

# My Project

Generated by Doxygen 1.9.8



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 LSRecvUDPManager Class Reference	5
3.1.1 Constructor & Destructor Documentation	5
3.1.1.1 LSRecvUDPManager()	5
3.1.1.2 ~LSRecvUDPManager()	5
3.1.2 Member Function Documentation	6
3.1.2.1 closeSocket()	6
3.1.2.2 LSSocketOpen()	6
3.1.2.3 receiveData()	6
3.2 MFRSendUDPManager Class Reference	6
3.2.1 Constructor & Destructor Documentation	7
3.2.1.1 MFRSendUDPManager()	7
3.2.1.2 ~MFRSendUDPManager()	7
3.2.2 Member Function Documentation	7
3.2.2.1 MFRSocketOpen()	7
3.2.2.2 sendData()	7
3.3 MissileInfo Struct Reference	7
3.3.1 Member Data Documentation	8
3.3.1.1 angle	8
3.3.1.2 id	8
3.3.1.3 mockType	8
3.3.1.4 speed	8
3.3.1.5 x	8
3.3.1.6 y	8
3.3.1.7 z	8
3.4 MockMissile Class Reference	9
3.4.1 Constructor & Destructor Documentation	9
3.4.1.1 MockMissile()	9
3.4.1.2 ~MockMissile()	9
3.4.2 Member Function Documentation	9
3.4.2.1 updatePosMissile()	9
3.5 MockMissileManager Class Reference	10
3.5.1 Constructor & Destructor Documentation	10
3.5.1.1 MockMissileManager()	10
3.5.2 Member Function Documentation	10
3.5.2.1 flightMissile()	10
3.6 MockTarget Class Reference	10

3.6.1 Constructor & Destructor Documentation	11
3.6.1.1 MockTarget()	11
3.6.1.2 ~MockTarget()	11
3.6.2 Member Function Documentation	11
3.6.2.1 updatePos()	11
3.7 MockTargetManager Class Reference	11
3.7.1 Constructor & Destructor Documentation	11
3.7.1.1 MockTargetManager()	11
3.7.2 Member Function Documentation	12
3.7.2.1 addTarget()	12
3.7.2.2 flightTarget()	12
3.7.2.3 RaedTargetIni()	12
3.8 Simulator Class Reference	12
3.8.1 Constructor & Destructor Documentation	13
3.8.1.1 Simulator()	13
3.8.1.2 ~Simulator()	13
3.8.2 Member Function Documentation	13
3.8.2.1 init()	13
3.8.2.2 start()	13
3.9 TargetInfo Struct Reference	14
3.9.1 Member Data Documentation	14
3.9.1.1 angle	14
3.9.1.2 id	14
3.9.1.3 mockType	14
3.9.1.4 speed	14
3.9.1.5 x	14
3.9.1.6 y	14
3.9.1.7 z	14
<b>4 File Documentation</b>	<b>15</b>
4.1 Simulator/main.cpp File Reference	15
4.1.1 Function Documentation	15
4.1.1.1 main()	15
4.2 Simulator/Mock/info/MissileInfo.h File Reference	16
4.3 MissileInfo.h	17
4.4 Simulator/Mock/info/TargetInfo.h File Reference	17
4.5 TargetInfo.h	18
4.6 Simulator/Mock/MockMissile.cpp File Reference	18
4.7 Simulator/Mock/MockMissile.h File Reference	19
4.8 MockMissile.h	20
4.9 Simulator/Mock/MockMissileManager.cpp File Reference	20
4.10 Simulator/Mock/MockMissileManager.h File Reference	21

---

4.11 MockMissileManager.h . . . . .	21
4.12 Simulator/Mock/MockTarget.cpp File Reference . . . . .	22
4.13 Simulator/Mock/MockTarget.h File Reference . . . . .	22
4.14 MockTarget.h . . . . .	23
4.15 Simulator/Mock/MockTargetManager.cpp File Reference . . . . .	24
4.16 Simulator/Mock/MockTargetManager.h File Reference . . . . .	24
4.17 MockTargetManager.h . . . . .	25
4.18 Simulator/Simulator/Simulator.cpp File Reference . . . . .	26
4.19 Simulator/Simulator/Simulator.h File Reference . . . . .	26
4.20 Simulator.h . . . . .	27
4.21 Simulator/UDPCommunicate/LSRecvUDPManager.cpp File Reference . . . . .	28
4.22 Simulator/UDPCommunicate/LSRecvUDPManager.h File Reference . . . . .	28
4.23 LSRecvUDPManager.h . . . . .	29
4.24 Simulator/UDPCommunicate/MFRSendUDPManager.cpp File Reference . . . . .	30
4.25 Simulator/UDPCommunicate/MFRSendUDPManager.h File Reference . . . . .	30
4.26 MFRSendUDPManager.h . . . . .	31
<b>Index</b>	<b>33</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">LSRecvUDPManager</a>	5
<a href="#">MFRSendUDPManager</a>	6
<a href="#">MissileInfo</a>	7
<a href="#">MockMissile</a>	9
<a href="#">MockMissileManager</a>	10
<a href="#">MockTarget</a>	10
<a href="#">MockTargetManager</a>	11
<a href="#">Simulator</a>	12
<a href="#">TargetInfo</a>	14





# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

Simulator/ <a href="#">main.cpp</a> . . . . .	15
Simulator/Mock/ <a href="#">MockMissile.cpp</a> . . . . .	18
Simulator/Mock/ <a href="#">MockMissile.h</a> . . . . .	19
Simulator/Mock/ <a href="#">MockMissileManager.cpp</a> . . . . .	20
Simulator/Mock/ <a href="#">MockMissileManager.h</a> . . . . .	21
Simulator/Mock/ <a href="#">MockTarget.cpp</a> . . . . .	22
Simulator/Mock/ <a href="#">MockTarget.h</a> . . . . .	22
Simulator/Mock/ <a href="#">MockTargetManager.cpp</a> . . . . .	24
Simulator/Mock/ <a href="#">MockTargetManager.h</a> . . . . .	24
Simulator/Mock/info/ <a href="#">MissileInfo.h</a> . . . . .	16
Simulator/Mock/info/ <a href="#">TargetInfo.h</a> . . . . .	17
Simulator/Simulator/ <a href="#">Simulator.cpp</a> . . . . .	26
Simulator/Simulator/ <a href="#">Simulator.h</a> . . . . .	26
Simulator/UDPCommunicate/ <a href="#">LSRecvUDPManager.cpp</a> . . . . .	28
Simulator/UDPCommunicate/ <a href="#">LSRecvUDPManager.h</a> . . . . .	28
Simulator/UDPCommunicate/ <a href="#">MFRSendUDPManager.cpp</a> . . . . .	30
Simulator/UDPCommunicate/ <a href="#">MFRSendUDPManager.h</a> . . . . .	30



