

```
1  /*Authors: Daniel Takyi & Dwijen Kapadia*/
2  /*Filename: main_PathFollow.c*/
3
4
5  #include <stdint.h>
6  #include "stm32f10x.h"
7  #include "clocks.h"
8  #include "pwm.h"
9  #include "sensors.h"
10
11
12
13
14  uint32_t convert(uint32_t reg_data)//converts data to a percentage value --- Range: 0-100
15  {
16      reg_data *= 0x64;
17      reg_data /= 0xffff;
18
19
20      return reg_data;
21  }
22
23
24  void delay(uint32_t count)//value of 6000 gives approximately 1ms of delay
25  {
26      int i = 0;
27      for(i = 0; i < count; i++)
28      {
29      }
30  }
31
32
33  int main()
34  {
35      clockInit();
36
37      pwmInit();
38
39      RIGHTsense_init();
40
41      LINEsense_init();
42
43      FRONTSense_init();
44
45
46      while((GPIOA->IDR & 0x1) != 0x1)
47      {}
48
49      int i;
50      int count = 0;
51
52
53      while(1)
54      {
55          while(LINEsense_read() == 0x1 ) // when the line is sensed the rover moves forward
56          {
57              PWMForward();
58
59              if(FRONTDIGsense_read() == 0x0 && count == 0) // if the front digital sensor sees an obstacle,
take the avoidance path
60              {
61                  PWMStop();
62                  delay(2000000);
63                  PWMLeft();
64                  delay(5800000);
65                  PWMStop();
66                  delay(2000000);
67                  PWMForward();
68                  delay(2000000);
69                  PWMStop();
70                  delay(2000000);
71                  PWMRight();
72                  delay(7800000);
```

```
73     PWMStop();
74     delay(2000000);
75     PWMForward();
76     delay(2000000);
77     PWMStop();
78     delay(2000000);
79     PWMRight();
80     delay(7800000);
81     PWMStop();
82     delay(2000000);
83     while(LINEsense_read() == 0x0)
84     {
85         PWMForward();
86     }
87     delay(2000000);
88     PWMStop();
89     delay(2000000);
90     count++;
91 }
92 while(convert(FRONTsense_read()) > 80 && (count == 1)) //if front analog sensor senses home,
then stop
93 {
94     PWMStop();
95 }
96 }
97 }
98
99 delay(3000);
100 if(LINEsense_read() == 0x0) // if rover goes off the line
101 {
102     PWMStop(); // stop
103     delay(6000);
104
105     for(i = 0; i < 400000 && LINEsense_read() == 0x0; i++) // check for line to the left
106     {
107         PWMSharpLeft();
108     }
109     PWMStop();
110     delay(6000);
111
112
113     for(i = 0; i < 800000 && LINEsense_read() == 0x0; i++) // check for line to the right
114     {
115         PWMSharpRight();
116     }
117
118
119     while(LINEsense_read() == 0x0) //if still not sensed, then stop
120     {
121         PWMStop();
122     }
123 }
124
125 }
126
127 }
128
129
130
```