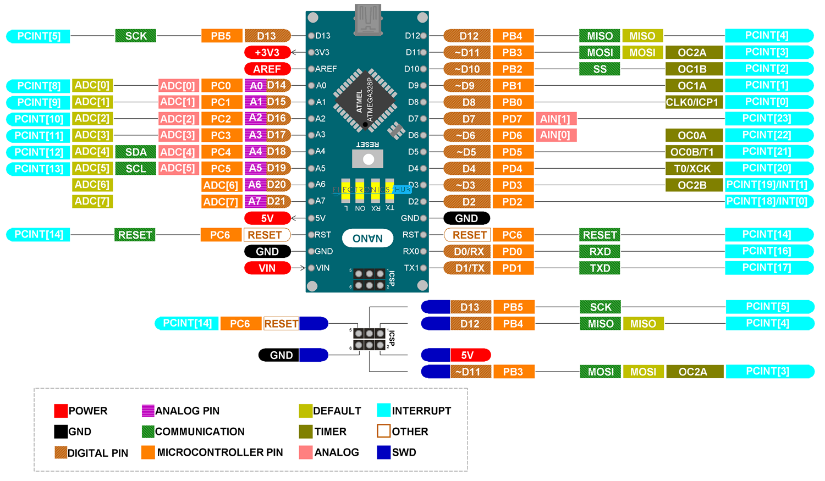
**Распиновка NANO на 328P**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № пина | Название | № пина на МК | Описание | Альтернативные функции |
| 1 | D1 / TX TXD – PCINT[17] | 31 | Цифровой ввод-вывод, контакт 1 Последовательный контакт TX | Обычно используется как TX |
| 2 | D0 / RX  RXD – PCINT[16] | 30 | Цифровой ввод-вывод, контакт 0 Последовательный контакт RX | Обычно используется как RX |
| 3 | RESET  PC6 – PCINT[14] | 29 | Сброс (Активный НИЗКИЙ) |  |
| 4 | GND | 3,5,21 | Земля |  |
| 5 | D2 / PD2  PCINT[18] / INT[0] | 32 | Цифровой ввод-вывод, контакт 2 |  |
| 6 | D3 / PD3  OC2B – PCINT[19] / INT[1] | 1 | Цифровой ввод-вывод, контакт 3 | Таймер (OC2B) |
| 7 | D4 / PD4  T0 / XCK – PCINT[20] | 2 | Цифровой ввод-вывод, контакт 4 | Таймер (T0/XCK) |
| 8 | D5 / PD5  OS0B / T1 – PCINT[21] | 9 | Цифровой ввод-вывод, контакт 5 | Таймер (OC0B/T1) |
| 9 | D6 / PD6  AIN[0] – OC0A / PCINT[22] | 10 | Цифровой ввод-вывод, контакт 6 |  |
| 10 | D7 / PD7  AIN[1] – PCINT[23] | 11 | Цифровой ввод-вывод, контакт 7 |  |
| 11 | D8 / PB0  CLK0 / ICP1 – PCINT[0] | 12 | Цифровой ввод-вывод, контакт 8 | Таймер (CLK0/ICP1) |
| 12 | D9 / PB1  OC1A / PCINT[1] | 13 | Цифровой ввод-вывод, контакт 9 | Таймер (OC1A) |
| 13 | D10 / PB2  SS – OC1B / PCINT[2] | 14 | Цифровой ввод-вывод, контакт 10 | Таймер (OC1B) |
| 14 | D11 / PB3  MOSI – OC2A / PCINT[3] | 15 | Цифровой ввод-вывод, контакт 11 | SPI (MOSI) и таймер (OC2A) |
| 15 | D12 / PB4  MISO – PCINT[4] | 16 | Цифровой ввод-вывод, контакт 12 | SPI (MISO) |
| 16 | D13 / PB5  SCK – PCINT[5] | 17 | Цифровой ввод-вывод, контакт 13 | SPI (SCK) |
| 17 | 3V3 | 4, 6 | Питание |  |
| 18 | AREF | 20 | Опорное напряжение |  |
| 19 | A0 / D14 -- PC0  ADC[0] – PCINT[8] | 23 | Аналоговый вход 0 |  |
| 20 | A1 / D15 – PC1  ADC[1] – PCINT[9] |  | Аналоговый вход 1 |  |
| 21 | A2 / D16 -- PC2  ADC[2] – PCINT[10] |  | Аналоговый вход 2 |  |
| 22 | A3 / D17 – PC3  ADC[3] – PCINT[11] |  | Аналоговый вход 3 |  |
| 23 | A4 / D18 – PC4  ADC[4] – SDA / PCINT[12] |  | Аналоговый вход 4 | I2C (SDA) |
| 24 | A5 / D19 – PC5  ADC[5] – SCL / PCINT[13] |  | Аналоговый вход 5 | I2C (SCL) |
| 25 | A6 / D20  ADC[6] – PCINT[??] |  | Аналоговый вход 6 |  |
| 26 | A7 / D21  ADC[7] – PCINT[??] |  | Аналоговый вход 7 |  |
| 27 | 5V | -- | Выход +5В от регулятора или регулируемый вход +5В |  |
| 28 | PC6 / RESET – PCINT[14] | 29 | Сброс (Активный НИЗКИЙ) |  |
| 29 | GND | 3, 5, 21 | ЗЕМЛЯ (-) |  |
| 30 | VIN |  | Нерегулируемое питание от 6 до 20 вольт |  |

