

## Системы счисления используемые в контроллерах DirectLOGIC

### Двоичная система счисления - Binary Numbering System

Компьютеры и программируемые контроллеры используют систему счисления с основанием «2», которую называют двоичной или булевой (Binary или Boolean). В этой системе используются только две цифры НОЛЬ и ЕДИНИЦА (или ON и OFF).

Для представления больших используются комбинации из нескольких цифр.

- Каждая цифра двоичной системе при использовании в компьютере называется – «бит».
- Группа из четырех бит образует – тетраду (nibble).
- Восемь бит или две тетрады – это байт (byte).
- Шестнадцать бит или два байта – это слово (word).
- Тридцать два бита или два слова – это двойное слово (double word).

Word (Слово)															
Byte (Байт)								Byte (Байт)							
Nibble (Тетрада)				Nibble (Тетрада)				Nibble (Тетрада)				Nibble (Тетрада)			
Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

В таблице показаны двоичные числа и их десятичные эквиваленты.

Например: значение байта 110101012 равно 213 ( $1*2^7 + 1*2^6 + 1*2^4 + 1*2^2 + 1*2^0$  или  $128_{10} + 64_{10} + 16_{10} + 4_{10} + 1_{10}$ )

	<b>Binary / Decimal – Двоичные / Десятичные числа</b>															
Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Десятичное значение числа	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Макс. значение	<b>65535<sub>10</sub></b>															

### Шестнадцатеричная система счисления - Hexadecimal Numbering System

Недостаток двоичной системы при ее использовании человеком являются большая длина числа и сложная интерпретация / преобразование числа.

Для устранения этих недостатков используются промежуточные системы. Одна из альтернативных систем счисления – это шестнадцатеричная (Hexadecimal или Hex).

Основание этой системы число 16. Представление чисел осуществляется цифрами от 0 до 10 и буквами от А до F

Десятичная	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Шестнадцатеричная	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

При работе с двоичной системой каждая тетрада (Nibble) может быть легко преобразована в шестнадцатеричный и десятичный эквивалент:

**Тетрада 1111<sub>2</sub> эквивалентна 15<sub>10</sub> или F Hex.**

Шестнадцатеричное слово можно представить в виде шестнадцатеричного эквивалентного числа в диапазоне от 0000 до FFFF Hex.

	<b>Шестнадцатеричное число - Hexadecimal</b>															
Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень	$16^3$				$16^2$				$16^1$				$16^0$			
Значение бита	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
Максим. значение	<b>F</b>				<b>F</b>				<b>F</b>				<b>F</b>			

A6D4<sub>16</sub> эквивалентно 42708<sub>10</sub> ( $10*16^3 + 6*16^2 + 13*16^1 + 4*16^0$  или  $40960_{10} + 1536_{10} + 208_{10} + 4_{10}$ ).

## Восьмеричная система счисления - Octal Numbering System

Восьмеричная система счисления похожа на шестнадцатеричную систему в интерпретации бит. Отличия: основание –8 и максимальное число-7. Двоичное слово разбивается на триады.

		<b>Восьмеричное число - Octal</b>															
Номер бита		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень		$8^3$			$8^2$			$8^1$			$8^0$						
Значение бита		1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1
<b>Максим. значение</b>		<b>7</b>			<b>7</b>			<b>7</b>			<b>7</b>			<b>7</b>			

Пример: Число  $63_8$  эквивалентно  $51_{10}$  ( $6 \cdot 8^1 + 3 \cdot 8^0$  или  $4810 + 310$ ).

## Двоично-десятичная система счисления - BCD Numbering System

BCD (Binary Coded Decimal – Двоично Кодированная Десятичная) система подобна Octal и Hexadecimal системам счисления. Она также основана на двоично-кодированных данных (см. таблицу). В ее основании число 10. Как видите, между BCD и Binary большое различие.

