**4.2 ====================================================**

1. #include <msp430f2112.h>
3. #include <stdio.h>
5. #include <stdlib.h>
7. #define SMCLK\_F 1000000 // frequency of Sub-System Master Clock in Hz
8. #define BAUDRATE 9600 // may be ... 1200, 2400, 4800, 9600, 19200, ...
9. #define UART\_RX\_INT\_EN 0 // cau hinh nhan du lieu: 0 - khong dùng INT; 1 - dùng INT
11. //----for key------------
12. #define K1 BIT0
13. #define K4 BIT3
15. //------------- Khai báo prototype hàm
16. void initUART();
17. void uart\_putc(char c);
18. void uart\_puts(const char \* s);
19. char uart\_data\_ready();
20. char uart\_getc();
21. void uart\_gets(char \* s);
22. void delayms(int ms);
24. //-------Khai báo biến-----------------
25. char chrin;
27. //---Định nghĩa chương trình chính-------
28. void main(void) {
30. WDTCTL = WDTPW + WDTHOLD;
32. P2DIR = 0xF0;
33. P2REN = 0x09;
34. P2OUT = 0x09;
36. P1DIR = 0xFF;
38. initUART(); // khởi động UART
39. uart\_putc('@');
41. delayms(1000);
43. // Vòng lặp chương trình chính
44. while (1) {
46. // nhận phím ấn
47. if ((P2IN & K1) == 0) uart\_putc('1');
48. if ((P2IN & K4) == 0) uart\_putc('4');
50. //kiểm tra và xử lý ký tự nhận được từ PC
51. if (uart\_data\_ready()) {
52. char kytu = uart\_getc();
53. if (kytu != 0) {
54. // nhận được ký tự và đưa ra LED đơn
55. P1OUT = kytu;
56. }
57. }
59. delayms(20);
60. } // end while
61. } // end main
63. //---------------Thu vien UART------------------------
64. void initUART() {
65. unsigned int tmp;
66. // cau hinh bo dao dong DCO Fsc = 1MHZ
67. BCSCTL1 = CALBC1\_1MHZ;
68. DCOCTL = CALDCO\_1MHZ;
70. // cau hinh chan cho UART
71. P3SEL |= BIT4 + BIT5;
72. //P3SEL2 |= BIT4+BIT5;
74. // khung du lieu
75. UCA0CTL0 = 0x00; // 8 bit, none, 1 stop
77. // Chon xung CLK
78. UCA0CTL1 |= UCSSEL\_2; // chon clk = SMCLK = 1MHZ
80. // thiet lap toc do baud
81. //UCA0BR0 = 104; // 9600
82. //UCA0BR1 = 0;
83. tmp = SMCLK\_F / BAUDRATE;
84. UCA0BR0 = tmp & 0x00FF;
85. UCA0BR1 = ((tmp >> 8) & 0x00FF);
87. UCA0MCTL = UCBRS\_1; // cau hinh thu tu bit truyen--> MSB first
89. UCA0CTL1 &= ~UCSWRST; // Start UART
90. if (UART\_RX\_INT\_EN) {
91. IE2 |= UCA0RXIE;
92. \_\_bis\_SR\_register(GIE);
93. }
94. }
95. //==============Xu ly ngat--UART-------------------------
96. #pragma vector = USCIAB0RX\_VECTOR
97. \_\_interrupt void USCI0RX\_ISR(void) {
98. // chrin = uart\_getc();
99. // switch(chrin){
100. // case '1': P3OUT |=LEDR;
101. // break;
102. // case '2': P3OUT &=~LEDR;
103. // break;
104. // }
105. }
107. //------------------------------------
108. void uart\_putc(char c) {
109. while (!(IFG2 & UCA0TXIFG)); // Wait until USCI\_A0 TX buffer empty
110. UCA0TXBUF = c; // assign character to TX buffer
111. }
112. //------------------------------------
113. void uart\_puts(const char \* s) {
114. while ( \* s != '\0') {
115. uart\_putc( \* s);
116. s++;
117. }
118. }
119. //------------------------------------
120. char uart\_getc() {
121. while (!(IFG2 & UCA0RXIFG)); // Wait until USCI\_A0 RX receive a complete character
122. return UCA0RXBUF; // assign RX buffer to function 's name
123. }
124. //------------------------------------
125. void uart\_gets(char \* s) {
126. // if (IFG2&UCA0RXIFG){
127. \* s = uart\_getc();
128. while ( \* s != 0x0d) {
129. s++;
130. \* s = uart\_getc();
131. }
132. \* s = 0;
133. // }
134. }
135. //------------------------------------
136. char uart\_data\_ready() {
137. if (IFG2 & UCA0RXIFG) return 1;
138. else return 0;
139. }
140. //----------------------------
141. void delayms(int ms) {
142. int i;
143. for (i = 0; i < ms; i++)
144. \_\_delay\_cycles(100);
145. }

