

# QT Eğitimi - 4 : İkili Saat (Binary Clock)

## Önder Arslan

<[onder@xcoders.net](mailto:onder@xcoders.net)>

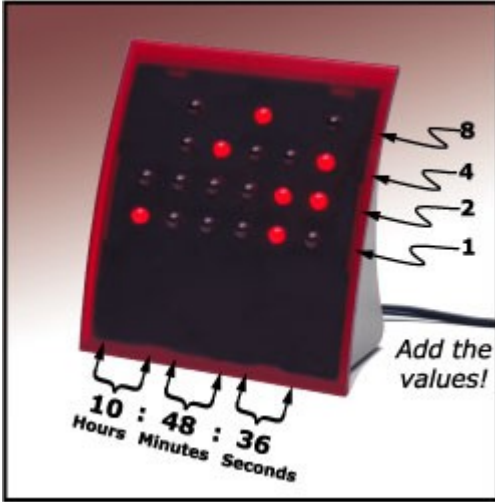
Sürüm 1.0

### Özet

Bu belge qt öğrenimi için bir dizi şeklinde hazırlanmıştır. C++ bilmek ön koşuldur.

### Giriş

Kullanıcı arabirimi oluşturmayı öğrenirken insan arada şöyle çalışan bir şeyler de yapmak istiyor. QPushButton 'a bas mesaj kutusu çıksın , QPushButton 'a bas metin kutuları temizlensin sıkıcı oluyor. Bu nedenle dördüncü eğitimde "binary clock" uygulaması yazacağız. "Binary clock" ne dersiniz budur :



[thinkgeek](http://thinkgeek.com) sitesinde satılır, kol saati ve masa saati biçimleri de vardır :) Kol saati olanını hep almak istemişimdir ama nasip olmamıştır :( "Eee alamıyorsam kendim yapayım" dedim ve böylece qt uygulamamızın konusu oldu. Ama bu uygulamayı yaparken yeni sınıflar öğrenmeyi de ihmal etmeyeceğiz. Öğreneceğimiz sınıflar : QLabel, QTimer, QPixmap, QTime ...

### Saat

```
/*--- form.h ---*/

#ifndef FORM_H
#define FORM_H

#include <QtGui>

class Form : public QDialog
{
    Q_OBJECT
public:
    Form();
};
```

```

    QLabel *lbl_hour_first_1;
    QLabel *lbl_hour_first_2 ;

    QLabel *lbl_hour_second_1;
    QLabel *lbl_hour_second_2;
    QLabel *lbl_hour_second_4;
    QLabel *lbl_hour_second_8;

    QLabel *lbl_second_first_1;
    QLabel *lbl_second_first_2;
    QLabel *lbl_second_first_4;

    QLabel *lbl_second_second_1;
    QLabel *lbl_second_second_2;
    QLabel *lbl_second_second_4;
    QLabel *lbl_second_second_8;

public slots:

    void Update();

};

#endif

```

### form.h:

Başlık dosyamızda sizinde farketmediğiniz gibi QLabel dışında öğrenmediğimiz bir şey yok ;) QLabel metin ve imaj görüntülemek için tasarlanmış bir sınıf. Aslında qt dışında (c# ta yada vb 'de label kullanımında) bir label 'ın imaj gösterebilmesinin böyle üstüne basıldığını (The QLabel widget provides a text or images display.) görmemişim. Bugüne kadar kullanımlarım hep metin gösterimi şeklindeydi. Bir Label benim için kullanıcı arabiriminde ya metin kutusunun kutusunun içine ne yazılması gerektiğini ya da tıklanacak düğmenin ne iş yapacağını anlatırdı felan. Bu sebeple aslında ilk aklıma gelen programda saati ve dakikayı gösteren ışıkları qt 'ye çizdirmektir. Ama bu uygulamada QLabel kullanacağız ;) :

```

lbl_hour_first      saatin ilk hanesini (ör: 13:29 da 'da 1 'i gösterir)
lbl_hour_second     saatin ikinci hanesini (ör: 13:29 'da 3 'ü gösterir)
lbl_second_first    dakikanın birinci hanesini gösterir (ör: 13:29 'da 2 'yi gösterir)
lbl_second_second   dakikanın ikinci hanesini gösterir (ör:13:29 'da 9 'u gösterir)

