Getting Started with Matlab and the Dynamixel SDK

Let's light the LED on AX-12 ID #1, the "Hello World" demo for Matlab and the Dynamixel SDK.

You must have a USB2Dynamixel, Parallax Propeller running DynaBus Embedded, or comparable PC to AX-12 bus hardware interface.

This project contains six files.

- Led1On.m
- Led1Off.m
- Led1OnRegWrite.m
- SerialLed.m
- SyncWriteDemo.m
- DualDynamixel180Phase.m

To get the ball rolling, we need to register dynamixel.dll and dynamixel.h.

If you need more help, please post your questions on Agave Robotics community forum http://forum.agaverobotics.com/

Register dynamixel.dll and dynamixel.h

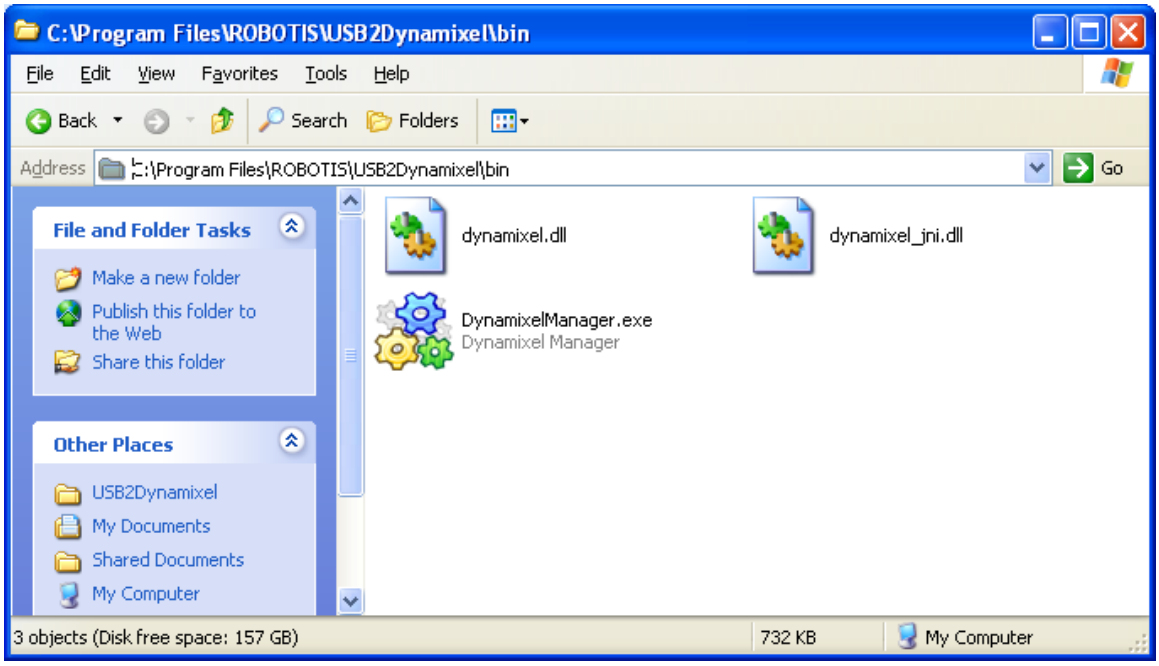
You must register dynamixel.dll and dynamixel.h in Matlab if you want to use the API in your Matlab project.

> The SDK is not required to control an AX-12 network! Check out SerialLed.m.

