

# HomeEnergyManager

---

An ALASCA project

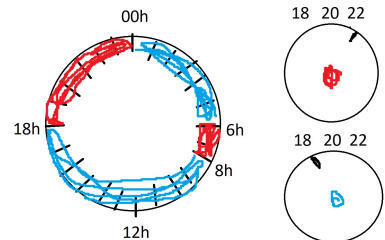
Bruno PINTO  
Pierre GOMEZ

# Objectifs

- Logement autonome dans sa gestion d'énergie
- Équipements intelligents offrant une interface d'utilisation
- Contrôleur centralisant la gestion de tous les équipement inscrits
- Simuler une consommation d'énergie et une utilisation

# Choix des équipements

- Production d'énergie
  - Éolienne : aléatoire
  - Générateur à essence : intermittent
- Stockage
  - Super-Batterie du logement : production/consommation
- Consommation
  - Télévision : incontrôlable
  - Réfrigérateur : suspensible
  - Chauffage : suspensible planifiable
    - 2 thermostats planifiés sur plages horaires
    - Exemple : 22° de 6h à 8h et de 18h à 00h  
18° de 8h à 16h et de 00h à 6h



# Réalisation

Exemple de connections entre le Controller et le GasolineGenerator :

Le Controller appelle la `setHigh()` sur son `GasolineGeneratorOutBoundPort`

Celui-ci appelle `setHigh()` sur le connecteur qui lui est associé

Le connecteur se charge de faire le lien avec l'interface des services proposés par `GasolineGenerator`, c'est à dire :

- Pour faire `setHigh` (la méthode n'existe pas forcément dans l'équipement)

- Il faut appeler `set.mode(Mode.High)`

- Puis appeler `setPowerOutput_W(3000)`

# Scénario de test du Controller

```
public void scenario() throws Exception {
    System.out.println("Hello ! This is a new day !");
    System.out.println("Checking current consumption :");
    System.out.println(this.energyCounterOutboundPort.getInstantConsumption() + "W");
    System.out.println("Checking current battery level :");
    System.out.println(this.batteryOutboundPort.getCurrentState() + "Wh");
    System.out.println("Checking wind speed : ");
    double windspeed = this.windturbineOutboundPort.getWindSpeed();
    System.out.println(windspeed + " m/s");
    if ((windspeed < 3.5) || (windspeed > 25) ) {
        System.out.println("Shutting off windTurbine -- wind is too low or too high");
        this.windturbineOutboundPort.switchOff();
        System.out.println("WindTurbine on ? " + this.windturbineOutboundPort.isOn());
    }
    System.out.println("Starting Generator on High mode");
    this.generatorOutboundPort.switchOn();
    this.generatorOutboundPort.setHigh();
    System.out.println("Generator is On ? " + this.generatorOutboundPort.isOn() + " and selected mode is " + this.generatorOutboundPort.getMode());
    System.out.println("Generator power output : " + this.generatorOutboundPort.getPowerOutput() + "W");
    System.out.println("Is the fridge on ? : " + this.fridgeOutboundPort.getCurrentState());
    System.out.println("Turning on fridge");
    this.fridgeOutboundPort.switchOn();
    System.out.println("Is the fridge on ? : " + this.fridgeOutboundPort.getCurrentState());
    System.out.println("Checking home temperature : ");
    System.out.println("It's currently " + this.heatingOutboundPort.getHomeTemp() + "°C at home");
    System.out.println("Turning on heating");
    this.heatingOutboundPort.switchOn();
    System.out.println("Is heating on ? : " + this.heatingOutboundPort.getState());
    System.out.println("Is Tv on ? : " + this.tvOutboundPort.getState());
}
```

# Scénario de test du Controller

```
executing...
Hello ! This is a new day !
Checking current consumption :
0.0W
Checking current battery level :
0.0Wh
Checking wind speed :
0.0 m/s
Shutting off windTurbine -- wind is too low or too high
WindTurbine on ? false
Starting Generator on High mode
Generator is On ? true and selected mode is HIGH
Generator power output : 3000.0W
Is the fridge on ? : false
Turning on fridge
Is the fridge on ? : true
Checking home temperature :
It's currently 20.0°C at home
Turning on heating
Is heating on ? : true
Is Tv on ? : false
finalising...
shutting down...
ending...
```

# Conclusion

- Connexion fonctionnelle des composants grâce à BCM