

Hrátky s arduinem

Autoři: Vojta Lajka, Honza Lajka

Vedoucí projektu: Václav Kohout

29.6 – 13.7.2024

Plasnice v Orlických horách

Anotace

Cílem projektu bylo naučit se pracovat s jednodeskovým počítačem Arduino. Zjistit, jak počítač programovat, jak k němu připojovat jiná elektronická zařízení a jak je pomocí něho ovládat. Námí zvoleným výstupem projektu byla konstrukce funkčního elektronického trezoru.

Poděkování

Chtěli bychom poděkovat našemu vedoucímu Václavu Kohoutovi za to, že nám poskytl veškeré materiály, nápady a odbornou pomoc kterou jsme k vyhotovení projektu potřebovali. Zároveň bychom rádi poděkovali i Bobrovi za další pomoc při skládání trezoru dohromady. Martina a Šimon nám poskytli dobré rady ke konstrukci a lepení.

Obsah

Úvod	1
Učení se s Arduinem	2
Trezor	3
LED displej	3
Číselná klávesnice	4
Zamykací mechanismus	5
Pospojování součástek a ovládací kód	6
Konstrukce trezoru	8
Použitý materiál a el. součástky	8
Postup	8
Závěr	9
Zdroje	10

Úvod

Arduino je jednodeskový počítač ideální pro technické projekty. Dá se programovat pomocí jazyku Wiring (odvozeného z C#) přes program Arduino IDE. Pro Wiring existuje spousta knihoven na ovládání různých elektrických zařízení (my využíváme knihovnu na monitor, servo a číselnou klávesnici).

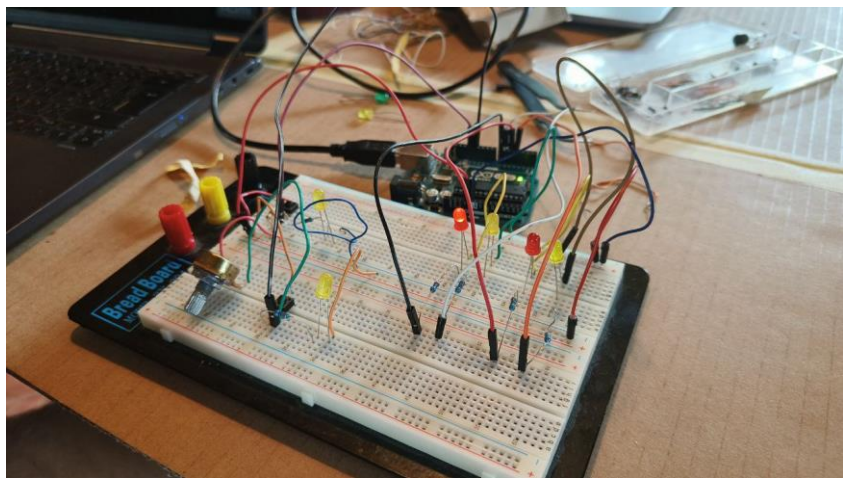
Při práci na projektu jsme se hlavně snažili s Arduinem naučit zacházet a následně své znalosti otestovat větším projektem za který jsme si na doporučení od vedoucího vybrali sestavení trezoru s číselnou klávesnicí a čtyřmístným kódem pro otevření.

Učení se s Arduinem

Aby nedošlo k poškození fyzického Arduina, rozhodli jsme se spolu s vedoucím, že by bylo dobré seznámit se s Arduinem nejprve ve webovém simulátoru. Našli jsme simulátor Arduina Uno na webové stránce Wokwi a začaly podle návodu (odkaz ve zdrojích) zapojovat obvody s LEDkami a tlačítky.

Když jsme si byli jistí, že Arduino nezkratujeme prvním sestaveným obvodem, tak jsme se pustili do hraní s Arduino UNO (které je uživatelsky nejpřívětivější) a zapojili své první reálné obvody s diodami a tlačítky. S tlačítkem jsme měli ze začátku problémy, než jsme zjistili, jak ho správně zapojit. Potom jsme se vrhli trochu více do programování a sepsali jsme jednoduchý program který postupně rozsvítil a zhasnul diody, když se zmáčkla 2 tlačítka ve stejnou chvíli. Ke každému tlačítku jsme ještě dali kontrolní diodu, která se rozsvítila, když se zmáčklo. Níže je vidět použitý kód.

```
int Led1 = 4;
int Led2 = 5;
int Led3 = 6;
int Led4 = 7;
int Button1 = 12;
int Button2 = 13;
void setup() {
  pinMode(Led1, OUTPUT);
  pinMode(Led2, OUTPUT);
  pinMode(Led3, OUTPUT);
  pinMode(Led4, OUTPUT);
  pinMode(Button1, INPUT);
  pinMode(Button2, INPUT);
}
int x = 500;
void loop() {
  if (digitalRead(13) == HIGH && digitalRead(12) == HIGH)
  {
    for (int i = 4; i <= 7; i++)
    {
      digitalWrite(i, HIGH);
      delay(x);
      digitalWrite(i, LOW);
    }
  }
}
```



