Use only the IDE 1.8.0 version (otherwise, there will be a lot of errors while debugging the existing code)

Every time when you debug the bootloader; the memory should be erased fully before debugging.

OTA Bootloader code implementing procedure

1. Link:https://drive.google.com/drive/u/0/folders/1DO_bEmZmQvEaX2nyyqz2ODRTIiKHgAtu

Bootloader-
https://drive.google.com/drive/folders/1FUvmFWN0EjUa27YLsqgKhsZBY00Wev99?usp=sharing

User code-
https://drive.google.com/file/d/1LJC57K4lmpbP_r4qsyN0LX3eWEjtTB54/view?usp=sharing

2. Get IMEI number through the live expression topic

3. Subscribe the topicD2S/SA/V1/your IMEI no and add 0 at the end/S hiveMQ broker (http://www.hivemq.com/demos/websocket-client/?)

- Connection: - HOST- mqtt.senzmate.com
- Subscription: - Add new topic D2S/SA/V1/your IMEI no and add 0 at the end/S
  QoS=0

4. Clear your board memory through Cube Programmer. (Not necessary)

5. Run the OTA_senzagro code with the flag=1. Track the variable 'downloadState' in the live expression. And wait until the bin file is downloaded from the server. (The download function will loop again and again unless the flag is set to 0. Therefore, stop the debugging while the 'downloadState' reach value 1 '\1' or put a breaking point before)

6. Set flag=0 and debug the code again. (To run the user code directly)

7. The readings will be displayed in the hiveMQ broker website. (Above mentioned link). The message will be received once every 10 mins.

CREATING BIN FILE

- Launch the code at starting address at 0x08000000
- Then delete the bin file created and change the following settings and just build the project (don't debug or run)
  - Setting flash starting address
    STM32L071CZTX_FLASH.ld—> MEMORY—> FLASH (rx) : ORIGIN-0x08008000
  - Setting vector table offset in .c file
    Core—> Src—>system_stm32l0xx.c—>
    - Uncomment '/* **#define USER_VECT_TAB_ADDRESS** */'
    - Set the vector tale offset as bellow
      #define VECT_TAB_OFFSET   0x00008000U

FLASH IMPLEMENTATION IN OTA BOOTLOADER (28/6/2022)

AIM OF THE CODE:

1. Read the memory address- 0x0802FF70 (flag)
2. If flag=0; download the code.
3. After downloading; write the flag (0x0802FF70) to 1.
4. Again read the memory.
5. If flag=1; run the user code .

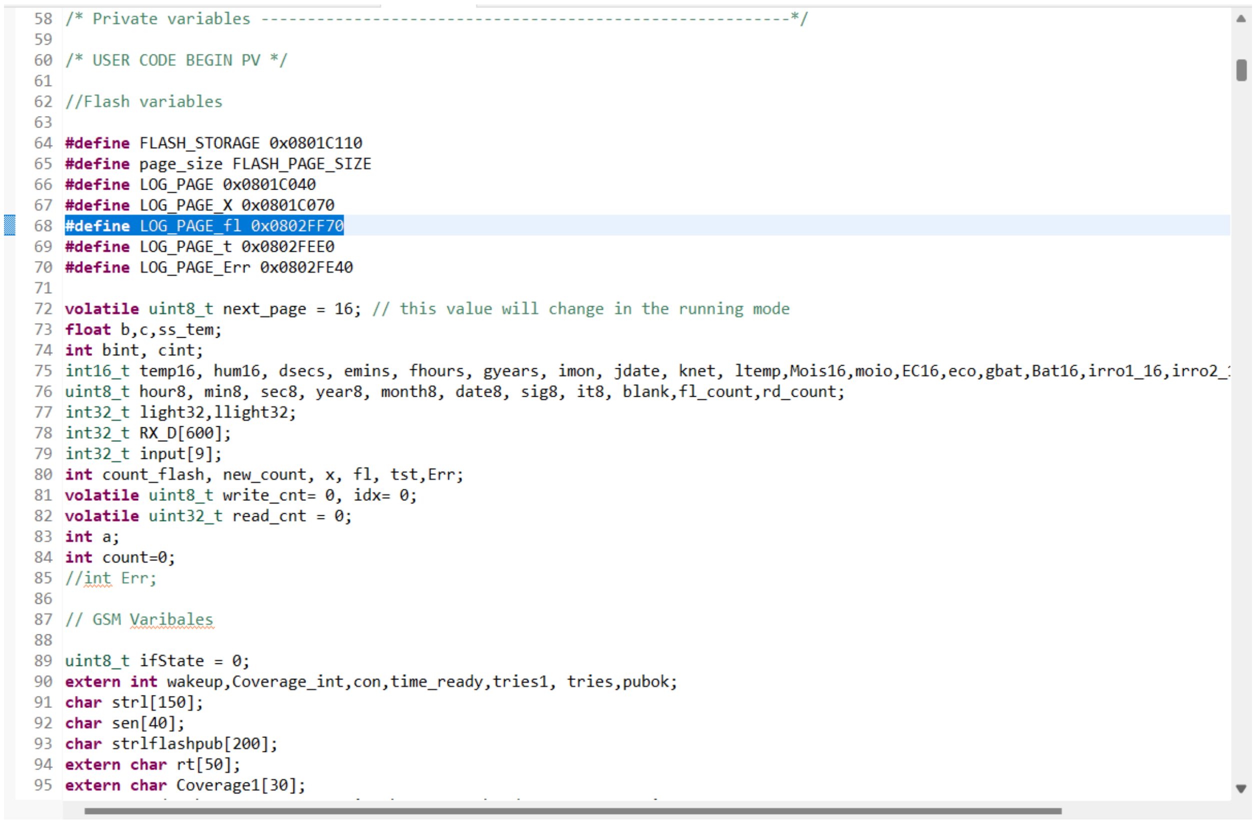
ERRORS:

- Faced an error while implementing flash to the OTA bootloader using OTA_FLASH_Errored. **When reading and writing the flag using the flash function, 'AT' command couldn't be initialized.** HAL_UART_Receive(&huart1, gsmreply, size, RX_TIMEOUT1); couldn't reply 'OK'. Therefore it tried to connect 4 times and failed to connect(worked as we defined in our code). It resulted in resetting as we predefined. When I tried this flash implementation to the user code and tested it alone without an OTA bootloader, It resulted in the same thing. It struggled to publish. When I tested the OTA bootloader part by part, It was found that the flash implementation is the only reason for the struggle. When checking both points in flash and GSM initialization, there are no similarities except **HAL library and timer.** But the exact reason couldn't be found.
  (In addition to this process, while trying to solve the error, when UART_Deinit(); added, in the power_toggle(), flash memory overflowed.)

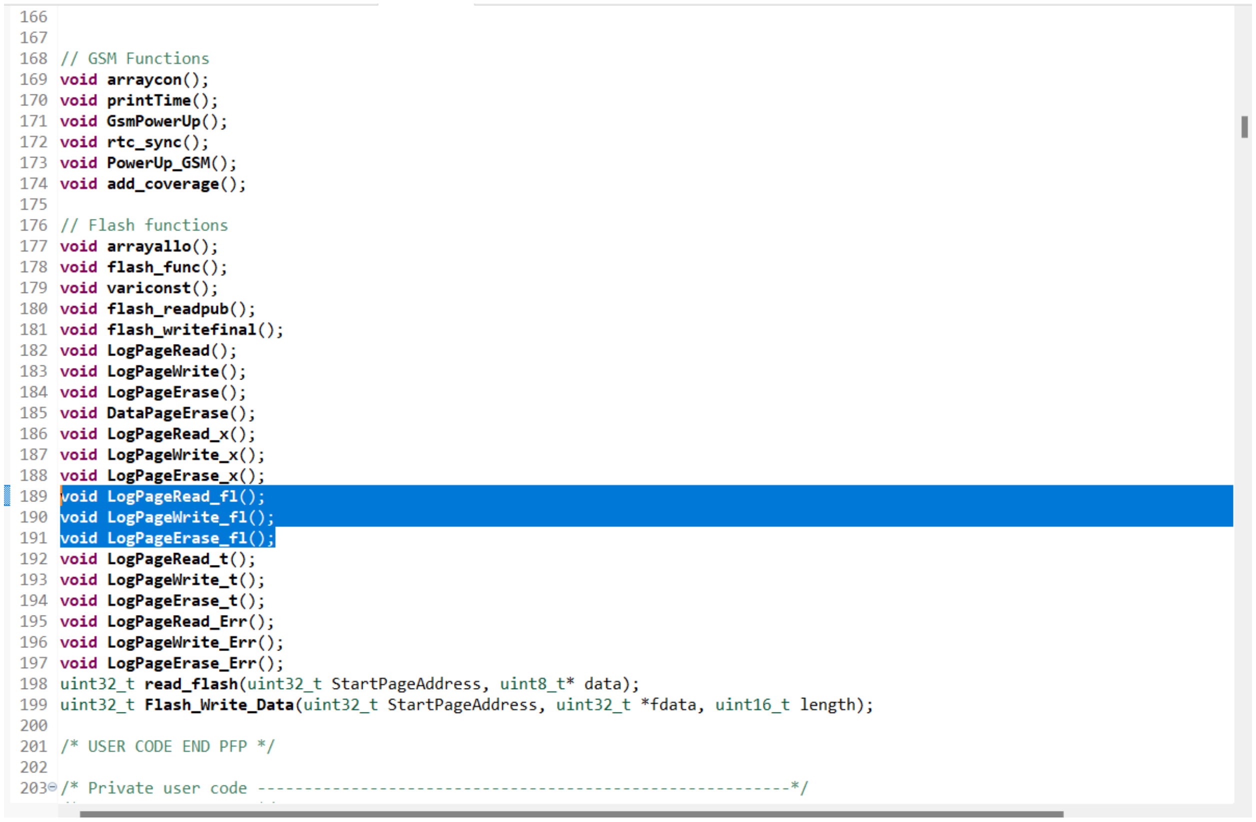
- Randomly checked with another flash write and read functions. It worked perfectly.
  OTA_FLASH. Therefore, the issue is solved.

Running code with flash written flag control

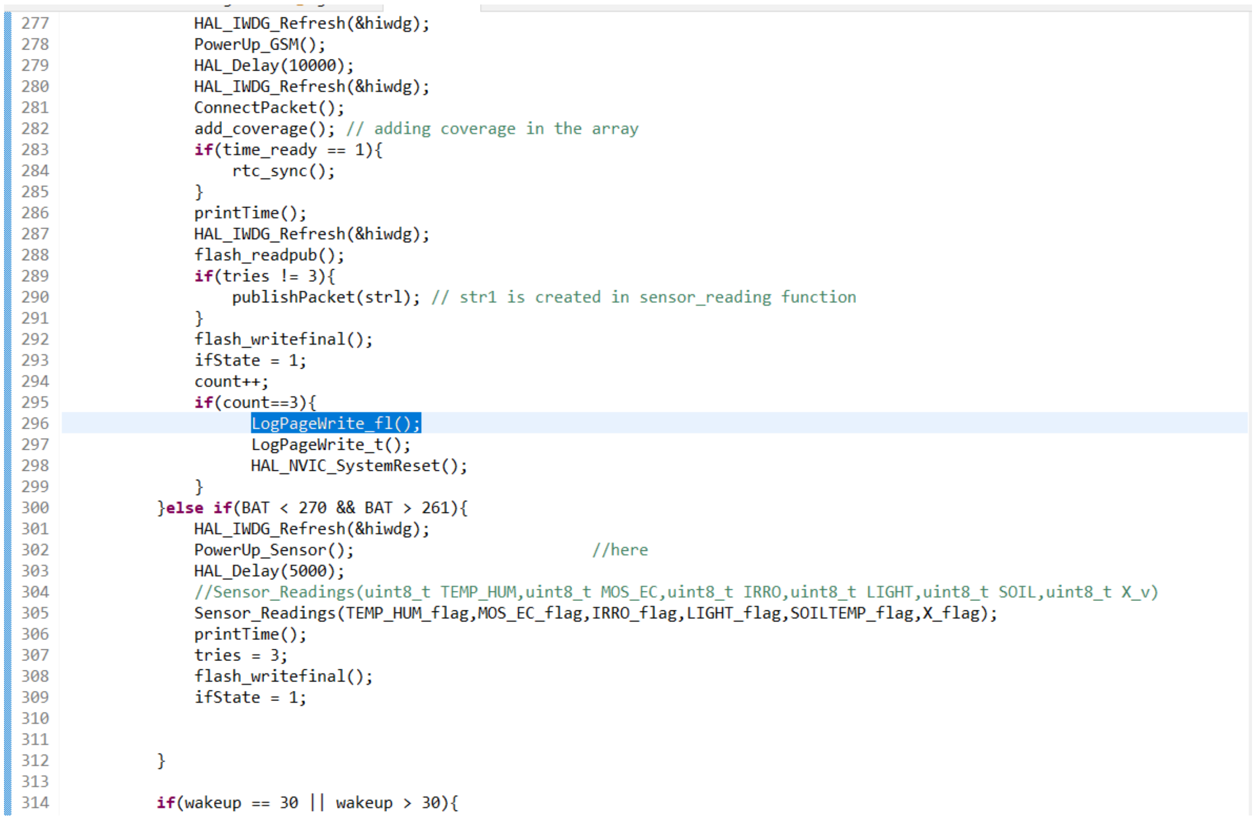
Have to implement flash writing functions in your user code.
Commit - `dd0a953`



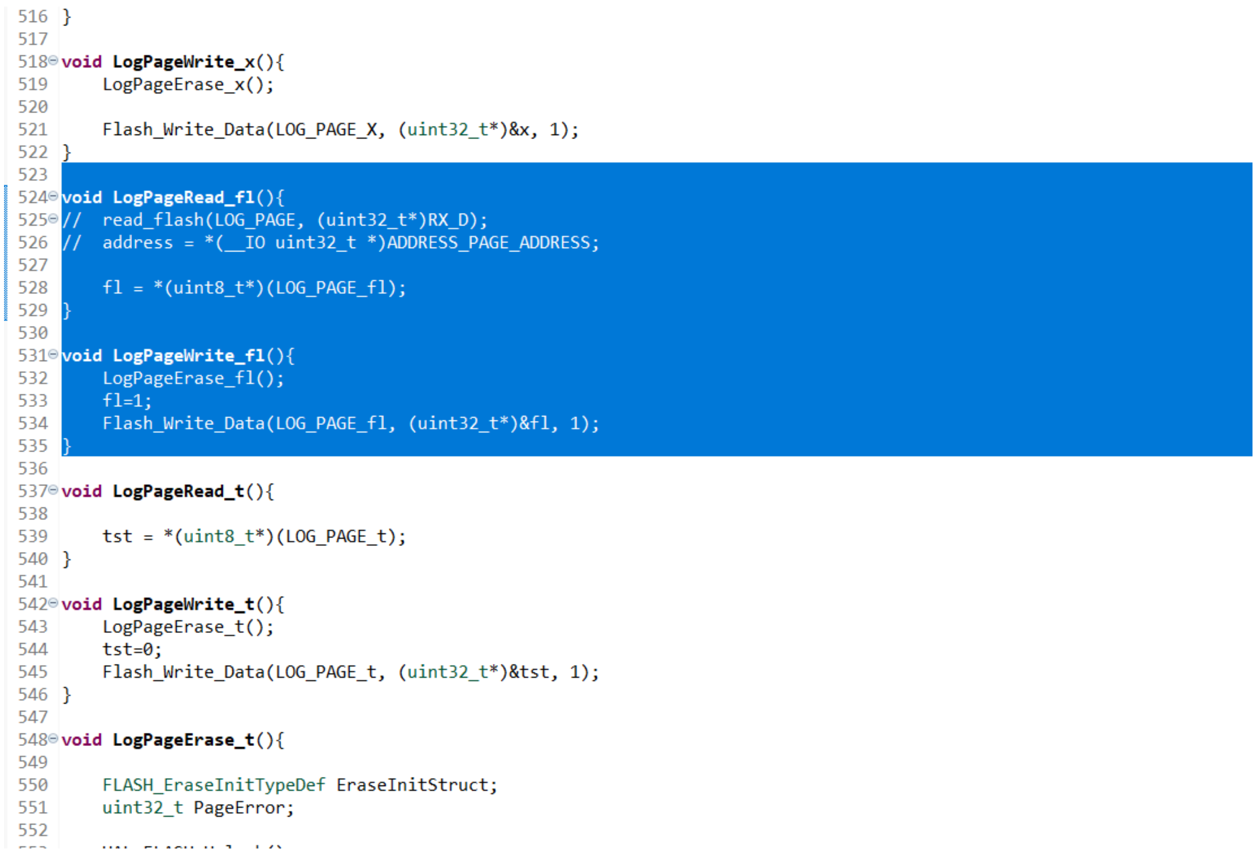
```c
/* Private variables ---------------------------------------------------------*/

/* USER CODE BEGIN PV */

//Flash variables

#define FLASH_STORAGE 0x0801C110
#define page_size FLASH_PAGE_SIZE
#define LOG_PAGE 0x0801C040
#define LOG_PAGE_X 0x0801C070
#define LOG_PAGE_fl 0x0802FF70
#define LOG_PAGE_t 0x0802FEE0
#define LOG_PAGE_Err 0x0802FE40

volatile uint8_t next_page = 16; // this value will change in the running mode
float b,c,ss_tem;
int bint, cint;
int16_t temp16, hum16, dsecs, emins, fhours, gyears, imon, jdate, knet, ltemp,Mois16,moio,EC16,eco,gbat,Bat16,irro1_16,irro2_
uint8_t hour8, min8, sec8, year8, month8, date8, sig8, it8, blank,fl_count,rd_count;
int32_t light32,llight32;
int32_t RX_D[600];
int32_t input[9];
int count_flash, new_count, x, fl, tst,Err;
volatile uint8_t write_cnt= 0, idx= 0;
volatile uint32_t read_cnt = 0;
int a;
int count=0;
//int Err;

// GSM Varibales

uint8_t ifState = 0;
extern int wakeup,Coverage_int,con,time_ready,tries1, tries,pubok;
char strl[150];
char sen[40];
char strlflashpub[200];
extern char rt[50];
extern char Coverage1[30];
```



