حافظه EEPROM در SOC و ماژول های وایفای ESP8266

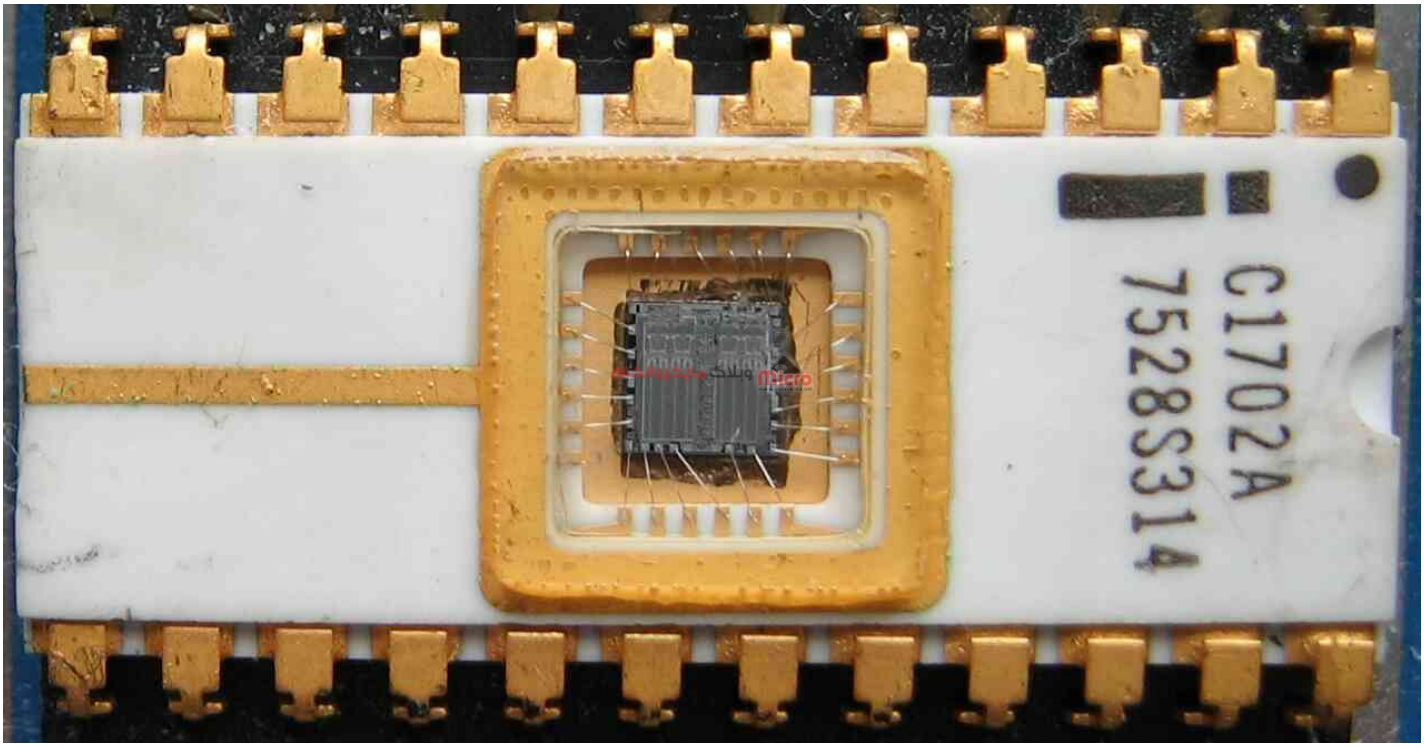
تاریخ انتشار۱۳ تیر, ۱۴۰۱ توسط سید حسین سلطانی

سلام خدمت همه شما مایکروالکامی ها. در مطالب قبلی از سری مطالب مربوط به ماژول WiFi مدل ESP8266 به راه اندازی یک لوکال وب سرور و ایجاد یک فرم برای دریافت اطلاعات از کاربر پرداخته شد. گاهی نیاز است که یکسری دیتا و مشخصاتی را در حافظه خود ذخیره کنیم. برای جلوگیری از بین رفتن این اطلاعات از حافظه های EEPROM می‌توان استفاده کرد. در این مطلب به نحوه استفاده از این حافظه در ماژول وایفای ESP8266 پرداخته خواهد شد. پس با من تا انتهای مطلب همراه باشید. همچنین شما میتوینند سایر مطالب من رو از این لینک مطالعه و بررسی کنید.

