Manažment projektov

Robot pre súťaž Istrobot

Bc. Eduard Mráz

Bc. Adam Trizuljak

Bc. Dominik Varga

Bc. Ondrej Zsíros

Bc. Michal Nagy

Bc. Michal Hubík

1 Obsah

2 Úvod

Cieľom tohto projektu je navrhnúť, vytvoriť a spojazdniť model robota pre súťaž Istrobot. Súťažnou kategóriou bude Stopár. Robot, ktorý sa zúčastní tejto kategórie by mal byť schopný autonómneho pohybu po určenej dráhe a v časovom limite prísť do cieľa. Smer a trasa je daná tmavým pruhom na bielom podklade, pričom na dráhe sú umiestnené rozličné prekážky, napr. tehla, záclona a mostík. Robot je taktiež limitovaný šírkou a výškou, ktoré nesmú presiahnuť 20 centimetrov.

Výsledkom projektu bude model robota, ktorý bude schopný pohybu po zvolenej dráhe. Bude obsahovať nízko úrovňové a vysoko úrovňové riadenie. Bude obsahovať všetky komponenty potrebné pre zdolanie všetkých prekážok súťaže v danej kategórií. Detegovanie čiary bude zabezpečené pomocou jednoduchého vizuálneho systému.

Nakoľko je súťaž až v nasledujúcom semestri, robot na ňu nemusí byť plne pripravený v čase odovzdávania tohto projektu. A teda ďalšou prácou pre splnenie podmienok súťaže Istrobot, bude tvorba ďalších algoritmov na zdolávanie jednotlivých prekážok.

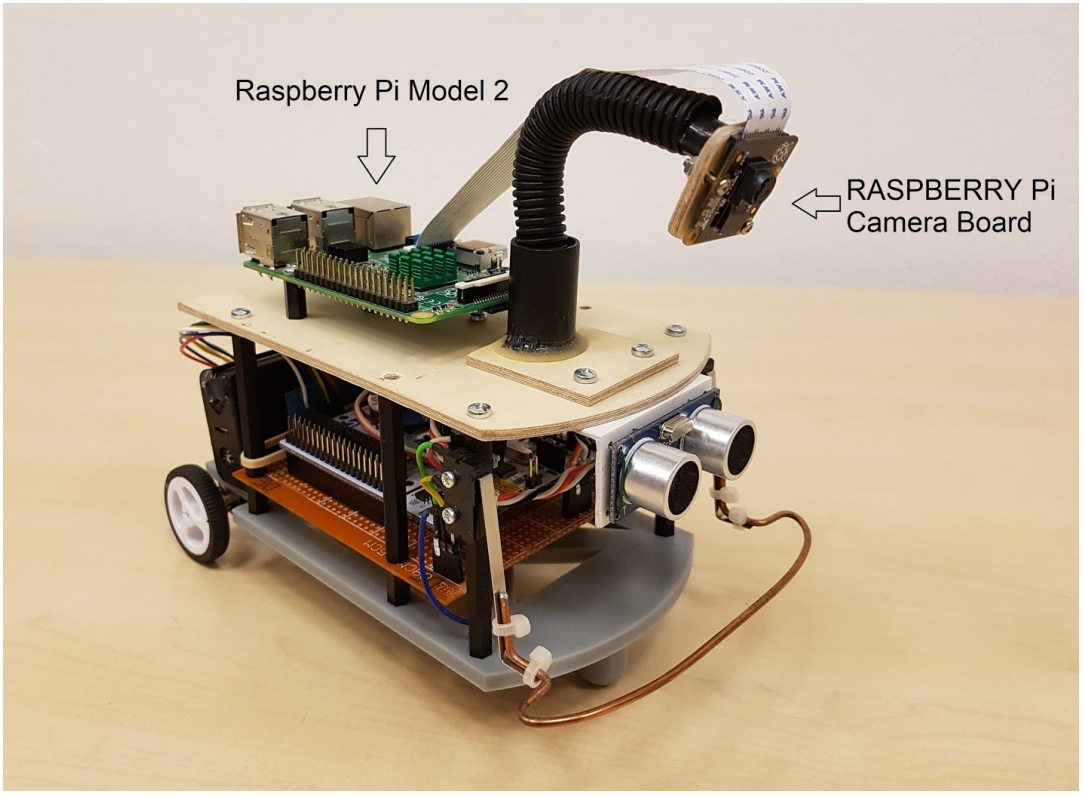
3 Opis robota

Pri návrhu koncepcie sme robota rozdelili na dve základné časti, ktoré medzi sebou komunikujú prostredníctvom sériovej komunikácie, a to vysokoúrovňová časť a nízkoúrovňová časť. Obe tieto časti sú umiestnené na samostatnej doske, kde majú k sebe pripojené senzory potrebné pre súťaž Istrobot.

3.1 Vysokoúrovňová časť

Ako už bolo spomenuté robot bude súťažiť v kategórií Stopár, kde musí sledovať čiernu čiaru. Na detekovanie čiary musí byť robot vybavený senzorom, ktorý bude schopný čiaru rozlíšiť oproti bielej podložke. My sme si zvolili kameru RASPBERRY Pi Camera Board, ktorá je doplnená o osvetlenie pre prekážku tunel. Kamera je pripojená cez rozhranie CSI k výpočtovej jednotke Raspberry Pi Model 2.

Okrem výpočtovej jednotky a kamery je vo vysokoúrovňovej časti zabudovaný aj napäťový regulátor, ktorého vstupom je napätie akumulátora a výstupom je stabilizované napätie 5V, ktoré slúži na napájanie RPi a mikroprocesora v nízkoúrovňovej časti. Regulátor je umiestnený zo spodnej strany dosky. Pre lepšiu manipuláciu s robotom sme do vysokoúrovňovej časti umiestnili aj vypínač, ktorý odpája napájanie motorov.