Obrázek 1: Zpuštění cyklu LEDek stisknutím dvou tlačítek najednou

Po dokončení práce s jednoduchými obvody jsme se rozhodli, že je ideální čas pustit se do o velkého závěrečného projektu.

Trezor

LED displej

Displej, který jsme měli k dispozici, byl v principu snadnou součástíku na zapojení a ovládání, neboť pro něj existuje v jazyce Wiring knihovna. V počátečních fázích jsme měli problémy se správným definováním pinů. Poté již s displejem nebylo moc komplikací. Jediný trochu větší problém, který nás potkal bylo, že pro přejmenování segmentů monitoru bylo zapotřebí použít typ proměnné int a ne string. Níže okopírovaný kód rozsvěcí na monitoru písmeno S na prvním místě.

```
#include <TM1637Display.h>
#define CLK 2
#define DIO 1
TM1637Display display = TM1637Display(CLK, DIO);
int up = SEG_A;
int down = SEG_D;
int middle = SEG_G;
int rightUp = SEG_B;
int rightDown = SEG_C;
int leftDown = SEG_E;
int leftUp = SEG_F;
void setup() {
    display.clear();
    display.setBrightness(5);
}
void loop() {
    const uint8_t text[] = { up | middle | rightUp | leftDown | down, 0,0 ,0 };
    display.setSegments(text); // Displays S
}
```

Číselná klávesnice

S touto součástí nebyly téměř žádné problémy. Jediná obtížnost byla s převodem výchozí hodnoty z klávesnice do hodnoty použitelné v dalším kódu. Tento problém jsme vyřešili námi vytvořenou funkcí `ReadKey()` která převádí typ `char` na typ `int`, se kterým se dá dále manipulovat jako s číslem. Níže napsaný kód zobrazuje hodnotu stisknuté klávesy v Seriál monitoru a zároveň jí ukládá do proměnné `x`.

```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 3;
char hexaKeys [ROWS] [COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};
};
byte rowPins[ROWS] = {9, 8, 7, 6};
byte colPins[COLS] = {5, 4, 3};

int ReadKey(char f)
{
  if (f == '0') {return 0;}
  if (f == '1') {return 1;}
  if (f == '2') {return 2;}
  if (f == '3') {return 3;}
  if (f == '4') {return 4;}
  if (f == '5') {return 5;}
  if (f == '6') {return 6;}
  if (f == '7') {return 7;}
  if (f == '8') {return 8;}
  if (f == '9') {return 9;}
}

int x;
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
void setup() {
  Serial.begin(9600)
}
void loop() {
  char customKey = customKeypad.getKey();
  if (customKey){
    x = ReadKey(customKey);
    Serial.println(customKey);
  }
}
```



Obrázek 2: Číselná klávesnice

Zamykací mechanismus

K zamknutí trezoru jsme se rozhodli použít servo motůrek na principu západky. Se servem jsme měli asi nejvíce problémů ze všech součástí, protože vůbec nereagovalo na příkazy tak, jak mělo. Nakonec jsme zjistili, že jsme se celou dobu snažili ovládat rozbité servo a po výměně už všechno šlapalo tak, jak mělo. Níže napsaný kód otáčí servo o 180° tam a zpátky.

```
#include <Servo.h>
Servo servo;
void setup() {
  servo.attach(11);
}
void loop()
{
  for(int angle = 0; angle <= 180; angle++)
  {
    servo.write(angle);
    delay(15);
  }
  for(int angle = 180; angle >= 0; angle--)
  {
    servo.write(angle);
    delay(15);
  }
}
```



