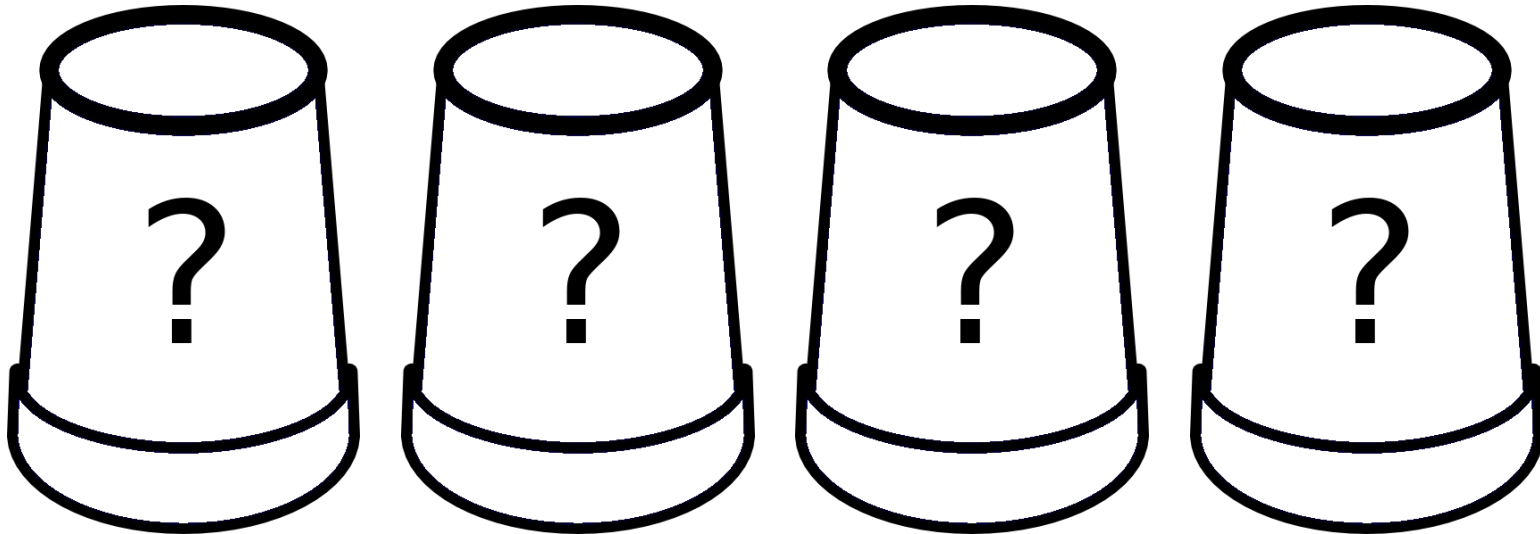


Программирование микроконтроллеров 1

(информация, процессоры,
языки программирования)

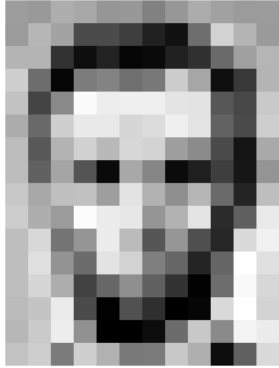
Информация и её количество

Количество информации



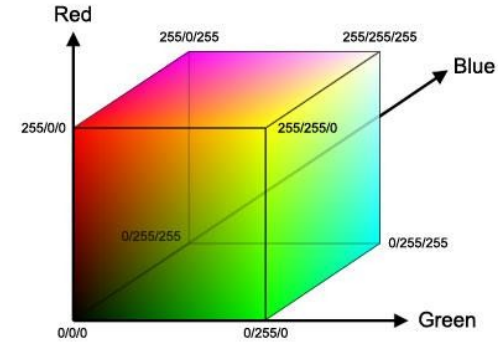
Игра в напёрстки. Известно, что горошина под каким-то из 4 напёрстков. Информация, как мера устранения неопределённости показывает какой вариант из 4 возможных верный. Получение сообщения о “координатах” горошины уменьшит неопределённость в 4 раза.

Виды информации



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	67	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	35	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	67	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	35	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

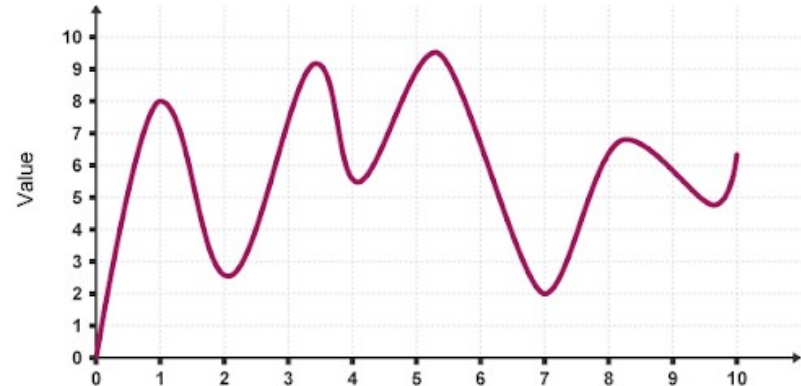


Кодировка цвета

Кодировка изображений

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Кодировка текста



Кодировка звука

Виды информации



Прогноз погоды

Championship subscribe

	Team	MP	W	D	L	F	A	D	P	Last 5 matches
13	Nottingham Forest	11	5	0	6	15	19	-4	15	W L L L W
14	▲ Queens Park Ra...	12	3	5	4	15	16	-1	14	D L D D L
3	Sheffield United	12	8	0	4	16	10	+6	24	W L W W L
4	Bristol City	12	5	6	1	20	11	+9	21	D W W D W
5	▲ Preston North End	12	5	6	1	15	8	+7	21	D D W D W
6	▼ Leeds United	12	6	2	4	18	12	+6	20	L L L W L
7	Aston Villa	12	5	4	3	17	12	+5	19	L W W W W
8	▲ Norwich City	12	5	4	3	12	14	-2	19	D W W D W
9	▲ Fulham	12	4	6	2	16	12	+4	18	D W W D L
10	▼ Ipswich Town	11	6	0	5	20	17	+3	18	L L W L W
11	Middlesbrough	12	4	5	3	15	11	+4	17	D D L D W
12	Sheffield Wedn...	12	4	4	4	16	14	+2	16	L W L L D

Различные таблицы

«Сложные» виды информации



Вкус



Тактильные ощущения



Запах

Количество информации

- Получение информации — выбор одного сообщения из конечного множества из N равновероятных сообщений. Количество информации I , содержащееся в выбранном сообщении это двоичный логарифм N .
- Для оценки количества информации для равновероятных сообщений используется формула Хартли

$$I = \log_2 N$$

- Для неравновероятных сообщений используется формула Шеннона

$$I = -(p_0 \log_2 p_0 + p_1 \log_2 p_1 + \dots + p_N \log_2 p_N)$$

Количество информации



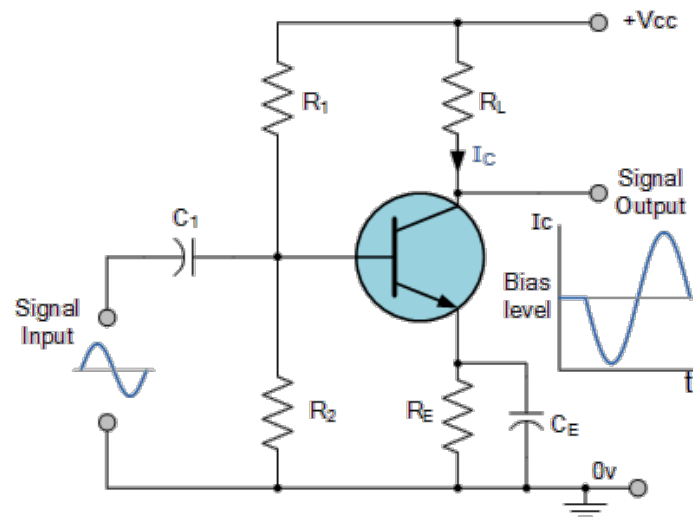
- Один бит это такое количество информации, которое уменьшает имеющуюся неопределённость в два раза. $\log_2 2 = 1$ бит

Аналоговая техника

- Аналоговая техника - техника для работы с аналоговыми сигналами. Аналоговый сигнал представляет собой непрерывную функцию, с неограниченным числом значений в различные моменты времени.

