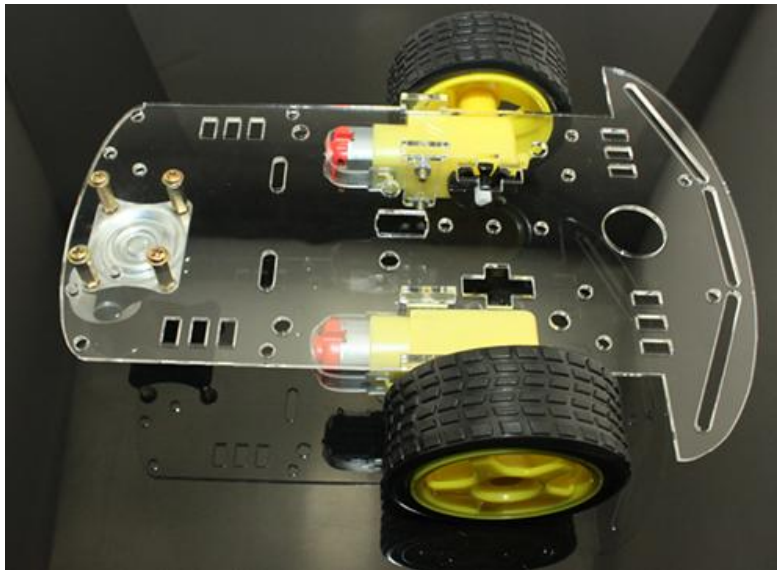


PA1 Semestrální práce - Vytvoření samohybného vozidla

Kontext

Své znalosti nabité v předmětu PA1 si otestujete realizací semestrální práce. Cílem semestrální práce je postavit funkční dvoukolové samohybné vozidlo řízené mikrokontrolérem Arduino UNO.



Zadání

Zadáním semestrální práce je ze zapůjčených součástek (*nebo ze součástek vlastních schválených cvičím*) sestavit funkčního robota.

Semestrální práce se zpracovává ve dvojci. Hodnocena je jako jeden celek a oba studenti dostanou stejné bodové ohodnocení (*které je bodovým ohodnocením práce jako celku*).

Robot bude dvou-kolové samohybné vozidlo řízené Arduinem UNO.

Robot musí mít následující součásti:

- Arduino UNO
- Podvozek se dvěma motory, všesměrovým předním kolem a držákem baterií
- H-můstek, který propojuje motory a Arduino
- Servo a na něm nasazený jeden ultrazvukový senzor, který robot používá ke snímání překážek
- Kabely, vodiče, apod.
- Jedno mikrotlačítko (pro přepínání režimů)
- Minimálně jedna LED dioda (pro signalizaci aktivního režimu)
- Robot je napájen 4xAA bateriemi (pro motory) a 1x 9V baterií (pro Arduino)

Pro úspěšné splnění semestrální práce musí vytvořený robot splňovat následující:

1. Robot musí být vybaven hardwarovým vypínačem (který bude vypínat motory)
2. Robot musí být řízen pouze mikrokontrolérem Arduino UNO (BEZ jakéhokoli dalšího řídicího či rozhodovacího zařízení, nebo externího ovládání)

