

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Automatický zavlažovací systém

Ondřej Pecháček

© 2021 ČZU v Praze



Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce: Ondřej Pecháček
Studijní program: Systémové inženýrství a informatika
Obor: Informatika
Vedoucí práce: Ing. Marek Pícka, Ph.D.
Garantující pracoviště: Katedra informačního inženýrství
Jazyk práce: Čeština

Název práce: **Automatický zavlažovací systém**

Název anglicky: **Automatic irrigation system**

Cíle práce: Cílem práce je vytvořit lokální automatický zavlažovací systém pro jeden květináč. Systém bude obsahovat řídicí jednotku v podobě mikrořadiče, soustavu pro automatické zavlažování řízenou čidlem vlhkosti půdy a čidla pro měření různých údajů jako například teplotu.

Metodika: První částí práce bude studium relevantní literatury pro získání potřebných znalostí pro zhotovení samotné práce. Následovat bude prostudování dostupných mikrořadičů z řady Arduino a výběr vhodného mikrořadiče pro tuto práci. Následně proběhne průzkum a výběr ostatních komponent potřebných pro tuto práci. Poté proběhne zhotovení samotného systému. Po zhotovení systému proběhne jeho testování v domácím prostředí.

Doporučený rozsah práce: 30-40 stran

Klíčová slova: Arduino, Mikrořadiče, Zavlažovací systém, Automatizace zemědělství

Doporučené zdroje informací:

1. DUKISH, Bod. Coding the Arduino: Building Fun Programs, Games, and Electronic Projects. Apress, 2018. ISBN 978-14-84235-10-2.
2. KARVINEN, Kimmo a Terro KARVINEN. Getting Started with Sensors: Measure the World with Electronics, Arduino, and Raspberry Pi. Maker Media, 2014. ISBN 978-14-49367-08-4
3. OSBORNE, William. Learn to Program in Arduino C. 18 Lessons, From setup() to Robots. CreateSpace, 2017. ISBN 978-09-98128-71-9.
4. VODA, Zbyšek a tým HW kitchen. Průvodce světem Arduina. 2. vydání. Martin Stříž, 2017. ISBN 978-80-87106-90-7.

Předběžný termín obhajoby: 2020/21 LS - PEF

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Automatický zavlažovací systém" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou v práci citovány, a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.03.2021

Ondřej Pecháček

Poděkování

Rád bych tímto poděkoval svému vedoucímu práce Ing. Marku Píckovi, Ph. D za pomoc a konzultace při psaní této bakalářské práce.

Automatický zavlažovací systém

Abstrakt

Cílem práce je vytvoření zavlažovacího systému založeného na mikrořadiči Arduino. Teoretická část se zabývá analýzou vhodných komponent pro tuto práci a výběrem vhodných komponent pro zhotovení práce. Dále je popsán mikrořadič Arduino, jeho základní části a parametry. Praktickou část tvoří samotný návrh systému, jeho zkonstruování, vytvoření programu pro řízení pro mikrořadiče a ověření funkčnosti systému.

Klíčová slova: Arduino ,Mikrořadiče, Zavlažovací systém, Automatizace zemědělství, Arduino UNO, LCD displej, Půdní vlhkoměr, Domácí zavlažování

Automatic irrigation system

Abstract

The aim of the work is to create an irrigation system based on the Arduino microcontroller. The theoretical part deals with the analysis of suitable components for this work and the selection of suitable components for the work. The Arduino microcontroller, its basic parts and parameters are also described. The practical part consists of the design of the system itself, its construction, the creation of a control program for microcontrollers and verification of system functionality.

Keywords: Arduino, Microcontrollers, Irrigation system, Agricultural automation, Arduino UNO, LCD display, Soil moisture meter, Home irrigation

Obsah

1. Úvod.....	6
2. Cíl práce a metodika.....	7
2.1. Cíl Práce	7
2.2. Metodika	7
3. Teoretická východiska	8
3.1. Možnosti automatizace závlahových systémů	8
3.2. Možnosti konstrukce závlahových systémů.....	8
3.2.1. Zisk vody ze zdroje.....	8
3.2.2. Doprava vody.....	9
3.3. Platforma Arduino.....	9
3.3.1. Alternativy k platformě Arduino	10
3.4. Arduino UNO.....	10
3.4.1. Rozložení součástek na desce	11
3.4.2. Architektury počítačů a architektura mikrořadiče ATmega328P.....	12
3.4.3. Typy paměti a paměti v mikrořadiči ATmega328P.....	13
3.4.4. Popis pinů na desce Arduino UNO.....	15
3.4.5. PWM (Pulse width modulation)	18
3.4.6. SPI (Serial peripheral interface)	18
3.4.7. I2C protokol	18
3.5. Arduino IDE a programování v něm.....	19
3.5.1. Knihovny pro Arduino IDE	20
4. Vlastní práce.....	21
4.1. Použité komponenty.....	21
4.1.1. Arduino UNO	21
4.1.2. Půdní vlhkoměr pro jednodeskové počítače	21
4.1.3. Ponorné čerpadlo mini 120 L/H.....	22
4.1.4. LCD displej 1602A	23
4.1.5. Teplotní senzor LM35	24
4.2. Zapojení a testování půdního vlhkoměru	25
4.3. Zapojení a otestování ponorného čerpadla.....	26
28	
4.4. Zapojení a testování teploměru a LCD displeje.....	28
30	
4.5. Konečné zapojení systému	31
4.6. Řídící program systému	32

5. Zhodnocení.....	35
6. Závěr	36
7. Zdroje	37

Seznam obrázků

Obrázek č. 1 : Rozložení desky.....	11
Obrázek č. 2 : Napájecí piny.....	15
Obrázek č. 3 : Analogové piny.....	16
Obrázek č. 4 : Digitální piny.....	17
Obrázek č. 5 : Ukázkový kód.....	19
Obrázek č. 6 : Správce knihoven v Arduino IDE.....	20
Obrázek č. 7 : Půdní vlhkoměr.....	22
Obrázek č. 8 : ponorné čerpadlo.....	23
Obrázek č. 9 : LCD displej.....	24
Obrázek č. 10 : LM35 teploměr.....	25
Obrázek č. 11 : Schéma zapojení vlhkoměru.....	26
Obrázek č. 12 : Program pro test vlhkoměru.....	27
Obrázek č. 13 : Schéma zapojení čerpadla.....	28
Obrázek č. 14 : Program pro test čerpadla.....	29
Obrázek č. 15 : Schéma zapojení teploměru a LCD displeje.....	30
Obrázek č. 16 : Počáteční část programu pro test teploměru a displeje.....	30
Obrázek č. 17 : Funkce void setup() programu pro test teploměru a displeje.....	31
Obrázek č. 18 : Funkce void loop() programu pro test teploměru a displeje.....	31
Obrázek č. 19 : Schéma konečného zapojení.....	32
Obrázek č. 20 : Obrázek hotového systému.....	33
Obrázek č. 21 : Počáteční část konečného programu.....	33
Obrázek č. 22 : Funkce void setup() konečného programu.....	34
Obrázek č. 23 : Funkce void loop() konečného programu.....	35

Seznam použitých zkratek

USB – Universal serial bus – specifikace pro určitý druh komunikace mezi počítači

IDE – Integrated development enviroment – vývojové prostředí pro programátory

LED – Light emitting diode – svítivá dioda

ICSP – In system serial programming – označení pro schopnost programovat mikročipy, které jsou už nainstalovány v kompletních zařízeních

GND – GrouND – označení pro zemnicí piny

VCC – Voltage common collector – označení pro napájecí piny

LCD – Liquid Crytal Display – typ displeje, který využívá světlo-ovlivňující vlastnosti tekutých krystalů pro zobrazení

1. Úvod

Pěstování rostlin v domácím prostředí není příliš složitá záležitost, většina rostlin vyžaduje pro svůj růst pouze pravidelný přísun vody a sluneční paprsky. Skoro každý člověk má u sebe doma alespoň pár květináčů s okrasnými rostlinami. Starat se o takovéto rostliny je velice jednoduché a samotné rostliny mají pozitivní vliv na vzhled domácnosti. Lidé, kteří žijí na vesnici obvykle mívají celé zahrady nebo záhony, kde si pěstují domácí plodiny pro vlastní konzumaci nebo uchování do budoucna. Pravidelně zalévat takovéto plochy může být zdoluhavé a pro některé osoby i fyzicky namáhavé. Taky se může stát, že na zalití svých záhonů člověk zapomene, a to povede v některých případech ke ztrátě rostlin.

Automatizované zavlažovací systémy řeší tento problém tím, že kompletně eliminují nutnost pravidelného zalévání. U většiny těchto systémů postačí pouze občasné zkontrolování zdroje vody pro systém nebo jeho doplnění. Některé systémy jsou navrženy tak, že si mohou brát vodu třeba ze studny, a tím kompletně eliminovat nutnost pravidelné kontroly. Díky tomu se osobám, které využívají tyto systémy u větších ploch, uvolní čas, který by jinak strávili zaléváním. Zalévání je také vcelku jednoduchá činnost a lze ji snadno automatizovat.

2. Cíl práce a metodika

2.1. Cíl Práce

Cílem práce je vytvořit lokální automatický zavlažovací systém pro jeden květináč. Systém bude obsahovat řídicí jednotku v podobě mikrořadiče, soustavu pro automatické zavlažování řízenou čidlem vlhkosti půdy a čidla pro měření různých údajů, například teplotu.

V teoretické části práce je cílem vyhledat potřebné komponenty ke zhotovení práce a vybrat mikrořadič z řady Arduino pro zhotovení práce.

V praktické části dojde k samotnému sestavení systému a jeho otestování.

2.2. Metodika

První částí práce bude studium relevantní literatury pro získání potřebných znalostí pro zhotovení samotné práce. Následovat bude prostudování dostupných mikrořadičů z řady Arduino a výběr vhodného mikrořadiče pro tuto práci.

