## Документација за SmartFlora

# Планирање и анализа

Идејата при дизајнирање на SmartFlora системот е инспирирана од потребата на луѓето од помошник кој би ги „советувал“ како да ги одгледуваат своите растенија во домот. Многу често растенијата во домот се сушат или умираат од несоодветните услови во кои се одгледувани, затоа уредот на кој ќе работиме се грижи корисникот да ги одгледува растенијата на правилен начин, во правилни услови. Уредот им дава информации на корисниците, кои не знаат многу за растенијата, но сепак сакаат да ги одгледуваат во својот дом. Доколку условите во кои е растенијето се несоодветни за тоа растение корисникот ќе биде информиран како да ги подобри.

Бидејќи често оние кои имаат растенија во домот не ги знаат условите во кои тоа се одгледува, овој хардвер би го решил тој проблем на едноставен начин. Токму тие се главната целна група која би го користела овој продукт, но и оние кои имаат искуство со одгледување на растенија, но сепак не се сигурни за некои од условите во кои ги чуваат во домот или сакаат да бидат информирани за моменталните услови каде растението се одгледува. Соодветен е и за луѓето кои имаат многу растенија во домот, па немаат време да ги подредуваат сите нив. Најчесто крајните корисници би биле обожаватели на растенија, независно од возраста и родот, кои сакаат зеленило во својот дом, но немаат време да ги учат условите под кои тоа се одгледува. Со помош на овој уред може да се учат и децата на обврски кои треба да ги извршуваат во домот.

Уредот би користел микроконтролер и со помош на сензори ќе ги чита информациите од околината на растението, со тоа што уредот треба да се постави во самата саксија од растението, притоа би ги отчитувал најсличните дразби за вода, светлина и топлина, како и влажност на воздух кои ги добива растението. При избирање на микроконтролер врз кој ќе го базираме SmartFlora уредот, решивме дека најдобар избор во овој случај би бил ESP32 микроконтролерот, кој поддржува и конекции преку WiFi и Bluetooth. Уредот би бил сместен во дрвено куќиште со тоа нема да се промени природниот изглед кој го носат растенијата во нашите домови.

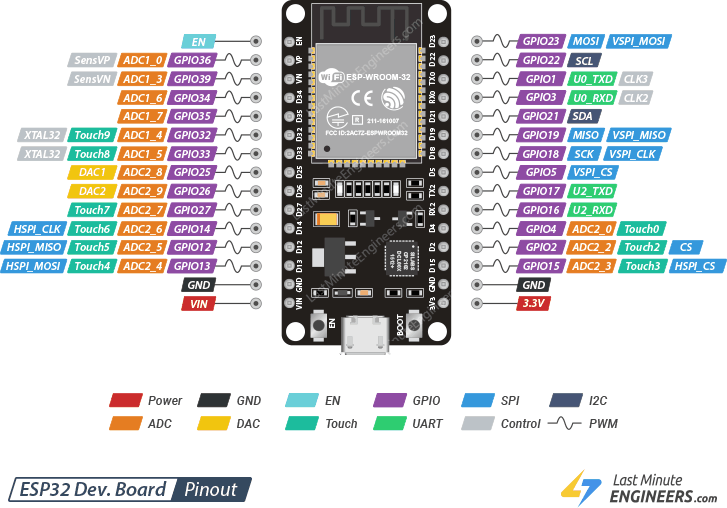
Овој уред претставува единствен уред, иако постојат слични уреди, сепак предноста која ја имаме при користење на овој уред е тоа што сите податоци за растението може да се пристапат онлајн преку исцртани графици, а доколку некој од условите не одговара за конкретното растение добиваме WhatsApp порака со содржина што конкретно треба да се промени. Овој уред го сместуваме во паметни уреди за одгледување на растенија. Ќе ве извести кои услови за растението треба да се подобрат, но сепак контактот помеѓу корисникот и растението останува.

# Дизајнирање

По планирањето и анализата на потребата од продуктот, односно SmartFlora помошникот за одгледување растенија, следи фазата на дизајн на продуктот. Во оваа фаза ќе ги дефинираме архитектурата на системот и техничките ресурси кои се користат.

**Архитектура**

Микроконтролер



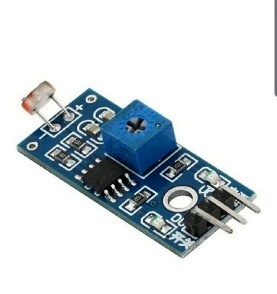
Слика 2.1 – ESP32 микроконтролер

За овој проект искористивме ESP32 модул (слика 2.1), односно готов сет за развој со кој полесно ќе можеме да го изградиме уредот. ESP32 е всушност кит базиран на микроконтролер кој може да се користи или директно со купување или може да се направи дома користејќи ги компонентите, поради неговата хардверска карактеристика со отворен код. Во основа се користи во комуникациите и при контролирање или управување со многу уреди.

