Conceptions Systemes en C++: Borne de recharge de vehicule

By Kossi Pascal SEWODA & Kahina ACHARI, M1 SME, 2020-2021

Generated by Doxygen 1.9.1

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 BaseClient Class Reference

Public Member Functions

- int authentifier (int num_carte)

  Methode qui permet d'authentifier un client.
- int recherche (int num)

  verifie si le numero fourni existe dans la base
- int enregistrer (int numeroClient)

  Enrgitrer un numero client Enregistre un client dans la base de donnees database.txt qui est une base de donnees locale sous forme de fichier texte.
- void ajout ()

  Interface permttant un enregisstrement de client Cette fonction utilise la fonction enregistrer pour permettre a un administrateur de la borne d'ajouter un client la base de donnees.
- int lire ()

  Mets a jour la variable dbClient: tableau dynamique avec vector Cette fonction est appeler au demarrage de la borne pour mettre a jour la variable dbClient qui est un vector local. Ceci permet de recuperer tout les clients et ainsi reduire le temps d'authenfication d'un client. Elle est egalement appele apres l'ajout() d'un client dans la base. NB: Ceci n'est pas neccessaire dans notre car la DB n'est pas distant.
- void supprimer_clients ()

  Supprime un client dans la DB et met a jour le vecteur dbClient avec lire().

3.1.1 Detailed Description

Definition at line 21 of file baseclient.h.

3.1.2 Member Function Documentation

3.1.2.1 authentifier()

```
int BaseClient::authentifier (
            int num_carte )
```

Methode qui permet d'authentifier un client.

Parameters

void,pas	de parametre

Returns

Booleen

Pour permettre a l'utilisateur de recharger son vehicule il faut l'authentifier au prealable d'ou cette fonction qui verifie bien que l'utilisateur est enregistrer dans la bas de donnees.

Definition at line 7 of file baseclient.cpp.

The documentation for this class was generated from the following files:

- baseclient.h
- baseclient.cpp

3.2 Boutons Class Reference

Public Member Functions

- void initialiser ()

  Permet d'initialiser les boutons.
- int charge ()

  Signale l'appui sur le bouton charge. L'appui sur le bouton charge doit se faire avnat 8 secondes. pour pouvoir demarrer un cycle de charge.
- int stop ()

  Signale l'appui sur le bouton stop de la borne. L'appui sur le bouton charge doit se faire avnat 8 secondes. pour pouvoir demarrer un cycle de charge.

3.2.1 Detailed Description

Definition at line 12 of file boutons.h.

3.2.2 Member Function Documentation

3.2.2.1 charge()

```
int Boutons::charge ( )
```

Signale l'appui sur le bouton charge. L'appui sur le bouton charge doit se faire avnat 8 secondes. pour pouvoir demarrer un cycle de charge.

Parameters

void	

Returns

int

Definition at line 17 of file boutons.cpp.

3.2.2.2 initialiser()

```
void Boutons::initialiser ( )
```

Permet d'initialiser les boutons.

Autorise l'acces a la memoire partagee. Appeler a chaque interaction avec la borne. Sinon affiche erreur d'acces

Parameters

void	

Returns

void

Definition at line 8 of file boutons.cpp.

3.2.2.3 stop()

```
int Boutons::stop ( )
```

Signale l'appui sur le bouton stop de la borne. L'appui sur le bouton charge doit se faire avnat 8 secondes. pour pouvoir demarrer un cycle de charge.

Parameters

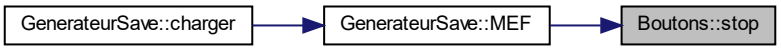
void	

Returns

int

Definition at line 24 of file boutons.cpp.

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- boutons.h
- boutons.cpp

3.3 GenerateurSave Class Reference

Public Member Functions

- void **initialiser** ()
- void genererPWM (pwm tension)

  genere une tension PWM (modulation d'amplitude)
- int tension ()

  Recupere la tension sur le fil pilote.
- void MEF (Etat init)

  Machine a etat fini de la charge.
- void charger ()

  Demarre la machine a etat fini (MEF)
- void ouvert ()

  Ouvre lle contacteur lorsque le vehicule est banche pour commancer un cycle de charge.
- void fermer ()

  Fermer le contacteur lorsque le vehicule est banche pour commancer un cycle de charge.
- void deconnecter ()

  Remet la borne a l'etat initial. Etat E : Quand le véhicule détecte que la batterie est chargée, S2 est ouvert. Le signal remonte à 9V/-12V, indiquant à la borne d'ouvrir le contacteur AC. Le voyant « charge » s'allume en Vert. Etat F : retour à 12V quand la prise est déconnectée. Le voyant « charge » s'éteint.Lors de la reprise du véhicule, le client s'identifie à nouveau sur la borne. Si le véhicule est encore en charge, il appuie sur le bouton « Stop », sinon il peut le récupérer directement après l'avoir débranché. Le contacteur AC doit être ouvert, la prise déverrouillée. Une fois la prise débranchée par le client, la trappe doit être verrouillée, voyant « Prise » éteint et le voyant « disponible » allumé.

