Controler sa maison à l'aide d'un Raspberry: Exemples de codes de controle des composants

Essig Meryll Fortin Loic Rocacher Tamara

4 février 2016

Introduction

Dans l'attente du materiel electronique, nous avons commencé à organiser le code de controle des differents composants. Pour cela nous avons définit les premiers éléments que nous voulions utiliser pour le controle de la maison. Dans l'idée d'etre proche de la réalité, nous voulons travailler sur une maquette de maison, ce qui implique des limites sur ce qui es controlable et sur la taille des composants. Ainsi, nous avons décider de commencer avec les actions suivantes :

- controler la lumières des pièces de vie,
- controler la mise en marche du chauffage principale,
- controler l'ouverture des volets de la chambre

Chaque composant pourra etre controlé sur demande de l'utilisateur, via l'interface, ou automatiquement dans different modes pré-définis :

- mode jour : lorsque le soleil se lève, les volets s'ouvrent,
- mode nuit : lorsque le soleil se couche, les volets se ferment, les lumières s'allument
- mode hiver : le chauffage se met en marche sur une periode définie par l'utilisateur

D'autres modes ou d'autres actions pourront etre ajoutés par la suite. Cependant nous sommes partis de ces trois composants pour rechercher et construire le code général de chacun.

Controle d'une LED pour la lumière

```
1
2  import com.pi4j.io.gpio.GpioController; // Permet de contrler les GPIO
3  import com.pi4j.io.gpio.GpioFactory; // Contient les mthodes statiques permettant de crer des instances de
       GpioController.
4  import com.pi4j.io.gpio.GpioPinDigitalOutput; // Les sorties sur le GPIO (contrle sortant des GPIO)
5  import com.pi4j.io.gpio.PinState; // Permet d'avoir des informations sur l'tat du pin
6  import com.pi4j.io.gpio.RaspiPin; // Les diffrents pins du Raspeberry
7  import java.lang.System;
8  import java.lang.Thread;
9  import java.lang.InterruptedException;
10
11 public class led {
12     private final GpioPinDigitalOutput pin; // Permet l'accs en sortie vers le pin.
13
14     public led(int pin) {
15         // Pin fix sur 1 dans le code. Ne tien pas compte de l'argument de fonction
16
17         // Il faudrait vrifier que le pin est bon, mais on va considrer que oui
18         this.pin = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01, "LedControl", PinState.LOW); // L'accs
               en criture au pin pour allumer la led. Par dfaut, la led est teinte .
19     }
20  /*
21     Allume la led
22      */
23     public void on() {
24         //this.pin.setShutdownOptions(true, PinState.HIGH);
25         this.pin.high();
26     }
27
28     /*
29     Eteint la led
30      */
31     public void off() {
32         //this.pin.setShutdownOptions(true, PinState.LOW);
33         this.pin.low();
34     }
35
36
37     /*
38     Vrai si la led est teinte , faux sinon.
39      */
40     public boolean isLow() {
41         return this.pin.isLow();
42     }
43
44     /*
45     Vrai si la led est allume , faux sinon
46      */
47     public boolean isHigh() {
48         return this.pin.isHigh();
49     } /*
50       Rcupre l' tat du pin (HIGH ou LOW)
51      */
52     public boolean getState() {
53         return this.pin.getState();
54     }
```

```
    /*
    Eclairage clignotant de la led

    time int Reprsente le temps que doit rester la led alllume
     */
    public void pulse(int time) {
        // Mme chose, il faudrait vrifier que les valeurs de time soient correctes.
        this.pin.pulse(time, true);
    }
}

public class ledControl {
    /*
    Programme principal, permet de lancer ( thoriquement ...) un test de led sur le raspberry.
    Note : un thread est susceptible de lancer un InterruptedException. Il est ignor ici.
     */
    public static void main(String[] args) throws InterruptedException {
        led ledtest = new led(1);

        // Donne un accs au gpio
        final GpioController gpio = GpioFactory.getInstance();

        ledtest.on();
System.out.println("LED allume.");

        Thread.sleep(5000);

        ledtest.off();

        System.out.println("LED teinte.");

        Thread.sleep(5000);

        System.out.println("LED blink (interval : 1s).");

        ledtest.pulse(1000);

        // Libre les accs GPIO
        gpio.shutdown();
    }
}
```

Controle d'un micro-moteur pour les volets électriques

Controle d'un luxmetre pour la detection de luminosité exterieure