Poté proběhne průzkum a výběr ostatních komponent potřebných pro tuto práci. Na to proběhne otestování funkčnosti jednotlivých komponent. Následně proběhne konečné sestavení celého systému a vytvoření ovládacího programu pro něj. Nakonec proběhne testování funkčnosti celého systému.

3. Teoretická východiska

3.1. Možnosti automatizace závlahových systémů

Existuje mnoho způsobů jak vytvořit automatický systém pro zavlažování. Vše závisí na konkrétních požadavcích a podmínkách každého systému. Závlahové systémy se mohou lišit od jednoduchých soustav určených pro skleníky až po velké zemědělské systémy, které mohou pokrýt velké půdní plochy. Mezi další rozdíly patří způsob provedení systému a jeho použití a také cena.

Automatizované systémy mají mnoho výhod oproti manuálnímu zavlažování. Největší z nich je eliminace lidského faktoru. Člověk, který se stará o vlastní záhony či skleníky, může na zalití prostě zapomenout. Toto může vést obzvláště u citlivých rostlin k jejich poškození z nedostatku vody a někdy i k jejich zahubení.

Automatické systémy tento problém nemají, jelikož pracují s vlastními hodinami, kde po určité uplynulé době se systém sám spustí nebo s půdními senzory vlhkosti, kde se systém spustí po propadu vlhkosti pod hranici určenou řídicím programem. (2)(3)

3.2. Možnosti konstrukce závlahových systémů

Konstrukce automatických závlahových systémů se nejvíce liší ve dvou oblastech. Tyto oblasti jsou zisk vody z vodního zdroje a samotná doprava vody do místa určení

3.2.1. Zisk vody ze zdroje

Prvním a nejrozšířenějším způsobem je čerpání vody pomocí vodního čerpadla. Tato čerpadla mohou být ponorná, která se celá ponoří do vody, nebo pozemní samonasávací, kde čerpadlo zůstává na zemi a do vodního zdroje vede jenom hadice.

Hlavní výhodou použití vodního čerpadla je vysoký tlak vody. Díky tomu čerpadla vždy dopraví vodu na místo určení i například do kopce, pokud to konkrétní systém vyžaduje. Největší nevýhodou je zvýšená cena celého systému. Také pokud se čerpadlo rozbije, je ho potřeba nahradit novým, což ještě více zvýší náklady na systém.

Druhým způsobem je doprava vody samospádem. U tohoto způsobu musí být zdroj vody na vyšším místě, než kam chceme vodu dopravit. Voda je poté dopravena místo určení vlastním tlakem. Hlavní výhodou tohoto způsobu jsou nižší náklady na systém a jeho údržbu. Mezi nevýhody patří nízký tlak proudu vody. Díky tomu tento způsob může mít problémy pokrýt větší plochy. (4)

3.2.2. Doprava vody

Prvním způsobem je podpovrchové zavlažování. Při tomto způsobu je voda vedena podzemním potrubím přímo ke kořenům rostlin. Takovýto systém je složitý na instalaci a údržbu, ale na druhou stranu povrch nebude rozmočený, protože voda je přiváděna zespoda. Tento systém je také využíván pro okrasné zahrady, protože samotné potrubí je schované v podzemí.

Druhým způsobem je zavlažování postřikem. Při tomto způsobu je voda dopravena potrubím na místo určení, kde je rozstřikovače roznesou na danou oblast. Při takovémto způsobu musí být voda přiváděna pod vysokým tlakem, aby rozstřikovače mohly pokrýt oblast určení.

Třetím způsobem je zavlažování lokalizované, kde vodu přivedeme ke spodkům rostlin pomocí děrovaného potrubí položeného na povrchu. Tento způsob je nejjednodušší na instalaci a údržbu. (2)(3)(5)

3.3. Platforma Arduino

Arduino je jednoduchá otevřená elektronická platforma pro vývoj různých projektů založená na jednoduchém hardwaru a softwaru. Tato platforma vznikla v roce 2005 ve městě Ivrea v Itálii. Cílem tvůrců této platformy (David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis, a Massimo Banzì) bylo vyrobit jednoduchou, cenově nenákladnou platformu se zaměřením na méně technicky zdatné osoby. Samotná platforma je dodávána s jednoduchým IDE založeným na open source licenci a je programovatelná přes USB.

Díky těmto aspektům je Arduino určeno nejen pro firmy zabývající se mikroprocesorovou technologií, ale i pro zájmové konstruktéry a studenty. Arduino je schopné vnímat okolí pomocí vstupů různých senzorů a ovlivňovat jej díky připojení různých výstupních periférií jako jsou diody, malé motory či obrazovky.

Samotný mikročip na desce Arduino je založen na rodině AVR mikročipů od firmy Atmel. Mikročip na Arduinu se programuje pomocí speciálního programovacího jazyka (založený na jazyku Wiring, který je podobný jazyku C) ve vlastním vývojovém prostředí.

Desky Arduino je možné si koupit již sestavené, nebo si lze koupit součástky a ručně si sestavit vlastní desku. Návrhy Arduino desek jsou k dispozici pod otevřenou licenci a lze je tedy upravovat podle vlastních potřeb. Arduino má nyní již více jak 10 modelů. (1)

3.3.1. Alternativy k platformě Arduino

Jak už je dnes zvykem žádná technologie nezůstane na trhu sama a dříve nebo později se objeví její kopie. V dnešní době existuje mnoho tzv. „duin - klonů“. Hlavním důvodem je to, že samotné Arduino je plně open source, tedy jeho hardware, firmware a IDE je plně dostupné pro všechny.

Díky tomu si kdokoliv může platformu Arduino jakkoliv upravit, zjednodušit či vylepšit. Ovšem nesmí použít název Arduino, který je pod ochrannou známkou. Na trhu existuje mnoho variant platformy Arduino. Většina z nich nese konec názvu Arduino (-duino) a změní první část názvu např.: „Freeduino“ , „Boardduino“ či „Seeduino“.

Tento styl pojmenování dává najevo, že tyto produkty jsou kompatibilní s oficiálními Arduino produkty. Míra kompatibility se mění od produktu k produktu.

Arduino klony mají také mnoho výhod. Jednou z nich je cena. Některé kopie jsou vyrobeny ze součástek nižší kvality. Díky tomu jsou levnější, ale také rizikovější na použití. Další z těchto výhod je specializace pro různé projekty, pro které se oficiální arduino desky nemusí hodit. Například platforma LilypadArduino, která má tvar kruhu o průměru 50 milimetrů, a je díky tomu vhodná pro projekty, které vyžadují úspornost prostoru. (6)(1)

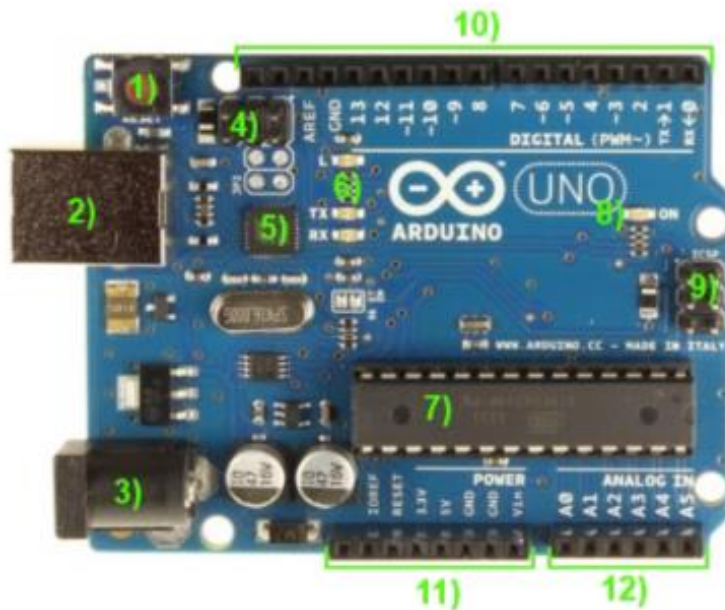
3.4. Arduino UNO

Pro tuto práci byla zvolena platforma Arduino UNO.

Arduino UNO je zdaleka nejrozšířenější verze platformy Arduino. Tato platforma je nástupce vývojových řad se sériovým portem. Mezi tyto řady patří například Arduino Extreme nebo Arduino NG.

Samotné Arduino UNO se stalo základem pro mnoho variant jako například Arduino Ethernet, který se dokáže připojit do sítě přes zabudovaný ethernet konektor nebo Arduino Bluetooth, který umožňuje komunikaci přes technologii Bluetooth. (1)

3.4.1. Rozložení součástek na desce



Obrázek č. 1 : Rozložení desky

- 1) Tlačítko Reset – po stisknutí tohoto tlačítka se zastaví chod běžícího programu a celá deska se restartuje. Před znovu spuštěním by měla LED dioda několikrát zablikat (číslo 6 na obrázku). Poté se automaticky spustí program nahraný na desce. Stejného efektu lze dosáhnout odpojením a znovu připojením USB kabelu (číslo 2 na obrázku)
- 2) USB port typ B – tento port slouží ke komunikaci s počítačem a nahrávání programů na desku. USB port také slouží jako hlavní napájení desky. Standardní napětí a proud, který USB port dodává, je 5V a 500 mA.
- 3) Napájecí konektor – tento konektor slouží pro napájení v případě že Arduino není připojené k počítači. Do tohoto konektoru lze zapojit baterii nebo napájet desku přímo ze sítě pomocí adaptéru. Na rozdíl od napájení přes USB lze přes tento konektor ovlivnit napětí. Standardní napájení pro desku uvedené výrobcem je 7 – 12 voltů. Limit pro napájení je 6 – 20 voltů. Při napětí, které překročí 20 voltů existuje riziko přehřátí a trvalého poškození desky.
- 4) ICSP piny – ICSP (In Circuit Serial Programming) piny slouží k přímému programování USB / Serial převodníku

- 5) USB / Serial převodník – Tato součástka slouží k převodu sériové komunikace na desce na USB standard, tento převod je nezbytný pro komunikaci s počítačem.
- 6) LED diody L, Rx, Tx. Diody L lze využít místo externí LED diody protože je připojena na výstup číslo 13. Rx a Tx diody signalizují komunikaci přes sériovou linku.
- 7) 8 - bitový procesor ATmega328P. Hlavní řídicí jednotka desky Arduino.
- 8) Diody, která slouží k indikaci připojeného napájení
- 9) ICSP piny pro ATmega328P – tyto ICSP piny slouží k přímému programování mikrořadiče.
- 10) Digitální výstupní piny - tyto piny slouží ke komunikaci s připojenými periferiemi.
- 11) Napájecí výstupní piny - tyto piny slouží k napájení připojených periferií
- 12) Analogové vstupní piny

Zdroje: (7)(8)

3.4.2. Architektury počítačů a architektura mikrořadiče ATmega328P

Dnes existují dvě hlavní architektury počítačů, Harvardská a von Neumanova. Hlavní rozdíl mezi těmito architekturami spočívá v jejich pamětech. Harvardská architektura má společnou paměť jak pro program tak i pro data. Von Neumanova má paměť pro program a data oddělenou.

Dalším rozdělením architektur je dělení podle způsobu zpracování instrukcí na architektury RISC a CISC.(1)(9)

3.4.2.1. Architektura RISC

Název této architektury (přeloženo do češtiny „počítač se sníženou instrukční sadou“ je zavádějící, přestože jeho překlad může působit tak, že tato architektura má omezený počet instrukcí oproti architektuře CISC, není tomu tak. Některé počítače založené na RISC architektuře mohou mít více instrukcí než počítače založené na CISC architektuře.

Architektura RISC se zaměřuje na zjednodušení a optimalizaci jednotlivých instrukcí pro jejich nejrychlejší vykonání a co nejmenší nároky na paměť. Všechny instrukce v jednom RISC jsou obvykle stejné délky (nejčastěji 32 bitů). (9)

3.4.2.2. Architektura CISC

Na rozdíl od RISC má tato architektura složitější instrukce, kde každá instrukce může vykonat několik jednodušších operací po sobě např. zápis do paměti, čtení z paměti, nebo aritmetický výpočet. Název této architektury (přeloženo do češtiny „počítač s komplexní instrukční sadou“) byl zaveden po představení architektury RISC a představuje pojem, pod který spadají všechny počítače, které nejsou založeny na architektuře RISC. (9)

Mikrořadič ATmega328P je založen na architektuře AVR. Tato architektura je modifikace harvardské RISC architektury, která byla vytvořena dvěma studenty Norského institutu technologií. AVR byla jedna z prvních architektur, která využila ve svých mikrořadičích paměť FLASH jako hlavní paměť, na rozdíl od podobných mikrořadičů z té doby, které dosud používali paměti ROM, EPROM nebo EEPROM. (9)(25)

3.4.3. Typy pamětí a paměti v mikrořadiči ATmega328P

Paměti jsou nedílnou součástí každého počítače, které slouží pro ukládání dat a softwaru. Pamětí existuje velké množství a dělí se podle různých vlastností, například podle kapacity, výrobní technologie nebo schopnosti uchování informace po odpojení napájení. Jedním z nejzákladnějších dělení je na energeticky závislé a nezávislé paměti.

Energeticky nezávislé paměti si uchovávají data i po odpojení napájení.

Typy těchto pamětí:

- 1) Paměť ROM (Read only memory) – obsah této paměti je dán při výrobě, nelze pak její obsah měnit, ale pouze číst.
- 2) Paměť PROM (Programmable read only memory) – tuto paměť lze naprogramovat pouze jednou, poté její obsah nejde změnit ani smazat.
- 3) Paměť EPROM (Erasable programmable read only memory) – tuto paměť si může naprogramovat uživatel. K programování se používá speciální programátor. Tyto paměti mají malé křemenové okno, přes které jejich obsah lze smazat s pomocí Ultrafialového záření.
- 4) Paměť EEPROM (Electrically erasable programmable read only memory) – paměť, která jde programovat i mazat. Na rozdíl od EPROM se k mazání nepoužívá UV záření. Tuto paměť lze mazat i programovat v systému kde je paměť zapojena, ale programování i mazání je pomalé.

- 5) Paměť FLASH – Programování a mazání je stejně možné jako u EEPROM. Rozdíl je v mazání dat, které je u tohoto typu podstatně rychlejší. Existují dva typy založené na logických obvodech NOR a NAND.

6)

Energeticky závislé paměti potřebují ke svému provozu stálý přísun elektrického napětí. Při ztrátě napětí tyto paměti ztrácí svůj obsah.

Typy těchto pamětí.

- 1) Paměť RAM (Random access memory) – volatilní paměť, která je při spuštění prázdná a při zastavení napájení se její obsah ztratí. Existují dva typy RAM:
- a) SRAM (Static random access memory) – paměť, která používá klopné obvody pro ukládání jednotlivých bitů. Data zůstávají zapsána, dokud nedojde ke změně dat.
 - b) DRAM (Dynamic random access memory) – tato paměť využívá buňky složené z kondenzátoru a tranzistoru k uložení jednotlivých bitů. Na rozdíl od SRAM tato paměť vyžaduje opakované obnovení náboje v kondenzátoru jinak dojde ke ztrátě dat.

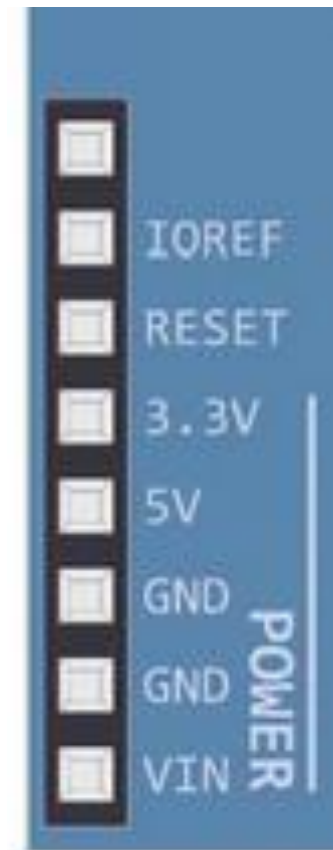
Zdroje: (10)(11)

Mikrokontroler Atmel ATmega328P obsahuje FLASH paměť s kapacitou 32 kB. 0.5 kB této paměti je rezervováno pro BootLoader. Bootloader je program, který zajišťuje kompatibilitu s Arduino IDE na počítačích. Zbýlých 31.5 kB je využito pro uložení programu uživatele.

ATmega328P dále obsahuje 2 kB paměti SRAM, kde se ukládají proměnné používané programem, a 1 kB paměti EEPROM kde může uživatel pomocí programu ukládat informace.(1)

3.4.4. Popis pinů na desce Arduino UNO

3.4.4.1. Napájecí piny

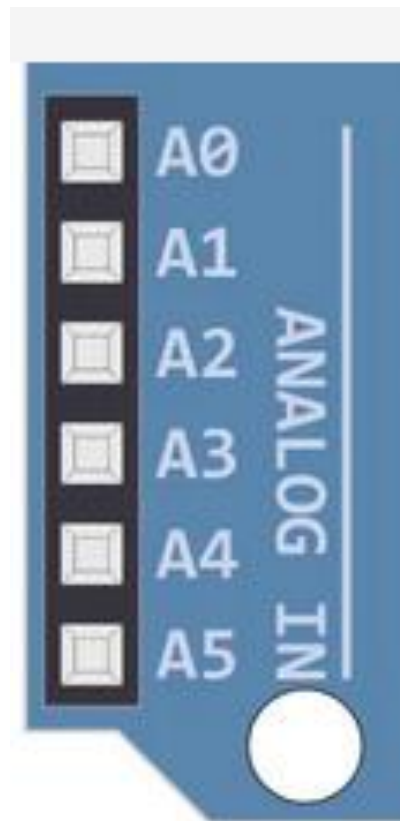


Obrázek č. 2 : Napájecí piny

- 1) IOREF – poskytuje referenci napětí se kterým mikrořadič na desce pracuje
- 2) RESET – pomocí tohoto pinu lze restartovat celou desku.
- 3) 3.3V – tento pin poskytuje napájení 3.3V pro připojené periferie
- 4) 5V - tento pin poskytuje napájení 5V pro připojené periferie
- 5) GND – tyto dva piny slouží k uzemnění
- 6) VIN – tento pin lze použít k napájení desky, napětí by mělo být v rozmezí 6 – 20V
- 7) neoznačený pin nad pinem IOREF není připojen a neslouží žádnému účelu

Zdroje:(12)

3.4.4.2. Analogové vstupní piny



Obrázek č. 3 : Analogové piny

Všechny piny využívají konverzi analogového signálu na digitální (ADC – Analog/Digital Converter), tedy dokážou přečíst analogový vstup (hodnotu napětí) a přeměnit ho na digitální hodnotu, kterou potom může mikrořadič na desce pochopit a dále s těmito hodnotami pracovat. Tuto funkci využívá mnoho periferních zařízení, která reprezentují svá data jako napětí jako například mikrofony. ADC na této desce má rozlišení 10 bitů. Těchto 10 bitů dává 1024 úrovní, kterými může ADC reprezentovat vstupní napětí. Dále všechny piny mohou fungovat jako digitální vstupy a výstupy. (12)

3.4.4.3. Digitální piny

- 1) Piny 0 -13 slouží jako digitální vstupy a výstupy.
- 2) Piny 3, 5, 6, 9, 10 a 11 mají schopnost PWM (Pulse width modulation)
- 3) Piny 0 a 1 (Rx a Tx) lze použít k sériové komunikaci. Pomocí těchto pinů může deska navázat komunikaci s jinými sériovými zařízeními například počítači.
- 4) Piny 10 – 13 mají možnost SPI komunikace.
- 5) GND pin slouží k uzemnění.
- 6) Pin AREF a dva neoznačené piny nad ním slouží ke komunikaci přes I2C protokol

Zdroje:(12)



Obrázek č. 4 : Digitální piny

3.4.5. PWM (Pulse width modulation)

PWM je technika modulace, kde je zpráva zakódována jako pulzní signál. PWM má dva základní prvky, frekvenci a délku výšky cyklu.

