



Вградени системи и  
ОС.

# Вградена система

- ▶ „Компютърна система, предазначена за управлението на процес или обект (в реално време)“
- ▶ Тази компютърна система е „вградена“ в обекта, който управлява

# ES vs PC

- ▶ Малки размери
- ▶ Ниска консумация на енергия
- ▶ Ниска производствена цена
- ▶ Проектирана единствено за управлението на процеса или обекта
- ▶ Специализирани архитектури

# Приложения

- ▶ Телекомуникации
- ▶ Потребителска електроника
- ▶ Домакинска електроника
- ▶ Транспортни системи
- ▶ Медицинско оборудване
- ▶ Сензорни мрежи

# Приложения - телекомуникации

- ▶ Телефонни централи
- ▶ Мобилни телефони
- ▶ Рутери
- ▶ Суичове

# Приложения – домакинска електроника

- ▶ Микровълнови фурни
- ▶ Миялни машини
- ▶ Перални
- ▶ Хладилници
- ▶ Охранителни системи
- ▶ Home automation системи

# Приложения - Транспорт

- ▶ Бордови компютри
- ▶ Светофарни уредби
- ▶ Управление на двигатели (електрически и ДВГ)
- ▶ **ABS - anti-lock braking system** (Микрокомпютъра обработва получената информация и чрез електромагнитни клапани управлява налягането на спирачната течност (флуид) към колелата. Когато едно колело (или повече) при спиране е на път да блокира, системата намалява налягането към спирачния му механизъм, отпуска го и мигновено отново увеличава налягането, като го поддържа на границата на блокиране. Това водачът на превозното средство го усеща като пулсации в спирачния педал, при което вибрира и целият автомобил)
- ▶ **ESP -Electronic Stability Program** (ESP препятства нестабилността на автомобила като разпознава критичните пътни ситуации (напр. опасност от поднасяне), задейства задната или предна външна на завоя спирачка и управлява двигателя и скоростната кутия.)
- ▶ Traction control
- ▶ Cruise control

Антиблокиращата система (ABS) е система за сигурност в автомобилите, която предотвратява блокирането на колелата при спиране на автомобила. Това помага да се поддържа контрол върху автомобила и да се намали дистанцията за спиране.

Ето някои от основните принципи на работа на ABS при автомобил:

1. Сензори за скорост на колелата: ABS използва сензори, които монтират на всяко колело и измерват скоростта на въртене на колелото. Те изпращат информация към ABS управляващата единица.
2. Анализ на скоростта на колелата: ABS управляващата единица анализира скоростта на въртене на колелата и сравнява скоростта на всяко колело със скоростта на другите колела. Ако забележи значително разлика в скоростта на едно или повече колела, системата може да предположи, че колата започва да блокира.
3. Регулиране на спирачките: Ако ABS управляващата единица забележи блокиране на колело, тя превключва на режим на регулиране на спирачките. Това се постига чрез управление на хидравличното налягане върху спирачките на блокиращото се колело, което намалява налягането и позволява колелото да се върти отново.
4. Поддържане на контрола върху автомобила: Като регулира налягането върху спирачките на блокиращите се колела, ABS помага да се поддържа контрол върху автомобила по време на спиране. Това позволява на шофьора да продължи да маневрира и да избегне препятствия, като в същото време намалява дистанцията за спиране.

ABS е важна система за сигурност в автомобилите, която помага да се предотврати блокирането на колелата и да се поддържа контрол върху автомобила при спиране. Това значително подобрява сигурността и управляемостта на автомобила.

Електронната програма за стабилизиране (EPS) е система за сигурност в автомобилите, която помага да се поддържа стабилността и контрола върху автомобила по време на маневриране. Тя осигурява подпомагане на волана и коригира силата на въртене на предните колела, за да помогне на шофьора да задържи автомобила на прав път и да избегне изпълзване или прекомерно завиване.

Ето някои от основните принципи на работа на EPS при автомобил:

1. Сензори за волана: EPS използва сензори, които монтират на волана и измерват силата и посоката на въртене на волана от страна на шофьора. Те изпращат информация към EPS управляващата единица.

2. Анализ на волана: EPS управляващата единица анализира информацията от сензорите за волана и определя силата и посоката на въртене на волана от страна на шофьора. Тя също така сравнява въртенето на волана с други параметри като скоростта на автомобила и наклонът на пътя.

3. Корекция на силата на въртене: Ако EPS управляващата единица забележи, че въртенето на волана от страна на шофьора не съответства на параметрите на автомобила, тя коригира силата на въртене на предните колела. Това се постига чрез управление на електромотор, който помага на волана да се върти по-лесно и да се коригира посоката на колелата.

4. Поддържане на стабилността и контрола: Като коригира силата на въртене на предните колела, EPS помага да се поддържа стабилността и контрола върху автомобила по време на маневриране. Това помага на шофьора да задържи автомобила на прав път, да избегне изпълзване и да се справи с прекомерно завиване.

EPS е важна система за сигурност в автомобилите, която помага да се поддържа стабилността и контрола върху автомобила по време на маневриране. Тя подпомага шофьора и предотвратява потенциални опасности, свързани със загуба на контрол или изпълзване на автомобила.

