Data input/output
Serial Interfaces

LINUX KERNEL UNIVERCITY COURSE

MAXIM LIPCHANSKYI

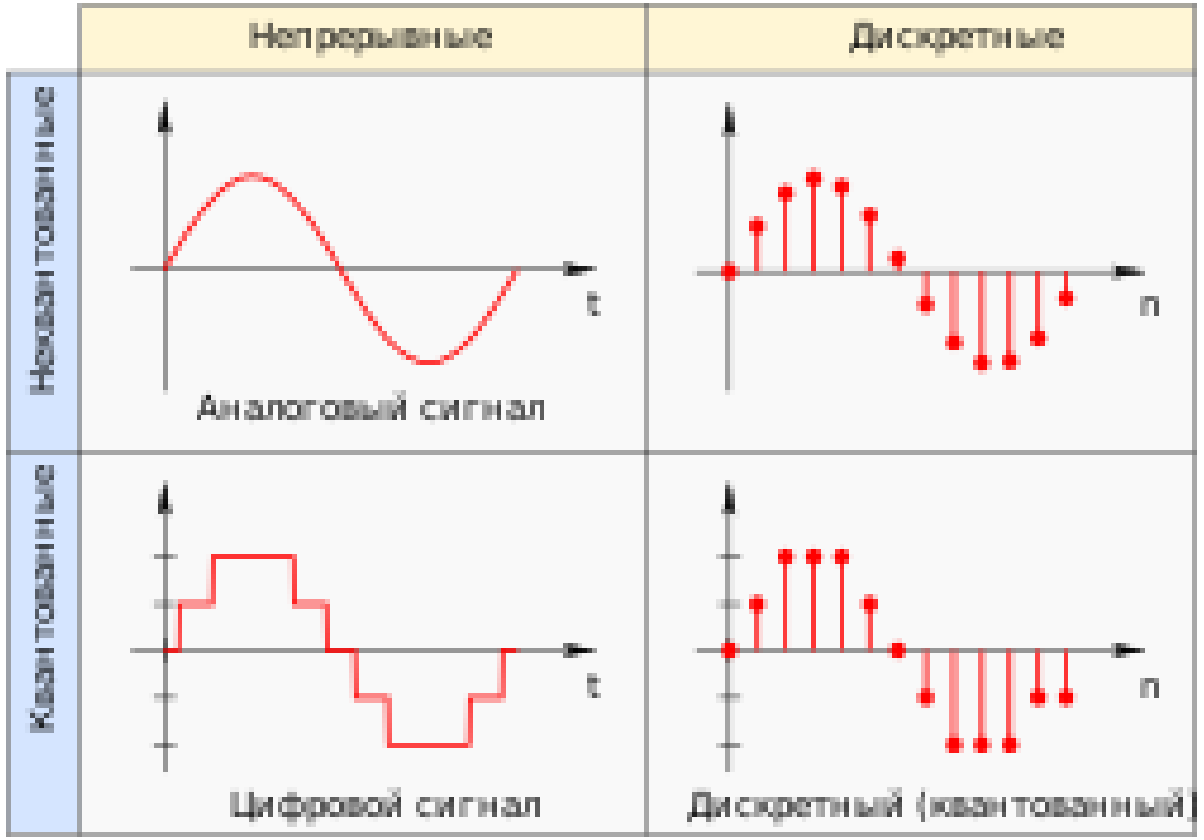
Agenda

- Signal types
- Input/output of analog signals
- Input/output of discrete signals
- Serial Interfaces
- Parallel Interfaces

Signal types

Сигнал - физический процесс (например, изменяющееся во времени напряжение), отображающий некоторую информацию или сообщение.

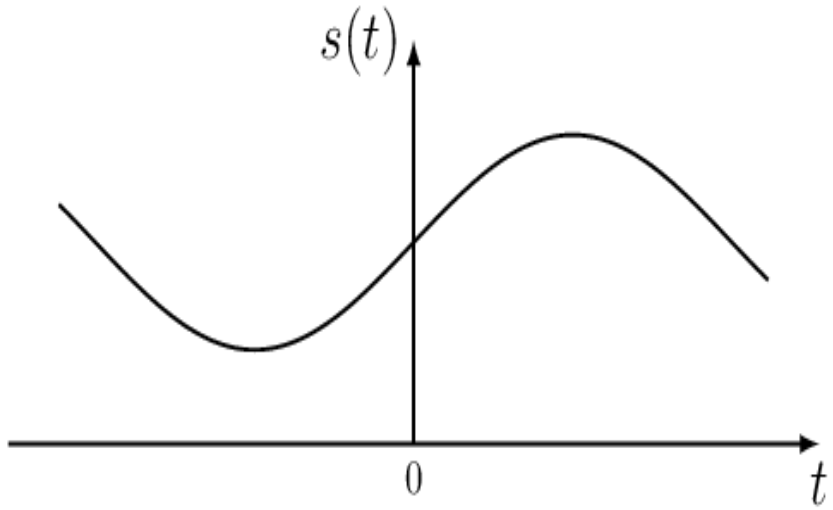
Математически может быть описан функцией определенного типа.



Signal types

Аналоговый сигнал определен на непрерывной оси времени ***t***, и в каждый момент может принимать произвольные значения.

Описывается непрерывной или кусочно-непрерывной функцией ***s(t)***.

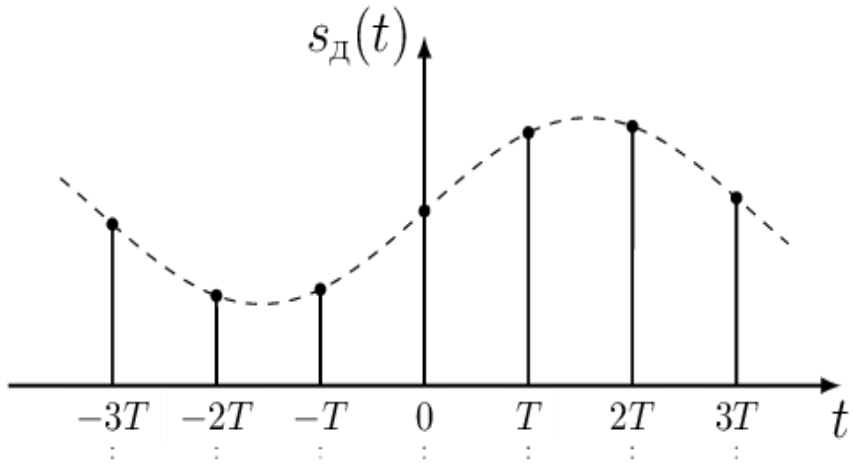


Signal types

Дискретные сигналы принимают произвольные значения только в фиксированные (дискретные) моменты времени – $t_n = nT$ (n=0, 1, 2, …. , T=const – период дискретизации).

Описываются решетчатыми функциями – последовательностями – ***Sд(nT)***, которые могут в дискретные моменты принимать произвольные значения на некотором интервале.

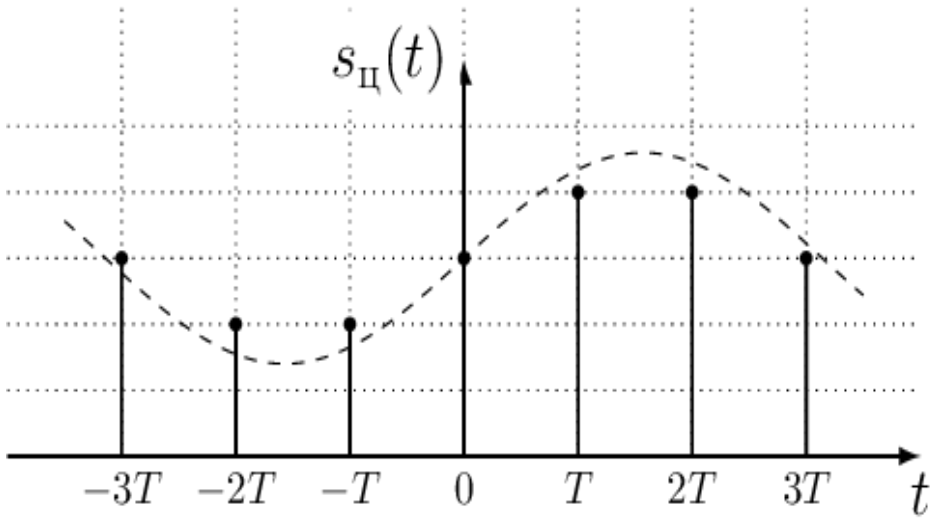
Могут обозначаться ***x(n)*** или ***xn***.



Signal types

Цифровые сигналы представляют собой дискретные сигналы, которые в дискретные моменты времени могут принимать лишь конечный ряд дискретных значений – уровней квантования (число конечной разрядности в одной из систем счисления).

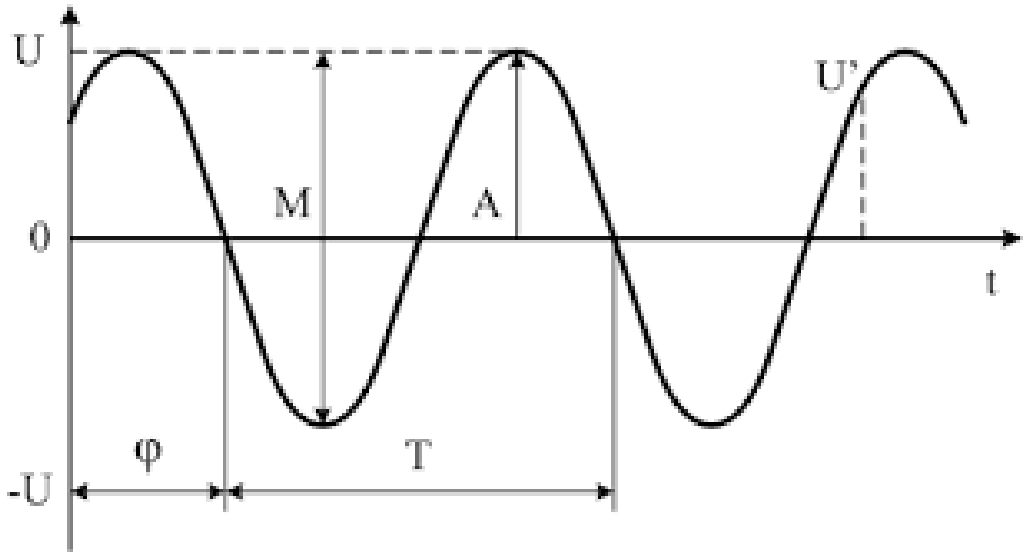
Описываются квантованными решетчатыми функциями ***Sц(nT)***.



Signal types

Характеристики аналогового сигнала:

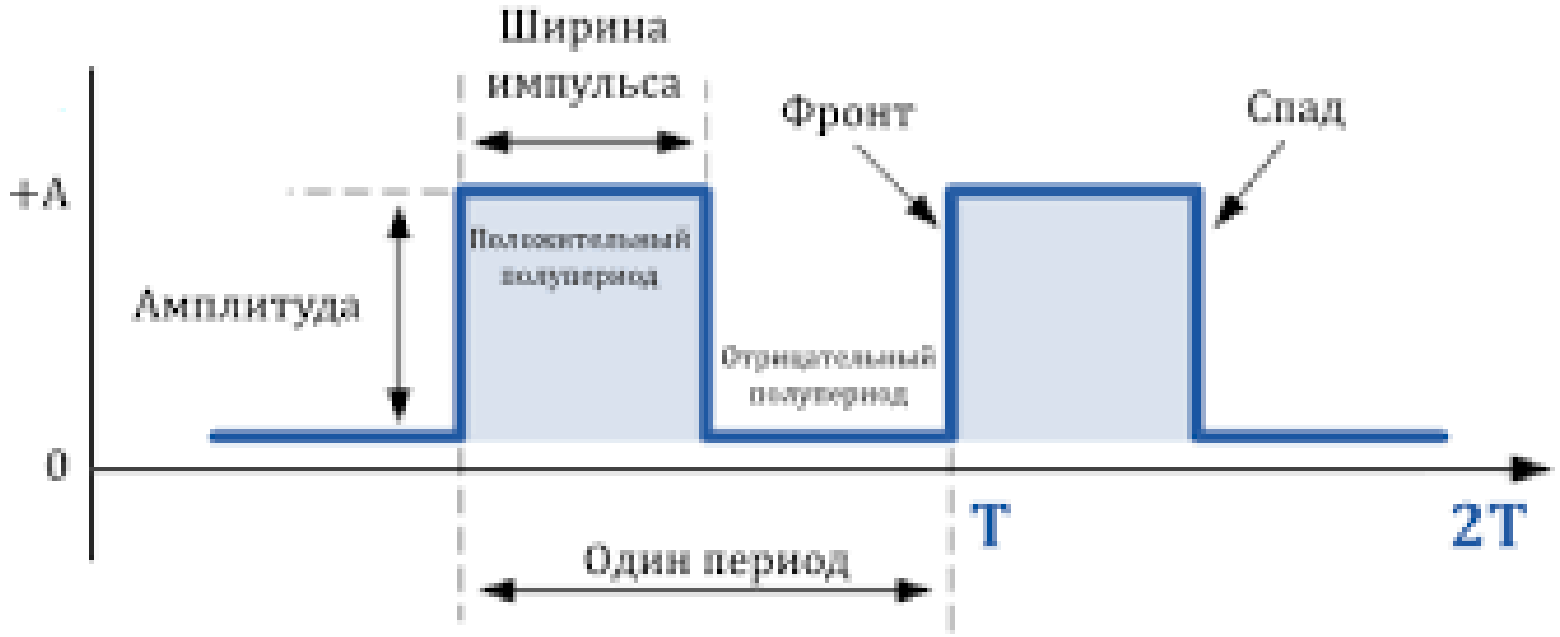
- Амплитуда (максимальное отклонение напряжения сигнала от нулевого порога)
- Частота (количество колебаний сигнала в единицу времени)
- Фаза (смещение сигнала)



Signal types

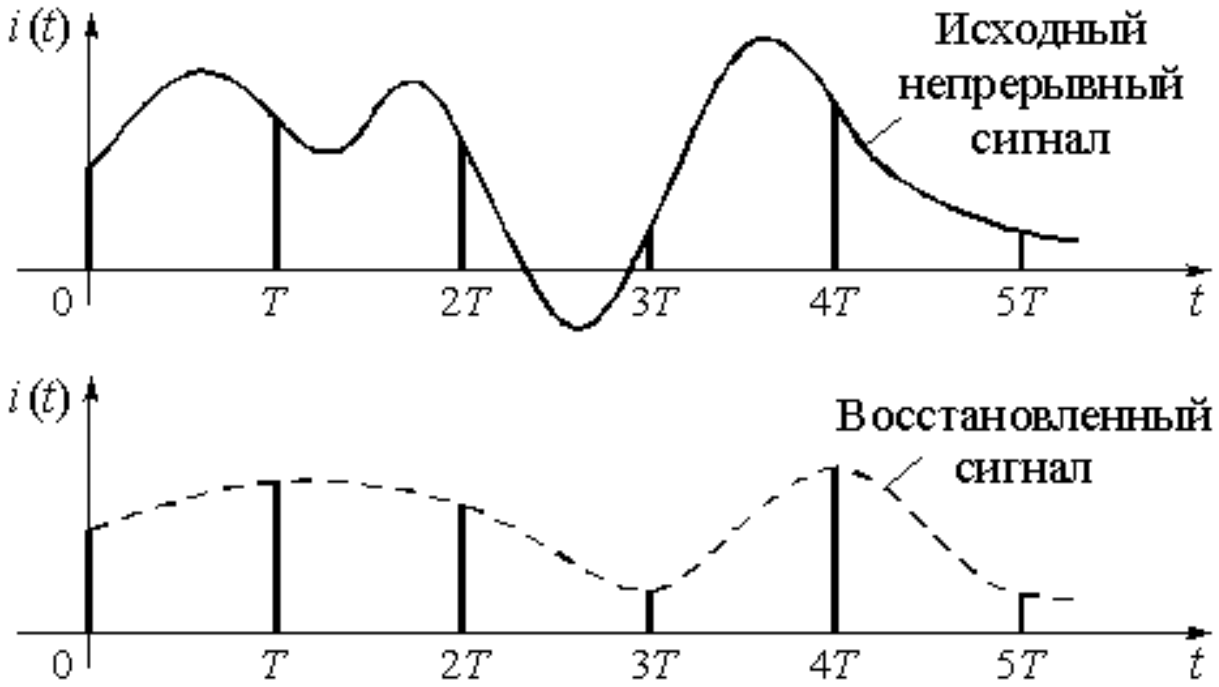
Характеристики цифрового сигнала:

- Амплитуда
- Частота
- Длительность импульса/паузы
- Длительность фронта/спада



Signal types

Преобразование аналогового сигнала в дискретный



Signal types

Ввод аналогового (непрерывного) сигнала:

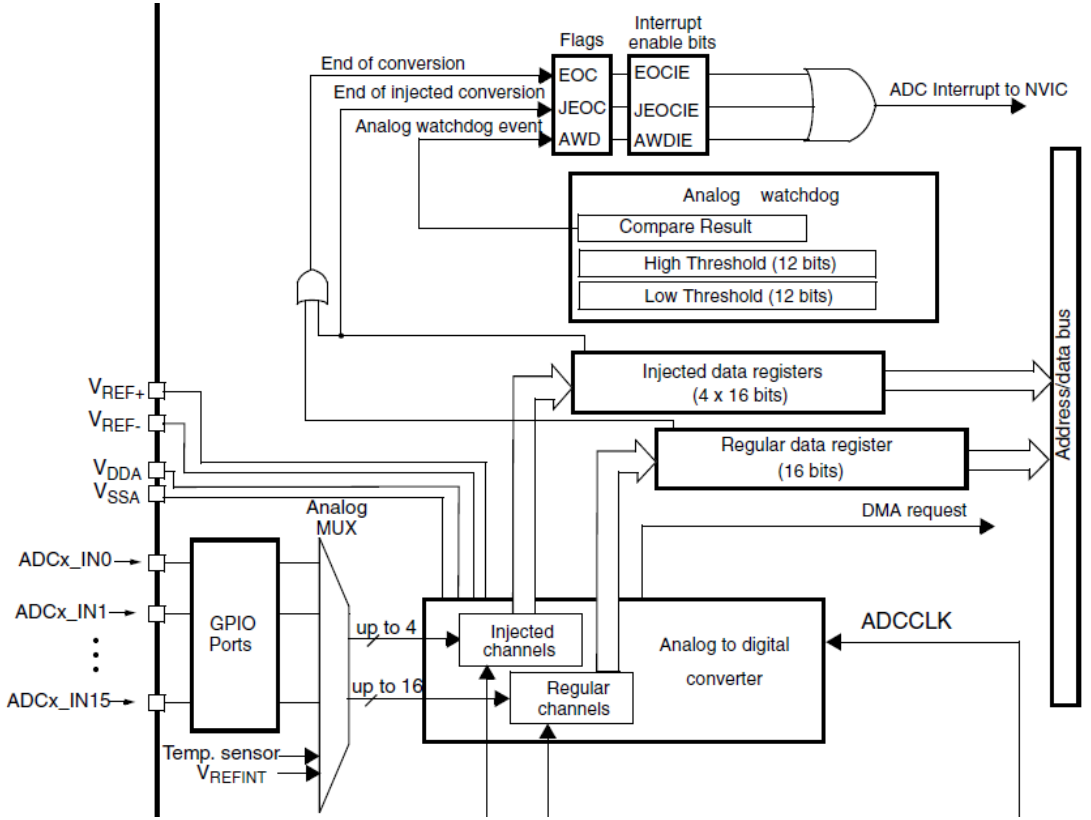
- 18 каналов ввода (16 внешних и 2 внутренних)
- разрешение 12 бит
- режимы преобразования:
  - однократное
  - непрерывное
  - по триггеру
  - по таймеру

Signal types

- выравнивание битов результата
- генерирование прерываний и сигналов для DMA
- скорость преобразования — до 0.9 MSPS
- автокалибровка
- сканирования входов по списку
- аналоговый Watchdog

Signal types

Структура модуля АЦП



Signal types

Опорное напряжение:

- V_REF- GND
- V_REF+
  - напряжение питания процессора (+3.3 В) 👎
  - внешний источник опорного напряжения (ИОН) 👍

Результат АЦП = V_In / (V_Ref+ — V_Ref-) * 4095

Signal types

Инициализация АЦП:

- Включить тактирование модуля АЦП
- Настроить параметры модуля
- Включить модуль АЦП
- Настроить вход (номер канала АЦП)
- Выполнить калибровку

Signal types

Измерение сигнала:

- Запустить преобразование
- Ожидать окончание преобразования
  (флаг EOC = End Of Conversion)
- Прочитать результат из регистра DR

Signal types

Измерение сигнала:

```
ADC_SoftwareStartConvCmd(ADC1, ENABLE);
while(ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC) == RESET);
adcResult = ADC_GetConversionValue(ADC1);
```

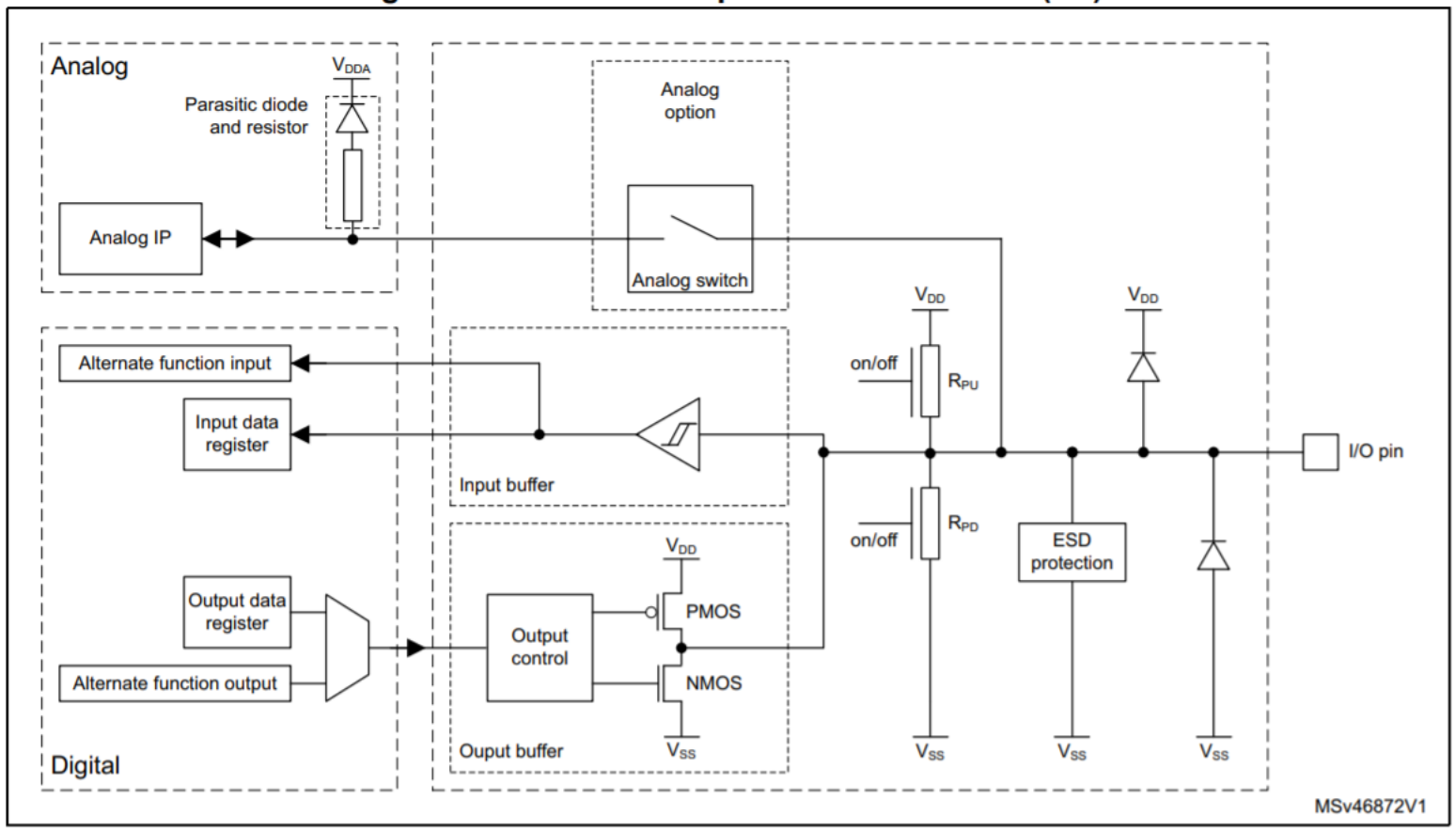
Signal types

Ввод и вывод цифрового сигнала:

- Структура порта
- Согласование уровней
- Резисторы подтяжки
- Альтернативные функции порта
- Устранение дребезга
- Периодичность опроса
- Использование прерываний
- Формирование сигналов PWM

Signal types

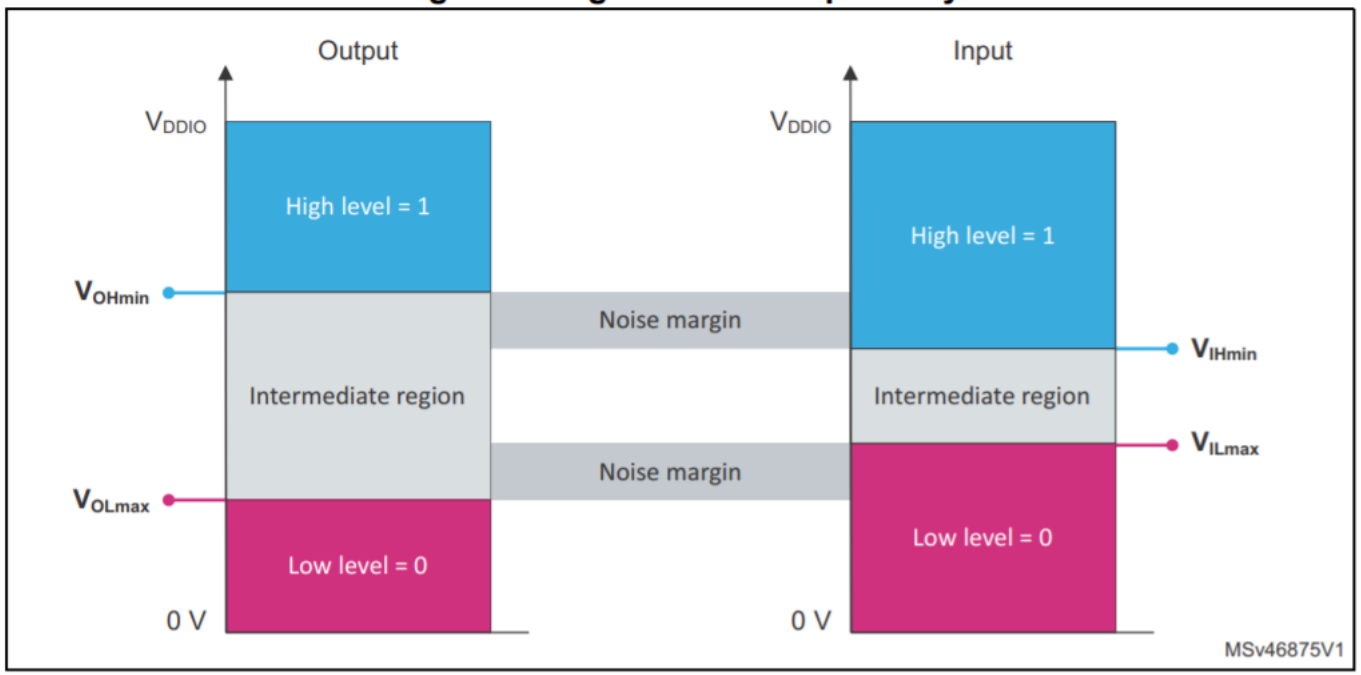
Figure 1. Three-volt compliant GPIO structure (TC)



Signal types

Согласование уровней

Figure 4. Logical level compatibility



Signal types

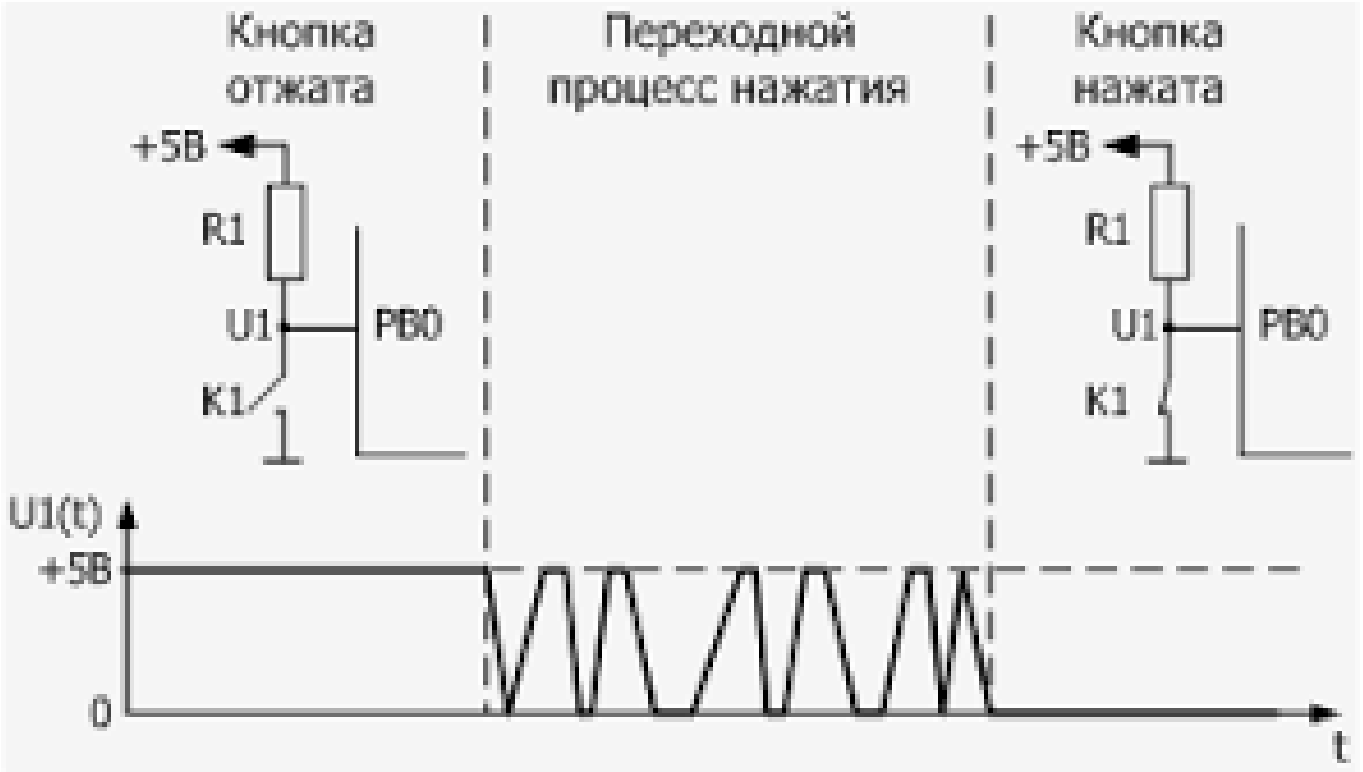
- Альтернативные функции порта

Table 7. STM32F40xxx pin and ball definitions (continued)

Pin number						Pin name (function after reset)(1)	Pin type	I / O structure	Notes	Alternate functions	Additional functions
LQFP64	WLCSP90	LQFP100	LQFP144	UFBGA176	LQFP176						
	D9			L4	48	BYPASS_REG	I	FT	-	-	-
19	E4	28	39	K4	49	V_{DD}	S	-	-	-	-
20	J9	29	40	N4	50	PA4	I/O	TTa	(4)	SPI1_NSS / SPI3_NSS / USART2_CK / DCMI_HSYNC / OTG_HS_SOF/ I2S3_WS/ EVENTOUT	ADC12_IN4 /DAC_OUT1

