

Issues and Tips

Solution for .text too big issues

Adding CFLAGS in Makefile

```
-ffunction-sections  
-fdata-sections  
CFLAGS = -I. -mlongcalls -ffunction-sections
```

Adding LDFLAGS IN Makefile

```
-Wl,--gc-sections -Wl  
LDFLAGS = -DICACHE_FLASH -Tlinker.ld -Wl,--gc-sections -Wl,-Map=output.map -  
Wl,--cref
```

See more at:

- <http://www.esp8266.com/viewtopic.php?f=9&t=9012#sthash.lbNSPHkP.dpuf>
- <http://www.esp8266.com/viewtopic.php?f=9&t=9012>
- <http://www.esp8266.com/viewtopic.php?f=9&t=3796&start=4>
- <http://kacangbawang.com/esp8266-512k-flash-compiling-using-rtos-sdk-1-3/>
- <http://wiki.serafim.eti.br/ht-txt:esp-too-big-irom0>

MODIFY LOADER SCRIPT

Force some libraries to be loaded in the irom section (flash), saving ram. This can be achieved by adding the following line in the .irom.text section:

- liblwip.a:(.literal .text .literal.* .text.*)

```
.irom0.text : ALIGN(4)  
{  
    _irom0_text_start = ABSOLUTE(.);  
  
    *libmbedtls.a:(.literal .text .literal.* .text.*)  
    *liblwip.a:(.literal .text .literal.* .text.*)
```

STRIPPING libgcc.a

Next to the SDK, you need a crosstool (crosstool-ng). This crosstool will build the latest libgcc for the compiler you choose. The SDK includes libgcc.a as well and this conflicts with the toolchain version.

The only difference between the SDK libgcc.a and the default one built by the toolchain is the removal of several low level math routines that exist in the ROM. The SDK /lib directory contains its own libgcc.a, which is >200k smaller in size than the one built and installed as part of crosstool-ng. Its position in the toolchain either when copied or when used in a separate SDK means that the crosstool built libgcc is not used

I would recommend that libgcc.a be removed from the SDK, or at least placed somewhere other than with the other libraries. I would also recommend that a script be created that will strip from the library any functions that exist in ROM. This would look something like:

```
export PATH=$(TOOLCHAIN)/bin:${PATH}
cd $(TOOLCHAIN)/lib/gcc/xtensa-lx106-elf/<version>
xtensa-lx106-elf-ar -M < strip_libgcc_funcs.txt
```

This script for ar will remove the functions from libgcc.a which are in the esp8266 ROM. This then means that applications linked with the crosstool-NG built libgcc will now fit.

```
-----strip_libgcc_funcs.txt-----
OPEN libgcc.a
DELETE _divsi3.o
DELETE _udivsi3.o
DELETE _umodsi3.o
DELETE _umulsidi3.o
DELETE _addsubsf3.o
DELETE _fixunssfsi.o
DELETE _floatsisf.o
DELETE _floatunsisf.o
DELETE _sqrtf.o
DELETE _recipsf2.o
DELETE _rsqrtsf2.o
DELETE _addsubdf3.o
DELETE _muldf3.o
DELETE _divdf3.o
DELETE _fixdfsi.o
DELETE _fixunsdfsi.o
DELETE _floatsidf.o
DELETE _floatunsidf.o
DELETE _truncdfsf2.o
DELETE _extendsfdf2.o
DELETE _sqrt.o
DELETE _recipdf2.o
DELETE _rsqrtdf2.o
DELETE _muldi3.o
DELETE _divdi3.o
DELETE _udivdi3.o
DELETE _umoddi3.o
SAVE
EXIT
-----strip_libgcc_funcs.txt-----
```

From:

<https://wiki.oscardegroot.nl/> - **HomeWiki**

Permanent link:

<https://wiki.oscardegroot.nl/doku.php?id=esp:esp8266:open-sdk:tips>

Last update: **2022/01/15 11:38**

