# گزارش‌کار در آزمایشگاه

در این بخش، گزارش کار آزمایشگاه به تفصیل و شرح لازم می‌بایست درج شود

(با درج عکس ها و کد های مرتبط - از نظر تعداد صفحه، محدودیتی ندارد و فراتر بودن گزارش از حد استاندارد، شامل نمره اضافه می‌گردد).

در متد اصلی

ابتدا رام سپس سیم ۸۰۰ و سپس وای فای را کانفیگ کن و لیست مخاطبین را نشان بده (درصورت وجود) و سپس اماده ارسال صغحه اصلی !

//========================================= Init ========================================= //

HAL\_UART\_Transmit(&huart6, "\r\n========Debug=======\r\n", 24, 1000);

HAL\_UART\_Transmit(&huart6, "EEPROM Init", 11, 1000);

EEPROM\_Init();

HAL\_UART\_Transmit(&huart6, ": OK\r\nSimCom Init", 17, 1000);

SIMCOM\_Init();

HAL\_UART\_Transmit(&huart6, ": OK\r\nESPMOD Init", 17, 1000);

ESP\_Init(Wifi\_Init);

HAL\_UART\_Transmit(&huart6, ": OK\r\n====================\r\n", 28, 1000);

EEPROM\_ListContacts();

HAL\_GPIO\_WritePin(GPIOD, LD3\_Pin | LD4\_Pin | LD5\_Pin | LD6\_Pin, GPIO\_PIN\_SET);

HAL\_Delay(800);

HAL\_GPIO\_WritePin(GPIOD, LD3\_Pin | LD4\_Pin | LD5\_Pin | LD6\_Pin, GPIO\_PIN\_RESET);

HAL\_UART\_Transmit(&huart6, "========Ready=======\r\n", 22, 1000);

//========================================= Init ========================================= //

در کالبک یورات وای فای

void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*huart)

{

if (huart->Instance == USART3)

{

اگر از یورات ۳ داده دریافت کردی داده را بخوان تا به قسمت (گت) برسی همچنین شماره کانکشن را بردار.

اگر به گت رسیدی تمام هدر را بخوان تا به کاراکتر فضای خالی برسی.

چون در ریکوست فضای خالی نداریم یعنی ریکوست تمام شده و میتوان آن را پردازش کرد.

if (request == 1)

{

if (EspRxBuffer[0] == ' ' || RequestHeaderCounter == sizeof(RequestHeader))

{

//RequestHeaderCounter = 0; we need this to calc tx buffer size

request = 0;

ready = 1;

}

else

{

RequestHeader[RequestHeaderCounter] = EspRxBuffer[0];

RequestHeaderCounter++;

}

}

خواندن شماره کلاینت

EspRxArray[rxCounter] = EspRxBuffer[0];

if (EspRxArray[rxCounter - 4] == 'I' && EspRxArray[rxCounter - 3] == 'P' && EspRxArray[rxCounter - 2] == 'D')

{

clientNumber = EspRxBuffer[0] - 48;

}

if (EspRxArray[rxCounter - 4] == 'G' && EspRxArray[rxCounter - 3] == 'E' && EspRxArray[rxCounter - 2] == 'T')

// && EspRxArray[rxCounter - 1] == ' ' && EspRxArray[rxCounter] == '/'

{

request = 1;

// Make ESP Rx Buffer to Zero Byte

rxCounter = 0;

memset(EspRxArray, 0, sizeof(EspRxArray));

memset(RequestHeader, 0, sizeof(RequestHeader));

RequestHeaderCounter = 0;

HAL\_UART\_Receive\_IT(&huart3, EspRxBuffer, 1);

}

else

{

HAL\_UART\_Receive\_IT(&huart3, EspRxBuffer, 1);

}

rxCounter++;

}

}

در حلقه اصلی اگر داده ریکوست اماده بود میتوانی ان را پردازش کنی

if (ready)

{

ESP\_HandleRequset();

//SIM800\_HandleRequset();

ready = 0;

HAL\_UART\_Receive\_IT(&huart3, EspRxBuffer, 1);

}

اگر به ارور رسیدی اینطور نمایش بده تا متوجه بشیم ! با کمک LED

void Error\_Handler(void)

{

/\* USER CODE BEGIN Error\_Handler\_Debug \*/

/\* User can add his own implementation to report the HAL error return state \*/

HAL\_UART\_Transmit(&huart6, "\r\n====================", 22, 1000);

HAL\_UART\_Transmit(&huart6, "\r\nAn Error occured\r\n", 20, 1000);

HAL\_UART\_Transmit(&huart6, "====================\r\n", 22, 1000);

while (1)

{

HAL\_GPIO\_WritePin(LD3\_GPIO\_Port, LD3\_Pin, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(LD4\_GPIO\_Port, LD4\_Pin, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(LD5\_GPIO\_Port, LD5\_Pin, GPIO\_PIN\_SET);

