```
 1
 2
 3 from DataHandlerNew import datahandler
 4 from UDPvideoStream import videoStream
 5 import threading
 6
 7 #the main class of the ROV. Starts the datahandler and the
    viedeostreamThread
 8
 9 def main():
10     # Use the Ip adress of the machine running the GUI
   here
11     guiIpadress= '192.168.0.103'
12     print("Starting")
13     #creates a thread of the UDP videostream
14     #t = threading.Thread(target=videoStream, name="
   thread1", args=(guiIpadress, 12342))
15     #t.start()
16     #starts the datahandler
17     comm = datahandler(readGUIPort=9876,sendGUIDataPort=
   8765,iAdress=(guiIpadress))
18
19 if __name__ == '__main__':
20     main()
21
```

```
 1 import queue
 2 import serial
 3 from threading import Thread
 4 import time
 5
 6 #Reads serialcom from the Arduino on the port "serialPort"
 7 class SerialRead ():
 8     def __init__(self,serialPort,semaPhore):
 9
10         self.q = queue.LifoQueue()
11         self.semaphore = semaPhore
12         self.serialPort = serialPort
13         self.dataArduino = bytearray(23)
14
15     def start(self):
16
17        t = Thread(target=self.run,name="thread3", args=())
18        t.daemon = True
19        t.start()
20
21
22  #runns when called thread.start. reads data from arduino
   and setts the input to the arranged value
23
24     def run(self):
25         try :
26             while (True):
27                 self.semaphore.acquire()
28
29                 data = self.serialPort.read(23)
30                 self.semaphore.release()
31                 if data.__len__()>0 :
32
33
34                   arrangedData = self.checkDataArrangement
   (data)
35
36                   self.q.put(arrangedData)
37
38                   time.sleep(0.5)
39
40
41         except serial.SerialException as e :
42            print("SerialPortException i SerialRead")
43
```

```
44
45      def getSerialData(self):
46
47          # Return the latest serialData
48          if not self.q.empty():
49              newCommand = self.q.get()
50
51             # while not self.q.empty():
52                 # trashBin = self.q.get()
53
54              return newCommand
55          else:
56              #print("empty Queue in seiralRead")
57              nodata=bytearray(23)
58              return nodata
59
60      def queReady(self):
61        if not self.q.empty():
62           return True
63        else:
64           return False
65
66
67
68
69   #checs the position of dthe data read from arduino and
    sets it on the right place
70   #@param data
71   #@return arrangedData
72
73      def checkDataArrangement(self, data):
74          realData = bytearray(23)
75          for x in range(0,22):
76
77
78              #print(x)
79              #print(str(data[x])+"Xval= "+ str(x))
80              if data[x] == (128):
81                   if x == 22:
82                      realData[0] = data[x]
83                      realData[1] = data[x - 22]
84                      realData[2] = data[x - 21]
85                      realData[3] = data[x - 20]
86                      realData[4] = data[x - 19]
87                      realData[5] = data[x - 18]
```

```
 88                        realData[6] = data[x - 17]
 89                        realData[7] = data[x - 16]
 90                        realData[8] = data[x - 15]
 91                        realData[9] = data[x - 14]
 92                        realData[10] = data[x - 13]
 93                        realData[11] = data[x - 12]
 94                        realData[12] = data[x - 11]
 95                        realData[13] = data[x - 10]
 96                        realData[14] = data[x - 9]
 97                        realData[15] = data[x - 8]
 98                        realData[16] = data[x - 7]
 99                        realData[17] = data[x - 6]
100                        realData[18] = data[x - 5]
101                        realData[19] = data[x - 4]
102                        realData[20] = data[x - 3]
103                        realData[21] = data[x - 2]
104                        realData[22] = data[x - 1]
105
106                    elif (x == 21):
107                        realData[0] = data[x]
108                        realData[1] = data[x +1]
109                        realData[2] = data[x - 21]
110                        realData[3] = data[x - 20]
111                        realData[4] = data[x - 19]
112                        realData[5] = data[x - 18]
113                        realData[6] = data[x - 17]
114                        realData[7] = data[x - 16]
115                        realData[8] = data[x - 15]
116                        realData[9] = data[x - 14]
117                        realData[10] = data[x - 13]
118                        realData[11] = data[x - 12]
119                        realData[12] = data[x - 11]
120                        realData[13] = data[x - 10]
121                        realData[14] = data[x - 9]
122                        realData[15] = data[x - 8]
123                        realData[16] = data[x - 7]
124                        realData[17] = data[x - 6]
125                        realData[18] = data[x - 5]
126                        realData[19] = data[x - 4]
127                        realData[20] = data[x - 3]
128                        realData[21] = data[x - 2]
129                        realData[22] = data[x - 1]
130
131                    elif (x == 20):
132                        realData[0] = data[x]
```

```
133                         realData[1] = data[x +1]
134                         realData[2] = data[x +2]
135                         realData[3] = data[x - 20]
136                         realData[4] = data[x - 19]
137                         realData[5] = data[x - 18]
138                         realData[6] = data[x - 17]
139                         realData[7] = data[x - 16]
140                         realData[8] = data[x - 15]
141                         realData[9] = data[x - 14]
142                         realData[10] = data[x - 13]
143                         realData[11] = data[x - 12]
144                         realData[12] = data[x - 11]
145                         realData[13] = data[x - 10]
146                         realData[14] = data[x - 9]
147                         realData[15] = data[x - 8]
148                         realData[16] = data[x - 7]
