

Технически спецификации на Ардуино Уно

инж. Борислав Чакъров – Лом – 1.3.2018 г.

В доста коментари във Фейса и в други форуми прозира липса на технически знания и разбиране на техническите параметри на платката на Ардуиното, особено по отношение на възможностите за свързване на периферия: датчици, бутони, външни изпълнителни устройства.

Въпреки, че платката е много грамотно проектирана и е доста „дуракоустойчива“, все пак е добре да спазваме някои основни правила, за да избегнем съзнателно нейното претоварване и вероятността да дефектира по наша вина.

Всички подобни устройства имат два вида технически данни. Обикновено при закупуването обръщаме внимание на чисто „компютърните“ параметри. За Ардуино Уно те са:

- Процесор: Atmega 328
- Тактова честота: 16 MHz
- Flash памет: 32 kB
- RAM: 2 kB
- Интерфейс към PC: USB
- Цифрови входове/изходи: 14
- Аналогови входове/изходи: 6 с 10 битови ЦАП (Цифрово-аналогови преобразуватели) – 1024 нива.
- Аналогови изходи: 6 от 14-те цифрови. ШИМ – широчинно-импулсна модулация – 8 битова - 256 нива
- Език за програмиране: Processing

За повечето колеги тези данни са достатъчни и дават ясна представа за възможностите на контролера. Проблемът обаче може да се яви от незнание на параметри на чисто физическо ниво:

- Захранващо напрежение.
- Максимален ток на допълнително включените периферни устройства – общ.
- Максимално допустим ток на включен товар към всеки отделен изход.
- Максимално допустимо напрежение, което може да се подаде на всеки вход без да го повреди.
- Какво означава 1024 нива резолюция на аналоговите входове?
- Какво означава ШИМ и как се ползва, като аналогов сигнал от изход?
- Стандарт на цифровите входове и изходи – електрически параметри на сигналите.

На база на тези параметри е необходимо да разгледаме и правилните схемни решения – методите и начините за правилно включване на външните устройства към контролера, които да отговарят на три главни условия:

- да са безопасни за контролера;
- да могат да изпълняват предвидената си функция;
- да подават към контролера „разбираеми“ за него сигнали.

Ще разгледаме тези електрически параметри и схемни решения подробно в следващите точки.

1. Захранващо напрежение на Ардуино Уно и другите версии на контролера.

Основната версия на платката има два начина на захранване: чрез USB терминала и чрез захранваща букса. Напрежението на USB е стандартно – 5 V и захранва директно процесора.

Напрежението през буксата обаче минава през допълнителен интегрален стабилизатор за 5 V преди да бъде подадено към процесора. По тази причина напрежението, подавано на буксата трябва да е поне 7 V. Това напрежение, също така, не е желателно да превишава 12 V (дават максимално допустимо 20 V, но не го препоръчвам) за да не повреди 5 волтовия стабилизатор на платката.

На изходите на платката, означени с 5V не бива да се включват допълнителни платки и други консуматори, които черпят ток повече от 0,5 A. В противен случай може да изгори вграденият стабилизатор.

На изхода означен с 3,3 V има стабилизирано напрежение с тази стойност. С него се захранват допълнителни устройства с номинално захранващо напрежение 3,3 V и максимален ток 0,1 A.

Съществуват и други версии на Микроконтролера със захранващо напрежение 3,3 V или 5 V – Arduino mini, nano и др. Непременно трябва да знаете захранващото напрежение на устройството преди да го закупите и преди да го ползвате.

2. Максимален ток на допълнително включените периферни устройства – общ.

USB версия до 2.0 осигурява ток за външното устройство до 0,5 A. USB версия 3.0 и нагоре осигурява 0,9 A и дори по-големи токове.

Микроконтролерът обаче и периферията не бива да черпят повече от 0,5 A за да бъдат в безопасност захранващите вериги. Общият ток по принцип се контролира от автоматичен предпазител, монтиран на платката и ще ви спаси при късо съединение на външна верига. Той обаче не помага при следващата точка.

