

Lab 5.1 – Skapa ett run-time bibliotek för MD407

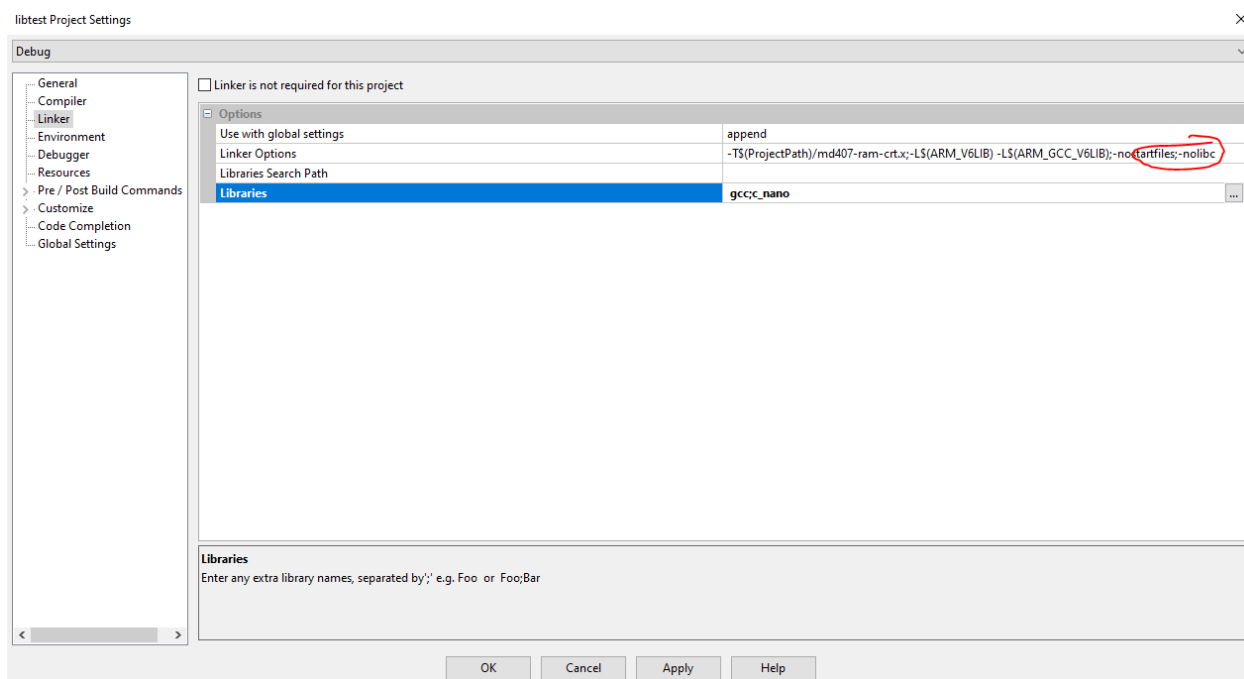
I den här labben skall ni skapa ett runtime bibliotek för MD407, som låter oss använda standard funktioner som `printf()` och `scanf()`, och låter oss hämta input från keypaden och skriva ut till ASCII displayen. Programmet kommer använda det färdigskrivna standard c biblioteket för alla platforms oberoende funktioner, men för att få plats på datorns begränsade minne använder vi en extremt liten variant av biblioteket: `libc_nano`.

Hur ni bär er åt för att komma igång beskrivs i detalj i Kapitel 8 i arbetsboken, men nedan följer en kort overview av stegen.

Skapa ett testprogram

Ni börjar med att starta ett testprogram, som skall användas för att testa erat library. Skapa en ny workspace i CodeLite, och sedan ett nytt projekt som använder templatén `startup_crt`. Detta projekt innehåller redan den kod som behövs för att komma igång med `libc_nano`, men inga platformsberoende implementationer.

Viktigt: I templatén är det redan ordnat så att programmet länkas utan standard startup koden och istället använder den kod som är implementerad i c filen. Dock behöver vi *också* säga till länkaren att inte använda standard versionen av `libc`. Läggs därför till flaggan `-no_libc` i projektets länkarsettings.



