



KEYESTUDIO SMART HOME STEM KIT

FOR ARDUINO KIT

PERE MANEL VERDUGO ZAMORA

pereverdugo@gmail.com

Contenido

KS0085 Keystudio de hogar inteligente	2
1.- Introducción	2
2.- Lista de equipos	2
3.- Descargue software	6
4.- Placa de control Keystudio PLUS.....	8
5.- Instalación del controlador	9
6.- Configuración del IDE de Arduino	12
7.- Comienza tu primer programa.....	15
8.- ¿Cómo agregar una biblioteca?	16
9.- Proyectos.....	19
Proyecto 1: Parpadeo LED	19
Proyecto 2: Respirando Luz.....	24
Proyecto 3: Zumbador pasivo	28
Proyecto 4: Control de LED por módulo de botón	36
Proyecto 5: Módulo de relé de 1 canal	38
Proyecto 6: Sensor de fotocélula	41
Proyecto 7: Ajuste del ángulo del servomotor.....	44
Proyecto 8: Módulo de ventilador	47
Proyecto 9: Sensor de vapor	49
Proyecto 10: Sensor de movimiento PIR.....	52
Proyecto 11: Sensor analógico (MQ-2)	55
Proyecto 12: Pantalla LCD 1602	58
Proyecto 13: Sensor de humedad del suelo.....	60
Proyecto 14: Prueba de Bluetooth.....	64
10.- Guía de montaje.....	72
Proyecto final: kit de hogar inteligente multiusos	90

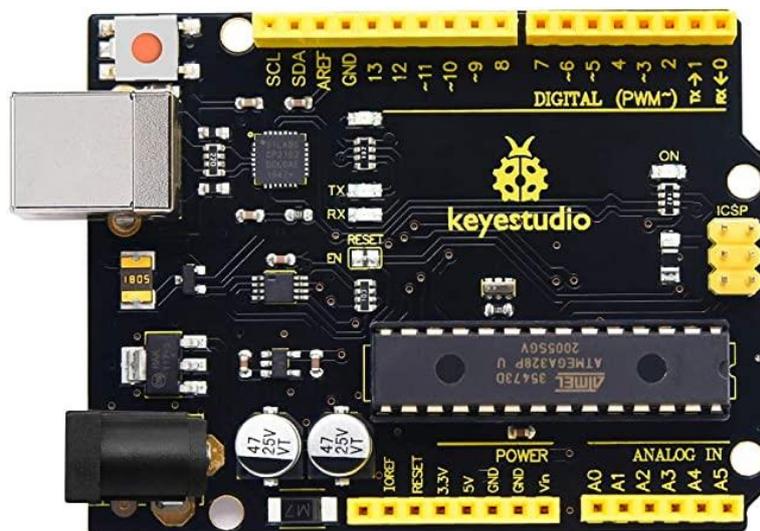
KS0085 Keystudio de hogar inteligente



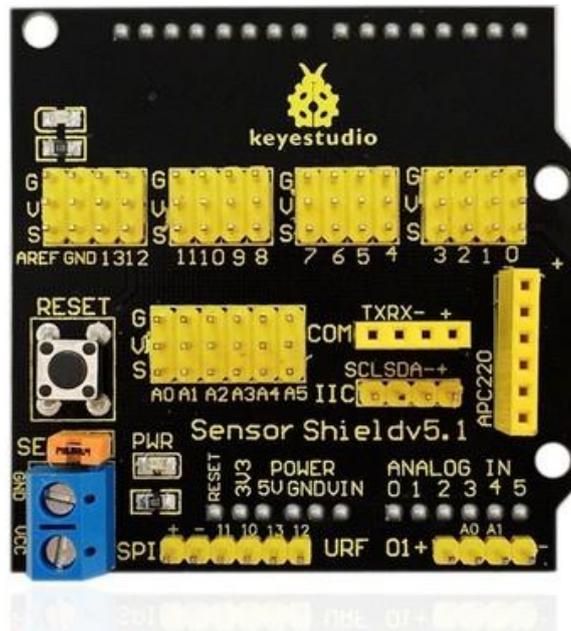
1.- Introducción

El hogar inteligente es una tendencia en el futuro. Imagina que te acuestas en el sofá y controlas todo con un teléfono celular común y corriente. ¿Tan increíble! Las cortinas se pueden cambiar automáticamente, la temperatura del agua puede cambiar al modo templado cuando está listo para un baño acogedor. Alternativamente, puede obtener la escena de luz distinta en su hogar. Creo que no puedes esperar para vivir así. En esta lección, simulamos el hogar inteligente para hacer un kit de hogar inteligente de bricolaje con el software Arduino. ¡Síguenos para disfrutar de la alegría del bricolaje!.

2.- Lista de equipos

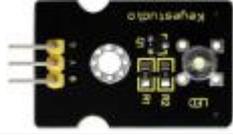
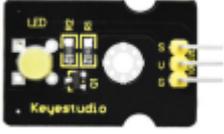


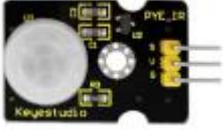
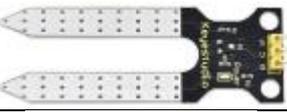
PLUS Control Board (Compatible con Arduino UNO).



Keystudio Sensor Shield.



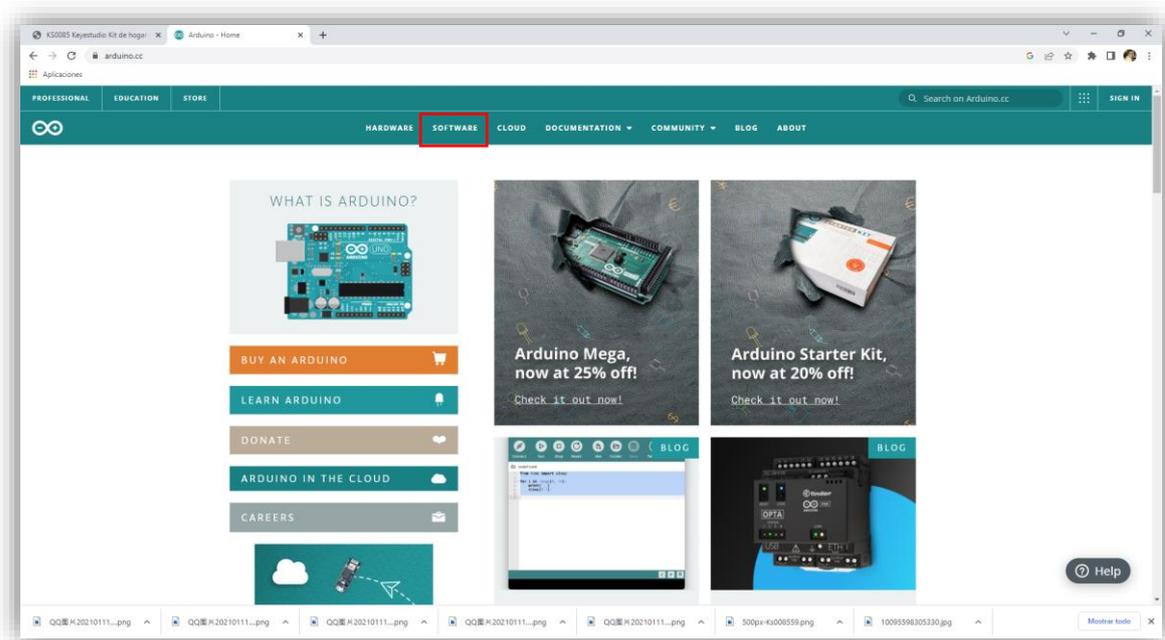
Descripción	Cantidad	Fotografía
Módulo de LED blanco.	1	
Módulo de LED Amarillo.	1	
Botón sensor	2	

Sensor de fotocélula.	1	
Sensor de movimiento PIR.	1	
Sensor de gas MQ-2.	1	
Módulo de relé.	1	
Módulo Bluetooth HM-10.	1	
Sensor de zumbador pasivo.	1	
Módulo de ventilador	1	
Sensor de vapor	1	
Servo motor.	2	
Módulo de pantalla LCD1602.	1	
Sensor de humedad del suelo.	1	
Cable USB	1	
Líneas Dupont hembra a hembra.	40	

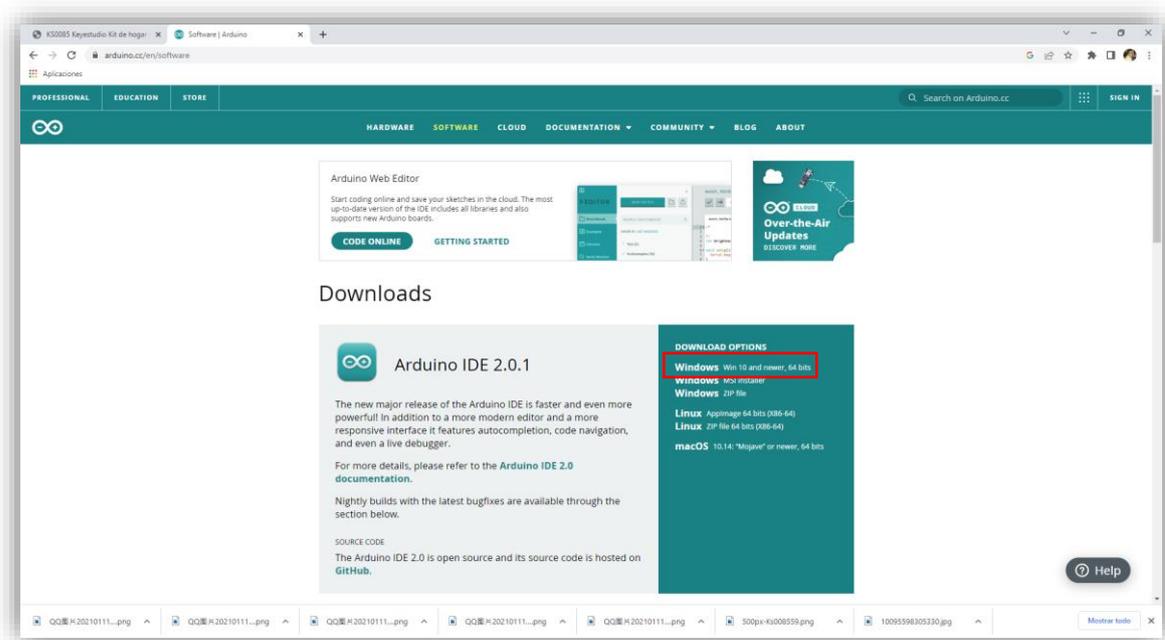
Línea Dupont Macho a hembra.	6	
Tuerca niquelada M3	25	
Tornillo de cabeza redonda M2*12MM	6	
Tuerca niquelada M2.	6	
Pilar de cobre de doble paso M3*10MM.	4	
Tornillo de cabeza redonda M3*6MM.	8	
Tuerca autoblocante de acero inoxidable M3 304.	4	
Tornillo de cabeza redonda M3*10MM.	20	
Tornillo de cabeza redonda M2.5*10MM.	6	
Cabeza de níquel M2.5.	6	
Tornillo de cabeza redonda M3*12MM.	6	
Tornillo de cabeza plana M3*10MM.	2	
Tornillo autor roscable de cabeza redonda M1.2MM.	10	
Porta pilas AA de 6 celdas con cabezal de CC y línea de rocío de 15 cm.	1	
Destornillador cruzado con mango negro-amarillo de 3*40MM.	1	
Cable puente H-H de 2,54 de 3 pines de 20 cm.	13	
Cable puente H-H de 2,54 de 4 pines de 20 cm.	2	

3.- Descargue software

Cuando obtengamos la placa de control, primero debemos descargar Arduino IDE y el controlador. Puede descargar Arduino IDE desde el sitio web oficial: <https://www.arduino.cc/>



Hay varias versiones para Arduino, simplemente descargue una versión adecuada para su sistema, tomaremos el sistema Windows como ejemplo para mostrarle cómo descargar e instalar.



DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux Appliance 64 bits (X86-64)

Linux ZIP file 64 bits (X86-64)

macOS 10.14: "Mojave" or newer, 64 bits

Hay dos versiones para el sistema Windows, una es la versión instalada, otra es la versión de descarga, solo necesitas descargar el archivo al ordenador y descomprimirlo. Estas dos versiones se puede utilizar normalmente. Elija uno y descárguelo en su ordenador.

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **67.058.097** times — impressive! Help its development with a donation.

\$3 \$5 \$10 \$25 \$50 Other

JUST DOWNLOAD **CONTRIBUTE & DOWNLOAD**

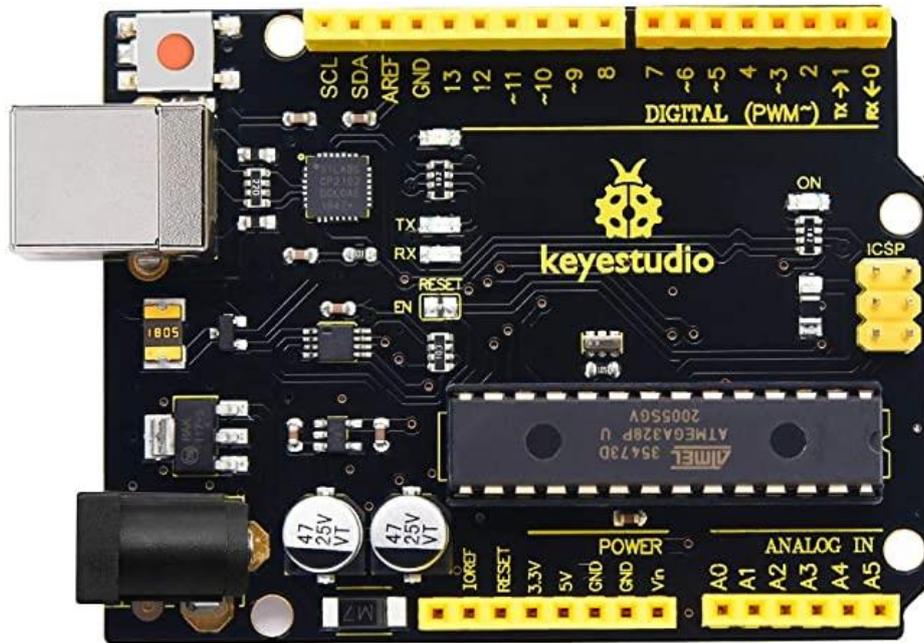


Learn more about [donating to Arduino](#).

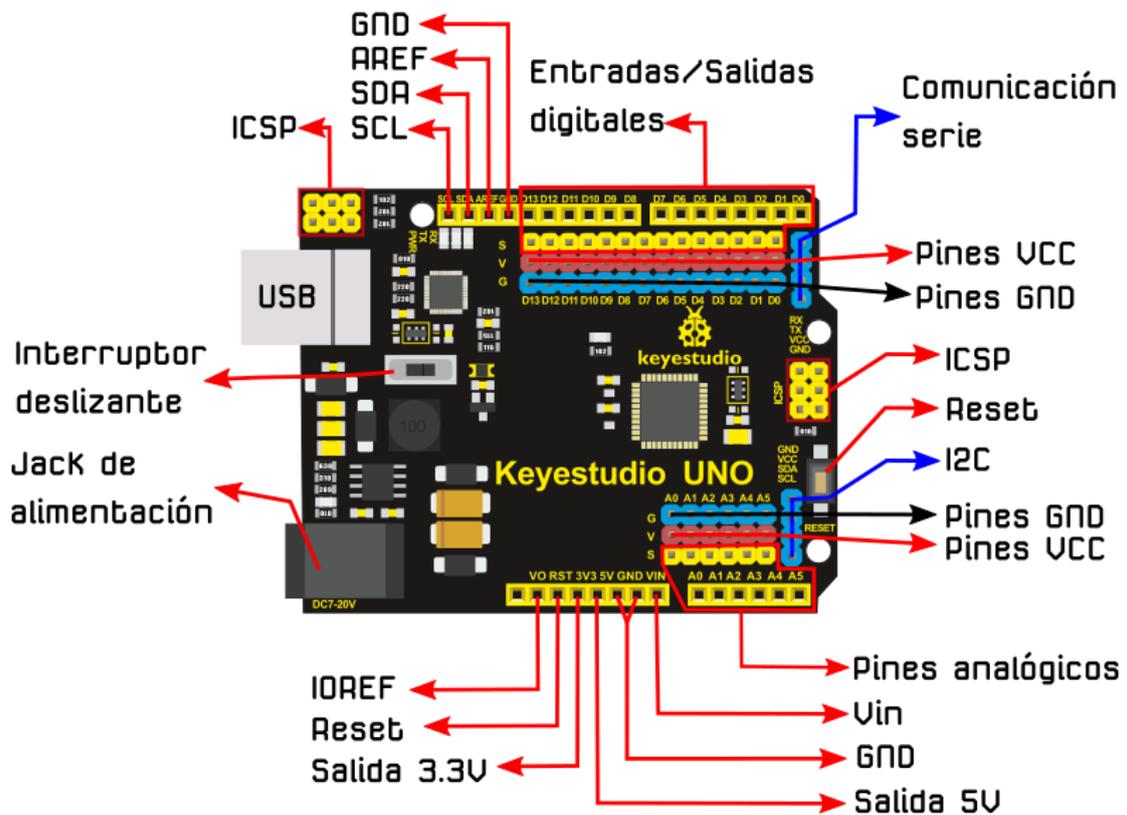
Solo necesitas hacer clic en SOLO DESCARGA, luego haga clic en el archivo descargado para instalarlo. Y cuando se descarga el archivo ZIP, puede descomprimirlo e iniciarlo directamente.

4.- Placa de control Keystudio PLUS.

Descripción



La placa de control Keystudio PLUS es totalmente compatible con el entorno de desarrollo Arduino IDE. Contiene todas las funciones de la placa Arduino UNO. Además, algunas mejoras que hicimos fortalecen mucho su función. Es la mejor opción para aprender a construir circuitos y escribir código también. ¡Empecemos!

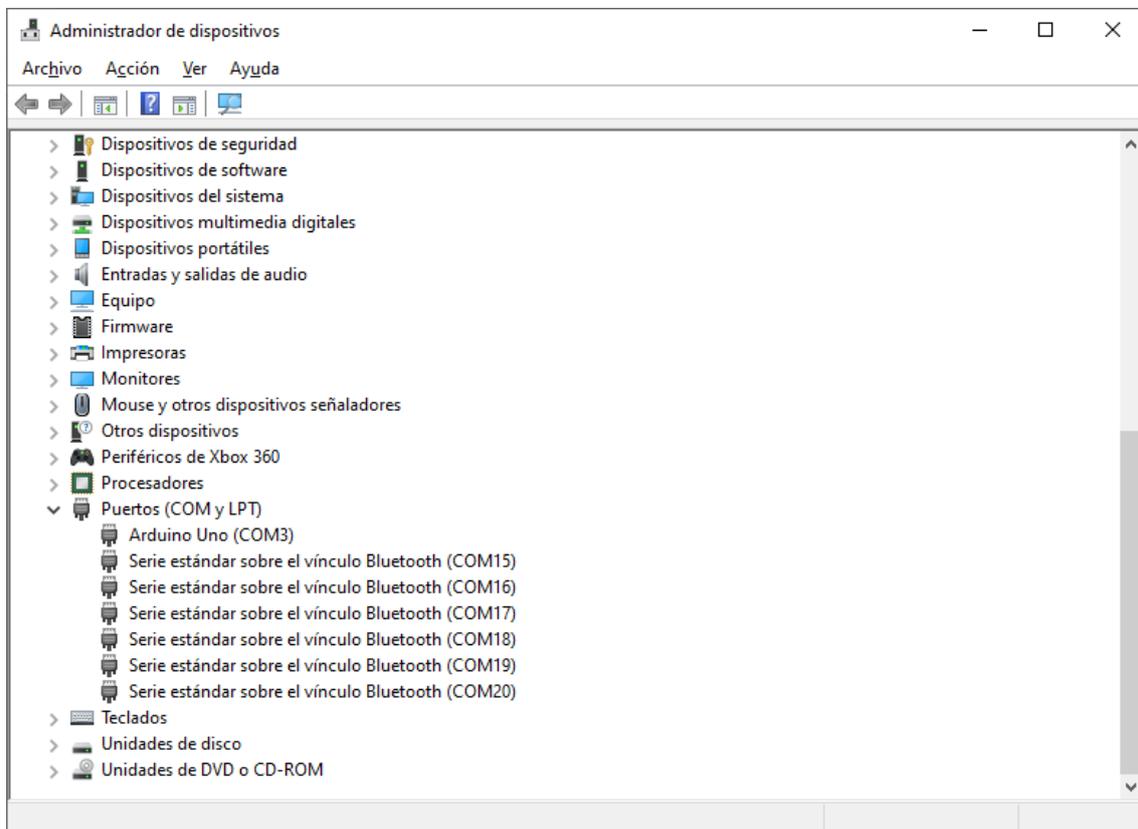


- Interfaz de comunicación en serie: D0 es RX, D1 es TX.
- Interfaz PWM (modulación de ancho de pulso): D3 D5 D6 D9 D10 D11
- Interfaz de interrupción externa: D2 (interrupción 0) Y D3 (interrupción 1)
- Interfaz de comunicación SPI: D10 es SS, D11 es MOSI, D12 es MISO, D13 es SCK.
- Puerto de comunicación IIC: A4 es SDA, A5 es SCL.

