

# **Wprowadzenie do biblioteki Qt w C++**

**SZPC++ #7**

*Tomasz Pietrzak*

# Parametry transmisji szeregowej

- Baudrate (9600, 57600, 115200 bps)
- Databits (6, 7, 8)
- Parity (N, E, O)
- Stop bits (1, 2)
- Flow control (No, Hardware, Software)

# Ramka danych



- Bit startu
- Bity danych (B1 ... B8)
- Bit parzystości (P1)
- Bit(y) stopu

# Skrótowy zapis parametrów transmisji

57600 8-N-1

- Baudrate (9600, 57600, 115200 bps)
- Databits (6, 7, 8)
- Parity (N, E, O)
- Stop bits (1, 2)

# QSerialPort

```
#include <QSerialPort/QSerialPort>

QSerialPort szeregowy;

// Ustawiamy 1200 7-N-2
szeregowy.setPortName ("/dev/ttyUSB0");
szeregowy.open (QIODevice::ReadWrite);
szeregowy.setBaudRate (QSerialPort::Baud1200);
szeregowy.setDataBits (QSerialPort::Data7);
szeregowy.setStopBits (QSerialPort::TwoStop);
szeregowy.setParity (QSerialPort::NoParity);
szeregowy.setFlowControl (QSerialPort::NoFlowControl);

// (...)

szeregowy.close ();
```

<http://doc.qt.io/qt-5/qserialport-members.html>

# QSerialPort::setBaudRate ()

QSerialPort::Baud1200

QSerialPort::Baud2400

QSerialPort::Baud4800

QSerialPort::Baud9600

QSerialPort::Baud19200

QSerialPort::Baud38400

QSerialPort::Baud57600

QSerialPort::Baud115200

<http://doc.qt.io/qt-5/qserialport.html#BaudRate-enum>



# QSerialPort::setDataBits ()

QSerialPort::Data5

QSerialPort::Data6

QSerialPort::Data7

QSerialPort::Data8

<http://doc.qt.io/qt-5/qserialport.html#DataBits-enum>

# QSerialPort::setStopBits ()

QSerialPort::OneStop

QSerialPort::OneAndHalfStop

QSerialPort::TwoStop

<http://doc.qt.io/qt-5/qserialport.html#StopBits-enum>



# QSerialPort::setParity ()

// No parity bit is sent. This is the most common parity setting. Error detection is handled by the communication protocol.

QSerialPort::NoParity

// The number of 1 bits in each character, including the parity bit, is always even.

QSerialPort::EvenParity

// The number of 1 bits in each character, including the parity bit, is always odd. It ensures that at least one state transition occurs in each character.

QSerialPort::OddParity

<http://doc.qt.io/qt-5/qserialport.html#Parity-enum>

# QSerialPort::setFlowControl ()

```
// No flow control.
```

```
QSerialPort::NoFlowControl
```

```
// Hardware flow control (RTS/CTS).
```

```
QSerialPort::HardwareControl
```

```
// Software flow control (XON/XOFF).
```

```
QSerialPort::SoftwareControl
```

<http://doc.qt.io/qt-5/qserialport.html#FlowControl-enum>

# Zapis / odczyt

```
// Pisanie do portu
szeregowy.write ("?1\r");
szeregowy.waitForBytesWritten (1000);           // czeka max 1000 ms

// Czytanie z portu
char w = 0;
szeregowy.waitForReadyRead (1000);              // czeka max 1000 ms
szeregowy.getChar (&w);

// Albo
char bufor [32];
szeregowy.waitForReadyRead (1000);              // czeka max 1000 ms
szeregowy.read (bufor, 31);                     // może nie działać zgodnie z oczekiwaniami!

// Albo
if (szeregowy.canReadLine ())
szeregowy.readLine (bufor, 31);                 // może nie działać zgodnie z oczekiwaniami!
```

# Odczyt linii

```
while (!szeregowy.canReadLine ())
{
    std::this_thread::sleep_for (std::chrono::milliseconds (100));
    QApplication::processEvents ();
}
QByteArray odczyt = szeregowy.readLine ();

std::string tekst = odczyt.toStdString ();
```

# Komunikacja z urządzeniami GPIB



**Dziękuję za uwagę!**

