

Digital Cluster

Generated by Doxygen 1.9.1

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 System Class Reference	7
4.1.1 Detailed Description	9
4.1.2 Constructor & Destructor Documentation	9
4.1.2.1 System()	9
4.1.3 Member Function Documentation	9
4.1.3.1 batteryPer()	10
4.1.3.2 brakeLight()	10
4.1.3.3 emergencyLights()	10
4.1.3.4 headLights()	10
4.1.3.5 setBatteryPer()	10
4.1.3.6 setBrakeLight()	11
4.1.3.7 setEmergencyLights()	11
4.1.3.8 setHeadLights()	12
4.1.3.9 setSpeed()	12
4.1.3.10 setTotalDistance()	13
4.1.3.11 setTurnLightLeft()	13
4.1.3.12 setTurnLightRight()	14
4.1.3.13 speed()	14
4.1.3.14 totalDistance()	15
4.1.3.15 turnLightLeft()	15
4.1.3.16 turnLightRight()	15
4.1.4 Property Documentation	15
4.1.4.1 batteryPer	15
4.1.4.2 brakeLight	16
4.1.4.3 emergencyLights	16
4.1.4.4 headLights	17
4.1.4.5 speed	17
4.1.4.6 totalDistance	17
4.1.4.7 turnLightLeft	18
4.1.4.8 turnLightRight	18
4.2 ZMQReader Class Reference	19
4.2.1 Detailed Description	20
4.2.2 Constructor & Destructor Documentation	20

4.2.2.1 ZMQReader()	20
4.2.2.2 ~ZMQReader()	20
4.2.3 Member Function Documentation	21
4.2.3.1 batteryReceived	21
4.2.3.2 brakeLightReceived	21
4.2.3.3 emergencyLightsReceived	22
4.2.3.4 headLightsReceived	22
4.2.3.5 run()	23
4.2.3.6 speedReceived	23
4.2.3.7 stop()	24
4.2.3.8 totalDistanceReceived	24
4.2.3.9 turnLightLeftReceived	25
4.2.3.10 turnLightRightReceived	25
5 File Documentation	27
5.1 Digital_Cluster/main.cpp File Reference	27
5.1.1 Detailed Description	27
5.1.2 Function Documentation	28
5.1.2.1 main()	28
5.2 Digital_Cluster/System.cpp File Reference	29
5.2.1 Detailed Description	30
5.3 Digital_Cluster/System.h File Reference	30
5.3.1 Detailed Description	31
5.4 Digital_Cluster/zmqreader.cpp File Reference	31
5.4.1 Detailed Description	32
5.5 Digital_Cluster/zmqreader.h File Reference	32
5.5.1 Detailed Description	32
Index	33

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

QObject	
System	7
QThread	
ZMQReader	19

Chapter 2

Class Index

2.1 Class List

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

System	Manages vehicle properties and states for QML integration	7
ZMQReader	Class responsible for receiving data via ZeroMQ in a separate thread	19

Chapter 3

File Index

3.1 File List

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

Digital_Cluster/ main.cpp	
Entry point for the Digital Cluster application	27
Digital_Cluster/ System.cpp	
Implementation of the System class	29
Digital_Cluster/ System.h	
Declaration of the System class	30
Digital_Cluster/ zmqreader.cpp	
Implementation of the ZMQReader class for reading messages via ZeroMQ	31
Digital_Cluster/ zmqreader.h	
Declares the ZMQReader class for reading messages via ZeroMQ	32

Chapter 4

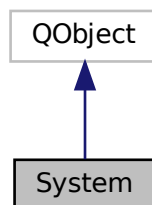
Class Documentation

4.1 System Class Reference

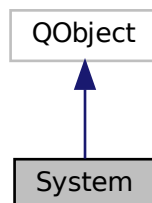
The [System](#) class manages vehicle properties and states for QML integration.

```
#include <System.h>
```

Inheritance diagram for System:



Collaboration diagram for System:



Signals

- void [speedChanged](#) ()
Emitted when vehicle speed changes.
- void [batteryPerChanged](#) ()
Emitted when battery percentage changes.
- void [headLightsChanged](#) ()
Emitted when headlights status changes.
- void [brakeLightChanged](#) ()
Emitted when brake light status changes.
- void [turnLightLeftChanged](#) ()
Emitted when left turn signal status changes.
- void [turnLightRightChanged](#) ()
Emitted when right turn signal status changes.
- void [emergencyLightsChanged](#) ()
Emitted when emergency lights status changes.
- void [totalDistanceChanged](#) ()
Emitted when total distance changes.