Obrázek 3: Servo motůrek

Pospojování součástí a ovládací kód

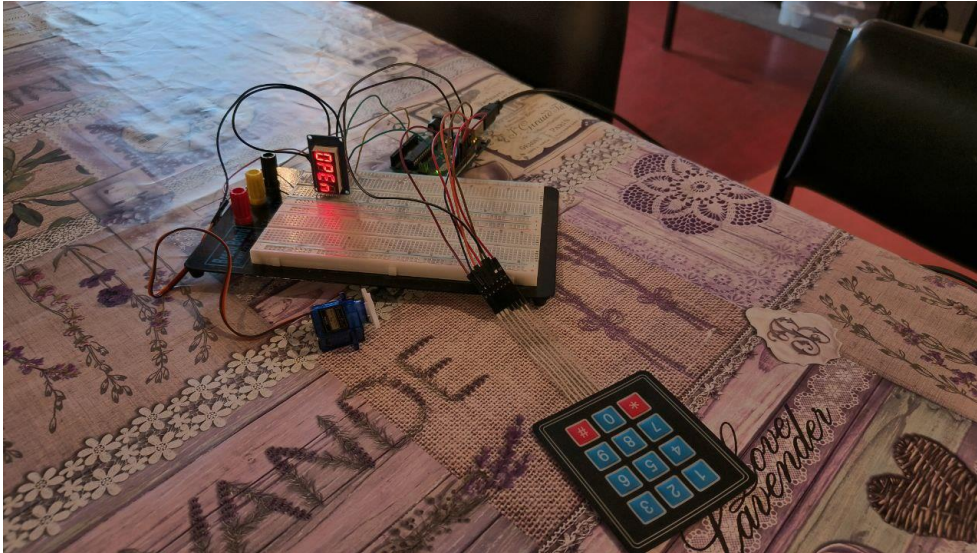
Pospojování všech předchozích součástí nebylo příliš obtížné, alespoň oproti kódu, který jsme museli v jeden moment dokonce celý přepsat, protože v něm něco zásadního nefungovalo. Při prvním pokusu jsme se pokusili použít pole pro ukládání a čtení hesla, což se nám rychle zvrtilo, další problém jsme měli s použitím Serial monitoru, kterým jsme se snažili zjišťovat co v kódu funguje a co ne, který sám o sobě kód rozbíjel.

Po katastrofálním prvním pokusu jsme se rozhodli použít pro hesla několik proměnných int. Pro kontrolu hesel jsme vytvořili níže zobrazenou funkci CheckCode(), která trezor dokázala zavřít, otevřít a napsat na monitor, že zadané heslo je špatné. Na zapisování hesla jsme použili podmínky a funkci na přečtení klávesnice.

```
bool CheckCode( int q, int w, int e, int r, int z, int u, int i, int o )
{
  if (open == true && q == 0 && w == 0 && e == 0 && r == 0 )
  {
    open = false;
    const uint8_t text[] = { leftUp | up | down | rightDown | middle,
                           leftUp | leftDown | middle | rightDown ,
                           down | leftDown | rightDown ,
                           leftDown | middle | leftUp | down };
    display.setSegments(text);
    for(int angle = 90; angle >= 0; angle--)
    {
      servo.write(angle);
      delay(15);
    }
    n = 0;
    return false;
  }
  if (q == z && w == u && e == i && r == o )
  {
    return true;
  }
  else
  {
    const uint8_t text[] = { leftDown | middle | rightDown,
                           up | rightUp | leftUp | leftDown | rightDown | down ,
                           up | leftUp | rightUp | middle | leftUp | leftDown ,
                           up | leftUp | middle | leftDown | down };
    display.setSegments(text);
    n = 0;
    return false;
  }
}
```

Protože jsme chtěli mít možnost měnit heslo bez přepisování programu, tak jsme se rozhodli přidat možnost změnit heslo přímo na trezoru. Ke změně hesla stačí zadat administrátorský kód, když je trezor otevřený. Po jeho zadání je následně možné zadat nové heslo.

Na monitoru zobrazujeme pomlčky, když se zadává heslo, „Nope“ když je heslo zadáno špatně, „Open“, když se heslo zadá správně, „Shut“, když se trezor zamkne a „Code“ když se správně zadá administrátorský kód. Při psaní nového hesla se píšou místo pomlček číslice.