149                         realData[17] = data[x - 6]
150                         realData[18] = data[x - 5]
151                         realData[19] = data[x - 4]
152                         realData[20] = data[x - 3]
153                         realData[21] = data[x - 2]
154                         realData[22] = data[x - 1]
155
156                     elif (x == 19):
157                         realData[0] = data[x]
158                         realData[1] = data[x +1]
159                         realData[2] = data[x +2]
160                         realData[3] = data[x +3]
161                         realData[4] = data[x - 19]
162                         realData[5] = data[x - 18]
163                         realData[6] = data[x - 17]
164                         realData[7] = data[x - 16]
165                         realData[8] = data[x - 15]
166                         realData[9] = data[x - 14]
167                         realData[10] = data[x - 13]
168                         realData[11] = data[x - 12]
169                         realData[12] = data[x - 11]
170                         realData[13] = data[x - 10]
171                         realData[14] = data[x - 9]
172                         realData[15] = data[x - 8]
173                         realData[16] = data[x - 7]
174                         realData[17] = data[x - 6]
175                         realData[18] = data[x - 5]
176                         realData[19] = data[x - 4]
177                         realData[20] = data[x - 3]
```

```
178                         realData[21] = data[x - 2]
179                         realData[22] = data[x - 1]
180
181                     elif (x == 18):
182                         realData[0] = data[x]
183                         realData[1] = data[x +1]
184                         realData[2] = data[x +2]
185                         realData[3] = data[x +3]
186                         realData[4] = data[x +4]
187                         realData[5] = data[x - 18]
188                         realData[6] = data[x - 17]
189                         realData[7] = data[x - 16]
190                         realData[8] = data[x - 15]
191                         realData[9] = data[x - 14]
192                         realData[10] = data[x - 13]
193                         realData[11] = data[x - 12]
194                         realData[12] = data[x - 11]
195                         realData[13] = data[x - 10]
196                         realData[14] = data[x - 9]
197                         realData[15] = data[x - 8]
198                         realData[16] = data[x - 7]
199                         realData[17] = data[x - 6]
200                         realData[18] = data[x - 5]
201                         realData[19] = data[x - 4]
202                         realData[20] = data[x - 3]
203                         realData[21] = data[x - 2]
204                         realData[22] = data[x - 1]
205
206                     elif (x == 17):
207                         realData[0] = data[x]
208                         realData[1] = data[x +1]
209                         realData[2] = data[x +2]
210                         realData[3] = data[x +3]
211                         realData[4] = data[x +4]
212                         realData[5] = data[x +5]
213                         realData[6] = data[x - 17]
214                         realData[7] = data[x - 16]
215                         realData[8] = data[x - 15]
216                         realData[9] = data[x - 14]
217                         realData[10] = data[x - 13]
218                         realData[11] = data[x - 12]
219                         realData[12] = data[x - 11]
220                         realData[13] = data[x - 10]
221                         realData[14] = data[x - 9]
222                         realData[15] = data[x - 8]
```

```
223                         realData[16] = data[x - 7]
224                         realData[17] = data[x - 6]
225                         realData[18] = data[x - 5]
226                         realData[19] = data[x - 4]
227                         realData[20] = data[x - 3]
228                         realData[21] = data[x - 2]
229                         realData[22] = data[x - 1]
230
231                     elif (x == 16):
232                         realData[0] = data[x]
233                         realData[1] = data[x +1]
234                         realData[2] = data[x +2]
235                         realData[3] = data[x +3]
236                         realData[4] = data[x +4]
237                         realData[5] = data[x +5]
238                         realData[6] = data[x + 6]
239                         realData[7] = data[x - 16]
240                         realData[8] = data[x - 15]
241                         realData[9] = data[x - 14]
242                         realData[10] = data[x - 13]
243                         realData[11] = data[x - 12]
244                         realData[12] = data[x - 11]
245                         realData[13] = data[x - 10]
246                         realData[14] = data[x - 9]
247                         realData[15] = data[x - 8]
248                         realData[16] = data[x - 7]
249                         realData[17] = data[x - 6]
250                         realData[18] = data[x - 5]
251                         realData[19] = data[x - 4]
252                         realData[20] = data[x - 3]
253                         realData[21] = data[x - 2]
254                         realData[22] = data[x - 1]
255
256                     elif (x == 15):
257                         realData[0] = data[x]
258                         realData[1] = data[x +1]
259                         realData[2] = data[x +2]
260                         realData[3] = data[x +3]
261                         realData[4] = data[x +4]
262                         realData[5] = data[x +5]
263                         realData[6] = data[x + 6]
264                         realData[7] = data[x +7]
265                         realData[8] = data[x - 15]
266                         realData[9] = data[x - 14]
267                         realData[10] = data[x - 13]
```

```
268                         realData[11] = data[x - 12]
269                         realData[12] = data[x - 11]
