Hola amigo, gracias por descargar este archivo,

te agradecería que te suscribas y que le des «me gusta» a mi cuenta de youtube

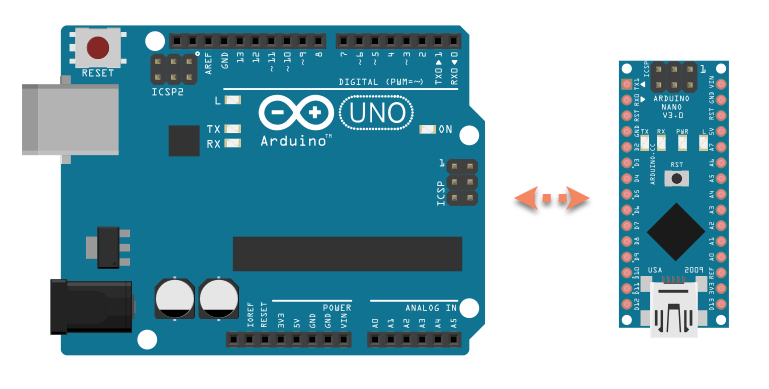
la verdad es la primera vez que hago esto y no sé si me vaya a ayudar mucho :v pero igual agradezco tu ayuda y el apoyo que me brindas solo con eso.

Cabe resaltar que la información no ha sido 100% mia y que he buscado por internet, recolectando información para poderles traer algo un poco más completo, espero les guste.

Aprovecha que es GRATIS.

A continuación, te describiré los materiales que necesitas para crear tu tablero 8x8:

1. Lo primero necesitas es Arduino UNO o Arduino NANO

[](http://robots-argentina.com.ar/didactica/wp-content/uploads/comparacion-UNO-NANO-1.png)

En la parte izquierda vemos Arduino UNO | en la parte derecha vemos a Arduino nano

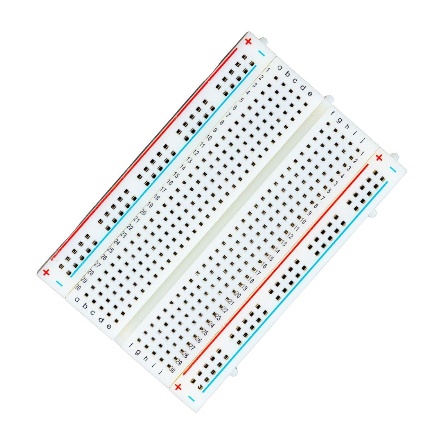
1. el cable de datos Arduino

Imagen que contiene cable

imagen tomada de https://ae01.alicdn.com/ (este cable es para arduino uno)

En la parte izquierda esta el cable de datos de Arduino uno | en la parte derecha este el cable de Arduino nano

