

基于车流量控制通行时间的十字路口信号灯系统设计

15331319 吴志慧

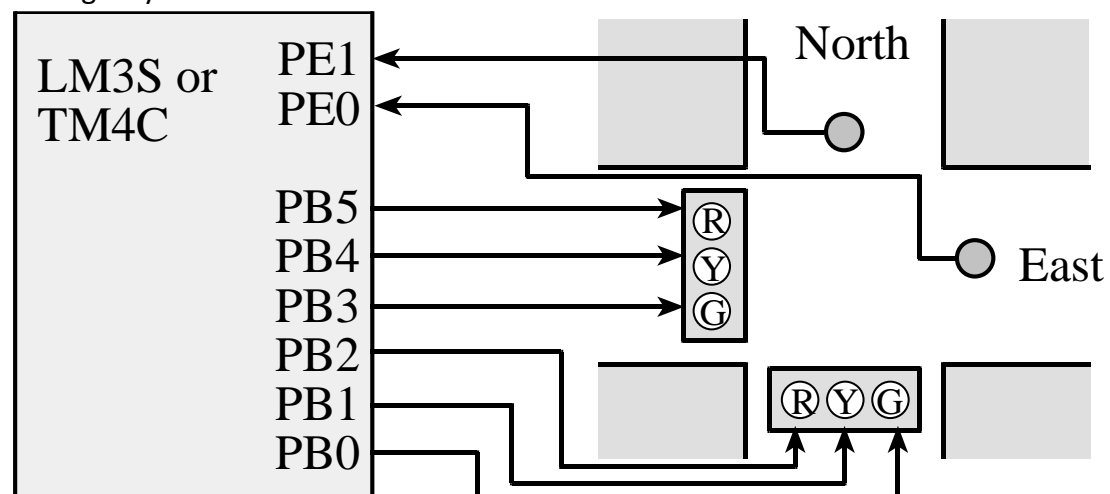
简介：本系统融合了 FSM 设计，A/D 转换，边缘触发中断。基于期中实验进行了两点改进：第一，将东与北两个方向的车流量进行量化，以此来给定某方向车辆的通行时间，即绿灯时间，车流量大的一方将有更多的通行时间，此处用到了数模转换。第二，加入关闭信号灯系统的功能，在任何情况下，按下关闭系统的开关，将导致所有灯熄灭，松开关闭系统的开关，系统继续正常运行，且并不需要从起始状态开始运行，即不同于断电式的关闭系统，此处用到了边缘触发的中断。

原来的情况：

Limitations

- four 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.

本实验：

Limitations

- four switches, PE2=1 emergency mode, PF4=1 stop system
- six LEDs , PB0-PB5, as figure, PB6 connected to red LED

PF4=0, PE2=0, PE1=0, PE0=0 means no cars exist on either road

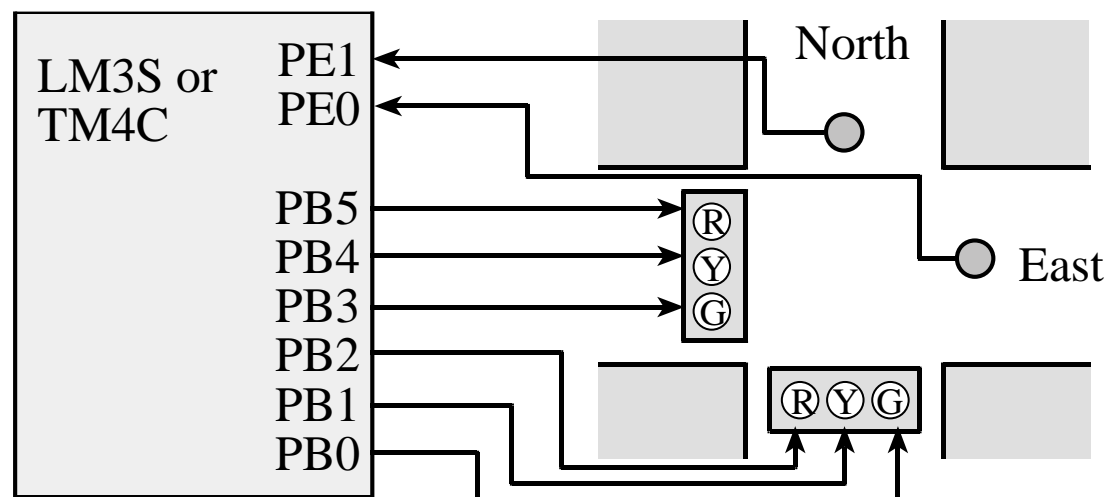
PF4=0, PE2=0, PE1=0, PE0=1 means there are cars on the East road

