

```

const int TRIG_PIN = 2;
const int ECHO_PIN = 3;
const int TRIG_PIN2 = 4;
const int ECHO_PIN2 = 5;
const int SMER_A1 = 8;
const int SMER_A2 = 11;
const int RYCHLOST_A = 9;
const int SMER_B1 = 12;
const int SMER_B2 = 13;
const int RYCHLOST_B = 10;
const int PL_SMEROVKA = 29;
const int PP_SMEROVKA = 47;
const int ZL_SMEROVKA = 27;
const int ZP_SMEROVKA = 32;
const int L_STRETAVACIE = 45;
const int P_STRETAVACIE = 24;
const int P_DIALKOVE = 28;
const int L_DIALKOVE = 35;
const int P_ZADNE = 23;
const int L_ZADNE = 50;
const int P_BRZDNE = 22;
const int L_BRZDNE = 51;
const int SPIATOCKA = 49;
const int SILA_DENNYCH_SVETIEL = 20;
const int FOTOREZISTOR = A3;
const int AIRBAG = 37;

void zapniTeplomer();
void zacniMeratTeplotu();
void zistiOdmeranuTeplotu();

void zapniAkcelerometer();
void merajZrychlenie();
//void zobrazNastavenieAkcelerometra();

void setup() {
  Serial.begin(115200);
  Serial.println("Seriovy port nastartovany.");

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(TRIG_PIN2, OUTPUT);
  pinMode(ECHO_PIN2, INPUT);
  pinMode(SMER_A1, OUTPUT);
  pinMode(SMER_A2, OUTPUT);
  pinMode(RYCHLOST_A, OUTPUT);
  pinMode(SMER_B1, OUTPUT);
  pinMode(SMER_B2, OUTPUT);
  pinMode(RYCHLOST_B, OUTPUT);
  pinMode(PP_SMEROVKA, OUTPUT);
  pinMode(PL_SMEROVKA, OUTPUT);
  pinMode(ZP_SMEROVKA, OUTPUT);
  pinMode(ZL_SMEROVKA, OUTPUT);
  pinMode(L_STRETAVACIE, OUTPUT);
  pinMode(P_STRETAVACIE, OUTPUT);
  pinMode(P_DIALKOVE, OUTPUT);
  pinMode(L_DIALKOVE, OUTPUT);
  pinMode(P_ZADNE, OUTPUT);
  pinMode(L_ZADNE, OUTPUT);
  pinMode(P_BRZDNE, OUTPUT);
  pinMode(L_BRZDNE, OUTPUT);
  pinMode(SPIATOCKA, OUTPUT);
  pinMode(AIRBAG, OUTPUT);
  Serial.println("Piny nastavene.");

  svetla();
  Serial.println("Svetla zapnute.");

  zapniTeplomer();
  zacniMeratTeplotu();

  zapniAkcelerometer();
}

const int DOPREDU = 1;
const int DOZADU = 2;
const int DOPRAVA = 3;
const int DOLAVA = 4;
const int STAT = 0;

long citlivostL = 0;
boolean jeZapnutyL = false;

long vypnutBrzdoveSvetla = 0;

```

```

int smer = STAT;

boolean zapnutyAsistentNarazu = 1;
int vyhýbaniePrekazke = 0;           // 0 alebo DOPRAVA alebo DOLAVA
int ktoraRychlost = 0;               // "tempomat" od 0 (plná rýchlosť) do 8 (najpomalšie)

float distance;                     // vzdialenosť od prekážky - z predného senzora (ak sa nejde dozadu)
float distanceS = 0;                 // vzdialenosť od prekážky z minulého cyklu - používa sa na meranie rýchlosti
float distanceDZ = 0;                // vzdialenosť od prekážky - zo zadného senzora (ak sa ide dozadu)

int kedyZastat = 10;                 // kedy má robot začať brzdiť pred prekážkou (v cm)
long rychlostMerania = 102;          // rýchlosť opakovania celého cyklu (v ms)

float casRN = 0;                     // čas - rýchlosť - nový: aktuálny čas, ktorý sa používa na meranie rýchlosti
float casRS = 0;                     // čas - rýchlosť - starý: čas z minulého cyklu, ktorý sa používa na meranie rýchlosti
float rychlost = 0;                  // nameraná rýchlosť

float teplota = 0;                   // nameraná teplota
float akcelerometerX = 0;            // zrýchlenie v osiach X, Y, Z
float akcelerometerY = 0;
float akcelerometerZ = 0;