Frekvence určuje, jak dlouho trvá dokončení jednoho cyklu (pulzu) a jak dlouho trvá změna signálu z nízké na vysokou hodnotu (z 0 na 1).

Délka výšky cyklu určuje, jak dlouho zůstane signál na vrcholu v jednom cyklu. Tato hodnota je reprezentována jako procento z celkové délky cyklu.

U Arduina je frekvence pevně daná na 500 Hz a délka výšky cyklu je daná programem uživatele. PWM signály se používají například u ovládání rychlosti motorů nebo jasnosti LED diody.(12)

3.4.6. SPI (Serial peripheral interface)

SPI je komunikační protokol, který umožňuje mikrořadiči komunikovat s jedním nebo více externími zařízeními, nebo ke spojení dvou mikrořadičů. Při SPI komunikaci je jeden mikrořadič označen jako Master a ostatní zařízení jako Slave.

Každé zařízení schopné SPI má následující piny.

- 1) MISO (Master In Slave Out) – pin sloužící k posílání dat Master zařízení (pin 12)
- 2) MOSI (Master Out Slave In) – pin sloužící k posílání dat periferiím (pin 11)
- 3) SCK (Serial Clock) – pin pro přenos hodinového signálu generovaného Master zařízením. Slouží k synchronizaci komunikace a datového přenosu mezi zařízeními.

Zdroje:(12)

3.4.7. I2C protokol

Modernější sériový komunikační protokol. Využívá pouze 3 piny.

- 1) AREF – poskytuje referenční napětí pro analogové vstupy.
- 2) SDA (první neoznačený pin nad pinem AREF) – tento pin je využíván pro přenos dat
- 3) SCL (druhý neoznačený pin nad pinem AREF) -tento pin slouží jako hodinový signál k synchronizaci komunikace

Zdroje:(12)

3.5. Arduino IDE a programování v něm

Arduino IDE (Integrated development environment) je vývojové prostředí, které slouží k psaní a nahrávání programů na Arduino desky. Arduino IDE je napsáno v programovacím jazyku Java a jeho zdrojový kód je volně dostupný.

V Arduinu IDE využívá programovací jazyk Wiring, který je postaven na jazycích C a C++. Jazyk Wiring byl vyvinut pro Processing IDE. Arduino IDE vychází z tohoto vývojového prostředí a převzalo z něj tento jazyk.(14)

Programy v jazyce Wiring mají dvě hlavní funkce.

- 1) *void setup()* – tato funkce se spustí pouze jednou na začátku programu a zpravidla obsahuje počáteční nastavení
- 2) *void loop()* – tato funkce se spustí po ukončení *void setup()*, a je automaticky opakovaně volána, dokud je deska Arduino připojená k napájení

Následující obrázek ukazuje jednoduchý Kód, který rozbliká LED diodu na desce
Zdroje:(14)

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

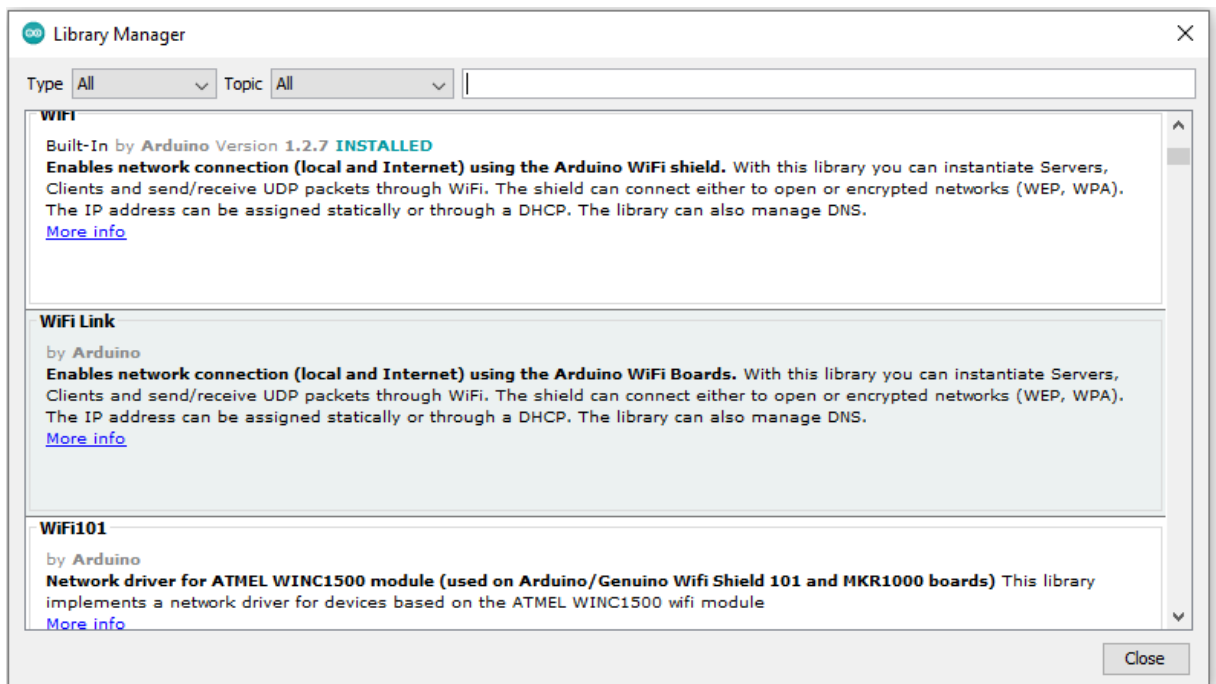
Obrázek č. 5 : Ukázkový kód

3.5.1. Knihovny pro Arduino IDE

Samotné Arduino IDE obsahuje mnoho procedur a příkazů pro programování a Arduina a jeho periférií. Ovšem existuje tak mnoho periferních zařízení a s průběhem času jsou navrhována a vyráběna další, že je téměř nemožné všechny tyto zařízení správně pokrýt programovým vybavením pro jejich ovládání.

Řešením tohoto problému jsou knihovny, pomocí kterých lze rozšířit možnosti určitého programu. Většina knihoven umožňuje Arduino komunikovat a ovládat určité periferie. Příkladem je např. knihovna *LiquidCrystal.h*, která umožňuje Arduino ovládat LCD displeje. Jiné knihovny například umožňují komunikaci s chytrými telefony nebo jinými zařízeními.

Existuje také možnost napsat si vlastní knihovny. Tato funkce je využívána převážně programátory, kteří si chtějí vytvořit vlastní sadu procedur a funkcí. (1)(14)



Obrázek č. 6 : správce knihoven v Arduino IDE

4. Vlastní práce

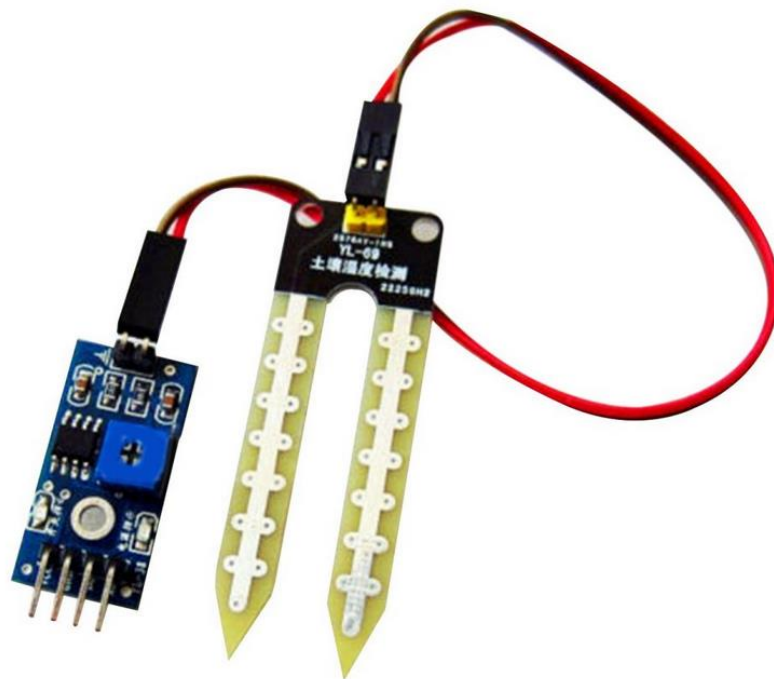
4.1. Použité komponenty

4.1.1. Arduino UNO

Pro srdce systému byla deska Arduino UNO rev3 (třetí revize). Detailní popis této desky byl proveden v kapitole Arduino UNO.

