Arduino Remote Library

Generated by Doxygen 1.8.9.1

Wed Aug 19 2015 01:07:25

Contents

1 Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

2 Class Index

2.1 Class List

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

3 File Index

3.1 File List

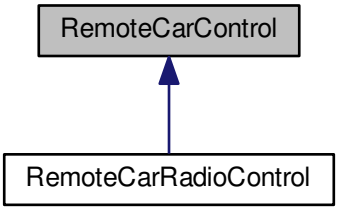
Here is a list of all files with brief descriptions:

4 Class Documentation

4.1 RemoteCarControl Class Reference

```
#include <RemoteCarControl.h>
```

Inheritance diagram for RemoteCarControl:



Public Member Functions

- virtual void apply ()=0
- void setThrottle (int throttle)
- void stop ()
- void forward ()
- void backward ()
- void turnLeft ()
- void turnRight ()

Protected Attributes

- char isForward
- int throttle
- int leftWheel
- int rightWheel

4.1.1 Detailed Description

Arduino - Remote Car Control.

RemoteCarControl.h

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file RemoteCarControl.h.

4.1.2 Member Function Documentation

4.1.2.1 virtual void RemoteCarControl::apply ( ) [pure virtual]

Implemented in RemoteCarRadioControl.

4.1.2.2 void RemoteCarControl::backward ( )

Definition at line 54 of file RemoteCarControl.cpp.

4.1.2.3 void RemoteCarControl::forward ( )

Definition at line 47 of file RemoteCarControl.cpp.

4.1.2.4 void RemoteCarControl::setThrottle (int *throttle*)

Definition at line 14 of file RemoteCarControl.cpp.

4.1.2.5 void RemoteCarControl::stop ( )

Definition at line 41 of file RemoteCarControl.cpp.

4.1.2.6 void RemoteCarControl::turnLeft ( )

Definition at line 19 of file RemoteCarControl.cpp.

4.1.2.7 void RemoteCarControl::turnRight ( )

Definition at line 30 of file RemoteCarControl.cpp.

4.1.3 Member Data Documentation

4.1.3.1 char RemoteCarControl::isForward [protected]

Definition at line 16 of file RemoteCarControl.h.

4.1.3.2 int RemoteCarControl::leftWheel [protected]

Definition at line 18 of file RemoteCarControl.h.

4.1.3.3 int RemoteCarControl::rightWheel [protected]

Definition at line 19 of file RemoteCarControl.h.

4.1.3.4 int RemoteCarControl::throttle [protected]

Definition at line 17 of file RemoteCarControl.h.

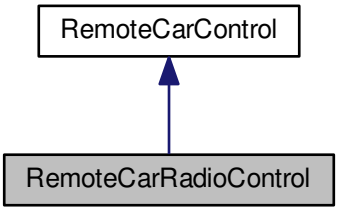
The documentation for this class was generated from the following files:

- RemoteCarControl.h
- RemoteCarControl.cpp

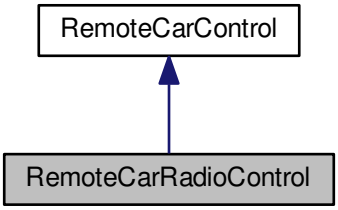
4.2 RemoteCarRadioControl Class Reference

```
#include <RemoteCarRadioControl.h>
```

Inheritance diagram for RemoteCarRadioControl:



Collaboration diagram for RemoteCarRadioControl:



Public Member Functions

- RemoteCarRadioControl (SerialRadioFrequency *serialRadioFrequency, int leftForwardPin, int leftBackwardPin, int rightForwardPin, int rightBackwardPin)
- virtual void apply ()
- void parseCommand (char *cmdBuf, char cmdBufPointer)
- void loop ()

Private Attributes

- SerialRadioFrequency * serialRadioFrequency
- int leftForwardPin
- int leftBackwardPin
- int rightForwardPin
- int rightBackwardPin

Additional Inherited Members

4.2.1 Detailed Description

Arduino - Remote Car Radio Control.