3. Robot se musí být schopen pohybovat BEZ externího napájení, tedy při napájení pouze z palubních baterií (motory musí být napájeny 4xAA bateriemi, Arduino pak 1x9V baterií)
4. Robot musí být schopen jízdy vpřed, zatočení vlevo, zatočení vpravo a couvání (couvání je možné používat pouze omezeně - např. jako reakci na překážku - robot totiž nevidí vzad)
5. Robot musí být schopen pohybu vpřed minimálně třemi rychlostmi (např. pomalu, normálně, rychle)
6. Robot musí být vybaven pouze jediným ultrazvukovým senzorem, který bude nasazen na otočném servomotoru. Robot musí servomotor i ultrazvukový senzor využívat při svém pohybu (resp. pro zkoumání zda mu v případném, či aktuálním pohybu brání překážka), zda se k němu překážka blíží a má tudíž couvnout, apod.
7. Robot musí "zatáčet jako tank" tedy zpomalováním, nebo vypínáním, či přepínáním směru levého a pravého poháněného kola
8. Řídící systém robota musí tedy "najednou" ovládat motory pomocí H-můstku, otáčet servem a snímat vzdálenost pomocí ultrazvukového senzoru. Jelikož "najednou" se na Arduino UNO, které nepodporuje multi-tasking, dělá těžko, musí řídící systém robota správně implementovat pre-emptivní multitasking. Není nutné používat hardwarová ani softwarové přerušování (interrupts).
9. Algoritmus, který bude robot používat pro své řízení není (úmyslně) pevně stanoven a je na konkrétní dvojici studentů jaký přístup zvolí. Navigační systém robota nicméně musí splňovat následující body
10. Robot se musí být schopen vyhnout překážce před ním (nenajet do ní) když se pohybuje (překážkou rozumíme překážku jejíž detekce je v rámci omezených možností ultrazvukového senzoru - překážkou tedy nejsou věci jako nohy od stolu či židle, apod.)
11. Robot musí být schopen trochu couvnout a poté pokračovat v jízdě pokud ho k překážce postavím příliš blízko (a až poté zapnu), nebo pokud se nějakým způsobem dostane k překážce příliš blízko, že už se jí nemůže vyhnout zatočením a pokračováním v jízdě, nebo pokud se překážka přisune k němu
12. Robot se musí být schopen vyhnout překážce ve směru kam zatáčí
13. Robot musí být vybaven tlačítkem, pomocí kterého jej bude možné přepínat mezi režimem "**jedu a vyhýbám se překážkám**" a "**stojím, ale jsem ostražitý**"
 1. V režimu "jedu a vyhýbám se překážkám" robot jede a vyhýbá se překážkám, překážky se mohou i pomalu pohybovat (robot tedy i během jízdy musí být svým způsobem ostražitý)
 2. V režimu "stojím, ale jsem ostražitý" bude robot stát na místě, ale když před něj položíme překážku, kterou k němu budeme přisunovat, tak couvne, nebo se jí pokusí jiným způsobem vyhnout
 3. Robot musí být vybaven minimálně jednou LED diodou, která bude signalizovat jaký režim robota je zapnutý (jak to bude signalizovat je již na vás)
 4. Robot může (nemusí, toto je na vás) umět své dva režimy přepnout i sám na základě nějakých podmínek (například, když je zablokován ze předoleva i předoleva i předoleva, může se rozhodnout přepnout se z režimu jízdy do režimu stojím a jsem ostražitý). Pokud bude robot své režimy automaticky přepínat, je nutné aby to uměl oběma směry (v našem příkladu by se například po odblokování přepnul opět do režimu jízdy)
14. Robot může (ale nemusí) být vybaven LED diodami, které signalizují co právě provádí (toto vřele doporučujeme, robot se vám totiž pak bude zásadně lépe debugovat)
15. Kvůli omezením ohledně počtu senzorů a omezující fyzikální povaze senzorů není třeba, aby vyhýbání se překážkám bylo "dokonalé". Je v pořádku že se tedy robot nebude vyhýbat překážkám při couvání (neboť dozadu pouze s jedním ultrazvukovým senzorem, který má vpředu, nevidí), nebo bude vrážet do věcí jako jsou nohy od židle, či od stolu atd. Neměl by však vrážet do stěny, pokud k ní jede kolmo, apod. Pohyb robota bude testován v prostředí kde se nohy od židle (a další problematické překážky) vyskytovat nebudou.
16. Řídící systém robota může fungovat **ostrážitě** nebo **reaktivně** (kterou variantu zvolíte je na vás, kdo chce výzvu, může vytvořit systém, který bude podporovat obě a bude se mezi nimi sám přepínat ;))
 1. **ostrážitě** přístup znamená že se robot vždy rozhlédne a teprve potom vykoná nějakou akci (popojede, zatočí, apod.), pak se znovu rozhlédne, a vše se opakuje - tento přístup je snazší na realizaci, ale robot se pak chová "značně tupěji" (ale stále musí umět více věcí naráz (např. rozhlížet se a přitom měřit, nebo couvnout či se zcela zastavit při přibližující se překážce - takže bez pre-emptivního multitaskingu se neobejdete)
 2. **reaktivní** přístup znamená, že robot neustále vykonává nějakou akci a v reálném čase *reaguje* na okolní hrozby. Tento způsob je náročnější na realizaci, ale je značně zábavnější a je s ním více legrace. A robot se pak chová "chytřeji" (a rovněž se neobejde bez pre-emptivního multi-taskingu).
17. Řídící systém robota bude stavový automat. Kolik a jakých stavů však bude mít už je na vás. Stejně tak množství různých akcí, které bude váš robot podporovat (např. vpřed na 1/4, 1/2. 1/1, vzad, trochu vlevo, hodně vlevo, rozhlédni se, apod.) - je na vás jaké akce (= stavy) bude váš robot mít. Vřele doporučujeme si vše napřed namalovat na papír (třeba formou stavového diagramu) než to začnete programovat.