PF4=0, PE2=0, PE1=1, PE0=0 means there are cars on the North road

PF4=0, PE2=0, PE1=1, PE0=1 means there are cars on both roads

PF4=0, PE2=0=1XX, means emergency mode, cars do not enter in both directions. PB0-PB6 =0, emergency led PB6=1.

PF4=1, PE2=0=xxx, means stop system, turn off all led, PB6-0=0



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.

Stop, PB6-0=0000000 makes all led off

设计思路：

- 一．基于车流量控制通行时间。在 goN 和 goE 两个状态，状态停留的时间 time 将不同，时间 time 的设定规则如下。假设在 goN 状态下，N 方向的车流量与 E 方向车流量的比是 $x/1000$ ，其中 x 是模数转换的量化值（输入 PE4），范围是 0-4095，也即该系统支持某个方向的车流量是另一个方向车流量倍数为 0 到 4 倍，这在现实应用中是合理的。 $time = x/1000 * 30s$ 。比如 N 方向车流量是 E 方向车流量的两倍（ $x=2000$ ），则 N 方向通行 60 秒，E 红灯 60 秒；如果 N 和 E 方向车流量相同，则 N 通行 30，E 红灯 30 秒。程序中代码如下：

```
time = Pt==goN || Pt==goE? (int)(Pt->Time * (float)ADCvalue / 1000) : Pt->Time;  
//the state time with ADCvalue on goN or goE  
SysTick_Wait10ms(time); // wait 10 ms * current state's time
```

- 二．关闭系统，且开启系统后不必从初始状态开始运转，直接以当前情况开始运转。引入 stop 变量，用 PF4 输入边缘触发，上下边缘都触发，使 stop 交替变化。当中断产生，stop 从 0 变为 1，同时关闭所有灯，代表关闭系统，或者从 1 变为 0，代表开启系统。只要 stop=1，灯就一定全灭，但是输入传感器以及数模转换继续工作，当 PF4 下降沿触发 stop 变为 0 时，系统从当前传感器的输入的值以及数模转换的值的条件下正常工作。程序中代码如下：

```

//中断处理函数,开启与关闭系统切换
void GPIOPortF_Handler(void) {
    GPIO_PORTF_ICR_R = 0x10;        // acknowledge flag4
    stop ^= 1;    //stop or start the system

    if (stop)
        LIGHT = 0x00;
}

while(1){
    //stop时,灯一定全灭(中断处理程序也会熄灭灯),判断状态停留时间,ADC转换,输入传感器继续工作,一旦开启系统(松开PF4),直
    if (stop)
        LIGHT = 0x00;
    else
        LIGHT = Pt->Out;    // set lights to current state's Out value

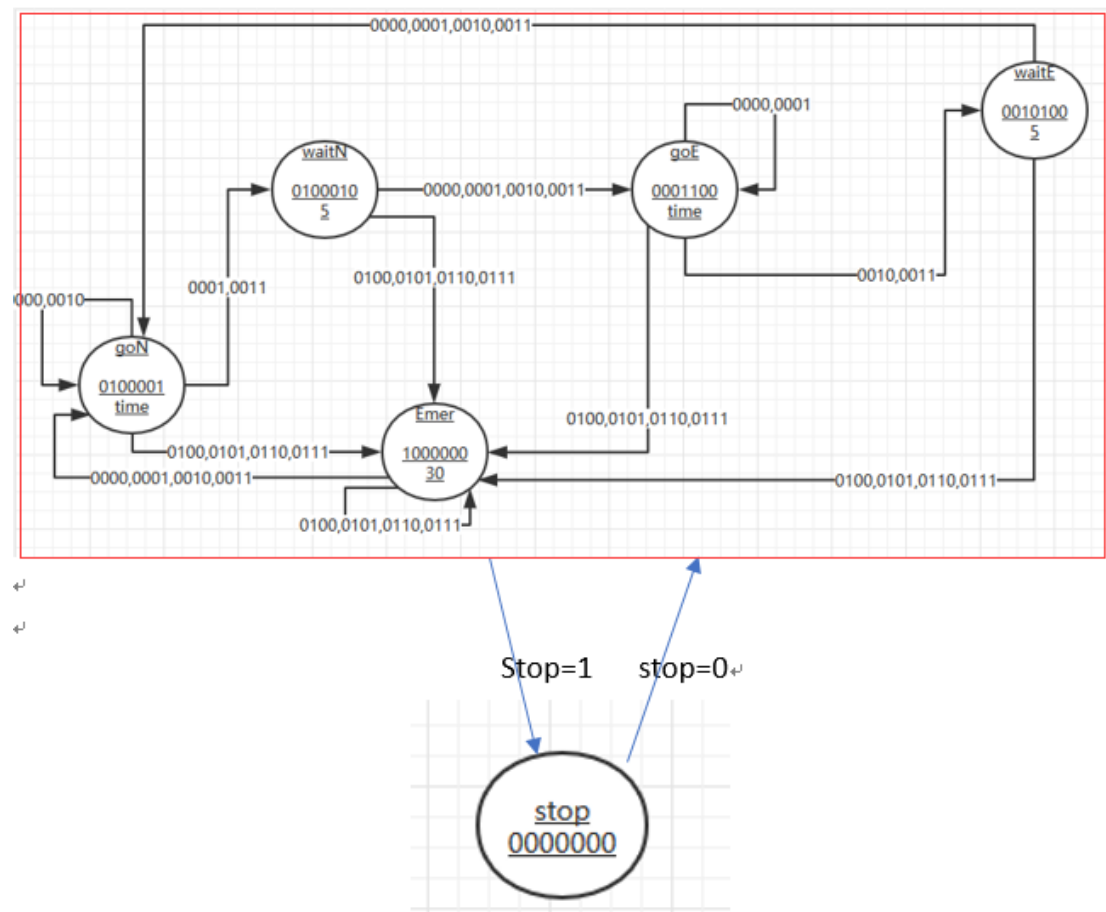
    ADCValue = ADC0_InSeq3();    //get ADCvalue from traffic
    time = Pt==goN || Pt==goE? (int)(Pt->Time * (float)ADCValue / 1000) : Pt->Time;    //the state time with ADCvalue on goN or
    SysTick_Wait10ms(time);    // wait 10 ms * current state's time
    Input = SENSOR;    // get new input from car detectors
    Pt = Pt->Next[Input];    // transition to next state
}

```

说明:

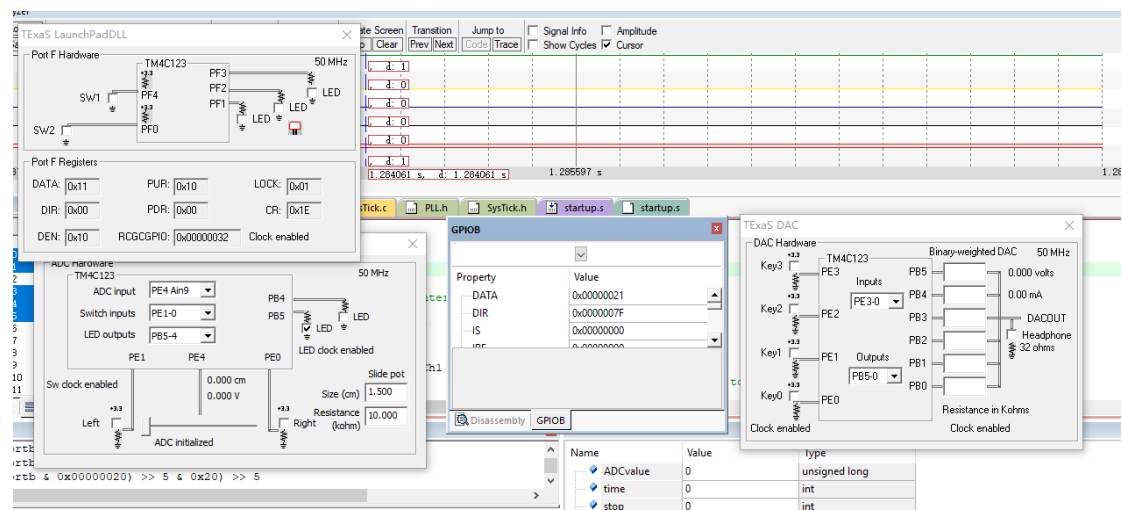
1. 任何状态下,只要接收到紧急信号(紧急状态不是关闭系统),则下一状态进入紧急状态
2. 紧急状态保持 30 秒(此处指应用在现实中 30 秒,程序中为了加快仿真时状态的转变,方便观察,只停留 3 秒,同理其他状态的时间都缩短十倍)
3. 在紧急状态下,如果没有接收到紧急信号,则下一状态进入初始状态 goN
4. 在任何情况下,只要 PF4 从 0 变为 1,产生中断,灯全灭,即关闭系统,直到 PF4 从 1 变为 0,产生中断,系统继续正常工作。

