

Qt Creator

**IDE
&
Framework**





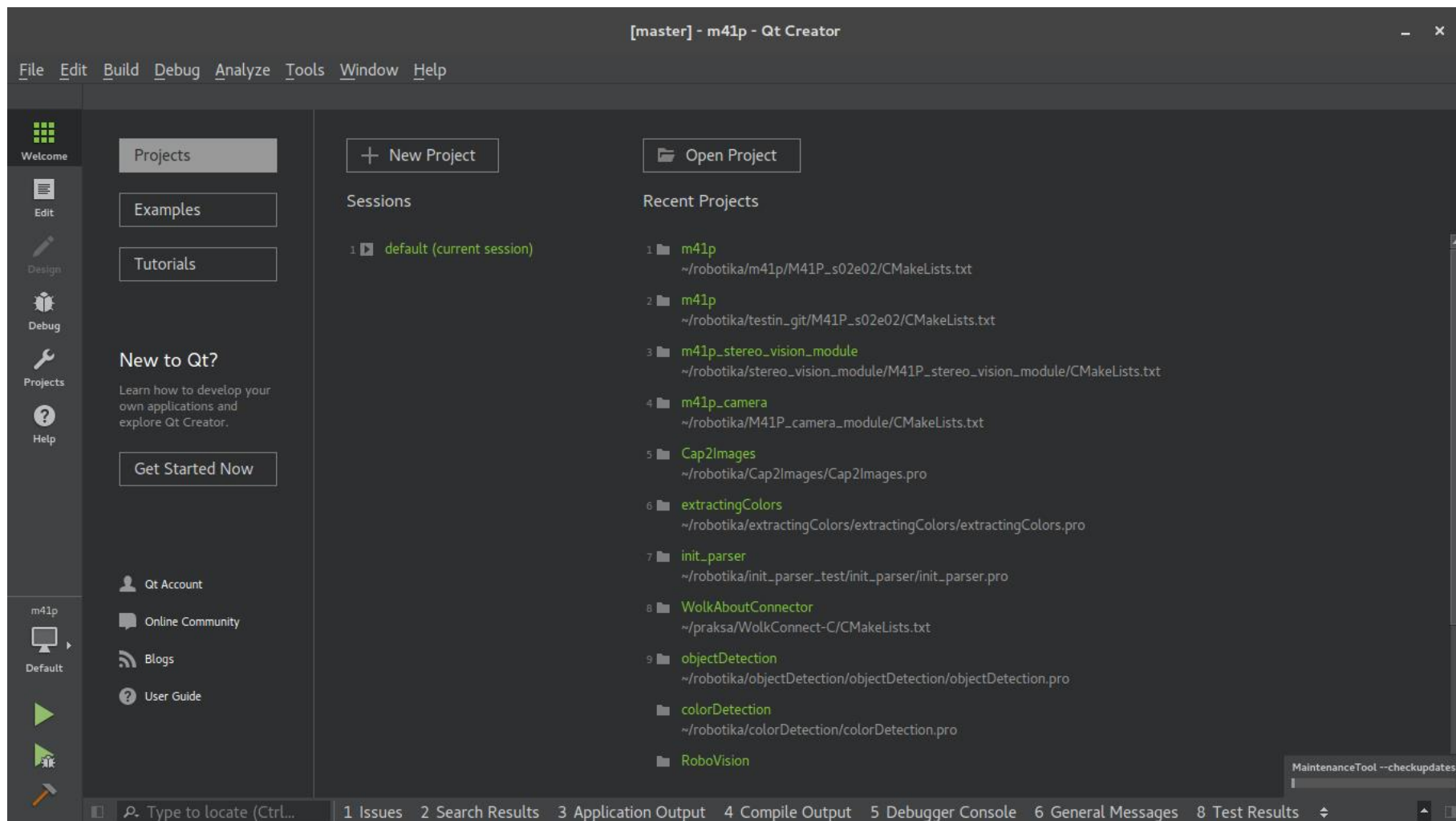
- IDE
- Cross-platform razvoj
- Qt framework
- Signali i slotovi



- Cross-platform IDE
- Podrška za C, C++, Java, Python, JavaScript i QML
- Podrška za sisteme za verzionisanje koda npr. Git, SVN
- Ugrađen Debugger
- Podrška za razne build sisteme npr. Qmake, CMake