```

```

/*--- form.cpp ---*/

#include "form.h"
#include <map>
#include <algorithm>
#include <functional>

using namespace std;

typedef pair<int,int> light_pair;

static int first, second;

```

```

static map<int,light_pair> mymap;

pair<int,int> getTime()
{
    QTime time = QTime::currentTime();
    return make_pair(time.hour(),time.minute());
}

Form::Form()
{

    mymap.insert(make_pair(1,light_pair(1,0)));
    mymap.insert(make_pair(2,light_pair(2,0)));
    mymap.insert(make_pair(3,light_pair(1,2)));
    mymap.insert(make_pair(4,light_pair(4,0)));
    mymap.insert(make_pair(5,light_pair(1,4)));
    mymap.insert(make_pair(6,light_pair(2,4)));
    mymap.insert(make_pair(7,light_pair(7,7)));
    mymap.insert(make_pair(8,light_pair(8,0)));
    mymap.insert(make_pair(9,light_pair(1,8)));

    lbl_hour_first_1 = new QLabel(this);
    lbl_hour_first_2 = new QLabel(this);

    lbl_hour_second_1 = new QLabel(this);
    lbl_hour_second_2 = new QLabel(this);
    lbl_hour_second_4 = new QLabel(this);
    lbl_hour_second_8 = new QLabel(this);

    lbl_second_first_1 = new QLabel(this);
    lbl_second_first_2 = new QLabel(this);
    lbl_second_first_4 = new QLabel(this);

    lbl_second_second_1 = new QLabel(this);
    lbl_second_second_2 = new QLabel(this);
    lbl_second_second_4 = new QLabel(this);
    lbl_second_second_8 = new QLabel(this);

    lbl_hour_first_1->move(5, 140);
    lbl_hour_first_2->move(5, 100);

    lbl_hour_second_1->move(50, 140);
    lbl_hour_second_2->move(50, 100);
    lbl_hour_second_4->move(50, 60);
    lbl_hour_second_8->move(50, 20);

    lbl_second_first_1->move(100, 140);
    lbl_second_first_2->move(100, 100);
    lbl_second_first_4->move(100, 60);

    lbl_second_second_1->move(145, 140);
    lbl_second_second_2->move(145, 100);
    lbl_second_second_4->move(145, 60);
    lbl_second_second_8->move(145, 20);
}

```

```

lbl_hour_first_1->setPixmap(QPixmap("bir.gif"));
lbl_hour_first_2->setPixmap(QPixmap("bir.gif"));

lbl_hour_second_1->setPixmap(QPixmap("bir.gif"));
lbl_hour_second_2->setPixmap(QPixmap("bir.gif"));
lbl_hour_second_4->setPixmap(QPixmap("bir.gif"));
lbl_hour_second_8->setPixmap(QPixmap("bir.gif"));

lbl_second_first_1->setPixmap(QPixmap("bir.gif"));
lbl_second_first_2->setPixmap(QPixmap("bir.gif"));
lbl_second_first_4->setPixmap(QPixmap("bir.gif"));

lbl_second_second_1->setPixmap(QPixmap("bir.gif"));
lbl_second_second_2->setPixmap(QPixmap("bir.gif"));
lbl_second_second_4->setPixmap(QPixmap("bir.gif"));
lbl_second_second_8->setPixmap(QPixmap("bir.gif"));

resize(200,200);

QTimer *timer = new QTimer(this);
connect(timer, SIGNAL(timeout()), this, SLOT(Update()));
timer->start(1000);
}

