

## ЛАБОРАТОРНО УПРАЖНЕНИЕ №6

### Четене на цифрови входове, управление изпълнението на програмата

#### Цел на упражнението:

Студентите да придобият практически умения за работа с програмируеми електронни устройства и да получат основни познания за четене на цифрови входове и управление на цифрови изходи по зададен алгоритъм.

#### I. ЗАДАНИЕ

1. Да се разучат основни функции и конструкции на езика за програмиране на *Arduino*.

2. Да се въведе и зареди програмата за включване и изключване на светодиода, свързан към цифров извод 13 при натискане на бутона, включен към цифров извод 2, чрез използване на конструкцията **if... else**. Програмата е дадена в т.Ш.

3. Да се състави програма за алтернативна смяна на състоянието на светодиодите, включени към цифрови изводи 9 и 13 при натискане на бутона, включен към цифров извод 2, като се допълни предната програма.

4. Да се въведе и зареди програмата за управление на светодиодите, включени към цифрови изводи 9 и 13 при натискане на бутона, включен към цифров извод 2, чрез използване на конструкцията **switch**. Програмата е дадена в т.Ш.

5. С използването на монтажна основа да се реализира макет на светофар с ръчно управление. За целта да се състави програма за алтернативна смяна на състоянието на червен, жълт и зелен светодиод, свързани към цифрови изводи 3, 4 и 5 при натискане на бутона, включен към цифров извод 2, като се допълни предната програма.

#### II. ТЕОРЕТИЧНИ ПОЯСНЕНИЯ

В по-сложните програми не се изпълнява само линеен алгоритъм, който се състои от последователни действия, а изпълнението на програмата се управлява в зависимост от различни условия. Тези условия могат да бъдат стойностите на цифрови или аналогови входове. За целта трябва да се прочете състоянието на цифров вход.

#### Функции и конструкции за четене на цифрови входове и управление изпълнението на програмата

##### **pinMode(pin, rejim)**

Използва се във **void setup()** за инициализиране на определен извод (pin) като вход (INPUT) или изход (OUTPUT).

```
pinMode(pin, INPUT); // инициализира се извод 'pin' като вход 'INPUT'
```

По подразбиране изводите на *Arduino* са входове (INPUT) и не е необходимо да се инициализират като входове чрез **pinMode()**.

##### **digitalRead(pin)**

Отчита показанието от определен цифров вход и връща като резултат HIGH или LOW. Входът може да се зададе като променлива или константа.

```
stojnost = digitalRead(pin); // записва в 'stojnost' състоянието на 'pin'
```

Проверката на стойността и управлението изпълнението на програмата се извършва с използване на конструкциите **if** и **switch**.

### **if**

**if** конструкциите проверяват дали дадено условие е изпълнено, като например дали аналоговите стойности са над определена граница, и задават изпълнението на командите в къдравите скоби, ако условието е изпълнено. Ако условието не е изпълнено програмата прескача тези команди. Форматът на **if** конструкциите е следният:

```
if (nyakakvaPromenliva ?? stojnost)
{
    izpylniKomanda;
}
```

### **if... else**

**if... else** позволява да се взимат “или-или” решения. Например, ако трябва да се провери състоянието на цифров вход и да се зададе една команда, ако показанието е **HIGH** или да се зададе друга команда, ако показанието е **LOW**, това може да се направи по следния начин:

```
if (inputPin == HIGH)
{
    napraviA;
}
else
{
    napraviB;
}
```

### **switch**

**switch** е конструкция, която е много подходяща за използване при избор от множество възможности. Дадена стойност се сравнява последователно със списък целочислени или константи със знак. Когато се намери съвпадение, се изпълнява поредицата от конструкции, асоциирана с него. Следващият пример показва формата на конструкцията **switch**:

```
switch (stojnost) {
    case konstanta1:
        poredica ot konstrukcii
        break;
    case konstanta2:
        poredica ot konstrukcii
        break;
    .
    .
    .
    case konstanta3:
        poredica ot konstrukcii
        break;
```

При намиране на съвпадение, конструкциите, асоциирани с него се изпълняват до срещане на **break**.

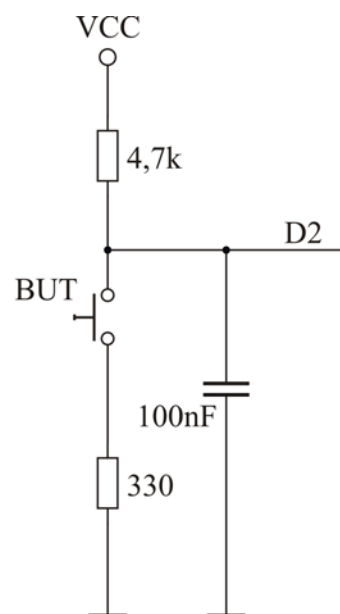
### III. УКАЗАНИЯ ЗА ИЗПЪЛНЕНИЕ НА ЗАДАНИЕТО

На фиг. 1 е показана схемата на вградения върху развойната платка бутон, който подава цифров сигнал към извод 2 на микроконтролера. Когато бутонът е натиснат към цифровия вход ще бъде подадена логическа 0 (стойност *LOW*), а когато бутонът не е натиснат (отпуснат) – логическа 1 (стойност *HIGH*).

Допълнително светодиоди и бутони могат да се свързват към изводите на развойната платка чрез използване на външна монтажна основа, като се запази същата конфигурация на схемите и стойностите на елементите.

Текстът на основната програма за управление състоянието на цифров изход в зависимост от стойността на цифров вход е даден по-долу. Светодиодът е свързан към цифров изход 13, а бутонът е включен към цифров изход 2.