3. Максимално допустим ток на включен товар към всеки отделен изход.

По спецификация максимално допустимият ток от всеки изход може да е до 40 mA (милиампера). Ако конфигурираме всички достъпни терминали на контролера като изходи и ги натоварим с максималния ток ще получим консумация 0,72 A, което е повече от максимално допустимия ток за цялата платка. Това на практика, обаче, едва ли ще ни се наложи някога, но трябва да се знае.

За онагледяване на консумацията ще кажа, че 40 mA например е токът за захранване на 4 светодиода (по 10 mA). Или реле – НО **ВНИМАВАЙТЕ!** Много малко са релетата, които работят с толкова малък ток. Непременно измерете тока на релето при 5V преди да го включите към контролера. Ако релето е с намотка, освен токът още нещо може да ви създаде проблем – самоиндукцията. Това е напрежение, което може да бъде в пъти по-голямо от захранващото, което сте подали на релето, и с полярност обратна на подадената, което се връща по захранващите проводници от релето към платката и в най-добрия случай може да обърка контролера и той да „зависне“. В по-лошия случай може да повреди изхода на контролера.

Начините за включване на релета и други външни устройства ще разгледаме в последната точка. Тази точка е основата, когато избирате захранващо устройство (например адаптор) за захранване на творението ви. Ако ще го включвате към захранващата бухса трябва да е с напрежение между 7 и 12 V и да може да отдава ток поне 0,5A. Ако ви трябва захранване за „Ардуино про мини, работещо с 5V“ например, ви трябва адаптор за точно 5V и ток 0,5A.

4. Максимално допустимо напрежение, което може да се подаде на всеки вход без да го повреди.

За съжаление не успях да намеря надеждни данни за максимално допустимите напрежения, които може да издържат входовете на контролера, но смятам, че за цифровите входове това не е и необходимо. За аналоговите входове намерих противоречива информация, затова ще ви посъветвам следното: За всички входове се отнася – **НЕ подавайте напрежения към тях, които да превишават захранващото на платката 5 V**, както и отрицателни спрямо маса напрежения. За версии със захранване 3.3 V също е желателно максималното напрежение да не превишава захранващото.

ВАЖНО!

ПРЕДИ да подадете външни напрежения към пиновете на платката СЕ УВЕРЕТЕ, че те вече са конфигурирани софтуерно като входове. Ако подадете външно напрежение към пинове, конфигурирани като изходи, това почти сигурно ще доведе до повреда на този вход/изход на процесора. Конфигурираните като изходи пинове са в нискоимпедансно състояние (т.е. са с ниско електрическо съпротивление), което предполага протичане на недопустимо силни токове при наличие на външно напрежение.

5. Какво означава 1024 нива резолюция на аналоговите входове?

Аналого-цифровото преобразуване е обширна тема и тук няма да разглеждаме всички видове и методи за постигането му. Най-общото определение е: Процес, при който аналогова величина (в случая ел. напрежение) се преобразува в двоичен цифров код. Най-простото АЦП е еднобитовото – няма напрежение = логическа 0, има напрежение = логическа 1. Кодирането с повече битове е много по-точно (с по-голяма резолюция), но е по-бавно и по-трудно от схемотехническа гледна точка. Съвременните технологии обаче преодоляват тези пречки. Преди 20 години беше немислимо в една интегрална схема да има 6 10-битови АЦП. (в интерес на истината не зная дали са 6 или е само един, който

програмно се превключва между различните входове). Но това е без значение - важен е крайният резултат. А той е, че подаденото на аналогов вход напрежение в обхвата от 0 до +5 V се изразява цифрово в десетбитова десетична стойност от 0 до 1023. Каква е резолюцията на преобразуването при тия данни? Разделете 5 V на 1024. Отговорът е 0,0048828... V. Т.е. всяка една цифрова стойност представя разлика спрямо съседната си с приблизително 4,88 мV (миливолта). Това означава, че ако контролирате някакво външно напрежение, в интервала до 5 волта, всяко негово изменение с приблизително 4,88 миливолта ще предизвика промяна в стойността, прочетена от контролера с единица. Можем ли да измерваме с контролера и по-високи напрежения? Можем. Как? Ще видим в последната точка.

