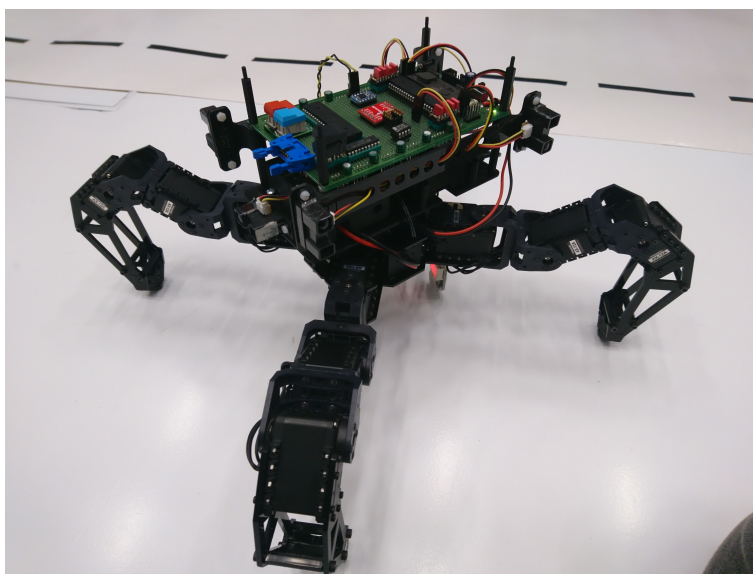


Användarhandledning

Redaktör: Viktor Blidh, Tobias Fellegi, Aron Gosch,
Alfred Hagberg, Mustaf Musse, Niclas Byrsten

23 maj 2018

Version 1.0



Status

Granskad		
Godkänd		

Projektidentitet

Namn	Ansvar	Telefon	E-post
Viktor Blidh	Projektansvarig (PL)	070-6941166	vikbl327@student.liu.se
Tobias Fellegi	Dokumentansvarig (DOK)	070-4282800	tobfe318@student.liu.se
Niclas Byrsten	Designansvarig (DES)	072-8525164	nicby889@student.liu.se
Mustaf Musse	Testansvarig (TST)	076-2633149	musab250@student.liu.se
Aron Gosch	Kvalitetssamordnare (QS)	073-0744622	arogo305@student.liu.se
Alfred Hagberg	Implementationsansvarig (IMP)	076-1459838	alfha306@student.liu.se

E-postlista för hela gruppen: vikbl327@student.liu.se,
tobfe318@student.liu.se, nicby889@student.liu.se,
musab250@student.liu.se, arogo305@student.liu.se,
alfha306@student.liu.se

Kund: Tomas Svensson, tomas.svensson@liu.se

Kursansvarig: Tomas Svensson, tomas.svensson@liu.se

Handledare: Anders Nilsson, anders.p.nilsson@liu.se

Innehåll

1	Inledning	1
2	Roboten	1
3	Start av systemet	2
3.1	Start av autonomt läge	3
3.2	Start av manuellt läge	3
	Appendix	5
A	Banspecifikation	5

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
1.0	2017-12-14	Första versionen	TF, AG	2017-12-18
1.1	2017-12-18	Utfört kompletteringar	TF	

