Jason Esquivel
ECE 590
11/8/2016

Intro

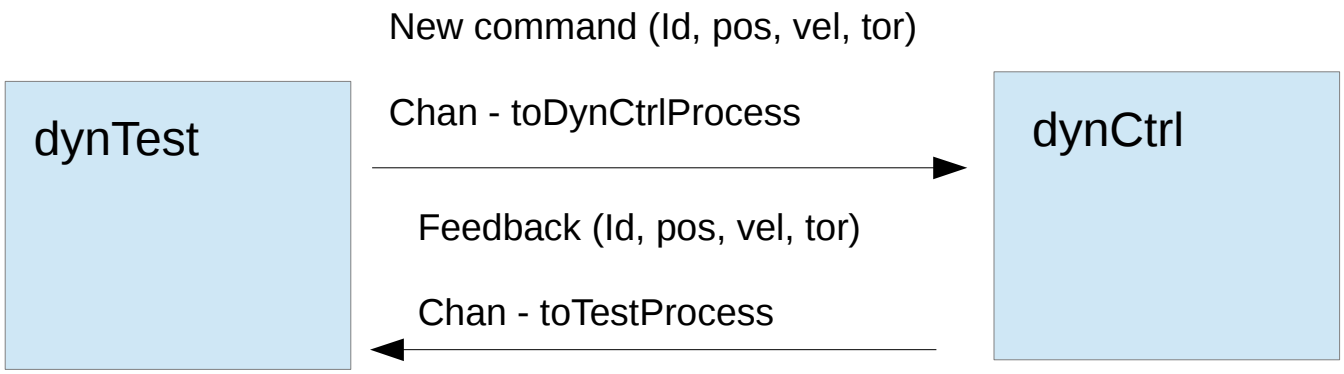
The purpose of this assignment is to control the position, velocity and torque of various servos. To get a feedback of position, velocity and torque from the servos. Two processes need to be run, “dynTest.py” will send new commands and receiving feedback data. “DynCtrol” will get the new commands and convert to servo packets through serial for servos. Will also periodically get a feedback from the servos about the current position, velocity and torque.

Tutorial

1) Connecting to computer – Check to make sure you can see your device is visible to the PC:
   -$ ls -oh /dev | grep USB

2) To make sure you have write permissions to your device use the following :
   -$ sudo chmod 777 /dev/ttyUSB0

3) Two channels need to be setup for communication between dynTest.py and dynCtrl.py.
   -$ ach -1 -C "toTestProcess" -m 20 -n 3000
   -$ ach -1 -C "toDynCtrlProcess" -m 20 -n 3000



4) To run dynCtrl.py- will wait for new commands from channel “toDynCtrlProcess” and will periodicallly send feedback through channel “toTestProcess”
   python dynCtrl.py

5) To run dynTest.py. - will run test file to send commands troough “toDynCtrlProcess” and receive feedback from “toTestProcess”
   python dynTest

Channel Communication

Packets sent over Channel in the following format.

Command Fields are as follows:

DYNDATAUPDATE = 100
CMDPOS = 101
CMDPOSVEL = 102
CMDPOSVELTOR = 103

Packet format over both channels:

```
class DYNDATA(Structure):
   _pack_ = 1
   _fields_ = [("err"    , c_int8),
               ("cmd" , c_int8),
               ("id"     , c_int8),
               ("pos"   , c_double),
               ("vel"   , c_double),
               ("tor"   , c_double)
                ]
```

Functions

Will Convert Position in degrees to 8 bit fields for servo packets.
Return : dynPosHigh Highest 8 bits of converted Value
dynPosLow Lowest 8 bits of converted Value

def degToDynPos(deg):

Will convert Servo position in bits to position in Degrees
Return position in Degrees

def dynPosToDeg(inPos):

Will Convert Velocity in rpm to 8 bit fields for servo packets.
Return : dynVelHigh Highest 8 bits of converted Value
dynVelLow Lowest 8 bits of converted Value

def rpmToDynVel(vel):

Will convert Servo Velocity in bits to velocity in RPM
Return velocity in RPM

def dynVelToRPM(inVel):

Will Convert torque in kgfcm to 8 bit fields for servo packets.
Return : dynTorHigh Highest 8 bits of converted Value
dynTorLow Lowest 8 bits of converted Value

def kgfcmToDynTorque(torque):

Will convert Servo Torque in bits to Torque in kgfcm
Return torque in kgfcm

def dynTorqueToKGFCM(inTorque):

Will get feedback data from Servo
Input- dynId Id of Servo data is wanted from.
Output inError – Any Error message from returned packet
inId - Id of servo where data is from
pos - current position in deg
vel - current velocity in rpm
tor - current torque in kgfcm

def fromDyn(dynId):

Overloaded function for new command to servos
Input dynId - Id of servo for new command
deg - new position in degrees for servo
vel - (optional) new velocity in rpm for servo
torque - (optional) new torque in kgfcm for servo
Returns None

def toDyn(dynId, deg, vel = None , torque = None):

Interrupt handler to send feedback data. Will call function fromDyn and send data to channel 'toTestProcess'

def fromDynInterruptHandler(signum, stack):