Процесорот на ESP32 во основа ја користи архитектурата на Харвард каде што програмскиот код и програмските податоци имаат посебна меморија. Покрај главниот процесор со две јадра: Tensilica Xtensa LX6, на плочката има и ко-процесирачка единица која служи за ултра заштеда на енергија. Се состои од две мемории - програмска меморија и податочна меморија. Има 320 KB SRAM меморија, како и 4MB флеш меморија и специфично е тоа што поддржува Bluetooth и Wi-Fi конекција.

Покрај микроконтролерот, ќе користиме и сензори кои ќе служат за примање на дразби од околината. Бидејќи уредот работи слично како што растенијата примаат дразби од околината, мора да ги има сензорите кои ќе му овозможат да прима светлосни, температурни и сигнали за влажност на почвата и воздухот.

Сензор за светлина (Фотоотпорник)



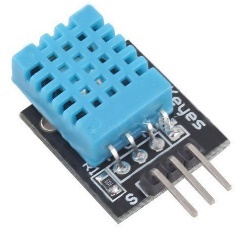
Слика 2.2 – LDR модул

Фотоотпорниците, познати и како отпорници зависни од светлина (LDR), се уреди чувствителни на светлина кои најчесто се користат за означување на присуството или отсуството на светлина или за мерење на интензитетот на светлината. Во темнина, нивниот отпор е многу висок, понекогаш и до 1 MΩ, но кога LDR сензорот е изложен на светлина, отпорот драстично опаѓа, дури и до неколку оми, во зависност од интензитетот на светлината. LDR отпорниците имаат чувствителност која варира во зависност од брановата должина на применетата светлина и се нелинеарни уреди.

Фотоотпорникот ќе биде прикачен на GPIO пин на ЕЅР32, но преку модул (слика 2.2). Исто така, овој модул помага во решавање на проблемот при поставување на threshold доколку се користи дигиталниот пин, како и можност да се постави колку ќе биде фотоотпорникот сензитивен на светлина со помош на потенциометарот на самиот модул. За SmartFlora уредот ќе се користи аналогниот излез од модулот, кој ни дава вредности врз база на јачината на светлината која паѓа на фотоотпорникот.

Со помош на софтвер прилагоден е сензорот така што јачината на светлина ја мери само кога е ден, според часот кога изгрева и заоѓа сонцето, бидејќи светлината од сијалиците не е доволна за растенијата. Ова е направено со помош на библиотеката „time.h“ која ни го дава точното време според временската зона.

DHT11 сензор за температура



Слика 2.3 – DHT11 сензор

DHT11 (слика 2.3) е најчесто користениот температурен сензор, кој оперира од 3,3V до 5V и користи т.н. интерфејс на една жица. Иако сензорот има 4 пина, уникатниот интерфејс 1-Wire бара само еден дигитален пин за двонасочна комуникација со микроконтролер. За полесно користење за овој сензор се користи библиотека која има улога и на pull-up отпорник. Покрај температура овој сензор дава и мерења за влажноста на воздухот кој е исто така важен фактор за одгледување на растенија во домот.

Капацитивен сензор за влага на почва (v1.2)



Слика 2.4 – Сензор за влага на почва

Сензорот за влага во почвата (слика 2.4) дава мерења врз основа на капацитативноста помеѓу сензорот и почвата, која се менува според тоа колку е влажна почвата. При избор на сензори кои ќе се користат за проектот го избравме капацитивниот сензор, наспроти сензорот кој ја мери влажноста на почвата со помош на електрична енергија, односно отпор. Проблемот кај ваквите сензори кои спроведуваат струја низ почвата е тоа што имаат отворени контакти кои брзо кородираат поради изложеноста на влага, од друга страна капацитивниот сензор е многу подолготраен.

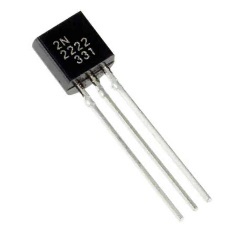
Дополнително, бидејќи се работи за паметен уред кој според внесените параметри од корисникот ќе помага во одгледувањето на растението за кое е наменет, покрај сензорите во системот е вклучен и актуатор, односно електрична пумпа за наводнување.

