

**Slovenská technická univerzita v Bratislave
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**Programátorská dokumentácia
Semestrálny projekt
Vozidlo s bug0 algoritmom**

Obsah

Obsah	2
Úvod	3
Použité periférie.....	3
Hlavný algoritmus	3
Ovládač pre servo-motory	3
Ovládač pre diaľkomery	3
Ovládač pre magnetometer	3
Ovládač pre Bluetooth	4
Indikačná LED	4
Hlavný algoritmus	4
Ovládač pre servo-motory	5
Ovládač pre diaľkomery.....	6
Ovládač pre magnetometer	6
Ovládač pre Bluetooth	7
Pozostáva zo súborov:	7
Indikačná LED	7

Úvod

Firmvér pre stm32l152 MCU robota využívajúceho bug0 algoritmus sme programovali v jazyku C pomocou vývojového prostredia Atollic TrueStudio.

Podľa špecifikácie robota, jeho hardvérových súčastí a požadovaného správania firmvér obsahuje nasledovné súčasti:

- Hlavný algoritmus
- Ovládač pre servo-motory
- Ovládač pre diaľkomery
- Ovládač pre magnetometer
- Ovládač pre Bluetooth
- Indikačná LED

Použité periférie

Hlavný algoritmus

Časovač TIM 9 – spúšťanie pravidelného odosielania uhla smerovania robota

Ovládač pre servo-motory

Časovač TIM 2 – šírko-impulzne modulovaný signál pre riadenie servo-motorov.

Výstup PB11 – šírko-impulzne modulovaný signál pre ľavé servo.

Výstup PB10 - šírko-impulzne modulovaný signál pre pravé servo.

Ovládač pre diaľkomery

Výstup PA9 – spúšťanie ľavého diaľkomeru.

Výstup PA11 – spúšťanie predného diaľkomeru.

Výstup PA12 – spúšťanie pravého diaľkomeru.

Vstup PB0 – meranie dĺžky impulzu z ľavého diaľkomeru.

Vstup PB1 – meranie dĺžky impulzu z predného diaľkomeru.

Vstup PB4 – meranie dĺžky impulzu z pravého diaľkomeru.

Časovač TIM3 – meranie dĺžky impulzov z diaľkomerov.

Časovač TIM7 – časovanie spúšťacích impulzov pre diaľkomery.

Časovač TIM10 – spúšťanie merania diaľkomerov v pravidelnom intervale.

Ovládač pre magnetometer

Časovač TIM9 – spúšťanie čítania z magnetometra v pravidelnom intervale.

Výstup PB8 – SCL i2c zbernice.

Výstup/vstup PB9 – SDA i2c zbernice.

Ovládač pre Bluetooth

Bluetooth modul je pripojený na hardvér USART, takže po hardvérovej stránke sa Bluetooth pre MCU správa ako USART.

Sériový port USART 3.

Výstup PC10 – TXD odosiela linka sériového portu.

Vstup PC11 – RXT prijíma linka sériového portu.

Indikačná LED

Časovač TIM6 – spúšťanie a vypínanie indikačnej LED.

Výstup PA5 – indikačná LED.