## Chapter 3

# Class Documentation

### 3.1 LSRecvUDPManager Class Reference

```
#include <LSRecvUDPManager.h>
```

#### Public Member Functions

- [LSRecvUDPManager\(\)](#)
- [~LSRecvUDPManager\(\)](#)
- [bool LSSocketOpen](#) (int port)
- [bool receiveData](#) (char \*buffer, int bufferSize)
- [void closeSocket](#) ()

#### 3.1.1 Constructor & Destructor Documentation

##### 3.1.1.1 LSRecvUDPManager()

```
LSRecvUDPManager::LSRecvUDPManager ( )
```

##### 3.1.1.2 ~LSRecvUDPManager()

```
LSRecvUDPManager::~~LSRecvUDPManager ( )
```

Here is the call graph for this function:

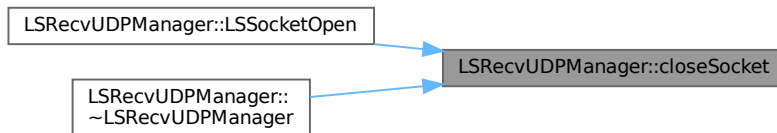


### 3.1.2 Member Function Documentation

#### 3.1.2.1 closeSocket()

```
void LSRecvUDPManager::closeSocket ( )
```

Here is the caller graph for this function:



#### 3.1.2.2 LSSocketOpen()

```
bool LSRecvUDPManager::LSSocketOpen (
    int port )
```

Here is the call graph for this function:



#### 3.1.2.3 receiveData()

```
bool LSRecvUDPManager::receiveData (
    char * buffer,
    int bufferSize )
```

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

- Simulator/UDPCommunicate/[LSRecvUDPManager.h](#)
- Simulator/UDPCommunicate/[LSRecvUDPManager.cpp](#)

## 3.2 MFRSendUDPManager Class Reference

```
#include <MFRSendUDPManager.h>
```

## Public Member Functions

- [MFRSendUDPManager](#) ()
- [~MFRSendUDPManager](#) ()
- bool [MFRSocketOpen](#) (const std::string &ip, int port)
- bool [sendData](#) (const char \*data, int dataSize)

## 3.2.1 Constructor & Destructor Documentation

### 3.2.1.1 MFRSendUDPManager()

```
MFRSendUDPManager::MFRSendUDPManager ( )
```

### 3.2.1.2 ~MFRSendUDPManager()

```
MFRSendUDPManager::~~MFRSendUDPManager ( )
```

## 3.2.2 Member Function Documentation

### 3.2.2.1 MFRSocketOpen()

```
bool MFRSendUDPManager::MFRSocketOpen (
    const std::string & ip,
    int port )
```

### 3.2.2.2 sendData()

```
bool MFRSendUDPManager::sendData (
    const char * data,
    int dataSize )
```

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

- Simulator/UDPCommunicate/[MFRSendUDPManager.h](#)
- Simulator/UDPCommunicate/[MFRSendUDPManager.cpp](#)

## 3.3 MissileInfo Struct Reference

```
#include <MissileInfo.h>
```

## Public Attributes

- uint8\_t [mockType](#)
- unsigned int [id](#)
- long long [x](#)
- long long [y](#)
- long long [z](#)
- double [angle](#)
- int [speed](#)

## 3.3.1 Member Data Documentation

### 3.3.1.1 [angle](#)

```
double MissileInfo::angle
```

### 3.3.1.2 [id](#)

```
unsigned int MissileInfo::id
```

### 3.3.1.3 [mockType](#)

```
uint8_t MissileInfo::mockType
```

### 3.3.1.4 [speed](#)

```
int MissileInfo::speed
```

### 3.3.1.5 [x](#)

```
long long MissileInfo::x
```

### 3.3.1.6 [y](#)

```
long long MissileInfo::y
```

### 3.3.1.7 [z](#)

```
long long MissileInfo::z
```

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

- Simulator/Mock/info/[MissileInfo.h](#)

## 3.4 MockMissile Class Reference

```
#include <MockMissile.h>
```

### Public Member Functions

- [MockMissile](#) (const [MissileInfo](#) &missile\_info, std::shared\_ptr< [MFRSendUDPManager](#) > mfr\_send\_manager)
- [~MockMissile](#) ()
- void [updatePosMissile](#) ()

### 3.4.1 Constructor & Destructor Documentation

#### 3.4.1.1 MockMissile()

```
MockMissile::MockMissile (
    const MissileInfo & missile_info,
    std::shared_ptr< MFRSendUDPManager > mfr_send_manager )
```

#### 3.4.1.2 ~MockMissile()

```
MockMissile::~MockMissile ( ) [inline]
```

### 3.4.2 Member Function Documentation

#### 3.4.2.1 updatePosMissile()

```
void MockMissile::updatePosMissile ( )
```

Here is the caller graph for this function:



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

- Simulator/Mock/[MockMissile.h](#)
- Simulator/Mock/[MockMissile.cpp](#)

## 3.5 MockMissileManager Class Reference

```
#include <MockMissileManager.h>
```

### Public Member Functions

- [MockMissileManager](#) (std::shared\_ptr< [MockTargetManager](#) > target\_manager, std::shared\_ptr< [MFRSendUDPManager](#) > mfr\_send\_manager)
- void [flightMissile](#) (const [MissileInfo](#) &[MissileInfo](#))

### 3.5.1 Constructor & Destructor Documentation

#### 3.5.1.1 MockMissileManager()

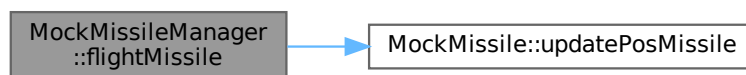
```
MockMissileManager::MockMissileManager (
    std::shared_ptr< MockTargetManager > target_manager,
    std::shared_ptr< MFRSendUDPManager > mfr_send_manager )
```

### 3.5.2 Member Function Documentation

#### 3.5.2.1 flightMissile()

```
void MockMissileManager::flightMissile (
    const MissileInfo & MissileInfo )
```

Here is the call graph for this function:



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

- Simulator/Mock/[MockMissileManager.h](#)
- Simulator/Mock/[MockMissileManager.cpp](#)

## 3.6 MockTarget Class Reference

```
#include <MockTarget.h>
```



## Public Member Functions

- [MockTarget](#) (const [TargetInfo](#) &target\_info, std::shared\_ptr< [MFRSendUDPManager](#) > mfr\_send\_manager)
- [~MockTarget](#) ()
- void [updatePos](#) ()

## 3.6.1 Constructor & Destructor Documentation

### 3.6.1.1 MockTarget()

```
MockTarget::MockTarget (
    const TargetInfo & target_info,
    std::shared_ptr< MFRSendUDPManager > mfr_send_manager )
```

### 3.6.1.2 ~MockTarget()

```
MockTarget::~MockTarget ( )
```

## 3.6.2 Member Function Documentation

### 3.6.2.1 updatePos()

```
void MockTarget::updatePos ( )
```

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

- Simulator/Mock/[MockTarget.h](#)
- Simulator/Mock/[MockTarget.cpp](#)