# Приложения – медицина

- ▶ Животоподдържащи системи
- ▶ Електронни стетоскопи
- ▶ Образна диагностика

# Вградени системи – Потребителски интерфейс

- ▶ Без потребителски интерфейс
- ▶ LED индикация
- ▶ Бутони/Клавиатура/Джойстик
- ▶ LCD екран (текстови/графичен)
- ▶ Тъчскрийн

# Вградени системи – Потребителски интерфейс

- ▶ Без потребителски интерфейс
- ▶ LED индикация
- ▶ Бутони/Клавиатура/Джойстик
- ▶ LCD екран (текстови/графичен)
- ▶ Тъчскрийн

# Вградени системи - SoC

- ▶ SoC – System-On-Chip - **Едночипова система**
- ▶ **ASIC - Application-Specific Integrated Circuit** - която е специално пригодена за конкретна употреба, а не за общо предназначение. Например, чип, предназначен за работа в цифров диктофон
- ▶ **FPGA - Field-Programmable Gate Array**
  - цифрова интегрална схема, съдържаща програмируема цифрова логика. Използват се програмните езици Verilog и VHDL. Те осигуряват необходимото бързодействие за голяма група от приложения като тактовата им честота е в обхвата между 50 MHz и 400 MHz

# Вградени системи

- ▶ Microcontroller
- ▶ Микропроцесор – (ARM, DSP, PIC, ...)
- ▶ ROM / Flash памет
- ▶ RAM
- ▶ Входно/изходни интерфейси

# Вградени системи - Периферии

- ▶ Серийна комуникация – RS232, UART, RS485
- ▶ Синхронна серийна комуникация – I2C, SPI
- ▶ USB
- ▶ MMC, SD карти
- ▶ Мрежова – Ethernet
- ▶ Индустриски мрежи – CANBus, LINbus
- ▶ GPIO - General Purpose Input/Output
- ▶ ЦАП и АЦП (Цифрово-аналоговият преобразувател)
- ▶ Debug - JTAG

# Вградени системи - Софтуер

- ▶ Firmware
- ▶ Операционна система
  - Real-Time Operating System
  - Linux
  - Windows CE
  - други

# FW vs OS(SW)

- ▶ **Firmware**
  - ▶ ☐ Високо бързодействие
  - ▶ ☐ Повече време за разработка
  - ▶ ☐ Платформено зависимо
  
- ▶ **Operating system**
  - ▶ ☐ Сравнително по-бавно бързодействие от FW
  - ▶ ☐ По-малко време за разработка
  - ▶ ☐ Платформено независимо

# Real-Time Operating System

- ▶ Намалено време за влизане в прекъсване
- ▶ □ Намалено време за превключване между паралелни задачи
- ▶ □ Подходяща за управление на критични във времето процеси

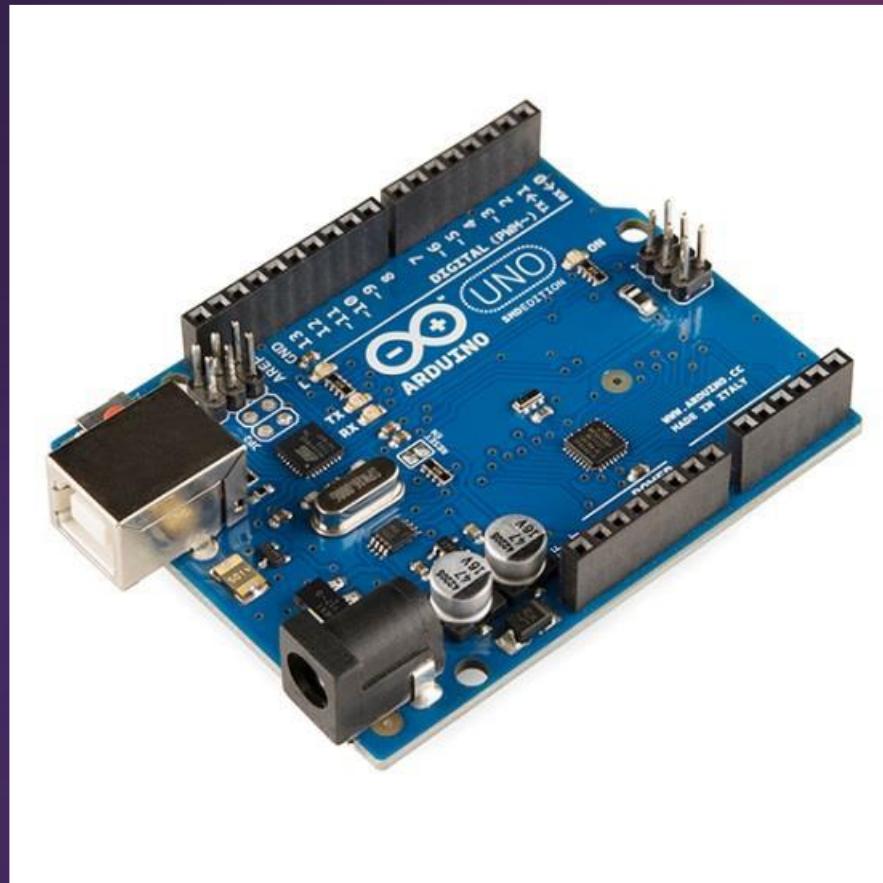
# Linux for embedded

- ▶ uCLinux
- ▶ Embedded Debian
- ▶ Rasbian
- ▶ Android
- ▶ BusyBox – пакет с основни UNIX инструменти