Public Member Functions

- [System](#) (QObject *parent=nullptr)
Constructs a new [System](#) object.
- QString [speed](#) () const
Gets current vehicle speed.
- void [setSpeed](#) (const QString &newSpeed)
Sets new vehicle speed.
- QString [batteryPer](#) () const
Gets current battery percentage.
- void [setBatteryPer](#) (const QString &newBatteryPer)
Sets new battery percentage.
- QString [headLights](#) () const
Gets current headlights status.
- void [setHeadLights](#) (const QString &newHeadLights)
Sets headlights status.
- QString [brakeLight](#) () const
Gets brake light status.
- void [setBrakeLight](#) (const QString &newBrakeLight)
Sets brake light status.
- QString [turnLightLeft](#) () const
Gets left turn signal status.
- void [setTurnLightLeft](#) (const QString &newTurnLightLeft)
Sets left turn signal status.
- QString [turnLightRight](#) () const
Gets right turn signal status.
- void [setTurnLightRight](#) (const QString &newTurnLightRight)
Sets right turn signal status.
- QString [emergencyLights](#) () const
Gets emergency lights status.
- void [setEmergencyLights](#) (const QString &newEmergencyLights)
Sets emergency lights status.
- QString [totalDistance](#) () const
Gets total distance traveled.
- void [setTotalDistance](#) (const QString &newTotalDistance)
Sets total distance traveled.

Properties

- QString [speed](#)
Current vehicle speed in kilometers per hour (km/h).
- QString [batteryPer](#)
Current battery charge percentage (0-100%).
- QString [headLights](#)
Headlights status.
- QString [brakeLight](#)
Brake light status.
- QString [turnLightLeft](#)
Left turn signal status.
- QString [turnLightRight](#)
Right turn signal status.
- QString [emergencyLights](#)
Emergency/hazard lights status.
- QString [totalDistance](#)
Total distance traveled in kilometers (km).

4.1.1 Detailed Description

The [System](#) class manages vehicle properties and states for QML integration.

This class provides Qt properties and signals to monitor and control various vehicle metrics including speed, battery level, lighting systems, and odometer readings. All properties are bindable from QML and thread-safe.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 System()

```
System::System (
    QObject * parent = nullptr ) [explicit]
```

Constructs a new [System](#) object.

Parameters

<i>parent</i>	Parent QObject (optional)
---------------	---------------------------

4.1.3 Member Function Documentation

4.1.3.1 batteryPer()

```
QString System::batteryPer ( ) const
```

Gets current battery percentage.

Returns

QString Battery percentage (0-100%)

4.1.3.2 brakeLight()

```
QString System::brakeLight ( ) const
```

Gets brake light status.

Returns

QString "true" if active, "false" if inactive

4.1.3.3 emergencyLights()

```
QString System::emergencyLights ( ) const
```

Gets emergency lights status.

Returns

QString "true" if on, "false" if off

4.1.3.4 headLights()

```
QString System::headLights ( ) const
```

Gets current headlights status.

Gets headlights status.

Returns

QString "true" if on, "false" if off

4.1.3.5 setBatteryPer()

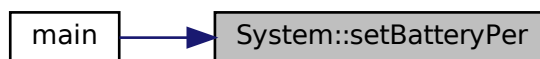
```
void System::setBatteryPer (
    const QString & newBatteryPer )
```

Sets new battery percentage.

Parameters

<i>newBatteryPer</i>	Battery percentage value (0-100%)
----------------------	-----------------------------------

Here is the caller graph for this function:



4.1.3.6 setBrakeLight()

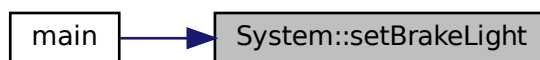
```
void System::setBrakeLight (
    const QString & newBrakeLight )
```

Sets brake light status.

