

```

1  from serial import Serial
2  class Icom():
3      def __init__( self,*, port, adr , debug = False):
4          self.adr=adr
5          self.port=port
6          self.debug=debug
7
8      def frequenz(self):
9          ''' kommando zusammenstellen'''
10         request = bytes ( [0xFE, 0xFE, self.adr , 0xE0, 0x03, 0xFD] )
11         if self.debug:
12             hex_bytes=bytes.hex(request).upper()
13             print( f'gesendet : {hex_bytes} (Frequenzabfrage ic-705)' )
14         try:
15             serial= Serial( self.port, \
16                             baudrate=9600, timeout=1, writeTimeout=1)
17         except Exception as e:
18             ''' fehler ser. schnittstelle '''
19             x= e.args[1]
20             print(x)  # [0:x.index(':')]
21             return 99 # fehlercodierung , kommt real nicht vor
22
23         '''senden '''
24         serial.write(request)
25         if self.adr != 0xa4:
26             response=serial.read_until(bytes([0xfd])) #echo
27             print(f'*** Test mit Icom 706, adr = {hex(self.adr)} ***')
28             response=serial.read_until(bytes([0xfd]))
29             serial.close()
30             ng_message = b'\xfe\xfe\xe0\xa4\xfa\xfd'
31             if response==b'': # nix da
32                 print( " *** no data ***" )
33                 return(92)  # fehlercodierung , kommt real nicht vor
34             else:
35                 if response == ng_message:
36                     hex_bytes=bytes.hex(ng_message).upper()
37                     print( f'empfangen: {hex_bytes} (Antwort vom ic-705) ' )
38                     s=f'***ERROR*** frequenz abfragen : antwort "NotGood" vom ic705'
39                     print(s)
40                     khz=93 # fehlercodierung , kommt real nicht vor
41                 else:
42
43                     ''' daten interpretieren'''
44                     hex_bytes=bytes.hex(response)
45                     print(f'von ICOM: {hex_bytes}')
46                     s=''
47                     for i in range(18,9,-2):
48                         s=s+hex_bytes[i:i+2]
49                     print(f'    Daten: {s}')  # s= "0438875000"
50                     try:
51                         khz= int(int(s)/1e3)
52                         print(f'    [kHz]: {khz}')
53                     except:
54                         khz=94
55                 return khz
56
57     def maxpower(self,proz):
58         ''' leistungbegrenzung des ic-705 einstellern
59         port : Ser.Schnittstelle #'/dev/ttyUSB0'
60         adr  : Icom CI-V  0xa4 für ic705

```

```

61         Return: ok/ Fehler als text
62     ...
63     if self.adr != 0xa4:
64         return '*** kein ic-705 ***'
65
66     ''' parameter fuer leistungsbegrenzung '''
67     pow_code=0x04
68     if proz>=100:
69         proz=100
70         pow_code=0x04
71     elif proz >= 50:
72         proz=50
73         pow_code=0x03
74     elif proz >= 25:
75         proz=25
76         pow_code=0x02
77     elif proz >= 10:
78         proz=10
79         pow_code=0x01
80     else :
81         proz=5
82         pow_code=0x00
83
84     ''' kommandodaten zusammenstellen '''
85     cmd = bytes ([0xFE, 0xFE, self.adr , 0xe0, 0x1a, 0x05,0x00, 0x37, pow_code, 0x00, 0xFD])
86     ''' ser. schnittstelle offnen '''
87     try:
88         serial= Serial( self.port, \
89                         baudrate=9600, timeout=3, writeTimeout=3 )
90     except Exception as e:
91         ''' fehler ser. schnittstelle '''
92         x= e.args[1]
93         print(x[0:x.index(':')])
94         return '*** fehler serial ***'
95
96     '''senden '''
97     serial.write(cmd)
98     response=serial.read_until(bytes([0xfd]))
99     serial.close()
100    x= bytes.hex(cmd, ' ').upper()
101    #print(f'maxpower={proz}% code= {x}')
102    #print('response:',response)
103    ok_message = b'\xfe\xfe\xe0\xa4\xfb\xfd'
104    ng_message = b'\xfe\xfe\xe0\xa4\xfa\xfd'
105    if response == ok_message:
106        s=f'maxpower={proz}% eingestellt'
107    elif response ==ng_message:
108        s=f'***ERROR*** maxpower einstellung : antwort "NotGood" vom ic705'
109    return s
110
111 if __name__ == '__main__':
112     icom= Icom(port='/dev/ttyACM0', adr= 0xa4, debug=1)
113     print(f'Frequenz: {icom.frequenz()}')
114     print(icom.maxpower(100))
115
116

```