		<b>BCD</b>															
Номер бита		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень		$10^3$				$10^2$				$10^1$				$10^0$			
Значение бита		8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
<b>Максим. значение</b>		<b>9</b>				<b>9</b>				<b>9</b>				<b>9</b>			

Достоинство BCD чисел в том, что их можно «читать» как, т.е.:

867 BCD - означает - 867 десятичное. Не нужны преобразования.

## Реальные числа в формате с плавающей запятой - Real / Floating Point Numbering System

Под терминами **Реальные числа** и **числа с плавающей запятой** понимаются числа в формате соответствующем IEEE-754. Большинство ПЛК используют 32-х битовое представление чисел с плавающей точкой или реальных (Real).

Формула преобразования следующая:

$$N = 1.M \times 2^{(E-127)}$$

N: Число, которое надо представить в формате с плавающей запятой

M: Мантисса

E: Экспонента

		<b>Реальные числа с плавающей запятой - Real (Floating-Point 32)</b>																
Номер бита		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
		Знак	Экспонента								Мантисса							
Номер бита		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		Мантисса																

Вычисления с числами в Реальном формате сложны. Просто и наглядно показать работу с ними трудно. Кроме того, у этого формата есть ряд неотъемлемых внутренних ошибок. В большинстве приложений для ПЛК Вам надо быть аккуратным на 100%, чтобы не было проблем.

## Представление чисел в дополнительном двоичном коде

Двоичные числа, используемые на практике, могут иметь знак. Для представления чисел со знаком используются **прямой, обратный и дополнительный** коды.

Прямой код предполагает наличие знакового (старшего) разряда.

Обратный код двоичного числа является промежуточным между прямым и дополнительным кодом.

Правило перевода числа из прямого кода в дополнительный код:

- дополнительный код положительного числа совпадает с прямым кодом;
- для перевода отрицательного числа необходимо:

– инвертировать все биты, т. е. заменить все “1” на “0” и все “0” на “1” во всех разрядах, кроме знакового;

– к полученному числу прибавить единицу.

**Пример:**

Исходное число со старшим знаковым разрядом (прямой код)	$0110_2 = 4 + 2 = 6_{10}$ (положительное)	$11110_2 = -16 + 8 + 4 + 2 = -14_{10}$ (отрицательное)
1. Инверсия	$0110_2$ (нет действия)	$10001_2$
2. Добавление единицы (результат в дополнительном коде)	$0110_2$ (нет действия)	$10010_2 = -14_{10}$

**Особенности дополнительного кода:**

- при использовании дополнительного кода операции сложения и вычитания над числами со знаками производятся, как над положительными беззнаковыми числами, т.е. операции со знаковыми разрядами производятся так же, как со значащими разрядами;
- правило перевода числа из дополнительного кода в прямой код совпадает с приведенным выше правилом перевода из прямого в дополнительный код.

## **Арифметические операции над числами в дополнительном коде**

Как уже было отмечено, при использовании дополнительного кода операции сложения и вычитания чисел производятся, как над целыми беззнаковыми числами по правилам обычной арифметики. При этом знаковые разряды участвуют в операциях наравне с числовыми разрядами. В результате получается алгебраическая сумма в дополнительном коде. Это позволяет производить операции, не обращая внимания на знаки чисел, а также использовать одни и те же устройства для операций как над числами со знаком, так и над беззнаковыми числами.

Пример. Операция сложения  $6 + (-14) = -8$

$$\begin{array}{r}
 (6)_{\text{ДК}} = 000110 \\
 (-14)_{\text{ДК}} = 110010 \\
 (s)_{\text{ДК}} = 111000 \\
 (s)_{\text{ПК}} = 111000 \longrightarrow 100111 + 1 \longrightarrow 101000 = (-8)_{10}
 \end{array}
 \qquad
 \begin{array}{r}
 + \quad 000110 \\
 \quad 110010 \\
 \hline
 111000
 \end{array}$$

Пример. Операция вычитания  $6 - (-14) = 20$

$$\begin{array}{r}
 000110 \\
 - \quad 110010 \\
 \hline
 010100 \longrightarrow 20_{10}
 \end{array}$$

При выполнении операции вычитания производится заем из мнимых (отсутствующих) разрядов. Результат операции вычитания получился положительный, и дополнительный код результата совпадает с прямым.