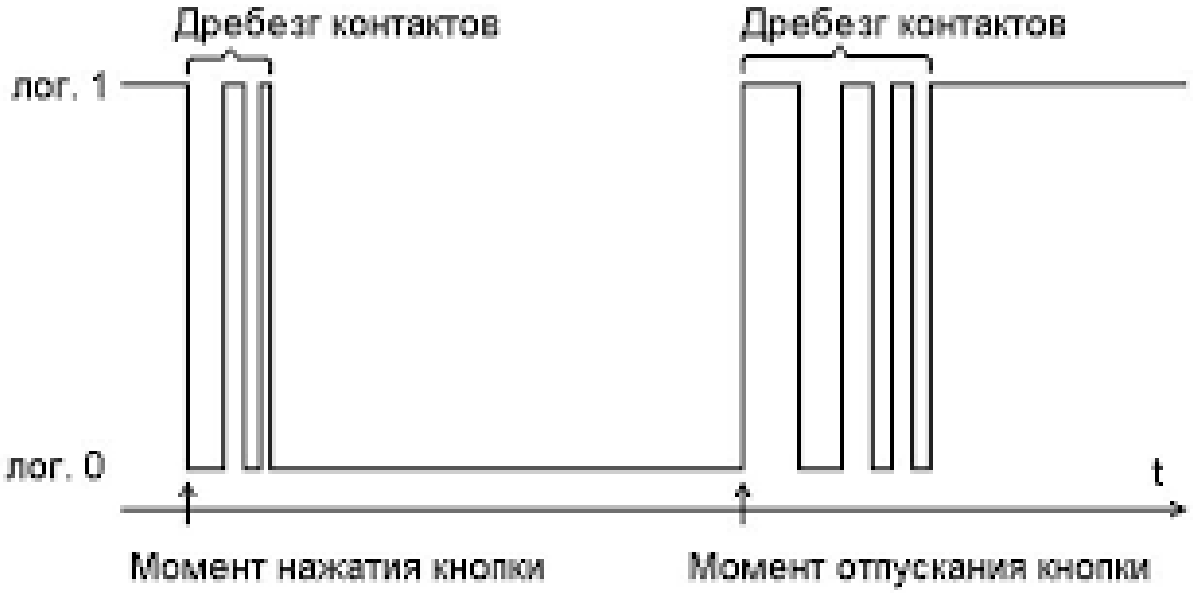
Signal types

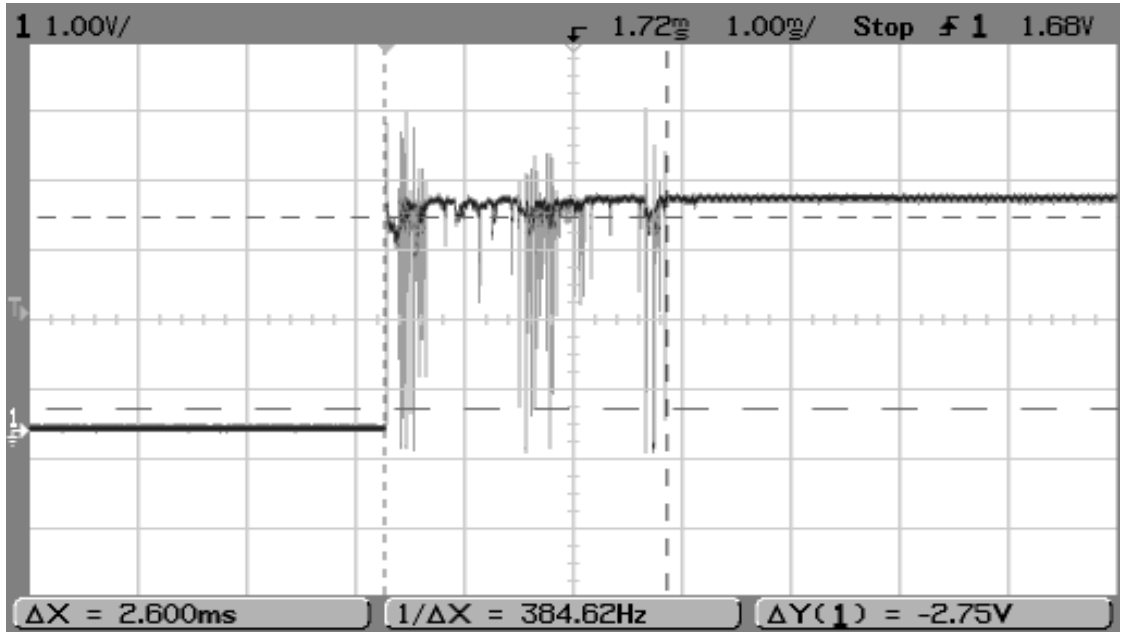
- Дребезг контактов



Signal types

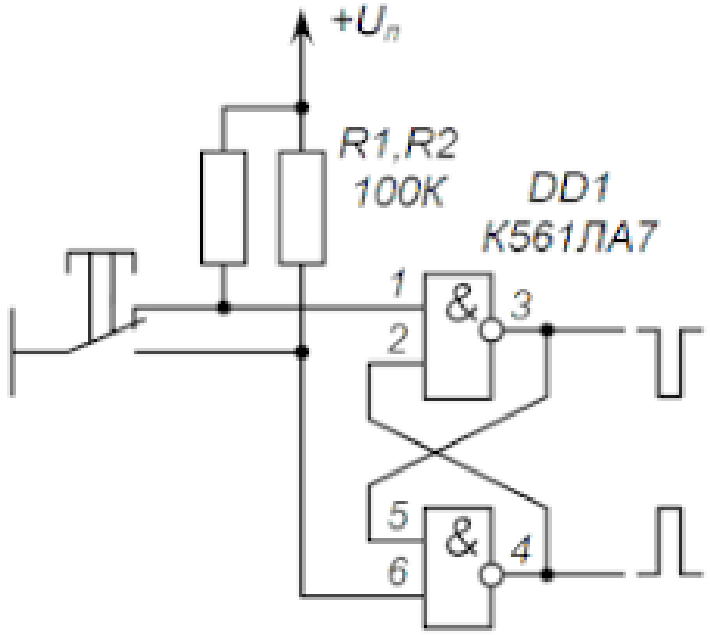
- Дребезг контактов

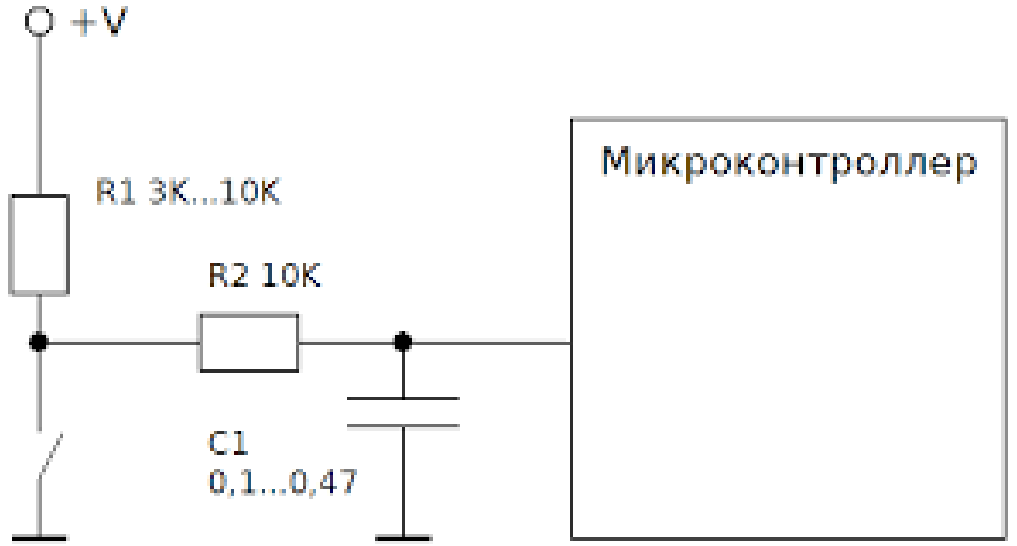




Signal types

- Дребезг контактов. Аппаратное устранение

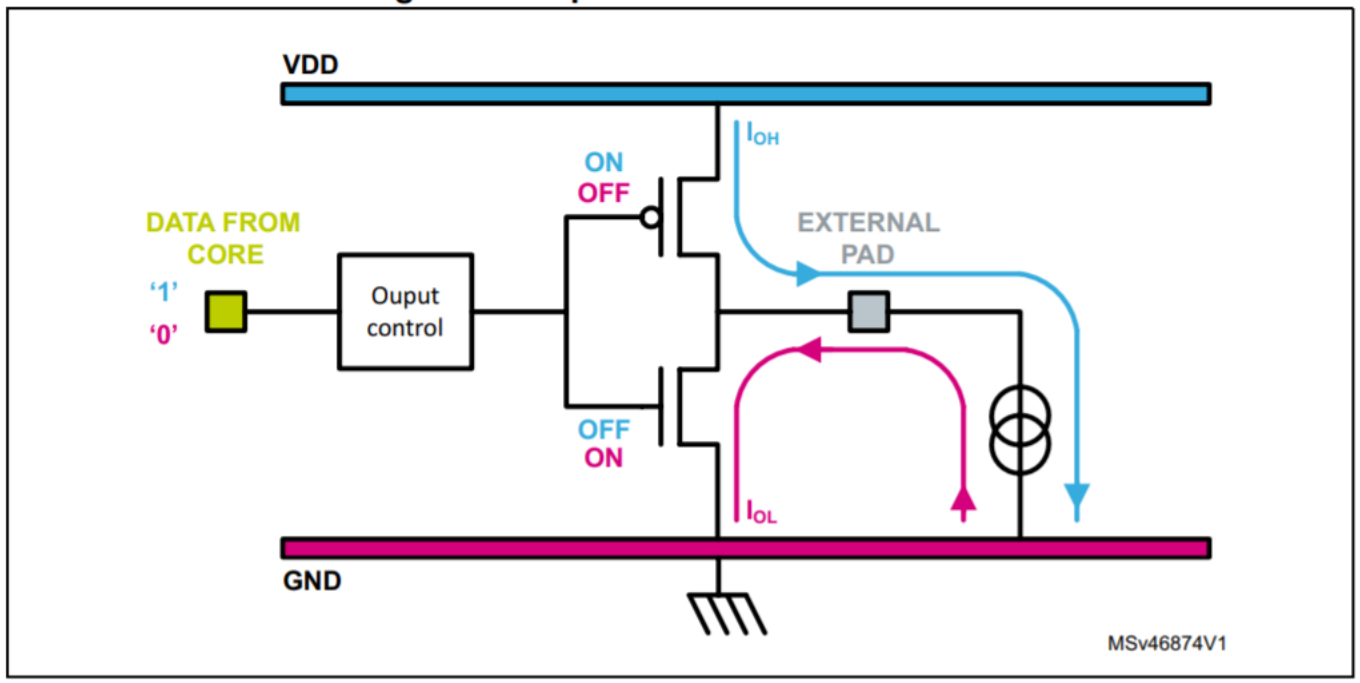




Signal types

Согласование уровней

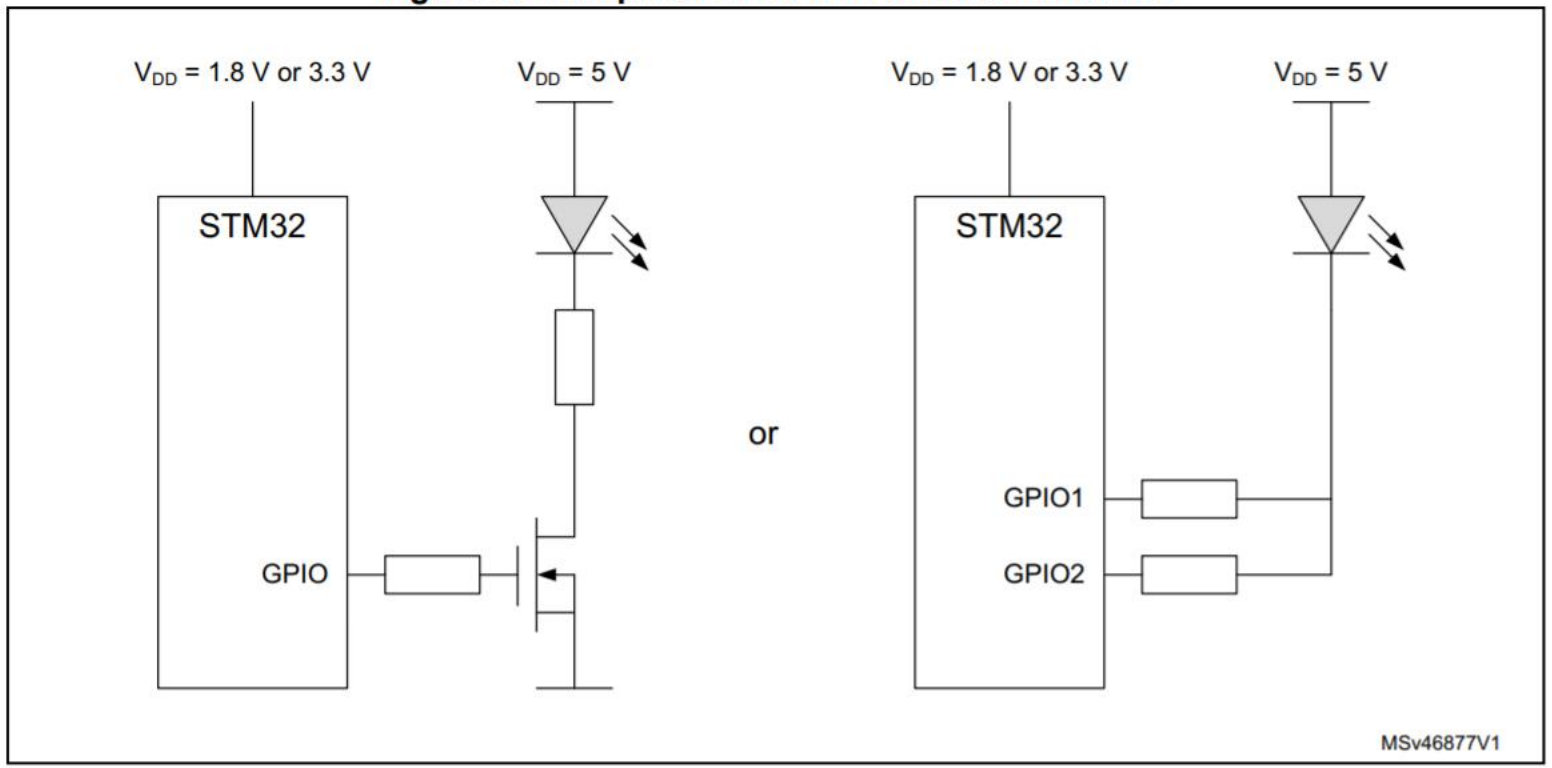
Figure 3. Output buffer and current flow



Signal types

Согласование уровней

Figure 6. Example of white LED drive connections



Signal types

Настройка порта

```
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOD_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);

  /*Configure GPIO pin : PD13 */
  GPIO_InitStruct.Pin = GPIO_PIN_12;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
}
```

```
typedef enum
{
  GPIO_PIN_RESET = 0,
  GPIO_PIN_SET
}GPIO_PinState;
```

```
#define  GPIO_MODE_INPUT
#define  GPIO_MODE_OUTPUT_PP
#define  GPIO_MODE_OUTPUT_OD
#define  GPIO_MODE_AF_PP
#define  GPIO_MODE_AF_OD

#define  GPIO_MODE_ANALOG

#define  GPIO_MODE_IT_RISING
#define  GPIO_MODE_IT_FALLING
#define  GPIO_MODE_IT_RISING_FALLING

#define  GPIO_MODE_EVT_RISING
#define  GPIO_MODE_EVT_FALLING
#define  GPIO_MODE_EVT_RISING_FALLING
```

Signal types

Настройка порта

```
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOD_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);

  /*Configure GPIO pin : PD13 */
  GPIO_InitStruct.Pin = GPIO_PIN_12;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
}
```

```
#define   GPIO_NOPULL
#define   GPIO_
#define   GPIO_PULLDOWN
```

```
#define   GPIO_SPEED_FREQ_LOW
          // IO works at 2 MHz
#define   GPIO_SPEED_FREQ_MEDIUM
          // range 12,5 MHz to 50 MHz
#define   GPIO_SPEED_FREQ_HIGH
          // range 25 MHz to 100 MHz
#define   GPIO_SPEED_FREQ_VERY_HIGH
          // range 50 MHz to 200 MHz
```

Signal types

Чтение, запись порта

```
HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
```

returns GPIO_PIN_SET
or GPIO_PIN_RESET

```
HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12 | GPIO_PIN_13);
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
```

Signal types

Использование прерываний

- Настроить порт на внешнее прерывание

```
GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
// GPIO_MODE_IT_FALLING or GPIO_MODE_IT_RISING_FALLING
```

- Настроить и разрешить внешние прерывания EXTI0_IRQn

```
/* EXTI interrupt init*/
  HAL_NVIC_SetPriority(EXTI0_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(EXTI0_IRQn);
```

Signal types

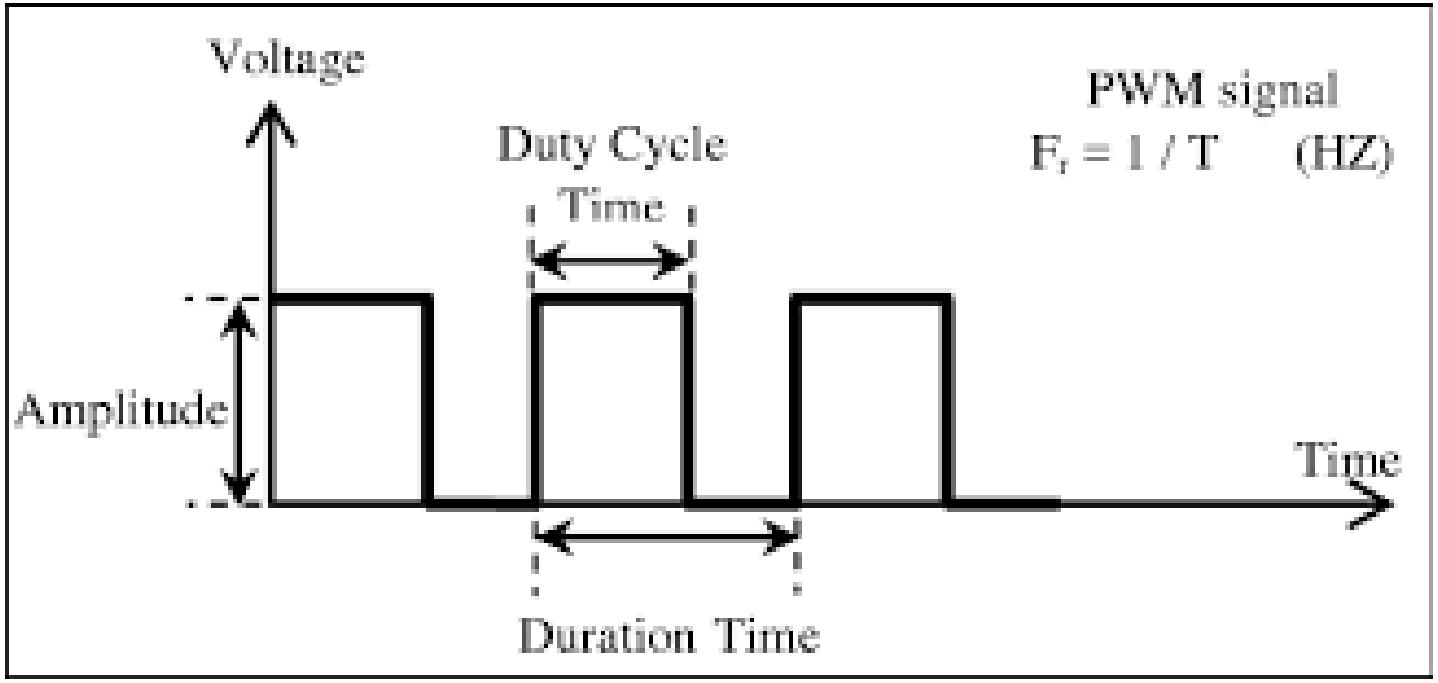
Использование прерываний

- Написать обработчик прерывания (*файл* ***stm32f4xx_it.c***)

```
void EXTI0_IRQHandler(void) {
    /* USER CODE BEGIN EXTI0_IRQn 0 */
    key_pressed = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
    /* USER CODE END EXTI0_IRQn 0 */
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
    /* USER CODE BEGIN EXTI0_IRQn 1 */
    /* USER CODE END EXTI0_IRQn 1 */
}
```

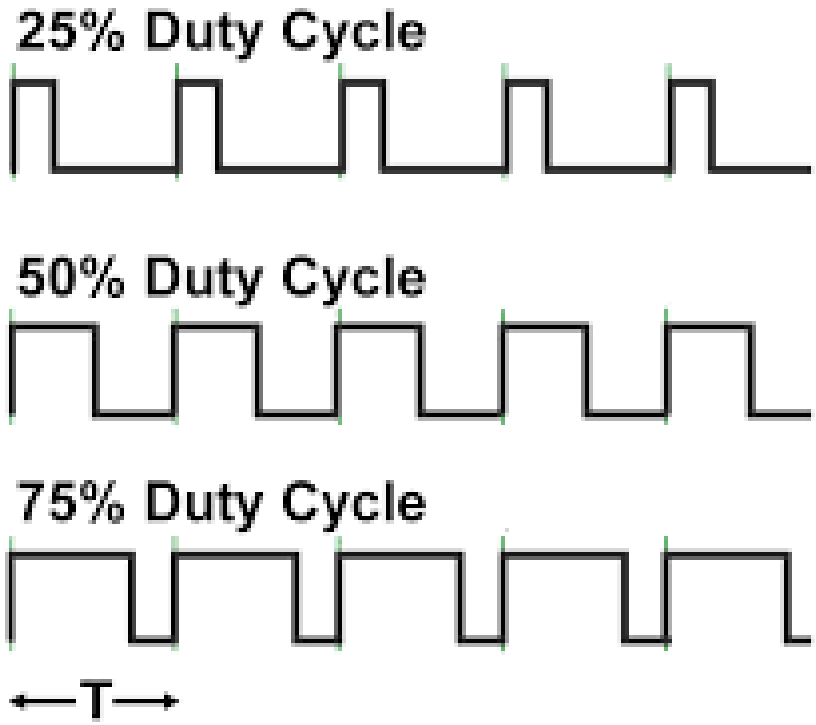
Signal types

Формирование сигналов PWM



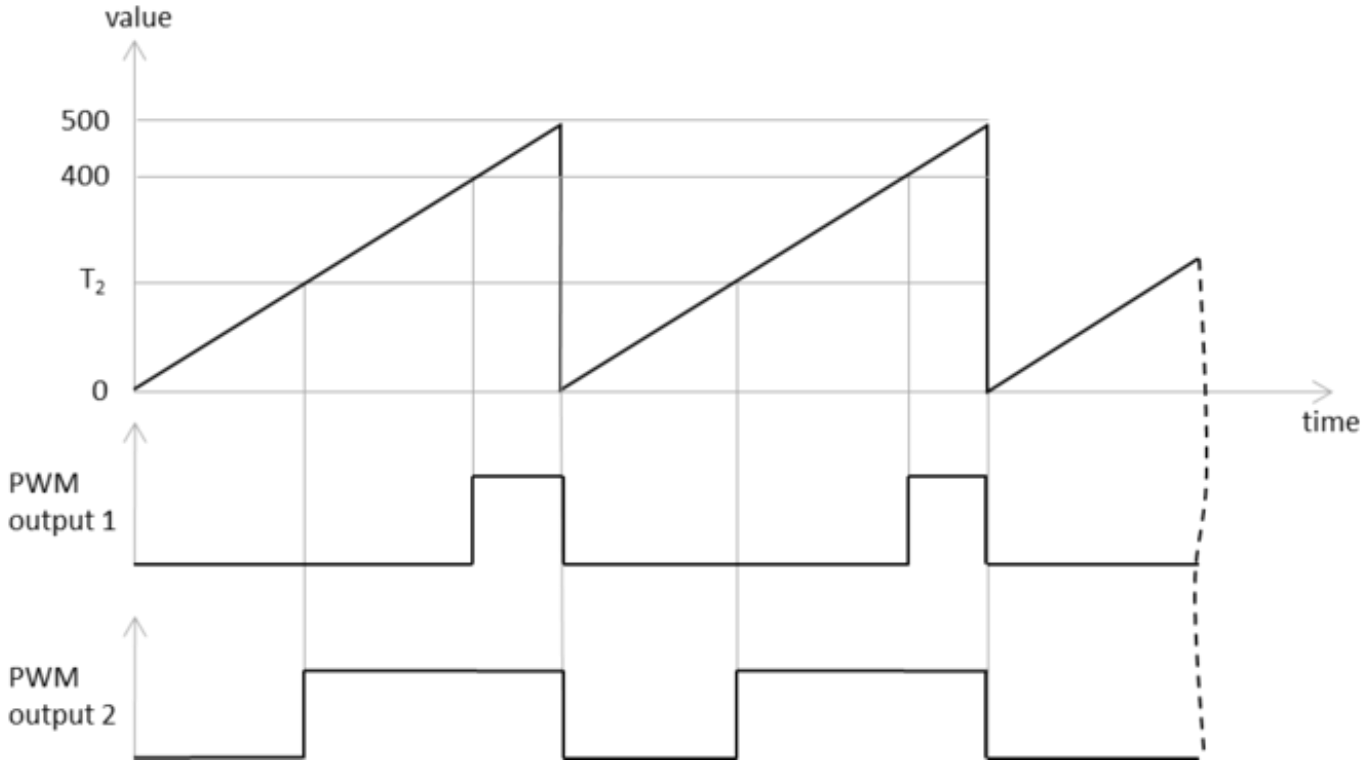
Signal types

Формирование сигналов PWM



Signal types

Формирование сигналов PWM



Signal types

Table 7. STM32F40xxx pin and ball definitions (continued)

Pin number						Pin name (function after reset)(1)	Pin type	I / O structure	Notes	Alternate functions	Additional functions
LQFP64	WLCSP90	LQFP100	LQFP144	UFBGA176	LQFP176						
-	-	60	82	M15	101	PD13	I/O	FT	-	FSMC_A18/TIM4_CH2/ EVENTOUT	-
-	G2	59	81	N13	100	PD12	I/O	FT	-	FSMC_ALE/ FSMC_A17/TIM4_CH1 / USART3_RTS/ EVENTOUT	-
-	F2	61	85	M14	104	PD14	I/O	FT	-	FSMC_D0/TIM4_CH3/ EVENTOUT/ EVENTOUT	-
-	F1	62	86	L14	105	PD15	I/O	FT	-	FSMC_D1/TIM4_CH4/ EVENTOUT	-

Signal types

Формирование PWM сигнала:

- Выбрать таймер для формирования PWM сигнала
- Настроить порт как канал таймера
- Настроить базовый таймер
- Настроить канал таймера
- Стартовать формирование PWM сигнала

Signal types

Управление значением PWM сигнала:

- Настроить канал таймера
- Стартовать формирование PWM сигнала

Serial Interfaces

Последовательный интерфейс - интерфейс, в котором все информационные сигналы передаются по одной линии.

Интерфейс - совокупность средств, методов и правил взаимодействия (управления, контроля и т.д.) между элементами системы.

Serial Interfaces

- синхронный и асинхронный режим
- однонаправленный и двунаправленный режим
- количество линий связи
- количество устройств
- топология (общая шина, звезда, дерево)
- скорость передачи

Serial Interfaces

- 1-Wire (OneWire)
- SPI
- I2C (TWI, SMBus)
- I2S
- UART
- SDIO
- CAN
- USB
- Ethernet

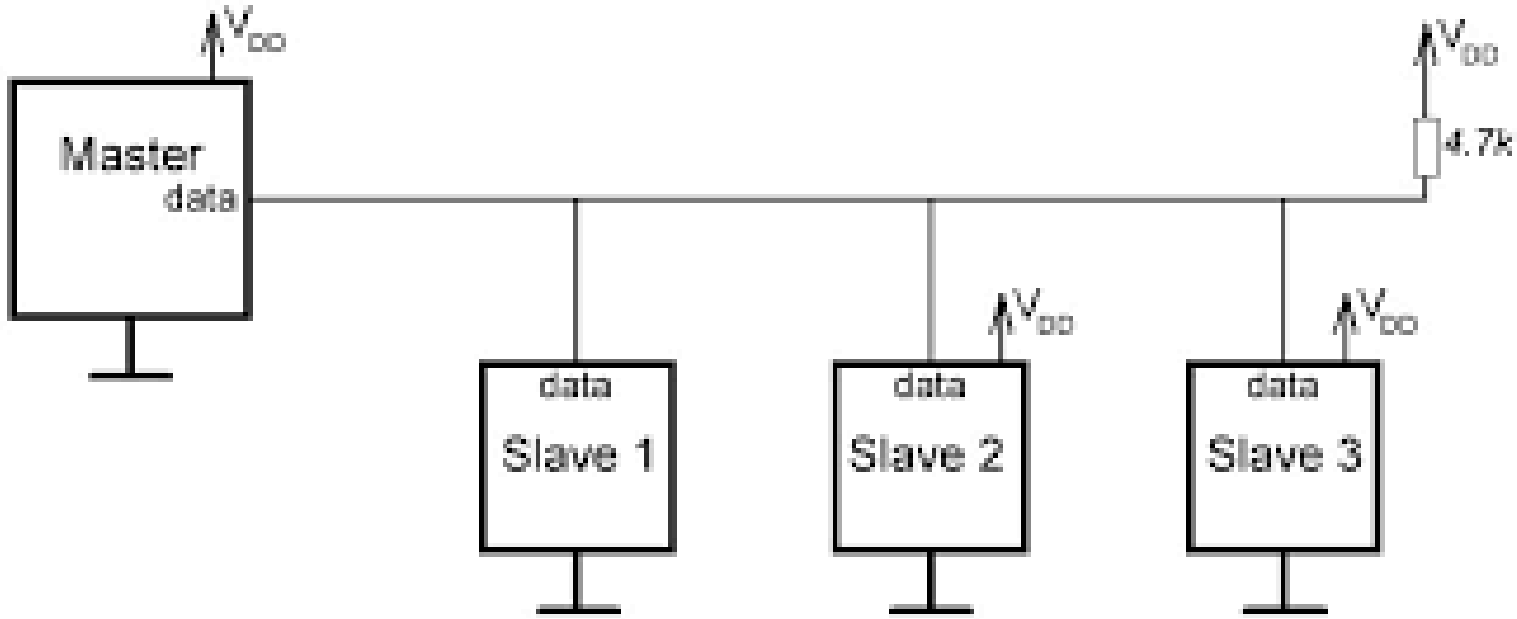
Serial Interface 1-Wire

Интерфейс **1Wire** (**OneWire**) разработан компанией **Dallas Semiconductor** (с 2001 — **Maxim Integrated**)