**Thông số uart:**

**Clock: 1mhz**

**Baud : 9600**

**data bit: 8**

**parity: none:**

**stop bit: 1**

**Kết quả ra đúng yêu cầu nhưng có hiện tượng dội phím**

**4.3 ==================================================**

1. #include <msp430f2112.h>
3. #include <stdio.h>
5. #include <stdlib.h>
7. #define SMCLK\_F 1000000 // frequency of Sub-System Master Clock in Hz
8. #define BAUDRATE 9600 // may be ... 1200, 2400, 4800, 9600, 19200, ...
9. #define UART\_RX\_INT\_EN 0 // not use INT
11. // led 7seg pin
12. #define PDAT P1OUT
13. #define PROW P3OUT
15. // key button pin
16. #define K1 BIT0
17. #define K4 BIT3
19. // function prototype declaration
20. void initUART();
21. void uart\_putc(char c);
22. void uart\_puts(const char \* s);
23. char uart\_data\_ready();
24. char uart\_getc();
25. void uart\_gets(char \* s);
27. void delayms(int ms);
28. void scanled();
29. void toDisplay(int m);
31. // variable declaration
32. char chrin;
33. char code7seg[] = {
34. 0xC0,
35. 0xF9,
36. 0xA4,
37. 0xB0,
38. 0x99,
39. 0x92,
40. 0x82,
41. 0xF8,
42. 0x80,
43. 0x90
44. };
46. char code74HC[] = {
47. 0x00,
48. 0x01,
49. 0x02,
50. 0x03
51. };
53. char buffer[4] = {
54. 0,
55. 0,
56. 0,
57. 0
58. };
59. int idx = 0;
61. void main(void) {
63. WDTCTL = WDTPW + WDTHOLD;
65. P2DIR = 0xF0;
66. P2REN = 0x09;
67. P2OUT = 0x09;
69. P1DIR = 0xFF;
70. P3DIR = 0xFF;
72. initUART();
73. uart\_puts("Chao cac ban !\n");
75. int delay\_time = 200;
77. while (1) {
79. // nhận phím ấn
80. if ((P2IN & K1) == 0) {
81. uart\_puts("VDK MSP430g2553 !\n");
82. delayms(delay\_time);
83. }
84. if ((P2IN & K4) == 0) {
85. uart\_puts("TDH - CDT !\n");
86. delayms(delay\_time);
87. }
89. if (uart\_data\_ready()) {
90. char c = uart\_getc();
91. toDisplay(c);
92. }
94. scanled();
95. delayms(5);
96. }
97. }
99. //---------------Thu vien UART------------------------
100. void initUART() {
101. unsigned int tmp;
102. // cau hinh bo dao dong DCO Fsc = 1MHZ
103. BCSCTL1 = CALBC1\_1MHZ;
104. DCOCTL = CALDCO\_1MHZ;
106. // cau hinh chan cho UART
107. P3SEL |= BIT4 + BIT5;
108. //P3SEL2 |= BIT4+BIT5;
110. // khung du lieu
111. UCA0CTL0 = 0x00; // 8 bit, none, 1 stop
113. // Chon xung CLK
114. UCA0CTL1 |= UCSSEL\_2; // chon clk = SMCLK = 1MHZ
116. tmp = SMCLK\_F / BAUDRATE;
117. UCA0BR0 = tmp & 0x00FF;
118. UCA0BR1 = ((tmp >> 8) & 0x00FF);
120. UCA0MCTL = UCBRS\_1; // cau hinh thu tu bit truyen--> MSB first
122. UCA0CTL1 &= ~UCSWRST; // Start UART
123. if (UART\_RX\_INT\_EN) {
124. IE2 |= UCA0RXIE;
125. \_\_bis\_SR\_register(GIE);
126. }
127. }
129. void uart\_putc(char c) {
130. while (!(IFG2 & UCA0TXIFG)); // Wait until USCI\_A0 TX buffer empty
131. UCA0TXBUF = c; // assign character to TX buffer
132. }
134. void uart\_puts(const char \* s) {
135. while ( \* s != '\0') {
136. uart\_putc( \* s);
137. s++;
138. }
139. }
140. char uart\_getc() {
141. while (!(IFG2 & UCA0RXIFG)); // Wait until USCI\_A0 RX receive a complete character
142. return UCA0RXBUF; // assign RX buffer to function 's name
143. }
144. void uart\_gets(char \* s) {
145. \* s = uart\_getc();
146. while ( \* s != 0x0d) {
147. s++;
148. \* s = uart\_getc();
149. }
150. \* s = 0;
151. }
152. char uart\_data\_ready() {
153. if (IFG2 & UCA0RXIFG) return 1;
154. else return 0;
155. }
157. void delayms(int ms) {
158. int i;
159. for (i = 0; i < ms; i++)
160. \_\_delay\_cycles(1000);
161. }
163. void scanled() {
165. PDAT = code7seg[buffer[idx]];
166. PROW |= code74HC[idx];
168. \_\_delay\_cycles(1000);
169. PDAT = 0xFF;
170. PROW &= ~code74HC[idx];
172. idx = idx == 3 ? 0 : idx + 1;
174. }
176. void toDisplay(int value) {
178. buffer[3] = value % 10; value /= 10;
179. buffer[2] = value % 10; value /= 10;
180. buffer[1] = value % 10; value /= 10;
181. buffer[0] = value % 10;
182. }

