True Engineering

# Датчики

# QT Sensors

- **QML**
  - import QtSensors 5.2
  - Requires в spec: qt5-qtdeclarative-import-sensors
- **C++**
  - pro-файл: QT += sensors
  - Requires в spec: qt5-qtsensors
  - #include <QtSensor>

# Компоненты и классы датчиков в Qt

| Компонент QML | Класс C++ | Описание |
|---|---|---|
| ProximitySensor | QProximitySensor | Датчик приближения |
| LightSensor | QLightSensor | Датчик света |
| AmbientLightSensor | QAmbientLightSensor | Датчик освещённости |
| Accelerometer | QAccelerometer | Акселерометр |
| OrientationSensor | QOrientationSensor | Датчик ориентации |
| RotationSensor | QRotationSensor | Датчик поворота |
| Gyroscope | QGyroscope | Гироскоп |
| Magnetometer | QMagnetometer | Магнитометр |
| Compass | QCompass | Компас |

# Список доступных датчиков

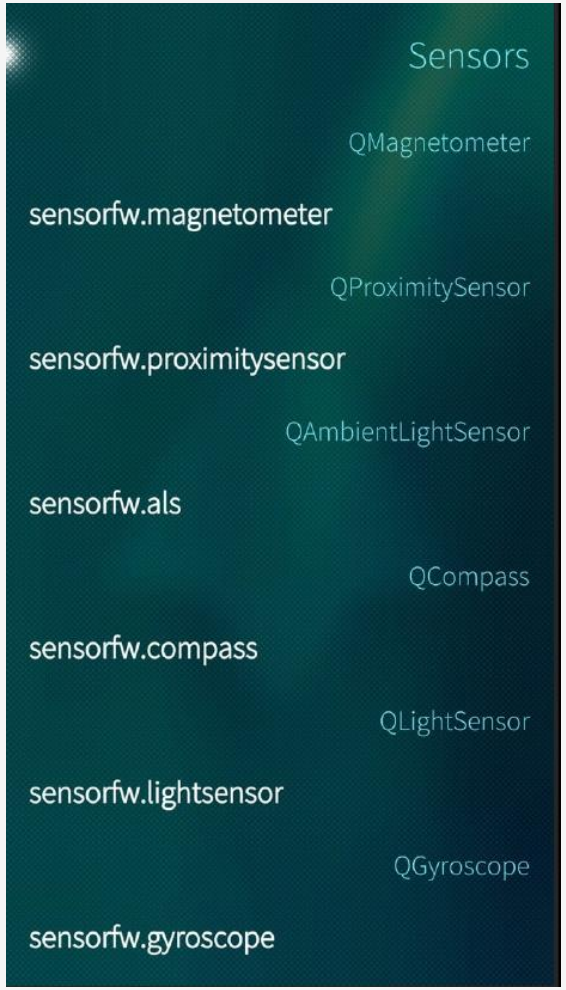
- **Список доступных типов датчиков**
  - QmlSensors.sensorTypes() : list<string>
  - QSensor::sensorTypes() : Qlist<QbyteArray>

- **Список датчиков указанного типа**
  - QmlSensors.sensorsForType(type) : list<string>
  - QSensor::sensorsForType(const QByteArray &type) : Qlist<QbyteArray>

- **Стандартный датчик указанного типа**
  - QmlSensors.defaultSensorForType(type) : string
  - QSensor::defaultSensorForType(const QByteArray &type) : QByteArray

# Список доступных датчиков

```
Component.onCompleted: {
    QmlSensors.sensorTypes().forEach(function (sensorType) {
        console.log(sensorType);
        var defaultSensor =
            QmlSensors.defaultSensorForType(sensorType);
        QmlSensors.sensorsForType(sensorType)
                          .forEach(function (sensor) {
            model.append({
                sensorIdentifier: sensor,
                type: sensorType,
                isDefault: sensor === defaultSensor
            })
        });
    }); }
```



