



УНИВЕРЗИТЕТ У НИШУ
ЕЛЕКТРОНСКИ ФАКУЛТЕТ



Пројектовање и реализација система даљинског управљања паметног прекидача

Мастер рад

Студијски програм: Електроника и микросистеми

Студент:

Александар Пантовић, бр. инд. 1593

Ментор:

Доц. др Горан Николић

Ниш, септембар 2024. година

Универзитет у Нишу
Електронски факултет

Пројектовање и реализација система даљинског управљања паметног прекидача
Design and implementation of a remote control system for a smart switch

Мастер рад
Студијски програм: Електроника и микросистеми

Студент: Александар Пантовић, бр. инд. 1593
Ментор: Доц. др Горан Николић

Задатак: Упознати се са местом и извором класичног прекидача и утичнице у систему кућног напајања наизменичном струјом. Пројектовати и реализовати систем даљинске контроле базиран на ESP32 микроконтролеру са одговарајућом оптокаплерском и драјверском спрегом за контролу путем релеја. Верификовати исправност система примером укључивања и исклучивања потрошача.

Датум пријаве рада: 01.07.2024.

Датум предаје рада: 23.06.2024.

Датум одбране рада: 26.09.2024.

Комисија за оцену и одбрану:

1. Доц. др Горан Николић, Председник Комисије

2. Проф. др Татјана Николић, Члан

3. Проф. др Сандра Ђошић, Члан

Садржај

Увод	1
Реализација паметног прекидача (Arduino Uno).....	2
Arduino Uno	3
IR предајник и IR пријемник	6
IR предајник	6
IR пријемник.....	7
Једноканални релејни модул	8
Принцип рада релеја.....	9
Електрична шема једноканалног релејног модула	10
Практична реализација паметног прекидача (Arduino Uno)	11
Fritzing	11
Софтверска реализација паметног прекидача (Arduino Uno).....	12
Коришћење библиотеке Irremote	12
Објашњење кода паметног прекидача	15
Демонстрација пројекта паметног прекидача (Arduino Uno).....	17
Реализација паметног прекидача (ESP32-S NodeMCU).....	19
ESP32-S NodeMCU	20
ESP-WROOM-32s.....	22
Пинови за strapping	24
CPU и унутрашња меморија.....	25
Спољашњи флеш и SRAM	26
Распоред физичких пинова плочице ESP32-S NodeMCU	26
Практична реализација паметног прекидача (ESP32-S NodeMCU)	29
Софтверска реализација паметног прекидача (ESP32-S NodeMCU).....	30
Коришћење библиотеке WiFi	30
Коришћење библиотеке WebServer.....	32
Објашњење кода паметног прекидача (ESP32-S NodeMCU)	33
Демонстрација пројекта паметног прекидача (ESP32-S NodeMCU).....	37
Паметна утичница за кућну употребу	40
AC/DC Претварач	41
Step-down трансформатор	43
Конструкција трансформатора.....	44
Принцип рада трансформатора	44
Исправљачи	45

Принцип рада полуталасног исправљача.....	45
Принцип рада пуноталасног исправљача	46
Регулатор напона	49
Принцип рада прекидачког регулатора.....	50
Практична реализација паметне утичнице за кућне услове	51
Blynk платформа	52
Поступак за повезивање плочице са Blynk апликацијом.....	53
Софтверска реализација паметне утичнице за кућне услове	54
Коришћење библиотеке BlynkSimpleEsp32.....	54
Објашњење кода паметне утичнице за кућне услове	55
Коначна реализација и демонстрација паметне утичнице	58
Закључак	60
Литература.....	61

Увод

Internet of things (IoT) представља концепт повезивања физичких уређаја, возила, кућних апаратова, и других предмета са Интернетом, омогућавајући им да међусобно комуницирају и размењују податке. Ова технологија омогућава стварање паметних система у којима уређаји могу радити аутономно или се контролисати на даљину путем апликација, чиме се олакшава свакодневни живот и повећава ефикасност у различитим областима. IoT је трансформисао начин на који доживљавамо технологију, спајајући дигитални и физички свет. Уређаји повезани на IoT могу прикупљати податке из свог окружења, анализирати их и доносити одлуке на основу тих информација, а све то без потребе за људском интервенцијом. Примери IoT уређаја укључују паметне термостате који аутоматски подешавају температуру у дому, сензоре у индустрији који прате стање машине, као и носиве уређаје који прате здравствене параметре корисника.

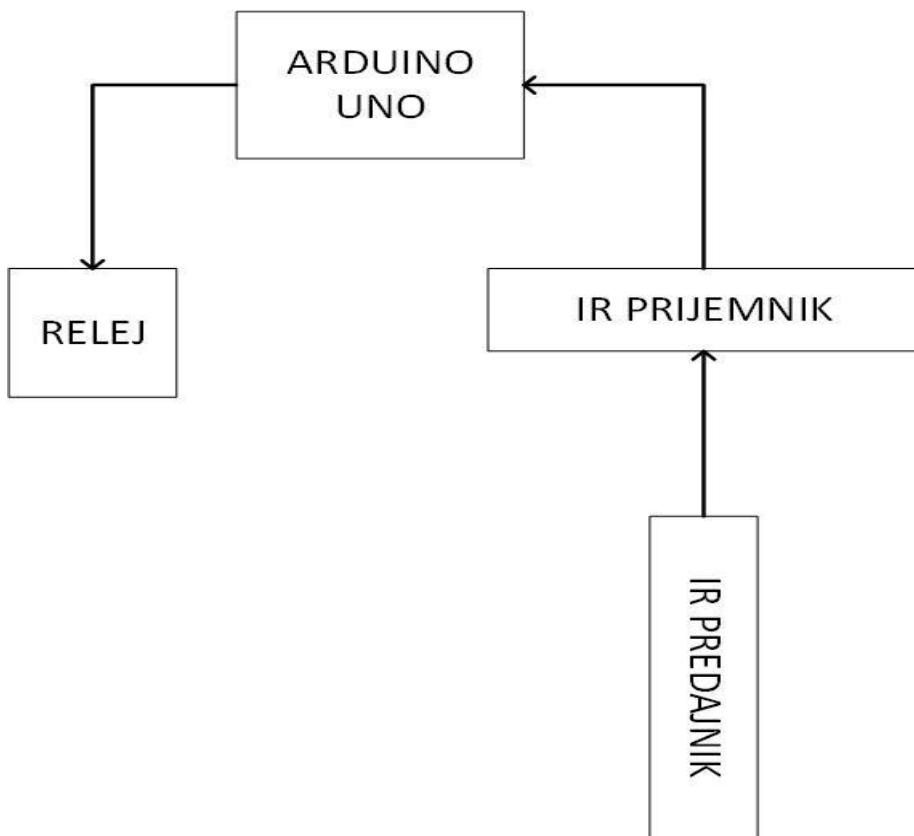
Кључне компоненте IoT система укључују сензоре, мрежну повезивост, анализику и корисничке интерфејсе. Сензори су задужени за прикупљање података из околине, мрежна повезивост омогућава пренос тих података на сервере или облаке, где се анализира, док кориснички интерфејси пружају начин за интеракцију са подацима и уређајима. Један од главних изазова IoT-а је сигурност, јер сваки повезани уређај може потенцијално бити мета cyber напада. Зато је потребно обезбедити робусне сигурносне протоколе и редовно ажурирати софтвер како би се заштитили подаци и инфраструктура. IoT такође отвара врата за нову еру аутоматизације у индустрији, познату као Индустрија 4.0. У овој ери, производне линије и логистички системи постају аутономни, чиме се смањује потреба за људском интервенцијом и повећава ефикасност.

У оквиру овог мастер рада, истражен је развој система за даљинско управљање уређајима користећи IoT технологије. Конкретно, фокус је на имплементацији система за контролу релеја путем даљинског управљача и интернет апликације (Blynk), користећи ESP32 микроконтролер као срце система. Овај пројекат демонстрира како се IoT може искористити за побољшање флексибилности и функционалности система за аутоматизацију, омогућавајући корисницима да контролишу уређаје с било које локације, у реалном времену. На овај начин, корисници имају већу контролу над својим окружењем, што може значајно повећати комфор и енергетску ефикасност. Мастер рад обухвата израду утикача (користећи комерцијално доступну OG утичницу) и демонстрацију рада истог уз помоћ потрошача у виду сијалице. Контрола укључивања/искључивања се реализује путем релеја.

У првом делу рада, фокус ће бити на традиционалнијем приступу користећи Arduino Uno микроконтролер, где се контрола релеја остварује искључиво путем даљинског управљача. У другом делу, прелази се на напреднији приступ са ESP32, који омогућава не само контролу преко даљинског управљача, већ и путем уређаја са Интернетом, чиме се постиже много већа флексибилност и могућност интеграције са другим IoT системима.

Овај рад не само да истражује техничке аспекте ових система, већ и анализира њихове предности, мање и потенцијалне примене, пружајући читаоцу свеобухватан увид у могућности IoT технологија у области аутоматизације. У будућности, очекује се да ће IoT постати још више интегрисан у наш свакодневни живот, са све већим бројем уређаја који ће бити повезани на Интернет, омогућавајући паметније и ефикасније управљање ресурсима, енергијом, и нашим временом.

Реализација паметног прекидача (Arduino Uno)



Слика 1. Блоковска шема реализације паметног прекидача

Блоковска шема приказана на слици 1. илуструје основну архитектуру система паметног прекидача коришћењем Arduino Uno микроконтролера, IR пријемника, IR предајника (даљински управљач) и релеја.

- Arduino Uno је централна компонента система која контролише релеј на основу сигнала примљених од IR пријемника. Микроконтролер прима сигнал из IR пријемника, обрађује га и доноси одлуку да ли ће укључити или искључити релеј.
- IR пријемник је повезан са Arduino Uno плочицом и служи за примање инфрацрвених (IR) сигнала са даљинског управљача (IR предајник). Када корисник притисне дугме на даљинском управљачу, сигнал се преноси до IR пријемника.
- IR предајник (даљински управљач) емитује инфрацрвени сигнал када корисник притисне одговарајуће дугме. Овај сигнал садржи информације о жељеној акцији (нпр. укључивање или искључивање уређаја) које IR пријемник декодира и шаље даље ка Arduino Uno плочици.

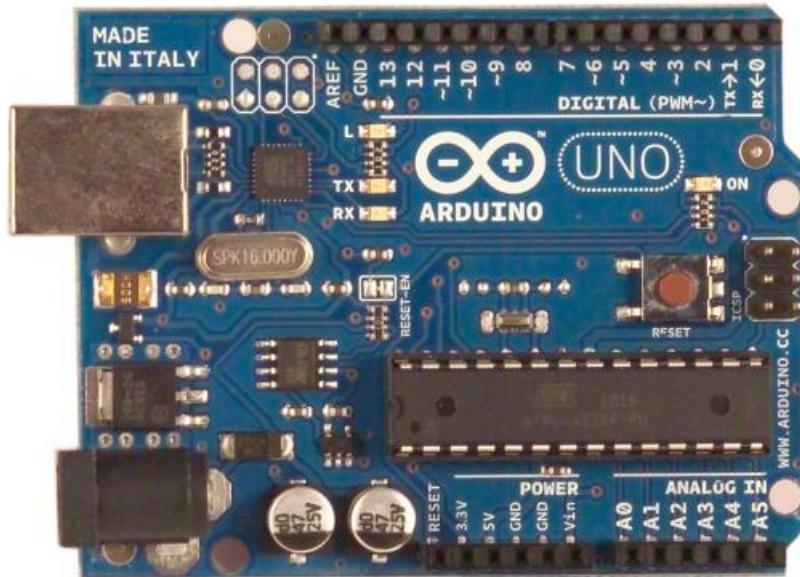
- Релеј је електронски склоп који се понаша као прекидач, контролисан од стране Arduino Uno микроконтролера. На основу примљених инструкција, релеј може укључити или искључити потрошач повезан са њим.

У овом систему, када корисник притисне дугме на даљинском управљачу, у нашем случају ће бити коришћен даљински од телевизора, IR предајник шаље сигнал ка пријемнику. IR пријемник прослеђује сигнал ка Arduino Uno микроконтролеру, који затим, у зависности од примљене команде, укључује или искључује релеј. Ова архитектура омогућава једноставно даљинско управљање уређајем преко инфрацрвених сигнала.

Arduino Uno

Arduino Uno је микроконтролерска плоча заснована на ATmega328. Има 14 дигиталних улазно/излазних пинова (од којих 6 може бити коришћено као PWM излази), 6 аналогних улаза, кристални осцилатор од 16 MHz, USB конекцију, напонски прикључак, ICSP заглавље и тастер за ресетовање. Садржи све што је потребно за подршку микроконтролеру; доволно је само повезати га са рачунаром путем USB кабла или напајати га AC-DC адаптером или батеријом како бисте започели рад.

Uno се разликује од свих претходних плоча по томе што не користи FTDI USB-to-serial драјвер чип. Уместо тога, садржи Atmega8U2 чип програмиран као USB-to-serial конвертор. "UNO" значи "један" на италијанском и назив је одабран у част предстојећег издања Arduino верзије 1.0. Uno и верзија 1.0 ће бити референтне верзије Arduino платформе у будућности. Uno је најновији у серији USB Arduino плоча и референтни модел за Arduino платформу.



Слика 2. Arduino плочица

Arduino Uno може бити напајан путем USB везе или екстерног извора напајања. Извор напајања се аутоматски бира. Екстерно (не-USB) напајање може долазити или из AC-DC адаптера (пуњача) или батерије. Адаптер се може повезати тако што се 2.1 mm центар-позитивни прикључак убаци у напонски прикључак на плочи. Прикључци са батерије могу се повезати на GND и Vin пинове POWER конектора. Плоча може радити на екстерном напајању од 6 до 20 V. Међутим, ако се напаја са мање од 7V, пин од 5V може испоручивати мање од пет волти, што може учинити плочу нестабилном. Ако се користи више од 12V, напонски регулатор може да се прегреје и оштети плочу. Препоручени опсег напона је од 7 до 12 V.

Пинови за напајање су следећи:

- VIN: Улазни напон за Arduino плочу када се користи екстерни извор напајања (за разлику од 5 V из USB везе или другог регулисаног извора напајања). Можете довести напон кроз овај пин, или, ако напајате плочу путем напонског прикључка, можете му приступити кроз овај пин.
- 5V: Регулисан извор напајања који се користи за напајање микроконтролера и других компоненти на плочи. Ово напајање може долазити са VIN пина путем уграђеног регулатора, или бити обезбеђено путем USB-а или другог регулисаног извора напајања од 5V.
- 3V3: Извор напона од 3.3 V генерисан путем уграђеног регулатора. Максимална струја која се може повући је 50 mA.
- GND: Пинови за уземљење (ground).

Atmega 328 има 32 KB flash меморије за складиштење кода (од чега се 0,5 KB користи за bootloader); такође има 2 KB SRAM-а и 1 KB EEPROM-а.

Сваки од 14 дигиталних пинова на Uno плочици може се користити као улаз или излаз, користећи функције pinMode(), digitalWrite() и digitalRead(). Раде на 5 V. Сваки пин може давати или примати максимално 40 mA и има унутрашњи pull-up отпорник (који је по default искључен) од 20-50 kΩ. Поред тога, неки пинови имају специјализоване функције:

- Серијски: 0 (RX) и 1 (TX). Користе се за примање (RX) и слање (TX) TTL серијских података. Ови пинови су повезани са одговарајућим пиновима ATmega8U2 USB-to-TTL серијског чипа.
- Екстерни прекиди: 2 и 3. Ови пинови се могу конфигурисати да изазову прекид при ниској вредности, растућом или опадајућом ивицом, или промени вредности. Погледајте функцију attachInterrupt() за детаље.
- PWM: 3, 5, 6, 9, 10 и 11. Пружају 8-битни PWM излаз користећи функцију analogWrite().
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ови пинови подржавају SPI комуникацију, која, иако је подржана од стране хардвера, тренутно није укључена у ARDUINO језик.
- LED: 13. На дигитални пин 13 је повезан уграђени LED. Када је пин у HIGH стању, LED је укључен, а када је у LOW стању, LED је искључен.

Uno има 6 аналогних улаза, од којих сваки пружа 10 бита резолуције (тј. 1024 различите вредности). По default мере од нулте до 5 V, иако је могуће променити горњи

deo њиховог опсега користећи AREF pin и функцију analogReference(). Поред тога, неки пинови имају специјализоване функционалности:

- I2C: 4 (SDA) и 5 (SCL). Подржавају I2C (TWI) комуникацију користећи Wire библиотеку.

Постоји неколико других пинова на плочи:

- AREF. Референтна напонска вредност за аналогних улазе. Користи се са функцијом analogReference().
- Reset. Користи се за ресетовање плочице.

Arduino Uno има саморесетујући осигурач који штити USB портове вашег рачунара од кратког споја и преоптеређења. Иако већина рачунара пружа сопствену интерну заштиту, осигурач пружа додатни ниво заштите. Ако се на USB порту примени више од 500 mA, осигурач ће аутоматски прекинути везу све док се кратки спој или преоптеређење не уклоне.

Arduino Uno може бити програмиран помоћу Arduino софтвера. Одаберите "Arduino Uno ATmega328" из менија Tools > Board. ATmega328 на Arduino Uno долази са унапред учитаним bootloader-ом који омогућава учитавање новог кода без коришћења спољног хардверског програматора. Комуницира користећи оригинални STK500 протокол. Такође можете заобићи bootloader и програмирати микроконтролер путем ICSP (In-Circuit Serial Programming) конектора.

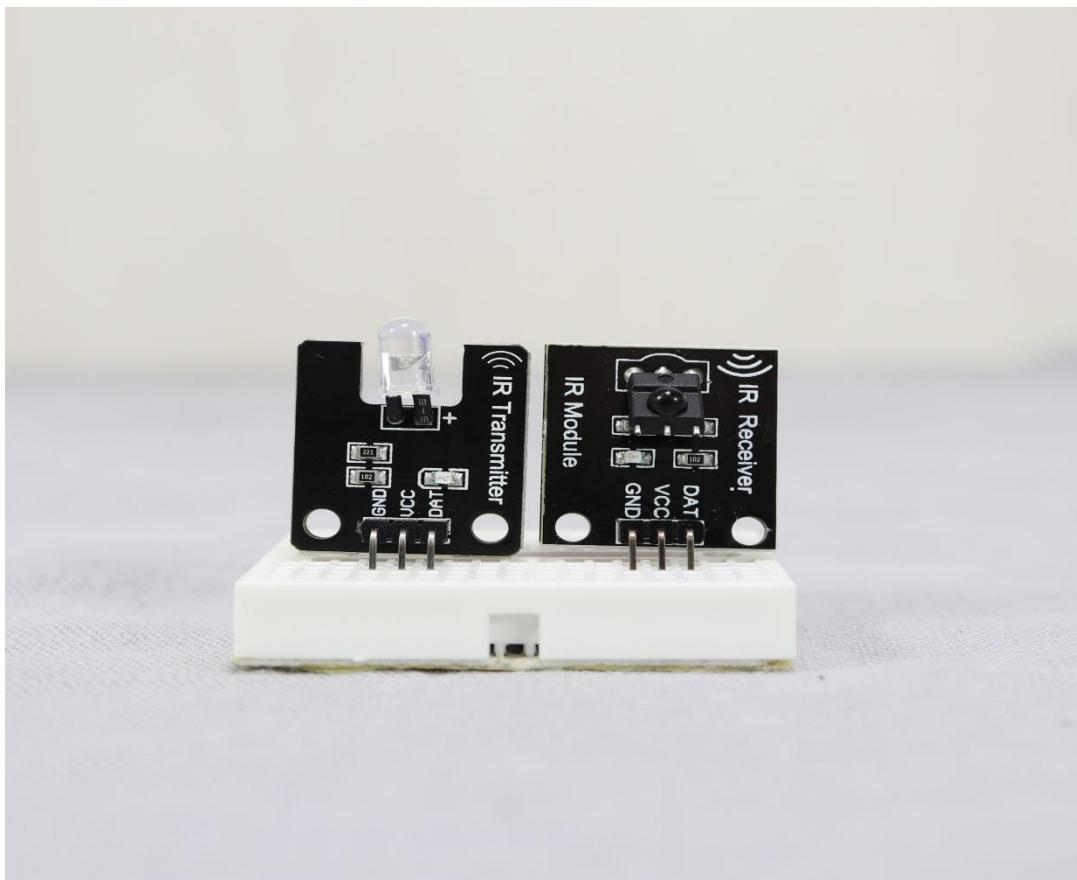
Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Слика 3. Карактеристике плочице

IR предајник и IR пријемник

Инфрацрвена (IR) комуникација је широко коришћена технологија комуникације. Типичан пример за IR комуникацију је даљински управљач за ТВ, који шаље IR сигнале телевизору. Конкретно, IR светло даљинског управљача се укључује и искључује (обично 38 kHz / 38.000 пута у секунди). Предности IR технологије су то што је јефтина и невидљива људском оку. Недостатак је што IR светло мора бити усмерено ка IR пријемнику. Као последица тога, IR комуникација се не би требала примењивати ако су пошиљалац и пријемник у различитим собама (у том случају, Bluetooth или WiFi би могли бити бољи избор). Овај рад пружа увод у модуле „IR предајник“ и „IR пријемник“. IR предајник и пријемник у овом пројекту су производи без имена. Често се ови производи налазе под различитим именима (нпр. IR емитер, IR пошиљалац) или са верзионим бројем као суфиксом (нпр. V1.0). Добра вест је да су веома слични по конструкцији. Стога, овај опис би требао послужити и за сличне производе (XX-M121, XX-53, KY-005, KY-022, YL-55) других производа и добављача (Keyes, Iduino, Open Smart,...).



Слика 4. IR предајник (лево) и IR пријемник (десно)

IR предајник

IR предајник модул има IR LED диоду која шаље сигнале. Обично, LED диода ради на око 940 nm, што значи да је веома светла у IR спектру. Поред LED диоде, модул има три пина за напајање и контролу модула: GND, VCC и DAT. И то је практично све што је потребно за слање сигнала путем IR комуникације. У ствари, ово је разлог зашто је IR комуникација тако јефтина. Потребна вам је само (IR) LED диода да бисте започели

комуникацију. Пинови GND и VCC служе за напајање модула са 5V извором напајања (нпр. GND и 5V пин на Arduino). Пин DAT одговара сигналном сигналу који треба да се пошаље IR пријемнику. Ако имате други тип IR предајника, пин SIG може имати другачије име. На пример, S за Iduino модул и KY-005, итд. У овом пројекту, уместо класичног предајника коришћен је даљински управљач од телевизора. Марка и модел нису од интереса у овом пројекту.



