



## **Středoškolská technika 2018**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **Mobilní robot Arduino I**

Jakub Příborský

VOŠ, SPŠ a JŠ Kutná Hora  
Masarykova 197

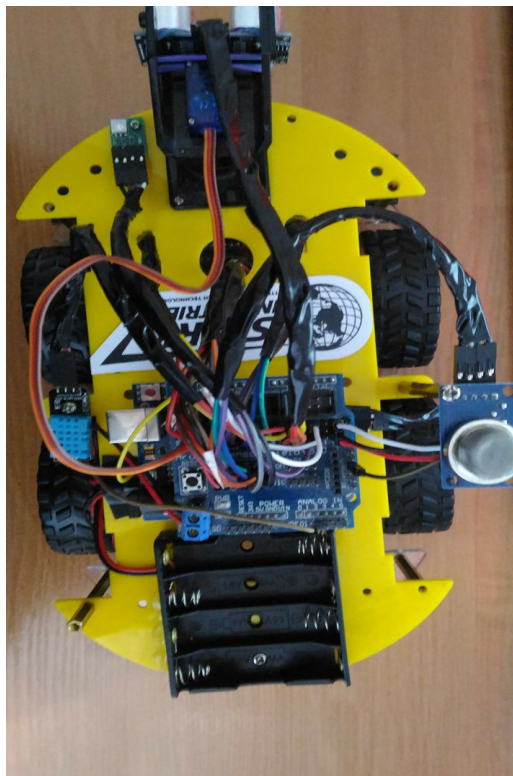
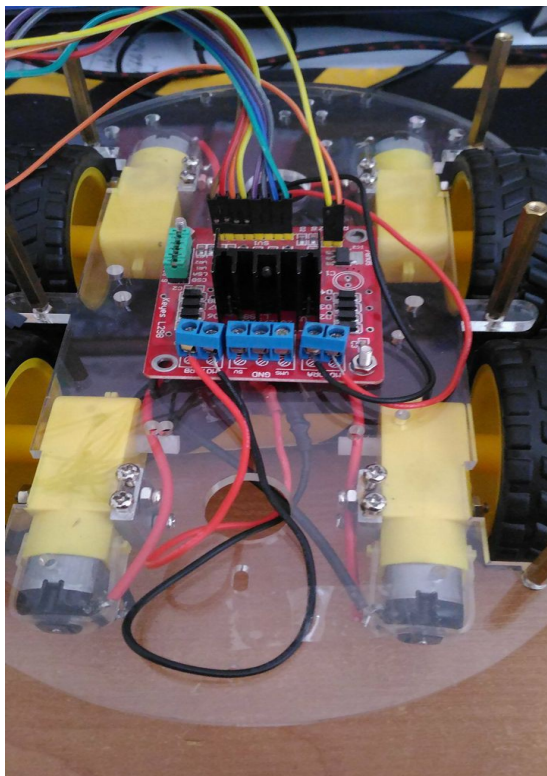


## Úvod

Pro praktickou maturitní zkoušku, jsem se rozhodl využít možnosti vlastního návrhu tématu. Cílem práce bylo vytvořit robota, který by nebyl, z hlediska jízdy, závislý na mých pokynech. Tuto myšlenku se mi podařilo realizovat pomocí ultrazvukového senzoru a IR senzorů. Mým konečným záměrem, ale bylo přidělat k robotovi senzory pro měření teploty, vlhkosti a kvality vzduchu, a tak vytvořit z robota „mobilní meteorologickou stanici“, která bude schopná odesílat data do mobilu. Během práce jsem přemýšlel jak a co udělat, jaké senzory použít či jestli se mi to vůbec podaří dokončit, ale přes tohle všechno se mi nakonec podařilo robota sestavit a naprogramovat.

### Části robota

Robot je rozdělen dvěma deskami. Na spodní desce jsou umístěny 4 převodové motory, IR senzory, box na baterie a řídicí deska pro motory L298N. Horní deska na sobě nese servo s ultrazvukovým senzorem, druhý box na baterie, senzor MQ-135, senzor DHT11 a arduino UNO s shieldem pro senzory.