# Sensor и QSensor

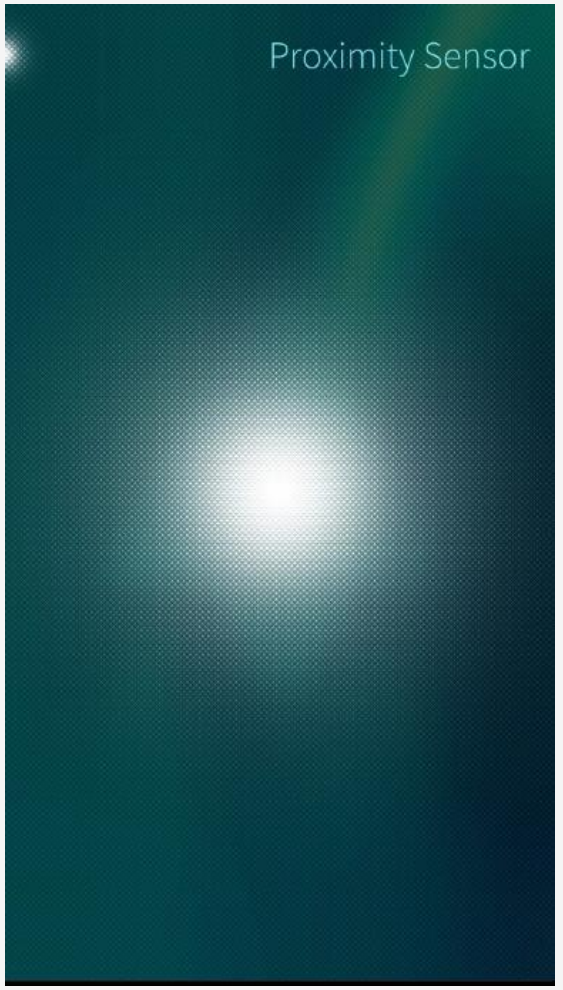
- description, type : string — описание и тип
- active : bool — включен ли датчик
- start() — начать получать данные
- stop() — прекратить получать данные
- busy : bool — доступен ли для использования

- alwaysOn : bool — работает ли при выключенном экране
- dataRate : int — частота обновления в герцах
- skipDuplicates : bool — пропускать ли повторы
- reading : SensorReading — актуальные значения датчика
  - timestamp : quint4 — отсчёт времени в микросекундах

# ProximitySensor

- reading : ProximityReading
  - near : bool — есть ли объект поблизости

```
ProximitySensor {
    id: proximitySensor
    active: true
}
Switch {
    anchors.centerIn: parent
    enabled: proximitySensor.active
    checked: proximitySensor.active &&
                proximitySensor.reading.near
    automaticCheck: false; scale: 10
}
```