Слика 5. IR предајник за потребе пројекта

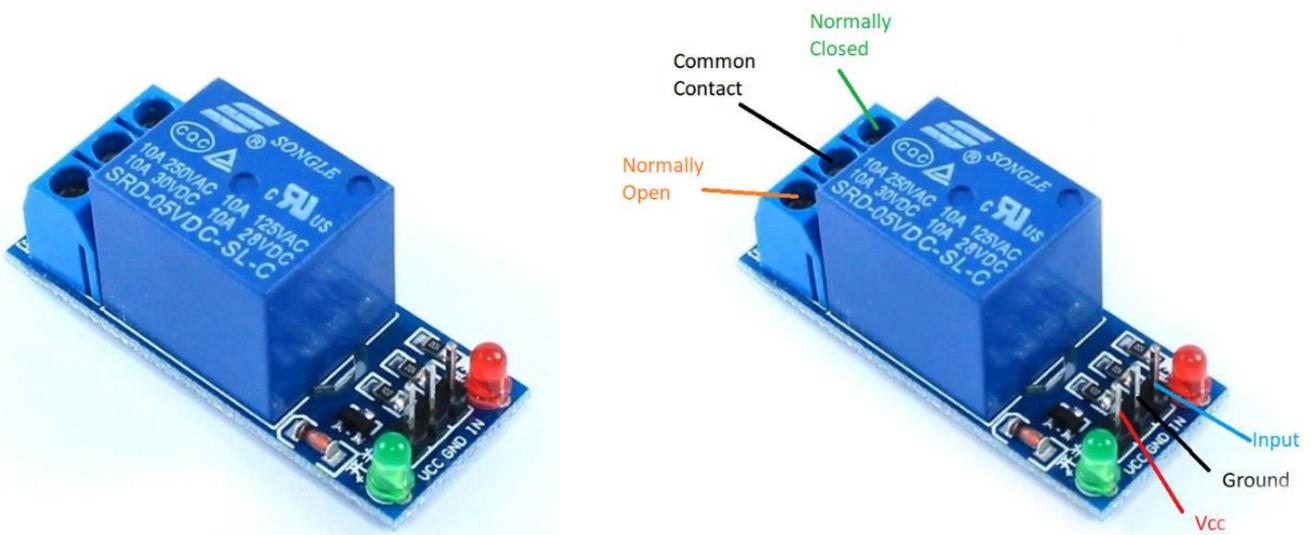
IR пријемник

IR пријемни модули обично имају интегрисану IR пријемну јединицу, као што су TSOP382, TSSP4P38 и слично. Ове јединице већ садрже све што је потребно за детекцију IR сигнала (PIN диода, пропусни филтер, демодулатор,...). Распоред пинова је врло сличан као код IR предајника. Ту су GND и VCC пинови за напајање модула напоном од 5V. Примљени IR подаци могу сеочитати са DAT пина. Распоред пинова је углавном исти и за друге IR пријемне модуле. Пин DAT често има различито име. На пример, S за KY-022, DO за YL-55...

Постоје различити протоколи који дефинишу како се сигнал/порука шаље са IR предајника на IR пријемник. Коришћење различитих протокола је, на пример, један од разлога зашто даљински управљач једног телевизора често не ради са другим телевизором. У овом пројекту се не бавимо детаљно темом IR протокола. Ипак, добро је знати да постоје различити протоколи (NEC, JVC, Samsung,...) за IR комуникацију. Идеја пројекта је да се користе IR пријемник у виду даљинског управљача и IR предајни модул. Пријемни модул је контролисан од стране Arduino Uno плочице. Arduino Uno прима сигнале који долазе путем даљинског управљача. Arduino Uno када прими одређене кодове за последицу ће имати команду за укључивање или искључивање релеја.

Једноканални релејни модул

Релеј је електромеханички уређај који користи електричну струју за отварање или затварање контаката прекидача.



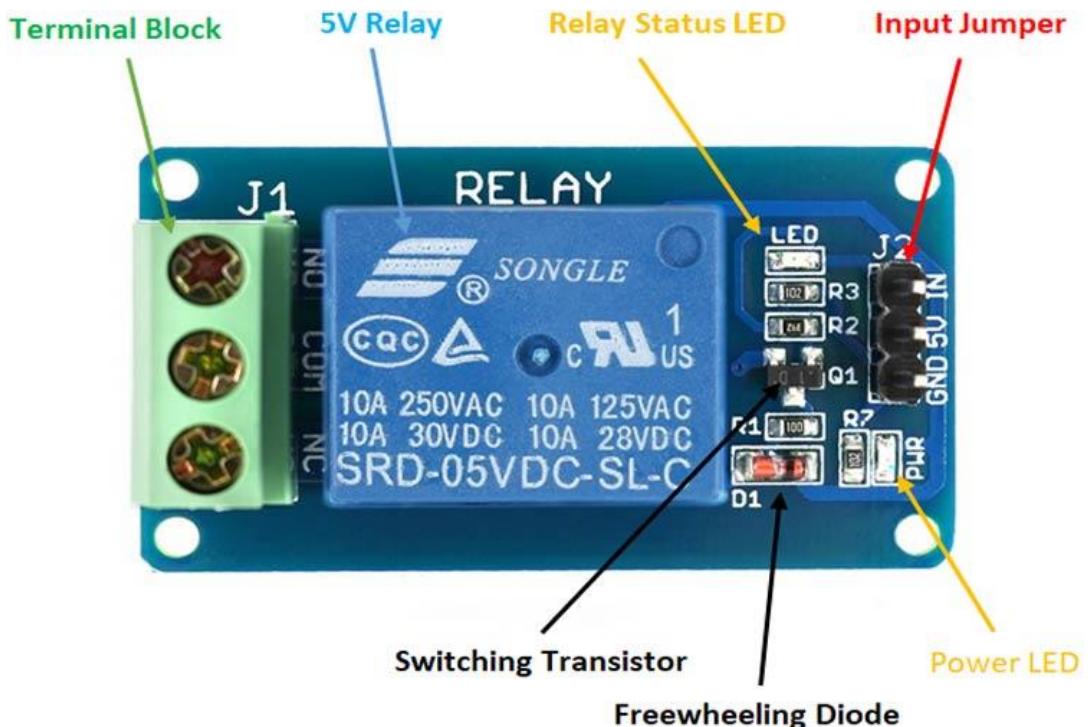
Слика 6. Једноканални релејни модул

Једноканални релејни модул је много више од обичног релеја, јер садржи компоненте које олакшавају пребацивање и повезивање, а служе и као индикатори за приказивање да ли је модул под напајањем и да ли је релеј активан.

Прво је блок са шрафовима. Ово је део модула који је у контакту са мрежним напајањем, па је потребна поуздана веза. Додавање шрафова омогућава лакше повезивање дебелих каблова за мрежно напајање, које би било тешко директно залемити. Три везе на блоку са шрафовима су повезане са нормално отвореним, нормално затвореним и заједничким терминалима релеја.

Друго је сам релеј, који је у овом случају у пластичном кућишту плаве боје. Пуно информација се може добити из ознака на самом релеју. Број модела релеја на доњој страни има ознаку "05VDC", што значи да се калем релеја активира при минималном напону од 5V – било који напон нижи од овог неће моћи поуздано да затвори контакте релеја. Такође постоје ознаке напона и струје, које представљају максимални напон и струју које релеј може пребацити. На пример, ознака у горњем левом углу каже "10A 250VAC", што значи да релеј може пребацити максимално оптерећење од 10A када је повезан на мрежни напон од 250V. Ознака у доњем левом углу каже "10A 30VDC", што значи да релеј може пребацити максималну струју од 10A једносмерне струје пре него што контакти буду оштећени. LED диода за статус релеја се пали када год је релеј активан и пружа индикацију да кроз калем релеја тече струја. Пин за напајање (5V) се користи за напајање калема релеја и LED диода. Такође има улазни пин (IN), који активира релеј када је подигнут на високи ниво. Прекидачки транзистор прима улаз који не може обезбедити доволно струје за директно покретање калема релеја и појачава га користећи напон напајања за покретање калема релеја. На овај начин, улаз се може покретати из микроконтролера или излаза сензора. Диоде спречавају пренапоне у пику

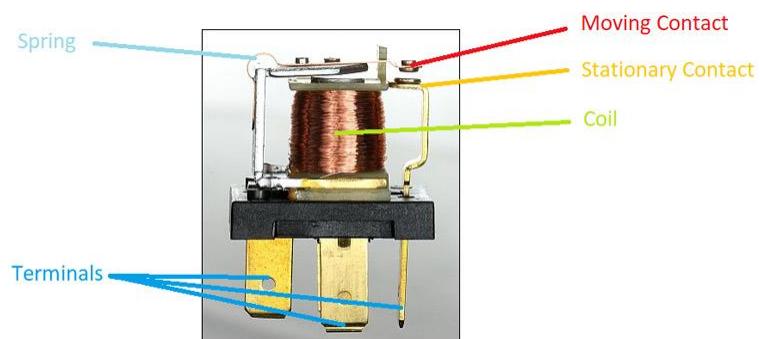
када се релеј искључи. LED диода за напајање је повезана на VCC и укључује се када год је модул под напајањем.



Слика 7. Разумевање једноканалног релејног модула

Принцип рада релеја

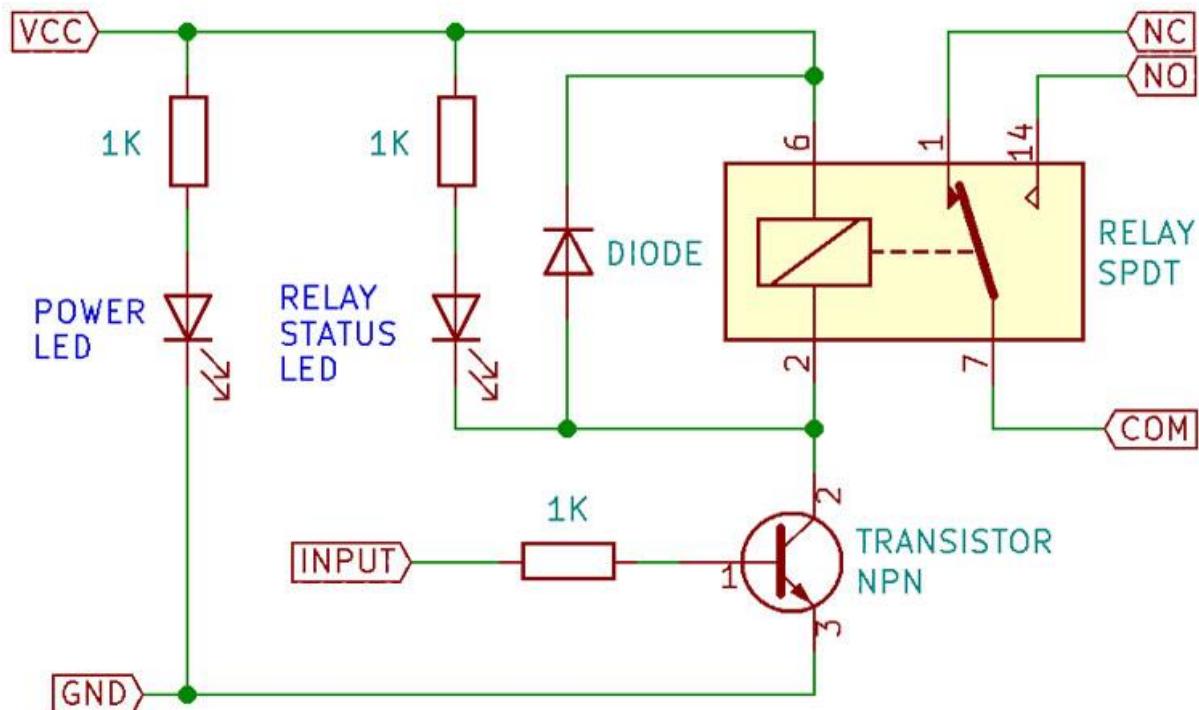
Релеј користи електричну струју за отварање или затварање контаката прекидача. Ово се обично постиже помоћу калема који привлачи контакте прекидача и спаја их када се активира, док их опруга раздваја када калем није под напоном. Постоје две предности овог система – Прво, струја потребна за активацију релеја је много мања од струје коју контакти релеја могу пребацивати, и друго, калем и контакти су галвански изоловани, што значи да између њих не постоји електрична веза. Ово омогућава да се релеј користи за пребацање мрежне струје кроз изоловани нисконапонски дигитални систем, попут микроконтролера.



Слика 8. Принцип рада релеја

Електрична шема једноканалног релејног модула

Додатне компоненте, осим самог релеја, су присутне јер није могуће директно управљати релејом преко пинова микроконтролера, дигиталне логике или сензора. Иако калем троши много мање струје него што релеј може пребацити, он и даље захтева релативно значајну струју – релеји мале снаге троше око 50mA, док релеји веће снаге троше око 500mA. Калем је такође индуктивно оптерећење, па када се искључи, развија се велики повратни напон који може оштетити уређај који га укључује и искључује. Због тога се додаје повратна диода постављена антипаралелно са калемом релеја како би се ограничио повратни напон. LED диоде могу бити додате овом основном кругу као индикатори, а понекад се додаје и оптичка изолација на улазу ради додатног побољшања изолације.



Relay Module Basic Schematic

Слика 9. Електрична шема једноканалног модула

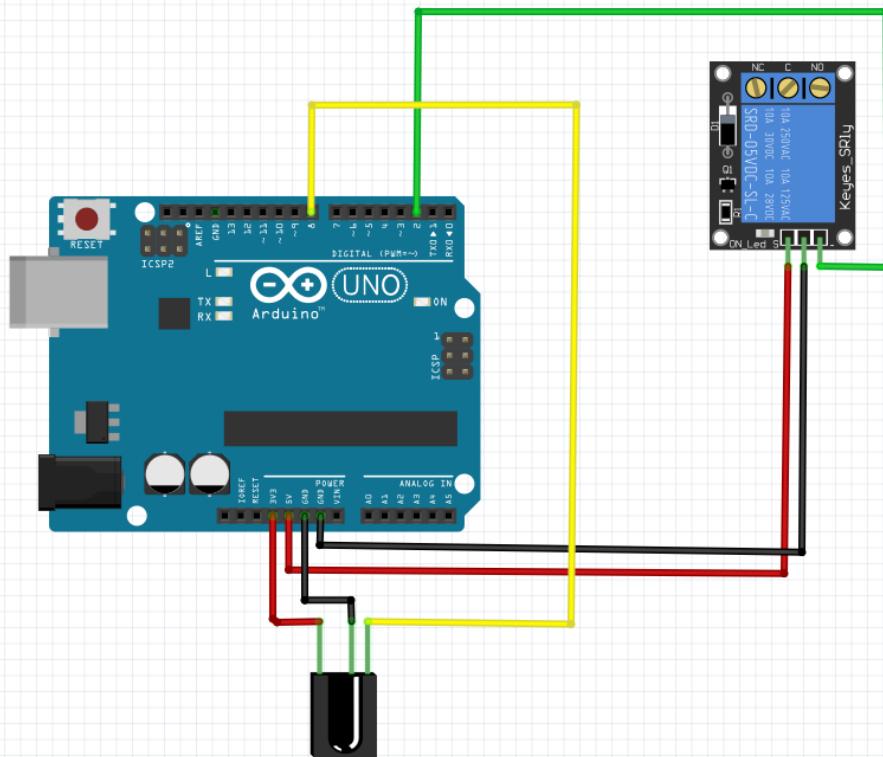
Број пинова	Име пина	Опис
1.	Relay Trigger	Улаз за активирање релеја
2.	Ground	0 V
3.	VCC	Напајање калема релеја
4.	Normally Open	Када се доведе 0 V на пин Relay Trigger долази до спрече између Common и Normally Open
5.	Common	Заједнички контакт
6.	Normally Closed	Када релеј нема напајање овај пин и Common су у спрези

Практична реализација паметног прекидача (Arduino Uno)

Након детаљног теоријског увода у први део пројекта, остаје нам практична реализација која ће обухватати хардверско повезивање и писање кода. Хардверско повезивање урадићемо у алату који се назива Fritzing, пре него што кренемо са повезивање у реалности. На овај начин долазимо до сигурнијег и јаснијег повезивања.

Fritzing

Након претходно образложених критеријума који су нас водили до избора компоненти за реализацију паметног прекидача можемо да кренемо у њихово повезивање. За почетак неопходно је да блоковску шему приказану у претходном поглављу преточимо у електронску шему на нивоу модула коришћењем програма Fritzing. Избор је последица поседовања визуелне подршке за Arduino развојне плочице. Fritzing је бесплатан софтвер за дизајнирање електронских шема, цртање повезивања компонената и развој електронских пројеката. Овај алат је посебно популаран међу особама које се баве хоби електроником, едукацијом или прототипирањем. Fritzing користи интуитиван и визуелан приступ за цртање електронских шема. Корисници могу да повлаче и спуштају компоненте из библиотеке на радну површину и повезују их жицама како би визуелно представили електричне везе. Fritzing долази са широком палетом електронских компонената, укључујући микроконтролере, сензоре, LED диоде, отпорнике, кондензаторе и многе друге. Ова библиотека омогућава корисницима да брзо и једноставно додају компоненте у своје пројекте. Fritzing има подршку за Arduino микроконтролере, што га чини идеалним алатом за Arduino пројектанте. Корисници могу лако повући Arduino микроконтролер на радну површину и повезати га са осталим компонентама. Fritzing омогућава корисницима да креирају детаљне шеме својих 24 пројеката, што олакшава документацију и дијалог са другима у заједници. Пројекти се могу сачувати, делити или штампати како би се олакшало прототипирање и израда електронских уређаја. Fritzing је посебно користан за почетнике у електроници, јер пружа једноставан и приступачан начин за разумевање основа електричних веза и електронике. Fritzing је open-source софтвер, што значи да је доступан бесплатно за употребу и да је заједница допринела његовом развоју и унапређењу током година. Fritzing је користан алат за све оне који желе да планирају, дизајнирају и документују своје електронске пројекте на лак и визуелно привлачен начин.



Слика 10. Fritzing шема повезивања

Приказан је пројекат паметног прекидача. Његов изглед и повезивање можете видети на слици 10. Повезивање је урађено на следећи начин:

- Релеj (VCC - 5V, GND – GND, IN – Пин 2)
 - IR приjемник (VCC – 3V3, GND – GND, DAT – Пин 8)

Након повезивања хардвера, потребно је написати софтвер за један овакав пројекат. Наша идеја је таква да ћемо користити даљински управљач umesto IR предајника. Овде нам се јавља проблем који обухвата чињеницу да морамо знати код тастера који долази са даљинског управљача. Ово ћемо урадити на тај начин што ћемо користити библиотеку "Irremote.h".

Софтверска реализација паметног прекидача (Arduino Uno)

Коришћење библиотеке Irremote

Библиотека IRremote.h је популарна Arduino библиотека која омогућава рад са инфрацрвеним (IR) даљинским управљачима. Ова библиотека олакшава декодирање IR

сигнала, као и слање IR команда, омогућавајући једноставну интеграцију са различитим уређајима који користе инфрацрвену технологију.

Кључне карактеристике IRremote.h библиотеке:

- Декодирање IR сигнала: Библиотека може декодирати сигнале са различитих врста даљинских управљача (нпр. NEC, Sony, RC5, RC6, итд.). Када се сигнал прими путем IR пријемника, библиотека га декодира и пружа приступ стварним подацима које је послao даљински управљач (нпр. код тастера).
- Слање IR сигнала: Осим декодирања, библиотека омогућава и слање IR сигнала путем IR LED-а. Ова функција је корисна ако желиш да твој Arduino пројекат емитује сигнале и контролише друге IR уређаје, као што су телевизори или други кућни апарати.
- Подршка за различите протоколе: Библиотека подржава широк спектар IR протокола који се користе у различитим уређајима, што значи да можеш користити исти Arduino код са различитим даљинским управљачима.
- Једноставна употреба: Функције које пружа IRremote.h библиотека су једноставне за коришћење и омогућавају брзо развијање прототипова.

Основне функције библиотеке: **IRrecv**: Класа која управља пријемом IR сигнала, **decode()**: Функција која декодира примљени сигнал и враћа структуру са подацима (нпр. врста протокола, код тастера), **enableIRIn()**: Омогућава пријем IR сигнала, **resume()**: Ресетује пријемник након обраде сигнала, омогућавајући пријем следећег сигнала, **IRsend**: Класа која управља слањем IR сигнала, **sendNEC()**, **sendSony()**, **sendRC5()**, итд.: Функције које шаљу IR сигнал у зависности од протокола који се користи.

```
//Kod je za verziju 2.6.0 biblioteke IRremote

#include <IRremote.h>

const int RECV_PIN = 8; // Pin na koji je povezan IR prijemnik
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup() {
    Serial.begin(9600);
    irrecv.enableIRIn(); // Počni prijem IR signala
}

void loop() {
    if (irrecv.decode(&results)) {
        Serial.print("IR Code: ");
        Serial.println(results.value, HEX); // Prikazuje IR kod u heksadecimalnom formatu
        irrecv.resume(); // Prima sledeći signal
}
```

```
    }  
}
```

Микроконтролер флешујемо кодом који видимо изнад. Овај код нам користи да видимо кодове које даљински управљач шаље када притиснемо жељене тастере. Користићемо тастер 1 (овај тастер служи да упалимо релеј) и тастер OFF (овај тастер служи за гашење релеја и враћање у првобитни положај). Након флешовања микроконтролера потребно је да принесемо даљински управљач у правцу IR пријемника и да притиснемо тастере 1 и OFF. На серијском монитору можемо видети жељене кодове (Тастер 1. IR Code: 4FB00FF, Тастер OFF. IR Code: 4FB4AB5). Кодови у загради ће нам послужити у даљем наставку за писање кода.