HAL\_GPIO\_WritePin(LD6\_GPIO\_Port, LD6\_Pin, GPIO\_PIN\_SET);

HAL\_Delay(1000);

HAL\_GPIO\_WritePin(LD3\_GPIO\_Port, LD3\_Pin, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(LD4\_GPIO\_Port, LD4\_Pin, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(LD5\_GPIO\_Port, LD5\_Pin, GPIO\_PIN\_RESET);

HAL\_GPIO\_WritePin(LD6\_GPIO\_Port, LD6\_Pin, GPIO\_PIN\_RESET);

HAL\_Delay(1000);

}

/\* USER CODE END Error\_Handler\_Debug \*/

}

متدهای AT کاملا گویا بوده و نیازی به توضیح اضافی ندارند

راه اندازی اولیه سیم ۸۰۰

void SIMCOM\_Init()

{

// sprintf(buffer, "AT\r\n");

// HAL\_UART\_Transmit(&huart2, buffer, 4, HAL\_MAX\_DELAY);

// HAL\_Delay(1000);

sprintf(buffer, "AT+CNMI=1, 2, 0, 0, 0\r\n");

HAL\_UART\_Transmit(&huart2, buffer, 23, HAL\_MAX\_DELAY);

HAL\_Delay(1000);

sprintf(buffer, "AT+CBAND=\"ALL\_BAND\"\r\n");

HAL\_UART\_Transmit(&huart2, buffer, 21, HAL\_MAX\_DELAY);

//HAL\_Delay(5000);

}

متد ارسال پیامک با توجه به دیتاشیت at

void SIMCOM\_SendSMS()

{

sprintf(buffer, "AT+CMGF=1\r\n");

HAL\_UART\_Transmit(&huart2, buffer, 11, HAL\_MAX\_DELAY);

HAL\_Delay(1000);

sprintf(buffer, "AT+CMGS=\"%s\"\r\n", ContactNumberTemp);

HAL\_UART\_Transmit(&huart2, buffer, 23, HAL\_MAX\_DELAY);

HAL\_Delay(2000);

sprintf(buffer, "%s\r\n", TextMessage);

HAL\_UART\_Transmit(&huart2, buffer, TextLength + 2, HAL\_MAX\_DELAY);

HAL\_Delay(1000);

sprintf(buffer, "%c", 0x1a);

HAL\_UART\_Transmit(&huart2, buffer, 1, HAL\_MAX\_DELAY);

HAL\_Delay(1000);

}

متد ارسال و قطع تماس

void SIMCOM\_MakeCall()

{

sprintf(buffer, "ATD%s;\r\n", ContactNumberTemp);

HAL\_UART\_Transmit(&huart2, buffer, 17, HAL\_MAX\_DELAY);

}

void SIMCOM\_EndCall()

{

sprintf(buffer, "ATH\r\n");

HAL\_UART\_Transmit(&huart2, buffer, 5, HAL\_MAX\_DELAY);

}

راه اندازی اولیه رام . اگر حافظه پاک شده باشد. برای ان اختصاص داده میشود

void EEPROM\_Init()

{

HAL\_FLASH\_Unlock();

EE\_Init();

HAL\_FLASH\_Lock();

HAL\_FLASH\_Unlock();

//get count of contacts in eeprom

ReadAddr = VirtAddVarTab[ContactCountROMAddrIndex];

WriteAddr = VirtAddVarTab[ContactCountROMAddrIndex];

if (EE\_ReadVariable(ReadAddr, &ContactCount) != HAL\_OK)

{

HAL\_UART\_Transmit(&huart6, "Full erase detected!\r\n", 22, HAL\_MAX\_DELAY);

//so if not defined set ContactCount to 0

HAL\_FLASH\_Unlock();

EE\_WriteVariable(WriteAddr, ContactCount);

HAL\_FLASH\_Lock();

}

else

{

// sprintf(buffer, "Number of saved Contacts: %d\r\n", ContactCount);

// HAL\_UART\_Transmit(&huart6, buffer, 29, HAL\_MAX\_DELAY);

}

HAL\_FLASH\_Lock();

}

گرفتن لیست مخاطبین

void EEPROM\_ListContacts()

{

اگر مخاطبی نبود

if (ContactCount == 0)

{

HAL\_UART\_Transmit(&huart6, "No Contact\r\n", 12, HAL\_MAX\_DELAY);

return;

}

در غیر اینصورت اختصاص مخاطب تمپ

sprintf(buffer, "Contact List (%d):\r\n", ContactCount);

HAL\_UART\_Transmit(&huart6, buffer, 19, HAL\_MAX\_DELAY);

unsigned char tempName[ContactNameMaxLength];

uint16\_t tempNameLength = 0;

unsigned char tempNumber[ContactNumberLength];

uint16\_t temp;

برای هر مخاطب ذخیره شده ابتدا شماره و سپس نام متناظر را پیدا کن

ادرس تولید شده در حلقه یک عدد یک عدد جلو میرد

و بر اساس شماره حلقه ضرب در طول شماره تلفن بعلاوه ایندکس بدست میآید.