270                         realData[13] = data[x - 10]
271                         realData[14] = data[x - 9]
272                         realData[15] = data[x - 8]
273                         realData[16] = data[x - 7]
274                         realData[17] = data[x - 6]
275                         realData[18] = data[x - 5]
276                         realData[19] = data[x - 4]
277                         realData[20] = data[x - 3]
278                         realData[21] = data[x - 2]
279                         realData[22] = data[x - 1]
280
281                     elif (x == 14):
282                         realData[0] = data[x]
283                         realData[1] = data[x +1]
284                         realData[2] = data[x +2]
285                         realData[3] = data[x +3]
286                         realData[4] = data[x +4]
287                         realData[5] = data[x +5]
288                         realData[6] = data[x + 6]
289                         realData[7] = data[x +7]
290                         realData[8] = data[x +8]
291                         realData[9] = data[x - 14]
292                         realData[10] = data[x - 13]
293                         realData[11] = data[x - 12]
294                         realData[12] = data[x - 11]
295                         realData[13] = data[x - 10]
296                         realData[14] = data[x - 9]
297                         realData[15] = data[x - 8]
298                         realData[16] = data[x - 7]
299                         realData[17] = data[x - 6]
300                         realData[18] = data[x - 5]
301                         realData[19] = data[x - 4]
302                         realData[20] = data[x - 3]
303                         realData[21] = data[x - 2]
304                         realData[22] = data[x - 1]
305
306                     elif (x == 13):
307                         realData[0] = data[x]
308                         realData[1] = data[x +1]
309                         realData[2] = data[x +2]
310                         realData[3] = data[x +3]
311                         realData[4] = data[x +4]
312                         realData[5] = data[x +5]
```

```
313                         realData[6] = data[x + 6]
314                         realData[7] = data[x +7]
315                         realData[8] = data[x +8]
316                         realData[9] = data[x +9]
317                         realData[10] = data[x - 13]
318                         realData[11] = data[x - 12]
319                         realData[12] = data[x - 11]
320                         realData[13] = data[x - 10]
321                         realData[14] = data[x - 9]
322                         realData[15] = data[x - 8]
323                         realData[16] = data[x - 7]
324                         realData[17] = data[x - 6]
325                         realData[18] = data[x - 5]
326                         realData[19] = data[x - 4]
327                         realData[20] = data[x - 3]
328                         realData[21] = data[x - 2]
329                         realData[22] = data[x - 1]
330
331                     elif (x == 12):
332                         realData[0] = data[x]
333                         realData[1] = data[x +1]
334                         realData[2] = data[x +2]
335                         realData[3] = data[x +3]
336                         realData[4] = data[x +4]
337                         realData[5] = data[x +5]
338                         realData[6] = data[x + 6]
339                         realData[7] = data[x +7]
340                         realData[8] = data[x +8]
341                         realData[9] = data[x +9]
342                         realData[10] = data[x +10]
343                         realData[11] = data[x - 12]
344                         realData[12] = data[x - 11]
345                         realData[13] = data[x - 10]
346                         realData[14] = data[x - 9]
347                         realData[15] = data[x - 8]
348                         realData[16] = data[x - 7]
349                         realData[17] = data[x - 6]
350                         realData[18] = data[x - 5]
351                         realData[19] = data[x - 4]
352                         realData[20] = data[x - 3]
353                         realData[21] = data[x - 2]
354                         realData[22] = data[x - 1]
355
356                     elif (x == 11):
357                         realData[0] = data[x]
```

```
358                         realData[1] = data[x +1]
359                         realData[2] = data[x +2]
360                         realData[3] = data[x +3]
361                         realData[4] = data[x +4]
362                         realData[5] = data[x +5]
363                         realData[6] = data[x + 6]
364                         realData[7] = data[x +7]
365                         realData[8] = data[x +8]
366                         realData[9] = data[x +9]
367                         realData[10] = data[x +10]
368                         realData[11] = data[x + 11]
369                         realData[12] = data[x - 11]
370                         realData[13] = data[x - 10]
371                         realData[14] = data[x - 9]
372                         realData[15] = data[x - 8]
373                         realData[16] = data[x - 7]
374                         realData[17] = data[x - 6]
375                         realData[18] = data[x - 5]
376                         realData[19] = data[x - 4]
377                         realData[20] = data[x - 3]
378                         realData[21] = data[x - 2]
379                         realData[22] = data[x - 1]
380
381                     elif (x == 10):
382                         realData[0] = data[x]
383                         realData[1] = data[x +1]
384                         realData[2] = data[x +2]
385                         realData[3] = data[x +3]
386                         realData[4] = data[x +4]
387                         realData[5] = data[x +5]
388                         realData[6] = data[x + 6]
389                         realData[7] = data[x +7]
390                         realData[8] = data[x +8]
391                         realData[9] = data[x +9]
392                         realData[10] = data[x +10]