Алгоритм выполнения операции умножения над числами в дополнительном коде отличается от алгоритма умножения чисел в прямом коде. Один из возможных алгоритмов (алгоритм умножения, начиная с младших разрядов множителя, со сдвигом сумм частичных произведений вправо) выглядит следующим образом:

1. Исходное значение сумм частичных произведений принимается равным 0.
2. Если анализируемая цифра множителя равна 1, то к сумме частичных произведений прибавляется множимое. Если цифра равна 0, прибавление не производится.
3. Сумма частичных произведений сдвигается на один разряд вправо, при этом, если сумма отрицательна, осуществляется модифицированный сдвиг, т. е. сдвиг с расширением знака.
4. Последовательно выполняются пп. 2 и 3 для всех цифровых разрядов множителя, начиная с младшего.
5. Если множитель – положительное число, полученный результат представляет произведение в дополнительном коде. Если множитель отрицателен, то для получения произведения к результату прибавляется множимое с обратным знаком в дополнительном коде.

Пример:  $(-4) * (-3) = 12$

$$(-4)_{\text{ДК}} = 1100$$

$$(-3)_{\text{ДК}} = 1101$$

	*	1100
		1101
		<hr/>
сумма		1100
модифицированный сдвиг		11100
		<hr/>
		0000
сумма		11100
модифицированный сдвиг		111100
		<hr/>
		1100
сумма		101100
модифицированный сдвиг		1101100
		<hr/>
		1100
сумма		1001100
модифицированный сдвиг		11001100
		<hr/>
добавление +4		0100
сумма = результат		<hr/> 00001100

При выполнении действий сложения частичных сумм переносы влево за пределы разрядной сетки игнорируются. Произведение имеет количество разрядов, равное сумме разрядов сомножителей, и, следовательно, два старших разряда – знаковые.

## BCD / Binary / Decimal / Hex / Octal - Как различить?

<i>Binary / Decimal – Двоичные / Десятичные числа</i>																
Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Десятичное значение числа	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
<b>Макс. значение</b>	<b>65535<sub>10</sub></b>															
<i>Шестнадцатеричное число - Hexadecimal</i>																
Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень	$16^3$				$16^2$				$16^1$				$16^0$			
Значение бита	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
<b>Макс. значение</b>	<b>F</b>				<b>F</b>				<b>F</b>				<b>F</b>			
<i>Восьмеричное число - Octal</i>																
Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень	$8^5$	$8^3$			$8^2$			$8^1$			$8^0$					
Значение бита	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1
<b>Макс. значение</b>	<b>1</b>	<b>7</b>			<b>7</b>			<b>7</b>			<b>7</b>			<b>7</b>		
<i>Двоично-кодированное десятичное число - BCD</i>																
Номер бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Степень	$10^3$				$10^2$				$10^1$				$10^0$			
Значение бита	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
<b>Максим. значение</b>	<b>9</b>				<b>9</b>				<b>9</b>				<b>9</b>			

В ПЛК используются формат данных BCD, формат адресов Octal. В других случаях применяются форматы Binary и Real. Хотя все эти данные хранятся одинаково в виде нулей и единиц, ПЛК интерпретирует их по-разному.

Путаница типов является общей проблемой при использовании операторского интерфейса. Так как формат BCD является внутренним форматом ПЛК, многие предполагают, что он взаимозаменяемый с форматом Binary (Unsigned Integer). В таблице справа показаны сходство и отличия чисел в формате BCD и Binary. Как видно из таблицы, форматы BCD и Binary используют одни и те же наборы бит (bit pattern) для чисел до 10 (десятичного), но далее все меняется резко. Битовое представление BCD числа 10 соответствует битовому представлению числа 16 в Binary формате. Чем больше число, тем больше разница.