float airbagDZ = 8;                  // nastavenie najnižšej hodnoty pre airbagy pri smere dozadu
float airbagDO = -8;                 // nastavenie najnižšej hodnoty pre airbagy pri smere dopredu
long vypnutAirbag = 0;

void loop() {
    unsigned long casStart = millis();
    spracujPrikazZTabletu();
    merajZrychlenie();
    if (zapnutyAsistentNarazu) {
        if (smer != DOZADU) {
            testujVzdialenosť();
        } else {
            testujVzdialenosťDZ();
        }
    }
    if (jeZapnutýL) {
        lightAssist();
    }
    if (vypnutBrzdoveSvetla > 0 && millis() >= vypnutBrzdoveSvetla) {
        digitalWrite(P_BRZDNE, LOW);
        digitalWrite(L_BRZDNE, LOW);
        vypnutBrzdoveSvetla = 0;
    }
    if (vypnutAirbag > 0 && millis() >= vypnutAirbag) {
        digitalWrite(AIRBAG, LOW);
        vypnutAirbag = 0;
    }
    zistiOdmeranuTeplotu();
    diagnostika();
    unsigned long presielCas = millis() - casStart;
    long esteCakat = rychlostMerania - presielCas;
    if (esteCakat < 0) {
        Serial.print("Zaporná doba čakania: ");
        Serial.println(esteCakat);
        esteCakat = 10;
    }
    cakajAMerajZrychlenie(esteCakat);
}

void cakajAMerajZrychlenie(long kolko) {
    long start = millis();
    for (;;) {
        if (vypnutAirbag == 0) {
            merajZrychlenie();
            testujAirbag();
        }
        long este = kolko - (millis() - start);
        if (este <= 0) {
            break;
        }
        delay(este < 10 ? este : 10);
    }
}

void diagnostika() {
    Serial.print("[");
    Serial.print("D=");
    if (smer != DOZADU) {
        Serial.print(distance);
    } else if (smer == DOZADU) {
        Serial.print(distanceDZ);
    }
}

```

```

Serial.print(" T=");
Serial.print(ktoraRychlost);
Serial.print(" S=");
Serial.print(smer);
Serial.print(" L=");
Serial.print(jeZapnutyL);
Serial.print(",");
Serial.print(citlivostL);
Serial.print(" AN=");
Serial.print(zapnutyAsistentNarazu);
Serial.print(" O=");
Serial.print(vyhybaniePrekazke);
Serial.print(" KZ=");
Serial.print(kedyZastat);
Serial.print(" RM=");
Serial.print(rychlostMerania);
Serial.print(" R=");
Serial.print(rychlost);
Serial.print(" C=");
Serial.print(casRN);
Serial.print(" t=");
Serial.print(teplota);
Serial.print(" aX=");
Serial.print(akcelerometerX);
Serial.print(" aY=");
Serial.print(akcelerometerY);
Serial.print(" aZ=");
Serial.print(akcelerometerZ);
Serial.println("]");
}

void lightAssist() {
  float u = (analogRead(FOTOREZISTOR) / 1023.0) * 5.0;
  float u2 = 5 - u;
  float r2 = 2200000;
  float i = u2 / r2;
  float r = u / i;
  if (r > citlivostL) {
    digitalWrite(L_DIALKOVE, HIGH);
    digitalWrite(P_DIALKOVE, HIGH);
    analogWrite(L_STRETAVACIE, 255);
    analogWrite(P_STRETAVACIE, 255);
  } else {
    digitalWrite(L_DIALKOVE, LOW);
    digitalWrite(P_DIALKOVE, LOW);
    analogWrite(L_STRETAVACIE, SILA_DENNYCH_SVETIEL);
    analogWrite(P_STRETAVACIE, SILA_DENNYCH_SVETIEL);
  }
}

void spracujPrikazZTabletu() {
  while (Serial.available() > 0) {
    int znak = Serial.read();
    switch (znak) {
      case 'A': dopredu();
        Serial.println("Idem dopredu");
        smer = DOPREDU;
        break;
      case 'B': dolava();
        Serial.println("Idem dolava");
        smer = DOLAVA;