### Senzor MQ-135

Senzor MQ-135 je, podle mě, nejzajímavější senzor, který na robotovi mám. Dokáže detekovat několik druhů plynů, mezi ně patří například oxid uhličitý, amoniak nebo oxidy dusíku. Díky citlivosti na tyto plyny tento senzor může sloužit například jako kouřový senzor při požáru nebo jako v mém případě senzor na měření kvality ovzduší.



V programu jsem si určil kritickou hranici pro znečištění ovzduší, takže když bude překročena určitá koncentrace plynu, tak mě robot pomocí RGB LED diody varuje. Pro první hranici kvality vzduchu používám zelenou barvu, kdy koncentrace plynů je nižší jak 5 ppm. Pokud

senzor naměří kvalitu vzduchu do této hodnoty, tak je dle mého vzduch čistý, bez jakéhokoliv nebezpečí. Druhá hranice je do 10 ppm. Zde používám signalizaci pomocí modré barvy. Tuto mez používám spíše jen jako informační a naměřím hodnoty v tomto rozsahu nejčastěji v uzavřeném prostoru, kde je vzduch „vydýchaný“. Předposlední signalizaci mám nastavenou do 200 ppm, kde je dle mého názoru vzduch stále dýchatelný, ale už rozhodně ne kvalitní. Neměl jsem možnost takovou hranici otestovat, a tak je to pouze odhad. Protože mám rozsah měřených hodnot nastavený podle sebe, nemohl jsem je porovnat s hodnotami od meteorologických stanic. Po překročení hranice 200 ppm už považuji naměřené hodnoty za nebezpečné. Pokud hodnoty stoupnou až nad 200 ppm, robot okamžitě ukončí jakoukoliv činnost a začne varovně blikat červená LED dioda doprovázená varováním odesíláném do mobilu. Ale jak již bylo zmíněno, tyto hranice jsem určoval podle svého měření a odhadů, takže od reálné nebezpečné hodnoty se mohou lišit.

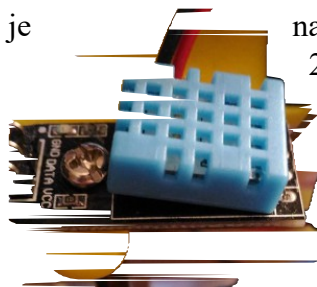
Z hlediska programové části používám příkaz „float ppm“, kdy do „ppm“ zapisuji hodnoty ze senzoru a pomocí příkazu „Serial.print“ je tisknu do sériové komunikace. Na začátku programu pak mám vloženou knihovnu a definovaný analogový a digitální pin. Dále mám vytvořené díky „int“ proměnné pro piny od diody.

```
//senzor plynů
#include "MQ135.h"           //měření
#define pinA A3             float ppm = senzorMQ.getPPM();
#define pinD 2              Serial.print("Kvalita vzduchu: ");
MQ135 senzorMQ = MQ135(pinA); Serial.println(ppm);
int modra=3;
int zelena=A4;
int cervena=A5;
```

## Senzor DHT11

Senzor DHT11 dokáže měřit teplotu a vlhkost. Tento senzor byl mezi prvními, které jsem na robota přidělal. Díky schopnosti měřit teplotu a vlhkost je velice užitečný. Lze ho použít jen jako obyčejný teploměr jak pro měření teploty vevnitř, tak i venku nebo z něj může být i skvělý základ pro domácí meteorologickou stanici. Ovšem má i nevýhodu v tom, že měří pouze do 0°C. Nejvyšší možná teplota, kterou tento senzor může změřit je 50°C.