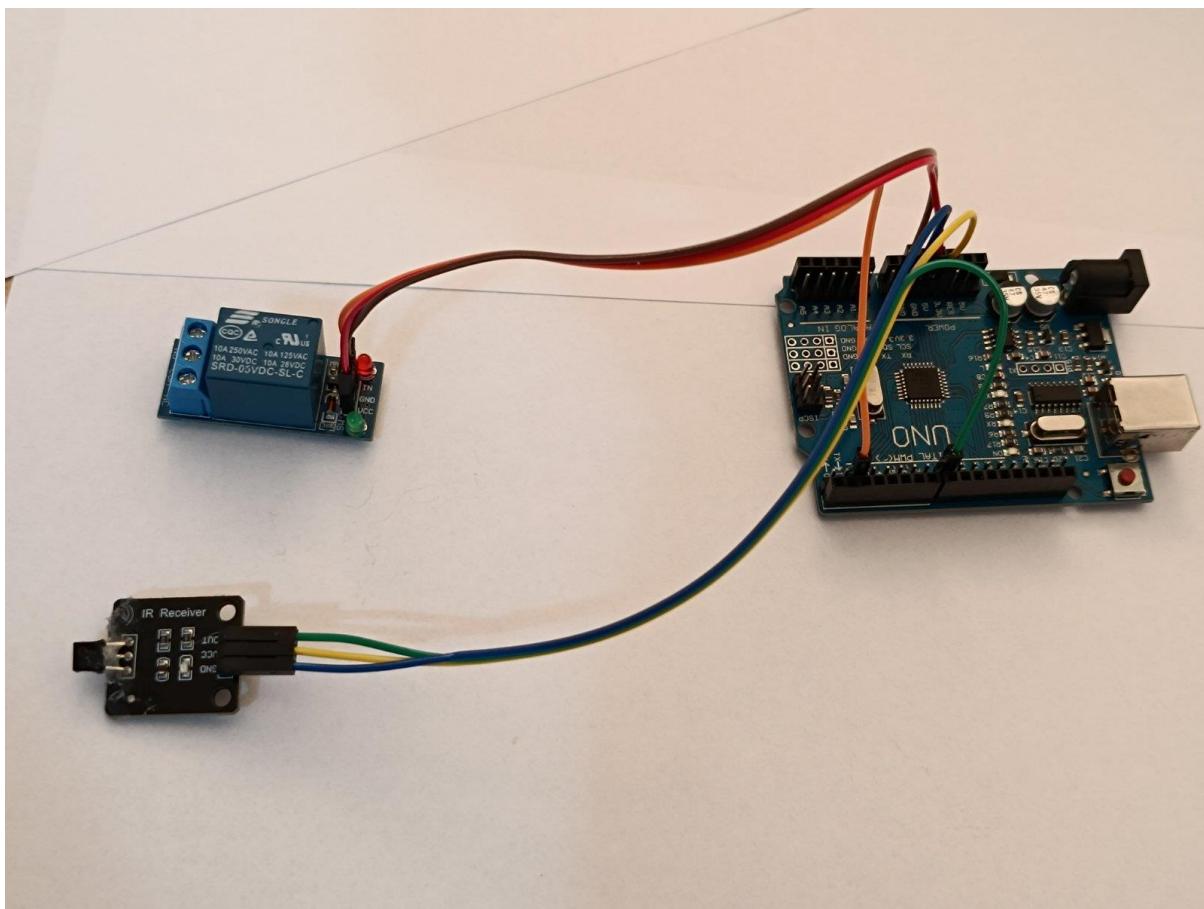
The screenshot shows the Arduino IDE interface. The code area contains the following sketch:

```
1 //Kod je za verziju 2.6.0 biblioteke IRremote  
2  
3 #include <IRremote.h>  
4  
5 const int RECV_PIN = 8;  
6 IRrecv irrecv(RECV_PIN);  
7 decode_results results;  
8  
9 void setup() {  
10   Serial.begin(9600);  
11   irrecv.enableIRIn();  
12 }  
13  
14 void loop() {  
15   if (irrecv.decode(&results)) {  
16     Serial.print("IR Code: ");  
17     Serial.println(results.value, HEX);  
18     irrecv.resume();  
19   }  
20 }  
21  
22
```

The Serial Monitor window shows the following output:

Time	Message
19:48:59.117	-> IR Code: 4FB00FF Taster 1
19:49:04.973	-> IR Code: 4FB4AB5 Taster OFF

Слика 11. Приказ кодова тастера даљинског управљача



Слика 12. Хардверско повезивање

Објашњење кода паметног прекидача

У овом програму потребно је написати код који ће детектовати долазак одређене вредности (Тастер 1. IR Code: 4FB00FF, Тастер OFF. IR Code: 4FB4AB5) и извршити одређене акције у зависности од те вредности. Односно, када се притисне тастер 1 долази до паљења релеја, када се притисне тастер OFF долази до гашења релеја. Релеј се укључује када се на пин VIN пошаље логичка нула (LOW), релеј се искључује када се на пин VIN пошаље логичка јединица (HIGH). Циљ је омогућити уређају да реагује на специфичне команде које долазе путем даљинског управљача.

- Прво, потребно је дефинисати пинове који ће се користити за пријем IR сигнала и управљање уређајем (релејем и IR пријемником). У нашем случају IR пријемник користи пин 8, релеј користи пин 2.
- У функцији `setup()`, потребно је поставити пинове као излазе и покренути IR пријемник.
- У функцији `loop()`, користи се условна структура која ће проверавати да ли је примљена одређена вредност. Ако се препознаје очекивана вредност, програм ће извршити одређену акцију.

Код који одговара захтевима пројекта:

```
#include <IRremote.h>

const int RECV_PIN = 8; // Pin na koji je povezan IR prijemnik
const int RELAY1_PIN = 2; // Pin na koji je povezan Relej

IRrecv irrecv(RECV_PIN);
decode_results results;

void setup() {
    Serial.begin(9600);
    irrecv.enableIRIn(); // Počni prijem IR signala

    // Postavi pin za relej kao izlazni
    pinMode(RELAY1_PIN, OUTPUT);

    // Osiguraj da je relej isključen na početku
    digitalWrite(RELAY1_PIN, HIGH);
}

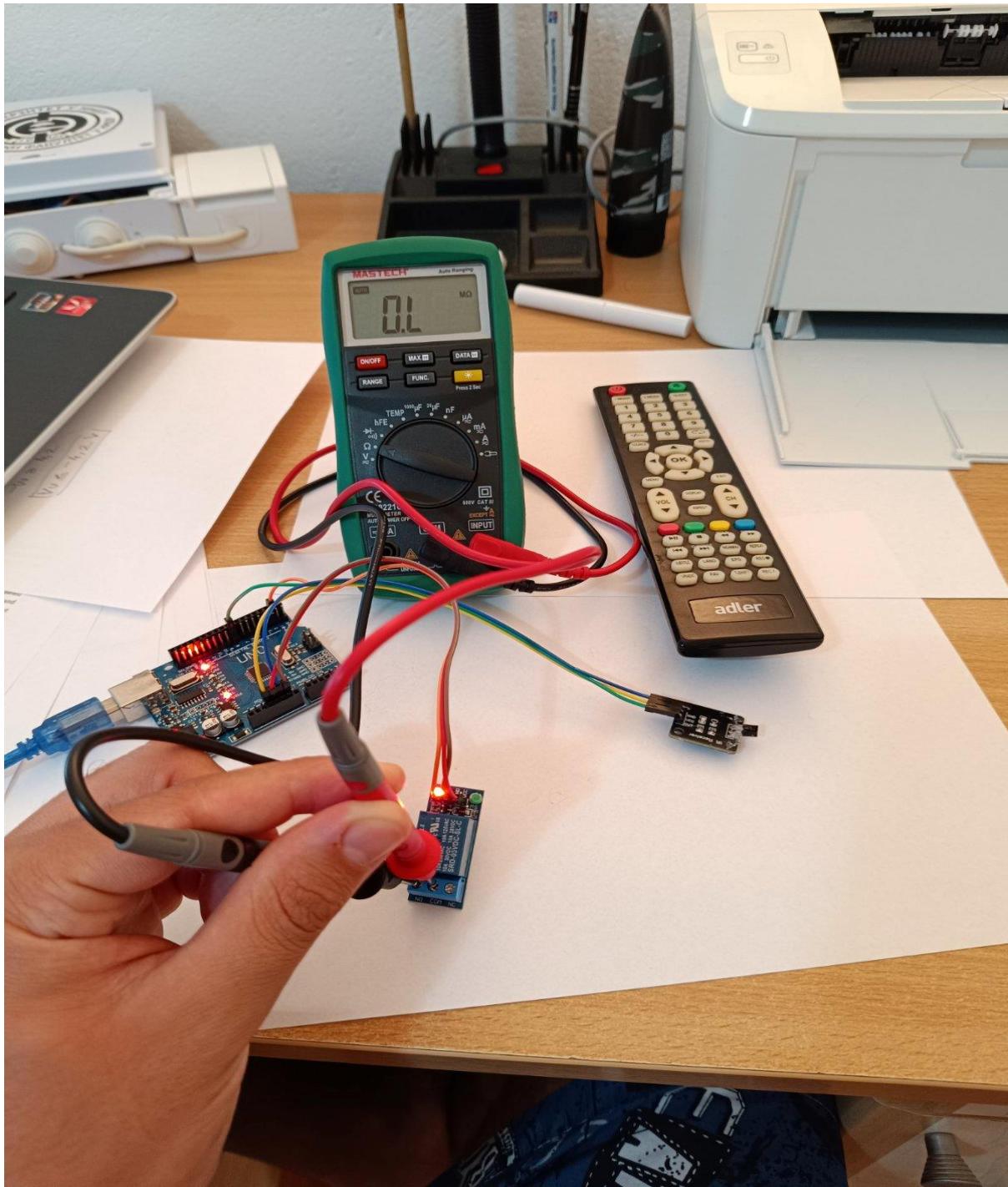
void loop() {
    if (irrecv.decode(&results)) {
        unsigned long irCode = results.value;
        Serial.print("IR Code: ");
        Serial.println(irCode, HEX); // Prikazuje IR kod u
        heksadecimnalnom formatu

        switch (irCode) {
            case 0x4FB0FF: // Kod za taster 1
                digitalWrite(RELAY1_PIN, LOW);
                break;
            case 0x4FB4AB5: // Kod za taster za isključivanje releja
                digitalWrite(RELAY1_PIN, HIGH);
                break;
            default:
                // Ako kod nije prepoznat, ne radi ništa
                break;
        }

        irrecv.resume(); // Prima sledeći signal
    }
}
```

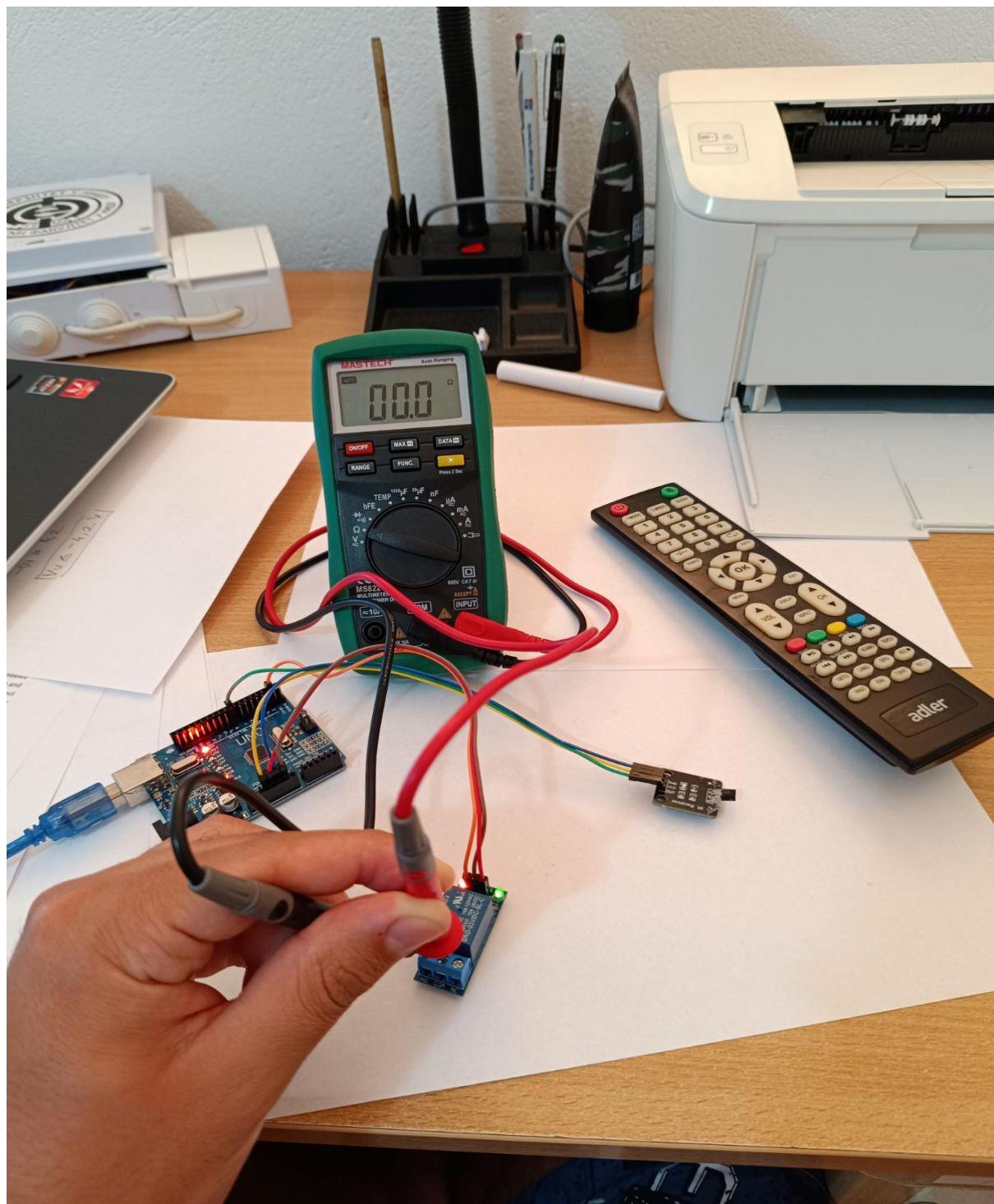
Демонстрација пројекта паметног прекидача (Arduino Uno)

На слици 13. можемо видети коначну верзију пројекта (хардвер и софтвер). Видимо да унимер показује OL (overflow) вредност између контаката СОМ и НО. То значи да имамо прекид између ова два контаката.



Слика 13. Приказ пројекта када је релеј угашен

Са друге стране на слици 14. можемо видети да је вредност на унимеру 0 Ω. То указује да имамо кратак спој. Односно, да је дошло до затварања прекидача. На самом релеју можемо видети да поред црвене LED диоде светли и зелена LED диода, што нам указује да је релеј извршио жељену активност.

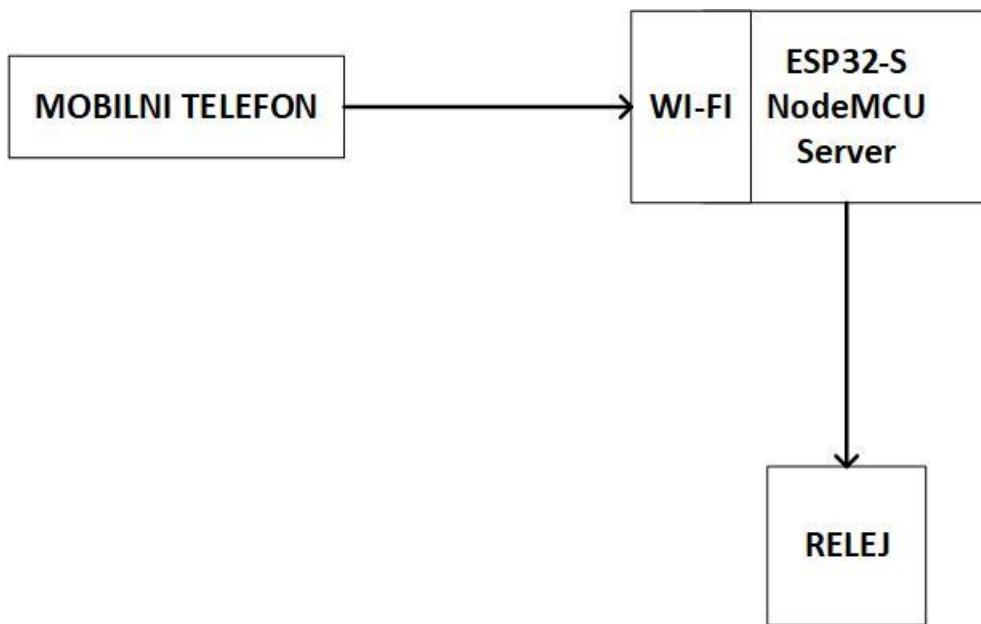


Слика 14. Приказ пројекта када је релеј укључен (упаљена зелена LED диода)

Реализација паметног прекидача (ESP32-S NodeMCU)

У првој фази овог истраживања, развијен је паметни прекидач заснован на Arduino Uno платформи, користећи IR пријемник и даљински управљач за контролу релеја. Овај систем омогућава локално управљање уређајима помоћу инфрацрвених сигнала, пружајући једноставно, али ефикасно решење за контролу електричних уређаја. Међутим, са развојем технологије и све већом потребом за удаљеним управљањем, показала се потреба за системом који омогућава контролу уређаја путем Интернета. Ова фаза истраживања фокусира се на имплементацију таквог система користећи ESP32-S NodeMCU микроконтролер, који ће функционисати као сервер за контролу релеја путем Wi-Fi мреже.

ESP32-S NodeMCU је моћна IoT платформа која интегрише Wi-Fi и Bluetooth функционалности, омогућавајући корисницима да повежу своје уређаје на Интернет и управљају њима са било које локације путем мобилних уређаја. Циљ ове фазе је развити систем који омогућава корисницима да једноставно контролишу релеј путем web интерфејса на мобилном телефону, чиме се додатно унапређује функционалност паметног прекидача из прве фазе. Овај систем не само да омогућава бежичну контролу, већ пружа и основу за даље унапређење у оквиру IoT екосистема, омогућавајући потенцијалну интеграцију са паметним кућним системима и другим IoT уређајима. Блоковску шему израде оваквог једног система можемо видети на слици испод.



- Мобилни телефон, у овој шеми, се директно повезује на ESP32-S NodeMCU који функционише као Access Point (AP). Корисник се повезује на Wi-Fi мрежу коју генерише ESP32, а затим користи web претраживач на телефону да приступи web серверу хостованом на ESP32. Кроз овај web

интерфејс, корисник може слати команде за укључивање или искључивање релеја.

- ESP32-S NodeMCU Сервер, ESP32-S NodeMCU у овом случају функционише као Access Point који омогућава мобилном телефону да се директно повеже на њега. ESP32-S NodeMCU подиже web сервер на којем је хостован кориснички интерфејс за контролу релеја. Када корисник са мобилног телефона пошаље команду преко web интерфејса, ESP32 обрађује захтев и контролише релеј на основу добијених инструкција.
- Релеј је повезан са ESP32-S NodeMCU и користи се за управљање електричним уређајима. Када ESP32-S NodeMCU прими команду путем web сервера, он контролише релеј, укључујући или искључујући повезани уређај. Релеј осигурува сигурно пребацивање високог напона, омогућавајући контролу различитих уређаја.

ESP32-S NodeMCU

Ово је NodeMCU развојна плоча заснована на ESP32, која садржи Wi-Fi и Bluetooth повезивање, уgraђени CP2102 чип и тастере. Поред тога, сви I/O пинови ESP32 модула су доступни преко екstenзионих хедера. Захваљујући богатим ресурсима отвореног кода, подржава развој на различите начине, као што су Lua/AT команде/MicroPython/Arduino/IoT изворни код, што вам помаже да брзо направите IoT апликације. Језгро овог модула је ESP32 чип, који је скалабилан и адаптиван. Два CPU језгра могу се контролисати индивидуално. Фреквенција такта је подесива од 80 MHz до 240 MHz и подржава RTOS. Ово је вишенаменски Wi-Fi+BT+BLE MCU модул ESP-WROOM-32s.

Модул интегрише традиционални Bluetooth, Bluetooth ниске потрошње (BLE) и Wi-Fi. Широка примена: Wi-Fi подржава широк спектар комуникационих веза, као и директну конекцију на интернет путем рутера; Bluetooth омогућава корисницима да се повежу на мобилни телефон или еmitују BLE Beacon за детекцију сигнала. Модул подржава брзине преноса података до 150 Mbps и излазну снагу антене од 20 dBm за максималну бежичну комуникацију. Као резултат тога, овај модул има водеће индустријске спецификације и добро функционише у погледу високе интеграције, удаљености бежичног преноса, потрошње енергије и повезивања на мрежу.

WIFI модул: ESP32

Процесор: ESP32

Угађени Flash: 32Mbit

Антена: На плочи PCB антена

Периферијски интерфејси: UART/GPIO/ADC/DAC/SDIO/PWM/I2C/I2S

WiFi протокол: IEEE 802.11 b/g/n

Bluetooth: Bluetooth 4.2

Фреквенцијски опсег: 2.4G ~ 2.5G (2400M ~ 2483.5M)

WIFI мод: Station / SoftAP / SoftAP+Station

Напајање: 5V

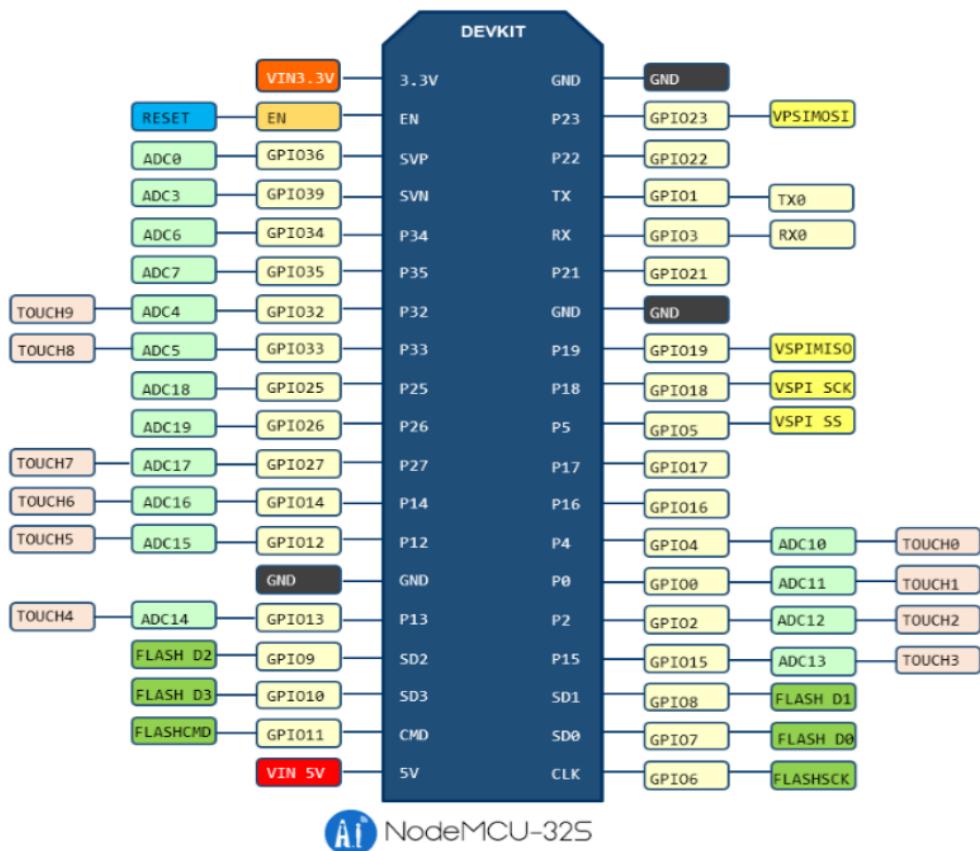
Вредност дигиталне јединице: 3.3V

Димензије: 51.4mm x 25.4mm



Слика 16. ESP32-S NodeMCU

ESP-WROOM-32s модул има укупно 38 интерфејса, како је приказано на слици 17 , док је у табели приказана дефиниција интерфејса.



