ЛАБОРАТОРНО УПРАЖНЕНИЕ №1

Инсталиране на развойна система тип *Arduino*, управление на цифрови изходи

Цел на упражнението:

Студентите да се запознаят с възможностите на развойната система *Arduino*, да придобият практически умения за работа с програмируеми електронни устройства и да получат основни познания за светодиоди.

I. ЗАДАНИЕ

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

2. Да се разучи развойната среда за програмиране на езика на *Arduino*. За целта да се зареди програмата за включване и изключване на светодиода, свързан към цифров извод 13. Да се обясни нейното действие.

3. Да се допълни същата програма с добавяне на редове за алтернативно управление на светодиодите, свързани към цифрови изводи 9 и 13.

4. Да се промени времевия интервал за включване и изключване на светодиодите. Да се намери неговата стойност за случая, когато не се забелязва мигането на светодиодите.

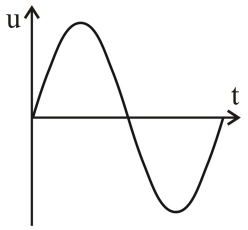
5. С използването на монтажна основа да се реализира макет на светофар с използване на червен, жълт и зелен светодиод, свързани към цифрови изводи 3, 4 и 5. Да се състави програма за автоматично управление на светофара, като се задават различни продължителности на светене на светлините.

II. ТЕОРЕТИЧНИ ПОЯСНЕНИЯ

Програмируемите електронни устройства изпълняват различни операции под управлението на управляваща програма. Това ги прави много гъвкави, тъй като промяната на действието на такова устройство става след зареждане на новата програма. Те извършват логически и аритметични операции и са предназначени за обработка на информация, приемане и предаване на данни, генериране на сигнали и управление. Съвременните програмируеми електронни устройства са изградени с микроконтролери. Микроконтролерът е интегрална схема с висока степен на интеграция, включваща в себе си аритметично-логическо устройство, памети RAM и ROM, периферни схеми за въвеждане и извеждане на цифрови и аналогови сигнали, интерфейсни схеми за комуникация. Главното им предназначение е за управление на обекти и технологични операции.

Микроконтролерът е цифрово устройство и работи с двоични числа. Тъй като в двоичната бройна система има само две числени стойности – 0 и 1, на тези стойности могат лесно да се съпоставят стойности на напрежението. В съвременните цифрови схеми на числото 1 отговаря стойността на захранващото напрежение (най-често +5V), а на числото 0 - 0V. Следователно цифровият сигнал има само две стойности (фиг. 1.1а). Един двоичен разряд представлява количеството информация **бит (bit)**.

Фиг. 1.1а. Цифров сигнал



Фиг. 1.1б. Аналогов сигнал

За разлика от него аналоговият сигнал може да заема безкраен брой стойности (фиг. 1б). В природата повечето процеси имат аналогов характер, следователно сигналите, които ги характеризират са аналогови. За съгласуване на аналоговия свят с цифровите управляващи устройства се използват аналогово-цифрови и цифрово-аналогови преобразуватели.

За първоначално навлизане в света на микроконтролерите много подходящ инструмент е платформата *Arduino*, която се използва масово поради своята достъпност и удобство при настройката на програмите. Използваната в учебния процес платформа *Olimexino 328* (фиг. 1.5) е аналог на *Arduino*, но има и някои предимства пред оригинала. Отделните изводи (pins) на устройството могат да имат различно предназначение. Изводите, означени като Digital 0 до Digital 13 могат да бъдат използвани като цифрови входове и изходи. Тяхното предназначение се задава в програмата.

Основна структура на езика за програмиране на *Arduino*

Основната структура на езика за програмиране на *Arduino* е относително проста и се състои от поне две части. Тези две задължителни части (или функции) съдържат блокове конструкции (statements). Конструкцията задава действие, което трябва да се извърши от програмата. Всички конструкции завършват с точка и запетая.

```
void setup ()
{
      statements;
}
void loop ()
{
      statements;
}
```

В горния пример **setup()** е подготвителната функция (preparation), а **loop()** - изпълнителната (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 задаване режима на изводите (pinMode) или за инициализиране на серийната комуникация.

