Fusion Radar Project

1.0.0

Generated by Doxygen 1.9.6

Chapter 1

Fusion Radar Project

Fusion-Radar Projesi README

Chapter 2

README

Fusion-Radar-Project

Demonstration of Server-Client project with some boost,cmake tools for demo!
Explore the docs »

View Demo · Report Bug · Request Feature

Table of Contents

2.0.1 About The Project

Kullanılacak Teknolojiler: ∗VSCode ∗Boost Kütüphanesi ∗NLohman Json ∗ZeroMQ ∗Docker ∗CMake ∗WSL ∗↩ Jenkins Test Automation?

Proje Gereksinimi: 2 Adet Uygulama yazılımı. 1 adet Radar , 1 adet Füzyon.

App1: -A.noktasından B noktasına 2 saniyede bir veri TCP/IP üzerinden veri aktarımı yapılacaktır. -Radar konumuna göre enu veya ecef output json formatında gönderilecek. -Veri hassasiyeti 10^-6 olacaktır. -Loglama yapılacak.

App2: -Json Output veri alacak. -App 1 den konum sorgusu çekecektir. -Bu konuma göre App 2 data işlenecek ve 0.5 saniyede bir konum gönderecek(LLA). -Loglama yapılacak.

Proje devamında web servis üzerinde çalışma yapılması ve test otomasyonu üzerinde geliştirme denemesi düşünülmektedir.

(back to top)

2.0.1.1 Built With

This section should list any major frameworks/libraries used to bootstrap your project. Leave any add-ons/plugins for the acknowledgements section. Here are a few examples.

- C++
- VSCode
- Boost
- Json
- CMake
- GCC/MinGW

(back to top)

2.0.2 Getting Started

This is an example of how you may give instructions on setting up your project locally. To get a local copy up and running follow these simple example steps.

2.0.2.1 Prerequisites

This is an example of how to list things you need to use the software and how to install them.

1. VSCode Extension CMake indir
2. VSCode Extension CMake Tools indir
3. VSCode Extension WSL indir
4. Microsoft Store Ubuntu 20.04 indir
5. Ubuntu Kurulumu sonrası komutlar
```
sudo apt-get update
sudo apt-get install build-essential
sudo apt-get install cmake
sudo apt-get install libboost-all-dev
sudo apt-get install code
```

2.0.2.2 Installation

Below is an example of how you can instruct your audience on installing and setting up your app. This template doesn't rely on any external dependencies or services.

1. Get a free API Key at https://github.com/CagriDK/FusionRadar-Project
2. Clone the repo
```
git clone
[https://github.com/your_username_/Project-Name.git](https://github.com/CagriDK/FusionRadar-Project.git)
```
3. Git Clone sonrası VSCode ile dosya açılır.
4. Clone dosya içerisine terminalde
```
mkdir build
cd .\build
cmake -S ..\ -B .
```
5. VSCode içerisinde Control+P >CMake Clear/Rebuild seçilerek derlenir.

(back to top)

2.0.3 Usage

Use this space to show useful examples of how a project can be used. Additional screenshots, code examples and demos work well in this space. You may also link to more resources.

For more examples, please refer to the Documentation

(back to top)

2.0.4 Roadmap

- [x] Add Changelog
- [x] Add back to top links
- [ ] Add Additional Templates w/ Examples
- [ ] Add "components" document to easily copy & paste sections of the readme
- [ ] Multi-language Support
  - [ ] Chinese
  - [ ] Spanish

See the `open issues` for a full list of proposed features (and known issues).

(back to top)

2.0.5 Contributing

Contributions are what make the open source community such an amazing place to learn, inspire, and create. Any contributions you make are **greatly appreciated**.

If you have a suggestion that would make this better, please fork the repo and create a pull request. You can also simply open an issue with the tag "enhancement". Don't forget to give the project a star! Thanks again!

1. Fork the Project
2. Create your Feature Branch (`git checkout -b feature/AmazingFeature`)
3. Commit your Changes (‘git commit -m 'Add some AmazingFeature’)
4. `Push to the Branch` (git push origin feature/AmazingFeature`)
5. Open a Pull Request

(back to top)

2.0.6 License

Distributed under the MIT License. See `LICENSE.txt` for more information.

