1 Hermes

Zur Kommunikation zweier getrennter ROS-Systeme wird eine zusätzliche UDP-Verbindung verwendet. Deren Verwendung und Aufbau wird im folgenden erläutert.

1.1 Hermes Aufbau

Bei einer TCP/IP-Verbindung muss jedes Paket ankommen, dass verzögert den Datenaustausch. Im Gegensatz dazu erlaubt es die UDP/IP-Verbindung nur die aktuellen Pakete zu lesen. Deshalb ist nur die UDP/IP-Verbindung für eine Echtzeit-Kommunikation geeignet und wird auch in diesem Fall angewendet. Siehe Abbildung 1, Abbildung 2 und Anhang Seite 10.

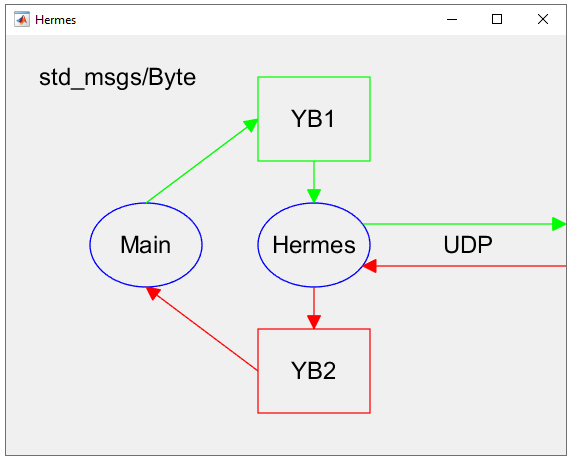

Abbildung 1: Hermes MATLAB figure.

Beim Ausführen des Hermes Scriptes wird eine Figure geöffnet. Diese dient dem Benutzter als Hilfe zum Verständnis und um das Programm kontrolliert zu beenden, mittels schließen diese Fensters. In der Abbildung 1 ist diese Figure abgebildet. In den quadratischen Boxen sind die ROS Topics dargestellt, in den Kreisen die MATLAB Programme, Hermes.m und Hauptprogramm (Main.m). Rechts ist die UDP-Verbindung angedeutet. Diese UDP-Verbindung sendet Daten von dem Hermes Script des eigenen youBots und empfängt die des anderen youBots.