393                         realData[11] = data[x + 11]
394                         realData[12] = data[x +12]
395                         realData[13] = data[x - 10]
396                         realData[14] = data[x - 9]
397                         realData[15] = data[x - 8]
398                         realData[16] = data[x - 7]
399                         realData[17] = data[x - 6]
400                         realData[18] = data[x - 5]
401                         realData[19] = data[x - 4]
402                         realData[20] = data[x - 3]
```

```
403                         realData[21] = data[x - 2]
404                         realData[22] = data[x - 1]
405                     elif (x == 9):
406                         realData[0] = data[x]
407                         realData[1] = data[x +1]
408                         realData[2] = data[x +2]
409                         realData[3] = data[x +3]
410                         realData[4] = data[x +4]
411                         realData[5] = data[x +5]
412                         realData[6] = data[x + 6]
413                         realData[7] = data[x +7]
414                         realData[8] = data[x +8]
415                         realData[9] = data[x +9]
416                         realData[10] = data[x +10]
417                         realData[11] = data[x + 11]
418                         realData[12] = data[x +12]
419                         realData[13] = data[x +13]
420                         realData[14] = data[x - 9]
421                         realData[15] = data[x - 8]
422                         realData[16] = data[x - 7]
423                         realData[17] = data[x - 6]
424                         realData[18] = data[x - 5]
425                         realData[19] = data[x - 4]
426                         realData[20] = data[x - 3]
427                         realData[21] = data[x - 2]
428                         realData[22] = data[x - 1]
429                     elif (x == 8):
430                         realData[0] = data[x]
431                         realData[1] = data[x +1]
432                         realData[2] = data[x +2]
433                         realData[3] = data[x +3]
434                         realData[4] = data[x +4]
435                         realData[5] = data[x +5]
436                         realData[6] = data[x + 6]
437                         realData[7] = data[x +7]
438                         realData[8] = data[x +8]
439                         realData[9] = data[x +9]
440                         realData[10] = data[x +10]
441                         realData[11] = data[x + 11]
442                         realData[12] = data[x +12]
443                         realData[13] = data[x +13]
444                         realData[14] = data[x +14]
445                         realData[15] = data[x - 8]
446                         realData[16] = data[x - 7]
447                         realData[17] = data[x - 6]
```

```
448                         realData[18] = data[x - 5]
449                         realData[19] = data[x - 4]
450                         realData[20] = data[x - 3]
451                         realData[21] = data[x - 2]
452                         realData[22] = data[x - 1]
453                     elif (x == 7):
454                         realData[0] = data[x]
455                         realData[1] = data[x +1]
456                         realData[2] = data[x +2]
457                         realData[3] = data[x +3]
458                         realData[4] = data[x +4]
459                         realData[5] = data[x +5]
460                         realData[6] = data[x + 6]
461                         realData[7] = data[x +7]
462                         realData[8] = data[x +8]
463                         realData[9] = data[x +9]
464                         realData[10] = data[x +10]
465                         realData[11] = data[x + 11]
466                         realData[12] = data[x +12]
467                         realData[13] = data[x +13]
468                         realData[14] = data[x +14]
469                         realData[15] = data[x +15]
470                         realData[16] = data[x - 7]
471                         realData[17] = data[x - 6]
472                         realData[18] = data[x - 5]
473                         realData[19] = data[x - 4]
474                         realData[20] = data[x - 3]
475                         realData[21] = data[x - 2]
476                         realData[22] = data[x - 1]
477
478                     elif (x == 6):
479                         realData[0] = data[x]
480                         realData[1] = data[x +1]
481                         realData[2] = data[x +2]
482                         realData[3] = data[x +3]
483                         realData[4] = data[x +4]
484                         realData[5] = data[x +5]
485                         realData[6] = data[x + 6]
486                         realData[7] = data[x +7]
487                         realData[8] = data[x +8]
488                         realData[9] = data[x +9]
489                         realData[10] = data[x +10]
490                         realData[11] = data[x + 11]
491                         realData[12] = data[x +12]
492                         realData[13] = data[x +13]
```

```
493                         realData[14] = data[x +14]
494                         realData[15] = data[x +15]
495                         realData[16] = data[x +16]
496                         realData[17] = data[x - 6]
497                         realData[18] = data[x - 5]
498                         realData[19] = data[x - 4]
499                         realData[20] = data[x - 3]
500                         realData[21] = data[x - 2]
501                         realData[22] = data[x - 1]
502
503                     elif (x == 5):
504                         realData[0] = data[x]
505                         realData[1] = data[x +1]
506                         realData[2] = data[x +2]
507                         realData[3] = data[x +3]
508                         realData[4] = data[x +4]
509                         realData[5] = data[x +5]
510                         realData[6] = data[x + 6]
511                         realData[7] = data[x +7]
