

```

1 //=====
2 // Step 30
3 // Serial tx only
4 //=====
5 #include "mbed.h"
6 #include "TextLCD.h"
7
8 TextLCD lcd(p17, p12, p27, p28, p29, p30);
9 DigitalOut led1(LED1);
10 Serial pc(USBTX, USBRX);          // tx, rx
11
12 Ticker tik;                       // recurring interrupt
13
14 int tikFlg = 0;
15 int tCnt = 0;
16
17 //=====
18 // Ticker
19 //=====
20 void attime(void)
21 {
22     tikFlg = 1;
23     tCnt++;
24 }
25
26 //=====
27 // Main
28 //=====
29 int main(void)
30 {
31     char sdt[100];
32
33     lcd.cls();
34     lcd.locate(0, 0);              // x, y
35
36     //=====
37     // RTC check
38     //=====
39     time_t now_time = time(NULL); // today
40     struct tm *s_tm = localtime(&now_time);
41     if(s_tm->tm_year < 118 )       // since 1900
42     { // RTC set
43         struct tm t;
44         t.tm_sec = 0;              // 0-59
45         t.tm_min = 0;              // 0-59
46         t.tm_hour = 10;            // 0-23
47         t.tm_mday = 17;            // 1-31
48         t.tm_mon = 7-1;            // 0-11
49         t.tm_year = 118;           // year since 1900
50
51         set_time(mktime(&t));      // Write RTC
52     }
53
54     tik.attach(&attime, 1);        // 1s -> call attime
55
56     while(1)
57     {
58         if( tikFlg == 1 )
59         {
60             tikFlg = 0;
61             now_time = time(NULL); // today
62             s_tm = localtime(&now_time);
63             lcd.locate(0, 0);
64             sprintf(sdt, "%02d/%02d %02d:%02d:%02d", s_tm->tm_mon+1, s_tm->tm_mday, s_tm
->tm_hour, s_tm->tm_min, s_tm->tm_sec);
65             lcd.printf(sdt);

```

```
66     }
67     if( tCnt > 4 )
68     {
69         tCnt = 0;
70         pc.printf( "%04d/%s¥r¥n", s_tm->tm_year+1900, sdt );
71     }
72 }
73
74 return 0;
75 }
```