Parameters

<i>newBrakeLight</i>	"true" to activate, "false" to deactivate
----------------------	---

Here is the caller graph for this function:



4.1.3.7 setEmergencyLights()

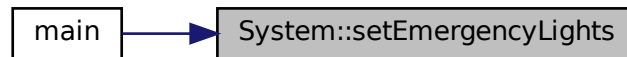
```
void System::setEmergencyLights (
    const QString & newEmergencyLights )
```

Sets emergency lights status.

Parameters

<i>newEmergencyLights</i>	"true" to enable, "false" to disable
---------------------------	--------------------------------------

Here is the caller graph for this function:

**4.1.3.8 setHeadLights()**

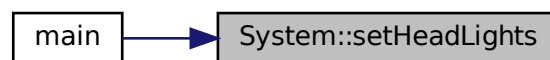
```
void System::setHeadLights (
    const QString & newHeadLights )
```

Sets headlights status.

Parameters

<i>newHeadLights</i>	"true" to enable, "false" to disable
----------------------	--------------------------------------

Here is the caller graph for this function:

**4.1.3.9 setSpeed()**

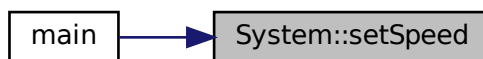
```
void System::setSpeed (
    const QString & newSpeed )
```

Sets new vehicle speed.

Parameters

<i>newSpeed</i>	Speed value in km/h
-----------------	---------------------

Here is the caller graph for this function:



4.1.3.10 setTotalDistance()

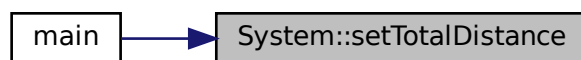
```
void System::setTotalDistance (
    const QString & newTotalDistance )
```

Sets total distance traveled.

Parameters

<i>newTotalDistance</i>	Distance value in kilometers
-------------------------	------------------------------

Here is the caller graph for this function:



4.1.3.11 setTurnLightLeft()

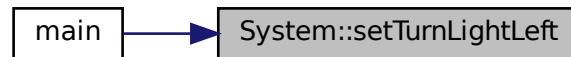
```
void System::setTurnLightLeft (
    const QString & newTurnLightLeft )
```

Sets left turn signal status.

Parameters

<i>newTurnLightLeft</i>	"true" to activate, "false" to deactivate
-------------------------	---

Here is the caller graph for this function:

**4.1.3.12 setTurnLightRight()**

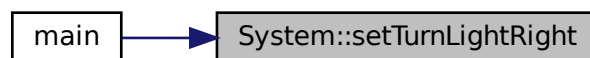
```
void System::setTurnLightRight (
    const QString & newTurnLightRight )
```

Sets right turn signal status.

Parameters

<i>newTurnLightRight</i>	"true" to activate, "false" to deactivate
--------------------------	---

Here is the caller graph for this function:

**4.1.3.13 speed()**

```
QString System::speed ( ) const
```

Gets current vehicle speed.

Returns

QString Current speed in km/h

4.1.3.14 totalDistance()

```
QString System::totalDistance ( ) const
```

Gets total distance traveled.

Returns

QString Total distance in kilometers

4.1.3.15 turnLightLeft()

```
QString System::turnLightLeft ( ) const
```

Gets left turn signal status.

Returns

QString "true" if active, "false" if inactive

4.1.3.16 turnLightRight()

```
QString System::turnLightRight ( ) const
```

Gets right turn signal status.

Returns

QString "true" if active, "false" if inactive

4.1.4 Property Documentation

4.1.4.1 batteryPer

```
QString System::batteryPer [read], [write]
```

Current battery charge percentage (0-100%).

The property is accessible via:

- [batteryPer\(\)](#) to get the current battery percentage.
- [setBatteryPer\(\)](#) to update the battery percentage.

See also

[batteryPerChanged\(\)](#)

4.1.4.2 brakeLight

`QString System::brakeLight [read], [write]`

Brake light status.

Indicates whether the brake light is active.

- "true" for active.
- "false" for inactive.

The property is accessible via:

- `brakeLight()` to get the status.
- `setBrakeLight()` to update the status.

See also

[brakeLightChanged\(\)](#)

4.1.4.3 emergencyLights

`QString System::emergencyLights [read], [write]`

Emergency/hazard lights status.

Indicates whether the emergency lights are on or off.

