

```

1 #include <Adafruit_MotorShield.h>
2
3 #define cleft_pin A2 // c stands for center
4 #define crigt_pin A1
5 #define fleft_pin A0 // f stands for far
6 #define frigt_pin A3 // we spelt it rigt b/c we wanted it to match the 4 letters of left
7
8 #define cleft_led_pin 4 // the pin to turn on the emitter
9 #define fleft_led_pin 5
10 #define crigt_led_pin 7
11 #define frigt_led_pin 6
12
13 #define left_motor_pin 2 // the connector the motor is hooked up to
14 #define right_motor_pin 1
15 #define thresh_on 690 // for previous bang bang code
16 #define thresh_off 400
17 #define isave .9 // for the i factor
18
19
20 float p =0.2; // for pid controller
21 float d =0.0;
22 float i =0.00;
23 int cweight =2; // the weight of the center sensors
24 int fweight =4; // the weight of the far sensors
25 int fspeed =40; // forward speed
26 int tspeed = 60; // turning speed
27
28
29 #include <Adafruit_MotorShield.h>
30
31 // Create the motor shield object with the default I2C address
32 Adafruit_MotorShield AFMS = Adafruit_MotorShield();
33 // Or, create it with a different I2C address (say for stacking)
34 // Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x61);
35
36 // Select which 'port' M1, M2, M3 or M4. In this case, M1
37 Adafruit_DCMotor *left = AFMS.getMotor(left_motor_pin);
38 Adafruit_DCMotor *right = AFMS.getMotor(right_motor_pin);
39
40 // You can also make another motor on port M2
41 //Adafruit_DCMotor *myOtherMotor = AFMS.getMotor(2);
42
43 void setup() {
44   // put your setup code here, to run once:
45   Serial.begin(115200);
46   pinMode(cleft_pin,INPUT); // init all sensor pins
47   pinMode(cleft_led_pin,OUTPUT);
48   pinMode(fleft_pin,INPUT);
49   pinMode(fleft_led_pin,OUTPUT);
50   pinMode(crigt_pin,INPUT);
51   pinMode(crigt_led_pin,OUTPUT);
52   pinMode(frigt_pin,INPUT);
53   pinMode(frigt_led_pin,OUTPUT);
54   if (!AFMS.begin()) { // create with the default frequency 1.6KHz
55     // if (!AFMS.begin(1000)) { // OR with a different frequency, say 1KHz
56       Serial.println("Could not find Motor Shield. Check wiring.");
57       while (1);
58     }
59   digitalWrite(fleft_led_pin,HIGH); // set the pins high

```

```

60 digitalWrite(cleft_led_pin,HIGH); // thought we might need to multiplex them but we didn't need
61 digitalWrite(cright_led_pin,HIGH);
62 digitalWrite(fright_led_pin,HIGH);
63 left->run(BACKWARD);
64 right->run(BACKWARD);
65 left->setSpeed(fspeed);
66 right->setSpeed(fspeed); //lurch it so we know it's on
67 delay(100);
68 left->run(RELEASE);
69 right->run(RELEASE);
70 delay(5000); // wait to start
71 }
72
73 void motors(int l,int r){ // get around the issue that you can't set a neg speed
74     if(l<0){
75         left->run(BACKWARD);
76     }else{
77         left->run(FORWARD);
78     }
79     if(r<0){
80         right->run(BACKWARD);
81     }else{
82         right->run(FORWARD);
83     }
84     left->setSpeed(abs(l)); // and print the speeds
85     right->setSpeed(abs(r));
86     Serial.print(",|, ");
87     Serial.print(l);
88     Serial.print(", ");
89     Serial.print(r);
90 }
91
92
93 int last_t=0; //for d
94 float d_val;
95 int i_holder; //keeps track of the i without the coef
96 void loop() {
97     if(Serial.available()>0){ //handle serial data
98         char c = Serial.read();
99         switch(c){
100             case 'p'://set p coef
101                 p = Serial.parseFloat();
102                 break;
103             case 'd'://set p coef
104                 d = Serial.parseFloat();
105                 break;
106             case 'i'://set p coef
107                 i = Serial.parseFloat();
108                 break;
109             case 'c'://set center weight coef
110                 cweight = Serial.parseInt();
111                 break;
112             case 'f'://set far weight coef
113                 fweight = Serial.parseInt();
114                 break;
115             case 's': //set forward speed
116                 fspeed = Serial.parseInt();
117                 break;
118             case 't'://set turning speed
119                 tspeed = Serial.parseInt();