3.3.1 Detailed Description

Definition at line 22 of file generateur_save.h.

3.3.2 Member Function Documentation

3.3.2.1 genererPWM()

```
void GenerateurSave::genererPWM (
            pwm tension )
```

genere une tension PWM (modulation d'amplitude)

Parameters

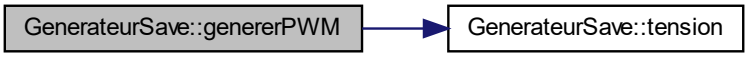
pwm	tension : genere une tension pwm specifier en entree. La variable tension = (STOP,DC,AC_1K,AC_CL). Ces differentes tension sont uitliser pour communiquer avce le vehicule.

Returns

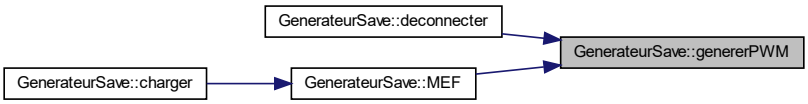
void A travers la variable memoire gene_pwm elle genere la tension pwm selon les different commande existant.

Definition at line 15 of file generateur_save.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



3.3.2.2 MEF()

```
void GenerateurSave::MEF (
            Etat init )
```

Machine a etat fini de la charge.

Parameters

Etat	init : Etat initial systeme c'est a dire quand aucun vehicule n'est en charge.

Returns

void

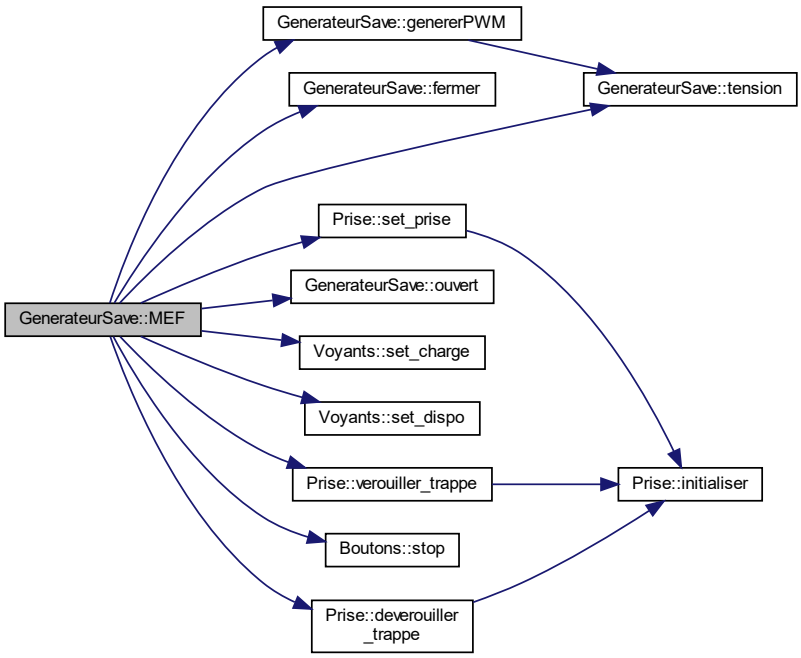
Cette valeur de tension est modifier par le simulateur. Donne l'etat courant du systeme. Un contrôleur de recharge, nommé générateur SAVE et situé côté infrastructure, vérifie les éléments suivants avant d'enclencher la recharge : érification que le véhicule est bien connecté au système (Etat B); Vérification que la masse du véhicule est bien reliée au circuit de protection de l'installation (Etat C); Vérification de la cohérence des puissances entre le câble, le véhicule et le circuit de recharge (Etat D); Détermination de la puissance maximale de recharge qui sera allouée au véhicule. L'ensemble de ces vérifications et de la communication se font au travers d'une communication sur fil spécifique, dit « fil PILOTE ». Voir la représentation d'une prise Type 3 sur la Figure 2 et la connectique entre la borne et le véhicule en Figure 3 (on notera la présence du connecteur S2 sur le véhicule). La figure 4 représente la valeur de la tension durant le processus. Ainsi, la charge se fait en fonction du dialogue suivant entre le véhicule et la borne (Figure 4):

Etat A : véhicule électrique non connecté, le générateur SAVE fournit une tension de + 12V. Le voyant « charge » s'allume en rouge. Etat B : véhicule électrique connecté et système d'alimentation non disponible, la tension chute à +9V. S2 est ouvert.

