**3.1 ===========================================**

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);
17. unsigned int ReadADC10(int chanel);
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;
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. TA0CTL = TASSEL\_2 + ID\_0 + MC\_1; // SMCLK as clock source, divider 0, up-mode
70. // CCTL0 = CCIE; // Enable interrupt for CCR0
71. // CCR0 = 5000; // 5ms
72. // \_BIS\_SR(GIE); // Enable global interrupts
74. // Khởi động ADC
75. ADC10CTL1 = ADC10DIV\_3;
76. ADC10CTL0 = SREF\_1 + ADC10SHT\_3 + REFON + ADC10ON;
78. toDisplay(value);
80. for (;;) {
82. // xử lý phím ấn
83. if ((P1IN & K1) == 0) {
84. delayms(100);
85. value = ReadADC10(10);
86. }
88. if ((P1IN & K2) == 0) {
89. delayms(100);
90. value = ReadADC10(10);
91. value = ((value - 673) \* 423) / 1024;
92. }
94. if ((P1IN & K3) == 0) {
95. delayms(100);
96. value = 0;
97. }
99. toDisplay(value);
101. scanled();
102. delayms(5);
103. }
105. }
107. //=============================== timer =============================================
108. #pragma vector = TIMER0\_A0\_VECTOR
109. \_\_interrupt void Timer\_A0(void) {
111. }
113. //============================= function ===============================
114. unsigned int ReadADC10(int chanel) {
115. ADC10AE0 |= (BIT0<<chanel);
116. ADC10CTL1 |= (chanel\*0x1000u);
117. ADC10CTL0 |= ENC + ADC10SC;
118. while(!(ADC10CTL0 & ADC10IFG));
119. return ADC10MEM;
120. }
122. void delayms(int ms) {
123. int i;
124. for (i = 0; i < ms; i++)
125. \_\_delay\_cycles(1000);
126. }
128. void scanled() {
130. PDAT = code7seg[buffer[idx]];
131. PROW |= code74HC[idx];
133. \_\_delay\_cycles(1000);
134. PDAT = 0xFF;
135. PROW &= ~code74HC[idx];
137. idx = idx == 3 ? 0 : idx + 1;
139. }
141. void toDisplay(int value) {
143. buffer[3] = value % 10;
144. value /= 10;
145. buffer[2] = value % 10;
146. value /= 10;
147. buffer[1] = value % 10;
148. value /= 10;
149. buffer[0] = value % 10;
150. }

**Giải thích các đoạn tô đậm:**

1. // Khởi động ADC
2. ADC10CTL1 = ADC10DIV\_3; // chọn bộ chia clock là 3
3. ADC10CTL0 = SREF\_1 + ADC10SHT\_3 + REFON + ADC10ON;
4. // SREF\_1 : chọn tham chiếu là 1 ( VR+ = VREF+ and VR- = VSS )
5. // ADC10SHT\_3: chọn chế độ lấy mẫu là 3 ( 16 × ADC10CLKs )
6. // REFON : bật điện áp tham chiếu
7. // ADC10ON : bật ADC
9. unsigned int ReadADC10(int chanel) {
10. ADC10AE0 |= (BIT0<<chanel);
11. // Bật ADC ở chanel tương ứng
13. ADC10CTL1 |= (chanel\*0x1000u);
14. // Bật ADC ở chanel tương ứng
16. ADC10CTL0 |= ENC + ADC10SC;
17. // ENC : Enable conversion
18. // ADC10SC: start conversion
20. while(!(ADC10CTL0 & ADC10IFG));
21. // waiting to conversion end
23. return ADC10MEM;
24. }

**3.2 ===========================================**

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);
17. unsigned int ReadADC10(int chanel);
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;
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. TA0CTL = TASSEL\_2 + ID\_0 + MC\_1; // SMCLK as clock source, divider 0, up-mode
70. CCTL0 = CCIE; // Enable interrupt for CCR0
71. CCR0 = 5000; // 5ms
72. \_BIS\_SR(GIE); // Enable global interrupts
74. // Khởi động ADC
75. ADC10CTL1 = ADC10DIV\_3;
76. ADC10CTL0 = SREF\_1 + ADC10SHT\_3 + REFON + ADC10ON;
78. toDisplay(value);
80. for (;;) {
82. // xử lý phím ấn
83. if ((P1IN & K1) == 0) {
84. delayms(100);
85. value = ReadADC10(10);
86. }
88. if ((P1IN & K2) == 0) {
89. delayms(100);
90. value = ReadADC10(10);
91. value = ((value - 673) \* 423) / 1024;
92. }
94. if ((P1IN & K3) == 0) {
95. delayms(100);
96. value = 0;
97. }
99. toDisplay(value);
101. //scanled();
102. //delayms(5);
103. }
105. }
107. //=============================== timer =============================================
108. #pragma vector = TIMER0\_A0\_VECTOR
109. \_\_interrupt void Timer\_A0(void) {
110. scanled();
111. }
113. //============================= function ===============================
114. unsigned int ReadADC10(int chanel) {
115. ADC10AE0 |= (BIT0<<chanel);
116. ADC10CTL1 |= (chanel\*0x1000u);
117. ADC10CTL0 |= ENC + ADC10SC;
118. while(!(ADC10CTL0 & ADC10IFG));
119. return ADC10MEM;
120. }
122. void delayms(int ms) {
123. int i;
124. for (i = 0; i < ms; i++)
125. \_\_delay\_cycles(1000);
126. }
128. void scanled() {
130. PDAT = code7seg[buffer[idx]];
131. PROW |= code74HC[idx];
133. \_\_delay\_cycles(1000);
134. PDAT = 0xFF;
135. PROW &= ~code74HC[idx];
137. idx = idx == 3 ? 0 : idx + 1;
139. }
141. void toDisplay(int value) {
143. buffer[3] = value % 10;
144. value /= 10;
145. buffer[2] = value % 10;
146. value /= 10;
147. buffer[1] = value % 10;
148. value /= 10;
149. buffer[0] = value % 10;
150. }