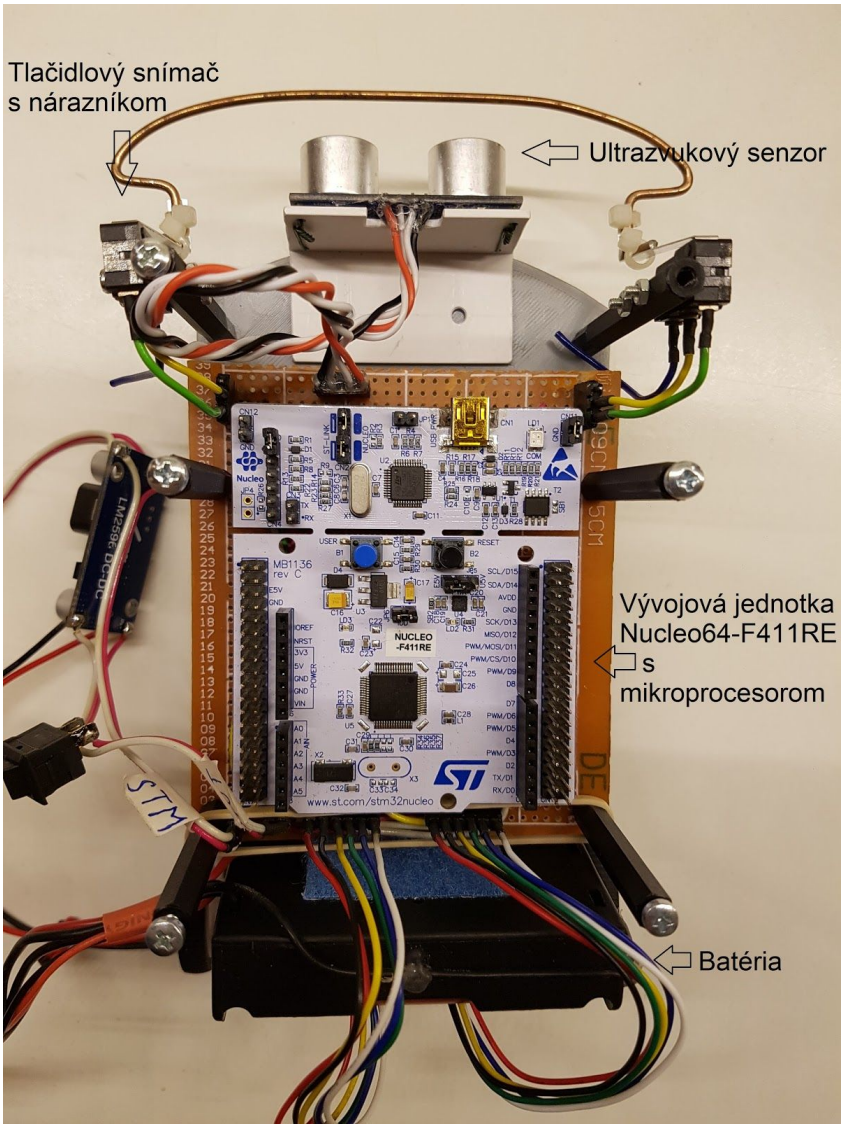


3.2 Nízkoúrovňová časť

Sériovou komunikáciou vysokoúrovňová časť posiela signály do nízkoúrovňovej časti konkrétne do vývojovej jednotky Nucleo64-F411RE, ktoré súčasťou je mikroprocesor

STMicroelectronics STM32F411. V mikroprocesore signály poslané vysokoúrovňovou časťou spracujú a vyhodnotia sa potrebné uhlové rýchlosti otáčania oboch kolies. Signál určujúci veľkosť uhlovej rýchlosti kolies sa pošle do budiča motorov, ktorý reguláciou napájacieho napätia motorov určí ich výslednú rýchlosť otáčania. Regulácia napätia je uskutočňovaná pulzne šírkovou moduláciou.

Keďže jedna z prekážok na súťaži je tehla (tehla položená cez čiernu čiaru) robot musí vedieť obísť túto prekážku. Na to nám slúžia dva typy senzorov - ultrazvukový snímač a tlačidlový spínač na prednom nárazníku. Keď sa robot bude približovať k tehle kamera nebude môcť detekovať čiaru. Robot sa bude pohybovať ďalej ale čím bližšie bude k tehle tým pomalšie pôjde. Toto spomaľovanie spôsobuje ultrazvukový snímač, ktorý spomaľuje robota až kým nepríde k tehle. Akonáhle robot narazí na tehlu zopnú sa tlačidlové spínače na nárazníku a robot začne prekážku obchádzať. Po obídení prekážky kamera znova začne detekovať čiaru a robot pokračuje ako doposiaľ. Na rovnakom princípe sa robot bude pohybovať aj pri prekážke typu záclona, kde robot bude spomaľovať ako sa bude približovať k záclone. Keďže záclona nie je pevná prekážka, tlačidlové snímače sa nezopnú a robot bude pomaly pokračovať cez záclonu až kým sa robot znovu nezrýchli za prekážkou.



4 Konštrukcia

Po koncepčnom návrhu jednotlivých funkcionalít, potrebných pre úspešné prejdenie celého súťažného kola súťaže Istrobot v kategórii Stopár, vrátane všetkých špecifikovaných prekážok, bolo potrebné navrhnúť fyzickú konštrukciu celého robota.

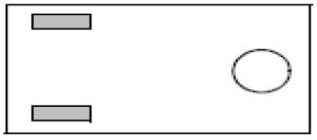
Najzákladnejším parametrom konštrukcie, ktorý bolo potrebné stanoviť ako prvý, boli vonkajšie rozmery, čiže dĺžka, šírka a výška. Azda najviac limitujúcim faktorom na súťažnej dráhe je prekážka “Tunel”, ktorá má vstupný otvor rozmerov 20x20cm. Pre úspešné prejdenie tejto prekážky je potrebné aby robot vošiel do tunela - jeho šírka a výška musia byť menšie ako 20cm, a zároveň dokázal v neosvetlenom priestore sledovať čiaru. My sme sa rozhodli zvoliť celkovú šírku 12cm. Táto šírka plne postačuje na dostatok priestoru pre všetky komponenty a zároveň poskytuje dostatočnú rezervu pre vstup do tunela bez prípadnej kolízie z dôvodu nepresnosti riadenia. Dĺžku robota sme po rýchlom prototypovaní - pokusné rozmiestňovanie jednotlivých komponentov na čo najmenšej ploche, zvolili 25 centimetrov. Dĺžka robota nie je obmedzená žiadnou prekážkou, no nemala by byť príliš veľká z dôvodu možných problémov napríklad pri obchádzaní prekážky “Tehla” alebo pri prejazde zatáčok s malým polomerom. Na výslednú dĺžku mali vplyv hlavne rozmery riadiacich dosiek, akumulátorov, motorov a nárazníka.

Po stanovení rozmerov bolo potrebné zvoliť typ podvozku robota. Najčastejšie sa v tejto súťaži vyskytujú dvojkolesové alebo štvorkolesové diferenciálne podvozky. Po uvážení faktorov ako napríklad:

- Celková hmotnosť
- Rozmery a potrebná kapacita akumulátorov
- Náročnosť konštrukcie
- Cena komponentov
- Zložitosť riadenia

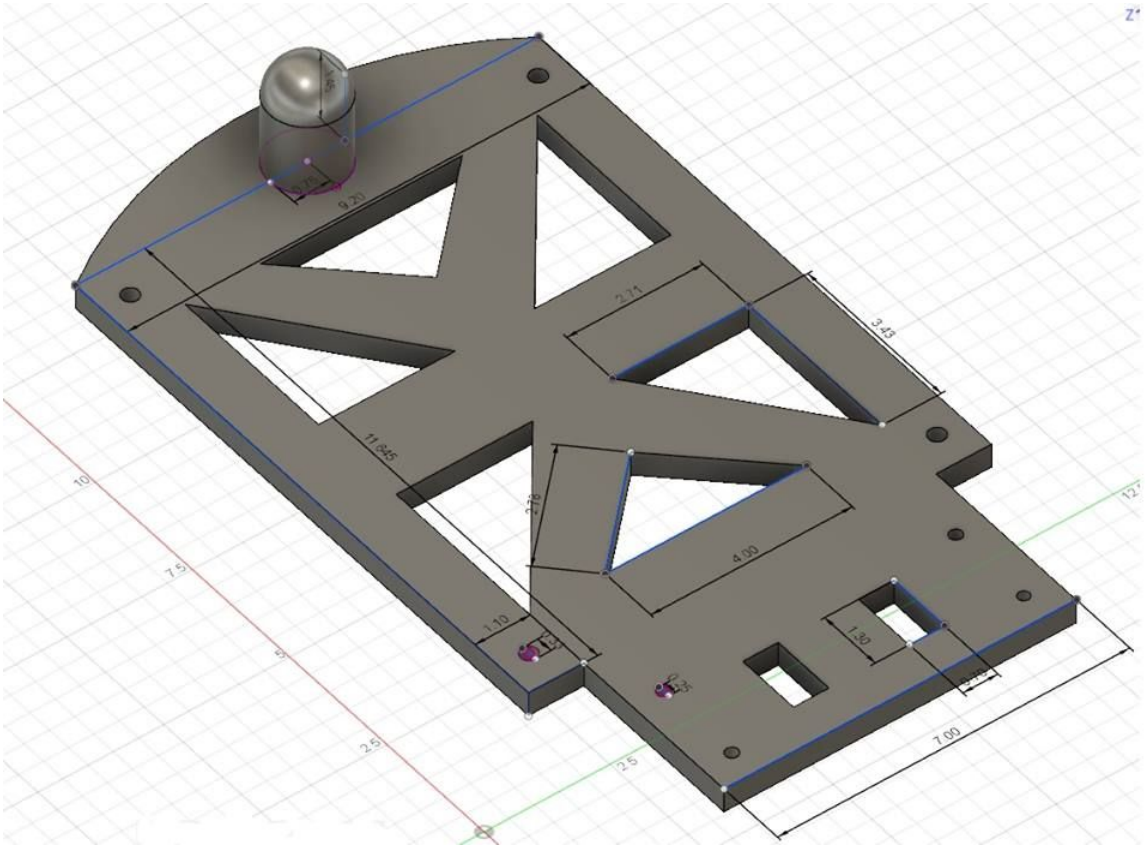
sme sa rozhodli pre dvojkolesový diferenciálny podvozok v jedným oporným kolesom. Konfigurácia kolies je nasledovná:

