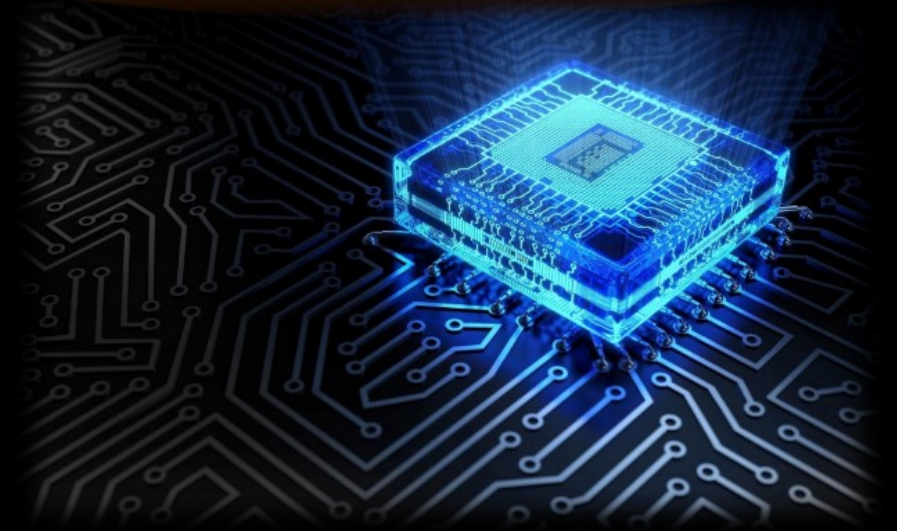
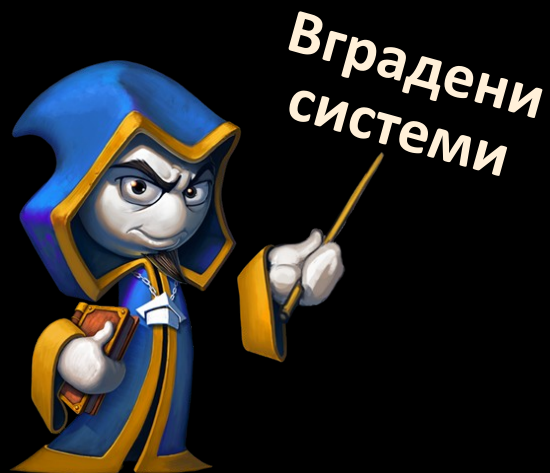


Arduino микроконтролер. Програмен език Arduino



Инж. Венцеслав Кочанов

Малко история

Arduino е компания за хардуер и софтуер с отворен код, проект и потребителска общност, която проектира и произвежда едноплаткови микроконтролери и комплекти микроконтролери за изграждане на цифрови устройства.

Дизайнът на платките Arduino използва различни микропроцесори и контролери. Платките са оборудвани с комплекти цифрови и аналогови входно/изходни (I/O) щифтове, които могат да бъдат свързани към различни разширителни платки („щитове“) или макетни платки (за създаване на прототипи) и други схеми.

Платките разполагат със серийни комуникационни интерфейси, включително универсална серийна шина (USB) на някои модели, които също се използват за зареждане на програми. Микроконтролерите могат да бъдат програмирани с помощта на езиците за програмиране C и C++ (Embedded C), като се използва стандартен API.

Проектът Arduino започна през 2005 г. като инструмент за студенти в Института за дизайн на взаимодействие Ивреа, Италия [3], целящ да предостави евтин и лесен начин за начинаещи и професионалисти да създават устройства, които взаимодействат с тяхната среда, използвайки сензори и задвижващи механизми.