1. Dos protoboard

Protoboard mediana
imagen tomada de https://cdn.shopify.com/

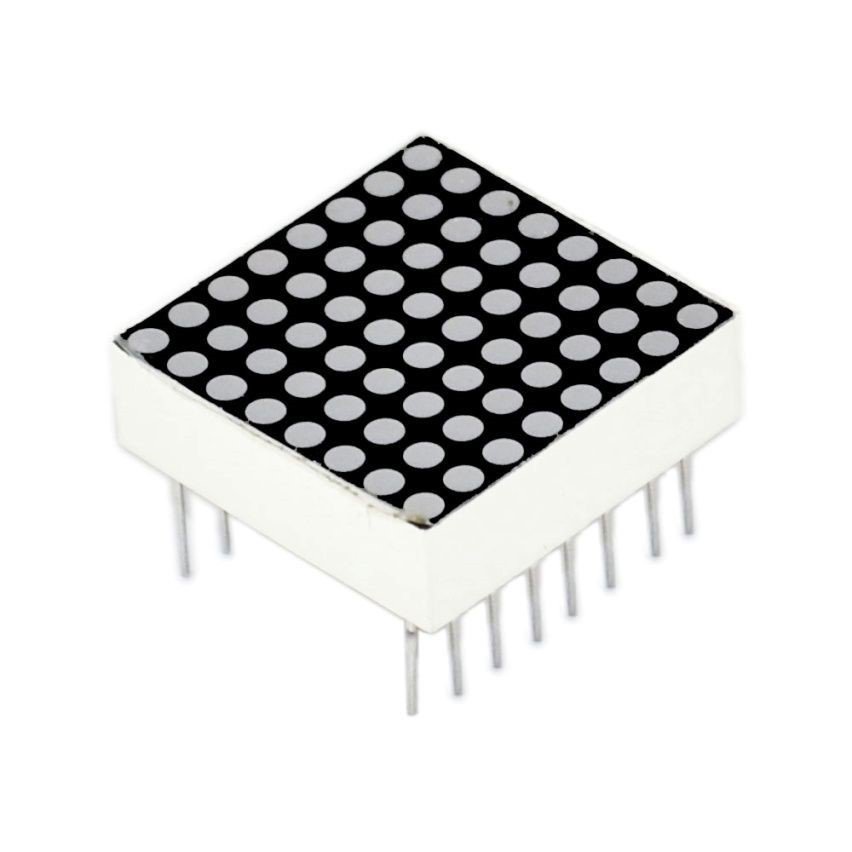
En este caso no hay diferencia y si notan en la parte de debajo de sus protoboards van a encontrar unas pestañitas y unos huequitos en tal caso esto es para unirlos, si no tienen protoboard no se preocupenen el siguiente punto encontraran otra opción

1. 8 Jumpers macho-macho o macho-hembra

estos son jumpers macho- hembra 
la diferencia es que el macho tiene las punticas las cuales se insertan en la protoboard o en tu arduino y la hembra es para conectar los dispositivos en este caso una tabla 8x8
imagen tomada de https://cdn.shopify.com/

En este caso vemos a mano izquierda jumpers macho-macho y a lado derecho jumpers macho- hembra en caso que no tengas la protoboard puedes usar macho-hembra para hacer la conexión

1. Matriz led 8x8 en este caso no vamos a usar el max 7219 ya que no lo tengo :’( soy pobre :v



Matriz led 8x8, la parte frontal es la que tiene un numero marcado en la parte de adelante

Una vez tengamos todos nuestros componentes lo que vamos a hacer es ponernos manos a la obra.

Lo primero es nuestras conexiones, en este caso acá esta la imagen de las conexiones

Diagrama

Descripción generada automáticamente

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pin Arduino | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| matriz 8x8 | atrás 5 | adelante 6 | adelante 5 | atrás 2 | adelante3 | atrás 3 | atrás 7 | atrás 8 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pin Arduino | 10 | 11 | 12 | 13 | a0 | a1 | a2 | a3 |
| matriz 8x8 | atras1 | atrás 6 | adelante 1 | atrás 4 | adelante 8 | adelante 2 | adelante 7 | adelante 4 |

Recuerda que la parte de adelante es donde esta el rotulo o los números

En este caso no necesitamos más si no esto y nuestro código.

Abre un proyecto para Arduino Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Y allí puedes eliminar todo el contenido con las teclas ctrl + a

Imagen que contiene Texto

Descripción generada automáticamente

Ahora acá esta el código

//SE DECLARA ARRAY DE BYTES CON LOS VALORES NECESARIOS

//PARA FORMAR CADA LETRA 0-> APAGADO, 1->ENCENDIDO

//SON 8 VALORES, UNO POR CADA FILA

//COMO USO LA CALCULADORA PARA HACER UN NUEVO CARACTER ?

