

```

#!/usr/bin/env python

import sys
import time
from rflib import *
from struct import *

d = RfCat()

txLen = 0
frequency = int(raw_input("What frequency should we transmit on? "))
baudRate= int(raw_input("What baud rate should we use? "))
key = str(raw_input("What key are we transmitting? "))

def ConfiguredD(d):
    d.setMdmModulation(MOD_ASK_00K)
    d.setFreq(frequency)
    d.makePktFLEN(txLen)
    d.setMdmSyncMode(0)
    d.setMdmDRate(baudRate)
    d.setMaxPower()

pre_length = 12
pre_str = ""
for i in range(pre_length):
    pre_str = pre_str + "10"

print "Preamble string:",pre_str
gap_str = "00000000"
print "Preamble gap:",gap_str

bin_sec_key = str(key)
print "Binary (Non-PWM) Key:",bin_sec_key
pwm_sec_key = ""

for k in bin_sec_key:
    x = "*"
    if (k == "0"):
        x = "110"
    if (k == "1"):
        x = "100"
    pwm_sec_key = pwm_sec_key + x

print "Binary (PWM) Key:", pwm_sec_key
full_tx_bin = pre_str + gap_str + pwm_sec_key

print "Full Transmission (Binary):", full_tx_bin

# calculate the number of characters left after conversion
tail = len(full_tx_bin) % 4

```

```

end = len(full_tx_bin) / 4

full_tx_hex = ""
for l in range(end + 1):
    start = l * 4
    end = (l * 4) + 4
    full_tx_hex = full_tx_hex + hex(int(full_tx_bin[start:end],2))
[2:]

if (tail == 1):
    full_tx_hex = full_tx_hex + hex(int(full_tx_bin[-1:] +
"000",2))[2:]

if (tail == 2):
    full_tx_hex = full_tx_hex + hex(int(full_tx_bin[-2:] +
"00",2))[2:]

if (tail == 3):
    full_tx_hex = full_tx_hex + hex(int(full_tx_bin[-3:] + "0",2))
[2:]

if (len(full_tx_hex) % 2 == 1):
    full_tx_hex = full_tx_hex + "0"

print "Full Transmission (Hex):",full_tx_hex

txLen = len(full_tx_hex)
ConfigureD(d)
print "TX'ing key..."
d.RFxmit(full_tx_hex.decode('hex'), 25)
print "Done"

```