Какво е Arduino

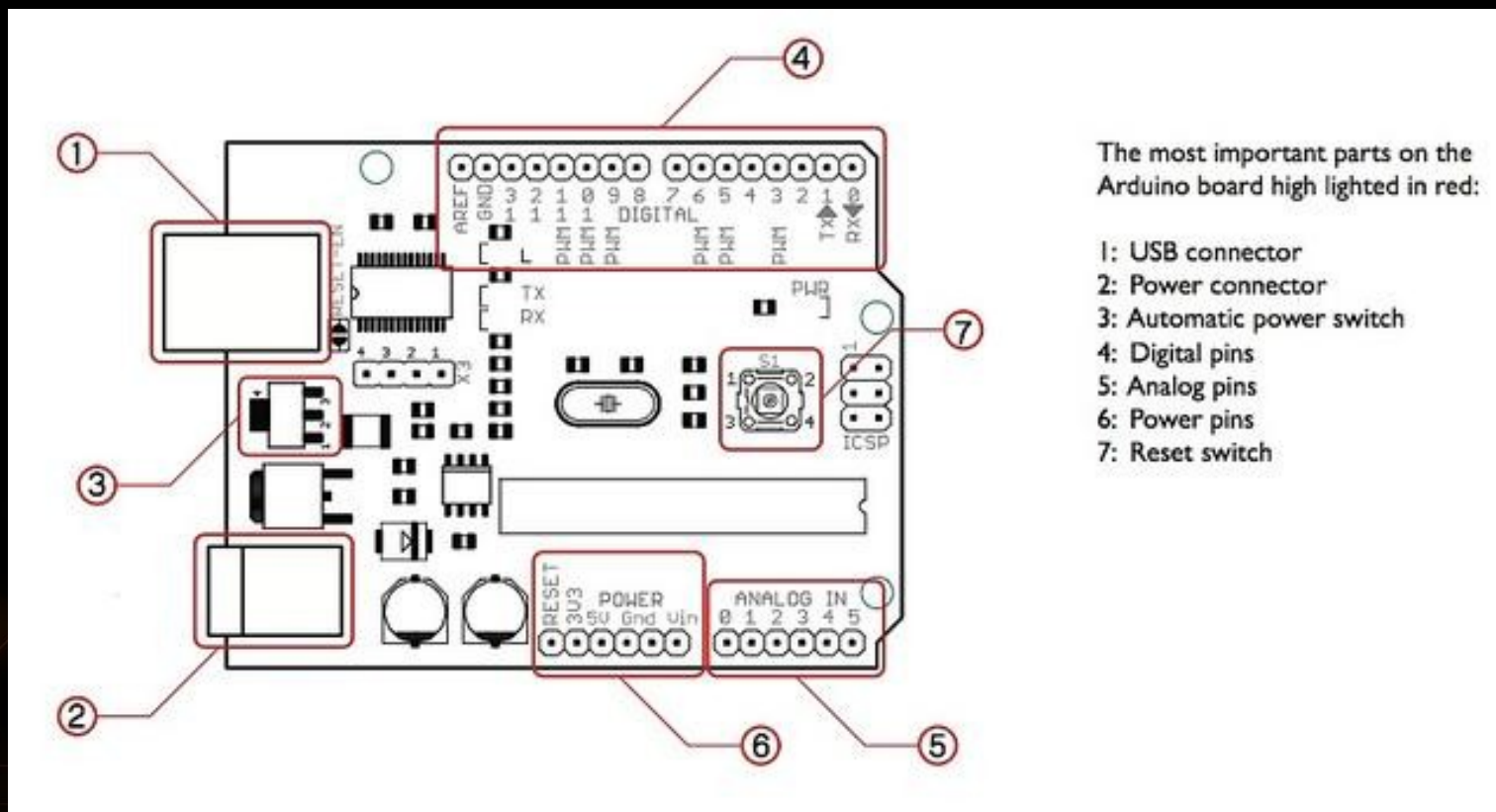
Arduino всъщност е базиран на микроконтролер комплект, който може да се използва директно чрез закупуване от доставчика или може да се направи у дома с помощта на компонентите, благодарение на хардуерната си функция с отворен код. Основно се използва в комуникациите и при управлението или управлението на много устройства. Основана е от Massimo Banzi и David Cuartielles през 2005 г.