- "true" for on.
- "false" for off.

The property is accessible via:

- `emergencyLights()` to get the status.
- `setEmergencyLights()` to update the status.

See also

[emergencyLightsChanged\(\)](#)

4.1.4.4 headLights

`QString System::headLights [read], [write]`

Headlights status.

Indicates whether the headlights are on or off.

- "true" for on.
- "false" for off.

The property is accessible via:

- [headLights\(\)](#) to get the status.
- [setHeadLights\(\)](#) to update the status.

See also

[headLightsChanged\(\)](#)

4.1.4.5 speed

`QString System::speed [read], [write]`

Current vehicle speed in kilometers per hour (km/h).

The property is accessible via:

- [speed\(\)](#) to get the current speed.
- [setSpeed\(\)](#) to update the speed.

See also

[speedChanged\(\)](#)

4.1.4.6 totalDistance

`QString System::totalDistance [read], [write]`

Total distance traveled in kilometers (km).

The property is accessible via:

- [totalDistance\(\)](#) to get the total distance.
- [setTotalDistance\(\)](#) to update the total distance.

See also

[totalDistanceChanged\(\)](#)

4.1.4.7 turnLightLeft

QString System::turnLightLeft [read], [write]

Left turn signal status.

Indicates whether the left turn signal is active.

- "true" for active.
- "false" for inactive.

The property is accessible via:

- [turnLightLeft \(\)](#) to get the status.
- [setTurnLightLeft \(\)](#) to update the status.

See also

[turnLightLeftChanged\(\)](#)

4.1.4.8 turnLightRight

QString System::turnLightRight [read], [write]

Right turn signal status.

Indicates whether the right turn signal is active.

- "true" for active.
- "false" for inactive.

The property is accessible via:

- [turnLightRight \(\)](#) to get the status.
- [setTurnLightRight \(\)](#) to update the status.

See also

[turnLightRightChanged\(\)](#)

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

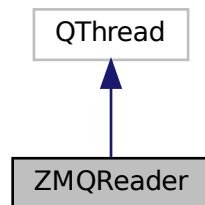
- Digital_Cluster/[System.h](#)
- Digital_Cluster/[System.cpp](#)

4.2 ZMQReader Class Reference

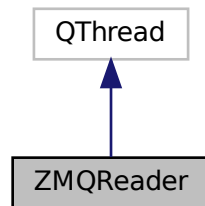
Class responsible for receiving data via ZeroMQ in a separate thread.

```
#include <zmqreader.h>
```

Inheritance diagram for ZMQReader:



Collaboration diagram for ZMQReader:



Signals

- void [speedReceived](#) (QString speed)
Signal emitted when speed data is received.
- void [batteryReceived](#) (QString battery)
Signal emitted when battery charge data is received.
- void [headLightsReceived](#) (QString headLights)
Signal emitted when the headlight status is received.
- void [brakeLightReceived](#) (QString brakeLight)
Signal emitted when the brake light status is received.
- void [turnLightLeftReceived](#) (QString turnLightLeft)
Signal emitted when the left turn signal status is received.
- void [turnLightRightReceived](#) (QString turnLightRight)
Signal emitted when the right turn signal status is received.
- void [emergencyLightsReceived](#) (QString emergencyLights)
Signal emitted when the emergency lights status is received.
- void [totalDistanceReceived](#) (QString totalDistance)
Signal emitted when the total distance traveled is received.

Public Member Functions

- [ZMQReader](#) (const QString &address, QObject *parent=nullptr)
Constructor for the [ZMQReader](#) class.
- [~ZMQReader](#) () override
Destructor for the [ZMQReader](#) class.
- void [stop](#) ()
Stops data reception and terminates the thread.

Protected Member Functions

- void [run](#) () override
Main method of the thread, responsible for processing received data via ZeroMQ.

4.2.1 Detailed Description

Class responsible for receiving data via ZeroMQ in a separate thread.

This class manages a ZeroMQ connection to receive data about speed, battery, and light status.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 ZMQReader()

```
ZMQReader::ZMQReader (
    const QString & address,
    QObject * parent = nullptr ) [explicit]
```

Constructor for the [ZMQReader](#) class.

[ZMQReader](#) class constructor.

