



Основи на програмирането с Микроконтролер Arduino NANO

Разработил: инж.А.Анчев



Общи сведения за микроконтролерите

- ❗ Микроконтролерът е едночипова компютърна система, съчетаваща в себе си микропроцесор, тактов генератор, оперативна памет и програмируеми входно-изходни устройства.
- ❗ Често на същия чип има и различни видове компютърна памет. За разлика от микропроцесорите, които се използват в персоналните и други компютри,





Abir



микроконтролерите са незаменими във вградените системи и са особено полезни, когато трябва да се реализира компютърно устройство, изпълняващо голям брой или сравнително сложни функции, например – комуникация с други устройства, управление на буквено-цифрови или графични дисплеи, измерване на различни величини, управление на технологични процеси и др.

SOCIETY
ROBOTIC-





AVR микроконтролери

- ✦ AVR е семейство микроконтролери, разработени от 1996 г. от **Atmel**, придобити от **Microchip Technology** през 2016 г. Това са модифицирани 8-битови RISC едноцименни микроконтролери от Харвардската архитектура;
- ✦ AVR е една от първите групи микроконтролери, които използват флаш памет като място за програмно съхранение;





Abirah

SOCIETY
ROBOTIC—



- 💡 **Flash** , **EEPROM** и **SRAM** са интегрирани в един чип, премахвайки необходимостта от външна памет в повечето приложения.
- 💡 Някои устройства имат паралелна външна шина, за да позволят добавянето на допълнителна памет или устройства с карти с памет;
- 💡 Инструкциите на програмата се съхраняват в енергонезависима флаш памет;



Dr. H. H. H.

✦ Всеки GPIO порт се управлява от три 8-битови регистри - **DDR x** , **PORT x** и **PIN x** , където **x** е идентификатор на порт;

DDR x - Регистър на посоката на данните, конфигурира пина като вход или изход;

PORT x - Регистър на изходния порт. Задава изходната стойност на изводите, конфигурирани като изходи. Активира или деактивира изтеглящия резистор на пина, конфигуриран като вход;





PIN x - Входен регистър, използван за четене на входен сигнал. На някои устройства този регистър може да се използва за превключване на пина - записването на логика към бит за **PIN x** превключва съответния бит в PORT x , независимо от настройката на DDR x бита.



- ✱ AVR устройствата обикновено поддържат тактова честота от 0 до 20 MHz, като някои устройства достигат до 32 MHz. Работата с ниска мощност обикновено изисква намалена тактова честота;
- ✱ Всички съвременни AVR разполагат с осцилатор, който премахва нуждата от външен тактов генератор или резонаторни схеми;





- ✿ Някои AVR устройства също имат системен часовник, който може да дели системния часовник с до 1024;
- ✿ Тъй като всички операции (с изключение на умножаването и 16-битовото добавяне/изваждане) на регистрите R0-R31 са едноциклични, AVR може да постигне до 1 MIPS (Millions of instructions per second) за 1 MHz, т.е. 8 MHz процесор може да постигне до 8 MIPS. Зарежданията и съхраняванията към и от паметта отнемат два цикъла;





Dr. H. H. H.

- ✱ AVR устройствата предлагат широк набор от функции:
 - Многофункционални, двупосочни I / O портове за общо предназначение с конфигурируеми вградени pull-up резистори;
 - Няколко вътрешни осцилатора, включително осцилатор с RC без външни части;
 - Вградена флаш памет с инструкции за самопрограмиране до 256 KB (384 KB на X-Mega);





Dr. H. H. H.

SOCIETY
ROBOTICS



- Програмируеми в системата чрез използване на серийни / паралелни нисковолтови собствени интерфейси или JTAG;
- Секция за избор на код за зареждане с независими битове за защита;
- Дебъгер за отстраняване на грешки (OCD) чрез JTAG или debugWIRE на повечето устройства;



- Сигналите JTAG (TMS, TDI, TDO и TCK) са мултиплексирани на GPIO . Тези пинове могат да бъдат конфигурирани да функционират като JTAG или GPIO в зависимост от настройката на резистора, който може да бъде програмиран чрез ISP или HVSP. По подразбиране AVRс с JTAG идват с активиран интерфейс JTAG;



- Разполагат с EEPROM до 4 KB;
- Разполагат с SRAM до 16 KB (32 KB на XМega);
- Режимите за адресиране на данни са разширени, за да могат да се адресират директно до 16 MB памет;
- Разполагат с 8-битови и 16-битови таймери;
- PWM изход (някои устройства имат подобрена PWM периферия, която включва така наречения «dead-time» генератор);



ARDUINO

- ✱ **Arduino** е проект с отворен код и едноименна компания, произлязла от него, както и свързаното с него потребителско общество.
- ✱ **Arduino** е сбор от три компоненти - **хардуер**, **софтуер** и **общност**.
- ✱ За да създавате проекти с Arduino, трябва да имате основни понятия от три направления:
 - **механика**;
 - **електроника**;
 - **програмиране**;