Пумпа за наводнување

Пумпата за наводнување (слика 2.5) го прави токму она кое најчесто го грешат одгледувачите на растенија, ги наводнуваат растенијата повеќе од потребно или напротив забораваат. Постојат 2 вида на пумпи кои може да се искористат во овој проект, а тоа се потопни пумпи или пумпи кои влечат вода. За поврзување на пумпата се користи и NPN (2N222) (слика 2.6) транзистор во улога на прекинувач. Сигналот за контрола од микроконтролерот се носи до базата на транзисторот. Емитерот на транзисторот е директно поврзан на заземјување. 5V од микроконтролерот се водат директно на влезот на пумпата, а негативната страна на пумпата се води на колекторот на транзисторот. Дополнително, тука е поставена и обратно поларизирана диода, која служи за заштита.



Слика 2.5 – пумпа за наводнување

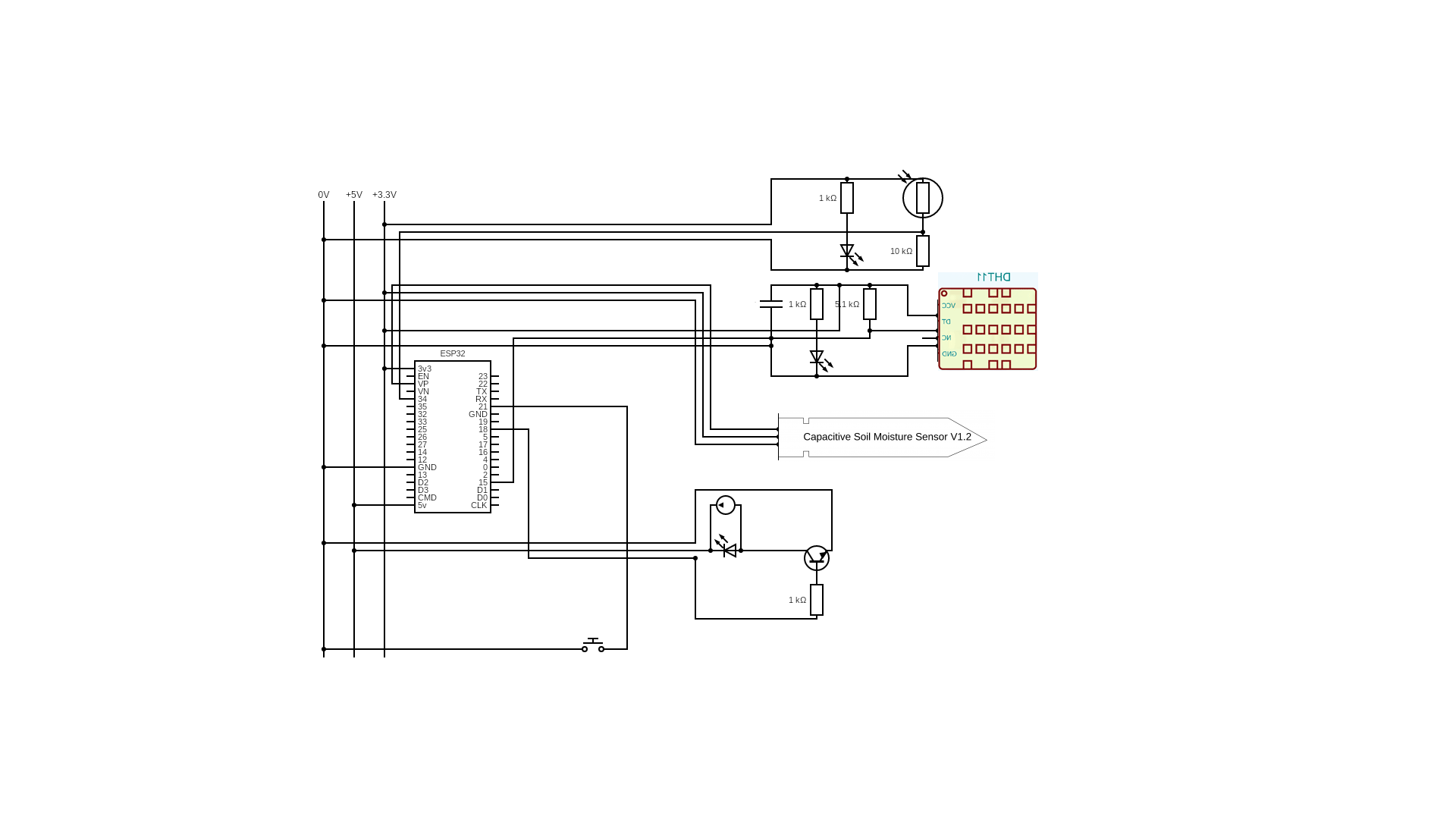


Слика 2.6 – 2N222 транзистор

**Поврзување на компонентите**

Шематски приказ на поврзувањето на компонентите може да се види на слика 2.7:

Слика 2.7



**Софтвер**

Arduino IDE

Софтверот Arduino (IDE) со отворен код го олеснува пишувањето код и поставувањето на плочката. Иако во овој проект се работи со ESP32, Arduino IDE дава поддршка и за овој тип на микроконтролери со инсталирање на библиотеката „ESP32 Dev Module“.

