Question 1. What is a thread?

线程是操作系统调度和分派系统资源的最小单位

Question 2. What is the main thread? What are interrupt threads?

一个程序被操作系统启动时，会运行一个线程，就是主线程。

中断线程是使 CPU 暂停当前线程，执行另外一个代码段的线程.

Question 3. What are the five steps that occur automatically (in hardware) as the context switches from the main thread to an interrupt thread?

1. 完成当前指令
2. 挂起当前线程的执行并将八个寄存器的值（R0-R3, R12, LR, PC, PSR)推入堆栈。
3. 将 LR 置为 0XFFFFFFF0
4. IPSR 置为中断号
5. 将 PC 置为 ISR 地址

Question 4. Define the following terms as they relate to interrupts.

Hardware trigger
Interrupt enable bit I in the PRIMASK register
Interrupt enable bit in the NVIC_EN0_R register
Interrupt priority in the NVIC_SYS_PRI3_R or NVIC_PRI1_R register
Interrupt arm bit like bit1 (INTEN) in the NVIC_ST_CTRL_R register
Interrupt vector

1. Hadrware trigger

   硬件设置标志位以请求中断
2. PRIMASK 的 I=1 的时候，大多数中断不被允许，为 0 的时候，中断被允许.
3. 控制中断号在 16-47 之间中断的使能信号。
4. NVIC_SYS_PRI3_R 定义 SysTick,PendSV,Debug 中断源的优 NVIC_PRI1_R 定义 SSI0，UART1，UART0，GPIO Port E 中断源的优先级
5.
6. 中断服务程序的入口地址

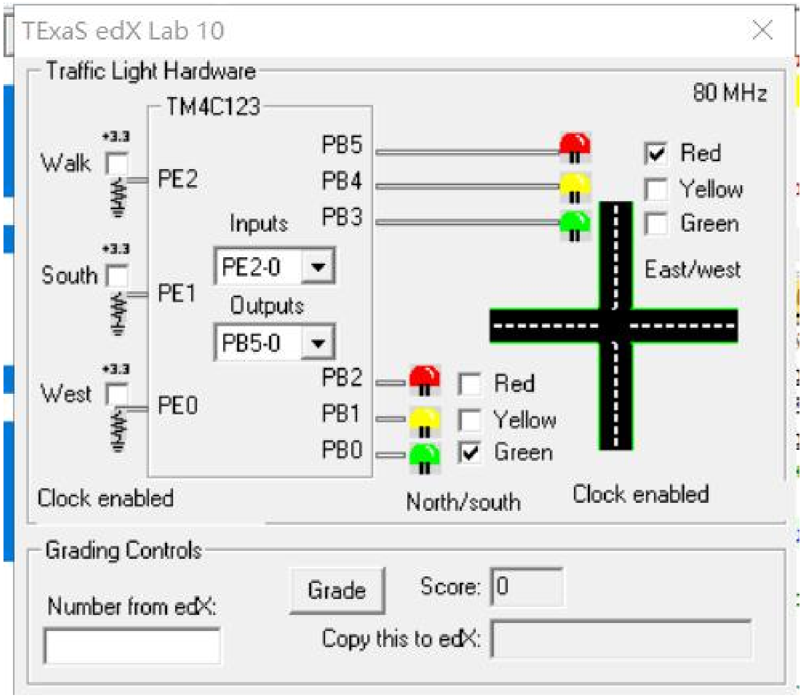
Question 5. What is an interrupt acknowledge? How does the SysTick interrupt get acknowledged and how is SysTick acknowledge different from the other interrupts?