Po několika sériích testů na přesnost měření teploty jsem dospěl ke stejné odchylce 2°C, která je napsané v datasheetu. Co se týče měření vlhkosti, tak tam je rozsah 20% až 90% vlhkosti s přesností na 5%. Tady jsem bohužel neměl šanci ověřit, zda je přesnost skutečně taková, ale soudě dle naměřených hodnot, které jsem v průběhu práce na robotovi změřil, by se dali považovat za reálné.



U tohoto senzoru jsem taktéž použil knihovnu DHT.h. Dále mám zapojené na 7 digitálním pinu čtení přímo ze senzoru a definovaný typ senzoru, v tomto případě DHT11. Pro zapisování hodnot jsem si zvolil proměnné „teplota“ a „vlhkost“, do kterých zapisuji hodnoty

díky „mojeDHT.readTemperature“ a „mojeDHT.readHumidity“ Taktéž jako u MQ-135 i tady hodnoty následně tisknu do sériové komunikace.

```
//Teplota a vlhkost
#include "DHT.h"
#define pinDHT 7
#define typDHT11 DHT11
DHT mojeDHT(pinDHT, typDHT11);

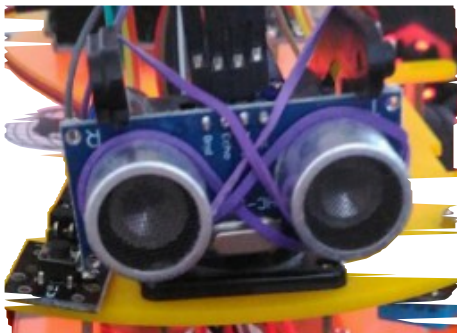
float teplota = mojeDHT.readTemperature();
Serial.print("Teplota: ");
Serial.println(teplota);
delay(500);
float vlhkost = mojeDHT.readHumidity();
Serial.print("Vlhkost: ");
Serial.print(vlhkost);
Serial.println("% ");
delay(500);
```

## Jízda

Samotné rozpořádání robota bylo ze začátku pro mě problém, ale když se mi to podařilo tak mě čekala ta nejdůležitější část a hlavní úkol mého projektu. Vytvořit program, který by dokázal robotem pohybovat a při tom si dávat pozor na překážky nebo sledovat černou čáru. Nebylo tak těžké udělat základ pro samostatnou jízdu, ale následné rozšíření a sestavení podmínek, tak aby bylo přesně dáno, co má dělat v případě překážky před sebou nebo sjetí z dráhy. Než jsem došel ke správnému pořadí příkazů a podmínek, aby se mezi sebou „nepletli“, tak jsem nad tím strávil hodně času, ale díky tomu mi to funguje, tak jak jsem si představoval.

Momentálně má robot dva způsoby jízdy pomocí 4 převodových motorů. První způsob je podle ultrazvukového senzoru pro měření vzdálenosti. Tento senzor má mnohonásobné využití. Co se mě týče, tak bych ho použil i například jako senzor na detekci pohybu u bezpečnostního systému nebo u parkovacího senzoru pro auto. Na robotovi ho mám zabudovaný kvůli detekci překážek. Tento senzor dokáže měřit s přesností na 3 mm a od vzdálenosti 2 cm do 4 m.

Ultrazvukový senzor v mém programu je nastavený tak, že pokud detekuje před robotem objekt ve vzdálenosti menší než je 25 cm, okamžitě zastaví. Poté se na 0,2 sekundy zapne na motorech opačný chod a robot tak od překážky odjede, protože někdy se stane, že kvůli špatnému úhlu robot nestihne včas odhadnout vzdálenost od objektu a narazí do něj. Poté, co couvne se ultrazvukový senzor pomocí serva otočí o 90° na levou stranu a změří vzdálenost.

