Sources Code:

|  |
| --- |
| ' |
|  |  |
|  |  |
|  |  |
|  |  |
|  | Li-Fi COVID 19 Code |
|  | ''' |
|  | import serial |
|  | import binascii |
|  | import sys |
|  | import random |
|  | from tqdm import tqdm |
|  | import base64 |
|  | import time |
|  |  |
|  | def pic(s): |
|  | ''' |
|  | :param s: The image link |
|  | Function reads the imagefile and converts it to Base-64 string list. |
|  | :return: Base-64 string file. |
|  | ''' |
|  | with open(s,"rb") as imageFile: |
|  | strr = base64.b64encode(imageFile.read()) |
|  | print(str(strr)[2:-1]) |
|  | return str(strr)[2:-1] |
|  | def bin\_str(s): |
|  | ''' |
|  | :param s: a character |
|  | :return: The ASCII number in binary with padding 0s to make them uniform. |
|  | ''' |
|  | data = "" |
|  | for i in s: |
|  | temp = str(bin(ord(i)))[2:] |
|  | data +=temp.rjust(8,'0') |
|  | return data |
|  |  |
|  | ser = serial.Serial('COM4',115200,timeout=.1) #Begin the Serial connection |
|  | print("######################################################\n" |
|  | "##--------------------START-------------------------##\n" |
|  | "######################################################\n" |
|  | "##-------------------Emitter------------------------##\n" |
|  | "##---------------------Side-------------------------##\n" |
|  | "##----------------Start Sending-- ------------------##\n" |
|  | "##--------------------------------------------------##\n" |
|  | "######################################################\n" |
|  | ) |
|  |  |
|  | img\_link = "C:/Users/Ramon Qu/Desktop/vlc/untitled/exp2.jpg" |
|  | test = ["Ramon Qu", |
|  | "Lyndon Institute", |
|  | ":-)", |
|  | "Happy", |
|  | "Today is May 1st", |
|  | "Happy Monday", |
|  | "Welcome to the fair"] |
|  |  |
|  | while(not "DONE".encode() in ser.readline()): |
|  | pass |
|  | flag = False |
|  | while True: |
|  | a = input() |
|  | ''' |
|  | input 's', 't' or 'd' to select the mode you would like to use. |
|  | 's' - Transmit the image (Set the link before running it) |
|  | 't' - Text Mode. You can input the text and send to the other side |
|  | 'd' - Auto Text Mode. THere are test string list. Every time, the function will randomly pick one and send to the other side. |
|  | ''' |
|  | if (a=="s"): |
|  | print( |
|  | "##-------------Transmitting Image Mode--------------##\n" |
|  | "##--------------------------------------------------##\n" |
|  | "||||||||||||||||||||||||||||||||||||||||||||||||||||||\n" |
|  | "VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV\n" |
|  | " \n" |
|  | ) |
|  | pbar = tqdm(total=100) # Progress bar function. |
|  | t0 = time.clock() # Initial the clock |
|  | data = bin\_str(pic(img\_link)) |
|  | temp = len(data) |
|  | # Parse the data to 540 bits/ group |
|  | data = [ data[i:i+540] for i in range(0, len(data), 540) ] |
|  | # Send the number of packets to emitter controller. |
|  | ser.write(bytes("{0:b}".format(temp//8).rjust(15,'0')+"\n",encoding="ascii")) |
|  | print(temp//8) |
|  | ser.flush() |
|  | print(len(data)) |
|  | # Wait until the emitter confirming the message has been sent |
|  | while(not "DONE".encode() in ser.readline()): |
|  | pass |
|  | # Send the packets. 540 bits/ group. The emitter will regroup them into 15 bits/ packet |
|  | for i in range(len(data)): |
|  | ser.write(bytes(data[i]+"\n",encoding="ascii")) |
|  | ser.flush() |
|  | pbar.update(1/len(data)\*100 |
|  | while(not "DONE".encode() in ser.readline()): |
|  | pass |
|  | print(bin\_str(pic(img\_link))) |
|  | print(time.clock()-t0) |
|  | if(a=="t"): |
|  | print( |
|  | "##-------------Transmitting Text Mode---------------##\n" |
|  | "##--------------------------------------------------##\n" |
|  | "## Please Type in what you would like to transmit ##\n" |
|  | "||||||||||||||||||||||||||||||||||||||||||||||||||||||\n" |
|  | "VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV\n" |
|  | " \n" |
|  | ) |
|  | count = 0 |
|  | while(1): |
|  | data = input() |
|  | print("No."+str(count)+" -- Raw Data: ") |
|  | # Serial write a start signal character |
|  | ser.write(bytes("00010\n",encoding="ascii")) |
|  | ser.flush() |
|  | while(not "DONE".encode() in ser.readline()): |
|  | pass |
|  | #data is the input. It turns very character to 8 bit binary string and send to the emitter controller |
|  | for i in data: |
|  | ser.write(bytes(str(bin(ord(i))).rjust(8,'0')+"\n",encoding="ascii")) |
|  | ser.flush() |
|  | print(bytes(str(bin(ord(i))).rjust(8,'0'), encoding="ascii"),end="") |
|  | while(not "DONE".encode() in ser.readline()): |
|  | pass |
|  | #Send the end signal character |
|  | ser.write(bytes("00011\n",encoding="ascii")) |
|  | ser.flush() |
|  | print() |
|  | count+=1 |
|  | while(not "DONE".encode() in ser.readline()): |
|  | pass |
|  | if (a=="d"): |
|  | print( |
|  | "##----------Auto Transmitting Text Mode-------------##\n" |
|  | "##--------------------------------------------------##\n" |
|  | "||||||||||||||||||||||||||||||||||||||||||||||||||||||\n" |
|  | "VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV\n" |
|  | " " |
|  | ) |
|  | last = -1 |
|  | count = 0 |
|  | while (1): |
|  | #Random Select one |
|  | temp = random.randint(0, len(test) - 1) |
|  | while(temp ==last): |
|  | temp = random.randint(0, len(test) - 1) |
|  | last = temp |
|  | data = test[temp] |
|  | # The same method used in the text mode. |
|  | print("No."+str(count)+"---> "+str(data)) |
|  | count+=1 |
|  | ser.write(bytes("00010\n", encoding="ascii")) |
|  | ser.flush() |
|  | while (not "DONE".encode() in ser.readline()): |
|  | pass |
|  | for i in data: |
|  | ser.write(bytes(str(bin(ord(i))).rjust(8, '0') + "\n", encoding="ascii")) |
|  | ser.flush() |
|  | while (not "DONE".encode() in ser.readline()): |
|  | pass |
|  |  |
|  | ser.write(bytes("00011\n", encoding="ascii")) |
|  | ser.flush() |
|  | while (not "DONE".encode() in ser.readline()): |
|  | pass |
|  | time.sleep(1) |