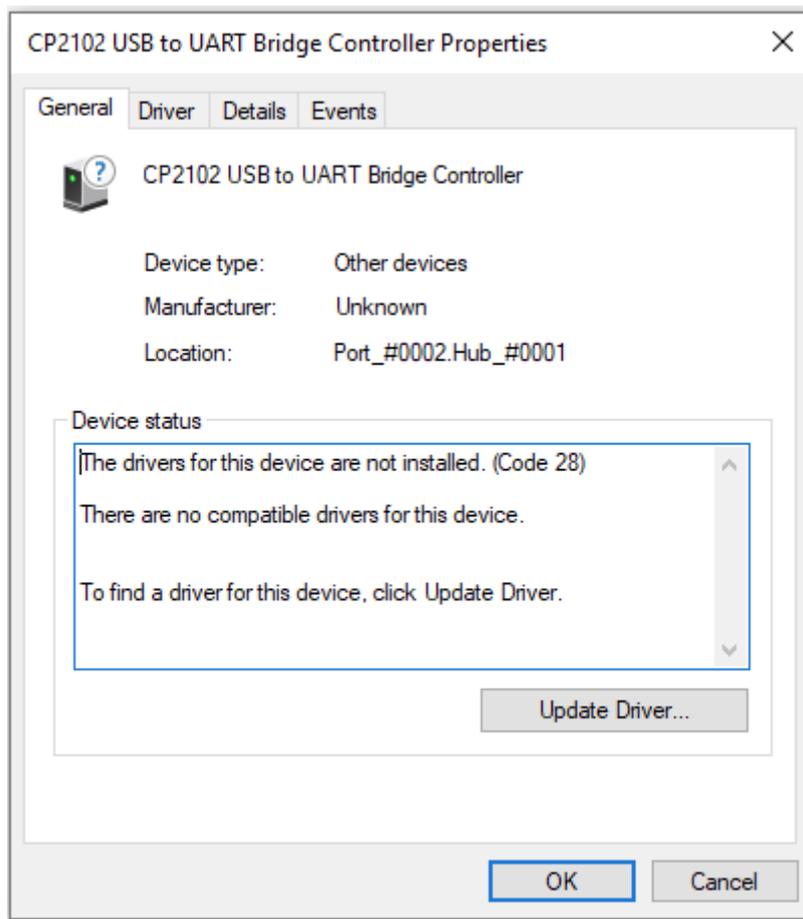
5.- Instalación del controlador

Instalamos el controlador de la placa de control Keystudio PLUS. El chip USB-TTL en la placa PLUS adopta el chip serie CP2102. El programa controlador de este chip está incluido en la versión Arduino 1.8 y superior, lo cual es conveniente. Conecte el puerto USB de la placa, ca computadora puede reconocer el hardware e instalar automáticamente el controlador de CP2102.

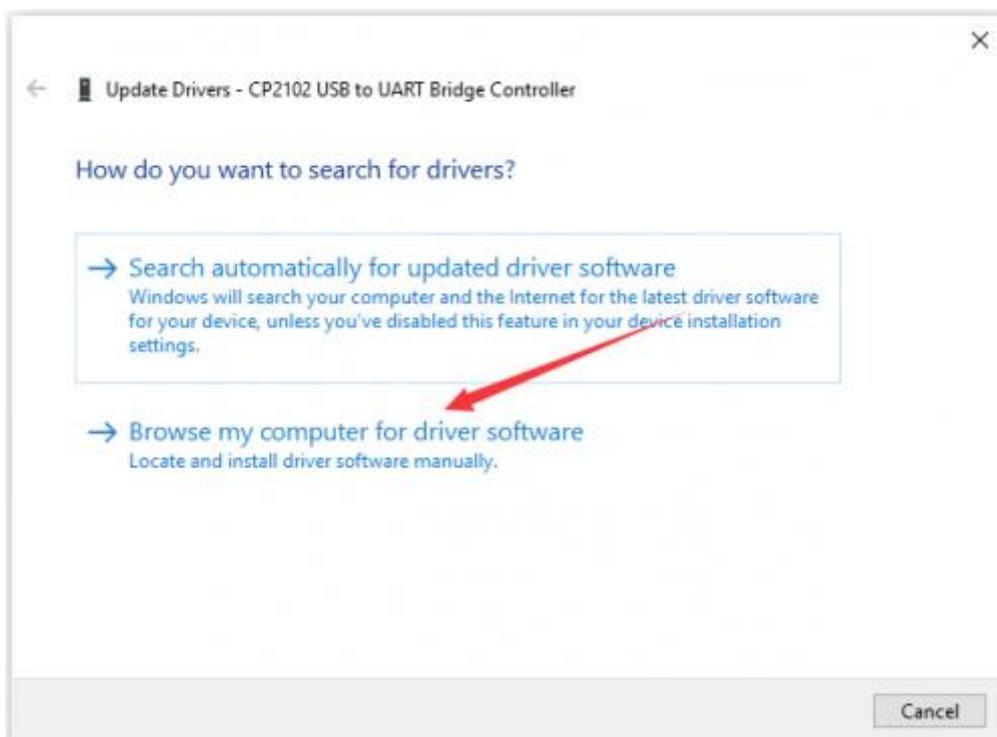
Si la instalación no tuvo éxito, o tiene la intención de instalarla manualmente, abra el administrador de dispositivos de la computadora, haga clic derecho en Equipo – Propiedades – Administrador de dispositivos.



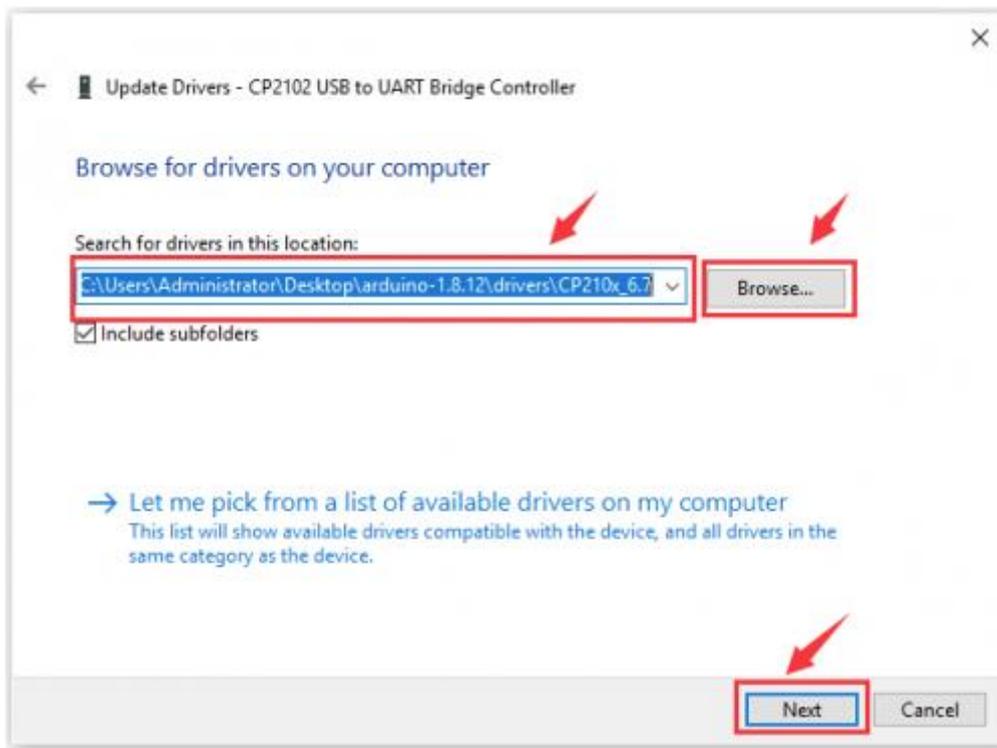
Hay un signo de exclamación amarillo en la página, lo que implica que la instalación del controlador de CP2102 no se realizó correctamente. Luego hacemos doble clic en hardware y actualizaremos el controlador.



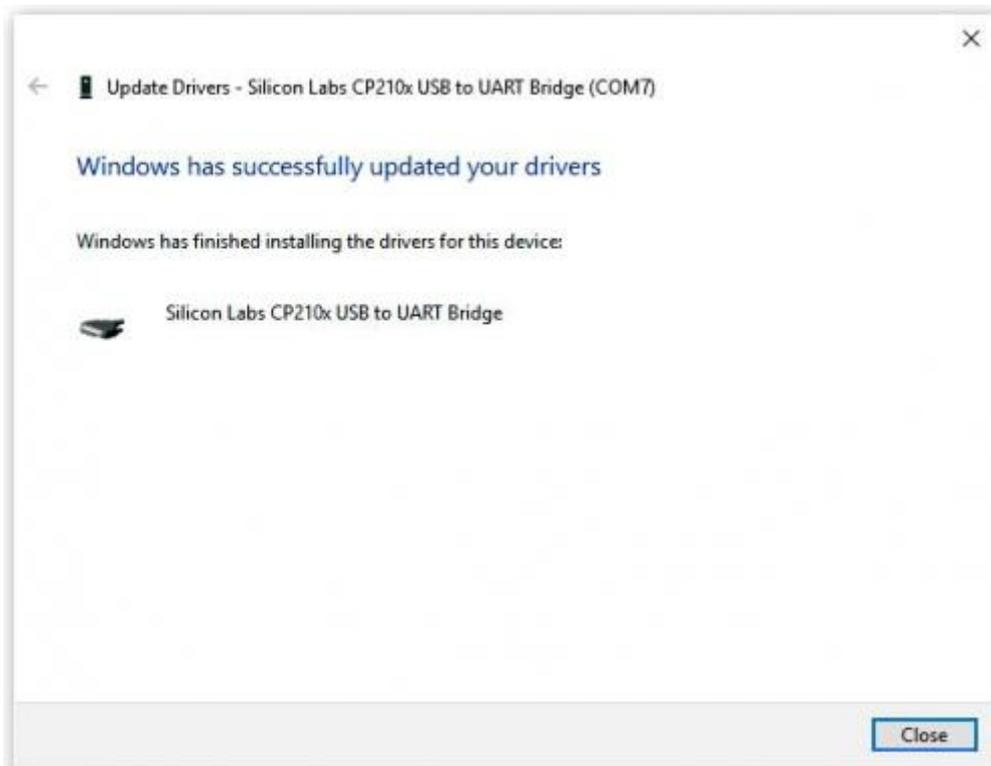
Haga clic en “Aceptar” para ingresar a la página siguiente, haga clic en “Buscar en mi computadora el software del controlador”, descubra el software ARDUINO instalado o descargado. Como se muestra abajo.

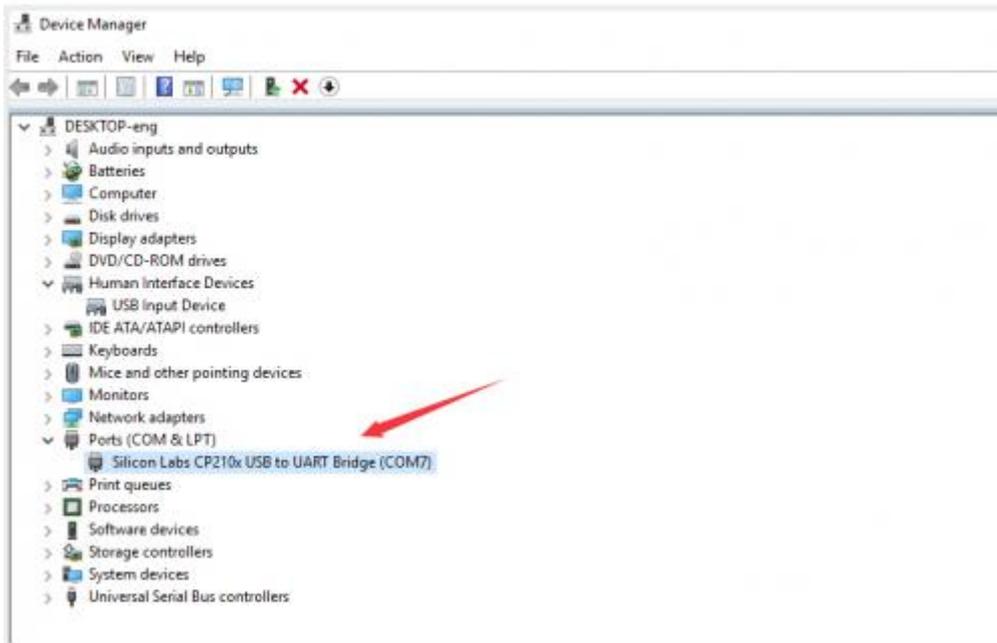


Hacemos clic en “Examinar”, luego buscamos la carpeta del controlador, o puede ingresar “controlador” para buscar en el cuadro rectangular, luego haga clic en “siguiente”, el controlador se instalará correctamente. (Colocó la carpeta de software Arduino en el escritorio, puedes seguir el camino.

