

Metodický list pro robotickou pomůcku

Zařazení aktivity do RVP: <https://www.edu.cz/wp-content/uploads/2021/07/RVP-ZV-2021-zmeny.pdf>

Očekávané výstupy aktivity dle RVP:

I-9-2-01 – žák po přečtení jednotlivých kroků programu vysvětlí celý postup a určí problém, který je programem řešen

I-9-2-03 – žák vybere vhodný algoritmus pro řešení problému a zdůvodní svůj výběr

I-9-2-06 – ověří správnost programu a opraví v něm případnou chybu

I-9-2-08 – navrhne a vytvoří jednoduchý program pro řízení robota nebo jiného zařízení

Cílené dimenze informatického myšlení: algoritmizace, ladění (debugging), testování, optimalizace

Další vzdělávací cíle aktivity:

Afektivní cíle:

Žák spolupracuje se spolužákem, respektuje pravidla bezpečnosti, přebírá odpovědnost za výsledek práce.

Psychomotorické cíle:

Žák sestaví robota z 3D tištěných dílů, připojí základní elektronické komponenty a provede testování.

Kognitivní cíle:

Žák pochopí základní principy činnosti elektrických zařízení, rozpozná hlavní komponenty (motor, napájení, mikroprocesor), sestaví funkční program pro pohyb robota.

Technologické a materiální zajištění:

Didaktická stavebnice Robot BB42 (xduino.cz) – https://xduino.cz/index.php?id_product=215&id_product_attribute=0&rewrite=didakticka-stavebnice-robot-bb42&controller=product

Součásti pro konstrukci těla robota – <https://bitbeam4.cz/>

Notebook s nainstalovaným Arduino IDE

USB kabel, powerbanka (napájení robota)

3D tištěné díly robota Šmírák (tisk zajišťuje škola)

Základní ruční nářadí, kabeláž, šroubováky

Průvodce aktivitou:

Cílem je propojit 3D tisk, jednoduchou elektroniku a programování do jednoho funkčního projektu.

Žáci vytvoří vlastního robota „Šmíraka“, který dokáže sledovat černou čáru podle zadaného programu.

V rámci aktivity se žáci seznámí s funkcí elektrických obvodů, zapojením motoru a principem řízení pomocí mikrokontroléru Arduino.

Popis aktivity:

1. Úvod

Představení žákům projekt „Robot Šmírák“.

Vysvětlení principu robota, ukázka hotového modelu a stručné objasnění, jak funguje napájení, motory a ovládací deska.

Motivací pro žáky je možnost vytvořit si vlastního robota, který se skutečně pohybuje podle jejich programu.

2. Instruktaž

Seznámení žáků se zásadami bezpečné práce s elektrozařízeními (správné zapojení kabelů, manipulace s powerbankou, kontrola polarity). Žáci pracují ve dvojicích (doporučeno kombinovat žáky různé úrovně). Každá dvojice obdrží součástky a pracovní návod ke stavbě.

Vysvětlení základních částí:

- deska Arduino
- napájení z powerbanky
- motory
- propojení s USB

3. Vlastní aktivita žáka

1. fáze – konstrukce a zapojení:

Žáci sestaví robota podle návodu, připojí napájení, otestují pohyb motorů.

2. fáze – programování:

V Arduino IDE napíší jednoduchý program, který řídí pohyb robota (vpřed, vzad, otáčení, pauza).

Po nahrání programu robota otestují, sledují chování a ladí případné chyby.

Učitel žáky vede k samostatnému řešení problémů – ladění chyb, optimalizace kódu.

3. fáze – testování:

Žáci nechají robota projet připravenou trasu. Zaznamenají, jak robot reaguje na příkazy, a v případě potřeby upraví kód.

4. Závěr

Žáci prezentují své roboty, ukazují pohybové sekvence a krátce popíší, jak program funguje.

Učitel vede závěrečnou reflexi: jaké problémy museli řešit, co se naučili, jak by robota mohli vylepšit.

Následuje úklid pracoviště a odevzdání zapůjčených komponent.