– dve neotočné poháňané kolesá a jedno všesmerové nepoháňané koleso



Po zvolení koncepcie podvozku, bolo potrebné navrhnúť jeho konštrukciu. Po zvážení rôznych konštrukčných požiadaviek sme sa rozhodli vytvoriť základnú podvozkovú dosku formou návrhu 3D modelu a následnej 3D tlače z plastu typu PLA. Voľbou tohto postupu sme zabezpečili nízku hmotnosť, dostatočnú pevnosť a tiež presnosť upevnení motorov. Presnosť bola veľmi dôležitým faktorom, nakoľko sa oba motory (obe osi kolies) musia z dôvodu požadovanej trajektórie podvozku nachádzať na jednej osi. To zabezpečí možnosť

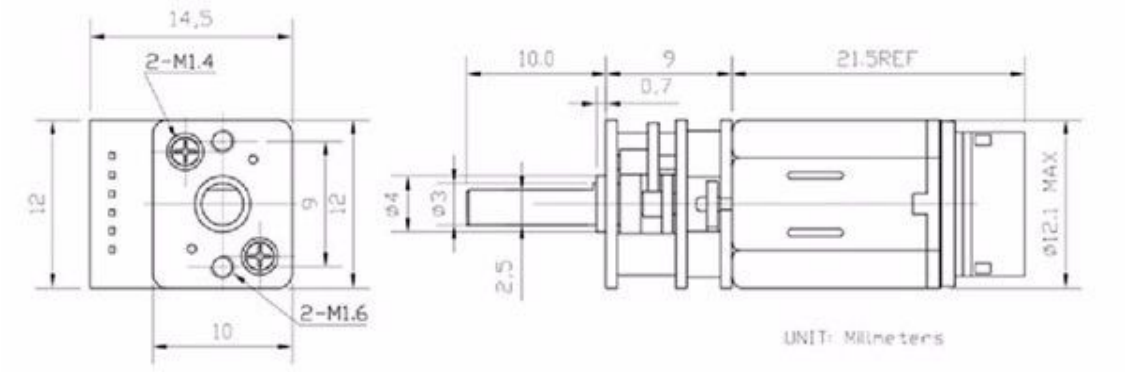
priamočiareho pohybu ako aj otáčania na mieste. Okrem dier potrebných k upevneniu motorov obsahuje táto základná doska aj diery potrebné pre uchytenie dištančných stĺpikov. Na tie sa následne montovala doska nízkoúrovňového a vysokoúrovňového riadenia. Zároveň dva predné stĺpiky slúžia aj na uchytenie mikrospínačov nárazníka v požadovanej výške 75 mm od zeme. Na uchytenie vypínačov poslúžili krížové skrutky M2 x 15 mm. K základnej doske je nad osou motorov pripevnený pomocou suchého zipsu pohonný akumulátor tvorený dvoma Li-Ion článkami typu 18650 akumulátorov a pre istotu je jeho pozícia fixovaná pomocou gumičky o zadné dištančné stĺpiky.



3D CAD model podvozku robota

Po vytlačení a opracovaní hrán základnej dosky sme ako prvé pripevnili motory - konkrétne sme zvolili jednosmerné motory s nasledovnými parametrami:

- Rýchlosť: 90 otáčok/min.
- Napájacie napätie: 6V
- Prevodový pomer prevodovky: 100:1
- Krútiaci moment (kg.cm): 0.7
- Elektrický prúd (mA): ≤170
- Počet impulzov na jednu otáčku kolesa: 2800



Po primontovaní kolies s priemerom 35mm na hriadele prevodovky bol v podstate dokončený celý podvozok. Následne sme na základnú dosku pripevnili prvé dištančné stĺpiky s dĺžkou 11 mm na ktoré bola upevnená univerzálna doska plošných spojov - DPS, s rozmermi 10 x 9 centimetrov. Táto doska - nazývaná aj ako “doska nízkoúrovňového riadenia”, obsahuje:

- Päticu pre mikrokontrolér
- Mikrokontrolér STM Nucleo F411RE
- 2x dolnopriepustný filter pre spínače nárazníka
- Napájacie a komunikačné konektory
- Driverovú dosku zabezpečujúcu chod motorov
- Kabeláž periférii k nízkoúrovňovému mikrokontroléru

