

Střední odborná škola strojní a elektrotechnická Velešín

Informační technologie



MATURITNÍ PRÁCE

Návod na chytrou zahradu za pomoci jednodeskového
počítače Arduino

Tomáš Zajíček

NÁZEV PRÁCE: Návod na chytrou zahradu za pomoci jednodeskového počítače Arduino

JMÉNO AUTORA: Tomáš Zajíček

STŘEDNÍ ODBORNÁ ŠKOLA STROJNÍ A ELEKTROTECHNICKÁ VELEŠÍN

STUDIJNÍ OBOR: 18-20-M/01 - Informační technologie

ŠKOLNÍ ROK: 2021/2022

VEDOUCÍ PRÁCE: Jiří Bendík

KONTAKT NA AUTORA PRÁCE: tomas.zajicek02@gmail.com

ANOTACE:

Cílem této práce je poukázat na využití jednodeskového počítače v běžném životě. Tato práce si dává za úkol ukázat postup od naprogramování a spojování jednotlivých modulů, až po konečný produkt. Produktem v tomto případě se stává jednoduchá chytrá zahrada, která bude kontrolovat stav rostliny a zalévat ji.

Tímto bych moc rád poděkoval mému vedoucímu práce, který k mé velké radosti byl Jiří Bendík. Děkuji mu za všechny rady a jeho čas při pročítání této práce. A další důležitá osoba, bez které by tato práce nebyla úplná, je Ing. Vít Podroužek, který mi ochotně vytiskl krabičku pro Arduino. Ještě bych rád poděkoval své rodině za podporu a trpělivost. Děkuji vám moc za pomoc a za podílení se na této práci se mnou.

Prohlašuji, že jsem tuto práci zpracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Datum

Podpis

Obsah

1	Úvod	8
2	Seznámení s jednodeskovými počítači	9
3	Arduino	10
3.1	Vývojové prostředí	10
4	Co bude zahrada umět	12
5	Co bude potřeba pro sestavení zahrady	13
5.1	Arduino	13
5.2	Půdní vlhkoměr (LM393 – Soil hygrometer)	13
5.3	Modul relé	15
5.4	Bluetooth modul (HC-06 RS232)	15
5.5	OLED displej (I2C OLED)	16
5.6	Ostatní	16
6	Zapojení nanečisto	17
6.1	Zapojení pro zjištění vlhkosti	17
6.2	Zapojení s čerpadlem	18
6.3	Přidání Bluetooth	19
6.4	Grafický výstup	20
7	Programovací část	22
7.1	Programování Arduina	22
7.1.1	Seřizování hodnot	22
7.1.2	Přidání relé spínače a čerpadla	24
7.1.3	Zprovoznění Bluetooth	25
7.1.4	Programování grafického výstupu	26
7.2	Vytvoření jednoduché aplikace na ovládání pro Android	28
7.2.1	Aplikace Arduino BlueControl	28
7.2.2	Vytvoření vlastní aplikace v MIT AppInventor	29

8	Návrh a 3D tisk krabicky pro Arduino	34
8.1	Modelování v Tinkercadu	34
8.2	3D tisk	35
9	Konečné zapojení	37
10	Finální produkt.....	38
11	Závěr	39
	Seznam použité literatury.....	40
	Seznam použitých obrázků	41
	Seznam příloh	42

1 Úvod

Toto téma jsem si vybral, protože používání Arduina mi přijde velmi zajímavé i užitečné. Cílem této práce je ukázat a vysvětlit, jak Arduino funguje a hlavní program tvoří sestavení vlastního projektu, který je v tomto případě chytrá zahrada. Tento projekt není ani příliš jednoduchý ani těžký, proto je podle mě ideální pro čtenáře, kteří s Arduinem mají nějaké zkušenosti, ale nemyslím si, že se v něm budou úplní začátečníci ztrácet. Doufám v to, že touto prací navnadím ty čtenáře, kteří ještě neměli možnost si vyzkoušet zhotovit podobný projekt, aby se také začali zajímat o Arduino a vytvářeli své vlastní kreativní projekty.

Na začátku práce se zaměřuji spíše na teoretickou část. Praktická část přichází po seznámení s Arduinem a vysvětlení modulů, které budou potřeba.

Teoretická část seznámí čtenáře hlavně se základy jednodeskových počítačů a Arduinem jako takovým.

Praktická část bude obsahovat programování Arduina a aplikace pro ovládání Arduina. Zahrnuto bude i modelování a 3D tisk krabičky pro Arduino a závěr praktické části bude tvořit finální zapojení a zprovoznění chytré zahrady.

V závěru se budu vyjadřovat k mé spokojenosti s finální formou projektu a dalším poznatkům.

2 Seznámení s jednodeskovými počítači

Jednodeskový počítač je typ počítače, který má všechny své komponenty (RAM, CPU, GPU, ...) na jedné desce. Tímto sice přicházíte o možnost pozdějších hardwarových úprav, jako je možné například u stolních počítačů, ale na druhou stranu tyto počítače bývají levné, proto koupení nové generace není velká investice.^[1] Jednodeskové počítače navíc zpravidla mívají vstupní a výstupní konektory. Mezi základní konektory patří vstup pro nahrávání softwaru, připojení k internetu, klasické USB, ale ani grafický výstup není dnes žádnou výjimkou.

Jednodeskové počítače nejsou většinou určeny pro stejné účely jako stolní počítače, ačkoliv výjimky jako například Raspberry Pi 4 slibuje skoro plnohodnotný zážitek jako u stolního počítače. A každým rokem jsou o něco blíže k dosažení stejných výsledků jako klasické stolní počítače.

Většinou tyto počítače najdeme v továrnách, kde mohou třeba ovládat ventily, teplotu nebo hlídat, aby nikdo nepřekročil čáru, ale díky jejich ceně je můžeme najít i u nadšenců doma.

Jednodeskové počítače jsou si dost podobné, ale přeci se od sebe liší. A to hlavně v tom, na co je budete chtít použít. Budete potřebovat grafický výstup? Chcete připojení k internetu za pomoci WIFI? Kolik RAM budete potřebovat? Díky těmto otázkám můžete dojít k zařízení, které bude splňovat všechna vaše kritéria.

3 Arduino

„Arduino je otevřená elektronická platforma, založená na jednoduché počítačové desce (hardware) a vývojovém prostředí, které slouží k tvorbě software.“^[2]
Původně bylo zamýšleno jako učební nástroj pro studenty^[3], ale postupem času se stalo populární i v profesním světě.

„Hlavními vývojáři Arduina jsou Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis a Nicholas Zambetti.“^[3]

Pro tento projekt jsem si zvolil používat Arduino klon. Arduino klon je jednodeskový počítač, který je sestaven podle Arduina, akorát ho produkuje jiná firma. Těchto klonů je několik. Pro svůj projekt jsem zvolil Lafvin UNO a Arduino Nano. Přičemž Lafvin UNO bude kvůli své velikosti sloužit jako testovací deska.

3.1 Vývojové prostředí

Arduino IDE je vývojové prostředí potřebné k nahrávání kódu do Arduina. Programování je založeno na jazyce Wiring a je uživatelsky přívětivé. Software je zdarma ke stažení na Windows, Mac i Linux (možné je také použít webové rozhraní) a obsahuje knihovny pro všechny desky Arduino, ale i většinu klonů.^[2] Kromě knihoven pro samotné desky, můžeme v Arduinu IDE najít také oficiální a komunitní knihovny pro většinu známých modulů.



Obrázek 1: Vývojové prostředí Arduino IDE

Základ textového pole tvoří ***void setup(){}*** , do kterého píšeme tu část kódu, která proběhne pouze jednou a ***void loop(){}*** , ve kterém je kód, který se neustále opakuje. Nad ***void setup(){}*** se zpravidla zapisují proměnné. Tyto dvě metody se automaticky vytvoří s novým projektem. Příkazy se jako ve většině programovacích jazycích ukončují středníkem.

V pravé části okna můžeme vidět ikonu lupy, která znázorňuje ***Serial monitor***, který používáme při výpisu hodnot z Arduina.

Pro náhrání projektu do Arduina použijeme obrázek šipky, který se nachází v levém horním rohu. Tato akce přenesení kód do paměti Arduina pomocí USB kabelu a Arduino poté může vykonávat kód i bez připojení k počítači.

V Arduino IDE vždy zkontrolujte v liště na tlačítku ***Nástroje*** v sekci ***Vývojová deska***, že máte zvolený správný typ Arduina (v mém případě Arduino UNO) a v sekci ***Port*** správné číslo portu vašeho připojení k Arduinu (toto nastavení se u každého liší).

4 Co bude zahrada umět

Chytrá zahrada se postará o to, abyste nemuseli myslet na zalévání rostlin. Jednodeskový počítač sám rozpozná, kdy je půda u rostliny vysušená a když bude třeba zalije ji. Dále bude možné zahradu ovládat aplikací, která je připojena přes Bluetooth.