این الگو در تمام کد استفاده شده است.

// For each contact

for (int i = 0; i < ContactCount; i++)

{

// Read Number

ReadAddr = VirtAddVarTab[ContactNumberROMAddrIndex] + (i \* ContactNumberLength);

for (int j = 0; j < ContactNumberLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

EE\_ReadVariable(ReadAddr, &temp);

HAL\_FLASH\_Lock();

tempNumber[j] = temp;

}

// Read Name

memset(tempName, 0, sizeof(tempName));

tempNameLength = 0;

ReadAddr = VirtAddVarTab[ContactNameROMAddrIndex] + (i \* ContactNameMaxLength);

for (int j = 0; j < ContactNameMaxLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

if (EE\_ReadVariable(ReadAddr, &temp) != HAL\_OK)

{

break; // Its not because an error, the name ENDS if it cant read.

}

HAL\_FLASH\_Lock();

if ((temp >= 0x41 && temp <= 0x5A) || (temp >= 0x61 && temp <= 0x7a))

{

tempNameLength++;

tempName[j] = temp;

}

}

اکنون ان را چاپ کن

// sprintf(buffer, "Contact Name Length:%d\r\n", tempNameLength);

// HAL\_UART\_Transmit(&huart6, buffer, 23, 10000);

memset(buffer, 0, sizeof(buffer));

sprintf(buffer, "%d- %s, %s", i + 1, tempName, tempNumber);

HAL\_UART\_Transmit(&huart6, buffer, 5 + tempNameLength + ContactNumberLength, 10000);

HAL\_UART\_Transmit(&huart6, "\r\n", 2, 10000);

}

}

اضافه کردن مخاطب جدید

int EEPROM\_AddContact()

{

HAL\_UART\_Transmit(&huart6, "Adding Contact in EEPROM\r\n", 26, HAL\_MAX\_DELAY);

بررسی نام و یا شماره تکراری

if (EEPROM\_GetName() != -1)

{

sprintf(ErrorText, "contact with this number exists, its name is: %s", ContactNameTemp);

ErrorTextLength = 46 + ContactNameLength;

return -1;

}

if (EEPROM\_GetNumber() != -1)

{

sprintf(buffer, "contact with this name exists, its number is: %s", ContactNumberTemp);

ErrorTextLength = 46 + ContactNumberLength;

return -1;

}

حال که تکراری نیست ابتدا شماره را ذخیره کن

// Write Number

HAL\_FLASH\_Unlock();

//add contact to the end of list

WriteAddr = VirtAddVarTab[ContactNumberROMAddrIndex] + (ContactCount \* ContactNumberLength);

for (int i = 0; i < ContactNumberLength; i++, WriteAddr++)

{

if (ContactNumber[i] >= 0x30 && ContactNumber[i] <= 0x39)

EE\_WriteVariable(WriteAddr, ContactNumber[i]);

}

HAL\_FLASH\_Lock();

ذخیره نام

// Write Name

HAL\_FLASH\_Unlock();

WriteAddr = VirtAddVarTab[ContactNameROMAddrIndex] + (ContactCount \* ContactNameMaxLength);

//Note : Writeadd use MAX length but FOR use Lenght from HTTP Header

for (int i = 0; i < ContactNameMaxLength; i++, WriteAddr++)

{

if ((ContactName[i] >= 0x41 && ContactName[i] <= 0x5A) || (ContactName[i] >= 0x61 && ContactName[i] <= 0x7a))

{

EE\_WriteVariable(WriteAddr, ContactName[i]);

}

else

{

break;

}

}

HAL\_FLASH\_Lock();

ذخیره شماره تعداد محاطبین

// Write Contact Count

ContactCount = ContactCount + 1;

HAL\_FLASH\_Unlock();

WriteAddr = VirtAddVarTab[ContactCountROMAddrIndex];

EE\_WriteVariable(WriteAddr, ContactCount);

HAL\_FLASH\_Lock();

// memset(buffer, 0, sizeof(buffer));

// sprintf(buffer, "%s, %s Added Successfully\r\n", ContactName, ContactNumber);

// HAL\_UART\_Transmit(&huart6, buffer, 23 + ContactNameLength + ContactNumberLength, 10000);

نمایش لیست مخاطبین بروز شده

EEPROM\_ListContacts();

return 0;

}

دریافت نام مخاطب به کمک شماره

int EEPROM\_GetName()

{

// We have Number => find number index => find index => find name

HAL\_UART\_Transmit(&huart6, "Checking for Number in EEPROM\r\n", 31, HAL\_MAX\_DELAY);

int index = -1;

uint16\_t temp;

unsigned char tempNumber[ContactNumberLength];

اختصاص شماره تمپ و گردش بین تمام شماره تلفن ها برای پیدا کردن ایندکس

// For each contact

for (int i = 0; i < ContactCount; i++)