512                         realData[8] = data[x +8]
513                         realData[9] = data[x +9]
514                         realData[10] = data[x +10]
515                         realData[11] = data[x + 11]
516                         realData[12] = data[x +12]
517                         realData[13] = data[x +13]
518                         realData[14] = data[x +14]
519                         realData[15] = data[x +15]
520                         realData[16] = data[x +16]
521                         realData[17] = data[x +17]
522                         realData[18] = data[x - 5]
523                         realData[19] = data[x - 4]
524                         realData[20] = data[x - 3]
525                         realData[21] = data[x - 2]
526                         realData[22] = data[x - 1]
527
528                     elif (x == 4):
529                         realData[0] = data[x]
530                         realData[1] = data[x +1]
531                         realData[2] = data[x +2]
532                         realData[3] = data[x +3]
533                         realData[4] = data[x +4]
534                         realData[5] = data[x +5]
535                         realData[6] = data[x + 6]
536                         realData[7] = data[x +7]
537                         realData[8] = data[x +8]
```

```
538                         realData[9] = data[x +9]
539                         realData[10] = data[x +10]
540                         realData[11] = data[x + 11]
541                         realData[12] = data[x +12]
542                         realData[13] = data[x +13]
543                         realData[14] = data[x +14]
544                         realData[15] = data[x +15]
545                         realData[16] = data[x +16]
546                         realData[17] = data[x +17]
547                         realData[18] = data[x +18]
548                         realData[19] = data[x - 4]
549                         realData[20] = data[x - 3]
550                         realData[21] = data[x - 2]
551                         realData[22] = data[x - 1]
552
553                     elif (x == 3):
554                         realData[0] = data[x]
555                         realData[1] = data[x +1]
556                         realData[2] = data[x +2]
557                         realData[3] = data[x +3]
558                         realData[4] = data[x +4]
559                         realData[5] = data[x +5]
560                         realData[6] = data[x + 6]
561                         realData[7] = data[x +7]
562                         realData[8] = data[x +8]
563                         realData[9] = data[x +9]
564                         realData[10] = data[x +10]
565                         realData[11] = data[x + 11]
566                         realData[12] = data[x +12]
567                         realData[13] = data[x +13]
568                         realData[14] = data[x +14]
569                         realData[15] = data[x +15]
570                         realData[16] = data[x +16]
571                         realData[17] = data[x +17]
572                         realData[18] = data[x +18]
573                         realData[19] = data[x +19]
574                         realData[20] = data[x - 3]
575                         realData[21] = data[x - 2]
576                         realData[22] = data[x - 1]
577
578                     elif (x == 2):
579                         realData[0] = data[x]
580                         realData[1] = data[x +1]
581                         realData[2] = data[x +2]
582                         realData[3] = data[x +3]
```

```
583                         realData[4] = data[x +4]
584                         realData[5] = data[x +5]
585                         realData[6] = data[x + 6]
586                         realData[7] = data[x +7]
587                         realData[8] = data[x +8]
588                         realData[9] = data[x +9]
589                         realData[10] = data[x +10]
590                         realData[11] = data[x + 11]
591                         realData[12] = data[x +12]
592                         realData[13] = data[x +13]
593                         realData[14] = data[x +14]
594                         realData[15] = data[x +15]
595                         realData[16] = data[x +16]
596                         realData[17] = data[x +17]
597                         realData[18] = data[x +18]
598                         realData[19] = data[x +19]
599                         realData[20] = data[x +20]
600                         realData[21] = data[x - 2]
601                         realData[22] = data[x - 1]
602
603                     elif (x == 1):
604                         realData[0] = data[x]
605                         realData[1] = data[x +1]
606                         realData[2] = data[x +2]
607                         realData[3] = data[x +3]
608                         realData[4] = data[x +4]
609                         realData[5] = data[x +5]
610                         realData[6] = data[x + 6]
611                         realData[7] = data[x +7]
612                         realData[8] = data[x +8]
613                         realData[9] = data[x +9]
614                         realData[10] = data[x +10]
615                         realData[11] = data[x + 11]
616                         realData[12] = data[x +12]
617                         realData[13] = data[x +13]
618                         realData[14] = data[x +14]
619                         realData[15] = data[x +15]
620                         realData[16] = data[x +16]
621                         realData[17] = data[x +17]
622                         realData[18] = data[x +18]
623                         realData[19] = data[x +19]
624                         realData[20] = data[x +20]
625                         realData[21] = data[x +21]
626                         realData[22] = data[x - 1]
627
```

```
628                     else:
629                         realData = data
630
631
632
633             return realData
634
635
636
```

```
 1 import serial
 2 import queue
 3 from threading import Thread
 4
 5
 6
 7
 8 class SerialSend ():
 9     def __init__(self,serialPort,semaPhore):
10
11         #Semaphore to avoid serail send/recieve collision.