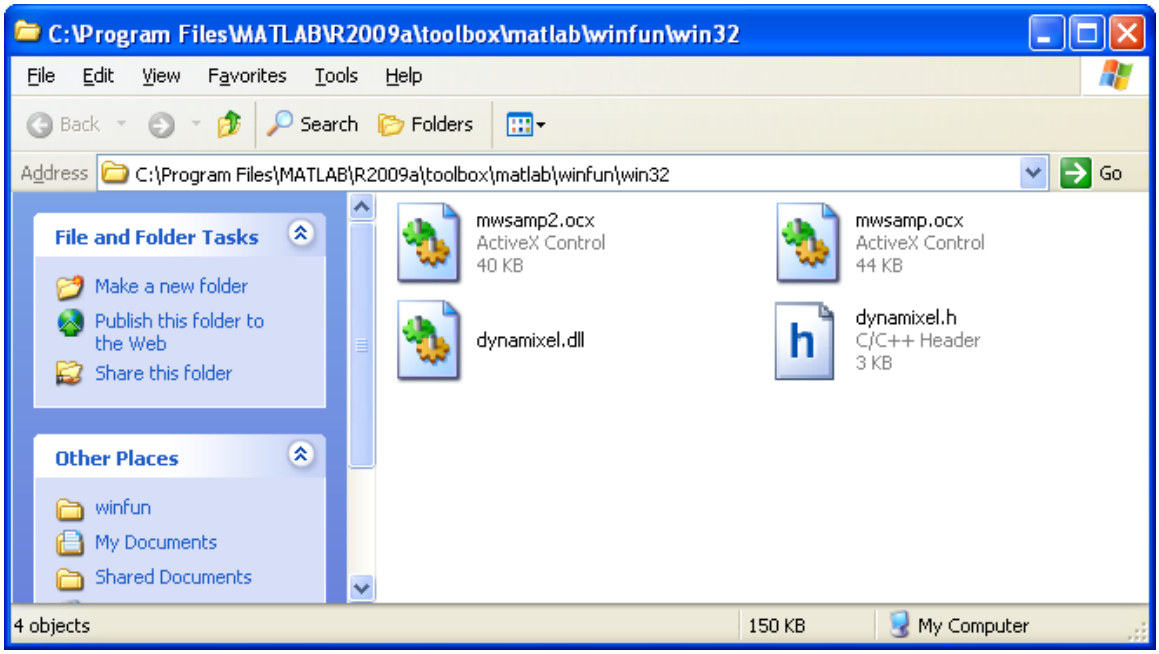
All you have to do is copy two files. This is not very clear in the SDK manual but here's how to do it.

This setup is for Windows with the default installations of the USB2Dynamixel SDK and MatLab.

SDK Source Location
C:\Program Files\ROBOTIS\USB2Dynamixel\bin\dynamixel.dll
C:\Program Files\ROBOTIS\USB2Dynamixel\import\dynamixel.h

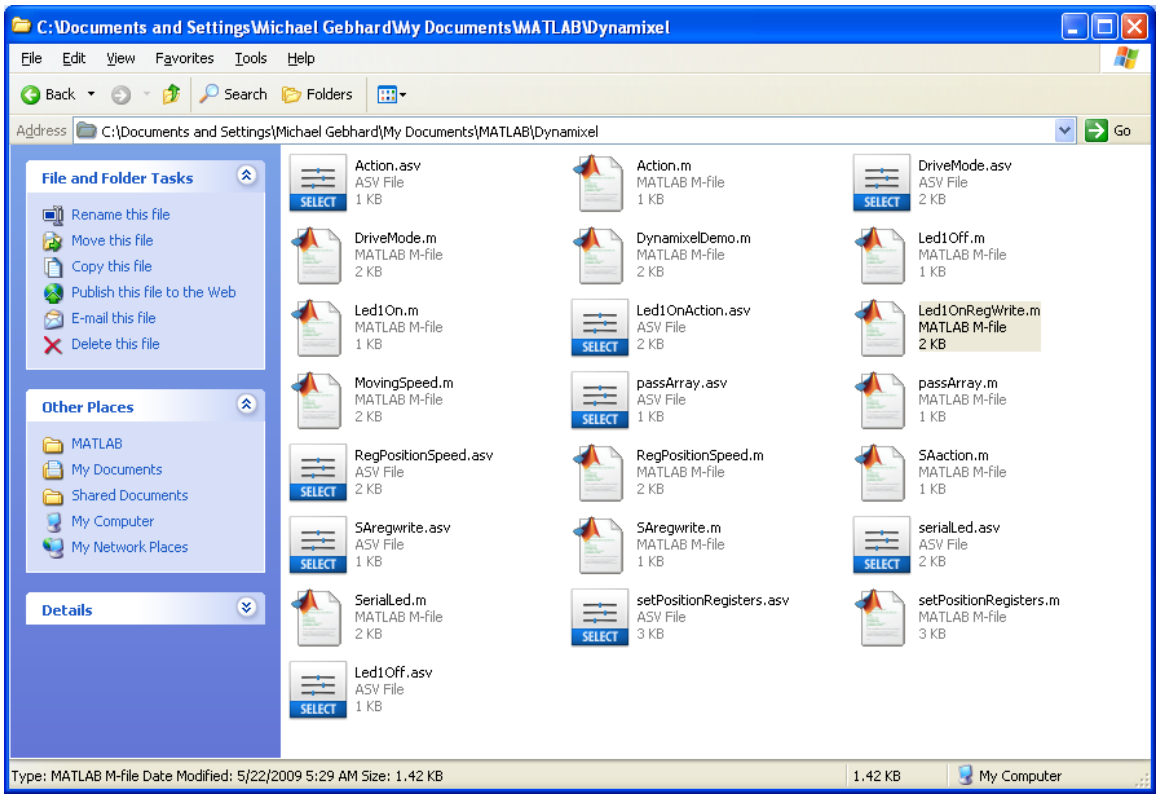


MatLab Destination folder
C:\Program Files\MATLAB\R2009a\toolbox\matlab\winfun\win32