## 3.7 MockTargetManager Class Reference

```
#include <MockTargetManager.h>
```

## Public Member Functions

- [MockTargetManager](#) (std::shared\_ptr< [MFRSendUDPManager](#) > mfr\_send\_manager)
- void [RaedTargetIni](#) ()
- void [addTarget](#) (std::shared\_ptr< [MockTarget](#) > &target)
- void [flightTarget](#) ()

## 3.7.1 Constructor & Destructor Documentation

### 3.7.1.1 MockTargetManager()

```
MockTargetManager::MockTargetManager (
    std::shared_ptr< MFRSendUDPManager > mfr_send_manager )
```

## 3.7.2 Member Function Documentation

### 3.7.2.1 addTarget()

```
void MockTargetManager::addTarget (
    std::shared_ptr< MockTarget > & target )
```

Here is the caller graph for this function:



### 3.7.2.2 flitghtTarget()

```
void MockTargetManager::flitghtTarget ( )
```

### 3.7.2.3 RaedTargetIni()

```
void MockTargetManager::RaedTargetIni ( )
```

Here is the call graph for this function:



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

- Simulator/Mock/[MockTargetManager.h](#)
- Simulator/Mock/[MockTargetManager.cpp](#)

## 3.8 Simulator Class Reference

```
#include <Simulator.h>
```

## Public Member Functions

- [Simulator](#) ()
- [~Simulator](#) ()
- bool [init](#) ()
- void [start](#) ()

## 3.8.1 Constructor & Destructor Documentation

### 3.8.1.1 Simulator()

```
Simulator::Simulator ( )
```

### 3.8.1.2 ~Simulator()

```
Simulator::~~Simulator ( )
```

## 3.8.2 Member Function Documentation

### 3.8.2.1 init()

```
bool Simulator::init ( )
```

Here is the caller graph for this function:



### 3.8.2.2 start()

```
void Simulator::start ( )
```

Here is the caller graph for this function:



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

- Simulator/Simulator/[Simulator.h](#)
- Simulator/Simulator/[Simulator.cpp](#)

## 3.9 TargetInfo Struct Reference

```
#include <TargetInfo.h>
```

### Public Attributes

- uint8\_t [mockType](#)
- unsigned int [id](#)
- float [x](#)
- float [y](#)
- float [z](#)
- float [angle](#)
- int [speed](#)

### 3.9.1 Member Data Documentation

#### 3.9.1.1 angle

```
float TargetInfo::angle
```

#### 3.9.1.2 id

```
unsigned int TargetInfo::id
```

#### 3.9.1.3 mockType

```
uint8_t TargetInfo::mockType
```

#### 3.9.1.4 speed

```
int TargetInfo::speed
```

#### 3.9.1.5 x

```
float TargetInfo::x
```

#### 3.9.1.6 y

```
float TargetInfo::y
```

#### 3.9.1.7 z

```
float TargetInfo::z
```

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

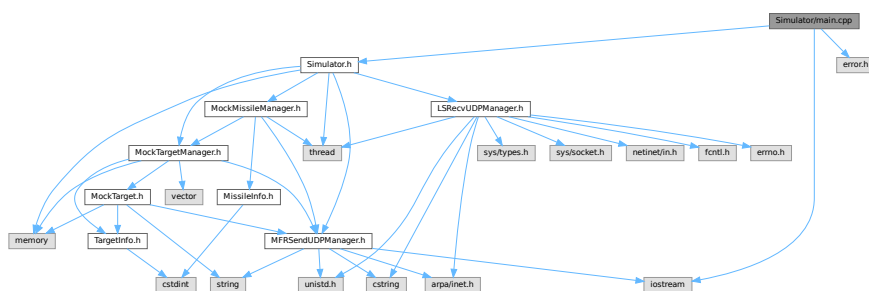
- Simulator/Mock/info/[TargetInfo.h](#)

## Chapter 4

# File Documentation

### 4.1 Simulator/main.cpp File Reference

```
#include "Simulator.h"  
#include <error.h>  
#include <iostream>  
Include dependency graph for main.cpp:
```



#### Functions

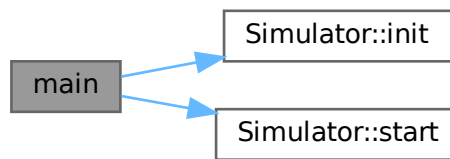
- int [main](#) ()

#### 4.1.1 Function Documentation

##### 4.1.1.1 main()

```
int main ( )
```

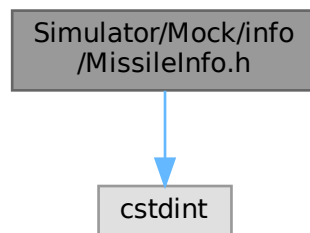
Here is the call graph for this function:



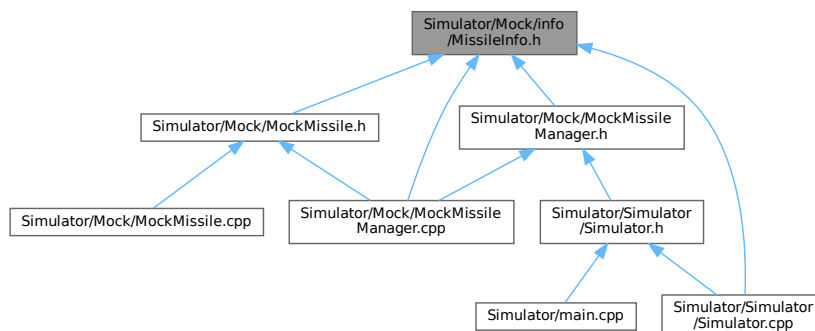
## 4.2 Simulator/Mock/info/MissileInfo.h File Reference

```
#include <cstdint>
```

Include dependency graph for MissileInfo.h:



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



## Classes

- struct [MissileInfo](#)