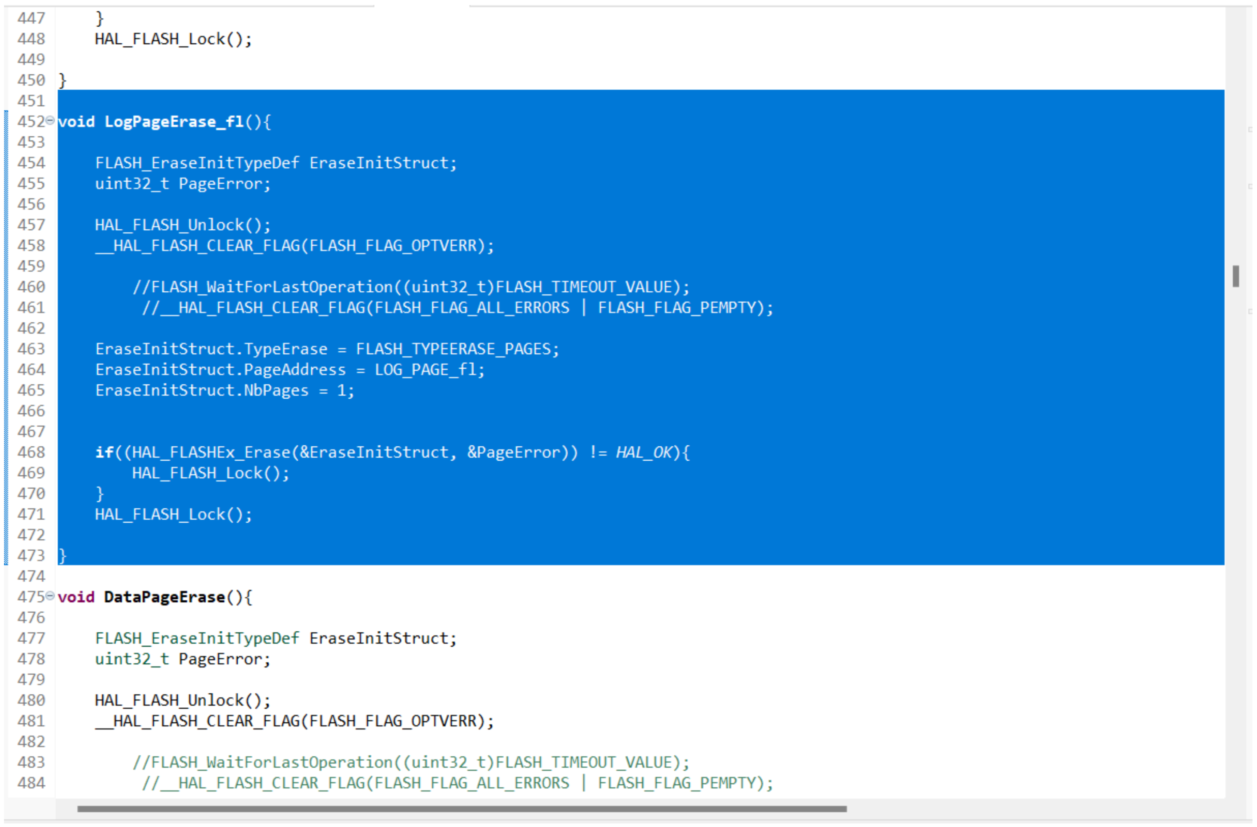
```c


// GSM Functions
void arraycon();
void printTime();
void GsmPowerUp();
void rtc_sync();
void PowerUp_GSM();
void add_coverage();

// Flash functions
void arrayallo();
void flash_func();
void variconst();
void flash_readpub();
void flash_writefinal();
void LogPageRead();
void LogPageWrite();
void LogPageErase();
void DataPageErase();
void LogPageRead_x();
void LogPageWrite_x();
void LogPageErase_x();
void LogPageRead_fl();
void LogPageWrite_fl();
void LogPageErase_fl();
void LogPageRead_t();
void LogPageWrite_t();
void LogPageErase_t();
void LogPageRead_Err();
void LogPageWrite_Err();
void LogPageErase_Err();
uint32_t read_flash(uint32_t StartPageAddress, uint8_t* data);
uint32_t Flash_Write_Data(uint32_t StartPageAddress, uint32_t *fdata, uint16_t length);

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
```



```c
              HAL_IWDG_Refresh(&hiwdg);
              PowerUp_GSM();
              HAL_Delay(10000);
              HAL_IWDG_Refresh(&hiwdg);
              ConnectPacket();
              add_coverage(); // adding coverage in the array
              if(time_ready == 1){
                  rtc_sync();
              }
              printTime();
              HAL_IWDG_Refresh(&hiwdg);
              flash_readpub();
              if(tries != 3){
                  publishPacket(strl); // str1 is created in sensor_reading function
              }
              flash_writefinal();
              ifState = 1;
              count++;
              if(count==3){
                    LogPageWrite_fl();
                    LogPageWrite_t();
                    HAL_NVIC_SystemReset();
              }
          }else if(BAT < 270 && BAT > 261){
              HAL_IWDG_Refresh(&hiwdg);
              PowerUp_Sensor();                         //here
              HAL_Delay(5000);
              //Sensor_Readings(uint8_t TEMP_HUM,uint8_t MOS_EC,uint8_t IRRO,uint8_t LIGHT,uint8_t SOIL,uint8_t X_v)
              Sensor_Readings(TEMP_HUM_flag,MOS_EC_flag,IRRO_flag,LIGHT_flag,SOILTEMP_flag,X_flag);
              printTime();
              tries = 3;
              flash_writefinal();
              ifState = 1;


          }

          if(wakeup == 30 || wakeup > 30){
```

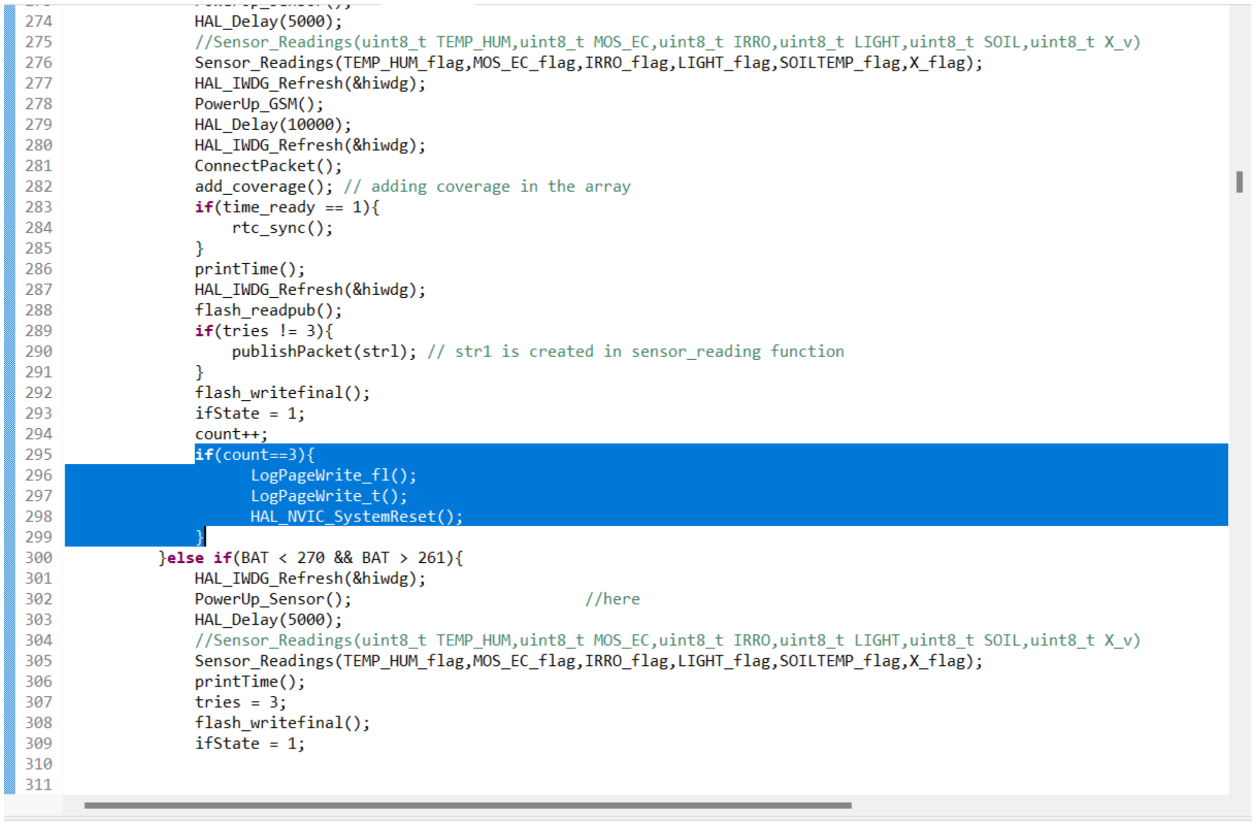


```c
}

void LogPageWrite_x(){
    LogPageErase_x();

    Flash_Write_Data(LOG_PAGE_X, (uint32_t*)&x, 1);
}

void LogPageRead_fl(){
//  read_flash(LOG_PAGE, (uint32_t*)RX_D);
//  address = *(__IO uint32_t *)ADDRESS_PAGE_ADDRESS;

    fl = *(uint8_t*)(LOG_PAGE_fl);
}

void LogPageWrite_fl(){
    LogPageErase_fl();
    fl=1;
    Flash_Write_Data(LOG_PAGE_fl, (uint32_t*)&fl, 1);
}

void LogPageRead_t(){

    tst = *(uint8_t*)(LOG_PAGE_t);
}

void LogPageWrite_t(){
    LogPageErase_t();
    tst=0;
    Flash_Write_Data(LOG_PAGE_t, (uint32_t*)&tst, 1);
}

void LogPageErase_t(){

    FLASH_EraseInitTypeDef EraseInitStruct;
    uint32_t PageError;

```



```c
    }
    HAL_FLASH_Lock();

}

void LogPageErase_fl(){

    FLASH_EraseInitTypeDef EraseInitStruct;
    uint32_t PageError;

    HAL_FLASH_Unlock();
    __HAL_FLASH_CLEAR_FLAG(FLASH_FLAG_OPTVERR);

        //FLASH_WaitForLastOperation((uint32_t)FLASH_TIMEOUT_VALUE);
         //__HAL_FLASH_CLEAR_FLAG(FLASH_FLAG_ALL_ERRORS | FLASH_FLAG_PEMPTY);

    EraseInitStruct.TypeErase = FLASH_TYPEERASE_PAGES;
    EraseInitStruct.PageAddress = LOG_PAGE_fl;
    EraseInitStruct.NbPages = 1;


    if((HAL_FLASHEx_Erase(&EraseInitStruct, &PageError)) != HAL_OK){
        HAL_FLASH_Lock();
    }
    HAL_FLASH_Lock();

}

void DataPageErase(){

    FLASH_EraseInitTypeDef EraseInitStruct;
    uint32_t PageError;

    HAL_FLASH_Unlock();
    __HAL_FLASH_CLEAR_FLAG(FLASH_FLAG_OPTVERR);

        //FLASH_WaitForLastOperation((uint32_t)FLASH_TIMEOUT_VALUE);
         //__HAL_FLASH_CLEAR_FLAG(FLASH_FLAG_ALL_ERRORS | FLASH_FLAG_PEMPTY);
```

IF YOU WANT TO LOOP THE CODE FOR DOWNLOADING AND USER CODE RUNNING (DOWNLOAD AFTER 3 PUBLISHES….)



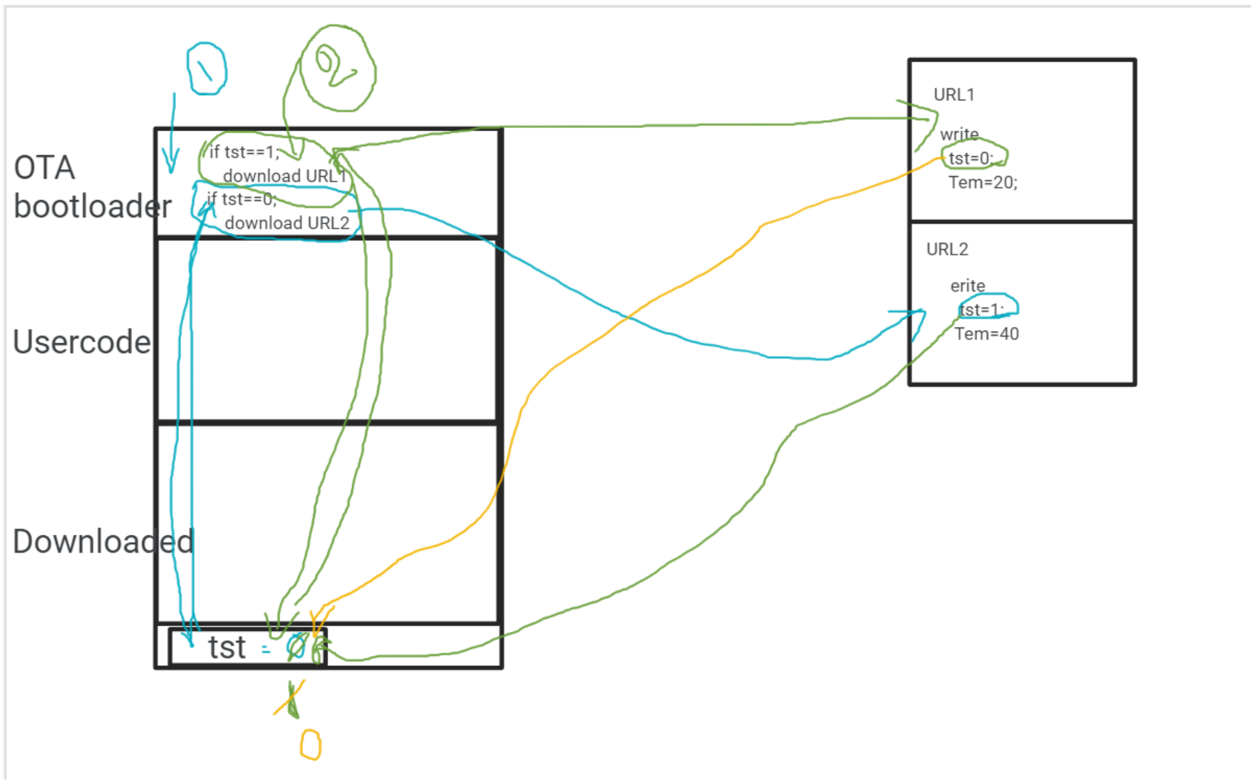
```
            HAL_Delay(5000);
            //Sensor_Readings(uint8_t TEMP_HUM,uint8_t MOS_EC,uint8_t IRRO,uint8_t LIGHT,uint8_t SOIL,uint8_t X_v)
            Sensor_Readings(TEMP_HUM_flag,MOS_EC_flag,IRRO_flag,LIGHT_flag,SOILTEMP_flag,X_flag);
            HAL_IWDG_Refresh(&hiwdg);
            PowerUp_GSM();
            HAL_Delay(10000);
            HAL_IWDG_Refresh(&hiwdg);
            ConnectPacket();
            add_coverage(); // adding coverage in the array
            if(time_ready == 1){
                rtc_sync();
            }
            printTime();
            HAL_IWDG_Refresh(&hiwdg);
            flash_readpub();
            if(tries != 3){
                publishPacket(strl); // str1 is created in sensor_reading function
            }
            flash_writefinal();
            ifState = 1;
            count++;
            if(count==3){
                  LogPageWrite_fl();
                  LogPageWrite_t();
                  HAL_NVIC_SystemReset();
            }
        }else if(BAT < 270 && BAT > 261){
            HAL_IWDG_Refresh(&hiwdg);
            PowerUp_Sensor();                          //here
            HAL_Delay(5000);
            //Sensor_Readings(uint8_t TEMP_HUM,uint8_t MOS_EC,uint8_t IRRO,uint8_t LIGHT,uint8_t SOIL,uint8_t X_v)
            Sensor_Readings(TEMP_HUM_flag,MOS_EC_flag,IRRO_flag,LIGHT_flag,SOILTEMP_flag,X_flag);
            printTime();
            tries = 3;
            flash_writefinal();
            ifState = 1;
```

For looped testing with 2 bin files

For testing a flag written variable is added in bootloader code to loop the 2 code downloading alternatively. So that variable writing should be added to the user code also.