//DEBES IR CON CTRL+R Y PONER CALC

//UNAVEZ ALLÍ LE DAS CLIC A LAS TRES LINEAS Y SELECCIONAS PROGRAMADOR

//LUEGO SELECCIONA LA OPCIÓN DE BIN(BINARIO)

// ALLI DEBES TENER EN CUENTA QUE DEBES PONER QUE LED QUIERES QUE ESTE ENCENDIDO

//EJEMPLO SI QUIERES QUE HAYA UN PUNTO ENCENDIDO EN TU LED 8X8

//DEBES PONER 10000000 Y REVISA EL VALOR QUE TE DA LA CALCULADORA EN HEX(HEXADECIMAL)

//ESE ES EL VALOR QUE PINTARAS EN LA PRIMERA LINEA

byte A[] = { 0x3c, 0x42, 0x42, 0x7E, 0x7E, 0x42, 0x42, 0x42 };//

byte B[] = { 0x3c, 0x42, 0x42, 0x7c, 0x42, 0x42, 0x42, 0x3c };//

byte C[] = { 0x1e, 0x20, 0x40, 0x40, 0x40, 0x40, 0x20, 0x1e };//

byte D[] = { 0x78, 0x7C, 0x6E, 0x66, 0x66, 0x6E, 0x7C, 0x78 };//

byte E[] = { 0x7C, 0x40, 0x40, 0x78, 0x78, 0x40, 0x40, 0x7C }; //

byte F[] = { 0x7C, 0x40, 0x40, 0x78, 0x78, 0x40, 0x40, 0x40 };//

byte G[] = { 0x1e, 0x20, 0x40, 0x5c, 0x42, 0x42, 0x22, 0x1c };//

byte H[] = { 0x42, 0x42, 0x42, 0x7E, 0x7E, 0x42, 0x42, 0x42 };//

byte I[] = { 0x7E, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18, 0x7E };//

byte J[] = { 0x3e, 0xc, 0xc, 0xc, 0xc, 0xc, 0x4c, 0x7c };//

byte K[] = { 0x63, 0x66, 0x6C, 0x78, 0x78, 0x6C, 0x66, 0x40 }; //

byte L[] = { 0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x7e, 0x7e };//

byte M[] = { 0x66, 0x7E, 0x5A, 0x42, 0x42, 0x42, 0x42, 0x42 }; //

byte N[] = { 0x42, 0x42, 0x62, 0x52, 0x4a, 0x46, 0x42, 0x42 };//

byte O[] = { 0x3C, 0x42, 0x42, 0x42, 0x42, 0x42, 0x42, 0x3C }; //

byte P[] = { 0x3c, 0x42, 0x42, 0x7c, 0x40, 0x40, 0x40, 0x40 }; //

byte Q[] = { 0x3c, 0x42, 0x42, 0x42, 0x42, 0x4a, 0x44, 0x3a };//

byte R[] = { 0x3c, 0x42, 0x42, 0x7c, 0x44, 0x42, 0x42, 0x42 };//

byte S[] = { 0x3c, 0x40, 0x40, 0x3c, 0x2, 0x2, 0x2, 0x3c }; //

byte T[] = { 0x7e, 0x5a, 0x18, 0x18, 0x18, 0x18, 0x18, 0x18 };//

byte U[] = { 0x42, 0x42, 0x42, 0x42, 0x42, 0x42, 0x42, 0x3c };//

byte V[] = { 0x42, 0x42, 0x42, 0x42, 0x42, 0x42, 0x24, 0x18 };//

byte W[] = { 0x42, 0x42, 0x42, 0x42, 0x5a, 0x5a, 0x5a, 0x3c };//

byte X[] = { 0x42, 0x42, 0x24, 0x24, 0x18, 0x24, 0x42, 0x42 };//

byte Y[] = { 0xC3, 0xC3, 0x66, 0x3C, 0x18, 0x18, 0x18, 0x18 };//

byte Z[] = { 0x7e, 0x46, 0xc, 0xc, 0x18, 0x30, 0x62, 0x7e };

byte face[] = { 0x8, 0x4, 0x42, 0x2, 0x2, 0x42, 0x4, 0x8 };

byte qu[] = { 0x18, 0x24, 0x42, 0xc, 0x18, 0x18, 0x00, 0x18 };

byte uno[]={0x18,0x38,0x78,0x18,0x18,0x18,0xff,0xff};

byte dos[]={0x1c,0xff,0xe7,0x0e,0x1c,0x38,0x7f,0xff};

byte tres[]={0x7c,0x7e,0x07,0x7f,0x7f,0x07,0x7e,0x7c};