4.1.2. Půdní vlhkoměr pro jednodeskové počítače

Tento půdní vlhkoměr pracuje na principu měření odporu mezi dvěma vidlicemi sondy. Čím vyšší odpor, tím nižší vlhkost, rozsah měřeného odporu je 0 - 1023. Modul vlhkoměru obsahuje samotnou sondu a převodník určený ke komunikaci s jednodeskovými počítači. Samotný vlhkoměr má analogový i digitální výstup, kde analogový výstup poskytuje přesnou hodnotu odporu a digitální výstup 0 nebo 1. U digitálního výstupu záleží na tom, zda naměřená hodnota na sondě překročila hranici, kterou lze nastavit trimrem na převodníku. Pracovní napětí je 3.3 – 5 V. (15)



Obrázek č. 7 : Půdní vlhkoměr

4.1.3. Ponorné čerpadlo mini 120 L/H

Toto čerpadlo pracuje se stejnosměrným napětím 2.5 – 6 V, tedy nepotřebuje žádné externí napětí a lze ho napájet přímo z Arduino desky. Maximální proud je 220 mA. Průtok čerpadla je 80 - 120 litrů za hodinu a zdvih vody 40 – 110 centimetrů. Tento model využívá rotační odstředivou sílu ke zdvihu kapaliny. Rozměry čerpadla jsou 33 milimetrů na výšku a 45 milimetrů na délku.(17)



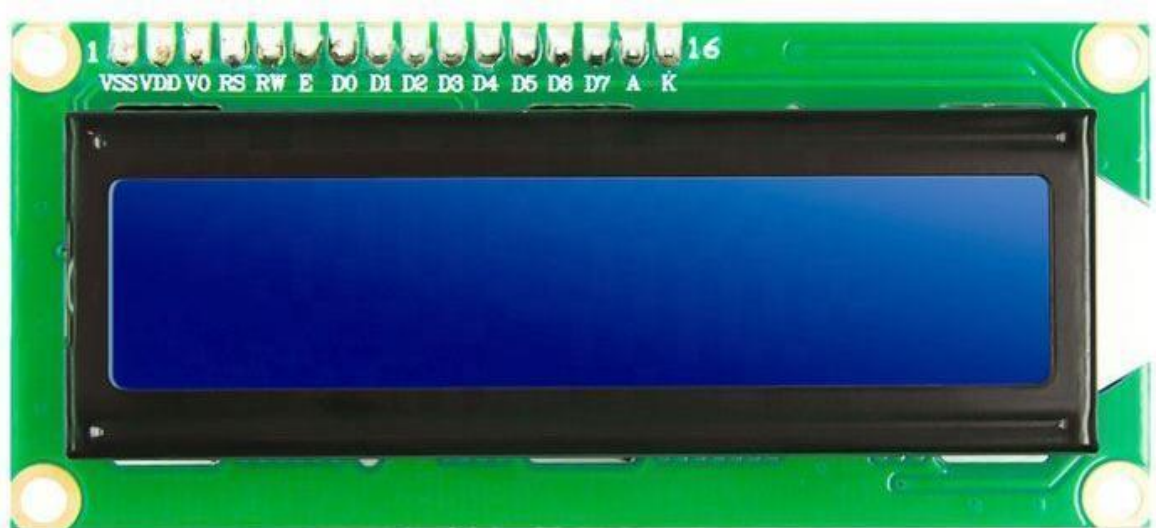
Obrázek č. 8: ponorné čerpadlo

4.1.4. LCD displej 1602A

Displej 1602A má dva řádky a na každém 16 znaků. Displej může zobrazit všechny znaky z ASCII tabulky nebo lze manuálně vytvořit vlastní znaky pomocí 8 krát 8 bitového pole. Displej má 16 pinů k propojení.

- 1) VSS – zemnicí pin
- 2) VCC – napájecí pin, pracovní napětí je 5 V.
- 3) VO – určuje kontrast displeje pomocí měření napětí na pinu,
rozsah měření je 0 – 5 V, kde 0 V je nejmenší kontrast
- 4) RS – výběr registru určuje, jestli displej přijímá data nebo příkazy
0 V – příjem příkazů, 5 V - příjem dat
- 5) R/W – výběr čtení z displeje nebo zápisu na něj, 0 V – zápis, 5 V - čtení
- 6) E – zapínací pin, při 0 V je displej vypnutý, při 5 V je zapnutý
- 7 – 14) – datové piny
- 15,16) – anoda a katoda, používají se k ovládání podsvícení displeje

Zdroje:(19)



Obrázek č. 9 : LCD displej

4.1.5. Teplotní senzor LM35

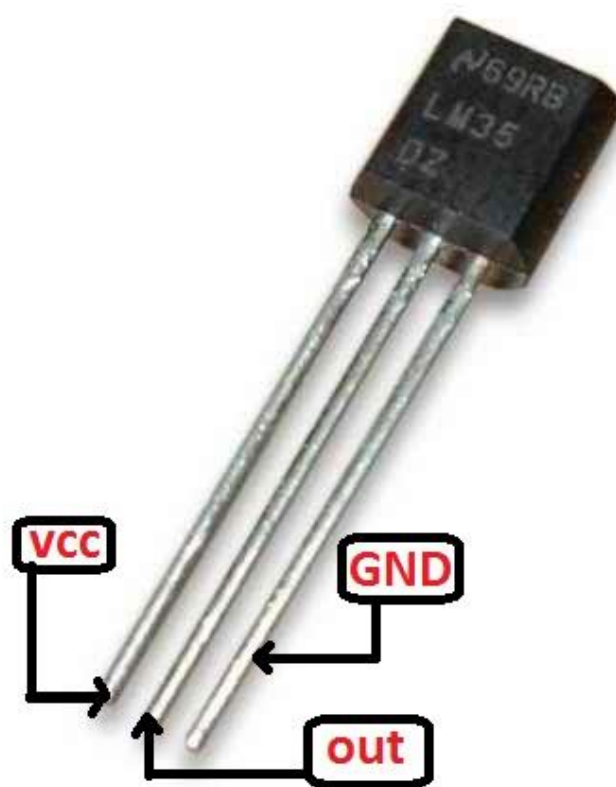
Jednoduchý a levný teplotní senzor. Pracovní napětí je -2 až 35 V a dokáže měřit teploty od -55°C do 150°C s průměrnou chybou 0.5°C. LM35 má 3 piny. Měřenou hodnotu teploty vrací v podobě napětí.

- 1) VCC – pin pro vstupní napětí
- 2) Analog out – tento pin poskytuje měřenou teplotu ve formě napětí.

Poměr je 10mV / °C

- 3) GND – zemnicí pin

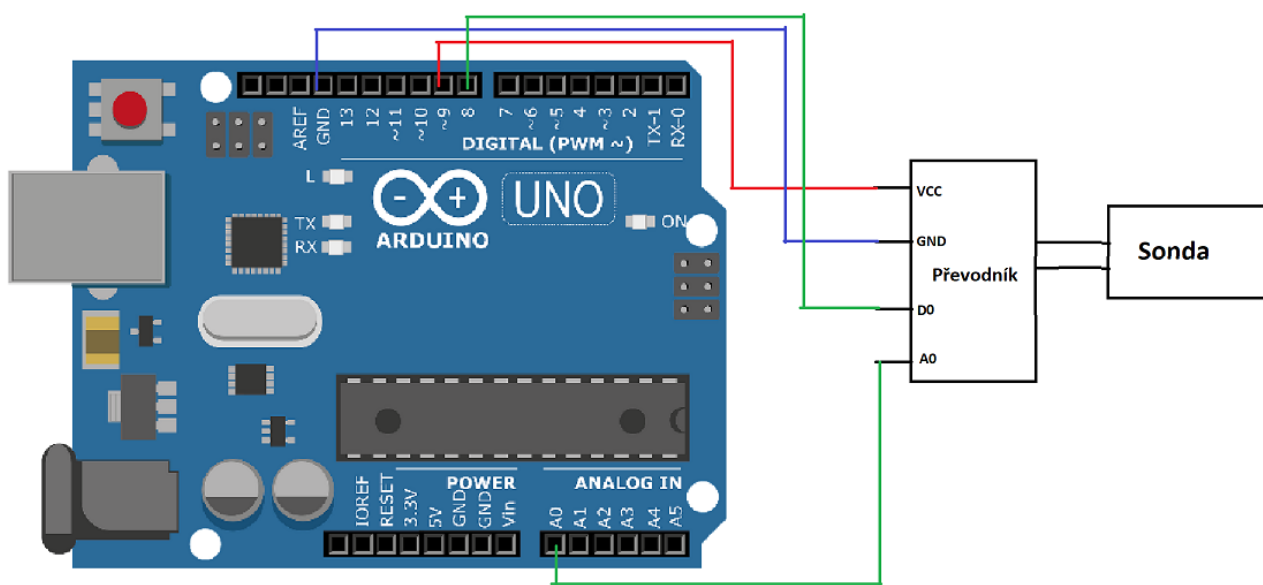
Zdroje:(21)



Obrázek č. 10 : LM35 teploměr

4.2. Zapojení a testování půdního vlhkoměru

Prvním krokem při stavbě systému bylo zapojení a otestování funkce půdního vlhkoměru. Vlhkoměr byl zapojen dle obrázku č. 11. Pin VCC převodníku byl připojen na digitální pin 9 na arduinu. Díky tomu můžeme ovládat pracovní napětí vlhkoměru a ovládat tím, zda je vlhkoměr zapnutý či vypnutý. GND převodníku pin byl připojen na GND pin Arduina. Pin pro digitální výstup převodníku byl připojen na digitální pin 8 arduina. Analogový výstup byl připojen na analogový pin A0 arduina.



Obrázek č. 11 : Schéma zapojení vlhkoměru

Dále byl vlhkoměr otestován pomocí jednoduchého programu (obrázek č. 12). Program začne definováním názvu pinů a vytvořením proměnné pro ukládání času.

Ve funkci `void setup()` započneme komunikaci po sériové lince a nastavíme analogový pin a digitální pin jako vstup a napěťový pin jako výstup. Dále zapíšeme digitální nulu na napěťový pin převodníku a tím ho efektivně předběžně vypneme.

Ve funkci `void loop()` máme podmínku pro měření času. Tato podmínka se spustí, pokud od posledního cyklu uběhly více jak 3 sekundy. Nejprve se zapíše digitální jednička na napěťový pin, tím se zapne převodník a počká se 100 milisekund pro vyrovnání napětí.

Poté se vytvoří proměnné pro uložení analogových a digitálních hodnot z vlhkoměru a provede se čtení a uložení hodnot z datových pinů do nich.