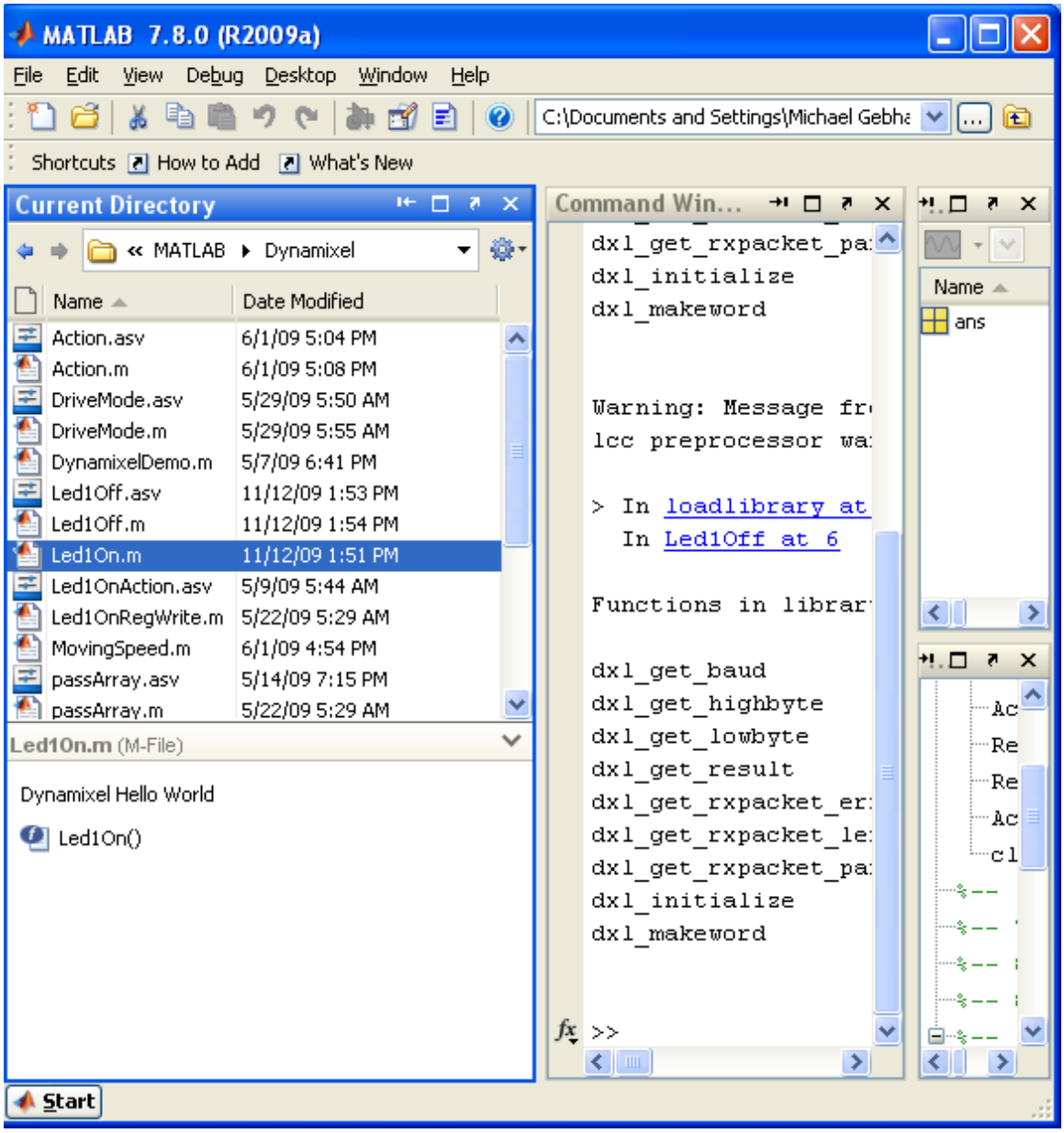


Running the files

To make life easy, copy the Dynamixel folder from this download to your Matlab folder. For Windows users, the Matlab folder is in the “My Documents\MATLAB” folder.