byte cuatro[]={0x66,0x66,0x66,0x7e,0x7e,0x06,0x06,0x06};

byte cinco[]={0x7e,0x7e,0x60,0x7c,0x7e,0x02,0x7e,0x7c};

byte seis[]={0x7e,0x7e,0x60,0x7c,0x7e,0x62,0x7e,0x3c};

byte siete[]={0x7e,0x7e,0x06,0x0c,0x18,0x30,0x60,0xc0};

byte ocho[]={0x7e,0xff,0xc3,0xff,0xff,0xc3,0xff,0x7e};

byte nueve[]={0x7e,0xff,0xc3,0xff,0x7f,0x03,0x7f,0x7e};

byte cero[]={0xff,0xff,0xc7,0xcb,0xd3,0xe3,0xc3,0xff};

//ESTE ARRAY ES PARA LIMPIAR MATRIZ

byte sp[]= { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };

//SE INICIALIZA LOS PINES COMO SALIDA

void setup() {

for (int j=2; j<19; j++)

pinMode(j, OUTPUT);

Serial.begin(9600);

}

//FUNCION PARA IMPRIMIR LETRA EN MATRIZ DE LEDS

void SetChar(char p) {

Clear(); //FUNCION PARA LIMPIAR MATRIZ

for (int fil = 0; fil <8 ; fil++) {

digitalWrite( fil + 10 , HIGH) ;

byte F = Selecciona( p, fil);

for (int col =7; col >= 0 ; col--) {

digitalWrite(8-col, HIGH);

bool b = GetBit(F, col);

if (b)

digitalWrite( 9 - col ,LOW);

else

digitalWrite( 9 - col ,HIGH);

}

digitalWrite( fil + 10 , LOW) ;

}

}

//FUNCIÓN PARA CONVETIR HEXADECIMAL A BINARIO

bool GetBit( byte N, int pos) {

int b = N >> pos ;

b = b & 1 ;

return b ;

}

//FUNCIÓN PARA LIMPIAR MATRIZ-TODOS LOS LEDS APAGADOS

void Clear() {

for (int j=2; j<10; j++)

digitalWrite(j, HIGH);

for (int k= 10 ; k<18 ; k++)

digitalWrite(k, LOW);

}

//SELECCIONA FILA DE ARRAY DE LETRA A IMPIRMIR

byte Selecciona( char c, byte fil) {

if (c == 'A') return( A[fil]);

if (c == 'B') return( B[fil]);

if (c == 'C') return( C[fil]);

if (c == 'D') return( D[fil]);

if (c == 'E') return( E[fil]);

if (c == 'F') return( F[fil]);

if (c == 'G') return( G[fil]);

if (c == 'H') return( H[fil]);

if (c == 'I') return( I[fil]);

if (c == 'J') return( J[fil]);

if ( c == 'K') return(K[fil]);

if ( c == 'L') return(L[fil]);

if (c == 'M') return( M[fil]);

if (c == 'N') return( N[fil]);

if (c == 'O') return( O[fil]);

if (c == 'P') return( P[fil]);

if (c == 'Q') return( Q[fil]);

if (c == 'R') return( R[fil]);

if (c == 'S') return( S[fil]);

if (c == 'T') return( T[fil]);

if (c == 'U') return( U[fil]);

if (c == 'V') return( V[fil]);

if (c == 'W') return( W[fil]);

if (c == 'X') return( X[fil]);

if (c == 'Y') return( Y[fil]);

if (c == 'Z') return( Z[fil]);

if (c == ' ') return( sp[fil]);

if (c == ':') return( face[fil]);

if (c == '?') return( qu[fil]);

if (c == '1') return( uno[fil]);

if (c == '2') return( dos[fil]);

if (c == '3') return( tres[fil]);

if (c == '4') return( cuatro[fil]);

if (c == '5') return( cinco[fil]);

if (c == '6') return( seis[fil]);

if (c == '7') return( siete[fil]);

if (c == '8') return( ocho[fil]);

if (c == '9') return( nueve[fil]);

if (c == '0') return( cero[fil]);