# Представители на Open Source Hardware

- ▶ Arduino
- ▶ Raspberry Pi
- ▶ BeagleBone
- ▶ ... и многое други

# Arduino Uno



Arduino

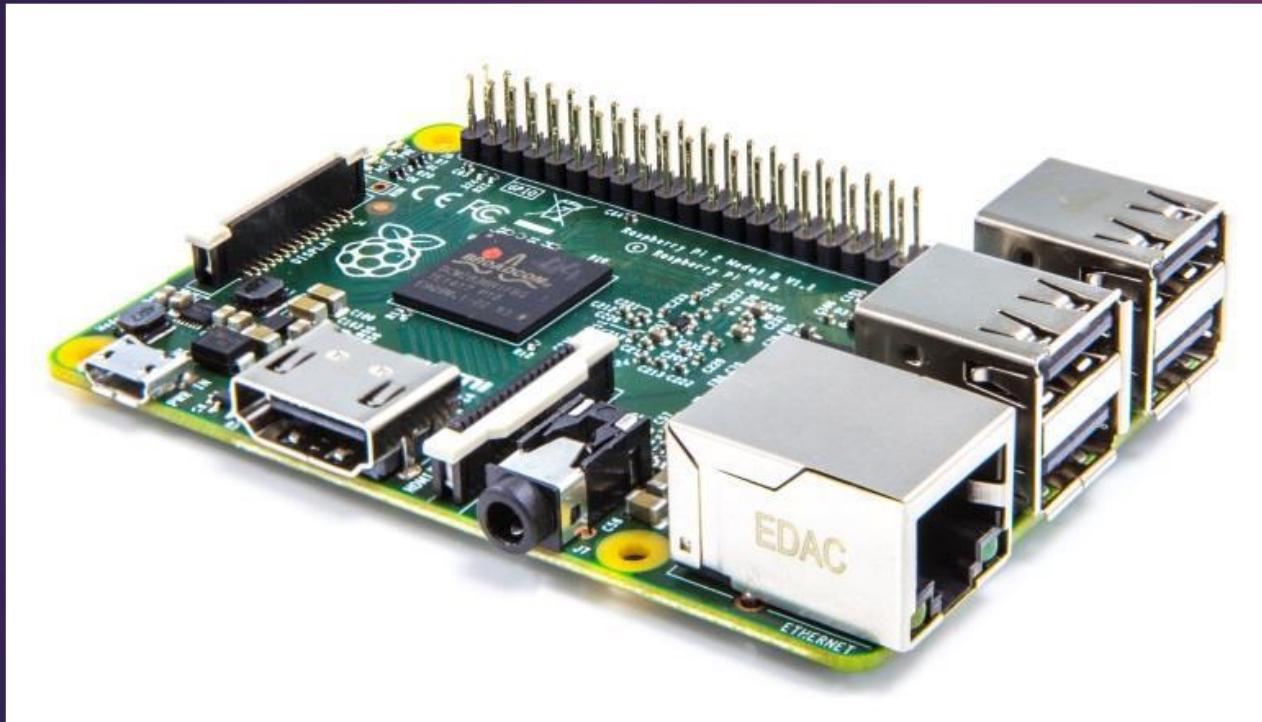


The image shows the Arduino Uno SMD R3 microcontroller board, which is a blue single-board microcontroller. It has a standard pinout with two rows of pins on the top and bottom. The board is populated with surface-mount components, including a microcontroller chip, memory chips, and various connectors like USB and audio jacks.

Arduino Uno SMD R3

<b>Developer</b>	<a href="#">arduino.cc</a>
<b>Manufacturer</b>	Arduino
<b>Type</b>	Single-board microcontroller
<b>Operating system</b>	None (default) / Xina
<b>CPU</b>	Atmel AVR (8-bit) ARM Cortex-M0+ (32-bit) ARM Cortex-M3 (32-bit) Intel Quark (x86) (32-bit)
<b>Memory</b>	SRAM
<b>Storage</b>	Flash, EEPROM
<b>Website</b>	<a href="#">arduino.cc</a>

# Raspberry Pi 2



Digital Ports: 40

- PWM: 1
- Analog Inputs: 0
- Serial Ports: 1
- USB Host, Etherner, Video out, HDMI
- Serial Camera Interface
- Serial Display Interface

Raspberry Pi е единична платка компютър, която може да бъде използвана за различни приложения. Ето някои от тях:

1. Медиа център: Raspberry Pi може да бъде използван като медиа център за стриймване на филми, музика и снимки на телевизора. **Можете да инсталирате операционна система като Kodi или Plex и да използвате Raspberry Pi за гледане на филми и сериали от различни източници.**
2. Умна къща: Raspberry Pi може да бъде използван за автоматизиране на домашните устройства и системи. **Можете да свържете Raspberry Pi с различни сензори и актуатори, за да контролирате осветление, отопление, охранителни системи и други устройства в дома си.**
3. Интернет на нещата (IoT): Raspberry Pi може да бъде използван за създаване на IoT устройства и приложения. **Можете да свържете Raspberry Pi с различни сензори и актуатори, за да събирате данни и да контролирате устройства през интернет.**
4. Образование: Raspberry Pi се използва често в образователни институции, за да се насърчи интересът и уменията в областта на програмирането и компютърните науки. **Можете да използвате Raspberry Pi за създаване на различни образователни проекти и упражнения.**
5. Роботика: Raspberry Pi може да бъде използван за създаване на роботи и автономни системи. **Можете да свържете Raspberry Pi с различни сензори и актуатори, за да контролирате движението и поведението на роботите.**

