

stop_go.c.c

```
1 // Stop Go C/ASM Mix Example
2 // Jason Losh
3
4 //-----
5 // Hardware Target
6 //-----
7
8 // Target Platform: EK-TM4C123GXL Evaluation Board
9 // Target uC:      TM4C123GH6PM
10 // System Clock:   40 MHz
11
12 // Hardware configuration:
13 // Red LED:
14 //   PF1 drives an NPN transistor that powers the red LED
15 // Green LED:
16 //   PF3 drives an NPN transistor that powers the green LED
17 // Pushbutton:
18 //   SW1 pulls pin PF4 low (internal pull-up is used)
19
20 //-----
21 // Device includes, defines, and assembler directives
22 //-----
23
24 #include <stdint.h>
25 #include <stdbool.h>
26 #include "tm4c123gh6pm.h"
27
28 #define RED_LED      (*((volatile uint32_t *) (0x42000000 + (0x400253FC-0x40000000)*32 + 1*4)))
29 #define GREEN_LED    (*((volatile uint32_t *) (0x42000000 + (0x400253FC-0x40000000)*32 + 3*4)))
30 #define PUSH_BUTTON  (*((volatile uint32_t *) (0x42000000 + (0x400253FC-0x40000000)*32 + 4*4)))
31
32 //-----
33 // Subroutines
34 //-----
35
36 // Blocking function that returns only when SW1 is pressed
37 extern void waitPbPress();
38
39 // Initialize Hardware
40 void init_hw()
41 {
42     // Configure HW to work with 16 MHz XTAL, PLL enabled, system clock of 40 MHz
43     SYSCTL_RCC_R = SYSCTL_RCC_XTAL_16MHZ | SYSCTL_RCC_OSCSRC_MAIN | SYSCTL_RCC_USESYSDIV | (4
44     << SYSCTL_RCC_SYSDIV_S);
45
46     // Set GPIO ports to use APB (not needed since default configuration -- for clarity)
47     SYSCTL_GPIOHBCCTL_R = 0;
48
49     // Enable GPIO port F peripherals
50     SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF;
51
52     // Configure LED and pushbutton pins
53     GPIO_PORTF_DIR_R = 0x0A; // bits 1 and 3 are outputs, other pins are inputs
54     GPIO_PORTF_DR2R_R = 0x0A; // set drive strength to 2mA (not needed since default
55     configuration -- for clarity)
56     GPIO_PORTF_DEN_R = 0x1A; // enable LEDs and pushbuttons
```

stop_go.c.c

```
55  GPIO_PORTF_PUR_R = 0x10;  // enable internal pull-up for push button
56 }
57
58 //-----
59 // Main
60 //-----
61
62 int main(void)
63 {
64     // Initialize hardware
65     init_hw();
66
67     // Turn on red LED, turn off green LED
68     RED_LED = 1;
69     GREEN_LED = 0;
70
71     // Wait for PB press
72     waitPbPress();
73
74     // Turn off red LED
75     RED_LED = 0;
76
77     // Turn on green LED
78     __asm("        PUSH {R0, R1}");
79     __asm("        B    next");
80     __asm("GREEN_LED: .fi el d 0x424A7F8C");
81     __asm("next:    LDR R0, GREEN_LED");
82     __asm("        MOV R1, #1");
83     __asm("        STR R1, [R0]");
84     __asm("        POP {R0, R1}");
85
86     // Endless loop
87     while(1);
88 }
89
```