

به نام خدا

«آزمایشگاه ریز پردازنده و زبان اسمبلی»

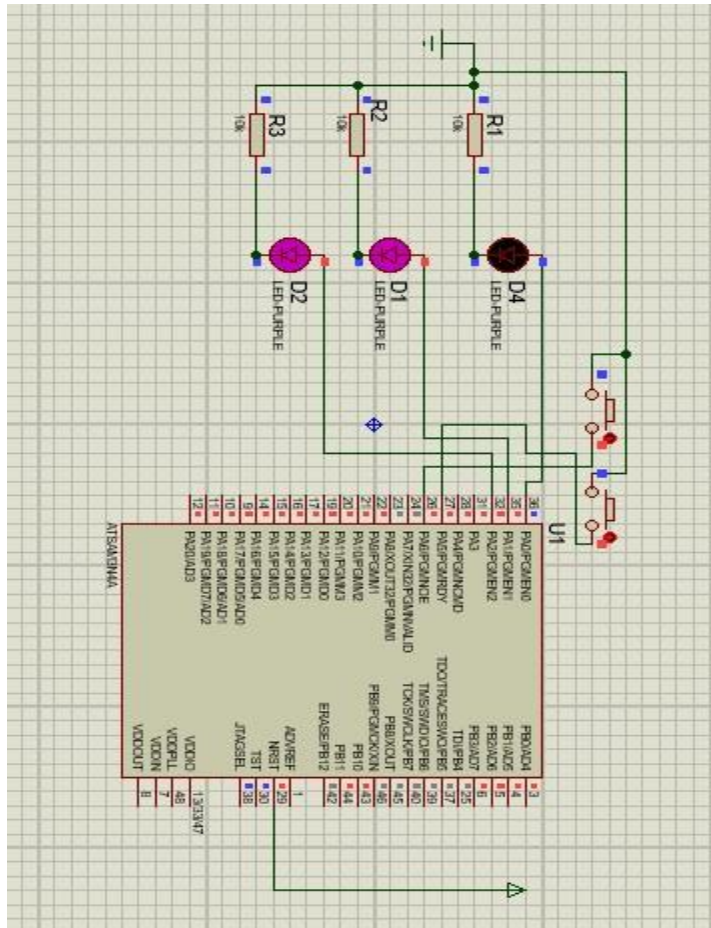
## آزمایش اسمبلی-۱

اعضای گروه: هلیاسادات هاشمی پور ۹۸۳۱۱۰۶-روژینا کاشفی ۹۸۳۱۱۱۸

### شرح آزمایش

در این آزمایش ما از ۳ LED و دو باتن استفاده کرده ایم که در ابتدا هر سه LED خاموش هستند حال اگر دکمه نخست فشرده شود هر سه LED روشن شده با شروع به چشمک زدن می کنند با این تفاوت که از چپ به راست سرعت چشمک زدن آن ها افزایش می یابد. برای این که این بخش را داشته باشیم با توجه به کد زده شده و اعمال تاخیر، LED نخست سه بار روشن شده و LED دوم دوبار و LED سوم یک بار روشن می شود. اگر دکمه دوم فشرده شود LEDها خاموش می شوند.

مدار پیاده سازی شده به صورت زیر است.



## کد

در کد نوشته شده توضیحات مختصری در مورد نحوه پیاده سازی و کارکرد دستورات استفاده شده با استفاده از

کامنت هایی که نوشته شده است، دادیم.

```
1  PIO_PER EQU 0x400E0E00
2  PIO_SODR EQU 0x400E0E30
3  PIO_OER EQU 0x400E0E10
4  PIO_PDSR EQU 0x400E0E3C
5  PIO_ISR EQU 0x400E0E4C
6  PIO_CODR EQU 0x400E0E34
7
8
9      AREA myLEDBlinkCode, CODE, READONLY
10     EXPORT __main
11     entry
12
```

ثبات های مورد استفاده(با

استفاده از دستور ) EQU و نوع

کد مشخص شده است.

```
12     __main
13         BL EnablePio_1 ;branch and link =>call EnablePio_1
14         BL EnablePio_2 ;branch and link =>call EnablePio_2
15         BL EnablePio_3 ;branch and link =>call EnablePio_3
```

سه لیبل صدا زده شده اند.

