# simple script to use digital-to-analog outputs on LabJack U3 as an adjustable voltage source

import u3 # bring in Python library supplied by LabJack co.

import time

# Open the LabJack, create "device" called d

# This will cause an error if there is no LabJack U3 hardware attached.

d = u3.U3()

## get analog calibration data, for applying the proper calibration to readings.

d.getCalibrationData()

# find out which DAC the user wants. raw\_input gives string response, so we must convert it to integer.

dacstr = raw\_input("DAC to use? (0 or 1)" )

dac = int(dacstr)

raw\_input("Connect a wire from DAC%d to AIN0. Hit Enter when ready." % dac)

alphabet = ' abcdefghijklmnopqrstuvwxyz'

vlist = []

a = []

vstart = 0.2

vstep = 0.17

for char in "entropy conquers all":

position = alphabet.find(char)

voltage = vstart + (vstep \* position)

# resp = raw\_input("Desired output voltage? (0.0-5.0, negative to exit) ")

volts = float(voltage) # convert string to floating point number

if volts<0 : break # get out of loop if input is negative

voltval = d.voltageToDACBits(volts, dacNumber = dac, is16Bits = True) # convert voltage to binary value

# # put out a voltage on desired DAC

d.getFeedback(u3.DAC16(Dac = dac, Value = voltval))

# # wait a moment and read voltage on AIN0

time.sleep(.1)

newvolt=float(d.getAIN(0))

print "reading ", char, newvolt, " V"

vlist.append(newvolt)

phil = ((newvolt - vstart) / vstep)

pos = int(round(phil))

alphabet[pos]

a.append(alphabet[pos])

msg = ""

for c in a:

msg = msg + c

print msg

#while True: # will keep repeating until 'break' is encountered

# resp = raw\_input("Desired output voltage? (0.0-5.0, negative to exit) ")

# volts = float(resp) # convert string to floating point number

# if volts<0 : break # get out of loop if input is negative

# voltval = d.voltageToDACBits(volts, dacNumber = dac, is16Bits = True) # convert voltage to binary value

#

# # put out a voltage on desired DAC

# d.getFeedback(u3.DAC16(Dac = dac, Value = voltval))

#

# # wait a moment and read voltage on AIN0

# time.sleep(0.1)

# print "reading ", d.getAIN(0), " V"

# we land here when we break from while loop

d.close() # disconnect from LabJack

exit()