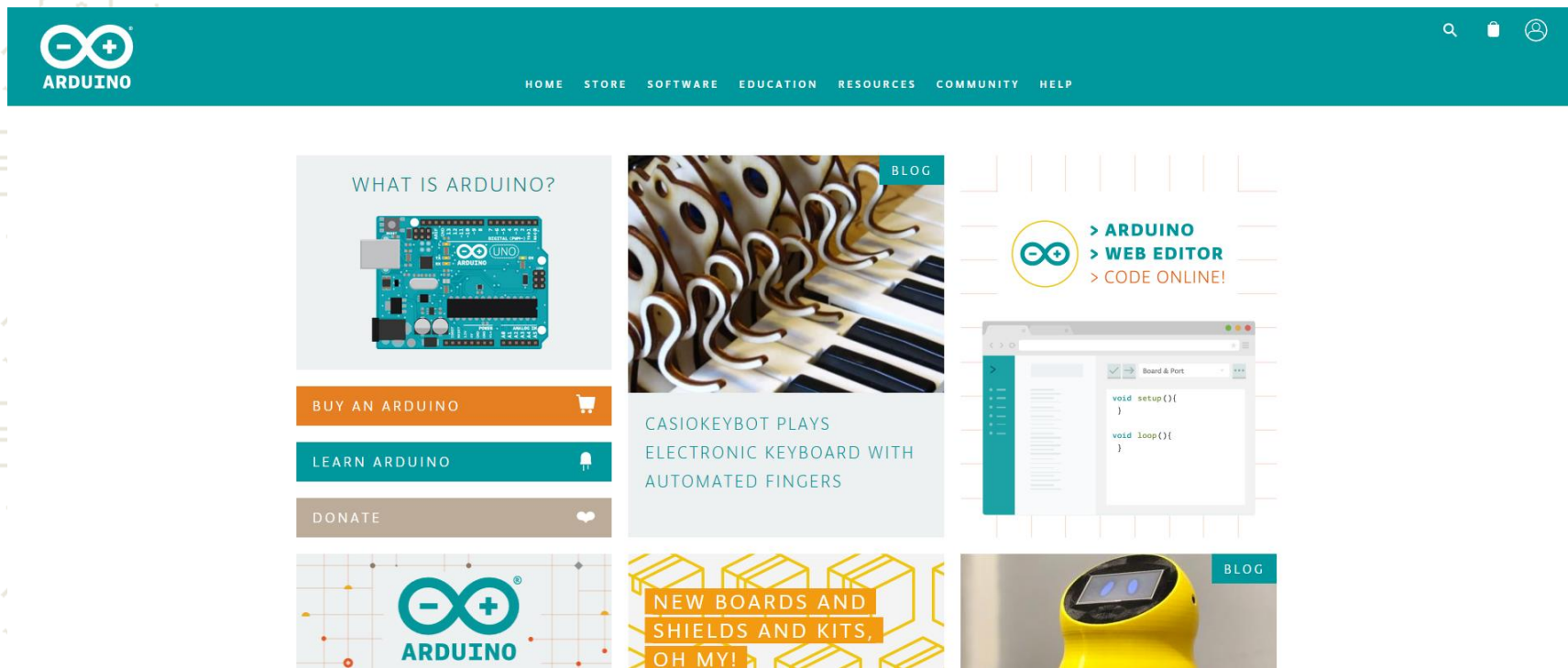




SOCIETY OF ROBOTICS



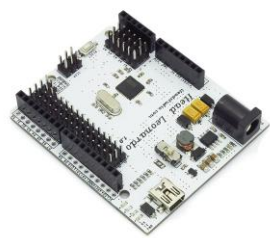
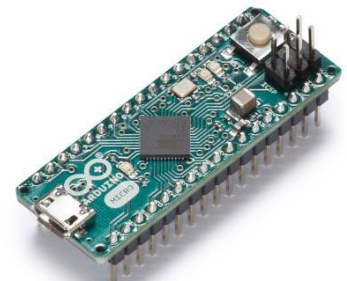
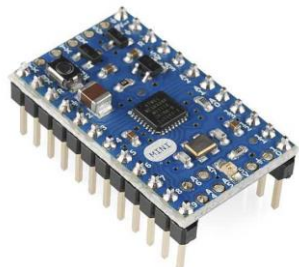
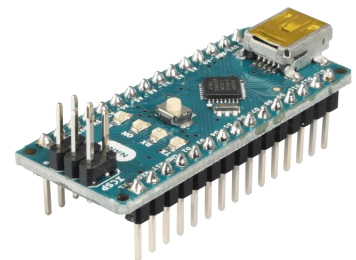
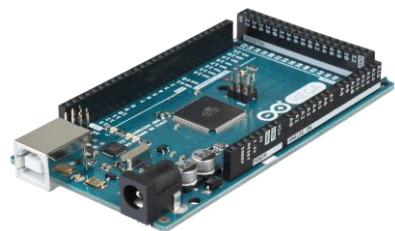
🔦 Официален сайт - www.arduino.cc





Abir

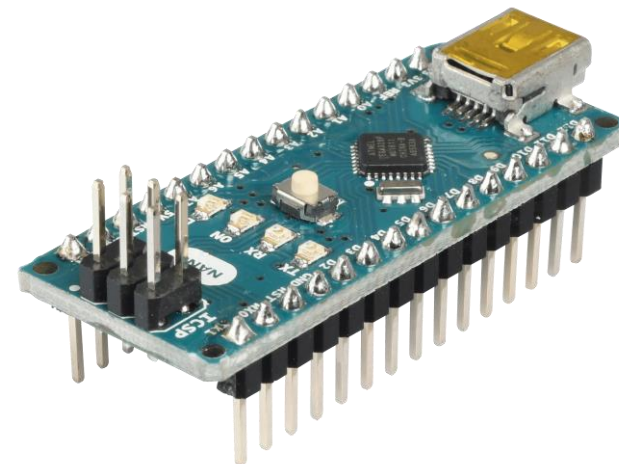
**SOCIETY
ROBOTIC**





Arduino NANO

- Изградена е с микроконтролер **ATmega328P**. Наличен е USB-сериен порт конвертор **CP2102**.
- Работи с тактова честота 16MHz, поддържа UART, I2C и SPI интерфейси и има 20 I/O порта, от които до 8 аналогови входа и до 6 PWM изхода.