حافظه های EEPROM و FLASH

حافظه های EEPROM از دسته حافظه هایی هستند که با قطع شدن برق، اطلاعات ذخیره شده در آنها از بین نخواهد رفت. برای پاک کردن اطلاعات از روی این تراشه ها نیاز به سیگنال های الکتریکی بوده و براحتی این کار انجام خواهد شد.

نوع دیگر حافظه که پیش از EEPROM ها معرفی و بکار گرفته شده بود به EPROM معروف بوده است. برای پاک کردن دیتا از روی این تراشه، عملا نیازی به نور ماوراء بنفش بوده و بایستی به لنز شیشه ای تعبیه شده برروی تراشه به همین منظور استفاده می‌شد. طبیعتا برای محافظت از دیتا ذخیره شده از پاک شدن آن در صورت عدم نیاز، بایستی حتما محفظه شیشه ای موجود روی تراشه را پوشانید.



تصویری از حافظه *EPROM*

پس از EPROM ها EEPROM ها معرفی شد و مشکل نسل قبلی خود را مرتفع نمود. با این وجود باز هم در سرعت عملکرد آنچنان که باید سریع نبود. در این بین در سال 1980 شرکت توشیبا تراشه حافظه جدیدی تحت عنوان حافظه

FLASH را معرفی و مشکلات مربوط به ROM ها را رفع کرد. این حافظه گونه ای از EEPROM بوده اما بجای دسترسی بایت به بایت به حافظه، بصورت بلوکی به خانه های حافظه دسترسی و ارتباط خواهیم داشت. تفاوت دیگری که بین EEPROM و FLASH وجود دارد نوع آن است. EEPROM از نوع NOR بوده اما FLASH ها از حافظه های نوع NAND بوده و همین منجر به افزایش سرعت این حافظه ها شده است.

در مطالب قبلی، نحوه ارتقا و افزایش حافظه فلش موجود برروی ماژول های ESP8266 توضیح داده شده است. برای اطلاع بیشتر و نحوه انجام این مورد می‌توانید از این لینک اقدام فرمایید.

قطعات مورد نیاز

- ماژول ESP8266
- برد بورد
- مبدل سریال
- سیم برد بوردی
- کلید فشاری
- مبدل کاهنده ولتاژ

ایجاد EEPROM در ماژول های وایفای ESP8266

تراشه های ESP2866 در حقیقت فاقد حافظه EEPROM برروی خود می‌باشند. پس چگونه می‌توان این حافظه را پیاده سازی کرد؟ برای اینکار اصطلاحا با حافظه FLASH موجود برروی ماژول یا در کنار ESP8266، حافظه EEPROM را ساخته یا شبیه سازی می‌کنیم. همچنین برای پیاده سازی آن می‌توان با استفاده از کتابخانه EEPROM.h و توابع موجود در Arduino IDE استفاده کرد.

ذکر این نکته مهم است که باید در نظر داشت طبیعتاً از کل فضای حافظه FLASH نمی‌توان بعنوان حافظه EEPROM استفاده کرد. چرا که بخشی از آن در اختیار برنامه اصلی بوده و فقط می‌توان فضایی محدود را برای استفاده از آن بعنوان EEPROM در نظر گرفت. مابقی در اختیار برنامه نوشته شده و پروگرام شده قرار خواهد گرفت. البته اگر حافظه فلش را مطابق با توضیحات قبلی افزایش دهیم، میزان حافظه EEPROM بیشتری نیز در ESP8266 بدست خواهیم آورد.

حافظه *FLASH* و *EEPROM* در ماژول *ESP8266*

استفاده از کتابخانه EEPROM دقیقا مشابه استفاده از آن در برد های آردوینو بوده اما یک تفاوت دارد که بایستی در نظر گرفت. در ادامه تابع های مورد نیاز ذکر شده است. در میان این، تابع get() بایستی حتما استفاده شود. در غیر اینصورت عملکردی صحیح نخواهیم داشت. سایر توابع موجود در این کتابخانه نیز می‌تواند متناسب با نیاز کاربر براحتی مورد استفاده قرار گیرد.

