

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
PREŠOV

IV.SA - 13

ŠK. R. 2018 – 2019

PČOZ –
AUTOMATIZOVANÝ
SKLENÍK – APLIKÁCIA

FILIP KARNIŠ

Konzultant: Ing. Mária Hedvigová

Anotácia v slovenskom jazyku

Touto prácou sme dosiahli vytvorenie mobilnej aplikácie, ktorá je schopná bezdrôtovo monitorovať a ovládať automatizovaný skleníček. V tejto práci sme v priebehu pár mesiacov dokázali vytvoriť mobilnú aplikáciu, ktorá je využiteľná či už len pre väčšie pohodlie majiteľa skleníčka alebo pre majiteľa, ktorý sa v blízkosti skleníčka nenachádza.

Anotácia v anglickom jazyku

With this work, we have created a mobile application that is able to monitor and control an automated greenhouse wirelessly. In this work, within a few months, we were able to create a mobile application that is usable either for the convenience of the greenhouse owner or for the owner who is not located near the greenhouse.

Čestné vyhlásenie

Vyhlasujem, že celú prácu PČOZ na tému „Automatizovaný skleník - Aplikácia“ som vypracoval samostatne, s použitím uvedenej literatúry.

Som si vedomý zákonných dôsledkov, ak v nej uvedené údaje nie sú pravdivé.

Prešov, 29. apríl 2019

.....

vlastnoručný podpis

Pod'akovanie

Chcel by som sa pod'akovať môjmu konzultantovi Ing. Márii Hedvigovej za všetky konzultácie, za odbornú pomoc pri uskutočňovaní tohto projektu a za povzbudenie v čase, keď na tomto projekte nič nefungovalo.

Obsah

Úvod	6
1 Cieľ práce.....	7
2 Metodika práce	8
3 Úvod do problematiky	9
4 Použité programy	11
4.1 Arduino IDE	11
4.2 Android Studio.....	11
4.3 PhpStorm	12
5 Súčiastky skleníka	13
5.1 Snímače.....	13
5.2 Servomotor.....	14
5.3 Čerpadlo.....	14
6 WiFi	15
6.1 WiFi modul	15
7 HTTP komunikácia.....	17
7.1 PHP	17
8 Android Projekt.....	18
8.1 Manifest	18
8.2 Sources - Zdroje.....	18
8.3 Activities - Aktivita	18
8.4 Resources - Prostriedky	19
8.5 Layout - Rozmiestnenie	19
9 Výsledky a diskusia	20
10 Závery práce	21
Zhrnutie	22
Zoznam použitej literatúry	23

Úvod

Tento projekt som si vybral, pretože ma zaujal iný „*nedokonalý*“ projekt – Automatizovaný skleník.

Tento skleník je schopný zabezpečiť vhodné podmienky pre pestovanie samostatne, a to aj bez prístupu slnečného svetla, vďaka elektrickému osvetleniu. Nápomocný môže byť hlavne ľuďom, ktorí majú málo voľného času, no radi si vychutnajú chuť čerstvej domácej zeleniny či ovocia, no taktiež ľuďom, ktorí často cestujú a doma sa zdržia len na krátko.

No každý z nás nemôže neustále sledovať skleník, ktorý je potrebné kontrolovať aj napriek tomu, že je automatizovaný. Predsa len niekedy musíme odcestovať, niekedy nemáme čas na skleník a vôbec, komu by sa vlastne chcelo neustále kontrolovať či je v skleníku správna teplota? Či je v nádrži dost vody pre čerpadlo? A celkovo ako na tom náš skleník je?

Tento problém nám pomôže vyriešiť tento projekt, vďaka ktorému máme všetko pod kontrolou. Jednoducho si môžeme zobrazit' stav nášho skleníka, nastaviť mu hodnoty alebo mu povedať čo má robiť. A to všetko z pohodlia domova bez nutnosti sa vôbec postaviť, či naopak z druhého konca sveta.