Now you can see the “Dynamixel” directory in your Matlab directory browser. Simply double click the file to open the source code in the Matlab editor.


MATLAB 7.8.0 (R2009a)
File Edit View Debug Desktop Window Help
C:\Documents and Settings\Michael Gebha
Shortcuts How to Add What's New
Current Directory
« MATLAB ▸ Dynamixel
Name Date Modified
Action.asv 6/1/09 5:04 PM
Action.m 6/1/09 5:08 PM
DriveMode.asv 5/29/09 5:50 AM
DriveMode.m 5/29/09 5:55 AM
DynamixelDemo.m 5/7/09 6:41 PM
Led1Off.asv 11/12/09 1:53 PM
Led1Off.m 11/12/09 1:54 PM
Led1On.m 11/12/09 1:51 PM
Led1OnAction.asv 5/9/09 5:44 AM
Led1OnRegWrite.m 5/22/09 5:29 AM
MovingSpeed.m 6/1/09 4:54 PM
passArray.asv 5/14/09 7:15 PM
passArray.m 5/22/09 5:29 AM
Led1On.m (M-File)
Dynamixel Hello World
Led1On()
Command Win...
dxl_get_rxpacket_pa
dxl_initialize
dxl_makeword
Warning: Message fr
lcc preprocessor wa
> In loadlibrary at
In Led1Off at 6
Functions in librar
dxl_get_baud
dxl_get_highbyte
dxl_get_lowbyte
dxl_get_result
dxl_get_rxpacket_er
dxl_get_rxpacket_le
dxl_get_rxpacket_pa
dxl_initialize
dxl_makeword
>>
Name
ans
Start

You might receive the following warning

```
Warning: Message from C preprocessor:
lcc preprocessor warning: C:\Program
Files\ROBOTIS\USB2Dynamixel\import\dynamixel.h:98 No
newline at end of file

> In loadlibrary at 371
  In Led1On at 6
```

Not to worry, just open the file specified in the warning, add a line return to the end of the file, and save the file.

```
C:\Program Files\ROBOTIS\USB2Dynamixel\import\dynamixel.h
```

