



Laboration 1

Del a:

a1: Skriv följande drivrutiner i Matlab för kommunikation med Khepera-roboten. För ytterligare information se dokumentet "Drivrutiner för seriekommunikation".

comm_open

```
function status = comm_open(port_id, baud, timeout, debugg)
% function comm_open(port_id, baud, timeout, debugg)
%
% port_id - The communication port to be used. 0 for COM1, 1 for COM2.
% baud - Set the transfer speed. E.g: 9600.
% timeout - No. Of seconds before the read routines timeout.
% debugg - Level of debugg printouts. 0 for no printouts.
```

De tre första parametrarna enligt kopen. debugg kan sätta en global variabel debugga som kan användas i de andra rutinerna för felsökning. comm_open anropar kopen och definierar då en globala variabel port som kan användas i de andra rutinerna. comm_open kan börja med att anropa comm_close för att stänga en eventuellt öppnad linje.

De Kheperor som används under kursen är inställda på högsta kommunikationshastighet: **38400 baud**.

comm_close

```
function status = comm_close
% Close the communication port.
```

Använder den globala variabeln port.

OBS: Om kclose anropas med en icke-initialiserad port-variabel måste datorn bootas om. Testa med ~isempty(port) innan kclose anropas.

KhepCom

```
Function [ok, values] = KhepCom(cmd, arg, N, turretID)
% Sends a command cmd with arguments arg to the Khepera robot.
% Retries at most 5 times or until N numbers are read correctly from the
% Khepera. The read values are returned in values.
% A vector with N NaN is returned if unsuccessful.
% ok is returned == 1 iff succesful
% Example 1: ok = KhepCom('D', [12 15], 0) % Set speed
% Example 2: [ok, values] = KhepCom('O', [], 8) % Read ambient light sensors
% Example 3: [ok, values] = KhepCom('N', [], 64, 2); % Read K213
%
% the turretID parameter is used to send commands to turrets instead of to
% the main Khepera. Refer to the T command for details
```

a2:

När *KhepCom* är skriven och avtestad är följande rutiner väldigt enkla att skriva (1 rad kod var):

```
function ok = set_speed(speed)
%   Set motor speed. The unit is the pulse, corresponding to 8 mm/s
%
%   speed - 1x2 vector containing: the speed for the left motor,
%                                   the speed for the right motor
%   See the function D in the Khepera User Maunual for details

function [values, ok] = read_prox
%   Read the 10 bit values from the proximity meter sensors.
%   A vector with 8 NaN are returned if unsuccessful
%   See the function N in the Khepera User Maunual for details

% function [values, ok] = read_amb
%   Read the 10 bit values from the ambient light meter sensors.
%   A vector with 8 NaN are returned if unsuccessful
%   See the function O in the Khepera User Maunual for details

% function [values, ok] = read_vision
%   Read the 64 8 bit values from the K213 camera turret.
%   A vector with 64 NaN are returned if unsuccessful
%   See the function N in the Khepera User Maunual for details
```

Slutligen: Skriv ett test-program som använder de utvecklade rutinerna.

För att visa kamerabilden grafiskt kan man använda rutinen *imshow*. Tex:

```
[image, ok] = read_vision;
figure(2)
xmul=30;ymul=6;
imshow( repmat( reshape( repmat( image', ymul, 1), 64*ymul, 1), 1, xmul), [0 255] );
```

repmat –anropet duplicerar *image* i y- och x-led så att den blir lagom stor.

Allmänna tips:

- `sscanf(str, ',%d')` kan användas för att extrahera komma-separerade heltal från *str*.
- Använd *try* och *catch* för smidig felhantering i Matlab.
- Testa rutiner *KhepCom* ordentligt med korrekta och felaktiga simulerade inläsningar från Kheperan. Testa sedan med roboten inkopplad.
- Använd debuggern flitigt. Sätt en breakpoint där data ska *ksend* anropas för att kontrollera formatet. Stega fram och kontrollera den sträng som *ksend* returnerar och se om dess format stämmer med det du förutsätter. Följ strängen uppåt i avkodningen.
- Kom ihåg att rutinerna måste fungera oavsett vad Kheperan returnerar. Och även i det fall då ingenting returneras. Korrekt timeout-hantering är mycket viktig.

Del b:

Implementera en "Breitenberg vehicle" i Matlab för Kheperan.
Eftersom vi har 8 reflex-givare krävs en något mer komplicerad funktion som avbildar stimuli till respons. Använd en enkel perceptron med linjär aktiveringsfunktion för varje motor:

$$m_L = \sum_{i=1}^8 w_i \cdot r_i + w_0$$

$$m_R = \sum_{i=1}^8 v_i \cdot r_i + v_0$$

w_0 och v_0 är en "offset" som gör att roboten rör på sig även om inga signaler kommer från reflexgivarna (dvs. inga hinder finns i närheten). Det kan vara lämpligt att lägga in en förstärknings-faktor (omkr. 1/400) som påverkar alla vikter för att enkelt kunna förändra farten på roboten samt för att få hanterbar storlek på vikterna.

Programmet kan använda rutinerna *comm_close*, *comm_open*, *read_prox* samt *set_speed* från lab-del a.

Experimentera med olika värden på viktvektorerna för att få beteendena:

1: "Undvik hinder";

2: "Följ väggen" : roboten ska kunna gå genom en labyrinth genom att hela tiden följa tex. vänsterväggen.

Kan symmetrin vänster/höger användas för att förenkla inställningen av vikter?

Kan man kalibrera vikterna genom att "visa" roboten hur man står nära en vägg?

Kan du hitta på andra sätt?

Mer information om Braitenberg's vehicles:

<http://www.mindspring.com/~gerken/vehicles>

Rapport:

Normal CS-rapport med källkod bifogad som appendix.

Funktionen på del b besiktigas av en annan grupp (som inte er grupp besiktigar).

Efter godkänd besiktning bifogas testprotokollet till labrapporten.

Kom ihåg Namn och email-address på försidan !

Besiktningssprotokoll

Laboration 1 kurs AI2

för gruppen

.....

Testpunkter:

1.

Verifiera del b:

1: "Undvik hinder"

2: "Följ väggen"

2.

Koppla loss sladden till nätadaptern.

Roboten ska nu stanna. (om inte: kontakta genast mig samt TV4)

Sätt sedan tillbaka sladden.

Roboten ska nu återgå till normal funktion.

Om sladden varit bortkopplad för länge ska programmet ha "tattmat ur" och stoppat.

Vi har genomfört ovanstående tester med idel godkända resultat

.....

namnförtydligande: