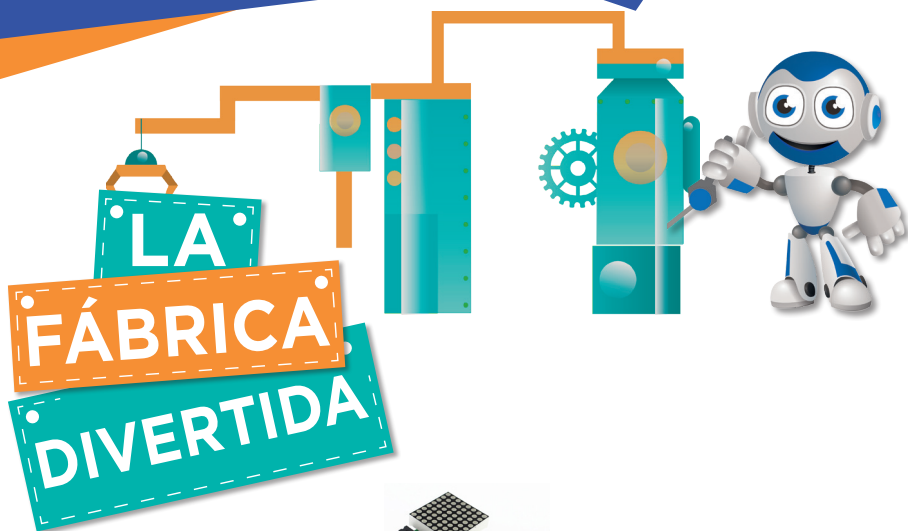
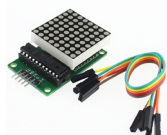


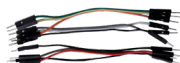
RoverLab Plus



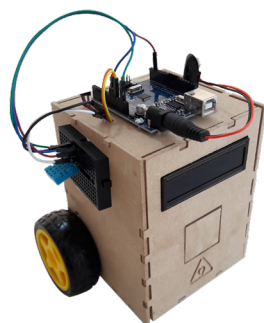
5 Cables puente hembra – hembra de 20 cm



1 Matriz de LED 8x8 MAX7219



5 Cables puente macho – macho de 10 cm

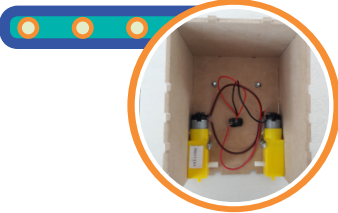


1 RoverLab de la unidad anterior

También necesitarás:

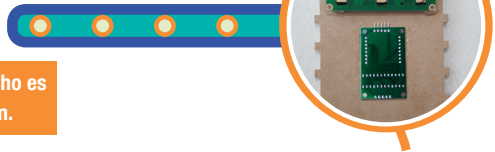
- 1 Desarmador plano
- 1 Pinza para pelar cables
- Pegamento: silicón, Kola-loka, etc.
- 1 Pila de 9 V

1 Abre la caja de tu robot RoverLab sin desconectar nada.



Nota: Esta será la versión avanzada de tu RoverLab, así que no desconectes ni desatornilles nada ya que ésta será necesaria para armar el RoverLab Plus.

2 Quita el cuadrado del rectángulo frontal de 10 x 13 cm del RoverLab y pega por dentro la matriz de LED de manera que sus pines queden del lado contrario al módulo LCD.



Nota: El extremo macho es el que tiene un pin.

3 Conecta un extremo de un cable puente hembra – hembra de 20 cm en VCC de la matriz de LED, y conecta en el extremo libre de este cable en un extremo de un cable puente macho – macho de 10 cm. Pasa este último cable por la ranura del cuadrado MDF de 10 x 10 cm y el extremo libre de éste conéctalo a [I, 11] de la mini protoboard.

