СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”

Факултет по математика и информатика
Специалност: Информатика
Магистърска Програма: Вградени Системи
Асистент: Тома Томов

ПРОЕКТИРАНЕ НА РОБОТИЗИРАНИ СИСТЕМИ

Дисциплина:
Проектиране на роботизирани системи

E-mail:
ttomov75@gmail.com

Тема на упражнението

1. **Предговор**
2. **Функции**
3. **Къдрави скоби (curly braces)**
4. **Точка и запетая**
5. **Коментари**
6. **Променливи**

Ардуино

- Програмиране

Тук setup() е подготовката (preparation), а loop() е изпълнението (execution). И двете функции са задължителни за да работи програмата.

Функцията setup() трябва да е след декларирането на променливи, което се прави в самото начало на програмата. Тя е първата функция, която се изпълнява, протича само веднъж и служи за да зададе pinMode (режим на пиновете) или да отвори серийната комуникация.

loop() функцията се изпълнява втора и съдържа код, който се изпълнява за продължителен период от време – отчита показания от пиновете, пуска сигнали до пиновете и т.н. Тази функция е ядрото на всяка Ардуино програма и изпълнява повечето задачи.

Ардуино

setup()
Функцията setup() се повиква веднага щом програмата тръгне. Използва се при задаване ролята на пиновете или за да започне серийна комуникация. Тя трябва да бъде включена в програмата дори ако не съдържа никакви изявления.

```
void setup()
{
pinMode(pin, OUTPUT);              // zadava OUTPUT rejim na pina
}
```

loop()
След като функцията setup() е повикана, loop() функцията прави точно това, което името и подсказва че ще прави – цикли постоянно и позволява на програмата да се променя, да реагира и да контролира Ардуино платката.

```
void loop()
{
digitalWrite(pin, HIGH);           // vklyuchva pina
delay(1000);                       // izchakva edna sekunda
digitalWrite(pin, LOW);            // izklyuchva pina
delay(1000);                       // izchakva edna sekunda
}
```

Ардуино

Функции (functions)

Функцията е откъс код с име и блок от изявления (statements), които се изпълняват когато функцията бъде повикана. Функциите void loop() и void setup() вече бяха обяснени, а други функции ще разгледаме по-късно.

Индивидуализирани (custom) функции могат да се използват за повтарящи се задачи и за да бъде кодът по-подреден и разбираем. Функциите се декларират като първо се декларира вида на функцията. Това е вида стойност, например int за цяло число (integer), който функцията връща като резултат. Ако функцията не връща никакъв резултат тя е невалидна. След вида се декларира името, което даваме на функцията и в скоби се поставят различни параметри, които се подават на функцията.

Ардуино

Функции

Следната функция от вид „integer” izchakvaneStojnost() се използва за да се зададе изчакване в програма на базата на отчетена стойност от потенциометър. Първоначално тя декларира локална променлива v, задава й стойност от потенциометъра (число от 0 до 1023), после разделя числото на 4 за да се получи крайна стойност между 0 и 255, и накрая връща

Ардуино

{} къдрави скоби (curly braces)
Къдравите скоби обозначават началото и края на блокове от функции или изявления, като например void loop() функцията и for и if изявленията (for and if statements).
вид функция()
{
изявления;
}
Всяка отварящата къдрава скоба { трябва да е последвана от затваряща }. Това се нарича баланс на скобите. Небалансираните скоби могат да доведат до криптически, и неразгадаеми грешки при компилирането и понякога могат да бъдат трудни за откриване в по-голяма програма.

Средата за програмиране на Ардуино предлага удобна възможност за проверка баланса на къдравите скоби. Просто маркирайте някоя къдрава скоба, или празното разстояние веднага след нея, и логическият и спътник ще бъде посочен.

