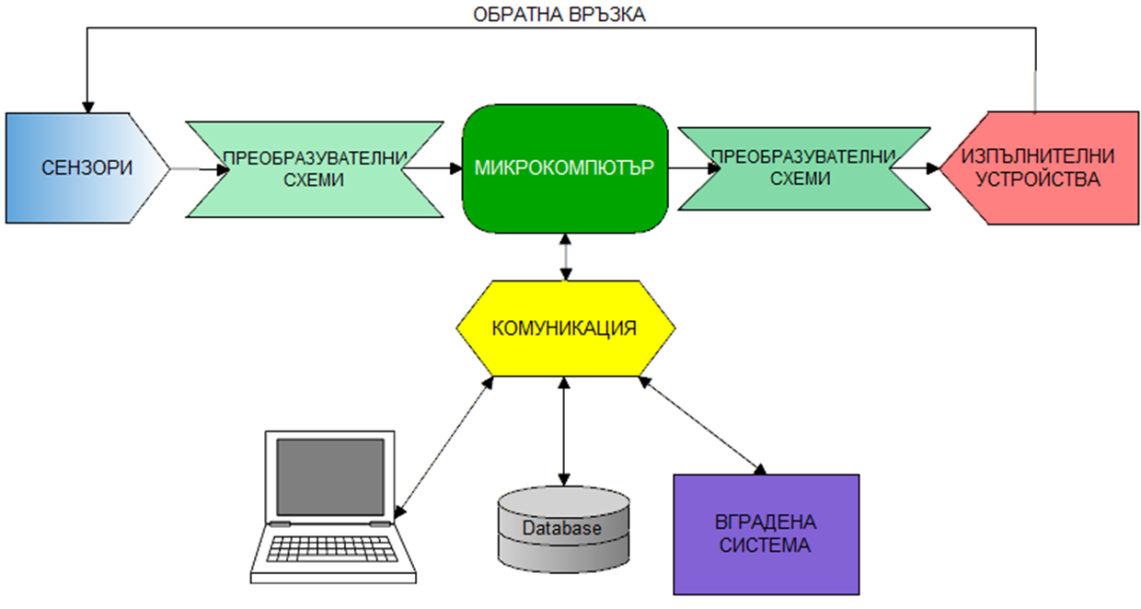
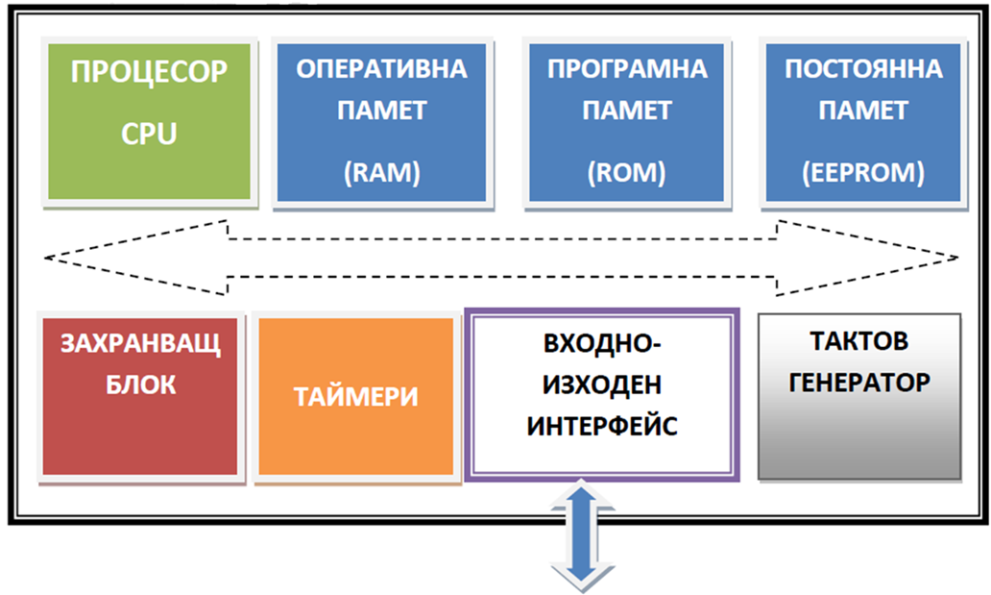
## Изпитна тема № 13: Вградени системи

**План-тезис:** Основи на електротехниката и електрониката. Електронни елементи: резистори, кондензатори, светодиоди, бутони и потенциометри. Аналогово/цифрови входове и изходи. Сензори. Управление на периферия. Широчинно импулсна модулация.

