



ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ
към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

ДИПЛОМНА РАБОТА

по професия код 523050 „Техник на компютърни системи“
специалност код 5230502 „Компютърни мрежи“

Тема: Разработка на Weather Station с уеб приложение

Дипломант:

Димитър Божидаров Владимир

Дипломен ръководител:

маг. инж. Константин Заимов

СОФИЯ

2023

УВОД

1.0. Въведение в областта на IoT метеорологичните станции

IoT метеорологичните станции са вид метеорологични станции, които използват IoT технологии, за да събират, изпращат и анализират метеорологични данни. Тези системи използват мрежа от сензори и устройства свързани към интернет за достъп до въпросните данни в реално време и тяхното изпращане до централизирано място за съхранение и анализ.

IoT метеорологичните станции включват сензори за измерване на температура, влажност на въздуха, посока и скорост на вятъра, както и количество валежи и атмосферно налягане.

Информацията от сензорите се събира от устройства от типа на микроконтролерите, а след това бива изпратена на облачно-базирана платформа за съхранение и анализ. Това позволява наблюдаване на времевите условия в реално време, както и достъп към тях от всяко едно място с достъп до интернет връзка.

Събираната от станциите информация може да бъде анализирана през определен период от време като по този начин могат да наблюдават времеви тенденции, които в последствие да помогнат за образуването на прогнози и решения.

IoT станциите могат да бъдат интегрирани към други IoT устройства и системи, като “smart homes”, сгради, градове, за да се оптимизира консумацията на енергия или да се подобри безопасността или най-общо - качеството на живот.

Този тип станции обикновено са лесни за инсталирани и не са особено скъпи. Освен това те могат да се комбинират с Machine Learning алгоритми и събираните данни да бъдат включвани в големи анализи, което би допринесло за прогнозиране на бъдещи тенденции във времето.

1.1. Цели и задачи на дипломния проект

Целта на проекта е да се изгради въпросното IoT метеорологично устройство на базата на микроконтролер ESP32, който да може да бъде поставен на отдалечено място и да измерва следните данни: Температура и влажност на

въздуха, количество валежи, скорост и посока на вятъра. Измерените данни след това да бъдат записани на уеб сървър с база данни, към която да е свързано уеб приложение, където след това те да могат да се наблюдават по удобен начин.

ПЪРВА ГЛАВА

1.2. Преглед на подобни системи и продукти

Първо трябва да се знае какъв тип метеорологична станция се прави, тъй като има много различни видове такива. Те включват:

- Домашни метеорологични станции (фиг.1.1)

Домашните метеорологични станции са компактни и достъпни устройства, които позволяват на човек да наблюдава локалните данни за метеорологичните условия от удобните си домове. Тези системи включват различни сензори, които измерват основни неща като температура, влажност, атмосферно налягане, скорост и посока на вятъра, а някои по-скъпи от тях дори предлагат други измервания.

Едно от основните им предимства пред други системи е тяхното удобство. Лесно могат да бъдат инсталирани на покриви и други високи места, освен това доста модели са безжични, което елиминира нуждата от ненужно, дълго и досадно окабеляване. Веднъж монтирани, устройствата могат да изпращат информация до друго устройство с дисплей, което се намира в дома на крайния потребител, което му позволява лесно да наблюдава условията навън и да наблюдава премине в дългосрочен план.

Домашните станции могат да бъдат особено полесни на различни видове ентузиастични като например градинари, които искат да са информирани за времевите условия. Например, градинарите могат да използват информация, доставена от станцията и да по този начин да наблюдават температурата на почвата и да променят поливанията си според изискванията. Могат да планират също и активността си и да остана в безопасност при бедствени метеорологични условия.

Освен това, домашните метеорологични станции могат да доставят ценна информация за енергийната употреба в дома. Чрез проследяване на температурите на въздуха и влажността му, обитавашите могат да оптимизират

техните климатични системи, за да намалят енергийната си употреба и съответно сметките за плащане.

При избор на домашна метеорологична станция е важно да се имат в предвид фактори като точност, надеждност и лесна употреба. Някои модели могат да изискват допълнителен монтаж и поддръжка, докато други могат да бъдат максимално опростени за крайния потребител, което би ги направило по достъпни за по-голям диапазон от хора.

Домашните метеорологични станции са достъпен и ценово ефикасен начин, за собствениците на имоти да се информират за локалните метеорологични условия. Чрез доставянето на информация в реално време, обитаваните могат да взимат по-добри решения и да са подготвени за промяната във времето.

- Smart метеорологични станции

Тези метеорологични станции представляват напреднали системи за мониторинг на атмосферните условия, които използват последните технологии за сензорно измерване и свързване, за да предоставят на потребителя точна и детайлна информация за локалните метеорологични условия. Тези устройства са предназначени както за домашна, така и за комерсиална употреба и все повече набират популярност, заради своята гъвкавост, лесна употреба и удобство.

Едно от допълнителните им свойства, е че са способни да се свързват с интернет и да изпращат информацията в уеб пространството. Това означава че потребителите могат да наблюдават събираната информация в реално време от всякъде, където и да се намират стига да имат достъп до телефон, таблет или компютър и интернет. Повечето такива станции имат собствени приложения за телефон или уеб базирани табла за достъп, които позволяват на потребителите лесно и бързо да наблюдават и анализират събираната информация.

Smart метеорологичните станции също така могат да бъдат адаптирани и приспособявани, което позволява на потребители да избират сензорите и възможностите, от които имат най-голяма нужда. Освен основните стойности за

измерване, те могат да включват и по сложни сензори за наблюдаване на UV радиация, замърсяване на почвата и други параметри.

Станциите понякога също използват и изкуствен интелект (AI) и machine learning алгоритми, за да анализират и предоставят прогнози и приравнени стойности. Това се оказва особено полезно за индустрии като агрикултурата, където фермерите могат да използват информация от “умните” станции, за да взимат адекватни решения за наторяване на почвите, борба с вредители и други.

Освен това, някои станции могат да се свързват към други “умни” устройства, като например системи за поливане, за да оформят автоматизирани графици за поливане на разтенията, изцяло на базата на събраната в реално време информация.

“Умните” метеорологични станции са мощни инструменти за всеки един човек, който има нужда от информацията, която предоставят без значение дали се занимават професионално или любителски.

- Професионални метеорологични станции (фиг.1.2)

Професионалните метеорологични станции са сложни системи, създадени за измерване и анализиране на широка гама от атмосферни параметри. Тези устройства често се използват от метеоролози, климатолози, експерти по агрикултурата и други професионалисти, които разчитат на тяхната точност в сферата на измерванията.

За разлика от станциите от предишните класове, професионалните модели са високо технологични устройства, които често се правят по нуждите на крайния потребител. Обикновено включват доста повече сензори, които освен обичайните стойности могат да измерват и други като например слънчевата радиация и други природни променливи.

Професионалните метеорологични станции често се използват в изследвания, които се нуждаят от висока точност и постоянност. Често се използват в индустрии като агрикултурата като точното измерване на метеорологичните стойности може да помага за вземането на най-различни

решения както и в авиацията, където актуалната информация е неизменна част от безопасните и ефективни летателни операции.

Освен, че доставят на крайния потребител актуална информация, много професионални станции могат да се настроят да предоставят исторически данни и да анализират промени, които се повтарят. Това може да бъде полезно за проследяване на промени в метеорологичните промени за различни периоди от време и дългосрочното наблюдаване на повторения и аномалии.

При избора на професионални метеорологични станции е важно да се намесят фактори като точност, надеждност и лесна употреба. Някои марки и модели могат да изискват специална подготовка и обучение или експертни нива в областта, за да се оперират ефективно, докато други такива устройства могат да бъдат особено лесни за употреба от крайния потребител, което ги прави достъпни за по-голяма част от хората.

Професионалните метеорологични станции са неизменен инструмент за широк набор от професионалисти, които разчитат на точно измерена информация за работата си.

- Отдалечени метеорологични станции (фиг.1.3)

Тези станции се изграждат или поставят на места, които са твърде далеч или са трудно достъпни за малък период от време. Изграждат се от най-различни сензори, които измерват влажността на въздуха, скоростта и посоката на вятъра, атмосферното налягане, количеството валежи и други метеорологични променливи. Тези сензори са свързани към система за записване на данни, които записват информацията през определени интервали, обикновено през няколко минути.

Информацията отчетена от отдалечените метеорологични станции се изпраща и записва в централен сървър или база данни през различни методи, които може да включват сателити, мобилни и радио мрежи. Тази информация след това се използва от учени и други професионалисти за задълбочено изучаване на повтарящи се метеорологични събития и предвиждане на промени в климата.

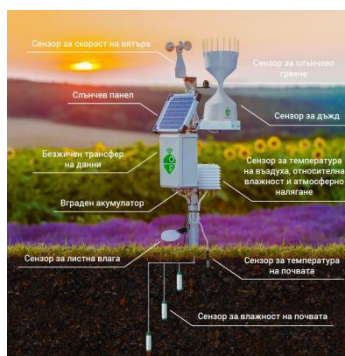
Могат да се използват за най-различни цели, от мониторинг на промяната в климата до подобряване на агрикултурата и разпределението на водите. Също така са полезни за получаване на информация в реално време на екипи за бързо реагиране при случаи на бедствия като урагани, наводнения и пожари.

Въпреки всички тези неща, които могат да извършват, имат недостатъци като например, че не винаги могат да забележат метеорологични условия, които се случват на морското равнище като мъгла и облаци, които са ниско. Освен това отдалечената им локация ги прави трудни за поддръжка и може да станат жертва на природата, например сензорите могат да замръзнат.

Отдалечените метеорологични системи са ценен инструмент за събиране и доставяне на метеорологична информация на места, които не са достатъчно практични или достатъчно лесни за достъпване от човека. С напредването на технологиите станциите се подобряват, а с тях и точността и надеждността на измерванията им.



Фигура 1.1



Фигура 1.2



Фигура 1.3

1.3. Преглед на технологиите, които се използват в метеорологичните станции

Метеорологичните станции използват доста различни технологии за измерване и записване на отчетените данни. В това число влизат сензори за измерване на температура, влажност на въздуха, скорост и посока на вятъра,

сензори за количество валежи, както и за налягане на въздуха. Някои метеорологични станции са снабдени и със сензори за слънчева и UV радиация.

Използват се анемометри и ветропоказатели за измерване на скоростта и посока на вятъра. За температура и влажност на въздуха се използват сензори за температура и влажност, като например термистори (терморезистори) и влагомери. За измерване на валежите се използват технологии от сорта на “Tipping Bucket” (фиг.1.4 и фиг1.5), а за измерване на атмосферното налягане се използват барометри. (фиг.1.6)

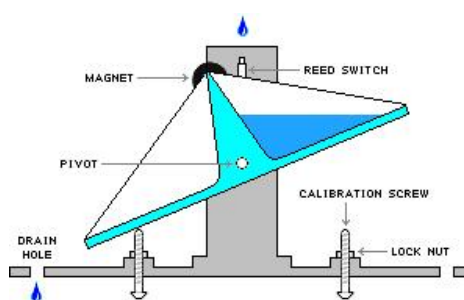
Информацията от описаните по-горе сензори се изпраща на централизиран дисплей или компютър с жична или безжична връзка. Някои метеорологични станции също така могат да изпращат и информация към отдалечени места посредством интернет или мобилни мрежи.

