Serial

main.cpp

```
#include <QCoreApplication>
#include <QObject>

#include "serial.h"

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

    serial *s = new serial();

    QObject::connect(s, SIGNAL(sig_quit()), &a, SLOT(quit()));

    s->open("COM3");

    s->write("Hello World!");

    return a.exec();
}
```

serial.h

```
#ifndef SERIAL_H
#define SERIAL_H

#include <iostream>

#include <QObject>
#include <QSerialPort>

class serial : public QObject
{
    Q_OBJECT
public:
    explicit serial(QObject *parent = nullptr);

    void open(const QString &name);
    void write(const QString &message);

public slots:
    void slot_messageWritten();

private:
    QSerialPort *serialPort;
signals:
    void sig_quit();
};

#endif // SERIAL_H
```

serial.cpp

```
#include "serial.h"

serial::serial(QObject *parent) : QObject(parent)
{
    serialPort = new QSerialPort();

    connect(serialPort, SIGNAL(bytesWritten(qint64)), this,
SLOT(slot_messageWritten()));

    return;
}

void serial::open(const QString &name)
{

    serialPort->setPortName(name);
    serialPort->setBaudRate(115200);
    serialPort->setDataBits(QSerialPort::Data8);
    serialPort->setParity(QSerialPort::NoParity);
    serialPort->setStopBits(QSerialPort::OneStop);

    if (!serialPort->isOpen())
    {
        serialPort->close();
    }

    serialPort->open(QIODevice::ReadWrite);

    return;
}

void serial::write(const QString &message)
{

    std::cout << "Message: " << message.toStdString() << std::endl;

    int bytesWritten = serialPort->write(message.toUtf8());

    std::cout << "Bytes written: " << bytesWritten << std::endl;

    return;
}
```

```
void serial::slot_messageWritten()
{

    serialPort->close();

    emit sig_quit();

    return;
}
```