Najskôr sme si museli naštudovať ako daný skleník funguje. S hotovou analýzou sme sa mohli pustiť do vytvárania aplikácie a funkčnej komunikácie s Arduino, mikrokontrolerom, ktorý skleník automatizuje. Pokračovali sme k dizajnu aplikácie, obrázkom a celkovému riešeniu interakcie s používateľom.

Táto práca pozostáva z teoretickej a praktickej časti. V teoretických východiskách sme sa venovali špecifikácii a zdôvodneniu použitia jednotlivých častí a technických súčiastok. V praktickej časti je popísaný postup montáže skleníka a nastavenia technických častí.

1 Cieľ práce

Cieľom našej práce je vytvoriť aplikáciu, vďaka ktorej budeme môcť monitorovať a ovládať automatizovaný skleník.

Táto mobilná aplikácia bude slúžiť na zobrazovanie údajov zo skleníka a na ovládanie skleníka prostredníctvom Internetu. Aplikácia by taktiež mala byť dostupná pre čo najširšiu škálu obyvateľstva.

Ďalším cieľom je navrhnutie prívetivého používateľského rozhrania a menu aplikácie, ktoré budú prehľadné a budú sa v nich vedieť zorientovať aj starší ľudia alebo ľudia, ktorým nie je technika až tak blízka.

Naprogramovanie WiFi modulu ESP8266, aby získaval a odosiela údaje do skleníka prostredníctvom už vytvoreného programu v Arduine.

Naprogramovanie prenosu dát medzi skleníkom a mobilnou aplikáciou pomocou bezdrôtového pripojenia. Vytvorenie súborov na webovom hostingu, ktoré budú obsluhovať HTTP komunikáciu.

Všetko to dať do jedného funkčného celku, aby všetko medzi sebou spolupracovalo a fungovalo bez problémov.

2 Metodika práce

Na začiatku sme urobili analýzu automatizovaného skleníka a jeho komponentov. Následne sme si určili, čo a ako budeme zobrazovať a nastavovať v aplikácii.

V ďalšej etape sme začali študovať ako funguje WiFi modul a samotné Arduino. Tieto informácie sme čerpali najmä z dokumentácií, ale aj z iných zdrojov na Internete ako fóra a príklady použitia. Postupne sme začali skúšať prijímať a odosielať informácie prostredníctvom WiFi modulu. Po úspešnom zrealizovaní sme ešte vyriešili komunikáciu medzi modulom a Arduinom.

Paralelne s týmito krokmi sme začali vyvíjať našu mobilnú aplikáciu. Po rýchlom prieskume sme sa rozhodli, že najlepšie bude vytvoriť aplikáciu pre Android. Pre tieto účely sme čerpali informácie hlavne z Android dokumentácie a onedlho sme začali s vývojom.

Pre prijímanie a odosielanie informácií sme museli pochopiť základy HTTP komunikácie a vytvoriť spojenie s Internetom. Taktiež sme museli preposielať informácie v nejakom špecifickom tvare a rozhodli sme sa pre JSON (JavaScript Object Notation), ktorý je veľmi rozšírený a jednoduchý na pochopenie.

Po zvládnutí prijímania a odosielania jednoduchých hodnôt sme sa mohli pustiť do navrhnutia celkového dizajnu aplikácie a implementovania metód na internetovú komunikáciu do našej používateľsky prívetivej aplikácie.

Nakoniec sme to všetko spojili do jedného celku, otestovali to a zefektívnili naše programy, aby fungovali čo najplynulejšie.

3 Úvod do problematiky

V tejto kapitole sa nachádzajú teoretické poznatky pre túto prácu.

Arduino

Arduino je open-source platforma založená na mikrokontroléri ATMega od firmy Atmel a grafickom vývojovom prostredí, ktoré vychádza z prostredia Wiring (podobný projekt ako Arduino, teda doska s mikrokontrolérom a IDE) a Processing (prostredie pre výuku programovania). Arduino môže byť použité k vytváraniu samostatných interaktívnych zapojení alebo môže byť pripojené k softvéru na počítači (napr. Macromedia Flash, Processing, Max/MSP, Pure Data, SuperCollider). Momentálne možno kúpiť verzie, ktoré sú už skomplementované, schéma a návrh plošného spoja je dostupný pre tých, ktorí si chcú postaviť Arduino sami.

