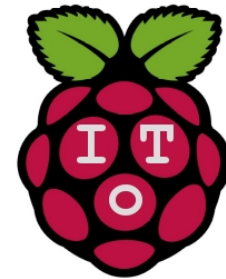




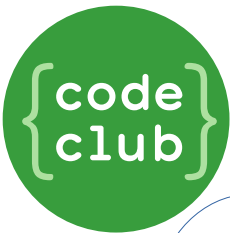
# Internet of Things



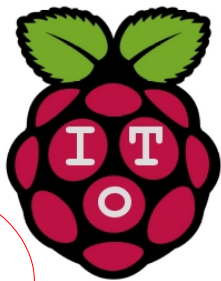
## Измерение и принятие решений

Шадринск  
2018-2019

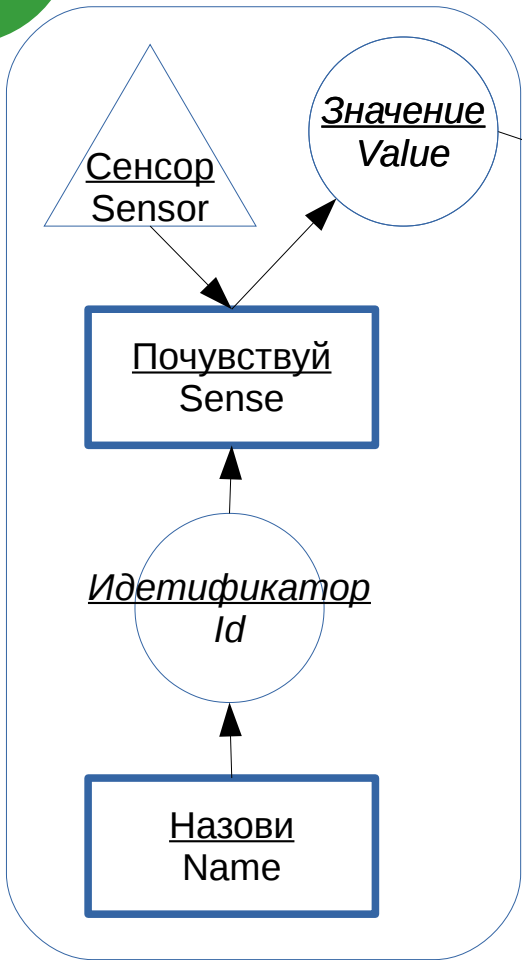
*М. В. Шохирев*



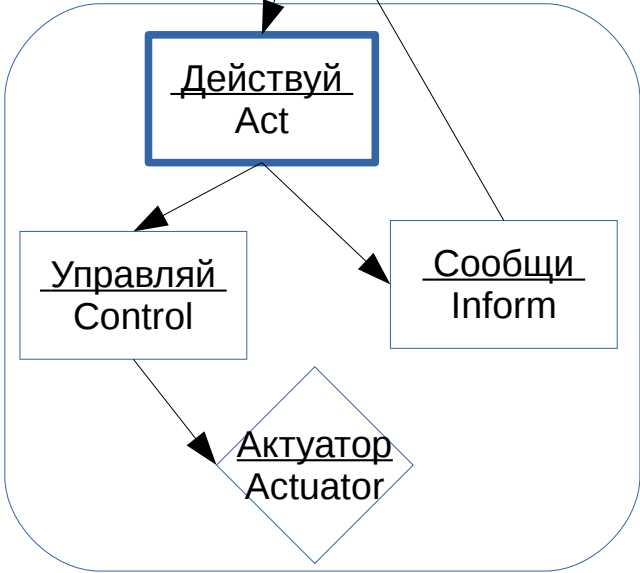
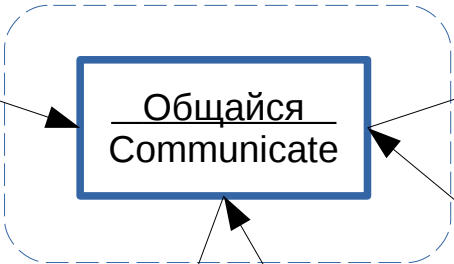
# Схема IoT



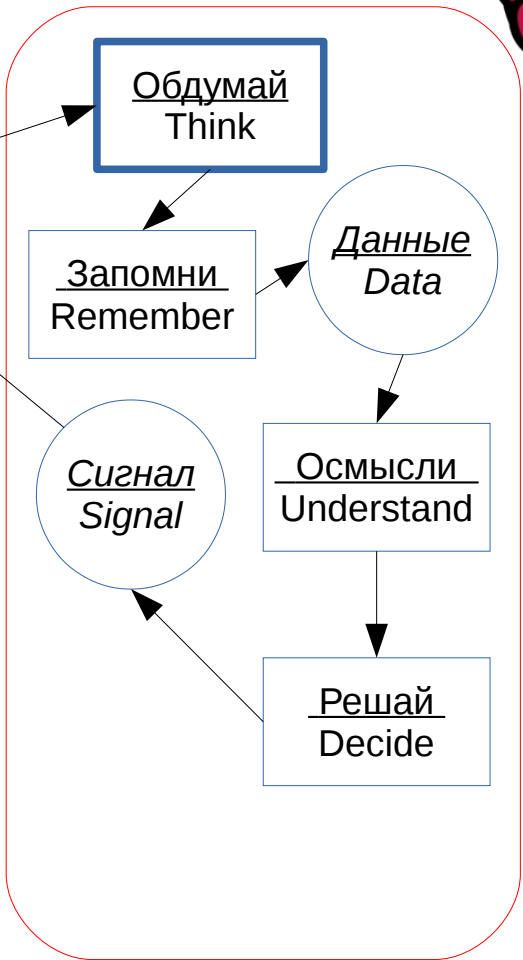
Клиент  
Client



Шлюз  
Gateway

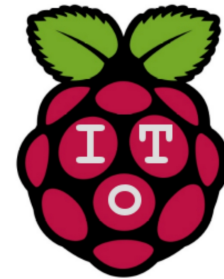


Сервер  
Server





# Сигналы и данные



*Сигнал* — (отсутствие изменения или) изменение физической величины во времени, несущее информацию о ней.

## Сведения о физических явлениях

- Вкусы, запахи
- Звуковые колебания
- Освещённость, цвета
- Температура, давление, влажность
- Сила, скорость, ускорение
- Направление, расстояние
- Время
- Электричество:  $R$ ,  $V$ ,  $I$
- ...