{

// Read Number

ReadAddr = VirtAddVarTab[ContactNumberROMAddrIndex] + (i \* ContactNumberLength);

for (int j = 0; j < ContactNumberLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

EE\_ReadVariable(ReadAddr, &temp);

HAL\_FLASH\_Lock();

tempNumber[j] = temp;

}

بررسی شماره برای برابری

for (int j = 0; j < ContactNumberLength; j++)

{

if (tempNumber[j] != ContactNumber[j])

break;

else if (j == (ContactNumberLength - 1))

index = i;

}

}

if (index != -1)

{

اگر شماره پیدا شد با توجه به ایندکس ، نام را از حافظه بخوان

memset(ContactName, 0, sizeof(ContactName));

// Read Name

ContactNameLength = 0;

ReadAddr = VirtAddVarTab[ContactNameROMAddrIndex] + (index \* ContactNameMaxLength);

for (int j = 0; j < ContactNameMaxLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

if (EE\_ReadVariable(ReadAddr, &temp) != HAL\_OK)

{

break; // Its not because an error, the name ENDS if it cant read.

}

HAL\_FLASH\_Lock();

if ((temp >= 0x41 && temp <= 0x5A) || (temp >= 0x61 && temp <= 0x7a))

{

ContactNameLength++;

ContactNameTemp[j] = temp;

}

}

اطلاعات را چاپ کن

memset(buffer, 0, sizeof(buffer));

sprintf(buffer, "Found Contact with this Number, Name is: %s\r\n", ContactNameTemp);

HAL\_UART\_Transmit(&huart6, buffer, 43 + ContactNameLength, HAL\_MAX\_DELAY);

}

else

{

sprintf(buffer, "Contact Number doesnt exist\r\n");

HAL\_UART\_Transmit(&huart6, buffer, 29, 10000);

index = -1;

}

HAL\_FLASH\_Lock();

return index;

}

دریافت شماره مخاطب به کمک نام

int EEPROM\_GetNumber()

{

این متد دقیقا شبیه متد قبل عمل میکند با این تفاوت که بجای شماره ، نام و بجای نام ، شماره جایگذاری شده است

// We have name => find name index => find index

HAL\_UART\_Transmit(&huart6, "Checking for Name in EEPROM\r\n", 29, HAL\_MAX\_DELAY);

int index = -1;

uint16\_t temp;

unsigned char tempName[ContactNameMaxLength];

uint16\_t tempNameLength = 0;

// For each contact

for (int i = 0; i < ContactCount; i++)

{

// Read Name

memset(tempName, 0, sizeof(tempName));

tempNameLength = 0;

ReadAddr = VirtAddVarTab[ContactNameROMAddrIndex] + (i \* ContactNameMaxLength);

for (int j = 0; j < ContactNameMaxLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

if (EE\_ReadVariable(ReadAddr, &temp) != HAL\_OK)

{

break; // Its not because an error, the name ENDS if it cant read.

}

HAL\_FLASH\_Lock();

if ((temp >= 0x41 && temp <= 0x5A) || (temp >= 0x61 && temp <= 0x7a))

{

tempNameLength++;

tempName[j] = temp;

}

}

for (int j = 0; j < ContactNameLength; j++)

{ // upper and lower case check

if (tempName[j] >= 0x41 && tempName[j] <= 0x5A && ContactName[j] >= 0x61 && ContactName[j] <= 0x7a)

{

ContactName[j] -= 0x20;

}

if (tempName[j] >= 0x61 && tempName[j] <= 0x7a && ContactName[j] >= 0x41 && ContactName[j] <= 0x5A)

{

tempName[j] -= 0x20;

}

if (tempName[j] != ContactName[j])

break;

else if ((j == (ContactNameLength - 1)) && (ContactNameLength == tempNameLength))

index = i;

}

}

if (index != -1)

{

// Read Number

ReadAddr = VirtAddVarTab[ContactNumberROMAddrIndex] + (index \* ContactNumberLength);

for (int j = 0; j < ContactNumberLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

if (EE\_ReadVariable(ReadAddr, &temp) != HAL\_OK)

{

break; // Its not because an error, the name ENDS if it cant read.

}

HAL\_FLASH\_Lock();

ContactNumberTemp[j] = temp;

}

sprintf(buffer, "Found Contact with this Name, Number is: %s\r\n", ContactNumberTemp);

HAL\_UART\_Transmit(&huart6, buffer, 43 + ContactNumberLength, HAL\_MAX\_DELAY);

}

else

{

sprintf(buffer, "Contact Name doesnt exist\r\n");

HAL\_UART\_Transmit(&huart6, buffer, 27, 10000);

}

return index;

}

راه اندازی اولیه ماژول وای فای

با توجه به مطالب تدریس شده این قسمت نیازی به توضیح ندارد !

void ESP\_Init(int FullInit)

{

// HAL\_UART\_Transmit(&huart3, "AT\r\n", sizeof("AT\r\n"), 1000);

// HAL\_Delay(1000);

if (FullInit != 0)