Функцията **loop()** се изпълнява втора и съдържа код, който се изпълнява продължителен период от време – отчита стойности от изводите, генерира

сигнали на изводите и т.н. Тази функция е ядрото на всяка *Arduino* програма и изпълнява повечето задачи.

{} големи (къдрави) скоби (curly braces)

Големите скоби обозначават началото и края на блокове от функции или конструкции, като например **void loop()** функцията и **for** и **if** конструкциите.

вид функция()
{
конструкции;
}

Всяка отварящата голяма скоба { трябва да е последвана от затваряща }. Това се нарича баланс на скобите. Небалансираните скоби могат да доведат до синтактични и логически грешки при компилирането и понякога могат да бъдат трудни за откриване в по-голяма програма.

Средата за програмиране на *Arduino* предлага удобна възможност за проверка баланса на големите скоби. Когато се маркира някоя голяма скоба, или празното разстояние веднага след нея, то логическият й спътник ще бъде посочен.

; точка и запетая (semicolon)

В края на всяка конструкция, 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).

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

Блок коментарите, или коментарите от по няколко реда, са части текст които, програмата не взима предвид и се използват за по-дълги текстови описания. Те помагат за по-лесното разчитане и разбиране какви действия изпълняват частите на програмата. Блок коментарите започват с **/*** и завършват с ***/** и могат да се простират на няколко реда.

// коментар на реда (line comment)

Коментарите 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.

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.

обхват на променливата (variable scope)

Променливата може да бъде декларирана в началото на програмата преди **void setup()**, локално във функцията и понякога в блок конструкции като **for** циклите. Мястото, където е декларирана променливата определя нейния обхват (или възможността на определени части от програмата да я използват).

байт (byte)

Байтът съхранява 8-битово цяло число от 0 до 255.

цяло число (int)

Целите числа (**integer**) са основния вид данни и съхраняват 16-битови числа от 32767 до -32768.

дълга дума (long)

Това е разширен вид данни за по-големи цели числа и съхраняват 32-битови числа от 2147483647 до -2147483648.

плаваща запетая (float)

Вид данни за числа с десетична запетая (дробни числа). Данни от този вид са с по-голям обхват от integer, запазват се като 32-битови числа от 3.4028235E+38 до -3.4028235E+38.

аритметика (arithmetic)

Операторите за аритметика включват събиране, изваждане, умножение и деление. Те намират сбора, разликата, произведението, или частното на две величини.

```
y = y + 3;
x = x – 7;
i = j * 6;
r = r / 5;
```

Аритметични действия могат да се задават и по следния начин:

```
x++; // същото като x = x + 1, или увеличава x с + 1
x– –; // същото като x = x - 1, или намалява x с - 1
x+=y;  // същото като x = x + y, или увеличава x с + y
x–=y;  // същото като x = x - y, или намалява x с - y
x *=y;  // същото като x = x * y, или умножава x с y
x /=y;  // същото като x = x / y, или дели x на y
```

pinMode(pin, rejim)

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

```
pinMode(pin, OUTPUT); // инициализира се извод ‘pin’ като изход
                      // ‘OUTPUT’
```

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

digitalWrite(pin, stojnost)

Задава логическа 1 (HIGH) или логическа 0 (LOW) (включва или изключва), определен извод. Изводът може да се определи като променлива или константа.

```
digitalWrite(pin, HIGH); // задава стойност ‘HIGH’ на извод ‘pin’
```

изчакване(милисекунди) delay(ms)

```
delay(1000); // изчакване една секунда
```

Спира четенето на програмата за зададения в милисекунди период от време, като 1000 се равнява на 1 секунда.