        break;
      case 'C': doprava();
        Serial.println("Idem doprava");
        smer = DOPRAVA;
        break;
      case 'D': dozadu();
        Serial.println("Idem dozadu");
        smer = DOZADU;
        break;
      case 'E': zastav();
        Serial.println("Zastavam");
        smer = STAT;
        break;
      case 'F': zapnutyAsistentNarazu = true;
        Serial.println("Asistent narazu zapnuty");
        break;
      case 'G': zapnutyAsistentNarazu = false;
        Serial.println("Asistent narazu vypnuty");
        break;
      case 'H': ktoraRychlost = 1;
        if (smer != STAT) {
          nastavRychlostMotorov(0);
        }
        Serial.println("Nastaveny tempomat \"najrychlejsie\"");
        break;
    }
  }
}

```

```

case 'I': ktoraRychlost = 2;
  if (smer != STAT) {
    nastavRychlostMotorov(0);
  }
  Serial.println("Nastaveny tempomat \\"velmi rychlo\\"");
  break;
case 'J': ktoraRychlost = 3;
  if (smer != STAT) {
    nastavRychlostMotorov(0);
  }
  Serial.println("Nastaveny tempomat \\"rychlo\\"");
  break;
case 'K': ktoraRychlost = 4;
  if (smer != STAT) {
    nastavRychlostMotorov(0);
  }
  Serial.println("Nastaveny tempomat \\"viac nez stredne\\"");
  break;
case 'L': ktoraRychlost = 5;
  if (smer != STAT) {
    nastavRychlostMotorov(0);
  }
  Serial.println("Nastaveny tempomat \\"stredne\\"");
  break;
case 'M': ktoraRychlost = 6;
  if (smer != STAT) {
    nastavRychlostMotorov(0);
  }
  Serial.println("Nastaveny tempomat \\"menej nez stredne\\"");
  break;
case 'N': ktoraRychlost = 7;
  if (smer != STAT) {
    nastavRychlostMotorov(0);
  }
  Serial.println("Nastaveny tempomat \\"pomaly\\"");
  break;
case 'O': ktoraRychlost = 8;
  if (smer != STAT) {
    nastavRychlostMotorov(0);
  }
  Serial.println("Nastaveny tempomat \\"najpomalsie\\"");
  break;
case 'Q':
  vyhybaniePrekazke = DOPRAVA;
  Serial.println("Nastavene vyhybanie sa prekazkam doprava");
  break;
case 'P':
  vyhybaniePrekazke = DOLAVA;
  Serial.println("Nastavene vyhybanie sa prekazkam dolava");
  break;
case 'R': vyhybaniePrekazke = 0;
  Serial.println("Vypnute vyhybanie sa prekazkam");
  break;
case 'S':
  analogWrite(L_STRETAVACIE, 255);
  analogWrite(P_STRETAVACIE, 255);
  jeZapnutýL = false;
  break;
case 'T': analogWrite(L_STRETAVACIE, SILA_DENNYCH_SVETIEL);
  analogWrite(P_STRETAVACIE, SILA_DENNYCH_SVETIEL);
  break;
case 'U': analogWrite(L_STRETAVACIE, 255);
  analogWrite(P_STRETAVACIE, 255);
  digitalWrite(L_DIALKOVE, HIGH);
  digitalWrite(P_DIALKOVE, HIGH);
  jeZapnutýL = false;
  break;
case 'V':
  digitalWrite(L_DIALKOVE, LOW);
  digitalWrite(P_DIALKOVE, LOW);
  break;
case 'W':
  {
    jeZapnutýL = true;
    long cas = millis();
    while (Serial.available() <= 0 && millis() - cas < 1000) {
    }
    if (Serial.available() > 0) {
      int dalsi = Serial.read();
      citlivostL = (dalsi - '0' + 1) * 8000000;
    }
  }
  break;
case 'X': jeZapnutýL = false;
  digitalWrite(L_DIALKOVE, LOW);
  digitalWrite(P_DIALKOVE, LOW);

```

```

    analogWrite(L_STRETAVACIE, SILA_DENNYCH_SVETIEL);
    analogWrite(P_STRETAVACIE, SILA_DENNYCH_SVETIEL);
    break;
case 'Y':
{
    zapnutyAsistentNarazu = 1;
    long c = millis();
    while (Serial.available() <= 0 && millis() - c < 1000) {