И още един интересен въпрос: Тъй като ще подаваме всякакви напрежения към входовете, а те ще бъдат привеждани винаги в стойност между 0 и 1023, как да направим така, че на дисплея на Ардуиното или в серийния монитор да виждаме истинската стойност във волтове. (5 V няма нищо общо с числото 1023, нали). Този проблем се решава програмно с вградена функция на програмния език. И решава проблема при включване на всякакви външни аналогови датчици.

6. Какво означава ШИМ и как се ползва, като аналогов сигнал от изход?

Това е най-трудният за обяснение въпрос, защото обяснението му включва интегриране от висшата математика. Но ще опитам да го обясня графично и на достъпен език.

ШИМ означава „Широчинно-импулсна модулация“ и в случая се ползва като метод за обратното на АЦП – ЦАП (цифрово-аналогов преобразувател). Това е ЦАП, който обаче не е точно в пълния смисъл ЦАП, защото на изхода на контролера не получаваме аналогово напрежение, не можем да получим определено напрежение (например 3,5 V). Този ЦАП работи, да го наречем половинчато. Контролерът върши половината работа, а товарът върши останалата половина. Това означава ни повече ни по-малко, че ако няма товар – нямаме и преобразуване. На изхода ще намерим просто някакви цифрови импулси. От тях трябва да започне обяснението.

Широчинно-импулсна модулация: Метод, при който под въздействие на управляващия сигнал се променя ширината (или коефициента на запълване) на импулсите, като тяхната честота, фаза и амплитуда остават неизменни.



Кой е управляващият сигнал в нашия случай? Това е цифрова стойност от 0 до 254, която се подава от контролера към този вграден ШИМ модулатор. Когато подадем ниска стойност, коефициентът на запълване на импулсите е нисък (импулсите са тесни), а когато подадем висока стойност, импулсите са широки. Ширината им е пряко свързана с подадената стойност.



Това е всичко, което прави контролерът и неговият ШИМ модулатор. Другата половина от ЦАП се извършва от самия товар или консуматор. Това, което трябва да се знае за аналоговите консуматори е, че тяхната мощност зависи пряко от тока и напрежението и се изразява в тяхното произведение. $P = UI$. Как, обаче, се изчислява мощността, когато захранващото напрежение е импулсно и има формата, показана на горните фигури? С една дума – интегриране.

Реално подадената към консуматора мощност е интеграл от импулсите за целия период на сигнала (от началото на единия до началото на следващия импулс). Това интегриране може да се изрази графично, като приемем, че подадената мощност за един период е изразена от площта на импулса „1“ за този период. Вижте следващите фигури.



Интегриране на тесните импулси - малка мощност

Площта на тесните импулси (червено) е интегрирана и е разпределена равномерно за целите периоди (зелено). Нивото, до което достига зеленото е и интегрираната мощност (реалната) подадена към консуматора.



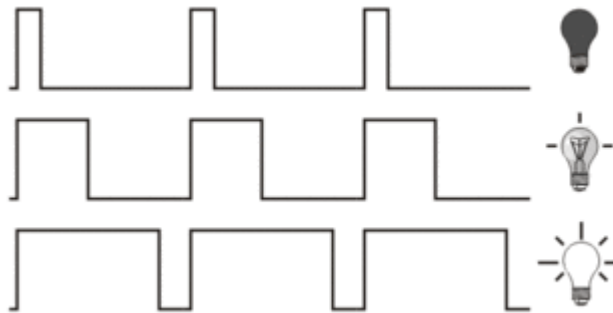
Интегриране на широките импулси - голяма мощност

Площта на широките импулси (синьо) е интегрирана и е разпределена равномерно за целите периоди (жълто). Нивото, до което достига жълтото е и интегрираната мощност (реалната) подадена към консуматора.