За програмирање на микроконтролерот, го користев стандардното Ардуино IDE. Кодот, што работи на компјутерот кој ги прима податоците од сериската порта и ги декодира за извршување на различни функции, е напишан во C.

При креирање на проект кој содржи повеќе компоненти, како овој помошник за одгледување растенија, обично се користат готови библиотеки кои ни овозможуваат лесно конектирање на компонентите и користење преку готови методи кои се веќе тестирани, со што грешката на програмерот се намалува.

За креирање на овој уред беа искористени повеќе библиотеки кои може да се додадат на Ардуино околината преку package manager.

**DHTesp.h** ESP библиотека за DHT11 сензорски модул (со 3 пина) за мерење на температура и влажност на воздух. Оваа библиотека е многу едноставна и интуитивна за користење и поддржува сензори преку автоматско откривање и зафаќа многу мал мемориски простор, бидејќи има краток код.

**WiFiManager.h** библиотека за поврзување на WiFi мрежа. Со цел корисникот да може да го поврзи уредот, односно ESP WiFi модулот на одредена мрежа, се користи оваа библиотека која нуди пронаоѓање на сите околни мрежи и лесно поврзување на нив. Исто така преку оваа библиотека може да се сетира IP адресата на која ќе биде подигнат серверот.

**WiFi.h** oвозможува мрежна врска со помош на shield за WiFi на Arduino или ESP. Со оваа библиотека можете да инстанцирате сервери, клиенти и да испраќате или примате UDP пакети преку WiFi.

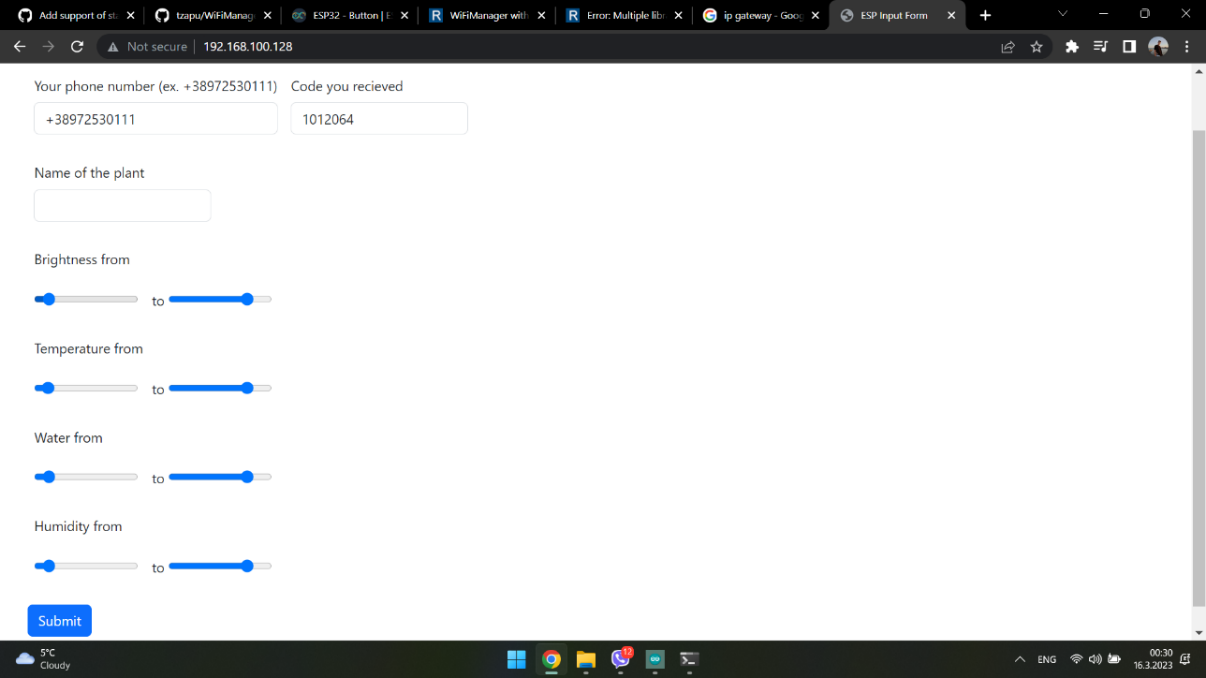
**AsyncTCP.h** е библиотека наменета за ТСР асинхрони повици за ESP32, како додаток на оваа библиотека се користи и **ESPAsyncWebSrv.h** која нуди асинхрон HTTP и WebSocket сервер.