III. УКАЗАНИЯ ЗА ИЗПЪЛНЕНИЕ НА ЗАДАНИЕТО

Среда за програмиране

След стартиране на развойната среда на *Arduino* се появява прозорецът от фиг. 1.2. Програмите на езика за *Arduino* се наричат „скици” (sketch). Зареждането на примерна програма става чрез навигация File menu ➤ Examples ➤ 01. Basics и избор на програма Blink. Това може да бъде изпълнено и с използване на функционалните бутони. Бутоните от управляващата лента на средата изпълняват следните функции:

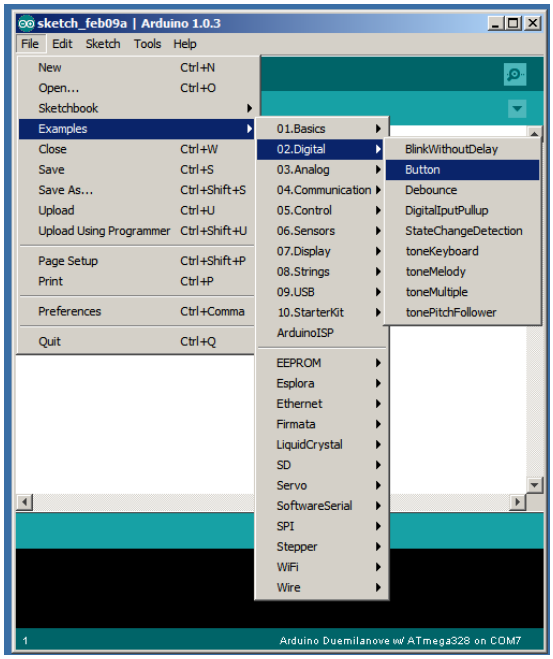
 - verify. Компилира написания код и прави съобщения за грешки.

 - upload. Зарежда компилираната програма в микроконтролера.

 - new. Изчиства прозореца за писане на нова програма.

 - open. Отваря файл с предварително записана програма.

 - save. Запомня написания код във файл.



Фиг. 1.2. Развойна среда на *Arduino*

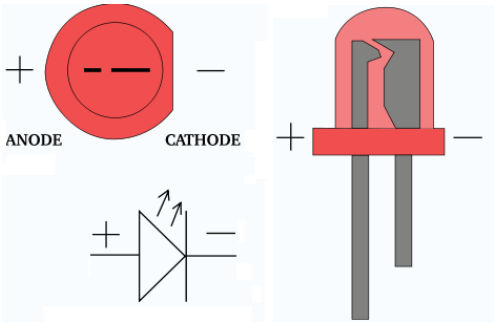
Текстът на основната програма за управление на светодиод, свързан към извод 13 е показан по-долу.

```
int led = 13; // наименование на извод 13
void setup () // изпълнява се веднъж
{
pinMode(led, OUTPUT); // инициализиране на извод led като изход
}
void loop () // изпълнява се непрекъснато
{
digitalWrite (led, HIGH); // включва светодиода
delay (1000); // изчакване 1 секунда
digitalWrite (led, LOW); // изключва светодиода
delay (1000); // изчакване 1 секунда
}
```

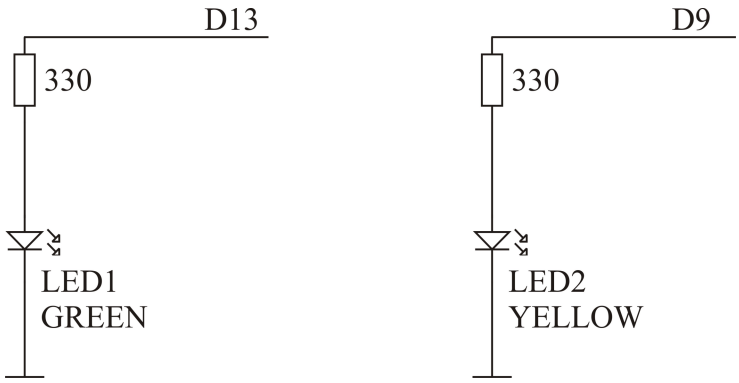
Управление на цифрови изходи

За точкови индикатори, показващи състоянието на цифрови (логически) изходи най-често се използват светодиоди (light emitting diode – LED). За да излъчва светлина, е необходимо на анода (anode) на светодиода да се подаде