1. Пример за светещ светодиод:

```
import RPi.GPIO as GPIO
import time

LED_PIN = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)

try:
    while True:
        GPIO.output(LED_PIN, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(LED_PIN, GPIO.LOW)
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

Този код използва библиотеката RPi.GPIO, за да свети светодиода, свързан с пин 17 на Raspberry Pi. Свети 1 секунда, след което се изключва за 1 секунда, и така непрекъснато, докато не бъде прекъснат с клавишка комбинация.

1. Пример за четене на стойност от сензор на Raspberry Pi:

```
import RPi.GPIO as GPIO

SENSOR_PIN = 18

GPIO.setmode(GPIO.BCM)
GPIO.setup(SENSOR_PIN, GPIO.IN)

try:
    while True:
        if GPIO.input(SENSOR_PIN):
            print("Детектирано движение")
        else:
            print("Няма движение")
except KeyboardInterrupt:
    GPIO.cleanup()
```

Този код използва библиотеката RPi.GPIO, за да чете стойността на сензора, свързан с пин 18 на Raspberry Pi. Ако сензорът детектира движение, се извежда съобщение "Детектирано движение", в противен случай се извежда "Няма движение".

## Raspberry Pi 1



Raspberry Pi 1 модел B+

<b>Дата на издаване</b>	февруари 2012; преди 11 години
<b>Начална цена</b>	US\$25 (model A, B+[1]), US\$20 (model A+), US\$35 (RPi 1 model B, RPi 2 model B), US\$30 (CM)
<b>Операционна система</b>	Linux (Raspbian), RISC OS, FreeBSD, NetBSD, Plan 9, Inferno, AROS
<b>Процесор</b>	700 MHz single-core ARM1176JZF-S (model A, A+, B, B+, CM)[2]
<b>Памет</b>	256 MB (model A, A+, B rev 1) 512 MB (model B rev 2, B+, CM)
<b>Архитектура</b>	SDHC slot (model A and B), MicroSDHC slot (model A+ and B+), 4 GB eMMC IC chip (model CM)
<b>Графика</b>	Broadcom VideoCore IV[2]
<b>Захранване</b>	1.5 W (model A), 1.0 W (model A+), 3.5 W (model B) or 3.0 W (model B+)

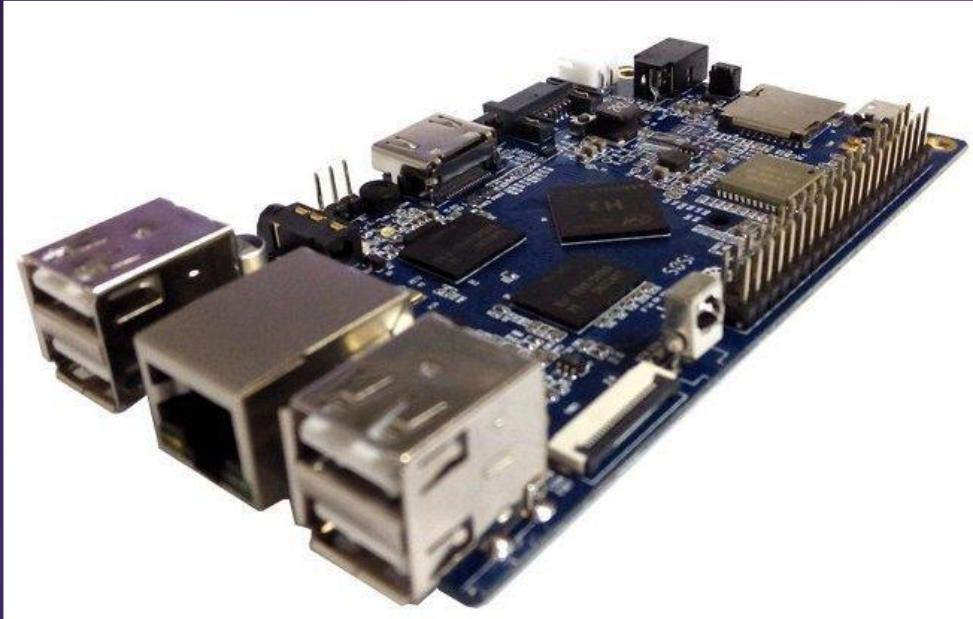
## Arduino



Arduino Uno SMD R3

<b>Developer</b>	<a href="http://arduino.cc">arduino.cc</a>
<b>Manufacturer</b>	Arduino
<b>Type</b>	Single-board microcontroller
<b>Operating system</b>	None (default) / Xinu
<b>CPU</b>	Atmel AVR (8-bit) ARM Cortex-M0+ (32-bit) ARM Cortex-M3 (32-bit) Intel Quark (x86) (32-bit)
<b>Memory</b>	SRAM
<b>Storage</b>	Flash, EEPROM
<b>Website</b>	<a href="http://arduino.cc">arduino.cc</a>

# И неговите клони

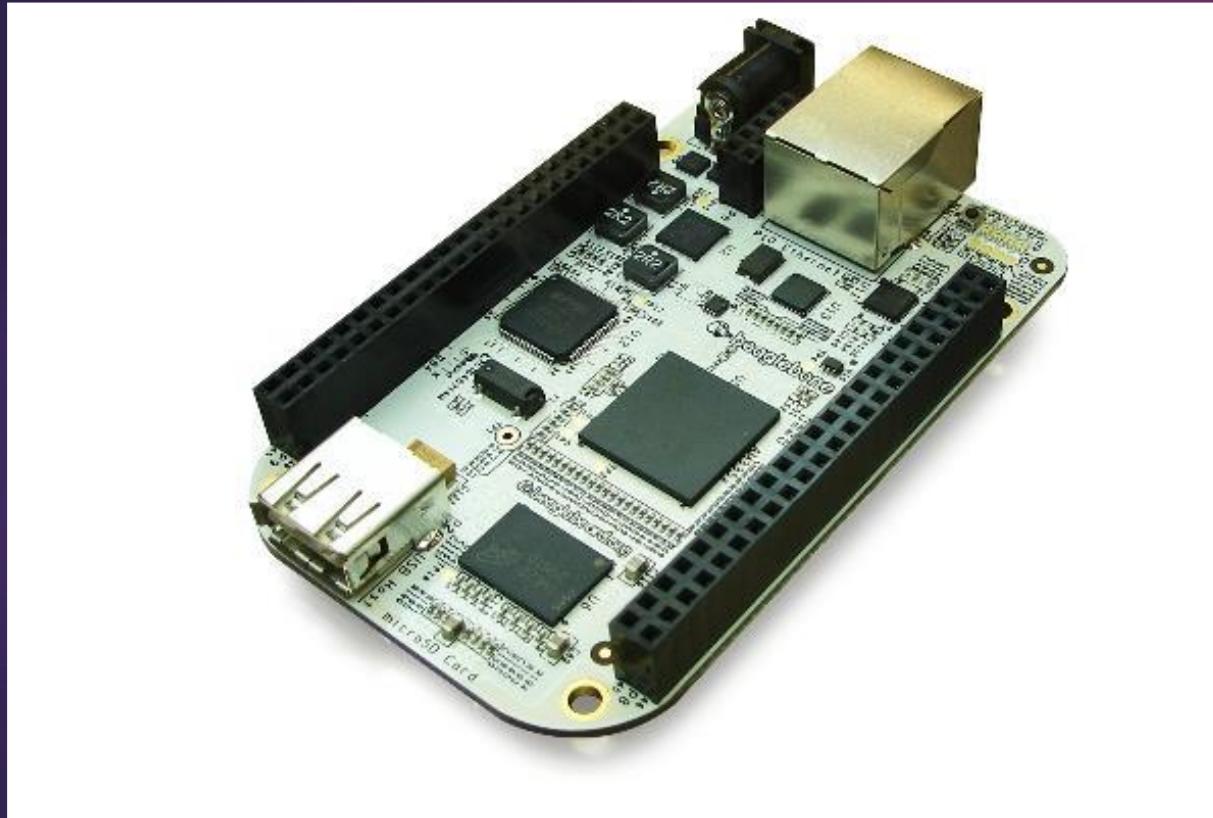


Orange Pi



Banana Pi

# BeagleBone Black



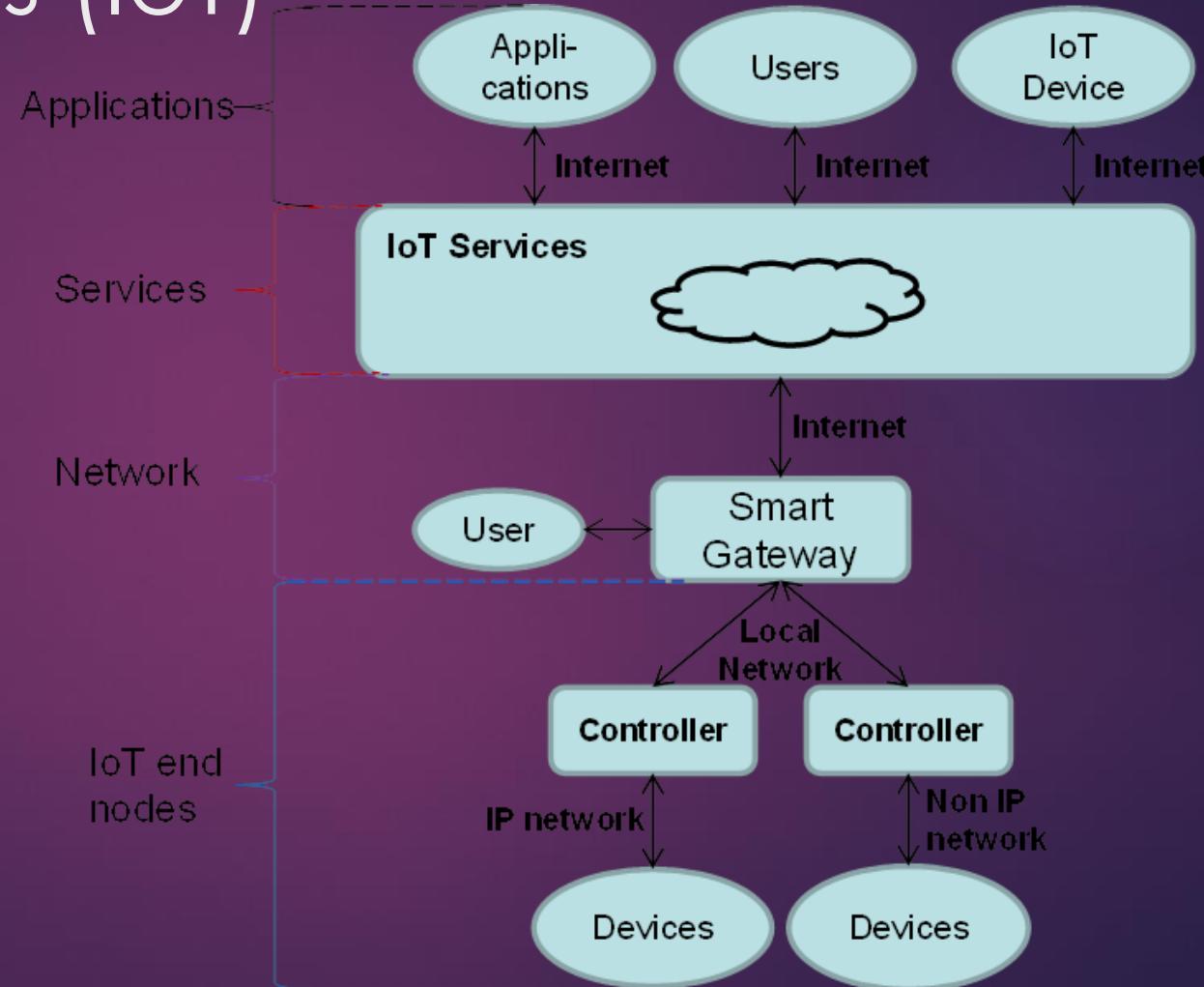
Digital Pins: 23

- PWM: 8
- Analog Inputs: 8
- Serial Ports: 6
- USB Host, Ethernet, Video out (external), HDMI (external)

# Internet Of Things (IoT)



- ▶ Концепция, при която множество от вградени системи са свързани към Internet.
- ▶ Отдалечено управление и обмяна на информация
- ▶ Cloud системи за управление
- ▶ Internet