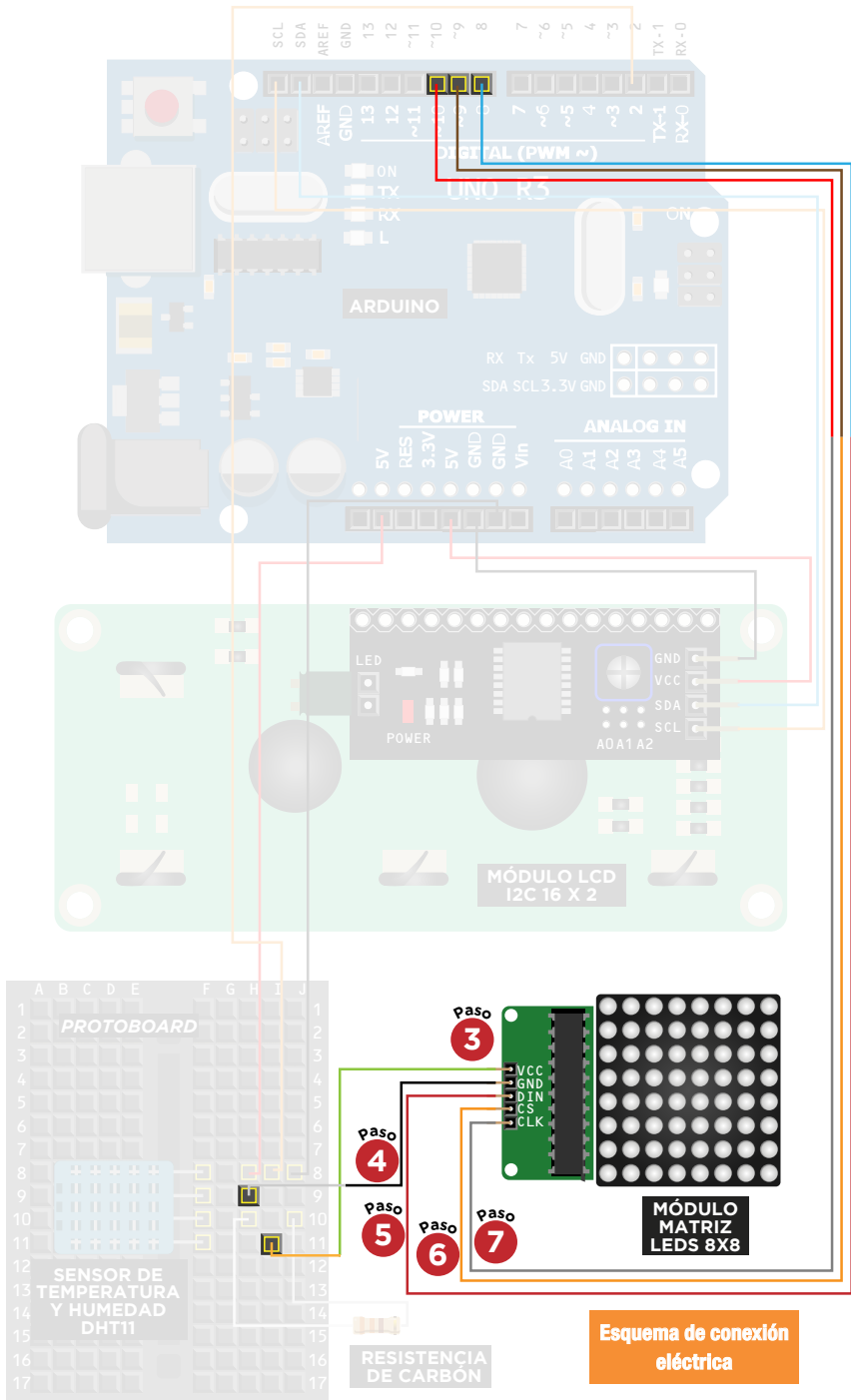
Nota: Los cables macho – macho de 10 cm que se pasan por el cuadrado MDF de 10 x 10 cm deben pasar de abajo hacia arriba, de manera que queden por el lado de la tarjeta arduino UNO para que los puedas conectar con facilidad.

Nota: El extremo hembra tiene un orificio al centro.



4 Conecta un extremo de un cable puente hembra – hembra de 20 cm en GND de la matriz de LED, y conecta en el extremo libre de este cable un extremo de un cable puente macho – macho de 10 cm. Pasa este último cable por la ranura del cuadrado MDF de 10 x 10 cm y el extremo libre de éste conéctalo en [H, 9] de la mini protoboard.

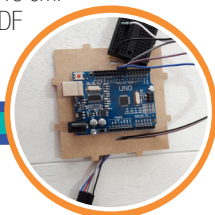
5 Conecta un extremo de un cable puente hembra – hembra de 20 cm en DIN de la matriz de LED, y conecta en el extremo libre de este cable un extremo de un cable puente macho – macho de 10 cm. Pasa este último cable por la ranura del cuadrado MDF de 10 x 10 cm y el extremo libre de éste conéctalo a la entrada DIGITAL 8 de la Arduino.



