## Popis funkcií zdrojového kódu

V rámci zdrojového kódu budú popisované iba časti kódu využívané na mapovanie.

### **Funkcia processThisRobot**

V rámci tejto funkcie sa sleduje iba poloha robota. Poloha robota je potrebná pre korektné mapovanie prekážok v dvojrozmernom priestore.

* 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 vzdialenost vypočítaná ako priemerná hodnota **lr** a **ll**.
     + **phi** je uhol natočenia robota na 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**.

### **Funkcia proccesThisLidar**

V rámci tejto funcie

* 1. Cyklus **for** iteruje cez skeny získané zo senzora laserového diaľkomera. Skeny obsahujú informácie o vzdialenosti objektu na ktorý dopadol lúč diaľkomera a o uhle natočenia diaľkomera.
  2. Pre každý sken sa vypočítavajú súradnice prekážky **xo** a **yo** na základe uhla a vzdialenosti prekážky a aktuálnej pozície robota
  3. Na skeny sa aplikujú nasledujúce podmienky:
* Ak skenovaná vzdialenosť spĺňa určité kritériá (medzi 150 a 640 alebo medzi 700 a 3000), pokračuje sa v kóde vnútri podmienky.
  1. Hodnoty **xo** a **yo** sa mapujú do iného súradnicového systému. Delia sa 100 a priraďujú sa do premenných **mapX** a **mapY**.
  2. Kontrolujú sa hranice súradníc **mapX** a **mapY**. Ak hodnoty spĺňajú kritériá (medzi 0 a 120), nastaví sa hodnota 1 v príslušnom mieste matice **map**, ktorá predstavuje mapu prostredia.

## Zoznam premenných

* long double celkova = 0;
* 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 phiOldValue = 0.0;
* double d\_alfa = 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 iniEncRight = 0;
* double iniEncLeft = 0;
* double iniPhi = 0;