状态转换图：

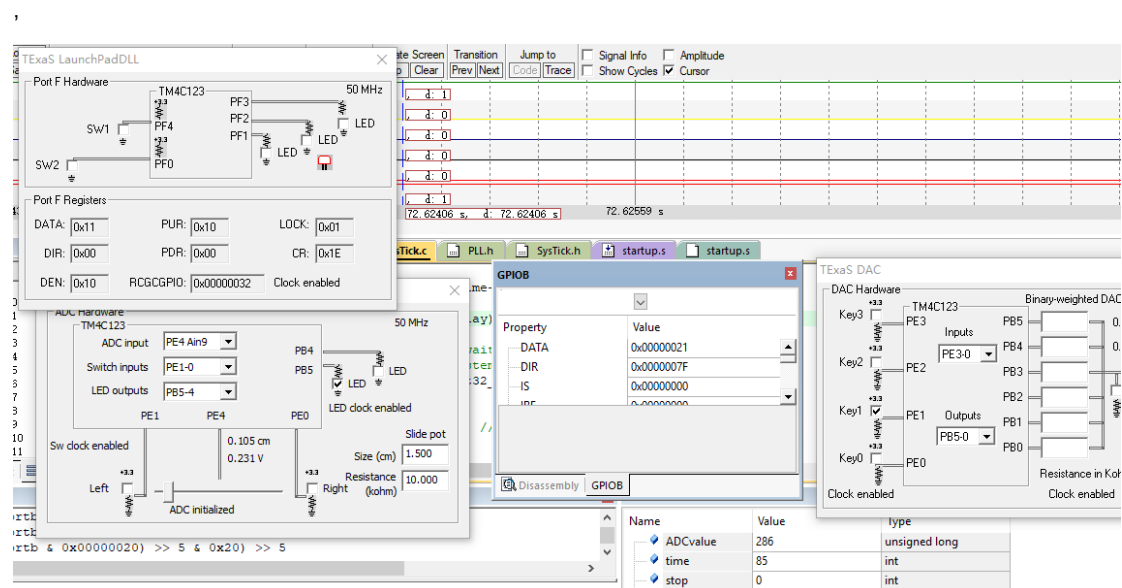


程序运行现象：

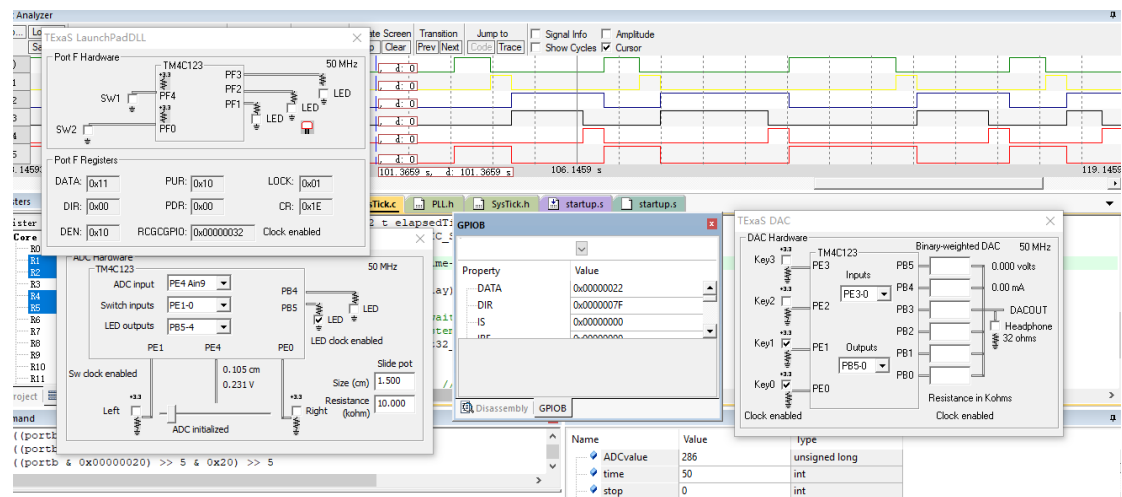
开始调试，进入初始状态，即 goN，此时 PF4=0，PE4=0，PE2=0，PE1=0，PE0=0 means no cars exist on either road，输出 0100001，makes it green on North and red on East。此时 ADCvalue=0，所以 time=0，由于 PE2=0，PE1=0，所以一直输出 0100001



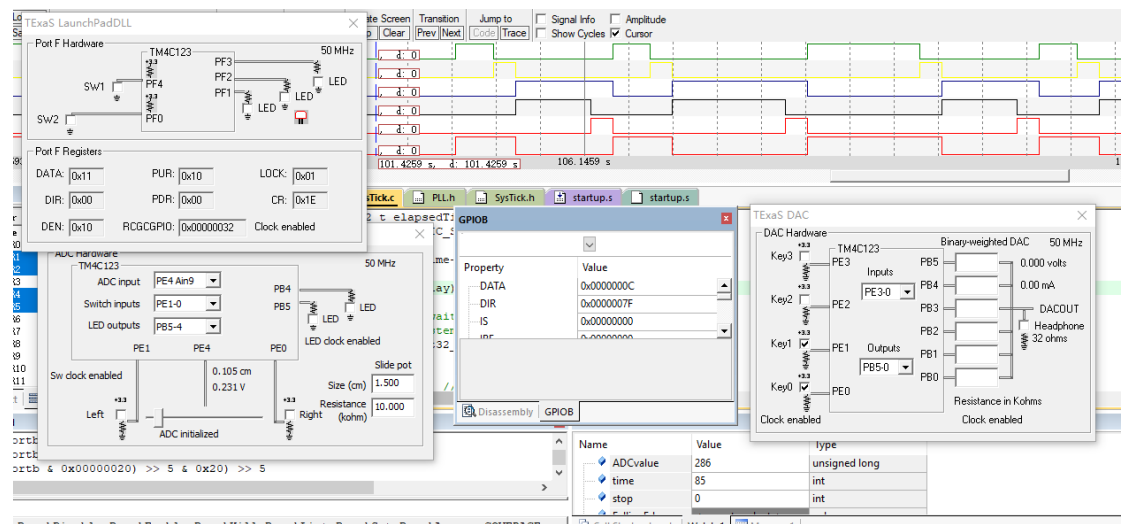
超过 3 秒后输入 PF4=0, PE4 = 0, PE2=0, PE1=1, PE0=0 means there are cars on the North road, 状态依旧为 goN