Appendix A

MatLab source code

```
function Led1On()
    % Dynamixel Hello World
    % Light the LED on ID 1

    % setup the dynamixel API
    loadlibrary('dynamixel','dynamixel.h');
    libfunctions('dynamixel');
    res = calllib('dynamixel','dxl_initialize');

    if res == 1
        %dynamixel 1
        calllib('dynamixel','dxl_set_txpacket_id',1);
        %length = number of parameter + 2 (2 + 2)
        calllib('dynamixel','dxl_set_txpacket_length',4);
        %writing
        calllib('dynamixel','dxl_set_txpacket_instruction',3);
        %Build instruction parameters
        %Parameter 0 = Address
        calllib('dynamixel','dxl_set_txpacket_parameter',0, 25);
        %Parameter 1 = value
        calllib('dynamixel','dxl_set_txpacket_parameter',1, 1);
        %transmit
    calllib('dynamixel','dxl_tx_packet');
    else
        disp('Failed to open USB2Dynamixel!');
    end

    % clean up
    calllib('dynamixel','dxl_terminate');
    unloadlibrary('dynamixel');
end
```

```
function Led1Off()
    %Light LED1

    loadlibrary('dynamixel','dynamixel.h');
    libfunctions('dynamixel');
    res = calllib('dynamixel','dxl_initialize');

    if res == 1
        %dynamixel 1
        calllib('dynamixel','dxl_set_txpacket_id',1);
        %length = number of parameter + 2 (2 + 2)
        calllib('dynamixel','dxl_set_txpacket_length',4);
        %writing
        calllib('dynamixel','dxl_set_txpacket_instruction',3);
        %Build instruction parameters
        %Parameter 0 = Address
        calllib('dynamixel','dxl_set_txpacket_parameter',0, 25);
        %Parameter 1 = value
        calllib('dynamixel','dxl_set_txpacket_parameter',1, 0);
        %transmit
    calllib('dynamixel','dxl_tx_packet');
    else
        disp('Failed to open USB2Dynamixel!');
    end
    calllib('dynamixel','dxl_terminate');
    unloadlibrary('dynamixel');
end
```

```
function Led1OnRegWrite()

    loadlibrary('dynamixel','dynamixel.h');
    libfunctions('dynamixel');
    res = calllib('dynamixel','dxl_initialize');
    pause on

    if res == 1
        %dynamixel 1
        calllib('dynamixel','dxl_set_txpacket_id',1);
        %length = number of parameter + 2 (2 + 2)
        calllib('dynamixel','dxl_set_txpacket_length',4);
        %reg writing
        calllib('dynamixel','dxl_set_txpacket_instruction',4);
        %Build instruction parameters
        %Parameter 0 = Address
        calllib('dynamixel','dxl_set_txpacket_parameter',0, 25);
        %Parameter 1 = value
        calllib('dynamixel','dxl_set_txpacket_parameter',1, 1);
        %Transmit
        calllib('dynamixel','dxl_tx_packet');


        %Zero out previous parameters
        calllib('dynamixel','dxl_set_txpacket_parameter',0, 0);
        calllib('dynamixel','dxl_set_txpacket_parameter',1, 0);

        %Action Instruction
        calllib('dynamixel','dxl_initialize');
        calllib('dynamixel','dxl_set_txpacket_id',254);
        calllib('dynamixel','dxl_set_txpacket_length',2);
        calllib('dynamixel','dxl_set_txpacket_instruction',5);

        %Transmit
        calllib('dynamixel','dxl_tx_packet');
    else
        disp('Failed to open USB2Dynamixel!');
    end
    calllib('dynamixel','dxl_terminate');
    unloadlibrary('dynamixel');

    %Action();
end
```

```
function SerialLed()
   % This examples shows how to you can use native MatLab fucntions
   % to control an AX-12.  If have problems getting this to work,
   % make sure that the serial port is closed!

    % Set the port paramenter
    s=serial('COM3', 'BaudRate', 1000000, 'Parity', 'none', 'DataBits',
8, 'StopBits', 1);

    % open the port
    fopen(s);

    % display the com port resources
    com = instrfind;
    disp(com);

    %-------- [LED 1 On ] -------------------------------
    %FF FF 01 04 03 19 01 DD
    a = [255, 255, 1, 4, 3, 25, 01, 221];
    %-----------------------------------------------------

    %-------- [LED 1 Off ] ------------------------------
    %FF FF 01 04 03 19 00 DE
    %a = [255, 255, 1, 4, 3, 25, 00, 222];
    %-----------------------------------------------------

    % display the values in a
    disp(a)

    % binary write
    fwrite(s, a);

    % Expecting a 6 byte status packet
    out=fread(s, 6);

    % Display status packet
    disp(out);

    % Clean up
    fclose(s);
    delete(s);
    clear s;
end
```

```
function SyncWriteDemo()
    loadlibrary('dynamixel','dynamixel.h');
    %libfunctions('dynamixel');
    res = calllib('dynamixel','dxl_initialize');
    pause on

    numberOfDynamixels = 2;

    if res == 1

        for i = 1:numberOfDynamixels
            id(1,i) = i;
            phase(1,i) = (2 * pi) * i/numberOfDynamixels;
        end

        %Broadcast id 0xFE
        calllib('dynamixel','dxl_set_txpacket_id',254);

        %Length is 14
        %That handles position and speed for two dynamixels
        calllib('dynamixel','dxl_set_txpacket_length',14);

        %SyncWrite instruction 0x83
        calllib('dynamixel','dxl_set_txpacket_instruction',131);

        %Starting address
        calllib('dynamixel','dxl_set_txpacket_parameter',0, 30);

        %length of data to write to each dynamixel
