

```

1  GPIO_PORTB_DATAIN EQU 0x4000503C ;data address to all pins
2  GPIO_PORTB_DATAOUT EQU 0x400053C0
3  GPIO_PORTB_DIR EQU 0x40005400
4  GPIO_PORTB_AFSEL EQU 0x40005420
5  GPIO_PORTB_AMSEL EQU 0x40005428
6  GPIO_PORTB_DEN EQU 0x4000551C
7  GPIO_PORTB_PUR EQU 0x40005510
8  IOB EQU 0xF0
9  PUB EQU 0x0F
10
11 SYSCTL_RCGCGPIO EQU 0x400FE608
12
13
14 AREA main, READONLY, CODE
15 THUMB
16 EXTERN delay
17 EXPORT __main ; Make available
18
19 __main PROC
20 LDR R1, =SYSCTL_RCGCGPIO
21 LDR R0, [R1]
22 ORR R0, #0x02
23 STR R0, [R1]
24
25 NOP
26 NOP
27 NOP ;Stabilize clock
28
29 LDR R1, =GPIO_PORTB_DIR
30 LDR R0, [R1]
31 BIC R0, #0xFF ;INPUTS
32 ORR R0, #IOB ;s1 pb3 d1
33 STR R0, [R1] ;s2 pb2 d2 pb6
34 ;s3 pb1 d3 pb5
35 LDR R1, =GPIO_PORTB_AFSEL ;s4 pb0 d4 pb4
36 LDR R0, [R1]
37 BIC R0, #0xFF
38 STR R0, [R1]
39
40 LDR R1, =GPIO_PORTB_DEN
41 LDR R0, [R1]
42 MOV R0, #0xFF
43 STR R0, [R1]
44
45 LDR R1, =GPIO_PORTB_AMSEL ;PORTB initilization part
46 LDR R0, [R1]
47 BIC R0, #0xFF
48 STR R0, [R1]
49
50 LDR R1, =GPIO_PORTB_PUR
51 MOV R0, #PUB
52 STR R0, [R1]
53
54
55 LDR R1, =GPIO_PORTB_DATAOUT
56 LDR R0, [R1]
57 ORR R0, #0xF0
58 STR R0, [R1]
59
60 checkrows LDR R1, =GPIO_PORTB_DATAIN ;Debounce algorithm for pressing
61 LDR R10, [R1]
62 CMP R10, #0x0F ;wait a delay between two data
63 BEQ checkrows ;samples and if they are the same
64 BLNE delay
65 LDR R1, =GPIO_PORTB_DATAIN ;it continues to check columns
66 LDR R9, [R1]
67 CMP R9, R10 ;it loads the data onto R9 reg.
68 BEQ pressed
69 B checkrows
70
71 pressed LDR R1, =GPIO_PORTB_DATAIN
72 LDR R10, [R1]
73 CMP R9, R10
74 BEQ pressed
75 B led
76

```

```
77     led           LDR     R1,=GPIO_PORTB_DATAOUT
78                     CMP     R9,#0x07
79                     BEQ     first
80                     CMP     R9,#0x0B
81                     BEQ     second
82                     CMP     R9,#0x0D
83                     BEQ     third
84                     CMP     R9,#0x0E
85                     BEQ     fourth
86
87     first          LDR     R0,[R1]
88                     ANDS     R5,R0,#0x80
89                     ORREQ    R0,#0x80
90                     STREQ    R0,[R1]
91                     BEQ     checkrows
92                     BIC     R0,#0x80
93                     STR     R0,[R1]
94                     B        checkrows
95
96
97     second          LDR     R0,[R1]
98                     ANDS     R5,R0,#0x40
99                     ORREQ    R0,#0x40
100                    STREQ    R0,[R1]
101                    BEQ     checkrows
102                    BIC     R0,#0x40
103                    STR     R0,[R1]
104                    B        checkrows
105
106    third           LDR     R0,[R1]
107                    ANDS     R5,R0,#0x20
108                    ORREQ    R0,#0x20
109                    STREQ    R0,[R1]
110                    BEQ     checkrows
111                    BIC     R0,#0x20
112                    STR     R0,[R1]
113                    B        checkrows
114
115    fourth          LDR     R0,[R1]
116                    ANDS     R5,R0,#0x10
117                    ORREQ    R0,#0x10
118                    STREQ    R0,[R1]
119                    BEQ     checkrows
120                    BIC     R0,#0x10
121                    STR     R0,[R1]
122                    B        checkrows
123
124
125                                     ;if everything goes fine code prints the
126                                     ;key's character since it already holds
127                                     ;it as ASCII value
128
129
130
131                    B        checkrows                                     ;code starts over
132
133                    ENDP
134                    ALIGN
135                    END
136
```