Hlavný algoritmus

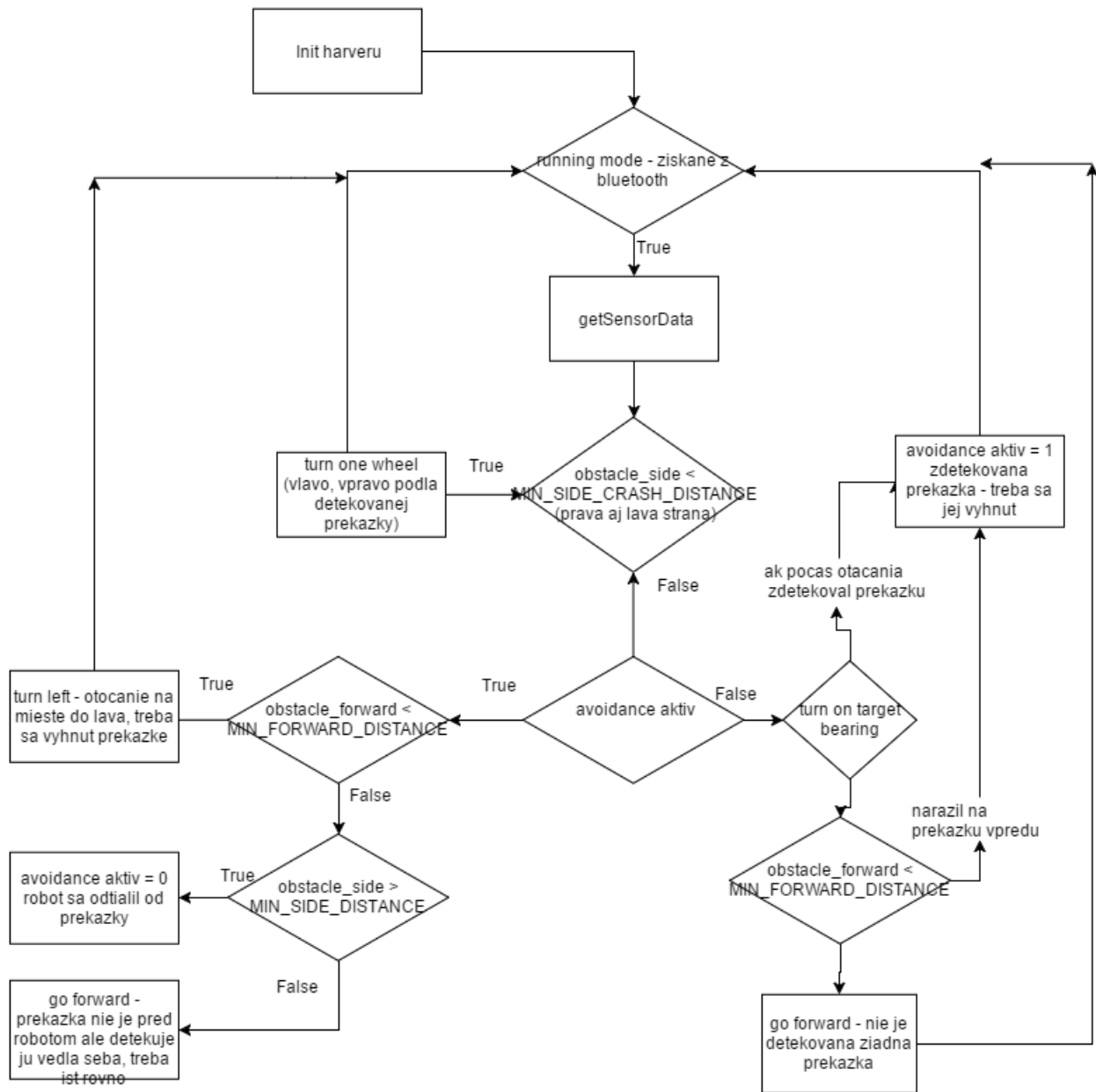
Súbormi hlavného algoritmu sú súbory:

- bug0.h,
- bug0.c.

Tieto súbory obsahujú funkciu **init**, ktorou sa inicializujú všetky ostatné periférie a všetky funkcie potrebné pre chod bug 0 algoritmu:

- run – jeden cyklus programu
- stop – zastavenie vozidla
- go_forward – chod vpred
- turn_left – otáčanie vľavo na mieste
- turn_left_one_wheel – otáčanie vľavo jedným kolesom
- turn_right – otáčanie vpravo na mieste
- turn_right_one_wheel – otáčanie vpravo jedným kolesom
- turn – otáčanie vozidla na žiadaný smer

Kompas sa využíva iba pri vyrovnávaní sa na žiadaný smer, vyhýbanie sa prekážkam je riadene len pomocou diaľkomerov. Náhľad algoritmu môžete vidieť vo vývojovom diagrame. Ak sa zdetegovala prekážka robot sa dostal do stavu avoidance_aktiv a keď sa od prekážky vzdialil na definovanú vzdialenosť sa tento stav zrušil a robot sa snažil opäť dostať na žiadaný smer.



Obrázok 1 Vývojový diagram riadiaceho algoritmu

Ovládač pre servo-motory

Ovládač pre servá pozostáva zo súborov:

- motorctrl.h,
- motorctrl.c.

Servá sa inicializujú pomocou funkcie **Motor_init**, ktorá inicializuje časovač pre PWM výstupné porty. Ďalej tento ovládač obsahuje funkcie na zmenu šírky impulzov, ktorými sa ovládajú servá:

- **left_motor_set_speed**
- **right_motor_set_speed**

Servo sa ovláda impulzami dĺžky 1,3ms až 1,7ms, pričom 1,5ms je stav kedy servá stoja. Funkcie **motor_set_speed** umožňujú reguláciu rýchlosti v rozsahu -20 až 20 jednotiek na ktoré sú rozdelené impulzy. Riadiaci impulz prichádza do serva minimálne každých 20ms.