**Blynk.h** е библиотека која не поврзува со онлајн платформа изградена за целосна конекција и мониторирање на хардвер. Преку Blynk може да исцртуваме графици или да гледаме моменталната состојба на сите сензори. Предноста е во тоа што не треба да го оптоваруваме нашиот хардвер да ги прави сите овие сложени операции и чување на податоци.

**Кориснички интерфејс**



Слика 2.8

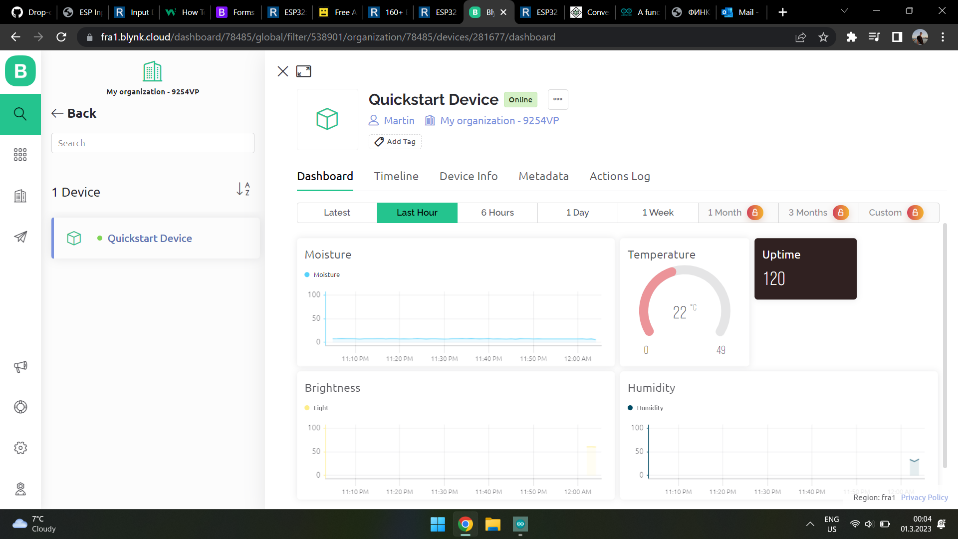


Слика 2.9

Од страна на услужување на корисниците, односно кориснички интерфејси, SmartFlora помошникот за одгледување тоа го прави на неколку начини. На самиот почеток за да се подеси уредот треба да се отвори корисничкиот интерфејс кој е на IP адреса 192.168.4.1, каде добиваме преглед на сите достапни WiFi мрежи во околината на уредот и добиваме опција да го вклучиме уредот на некоја мрежа и да ги подесиме опциите за IP адреса на серверот (слика 2.8). Тука се сетира статичка IP адреса на која подоцна би работел серверот, во идни верзии на уредот овој дел може да се промени во динамичка IP адреса, која потоа корисникот ќе треба рачно да ја внесе во својот прелистувач.

Секако корисниците имаат можност да го променат подесувањето односно уредот да ја заборави мрежата на која сме се приклучиле. Уредот треба да се извади од напојување, да се притисне копчето и држејќи го копчето уредот да се врати на напојување. По 5 секунди може да се отпушти копчето и уредот ќе треба повторно да го поврземе на некоја од достапните мрежи.

Следниот дел каде корисникот може да пристапи е серверот кој се подига на ESP32 микроконтролерот. Тука корисникот може да ги постави сите променливи кои се однесуваат на растението, како што се температура, влажност, име и слично, како и променливите кои се однесуваат на примањето на пораки (слика 2.9). Оттука корисникот е пренасочен на екран кој му ги покажува моменталните сетирани опции.



Слика 2.10

Третата можност која ја нуди SmartFlora помошникот за одгледување на растенија е мониторирање на моменталните услови во кои се наоѓа растението, односно просек земен од претходни неколку мерења. За појасна репрезентација овој просек може да се види на график кој покажува како се движела температурата, влажноста и светлината во последните неколку часа. Сето ова се одвива преку Blynk сервисот, кој може да се пристапи преку прелистувач или мобилна апликација (слика 2.10).