Преимущества:

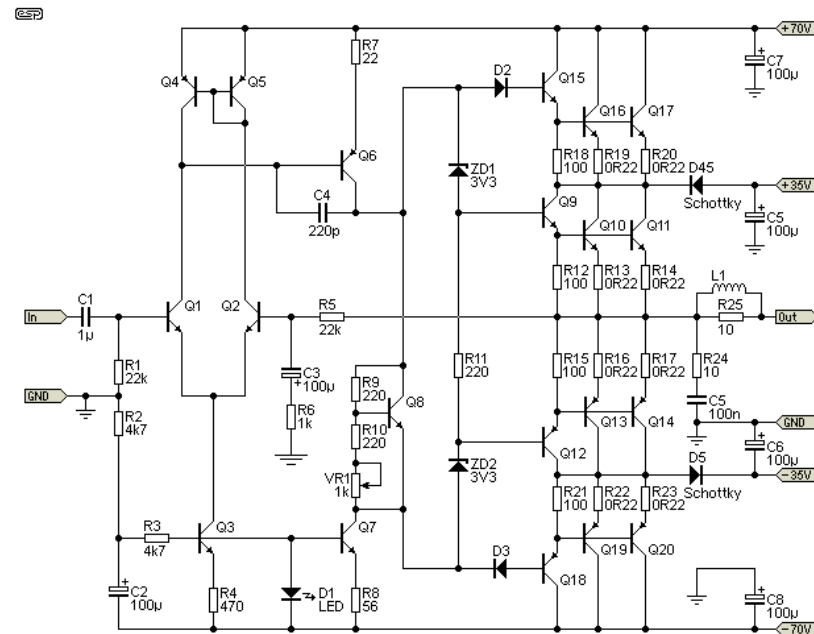
- Высокое быстродействие
- Непрерывность
- Простота
- ...



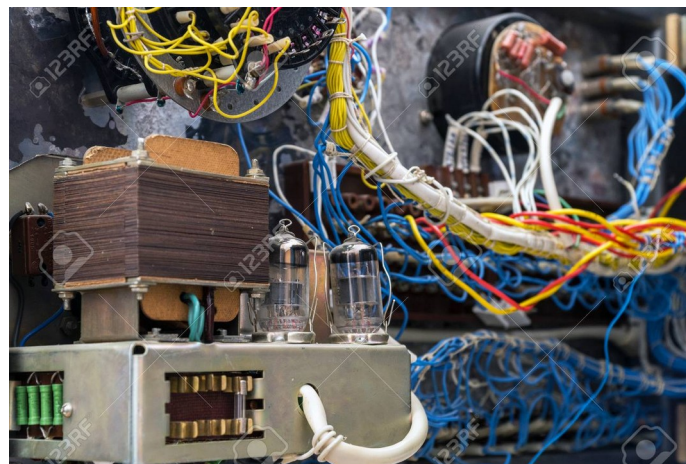
Аналоговая техника

Недостатки:

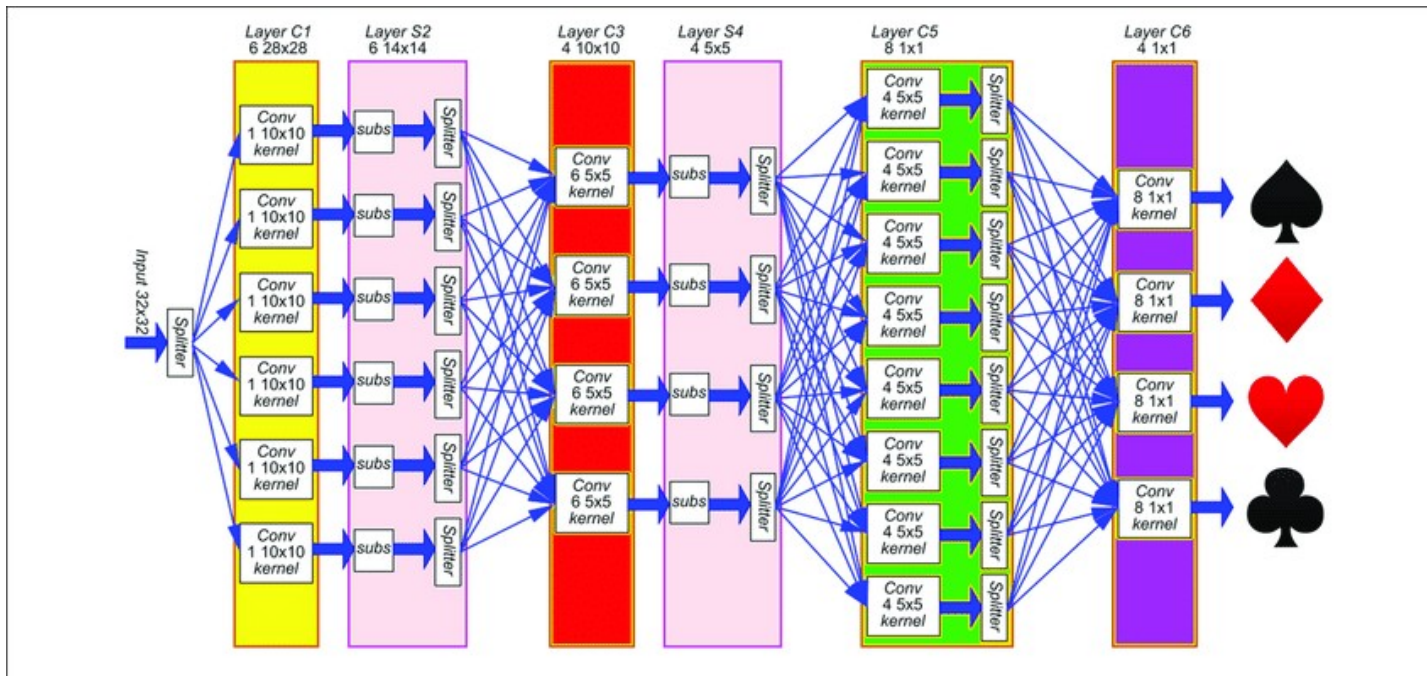
- Значительная подверженность влиянию внешних факторов
- Искажение информации
- Сравнительно низкий КПД
- Ограничения на значения физических величин
- Сравнительно большие габариты и масса
- Привязанность к физическому “носителю”



Аналоговая техника

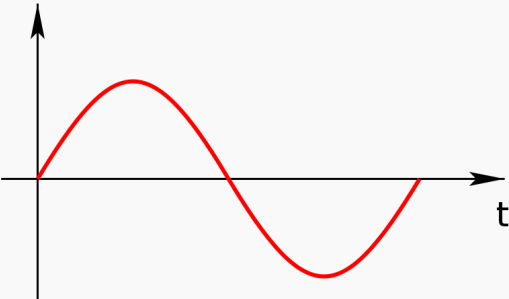
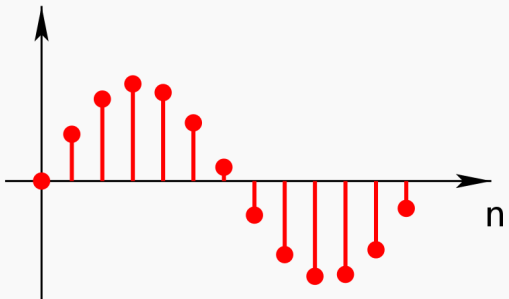
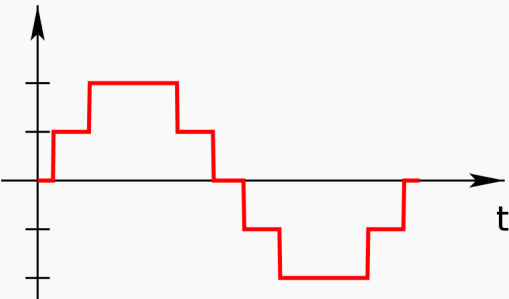
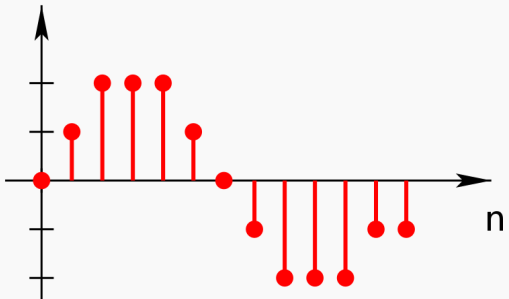


Аналоговая техника



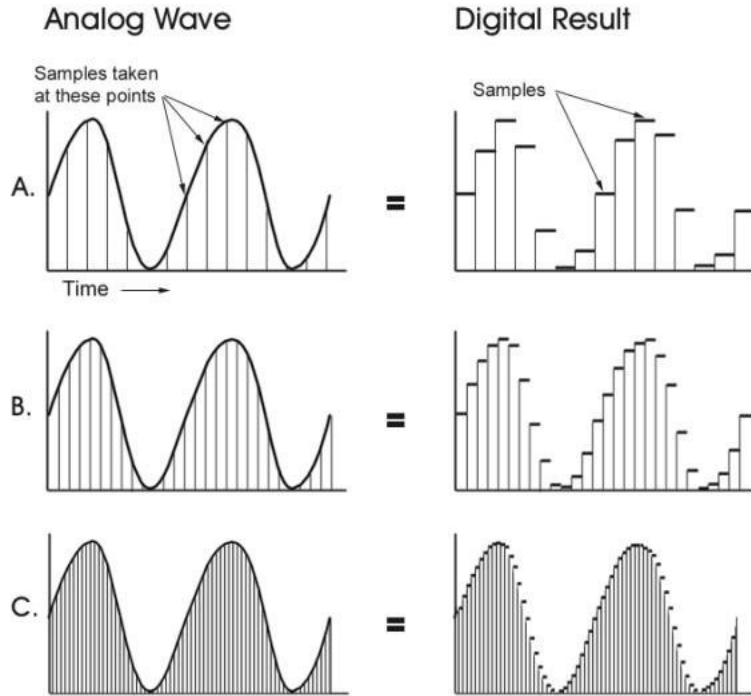
Аналоговые нейросети?