Освен традиционните сензори, някои модерни станции също така включват камери, сензори за светлина и други. Тези сензори се използват за допълнителна информация, която може да се използва за подобряване на точността на прогнозите за времеви условия.

Данните от метеорологичните станции могат да бъдат използвани по ред причини включващи: прогнози, изучаване на климата, както и аграрни и индустриални нужди.



Фигура 1.4



Фигура 1.5

1.4. Преглед на устройствата, използвани в дипломния проект

1.4.1. Избор на микроконтролер за изпълнение на дипломния проект

- ESP32 (фигура 1.6)

ESP32 е мощен микроконтролер с Wi-Fi модул, който е създаден за Internet of Things (IoT) приложения. Произвежда се от Espressif Systems и в последно време се е превърнал в популярен избор измежду разработчиците, заради достъпността и производителността си.

Едно от главните неща, които микроконтролера притежава е двукратния му процесор, който позволява висока производителност, докато едновременно с това спазва нисък енергиен разход. Освен това има вграден Wi-Fi модул, който поддържа много различни мрежови протоколи, включително TCP/IP, SSL и HTTP, което го прави идеален за IoT приложения и проекти, които се нуждаят от безжично свързване.

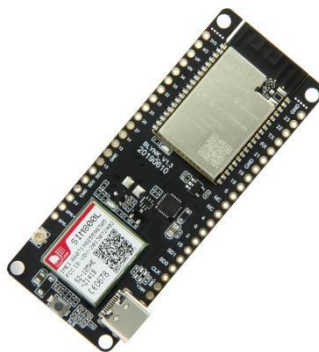
ESP32 също така има широк набор от input/output (I/O) пинове, които могат да се използват за свързване на различни сензори и периферии. Също така поддържа различни комуникационни протоколи включително SPI, I2C, UART и CAN, което го прави подходящ за доста различни видове проекти.

Едно от уникалните му предимства е поддръжката на Bluetooth Low Energy (BLE), което му позволява да се свързва към широк набор от BLE устройства като смартфони, фитнес уреди и други IoT устройства. Освен това поддържа Near Field Communication (NFC), което му позволява да контактува с други NFC устройства, като например пунктове за безконтактно електронно плащане.

ESP32 също може да бъде програмиран на много компютърни езици, които включват C++, Python и JavaScript. Освен това поддържа много системи и среди за разработка като например Arduino IDE и Espressif IoT Development Framework (ESP-IDF).

За цялостната изработка на устройството от проекта се използва версия на ESP32 наречена SIM800L, която поддържа мобилни мрежи, чрез SIM карта, което е перфектно за приложението и планираната отдалеченост на системата. За експеримента обаче ще се използва нормално ESP32 свързано чрез Wi-Fi, понеже средата ще бъде локална освен това SIM изпълнението на

микроконтролера не е толкова бюджетно, което го прави нерентабилно за експеримента.



Фигура 1.6 - Микроконтролер ESP32WROOM

- ESP8266 (фигура 1.13)

ESP8266 е бюджетен микроконтролер с поддръжка за Wi-Fi, който е широко използван в Internet of Things (IoT) и други проекти, които използват безжична връзка. Произвежда се от Espressif Systems е първата му версия излиза през 2014 година.

Едно от основните му предимства пред другите е възможността за безжично свързване чрез Wi-Fi протокол. ESP8266 може да се използва самостоятелно или като Wi-Fi модул, който може да бъде интегриран към други системи.

Освен това е много практичен и сравнително мощен микроконтролер с 32 битов процесор и работна честота до 80 MHz. Има 512 KB оперативна памет и до 16 MB вградена флаш памет за запазване на код. Това го прави подходящ избор за широка гама от приложения, от просто четене на информация от сензор до сложно IoT устройства.

Още едно предимство на този микроконтролер е лесната му употреба. Може да се програмира на различни езици, включително C++, Lua и MicroPython. Освен това има много налични библиотеки за неща, които често се разработват, което го прави лесен и бърз за научаване.

Освен способността си за безжична връзка и мощни изчислителни умения, ESP8266 е и доста евтин микроконтролер. Това го прави популярен избор между

ентусиастите и професионалистите, които искат да разработват IoT устройства, но не искат да си приготвят голям бюджет.

Финално решение за микроконтролер, който измерва влажността във въздуха на устройството на дипломния проект:

За изпълнение на технологичните изисквания в дипломния проект избирам да използвам микроконтролера ESP32 на Espressif Systems. Причината е че за множеството сензори, които използвам имам нужда от по-висока производителност. Освен това за крайното устройство (различаващо се от това на експеримента) е необходима SIM връзка за отдалеченият достъп до интернет, който само специална версия на ESP32 поддържа, което би направило използването на микроконтролер като ESP8266 невъзможно за апликацията.

1.4.2. Избор на сензор за температура и влажност на въздуха

- DHT11 (фигура 1.7)

DHT11 е бюджетен, цифров сензор за измерване на температура и влажност във въздуха, който често се използва в широк набор от приложения в сферите на автоматизирането на дома, метеорологичните наблюдения и вътрешни системи за климатизация. Произвежда се от Aosong Electronics и е популярен избор измежду любители и професионалисти в областите изброени по-горе заради простотата, точността и ниската си цена.

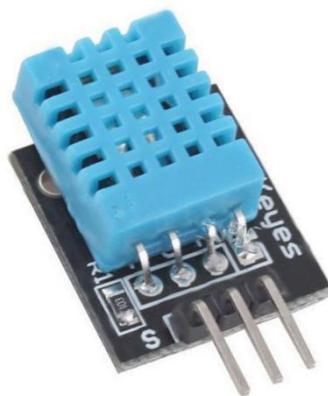
Едно от удобствата му е, че може да измерва температура и влажност във въздуха с един сензор. Използва кондензаторен сензор за измерването на влажността, а температурата измерва с термистор, след което тази информация се преобразува в цифров сигнал, който може да се прочете от микроконтролер или друго изчислително устройство.

DHT11 има диапазон на температурно измерване от 0°C до 50°C с толеранс от +/-2°C, докато влажността във въздуха с диапазон от 20% до 90% с толеранс от +/-5%. Докато тези нива на точност може да не са толкова високи, колкото при други сензори, DHT11 често се избира заради лесната си употреба и ниската цена.

DHT11 освен това е лесен за употреба, поради единичният си проводников интерфейс.

Това означава, че лесно може да се интегрира в много различни проекти, освен това има готови библиотеки за популярни микроконтролери като Arduino и Raspberry Pi.

Освен лесната си употреба и ниската си цена, сензора също така използва много малко енергия (1mA електричество по време на активна употреба). Това го прави идеален за устройства, които се захранват от батерии, като например безжични системи за измерване на температура и влажност във въздуха.



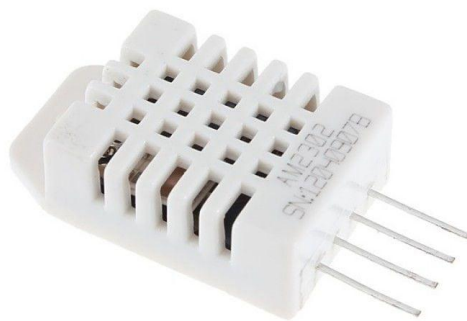
Фигура 1.7 - Сензор за измерване на влажността във въздуха и температурата - DHT11

- DHT22 (фигура 1.9)

DHT22 е дигитален сензор за измерване на температура и влажност във въздуха, който се използва в много IoT приложения и устройства. Другото му наименование е сензор AM2302 и е наследник на DHT11, описан по-горе. Разработен е да измерва същата метеорологична информация като DHT11, но с по-голяма точност в различни среди.

Сензорът използва сериен интерфейс с 2 проводника, за да доставя данните на микроконтролера/компютъра. Има 4 пина, 2 от които се използват за захранването, а другите 2 се използват за пренасяне на събираната информация. Сензорът работи на напрежение между 3.3V и 5V и изразходва малко енергия, което го прави перфектен за устройства, които се захранват чрез батерия.

Едно от предимствата на DHT22 е високата му точност и нисък толеранс на измерване на температура и влажност на въздуха. Сензорът има диапазон на измерване на температура от -40°C до 80°C с точност от $\pm 0.5^{\circ}\text{C}$ и диапазон на измерване на влажността във въздуха от 0% до 100% с точност от $\pm 2\%$.



Фигура 1.9 - Сензор за измерване на температура и влага във въздуха DHT22

Финално решение за сензор, който измерва влажността във въздуха на устройството на дипломния проект:

За изпълнение на технологичните изисквания в дипломния проект избирам сензор за измерване на влажността във въздуха DHT11, понеже нямам нужда от температурния модул, а разликата в измерването на влажността не е достатъчна, за да оправдае сравнително по-голямата цена на DHT22.

1.4.3. Избор на сензор за атмосферно налягане и температура - BMP180 (фигура 1.8)

BMP180 е бюджетен сензор с висока точност, цифров баромер и температурен сензор с ниска консумация на енергийна мощност, който може да бъде използван за широк набор от приложения, включително прогнозиране на времето, измерване на надморска височина и навигация на дроне. Произвежда

се от Bosch Sensortec и е популярен избор между любители и професионалисти, заради точността, надеждността и лесната си употреба.

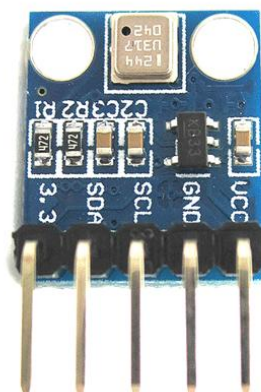
Едно от ключовите предимства на сензора е способността му да измерва едновременно атмосферното налягане и температурата с висока точност. Използва MEMS (micro-electro-mechanical-system) сензор, за да наблюдава за промени в стойностите, които измерва и след това преобразува информацията в цифрови сигнали, която микроконтролер може да прочете и използва.

Сензора има диапазон на измерване от 300 до 1100 hPa, с толеранс от ± 1 hPa. Това го прави използваем за много приложения от наблюдаване на метеорологичните условия до измерване на височина. Диапазонът на температурно измерване е от -40°C до $+85^{\circ}\text{C}$ с толеранс от $\pm 1^{\circ}\text{C}$.

Освен това BMP180 е лесен сензор за употреба, комуникира чрез стандартен I2C (Inter-Integrated Circuit) интерфейс. Това означава че може лесно да бъде интегриран в много различни проекти, а освен това съществуват и много библиотеки за популярни микроконтролери като Arduino и Raspberry Pi.

Към точността и лесната си употреба, сензора използва много малко електроенергия, консумирайки само $0.5\mu\text{A}$ ток в standby режим. Това го прави идеален избор за отдалечени устройства, които използват преносимо захранване (батерии).