**Thông số uart:**

**Clock: 1mhz**

**Baud : 9600**

**data bit: 8**

**parity: none:**

**stop bit: 1**

**Kết quả ra đúng yêu cầu nhưng có hiện tượng dội phím**

**4.4 =======================================**

1. #include <msp430f2112.h>
3. #include <stdio.h>
5. #include <stdlib.h>
7. #define SMCLK\_F 1000000 // frequency of Sub-System Master Clock in Hz
8. #define BAUDRATE 9600 // may be ... 1200, 2400, 4800, 9600, 19200, ...
9. #define UART\_RX\_INT\_EN 0 // not use INT
11. // led 7seg pin
12. #define PDAT P1OUT
13. #define PROW P3OUT
15. // key button pin
16. #define K1 BIT0
17. #define K4 BIT3
19. // function prototype declaration
20. void initUART();
21. void uart\_putc(char c);
22. void uart\_puts(const char \* s);
23. char uart\_data\_ready();
24. char uart\_getc();
25. void uart\_gets(char \* s);
27. void delayms(int ms);
28. void scanled();
29. void toDisplay(int m);
31. unsigned int ReadADC10(int chanel);
33. // variable declaration
34. char chrin;
35. char code7seg[] = {
36. 0xC0,
37. 0xF9,
38. 0xA4,
39. 0xB0,
40. 0x99,
41. 0x92,
42. 0x82,
43. 0xF8,
44. 0x80,
45. 0x90
46. };
48. char code74HC[] = {
49. 0x00,
50. 0x01,
51. 0x02,
52. 0x03
53. };
55. char buffer[4] = {
56. 0,
57. 0,
58. 0,
59. 0
60. };
61. int idx = 0;
62. char str[50];
63. void main(void) {
65. WDTCTL = WDTPW + WDTHOLD;
67. P2DIR = 0xF0;
68. P2REN = 0x09;
69. P2OUT = 0x09;
71. P1DIR = 0xFF;
72. P3DIR = 0xFF;