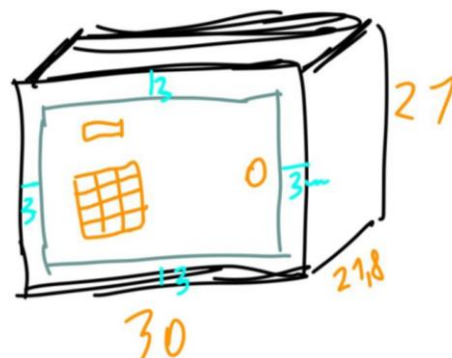


Obrázek 4: Zapojení obvodu na Arduino Uno

Konstrukce trezoru

Použitý materiál a el. součástky

- Překližky hloubky 4,5mm
- Překližka hloubky 8 mm
- Plexisklo
- Dřevěná tyčka, dřevěná prkénka
- Šrouby, matice, panty, samořezné šrouby
- Lepidlo na dřevo D3
- Tavící pistole
- Arduino, servomotor SG90, čtyřmístný sedmissegmentový displej, klávesnice 3×4, držák na 4 baterie AA



Obrázek 5: Plánek trezoru

Postup

Jako první jsme vyřezali topolové překližky jako stěny trezoru. Základ až na dveře a rám dveří jsme slepili lepidlem na dřevo, viz. obrázek.

Následně jsme vyřezali rám dveří trezoru i se dveřmi z hlubší překližky. Jisté imperfekce řezu přímočárnou pilou jsme zahladili smirkovým papírem. Přivrtali jsme panty ke dveřím a k jejich rámu. Pro lepší úchyt jsme přidali i špalík pro otevírání. Získali jsme tím funkční dveře. Do dveří jsme vyřezali otvory na displej a klávesnici. Následně nám Bobr poradil, jak vyřešit zamykací mechanismus (viz obrázek).

Pro přivrtání dveří k trezoru, které jsou moc tenké, jsme přilepili ke stěnám kousky dřeva. To umožní čistý vrt bez zaštipnutí.

Aby trezor fungoval, museli jsme ho připojit k Arduino Nano, čímž pádem jsme museli pájet. Veškeré kódy a spoje zůstaly stejné, maximálně jsme si museli přidat piny, takže to bylo zdouhavé, ale lehké. Aby bylo možné trezor používat bez připojení k počítači kabelem, použili jsme pouzdro na čtyři AA baterie, které zajistí bezproblémové napájení jak Arduina pomocí VIN pinu a taktéž serva.

Jeden z nejdůležitějších kroků byl zprovoznit trezor technologicky. Přivrtali jsme servo na jeho místo, displej jsme opatrně přilepili tavnou pistolí. Klávesnice byla samolepící, takže s ní nebyl žádný problém. Arduino Nano bylo také připevněno tavnou pistolí.

Následně jsme dřevem udělali rám okolo vnitřností trezoru, které jsme měli v plánu zakrýt dřevem, aby byly chráněné a nedaly se lehce odlepit. Nakonec jsme se ale rozhodli použít na zakrytí plexisklo s otvorem na výměnu baterek, aby dovnitř bylo hezky vidět.



Obrázek 6: Hotové vnitřnosti trezoru

Závěr

V tomto projektu jsme sestavili funkční trezor a zároveň se naučili základy ovládání počítače Arduino.

S Arduinem jsme se naučili na základních obvodech s LEDkami a tlačítky. Dále jsme své chápání zlepšili při zapojování jednotlivých součástek trezoru. Své znalosti jsme následně otestovali při zprovoznování celého trezoru. Z počátku nám většina věcí šla hladce, ale ke konci jsme měli hodně problémů a stresu se zamykacím mechanismem a přechodem z Arduino UNO na Arduino Nano.



Obrázek 7: Finální podoba trezoru

Zdroje

Koumaris. Nick. electronics-lab.com: using sg90 servo motor Arduino [online]. 2019[cit. 2024-07-10]. Dostupné z: <https://www.electronics-lab.com/project/using-sg90-servo-motor-arduino/>

Tarantula3. instructables.com: How to Use the TM1637 Digit Display With Arduino [online]. [cit. 2024-07-10]. Dostupné z: <https://www.instructables.com/How-to-Use-the-TM1637-Digit-Display-With-Arduino/>

Pattabiraman. Krishna. circuitbasics.com: how to set up a Keypad on an Arduino[online]. [cit. 2024-07-10]. Dostupné z: <https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>

Voda. Zbyšek. Tým HW Kitchen. arduino.adamit.eu: Průvodce světem Arduina[online]. [cit. 2024-07-10]. Dostupné z: https://arduino.adamit.eu/books/Zbysek_Voda_2_vydanie_2018_Pr%C5%AFvodce-sv%C4%9Btem-Arduina-CZ.pdf