{

HAL\_UART\_Transmit(&huart3, "AT+CWMODE=1\r\n", 13, 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, "AT+CWLAP\r\n", 10, 1000);

HAL\_Delay(5000);

HAL\_Delay(10000);

}

HAL\_UART\_Transmit(&huart3, "AT+CIPSTA=\"192.168.21.120\",\"192.168.21.1\",\"255.255.255.0\"\r\n", 59, 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, "AT+CWJAP=\"ZMI-MF885\",\"51176915\"\r\n", 33, 1000);

HAL\_Delay(10000);

HAL\_UART\_Transmit(&huart3, "AT+CIPMUX=1\r\n", 13, 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, "AT+CIPSERVER=1,80\r\n", 19, 1000);

//HAL\_Delay(1000);

//HAL\_UART\_Transmit(&huart3, "AT+CIFSR\r\n", sizeof("AT+CIFSR\r\n"), 1000);

//HAL\_Delay(1000);

HAL\_UART\_Receive\_IT(&huart3, EspRxBuffer, 1);

}

هندل کردن ریکوست دریافتی

void ESP\_HandleRequset()

{

//Debug

HAL\_UART\_Transmit(&huart6, "\r\n====================\r\n", 24, 1000);

sprintf(buffer, "Client %d Connected\r\n", clientNumber);

HAL\_UART\_Transmit(&huart6, buffer, 20, 1000);

sprintf(buffer, "Request is %s\r\n", RequestHeader); // Have 13 word

HAL\_UART\_Transmit(&huart6, buffer, 13 + RequestHeaderCounter, 1000);

//Debug

memset(ContactName, 0, sizeof(ContactName));

memset(ContactNumber, 0, sizeof(ContactNumber));

memset(TextMessage, 0, sizeof(TextMessage));

بررسی اینکه ریکوست چیست

کد کاملا گویاست !

if (RequestHeader[0] == 'c' && RequestHeader[1] == 'o' && RequestHeader[2] == 'n' && RequestHeader[3] == 't' && RequestHeader[4] == 'a' && RequestHeader[5] == 'c' && RequestHeader[6] == 't')

ESP\_HandleContact();

else if (RequestHeader[0] == 's' && RequestHeader[1] == 'm' && RequestHeader[2] == 's')

ESP\_HandleSMS();

else if (RequestHeader[0] == 'e' && RequestHeader[1] == 'n' && RequestHeader[2] == 'd' && RequestHeader[3] == 'c' && RequestHeader[4] == 'a' && RequestHeader[5] == 'l' && RequestHeader[6] == 'l')

ESP\_HandleEndCall();

else if (RequestHeader[0] == 'c' && RequestHeader[1] == 'a' && RequestHeader[2] == 'l' && RequestHeader[3] == 'l')

ESP\_HandleCall();

else if (RequestHeader[0] == 'e' && RequestHeader[1] == 'r' && RequestHeader[2] == 'a' && RequestHeader[3] == 's' && RequestHeader[4] == 'e')

ESP\_HandleErase();

else

ESP\_SendIndex();

}

اگر درخواست پاک کردن حافظه داشتیم شماره مخاطبین را صفر کن و برگرد

void ESP\_HandleErase()

{

HAL\_UART\_Transmit(&huart6, "Handle Erase Req ....\r\n", 22, 1000);

HAL\_UART\_Transmit(&huart6, "======Removing======\r\n", 22, 1000);

EEPROM\_ListContacts();

ContactCount = 0;

HAL\_FLASH\_Unlock();

WriteAddr = VirtAddVarTab[ContactCountROMAddrIndex];

EE\_WriteVariable(WriteAddr, ContactCount);

HAL\_FLASH\_Lock();

HAL\_UART\_Transmit(&huart6, "========Done========\r\n", 22, 1000);

ESP\_SendSuccess();

}

درخواست تماس

نام را بگیر و شماره تلفن را استخراج کن اگر مشکلی نبود => تماس بگیر

اگر مشکل بود صفحه ارور ارسال کن

void ESP\_HandleCall()

{

HAL\_UART\_Transmit(&huart6, "Handle Call Req ....\r\n", 22, 1000);

int haveReq = 0;

for (uint8\_t i = 0; i < RequestHeaderCounter; i++)

if (RequestHeader[i - 5] == 'n' && RequestHeader[i - 4] == 'a' && RequestHeader[i - 3] == 'm' && RequestHeader[i - 2] == 'e')

haveReq = 1;

if (haveReq)

{

if ((ESP\_HandleHeaderParam() == 0) && (EEPROM\_GetNumber() != -1))

{

HAL\_UART\_Transmit(&huart6, buffer, 13 + ContactNumberLength, HAL\_MAX\_DELAY);

HAL\_UART\_Transmit(&huart6, "Making Call\r\n", 13, 1000);

SIMCOM\_MakeCall();

ESP\_SendTalking();