Дискретная и цифровая техника

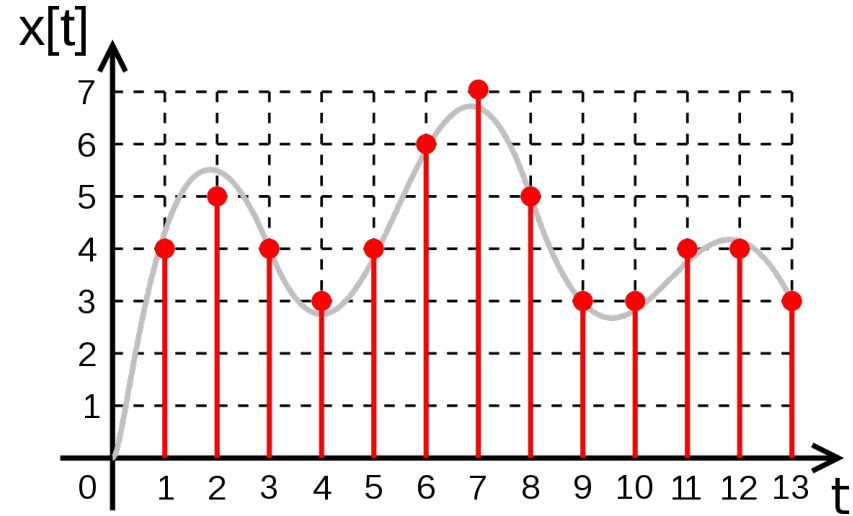
	Непрерывные	Дискретные
Неквантованные	 <p>Аналоговый сигнал</p>	
Квантованные	 <p>Цифровой сигнал</p>	 <p>Дискретный (квантованный)</p>

Дискретная и цифровая техника

Increasing Sample Rates

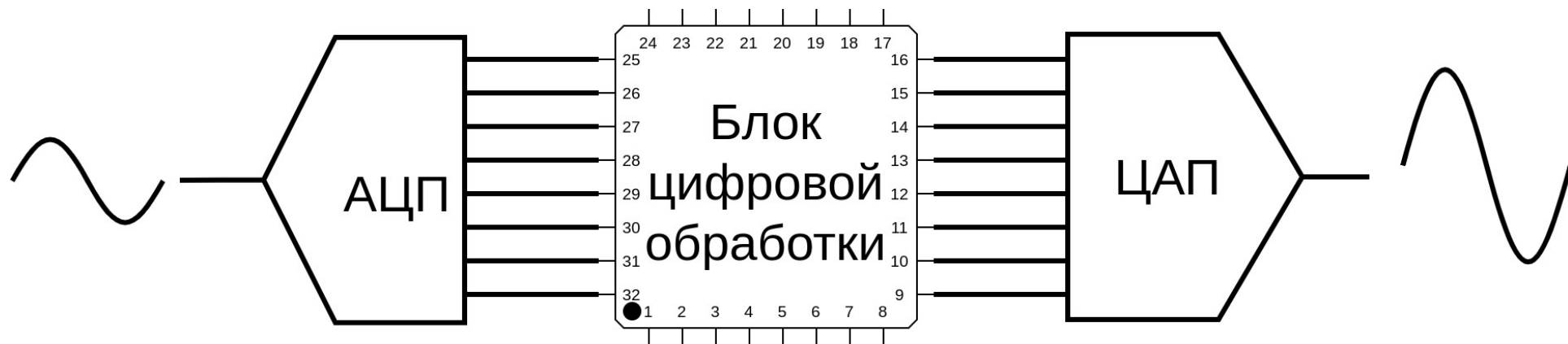


Процесс оцифровки звука



Дискретизация и по времени и по значению

Дискретная и цифровая техника



Цифровая обработка звука

Преимущества и недостатки цифровой техники

Преимущества цифровой техники:

- Предсказуемая работа во всем допустимом диапазоне внешних условий
- Независимость от величин и представления обрабатываемых значений
- Гибкость и универсальность
- Воспроизводимость

Преимущества и недостатки цифровой техники

Недостатки:

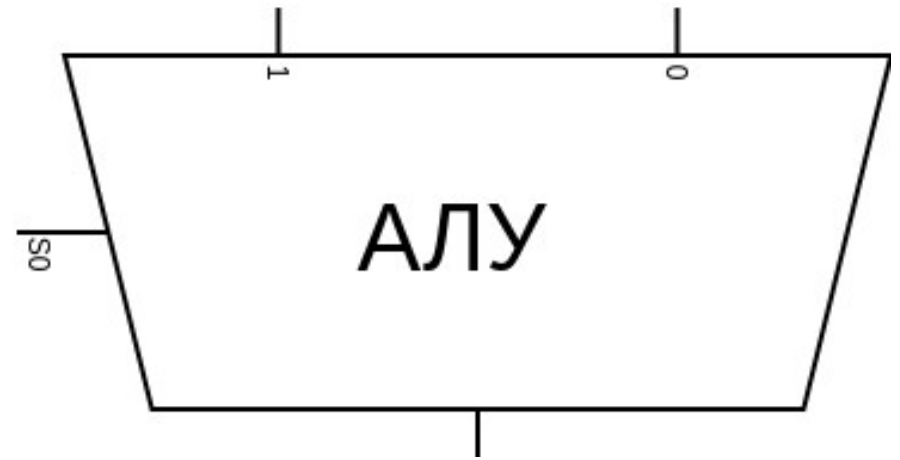
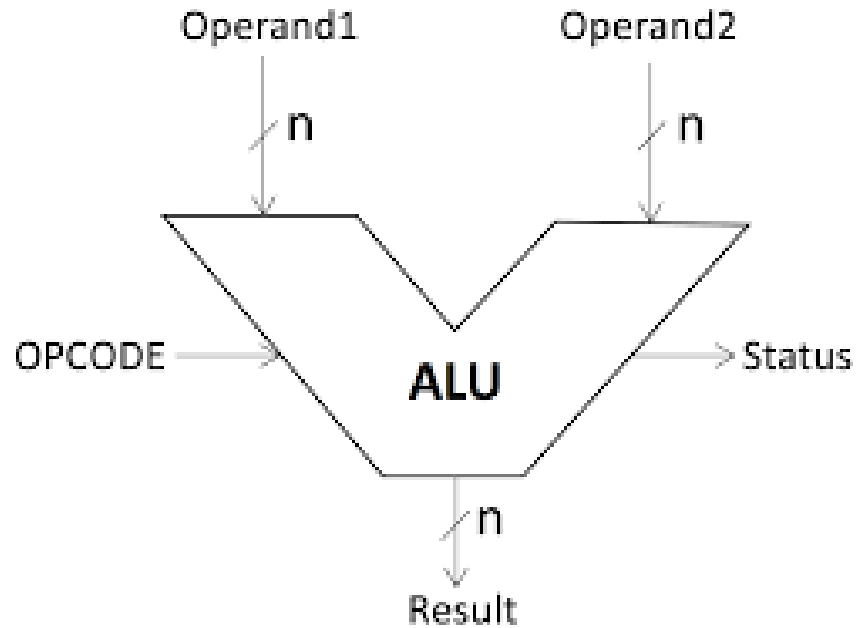
- Искажение информации (дискретизация)
- Сравнительно низкое быстродействие
- Сложность
- . . .

Микропроцессорная техника

Если любая информация представима в виде чисел, то машина, способная производить определённый набор операций с числами способна обрабатывать любую информацию.