Etat C : véhicule électrique connecté et système d'alimentation disponible, le générateur SAVE fournit un signal carré +9V / -12V de fréquence 1 kHz qui aura pour effet de fermer le contacteur S2 sur le véhicule. Etat D : S2 est fermé et engendre une nouvelle chute de tension à 6V. Le générateur SAVE fournit un signal (+6V/-12V) de fréquence 1 kHz à rapport cyclique variable. (signal PWM modulation en largeur d'impulsion). Ce rapport cyclique indique la puissance que la borne peut fournir au chargeur. La largeur d'impulsion varie linéairement entre 100 us (6A fourni par la borne) et 800 us (48A). La position fermée le S2 indique que le chargeur du véhicule électrique peut recevoir de l'énergie. La fermeture de S2 entraîne la fermeture d'un contacteur du circuit puissance sur la borne de recharge (AC). (il n'apparait pas sur le schéma) Etat E : Quand le véhicule détecte que la batterie est chargée, S2 est ouvert. Le signal remonte à 9V/-12V, indiquant à la borne d'ouvrir le contacteur AC. Le voyant « charge » s'allume en Vert. Etat F : retour à 12V quand la prise est déconnectée. Le voyant « charge » s'éteint.Lors de la reprise du véhicule, le client s'identifie à nouveau sur la borne. Si le véhicule est encore en charge, il appuie sur le bouton « Stop », sinon il peut le récupérer directement après l'avoir débranché. Le contacteur AC doit être ouvert, la prise déverrouillée. Une fois la prise débranchée par le client, la trappe doit être verrouillée, voyant « Prise » éteint et le voyant « disponible » allumé. Implementation de la machine a etat fini correspondant au cycle charge

Definition at line 45 of file generateur_save.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

3.3.2.3 tension()

```
void int GenerateurSave::tension ( )
```

Recupere la tension sur le fil pilote.

Parameters

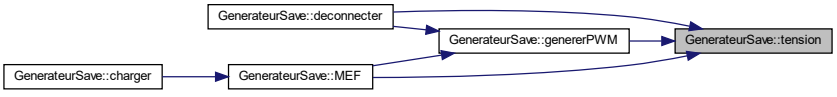
void	

Returns

int \Cette valeur de tension est modifier par le simulateur. Donne l'etat courant du systeme.

Definition at line 21 of file generateur_save.cpp.

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- generateur_save.h
- generateur_save.cpp

3.4 LecteurCarte Class Reference

```
#include <lecteurcarte.h>
```

Public Member Functions

- void **initialiser** ()
- void lire_carte ()

  Lecteur de carte.
- void reprise ()

  Permet deverrouiller la trappe pour que l'utilisateur reprenne son vehicule. Pour etre sur que le vehicule appartient a ce dernier, on verifie egalement que la carte inserer correspond a la derniere carte inserer.

3.4.1 Detailed Description

Author

SEWODA & ACHARI

Date

01 April 2021

Version

1.3

Definition at line 23 of file lecteurcarte.h.

3.4.2 Member Function Documentation

3.4.2.1 lire_carte()

```
void LecteurCarte::lire_carte ( )
```

Lecteur de carte.

Cette fonction represente le point d'entree de rechargement d'un vehicule. Elle fait appel a d'autre fonction pour :

authentifier() charger() reprise() deconnecter()

Definition at line 11 of file lecteurcarte.cpp.

The documentation for this class was generated from the following files:

- lecteurcarte.h
- lecteurcarte.cpp

3.5 Prise Class Reference

```
#include <prise.h>
```

Public Member Functions

- void initialiser ()

  Permet d'initialiser les acces memoire.
- void set_prise (led etat)

  Donne l'etat de la prise, si elle est libre ou connectee. Utilise les pointeurs pour acceder aux variables systemes.
- void verouiller_trappe ()

  Permet de voir l'etat de la prise: verouiller. Dans cette etat la voiture ne peut pas etre brancher. Utilise les pointeurs pour acceder aux variables systemes.
- void deverouiller_trappe ()

  Permet de voir l'etat de la prise: deverouiller. Dans cette etat la voiture peut etre brancher. Utilise les pointeurs pour acceder aux variables systemes.

3.5.1 Detailed Description

Author

SEWODA & ACHARI

Date

01 April 2021

Version

1.3

Definition at line 17 of file prise.h.

3.5.2 Member Function Documentation

3.5.2.1 deverouiller_trappe()

```
void Prise::deverouiller_trappe ( )
```

Permet de voir l'etat de la prise: deverouiller. Dans cette etat la voiture peut etre brancher. Utilise les pointeurs pour acceder aux variables systemes.

Parameters

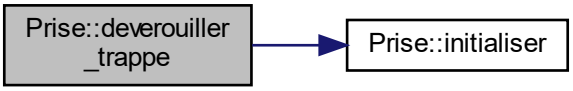
led	etat: allumee pour signifier libre

Returns

void: ne retourne pas de valeur

Definition at line 19 of file prise.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

3.5.2.2 initialiser()

```
void Prise::initialiser ( )
```

Permet d'initialiser les acces memoire.