BMP180 е удобен и надежден сензор, който може да се използва за най-различни приложения от наблюдаване на времето, до измерване на надморска височина. Точността, лесната му употреба и ниската му електро консумация го правят известен избор между всички в областта.



BMP280 (фигура 1.12)

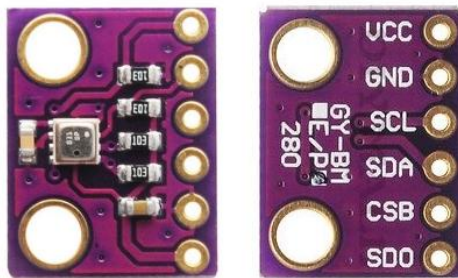
BMP280 е дигитален барометричен сензор с висока точност, който може да измерва температура и атмосферно налягане и се използва в широк набор от приложения, включително системи за наблюдаване на метеорологичните условия, контрол на височината при дроновете и следене на качеството на въздуха на закрито. Произвежда се от Bosch Sensortec и е представлява подобрена версия на BMP180.

Едно от главните предимства на сензора е способността му да измерва едновременно атмосферно налягане и температура с висока точност. MEMS сензорът и сензорът за температура се използват за следене за разлика в измерваните стойности, след това събраната информация се преобразува в дигитален сигнал и се изпраща на микроконтролера или компютъра.

BMP280 има диапазон на измерване на атмосферно налягане от 300 hPa до 1100 hPa, с точност от ± 1 hPa и температурен диапазон от -40°C до 85°C с точност от $\pm 1^{\circ}\text{C}$. Тези високи нива на точност и големи диапазони на измерване превръшат сензора в универсален за почти всеки един проект, който се нуждае от прецизно измерване на температура и атмосферно налягане.

Сензорът освен това се използва сравнително лесно, понеже комуникира чрез I2C или SPI интерфейс. Това означава че лесно може да се интегрира в широка гама от проекти, както и че съществуват библиотеки за популярни микроконтролери в областта като Arduino и Raspberry Pi.

Освен високата си точност, той използва изключително малко енергия, консумирайки само μA ток по време на активна употреба. Това го прави идеален сензор за използване в проекти, които се захранват с батерия, като отдалечени метеорологични станции.



BMP280

Фигура 1.12 - Сензор за температура и атмосферно налягане BMP280.

Финално решение за сензор, който измерва атмосферното налягане и температурата за дипломния проект:

За изпълнение на технологичните изисквания в дипломния проект избирам сензор за атмосферно налягане и температура BMP180, понеже в този прототипен вариант на устройството BMP280 не е рентабилен избор освен това единствената огромна разлика между двата сензора е температурният диапазон на BMP280, който е по-голям едва с 5°C, откъм положителната страна на температурата.

1.4.4. Избор на сензор за скорост и посока на вятъра

- КИТ Анемометър - COM-B020

- За измерването на скоростта и посока на вятъра необходими за правилната работа на устройството за проекта е избран специален комплект, който съдържа 2 устройства - Анемометър и Ветропоказател. Двете устройства се свързват външно всяко чрез RJ11 конектор. Ветропоказателят работи на принципа на променливото съпротивление. Когато показателят се върти, той закача определени проводници с различни съпротивления свързани към тях. Според различното напрежение, регулирано от съпротивлението стойностите са различни и след калибрация може да се изчисли посоката на вятъра.

1.5. Преглед на технологиите, използвани в дипломния проект

- Уеб сървър - Apache HTTP Server

Apache - както се нарича накратко е безплатен, open-source уеб сървър. Използва се главно за издигане на уеб сайтове и уеб приложения. Първата му версия е представена през 1995 година и оттогава е един от най-използваните уеб сървъри в Света.

Известен е със своята висока производителност, стабилност и гъвкавост. Способен е да представя едновременно статично и динамично съдържание и може да бъде конфигуриран да работи с голяма гама от други програмни езици и платформи като PHP, Python и Java.

Едно от основните предимства на Apache е модуларната му архитектура, която позволява на разработчиците да добавят или премахват функционалности ако имат нужда. Това прави възможно приспособяването на сървъра с различни ситуации, за да отговаря на специфични нужди, като например подобрена производителност или подобрена сигурност.

Още едно предимство на Apache е подробната документация и активната общност, която го използва. Има много ресурси в уеб пространството, които са достъпни до разработчиците за помощ как да използват и модифицират сървъра, също така има и голям набор от хора, които го използват, които могат да помагат един на друг.

Apache Web Server е мощен и надежден софтуер за уеб сървър, който до момента е изиграл значима роля в разработката на модерното уеб пространство. Гъвкавостта, модуларността и активната му общност го правят чест избор за разработчици от всякакви нива с всякакви умения.

- База данни - MariaDB

MariaDB е популярна open-source система за управление на данни, която е била създадена от общността с потребители използващи системата за управление на данни MySQL. Първата версия излиза през 2009 година и

оттогава се е превърнала в често използвано решение за бази данни за уеб приложения и други видове софтуер.

MariaDB е известна със своята висока производителност, възможност за уголемяване и средства за сигурност. Поддържа широк набор от програмни езици и платформи, което я прави популярна система за изграждане на уеб приложения, които се нуждаят от бърз, сигурен и надежден достъп до данни.

Един от големите плюсове на системата е нейната съвместимост с MySQL, което означава че приложения и инструменти създадени за MySQL могат да бъдат използвани без особени промени. Това улеснява разработчиците да мигрират своите системи от MySQL към MariaDB без проблеми със съвместимостта.

Още едно предимство на MariaDB, е че системата е фокусирана върху това тя да се разработва от общността и - именно хората, които я използват. Именно това предимство помага на системата да остане актуална до ден днешен, да има чести актуализации, в които се оправят бъгове и проблеми в сигурността.

MariaDB е мощна и сигурна система за управление на данни, която е и популярен избор за уеб разработчици и софтуерни инженери. Нейната висока производителност я прави идеален избор за създаването на уеб приложение, които изискват бърз и надежден достъп до данни.

1.6. Преглед на програмните езици, използвани в проекта

- HTML

HTML (Hypertext Markup Language) е стандартен програмен език, който се използва за създаването на уеб страници. Първата му версия е представен през 1993 година и оттогава той се е превърнал в неизменна част от уеб разработката.

Използва се за структуриране на съдържанието в уеб страници чрез дефиниране на различни елементи, които се използват в създаването на една страница. Тези елементи могат да включват заглавия, параграфи, таблици,

снимки, списъци и други видове медия. HTML е markup език, което означава че използва специални тагове и атрибути за дефиниране на съдържанието.

Основното му предимство е колко лесно се използва и колко е елементарен. Изучава се доста бързо и може да бъде писан в абсолютно всяка програма за писане на текст. Освен това той е стандартен компютърен език, което означава че разработчиците могат да са сигурни, че техният код ще бъде интерпретиран и показан правилно във всеки един уеб браузър.

Освен това той поставя основа за други езици за разработка в уеб пространството като CSS и JavaScript. Чрез използването на HTML с другите езици, разработчиците могат да създават много итерактивни и ангажиращи уеб среди, които са достъпни и оптимизирани измежду различни платформи.

HTML е неизменен инструмент за уеб разработка, който се занимава с основната структура и съдържание на уеб страница. Елементарността му и лесната му употреба както и съвместимостта му с други технологии за разработка на уеб среди го превръщат в задължителен език за всеки човек, който иска да се занимава с уеб разработка и уеб базирани приложения.

- CSS

CSS (Cascading Style Sheets) е компютърен език, който се използва, за да се дефинира визуалното разположение и дизайн на една уеб страница. Това е език, който описва как HTML елементите в една страница да бъдат разположени. Първата му версия излиза през 1996 година и оттогава се е превърнал в съществена част от уеб разработката.

Едно от основните му предимства, е че той е способен да разделя съдържанието на една страница от презентирането и. Това означава, че уеб разработчиците могат да използват HTML, за да структурират съдържанието на една уеб страница, а след това да използват CSS, за да дефинират визуалната подредба и стилизирането. Това разделяне на работата улеснява поддръжката и актуализациите на уеб страниците, понеже по този начин дизайна може да бъде променен независимо от съдържанието.

CSS предлага широк набор от стилови варианти, включително и възможности за промяна на шрифта, цветовете, фона, границите и позициите на елементите на една страница. Освен това той предлага и различни варианти за разположението, които позволяват много лесно сложни уебсайтове да се приспособяват към различни размери дисплеи и устройства.

Още едно предимство на CSS е, че един отделен CSS файл може да бъде приложен към изключително много страници, което като цяло улеснява много поддръжката и създаването на страниците и спестява доста работа.

Като цяло CSS е съществено важен инструмент за съвременната уеб разработка. Неговите предимства, описани по-горе го правят ефективен начин за създаване на красиви и приспособяеми уеб страници.

- PHP

PHP (Hypertext Preprocessor) е сървърен програмен език, който се използва за създаването и разработката на уеб приложения. Първата му версия излиза през 1995 година и оттогава езикът се превръща в един от най-използваните за уеб разработка. PHP е open-source език, което означава, че разработчиците могат да го променят, за да изпълняват нуждите си без ограничения.

Един от големите плюсове на PHP е неговата многофункционалност. Той може да се използва за широка гама от задачи в уеб разработката - от създаването на простички статични уебсайтове до създаването на сложни уеб приложения, които работят с бази данни и други уеб услуги. Допълнително, PHP е лесен език за научаване и използване, което го прави чест избор при прощаващи разработчици, които искат бързо да започнат да пишат скриптове.

Още едно предимство на PHP е голямата общност от разработчици и ресурси, които са налични. Има много форуми в уеб пространството, както и ресурси, които програмистите могат да използват, за да намират решения на проблемите си, да научават нови неща, както и да черпят идеи за своите проекти. Освен това има много предварително създадени библиотеки и framework-ове,

които са достъпни и могат да забързат времето за разработка и да улеснят процеса по създаването на един проект.

- C

C е общоизползван програмен език, който е често използван за разработката на операционни системи, embedded устройства и други софтуерни приложения. Първоначално е разработен през 70-те години на 20-ти век от Dennis Ritchie в Bell Labs и оттогава се е превърнал в един от най-използваните програмни езици в света.

Един от основните му плюсове пред другите езици е неговият нисък разход на оперативна памет, което позволява висока ефективност и бързина в представянето. Често се използва в системното програмиране, понеже има достъп до ниските слоеве на хардуера и е способен да работи директно с операционната система на устройството, на което се използва.

C също така се избира заради това, че е сравнително елементарен език и е стандартизиран език, който може да бъде компилиран и изпълняван на много различни платформи. Синтаксисът му е лесен за разбиране и това го прави популярен избор както за навлизащи, така и за напреднали програмисти.

Въпреки че C съществува от над 50 години, той остава популярен език за разработка на нови софтуерни приложения. Ниският му разход на ресурси и хардуерна мощ, комбинирани с неговата простота го превръщат в силен инструмент за най-различни програмни задачи и проекти.