Ovládač pre diaľkomery

Ovládač pre diaľkomery pozostáva zo súborov:

- sensor.h
- sensor.c

Senzory sa inicializujú pomocou funkcie **sensorInit()**, ktorá volá inicializáciu jednotlivých periférií potrebných pre diaľkomery. Ide o funkcie začínajúce reťazcom **sensorInit**.

Časovač pravidelne volajúci meranie diaľkomermi spustí meranie funkciami **SensorMeasure**, ktoré vytvoria spúšťací impulz do diaľkomerov. Dĺžka impulzu je časovaná časovačom pre spúšťanie senzorov.

Diaľkomery po nameraní hodnoty vyšlú impulz, ktorého dĺžka je zaznamenaná časovačom merania dĺžky impulzu, ktorý vyvolá prerušenie pri nábežnej a dobežnej hrane impulzu. Tento časovač má tri rôzne handlers podľa toho ktorý diaľkometer je aktívny, **SensorCaptureHandler**. Handler prerušenia ukladá hodnoty trvania impulzu do globálnej štruktúry

Hlavný program preberá **SensorCaptureStruct** odkiaľ sú tieto hodnoty prevzaté a prepočítané na hodnotu vzdialenosti funkciami **SensorGetDistance**, ktorými ich hlavný program vie získať.

Ovládač pre magnetometer

Ovládač magnetometra pozostáva zo súborov:

- HMC5883L.h
- HMC5883L.c
- I2c.h – upravené
- I2c.c – upravené

Na nastavenie 3-osového digitálneho kompasu HMC5883L slúži funkcia **writeBytesToCompass**. Pomocou I2C komunikácie zapisujeme hodnoty do prvých troch adries v registroch, ktoré slúžia na nastavenie kompasu.

Nastavenie prvej adresy v registri: **writeByteI2C1(HMC5883L_ADDRESS_W, 0x00, 0x70)**; Adresu 0x00 sme nastavili na hodnotu 0x70, čím sme dosiahli: frekvenciu merania = 15 Hz.

Nastavenie druhej adresy v registri: **writeByteI2C1(HMC5883L_ADDRESS_W, 0x01, 0x0A)**; Adresu 0x01 sme nastavili na hodnotu 0x0A, čím sme dosiahli: zosilnenie meraného magnetického poľa = 5.

Nastavenie tretej adresy v registri: **writeByteI2C1(HMC5883L_ADDRESS_W, 0x02, 0x00)**; Adresu 0x02 sme nastavili na hodnotu 0x00, čím sme dosiahli nepretržitý režim merania.

Na prečítanie dát z 3-osového digitálneho kompasu HMC5883L slúži funkcia **readDataCompass**. Pomocou I2C komunikácie prečítame hodnoty štvrtej až deviatej adresy v registroch, z ktorých dostaneme silu magnetického poľa osí X, Y a Z. Z magnetometra čítame pomocou funkcie **readDataHMC5883L**.

Nasledovne na prepočítanie zosilnení na uhol natočenia (od -180° do 180°) v osi XY slúži funkcia:

```
heading = 180 * atan2((double)xForce,(double)yForce)/M_PI;
```

Na uľahčenie výpočtov sme si upravili rozsah natočenia od 0° do 360° pomocou funkcie:

```
if(heading < 0)
    heading += 360;;
```

Meranie magnetometra je pravidelne spúšťané časovačom a ukladané do globálnej premennej, odkiaľ si hodnotu smerovania robota môže hlavný program prevziať pomocou funkcie **compass_get_heading**.

Ovládač pre Bluetooth

Pozostáva zo súborov:

- bluetooth.h
- bluetooth.c

V týchto súboroch sa nachádza funkcia **initUSART3** pre inicializáciu sériovej linky USART 3. Handler prerušení zo sériovej linky vyhodnocuje prijaté znaky a podľa ich hodnoty zapíše príznakové globálne premenné, ktoré určujú ďalšie chovanie robota podľa užívateľského manuálu.

Hlavný program môže posilať informácie po sériovej linke pomocou funkcie **sendValue**, ktorá sa vykonáva iba v debug móde. Pravidelné odosielanie smerovania robota časovačom na sériovú linku je uskutočňované priamo funkciou **putcUSART3**.

Indikačná LED

Firmvér pre indikačnú LED sa nachádza v súboroch:

- runIndicatorLED.h
- runIndicatorLED.c

Tieto súbory obsahujú funkcie na inicializáciu opakovaného blikania indikačnej LED **init_indicator_LED**, po ktorej spustení LED začne blikať s frekvenciou 1Hz.