**ESP32C3:**

**Sync cmd: (46 bytes)**

C0 00 08 24 00 00 00 00 00 07 07 12 20 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 C0

**Rcv data: (112 bytes)**

C0 01 08 04 00 07 07 12 20 00 00 00 00 C0 C0 01 08 04 00 07 07 12 20 00 00 00 00 C0 C0 01 08 04 00 07 07 12 20 00 00 00 00 C0 C0 01 08 04 00 07 07 12 20 00 00 00 00 C0 C0 01 08 04 00 07 07 12 20 00 00 00 00 C0 C0 01 08 04 00 07 07 12 20 00 00 00 00 C0 C0 01 08 04 00 07 07 12 20 00 00 00 00 C0 C0 01 08 04 00 07 07 12 20 00 00 00 00 C0

**Read mac cmd: (28 bytes)**

C0 00 0A 04 00 00 00 00 00 44 88 00 60 C0

C0 00 0A 04 00 00 00 00 00 48 88 00 60 C0

**Rcv data: (28 bytes)**

**Eg:** 7C-DF-A1-01-5B-E6

C0 01 0A 04 00 E6 5B 01 A1 00 00 00 00 C0

C0 01 0A 04 00 DF 7C 00 00 00 00 00 00 C0

**Rcv data: (30 bytes)**

30-ae-a4-d8-af-c0

C0 01 0A 04 00 DB DC AF D8 A4 00 00 00 00 C0

C0 01 0A 04 00 AE 30 DB DD 00 00 00 00 00 C0

**ESP32S2:**

**Sync cmd: (46 bytes)**

C0 00 08 24 00 00 00 00 00 07 07 12 20 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 C0

**Rcv data: (112 bytes)**

C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0

**Read mac cmd: (28 bytes)**

C0 00 0A 04 00 00 00 00 00 44 A0 41 3F C0

C0 00 0A 04 00 00 00 00 00 48 A0 41 3F C0

**Rcv data: (28 bytes)**

**Eg:** 7C-DF-A1-01-5B-E6

C0 01 0A 04 00 E6 5B 01 A1 00 00 00 00 C0

C0 01 0A 04 00 DF 7C 00 00 00 00 00 00 C0

**Rcv data: (30 bytes)**

30-ae-a4-d8-af-c0

C0 01 0A 04 00 DB DC AF D8 A4 00 00 00 00 C0

C0 01 0A 04 00 AE 30 DB DD 00 00 00 00 00 C0

**ESP32:**

**Sync cmd: (46 bytes)**

C0 00 08 24 00 00 00 00 00 07 07 12 20 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 C0

**Rcv data: (112 bytes)**

C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0 C0 01 08 04 00 07 12 20 55 00 00 00 00 C0

**Read mac cmd: (28 bytes)**

C0 00 0A 04 00 00 00 00 00 04 A0 01 60 C0 C0 00 0A 04 00 00 00 00 00 08 A0 01 60 C0

**Rcv data: (28 bytes)**

**Eg:** 34CE00894223

C0 01 0A 04 00 23 42 89 00 00 00 00 00 C0

C0 01 0A 04 00 CE 34 2B 00 00 00 00 00 C0

**Rcv data: (30 bytes)**

30-ae-a4-d8-af-c0

C0 01 0A 04 00 DB DC AF D8 A4 00 00 00 00 C0

C0 01 0A 04 00 AE 30 DB DD 00 00 00 00 00 C0

**ESP8266:**

**Sync cmd: (46 bytes)**

C0 00 08 24 00 00 00 00 00 07 07 12 20 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 C0

**Rcv data: (96 bytes)**

C0 01 08 02 00 07 07 12 20 00 00 C0 C0 01 08 02 00 07 07 12 20 00 00 C0 C0 01 08 02 00 07 07 12 20 00 00 C0 C0 01 08 02 00 07 07 12 20 00 00 C0 C0 01 08 02 00 07 07 12 20 00 00 C0 C0 01 08 02 00 07 07 12 20 00 00 C0 C0 01 08 02 00 07 07 12 20 00 00 C0 C0 01 08 02 00 07 07 12 20 00 00 C0

**Read mac cmd: (42 bytes)**

C0 00 0A 04 00 00 00 00 00 50 00 F0 3F C0 C0 00 0A 04 00 00 00 00 00 54 00 F0 3F C0 C0 00 0A 04 00 00 00 00 00 5C 00 F0 3F C0

**Rcv data: (36 bytes)**

**Eg:** DC4F2204DEA9

C0 01 0A 00 00 00 00 EC A9 00 00 C0

C0 01 0A 00 00 DE 04 00 02 00 00 C0

C0 01 0A 00 00 22 4F DC 00 00 00 C0

**附说明：**

发送指令说明：

数据格式：C0 00 0A 04 00 00 00 00 00 \*\* \*\* \*\* \*\* C0，其中\*\* \*\* \*\* \*\* ：表示寄存器地址

1. C0 00 0A 04 00 00 00 00 00 04 A0 01 60 C0
2. C0 00 0A 04 00 00 00 00 00 08 A0 01 60 C0

返回指令说明：

数据格式：C0 01 0A 04 00 \*\* \*\* \*\* \*\* 00 00 00 00 C0 ，其中\*\* \*\* \*\* \*\*表示对应寄存器的返回值。

注意：若\*\* \*\* \*\* \*\*出现以下情况，需要做数据替换。

**DB DC ------>C0**

**DB DD ------>DB**

**ESP32**

04 A0 01 60寄存器对应的为MAC地址低4字节

08 A0 01 60寄存器对应的为MAC地址高2字节

**ESP8266**

读取返回三个寄存器的返回值。

如例子：C0 01 0A 00 00 00 00 EC A9 00 00 C0

C0 01 0A 00 00 DE 04 00 02 00 00 C0

C0 01 0A 00 00 22 4F DC 00 00 00 C0

分别对应如下寄存器的值。

50 00 F0 3F

54 00 F0 3F

5C 00 F0 3F

其计算MAC地址的算法如下：

mac0 = self.read\_reg(self.ESP\_OTP\_MAC0)

mac1 = self.read\_reg(self.ESP\_OTP\_MAC1)

mac3 = self.read\_reg(self.ESP\_OTP\_MAC3)

if (mac3 != 0):

oui = ((mac3 >> 16) & 0xff, (mac3 >> 8) & 0xff, mac3 & 0xff)

elif ((mac1 >> 16) & 0xff) == 0:

oui = (0x18, 0xfe, 0x34)

elif ((mac1 >> 16) & 0xff) == 1:

oui = (0xac, 0xd0, 0x74)

else:

raise FatalError("Unknown OUI")

return oui + ((mac1 >> 8) & 0xff, mac1 & 0xff, (mac0 >> 24) & 0xff)

其中mac0,mac1,mac2分别为00 00 EC A9， DE 04 00 02，22 4F DC 00的反转值。