



МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКИТЕ
ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО ТЕЛЕКОМУНИКАЦИИ
ГР. СОФИЯ

Изпитна тема №16:
Програмиране за вградени системи

Точка 1: Дефинира понятията: електричество, променлив ток (AC), прав ток (DC), широчинно-импулсна модулация и др. (4т.)

- **Електричество** - Явленията, които са резултат от движението или натрупването на електрически заряди.
- **Променлив ток (AC)** - Електрически ток, който изменя периодично във времето своята големина и посока.
- **Прав ток (DC)** – Не се изменя по посока и големина във времето.
- **Широчинно-импулсна модулация (ШИМ)** - Широко-импулсна модулация (PWM) е метод за преобразуване на сигнал, който променя продължителността на импулса (пространствена) и честотата остава постоянна. Колкото по-голям е коефициентът на запълване, толкова по-ярко ще изглежда, че свети обекта. (Например светодиоди).

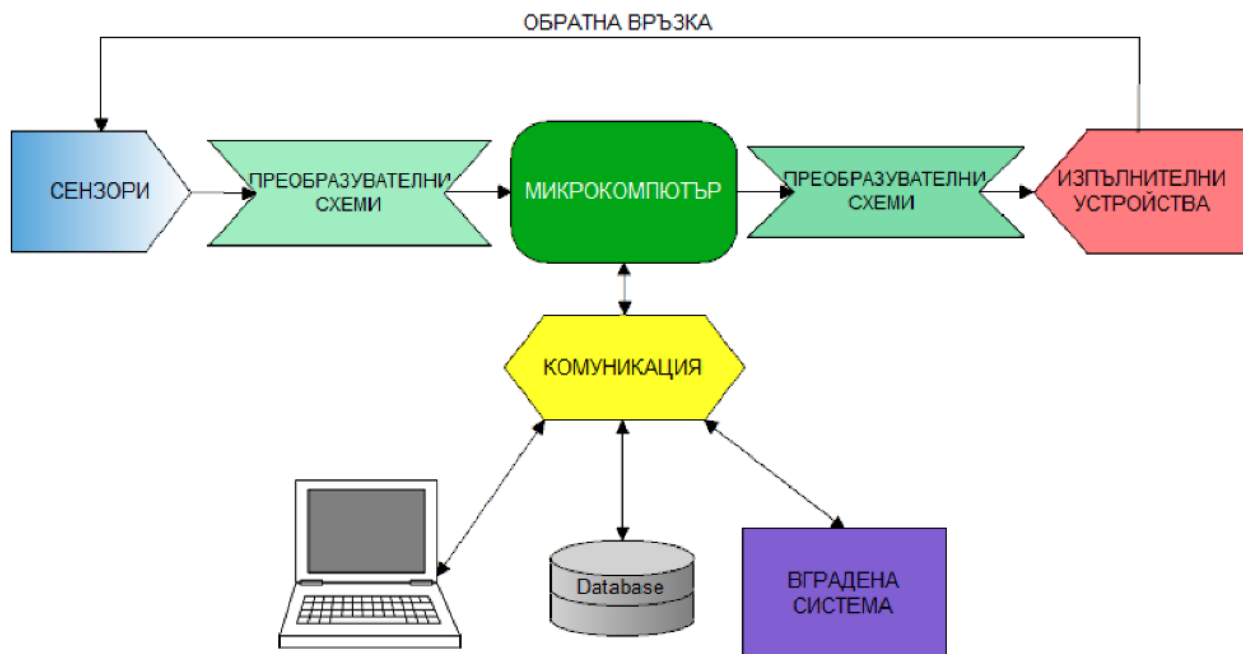
Точка 2: Различава видовете развойни платки и микроконтролери. (4т.)

- Развойните платки имат вече готови прикачени към тях компоненти, подготвени за стартиране на проект по платката, например портове за свързване, вече готов микроконтролер и т.н., а микроконтролерите нямат готови допълнителни елементи.
- **Типове развойни платки:**
 - Платки, базирани на микроконтролери;
 - System on a chip (SoC) boards;
 - Едноплаткови компютри (Single-board Computers (SBC)).
- **Типове микроконтролери:**
 - 8, 16, 32 битови микроконтролери.

Точка 3: Обяснява основните характеристики и особености на вградените системи. Дава пример за блокова схема на вградена система. (8т.)

- **Характеристики:**
 - Усъвършенствана функционалност;
 - Операция в реално време;
 - Ниски производителни разходи;
 - Процесор и памет

- **Блокова схема:**



Точка 4: Обяснява и дава пример за принципа на работа на вградена система. Избира електронни компоненти за вградена система по поставена задача. (16т.)

