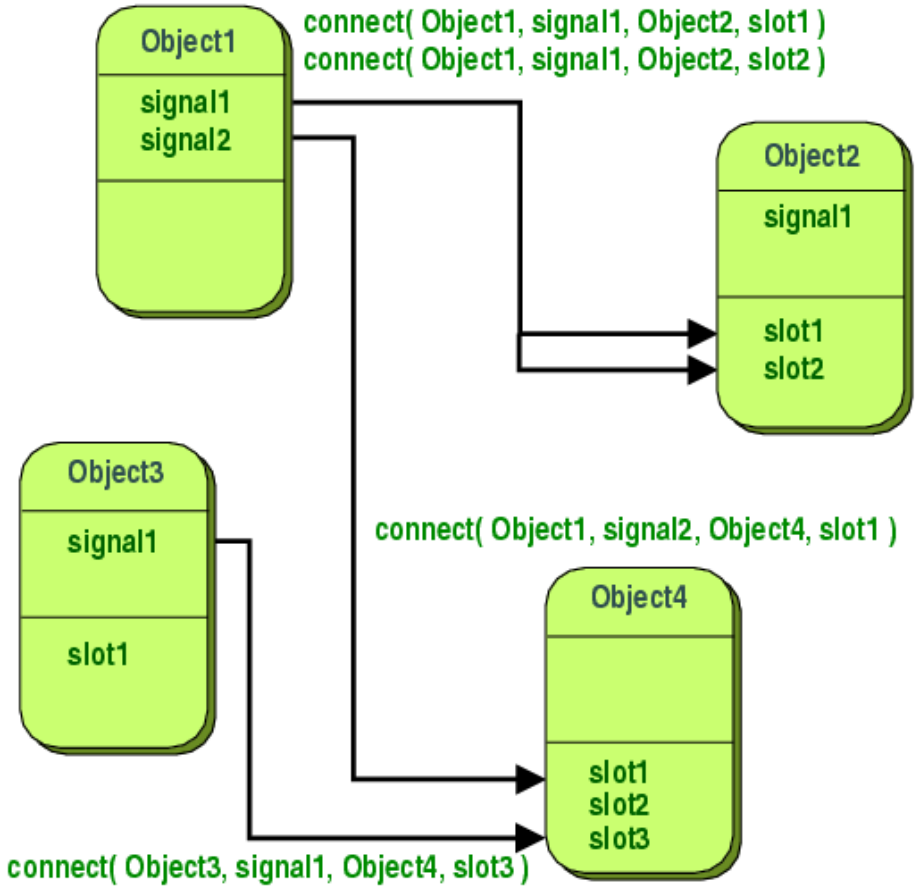
IEEE | IPCA Student Branch

Qt Workshop Sabado 05 Dez.

By António Real

Conceitos básicos e conectividade com hardware

Signals e slots



- Um **signal** é emitido quando ocorre um evento em particular (por exemplo, clique de um botão). Os Widgets do Qt têm vários sinais predefinidos mas é possível adicionar os nossos próprios sinais através de subclasses ou até mesmo adicionar sinais a classes que não sejam widgets.

- Um **slot** é uma função que é chamada em resposta a um sinal. Os widgets do Qt também têm vários slots pré-definidos mas de igual forma aos signals, conseguimos criar os nossos próprios slots a esses widgets ou às nossas classes.

Um signal pode ser ligado a um ou mais slots e um slot também pode ligado a vários signals.

Signals e slots

```
signals:
    void mySignal(int foo, int bar);

public slots:
    void mySlot1(QString foo, QString bar);
    void mySlot2(int foo, int bar, int blahblah);
    void mySlot3(int foo, int bar);
    void mySlot4(int foo);
```

Ao emitir um signal é possível passar argumentos e o slot ao qual se liga pode recebê-los. No entanto o slot que se liga ao signal nunca pode receber mais argumentos do que o signal envia (mas pode receber menos)e os tipos dos argumentos têm de ser os mesmos.

Ligando “mySignal” aos vários slots, teríamos as seguintes situações:

- Slot1 daria error porque os tipos dos argumentos são incompatíveis
- Slot2 daria erro porque tenta receber mais argumentos do que o signal emite.
- Slot3 e Slot4 funcionam pois o tipo de argumentos são compatíveis e o número de argumentos que recebem é igual ou menor aos que o signal emite

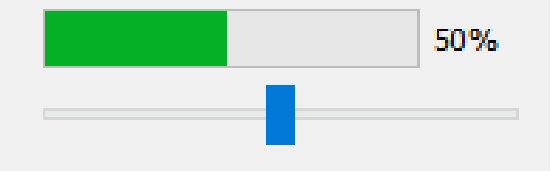
Requesitos para uso de signals e slots

```
#include <QObject>

class Counter : public QObject
{
    Q_OBJECT

public:
    Counter() { m_value = 0; }

    int value() const { return m_value; }

public slots:
    void setValue(int value);

signals:
    void valueChanged(int newValue);

private:
    int m_value;
};
```

- Para que uma determinada classe seja capaz de utilizar o mecanismo de signals e slots esta tem de, no mínimo, ser uma subclasse de QObject e ter na sua definição a macro Q_OBJECT como no exemplo ao lado.

Sintaxe de signals e slots

Para fazer a ligação entre um signal e um slot, podem ser usadas duas sintaxes:




```
// Método 1
connect(ui->slider, SIGNAL(valueChanged(int)), ui->progressBar, SLOT(setValue(int)));
// Método 2
connect(ui->slider, &QSlider::valueChanged, ui->progressBar, &QProgressBar::setValue);
```

Ambos os métodos são válidos para fazer a conexão, no entanto o método 2 é preferível pois força o compilador a verificar se o slot é compatível com o signal no momento da compilação e, caso não se verifique, dá erro na compilação. No caso do método 1 a aplicação seria compilada e o erro apenas se daria durante a sua utilização.

Objetivos deste workshop

Neste workshop irá ser feita uma aplicação capaz de comunicar com um microcontrolador e receber dados deste (por exemplo, de um sensor de temperatura). A comunicação será feita utilizando o protocolo RS232 (ou porta série). A aplicação irá reunir várias funcionalidades:

- Receber dados do microcontrolador
- Criar diálogos para o utilizador introduzir dados
- Utilização do Qt Designer para fazer a interface
- Noções básicas de utilização do debugger
- Guardar dados em ficheiros de texto.

Exemplo de aplicação:

RS232
(Porta série)