## 4.3 MissileInfo.h

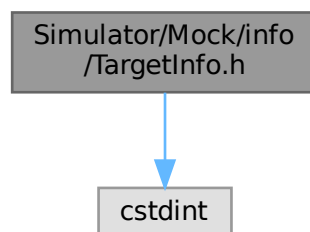
[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 #include <stdint>
00003 #pragma pack(push, 1)
00004 struct MissileInfo
00005 {
00006     uint8_t mockType;
00007     unsigned int id;
00008     long long x;
00009     long long y;
00010     long long z;
00011     double angle;
00012     int speed;
00013 };
00014 #pragma pack(pop)
```

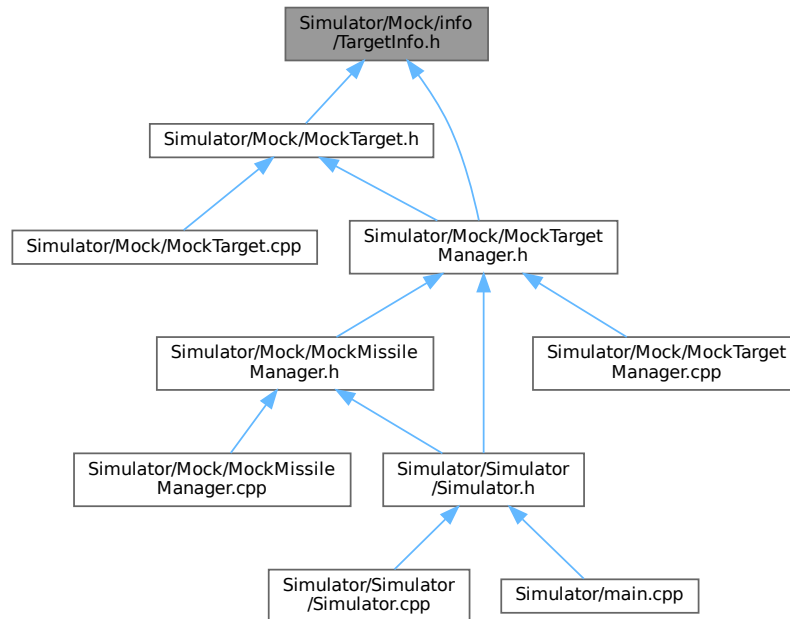
## 4.4 Simulator/Mock/info/TargetInfo.h File Reference

```
#include <stdint>
```

Include dependency graph for TargetInfo.h:



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



## Classes

- struct [TargetInfo](#)

## 4.5 TargetInfo.h

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002 #include <stdint>
00003
00004 #pragma pack(push, 1)
00005 struct TargetInfo
00006 {
00007     uint8_t mockType;
00008     unsigned int id;
00009     float x;
00010     float y;
00011     float z;
00012     float angle;
00013     int speed;
00014 };
00015 #pragma pack(pop)
  
```

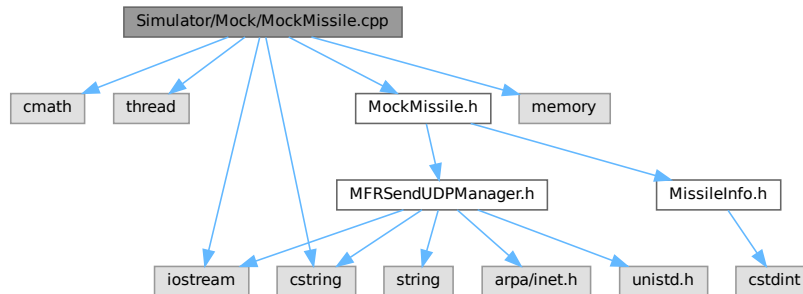
## 4.6 Simulator/Mock/MockMissile.cpp File Reference

```

#include <cmath>
#include <thread>
#include <iostream>
#include <cstring>
  
```

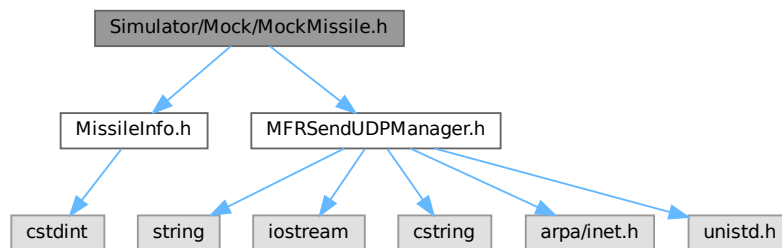


```
#include <memory>
#include "MockMissile.h"
Include dependency graph for MockMissile.cpp:
```

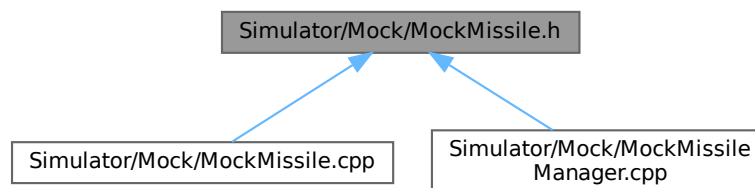


## 4.7 Simulator/Mock/MockMissile.h File Reference

```
#include "MissileInfo.h"
#include "MFRSendUDPManager.h"
Include dependency graph for MockMissile.h:
```



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



## Classes

- class [MockMissile](#)

## 4.8 MockMissile.h

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002 #include "MissileInfo.h"
00003 #include "MFRSendUDPManager.h"
00004
00005 class MockMissile
00006 {
00007 public:
00008     MockMissile(const MissileInfo &missile_info, std::shared_ptr<MFRSendUDPManager> mfr_send_manager);
00009     ~MockMissile() {}
00010
00011     void updatePosMissile();
00012
00013 private:
00014     MissileInfo missile_info_;
00015     std::shared_ptr<MFRSendUDPManager> mfr_send_manager_;
00016
00017     void sendData();
00018 };

```

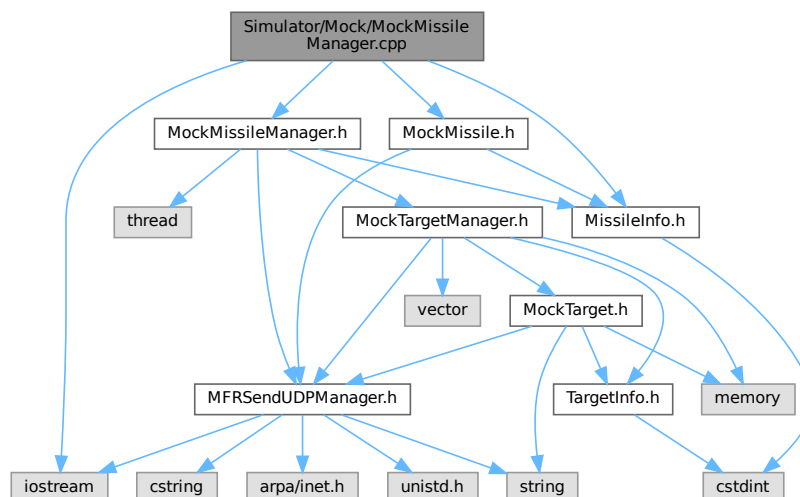
## 4.9 Simulator/Mock/MockMissileManager.cpp File Reference

```

#include <iostream>
#include "MockMissileManager.h"
#include "MissileInfo.h"
#include "MockMissile.h"

