

```
1  /*ENEL 387*/
2  /*Authors: Daniel Takyi & Dwijen Kapadia*/
3  /*Filename: sensors.c*/
4
5  #include "stm32f10x.h"
6  #include "sensors.h"
7
8
9
10 void RIGHTsense_init(void)
11 {
12     RCC->APB2ENR |= RCC_APB2ENR_IOPBEN; //ENABLE clock for Port B I/O
13 }
14
15
16 uint32_t RIGHTsense_read(void)
17 {
18     uint32_t temp;
19     temp = GPIOB->IDR;
20     temp &= 0x40;
21     temp = (temp >> 0x06); //bit manipulation to read PB6
22
23     return temp;
24 }
25
26
27 void FRONTSense_init(void)
28 {
29     RCC->APB2ENR |= RCC_APB2ENR_ADC1EN | RCC_APB2ENR_AFIOEN | RCC_APB2ENR_IOPAEN | RCC_APB2ENR_IOPBEN;
30     //Enable clocks for ADC1, Port A, Port B, and alternate function I/O
31
32     GPIOA->CRL &= 0xFFFF0FFF; //Analog input mode on PA3
33
34     ADC1->CR2 = 0x1;
35 }
36
37
38
39 uint32_t FRONTSense_read(void)
40 {
41     ADC1->SQR3 = 0x3; //use ADC CH3
42     ADC1->CR2 = 0x1;
43     uint32_t test = ADC1->SR;
44     test &= 0x2;
45     while (test != 0x2) // wait for eoc bit to stablilize
46     {
47         test = ADC1->SR;
48         test &= 0x2;
49     }
50     return ADC1->DR; // return unformatted data read
51 }
52
53
54 uint32_t FRONTDIGsense_read(void)
55 {
56     uint32_t temp;
57     temp = GPIOB->IDR;
58     temp &= 0x80;
59     temp = (temp >> 0x07); //bit manipulation to read PB7
60
61     return temp;
62 }
63
64 void LINEsense_init(void)
65 {
66     RCC->APB2ENR |= RCC_APB2ENR_IOPCEN; //ENABLE clock for Port C I/O
67 }
68
69
70 uint32_t LINEsense_read(void)
71 {
72     uint32_t temp;
73     temp = GPIOC->IDR;
```

```
74     temp &= 0x2000;
75     temp = (temp >> 0x0D); //bit manipulation to read PC13
76
77     return temp;
78 }
79
80
```