New issue Jump to bottom

OTAA example aborts on Mcu.init() call on WiFi LoRa 32 V2 board #91

Open **jcwren** opened this issue on Oct 10, 2022 · 13 comments

jcwren commented on Oct 10, 2022

This seems to be the same issue as #90. I have a license key and replaced that into the OTAA.ino file. After programming, the ESP32 is boot looping with the following output:

```
ESP32 MCU init...

abort() was called at PC 0x400820a2 on core 1

Backtrace:0x4008376d:0x3ffb26700x40088fe1:0x3ffb2690 0x4008dda9:0x3ffb26b0
0x400820a2:0x3ffb2730 0x40082674:0x3ffb2780 0x400d17f8:0x3ffb27a0
0x400d1e71:0x3ffb27c0 0x400d1579:0x3ffb27f0 0x400d4e1a:0x3ffb2820

ELF file SHA256: 0000000000000000
```

If the sentence from the README.md are relevant to PlatformIO, it is COMPLETELY unclear how it would be installed.

```
This library must work with newest [Heltec_ESP32 development framework]
(https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series)
For now, new features still not included in the release version, must install this
framework via Git.
```

platformio.ini:

```
[env:heltec_wifi_lora_32_V2]
platform = espressif32
board = heltec_wifi_lora_32_V2
framework = arduino
lib_deps = heltecautomation/ESP32_LoRaWAN@^2.1.1
lib_ldf_mode = deep+
build_flags =
   -D REGION_US915
   -D ACTIVE_REGION=LORAMAC_REGION_US915
   -D LoRaWAN_DEBUG_LEVEL=0
monitor_speed = 115200
```

```
/*
 * HelTec Automation(TM) LoRaWAN 1.0.2 OTAA example use OTAA, CLASS A
 *
 * Function summary:
 *
 * - use internal RTC(150KHz);
 *
 * - Include stop mode and deep sleep mode;
 *
 * - 15S data send cycle;
 *
 * - Informations output via serial(115200);
 *
 * - Only ESP32 + LoRa series boards can use this library, need a license
 *   to make the code run(check you license here: http://www.heltec.cn/search/);
 *
 * You can change some definition in "Commissioning.h" and "LoRaMac-definitions.h"
 *
 * HelTec AutoMation, Chengdu, China.
 * 成都惠利特自动化科技有限公司
 * https://heltec.org
 * support@heltec.cn
 *
 *this project also release in GitHub:
 *https://github.com/HelTecAutomation/ESP32_LoRaWAN
*/

#include <ESP32_LoRaWAN.h>
#include "Arduino.h"

/*license for Heltec ESP32 LoRaWan, quary your ChipID relevant license:
http://resource.heltec.cn/search */
uint32_t  license[4] = {0x6A3F7214, 0x0BCB2C9B, 0xFE0A5C7D, 0x3F7DE4D3};

/* OTAA para*/
uint8_t DevEui[] = { 0x22, 0x32, 0x33, 0x00, 0x00, 0x88, 0x88, 0x02 };
uint8_t AppEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
uint8_t AppKey[] = { 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
0x88, 0x88, 0x88, 0x66, 0x01 };

/* ABP para*/
uint8_t NwkSKey[] = { 0x15, 0xb1, 0xd0, 0xef, 0xa4, 0x63, 0xdf, 0xbe, 0x3d, 0x11,
0x18, 0x1e, 0x1e, 0xc7, 0xda,0x85 };
uint8_t AppSKey[] = { 0xd7, 0x2c, 0x78, 0x75, 0x8c, 0xdc, 0xca, 0xbf, 0x55, 0xee,
0x4a, 0x77, 0x8d, 0x16, 0xef,0x67 };
uint32_t DevAddr =  ( uint32_t )0x007e6ae1;

/*LoraWan channelsmask, default channels 0-7*/
uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };

/*LoraWan Class, Class A and Class C are supported*/
DeviceClass_t  loraWanClass = CLASS_A;

/*the application data transmission duty cycle.  value in [ms].*/
uint32_t appTxDutyCycle = 15000;
```

```
/* ADR enable */
bool loraWanAdr = true;

/* Indicates if the node is sending confirmed or unconfirmed messages */
bool isTxConfirmed = true;

/* Application port */
uint8_t appPort = 2;

/*!
* Number of trials to transmit the frame, if the LoRaMAC layer did not
* receive an acknowledgment. The MAC performs a datarate adaptation,
* according to the LoRaWAN Specification V1.0.2, chapter 18.4, according
* to the following table:
*
* Transmission nb | Data Rate
* ----------------|-----------
* 1 (first)       | DR
* 2               | DR
* 3               | max(DR-1,0)
* 4               | max(DR-1,0)
* 5               | max(DR-2,0)
* 6               | max(DR-2,0)
* 7               | max(DR-3,0)
* 8               | max(DR-3,0)
*
* Note, that if NbTrials is set to 1 or 2, the MAC will not decrease
* the datarate, in case the LoRaMAC layer did not receive an acknowledgment
*/
uint8_t confirmedNbTrials = 8;

/*LoraWan debug level, select in arduino IDE tools.
* None : print basic info.
* Freq : print Tx and Rx freq, DR info.
* Freq && DIO : print Tx and Rx freq, DR, DIO0 interrupt and DIO1 interrupt info.
* Freq && DIO && PW: print Tx and Rx freq, DR, DIO0 interrupt, DIO1 interrupt and MCU
deepsleep info.
*/
uint8_t debugLevel = LoRaWAN_DEBUG_LEVEL;

/*LoraWan region, select in arduino IDE tools*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

static void prepareTxFrame( uint8_t port )
{
    appDataSize = 4;//AppDataSize max value is 64 ( src/Commissioning.h -> 128 )
    appData[0] = 0x00;
    appData[1] = 0x01;
    appData[2] = 0x02;
    appData[3] = 0x03;
}

// Add your initialization code here
void setup()
{
  Serial.begin(115200);
  while (!Serial);
}

// The loop function is called in an endless loop
void loop()
{
  switch( deviceState )
  {
    case DEVICE_STATE_INIT:
    {
#if(LORAWAN_DEVEUI_AUTO)
                        LoRaWAN.generateDeveuiByChipID();
#endif
      LoRaWAN.init(loraWanClass,loraWanRegion);
      break;
    }
    case DEVICE_STATE_JOIN:
    {
      LoRaWAN.join();
      break;
    }
    case DEVICE_STATE_SEND:
    {
      prepareTxFrame( appPort );
      LoRaWAN.send(loraWanClass);
      deviceState = DEVICE_STATE_CYCLE;
      break;
    }
    case DEVICE_STATE_CYCLE:
    {
      // Schedule next packet transmission
      txDutyCycleTime = appTxDutyCycle + randr( -APP_TX_DUTYCYCLE_RND,
APP_TX_DUTYCYCLE_RND );
      LoRaWAN.cycle(txDutyCycleTime);
      deviceState = DEVICE_STATE_SLEEP;
      break;
    }
    case DEVICE_STATE_SLEEP:
    {
      LoRaWAN.sleep(loraWanClass,debugLevel);
      break;
    }
    default:
    {
      deviceState = DEVICE_STATE_INIT;
      break;
    }
  }
}
```