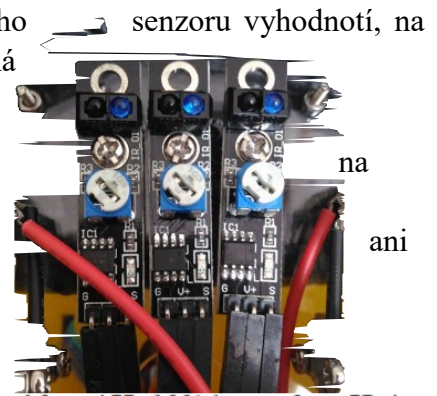


Následně se otočí na pravou stranu, takže o 180° a proces se opakuje. Když doměří, servo se vrátí zpátky na svou původní pozici (90°) a robot vyhodnotí na jaké straně je více místa. Poté se na tu stranu otočí. Nemohu říct, že se robot otočí přesně o 90°, protože záleží, jestli jsou baterie plně nabité. Po otočení se robot znovu rozjede a opět vyhodnocuje, zda je před ním místo či ne.

Druhý způsob jízdy je podle infračervených senzorů. Na robotovi mám umístěné tři tyto senzory. Nejdůležitější je prostřední IR senzor, podle kterého se celá jízda řídí. Senzory na levé a pravé straně mají za úkol, v případě sjetí z dráhy, zkontrolovat zda se pod autem

nachází černá čára. Poté se podobně jako u ultrazvukového jakou stranu se má otočit. Vzhledem k tomu, že čára má malou šířku, tak se auto otáčí pouze 0,1 sekundy.

Jak jsem již výše psal, tak robot se musí nejprve rozhodnout, jakou stranu se má otočit. Tuto možnost mi umožňuje funkce „if“ a po ní hned následuje funkce „goto“. Pokud ale na jednom IR senzoru není zachycená čára, přichází na řadu poslední podmínka. Když se k ní robot dostane, tak se začne otáčet na levou stranu, tentokrát ale o 0,2 sekundy. Po tomhle kroku, se program vrátí zpět na měření hodnot pomocí IR senzorů a následné vyhodnocování na jakou stranu se tento proces bude neustále opakovat, do té doby robot nenajde čáru.



```
if(levyIR>100&&hlavniIR<100){goto levaIR;}
if(pravyIR>100&&hlavniIR<100){goto pravaIR;}
if(hlavniIR>100){goto konecIR;}
if(pravyIR<100&&levyIR<100) má otočit, dokud
```

## Modul HC-06

HC-06 je bluetooth modul určený v mém případě pro komunikaci mezi robotem a mobilním zařízením. Toto z něj činí pro mě nejdůležitější část robota, protože díky němu můžu robotovi dávat příkazy.

U tohoto modulu používám stejné příkazy jako u sériové komunikace, takže mi stačí pomocí zapsání příkazů „Serial.begin“ zahájit komunikaci a následně používat „Serial.print“. Díky tomuto modulu tedy odesílám do mobilu veškeré informace o teplotě, vlhkosti, kvalitě vzduchu a následném varování před případným znečištěním ovzduší nebo překážkou, kterou robot pomocí ultrazvukového senzoru detekuje před sebou.

Pro následnou komunikaci používám mobilní aplikaci Bluetooth terminal, kterou jsem stáhl z Google Play od uživatele pod jménem „Qwerty“.

## Napájení robota

Pro napájení robota momentálně používám čtyři baterie AA Panasonic Ni-MH 2700 mAh, 1,2V a dvě baterie Panasonic 3400 mAh, 3,7V. Na baterie mám 2 boxy, jeden box je pro první čtyři zmíněné baterie a tento box je zapojen přímo do desky Arduino Uno, ze které jsou pak napájeny ostatní senzory. Druhý box pro baterie Panasonic 3400 mAh je zapojen do desky pro řízení motorů a napájím tak z ní samostatně všechny čtyři motory.

## Program

Program, který jsem vytvořil je rozdělen na tři části. Každá část je pomocí funkce „while“ uzavřena do smyčky, která se bude neustále opakovat, dokud nepošlu přes aplikaci v mobilu pomocí bluetooth příkaz, v mém případě číslo, tak smyčka se neukončí.