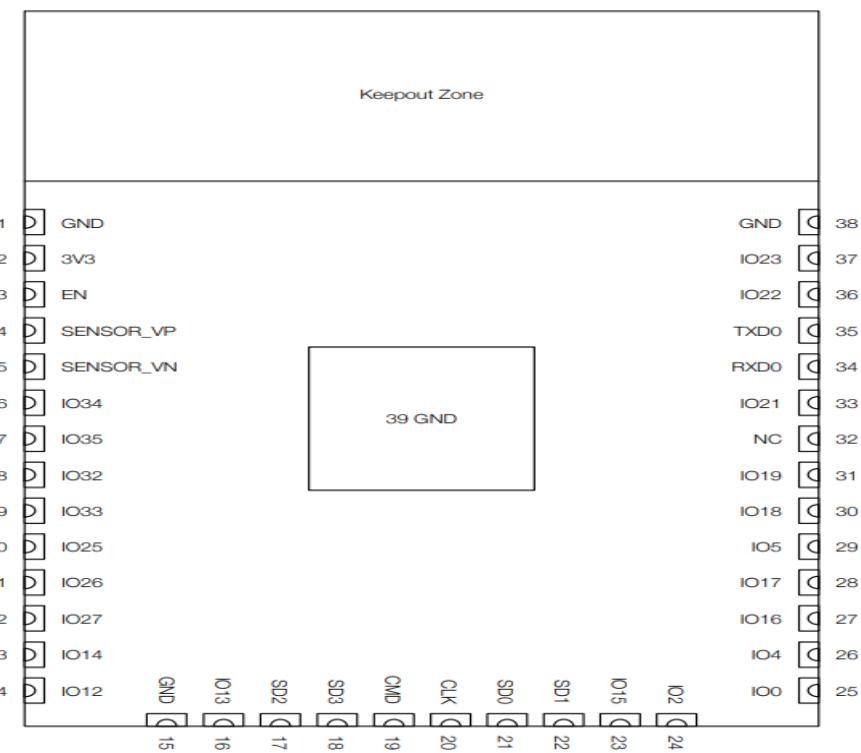
Слика 17. ESP-WROOM-32s пин дијаграм

ESP-WROOM-32s

ESP32-WROOM-32 је снажан, универзални Wi-Fi + Bluetooth + Bluetooth LE MCU модул који је намењен широком спектру апликација, од мрежа сензора са ниском потрошњом енергије до најзахтевнијих задатака, као што су кодирање гласа, стримовање музике и декодирање MP3 формата.

У срцу овог модула налази се ESP32-D0WDQ6 чип. Овај уgraђени чип је дизајниран да буде скалабилан и адаптиван. Има два CPU језгра која се могу контролисати појединачно, а фреквенција такта CPU-а је подесива од 80 MHz до 240 MHz. Чип такође има копроцесор са ниском потрошњом енергије који може да се користи уместо CPU-а како би се уштедела енергија приликом извршавања задатака који не захтевају велику рачунарску снагу, као што је надгледање периферија. ESP32 интегрише богат скуп периферних уређаја, укључујући сензоре за капацитивни додир, интерфејс за SD картицу, Ethernet, брзи SPI, UART, I2S и I2C.

Интеграција Bluetooth-a, Bluetooth LE и Wi-Fi омогућава циљање широког спектра апликација и чини модул свестраним: коришћење Wi-Fi-а омогућава велику физичку удаљеност и директну везу са интернетом путем Wi-Fi рутера, док коришћење Bluetooth-а омогућава кориснику да се лако повеже са телефоном или еmitује нисконапонске beacon-е за њихово откривање. Струја у стању мirovanja ESP32 чипа је мања од 5 μ A, што га чини погодним за апликације са батеријским напајањем и носиву електронику. Модул подржава брзину преноса података до 150 Mbps и излазну снагу од 20 dBm на антени како би се осигурала највећа физичка удаљеност. Као такав, модул нуди индустријски водеће спецификације и најбоље перформансе за електронску интеграцију, домет, потрошњу енергије и повезивост. Оперативни систем изабран за ESP32 је freeRTOS са LwIP; TLS 1.2 са хардверском акцелерацијом је такође уgraђен.



Слика 18. Распоред пинова чипа

корисници могу ажурирати своје пројекте чак и након њиховог пуштања, уз минималне трошкове и труд. Такође је подржано сигурно (криптовано) бежично (OTA) ажурирање, тако да корисници могу ажурирати своје производе чак и након њиховог пуштања, уз минималне трошкове и труд.

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER

IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Слика 19. Опис пинова чипа

Пинови SCK/CLK, SDO/SD0, SDI/SD1, SXD/SD2, SWP/SD3 и SCS/CMD, односно GPIO6 до GPIO11, су повезани са интегрисаним SPI флешом на модулу и није их препоручљиво користити за друге намене.

Пинови за strapping

Пинови за strapping су специјални пинови на чипу ESP32 (и другим микроконтролерима) који се користе за одређивање одређених конфигурација или подешавања приликом покретања (бутовања) уређаја. Ови пинови се обично користе за :

- Одређивање режима покретања: Пинови за strapping омогућавају корисницима да изаберу начин на који ће чип бити покренут, као што су режими за покретање из флеша, серијске меморије или других извора.
- Поставке напајања: Могу се користити за конфигурацију радног напона или других подешавања система.
- Држање стања приликом ресетовања: Када се чип ресетује (на пример, приликом укључивања напајања или због watchdog ресетовања), пинови за strapping узимају узорак нивоа напона и задржавају то стање док се чип не искључи.

Ови пинови су обично повезани са унутрашњим pull-up или pull-down отпорницима, што значи да ће њихов ниво напона бити подразумевано одређен на основу повезаних спољних компоненти или отпорника, ако нису директно повезани.

ESP32 има пет пинова за strapping:

- MTDI
- GPIO0
- GPIO2
- MTDO

- GPIO5

Софтвер може да прочита вредности ових пет бита из регистра "GPIO_STRAPPING". Током ослобађања ресетовања система чипа (ресетовање приликом укључивања, RTC watchdog ресет и brownout ресет), latch-еви пинова за strapping узимају узорак нивоа напона као strapping битове "0" или "1" и задржавају те битове док се чип не искључи. Strapping битови конфигуришу режим покретања уређаја, радни напон VDD_SDIO и друга почетна подешавања система. Сваки пин за strapping је повезан са својим интерним pull-up/pull-down отпорницима током ресетовања чипа. Сходно томе, ако је пин за strapping неповезан или је повезан у спољашње коло које има висок отпор (то значи да не може усмерити лако ниво напона на 1 или 0), унутрашњи pull-up/pull-down ће одредити подразумевани улазни ниво пинова за strapping. Да би променили вредности strapping битова, корисници могу применити pull-up/pull-down отпорнике или користити GPIO-ове пинове MCU-а за контролу нивоа напона ових пинова приликом укључивања ESP32. Након ослобађања ресетовања, пинови за strapping функционишу као нормални функционални пинови.

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V	1.8 V		
MTDI	Pull-down	0	1		
Boot Mode					
Pin	Default	SPI Boot	Download Boot		
GPIO0	Pull-up	1	0		
GPIO2	Pull-down	Don't-care	0		
Enabling/Disabling Debugging Log Print over U0TXD During Booting					
Pin	Default	U0TXD Active	U0TXD Silent		
MTDO	Pull-up	1	0		
Timing of SDIO Slave					
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Слика 20. Strapping пинови

CPU и унутрашња меморија

ESO32-D0WDQ6 садржи два микропроцесора ниске потрошње Xtensa 32-бит LX6. Унутрашња меморија укључује:

- 448 KB ROM-а за покретање и основне функције.
- 520 KB SRAM-а на чипу за податке и инструкције.
- 8 KB SRAM-а у RTC, који се назива RTC FAST меморија и може се користити за складиштење података; приступа му главни CPU током RTC покретања из режима дубоког спавања.
- 8 KB SRAM-а у RTC, који се назива RTC SLOW меморија и може му приступити процесор током режима дубоког спавања.

- 1 Kbit eFyse: 256 бита се користи за систем (MAC адреса и конфигурација чипа), а преосталих 768 бита је резервисано за корисничке апликације, укључујући енкрипцију флеша и ID чипа.

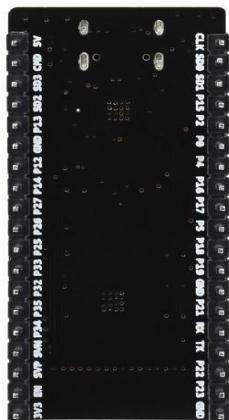
Спољашњи флаши и SRAM

ESP32 подржава више екстерних QSPI флеша и SRAM чипова. ESP32 може приступити екстерном QSPI флешу и SRAM-у путем брзих кешева. Екстерни флеш може бити мапиран у меморијски простор CPU инструкција и простор меморије само за читање симултано. Када је екстерни флеш мапиран у меморијски простор CPU инструкција, може се мапирати до 11 MB + 248 KB у исто време. Имајте на уму да ако се мапира више од 3 MB + 248 KB, перформансе кеша ће се смањити због спекулативног читања од стране CPU-а. Када је екстерни флеш мапиран у простор меморије података само за читање, може се мапирати до 4 MB у исто време. Подржана су читања од 8 бита, 16 бита и 32 бита. Екстерни SRAM може бити мапиран у простор меморије података CPU-а. Може се мапирати до 4 MB у исто време. Подржана су читања и писања од 8 бита, 16 бита и 32 бита.

ESP32-WROOM-32 интегрише 4 MB SPI флеша, који је повезан на GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 и GPIO11. Ових шест пинова не може се користити као редовни GPIO пинови. Модул користи 40-MHz кристални осцилатор.

Распоред физичких пинова плочице ESP32-S NodeMCU

Физички пинови на ESP32-S NodeMCU модулу су распоређени тако да омогућавају једноставно повезивање и коришћење различитих функционалности. Приликом спреге између плочице и релеја, нама није од интереса велики број пинова. Користићемо само пин за контролу релеју P16. У даљем наставку видећемо све физичке пинове на плочици са циљем да се што боље упознамо са плочицом .



Слика 21. Физичко нумерисање пинова

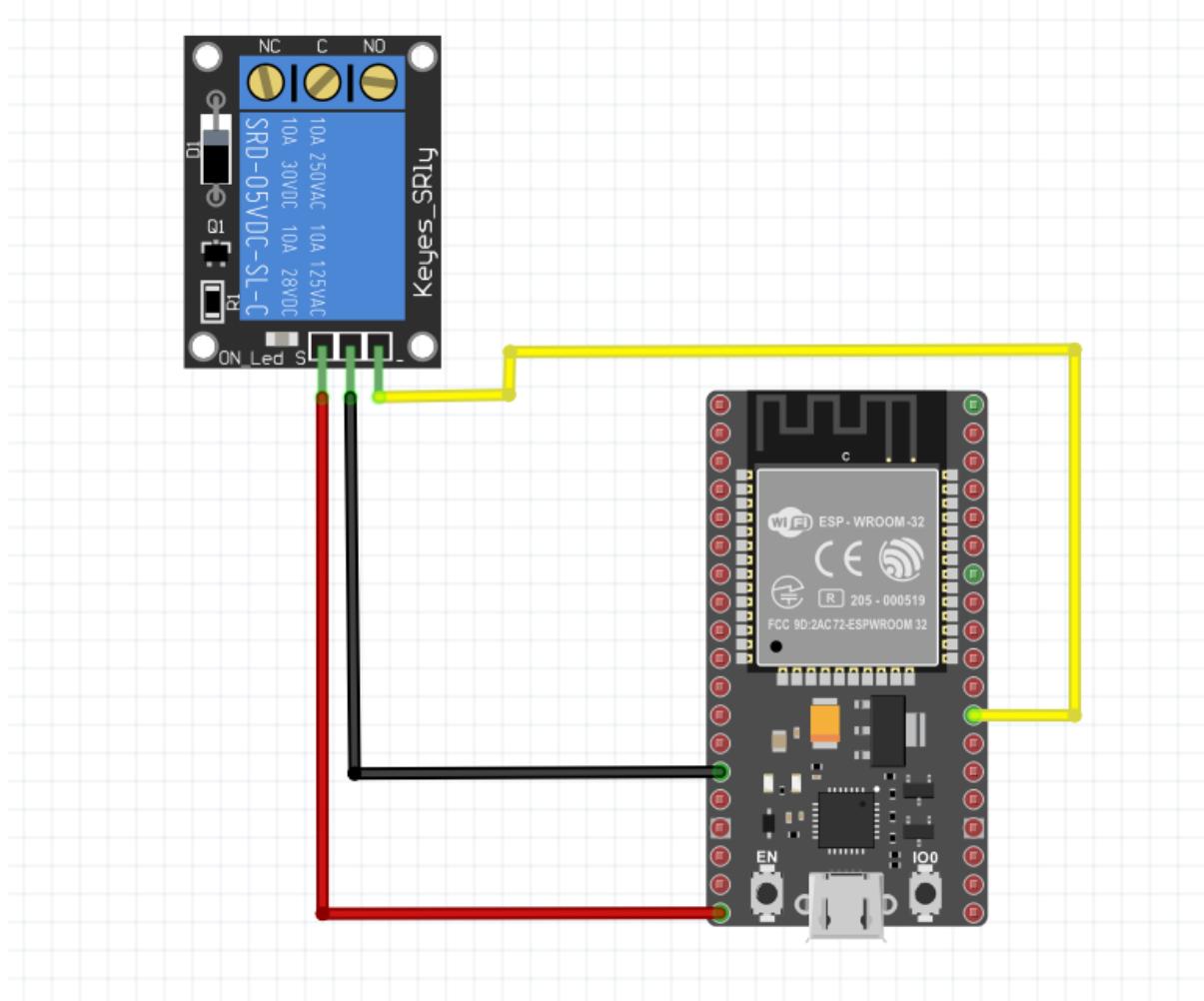
No.	Pin Name	Functional Description
1	3.3V	Module power supply pin
2	EN	Chip Enabled Pin, Active High
3	SVP	GPIO36, ADC1_CH0, RTC_GPIO0
4	SVN	GPIO39, ADC1_CH3, RTC_GPIO3
5	P34	GPIO34, ADC1_CH6, RTC_GPIO4
6	P35	GPIO35, ADC1_CH7, RTC_GPIO5
7	P32	GPIO32, XTAL_32K_P (32.768kHz Crystal input), ADC1_CH4, TOUCH9,RTC_GPIO9
8	P33	GPIO33, XTAL_32K_N (32.768kHz Crystal output), ADC1_CH5, TOUCH8,RTC_GPIO8
9	P25	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
10	P26	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RX_DV
11	P27	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
12	P14	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
13	P12	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
14	GND	GND
15	P13	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPIID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
16	SD2	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
17	SD3	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
18	CMD	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
19	5V	Module power supply pin
20	CLK	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
21	SD0	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
22	SD1	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
23	P15	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICSO, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
24	P2	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
25	P0	GPIO0, ADC2_CH1, TOUCH1, CLK_OUT1, RTC_GPIO11, EMAC_TX_CLK; Download mode:

Слика 22. Дефинисање пинова

		external pull low, running mode: floating or external pull high
26	P4	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
27	P16	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
28	P17	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
29	P5	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
30	P18	GPIO18, VSPICLK, HS1_DATA7
31	P19	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
32	GND	GND
33	P21	GPIO21, VSPIHD, EMAC_TX_EN
34	RX	GPIO3, U0RXD, CLK_OUT2
35	TX	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
36	P22	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
37	P23	GPIO23, VSPIID, HS1_STROBE
38	GND	GND

Слика 23. Дефинисање пинова (наставак)

Практична реализација паметног прекидача (ESP32-S NodeMCU)



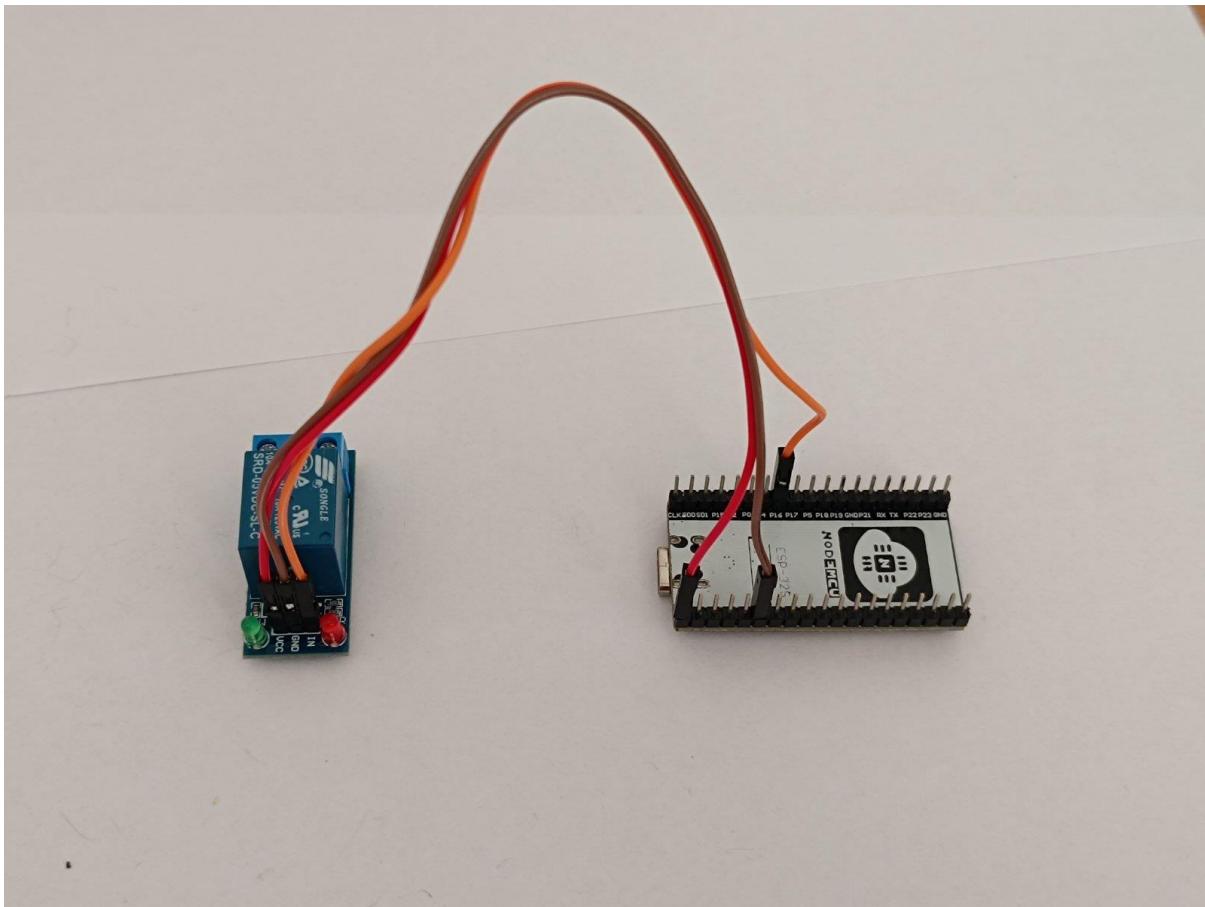
Слика 24. Fritzing повезивање

На слици 24. је приказан спој између ESP32 развојне плочице и релејног модула. Спој је направљен помоћу жица које повезују пинове на ESP32 и релеја.

- Напајање (VCC): Релеј је напајан са 5V пин-а на ESP32 плочици (црвена жица).
- Уземљење (GND): GND пин на релеју је повезан са GND пином на ESP32 (црна жица).
- Сигнал (IN): Сигнални пин на релеју (IN) је повезан с пином GPIO16 на ESP32 (жута жица).

Овај спој омогућава управљање релејом путем ESP32 плочице. Када се на GPIO16 пин пошаље HIGH или LOW сигнал, релеј ће се искључити или укључити, омогућавајући или прекидајући проток струје кроз релејни модул. Ово подешавање се често користи у пројектима паметне аутоматизације, где се релеј може контролисати даљински путем Wi-Fi мреже. У овом случају, ESP32 може служити као сервер који омогућава кориснику да преко web интерфејса контролише релеј, а самим тим и уређај

који је повезан на релеј (нпр. светло или мотор). Можемо уочити да хардверски део је врло једноставан, суштина пројекта лежи у софтверском делу. О коме ће касније бити речи.



