## Popis funkcií zdrojového kódu

V rámci zdrojového kódu budú popisované iba časti kódu využívané na lokalizáciu a polohovanie.

1. **Inicializácia hodnôt:**
   * V tejto časti kódu sa inicializujú hodnoty, keď **datacounter** je rovné 0. Tieto hodnoty sa používajú pre ďalšie výpočty.
   * **encoderLeftValue** a **encoderRightValue** sa nastavia na aktuálne hodnoty **robotdata.EncoderLeft** a **robotdata.EncoderRight**.
   * **iniPhi** sa vypočíta ako **robotdata.GyroAngle** vynásobené 2π/360.0, prevedené na radiány.
2. **Overflow pre enkodéry:**
   * Táto časť kódu kontroluje pretečenie enkodérov (EncoderLeft a EncoderRight).
   * Ak je hodnota **robotdata.EncoderLeft** menšia ako **encoderLeftValue**, kontroluje sa, či rozdiel je väčší ako 60000. Ak áno, inkrementuje sa **encoderLeftOverflow**.
   * Podobne sa kontroluje aj **robotdata.EncoderRight** voči **encoderRightValue**.
3. **Výpočet premenných prejdenej vzdialenosti pre pravé (lr) a ľavé (ll) koleso:**
   * Premenné **lr** a **ll** predstavujú hodnoty z enkodérov prepočítané na metre pomocou **tickToMeter** faktoru.
   * Aktualizujú sa hodnoty **encoderRightValue** a **encoderLeftValue** na aktuálne hodnoty enkodérov.
   * Nastavujú sa **encoderRightOverflow** a **encoderLeftOverflow** na 0.
4. **Výpočet l a phi:**
   * **l** je celková vzdialenosť vypočítaná ako priemerná hodnota **lr** a **ll**.
   * **phi** je uhol natočenia robota získaný z **robotdata.GyroAngle** a **iniPhi**.
5. **Aktualizácia pozície robota:**
   * Na základe hodnôt **l** a **phi** sa aktualizuje pozícia robota.
   * Súčasná pozícia **x** a **y** sa upravujú o novú hodnotu v smere **phi**.
6. **Nastavenie cieľovej pozície:**
   * Na tomto mieste je potrebné zadefinovať cieľovú pozíciu robota.
   * **targetX** a **targetY** predstavujú požadované súradnice cieľovej pozície.
7. **Riadenie pohybu robota:**
   * Ak je premenná **go** nastavená na *true*, robí sa riadenie pohybu robota na základe cieľovej pozície.
   * Používajú sa nasledujúce premenné:
     + **P1** a **P2** sú faktory, ktoré ovplyvňujú rýchlosť otáčania a pohybu robota.
     + **diffX** a **diffY** predstavujú rozdiel medzi cieľovou pozíciou a aktuálnou pozíciou robota.
     + **pa** je vypočítaný uhol medzi cieľovou pozíciou a aktuálnou pozíciou robota.
   * Ak je odchýlka uhla otáčania robota (**pa - phi**) väčšia ako určený prah (PI/25.0), robot sa otáča s rýchlosťou **(pa - phi) \* P1**.
   * V opačnom prípade sa určuje rýchlosť pohybu robota, na základe vzdialenosti medzi aktuálnou a cieľovou pozíciou. Rýchlosť je obmedzená na hodnoty medzi 50 a 200.
   * Ak sú odchýlky v oboch smeroch menšie ako 5.0, robot sa zastaví a vypíše sa správa o dosiahnutí cieľovej pozície.
8. **Aktualizácia hodnôt a výstup:**
   * **datacounter** sa inkrementuje.
   * Vypisujú sa hodnoty **robotdata.GyroAngle** a **robotdata.EncoderRight** na štandardný výstup.

## Zoznam premenných

* long double d = 0.23;
* long double tickToMeter = 0.000085292090497737556558;
* double lr = 0.0;
* double ll = 0.0;
* double encoderLeftValue = 0.0;
* double encoderRightValue = 0.0;
* double x = 0.0;
* double y = 0.0;
* double l = 0.0;
* double phi = 0.0;
* int encoderLeftOverflow = 0;
* int encoderRightOverflow = 0;
* double targetX = 0.0;
* double targetY = 0.0;
* boolean go = false;
* double iniPhi = 0;

## Spúšťanie programu pre navigáciu

Pohyb robota smerom k cieľu sa aktivuje nastavením premennej go na *true*. Táto hodnota sa nastavuje na danú hodnotu vo funkcii, ktorá spracováva stlačenia tlačidla cez UI. Po stlačení tlačidla smerom v pred (forward) sa nastaví hodnota na *true* a pri najbližšom zavolaní funkcie processThisRobot sa robot pohne smerom k cieľu. Tento program sa volá cyklicky a teda všetky parametre si robot skontroluje v každom behu cyklu. Pokiaľ robot nemieri priamo na cieľ, zastaví a dotočí sa do želanej pozície. Robot sa úplne zastaví, ak dosiahne cieľovú pozíciu s toleranciou na 5 cm. Premenná go sa nastaví na *false* stlačením tlačidla späť (Beck).