1. Způsob, jakým bude stavový automat implementován je na vás. V zadání je pouze řečeno, že musí být přítomen. A musí mít minimálně 3 různé stavy. Implementován může být formou několika IF podmínek, nebo pomocí knihovny FSM, nebo jiným způsobem.
 2. Je povoleno mít stavových automatů více, nebo klidně hierarchický stavový automat, apod.
 3. Řídící systém může používat i další součásti než jen pouze stavový automat. Stavový automat alespoň se třemi stavy musí být přítomen a používán.
18. Robot nesmí používat žádné jiné senzory kromě jednoho ultrazvukového senzoru na servu - žádné enkodéry, další senzory, IMU jednotky a další.

Zdrojový kód řídicího systému robota musí splňovat následující:

1. PINová konfigurace (aneb čísla Arduino PINů, na kterých jsou připojeny jednotlivé součástky) musí být ve zdrojovém kódu uložena jako #define konstanty v samostatném .h souboru
2. Názvy všech funkcí a proměnných pojmenovávajíte smysluplně (např. ne varA, nebo pinA, apod.), anglicky a dle Arduino konvencí (camelCase(), myCoolVariable), názvy konstant MY_CONSTANT
3. Kód musí být pěkně odsazen a zformátován
4. Nevytvářejte špagetové funkce
5. Ve funkcích setup() a loop() nemějte přímo žádný kód, kromě volání dalších funkcí
6. Neváhejte program rozdělit do jednoho .ino a několika .h souborů

Tipy

- Pozor na setrvačnost! Když robot přestane zrychlovat, tak to neznamená, že bude brzdít...
- Projděte si všechny zdroje dole!

FAQ

Je možné použít vlastního robota?

Pokud si sami stavíte robota ze svých vlastních součástek, je možné jej využít po dohodě s vyučujícím pokud robot splňuje všechny body výše.

Kde je možné součástky na robota koupit?

Pokud si chcete robota po skončení (ideálně po úspěšném absolvování) předmětu nechat, je nutné si ho postavit nikoli ze školních, nýbrž z vlastních součástek. Níže uvádíme jejich seznam společně s odkazy kde si je můžete koupit.

- Arduino Starter Kit - <https://www.alza.cz/arduino-starter-kit-d569011.htm?o=1>
- Podvozek s koly a motory - <https://www.aliexpress.com/item/New-Motor-Smart-Robot-Car-Chassis-Kit-Speed-Encoder-Battery-Box-2WD-For-Arduino-Free-Shipping/32568348378.html>
- H-můstek - <https://www.gme.cz/h-mustek-modul-l9110s> nebo <http://arduino-shop.cz/arduino/877-arduino-h-mustek-pro-krokovy-motor-l298n-dual-h-most-dc-1420490399.html>
- Ultrazvukový senzor - <https://www.aliexpress.com/item/HC-SR04-Ultrasonic-Module-Wave-Sensor-Ranging-Detector-Distance-Module-for-Arduino-Free-Shipping-Dropshipping/1738728686.html>
- 1x kabel k 9V baterii (pro napájení Arduina) - <http://arduino-shop.cz/arduino/1029-napajeci-kabel-k-9v-baterii-clip-jack.html>
- 4x AA baterie
- 1x 9V baterie

Řešení

Odevzdání semestrální práce má dvě části

1. Odevzdání .zip souboru se adresářem projektu (do popisu úkolu uveďte jméno svého robota - např. "Můj robot se jmenuje Lokomotiva")
2. Obhajoba a ukázka funkčního robota na posledním semináři

Zdrojový kód robota je NUTNÉ odevzdat PŘED posledním seminářem, kde bude robot předveden vyučujícím.