از دستور BL

استفاده کرده ایم.

```
17 BEGIN
18     LDR R1, =PIO_ISR ;R1=PIO_ISR
19     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
20     CMP R2,#2_100000 ;compare for chooshing which button is pressed
21     BEQ loop ;branch if equal =>Z=1
22     B BEGIN ;if Z=0
23
24
25 Finish
26     BL LEDOff_1 ;branch and link =>call LEDOff_1
27     BL LEDOff_2 ;branch and link =>call LEDOff_2
28     BL LEDOff_3 ;branch and link =>call LEDOff_3
29     B BEGIN ;branch the begin
30
```

آدرس PIO\_ISR در رجیستری می ریزیم و چک

می کنیم که با وصل بودن دکمه شروع، حلقه اصلی

که مربوط به چشمک زدن می باشد، اجرا شود. در

حلقه ای بسته شدن کلید مربوط به خاموش شدن

چراغ ها را چک می کنیم و در صورت اتصال، به

بخش Finish رفته و با چک کردن قطع بودن، به

Begin بر می گردد در اصل کلید مربوط به شروع

دوباره اجرای برنامه، از سر گرفته می شود.

```

32 loop
33     LDR R1, =PIO_ISR ;R1=PIO_ISR
34     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
35     CMP R2, #2_1000000
36     BEQ Finish ;branch if equal =>Z=1
37
38     ;if Z=0
39     BL LEDOn_1 ;branch and link =>call LEDOn_1
40     BL LEDOn_2 ;branch and link =>call LEDOn_2
41     BL LEDOn_3 ;branch and link =>call LEDOn_3
42
43     BL Delay ;call Delay
44
45     LDR R1, =PIO_ISR ;R1=PIO_ISR
46     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
47     CMP R2, #2_1000000
48     BEQ Finish ;branch if equal =>Z=1
49
50     ;if Z=0
51     BL LEDOff_1 ;branch and link =>call LEDOff_1
52     BL LEDOn_2 ;branch and link =>call LEDOn_2
53     BL LEDOn_3 ;branch and link =>call LEDOn_3
54     BL Delay ;call Delay
55
56     LDR R1, =PIO_ISR ;R1=PIO_ISR
57     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
58     CMP R2, #2_1000000
59     BEQ Finish ;branch if equal =>Z=1
60
61     BL LEDOn_1 ;branch and link =>call LEDOn_1
62     BL LEDOff_2 ;branch and link =>call LEDOff_2
63     BL LEDOn_3 ;branch and link =>call LEDOn_3
64     BL Delay ;call Delay
65
66     LDR R1, =PIO_ISR ;R1=PIO_ISR
67     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
68     CMP R2, #2_1000000
69     BEQ Finish ;branch if equal =>Z=1
70
71     BL LEDOff_1 ;branch and link =>call LEDOff_1
72     BL LEDOff_2 ;branch and link =>call LEDOff_1
73     BL LEDOff_3 ;branch and link =>call LEDOff_3
74     BL Delay ;call Delay
75
76     LDR R1, =PIO_ISR ;R1=PIO_ISR
77     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
78     CMP R2, #2_1000000
79     BEQ Finish ;branch if equal =>Z=1
80
81     BL LEDOn_2 ;branch and link =>call LEDOn_2
82     BL LEDOn_1 ;branch and link =>call LEDOn_1
83     BL LEDOff_3 ;branch and link =>call LEDOff_3
84     BL Delay ;call Delay
85

```

با چند تاخیر، سرعت چشمک زدن چراغ ها را کنترل می کنیم(طبق خواسته دستور کار) در هر بار اجرای حلقه، چراغ اول به اندازه سه بار چشمک می زند، چراغ دوم ما چشمک زده و چراغ سومی یک بار چشمک می زند. هر بار هم وصل بودن کلید مربوط به خاموش شدن را چک می کنیم.

```

86     LDR R1, =PIO_ISR ;R1=PIO_ISR
87     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
88     CMP R2, #2_1000000
89     BEQ Finish ;branch if equal =>Z=1
90
91     BL LEDOff_1 ;branch and link =>call LEDOff_1
92     BL LEDOn_2 ;branch and link =>call LEDOn_2
93     BL LEDOff_3 ;branch and link =>call LEDOff_3
94     BL Delay ;call Delay
95
96     LDR R1, =PIO_ISR ;R1=PIO_ISR
97     LDR R2, [R1] ; R2 be loaded with location of 0x400E0E4C
98     CMP R2, #2_1000000 ;branch if equal =>Z=1
99
100    BEQ Finish
101    BL LEDOff_1 ;branch and link =>call LEDOff_1
102    BL LEDOff_2 ;branch and link =>call LEDOff_2
103    BL LEDOff_3 ;branch and link =>call LEDOff_3
104
105    BL loop ;return to loop