- асинхронный
- Двунаправленный (полудуплекс)
- топология — общая шина (сеть – MicroLan)
- скорость передачи 16,3 Кбит/с

Serial Interface 1-Wire

Топология сети **1Wire**



Serial Interface 1-Wire

Достоинства:

- два провода для связи: данные и общий
- паразитное питание (конденсатор 800 пФ для питания от линии данных)
- большое расстояние передачи (до 300м)
- изменяемость конфигурации в процессе работы
- уникальный идентификатор для каждого устройства

Serial Interface 1-Wire

Структура идентификатора

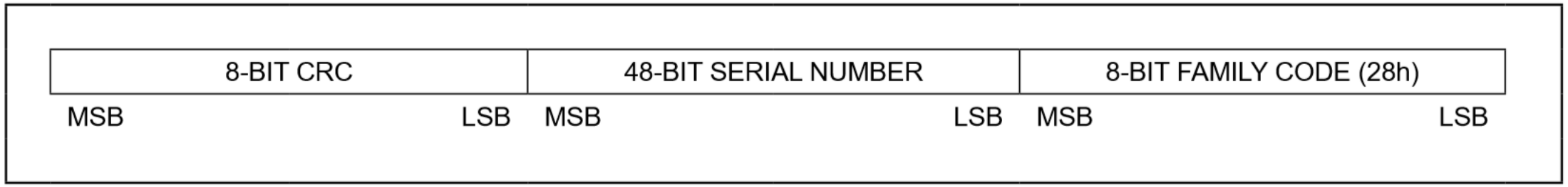


Figure 8. 64-Bit Lasered ROM Code

```
0x28, 0xFF, 0x2B, 0x45, 0x4C, 0x04, 0x00, 0x10
```

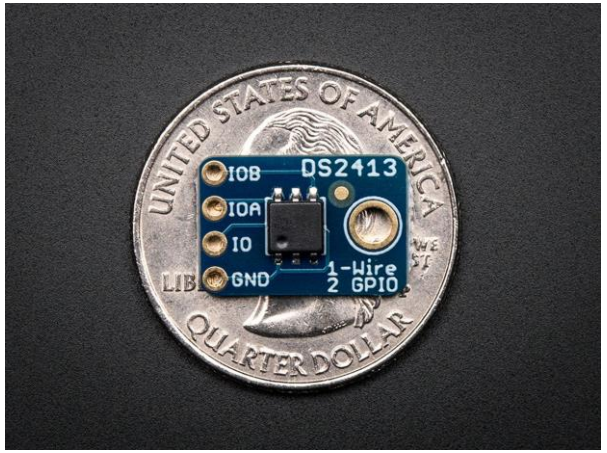
Serial Interface 1-Wire

Код семей-ства (HEX)	**Устройства** ()iButton Package	**Оригинальное описание** (Memory size in bits unless specified)	**Описание**
01	(DS1990A)*, DS2401	1-Wire net address (serial num-ber) only	Уникальный серийный номер-ключ
04	(DS1994), DS2404	4k NV RAM memory and clock, timer, alarms	4К энергонезависимого ОЗУ + часы, таймер и будильник
05	DS2405	Single addressable switch	Одиночный адресуемый ключ
06	(DS1993)	4k NV RAM memory	4К энергонезависимого ОЗУ
08	(DS1992)	1k NV RAM memory	1К энергонезависимого ОЗУ
09	(DS1982), DS2502	1k EPROM memory	1К электрически программируе-мого ПЗУ
0A	(DS1995)	16k NV RAM memory	16К энергонезависимого ОЗУ

Serial Interface 1-Wire

Применение:

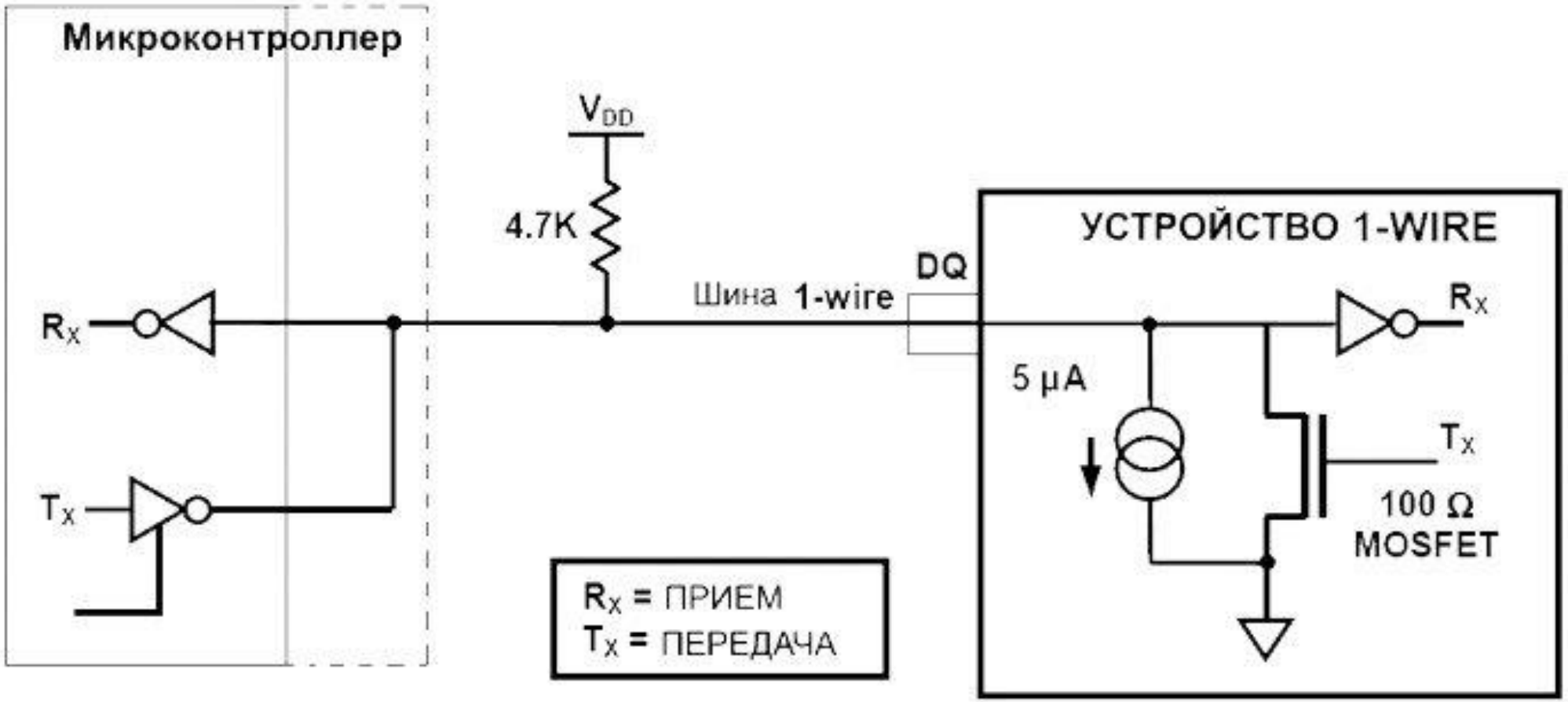
DS1990A



DS2413

DS18B20

Serial Interface 1-Wire



Serial Interface 1-Wire

Особенности программной реализации интерфейса:

- Режим **hotplug** («горячее» подключение и отключение устройств от шины)
- Обмен всегда инициируется ведущим устройством (микроконтроллером)
- Обмен всегда начинается с импульса «**reset**» (генерируется мастером)
- Обмен ведётся **тайм-слот**ами (промежутки времени передачи одного бита)
- Данные передаются, начиная с менее значимого (младшего) бита
- Каждый пакет данных сопровождается контрольной суммой (**CRC-8**)

Serial Interface 1-Wire

Сеанс обмена:

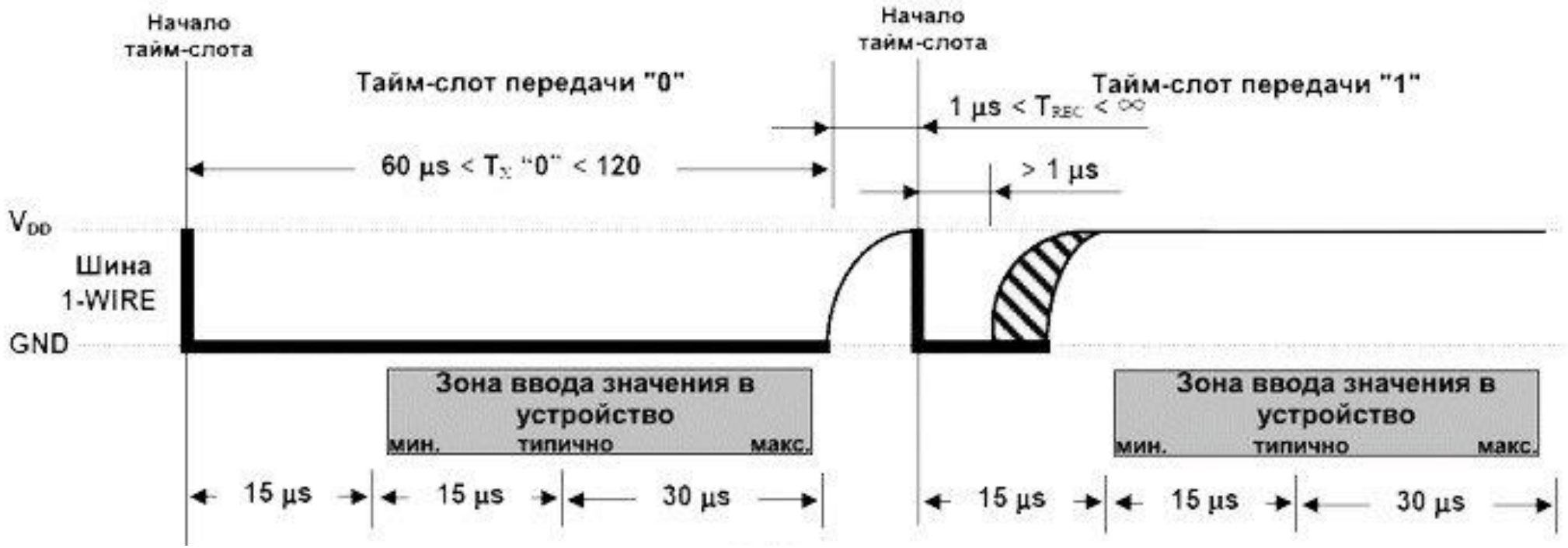
- Мастер посылает сигнал **RESET**
- Если на шине есть хотя бы одно устройство, оно отвечает на импульс, посылая свой импульс **PRESENCE** (этот же импульс отсылает устройство после подачи питания, **power-upreset**)
- Ведется передача данных
- Передача **RESET** мастером (либо для досрочного прекращения обмена, либо для завершения сеанса)

Serial Interface 1-Wire

Диаграмма сигналов **RESET** и **PRESENC**

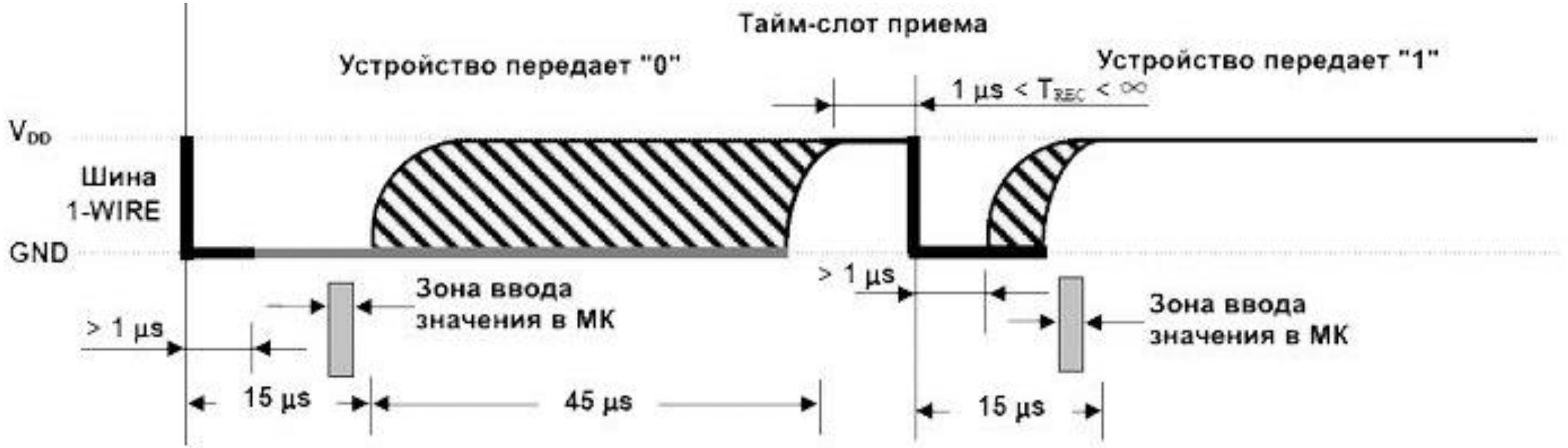
Serial Interface 1-Wire

Диаграммы тайм-слотов для передачи



Serial Interface 1-Wire

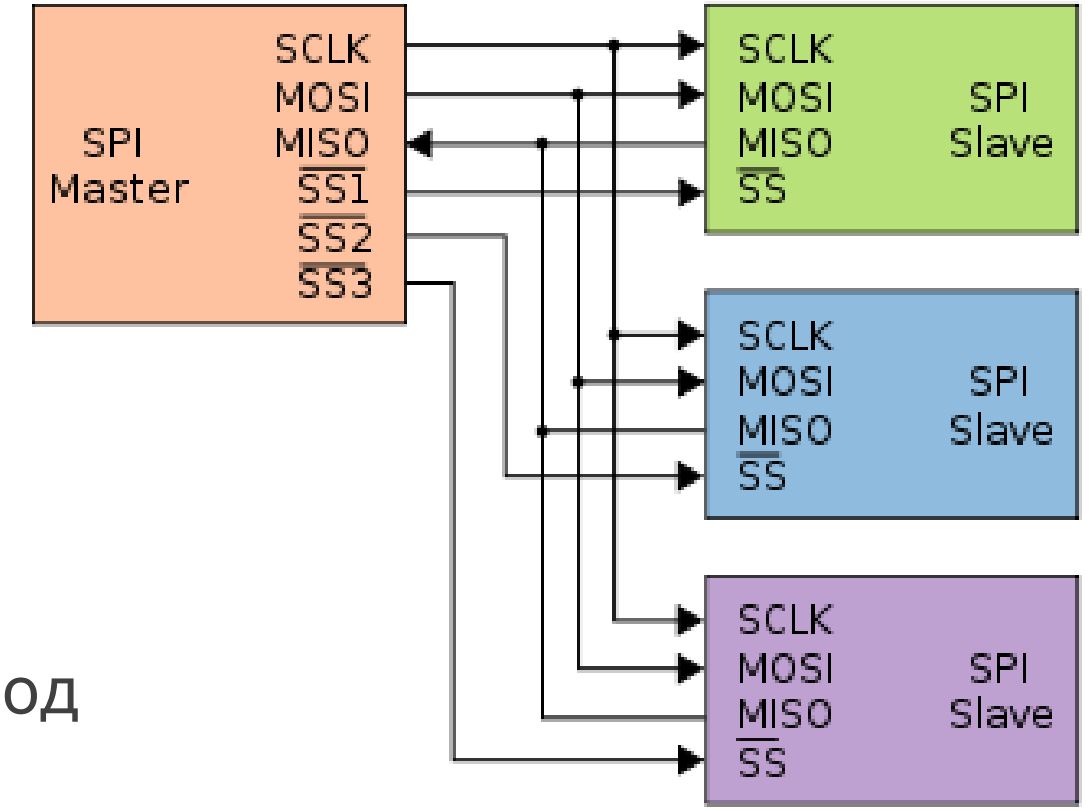
Диаграммы тайм-слотов для приема



Serial Interface SPI

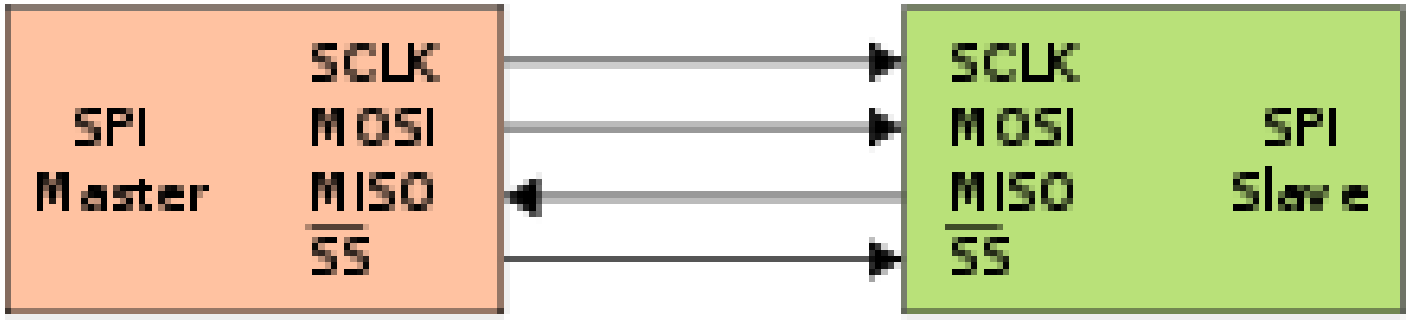
Serial **P**eripheral **I**nterface

- синхронный
- полный дуплекс
- топология – общая шина
- разрядность данных – 8 (16)
- скорость передачи – до 10 МБод



Serial Interface SPI

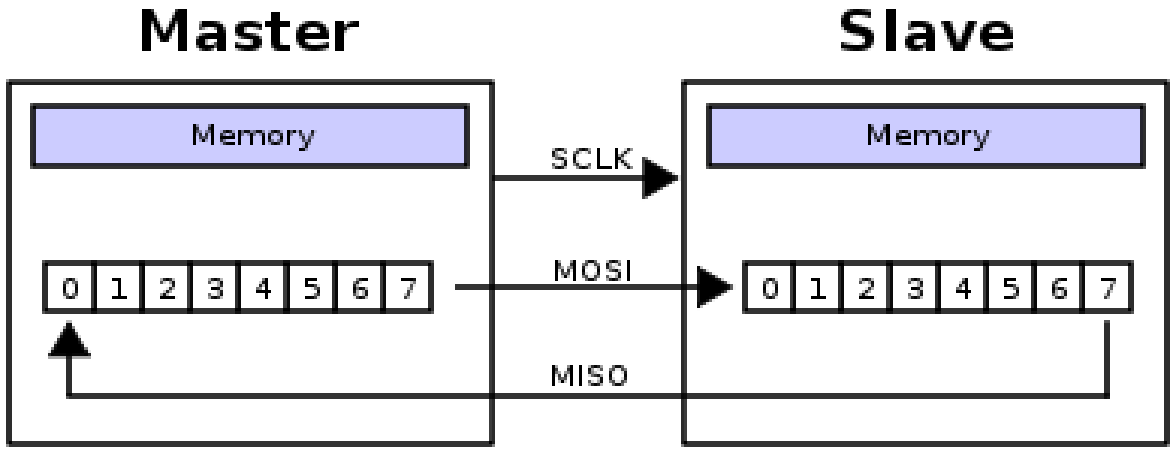
- MOSI — Master Out Slave In (SDO, DO, SO)
- MISO — Master In Slave Out (SDI, DI, DIN, SI)
- SCLK — Serial Clock (SCK, CLK)
- CS или SS — Chip Select, Slave Select (nCS, nSS)