1 Inledning

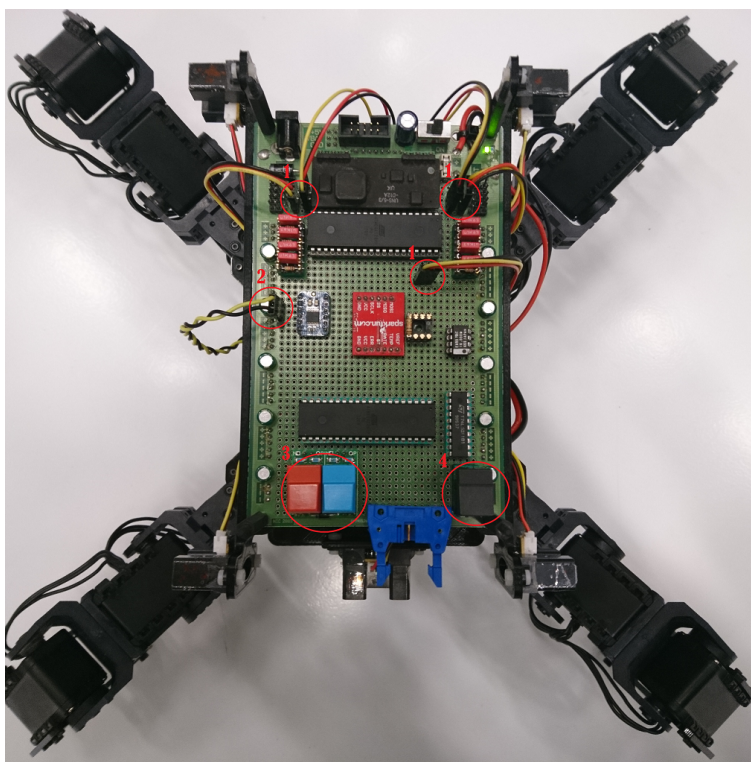
Detta är användarmanualen till roboten Brobot, en fyrbenig robot som har konstruerats i kursen TSEA29 – Konstruktion med mikrodatorer. Syftet med detta dokument är att ge en användare tillräcklig förståelse om hur roboten hanteras och dess funktionalitet. För en mer ingående beskrivning av konstruktion och implementation hänvisas läsaren till projektets tekniska dokumentation.

2 Roboten

Brobot har två lägen, ett manuellt läge, där den styrs av en ansluten dator med hjälp av vårt GUI, samt ett autonomt läge där Brobot själv löser en labyrint med hjälp av sin sensormodul. Banspecifikationen finns bifogad i appendix A. Brobot startar i ett pausat läge där den väntar på att ett GUI ska ta kontakt och ge den kommandon eller att man ska trycka på knappen för det autonoma läget. GUI:t kan utöver att ge kommandon även ändra mellan manuellt och autonomt läge.

3 Start av systemet

För att starta systemet måste först komponenterna förses med spänning, detta görs genom att koppla in ett 3-celligt LiPo batteri till robotplattformen. Det är viktigt att se till att batteriet är ordentligt laddat vid användning, detta kontrolleras lämpligen med att ha en voltmätare *LiPo-volt tester* som ger utslag om spänningen faller alltför lågt. Virkoret matas med en svart och röd kabel som sticker ut baktill på virkoret, kontakten på denna kabel kopplas till servonäs matningsspänning som sitter undertill den svarta raspberry pi hållaren. Raspberryn matas med hjälp av en mikro-usb kontakt som sitter fast baktill på hållaren precis vid ingången på Raspberry Pi. Slutligen startas systemet genom att slå på spänningen med hjälp av en strömbrytare på sidan av roboten. Det kan vara lämpligt att kontrollera att alla komponenter är korrekt inkopplade på kortet vid uppstart. För att data från sensorer och UART ska skickas måste de vara inkopplade på kortet enligt figur 1.



Figur 1: Bild av robot ovanifrån.