Projekt Arduino získal ocenenie v kategórii digitálnych komunít na Prix Ars Electronica 2006.

Knižnice Arduina

Programovanie Arduina uľahčuje existencia obrovského množstva knižníc. Knižnica je ucelený zdrojový kód, ktorý je obvykle umiestnený na serveri GitHub. Okrem zdrojového kódu v C++ je doplnená metaznačkami, ktoré umožňujú IDE indexovať tieto zdrojové kódy a poskytovať ich na pohodlnú inštaláciu prostredníctvom Manažéra knižníc. Knižnica obvykle poskytuje ucelenú sadu funkcií alebo tried určených na ovládanie konkrétneho hardvéru.

Android

Android je rozsiahla open source platforma, ktorá vznikla najmä pre mobilné zariadenia. Zahŕňa v sebe operačný systém, middleware, používateľské rozhranie a aplikácie.

Ovládanie skleníka sme sa rozhodli urobiť pomocou mobilnej aplikácie, pretože v dnešnej dobe ma so sebou mobilný telefón úplne každý bez ohľadu na to, kde sa nachádza a je to veľmi používateľsky prívetivé riešenie ovládania skleníka na diaľku.

Pre Android sme sa rozhodli pretože má 80% podiel na trhu a stále rastie. Navyše ponúka bezplatný a jednoduchý štart. Všetky nástroje sú bezplatné a dostupné na každej

platforme. Dokonca, ak by sme my nemali zariadenie s Androidom, tak vývojové prostredie ponúka aj softwarový emulátor – virtuálny stroj s ľubovoľnými rozmermi a ľubovoľnou verziou Androidu, čo je skvelé pre testovanie aplikácií.

4 Použité programy

Pri tvorbe projektu sme používali viacero programov, ktoré vám popíšeme v tejto sekcii.

4.1 Arduino IDE

Arduino IDE (Integrated Development Enviroment – Integrované Vývojové Prostredie) je multiplatformová aplikácia (pre Windows, MacOS, Linux), ktorá je napísaná v programovacom jazyku Java. Používa sa na písanie a nahrávanie programov do Arudino kompatibilných dosiek.

Arduino IDE podporuje jazyky C a C++ pomocou špeciálnych pravidiel štruktúrovania kódu. IDE obsahuje softvérovú knižnicu s názvom Wiring, ktorá poskytuje mnoho spoločných vstupných a výstupných postupov.

4.2 Android Studio

Android Studio je oficiálne vývojové prostredie spoločnosti Google založené na IntelliJ IDEA od spoločnosti JetBrains dizajnované špeciálne pre vývoj mobilných aplikácií. Android Studio bolo firmou Google oficiálne predstavené 16. mája 2013 na konferencii Google I/O. Od júna 2013 je zdarma k dispozícii pre používateľov na platformách Windows, Mac, OS X a Linux.

Android Studio slúži na vytváranie mobilných aplikácií pre operačné systémy Android v programovacích jazykoch Java a Kotlin, no v programoch dokáže využívať aj kód napísaný v C a C++.

Aplikácia si však nevystačí len so syntaxou Javy. Potrebuje aj API, teda triedy a metódy, ktoré umožňujú interagovať v programe s platformou Android. SDK obsahuje aj kompilátor zdrojových kódov, ktorý je prepojený s Android Studiom.

4.3 PhpStorm

JetBrains PhpStorm je komerčné multiplatformové IDE (integrated development environment) pre PHP založené na platforme IntelliJ IDEA od spoločnosti JetBrains.

PhpStorm poskytuje editor kódu pre PHP, HTML a JavaScript s kódovou analýzou, prevenciou chýb, rekonštrukciou napísaného kódu a automatickým doplnením kódu (napr. generátory, kľúčové slová, ...).

