

Модуль Sensors

Модуль библиотеки, для чтения данных сигналов с датчиков углового положения:

- [Чтение сырых данных](#)
- [Чтение приведённых данных](#)
- [Поправочные коэффициенты](#)
- [Примеры работы](#)

Синтаксис

```
ROT1.GetRaw()  
ROT1.GetValue()  
  
ROT1.SetCoeffs(10, 120)
```

Датчик углового положения представляет собой поворотный потенциометр номиналом 10 кОм. Потенциометр работает как делитель напряжения, а значит, что при изменении угла поворота, меняется уровень напряжения на выходе датчика.

Модуль управления Smartli Education использует 10 разрядный АЦП для работы с сигналами датчиков угла поворота. Это значит, что диапазон возможных значений лежит между 0 и 1023.

В модуле управления Smartli Education предусмотрена установка пяти датчиков углового положения: ROT1 ROT2 ROT3 ROT4 ROT5

Все примеры кода будут приводиться для датчика ROT1. Примеры для остальных датчиков углового положения аналогичны.

Чтение сырых данных

Сырыми называют данные, которые получаются непосредственно с аппаратной части Модуля управления Smartli Education.

Чтение сырых данных: `int GetRaw()`

Метод позволяет получить сырые (чистые) данные непосредственно с АЦП в том виде, в каком их возвращает АЦП.

Пример

```
int rawValue = ROT1.GetRaw(); // читаем значение уровня сигнала датчика
```

Чтение приведённых данных: int GetValue()

Метод позволяет получить уровень сигнала датчика, приведённый к диапазону 0 - 100%

Пример

```
int prcValue = ROT1.GetValue(); // читаем значение уровня сигнала ЭМГ-датчика в %
```

Поправочные коэффициенты: void SetCoeffs()

Для улучшения качества, а так же уменьшения шага дискрета приведённого значения показаний датчика, используются коэффициенты аппроксимации.

minCoef - нижняя граница для аппроксимации **maxCoef** - верхняя граница для аппроксимации

Поправочные коэффициенты добавляются к границам диапазона измерений: 0 .. 1023. Формула пересчета:

1. Растягиваем диапазон от: minCoef, maxCoef до: 0, 1023
2. Преобразуем диапазон от: 0, 1023 до: 0, 100

Пример

```
ROT1.SetCoeffs(100, 800);

int value = ROT1.GetValue(); // при сыром значении 763, в переменной value мы получим значение 83.
```

Объяснение

```
GetValue()
{
    int raw = GetRaw();
    int rawCorrection = map(raw, _minC, _maxC, 0, 1023);

    return map(rawCorrection, 0, 1023, 0, 100);
}
```

Первое преобразование с помощью функции `map()` выглядит так:

```
int rawCorrection = map(raw, _minC, _maxC, 0, 1023);
```

Здесь мы подставляем значения переменных:

```
int rawCorrection = map(763, 100, 800, 0, 1023);
```

Это означает, что мы хотим получить значение, которое соответствует положению 763 в диапазоне от 100 до 800, но в диапазоне от 0 до 1023. Для этого мы можем использовать пропорцию:

$$(763 - 100) / (800 - 100) = (x - 0) / (1023 - 0)$$

Решая ее, мы получаем:

```
x = (763 - 100) * (1023 - 0) / (800 - 100)
x = 853.9
```

Округляя до целого числа, мы получаем:

```
rawCorrection = 854
```

Второе преобразование с помощью функции `map()` остается таким же:

```
return map(rawCorrection, 0, 1023, 0, 100);
```

Здесь мы подставляем значение переменной `rawCorrection`:

```
return map(854, 0, 1023, 0, 100);
```

Это означает, что мы хотим получить значение, которое соответствует положению 854 в диапазоне от 0 до 1023, но в диапазоне от 0 до 100. Для этого мы можем использовать пропорцию:

$$(854 - 0) / (1023 - 0) = (y - 0) / (100 - 0)$$

Решая ее, мы получаем:

```
y = (854 - 0) * (100 - 0) / (1023 - 0)
y = 83.4
```

Округляя до целого числа, мы получаем:

```
y = 83
```

Таким образом, метод GetValue() вернет значение **83**.

Примеры

Работа с датчиком углового положения

```
#include <smartlieducation.h>

void setup() {
  ROT1.SetCoeffs(100, 800);
}

void loop() {
  int angel = map(ROT1.GetValue(), 0, 100, 0, 359);

  if (angel > 120) {
    Smartli.Vibrate(200, 2);
  } else if (angel > 270) {
    Smartli.Vibrate(530, 4);
  } else {
    Smartli.Vibrate(1000, 1);
  }
}
```