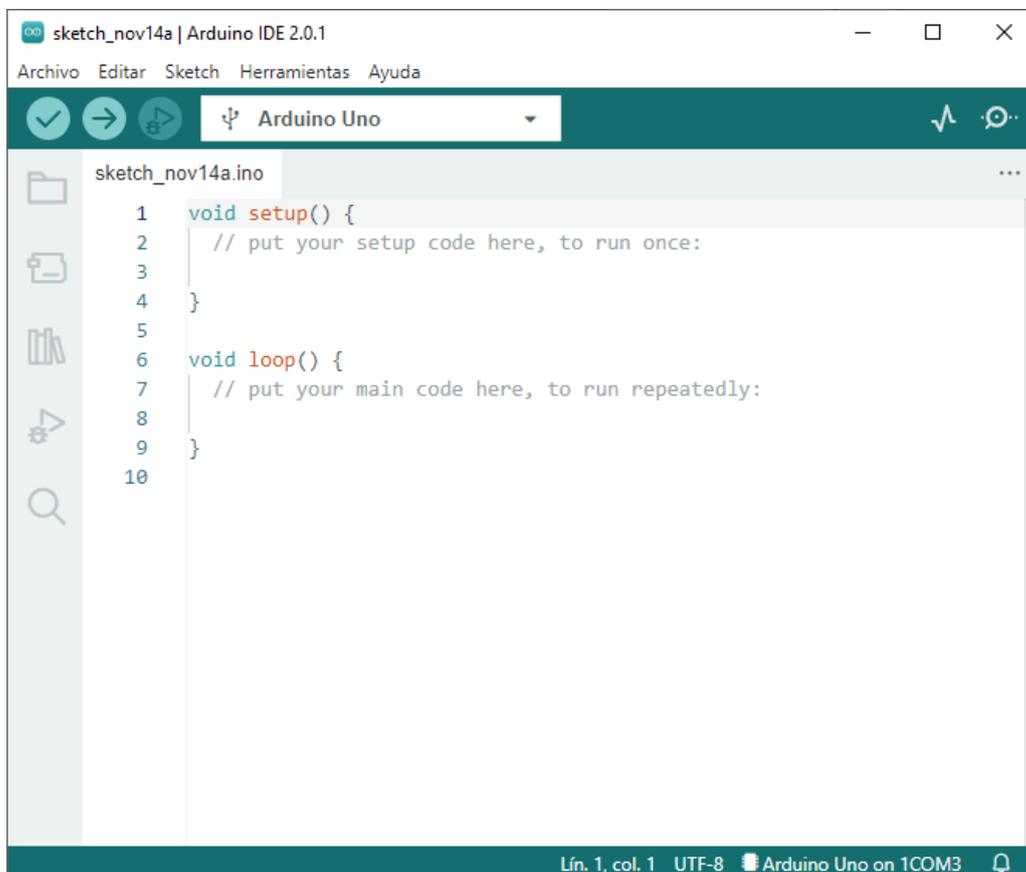


Abra el administrador de dispositivos, encontrará que desaparece el signo de exclamación amarillo. El controlador de CP2102 se instaló correctamente.

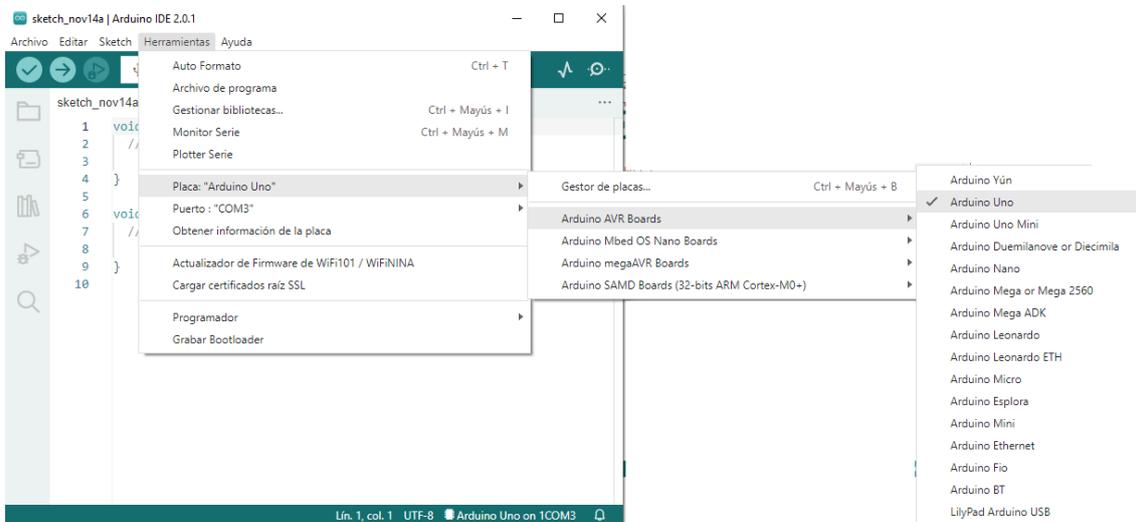




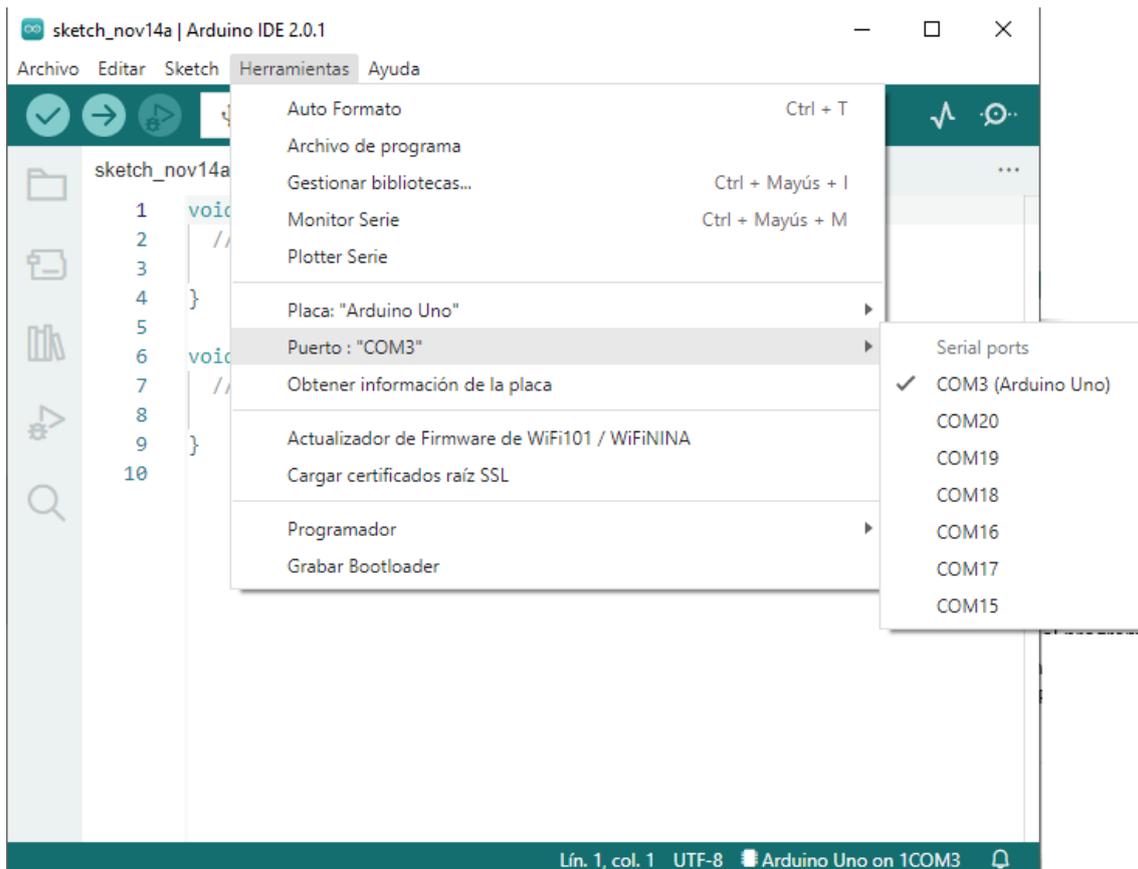
6.- Configuración del IDE de Arduino
Haga clic en el icono, abra Arduino IDE.

