```
'''goals2democode.py

   Goals 2 Demonstration Code for the HEBI motors

   Please read through, fix the motor name, and add to the end.

'''

# Import useful packages
import hebi
import numpy as np              # For future use

from math import pi, sin, cos, asin, acos, atan2
from time import sleep, time


#
#  HEBI Initialization
#
# Create a lookup object to discover/find/connect to the HEBI motors
# on your local the network.
lookup = hebi.Lookup()


#
#  HEBI Discovery - Optional, useful if you don't know the names!
#
# If you already know the name(s), set to False to skip...
if True:
    # Give the lookup process 2 seconds to discover all modules.
    sleep(2)

    # Print the results.
    print('HEBI motors found on network:')
    for entry in lookup.entrylist:
        # Extract the family/name/address
        family  = entry.family
        name    = entry.name
        address = entry.mac_address

        # Print...
        print(f'family {family}  name {name}  address {address}')
    print('---------------------------------------------------')


#
#  Select the HEBI Motors
#
# Create a group from your motor names. Change motor numbers to yours!
# The 'robotlab' is the family name, which is the same for every motor.
# For two motors this will become: names = ['9.0', '9.2']
names = ['5.5']
group = lookup.get_group_from_names(['robotlab'], names)
print(f'Using motors {names}')

# Make sure this worked.
if group is None:
  print("Unable to find both motors " + str(names))
  raise Exception("Unable to connect to motors")

# Allocate command and feedback spaces.  We'll use (command) to send
# commands and (feedback) to receive motor position/velocity/effort
# data.  Pre-allocating makes the code faster and more predictable.
command  = hebi.GroupCommand(group.size)
feedback = hebi.GroupFeedback(group.size)


#
#  Set the Command Lifetime
#
```

```python
# The HEBI motors have a safety system, where they stop moving if they
# have not received a new command after N milliseconds.  This creates
# a TIME-OUT DURATION.  The default value is 0.25sec.  But as we want
# to update the command once per second, out lifetime needs to be at
# least 1sec!
group.command_lifetime = 1200   # Being 1.2sec


#
#  Example of getting the HEBI positions.
#
feedback = group.get_next_feedback(reuse_fbk=feedback)
pos = feedback.position[0]

print(f'Starting position {pos}')


#
#  Example of commanding HEBI positions.
#
# Note this has to be a list, in this case of 1 number.  The position
# is the motor angle in radians.
while (True):
    t = 0
    dt = 0.01
    TCLOCK = 15.0
    TRETURN = 1.0
    while t < TCLOCK:
        p = pi/(2 * TCLOCK) * t
        command.position = [p]
        group.send_command(command)
        sleep(dt)
        t += dt

    sleep(TRETURN)
```