Бројот на мерења во единица време се подесува во кодот и во моментов е сетиран да мери на секои 10 секунди и заокружувањето да го прави на секои 10 мерења, со цел да биде полесно тестирањето, во продукција би требало растението да мери нови вредности од сензорите на секој половина час, а на секое дванаесетто мерење да прави просек кој ќе го споредува со границите за конкретното растение.

const int avgMesNum = 10; // na kolku merenja da se napravi prosek

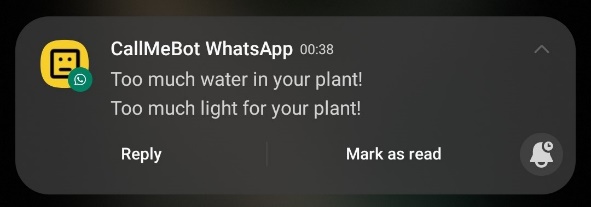
const int timeForMes = 10000L; // na kolku milisekundi da se odviva sekoe merenje

if(mesNum >= avgMesNum) {

waterAvg = waterAvg / mesNum;

if(waterAvg > waterMax.toInt()){

sendMessage("Too much water in your plant!"); }}



Слика 2.11

Овој дел исто така се користи и за праќање на WhatsApp пораки до корисникот доколку нешто не е во ред со условите во кои се наоѓа растението, а со цел корисникот да не добива спам пораки.

**Комуникација**

Ја користевме сериската комуникација (позната и како USB) за испраќање податоци од Arduino IDE на компјутерот преку USB сериски COM порти. Со помош на ваква жичана комуникација е ставен целиот код на ЕЅР32 микроконтролерот.

Universal Serial Bus, скратено USB, е индустриски стандард кој дефинира спецификации за кабли и протоколи за комуникација и напојување. USB вообичаено се користи заедно со комуникациските стандарди како што се EIA, RS-232, RS-422 или RS-485. Постојат повеќе хардверски USB порти, на ESP32 микроконтролерот се користи Micro-B портата која го користи стандардот донесен од OMTP. Го користевме USB за префрлање на податоци од компјутер на ESP преку сериски COM порти.

Откако, уредот ќе почне да се користи од страна на корисниците, micro USB портата може да биде користена само за напојување, иако тоа може да биде и преку специјалните пинови за напојување на микроконтролерот.

Уредот во главно се потпира на WiFi комуникација. Кога вашиот ESP ќе се вклучи, се поставува во Station mode и се обидува да се поврзе со претходно зачуван Access point. Ако ова е неуспешно (или нема зачувана претходна мрежа), ESP се преместува во режим на Access point и се подига веб-сервер (на стандардна IP 192.168.4.1). Користејќи кој било уред со вклучен WiFi со прелистувач (компјутер, телефон, таблет) поврзете се со новосоздадената пристапна точка “PlantAssistant” и внесете ја лозинката “password”. Потоа ќе добиете pop-up прозорец од типот „Join to network“ или ќе бидете пренасочени до конфигурацискиот портал (192.168.4.1). Доколку не бидете пренасочени секогаш може да ја впишете IP адресата каде може да изберете една од скенираните Access point-и (SSIDs), внесете лозинка, кликнете зачувај. ESP ќе се обиде да се поврзе. Ако е успешнo, ја препушта контролата назад на вашата апликација. Ако не, повторно поврзете се на PlantAssistant access point-от и повторно конфигурирајте.

Дополнително, на страната може да се промени IP адресата на која ќе работи микроконтролерот, односно адресата на која ќе биде подигнат серверот, како и маската и gateway-от. Доколку, ништо од тоа не се промени автоматски серверот ќе биде подигнат на 192.168.100.128/24, со gateway 192.168.100.1.

Доколку корисникот сака да ги промени поставувањата за WiFi треба да го исклучи напојувањето на уредот, да го притисне копчето, да го вклучи уредот на напојување и по 5 секунди сите WiFi поставувања ќе бидат избришани при што корисникот ќе треба да се закачи на PlantAssistant мрежата и да го сетира уредот.

Откако SmartFlora уредот ќе биде успешно поврзан на WiFi мрежата на корисникот, тој го стартува веб серверот на внесената IP адреса. На таа адреса добиваме веб страна каде може да се нагодат сите барања за растението кое во моментов го мониторираме. Дополнително, на истата веб страна може да се сетира на кој телефонски број сака корисникот да добива известувања за состојбата на растението (каде и се дадени чекори за сетирање). Целиот систем работи со НТТР пакети кои се пренесуваат до микроконтролерот кој има улога на НТТР сервер. Потоа информациите добиени со тие пакети се користат за подесување на границите на сензорите и слично.