- Принципът на работа на една вградена система е следния: системата получава информация от сензори; чрез тях се следят състоянията на наблюдаваните величини; на основата на това и в съответствие със заложеното в алгоритъма се управляват изпълнителни устройства; с тяхна помощ се контролира механичната или електрическа система.
- Информацията бива съхранявана в компютъра в двоична форма. Накратко „нули“ (0) и „единици“ (1). В програмирането, единицата служи за „истина“ (true), а нулата за „лъжа“. Във вградените системи, най-често се използват команди за задаване на напрежение на даден пин, като липсата на напрежение се записва в компютърната памет като нула (0), а наличието на някакво напрежение, като единица (1).
- Етапи за разработка на програма за вградена система
 - **Инициализиране** – Осигурява се доставяне на данните
 - **Вход (Input)** – Това са последователност от изпълними редове, които са необходими за да се получи необходимата информация, за да се реши задачата. Тази информация се получава от потребителя.
 - **Обработка (Process)** – Изпълняват се необходимите действия, в резултат на които, от множество входни данни, се получава ново множество изходни данни.
 - **Изход (Output)** – Тази стъпка е отговорна за използването на резултатите от обработката на програмата. Това може да означава просто визуализиране на новите данни върху дисплей или изпращане на нова стойност към друга програма.

- **Край (End)** – След като програмата приключи изпълнението си, системата ще се върне в първоначалното си състояние.

Точка 4 – Задача: Избира електронни компоненти за вградена система по поставена задача.

Условие: Да се изберат електронни компоненти за създаване на вградена система, която при натиснат бутон, ще активира два червени светодиода, които мигат един след друг и ще изключи един зелен светодиод, а когато бутонът не е натиснат, зеленият светодиод ще свети, а червените ще са изгасени.

Отговор:

- Arduino платка;
- USB кабел за комуникация;
- Един брой 10 KiloOhm резистор;
- Три броя 220 Ohm (Ω) резистора;
- Два червени светодиода;
- Един зелен светодиод;
- Бутон (switch);
- Кабел за “+” и “-”;
- Кабели за свързване по платката

Точка 5: Разработва и проектира вградена система, така че да реши поставена задача. (6т.)

Условие: Проектирайте вградена система за създаване на „Командно табло на космически кораб“ и направете схема за работа на вградената система. При ненатиснат бутон, да свети зелен светодиод и два червени да са изгасени, а при натиснат бутон, двата червени светодиода да мигат един след друг през интервал от една секунда и зеленият светодиод да е изгасен.

Отговор: За създаването на „Командно табло на космически кораб“ ще са ни нужни следните компоненти:

- Arduino платка;
- USB кабел за комуникация;
- Един брой 10 KiloOhm резистор;
- Три броя 220 Ohm (Ω) резистора;
- Два червени светодиода;
- Един зелен светодиод;
- Бутон (switch);
- Кабел за “+” и “-”;
- Кабели за свързване по платката

След като компонентите са свързани към платката и тя е свързана към компютър с инсталирано Arduino IDE, ще трябва да създадем нужните променливи във “void setup” метода, като също

инициализираме кой пин какво ще бъде. Дали ще е output или input за даден светодиод. В нашия случай, ще ни трябват три пина, които да са output за светодиодите и един input за бутона. След инициализирането на пиновете, във “void loop” метода ще напишем код с една if/else проверка, която ще накара вградената система да изпълнява задачата от условието.

Точка 6: Описва специфичните изисквания и особености на софтуера, предназначен за вградени системи. (2т.)

- Особеностите на програмата Arduino, спрямо другите среди за разработка са:
 - езикът на писане – C;
 - начинът на структуриране на програмата – void setup метод и void loop метод;
 - изпълняването на кода – Verify и Upload

Точка 7: Обяснява структурата на програма за вградена система. (4т.)

В Ардуино има:

- Два главни метода (функции):
 - **void setup()** – Изпълнява се веднъж. Използва се за инициализацията на **контролера**;
 - **void loop()** – Функция, която се изпълнява многократно. Има същинска функционалност.
- Три типа променливи:
 - **char** – 8bit
 - **int** – 16bit
 - **long** – 32bit
- Константи:
 - **HIGH** – Задава напрежение 5V на пина.
 - **LOW** – Задава напрежение 0V на пина.
 - **INPUT** – Дефиниране на пина, като вход.
 - **OUTPUT** – Дефиниране на пина, като изход.

Точка 8: Демонстрира код за управление на състоянието на изводите. (6т.)

```
void setup() {  
    // Задаваме на програмата, че пин №10 ще служи за изход (output)  
    pinMode(10, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(10, HIGH); // Ще включим светодиода  
    delay(1000);           // Задава изчакване от 1 секунда  
    digitalWrite(10, LOW);  // Ще изключи светодиода  
    delay(1000);           // Задава изчакване от 1 секунда  
}
```

Точка 9: Демонстрира код за управление на електронни компоненти с развойна платка по зададена задача. (6т.)

Условие: Направете програма на „Командно табло на космически кораб“, която при натискане и задържане на бутона (switch-а) на ардуино платката, двата червени светодиода трябва да започнат да светят, като се редуват, а зеленият светодиод да изгасне. При пуснат бутон (switch), да свети само зеленият светодиод.

Отговор:

```
int switchState=0;

void setup() {
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  switchState=digitalRead(2);
  if (switchState==LOW) { //the button is not pressed

    digitalWrite(3, HIGH); //green led
    digitalWrite(4, LOW); //red led
    digitalWrite(5, LOW); //red led
  }
  else { //the button is pressed
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);

    delay(250); //wait for a quarter second
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    delay(250); //wait for a quarter second
  }
} //go back to the beginning of the loop
```

Точка 10: Прави заключения и изводи за серийната комуникация. (8т.)