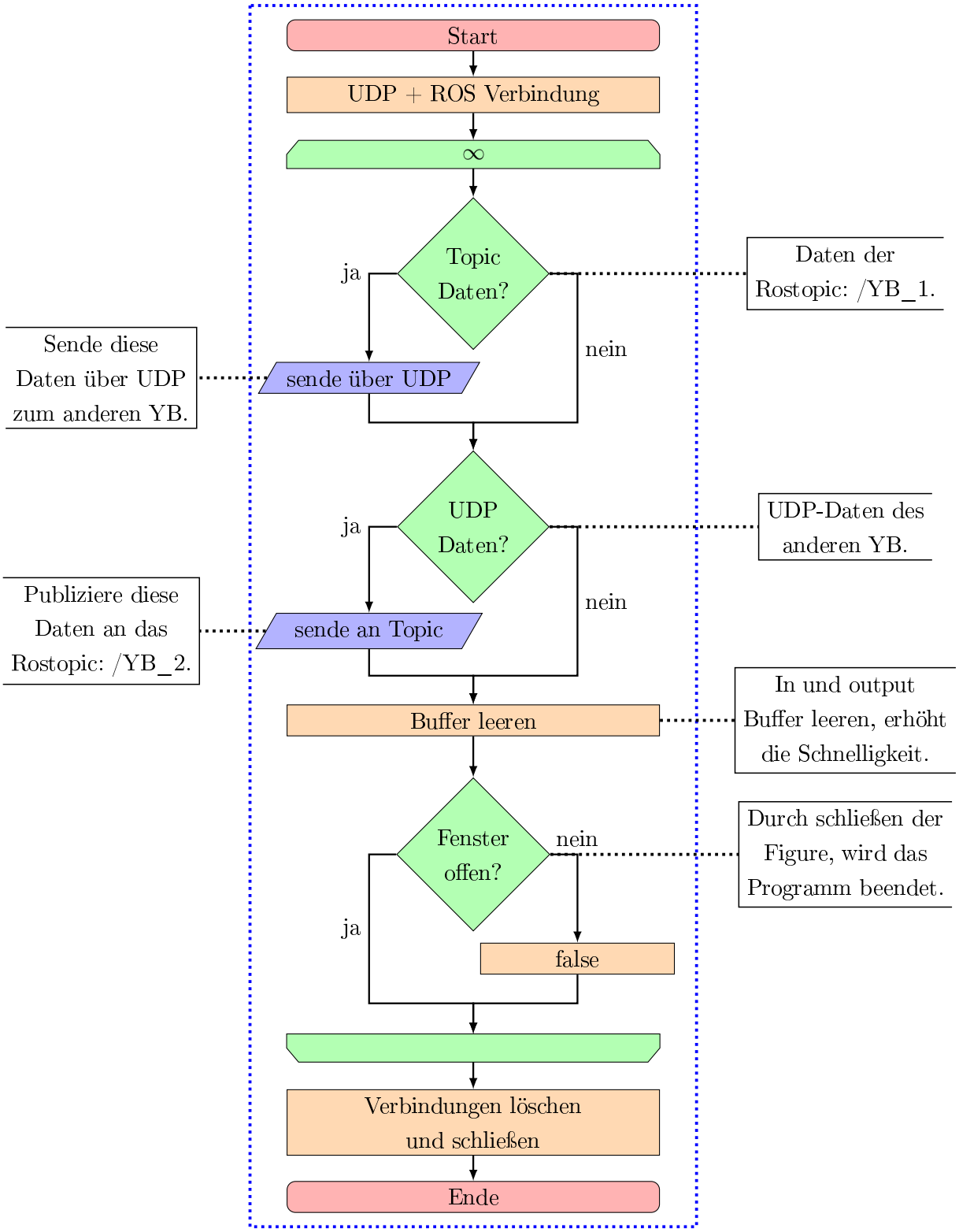

Abbildung 2: PAP des Datenaustausches.

1.2 Timer Funktionen

- **Kommunikation mit Hermes:**
  Teil des Hauptprogrammes welches die Kommunikation mit Hermes bereitstellt. Wird durch den Timer alle X-Sekunden aufgerufen, siehe Anhang Seite 8.

- **Daten aus Hermes lesen:**
  Teil des Hauptprogrammes welches die gesendeten Daten aus Hermes liest, siehe Anhang Seite 9.

Anhang

Initialisierung

```
% RSY Projekt Tyr, Initialisierung.

%===================================/\===================================
% Westfaelische Hochschule - FB Maschinenbau
% Labor fuer Mikroelektronik und Robotik
%-------------------------------------------------------------------------
% Martin Kondring
% Sebastian Flores
% Karsten Flores
%-------------------------------------------------------------------------
% Initialisierung.m
% Erst Erstellung : 8.01.2019
%-------------------------------------------------------------------------
% Teil des Hauptprogrammes welches alle Verbindungen startet und Variablen
% setzt sowie den Timer erstellt und startet.
%===================================\/===================================

%% ROS starten und alle Verbindungen erstellen
Erstellung = true;
while Erstellung == true
    if ~exist('ROS','var')
        ROS = runROS();
    else
        fprintf('ROS wurde bereits erstellt, es wurden keine neue Verbindung angelegt. \n');
        Erstellung=false;
    end
```

```
27 end
28
29 %% Timer erstellen und konfigurieren
30 % Timer starten, dieser sendet immer die letzte Nachricht an Hermes
31 delete(timerfindall); % Loescht evtl. bestehende Timer
32 KomTimer = timer;
33 KomTimer.TimerFcn = @KomFcn;
34 KomTimer.Period = 0.5;
35 KomTimer.ExecutionMode = 'fixedRate';
36 KomTimer.UserData.send = '00000000';
37 KomTimer.UserData.receive = '00000000';
38 KomTimer.UserData.ROS = ROS;
39 start(KomTimer);
40
41 %% Variablen initialisieren
42 % Konstanten
43 Abstand = 30;                          % Dicke Kollisionsebene [mm]
44 AbstandPrePos = Abstand + 10;          % Abstand der VorPos zur Kollisionsebene [mm]
45 AbstandRefPunkt = 120;                 % Abstand beider Symbole zum Referenzpunkt des YB-Arms [mm]
46 AbstandObj = 20;                       % Sicherheitsabstand vom Objekt in Z-Achse [mm]
47 minAbstandGehaeuse = 275;              % Abstand vom Gehaeuse zur Schonnung der Kamera [mm]
48 epsilon = 0.01;                       % Toleranz zwischen Achswinkel |Soll-Ist| [rad]
49 Schrittweite = 0.1;                   % max. Schrittweite bei der linearen Interpolation [mm]
50 run = ' ';                            % Hauptschleifen Laufbedingung
51 Master = false;                       % Teilnehmer Rang (false = Slave, true = Master)
52 ROS.Debug.Bild = true;                % Wenn true -> Ausgabe Livebild
53 ROS.Debug.Komm = true;                % Wenn true -> Kommunikation vorgaugeln
54 GreiferAuf = 20;                      % Greifer ist offen [mm]
55 GreiferZu = 7;                        % Greifer soweit zusammen, dass das Objekt gegriffen wird [mm]
56 PosKameraYB = [0, 0, 500, -pi/4, 0];  % Postition des Armes um den anderen YouBot zu erkennen [mm, mm, mm, rad, rad]
57
58 % Winkel des Armes um das Objekt und den Ablagenort des Objekts zu erkennen
```

```
59 if strcmp(ROS.User.Name, 'youbot-02') % in Abhaengigkeit des YouBots-Namen (Rechts und Links vom YouBot greifen)
60     WinkelKameraObj = [90, 0, 0, 0, 0];
61 elseif strcmp(ROS.User.Name, 'youbot-03')
62     WinkelKameraObj = [-90, 0, 0, 0, 0];
63 else
64     error('Kameraposition fuer diesen YB nicht bekannt!')
65 end
```