清除中断标志位

Systick 自动确认

Question 6.这周实验交通灯执行现象截图并说明原理。

当两个方向都没有检测到有人/车时，交通灯保持南北方向绿灯



TExaS edX Lab 10

Traffic Light Hardware

80 MHz

TM4C123

Walk □ PE2

South □ PE1

West □ PE0

Inputs: PE2-0

Outputs: PB5-0

PB5 ☑ Red

PB4 □ Yellow

PB3 □ Green

East/west

PB2 □ Red

PB1 □ Yellow

PB0 ☑ Green

North/south

Clock enabled

Clock enabled

Grading Controls

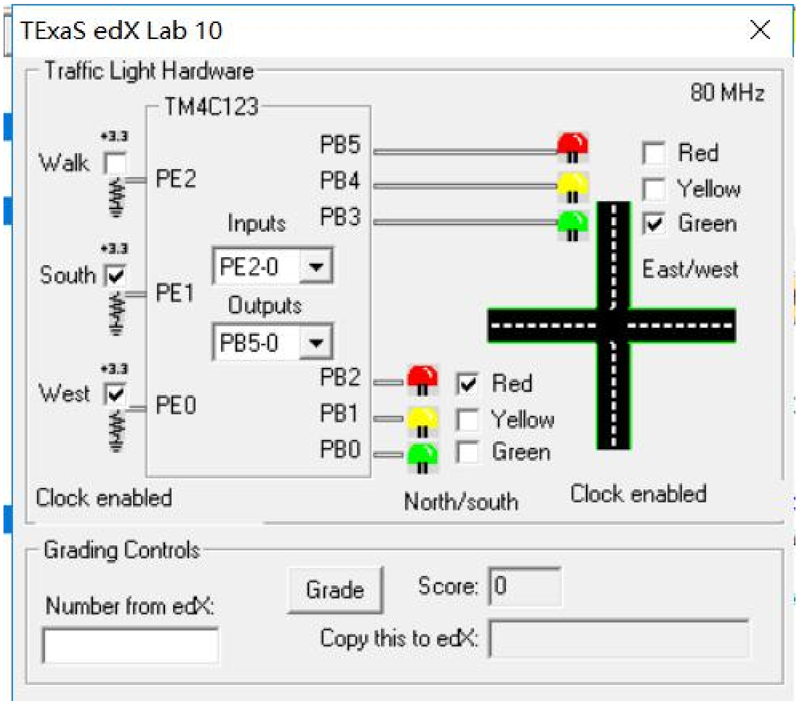
Number from edX:

Score: 0

Copy this to edX:

东西南北两个方向同时有车/人时，交通灯切换

状态 a:东西方向绿灯 30s



TExaS edX Lab 10

Traffic Light Hardware

80 MHz

TM4C123

Walk □ PE2

South ☑ PE1

West ☑ PE0

Inputs: PE2-0

Outputs: PB5-0

PB5 □ Red

PB4 □ Yellow

PB3 ☑ Green

East/west

PB2 ☑ Red

PB1 □ Yellow

PB0 □ Green

North/south

Clock enabled

Clock enabled

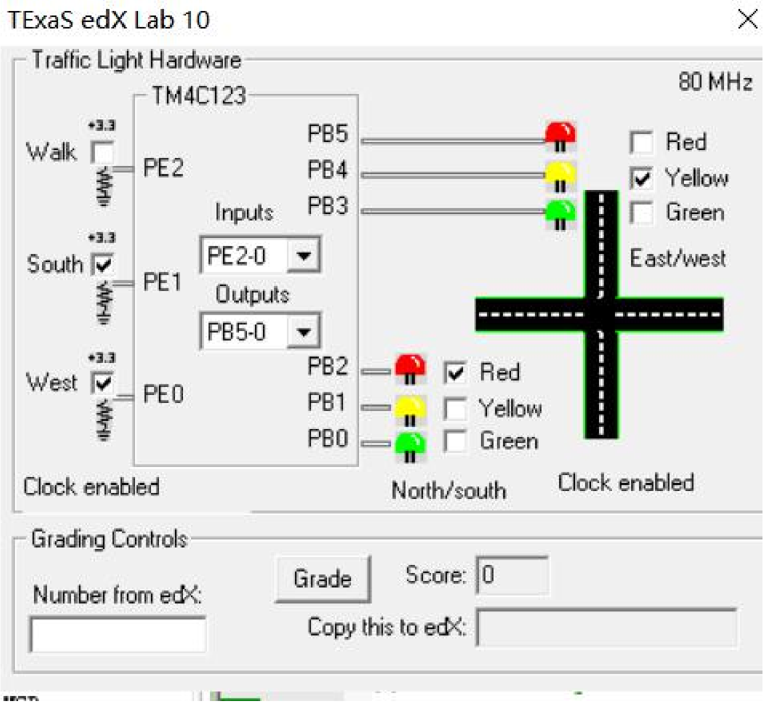
Grading Controls

Number from edX:

Score: 0

Copy this to edX:

状态 b:东西方向黄灯 3s



TExaS edX Lab 10

Traffic Light Hardware

80 MHz

TM4C123

Walk □ PE2

South ☑ PE1

West ☑ PE0

Inputs PE2-0

Outputs PB5-0

PB5 □ Red

PB4 ☑ Yellow

PB3 □ Green

East/west

PB2 ☑ Red

PB1 □ Yellow

PB0 □ Green

North/south

Clock enabled

Clock enabled

Grading Controls

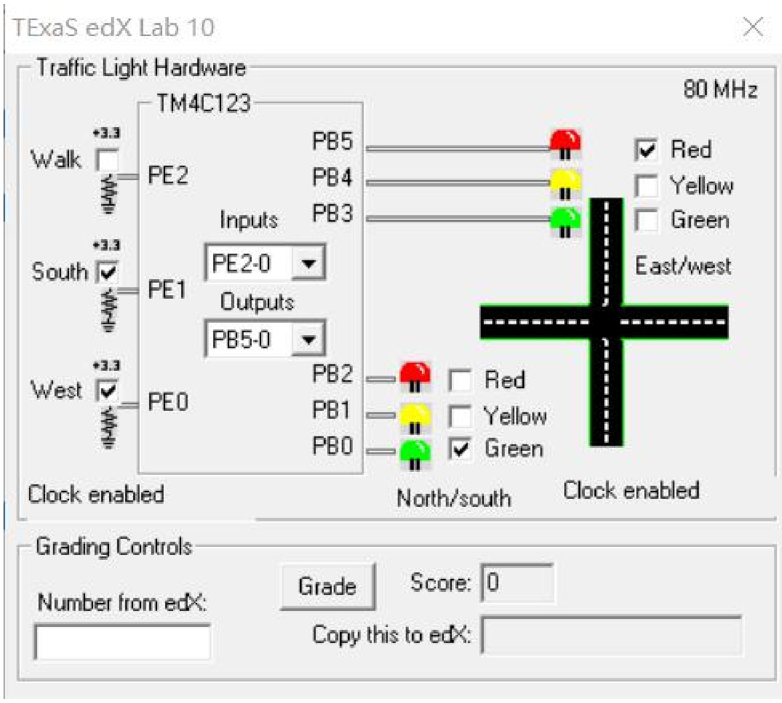
Number from edX:

Grade

Score: 0

Copy this to edX:

状态 c:南北方向绿灯 30s



TExaS edX Lab 10

Traffic Light Hardware

80 MHz

TM4C123

Walk □ PE2

South ☑ PE1

West ☑ PE0

Inputs PE2-0

Outputs PB5-0

PB5 ☑ Red

PB4 □ Yellow

PB3 □ Green

East/west

PB2 □ Red

PB1 □ Yellow

PB0 ☑ Green

North/south

Clock enabled

Clock enabled

Grading Controls

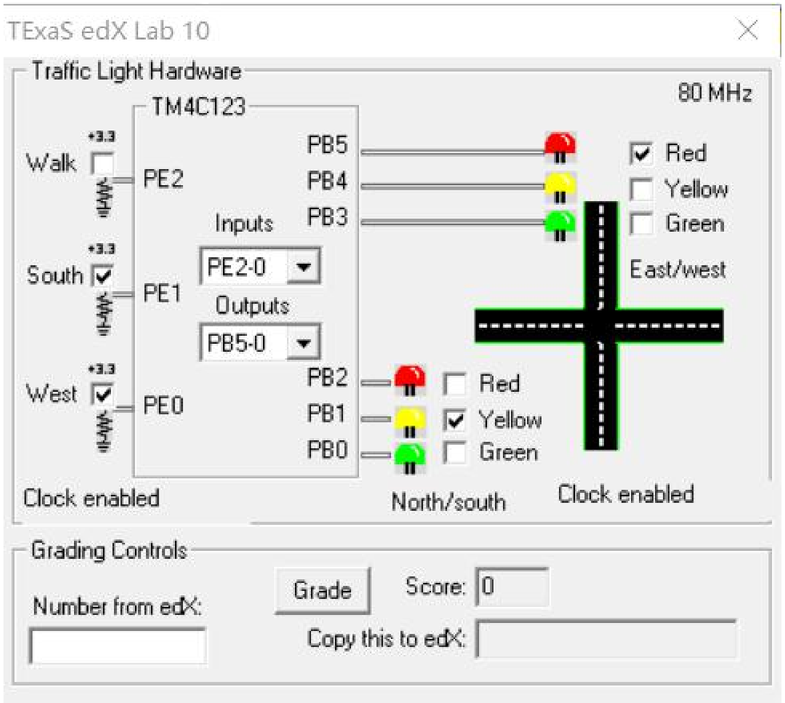
Number from edX:

Grade

Score: 0

Copy this to edX:

状态 d：南北方向黄灯 3s



TExaS edX Lab 10

Traffic Light Hardware

TM4C123

80 MHz

Walk □ PE2

South ☑ PE1

West ☑ PE0

Inputs: PE2-0

Outputs: PB5-0

East/west: ☑ Red □ Yellow □ Green

North/south: □ Red ☑ Yellow □ Green

Clock enabled

Clock enabled

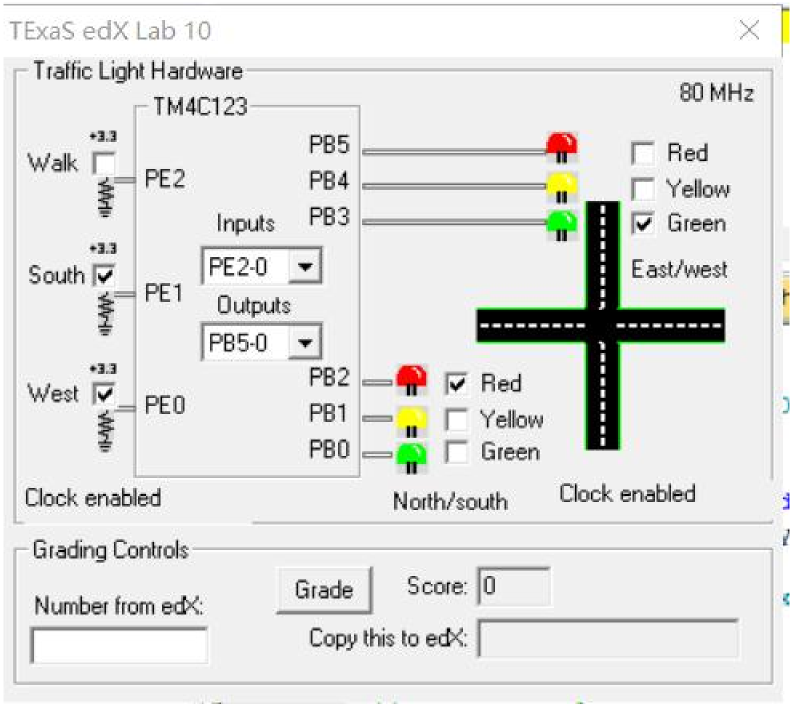
Grading Controls

Number from edX:

Score: 0

Copy this to edX:

状态 a:东西方向绿灯 30s



TExaS edX Lab 10

Traffic Light Hardware

TM4C123

80 MHz

Walk □ PE2

South ☑ PE1

West ☑ PE0

Inputs: PE2-0

Outputs: PB5-0

East/west: □ Red □ Yellow ☑ Green

North/south: ☑ Red □ Yellow □ Green

Clock enabled

Clock enabled

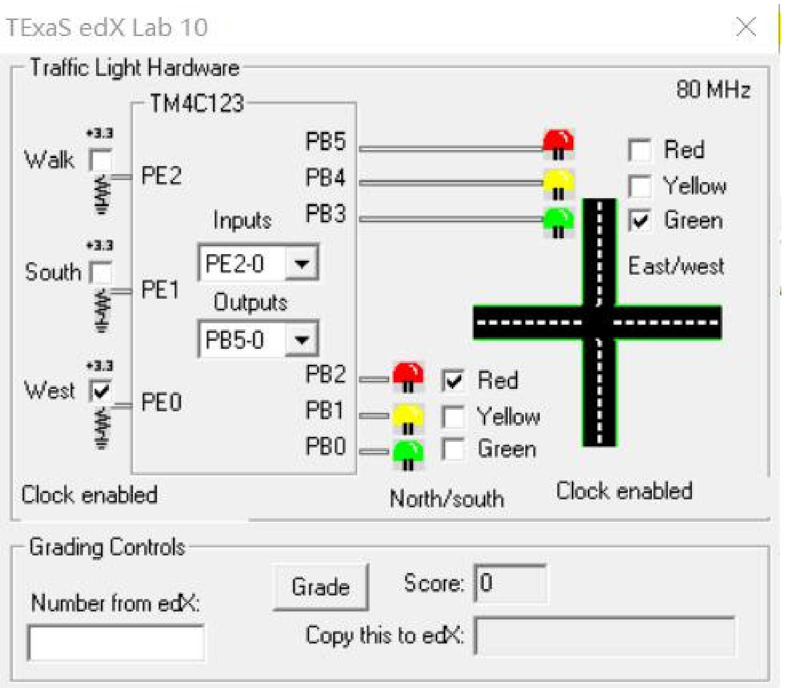
Grading Controls

Number from edX:

Score: 0

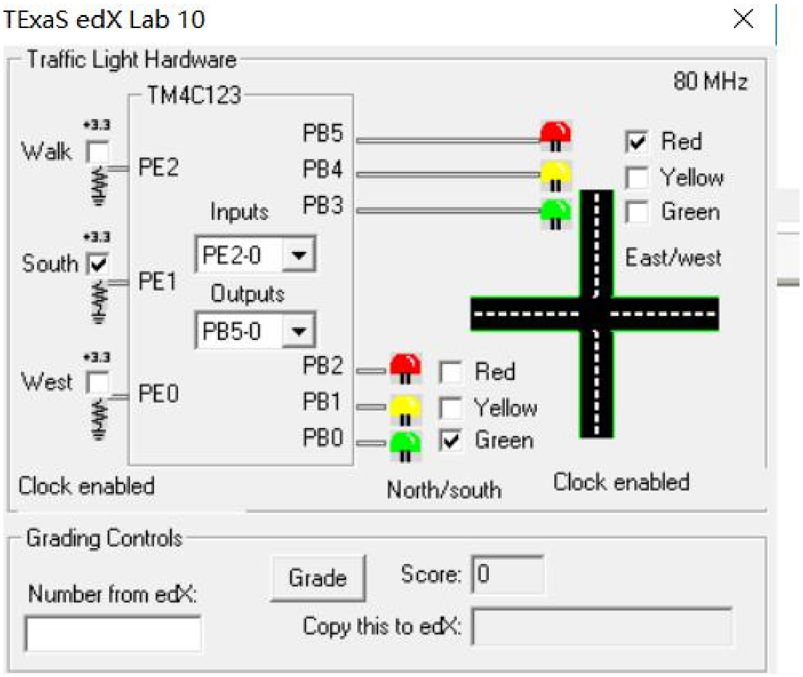
Copy this to edX:

当仅东西方向检测到有人/车时
保持东西方向绿灯

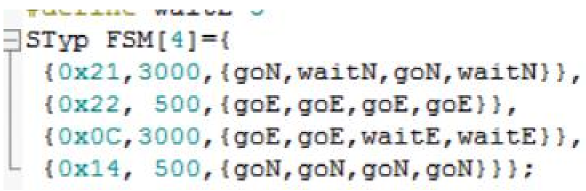


当仅南北方向检测到有人/车时：

保持南北方向绿灯

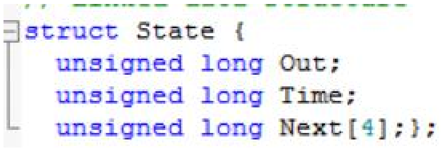


原理：



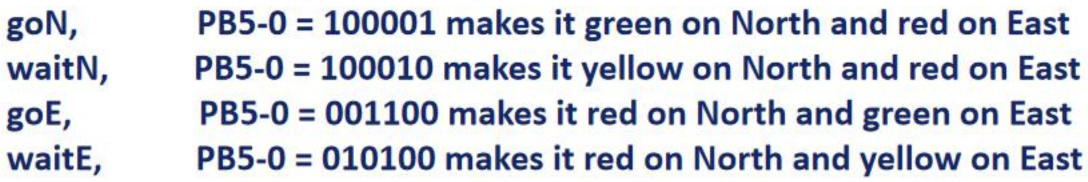
```
#define waitE 3
STyp FSM[4]={
 {0x21,3000,{goN,waitN,goN,waitN}},
 {0x22, 500,{goE,goE,goE,goE}},
 {0x0C,3000,{goE,goE,waitE,waitE}},
 {0x14, 500,{goN,goN,goN,goN}}};
```

程序通过有限状态机实现，有限状态机一共四个状态，每一个状态由一个结构体定义其状态，结构体字段如下：



```
struct State {
  unsigned long Out;
  unsigned long Time;
  unsigned long Next[4];};
```

其中 Out 控制交通灯的亮灭。规则如下：

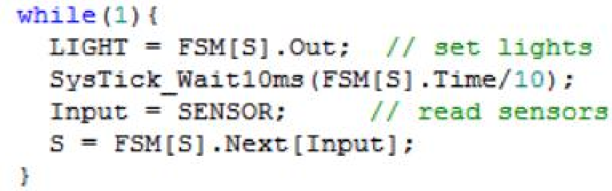


goN, **PB5-0 = 100001 makes it green on North and red on East**
waitN, **PB5-0 = 100010 makes it yellow on North and red on East**
goE, **PB5-0 = 001100 makes it red on North and green on East**
waitE, **PB5-0 = 010100 makes it red on North and yellow on East**

Time 表示状态持续时间

Next 表示下一个状态的索引。即在在四个状态中的索引。

程序主要逻辑如下：



```
while(1){
  LIGHT = FSM[S].Out;  // set lights
  SysTick_Wait10ms(FSM[S].Time/10);
  Input = SENSOR;     // read sensors
  S = FSM[S].Next[Input];
}
```