

Misión Cube

Mg. Fausto M. Lagos S., † Dpto. Tecnología e Informática

Ser un hacker del movimiento maker requiere desarrollar una serie de habilidades no todas relacionadas al código fuente o la electrónica, también se requiere saber diseñar, trabajar materiales como la madera, documentar, pintar pero sobre todo requiere aprender a trabajar en equipo ya que no necesariamente este conjunto de habilidades pueden ser dominadas por una sola persona. En esta misión aprenderán a trabajar en equipo para construir tu primer proyecto de electrónica digital y desarrollarás algunas otras habilidades que son propias de todo hacker maker.

1. Introducción

En esta misión replicarás el montaje de un cubo de leds, no tendrás que inventarte nada aunque podrás mejorar el diseño e instrucciones de funcionamiento del proyecto que se te entregue. En tecnología la labor de copiar es fuente fundamental de conocimiento y constante mejora. Recibirás un proyecto terminado que tendrás que montar por etapas y hacerlo funcionar plenamente, depende de qué tanto intervengas para mejorarlo ganarás puntos para tu tabla de medallería.

2. Etapas de desarrollo y construcción

Etapa 1: Aprende

Completa la Lección 1 - en esta lección aprenderás lo básico sobre el montaje de prototipos utilizando la breadboard y Arduino, también sobre cómo soldar, diseñar y quemar el PCB para tus circuitos.

Etapa 2: Conformar tu equipo

Esta misión requiere de muchas habilidades juntas y puede que tu solo no puedas hacer con todo el trabajo que se requiere por lo tanto puedes conformar tu equipo con máximo tres compañeros que asuman cada uno un rol en el proyecto y te permitan completarlo satisfactoriamente, tu por ser quien convoca al equipo asumirás el rol *Achiever* y serás quien dirija a tus compañeros al éxito. Necesitarás un *Hacker* que se encargue de hacer funcionar la programación un *Pioneer* quien construya un proyecto realmente bonito, podrás asumir cualquier otro rol además de *Achiever* pero ten presente que reducirás el número de personas que trabajen contigo.



Cuidado: Tu eres el responsable de que tu equipo tenga éxito por tanto en cada etapa tu puntaje será superior al de tu equipo pero cuidado, si tu equipo fracasa tu serás quien más puntos pierda.

Etapa 3: Materiales

Achiever prepara todo lo necesario, eres tú el responsable de que tu equipo tenga todo lo necesario para trabajar y justo a tiempo.

- Prepara el presupuesto y adquiere los materiales. Actualiza esta lista con los precios.
 - 64 leds de 5mm o 3mm.
 - 4 resistencias de 100 Ohm cada una.
 - 1 switch o interruptor pequeño.
 - Pines macho.
 - 1 Breadboard de 60 pines.
 - 1 placa arduino.
 - 1 cable USB A macho - B macho.
 - 1 batería de 9v o batería USB.
 - 1 mts termoencogible
 - 3 mts de alambre encauchetado calibre 22.
 - 1 pinzas de punta.
 - 1 pinzas de corte.
 - 1 par de guantes de nitrilo.
 - 1 par de gafas de protección.
 - 1 cautín y soldadura de estaño.
 - 1 bayetilla.
 - 1 placa de baquelita tamaño mediano.
 - 1 esponja bombrill.
 - 2 oxido férrico.
 - 2 vasijas plásticas de tamaño mediano.
 - 2 acetato transparente.
 - 1 plancha de ropa.
- No todos estos materiales tienen que ser nuevos, aquellos materiales que puedas reciclar te ahorrarán presupuesto.

Consejo: Siempre que compres materiales eléctricos o electrónicos como leds, pulsadores, resistencias, etc. compra con redundancia i.e. compra el doble de lo necesario. Si llegase a fallar algún componente tendrás cómo reemplazarlo.



