smovtoCAng

Simultaneously move all 3 different Herkulex motors to 3 different desired angles

Contents

Syntax

```
smovtoCAng(sObject,pID1,ang1,pID2,ang2,pID3,ang3,varargin)
```

Description

Controlling multiple motors simultaneously by extending the original single motor controlling's packet. Note that the angle is based on calibrated position.

- Packet requires 1 byte for Playtime and 4 bytes for each motor. Eg, optional data length of 41 bytes required if sending instructions to 10 motors simultaneously.
- Default playtime value: 60 (60*11.2ms = 672ms)
- For code to be executed properly, please follow the following convention of motor ID setting: Btm max ID, Mid intermediate ID, Top min ID. Eg, Btm pID: 253, Mid pID: 252, Top pID: 251.

Input Arguments

- sObject - serial port object
- pID1, pID2, pID3 - integer
- ang1, ang2, ang3 - integer
- varargin - integer

Function Codes

```
function smovtoCAng(sObject, pID1, CAng1, pID2, CAng2, pID3, CAng3,varargin)



    % Set default playtime if not provided

    if nargin == 7

        playtime = 672/11.2;     % Default playtime value 672ms/11.2ms = 60
```

```
        pTime = dec2hex(int64(playtime),2); % Value converted into hex for packet
    elseif nargin == 8
        V = cell2mat(varargin);  % Convert varargin into number
        playtime = V/11.2;  % Convert into value
        pTime = dec2hex(int64(playtime),2);
    else
        error('Please input only 7 to 8 arguments!');
    end


    % Initialize variables
    ang = [CAng1,CAng2,CAng3];
    pID = [pID1,pID2,pID3];
    CVal = [512,512,512];


    % Convert angles to respective position values
    for i=1:3
        CVal(i) = fix(512 + ang(i)/0.325);


        % Check input value range for top and btm motors
        if(pID(i)==max(pID) || pID(i)==min(pID))
            % Using general check value function
            checkCVal(sObject, pID(i), CVal(i));
```

```
        % Check middle motor specifically due to smaller range (middle bracket)
        else
            checkCVal(sObject, pID(i), CVal(i),252);
        end
    end


    % Convert values into hex for packet
    % Byte in reverse order by Little Endian Order
    % pos1: motor pID1, pos2: motor pID2, pos3: motor pID3
    pos1 = dec2hex(CVal(1),4);
    pos1 = strcat(pos1(3:4),pos1(1:2));
    pos2 = dec2hex(CVal(2),4);
    pos2 = strcat(pos2(3:4),pos2(1:2));
    pos3 = dec2hex(CVal(3),4);
    pos3 = strcat(pos3(3:4),pos3(1:2));


    data =
strcat([pTime,pos1,'04',dec2hex(pID1,2),pos2,'04',dec2hex(pID2,2),pos3,'04',dec2hex(pI
D3,2)]);
    packet = pkGen(254,06,data);
    inHkx(sObject, packet);
```

```
    % Wait for the operation to complete
    pause(1);


    % Confirm end positions of each motor
    for id = pID(1):pID(3)
        CPos = getCPos(sObject,id);
        CAng = fix((CPos-512)*0.325);
        fprintf('Motor %d at angle %d\n', id, CAng);
    end


end
```

Published with MATLAB® R2016a