1. Sensoranslutningar.
2. UART-anslutning.
3. RESET-knapp (den röda, den blå är ej inkopplad).
4. AUTONOMT-läge-knapp.

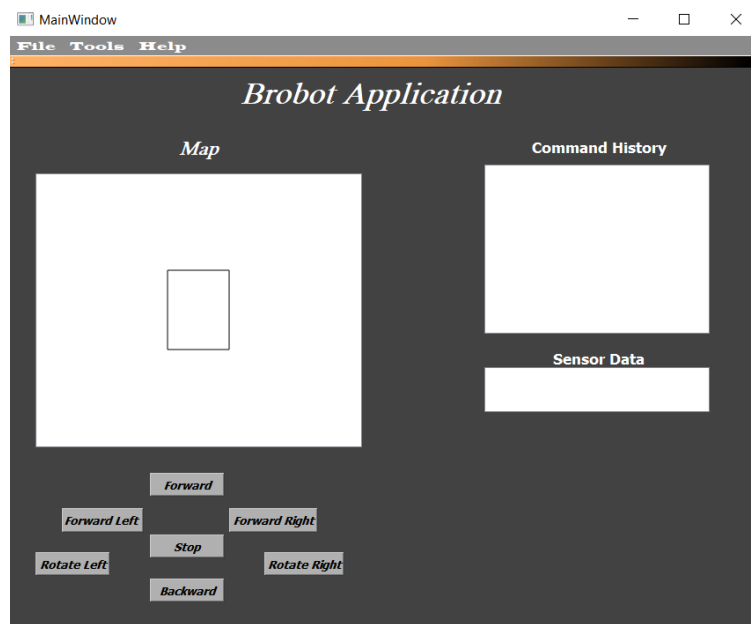
Den svarta knappen (nummer 4 på bilden) byter mellan manuellt och autonomt läge. Vid start är manuellt läge aktivt. Den röda knappen (nummer 3 på bilden) startar om sensor och styrenhet och kan användas för att stoppa roboten. Vid nedtryckning av RESET-knapp ställer sig roboten i sitt startläge och står still.

3.1 Start av autonomt läge

Kommunikationsmodulens Raspberry Pi kör vid start ett skript som väntar på kommando från GUI eller att man ska klicka på den svarta knappen på virkörtet. I autonomt läge så kör Raspberryn sitt program för att lösa en labyrint, för att roboten ska fungera korrekt så måste roboten först ställas i en godkänd startposition innan autonomt läge startas. En godkänd startposition har väggar runt om robotens sidor. När roboten satts i ett godkänt startläge och autonomt läge startats så börjar Raspberryn att hämta sensordata från sensormodulen för att göra val om hur roboten ska gå.

3.2 Start av manuellt läge

För att enkelt kontrollera roboten så har användaren en laptop med robotens medföljande GUI. När Raspberryn startar så sänder den ut en wifi-signal med namnet Brobot, genom att ansluta laptopen till detta wifi med lösenordet FuktigaGrupp10 och sedan starta GUI så skapas en socket som agerar kontakt mellan Raspberry och GUI. När kontakt är upprättat så kan användaren börja att skicka kommandon från GUI. Från manuellt läge är det även möjligt att byta till Autonomt läge i GUI, nedan följer en bild på hur GUI ser ut och dess funktionalitet.



Figur 2: GUI för systemet

1. Map: Visar en bild på robotens läge.
2. Styrknappar: Varje knapp svarar mot ett rörelsekommando för roboten
3. Kommandohistorik: Här kan man se vilka tidigare kommandon som har givits till roboten.
4. Sensordata: Visar mätvärden från robotens sensorer.

Tabellen nedan visar vad varje knapp motsvarar för funktionalitet.

Kommando	Förklaring
FORWARD	Rakt framåt
FORWARD-LEFT	Diagonalt åt vänster
FORWARD-RIGHT	Diagonalt åt höger
BACKWARDS	Rakt bakåt
TURN-RIGHT	Rotation 90 grader åt höger
TURN-LEFT	Rotation 90 grader åt vänster

Appendix

A Banspecifikation

Banan ska bestå av en enkel labyrint med kartongväggar. Avståndet mellan väggarna är 80 cm. Banan har endast 3-vägs korsningar med återvändsgränder. Återvändsgränderna är 80 cm djupa. Banan rymmer inom ett område på maximalt 10*10 meter. Roboten startar i ett hörn och målet är när den kommit ur labyrinten, dvs. när den inte har några väggar kring sig varvid den ska stanna.