

18 定时器事件

18.1 基于信号——槽的定时器编程

用QTimer类创建定时器对象，调用该对象的start方法启动定时器。该方法的参数为以毫秒为单位的定时周期。此后每隔指定毫秒，定时器对象即发送一个timeout信号。通过将该信号连接到某个槽函数，实现定时任务。想要停止定时器，只需调用定时器对象的stop方法即可。

18.2 基于事件的定时器编程

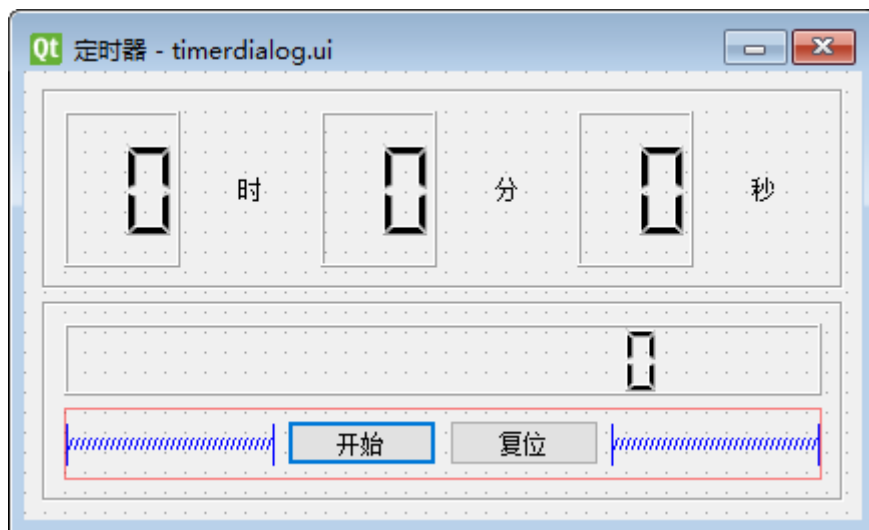
任何QObject类的子类对象都可以调用其继承自基类的startTimer方法启动定时器。该方法的参数为以毫秒为单位的定时周期。此后每隔指定毫秒，系统都会触发一次定时器事件。QObject类的timerEvent虚函数会被调用。通过覆盖该虚函数，实现定时任务。想要停止定时器，只需调用继承自QObject类的killTimer方法即可。