Autorise l'acces a la memoire partagee. Appeler a chaque interaction avec la borne. Sinon affiche erreur d'acces

Parameters

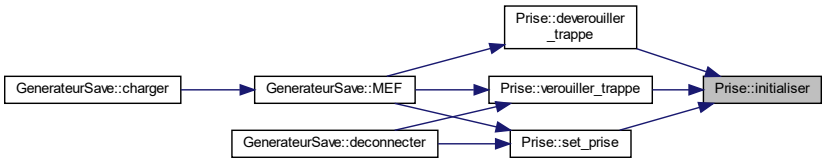
void	

Returns

void

Definition at line 4 of file prise.cpp.

Here is the caller graph for this function:



3.5.2.3 set_prise()

```
void Prise::set_prise (
            led etat )
```

Donne l'etat de la prise, si elle est libre ou connectee. Utilise les pointeurs pour acceder aux variables systemes.

Parameters

led	etat: allumee pour signifier libre

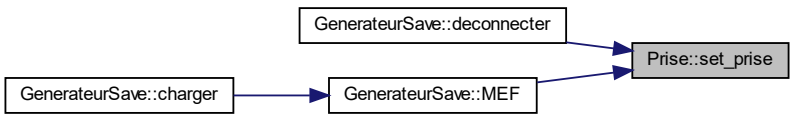
Returns

void: ne retourne pas de valeur

Definition at line 25 of file prise.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:



3.5.2.4 verouiller_trappe()

```
void Prise::verouiller_trappe ( )
```

Permet de voir l'etat de la prise: verouiller. Dans cette etat la voiture ne peut pas etre brancher. Utilise les pointeurs pour acceder aux variables systemes.

Parameters

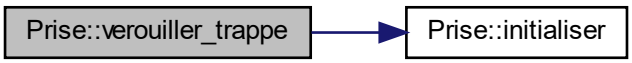
led	etat: allumee pour signifier libre

Returns

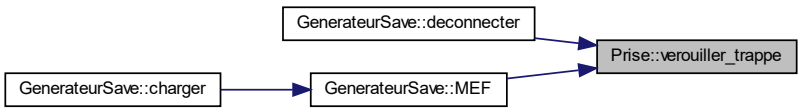
void: ne retourne pas de valeur

Definition at line 13 of file prise.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- prise.h
- prise.cpp

3.6 Timer Class Reference

Public Member Functions

- Timer ()

  represente le constructeur de la classe Timer.
- void initialiser ()

  Permet d'initialiser les acces memoires.
- void raz ()

  Permet d'initialiser le compteur de temps.
- int valeur ()

  Renvoi la valeur actuel du compteur temps systeme.
- void pause (int s)

  Permet de faire une pause de s secondes dans l'execution des taches par le systemes.

3.6.1 Detailed Description

Definition at line 19 of file timer.h.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 Timer()

```
void Timer::Timer ( )
```

represente le constructeur de la classe Timer.

Permet d'initialiser le timer.

Parameters

void	

Returns

void

Definition at line 5 of file timer.cpp.

3.6.3 Member Function Documentation

3.6.3.1 initialiser()

```
void Timer::initialiser ( )
```

Permet d'initialiser les acces memoires.

Autorise l'acces a la memoire partagee. Appeler a chaque interaction avec la borne. Sinon affiche erreur d'acces

Parameters

void	

Returns

void

Definition at line 6 of file timer.cpp.

3.6.3.2 pause()

```
void Timer::pause (
            int s )
```

Permet de faire une pause de s secondes dans l'execution des taches par le systemes.

Parameters

int	s

Returns

void

Definition at line 15 of file timer.cpp.

Here is the call graph for this function:

3.6.3.3 raz()

```
void Timer::raz ( )
```

Permet d'initialiser le compteur de temps.

Communique a travers un pointeur

Parameters

void	

Returns

void

Definition at line 23 of file timer.cpp.

3.6.3.4 valeur()

```
void Timer::valeur ( )
```

Renvoi la valeur actuel du compteur temps systeme.

Parameters

void	

Returns

void

Definition at line 28 of file timer.cpp.

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- timer.h
- timer.cpp

3.7 Voyants Class Reference

Public Member Functions

- void initialiser ()

  Permet de configurer l'acces a la memoire partagee a l'aide la variable shmid (share memory id) pour permettre de communiquer avec la carte. Un message d'erreur s'affiche en cas de probleme d'acces.
- void set_charge (led ledCoul)

  Allume le voyant charge quand le vehicule charge.
- void set_dispo (led ledCoul)

  Allume le voyant disponible a la fin de la charge.
- void blink_charge ()

  Clignoter la led charge pendant 8 seconde.
- void set_defaut ()

  Allume la LED defaut de la borne.
- int **dispo** ()
- void blink_defaut ()

  clignoter la LED defaut pendant 8 secondes
- int **getStatus_bouton** ()

3.7.1 Detailed Description

Definition at line 20 of file voyants.h.