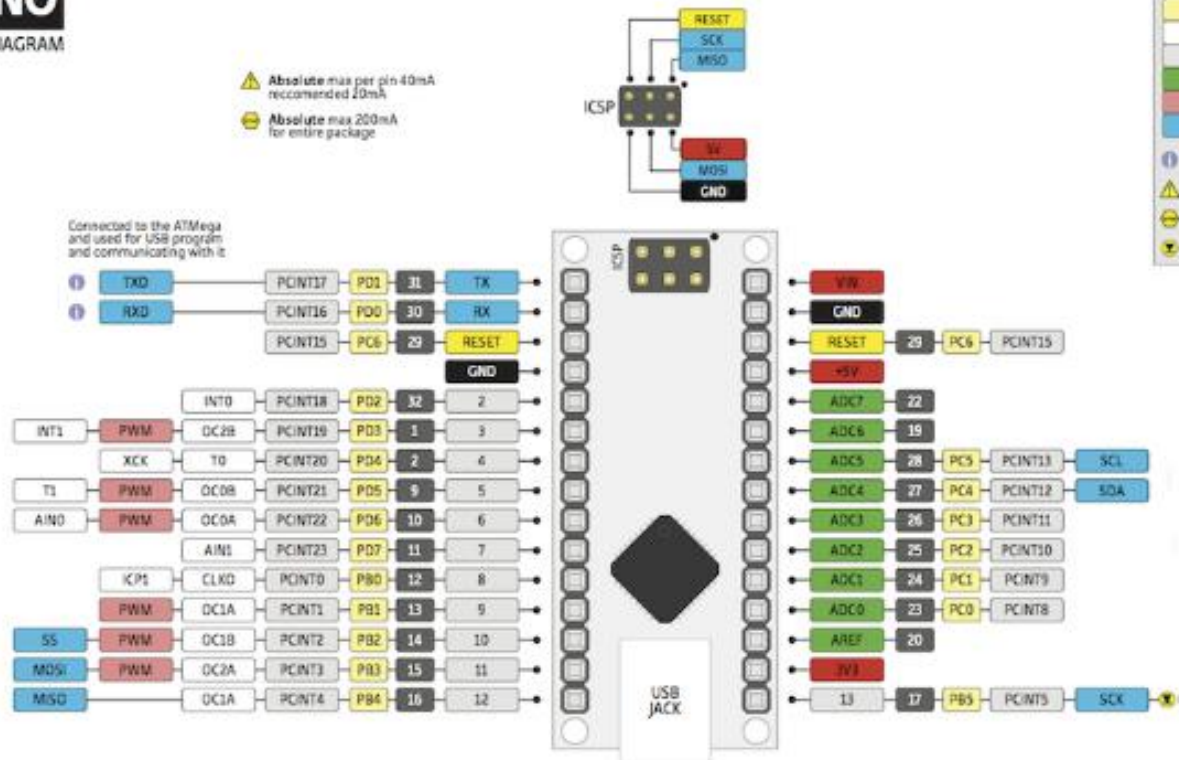


PinMap

THE UNOFFICIAL ARDUINO NANO PINOUT DIAGRAM

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

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



| LEGEND | |
|--------|-------------------------|
| | GND |
| | POWER |
| | CONTROL |
| | PHYSICAL PIN |
| | PORT PIN |
| | ATMEGA328 PIN FUNC |
| | DIGITAL PIN |
| | ANALOG-RELATED PIN |
| | PWM PIN |
| | SERIAL PIN |
| | General information |
| | Pay Attention |
| | No Really PAY ATTENTION |
| | LED |



Спецификация

| Параметър | Стойност |
|----------------------------|---|
| Микроконтролер | ATmega328P |
| Архитектура | AVR |
| Работно напрежение | 5V |
| Флаш памет | 32 KB от които 2 KB, използвани от Bootloader |
| SRAM | 2 KB |
| Тактова честота | 16 MHz |
| Аналогови I / O пина | 8 бр. |
| Цифрови I / O пина | 14 бр. |
| EEPROM | 1 KB |
| DC Current за I / O пинове | 40 mA |
| Захранващо напрежение | 7 – 12V |
| Пинове с PWM изход | 6 бр. |
| Консумация на енергия | 19 mA |



| Features | ATmega328/P | | | | | | |
|---------------------------|--|------|---------------------|------------|------------|---------|------------------|
| Microcontroller | 8-bit family | | | | | | |
| Architecture | RISC | | | | | | |
| Pin Count | 28 or 32 | | | | | | |
| Flash (Bytes) | 32K | | | | | | |
| SRAM (Bytes) | 2K | | | | | | |
| EEPROM (Bytes) | 1K | | | | | | |
| General Purpose I/O Lines | 23 | | | | | | |
| SPI | 2 | | | | | | |
| TWI (I2C) | 1 | | | | | | |
| USART | 1 | | | | | | |
| ADC | Internal Single 10-bitADC | | | | | | |
| ADC Channels | 8 | | | | | | |
| 8-bit Timer/ Counters | 2 | | | | | | |
| 16-bit Timer/ Counters | 1 | | | | | | |
| Operating Voltage | 1.8V-5.5V | | | | | | |
| PWM Channels | 6 | | | | | | |
| Six Sleep Modes | <table><tr><td>Idle</td><td>ADC Noise Reduction</td></tr><tr><td>Power-save</td><td>Power-down</td></tr><tr><td>Standby</td><td>Extended standby</td></tr></table> | Idle | ADC Noise Reduction | Power-save | Power-down | Standby | Extended standby |
| Idle | ADC Noise Reduction | | | | | | |
| Power-save | Power-down | | | | | | |
| Standby | Extended standby | | | | | | |



Чипът **FTDI FT232RL** на Arduino Nano се захранва само ако платката се захранва през USB порта. В резултат на това, когато работите на външно (без USB) захранване, изхода от 3.3V (който се доставя от чипа FTDI) не е наличен, а RX и TX светодиодите ще мигат, само ако цифровите пинове 0 или 1 са в състояние **"HIGH"**;