```

```

97 EnablePio_1
98     MOV R4, #2_1 ;mov the binary number of the specific led into the register
99     LDR R5,=PIO_PER ;R5=PIO_PER
100    STR R4, [R5] ;store R4 in location 0x400E0E00
101    LDR R5,=PIO_OER ;R5=PIO_OER
102    STR R4, [R5] ;store R4 in location 0x400E0E10
103    BX LR ;return to caller
104
105 EnablePio_2
106     MOV R4, #2_10 ;R4=10=>binary
107     LDR R5,=PIO_PER ;R5=PIO_PER
108     STR R4, [R5] ;store R4 in location 0x400E0E00
109     LDR R5,=PIO_OER ;R5=PIO_OER
110     STR R4, [R5] ;store R4 in location 0x400E0E10
111     BX LR ;return to caller
112
113 EnablePio_3
114     MOV R4, #2_100 ;R4=100=>binary
115     LDR R5,=PIO_PER ;R5=PIO_PER
116     STR R4, [R5] ;store R4 in location 0x400E0E00
117     LDR R5,=PIO_OER ;R5=PIO_OER
118     STR R4, [R5] ;store R4 in location 0x400E0E10
119
120     BX LR

```

هر سه لیبل با انتقال عدد باینری مربوط  
 به پایه ای که هر کدام از LED ها است را  
 داخل رجیستری می ریزد و سپس آن را  
 درون PIO\_PER و PIO\_OER استور  
 می کنیم و در نهایت به محل فراخوانی  
 باز می گردیم.



```

122 LEDOn_1
123     MOV R4, #2_1 ;R4=1=>binary
124     LDR R5,=PIO_SODR ;R5=PIO_SODR
125     STR R4, [R5] ;store R4 in location 0x400E0E30
126     BX LR ;return to caller
127
128 LEDOn_2
129     MOV R4, #2_10 ;R4=10=>binary
130     LDR R5,=PIO_SODR ;R5=PIO_SODR
131     STR R4, [R5] ;store R4 in location 0x400E0E30
132     BX LR ;return to caller
133
134 LEDOn_3
135     MOV R4, #2_100 ;R4=100=>binary
136     LDR R5,=PIO_SODR ;R5=PIO_SODR
137     STR R4, [R5] ;store R4 in location 0x400E0E30
138     BX LR ;return to caller
139

```

روشن شدن چراغ ها با استفاده از  
رجیستر PIO\_SODR و پایه مربوط  
به LED مربوطه می باشد.

```

140 LEDOff_1
141     MOV R4, #2_1 ;R4=1=>binary
142     LDR R5,=PIO_CODR ;R5=PIO_CODR
143     STR R4, [R5] ;store R4 in location 0x400E0E34
144     BX LR ;return to caller
145
146 LEDOff_2
147     MOV R4, #2_10 ;R4=10=>binary
148     LDR R5,=PIO_CODR ;R5=PIO_CODR
149     STR R4, [R5] ;store R4 in location 0x400E0E34
150     BX LR ;return to caller
151
152 LEDOff_3
153     MOV R4, #2_100 ;R4=100=>binary
154     LDR R5,=PIO_CODR ;R5=PIO_CODR
155     STR R4, [R5] ;store R4 in location 0x400E0E34
156     BX LR ;return to caller
157

```

خاموش شدن چراغ ها با استفاده از  
رجیستر PIO\_CODR و پایه  
مربوط به LED مربوطه می باشد.

```

158 Delay
159     MOV R4, #0 ;R4=0
160     LDR R5, =0x00C0000 ;R5 =0x00C0000
161
162 LoopDelay
163     ADD R4,R4,#1 ;R4=R4+1
164     CMP R4,R5 ;compare R4 with R5
165     BNE LoopDelay
166     BX LR ;return to caller
167
168     END

```

قسمت تاخیر کد ما می باشد که یک شمارنده تعریف کرده ایم که تا مقدار مورد نظر را می شمرد و با این کار موجب توقف برنامه می شود.ر اصل ابتدا در R4 مقدار صفر را قرار می دهیم و در رجیستر R5 هم یک عدد بزرگی قرار می دهیم سپس با استفاده از LoopDelay هر بار یک واحد به رجیستر R4 اضافه کرده و با R5 مقایسه می کنیم اگر برابر نباشند دوباره لوپ را اجرا کرده و این روند تکرار می شود تا مقدار R4 با R5 برابر شود.

## خروجی آزمایش