Serial Interface SPI

Режимы передачи

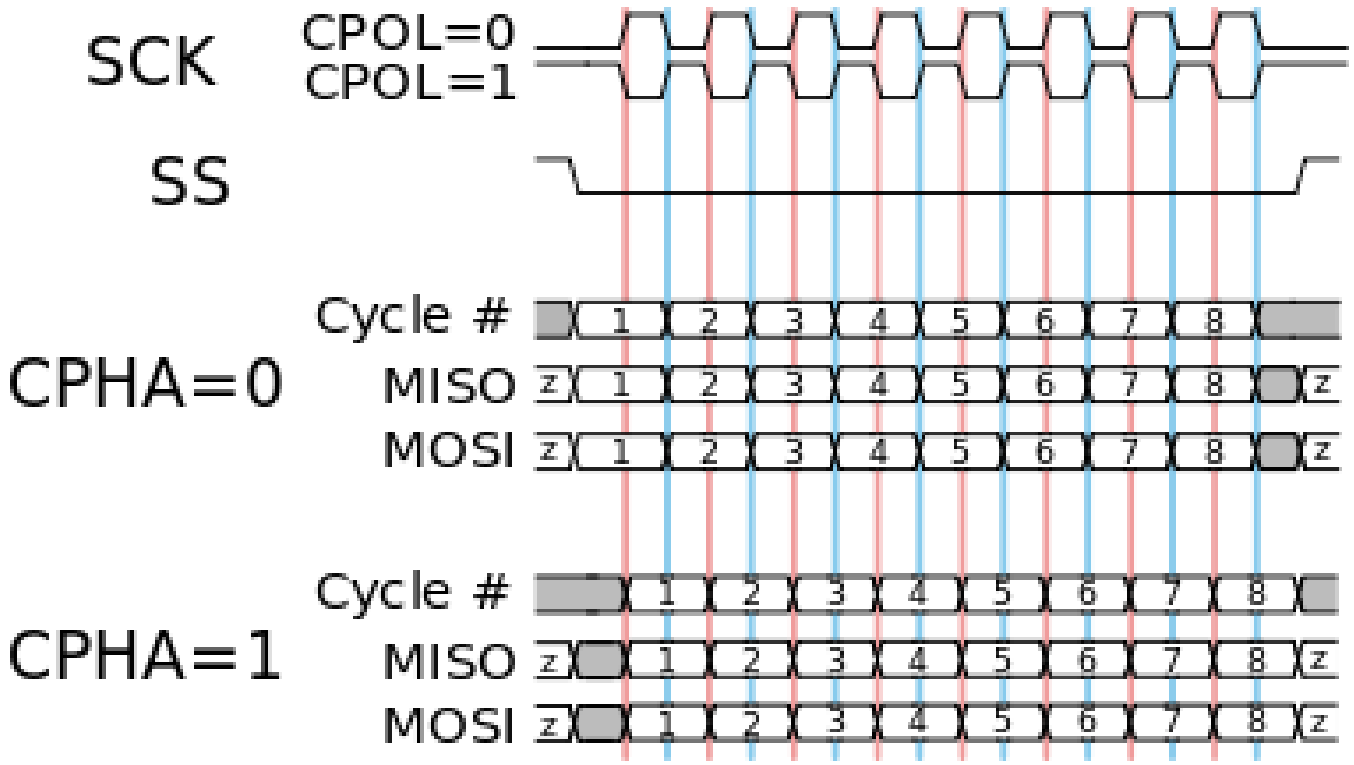
- режим 0 (CPOL = 0, CPHA = 0)
- режим 1 (CPOL = 0, CPHA = 1)
- режим 2 (CPOL = 1, CPHA = 0)
- режим 3 (CPOL = 1, CPHA = 1)



Serial Interface SPI

- CPOL = 0 — сигнал синхронизации начинается с низкого уровня
- CPOL = 1 — сигнал синхронизации начинается с высокого уровня
- CPHA = 0 — выборка данных производится по переднему фронту SCLK
- CPHA = 1 — выборка данных производится по заднему фронту SCLK

Serial Interface SPI



Serial Interface I2C

Inter-**I**ntegrated **C**ircuit (IIC) (разработчик – Philips)

- SMBus (System Management Bus)
- TWI (Two Wire Interface)
- TWSI (Two Wire Serial Interface)

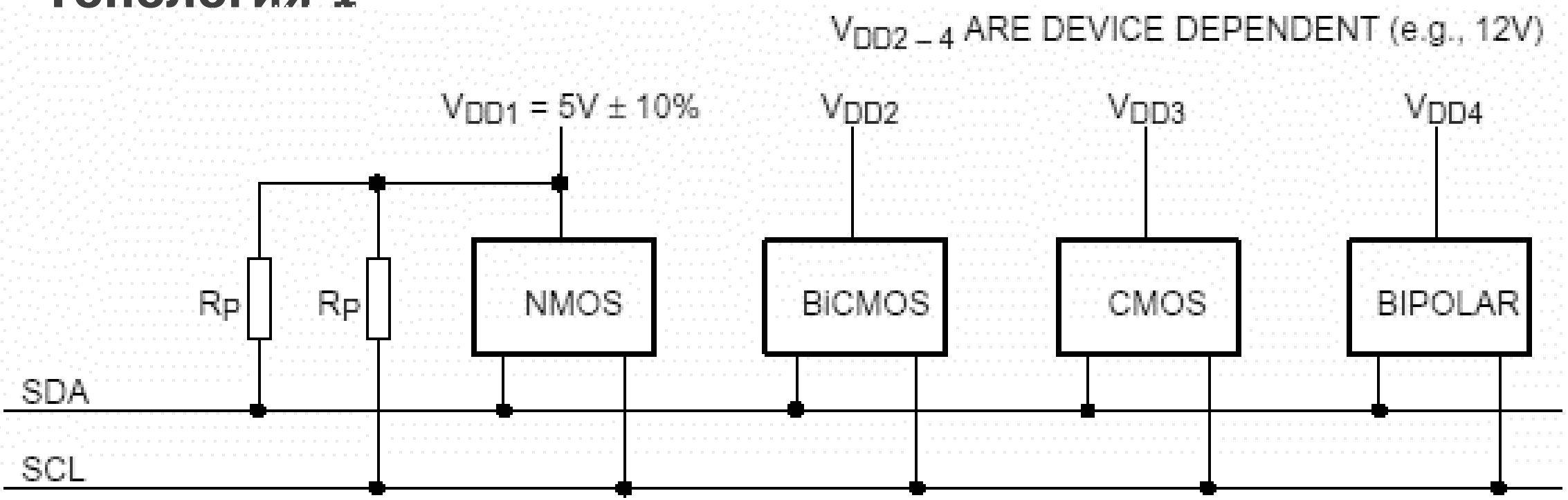
Serial Interface I2C

Inter-**I**ntegrated **C**ircuit (IIC)

- синхронный
- полудуплекс
- топология – общая шина
- разрядность данных – 8
- скорость передачи – до 100 кБод (400 кБод))

