

TCP Socket Example 2

- In this example, we will learn how to use **QTcpSocket**. We're going to use Signal and Slot mechanism instead of calling functions manually
- Create project type **Application->Qt Console Application** named **MyTCPSocketWithSignals**
- Add network module to project file: **QT += network**.
- Run **qmake (Build->Run qmake)** and Build project! You can close project file.
- Add C++ Class **MyTCPSocketClass** to project as we did in example 1

IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

```
#include <QObject>
#include <QTcpSocket>
#include <QDebug>

class MyTCPSocketClass : public QObject
{
    Q_OBJECT
public:
    MyTCPSocketClass(QObject *parent = nullptr);
    ~MyTCPSocketClass();
    void connectToServer();

public slots:
    void myConnectedSlot();
    void myDisconnectedSlot();
    void myBytesWrittenSlot(qint64 bytes);
    void myReadyReadSlot();

signals:
    void finished();

private:
    QTcpSocket *socket;
};
```

TCP Socket Example 2 (continues...)

- Open **mytcpsocketclass.h** and **add lines which are red** to file

TCP Socket Example 2 (continues...)

- Open file **mytcpsocketclass.cpp** and add lines below to constructor function

```
socket = new QTcpSocket();  
qDebug() << "1: Socket Created" << endl;  
  
connect(socket, SIGNAL(connected()),this, SLOT(myConnectedSlot()));  
connect(socket, SIGNAL(disconnected()),this, SLOT(myDisconnectedSlot()));  
connect(socket, SIGNAL(bytesWritten(qint64)),this, SLOT(myBytesWrittenSlot(qint64)));  
connect(socket, SIGNAL(readyRead()),this, SLOT(myReadyReadSlot()));
```

TCP Socket Example 2 (continues...)

- Create implementation of destructor function and add lines below in it

```
delete socket;  
socket = nullptr;  
qDebug() << "10: Socket Deleted";
```

- Create skeleton implementation of function **void connectToServer()**
- Create skeleton implementations of **all** public slots functions in the class
- Build the project!

TCP Socket Example 2 (continues...)

- Write implementation of function **void connectToServer** as below

```
void MyTCPSocketClass::connectToServer ()
{
    qDebug() << "2: Connecting To Server";
    socket->connectToHost("oamk.fi", 80);
    if(!socket->waitForConnected(3000))
    {
        qDebug() << "Error: " << socket->errorString();
        socket->deleteLater();
        exit(0);
    }
}
```



<http://doc.qt.io/qt-5/qobject.html#deleteLater>

TCP Socket Example 2 (continues...)

- Write implementation of function **void myConnectedSlot()** as below

```
void MyTCPSocketClass::myConnectedSlot ()
{
    qDebug() << "3: Client Connected To Server" << endl;
    qDebug() << "4: Write message To Server.";
    socket->write("HEAD / HTTP/1.0\r\n\r\n\r\n\r\n\r\n");
}
```

- Write implementation of function **void myDisconnectedSlot()** as below

```
void MyTCPSocketClass::myDisconnectedSlot ()
{
    qDebug() << "8: Client Disconnected from Server";
    emit finished();
}
```

TCP Socket Example 2 (continues...)

- Write implementation of function **void myBytesWrittenSlot (qint64 bytes)** as below

```
void MyTCPSocketClass::myBytesWrittenSlot (qint64 bytes)
{
    qDebug() << "5:" << bytes << " bytes written." << endl;
    qDebug() << "6: Wait for Server to answer..." << endl;
}
```

- Write implementation of function **void myReadyReadSlot()** as below

```
void MyTCPSocketClass::myReadyReadSlot ()
{
    qDebug() << "7: Reading data from Server";
    qDebug() << socket->readAll() << endl;
    qDebug() << "8: Close Client Socket" << endl;
    socket->close();
}
```

IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

```
#include <QCoreApplication>
#include "mytcpsocketclass.h"

int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);

    MyTCPSocketClass *objectMyTCPSocketClass;
    objectMyTCPSocketClass = new MyTCPSocketClass;

    QObject::connect(objectMyTCPSocketClass, SIGNAL(finished()), &a, SLOT(quit()));

    objectMyTCPSocketClass->connectToServer();

    int eventLoop;
    eventLoop = a.exec();

    delete objectMyTCPSocketClass;
    objectMyTCPSocketClass= nullptr;

    return eventLoop;
}
```

- Build and run the project!

TCP Socket Example 2 (continues...)

- Open **main.cpp** and code it as code on the left