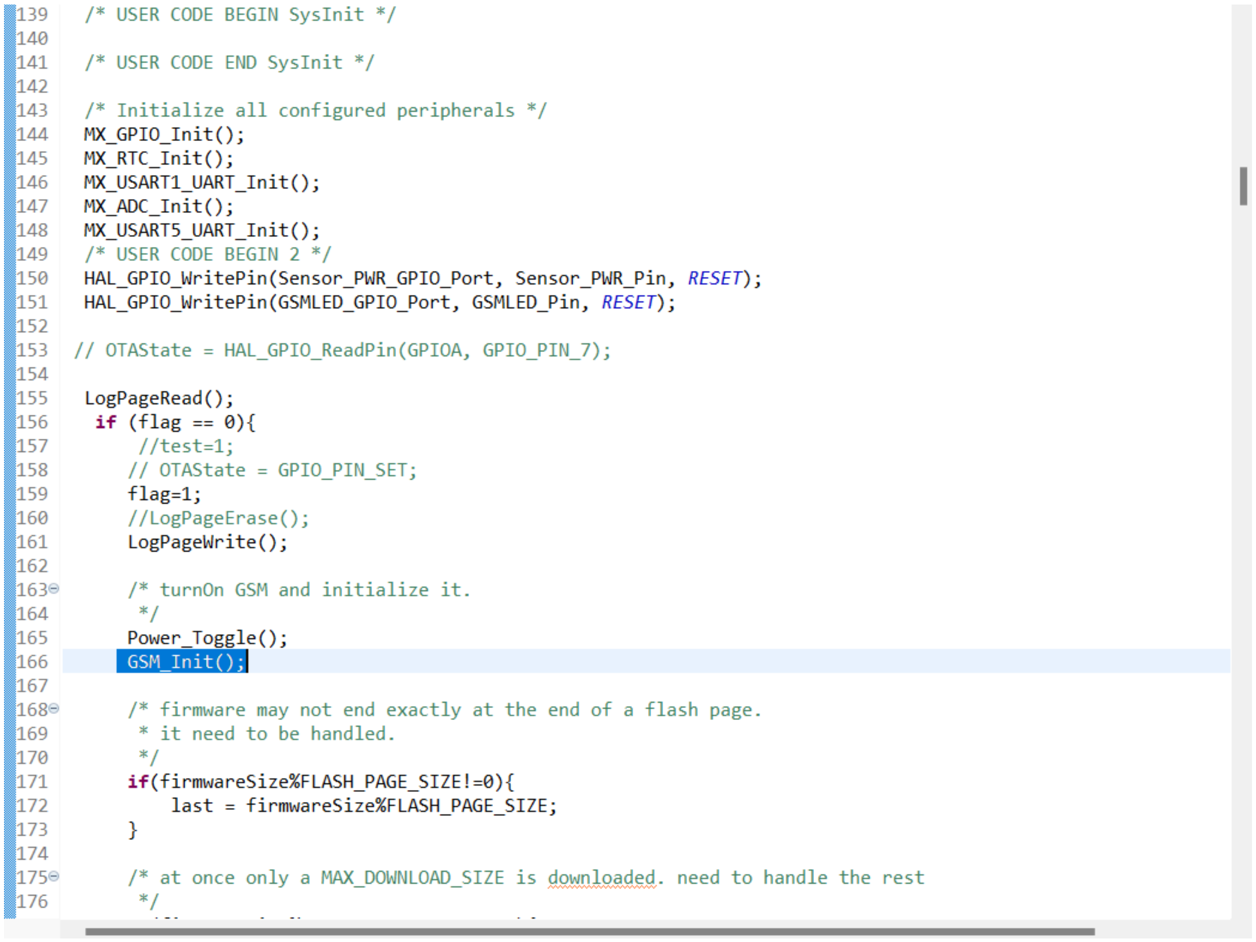
Bootloader commit- `440f6ac`

Usercode commit- `40776b9`



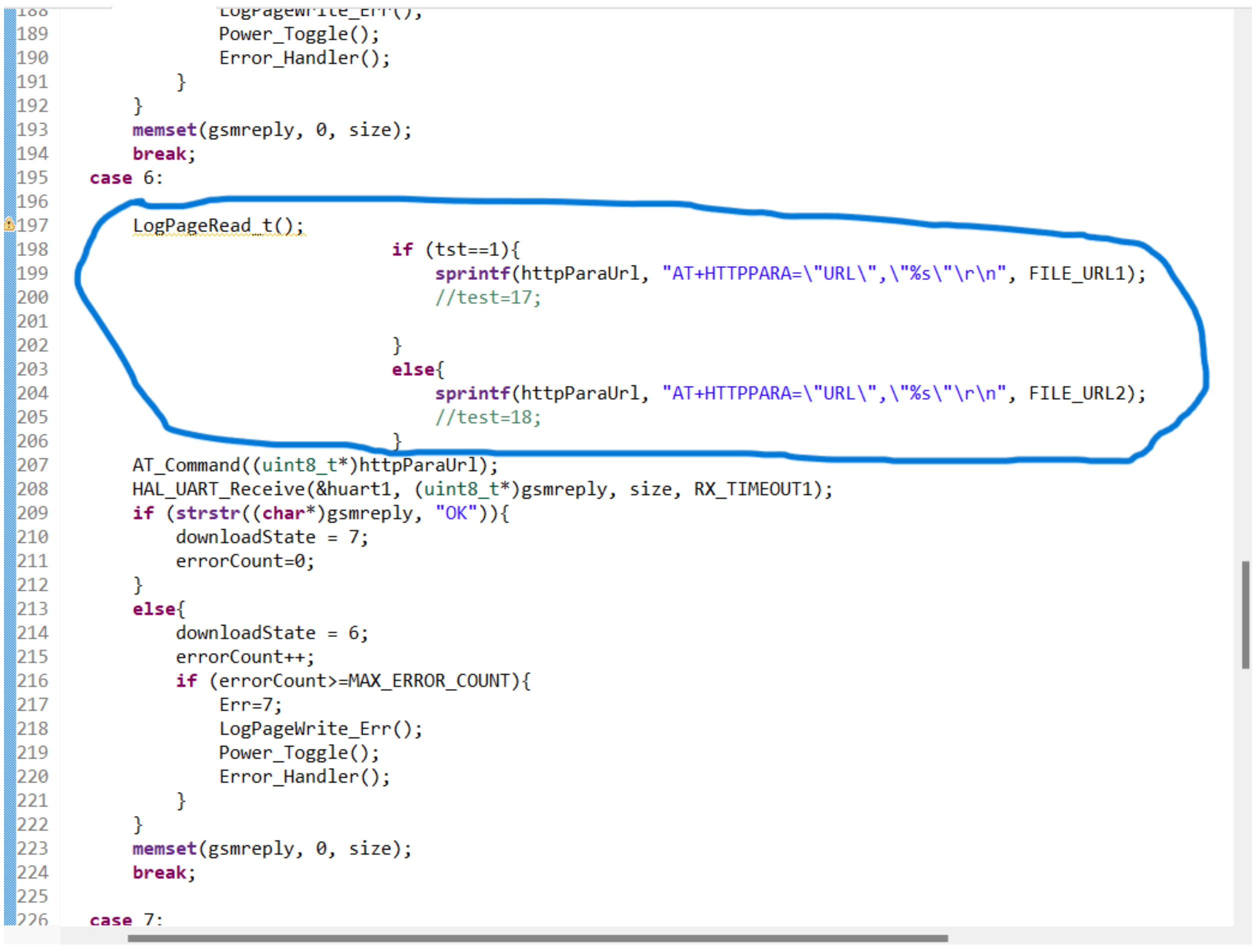
OTA Bootloader

- Go into the GSM_Init().



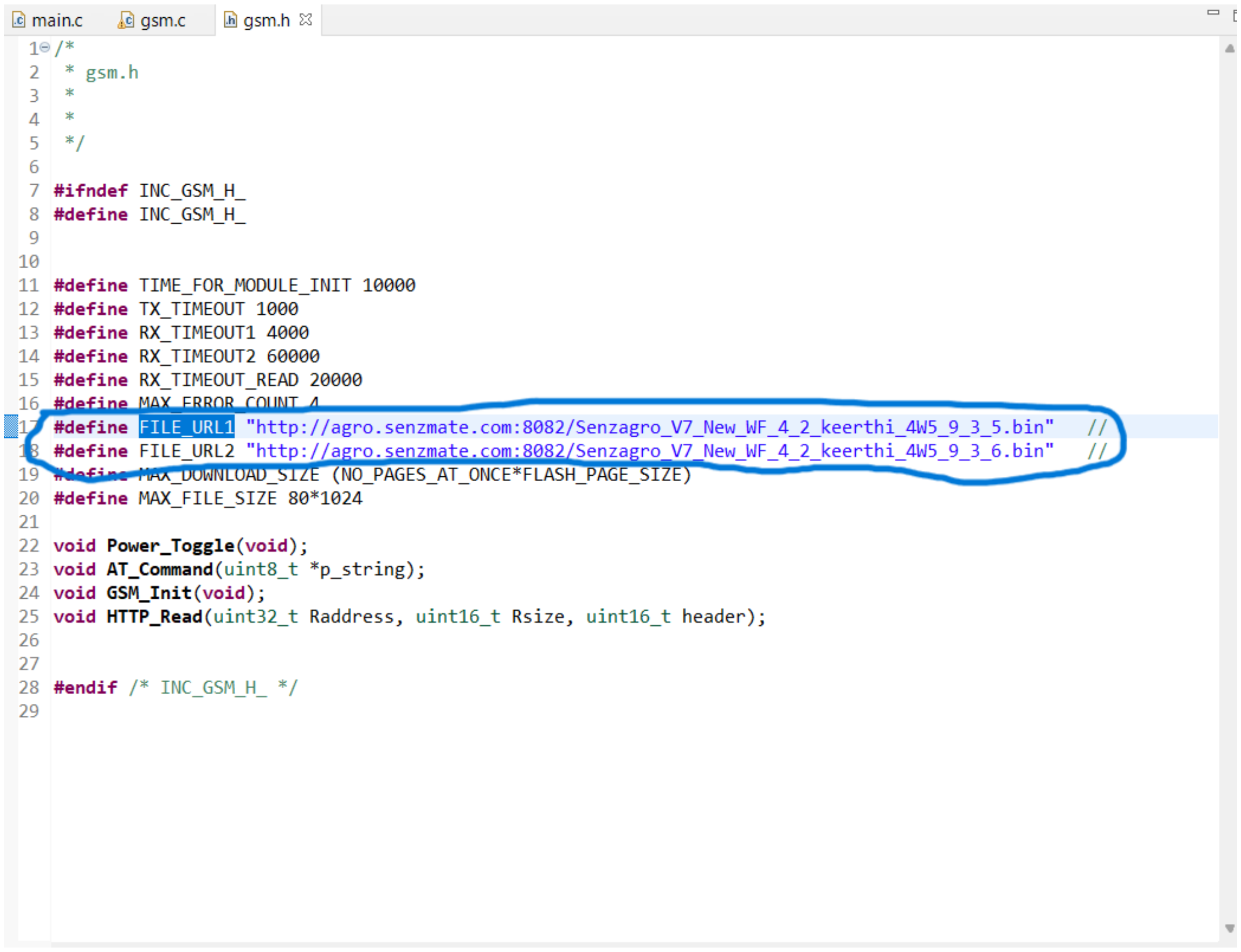
```
  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_RTC_Init();
  MX_USART1_UART_Init();
  MX_ADC_Init();
  MX_USART5_UART_Init();
  /* USER CODE BEGIN 2 */
  HAL_GPIO_WritePin(Sensor_PWR_GPIO_Port, Sensor_PWR_Pin, RESET);
  HAL_GPIO_WritePin(GSMLED_GPIO_Port, GSMLED_Pin, RESET);

 // OTAState = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_7);

  LogPageRead();
   if (flag == 0){
       //test=1;
      // OTAState = GPIO_PIN_SET;
      flag=1;
      //LogPageErase();
      LogPageWrite();

      /* turnOn GSM and initialize it.
       */
      Power_Toggle();
      GSM_Init();

      /* firmware may not end exactly at the end of a flash page.
       * it need to be handled.
       */
      if(firmwareSize%FLASH_PAGE_SIZE!=0){
          last = firmwareSize%FLASH_PAGE_SIZE;
      }

      /* at once only a MAX_DOWNLOAD_SIZE is downloaded. need to handle the rest
       */
```

- Every alternate iteration, the URL has to be altered.
- For that, The loop is implemented with ‘tst’ variable.



```
                LogPageWrite_Err();
                Power_Toggle();
                Error_Handler();
            }
        }
        memset(gsmreply, 0, size);
        break;
    case 6:

        LogPageRead_t();
                                if (tst==1){
                                    sprintf(httpParaUrl, "AT+HTTPPARA=\"URL\",\"%s\"\r\n", FILE_URL1);
                                    //test=17;

                                }
                                else{
                                    sprintf(httpParaUrl, "AT+HTTPPARA=\"URL\",\"%s\"\r\n", FILE_URL2);
                                    //test=18;
                                }
        AT_Command((uint8_t*)httpParaUrl);
        HAL_UART_Receive(&huart1, (uint8_t*)gsmreply, size, RX_TIMEOUT1);
        if (strstr((char*)gsmreply, "OK")){
            downloadState = 7;
            errorCount=0;
        }
        else{
            downloadState = 6;
            errorCount++;
            if (errorCount>=MAX_ERROR_COUNT){
                Err=7;
                LogPageWrite_Err();
                Power_Toggle();
                Error_Handler();
            }
        }
        memset(gsmreply, 0, size);
        break;

    case 7:
```

