

Instalacija, podešavanje i upoznavanje Qt radnog okvira

10.1. Qt radni okvir

Qt je C++ biblioteka specijalizovana za razvoj grafičkih korisničkih interfejsa. Pored standardnih funkcionalnosti C++ programskog jezika, u Qt su uvedeni i neki dodatni mehanizmi, kao što su signali i slotovi, o čemu će biti reči kasnije. Jedna od najvećih prednosti Qt je to što je *cross-platform*, što omogućava da aplikacije kreirane na jednoj platformi, mogu da se izvršavaju i na bilo kojoj drugoj platformi. Na ovim vežbama će se koristiti Qt5 verzija, koja se može instalirati kucanjem sledeće komande u terminalu:

```
sudo apt-get update
sudo apt-get install qt5-default
```

Iscrpne informacije o Qt5 verziji, njenim klasama i mogućnostima mogu se naći na [31].

Za projektovanje interfejsa upotrebom Qt se najčešće koristi *cross-platform* integrisano razvojno okruženje *Qt Creator*, takođe razvijeno od strane Qt tima. Više o *Qt Creator*-u se može naći u [32]. Ovo okruženje znatno olakšava razvoj GUI-a nudeći raznovrsne mogućnosti pri radu. Instalira se sledećom komandom:

```
sudo apt-get install qtcreator
```

Radno okruženje se može pokrenuti preko *Menu -> Programming -> Qt Creator* ili iz komandne linije kucanjem `qtcreator`. Ukoliko se početna strana sporo učitava, *Qt Creator* se može pokrenuti i komandom `qtcreator -noload Welcome`, čime se ona preskače. Pri prvom pokretanju je potrebno podesiti kompajler. To se može uraditi na sledeći način.

Ući u *Tools -> Options -> Build & Run -> Compilers*. Odabrati opciju *Manual*, kliknuti na *Add* i u padajućem meniju odabrati *GCC*. Zatim, kada se selektuje *GCC* u *Manual* listi, ispod liste se prikažu opcije, gde u *Compiler Path* treba ukucati `/usr/bin/gcc`

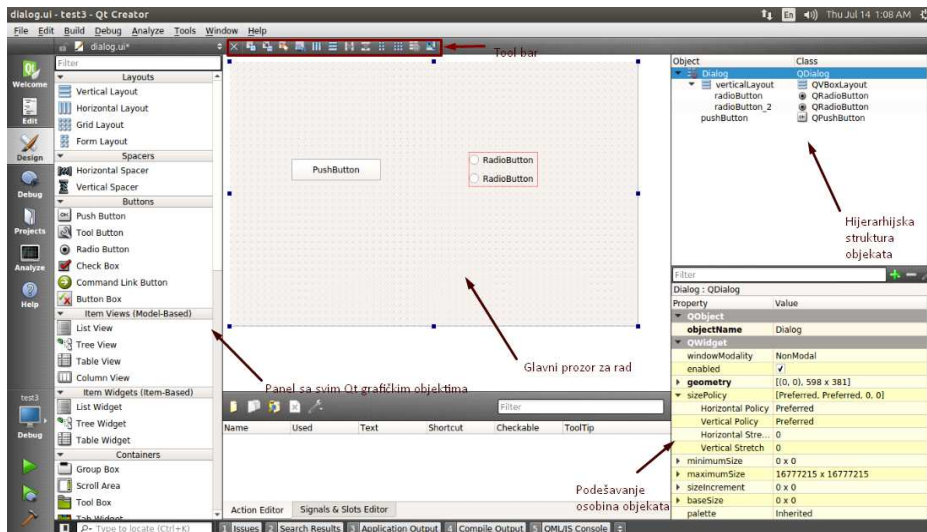
Nakon toga, u *Kits* (takođe u *Build & Run* meniju), odabrati *Desktop(default)* kit, i u njegovim opcijama, pri dnu, odabrati Qt verziju koja je instalirana. Pored toga podesiti kompajler odabirom *GCC* u *Compiler: C++:* polju.

Kreirati novi projekat preko *File -> New File or Project*, izabrati *Qt Widgets Application* i kliknuti *Choose*, zatim podesiti putanju, kliknuti *Next*, odabrati kit *Desktop*, kliknuti *Next*, nakon čega u padajućem meniju *Base Class* izabrati *QDialog* (to je najjednostavnija klasa, dok *QMainWindow* dolazi sa gotovim *tool*, *status* i *menu bar*-ovima), kliknuti još jednom *Next*, pa *Finish*.

Sada bi trebalo da se otvori tekst editor u kome se vidi kôd datoteke `dialog.cpp`, koji u početku sadrži samo implementaciju konstruktora i destruktoru klase `Dialog`. Testirati program klikom na *Run* u donjem levom uglu. Trebalo bi da se pojavi prazan prozor `Dialog`.