# Приложения на IoT

- ▶ Екологично наблюдение – вода, въздух, почва
- ▶ Надзор на инфраструктурата – състоянието на пътища, мостове, вятърни турбини
- ▶ Производство и дистрибуция
- ▶ Автоматизация на дома
- ▶ Развитие на медицинската апаратура
- ▶ Транспорт – управление на трафика

# Основни характеристики на хардуерните и софтуерни компоненти на embedded системите

- ▶ Отличителна характеристика на вградените системи е ограниченият брой предварително дефинирани функции, които са проектирани да изпълняват
- ▶ **Интерфейси във вградените системи**
- ▶ **Платформи за изграждане на вградени системи**
- ▶ **Хардуерни компоненти на системите**

# Хардуерни компоненти на системите

- ▶ Процесор. За целите на вграденото управление намират приложение RISC процесори. Водещ принцип при концепцията RISC (Reduced Instruction Set Computer – компютър с намален набор инструкции) е изпълнението на максимално компактни и опростени команди, които изискват минимално време (тактове). Този принцип опростява архитектурата, позволява процесорът да работи на по-висока тактова честота и намалява цената му. За целите на вграденото управление са добили популярност различни фамилии RISC процесори: ARC (на ARC International), ARM (на ARM Holdings), AVR (на Atmel), PIC (на Microchip), MSP430 (на TI) и др. Като противоположност на RISC съществува концепцията CISC (Complex Instruction Set Computer – компютър с комплексен набор инструкции), но тя е с по-ограничено приложение при вградените системи. По отношение дължината на думата на процесорите, използвани за вградено управление, съществува голямо разнообразие – те могат да са от 4- до 64-битови и повече (главно при DSP – цифрови сигнални процесори), но в най-масова употреба са 8- и 16-битовите.

# Комуникация

Комуникационните възможности при вградените системи се характеризират с голямо разнообразие:

- Серийни комуникационни интерфейси - RS-232, RS-422, RS-485. Използват се главно за комуникация с компютри или с други вградени системи;
- UART/ USART (Universal Synchronous and Asynchronous Receiver/Transmitter – универсален синхронен и асинхронен приемник/предавател).
- Синхронни серийни комуникационни интерфейси. Намират приложение като прям обмен между микроконтролери, за първоначално програмиране на чипа (зареждане на програма и на данни) и др. Към тази група спадат I2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface Bus), SSC и ESSI (Enhanced Synchronous Serial Interface).
- USB (Universal Serial Bus) - един наложил се в последните години комуникационен интерфейс, с който са снабдени голяма част от съвременните устройства;
- Протоколи за мрежова комуникация като Ethernet, CAN (Controller Area Network), LonWorks и др. Наличието на такава периферия в микроконтролера силно редуцира броя на допълнителните външни компоненти, опростява печатната платка и снижава цената на вградената система.

# Програмно осигуряване на вградените системи

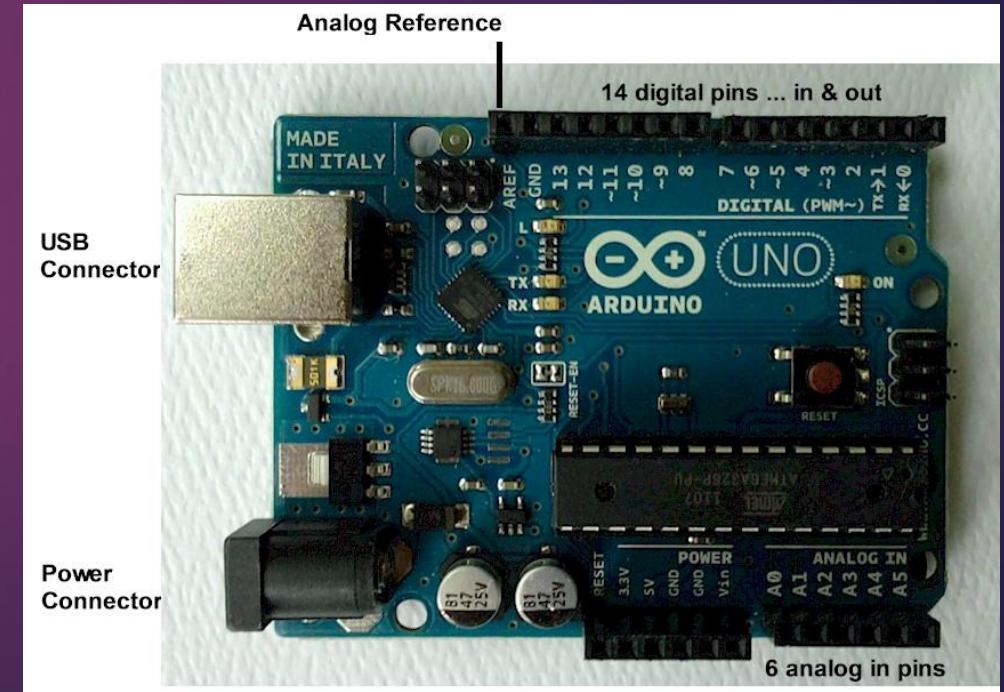
- ▶ Програмното осигуряване за вградените системи е познато като фърмуер (firmware) или "твърда" програма, тъй като е записано в енергонезависима памет и в идеалния случай се предполага, че няма да се променя в процеса на експлоатация на системата, който може да бъде десетки години. В много съвременни системи програмите са записани във флашпамет и могат (при определени мерки за сигурност) да се подменят с цел модернизация. Това дава възможност за развитие на вградените системи, които вече са в експлоатация, но създава и предпоставки за злоупотреби. Сред най-популярните операционни системи (ОС) за вградените системи са QNX4 RIOS, Embedded Linux, Windows CE и др.

# Инструменти за тестване и коригиране

Процесът на изпитване и корекция на програмното осигуряване, наричан още дебъгинг (debugging), отнема голяма част от усилията и времето на разработчиците. Поради тази причина, всички производители на микропроцесори и микроконтролери за вградени системи са предвидили инструменти, чрез които той да бъде улеснен. Комплексът от средства за дебъгинг включва три съставни части. Първата е наличието на необходимата периферия в микроконтролера. Широко разпространение са добили JTAG (Joint Test Action Group) – специализиран интерфейс за изпитване на наситени печатни платки; ISP (In-System Programming) – вътрешносхемно програмиране; ICSP (In Circuit Serial Programming) – метод за директно програмиране на микроконтролери, например от сериите PIC и AVR; BDM (Background Debug Mode) – интерфейс, използван главно в продуктите на Freescale и др. Втората съставна част представлява специализиран хардуер (вградена система във вид на печатна платка или самостоятелно устройство), чрез който се извършва програмиране, симулиране или емулиране. Третият елемент от средствата за дебъгинг е програмното осигуряване за персонален компютър. Популярност са добили софтуерни пакети от типа IDE (Integrated Development Environment) – интегрирана среда за разработка. Те дават възможност за писане на програми, за записването им в паметта на микроконтролерите или за прочитането им, за симулиране действието на програмата и поведението на цялата вградена система и др.

# Ардуино

- ▶ Ардуино е проект с отворен код и едноименна компания, произлязла от него, както и свързаното с него потребителско общество. Основната цел е проектиране и производство на електронна платформа с лесен за ползване свободен хардуер и софтуер, позволяващи постигането на интерактивност за неспециалисти.



- ▶ **Arduino Leonardo** – Леонардо е една от най-новите платки на Arduino. Има същия вид като останалите, но микроконтролерният чип е различен. Той дава предимството платката да бъде разпозната като [клавиатура](#) или [мишка](#) от дадения компютър.