Skapa ett CRT bibliotek

Nu skapar ni ett nytt projekt för CRT biblioteket. Se till att använda templatén `static library`. Följ anvisningarna i föreläsningen eller arbetsboken. Utgå ifrån filerna som följer i det här dokumentet. Notera att mycket i `general.c` bara flyttats from `main` filen.

```

1.  /*
2.  * libMD407.h
3.  * Declaration of library functions, constants etc...
4.  */
5.  #include    <stdio.h>
6.  //#include  <errno.h>
7.  #include    <sys/stat.h>
8.  /* Type definitions */
9.  #include <sys/stat.h>
10.
11. typedef struct
12. {
13.     char name[16];
14.     int (*init) (int);
15.     void (*deinit) (int);
16.     int (*fstat)(struct stat *st);
17.     int (*isatty)(void);
18.     int (*open)(const char name,int flags,int mode);
19.     int (*close)(void);
20.     int (*lseek)(int ptr, int dir);
21.     int (*write)(char *ptr, int len);
22.     int (*read)(char *ptr, int len);
23. } DEV_DRIVER_DESC, *PDEV_DRIVER_DESC;
24.
25. /* File handle constants */
26. enum {STDIN=0,STDOUT,STDERR,KEYPAD,ASCIIDISPLAY};
27. #define MAX_FILENO  ASCIIDISPLAY
28.
29. /* Library defined functions */
30. int _isatty(int file);

```

```

1.  /*
2.  *  libMD407
3.  *  general.c
4.  *  includes: _crt_init, _crt_deinit, _sbrk for malloc-support.
5.  *  template functions for stdio:
6.  */
7.  #include    "libMD407.h"
8.
9.  extern DEV_DRIVER_DESC StdIn,StdOut,StdErr,KeyPad,AsciiDisplay;
10.
11. PDEV_DRIVER_DESC device_table[MAX_FILENO+1] =
12. {
13.     &StdIn,
14.     &StdOut,
15.     &StdErr,
16.     &KeyPad,
17.     &AsciiDisplay,
18. };
19.
20. static char *heap_end;
21. extern char __heap_low; /* Defined by the linker */
22. extern char __heap_top; /* Defined by the linker */
23.
24. char * _sbrk(int incr) {
25.     char *prev_heap_end;
26.     if (heap_end == 0) {
27.         heap_end = &__heap_low;
28.     }
29.     prev_heap_end = heap_end;
30.     if (heap_end + incr > &__heap_top) {
31.         return (char *)-1;
32.     }
33.     heap_end += incr;
34.     return (char *) prev_heap_end;
35. }
36.
37. __attribute__ ( (used) )          /* Will be optimised away otherwise... */
38. volatile void _crt_init() {
39.     extern char __bss_start__; /* Defined by the linker */
40.     extern char __bss_end__;   /* Defined by the linker */
41.     char *s;
42.     PDEV_DRIVER_DESC fd;
43.
44.     heap_end = 0;
45.
46.     // NOTE: While testing, you may want to comment out the following four
47.     //       lines as they take a lot of time to run in the simulator.
48.     s = &__bss_start__;
49.     while( s < &__bss_end__ ) *s++ = 0;
50.     s = &__heap_low;
51.     while( s < &__heap_top ) *s++ = 0;
52.
53.     for( int i = 0; i <= MAX_FILENO; i++ )
54.     {
55.         fd = device_table[i];
56.         if( fd && fd->init != 0)
57.             (void) fd->init( 0 );
58.     }
59. }
60.
61.