Parameters

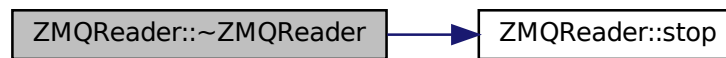
<i>address</i>	ZeroMQ address to connect to.
<i>parent</i>	Optional parent object.

4.2.2.2 ~ZMQReader()

```
ZMQReader::~ZMQReader ( ) [override]
```

Destructor for the [ZMQReader](#) class.

[ZMQReader](#) class destructor. Ensures the thread is properly stopped. Here is the call graph for this function:



4.2.3 Member Function Documentation

4.2.3.1 batteryReceived

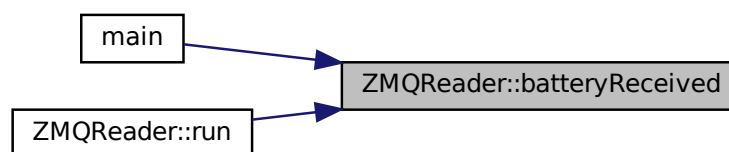
```
void ZMQReader::batteryReceived (
    QString battery ) [signal]
```

Signal emitted when battery charge data is received.

Parameters

<i>battery</i>	Battery percentage.
----------------	---------------------

Here is the caller graph for this function:



4.2.3.2 brakeLightReceived

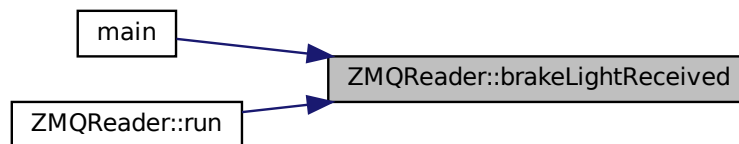
```
void ZMQReader::brakeLightReceived (
    QString brakeLight ) [signal]
```

Signal emitted when the brake light status is received.

Parameters

<i>brakeLight</i>	Brake light status (on/off).
-------------------	------------------------------

Here is the caller graph for this function:



4.2.3.3 emergencyLightsReceived

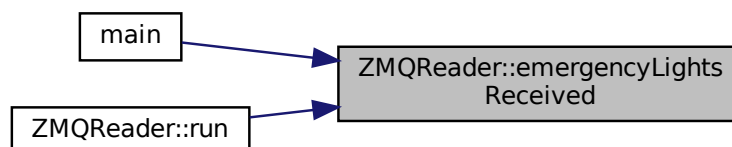
```
void ZMQReader::emergencyLightsReceived (
    QString emergencyLights ) [signal]
```

Signal emitted when the emergency lights status is received.

Parameters

<i>emergencyLights</i>	Emergency lights status (on/off).
------------------------	-----------------------------------

Here is the caller graph for this function:



4.2.3.4 headLightsReceived

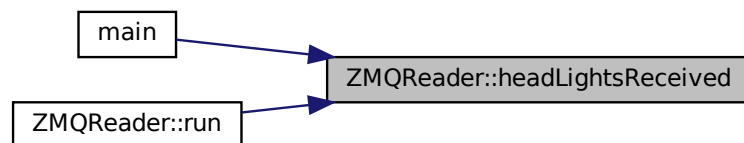
```
void ZMQReader::headLightsReceived (
    QString headLights ) [signal]
```

Signal emitted when the headlight status is received.

Parameters

<i>headLights</i>	Headlight status (on/off).
-------------------	----------------------------

Here is the caller graph for this function:



4.2.3.5 run()

```
void ZMQReader::run ( ) [override], [protected]
```

Main method of the thread, responsible for processing received data via ZeroMQ.

Main thread method responsible for reading messages from the ZeroMQ socket.

4.2.3.6 speedReceived

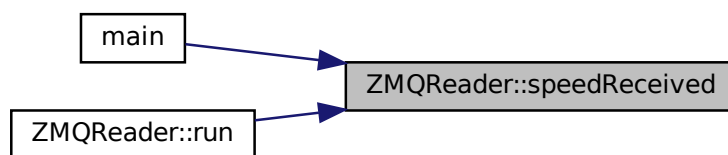
```
void ZMQReader::speedReceived (
    QString speed ) [signal]
```

Signal emitted when speed data is received.