- URLs are defined in the gsm.h file

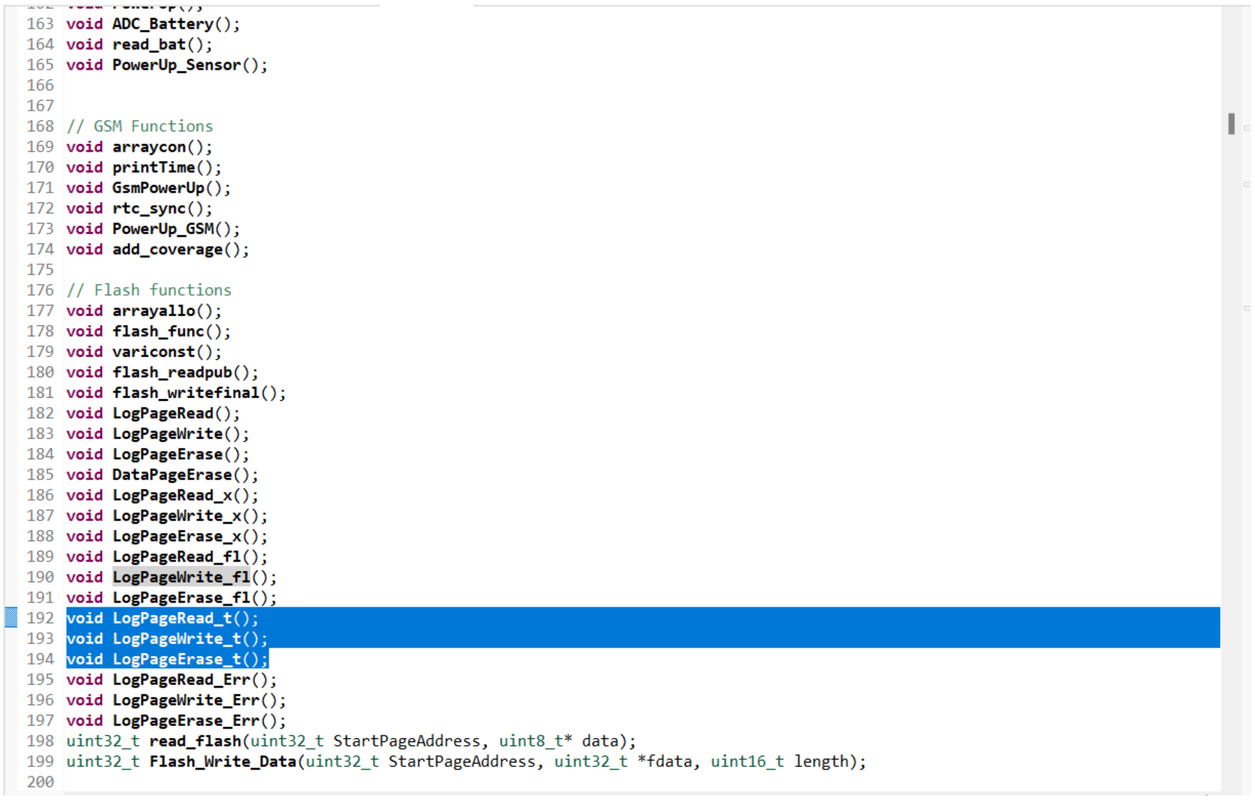


```c
/*
 * gsm.h
 *
 *
 */

#ifndef INC_GSM_H_
#define INC_GSM_H_


#define TIME_FOR_MODULE_INIT 10000
#define TX_TIMEOUT 1000
#define RX_TIMEOUT1 4000
#define RX_TIMEOUT2 60000
#define RX_TIMEOUT_READ 20000
#define MAX_ERROR_COUNT 4
#define FILE_URL1 "http://agro.senzmate.com:8082/Senzagro_V7_New_WF_4_2_keerthi_4W5_9_3_5.bin"   //
#define FILE_URL2 "http://agro.senzmate.com:8082/Senzagro_V7_New_WF_4_2_keerthi_4W5_9_3_6.bin"   //
#define MAX_DOWNLOAD_SIZE (NO_PAGES_AT_ONCE*FLASH_PAGE_SIZE)
#define MAX_FILE_SIZE 80*1024

void Power_Toggle(void);
void AT_Command(uint8_t *p_string);
void GSM_Init(void);
void HTTP_Read(uint32_t Raddress, uint16_t Rsize, uint16_t header);


#endif /* INC_GSM_H_ */
```

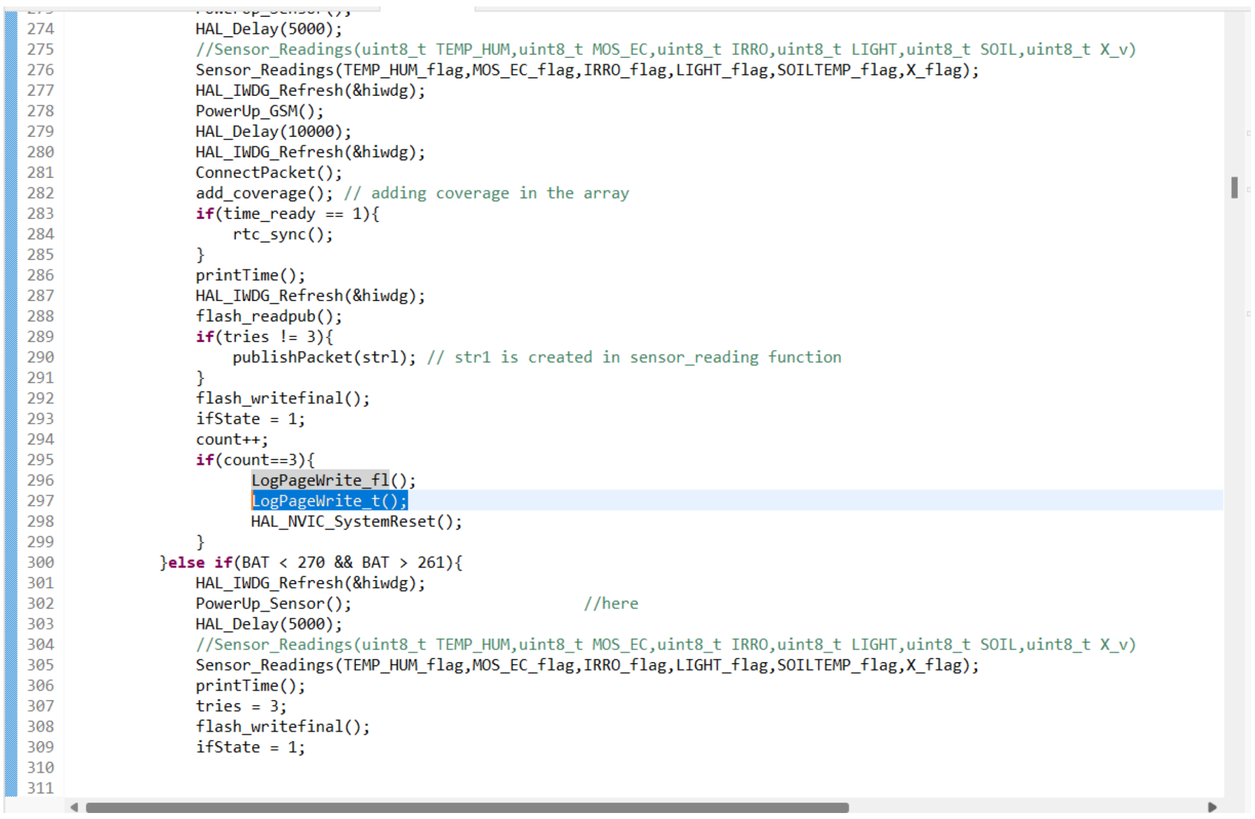
- The test variable should be written tst=1 in one bin file and tst=0 in another file (INSIDE THE logPageWrite_t() FUNCTION).

- File with tst=0; uploaded under the If(tst==1) condition in bootloader.
- File with tst=1; uploaded under the If(tst==0) condition in bootloader.

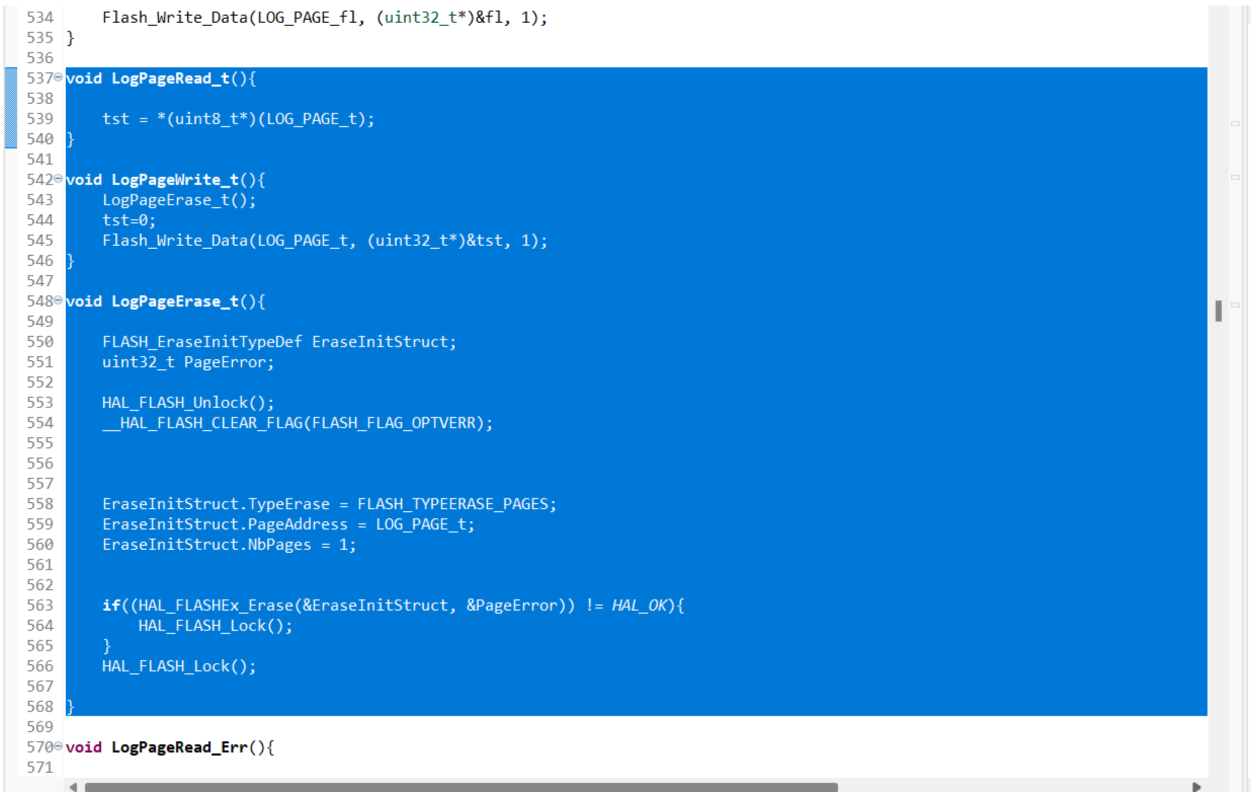
User code for FILE_URL1



```
void ADC_Battery();
void read_bat();
void PowerUp_Sensor();


// GSM Functions
void arraycon();
void printTime();
void GsmPowerUp();
void rtc_sync();
void PowerUp_GSM();
void add_coverage();

// Flash functions
void arrayallo();
void flash_func();
void variconst();
void flash_readpub();
void flash_writefinal();
void LogPageRead();
void LogPageWrite();
void LogPageErase();
void DataPageErase();
void LogPageRead_x();
void LogPageWrite_x();
void LogPageErase_x();
void LogPageRead_fl();
void LogPageWrite_fl();
void LogPageErase_fl();
void LogPageRead_t();
void LogPageWrite_t();
void LogPageErase_t();
void LogPageRead_Err();
void LogPageWrite_Err();
void LogPageErase_Err();
uint32_t read_flash(uint32_t StartPageAddress, uint8_t* data);
uint32_t Flash_Write_Data(uint32_t StartPageAddress, uint32_t *fdata, uint16_t length);

```



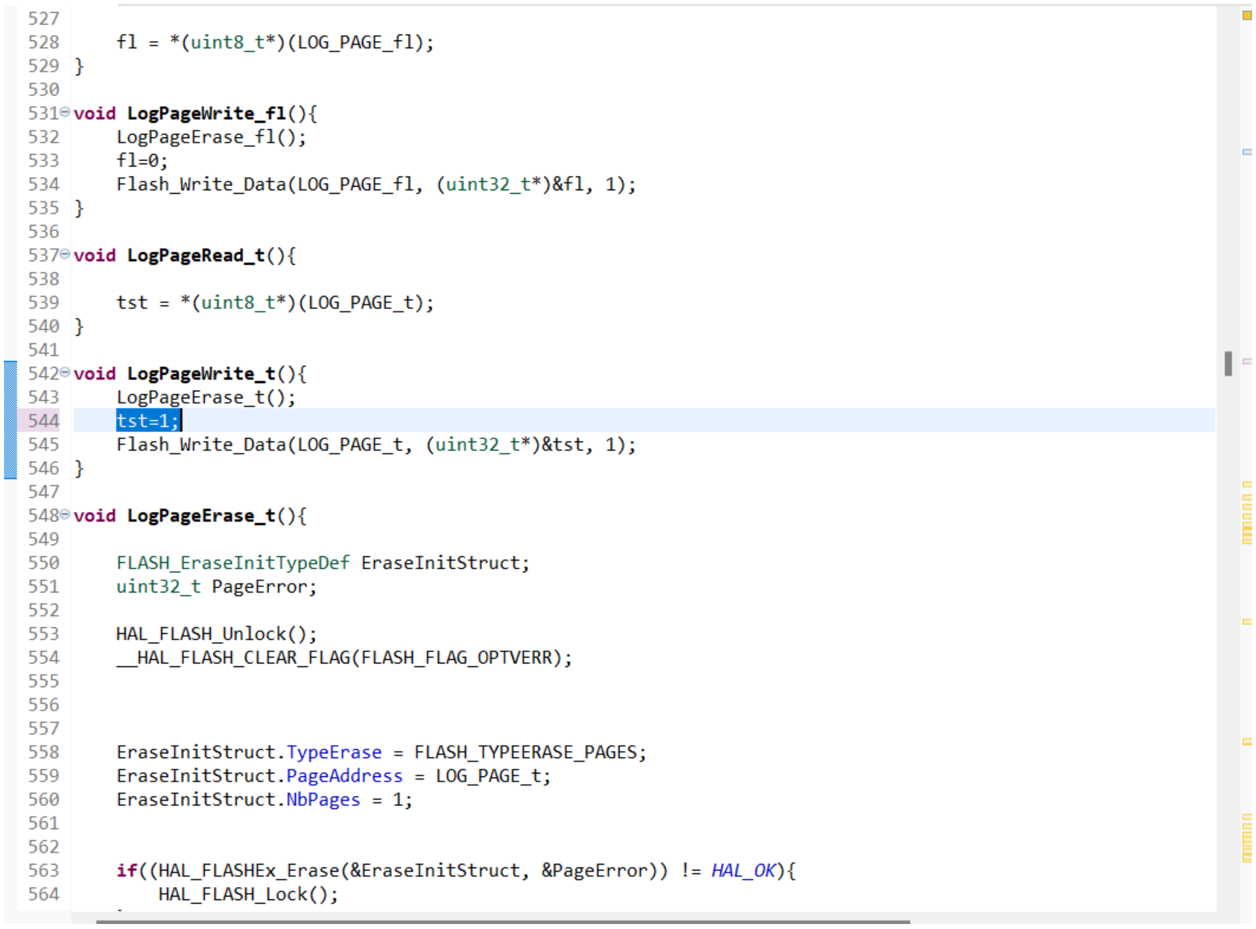
```c
              HAL_Delay(5000);
              //Sensor_Readings(uint8_t TEMP_HUM,uint8_t MOS_EC,uint8_t IRRO,uint8_t LIGHT,uint8_t SOIL,uint8_t X_v)
              Sensor_Readings(TEMP_HUM_flag,MOS_EC_flag,IRRO_flag,LIGHT_flag,SOILTEMP_flag,X_flag);
              HAL_IWDG_Refresh(&hiwdg);
              PowerUp_GSM();
              HAL_Delay(10000);
              HAL_IWDG_Refresh(&hiwdg);
              ConnectPacket();
              add_coverage(); // adding coverage in the array
              if(time_ready == 1){
                  rtc_sync();
              }
              printTime();
              HAL_IWDG_Refresh(&hiwdg);
              flash_readpub();
              if(tries != 3){
                  publishPacket(strl); // str1 is created in sensor_reading function
              }
              flash_writefinal();
              ifState = 1;
              count++;
              if(count==3){
                    LogPageWrite_fl();
                    LogPageWrite_t();
                    HAL_NVIC_SystemReset();
              }
          }else if(BAT < 270 && BAT > 261){
              HAL_IWDG_Refresh(&hiwdg);
              PowerUp_Sensor();                         //here
              HAL_Delay(5000);
              //Sensor_Readings(uint8_t TEMP_HUM,uint8_t MOS_EC,uint8_t IRRO,uint8_t LIGHT,uint8_t SOIL,uint8_t X_v)
              Sensor_Readings(TEMP_HUM_flag,MOS_EC_flag,IRRO_flag,LIGHT_flag,SOILTEMP_flag,X_flag);
              printTime();
              tries = 3;
              flash_writefinal();
              ifState = 1;