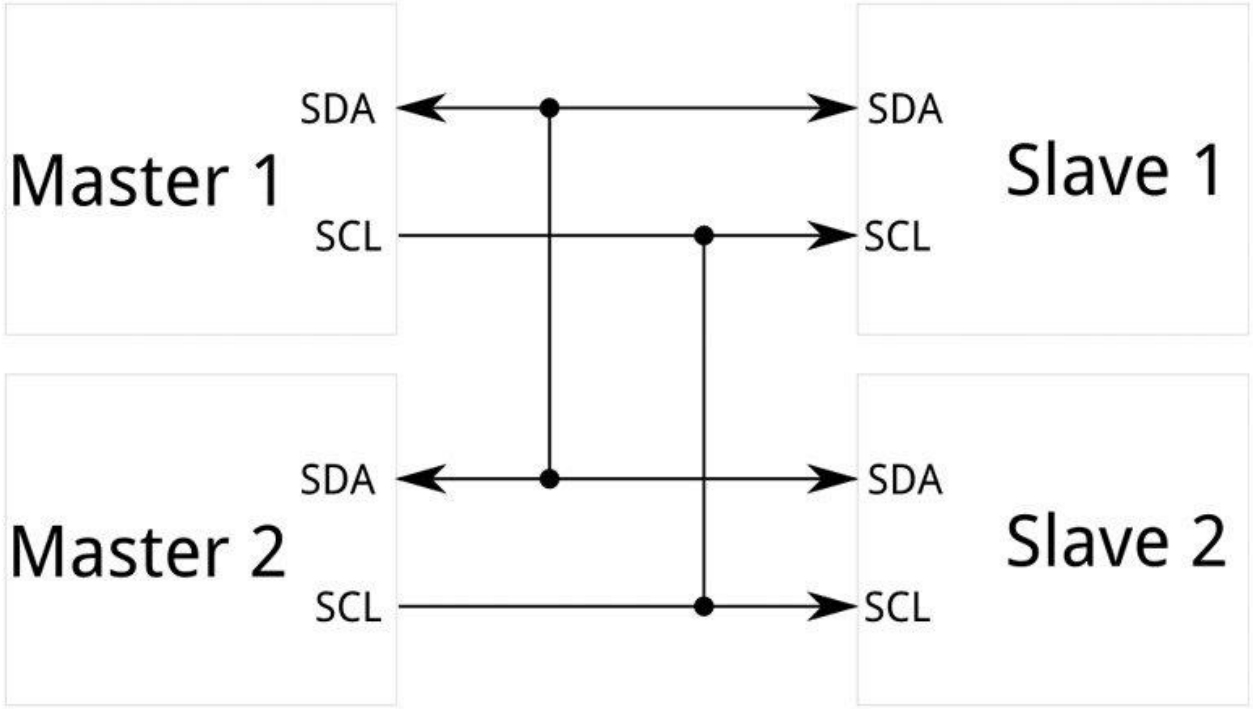
Serial Interface I2C

Топология 1

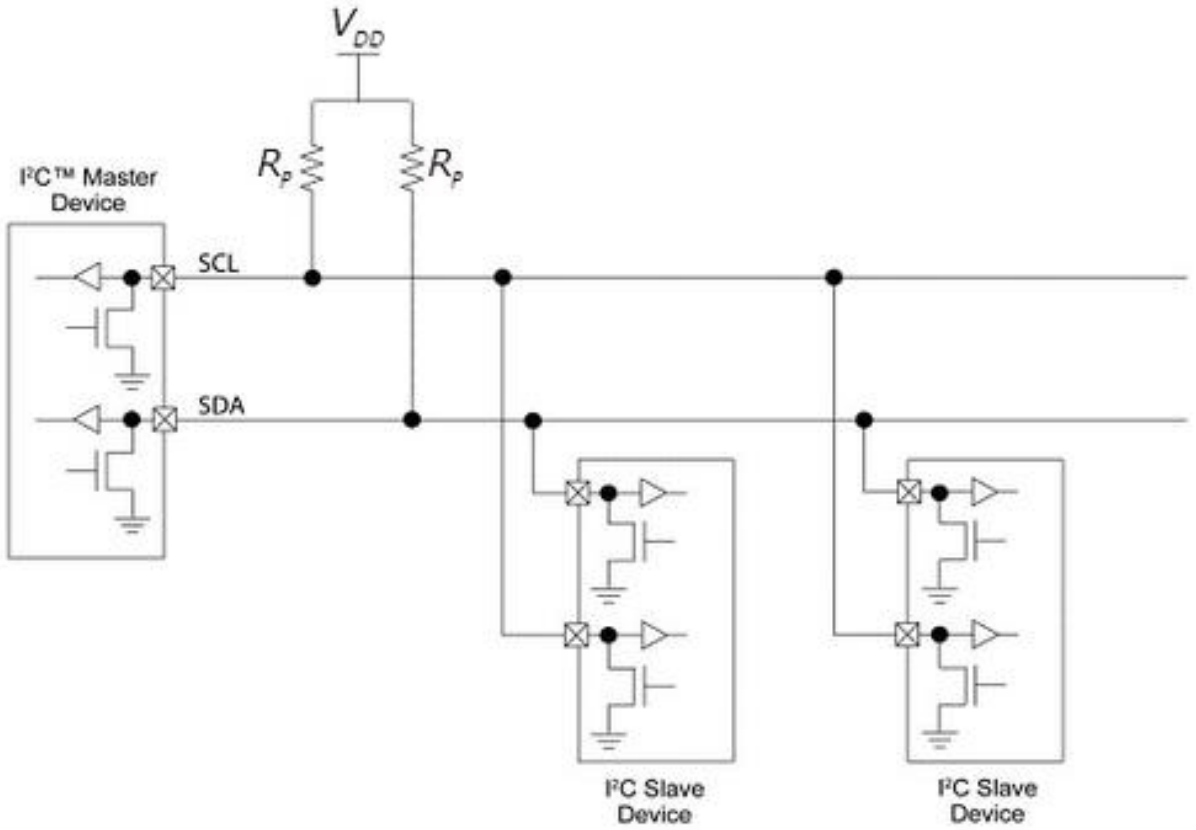


Serial Interface I2C

Топология 2

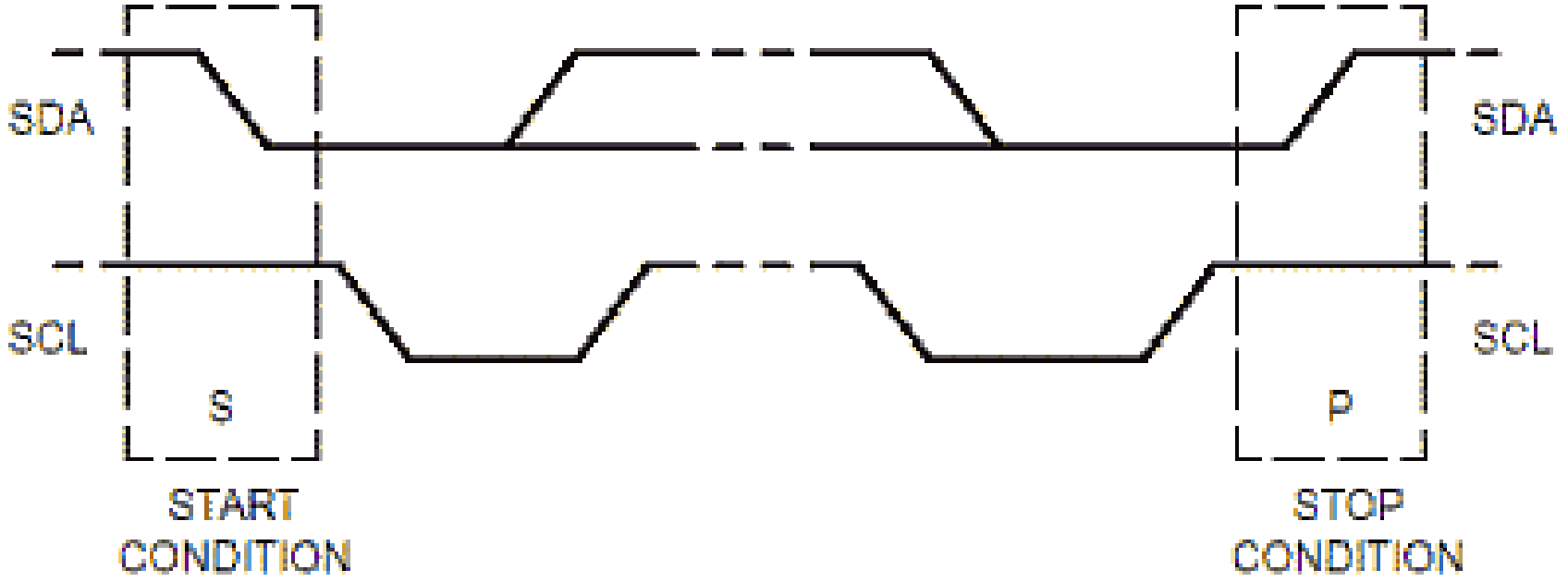


Serial Interface I2C



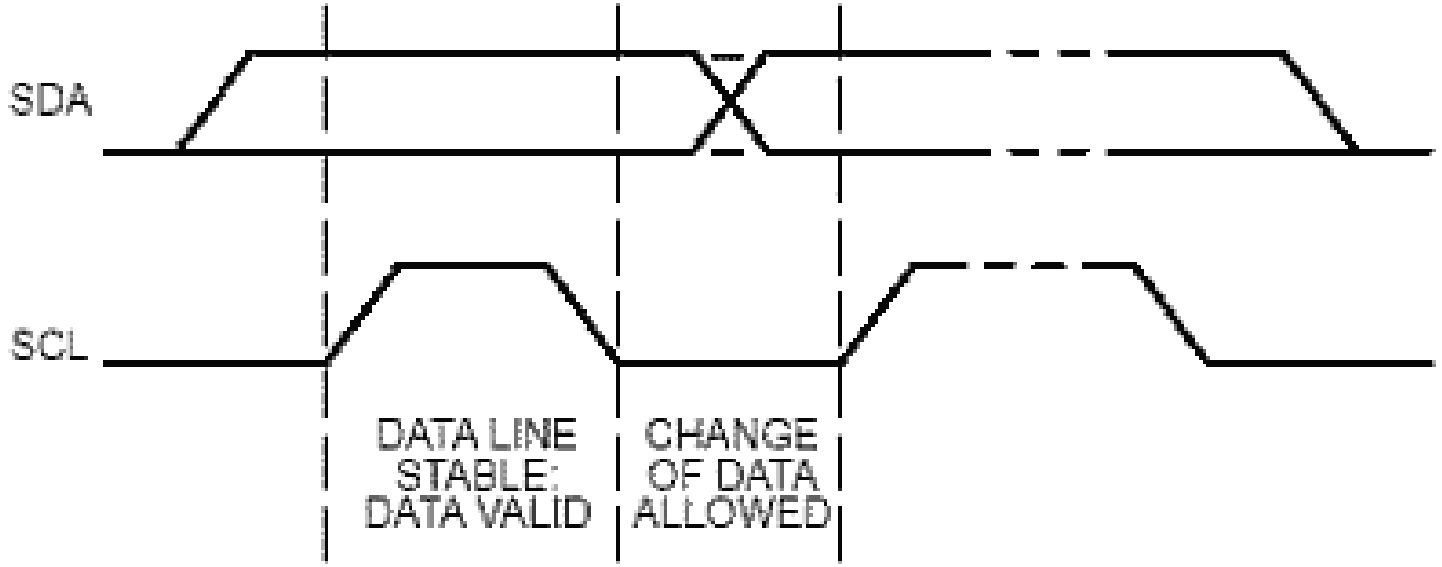
Serial Interface I2C

Состояния



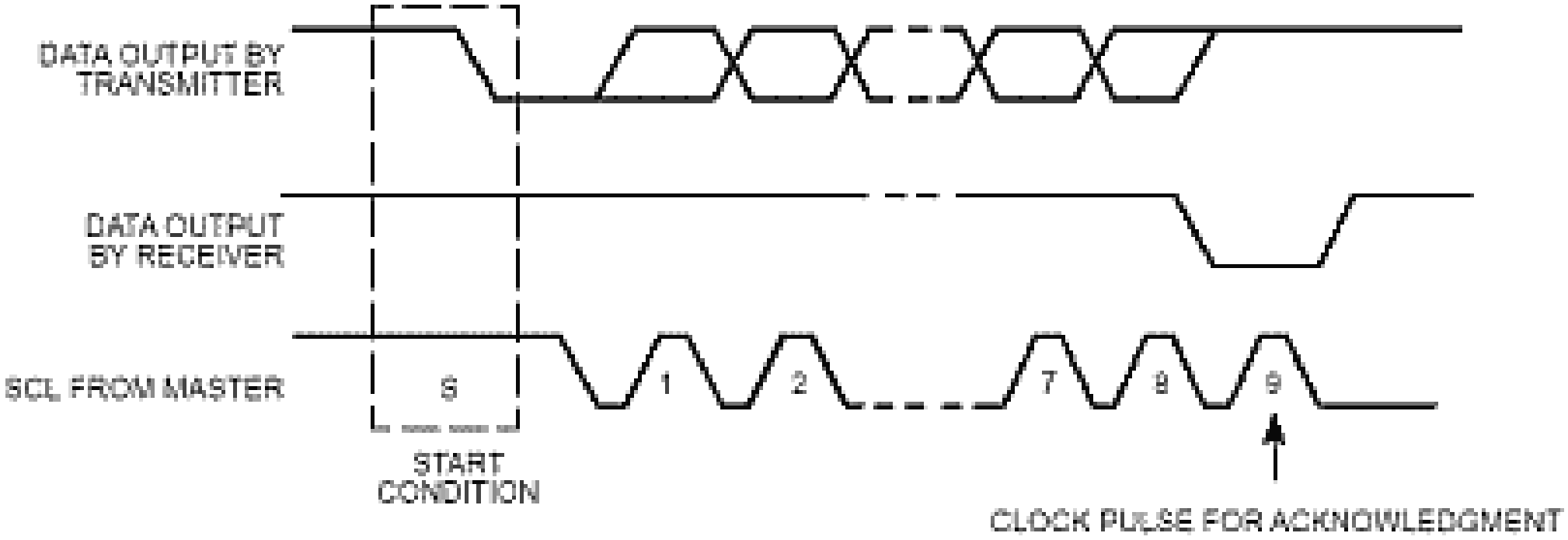
Serial Interface I2C

Передача данных



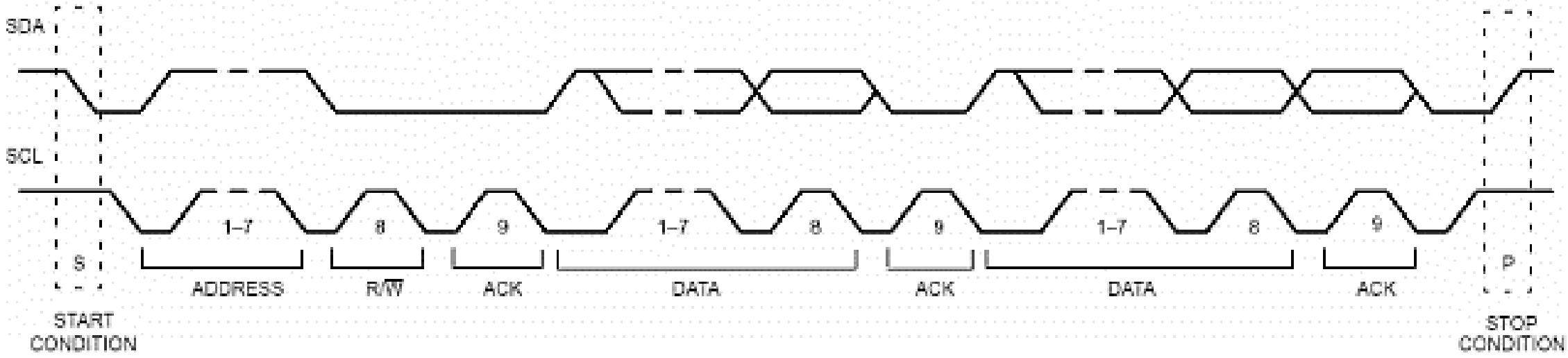
Serial Interface I2C

Подтверждение



Serial Interface I2C

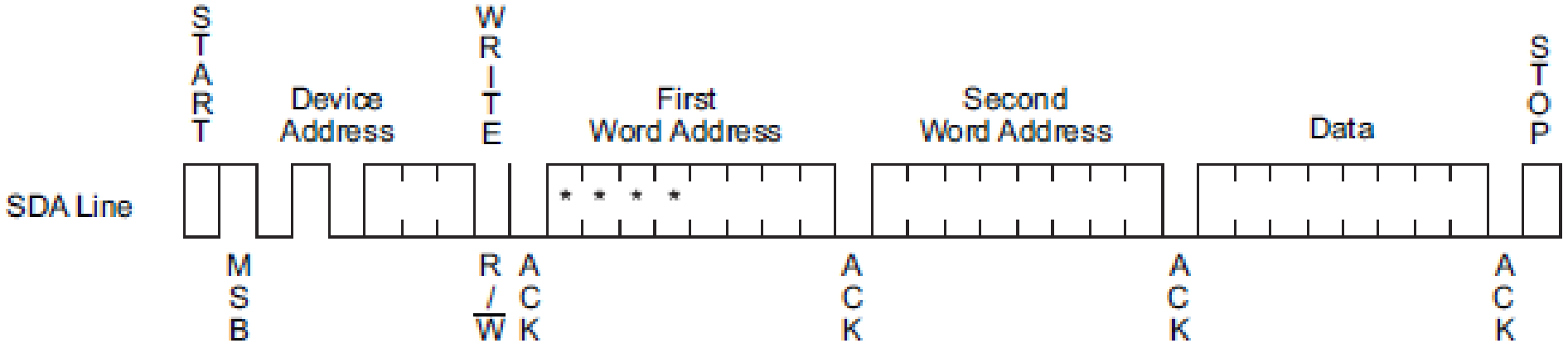
Сеанс обмена



Serial Interface I2C

Формат обмена

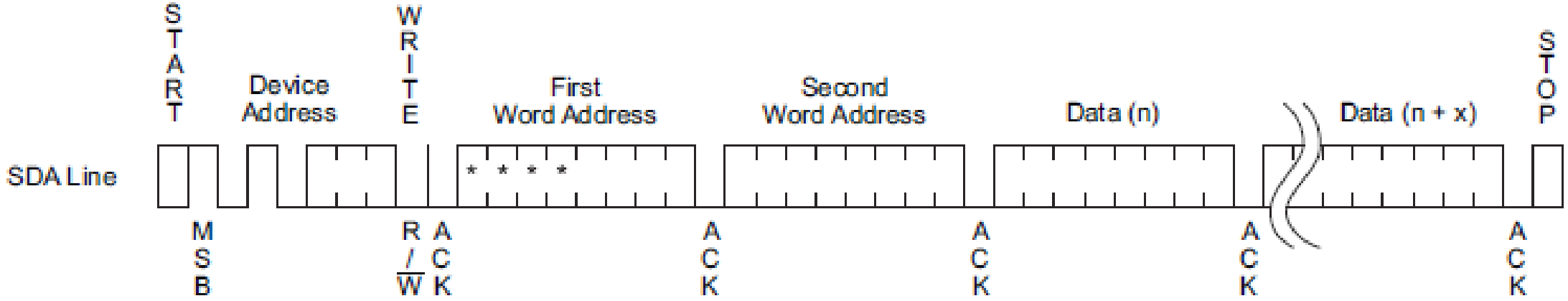
Figure 8-1. Byte Write



Serial Interface I2C

Формат обмена

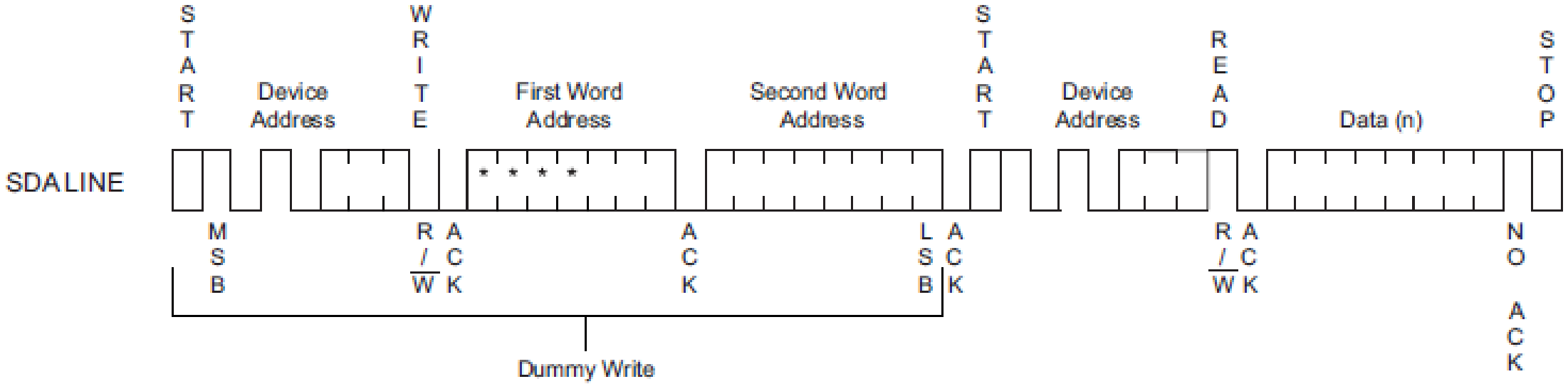
Figure 8-2. Page Write



Serial Interface I2C

Формат обмена

Figure 9-2. Random Read



Note: * = Don't care bit.

Serial Interface I2S

Integrated **I**nter-chip **S**ound

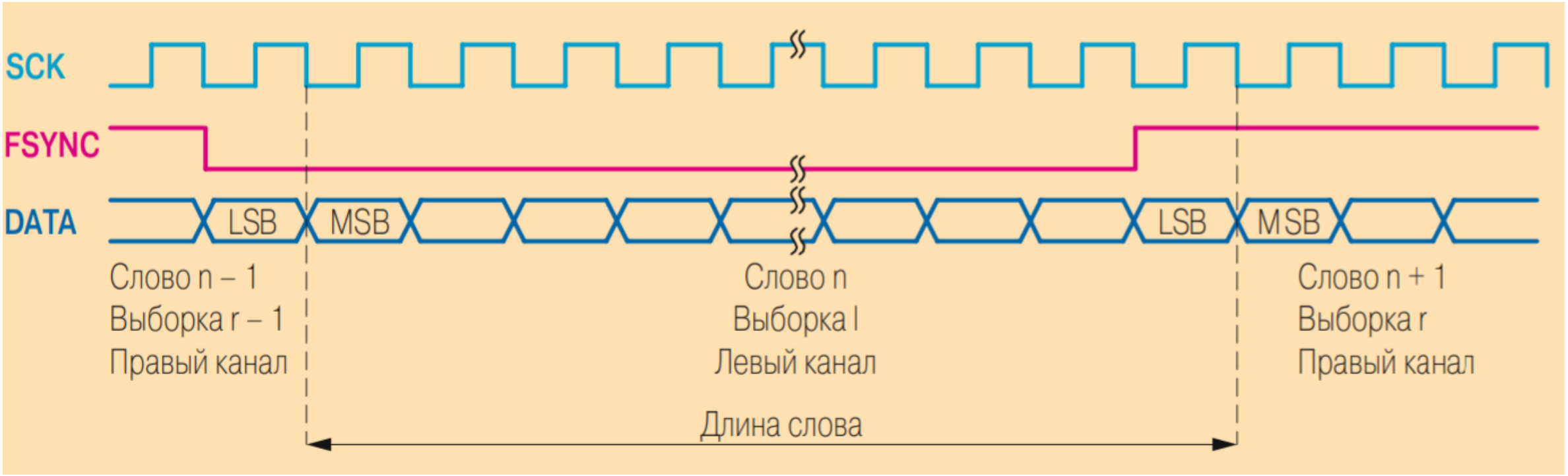
- SCK (CLK) - тактовый сигнал битовой синхронизации
- FSYNC (WS) - тактовый сигнал фреймовой (по словам) синхронизации
- DATA (SD) - сигнал данных, который может передавать или принимать 2 разделённых по времени канала

Serial Interface I2S

Integrated **I**nter-chip **S**ound

- синхронный
- полудуплекс
- топология – “устройство - устройство”
- разрядность данных – 16, 24, 32
- Тактовая частота – от 6,75 кГц до 400 кГц

Serial Interface I2S



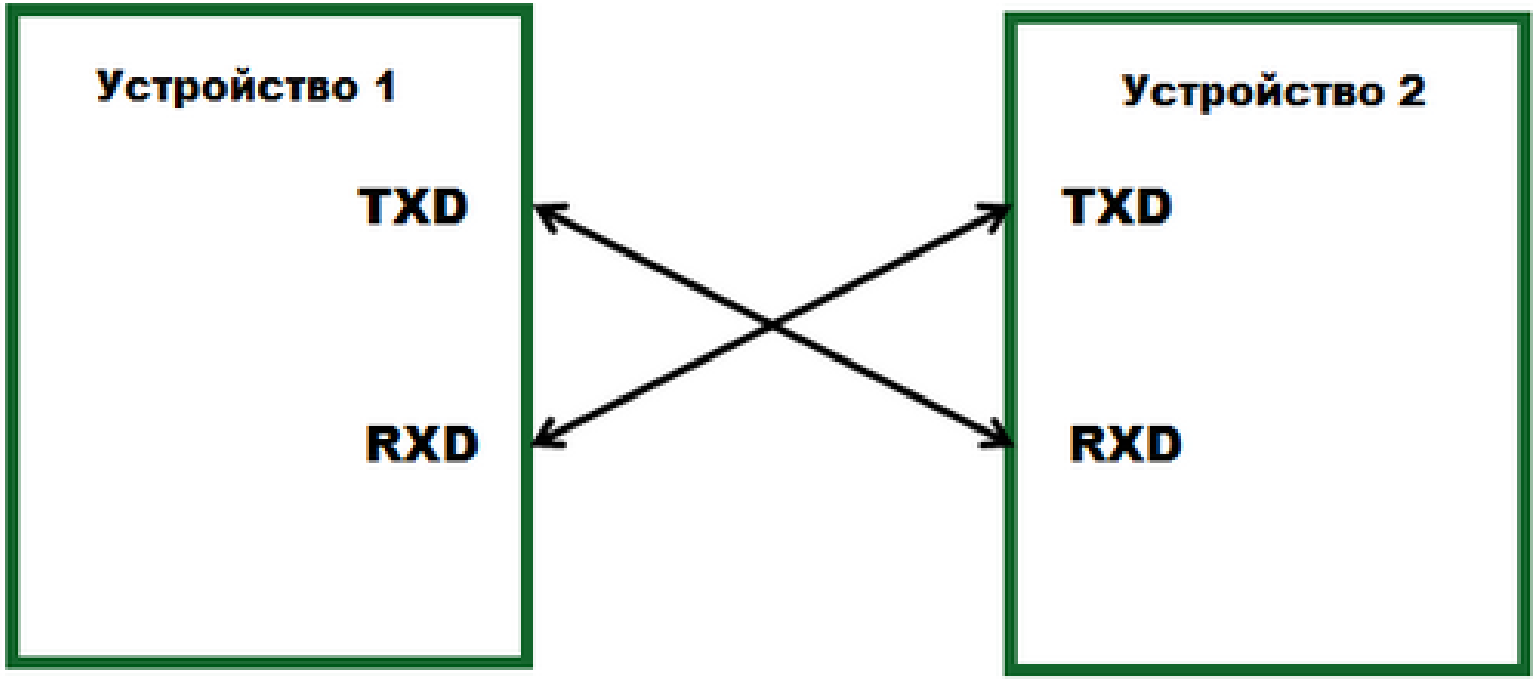
Serial Interface UART

Universal **A**synchronous **R**eceiver-**T**ransmitter

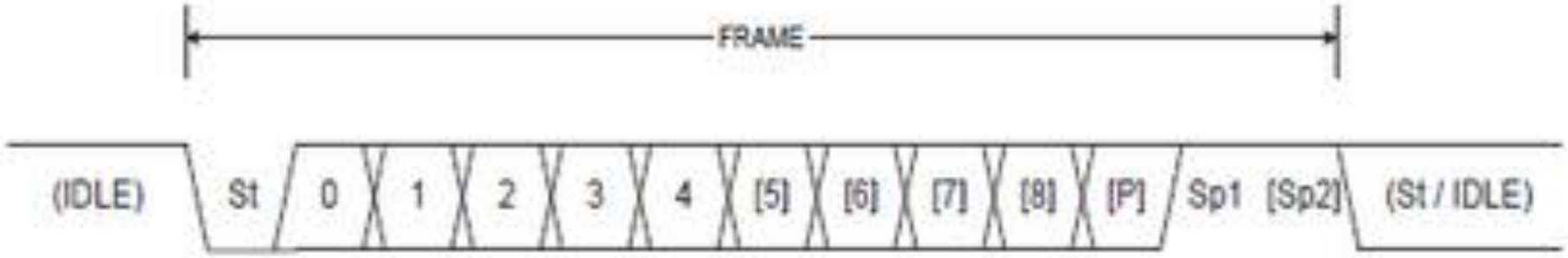
- асинхронный
- полный дуплекс
- топология – “устройство - устройство”
- разрядность данных – 8 (5-9)
- скорость передачи – до 115,2 кБод (921,6 кБод)

Serial Interface UART

Топология



Serial Interface UART



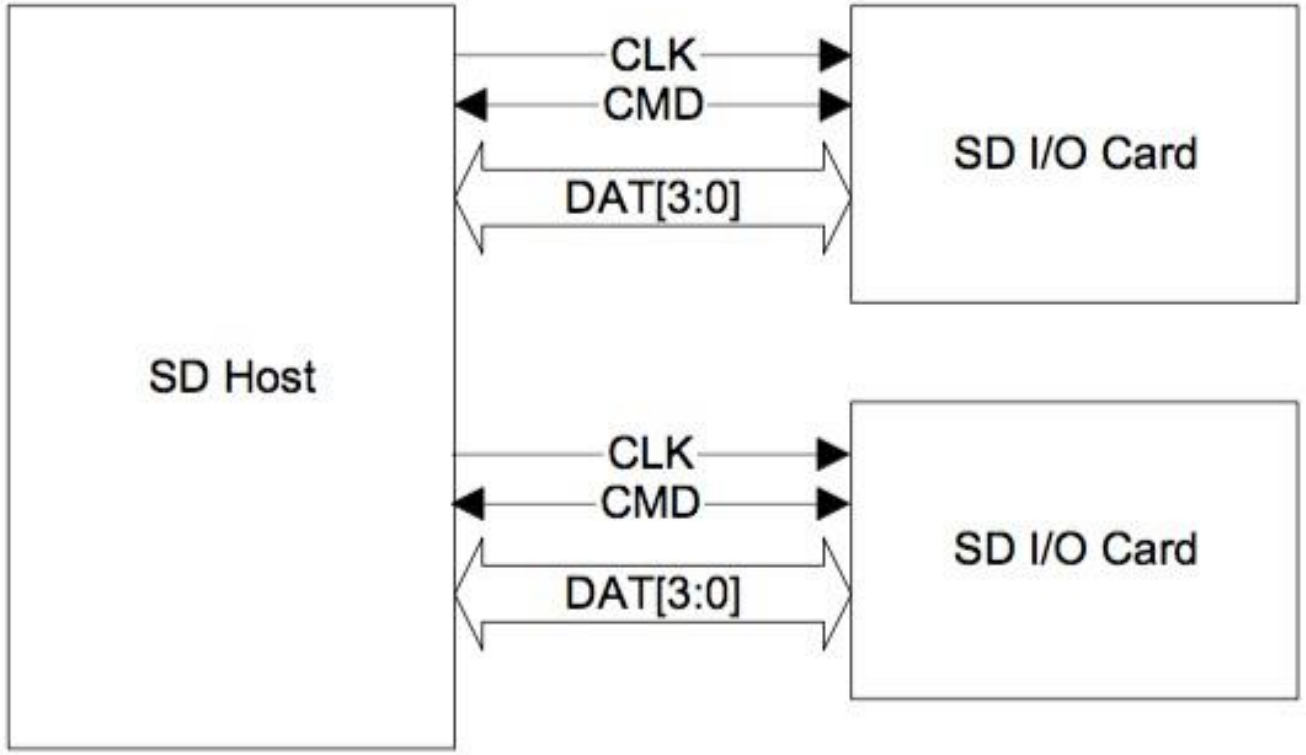
St	Start bit, always low
(n)	Data bits (0 to 8)
P	Parity bit. Can be odd or even
Sp	Stop bit, always high
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high

Serial Interface SDIO

Secure Digital Input Output:

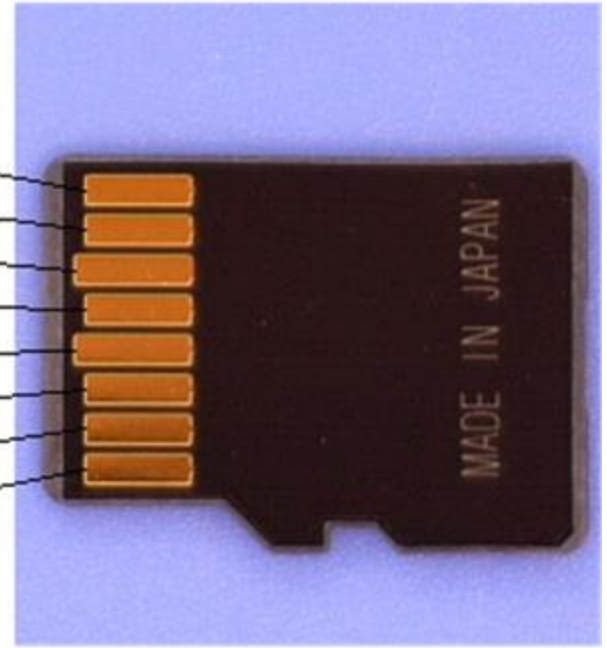
- GPS
- Camera
- Wi-Fi
- Ethernet
- Barcode readers
- Bluetooth