Celý proces bude zpracovávat Arduino, které bude připojeno na půdní vlhkoměr. Ten bude komunikovat s relátkem připojeném na čerpadle. Vzdálenou komunikaci bude zprostředkovávat Bluetooth modul.

5 Co bude potřeba pro sestrojení zahrady

5.1 Arduino

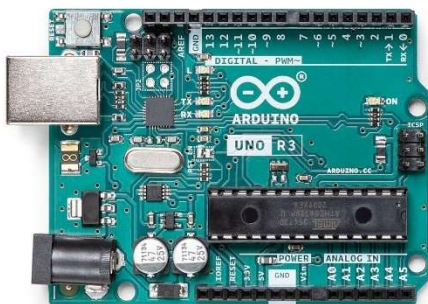
Hlavní komponent samozřejmě tvoří Arduino, já budu pracovat hlavně s klonem Arduina UNO – Lafvin UNO. Prakticky je ale jedno jaké Arduino nebo klon budete používat, hlavní je si dát pozor na napětí a na piny, které budete potřebovat k tomuto projektu.

Na tomto Arduinu se nacházejí 2 napájecí konektory. Konektor USB typu B slouží k napájení a nahrávání kódu. Dále máme konektor typu Jack, který se používá pouze na napájení.

Kromě těchto dvou konektorů máme možnost napájet Arduino přes **Vin** pin, ale musíme si dávat pozor, protože **tento pin nemá žádnou ochranu napětí** a Arduino se může poškodit.

Pořizovací cena originálního Arduina UNO se pohybuje okolo 600 Kč (5. 1. 2022), ale klony můžete sehnat do 200 Kč (5. 1. 2022).

Pozor, většina výstupů Arduina pracuje na napětí 5 V a moduly, které jsou na 3.3 V by se mohly poškodit.



Obrázek 2: Arduino UNO. ([zdroj](#))

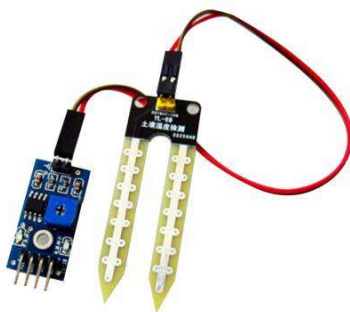
5.2 Půdní vlhkoměr (LM393 – Soil hygrometer)

Tento modul je klíčový pro náš projekt. Je vytvořen přímo na měření vlhkosti v půdě.

Princip jeho fungování spočívá v účinnosti přenosu elektřiny mezi „packami“. Čím více je půda vlhká, tím je přenos elektřiny jednodušší.

Deska, která je připojená k vlhkoměru, slouží jako prostředník při spojení s Arduinem a také jako nastavení citlivosti přenosu na digitální pin. Máme dvě možnosti, jak posílat data přes tuto desku do Arduina. První je analogový, druhý je digitální. Při použití digitálního přenosu je nutné odladit hodnoty vlhkosti pomocí šroubu umístěném na desce. V tomto projektu budeme používat analogový přenos, který posílá data do analogového pinu na Arduinu a pomocí hodnot, které obdržíme, upravujeme citlivost v kódu.

Existuje také vylepšená a o něco dražší varianta, která chrání modul před korozí, ale ta funguje na trochu jiném principu.



Obrázek 3: Půdní vlhkoměr pro jednodeskové počítače. ([zdroj](#))

5.3 Modul relé

Používám relé ve formě modulu, který musíte už jen zapojit. Používám relé na 5 V. Pro sestavu pouze jednoho čerpadla nám stačí jednokanálové relé.

Princip fungování relé bude dále vysvětlen v kapitole Zapojení s čerpadlem.



Obrázek 4: Modul relé. ([zdroj](#))

5.4 Bluetooth modul (HC-06 RS232)

Tento modul bude sloužit jako další funkce ovládání Arduina. Není tedy přímo nutné jej mít, ale je to zajímavé a snadné ovládání.



Obrázek 5: Bluetooth modul. ([zdroj](#))

Modul disponuje vestavěnou anténou, Bluetooth verzí 2.0 a připojuje se pomocí 4 pinů. Dva z toho jsou napájecí (**VCC** a **GND**) a zbytek je na přijímání a posílání dat (**RX** a **TX**).^[5]

5.5 OLED displej (I2C OLED)

Tento modul se bude starat o výpisy dat a stejně jako Bluetooth modul je toto pouze další funkce a není tedy nutná pro základní fungování tohoto projektu.

Displej, který jsem si vybral má rozlišení 128x32 a má úhlopříčku 0,91 palců. Je možné jej napájet přímo z Arduina a to 3.3 až 5 V. Displej má 4 piny, z nich jsou 2 pro napájení (*VCC* a *GND*), další je pro časování výpisu (*SCL*) a poslední se stará o data (*SDA*).^[6]



Obrázek 6: OLED displej. ([zdroj](#))

5.6 Ostatní

Zde uvedu v bodech maličkosti, které se dají sehnat za pár korun nebo jsou součástí výbavy dílny.

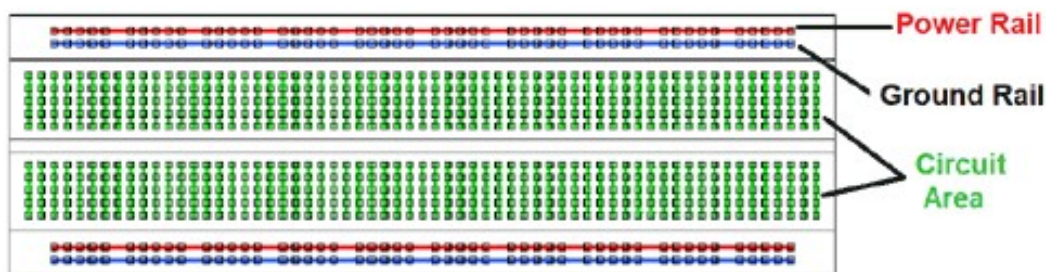
- Rezistory (1 k Ω a 2 k Ω)
- Kabeláž
- Šroubováky, kleště, pájka, ...
- Nepájivé pole (pro lehčí prvotní zapojení)

6 Zapojení nanečisto

Pro zapojení budeme používat nepájivé pole, což je „opakovaně použitelná pomůcka k navrhování prototypů elektrických obvodů a pro experimentování s obvody bez letování součástek.“^[4]

Části označené červenou a modrou čarou a znaky plus a mínus jsou spojeny horizontálně, a proto jsou většinou použité pro přívod napájení. Ostatní vstupy na desce jsou spojeny vertikálně. Tyto vstupy se používají pro ostatní zapojení.

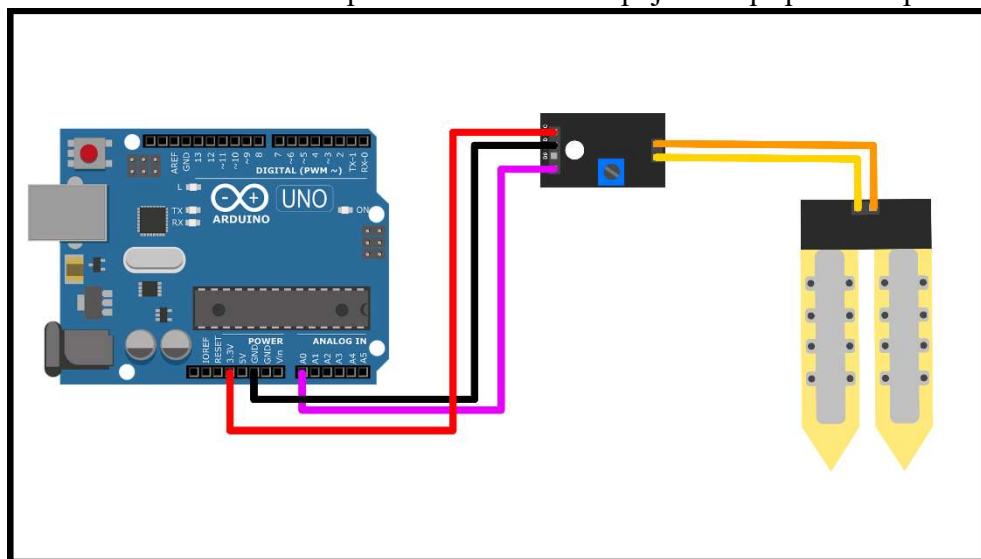
Popisné schéma se nachází na: Obrázek 7: Schéma propojení nepájivého pole.