HansLoehner commented on Oct 12, 2022

proffalken commented on Oct 16, 2022

Can confirm.

Currently using platformio.org as my IDE/environment and get the following output when running with the ESP32 debugger:

```
Rebooting...
ets Jun  8 2016 00:22:57

rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1184
load:0x40078000,len:13132
load:0x40080400,len:3036
entry 0x400805e4
[   332][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[   332][E][Wire.cpp:526] write(): NULL TX buffer pointer
[   332][E][Wire.cpp:526] write(): NULL TX buffer pointer
[   337][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
[   343][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[   350][E][Wire.cpp:526] write(): NULL TX buffer pointer
[   355][E][Wire.cpp:526] write(): NULL TX buffer pointer
[   360][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
[   366][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[   372][E][Wire.cpp:526] write(): NULL TX buffer pointer
[   377][E][Wire.cpp:526] write(): NULL TX buffer pointer
[   382][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
ESP32 MCU init...

abort() was called at PC 0x4008239a on core 1


Backtrace:0x40083ab5:0x3ffb24200x4008e495:0x3ffb2440 0x400937c5:0x3ffb2460
0x4008239a:0x3ffb24e0 0x40082970:0x3ffb2530 0x400d4c90:0x3ffb2550
0x400d5315:0x3ffb2570 0x400d3a9d:0x3ffb25a0 0x400ddeb6:0x3ffb2820

  #0  0x40083ab5:0x3ffb2420 in panic_abort at /Users/ficeto/Desktop/ESP32/ESP32S2/esp-
idf-public/components/esp_system/panic.c:402
  #1  0x4008e495:0x3ffb2440 in esp_system_abort at
/Users/ficeto/Desktop/ESP32/ESP32S2/esp-idf-
public/components/esp_system/esp_system.c:128
  #2  0x400937c5:0x3ffb2460 in abort at /Users/ficeto/Desktop/ESP32/ESP32S2/esp-idf-
public/components/newlib/abort.c:46
  #3  0x4008239a:0x3ffb24e0 in esp_flash_erase_region at
/Users/ficeto/Desktop/ESP32/ESP32S2/esp-idf-
public/components/spi_flash/esp_flash_api.c:541 (discriminator 5)
  #4  0x40082970:0x3ffb2530 in spi_flash_erase_range at
```

```
D:\/D:\\61722\\Documents\\Arduino\\libraries\\ESP32_LoRaWAN_VASR\\src\\Mcu.cpp:53
  #6  0x400d5315:0x3ffb2570 in McuClass::init(int, int, int, int, unsigned int*) at
D:\/D:\\61722\\Documents\\Arduino\\libraries\\ESP32_LoRaWAN_VASR\\src\\Mcu.cpp:154
  #7  0x400d3a9d:0x3ffb25a0 in setup() at src/main.cpp:74
  #8  0x400ddeb6:0x3ffb2820 in loopTask(void*) at
/home/proffalken/.platformio/packages/framework-
arduinoespressif32/cores/esp32/main.cpp:42
```

line 74 of `src/main.cpp` is copied directly from the OTAA Example and `license` is set as shown in that document as well.

proffalken commented on Oct 16, 2022

OK, I'm now running the latest version of the library thanks to this post on the platformio forums

You need to update your `libdeps` in your platformio.ini file to reference the HTTP Clone URL rather than the PlatformIO registry version:

```
lib_deps =
        bblanchon/ArduinoJson@^6.19.4
        heltecautomation/Heltec ESP32 Dev-Boards@^1.1.1
    https://github.com/HelTecAutomation/ESP32_LoRaWAN.git
```

You then need to edit all of the files in `.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/region/` as documented at #87.

I'm still getting the same error, but at least I know I'm using the latest version of the library...

jcwren commented on Oct 16, 2022 • edited

OK, it's working here with your changes. You can get around the license server BS by editing `.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/Mcu.s` and adding the `retw.n` just after the `call8 getLicenseServer`. The reality is that nothing actually checks the license server other than the `Mcu.init()` function, which only writes it.

Oh, and there's no reason to edit all the region files. Just add `-D LORAWAN_PREAMBLE_LENGTH=8` to your build flags in `platformio.ini` and all LoRaWAN regions use 8.

```
_Z12writelicensev:
.LFB2440:
        .loc 1 31 0
        entry   sp, 32
.LCFI3:
        .loc 1 32 0
        call8   getLicenseAddress
        retw.n                                     # <--- Add this line
.LVL3:
        .loc 1 33 0
        l32r    a11, .LC4
        .loc 1 32 0
        mov.n   a2, a10
.LVL4:
        .loc 1 33 0
        call8   spi_flash_erase_range
.LVL5:
        .loc 1 34 0
        l32r    a11, .LC5
        movi.n  a12, 0x10
        mov.n   a10, a2
        call8   spi_flash_write
.LVL6:
        retw.n
.LFE2440:
        .size   _Z12writelicensev, .-_Z12writelicensev
```

🚀 1

jcwren mentioned this issue on Oct 16, 2022

LORAWAN_PREAMBLE_LENGTH doesn't found #87

Open

proffalken commented on Oct 16, 2022

@jcwren perfect, thank you!

Now I just need to get my keys translated from an ArduinoJSON Dict to a `uint8_t` and I'll be set!