- ▶ **Arduino Uno** – микроконтролерна развойна платка с ATmega328P AVR микроконтролер. Има 14 цифрови входно-изходни (I/O) порта, 6 аналогови входа, 16 MHz квартов резонатор, четири светодиода (един потребителски, свързан на 13-ти цифров I/O порт и три, които индикират работата на платката: ON, Tx и Rx), USB конектор, захранващ куплунг, бутон за рестартиране и ICSP конектор. Шест от цифровите I/O порта могат да се използват като PWM (ШИМ) изходи. Свързването с компютър се осъществява чрез USB кабел USB A – USB B. Uno може да се захранва през USB порта на компютъра или от външен източник, като превключването между различните начини за захранване е автоматично. Външният източник на захранване може да е DC адаптер 7-12V или [батерия](#).
- ▶ **Arduino Mega 2560 R3** – както подсказва името ѝ, тази платка е по-голяма от Uno варианта. Създадена е за хора, които изискват повече – повече входове, повече изходи и повече процесорна сила.
- ▶ **Arduino Mega ATD** – тази платка е същата като Arduino Mega 2560, но е проектирана за работа с мобилни устройства с [Андроид](#). Когато се споделя информация между Андроид устройствата (телефони или таблети), Arduino разширява обхвата на всяко едно от тях.
- ▶ **Arduino Nano 3.0** – Arduino Nano копира Uno, но е с размери само 1,8 см x 4,3 см. Този размер е идеален за направата на по-малки проекти. Нано има цялата спецификация на Arduino Uno, използва същия ATmega328 микроконтролер, но е по-оптимизирана като размер и това го прави идеалния [прототип](#).
- ▶ **Arduino Mini R5** – както подсказва името, Мини вариантът на Arduino е по малък и от Nano. Тази платка също използва микроконтролерния чип ATmega328, но от към дизайн е още по-оскъдна, защото са премахнати всички колекторни щифтове, както и Mini-USB конектора на Nano варианта. Тази платка е идеалният избор, ако мястото е приоритет, но създава много грижи при свързване, защото едно грешно свързване може да унищожи платката.
- ▶ **Arduino Ethernet** – е като Arduino Uno, но е специално пригодена за свързване с Интернет. Може да се настрои за автоматичен достъп без значение на трафика от данни. Уеб браузърите интерпретират зададения текст точно както се изобразява на екрана ви (например: подравняване, форматиране, използване на изображения и т.н.). Ако въведените команди са познати, Arduino Ethernet може да достъпи до текста директно и да бъде използван и за други цели. За използването на тази платка могат да се намери много документация онлайн.
- ▶ **Arduino BT** – тази платка позволява свързване с [Bluetooth](#) устройства в обхвата ѝ и дава възможност за взаимодействие с мобилни телефони, таблети и други устройства, притежаващи тази екстра.

- Мультиплатформен
- Java , C, C++