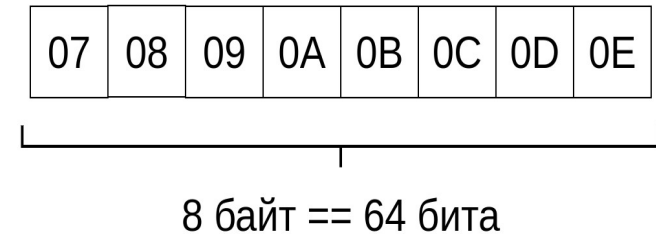
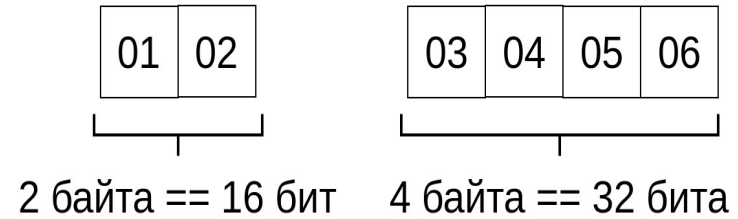
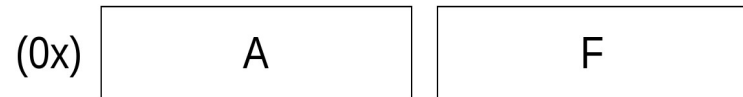
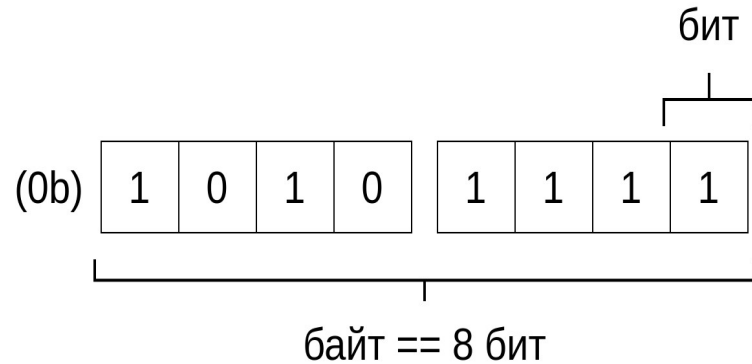
Процессоры

Процессор и его начинка (АЛУ)



Арифметико-логическое устройство

Процессор и его начинка (биты и байты)



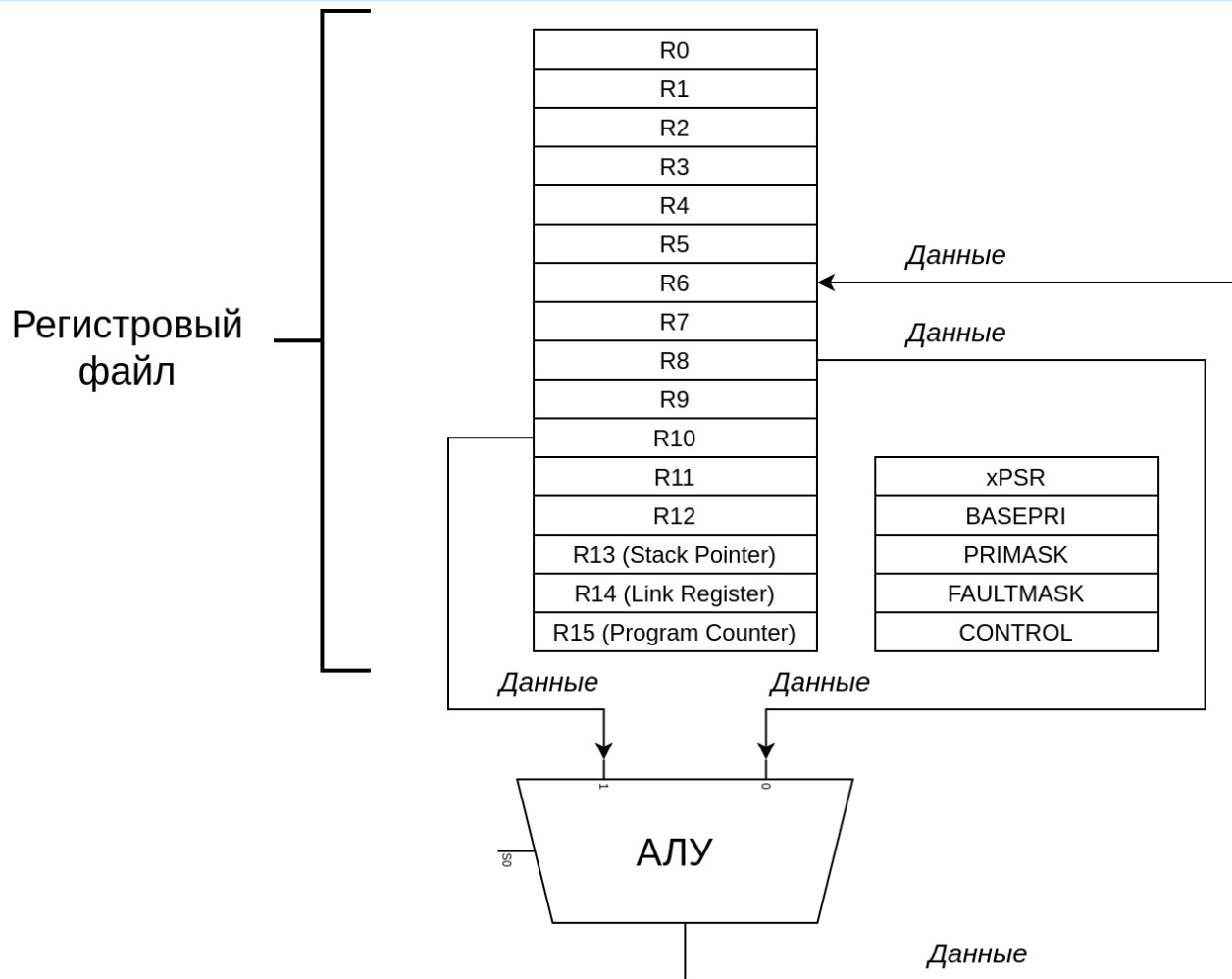
$$2^8 = 256$$

$$2^{16} = 65\,536$$

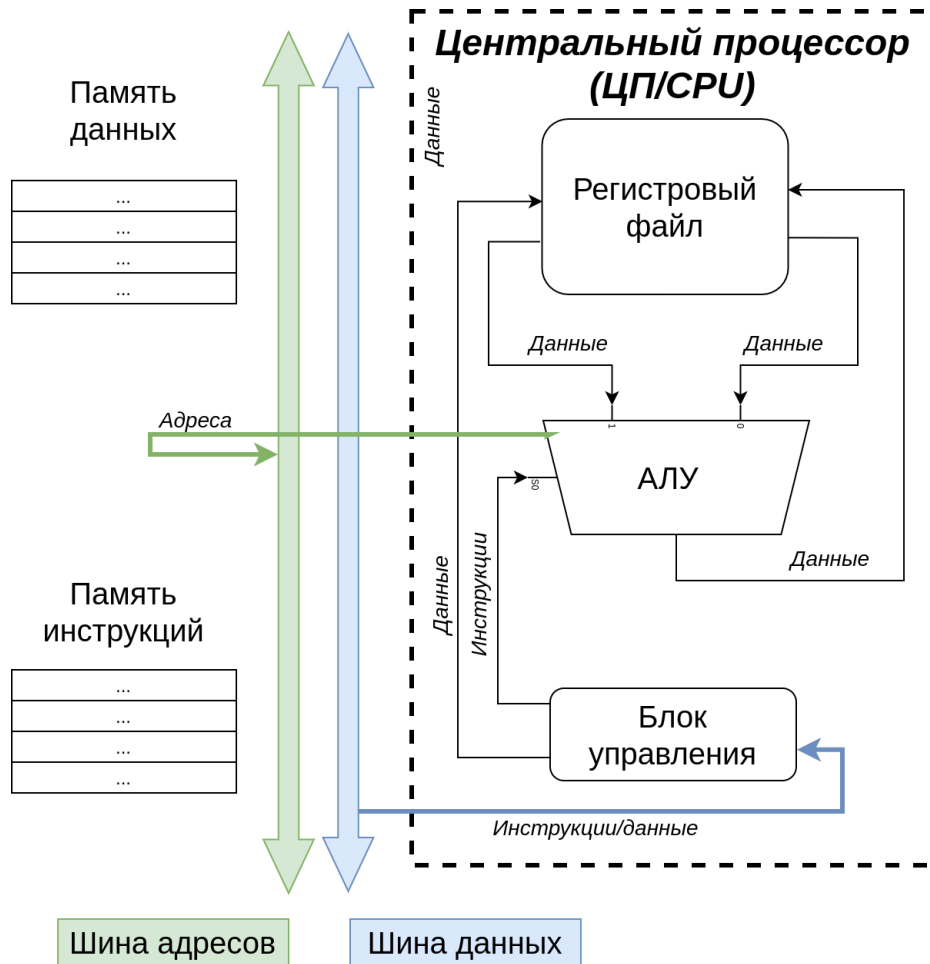
$$2^{32} = 4\,294\,967\,296$$

$$2^{64} = 18\,446\,744\,073\,709\,551\,616$$

Процессор и его начинка (регистровый файл)



Процессор и его начинка (шины)