Serial Interface SDIO

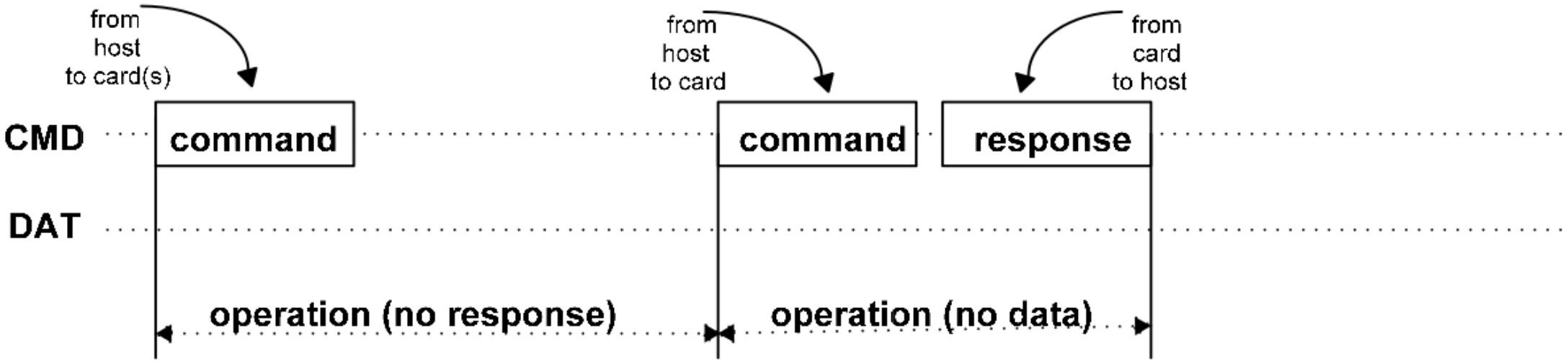


microSD

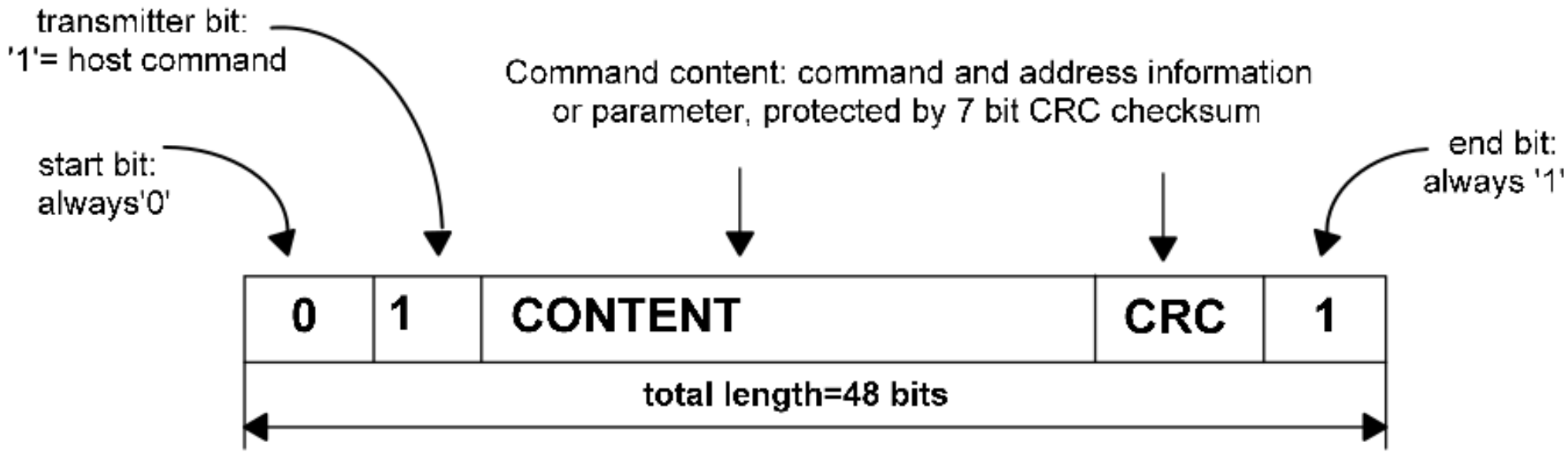
No	SD	SPI
8	DAT1	
7	DAT0	DO
6	Vss	
5	CLK	SCLK
4	Vdd	
3	CMD	DI
2	DAT3	CS
1	DAT2	



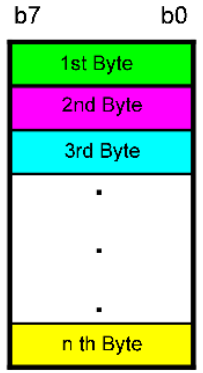
Serial Interface SDIO



Serial Interface SDIO



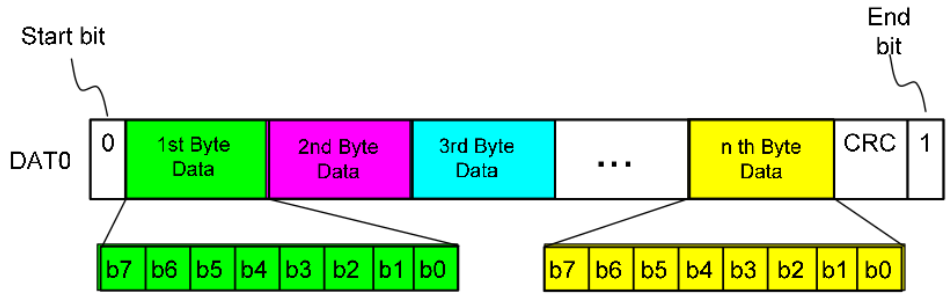
Serial Interface SDIO



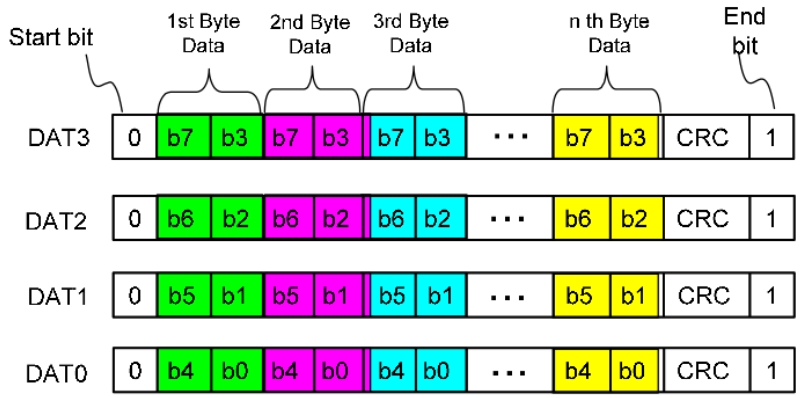
8bit width Data

Ex

[SDIO]
CMD53
[SD memory]
CMD17, CMD18,
CMD24, CMD25,
ACMD18, ACMD25,
etc



Data Packet Format for Standard Bus (only DAT0 used)



Data Packet Format for Wide Bus (all four lines used)

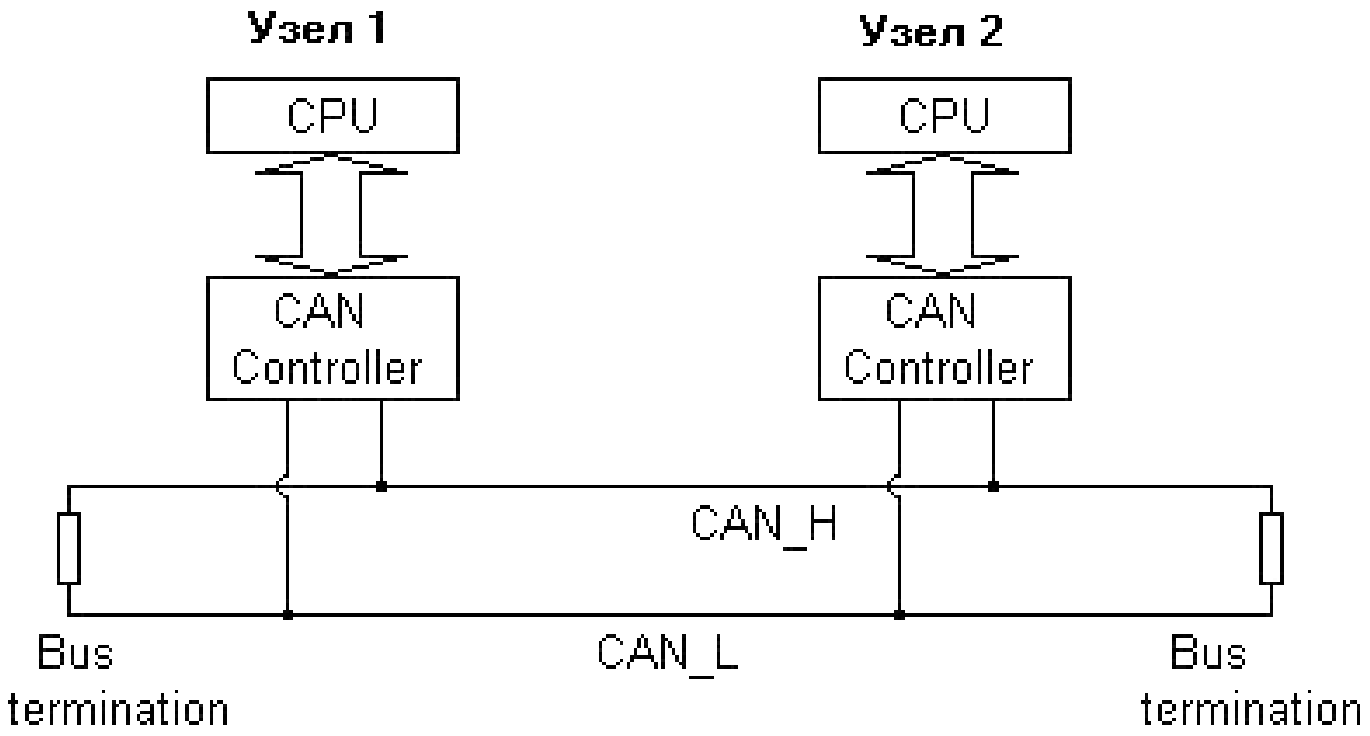
Serial Interface CAN

Controller **A**rea **N**etwork (разработчик – Philips)

- асинхронный
- полудуплекс
- топология – общая шина
- разрядность данных – 8
- скорость передачи – до 1 МБит/с
- расстояние до 40м (1 МБит/с), до 5 км (10 кБит/с)

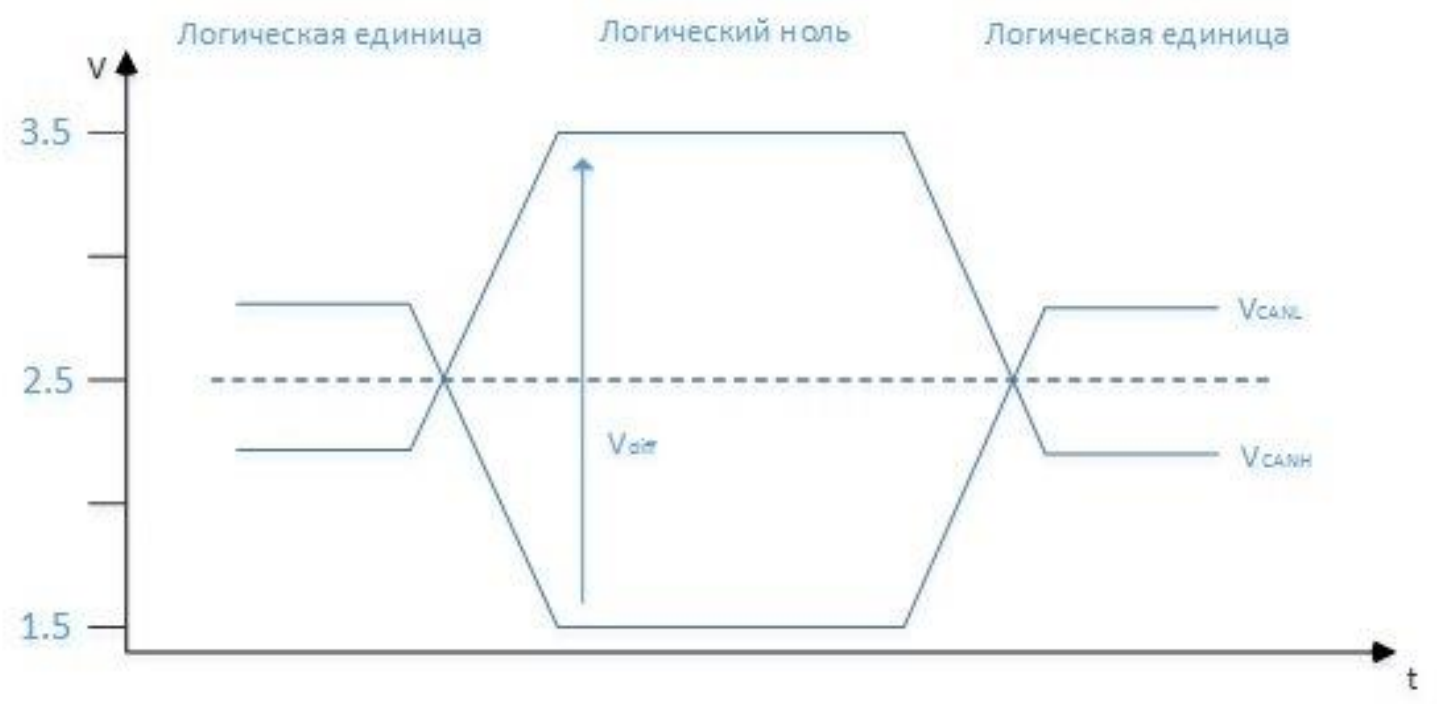
Serial Interface CAN

Топология



Serial Interface CAN

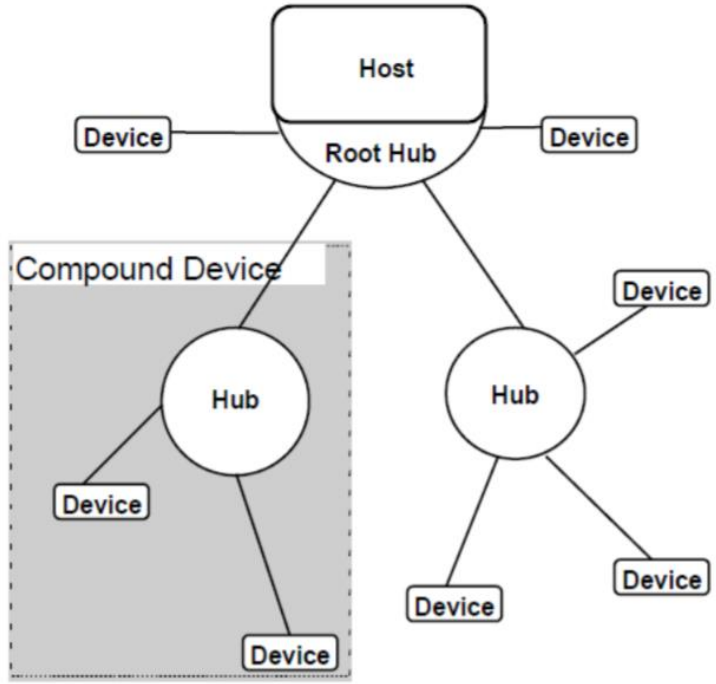
Уровни сигналов



Serial Interface USB

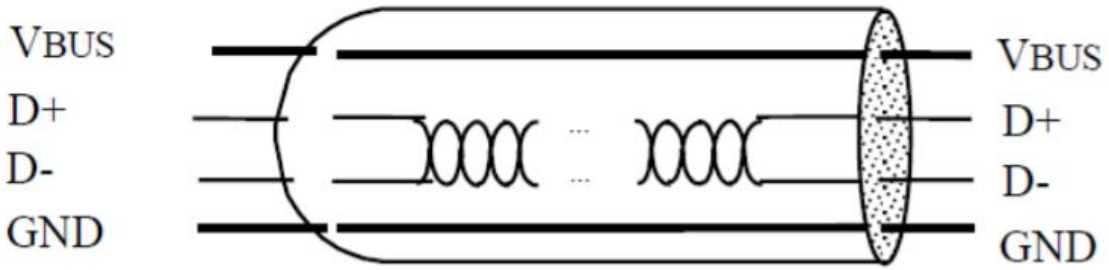
Universal **S**erial **B**us

- асинхронный
- полудуплекс
- топология – активное дерево
- количество устройств – до 127
- расстояние до 5м
- скорость передачи – 1,5(LS), 12(FS), 480(HS), 5000 (SS) 10000(SS+) МБит/с



Serial Interface USB

Pinout



USB 1.x/2.0 standard pinout

Pin	Name	Wire color	Description
1	V_{BUS}	Red (or Orange)	+5 V
2	D−	White (or Gold)	Data−
3	D+	Green	Data+
4	GND	Black (or Blue)	Ground

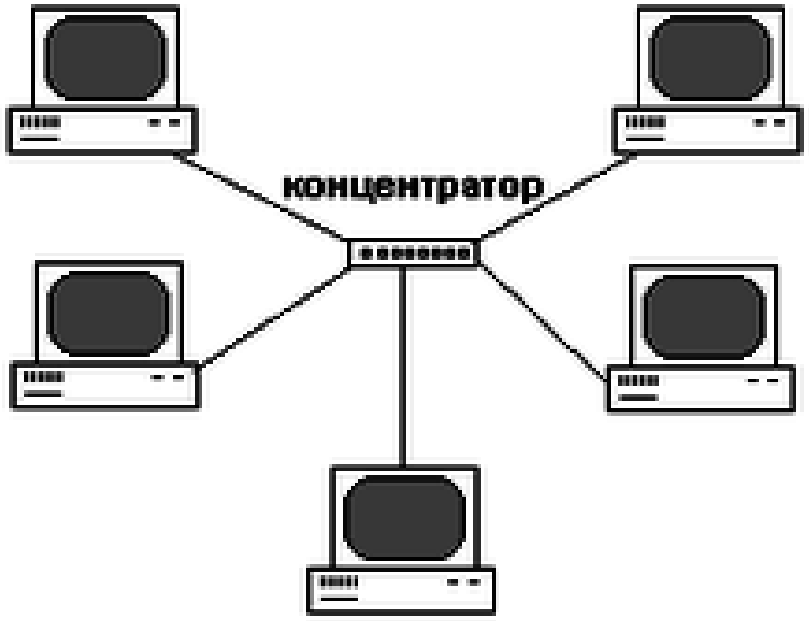
Serial Interface USB

Классы USB устройств:

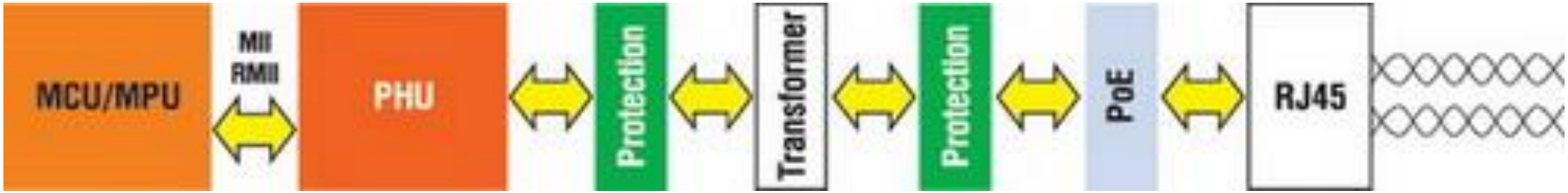
- Audio Device Class
- DFU - Download Firmware Update Class
- HID - Human Interface Device
- CDC - Communication Device Class
- MSC - Mass Storage Device (MSC - Mass Storage Class)
- CCID - Circuit Card Interface Device

Serial Interface Ethernet

- асинхронный
- полудуплекс
- топология – звезда
- скорость передачи – 10, 100 МБит/с



Serial Interface Ethernet



Parallel Interfaces

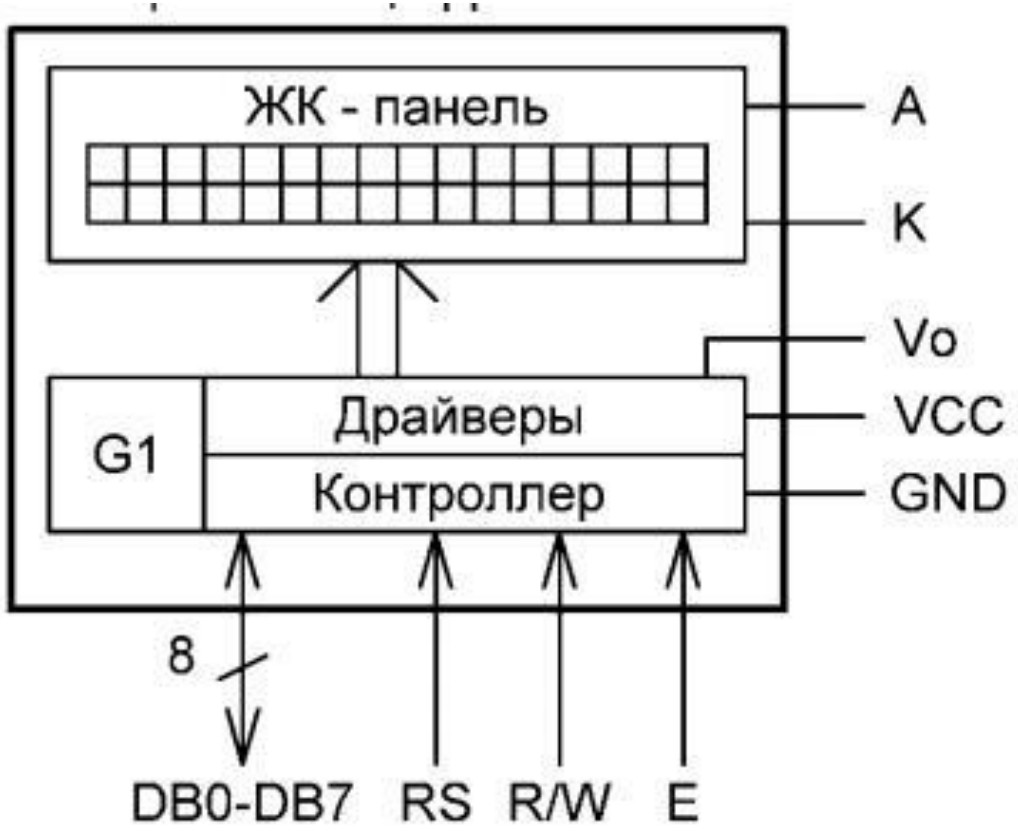
- HD44780
- FSMC

Parallel Interface HD44780

HD44780 (KS0066) - контроллер монохромных жидкокристаллических знакосинтезирующих дисплеев с параллельным 4- или 8-битным интерфейсом (Hitachi)