U *Design* mod se može ući duplim klikom na datoteku *Forms -> dialog.ui*. To je zapravo *xml* datoteka koja se može modifikovati isključivo u *Design* modu.

Grafički korisnički interfejs prikazan je na slici 10.1.



10.1. Qt Creator GUI

10.2. Signali i slotovi

Mehanizam signala i slotova je najvažnija karakteristika Qt radnog okvira, i on ga razlikuje od većine drugih alata slične namene. Za razliku od C++, gde klase sadrže samo atribute i metode, svaka Qt klasa ima i svoje signale i slotove. Ove komponente Qt služe za komunikaciju između objekata GUI.

Komunikacija se zasniva na principu događaja (eng. *event*), kao što su klik na taster, pomeranje slajdera i sl. Kada se desi neki događaj vezan za objekat, on emituje neki od svojih signala. Slot je funkcija koja predstavlja odgovor nekog objekta na emitovani signal, i projektant ima zadatak da ih poveže tako da se postigne željena funkcionalnost. Takođe, projektant može da definiše svoje signale i slotove unutar bilo koje klase. Definisani signali se onda mogu emitovati iz bilo koje druge metode ili slota te klase upotrebom ključne reči `emit`.

Signali i slotovi se nekad mogu povezati grafički u *Design* modu, ali u opštem slučaju potrebno je koristiti naredbu:

```
connect(const QObject *sender,
        SIGNAL(senders_signal(parameters)),
        const QObject *receiver,
        SLOT(receivers_slot(parameters)))
```

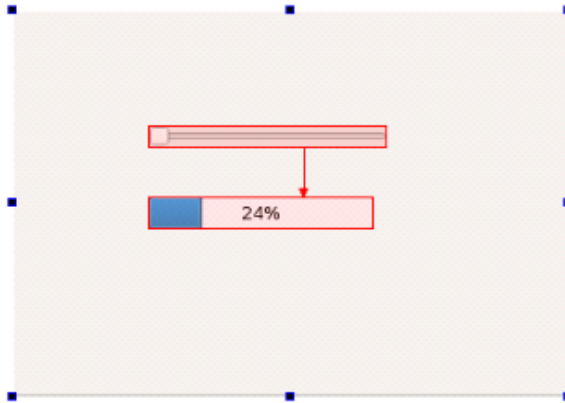
Povezani signal i slot moraju da imaju parametre istog tipa. Takođe, kada je potrebno implementirati svoje slotove, vrlo korisna je opcija *Go to slot...* koja se može aktivirati desnim klikom tastera miša na neki objekat u *Design* modu.

Detaljnije o signalima i slotovima može se naći na [33].

Primer 1

U ovom primeru je prikazana realizacija *slider* i *progress bar*-a, koji su povezani tako da stanje *progress bar*-a zavisi direktno od pozicije *slider*-a.

Nakon što se ubace ova dva elementa u glavni prozor u *Design* modu, odabrati opciju *Edit Signals/Slots* klikom na ikonicu na *toolbar*-u iznad. Zatim selektovati *slider* i prevući mišem do *progress bar*-a čime se dobija izgled kao na slici 10.2.



10.2. Kreiranje GUI

Odabrati signal `valueChanged(int)` od *slider*-a i slot `setValue(int)` od *progress bar*-a. U podešavanjima *progress bar*-a podesiti početnu vrednost na 0%. Pokrenuti program klikom na *Run* ili prečicom `Ctrl+R`, a izgled dobijenog dijaloga prikazan je na slici 10.3.



10.3. Izgled GUI

Drugi način povezivanja signala i slota je da se u datoteci `dialog.cpp` u konstruktor klase `Dialog` upiše:

```
connect(ui->horizontalSlider,
        SIGNAL(valueChanged(int)),
        ui->progressBar, SLOT(setValue(int)));
```

Napomena: U Qt-u postoji *parent-child* odnos (ili odnos posedovanja, odnos roditelj dete) između nekih objekata. Osnovna ideja je da su svi objekti koji se nalaze unutar nekog drugog objekta njegovi potomci (eng. *child*). Npr. svi objekti koji se nalaze grafički unutar objekta `ui` klase `Dialog` su njegovi *child* objekti i pristupa im se

ovako: `ui->imeObjekta`. Kada se zatvori objekat roditelj, tada se zatvaraju i svi njegovi potomci, tako da u njihovom slučaju ne mora da se vodi računa o oslobađanju memorije sa *heap*-a. Ovo nije isto što i nasleđivanje! Više o ovome se može pogledati na [34].