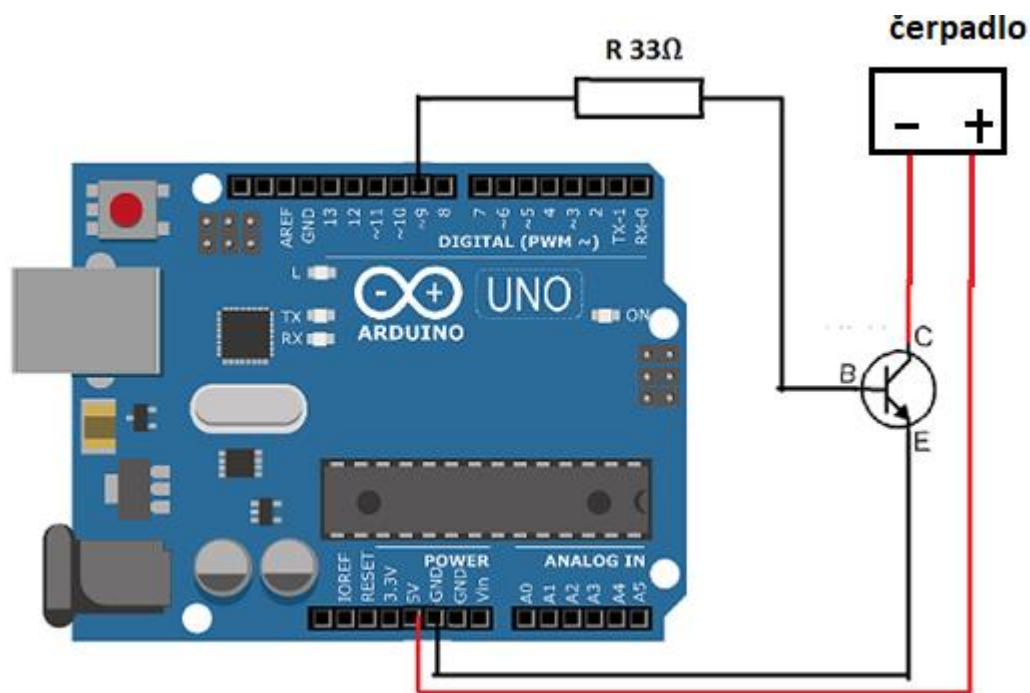
Následně se tyto hodnoty pošlou přes sériovou linku do Arduino IDE, kde si je lze zobrazit v sériovém monitoru. Poté se zapíše digitální nula na napěťový pin, aktuální čas se uloží do proměnné času pro porovnání v podmínce při dalším cyklu.

```
#define analogPin A0
#define digitalPin 8
#define vccPin 9
unsigned long cas = 0;
void setup() {
  Serial.begin(9600);
  pinMode(analogPin, INPUT);
  pinMode(digitalPin, INPUT);
  pinMode(vccPin, OUTPUT);
  digitalWrite(vccPin, LOW);
}
void loop() {
  if (millis() - cas > 3000) {
    digitalWrite(vccPin, HIGH);
    delay(100);
    int analog = analogRead(analogPin);
    bool digital = digitalRead(digitalPin);
    Serial.print("Analog ");
    Serial.print(analog);
    Serial.print("Digital ");
    Serial.print(digital);
    Serial.println();
    digitalWrite(vccPin, LOW);
    cas = millis();
  }
}
```

Obrázek č. 12 : Program pro test vlhkoměru

4.3. Zapojení a otestování ponorného čerpadla

V této části byla ověřena funkce ponorného čerpadla. Čerpadlo bylo zapojeno dle obrázku č. 13. Pro spínání čerpadla byl použit NPN Tranzistor SS8050 a bázový odpor 33Ω . Báze tranzistoru byla připojena přes bázový odpor na digitální pin 9 na desce. Emitor tranzistoru byl připojen na GND pin na desce a kolektor na nulový pin čerpadla. Napěťový pin čerpadla byl připojen na 5 V pin na desce.



Obrázek č. 13 : Schéma zapojení čerpadla

Pro otestování čerpadla byl použit následující program (obrázek č. 14). Nejdříve se pin 3 nadefinuje jako čerpadlo, vytvoří se boolean proměnná, a nastaví se jako false.

Ve funkci *void setup()* se nastaví pin čerpadla jako výstup a zapíše na něj digitální nula tedy 0 V.

Ve funkci *void loop()* se nachází podmínka, která vyhodnocuje hodnotu proměnné *zalito*. Pokud je proměnná false tak se spustí cyklus, který bude po dobu 3000 milisekund zasílat na pin čerpadla digitální jedničku, tedy 5 V. Tímto se čerpadlo uvede do provozu. Po dokončení cyklu se proměnná *zalito* nastaví na true. Díky tomu se podmínka po druhé nespustí a zamezí tím opakovanému spínání čerpadla.

```

#define cernpadlo 3
boolean zalito = false;
void setup() {
    pinMode(cernpadlo, OUTPUT);
    digitalWrite(cernpadlo, LOW);
}

void loop() {

if(zalito == false)
{
    for(int a = 0; a < 3000; a += 1)
    {
        digitalWrite(cernpadlo, HIGH);

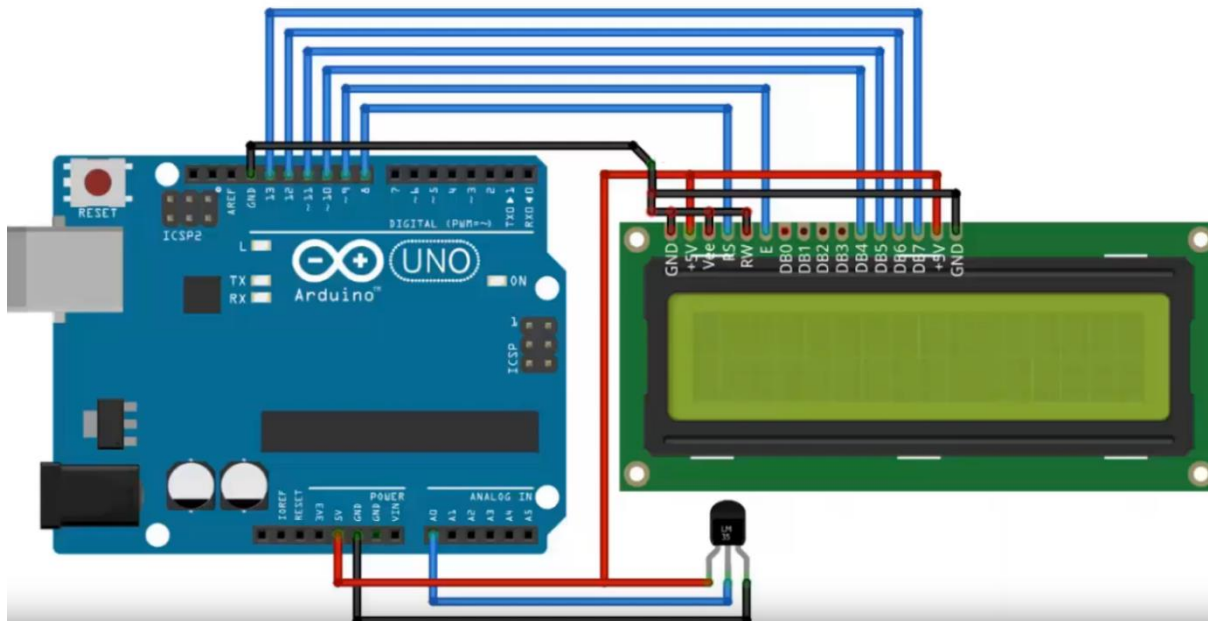
    }
    zalito = true;
}
}

```

Obrázek č. 14 : Program pro test čerpadla

4.4. Zapojení a testování teploměru a LCD displeje

V této části byla ověřena funkce teploměru LM35 spolu s LCD displejem. Teploměr a displej byly zapojeny dle obrázku č. 15. GND pin teploměru byl připojen k GND pinu na desce a VCC pin k napěťovému pinu 5 V. Analogový výstupní pin teploměru byl připojen na pin A0. Pin VCC displeje byl připojen na napětí 5 V. Piny GND, V0 a R/W displeje byly připojeny k GND pinu desky. Pin GND efektivně znamená nulové napětí. Displej se tedy přepne do módu zapisování na něj a kontrast se zvedne na maximum. RS pin byl připojen a pin 8 a E pin displeje by připojen na pin 9 na desce. Datové piny displeje D0 - D3 nebyly využity. Datové piny D4 – D7 byly připojeny na piny 10 – 13. Piny A a K byly připojeny na stálé napětí 5 V a GND pro nejvyšší úroveň podsvícení.



Obrázek č. 15: Schéma zapojení teploměru a LCD displeje

Pro otestování byl využit následující program (obrázky č. 16 , 17 a 18). Nejprve se do programu přidá knihovna *LiquidCrystal.h*, která obsahuje příkazy pro ovládání LCD displeje. Následně se vytvoří ovládací proměnná *lcd* pro displej a přidají se do ní piny 8 – 13. Piny v závorkách proměnné *LiquidCrystal* korespondují s piny na LCD displeji a jsou seřazeny následovně (RS, R/W, E, D4, D5, D6, D7). Poté se definuje pin A0 pro teploměr. Poté se definuje 8 krát 8 bitové pole, které bude použito pro vytvoření vlastního znaku.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);
#define analog A0
byte degree [8] =
{
    0b00011,
    0b00011,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000
};
```

Obrázek č. 16 : Počáteční část programu pro test teploměru a displeje

Ve funkci `void setup()` se inicializuje LCD displej a vytvoří se speciální znak pomocí již zmíněného pole. Následně se nastaví analogový pin jako vstup. Poté se na první řádek displeje vypíše „LCD“ a na druhý „Teploměr“. Program poté vyčká 2 sekundy a vymaže displej.