- JavaScript

JavaScript е динамичен, програмен език, който често се използва за уеб разработка. Може да се използва едновременно за front-end и back-end разработки. Използва се главно за създаването на интерактивни потребителски интерфейси, динамично уеб съдържание и за манипулиране на информация от страната на клиента.

След разработката му през 1995, езикът се превръща един от най-популярните езици в света и бива използван от милиони разработчици за най-

различни цели. Освен за уеб приложения, той често бива използван за разработка на desktop и мобилни приложения, както и server-side приложения и IoT устройства.

Едно от главните му предимства при работа с HTML и CSS, другите две основни технологии на уеб пространството, е че разработчиците могат динамично да манипулират съдържанието на уеб страници, да отговарят по определен начин на client-side заявки, както и да извършват сложни операции с данни без нуждата от презареждане на страницата. Тези интерактивност и удобен дизайн са огромна част от популярността на езика в създаването на модерни уеб приложения.

Генерално, JavaScript е гъвкав език, който се е превърнал в почти задължителен инструмент за уеб разработчиците по целия свят. Практичността и лесната му употреба са го превърнали в харесван език между всички уеб дизайните, а продължителната му разработка и поддръжка го държат в уеб разработката през последните 2 десетилетия.

ВТОРА ГЛАВА

2.1. Функционални изисквания към дипломния проект

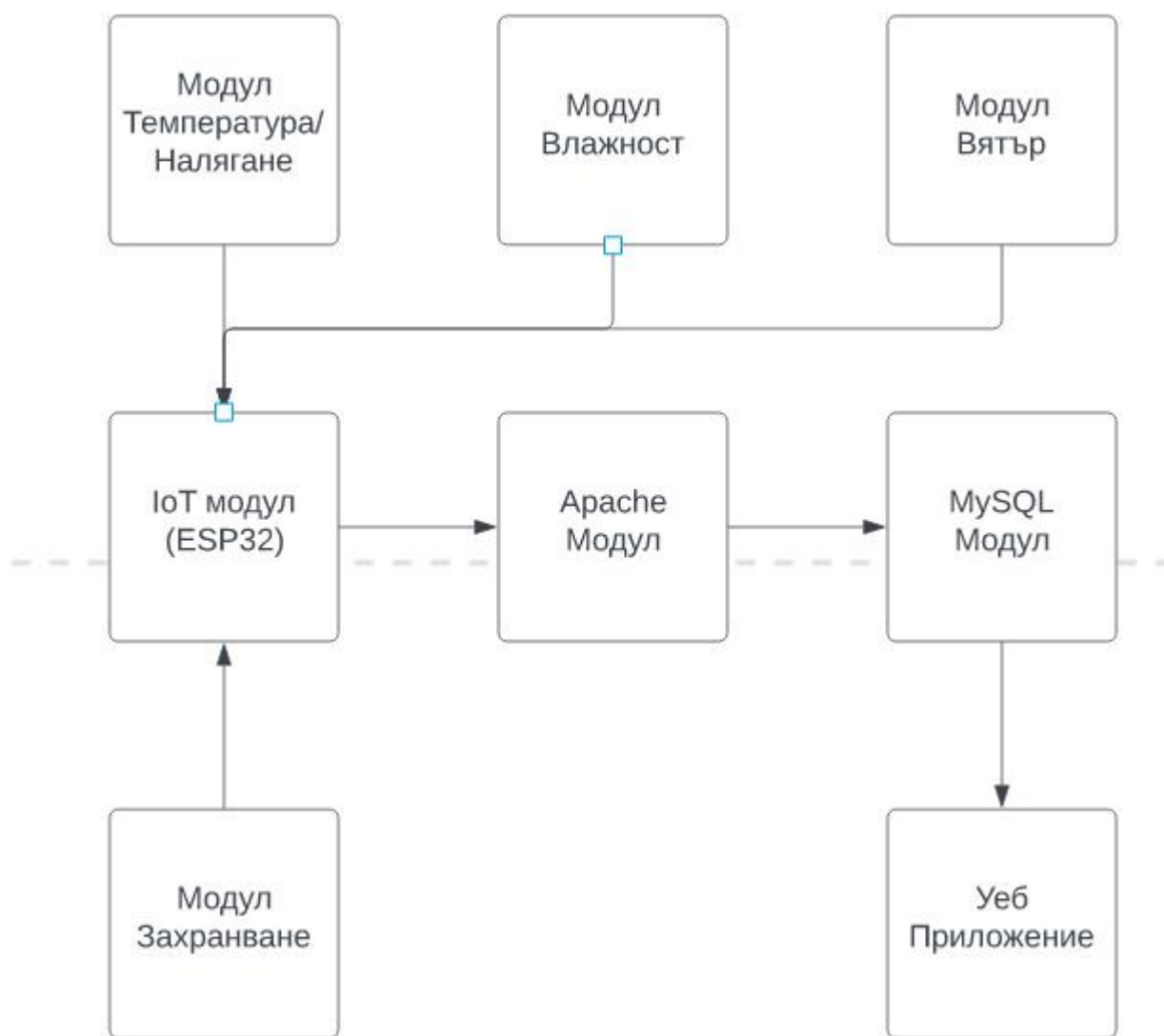
- Устройството трябва да събира метеорологична информация за следните параметри: температура, атмосферно налягане, влажност на въздуха, скорост на вятъра, посока на вятъра посредством посочените в този документ сензори и елементи.

- След това събраната информация се избира до MySQL сървър чрез интернет пространството, където се съхранява информация за всеки един астрономически час от последната 1 седмица (7 дни).

- Вече събраната информация се изобразява в табличен вид в приложение, което се изпълнява от уеб сървър. Приложението също така съдържа препратки към административното меню на базата данни. Всеки един вид информация - без посоката на вятъра, може да бъде избран отделно и да се наблюдават най-ниските, най-високите и средните стойности за последните 24 часа.

- Приложението поддържа до 2 отделни метеорологични станции.

2.2. Проектиране на блоковата схема на устройството за метеорологични измервания



Фигура 2.1 - Блокова Схема на устройството

2.3. Разглеждане на отделните модули от блоковата схема (фиг 2.1)

- Модул Захранване

Този модул представлява захранването на самото устройство. ESP32 поддържа usb захранване, което е удобно, но за нуждите на този проект е най-вероятно да се използва батерия с голям капацитет, която ще бъде зареждана соларно, за да може да се улесни отдалечената поддръжка.

- IoT модул (ESP32)

Този модул представлява мозъка на проекта. По план там се намира ESP32 SIM 800L, за да може нашата станция да се свързва отдалечено през мобилна мрежа посредством SIM карта с достъп до интернет.

За експеримента се използва нормално ESP32 с WiFi връзка, понеже в такава среда няма нужда от мобилна мрежа.

- Модул Температура/Налягане

Този модул съдържа елемента BMP 180, прегледан в Първата глава на този документ. Той измерва температура и атмосферно налягане и ги изпраща на IoT модула (ESP32).

- Модул Влажност на въздуха

Този модул съдържа елемента DHT11, който измерва влажността във въздуха и изпраща информацията в IoT модула (ESP32).

- Модул Вятър

Модулът се реализира на базата на анемометър, който съответно отчита скоростта и посоката на вятъра и изпраща информацията в IoT модула (ESP32).

- Apache модул

Този модул е първият, който е далеч от устройството. След като IoT модула осъществи връзка с интернет, той се свързва към уеб сървър посредством интернет адрес, потребителско име и парола.

- MySQL модул

След като връзката с уеб сървър бива осъществена вече устройството може да записва информация в базата данни с информацията посочена по - горе. Там се достъпва база данни с име “weatherstation”, в която се записва

информация в таблицата 'weather_data', която съдържа отделни колони за всяка една стойност, която нашата станция измерва.

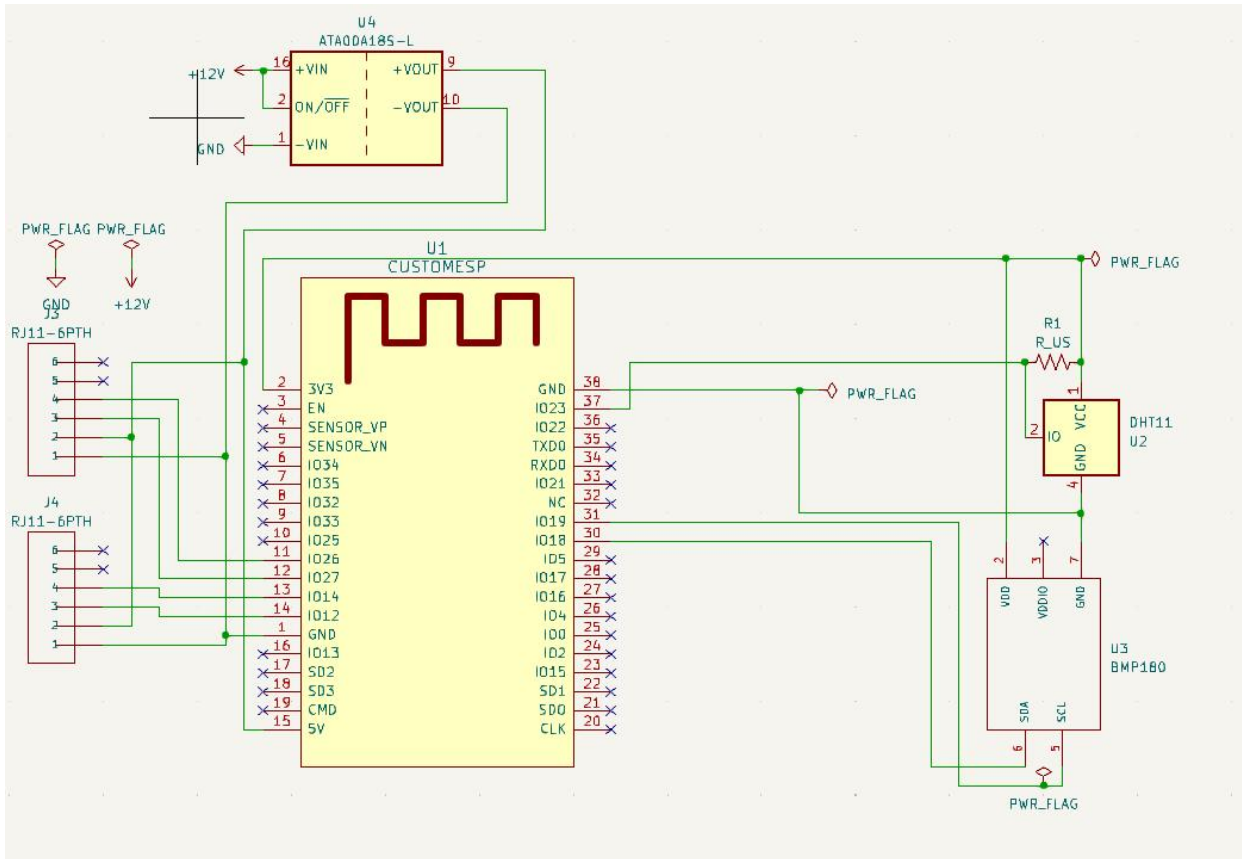
- Уеб Приложение

След като информацията от сензорите е записана в MySQL сървъра и вече е достъпна в интернет пространството, тя вече се показва и в уеб приложението чрез PHP скрипт, чийто изход е графично обработен посредством HTML, CSS и JavaScript.