👍 1

proffalken commented on Oct 20, 2022

```
[  5681][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[  5685][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5690][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5695][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
[  5701][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[  5707][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5712][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5717][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
[  5723][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[  5730][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5735][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5740][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
ESP32 MCU init...
ESP32 MCU inited OK!
HEX: CC51E381ABC46680
DevEui=AA51E381ABC46680
LoRaWAN EU868 Class A start!

joining...
LORAMAC_STATUS_OK: 0
Guru Meditation Error: Core  1 panic'ed (StoreProhibited). Exception was unhandled.

Core  1 register dump:
PC      : 0x400d7dd3  PS      : 0x00060f30  A0      : 0x800d7b7b  A1      : 0x3ffb2760
A2      : 0x3ffb2794  A3      : 0x3ffb278c  A4      : 0x00000031  A5      : 0x00000000
A6      : 0x3ffb8a28  A7      : 0x00000000  A8      : 0x00000000  A9      : 0x00000001
A10     : 0x00000000  A11     : 0x3ffc5e00  A12     : 0x3ffc343c  A13     : 0x00000000
A14     : 0x00ff0000  A15     : 0xff000000  SAR     : 0x0000000e  EXCCAUSE: 0x0000001d
EXCVADDR: 0x00000000  LBEG    : 0x4008af29  LEND    : 0x4008af39  LCOUNT  : 0xfffffffb


Backtrace:0x400d7dd0:0x3ffb27600x400d7b78:0x3ffb2780 0x400d5275:0x3ffb27c0
0x400d2ff7:0x3ffb2800 0x400e311d:0x3ffb2820

  #0  0x400d7dd0:0x3ffb2760 in LoRaMacConfirmQueueAdd at
.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/LoRaMacConfirmQueue.c:134
  #1  0x400d7b78:0x3ffb2780 in LoRaMacMlmeRequest at
.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/LoRaMac.c:3181
  #2  0x400d5275:0x3ffb27c0 in LoRaWanClass::join() at
.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/ESP32_LoRaWAN.cpp:484
  #3  0x400d2ff7:0x3ffb2800 in loop() at src/main.cpp:289
  #4  0x400e311d:0x3ffb2820 in loopTask(void*) at
/home/proffalken/.platformio/packages/framework-
arduinoespressif32/cores/esp32/main.cpp:50




ELF file SHA256: 0000000000000000
```

This happens regardless of how I define the TTN DevEui/AppEui/AppKey variables, so I'm not too sure what's going on :(

jcwren commented on Oct 21, 2022

@proffalken, no issues at all right now, it's working fine both in Class A and Class C modes. My sign-on message is a little different, unless your code is printing the `HEX: CC51E381ABC46680`. If you're not printing it, perhaps we're using different versions of the library? Here's my `libdeps` line:

```
lib_deps =
  bblanchon/ArduinoJson@^6.19.4
  heltecautomation/Heltec ESP32 Dev-Boards@^1.1.1
  https://github.com/HelTecAutomation/ESP32_LoRaWAN.git
```

If it's useful to you, I can probably carve everything not useful out of my app and reduce it down to some demo code like the OTAA.ino (although that's basically what I started from).

proffalken commented on Oct 22, 2022

Ok, thanks.

My setup is fairly unique in that I'm getting the DevEUI etc from a json-based webservice and then trying to convert it from a JsonArray to the hex values expected by the library, so if yours is working fine then it's probably that conversion that's causing the issue.

The `HEX` bit is some debugging statements I've added to the library, so I'll reset the library again, remove the licensing stuff, and then try again.

Why it's so difficult to convert a JSON array like `{"0xC0", 0xB1, ... }` to the `uint8_t DevEui {0xC0, 0xB1, ...}` required by the library is absolutely beyond me! 😅

proffalken commented on Nov 4, 2022

Still no joy with this.