```

```

void Form::Update()
{
    first = mymap[ (getTime().first - getTime().first %10) /10 ]
.first;
    second = mymap[(getTime().first - getTime().first %10) /
10].second;

    lbl_hour_first_1->setPixmap(QPixmap("bir.gif"));
    lbl_hour_first_2->setPixmap(QPixmap("bir.gif"));

    if (first == 1 || second == 1)
        lbl_hour_first_1->setPixmap(QPixmap("iki.gif"));
    if (first == 2 || second == 2)
        lbl_hour_first_2->setPixmap(QPixmap("iki.gif"));

    first = mymap[getTime().first %10].first;
    second = mymap[getTime().first%10].second;

    lbl_hour_second_1->setPixmap(QPixmap("bir.gif"));
    lbl_hour_second_2->setPixmap(QPixmap("bir.gif"));
    lbl_hour_second_4->setPixmap(QPixmap("bir.gif"));
    lbl_hour_second_8->setPixmap(QPixmap("bir.gif"));
}

```

```

if (first == 1 || second == 1)
    lbl_hour_second_1->setPixmap(QPixmap("iki.gif"));
if (first == 2 || second == 2)
    lbl_hour_second_2->setPixmap(QPixmap("iki.gif"));
if (first == 4 || second == 4)
    lbl_hour_second_4->setPixmap(QPixmap("iki.gif"));
if (first == 8 || second == 8)
    lbl_hour_second_8->setPixmap(QPixmap("iki.gif"));
if (first == 7 || second == 7)
{
    lbl_hour_second_1->setPixmap(QPixmap("iki.gif"));
    lbl_hour_second_2->setPixmap(QPixmap("iki.gif"));
    lbl_hour_second_4->setPixmap(QPixmap("iki.gif"));
}

first = mymap[ (getTime().second - getTime().second %10) /10 ]
.first;
second = mymap[(getTime().second - getTime().second %10)
/10].second;

lbl_second_first_1->setPixmap(QPixmap("bir.gif"));
lbl_second_first_2->setPixmap(QPixmap("bir.gif"));
lbl_second_first_4->setPixmap(QPixmap("bir.gif"));

if (first == 1 || second == 1)
    lbl_second_first_1->setPixmap(QPixmap("iki.gif"));
if (first == 2 || second == 2)
    lbl_second_first_2->setPixmap(QPixmap("iki.gif"));
if (first == 4 || second == 4)
    lbl_second_first_4->setPixmap(QPixmap("iki.gif"));

first = mymap[getTime().second %10].first;
second = mymap[getTime().second %10].second;

lbl_second_second_1->setPixmap(QPixmap("bir.gif"));
lbl_second_second_2->setPixmap(QPixmap("bir.gif"));
lbl_second_second_4->setPixmap(QPixmap("bir.gif"));
lbl_second_second_8->setPixmap(QPixmap("bir.gif"));

if (first == 1 || second == 1)
    lbl_second_second_1->setPixmap(QPixmap("iki.gif"));
if (first == 2 || second == 2)
    lbl_second_second_2->setPixmap(QPixmap("iki.gif"));
    if (first == 4 || second == 4)
        lbl_second_second_4->setPixmap(QPixmap("iki.gif"));
if (first == 8 || second == 8)
    lbl_second_second_8->setPixmap(QPixmap("iki.gif"));
if (first == 7 || second == 7)
{
    lbl_second_second_1->setPixmap(QPixmap("iki.gif"));
    lbl_second_second_2->setPixmap(QPixmap("iki.gif"));
    lbl_second_second_4->setPixmap(QPixmap("iki.gif"));
}
}

```

## form.cpp

Form.cpp kaynak dosyasının uzunluğu biraz büyük görüncede gözünüzü korkutmasın. Çünkü kullanılan kontrollerin çokluğu nedeniyle biraz fazla, sadece ışıkları göstermek için 13 adet QLabel kullandığımızı ve bunların 2 özelliğini kullandığımızda 26 satır ettiğini düşünürseniz korkacak birşey kalmıyor :) Bir saat yapabilmek için öncelikle saatin kaç olduğunu bilmemiz lazım :) Bu nedenle getTime() adında bir fonksiyon yazıyoruz. getTime() içinde **currentTime()** değeriyle set edilmiş bir **QTime** nesnesi oluşturuyoruz. bu nesnenin **hour()** fonksiyonuyla saati **minute()** fonksiyonu ile dakikayı alıp geri dönüş değeri olarak döndürüyoruz.

