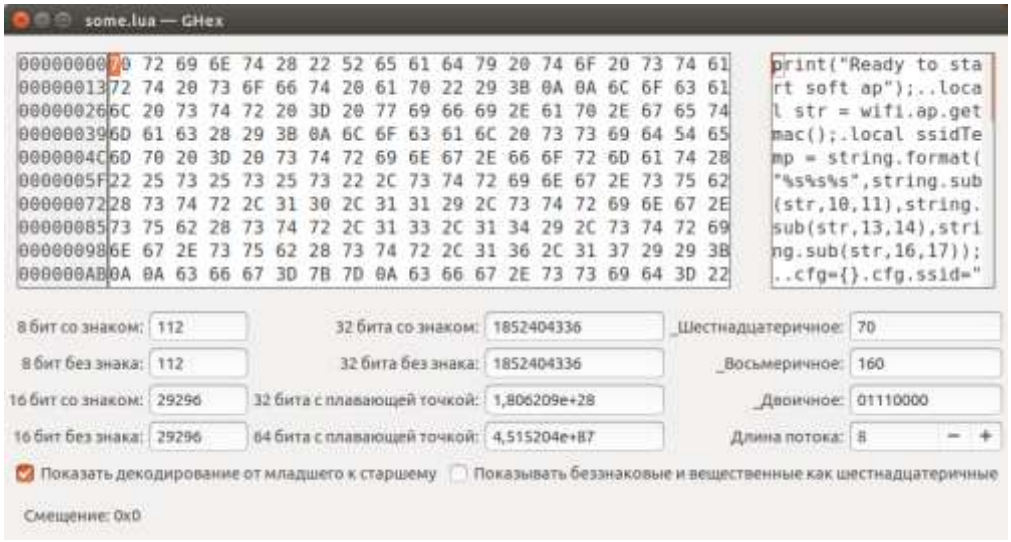
Что понимает микроконтроллер?

В своей бытовой (и не совсем) деятельности людей учат и они привыкают работать с десятичной системой счисления (да, да — те десять арабских циферок). Но при работе с цифровой техникой и написании программного обеспечения для персонального компьютера, и тем более микроконтроллера, часто используются другие системы счисления. О них мы и поговорим ниже.



Десятичная система счисления

Здесь ничего сложного нет, вы знакомы с этой системой с детства. Алфавит включает числа `{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}`, а основание — `10`. Давайте распишем число 256:

> 2 | 1 | 0 2 | 5 | 6

Тогда:

> $2 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 = 200 + 50 + 6 = 256_{10}$

Внимание! При написании программ на языке Си стоит помнить, что 45 ≠ 045. Ведущий ноль обозначает **восьмеричную систему счисления**. Избегайте желания сделать код красивым путем дописывания нолей.

Двоичная система счисления

Основание, как несложно догадаться, в этой системе — число 2, а алфавит состоит из `{0, 1}`. Допустим, у нас есть число, записанное в бинарном виде, 1010_2. Давайте его распишем:

> $3|2|1|0\ 1|0|1|0_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8 + 0 + 2 + 0 = 12_{10}$

Некоторые компиляторы языка Си (такие как GCC) поддерживают префикс `0b` для записи бинарных значений, например: `0b1010` = 12.

Шестнадцатеричная система счисления

Основание этой системы — `16`, алфавит `{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}`. В языке Си вы можете использовать специальный префикс `0x` для того, чтобы обозначить шестнадцатеричную систему.

> $1111_2 = 0xF_{16} = 15_{10}$

Одно шестнадцатеричное число составляет один полубайт (4 бита), который также называют нибблом. Переведем в десятичную систему:

> $0xA5_{16} = 11 * 16^1 + 5 * 16^0 = 176 + 5 = 181_{10}$

Быстро преобразовать из двоичной системы в шестнадцатеричную (и обратно) можно так:

> **bin**: 0101010010110101 **nibbles**: 0101 0100 1011 0101 **hex:** 5 4 B 5

Представление чисел

Процессор (или микроконтроллер) не умеет работать с десятичными числами, вся информация представляется в виде нулей и единиц. Мы уже посмотрели, как записывается положительное целое число:

> $1011\ 0100_2 = 180_{10}$

Здесь весь байт (8 бит) описывает неотрицательное число (англ. unsigned). Для того чтобы представить число со знаком (англ. signed), один старший разряд отводится под знак. «0» означает положительное число, «1» — отрицательное (в двухкомпонентном случае последний бит равен -128, в однокомпонентном просто инвертируется знак).

> $0000\ 1111_2 = 8 + 4 + 2 + 1 = 15_{10}\ 1000\ 1111_2 = -128 + 8 + 4 + 2 + 1 = -113_{10}$

Впрочем, так глубоко вам залезать, скорее всего, не придется. Самое время поговорить о логических уровнях.

Назад | Оглавление | Дальше