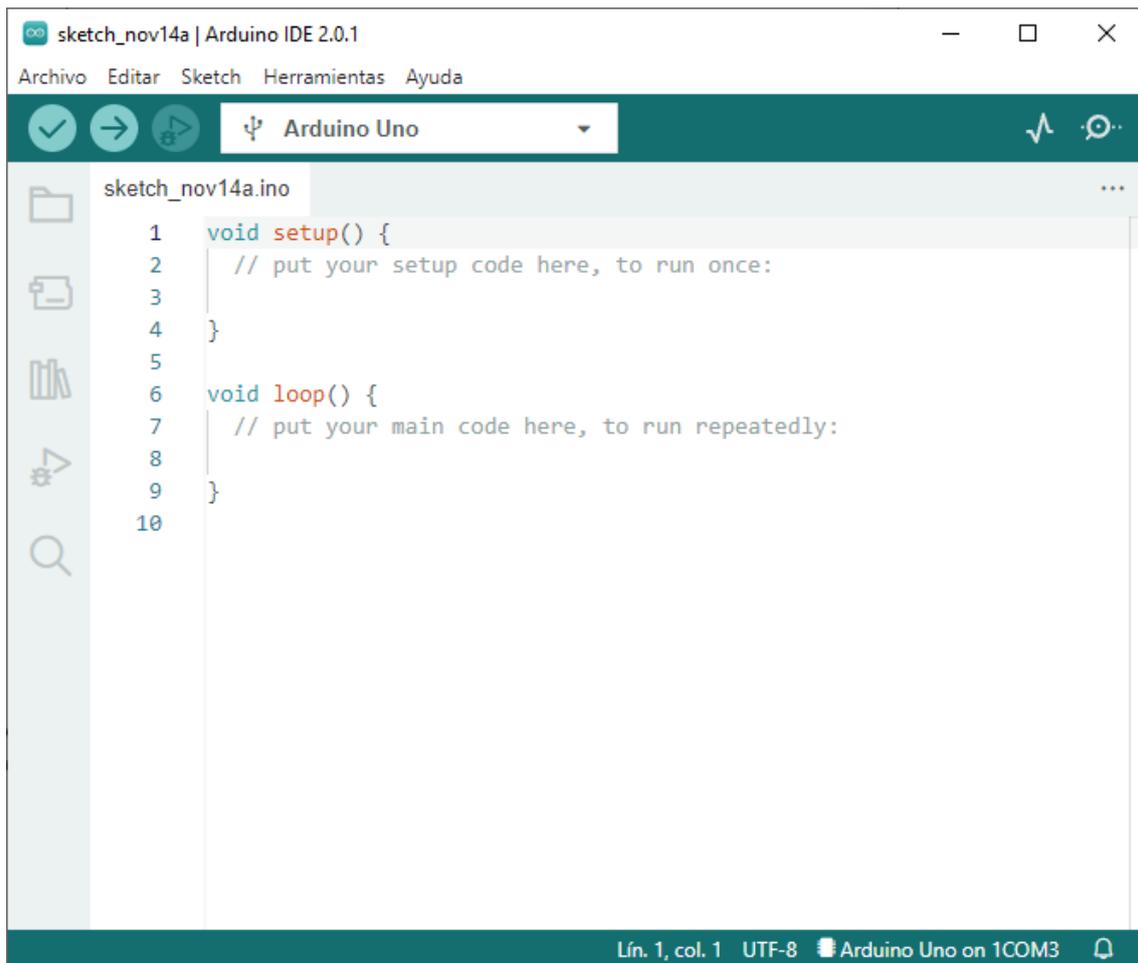


Para evitar errores al cargar el programa en la placa, debe seleccionar la placa Arduino correcta que coincida con la palca conectada en la computadora. Luego regrese al software Arduino, debe hacer clic en Herramientas, Placa como se muestra en la figura de abajo.



Luego, seleccione el puerto COM correcto (puede ver el puerto COM correspondiente después de que el controlador se haya instalado correctamente).





Antes de cargar el programa en la placa, demostraremos la función de cada símbolo en la barra de herramientas del IDE de Arduino.



Se utiliza para verificar si hay errores de compilación o no.



Se utiliza para cargar el programa en su placa Arduino.



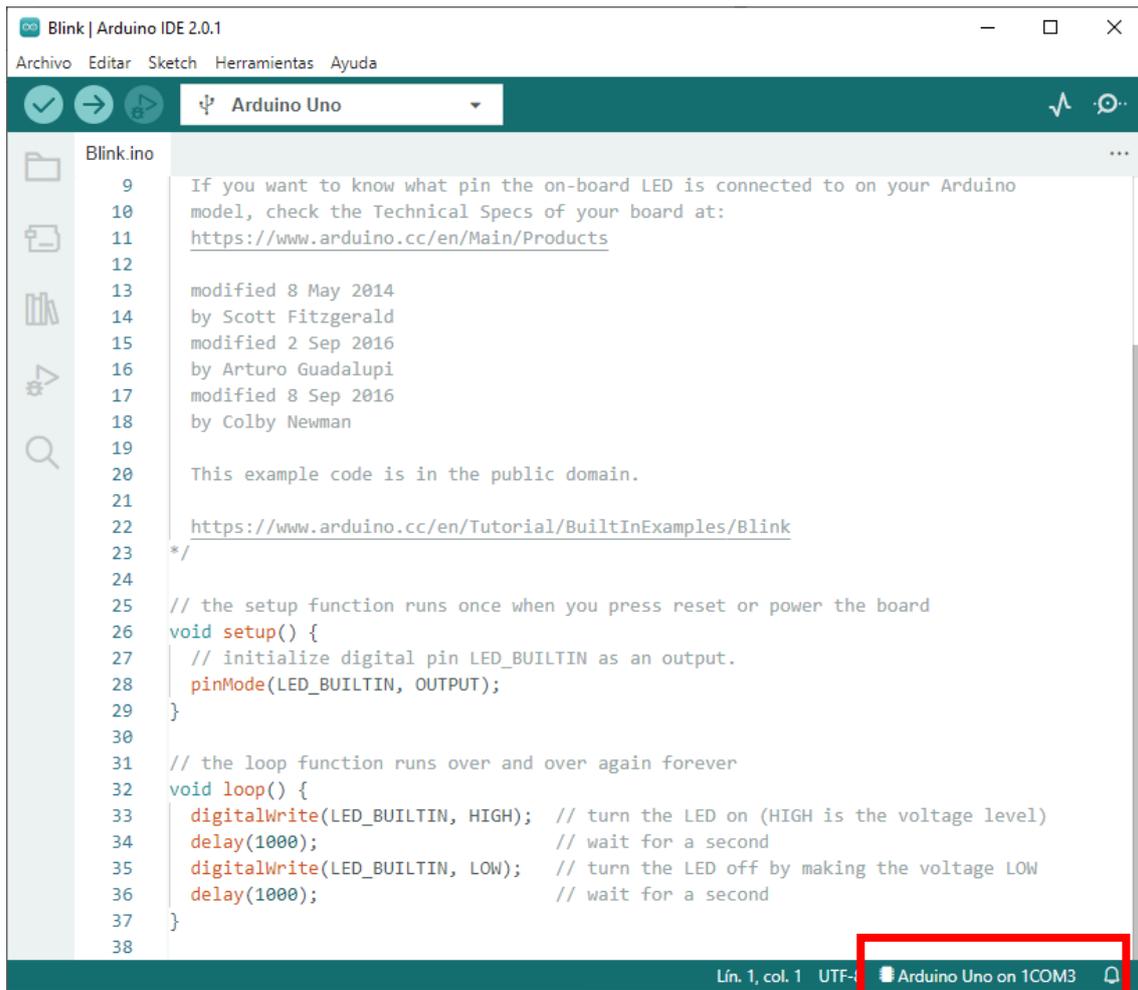
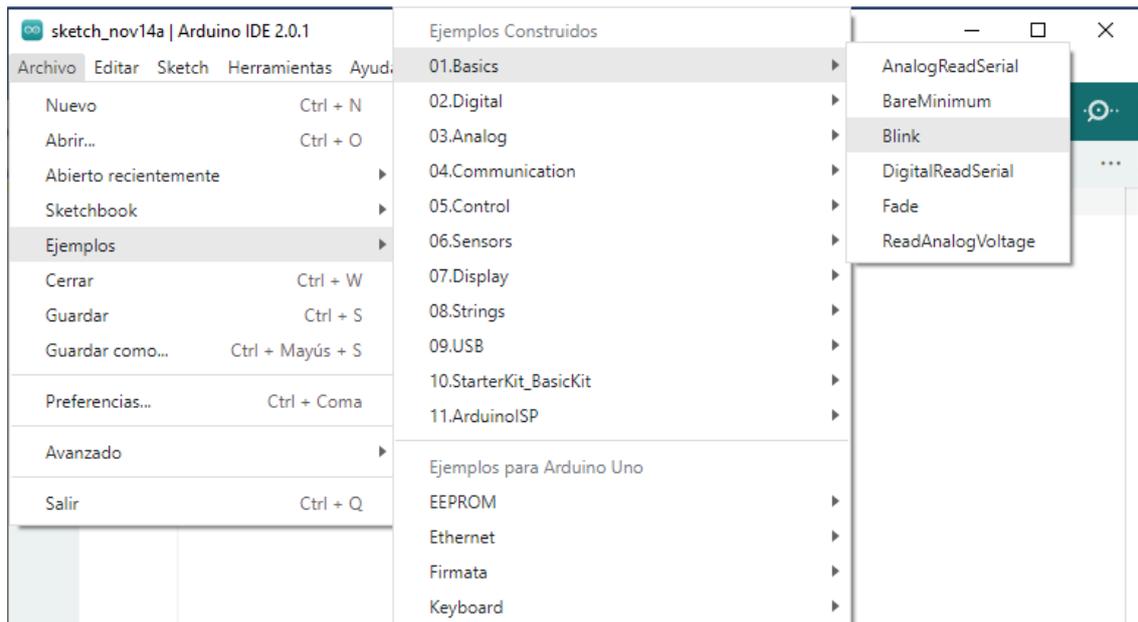
Muestra el Plotter Serial



Se utiliza para enviar y recibir datos desde la placa Arduino al ordenador.

7.- Comienza tu primer programa.

Abra el archivo para seleccionar Ejemplo, elija BLINK de BÁSICO, como se muestra a continuación:



Configure la placa y el puerto COM, la placa y el puerto COM correspondientes se muestran en la parte inferior derecha de IDE.

Haga clic para comenzar a compilar el programa, verifique los errores:



Haga clic para cargar el programa, cargar con éxito.



Cargue el programa con éxito, las luces LED integradas se encienden durante 1 segundo y se apagan durante 1 segundo. Enhorabuena, has terminado el primer programa.

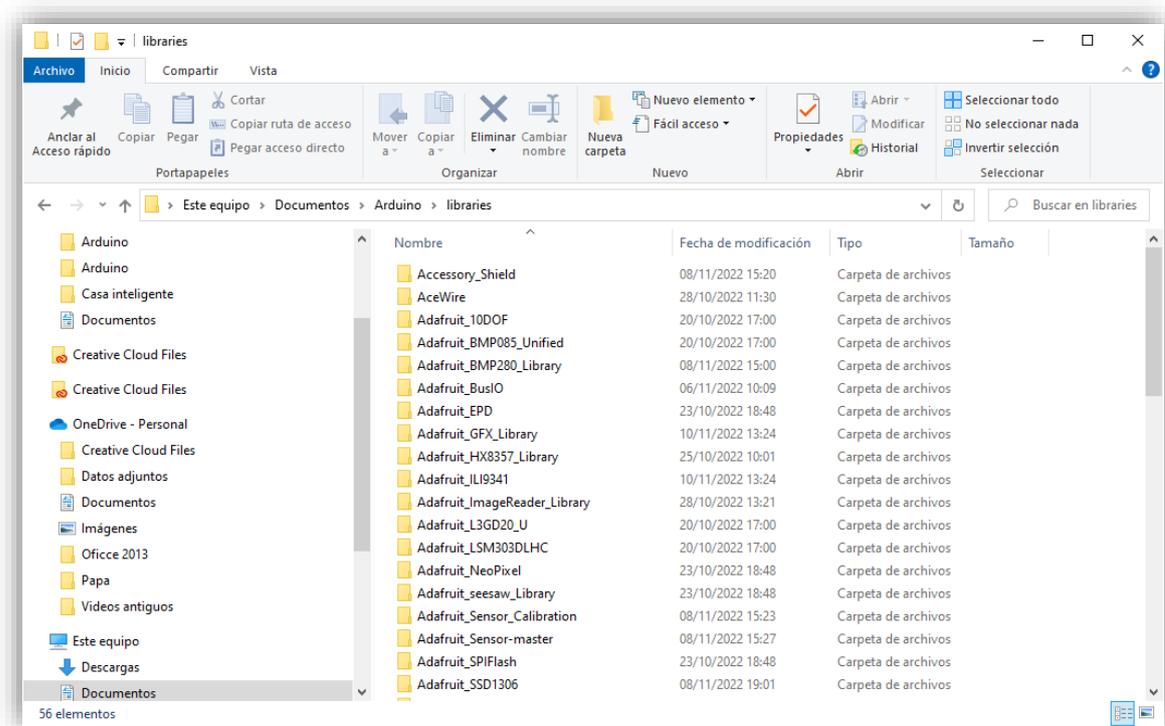
8.- ¿Cómo agregar una biblioteca?