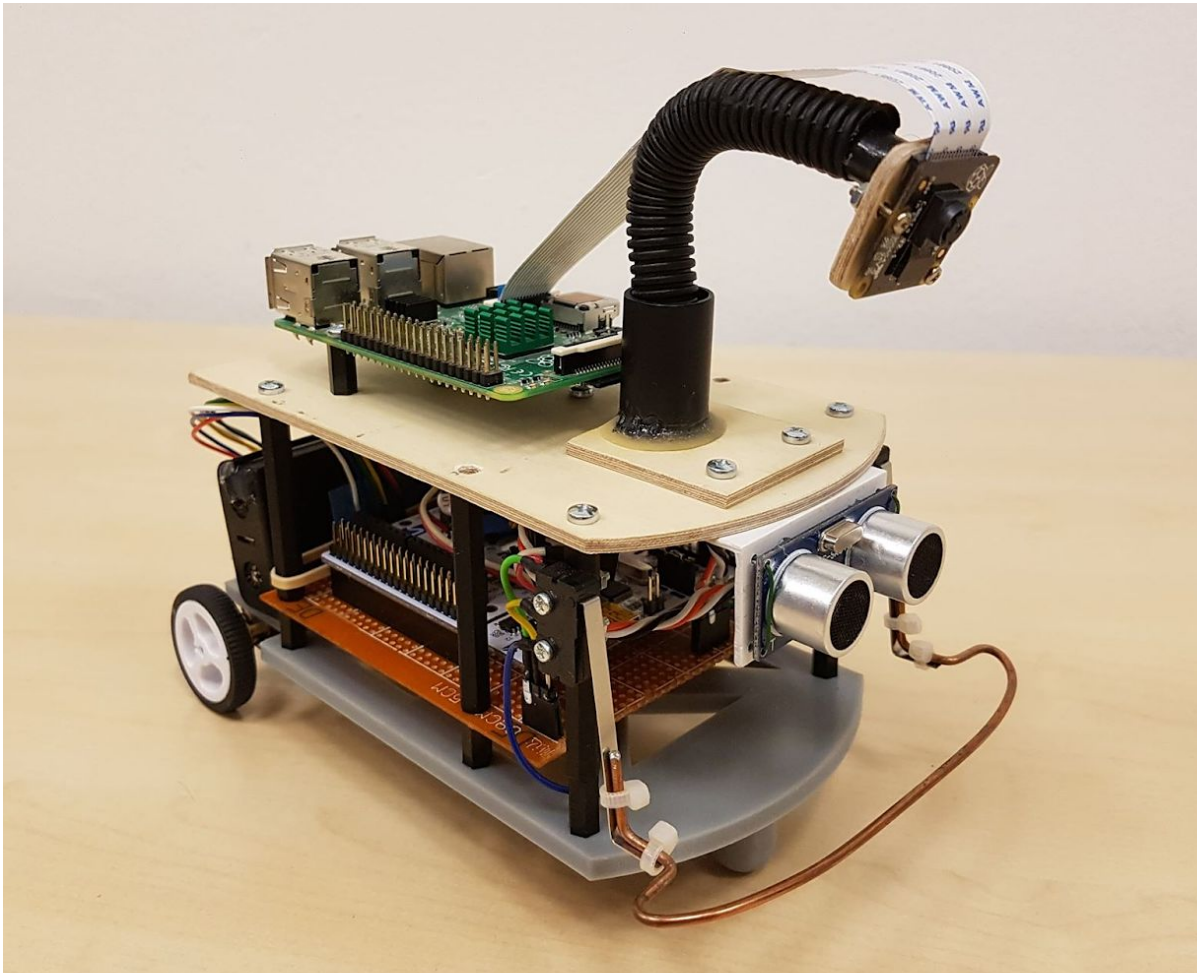
Keďže by sa na jedno podlažie robota všetky potrebné súčiastky nezmestili, bolo potrebné navrhnúť a vyrobiť aj druhé podlažie, kam by sa umiestnili zvyšné súčiastky.

Ako nosnú konštrukciu druhého podlažia robota sme opäť zvolili plastové dištančné stĺpiky dĺžky 50mm, na ktoré sme umiestnili z leteckej preglejky hrúbky 3mm vyrezávanú dosku v tvare základnej dosky podvozku. Dosku druhého podlažia nazývame aj doska vysokoúrovňového riadenia. Do nej sme následne vŕtaním vyrobili diery potrebné pre uchytenie následných komponentov:

- Napäťový regulátor
- Mikropočítač Raspberry Pi 2
- Kolískový vypínač napájania motorov
- Držiak vizuálneho systému + vizuálny systém

- Ultrazvukový senzor

Použitý napäťový regulátor LM2596 DC-DC slúži na konvertovanie vstupného jednosmerného napätia z akumulátora na stabilizované napätie 5V potrebné na napájanie oboch dosiek riadenia. Jeho uchytenie je realizované dištančnými stĺpikmi. Kolískový vypínač umiestnený vedľa mikropočítača Raspberry Pi z hornej strany dosky slúži na prerušenie napájania driverovej dosky - robot sa prestane hýbať. Držiak kamery je umiestnený v prednej časti hornej dosky, a poskytuje 6 stupňov voľnosti pri nastavovaní polohy a orientácie kamery. Kamera je k držiaku pripevnená dvomi krížovými skrutkami M2x20 mm Na záver je horná doska upevnená k dištančným stĺpikom krížovými skrutkami M3 x 20mm.



5 Nízkoúrovňové riadenie

Nízkoúrovňové riadenie prijíma príkazy z vysokoúrovňového riadenia prebiehajúceho v Raspberry Pi a zabezpečuje predovšetkým riadenie rýchlosti motorov. Taktiež počíta aktuálnu odometriu, spracováva údaje zo senzorov a odosiela ich do Raspberry Pi, kde prebieha ich vyhodnotenie.

Riadenie je realizované mikroprocesorom STMicroelectronics STM32F411 (ďalej len ***STM***), čo je 32-bitový ARM Cortex-M4 procesor s frekvenciou 100MHz, s 512KB programovej pamäte a 128KB RAM. Tento procesor je súčasťou vývojovej dosky Nucleo64-F411RE, ktorá obsahuje aj programátor/debugger s USB rozhraním pre jednoduché a pohodlné programovanie.

Po spracovaní senzorov sa výsledné údaje spolu s údajmi odometrie odosielajú po sériovej linke do Raspberry Pi.

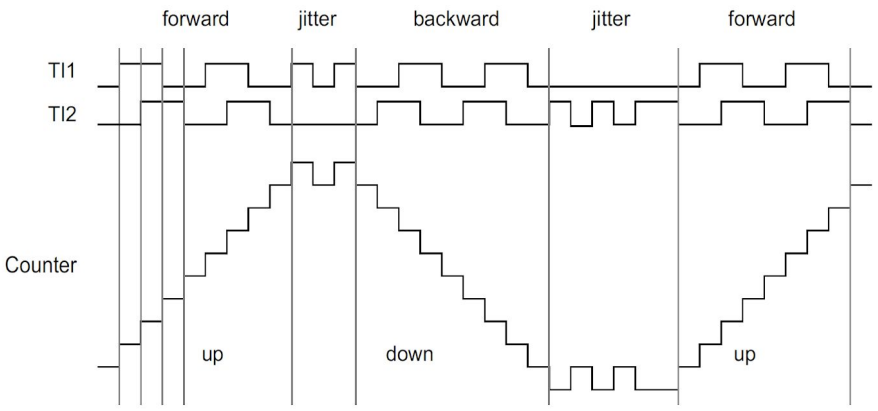
5.1 Spracovanie senzorov

Mikroprocesor realizuje spracovanie dát z nasledovných senzorov:

- 2x inkrementálny enkóder snímajúci otáčky motorov
- 2x tlačidlový spínač predného nárazníku
- Ultrazvukový senzor vzdialenosti HC-SR04

Tlačidlá nárazníku sú pripojené na dva digitálne vstupy. Program v hlavnej riadiacej slučke periodicky číta ich logické hodnoty (0/1).