5 Súčiastky skleníka

Kapitola je venovaná súčiastkam, vďaka ktorým dokážeme skleník monitorovať a ovládať.

5.1 Snímače

Snímač alebo senzor alebo receptor je fyzikálny systém alebo technické zariadenie reagujúce na zmeny meranej veličiny (fyzikálnej, chemickej) okamžite, alebo reaguje na časový priebeh zmeny.

Najbežnejšie snímače sú elektrické, transformujúce informáciu o meranej veličine do inej fyzikálnej oblasti, najčastejšie na elektrický signál alebo na elektrický parameter. Umožňujú transformáciu neelektrickej veličiny na elektrické signály.

Použité snímače

V skleníku bol použitý senzor teploty a vlhkosti AM2301. Je to kapacitný modul na snímanie teploty a vlhkosti vzduchu. Tento modul má vysokú spoľahlivosť a vynikajúcu dlhodobú stabilitu.

Ďalej sa tu nachádzal senzor vlhkosti pôdy, čo je vlhkomer na meranie vlhkosti pôdy. V skleníku bol použitý odporový vlhkomer, ktorého činnosť sa dá popísať takto: pri menšom odpore pôdy je vyššia vlhkosť pôdy.

Nakoniec plavákový senzor. Plavákový senzor hladiny je možné umiestniť na vrchnú aj spodnú časť hladiny. Modul je možné zapojiť aj ako spínací alebo rozpínací kontakt a teda môže ovládať alarm, čerpadlo alebo iný ukazovateľ. Tento senzor funguje na jednoduchom princípe: ak je plavák vo vrchnej polohe, tak v nádrži je dostatok vody a ak plavák klesne do dolnej polohy, tak v nádrži je málo vody.

5.2 Servomotor

Servomotor je otočný alebo lineárny pohon, ktorý umožňuje presné riadenie uhlovej alebo lineárnej polohy, rýchlosti a zrýchlenia. Pozostáva z vhodného motora pripojeného k snímaču polohy.

Použitý Servomotor

Pre skleník sa zvolil servomotor typu MG996R, ktorý je vylepšená verzia MG995. Tento typ ponúka kovové prevody a nárazu vzdorný obal. Tento servomotor sa využíva na otváranie a zatváranie okien.

5.3 Čerpadlo

Čerpadlo alebo pumpa je zariadenie, mechanický stroj, ktorý dodáva kinetickú, potenciálnu alebo tlakovú energiu tekutine, ktorá cez neho preteká. Pomocou čerpadla sa prepravujú tekuté alebo plynné látky.

V princípe čerpadlo využíva podtlak na vstupe pre vtiahnutie tekutiny do zásobníka a následne pretlak pre jej vypustenie. Ako pohon čerpadla sa využíva vonkajšia sila napr. elektrická energia.

Použité čerpadlo

V skleníku bolo použité ponorné čerpadlo značky Barwig, ktorého dopravná výška je 10 metrov, maximálny prietok: 1080 l/h, čerpací tlak: 1 bar, prevádzkové napätie: 12V a prúd od 2,5-3,8 A.

6 WiFi

Wifi je súbor štandardov umožňujúci pripojenie zariadení na bezdrôtovú lokálnu sieť (WLAN), založených na špecifikácii IEEE 802.11. Najčastejšie sa využíva pre pripojenie mobilných zariadení (mobilný telefón, laptop, ...) k Internetu.

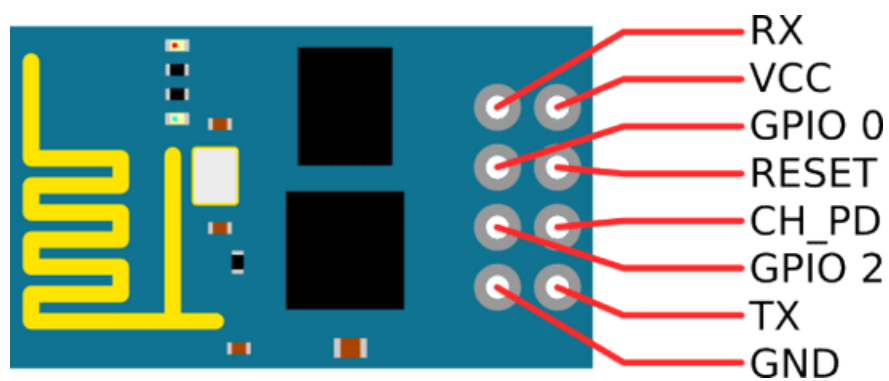
6.1 WiFi modul

Pre prepojenie Arduina s mobilnou aplikáciou sme použili WiFi modul ESP8266-01. ESP8266-01 je mikročip s napájaním 3.3V a s úplnou TCP/IP kompatibilitou. Tento malý modul nám umožňuje pripojenie na WiFi a vytváranie jednoduchých TCP/IP pripojení. Využíva štandard IEEE 802.11 b/g/n.

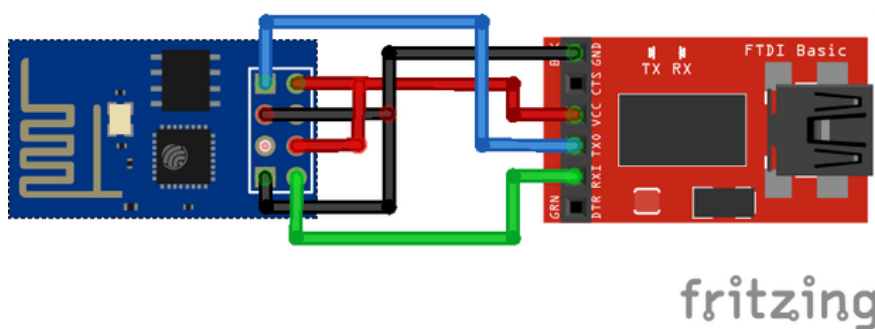
Modul môžeme nastaviť do 3 hlavných módov:

- a) Station (klient) – pripojenie na WiFi a správanie sa ako WiFi klient,
- b) AP (host) – modul slúži ako Access Point, je možné sa naň pripojiť z iných zariadení,
- c) AP+Station – kombinácia týchto 2 módov.

Modul je možné konfigurovať pomocou AT príkazov prostredníctvom sériovej komunikácie, alebo priamym nahraním kódu do modulu napríklad prostredníctvom Arduino IDE. V tomto projekte sme použili druhú uvedenú možnosť. Pred použitím sme museli pripojiť voľne prístupnú knižnicu http://arduino.esp8266.com/stable/package_esp8266com_index.json a následne nainštalovať nový board prostredníctvom board managera v Arduino IDE. Po napísaní programu môžeme program nahráť do modulu vďaka FT232RL Programátoru, ktorý slúži na prevod komunikácie z USB do sériovej formy, ktorá umožňuje programovanie čipov cez UART (RX a TX piny). Je dôležité, aby pri nahrávaní nového kódu bol GPIO0 pin pripojený na zem.



Obrázok 1 ESP8266-01 Pinout (https://i2.wp.com/randomnerdtutorials.com/wp-content/uploads/2015/01/esp8266_pinout_h-569x236.png?resize=569%2C236&ssl=1)



Obrázok 2 Pripojenie ESP8266 k FT232RL (<https://i2.wp.com/randomnerdtutorials.com/wp-content/uploads/2015/02/ESP8266-Flasher.png?ssl=1>)

7 HTTP komunikácia

HTTP (Hypertext Transfer Protocol) funguje na báze odpovedania na request - požiadavku.

GET Request - najčastejšie používaný na získavanie dát od servera pre klienta (napr. web browser – klient si vyžiada web stránku od PC hostujúceho web - servera).

POST Request - najčastejšie používaný na odosielanie dát klienta na server (napr. web browser – klient pošle dáta na prihlásenie prostredníctvom formuláru na PC hostujúci web - server).