Master-Slave Entscheidung

```
1  % RSY Projekt Tyr, Master Slave Entscheidung.
2
3  %==================================/\====================================
4  % Westfaelische Hochschule - FB Maschinenbau
5  % Labor fuer Mikroelektronik und Robotik
6  %-------------------------------------------------------------------------
7  % Martin Kondring
8  % Sebastian Flores
9  % Karsten Flores
10 %-------------------------------------------------------------------------
11 % Master_Slave_Entscheidung.m
12 % Erst Erstellung : 7.01.2019
13 %-------------------------------------------------------------------------
14 % Teil des Hauptprogrammes welches entscheidet ob dieser youBot Slave oder
15 % Master ist.
16 %==================================\/====================================
17
18 %% Kameraposition Objekterkennung anfahren
19 fprintf('Kameraposition Objekterkennung anfahren. \n')
```

```
GelenkPos(ROS,WinkelKameraObj); % Anfahren der Kameraposition, um das Objekt zu erkennen
fprintf('Suche Objekt! \n')

%% Bildaufnahme auswerten ob Master oder Slave
while 1
    % Senswert anpassen um gegebenfalls andere Kreise auszuschliessen
    Ausgabe = KreisErkennung(ROS,'w','1',21,'Dtol',1,'Sens',0.7);
    if length(Ausgabe) == 1 && ~isempty(Ausgabe(1).X)
        try
            % Anpassung so das das Objekt besser gegriffen wird
            if strcmp(ROS.User.Name, 'youbot-02') % in Abhaengigkeit des YouBots-Namen (Rechts und Links vom YouBot greifen)
                Obj_Pos = [(-Ausgabe.X) (Ausgabe.Y) (Ausgabe.Z) pi 0];
            elseif strcmp(ROS.User.Name, 'youbot-03')
                Obj_Pos = [(Ausgabe.X) (-Ausgabe.Y) (Ausgabe.Z) pi 0];
            else
                error('Kameraposition fuer diesen YB nicht bekannt!')
            end
            % minimaler Abstand vom Gehaeuse zur Schonnung der Kamera
            if abs(Obj_Pos(2)) > abs(minAbstandGehaeuse)
                pre_Obj_Pos = Obj_Pos;
                % vor position nach oben verschieben
                pre_Obj_Pos(3) = pre_Obj_Pos(3) + AbstandObj;

                pfad_theta_ges_temp=calc_winkel(pre_Obj_Pos);
                pre_Obj_Pos_pfad_theta=kontrolle(pfad_theta_ges_temp);
                fprintf('Objekt erkannt. Position betraegt: X: %.3f, Y: %.3f, Z: %.3f. \n',Obj_Pos(1), Obj_Pos(2), Obj_Pos(3));
                Master = true;
                % Senden das ich Master bin
                KomTimer.UserData.send = '00000001';
                fprintf('Dieser youBot ist jetzt Master. \n')
                break;
            else
```

```
                warning('Objekt zu nah am Gehaeuse!')
            end
        catch
            warning('Objekt ausserhalb des Greifbereiches!')
        end
    else
        % Abfrage, bin ich Slave? Wenn ja break
        lauschen = readHermes(ROS);
        if strcmp(lauschen,'00000001')
            fprintf('Kein Objekt gefunden. Anderer youBot hat ein Objekt erkannt. Dieser youBot ist jetzt Slave. \n')
            Master = false;
            break;
        end
    end
end
pause(1);
```

