r4-freertos build出来的elf需要嵌入lpp.bin，可以参考Linux lernel中的实现。

in arch/arm/boot/compressed directory

```
$ cat piggy.gzip.S
    .section .piggydata,#alloc
    .globl  input_data
input_data:
    .incbin "arch/arm/boot/compressed/piggy.gzip"
    .globl  input_data_end
input_data_end:
```

嵌入的piggy.gzip　blob位于 `.piggydata` section，并且可以使用
input_data
input_data_end
这两个变量来访问。

```
$ arm-linux-gnueabi-nm arch/arm/boot/compressed/vmlinux | grep input_data
00004618 R input_data
0030bed9 R input_data_end
```

```
$ arm-linux-gnueabi-readelf -S arch/arm/boot/compressed/vmlinux
There are 22 section headers, starting at offset 0x31a514:

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .text             PROGBITS        00000000 008000 003954 00  AX  0   0 32
  [ 2] .rodata           PROGBITS        00003954 00b954 000cc4 00   A  0   0  4
  [ 3] .piggydata        PROGBITS        00004618 00c618 3078c1 00   A  0   0  1
  [ 4] .got.plt          PROGBITS        0030bedc 313edc 00000c 04  WA  0   0  4
  [ 5] .got              PROGBITS        0030bee8 313ee8 000028 00  WA  0   0  4
  [ 6] .pad              PROGBITS        0030bf10 313f10 000008 00  WA  0   0  1
  [ 7] .bss              NOBITS          0030bf18 313f18 00001c 00  WA  0   0  4
  [ 8] .stack            NOBITS          0030bf38 313f18 001000 00  WA  0   0  1
  [ 9] .comment          PROGBITS        00000000 313f18 000011 01  MS  0   0  1
  [10] .ARM.attributes   ARM_ATTRIBUTES  00000000 313f29 00002d 00      0   0  1
  [11] .debug_line       PROGBITS        00000000 313f56 0016af 00      0   0  1
  [12] .debug_info       PROGBITS        00000000 315605 002590 00      0   0  1
  [13] .debug_abbrev     PROGBITS        00000000 317b95 0006eb 00      0   0  1
  [14] .debug_aranges    PROGBITS        00000000 318280 000128 00      0   0  8
  [15] .debug_ranges     PROGBITS        00000000 3183a8 000150 00      0   0  8
  [16] .debug_frame      PROGBITS        00000000 3184f8 0003c4 00      0   0  4
  [17] .debug_loc        PROGBITS        00000000 3188bc 000fd3 00      0   0  1
  [18] .debug_str        PROGBITS        00000000 31988f 000bb1 01  MS  0   0  1
  [19] .shstrtab         STRTAB          00000000 31a440 0000d2 00      0   0  1
  [20] .symtab           SYMTAB          00000000 31a884 001000 10     21 193  4
  [21] .strtab           STRTAB          00000000 31b884 000819 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
```

```
  O (extra OS processing required) o (OS specific), p (processor specific)
```

> [3] .piggydata PROGBITS 00004618 00c618 3078c1 00 A 0 0 1

在c中访问blob可以做如下declaration

```
extern char *input_data;
extern char *input_data_end;
```

Sample

```
$ cat bin.s
    .section .bindata

    .globl bin_data_start
    .globl bin_data_end

bin_data_start:
    .incbin "tx.bin"
bin_data_end:

$ cat test.c
#include <stdio.h>

extern char *bin_data_start;
extern char *bin_data_end;


int main()
{
    int length = bin_data_end - bin_data_start;
    printf("binsize = %d\n", length);

    return 0;
}
```

```
$ as -o bin.o bin.s
$ gcc -c -o test.o test.c
$ gcc -o test test.o bin.o
```

```
$ readelf -S test | grep "\.bindata"
  [27] .bindata          PROGBITS         0000000000000000  0000106b

$ nm test | grep bin_data
00000000000e0e58 N bin_data_end
0000000000000000 N bin_data_start
```