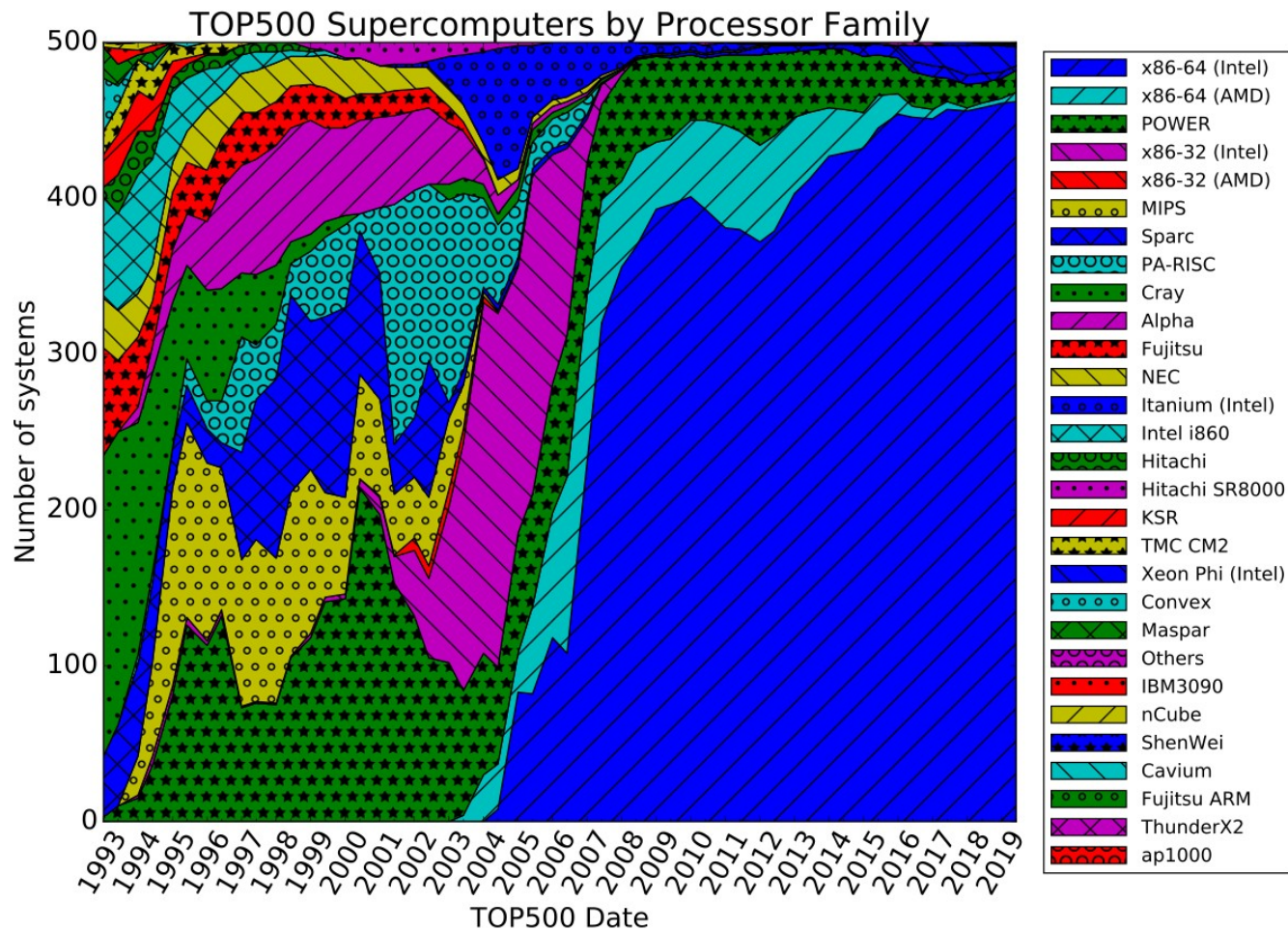


Набор инструкций (Thumb-2)

<https://documentation-service.arm.com/static/5ed66080ca06a95ce53f932d?token=>

- **ADD** - сложение
- **SUB** - вычитание
- **MUL** - умножение
- **RM** - деление
- **MOV** - перемещение (копирование)
- ***SR** - побитовое смещение (*2;/2)
- **логика** - логические И, ИЛИ, НЕ, и т.п.
- **LDR** - загрузка из памяти
- **STR** - выгрузка в память