От фигурите се вижда, че колкото по-широки са импулсите, толкова повече мощност се подава към консуматора.

Логичният въпрос е: Кой извършва това интегриране? Кой върши другата половина от ЦАП? Ами – самият товар. Единственото условие е този товар да бъде достатъчно инертен, за да извърши това интегриране. Или дори да не е инертен, но резултатът да „лъже“ човешките сетива, че всичко върви гладко. Ето два примера:

Първи пример – лампа с нажежаема жичка. Импулсите следват толкова бързо, че жичката не успява да изстине в паузите между импулсите. Наблюдателят вижда постоянно светеща крушка с различен интензитет в зависимост от запълването на импулсите.



Втори пример – заменяме лампата със светодиода. Той е неинертен – това означава, че когато има импулс ще свети, а в паузите ще е угаснал. Ще видим ли импулсите? Всъщност не, защото както казахме, те следват с висока честота, а човешкото око е инертно и не забелязва тези примигвания. В този случай интегрирането се извършва от човешкото око, което става част от ЦАП. Все пак ще видим изменението на интензитета на светене на светодиода, защото всъщност към него се подава различна мощност при различна стойност от контролера.

7. Стандарт на цифровите входове и изходи: електрически параметри на сигналите.

За малкото време, с което разполагам не успях да намеря надеждни данни за спецификациите, на които отговарят входовете и изходите на контролера (микропроцесора). Знаем само, че е направен на базата на CMOS технология и изходите/входовете му са съвместими със стандарта TTL (транзисторно-транзисторна логика). Също така знаем, че когато са конфигурирани като входове, изходите са във високоимпедансно състояние (с високо съпротивление), а когато са конфигурирани като изходи са в нискоимпедансно състояние.

От всички тези данни можем да направим следните изводи:

- можем да свързваме към контролера устройства, които отговарят на стандарта TTL.
- ако сме конфигурирали вход, на всяка цена трябва да дефинираме началното му състояние (да го свържем през резистор към „+“ на захранването или към маса (GND)). В противен случай високоимпедансното му състояние ще способства за лесно проникване на паразитни и смущаващи сигнали през него.
- ако сме конфигурирали изход (нискоимпедансно състояние) трябва да се уверим, че няма да подадем към него външно напрежение, тъй като това може да предизвика протичането на недопустимо силен ток през електрониката на чипа (с много отрицателни последици).

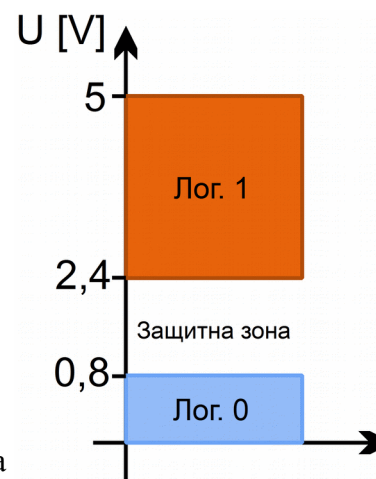
Няколко думи за TTL (на български TTL – транзисторно-транзисторна логика).
Вижте фигурата:

Това е графично представяне на сигналите в стандарта TTL

За да се приеме един сигнал за логическа 0 той трябва да е между 0 и 0,8 V.

Логическа единица имаме, когато сигналът е с напрежение над 2,4 V.

Зоната от 0,8 до 2,4 волта се нарича защитна зона и осигурява да не бъде грешно разчетена информацията, ако е близо до границите на определените логически нива. С една дума тази зона защитава системата от смущения. По същата причина логическата нула не е дефинирана строго на 0 V. Ако беше така проникналите смущения, дори с минимална амплитуда биха могли да смутят работата на устройството.



