```
GPIO_PORTB_DATA     EQU     0x400053FC      ;data address to all pins
GPIO_PORTB_DIR      EQU     0x40005400
GPIO_PORTB_AFSEL    EQU     0x40005420
GPIO_PORTB_AMSEL    EQU     0x40005428
GPIO_PORTB_DEN      EQU     0x4000551C
GPIO_PORTB_PUR      EQU     0x40005510
IOB                 EQU     0xF0
PUB                 EQU     0x0F

SYSCTL_RCGCGPIO     EQU     0x400FE608


            AREA        main, READONLY, CODE
            THUMB
            EXTERN      delay
            EXTERN      OutChar
            EXPORT      __main  ; Make available

__main      PROC
            LDR         R1, =SYSCTL_RCGCGPIO
            LDR         R0,[R1]
            ORR         R0,#0x02
            STR         R0,[R1]

            NOP
            NOP
            NOP                                 ;Stabilize clock

            LDR         R1,=GPIO_PORTB_DIR
            LDR         R0,[R1]
            BIC         R0,#0xFF                            ;INPUTS          OUTPUTS
            ORR         R0,#IOB                         ;r1 pb3          l1
    pb7
            STR         R0,[R1]                         ;r2 pb2          l2  pb6
                                                        ;r3 pb1          l3  pb5
            LDR         R1,=GPIO_PORTB_AFSEL            ;r4 pb0          l4  pb4
            LDR         R0,[R1]
            BIC         R0,#0xFF
            STR         R0,[R1]

            LDR         R1,=GPIO_PORTB_DEN
            LDR         R0,[R1]
            MOV         R0,#0xFF
            STR         R0,[R1]

            LDR         R1,=GPIO_PORTB_AMSEL            ;PORTB initilization part
            LDR         R0,[R1]
            BIC         R0,#0xFF
            STR         R0,[R1]

            LDR         R1,=GPIO_PORTB_PUR
            MOV         R0,#PUB
            STR         R0,[R1]
            MOV         R3,#0x1

start       MOV         R11,#0x70
checkrows   LDR         R1,=GPIO_PORTB_DATA
            LDR         R0,[R1]
            BIC         R0,#0xFF
            ORR         R0,R11
            STR         R0,[R1]

debnc_inp   LDR         R1,=GPIO_PORTB_DATA             ;Debounce algorithm for pressing
            LDR         R10,[R1]                        ;wait a delay between two data
            BL          delay                           ;samples and if they are the same
            LDR         R1,=GPIO_PORTB_DATA             ;it continues to check columns
            LDR         R9,[R1]
            CMP         R9,R10                          ;it loads the data onto R9 reg.
            BEQ         check
            B           debnc_inp

check       CMP         R3,#0x1
            BEQ         rows1
            B           rows2

rows1       CMP         R9,#0x77                        ;rows part checks each column
            MOVEQ       R10,#48                         ;it starts with the first row
```

```
                BEQ.W       cont                    ;checks if the data in R9 is equal
                CMP         R9,#0x7B                ;to any of the 16 keys by simply looking
                MOVEQ       R10,#49                 ;two hex numbers
                BEQ.W       cont                    ;first one for the output in other
                CMP         R9,#0x7D                ;word rows: 7: First row, B: Second
                MOVEQ       R10,#50                 ;D: Third, E: Fourth
                BEQ.W       cont
                CMP         R9,#0x7E                ;the second hex number is for the columns
                MOVEQ       R10,#51                 ;7: First and so on
                BEQ.W       cont                    ;then R10 is loaded with the corresponding
                                                    ;ASCII value of the pressed key
                CMP         R9,#0xB7
                MOVEQ       R10,#52
                BEQ.W       cont
                CMP         R9,#0xBB
                MOVEQ       R10,#53
                BEQ.W       cont
                CMP         R9,#0xBD
                MOVEQ       R10,#54
                BEQ.W       cont
                CMP         R9,#0xBE
                MOVEQ       R10,#55
                BEQ.W       cont

                CMP         R9,#0xD7
                MOVEQ       R10,#56
                BEQ.W       cont
                CMP         R9,#0xDB
                MOVEQ       R10,#57
                BEQ.W       cont
                CMP         R9,#0xDD
                MOVEQ       R10,#65
                BEQ.W       cont
                CMP         R9,#0xDE
                MOVEQ       R10,#66
                BEQ.W       cont

                CMP         R9,#0xE7
                MOVEQ       R3,#1
                BEQ         start
