bluetooth HC-05 se met à clignoter, attendant un appairage Bluetooth. Lorsqu'un utilisateur à partir de l'application sur son Smartphone clic sur le bouton **"Connection"**, une liste de Bluetooth visible et disponible à sa portée s'affiche; un choix lié au module Bluetooth HC-05 permet la connection. Une fois connecté, la température du local et l'état actuel du climatiseur s'afficheront directement sur son écran. Un appui sur le bouton **"Allumer"** met la lampe 230V en marche, par contre un appui sur le bouton **"Eteindre"** permet de l'éteindre. Enfin, une action sur le bouton **"Déconnection"** provoque la perte de connectivité entre le module Bluetooth et le Smartphone.

Mise en oeuvre

Etats possibles de l'application Home_Control



Mise en oeuvre

Solutions logicielles: Application Androïde

Pour développer notre application nous avons opté pour un environnement de programmation visuel et intuitif: **MIT App Inventor**. Ce site disponible ici <https://appinventor.mit.edu/>, met à notre disposition les blocks où avec une logique de programmation, facilitent la création d'applications en beaucoup moins de temps. Vous verrez ci-dessous comment nous avons utiliser ces blocks pour notre application nommée: **Home_Control**.

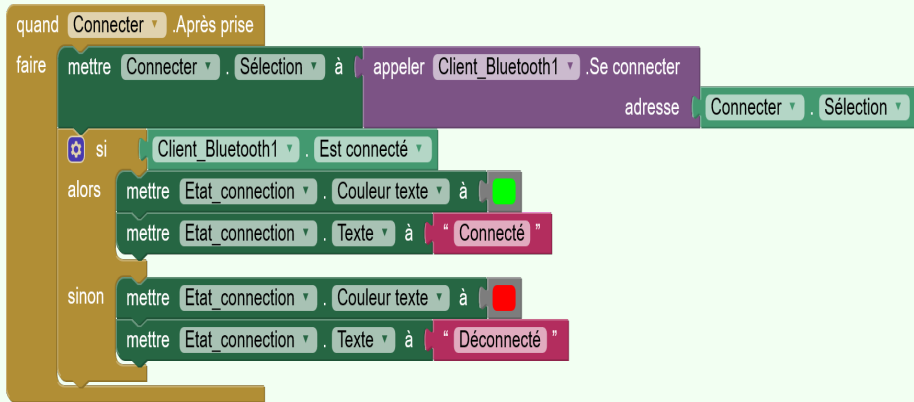
Mise en oeuvre

Solutions logicielles: Application Androïde



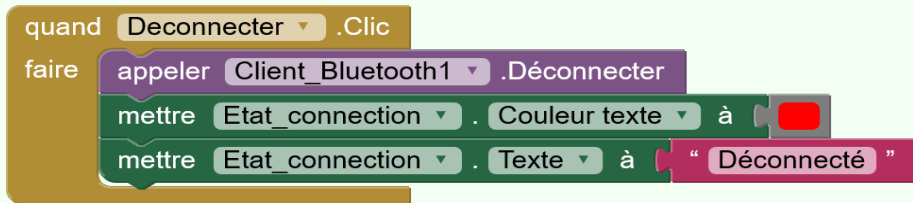
Mise en oeuvre

Solutions logicielles: Application Androïde



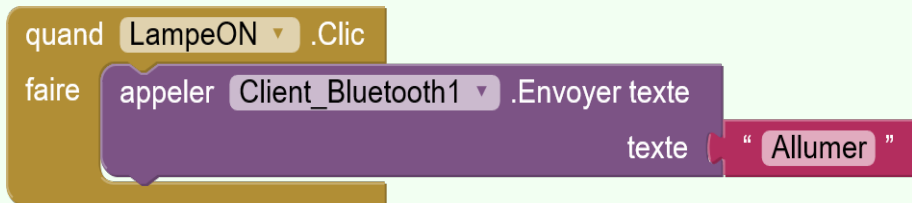
Mise en oeuvre

Solutions logicielles: Application Androïde



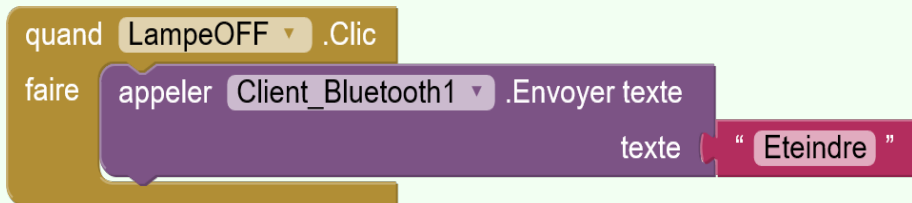
Mise en oeuvre

Solutions logicielles: Application Androïde



Mise en oeuvre

Solutions logicielles: Application Androïde



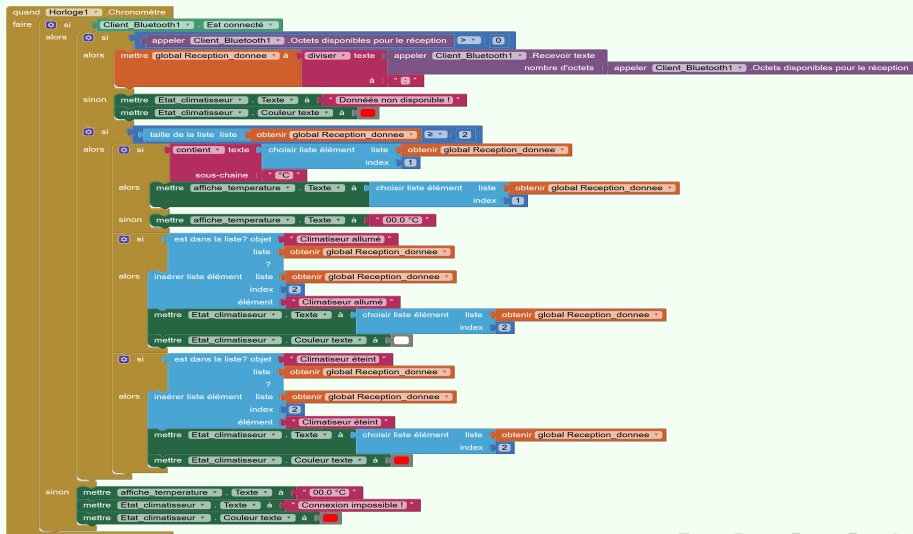
Mise en oeuvre

Solutions logicielles: Application Androïde

initialise global `Reception_donnee` à  créer une liste vide

Mise en oeuvre

Solutions logicielles: Application Androïde



Mise en oeuvre

Solution logicielle: Programme Arduino

```
1  /* ~~~~~Home_Control~~~~~
2
3  Ce programme permet:
4  De commander une lampe à partir dun smartphone.
5  De lire et afficher sur le smartphone la température dun local.