```

```

120     break;
121     case 'v': // print code that we can copy and paste into Arduino
122
123         Serial.print("float p =");
124         Serial.print(p);
125         Serial.println(";");
126         Serial.print("float d =");
127         Serial.print(d);
128         Serial.println(";");
129         Serial.print("float i =");
130         Serial.print(i);
131         Serial.println(";");
132         Serial.print("int cweight =");
133         Serial.print(cweight);
134         Serial.println(";");
135         Serial.print("int fweight =");
136         Serial.print(fweight);
137         Serial.println(";");
138         Serial.print("int fspeed =");
139         Serial.print(fspeed);
140         Serial.println(";");
141         Serial.print("int tspeed =");
142         Serial.print(tspeed);
143         Serial.println(";");
144         delay(8000);
145         break;
146     }
147 }
148 // put your main code here, to run repeatedly:
149 int val_cl = analogRead(cleft_pin);
150 int val_fr = analogRead(fright_pin);
151 int val_cr = analogRead(cright_pin); // read once save time
152 int val_fl = analogRead(fleft_pin);
153 Serial.print(val_fl);
154 Serial.print(", ");
155 Serial.print(val_cl);
156 Serial.print(", ");
157 Serial.print(val_cr);
158 Serial.print(", ");
159 Serial.print(val_fr); // print vals
160 int f_ratio;
161 int c_ratio;
162 c_ratio = (val_cr-val_cl)*cweight; // get ratios
163 f_ratio = (val_fr-val_fl)*fweight;
164 Serial.print(", ");
165 Serial.print(c_ratio);
166 Serial.print(", ");
167 Serial.print(f_ratio);
168 int cr_clean = round(c_ratio*(f_ratio/abs(f_ratio))); // this was to remove wierd vals and p
169 if(cr_clean == 0){
170     cr_clean = cweight;
171 }
172 int t = ( // make a sum of c and f differences
173 //      (f_ratio*10/cr_clean)+// didnt pan out
174         c_ratio+
175         f_ratio
176         );
177 float t_ratio = (p*t);//p value
178
179 d_val = (d*(t-last_t)); // getting d from last value of t

```

```

180 last_t = t; // set last t for d next run
181 i_holder = round(t+isave*i_holder); //save the integration total for later
182 float i_val = (i_holder*i); //get i val
183 int lsped = (fspeed); // set values for old code no longer used
184 int rsped = (fspeed);
185 float mdiff = (t_ratio+d_val+i_val)/500; // get diff to turn by
186 Serial.print(", ");
187 Serial.print(mdiff);
188 //if(mdiff > 0){
189     rsped = round(fspeed-tspped*mdiff); // set speeds
190 //}
191 //if(mdiff < 0){
192     lsped = round(fspeed+tspped*mdiff);
193 //}
194
195 //int rsped = round(right_speed-right_speed*(t_ratio+d_val+i_val));
196 motors(lsped,rsped); // and push speeds to motors
197 /*
198 if(val_fl > thresh_on){ //on the line
199     left->setSpeed(left_speed);
200     right->setSpeed(right_speed);
201     if(val_fr > thresh_on){
202         right->run(RELEASE);
203         right->run(BRAKE);
204         left->run(RELEASE);
205         left->run(BRAKE);
206         delay(1000);
207         right->run(FORWARD);
208         left->run(FORWARD);
209     } else if(val_fr < thresh_off){
210         right->run(FORWARD);
211         left->run(BACKWARD);
212     }
213 } else if(val_fl < thresh_off){ // off the line
214     if(val_fr > thresh_on){
215         left->setSpeed(left_speed);
216         right->setSpeed(right_speed);
217         right->run(BACKWARD);
218         left->run(FORWARD);
219     } else if(val_fr < thresh_off){
220         int ratio = val_cr-val_cl;
221         int lsped = round(left_speed+(ratio/oneoverp));
222         int rsped = round(right_speed-(ratio/oneoverp));
223         motors(lsped,rsped);
224     }
225 }
226 } //*/
227 Serial.println("."); // end serial line
228 }

```