**2.1 =====================================**

1. #include "msp430g2553.h"
3. // led 7seg pin
4. #define PDAT P2OUT
5. #define PROW P1OUT
7. //----for key------------
8. #define K1 BIT0
9. #define K2 BIT1
10. #define K3 BIT2
11. #define K4 BIT3
13. // function prototype declaration
14. void delayms(int);
15. void scanled();
16. void toDisplay(int);
18. // variable declaration
19. char code7seg[] = {
20. 0xC0,
21. 0xF9,
22. 0xA4,
23. 0xB0,
24. 0x99,
25. 0x92,
26. 0x82,
27. 0xF8,
28. 0x80,
29. 0x90
30. };
32. char code74HC[] = {
33. 0x00,
34. 0x40,
35. 0x80,
36. 0xC0
37. };
39. char buffer[4] = {
40. 0,
41. 0,
42. 0,
43. 0
44. };
45. int idx = 0;
46. int value = 0;
48. // =============================== main ============================
50. void main(void) {
51. // Stop watchdog timer to prevent time out reset
52. WDTCTL = WDTPW + WDTHOLD;
54. //Set MCLK = SMCLK = 1MHz
55. DCOCTL = 0;
56. BCSCTL1 = CALBC1\_1MHZ;
57. DCOCTL = CALDCO\_1MHZ;
59. //P1SEL = 0x00;
60. P1DIR = 0xF0;
61. P1REN = 0x0F;
62. P1OUT = 0xFF;
64. P2SEL = 0x00;
65. P2DIR = 0xFF;
67. toDisplay(value);
69. for (;;) {
71. if ((P1IN & K1) == 0) {
72. delayms(200);
73. value++;
75. }
77. if ((P1IN & K2) == 0) {
78. delayms(200);
79. value--;
81. }
83. if ((P1IN & K3) == 0) {
84. delayms(100);
85. value = 0;
86. }
88. toDisplay(value);
90. scanled();
91. delayms(5);
92. }
94. }
96. //============================= function ===============================
97. void delayms(int ms) {
98. int i;
99. for (i = 0; i < ms; i++)
100. \_\_delay\_cycles(1000);
101. }
103. void scanled() {
105. PDAT = code7seg[buffer[idx]];
106. PROW |= code74HC[idx];
108. \_\_delay\_cycles(1000);
109. PDAT = 0xFF;
110. PROW &= ~code74HC[idx];
112. idx = idx == 3 ? 0 : idx + 1;
114. }
116. void toDisplay(int value) {
118. buffer[3] = value % 10;
119. value /= 10;
120. buffer[2] = value % 10;
121. value /= 10;
122. buffer[1] = value % 10;
123. value /= 10;
124. buffer[0] = value % 10;
125. }

**2.2 =============================**

1. #include "msp430g2553.h"
3. // led 7seg pin
4. #define PDAT P2OUT
5. #define PROW P1OUT
7. //----for key------------
8. #define K1 BIT0
9. #define K2 BIT1
10. #define K3 BIT2
11. #define K4 BIT3
13. // function prototype declaration
14. void delayms(int);
15. void scanled();
16. void toDisplay(int);
18. // variable declaration
19. char code7seg[] = {
20. 0xC0,
21. 0xF9,
22. 0xA4,
23. 0xB0,
24. 0x99,
25. 0x92,
26. 0x82,
27. 0xF8,
28. 0x80,
29. 0x90
30. };
32. char code74HC[] = {
33. 0x00,
34. 0x40,
35. 0x80,
36. 0xC0
37. };
39. char buffer[4] = {
40. 0,
41. 0,
42. 0,
43. 0
44. };
45. int idx = 0;
46. int value = 0;
47. int interval = 0;
48. int status = 0;
49. // =============================== main ============================
51. void main(void) {
52. // Stop watchdog timer to prevent time out reset
53. WDTCTL = WDTPW + WDTHOLD;
55. //Set MCLK = SMCLK = 1MHz
56. DCOCTL = 0;
57. BCSCTL1 = CALBC1\_1MHZ;
58. DCOCTL = CALDCO\_1MHZ;
60. //P1SEL = 0x00;
61. P1DIR = 0xF0;
62. P1REN = 0x0F;
63. P1OUT = 0xFF;
65. P2SEL = 0x00;
66. P2DIR = 0xFF;
68. toDisplay(value);
70. for (;;) {