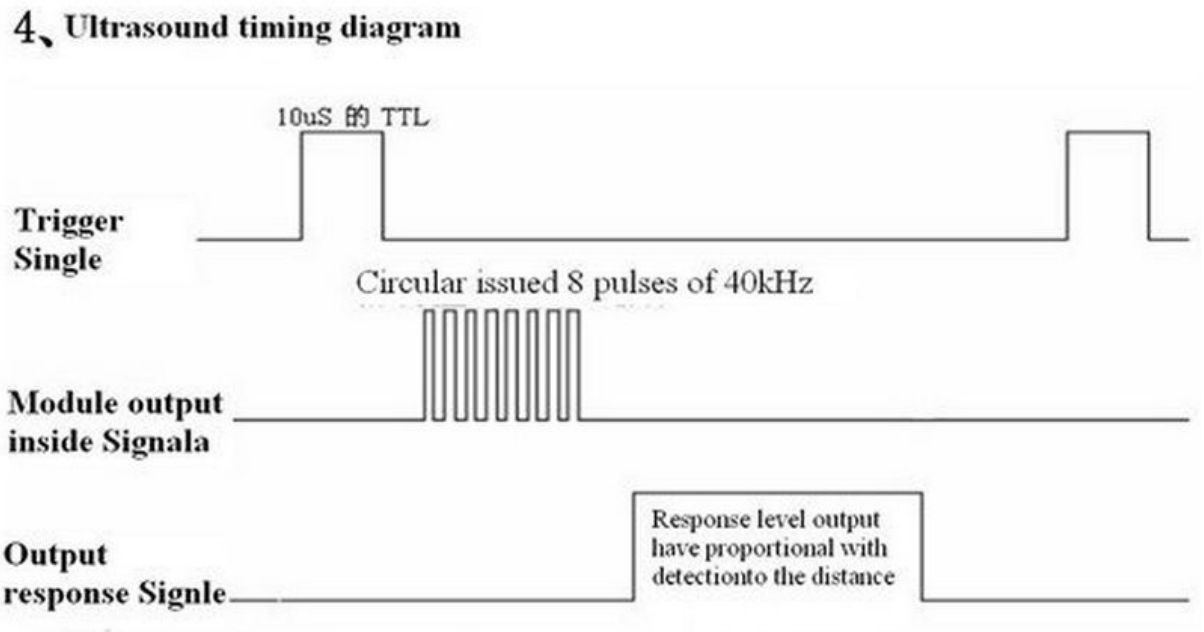
Inkrementálne enkódery merajú aktuálnu rýchlosť otáčania motorov, ktorá sa používa na PID reguláciu rýchlosti a výpočet odometrie. Enkódery generujú 7 pulzov na otáčku motora, čo na výstupe prevodovky 100:1 dáva 700 pulzov na otáčku kolesa. Tieto pulzy sú automaticky počítané časovačmi TIM4 a TIM5, ktoré sú použité v režime enkóderového rozhrania. Časovače sú nastavené na režim snímania 4x (počíta sa nábežná aj dobežná hrana oboch signálov enkóderu), celkovo teda máme rozlíšenie 2800 pulzov na otáčku kolesa. Uhlovú rýchlosť motorov získame z rozdielu aktuálnej a predošlej hodnoty časovača.



Hodnota časovača v režime enkóderového rozhrania. TI1,2 - kvadratúrny výstup enkódera, Counter - hodnota počítadla

Výstup ultrazvukového senzora je pulz (ECHO), ktorého dĺžka je rovnaká ako čas od vyslania ultrazvukového pulzu po jeho prijatie. Na presné meranie času je použitý časovač TIM2 v režime počítania nahor s frekvenciou 10MHz, čo zodpovedá časovému rozlíšeniu 0.1µs. Funkcia pre meranie vzdialenosti najprv vyšle do senzoru štartovací pulz (TRIGGER) s dĺžkou 10µs. Následne sa v prerušení čaká na nábežnú hranu signálu ECHO, kde sa vynuluje a spustí časovač. S príchodom dobežnej hrany sa v prerušení zastaví časovač TIM2 a z jeho hodnoty (TIM2CNT) sa vypočíta nameraná vzdialenosť v metroch podľa vzťahu

$$d = \frac{TIM2CNT}{58000}$$



Časový priebeh komunikácie ultrazvukového senzoru

5.2 Riadenie motorov a odometria

Kvôli meniacemu sa napätiu batérie a rôznemu sklonu trate (na mostíku) je potrebné zabezpečiť aktívne spätnoväzobné riadenie rýchlosti motorov. Na tento účel je použitý PID regulátor rýchlosti motorov.

Výstupom vysokoúrovňového riadenia je požadovaná dopredná rýchlosť v a uhlová rýchlosť ω. Z toho vypočítame žiadané uhlové rýchlosti ľavého a pravého motora ω_L, ω_R podľa vzťahov

$$\omega_l = \frac{v - d.\omega}{r} \qquad \omega_p = \frac{v + d.\omega}{r}$$

kde d je vzdialenosť kolies od ťažiska robota a r je polomer kolies.

Riadenie prebieha s frekvenciou 250Hz a je vykonávané pre oba motory samostatne. Vstupom je žiadaná rýchlosť kolesa v radiánoch za sekundu a aktuálna rýchlosť motora získaná z enkóderov, z ktorých sa vypočíta regulačná odchýlka. Hodnota integračnej zložky regulátora je obmedzená na rozsah (-1;1), aby sa zabránilo windupu. Výstupom je bezrozmerný riadiaci signál *u* v rozsahu (-1;1), kde 0 znamená zastavené motory, 1 je maximálna rýchlosť vpred a -1 je maximálna rýchlosť vzad. Táto hodnota je následne prepočítaná na číslo v rozsahu (0;4095), ktoré udáva striedu pulzne šírkovej modulácie (PWM), ktorou je ovládaná rýchlosť motorov. Celková schéma regulátora je na nasledujúcom obrázku.

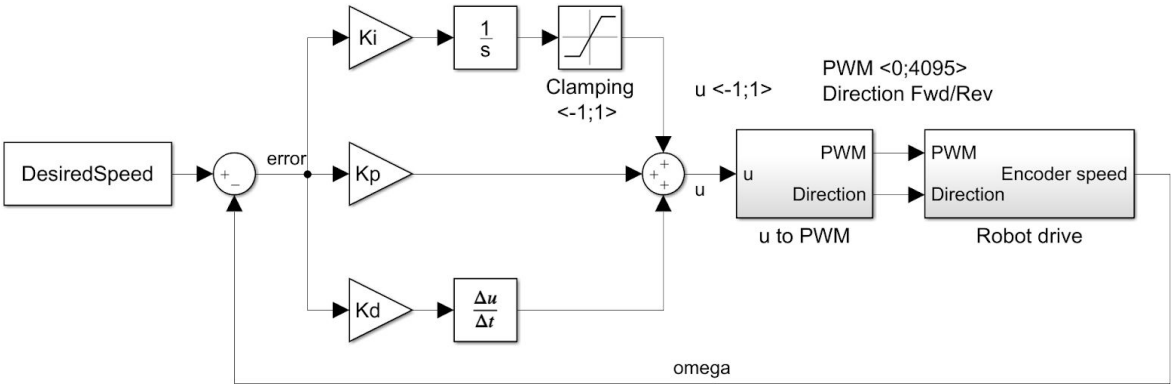
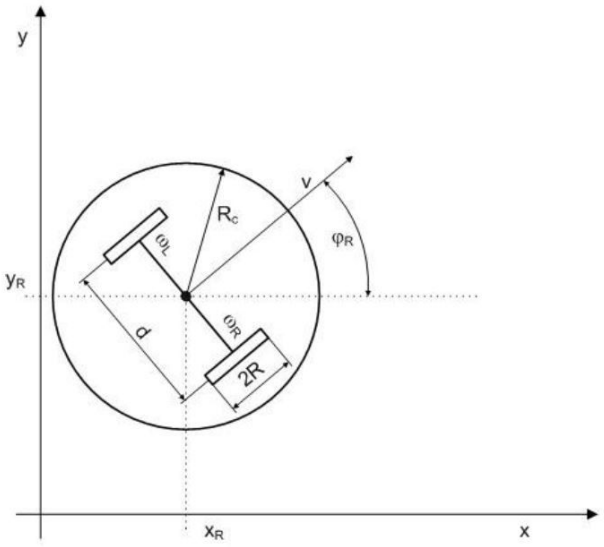


Schéma PID regulátora rýchlosti motorov

Súčasne s regulátorom prebieha aj výpočet odometrie. Odometria poskytuje odhad aktuálnej polohy a otočenia robota v priestore na základe rýchlosti otáčania kolies. Na nasledujúcom obrázku je znázornená kinematická štruktúra dvojkolesového diferenciálneho podvozku, kde *v* je dopredná rýchlosť, φ_R je aktuálny uhol otočenia, d je vzdialenosť kolies, R je polomer kolies, ω_L, ω_R sú uhlové rýchlosti motorov a x_R, y_R sú aktuálne súradnice ťažiska robota.