## Сигналы → измерения → величины

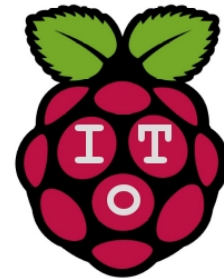
Сигналы с измерительных приборов преобразуются в *показания*, которые можно регистрировать и обрабатывать.

*Величины* показаний могут выражаться в определённых *единицах измерения*.

*Данные* — зарегистрированная в понятной форме информация о показаниях, событиях, фактах.

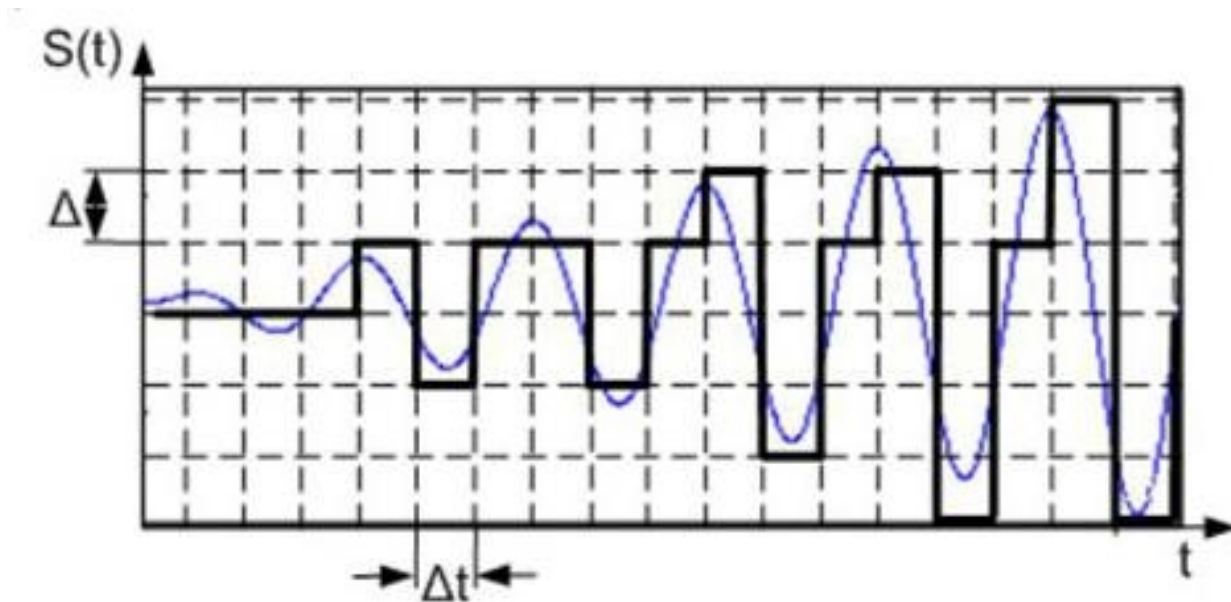


# Аналоговый | цифровой



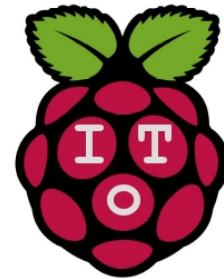
Большинство сигналов — **аналоговые сигналы**, которые непрерывно изменяются во времени и могут принимать любые значения на некотором интервале.

**Цифровые сигналы** представляются последовательностями отдельных цифровых значений, которые могут быть получены при «оцифровке» аналоговых сигналов.

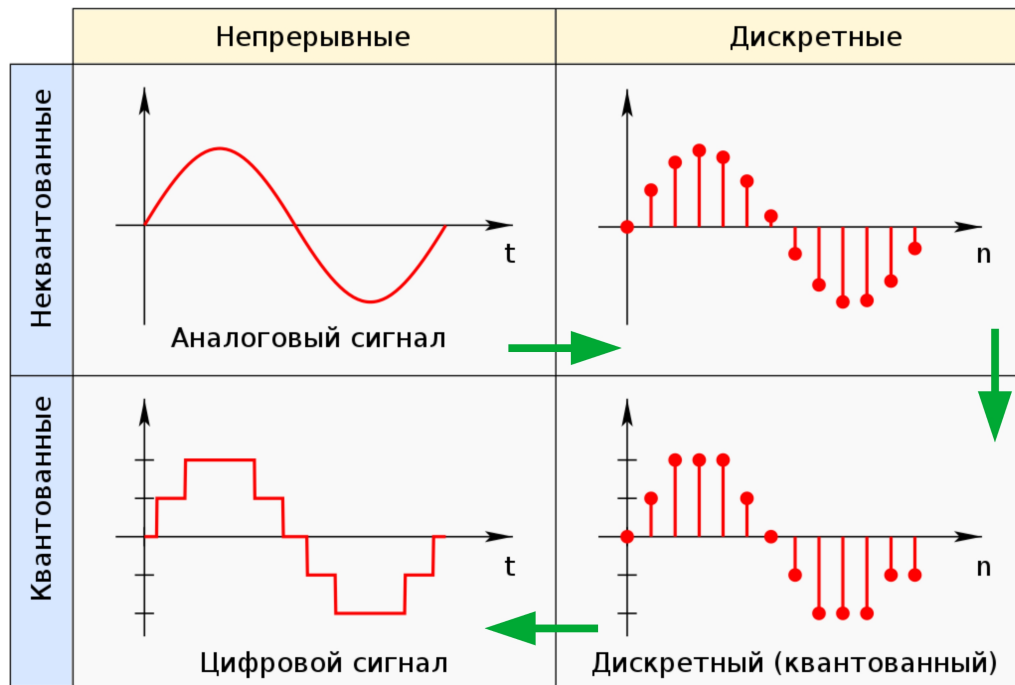




# Непрерывный | дискретный

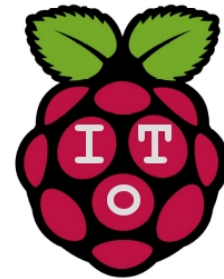


Термины *аналоговый* и *цифровой* соотносятся терминами *непрерывный* (непрерывно изменяющийся во времени) и *дискретный* (прерывистый — измеренный в определенные периоды времени).