Ukázka hotového programu:

```
//Připojení knihoven pro měření světla
```

```
#include <Wire.h>
```

```
#include <BH1750.h>

BH1750 luxSenzor;

//Měření vzdálenosti

#define encoderPinA 2

#define encoderPinB 3

int pinA = 0;

int pinB = 0;

float circuit = 188.5;

float encoderPulse = 960;

float distanceA = 0;

float distanceB = 0;

float distance = 0;

volatile long encoderCountA = 0;

volatile long encoderCountB = 0;

//piny motory

int mLs = 9; //směr levý motor

int mRs = 6; //směr pravý motor

int mLp = 7; // rychlost levý motor

int mRp = 8; // rychlost pravý motor

//pomocne promenne pro rychlost

int speed = 125; //volim z 100..127

int mLp_speed; //rychlost motoru po korekci

int mRp_speed;

//promenne pro vypocet korekce motoru

float p;

float d = 1;

float Err;

float lastErr;

int Corr;

//proporcionální korekce (nize ji pocitam)

//diferencialni korekce, menim kdyz se vlni moc kolem cary
```

```
//chyba

// posledni merena chyba kvuli diferencialni korekci

//celkova korekce - ovlada rychlost, která ma jencela cisla 0 - 255

float Hmax0, Hmax1, Hmax2, Hmax3; //max hodnoty cerne pri kalibraci

float Hmin0, Hmin1, Hmin2, Hmin3; // min hodnory bile urcene pri kalibraci

float Hakt0, Hakt1, Hakt2, Hakt3; //opakovane nacistane hodnoty z cidel a zkalibrujeme a ulozime do Hakt

// promenne pro mereni vzdalenosti

float vzdalenost;

float odezva;

int pTrig = 13;

int pEcho = 12;

//Bluetooth

// připojení knihovny SoftwareSerial

#define RX 11

#define TX 10

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(TX, RX);

void setup() {

  Serial.begin(9600); // seriova konzole kvuli odecitani hodnot

  BTSerial.begin(9600);

  luxSenzor.begin();

  //pripojeni serva na pin 9 a otoceni na uhel 90

  /*myservo.attach(9);

  myservo.write(pos);*/

  //LED pin

  //na kterem pinu je pripojene servo

  pinMode(LED_BUILTIN, OUTPUT);

  //nastaveni pinu ovladajici motory jako vystupni

  pinMode(mLs, OUTPUT);

  pinMode(mRs, OUTPUT);
```

```
pinMode(mLp, OUTPUT);
pinMode(mRp, OUTPUT);
pinMode(encoderPinA, INPUT);
pinMode(encoderPinB, INPUT);
// nastavení pTrig s pEcho (ultrazvukové čidlo)
pinMode(pTrig, OUTPUT);
pinMode(pEcho, INPUT);
// nastavení smeru a zastavení (pro jistotu) motoru
digitalWrite(mLs, LOW); //směr dopředu
digitalWrite(mRs, LOW);
analogWrite(mLp, 0); //rychlost 0
analogWrite(mRp, 0);
// kalibrace černé bílé na všech senzorech
for (int i = 0; i < 25; i++) {
    digitalWrite(LED_BUILTIN, HIGH); //když bliká rychle nasadíme robota na černou caru
    delay(100); // čeká 100 ms
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
}

Hmax0 = analogRead(A0);
Hmax1 = analogRead(A1);
Hmax2 = analogRead(A2);
Hmax3 = analogRead(A3);
delay(1000);
for (int i = 0; i < 8; i++) {
    digitalWrite(LED_BUILTIN, HIGH); //když bliká pomalu nasadíme robota na bílou caru
    delay(500);
    digitalWrite(LED_BUILTIN, LOW);
    delay(200);
}

Hmin0 = analogRead(A0);
```

```
Hmin1 = analogRead(A1);
Hmin2 = analogRead(A2);
Hmin3 = analogRead(A3);
delay(3000);

//vypocet p
p = speed / 150.0; //kdyz se vlni na care, jeste vynasobim treba 0.8 nebo 150.0 zmenim na 200.0
/*Serial.println("");
Serial.print ("Parametr p: ");
Serial.println (p);
Serial.println("Hodnoty Hmax: ");
Serial.print("Hmax0: ");
Serial.print(Hmax0);
Serial.print(" Hmax1: ");
Serial.print(Hmax1);
Serial.print(" Hmax2: ");
Serial.print(Hmax2);
Serial.print(" Hmax3: ");
Serial.println(Hmax3);
Serial.println("Hodnoty Hmin: ");
Serial.print("Hmin0: ");
Serial.print(Hmin0);
Serial.print(" Hmin1: ");
Serial.print(Hmin1);
Serial.print(" Hmin2: ");
Serial.print(Hmin2);
Serial.print(" Hmin3: ");
Serial.println(Hmin3);
delay (10000);*/
attachInterrupt(digitalPinToInterrupt(encoderPinA),handleEncoderA,RISING);
attachInterrupt(digitalPinToInterrupt(encoderPinB),handleEncoderB,RISING);
```

```

}

void loop() {

    //čtení hodnot z infra čidel a převod na Err
    Hakt0 = -(analogRead(A0) - Hmin0) / (Hmax0 - Hmin0) * 150.0; // vahy muzeme menit
    Hakt1 = -(analogRead(A1) - Hmin1) / (Hmax1 - Hmin1) * 50.0;
    Hakt2 = (analogRead(A2) - Hmin2) / (Hmax2 - Hmin2) * 50.0;
    Hakt3 = (analogRead(A3) - Hmin3) / (Hmax3 - Hmin3) * 150.0;

    //secteni hodnot
    Err = Hakt0 + Hakt1 + Hakt2 + Hakt3;


    //Vypocet korekce
    Corr = p * Err + d * (Err - lastErr);

    // nastav rychlosti motoru
    Corr = p * Err + d * (Err - lastErr);
    int mLP_speed = constrain(speed + Corr, 0, 255);
    int mRp_speed = constrain(speed - Corr, 0, 255);
    analogWrite(mLp, mLp_speed);
    analogWrite(mRp, mRp_speed);
    lastErr = Err;


    uint16_t lux = luxSenzor.readLightLevel();
    measureDistance();


    //zjisteni vzdalenosti
    /*digitalWrite(pTrig, LOW);
    delayMicroseconds(3);
    digitalWrite(pTrig, HIGH);
    digitalWrite(pTrig, LOW);
    odezva = pulseIn(pEcho, HIGH);

    vzdalenost = odezva / 58.31; // nebo vzdalenost = odezva * 0.01715, odezva v mikrosek., vzdálenost
    v cm, zvuk 343 m/s