75. // Khởi động ADC
76. ADC10CTL1 = ADC10DIV\_3;
77. ADC10CTL0 = SREF\_1 + ADC10SHT\_3 + REFON + ADC10ON;
79. initUART();
80. uart\_puts("Chao cac ban !\n");

83. scanled();
85. int value = 0;
87. while (1) {
89. value = ((ReadADC10(10) - 673) \* 423) / 1024;


93. sprintf(str, "Gia tri cua ADC = %d\n", value);
95. uart\_puts(str);
97. delayms(2000);
99. }
100. }
102. //================================= uart ======================================
103. void initUART() {
104. unsigned int tmp;
105. // cau hinh bo dao dong DCO Fsc = 1MHZ
106. BCSCTL1 = CALBC1\_1MHZ;
107. DCOCTL = CALDCO\_1MHZ;
109. // cau hinh chan cho UART
110. P3SEL |= BIT4 + BIT5;
111. //P3SEL2 |= BIT4+BIT5;
113. // khung du lieu
114. UCA0CTL0 = 0x00; // 8 bit, none, 1 stop
116. // Chon xung CLK
117. UCA0CTL1 |= UCSSEL\_2; // chon clk = SMCLK = 1MHZ
119. tmp = SMCLK\_F / BAUDRATE;
120. UCA0BR0 = tmp & 0x00FF;
121. UCA0BR1 = ((tmp >> 8) & 0x00FF);
123. UCA0MCTL = UCBRS\_1; // cau hinh thu tu bit truyen--> MSB first
125. UCA0CTL1 &= ~UCSWRST; // Start UART
126. if (UART\_RX\_INT\_EN) {
127. IE2 |= UCA0RXIE;
128. \_\_bis\_SR\_register(GIE);
129. }
130. }
132. void uart\_putc(char c) {
133. while (!(IFG2 & UCA0TXIFG)); // Wait until USCI\_A0 TX buffer empty
134. UCA0TXBUF = c; // assign character to TX buffer
135. }
137. void uart\_puts(const char \* s) {
138. while ( \* s != '\0') {
139. uart\_putc( \* s);
140. s++;
141. }
142. }
143. char uart\_getc() {
144. while (!(IFG2 & UCA0RXIFG)); // Wait until USCI\_A0 RX receive a complete character
145. return UCA0RXBUF; // assign RX buffer to function 's name
146. }
147. void uart\_gets(char \* s) {
148. \* s = uart\_getc();
149. while ( \* s != 0x0d) {
150. s++;
151. \* s = uart\_getc();
152. }
153. \* s = 0;
154. }
155. char uart\_data\_ready() {
156. if (IFG2 & UCA0RXIFG) return 1;
157. else return 0;
158. }
160. // ===================== adc ==================
161. unsigned int ReadADC10(int chanel) {
162. ADC10AE0 |= (BIT0<<chanel);
163. ADC10CTL1 |= (chanel\*0x1000u);
164. ADC10CTL0 |= ENC + ADC10SC;
165. while(!(ADC10CTL0 & ADC10IFG));
166. return ADC10MEM;
167. }
169. //====================================================
170. void delayms(int ms) {
171. int i;
172. for (i = 0; i < ms; i++)
173. \_\_delay\_cycles(1000);
174. }
175. //====================================================
177. void scanled() {
179. PDAT = code7seg[buffer[idx]];
180. PROW |= code74HC[idx];
182. \_\_delay\_cycles(1000);
183. PDAT = 0xFF;
184. PROW &= ~code74HC[idx];
186. idx = idx == 3 ? 0 : idx + 1;
188. }
189. //====================================================
191. void toDisplay(int value) {
193. buffer[3] = value % 10; value /= 10;
194. buffer[2] = value % 10; value /= 10;
195. buffer[1] = value % 10; value /= 10;
196. buffer[0] = value % 10;
197. }