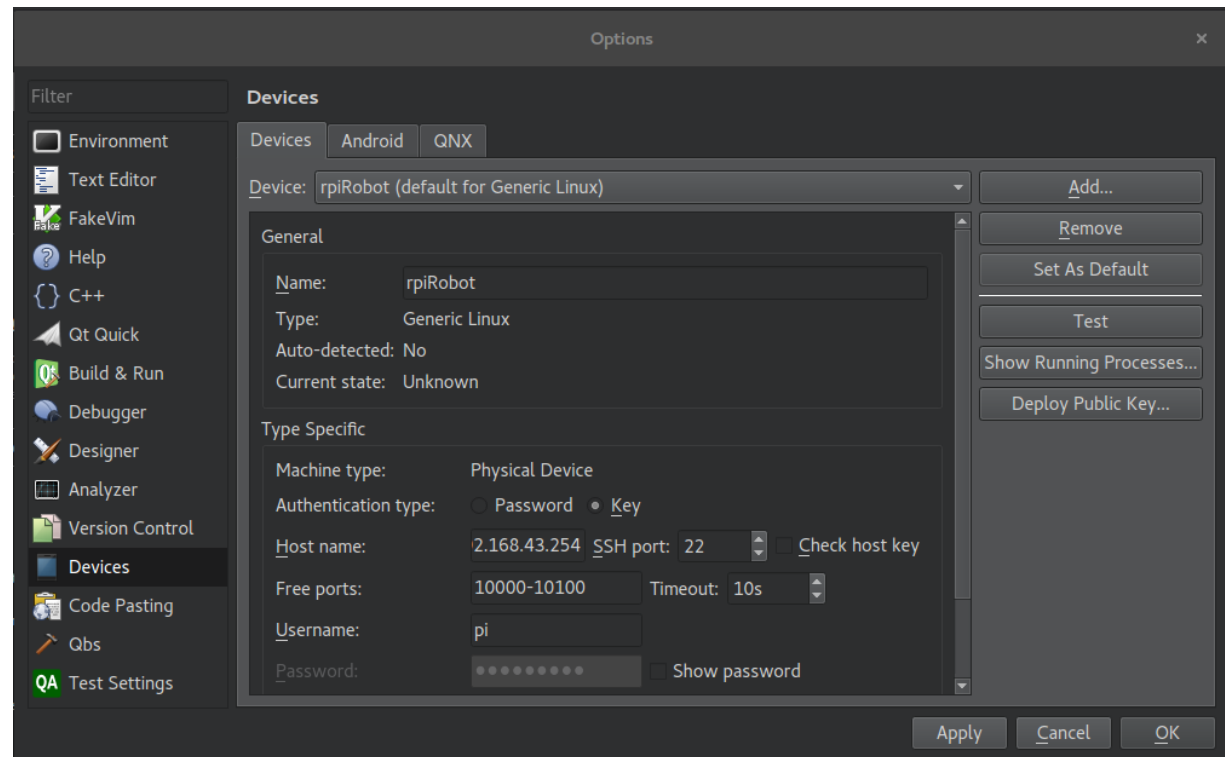
IDE

MRKiRM 1
2018





- Mogućnost pravljenja svojih uređaja (devices)
- Mogućnost dodavanja željenog prevodioca za određenu arhitekturu





- Mogućnost dodavanja cross-platform debuggera
- Mogućnost dostavljanja izvršne datoteke na odredišnu arhitekturu i pokretanje kontrolisanog izvršenja
- Primer razvoja za Raspberry Pi:
 - Uređaj: - Generic Linux Device
 - Prevodilac: gcc-arm-linux-gnueabi-hf-
 - Debugger: gdb-multiarch
 - Dostavljanje izvršne datoteke putem SSH



- Razvoj multi-platform aplikacija
- Razvoj GUI i konzolnih aplikacija
- Podrška za:
 - Rukovanje nitima
 - Mreže
 - Parsiranje XML, JSON
 - Rad sa SQL bazama
 - Rukovanje prozorima i widgetima
 - UI i UX

Qt framework

MRKiRM 1
2018



- Primeri aplikacija razvijenih sa Qt frameworkom:
 - Skype
 - Teamviewer
 - Google Earth
 - VLC Media Player
 - Tesla Model S in-car UI