```

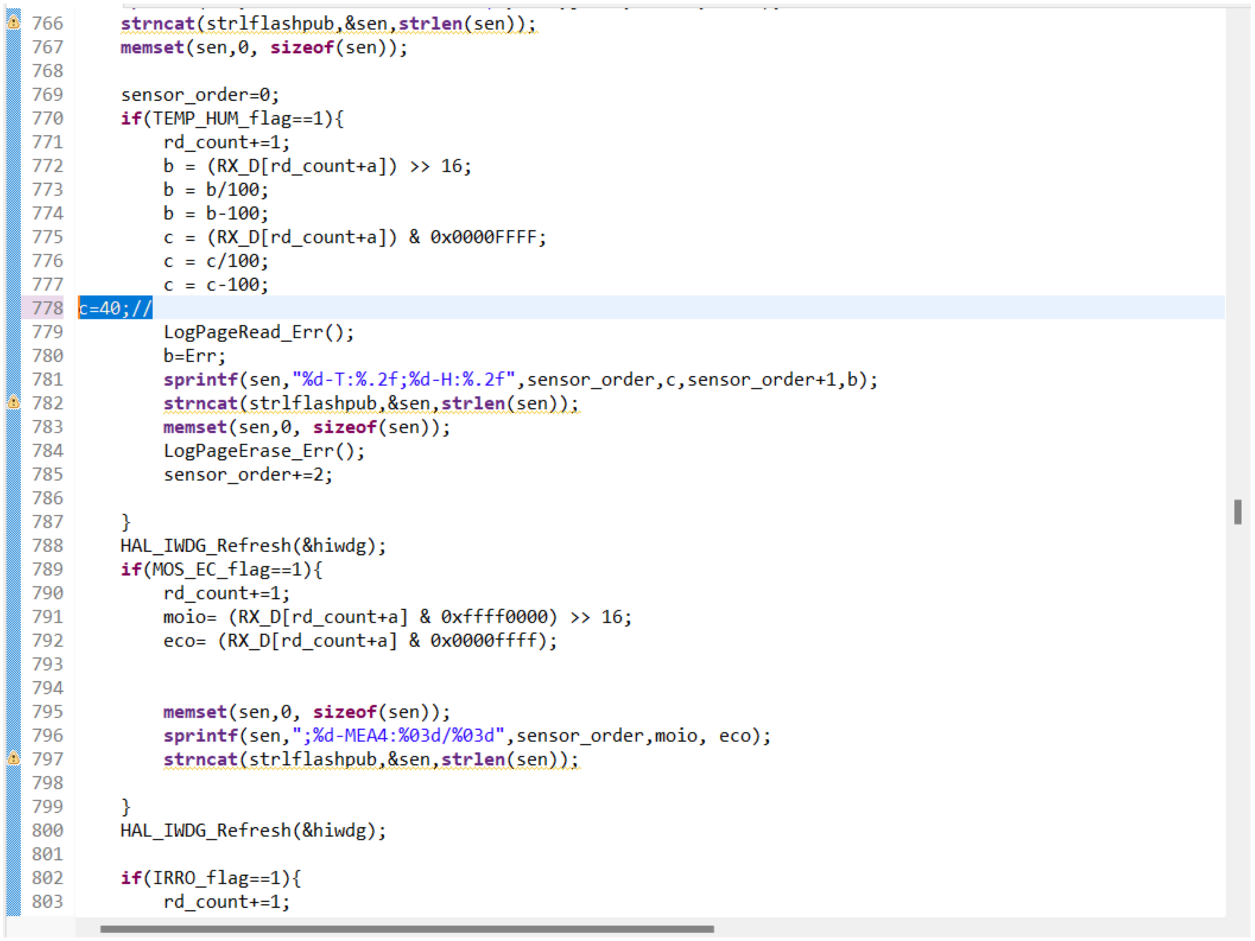


```c
    Flash_Write_Data(LOG_PAGE_fl, (uint32_t*)&fl, 1);
}

void LogPageRead_t(){

    tst = *(uint8_t*)(LOG_PAGE_t);
}

void LogPageWrite_t(){
    LogPageErase_t();
    tst=0;
    Flash_Write_Data(LOG_PAGE_t, (uint32_t*)&tst, 1);
}

void LogPageErase_t(){

    FLASH_EraseInitTypeDef EraseInitStruct;
    uint32_t PageError;

    HAL_FLASH_Unlock();
    __HAL_FLASH_CLEAR_FLAG(FLASH_FLAG_OPTVERR);



    EraseInitStruct.TypeErase = FLASH_TYPEERASE_PAGES;
    EraseInitStruct.PageAddress = LOG_PAGE_t;
    EraseInitStruct.NbPages = 1;


    if((HAL_FLASHEx_Erase(&EraseInitStruct, &PageError)) != HAL_OK){
        HAL_FLASH_Lock();
    }
    HAL_FLASH_Lock();

}

void LogPageRead_Err(){

```

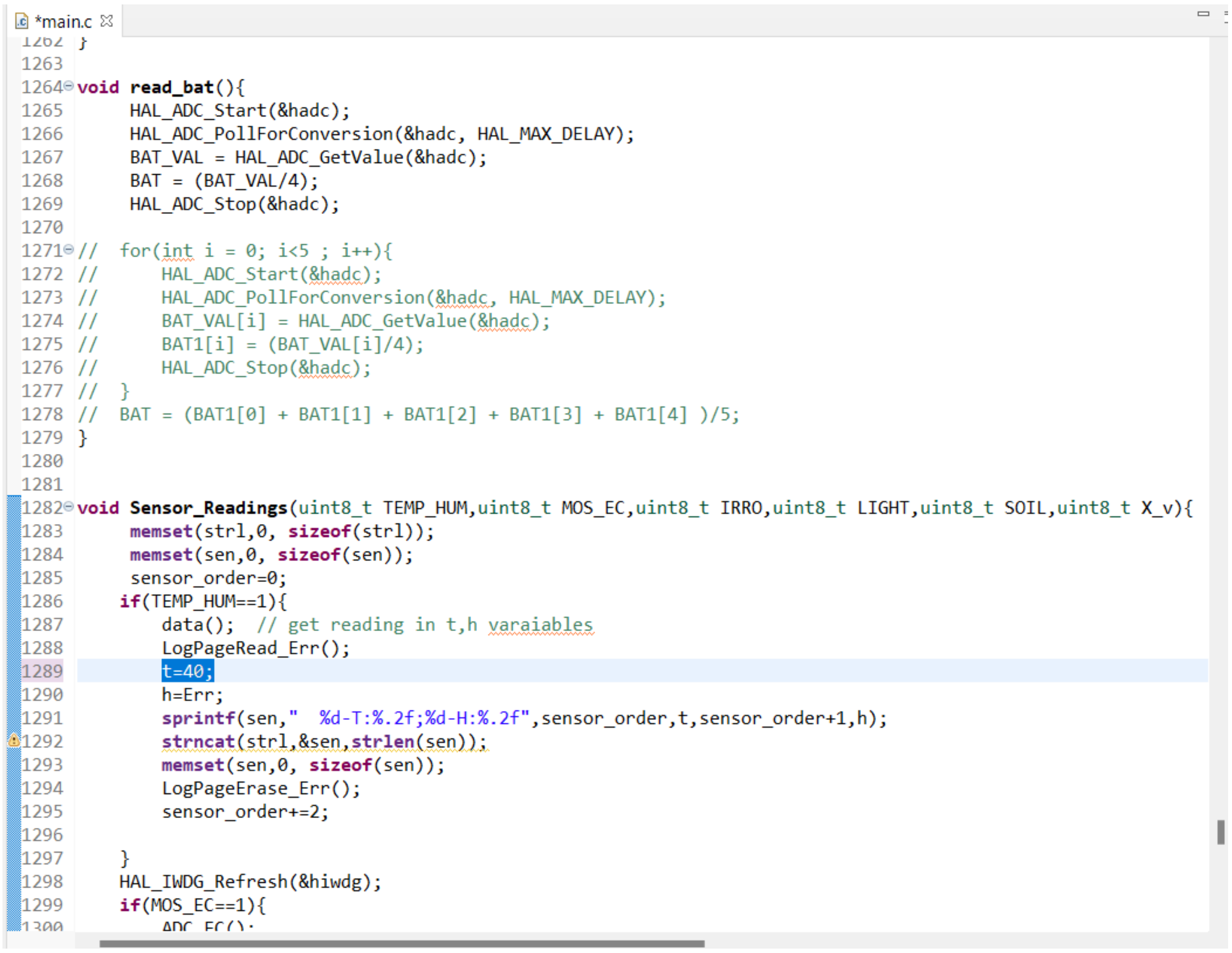
User code for FILE_URL2

- Change the following values in the Previously edited Usercode for **FILE_URL1**


```
    fl = *(uint8_t*)(LOG_PAGE_fl);
}

void LogPageWrite_fl(){
    LogPageErase_fl();
    fl=0;
    Flash_Write_Data(LOG_PAGE_fl, (uint32_t*)&fl, 1);
}

void LogPageRead_t(){

    tst = *(uint8_t*)(LOG_PAGE_t);
}

void LogPageWrite_t(){
    LogPageErase_t();
    tst=1;
    Flash_Write_Data(LOG_PAGE_t, (uint32_t*)&tst, 1);
}

void LogPageErase_t(){

    FLASH_EraseInitTypeDef EraseInitStruct;
    uint32_t PageError;

    HAL_FLASH_Unlock();
    __HAL_FLASH_CLEAR_FLAG(FLASH_FLAG_OPTVERR);



    EraseInitStruct.TypeErase = FLASH_TYPEERASE_PAGES;
    EraseInitStruct.PageAddress = LOG_PAGE_t;
    EraseInitStruct.NbPages = 1;


    if((HAL_FLASHEx_Erase(&EraseInitStruct, &PageError)) != HAL_OK){
        HAL_FLASH_Lock();
```



```
    strncat(strlflashpub,&sen,strlen(sen));
    memset(sen,0, sizeof(sen));

    sensor_order=0;
    if(TEMP_HUM_flag==1){
        rd_count+=1;
        b = (RX_D[rd_count+a]) >> 16;
        b = b/100;
        b = b-100;
        c = (RX_D[rd_count+a]) & 0x0000FFFF;
        c = c/100;
        c = c-100;
c=40;//
        LogPageRead_Err();
        b=Err;
        sprintf(sen,"%d-T:%.2f;%d-H:%.2f",sensor_order,c,sensor_order+1,b);
        strncat(strlflashpub,&sen,strlen(sen));
        memset(sen,0, sizeof(sen));
        LogPageErase_Err();
        sensor_order+=2;

    }
    HAL_IWDG_Refresh(&hiwdg);
    if(MOS_EC_flag==1){
        rd_count+=1;
        moio= (RX_D[rd_count+a] & 0xffff0000) >> 16;
        eco= (RX_D[rd_count+a] & 0x0000ffff);


        memset(sen,0, sizeof(sen));
        sprintf(sen,";%d-MEA4:%03d/%03d",sensor_order,moio, eco);
        strncat(strlflashpub,&sen,strlen(sen));

    }
    HAL_IWDG_Refresh(&hiwdg);

    if(IRRO_flag==1){
        rd_count+=1;
```



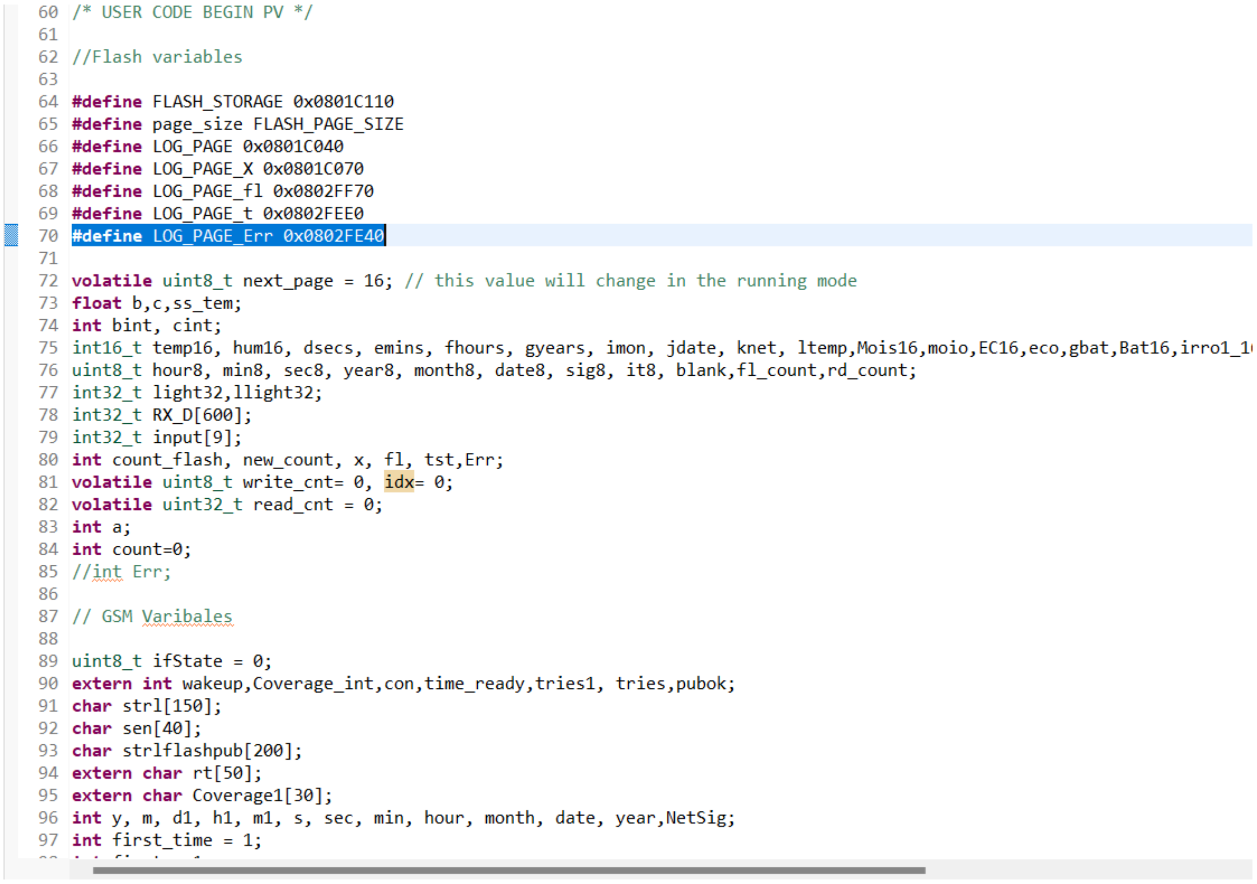
*main.c

```
}

void read_bat(){
     HAL_ADC_Start(&hadc);
     HAL_ADC_PollForConversion(&hadc, HAL_MAX_DELAY);
     BAT_VAL = HAL_ADC_GetValue(&hadc);
     BAT = (BAT_VAL/4);
     HAL_ADC_Stop(&hadc);

//  for(int i = 0; i<5 ; i++){
//      HAL_ADC_Start(&hadc);
//      HAL_ADC_PollForConversion(&hadc, HAL_MAX_DELAY);
//      BAT_VAL[i] = HAL_ADC_GetValue(&hadc);
//      BAT1[i] = (BAT_VAL[i]/4);
//      HAL_ADC_Stop(&hadc);
//  }
//  BAT = (BAT1[0] + BAT1[1] + BAT1[2] + BAT1[3] + BAT1[4] )/5;
}


void Sensor_Readings(uint8_t TEMP_HUM,uint8_t MOS_EC,uint8_t IRRO,uint8_t LIGHT,uint8_t SOIL,uint8_t X_v){
     memset(strl,0, sizeof(strl));
     memset(sen,0, sizeof(sen));
     sensor_order=0;
    if(TEMP_HUM==1){
        data();  // get reading in t,h varaiables
        LogPageRead_Err();
        t=40;
        h=Err;
        sprintf(sen,"  %d-T:%.2f;%d-H:%.2f",sensor_order,t,sensor_order+1,h);
        strncat(strl,&sen,strlen(sen));
        memset(sen,0, sizeof(sen));
        LogPageErase_Err();
        sensor_order+=2;

    }
    HAL_IWDG_Refresh(&hiwdg);
    if(MOS_EC==1){
```