    }
    if (Serial.available() > 0) {
        int dalsi = Serial.read();
        kedyZastat = (dalsi - '0') * 2;
    }
}
break;
case 'Z':
{
    zapnutyAsistentNarazu = 1;
    long c = millis();
    while (Serial.available() <= 0 && millis() - c < 1000) {
    }
    if (Serial.available() > 0) {
        int dalsi = Serial.read();
        rychlostMerania = (dalsi - '0' + 1) * 6;
    }
}
break;
}
}

void testujVzdialenost() {
    zistiVzdialenost();
    if (distance >= 400 || distance <= 0) {
        rychlost = 0;
    } else if (distance < 50) {
        if (vyhybaniePrekazke == DOPRAVA && smer == DOPREDU) {
            otacajSaKymJePrekazkaDoprava();
        } else if (vyhybaniePrekazke == DOLAVA && smer == DOPREDU) {
            otacajSaKymJePrekazkaDolava();
        } else {
            if (distance < kedyZastat && smer == DOPREDU) {
                zastav();
            }
        }
    }
}

void testujVzdialenostDZ() {
    zistiVzdialenostDZ();
    if (distanceDZ > 0 && distanceDZ < 50) {
        Serial.print(distanceDZ);
        Serial.println(" cm POZOR PREKAZKA BLIZSIE AKO 50 CM BRZDIT ALEBO ZMENIT SMER ");
        if (distanceDZ < kedyZastat) {
            zastav();
        }
    }
}

void testujAirbag() {
    if (smer == DOZADU && akcelerometerY > airbagDZ) {
        Serial.print("Naraz pri jazde dozadu. Zrychlenie = ");
        Serial.println(akcelerometerY);
        zastav();
    } else if (smer == DOPREDU && akcelerometerY < airbagDO) {
        Serial.print("Naraz pri jazde dopredu. Zrychlenie = ");
        Serial.println(akcelerometerY);
        zastav();
        zapniAirbag();
    }
}

void zapniAirbag() {
    digitalWrite(AIRBAG, HIGH);
    vypnutAirbag = millis() + 1500;
}

void svetla() {
    digitalWrite(P_ZADNE, HIGH);
    digitalWrite(L_ZADNE, HIGH);
    digitalWrite(P_DIALKOVE, LOW);
    digitalWrite(L_DIALKOVE, LOW);
    digitalWrite(P_SMEROVKA, LOW);
    digitalWrite(PL_SMEROVKA, LOW);
    digitalWrite(ZP_SMEROVKA, LOW);
    digitalWrite(ZL_SMEROVKA, LOW);
}

```

```

    digitalWrite(P_BRZDNE, LOW);
    digitalWrite(L_BRZDNE, LOW);
    digitalWrite(SPIATOCKA, LOW);
    analogWrite(L_STRETAVACIE, SILA_DENNYCH_SVETIEL);
    analogWrite(P_STRETAVACIE, SILA_DENNYCH_SVETIEL);
}

void dopredu() {
    smerA(1);
    smerB(1);
    nastavRychlostMotorov(255);
    digitalWrite(PP_SMEROVKA, LOW);
    digitalWrite(PL_SMEROVKA, LOW);
    digitalWrite(ZP_SMEROVKA, LOW);
    digitalWrite(ZL_SMEROVKA, LOW);
    digitalWrite(P_BRZDNE, LOW);
    digitalWrite(L_BRZDNE, LOW);
    digitalWrite(SPIATOCKA, LOW);
}

void dozadu() {
    smerA(0);
    smerB(0);
    nastavRychlostMotorov(255);
    digitalWrite(SPIATOCKA, HIGH);
    digitalWrite(PP_SMEROVKA, LOW);
    digitalWrite(PL_SMEROVKA, LOW);
    digitalWrite(ZP_SMEROVKA, LOW);
    digitalWrite(ZL_SMEROVKA, LOW);
    digitalWrite(P_BRZDNE, LOW);
    digitalWrite(L_BRZDNE, LOW);
}

void doprava() {
    smerA(1);
    smerB(0);
    nastavRychlostMotorov(150);
    digitalWrite(PP_SMEROVKA, HIGH);
    digitalWrite(ZP_SMEROVKA, HIGH);
    digitalWrite(PL_SMEROVKA, LOW);
    digitalWrite(ZL_SMEROVKA, LOW);
    digitalWrite(P_BRZDNE, LOW);
    digitalWrite(L_BRZDNE, LOW);
    digitalWrite(SPIATOCKA, LOW);
}

void dolava() {