Parameters

<i>speed</i>	Speed value as a string.
--------------	--------------------------

Here is the caller graph for this function:

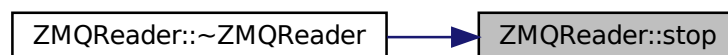


4.2.3.7 stop()

```
void ZMQReader::stop ( )
```

Stops data reception and terminates the thread.

Stops the reading thread execution. Waits until the thread has finished execution. Here is the caller graph for this function:



4.2.3.8 totalDistanceReceived

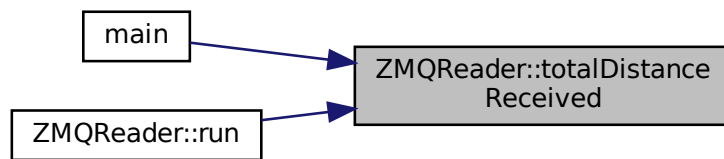
```
void ZMQReader::totalDistanceReceived (
    QString totalDistance ) [signal]
```

Signal emitted when the total distance traveled is received.

Parameters

<i>totalDistance</i>	Total distance traveled.
----------------------	--------------------------

Here is the caller graph for this function:



4.2.3.9 turnLightLeftReceived

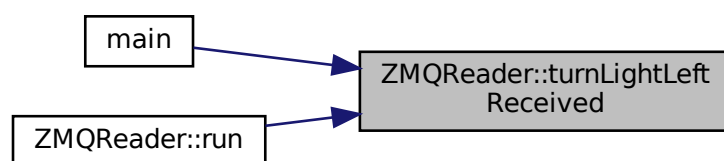
```
void ZMQReader::turnLightLeftReceived (
    QString turnLightLeft ) [signal]
```

Signal emitted when the left turn signal status is received.

Parameters

<i>turnLightLeft</i>	Left turn signal status (on/off).
----------------------	-----------------------------------

Here is the caller graph for this function:



4.2.3.10 turnLightRightReceived

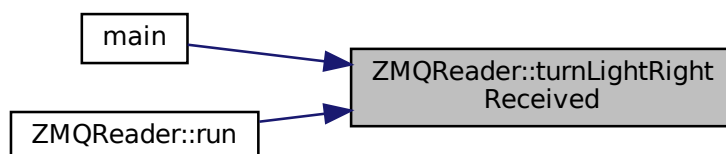
```
void ZMQReader::turnLightRightReceived (
    QString turnLightRight ) [signal]
```

Signal emitted when the right turn signal status is received.

Parameters

<i>turnLightRight</i>	Right turn signal status (on/off).
-----------------------	------------------------------------

Here is the caller graph for this function:



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

- Digital_Cluster/[zmqreader.h](#)
- Digital_Cluster/[zmqreader.cpp](#)

Chapter 5

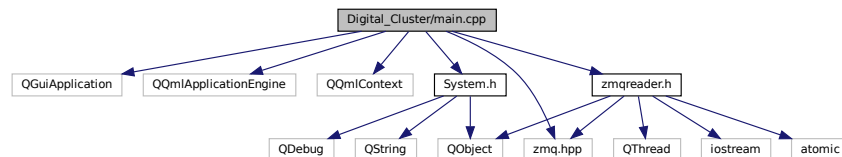
File Documentation

5.1 Digital_Cluster/main.cpp File Reference

Entry point for the Digital Cluster application.

```
#include <QGuiApplication>
#include <QQmlApplicationEngine>
#include <QQmlContext>
#include <zmq.hpp>
#include "System.h"
#include "zmqreader.h"
```

Include dependency graph for main.cpp:



Functions

- `int main (int argc, char *argv[])`
Main function for the application.

5.1.1 Detailed Description

Entry point for the Digital Cluster application.

This file initializes the Qt application, sets up the QML engine, and starts the [ZMQReader](#) to receive data from a ZeroMQ socket.

5.1.2 Function Documentation

5.1.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for the application.

Initializes the Qt application, loads the QML interface, and starts the [ZMQReader](#) to listen for real-time data updates.