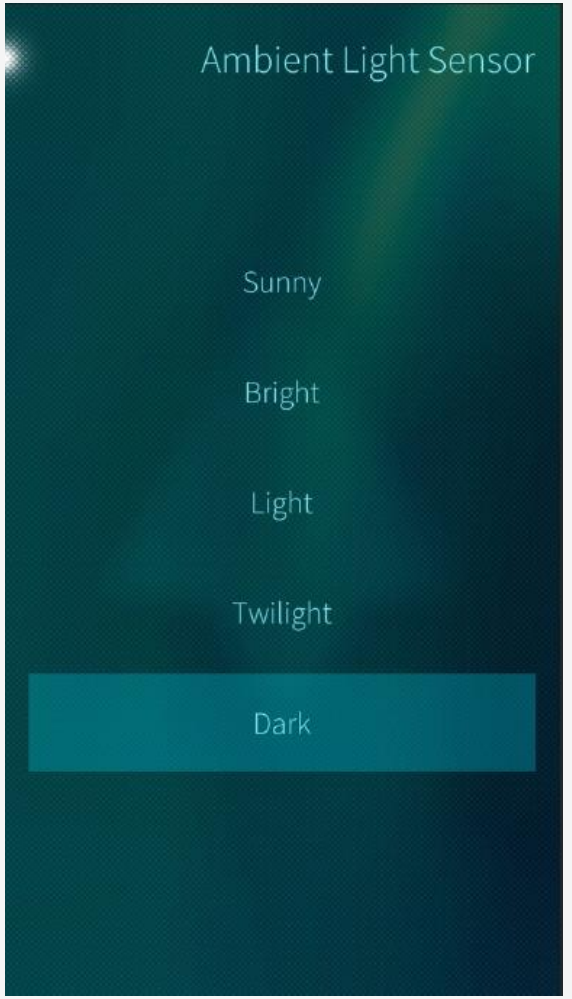
# LightSensor – датчик освещённости

- fieldOfView : real — угол зрения
- reading : LightReading
  - illuminance : real — освещённость в люксах

```
LightSensor {id: lightSensor; active: true }
Label {
    anchors.centerIn: parent
    text: lightSensor.reading ?
      qsTr("%1 lux").arg(lightSensor.reading.illuminance) :
      qsTr("Unknown")
    font.pixelSize: Theme.fontSizeExtraLarge
    color: Theme.highlightColor
}
```

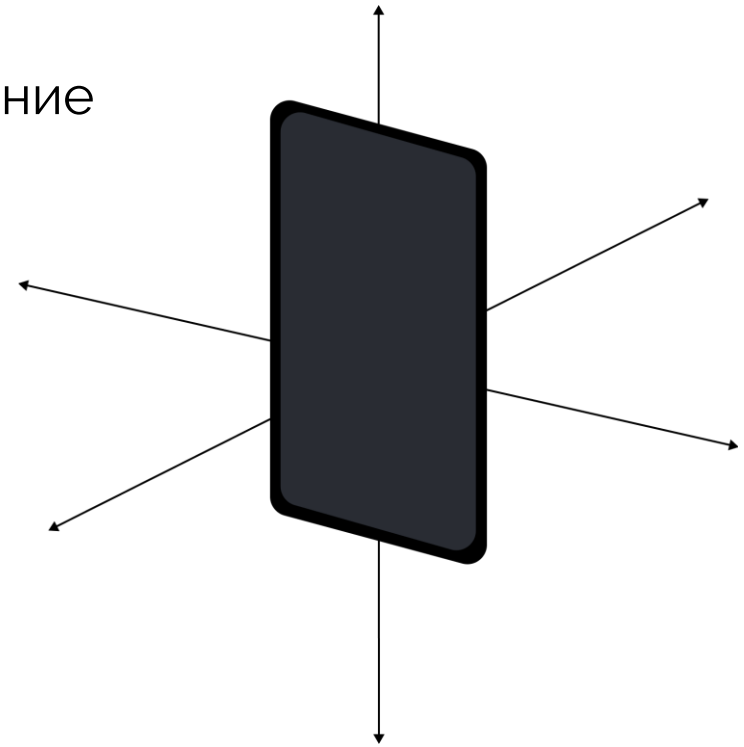
# AmbientLightSensor

- reading : AmbientLightReading
  - lightLevel : LightLevel — уровень освещённости
    - AmbientLightReading.Undefined — не определено
    - AmbientLightReading.Dark — темно
    - AmbientLightReading.Twilight — сумерки
    - AmbientLightReading.Light — светло
    - AmbientLightReading.Bright — ярко
    - AmbientLightReading.Sunny — очень ярко