Променливата **buttonState** съдържа състоянието на бутона (*HIGH* или *LOW*). Нейната стойност се записва след изпълнение на командата за четене на цифровия вход. Цифровият изход се управлява след проверка с използване на конструкцията **if... else**.



Фиг. 1. Схема на свързване на бутон към цифров вход

```
const int buttonPin = 2; // наименование на цифров изход 2
const int ledPin = 13; // наименование на цифров изход 13
int buttonState = 0; // променлива за състоянието на бутона
```

```
void setup() {
  // инициализиране на изход ledPin като изход:
  pinMode(ledPin, OUTPUT);
  // инициализиране на изход buttonPin като вход:
  pinMode(buttonPin, INPUT);
}
```

```
void loop(){
  // записване на стойността на цифровия вход:
  buttonState = digitalRead(buttonPin);

  // проверка дали бутонът е натиснат.
  // ако е, стойността на buttonState е LOW:
  if (buttonState == LOW) {
    // включване на светодиода:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // изключване на светодиода:
    digitalWrite(ledPin, LOW);
  }
}
```

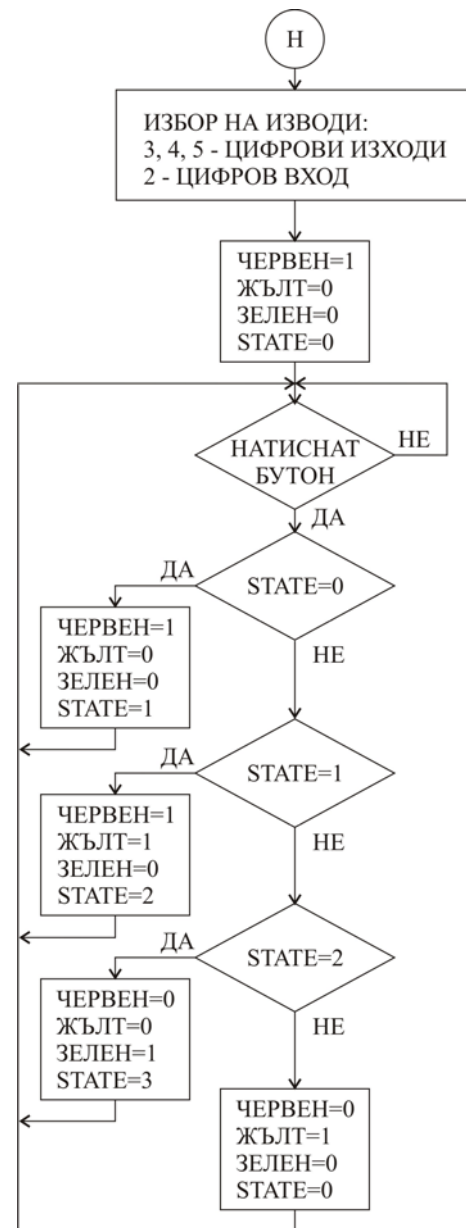
Използването на конструкцията **switch** е илюстрирано със следващата програма. Тя управлява включването и изключването на двата светодиода по определен начин след натискането на бутона. Променливата `state` показва състоянието на светодиодите и може да има стойности 0, 1, 2 и 3. При стойност 0 и двата светодиода са изключени, при стойност 1 е включен само зеленият светодиод, при стойност 2 – само жълтият, а при стойност 3 – и двата. След натискане на бутона променливата `state` приема следващата стойност.

```
int button = 0; // променлива за съхранение състоянието на бутона
int state = 0; // променлива за състоянието на светодиодите

void setup() {
    pinMode(9, OUTPUT); // инициализиране на извода като изход
    pinMode(13, OUTPUT); // инициализиране на извода като изход
    pinMode(2, INPUT); // инициализиране на извода като вход
}
void loop()
{
    button = digitalRead(2); // четене състоянието на бутона
    if (button == LOW) { // проверка дали бутонът е натиснат
        switch (state) {
            case 0:
                digitalWrite (13, LOW);
                digitalWrite (9, LOW);
                state = 1;
                break;
            case 1:
                digitalWrite (13, HIGH);
                digitalWrite (9, LOW);
                state = 2;
                break;
            case 2:
                digitalWrite (13, LOW);
                digitalWrite (9, HIGH);
                state = 3;
                break;
            case 3:
                digitalWrite (13, HIGH);
                digitalWrite (9, HIGH);
                state = 0;
                break;
        }
        delay(300); // закъснение 300 ms
    }
}
```

Програмата за ръчно управление на светофар може да изпълнява алгоритъма, показан на фиг. 2.

Първоначално изводи 3, 4 и 5 се инициализират като цифрови изходи, а извод 2 – като цифров вход. Променливата STATE показва текущото състояние на светофара. Тя може да бъде равна на 0, 1, 2 или 3. При състояние „червено - свети, жълто - не свети, зелено - не свети”, променливата е равна на 0. При състояние „червено - свети, жълто - свети, зелено - не свети”, променливата е равна на 1. При състояние „червено – не свети, жълто - не свети, зелено - свети”, променливата е равна на 2. При състояние „червено – не свети, жълто - свети, зелено - не свети”, променливата е равна на 3. Началната стойност на променливата STATE се задава 0. При натискане на бутона светофарът преминава в ново състояние. След всяко натискане на бутона състоянието на променливата STATE се променя според приетите по-горе съответствия. Зададеното състояние на светофара се задържа до ново натискане на бутона.



Фиг. 2. Блокова схема на програма за ръчно управление на светофар

Макетът на светофар, реализиран върху монтажна основа е показан на фиг. 1.5. в Лабораторно упражнение № 5.