```
1    /*
2     * Basic Pin setup:
3     * ------------                        ---u----
4     * ARDUINO   13|-> SCLK (pin 25)           OUT1 |1    28| OUT channel 0
5     *        12|                        OUT2 |2    27|-> GND (VPRG)
6     *        11|-> SIN (pin 26)          OUT3 |3    26|-> SIN (pin 11)
7     *        10|-> BLANK (pin 23)         OUT4 |4    25|-> SCLK (pin 13)
8     *         9|-> XLAT (pin 24)            . |5    24|-> XLAT (pin 9)
9     *         8|                       . |6    23|-> BLANK (pin 10)
10    *         7|                       . |7    22|-> GND
11    *         6|                       . |8    21|-> VCC (+5V)
12    *         5|                       . |9    20|-> 2K Resistor -> GND
13    *         4|                       . |10   19|-> +5V (DCPRG)
14    *         3|-> GSCLK (pin 18)          . |11   18|-> GSCLK (pin 3)
15    *         2|                       . |12   17|-> SOUT (if daisy-chained, this goes to SIN of the next TLC)
16    *         1|                       . |13   16|-> XERR
17    *         0|                       OUT14|14   15| OUT channel 15
18    * ------------                        --------
19    *
20    * - Put the longer leg (anode) of the LEDs in the +5V and the shorter leg (cathode) in OUT(0-15).
21    * - +5V from Arduino -> TLC pin 21 and 19    (VCC and DCPRG)
22    * - GND from Arduino -> TLC pin 22 and 27    (GND and VPRG)
23    * - digital 3      -> TLC pin 18         (GSCLK)
24    * - digital 9      -> TLC pin 24         (XLAT)
25    * - digital 10     -> TLC pin 23         (BLANK)
26    * - digital 11     -> TLC pin 26         (SIN)
27    * - digital 13     -> TLC pin 25         (SCLK)
28    * - The 2K resistor between TLC pin 20 and GND will let ~20mA through each LED.
29    *
30    * If you are daisy-chaining more than one TLC, connect the SOUT of the first TLC to
31    * the SIN of the next.  All the other pins should just be connected together:
32    *   BLANK of TLC1 -> BLANK of TLC2 -> ...
33    * The one exception is that each TLC needs it's own resistor between pin 20 and GND.
34    *
35    * Alex Leone <acleone ~AT~ gmail.com>, 2008-11-26
36    */
37
38   /* These two includes should go at the top of any file that uses the library */
39   #include "tlc_config.h"
40   #include "Tlc5940.h"
41   //define round
42   #define round(x)  ((x)>=0?(long)((x)+0.5):(long)((x)-0.5))
43   //declare joystick in pins
44   int xpin = ;
45   int ypin = ;
46   int Tpin = ;
47   int zpin = ;
48   //declare x,y,T,z,channel vars
49   int x,y,T,z,chan;
50   //define percent x,y,T vars
51   float xper,yper,Tper
52   //declare Motors to x,y,T - {1,2,3,4}
53   int Motx[4] = {-1,1,1,-1};
```

```cpp
54    int Moty[4] = {-1,-1,1,1};
55    int MotT[4] = {-1,1,-1,1};
56    //declare motor power array
57    float MotPow[5];
58    //declare temp var for calculations
59    float temp;
60
61    void setup()
62    {
63      /* Tlc.init() has to be called before using any of the library functions */
64      Tlc.init();
65    }
66
67    void loop() {
68      //reset MotPow and clear old PWM settings
69      Tlc.clear();
70      MotPow = {0,0,0,0};
71
72      //gets input from joysticks stores in x,y,T,z
73      //corrects for 0-1023 and makes it -511 to 512
74      x=(analogRead(xpin)+1)-512;
75      y=(analogRead(ypin)+1)-512;
76      T=(analogRead(Tpin)+1)-512;
77      z=(analogRead(zpin)+1)-512;
78
79      //gets percentage of power for x,y,T
80      xper = x/512.0;
81      yper = y/512.0;
82      Tper = T/512.0;
83
84      //find power for each motor based on x,y,T inputs
85      for(int i=0; i<=3; i++) {
86        temp = (xper*Motx[i])+(yper*Moty[i])+(Tper*MotT[i]);
87        MotPow[i] = (temp > 1) ? 1 : temp; //if % > 1 then = 1
88      }
89      //direct mapping of z to motors 5+6
90      MotPow[4] = z;
91      MotPow[5] = z;
92
93      //loop through MotPow array and set the PWM values for each motor
94      for(int i=0; i<=5; i++) {
95        MotPow[i] = round(MotPow[i]*4095);
96        chan = i*2;
97        if(MotPow[i]<0) {
98          chan = chan+1;
99          MotPow[i] = (-1)*MotPow[i];
100       }
101       Tlc.set(chan, MotPow[i]);
102     }
103
104     //send PWM signals to tlc chip
105     Tlc.update();
106   }
```