超过 3 秒后输入 PF4=0, PE2=0, PE1=0, PE0=1, 滑动 PE4 代表的模拟输入至 ACDvalue=286, 所以 $\text{time} = 300 \times 286 / 1000 = 85$, means there are cars on the East road, 此时状态从 goN 停留 0.85 秒后, 变为 waitN, 输出 0100010, makes it yellow on North and red on East



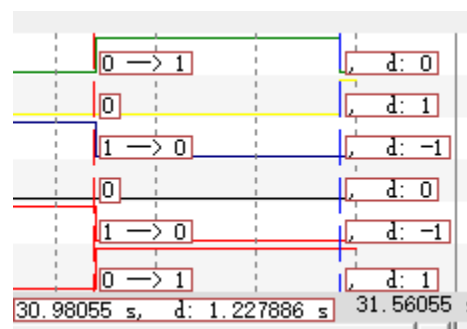
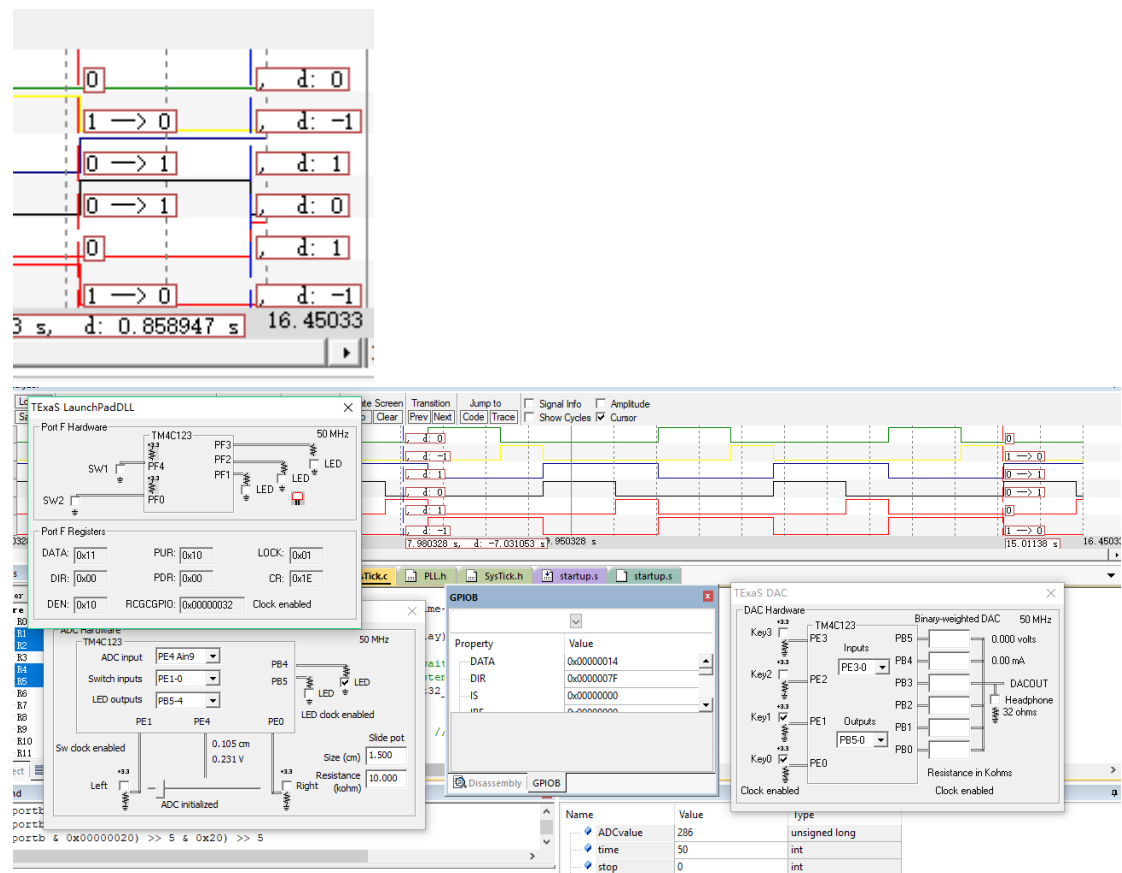
保持输入 PF4=0, PE2=0, PE1=0, PE0=1 means there are cars on the East road, 0.5 秒后, 状态变为 goE, 输出 00011000, makes it red on North and green on East