Архитектура на Arduino:

Процесорът на Arduino основно използва архитектурата на Харвард, където програмният код и програмните данни имат отделна памет. Състои се от две памети - *програмна памет* и *памет за данни*. Кодът се съхранява във флаш програмната памет, докато данните се съхраняват в паметта за данни. Atmega328 има 32 KB флаш памет за съхраняване на код (от които 0,5 KB се използва за буутлоудър), 2 KB SRAM и 1 KB EEPROM и работи с тактова честота от 16MHz.

Пин диаграма на Arduino

Типичен пример за платка Arduino е Arduino Uno. Състои се от ATmega328 - 28 пинов микроконтролер



Arduino Uno се състои от 14 цифрови входно/изходни пина (от които 6 могат да се използват като PWM изходи), 6 аналогови входа, 16 MHz кристален осцилатор, USB връзка, жак за захранване, ICSP конектор и бутон за нулиране

Захранващ жак: Arduino може да се захранва от компютъра през USB или чрез външен източник като адаптер или батерия. Може да работи с външно захранване от 7 до 12V. Захранването може да се приложи външно чрез щифта Vin или чрез подаване на референтно напрежение през щифта IOREf.

Цифрови входове: Състои се от 14 цифрови входа/изхода, всеки от които осигурява или поема 40mA ток. Някои от тях имат специални функции като щифтове 0 и 1, които действат съответно като Rx и Tx, за серийна комуникация, щифтове 2 и 3-които са външни прекъсвания, щифтове 3,5,6,9,11, които осигуряват pwm изход и щифт 13, където е свързан LED.

Няколко от основните функции на Arduino са:

- `digitalRead(pin)`: Чете цифровата стойност на дадения `pin`.
- `digitalWrite(pin, value)`: Записва цифровата стойност в дадения `pin`.
- `pinMode(pin, режим)`: Настройва щифта на входен или изходен режим.
- `analogRead(pin)`: Чете и връща стойността.
- `analogWrite(pin, value)`: Записва стойността на този `pin`.
- `serial.begin(скорост на предаване)`: Задава началото на серийната комуникация чрез задаване на битовата скорост.

Стъпки за програмиране на Arduino

Програмите, написани на Arduino, са известни като скици /sketch/.

1. Основната скица се състои от 3 части:

- Декларация на променливи
- Инициализация: Пише се във функцията `setup ()`.
- Контролен код: Записан е във функцията `loop ()`.

2. Скицата се записва с разширение `.ino`. Всички операции като проверка, отваряне на скица, запазване на скица могат да се извършват с помощта на бутоните на лентата с инструменти или с помощта на менюто с инструменти.

