

Traffic Light Controller

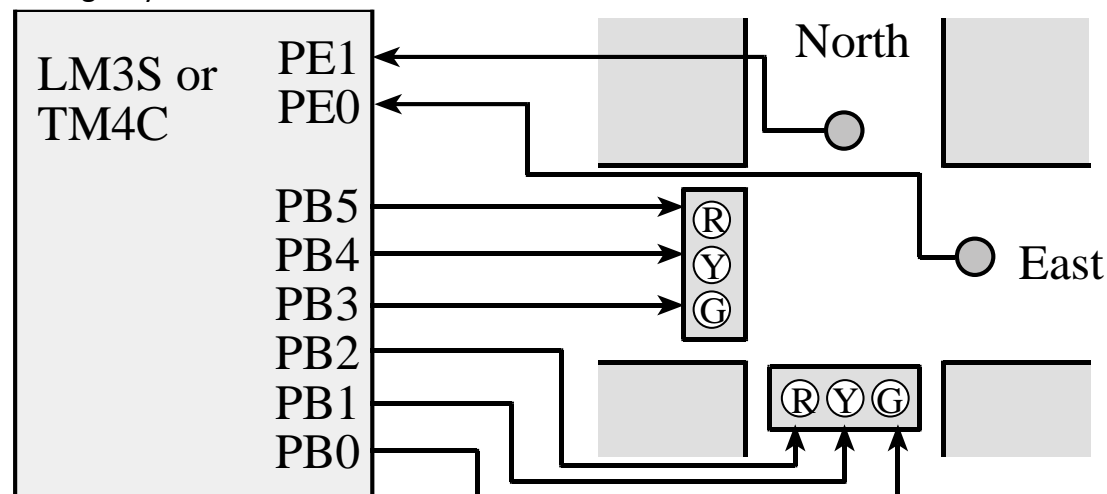
Design a traffic light controller for the intersection of two equally busy one-way streets. The goal is to maximize traffic flow, minimize waiting time at a red light, and avoid accidents.

The intersection has two one-way roads with the same amount of traffic: North and East, as shown in Figure. In this system, the light pattern defines which road has right of way over the other. Since an output pattern to the lights is necessary to remain in a state, we will solve this system with a Moore FSM. It will have three inputs (car sensors on North and East roads and Emergency signal) and seven outputs (one for each light in the traffic signal.) The seven traffic lights are interfaced to Port B bits 6–0, and the three sensors are connected to Port E bits 2–0.

Limitations

- three switches, PE2=1 emergency mode
 - six LEDs, PB0-PB5, as figure, PB6 connected to red LED
- PE2=0, PE1=0, PE0=0 means no cars exist on either road
 PE2=0, PE1=0, PE0=1 means there are cars on the East road
 PE2=0, PE1=1, PE0=0 means there are cars on the North road
 PE2=0, PE1=1, PE0=1 means there are cars on both roads

PE2=0=1XX, means emergency mode, cars do not enter in both directions. PB0-PB6 =0, emergency led PB6=1.



goN, PB6-0 = 0100001 makes it green on North and red on East
waitN, PB6-0 = 0100010 makes it yellow on North and red on East
goE, PB6-0 = 0001100 makes it red on North and green on East
waitE, PB6-0 = 0010100 makes it red on North and yellow on East
Emer, PB6-0 = 1000000 makes emergency led red on.

Note:

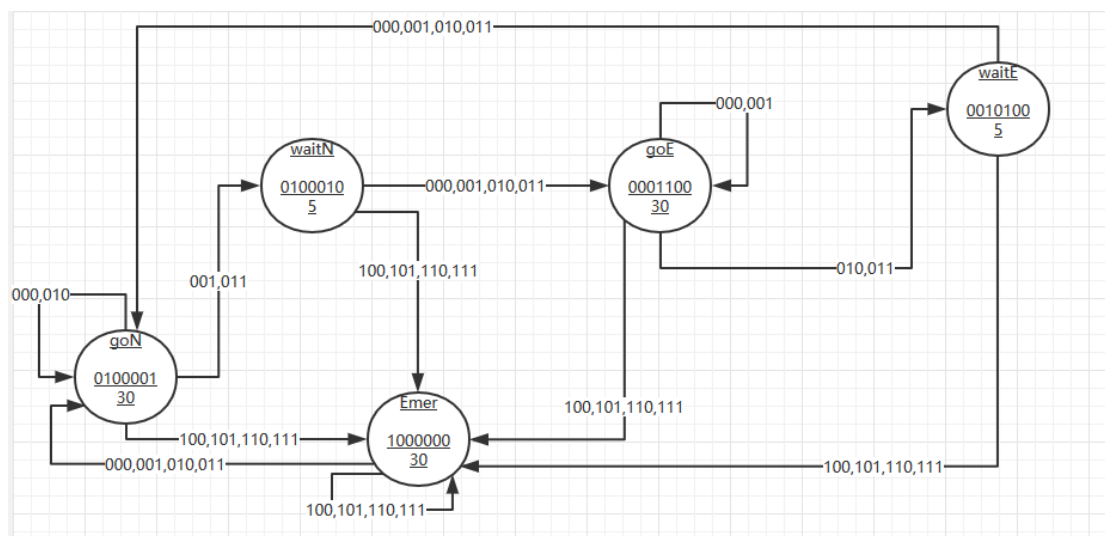
To add more complexity, we simply increase the size of the fsm[] structure, define the Out, Time, and Next pointers, increase the size of Next[8]. size = 2**(number of inputs)

设计思路：

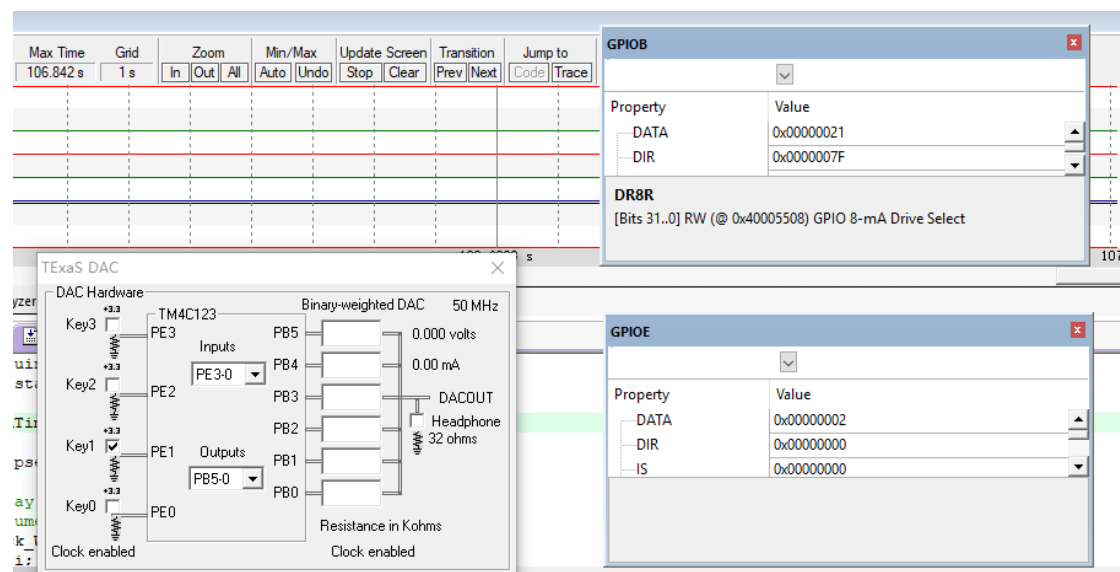
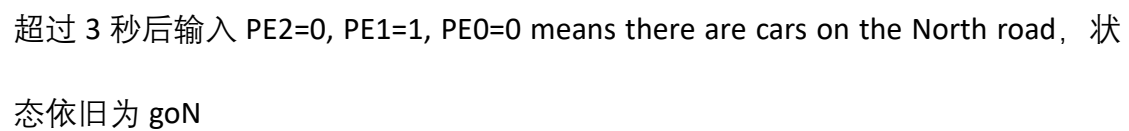
此实验在原来的基础上, 增加了 Emer 状态, 导致增加一个输入位和一个输出位, 用来表示紧急状态产生条件和输出紧急状态指示, 通过在 SENSOR (输入) 中增加一位 PE2 (输入 1) 表示紧急状态产生条件, 在 LIGHT (输出) 中增加一位 PB6 驱动一个红灯 (输出 1) 来输出紧急状态指示。

新的状态图如下, 比原来的实验做出了如下改变：

1. 任何状态下, 只要接收到紧急信号, 则下一状态进入紧急状态
2. 紧急状态保持 30 秒 (此处指应用在现实中 30 秒, 程序中为了加快状态的转变, 方便观察, 只停留 3 秒)
3. 在紧急状态下, 如果没有接收到紧急信号, 则下一状态进入初始状态 goN



开始调试，进入初始状态，即 goN，此时 PE2=0, PE1=0, PE0=0 means no cars exist on either road，输出 0100001，makes it green on North and red on East



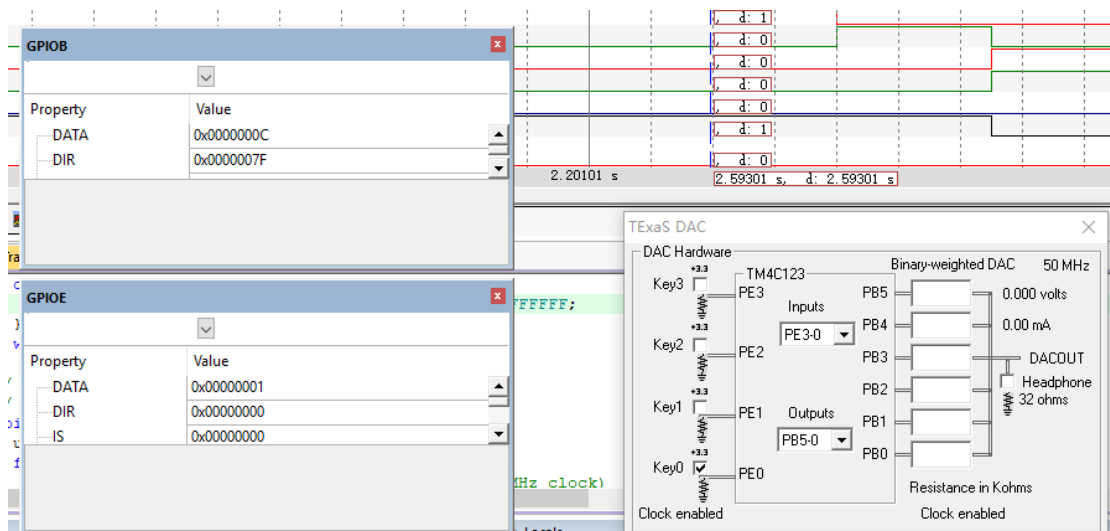
The screenshot displays the TMS320C64x DSP software interface. The top section shows the 'Max Time' (144.1137 s) and 'Grid' (1 s). Below this is a table with columns for 'Zoom', 'Min/Max', 'Update Screen', 'Transition', and 'Jump to'. The 'Zoom' column has sub-columns for 'In', 'Out', and 'All'. The 'Min/Max' column has sub-columns for 'Auto', 'Undo', 'Stop', and 'Clear'. The 'Transition' column has sub-columns for 'Prev', 'Next', 'Code', and 'Trace'. The 'Jump to' column has sub-columns for 'Prev', 'Next', 'Code', and 'Trace'.

The 'GPIOB' configuration window is open, showing the 'Property' (DATA, DIR) and 'Value' (0x00000022, 0x0000007F) settings. The 'DR8R' (Bits 31..0) RW (@ 0x40005508) GPIO 8-mA Drive Select is also visible.

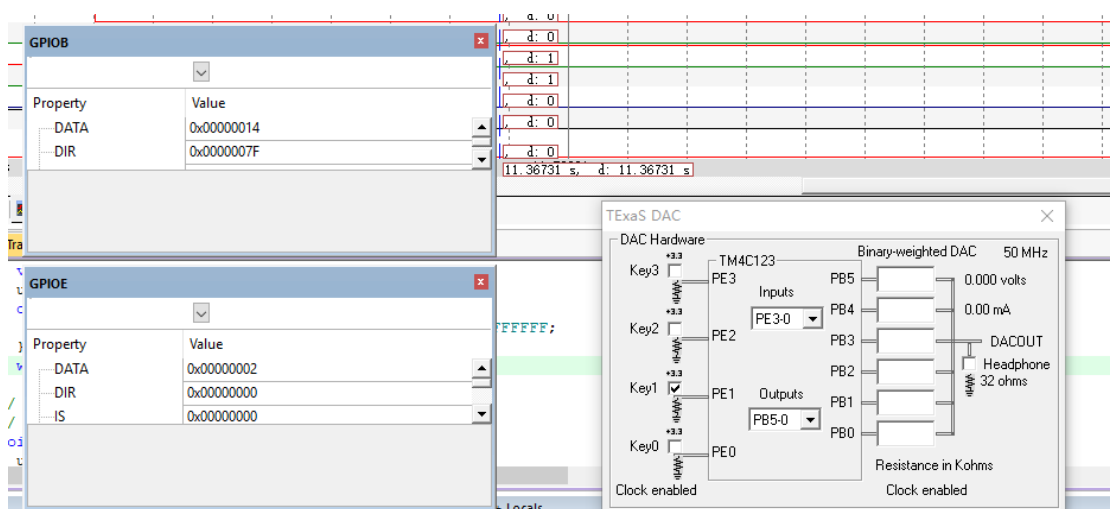
The 'TMS320C64x DAC' configuration window is open, showing the 'DAC Hardware' (TM4C123) and 'Binary-weighted DAC' (50 MHz) settings. The 'DACOUT' is set to 'Headphone 32 ohms'. The 'Resistance in Kohms' is set to 'Clock enabled'.

The 'GPIOE' configuration window is open, showing the 'Property' (DATA, DIR, IS) and 'Value' (0x00000001, 0x00000000, 0x00000000) settings.

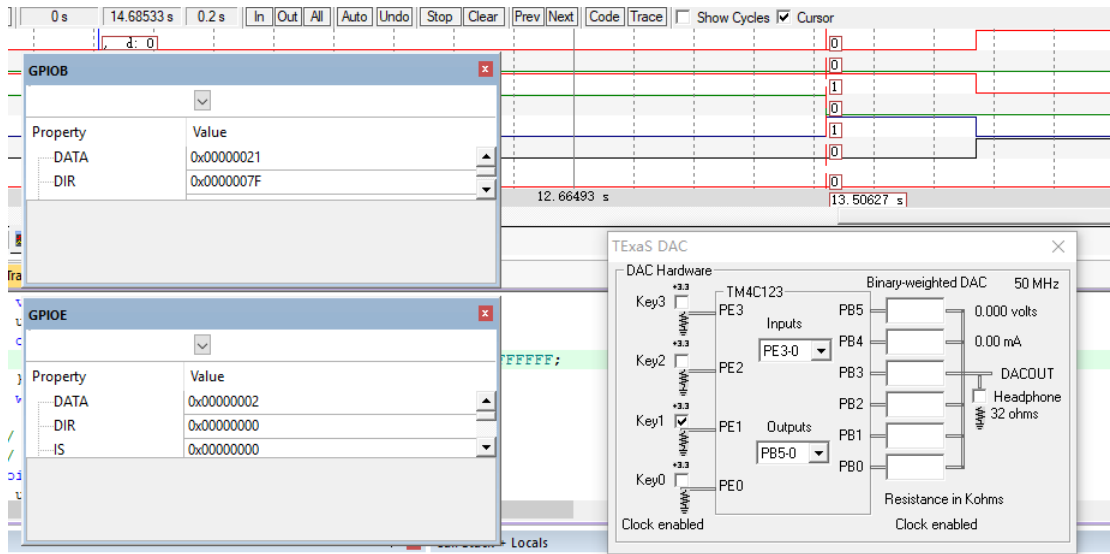
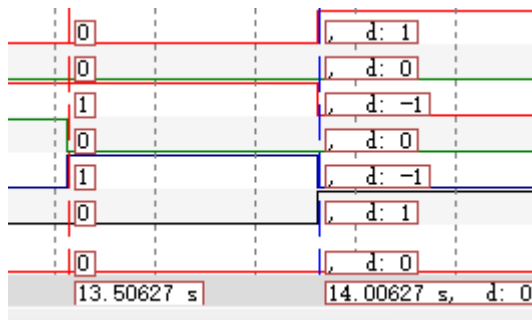
Index	Code	Trace	Show Cycles	Cursor
	0			d: 0
	1			d: -1
	0			d: 1
	0			d: 1
	0			d: 0
	1			d: -1
	0			d: 0
8.9133 s	144.0133 s			144.5173 s, d: 0.504 s



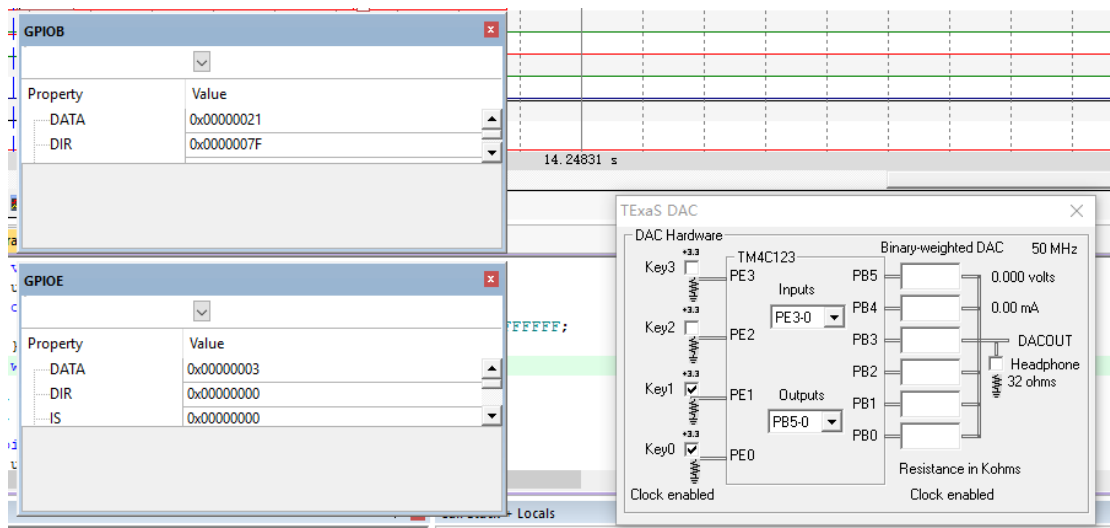
3 秒后输入 PE2=0, PE1=1, PE0=0 means there are cars on the North road, 状态变为 waitE, 输出 0010100, makes it red on North and yellow on East



保持输入 PE2=0, PE1=1, PE0=0 means there are cars on the North road, 0.5 秒后状态变为 goN, 输出 0100001, makes it green on North and red on East



输入 PE2=0, PE1=1, PE0=1 means there are cars on both roads, 此时状态在 goN, waitN, goE, waitE 之间循环, 间隔时间分别为上一阶段停留的时间。



GPIOB

Property	Value
DATA	0x00000022
DIR	0x0000007F

15.14106 s

GPIOE

Property	Value
DATA	0x00000003
DIR	0x00000000
IS	0x00000000

TEaS DAC

DAC Hardware

TM4C123

Key3 PE3

Key2 PE2

Key1 PE1

Key0 PE0

Inputs

PE3-0

Outputs

PB5-0

Binary-weighted DAC

50 MHz

0.000 volts

0.00 mA

DACOUT

Headphone

32 ohms

Resistance in Kohms

Clock enabled

GPIOB

Property	Value
DATA	0x0000000C
DIR	0x0000007F

15.76796 s

GPIOE

Property	Value
DATA	0x00000003
DIR	0x00000000
IS	0x00000000

TEaS DAC

DAC Hardware

TM4C123

Key3 PE3

Key2 PE2

Key1 PE1

Key0 PE0

Inputs

PE3-0

Outputs

PB5-0

Binary-weighted DAC

50 MHz

0.000 volts

0.00 mA

DACOUT

Headphone

32 ohms

Resistance in Kohms

Clock enabled

Name	Location/Value	Type
SysTick_Wait	0x00000360	void f(unsigned int)

GPIOB

Property	Value
DATA	0x00000014
DIR	0x0000007F

16.555 s

GPIOE

Property	Value
DATA	0x00000003
DIR	0x00000000
IS	0x00000000

TEaS DAC

DAC Hardware

TM4C123

Key3 PE3

Key2 PE2

Key1 PE1

Key0 PE0

Inputs

PE3-0

Outputs

PB5-0

Binary-weighted DAC

50 MHz

0.000 volts

0.00 mA

DACOUT

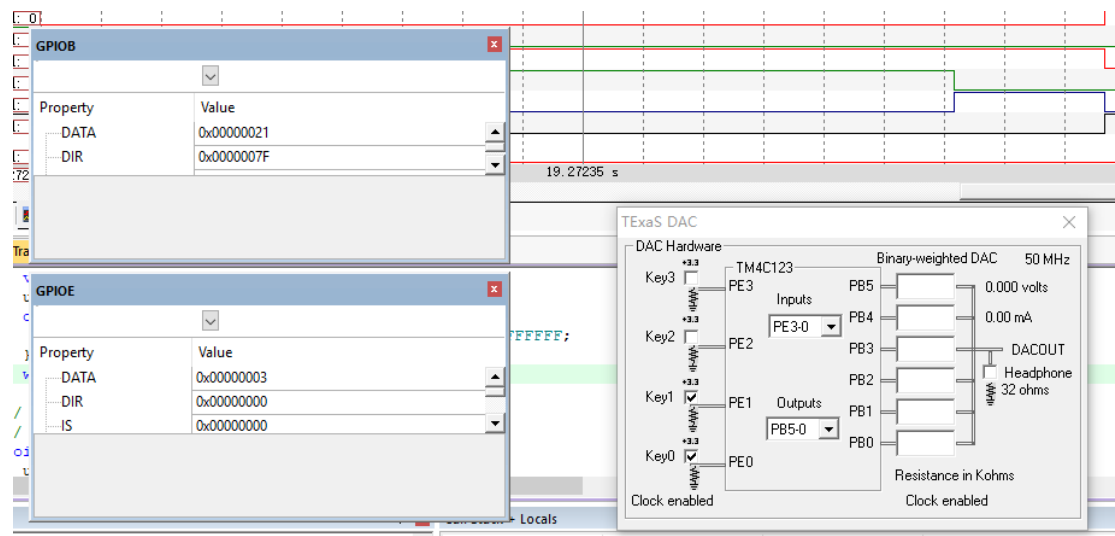
Headphone

32 ohms

Resistance in Kohms

Clock enabled

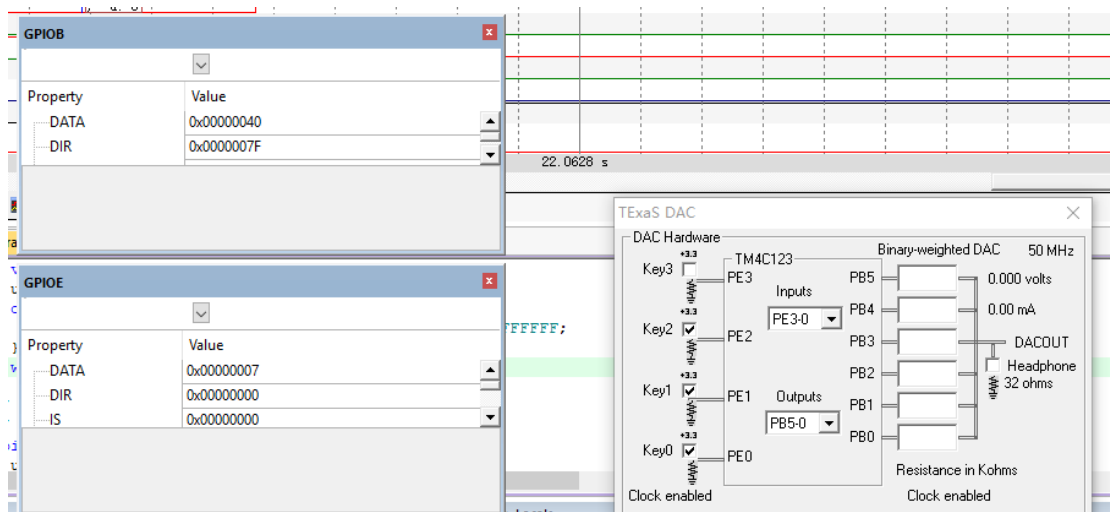
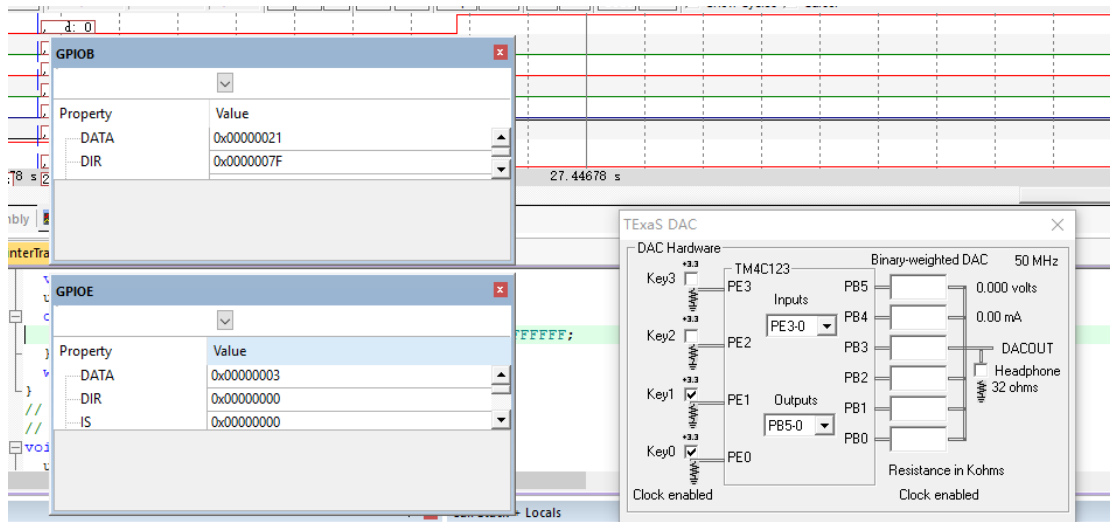
Name	Location/Value	Type
SysTick_Wait	0x00000360	void f(unsigned int)



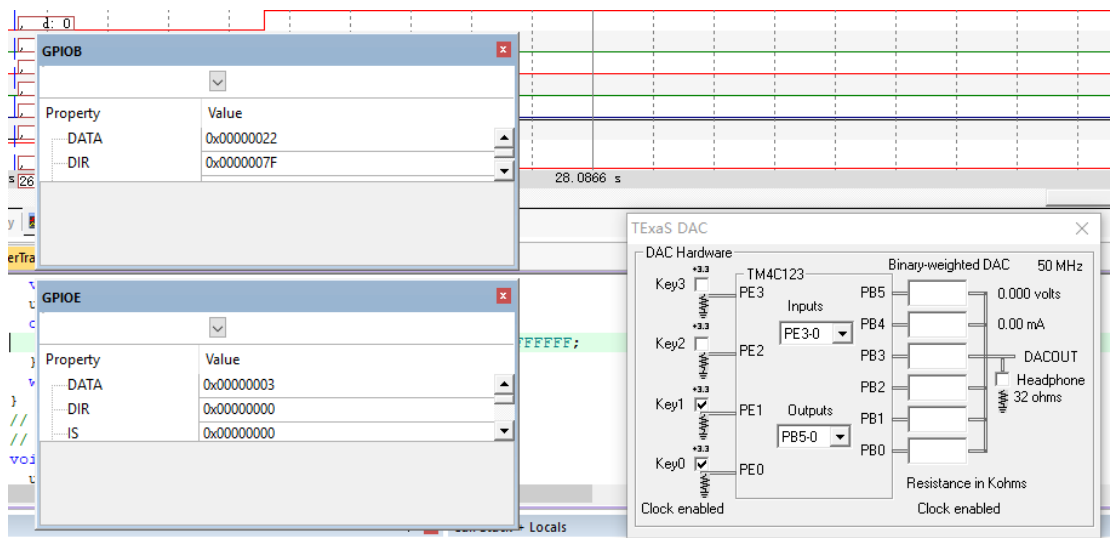
至此，已测试原来实验的所有内容，全部符合要求。下面测试本实验增加的紧急状态。

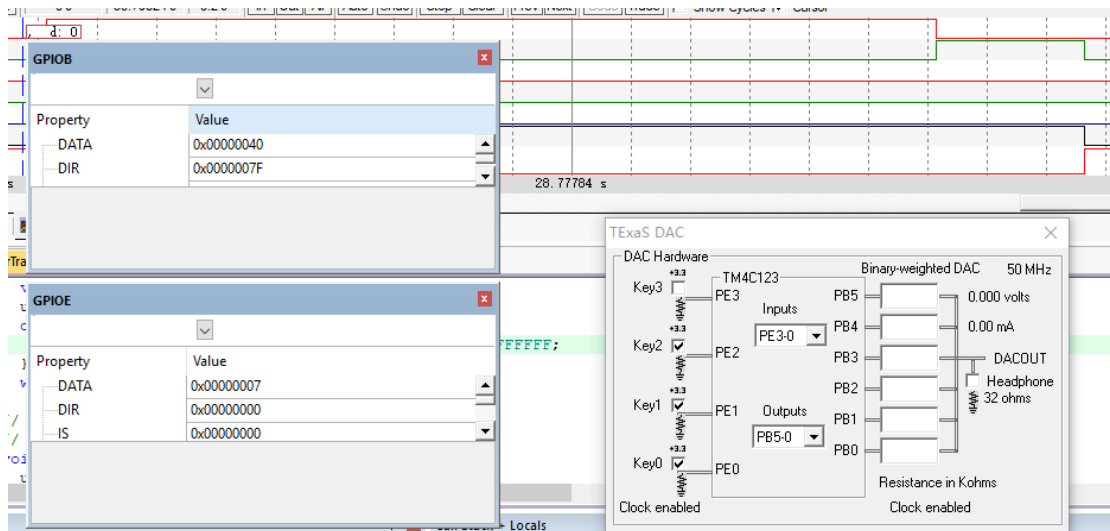
测试方法：保持 $PE1=1, PE0=1$ ，使状态在 `goN`，`waitN`，`goE`，`waitE` 之间循环，然后在四个状态之中的每一个状态下，输入 $PE2=1$ ，观察是否在相应的状态保留时间之后，跳转至紧急状态。测试完最后的 `waitE` 跳入紧急状态后，输入 $PE2=0$ ，观察 3 秒后是否从紧急状态跳入初始状态，即 `goN`

从 `goN` 到紧急状态

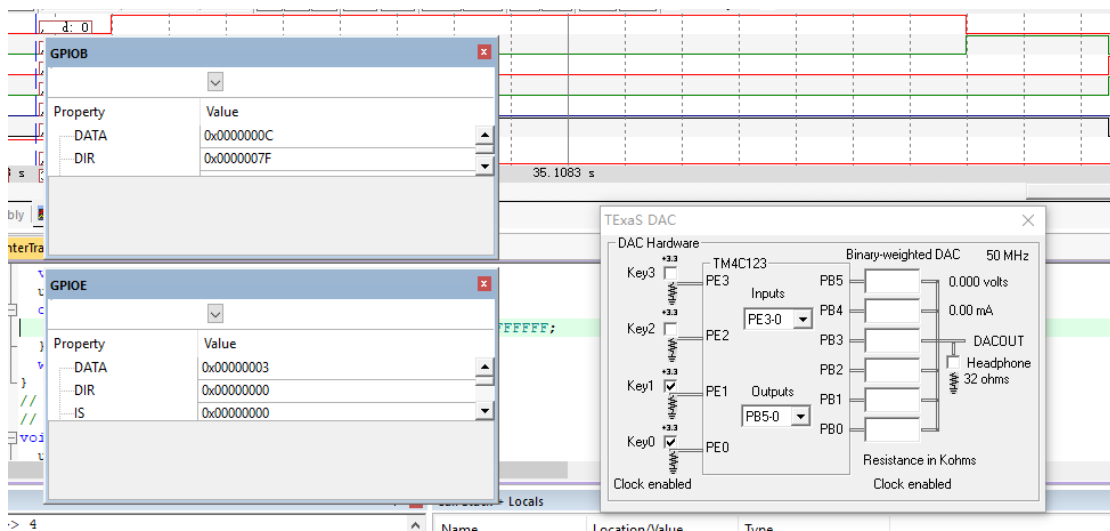


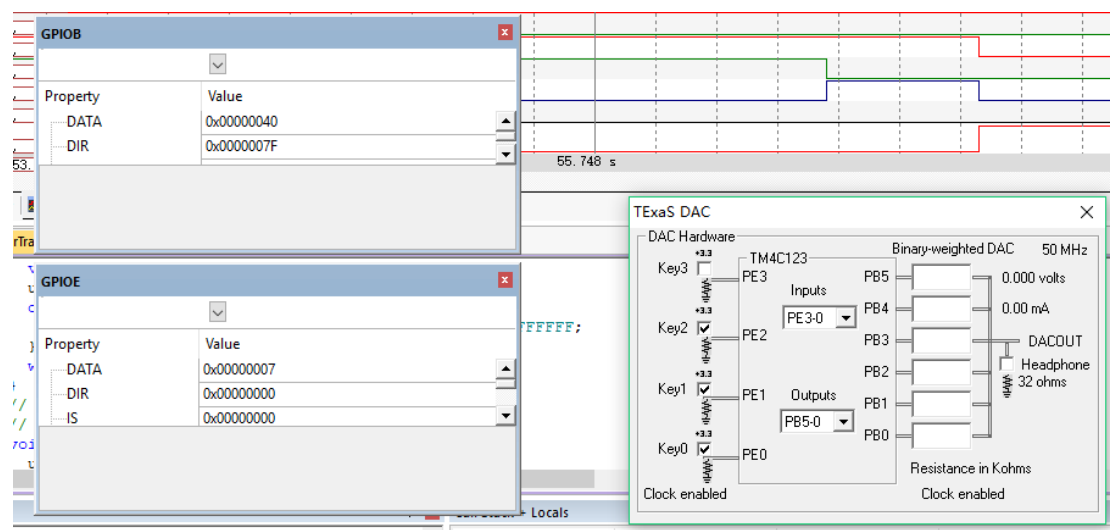
从 waitN 到紧急状态



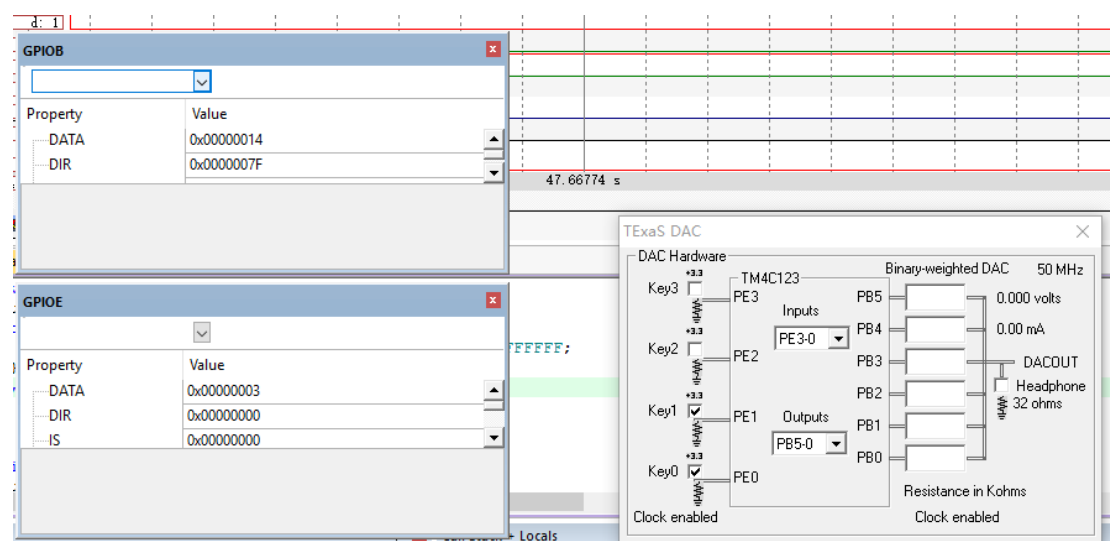


从 goE 到紧急状态





从 waitE 到紧急状态



The screenshot shows a software interface with two register windows and a DAC configuration window. The GPIOB window displays DATA as 0x00000040 and DIR as 0x0000007F. The GPIOE window displays DATA as 0x00000007, DIR as 0x00000000, and IS as 0x00000000. The TExaS DAC window is configured for a TM4C123 with a 50 MHz clock. It shows a binary-weighted DAC with inputs PE3-0 and outputs PB5-0. The DACOUT is connected to a Headphone with 32 ohms resistance. The DAC is currently outputting 0.000 volts and 0.00 mA.

Property	Value
DATA	0x00000040
DIR	0x0000007F

Property	Value
DATA	0x00000007
DIR	0x00000000
IS	0x00000000

TExaS DAC Configuration:

- DAC Hardware: TM4C123
- Binary-weighted DAC: 50 MHz
- Inputs: PE3-0
- Outputs: PB5-0
- DACOUT: Headphone 32 ohms
- Resistance in Kohms: 32
- Clock enabled: Yes

从紧急状态到初始状态 goN

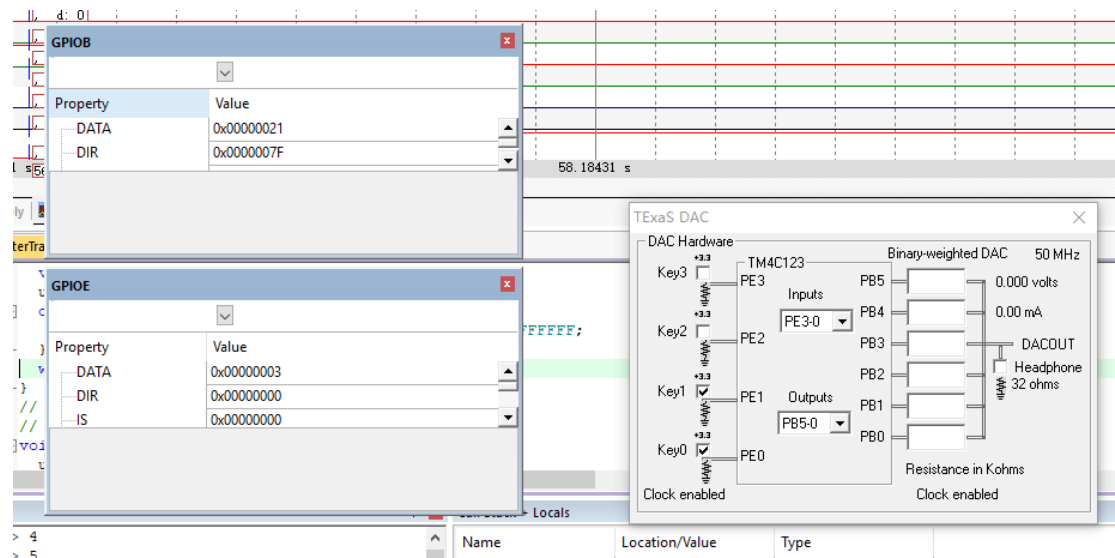
This screenshot is identical to the one above, showing the same GPIOB and GPIOE register values and the TExaS DAC configuration. The DAC configuration window is highlighted with a green border, indicating it is the active window. The DAC is still outputting 0.000 volts and 0.00 mA.

Property	Value
DATA	0x00000040
DIR	0x0000007F

Property	Value
DATA	0x00000003
DIR	0x00000000
IS	0x00000000

TExaS DAC Configuration:

- DAC Hardware: TM4C123
- Binary-weighted DAC: 50 MHz
- Inputs: PE3-0
- Outputs: PB5-0
- DACOUT: Headphone 32 ohms
- Resistance in Kohms: 32
- Clock enabled: Yes



至此，已完成所有的测试，符合要求