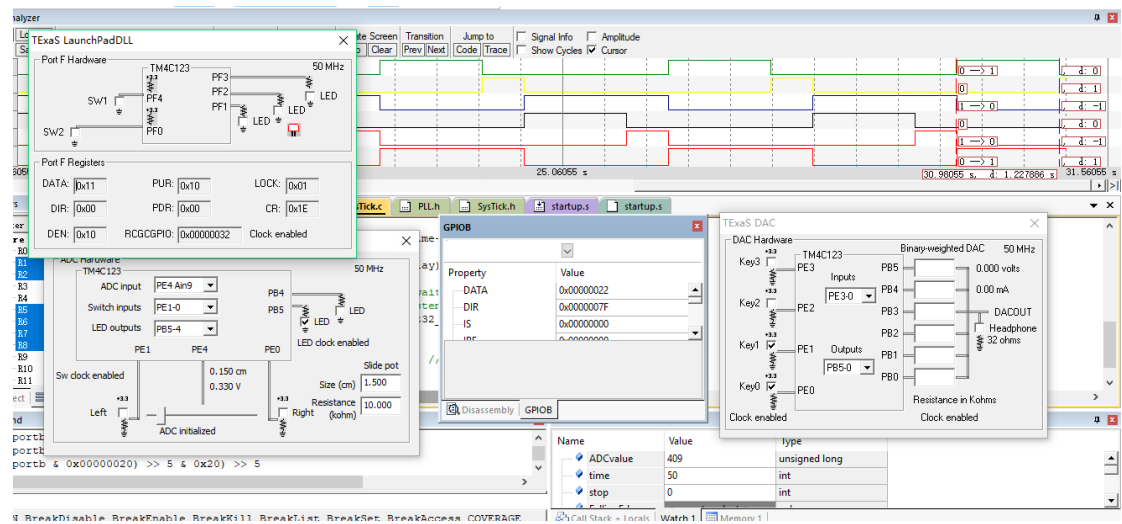


上面是简单测试, 以下来测试一个循环

输入 PF=0, PE2=0, PE1=1, PE0=1 means there are cars on both roads, 在 waitN 时

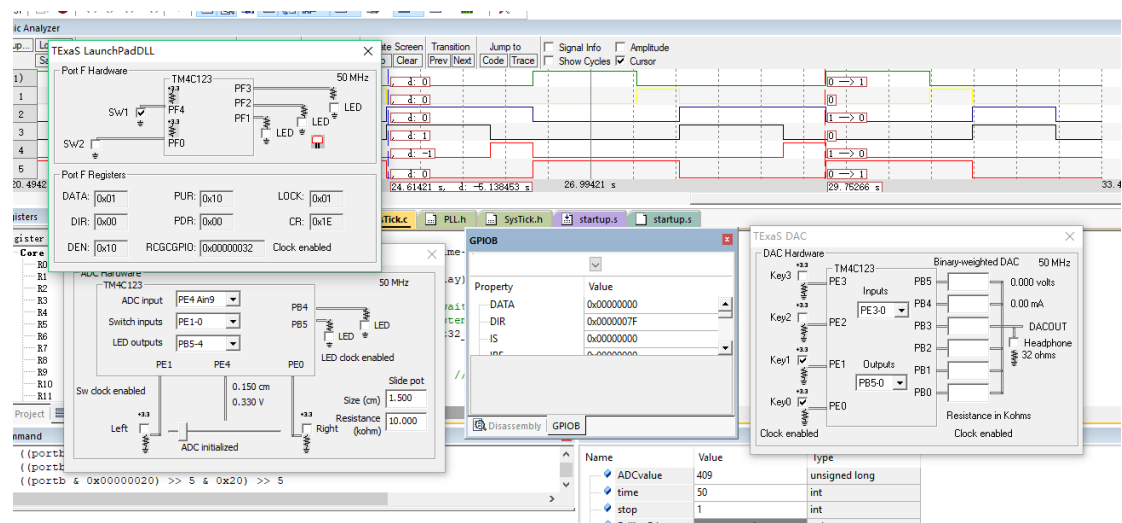
滑动 PE4 代表的模拟输入至 ACDvalue=286, 所以 $\text{time} = 300 \times 286 / 1000 = 0.85$ 此时状态在 goE 会停留 0.85 秒, 在 waitE 时, 滑动 PE4 代表的模拟输入至 ACDvalue=409, 所以 $\text{time} = 300 \times 368 / 1000 = 1.12$ 此时状态在 goN 会停留 1.12 秒,



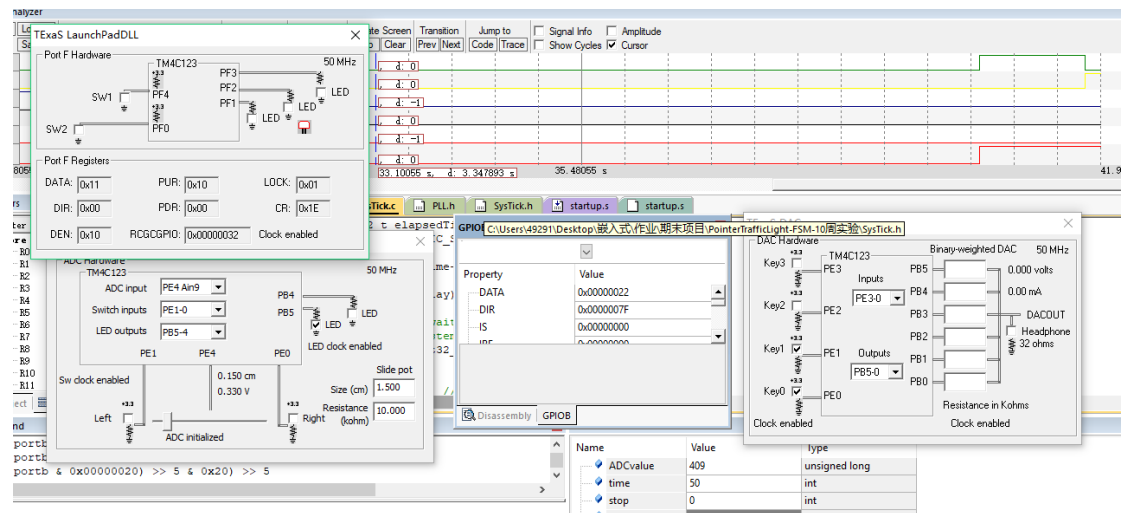


如上，证明实现了按流量控制通行时间的功能。

输入 PF4 从 0 到 1，触发中断，此时输出立即为 0000000



输入 PF4 从 1 到 0，触发中断，此时系统重新启动



如上证明了系统重新启程，直接开始工作，不必进入初始状态

至此，已测试实验的原有的功能以及增加的两个功能，全部符合要求。