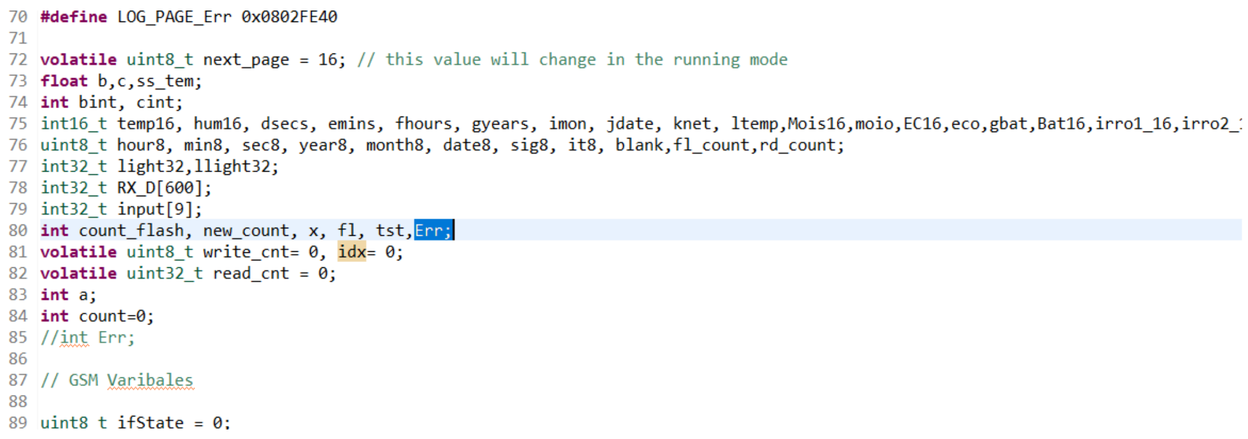
<u>To use the Bootloader with error throw</u>

- A variable for error throw should be written in the user code.
- Commit - `dd0a953`

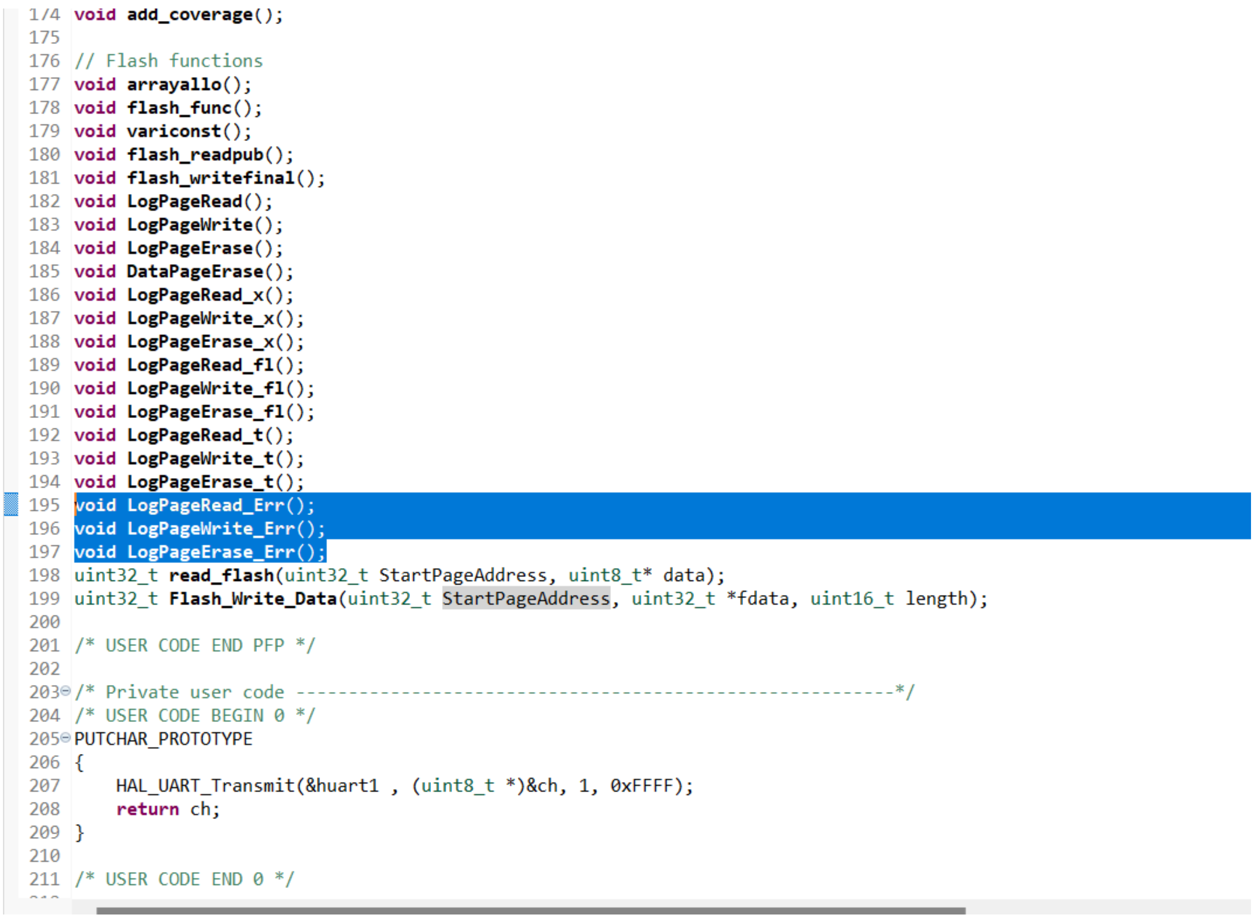
Error throw= A flash writing to display where the code downloading process stopped. This is implemented on the OTA bootloader codeBut the value is displayed with the user code publish message. The Error throw value is assigned to the Hum. idity value.



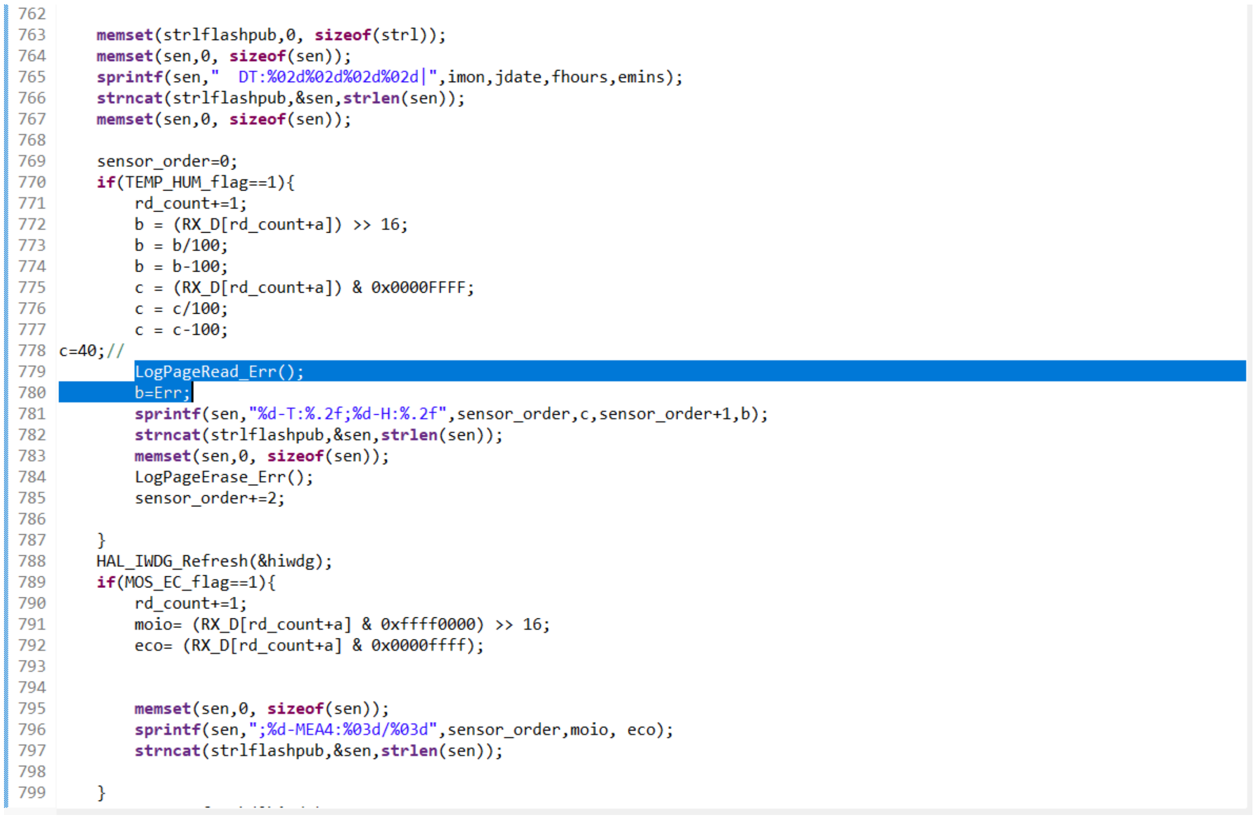
```c
/* USER CODE BEGIN PV */

//Flash variables

#define FLASH_STORAGE 0x0801C110
#define page_size FLASH_PAGE_SIZE
#define LOG_PAGE 0x0801C040
#define LOG_PAGE_X 0x0801C070
#define LOG_PAGE_fl 0x0802FF70
#define LOG_PAGE_t 0x0802FEE0
#define LOG_PAGE_Err 0x0802FE40

volatile uint8_t next_page = 16; // this value will change in the running mode
float b,c,ss_tem;
int bint, cint;
int16_t temp16, hum16, dsecs, emins, fhours, gyears, imon, jdate, knet, ltemp,Mois16,moio,EC16,eco,gbat,Bat16,irro1_1
uint8_t hour8, min8, sec8, year8, month8, date8, sig8, it8, blank,fl_count,rd_count;
int32_t light32,llight32;
int32_t RX_D[600];
int32_t input[9];
int count_flash, new_count, x, fl, tst,Err;
volatile uint8_t write_cnt= 0, idx= 0;
volatile uint32_t read_cnt = 0;
int a;
int count=0;
//int Err;

// GSM Varibales

uint8_t ifState = 0;
extern int wakeup,Coverage_int,con,time_ready,tries1, tries,pubok;
char strl[150];
char sen[40];
char strlflashpub[200];
extern char rt[50];
extern char Coverage1[30];
int y, m, d1, h1, m1, s, sec, min, hour, month, date, year,NetSig;
int first_time = 1;
```

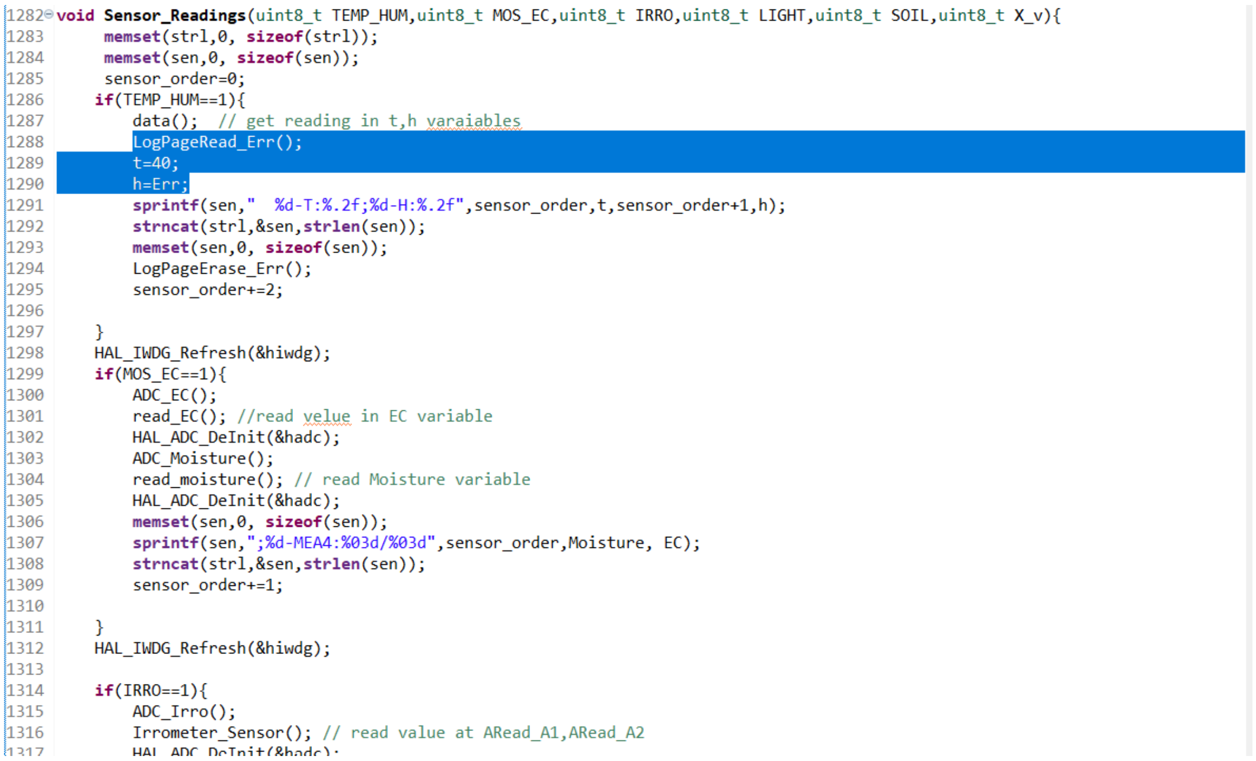


```c
#define LOG_PAGE_Err 0x0802FE40

volatile uint8_t next_page = 16; // this value will change in the running mode
float b,c,ss_tem;
int bint, cint;
int16_t temp16, hum16, dsecs, emins, fhours, gyears, imon, jdate, knet, ltemp,Mois16,moio,EC16,eco,gbat,Bat16,irro1_16,irro2_
uint8_t hour8, min8, sec8, year8, month8, date8, sig8, it8, blank,fl_count,rd_count;
int32_t light32,llight32;
int32_t RX_D[600];
int32_t input[9];
int count_flash, new_count, x, fl, tst,Err;
volatile uint8_t write_cnt= 0, idx= 0;
volatile uint32_t read_cnt = 0;
int a;
int count=0;
//int Err;

// GSM Varibales

uint8_t ifState = 0;
```

