D:\OKUL\4-1\447 LAB\EXP-2\variation\Program_Directives.s

```
GPIO_PORTB_DATA
                                     0x400053FC
                            EOU
                                                       ;data address to all pins
     GPIO_PORTB_DIR
GPIO_PORTB_AFSEL
                                     0x40005400
 2
                            EQU
 3
                            EQU
                                     0x40005420
     GPIO PORTB AMSEL
 4
                            EQU
                                     0x40005428
     GPIO PORTB DEN
                            EQU
                                     0x4000551C
 6
     GPIO_PORTB_PUR
                            EQU
                                     0x40005510
 7
     IOB
                            EQU
                                     0xF0
 8
     PUB
                            EQU
                                     0x0F
 9
                                     0x400FE608
10
     SYSCTL_RCGCGPIO
                            EQU
11
12
                                main, READONLY, CODE
13
                   AREA
14
                   THUMB
15
                   EXTERN
                                delav
                   EXTERN
                                OutChar
17
                   EXPORT
                                main ; Make available
18
19
      main
                   PROC
20
                                R1, =SYSCTL_RCGCGPIO
                   LDR
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]
                                                                                                    OUTPUTS
31
                                R0,#0xFF
                                                                                  ; INPUTS
                   BIC
32
                   ORR
                                RO, #IOB
                                                                             ;r1 pb3
                                                                                               11
     pb7
                                                                                                    pb6
33
                   STR
                                R0,[R1]
                                                                                               12
                                                                             ;r2 pb2
34
                                                                             ;r3 pb1
                                                                                               13
                                                                                                    pb5
35
                   LDR
                                R1, =GPIO_PORTB_AFSEL
                                                                             ;r4 pb0
                                                                                               14
                                                                                                    pb4
36
                   LDR
                                R0,[R1]
37
                   BIC
                                R0,#0xFF
38
                   STR
                                R0, [R1]
39
40
                   LDR
                                R1,=GPIO PORTB DEN
41
                   LDR
                                R0, [R1]
                   MOV
                                R0,#0xFF
42
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
                                RO, #PUB
52
                   STR
                                R0, [R1]
53
                   MOV
                                R3, #0x1
54
55
     start
                  MOV
                                R11, #0x70
56
                   LDR
                                R1, =GPIO PORTB DATA
     checkrows
57
                   LDR
                                R0, [R1]
58
                   BIC
                                R0,#0xFF
59
                   ORR
                                R0, R11
60
                   STR
                                R0, [R1]
61
62
     debnc inp
                                R1,=GPIO PORTB DATA
                                                                         ; Debounce algorithm for pressing
                   LDR
63
                                R10,[R1]
                   LDR
                                                                         ; wait a delay between two data
64
                   _{\mathrm{BL}}
                                delay
                                                                         ; samples and if they are the same
65
                   LDR
                                R1,=GPIO_PORTB_DATA
                                                                         ;it continues to check columns
66
                   LDR
                                R9,[R1]
                                R9, R10
67
                   CMP
                                                                         ;it loads the data onto R9 reg.
68
                   BEQ
                                check
                                debnc_inp
69
                   В
70
71
     check
                                R3, #0x1
                   CMP
72
                   BEO
                                rows1
73
                   В
                                rows2
74
75
     rows1
                                R9, #0x77
                                                                         ; rows part checks each column
76
                   MOVEO
                                R10,#48
                                                                         ;it starts with the first row
```