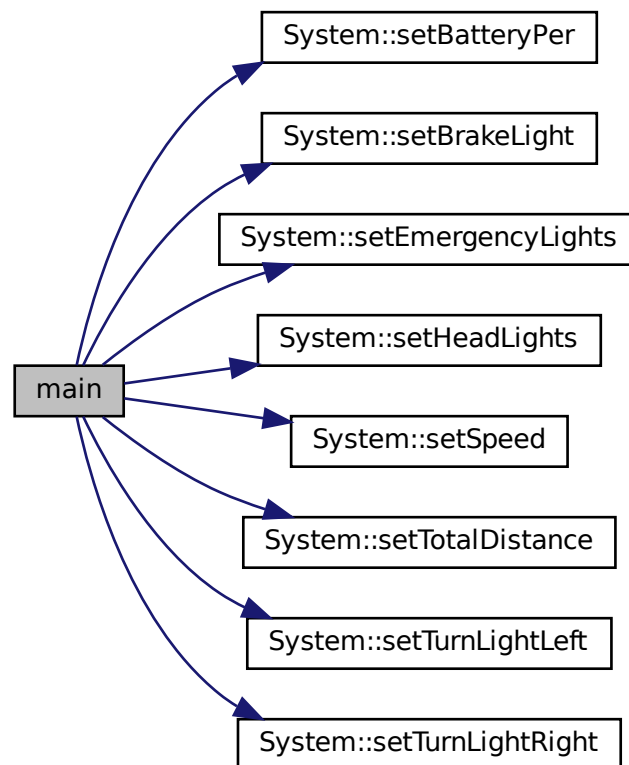
Parameters

<i>argc</i>	Argument count.
<i>argv</i>	Argument vector.

Returns

int Application exit code.

Here is the call graph for this function:

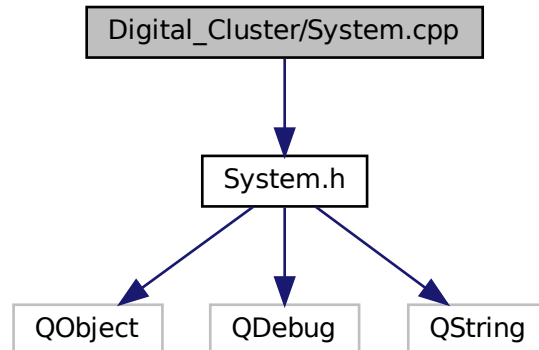


5.2 Digital_Cluster/System.cpp File Reference

Implementation of the [System](#) class.

```
#include "System.h"
```

Include dependency graph for System.cpp:



5.2.1 Detailed Description

Implementation of the [System](#) class.

This file contains the implementation of the [System](#) class methods, which manage vehicle properties such as speed, battery, and light states.

5.3 Digital_Cluster/System.h File Reference

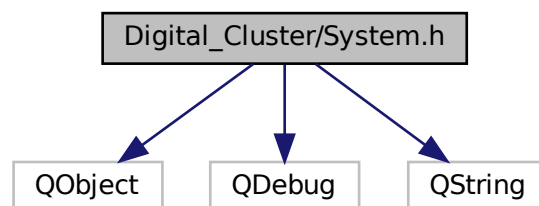
Declaration of the [System](#) class.

```
#include <QObject>
```

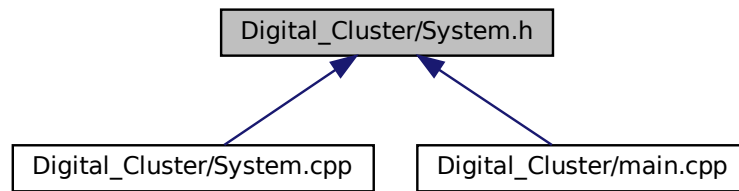
```
#include <QDebug>
```

```
#include <QString>
```

Include dependency graph for System.h:



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



Classes

- class [System](#)

The [System](#) class manages vehicle properties and states for QML integration.

5.3.1 Detailed Description

Declaration of the [System](#) class.

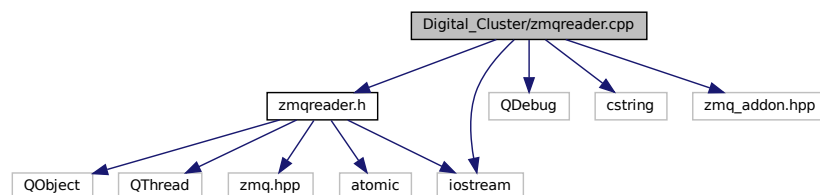
This file contains the declaration of the [System](#) class, which manages properties related to the vehicle's state, such as speed, battery percentage, and light states.