Abir

Arduino Digital and Analog I/O Pins

- ▶ Digital pins:
 - ▶ Pins 0 – 7: PORT D [0:7]
 - ▶ Pins 8 – 13: PORT B [0:5]
 - ▶ Pins 14 – 19: PORT C [0:5] (Arduino analog pins 0 – 5)
 - ▶ digital pins 0 and 1 are RX and TX for serial communication
 - ▶ digital pin 13 connected to the base board LED
- ▶ Digital Pin I/O Functions
 - ▶ `pinMode(pin, mode)`
 - ▶ Sets pin to INPUT or OUTPUT mode
 - ▶ Writes 1 bit in the DDRx register
 - ▶ `digitalWrite(pin, value)`
 - ▶ Sets pin value to LOW or HIGH (0 or 1)
 - ▶ Writes 1 bit in the PORTx register
 - ▶ `int value = digitalRead(pin)`
 - ▶ Reads back pin value (0 or 1)
 - ▶ Read 1 bit in the PINx register





Abir

Arduino Analog I/O

- ▶ Analog input pins: 0 – 5
- ▶ Analog output pins: 3, 5, 6, 9, 10, 11 (digital pins)
- ▶ Analog input functions
 - ▶ `int val = analogRead(pin)`
 - ▶ Converts 0 – 5v. voltage to a 10-bit number (0 – 1023)
 - ▶ Don't use `pinMode`
 - ▶ `analogReference(type)`
 - ▶ Used to change how voltage is converted (advanced)
- ▶ Analog output
 - ▶ `analogWrite(pin, value)`
 - ▶ value is 0 – 255
 - ▶ Generates a PWM output on digital pin (3, 5, 6, 9, 10, 11)
 - ▶ @490Hz frequency