Okrem toho existujú ešte PUT, DELETE, HEAD, CONNECT, OPTIONS a TRACE Requesty.

7.1 PHP

Ak by sme sa s aplikáciou napájali priamo na WiFi modul, tak by nám toto pripojenie fungovalo len na 1 spoločnej WiFi sieti, pretože WiFi modul by mal pridelenú len privátnu IP adresu. Pre naše účely to bolo nevyhovujúce, takže sme vytvorili “prostredníka” stránku na free web hostingu 000webhost.com.

Pre lepšie rozlíšenie je PHP rozdelené do 2 súborov. PHP čaká na POST Request, z ktorého získa údaje a zapíše ich vo forme JSON do daného súboru odkiaľ ich neskôr WiFi modul / aplikácia získava pomocou GET Requestu.

Následne komunikácia vyzerala takto:

1. WiFi modul pošle POST Request s údajmi zo senzorov na web, kde sa údaje uložia.
2. Aplikácia si získa údaje cez GET Request na web a zobrazí ich používateľovi.
3. Aplikácia po prijatí údajov od užívateľa pošle POST Request na web, kde sa tieto údaje uložia.
4. WiFi modul kontroluje prostredníctvom GET Requestu na webe, či sa nejaké nastavenia zmenili a následne ich použije v programe Arduina.

8 Android Projekt

Kapitola je venovaná základným častiam každého Android projektu.

8.1 Manifest

Každá aplikácia musí obsahovať `AndroidManifest.xml`, v ktorom sú definované základné informácie o aplikácii pre Android build tools, Android OS a Google Play.

Žiadané informácie:

1. názov balíka – aby build tools vedeli, kde sa nachádzajú ostatné súbory,
2. všetky komponenty aplikácie – definované mená aktivít, služieb, prijímačov, poskytovateľov,
3. práva – aplikácia ich potrebuje na prístup ku chráneným častiam systému alebo iných aplikácií (v našom prípade je to len prístup na Internet),
4. Hardware a software príslušenstvo – aplikácia ich potrebuje pre správne fungovanie (Bluetooth, WiFi, ...).

8.2 Sources - Zdroje

Zdroje sú všetky súbory s kódom, ktoré sú uložené v priečinku “src”. Patria sem aktivity, fragmenty, služby a všetky doplnujúce Java class súbory.

8.3 Activities - Aktivita

Aktivita je samostatná časť, ktorú používateľ môže zobraziť. Takmer všetky aktivity ponúkajú interakciu s používateľom. Aktivita zodpovedá za vytvorenie a zobrazenie okna, kde si môžeme nastaviť náš UI (User Interface) pomocou metódy

setContentView(View). Aktivita sa môže zobrazovať ako celok alebo len ako časť inej aktivity. Aktivitu môžeme spustiť pomocou metódy Context.startActivity().

Je potrebné implementovať metódu onCreate(Bundle), ktorá sa vykoná pri inicializácii aktivity. Najdôležitejšie je tu nastaviť setContentView(int) s layout súborom, v ktorom je definované UI našej aktivity a inicializácia elementov pomocou findViewById(int) kde vytvoríme referenciu na element vytvorený v našom layout súbore pomocou ID. Neskôr môžeme v aktivite nastavovať nášmu elementu parametre pomocou tejto referencie (napr. setText(), setVisibility(), ...), získať z neho dáta pomocou getText() alebo mu vytvoriť onClickListener(), ktorý sa spustí keď používateľ klikne na daný element.

8.4 Resources - Prostriedky

Sú to pripojené súbory a statický obsah, ktoré používa kód. Sú uložené v priečinku “res” a z kódu na nich môžeme referovať pomocou R.“prostriedok”.“názov”. Patria sem ID, layouty, obrázky, farby, text, a všetko čo si sem neskôr pridáme.