(back to top)

2.0.7 Contact

Çağrı Dükünlü - `@cagrideka -cagridknl@gmail.com`

Project Link: `FusionRadar-Project`

(back to top)

2.0.8 Acknowledgments

Software Team Leader - Ali Emre Tunca

Software Engineer - Çağrı Dükünlü

Software Engineer - Zeynep Dilan Kılıç

(back to top)

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Class Documentation

5.1 TcpClient Class Reference

TcpClient_Sınıfı

`#include <Client.h>`

Public Member Functions

- TcpClient (const std::string &host, unsigned short port)

  Construct a new Tcp Client:: Tcp Client object TcpClient Ctor çağırıldığında "personClient.json" oluşturulur. Bir timer aracılığı ile bağlantı yapılana kadar bekleme yapılır. Bağlantı yapıldıktan sonra döngüden çıkar.
- void send (const std::string &message)

  Sockete veri yazdırma bloğu async_write fonksiyonu ile buffera arguman olarak gönderilen mesaj yazdırılır. Mesajın ulaşamama durumunda hata komutu ekrana yazdırılır.
- json getJData () const

  Json dönderen blok Serverden gelen json verisini geri dönderir.

5.1.1 Detailed Description

TcpClient_Sınıfı

5.1.2 Constructor & Destructor Documentation

5.1.2.1 TcpClient()

```
TcpClient::TcpClient (
            const std::string & host,
            unsigned short port )
```

Construct a new Tcp Client:: Tcp Client object TcpClient Ctor çağırıldığında "personClient.json" oluşturulur. Bir timer aracılığı ile bağlantı yapılana kadar bekleme yapılır. Bağlantı yapıldıktan sonra döngüden çıkar.

Parameters

host	// Hangi Adrese bağlantı yapılacağı
port	// Hangi Porta bağlantı yapılacağı

5.1.3 Member Function Documentation

5.1.3.1 getJData()

```
json TcpClient::getJData ( ) const  [inline]
```

Json dönderen blok Serverden gelen json verisini geri dönderir.

Returns

json

5.1.3.2 send()

```
void TcpClient::send (
            const std::string & message )
```

Sockete veri yazdırma bloğu async_write fonksiyonu ile buffera arguman olarak gönderilen mesaj yazdırılır. Mesajın ulaşamama durumunda hata komutu ekrana yazdırılır.

Parameters

message	

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

- GitProject/FusionRadar-Project/Client.h
- GitProject/FusionRadar-Project/Client.cpp

5.2 TcpServer Class Reference

TcpServer_Sınıfı

```
#include <Server.h>
```

Public Member Functions

- TcpServer (unsigned short port)

  Construct a new Tcp Server object JsonReaderWriter ile "personServer.json" oluşturulur. jData içerisine içerik kayıt edilir. Server bağlantısı başlatılır.
- void stop ()

  Manuel Server kapatma.
- bool getm_IsConnected ()

  Server-Socket bağlantı kontrolü

5.2.1 Detailed Description

TcpServer_Sınıfı

5.2.2 Constructor & Destructor Documentation

5.2.2.1 TcpServer()

```
TcpServer::TcpServer (
            unsigned short port )
```

Construct a new Tcp Server object JsonReaderWriter ile "personServer.json" oluşturulur. jData içerisine içerik kayıt edilir. Server bağlantısı başlatılır.

Parameters

port	

5.2.3 Member Function Documentation

5.2.3.1 getm_IsConnected()

```
bool TcpServer::getm_IsConnected ( )
```

Server-Socket bağlantı kontrolü

Returns

true

false

5.2.3.2 stop()

```
void TcpServer::stop ( )
```

Manuel Server kapatma.

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

- GitProject/FusionRadar-Project/Server.h
- GitProject/FusionRadar-Project/Server.cpp

5.3 Test Class Reference

Test_Sınıfı

```
#include <TestFunctions.h>
```

Public Member Functions

- Test ()
  Construct a new Test object.