Static **QLabel** nesnelimizi oluşturduktan sonra ( ki kodun 13 satırı burası ;) ) **move** komutu ile onları istediğimiz yere taşıyoruz (kodun 13 satırı da bu etti 26 satır).

Qt resim gösterimi için dört sınıf sunar : **QPixmap**, **QImage**, **QBitmap** ve **QPicture** . **QPixmap** resim gösterimi için kullanacağımız sınıftır. Bu sınıf ile QLabel 'da **setPixmap()** fonksiyonunu kullanarak resim gösterebiliriz. Hatta **QAbstractButton** sınıfının alt sınıfları olan **QPushButton**, **QToolButton** gibi kontroller üzerinde de resim gösterebiliriz. Bizde QLabel 'larımıza setPixmap() fonksiyonu ile QPixmap ' nesnelimizi atıyoruz (yani bir.gif resmini).

**QTimer** bize tekrarlanan yada tek-atışlık zamanlayıcı olanağı sunar. Biz programızda zamanlayıcıyı belirli zamanlarda (bu programda 1000ms 'de) çalıştırarak saatte olan değişiklikleri görebilmek için Update() komutunu çağırıyoruz. Örneğin **connect(timer, SIGNAL(timeout()), this, SLOT(Update()))**; satırında zamanlayıcı her **timeout()** fonksiyonunu çalıştırdığında Update() slotumuz çalışıyor. QTimer nesnesinin timeout değeri milisaniye olarak interval değişkeninde saklanır. Bu değeri ayarlamak için **setInterval()** fonksiyonu kullanılır. Biz kodumuzda **start()** fonksiyonunun parametre değişkenini kullanarak bu değeri(interval) ayarladık.

Update() fonksiyonu içinde setPixmap() fonksiyonu ile saate göre uygun imajı QLabel 'lara atamaktan başka bir işlem yapılmıyor.

```
/*--- main.cpp ---*/
#include <QApplication>
#include "form.h"
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    Form form;
    form.show();
    return app.exec();
}
```

## main.cpp

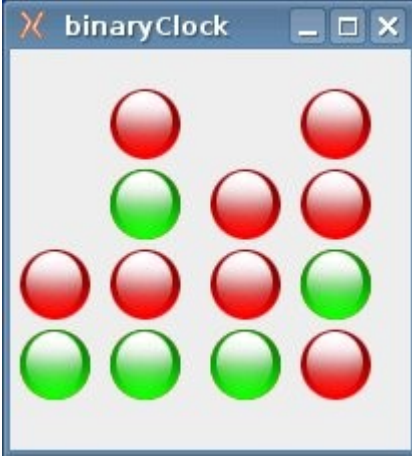
main.cpp dosyası yine aynı ;)

**Derleme**

**ve**

**Çalıştırma**

Program derlenip çalıştırıldığında aşağıdaki gibi görünür :



Saatin 15:12 olduğunu söylememe gerek yok herhalde ;)

### Son Söz

Programın algoritmasını tabiki açıklamadım, konumuz algoritmalar değil ;). Ben algoritmayı STL kullanarak oluşturdum. Siz çok daha farklı algoritmalar kullanıp kodu benden çok daha kısa yazabilirsiniz. ;)

### Telif Hakkı ve Lisans

Bu belgenin, *QT Eđitimi - 4 : İkili Saat (Binary Clock)*, 1.0 sürümünün **telif hakkı © 2006 Önder ARSLAN**'a aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.1 ya da daha sonraki sürümünün koşullarına bağlı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın bir kopyasını <http://www.gnu.org/copyleft/fdl.html> adresinde bulabilirsiniz.

Linux, Linus Torvalds adına kayıtlı bir ticarî isimdir.

Qt, TrollTech adına kayıtlı bir ticari isimdir.

#### Feragatname

Bu belgedeki bilgilerin kullanımından doğacak sorumluluklar, ve olası zararlardan belge yazarı sorumlu tutulamaz. Bu belgedeki bilgileri uygulama sorumluluğu uygulayana aittir.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim bir ticarî isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.