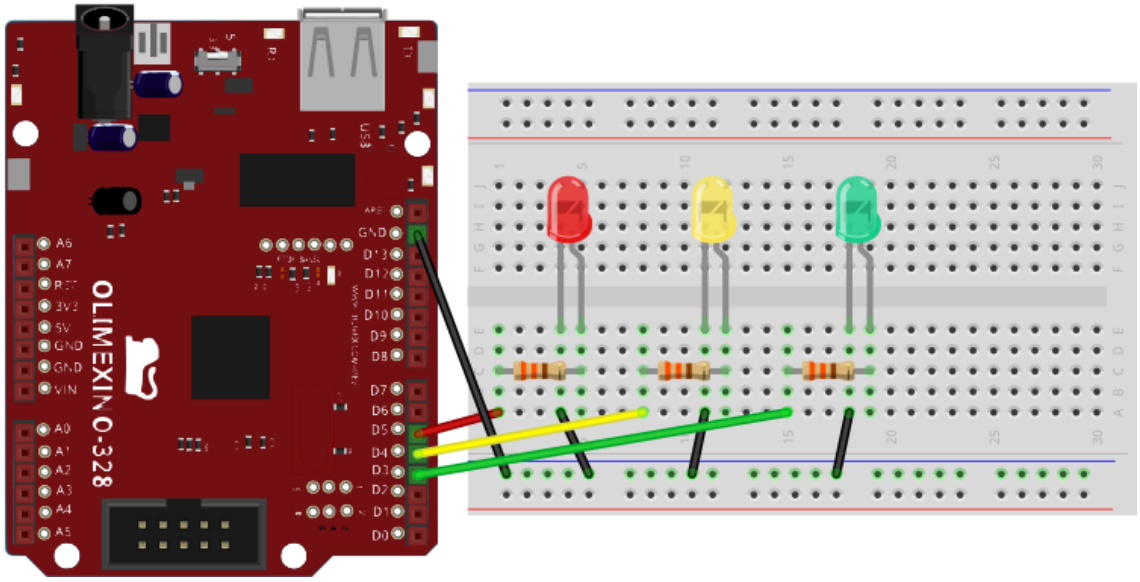
положително спрямо катода (cathode) напрежение (фиг. 1.3). Един начин за управление на светодиоди от цифрови изходи на микроконтролера е показан на фиг. 1.4, който съответства на схемата на развойната платка. При подаване на високо ниво (запис HIGH) на извод 13 ще се включи зеленият светодиод, а при подаване на високо ниво на извод 9 – жълтият. Когато се подаде ниско ниво (запис LOW) изходът е изключен и съответният светодиод няма да свети.



Фиг. 1.3. Символ и изображение на светодиоди (LED)



Фиг. 1.4. Схеми за управление на светодиоди от цифрови изходи



Фиг. 1.5. Изглед на платформа *Olimexino 328* с макет на светофар върху монтажна основа

IV. КОНТРОЛНИ ВЪПРОСИ

1. Кои са задължителните функции на една програма на езика за програмиране на *Arduino*?

2. Какви видове променливи се използват в езика за програмиране на *Arduino*?

3. Коя функция на езика за програмиране на *Arduino* се използва за задаване на режима на избран извод на микроконтролера? Напишете функцията за инициализиране на извод 13 като изход.

4. Кой елемент най-често се използва за индициране състоянието на цифрови изходи?

5. Как трябва да се поляризират изводите на светодиода, за да излъчва светлина?