Obrázek 7: Schéma propojení nepájivého pole

6.1 Zapojení pro zjištění vlhkosti

Pro správné fungování tohoto projektu, musíme nejdříve zjistit s jakými hodnotami vlhkosti budeme pracovat. Na toto zapojení si připravíme pouze Arduino



Obrázek 8: Schéma zapojení půdního vlhkoměru

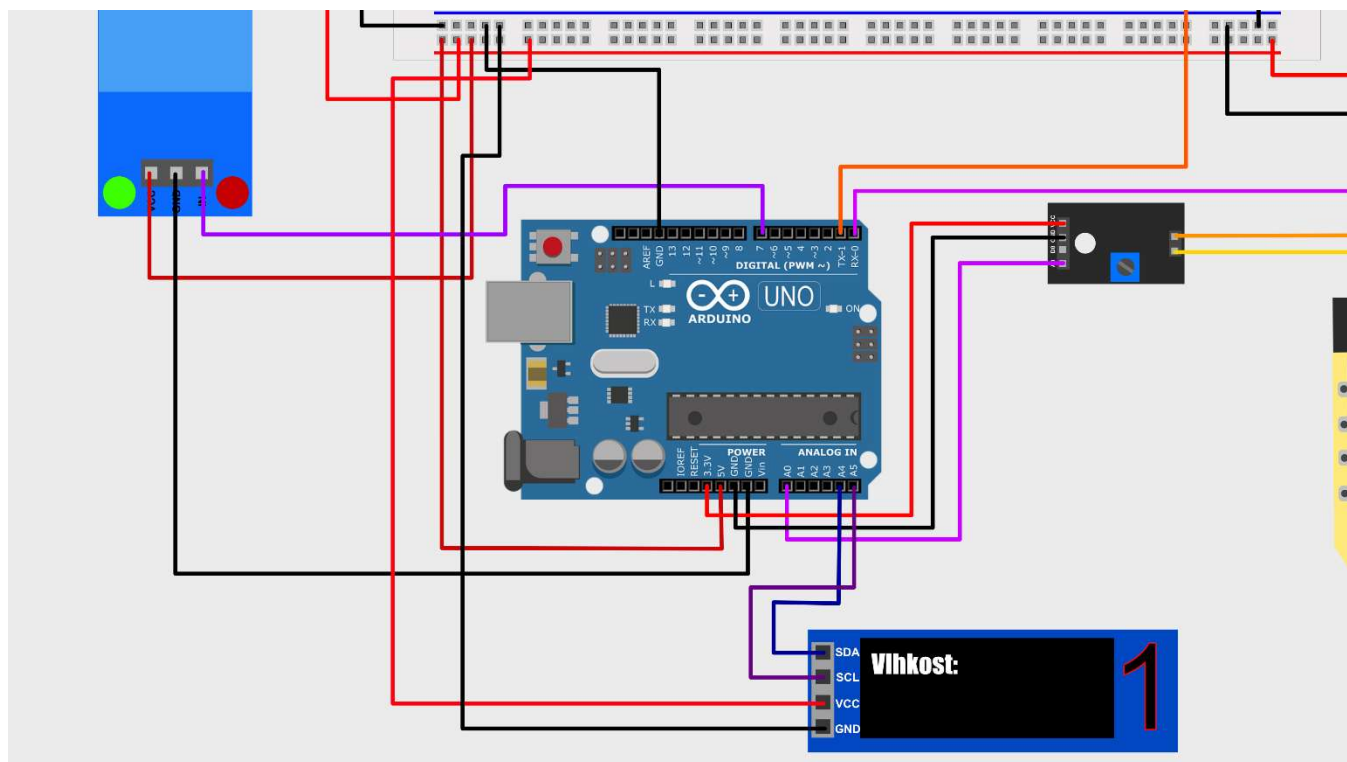
TX, do kterého bude zapojen. To provedeme tak, že vyvedeme kabel z pinu **RX** do nepájivého pole, do sloupce pod něj zapojíme rezistor **R1**, který bude zasahovat do vedlejšího sloupce. Na konci prvního rezistoru do stejného sloupce vyvedeme kabel do pinu **TX** na Arduino, pod tento kabel zapojíme rezistor **R2**, který bude opět přesahovat do vedlejšího sloupce a z jeho konce vyvedeme kabel do **GND**, který jsme si vyvedli z Arduina do nepájivého pole, kde je označen modrou čarou.

S ostatními piny na modulu již nebude problém. Pin **TX** zapojíme do **RX**, pin **GND** do podélné modré části a pin **5 V** do červeně označené podélné části.

Pozor, před nahráváním kódu do Arduina je nutné odpojit piny RX a TX od Arduina. Pokud tak neuděláte, nahrávání kódu selže.

6.4 Grafický výstup

Pro grafický výstup jsem si vybral I2C OLED display, který mě zaujal hlavně pro svoji jednoduchost zapojení. Na tomto modulu se nachází celkem 4 piny. Zapojení podle: Obrázek 11: Schéma zapojení grafického výstupu.



Obrázek 11: Schéma zapojení grafického výstupu

Z modulu (1) první pin **SDA** připojíme do pinu **A4** a pin **SCL** zapojíme do **A5** (toto je základní zapojení tohoto displeje do Arduina UNO, u jiných verzí nebo značek se může zapojení lišit). Poté už jen zbývá připojit **VCC** do napájení 5 V, které jsem si převedl na nepájivém poli v červeném řádku a pin **GND** připojit na minus, které je opět na nepájivém poli a řádek je označen modrou barvou

Displej nyní musíme nastavit v kapitole: Programování grafického výstupu.

7 Programovací část

7.1 Programování Arduina

7.1.1 Seřizování hodnot

V této části budeme programovat a nastavovat Arduino podle hodnot, které vzešly z vlhkoměru. Pro programování v této části využijte zapojení z obrázku: Obrázek 8: Schéma zapojení půdního vlhkoměru.

Pro správné nastavení budeme potřebovat 3 nádoby. Do první nádoby vložíme suchou hlínu, do druhé vlhkou hlínu a do třetí vodu. Tyto nádoby budou sloužit jako měřítko, podle kterého zjistíme, jaké hodnoty máme nastavit.

Pro výpis hodnot budeme potřebovat nahrát kód do Arduina. Připojte tedy Arduino k počítači pomocí USB a spusťte program Arduino IDE. Kód je obsažen v: Obrázek 12: Kód pro výpis hodnot z vlhkoměru.

```
#define SensorPin A0 //definujeme pin vlhkoměru
int sensorValue = 0; //proměnná hodnot z vlhkoměru

void setup() {
  Serial.begin(9600); //spuštění výpisů do Serial monitoru
}

void loop() {
  sensorValue = analogRead(SensorPin); //do sensorValue se uloží výpis z vlhkoměru
  Serial.println(sensorValue); //vypíše se hodnota
  delay(30); //prodleva 30ms
}
}
```

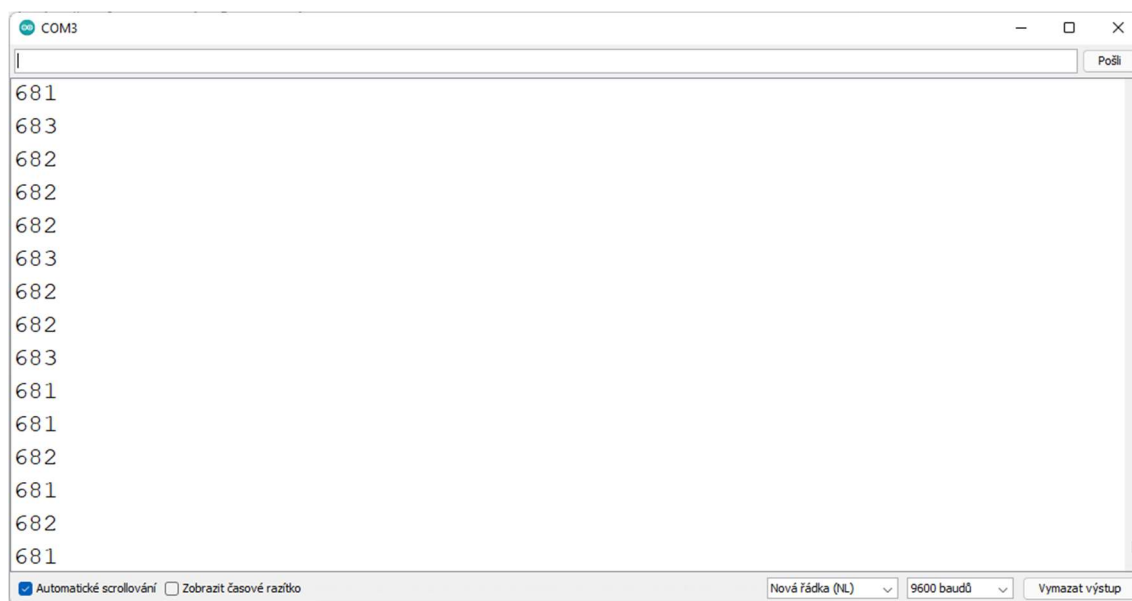
Obrázek 12: Kód pro výpis hodnot z vlhkoměru

Na začátku kódu pomocí funkce **#define** uložíme pin, do kterého je připojen vlhkoměr pod jméno **SensorPin** (příkaz je zapsán následovně: **#define <název> <co ukládáme>**). Pod touto funkcí si deklarujeme proměnnou **sensorValue**, do které uložíme počáteční hodnotu 0. Do této proměnné se bude ukládat výpis z vlhkoměru a jelikož se jedná o celé číslo, datový typ této proměnné bude **int** (Integer).