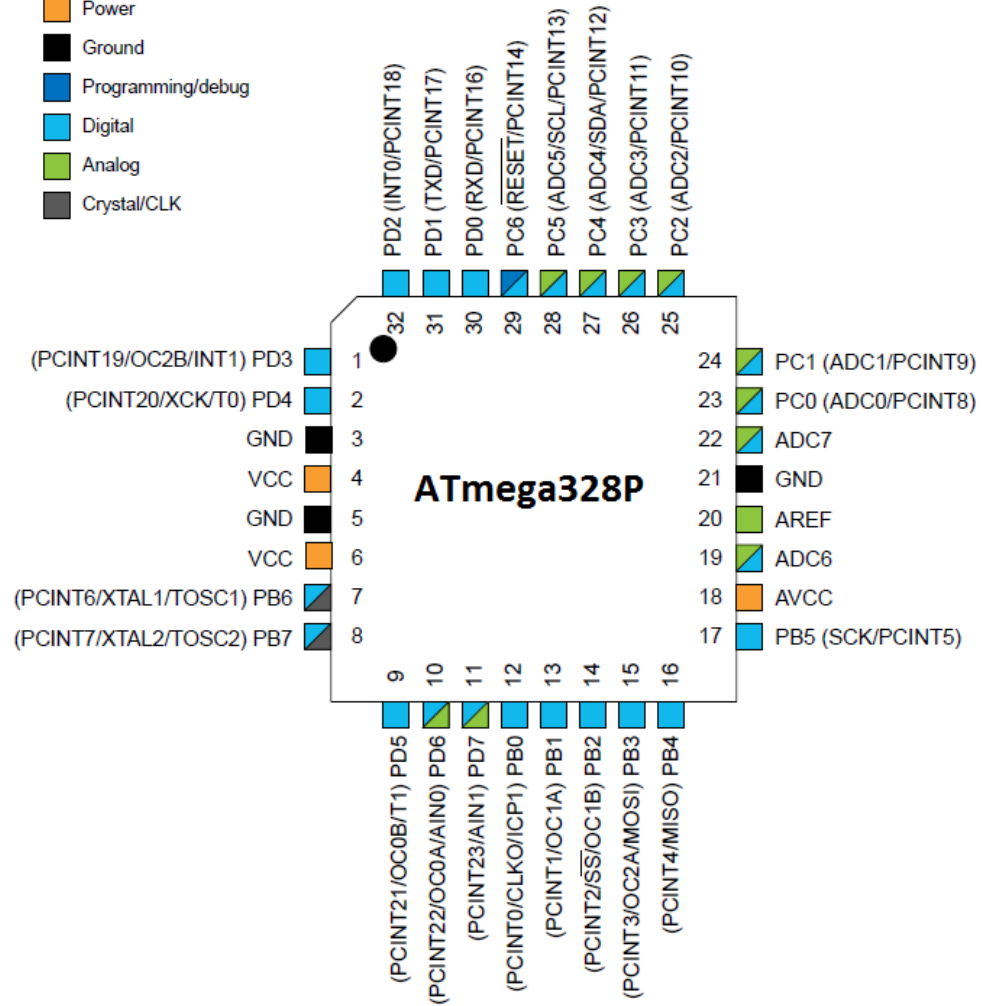
Abirah

SOCIETY
ROBOTIC



32-pin TQFP Top View

- Power
- Ground
- Programming/debug
- Digital
- Analog
- Crystal/CLK





Dr. Arshad

SOCIETY
ROBOTICS



ARDUINO IDE

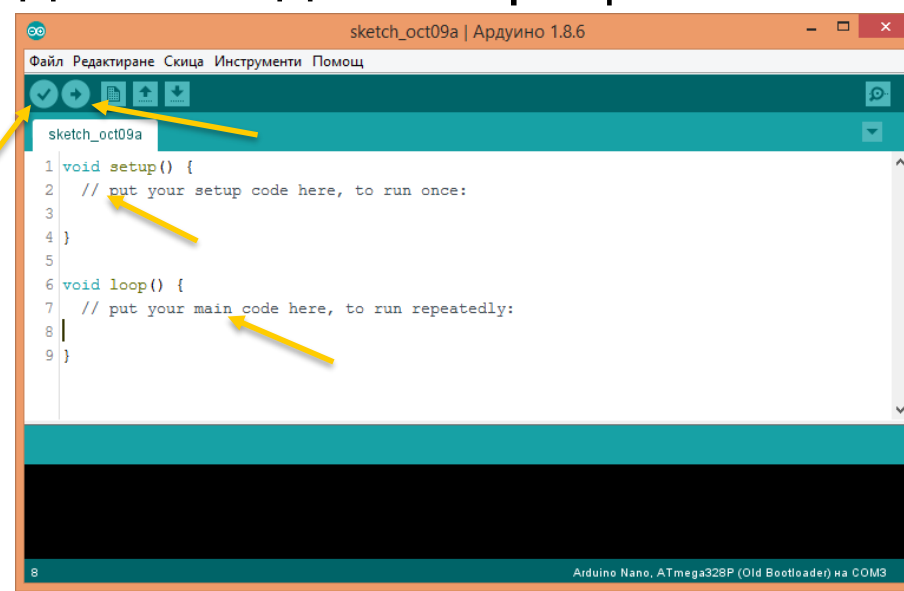
- ✱ Интегрираната среда за разработка на Arduino IDE е мултиплатформено приложение, написано на програмния език Java и произлиза от езика за програмиране „Processing“ и проекта „Wiring“.
- ✱ Конструирана е така, че да улесни програмирането от хора, които не са запознати с писането на софтуер.
- ✱ Програмира се на **Arduino Programming Language (APL)** – език, подобен на C/C++.



- ✦ Съдържа редактор на код с функции като:
 - **syntax highlighting** (показване на текст в различен цвят в зависимост от принадлежността на термините)
 - **brace matching** (функция, която следи отварящите и затварящите скоби с цел по-лесна навигация през програмния код),
 - **автоматично подравняване**
- ✦ Също така е способен да компилира и да качва програми към устройството с един клик.



- ⚡ Програма, написана за Arduino, се нарича „скеч“ и е написана на програмните езици C и C++.
- ⚡ Средата за разработка Arduino IDE идва със софтуерна библиотека „Wiring“, което улеснява много входни/изходни операции.





Archie

SOCIETY
ROBOTICS



- ✦ За да направят циклично изпълняваща се програма, потребителите трябва да дефинират само две функции,:
- ✦ **Setup()** - Функция, която се изпълнява веднъж в началото на програмата и може да инициализира настройки;
- ✦ **Loop()** - Функция, която не спира да се извиква докато Ардуино не се изключи.



Dr. H. H. H.

Последователност

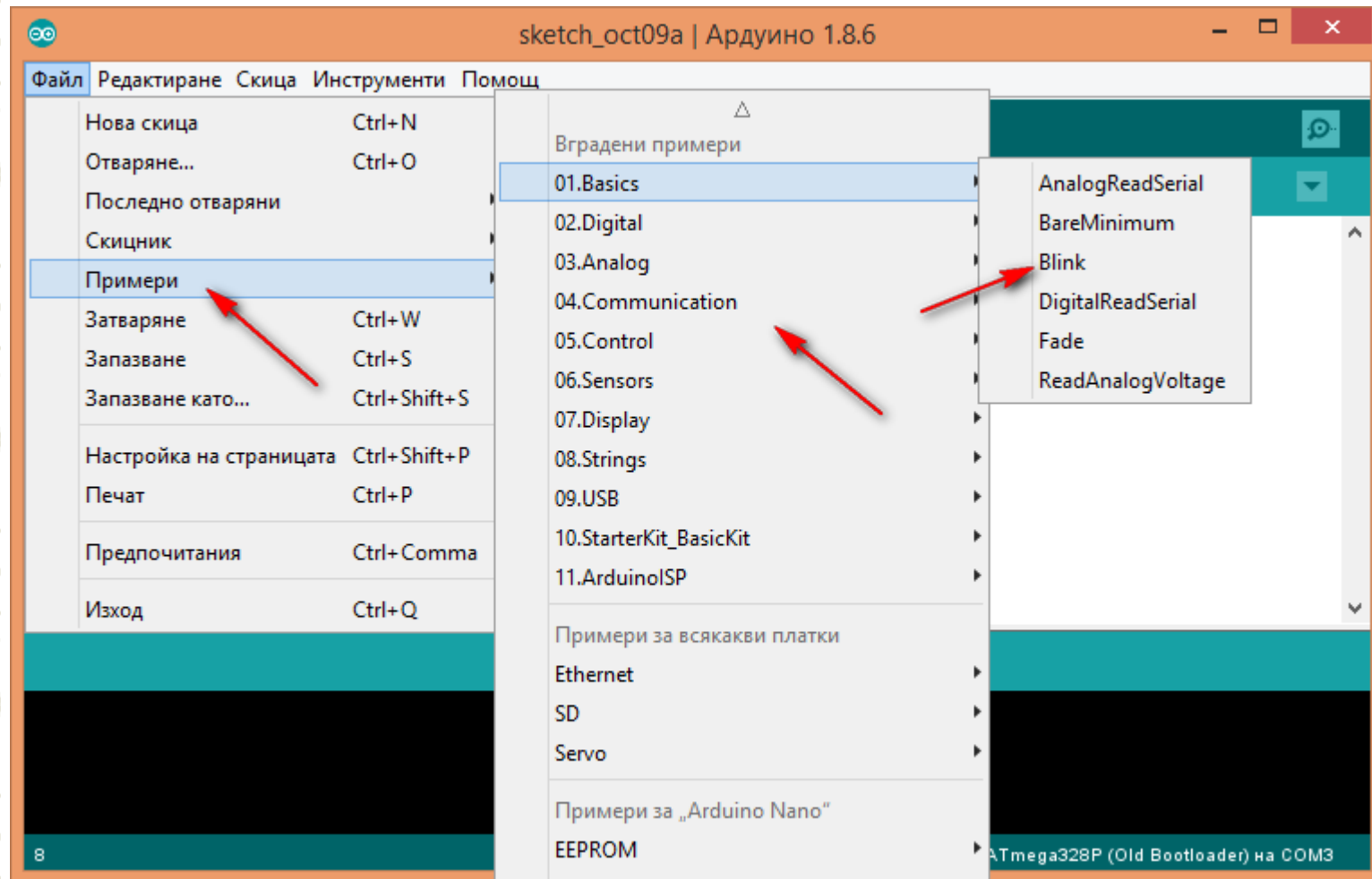
- ✦ Стартиране на IDE
- ✦ Писане на кода и логиката
- ✦ Проверка с verify/compile бутона за грешки и коректност
- ✦ Свързване на платката с USB към компютъра
- ✦ Инсталиране на драйверите (само еднократно на даден PC)
- ✦ Избор и настройка на серийния порт (еднократно, при нужда)
- ✦ Избор на платката (еднократно, при нужда)
- ✦ Програмиране на контролера чрез Upload бутона и тестване