Static Public Member Functions

- template<typename T >
  static T sumup (T a, T b)
  Toplama Test Methodu.
- template<typename T >
  static T subup (T a, T b)
  Çıkarma Test Methodu.
- template<typename T >
  static T divideup (T a, T b)
  Bölme Test Methodu.
- template<typename T >
  static T modup (T a, T b)
  Modunu Alma Test Methodu.

5.3.1 Detailed Description

Test_Sınıfı

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Test()

```
Test::Test ( )
```

Construct a new Test object.

5.3.3 Member Function Documentation

5.3.3.1 divideup()

```
template<typename T >
static T Test::divideup (
            T a,
            T b )  [inline], [static]
```

Bölme Test Methodu.

Template Parameters

T	

Parameters

a	
b	

Returns

T

5.3.3.2 modup()

```
template<typename T >
static T Test::modup (
            T a,
            T b )  [inline], [static]
```

Modunu Alma Test Methodu.

Template Parameters

T	

Parameters

a	
b	

Returns

T

5.3.3.3 subup()

```
template<typename T >
static T Test::subup (
            T a,
            T b )  [inline], [static]
```

Çıkarma Test Methodu.

Template Parameters

T	

Parameters

a	
b	

Returns

T

5.3.3.4 sumup()

```
template<typename T >
static T Test::sumup (
            T a,
            T b )  [inline], [static]
```

Toplama Test Methodu.

Template Parameters

T	

Parameters

a	
b	

Returns

T

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

- GitProject/FusionRadar-Project/test/TestFunctions.h
- GitProject/FusionRadar-Project/test/TestFunctions.cpp

Chapter 6

File Documentation

6.1 GitProject/FusionRadar-Project/Client.cpp File Reference

```
#include <Client.h>
#include <pthread.h>
#include <unistd.h>
```

Functions

- void * clientThread (void *arg)

  PThreadın çalışma fonksiyonudur Threadın çalışma aktivitesi tanımlanmıştır. Her 1 saniyede bir (Toplam 10 saniye) boyunca tempMessage sockete yazdırılır.
- int main (int, char **)

  Ana fonksiyon bloğu Thread oluşturulur, TcpClient oluşturulur ve thread ana bloğun bitimi ile tamamlanır.

6.1.1 Function Documentation

6.1.1.1 clientThread()

```
void * clientThread (
            void * arg )
```

PThreadın çalışma fonksiyonudur Threadın çalışma aktivitesi tanımlanmıştır. Her 1 saniyede bir (Toplam 10 saniye) boyunca tempMessage sockete yazdırılır.

Parameters

arg	PThread function pointer argumanı

Returns

void∗

6.1.1.2 main()

```
int main (
            int ,
            char **  )
```

Ana fonksiyon bloğu Thread oluşturulur, TcpClient oluşturulur ve thread ana bloğun bitimi ile tamamlanır.

Returns

int

6.2 GitProject/FusionRadar-Project/Client.h File Reference

```
#include <boost/asio.hpp>
#include <iostream>
#include <json_reader_writer.h>
```

Classes

- class TcpClient

  TcpClient_Sınıfı

6.3 Client.h

Go to the documentation of this file.

```
00001 #pragma once
00002 #include <boost/asio.hpp>
00003 #include <iostream>
00004 #include <json_reader_writer.h>
00005
00006 using namespace boost::asio;
00011 class TcpClient {
00012 public:
00021     TcpClient(const std::string& host, unsigned short port);
00022
00029     void send(const std::string& message);
00030
00036     json getJData()const{
00037         return jData;
00038     };
00039 private:
00046     void receive();
00047
00052     io_service m_ioService;
00053
00058     ip::tcp::socket m_socket;
00059
00064     boost::asio::streambuf m_buffer;
00065
00070     boost::system::error_code ec;
00071
00076     json jData;
00077 };
```

6.4 GitProject/FusionRadar-Project/DoxyFile.md File Reference

6.5 GitProject/FusionRadar-Project/main.cpp File Reference

```
#include <iostream>
#include <boost/asio.hpp>
#include <Server.h>
#include <pthread.h>
```