V metodě **setup** napíšeme příkaz pro spuštění záznamu do **Serial monitor**. Hodnota tohoto příkazu je 9600 bitů za vteřinu.

V metodě **loop**, ve které se bude kód stále opakovat, uložíme do proměnné **sensorValue** výpis z vlhkoměru pomocí příkazu **analogRead()**. Pod tímto příkazem už jen vypíšeme uloženou hodnotu pomocí **Serial.println()** a nastavíme prodlevu, kterou jsem zvolil na 30 milisekund.

Kód šipkou v levém rohu nahrajeme do Arduina a spustíme **Serial monitor**. Pokud vše proběhlo správně, měl by výpis vypadat podobně jako na: Obrázek 13: Výpis hodnot.



Obrázek 13: Výpis hodnot

Tyto hodnoty jsem získal, když byl vlhkoměr položen na stole. Pro přesnější měření si připravíme nádobu se suchou hlínou. Řádně do ní vpravíme vlhkoměr a průměr ze získaných hodnot si zapíšeme. Dále vlhkoměr vložíme do nádoby s vlhkou hlínou a opět si průměr z hodnot zapíšeme, a to samé uděláme i s nádobou, ve které je voda.

Nyní se můžeme orientovat ve výpisu a podle zapsaných hodnot vyhodnotit jaké hodnoty budeme zapisovat do kódu.

Mé hodnoty jsou (při připojení na 3.3 V):

- Suchá půda = 703
- Vlhká půda = 315
- Voda = 420

7.1.2 Přidání relé spínače a čerpadla

V této podkapitole upravíme kód tak, aby fungoval se schématem vyobrazeném na: Obrázek 9: Schéma zapojení s čerpadlem a relé.

```
#define SensorPin A0 //definujeme pin vlhkoměru
int sensorValue = 0; //proměnná hodnot z vlhkoměru
int sensorValueP; //proměnná, do které převádíme hodnoty vlhkoměru na procenta

int suchost = 30; // proměnná, do které v procentech zapíšeme požadovanou porovnávací hodnotu
#define rele 7

void setup() {
  Serial.begin(9600); //spuštění výpisů do Serial monitoru

  pinMode(rele, OUTPUT);
}

void loop() {
  sensorValue = analogRead(SensorPin); //do sensorValue se uloží výpis z vlhkoměru
  sensorValueP = map(sensorValue, 703, 420, 0, 100); //udělá měřítko od 0 do 100
  Serial.println(sensorValueP); //vypíše se hodnota v procentech

  //logická část pro relé
  if(sensorValueP <= suchost){
    digitalWrite(rele, LOW); //pokud je hodnota na senzoru menší nebo rovna suchosti, spustí se čerpadlo
  }else{
    digitalWrite(rele, HIGH); //pokud je výsledek cokoliv jiného nic se nestane
  }
}
```

Obrázek 14: Kód s čerpadlem

Před metodu *setup* jsem přidal proměnnou pro nastavování suchosti a definoval pin pro relé, které je v mém případě, podle schématu, zapojen pinem IN do pinu 7. A pro lepší přehlednost jsem vytvořil proměnnou *sensorValueP*, která bude hodnoty z vlhkoměru přepočítávat na procenta.

Metodu *setup* jsem doplnil o příkaz *pinMode()*, který říká Arduino, že s pinem obsaženém v závorce budu pracovat a atribut *OUTPUT* říká, že pin bude někam posílat hodnoty. V našem případě to bude do relé spínače.

V metodě *loop* první změna, která proběhla je, že naše nová proměnná *sensorValueP* ukládá hodnoty ve formě procent pomocí příkazu *map*. Tento příkaz funguje tak, že mu zadáme hodnoty, které představují minimum a maximum (použil jsem hodnoty z předešlé podkapitoly) a poslední dvě hodnoty slouží jako rozmezí, v kterém má hodnoty přepočítat (syntaxe: *map(<název proměnné obsahující hodnoty>, <minimální hodnota>, <maximální hodnota>, <přepočítaná minimální hodnota>, <přepočítaná maximální hodnota>);*). Tuto úpravu musíme zahrnout ve výpisu, proto jsem změnil proměnnou v *Serial.println* na *sensorValueP*.

Pomocí podmínky **if** rozhodujeme, kdy má relé sepnout a tím spustit čerpadlo. Porovnáváme tedy jestli hodnota z vlhkoměru je menší nebo rovna hodnotě uložené v proměnné **suchost**. Pokud je podmínka splněna, proběhne kód v první větvi, který nastaví pinu **rele** hodnotu **LOW** a tím pošle signál do relé. Pokud podmínka splněná není, provede se větev **else**, ve které nastavujeme pin **rele** na **HIGH**, relé nesepe.

Nyní již máme automatickou funkci zalévání.

Pozor, hodnoty z vlhkoměru nejsou úplně přesné a mohou se časem měnit, takže se může stát, že i po převedení na procenta mohou být čísla v záporu nebo nad 100.

7.1.3 Zprovoznění Bluetooth

Po zapojení podle schématu Obrázek 10: Schéma zapojení Bluetooth, můžeme pokračovat s úpravou kódu.

```
char income = 0; //proměnná pro BT

void loop() {
  //čtení signálů z Bluetooth

  if(Serial.available()>0){
    income = Serial.read();
    Serial.println(income);
  }
  switch(income){
    case '1':
      digitalWrite(rele, LOW);
      break;
    default:
      sensorValue = analogRead(SensorPin); //do sensorValue se uloží výpis z vlhkoměru
      //Serial.println(); //vypíše se hodnota
      delay(30); //prodleva 30ms
      if(map(sensorValue, 703, 420, 0, 100) >= suchost){ //kontrola když je hodnota z vlhkoměru vyšší než suchost
        digitalWrite(7, HIGH); //sepe relé
      }else{
        digitalWrite(7, LOW); //nesepe relé
      }
    }
  }
}
```

Obrázek 15:Kód pro zprovoznění Bluetooth

Nad metodu **setup** vytvoříme novou proměnnou, kterou jsem pojmenoval **income**, protože bude přijímat signál z Bluetooth, a přiřadil ji datový typ **char**.

Další změny jsou až v metodě **loop**. První změna je podmínky **if**, která se stará o příjem dat a ukládá je do proměnné **income**. Pod touto podmínkou se nachází cyklus

switch, který přepíná mezi úkony Arduina. V tomto cyklu jsou celkem dvě možnosti, označené slovem **case**. První je **1**, což je příchozí hodnota z aplikace, která bude popsána v kapitole: Vytvoření jednoduché aplikace na ovládání pro Android. Druhá je **default**, která se provede vždy, když je výsledkem proměnné **income** cokoliv jiného než **1** (zajišťuje tedy automatický mód zalévání).

Také musíme převádět **sensorValue** na **sensorValueP** až v podmínce. Tato změna je důležitá, protože jinak Bluetooth signály nemusí pracovat správně.

7.1.4 Programování grafického výstupu

Pro programování displeje jsem využil knihovny **Adafruit_GFX** a **Adafruit_SSD1306** a kód, který k nim byl napsán jako příklad, jsem poté upravil svým potřebám. Pokud tedy chcete, aby kód fungoval musíte si v Arduino IDE přes **projekt** v **přidat knihovnu** najít a nainstalovat tyto knihovny.

```
//knihovny Adafruit
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Resetuje piny (důležité pro knihovnu; v kódu nepoužito)
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);

display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //zde sdělíme kódu, že budeme používat displej

void displaySensor(){
  //prodleva pro načtení displeje
  delay(2000);
  // Vyčistí displej od minulého zobrazení
  display.clearDisplay();
  //nastaví barvu (správně by se jiná než bílá neměla používat)
  display.setTextColor(WHITE);
  //nastaví velikost písma
  display.setTextSize(1);
  //nastaví pozici kurzoru pro výpis
  display.setCursor(0,0);
  //vypíše vedle sebe následující: Vlhkost: <hodnota>%
  display.print("Vlhkost: ");
  display.print(map(sensorValue, 703, 420, 0, 100));
  display.print("%");
}
```

Obrázek 16: Kód pro nastavení displeje

V Obrázek 16: Kód pro nastavení displeje, jsem kód rozdělil do tří sekcí, které jsem oddělil černou čarou. První oddíl se zapisuje nad metodu **setup** a slouží k přidání knihoven a dva poslední řádky slouží pro nastavení knihovny.

Druhý oddíl je jediný řádek, který přidáme do metody **setup** a zařizuje oznámení o displeji na úrovni kódu.