ТРЕТА ГЛАВА

ПРОЕКТИРАНЕ НА ПРИНЦИПНА ЕЛЕКТРИЧЕСКА СХЕМА НА УСТРОЙСТВОТО

3.1 Принципна електрическа схема на устройството за измерване на метеорологични условия (фигура 3.1)



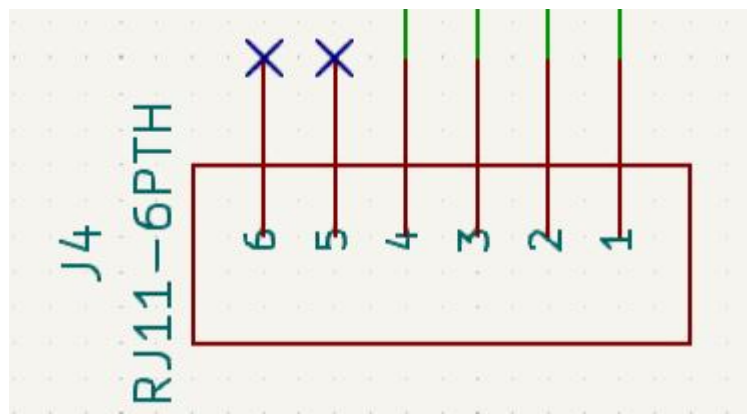
Фигура 3.1 - Принципна електрическа схема на устройството

Принципната електрическа схема на устройството е начертана и разработена на KICAD.

3.2. Разглеждане на елементите от електрическата схема

3.2.1. Конектори използвани в проекта

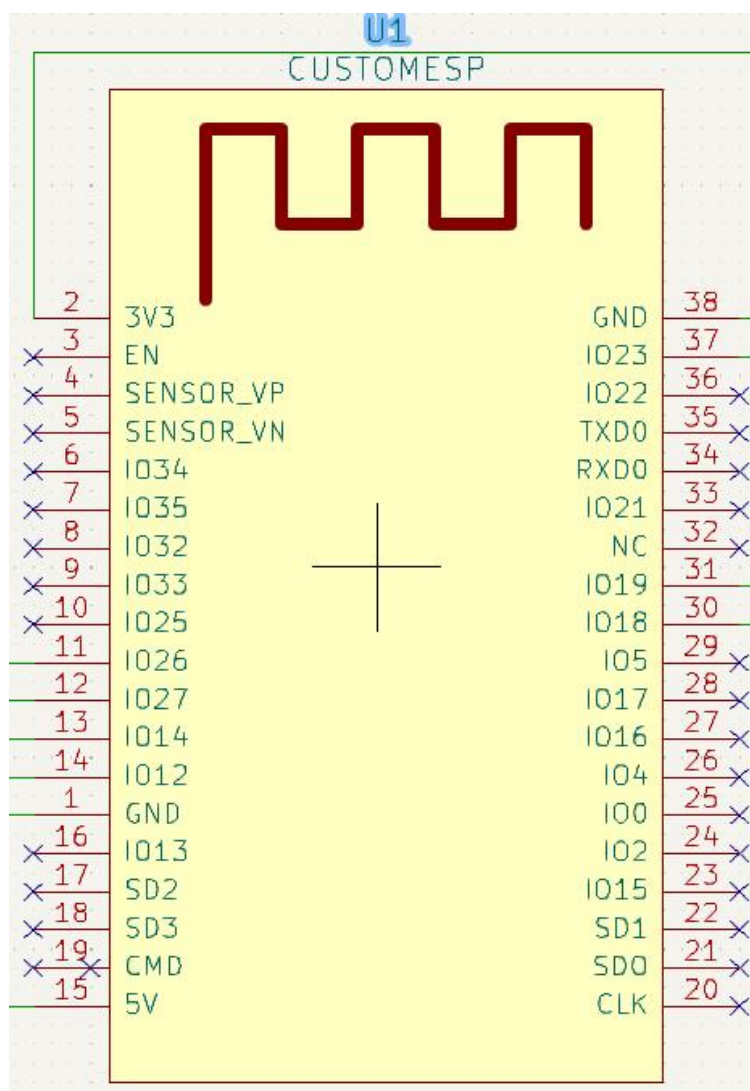
- **J3, J4** - RJ11 (Фигура 3.2.1.1) конектори за връзка на уредите за измерване на скорост и посока на вятъра



Фигура 3.2.1.1 - RJ11 Конектор за връзка на уредите за измерване на стойностите на вятъра

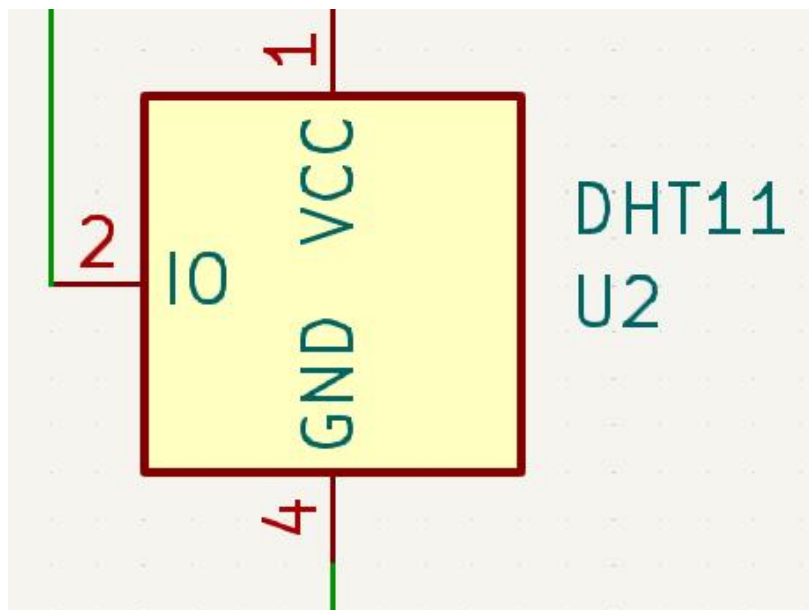
3.2.2. Интегрални схеми използвани в проекта

- U1 - ESP32WROOM (Фигура 3.2.2.1) - Основен микроконтролер използван за разработката на дипломният проект и устройството.



Фигура 3.2.2.1 - Микроконтролер ESP32

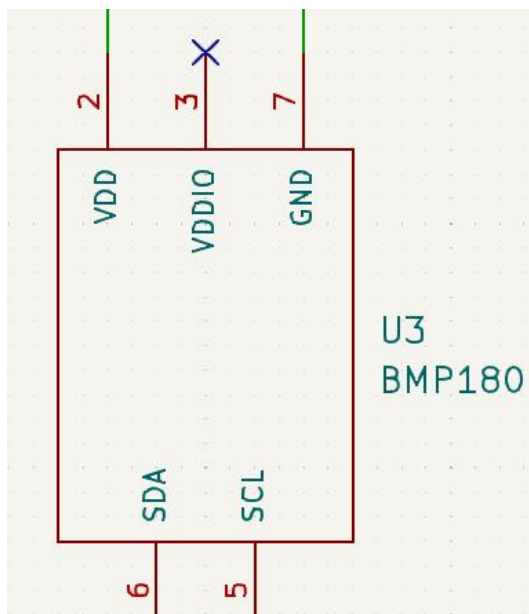
- **U2** - DHT11 (Фигура 3.2.2.2) - Цифров сензор за измерване на влажността на въздуха и температурата. В този проект се използва само измерването на влажността, понеже BMP180 се използва за измерване на температурата.



Фигура 3.2.2.2 - Сензор за измерване на влажността на въздуха DHT11

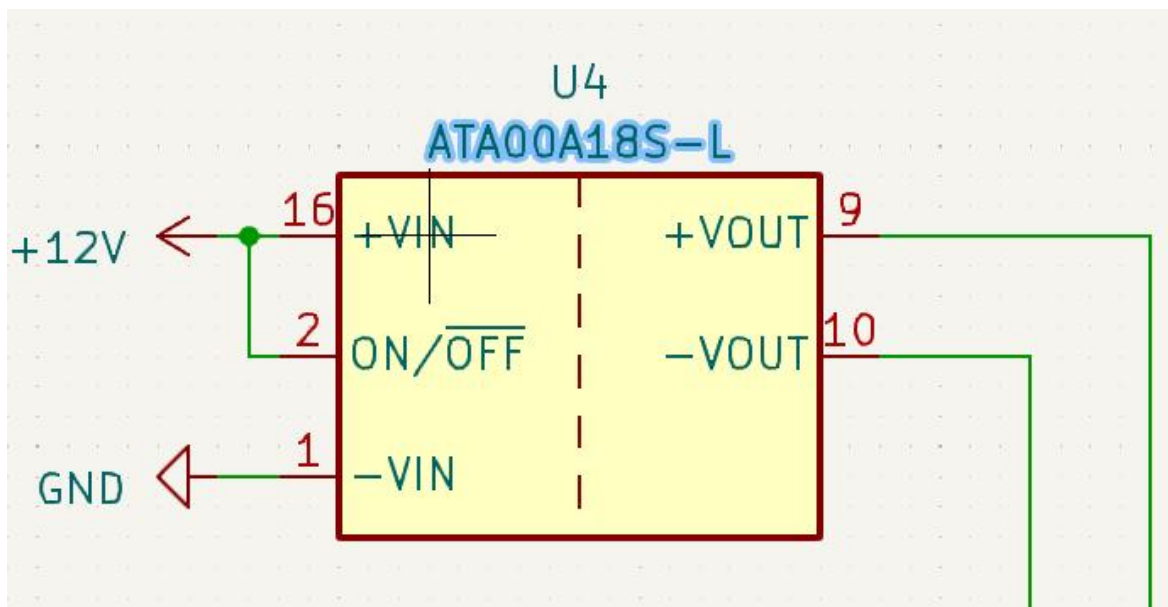
- **U3** - BMP180 (Фигура 3.2.2.3) - Цифров сензор за измерване на температура и атмосферно налягане.

В проекта се използват и двете стойности, които измерва.



Фигура 3.2.2.3 - Сензор за измерване на темп. и атмосферно налягане BMP180

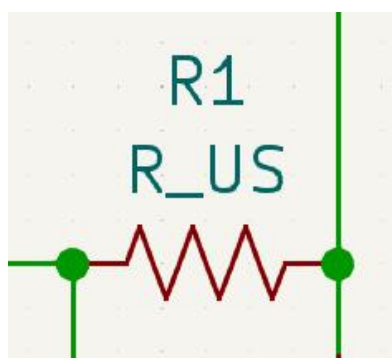
- **U4** - ATA00A18S-L (Фигура 3.2.2.4) - Делител на напрежение, който променя входното напрежение от 12V на 5V, които отиват директно във 5V входа на U1 (ESP32).