### BCD - Binary

Decimal	BCD	Binary
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	0001 0000	0000 1010
11	0001 0001	0000 1011
.....		
4096	0100 0000 1001 0110	0001 0000 0000 0000

### BCD - Hexadecimal

Как видно из общей таблицы, форматы BCD и Hexadecimal похожи, хотя максимальное число в каждой тетраде (9 - BCD, F - Hexadecimal). Это позволяет использовать одни и те же способы представления человеку, но если тип данных не указан, Вы не можете достоверно установить тип данных (если, конечно, в числе не использованы буквы A-F).

### Целые числа со знаком или без знака – Sign / Unsigned Integers

Теперь рассмотрим, как обстоят дела при работе с отрицательными числами и форматами чисел со знаком.

Формат BCD не может быть использован для работы с отрицательными числами.

Для указания того, что число отрицательное мы должны использовать специальный бит.

Обычно это MSB(most significant bit) – наиболее значимый бит. Для 16-битовых чисел мы будем иметь диапазон представления чисел от -32767 до 32767.

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSB															LSB

Такая форма представления отрицательного числа называется Magnitude plus Sign (Значение плюс Знак).

Magnitude Plus Sign	
Decimal	Binary
100	0000 0000 0110 0100
-100	1000 0000 0110 0100

Вторая форма представления отрицательных чисел – это Two's Complement (Двоичный дополнительный код). Такой код получают, инвертируя двоичное число и добавляя единицу. О работе с числами в дополнительном коде было написано ранее.

Two's Complement	
Decimal	Binary
100	0000 0000 0110 0100
-100	1111 1111 1001 1100

## **Типы данных в контроллерах *DirectLogic* и других изделиях *Automationdirect*.**

В семействе контроллеров DirectLogic plc family для работы с модулями ввода/вывода используется восьмеричная система счисления (Octal).

Вся информация в V-памяти хранится в BCD формате, если формат специально не изменен программистом. Это значит, что все математические операции должны производиться при помощи BCD функций. В некоторых моделях ПЛК Вы можете использовать формы: You Binary, Real или Double-Word BCD. Для изменения типа данных (data types), необходимо использовать Функциональные блоки (function box). Для перевода из BCD в Binary используют функциональный блок BIN. Из BCD в Real Вам сначала надо использовать блок BIN, а затем блок BTOR. Вы не можете сложить число в формате BCD или Binary с числом в формате Real, или BCD с числом в Binary и получить корректный результат. Типы данных должны совпадать.

Есть несколько моментов связанных с типами данных, на которые надо обращать внимание при работе.

1. Модули аналогового ввода могут быть настроены на выдачу результата измерений в формате Binary или BCD, это не надо забывать.
2. Вторая область, где не все в BCD, это ПИД-регулятор (PID). Почти все параметры регулятора хранятся, как двоичные числа (Binary). В ПИД-регуляторе используется представление отрицательных двоичных чисел в виде числа со знаком (Magnitude Plus Sign), тогда как в других математических функциях ПЛК используется представление отрицательных чисел в дополнительном коде (Two's Complement).
3. Будьте внимательны при работе с окнами просмотра данных (data view). Как ни странно это звучит, но все числа: Binary, Hex, и Decimal хранятся в ПЛК одинаково и называются все данными двоичного (Binary) формата. Различить Вы их можете в режиме просмотра данных программы - data view. Удостоверьтесь в том, что Вы выбрали соответствующий формат представления данных в ниспадающем меню.

И последнее. Вы заметили, что формат BCD в data view обозначается - BCD/Hex формат.

Ранее было показано, что эти числа одной длины и различаются только использованием букв A - F, поэтому используется одно окно представления.

## **Операторские панели EZ-Touch/EZ-Text**

В панелях оператора EZ-Touch и EZ-Text:

- Формат 16-bit BCD обозначается, как BCD\_INT\_16.
- Двоичный (Binary) формат, как Unsigned\_Int\_16 или Signed\_Int\_16 в зависимости от того - может ли значение быть отрицательным.
- Формат чисел с плавающей запятой (Real) обозначается, как Floating\_Point\_32.