Třetí oddíl se také nachází v metodě **setup**, ale na rozdíl od předchozí části se v tomto oddílu nastavuje naše vlastní nová metoda, kterou jsem pro přehlednost nazval **displaySensor**. Tato metoda nastaví vše, co chceme na displeji. První příkaz **delay(2000);** se nastavuje aby měl displej čas se zapnout, pokud by se vyskytnul problém s výpisem na displeji, zkuste zvýšit tuto prodlevu. Další příkaz slouží pro vyčištění displeje od všeho, co by na něm mohlo zůstat při startu. Po těchto příkazech už je jen naše nastavení výpisu. Jako první nastavíme barvu textu, která by doporučeně neměla být jiná než bílá. Poté nastavujeme velikost fontu a pozici kurzoru, nastavení této pozice se může lišit podle rozlišení displeje (v mém případě kurzoru na souřadnicích **0,0** se text vypíše do levého horního rohu). A pod tímto nastavením si už můžeme zvolit, co se bude vypisovat. Vypisování se dělá příkazem **display.print();** (pokud za **print** přidáme ještě **ln** bude se text vypisovat pod sebe). Já si vypsals vedle sebe hlášku: **Vlhkost <hodnota>%,** přičemž **<hodnota>** bude naměřená vlhkost, která se převádí na procenta až ve výpisu.

Teď už jen stačí naši metodu zakomponovat do metody **loop**, což uděláme podle kódu na: Obrázek 17: Kód pro spuštění displeje.

```
void loop() {
  displaySensor(); //spustí se výše definovaná metoda
  display.display(); //provedou se všechny operace pro displej

  default:
  sensorValue = analogRead(SensorPin); //do sensorValue se uloží výpis z vlhkoměru
  delay(30); //prodleva 30ms
  if(map(sensorValue, 703, 420, 0, 100) >= suchost){ //kontrola když je hodnota z vlhkoměru vyšší než suchost
    digitalWrite(7, HIGH); //sepne relé
    delay(500); // pro lepší funkčnost displeje
  }else{
    digitalWrite(7, LOW); //nesepne relé
  }
}
```

Obrázek 17: Kód pro spuštění displeje

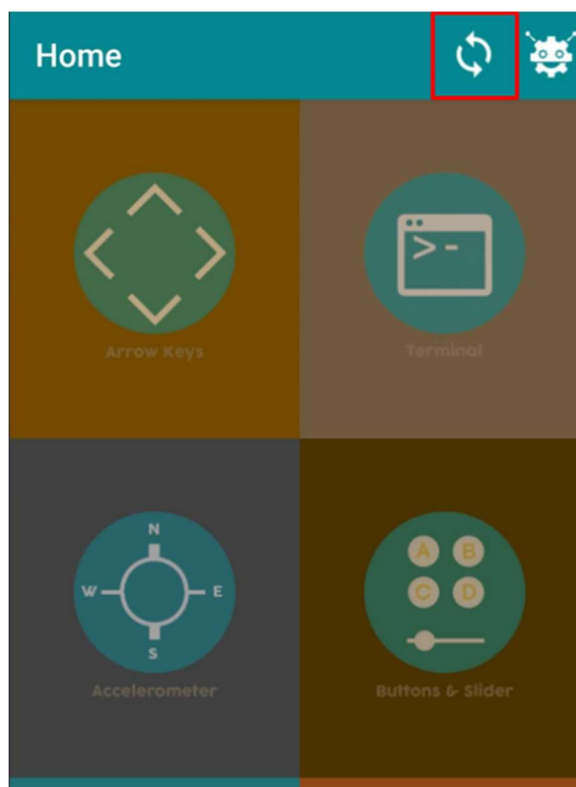
Zde hned pod metodu **loop** zapíšeme dva příkazy, které nastaví a spustí displej a potvrdí změny. V druhé části tohoto kódu upravíme v cyklu **switch** možnost **default**, do které přidáme **delay(500);** do první větve podmínky **if**. Tato prodleva se nachází pod příkazem **digitalWrite** a vylepšuje funkčnost displeje.

A tímto jsme zprovozнили displej.

7.2 Vytvoření jednoduché aplikace na ovládání pro Android

7.2.1 Aplikace Arduino BlueControl

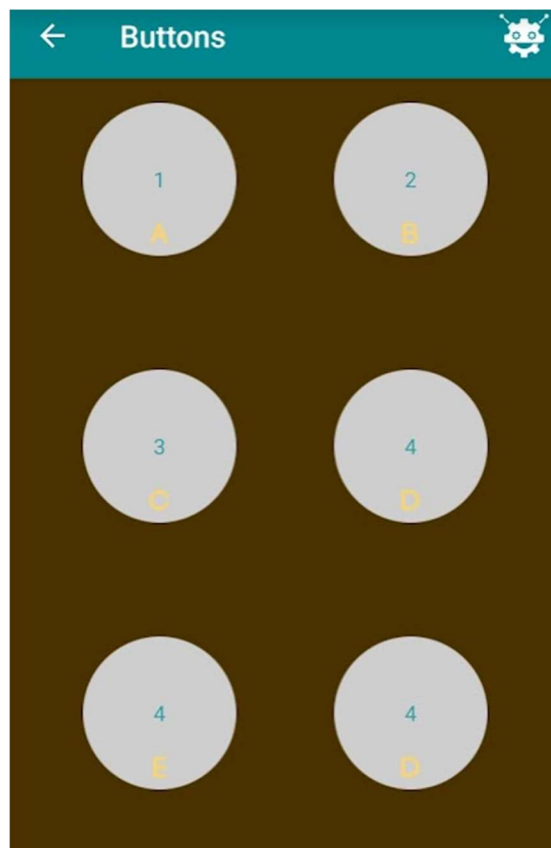
Pokud byste nechtěli tvořit vlastní aplikaci, doporučuji aplikaci Arduino BlueControl, která má mnoho funkcí a je kompatibilní s tímto projektem. Po stáhnutí aplikace se k ní musíte připojit. Pro připojení zmáčknete tlačítko, které je vyznačené na: Obrázek 18: Arduino BlueControl, připojení.



Obrázek 18: Arduino BlueControl, připojení

Po kliknutí se zobrazí tabulka, kde musíte najít zařízení s názvem **HC-06** (pokud nevidíte zařízení, zkuste se nejdříve připojit na modul přímo přes Bluetooth na mobilu a poté přes aplikaci). Pokud po vás bude chtít připojení heslo, stačí napsat **1234** (výchozí heslo pro tento modul).

Projekt poté ovládáme přes kategorii **Buttons & Slider** (Obrázek 19: Ovládání Arduino BlueControl), ve které budeme používat tlačítko **1 (A)** pro zapínání čerpadla a jakékoliv jiné pro vypnutí.



Obrázek 19: Ovládání Arduino BlueControl

7.2.2 Vytvoření vlastní aplikace v MIT AppInventor

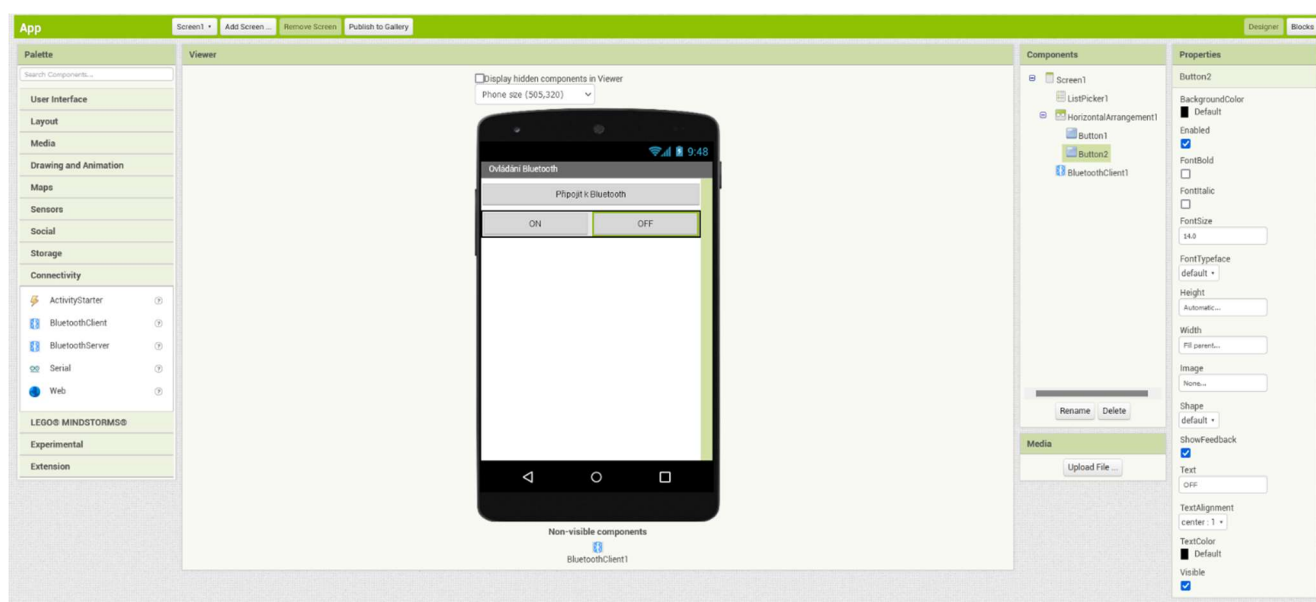
V této kapitole si vytvoříme aplikaci pomocí MIT AppInventoru. Tento nástroj je dostupný na internetu, takže se nemusí dodatečně nic stahovat. V tomto nástroji je již všechno, co potřebujeme předdefinované. Jediné, co musíme udělat je poskládat vizuální stránku a poté naprogramovat aplikaci pomocí bloků (stejné programování je například v aplikaci Scratch).