3.7.2 Member Function Documentation

3.7.2.1 blink_charge()

```
void Voyants::blink_charge ( )
```

Clignoter la led charge pendant 8 seconde.

Parameters

int	ledCoul: qui represente la couleur a afficher

Returns

void

Definition at line 27 of file voyants.cpp.

3.7.2.2 initialiser()

```
void Voyants::initialiser ( )
```

Permet de configurer l'acces a la memoire partagee a l'aide la variable shmid (share memory id) pour permettre de communiquer avec la carte. Un message d'erreur s'affiche en cas de probleme d'acces.

Parameters

void	

Returns

void

Definition at line 9 of file voyants.cpp.

3.7.2.3 set_charge()

```
void Voyants::set_charge (
            led ledCoul )
```

Allume le voyant charge quand le vehicule charge.

Parameters

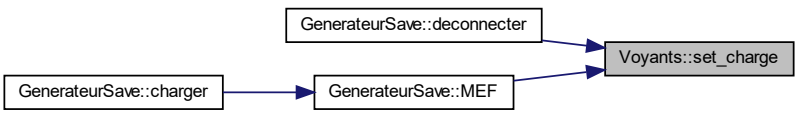
int	ledCoul: qui represente la couleur a afficher

Returns

void

Definition at line 18 of file voyants.cpp.

Here is the caller graph for this function:



3.7.2.4 set_defaut()

```
void Voyants::set_defaut ( )
```

Allume la LED defaut de la borne.

Parameters

int	ledCoul: qui represente la couleur a afficher

Returns

void

Definition at line 46 of file voyants.cpp.

3.7.2.5 set_dispo()

```
void Voyants::set_dispo (
            led ledCoul )
```

Allume le voyant disponible a la fin de la charge.

Parameters

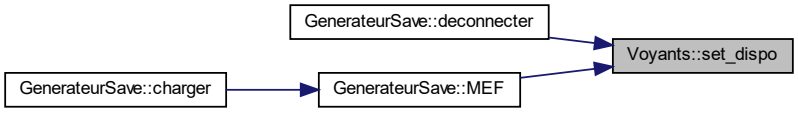
int	ledCoul: qui represente la couleur a afficher

Returns

void

Definition at line 23 of file voyants.cpp.

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- voyants.h
- voyants.cpp

Chapter 4

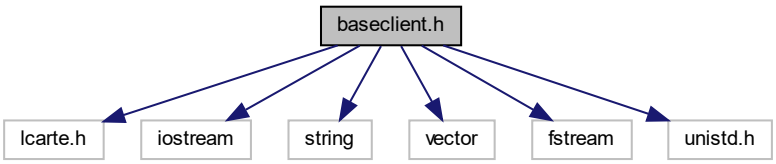
File Documentation

4.1 baseclient.h File Reference

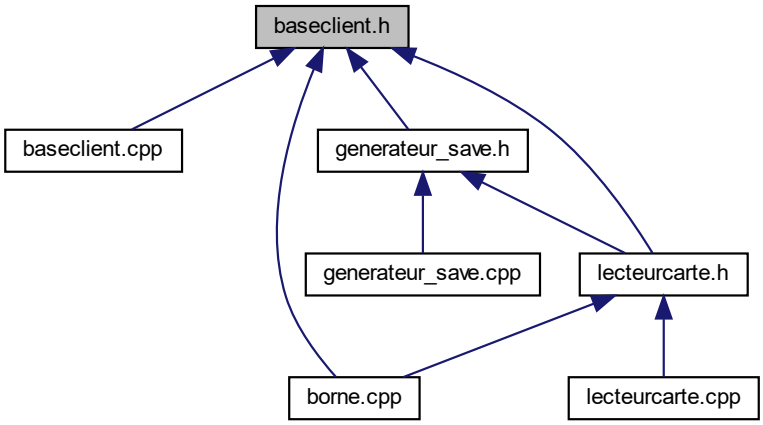
Permet d'enregistrer les clients.

```
#include <lcarte.h>
#include <iostream>
#include <string>
#include <vector>
#include <fstream>
#include <unistd.h>
```

Include dependency graph for baseclient.h:



This graph shows which files directly or indirectly include this file:



Classes

- class BaseClient

4.1.1 Detailed Description

Permet d'enregistrer les clients.