Друг важен параметър на TTL е максималната честота на подаваните импулси, която може да бъде обработена. Обикновената TTL може да работи с честоти до 30 MHz. Има модификации на TTL, които могат да работят и на по-високи честоти. Имайки предвид, че за Atmega328 дават максимална тактова честота от 20 MHz – теоретично би могъл да работи на такива честоти. На практика трябва да вземем предвид времената,

необходими за функциониране на съответните възли в процесора, както и времето необходимо на софтуера да обработи нашите сигнали, при което реалната скорост на обработка на сигналите драстично ще намалее. А това означава и намаляване на честотата на външните сигнали, които искаме да обработим. Затова тази честота зависи от конкретния случай.

Графиката с логическите зони дава още едно обяснение защо е необходимо да дефинираме входовете с външни резистори. Ако входът остане да виси, различни външни смущения, включително статично електричество могат да попаднат и да променят по случаен начин потенциала на входа, при което той ще отчита различни стойности (ту нула, ту единица или ще остане в неидентифицирано състояние ако напрежението е в рамките на защитната зона).

8. Практически схеми на включване на външни устройства и датчици към пиновете на микроконтролера.

8.1. Входи.

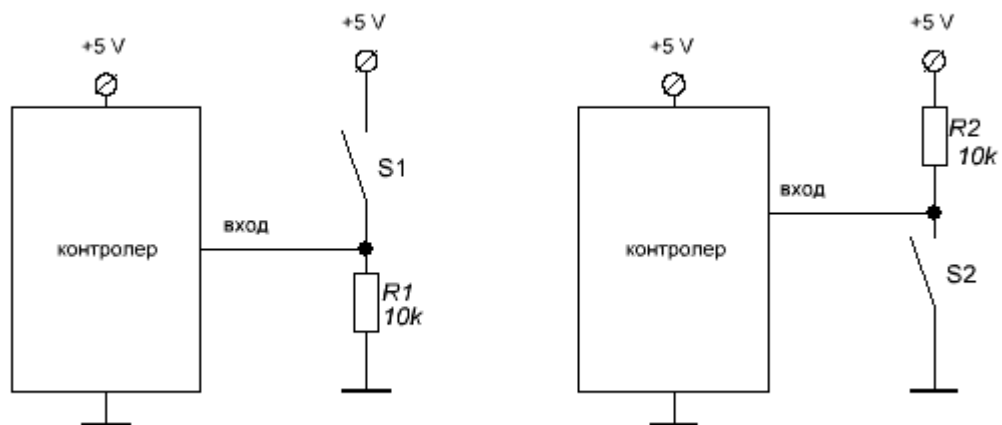
8.1.1. Цифрови входовете

Първо цифровите входове трябва да бъдат дефинирани по познатия начин.

Както вече на няколко пъти стана дума, (освен ако не подаваме към такъв вход стандартен сигнал от друго устройство), трябва да вземем мерки за неговото дефиниране в начално състояние. Ще дам пример с обикновен бутон, който в нормално състояние е с отворени контакти, а при натискане се затварят контактите. Двата възможни варианта на свързване към контролера са следните:

В първата схема входът е дефиниран към логическа нула. Очаква се кодът на контролера да обработи събитие, когато това състояние се измени от „LOW” на „HIGH“, т.е. когато бъде натиснат бутонът и контактите му се затворят.

Във втората схема входът е дефиниран към логическа единица. Очаква се кодът на контролера да обработи събитие, когато това състояние се измени от „HIGH” на „LOW“, т.е. когато бъде натиснат бутонът и контактите му се затворят.



Разбира се, софтуерно можем да направим точно обратното – натиснатият бутон да е изходното състояние, а неговото отпускане да генерира събитие.