Слика 25. Хардверско повезивање

Софтверска реализација паметног прекидача (ESP32-S NodeMCU)

Коришћење библиотеке WiFi

Библиотека WiFi.h је један од кључних алата за ESP32 и ESP8266 микроконтролере када је упитању повезивање уређаја на бежичне мреже. Ова библиотека поједностављује процес управљања Wi-Fi функционалношћу, омогућавајући корисницима да креирају сложене мрежне апликације без потребе за детаљним знањем о ниском нивоу Wi-Fi протокола. Библиотека WiFi.h омогућава вашем ESP32 (или ESP8266) уређају да се повеже на бежичну мрежу (Wi-Fi) и постане део локалне мреже или Интернета. Ова могућност је кључна за већину IoT (Internet of Things) пројектата, где је повезаност са Интернетом централни аспект. Са WiFi.h, можете

преузети податке са сервера, слати податке на сервер, контролисати уређаје путем мреже и много више. Главне карактеристике ове библиотеке су:

- Повезивање на Wi-Fi мрежу је једна од основних функција WiFi.h библиотеке је могућност једноставног повезивања ESP32 уређаја на Wi-Fi мрежу. Помоћу функције WiFi.begin(ssid, password);, можете покренути процедуру повезивања користећи назив мреже (SSID) и лозинку. Библиотека аутоматски управља сложеностима повезивања, као што су аутентификација, добављање IP адресе, и одржавање везе.
- WiFi.h омогућава ESP32 уређају да ради у неколико модова: **Station Mode (STA)**: У овом моду, уређај се понаша као клијент који се повезује на постојећу Wi-Fi мрежу. То је уобичајени мод када желите да повежете ESP32 на кућну или канцеларијску мрежу, **Access Point Mode (AP)**: У овом моду, ESP32 постаје приступна тачка, омогућавајући другим уређајима да се повежу на њега директно. Ова функција је корисна када желите да креирате локалну мрежу без рутера. Нама је од интереса други мод рада, када је ESP32 приступна тачка. **Station + Access Point Mode**: ESP32 може истовремено бити повезан на постојећу мрежу и функционисати као приступна тачка, омогућавајући максималну флексибилност.
- Управљање IP адресама, WiFi.h омогућава једноставно управљање IP адресама. Можете одабрати да ваш уређај добија IP адресу динамички (DHCP) или да му ручно доделите статичку IP адресу. Функција WiFi.config(local_ip, gateway, subnet); омогућава вам да прецизно подесите мрежне параметре.
- Сигурност је критични аспект у мрежним апликацијама. WiFi.h подржава различите методе аутентификације, укључујући WPA/WPA2, што осигурува безбедност ваше мреже. ESP32 такође може користити енкрипцију података како би се осигурала приватност комуникација.
- Wi-Fi скенер, једна од корисних функционалности WiFi.h библиотеке је способност скенирања околине за доступне Wi-Fi мреже. Функција WiFi.scanNetworks(); вам омогућава да претражите и прикажете све доступне мреже, што је корисно када желите да изаберете најбољу мрежу за повезивање на основу јачине сигнала или других фактора.
- Праћење статуса, кроз функцију WiFi.status();, можете лако пратити статус Wi-Fi везе. Ово је корисно за дијагностику и обезбеђивање да је уређај увек повезан. Ако веза није успостављена, можете имплементирати аутоматске процедуре за поновно повезивање.
- Једноставна употреба, библиотека је дизајнирана тако да буде једноставна за употребу, чак и за почетнике. Већина функција је високо апстрахована, што значи да се већина комплексних мрежних операција обавља "испод хаубе". Као резултат тога, програмери се могу фокусирати на логику апликације, док WiFi.h рукује техничким детаљима мрежне комуникације.

Коришћење библиотеке WebServer

Библиотека WebServer.h је кључна за имплементацију web сервера на ESP32 или ESP8266 микроконтролерима. Ова библиотека омогућава вашем уређају да служи као web сервер, прихватајући HTTP захтеве од клијената (као што су web претраживачи) и одговарајући на њих. Ово је основа за многе IoT пројекте где је потребно омогућити контролу уређаја или приступ подацима преко web интерфејса. Главне карактеристике су:

- Једноставна имплементација Web сервера, библиотека WebServer.h поједностављује процес стварања потпуно функционалног web сервера на вашем ESP32 уређају. Можете једноставно поставити руте (енг. routes) које дефинишу како сервер реагује на различите URL-ове или HTTP методе, као што су GET и POST захтеви. На пример, са функцијом server.on("/", HTTP_GET, handleRoot);, можете одредити да ваш сервер позове функцију handleRoot сваки пут када корисник приступи главној страници ("/").
- Обрада HTTP захтева, WebServer.h омогућава вашем ESP32 уређају да обради различите типове HTTP захтева. Можете лако поставити сервер да одговара на GET, POST, PUT, DELETE и друге типове HTTP метода. Ово је корисно када желите да креирате апликацију која омогућава корисницима да не само добијају информације, већ и шаљу податке серверу. GET захтеви, уобичајени захтеви за добијање информација са сервера. POST захтеви: користе се за слање података серверу, као што су форме, параметри, или подешавања.
- Динамично генерирање садржаја, помоћу WebServer.h библиотеке, ваш сервер може динамички генерирати HTML садржај, омогућавајући приказивање актуелних података, као што су сензорске вредности или статуси уређаја. Функција server.send(200, "text/html", "Hello, world!"); омогућава вам да сервер одговори са прилагођеним HTML садржајем на HTTP захтеве.
- Управљање параметрима, WebServer.h пружа једноставне методе за обраду параметара који се шаљу путем URL-а или HTTP POST захтева. На пример, ако корисник пошаље захтев са параметрима у URL-у, као што је http://your-esp32.local/led?state=on, можете лако добити вредност state параметра и прилагодити понашање уређаја у складу с тим. server.arg("name"); – Омогућава дохватање вредности параметра из URL-а или POST захтева.
- Статички фајлови, ова библиотека такође омогућава сервирање статичких фај洛va, као што су слике, CSS, или JavaScript фајлови. Ово је корисно за креирање богатијих web интерфејса на вашем ESP32 серверу.
- Аутентификација и сигурност, WebServer.h подржава основне сигурносне функције, укључујући једноставну HTTP аутентификацију. Ово вам омогућава да заштитите приступ одређеним деловима вашег web сервера,

обезбеђујући да само овлашћени корисници могу да приступе критичним функцијама или информацијама.

- Једноставна употреба и флексибилност, једна од главних предности WebServer.h библиотеке је њена једноставност и флексибилност. Постављање сервера, дефинисање рута и обрада захтева су директни и лако разумљиви, чак и за кориснике који су нови у мрежном програмирању. API библиотеке је дизајниран тако да омогући брзо постављање сервера и интуитивну контролу над њим.

Објашњење кода паметног прекидача (ESP32-S NodeMCU)

У наставку видећемо како написати код за један овакав пројекат, научићемо како да напишемо код који омогућава ESP32 уређају да функционише као web сервер за даљинско управљање релејем путем Wi-Fi мреже. Овај пројекат је одличан пример коришћења ESP32 за стварање паметних кућних уређаја, као што су паметни прекидачи, које можете контролисати путем web интерфејса са свог мобилног телефона или рачунара. Пре него што кренемо са писањем кода, уверимо се да имамо инсталiran Arduino IDE са додатком за ESP32. На почетку кода, потребно је укључити две кључне библиотеке: WiFi (ова библиотека омогућава ESP32 да ради као Access Point), WebServer (ова библиотека омогућава креирање једноставног web сервера на ESP32 који може обрадити HTTP захтеве и генерисати web страницу).

Након укључивања библиотека, следећи корак је конфигурација Wi-Fi мреже. У овом пројекту, ESP32 ће бити постављен као Access Point, што значи да ће емитовати сопствену Wi-Fi мрежу на коју се можете повезати, тако да је потребно подесити SSID и PASSWORD мреже (const char* ssid = "SMART-switch"; ,const char* password = "12345678"). Како би ESP32 могао да хостује web сервер, потребно је дефинисати статичку IP адресу, мрежну маску и gateway. Ово осигурава да ваш уређај увек има исту IP адресу, што олакшава приступ web страници (IPAddress local_IP(192, 168, 4, 1); IPAddress gateway(192, 168, 4, 1); IPAddress subnet(255, 255, 255, 0);). Web сервер на ESP32 је централни део овог пројекта. Он омогућава корисницима да приступе web страници путем претраживача и укључе или искључе релеј. Пин који смо изабрали за релеј је P16.

```
const int RELAY_PIN = 16;
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, HIGH); // Postavljamo da je relej
iskljen na pocetku
```

Затим, постављамо руте за web сервер. На пример, рута "/" ће приказати основну web страницу, док ће "/upaliprekidac" и "/ugasiprekidac" укључивати и искључивати релеј.

```
WebServer server(80);

server.on("/", handleRoot);
server.on("/upaliprekidac", handleRelayOn);
server.on("/ugasiprekidac", handleRelayOff);
```

Функције handleRelayOn() и handleRelayOff() ће обрадити захтеве за укључивање и искључивање релеја. Када корисник кликне на дугме на web страници, ове функције ће бити позване и релеј ће се укључити или искључити.

```
void handleRelayOn() {  
    digitalWrite(RELAY_PIN, LOW); // Uključi relej  
    server.send(200, "text/plain", "POTROSAC AKTIVAN");  
}  
  
void handleRelayOff() {  
    digitalWrite(RELAY_PIN, HIGH); // Isključi relej  
    server.send(200, "text/plain", "POTROSAC NEAKTIVAN");  
}
```

На крају, потребно је покренути web сервер у функцији setup() и континуирано обрадити захтеве корисника у loop() функцији.

```
server.begin();  
void loop() {  
    server.handleClient(); // Obradi HTTP zahteve  
}
```

Када све ове кораке спојимо у целини и притом допунимо пројекат са html кодом који има за циљ улепшавање дизајна web стране добијамо у целости софтверску реализацију паметног прекидача коришћењем плочице ESP32-S NodeMCU. Плочица на себи поседује IOO дугме, које је потребно држати приликом флешовању микроконтролера на тај начин плочица улази у мод за програмирање. Након што кликнемо на дугме UPLOAD у ArduinoIDE, задржаћемо дугме и тамо флешовати микроконтролер.

```
#include <WiFi.h>  
#include <WebServer.h>  
  
const char* ssid = "SMART-switch"; // Ime Access Pointa  
const char* password = "12345678"; // Lozinka za Access Point  
  
const int RELAY_PIN = 16; // Pin na koji je povezan relej  
  
WebServer server(80);  
  
// Definiši statičku IP adresu, mrežnu masku i gateway  
IPAddress local_IP(192, 168, 4, 1);  
IPAddress gateway(192, 168, 4, 1);  
IPAddress subnet(255, 255, 255, 0);  
  
void handleRoot() {  
    String html = "<html>\n<head>\n    <title>ESP32 Relay Control</title>\n    <style>\n
```

```

        body { font-family: Arial, sans-serif; text-align: center; padding: 50px; } \
        h1 { color: #333; } \
        .button { display: inline-block; padding: 15px 25px; font-size: 24px; cursor: pointer; text-align: center; text-decoration: none; outline: none; color: #fff; background-color: #4CAF50; border: none; border-radius: 15px; box-shadow: 0 9px 999; } \
        .button:hover { background-color: #3e8e41 } \
        .button:active { background-color: #3e8e41; box-shadow: 0 5px 6px #666; transform: translateY(4px); } \
        .button-off { background-color: #f44336; } \
        .button-off:hover { background-color: #da190b } \
        .button-off:active { background-color: #da190b; box-shadow: 0 5px 6px #666; transform: translateY(4px); } \
    </style> \
</head> \
<body> \
    <h1>Upravljanje prekidacem!</h1> \
    <p><a href="/upaliprekidac" class="button">UPALI PREKIDAC</a></p> \
    <p><a href="/ugasiprekidac" class="button button-off">UGASI PREKIDAC</a></p> \
</body> \
</html>;

    server.send(200, "text/html", html);
}

void handleRelayOn() {
    digitalWrite(RELAY_PIN, LOW); // Uključi relej
    server.send(200, "text/plain", "POTROSAC AKTIVAN");
}

void handleRelayOff() {
    digitalWrite(RELAY_PIN, HIGH); // Isključi relej
    server.send(200, "text/plain", "POTROSAC NEAKTIVAN");
}

void setup() {
    Serial.begin(9600);
    // Postavi pin za relej kao izlaz
    pinMode(RELAY_PIN, OUTPUT);

    // Osiguraj da je relej isključen na početku
    digitalWrite(RELAY_PIN, HIGH);

    // Postavi statičku IP adresu
    if (!WiFi.softAPConfig(local_IP, gateway, subnet)) {
        Serial.println("SoftAP Config Failed!");
    }
}

```

```
// Postavi ESP32 kao Access Point
WiFi.softAP(ssid, password);
Serial.println("Access Point started");
Serial.print("IP address: ");
Serial.println(WiFi.softAPIP());

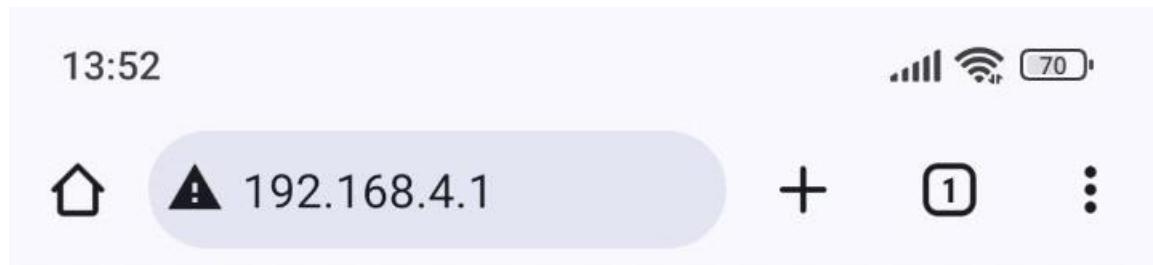
// Postavi rute za server
server.on("/", handleRoot);
server.on("/upaliprekidac", handleRelayOn);
server.on("/ugasiprekidac", handleRelayOff);

server.begin();
}

void loop() {
    server.handleClient(); // Obradi HTTP zahteve
}
```

Демонстрација пројекта паметног прекидача (ESP32-S NodeMCU)

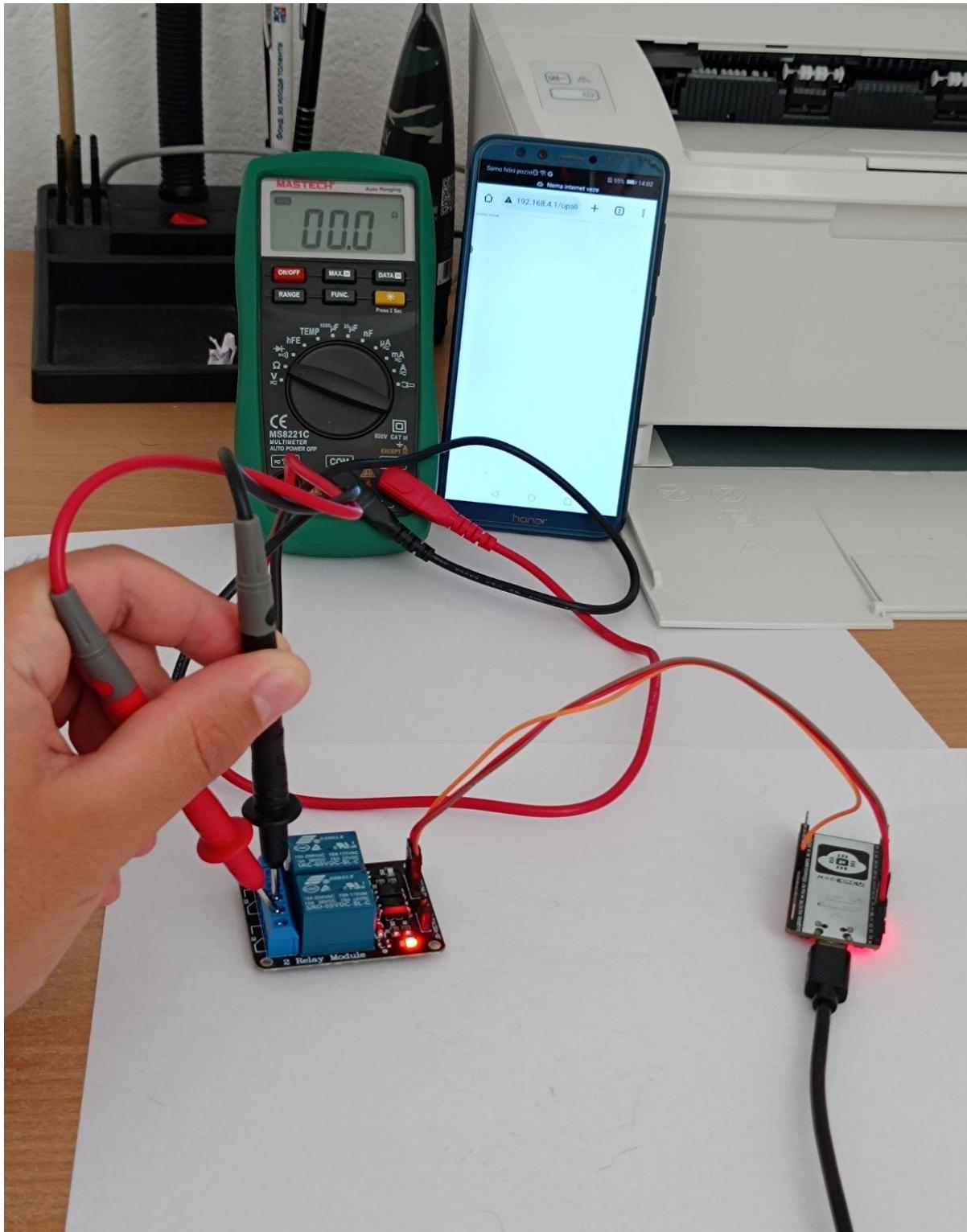
У овом делу пројекта приказана је демонстрација употребе ESP-32S NodeMCU плочице као Access Point-а за управљање релејем путем web интерфејса. Наиме, у тексту изнад видимо да смо поставили статичку IP адресу која ће нам представљати путању до web странице. Потребно је да се повежемо на SSID "SMART-switch" и да у претраживач унесемо **192.168.4.1**, након овог видећемо изглед наше web странице као на слици испод.



Слика 26. Изглед Web странице

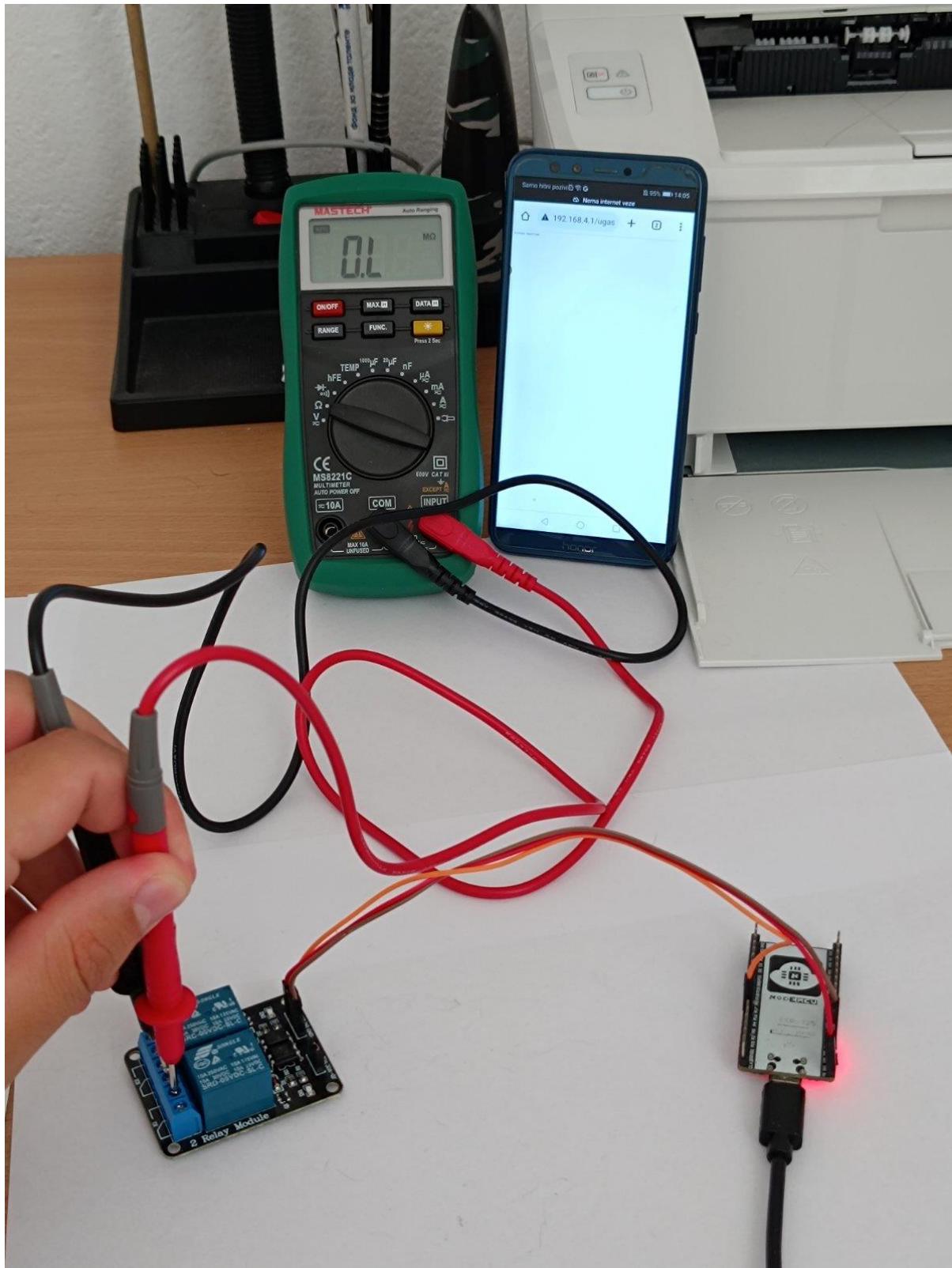
На слици 26. видимо web страница која је приказана на мобилном уређају. У горњем делу екрана налази се IP адреса 192.168.4.1, што сугерише да је корисник повезан на локалну мрежу коју хостује ESP32 уређај. Испод адресе налазе се два дугмента. Зелено дугме има текст "UPALI PREKIDAC", што активира релеј када се притисне. Црвено дугме има текст "UGASI PREKIDAC", што деактивира релеј када се притисне.

Тестирамо пројекат тако што прво палимо прекидач. Након притиска на зелено дугме добићемо поруку “POTROSAC AKTIVAN” и доћи ће до паљења релеја и индиковања путем паљења црвене диоде. Користићемо унимер да видимо да ли је прекидач затворен. Са слике 27. можемо видети да унимер показује $0\ \Omega$. То указује да имамо кратак спој. Односно, да имамо остварен контакт између COM и NO.



Слика 27. Релеј у фази провођења

Притиском на телефону на тастеру BACK, враћамо се на web страницу, где можемо тестирати гашење прекидача. Након притиска на црвено дугме добићемо поруку "POTROSAC NEAKTIVAN" и доћи ће до гашења релеја и индиковања путем гашења црвене диоде. На слици 28. можемо видети да унимер показује вредност OL (overflow) што указује да имамо прекид (немамо спрегу контаката COM и NO).