Архитектуры процессоров



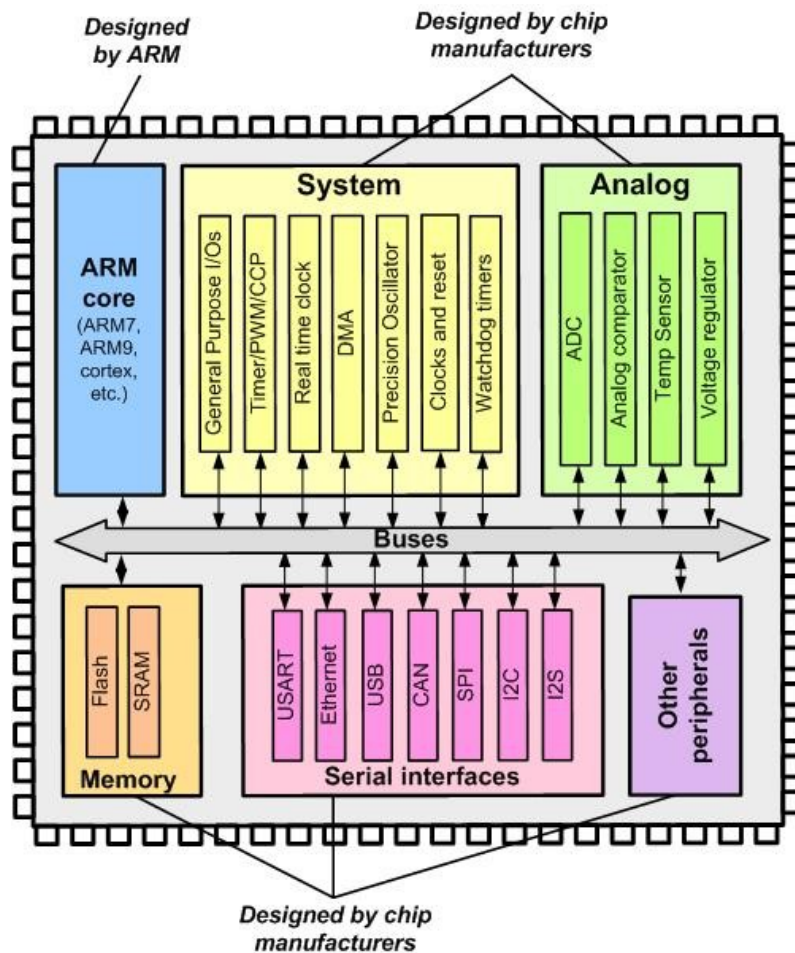
Архитектуры процессоров



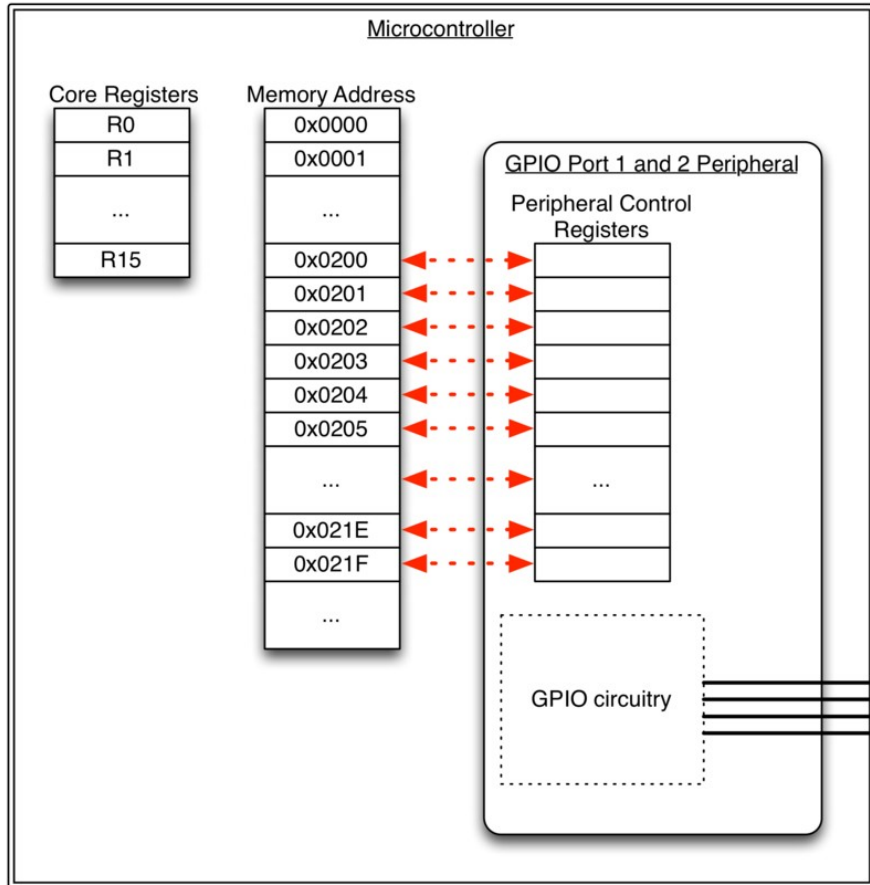
Семейство процессоров ARM



Микроконтроллеры

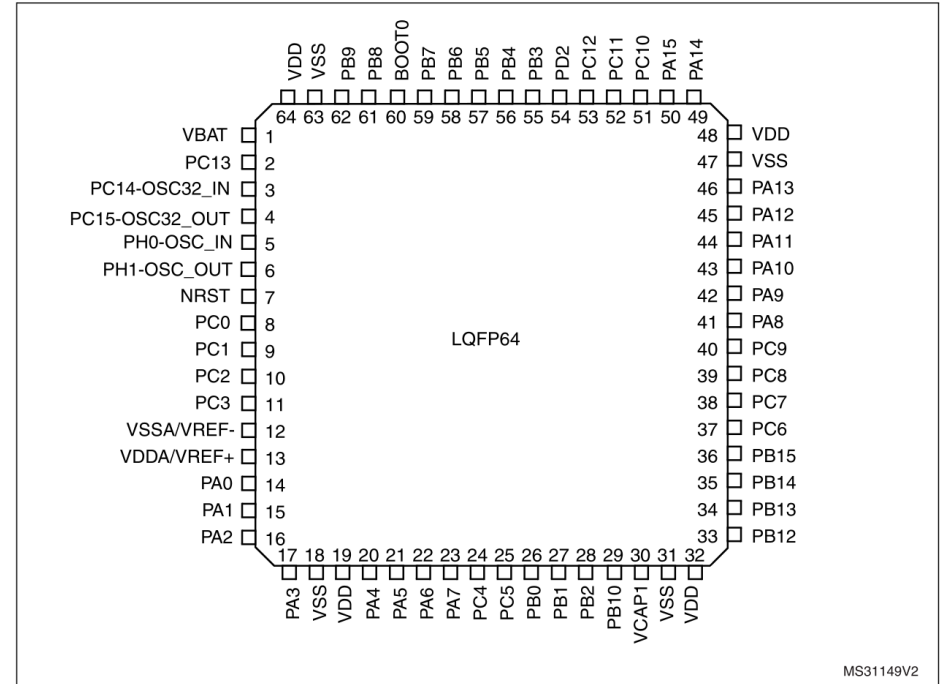


Управление периферией микроконтроллера

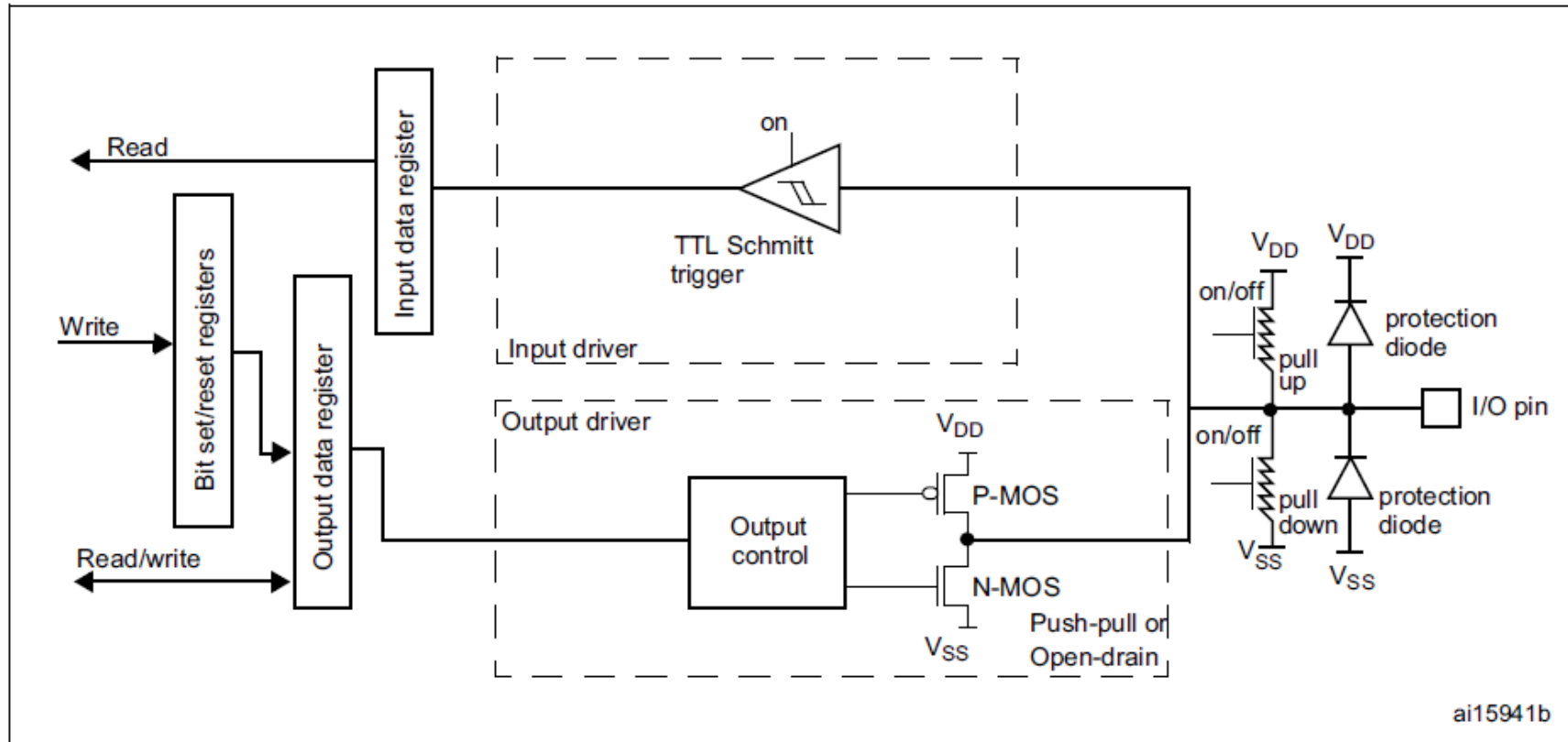


GPIO pins
outside of chip
(there are
16 pins in GPIO
Ports 1 and 2)

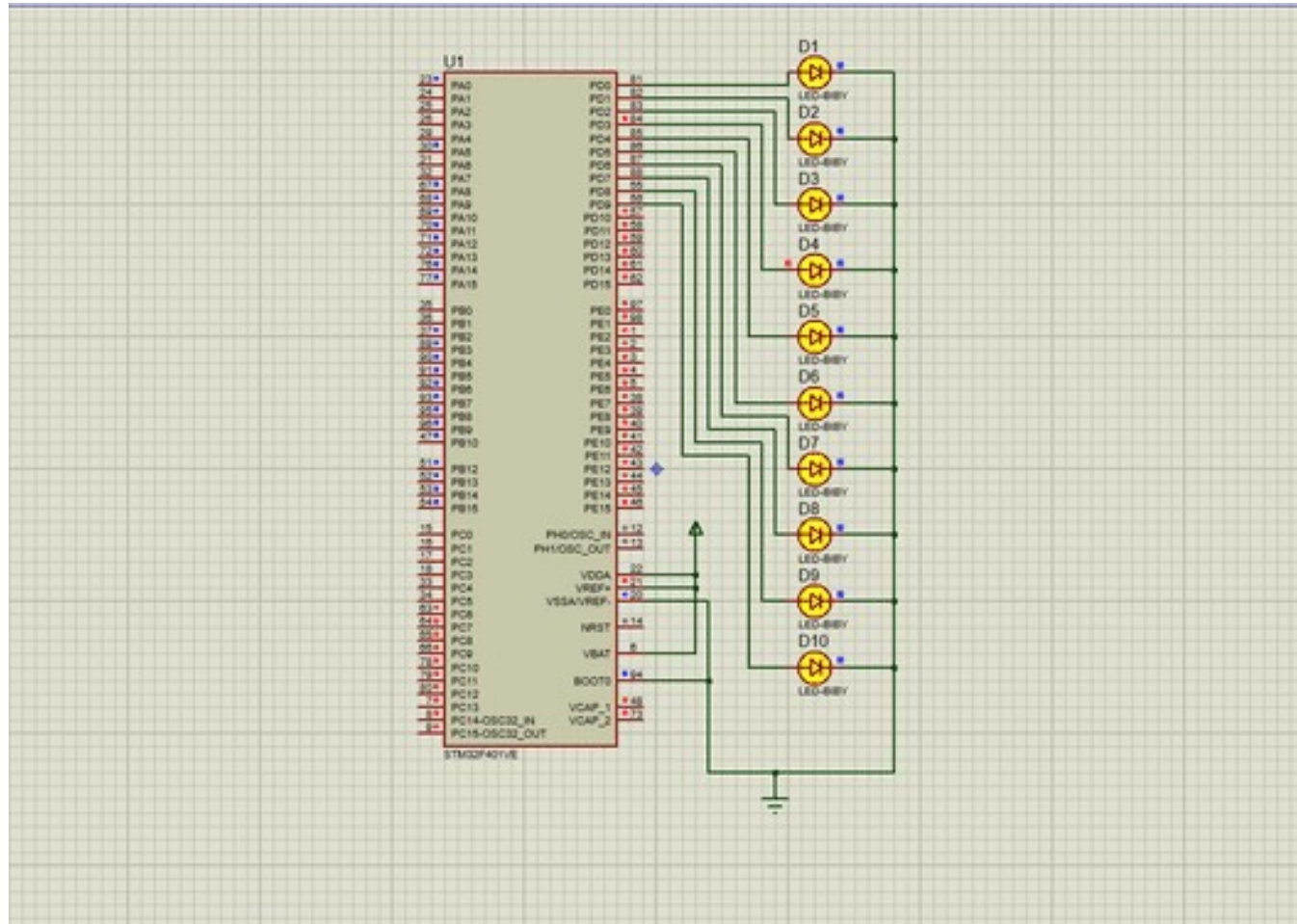
Figure 12. STM32F401xD/xE LQFP64 pinout



Порты ввода/вывода



Порты ввода/вывода



Языки программирования

Ассемблер

Язык ассемблера - Язык **очень** низкого уровня (ниже только рукописные машинные коды).

