

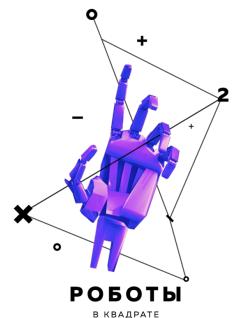
РОБОТЫ

В КВАДРАТЕ

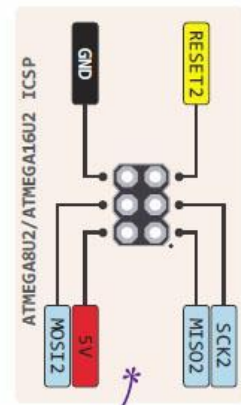
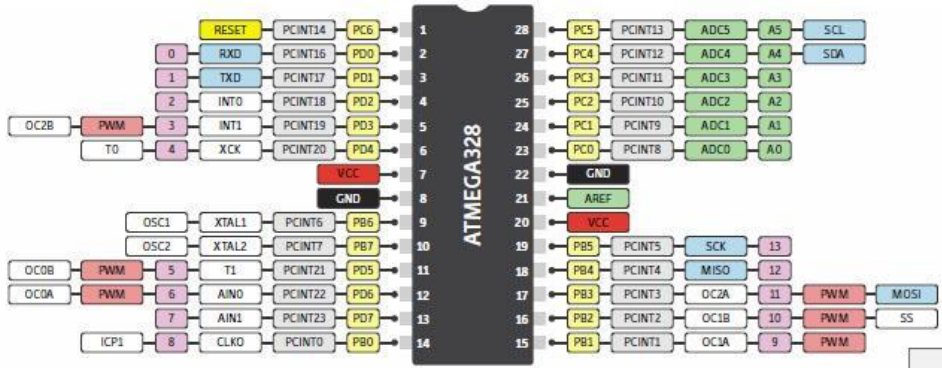


17.08.17

UART и создание функций в Arduino



THE DEFINITIVE ARDUINO UNO PINOUT DIAGRAM

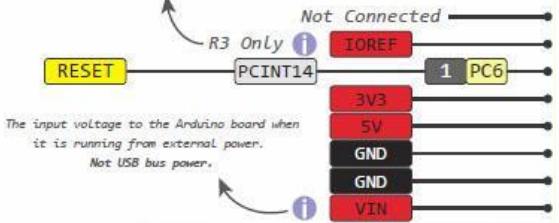


⚠ Absolute max per pin 40mA recommended 20mA
⚠ Absolute max 200mA for entire package

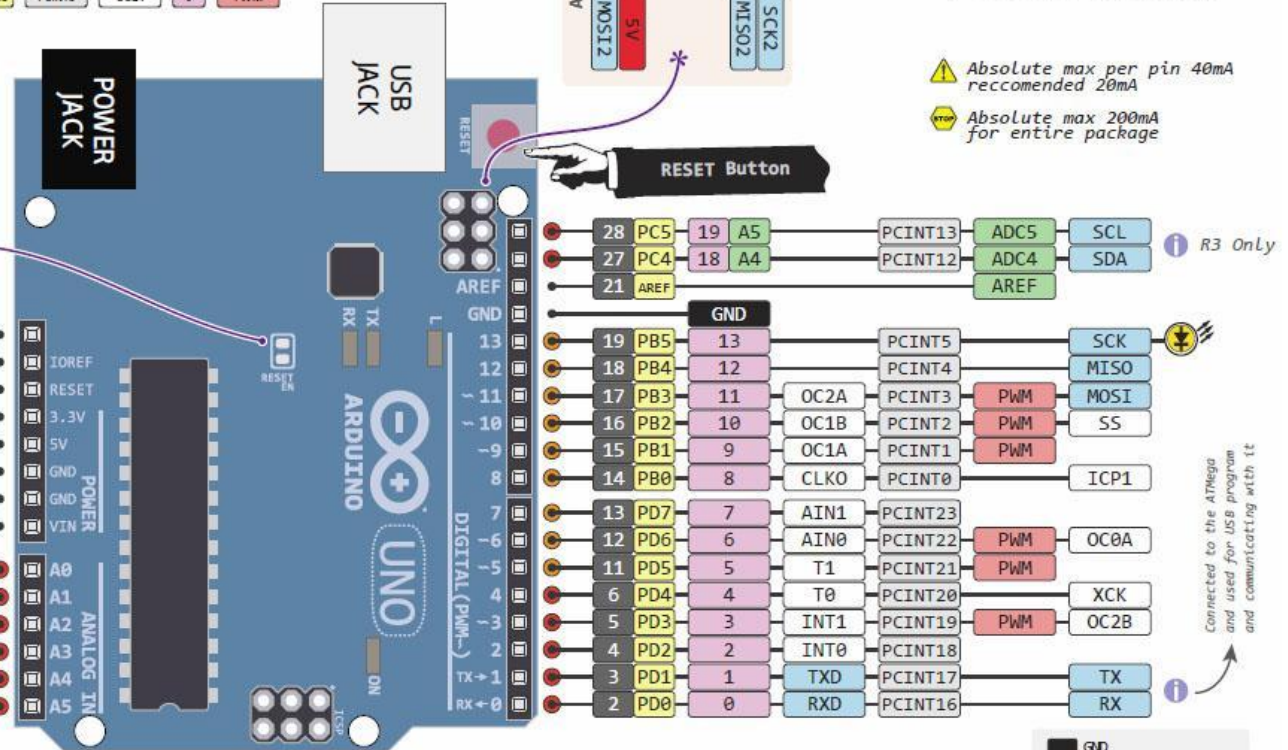


Cut to disable the auto-reset

This provides a logic reference voltage for shields that use it. It is connected to the 5V bus.