        calllib('dynamixel','dxl_set_txpacket_parameter',1, 4);

        %Parameters for syncwrite dynamixel id = 1
        % id | position | speed
        %ID = 1
        calllib('dynamixel','dxl_set_txpacket_parameter',2, 1);

        %Position = 512
        lowByte = calllib('dynamixel','dxl_get_lowbyte',512);
        highByte = calllib('dynamixel','dxl_get_highbyte', 512);
        calllib('dynamixel','dxl_set_txpacket_parameter',3, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',4, highByte);

        %Speed = 512
        lowByte = calllib('dynamixel','dxl_get_lowbyte',512);
        highByte = calllib('dynamixel','dxl_get_highbyte', 512);
        calllib('dynamixel','dxl_set_txpacket_parameter',5, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',6, highByte);



        %Parameters for syncwrite dynamixel id = 2
        % id | position | speed
        %ID = 2
        calllib('dynamixel','dxl_set_txpacket_parameter',7, 2);
```

```
        %Position = 512
        lowByte = calllib('dynamixel','dxl_get_lowbyte',512);
        highByte = calllib('dynamixel','dxl_get_highbyte', 512);
        calllib('dynamixel','dxl_set_txpacket_parameter',8, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',9, highByte);

        %Speed = 512
        lowByte = calllib('dynamixel','dxl_get_lowbyte',512);
        highByte = calllib('dynamixel','dxl_get_highbyte', 512);
        calllib('dynamixel','dxl_set_txpacket_parameter',10, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',11, highByte);

        %transmit
        calllib('dynamixel','dxl_tx_packet');

    else
        disp('Failed to open USB2Dynamixel!');
    end
    calllib('dynamixel','dxl_terminate');
    unloadlibrary('dynamixel');
end
```

```
function DualDynamixel180Phase(id, theta, speed)
%{
    DualDynamixel180Phase will rotate two dynamixels
    in opposite directions.

    is is a two element array that contains the
    Dynamixel IDs

    theta is the angle in radians to move from center

    speed is the moving speed 0-1023

    DualDynamixel180Phase(id, sind(0), 256)

    Sample SyncWrite command
    FF FF FE 0E 83 1E 04 01 00 02 00 02 02 00 01 00 01 45
%}

    %Load the dynamixel library
    loadlibrary('dynamixel','dynamixel.h');
    response = calllib('dynamixel','dxl_initialize');
    pause on

    %There should only be 2 Dynamixels total
    numberOfDynamixels = length(id);

    if response == 1

        % Phase allows us to position the servos
        % in some relationship to theta.
        phase = zeros(1,2);
        for i = 1:numberOfDynamixels
            phase(1,i) = (2 * pi) * (i)/numberOfDynamixels;
        end

        goalPosition = zeros(1,2);
        % Convert theta + phase to goal position
        for i = 1:numberOfDynamixels
            goalPosition(1, i) = int16((sin(theta + phase(1,i)) + 1) *
512);
        end

        %Broadcast ID
        calllib('dynamixel','dxl_set_txpacket_id', 254);

        %Length is 14
        %That handles position and speed for two dynamixels
        calllib('dynamixel','dxl_set_txpacket_length',14);

        %SyncWrite instruction
        calllib('dynamixel','dxl_set_txpacket_instruction',131);

        %Starting address (goal position)
```

```
        calllib('dynamixel','dxl_set_txpacket_parameter',0, 30);

        %length of data to write to each dynamixel
        %We're writing position and speed = 4 bytes
        calllib('dynamixel','dxl_set_txpacket_parameter',1, 4);

        %Parameters for syncwrite
        % id | position | speed
        %ID
        calllib('dynamixel','dxl_set_txpacket_parameter',2, 1);

        lowByte = calllib('dynamixel','dxl_get_lowbyte',
goalPosition(1,1));
        highByte = calllib('dynamixel','dxl_get_highbyte',
goalPosition(1,1));
        calllib('dynamixel','dxl_set_txpacket_parameter',3, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',4, highByte);

        %Speed = 512
        lowByte = calllib('dynamixel','dxl_get_lowbyte', speed);
        highByte = calllib('dynamixel','dxl_get_highbyte', speed);
        calllib('dynamixel','dxl_set_txpacket_parameter',5, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',6, highByte);

        %Parameters for syncwrite dynamixel id = 2
        % id | position | speed
        %ID = 2
        calllib('dynamixel','dxl_set_txpacket_parameter',7, 2);

        %Position = 512
        lowByte = calllib('dynamixel','dxl_get_lowbyte',
goalPosition(1,2));
        highByte = calllib('dynamixel','dxl_get_highbyte',
goalPosition(1,2));
        calllib('dynamixel','dxl_set_txpacket_parameter',8, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',9, highByte);

        %Speed = 512
        lowByte = calllib('dynamixel','dxl_get_lowbyte', speed);
        highByte = calllib('dynamixel','dxl_get_highbyte', speed);
        calllib('dynamixel','dxl_set_txpacket_parameter',10, lowByte);
        calllib('dynamixel','dxl_set_txpacket_parameter',11, highByte);



        %transmit
        calllib('dynamixel','dxl_tx_packet');

    else
        disp('Failed to open USB2Dynamixel!');
    end
    calllib('dynamixel','dxl_terminate');
    unloadlibrary('dynamixel');
end
```