6  De commander un climatiseur en fonction la température du local.
7  D'afficher également l'état du climatiseur sur le smartphone.
8  Tout ceci sera possible grâce à la carte arduino uno,
9  le module bluetooth HC05 et le capteur température DHT11
10 */
11
12 /* On aura aussi la possibilité de voir à travers le moniteur serie
13 de l'IDE arduino les différentes interactions entre les composants.
14 Chaque instruction ou block de code est précédé d'un commentaire.
15 */
16
17 /* Inclusion de la librairie Software Serial pour la communication
18 série logicielle sur n'importe quelles broches de la carte Arduino
19 */
20 #include <SoftwareSerial.h>
```

Mise en oeuvre

Solution logicielle: Programme Arduino

```
22  /* On définit la broche d'entrée digital 10 pour la réception
23  | des données et la nommée RX. Aussi la broche d'entrée digital
24  | 11 pour la transmission des données et la nomméé TX du module.
25  Broches sur lesquelles sera branché le module bluetooth.
26  */
27  #define RX 10
28  #define TX 11
29
30  /* Inlcusion de la librerie DHT qui contient les fonctions
31  nécessaires pour lire les données du capteur DHT11.
32  */
33  #include "DHT.h"
34
35  // On définit la broche d'entrée digital 2 et la nommée
36  // DHTPIN sur laquelle est branchée le DHT
37  #define DHTPIN 2
38  #define DHTTYPE DHT11 //On définit le type de module DHT à utiliser
39
```

Mise en oeuvre

Solution logicielle: Programme Arduino

```
40  /* On se sert du constructeur SoftwareSerial pour créer un objet serie
41  pour le hc05 "myHC05Serie" et lui passer les paramètres les broches
42  de reception et transmission.
43  */
44  SoftwareSerial myHC05Serie(RX, TX);
45
46  /* Déclaration d'une variable nommée "donnee" de type string pour
47  stocker les données reçues du smartphone à travers le module HC05.
48  */
49  String donnee;
50
51  // On crée une variable temperature de type float où stocker
52  // la valeur de température lue de notre DHT "mydht".
53  float temperature = 0.0;
54
55  // Déclaration d'une variable nommée "temperature_test", de type float
56  // pour définir le seuil d'allumage et extinction du climatiseur.
57  int temperature_value = 20;
```