12         self.semaphore = semaPhore
13         #Port of the serial(USB port)
14         self.serialPort = serialPort
15         #Data holder for the class
16         self.q = queue.LifoQueue()
17
18
19     #Creates an thread of the run function, and starts it.
20     def start(self):
21         # Start the thread send arduino data
22         t = Thread(target=self.run,name="thread2", args=()
   )
23         t.daemon = True
24         t.start()
25
26     #sends data to the serialport.
27     def run(self):
28         try:
29
30             while (True):
31                 #checks if the queue contains any data
32                 if not self.q.empty():
33                     dataToArduino = self.getNewDataString(
   )
34                     for x in range(0,5):
35                         print(dataToArduino[x])
36
37                     #aquires the semaphore so the
   serialport can be used.
38                     self.semaphore.acquire()
39
40
41                     print("SEND Serial " + str(
   dataToArduino))
```

```
42                      #sends the data to the serialport
43                      self.serialPort.write(dataToArduino)
44                      #releases the semaphore so the
   readerclass can use the port.
45                      self.semaphore.release()
46
47          except serial.portNotOpenError:
48                  print("SerialPortException i SerialSend")
49
50      # Updates the data to be sent to the SerialPort
51      def updateDataString(self,newData):
52          self.q.put(newData)
53
54      # Returns the lates queue object,and flushes the queue
   . If there is none and empty array is returned.
55      def getNewDataString(self):
56
57          if not self.q.empty():
58              newCommand = self.q.get()
59              while not self.q.empty():
60                  trashBin = self.q.get()
61
62
63              return newCommand
64          else:
65              nodata=bytearray(6)
66              return nodata
67
68
69
70
71
72
```

```
 1 import queue
 2 import socket
 3 from threading import Thread
 4 from struct import *
 5
 6
 7 #Reads data sendt from GUI throug UDP com
 8 class UDPreadGUI(Thread):
 9
10     def __init__(self,Port):
11         Thread.__init__(self)
12         #UDP read port
13         self.port = Port
14         #byte array of 6 bytes
15         self.data = bytearray(6)
16         #UDP socket
17         self.sock = socket.socket(socket.AF_INET,  #
   Internet
18                                   socket.SOCK_DGRAM)  # UDP
19         #connect to the socket with local adress
20         self.sock.bind(('',self.port))
21         #queue makes it possible to access data from the
   thread without data collision.
22         self.q = queue.LifoQueue()
23
24     # Creates an thread of the run function, and starts it
   .
25     def start(self):
26
27         t = Thread(target=self.run, args=())
28         t.daemon = True
29         t.start()
30
31
32      #runs this method when thread.start is called
33      #gets the data from GUI
34
35     def run(self):
36         while True:
37             try:
38
39                 #reads 6 bytes of data from socket.
40                 self.data = self.sock.recv(6)
41
42
```

```
43                  for x in range(0, 6):
44                      print(self.data[x])
45                 # print("Read UDP: " + str(data2))
46
47                  #puts the data to the queue
48                  self.q.put(self.data)
49
50              except socket.error as e :
51                  print("uhdfuhdf")
52
53
54
55
56       #retuns data recieved from GUI
57       # @return data
58
59
60      def getDataGUI(self):
61
62          # Return the latest queue data and removes the
    rest of the data if there is any left.
63          # #returns an empty array of 0 if the function is
    called without data in the queue.
64          if not self.q.empty():
65              newCommand = self.q.get()
66              while not self.q.empty():
67                  trashBin = self.q.get()
68
69              return newCommand
70          else:
71
72              nodata=bytearray(6)
73              return nodata
74
75
76      #checks if the queue contains new data.
77      def queReady(self):
78          if not self.q.empty():
79              return True
80          else:
81              return False
82
83
84
```

```
 1 import queue
 2 import socket
 3 import time
 4 from threading import Thread
 5
 6 class UDPsendData(Thread):
 7     # class constructor
 8     def __init__(self,Port,Adress):
 9         Thread.__init__(self)
10         #UDP send port
11         self.port = Port
12         #Adress of the reciever
13         self.address = Adress
14         #data to be sent, array of 23 bytes
15         self.data = bytearray(23)
16         #queue storage variable for new data to be sent.
17         self.q = queue.LifoQueue()
18
19
20
21
22     def start(self):
23         # creates and UDP socket.