```
void setup() {
  lcd.begin(16,2);
  lcd.createChar(1, degree);
  pinMode(analog, INPUT);
  lcd.setCursor(0,0);
  lcd.print("LCD");
  lcd.setCursor(0,1);
  lcd.print("Teploměr");
  delay(2000);
  lcd.clear();
}
```

Obrázek č. 17 : Funkce `void setup()` programu pro test teploměru a displeje

Ve funkci `void loop()` se vytvoří proměnná pro data z teploměru a rovnou se do něj načtou. Tato hodnota je poté vynásobena konstantou 0,48828125 pro převod hodnoty napětí, kterou poskytuje teploměr, na hodnotu ve stupních Celsia. Následně se na první řádek LCD displeje vypíše „Teplota“ a na druhý hodnota proměnné `temp` spolu se speciálním znakem, který má představovat znak stupně. Program poté vyčká 2 sekundy a celý proces se opakuje.

```
void loop() {
  float temp = analogRead(tmp);
  temp = temp * 0.48828125;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Teplota");
  lcd.setCursor(0,1);
  lcd.print(temp);
  lcd.write(1);
  delay(2000);
}
```

Obrázek č. 18 : Funkce `void loop()` programu pro test teploměru a displeje

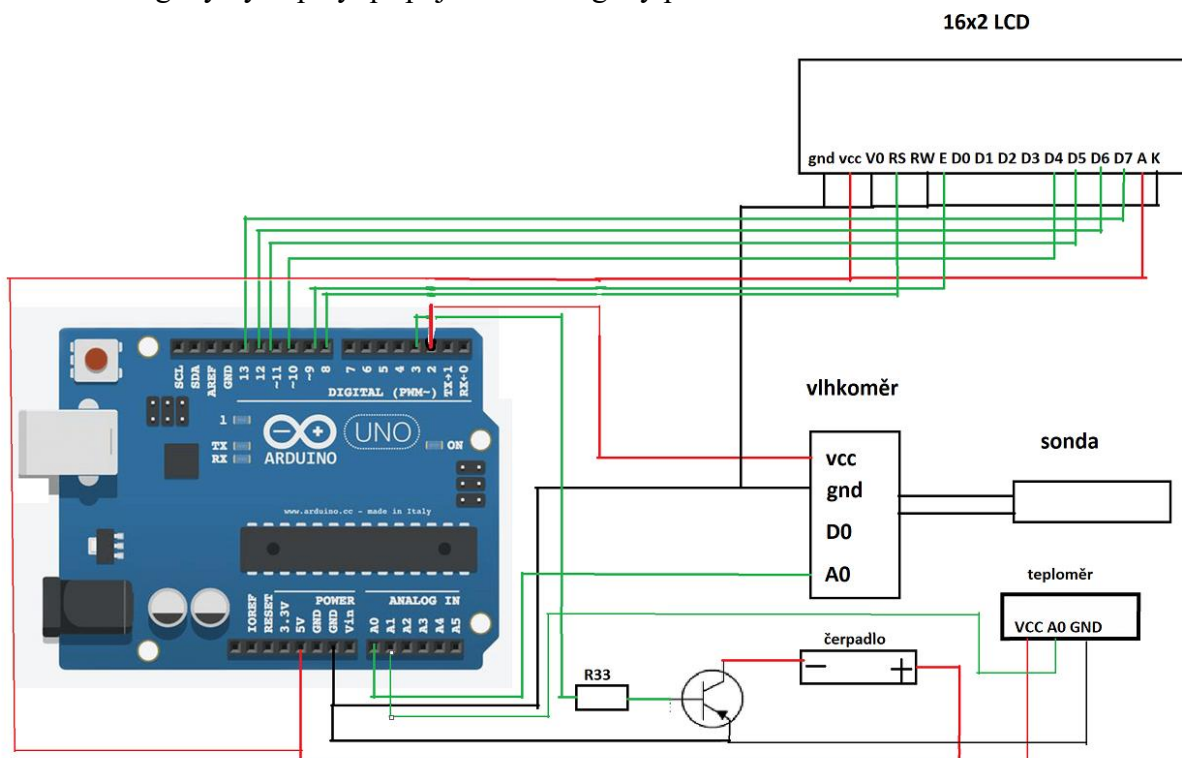
4.5. Konečné zapojení systému

U LCD displeje byly piny GND, V0 a R/W připojeny k GND pinu desky. Displej se díky tomu přepne do módu zapisování na něj a kontrast se zvedne na maximum. Pin VCC displeje byl připojen na napětí 5 V. RS pin byl připojen a pin 8 a E pin by připojen na pin 9 na desce. Datové piny displeje D0 - D3 nebyly využity. Datové piny D4 – D7 byly připojeny na piny 10 – 13. Piny A a K byly připojeny na stálé napětí 5 V a GND pro nejvyšší úroveň podsvícení.

Teploměr byl připojen následovně. GND pin k GND pinu na desce, VCC pin k napěťovému pinu 5 V a analogový výstupní pin teploměru byl připojen na pin A1.

Při zapojení čerpadla byl použit NPN Tranzistor S8050 a bázový odpor 33Ω. Báze tranzistoru byla připojena přes bázový odpor na digitální pin 3 na desce. Emitor tranzistoru byl připojen na GND pin na desce a kolektor na nulový pin čerpadla. Napěťový pin čerpadla byl připojen na 5 V pin na desce.

Vlhkoměr byl zapojen takto. Pin VCC vlhkoměru byl připojen na digitální pin 2 na arduinu. Díky tomu můžeme ovládat pracovní napětí vlhkoměru a ovládat tím, zda je vlhkoměr zapnutý či vypnutý. GND převodníku pin byl připojen na GND pin Arduina. Pin pro digitální výstup nebyl připojen, protože ho v řídicím programu nebudeme využívat. Analogový výstup byl připojen na analogový pin A0 arduina.



Obrázek č. 19 : Schéma konečného zapojení



Obrázek č. 20 : Obrázek hotového systému

4.6. Řídící program systému

Řídící program funguje následně. Nejdříve se importuje knihovna *LiquidCrystal* pro ovládání LCD displeje. Po té se vytvoří proměnná *lcd* a přiřadí se jí datové piny 8 – 13. Následně se definují ostatní piny a vytvoří se 8 krát 8 bitové pole pro vytvoření znaku stupně. Nakonec se vytvoří proměnné pro ukládání času a měřených hodnot z vlhkoměru a teploměru.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,10,11,12,13);
#define analog A0
#define temp A1
#define vcc 2
#define cernadlo 3
byte degree [8] =
{
    0b00011,
    0b00011,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000,
    0b00000
};
long cas = 0;
float tmp = 0;
float mos = 0;
```

Obrázek č. 21 : Počáteční část konečného programu

Ve funkci *void setup()* se inicializuje LCD displej a vytvoří se speciální znak stupně. Poté se nastaví piny analog a temp jako vstupy a piny vcc a čerpadlo jako výstupy. Nakonec se na piny vcc a čerpadlo pošle napětí 0 V.

```
void setup() {  
  lcd.begin(16,2);  
  lcd.createChar(1, degree);  
  pinMode(analog, INPUT);  
  pinMode(temp, INPUT);  
  pinMode(vcc, OUTPUT);  
  pinMode(čerpadlo, OUTPUT);  
  digitalWrite(vcc, LOW);  
  digitalWrite(čerpadlo, LOW);  
}
```

Obrázek č. 22 :Funkce *void setup()* konečného programu

Ve funkci *void loop()* se nejprve vyčistí LCD displej a vypíšu se na něj hodnoty z proměnných pro vlhkost a teplotu. Na první řádek hodnota odporu vlhkoměru a na druhý řádek teplota spolu se znakem stupně. Tyto hodnoty se vypisují při každém cyklu. Poté se vyhodnotí podmínka pro kontrolu času. Na rozdíl od testovacího programu, kde byl časový limit 3 sekundy, byl zde časový limit zvýšen na 1 minutu, protože při reálném použití není potřeba vlhkost kontrolovat tak často. Pokud uběhla více jak minuta, tak se na pin vcc pustí napětí 5 V, a tím se zapne vlhkoměr, a počká se 100 milisekund pro vyrovnaní napětí.

Poté se načtou měřené hodnoty vlhkoměru a teploměru do připravených proměnných. Vlhkoměr vrací hodnoty odporu mezi 0 a 1023. Teploměr vrací napětí úměrné měřené teplotě. Následně se hodnota z teploměru vynásobí konstantou pro převedení do stupňů Celsia.

Poté se vyhodnotí podmínka pro spuštění čerpadla. Pokud je hodnota vyšší než 800, znamená to, že v květináči je téměř sucho. Tato podmínka spustí cyklus, který po dobu pěti sekund posílá napětí 5 V na pin čerpadla, a tím ho spustí. Po dokončení cyklu program vyše nulové napětí na modul vlhkoměru a čerpadla, a tím je vypne. Nakonec program uloží aktuální čas do proměnné pro porovnání při dalším cyklu.

```

void loop() {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(mos);
  lcd.setCursor(0,1);
  lcd.print(tmp);
  lcd.write(1);
  if (millis() - cas > 60000) {
    digitalWrite(vcc, HIGH);
    delay(100);
    mos = analogRead(analog);
    tmp = analogRead(temp);
    tmp = tmp * 0.48828125;
    if(analog > 800){
      for(int a = 0;a < 5000;a +=1){
        digitalWrite(cerpadlo, HIGH);
      }
    }
    digitalWrite(cerpadlo, LOW);
    digitalWrite(vcc, LOW);
    cas = millis();
  }
}

```

Obrázek č. 23 :Funkce void loop() konečného programu

5. Zhodnocení

System navržený v této práci je určen pro lokalizované zavlažování jediného květináče. Je tedy vhodný pro domácí použití, například pro zavlažování jediné květiny, která je citlivá na nedostatek vody. System má také funkční teploměr, který, může být v domácím prostředí užitečný.