Author

Sewoda et Achari

Version

1.0

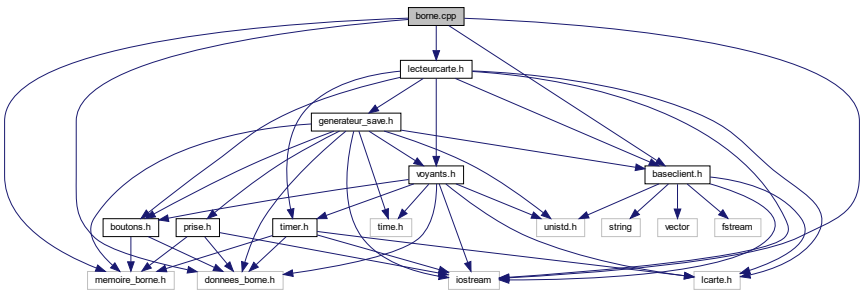
4.2 borne.cpp File Reference

Fonction principale de la borne Cette fonction principale fait appel a la fonction lecteur carte qui demarre un cycle de rechargement. Le circuit de recharge dédié et imposé dans le « Mode 3 » est défini dans la proposition de norme IEC 61851-1 « ELECTRIC VEHICLE CONDUCTIVE CHARGING SYSTEM ». Cela permet de garantir une sécurité maximale des utilisateurs lors de la recharge de leur véhicule électrique. La Figure 3 représente la connectique entre une borne et un véhicule.

```
#include <iostream>
#include <memoire_borne.h>
#include <donnees_borne.h>
#include "lecteurcarte.h"
```

```
#include "baseclient.h"
```

Include dependency graph for borne.cpp:



Macros

- #define **ADMIN** 1234

Functions

- int **main** ()

4.2.1 Detailed Description

Fonction principale de la borne Cette fonction principale fait appel a la fonction lecteur carte qui demarre un cycle de rechargement. Le circuit de recharge dédié et imposé dans le « Mode 3 » est défini dans la proposition de norme IEC 61851-1 « ELECTRIC VEHICLE CONDUCTIVE CHARGING SYSTEM ». Cela permet de garantir une sécurité maximale des utilisateurs lors de la recharge de leur véhicule électrique. La Figure 3 représente la connectique entre une borne et un véhicule.

Code admin =1234

Author

Sewoda et Achari

Version

1.0

Date

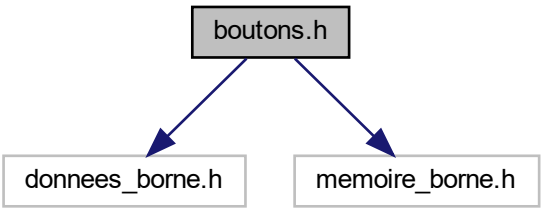
11/04/2021

4.3 boutons.h File Reference

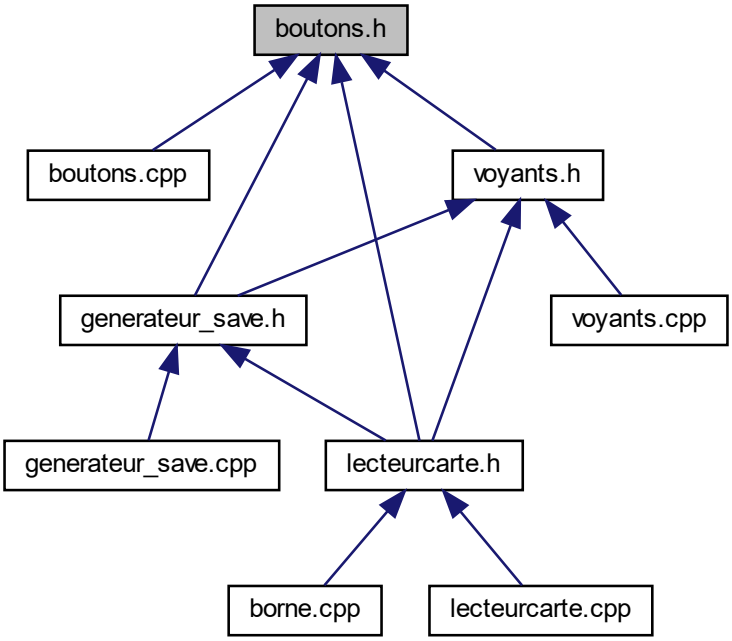
Cette classe permet de gerer les appuis sur les boutons a travers l'utilisation des pointeurs et des acces memoire.

```
#include <donnees_borne.h>
#include <memoire_borne.h>
```

Include dependency graph for boutons.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Boutons

4.3.1 Detailed Description

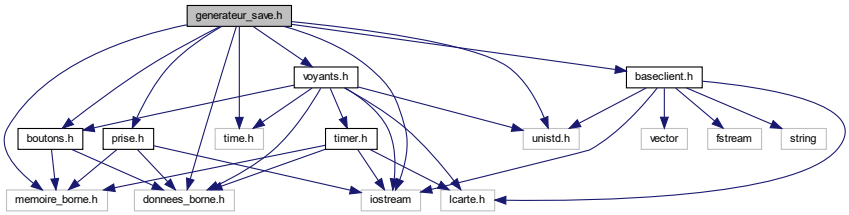
Cette classe permet de gerer les appuis sur les boutons a travers l'utilisation des pointeurs et des acces memoire.