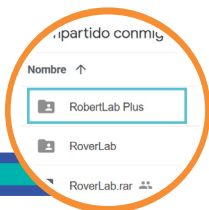
6 Conecta un extremo de un cable puente hembra – hembra de 20 cm en CS de la matriz LED, y conecta en el extremo libre de este cable un extremo de un cable puente macho – macho de 10 cm. Pasa este último cable por la ranura del cuadrado MDF de 10 x 10 cm y el extremo libre de éste conéctalo a la entrada DIGITAL ~9 de la Arduino.



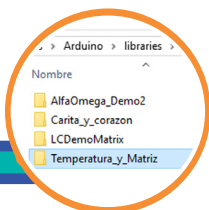
7 Conecta un extremo de un cable puente hembra – hembra de 20 cm en CLK de la matriz de LED, y conecta en el extremo libre de este cable un extremo de un cable puente macho – macho de 10 cm. Pasa este último cable por la ranura del cuadrado MDF de 10 x 10 cm y el extremo libre de éste conéctalo a ~10 de la tarjeta arduino UNO.



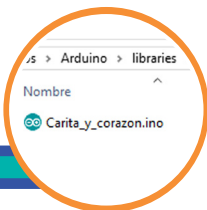
8 Da clic en la siguiente liga para descargar las librerías correspondientes en la carpeta **RoverLab Plus**, dentro de la carpeta **03 Tercer Grado** para empezar a programar <http://goo.gl/GfcqN>



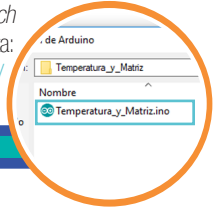
9 Copia cada carpeta en la librería de **Arduino**, en la carpeta **Documentos**:
<C:\Documentos\Arduino\libraries>



10 Abre el IDE (programa) de Arduino y carga el *sketch* **Carita_y_corazon.ino** siguiendo la ruta:
Arduino/libraries/RoverLab/Carita_y_corazon/Carita_y_corazon.ino



13 Abre el IDE (programa) de Arduino y carga el *sketch* **Temperatura_y_Matriz.ino** siguiendo la ruta: `Arduino/libraries/RoverLab/Temperatura_y_Matriz/Temperatura_y_Matriz.ino`



12 Prueba más datos a transmitir en la matriz de LED, para ello carga las diferentes librerías y prueba colocar íconos y textos, entre otras opciones.



11 Carga el *sketch* en tu RoverLab usando el botón  o da clic en el menú **Programa** y en **Subir**. Hazlo usando el cable USB y tu computadora y observa lo que aparece en la pantalla.



Nota: Cuando programes, verifica que los pines de tu programación coincidan con los de tu conexión.

```
Carita_y_corazon Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Carita_y_corazon
#include "LedControlMS.h"

LedControl lc=LedControl(8,10,9,1); // Los numeros se refieren a que pin de A
/* 8 para el DIN, 10 para el CLK, 9 para el CS y el 1 se refiere a la asigna

//Corazón pequeño
byte Corazon_datos[] = {
B00000000,
B01100110,
B11111111,
B11111111,
B01111110,
B00111100,
B00011000,
B00000000};

byte Cara_datos[] = {
```

← programación binaria necesaria para que la matriz LED proyecte un corazón

14

Revisa el videotutorial de la siguiente liga para comprender las acciones programadas en el *sketch* y prueba cambiar la frase que se muestra en la matriz de LED: <https://youtu.be/oeWPO5PSvFo>



```
Temperatura_y_Matriz
#include <MaxMatrix.h>

#include <LiquidCrystal_I2C.h>

#include <Wire.h>

#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_Sensor.h>

DHT dht(2, DHT11); /*Se crea una variable para el sensor */
LiquidCrystal_I2C lcd(0x3F,16,2); /* Se genera un objeto para identificar*/
/* La pantalla LCD */

PROGMEM unsigned char const CH[] = {
3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space
1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !
3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "
5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #
4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $
5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %
5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &
1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '
3, 8, B00011100, B01100010, B01000001, B00000000, B00000000, // (
3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, // )
5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // *
5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +
2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,
4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -
2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .
4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /
4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0
3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1
4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2
4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3
4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4
4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5
4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6
4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7
4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8
4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9
2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :
2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;
3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <
3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =
3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >
4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?
5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @
4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A
4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B
4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C
4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D
4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E
4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F
4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G
4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H
```

