

imu-convert.py

```
001 # Parsen und konvertieren von IMU-Daten
002 # #####
003
004 # Authors:
005 # Joshua Wolf
006 # Silas Teske
007 # Lasse Zeh
008 # Christopher Mahn
009
010 # #####
011
012 # Import of Libraries
013 # -----
014
015 # import math as m
016 # import string as st
017 # import random as r
018 import numpy as np
019 import os
020
021
022 # -----
023 # Debugging-Settings
024
025 verbose = True # Shows more debugging information
026
027
028 # Functions
029 # -----
030
031
032 # Classes
033 # -----
034
035
036 # Beginning of the Programm
037 # -----
038
039 if __name__ == '__main__':
040
041     # Import der Messwerte des Neigungssensors
042     file = open(os.path.join("data", "imu_data.txt"))
043     data = file.readlines()
044     file.close()
045
046     # Select active lines
047     data_split = []
048     for i, e in enumerate(data):
049         if e[0] != "#":
050             data_split.append(e.strip().split(" "))
051
052     # Select only marked lines
053     raw_markers = []
054     for i in data_split:
055         if i[0] != "0000":
056             raw_markers.append(i)
057
058     # Convert lines in float-lists
059     markers = []
060     for i, e in enumerate(raw_markers):
061         markers.append([])
062         for j in e:
063             if(j != ""): # Skip empty strings
064                 markers[i].append(float(j))
065
066     # Convert units
067     for i, e in enumerate(markers):
068         if(len(e) == 12):
069             markers[i][0] = int(e[0]) # marker-id to integer
070             markers[i][1] = e[1]
071             markers[i][2] = e[2] / 180 * np.pi # convert °/s to rad/s
072             markers[i][3] = e[3] / 180 * np.pi # convert °/s to rad/s
073             markers[i][4] = e[4] / 180 * np.pi # convert °/s to rad/s
074             markers[i][5] = e[5]
075             markers[i][6] = e[6]
076             markers[i][7] = e[7]
077             markers[i][8] = e[8] / 180 * np.pi # convert ° to s
078             markers[i][9] = e[9] / 180 * np.pi # convert ° to s
079             markers[i][10] = e[10] / 180 * np.pi # convert ° to s
080             markers[i][11] = e[11] - 273.15 # convert °K to °C
```

```
081
082 # Export
083 file = open(os.path.join("data", "imu_data_converted.txt"), f"w")
084 for i in markers:
085     for j, e in enumerate(i):
086         if(j == 0):
087             file.writelines(f"{e}")
088         else:
089             file.writelines(f"; {e}")
090     file.writelines(f"\n")
091 file.close()
```