За испраќање на порака до корисникот се користи НТТР post барање до сервер наречен callmebot.com, кој потоа испраќа WhatsApp порака до корисникот доколку тој дел е успешно сетиран. Се одлучивме да го користиме овој сервис за испраќање на пораки бидејќи е мултиплатформски и се што е потребно е корисникот да има WhatsApp профил. За Blynk апликацијата исто така се користат НТТР post барања. Тие барања се процесираат од т.н. Blynk Cloud, на кој потоа може да се приклучиме со телефон или компјутер и да ги гледаме излезите од сензорите. На оваа апликација се исцртуваат и графици кои ни овозможуваат да видиме кога во текот на денот растението немало вода, температурата била зголемена и слично.

## Имплементација

Целта на оваа фаза од животниот циклус е да се создаде прототип кој ќе може да биде тестиран од страна на крајните корисници, кои даваат повратна информација за квалитетот на системот. По оваа фаза следи последната фаза на праќање на продуктот во производство, за да стигнеме до таа фаза мора да согледаме дали продуктот ги исполнува стандардите за безбедност, пред се, но и дали продуктот е скалабилен, одржлив и надградлив.

Прво треба да се направи „груб“ прототип преку кој ќе може да ги согледаме сите функционалности на продуктот, но не во крајна форма, односно без крајниот „сјај“. Овој тип на прототип се нарекува алфа прототип и кај ваквиот прототип има место за подобрувања на функционалностите. Тука треба да се согледаат проблемите при функционирањето, за во следната фаза на бета прототип да бидат поправени. Бета прототипот е готов за продукција, него го тестираат крајни корисници кои даваат забелешки за функционалностите и дизајнот.

Имплементација (дел од алфа прототип): Сите компоненти му даваат податоци од читањата на ЕЅР32, каде што ги претвораме овие читања во код за ефикасно и брзо испраќање на бајтите до компјутерот. Внатре во јамка која никогаш не завршува, секое читање одговара на еден карактер од кодот. Така се креира пакет секој пат кога јамката работи, и тој се испраќа до компјутерот. Овој пакет ги содржи податоците што доаѓаат од секој сензор, и конечно ова треба да се декодира на компјутер за да се добијат отчитувањата на сериски монитор.

(Кај бета прототипот): Во крајниот продукт овие отчитувања ќе се извршуваат на самиот микроконтролер и податоците пресметани ќе се прикажуваат Blynk апликацијата или за нив ќе му пристигне порака на корисникот, со цел лесно да може да заклучи што му е потребно на растението.

Кодот во кој се напишани инструкциите на микроконтролерот е во програмскиот јазик С. За полесно да ги користиме компонентите како сензори и Wifi модул во овој проект користевме библиотеки кои содржат готов, тестиран код и методи за лесен пристап.

За едноставно да може да се прават промени за креирање на првичниот прототип користевме Breadboard преку кој ги поврзувавме компонентите. Откако го завршивме хардверскиот дел, односно поврзувањето следуваше делот на пишување код и тестирање на функционалностите. При овој процес, проучувајќи ја библиотеката за поврзување на Wifi мрежа (WifiManager), заклучивме дека уредот ја чува внесената мрежа, но бидејќи корисникот би имал потреба да ја внеси мрежата повторно, во хардверот додадовме копче со кое може да се ресетира зачуваните SSID и лозинка.

Овој прототип кој го имаме составено во моментов, ги има сите функционалности на планираниот продукт. При првично стартување уредот се поставува во фаза на чекање се додека не се внесат податоци за мрежа, откако тоа ќе биде успешно направено, корисникот треба да ги внесе податоците за растението и потоа уредот почнува да ги прави мерењата и да работи со податоците.

Во понатамошниот процес на креирање на уредот, ќе биде направено и куќиште од дрво со што ќе биде лесно вклоплив како додаток на секое растение.

# Масовно производство

Оваа е последната фаза од животниот циклус за развој на вградливи системи (EDLC) и во неа треба продуктот да биде подготвен за масовно производство и комерцијална продажба на крајни клиенти. Најзначајниот дел од оваа фаза е да се направи добра пазарна стратегија со цел продуктот да биде прифатен од потрошувачите на пазарот. Маркетингот на производи е многу различен за секој производ и исто така многу важен за успешно позиционирање на производот на пазарот.