}

else

{

ESP\_SendError(); // please enter valid name and number

}

}

else

{

ESP\_SendCall();

}

}

پایان تماس => کامند پایان را بفرست و صفحه اصلی را ارسال کن

void ESP\_HandleEndCall()

{

HAL\_UART\_Transmit(&huart6, "Handle Hang up Req ....\r\n", 25, 1000);

HAL\_UART\_Transmit(&huart6, "Ending Call ....\r\n", 18, 1000);

SIMCOM\_EndCall();

ESP\_SendIndex();

}

صفحه مخاطبین : نام و شماره را بگیر اگر به حافظه اضافه شد صفحه موفقیت ذخیره را ارسال کن وگرنه صفحه ارور

به همراه ارور و توضیحات

اگر هم فقط صفحه اصلی درخواست شده بود همان را بفرست!

void ESP\_HandleContact()

{

HAL\_UART\_Transmit(&huart6, "Handle Contact Req ....\r\n", 25, 1000);

int haveReq = 0;

for (uint8\_t i = 0; i < RequestHeaderCounter; i++)

if (RequestHeader[i - 5] == 'n' && RequestHeader[i - 4] == 'a' && RequestHeader[i - 3] == 'm' && RequestHeader[i - 2] == 'e')

haveReq = 1;

if (haveReq)

{

if (ESP\_HandleHeaderParam() == 0 && (EEPROM\_AddContact() == 0))

{

//ESP\_SendContactSaved();

ESP\_SendSuccess();

}

else

{

ESP\_SendError(); // please enter valid name and number

}

}

else

{

ESP\_SendContact();

}

}

دقیقا مشابه با صفحه تماس ولی با هدف ارسال پیامک : نام را بگیر و شماره را بخوان و پیام را به متد ارسال پیام بده تا ارسال شود. خطا در صورت مشکل

void ESP\_HandleSMS()

{

HAL\_UART\_Transmit(&huart6, "Handle SMS Req ....\r\n", 21, 1000);

int haveReq = 0;

for (uint8\_t i = 0; i < RequestHeaderCounter; i++)

if (RequestHeader[i - 5] == 'n' && RequestHeader[i - 4] == 'a' && RequestHeader[i - 3] == 'm' && RequestHeader[i - 2] == 'e')

haveReq = 1;

if (haveReq)

{

if ((ESP\_HandleHeaderParam() == 0) && (EEPROM\_GetNumber() != -1))

{

HAL\_UART\_Transmit(&huart6, "Sending SMS\r\n", 13, 1000);

SIMCOM\_SendSMS();

//ESP\_SendSMSSent();

ESP\_SendSuccess();

}

else

{

ESP\_SendError();

return;

}

}

else

{

ESP\_SendSMS();

}

}

اکنون وظیفه دریافت نام و شماره و تکست به عهده این تابع است

int ESP\_HandleHeaderParam()

{

شماره های اغاز و پایان

uint8\_t numberIndex;

uint8\_t nameIndex;

uint8\_t numberEndIndex;

uint8\_t nameEndIndex;

uint8\_t textIndex;

uint8\_t textEndIndex;

uint8\_t haveName = 0;

uint8\_t haveText = 0;

uint8\_t haveNumber = 0;

اگر در هدر شماره بود ایندکس را پیدا کن

for (uint8\_t i = 0; i < RequestHeaderCounter; i++)

{

if (RequestHeader[i - 7] == 'n' && RequestHeader[i - 6] == 'u' && RequestHeader[i - 5] == 'm' && RequestHeader[i - 4] == 'b' && RequestHeader[i - 3] == 'e' && RequestHeader[i - 2] == 'r')

{

numberIndex = i;

numberEndIndex = RequestHeaderCounter - 1;

//ContactNumberLength = numberEndIndex - numberIndex = 11

haveNumber = 1;

break;

}

}

اگر در هدر تکست بود ایندکس را پیدا کن

for (uint8\_t i = 0; i < RequestHeaderCounter; i++)

{

if (RequestHeader[i - 5] == 't' && RequestHeader[i - 4] == 'e' && RequestHeader[i - 3] == 'x' && RequestHeader[i - 2] == 't')

{

textIndex = i;

textEndIndex = RequestHeaderCounter - 1;

TextLength = textEndIndex - textIndex + 1;

haveText = 1;

break;

}

}

اگر در هدر نام بود ایندکس را پیدا کن

for (uint8\_t i = 0; i < RequestHeaderCounter; i++)

{

if (RequestHeader[i - 5] == 'n' && RequestHeader[i - 4] == 'a' && RequestHeader[i - 3] == 'm' && RequestHeader[i - 2] == 'e')

{

nameIndex = i;

if (haveNumber == 1)

{

nameEndIndex = numberIndex - 9;

}

if (haveText == 1)

{

nameEndIndex = textIndex - 7;

}

else if (haveNumber == 0 && haveText == 0)

{

nameEndIndex = RequestHeaderCounter - 1;

}

ContactNameLength = nameEndIndex - nameIndex + 1;

haveName = 1;

break;

}

}

حال نام و شماره و تکست اگر وجود داشتند بخوان (به کمک حلقه)

if (haveName)

{

for (uint8\_t i = 0; i < ContactNameLength; i++)

{

ContactName[i] = RequestHeader[nameIndex + i];

}

sprintf(buffer, "NameIndex= %d, NameEndIndex= %d\r\n", nameIndex, nameEndIndex);

HAL\_UART\_Transmit(&huart6, buffer, 32, 1000);

if (nameEndIndex - nameIndex < 1)

{

sprintf(ErrorText, "wrong name, at least type 1 character!");

ErrorTextLength = 38;

HAL\_UART\_Transmit(&huart6, "wrong Name\r\n", 12, 1000);

return -1;

}

for (int i = nameIndex; i <= nameEndIndex; i++)

{

if (!((RequestHeader[i] >= 0x41 && RequestHeader[i] <= 0x5A) || (RequestHeader[i] >= 0x61 && RequestHeader[i] <= 0x7a)))

{

sprintf(ErrorText, "wrong name, A-Z, a-z are allowed");

ErrorTextLength = 32;

HAL\_UART\_Transmit(&huart6, "wrong Name\r\n", 12, 1000);

return -1;

}

}

sprintf(buffer, "Name: %s\r\n", ContactName);

HAL\_UART\_Transmit(&huart6, buffer, 8 + ContactNameLength, 1000);

sprintf(buffer, "NameLength: %d\r\n", ContactNameLength);

HAL\_UART\_Transmit(&huart6, buffer, 15, 1000);

}

if (haveNumber)

{

for (uint8\_t i = 0; i < ContactNumberLength; i++)

{

ContactNumber[i] = RequestHeader[numberIndex + i];

}

sprintf(buffer, "NumberIndex= %d, NumberEndIndex= %d\r\n", numberIndex, numberEndIndex);

HAL\_UART\_Transmit(&huart6, buffer, 37, 1000);

اکنون که همه چیز خوانده شد ارور ها را بررسی کن

ارور تعداد کاراکتر

نوع کاراکتر

طول پیام

if (numberEndIndex - numberIndex != 10)

{

sprintf(ErrorText, "wrong number, it must be 10 number!");

ErrorTextLength = 35;

HAL\_UART\_Transmit(&huart6, "wrong Number\r\n", 14, 1000);

return -1;

}

for (int i = numberIndex; i <= numberEndIndex; i++)

{

if (!(RequestHeader[i] >= 0x30 && RequestHeader[i] <= 0x39))

{

sprintf(ErrorText, "wrong number, numbers are just number!");

ErrorTextLength = 38;

HAL\_UART\_Transmit(&huart6, "wrong Number\r\n", 14, 1000);

return -1;

}

}

sprintf(buffer, "Number: %s\r\n", ContactNumber);

HAL\_UART\_Transmit(&huart6, buffer, 10 + (numberEndIndex - numberIndex + 1), 1000);

}

if (haveText)

{

for (uint8\_t i = 0; i < TextLength; i++)

{

TextMessage[i] = RequestHeader[textIndex + i];

}

sprintf(buffer, "TextIndex= %d, TextEndIndex= %d\r\n", textIndex, textEndIndex);

HAL\_UART\_Transmit(&huart6, buffer, 33, 1000);

if (textEndIndex - textIndex < 1)

{

sprintf(ErrorText, "wrong text, at least type 1 character!");

ErrorTextLength = 38;

HAL\_UART\_Transmit(&huart6, "No Input Text\r\n", 15, 1000);

return -1;

}

sprintf(buffer, "Text: %s\r\n", TextMessage);

HAL\_UART\_Transmit(&huart6, buffer, 8 + TextLength, 1000);

sprintf(buffer, "TextLength: %d\r\n", TextLength);

HAL\_UART\_Transmit(&huart6, buffer, 15, 1000);

}

return 0;

}

متدهای ارسال صفحه توسط ماژول وای فای

کاملا ساده و بدون جزییات

void ESP\_SendIndex()

{

HAL\_UART\_Transmit(&huart6, "Sending Index page ....\r\n", 25, 1000);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, IndexPageCharCount);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, IndexPage, IndexPageCharCount, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPCLOSE=%d\r\n", clientNumber);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPCLOSE=%d\r\n"), 1000);

HAL\_Delay(300);

}

void ESP\_SendSMS()

{

HAL\_UART\_Transmit(&huart6, "Sending SMS Page ....\r\n", 23, 1000);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, SMSPageCharCount);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, SMSPage, SMSPageCharCount, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPCLOSE=%d\r\n", clientNumber);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPCLOSE=%d\r\n"), 1000);

HAL\_Delay(300);

}

void ESP\_SendSuccess()

{

HAL\_UART\_Transmit(&huart6, "Sending Success Page ....\r\n", 27, 1000);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, SuccessPageCharCount);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, SuccessPage, SuccessPageCharCount, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPCLOSE=%d\r\n", clientNumber);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPCLOSE=%d\r\n"), 1000);