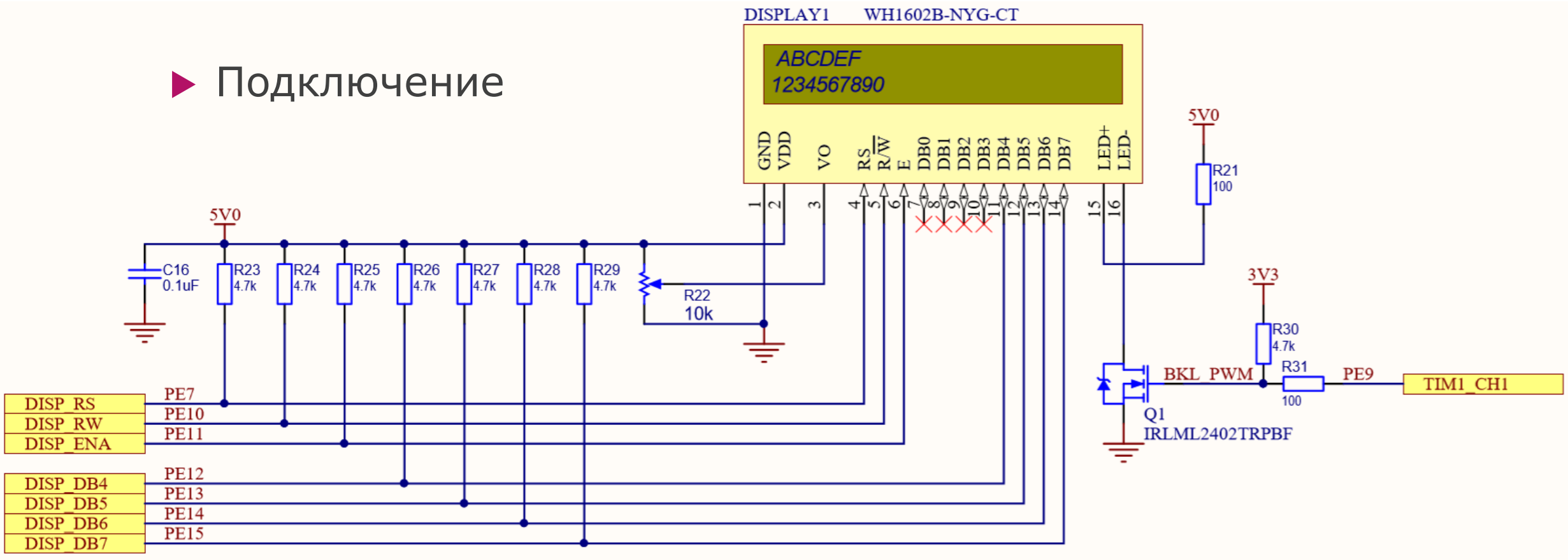
Parallel Interface HD44780

- Структура



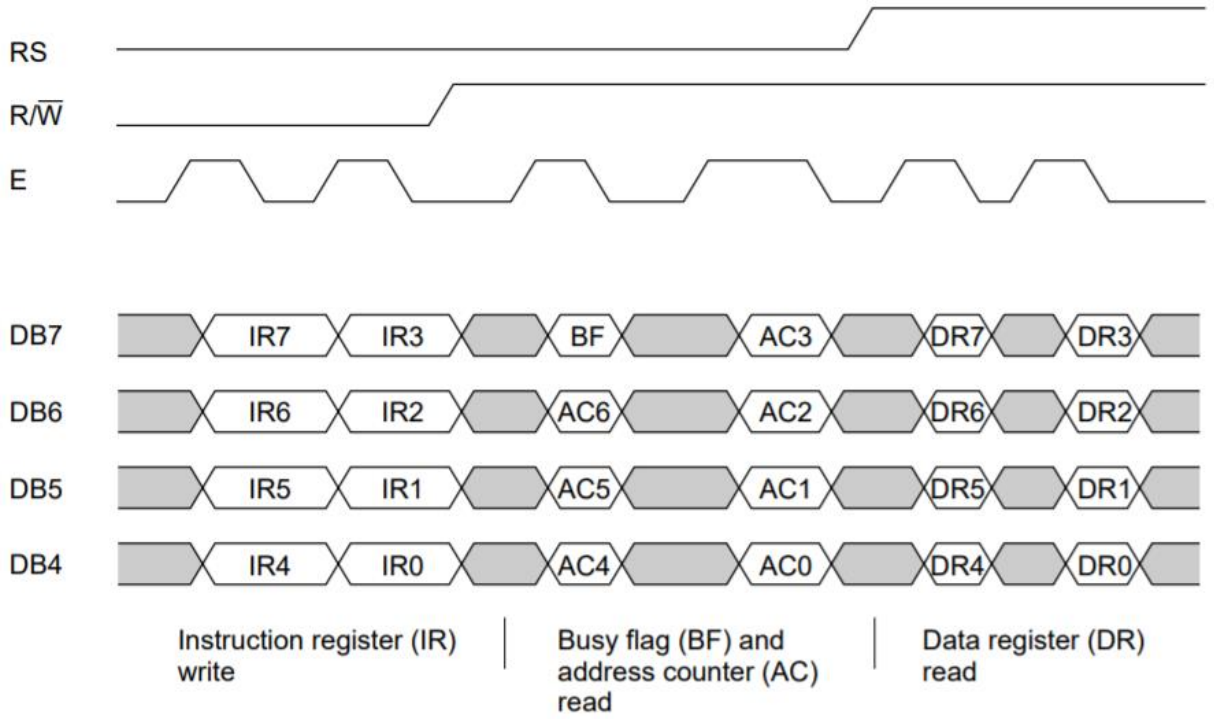
Parallel Interface HD44780

- Подключение



Parallel Interface HD44780

- 4-битный интерфейс



Parallel Interface HD44780

- Команды

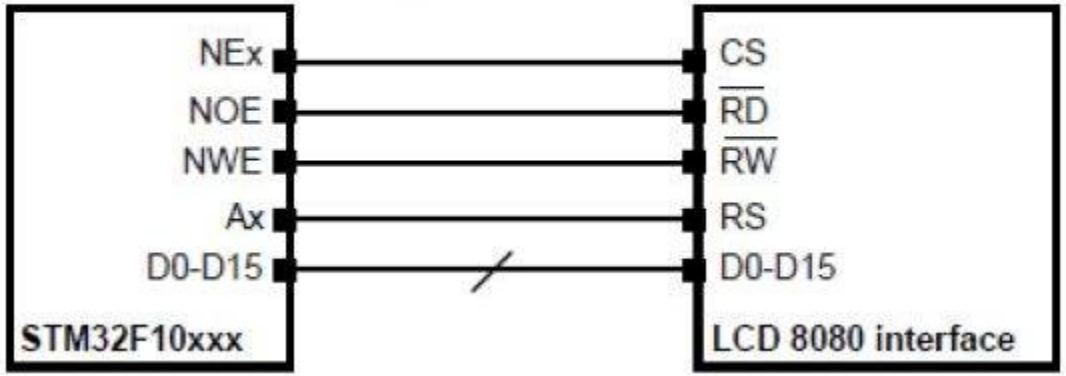
Команда	A0	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Описание	Время выпол-нения
Clear Display	0	0	0	0	0	0	0	0	0	1	Очищает модуль и помещает курсор в самую левую позицию	1,5 мс
Return Home	0	0	0	0	0	0	0	0	1	X	Перемещает курсор в левую позицию	40 мкс
Entry Mode Set	0	0	0	0	0	0	0	1	ID	SH	Установка направления сдвига курсора (ID=0/1—влево/вправо) и разрешение сдвига дисплея (SH=1) при записи в DDRAM	40 мкс
Display ON/OFF control	0	0	0	0	0	0	1	D	C	B	Включает модуль (D=1) и выбирает тип курсора (C, B), см. примечание 4	40 мкс
Cursor or Display Shift	0	0	0	0	0	1	SC	RL	X	X	Выполняет сдвиг дисплея или курсора (SC=0/1—курсор/дис-плей, RL=0/1—влево/вправо)	40 мкс

Parallel Interface HD44780

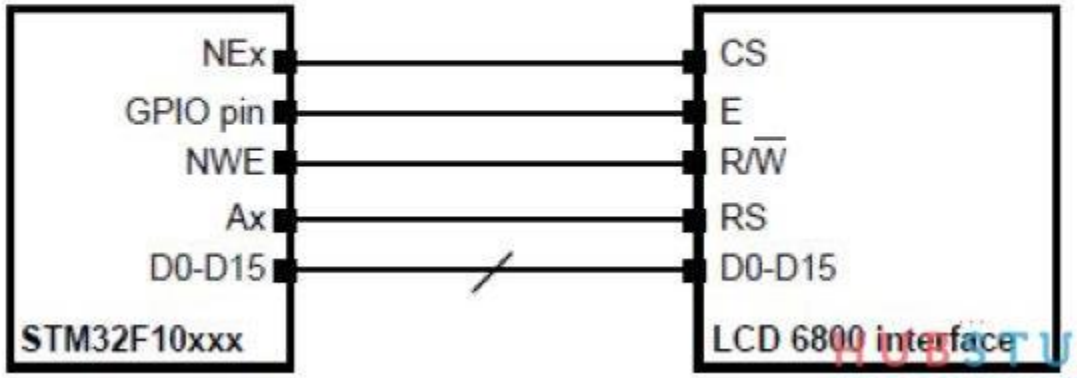
- Символы

Parallel Interface FSMC

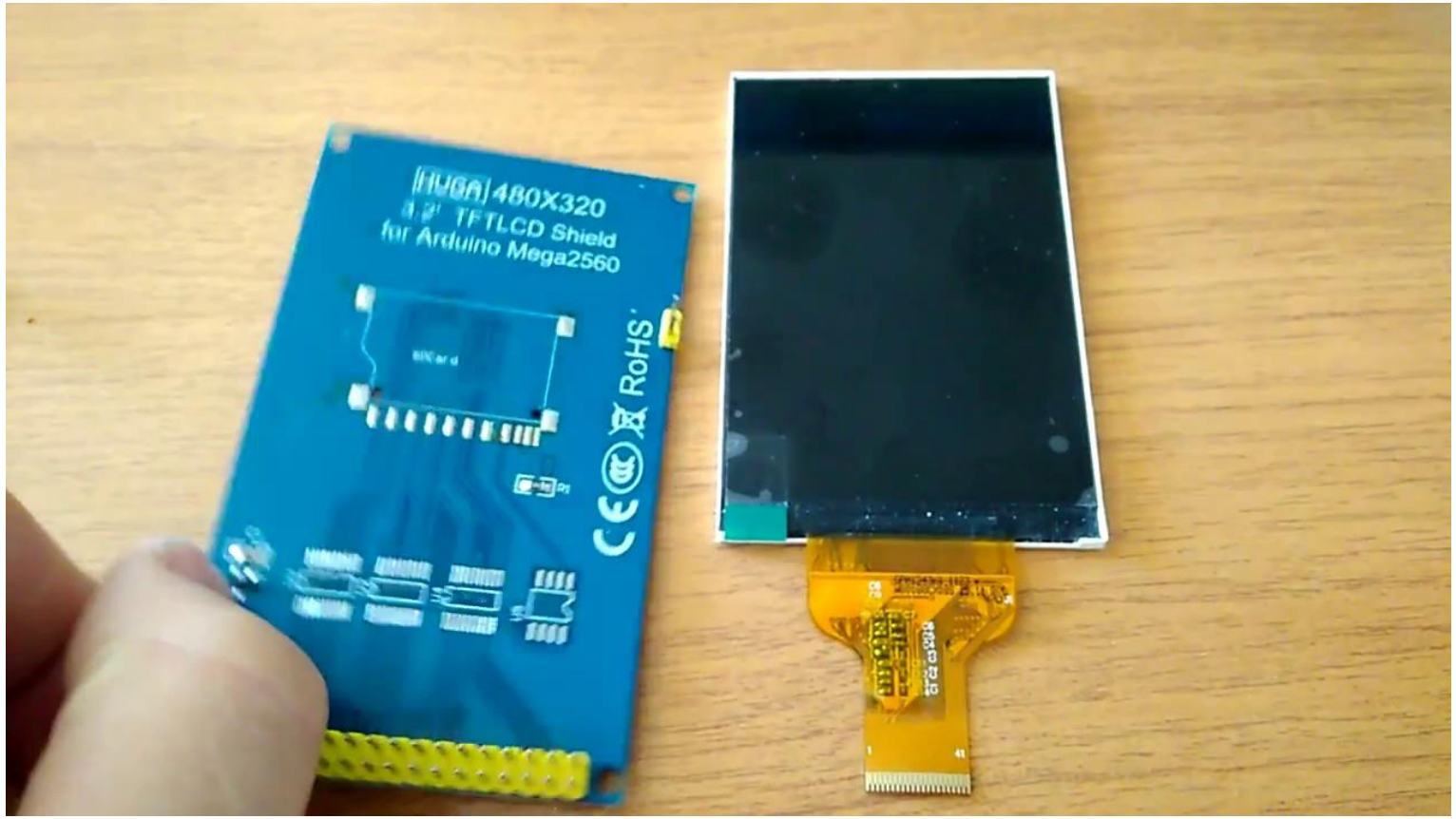
8080-й интерфейс



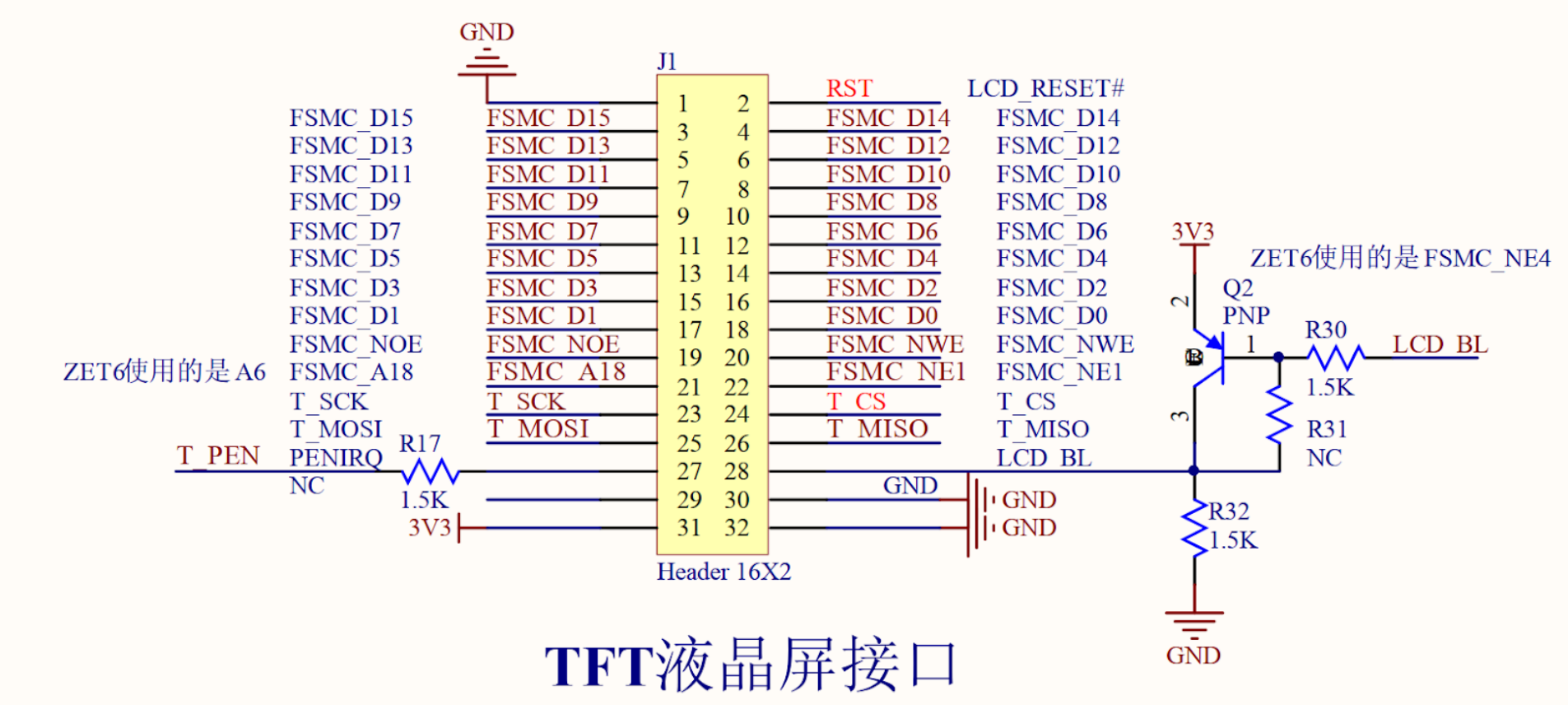
6800-й интерфейс



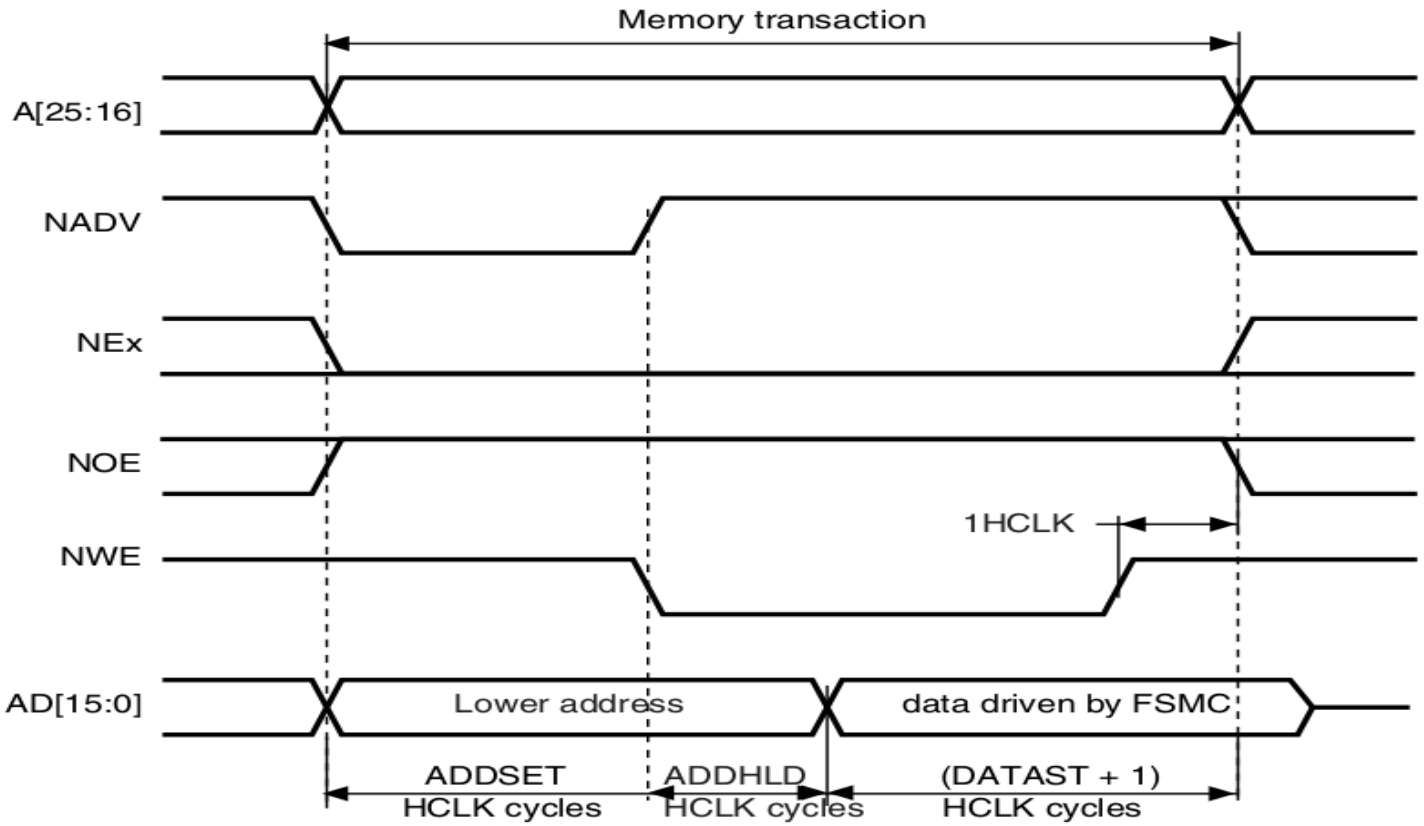
Parallel Interface FSMC



Parallel Interface FSMC



Parallel Interface FSMC



Parallel Interface FSMC

```
volatile uint16_t* fsmc = (uint16_t*)0x60000000;
uint16_t w[] = {
        0xFFFF, 0x0000, 0xFFFF, 0x0000,
        0xFFFF, 0x0000, 0xFFFF, 0x0000};

for(uint32_t i=0;i<8;i++) {
        fsmc[0] = w[i];
}
```