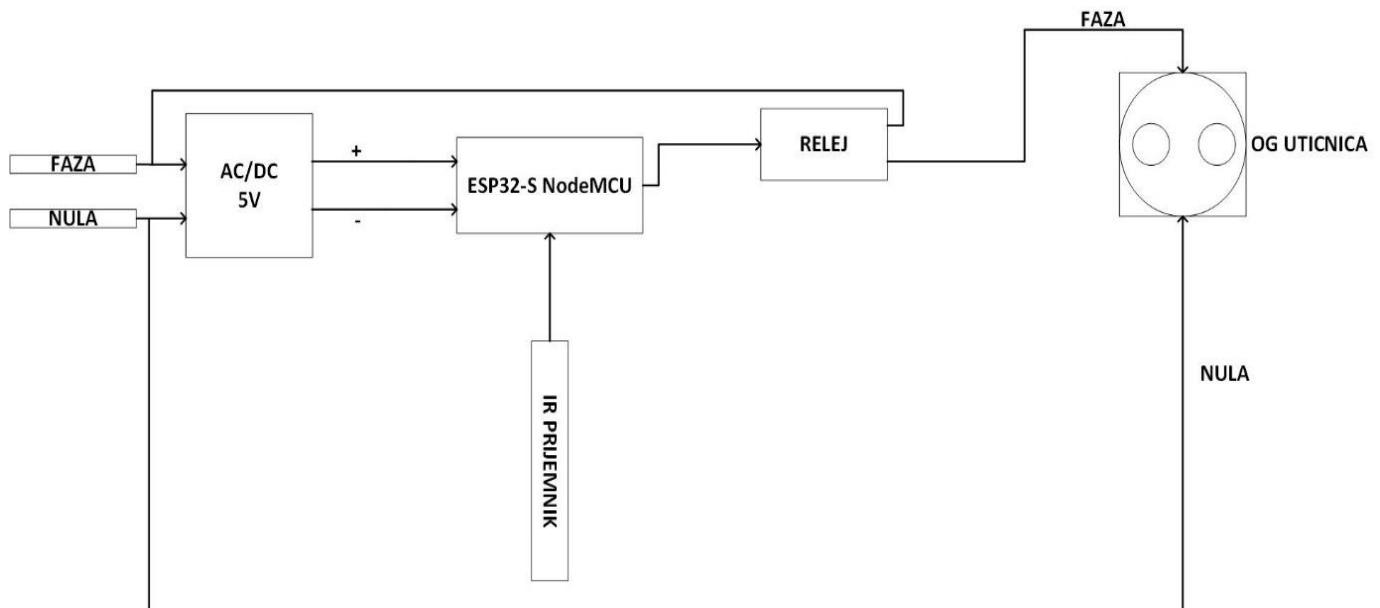


Слика 28. Релеј у фази непровођења

Паметна утичница за кућну употребу

У овој фази мастер рада фокус је на имплементацији паметне утичнице коришћењем ESP-32S NodeMCU плочице, IR пријемника и апликације Blynk. Ова имплементација се изводи у кућним условима, са циљем да се створи практичан и функционалан уређај који омогућава даљинско управљање потрошачима електричне енергије. Након претходно реализованих фаза пројекта, где је успешно имплементиран паметни прекидач коришћењем IR пријемника и управљање релејем путем web странице на којој је ESP32 функционисао као Access Point, сада је циљ креирати уређај који је погодан за кућну употребу. Овај уређај користи ESP-32S NodeMCU плочицу као централни контролни систем, који управља релејом повезан на стандардну електричну утичницу.

Посебна пажња посвећена је напајању система. За конверзију наизменичне струје у једносмерну, користи се електроника адаптера мобилног телефона, што омогућава стабилно напајање ESP32 плочице. Релеј у систему контролише проток струје до утичнице, у који се може укључити било који потрошач, чиме се омогућава даљинско управљање различитим електричним уређајима у домаћинству. Интеграцијом IR управљања и Blynk апликације, овај пројекат постаје флексибилно и модерно решење, омогућавајући кориснику да управља потрошачима чак и када није физички присутан. Кроз наредне секције биће детаљно описаны сви аспекти пројекта, укључујући употребљене компоненте, начин њиховог повезивања и комуникације, као и процес реализације овог паметне утичнице.



Слика 29. Блоковска шема паметне утичнице

Ова блоковска шема приказује како је повезан систем за даљинско управљање електричним уређајем путем ESP32-S NodeMCU, релеја и AC/DC претварача.

- Фаза и Нула (AC улаз): Овај део шеме приказује улазну фазу и нулу (наизменичну струју из мреже, обично 220V AC) која улази у систем.
- AC/DC Претварач (AC/DC 5V, користићемо електронику из пунјача телефона): AC/DC претварач служи за конверзију наизменичне струје (AC) у једносмерну струју (DC) од 5V и 2.1A. Овај блок је повезан са фазом и нулом на улазу и генерише DC напон који напаја ESP32-S NodeMCU.
- ESP32-S NodeMCU је централни микропроцесор који управља целокупним системом. Он прима напајање од AC/DC претварача и контролише релеј. Такође је повезан са IR пријемником, што омогућава даљинско управљање путем инфрацрвеног сигнала (даљински управљач).
- IR пријемник: IR пријемник је повезан са ESP32-S NodeMCU и омогућава управљање системом путем инфрацрвених сигнала (путем даљинског управљача).
- Релеј је електромеханички прекидач који је контролисан од стране ESP32-S NodeMCU. Релеј се укључује или искључује на основу команди примљених са ESP32-S, чиме се контролише проток струје ка излазној утичници.
- Излазна утичница (OG УТИЧНИЦА): Излазна утичница је место где можете прикључити уређај који желите да контролишете. Ова утичница прима фазу и нулу када је релеј укључен, чиме омогућава напајање прикљученог уређаја.

Овај систем омогућава даљинско управљање електричним уређајима путем ESP32-S NodeMCU контролера, што га чини погодним за аутоматизацију у домаћинству.

AC/DC Претварач

AC/DC адаптер се повезује на извор наизменичне струје (AC), обично из зидне утичнице (110V или 220V зависно од региона). AC/DC конвертор се састоји од: **улазног конектора** (место где се адаптер повезује на извор напајања (нпр. AC зидна утичница)). Обично се користи стандардни AC конектор), **трансформатор** (смањује високи улазни AC напон на нижи напон), **исправљач** (претвара AC у DC), **филтри** (кондензатори, користе се за филтрирање у излазном напону након исправљења), **регулатор напона** (стабилизује излазни напон на 5V), **заштита од пренапона** (користи се за заштиту уређаја од превисоког напона), **заштита од преоптерећења** (онемогућава рад адаптера када се прекорачи максимална излазна струја (2A)), **термална заштита** (спречава прегревање уређаја), **излазни конектор** (место где се излазни DC напон повезује на уређај који се напаја, USB), **оклон и кушиште** (заштита за унутрашње компоненте, обезбеђује сигурност и стабилност адаптера), **LED индикатор** (често присутан на адаптерима, сигнализира да је адаптер укључен и да функционише исправно).

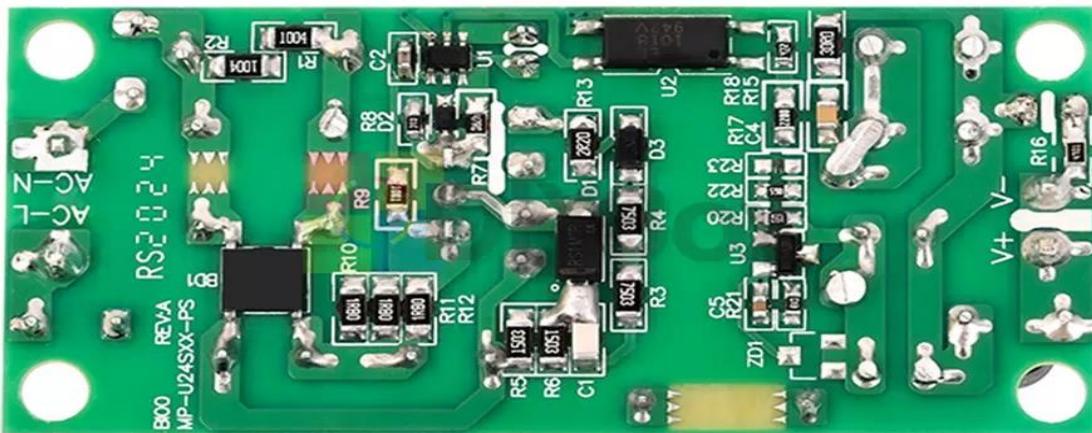
- Трансформатор: Први део адаптера је трансформатор који смањује високи улазни AC напон на нижи ниво. Трансформатор функционише на

принципу електромагнетне индукције, где се примарни намотај повезује са извором напајања, а секундарни намотај даје смањени напон.

- Исправљачи: Диодни или мостни исправљач: Након трансформације, AC напон се исправља у DC напон помоћу диода или мостног исправљача. Диода пропушта струју само у једном смеру, претварајући наизменичну струју (AC) у једносмерну (DC). У мостном исправљачу, четири диоде су повезане тако да омогућавају исправљење целокупног AC циклуса.
- Филтри, овде спадају кондензатори. Након исправљања, напон може имати таласасте облике. Кондензатори се користе за филтрирање ових таласања, смањујући шум и пружајући стабилнији DC напон. Обично се користе два кондензатора: један за филтрацију на улазу и други за излаз.
- Регулатор напона: Овај део осигурује да излазни напон остане константан (5V) без обзира на варијације у улазном напону или оптерећењу. То се може постићи коришћењем линеарних регулатора (нпр. LM7805) или PWM (Pulse Width Modulation) регулатора, који ефикасније управљају снагом.
- Заштита: Многи адаптери укључују заштитне склопове као што су: Защита од пренапона: Спречава оштећење уређаја у случају високог напона. Защита од преоптерећења: Онемогућава адаптер у случају да се превазилази максимална струја (2A). Термална заштита: Спречава прегревање уређаја.



Слика 30. AC/DC конвертор



Слика 31. AC/DC конвертор (штампа)

У наставку рада објаснићемо детаљно step-down трансформатор, исправљаче и регулатор напона. У пројекту ћемо узети AC/DC конвертор пуњача телефона. Ово је урађено због недостатка времена и немогућности набављања одговарајућих компоненти.

Step-down трансформатор

Step-down трансформатор је електрични уређај који смањује напон напајања наизменичном струјом (AC). Састоји се од примарног намотаја, секундарног намотаја и гвозденог језгра. Када се AC напон примени на примарни намотај, ствара се променљиво магнетно поље у гвозденом језгру. Ово магнетно поље затим индукује напон у секундарном намотају, али на нижем нивоу напона него што је то случај са примарним намотајем. Трансформатор ради на принципу међусобне индукције, познатом и као Фарадејев закон електромагнетне индукције, који каже да је величина напона директно пропорционална брзини промене магнетног флуksа. Да бисмо дубље разумели перформансе трансформатора, хајде да се упознамо са његовим основама. То значи да се струја индукује у калему када се нађе у близини калема кроз који тече струја и који има променљив магнетни флуks. Ова индукована струја је директно пропорционална брзини промене струје.

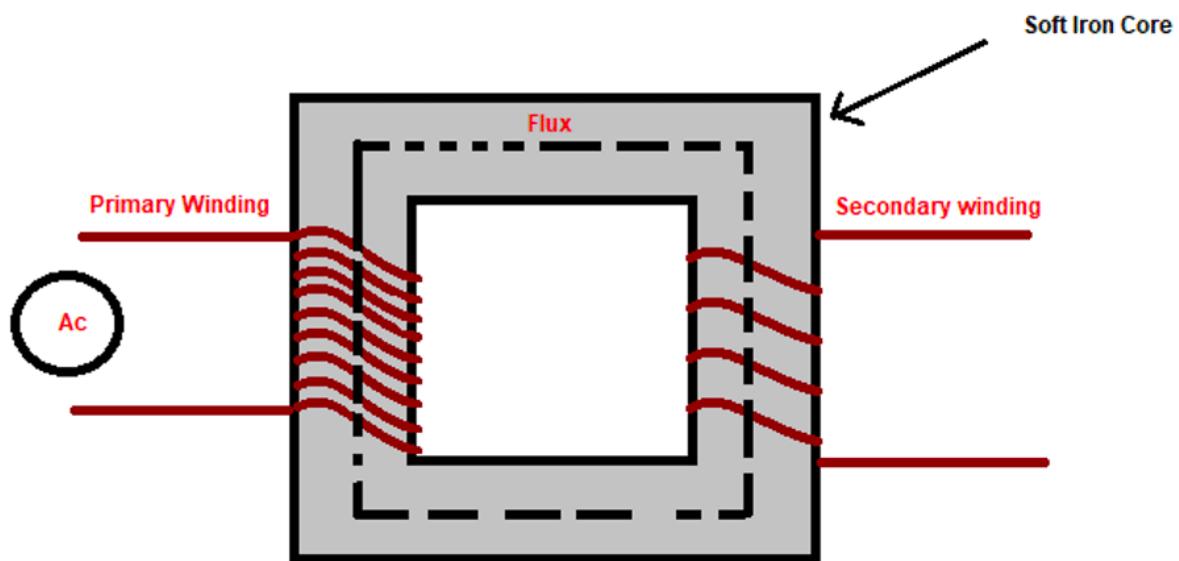
Конструкција трансформатора

Трансформатор се састоји од следећих главних делова:

- Језгро, калемови који су намотани на одређени материјал заједно формирају језгро трансформатора. Та језгра су израђена од материјала са веома високом пермеабилношћу који је способан да преноси флукс. Језгро трансформатора делује као пут или канал за лак проток магнетног флукса. Ова језгра се праве од феромагнетних материјала који имају високу пермеабилност, попут гвожђа. У трансформаторима користимо танке металне гвоздене листове уместо једног чврстог језгра, јер једно чврсто језгрво изазива већу генерацију вртложних струја, што смањује ефикасност трансформатора.
- Намотај, намотаји у трансформаторима су жице које се називају калемови. Овде користимо жице са мањим отпором и добром проводљивошћу, што је потребно за постизање добре ефикасности трансформатора. Обично се у намотајима трансформатора користи бакар, јер има добру електричну проводљивост и врло низак отпор у поређењу са другим материјалима. Такође, није скуп као злато, сребро или платина.

Принцип рада трансформатора

Трансформатор ради на принципу међусобне индукције. Дакле, када дође до промене струје у једном калему, електрична струја се такође индукује у другом калему који се налази у његовој близини. Сваки трансформатор се састоји од два калема или намотаја: примарног и секундарног. Примарни намотај је повезан са извором наизменичне струје (AC), док је секундарни повезан са оптерећењем. Када се



Слика 32. Попречни пресек трансформатора

наизменична струја доведе на примарни намотај, ствара се магнетни флукс. Кроз језгро трансформатора, магнетно поље завршава свој пут. Када секундарни намотај дође у контакт с овим магнетним флуксом, на њему се индукује електромоторна сила (EMS). Снага генерисане електромоторне силе зависи од броја намотаја у секундарном калему. $N_1 > N_2$ где је: N_1 = Број намотаја у првом калему, N_2 = Број намотаја у другом калему. Однос између напона и броја намотаја у калему је: $V_p/V_s = N_p/N_s$ где је: V_p = Напон у примарном калему, V_s = Напон у секундарном калему, N_p = Број намотаја у примарном калему, N_s = Број намотаја у секундарном калему.

Трансформатор који има већи број намотаја у примарном намотају, а мањи број у секундарном намотају назива се **step-down** трансформатор. Као што можемо видети из претходне једначине која описује однос између броја намотаја и напона, ако је број намотаја у примарном намотају већи од броја намотаја у секундарном, тада је електромоторна сила (EMS) генерисана у секундарном намотају мања од улазног напона у примарном. Због тога добијамо нижи напон у секундарном калему ste-down напонског трансформатора. Као што назив сугерише, step-down трансформатор се користи за претварање напона вишег нивоа у напон нижег нивоа.

Хајде сада да разумемо претходни процес кроз пример. Претпоставимо да имамо извор наизменичне струје који може произвести 210 V. Ако користимо трансформатор са N_p (број намотаја у примарном калему) = 20000, и N_s (број намотаја у секундарном калему) = 100, онда је V_s (напон на секундарном калему) дат формулом: $V_s = (V_p * N_s) / N_p$. Добијамо напон на секундару да је 1.05 V.

Исправљачи

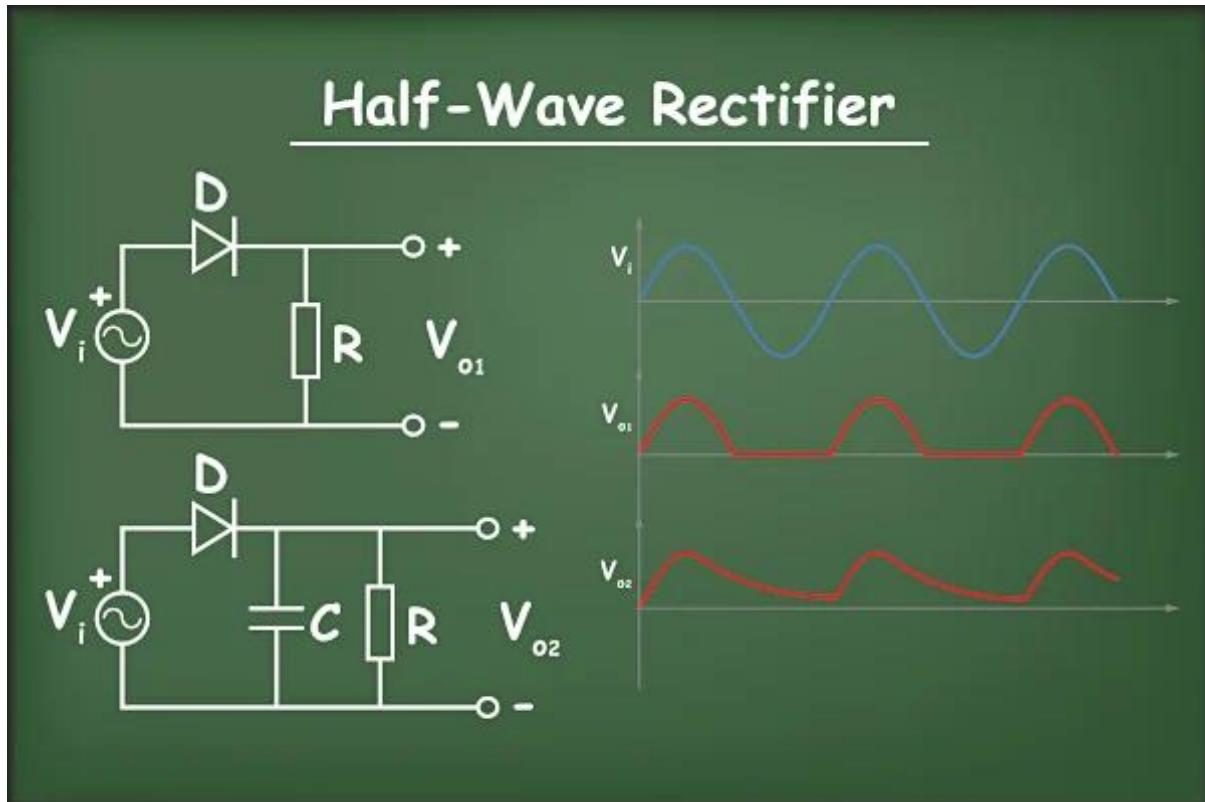
Исправљач (rectifier) је електронска компонента која претвара наизменичну струју (AC) у једносмерну струју (DC), обезбеђујући једносмеран ток електричног наелектрисања. Као што зnamо, електрична енергија долази до наших домаца из електроенергетских мрежа у облику наизменичне струје, али већина електричних уређаја које користимо, као што су мобилни телефони, рачунари, телевизори, фрижидери итд., раде коришћењем једносмерне струје. Због тога исправљач постаје веома корисна компонента у електроници. Овај део рада даје детаљно разумевање електронске компоненте познате као исправљач, укључујући његове типове, као што су пуноталасни и полуталасни исправљачи.

Исправљачи су електронски уређаји који претварају наизменичну струју (AC) у једносмерну струју (DC). Наизменична струја периодично мења смер, док једносмерна струја тече константно у једном смеру. Исправљачи су кључни у многим електронским апликацијама где је потребан стабилан DC напон. Већина исправљача користи P-N спој диоде за претварање AC у DC, али неки користе и различите врсте материјала, као што су полупроводнички материјали попут карборунда или специјализовани полупроводнички уређаји као што су тиристори и метал-оксидни полупроводнички транзистори са ефектом поља (MOSFET).

Принцип рада полуталасног исправљача

У полуталасном исправљању само се позитивна или само негативна полупериода наизменичног напона пропушта кроз исправљач, што зависи од поларизације диоде. Тиме се на излазу добија свака друга полупериода са нултом вредношћу између. Овакав тип исправљача се користи када се жели уштеда на

материјалу. Мана му је што отежава филтрирање, па се стога примењује само за изузетно мале снаге потрошача, којима не смета таласаст напон. Лако је конструисати полуталасни исправљач. Полуталасни исправљач се користи за претварање AC (наизменичне струје) сигнала у DC (једносмерну струју) сигнал пролазећи кроз негативну или позитивну полу-периоду. Једини недостатак је што је мање ефикасан у поређењу са пуноталасним исправљачем. Током позитивног полу-циклуса AC сигнала, диода D се укључује и омогућава проток струје кроз оптерећење (отпорник R). У овом тренутку, струја пролази кроз диоду, стварајући напон на излазу, што резултира позитивним напоном на оптерећењу. Ефикасност полуталасног исправљача је 40,5%, што се сматра низким од ефикасности пуноталасног исправљача.



Слика 33. Полуталасни исправљач

Принцип рада пуноталасног исправљача

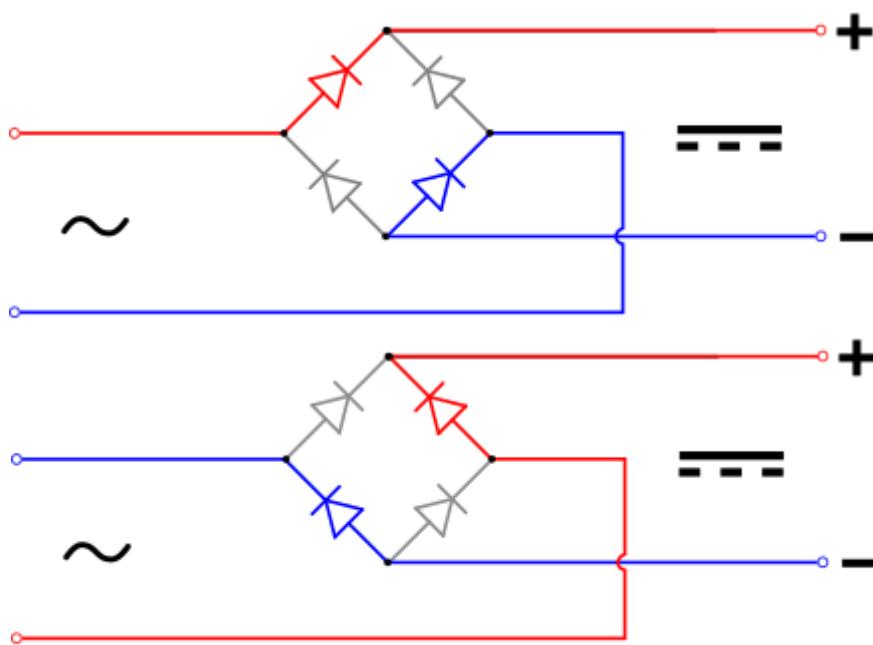
Пуноталасни исправљач захтева више диода за рад и конструкцију. Пуноталасни исправљач исправља и негативну и позитивну полу-периоду улазног наизменичног сигнала. Ефикасност пуноталасног исправљача је двоструко већа од ефикасности полуталасног исправљача. Због тога се ови исправљачи најчешће користе у свим електронским уређајима, јер могу издржати високе напоне и струје са интензивном снагом. У ову групу исправљача спада **Грецов мост**.