Преимущества:

- Полный контроль над аппаратурой
- Возможность написания очень быстрых или очень компактных программ
- Возможность использования узкоспециальных инструкций

Недостатки:

- Совершенно не похож на человеческий язык
- Большие тексты программ
- Сложности со структурированием кода
- Платформозависимость

Ассемблер

Область применения:

- Операционные системы (в частности диспетчеры)
- Там, где нужна невероятная скорость (драйвера, игры, обработка видео и прочее)
- Там, где нужен невероятно компактный код (загрузчики)
- Там, где нужно дотянуться вооон до того регистра (экзотическое железо)

Hello World на ассемблере

- <https://gist.github.com/Snork2/c3643c1eeb40737525af4fc138bcadcc>

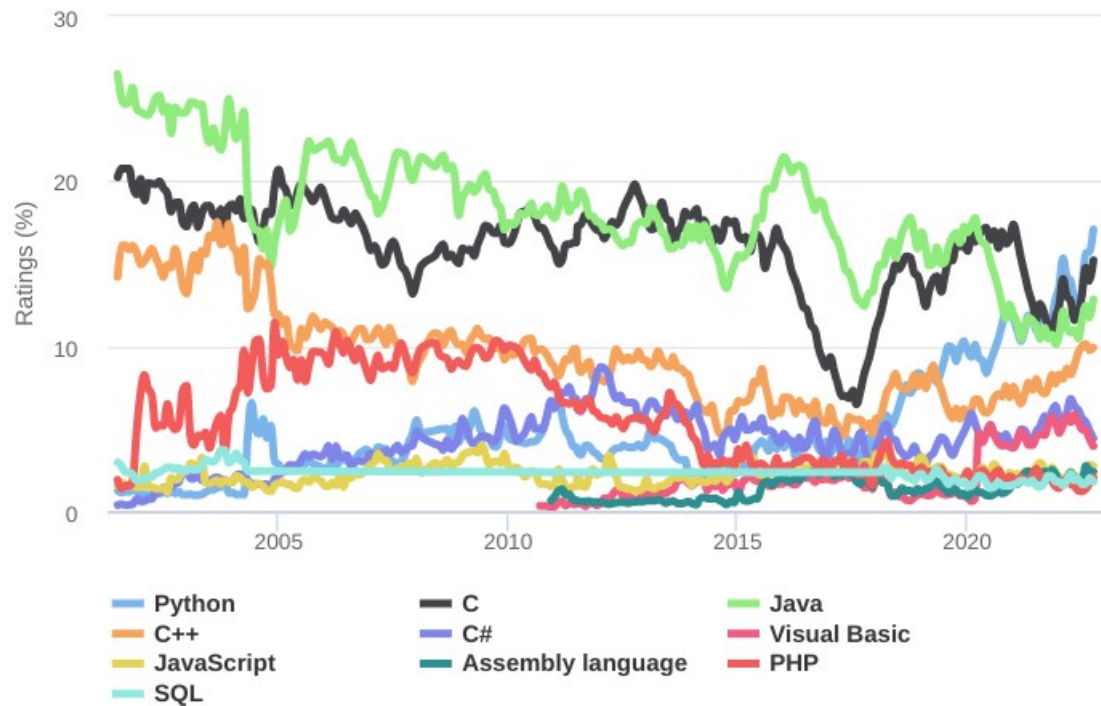
Язык Си

- С (рус. Си) — компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969 — 1973 годах сотрудником Bell Labs Деннисом Ритчи как развитие языка Би. Первоначально был разработан для реализации операционной системы UNIX, но, впоследствии, был перенесён на множество других платформ. Благодаря близости по скорости выполнения программ, написанных на Си, к языку ассемблера, этот язык получил широкое применение при создании системного программного обеспечения и прикладного программного обеспечения для решения широкого круга задач. Язык программирования С оказал существенное влияние на развитие индустрии программного обеспечения, а его синтаксис стал основой для таких языков программирования, как C++, C#, Java и D.

Язык Си

- Язык Си не то, чтобы очень активно, но развивается

Год	Событие
1972	Появление языка
1978	K&R «стандарт»
1989	ANSI стандарт
1999	C99
2011	C11
2017	C17
TBD (2023?)	C2x (2023?)



Специфика языка Си

Язык Си – язык низкого уровня абстракции. Это означает, что сущности которыми оперирует язык – очень близки к сущностям, которыми оперирует процессор. Это позволяет очень точно управлять ресурсами ЭВМ (а следовательно использовать их с максимальной эффективностью), но увеличивает сложность написания программ.

Области применения языка Си:

- Там, где строгие требования по быстродействию и потреблению ресурсов
- Там, где нужна возможность дотянуться вооон до того регистра
- Там, где важно, чтобы программа работала **везде**

Специфика языка Си

- Хорошо написанные на Си программы работают действительно **везде**



Специфика языка Си

Язык С очень похож по синтаксису на многие языки, так как повлиял на их развитие и был в каком-то смысле их предшественником

Язык С, как и С++ сохраняет обратную совместимость со всеми своими старыми стандартами (весь старый код, написанный согласно стандартам можно использовать без изменений с новыми компиляторами)

Специфика языка Си

- Язык С – *опасный язык*. Он не запрещает и не защищает от написания совершенной ерунды и совершения логических ошибок в программе.
- “Язык С — это инструмент, острый как бритва: с его помощью можно создать и элегантную программу, и кровавое месиво.”

Брайан Керниган

{Известны 10 преимуществ Паскаля перед Си:) Я приведу только одно, но самое важное:

На Си Вы можете написать:

```
for(;P("\n").R-;P("\ "))for(e=3DC;e-;P("_ "+(*u++/8)%2))P("| "+(*u/4)%2);
```

На Паскале Вы НЕ МОЖЕТЕ такого написать.}

Компилятор

Для преобразования исходных текстов программ, написанных на языке Си в исполнимые файлы (.exe) необходим набор программ, называемых tool chain (дословно – цепь инструментов) или на жаргоне – компилятор.

Компиляторов для языка Си **очень** много. Компилятор, как правило предназначен для конкретной целевой платформы. Платформа это совокупность операционной системы и процессора, на котором она работает. Операционная система определяет API и формат исполнимого файла, а процессор определяет набор машинных инструкций, которыми компилятор может пользоваться при создании машинного кода.

Например: Windows–x86 (win32); Windows–x86_64 (win64); Linux–x86_64; Androidarmeabi; IOS–armv7eabi

Мы будем использовать тулчейны семейства GNU COMPILER COLLECTION (GCC)

