

```
1  #include "keypad.h"
2  #include "lcd.h"
3
4  void keypad_ports()
5  {
6      //Pins 5-7 Port A, push pull output
7      GPIOA->CRL |= GPIO_CRL_MODE5 | GPIO_CRL_MODE6 | GPIO_CRL_MODE7;
8      GPIOA->CRL &= ~GPIO_CRL_CNF5 & ~GPIO_CRL_CNF6 & ~GPIO_CRL_CNF7;
9
10     //Pins 9-12 Port A, internal pull up used in input mode
11     GPIOA->CRH |= GPIO_CRH_CNF10_1 | GPIO_CRH_CNF11_1 | GPIO_CRH_CNF12_1;
12     GPIOA->CRH &= ~GPIO_CRH_CNF10_0 & ~GPIO_CRH_CNF11_0 & ~GPIO_CRH_CNF12_0;
13     GPIOA->CRL |= GPIO_CRL_CNF3_1;
14     GPIOA->CRL &= ~GPIO_CRL_CNF3_0;
15     GPIOA->ODR = 0x1C08;
16
17 }
18 uint8_t read_key()
19 {
20     GPIOA->BRR |= GPIO_BRR_BR5;
21     GPIOA->BSRR |= GPIO_BSRR_BS6 | GPIO_BSRR_BS7;
22
23     delay(300);
24     if ((GPIOA->IDR & GPIO_IDR_IDR3) == 0)
25         return '1';
26     if ((GPIOA->IDR & GPIO_IDR_IDR10) == 0)
27         return '4';
28     if ((GPIOA->IDR & GPIO_IDR_IDR11) == 0)
29         return '7';
30     if ((GPIOA->IDR & GPIO_IDR_IDR12) == 0)
31         return '*';
32
33     GPIOA->BRR |= GPIO_BRR_BR6;
34     GPIOA->BSRR |= GPIO_BSRR_BS5 | GPIO_BSRR_BS7;
35
36     delay(300);
37     if ((GPIOA->IDR & GPIO_IDR_IDR3) == 0)
38         return '2';
39     if ((GPIOA->IDR & GPIO_IDR_IDR10) == 0)
40         return '5';
41     if ((GPIOA->IDR & GPIO_IDR_IDR11) == 0)
42         return '8';
43     if ((GPIOA->IDR & GPIO_IDR_IDR12) == 0)
44         return '0';
45
46     GPIOA->BRR |= GPIO_BRR_BR7;
47     GPIOA->BSRR |= GPIO_BSRR_BS5 | GPIO_BSRR_BS6;
48
49     delay(300);
50     if ((GPIOA->IDR & GPIO_IDR_IDR3) == 0)
51         return '3';
52     if ((GPIOA->IDR & GPIO_IDR_IDR10) == 0)
53         return '6';
54     if ((GPIOA->IDR & GPIO_IDR_IDR11) == 0)
55         return '9';
56     if ((GPIOA->IDR & GPIO_IDR_IDR12) == 0)
57         return '#';
58
59     return 0xFF;
60 }
61 uint32_t get_password()
62 {
63
64     int x[6];
65     int i = 0;
66     uint32_t temp2 = 0;
67     while(1)
68     {
69         if (read_key() != 0xFF)
70         {
71             x[i] = read_key();
72             Data_2_LCD(0x2A);
73             delay_ms(300);
```

```
74     i++;
75     if (i == 6)
76         break;
77 }
78 }
79 int z = 20;
80 for (int i = 0; i < 6; i++)
81 {
82
83     uint32_t temp1 = 0;
84     temp1 = x[i] - 0x30;
85     temp1 = (temp1 << z);
86     z -= 4;
87     temp2 += temp1;
88 }
89 return temp2;
90 }
```