Грецов спој, Грецов мост или диодни мост представља повезивање четири (или више код вишеспособних система) диода или тиристора у колу на чијем се излазу добија напон истог поларитета без обзира какав је поларитет на улазу. Грецов спој се користи се код исправљача за претварање наизменичне струје у једносмерну струју. Грецов спој омогућава пуноталасно исправљање наизменичног напона. Грецов спој је добио име по

свом проналазачу, физичару Леу Грецу. Према конвенцији о току струје коју је увео још Бенцамин Френклин и коју данас користи већина инжењера, узима се да струја тече кроз проводник од позитивног ка негативном полу. У стварности, слободни електрони у проводнику се скоро увек крећу од негативног ка позитивном полу. Међутим, за већину примера у електротехници, потпуно је неважан смер тока струје. Објашњење доле следи Френклинову конвенцију. У позитивној полупериоди, леви прикључак моста је на вишем потенцијалу од прикључка на десном крају моста, тј. напон (разлика потенцијала) гледан од левог прикључка ка десном је позитиван. Горња лева диода је директно поларисана, па проводи струју, док је доња лева диода инверзно поларисана, па не дозвољава протицање струје кроз њу. Горња десна диода је инверзно поларисана, па не дозвољава електронима пролаз кроз њу. Струја је принуђена да тече из прикључка означеног са „+“ и да се врати у Грецов спој кроз прикључак означен са „-“. Како је доња лева диода, инверзно поларисана, једини пут за струју је кроз доњу десну диоду назад у извор.

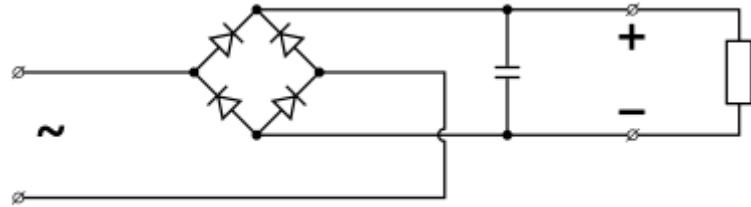
У негативној полупериоди, леви прикључак моста је на нижем потенцијалу од прикључка на десном крају моста, тј. напон гледан од левог прикључка ка десном је негативан. Сад је горња десна диода директно поларисана, па проводи струју, док је доња десна диода инверзно поларисана, па не дозвољава протицање струје кроз њу. Горња лева диода је инверзно поларисана, па не дозвољава електронима пролаз кроз њу. Струја је опет принуђена да тече из прикључка означеног са „+“ и да се врати у Грецов спој кроз прикључак означен са „-“. Како сада је доња десна диода инверзно поларисана, једини пут за струју је кроз доњу леву диоду назад у извор.

Струја у левој страни мреже наизменична, а у десном исправљена. У оба случаја (у и позитивној и у негативној полупериоди), горњи прикључак (означен са +) остаје на вишем потенцијалу, док онај доњи (означен са -) на нижем потенцијалу. Ово је тачно без обзира да ли је улаз (леви) крај наизменичан или једносмеран, па се ово коло не користи да се од наизменичног добије једносмерни напон, већ да се уведе заштита опреме од негативног поларитета, нпр када се батерије поставе наопачке или се обрне смер прикључака са једносмерног извора.



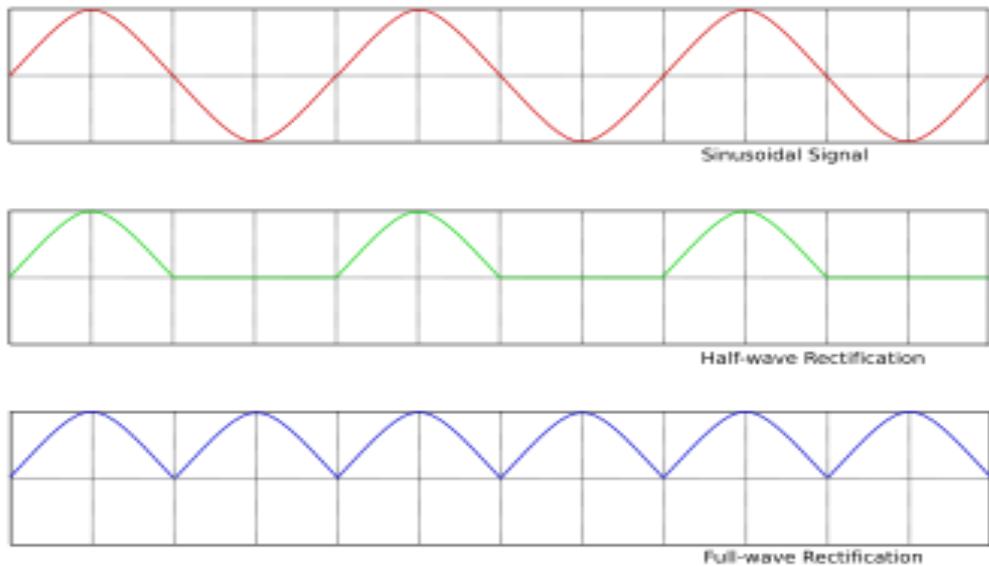
Слика 34. Грецов спој

За многе примене је пожељна употреба паралелног везаног кондензатора, јер је излазни напон, иако исправљен и даље пулсирајући. Улога кондензатора је да смањи варијације излазног напона. Једно објашњење пеглања је да кондензатор за променљиву струју представља пут мање импедансе, чиме се смањује напон између два краја пријемника, односно струја кроз њега. Друго објашење је да кондензатор служи да акумулира електричну енергију, коју ослобађа када опадне напон на његовима крајевима (а тиме и крајевима пријемника, пошто је у питању паралелна веза). Ослобођена енергија противе кроз пријемник, чиме се надокнађује опадање напона.



Слика 35. Грецов спој са кондензатором

Овакво једно коло се користи у AC/DC конвертору у нашем пројекту.

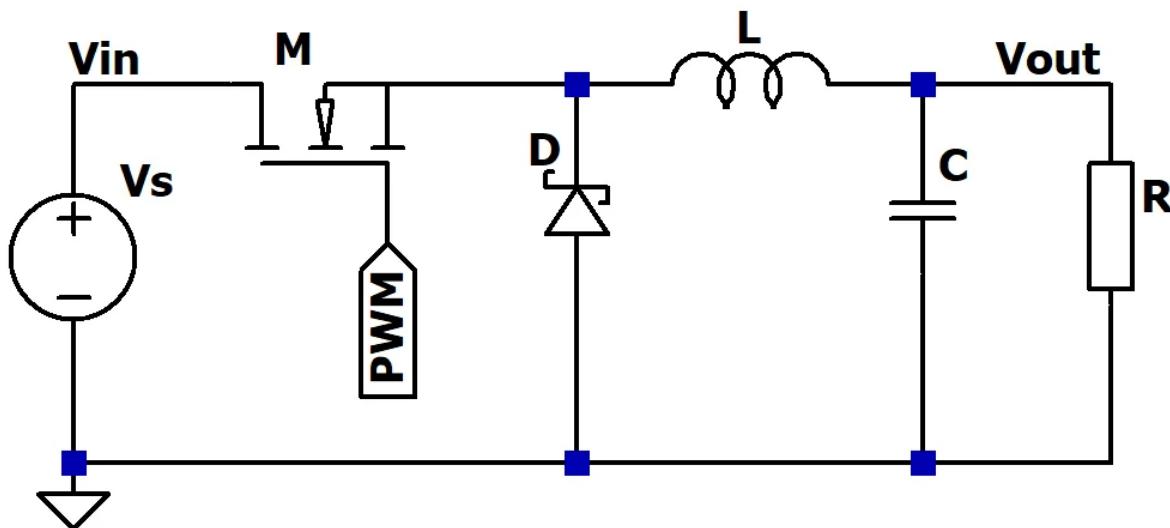


Слика 36. Таласни облици полуталасног и пуноталасног исправљача

Регулатор напона

Регулатори електричног напона представљају уређаје чија је основна улога обезбеђивање константног излазног напона. Треба знати да се излазни напон мења у зависности од промене улазног напона и јачине струје оптерећења. Многи електронски уређаји захтевају оперативни напон и јачину струје константних вредности, па је због тога неопходан одређени вид регулације. Користе се у електро-механичким склоповима, за регулацију напона пасивних или активних електричних компонената. У зависности од дизајна, могу се користити за регулацију једне или више различитих вредности наизменичних или једносмерних напона. Изузев пасивних шант регулатора, сви савремени електрични регулатори напона раде са упоређивањем стварног излазног напона са унапред дефинисаним, непромењивим, референтним напоном. Постоје различите врсте регулатора напона, нама је од интереса **прекидачки регулатор**. Код AC/DC конвертора уградјених у пуњаче за телефон (искоришћен у овом пројекту), најчешће се користи **switching (прекидачки) регулатор напона**, познат и као **buck converter** тј. конвертор наниже.

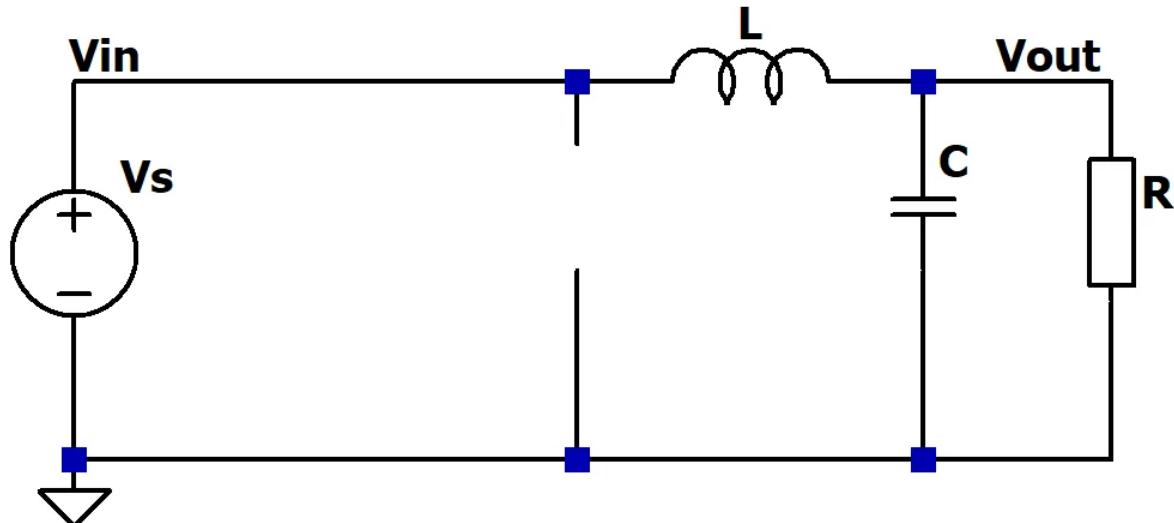
Овај тип регулатора је ефикасан и омогућава конверзију високог AC напона у ниски DC напон потребан за пуњење мобилних уређаја, обично 5V или више, зависно од типа пуњача. Прекидачки регулатори раде на принципу брзог укључивања и искључивања прекидача (обично транзистора), заједно с индуктором и кондензатором, што омогућава висок ниво ефикасности и смањење губитака енергије, у поређењу с линеарним регулаторима. Прекидачки регулатор је склоп који користи прекидач за напајање, индуктор и диоду за пренос енергије са улаза на излаз. Прекидач за напајање, обично транзистор са ефектом поља (FET), укључује се и искључује путем IC контролера прекидача, који прати излаз прекидачког регулатора у повратној петљи. Ово осигурава да регулатор одржава константан излаз под нормалним радним условима. У неким прекидачким регулаторима, FET је засебна компонента, ван контролера прекидача. У другим верзијама, FET и контролер су у истом интегрисаном колу (IC).



Слика 37. Buck конвертор

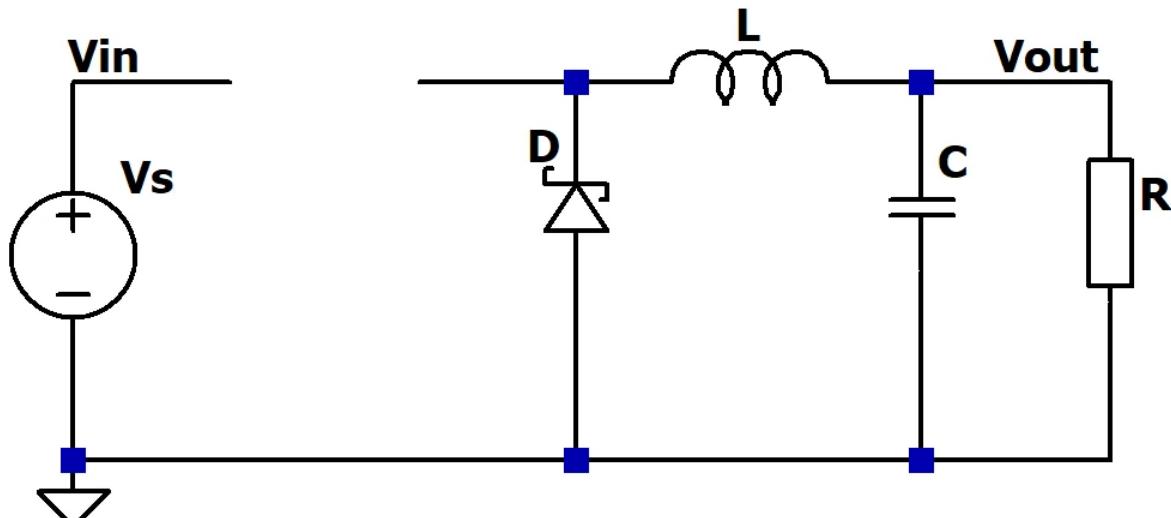
Принцип рада прекидачког регулатора

Када је прекидач затворен. У овом стању, MOSFET (M) је укључен, а диода (D) је поларизована у инверзном смеру; стога добијамо коло приказано на слици 38.



Слика 38. Buck конвертор када је прекидач затворен

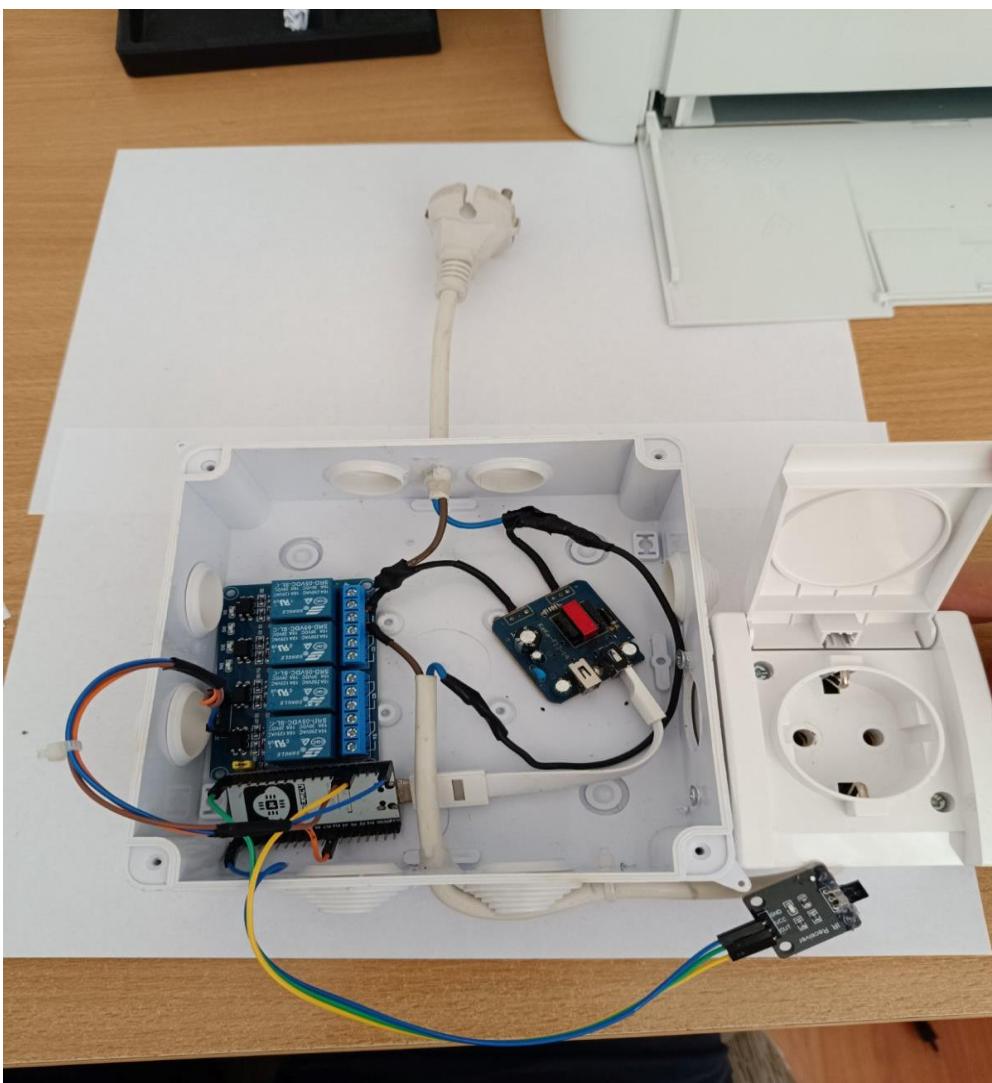
На почетку, када је MOSFET искључен, у коло не тече струја. Када се MOSFET укључи, струја ће почети да тече кроз индуктор (L), пунећи га. Струја кроз индуктор расте линеарно. Док се струја кроз индуктор повећава, магнетско поље у том индуктору ће се такође повећавати. Почетни ток струје ствара потенцијал преко индуктора који се противи напона напајања. Овај пад напона кроз индуктор смањује корисни напон на оптерећењу (R). Временом, брзина промене струје опада, а напон преко индуктора такође опада, повећавајући напон на оптерећењу. Током овог времена, индуктор прикупља енергију у облику магнетског поља. Енергија се прикуља у индуктору и излазном кондензатору (C). У овој фази, диода је обрнуто поларизована и не проводи струју.



Слика 39. Buck конвертор када је прекидач отворен

Када је прекидач отворен. У овом стању, MOSFET (M) је искључен, а сада је диода (D) у проводном режиму, зато добијамо коло приказано на слици 39. Ако је MOSFET искључен док се струја још мења, увек ће постојати пад напона преко индуктора. То значи да ће корисни напон на оптерећењу (R) увек бити мањи од улазног напона. Када се MOSFET искључи, енергија која је прикупљена у магнетном пољу око индуктора (L) се враћа у коло. Напон преко индуктора сада је супротног поларитета у односу на напон генерисан када се струја повећавала. Сада је диода (D) у проводном режиму, и пружа повратни пут за струју кроз индуктор. Прикупљена енергија у магнетном пољу индуктора напаја оптерећење кроз диоду са опадајућом струјом. Током ове фазе циклуса пребацања, индуктор постаје главни извор излазне снаге. Док индуктор наставља да испоручује своју прикупљену енергију оптерећењу, струја кроз индуктор ће почети да опада.

Практична реализација паметне утичнице за кућне услове



Слика 40. Практична реализација паметне утичнице

На слици 40. је приказана унутрашњост кутије 15x15 см у којој је постављена електронска опрема за реализацију паметне утичнице. У кутији се налазе:

- ESP32-S NodeMCU: Овај микроконтролер је постављен на доњем левом делу кутије. Жиће су повезане с њим за управљање уређајима, укључујући релеј и IR пријемник.
- Релејни модул: Налази се у горњем левом делу кутије. Овај модул има четири релеја, али у овом пројекту се користи један релеј за управљање струјом која иде према утичници.
- IR пријемник: Постављен је у доњем десном делу слике, изван кутије. Овај пријемник је повезан са ESP32-S NodeMCU-ом и користи се за пријем сигнала са даљинског управљача.
- Напајање: На слици се види адаптер који обезбеђује напајање за ESP32, напајање релеја и IR пријемника је преко плочице ESP32.
- Утичница: На десној страни слике налази се класична OG утичница у коју се укључују кућни уређаји. Релеј управља струјом која иде до ове утичнице.

У овом пројекту, pin P13 на ESP32-S NodeMCU је коришћен за повезивање IR пријемника, док је pin P16 повезан са релејем. Ова комбинација омогућава даљинско управљање утичницом преко инфрацрвеног даљинског управљача, као и путем мобилне апликације Blynk користећи ESP32.

Blynk платформа

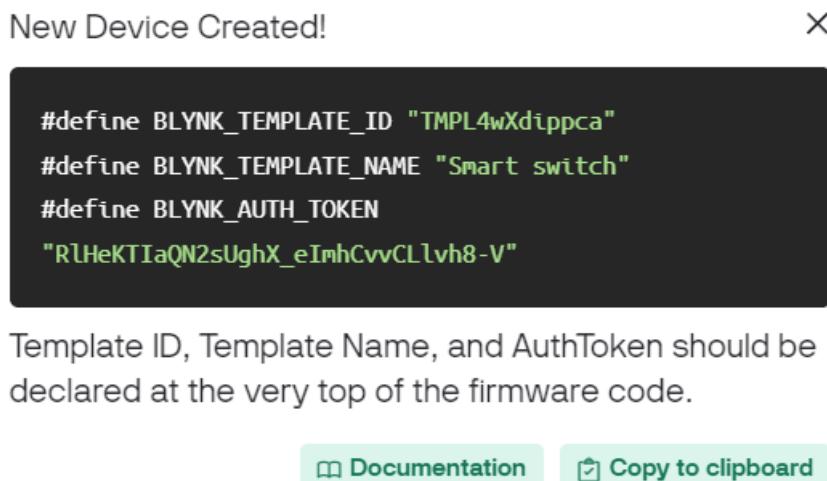
Blynk је платформа која омогућава креирање мобилних апликација за контролу IoT (Internet of Things) уређаја. Са Blynk-ом, можете контролисати различите хардверске платформе, као што су ESP32, Arduino, Raspberry Pi, и многе друге, преко Интернета. Blynk пружа једноставно корисничко окружење за креирање интерфејса који може укључивати дугмиће, клизаче, графове, и многе друге UI елементе који се повезују са вашим уређајем. Blynk подржава различите врсте мрежних конекција, укључујући Wi-Fi, Bluetooth, и GSM, што омогућава флексибилност у примени. Омогућава контролу и надзор ваших IoT уређаја преко мобилне апликације, без потребе за комплексним подешавањима. Апликација користи widgete (дугмићи, клизачи, графови, LED индикатори, итд.) за креирање корисничког интерфејса. Сваки widget може бити повезан са одређеним пином или функционалношћу на вашем IoT уређају. Blynk апликација је доступна за iOS и Android уређаје. Blynk омогућава прикупљање и визуализацију података у реалном времену, као и управљање уређајима путем апликације. Ваш уређај се може повезати са Blynk сервером у облаку, што омогућава приступ са било које локације која има интернет конекцију. Уређаји могу радити и у offline моду, са локалним Blynk сервером, што је корисно у ситуацијама када је интернет конекција нестабилна. Blynk је идеалан за хобисте, студенте и професионалце који желе брзо и једноставно имплементирати пројекте паметних кућа, аутоматизације, надзора и многих других IoT апликација. Коришћење Blynk платформе значајно убрзава развој IoT решења јер елиминише потребу за писањем комплексних интерфејса и омогућава фокус на основну функционалност пројекта.