**Стратегија при лансирање на продуктот**

Оваа стратегија ќе ги вклучи сите аспекти од лансирањето на нашиот производ, од маркетинг планот и кампањите пред лансирањето до финансиските и идните проекции. Општо земено, стратешкото размислување може да игра клучна улога во успешно лансирање на нов производ.

Треба да ги земеме предвид работите како што ќе се позиционирањето на пазарот, како нашите конкуренти ги лансираат своите производи, каде се движи пазарот и слично.

**Процес на набавка**

Овој дел дава преглед на барањата за процесот неопходни за управување со набавките на потребните електронски компоненти. Овој процес треба да вклучува:

* Иницирање барање
* Барање одобрување од економски сектор
* Преглед на понуда/предлог
* Управување со договор
* Барања за затворање на договорот

**Производство**

Знаењето како да започнеме производствен бизнис е само да знаеме што му треба на пазарот и како можеме да го испорачаме тоа. И секоја успешна производствена компанија мора да започне со истражување на пазарот. Ова е моментот кога почнуваме да учиме за производствениот сектор и индустријата како целина.

Производството се однесува на процесите и техниките што се користат за претворање на суровини или полупроизводи во готови производи или услуги со или без употреба на машини. Во овој случај бидејќи се користат готови делови, се работи за склопување на полупрозиводи.

Сметаме дека најуспешна би била токму традиционалната производна стратегија која се заснова на прогноза за побарувачка. Најдобро се користи кога има предвидлива побарувачка за производ, како и во овој случај. Меѓутоа, овој начин на производство може да биде проблематичен кога побарувачката е потешка да се предвиди. Кога се користи со бизнис или производ што има непредвидлив деловен циклус, може да доведе до премногу залихи и намалување на профитот, или премала залиха и пропуштена шанса.

**Тестирање на произведените продукти**

Постпродукциското тестирање е клучна услуга која додава значителна вредност на производството на склопени кола. Тестирањето овозможува подобрен квалитет и намалување на времето до пазарот. Во случајот со SmartFlora ваквото тестирање е неопходно бидејќи самиот уред е составен од многу полупроизводи, кои може да бидат во лоша состојба пред производството.

**Маркетинг**

Планирањето на лансирање на производ може да биде комплицирана процедура. Има многу работи што треба да се земат предвид, па затоа е неопходно да се подготви маркетинг план (табела 4.1). Ова ќе ни помогне да ги организираме активностите потребни за да стекнеме што е можно поголема почетна привлечност и пазарот да дознае за нашиот нов производ.

|  |  |  |
| --- | --- | --- |
| Пред лансирање на пазар | При лансирање на пазар | Кога сме веќе на пазар |
| Социјални медиуми | Социјални медиуми | Кориснички сведоштва |
| Платени реклами | Платени реклами | Студии на корисници |
| Настани (Прес) | Настани (Прес) | Платен маркетинг |
| Печатени реклами | Блог | Социјални медиуми |
|  | Маркетинг преку е-маил | Блог |
|  | Webinars | Форум на корисници |
|  | Дигитален ПР |  |

Табела 4.1

**Поддршка на корисници**

План за обезбедување континуирана економска логистичка поддршка и управување со системи по престанокот на производството на продуктот. Во овој дел треба да се напоменат и резервните делови кои би биле потребни да се променат на самиот SmartFlora уред. Како најчесто расиплив дел од системот треба да се наведе пумпата за наводнување, која заради изложеноста на различни минерали од водата треба односно препорачано е да се променува еднаш годишно. Бидејќи пумпата се наоѓа надвор од куќиштето на помошникот, секој корисник може сам да ја постави.

# Користени референци:

1. Natallia Sakovich, Embedded Product Development Life Cycle: Four Main Steps, <https://www.sam-solutions.com/blog/embedded-product-development-life-cycle/>, (пристапено 22.3.2023)
2. USB hardware, <https://en.wikipedia.org/wiki/USB_hardware>, (пристапено 22.3.2023)
3. Документација за Blynk, <https://blynk.io/developers>, (пристапено 22.3.2023)
4. Документација за библиотеката WifiManager, <https://github.com/tzapu/WiFiManager>, (пристапено 22.3.2023)
5. Документација за CallMeBot сервисот, <https://www.callmebot.com/blog/free-api-whatsapp-messages/>, (пристапено 22.3.2023)
6. Документација за вградените библиотеки во Arduino IDE, <https://www.arduino.cc/reference/en/libraries/>, (пристапено 22.3.2023)
7. Документација за ESP32, <https://www.espressif.com/en/products/socs/esp32>, (пристапено 22.3.2023)
8. 1-Wire протокол, <https://en.wikipedia.org/wiki/1-Wire>, (пристапено 22.3.2023)