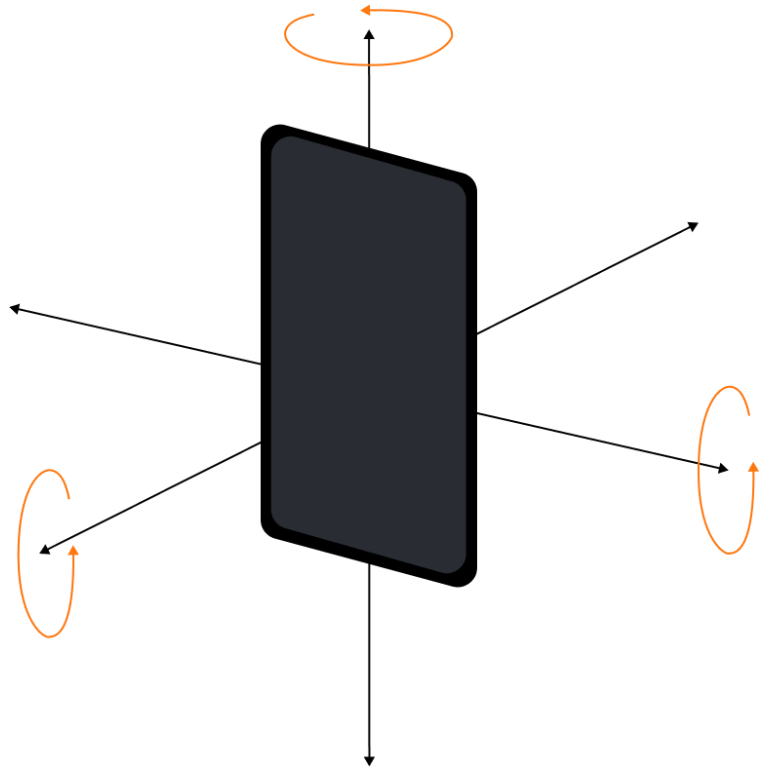
# Акселерометр

- **accelerationMode : AccelerationMode** — режим замеров (доступно не на всех устройствах)
  - Accelerometer.Combined — учитываются все силы
  - Accelerometer.Gravity — учитывается только направление ускорения свободного падения
  - Accelerometer.User — учитываются только перемещения устройства
- **reading : AccelerometerReading**
  - x : real — ускорение вдоль оси x в м/с²
  - y : real — ускорение вдоль оси y в м/с²
  - z : real — ускорение вдоль оси z в м/с²



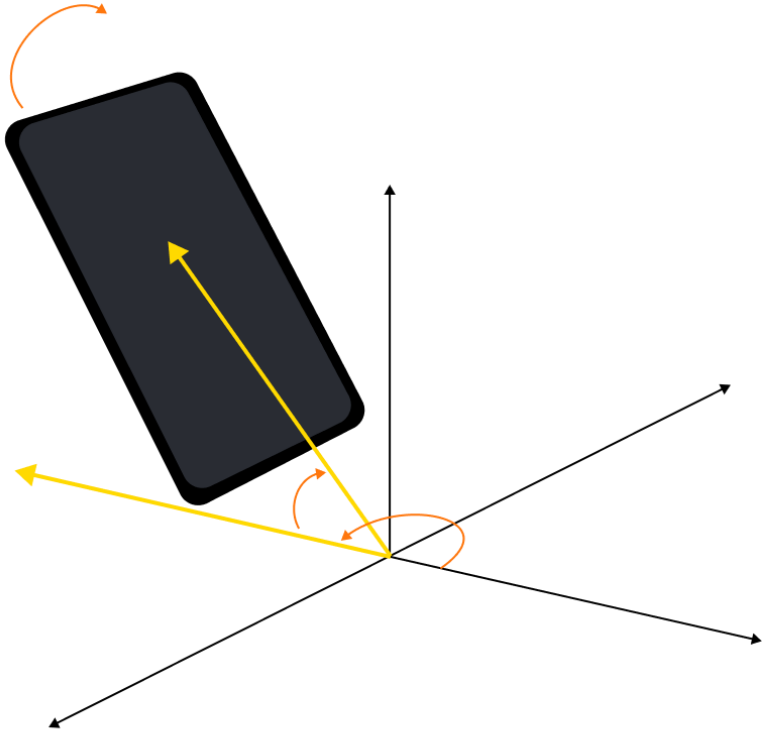
# Гироскоп

- reading : GyroscopeReading
  - x : real — отклонение по x
  - y : real — отклонение по y
  - z : real — отклонение по z



# RotationSensor

- hasZ : bool — доступен ли угол поворота вокруг оси z
- reading : RotationReading
  - x : real — угол поворота в градусах вокруг оси x
  - y : real — угол поворота в градусах вокруг оси y
  - z : real — угол поворота в градусах вокруг оси z