- تابع begin() برای آماده سازی اولیه و اختصاص میزان حافظه مورد نیاز EEPROM
- تابع put() برای نوشتن دیتا روی EEPROM در ماژول وایفای ESP8266
- تابع get() برای خواندن دیتا از روی EEPROM در ماژول وایفای ESP8266
- تابع commit() برای عملکرد صحیح و انجام درست کار پس از نوشتن دیتا برروی حافظه

محاسبه میزان فضای مورد نیاز EEPROM

با استفاده از تابع begin() می‌توان میزان حافظه مورد نظر خود را اختصاص دهیم. در صورتیکه حافظه FLASH ما نوع 4MB باید باشد، حداکثر 4096B حافظه در اختیار خواهیم داشت. چنانچه دیتا مورد ذخیره سازی طیف استاندار و مشخص باشد می‌توان از جدول زیر استفاده کرد.

8 بایت	4 بایت	2 بایت	1 بایت
double	int	short	bool
long long	long	uint16_t	char
uint64_t	float		byte
	uint32_t		uint8_t

- شایان ذکر است با استفاده از تابع sizeof() می‌توان براحتی طول دیتا یا متغیر مورد نظر را بدست آورده و به همان میزان فضای مورد نظر را اختصاص دهیم.

مثال

نوع متغییر زیر که از نوع structure است را در نظر بگیرید. در حقیقت می‌خواهیم دو متغیر تعریف شده درون این ساختار را در EEPROM ایجاد شده ذخیره کنیم. بجای "MAX_STRING_LENGTH" می‌بایست طول اعضای آرایه خود را لحاظ کنیم.

```
struct {
    char name[MAX_STRING_LENGTH] = "";
    char family[MAX_STRING_LENGTH] = "";
} name_family;
```

برای شروع ابتدا به ساکن نیاز داریم توسط تابع EEPROM.begin() میزان حافظه مورد نیاز را مشخص کنیم. برای اینکه دو متغیر مشخص شده کامل پوشش داده شود از sizeof مشابه کد زیر استفاده خواهیم کرد.

```
EEPROM.begin(sizeof(settings));
```

نوشتن دیتا برروی EEPROM در ماژول ESP8266

روال کلی نوشتن دیتا برروی EEPROM ایجاد شده در ماژول ESP8266 آسان است. باید دقت کرد ابتدا سایز EEPROM مشخص شود. در مرحله بعد آدرس مورد نظر را اعمال و دیتا خود را برروی آن بنویسیم.

نکته خیلی مهم این است که پس از نوشتن دیتا، حتما از دستور commit() باید استفاده کرد. تکه کد زیر جهت نوشتن دیتا برروی EEPROM در ماژول ESP8266 می‌باشد.

برای آگاهی از نحوه پروگرام کردن *ESP8266* از این مطلب استفاده کنید.

```
#include <EEPROM.h>

void setup()
{
  Serial.begin(9600);
  Serial.println("Write Data on EEPROM:");
 //مشخص کردن فضای مورد نیاز خود برای EEPROM
  EEPROM.begin(5);

 //نوشتن دیتا در آدرس صفر ام از حافظه EEPROM
  EEPROM.write(0,1010);
  EEPROM.commit();
}

void loop()
{

}
```

نکته: برای نوشتن دیتا روی EEPROM می‌توان از تابع write() نیز استفاده کرد. از تابع write برای کار با یک بایت استفاده شده و از put برای کار با چند بایت از آدرس مشخص شده استفاده می‌شود. (چندین بایت را از آدرس مشخص شده می‌نویسد.)