Nhận xét:

Kết quả giống như 3.1 nhưng led 7 đoạn không còn chớp tắt khi ta tác động vào công tắt

3.3 ===================================================

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);
17. unsigned int ReadADC10(int chanel);
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;
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. TA0CTL = TASSEL\_2 + ID\_0 + MC\_1; // SMCLK as clock source, divider 0, up-mode
70. CCTL0 = CCIE; // Enable interrupt for CCR0
71. CCR0 = 5000; // 5ms
72. \_BIS\_SR(GIE); // Enable global interrupts
74. // Khởi động ADC
75. ADC10CTL1 = ADC10DIV\_3;
76. ADC10CTL0 = SREF\_1 + ADC10SHT\_3 + REFON + ADC10ON;
78. toDisplay(value);
80. for (;;) {
82. // xử lý phím ấn
83. if ((P1IN & K1) == 0) {
84. delayms(100);
85. value = ReadADC10(11);
86. }
88. if ((P1IN & K2) == 0) {
89. delayms(100);
90. value = ReadADC10(11);
91. // cong thuc theo trang nay
92. //https://e2e.ti.com/support/microcontrollers/msp-low-power-microcontrollers-group/msp430/f/msp-low-power-microcontroller-forum/163466/measuring-vcc-using-adc-channel-11
93. // x10 cho xuat hien phan thap phan
94. value = (int)(2 \* value \* (1.5/1023.0) \* 10);
95. }
97. if ((P1IN & K3) == 0) {
98. delayms(100);
99. value = 0;
100. }
102. toDisplay(value);
104. //scanled();
105. //delayms(5);
106. }
108. }
110. //=============================== timer =============================================
111. #pragma vector = TIMER0\_A0\_VECTOR
112. \_\_interrupt void Timer\_A0(void) {
113. scanled();
114. }
116. //============================= function ===============================
117. unsigned int ReadADC10(int chanel) {
118. ADC10AE0 |= (BIT0<<chanel);
119. ADC10CTL1 |= (chanel\*0x1000u);
120. ADC10CTL0 |= ENC + ADC10SC;
121. while(!(ADC10CTL0 & ADC10IFG));
122. return ADC10MEM;
123. }
125. void delayms(int ms) {
126. int i;
127. for (i = 0; i < ms; i++)
128. \_\_delay\_cycles(1000);
129. }
131. void scanled() {
133. PDAT = code7seg[buffer[idx]];
134. PROW |= code74HC[idx];
136. \_\_delay\_cycles(1000);
137. PDAT = 0xFF;
138. PROW &= ~code74HC[idx];
140. idx = idx == 3 ? 0 : idx + 1;
142. }
144. void toDisplay(int value) {
146. buffer[3] = value % 10;
147. value /= 10;
148. buffer[2] = value % 10;
149. value /= 10;
150. buffer[1] = value % 10;
151. value /= 10;
152. buffer[0] = value % 10;
153. }

kết quả:

giá trị thô thu được : 438

khi đã chuyển đỗi sang giá trị điện áp: 1.4

3.4 ======================================

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);
17. unsigned int ReadADC10(int chanel);
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;
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. TA0CTL = TASSEL\_2 + ID\_0 + MC\_1; // SMCLK as clock source, divider 0, up-mode
70. CCTL0 = CCIE; // Enable interrupt for CCR0
71. CCR0 = 5000; // 5ms
72. \_BIS\_SR(GIE); // Enable global interrupts
74. // Khởi động ADC
75. ADC10CTL1 = ADC10DIV\_3;
76. ADC10CTL0 = SREF\_1 + ADC10SHT\_3 + REFON + ADC10ON;
78. toDisplay(value);
80. for (;;) {
82. //value = ReadADC10(0);
83. //value = ReadADC10(1);
84. //value = ReadADC10(2);
85. value = ReadADC10(3);
87. toDisplay(value);
88. delayms(100);
90. //scanled();
91. //delayms(5);
92. }
94. }
96. //=============================== timer =============================================
97. #pragma vector = TIMER0\_A0\_VECTOR
98. \_\_interrupt void Timer\_A0(void) {
99. scanled();
100. }
102. //============================= function ===============================
103. unsigned int ReadADC10(int chanel) {
104. ADC10AE0 |= (BIT0<<chanel);
105. ADC10CTL1 |= (chanel\*0x1000u);
106. ADC10CTL0 |= ENC + ADC10SC;
107. while(!(ADC10CTL0 & ADC10IFG));
108. return ADC10MEM;
109. }
111. void delayms(int ms) {
112. int i;
113. for (i = 0; i < ms; i++)
114. \_\_delay\_cycles(1000);
115. }
117. void scanled() {
119. PDAT = code7seg[buffer[idx]];
120. PROW |= code74HC[idx];
122. \_\_delay\_cycles(1000);
123. PDAT = 0xFF;
124. PROW &= ~code74HC[idx];
126. idx = idx == 3 ? 0 : idx + 1;
128. }
130. void toDisplay(int value) {
132. buffer[3] = value % 10;
133. value /= 10;
134. buffer[2] = value % 10;
135. value /= 10;
136. buffer[1] = value % 10;
137. value /= 10;
138. buffer[0] = value % 10;
139. }

Kết quả: Khi thử lần lượt các kênh A0 (LM35), A1(VR1) , A2(VR2), A3(LDR) thì kết quả cho ra có sự khác biệt nhỏ khi ấn phím tương ứng và không ấn.