73. if ((P1IN & K1) == 0) {
74. delayms(200);
75. status = 0;
76. value++;
78. }
80. if ((P1IN & K2) == 0) {
81. delayms(200);
82. status = 0;
83. value--;
85. }
87. if ((P1IN & K3) == 0){ delayms(500); status = 1;}
88. if ((P1IN & K4) == 0){ delayms(500); status = 2;}
90. if (interval == 200) {
91. interval = 0;
92. switch (status) {
93. case 1: value++; break;
94. case 2: value = value > 0 ? value - 1 : 0; break;
95. }
96. }
98. toDisplay(value);
100. scanled();
101. delayms(5);
102. interval++;
103. }
105. }
107. //============================= function ===============================
108. void delayms(int ms) {
109. int i;
110. for (i = 0; i < ms; i++)
111. \_\_delay\_cycles(1000);
112. }
114. void scanled() {
116. PDAT = code7seg[buffer[idx]];
117. PROW |= code74HC[idx];
119. \_\_delay\_cycles(1000);
120. PDAT = 0xFF;
121. PROW &= ~code74HC[idx];
123. idx = idx == 3 ? 0 : idx + 1;
125. }
127. void toDisplay(int value) {
129. buffer[3] = value % 10;
130. value /= 10;
131. buffer[2] = value % 10;
132. value /= 10;
133. buffer[1] = value % 10;
134. value /= 10;
135. buffer[0] = value % 10;
136. }

**2.3 ==============================================**

1. #include "msp430g2553.h"
3. // led 7seg pin
4. #define PDAT P2OUT
5. #define PROW P1OUT
7. //----for key------------
8. #define K1 BIT0
9. #define K2 BIT1
10. #define K3 BIT2
11. #define K4 BIT3
13. // function prototype declaration
14. void delayms(int);
15. void scanled();
16. void toDisplay(int);
18. // variable declaration
19. char code7seg[] = {
20. 0xC0,
21. 0xF9,
22. 0xA4,
23. 0xB0,
24. 0x99,
25. 0x92,
26. 0x82,
27. 0xF8,
28. 0x80,
29. 0x90
30. };
32. char code74HC[] = {
33. 0x00,
34. 0x40,
35. 0x80,
36. 0xC0
37. };
39. char buffer[4] = {
40. 0,
41. 0,
42. 0,
43. 0
44. };
45. int idx = 0;
46. int value = 0;
48. // =============================== main ============================
50. void main(void) {
51. // Stop watchdog timer to prevent time out reset
52. WDTCTL = WDTPW + WDTHOLD;
54. //Set MCLK = SMCLK = 1MHz
55. DCOCTL = 0;
56. BCSCTL1 = CALBC1\_1MHZ;
57. DCOCTL = CALDCO\_1MHZ;
59. //P1SEL = 0x00;
60. P1DIR = 0xF0;
61. P1REN = 0x0F;
62. P1OUT = 0xFF;
64. P2SEL = 0x00;
65. P2DIR = 0xFF;
67. TA0CTL = TASSEL\_2 + ID\_0 + MC\_1; // SMCLK as clock source, divider 0, up-mode
69. CCTL0 = CCIE; // Enable interrupt for CCR0
70. CCR0 = 5000; // 5ms
71. \_BIS\_SR(GIE); // Enable global interrupts
73. toDisplay(value);
75. for (;;) {

78. if ((P1IN & K1) == 0) {
79. value++;
80. delayms(300);

83. }
85. if ((P1IN & K2) == 0) {
86. value = value > 0 ? value - 1 : 0;
87. delayms(300);

90. }
92. if ((P1IN & K3) == 0) {
93. value = 0;
94. delayms(300);
96. }
98. toDisplay(value);