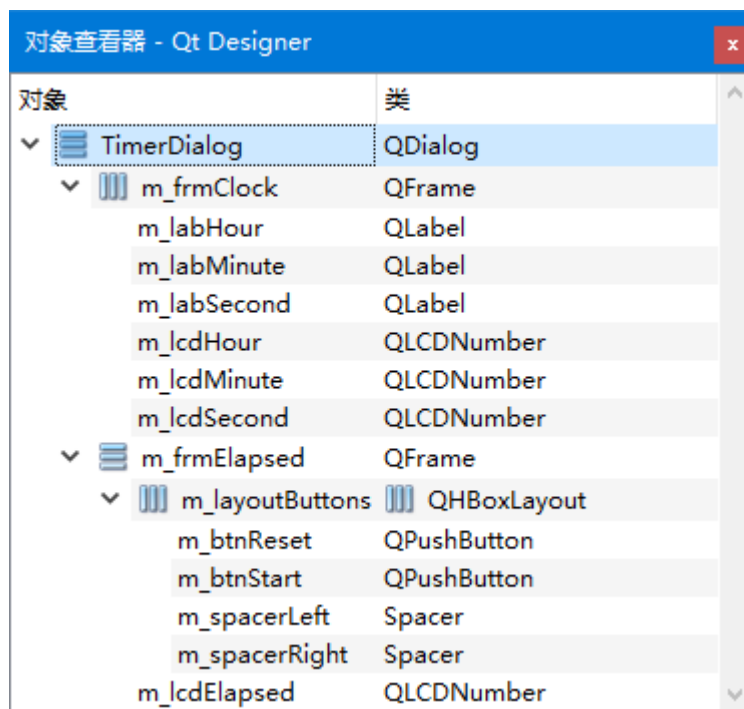
18.3 案例

18.3.1 创建项目

通过QtCreator，在C:\Users\Minwei\Projects\Qt路径下，创建名为Timer的项目。

18.3.2 设计界面





C:\Users\Minwei\Projects\Qt\Timer\timerdialog.ui:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3 <class>TimerDialog</class>
4 <widget class="QDialog" name="TimerDialog">
5 <property name="geometry">
6 <rect>
7 <x>0</x>
8 <y>0</y>
9 <width>418</width>
10 <height>224</height>
11 </rect>
12 </property>
13 <property name="windowTitle">
14 <string>定时器</string>
15 </property>
16 <layout class="QVBoxLayout" name="m_layoutDialog">
17 <item>
18 <widget class="QFrame" name="m_frmClock">
19 <property name="minimumSize">
20 <size>
21 <width>400</width>
22 <height>100</height>
23 </size>
24 </property>
25 <property name="frameShape">
26 <enum>QFrame::Box</enum>
27 </property>
28 <property name="frameShadow">
29 <enum>QFrame::Sunken</enum>
30 </property>
31 <layout class="QHBoxLayout" name="m_layoutClock">
32 <item>
33 <widget class="QLCDNumber" name="m_lcdHour">
34 <property name="digitCount">

```

```
35     <number>2</number>
36     </property>
37 </widget>
38 </item>
39 <item>
40     <widget class="QLabel" name="m_labHour">
41         <property name="text">
42             <string>时</string>
43         </property>
44         <property name="alignment">
45             <set>Qt::AlignCenter</set>
46         </property>
47     </widget>
48 </item>
49 <item>
50     <widget class="QLCDNumber" name="m_lcdMinute">
51         <property name="digitCount">
52             <number>2</number>
53         </property>
54     </widget>
55 </item>
56 <item>
57     <widget class="QLabel" name="m_labMinute">
58         <property name="text">
59             <string>分</string>
60         </property>
61         <property name="alignment">
62             <set>Qt::AlignCenter</set>
63         </property>
64     </widget>
65 </item>
66 <item>
67     <widget class="QLCDNumber" name="m_lcdSecond">
68         <property name="digitCount">
69             <number>2</number>
70         </property>
71     </widget>
72 </item>
73 <item>
74     <widget class="QLabel" name="m_labSecond">
75         <property name="text">
76             <string>秒</string>
77         </property>
78         <property name="alignment">
79             <set>Qt::AlignCenter</set>
80         </property>
81     </widget>
82 </item>
83 </layout>
84 </widget>
85 </item>
86 <item>
87     <widget class="QFrame" name="m_frmElapsed">
88         <property name="minimumSize">
89             <size>
90                 <width>0</width>
```

```
91     <height>100</height>
92     </size>
93 </property>
94 <property name="frameShape">
95     <enum>QFrame::Box</enum>
96 </property>
97 <property name="frameShadow">
98     <enum>QFrame::Sunken</enum>
99 </property>
100 <layout class="QVBoxLayout" name="m_layoutElapsed">
101     <item>
102         <widget class="QLCDNumber" name="m_lcdElapsed">
103             <property name="digitCount">
104                 <number>12</number>
105             </property>
106         </widget>
107     </item>
108     <item>
109         <layout class="QHBoxLayout" name="m_layoutButtons">
110             <item>
111                 <spacer name="m_spacerLeft">
112                     <property name="orientation">
113                         <enum>Qt::Horizontal</enum>
114                     </property>
115                     <property name="sizeHint" stdset="0">
116                         <size>
117                             <width>40</width>
118                             <height>20</height>
119                         </size>
120                     </property>
121                 </spacer>
122             </item>
123             <item>
124                 <widget class="QPushButton" name="m_btnStart">
125                     <property name="text">
126                         <string>开始</string>
127                     </property>
128                     <property name="default">
129                         <bool>true</bool>
130                     </property>
131                 </widget>
132             </item>
133             <item>
134                 <widget class="QPushButton" name="m_btnReset">
135                     <property name="text">
136                         <string>复位</string>
137                     </property>
138                 </widget>
139             </item>
140             <item>
141                 <spacer name="m_spacerRight">
142                     <property name="orientation">
143                         <enum>Qt::Horizontal</enum>
144                     </property>
145                     <property name="sizeHint" stdset="0">
146                         <size>
```

