

```

1 //=====
2 //
3 // Step 31-1
4 // Serial rx interrupt
5 // Command rx
6 //=====
7 #include "mbed.h"
8 #include "TextLCD.h"
9
10 TextLCD lcd(p17, p12, p27, p28, p29, p30); // rs, e, d4-d7
11 DigitalOut leds[] = { LED4, LED3, LED2, LED1, p19, p20, p21, p22, LED4, p24, p25, p26 }
12 ;
13 Serial pc(USBTX, USBRX); // tx, rx
14 PwmOut beep(p23); // JP7 (2-3)
15
16 Ticker tik; // recurring interrupt
17
18 int tikFlg = 0;
19 int tCnt = 0;
20 int ptr = 0;
21 int rx_flg = 0;
22 char rx_buf[100];
23 float freq = 1000;
24
25 //-----
26 // Ticker
27 //-----
28 void attime(void)
29 {
30     tikFlg = 1;
31     tCnt++;
32 }
33
34 //-----
35 // Rx Interrupt
36 //-----
37 void rx_int()
38 {
39     rx_buf[ptr] = pc.getc();
40     if( rx_buf[ptr] == 0x0d )
41     {
42         rx_buf[ptr]=0; // 文字列の最後を表すNULLをセット
43         ptr = 0;
44         rx_flg = 1; // コマンド受信フラグをセット
45     }
46     else
47         ptr++;
48 }
49
50 //-----
51 // Main
52 //-----
53 int main(void)
54 {
55     char sdt[100];
56
57     lcd.cls();
58     lcd.locate(0, 0); // x, y
59
60     //-----
61     // RTC check
62     //-----
63     time_t now_time = time(NULL); // today
64     struct tm *s_tm = localtime(&now_time);
65     if(s_tm->tm_year < 117) // since 1900
66     { // RTC set
67         struct tm t;

```

```

66     t.tm_sec = 0;           // 0->59
67     t.tm_min = 0;           // 0->59
68     t.tm_hour = 10;          // 0->23
69     t.tm_mday = 14;           // 1->31
70     t.tm_mon = 7-1;           // 0->11
71     t.tm_year = 118;          // year since 1900
72
73     set_time(mktime(&t));    // Write RTC
74 }
75
76 tik.attach(&attime,1);       // 1s -> call attime
77 pc.attach(&rx_int);
78
79 while(1)
80 {
81     //-----
82     if( rx_flg == 1 )
83     {
84         rx_flg = 0;
85         lcd.locate(0, 1);
86         //sprintf(sdt, "%s", &rx_buf[0]);
87         lcd.printf(rx_buf);
88
89         if( rx_buf[0] == 'L' && rx_buf[1] == 'E' && rx_buf[2] == 'D' )
90         { // LED, nn, x
91             int n = (rx_buf[4] & 0x0F)*10 + (rx_buf[5] & 0x0F);
92             if( n >= 0 && n <= 11 )
93             {
94                 leds[ n ] = rx_buf[7] & 0x0f;
95             }
96         }
97         else if( rx_buf[0] == 'B' && rx_buf[1] == 'U' && rx_buf[2] == 'Z' )
98         { // BUZ, nnnn, x
99             if( rx_buf[9] == '1' )
100             { // ----- beginneer
101                 //freq = (rx_buf[4] & 0x0F)*1000 + (rx_buf[5] & 0x0F)*100 + (rx_bu
f[6] & 0x0F)*10 + (rx_buf[7] & 0x0F);
102                 // ----- use atoi function
103                 rx_buf[8] = '0'; // NULL
104                 freq = atoi(&rx_buf[4]);
105
106                 beep.period(1.0/freq);
107                 beep.write(0.5); //Duty
108             }
109             else
110             {
111                 beep.write(0.0);
112             }
113         }
114     }
115 }
116
117 //-----
118 if( tikFlg == 1 )
119 {
120     tikFlg = 0;
121     now_time = time(NULL); // today
122     s_tm = localtime(&now_time);
123     lcd.locate(0, 0);
124     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);
125     lcd.printf(sdt);
126 }
127
128 //-----
129

```

```
130 |         if( tCnt > 4 )
131 |         {
132 |             tCnt = 0;
133 |             //led1 = 1;
134 |             //pc.printf( "%04d/%s¥r¥n", s_tm->tm_year+1900, sdt);
135 |             //led1 = 0;
136 |         }
137 |     }
138 |
139 | }
```