Etapa 4: Ensamblando el cubo

En esta etapa tendrá que trabajar todo tu equipo. Empieza por comprobar que todos tus leds estén funcionando correctamente, después de armados será muy incómodo tener que reemplazar alguno. Imprime la [plantilla](#) para el ensamble de cada una de las capas de leds. Ensambla el cubo de acuerdo a las instrucciones contenidas en la plantilla.



Etapa 5: Montaje del circuito

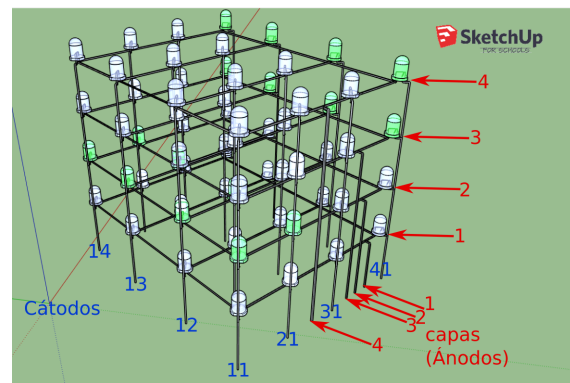
Una vez ensamblado el cubo debes comprobar que todos tus leds estan funcionando correctamente, conectarlos a Arduino y programar su funcionamiento.

1. Para comprobar el funcionamiento de cada uno de los leds conecta un jumper a la salida de 5 volts de su placa Arduino y otro a cualquier terminal de tierra. Agrega una resistencia de 100 Ohm al jumper de 5 volts. Conecta el jumper de 5 volts en cualquiera de los anodos de la primera capa y el jumper de tierra (GND) a cada uno de los 16 catodos. Deben encenderse uno a uno todos los leds de la primera capa. Repite el procedimiento con cada una de las cuatro capas.
2. Tu cubo esta configurado como una matriz 4×4 y un vector 4×1 . Conecta el cubo a Arduino tal como se muestra a continuacion, para esta conexion quema el PCB del shield cuyo diseno puedes encontrar en la carpeta [Documentos](#) en el repositorio en github. Encontraras el archivo .fzz de fritzing o los pdf listos para impresion en la carpeta [PCB.Cube](#).
3. Utiliza el archivo [PCB.Cube.etch_silk_top.pdf](#) para conocer el orden de conexion de los anodos y catodos en el shield de arduino que obtienes al quemar el PCB.



$$\text{anodos} \Rightarrow \begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{bmatrix}$$

$$\text{Catodos} \Rightarrow \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$



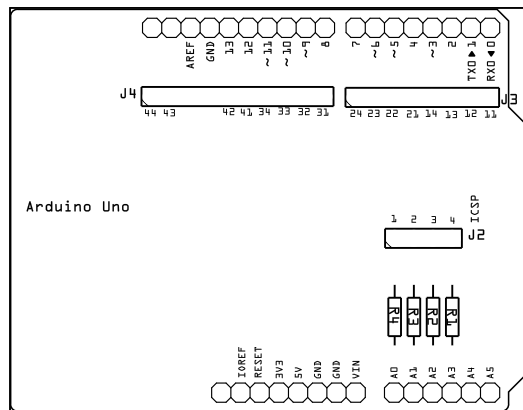
| Catodos | Pin | Catodos | Pin | Catodos | Pin | Catodos | Pin | anodos | Pin |
|----------|-----|----------|-----|----------|-----|----------|-----|---------|-----|
| 11 | D0 | 21 | D4 | 31 | D8 | 41 | D12 | 1 | A0 |
| 12 | D1 | 22 | D5 | 32 | D9 | 42 | D13 | 2 | A1 |
| 13 | D2 | 23 | D6 | 33 | D10 | 43 | A4 | 3 | A2 |
| 14 | D3 | 24 | D7 | 34 | D11 | 44 | A5 | 4 | A3 |