4.4 generateur_save.h File Reference

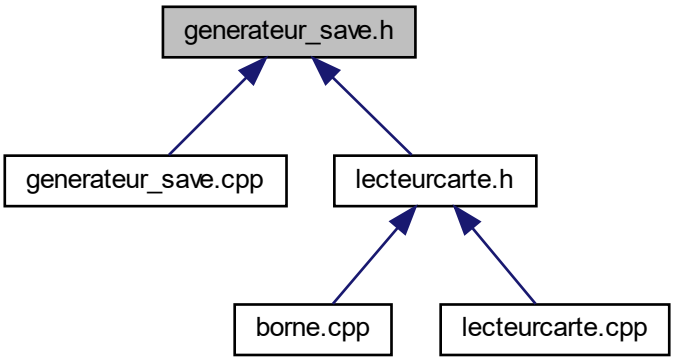
Cette classe permet de gerer les appuis sur les boutons a travers l'utilisation des pointeurs et des acces memoire.

```
#include <iostream>
#include <donnees_borne.h>
#include <memoire_borne.h>
#include <time.h>
#include <unistd.h>
#include "prise.h"
#include "boutons.h"
#include "baseclient.h"
#include "voyants.h"
```

Include dependency graph for generateur_save.h:



This graph shows which files directly or indirectly include this file:



Classes

- class GenerateurSave

4.4.1 Detailed Description

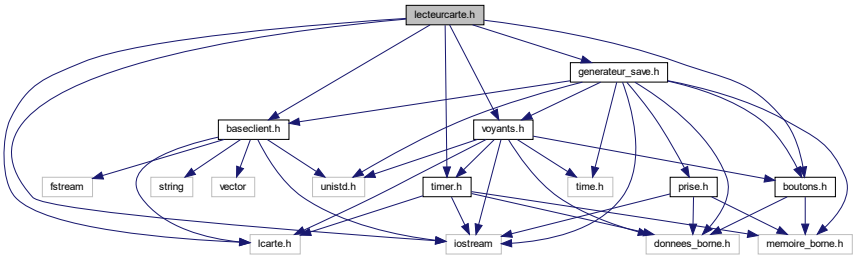
Cette classe permet de gerer les appuis sur les boutons a travers l'utilisation des pointeurs et des acces memoire.

4.5 lecteurcarte.h File Reference

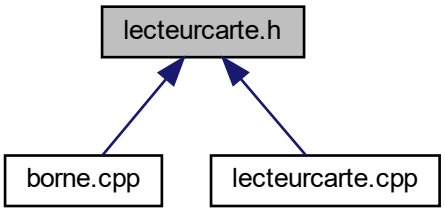
Cette classe permet d'interagir avec la borne avec la carte client la borne de recharge.

```
#include <lcarte.h>
#include <iostream>
#include "voyants.h"
#include "timer.h"
#include "boutons.h"
#include "baseclient.h"
#include "generateur_save.h"
```

Include dependency graph for lecteurcarte.h:



This graph shows which files directly or indirectly include this file:



Classes

- class LecteurCarte

4.5.1 Detailed Description

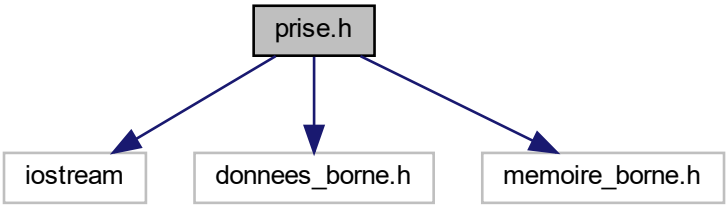
Cette classe permet d'interagir avec la borne avec la carte client la borne de recharge.

4.6 prise.h File Reference

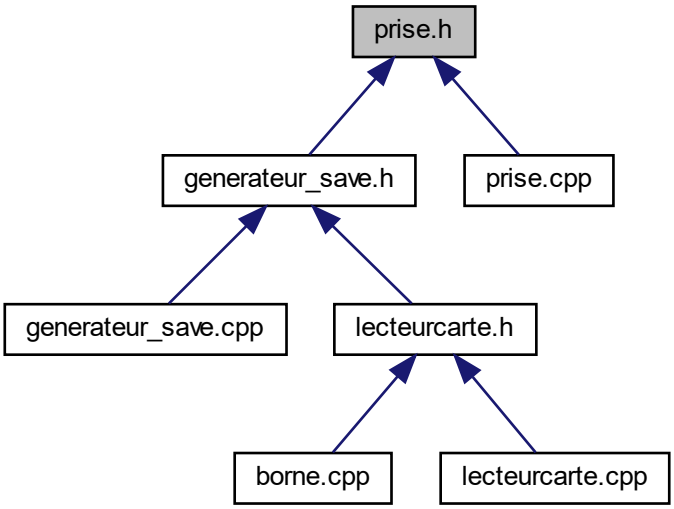
Cette classe gere les etats de la prise de la borne de recharge.

```
#include <iostream>
#include <donnees_borne.h>
#include <memoire_borne.h>
```

Include dependency graph for prise.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Prise

4.6.1 Detailed Description

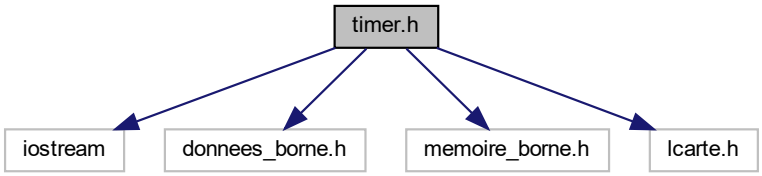
Cette classe gere les etats de la prise de la borne de recharge.