# Магнитометр

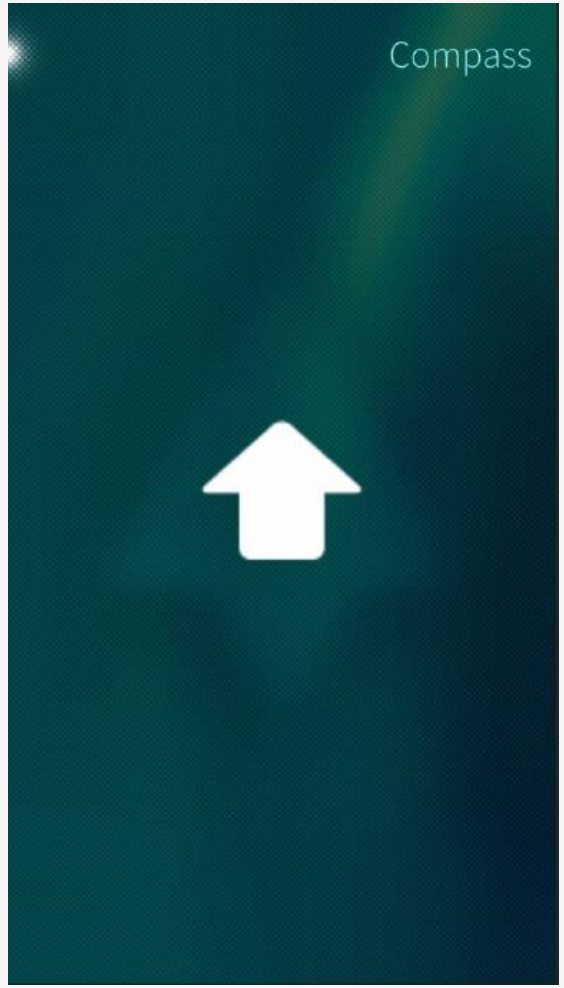
- reading : MagnetometerReading:
  - calibrationLevel : real — точность показаний датчика (от 0 до 1)
    - x : real — отклонение по x
    - y : real — отклонение по y
    - z : real — отклонение по z

# Компас

- reading : CompassReading
  - azimuth : real — угол между северным магнитным полюсом Земли и направлением верхней грани устройства
  - calibrationLevel : real — точность показаний датчика (от 0 до 1)

```
Compass { id: compass }

Image {
    anchors.centerIn: parent
    source: "image://theme/icon-m-capslock"
    width: Theme.pixelRatio*200
    height: Theme.pixelRatio*200
    rotation: {
        if (!compass.reading) return 0;
        return compass.reading ? - compass.reading.azimuth : 0
    }
}
```



