

Hello,

Several remarks :

- 1) In order to communicate the device has to exchange messages in a specific crypted way (using the device key)
 - a. Since the tinytuya programs is working that means that you have given it a **correct** key.
 - b. When looking at the logs, in the first message sent to the device, the tinytuya logs show the 84c long message as :

```
DEBUG:payload  
encrypted=b'000055aa0000000100000003000000441d413749d2967f950b794b6848fb8c7039dfe8d39fd2224e3932f8d7ee5e98e477ba51a5b1c0d7862e664dc1474b6596cf20e441de5af8e4bbd4ab9671d6adf0000aa55'
```

Where the esptuya message logs show

```
000055AA00000001000000030000004475B702DD3C2756A3C2C500F03957B8B5B1B772E3C1E615035C627F2F8BA4BF67B9A3F7D25C75D50D16426A3B95C0E369BBB4B01AE2D4AB67649197A259A49223000AA55
```

They should be exactly the same, the prefix and suffix are ok, but the core of the message is incorrect indicating that **the device key provided does not seem to be the same**

The correct message (tinytuya) is :

Prefix Header: 000055aa000000010000000300000044

Msgk 1d413749d2967f950b794b6848fb8c7039dfe8d39fd2224e3932f8d7ee5e98e4

CRC 77ba51a5b1c0d7862e664dc1474b6596cf20e441de5af8e4bbd4ab9671d6adf0

Suffix 0000aa55

The Msgk is the encryption of

303132333435363738396162636465661010101010101010101010101010101010

With the device key.

You can test it online at <http://aes.online-domain-tools.com/>

Enter input text : 303132333435363738396162636465661010101010101010101010101010101010

Hex

Function AES

Mode ECB

Key your device key

Encrypt shoud give you

1d413749d2967f950b794b6848fb8c7039dfe8d39fd2224e3932f8d7ee5e98e4

As long as you don't get this means you have the wrong device key.

The encryption with the device key you provided (6E24753F5976587C523A33595A352753)

gives 75B702DD3C2756A3C2C500F03957B8B5B1B772E3C1E615035C627F2F8BA4BF67

Which is not correct

2) I have a doubt on the Send/receive function, I don't recognize the displays

You could run a test like :

```
//-----  
void test( ){  
byte testmsg [84] = { 0x00, 0x00, 0x55, 0xaa, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x03,  
0x00, 0x00, 0x00, 0x44, 0x1d, 0x41, 0x37, 0x49, 0xd2, 0x96, 0x7f, 0x95, 0x0b, 0x79, 0x4b,  
0x68, 0x48, 0xfb, 0x8c, 0x70, 0x39, 0xdf, 0xe8, 0xd3, 0x9f, 0xd2, 0x22, 0x4e, 0x39, 0x32,  
0xf8, 0xd7, 0xee, 0x5e, 0x98, 0xe4, 0x77, 0xba, 0x51, 0xa5, 0xb1, 0xc0, 0xd7, 0x86, 0x2e,  
0x66, 0x4d, 0xc1, 0x47, 0x4b, 0x65, 0x96, 0xcf, 0x20, 0xe4, 0x41, 0xde, 0x5a, 0xf8, 0xe4,  
0xbb, 0xd4, 0xab, 0x96, 0x71, 0xd6, 0xad, 0xf0, 0x00, 0x00, 0xaa, 0x55 };  
  
    Serial.println();  
    Serial.println("-- Test Send --");  
    setDevice(lamp2);  
    tryConnect(10);  
    memcpy( Msg_key, dev.key, KEY_LENGTH );  
    //----- Send message  
    sendMsg( testmsg, 84 ) ;  
  
    //----- Receive response  
    int ret = receiveMsg( ) ;  
  
    lmes.TotL = ret ;  
    InitLmes( 'M' );  
    memcpy( crypt_key, dev.key, 16 ) ;  
  
    parse_header();  
    parse_suffix();  
    getmessage();  
  
    DumpMsg( ) ;  
}
```

It uses a correct message (from tinytuya) in order to test the send receive function.

You should get a response 120 c such as

```
000055AA0000E52300000004000000680000000.....0000AA55
```