Кое ще бъде конкретното схемно решение, което ще изберете, зависи от вас и от предназначението на устройството, което проектирате. Например ако ще бъде някакъв контролер, който ще следи отварянето на вратите на автомобил, тогава трябва да изберете вторият вариант, защото бутоните на вратите така или иначе са свързани към корпуса на автомобила (към общия проводник). Съпротивлението на резистора може да бъде в широки граници от стотици ома до десетки килооми. Колкото по-малко е съпротивлението, толкова по-шумоустойчив ще бъде входът, но не може да бъде прекалено малко, за да не представлява късо съединение при натискане на бутона.

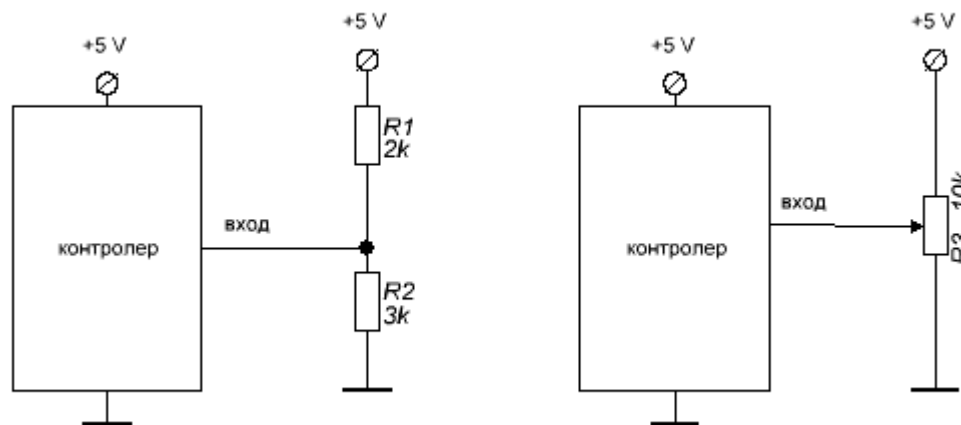
В допълнение към тази точка: Механичните контакти са източник на характерно смущение, наречено „трептене“, което може да предизвика при еднократно натискане на бутона, софтуерът да засече многократни импулси, които да объркат работата му (например брое 10 натискания, а софтуерът ви засича 45). Този проблем се разрешава със софтуерни средства.

8.1.2. Аналогови входове.

Както вече казахме на тях може да се подават напрежения от 0 до 5 волта, които ще бъдат кодирани в стойности от 0 до 1023. Най-лесният начин за подаване на такива напрежения е чрез делител на напрежение или потенциометър:

В първата схема съотношението на резисторите е 2 към 3. Това означава, че и напрежението ще се разпредели 2/3 върху тях. А това значи, че върху R2 ще имаме 3 волта и контролерът би трябвало да прочете стойност около 614 ($3V / 0,00488 V$)

По същият начин работи и схемата с потенциометър, но чрез него може да се регулира подаваното към контролера напрежение.



На базата на делител на напрежение могат да се включват към контролера всякакви аналогови датчици: фоторезистори, терморезистори (термистори), магнитни и всякакви други. Ето пример на възел за измерване на температура с термистор:

Термисторът е полупроводников елемент, чието съпротивление зависи обратнопропорционално на температурата – когато температурата се покачва – съпротивлението му намалява. Включен по този начин към + на схемата при покачването на температурата съпротивлението му ще намалява, а с него и падът на напрежение върху него. Това ще предизвика покачване на напрежението върху R2 съответно към входа на контролера, който ще прочете съответно по-висока стойност от изхода на АЦП.

Тук е мястото да отговорим на въпроси, които поставихме по-горе:

Можем ли да мерим по-високи напрежения от 5V? Можем ли да приведем прочетените от АЦП стойности към реални единици?

И двата отговора са: да, можем.

Задание първо: Да измерваме с контролера постоянно напрежение с максимална стойност 180 волта.