Semestrální odešlete k ohodnocení tlačítkem níže. Dodržujte pokyny pro vyplnění požadavku, například formát přílohy či název požadavku, pokud vyučující takové pokyny uvedl v zadání domácího úkolu.

Odeslat řešení domácího úk...

Na pozdě odevzdaná řešení nebude brán zřetel. O vteřinu pozdě odevzdané řešení je pozdě odevzdané řešení. Rozhodující je čas zaznamenaný systémem. Svá řešení můžete odevzdávat i dříve. **Odevzdávat dříve je doporučeno.**

Jak odevzdat domácí úkol

Vyplňte "Název" - někteří vyučující uvádějí požadovaný název v zadání domácího úkolu. (*Jméno odesílatele a předmět jsou doplněny automaticky školním systémem.*)

Pokud chcete do požadavku **vložit přílohy**, klikněte na tlačítko "Vložit přílohu". Zobrazí se dialogové okno, kde zadáte cestu k příslušnému souboru. Po kliknutí na tlačítko "OK" se zobrazí seznam příloh, následně zavřete dialogové okno.

Domácí úkol odevzdáte kliknutím na tlačítko "Odeslat". V záhlaví se zobrazí lišta s informací "Požadavek byl úspěšně odeslán ke zpracování." *Založené ale neodeslané požadavky se hlásí ve vašem úkolovníku aktivitou "Neodesláno!" O akceptaci (či zamítnutí) vyučujícím Vás bude systém informovat aktivitou a požadavek s domácím úkolem se následně odkáže na Vaši kartu studenta v předmětu.*

Podrobná nápověda k odevzdání domácího úkolu  [Jak odevzdat domácí úkol](#)

Zdroje

Ke zpracování semestrální práce použijte informace z jednotlivých přednášek a následující zdroje:

- Jak na Arduino UNO
 - <https://www.arduino.cc/en/Main/ArduinoBoardUno>
 - <http://arduino.cz/programujeme-arduino/>
 - <https://www.arduino.cc/en/Guide/HomePage>
- Jak na sériovou linku (debugging)
 - <https://www.arduino.cc/en/Reference/Serial>
 - <https://www.arduino.cc/en/Serial/Print>
 - <https://www.arduino.cc/en/Serial/Available>
 - <https://www.arduino.cc/en/Tutorial/DigitalReadSerial>
 - <https://www.arduino.cc/en/Tutorial/ReadASCIIString>
- Jak na tlačítka
 - <https://www.arduino.cc/en/Tutorial/Button>
 - <https://www.arduino.cc/en/Tutorial/Debounce>
 - <https://www.arduino.cc/en/Tutorial/StateChangeDetection?from=Tutorial.ButtonStateChange>
- Jak na LED Diody
 - <https://www.arduino.cc/en/tutorial/blink>
 - <https://www.arduino.cc/en/Tutorial/BlinkWithoutDelay>
 - <https://www.arduino.cc/en/Tutorial/Fade>