Na levé části obrazovky se nacházejí všechny komponenty, které můžete do projektu přetáhnout. Po kliknutí na nějaký komponent můžete v pravé části obrazovky nastavovat různé parametry. A v pravém horním rohu se nachází tlačítko **Blocks**, které po kliknutí otevře programovací prostředí, kde jsou standardní bloky a bloky komponentů.

Naše aplikace bude mít 3 tlačítka. Jedno velké (**ListPicker**), které bude zajišťovat připojení a dvě menší (**Button**), která budou sloužit pro ovládání. Z grafického hlediska nic jiného potřebovat nebudeme. Jen jsem pro přehlednost dal obě tlačítka do komponentu **HorizontalArrangement**, který umožní, aby byla tlačítka vedle sebe.

Tlačítka jsem pro jednoduchost pojmenoval **Připojit k Bluetooth**, **ON** a **OFF**.

Důležité je přetáhnout komponent **BluetoothClient**, který se nachází v kategorii **Connectivity** a umožní nám se připojit přes Bluetooth (tato funkce není na obrazovce mobilu viditelná, ale můžeme ji vidět pod mobilem pod nadpisem **Non-visible components**).



Obrázek 20: Prostředí MIT AppInventor

Teď již můžeme jít na programovací část. Překlikneme tedy na **Blocks** a v levé části obrazovky zvolíme kategorii **ListPicker1**, která označuje naše velké tlačítko. Do rozhraní přetáhneme příkaz **when ListPicker1.BeforePicking do**. Dále přetáhneme ze stejné kategorie ještě blok **set ListPicker1.Elements to** a připojíme k němu blok **BluetoothClient1.AddressesAndNames**. Tento set dvou příkazů vložíme do prvního bloku. Tímto zařídíme, aby si aplikace našla všechna dostupná zařízení Bluetooth.

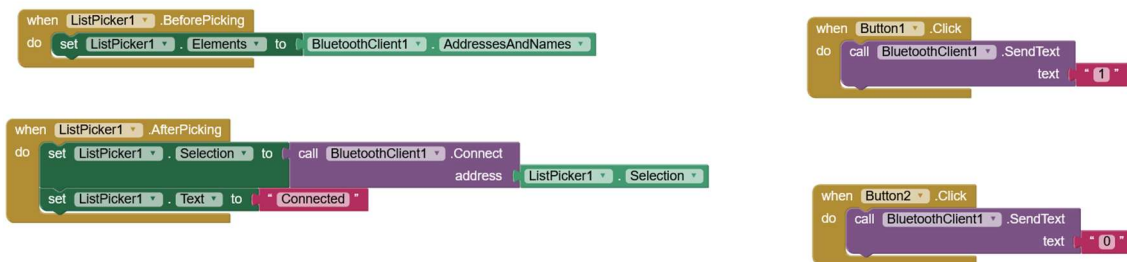
Pro **ListPicker1** přetáhneme další blok **when ListPicker1.AfterPicking do** a do něj vložíme následující bloky. Z kategorie **ListPicker1** přetáhneme **set ListPicker1.Selection to** a **set ListPicker1.Text to**. Dále z kategorie **Text** přesuneme prázdný blok (první nahoře), napíšeme do něj **Connected** a spojíme ho se **set ListPicker1.Text to**. Poté budeme potřebovat ještě z kategorie **BluetoothClient1** blok **call**

BluetoothClient1.Connect address a z kategorie **ListPicker1** blok **ListPicker1.Selection**, ten spojíme s předchozím blokem a tyto dva celky spojíme s blokem **set ListPicker1.Selection to**. Tato podmínka nám po kliknutí na tlačítko ukáže dostupná zařízení a po úspěšném připojení změní text velkého tlačítka na **Connected**.

Teď již stačí jen naprogramovat poslední dvě tlačítka. Z obou kategorií (**Button1** a **Button2**) si přetáhneme blok **when Button1/Button2.Click do**. Tyto podmínky zajistí to, co se bude dít po kliknutí na tlačítka. Do obou přetáhneme z kategorie **BluetoothClient1** blok **call BluetoothClient1.SendText text** a připojíme k nim opět prázdné textové pole. Do textového pole v **Button1** napíšeme **1** a do **Button2** zapíšeme **0**.

Kód je zhotoven v: Obrázek 21: Kód v MIT AppInvetoru.

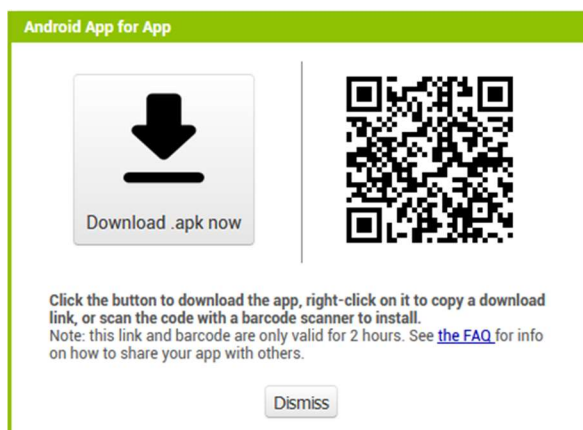
V tuto chvíli je možné aplikaci používat, ale musíme ji stáhnout do mobilu. Proto



Obrázek 21: Kód v MIT AppInvetoru

v horní liště zvolíme **Build** a klikneme na **Android App (.apk)**. Chvíli bude nástroj pracovat a poté se otevře okno podobné jako na: Obrázek 22: Výpis stažení MIT AppInventor. Budeme mít dvě možnosti:

- Naskenovat QR kód pomocí MIT AppInventor aplikace na mobilu.
- Stáhnout .apk soubor a přetáhnout a nainstalovat jej v mobilu.



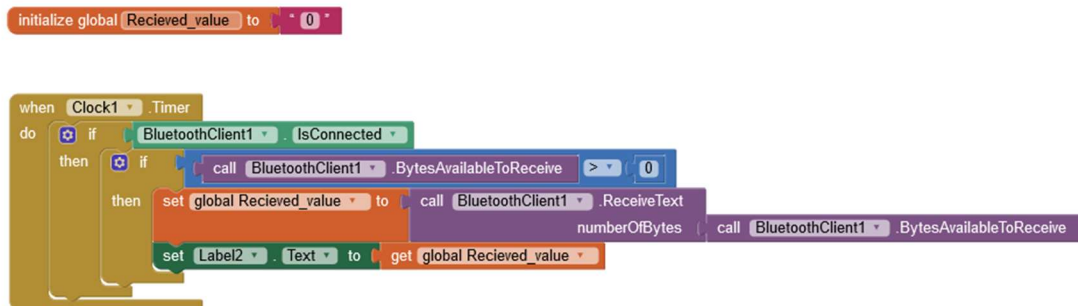
Obrázek 22: Výpis stažení MIT AppInventor

V telefonu po spuštění aplikace klikneme na tlačítko **Připojit k Bluetooth**, ze seznamu vybereme zařízení s názvem HC-06 (před názvy zařízení je MAC adresa). Pokud v seznamu není toto zařízení vidět, zkuste se připojit nejdříve přímo mobilem na Bluetooth modulu. Poté již tlačítka „ON“ a „OFF“ ovládáme čerpadlo.

Pokud chcete v aplikaci výpis hodnot z vlhkoměru, postupujte následovně. V MIT AppInventoru si v okně pro design aplikace vložíme **HorizontalArrangement**, do něj pak vložíme 3 pole **Label**. Nastavíme jim jméno na **Vlhkost**, druhou necháme prázdnou a třetí % . Poté přidáme z kategorie **Sensors** neviditelný prvek **Clock**. Nyní se přesuneme do sekce **Blocks**. Z kategorie **Clock1** přidáme podmínku **when Clock1.Timer do**. Do ní poté z kategorie **Control** přidáme dvakrát podmínku **if then**. K první podmínce připojíme z kategorie **BluetoothClient1** blok **BluetoothClient1.isConnected**. Dovnitř této podmínky přidáme naši druhou podmínku. K té připojíme z kategorie **Math** blok, který obsahuje porovnávání mezi hodnotami a změníme symbol rovná se (=) na větší než (>). Ze stejné kategorie přetáhneme na konec tohoto porovnávání blok s nulou. Do první části porovnávání přetáhneme z kategorie **BluetoothClient1** modul **call BluetoothClient1.BytesAvailableToRecieve**. Než dokončíme tuto podmínku musíme si vytvořit globální proměnnou. V kategorii **Variables** vybereme blok **initialize global name to** a místo **name** si tuto proměnnou pojmenujeme **Recieved_value**. K této proměnné připojíme prázdný textový blok a zapíšeme do něj nulu. Nyní se můžeme vrátit k vnořené podmínce, do které z kategorie **Variables** přetáhneme bloky **set global Recieved_value to** a **get global Recieved_value**. První z výše jmenovaných bloků vložíme do podmínky a v napsaném pořadí k němu připojíme dva bloky z kategorie **BluetoothClient1**, **call BluetoothClient1.RecieveText** a **call BluetoothClient1.BytesAvailableToReceive**. Pod tento příkaz do stejné podmínky

vložíme z kategorie **Label2** (prostřední textové pole v grafickém zobrazení) blok **set Label2.Text to** a připojíme k němu již přetažený blok **get global Recieved_value**.^[7]

Zhotovený kód na: Obrázek 23: Kód pro výpis hodnot v MIT AppInventoru.