Etapa 6: Programacion

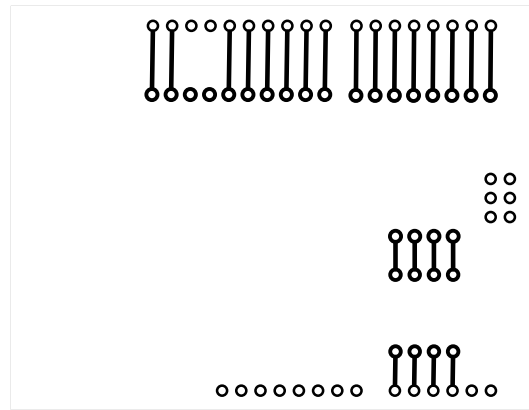
Una vez listo el PCB y la conexion del cubo con Arduino puedes descargar el codigo de ejemplo disponible en la carpeta [cube_basic_program](#) del repositorio en github y probar el funcionamiento del cubo. Este codigo de ejemplo ejecuta tres secuencias diferentes de encendido

1. **Columnas** Enciende y apaga una a una todas las columnas del cubo.
2. **Ejes** Enciende y apaga todos los leds de las caras del cubo a partir de los tres ejes X, Y y Z.
3. **Espiral** Enciende todo el cubo a partir de una espiral desde el centro.

Ninguna de estas secuencias seran tomadas en cuenta como parte de tu trabajo. Tu objetivo sera conseguir programar las secuencias de encendido mas complejas, no importa la cantidad de secuencias de encendido que



(a) *PCB_Cube_etch_silk_top*



(b) *PCB_Cube_etch_copper_bottom*

utilices sino la complejidad de éstas. Asegúrate de como en el ejemplo dejar un delay suficiente para que sean reconocidos el inicio y final de cada secuencia que programes.



donde n es el número de secuencias que programes.

```

1  /*****
2  * 2018 V1.0 Mg. Fausto M. Lagos S. * GPL V3.0
3  *
4  * Este sketch es un ejemplo de programacion para un
5  * cubo de leds 4X4X4 del curso de introduccion a Arduino
6  * disponible en
7  * https://github.com/piratax007/arduino\_course
8  *****/
9
10 /*
11  Catodos (Columnas)
12  (1, 1) - D0 | (2, 1) - D4 | (3, 1) - D8 | (4, 1) - D12
13  (1, 2) - D1 | (2, 2) - D5 | (3, 2) - D9 | (4, 2) - D13
14  (1, 3) - D2 | (2, 3) - D6 | (3, 3) - D10 | (4, 3) - A4
15  (1, 4) - D3 | (2, 4) - D7 | (3, 4) - D11 | (4, 4) - A5
16  Catodo_(X, Y) = 1 columna apagada
17  Catodo_(X, Y) = 0 columna encendida
18  -----
19  Anodos (Capas)
20  1 - PA0 | 2 - PA2 | 3 - PA2 | 4 - PA3
21  Anodo_X = 0 toda la capa estara apagada
22 */
23
24 int cathodes[16] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, A4, A5};
25 int anodes[4] = {A0, A1, A2, A3};
26
27 void setup() {
28     for (int i = 0; i < 4; i++) {
29         pinMode(anodes[i], OUTPUT);
30     }
31     for (int i = 0; i < 16; i++) {
32         pinMode(cathodes[i], OUTPUT);
33     }
34     allOff();
35 }
36
37 void loop() {
38     columns();
39     allOff();
40     delay(1000);