5.4 Digital_Cluster/zmqreader.cpp File Reference

Implementation of the [ZMQReader](#) class for reading messages via ZeroMQ.

```
#include "zmqreader.h"
#include <QDebug>
#include <cstring>
#include <iostream>
#include <zmq_addon.hpp>
```

Include dependency graph for `zmqreader.cpp`:



5.4.1 Detailed Description

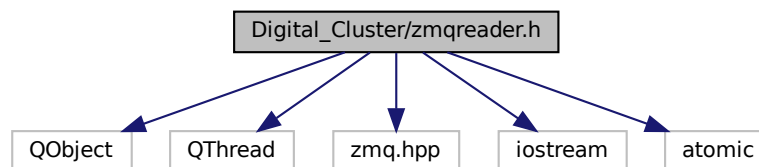
Implementation of the [ZMQReader](#) class for reading messages via ZeroMQ.

5.5 Digital_Cluster/zmqreader.h File Reference

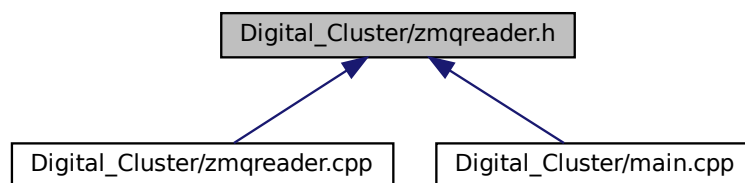
Declares the [ZMQReader](#) class for reading messages via ZeroMQ.

```
#include <QObject>
#include <QThread>
#include <zmq.hpp>
#include <iostream>
#include <atomic>
```

Include dependency graph for zmqreader.h:



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



Classes

- class [ZMQReader](#)

Class responsible for receiving data via ZeroMQ in a separate thread.

5.5.1 Detailed Description

Declares the [ZMQReader](#) class for reading messages via ZeroMQ.

Index

~ZMQReader
ZMQReader, 20

batteryPer
System, 9, 15
batteryReceived
ZMQReader, 21
brakeLight
System, 10, 15
brakeLightReceived
ZMQReader, 21

Digital_Cluster/main.cpp, 27
Digital_Cluster/System.cpp, 29
Digital_Cluster/System.h, 30
Digital_Cluster/zmqreader.cpp, 31
Digital_Cluster/zmqreader.h, 32

emergencyLights
System, 10, 16
emergencyLightsReceived
ZMQReader, 22

headLights
System, 10, 16
headLightsReceived
ZMQReader, 22

main
main.cpp, 28
main.cpp
main, 28

run
ZMQReader, 23

setBatteryPer
System, 10
setBrakeLight
System, 11
setEmergencyLights
System, 11
setHeadLights
System, 12
setSpeed
System, 12
setTotalDistance
System, 13
setTurnLightLeft
System, 13
setTurnLightRight

System, 14
speed
System, 14, 17
speedReceived
ZMQReader, 23
stop
ZMQReader, 24
System, 7
batteryPer, 9, 15
brakeLight, 10, 15
emergencyLights, 10, 16
headLights, 10, 16
setBatteryPer, 10
setBrakeLight, 11
setEmergencyLights, 11
setHeadLights, 12
setSpeed, 12
setTotalDistance, 13
setTurnLightLeft, 13
setTurnLightRight, 14
speed, 14, 17
System, 9
totalDistance, 14, 17
turnLightLeft, 15, 17
turnLightRight, 15, 18

totalDistance
System, 14, 17
totalDistanceReceived
ZMQReader, 24
turnLightLeft
System, 15, 17
turnLightLeftReceived
ZMQReader, 25
turnLightRight
System, 15, 18
turnLightRightReceived
ZMQReader, 25

ZMQReader, 19
~ZMQReader, 20
batteryReceived, 21
brakeLightReceived, 21
emergencyLightsReceived, 22
headLightsReceived, 22
run, 23
speedReceived, 23
stop, 24
totalDistanceReceived, 24
turnLightLeftReceived, 25

turnLightRightReceived, [25](#)
ZMQReader, [20](#)