Obrázek 23: Kód pro výpis hodnot v MIT AppInventoru

Ještě se musíme ujistit, že v kódu máme nastavený výpis do **Serial monitoru** jako je žlutě zvýrazněná část na: Obrázek 24: Upravený kód pro výpis dat do Serial monitoru.

```
default:
sensorValue = analogRead(SensorPin); //do sensorValue se uloží výpis z vlhkoměru
Serial.println(map(sensorValue, 703, 420, 0, 100));
delay(30); //prodleva 30ms
if(map(sensorValue, 703, 420, 0, 100) >= suchost){ //kontrola když je hodnota z vlhkoměru vyšší než suchost
digitalWrite(7, HIGH); //sepne relé
delay(500); // pro lepší funkčnost displeje
```

Obrázek 24: Upravený kód pro výpis dat do Serial monitoru

8 Návrh a 3D tisk krabičky pro Arduino

Pro tvorbu krabičky jsem si zvolil internetový nástroj Tinkercad od společnosti Autodesk. Je to software zdarma a je velmi snadné se v něm naučit modelovat, a to i bez předešlých zkušeností. Dobrá alternativa pro zkušenější může být Blender, je také zdarma a komplexnější, není ovšem online a naučit se v něm je podstatně těžší.

V Tinkercadu se tvoří, jako i v mnoha jiných programech, pomocí spojování, oddělování a dalších úprav základních objektů (krychle, koule, kvádr, ...). Pro naše potřeby nám bude stačit pouze trochu upravit krychli.

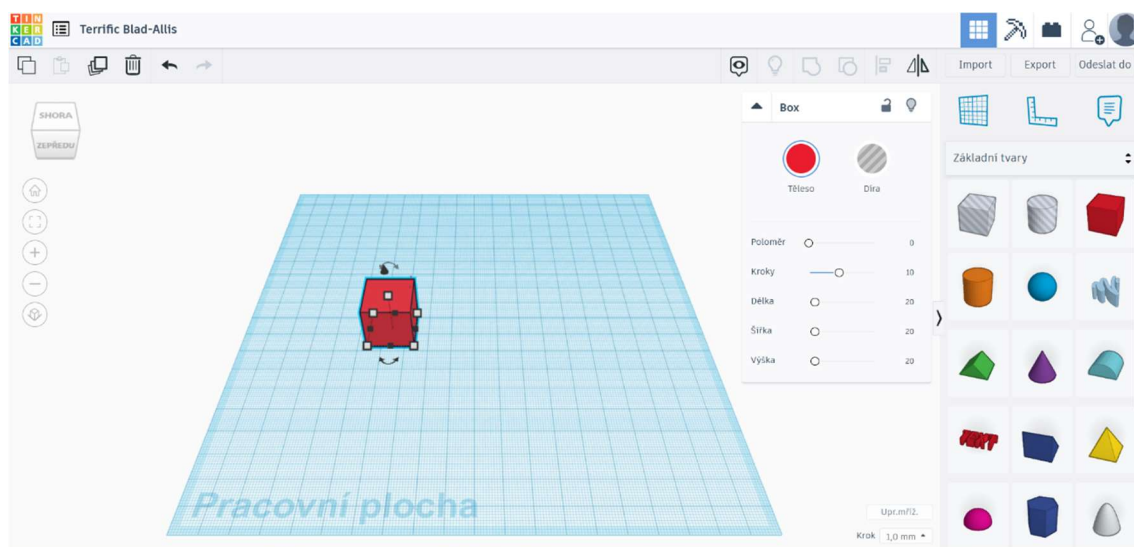
8.1 Modelování v Tinkercadu

Poté co se přihlásíte a vytvoříte nový projekt, ocitnete se ve 3D rozhraní pracovní plochy. V levém horním rohu najdeme jméno projektu a pod ním lištu akcí (kopírovat, zpět, smazat, ...). Vpravo na stejné liště se nachází nástroje pro úpravu objektů. V samotném rozhraní vidíme plochu, na které modelujeme, v levé části je ukázka úhlu pohledu a nastavení pohledu. Na druhé straně jsou objekty, které budeme vkládat. V seznamu objektů jsou první dva odlišné od ostatních. Jsou to tzv. díry. Slouží, podle jména, k vytváření děr nebo zářezů do normálních objektů.

Vkládání objektů funguje na principu přetažení do pracovní plochy. Po přesunutí na plochu se zobrazí další nastavení vybraného předmětu. Tady si můžeme všimnout, že z každého objektu lze udělat díra a naopak. Můžeme zde vybrat barvu a nastavit délky stran a další možnosti.

K upravování polohy objektů používáme šipky, které se zobrazí po vybrání objektu okolo něj. Když je objekt vybrán můžeme upravovat i jeho velikost pomocí bodů.

V pravém dolním rohu můžeme nastavit jednotky, což je velmi důležité pro modelování za účelem přesného tisku. Pod tímto nastavením je výběr toho, o jaký kus se objekt pohne.



Obrázek 25: Prostředí Tinkercad

8.2 3D tisk

K tisku budu využívat 3D tiskárnu MakerBot Replicator+, kterou mi pro tisk zapůjčil Ing. Vít Podroužek.

Tato tiskárna používá jako filament (materiál tisku) hlavně PLA, ale firma MakerBot nabízí i tvrzené plastové filamenty, které obsahují bronz, měď nebo dřevo pro zpevnění. Maximální plocha tisku je 29,5 x 19,5 x 16,5 (délka x šířka x výška). Tiskárna nabízí svůj vlastní software na tisk a dá se k ní připojit přes Ethernet, WIFI nebo USB. Nabízí také zajímavou funkci kamery, přes kterou můžete na dálku sledovat tisk ^[8]

Tiskárna je sice tvořená tak, aby ji dokázal používat i amatér, ale cena, která sahá ke 2 000 dolarům, omezuje okruh především na profesionály. Jelikož jinou tiskárnu jsem k dispozici neměl, použil jsem tuto, ale neměl by být problém můj model krabičky vytisknout na jiné tiskárně s podobnými nebo dokonce stejnými výsledky. Jako alternativu bych doporučil například tiskárny od společnosti Creality.

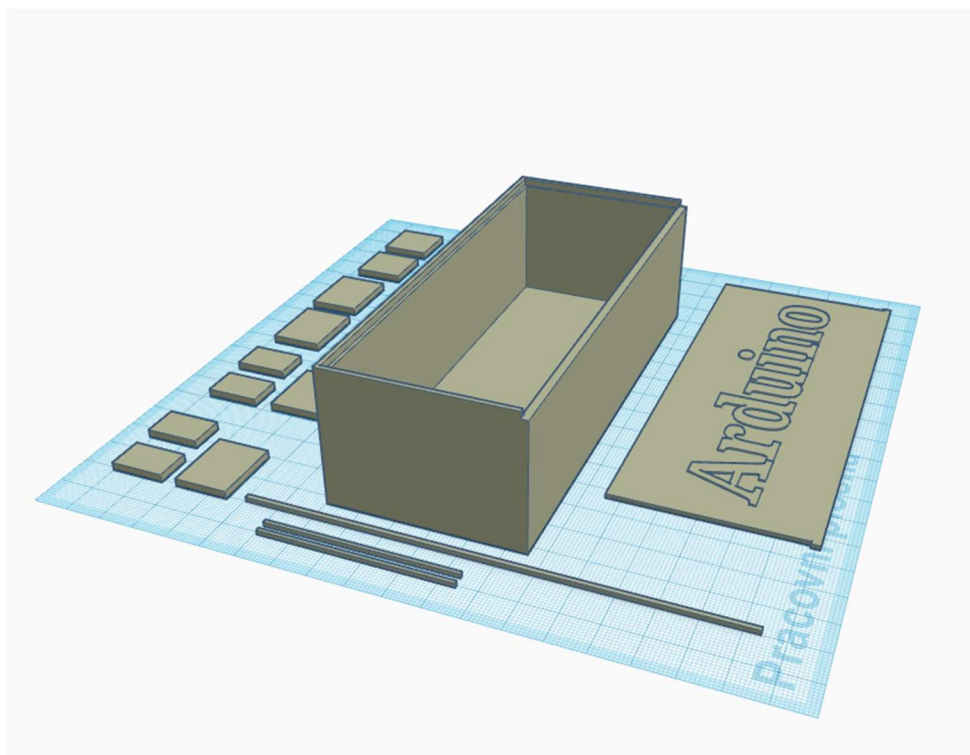
Kromě drobného broušení pro zahlázení chyb 3D tisku, bude na krabičce potřeba udělat otvory na displej, vývody kabelů a napájení. Tyto úpravy si může každý udělat podle svých představ.