HAL\_Delay(300);

}

void ESP\_SendCall()

{

HAL\_UART\_Transmit(&huart6, "Sending Call Page ....\r\n", 18, 1000);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, CallPageCharCount);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, CallPage, CallPageCharCount, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPCLOSE=%d\r\n", clientNumber);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPCLOSE=%d\r\n"), 1000);

HAL\_Delay(300);

}

void ESP\_SendTalking()

{

HAL\_UART\_Transmit(&huart6, "Sending Talking Page ....\r\n", 27, 1000);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, TalkingPageCharCount);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, TalkingPage, TalkingPageCharCount, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPCLOSE=%d\r\n", clientNumber);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPCLOSE=%d\r\n"), 1000);

HAL\_Delay(300);

}

void ESP\_SendError()

{

HAL\_UART\_Transmit(&huart6, "Sending Error page ....\r\n", 25, 1000);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, ErrorPageCharCount1);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(500);

HAL\_UART\_Transmit(&huart3, ErrorPage1, ErrorPageCharCount1, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, ErrorTextLength);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(500);

HAL\_UART\_Transmit(&huart3, ErrorText, ErrorTextLength, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, ErrorPageCharCount2);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(500);

HAL\_UART\_Transmit(&huart3, ErrorPage2, ErrorPageCharCount2, 1000);

HAL\_Delay(200);

sprintf(buffer, "AT+CIPCLOSE=%d\r\n", clientNumber);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPCLOSE=%d\r\n"), 1000);

HAL\_Delay(300);

}

پیچیده ترین قسمت ارسال صفحه مخاطبین به علت جدول موجود در وسط صفحه میباشد

void ESP\_SendContact()

{

HAL\_UART\_Transmit(&huart6, "Sending Contact Page ....\r\n", 27, 1000);

قسمت بالای صفحه را بفرست

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, ContactPageCharCount1);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, ContactPage1, ContactPageCharCount1, 1000);

HAL\_Delay(200);

unsigned char tempName[ContactNameMaxLength];

uint16\_t tempNameLength = 0;

unsigned char tempNumber[ContactNumberLength];

uint16\_t temp;

برای هر مخاطب شماره و نام را پیدا کن

// For each contact

for (int i = 0; i < ContactCount; i++)

{

// Read Number

ReadAddr = VirtAddVarTab[ContactNumberROMAddrIndex] + (i \* ContactNumberLength);

for (int j = 0; j < ContactNumberLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

EE\_ReadVariable(ReadAddr, &temp);

HAL\_FLASH\_Lock();

tempNumber[j] = temp;

}

// Read Name

memset(tempName, 0, sizeof(tempName));

tempNameLength = 0;

ReadAddr = VirtAddVarTab[ContactNameROMAddrIndex] + (i \* ContactNameMaxLength);

for (int j = 0; j < ContactNameMaxLength; j++, ReadAddr++)

{

HAL\_FLASH\_Unlock();

if (EE\_ReadVariable(ReadAddr, &temp) != HAL\_OK)

{

break; // Its not because an error, the name ENDS if it cant read.

}

HAL\_FLASH\_Lock();

if ((temp >= 0x41 && temp <= 0x5A) || (temp >= 0x61 && temp <= 0x7a))

{

tempNameLength++;

tempName[j] = temp;

}

}

اکنون آن را ارسال کن

unsigned char buffer3[tempNameLength + ContactNumberLength + 27];

sprintf(buffer3, "<tr><td>%s</td><td>%s</td></tr>", tempName, tempNumber);

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, sizeof(buffer3));

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, buffer3, sizeof(buffer3), 1000);

HAL\_Delay(300);

}

ادامه صفحه را ارسال کن

if(ContactCount ==0)

{

unsigned char buffer3[29];

sprintf(buffer3, "<tr><td>-</td><td>-</td></tr>");

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, sizeof(buffer3));

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, buffer3, sizeof(buffer3), 1000);

HAL\_Delay(300);

}

sprintf(buffer, "AT+CIPSEND=%d,%d\r\n", clientNumber, ContactPageCharCount2);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPSEND=%d,%d\r\n"), 1000);

HAL\_Delay(1000);

HAL\_UART\_Transmit(&huart3, ContactPage2, ContactPageCharCount2, 1000);

HAL\_Delay(300);

sprintf(buffer, "AT+CIPCLOSE=%d\r\n", clientNumber);

HAL\_UART\_Transmit(&huart3, buffer, sizeof("AT+CIPCLOSE=%d\r\n"), 1000);

HAL\_Delay(300);

}

از آنجایی که تمام سعی خود را کردم که پروژه به گویا ترین شکل ممکن پیاده سازی شود ، امیدوارم برای همه کارآمد بوده باشه.

جهت مشارکت :

<https://github.com/Goldenstarc/FUM-MicroLAB>

تشکر