Nejprve ale mám do programu přidané knihovny, které k některým sensorům potřebuji a zejména definované piny a proměnné, do kterých pak zapisují hodnoty ze senzoru nebo k jednotlivým smyčkám, které přepisují z mobilu a přepínám tak mezi těmi třemi částmi programu.

První část programu je založená pouze na měření, robot se tedy nepohybuje a pouze měří hodnoty pomocí sensorů. Tato část programu má číslo 1. Takže pokud toto číslo pošleme přes mobil robotovi, tak se měření automaticky zapne a začne odesílat hodnoty. Pokud číslo 1 přepíšeme, tak robot přestane měřit a odesílat data.

```
//zmena programu
  if(Serial.available()>0)
  {
    zmena=Serial.parseInt();
    m=2;
  }
//měření
  float ppm = sensorMQ.getPPM();
  Serial.print("Kvalita vzduchu: ");
  Serial.println(ppm);
  delay(500);
  float teplota = mojeDHT.readTemperature();
  Serial.print("Teplota: ");
  Serial.println(teplota);
  delay(500);
  float vlhkost = mojeDHT.readHumidity();
  Serial.print("Vlhkost: ");
  Serial.print(vlhkost);
  Serial.println("% ");
  delay(500);
```

Druhá část programu je jízda podle ultrazvukového senzoru, která je zařazena pod číslem 2. Během jízdy robot měří pouze vzdálenost a kvalitu ovzduší, na kterou upozorňuje LED diodou, ale žádná podrobná data nevypisuje. V případě překážky, které se musí robot vyhnout, je odeslána pouze varovná zpráva „POZOR!“. Celá jízda je pak založená na neustálém měření vzdálenosti a přepisováním proměnné „i“. Podle její hodnoty se robot rozhoduje, co má udělat.

Třetí a taky poslední část je sledování čáry. Tato část je zařazena pod číslem 3. Robot má podobně jako u druhé části proměnnou, která se mění podle aktuální hodnoty na prostředním IR senzoru a přepíná mezi hledáním čáry a jízdě po čáře. Rozdíl je ale v tom, že robot během jízdy po čáře neměří žádné další hodnoty kromě hodnot na IR senzorech. Po přepsání čísla 3 se program automaticky ukončí.

## Závěr

Z mého osobního hlediska si myslím, že se mi celé zadání podařilo dokončit a robot funguje, tak jak má. Potěšilo mě, že jsem měl možnost vymyšlení vlastní práce, a tak pracovat na něčem co mě baví. Chvilíma mě i překvapovalo proč něco nefunguje nebo to dělá něco jiného, ale pokaždé když se mi podařilo přijít na příčinu, tak mě to opět motivovalo



pokračovat dále v práci. Během programování a sestavování robota jsem se naučil pár dalších funkcí a příkazů ohledně Arduina, které můžu dále využívat jak ve škole, tak třeba i v budoucí práci. Závěrem bych chtěl poděkovat panu Ing. Stanislavu Moravcovi za jeho vedení mé maturitní práce a odborné rady a návrhy.

#### Zdroje

<http://www.instructables.com/id/How-to-use-the-L298-Motor-Driver-Module-Arduino-Tu/>

<https://learn.adafruit.com/adafruit-arduino-lesson-13-dc-motors/arduino-code>

<https://www.allaboutcircuits.com/projects/control-a-motor-with-an-arduino/>

<https://roboindia.com/tutorials/arduino-ir-proximity-color-detection-sensor>

<https://arduino-shop.cz/arduino/1720-arduino-bluetooth-modul-hc-06-rs232-1501765689.html>

<http://navody.arduino-shop.cz/arduino-projekty/servo-motor.html>

<http://navody.arduino-shop.cz/navody-k-produktum/meric-vzdalenosti-ultrazvukovy.html>

<http://navody.arduino-shop.cz/navody-k-produktum/teplotni-senzor-dht11.html>

<http://navody.arduino-shop.cz/navody-k-produktum/senzor-plynu-mq-135.html>