Carga de librerías a utilizar en la programación

Programación binaria necesaria para que la matriz LED proyecte cada letra o signo.

```

Temperatura_y_Matriz
4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // -
};

int data = 8; // DIN pin del modulo MAX7219
int load = 9; // CS pin del modulo MAX7219
int clock = 10; // CLK pin del modulo MAX7219

int maxInUse = 5; //Cambie este valor dependiendo del numero de matrices que use

MaxMatrix m(data, load, clock, maxInUse); // Define el modulo

byte buffer[10];

char mensaje1[] = " AlfaOmega "; //Escriba el mensaje a desplegar
char mensaje2[] = "Robotica";

void setup(){
  pinMode(9,OUTPUT); //Conexion a CS
  pinMode(8,OUTPUT); //Conexion a DIN
  pinMode(10,OUTPUT); //Conexion a CLK
  m.init(); // inicializa el modulo
  m.setIntensity(1); // intensidad de los puntos de la matriz, entre 1-5
  Serial.begin(9600); // inicializa el puerto serial
}

```

Edición de texto a mostrar en la matriz de LED

Nota: Este programa contiene la programación para configurar la matriz LED, sólo cambia el texto entre comillas. Puedes escribir utilizando las letras del abecedario en mayúsculas y minúsculas, así como números y signos como: admiración, comillas, más, guión, asterisco, espacio, paréntesis, corchetes, llaves, entre otros.

```

  lcd.init(); /*Sección que da inicio al display */
  lcd.backlight(); /*Se enciende la luz de fondo */
  dht.begin();
}

void loop(){

  delay(2000); /* Este comando retarda la toma de datos cada 2 segundos */
  lcd.setCursor(0,0); /*se declara la posición de columna y fila */
  lcd.print("Temperatura:"); /* El display mostrará texto y valor */
  lcd.print(dht.readTemperature()); /**valor de la temperatura */

  lcd.setCursor(0,1); /* El cursor cambia a la segunda fila */
  lcd.print("Humedad: "); /*Se muestra el texto y valor de la humedad */
  lcd.print(dht.readHumidity()); /*valor de la humedad relativa*/
}

```

Programación del módulo LCD con el sensor de temperatura

```

// Despliega los mensajes almacenados en las variables

printStringWithShift(mensaje1, 100); // El ultimo termino se usa para la velocidad del
printStringWithShift(mensaje2, 100);

}

void printCharWithShift(char c, int shift_speed){ // Imprime caracteres
  if (c < 32) return;
  c -= 32;
  memcpy_P(buffer, CH + 7*c, 7);
  m.writeSprite(maxInUse*8, 0, buffer);
  m.setColumn(maxInUse*8 + buffer[0], 0);
}

```

Manejo de variables para su óptimo funcionamiento

```

m.setColumn(maxInUse*8 + buffer[0], 0);

for (int i=0; i<buffer[0]+1; i++)
{
  delay(shift_speed);
  m.shiftLeft(false, false);
}
}

void printStringWithShift(char* s, int shift_speed) { // Imprime cadena de caracteres
while (*s != 0){
  printCharWithShift(*s, shift_speed);
  s++;
}
}

void printString(char* s) // Imprime cadena
{
  int col = 0;
  while (*s != 0)
  {
    if (*s < 32) continue;
    char c = *s - 32;
    memcpy_P(buffer, CH + 7*c, 7);
    m.writeSprite(col, 0, buffer);
    m.setColumn(col + buffer[0], 0);
    col += buffer[0] + 1;
    s++;
  }
}

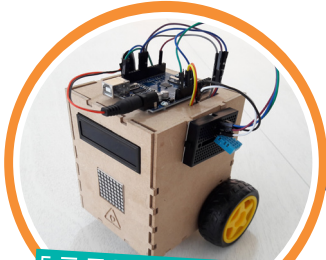
```

Manejo de variables para su óptimo funcionamiento

15 Carga el *sketch* en tu RoverLab Plus usando el botón  o da clic en el menú **Programa** y en **Subir**. Hazlo usando el cable USB y tu computadora. Observa lo que aparece en la pantalla.



16 Pon en funcionamiento tu robot colocándole una pila de 9 V al broche del portapila para que dé el clima y enciende el mini *switch* para que avance.



Robot terminado

 **Alfaomega Grupo Editor**
Te acerca al conocimiento

www.alfaomega.com.mx
Soporte escolar

e-mail: soportescolar@alfaomega.com.mx

Si te resulta complicado cargar la programación, contáctanos en www.robotik-age.com