У овом пројекту користе се free верзија, тако да могућности које даје ова платформа су оскудни. Имамо могућност постављања прекидача на површину без додатног дизајна и боље функционалности.

Поступак за повезивање плочице са Blynk апликацијом

Регистрација на Blynk платформи и креирање темплејта

1. Посетите адресу <https://blynk.io/> и региструјте се са вашом email адресом.
2. Након регистрације потребно је направити Template. Уђите на картицу Developer Zone/My Templates/New Template. Унесите име темплејта, хардвер и тип конекције. Сада је потребно подесити template по потребама. Одлазимо у datastream и додајемо виртуелни пин (V1) за релеј. Подешавамо Web Dashboard тако што превучемо прекидач на површину. Последњи корак је да сачувамо template.
3. Потребно је додати уређај. Одлазимо на картицу Devices/New Device/From template/Name template/Create.
4. Након корака број 3. имаћемо идентификационе податке који служе за повезивање плочице. Потребно их је сачувати и унети у код пројекта.

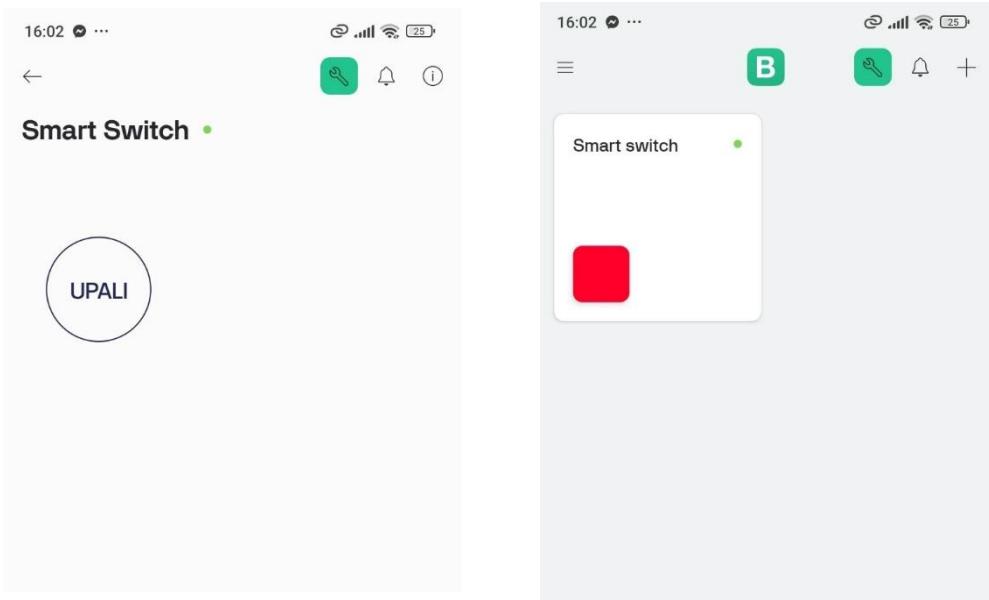


Слика 41. Идентификациони подаци

Преузимање Blynk апликације и креирање налога

1. Преузмите апликацију Blynk на мобилном телефону са Google Play Store-а или Apple App Store-а.
2. Пријавите се на исти налог на који смо направили template. Уђите у template Smart-switch. Додајте на површину дугме притиском на + и одабиром дугмета Button. Кликните на њега и изaberite Datastream у коме ће бити приказан прекидач који смо направили приликом конфигурисања template-а. Изберите мод рада прекидача, као и дизајн.
3. Уђите у template и идите на симбол подешавања и ту изaberite Tyle type да буде такав да template има дугме као пречицу. На овај начин не морамо

улазити у template и активирати дугме. Штедимо време и добијамо лепши дизајн.



Слика 42. Изглед апликације Blynk

Приликом уласка у апликацију добијамо прозор са десне стране. Ово је template који има пречицу дугме (црвена боја). Притиском на њега палимо и гасимо потрошач. Са леве стране имамо прозор који добијамо када уђемо у template. Овде такође палимо и гасимо потрошач притиском на дугме.

Софтверска реализација паметне утичице за кућне услове

Коришћење библиотеке BlynkSimpleEsp32

Библиотека BlynkSimpleEsp32.h је део Blynk екосистема која омогућава једноставно повезивање ESP32 микроконтролера са Blynk платформом. Ова библиотека омогућава ESP32 да комуницира са Blynk сервером, чиме омогућава контролу и надзор вашег ESP32 уређаја преко мобилне апликације. Основне функције библиотеке BlynkSimpleEsp32.h:

- Повезивање са Blynk сервером: BlynkSimpleEsp32.h пружа једноставан начин за повезивање ESP32 плочице са Blynk сервером користећи Wi-Fi конекцију. Функција Blynk.begin(auth, ssid, pass) аутоматски повезује уређај са Wi-Fi мрежом и Blynk сервером.

- Одржавање сталне везе са сервером: Функција Blynk.run() користи се у главној петљи (`void loop()`) да би се одржавала стална веза са Blynk сервером и омогућила размена података у реалном времену.
- Руковање Blynk Widget-има: Библиотека омогућава једноставно руковање Blynk widget-има, као што су дугмад, клизачи, графови итд., који су конфигурирани у Blynk апликацији. Функција `BLYNK_WRITE(Vx)` омогућава руковање уносима са апликације где `Vx` представља виртуални пин на који је widget повезан.
- Контрола хардверских пинова: Blynk омогућава директну контролу над хардверским пиновима ESP32 плочице помоћу мобилне апликације. Виртуални пинови (`Vx`) могу се користити за апстрактну контролу уређаја који су повезани на физичке пинове ESP32. У овом пројекту користили смо `V1`.
- Сигурност: Blynk библиотека користи Auth Token, који омогућава сигурно повезивање са Blynk сервером. Овај токен је јединствен за сваки Blynk пројект и повезује апликацију са специфичним уређајем.
- Подршка за више платформи: Иако је BlynkSimpleEsp32 специфична за ESP32 микроконтролер, Blynk библиотеке постоје за различите платформе, што омогућава сличан начин рада са различитим хардверским конфигурацијама.

Објашњење кода паметне утичице за кућне услове

У уводном делу кода, дефинишу се параметри који су неопходни за правилно функционисање система. Пре свега, неопходно је повезивање ESP32 микроконтролера са Blynk платформом. То се постиже коришћењем јединствених идентификатора као што су `BLYNK_TEMPLATE_ID`, `BLYNK_TEMPLATE_NAME`, `BLYNK_AUTH_TOKEN`. Ови идентификатори омогућавају да ваш уређај буде препознат на Blynk серверу, чиме се осигурува сигурна и поуздана комуникација. Затим, користе се библиотеке које олакшавају рад са различитим хардверским компонентама и мрежним повезивањем. Укључујемо IR библиотеку за управљање инфрацрвеним сигналима, Wi-Fi библиотеку за повезивање на бежичну мрежу, и Blynk библиотеку о којој је било речи у тексту изнад.

Да би овај пројекат функционисао, потребно је дефинисати име ваше Wi-Fi мреже и лозинку. Ови подаци омогућавају ESP32 микроконтролеру да се повеже на интернет, што је неопходно за комуникацију са Blynk сервером. Поред тога, дефинишу се и пинови на које су повезани IR пријемник и релеј. У нашем примеру, IR пријемник је повезан на пин P13, док је релеј повезан на пин P16. У `setup()` функцији, микроконтролер пролази кроз иницијалне кораке конфигурације. Прво се успоставља серијска комуникација, која омогућава праћење процеса преко серијског монитора. Затим се омогућава пријем IR сигнала, а пин на који је повезан релеј се поставља као излазни. То омогућава микроконтролеру да контролише релеј, односно да га укључује или искључује. Критичан део овог сегмента је и повезивање на Wi-Fi мрежу. Микроконтролер покушава да се повеже на мрежу користећи одговарајуће податке. Ако је повезивање успешно, на серијском монитору ће се исписати IP адреса уређаја. Ово је корисно јер вам омогућава да знате где се ваш уређај налази на мрежи, што може бити од помоћи приликом даљег развоја или решавања проблема. У коду, за управљање релејем користи се функција `BLYNK_WRITE(V1)`. Када корисник у апликацији притисне дугме повезано са

виртуелним пином V1, ова функција се активира. Ова функција затим проверава вредност коју је корисник послao—ако је вредност 1, релеј се укључује; ако је вредност 0, релеј се искључује. Овај једноставан али ефикасан механизам омогућава корисницима да на даљину контролишу уређаје повезане на релеј, пружајући максималну флексибилност.

Поред контроле путем Blynk апликације, овај пројекат омогућава управљање релејем и путем инфрацрвеног (IR) даљинског управљача. Ово се постиже декодирањем IR сигнала који прима IR пријемник повезан на ESP32. У функцији loop(), ESP32 стално ослушкује долазне IR сигнале. Када сигнал буде примљен, декодира се и претвара у хексадецимални формат који се затим користи за идентификацију команде. У зависности од примљеног кода, релеј се може укључити или искључити. У овом пројекту, ако се прими код 0x4FB00FF, релеј ће се укључити, док ће код 0x4FB4AB5 искључити релеј. Овај начин рада омогућава додатну контролу над уређајима путем даљинског управљача, што је идеално за ситуације када је мобилни уређај ван домашаја или када се жели брзо управљање без отварања апликације.

```
#define BLYNK_TEMPLATE_ID "TMPL4wXdippca"
#define BLYNK_TEMPLATE_NAME "Smart switch"
#define BLYNK_AUTH_TOKEN "RlHeKTIaQN2sUghX_eImhCvvCLlvh8-V"

#include <IRremote.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

const char* ssid = "xxxxxx"; // Naziv vaše Wi-Fi mreže
const char* password = "Srbija12356"; // Lozinka vaše Wi-Fi mreže

const int RECV_PIN = 13; // Pin na koji je povezan IR prijemnik
const int RELAY_PIN = 16; // Pin na koji je povezan relej

IRrecv irrecv(RECV_PIN);
decode_results results;

void setup() {
    Serial.begin(9600);
    irrecv.enableIRIn(); // Počni prijem IR signala

    // Postavi pin za relej kao izlaz
    pinMode(RELAY_PIN, OUTPUT);

    // Osiguraj da je relej isključen na početku
    digitalWrite(RELAY_PIN, HIGH);

    // Povezivanje na Wi-Fi mrežu
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
}
```

```

Serial.println("Connected to WiFi");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

// Inicijalizacija Blynk-a
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, password);
}

// Blynk funkcija za upravljanje relejem
BLYNK_WRITE(V1) {
    int pinValue = param.asInt();
    if (pinValue == 1) {
        digitalWrite(RELAY_PIN, LOW); // Uključi relej
    } else {
        digitalWrite(RELAY_PIN, HIGH); // Isključi relej
    }
}

void loop() {
    if (irrecv.decode(&results)) {
        unsigned long irCode = results.value;
        Serial.print("IR Code: ");
        Serial.println(irCode, HEX); // Prikazuje IR kod u
heksadecimalnom formatu

        switch (irCode) {
            case 0x4FB00FF: // Kod za uključivanje releja
                digitalWrite(RELAY_PIN, LOW);
                Blynk.virtualWrite(V1, 1); // Ažuriraj stanje u Blynk
aplikaciji
                break;
            case 0x4FB4AB5: // Kod za isključivanje releja
                digitalWrite(RELAY_PIN, HIGH);
                Blynk.virtualWrite(V1, 0); // Ažuriraj stanje u Blynk
aplikaciji
                break;
            default:
                // Ako kod nije prepoznat, ne radi ništa
                break;
        }

        irrecv.resume(); // Prima sledeći signal
    }
}

Blynk.run(); // Blynk handler
}

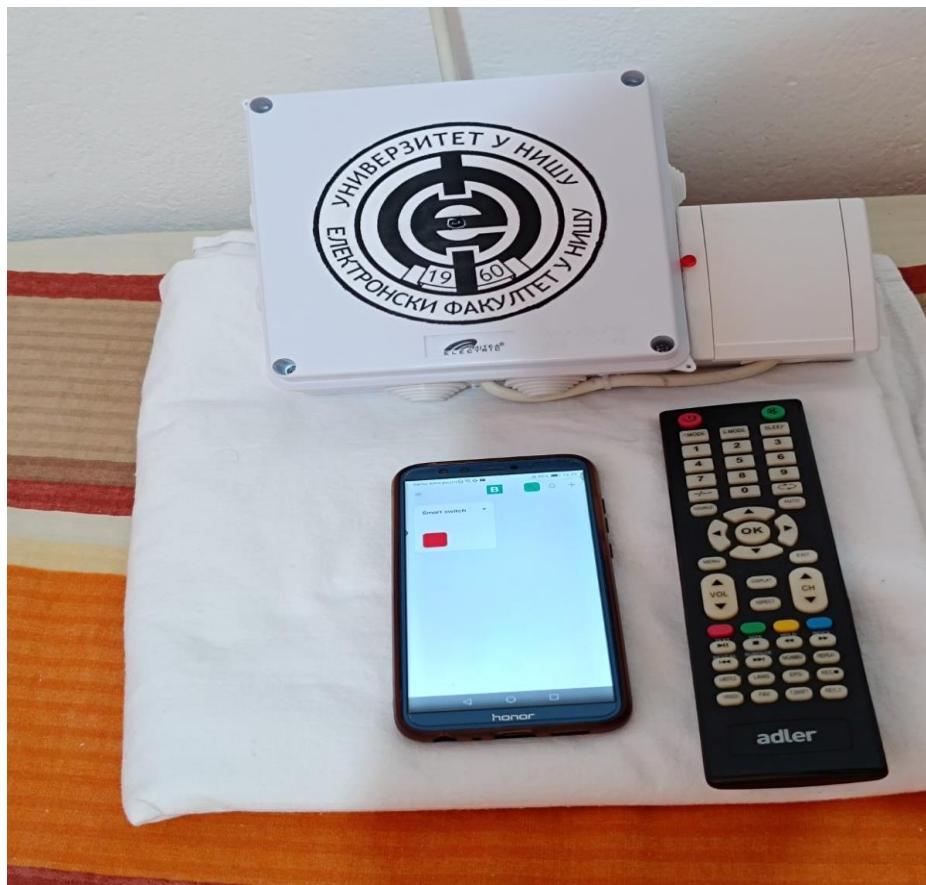
```

Коначна реализација и демонстрација паметне утичнице

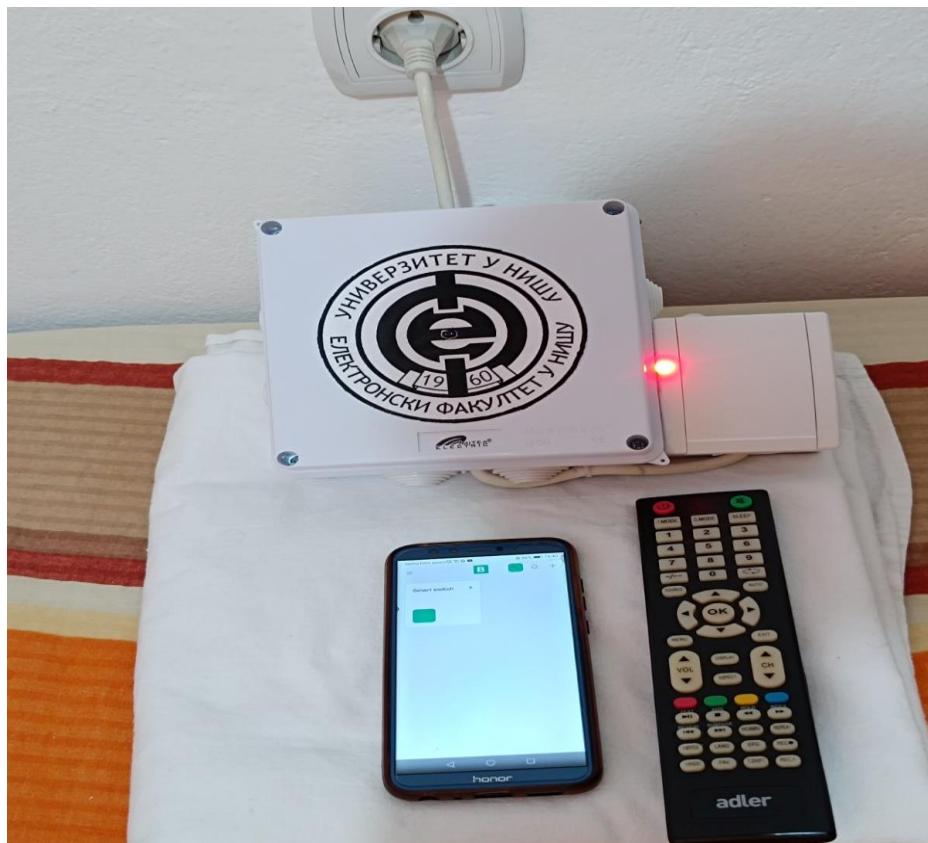


Слика 43. Коначна реализација

Са слике 43. можемо приметити присуство црвене LED диоде. Она је постављена накнадно из разлога да приликом промене стања прекидача можемо уочити промену. Измене у хадверском и софтверском делу су занемарљиве. Диода користи pin P18. Паљење диоде контролисано је путем PWM сигнала. Што значи да на местима где имамо паљење и гашење релеја требамо додати паљење и гашење диоде. Команда коју је потребно уметнути на тим местима *analogWrite(LED_PIN,0)* приликом гашења релеја и *analogWrite(LED_PIN,130)* приликом паљења релеја.



Слика 44. Угашен прекидач



Слика 45. Упален прекидач

Закључак

Овај мастер рад бавио се истраживањем, пројектовањем и реализацијом паметног прекидача. Сваки ниво активности изискивао је значајне напоре да би се дошло до оптималног решења и представљао је обједињавање целокупног знања стеченог током школовања што је и сврха мастер рада. Представљена је идеја која је значајан инжењерски изазов са више аспеката и данашњем свету нуди једно од могућих решења низа проблема које се свакодневно манифестишу, као што су проблеми са којима се сусрећемо у коришћењу кућних апаратова попут бојлера, ТА пећи, светла.....

Реализација паметног прекидача кроз три фазе овог пројекта успешно је демонстрирала примену различитих технологија за управљање електричним уређајима на даљину. У првој фази, коришћењем Arduino Uno плочице, IR пријемника и релеја, постављен је основни модел паметног прекидача који се активира даљинским управљачем. Ова фаза је пружила чврсту основу за разумевање рада релеја и њихове интеракције са микропроцесором.

Друга фаза пројекта донела је напредније функционалности користећи ESP32-S NodeMCU. У овој фази, ESP32 је конфигурисана као Access Point, омогућавајући корисницима да преко web странице контролишу релеј. Ово је значајно проширило могућности управљања уређајем, омогућавајући даљинско управљање путем Wi-Fi мреже, што је донело већу флексибилност и контролу.

Трећа фаза пројекта је представљала коначну имплементацију у реалним условима, где је креирана паметна утичица у домаћинству. Комбиновањем ESP32-S NodeMCU, IR пријемника и Blynk апликације, корисницима је омогућено да уређаје укључују и искључују путем даљинског управљача или апликације на паметном телефону. Овај интегрисани приступ омогућава не само локално, већ и удаљено управљање уређајима, чиме се постиже висок ниво удобности и ефикасности у свакодневном животу.

Све три фазе пројекта заједно представљају свеобухватан развој паметних кућних решења, од једноставног модела до сложеног система за управљање уређајима. Овај пројекат је доказао практичност и корисност коришћења модерне технологије за побољшање свакодневног живота, а резултати су показали да се оваква решења могу ефикасно применити у стварном свету. Закључак који можемо да изнесемо је да пројекат ради онако како је и замислено, а да уз мале измене може значајно да се побољша и унапреди. Главни акценат би био смањивање габарита, побољшање дизајна апликације, оптимизација потрошње по компонентама и целом систему. Врло важно је да напоменемо да је ово студентски пројекат и да је његова реализација базирана на комерцијално доступним модулима који сами по себи нису оптимални ни по потрошњи ни по габаритима. Нека коначна хардверска реализација подразумева комплетно пројектовање система од шематика до layout-а тј. до реализације штампане плочице са интегрисаним колима чиме се оптимално врши распоред компонената, управљање потрошњом и габаритност система. Овај пројекат је по нашој процени финансијски прихватљив и има могућност модификације на веома лак начин што га чини прилагодљивим.

Литература

- <https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf>
 - <https://mschoeffler.com/2021/05/01/arduino-tutorial-ir-transmitter-and-ir-receiver-hx-m121-hx-53-ky-005-ky-022-keyes-duino-open-smart/>
 - <https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet>
 - https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf
 - <https://www.waveshare.com/nodemcu-32s.htm>
 - <https://fritzing.org/download/>
 - <https://www.arduino.cc/>
 - https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
 - <https://www.electronicsforu.com/technology-trends/learn-electronics/step-down-transformer-working>
 - <https://www.geeksforgeeks.org/rectifier/>
 - <https://sr.wikipedia.org/sr-ec%D0%93%D1%80%D0%B5%D1%86%D0%BE%D0%B2%D1%81%D0%BF%D0%BE%D1%98>
 - <https://sr.wikipedia.org/sr-el%D0%98%D1%81%D0%BF%D1%80%D0%B0%D0%B0%D2%D1%99%D0%B0%D1%87>
 - <https://sr.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%83%D0%BB%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D0%BF%D0%BE%D0%BD%D0%B0>
 - https://www.analog.com/en/resources/glossary/switching_regulator.html
 - <https://shadyelectronics.com/how-a-buck-converter-works/>
 - <https://blynk.io/>