Ардуино

; точка и запетая (semicolon)

В края на всяко изявление, както и в края на всеки отделен елемент на програмата, трябва да се поставя точка и запетая. Точка и запетая се използват и за отделяне на елементите във for циклите (for loops).

```
int x = 13;          // deklarira promenlivata x kato integer-a 13
```

Забележка: Забравянето на точка и запетая в края на реда ще даде грешка при компилирането. Дефиницията на грешката може да е ясна, и да казва за липсващи точка и запетая, или пък да не е. Ако получите неразгадаема или привидно нелогична грешка при компилирането, едно от първите неща за които трябва да проверите е за липсваща точка и запетая близо до реда от който компилаторът се е оплакал.

Ардуино

/* ... */ блок коментар (block comments)

Блок коментарите, или коментарите от по няколко реда, са парчета текст които програмата не взима предвид и се използват за по-дълги текстови описания. Те помагат на другите да разберат по-лесно какви роли изпълняват частите на програмата. Блок коментарите започват с /* и завършват с */ и могат да се простират на няколко реда.

```
/* това е блок коментар
не забравяйте да затворите коментара
той също трябва да е балансиран
*/
```

Тъй като коментарите се игнорират от програмата и не заемат никакво място в паметта те не трябва да се пестят. Коментарите могат да се използват и за изключването на парчета код при отстраняването на бъгове (debugging).

Забележка: Въпреки че е възможно да обозначавате коментари от по един ред като блок коментари, не е позволено коментарът да съдържа втори коментар в себе си.

Ардуино

// коментар на реда (line comment)

Коментари от по един ред започват с // и завършват със следващия ред код. Също като блок коментарите те се игнорират от програмата и не заемат място в паметта.

```
// това е коментар от един ред
```

Коментарите на реда често се използват след изявления за да дадат информация за това какво постига изявлението или да послужат като подсказка при преработване на кода.

Ардуино

променливи (variables)

Променливата е начин да се именува и да се запази цифрова стойност която да се използва по-късно в програмата. Както предполага името им, променливите могат постоянно да се променят за разлика от константите, чиито стойности никога не се променят. Променливата трябва задължително да се декларира, а задаването на стойността която да се запази е по избор. Кодът долу декларира променлива наречена inputPromenliva и после й задава стойността получена от аналоговия входен пин 2:

```
int inputPromenliva = 0; // deklarira promenliva i

// j zadava stojnost 0

inputPromenliva = analogRead(2); // zadava na promenlivata stojnostta ot

// analogoviya pin 2
```

‘inputPromenliva’ е самата променлива. Първият ред декларира, че тя ще приема стойности от вида integer (int). Вторият ред задава на променливата стойността от аналоговия пин 2. По този начин стойността от пин 2 е достъпна и в други части на кода.

Ардуино

променливи (variables)

След като на променливата е зададена или повторно зададена стойност, може да се провери дали тази стойност отговаря на дадено условие или да се използва направо. Като пример ще покажем три полезни операции с променливи. Следният код проверява дали inputPromenliva е по-малка от 100, и ако условието е вярно задава на променливата стойност 100 и след това задава изчакване (delay) равно на стойността на променливата, която вече е равна поне на 100:

```
if (inputPromenliva < 100) // proveryava dali stojnostta e po-malka ot 100
{
inputPromenliva = 100; // ako gornoto uslovie e vyarno zadava stojnost 100
}
Delay(inputPromenliva); // izpolzva promenlivata za izchakvane (delay)
```

Ардуино

деклариране на променливите (variable declaration)

Всяка променлива трябва да бъде декларирана преди да може да се използва.
Декларирането на променливата означава да се дефинира нейния вид (например integer, long, float, и др.), да й се даде име, и по избор може да и се зададе първоначална стойност. Достатъчно е променливата да се декларира веднъж в програмата, а нейната стойност може да бъде променена по всяко време чрез аритметични функции или други методи на задаване.
Този пример декларира променливата inputPromenliva като integer (int) и й задава първоначална стойност 0.

```
int inputPromenliva = 0;
```

Променливата може да бъде декларирана на различни места в програмата и в зависимост от това къде е дефинирана се определя кои части на програмата могат да я ползват.