4.7 timer.h File Reference

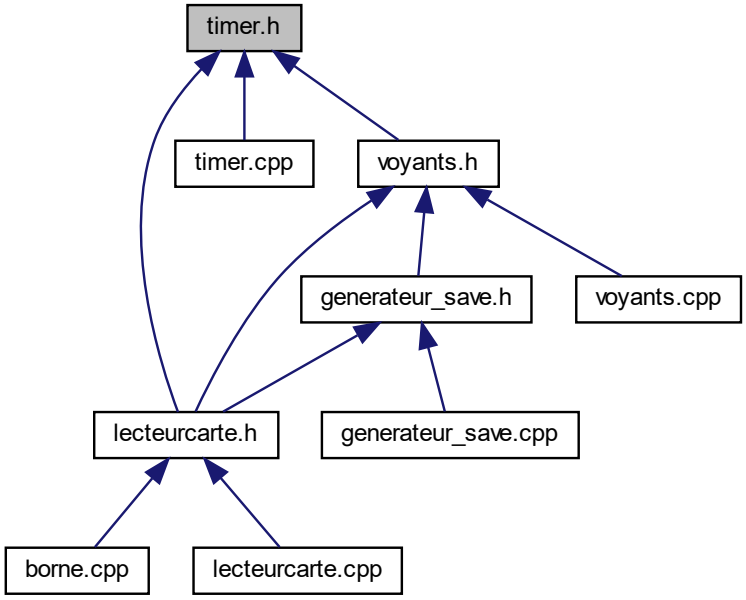
Cette classe permet de creer des methodes pour la gestion du temps dans le systeme de recharge de vehicule.

```
#include <iostream>
#include <donnees_borne.h>
#include <memoire_borne.h>
#include <lcarte.h>
```

Include dependency graph for timer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Timer

4.7.1 Detailed Description

Cette classe permet de creer des methodes pour la gestion du temps dans le systeme de recharge de vehicule.

Author

Sewoda et Achari

Version

1.0

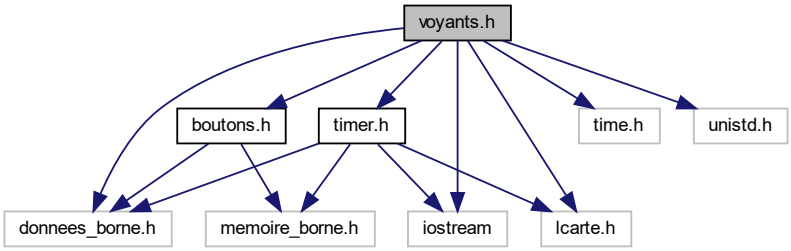
4.8 voyants.h File Reference

Gere les voyants de la borne.

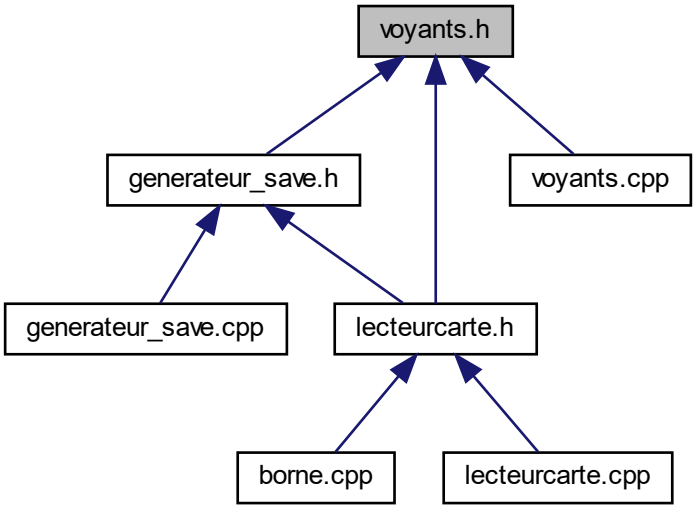
```
#include <donnees_borne.h>
#include <iostream>
#include <lcarte.h>
```

```
#include "timer.h"
#include "time.h"
#include "boutons.h"
#include "unistd.h"
```

Include dependency graph for voyants.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Voyants

4.8.1 Detailed Description

Gere les voyants de la borne.

Author

Sewoda Pascal and Achari Kahina

Version

1.0

Index