**Вградена система (Embedded system)** - система, в която има вграденен компютърен модул който я управлява и анализира

* Възможно е да бъде както самостоятелна система, така и част от по-голяма такава
* Най-важното нещо във всяка вградена система е **микрокомпютърът** . Той представялява програмируемият модул, в който е заредена програмата за управление на системата
  + Пример: едноплатков компютър, микроконтролер и т.н.
* **Сензорите (датчиците)** са устройства, които преобразуват определена физическа величина в електрически сигнал, който може да бъде разчетен от програмируемото устройство
  + Пример: сензори за температура, сензор за влажност и т.н.
* **Изпълнителните устройства** извършват определена полезна работа под управление на програмируемото устройство
  + Пример: електродвигатели, дисплеи, нагреватели и т.н.
* **Преобразувателните схеми** служат за допълнителна обработка на електрическите сигнали
  + Пример: усилватели, филтри, драйвери и т.н.
* Една вградена система не е необходимо да съдържа всички от изброените елементи
* Етапи не разработка
  + Проектиране и създаване на хардуер
  + Разработка на управляващ софтуер

**Микроконтролер** - едночипова (в корпуса на една интегрална схема) компютърна система. Един от най-използваните модули в съвременните вградени системи

* + малка консумация на ел. енергия
* + малки размери
* + ниска цена
*  - ниска изчислителна мощност
* Програмиране
  + Програмата се създава в среда за разработка, която обикновено се предоставя от производителя на микроконтролера. Нейната цел е да преведе сорс кода от езика от високо ниво до машинен код - *\*.hex* файл.
  + \*.hex файлът се изпраща през порт от компютъра до специално устройство наречено програматор и се записва в програмната памет на микроконтролера.

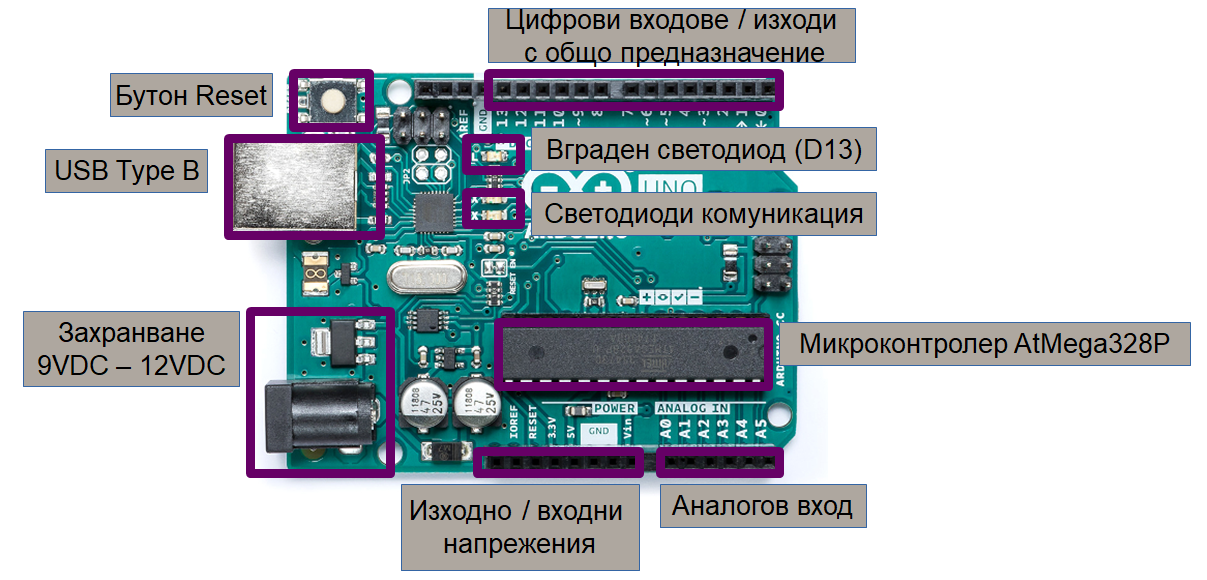
**Физични величини**

* **Електрическо напрежение, U**
  + Дефинира се между две точки;
  + U = φ2 – φ1 [V] ;
  + Мерна единица – ВОЛТ [V]
  + Измерване чрез успоредно свързам волтметър
* **Електричен ток, I**
  + Електричен заряд q преминал за единица време t:
  + I = dq / dt [A]
  + Мерна единица – ампер [A]
  + Посока - от по-висок към по-нисък потенциал
  + Измерване чрес последователно свързам амперметър
* **Съпротивление, R**
  + противопоставя се на протичането на електричния ток;
  + Среща се в консуматори, неиделани проводници, източници
  + Мерна единица – ОM [Ω]
  + Измерване чрез успоредно свързам омметър
* **Мултицет (Мултимер)** - комбиниран преносим цифров измервателен уред
* **Осцилоскоп, Спектороанализатор** - освен количествена оценка, позовляват и графична визуализация на измерваниете величини