```

147         <width>40</width>
148         <height>20</height>
149     </size>
150 </property>
151 </spacer>
152 </item>
153 </layout>
154 </item>
155 </layout>
156 </widget>
157 </item>
158 </layout>
159 </widget>
160 <resources/>
161 <connections/>
162 </ui>

```

18.3.3 实现功能

C:\Users\Minwei\Projects\Qt\Timer\timerdialog.h:

```

1  #ifndef TIMERDIALOG_H
2  #define TIMERDIALOG_H
3
4  #include <QDialog>
5  #include <QTimer>
6  #include <QTime>
7
8  QT_BEGIN_NAMESPACE
9  namespace Ui { class TimerDialog; }
10 QT_END_NAMESPACE
11
12 class TimerDialog : public QDialog
13 {
14     Q_OBJECT
15
16 public:
17     TimerDialog(QWidget *parent = nullptr);
18     ~TimerDialog();
19
20 protected:
21     void timerEvent(QTimerEvent*);
22
23 private slots:
24     void on_m_clock_timeout();
25     void on_m_btnStart_clicked();
26     void on_m_btnReset_clicked();
27
28 private:
29     Ui::TimerDialog *ui;
30     QTimer m_clock;
31     QTime m_elapsed;
32     int m_elapsedTimer;
33 };
34

```

C:\Users\Minwei\Projects\Qt\Timer\timerdialog.cpp:

```
1  #include "timerdialog.h"
2  #include "ui_timerdialog.h"
3
4  TimerDialog::TimerDialog(QWidget *parent)
5      : QDialog(parent)
6      , ui(new Ui::TimerDialog)
7  {
8      ui->setupUi(this);
9
10     connect(&m_clock, SIGNAL(timeout()),
11            this, SLOT(on_m_clock_timeout()));
12
13     on_m_clock_timeout();
14     on_m_btnReset_clicked();
15
16     m_clock.start(1000);
17 }
18
19 TimerDialog::~TimerDialog()
20 {
21     m_clock.stop();
22
23     delete ui;
24 }
25
26 void TimerDialog::timerEvent(QTimerEvent*)
27 {
28     m_elapsed = m_elapsed.addMsecs(1);
29     ui->m_lcdElapsed->display(m_elapsed.toString("HH:mm:ss.zzz"));
30 }
31
32 void TimerDialog::on_m_clock_timeout()
33 {
34     QTime now = QTime::currentTime();
35
36     ui->m_lcdHour->display(QString::number(now.hour()));
37     ui->m_lcdMinute->display(QString::number(now.minute()));
38     ui->m_lcdSecond->display(QString::number(now.second()));
39 }
40
41 void TimerDialog::on_m_btnStart_clicked()
42 {
43     if (ui->m_btnStart->text() == "开始")
44     {
45         m_elapsedTimer = startTimer(1);
46
47         ui->m_btnStart->setText("停止");
48         ui->m_btnReset->setEnabled(false);
49     }
50     else
51     {
```

```
52     killTimer(m_elapsedTimer);
53
54     ui->m_btnStart->setText("开始");
55     ui->m_btnReset->setEnabled(true);
56 }
57 }
58
59 void TimerDialog::on_m_btnReset_clicked()
60 {
61     m_elapsed.setHMS(0, 0, 0);
62     ui->m_lcdElapsed->display(m_elapsed.toString("HH:mm:ss.zzz"));
63 }
```

18.3.4 测试验证

运行效果如图所示：