Krabičku, kterou jsem pro tento projekt vymodeloval má rozměry:

- Délka 135 mm
- Šířka 55 mm
- Výška 40 mm

Účelně jsem nechal nějaký prostor prázdný, protože plánuji různá vylepšení a je lepší být připraven.

Model tvoří krabička pro Arduino a pár kotvicích prvků. Tento model je dostupný ve formátu .stl jako příloha této práce.



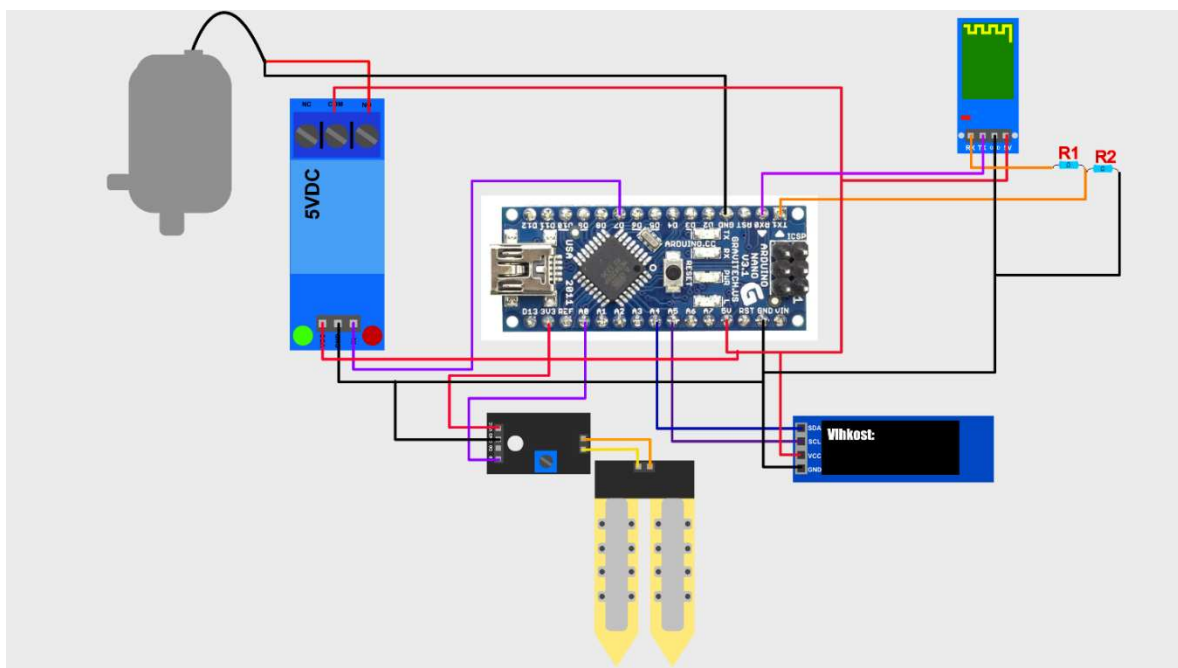
Obrázek 26: Finální 3D model



Obrázek 27: Vytisknutá krabička

9 Konečné zapojení

V této kapitole vyměníme Arduino UNO za Arduino Nano, které bude díky své kompaktnosti pasovat do vytisknuté krabičky. Tato část bude vyžadovat ovládání pájky. Zapojení bude stejné až na rozdíl, že tentokrát nebudeme používat nepájivé pole. Takže upravené schéma je na: Obrázek 28: Schéma konečného zapojení.



Obrázek 28: Schéma konečného zapojení

10 Finální produkt



Obrázek 29: Chytrá zahrada připojená ke květině



Obrázek 30: Zkompletovaná chytrá zahrada

11 Závěr

Při psaní této práce jsem si dal za úkol, abych předal mé dosavadní znalosti dál. Snažil jsem se popsat návod na vytvoření chytré zahrady tak, aby tomu mohl rozumět i amatér a snad se mi to podařilo.

Ač se projekt při psaní různě upravoval a měnil svoji formu, hlavního cíle jsem nakonec úspěšně dosáhl. Tímto cílem bylo zprovoznit Arduino, aby vykonávalo kód, který napíšu a popsat svůj postup pro ostatní.

Projekt měl původně mít více funkcí, ale zabránila tomu má nezkušenost s danými moduly. Naštěstí se dá Arduino modifikovat i později, a proto plánuji další funkce postupně přidávat.

Seznam použité literatury

- [1] *Single-Board Computer (SBC)* [online], (anglicky). Edmonton (Kanada): Techopedia, 2017 [cit. 2021-12-18]. Dostupné z: <https://www.techopedia.com/definition/9266/single-board-computer-sbc>
- [2] *Co je to Arduino?* [online]. HWKITCHEN, c2014-2021 [cit. 2021-12-19]. Dostupné z: <https://bastlima.hwkitchen.cz/co-je-to-arduino/>
- [3] *Lekce 1 - Seznámení s Arduinem* [online]. Adam Ježek; ITnetwork, 2014 [cit. 2021-12-19]. Dostupné z: <https://www.itnetwork.cz/hardware-pc/arduino/arduino-seznameni>
- [4] *Nepájivé pole* [online]. Wikipedia, c2022 [cit. 2022-02-27]. Dostupné z: https://cs.wikipedia.org/wiki/Nep%C3%A1jiv%C3%A9_pole
- [5] *Bluetooth modul HC-06 RS232* [online]. Drátek.cz, c2022 [cit. 2022-3-31]. Dostupné z: <https://dratek.cz/arduino/1720-bluetooth-modul-hc-06-rs232.html>
- [6] *IIC I2C OLED 0.91" 128x32 Modrý 3,3 V 5 V Pro IOT Arduino Raspberry* [online]. Drátek.cz, c2022 [cit. 2022-3-31]. Dostupné z: <https://dratek.cz/arduino/1480-iic-i2c-displej-oled-0.91-128x32-modry-3-3-v-5v-pro-iot-arduino-raspberry.html>
- [7] *Bluetooth Arduino RECEIVE data + Chart* [online], (anglicky). Autor: Electronoobs, na: YouTube [cit. 2022-4-1] Dostupné z: <https://www.youtube.com/watch?v=JQ3tDhpmSFE>.
- [8] *THE MAKERBOT REPLICATOR+* [online], (anglicky). MakerBot, c2022 [cit. 2022-3-31]. Dostupné z: <https://www.makerbot.com/3d-printers/replicator/>

Seznam použitých obrázků

Obrázek 1: Vývojové prostředí Arduino IDE	10
Obrázek 2: Arduino UNO. (zdroj)	13
Obrázek 3: Půdní vlhkoměr pro jednodeskové počítače. (zdroj).....	14
Obrázek 4: Modul relé. (zdroj)	15
Obrázek 5: Bluetooth modul. (zdroj)	15
Obrázek 6: OLED displej. (zdroj).....	16
Obrázek 7: Schéma propojení nepájivého pole.....	17
Obrázek 8: Schéma zapojení půdního vlhkoměru	17
Obrázek 9: Schéma zapojení s čerpadlem a relé.....	18
Obrázek 10: Schéma zapojení Bluetooth	19
Obrázek 11: Schéma zapojení grafického výstupu	20
Obrázek 12: Kód pro výpis hodnot z vlhkoměru.....	22
Obrázek 13: Výpis hodnot	23
Obrázek 14: Kód s čerpadlem.....	24
Obrázek 15:Kód pro zprovoznění Bluetooth	25
Obrázek 16: Kód pro nastavení displeje	26
Obrázek 17: Kód pro spuštění displeje	27
Obrázek 18: Arduino BlueControl, připojení	28
Obrázek 19: Ovládání Arduino BlueControl	29
Obrázek 20: Prostředí MIT AppInventor.....	30
Obrázek 21: Kód v MIT AppInventoru	31
Obrázek 22: Výpis stažení MIT AppInventor.....	32
Obrázek 23: Kód pro výpis hodnot v MIT AppInventoru	33
Obrázek 24: Upravený kód pro výpis dat do Serial monitoru	33
Obrázek 25: Prostředí Tinkercad	35
Obrázek 27: Finální 3D model.....	36
Obrázek 26: Vytisknutá krabička.....	37
Obrázek 28: Schéma konečného zapojení	37
Obrázek 29: Chytrá zahrada připojená ke květině	38
Obrázek 30: Zkompletovaná chytrá zahrada	38

Seznam příloh

Příloha 1 – serizovani_hodnot.ino

Příloha 2 – serizovani_hodnot.png

Příloha 3 – pridani_rele.ino

Příloha 4 – pridani_rele.png

Příloha 5 – Bluetooth.ino

Příloha 6 – Bluetooth.png

Příloha 7 – konecny_kod.ino

Příloha 8 – konecne_zapojeni.png

Příloha 9 – App.apk