خواندن از روی EEPROM در ماژول ESP8266

با استفاده از کد زیر و استفاده از تابع get()، می‌توان دیتا ذخیره شده در آدرس مشخص شده در آرگومان را دریافت و در متغیری ذخیره کرد. نحوه استفاده از این تابع در کد زیر مشخش شده است.

```
#include <EEPROM.h>
```

```
unsigned int eeprom_read;
void setup()
{
  Serial.begin(9600);
  Serial.println("Read Data from EEPROM:");
 // خواندن دیتا از آدرس صفر ام حافظه EEPROM و ذخیره آن در متغیر eeprom_read
  EEPROM.get(0, eeprom_read);
  Serial.print("Data: ");
  Serial.println(eeprom_read);
}

void loop()
{

}
```

نکته: برای خواندن دیتا از EEPROM می‌توان از تابع read() نیز استفاده کرد. از تابع read برای کار با یک بایت استفاده شده و از get برای کار با چند بایت از آدرس مشخص شده استفاده می‌شود. (چندین بایت را از آدرس مشخص شده می‌خواند.)

پاک کردن دیتا از روی EEPROM در ESP8266

اگر بنا به هر دلیلی خواستیم محتوای EEPROM در هر آدرس یا از هر آدرسی تا هر آدرس دیگری را پاک کنیم، کافی است از کدی مشابه زیر استفاده کنیم.

```
for (int i = 0;  i < Size;  EEPROM.write(i++, 0));
```

- برای پاک کردن دیتا در کل EEPROM کافی است از این دستور استفاده شود. با این دستور از آدرس صفر ام تا کل سایز EEPROM که با Size مشخص شده است، با صفر جایگزین می‌شود.

- اگر خواستیم از آدرس n تا m را فقط پاک کنیم، کافی است در کد بالا بجای i = 0 قرار دهیم i = n و همچنین بجای i < Size قرار دهیم i < m بقیه موارد سر جای خود باقی خواهد ماند.

مثال ذخیره دیتا در EEPROM ماژول ESP8266

نمونه کد زیر برای خواندن و نوشتن مقادیر مختلف در EEPROM در ماژول ESP8266 می‌باشد. **به کامنت های مشخص شده در کد های زیر برای گیرایی بیشتر حتما دقت نمایید.**

```
#include <EEPROM.h>
unsigned int address, data, read_data;
float temp, read_temp;
void setup()
{
 Serial.begin(9600);
 Serial.println("ESP8266 EEPROM Read and Write");

 EEPROM.begin(10);
 data = 1010;
 EEPROM.put(address, data);
//مشخص کردن آدرس جدید برای نوشتن دیتا های بعدی
 address += sizeof(data);
 temp = 29.2;
 EEPROM.put(address, temp);
 EEPROM.commit();

//خواندن دیتا از EEPROM
 address = 0;
 EEPROM.get(address, read_data);
 Serial.print("Read Data= ");
 Serial.println(read_data);
```

```
// مشخص کردن آدرس بعدی جهت خواندن از EEPROM
 address += sizeof(read_data);
 EEPROM.get(address, read_temp);
 /*
نیز مشابه زیر استفاده کرد readFloat بجای دستور بالا میتوان از دستور.
  read_temp = EEPROM.readFloat(address)
 */
 Serial.print("Read Temp= ");
 Serial.println(read_temp);
 EEPROM.end();
}
void loop()
{
}
```

عیب روش بیان شده

شاید جالب باشد بدانید طی نتیجه تستی که از این روش ذکر شده انجام شد، فقط در صورتیکه ESP8266 ری استارت یا قطع و وصل نشود، مناسب خواهد بود. به بیانی دیگر، در صورتیکه به هر دلیلی تغذیه ESP8266 قطع و وصل شود دیتا های نوشته شده در این حافظه از دست خواهد رفت و عملا با مقدار صفر جایگزین خواهند شد. این عیب ناشی از عدم وجود EEPROM در خود ماژول یا تراشه ESP8266 می‌باشد. در مطلب بعدی به حل این مشکل پرداخته خواهد شد.

نتیجه گیری

در این مطالب به نحوه استفاده و راه اندازی حافظه EEPROM و ذخیره اطلاعات در آن در ماژول های ESP8266 پرداخته شد. همچنین همانطور که بیان شد، با قطع و وصل تغذیه ESP8266 ریست شدن دیتا های ذخیره شده پاک شده و با مقادیر 0 جایگزین می‌گردد.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورت داشتن هرگونه نظر یا سوال درباره این آموزش اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ microelecom# به اشتراک بگذارید و پیج مایکروالکام (@microelecom) رو هم منشن کنید.