- Jak na servo
 - <https://www.arduino.cc/en/Tutorial/Sweep>
 - <https://www.arduino.cc/en/Reference/Servo>
- Jak na ultrazvukový senzor
 - 3D vytisknutelný držák ultrazvukového senzoru od jednoho z vašich kolegů: <http://www.thingiverse.com/thing:1895405>
 - <http://playground.arduino.cc/Main/UltrasonicSensor>
 - <http://www.instructables.com/id/Simple-Arduino-and-HC-SR04-Example/>
 - <https://bitbucket.org/teckel12/arduino-new-ping/wiki/Home>
 - <http://playground.arduino.cc/Code/NewPing>
 - <https://www.youtube.com/watch?v=aLkkAsrSibo>
- Jak na H-můstek, motory a PWM
 - <http://www.bajdi.com/l9110-h-bridge-module/>
 - <https://www.youtube.com/watch?v=SroGuuMZ9tY>
 - <http://www.robotic-studio.net/2015/08/tutorial-learn-to-use-dc-motor-driver.html#.WAOETJN96Rs>
 - [https://www.bananarobotics.com/shop/How-to-use-the-HG7881-\(L9110\)-Dual-Channel-Motor-Driver-Module](https://www.bananarobotics.com/shop/How-to-use-the-HG7881-(L9110)-Dual-Channel-Motor-Driver-Module)
 - <http://robodoupe.cz/2014/driver-stejnoserneho-motoru-l9110/>
 - <http://hardwarefun.com/tutorials/creating-robots-using-arduino-h-bridge>
 - <http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/>
 - https://www.reddit.com/r/arduino/comments/4rruqy/robot_with_l298n_motor_driver_or_l9110_hbridge/
- Jak na multitasking
 - https://en.wikipedia.org/wiki/Computer_multitasking
 - <https://www.arduino.cc/en/Tutorial/BlinkWithoutDelay>
 - <http://www.gammon.com.au/blink>
 - <http://arduino.stackexchange.com/a/16140>
 - <http://playground.arduino.cc/Code/Scheduler>
 - <http://playground.arduino.cc/Code/ArduOS#Use>
 - <http://playground.arduino.cc/Code/OS48>
 - <https://www.chrismoos.com/2012/12/05/avr-os-multitasking-on-arduino>
 - <http://www.rtos48.com/>
 - <https://github.com/ferdinandkeil/humidor-steuerung/blob/master/task.c>
- Jak na State Machine (stavový stroj) na Arduino (FSM)
 - <https://www.youtube.com/watch?v=hJIST1cEf6A>
 - <https://www.youtube.com/watch?v=6oe1Tmg9rjM>
 - https://cs.wikipedia.org/wiki/Kone%C4%8Dn%C3%BD_automat
 - https://en.wikipedia.org/wiki/Finite-state_machine
 - <http://playground.arduino.cc/Code/FiniteStateMachine>
 - <http://www.humblecoder.com/arduino-finite-state-machine-library/>
 - <https://github.com/jonblack/arduino-fsm>
 - <http://www.humblecoder.com/arduino-multitasking-using-finite-state-machines/>
 - <http://www.mathertel.de/Arduino/FiniteStateMachine.aspx>

Rozšíření

- Jak vytvořit vlastní Arduino Knihovnu
 - <https://www.arduino.cc/en/Hacking/LibraryTutorial>
- Jak na I2C

- https://www.youtube.com/watch?v=qeJN_80CiMU
- <https://www.arduino.cc/en/Reference/Wire>
- <http://arduino.cz/propojujeme-arduino-s-jinym-i-zarizenimi/>
- <https://www.arduino.cc/en/Tutorial/MasterWriter>
- <https://www.arduino.cc/en/Tutorial/MasterReader>
- Rozšíření
 - <http://tronixstuff.com/2010/10/20/tutorial-arduino-and-the-i2c-bus/>
 - <http://arduino8.webnode.cz/news/lekce-10-arduino-a-i2c-lcd/>
 - <http://www.itnetwork.cz/hardware-pc/arduino/arduino-a-i2c-sbornice>
 - Propojení Arduina s RaspberryPi - <https://oscarliang.com/raspberry-pi-arduino-connected-i2c/>
 - Propojení Arduina s Lego MindStorms - <http://www.dexterindustries.com/howto/connect-the-arduino-and-the-lego-mindstorms-together/>
- OOP V Arduinu
 - <https://github.com/mikaelpatel/Cosa>
 - <http://cosa-arduino.blogspot.com/>
- Další zdroje:
 - <http://arduino.cz/>
 - <http://arduino.cz/category/novinky/tutorialy/arduino-roboti/>
 - <http://arduino.cz/category/novinky/tutorialy/arduino-se-zbyskem-vodou/>
 - <http://arduino.cz/category/novinky/tutorialy/arduino-v-prikladech/>
 - <http://arduino.cz/category/novinky/tutorialy/arduino-s-massimem-banzim/>
 - <http://arduino.cz/category/novinky/tutorialy/arduino-projekty/>