SOCIETY
ROBOTICS





Примерни проекти





Мигане на светодиодиод

SOCIETY
ROBOTIC



```
Blink | Ардуино 1.8.6
Файл Редактиране Скица Инструменти Помощ

Blink$

1 // the setup function runs once when you press reset or power the board
2 void setup() {
3   // initialize digital pin LED_BUILTIN as an output.
4   pinMode(LED_BUILTIN, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
10  delay(1000); // wait for a second
11  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
12  delay(1000); // wait for a second
13 }
14
```

14 Arduino Nano, ATmega328P (Old Bootloader) на COM3



Dr. H. H. H.

SOCIETY
ROBOTICS



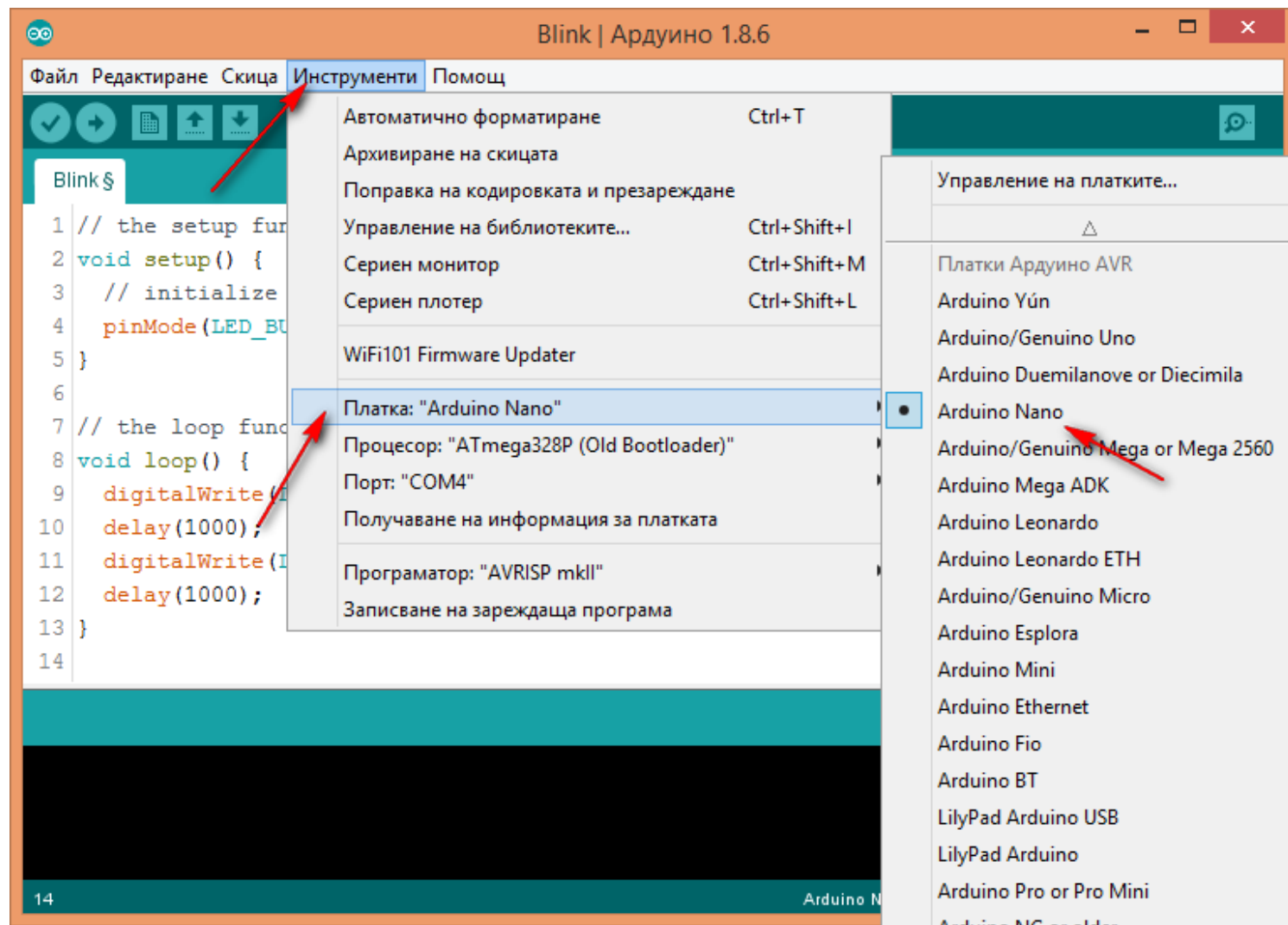
- ✦ Стандартният Arduino код не може да бъде видян от стандартен C++ компилатор като валидна програма, затова, когато потребителят кликне на бутона „Upload“ в средата за разработка, копие на кода се записва във временен файл, в който се добавя функцията **main()**, посочена по-горе, за да се валидира кода (нанасят се и други промени, разбира се, това се нарича preprocessing).
- ✦ Средата за разработка на Arduino използва **GNU toolchain** и **AVR Libc**, за да компилира програмите и използва **Avrdude**, за да качва програмите на устройството.