Фигура 3.2.2.4 - Делител на напрежение от 12V Input към 5V Output

3.2.3. Градивни елементи използвани в проекта

- **R1** (Фигура 3.2.3.1) - Резистор, който доставя съпротивление необходимо във връзката между захранването на проекта и информационния пин на микроконтролера, от който се чете информацията за влажност на въздуха от U2 - DHT11.



Фигура 3.2.3.1 - Резистор между захранването и инфо. пин на ESP32

3.3. Проверка на принципната електрическа схема на проекта за грешки

- **ERC** - Инструмент на KiCAD , който намира електрически грешки в разработения дизайн на електрическата схема на устройството
- Резултати от ERC (Фигура 3.3.1)

```
ERC report (2/28/2023 11:09:34 PM, Encoding UTF8)

***** Sheet /

** ERC messages: 0 Errors 0 Warnings 0
```

Фигура 3.3.1 - Снимка от ERC проверка за грешки в KiCAD

ЧЕТВЪРТА ГЛАВА

ПРОЕКТИРАНЕ НА ГРАФИЧЕН ОРИГИНАЛ НА ПЕЧАТНА ПЛАТКА

4.1. Информация за печатна платка

4.1.1. Информация за печатната платка

Печатната платка от фигура 4.1.4 е двуслойно опроводена със следните размери:

Широчина: 1.5000 in/3.81 cm

Дължина: 2.4000 in/6.09 cm

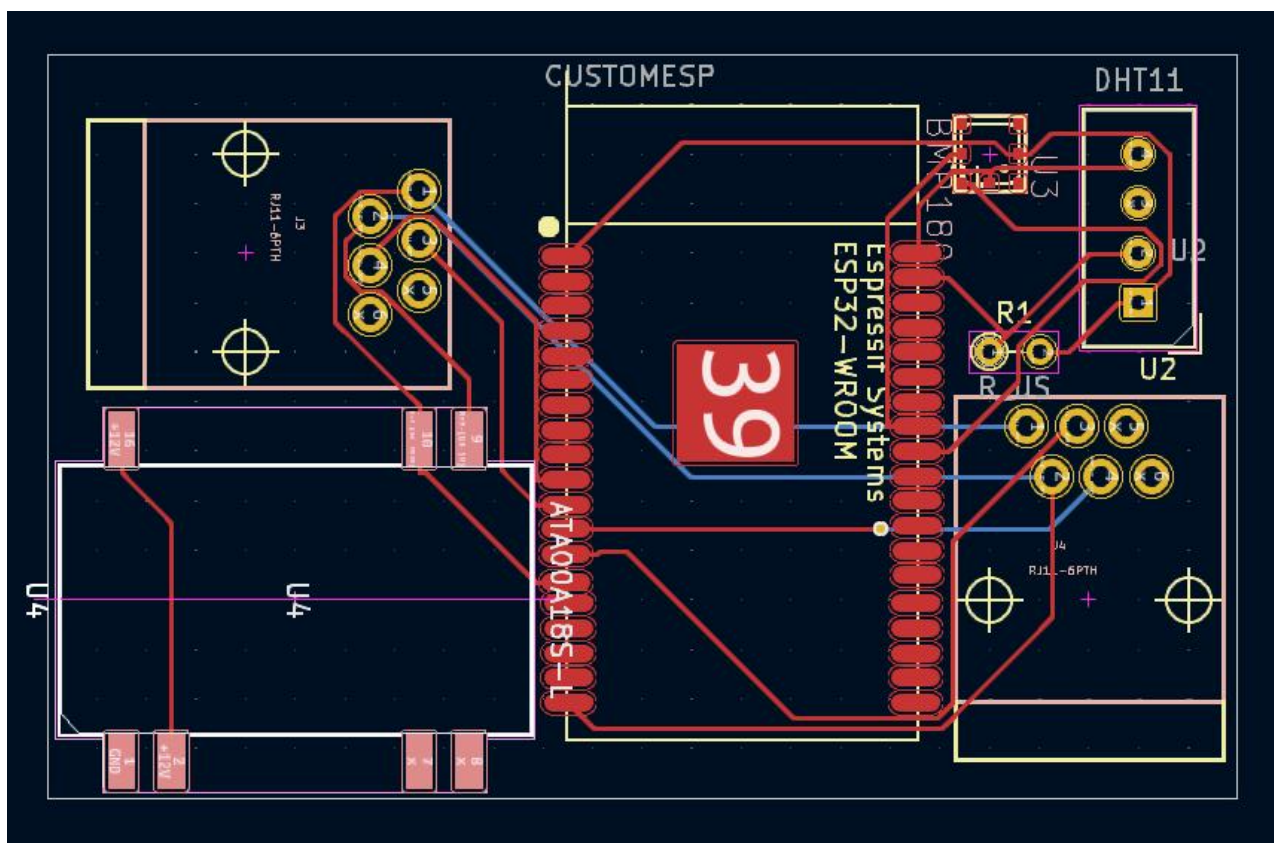
Площ: 3.6000 in²/8.89 cm²

4.1.2. Списък на мрежите в печатната платка

На фигура 4.13 е показана снимка от “Net Inspector” на KiCAD, която показва всички мрежи използвани в печатната платка. Това включва захранванията и информационните мрежи за комуникация между сензорите и микроконтролера.

Net	Name	Pad Count	Via Count	Via Length	Track Length	Die Length	Total Length	F.Cu	B.Cu
001	Net-(U1-3V3)	4	0	0.0000 in	1.9228 in	0.0000 in	1.9228 in	1.9228 in	0.0000 in
003	Net-(U1-IO19)	2	0	0.0000 in	0.6877 in	0.0000 in	0.6877 in	0.6877 in	0.0000 in
004	Net-(U1-IO18)	2	0	0.0000 in	1.1814 in	0.0000 in	1.1814 in	1.1814 in	0.0000 in
005	Net-(U2-GND)	3	0	0.0000 in	0.6081 in	0.0000 in	0.6081 in	0.6081 in	0.0000 in
006	Net-(U1-IO23)	3	0	0.0000 in	0.6101 in	0.0000 in	0.6101 in	0.6101 in	0.0000 in
012	Net-(U1-5V)	4	0	0.0000 in	3.5647 in	0.0000 in	3.5647 in	1.9722 in	1.5925 in
027	Net-(U1-IO12)	2	0	0.0000 in	1.6056 in	0.0000 in	1.6056 in	1.6056 in	0.0000 in
028	Net-(U1-IO14)	2	1	0.0608 in	1.1235 in	0.0000 in	1.1844 in	0.6344 in	0.4891 in
029	Net-(U1-IO27)	2	0	0.0000 in	0.6905 in	0.0000 in	0.6905 in	0.6905 in	0.0000 in
030	Net-(U1-IO26)	2	0	0.0000 in	0.8582 in	0.0000 in	0.8582 in	0.8582 in	0.0000 in
039	Net-(U4--VOUT)	4	0	0.0000 in	2.5883 in	0.0000 in	2.5883 in	1.1666 in	1.4218 in
046	GND	1	0	0.0000 in	0.0000 in	0.0000 in	0.0000 in	0.0000 in	0.0000 in
047	+12V	2	0	0.0000 in	0.6910 in	0.0000 in	0.6910 in	0.6910 in	0.0000 in

Фигура 4.1.3 - Списък на всички мрежи използвани в дизайна на печатната платка



Фигура 4.1.4 - Печатна платка на устройството на дипломния проект.

4.2. Разположение на частите в печатната платка

4.2.1. Разположение на сензорите върху печатната платка

Сензорите BME180 и DHT11 (U3 и U2) са разположени в горния десен ъгъл на платката от фигура 4.1.4, за да може да не се влияят от топлоотдаването на микроконтролера или захранването. Измерващата страна на U2 е насочена навън от платката с цел да може да се събира информация отвън, а не от елементите поставени върху самата платка. Придружаващият резистор на DHT11 е разположен между микроконтролера и самият сензор, за да се улесни опроводяването и свързването.

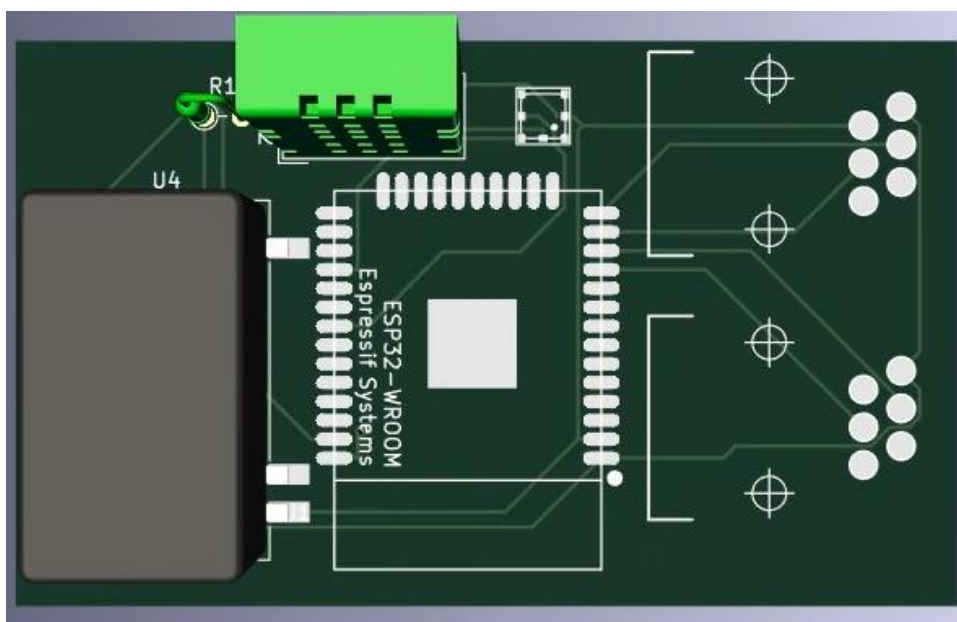
4.2.2. Разположение на конекторите върху печатната платка

Същият принцип на разположение използва и U3. Двата RJ11 конектора (J3 и J4) са разположени в горния ляв и долния десен ъгъл на платката от фигура 4.1.42. Разположени са така, за да са удобни за свързване. Входовете им сочат навън от платката, за да може свързването им да не е възпрепятствано.

4.2.3. Разположение на интегралните схеми върху печатната платка

Захранването (U4) е разположено в долния ляв ъгъл на платката от фигура 4.1.4, за да може да се свързва по лесен и удобен начин. Микроконтролера (U1) е разположен в средата на печатната платка, за да може всички елементи, сензори, конектори и схеми необходими за правилната работа на устройството да са еднакво далече, като по този начин опроводяването на платката да бъде по-лесно.