¿Qué son las bibliotecas? Las bibliotecas son una colección de código que facilita la conexión a un sensor, una pantalla, un módulo, etc. Por ejemplo, la biblioteca LiquidCrystal integrada ayuda a comunicarse con las pantallas LCD. Hay cientos de bibliotecas adicionales disponibles en Internet para descargar. Las bibliotecas integradas y algunas de estas bibliotecas adicionales se enumeran en la referencia.

Aquí le presentamos la forma más sencilla de agregar bibliotecas.

En la siguiente ruta se encuentran las librerías:

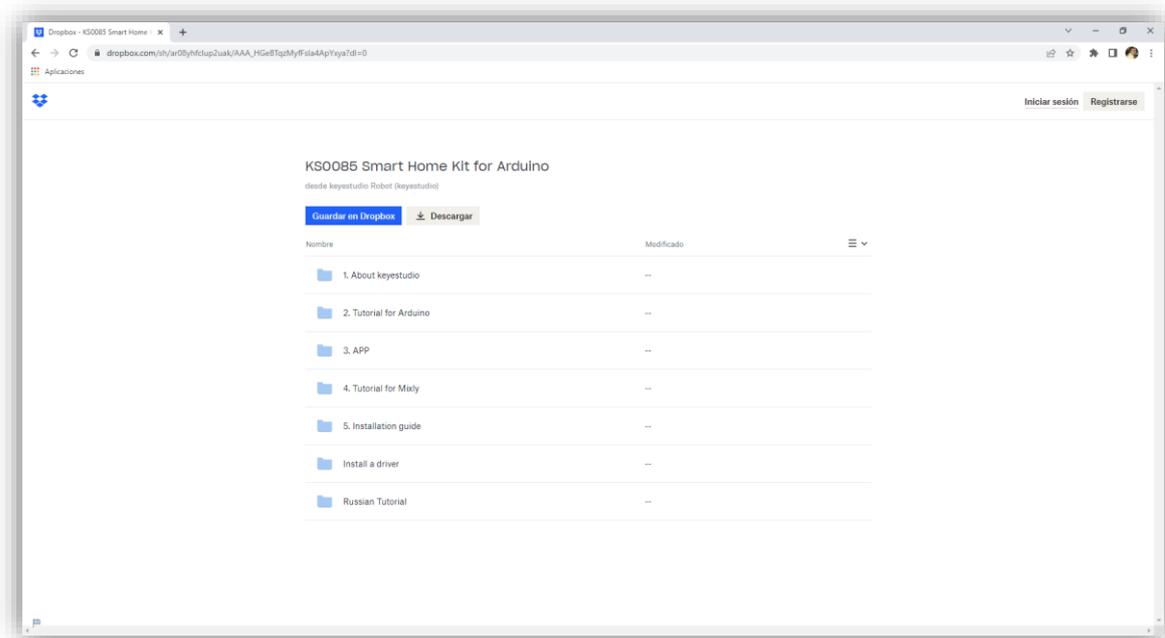
C:\Users\pmver\OneDrive\Documentos\Arduino\libraries



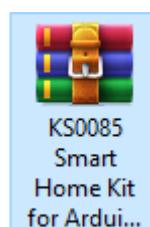
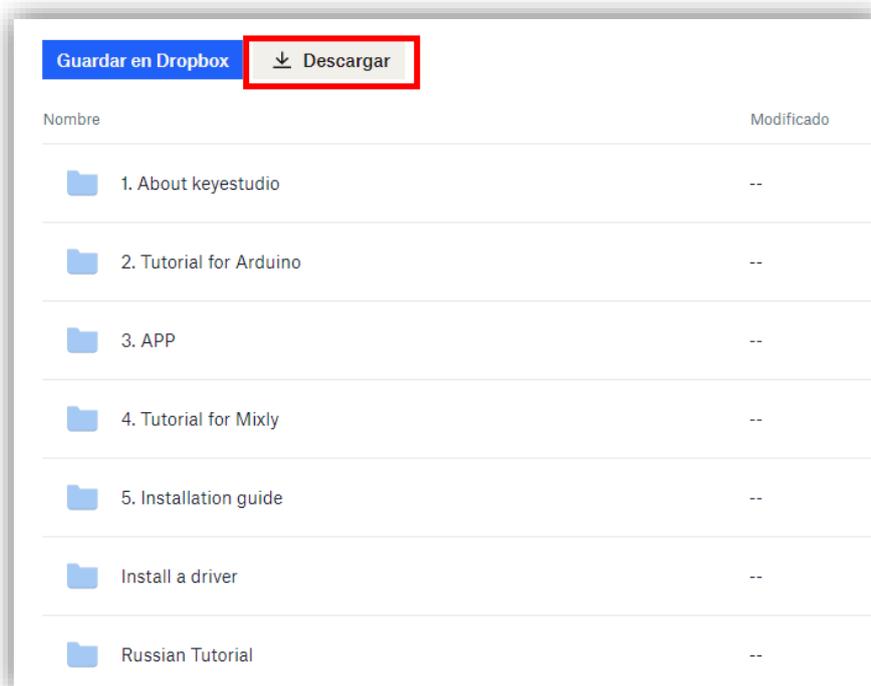
A continuación, busque las “Bibliotecas” del hogar inteligente (visto en el enlace:

<https://fs.keyestudio.com/KS0085>

Como se muestra a continuación.

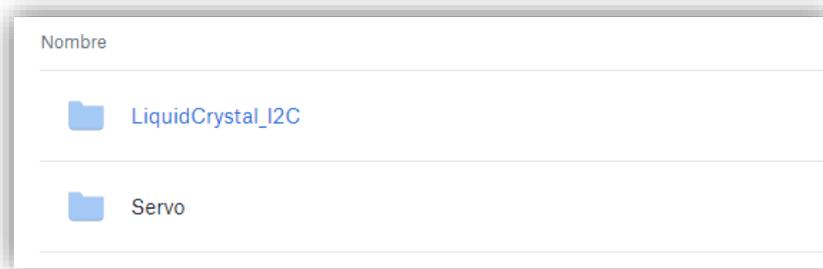


Seleccionaremos el botón Descargar.



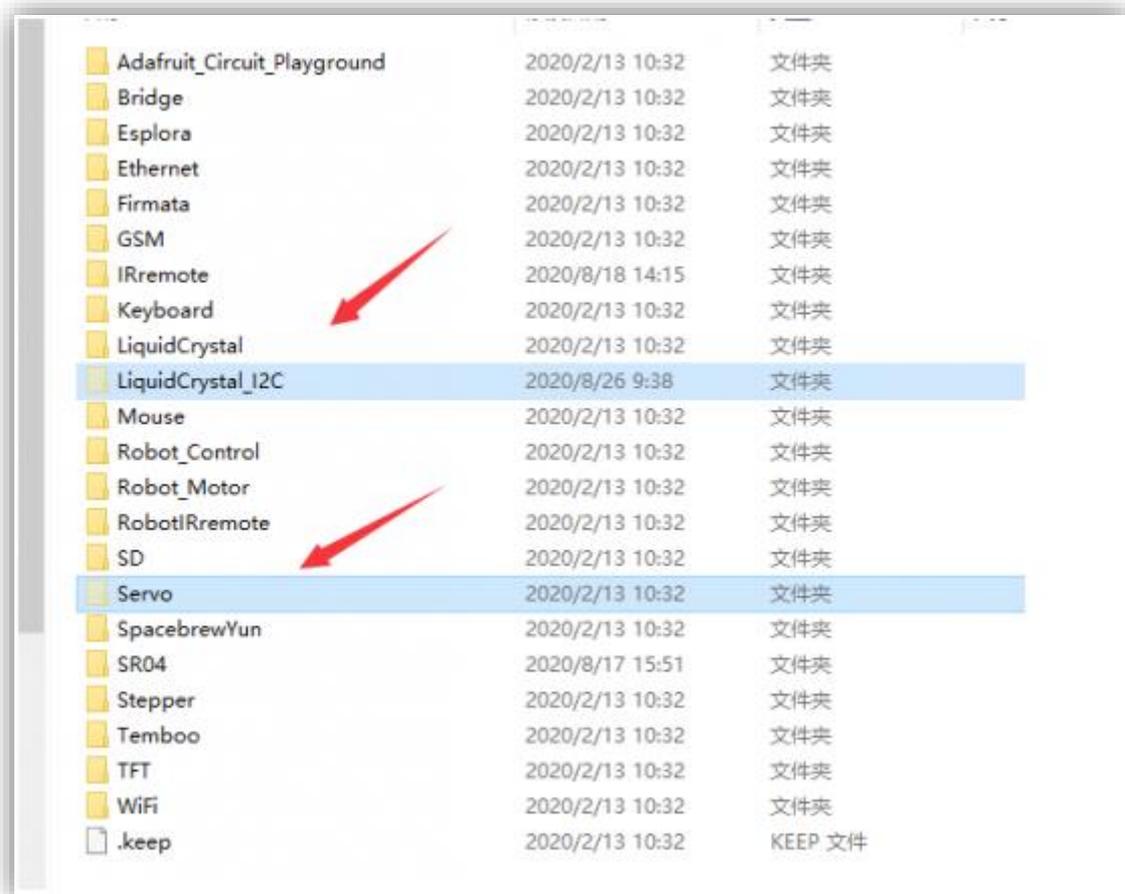
Seleccionamos la carpeta 2. Tutorial for Arduino.

Seleccionamos la carpeta 5. Arduino Library files.



Solo tienes que replicar y pegar en la carpeta de bibliotecas de Arduino IDE.

La biblioteca de home Smart se instaló con éxito, como se muestra a continuación.



9.- Proyectos



Muy bien, vayamos directamente a nuestros proyectos. En este kit, hay 14 sensores incluidos. Comenzaremos con el sensor simple para que conozcas profundamente el hogar inteligente. Sin embargo, si eres un entusiasta con conocimientos de Arduino. Puedes omitir estos pasos, ensamblar el kit de hogar inteligente directamente (hay un vídeo de ensamblaje en la carpeta).

Nota: en este curso, la interfaz de cada sensor/módulo marcado con (G, -. GND) indica el polo negativo, G está conectado a G o – o GND del escudo del sensor o placa de control “V” implica un polo positivo que está vinculado con V o VCC o 5V.

Proyecto 1: Parpadeo LED

Descripción:



En la lección anterior, instalamos el controlador de la placa de desarrollo keystuido V4.0, comenzamos con proyectos simples. Para esta lección, realizaremos “Arduino parpadea LED”, que es la práctica básica para principiantes. Proporcionamos el código de prueba para controlar el LED para que realice el parpadeo. En el código, puede configurar una escena intermitente distinta cambiando el tiempo de encendido y apagado. Conecte GND y VCC, el LED se encenderá cuando el extremo de la señal S tenga un nivel alto, por lo contrario, el LED se apagará cuando el extremo de la señal S tenga un nivel bajo.