    smerA(0);
    smerB(1);
    nastavRychlostMotorov(150);
    digitalWrite(PL_SMEROVKA, HIGH);
    digitalWrite(ZL_SMEROVKA, HIGH);
    digitalWrite(PP_SMEROVKA, LOW);
    digitalWrite(ZP_SMEROVKA, LOW);
    digitalWrite(P_BRZDNE, LOW);
    digitalWrite(L_BRZDNE, LOW);
    digitalWrite(SPIATOCKA, LOW);
}

void nastavRychlostMotorov(int rychlostBezTempomatu) {
    switch (ktoraRychlost) {
        case 1: rychlostA(255); rychlostB(255); break;
        case 2: rychlostA(225); rychlostB(225); break;
        case 3: rychlostA(200); rychlostB(200); break;
        case 4: rychlostA(175); rychlostB(175); break;
        case 5: rychlostA(150); rychlostB(150); break;
        case 6: rychlostA(125); rychlostB(125); break;
        case 7: rychlostA(100); rychlostB(100); break;
        case 8: rychlostA(75); rychlostB(75); break;
        case 0: rychlostA(rychlostBezTempomatu); rychlostB(rychlostBezTempomatu); break;
    }
}

void otacajSaKymJePrekazkaDoprava() {
    doprava();
    otacajSaKymJePrekazka();
}

void otacajSaKymJePrekazkaDolava() {
    dolava();
    otacajSaKymJePrekazka();
}

void otacajSaKymJePrekazka() {
    rychlostA(100);
    rychlostB(100);
}

```

```

    while (distance <= 50) {
        zistiVzdialenost();
    }
    Serial.println("Nasiel som volnu cestu.");
    dopredu();
}

void zistiVzdialenost() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    long duration = pulseIn(ECHO_PIN, HIGH, 25000);
    distance = (duration / 2) * 0.034;
    casRN = millis();

    float zmenaVzdialenosti = (float) (distanceS - distance) / (float) 100;
    float zmenaCasu = (float) (casRN - casRS) / (float) 1000;
    rychlost = zmenaVzdialenosti / zmenaCasu;

    distanceS = distance;
    casRS = casRN;
}

void zistiVzdialenostDZ() {
    digitalWrite(TRIG_PIN2, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN2, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN2, LOW);
    long durationDZ = pulseIn(ECHO_PIN2, HIGH, 25000);
    distanceDZ = (durationDZ / 2) * 0.034;
}

void zastav() {
    vypniA();
    vypniB();
    digitalWrite(SPIATOCKA, LOW);
    digitalWrite(PP_SMEROVKA, LOW);
    digitalWrite(PL_SMEROVKA, LOW);
    digitalWrite(ZP_SMEROVKA, LOW);
    digitalWrite(ZL_SMEROVKA, LOW);
    digitalWrite(P_BRZDNE, HIGH);
    digitalWrite(L_BRZDNE, HIGH);
    vypnutBrzdoveSvetla = millis() + 1500;
    smer = STAT;
}

void smerA(int vpred) {      // vpred = 1 ak chces ist dopredu, vpred = 0 ak chces ist dozadu
    if (vpred != 0) {
        digitalWrite(SMER_A1, LOW);
        digitalWrite(SMER_A2, HIGH);
    } else {
        digitalWrite(SMER_A1, HIGH);
        digitalWrite(SMER_A2, LOW);
    }
}

void zapniA() {
    digitalWrite(RYCHLOST_A, HIGH);
}

void vypniA() {
    digitalWrite(RYCHLOST_A, LOW);
}

void rychlostA(int kolko) {
    analogWrite(RYCHLOST_A, kolko);
}

void smerB(int vpred) {
    if (vpred != 0) {
        digitalWrite(SMER_B1, LOW);
        digitalWrite(SMER_B2, HIGH);
    } else {
        digitalWrite(SMER_B1, HIGH);
        digitalWrite(SMER_B2, LOW);
    }
}

void zapniB() {
    digitalWrite(RYCHLOST_B, HIGH);
}

void vypniB() {

```

```
    digitalWrite(RYCHLOST_B, LOW);  
}  
void rychlostB(int kolko) {  
    analogWrite(RYCHLOST_B, kolko);  
}
```