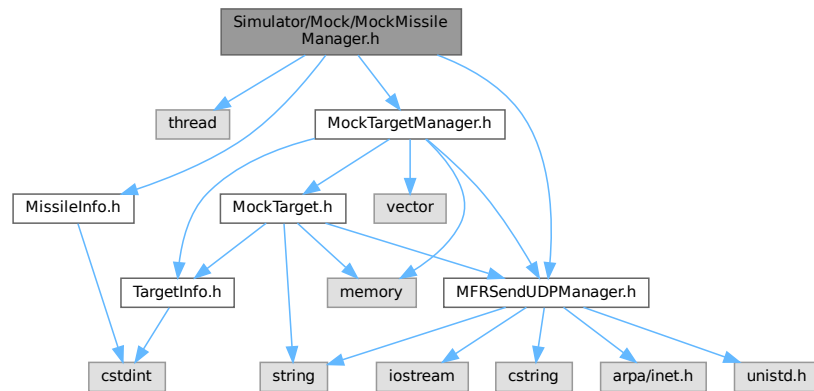
```

Include dependency graph for MockMissileManager.cpp:

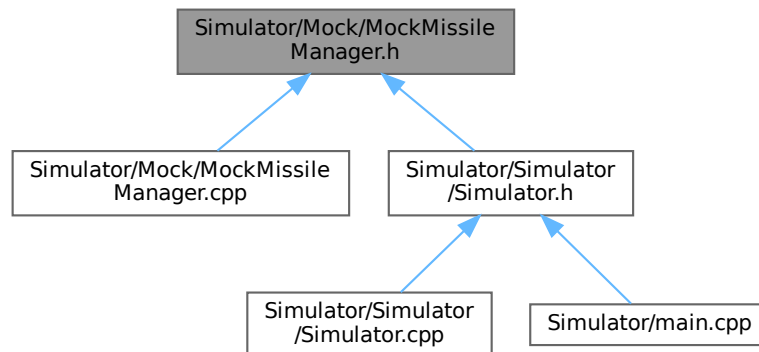


## 4.10 Simulator/Mock/MockMissileManager.h File Reference

```
#include <thread>
#include "MissileInfo.h"
#include "MockTargetManager.h"
#include "MFRSendUDPManager.h"
Include dependency graph for MockMissileManager.h:
```



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



### Classes

- class [MockMissileManager](#)

## 4.11 MockMissileManager.h

[Go to the documentation of this file.](#)

```

00001 #ifndef MOCK_MISSILE_MANAGER_H
00002 #define MOCK_MISSILE_MANAGER_H
00003
00004 #include <thread>
00005
00006 #include "MissileInfo.h"
00007 #include "MockTargetManager.h"
00008 #include "MFRSendUDPManager.h"
00009
00010 class MockMissileManager
00011 {
00012 private:
00013     std::shared_ptr<MockTargetManager> mock_target_manager_;
00014     std::shared_ptr<MFRSendUDPManager> mfr_send_manager_;
00015
00016     bool is_flight_ = false;
00017     std::thread flight_thread_;
00018
00019     int last_missile_id_ = 0;
00020     MissileInfo last_missile_info_;
00021
00022     void updateMissileID();
00023
00024 public:
00025     MockMissileManager(std::shared_ptr<MockTargetManager> target_manager,
00026         std::shared_ptr<MFRSendUDPManager> mfr_send_manager);
00027
00028     void flightMissile(const MissileInfo &MissileInfo);
00029 };
00030 #endif // MOCK_MISSILE_MANAGER_H

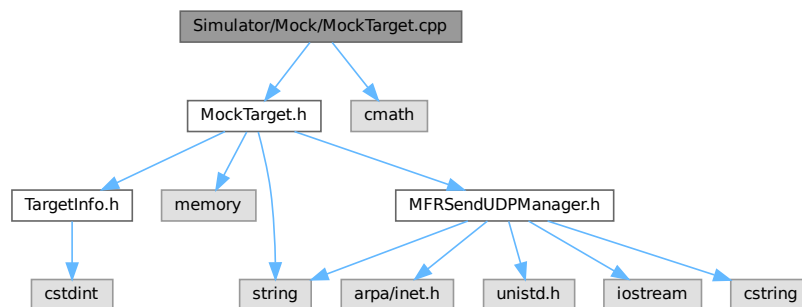
```

## 4.12 Simulator/Mock/MockTarget.cpp File Reference

```
#include "MockTarget.h"
```

```
#include <cmath>
```

Include dependency graph for MockTarget.cpp:



## 4.13 Simulator/Mock/MockTarget.h File Reference

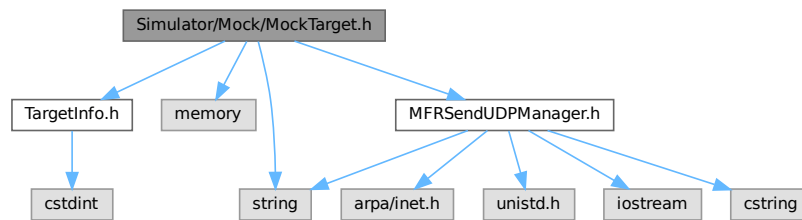
```
#include <string>
```

```
#include <memory>
```

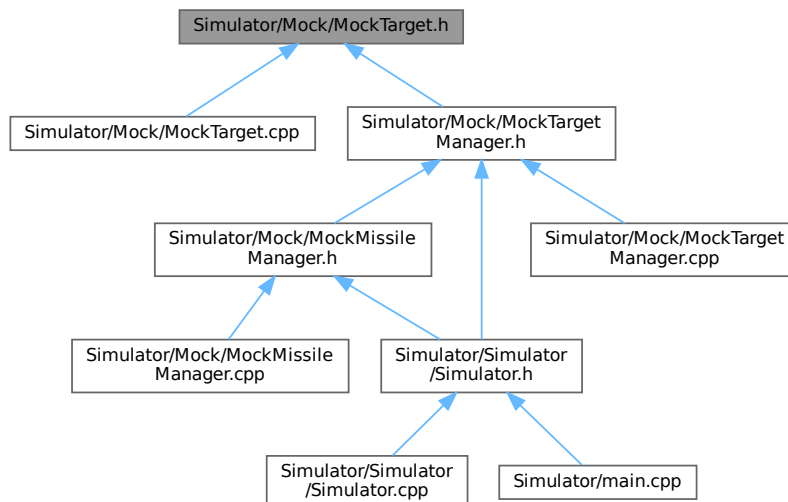
```
#include "TargetInfo.h"
```

```
#include "MFRSendUDPManager.h"
```

Include dependency graph for MockTarget.h:



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



## Classes

- class [MockTarget](#)

## 4.14 MockTarget.h

[Go to the documentation of this file.](#)

```

00001 #ifndef MOCK_TARGET_H
00002 #define MOCK_TARGET_H
00003
00004 #include <string>
00005 #include <memory>
00006
00007 #include "TargetInfo.h"
00008 #include "MFRSendUDPManager.h"
00009
00010 class MockTarget

```