4.3. 3Д визуализация на печатната платка



Фигура 4.1.5 - 3Д Визуализация на печатната платка

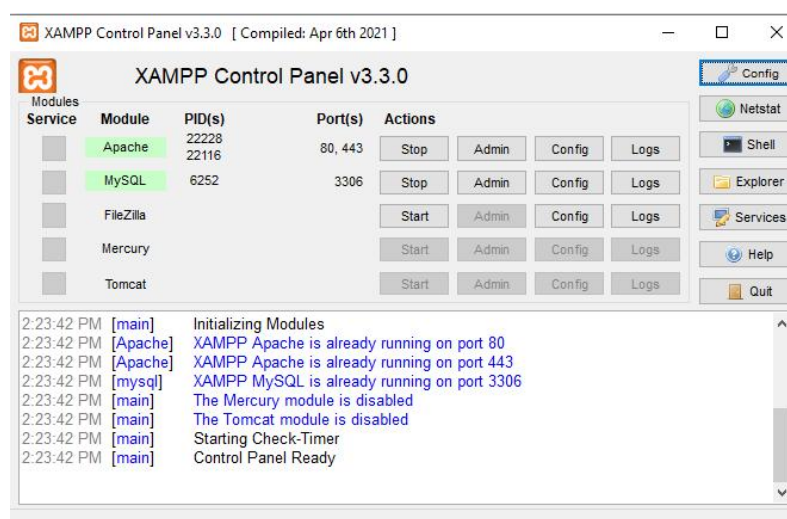
ПЕТА ГЛАВА

ПРОЕКТИРАНЕ НА АЛГОРИТЪМ, УПРАВЛЯВАЩ СОФТУЕР И БАЗА ДАННИ ЗА МИКРОКОНТРОЛЕРНА СИСТЕМА ЗА ИЗМЕРВАНЕ НА ТЕМПЕРАТУРА

5.1. Проектиране и издигане на база данни

- Информацията от сензорните блокове описани във втора глава се запазва в база данни на базата на MySQL сървър. За по-лесен административен достъп на машината със сървъра са инсталирани Apache Web Server както и PhpMyAdmin.

- За експеримента сървърите и приложенията описани в горната подточка са издигнати върху Windows програмата хампр. Тя позволява бързо и лесно на въпросната операционна система да бъдат инсталирани MySQL сървър, Apache Web Server и PhpMyAdmin. (фиг. 4.2) Версията на хампр, която използваме за проекта е 3.3.0. Тя съответно е инсталирала MariaDB сървър с версия 10.4.24, Apache Web Server версия 2.4.52 и PhpMyAdmin версия 5.2.1. Версията на PHP е 8.1.4.



Фигура 4.2 (Контролен Панел на ХАМРР)

Качването на уеб приложението на сървъра става чрез локалната директория на софтуера, а именно /xampp/htdocs/ където се прави отделна папка и вътре се

качват всички необходими файлове от типа на CSS, HTML, PHP както и JavaScript. В случаят на този проект вътре се съдържат следните файлове:

- index.php - Базовият файл с началната страница на приложението и бутон с препратка към главната страница, в която се изобразява информация събирана от сензорите. Стилизирането и дизайна на тази страница се управляват от файла “styles.css”, описан по-долу.

- weather-data.php - Това е главният файл на уеб приложението, в който се изобразяват метеорологичните стойности съхранявани в базата данни. Вътре се съдържат няколко вида код, най-важният от които е php кода, който осъществява връзка с базата данни и прочита информацията и я записва в табличен вид. Дизайна на тази страница се управлява от файла “styles2.css”, описан по-долу.

- main.js - В този файл се съдържа JavaScript код, който отговаря за bootstrap-a, както и за активните бутони в приложението, които позволяват различни видове стойности да бъдат разглеждани и да бъдат наблюдавани техните минимални, средни и пикови стойности.

- styles.css - В този файл се съдържат дизайна и стилизирането на началната страница “index.php”, които са оформени на базата на CSS код.

- styles2.css - В този файл се съдържа дизайна на основната страница и точно дизайна на таблиците, бутоните, фона и цялата изобразена информация на базата на CSS код.

5.2. Проектиране и издигане на уеб приложение за наблюдение на метеорологичните условия

Начална страница на уеб приложението (фигура 4.2.1.)

- Главната идея на първата страница е предимно визия и пригледност. Направена е на HTML и CSS, като е импортнат bootstrap (JavaScript) за границите и шрифтовете на бутоните. Страницата съдържа един бутон, който пренасочва потребителя към следващата страница, която съдържа информацията събирана от максималният брой метеорологични станции към момента (2 броя).

Единственото по специално нещо, което съдържа е css функция, която прави центрираният заглавен текст да има typewriter ефект. (фигура 4.2.2)

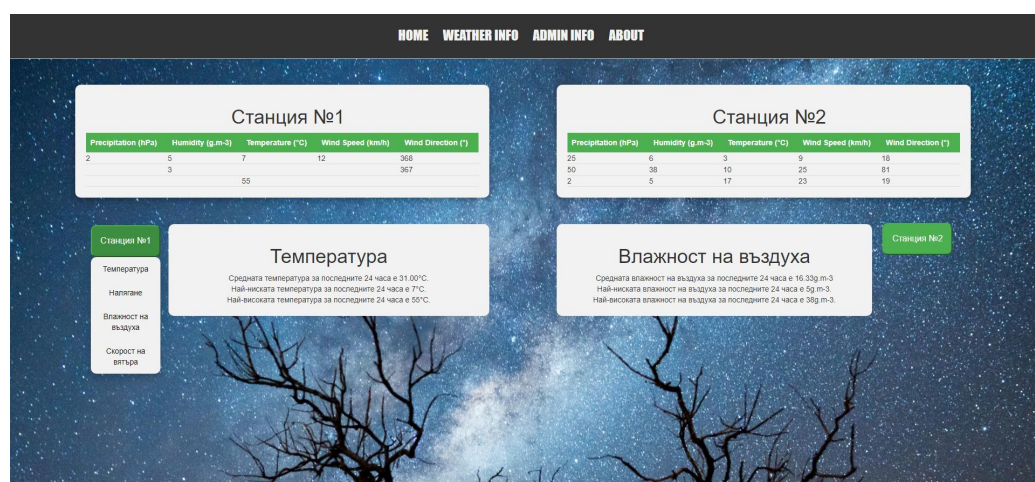
```
.headtext {
  font-family: "Copperplate", fantasy;
  overflow: hidden;
  position: absolute;
  color: white;
  scale: 158%;
  top: 39%;
  left: 44%;
  border-right: .15em solid orange;
  white-space: nowrap;
  margin: 0 auto;
  letter-spacing: .15em;
  animation:
    typing 2.5s steps(30, end),
    blink-caret .75s step-end infinite;
}

@keyframes typing {
  from {width: 0;}
  to {width: 17%}
}

@keyframes blink-caret {
  from, to {border-color: transparent}
  40% {border-color: orange;}
}
```

Фигура 4.2.2. - Снимка на кода за typewriter ефект в уеб приложението

Главна страница на уеб приложението (фигура 4.2.3.)



Фигура 4.2.3. - Главна страница на уеб приложението

Тази страница от уеб приложението съдържа няколко елемента:

- Центрирано меню за навигация (най-отгоре), което съдържа връщане до началната страница, метеорологична информация, пренасочване до уеб панел за достъп до база данни (phpMyAdmin) и меню за информация, което не е разработено все още.

- 2 модула за 2 метеорологични станции, което съдържа по една таблица, която изписва последните стойности записани в базата данни за влажност на въздуха, атмосферно налягане, температура както и скорост и посока на вятъра. Съдържа също така и падащо меню с избор за всяка една от изброените 5 стойности, което при натискане показва най-ниската, най-високата и средната стойност на съответната времева характеристика.

- Падащото меню е направено да работи с цикъл в JavaScript (фигура 4.2.4.), за да може с променлива да се отбелязва коя статистика се използва в момента, за да може менюто да се използва повече от веднъж за всяка една опция.

```
function showData(divId) {  
    var state = {  
        "graph": false,  
        "graphprec": false,  
        "graphhum": false,  
        "graphwinds": false  
    };  
    // hide all divs first  
    Object.keys(state).forEach(function(key) {  
        document.getElementById(key).style.display = "none";  
    });  
    // then show the selected div  
    document.getElementById(divId).style.display = "block";  
    state[divId] = true;  
}
```

Фигура 4.2.4. - Функция за избиране на опции от падащото меню на основната страница

- Информацията се събира от базата данни със “echo” функцията в PHP, която търси определена информация в база данни (фигура 4.2.5.)

```

$sql = "SELECT precipitation, humidity, temperature, wind_speed, wind_direction FROM weather";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    // Output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "<tr>";
        echo "<td>" . $row["precipitation"] . "</td>";
        echo "<td>" . $row["humidity"] . "</td>";
        echo "<td>" . $row["temperature"] . "</td>";
        echo "<td>" . $row["wind_speed"] . "</td>";
        echo "<td>" . $row["wind_direction"] . "</td>";
        echo "</tr>";
    }
}
mysqli_close($conn);

```

Фигура 4.2.5. - Функция за събиране на информация от базата данни

- Полето, в което се показват минималната, средната и най-високата стойност на атмосферното налягане, температурата, влажността на въздуха както и посоката и скоростта на вятъра използва също “echo” функцията на PHP, с предварително зададени настройки да се търсят средни, минимални и максимални стойности (фигура 4.2.6.)

```

$sql = "SELECT MIN(temperature) as min_temp, MAX(temperature) as max_temp, AVG(temperature) as avg_temp FROM weather";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        $avg_temp = number_format((float)$row['avg_temp'], 2, '.', '');
        echo "Средната температура за последните 24 часа е " . $avg_temp . "°C." . "<br>";
        echo "Най-ниската температура за последните 24 часа е " . $row["min_temp"] . "°C." . "<br>";
        echo "Най-високата температура за последните 24 часа е " . $row["max_temp"] . "°C." . "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>

```

Фигура 4.2.6. - Функция за показване на средна, минимална и максимална стойност

- По-долу са описани всички JavaScript библиотеки, използвани в двете PHP/HTML страници:

- bootstrap
- jquery
- npm/mysql

На Фигура 4.2.7. е показана частта от HTML/PHP кода, в която са “повикани” библиотеките/скриптовете

```

</head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/mysql@2.0.0/index.js"></script>
</head>

```

Фигура 4.2.7. - Главна част на HTML файла, в която се поставят хиперлинковете за връзка с въпросните библиотеки/скриптове описани по-горе.