**Електроника** - инженерна наука, чиято цел е генериране, разпространение и управление на електрическа енергия

* **Проводник** – пропуска ток
  + има свободни електрически заряди (токоносители)
  + метали, електролити, йонизиран газ.
* **Полупроводник** – при едни условия е проводник, при други е диелектрик
  + силиций, германий, GaAs….
  + **Основа на компютърните технологии**
* **Диелектрик (Изолатор)** – не пропуска ток
  + няма свободни електрически заряди
  + гума, стъкло, пластмаси, бакелит, дърво(сухо), масла, дестилирана вода, въздух...
* **Консуматор** - преобразува електрическата енергия в друг вид енергия
* **Електрическа верига** - съвкупност от елементи и преносни линии, които имат за цел да доставят и преобразуват електрическа енергия от източника до консуматора
* **Резистор -** пасивен електронен елемент, който се характеризира със съпротивление
  + Основно предназначение – да ограничава тока в електрическите вериги или да създава пад на напрежение
  + Основни параметри:
  + електрическо съпротивление R
  + мощност, която може да разсее
  + клас точност
  + паразитни параметри
* Светодиод
  + Обикновени диоди (едноцветни)
  + LED – RGB светлина
  + Когато светодиодите се захранват от източници на напрежение, е задължително последователно на тях да се свържат резистори
  + На излъчва (свети) през целия период, а само по време на активното ниво на импулса
  + Излъчва с един и същ интензитет през цялото време
  + Колкото по-дълго е включен, толкова по-ярко го „виждаме“ да излъчва
* Ключове и бутони
  + Служат за прекъсване на електрическа верига
  + Основен електрически параметър е максимален ток на превключване и пробивно напрежение
* **Потенциометър** - вид съпротивление, чиято стойност може да бъде променяна динамично по време на работа на усторйството
  + Съставен е от три терминала (извода) – два захранващи и един изходен
  + Промяната на стойността на съпротивлението е плавна
* **Транзистори** - полупроводников електронен елемент, при който може с малък ток/напрежение да се управлява по-голям ток
  + В зависимост от управляващата величина има два вида транзистори:
    - биполярни (управляват се по ток)
    - полеви (управляват се по напрежение)

**Ардуино** - микроконтролерна платка, съдържаща в себе си захранващ модул, USB порт за комуникация с компютър, микроконтролер и изходни рейки за свързване на периферни устройства.

* Улеснява процеса на създаване на микропроцесорни системи, като предоставя единна унифицирана платформа
* Предоставя отворен код и принципни електрически схеми
* Цифрови входове и изходи – 0 до 13
  + Управление на светодиоди
  + Управление на електромотори
  + Комуникация със сензори
  + Управление на дисплей
  + Комуникация с GSM модули
  + Комуникация с GPS модул;
* Аналогови входове - А0 до А5
  + Четене на температурен сензор
  + Четене на светлинен сензор
  + Четене на потенциометър
* Вградени функции и константи
  + **void setup()** - изпълнява се веднъж при стартиране, използва се за инициализация на контролера;
  + **void loop()** - безкраен цикъл, педставлява същинската функционалност
  + **HIGH** - 5 V
  + **LOW** - 0 V
  + **INPUT** - Дефиниране на пин, като вход
  + **OUTPUT** - Дефиниране на пин, като изход
  + **void pinMode(pin, direction)**
    - pin - номер на пин (0 – 13, А0 - А5)
    - direction - посока на пин (INPUT / OUTPUT)
  + **analogWrite(pin, value)**
    - pin - номер на пин
    - value - коефициент на запълване (от 0 - напълно изключен, до 255 - напълно включен)
  + **int analogRead(pin)**
    - pin - номер на пин за аналогов вход (А0 - А5)
    - връща стойност, прочетена от аналогово-цифровия преобразувател (0 - 1023)
* Широчинно-импулсна модулация (ШУМ, PWM)
  + управлява скоростта (оборотите) на въртене на електромотор;
  + управлява количеството топлина – температурата за загряване / охлаждане
  + управлява ъгълът на отклонение на СЕРВО двигател;
  + управлява цветът на RGB светодиод
* Аналогов сигнал
  + Има безкраен брой стойности във времето
  + Непрекъснат сигнал
  + Не може да бъде обработван от цифровата схемотехника – микропроцесори - те работят с конкретни и точно дефинирани числа
  + Нужни са допълнителни стъпки за преобразуване на аналогов сигнал до такъв, подходящ за микроконтролер
  + Процесът на преобразуване на аналогов сигнал в цифров се нарича дискретизация
  + Нужни са допълнителни електронни схеми за постигане на тази цел – аналогово-цифрво преобразувател (ЦАП)
* Постоннотокови електродвигатели
  + Основни параметри
    - Електрическа мощност: W, kW
    - Захранващо напрежение: V
    - Номинален ток: A
    - Номинални обороти: rpm
    - Въртящ момент: Nm
    - Размери на електродвигателя
    - Размер на изходящия вал
  + Поради по-голямата консумаця, електромоторите не могат да бъдат свързани директно към изходите на микроконтролерите. Един от начините да бъде изпълнено е чрез транзистори
* **Сервоелектромотор**
  + Има вградена система за обратна връзка и прави изключително лесно решаването на посочените задачи
  + Управляващия сигнал носи информация за ъгъла на който трябва да се завърти вала
  + Информацията е кодирана в продължителността на импулса
  + За управлението на тези севоелектромотори се използва вградената в Arduino IDE *“***Servo.h***”*
  + За всеки сервомотор се създава обект от класа Servo
  + **attach(pinNumber)** - конфигурира управляващ пин
  + **write(angleDegree)** - задава ъгълът, на който трябва да се завърти вала