

```

1 //=====
2 //
3 // Step 12
4 // TimeOut
5 //
6 //=====
7 #include "mbed.h"
8 #include "TextLCD.h"
9
10 TextLCD lcd(p17, p12, p27, p28, p29, p30); // rs, e, d4-d7
11 //Ticker tik; // recurring interrupt
12 Timeout t_out;
13 int t_cnt = 0;
14 int timFlg = 0;
15
16 //-----
17 // Time Out
18 //-----
19 void attimeout(void)
20 {
21     timFlg = 1;
22 }
23
24 //-----
25 // Main
26 //-----
27 int main(void)
28 {
29     lcd.cls();
30     lcd.locate(0, 0); // x, y
31
32     //-----
33     // RTC check
34     //-----
35     time_t now_time = time(NULL); // today
36     struct tm *s_tm = localtime(&now_time);
37     if(s_tm->tm_year < 119) // since 1900
38     { // RTC set
39         struct tm t;
40         t.tm_sec = 0; // 0-59
41         t.tm_min = 0; // 0-59
42         t.tm_hour = 10; // 0-23
43         t.tm_mday = 9; // 1-31
44         t.tm_mon = 7-1; // 0-11
45         t.tm_year = 119; // year since 1900
46
47         set_time(mktime(&t)); // Write RTC
48     }
49
50     t_out.attach(&attimeout, 1); // 1s -> call attimeout
51
52     while(1)
53     {
54         if( timFlg == 1 )
55         {
56             timFlg = 0;
57             t_out.attach(&attimeout, 1); // 1s -> call attimeout
58             now_time = time(NULL); // today
59             s_tm = localtime(&now_time);
60             lcd.locate(0, 0);
61             lcd.printf("%02d/%02d %02d:%02d:%02d", s_tm->tm_mon+1, s_tm->tm_mday, s_tm-
62 >tm_hour, s_tm->tm_min, s_tm->tm_sec);
63         }
64     }
65     return 0;

```