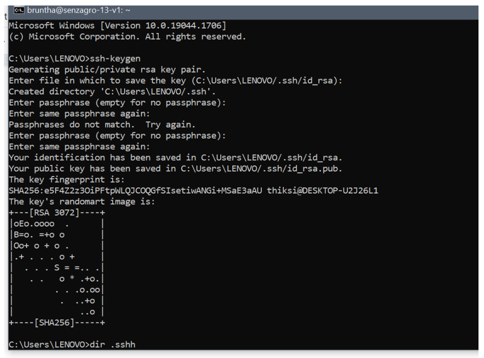


```c
void add_coverage();

// Flash functions
void arrayallo();
void flash_func();
void variconst();
void flash_readpub();
void flash_writefinal();
void LogPageRead();
void LogPageWrite();
void LogPageErase();
void DataPageErase();
void LogPageRead_x();
void LogPageWrite_x();
void LogPageErase_x();
void LogPageRead_fl();
void LogPageWrite_fl();
void LogPageErase_fl();
void LogPageRead_t();
void LogPageWrite_t();
void LogPageErase_t();
void LogPageRead_Err();
void LogPageWrite_Err();
void LogPageErase_Err();
uint32_t read_flash(uint32_t StartPageAddress, uint8_t* data);
uint32_t Flash_Write_Data(uint32_t StartPageAddress, uint32_t *fdata, uint16_t length);

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
PUTCHAR_PROTOTYPE
{
    HAL_UART_Transmit(&huart1 , (uint8_t *)&ch, 1, 0xFFFF);
    return ch;
}

/* USER CODE END 0 */
```

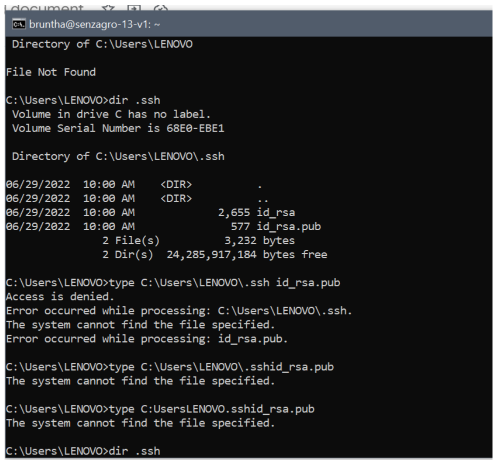

```c

    memset(strlflashpub,0, sizeof(strl));
    memset(sen,0, sizeof(sen));
    sprintf(sen,"  DT:%02d%02d%02d%02d|",imon,jdate,fhours,emins);
    strncat(strlflashpub,&sen,strlen(sen));
    memset(sen,0, sizeof(sen));

    sensor_order=0;
    if(TEMP_HUM_flag==1){
        rd_count+=1;
        b = (RX_D[rd_count+a]) >> 16;
        b = b/100;
        b = b-100;
        c = (RX_D[rd_count+a]) & 0x0000FFFF;
        c = c/100;
        c = c-100;
c=40;//
        LogPageRead_Err();
        b=Err;
        sprintf(sen,"%d-T:%.2f;%d-H:%.2f",sensor_order,c,sensor_order+1,b);
        strncat(strlflashpub,&sen,strlen(sen));
        memset(sen,0, sizeof(sen));
        LogPageErase_Err();
        sensor_order+=2;

    }
    HAL_IWDG_Refresh(&hiwdg);
    if(MOS_EC_flag==1){
        rd_count+=1;
        moio= (RX_D[rd_count+a] & 0xffff0000) >> 16;
        eco= (RX_D[rd_count+a] & 0x0000ffff);


        memset(sen,0, sizeof(sen));
        sprintf(sen,";%d-MEA4:%03d/%03d",sensor_order,moio, eco);
        strncat(strlflashpub,&sen,strlen(sen));

    }
```


```c
void Sensor_Readings(uint8_t TEMP_HUM,uint8_t MOS_EC,uint8_t IRRO,uint8_t LIGHT,uint8_t SOIL,uint8_t X_v){
     memset(strl,0, sizeof(strl));
     memset(sen,0, sizeof(sen));
     sensor_order=0;
    if(TEMP_HUM==1){
        data();  // get reading in t,h varaiables
        LogPageRead_Err();
        t=40;
        h=Err;
        sprintf(sen,"  %d-T:%.2f;%d-H:%.2f",sensor_order,t,sensor_order+1,h);
        strncat(strl,&sen,strlen(sen));
        memset(sen,0, sizeof(sen));
        LogPageErase_Err();
        sensor_order+=2;

    }
    HAL_IWDG_Refresh(&hiwdg);
    if(MOS_EC==1){
        ADC_EC();
        read_EC(); //read velue in EC variable
        HAL_ADC_DeInit(&hadc);
        ADC_Moisture();
        read_moisture(); // read Moisture variable
        HAL_ADC_DeInit(&hadc);
        memset(sen,0, sizeof(sen));
        sprintf(sen,";%d-MEA4:%03d/%03d",sensor_order,Moisture, EC);
        strncat(strl,&sen,strlen(sen));
        sensor_order+=1;

    }
    HAL_IWDG_Refresh(&hiwdg);

    if(IRRO==1){
        ADC_Irro();
        Irrometer_Sensor(); // read value at ARead_A1,ARead_A2
        HAL_ADC_DeInit(&hadc);
```

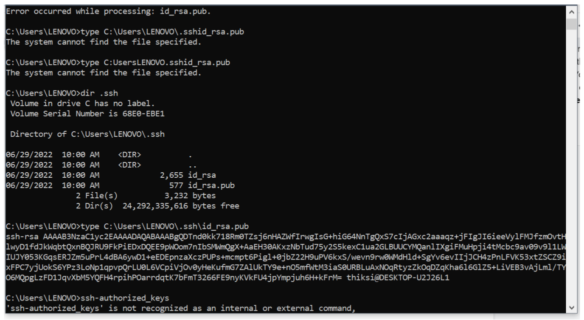
SSH key generation and verification of the access



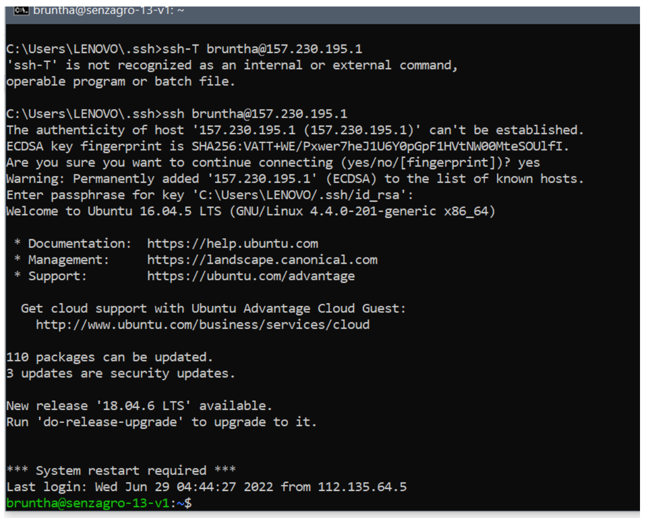
```
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\LENOVO/.ssh/id_rsa):
Created directory 'C:\Users\LENOVO/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match.  Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\LENOVO/.ssh/id_rsa.
Your public key has been saved in C:\Users\LENOVO/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:e5F4Z2z3OiPFtpWLQJCOQGfSIsetiwANGi+MSaE3aAU thiksi@DESKTOP-U2J26L1
The key's randomart image is:
+---[RSA 3072]----+
|oEo.oooo  .      |
|B=o. =+o o       |
|Oo+ o + o .      |
|.+ . . . o +     |
|  . . . S = =.. .|
|   . .   o * .+o.|
|         . . .o.oo|
|          .  ..+o |
|              ..o |
+----[SHA256]-----+

C:\Users\LENOVO>dir .sshh
```



```
 Directory of C:\Users\LENOVO

File Not Found

C:\Users\LENOVO>dir .ssh
 Volume in drive C has no label.
 Volume Serial Number is 68E0-EBE1

 Directory of C:\Users\LENOVO\.ssh

06/29/2022  10:00 AM    <DIR>          .
06/29/2022  10:00 AM    <DIR>          ..
06/29/2022  10:00 AM             2,655 id_rsa
06/29/2022  10:00 AM               577 id_rsa.pub
               2 File(s)          3,232 bytes
               2 Dir(s)  24,285,917,184 bytes free

C:\Users\LENOVO>type C:\Users\LENOVO\.ssh id_rsa.pub
Access is denied.
Error occurred while processing: C:\Users\LENOVO\.ssh.
The system cannot find the file specified.
Error occurred while processing: id_rsa.pub.

C:\Users\LENOVO>type C:\Users\LENOVO\.sshid_rsa.pub
The system cannot find the file specified.

C:\Users\LENOVO>type C:UsersLENOVO.sshid_rsa.pub
The system cannot find the file specified.

C:\Users\LENOVO>dir .ssh
```



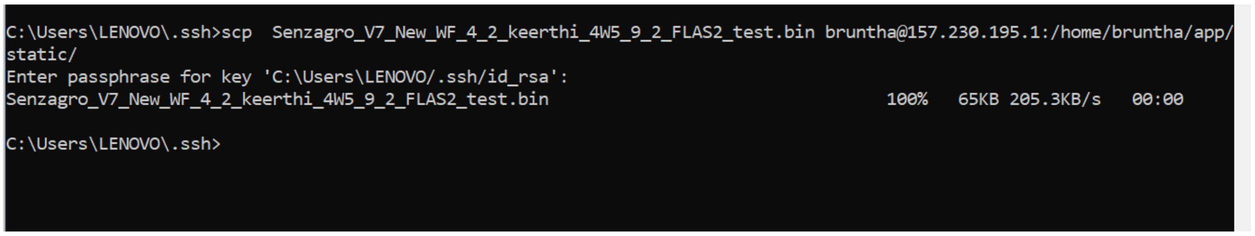
```
Error occurred while processing: id_rsa.pub.

C:\Users\LENOVO>type C:\Users\LENOVO\.sshid_rsa.pub
The system cannot find the file specified.

C:\Users\LENOVO>type C:UsersLENOVO.sshid_rsa.pub
The system cannot find the file specified.

C:\Users\LENOVO>dir .ssh
 Volume in drive C has no label.
 Volume Serial Number is 68E0-EBE1

 Directory of C:\Users\LENOVO\.ssh

06/29/2022  10:00 AM    <DIR>          .
06/29/2022  10:00 AM    <DIR>          ..
06/29/2022  10:00 AM             2,655 id_rsa
06/29/2022  10:00 AM               577 id_rsa.pub
               2 File(s)          3,232 bytes
               2 Dir(s)  24,292,335,616 bytes free

C:\Users\LENOVO>type C:\Users\LENOVO\.ssh\id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDTnd0kk718Rm0TZsj6nHAZWfIrwgIsG+hiG64NnTgQxS7cIjAGxc2aaaqz+jFIgJI6ieeVylFMJfzmOvtH
lwyD1fdJkWqbtQxnBQJRU9FkPiEDxDQEE9pWOom7nIbSMWmQgX+AaEH30AKxzNbTud75y2S5kexC1ua2GLBUUCYMQanlIXgiFMuHpji4tMcbc9av09v9l1LW
IUJY053KGqsERJZm5uPrL4dBA6ywD1+eEDEpnzaXczPUPs+mcmpt6Pigl+0jbZ22H9uPV6kxS/wevn9rw0WMdHld+SgYv6evIIjJCH4zPnLFVK53xtZSCZ9i
xFPC7yjUokS6YPz3LoNp1qpvpQrLU0L6VCpiVjOv0yHeKufmG7ZAlUkTY9e+nO5mfWtM3iaS0URBLuAxNOqRtyzZkOqDZqKha6l6GlZ5+LiVEB3vAjLml/TY
O6MQpgLzFD1JqvXbM5YQFH4rpihPOarrdqtK7bFmT3266FE9nyKVkFU4jpYmpjuh6H+kFrM= thiksi@DESKTOP-U2J26L1

C:\Users\LENOVO>ssh-authorized_keys
'ssh-authorized_keys' is not recognized as an internal or external command,
```



```
C:\Users\LENOVO\.ssh>ssh-T bruntha@157.230.195.1
'ssh-T' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\LENOVO\.ssh>ssh bruntha@157.230.195.1
The authenticity of host '157.230.195.1 (157.230.195.1)' can't be established.
ECDSA key fingerprint is SHA256:VATT+WE/Pxwer7heJ1U6Y0pGpF1HVtNW00MteSOUlfI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '157.230.195.1' (ECDSA) to the list of known hosts.
Enter passphrase for key 'C:\Users\LENOVO/.ssh/id_rsa':
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-201-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

110 packages can be updated.
3 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Wed Jun 29 04:44:27 2022 from 112.135.64.5
bruntha@senzagro-13-v1:~$
```

ssh-keygen
Ender
Password (when typing; it won't be visible as typing/ cursor won't move. So just type and enter)
dir .ssh (for confirm file)
type C:\Users\LENOVO\.ssh\id_rsa.pub
*bruntha anna have to give the access
After getting the access; Check the access
C:\Users\LENOVO\.ssh>ssh bruntha@157.230.195.1
yes

BIN FILE UPLOAD TO SERVER



```
C:\Users\LENOVO\.ssh>scp  Senzagro_V7_New_WF_4_2_keerthi_4W5_9_2_FLAS2_test.bin bruntha@157.230.195.1:/home/bruntha/app/
static/
Enter passphrase for key 'C:\Users\LENOVO/.ssh/id_rsa':
Senzagro_V7_New_WF_4_2_keerthi_4W5_9_2_FLAS2_test.bin                           100%   65KB 205.3KB/s   00:00

C:\Users\LENOVO\.ssh>
```

- Your file path> scp File name with extension bruntha@157.230.195.1:/home/bruntha/app/static/
- Enter the passphrase you assigned before
- URL:- http://agro.senzmate.com:8082/(uploarded file name)

File names can't have '(2)' like structure.

For each device, the client ID should be different in Usercode. So, binfile should be created with a relevant client ID and each bin file should be updated to the server.

EXISTING FEATURES

ERROR THROW FEATURE

Flash written

Address= LOG_PAGE_Err 0x0802FE40

User code Humidity value is set to display Error throw value.

1=downloadState-0	6=downloadState-5
2=downloadState-1	7=downloadState-6
3=downloadState-2	8=downloadState-7
4=downloadState-3	9=HTTPREAD()
5=downloadState-4	

```
Acdb21a = bitbucket link head-bootloader
```

Currently; error is around downloadStates 1,2

Auto binfile select (tst variable)

9_3_5- tst=1-T=20;

9_3_6- tst=0-T=40;

"http://agro.senzmate.com:8082/Senzagro_V7_New_WF_4_2_keerthi_4W5_9_3_5.bin"

"http://agro.senzmate.com:8082/Senzagro_V7_New_WF_4_2_keerthi_4W5_9_3_6.bin"

Download size &time optimisation

RX_READ_TIMEOUT= 20 s
Download size= 16kb
Writing 8 pages at once. 128/8 loops per download

Output:-

Total time for download:- <6 min (00:05:45)
For downloading and writing only:- nearly 00:02:42
Connection establishment:- 00:03:45

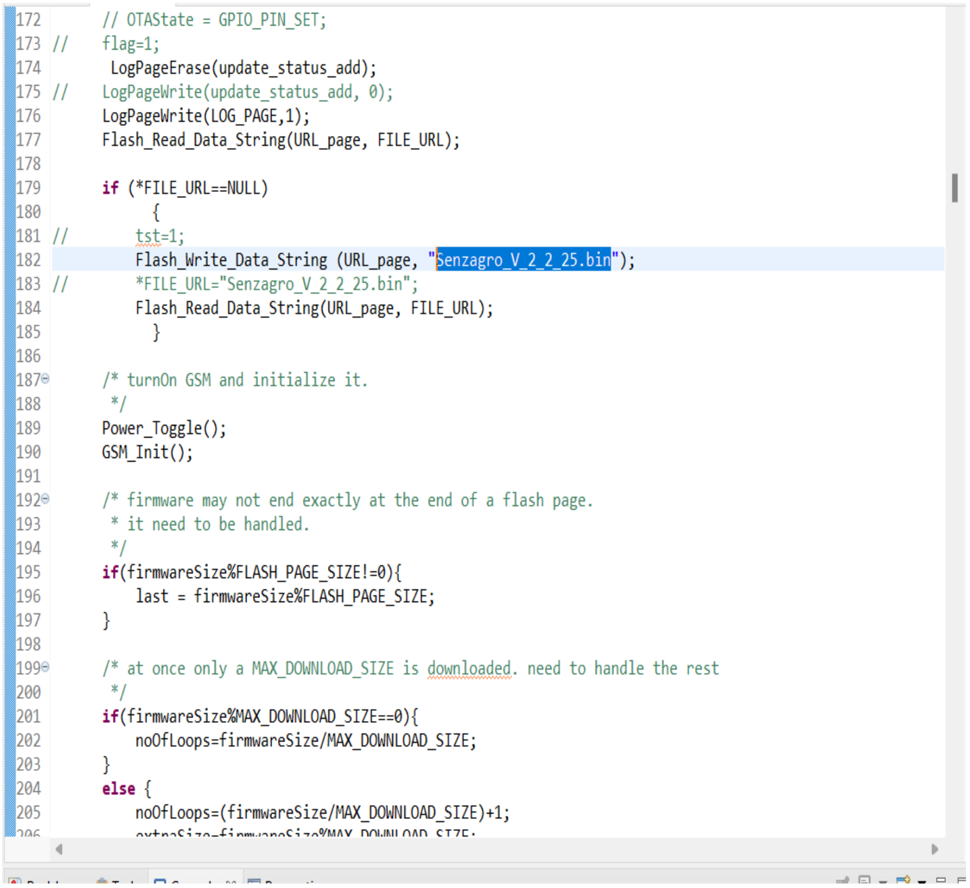
OTA parameters received through Subscribed message
bootloader=> commit no:-

