**LED Cube Library User Manual**

Parker Jones

**Installing the Library**

All code files for the project can be found here: <https://github.gatech.edu/jjones443/LEDCube>

The most updated version of the library(as of 5/4/22) can be found in LEDCube/Code/LED\_Cube\_PWM\_Library/libraries

Copy the library folder, named LEDCube, into your local Arduino libraries folder. On my machine this is located at Documents/Arduino/libraries.

In the Arduino sketch, add the line “#include <LEDCube.h>” to add the library.

**Using the Library**

The LED Cube should be connected as follows:

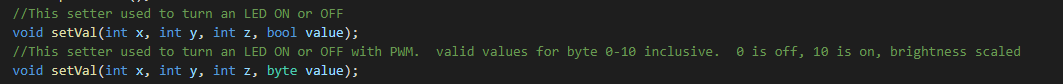
|  |  |
| --- | --- |
| Cube Pin | Arduino Nano Pin |
| VCC | 5V |
| GND | GND |
| DIO | D11 (SPI MOSI) |
| CLK | D13 (SPI SCK) |
| LOAD | Any Digital Output Pin |

Once these connections have been made, declare a LED Cube object using the constructor.



The argument for the constructor is the digital output pin connected to the LOAD pin on the LED Cube. In this case, this is D9.

Once a cube object has been declared, LEDs in the cube can be turned on or off using the following functions:



For these functions, x, y, and z are the coordinates of the LED to change. Coordinates are referenced as follows:

A picture containing indoor

Description automatically generated

+Z

+X

+Y

(0, 7, 0)

(0, 0, 7)

(7, 0, 0)

(0, 0, 0)

This function is overloaded with two different input arguments.

If not using PWM functionality, use the first function, where writing TRUE turns an LED on and FALSE turns an LED off.

If using PWM functionality, use the second function. The number written must be cast into a ‘byte’ datatype (needed for memory savings). 0 corresponds to completely OFF (equivalent to writing FALSE) while 10 corresponds to completely ON (equivalent to writing TRUE). All other values represent increasing brightness from 1-9.

Example use below:



The first write is setting a PWM value of 10 to the head LED of the snake.

The second write is setting a PWM value of 2 to the body LED of the snake.

Once the desired setVal() functions have been called, the update function must be called to update the actual cube with the new writes.



If not using PWM, this function can be called whenever the programmer wants to update the cube. For example, at the end of the main loop. See the incomplete snake demo for an example of this implementation.

If using PWM, this function should be called inside of the timer interrupt routine. See below for more information on PWM functionality.

The library also includes a getter function.

Text

Description automatically generated

This function returns TRUE if and LED is ON in any PWM state and FALSE otherwise.

NOTE: this function returns the status of the LED according to the private LED map in the library. It does not read any information from the LED Cube itself. If a getVal() is called before calling the updateCube() function, the returned value may not match the current state of the actual cube.

**Using the PWM features**

If using PWM, a hardware timer interrupt must be configured.

Instructions for setting hardware timers can be found here:

<https://www.instructables.com/Arduino-Timer-Interrupts/>

For the full snake demo, timer 0 is used. Future work should switch this to another timer, as timer 0 is used for the Arduino delay() function. Changing default parameters of this timer changes the scaling for this function, making the delay no longer in ms as seen in the full snake demo.

To set the timer, use the following function:

Graphical user interface, text, application, email

Description automatically generated

By changing the prescaler and using the formula for the OCR0A register, any frequency can be selected for this timer. From oscilloscope measurements, the fastest the Arduino nano can update the LED Cube is ~1000 Hz. This interrupt needs to be called 10 times per PWM cycle, giving a maximum refresh rate of 100Hz. However, this leaves very little time for other processor functions, so a value of ~700 Hz was chosen for a refresh rate of ~70Hz.

This function should be called in the setup function of the Arduino as seen in the full snake demo.

Once the timer interrupt has been set, the interrupt function needs to be added. See below.

Text

Description automatically generated

For PWM functionality, this interrupt should call the updateCube function. The library keeps track of which LEDs should be ON or OFF according to the PWM values set with the setVal() function.

Future work could include incorporating these setup and interrupt functions into the library. When this was attempted, errors resulted so they were not included.