}

void loop(){

String s = " SUSCRIBETE Y DALE LIKE ES GRATIS " ; //PALABRA A IMPRIMIR

int l = s.length();// CALCULO DE LONGITUD DE PALABRA

for ( int n = 0; n< l; n++ ) { //RECORRE PALABRA E IMPRIME EN MATRIZ

long t = millis();

char c = s[n]; //RECORRE PALABRA Y OBTIENE LETRA POR LETRA

while ( millis()< t+ 800)

SetChar(c); //LLAMADA A FUNCIÓN QUE IMPRIME EN MATRIZ

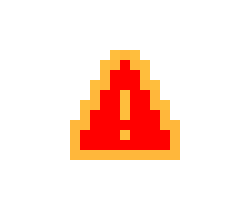
}

}

Imagen que contiene clip, estacionaria, interior, foto

Descripción generada automáticamente

Ya con esto seria todo podrás poner letras y adicione un “emoticono”, espero lo disfrutes y si quieres otro con los números solo comenta y suscríbete.



en tal caso que no funcione que te salga un error de compilación al abrir el proyecto nuevamente oprime ctrl+n copia y pega el código.

Ten en cuenta que el programa ha funcionado en mayúsculas es decir en el viod loop() en la parte de // palabra a imprimir, en este espacio se debe escribir en mayúscula, a menos que cambies las variables que hemos declarado al inicio del programa y los if ejemplo:

**byte A[] = { 0x3c, 0x42, 0x42, 0x7E, 0x7E, 0x42, 0x42, 0x42 };//**esta es nuestra primer línea o la primera variable la cual hemos declarado, para cambiarla a minúscula solo debemos hacer

**byte a[] = { 0x3c, 0x42, 0x42, 0x7E, 0x7E, 0x42, 0x42, 0x42 };//** adicional debemos buscar el if que nos trae esta variable así:

**if (c == 'A') return( A[fil]);** // y cambiarlo como lo hicimos anteriormente

**if (c == 'a') return( a[fil]);** en este caso lo que va dentro de ' ' es lo que nosotros en el campo de palabra a imprimir **String s = " SUSCRIBETE AMIGO ES GRATIS " ; //** es decir que si nosotros solo ponemos **'a'** en este caso ya cambiada la variable este nos llamara o nos retornara a fill, que fill hace referencia a la variable o a la cadena tipo char que declaramos anteriormente es decir al byte.

## Bibliografía

Imagen tomada de:

<http://robots-argentina.com.ar/didactica/wp-content/uploads/comparacion-UNO-NANO-1.png>

imagen tomada de:

<https://www.e-ika.com/images/thumbs/0006647_cable-usb-a-miniusb-30cm-para-arduino-nano.jpeg>

imagen tomada de:

<https://ae01.alicdn.com/kf/HTB1ukXeXfvi21VjSZK9q6yAEpXaM/1-pieza-Mini-30cm-Cable-USB-3-0-para-Arduino-Nano.jpg_Q90.jpg_.webp>

imagen tomada de:

<https://electronicamade.com/wp-content/uploads/2020/04/Protoboard.jpg>

imagen tomada de:

<https://robotecashop.com/wp-content/uploads/2021/11/Protoboard-1.jpg>

imagen tomada de:

<https://smoothie.com.co/images/male-male-jumpers.jpg>

imagen tomada de:

<https://cdn.shopify.com/s/files/1/1040/8806/products/photo_OS-00071_Jumper_M-F_20_02_700x700.png?v=1598042061>

imagen tomada de:

<https://www.makercreativo.com/store/wp-content/uploads/2018/06/Matriz-led-8x8-3mm.jpg>

imagen tomada de:

<https://img2.freepng.es/20190526/hkr/kisspng-paper-clip-invention-by-design-google-sheets-invention-by-design-how-engineers-get-from-though-5ceb2257d4a083.7676000415589136238709.jpg>

toda la información ha sido basada en:

<https://img2.freepng.es/20190526/hkr/kisspng-paper-clip-invention-by-design-google-sheets-invention-by-design-how-engineers-get-from-though-5ceb2257d4a083.7676000415589136238709.jpg>