- С ардуино можем да извършваме тази комуникация с компютъра чрез USB plug и TX/RX пиновете на ардуино. Серийната комуникация в Arduino се осъществява чрез пиновете, които са предназначени за тази цел. Всяка програма, която може да чете данни от сериен порт може и да си „говори“ (задава различни команди) със самото Arduino. Накратко, без серийната комуникация, Arduino платката няма как да контактува със софтуера на компютъра и така да получава и изпраща данни, които да помогнат за изпълняването на програмата.

Точка 11: Открива грешки в програмен код и го модифицира, така че да реши поставената задача. (12т.)

Условие: Променете програмата за контролното табло на космическия кораб от Точка №9, така че при натискане на бутона, зеленият светодиод да изгасне, единият червен светодиод да мигне 5 пъти, след което другият червен да мигне 3 пъти, като това действие да се повтаря, докато бутонът е натиснат.

Отговор:

```
int switchState=0;

void setup() {
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  switchState=digitalRead(2);

  int i;

  if (switchState==LOW) { //the button is not pressed
    digitalWrite(3, HIGH); //green led
    digitalWrite(4, LOW); //red led
    digitalWrite(5, LOW); //red led
  }
  else { //the buton is pressed
    digitalWrite(3, LOW);

    for(i = 0; i < 5; i++) {
      digitalWrite(4, LOW);
      digitalWrite(5, HIGH);
      delay(250);
      digitalWrite(5, LOW);
      delay(250);
    }

    for(i = 0; i < 3; i++) {
      digitalWrite(4, HIGH);
      digitalWrite(5, LOW);
      delay(250);
      digitalWrite(4, LOW);
      delay(250);
    }
  }
}
} //go back to the beginning of the loop
```

Точка 12: Анализира, определя и допълва програмен код, така че да реши поставената задача. (24т.)

Условие: Използвайки задачата от Точка №11, анализирайте кода и обяснете всеки един ред какво прави, като след това допълнете програмата, като направите зеленият светодиод да мига 5 пъти, заедно с първия червен светодиод и след това да мига 6 пъти самостоятелно, преди мигането на втория червен светодиод. Накрая всички светодиоди да светнат за 5 секунди.

Отговор:

```
int switchState=0;

void setup() {
  pinMode(3, OUTPUT); //Инициализиране на пин №3 като Output
  pinMode(4, OUTPUT); //Инициализиране на пин №4 като Output
  pinMode(5, OUTPUT); //Инициализиране на пин №5 като Output
  pinMode(2, INPUT); //Инициализиране на пин №2 като Input
}

void loop() {
  switchState=digitalRead(2); //Задаване на стойност на SwitchState променливата

  int i; //Деклариране на променлива

  if (switchState==LOW) { //Бутонът не е натиснат
    digitalWrite(3, HIGH); //Зеленият светодиод свети
    digitalWrite(4, LOW); //Червеният светодиод не свети
    digitalWrite(5, LOW); //Червеният светодиод не свети
  }
  else { //Бутонът е натиснат

    for(i = 0; i < 5; i++) { //For цикъл за повтаряне на действието
      digitalWrite(4, LOW); //Червеният светодиод не свети
      digitalWrite(5, HIGH); //Червеният светодиод свети
      digitalWrite(3, HIGH); //Зеленият светодиод свети
      delay(250); //Изчакване от 250мс
      digitalWrite(5, LOW); //Червеният светодиод не свети
      digitalWrite(3, LOW); //Зеленият светодиод не свети
      delay(250); //Изчакване от 250мс
    }

    for(i = 0; i < 6; i++) { //For цикъл за повтаряне на действието
      digitalWrite(4, LOW); //Червеният светодиод не свети
      digitalWrite(5, LOW); //Червеният светодиод не свети
      digitalWrite(3, HIGH); //Зеленият светодиод свети
      delay(250); //Изчакване от 250мс
      digitalWrite(3, LOW); //Зеленият светодиод не свети
      delay(250); //Изчакване от 250мс
    }

    for(i = 0; i < 3; i++) { //For цикъл за повтаряне на действието
      digitalWrite(4, HIGH); //Червеният светодиод свети
      digitalWrite(5, LOW); //Червеният светодиод не свети
      delay(250); //Изчакване от 250мс
      digitalWrite(4, LOW); //Червеният светодиод не свети
      delay(250); //Изчакване от 250мс
    }

    digitalWrite(3, HIGH); //Зеленият светодиод свети
    digitalWrite(4, HIGH); //Червеният светодиод свети
    digitalWrite(5, HIGH); //Червеният светодиод свети
    delay(5000); //Изчакване от 5 секунди
    digitalWrite(3, LOW); //Зеленият светодиод не свети
    digitalWrite(4, LOW); //Червеният светодиод не свети
    digitalWrite(5, LOW); //Червеният светодиод не свети

  }
} //go back to the beginning of the loop
```