- ✦ За да може средата Arduino да зареди програмата в микроконтролера, той трябва да има зареден т.нар. **bootloader** (програма за зареждане на код).
- ✦ Всички платки Arduino са с предварително качен такъв, поради тази причина „липсват“ 0.5 до 2.5kB от флаш паметта на контролера. Тази част от паметта е заета от кода на зареждащия модул.

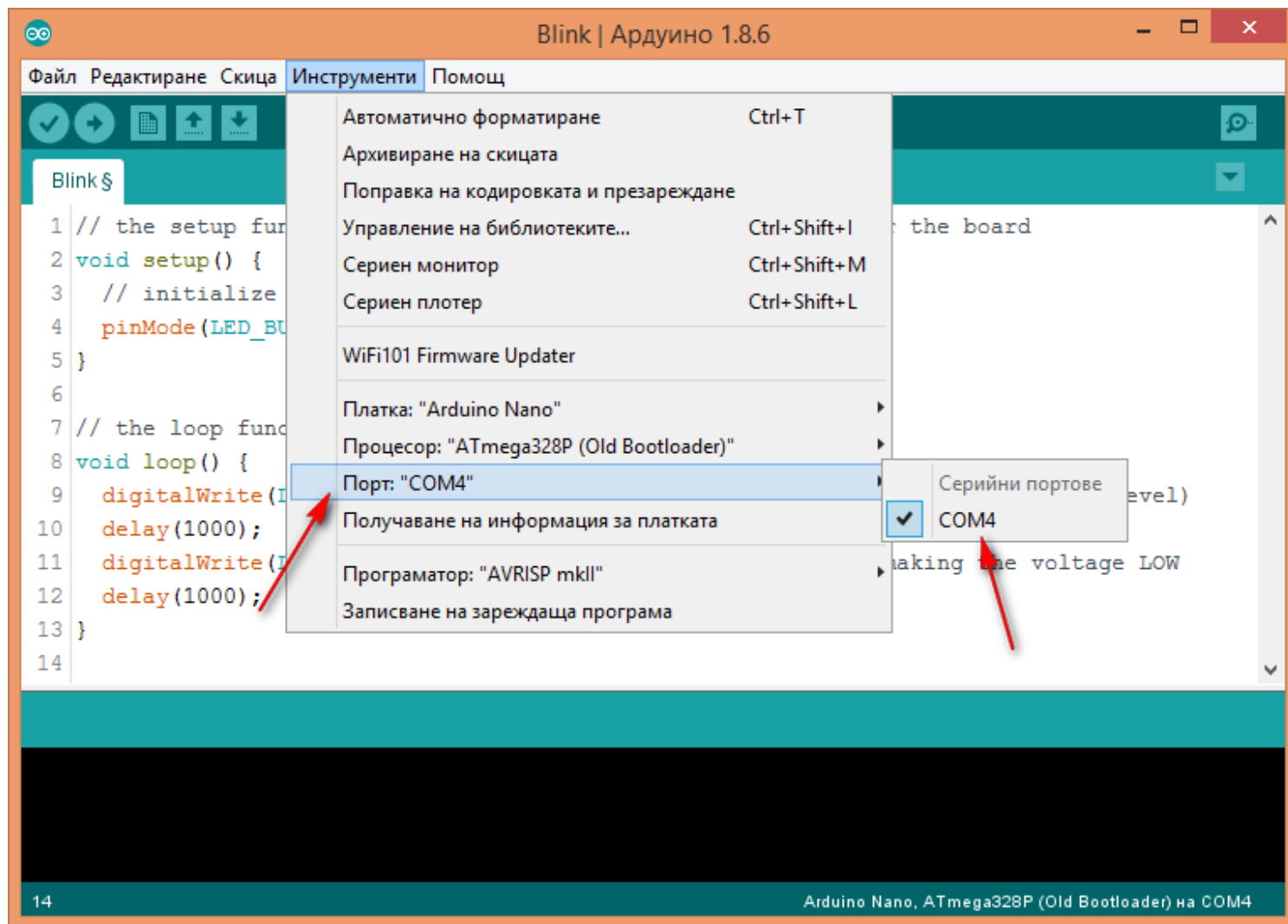


Избор на микроконтролер





Избор на COM порт





CodeBlocs Arduino IDE

- ✦ CodeBlocs Arduino IDE е програмна среда за разработване на проекти предназначени за Arduino;
- ✦ Средата включва стандартните Arduino библиотеки, AVR toolchain, Arduino Builder, сериен терминал и най-интересния симулатор Arduino на ниво API



Abirah

SOCIETY
ROBOTIC



test.ino [test] - Code:Blocks 13.12

File Edit View Search Project Build Debug Tools Plugins Settings Help

Management

Projects Symbols Files

Workspace

test

Sources

```
1 #include <Arduino.h>
2
3 /*
4  * Turns on an LED on for one second, then off for one second, repeatedly.
5  */
6
7 void setup()
8 {
9     Serial.begin(9600);
10
11     // initialize the digital pin as an output
12     // Pin 13 has an LED connected on most Arduino boards
13     pinMode(13, OUTPUT);
14 }
15
16 void loop()
17 {
18     Serial.println("Hello world!");
19
20     delay(100); // wait for a second
21     digitalWrite(13, HIGH); // set the LED on
22     delay(100); // wait for a second
23     digitalWrite(13, LOW); // set the LED off
24 }
25
```

Arduino Builder (Ver. 0.8.9)

Powered by ArduinoDev.com

Board Type: Arduino Nano (328)

COM4 USBASP ArduinoISP

Uploading completed! (elapsed: 3s)

Log Report

Starting upload for Arduino Nano (328) via COM4...

avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude.exe: Device signature = 0x1e950f

avrdude.exe: reading input file "C:\Arduino\test\build\test_nano328.hex"

avrdude.exe: writing flash (2536 bytes):