```

00011 {
00012 public:
00013     MockTarget(const TargetInfo &target_info, std::shared_ptr<MFRSendUDPManager> mfr_send_manager);
00014     ~MockTarget();
00015
00016     void updatePos();
00017
00018 private:
00019     TargetInfo target_info_;
00020     std::shared_ptr<MFRSendUDPManager> mfr_send_manager_ = nullptr;
00021
00022     void sendData();
00023 }
00024 #endif // MOCK_TARGET_H

```

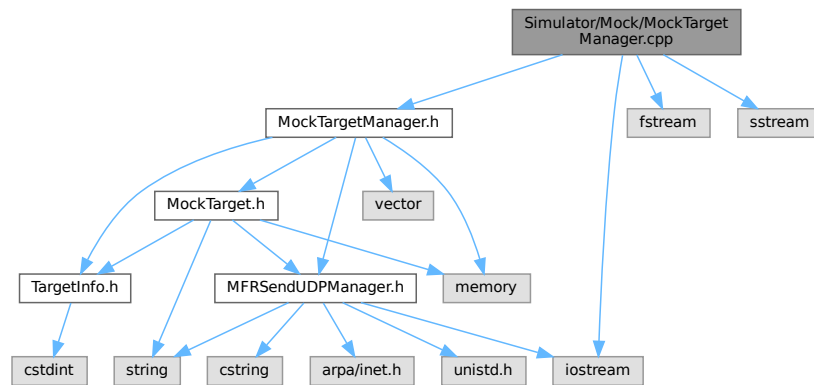
## 4.15 Simulator/Mock/MockTargetManager.cpp File Reference

```

#include "MockTargetManager.h"
#include <iostream>
#include <fstream>
#include <sstream>

```

Include dependency graph for MockTargetManager.cpp:



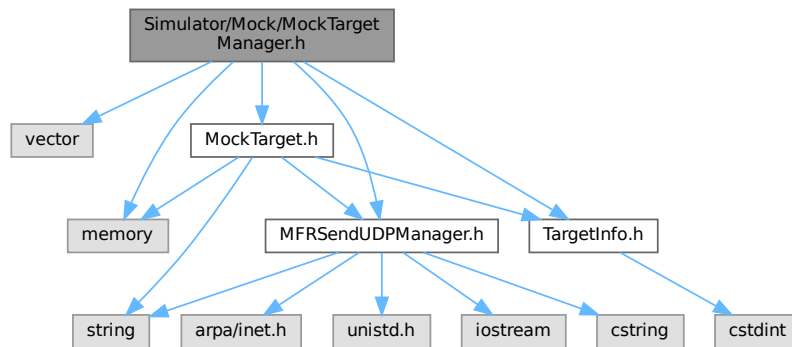
## 4.16 Simulator/Mock/MockTargetManager.h File Reference

```

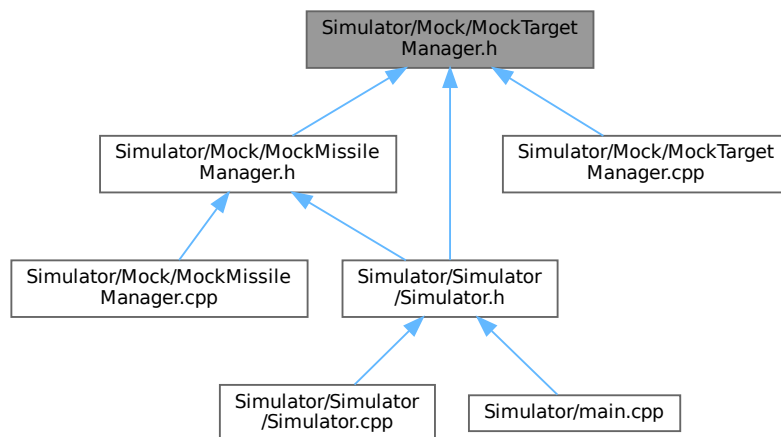
#include <vector>
#include <memory>
#include "MockTarget.h"
#include "TargetInfo.h"
#include "MFRSendUDPManager.h"

```

Include dependency graph for MockTargetManager.h:



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



## Classes

- class [MockTargetManager](#)

## 4.17 MockTargetManager.h

[Go to the documentation of this file.](#)

```

00001 #ifndef MOCK_TARGET_MANAGER_H
00002 #define MOCK_TARGET_MANAGER_H
00003
00004 #include <vector>
00005 #include <memory>
00006
00007 #include "MockTarget.h"
00008 #include "TargetInfo.h"
  
```

```

00009
00010 #include "MFRSendUDPManager.h"
00011
00012 class MockTargetManager
00013 {
00014 public:
00015     MockTargetManager(std::shared_ptr<MFRSendUDPManager> mfr_send_manager);
00016     void RaedTargetIni();
00017     void addTarget(std::shared_ptr<MockTarget> &target);
00018     void flitghtTarget();
00019
00020 private:
00021     std::vector<std::shared_ptr<MockTarget> > targets;
00022     std::shared_ptr<MFRSendUDPManager> mfr_send_manager_;
00023 };
00024
00025 #endif

```

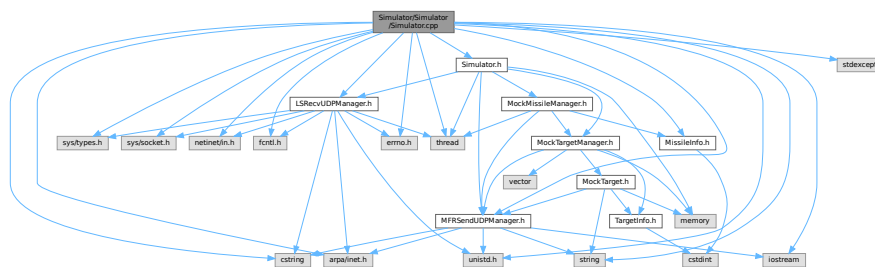
## 4.18 Simulator/Simulator/Simulator.cpp File Reference

```

#include "Simulator.h"
#include <iostream>
#include <cstring>
#include <arpa/inet.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <errno.h>
#include <string>
#include <stdexcept>
#include <thread>
#include "MissileInfo.h"
#include "LSRecvUDPManager.h"
#include "MFRSendUDPManager.h"

```

Include dependency graph for Simulator.cpp:



## 4.19 Simulator/Simulator/Simulator.h File Reference

```

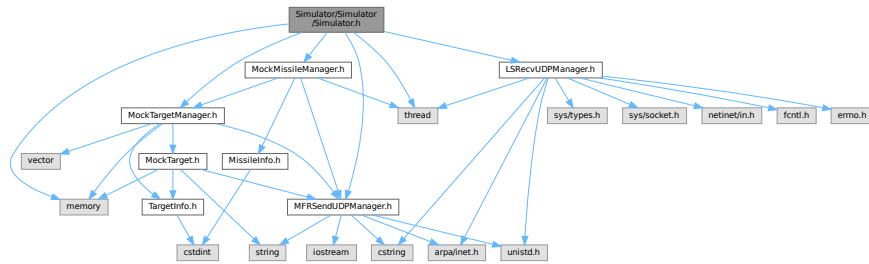
#include <thread>
#include <memory>
#include "LSRecvUDPManager.h"
#include "MFRSendUDPManager.h"
#include "MockTargetManager.h"

```

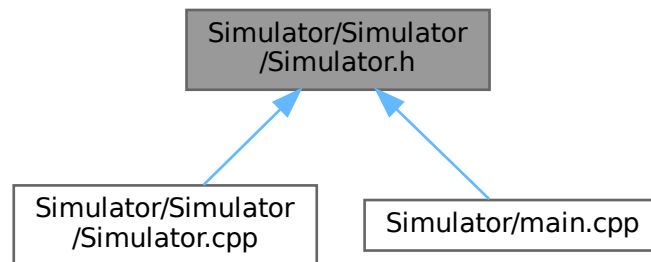