Functions

- void * ServerThread (void *arg)

  PThread fonksiyon tanımı yapılan blok Threadın çalışması bir döngüye alınmıştır. Bağlantı kurulana kadar threadı bekletir. Eğer bağlantı kapandı mesajı gelir ise (getm_IsConnected) thread kapatılır. Serverden socketden okunan mesaj tekrardan karşı tarafa yazdırılır. Yazdırma null yada false döner ise m_IsConnected false olur.
- int main (int, char **)

  Ana fonksiyon bloğu Thread ve TcpServer oluşturulur. Terminale "exit" yazdırılır ise manuel olarak server durdurulabilir.

6.5.1 Function Documentation

6.5.1.1 main()

```
int main (
            int ,
            char **  )
```

Ana fonksiyon bloğu Thread ve TcpServer oluşturulur. Terminale "exit" yazdırılır ise manuel olarak server durdurulabilir.

Returns

int

6.5.1.2 ServerThread()

```
void * ServerThread (
            void * arg )
```

PThread fonksiyon tanımı yapılan blok Threadın çalışması bir döngüye alınmıştır. Bağlantı kurulana kadar threadı bekletir. Eğer bağlantı kapandı mesajı gelir ise (getm_IsConnected) thread kapatılır. Serverden socketden okunan mesaj tekrardan karşı tarafa yazdırılır. Yazdırma null yada false döner ise m_IsConnected false olur.

Parameters

arg	

Returns

void*

6.6 GitProject/FusionRadar-Project/README.md File Reference

6.7 GitProject/FusionRadar-Project/Server.cpp File Reference

```
#include "Server.h"
```

6.8 GitProject/FusionRadar-Project/Server.h File Reference

```
#include <boost/asio.hpp>
#include <iostream>
#include <string>
#include <include/json_reader_writer.h>
```

Classes

- class TcpServer

  TcpServer_Sınıfı

6.9 Server.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include <boost/asio.hpp>
00004 #include <iostream>
00005 #include <string>
00006 #include <include/json_reader_writer.h>
00007
00012 class TcpServer
00013 {
00014 public:
00022     TcpServer(unsigned short port);
00023
00028     void stop();
00029
00036     bool getm_IsConnected();
00037
00038 private:
00043     void start_accept();
00044
00050     void handle_accept(const boost::system::error_code &error);
00051
00056     void start_read();
00057
```

```
00065     void handle_read(const boost::system::error_code &error, size_t bytes_transferred);
00066
00072     void start_write(std::string message);
00073
00079     void handle_write(const boost::system::error_code &error);
00080
00085     bool m_IsConnected;
00086
00091     boost::asio::io_service m_ioService;
00092
00097     boost::asio::ip::tcp::acceptor m_acceptor;
00098
00103     boost::asio::ip::tcp::socket m_socket;
00104
00109     boost::asio::streambuf m_buffer;
00110
00111 private:
00116     json jData;
00117 };
```

6.10 GitProject/FusionRadar-Project/test/TestFunctions.cpp File Reference

```
#include <TestFunctions.h>
```

6.11 GitProject/FusionRadar-Project/test/TestFunctions.h File Reference

Classes

- class Test

  Test_Sınıfı

6.12 TestFunctions.h

Go to the documentation of this file.

```
00001
00005 class Test
00006 {
00007 public:
00012     Test();
00013
00022     template <typename T>
00023     static T sumup(T a, T b)
00024     {
00025         T temp = a + b;
00026         return temp;
00027     }
00028
00037     template <typename T>
00038     static T subup(T a, T b)
00039     {
00040         T temp = a - b;
00041         return temp;
00042     }
00043
00052     template <typename T>
00053     static T divideup(T a, T b)
00054     {
00055         T temp = a / b;
00056         return temp;
00057     }
00058
00067     template <typename T>
00068     static T modup(T a, T b)
```

```
00069     {
00070         T temp = a % b;
00071         return temp;
00072     }
00073
00074 private:
00079     int m_x;
00080 };
```