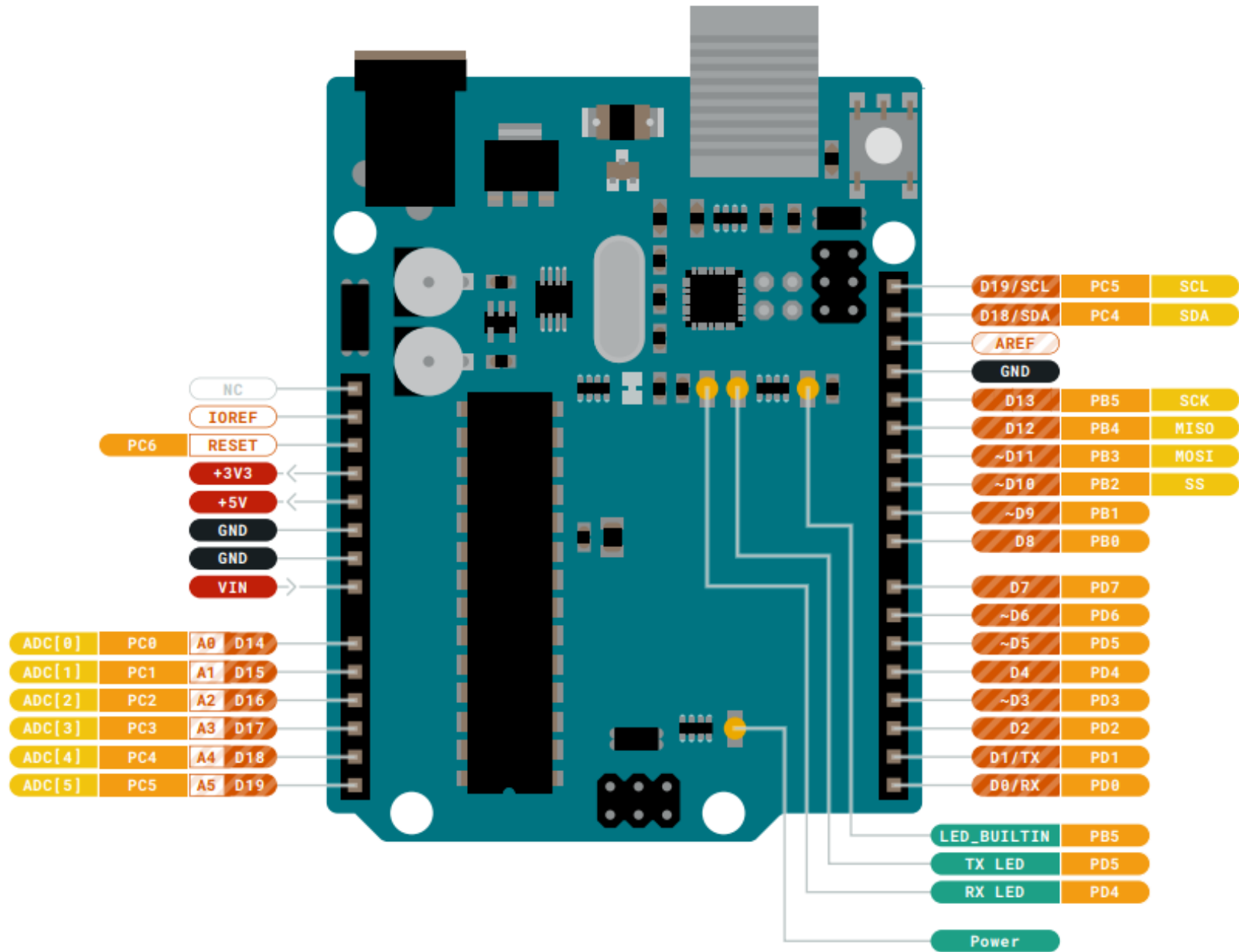
3. Скицата трябва да се съхранява в директорията на скицника.

4. Изберете подходящата платка от менюто с инструменти и номерата на серийния порт.

5. Щракнете върху бутона за качване или изберете качване от менюто с инструменти. По този начин кодът се качва от буутлоудъра върху микроконтролера.

Причини, поради които Arduino се предпочита в наши дни

- Евтин е
- Идва с хардуерна функция с отворен код, която позволява на потребителите да разработят свой собствен комплект, използвайки вече наличен като референтен източник.
- Софтуерът Arduino е съвместим с всички видове операционни системи като Windows, Linux и Macintosh и др.
- Той също така идва с функция за софтуер с отворен код, която позволява на опитни разработчици на софтуер да използват кода на Arduino, за да се слеят със съществуващите библиотеки на езика за програмиране и могат да бъдат разширявани и модифицирани.
- Лесен е за използване от начинаещи.
- Предлага се с лесна възможност за свързване с процесора на компютъра чрез серийна комуникация през USB, тъй като съдържа вградена схема за захранване и нупиране