Jednou z výhod tohoto systému je jeho cena a dostupnost. Cena všech použitých komponent nepřesáhla 1000 korun českých. Mikrořadiče jsou v dnešní době velice dostupné a lze je zakoupit v mnoha obchodech s elektronikou. Existuje také několik obchodů, které se specializují na mikrořadiče a vybavení pro ně. Ovšem i přes výhody, že si musí člověk tento system sám postavit a naprogramovat, to není pro každého.

Další z výhod je nezávislost napájení systému. Celý system byl navržen tak, aby mohl být napájen pouze ze samotného Arduina. Tato schopnost při stavbě a testování nebyla využita, ale deska Arduino UNO obsahuje konektor pro napájení z externího zdroje a existují adaptéry, které pro napájení využívají dvojici AA baterek. Díky tomu můžeme system v domácím prostředí použít téměř kdekoliv.

Jedním z věkých nedostatků systému je absence kontroly stavu vody ve vodním zdroji. Tedy pokud dojde voda, system nemá možnost, jak tento nedostatek detekovat, a bude běžet naprázdno. Díky tomu se čerpadlo velice rychle poškodí. Toto lze vyřešit přidáním senzoru vodní hladiny do systému. Poté by byla nutná modifikace programu, aby se čerpadlo nespouštělo pokud je ve zdroji nedostatek vody.

Dalším problémem může být postupné opotřebení sondy u senzoru vlhkosti. Při studiu technických parametrů sondy jsem zaznamenal, že výrobce upozorňuje na možnost degradace sondy vlhkoměru při dlouhodobém používání. Toto nebyl problém při testování systému, ale je možné, že při dlouhodobém využívání se může sonda poškodit. Na druhou stranu sondy vlhkoměru jsou dostupné k zakoupení samostatně za velice nízkou cenu, v době psaní této práce 13 korun českých za kus.

6. Závěr

Cílem této bakalářské práce bylo sestavit automatizovaný zavlažovací systém s využitím platformy Arduino. Pro realizaci tohoto projektu byla vybrána platforma Arduino UNO, která je zdaleka nejrozšířenější s platforem Arduino.

V teoretické části jsme se nejdříve seznámili s různými typy konstrukcí automatizovaných zavlažovacích systémů. Poté následoval základní popis platformy Arduina, a poté detailní popis platformy Arduino UNO. Před psaním této práce jsem měl pouze minimální zkušenosti s mikrořadiči a žádné zkušenosti s platformou Arduino. Psaní popisu desky Arduino UNO mi velice pomohlo se seznámit s jejími vlastnostmi a schopnostmi. Díky velké popularitě platformy Arduino je mnoho informací o ní dostupných na internetu. Na internetu jsem našel velké množství již hotových ukázkových projektů, návodů a popisů použití různých periférií. Tyto informace mi pomohly při pochopení funkcí zvolených komponent a způsobu jejich využití.

V praktické části došlo k postupnému sestavení celého systému. Nejprve došlo k zapojení a otestování funkce jednotlivých komponent s popisem programového vybavení pro jejich testování. Po úspěšném ověření funkčnosti všech částí systému byly tyto části spojeny do finální verze systému spolu s naprogramováním finální verze ovládacího programu a popisem jeho funkce.

V průběhu realizace systému jsem se setkal s nečekaným problémem. Při nahrávání programů z IDE na Arduino jsem opakovaně dostával chybu "avrduide: ser_open(): can't set com-state for "\\.\COM03". Po vyhledání problému na internetu se ukázalo, že se jedná o chybu při sériové komunikaci. Jedním z prvních důvodů je záměna originálního USB/Serial převodníku typu atmega16u2 na Arduino za jiný levnější typ, nejčastěji za převodník ch340. Po instalaci ovladačů pro ch340 se výskyt problému snížil, ale kompletně nezmizel. Tento problém neměl velký dopad na průběh práce. Při každém výskytu problému stačilo pokusit se opakovaně nahrát program na Arduino nebo přepojit USB kabel do jiného portu.

Cílem této práce bylo sestavit automatizovaný zavlažovací systém pro květináč a tohoto cíle bylo dosaženo. Ovšem při stavbě systému jsem si uvědomil, že má několik nedostatků jako například nepřítomnost funkce pro kontrolu vody ve zdroji. Tyto nedostatky by bylo možné jednoduše vyřešit přidáním potřebných periférií do systému a úpravou programového vybavení.

7. Zdroje

- 1) ARDUINO.CC [online]. [cit. 2021-01-15]. Dostupné z: <https://www.arduino.cc/>
- 2) Automatické zavlažování zahrady. *Chatař chalupář* [online]. [cit. 2021-01-15]. Dostupné z: <https://www.chatar-chalupar.cz/automaticke-zavlazovani-zahrady/>
- 3) Obecná pravidla návrhu, 2021. *Hunter závhahové systémy* [online]. [cit. 2021-01-19]. Dostupné z: https://zavlahy.irimon.cz/clanek_obecna_pravidla_navrhu_automaticke_zavlahy
- 4) SÝKORA, Radek. Úsporné zavlažování skleníku, 2012. *Recepty prima nápadů* [online]. [cit. 2021-01-20]. Dostupné z: <https://www.receptyprimanapadu.cz/zahrada/usporne-zavlazovani-skleniku>
- 5) Podpovrchová kapkovácí závlaha AS-GEOFLOW, ©2021. *Voda v domě* [online]. [cit. 2021-01-23]. Dostupné z: <https://www.vodavdome.cz/podpovrchova-kapkovaci-zavlahas-geoflow/>
- 6) LONG, Moe. Best Arduino Clones - Are Arduino Clones Any Good?, 2019. *electromaker.io* [online]. [cit. 2021-01-26]. Dostupné z: <https://www.electromaker.io/blog/article/best-arduino-clones>
- 7) Arduino For Beginners, ©2021. *Maker spaces* [online]. [cit. 2021-01-26]. Dostupné z: <https://www.makerspaces.com/arduino-uno-tutorial-beginners/>
- 8) BADAMI, Vasudhenra, A tour of the Arduino UNO board, 2016. *Hacker earth* [online]. [cit. 2021-01-28]. Dostupné z: <https://www.hackerearth.com/blog/developers/a-tour-of-the-arduino-uno-board/>
- 9) THORTON, Scott, RISC vs. CISC Architectures: Which one is better?, 2018. *Microcontroller tips* [online]. [cit. 2021-01-28]. Dostupné z: <https://www.microcontrollertips.com/risc-vs-cisc-architectures-one-better/>
- 10) RUBENZ, Paul, Types of Computer Memory, 2019. *Enterprise storage forum* [online]. [cit. 2021-01-29]. Dostupné z: <https://www.enterprisestorageforum.com/hardware/types-of-computer-memory/>
- 11) KRANZ, Garry & Carol SLIWA Flash memory, 2019. *Tech Target* [online]. [cit. 2021-01-29]. Dostupné z: <https://searchstorage.techtarget.com/definition/flash-memory>
- 12) ZAIT, Anat, An Introduction to Arduino Uno Pinout, 2019. *Circuit.io* [online]. [cit. 2021-02-01]. Dostupné z: <https://www.circuito.io/blog/arduino-uno-pinout/>

- 13) Arduino Uno Pinout – Diagram [obrázek] [online]. [cit. 2021-02-01]. Dostupné z:
https://images.prismic.io/circuito/8e3a980f0f964cc539b4cbbba2654bb660db6f52_arduino-uno-pinout-diagram.png?auto=compress,format
- 14) Everything You Need to Know About Arduino Code, 2019. *Circuit.io* [online]. [cit. 2021-02-03]. Dostupné z: <https://www.circuito.io/blog/arduino-code/>
- 15) M., Luboš, Půdní vlhkoměr, ©2021. *Návody Drátek* [online]. [cit. 2021-02-03]. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/pudni-vlhkomer.html>
- 16) Půdní vlhkoměr. *Návody Drátek* [obrázek] [online]. [cit. 2021-02-03]. Dostupné z: https://navody.dratek.cz/images/obr_clanky/120_pudni_vlhkomer/120_pudni_vlhkomer.jpg
- 17) M., Luboš, Mini DC čerpadlo, ©2021. *Návody Drátek* [online]. [cit. 2021-02-04]. Dostupné z: <https://navody.dratek.cz/navody-k-produktum/mini-dc-cerpadlo.html>
- 18) Mini DC čerpadlo. *Návody Drátek* [obrázek] [online]. [cit. 2021-02-04]. Dostupné z: https://navody.dratek.cz/images/obr_clanky/164_dc_cerpadlo/164_cerpadlo_dc.jpg
- 19) Interfacing 16×2 Character LCD Module with Arduino, © 2021. *Last minute engineers* [online]. [cit. 2021-02-06]. Dostupné z: <https://lastminuteengineers.com/arduino-1602-character-lcd-tutorial/>
- 20) Displej LCD1602A I2C, 16x2 znaků, modré podsvícení. *Hadex* [obrázek] [online]. [cit. 2021-02-06]. Dostupné z: <https://www.hadex.cz/img/zbozi/m505b.jpg>
- 21) LM35 Temperature Sensor, 2017. *Components 101* [online]. [cit. 2021-02-06]. Dostupné z: <https://components101.com/sensors/lm35-temperature-sensor>
- 22) LM35 Temperature Sensor [obrázek] [online]. [cit. 2021-02-06]. Dostupné z: http://4.bp.blogspot.com/-aBqzmENeE-4/VaPWza5ZPrI/AAAAAAAAANw/fXsVVuwjM_A/s1600/LM35.jpg
- 23) Arduino UNO board, *Hacker earth* [obrázek] [online]. [cit. 2021-02-08]. Dostupné z: <https://blog-c7ff.kxcdn.com/blog/wp-content/uploads/2017/02/Blog-Featured-Image.png>
- 24) Blink [obrázek] [online]. [cit. 2021-02-03]. Dostupné z: <https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>
- 25) Atmel "AVR" Architecture - Part I, ©2021. *TechiCodes* [online]. [cit. 2021-02-09]. Dostupné z: <https://technicodes.weebly.com/avrarch.html>