```
#define LED_PIN 13

void setup() {
    pinMode(LED_PIN, OUTPUT);          // Enable pin 13 for digital output
}

void loop() {
    digitalWrite(LED_PIN, HIGH);       // Turn on the LED
    delay(1000);                     // Wait one second (1000 milliseconds)
    digitalWrite(LED_PIN, LOW);        // Turn off the LED
    delay(1000);                     // Wait one second
}
```

# Разширения

GPS, Ethernet, LCD

# Raspberry Pi

- ▶ Raspberry Pi или RPI е серия от едноплаткови компютри с размери на кредитна карта, разработена в Обединеното кралство от специално създадена за целта фондация (Raspberry Pi Foundation) с цел популяризиране на обучението по основи на компютърните науки в училищата.



# Raspberry Pi

[Linux](#) (Raspbian), [RISC OS](#), [FreeBSD](#), [NetBSD](#), [Plan 9](#), Inferno, AROS

## Raspberry Pi 2



Raspberry Pi 2 model B

**Дата на**

февруари 2015;

**издаване**

преди 3 години

**Начална**

US\$35

**цена**

**Операционна**  
**система** Same as for Raspberry Pi 1 plus Windows 10 IoT Core<sup>[3]</sup> and additional distributions of Linux such as Ubuntu

**CPU** 900 MHz quad-core ARM Cortex-A7

**Памет** 1 GB RAM

**Архитектура** MicroSDHC slot

**Графика** Broadcom VideoCore IV

**Захранване** 4.0 W

## Arduino Software IDE

```
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

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

void setup() // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop() // run over and over again
{
  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
}
```

Done compiling.  
Binary sketch size: 1096 bytes (of a 14396 byte maximum)

Скриншот на Arduino IDE с програмата „Blink“,  
проста програма за начинаещи

### Информация

**Разработчик** Arduino Software

**Последна**  
**версия** 1.6.3

**Програмен**  
**език** Java, C и C++

**Операционна**  
**система** мултиплатформена

**Вид софтуер** интегрирана среда за разработка

**Лиценз** LGPL или GPL лиценз

**Уебсайт** <http://arduino.cc/en/Main/Software>

Ето един примерен код за Arduino, който включва светодиод и го включва и изключва в интервал от 1 секунда:

```
// Пинът, на който е свързан светодиода
int ledPin = 13;

void setup() {
    // Задаване на пинът като изходен
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // Включване на светодиода
    digitalWrite(ledPin, HIGH);
    delay(1000); // Изчакване 1 секунда

    // Изключване на светодиода
    digitalWrite(ledPin, LOW);
    delay(1000); // Изчакване 1 секунда
}
```

Този код използва пин 13 на Arduino за свързване на светодиода. В функцията `setup()` задаваме пина като изходен, а в функцията `loop()` включваме и изключваме светодиода, като използваме функциите `digitalWrite()` и `delay()` за задаване на времето за включване и изключване.

Ето няколко примерни кодови сегменти, които можете да използвате на Raspberry Pi:

1. Пример за светещ светодиод:

```
import RPi.GPIO as GPIO
import time

LED_PIN = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)

try:
    while True:
        GPIO.output(LED_PIN, GPIO.HIGH)
        time.sleep(1)
        GPIO.output(LED_PIN, GPIO.LOW)
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

Този код използва библиотеката `RPi.GPIO`, за да свети светодиода, свързан с пин `17` на Raspberry Pi. Свети 1 секунда, след което се изключва за 1 секунда, и така непрекъснато, докато не бъде прекъснат с клавишка комбинация.

1. Пример за четене на стойност от сензор на Raspberry Pi:

```
import RPi.GPIO as GPIO

SENSOR_PIN = 18

GPIO.setmode(GPIO.BCM)
GPIO.setup(SENSOR_PIN, GPIO.IN)

try:
    while True:
        if GPIO.input(SENSOR_PIN):
            print("Детектирано движение")
        else:
            print("Няма движение")
except KeyboardInterrupt:
    GPIO.cleanup()
```

Този код използва библиотеката `RPi.GPIO`, за да чете стойността на сензора, свързан с пин `18` на Raspberry Pi. Ако сензорът детектира движение, се извежда съобщение "Детектирано движение", в противен случай се извежда "Няма движение".