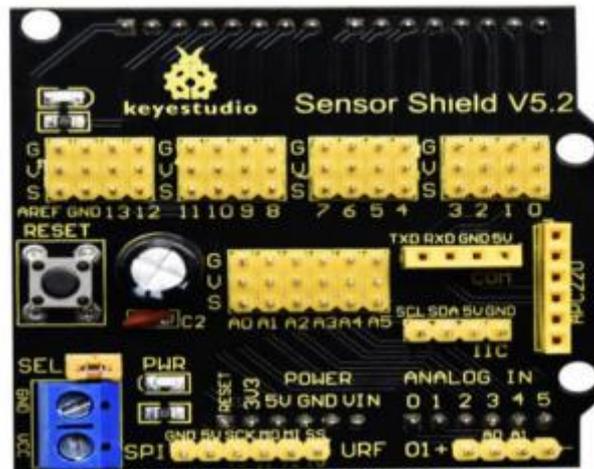
Especificaciones:

- Interfaz de control: puerto digital.
- Voltaje de funcionamiento: CC 3,3 – 5V.
- Paso de clavija: 2,54 mm.
- Color de la pantalla LED: blanco.
- Tamaño: 30 * 20 mm.
- Peso: 3 g.

Equipo:

PLUS control board*1	Sensor shield*1	White LED module *1	USB cable*1	3pin F-F Dupont line*1
				

Escudo sensor.



Por lo general, combinamos la placa de control Arduino con otros sensores, módulos y sensores múltiples, lo que es difícil de cablear. Por el contrario, este protector de sensor cubre este problema, solo necesita apilarlo en la placa de control keyestudio PLUS cuando lo use.

Este escudo se puede insertar directamente en los sensores 3PIN, también rompe los puertos de comunicación de uso común, como la comunicación en serie, la comunicación IIC, la comunicación SPI. Además, el escudo viene con un botón de reinicio y 2 luces de señal.

Descripción de pines

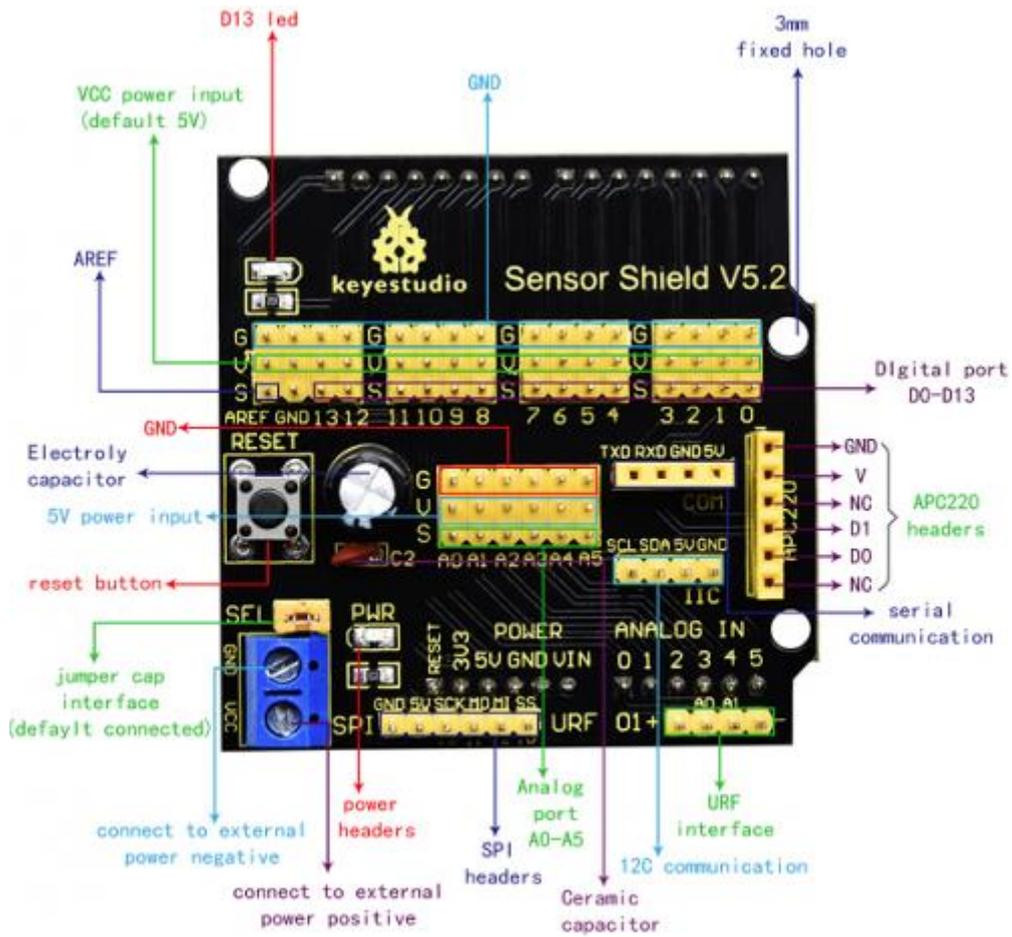
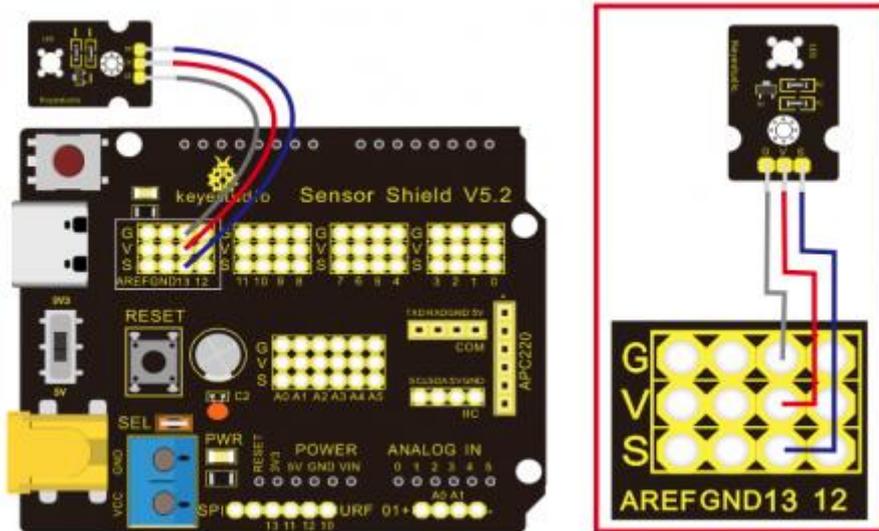


Diagrama de conexión:

Junto al cable, conecte el módulo LED con D13 de blindaje. Nota G, V y S del módulo LED blanco están vinculados con G, V y 13 de la placa de expansión.



Código de prueba:

```

/*
Keyestudio smart home kit for Arduino
Proyecto 1
Parpadear
htt://www.keyestudio.com
*/

void setup() {
  // Inicializa el pin digital 13 como salida.
  pinMode(13, OUTPUT);
}
// La función loop se ejecuta una o otra vez para siempre.
void loop() {
  digitalWrite(13, HIGH); // Encienda el LED (ALTO es el nivel de voltaje)
  delay(1000);           // Espera un segundo
  digitalWrite(13, LOW); // Apague el LED haciendo que el voltase sea BAJO.
  delay(1000);           // Espera un segundo
}

```

Resultado de la prueba:

Cargue el código de prueba con éxito, el LED blanco comienza a parpadear, se encienden durante 1000 ms, se apagar durante 1000 ms, alternativamente.

Código Explicación

El código parece largo y desordenado, pero la mayoría son comentarios. La gramática de Arduino se basa en C. Los comentarios generalmente tienen dos formas de expresión.

/* */: Adecuado para comentarios de párrafo largo.

//: Adecuado para comentarios de una sola línea.

Entonces, el código contiene mucha información vital, como el autor, el acuerdo emitido, etc.

La mayoría de las personas omiten los comentarios, el principiante debe desarrollar un buen hábito de revisar el código. En primer lugar, revisa los comentarios. Contiene la información proporcionada y lo ayudan a comprender el código de prueba rápidamente. En segundo lugar, forma el hábito de escribir comentarios.

```

// Las funciones de configuración se ejecutan cuando presiona
// reiniciar o enciende la placa
void setup() {
  // Inicializa el pin digital 13 como salida.
  pinMode(13, OUTPUT);
}

```

Según los comentarios, encontraremos que el autor define el modo pin D13 como salida en función setup(). Setup() es la función básica de Arduino. Se ejecutará una vez en la ejecución del programa, generalmente como pin de definición, define y asegura las variables.

```

// La función loop se ejecuta una o otra vez para siempre.
void loop() {
  digitalWrite(13, HIGH); // Encienda el LED (ALTO es el nivel de voltaje)
  delay(1000);           // Espera un segundo
  digitalWrite(13, LOW); // Apague el LED haciendo que el voltaje sea BAJO.
  delay(1000);           // Espera un segundo
}

```

Loop() es la función necesaria de Arduino, puede ejecutarse y repetirse todo el tiempo después de que "setup()" se ejecute una vez.

En la función loop(), el autor usa

`digitalWrite(13, HIGH);` //enciende el LED (ALTO es el nivel de voltaje).

`digitalWrite(13, HIGH);`: establece el voltaje de salida del pin en un nivel alto o bajo. Hacemos que la salida D13 sea de alto nivel, luego las luces LED se encienden.

`delay(1000);` // espera un segundo.

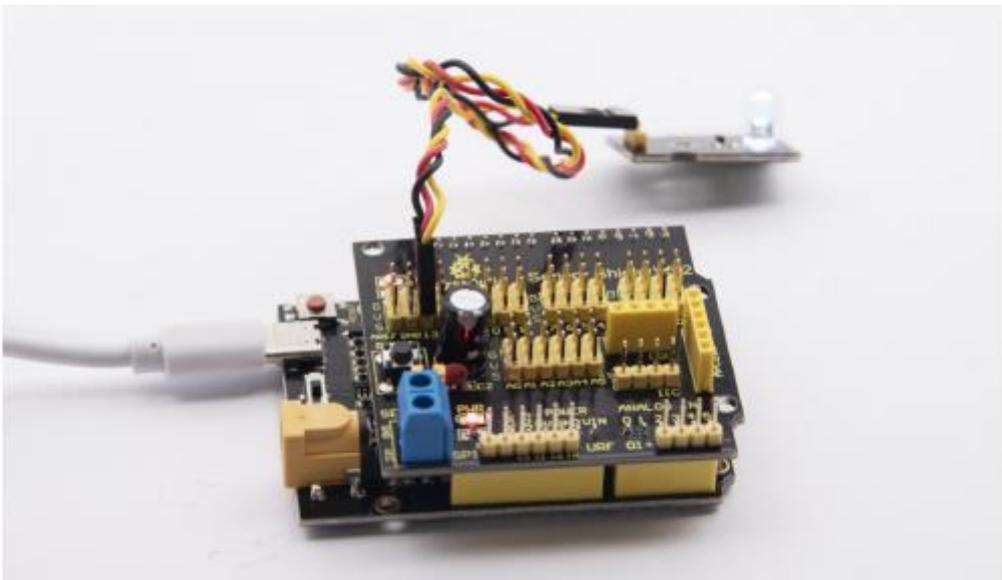
La función delay se usa para retrasar el tiempo, 1000 ms. Es 1 segundo, la unidad es en milisegundos.