```
#include "MockMissileManager.h"
```

Include dependency graph for Simulator.h:



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



## Classes

- class [Simulator](#)

## 4.20 Simulator.h

[Go to the documentation of this file.](#)

```

00001 #include <thread>
00002 #include <memory>
00003
00004 #include "LSRecvUDPManager.h"
00005 #include "MFRSendUDPManager.h"
00006 #include "MockTargetManager.h"
00007 #include "MockMissileManager.h"
00008
00009 class Simulator
00010 {
00011 private:
00012     std::unique_ptr<LSRecvUDPManager> ls_recv_manager_;
00013     std::shared_ptr<MFRSendUDPManager> mfr_send_manager_;
00014
00015     std::shared_ptr<MockTargetManager> mock_target_manager_;
00016     std::unique_ptr<MockMissileManager> mock_missile_manager_;
00017
00018     std::thread recv_thread_;
00019     std::thread flight_target_thread_;

```

```

00020
00021 public:
00022     Simulator();
00023     ~Simulator();
00024
00025     bool init();
00026     void start();
00027 };

```

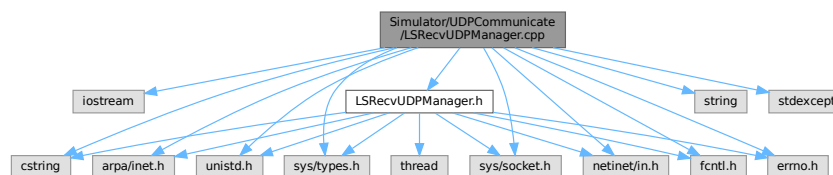
## 4.21 Simulator/UDPCommunicate/LSRecvUDPManager.cpp File Reference

```

#include <iostream>
#include <cstring>
#include <arpa/inet.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <errno.h>
#include <string>
#include <stdexcept>
#include "LSRecvUDPManager.h"

```

Include dependency graph for LSRecvUDPManager.cpp:



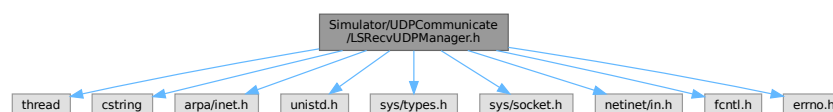
## 4.22 Simulator/UDPCommunicate/LSRecvUDPManager.h File Reference

```

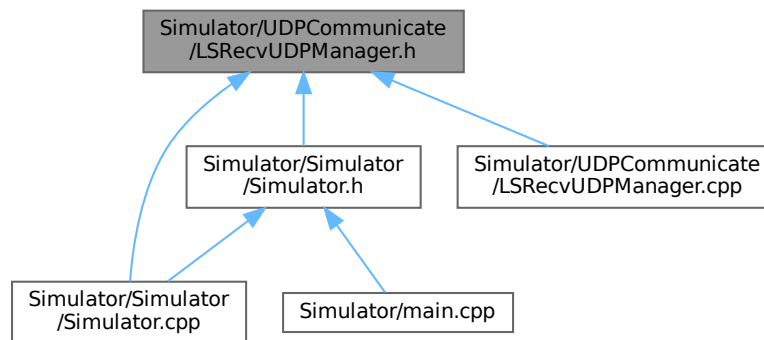
#include <thread>
#include <cstring>
#include <arpa/inet.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <errno.h>

```

Include dependency graph for LSRecvUDPManager.h:



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



## Classes

- class [LSRecvUDPManager](#)

## 4.23 LSRecvUDPManager.h

[Go to the documentation of this file.](#)

```

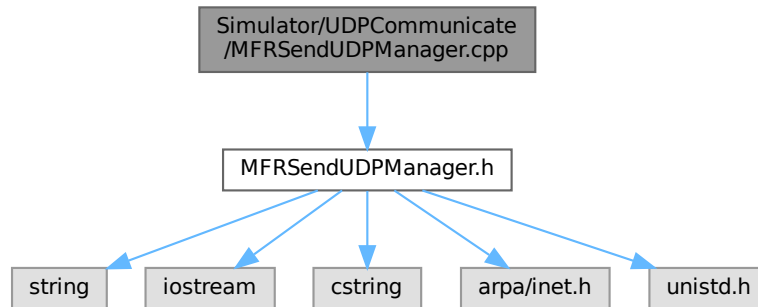
00001 #ifndef LS_RECV_UDP_MANAGER_H
00002 #define LS_RECV_UDP_MANAGER_H
00003
00004 #include <thread>
00005 #include <cstring>
00006 #include <arpa/inet.h>
00007 #include <unistd.h>
00008 #include <sys/types.h>
00009 #include <sys/socket.h>
00010 #include <netinet/in.h>
00011 #include <fcntl.h>
00012 #include <errno.h>
00013
00014 class LSRecvUDPManager
00015 {
00016 public:
00017     LSRecvUDPManager();
00018     ~LSRecvUDPManager();
00019
00020     // Initialize the UDP socket
00021     bool LSsocketOpen(int port);
00022
00023     // Receive data from the socket
00024     bool receiveData(char *buffer, int bufferSize);
00025
00026     // Close the socket
00027     void closeSocket();
00028
00029 private:
00030     int ls_socket_; // Socket file descriptor
00031     struct sockaddr_in server_addr_; // Server address structure
00032     socklen_t addr_len_; // Length of the address structure
00033     char buffer_[1024]; // Buffer for receiving data
00034     int buffer_size_; // Size of the buffer
00035     int recv_len_; // Length of the received data
00036     std::thread recv_thread_; // Thread for receiving data
00037 };
00038
00039 #endif // LS_RECV_UDP_MANAGER_H

```

## 4.24 Simulator/UDPCommunicate/MFRSendUDPManager.cpp File Reference