# QSensorFilter

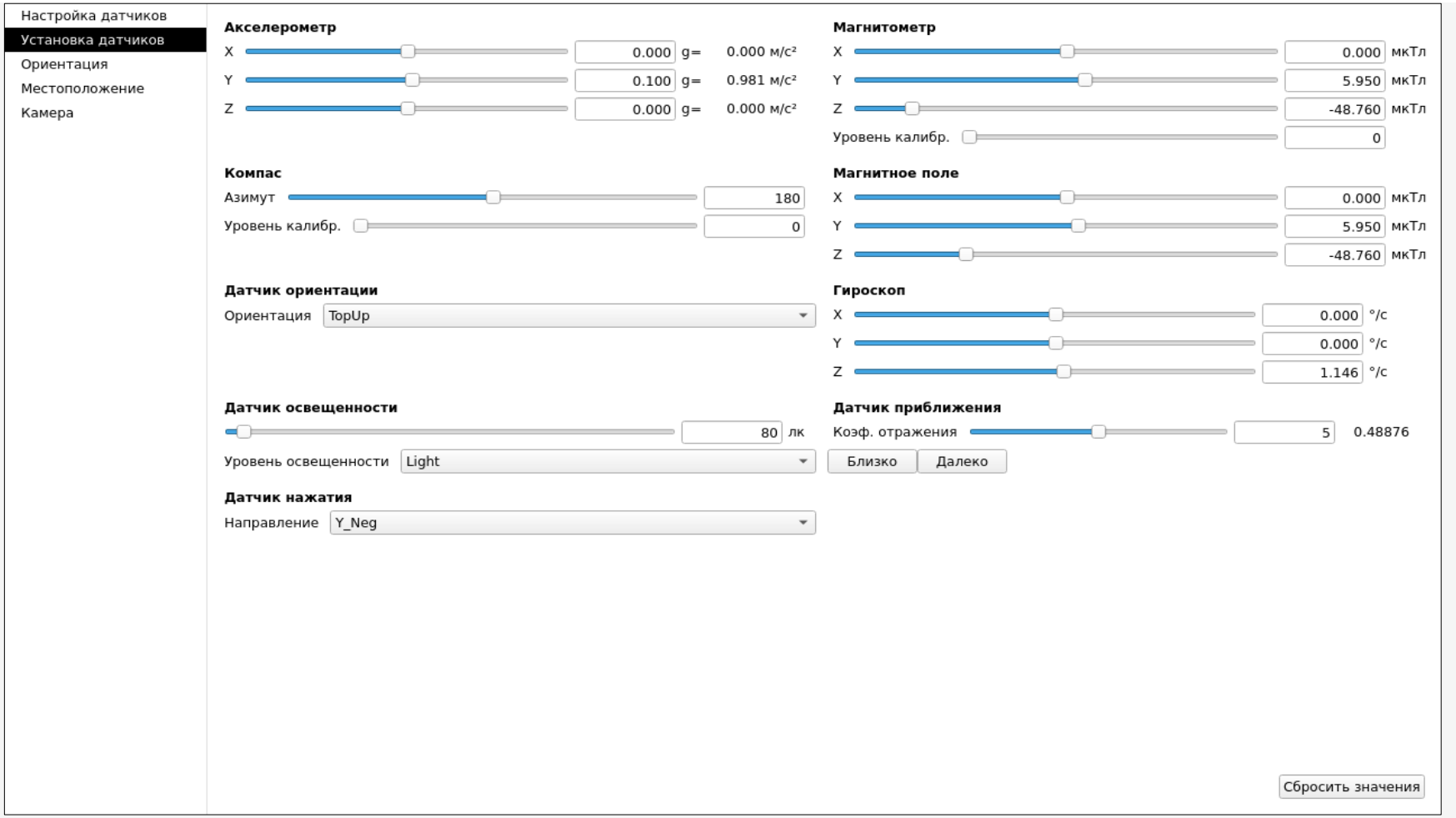
- Эффективная обработка данных датчиков с помощью callback-метода до испускания сигнала
- Возможность скорректировать данные датчика или отменить испускание сигнала

- Привязка фильтра к объекту класса QSensor
  - virtual bool *addFilter*(QSensorFilter *filter);
- Для использования — перегрузить метод
  - virtual bool *filter*(QSensorReading *reading);
- Должен вернуть true, чтобы значение reading было передано дальше

# Классы фильтров

| Класс датчика | Класс фильтра | Тип аргумента метода *filter* |
|---|---|---|
| QProximitySensor | QProximityFilter | QProximityReading* |
| QLightSensor | QLightFilter | QLightReading* |
| QAmbientLightSensor | QAmbientLightFilter | QAmbientLightReading* |
| QAccelerometer | QAccelerometerFilter | QAccelerometerReading* |
| QOrientationSensor | QOrientationFilter | QOrientationReading* |
| QRotationSensor | QRotationFilter | QRotationReading* |
| QGyroscope | QGyroscopeFilter | QGyroscopeReading* |
| QMagnetometer | QMagnetometerFilter | QMagnetometerReading* |
| QCompass | QCompassFilter | QCompassReading* |

# Эмуляция датчиков



True Engineering

**True Engineering**

630128, г. Новосибирск,
ул. Кутателадзе, 4г

(383) 363-33-51, 363-33-50
info@trueengineering.ru
trueengineering.ru

# Новосибирский Государственный Университет