Writing | ##### | 100% 0.72s

avrdude.exe: 2536 bytes of flash written

Logs & others

Code::Blocks Search results Build log Build messages CppCheck CppCheck messages Debugger Closed files list Cscope

Run: Arduino Nano (328) in test (compiler: GNU AVR GCC Compiler)

Checking for existence: C:\Arduino\test\build\test_nano328.elf

Executing: "C:\Users\Aurora\Downloads\CodeBlocks\cb_console_runner.exe" "C:\Arduino\test\build\test_nano328.elf" (in C:\Arduino\test\)

Process terminated with status 0 (0 minute(s), 0 second(s))

C:\Arduino\test\test.ino

Windows (CR+LF) WINDOWS-1251 Line 22, Column 1



Цифрови пинове

- ✱ Цифровите пинове могат да бъдат конфигурирани като входни и изходни. Това става във функцията `setup()` с командата `pinMode`, съответно `INPUT/OUTPUT`;
- ✱ На цифровите пинове на Arduino NANO могат да бъдат задавани следните състояния:
 - **HIGH** – цифровият пин е с напрежение 5V;
 - **LOW** – цифровият пин е с напрежение 0V;



Dr. Arshad

SOCIETY
ROBOTICS



- ✶ Прочитането на състоянието на даден пин става с командта **digitalRead(pin)** и връща като резултат **HIGH** или **LOW**;

```
int ledPin = 13;    // LED connected to digital pin 13
int inPin = 7;      // pushbutton connected to digital pin 7
int val = 0;        // variable to store the read value

void setup()
{
    pinMode(ledPin, OUTPUT);    // sets the digital pin 13 as output
    pinMode(inPin, INPUT);      // sets the digital pin 7 as input
}

void loop()
{
    val = digitalRead(inPin);    // read the input pin
    digitalWrite(ledPin, val);   // sets the LED to the button's value
}
```



Dr. Arshad

✦ Промяната на състоянието на даден цифров пин става с командата **digitalWrite(pin, value);**

```
void setup()
{
    pinMode(13, OUTPUT);           // sets the digital pin 13 as output
}

void loop()
{
    digitalWrite(13, HIGH);        // sets the digital pin 13 on
    delay(1000);                  // waits for a second
    digitalWrite(13, LOW);         // sets the digital pin 13 off
    delay(1000);                  // waits for a second
}
```





Аналогов вход

- ✱ Прочитане на аналогови стойности от хардуерно устройство се извършва с командата `analogRead(pin);`
- ✱ Резолуцията на ADC е 10-bit;
- ✱ Резултатът от четенето е цяло число в диапазона 0 – 1023 и трябва да се запише в променлива;

```
int analogPin = 3;    // potentiometer wiper (middle terminal) connected to analog pin 3
                      // outside leads to ground and +5V
int val = 0;          // variable to store the value read
void setup()
{
  Serial.begin(9600);  // setup serial
}
void loop()
{
  val = analogRead(analogPin); // read the input pin
  Serial.println(val);         // debug value
}
```





Аналогов изход

- ✦ Използва се за генериране на постоянна квадратна вълна с честота 490 Hz или 980 Hz на някои микроконтролери. Командата за задаване на аналогов изход е `analogWrite(pin, value);`
- ✦ Стойностите на `value` вариат в диапазона 0 – 255;

```
int ledPin = 9;      // LED connected to digital pin 9
int analogPin = 3;   // potentiometer connected to analog pin 3
int val = 0;         // variable to store the read value

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the pin as output
}

void loop()
{
  val = analogRead(analogPin); // read the input pin
  analogWrite(ledPin, val / 4); // analogRead values go from 0 to 1023, analogWrite values from 0 to 255
}
```



Извеждане на данни

- ✱ Извеждането на данни по същество представлява създаване на серийна комуникация между микроконтролера и компютъра по USB канал;
- ✱ За да създадем такава комуникация е необходимо във функцията `setup()` да зададем командата `Serial.begin(baud rate)`;
- ✱ За да изведем данни на серийния монитор е необходимо да използваме командата `Serial.print()` или `Serial.println()`;





Archie

SOCIETY
ROBOTIC



AnalogReadSerial | Ардуино 1.8.6

Файл Редактиране Скица Инструменти Помощ

AnalogReadSerial

```
11 */
12
13 // the setup routine runs once when you press reset:
14 void setup() {
15   // initialize serial communication at 9600 bits per second:
16   Serial.begin(9600);
17 }
18
19 // the loop routine runs over and over again forever:
20 void loop() {
21   // read the input on analog pin 0:
22   int sensorValue = analogRead(A0);
23   // print out the value you read:
24   Serial.println(sensorValue);
25   delay(1); // delay in between reads for stability
26 }
```

Arduino Nano, ATmega328P (Old Bootloader) на COM4



Dr. H. H. H.

- ✶ Функцията **delay(ms)** задава време за пауза на изпълнението на програмата на микроконтролера;
- ✶ Времето се задава в милисекунди;

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()
{
    pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
    digitalWrite(ledPin, HIGH); // sets the LED on
    delay(1000);                // waits for a second
    digitalWrite(ledPin, LOW);  // sets the LED off
    delay(1000);                // waits for a second
}
```



Добавяне на библиотека

- ✦ За да добавите библиотека в кода, от менюто Sketch изберете **Include Library**.
- ✦ Ако виждате нужната Ви библиотека в списъка, просто я изберете – тя ще се добави към проекта. От менюто **Add .ZIP Library** можете да добавите библиотека, която е предоставена в zip файл, така Arduino IDE ще я сложи на правилното място и включи във Вашия проект.



✦ От точката **Manage Libraries** се извиква **Library Manager на Arduino**, с чиято помощ може да търсите библиотеки по ключови думи (в полето „filter your search“) и да ги свалите и инсталирате веднага автоматично (изберете **Install now**, след като намерите нужната библиотека).