```



```
41 axes();
42 allOff();
43 delay(1000);
44 spiralInOut();
45 allOff();
46 delay(1000);
47 }
48
49 // Activa todas las capas
50 void allAnodesOn() {
51   for (int i = 0; i < 4; i++) {
52     digitalWrite(anodes[i], 1);
53   }
54 }
55
56 // Desactiva todas las capas
57 void allAnodesOff() {
58   for (int i = 0; i < 4; i++) {
59     digitalWrite(anodes[i], 0);
60   }
61 }
62
63 // Activa todas las columnas
64 void allCathodesOn() {
65   for (int i = 0; i < 16; i++) {
66     digitalWrite(cathodes[i], 0);
67   }
68 }
69
70 // Desactiva todas las columnas
71 void allCathodesOff() {
72   for (int i = 0; i < 16; i++) {
73     digitalWrite(cathodes[i], 1);
74   }
75 }
76
77 // Desactiva todo el cubo
78 void allOff() {
79   for (int i = 0; i < 16; i++) {
80     digitalWrite(cathodes[i], 1);
81   }
82   for (int i = 0; i < 4; i++) {
83     digitalWrite(anodes[i], 0);
84   }
85 }
86
87 // Enciende las columnas en espiral desde el centro hacia afuera
88 void spiralInOut() {
89   allAnodesOn();
90   delay(500);
91   digitalWrite(cathodes[9], 0);
92   delay(500);
93   digitalWrite(cathodes[10], 0);
94   delay(500);
95   digitalWrite(cathodes[6], 0);
96   delay(500);
97   digitalWrite(cathodes[5], 0);
98   delay(500);
99   digitalWrite(cathodes[4], 0);
100  delay(500);
101  digitalWrite(cathodes[0], 0);
102  delay(500);
103  digitalWrite(cathodes[1], 0);
104  delay(500);
105  digitalWrite(cathodes[2], 0);
106  delay(500);
107  digitalWrite(cathodes[3], 0);
108  delay(500);
109  digitalWrite(cathodes[7], 0);
110  delay(500);
111  digitalWrite(cathodes[11], 0);
```



```
112 delay(500);
113 digitalWrite(cathodes[15], 0);
114 delay(500);
115 digitalWrite(cathodes[14], 0);
116 delay(500);
117 digitalWrite(cathodes[13], 0);
118 delay(500);
119 digitalWrite(cathodes[12], 0);
120 delay(500);
121 digitalWrite(cathodes[8], 0);
122 delay(500);
123 }
124
125 // Enciende y apaga las columnas una a una
126 void columns() {
127     allAnodesOn();
128
129     for (int i = 0; i < 4; i++) {
130         digitalWrite(cathodes[i], 0);
131         delay(500);
132         digitalWrite(cathodes[i], 1);
133         digitalWrite(cathodes[i + 4], 0);
134         delay(500);
135         digitalWrite(cathodes[i + 4], 1);
136         digitalWrite(cathodes[i + 8], 0);
137         delay(500);
138         digitalWrite(cathodes[i + 8], 1);
139         digitalWrite(cathodes[i + 12], 0);
140         delay(500);
141         digitalWrite(cathodes[i + 12], 1);
142     }
143 }
144
145 // Enciende y recorre las caras del cubo en los ejes X, Y y Z
146 void axes() {
147     allAnodesOn();
148
149     for (int i = 0; i < 13; i += 4) {
150         digitalWrite(cathodes[i], 0);
151         digitalWrite(cathodes[i + 1], 0);
152         digitalWrite(cathodes[i + 2], 0);
153         digitalWrite(cathodes[i + 3], 0);
154         delay(500);
155         allCathodesOff();
156     }
157
158     delay(1000);
159
160     for (int i = 0; i < 4; i++) {
161         digitalWrite(cathodes[i], 0);
162         digitalWrite(cathodes[i + 4], 0);
163         digitalWrite(cathodes[i + 8], 0);
164         digitalWrite(cathodes[i + 12], 0);
165         delay(500);
166         allCathodesOff();
167     }
168
169     delay(1000);
170
171     for (int i = 0; i < 4; i++) {
172         allOff();
173         digitalWrite(anodes[i], 1);
174         allCathodesOn();
175         delay(500);
176     }
177 }
```

Listing 1: Código con tres secuencias de ejemplo.