

```

1  '''goals2democode.py
2
3  Goals 2 Demonstration Code for the HEBI motors
4
5  Please read through, fix the motor name, and add to the end.
6
7  '''
8
9  # Import useful packages
10 import hebi
11 import numpy as np          # For future use
12
13 from math import pi, sin, cos, asin, acos, atan2
14 from time import sleep, time
15
16
17 #
18 # HEBI Initialization
19 #
20 # Create a lookup object to discover/find/connect to the HEBI motors
21 # on your local the network.
22 lookup = hebi.Lookup()
23
24
25 #
26 # HEBI Discovery - Optional, useful if you don't know the names!
27 #
28 # If you already know the name(s), set to False to skip...
29 if True:
30     # Give the lookup process 2 seconds to discover all modules.
31     sleep(2)
32
33     # Print the results.
34     print('HEBI motors found on network:')
35     for entry in lookup.entrylist:
36         # Extract the family/name/address
37         family = entry.family
38         name = entry.name
39         address = entry.mac_address
40
41         # Print...
42         print(f'family {family} name {name} address {address}')
43     print('-----')
44
45 #
46 # Select the HEBI Motors
47 #
48 #
49 # Create a group from your motor names. Change motor numbers to yours!
50 # The 'robotlab' is the family name, which is the same for every motor.
51 # For two motors this will become: names = ['9.0', '9.2']
52 names = ['5.5']
53 group = lookup.get_group_from_names(['robotlab'], names)
54 print(f'Using motors {names}')
55
56 # Make sure this worked.
57 if group is None:
58     print("Unable to find both motors " + str(names))
59     raise Exception("Unable to connect to motors")
60
61 # Allocate command and feedback spaces. We'll use (command) to send
62 # commands and (feedback) to receive motor position/velocity/effort
63 # data. Pre-allocating makes the code faster and more predictable.
64 command = hebi.GroupCommand(group.size)
65 feedback = hebi.GroupFeedback(group.size)
66
67
68 #
69 # Set the Command Lifetime
70 #

```

```
71 # The HEBI motors have a safety system, where they stop moving if they
72 # have not received a new command after N milliseconds. This creates
73 # a TIME-OUT DURATION. The default value is 0.25sec. But as we want
74 # to update the command once per second, out lifetime needs to be at
75 # least 1sec!
76 group.command_lifetime = 1200 # Being 1.2sec
77
78
79 #
80 # Example of getting the HEBI positions.
81 #
82 feedback = group.get_next_feedback(reuse_fb=feedback)
83 pos = feedback.position[0]
84
85 print(f'Starting position {pos}')
86
87
88 #
89 # Example of commanding HEBI positions.
90 #
91 # Note this has to be a list, in this case of 1 number. The position
92 # is the motor angle in radians.
93 step_size = pi/30
94 while (True):
95     command.position = [0.0]
96     group.send_command(command)
97     for i in range(1, 16):
98         sleep(1.0)
99         command.position = [i * step_size]
100         group.send_command(command)
101
102     sleep(1.0)
103
104
```