```

```

62. __attribute__ ( (used) )          /* Will be optimised away otherwise... */
63. volatile void _crt_deinit() {
64.     PDEV_DRIVER_DESC fd;
65.     fflush(0); /* Will cause terminal flush... */
66.     for( int i = 0; i <= MAX_FILENO; i++ )
67.     {
68.         fd = device_table[i];
69.         if( fd && fd->deinit != 0 )
70.             fd->deinit( 0 );
71.     }
72. }
73.
74.
75. /* STDIO-templates */
76. int _close(int file) { return -1; }
77. int _open(const char *name, int flags, int mode) { return -1; }
78. int _fstat(int file, struct stat *st) { st->st_mode = S_IFCHR; return 0; }
79. int _lseek(int file, int ptr, int dir) { return 0; }
80. int _isatty(int file) {
81.     switch (file)
82.     {
83.         case STDIN: case STDOUT: case STDERR: return 1;
84.         default: return 0;
85.     }
86. }
87.
88. int _write(int file, char *ptr, int len) {
89.     // Your code here
90. }
91.
92. int _read(int file, char *ptr, int len) {
93.     // Your code here
94. }

```

```

1.  /*
2.  *  libMD407
3.  *  usart_driver.c
4.  *  Read buffered USART driver
5.  *  Drivers for STDIN, STDOUT and STDERR
6.  */
7.  #include    "libMD407.h"
8.
9.  /* Define functions here */
10. static int usart_init( int initval );
11. static void usart_deinit( int deinitval);
12. static int usart_write(char *ptr, int len);
13. static int usart_read(char *ptr, int len);
14.
15. DEV_DRIVER_DESC StdIn =
16. {
17.     {"stdin"},
18.     usart_init,
19.     usart_deinit,
20.     0,
21.     _isatty,
22.     0,
23.     0,
24.     0,
25.     0,
26.     usart_read
27. };
28. DEV_DRIVER_DESC StdOut =
29. {
30.     {"stdout"},
31.     usart_init,
32.     usart_deinit,
33.     0,
34.     _isatty,
35.     0,
36.     0,
37.     0,
38.     usart_write,
39.     0
40. };
41. DEV_DRIVER_DESC StdErr =
42. {
43.     {"stderr"},
44.     usart_init,
45.     usart_deinit,
46.     0,
47.     _isatty,
48.     0,
49.     0,
50.     0,
51.     usart_write,
52.     0
53. };
54.
55. /* USART types and constants definitions */
56.
57. /* USART implementation */
58. static int usart_init( int initval ) { ... }
59. static void usart_deinit( int deinitval){ ... }
60. static int usart_write(char *ptr, int len){ ... }
61. static int usart_read(char *ptr, int len){ ... }

```

```
1.  /*
2.  *  libMD407
3.  *  keypad_driver.c
4.  *  Keypad connected to PD8-PD15
5.  *  Driver for KEYPAD
6.  */
7.  #include    "libMD407.h"
8.
9.  static int keypad_init( int initval );
10. static void keypad_deinit( int deinitval);
11. static int keypad_read(char *ptr, int len);
12.
13. DEV_DRIVER_DESC KeyPad =
14. {
15.     {"Keypad"},
16.     keypad_init,
17.     keypad_deinit,
18.     0,
19.     0,
20.     0,
21.     0,
22.     0,
23.     0,
24.     keypad_read
25. };
26.
27. /* KEYPAD types and constants definitions */
28. ...
29. /* Define functions here */
30. static int keypad_init( int initval ){ ... }
31. static void keypad_deinit( int deinitval){ ... }
32. static int keypad_read(char *ptr, int len){ ... }
```

```

1.  /*
2.  *  libMD407
3.  *  asciidisplay_driver.c
4.  *  Display connected to PE
5.  *  Driver for ASCIIDISPLAY
6.  */
7.  #include    "libMD407.h"
8.
9.  static int asciidisplay_init( int initval );
10. static void asciidisplay_deinit( int deinitval);
11. static int asciidisplay_write(char *ptr, int len);
12.
13. DEV_DRIVER_DESC AsciiDisplay =
14. {
15.     {"AsciiDisplay"},
16.     asciidisplay_init,
17.     asciidisplay_deinit,
18.     0,
19.     0,
20.     0,
21.     0,
22.     0,
23.     asciidisplay_write,
24.     0
25. };
26.
27. /* ASCIIDISPLAY types and constants definitions */
28. // ...
29. /* Define functions here */
30. static int asciidisplay_init( int initval ){ ... }
31. static void asciidisplay_deinit( int deinitval){ ... }
32. static int asciidisplay_write(char *ptr, int len){ ... }

```