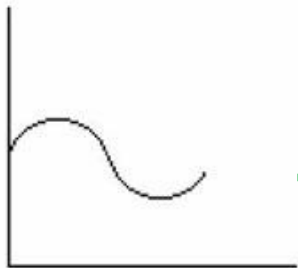




# Аналоговый → цифровой



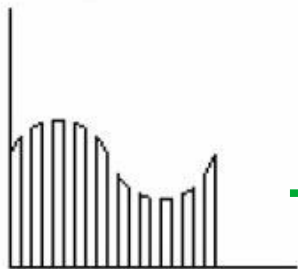
Аналоговый



Дискретизация



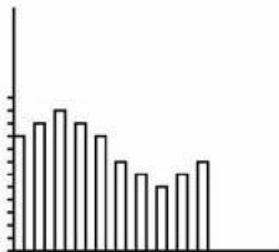
Дискретный



Квантование



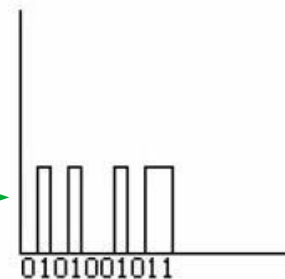
Квантованный



Кодирование



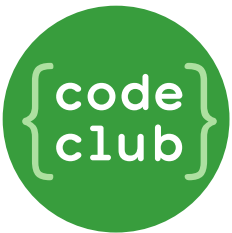
Цифровой



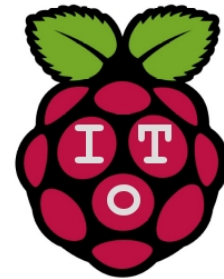
1. Чтобы представить непрерывно изменяющийся аналоговый сигнал как последовательность чисел, его следует сначала превратить в дискретный сигнал (последовательность значений в дискретные моменты времени).

2. Затем его нужно подвергнуть квантованию, в ходе которого дискретные значения сигнала на каждом промежутке (кванте) времени соотносятся с приближённым целым числом (уровнем квантования в числах заданной разрядности).

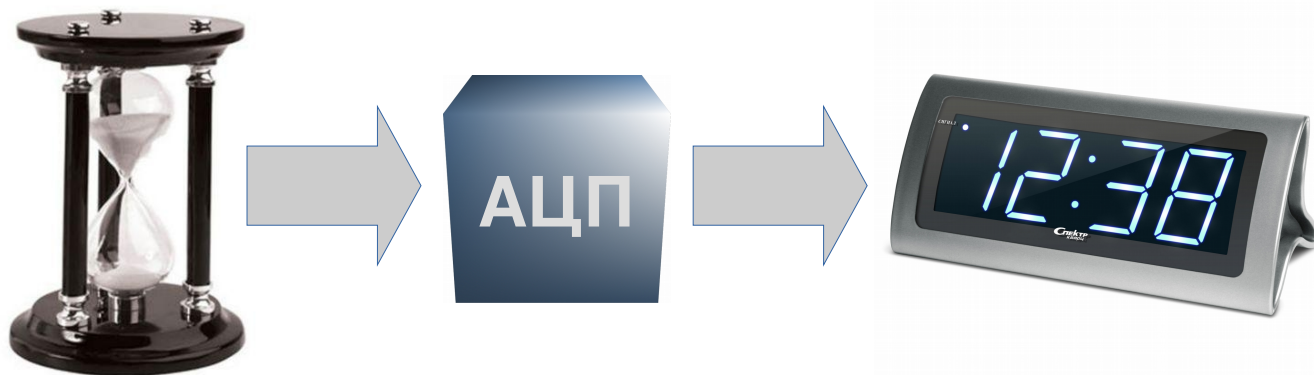
3. Если записать эти целые числа в двоичной системе счисления (закодировать), то получится последовательность нулей и единиц (код), которая и будет цифровым сигналом, представляющим исходный аналоговый сигнал.

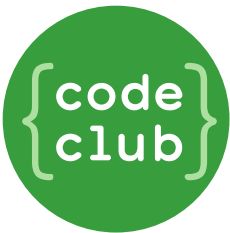


# ADC, DSP

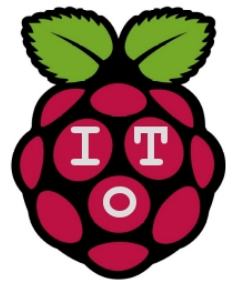


Преобразование производится с помощью специализированных устройств: *аналого-цифровых преобразователей* (АЦП или ADC = analog to digital converter) значений или *цифровых процессоров обработки сигналов* (ЦПОС или DSP = digital signal processor).

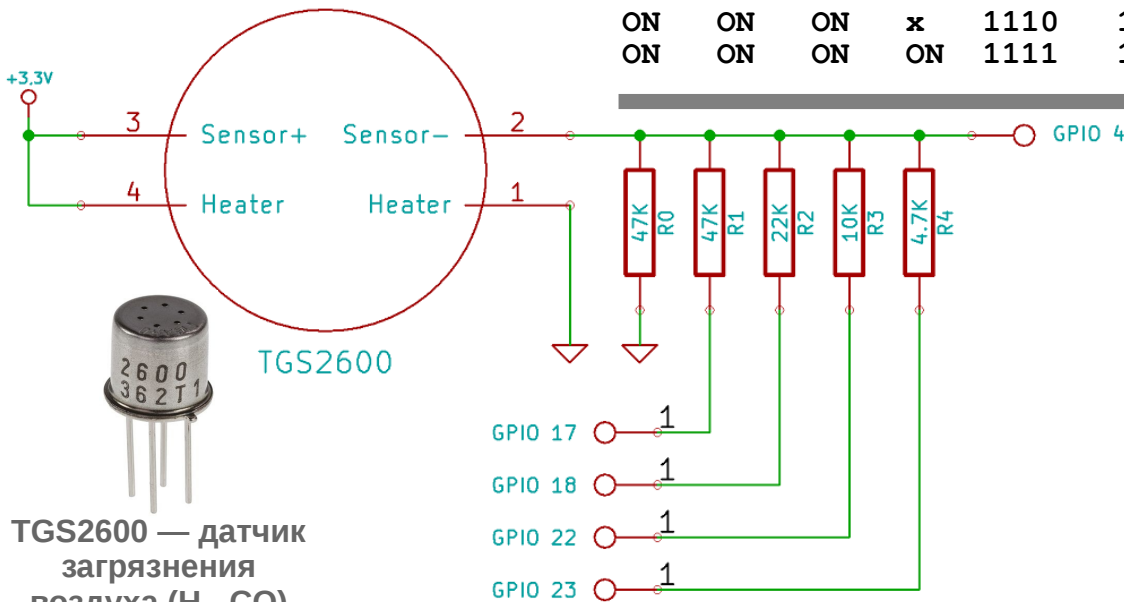




# Аналог → биты → число



Преобразование аналоговых значений в цифровые без АЦП с помощью резисторной матрицы (resistor ladder).



