

```

1  /*****
2  * Name:      Sensors.c
3  * Description: STM32 SR04 Ultrasonic Sensor interface
4  * Version: V1.00
5  * Author: Ammar Alvi & Shannon D'Souza
6  *
7  * This software is supplied "AS IS" without warranties of any kind.
8  *****/
9
10 #include "stm32f10x.h"
11 #include "GPIO.h"
12 #include "UTIL.h"
13
14 /*
15 * Name:      uint32_t FrontRightUSLen(void)
16              uint32_t RightUSLen(void)
17              uint32_t FrontUSLen(void)
18
19 * Paramaters: none
20 * Description: Write a logic 1 to trigger pin of ultrasonic sensor
21                for 50 us, then the read the width of pulse returned
22                from the echo pin which is proportional to the distance
23                measured.
24                All the pulse width on the trig and echo pins were measured
25                using oscilloscopes to provide accurate readings
26                Returns 32 bit value which represents the mm of distance
27                measured. The result measured to be within +/- 5mm accuracy
28                which is good enough for our application.
29 */
30
31 //A timeout variabe declared in case an interrupt causes the funciton
32 // to get stuck in an infinite loop
33 uint32_t timeout = 0;
34
35 uint32_t FrontRightUSLen(void)    //Front Right Ultrasonic Sensor
36 {
37     uint32_t mm = 0;
38     timeout = 0;
39     GPIOC->BSRR |= GPIO_BSRR_BR8;    //Set trigger pin to 0
40     delay(20);
41     GPIOC->BSRR |= GPIO_BSRR_BS8;    //Write logic 1 to trigger pin
42     delay(54);                      //Aproximately 10us
43     GPIOC->BSRR |= GPIO_BSRR_BR8;    //Wirte logic 0 to trigger pin
44     //Wait for the echo pin to go high with a timeout incase its unable to read
45     //a logic 1
46     while((GPIOC->IDR & GPIO_IDR_IDR9) != GPIO_IDR_IDR9)
47     {
48         timeout++;
49         if (timeout > 60000)
50         {
51             return 100;
52         }
53     }
54     //Measure the width of the pulse received on the echo pin
55     while((GPIOC->IDR & GPIO_IDR_IDR9) == GPIO_IDR_IDR9)
56     {
57         delay(12);
58         mm = mm+3;
59     }
60     mm = (mm*10)/58;    //Convert pulsewidth to milimeters
61     return mm;
62 }
63
64 uint32_t RightUSLen(void)         //Middle Right Ultrasonic Sensor
65 {
66     uint32_t mm2 = 0;
67     timeout = 0;
68     GPIOB->BSRR |= GPIO_BSRR_BR13;
69     delay(20);
70     GPIOB->BSRR |= GPIO_BSRR_BS13;
71     delay(54);
72     GPIOB->BSRR |= GPIO_BSRR_BR13;

```

```
73     while((GPIOB->IDR & GPIO_IDR_IDR14) != GPIO_IDR_IDR14)
74     {
75         timeout++;
76         if (timeout > 60000)
77         {
78             return 100;
79         }
80     }
81     while((GPIOB->IDR & GPIO_IDR_IDR14) == GPIO_IDR_IDR14)
82     {
83         delay(12);
84         mm2 = mm2+3;
85     }
86     mm2 = (mm2*10)/58;
87     return mm2;
88 }
89
90 uint32_t FrontUSLen(void)    //Front Ultrasonic Sensor
91 {
92     uint32_t mm3 = 0;
93     timeout = 0;
94     GPIOB->BSRR |= GPIO_BSRR_BR4;
95     delay(20);
96     GPIOB->BSRR |= GPIO_BSRR_BS4;
97     delay(54);
98     GPIOB->BSRR |= GPIO_BSRR_BR4;
99     while((GPIOA->IDR & GPIO_IDR_IDR15) != GPIO_IDR_IDR15)
100    {
101        timeout++;
102        if (timeout > 60000)
103        {
104            return 100;
105        }
106    }
107    while((GPIOA->IDR & GPIO_IDR_IDR15) == GPIO_IDR_IDR15)
108    {
109        delay(12);
110        mm3 = mm3+3;
111    }
112    mm3 = (mm3*10)/58;
113    return mm3;
114 }
115
116
```