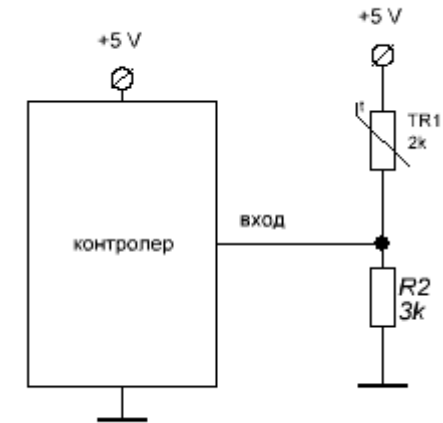
Решение: Първо нека приемем, че максималното напрежение може при аварийна ситуация да достигне 200 V и да вземем това напрежение за основа при нашите изчисления, тъй като тази кръгла стойност, кратна на 5 ще ни улесни. За да намалим напрежението от 200 на 5 волта на входа на контролера ще използваме обикновен делител на напрежение – виж първата схема с делителя. Ще изберем R2 да бъде 5 килоома. Сега трябва да изчислим съпротивлението на R1. Върху делителя ще бъде приложено общо напрежение 200 V. Приемаме, че при този случай напрежението върху R2 трябва да е максимум 5V. При това положение върху другото съпротивление R1 трябва да „падне“ останалото напрежение от 195 волта. Веднага се вижда зависимостта от 1 волт на килоом. Това означава, че R1 трябва да е със стойност 195 килоома.

Задание второ: Очевидно контролираното външно напрежение с максимална стойност 200V и минимална 0V ще се подава към контролера съответно с максимална стойност 5V и минимална 0V. При това прочетената цифрова стойност ще се изменя от 1023 до 0. Как да накараме контролера да ни подава директно стойността на външното напрежение във волтове?

За целта се ползва функцията `map()`.

В случая програмният код ще представлява нещо подобно: `outputValue = map(sensorValue,0,1023,0,200)`; където

- `sensorValue` е променливата, която съдържа стойността прочетена от АЦП;
- `0,1023` е диапазона на тази стойност (по правило може да се вземе и по-тесен диапазон);
- `0,200` е диапазонът на външната величина, към чиято стойност искаме да приведем крайния резултат.