101. }
103. }
105. //=============================== timer =============================================
106. #pragma vector = TIMER0\_A0\_VECTOR
107. \_\_interrupt void Timer\_A0(void) {
108. scanled();
109. }
111. //============================= function ===============================
112. void delayms(int ms) {
113. int i;
114. for (i = 0; i < ms; i++)
115. \_\_delay\_cycles(1000);
116. }
118. void scanled() {
120. PDAT = code7seg[buffer[idx]];
121. PROW |= code74HC[idx];
123. \_\_delay\_cycles(1000);
124. PDAT = 0xFF;
125. PROW &= ~code74HC[idx];
127. idx = idx == 3 ? 0 : idx + 1;
129. }
131. void toDisplay(int value) {
133. buffer[3] = value % 10;
134. value /= 10;
135. buffer[2] = value % 10;
136. value /= 10;
137. buffer[1] = value % 10;
138. value /= 10;
139. buffer[0] = value % 10;
140. }

**Nhận xét:**

Khi sử dụng ngắt timer để quét led, không còn hiện tượng chớp tắt led khi ta ấn các nút

**2.4 ==================================================**

2. #include "msp430g2553.h"
4. // led 7seg pin
5. #define PDAT P2OUT
6. #define PROW P1OUT
8. //----for key------------
9. #define K1 BIT0
10. #define K2 BIT1
11. #define K3 BIT2
12. #define K4 BIT3
14. // function prototype declaration
15. void delayms(int);
16. void scanled();
17. void toDisplay(int);
19. // variable declaration
20. char code7seg[] = {
21. 0xC0,
22. 0xF9,
23. 0xA4,
24. 0xB0,
25. 0x99,
26. 0x92,
27. 0x82,
28. 0xF8,
29. 0x80,
30. 0x90
31. };
33. char code74HC[] = {
34. 0x00,
35. 0x40,
36. 0x80,
37. 0xC0
38. };
40. char buffer[4] = {
41. 0,
42. 0,
43. 0,
44. 0
45. };
46. int idx = 0;
47. int value = 0;
48. int interval = 0;
49. int status = 0;
50. // =============================== main ============================
52. void main(void) {
53. // Stop watchdog timer to prevent time out reset
54. WDTCTL = WDTPW + WDTHOLD;
56. //Set MCLK = SMCLK = 1MHz
57. DCOCTL = 0;
58. BCSCTL1 = CALBC1\_1MHZ;
59. DCOCTL = CALDCO\_1MHZ;
61. //P1SEL = 0x00;
62. P1DIR = 0xF0;
63. P1REN = 0x0F;
64. P1OUT = 0xFF;
66. P2SEL = 0x00;
67. P2DIR = 0xFF;
69. TA0CTL = TASSEL\_2 + ID\_0 + MC\_1; // SMCLK as clock source, divider 0, up-mode
71. CCTL0 = CCIE; // Enable interrupt for CCR0
72. CCR0 = 5000; // 5ms
73. \_BIS\_SR(GIE); // Enable global interrupts
75. toDisplay(value);
77. for (;;) {

80. if ((P1IN & K1) == 0) {
81. delayms(300);
82. status = 0;
83. value++;
85. }
87. if ((P1IN & K2) == 0) {
88. delayms(300);
89. status = 0;
90. value--;
92. }
94. if ((P1IN & K3) == 0){ delayms(500); status = 1;}
95. if ((P1IN & K4) == 0){ delayms(500); status = 2;}

98. if (interval >= 200) {
99. interval = 0;
100. switch (status) {
101. case 1: value++; break;
102. case 2: value = value > 0 ? value - 1 : 0; break;
103. }
104. }


108. toDisplay(value);
110. //scanled();
111. delayms(5);
112. interval++;
113. }
115. }
117. //=============================== timer =============================================
118. #pragma vector = TIMER0\_A0\_VECTOR
119. \_\_interrupt void Timer\_A0(void) {
120. scanled();
121. }
123. //============================= function ===============================
124. void delayms(int ms) {
125. int i;
126. for (i = 0; i < ms; i++)
127. \_\_delay\_cycles(1000);
128. }
130. void scanled() {
132. PDAT = code7seg[buffer[idx]];
133. PROW |= code74HC[idx];
135. \_\_delay\_cycles(1000);
136. PDAT = 0xFF;
137. PROW &= ~code74HC[idx];
139. idx = idx == 3 ? 0 : idx + 1;
141. }
143. void toDisplay(int value) {
145. buffer[3] = value % 10;
146. value /= 10;
147. buffer[2] = value % 10;
148. value /= 10;
149. buffer[1] = value % 10;
150. value /= 10;
151. buffer[0] = value % 10;
152. }