24         self.sock = socket.socket(socket.AF_INET,  #
   Internet
25                                   socket.SOCK_DGRAM)
26         #Creates a thread of the run function and Starts
   the thread
27         t = Thread(target=self.run, args=())
28         t.daemon = True
29         t.start()
30
31     def  run(self):
32
33         while (True):
34             try:
35                 #checks if there is any data to be sent
36                 if self.queReady():
37
38                     packet1 = self.getNewDataString()
39
40
41
42                     self.sock.sendto(packet1, (self.
   address, self.port))
```

```
43                      time.sleep(0.5)
44
45              except Exception as e:
46                  print(e)
47
48
49
50      #Updates the data to be sent to the GUI
51      def updateDataString(self,newData):
52          self.q.put(newData)
53
54
55      # Returns the lates queue object. If there is none and
    empty array is returned.
56      def getNewDataString(self):
57
58
59          if not self.q.empty():
60              newCommand = self.q.get()
61              while not self.q.empty():
62                  trashBin = self.q.get()
63
64              return newCommand
65          else:
66              nodata=bytearray(23)
67
68              return nodata
69
70      #checks if the queue contains any data.
71      def queReady(self):
72          if not self.q.empty():
73              return True
74          else:
75              return False
76
77
```

```
 1 import socket
 2 import serial
 3 from SerialRead import SerialRead
 4 from SerialSend import SerialSend
 5 from UDPsendData import UDPsendData
 6 from UDPreadGUI import UDPreadGUI
 7 import threading
 8 from ThrusterControl import ThrusterControl
 9 import time
10
11
12 class datahandler:
13     # class constructor
14     def __init__(self,readGUIPort,sendGUIDataPort,iAdress)
   :
15         #setting up gui data variables
16
17         self.senderUDP = UDPsendData(sendGUIDataPort,
   iAdress)
18         self.readerUDP = UDPreadGUI(readGUIPort)
19         #setting up serial variables
20         self.SerialSemaphore = threading.Semaphore()
21         self.senderSerial = SerialSend(serialPort=serial.
   Serial('/dev/ttyUSB0', 19200,timeout=5),semaPhore=self.
   SerialSemaphore)
22         self.readerSerial = SerialRead(serialPort=serial.
   Serial('/dev/ttyUSB0', 19200,timeout=5),semaPhore=self.
   SerialSemaphore)
23         self.thrusterControl = ThrusterControl()
24         self.startThreads()
25         #command variables from GUI thruster cmnd:(fwd,
   back etc..)power:0-100 thruster.
26         # Used in the ThrusterControl class
27         self.thrusterCmd = 0
28         self.thrusterPower = 0
29
30         #starting the While loop for logic
31         self.runDatahandler()
32
33
34      # starting the threads when called
35
36     def  startThreads(self):
37         try:
38             #thrusterControl.start();
```

```
39              self.readerUDP.start()
40
41              self.senderSerial.start()
42              self.readerSerial.start()
43          except RuntimeError:
44              print("Cant Start Threads in Datahandler
   StartThread functionn")
45          try:  #Starts sending to GUI threadfo
46              self.senderUDP.start()
47
48          except RuntimeError:
49              print("Cant Start Threads in Datahandler
   StartThread senderUDP and SerialCom")
50
51
52
53
54       #Setting the trustervalues
55       #@param command
56       #@param thrusterPower"""""
57
58      def runDatahandler(self):
59          time.sleep(2)
60          while True:
61              if self.readerUDP.queReady():
62                  #get reformat data to send to arduino
63                  self.guiReadData= self.dataToArduino()
64                  #update serial data to arduino
65                  self.senderSerial.updateDataString(self.
   guiReadData)
66              #get last serial read array
67              if self.readerSerial.queReady():
68                  self.guiSendData= self.readerSerial.
   getSerialData()
69                  #send data to the gui
70                  self.senderUDP.updateDataString(self.
   guiSendData)
71
72
73
74
75      #Returns the  Bytearray to be sendt to arduino
76      #@return dataToArduino
77
78      def dataToArduino(self):
```

```
 79          data = self.readerUDP.getDataGUI()
 80         #// ************** Thruster Variables
     ***************** //
 81          self.thrusterCmd = data[0]
 82          self.thrusterPower = data[2]
 83          self.thrusterControl.thrusterValues(self.
    thrusterCmd, self.thrusterPower)
 84
    #//*************************************************
    //
 85          dataToArduino = bytearray(6)
 86
 87
 88          dataToArduino[0] = 101   #//Flag byte ( 101 )
 89          dataToArduino[1] = data[3]    #// Start byte ( 0
    to 1 )
 90          dataToArduino[2] = self.convertToSignedValues(
    value=self.thrusterControl.getThrusterThree()) #//
    Thruster 1 ( - 100 to 100 )
 91          dataToArduino[3] = self.convertToSignedValues(
    value=self.thrusterControl.getThrusterTwo())   #//
    Thruster 2 ( - 100 to 100 )
 92          dataToArduino[4] = self.convertToSignedValues(
    value=self.thrusterControl.getThrusterOne())#// Thruster
    3 ( - 100 to 100 )
 93          dataToArduino[5] = data[1]
           #// Light control ( 0 - 100 ) vAR DATA[2]
 94          return dataToArduino
 95
 96
 97      #the arduino is using signe bytes, range: -128-127,
    python is using unsigned range: 0-255
 98      #in binary from 0-127 is the same in both arduino and
     python.