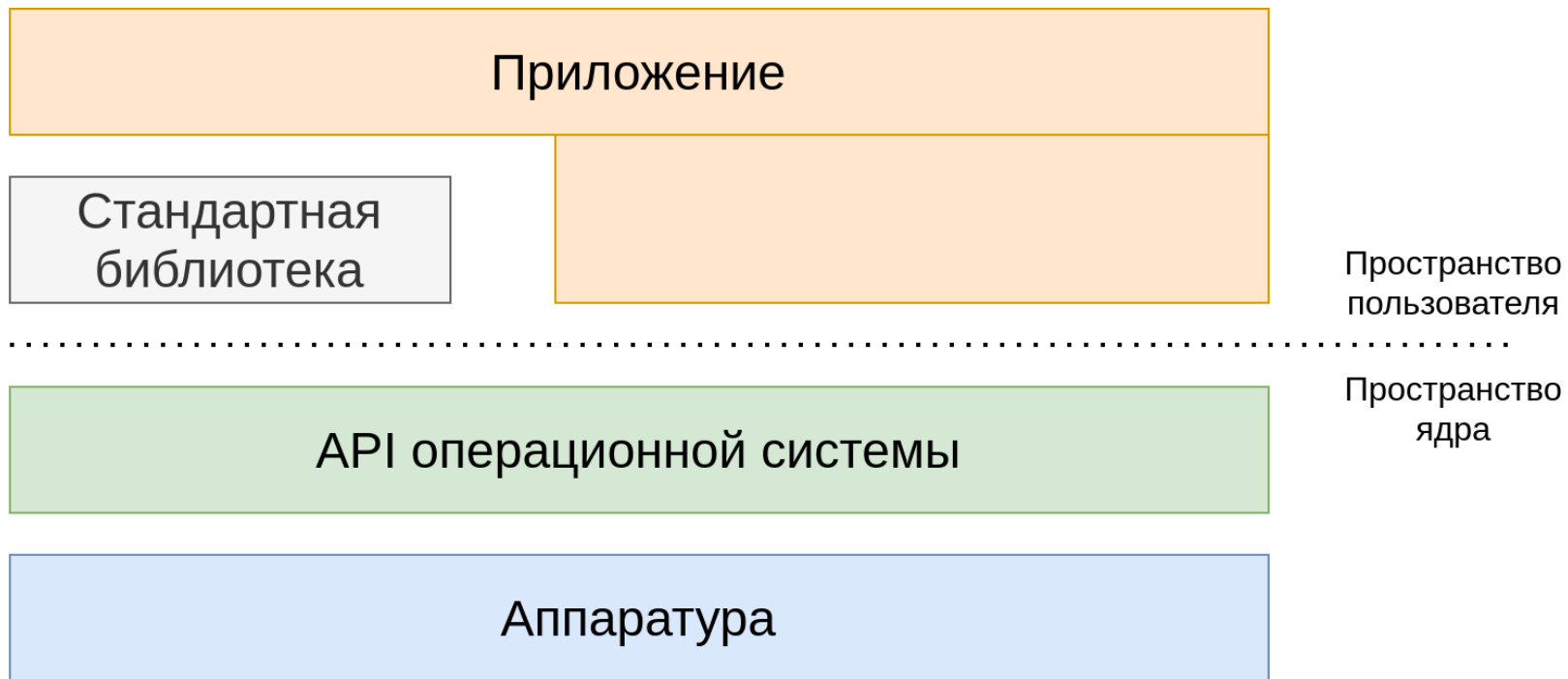
Платформа и стандартная библиотека

Платформы (аппаратные и программно–аппаратные) представляют различные программные интерфейсы для работы с ними. Поскольку мало кому хочется делать свой собственный интерфейс консоли для каждой платформы — существует стандартная библиотека, реализующая такие типовые интерфейсы.

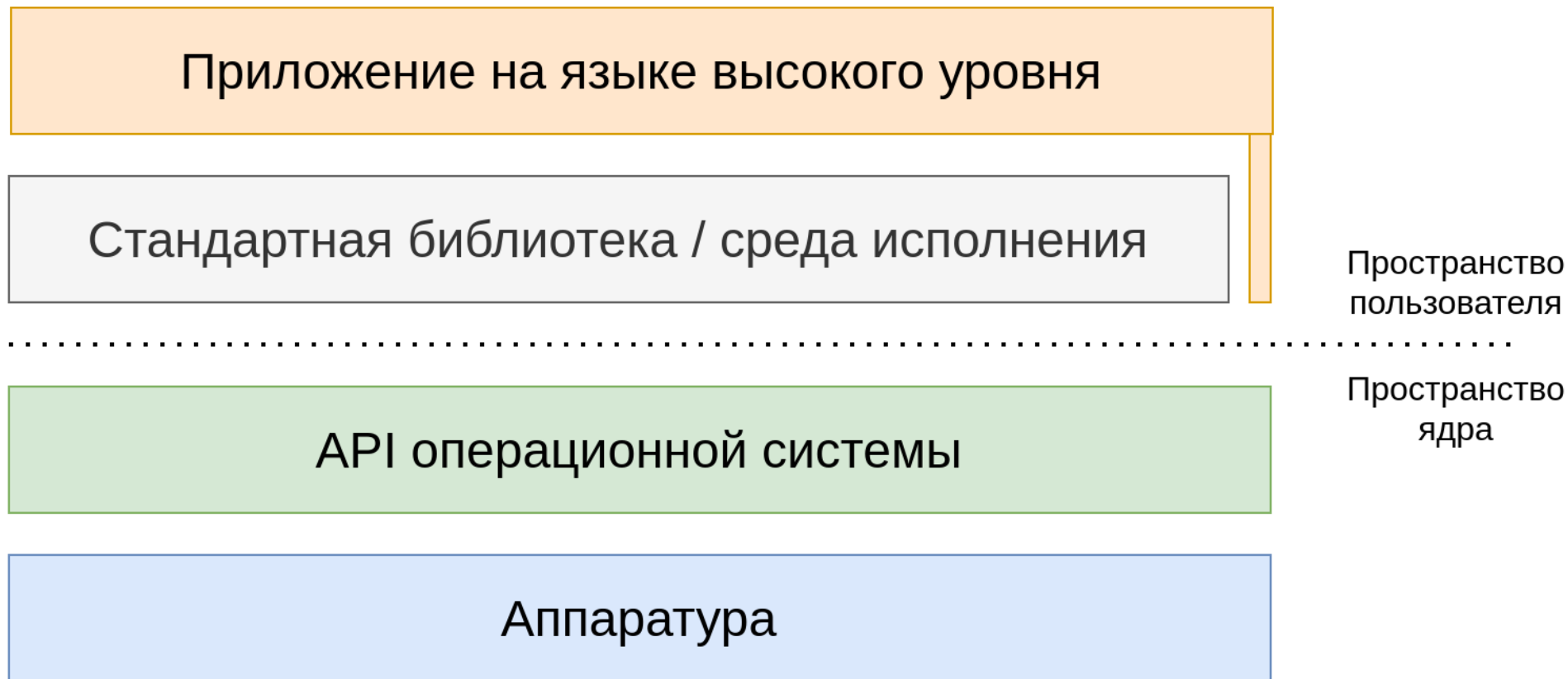
Помимо программных интерфейсов в стандартную библиотеку входят различные утилитарные сущности, такие как, функции для расчёта синуса и т. п.

Стандартная библиотека Си, как правило, является частью тулчейна и по сравнению со стандартными библиотеками других языков — довольно скудна.

Операционная система



Операционная система



Hello World на Си

- <https://gist.github.com/Snork2/688de589648e5a0f5073f0c389499be7>

Другие языки программирования для МК

C++

```
1 #include <vector>
2
3 int app_main()
4 {
5     std::vector<int>
6         vector(10);
7     vector.push_back(15);
8     for (auto & x: vector)
9         x += 10;
10
11     return 0;
12 }
```

Build Analyzer

Static Stack Analyzer

blinky-f4-cpp.elf - /blinky-f4-cpp/Debug - Nov 5, 2021, 5:29:47 PM

Memory Regions		Memory Details				
Region	Start add	End add	Size	Free	Used	Usage (%)
CCMRAM	0x100000	0x100100	64 KB	64 KB	0 B	0.00%
RAM	0x200000	0x200200	128 KB	126,33 KB	1,67 KB	1.31%
FLASH	0x080000	0x081000	1024 KB	1015,05 KB	8,95 KB	0.87%

Другие языки программирования для МК

- .NET MicroFramework
- Java ME ? .NET MicroFramework

Другие языки программирования для МК

microPascal

```
1 program LedBlinking;
2 begin
3     { * Set LED1 & LED2 to be digital outputs *}
4     GPIO_Pin_Digital_Output(_GPIO_PIN_NUM_60);
5
6     { * Turn OFF both LED1 & LED2 *}
7     GPIO_PIN60_bit := 0;
8
9     { * Toggle LEDs one by one *}
10    while true do
11        begin
12            GPIO_PIN60_bit := not GPIO_PIN60_bit;
13            Delay_ms(500);
14        end;
15    end.
```

Другие языки программирования для МК

Micropython

```
1 from machine import Pin
2
3 # create an I/O pin in output mode
4 p = Pin('X1', Pin.OUT)
5
6 # toggle the pin
7 p.high()
8 p.low()
```

Другие языки программирования для МК

low.js

```
1 let gpio = require('gpio');
2 gpio.pins[pinnumber].setType(gpio.OUTPUT);
3
4 let val = 0, dir = 0;
5 setInterval(() => {
6   val += dir ? 0.03 : -0.03;
7   if(val < 0) { val = 0; dir = 1; }
8   if(val > 1) { val = 1; dir = 0; }
9
10  gpio.pins[pinnumber].setValue(val);
11 }, 30);
```

Другие языки программирования для МК

RUST

```
// Получаем управление над аппаратными средствами
let cp = cortex_m::Peripherals::take().unwrap();
let dp = pac::Peripherals::take().unwrap();
let mut flash = dp.FLASH.constrain();
let mut rcc = dp.RCC.constrain();

let clocks = rcc.cfgr.freeze(&mut flash.acr);
let mut gpiob = dp.GPIOB.split(&mut rcc.apb2);

// Конфигурируем пин b12 как двухтактный выход.
// Регистр "crh" передается в функцию для настройки порта.
// Для пинов 0-7, необходимо передавать регистр "crl".
let mut led = gpiob.pb12.into_push_pull_output(&mut gpiob.crh);
// Конфигурируем системный таймер на запуск обновления каждую секунду.
let mut timer = Timer::syst(cp.SYST, &clocks)
.start_count_down(1.hz());
```