Výpočet odometrie je realizovaný nasledovnými vzťahmi:

$$v_K = (\omega_{L_K} + \omega_{R_K})\frac{R}{2}$$

$$\varphi_{R_{K+1}} = \varphi_{R_K} + \Delta t(\omega_{R_K} - \omega_{L_K})\frac{R}{d}$$

$$x_{R_{K+1}} = x_{R_K} - \frac{d(v_r + v_l)}{2(v_r - v_l)}(\cos\varphi_{K+1} - \cos\varphi_K)$$

$$y_{R_{K+1}} = y_{R_K} - \frac{d(v_r + v_l)}{2(v_r - v_l)}(\sin\varphi_{K+1} - \sin\varphi_K)$$

V prípade, ak sú obe rýchlosti motorov rovnaké:

$$x_{R_{K+1}} = x_{R_K} + \Delta t v_K \cos\varphi_{R_K}$$

$$y_{R_{K+1}} = y_{R_K} + \Delta t v_K \sin\varphi_{R_K}$$

6 Vysokoúrovňové riadenie

Náš robot používa vizuálny systém, a teda pracuje s videom v pomerne vysokej frekvencii. Zatiaľ je FPS (z ang. Frames Per Second - snímky za sekundu - frekvencia spracovania obrazu) nastavené na hodnotu 10. Kvôli tomuto je nutné použiť ďalšiu výpočtovú jednotku, keďže mikroprocesor, ktorý zabezpečuje nízko-úrovňové riadenie by takáto úlohu nezvládal.

Ako výpočtovú jednotku pre vysoko úrovňové riadenie sme zvolili Raspberry Pi Model 2 (ďalej v dokumente len ***RPi***). disponuje procesorom so štyrmi jadrami o takte 900 MHz a 1 GB operačnej pamäte.

V tejto úrovni riadenia budú riešené nasledovné úlohy:

- Spracovanie obrazu z kamery
- Detekcia čiary z obrazu kamery
- Riadiaci algoritmus - výpočet regulačnej odchýlky a výpočet akčného zásahu

Ukážka cyklu vysoko úrovňového riadenia

6.1 Vizuálny systém robota

Vizuálny systém robota je pomerne jednoduchý, keďže ako senzor používa len jeden prvok, a tým je kamera, ktorá je pripojená k RPi. Kamera, ktorú náš systém využíva má názov **RASPBERRY Pi Camera Board** a je to kamera určená na používanie priamo s RPi. K RPi sa pripája pomocou rozhrania **CSI**. Dokáže vyhotoviť video s FULL HD rozlíšením a 30 FPS. Tieto parametre su pre nás plne postačujúce, keďže hodnoty FPS aj rozlíšenie sú v našom prípade nižšie - rozlíšenie používame 320x240px a FPS s hodnotou 10.

Ako nástroj pre softvérový prístup ku kamere a získanie snímok z nej používame známu knižnicu **OpenCV** v prostredí programovacieho jazyka **Python.** Detekciu čiary na snímkach z kamery taktiež realizujeme pomocou nástrojov z knižnice OpenCV.

6.1.1 Detekcia čiary z obrazu kamery

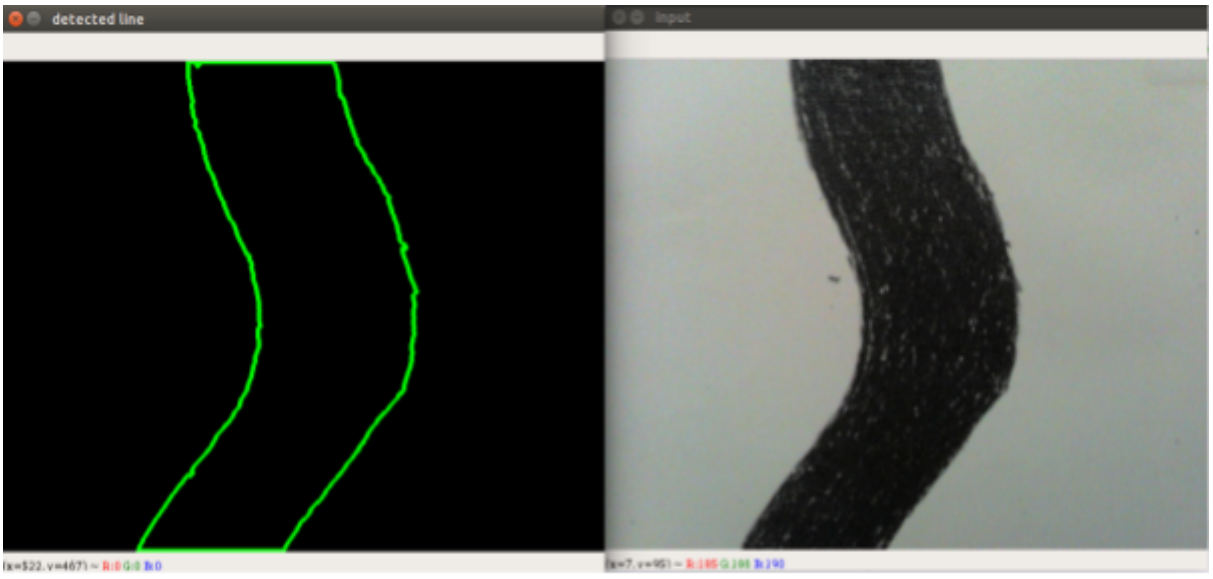
Čiaru z obrazu kamery dokážeme pomerne jednoducho identifikovať pomocou klasických deterministických metód počítačového videnia. Je to najmä vďaka faktu, že pri kategórii LineFollower v súťaži Istrobot je čiara viditeľná veľmi zreteľne - čiara je čierna na čisto bielom podklade.

v prvej fáze tejto úlohy potrebujeme vytvoriť binárnu masku, kde biele pixely (pixely s hodnotou 1) budú predstavovať časť obrazu, kde sa nachádza čiara. Ostatné pixely budú

čierne (s hodnotou 0). Toto dosiahneme aplikovaním nasledujúcich krokov (poradie je dôležité, čísla predstavujú poradie):

1. Konverzia snímku na čiernobiely (pixel má len jeden parameter - intenzitu)
2. Ignorovanie pixelov, ktoré majú intenzitu mimo definovaný rozsah - rozsah sa volí v závislosti od svetelných podmienok, svetelnej odrazivosti podkladu atď.
3. Dilatácia snímku
4. Nájdenie kontúr, pomocou funkcie **findContours** z knižnice OpenCV
5. Výpočet plochy všetkých nájdených kontúr pomocou funkcie **contourArea** z knižnice OpenCV
6. Ponechanie kontúry s najväčšou plochou - predpokladáme, že najväčší objekt na obraze je sledovaná čiara

V tomto momente máme identifikovanú kontúru sledovanej čiary a pomocou funkcie **fillPoly** dokážem splniť vyššie spomínaný cieľ tejto úlohy.