RemoteCarRadioControl.h

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 15 of file RemoteCarRadioControl.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 RemoteCarRadioControl::RemoteCarRadioControl (SerialRadioFrequency * *serialRadioFrequency,* int *leftForwardPin,* int *leftBackwardPin,* int *rightForwardPin,* int *rightBackwardPin*)

Definition at line 14 of file RemoteCarRadioControl.cpp.

4.2.3 Member Function Documentation

4.2.3.1 void RemoteCarRadioControl::apply ( ) [virtual]

Implements RemoteCarControl.

Definition at line 85 of file RemoteCarRadioControl.cpp.

4.2.3.2 void RemoteCarRadioControl::loop ( )

Definition at line 29 of file RemoteCarRadioControl.cpp.

4.2.3.3 void RemoteCarRadioControl::parseCommand (char * *cmdBuf,* char *cmdBufPointer*)

Definition at line 49 of file RemoteCarRadioControl.cpp.

4.2.4 Member Data Documentation

4.2.4.1 int RemoteCarRadioControl::leftBackwardPin [private]

Definition at line 19 of file RemoteCarRadioControl.h.

4.2.4.2 int RemoteCarRadioControl::leftForwardPin [private]

Definition at line 18 of file RemoteCarRadioControl.h.

4.2.4.3 int RemoteCarRadioControl::rightBackwardPin [private]

Definition at line 21 of file RemoteCarRadioControl.h.

4.2.4.4 int RemoteCarRadioControl::rightForwardPin [private]

Definition at line 20 of file RemoteCarRadioControl.h.

4.2.4.5 SerialRadioFrequency* RemoteCarRadioControl::serialRadioFrequency [private]

Definition at line 17 of file RemoteCarRadioControl.h.

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

- RemoteCarRadioControl.h
- RemoteCarRadioControl.cpp

4.3 RemoteFlightControl Class Reference

```
#include <RemoteFlightControl.h>
```

Private Member Functions

- virtual void setThrottle (int throttle)=0
- virtual void setPitch (int pitch)=0
- virtual void setRoll (int roll)=0
- virtual void setYaw (int yaw)=0

4.3.1 Detailed Description

Arduino - Remote Flight Control.

RemoteFlightControl.h

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file RemoteFlightControl.h.

4.3.2 Member Function Documentation

4.3.2.1 virtual void RemoteFlightControl::setPitch (int *pitch*) [private], [pure virtual]

4.3.2.2 virtual void RemoteFlightControl::setRoll (int *roll*) [private], [pure virtual]

4.3.2.3 virtual void RemoteFlightControl::setThrottle (int *throttle*) [private], [pure virtual]

4.3.2.4 virtual void RemoteFlightControl::setYaw (int *yaw*) [private], [pure virtual]

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

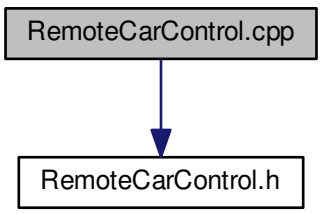
- RemoteFlightControl.h

5 File Documentation

5.1 RemoteCarControl.cpp File Reference

```
#include "RemoteCarControl.h"
```

Include dependency graph for RemoteCarControl.cpp:



Macros

- #define __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_CPP__ 1

5.1.1 Macro Definition Documentation

5.1.1.1 #define __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_CPP__ 1

Arduino - Remote Car Control.