`87fd342`

https://bitbucket.org/embedded-senz-agro/ota/commits/87fd3422295cb1a0f95e94539d8e7fa3a4e6110d

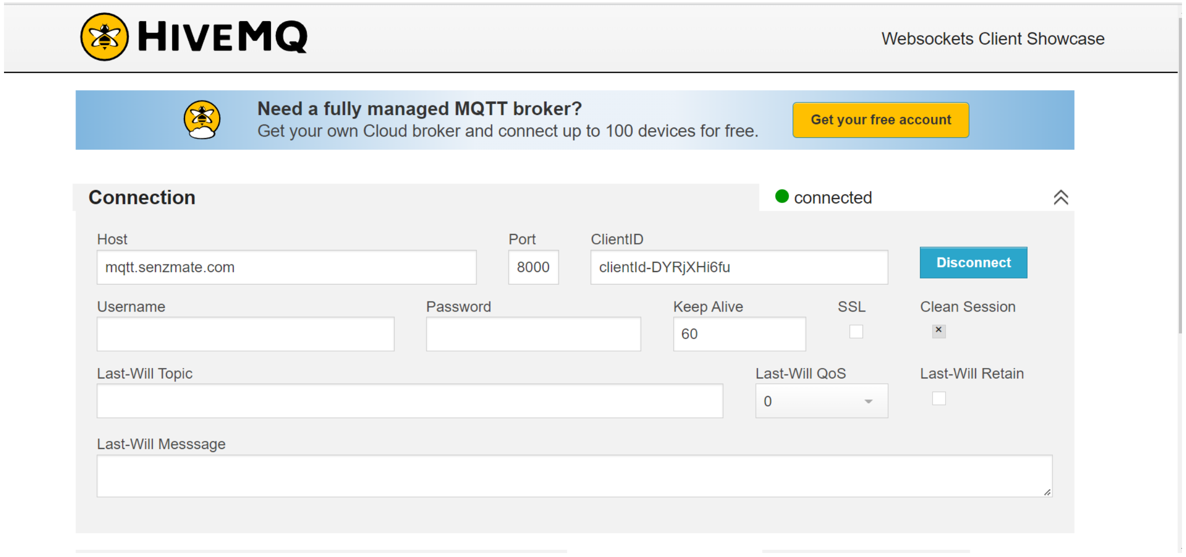
PROCEDURE

1. Clean the flash memory
2. Write the standard version of the board through the Cube programmer at 0x0802FF70 memory address.
3. Disconnect the cube programmer
4. Debug the bootloader code
5. Initially it will download the binfile Senzagro_V_2_2_25.bin.
    a. Upload a new binfile
    b. Change that file name to the relevant place as bellow(line-182)



```
    // OTAState = GPIO_PIN_SET;
//  flag=1;
     LogPageErase(update_status_add);
//  LogPageWrite(update_status_add, 0);
    LogPageWrite(LOG_PAGE,1);
    Flash_Read_Data_String(URL_page, FILE_URL);

    if (*FILE_URL==NULL)
          {
//      tst=1;
        Flash_Write_Data_String (URL_page, "Senzagro_V_2_2_25.bin");
//      *FILE_URL="Senzagro_V_2_2_25.bin";
        Flash_Read_Data_String(URL_page, FILE_URL);
          }

    /* turnOn GSM and initialize it.
     */
    Power_Toggle();
    GSM_Init();

    /* firmware may not end exactly at the end of a flash page.
     * it need to be handled.
     */
    if(firmwareSize%FLASH_PAGE_SIZE!=0){
        last = firmwareSize%FLASH_PAGE_SIZE;
    }

    /* at once only a MAX_DOWNLOAD_SIZE is downloaded. need to handle the rest
     */
    if(firmwareSize%MAX_DOWNLOAD_SIZE==0){
        noOfLoops=firmwareSize/MAX_DOWNLOAD_SIZE;
    }
    else {
        noOfLoops=(firmwareSize/MAX_DOWNLOAD_SIZE)+1;
        extraSize=firmwareSize%MAX_DOWNLOAD_SIZE;
```

6. Open the HiveMQ portal



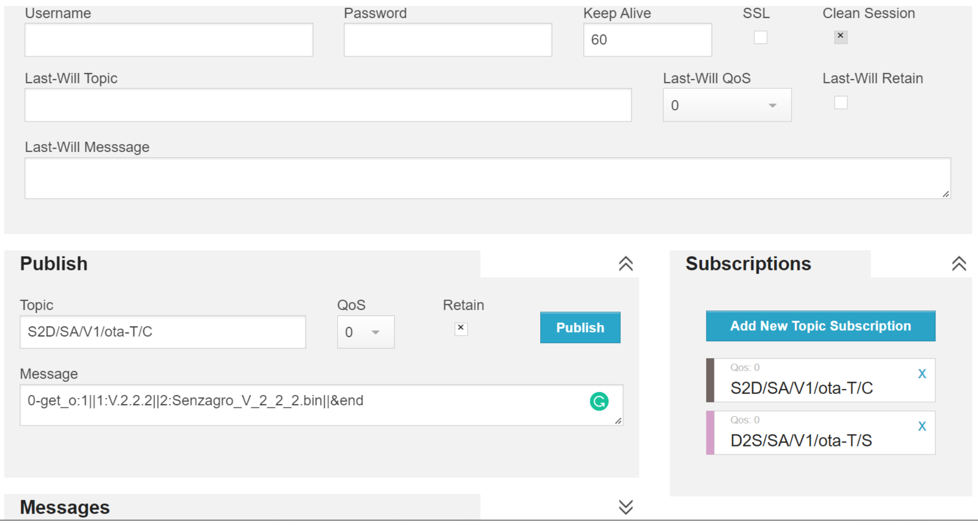
Need a fully managed MQTT broker?
Get your own Cloud broker and connect up to 100 devices for free.

Connection ● connected

Host	mqtt.senzmate.com	
Port	8000	
ClientID	clientId-DYRjXHi6fu	
Username		
Password		
Keep Alive	60	
SSL	☐	
Clean Session	☒	
Last-Will Topic		
Last-Will QoS	0	
Last-Will Retain	☐	
Last-Will Messsage		

7.

8. Subscribe to both S2D and D2S topics and fill the published portion with S2D topic and the message



Username

Password

Keep Alive: 60

SSL

Clean Session: ✓

Last-Will Topic

Last-Will QoS: 0

Last-Will Retain

Last-Will Messsage

Publish

Topic: S2D/SA/V1/ota-T/C

QoS: 0

Retain: ✓

Message: 0-get_o:1||1:V.2.2.2||2:Senzagro_V_2_2_2.bin||&end

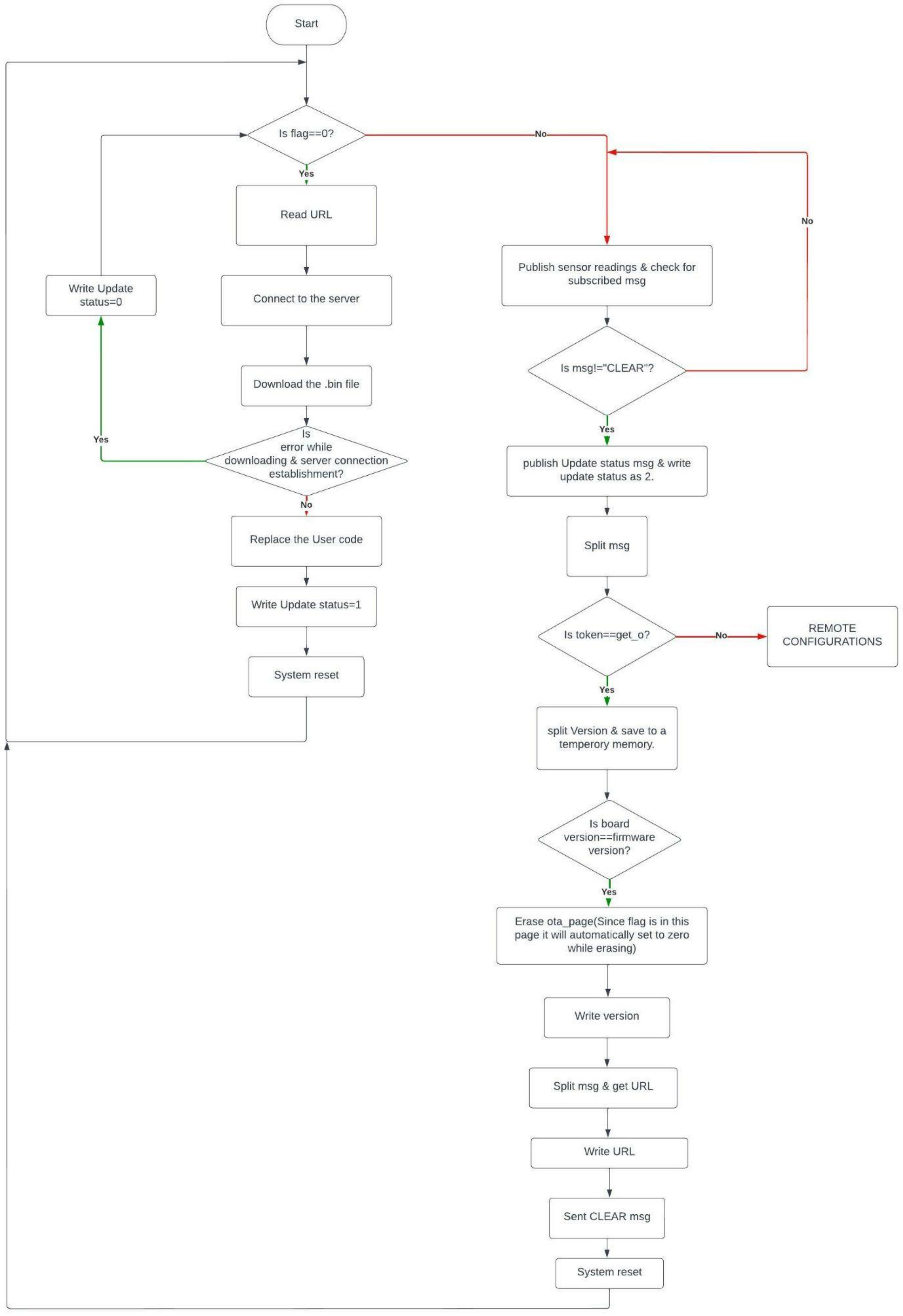
Subscriptions

Qos: 0 S2D/SA/V1/ota-T/C

Qos: 0 D2S/SA/V1/ota-T/S

Messages

9. Numbering meanings
   1-version of the code
   2-bin file name with extension
10. Publish the message


Start
Is flag==0?
Yes
No
Read URL
Connect to the server
Download the .bin file
Is error while downloading & server connection establishment?
Yes
No
Write Update status=0
Replace the User code
Write Update status=1
System reset
Publish sensor readings & check for subscribed msg
Is msg!="CLEAR"?
No
Yes
publish Update status msg & write update status as 2.
Split msg
Is token==get_o?
No
REMOTE CONFIGURATIONS
Yes
split Version & save to a temperory memory.
Is board version==firmware version?
Yes
Erase ota_page(Since flag is in this page it will automatically set to zero while erasing)
Write version
Split msg & get URL
Write URL
Sent CLEAR msg
System reset

TESTING RESULTS

OTA download continuous testing (3/8/2022-9/8/2022)

9.24 am to 9.24 am

8640 min/27 min=320 downloads

Actually happened= 294

Error rate= 8.1%

Increased downloading size and time optimization

03:09:38 -03:14:54 (2d 5min)-2885 mins

Downloads…

Have to be= 2885/26=110 downloads

Occurred=98

Error rate=11%

DAILY ERROR RATE ALSO AROUND 11%; ERROR RATE INCREASED (have to check on that……………………………..)

CONTINUOUS TESTING DEVICE 1 (OTA FARM1)

TIME=23/8/2022-20/9/2022

ERROR RATE=20% daily

ERR=1%=7=7

ERR=3 %=50+56+46+56+17+24=249

ERR=4%=1+2+1=4

ERR=5%=1+1=2

Error while downloading by HTTPREAD() is high next to err=3 like nearly 50.

On 4,5,6/9/2022 dates the downloading collapsed most. While analyzing it, the signal strength is 19 on average. On other days signal strength is above 20 and the average is around 21. Moreover, the error percentage is around 20%. Previously while the error percentage was 11%; signal strength is in the range between 25 and 30

FOR FUTURE PROGRESS

OTA VALIDATION RESEARCH

Possible errors

1. **time window has expired** - is returned when the time window has been exceeded, perhaps the result of a DoS attack. Note that a vendor service may learn about update failures by simply using a query() operation (as defined in the smart contract). In case an update fails for a particular device, the vendor service is responsible for re-initiating the operation.
2. **too many failed attempts** - is returned to indicate too many consecutive failures or attempts, as this could be the result of a device under attack (perhaps by overloading the verification interface).
3. **hash code mismatch** -is returned when the hash calculated by the IoT device does not match the associated Securing Over–The–Air IoT Firmware Updates using Blockchain.

Checks have to be done (For existing blockchain mechanism..)

a. **transaction_id**: a unique ID to reference a firmware update initiated by a vendor service
b. **timestamp_send:** a timestamp recording the time a vendor service pushes the update to the blockchain
c. **timestamp_receive:** a timestamp recording the time when the blockchain accepts and commits the transaction to the ledger
d. **device_type:** the IoT device type
e. **firmware_version:** the version of the firmware update
f. **firmware_hash:** SHA1 hash of the firmware update binary as provided by the vendor service
g. **status:** to indicate whether the targeted IoT device succeeded in updating the firmware. Its initial value is "unverified"
h. **failed_attempts:** a counter recording the number of failed firmware update attempts.

How to do checkings

- Code has to be sent with those parameters. (eg- through the Blockchain layer)
- From those parameters the code has to do checkings.
    - Version check
    - compare the hash value received and the calculated hash value of the downloaded code.

SUMMARY

- Have to send the hash/CRC value of the error-less code with the code to download.
- For high-level validation Other parameters also have to be sent with the bin file
- In the bootloader, the CRC/hash value of the downloaded code has to be calculated.
- If CRC calculated==CRC received; errorless
- For high-level check; other parameters also have to be checked.

REFERENCE

1. Blockchain mechanism- https://www.researchgate.net/publication/332278969_Securing_Over-The-Air_IoT_Firmware_Updates_using_Blockchain
2. Simple checking- https://www.ndss-symposium.org/wp-content/uploads/autosec2021_23028_paper.pdf https://github.com/Koenkk/zigbee-herdsman-converters/issues/1998
3. https://github.com/Embetronicx/STM32-Bootloader/blob/ETX_Bootloader_4.0/Bootloader_Example/Bootloader/Core/Src/main.c
4. https://ww1.microchip.com/downloads/en/AppNotes/00730a.pdf

HAVE TO BE DONE……………………

- [ ] Error handling
- [ ] OTA validation code
- [ ] Subscribed flag in user code

Subscribed msg to flag control in user code (HAVE TO BE DONE)

1. Read the retain msg from gsmreply1.
2. Sample function is in polar code.