The input voltage to the Arduino board when it is running from external power. Not USB bus power.



ⓘ R3 Only



Connected to the ATmega and used for USB program and communicating with it

 	GND
 	Power
 	Control
 	Physical Pin
 	Port Pin
 	Pin Function
 	Digital Pin
 	Analog Related Pin
 	PWM Pin
 	Serial Pin
 	IDE
 	Source Total 150mA

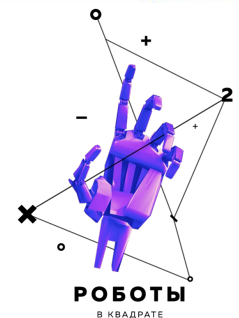


www.pighixx.com
 18 FEB 2013

ver 2 rev 2 - 05.03.2013

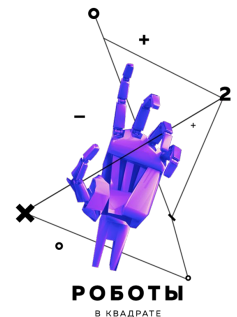
Виды последовательных интерфейсов

Метод	Преимущества	Недостатки	Реальная дальность
Двухпроводное соединение (I2C)	Просто, быстро, и к шине можно подключить несколько устройств.	С кабелем большой длины может не работать; медленнее, чем SPI в Arduino.	В пределах одного стола.
Трехпроводное соединение (SPI)	Работает на больших расстояниях по сравнению с I2C, легко написать код с помощью библиотеки SPI.	Требуется больше проводников, и для каждого подчиненного устройства нужен отдельный проводник.	В соседней комнате.
Последовательный порт (UART)	Простота, работает на довольно длинных расстояниях.	Медленный, может оказывать влияния на плату перепрограммирования; только одно соединение.	В гостиной.
RS485	Хорошие результаты и высокая скорость по длинному кабелю.	Нужны дополнительные устройства и хороший кабель; также загружает последовательную шину.	В пределах дома.
Беспроводное; соединение	Без проводов! Не так дорого, как вы думали - классно!	Более сложная настройка; может быть менее надежным, требуется много проводов.	Внизу в саду и дальше.



Описание интерфейса UART (Serial Port)

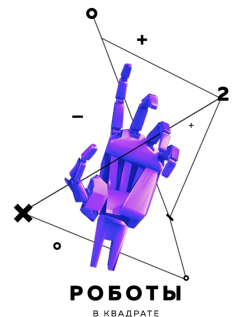
Набор функций Serial служит для связи устройства Ардуино с компьютером или другими устройствами, поддерживающими последовательный интерфейс обмена данными. Все платы Arduino имеют хотя бы один последовательный порт (UART, иногда называют USART). Для обмена данными Serial используют цифровые порты ввода/вывода 0 (RX) и 1 (TX), а также USB порт. Важно учитывать, что если вы используете функции Serial, то нельзя одновременно с этим использовать порты 0 и 1 для других целей.



Функции Serial

Среда разработки Arduino имеет встроенный монитор последовательного интерфейса (Serial monitor). Для начала обмена данными необходимо запустить монитор нажатием кнопки Serial monitor и выставить ту же скорость связи (baud rate), с которой вызвана функция `begin()`.

- [begin\(\)](#)
- [end\(\)](#)
- [available\(\)](#)
- [read\(\)](#)
- [flush\(\)](#)
- [print\(\)](#)
- [println\(\)](#)
- [write\(\)](#)
- [peek\(\)](#)

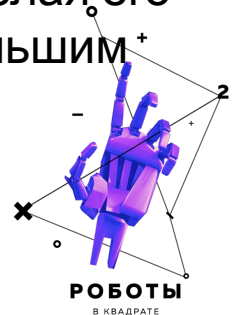


Создание функции в Arduino

Разбиение на сегменты кода функциями позволяет создавать части кода, которые выполняют определенные задания. После выполнения происходит возврат в место, откуда была вызвана функция. Причиной создания функции является необходимость выполнять одинаковое действие несколько раз.

Разделения кода на функции имеет ряд преимуществ:

- Функции позволяют организовать программу. Очень часто помогают заранее составить концепцию программы.
- Функции кодируют одно действие в одном месте программы. Далее необходимо только отладить код функции.
- Функции сокращают шансы на появление ошибки при необходимости изменения кода.
- Функции сокращают текст скетчей и делают его компактным, т.к. некоторые секции используются много раз.
- Функции облегчают использование кода в других программах делая его модульным. В этом случае функции обладают еще одним небольшим преимуществом, делая код программы легким для чтения.



Создание функции в Arduino

Пример

Синтаксис функции

тип возвращаемого значения
“void” если функция ничего
не возвращает

Параметры,
передаваемые
в функцию

Имя функции

```
int myMultiplyFunction(int x, int y){  
  int result;  
  result = x * y;  
  return result;  
}
```

Оператор, возвращающий
значение
соответствующего типа

Фигурные скобки, обязательны

Для вызова функции умножения ей передаются параметры данных:

```
void loop(){  
  int i = 2;  
  int j = 3;  
  int k;  
  
  k = myMultiplyFunction(i, j); // k содержит 6  
}
```