Програмен език Arduino

Програмният език Arduino може да бъде разделен на три основни части: функции, стойности (променливи и константи) и структура:

Функции

Използват се за управление на платката Arduino и извършване на изчисления. По-важните са:

- Digital I/O
- Analog I/O
- Time
- Communication
- Math
- Random Numbers

Променливи

Типове данни и константи на Arduino:

- Константи /Constants/ - HIGH | LOW | INPUT | OUTPUT | INPUT_PULLUP
- Типове данни /Data Types/ - array, bool, boolean, byte, char, double, float, int

Структура

Елементите на кода на Arduino (C++), по-важните от които са:

Sketch:

- loop()
- setup()

Arithmetic Operators, Pointer Access Operators, Control Structure Bitwise Operators и други

Функцията setup()

Функцията `setup()` се извиква при стартиране на скица. Използвайте го, за да инициализирате променливи, режими на закрепване, да започнете да използвате библиотеки и т.н. Функцията `setup()` ще се изпълнява само веднъж, след всяко включване или нулиране на платката Arduino.

Функцията loop()

Функцията `loop()` прави точно това, което подсказва името ѝ, и зацикля последователно, позволявайки на вашата програма да се променя и да реагира.

Аналогови I/O устройства

Аналоговите I/O устройства са подходящи за приложения, които изискват прецизно измерване или контрол на променливи, които могат да се променят непрекъснато, като температура, налягане, поток, скорост или ниво. Тези устройства могат да улавят или произвеждат широк диапазон от стойности, които съответстват на физическите явления, които представляват. Например, аналогов температурен сензор може да изведе напрежение, което е пропорционално на температурата, която измерва, а аналогов вентил може да регулира потока на течност чрез регулиране на отварянето му. Докато аналоговите I/O устройства предлагат по-точна и подробна информация от цифровите устройства, както и повече гъвкавост и съвместимост с различни типове сензори и изпълнителни механизми, те са по-скъпи и сложни от цифровите устройства. Освен това аналоговите I/O устройства са по-податливи на шум и смущения, които могат да повлияят на качеството и точността на сигнала и изискват повече окабеляване и калибриране

Цифрови I/O устройства

Цифровите I/O устройства са подходящи за приложения, които изискват просто и дискретно управление или наблюдение на променливи с две състояния, като включено/изключено, отворено/затворено или присъствие/отсъствие. Тези устройства могат да изпращат или получават двоични сигнали, които показват състоянието на устройството или процеса. Цифровите I/O устройства имат няколко предимства, като например, че са по-евтини и по-прости от аналоговите устройства, по-надеждни и здрави, по-малко засегнати от шум и смущения, изискват по-малко окабеляване и калибриране и лесно се свързват с PLC чрез стандартни цифрови модули. От друга страна, цифровите I/O устройства могат да осигурят по-малко информация и разделителна способност от аналоговите устройства, да обработват само прости и линейни процеси без фина настройка или обратна връзка и имат ограничена гъвкавост и съвместимост с различни типове сензори и изпълнителни механизми.

Какво представляват аналоговите сигнали?

Както споменахме, аналоговият сигнал е непрекъснат сигнал, при който варираща във времето величина, като напрежението, представлява друга променлива. Следователно една променлива е „аналог“ на другата. Това технически означава, че една аналогова система ще позволи представянето на безкраен брой стойности, дори когато това не изглежда така.

Какво представляват цифровите сигнали?

Цифровите сигнали са точно обратното, тъй като те използват това, което е известно като двоичен код, който е поредица от единици и нули, за ефективно предаване на информация. Те се изпращат като електронни импулси и могат да бъдат разбрани от други устройства, които използват цифрови сигнали. Цифровите сигнали имат много приложения, включително телекомуникации, цифрово аудио и видео. Те често се разглеждат като по-надеждни от аналоговите сигнали, тъй като в повечето случаи могат да

Вградени системи



Въпроси?