Kommunikation mit Hermes

```
% RSY Projekt Tyr, Daten aus Hermes lesen.

%==================================/\====================================
% Westfaelische Hochschule - FB Maschinenbau
% Labor fuer Mikroelektronik und Robotik
%--------------------------------------------------------------------------
% Martin Kondring
% Sebastian Flores
% Karsten Flores
%--------------------------------------------------------------------------
```

```
% KomFcn.m
% Erst Erstellung : 11.12.2018
%--------------------------------------------------------------------------
% Teil des Hauptprogrammes welches die Kommunikation mit Hermes
% bereitstellt. Wird durch den Timer alle X-Sekunden aufgerufen
%==================================\/====================================


function KomFcn(timer, ~)
timer.UserData.ROS.Hermes.Nach.Data = bin2dec(timer.UserData.send);
send(timer.UserData.ROS.Hermes.Pub,timer.UserData.ROS.Hermes.Nach);
end
```

Daten aus Hermes lesen

```
% RSY Projekt Tyr, Daten aus Hermes lesen.

%==================================/\====================================
% Westfaelische Hochschule - FB Maschinenbau
% Labor fuer Mikroelektronik und Robotik
%--------------------------------------------------------------------------
% Martin Kondring
% Sebastian Flores
% Karsten Flores
%--------------------------------------------------------------------------
% readHermes.m
% Erst Erstellung : 11.12.2018
%--------------------------------------------------------------------------
% Teil des Hauptprogrammes welches die gesendeten Daten aus Hermes liest
```

```
%==================================\/====================================

function receiveData = readHermes(ROS)
try
    if ~ROS.Debug.Komm
        SubData = receive(ROS.Hermes.Sub, 1);
        tempReceive = dec2bin(SubData.Data);
    else
        sBuffer = input("Nachricht vom anderen youBot [0, 1, 3, 4, 9, 16]: ",'s');
        tempReceive = dec2bin(str2double(sBuffer));
    end
    tempStr = '';
    for i=1:8-length(tempReceive)
        tempStr = strcat(tempStr,'0');
    end
    receiveData = strcat(tempStr,tempReceive);
catch
    receiveData = "00000000";
end
end
```