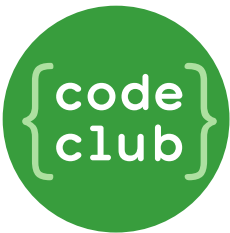
R1	R2	R3	R4	Binary	Decimal
x	x	x	x	0000	0
ON	x	x	x	1000	8
ON	ON	x	x	1100	12
ON	ON	ON	x	1110	14
ON	ON	ON	ON	1111	15

Можно программно считывать значения с контактов GPIO, соответствующих отдельным битам двоичного числа, и с помощью побитовых операций превращать набор битов в целое число, представляющее показание датчика.

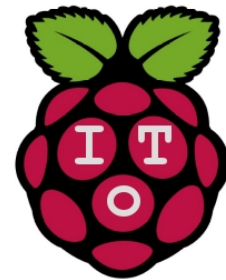
Такая резисторная матрица выполняет роль 4-битного АЦП.

TGS2600 — датчик загрязнения воздуха (H<sub>2</sub>, CO) аналоговый

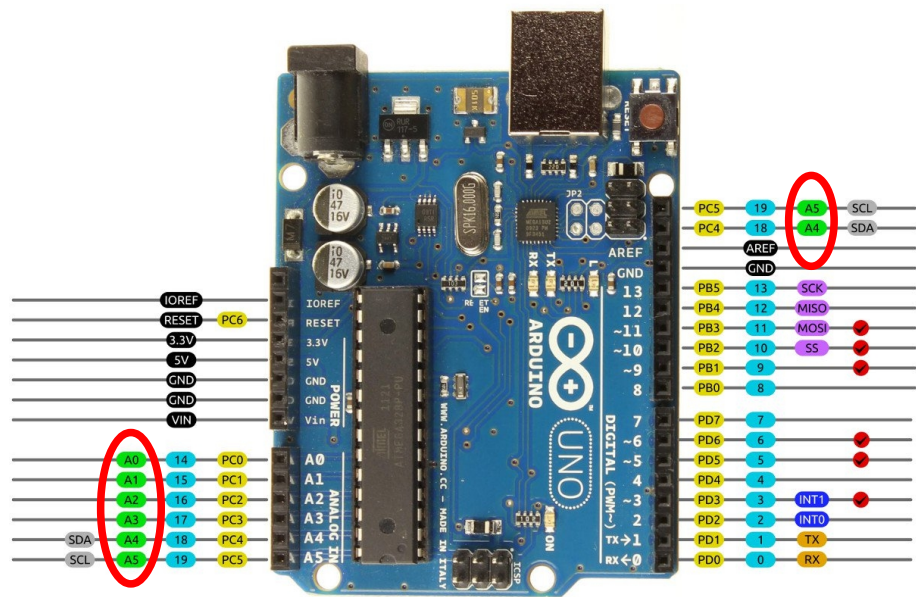




# Входы GPIO: ~~аналоговые~~ и цифровые



## Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

**Аналоговые входы** у микроконтроллеров и одноплатных компьютеров — это входы, оснащённые АЦП, которые оцифровывают аналоговые значения с заданной разрядностью: например, в Arduino они 10-битные.

Разрядность АЦП:

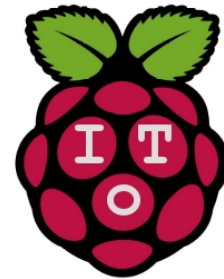
8 бит = 256 значений: 0..FF  
10 бит = 1024 значения: 0..3FF  
12 бит = 4096 значений: 0..FFF  
16 бит = 65536 значений: 0..FFFF  
20 бит = 1048576 значений: 0..FFFFFF  
24 бита = 16777216 значений: 0..FFFFFFFF  
32 бита = 4294967296 значений: 0..FFFFFFFF



2014 by Bouni  
Photo by Arduino.cc



# Типы датчиков



**Аналоговые датчики** выдают аналоговые значения, которые до использования нужно преобразовывать в цифровые с помощью АЦП.

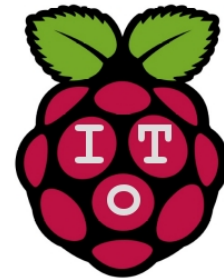
**Цифровые датчики** выдают данные сразу в цифровом виде.

3 типа цифровых датчиков:

<p>«<b>двоичные</b>»/ «<b>логические</b>»: 0/1, нет/да, не достигнут/ достигнут предел, выключено/включено</p> <p><u>Пример:</u> дверь открыта — 1, закрыта — 0;</p>	<p>«<b>относительные</b>»/ «<b>абстрактные</b>»: 0..предельное значение точности (8 бит: 0..255)</p> <p><u>Пример:</u> угол поворота 128 единиц;</p>	<p>«<b>абсолютные</b>»/ «<b>конкретные</b>»: значения в известных единицах измерения.</p> <p><u>Пример:</u> температура 25.5°C;</p>
--	--	---



# Аналоговые датчики

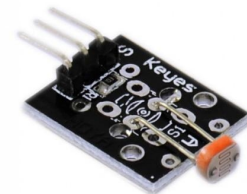


**Аналоговые датчики** выдают аналоговые значения, которые до использования нужно преобразовывать с помощью АЦП.



**TGS2600 — датчик  
загрязнения  
воздуха ( $H_2$ , CO)  
аналоговый**

Преобразует уровень газообразных загрязнений воздуха (окись углерода  $CO$ , водород  $H_2$ ) в величину постоянного напряжения, пропорциональную концентрации обнаруживаемых примесей в воздухе.

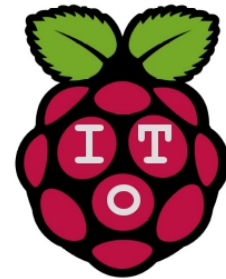


**KY-018 — датчик  
уровня освещённости  
аналоговый**

Фоторезистор меняет сопротивление в зависимости от интенсивности света: при отсутствии света сопротивление фоторезистора большое и доходит до 1 МОм, а при его освещении падает до нескольких Ом. С увеличением освещённости на выходе модуля напряжение будет падать: при ярком свете напряжение выхода будет около половины напряжения питания, а в темноте напряжение выхода будет близко к напряжению контакта +5 В.



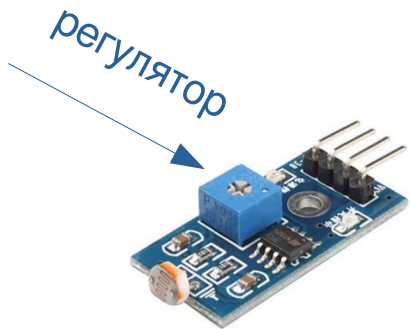
# Двоичные датчики



**Двоичные датчики** выдают только 2 цифровых логических значения: 0/1, нет/да, не достигнут/достигнут предел, выключено/включено.

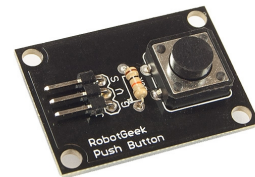
## Примеры:

ниже заранее  
отрегулированного уровня  
освещённости — 0, выше  
него — 0; дверь открыта —  
1, закрыта — 0; кнопка  
нажата — 1, отжата — 0;



КУ-018 — датчик  
уровня освещённости  
цифровой двоичный

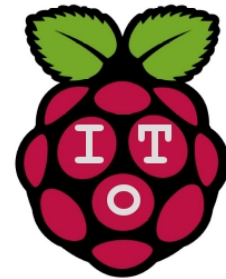
На модуле имеется регулятор (*переменный резистор*), с помощью которого можно установить пороговое значение уровня освещённости, по достижению которого на выходе будет выдаваться 1.



КУ-004 — кнопка  
контактная  
нефиксирующаяся



# Относительные датчики

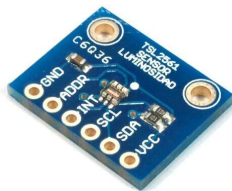


**Относительные датчики** выдают абстрактные данные в цифровом виде: 0..предельное значение точности (8 бит: 0..255, 10 бит: 0..1023, 12 бит: 0..4095, 16 бит: 0..65535, 24 бита: 16777215).

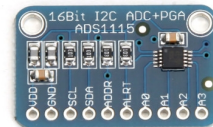
Такие же значения выдают АЦП, к которым подключены аналоговые датчики.

## Примеры:

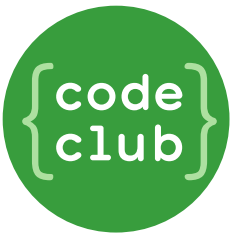
угол поворота 128 единиц;  
влажность 16 единиц;  
напряжение 240 единиц;  
уровень освещённости 32 единицы;



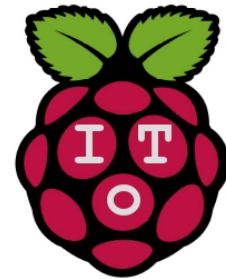
**TSL2561** — датчик  
уровня освещённости  
цифровой  
(преобразование  
значений в lux  
производится программно  
по известной  
эмпирической формуле)



**ADS1115** — АЦП  
4-канальный  
16-битный  
с интерфейсом I<sup>2</sup>C



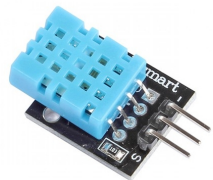
# Абсолютные датчики



**Абсолютные датчики** выдают данные в цифровом виде значения в конкретных единицах измерения (градусы, проценты, бары, люксы, метры, вольты, амперы, омы, секунд)

## Примеры:

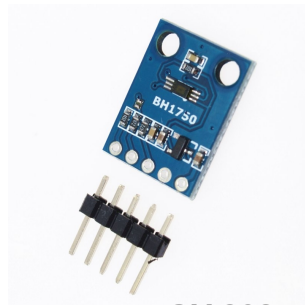
температура +20.2°C;  
влажность 32%;  
атмосферное давление 764 mm Hg;  
скорость ветра 4.5 м/сек;  
геомагнитное поле 2 балла;  
напряжение на батарее 4.8V;  
дифференциал (на корму) - 12°;  
курс 45° (от N);



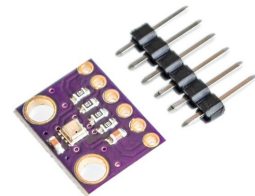
DHT11 — датчик температуры и влажности (целые значения)



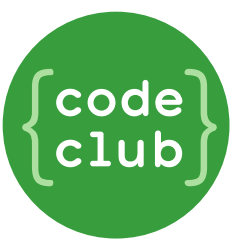
DHT22 — датчик температуры и влажности (дробные значения)



Модуль GY-302: цифровой датчик освещённости BH1750 с интерфейсом I<sup>2</sup>C



BME280 — цифровой датчик температуры, влажности и атмосферного давления с интерфейсом I<sup>2</sup>C



# Шкалы и единицы измерения

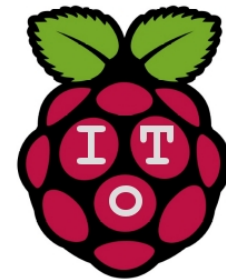
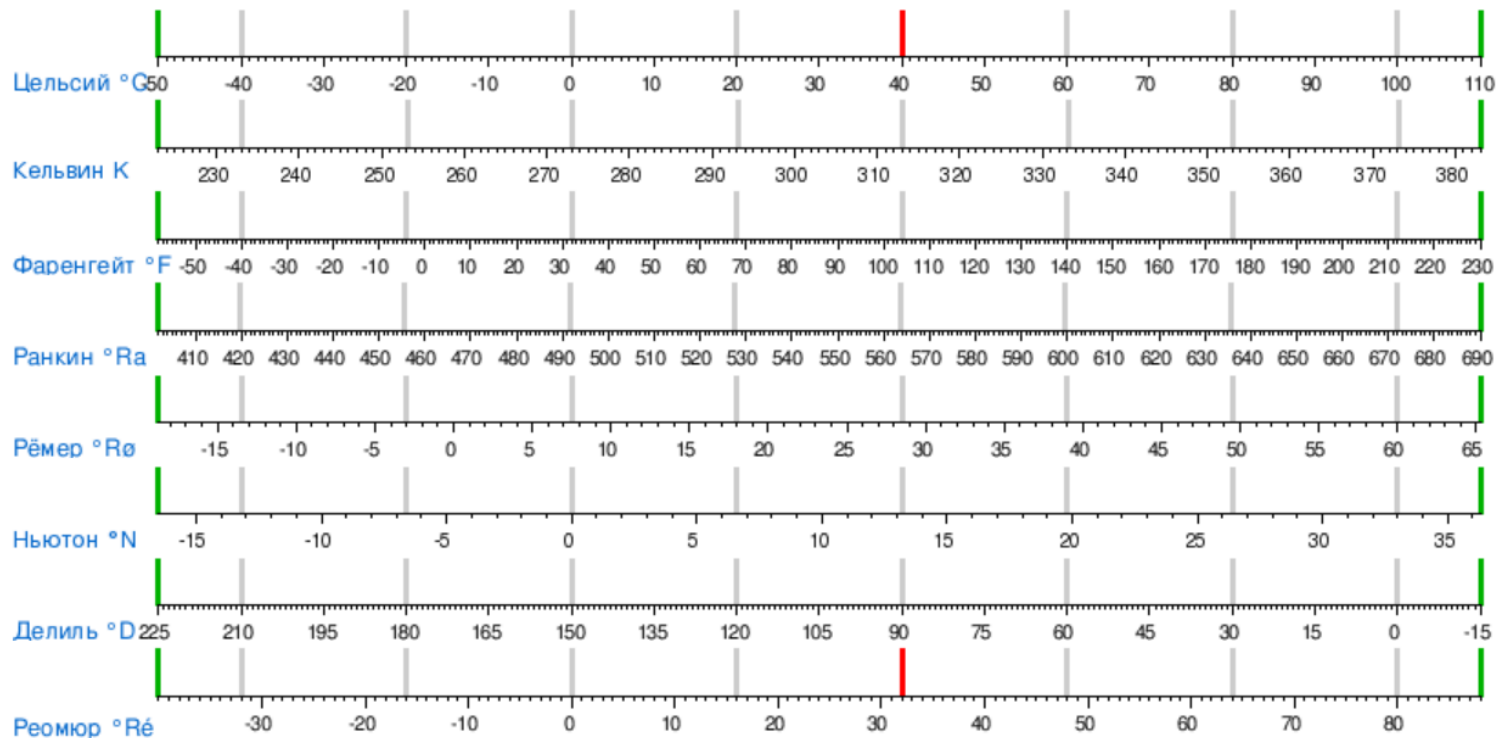
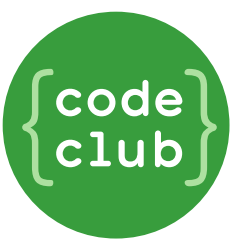


Диаграмма перевода температур [\[править\]](#) [\[править код\]](#)



40 °C = 313,15 K = 104 °F = 563,67 °Ra = 28,5 °Rø = 13,2 °N = 90 °D = 32 °R



# Шкала и выводы

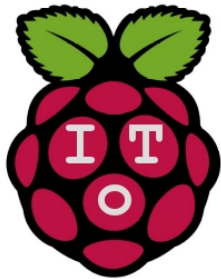
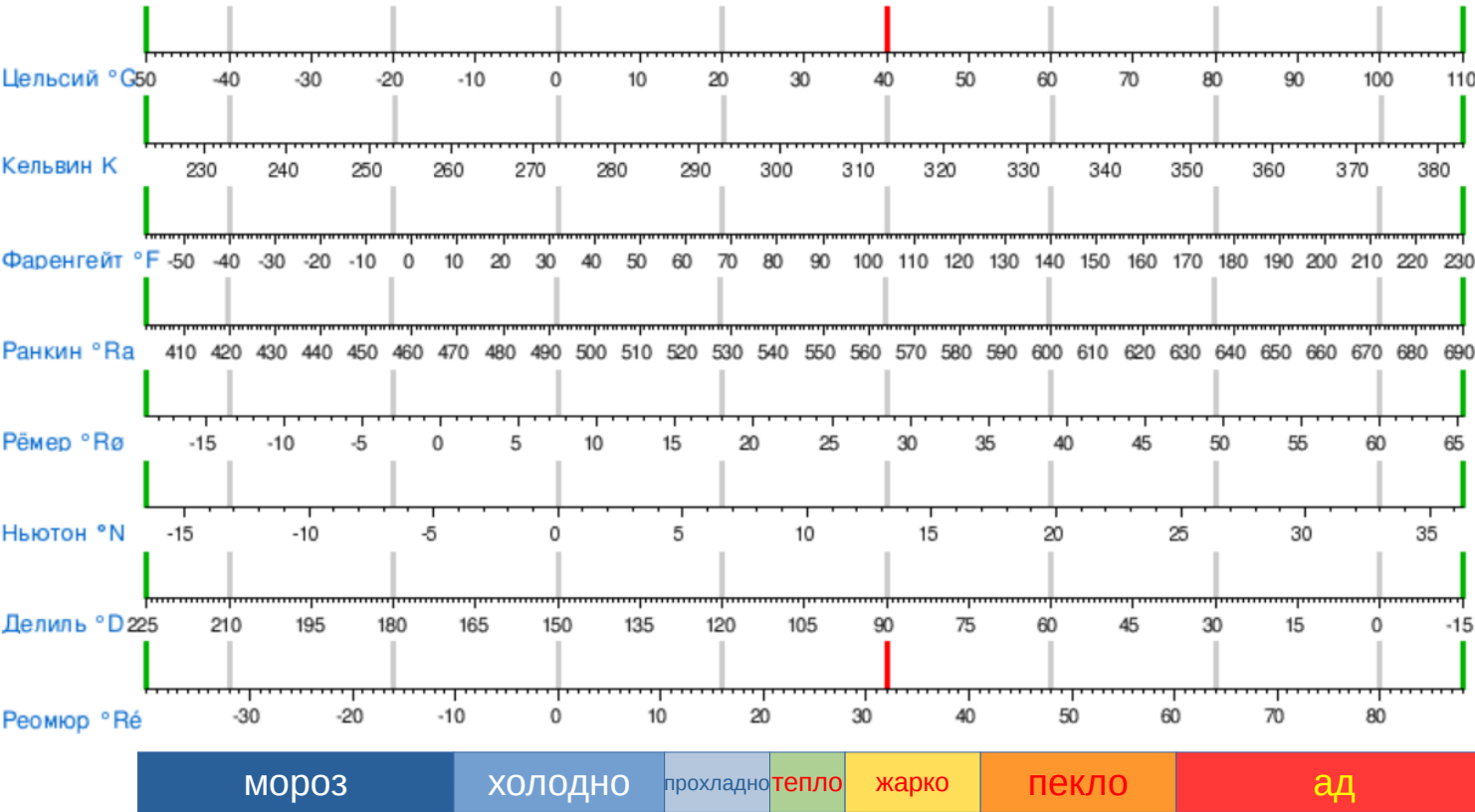
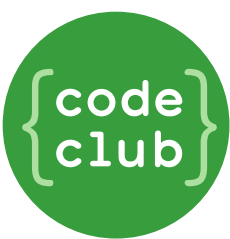


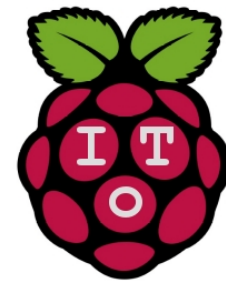
Диаграмма перевода температур [\[ править \]](#) [\[ править код \]](#)







# Практическая шкала



## Ветровая шкала Бофорта

Фрэнсис Бофорт (1774-1857),  
английский военный  
гидрограф и картограф,  
контр-адмирал, в 1806 г.  
предложил оценивать силу  
ветра по его воздействию на  
наземные предметы и по  
волнению моря; для этого он  
разработал условную 12-  
балльную шкалу.