                CMP         R9,#0xEB
                MOVEQ       R3,#2
                BEQ         start
                CMP         R9,#0xED
                MOVEQ       R10,#69
                BEQ         start
                CMP         R9,#0xEE
                MOVEQ       R10,#70
                BEQ.W       start
                B           devam

rows2           CMP         R9,#0x77                ;rows part checks each column
                MOVEQ       R10,#66                 ;it starts with the first row
                BEQ         cont                    ;checks if the data in R9 is equal
                CMP         R9,#0x65                ;to any of the 16 keys by simply looking
                MOVEQ       R10,#65                 ;two hex numbers
                BEQ         cont                    ;first one for the output in other
                CMP         R9,#0x7D                ;word rows: 7: First row, B: Second
                MOVEQ       R10,#57                 ;D: Third, E: Fourth
                BEQ         cont
                CMP         R9,#0x7E                ;the second hex number is for the columns
                MOVEQ       R10,#56                 ;7: First and so on
                BEQ         cont                    ;then R10 is loaded with the corresponding
                                                    ;ASCII value of the pressed key
                CMP         R9,#0xB7
                MOVEQ       R10,#55
                BEQ         cont
                CMP         R9,#0xBB
                MOVEQ       R10,#54
                BEQ         cont
                CMP         R9,#0xBD
                MOVEQ       R10,#53
                BEQ         cont
                CMP         R9,#0xBE
                MOVEQ       R10,#52
                BEQ         cont
```

```
154
155                 CMP         R9,#0xD7
156                 MOVEQ       R10,#51
157                 BEQ         cont
158                 CMP         R9,#0xDB
159                 MOVEQ       R10,#50
160                 BEQ         cont
161                 CMP         R9,#0xDD
162                 MOVEQ       R10,#49
163                 BEQ         cont
164                 CMP         R9,#0xDE
165                 MOVEQ       R10,#48
166                 BEQ         cont
167
168                 CMP         R9,#0xE7                    ;CTRL A
169                 MOVEQ       R3,#1
170                 BEQ         start
171                 CMP         R9,#0xEB                    ;CTRL B
172                 MOVEQ       R3,#2
173                 BEQ         start
174                 CMP         R9,#0xED
175                 MOVEQ       R10,#69
176                 BEQ         start
177                 CMP         R9,#0xEE
178                 MOVEQ       R10,#70
179                 BEQ         start
180
181 devam            CMP         R11,#0x70                       ;This small block changes
    rows
182                 MOVEQ       R11,#0xB0                       ;one by one
183                 BEQ         checkrows
184                 CMP         R11,#0xB0
185                 MOVEQ       R11,#0xD0
186                 BEQ         checkrows
187                 CMP         R11,#0xD0
188                 MOVEQ       R11,#0xE0
189                 BEQ         checkrows
190                 CMP         R11,#0xE0
191                 BEQ         start
192
193 cont
194 debnc_out        LDR         R1,=GPIO_PORTB_DATA             ;This debounce part looks for the
195                 LDR         R8,[R1]                         ;relase of the key
196                 AND         R7,R8,#0xF                      ;if it sees an input it loops until
197                 CMP         R7,#0xF                         ;it does not see one.
198                 BNE         debnc_out                       ;It also double checks with a delayed time
199                 BL          delay
200                 LDR         R1,=GPIO_PORTB_DATA
201                 LDR         R9,[R1]
202                 AND         R7,R9,#0xF
203                 CMP         R7,#0xF
204                 BNE         debnc_out
205                 MOV         R5,R10                          ;if everything goes fine code prints the
206                                                             ;key's character since it already holds
207                 NOP                                         ;it as ASCII value
208                 NOP
209                 NOP
210
211                 BL          OutChar
212
213                 B           start                           ;code starts over
214
215                 ENDP
216                 ALIGN
217                 END
218
```