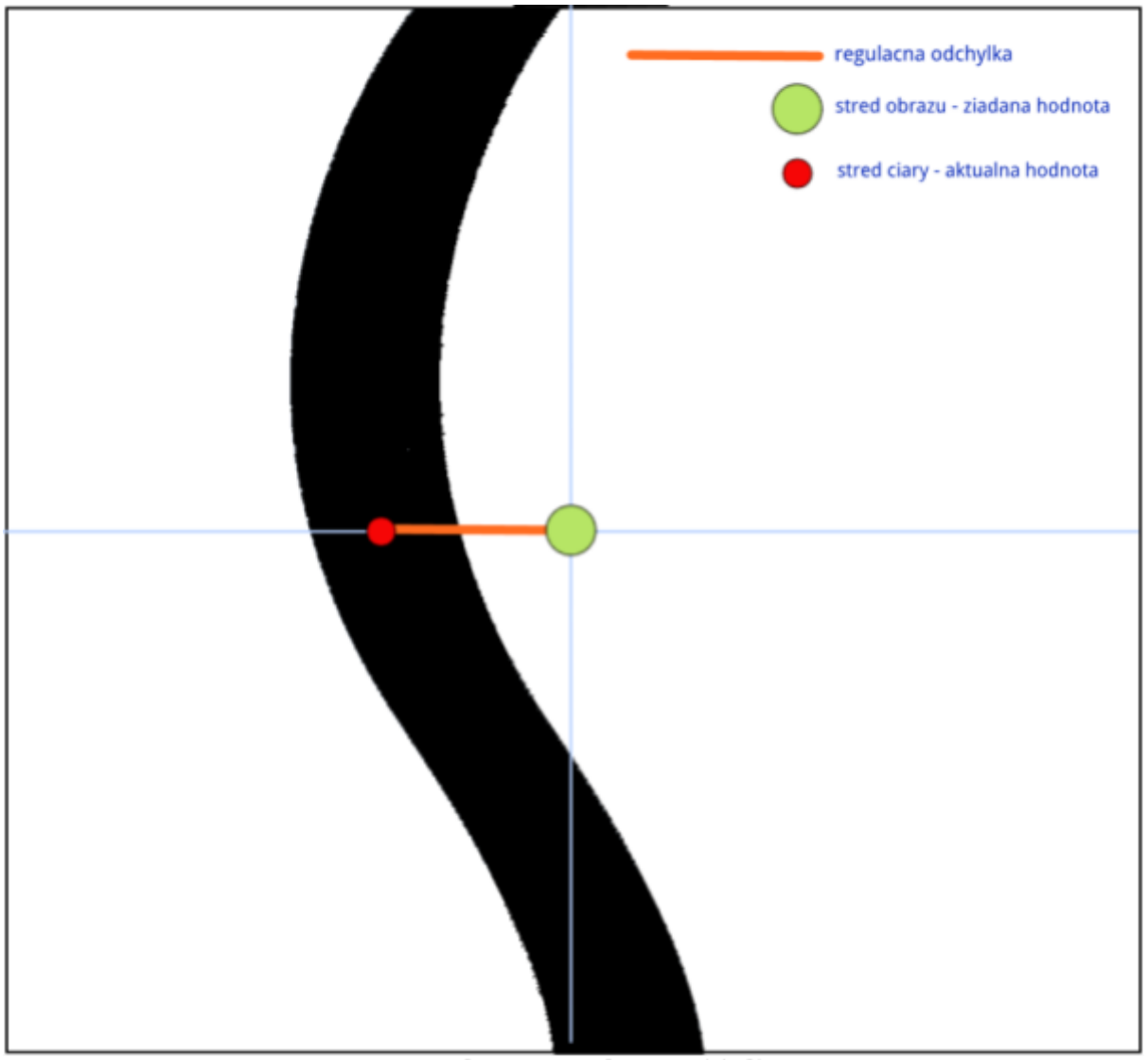


detekcia čiary z obrazu kamery - detekcia kontúr sledovanej čiary

6.1.2 Určenie regulačnej odchýlky

V predchádzajúcom kroku sme dokázali určiť polohu všetkých pixelov obrazu, na ktorých sa nachádza sledovaná čiara. Stále však potrebujeme kvantifikovať jednotnú regulačnú odchýlku ako skalár, na základe ktorej pošleme do nízkej úrovne riadenia riadiaci zásah. Ako regulačnú odchýlku sme zvolili vzdialenosť stredu čiary v línii stredu snímku od samotného

stredu snímku. Pre ilustráciu:



Vizualizácia regulačnej odchýlky

6.1.2.1 Určenie stredu čiary z obrazu kamery

V predchádzajúcej sekcii sme jasne pomenovali a určili regulačnú odchýlku - hodnotu, ktorú sa budeme snažiť minimalizovať. Pri implementácii sme však riešili problém, ako určiť stred čiary v línii stredu obrazu - aktuálnu hodnotu - červená kružnica na obrázku vyššie.

Počítame s tým, že poznáme všetky pixely, ktoré tvoria kontúru čiary, a tak urobíme nasledovnú postupnosť krokov:

1. **Iterujeme všetky body kontúry čiary**
2. **Ak bod kontúry má y súradnicu (výšku) v intervale +-3 pixely polovice výšky obrazu, pridajme tento bod do listu nájdených bodov**
3. **Urobíme priemer súradníc všetkých nájdených bodov z kroku 2**

7 Prepojenie oboch úrovní riadenia

Na začiatok si uvedieme schému fungovania celého riadenia robota.

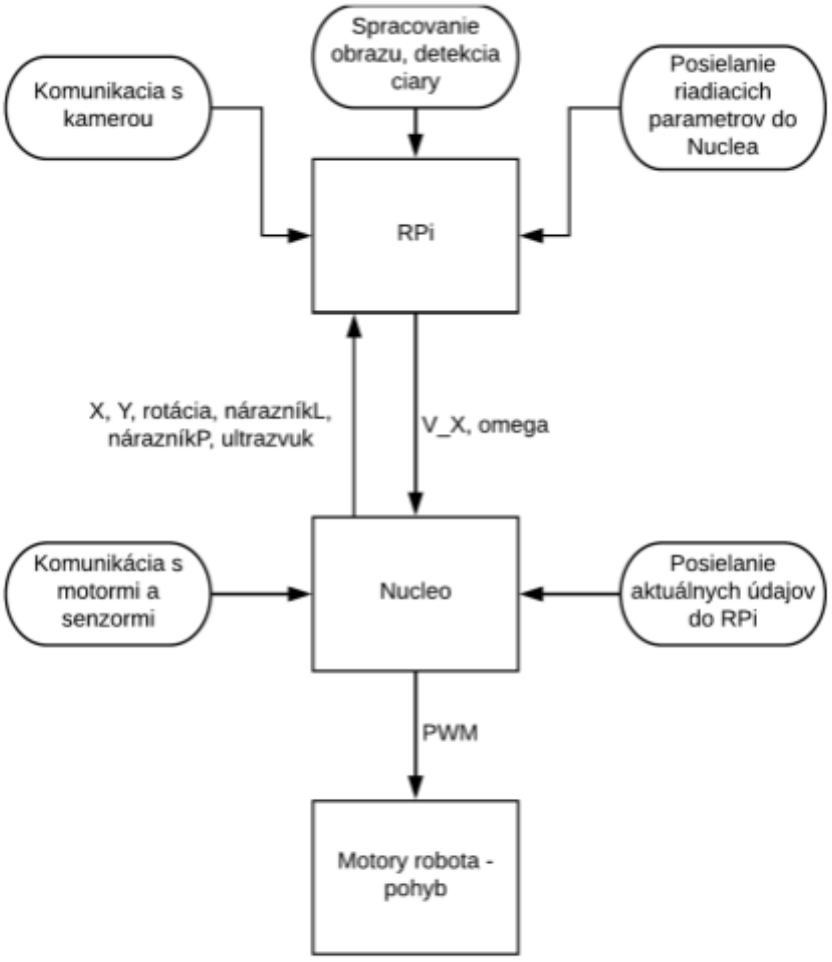


Schéma fungovania riadiaceho systému robota

Zo schémy vidíme základné fungovanie systému, no potrebujeme ešte doplniť pár informácií, aby bolo zrejmé, ako RPi a Nucleo spolu komunikujú:

- RPi a Nucleo na komunikáciu medzi sebou používajú UART
- Je definovaný nemenný formát správ - komunikačný protokol (uvedený bude nižšie)

7.1 Komunikačný protokol RPi <-> Nucleo

Pre správne fungovanie komunikácie medzi RPi a Nucleo musíme definovať komunikačný protokol.

- Prenos. rýchlosť 112500baud
- Protokol je definovaný ako textový - človekom čitateľný
- Premenná obsahujúca správu je **string**
- Ukončovací znak - **‘/n’,** po jeho prijatí sa spracuje to, čo je prijaté v bufferi
- Formát správy, ktorý sa posiela z RPi do Nuclea
    - `"v_x[-1..1%.3f],omega[-1..1%.3f]\n"`
        - **v_x** - žiadaná dopredná rýchlosť (m/s)
        - **omega** - žiadaná uhlová rýchlosť (rad/s)
- Formát správy, ktorý sa posiela z Nuclea do RPi
    - `"OdomX[float%.4f],OdomY[float%.4f],OdomRot[float%.4f],BumperL[0/1int],BumperR[0/1int],ultrasound[float%.2f]\n"`
        - **OdomX** - aktuálna súradnica X podľa odometrie (m)
        - **OdomY** - aktuálna súradnica Y podľa odometrie (m)
        - **OdomRot** - aktuálna rotácia podľa odometrie (rad)
        - **BumperL** - informácia, či je ľavý nárazník v kontakte s prekážkou
        - **BumperR** - informácia, či je pravý nárazník v kontakte s prekážkou
        - **ultrasound** - vzdialenosť prekážky pred robotom, údaj z ultrazvuku (m)
            - 0.0-2.0m: platný údaj
            - < 0: neplatný údaj (mimo rozsah)

8 User stories

8.1 User stories riešené v rámci semestra

AKO	MUSÍM	ABY SOM	MIERA SPLNENIA
Robot	spĺňať pravidlá súťaže Istrobot v kategórií Stopár	sa mohol zúčastniť	100%
Robot	mať konštrukciu	mal podvozok a moje komponenty mali miesto	100%
Robot	mať mikroprocesor (STM32 Nucleo)	mal nízko úrovňové riadenie	100%
Robot	mať Raspberry Pi	vysoko úrovňový prístup (kamera, riadiace príkazy)	100%
Robot	mať kameru	vedel detegovať čiaru s dostatočným rozlíšením	100%
Robot	čítať enkódery	mal funkčnú odometriu a reguláciu rýchlosti	100%
Robot	mať batérie	mal zdroj energie	100%
Robot	mať motory a kolesá	bol schopný pohybu	100%
Robot	mať nárazník so spínačom	vedel že narazil do prekážky	100%
Robot	mať senzor vzdialenosti	mohol detegovať prekážku pred sebou	100%
Robot	spätnoväzobné riadenie rýchlosti motorov	vedel presne riadiť smer a dokázal vyjsť na rampu	100%
Robot	počítať odometriu	vedel určiť svoju polohu a prejdenú vzdialenosť	90%
Robot	komunikačný protokol	si medzi RPI a Nucleo vedel vymieňať informácie	100%

Robot	mať tlačítko na spustenie motorov	sa vedel pohnúť	100%
RPI a Nucleo	byť prepojené cez sériový port	vedeli navzájom komunikovať	100%
RPi	detegovať čiaru z kamery	vedel riadiť sledovanie čiary	95%
RPi	mať nainštalované knižnice (OpenCV, piCamera)	vedel vyvíjať algoritmy pre sledovanie čiary	100%
RPi	dávať jednoznačnú odchýlku od čiary	vedel robot jednoznačne riadiť	100%
Súťažiaci	odštartovať robota	sa mohol zúčastnil súťaže	100%

8.2 Dodatočné user stories pre potreby súťaže Istrobot

AKO	**MUSÍM**	**ABY SOM**
Robot	mať algoritmus	obišiel prekážku tehla
Robot	mať algoritmus	prešiel prekážku olejová škvrna

9 Záver

Počas semestra sme dokázali doviesť projekt Stopára do štádia, kedy je všetko plne pripravené na samotný, čistý softvérový vývoj pokročilých riadiacich algoritmov sledovania čiary pomocou robota. To znamená (kompletná funkcionalita sa nachádza v User Stories):

- je pripravená platforma pre vizuálny systém, tj. prístup ku kamere práca s obrazom, je navrhnutý prvý prototyp sledovania čiary
- z vyššej vrstvy riadenia dokážeme posielať riadiace príkazy, a tým generovať ľubovoľnú trajektóriu
- údaje zo všetkých senzorov sú k dispozícii pre vývojárov, a to vrátane odometrie - už robot zastaví, ak je pred ním prekážka (podľa ultrazvuku)
- komunikácia medzi RPi a Nucleo je vyriešená
- kompletne vyriešená je konštrukcia, mechanické časti robota a elektronika

V rámci predmetu Manažment Projektov sme si taktiež vyskúšali agílny manažment tohto projektu. Každý týždeň sme absolvovali šprinty, kde sme sumarizovali, koľko práce sme za konkrétny šprint stihli, resp. nestihli. Taktiež sme plánovali do budúcna a snažili sa smerovať projekt tak, aby sme stíhali každý jeden šprint dodať dohodnutý výstup. Nástroje, ktoré nám pri agílnom vývoji pomohli, boli:

- Slack/Facebook group chat - komunikačné nástroje
- Trello - manažment úloh
- Google Docs/Sheets - nástroje na tvorbu dokumentov online, umožňujúce kolaboráciu
- Git - verzionovací systém, kde sme postupne zverejňovali svoje zdrojové kódy tak, aby ich mal každý k dispozícii. Repozitár je k dispozícii na adrese https://github.com/3zuli/mprojbot

Výsledok hodnotíme ako dobrý a sme s ním spokojní. Počas agílneho vývoja sme sa zhodli, že našim cieľom bude “rozhýbať” robot, tj. že robot bude schopný hýbať sa na základe riadiacich pokynov z vyššej vrstvy riadenia. Toto má za dôsledok, že sme sa ocitli v štádiu, kedy si ktokoľvek robota vezme domov a môže na ňom skúšať vyvíjať rôzne algoritmy. Nepotrebuje k tomu znalosti napríklad z vnorených systémov (nízka úroveň riadenia). Tým pádom si myslíme, že takto vytvoríme lepšiu možnosť paralelizácie ďalšieho vývoja a zvýšime tým naše šance, čo najlepšie sa umiestniť v súťaži Istrobot.