Zadatak 1

U ovom zadatku potrebno je projektovati GUI koji omogućava da na pritisak tastera pod nazivom „Toggle“, LED povezana sa RPi-em da menja svoje stanje. U podešavanjima tastera podesiti da bude tip "*checkable*" odabran.

Da bi se program kompajlirao, potrebno je dodati *wiringPi* biblioteku i putanju ka njoj na početak `imeProjekta.pro` datoteke:

```
LIBS += -L/usr/local/lib -lwiringPi
```

Nakon toga, potrebno je pozvati `wiringPiSetup()` i podesiti pin kao izlazni sa `pinMode(brojPina, OUTPUT)` na početku `main` funkcije u `main.cpp` datoteci. U *Design* modu ubaciti *push button* objekat, kliknuti desnim tasterom miša na njega i odabrati opciju *Go to slot...* Izabrati signal `clicked()`, nakon čega se otvara `dialog.cpp` datoteka sa praznim slotom `on_pushButton_clicked()`. Ovaj slot će se izvršiti kada *push button* bude kliknut i tu treba programirati traženo ponašanje LED, koristeći funkciju `digitalWrite(brojPina, vrednost)`.

Zadatak 2

Koristeći slider u GUI napraviti LED dimer.

10.3. Layout

Layout-i su korisni alati za grupisanje i poravnavanje objekata u glavnom prozoru GUI. Poravnavanje objekata se radi tako što se oni zajedno selektuju i iz *tool bar*-a klikne ikonica za neki od *layout*-a – horizontalni, vertikalni ili u vidu tabele (*grid*). Međusobna isključivost nekoliko *radio button*-a se postiže upravo njihovim grupisanjem u *layout*.

Zadatak 3

Napraviti GUI koji sadrži 4 prekidača (*checked push button*) gde svaki prekidač uključuje po jednu od LED na DVK512. Koristeći *Layouts* poravnati prekidače unutar dijaloga.

10.4. Dodavanje slika

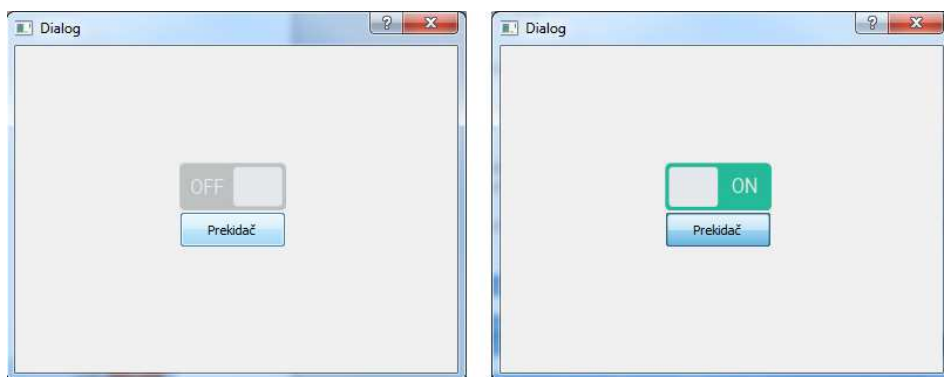
Za dodavanje slika u GUI, prvo je potrebno kopirati ih u projektni folder, a zatim

kreirati i dodati u projekat *resource file*, u kome će te slike biti smeštene. *Resource file* se može kreirati desnim klikom na ime projekta u panelu sa leve strane i odabiranjem opcije *Add New -> Qt -> Qt Resource File*. Potom treba dati ime i kliknuti *Finish*, nakon čega se otvara prazna `qrc` datoteka. Tada treba kliknuti *Add -> Add Prefix*, i dati ime prefiksu (npr. `./slike`). Zatim treba ići na *Add -> Add Files*, i dodati željene slike. Sada je u *Design* modu moguće napraviti labelu i preko njenog svojstva (*Property* u donjem desnom uglu) *pixmap* dodati sliku. Postoje takođe i opcije za veličinu slike i skaliranje.

Resource file-ovi se mogu koristiti i za razne druge namene, a iz koda se konkretnim datotekama pristupa preko putanje u obliku `"/prefix/fajl.ext"` (npr. `./slike/on.png`)

Zadatak 4

Nadograditi Zadatak 1 tako što u interfejsu treba dodati jednu labelu (*Label*) povezanu sa prekidačem i kojoj treba pridružiti jednu sliku za uključeno stanje prekidača i jednu za isključeno stanje prekidača (vidi sliku 10.4). Koristiti slot za događaj *clicked (bool)*. Promenu slike labele vršiti upotrebom `setPixmap` metode `QPixmap` klase. Za pristup elementima GUI se koristi globalni pokazivač `ui`.



10.4. Izgled GUI zadatka 4 zaviso od statusa prekidača