My current output is as follows (the Dev/App Eui and key have already been cycled since this output, but they matched the values in the TTN console at the time!)

```
DevEui JSON: CC50E381ABC41242
AppEui JSON: CC50E381ABC41242
AppKey JSON: EFE07F2B338DC55D60CA984E408DE5C3
```

```
to 16
hexToBin called with string: EFE07F2B338DC55D60CA984E408DE5C3. Length of string is 32
and size was set to 32

DevEui=CC50E381ABC41242
AppEui=CC50E381ABC41242
AppKey=EFE07F2B338DC55D60CA984E408DE5C3[  5802][E][Wire.cpp:422] beginTransmission():
could not acquire lock
[  5802][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5806][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5811][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
[  5817][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[  5823][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5828][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5833][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
[  5839][E][Wire.cpp:422] beginTransmission(): could not acquire lock
[  5845][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5850][E][Wire.cpp:526] write(): NULL TX buffer pointer
[  5855][E][Wire.cpp:448] endTransmission(): NULL TX buffer pointer
ESP32 MCU init...
ESP32 MCU inited OK!
LoRaWAN Initialising

DevEui=CC50E381ABC41242
LoRaWAN EU868 Class A start!

Attempting Join
joining...
Guru Meditation Error: Core  1 panic'ed (StoreProhibited). Exception was unhandled.

Core  1 register dump:
PC      : 0x400d803f  PS      : 0x00060f30  A0      : 0x800d7de7  A1      : 0x3ffb2760
A2      : 0x3ffb2794  A3      : 0x3ffb278c  A4      : 0x00000031  A5      : 0x3ffb26a0
A6      : 0x00000003  A7      : 0x00060023  A8      : 0x00000000  A9      : 0x00000001
A10     : 0x00000000  A11     : 0x3ffc5df8  A12     : 0x3ffc3434  A13     : 0x00000000
A14     : 0x00000001  A15     : 0x3ffb08c0  SAR     : 0x0000000e  EXCCAUSE: 0x0000001d
EXCVADDR: 0x00000000  LBEG    : 0x4008af45  LEND    : 0x4008af55  LCOUNT  : 0xfffffffd


Backtrace:0x400d803c:0x3ffb27600x400d7de4:0x3ffb2780 0x400d54e7:0x3ffb27c0
0x400d317d:0x3ffb2800 0x400e33c9:0x3ffb2820

  #0  0x400d803c:0x3ffb2760 in LoRaMacConfirmQueueAdd at
.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/LoRaMacConfirmQueue.c:134
  #1  0x400d7de4:0x3ffb2780 in LoRaMacMlmeRequest at
.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/LoRaMac.c:3181
  #2  0x400d54e7:0x3ffb27c0 in LoRaWanClass::join() at
.pio/libdeps/heltec_wifi_lora_32_V2/ESP32_LoRaWAN/src/ESP32_LoRaWAN.cpp:484
  #3  0x400d317d:0x3ffb2800 in loop() at src/main.cpp:401
  #4  0x400e33c9:0x3ffb2820 in loopTask(void*) at
/home/mmw/.platformio/packages/framework-arduinoespressif32/cores/esp32/main.cpp:50
```

I've added in some extra debugging so I can check the length of the bytes etc, but it seems to be when I call `LoRaMacConfirmQueueAdd` that the problems start, I just can't work out why!

jcwren commented on Nov 4, 2022 • edited

That's a pretty low level function for user code to be calling. I'd need to see the code that you're setting up to make the call, etc. And what you're trying to accomplish with it :)

proffalken commented on Nov 5, 2022 • edited

That's a fair shout.

I've stripped out some of the cruft I've put in for debugging (it fails with the same error though, so that's all good I guess?! :D) and uploaded both my code and the relevant sections of the LoRaWAN libraries to https://gist.github.com/proffalken/d7ef6f569cbbfaa2374e7aa6f20ae6a9

In essence, I have a webservice that I have written that the ESP32 device calls via WiFi on startup.

The web service talks to the TTN API and returns the provisioning data in the following format:

```
{
  "dev_eui": "CC50E381ABC41242",
  "app_key": "EFE07F2B338DC55D60CA984E408DE5C3",
  "app_eui": "CC50E381ABC41242"
}
```

I then convert these JSON Strings into their HEX equivalents such as `{0xCC, 0x50, ...}` and pass them to the appropriate variables so that the LoRaWAN library can use them.

I think i worded things badly in my last post - I'm calling `.join()` , and it's the library that filters that call down to `LoRaMacConfirmQueueAdd` rather than my code directly. Perhaps a case of me giving too much detail! :)

I'm now at a point where the init() itself works, but the join fails, so I'm happy to split off into another ticket if you'd prefer?

jcwren commented on Nov 5, 2022

I'm interested in helping you getting this working, but I don't think this is the appropriate forum for a bunch of back and forth. Drop me an email at jcwren@gmail.com and we'll see what we can do.

👍 1

proffalken commented on Nov 5, 2022 • edited

Thanks, appreciate it, I'll do that shortly.

~~Fwiw, if I strip everything back and hardcode the values, it works, so it's definitely something to do with the translation from JSON strings to the byte array needed for the uint8_t~~

Actually, it doesn't.

I'm now using the code from the `examples` directory and the keys from TTN and it's failing on the join.

TasyDevilsky mentioned this issue on Dec 7, 2022

Mcu.init() problem solved and LORAWAN_PREAMBLE_LENGTH #95

Closed

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants