Archived:How to get accelerometer data of N900 using Qt

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt applications should now access accelerometer data using the cross platform Qt Mobility Sensor API

Archived: This article demonstrates how to develop a Qt Application that will to get the information from the accelerometer sensor at N900 device using native methods.

02 May 2010

An accelerometer is a device that measures proper acceleration, the acceleration experienced relative to free-fall.Single- and multi-axis models are available to detect magnitude and direction of the acceleration as a vector quantity, and can be used to sense orientation, vibration and shock. Micro-machined accelerometers are increasingly present in portable electronic devices like mobile phones and video game controllers, to detect the orientation of the device or provide for game input.

Accelerometer Data at N900

The accelerometer data is **sysfs** file information. Sysfs is a virtual file system provided by Linux 2.6. Sysfs exports information about devices and drivers from the kernel device model to userspace, and is also used for configuration.

The three coordinates of accelerometer sensor is in a text file provided on one line and separated by white space. This file is at path below:

/sys/class/i2c-adapter/i2c-3/3-001d/coord

The values are in mG (milli G). 1000 = 1 G. The **g-force** experienced by an object is its acceleration relative to free-fall like described above.

At N900 device, the values are connected to the position showed below.

The Code

- At starts, i created a QThread. This thread will read the file with data at all the time.

```
class aThread : public QThread
{
  Q_OBJECT

public:
    aThread(QObject *parent = 0);
```


(C) Copyright Nokia 2013. All rights reserved.

```
    void run();

    QString x;
    QString y;
    QString z;

public slots:
    void updateCoords();
    void processCoords(QString &data);

signals:
    void deviceOrientation(QString x, QString y, QString z);
};
```

- At implementation of our thread, the **updateCoords()** method is called with 20ms interval.

```
void aThread::run()
{
    QTimer timer;
    connect(&timer, SIGNAL(timeout()), this, SLOT(updateCoords()));
    timer.start(20); // 50 Hz update rate
    exec();
}
```

- This method open the file, read and process the string found on the file.

```
void aThread::updateCoords()
{
     QFile file("/sys/class/i2c-adapter/i2c-3/3-001d/coord");
     if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
         return;
     QTextStream in(&file);
     QString data = in.readAll();
     processCoords(data);
}
```

- Then, in the processing of information, the string found is split by the white spaces creating three different strings.These strings are the X,Y,Z coordinates of accelerometer.The signal **deviceOrientation()** is emitted to allowing that other class can connect yours slots to this signal.

```
void aThread::processCoords(QString &data)
{
    QStringList data_splited = data.split(" ");

    x = data_splited[0];
    y = data_splited[1];
    z = data_splited[2];

    emit deviceOrientation(x, y, z);
}
```

- At constructor of our superior class, i created the thread described above . Then i connected the signal **deviceOrientation()** of our thread with the slot **showData** of superior class.

```
Accelerometer::Accelerometer(QWidget *parent)
    : QWidget(parent)
{
    ...
    myThread = new aThread(this);
    connect(myThread, SIGNAL(deviceOrientation(QString, QString, QString)),
            this, SLOT(showData(QString, QString, QString)));
    myThread->start(QThread::NormalPriority);
    ...
```

- Then the **showData** method shows the three axis information of accelerometer sensor at labels at the screen.

```
void Accelerometer::showData(QString x, QString y, QString z)
{
    xlabel->setText(x);
    ylabel->setText(y);
    zlabel->setText(z);
}
```

Screenshots

(C) Copyright Nokia 2013. All rights reserved.

Source Code

File:QtAccelerometer.zip

Note: The N900 allow that you use shortcuts with the combination of keys.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.