



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Autonóm robotok működésének futásidőbeli ellenőrzése



Gazder Bence III. évf, (BSc) mérnökinformatikus szakos hallgató
Konzulens: dr. Majzik Istán, MIT
Informatikai technológiák szakirány/Rendszertervezés
Önálló laboratórium összefoglaló
2014/15. II. félév

A félév során robotok futásidőbeli verifikációjával foglalkoztam ROS (Robot Operating System) környezetben. A cél az volt, hogy készítsek egy olyan operációs rendszer komponensét amely futásidőben ellenőrzi a ROS-t futtató robot működését.

A feladat két nagy részfeladatot foglalt magában. Az egyik részfeladat a monitor komponens kódjának megtervezése és implementálása volt, a másik pedig az ezt módosító kódgenerátor amely egy LTL kifejezés segítségével rögzített kifejezést kap mint feltétel és ez alapján a megfelelő monitor komponens forráskódját generálja le.

Mind a két részfeladat kódját C++ nyelven készítettem el a 11-es szabvány alapján. A generátort használó megoldással több előnyre is szerttettem. Ilyenek például a könnyen testreszabható ellenőzés, tehát szabadon szabályozható, hogy milyen funkciókat ellenőrizzen a komponens, illetve a robottípusok közötti váltás is csak egy generátor beállítás lehet majd a későbbiekben.

A félév során a kódgenerátorral sikerült megvalósítanom a tetszőleges fájlba való forráskód beillesztést, illetve a generálandó kódot leíró osztálypéldányok ismeretében való helyes forráskód generálást. A generátor LTL értelmezője amely az osztálypéldányokat hozná létre, nem készült el. Amennyiben az osztálypéldányokat a kód megfelelő részében létrehozzuk, akkor a forrásfájl kimenet az elvárt eredményt adja, és a komponens fordítható lesz.

A monitor komponens első verzióját sikerült elkészítenem a félév során. A komponens képes az ROS-ből érkező üzenet elfogására és kiírására. Teszteléshez érdemes a beépített teszt metódust használni, aminek segítségével a beérkező eseményeket egy előre megírt fájlból olvashatjuk be.

Kiértékelős során fő szempont volt a memória és a processzor erőforrások kímélése. Ennek úgy tettem eleget, hogy a kiértékelés nem egyszerre történik meg, hanem az értékelendő kifejezést apránként értékelem ki, és amennyiben a beérkező események függvényében a kifejezés kiértékelhető igaz vagy hamis értékre akkor nem folytatom a kiértékelést, hanem egy jelzést küldök, amit később komplex események kiváltására lehet felhasználni.