8.5 Layout - Rozmiestnenie

Layout definuje naše UI (ako má aktivita a elementy v nej vyzerat'). Je uložený v XML súbore. Do súboru môžeme pridať rôzne typy layoutov (napr. LinearLayout, RelativeLayout, ConstraintLayout, ...), ktoré nám definujú ako sa budú správať deti (elementy alebo ďalšie layouty) v tomto layoute. Následne môžeme týmto elementom nastaviť rôzne vlastnosti (výšku, šírku, farbu, ...). Hlavne im môžeme nastaviť ID, vďaka ktorému sa na nich budeme môcť odkazovať v aktivitách.

9 Výsledky a diskusia

Vytvorenie aplikácie k automatizovanému skleníku sa nám podarilo úspešne ukončiť. Všetko sme stihli úspešne dokončiť do stanoveného termínu.

Na začiatku sme analyzovali samostatný produkt – automatizovaný skleník, aby sme zistili, čo môžeme ovládať a aké sú naše možnosti na dosiahnutie našich cieľov. To zahŕňalo zobrazovanie vonkajšej a vnútornej teploty, vlhkosti pôdy, času a aktuálneho stavu „*ume-
lého*“ slnka. Ovládať sme mohli otváranie a zatváranie okien, zapínanie a vypínanie „*ume-
lého*“ slnka a spustenie zavlažovacieho systému.

Následne sme začali programovať mobilnú aplikáciu, v ktorej sme úspešne využili všetky tieto možnosti a vytvorili sme niekoľko možností pre odosielanie dát.

Po niekoľkých týždňoch sme túto časť dokončili a pustili sme sa do úpravy používateľského rozhrania.

Pôvodný plán spraviť jednoduchú aplikáciu k skleníku sa zmenil na štúdium nových možností, ako zlepšiť už hotové časti projektu a neustále upravovanie výsledného produktu.

Nakoniec sme sa pustili do zefektívnenia programu, pričom sme urýchlili ako chod programu, tak aj posielanie dát bez straty akejkoľvek funkčnosti.

10 Závery práce

Podarilo sa nám vytvoriť plne funkčnú mobilnú aplikáciu pre operačný systém Android, ktorá slúži na vzdialené ovládanie automatizovaného skleníka.

Skleník obsahuje množstvo senzorov, ktorých výstup je vypisovaný v mobilnej aplikácii, ktorá komunikuje so skleníkom pomocou WiFi. Taktiež využíva niekoľko ďalších komponentov, ktoré môžeme pomocou nami vytvorenej aplikácie ovládať.

Nami vytvorená aplikácia uľahčí život majiteľovi automatizovaného skleníka, upozorní ho na nebezpečné hodnoty v skleníku a umožní mu ľahko zaspať aj keď sa nachádza na druhej strane zemegule.

Zhrnutie

Prácou sa nám podarilo zrealizovať plne funkčnú mobilnú aplikáciu, ktorá komunikuje so skleníkom prostredníctvom WiFi modulu. Môžeme prehlásiť, že všetky ciele, ktoré sme si určili boli taktiež splnené. Hlavnými cieľmi bolo dosiahnuť: efektívny program Arduina, jednoduchá a prehľadná mobilná aplikácia a efektívna bezdrôtová komunikácia prostredníctvom Internetu.

Zoznam použitej literatúry

<https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>

<https://www.ecoprodukt.sk/solarne.cerpadla/dc.cerpadla.12v/ponorne.cerpadla.12v/ponorne.cerpadlo.barwig.typ.01.12v.dc.18lmin>

<http://ics.upjs.sk/~novotnyr/android/>

<https://randomnerdtutorials.com/flashing-nodemcu-firmware-on-the-esp8266-using-windows/>

<https://techfun.sk/>

[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

https://en.wikipedia.org/wiki/Android_Studio

<https://en.wikipedia.org/wiki/ESP8266>

<https://en.wikipedia.org/wiki/Servomotor>

<https://sk.wikipedia.org/wiki/Arduino>

<https://sk.wikipedia.org/wiki/Sn%C3%ADma%C4%8D>

<https://sk.wikipedia.org/wiki/Vlhkomer>