Mise en oeuvre

Solution logicielle: Programme Arduino

```
58
59  /* Création d'un objet DHT "mydht", définition du type et
60  de la broche sur laquelle il est branché comme ces paramètres.
61  */
62  DHT mydht(DHTPIN, DHTTYPE);
63
64  // La fonction "setup()" est exécutée une seule fois au démarrage de l'Arduino.
65  void setup() {
66      // Définition de la broche 12 comme sortie pour commander la lampe.
67      pinMode(12, OUTPUT);
68      // Définition de la broche 13 comme sortie pour commander le climatiseur.
69      pinMode(13, OUTPUT);
70
71      /* Initialisation de la communications séries entre la
72      carte Arduino et l'ordinateur à une vitesse de 9600 bauds.
73      */
74      Serial.begin(9600);
75      Serial.println("Prêt ...");
```

Mise en oeuvre

Solution logicielle: Programme Arduino

```
77  /* Définition de la vitesse de communication pour notre objet
78  myHC05Serie à 9600 bauds et initialise le capteur DHT11 en appelant
79  la fonction "begin()" sur notre objet "mydht".
80  */
81  myHC05Serie.begin(9600);
82  mydht.begin();
83  }
84
85  // La fonction "loop()" est exécutée en boucle continuellement
86  // après le démarrage de l'Arduino.
87  // Les fonctions "Serial.print()" et "Serial.println()"
88  // seront utilisées pour envoyer les données dans le moniteur série.
89  void loop() {
90      // Lecture des données du module HC05, Tant que les données sont
91      // disponibles dans le tampon série de notre objet "myHC05Serie".
92      while (myHC05Serie.available() > 0) {
93          donnee += (char)myHC05Serie.read();
94      }
```

Mise en oeuvre

Solution logicielle: Programme Arduino

```
95
96  /*Traitement des données reçues */
97
98  // On vérifie qu'une commande a été reçue, et on affiche dans
99  // le moniteur serie cette commande.
100  if (donnee != "") {
101      Serial.print("Une commande a ete recue : ");
102      Serial.println(donnee);
103  }
104  // Si la donnée reçue est "Allumer", on allume la lampe.
105  // On affiche dans le moniteur, "Lampe allumée"
106  if (donnee == "Allumer") {
107      digitalWrite(12, HIGH);
108      Serial.println("Lampe allumée");
109  }
110
111  // Si la donnée reçue est "Eteindre", on éteint la lampe.
112  // On affiche dans le moniteur, "Lampe éteinte"
113  if (donnee == "Eteindre") {
114      digitalWrite(12, LOW);
115      Serial.println("Lampe éteinte");
116  }
117  donnee = ""; // Réinitialisation de la variable donnee
118
```

Mise en oeuvre

Solution logicielle: Programme Arduino

```
119  /* Mesure et affichage console série température */
120
121  // lecture de la valeur de température de notre DHT "mydht".
122  temperature = mydht.readTemperature();
123
124  // On affiche dans la console serie la valeur de la température lue.
125  Serial.print("température du local : ");
126  Serial.print(temperature);
127  Serial.println(" °C");
128
129  /* Envoi température au module bluetooth HC05 pour le smartphone */
130  myHC05Serie.print(temperature + String(" °C :"));
131
```

Mise en oeuvre

Solution logicielle: Programme Arduino

```
132  /* On compare la valeur actuelle de la température avec le seuil
133  définit, pour allumer ou éteindre le climatiseur. On envoie également
134  l'état du climatiseur au module bluetooth HC05 pour le smartphone.
135  */
136  if (int(temperature) > temperature_value) {
137      digitalWrite(13, HIGH);
138      myHC05Serie.print("Climatiseur allumé:");
139  } else {
140      digitalWrite(13, LOW);
141      myHC05Serie.print("Climatiseur éteint:");
142  }
143
144  // On crée un délai de 2 secondes entre chaque lecture,
145  // pour éviter de surcharger la communication série.
146  delay(2000);
147 }
```


Mise en oeuvre

Charger le code au microcontrôleur de la carte Arduino



**Bouton pour vérifier
le éventuels erreurs
dans le code**

**Bouton pour
transférer le code
au microcontrôleur**

Conclusion

L'accomplissement de cet exposé dans le cadre de la commande d'éclairage et contrôle température sur Smartphone avec Arduino, nous a permis de nous introduire côté logiciel dans la programmation d'Arduino avec son IDE, Mit App Inventor. Ainsi que du matériel tel que l'Arduino UNO, capteur de température DHT11, module Bluetooth HC-05 et différents composants électroniques et électriques. Comme amélioration à ce travail nous comptons ne pas limiter le système à une salle, introduire un module wifi ou le GSM qui permet la couverture d'une plus grande zone, développer une application plus sécurisante en version mobile et Web.

Liens utiles

Arduino Documentation[en ligne]

Disponible sur: → <https://www.arduino.cc/>

MIT App Inventor Environnement de programmation visuel[en ligne]

Disponible sur: → <https://appinventor.mit.edu/>