**Nhận xét:**

Khi sử dụng ngắt timer để quét led, không còn hiện tượng chớp tắt led khi ta ấn các nút

**Vấn đề 1 ======================================**

1. #include "msp430g2553.h"
3. // led 7seg pin
4. #define PDAT P2OUT
5. #define PROW P1OUT
7. //----for key------------
8. #define K1 BIT0
9. #define K2 BIT1
10. #define K3 BIT2
11. #define K4 BIT3
13. // function prototype declaration
14. void delayms(int);
15. void scanled();
16. void toDisplay(int);
18. // variable declaration
19. char code7seg[] = {
20. 0xC0,
21. 0xF9,
22. 0xA4,
23. 0xB0,
24. 0x99,
25. 0x92,
26. 0x82,
27. 0xF8,
28. 0x80,
29. 0x90
30. };
32. char code74HC[] = {
33. 0x00,
34. 0x40,
35. 0x80,
36. 0xC0
37. };
39. char buffer[4] = {
40. 0,
41. 0,
42. 0,
43. 0
44. };
45. int idx = 0;
46. int ss = 0, SS = 0;
47. int status = 0;
48. // =============================== main ============================
50. void main(void) {
51. // Stop watchdog timer to prevent time out reset
52. WDTCTL = WDTPW + WDTHOLD;
54. //Set MCLK = SMCLK = 1MHz
55. DCOCTL = 0;
56. BCSCTL1 = CALBC1\_1MHZ;
57. DCOCTL = CALDCO\_1MHZ;
59. //P1SEL = 0x00;
60. P1DIR = 0xF0;
61. P1REN = 0x0F;
62. P1OUT = 0xFF;
64. P2SEL = 0x00;
65. P2DIR = 0xFF;
67. TA0CTL = TASSEL\_2 + ID\_0 + MC\_1; // SMCLK as clock source, divider 0, up-mode
69. CCTL0 = CCIE; // Enable interrupt for CCR0
70. CCR0 = 5000; // 5ms
71. \_BIS\_SR(GIE); // Enable global interrupts


75. for (;;) {
77. if ((P1IN & K1) == 0) {
78. delayms(300);
79. status = 1;
80. }
82. if ((P1IN & K2) == 0) {
83. status = 0;
84. delayms(300);
85. }
87. if ((P1IN & K3) == 0) {
88. SS = ss = 0;
89. delayms(300);
90. }
92. toDisplay( (SS\*100) + ss );
93. delayms(10);
95. if (status != 0) {
97. ss = ss >= 99 ? 0 : ss + 1;
99. if (ss == 0) {
100. SS = SS >= 99 ? 0 : SS + 1;
101. }

104. }
105. }
107. }
109. //=============================== timer =============================================
110. #pragma vector = TIMER0\_A0\_VECTOR
111. \_\_interrupt void Timer\_A0(void) {
112. scanled();
113. }
115. //============================= function ===============================
116. void delayms(int ms) {
117. int i;
118. for (i = 0; i < ms; i++)
119. \_\_delay\_cycles(1000);
120. }
122. void scanled() {
124. PDAT = code7seg[buffer[idx]];
125. PROW |= code74HC[idx];
127. \_\_delay\_cycles(1000);
128. PDAT = 0xFF;
129. PROW &= ~code74HC[idx];
131. idx = idx == 3 ? 0 : idx + 1;
133. }
135. void toDisplay(int value) {
137. buffer[3] = value % 10;
138. value /= 10;
139. buffer[2] = value % 10;
140. value /= 10;
141. buffer[1] = value % 10;
142. value /= 10;
143. buffer[0] = value % 10;
144. }