RemoteCarControl.cpp

Author

Dalmir da Silva dalmirdasilva@gmail.com

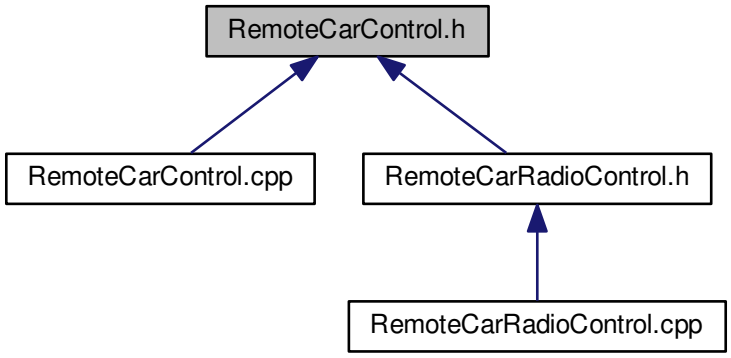
Definition at line 10 of file RemoteCarControl.cpp.

5.2 RemoteCarControl.cpp

```
00001
00009 #ifndef __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_CPP__
00010 #define __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_CPP__ 1
00011
00012 #include "RemoteCarControl.h"
00013
00014 void RemoteCarControl::setThrottle(int throttle) {
00015     this->throttle = throttle;
00016     this->apply();
00017 }
00018
00019 void RemoteCarControl::turnLeft() {
00020     if (this->isForward) {
00021         this->leftWheel = throttle;
00022         this->rightWheel = 0;
00023     } else {
00024         this->leftWheel = 0;
00025         this->rightWheel = -(throttle);
00026     }
00027     this->apply();
00028 }
00029
00030 void RemoteCarControl::turnRight() {
00031     if (this->isForward) {
00032         this->leftWheel = 0;
00033         this->rightWheel = throttle;
00034     } else {
00035         this->leftWheel = -(throttle);
00036         this->rightWheel = 0;
00037     }
00038     this->apply();
00039 }
00040
00041 void RemoteCarControl::stop() {
00042     this->leftWheel = 0;
00043     this->rightWheel = 0;
00044     this->apply();
00045 }
00046
00047 void RemoteCarControl::forward() {
00048     this->leftWheel = throttle;
00049     this->rightWheel = throttle;
00050     this->isForward = 1;
00051     this->apply();
00052 }
00053
00054 void RemoteCarControl::backward() {
00055     this->leftWheel = -(throttle);
00056     this->rightWheel = -(throttle);
00057     this->isForward = 0;
00058     this->apply();
00059 }
00060
00061 #endif /* __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_CPP__ */
```

5.3 RemoteCarControl.h File Reference

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



Classes

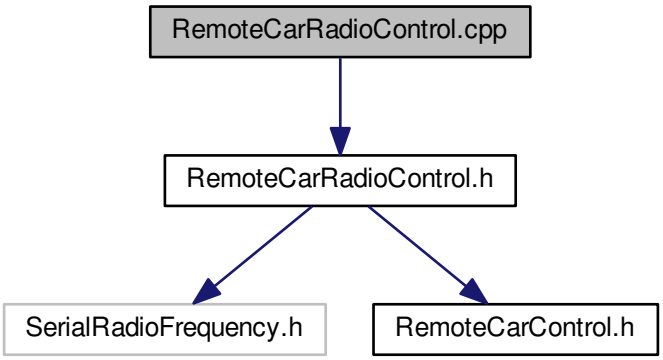
- class RemoteCarControl

5.4 RemoteCarControl.h

```
00001
00009 #ifndef __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_H__
00010 #define __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_H__ 1
00011
00012 class RemoteCarControl {
00013
00014 protected:
00015
00016   char isForward;
00017   int throttle;
00018   int leftWheel;
00019   int rightWheel;
00020
00021 public:
00022
00023   virtual void apply() = 0;
00024
00025   void setThrottle(int throttle);
00026
00027   void stop();
00028
00029   void forward();
00030
00031   void backward();
00032
00033   void turnLeft();
00034
00035   void turnRight();
00036 };
00037
00038 #endif /* __ARDUINO_LIBRARY_REMOTE_CAR_CONTROL_H__ */
```

5.5 RemoteCarRadioControl.cpp File Reference

`#include "RemoteCarRadioControl.h"`
Include dependency graph for RemoteCarRadioControl.cpp:



Macros

- #define __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_CPP__ 1

5.5.1 Macro Definition Documentation

5.5.1.1 #define __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_CPP__ 1

Arduino - Remote Car Radio Control.

RemoteCarRadioControl.cpp

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 10 of file RemoteCarRadioControl.cpp.

5.6 RemoteCarRadioControl.cpp

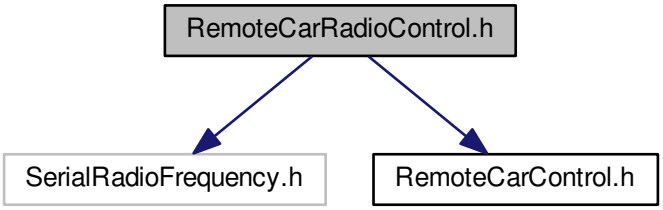
```
00001
00009 #ifndef __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_CPP__
00010 #define __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_CPP__ 1
00011
00012 #include "RemoteCarRadioControl.h"
00013
00014 RemoteCarRadioControl::RemoteCarRadioControl(
      SerialRadioFrequency* serialRadioFrequency,
00015       int leftForwardPin,
00016       int leftBackwardPin,
00017       int rightForwardPin,
00018       int rightBackwardPin) :
00019     RemoteCarControl(),
00020     serialRadioFrequency(serialRadioFrequency) {
00021       this->leftForwardPin = leftForwardPin;
00022       this->leftBackwardPin = leftBackwardPin;
00023       this->rightForwardPin = rightForwardPin;
00024       this->rightBackwardPin = rightBackwardPin;
```

```
00025       this->throttle = 200;
00026
00027 }
00028
00029 void RemoteCarRadioControl::loop() {
00030     char cmdBuf[8];
00031     unsigned char cmdBufPointer = 0;
00032     char endCommandMark = ';';
00033     while(true) {
00034         if (this->serialRadioFrequency->available()) {
00035             char c = this->serialRadioFrequency->read();
00036             serialRadioFrequency->write(c);
00037             if (c == endCommandMark || cmdBufPointer >= 8) {
00038                 if (cmdBufPointer > 1) {
00039                   parseCommand(cmdBuf, cmdBufPointer);
00040                 }
00041                 cmdBufPointer = 0;
00042             } else {
00043                 cmdBuf[cmdBufPointer++] = c;
00044             }
00045         }
00046     }
00047 }
00048
00049 void RemoteCarRadioControl::parseCommand(char* cmdBuf, char
    cmdBufPointer) {
00050     char action = cmdBuf[cmdBufPointer - 1];
00051     Serial.println(action);
00052     switch(action) {
00053         case 'B':
00054             serialRadioFrequency->println("backward");
00055             this->backward();
00056         break;
00057         case 'F':
00058             serialRadioFrequency->println("forward");
00059             this->forward();
00060         break;
00061         case 'S':
00062             serialRadioFrequency->println("stop");
00063             this->stop();
00064         break;
00065         case 'L':
00066             serialRadioFrequency->println("turnLeft");
00067             this->turnLeft();
00068         break;
00069         case 'R':
00070             serialRadioFrequency->println("turnRight");
00071             this->turnRight();
00072         break;
00073         /*
00074         case 'T':
00075             serialRadioFrequency->println("setThrottle");
00076             char throttle = cmdBuf[1];
00077             this->setThrottle(((int) throttle) | 0xff);
00078         break;
00079         */
00080         default:
00081             serialRadioFrequency->println("not recognized");
00082     }
00083 }
00084
00085 void RemoteCarRadioControl::apply() {
00086     if (this->leftWheel >= 0) {
00087         analogWrite(this->leftForwardPin, this->leftWheel);
00088         analogWrite(this->leftBackwardPin, 0);
00089     } else {
00090         analogWrite(this->leftForwardPin, 0);
00091         analogWrite(this->leftBackwardPin, abs(this->leftWheel));
00092     }
00093     if (this->rightWheel >= 0) {
00094         analogWrite(this->rightForwardPin, this->rightWheel);
00095         analogWrite(this->rightBackwardPin, 0);
00096     } else {
00097         analogWrite(this->rightForwardPin, 0);
00098         analogWrite(this->rightBackwardPin, abs(this->rightWheel));
00099     }
00100 }
00101
00102 #endif /* __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_CPP__ */
```