77		BEQ.W	cont	;checks if the data in R9 is equal
78		CMP	R9,#0x7B	to any of the 16 keys by simply looking;
79		MOVEQ	R10,#49	;two hex numbers
80		BEQ.W	cont	;first one for the output in other
81			R9,#0x7D	
		CMP	•	;word rows: 7: First row, B: Second
82		MOVEQ	R10,#50	;D: Third, E: Fourth
83		BEQ.W	cont	
84		CMP	R9,#0x7E	; the second hex number is for the columns
85		MOVEQ	R10 ,#51	;7: First and so on
86				;then R10 is loaded with the corresponding
		BEQ.W	cont	
87				;ASCII value of the pressed key
88		CMP	R9,#0xB7	
89		MOVEQ	R10,#52	
90		BEQ.W	cont	
91		CMP	R9,#0xBB	
92		MOVEQ	R10,#53	
93		BEQ.W	cont	
94		CMP	R9, #0xBD	
95		MOVEQ	R10,#54	
96		BEQ.W	cont	
97		CMP	R9,#0xBE	
98		MOVEO	R10,#55	
		~		
99		BEQ.W	cont	
100				
101		CMP	R9,#0xD7	
102		MOVEQ	R10,#56	
103		BEQ.W	cont	
104		CMP	R9,#0xDB	
105		MOVEQ	R10,#57	
106			cont	
		BEQ.W		
107		CMP	R9,#0xDD	
108		MOVEQ	R10,#65	
109		BEQ.W	cont	
110		CMP	R9,#0xDE	
111		MOVEQ	R10,#66	
112		BEQ.W	cont	
113				
		CLUD	DO 110 D7	
114		CMP	R9,#0xE7	
115		MOVEQ	R3,#1	
116		BEQ	start	
117		CMP	R9,#0xEB	
118		MOVEQ	R3,#2	
119		BEQ	start	
120		CMP	R9,# <mark>0xED</mark>	
121		MOVEQ	R10,#69	
122		BEQ	start	
123		CMP	R9,# <mark>0xEE</mark>	
124		MOVEQ	R10,# <mark>70</mark>	
125		BEQ.W	start	
126		В	devam	
		D	de vaiii	
127				
128				
		CMD	DO #077	
129	rows2	CMP	R9, #0x77	;rows part checks each column
130		MOVEQ	R10,#66	;it starts with the first row
131		BEQ	cont	; checks if the data in R9 is equal
				· · · · · · · · · · · · · · · · · · ·
132		CMP	R9 , #0x65	;to any of the 16 keys by simply looking
133		MOVEQ	R10,#65	;two hex numbers
134		BEQ	cont	;first one for the output in other
135		CMP	R9, #0x7D	;word rows: 7: First row, B: Second
136		MOVEQ	R10,#57	;D: Third, E: Fourth
137				, = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 =
		BEQ	cont	
138		CMP	R9,#0x7E	;the second hex number is for the columns
139		MOVEQ	R10,#56	;7: First and so on
				
140		BEQ	cont	; then R10 is loaded with the corresponding
141				;ASCII value of the pressed key
142		CMD	DQ #0vD7	
		CMP	R9,#0xB7	
143		MOVEQ	R10,# <mark>55</mark>	
144		BEQ	cont	
145		CMP	R9,#0xBB	
146		MOVEQ	R10,#54	
147				
		BEQ	cont	
148		CMP	R9 , # 0 xBD	
149		MOVEQ	R10,#53	
			•	
150		BEQ	cont	
151		CMP	R9,#0xBE	
152		MOVEQ	R10,#52	
153		BEQ	cont	

D:\OKUL\4-1\447 LAB\EXP-2\variation\Program_Directives.s 154 155 CMP R9, #0xD7156 MOVEQ R10, #51 157 cont BEO 158 R9, #0xDB159 MOVEQ R10, #50 160 BEQ cont 161 CMP R9, #0xDD162 MOVEQ R10,#49 163 BEQ cont 164 CMP R9, #0xDE165 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, #0xED175 MOVEQ R10,#69 176 BEQ start 177 CMP R9, #0xEE178 MOVEQ R10, #70 179 BEQ start 180 R11,#0x70 181 CMP ;This small block changes devam rows 182 MOVEQ R11, #0xB0 ; one by one 183 checkrows **BEO** 184 CMP R11, #0xB0 185 MOVEQ R11,#0xD0 186 checkrows BEO 187 R11,#0xD0 CMP 188 MOVEQ R11,#0xE0 189 BEQ checkrows 190 CMPR11, #0xE0 191 BEQ start 192 193 194 ; This debounce part looks for the debnc_out LDR R1, =GPIO_PORTB_DATA 195 ; relase of the key LDR R8, [R1] 196 AND R7,R8,#0xF; if it sees an input it loops until 197 CMP R7, #0xF;it does not see one. 198 BNE ;It also double checks with a delayed time debnc out 199 $_{ m BL}$ delay 200 LDR R1,=GPIO_PORTB_DATA 201 LDR R9, [R1] 202 R7,R9,#0xFAND R7, #0xF203 CMP 204 BNE debnc out $R5,R1\overline{0}$ 205 MOV ; if everything goes fine code prints the

; key's character since it already holds

;it as ASCII value

; code starts over

206

207

208

209

210 211

212 213

214 215

216

217

218

NOP

NOP

NOP

BL

В

ENDP

END

ALIGN

OutChar

start