```
#include "MFRSendUDPManager.h"
```

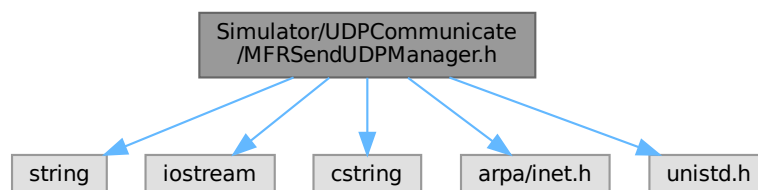
Include dependency graph for MFRSendUDPManager.cpp:



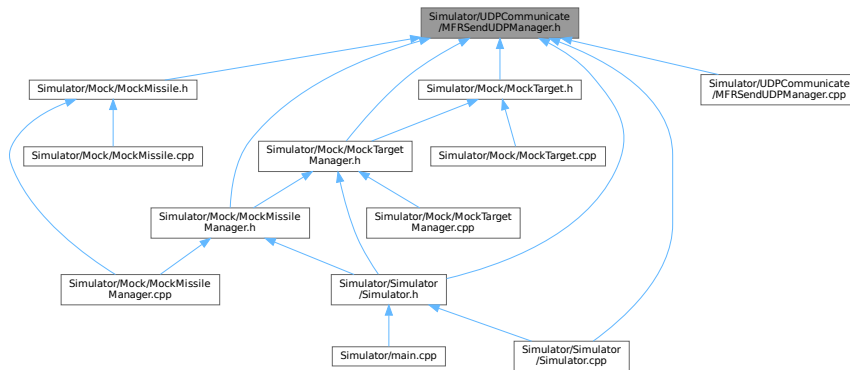
## 4.25 Simulator/UDPCommunicate/MFRSendUDPManager.h File Reference

```
#include <string>
#include <iostream>
#include <cstring>
#include <arpa/inet.h>
#include <unistd.h>
```

Include dependency graph for MFRSendUDPManager.h:



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



## Classes

- class [MFRSendUDPManager](#)

## 4.26 MFRSendUDPManager.h

[Go to the documentation of this file.](#)

```

00001 #ifndef MFR_SEND_UDP_MANAGER_H
00002 #define MFR_SEND_UDP_MANAGER_H
00003
00004 #include <string>
00005 #include <iostream>
00006 #include <cstring>
00007 #include <arpa/inet.h>
00008 #include <unistd.h>
00009
00010 class MFRSendUDPManager
00011 {
00012 private:
00013     int mfr_socket_;
00014     struct sockaddr_in client_addr_;
00015
00016 public:
00017     MFRSendUDPManager(/* args */);
00018     ~MFRSendUDPManager();
00019
00020     bool MFRSocketOpen(const std::string &ip, int port);
00021     bool sendData(const char *data, int dataSize);
00022 };
00023
00024 #endif // MFR_SEND_UDP_MANAGER_H

```



# Index

- ~LSRecvUDPManager
  - LSRecvUDPManager, [5](#)
- ~MFRSendUDPManager
  - MFRSendUDPManager, [7](#)
- ~MockMissile
  - MockMissile, [9](#)
- ~MockTarget
  - MockTarget, [11](#)
- ~Simulator
  - Simulator, [13](#)
- addTarget
  - MockTargetManager, [12](#)
- angle
  - MissileInfo, [8](#)
  - TargetInfo, [14](#)
- closeSocket
  - LSRecvUDPManager, [6](#)
- flightMissile
  - MockMissileManager, [10](#)
- flightTarget
  - MockTargetManager, [12](#)
- id
  - MissileInfo, [8](#)
  - TargetInfo, [14](#)
- init
  - Simulator, [13](#)
- LSRecvUDPManager, [5](#)
  - ~LSRecvUDPManager, [5](#)
  - closeSocket, [6](#)
  - LSRecvUDPManager, [5](#)
  - LSSocketOpen, [6](#)
  - receiveData, [6](#)
- LSSocketOpen
  - LSRecvUDPManager, [6](#)
- main
  - main.cpp, [15](#)
- main.cpp
  - main, [15](#)
- MFRSendUDPManager, [6](#)
  - ~MFRSendUDPManager, [7](#)
  - MFRSendUDPManager, [7](#)
  - MFRSocketOpen, [7](#)
  - sendData, [7](#)
- MFRSocketOpen
  - MFRSendUDPManager, [7](#)
- MissileInfo, [7](#)
  - angle, [8](#)
  - id, [8](#)
  - mockType, [8](#)
  - speed, [8](#)
  - x, [8](#)
  - y, [8](#)
  - z, [8](#)
- MockMissile, [9](#)
  - ~MockMissile, [9](#)
  - MockMissile, [9](#)
  - updatePosMissile, [9](#)
- MockMissileManager, [10](#)
  - flightMissile, [10](#)
  - MockMissileManager, [10](#)
- MockTarget, [10](#)
  - ~MockTarget, [11](#)
  - MockTarget, [11](#)
  - updatePos, [11](#)
- MockTargetManager, [11](#)
  - addTarget, [12](#)
  - flightTarget, [12](#)
  - MockTargetManager, [11](#)
  - RaedTargetIni, [12](#)
- mockType
  - MissileInfo, [8](#)
  - TargetInfo, [14](#)
- RaedTargetIni
  - MockTargetManager, [12](#)
- receiveData
  - LSRecvUDPManager, [6](#)
- sendData
  - MFRSendUDPManager, [7](#)
- Simulator, [12](#)
  - ~Simulator, [13](#)
  - init, [13](#)
  - Simulator, [13](#)
  - start, [13](#)
- Simulator/main.cpp, [15](#)
- Simulator/Mock/info/MissileInfo.h, [16](#), [17](#)
- Simulator/Mock/info/TargetInfo.h, [17](#), [18](#)
- Simulator/Mock/MockMissile.cpp, [18](#)
- Simulator/Mock/MockMissile.h, [19](#), [20](#)
- Simulator/Mock/MockMissileManager.cpp, [20](#)
- Simulator/Mock/MockMissileManager.h, [21](#)
- Simulator/Mock/MockTarget.cpp, [22](#)
- Simulator/Mock/MockTarget.h, [22](#), [23](#)
- Simulator/Mock/MockTargetManager.cpp, [24](#)

- Simulator/Mock/MockTargetManager.h, [24](#), [25](#)
- Simulator/Simulator/Simulator.cpp, [26](#)
- Simulator/Simulator/Simulator.h, [26](#), [27](#)
- Simulator/UDPCommunicate/LsRecvUDPManager.cpp,  
[28](#)
- Simulator/UDPCommunicate/LsRecvUDPManager.h,  
[28](#), [29](#)
- Simulator/UDPCommunicate/MFRSendUDPManager.cpp,  
[30](#)
- Simulator/UDPCommunicate/MFRSendUDPManager.h,  
[30](#), [31](#)
- speed
  - MissileInfo, [8](#)
  - TargetInfo, [14](#)
- start
  - Simulator, [13](#)
- TargetInfo, [14](#)
  - angle, [14](#)
  - id, [14](#)
  - mockType, [14](#)
  - speed, [14](#)
  - x, [14](#)
  - y, [14](#)
  - z, [14](#)
- updatePos
  - MockTarget, [11](#)
- updatePosMissile
  - MockMissile, [9](#)
- x
  - MissileInfo, [8](#)
  - TargetInfo, [14](#)
- y
  - MissileInfo, [8](#)
  - TargetInfo, [14](#)
- z
  - MissileInfo, [8](#)
  - TargetInfo, [14](#)