Контакты разъема ICSP.

|  |  |
| --- | --- |
| MISO | Master In Slave Out (Input or Output) |
| 5V | Supply |
| SCK | Clock (from Master to Slave) |
| MOSI | Master Out Slave In (Input or Output) |
| RESET | Reset (Active LOW |
| GND | Ground |

Из 30 контактов, доступных на плате Nano, 22 контакта связаны с вводом и выводом. При этом 14 контактов (от D0 до D13) являются настоящими цифровыми выводами ввода-вывода, которые можно настроить в соответствии с вашим приложением с помощью функций pinMode(), digitalWrite() и digitalRead().

Все эти контакты цифрового ввода-вывода способны подавать или потреблять ток до 40 мА. Дополнительной особенностью выводов Digital IO является наличие внутреннего подтягивающего резистора (который по умолчанию не подключен). Значение внутреннего подтягивающего резистора будет находиться в диапазоне от 20 кОм до 50 кОм. Для включения подтягивающего резистора необходимо указать:

*pinMode () Синтаксис функции*

Void setup () {

pinMode (pin , mode);

}

*где \*pin – номер пина;*

*\*mode - режим –( INPUT, OUTPUT или INPUT\_PULLUP)*

*Пример:*

int button = 5 ; // кнопка подключена к контакту 5

int LED = 6; // светодиод подключен к контакту 6

void setup () {

pinMode(button , INPUT\_PULLUP);

// установите цифровой вывод в качестве входа с подтягивающим резистором

pinMode(button , OUTPUT); // установить цифровой пин в качестве выхода

}

Также имеется 8 контактов аналогового входа (от A0 до A7). Это на пару больше, чем у Arduino UNO (у которой их всего 6). Все аналоговые входные контакты обеспечивают функцию АЦП с 10-битным разрешением, которую можно считывать с помощью функции AnalogRead().

Какие коммуникационные интерфейсы доступны на Arduino Nano?

Arduino Nano поддерживает три различных типа коммуникационных интерфейсов. Они есть:

* Serial
* I2C or I2C
* SPI

Возможно, наиболее распространенным коммуникационным интерфейсом во вселенной Arduino является последовательная связь. На самом деле платы Arduino (UNO, Nano или Mega) программируются с использованием последовательной связи.

Контакты цифрового ввода-вывода 0 и 1 используются в качестве контактов Serial RX и TX для приема и передачи последовательных данных. Эти контакты подключены к последовательным контактам встроенной микросхемы преобразователя USB в последовательный порт.

Аналоговые входные контакты A4 и A5 имеют альтернативные функции. Их можно настроить как SDA (A4) и SCK (A5) для поддержки связи I2C или I2C или двухпроводного интерфейса (TWI).

Последним коммуникационным интерфейсом является SPI. Выводы цифрового ввода-вывода 10, 11, 12 и 13 могут быть сконфигурированы как выводы SPI SS, MOSI, MISO и SCK соответственно.

Какие-то дополнительные функции?

На плате есть встроенный светодиод, подключенный к контакту 13 цифрового ввода-вывода. Используйте этот светодиод для выполнения операций мигания. Опорное напряжение для внутреннего АЦП по умолчанию установлено на 5 В. Но с помощью вывода AREF можно вручную установить верхний предел АЦП.

Для сброса микроконтроллера можно использовать кнопку RESET на плате.

Хотя вы можете запрограммировать Arduino Nano с помощью USB-кабеля, существует возможность программирования MCU с использованием интерфейса внутрисхемного последовательного программирования (ICSP).

Загрузчик UART, предварительно загруженный в микроконтроллер ATmega328P, позволяет программировать через последовательный интерфейс. Но ICSP не нуждается в загрузчике. Вы можете запрограммировать Arduino nano с помощью ISCP или использовать ISCP Arduino Nano для программирования других плат Arduino.

Выводы цифрового ввода-вывода 2 и 3 могут быть сконфигурированы как выводы внешних прерываний INT0 и INT1 соответственно. Используйте функцию attachInterrupt(), чтобы настроить прерывание для переднего фронта, заднего фронта или изменения уровня на выводе.