 99      # but over 128, reads as negativ by the arduino 128=-
    128
100      #if i want to send -100, i will send 255-100 +1
101      def convertToSignedValues(self,value):
102          if value>=0:
103              return value
104          else:
105              newValue= 255+value+1
106              return newValue
107
108
```

109
110

```
 1 import time
 2 import socket
 3 import cv2
 4
 5
 6
 7 def videoStream(ipadress,port):
 8     cam = cv2.VideoCapture(0)
 9     #cam.set(cv2.CAP_PROP_FPS,30)
10     UDP_IP = ipadress
11     UDP_PORT = port
12     sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM
   )
13
14
15
16     # warmup camera
17     time.sleep(0.1)
18     while True:
19         ret_val, img = cam.read(0)
20
21         img = cv2.flip(img, 1)
22         img = cv2.resize(img, (320, 240))
23
24         x = [int(cv2.IMWRITE_JPEG_QUALITY), 80]
25         _, compressed = cv2.imencode(".jpg", img, x)
26         sock.sendto(compressed, (UDP_IP, UDP_PORT))
27
28     rawCapture.truncate(0)
29
30
31
```

```
 1 class ThrusterControl(object):
 2     # class constructor
 3     def __init__(self):
 4         self.selfthrusterOne = 0
 5         self.thrusterTwo = 0
 6         self.thrusterThree = 0
 7
 8
 9
10
11
12
13  # sets the right thrustervalue from the command sendt
   from GUI.
14  # @param command  can be fwd,left,right ++
15  # @param power the persentage of the thrustervalue
16
17
18     def thrusterValues(self, command,power):
19
20         if (command == 2): # // FWD
21             self.thrusterOne = -power
22             self.thrusterTwo = power
23
24         elif (command == 18):  # Slide FWD Right
25
26             self.thrusterTwo = power
27             self.thrusterThree = power
28         elif command == 16: # Right
29             calculationVar = power * 58  # ThrusterOne and
    ThrusterTwo need 58% of ThrusterThree's Power.
30             self.thrusterOne = round(calculationVar/(100))
31             self.thrusterTwo = round(calculationVar/(100))
32             self.thrusterThree = power
33
34         elif command == 20:  # Slide Back Right
35             self.thrusterOne = power
36             self.thrusterThree = power
37         elif command == 4:# Back
38             self.thrusterOne = power
39             self.thrusterTwo = -power
40         elif command == 12: # Slide Back Left
41
42
43             self.thrusterTwo = -power
```

```
44              self.thrusterThree = -power
45          elif command == 8:  #Left
46              calculationVar = power * 58  # ThrusterOne and
    ThrusterTwo need 58% of ThrusterThree's Power.
47              self.thrusterOne = round(-(calculationVar/(100
   )))
48              self.thrusterTwo = round(-(calculationVar/(100
   )))
49              self.thrusterThree =-power
50          elif command == 10: # Slide Fwd Left
51              self.thrusterOne = -power
52              self.thrusterThree = -power
53          elif (command == 32): # Rotate Left
54              self.thrusterOne = -power
55              self.thrusterTwo = -power
56              self.thrusterThree = power
57          elif (command == 64): # Rotate Right
58              self.thrusterOne = power
59              self.thrusterTwo = power
60              self.thrusterThree = -power
61          else:
62              # Do nothing
63              self.thrusterOne = 0
64              self.thrusterTwo = 0
65              self.thrusterThree = 0
66
67
68
69       # return the trusterone value
70       # @return thrusterOne
71
72      def getThrusterOne(self):
73          return self.thrusterOne
74
75
76
77      #return the trusterTwo value
78      #return thrusterTwo
79
80      def getThrusterTwo(self):
81          return self.thrusterTwo
82
83
84
85      #return the trusterTree value
```

```
86      #return thrusterThree
87
88      def getThrusterThree(self):
89          return self.thrusterThree
90
91
92
```

```
 1 import time
 2 import socket
 3 import cv2
 4
 5
 6 cam = cv2.VideoCapture(0)
 7 cam.set(cv2.CAP_PROP_FPS,30)
 8 UDP_IP = '192.168.0.103'
 9 UDP_PORT = 12342
10 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
11
12
13
14 # warmup camera
15 time.sleep(0.1)
16 while True:
17     ret_val, img = cam.read(0)
18
19     img = cv2.rotate(img,cv2.ROTATE_90_COUNTERCLOCKWISE)
20     img = cv2.resize(img, (320, 240))
21
22     x = [int(cv2.IMWRITE_JPEG_QUALITY), 80]
23     _, compressed = cv2.imencode(".jpg", img, x)
24     sock.sendto(compressed, (UDP_IP, UDP_PORT))
25
26 rawCapture.truncate(0)
27
28
29
30
```