5.3. Проектиране и създаване на кода за устройството

5.3.1. Събиране на информация за температура и атмосферно налягане

- Събирането на информация за температура и атмосферно налягане се извършва от сензор BMP180 свързан към микроконтролера (ESP32). За използване на вградените функции за измерване е необходимо на инсталираме библиотека с име **Adafruit_BMP085**. След това стойността се вкарва в променлива както е показано на фигура 5.1.

```

void loop() {
    float temperature = bmp.readTemperature();
    float pressure = bmp.readPressure() / 100.0F;
}

```

Фигура 5.1 - Променливи за температура и налягане на сензор BMP180

5.3.2. Събиране на информация за влажността на въздуха

- Събирането на информация за влажност на въздуха се изпълнява от сензор DHT11 свързан към микроконтролера. Той поддържа вградени библиотеки, които включват функции за измерване. След като на устройството е инсталирана библиотека **DHT**, може да се използва функцията след променливата на фигура 5.2.

```

float humidity = dht.readHumidity();

```

Фигура 5.2. - Променлива за влажност на въздуха за сензор DHT11

5.3.3. Събиране на информация за посока на вятъра

- Събирането на информация за посока на вятъра се изпълнява от аналогов ветропоказател, който е свързан към микроконтролера на устройството посредством RJ11 конектор. За събирането на информацията от ветропоказателя се използва вградената функция `analogRead`. Прави се `float` функция (фигура 5.3), която оформя стойността на `analogRead` и я записва в променлива, която след това се използва за MySQL Query-то.

```
float Calculate_WindDirection() {  
    int winddirection = analogRead(36); // VP = 36 VN = 39  
    return map(winddirection,0,3095,0,359);  
}
```

Фигура 5.3. - Функция за събиране на информация от ветропоказателя

5.3.4. Събиране на информация за скоростта на вятъра

- Събирането на информация за скорост на вятъра се изпълнява от външно поставен анемометър, който е свързан към RJ11 конектор. За събиране на информацията от анемометъра се използват различни функции. Функцията от фигура 5.4. записва времевия интервал, за следващото измерване.

```
void IRAM_ATTR MeasureWindSpeed_ISR() {  
    portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;  
    portENTER_CRITICAL_ISR(&timerMux);  
    Last_Event_Time = millis();  
    portEXIT_CRITICAL_ISR(&timerMux);  
}
```

Фигура 5.4. Функция за запазване на часа за анемометъра

- Втората функция калкулира скоростта на вятъра, като сравнява предишните стойности и часа им и ги разделя на стойностите за калибрация, като по този начин се получава крайната информация за скоростта на вятъра (фигура 5.5.)

```

float Calculate_WindSpeed() {
    if ((millis() - Last_Event_Time) > 2) {
        WindSpeed = (1.00F / (((millis() - Last_Event_Time) / 1000.00F) * 2)) * WS_Calibration;
    }
    // Calculate average wind speed
    WS_Total = WS_Total - WSpeedReadings[WS_Samples_Index];
    WSpeedReadings[WS_Samples_Index] = WindSpeed;
    WS_Total = WS_Total + WSpeedReadings[WS_Samples_Index];
    WS_Samples_Index = WS_Samples_Index + 1;
    if (WS_Samples_Index >= WS_Samples) {
        WS_Samples_Index = 0;
    }
    WindSpeed = WS_Total / WS_Samples;
    if (Format == "M") WindSpeed = WindSpeed * 1.60934;
    return WindSpeed;
}

```

Фигура 5.5. - Функция за изчисляване на вятъра на базата на предишни измервания

5.3.5. Качване на събраната информация в базата данни на сървъра

- Качването на събраната информация в базата данни се осъществява чрез библиотеките MySQL_Connection и MySQL_Cursor. Събраната информация от всеки сензор се записва в променлива и през определен интервал от време (10 минути) се изпълнява SQL заявка към сървъра, която записва стойностите в една таблица с различни колони за всяка стойност. Колоните приемат стойности с точност до втория знак. Функцията за изпълняването на заявките е показана на фигура 5.6.

```

String query = "INSERT INTO `weather_data` (`temperature`, `pressure`, `humidity`, `wind_direction`, `wind_speed`) VALUES ('";
query += temperature;
query += "', '";
query += pressure;
query += "', '";
query += humidity;
query += "', '";
query += wind_direction;
query += "', '";
query += wind_speed;
query += "')";

MySQL_Cursor* cursor = new MySQL_Cursor(&conn);
cursor->execute(query);
delete cursor;

Serial.println("Data uploaded to MySQL database!");

delay(600000);
}

```

Фигура 5.6. - Функция за изпълняване на SQL заявки за качване на стойности на сървъра.

БЪДЕЩО РАЗВИТИЕ

6.1. Бъдещо развитие на устройството

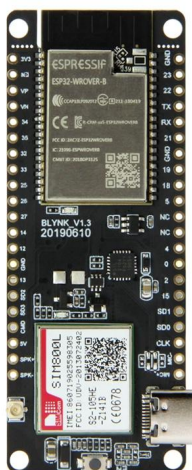
- За бъдещото развитие на самото устройство могат да бъдат променени следните компоненти:

- За измерване на атмосферно налягане и температура - от BMP180 (фигура) на BMP280 (фигура 1.12), поради по-големите си диапазони на измерване и по-нисък толеранс и за двете стойности.

- За измерване на влажността на въздуха - от DHT11 (фигура) на DHT22 (фигура 1.9), заради по-големия диапазон на измерване и по-нисък толеранс на въпросната стойност.

- За бъдещото развитие на устройството и улеснена поддръжка и комуникация с него могат да бъдат добавени следните неща:

- За успешен отдалечен монтаж на устройството, микроконтролера е различен от този на експеримента (нормален ESP32), а именно бива заменен със ESP32-SIM800L (фигура 5.1.1), заради поддръжката си за мобилни мрежи чрез SIM свързване. Това прави устройството достъпно навсякъде, където има мобилен обхват.



Фигура 5.1.1 - ESP32-SIM800L

- За улеснена поддръжка при липса на електрическа връзка към устройството могат да се добавят слънчеви панели, които при

благоприятни метеорологични условия да зареждат батерията на устройството.

6.2. Бъдещо развитие на приложението

- За бъдещото развитие на уеб приложението могат да се направят следните промени:

- За по-голяма сигурност на достъпа до данните да се добави автентикация с потребителско име и парола, която според зависи данните дава достъп до определена метеорологична станция и нейната таблица - както и данните, които визуализира в приложението.

- Освен сегашните таблици и други инструменти използвани за визуализиране на данните да се добавят графики, които да улесняват визуализацията на статистиките за определен период от време (например 1 час, 1 ден, 1 седмица).

- Да се добави поддръжка за повече от две метеорологични станции, която да е контролирана от достъпа по потребителско име и парола описан по-горе.

ЗАКЛЮЧЕНИЕ

При разработката на дипломния проект са разгледани съществуващи видове метеорологични станции - всеки вид, от които със своите употреби, ползи и среди, в които са създадени да се използват.

Този дипломен проект спада към Smart/IoT метеорологичните станции. Целта на проекта е да се създаде метеорологична станция, която да бъде поставена на отдалечено място и да комуникира с уеб сървър чрез мобилна мрежа. Устройството измерва следните метеорологични стойности:

- Влажност на въздуха
- Температура
- Атмосферно налягане
- Скорост на вятъра
- Посока на вятъра

Тези стойности след това се визуализират в Уеб Приложение, което поддържа до 2 метеорологични станции едновременно. Приложението може да показва минималните, средните и максималните стойности, измерени от устройството. Стойностите, които се визуализират в уеб приложението се съхраняват в база данни, надигната на MariaDB, която запазва стойностите от всеки час на всяко денонощие за период от 1 седмица (7 дни).

Предназначената среда на употреба е любителска/домашна. Устройството може да се използва например от хора, които имат отдалечени имоти. Така може да се монтира устройство на едното място и от всяко място с интернет достъп да се наблюдават въпросните стойности и да може да се реагира при определена ситуация.

ИСПОЛЗВАНА ЛИТЕРАТУРА

1. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
2. <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>
3. <https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>
4. <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
5. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
6. https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
7. <https://www.milesight-iot.com/lorawan/wts305-wts50x/>
8. <https://httpd.apache.org/docs/>
9. <https://www.php.net/docs.php>
10. https://www.w3schools.com/whatis/whatis_css.asp
11. <https://developer.mozilla.org/en-US/docs/Web/HTML>
12. <https://mariadb.org/documentation/>
13. https://eu.mouser.com/datasheet/2/633/ata_series_ds_01apr2015_79c25814fd-1308284.pdf
14. <https://docs.kicad.org/>

СЪДЪРЖАНИЕ

ДИПЛОМНА РАБОТА	1
УВОД	4
1.0. Въведение в областта на IoT метеорологичните станции	4
1.1. Цели и задачи на дипломния проект	4
ПЪРВА ГЛАВА	6
1.2. Преглед на подобни системи и продукти	6
- Домашни метеорологични станции (фиг.1.1)	6
- Smart метеорологични станции	7
- Професионални метеорологични станции (фиг.1.2)	8
- Отдалечени метеорологични станции (фиг.1.3)	9
1.3. Преглед на технологиите, които се използват в метеорологичните станции	10
1.4. Преглед на устройствата, използвани в дипломния проект	11
1.4.1. Избор на микроконтролер за изпълнение на дипломния проект	11
1.4.2. Избор на сензор за температура и влажност на въздуха	14
1.4.3. Избор на сензор за атмосферно налягане и температура	16
1.4.4. Избор на сензор за скорост и посока на вятъра	19
1.5. Преглед на технологиите, използвани в дипломния проект	20
1.6. Преглед на програмните езици, използвани в проекта	21
ВТОРА ГЛАВА	26
2.1. Функционални изисквания към дипломния проект	26
2.2. Проектиране на блоковата схема на устройството за метеорологични измервания	26
2.3. Разглеждане на отделните модули от блоковата схема (фиг 2.1)	27
ТРЕТА ГЛАВА	30
3.1. Принципно електрическа схема на устройството за измерване на метеорологични условия (фигура 3.1)	30
3.2. Разглеждане на елементите от електрическата схема	30
3.2.1. Конектори използвани в проекта	30
3.2.2. Интегрални схеми използвани в проекта	31
3.2.3. Градивни елементи използвани в проекта	33
3.3. Проверка на принципната електрическа схема на проекта за грешки	33
ЧЕТВЪРТА ГЛАВА	35
4.1. Информация за печатна платка	35
4.1.1. Информация за печатната платка	35
4.1.2. Списък на мрежите в печатната платка	35
4.2. Разположение на частите в печатната платка	36
4.2.1. Разположение на сензорите върху печатната платка	36
4.2.2. Разположение на конекторите върху печатната платка	36
4.2.3. Разположение на интегралните схеми върху печатната платка	37
4.3. 3D визуализация на печатната платка	37
ПЕТА ГЛАВА	38
5.1. Проектиране и издигане на база данни	38
5.2. Проектиране и издигане на уеб приложение за наблюдение на метеорологичните условия	39
5.3. Проектиране и създаване на кода за устройството	43
БЪДЕЩО РАЗВИТИЕ	46
6.1. Бъдещо развитие на устройството	46
6.2. Бъдещо развитие на приложението	47
ЗАКЛЮЧЕНИЕ	48
ИЗПОЛЗВАНА ЛИТЕРАТУРА	49
СЪДЪРЖАНИЕ	50