Hermes

```
%Hermes Daten mittels UDP schreiben und lesen

%==================================/\====================================
% Westfaelische Hochschule - FB Maschinenbau
% Labor fuer Mikroelektronik und Robotik
```

```
%--------------------------------------------------------------------------
% Karsten Flores
% Sebastian Flores
%--------------------------------------------------------------------------
% Hermes.m
% Version vom 10.12.2018
%--------------------------------------------------------------------------
% Daten Byte Austausch der YouBots
% - /ComByte_YB2       Kommunikationsbyte vom anderen YB (empfangen UDP)
% - /ComByte_YB1       Kommunikationsbyte vom eigenen YB (senden UDP)
%--------------------------------------------------------------------------
% im Terminal starten
% cd Dokumente/MATLAB/Grundbefehle
% /usr/local/MATLAB/R2018b/bin/matlab -nodesktop -nosplash -r Hermes
%====================================\/====================================
warning off;
%====================================/\====================================
%                                 ROS Init
%====================================\/====================================
rosshutdown;
rosinit('NodeName','Hermes');
Publisher=rospublisher('/YB2','std_msgs/Byte'); % an Main
Nachricht=rosmessage(Publisher);
Subscriber=rossubscriber('/YB1','std_msgs/Byte'); % von Main

%====================================/\====================================
%                                 Fenster
%====================================\/====================================
fontsz = 18;
HermesFenster=figure('Name','Hermes','NumberTitle','off','MenuBar','none');
textcreate(0.45,0.7,0.2,0.2,'YB1',fontsz,'center','middle','gre');
textcreate(0.45,0.1,0.2,0.2,'YB2',fontsz,'center','middle','red');
```

```
38 textcreate(0.55,0.5,0.0,0.0,'Hermes',fontsz,'center','middle','tra');
39 ellipsecreate(0.45,0.4,0.2,0.2,'blu');
40 textcreate(0.25,0.5,0.0,0.0,'Main',fontsz,'center','middle','tra');
41 ellipsecreate(0.15,0.4,0.2,0.2,'blu');
42 arrowcreate(0.55,0.4,0.55,0.3,'red');
43 arrowcreate(0.45,0.2,0.25,0.4,'red');
44 arrowcreate(1,0.45,0.637,0.45,'red');
45 arrowcreate(0.55,0.7,0.55,0.6,'gre');
46 arrowcreate(0.25,0.6,0.45,0.8,'gre');
47 arrowcreate(0.637,0.55,1,0.55,'gre');
48 textcreate(0.825,0.5,0.0,0.0,'UDP',fontsz,'center','middle','tra');
49 textcreate(0.2,0.9,0.0,0.0,'std\_msgs/Byte',fontsz,'center','middle','tra');
50 drawnow
51
52 %==================================/\====================================
53 %                                  UDP
54 %==================================\/====================================
55 ipS = '192.168.0.20'; % Ip des anderen PCs
56 % Verbindungsinfos
57 portS = 9091;
58 portM = 9190;
59 % Verbinden
60 udpVerbindung = udp(ipS,portS,'LocalPort',portM);
61 udpVerbindung.InputBufferSize=100;
62 fopen(udpVerbindung);
63
64 %==================================/\====================================
65 %               Loop der Daten lesen, schreiben und publishen
66 %==================================\/====================================
67 run=true;
68 while run
69     % eigene Info auslesen und senden
```

```
    try
        Sub_Data=receive(Subscriber,0.1);
        fprintf(udpVerbindung, '%s', num2str(Sub_Data.Data));
    catch
        disp('ROS no data')
    end
    % andere Info empfangen und publishen
    try
        if udpVerbindung.BytesAvailable>0
            UDP_Data=fscanf(udpVerbindung);
            Nachricht.Data = str2double(UDP_Data);
            send(Publisher,Nachricht);
        end
    catch
        disp('UDP no data')
    end

    % in und output buffer loeschen
    flushinput(udpVerbindung);
    flushoutput(udpVerbindung);

    % falls Fenster geschlossen wird, run=false
    if ishandle(HermesFenster)==0
        run=false;
    end
    pause(0.0001);
end

%==================================/\==================================
%                          ENDE der Verbindungen
%==================================\/==================================
fclose(udpVerbindung);
```

```
delete(udpVerbindung);
rosshutdown;

disp('---Flores---')
% exit();

%==================================/\==================================
%                            Fenster Funktion
%==================================\/==================================
function textcreate(x,y,v,w,txt,fsize,HorAli,VerAli,theme)
switch theme
    case 'red'
        color.edge=[1 0 0];
        color.back=[0.94 0.94 0.94];
        color.font=[0 0 0];
    case 'gre'
        color.edge=[0 1 0];
        color.back=[0.94 0.94 0.94];
        color.font=[0 0 0];
    case 'blu'
        color.edge=[0 0 1];
        color.back=[0.94 0.94 0.94];
        color.font=[0 0 0];
    case 'tra'
        color.edge=[0.94 0.94 0.94];
        color.back=[0.94 0.94 0.94];
        color.font=[0 0 0];
    otherwise
end
annotation('textbox',...
    [x y v w],...
    'String',txt,...
```

```
    'FontSize',fsize,...
    'FontName','Arial',...
    'HorizontalAlignment',HorAli,...
    'VerticalAlignment',VerAli,...
    'LineStyle','-',...
    'EdgeColor',color.edge,...
    'LineWidth',1,...
    'BackgroundColor',color.back,...
    'Color',color.font);
end

function ellipsecreate(x,y,v,w,theme)
switch theme
    case 'red'
        color.edge=[1 0 0];
    case 'gre'
        color.edge=[0 1 0];
    case 'blu'
        color.edge=[0 0 1];
    otherwise
end
annotation('ellipse',...
    [x y v w],...
    'LineStyle','-',...
    'LineWidth',1,...
    'Color',color.edge);
end

function arrowcreate(x1,y1,x2,y2,theme)
switch theme
    case 'red'
        color.arrow=[1 0 0];
```

```
        line.style = '-';
        line.width = 1;
        line.head = 'plain';
    case 'gre'
        color.arrow=[0 1 0];
        line.style = '-';
        line.width = 1;
        line.head = 'plain';
    otherwise
end
annotation('textarrow',...
    [x1 x2],[y1 y2],...
    'LineStyle',line.style,...
    'LineWidth',line.width,...
    'HeadStyle',line.head,...
    'Color',color.arrow);
end
```