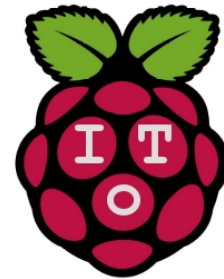
Скорость ветра, м/сек

0,0	0 баллов	
0,3 - 3,3	1-2 балла	
3,4 - 7,9	3-4 балла	
8,0 - 13,8	5-6 баллов	
13,9 - 20,7	7-8 баллов	
20,8 - 28,4	9-10 баллов	
28,3 - >33	11-12 баллов	

Иногда вместо точных значений в известных единицах измерения удобнее пользоваться упрощённой *практической шкалой*, диапазоны которой соотносятся с определёнными соображениями (ситуациями, выводами, решениями, действиями).



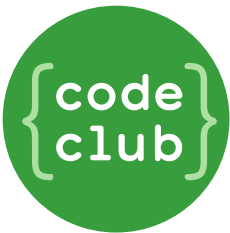
# Виды шкал



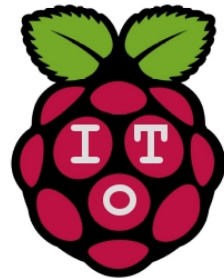
Практические шкалы, когда допустимые значения делятся на несколько *диапазонов*, часто используются как условия для оценки ситуации и принятия решений по выдаче управляющих сигналов.






- «**Единичная шкала**» — регистрация факта, что событие произошло.
- **Двоичная шкала** с пороговым значением: до достижения его — одно состояние, после достижения — другое.
- **Шкала-«светофор»** (для величин, изменяющихся в одну сторону) с тремя диапазонами: **зелёный** (нормальное состояние), **жёлтый** (допустимое отклонение от нормы), **красный** (опасное отклонение от нормы = исключительная).
- «**Двухсторонний светофор**» (для величин, изменяющихся в обе стороны) с пятью диапазонами: **красный** — **жёлтый** — **зелёный** — **жёлтый** — **красный**.
- **Циклическая шкала** — наступление очередного диапазона или периода времени: сезоны (весна — лето — осень — зима), месяцы, дни недели, время суток (утро — день — вечер — ночь), часы в сутках, минуты часа.

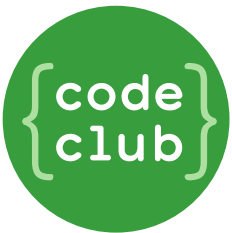
***А какие шкалы могут быть ещё?***



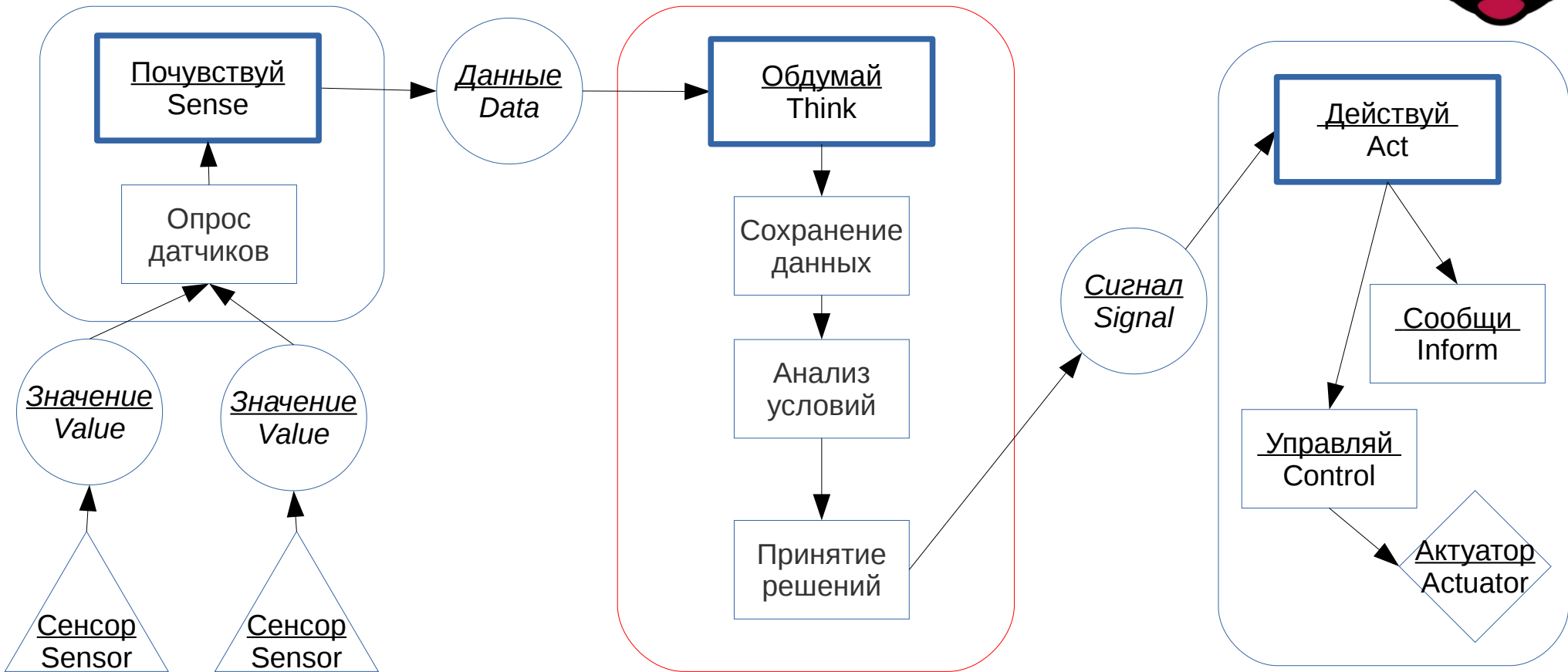
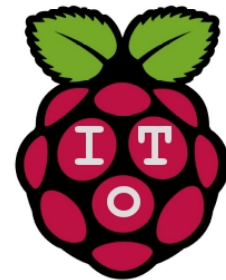
# Шкалы и значения



<b>«Единичная шкала»</b> 	определение наступления события	устройство подключилось к сети; пришло сообщение по э-почте; получен сигнал SIGINT прерывания (Ctrl-C) с терминала; ОС загрузилась;
<b>Двоичная шкала</b> 	(не)достижение порогового значения	батарея разрядилась до критически низкого уровня; температура процессора достигла уровня перегрева; скорость ветра не позволит летать;
<b>Шкала-светофор</b> 	3 диапазона: норма, допустимое и опасное отклонение от нормы	дискового пространства много — диск заполнен значительно — места почти не осталось; банкомат полон — пора пополнить банкноты — денег нет;
<b>Двухсторонний светофор</b> 	5 диапазонов: норма, ± допустимое и опасное отклонение от неё	вода опустилась ниже допустимого уровня — поднялась до нижней нормы — превысила верхнюю норму — дошла до верхнего предела;
<b>Циклическая шкала</b> 	нахождение в очередном диапазоне, наступление / окончание следующего периода времени	ночью производится резервное копирование данных; в конце года чистятся протоколы; датчик давления опрашивается каждый час; условия проверяются каждую минуту;

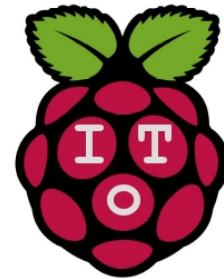


# Принятие решений





# Условия и решения



**Принятие управляющих решений основано на проверке условий :**

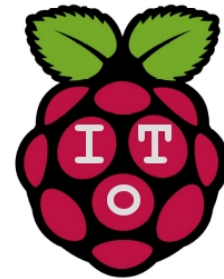
- Наступление (регистрация) события.
- Достижение порогового значения.
- Вхождение отслеживаемой величины в диапазон шкалы или выход из него.
- Нахождение в очередном диапазоне на циклической шкале, наступление или окончание следующего временного периода.
- ...

**Для принятия решений учитываются комбинации нескольких условий:**

- отрицание условия: логическое NOT (инверсия условия)
- объединение условий — логические операции:
  - AND (И, логическое умножение, одновременное выполнение условий)
  - OR (ИЛИ, логическое сложение, выполнение хотя бы одного из условий)
  - XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ, выполнение только одного из условий, но не обоих)



# Флаги и условия



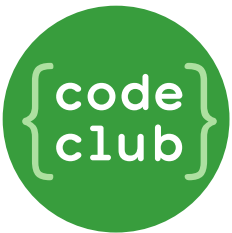
Для подготовки к оценке состояния (ситуации) удобно при проверке многочисленных условий устанавливать определённые логические переменные (*флаги*) которые будут хранить результаты предыдущих проверок условий и их комбинаций.

```
wind_ok = anemometer.wind_velocity < TOO_WINDY_TO_FLY
warm_outdoor = temperature_sensor.value < TEMPERTURE_TOO_HIGH
cold_outdoor = temperature_sensor.value > TEMPERTURE_TOO_LOW
weather_good = wind_ok && warm_outdoor && !cold_outdoor
board_temperature_normal = aircraft.temperature < CRITICAL_T
battery_normal = battery.procent > BATTERY_LEVEL_TOO_LOW
aircraft_ready = board_temperature_normal && battery_normal
gps_ok = gps_sensor.number_of_sattelites >= MIN_SATELLITES_LOCKED

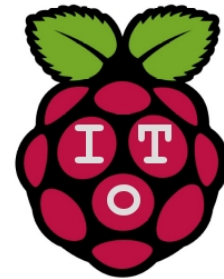
ready_to_fly = weather_good && aircraft_ready && gps_ok
```

Тогда итоговый анализ условий для принятия решения становится значительно проще:

```
if ready_to_fly then aircraft.takeoff
```



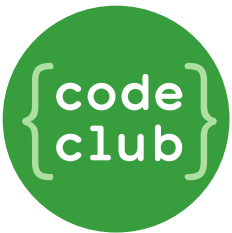
# Флаги и триггеры



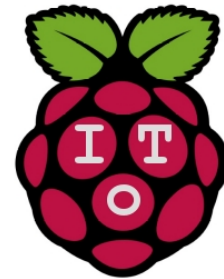
Часто реакцию на события (отправку управляющих сигналов) выполняют *триггеры* – программы, которые автоматически срабатывают при установке определённого флага в состояние `true`.

Такие программы часто работают *асинхронно* с основной программой в виде параллельных процессов.

При применении триггеров логика основной программы упрощается: в ней достаточно только установить нужный флаг.



# История наблюдений



Многие управляющие решения принимаются с учётом *изменения данных во времени*. Для этого все полученные показания вместе с моментом измерения нужно хранить в *журнале наблюдений*.

Исходя из истории наблюдений можно получить важные *вычисляемые данные*:

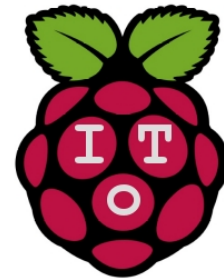
- время, прошедшее с последнего момента измерений (timeout);
- минимальное, максимальное, среднее значение (за период времени);
- тенденция изменения значений: уменьшение, сохранение, увеличение;
- скорость изменения значений (за период);
- ускорение изменений: скорость изменений уменьшается или увеличивается;

Эти данные позволяют судить о том, как изменяется ситуация, и даже делать предположения о том, какой она будет в ближайшем будущем (*прогнозировать развитие ситуации*).





# Решения на основе истории

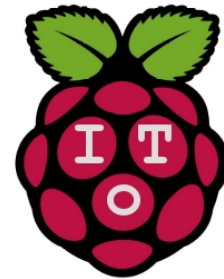


Примеры:

*«Отопительный период должен начинаться или заканчиваться со дня, следующего за днём окончания 5-дневного периода, в течение которого соответственно среднесуточная температура наружного воздуха ниже  $8^{\circ}\text{C}$  или среднесуточная температура наружного воздуха выше  $8^{\circ}\text{C}$ .»*



# Определение понятий



Для анализа данных и принятия решений может потребоваться программно определить понятия, изложенные в требованиях:

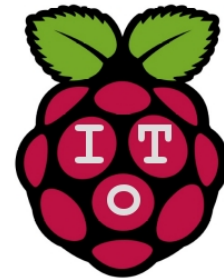
- «дневное / ночное время»;
- «последний месяц»;
- «допустимое время ожидания / задержки»;
- «предельное значение»;
- «комфортная температура»;
- . . .

Эти определения практично сделать в виде констант, вынесенных из программы в настроечный файл или базу данных.

***Какие ещё понятия вы слышали?***



# Источники



## Ссылки на Интернет-ресурсы:

- Чем отличаются аналоговый сигнал от цифрового – примеры использования
- Датчики и микроконтроллеры. Часть 1
-