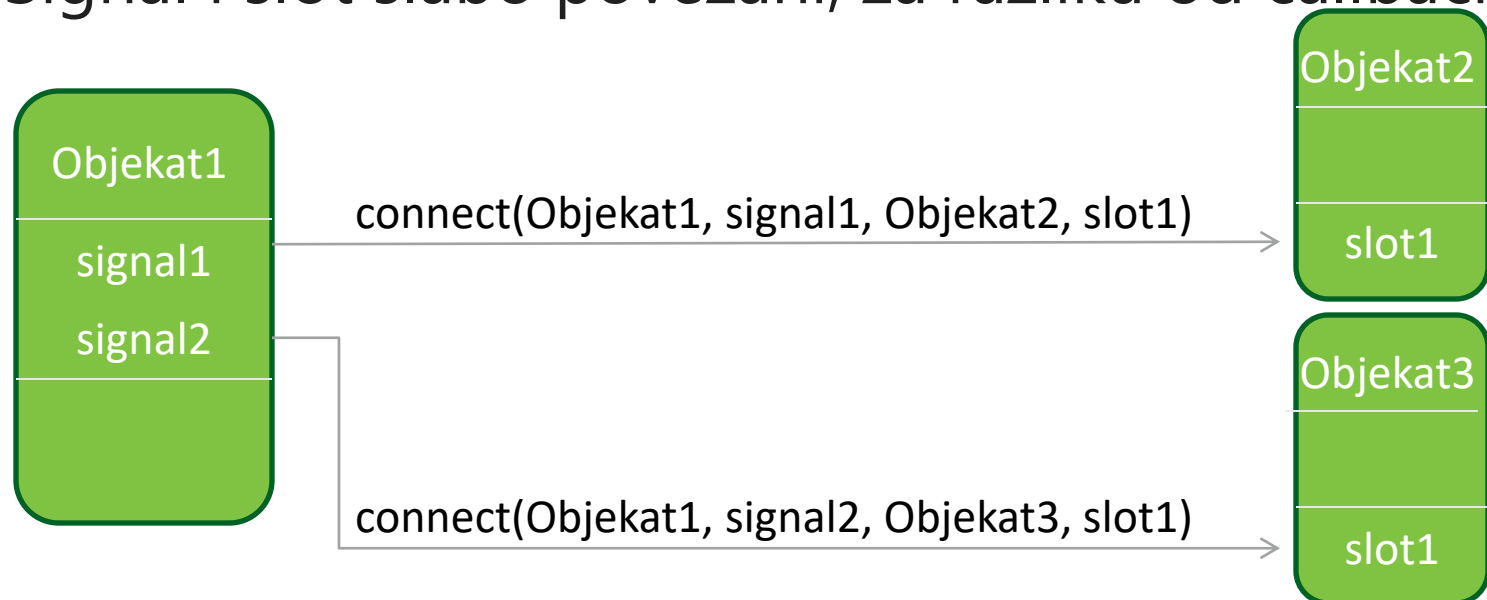


Signali i slotovi

MRKiRM 1
2018



- Najveća razlika između Qt i drugih frameworka
- Komunikacija između objekata
- Poput callback-a, ali mnogo bolje
 - Type-safety
 - Signal i slot slabo povezani, za razliku od callback-a



Signali i slotovi

MRKiRM 1
2018



```
< > dummy.h
2  #define DUMMY_H
3
4  #include <QObject>
5
6  class Dummy : public QObject
7  {
8      Q_OBJECT
9  public:
10     Dummy() { m_value = 0; }
11     int get_value() { return m_value; }
12
13     signals:
14         void value_changed(int value);
15     public slots:
16         void set_value(int value);
17
18     private:
19         int m_value;
20 };
21
22 #endif // DUMMY_H
23
```

```
dummy.cpp
1  #include "dummy.h"
2
3  void Dummy::set_value(int value) {
4      m_value = value;
5      emit value_changed(value);
6  }
7
```

Signali i slotovi

MRKiRM 1
2018



```
< > SignalsSlots/main.cpp # main(): int
1  #include <iostream>
2  #include "dummy.h"
3
4  using namespace std;
5
6  int main()
7  {
8      Dummy objekat1, objekat2;
9
10     QObject::connect(&objekat1, SIGNAL(value_changed(int)), &objekat2, SLOT(set_value(int)));
11
12     objekat2.set_value(20);
13     cout << "Vrednost polja m_value objekta 2 je: " << objekat2.get_value() << endl;
14     objekat1.set_value(10);
15     cout << "Vrednost polja m_value objekta 2 je: " << objekat2.get_value() << endl;
16
17     return 0;
18 }
19
```

Hvala na pažnji!