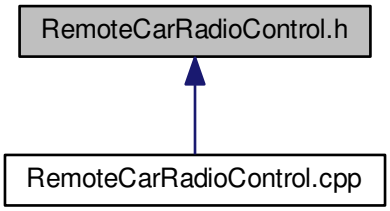
5.7 RemoteCarRadioControl.h File Reference

```
#include "SerialRadioFrequency.h"
```

`#include "RemoteCarControl.h"`
Include dependency graph for RemoteCarRadioControl.h:



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



Classes

- class RemoteCarRadioControl

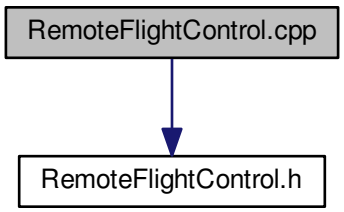
5.8 RemoteCarRadioControl.h

```
00001
00009 #ifndef __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_H__
00010 #define __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_H__ 1
00011
00012 #include "SerialRadioFrequency.h"
00013 #include "RemoteCarControl.h"
00014
00015 class RemoteCarRadioControl : public RemoteCarControl {
00016
00017     SerialRadioFrequency* serialRadioFrequency;
00018     int leftForwardPin;
00019     int leftBackwardPin;
00020     int rightForwardPin;
00021     int rightBackwardPin;
00022
00023 public:
00024
00025     RemoteCarRadioControl(SerialRadioFrequency* serialRadioFrequency,
00026         int leftForwardPin,
00027         int leftBackwardPin,
00028         int rightForwardPin,
00029         int rightBackwardPin);
00030
```

```
00031     virtual void apply();
00032
00033     void parseCommand(char* cmdBuf, char cmdBufPointer);
00034
00035     void loop();
00036 };
00037
00038 #endif /* __ARDUINO_LIBRARY_REMOTE_CAR_RADIO_CONTROL_H__ */
```

5.9 RemoteFlightControl.cpp File Reference

`#include "RemoteFlightControl.h"`
Include dependency graph for RemoteFlightControl.cpp:



Macros

- #define __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_CPP__ 1

5.9.1 Macro Definition Documentation

5.9.1.1 #define __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_CPP__ 1

Arduino - Remote Flight Control.

RemoteFlightControl.cpp

Author

Dalmir da Silva dalmirdasilva@gmail.com

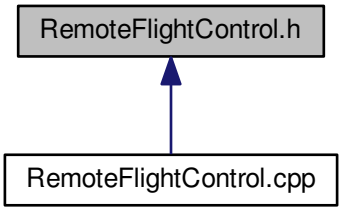
Definition at line 10 of file RemoteFlightControl.cpp.

5.10 RemoteFlightControl.cpp

```
00001
00009 #ifndef __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_CPP__
00010 #define __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_CPP__ 1
00011
00012 #include "RemoteFlightControl.h"
00013
00014 RemoteFlightControl::RemoteFlightControl() {}
00015
00016 #endif /* __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_CPP__ */
```

5.11 RemoteFlightControl.h File Reference

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



Classes

- class RemoteFlightControl

5.12 RemoteFlightControl.h

```
00001
00009 #ifndef __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_H__
00010 #define __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_H__ 1
00011
00012 class RemoteFlightControl {
00013
00014   virtual void setThrottle(int throttle) = 0;
00015
00016   virtual void setPitch(int pitch) = 0;
00017
00018   virtual void setRoll(int roll) = 0;
00019
00020   virtual void setYaw(int yaw) = 0;
00021 };
00022
00023 #endif /* __ARDUINO_LIBRARY_REMOTE_FLIGHT_CONTROL_H__ */
```

Index