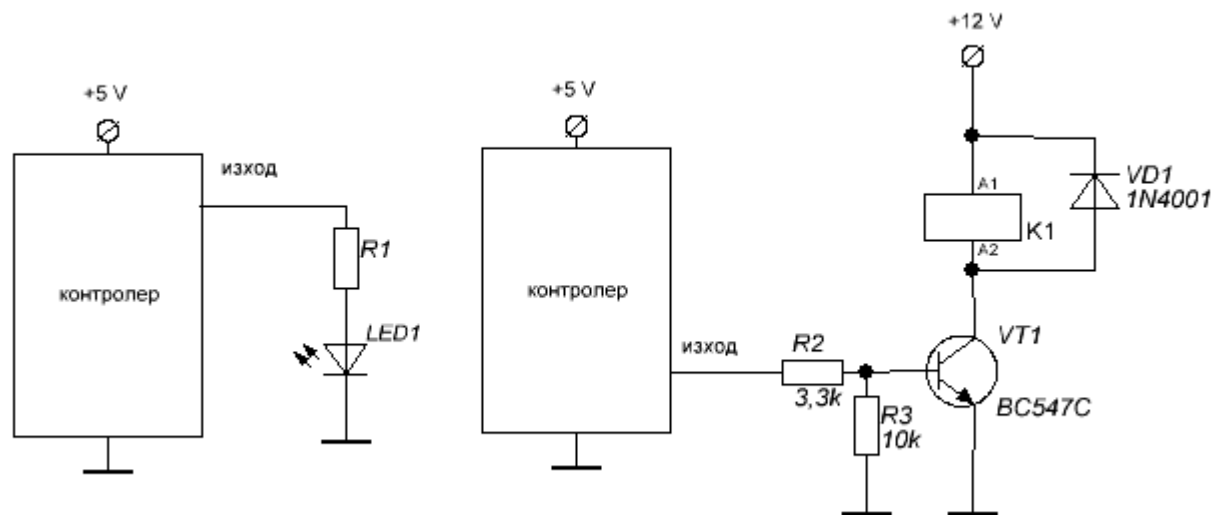


С думи тази функция казва нещо от сорта: „Ако прочетеш на входа стойност 1023, значи тъпакът, дето те гледа умно отсреща, си е подал на делителя 200 V. Покажи му 200!“ :)

8.2. Изходи.

Няма да ги делим на цифрови и аналогови. Както споменах най-важният им електрически параметър е максималният ток, който е до 40 mA. Значи, каквото и да включваме към изхода трябва да се уверим, че няма да черпи повече от 40 mA. Също така отново напомням: **УВЕРЕТЕ СЕ, ЧЕ НЕ СТЕ ПОДАЛИ НИКАКВО ВЪНШНО НАПРЕЖЕНИЕ** към пин, който ще дефинирате като изход.

За да защитим изхода ВИНАГИ трябва да ползваме баластно съпротивление. Колко да бъде то? Ами $5V / 0,04A$ показва 125 ома. Нека бъде поне 150 ома за всеки случай. Ако от изхода сигналът трябва да иде към устройство с транзисторен вход трябва да защитим този транзистор като включим между изхода на контролера и базата на транзистора резистор със стойност около 3 до 5 килоома. В противен случай изходният ток на контролера може да повреди базовия преход на транзистора. (Говорим за биполарни транзистори, при полевите този проблем не стои).



По първата схема нека изчислим баластното съпротивление на светодиода. Тук трябва да вземем предвид не само товарносимостта на изхода на контролера, но и максимално допустимия ток през светодиода. Обикновените червени, зелени, жълти светодиоди консумират около 10 – 15 mA, като падът на напрежение върху тях е около 1,5 волта. Значи от общото напрежение на изхода 5V вадим пада върху диода 1,5V. Остават 3.5V.

Делим ги на тока: $3,5V/0,015A$ и получаваме 233,3 ома. Избираме стандартна стойност 240 ома за баластното съпротивление. То ще ни гарантира защита на изхода на контролера, а също и на светодиода.

Обърнете внимание на втората схема. Тя е типична за свързване на външни мощни консуматори към контролера. R2 защитава базата на транзистора. Товарът е електромагнитно реле. Обърнете внимание на диода, свързан противоположно на захранващото напрежение. Помнете ли за проблема със самоиндукцията? Този диод решава проблема. При прекъсване на напрежението към релето (запушване на транзистора) магнитното поле, което е удържало до този момент котвата на релето в задействано положение много бързо се връща минавайки през навивките на релето. Това предизвиква индуцирането на многократно по-голямо напрежение от захранващото (може да бъде от порядъка на 100 и повече волта) и с обратна полярност. Протичащият ток може да повреди транзистора или дори да се върне до контролера. Диодът VD1 решава проблема като дава накъсо тока на самоиндукцията. Непременно монтирайте такива диоди, когато ползвате електромагнитни релета.

Другото решение е да ползвате SSR – Solid state relay. Реле с твърдо тяло. По същество това обикновено е фотосимистор: маломощен светодиод осветява прехода на симистора и го включва и съответно изключва. Така освен, че избягвате проблема със самоиндукцията, имате възможност да управлявате много мощни вериги – до няколко десетки киловати – директно през изхода на Ардуиното.

С това смятам да завърша този много полезен урок.

Разбира се, съществуват още много на брой схемни решения и за датчици и сензори, и за изпълнителни устройства, но най-важните принципи и правила, които да спазваме, за да не се сбогуваме със скъпоценните платки (внезапно и със светлинни, звукови ефекти и димни завеси) са описани според мен достатъчно подробно.



Успех!