```

```
//podminka pro zastaveni na 2s
if (vzdalenost < 10) {
    analogWrite(mLp, 0);
    analogWrite(mRp, 0);
    delay(2000);
}*/
/*Serial.print ("Hakt: ");
Serial.print("Hakt0: ");
Serial.print(Hakt0);
Serial.print(" Hakt1: ");
Serial.print(Hakt1);
Serial.print(" Hakt2: ");
Serial.print(Hakt2);
Serial.print(" Hakt3: ");
Serial.print(Hakt3);
Serial.print(" Err: ");
Serial.print(Err);
Serial.print (" Corr: ");
Serial.print (Corr);
Serial.print (" Levy motor: ");
Serial.print (mLp_speed);
Serial.print (" pravy motor: ");
Serial.println (mRp_speed);*/
Serial.print("A = ");
Serial.print(encoderCountA);
Serial.print(" , ");
Serial.print("B = ");
Serial.print(encoderCountB);
Serial.print(" , ");
Serial.print("mm = ");
Serial.print(distance);
```



```
Serial.print(" , ");
Serial.print("Luxů: ");
Serial.println(lux);
BTSerial.print("Distance:  ");
BTSerial.println(distanceA);
BTSerial.print("Luxů:
BTSerial.println(lux);
}
void handleEncoderA(){
encoderCountA++;
}
void handleEncoderB(){
encoderCountB++;
}
void measureDistance(){
    ");
distanceA = ((circuit*encoderCountA)/encoderPulse);
distanceB = ((circuit*encoderCountB)/encoderPulse);
distance = ((distanceA+distanceB)/2);
}
```

Ukázka hotového robota:

