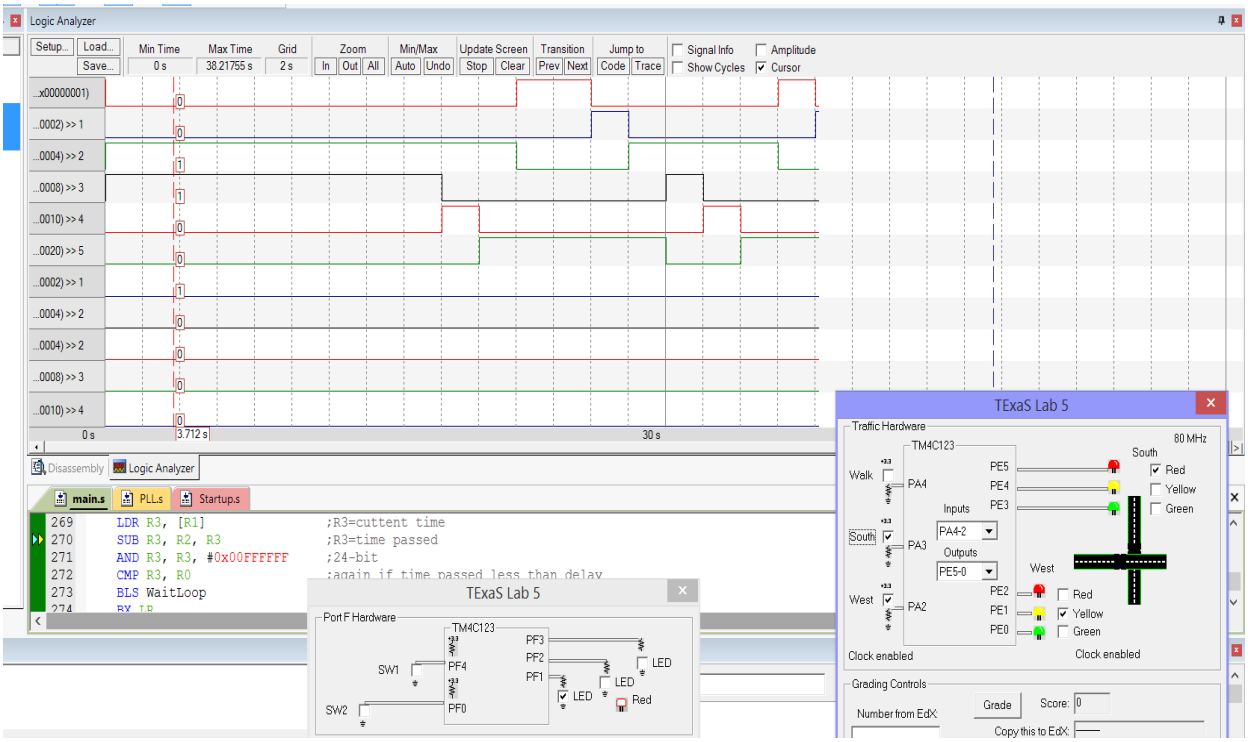
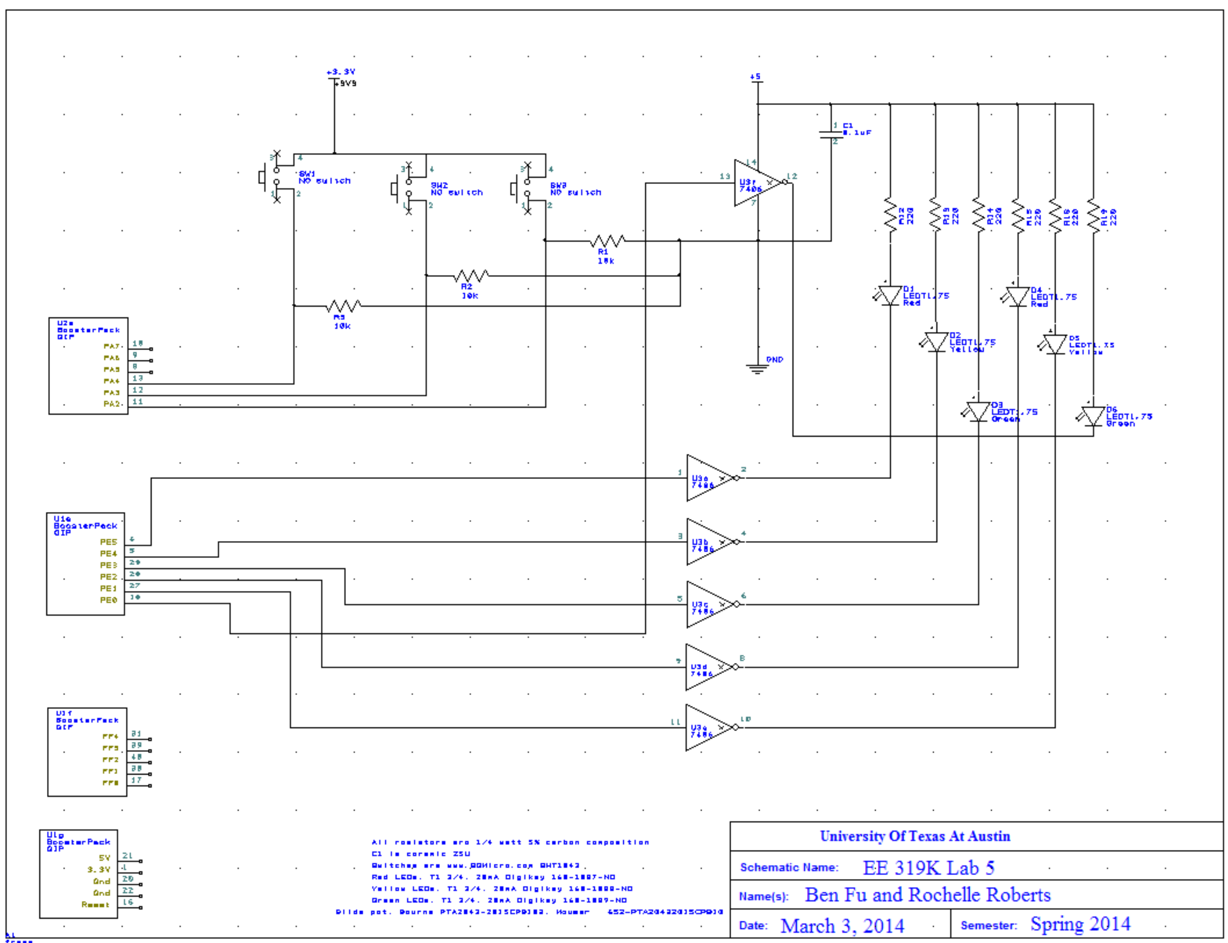
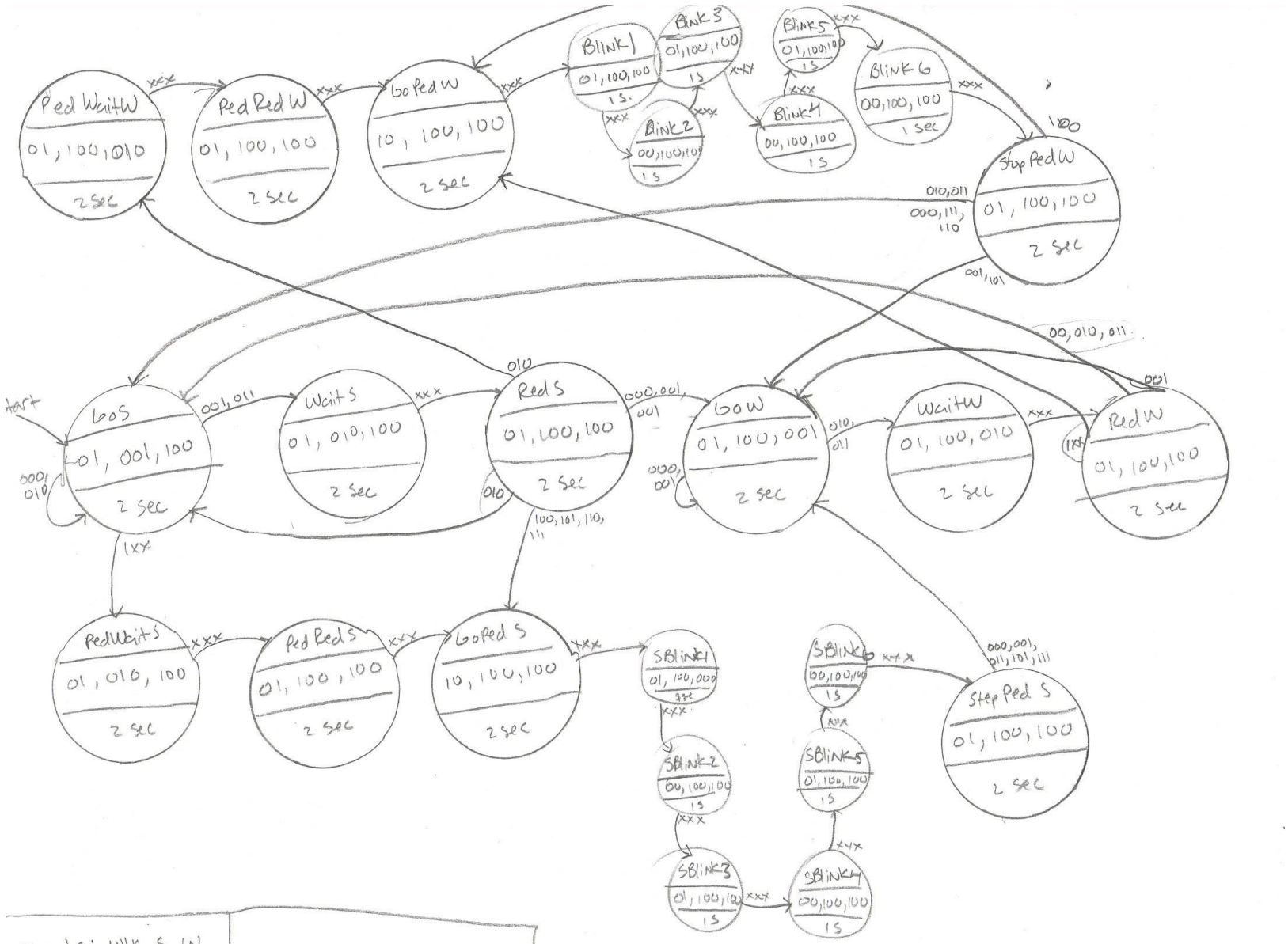
Ben Fu and Rochelle Roberts

Lab 5 Report




All resistors are 1/4 watt 5% carbon composition
C1 is ceramic Z5U
University Of Texas At Austin
Schematic Name: EE 319K Lab 5
Name(s): Ben Fu and Rochelle Roberts
Date: March 3, 2014
Semester: Spring 2014



Inputs: WlK, S, W

outputs: $\frac{G}{WlK}$, $\frac{RYG}{S}$, $\frac{RYG}{W}$

Rochelle Roberts and Ben Fu

```
;****************** main.s ***************
; Program written by: put your names here
; Date Created: 8/25/2013
; Last Modified: 10/9/2013
; Section 1-2pm     TA: Saugata Bhattacharyya
; Lab number: 5
; Brief description of the program
;   A traffic light controller with 3 inputs and 8 output
; Hardware connections
;The “don’t walk” and “walk” lights must be PF1 and PF3 respectively,
but where to attach the others have some flexibility.
;Obviously, you will not connect both inputs and outputs to the same
pin.

;Red south       PA7    PB5    PE5
;Yellow south    PA6    PB4    PE4
;Green south     PA5    PB3    PE3
;Red west        PA4    PB2    PE2
;Yellow west     PA3    PB1    PE1
;Green west      PA2    PB0    PE0
;Table 5.1. Possible ports to interface the traffic lights (PF1=red
don’t walk, PF3=green walk).

;Walk sensor     PA4    PD2    PE2
;South sensor    PA3    PD1    PE1
;West sensor     PA2    PD0    PE0
;Table 5.2. Possible ports to interface the sensors.

SYSCTL_RCGC2_R          EQU 0x400FE108
GPIO_PORTA_DATA_R       EQU 0x400043FC
GPIO_PORTA_DIR_R        EQU 0x40004400
GPIO_PORTA_AFSEL_R      EQU 0x40004420
GPIO_PORTA_DEN_R        EQU 0x4000451C
GPIO_PORTB_DATA_R       EQU 0x400053FC
GPIO_PORTB_DIR_R        EQU 0x40005400
GPIO_PORTB_AFSEL_R      EQU 0x40005420
GPIO_PORTB_DEN_R        EQU 0x4000551C
GPIO_PORTD_DATA_R       EQU 0x400073FC
GPIO_PORTD_DIR_R        EQU 0x40007400
GPIO_PORTD_AFSEL_R      EQU 0x40007420
GPIO_PORTD_DEN_R        EQU 0x4000751C
GPIO_PORTE_DATA_R       EQU 0x400243FC
GPIO_PORTE_DIR_R        EQU 0x40024400
GPIO_PORTE_AFSEL_R      EQU 0x40024420
GPIO_PORTE_DEN_R        EQU 0x4002451C
GPIO_PORTF_DATA_R       EQU 0x400253FC
GPIO_PORTF_DIR_R        EQU 0x40025400
GPIO_PORTF_AFSEL_R      EQU 0x40025420
GPIO_PORTF_DEN_R        EQU 0x4002551C
NVIC_ST_CURRENT_R       EQU 0xE000E018
NVIC_ST_CTRL_R          EQU 0xE000E010
NVIC_ST_RELOAD_R        EQU 0xE000E014
```

```
      AREA   DATA, ALIGN=2


      ALIGN
      AREA    |.text|, CODE, READONLY, ALIGN=2
      THUMB
      EXPORT  Start
      IMPORT  PLL_Init


Delay10ms  EQU 800000      ;for Systick_Wait10ms
Out             EQU 0                ;output
Wait       EQU 4                ;delay
Next       EQU 8                ;next state
Sensors         EQU 0x40004070
CarLight   EQU 0x400240FC
PedLight   EQU 0x40025018

GoS
     DCD 0x4C                   ;Walk=Red, South=Green, West=Red
     DCD 200                         ;2 second delay
     DCD GoS, WaitS, GoS, WaitS, PedWaitS, PedWaitS, PedWaitS,
PedWaitS   ;next state:0,1,2,3,4,5,6,7,8
WaitS
     DCD 0x54                   ;Walk=Red, South=Yellow, West=Red
     DCD 200
     DCD RedS, RedS, RedS, RedS, RedS, RedS, RedS, RedS
RedS
     DCD 0x64                   ;Walk=Red, South=Red, West=Red
     DCD 200
     DCD GoW, GoW, GoS, GoW, GoPedS, GoPedS, GoPedS, GoPedS
GoW
     DCD 0x61                   ;Walk=Red, South=Red, West=Green
     DCD 200
     DCD GoW, GoW, WaitW, WaitW, PedWaitW, PedWaitW, PedWaitW,
PedWaitW
WaitW
     DCD 0x62                   ;Walk=Red, South=Red, West=Yellow
     DCD 200
     DCD RedW, RedW, RedW, RedW, RedW, RedW, RedW, RedW
RedW
     DCD 0x64                   ;Walk=Red, South=Red, West=Red
     DCD 200
     DCD GoS, GoW, GoS, GoS, GoPedW, GoPedW, GoPedW, GoPedW
PedWaitS
     DCD 0x54                   ;Walk=Red, South=Yellow, West=Red
     DCD 200
     DCD PedRedS, PedRedS, PedRedS, PedRedS, PedRedS, PedRedS,
PedRedS, PedRedS
PedRedS
```

```
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 200
      DCD GoPedS, GoPedS, GoPedS, GoPedS, GoPedS, GoPedS, GoPedS,
GoPedS
GoPedS
      DCD 0xA4                    ;Walk=Green, South=Red, West=Red
      DCD 200
      DCD Blink1, Blink1, Blink1, Blink1, Blink1, Blink1, Blink1,
Blink1
Blink1
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 100
      DCD Blink2, Blink2, Blink2, Blink2, Blink2, Blink2, Blink2,
Blink2
Blink2
      DCD 0x24                    ;Walk=00, South=Red, West=Red
      DCD 100
      DCD Blink3, Blink3, Blink3, Blink3, Blink3, Blink3, Blink3,
Blink3
Blink3
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 100
      DCD Blink4, Blink4, Blink4, Blink4, Blink4, Blink4, Blink4,
Blink4
Blink4
      DCD 0x24                    ;Walk=00, South=Red, West=Red
      DCD 100
      DCD Blink5, Blink5, Blink5, Blink5, Blink5, Blink5, Blink5,
Blink5
Blink5
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 100
      DCD Blink6, Blink6, Blink6, Blink6, Blink6, Blink6, Blink6,
Blink6
Blink6
      DCD 0x24                    ;Walk=00, South=Red, West=Red
      DCD 100
      DCD StopPedS, StopPedS, StopPedS, StopPedS, StopPedS, StopPedS,
StopPedS, StopPedS
StopPedS
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 200
      DCD GoW, GoW, GoS, GoW, GoPedS, GoW, GoS, GoW
PedWaitW
      DCD 0x62                    ;Walk=Red, South=Red, West=Yellow
      DCD 200
      DCD PedRedW, PedRedW, PedRedW, PedRedW, PedRedW, PedRedW,
PedRedW, PedRedW
PedRedW                                ;Walk=Red, South=Red, West=Red
      DCD 0x64
      DCD 200
```

```
      DCD GoPedW, GoPedW, GoPedW, GoPedW, GoPedW, GoPedW, GoPedW,
GoPedW
GoPedW
      DCD 0xA4                    ;Walk=Green, South=Red, West=Red
      DCD 200
      DCD WBlink1, WBlink1, WBlink1, WBlink1, WBlink1, WBlink1,
WBlink1, WBlink1
WBlink1
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 100
      DCD WBlink2, WBlink2, WBlink2, WBlink2, WBlink2, WBlink2,
WBlink2, WBlink2
WBlink2
      DCD 0x24                    ;Walk=00, South=Red, West=Red
      DCD 100
      DCD WBlink3, WBlink3, WBlink3, WBlink3, WBlink3, WBlink3,
WBlink3, WBlink3
WBlink3
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 100
      DCD WBlink4, WBlink4, WBlink4, WBlink4, WBlink4, WBlink4,
WBlink4, WBlink4
WBlink4
      DCD 0x24                    ;Walk=00, South=Red, West=Red
      DCD 100
      DCD WBlink5, WBlink5, WBlink5, WBlink5, WBlink5, WBlink5,
WBlink5, WBlink5
WBlink5
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 100
      DCD WBlink6, WBlink6, WBlink6, WBlink6, WBlink6, WBlink6,
WBlink6, WBlink6
WBlink6
      DCD 0x24                    ;Walk=00, South=Red, West=Red
      DCD 100
      DCD StopPedW, StopPedW, StopPedW, StopPedW, StopPedW, StopPedW,
StopPedW, StopPedW
StopPedW
      DCD 0x64                    ;Walk=Red, South=Red, West=Red
      DCD 200
      DCD GoS, GoW, GoS, GoS, GoPedW, GoW, GoS, GoW

Start
    BL  PLL_Init            ;running at 80 MHz
      BL   SysTick_Init
;turn port clock on for port A, E and F
      LDR R0, =SYSCTL_RCGC2_R
      LDR R1, [R0]
      ORR R1, #0x31
      STR R1, [R0]
;wait 2 cycles
      NOP
```

```
      NOP
;set PortA DIR
      LDR R0, =GPIO_PORTA_DIR_R
      LDR R1, [R0]
      AND R1, #0xE3              ;set input, bit[4-2]=0
      STR R1, [R0]
;set PortE DIR
      LDR R0, =GPIO_PORTE_DIR_R
      LDR R1, [R0]
      ORR R1, #0x3F              ;set output, bit[5-0]=1
      STR R1, [R0]
;set PortF DIR
      LDR R0, =GPIO_PORTF_DIR_R
      LDR R1, [R0]
      ORR R1, #0x06              ;set PF output, bit[2-1]=1
      STR R1, [R0]
;turn off PortA AFSEL
      LDR R0, =GPIO_PORTA_AFSEL_R
      LDR R1, [R0]
      AND R1, #0xE3              ;dissable for bit[4-2]=0
      STR R1, [R0]
;turn off PortE AFSEL
      LDR R0, =GPIO_PORTE_AFSEL_R
      LDR R1, [R0]
      AND R1, #0xC0              ;dissable for bit[5-0]=0
      STR R1, [R0]
;turn off PortF AFSEL
      LDR R0, =GPIO_PORTF_AFSEL_R
      LDR R1, [R0]
      AND R1, #0xF9              ;dissable for bit[2-1]=0
      STR R1, [R0]
;enable PortA DEN
      LDR R0, =GPIO_PORTA_DEN_R
      LDR R1, [R0]
      ORR R1, #0x1C              ;set bit[4-2] to 1
      STR R1, [R0]
;enable PortE DEN
      LDR R0, =GPIO_PORTE_DEN_R
      LDR R1, [R0]
      ORR R1, #0x3F              ;set bit[5-0] to 1
      STR R1, [R0]
;enable PortF DEN
      LDR R0, =GPIO_PORTF_DEN_R
      LDR R1, [R0]
      ORR R1, #0x06              ;set bit[2-1] to 1
      STR R1, [R0]

;FSM engine begins
      LDR R4, =GoS               ;R4 holds Current State
      LDR R5, =Sensors      ;PortA
      LDR R6, =CarLight          ;PortE
      LDR R7, =PedLight          ;PortF
```

```
FSM
      LDR R0, [R4,#Out]           ;write output
      STR R0, [R6]
      LSR  R0, #5
      STR R0, [R7]
      LDR R0, [R4, #Wait]         ;delay based on CS
      BL   SysTick_Wait10ms
      LDR R0, [R5]
      ADD R0, R0, #Next
      LDR R4, [R4, R0]      ;update CS
      B FSM

SysTick_Init
      LDR R1, =NVIC_ST_CTRL_R          ; 1. disable timer, clear ctrl
      MOV R0, #0
      STR R0, [R1]
      LDR R1, =NVIC_ST_RELOAD_R  ; 2. load reload value
      LDR R0, =0x00FFFFFF
      STR R0, [R1]
      LDR R1, =NVIC_ST_CURRENT_R ; 3. clear current
      MOV R0, #0
      STR R0, [R1]
      LDR R1, =NVIC_ST_CTRL_R          ; 4. enable systick with core
source
      MOV R0, #0x05
      STR R0, [R1]
      BX LR

Systick_Wait
      LDR R1, =NVIC_ST_CURRENT_R
      LDR R2, [R1]                     ;R2= start time
WaitLoop
      LDR R3, [R1]                     ;R3=cuttent time
      SUB R3, R2, R3                   ;R3=time passed
      AND R3, R3, #0x00FFFFFF          ;24-bit
      CMP R3, R0                       ;again if time passed less than
delay
      BLS WaitLoop
      BX LR

SysTick_Wait10ms
      PUSH {R4,LR}
      MOVS R4, R0
      BEQ Wait10ms_Done
Wait10msLoop
      LDR R0, =Delay10ms
      BL Systick_Wait
      SUBS R4, R4, #1
      BHI Wait10msLoop
Wait10ms_Done
      POP {R4, PC}
```

```
ALIGN           ; make sure the end of this section is aligned
END             ; end of file
```