`digitalWrite(13, LOW);` // Apaga el LED haciendo que el voltaje sea BAJO.

De manera similar, hacemos que la salida D13 sea de bajo nivel, el LED se apagará.

`delay(1000);` // Espera un segundo.

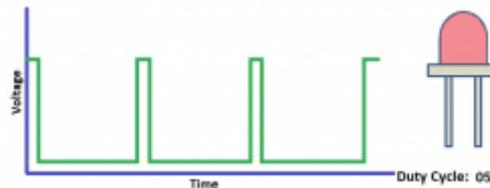
Retraso de 1 segundo, luz LED encendida, mantener 1 segundo, luz LED apagada, permanecer encendida un 1 segundo. El LED parpadea con un intervalo de 1 segundo. ¿Qué pasa si quieres hacer que el LED parpadee rápidamente? Solo necesitas modificar el valor del bloque de delay. Reproducir el valor de retardo implica que el tiempo de espera sea menor, es decir, que parpadee rápidamente. Por el contrario, puede hacer que el LED parpadee lentamente.



Proyecto 2: Respirando Luz



Descripción



En la lección anterior, controlamos el encendido y apagado del LED y lo hicimos parpadear. En este proyecto, controlaremos el brillo del LED a través de PWM para simular el efecto de respiración. De manera similar, puede cambiar la longitud del paso y el tiempo de retraso en el código para demostrar un efecto de respiración diferente.

PWM es un medio para controlar la salida analógica a través de medios digitales. El control digital se utiliza para generar ondas cuadradas con diferentes ciclos de trabajo (una señal que cambia constantemente entre niveles altos y bajos) para controlar la salida analógica. En general, el voltaje de entrada del puerto es de 0 V y 5 V. ¿Qué pasa si se requiere 3 V? ¿O qué pasa si cambia entre 1 V, 3 V y 5 V? No podemos cambiar la resistencia constantemente. Para esta situación, necesitamos controlar por PWM.

Para la salida de voltaje del puerto digital Arduino, solo hay LOW y HIGH, que corresponden a la salida del voltaje de 0 V y 5 V. Puede definir LOW como 0 y HIGH como 1, y dejar que Arduino emita quinientas señales 0 o 1 en 1 segundo. Si sale quinientos 1, eso es 5V, si todo es 0, eso es 0 V. Si emite 01010101010101 de esta manera, el puerto de salida es de 2,5 V, que es como mostrar una película. La película que vemos no es completamente continua. En realidad, emite 25 imágenes por segundo. En esta caso, el humano no puede decirlo, tampoco PWM. Si desea un voltaje diferente, debe controlar la relación 0 y 1. Cuantas más señales de salida 0, 1 por unidad de tiempo, habrá una mayor precisión.

Equipo:

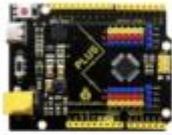
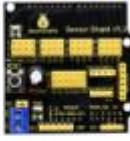
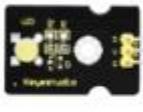
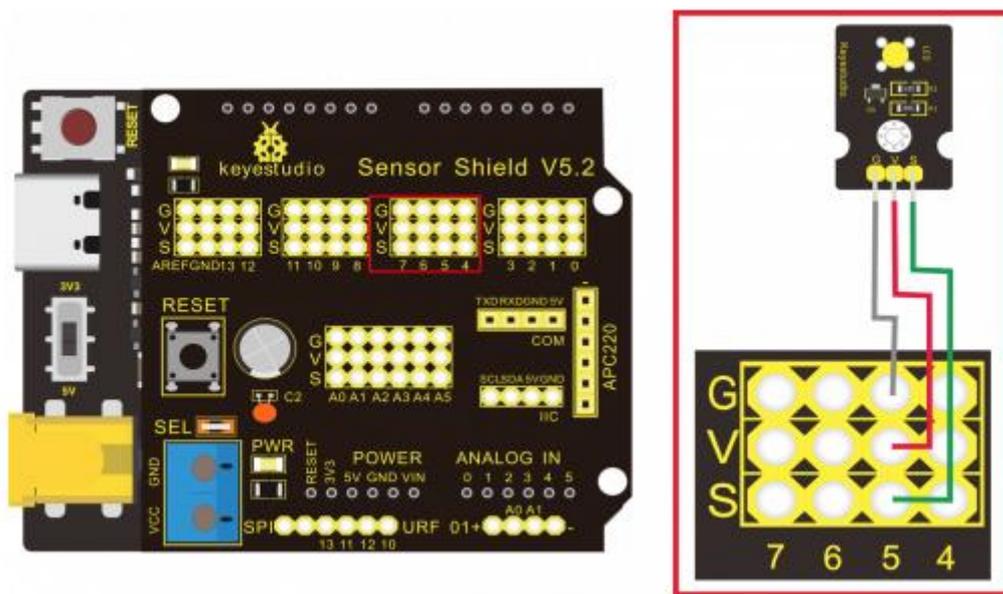
PLUS control board*1	Sensor shield*1	Yellow LED module*1	USB cable*1	3pin F-F Dupont line*1
				

Diagrama de conexión:



Nota: en el escudo del sensor, los pines G, V Y S del módulo LED amarillo están vinculados con G, V y 5.

Código de prueba:

```

/*
Keyestudio smart home Lit for Arduino
Proyecto 2
PWM
http://www.keyestudio.com
*/
int ledPin = 5; //Define el LED con el pin D5
void setup() {
  pinMode(ledPin, OUTPUT); //Inicializa ledPin como de salida
}

```

```

void loop() {
  for (int value = 0; value < 255; value = value + 1){
    analogWrite(ledPin, value); // El LED se enciende gradualmente
    delay(5); // retardo de 5 milisegundos
  }
  for(int value = 255; value > 0; value = value - 1 ){
    analogWrite(ledPin, value); // El Led se apaga gradualmente
    delay(5); // retardo de 5 milisegundos
  }
}

```

Resultado de la prueba:

Cargue el código de prueba con éxito. El LED gradualmente se vuelve más brillante y luego más oscuro, como el aliento humano, en lugar de encenderse y apagarse inmediatamente.



Análisis del código:

Quando necesitamos repetir algunas declaraciones, tenemos que usar la declaración “for” para el formato de declaración de la siguiente manera:

```

for (ciclo inicial; ciclo condición; declaración de ajuste de ciclo){
    Establece el cuerpo del bucle
}

```

Secuencia cíclica “for”.

```

for (int value = 0; value < 255; value=value+1){
    ...
}
for (int value = 255; value >0; value=value-1){
    ...
}

```

Las declaraciones “for” hacen que el valor aumente de 0 a 255, luego se reduzca de 255 a 0, luego aumente a 255, ...bucle infinito.

Hay una nueva función en la declaración “for” ----- analogWrite().

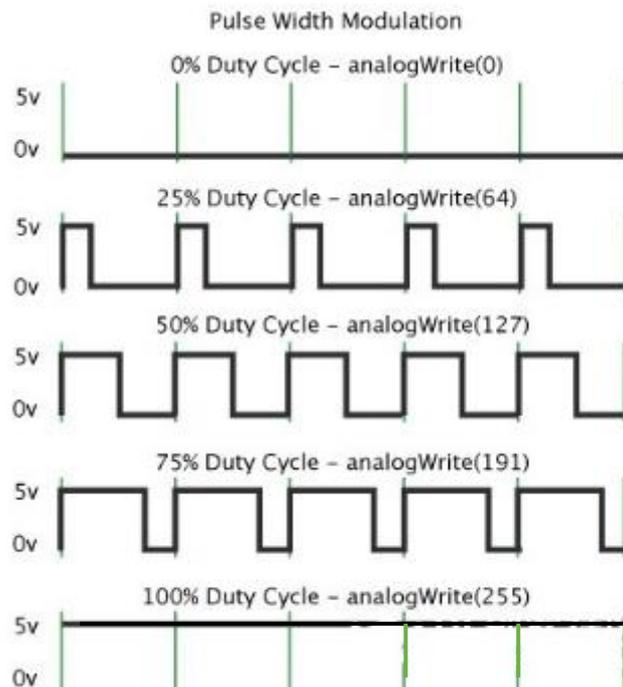
Sabemos que el puerto digital solo tiene dos estados de 0 y 1. Entonces, ¿Cómo enviar un valor analógico a un valor digital? Aquí, necesitamos esta función, observe la placa Arduino y encontrará 6 pines con “~”. Son diferentes de otros pines y pueden emitir señales PWM.

El formato de la función es la siguiente:

AnalogWrite(pin, valor) analogWrite() se usa para escribir un valor analógico de 0~255 para el puerto PWM, por lo que el valor está en el rango de 0~255, tenga en cuenta que solo escribe los pines digitales con Función PWM, como pin 3, 5, 6, 9, 10, 11.

PWM es una tecnología para obtener cantidades analógicas a través de métodos digitales. El control digital forma una onda cuadrada, y la señal de onda cuadrada solo tiene dos estados de conmutación (es decir, niveles alto o bajo de nuestros pines digitales). Al controlar la relación de la duración de encendido y apagado, se puede simular un voltaje de varía de 0 a 5 V. El tiempo necesario (referido académicamente como alto nivel) se denomina ancho de pulso, por lo que PWM también se denomina modulación de ancho de pulso.

A través de las siguientes cinco ondas cuadradas, sepamos más sobre PWM.

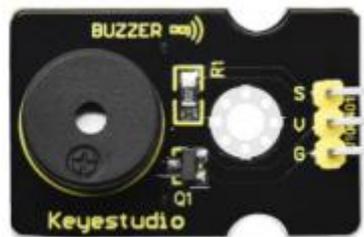


En la figura anterior, la línea verde representa un período y el valor de analogWrite() corresponde a un porcentaje que también se denomina ciclo de trabajo. El ciclo de trabajo implica que la duración de alto nivel se divide por la duración de bajo nivel en un ciclo. De arriba abajo, el ciclo de trabajo de la primera onda cuadrada es 0% y su valor correspondiente es 0. El brillo del LED es el más bajo, es decir, se apaga. Cuanto más tiempo dure el nivel alto, más brillante será el LED. Por lo tanto, el último ciclo de trabajo es 100%, que corresponde a 255, el LED es más brillante. 25% significa más oscuro.

PMW se usa principalmente para ajustar el brillo del LED o la velocidad de rotación del motor.

Proyecto 3: Zumbador pasivo

Descripción:



Hay prolíficos trabajos interactivos realizados por Arduino. El más común es el sensor de sonido y luz. Siempre usamos LED para hacer experimentos. Para esta lección, diseñamos un circuito para emitir sonido. Los componentes de sonido universales son el zumbador y las bocinas. El zumbador es más fácil de usar. Y el zumbador incluye un zumbador activo o pasivo. En este experimento, adoptamos un zumbador pasivo. Mientras usamos el zumbador pasivo, podemos controlar diferentes sonidos ingresando notas cuadradas con distintas frecuencias. Durante el experimento, controlamos el código para hacer sonar el zumbador, comenzar con un sonido "tick, tic", luego hacer que el zumbador pasivo emita "do re mi fa so la si do" y reproducir canciones específicas.

Equipo:

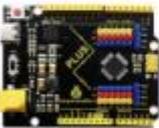
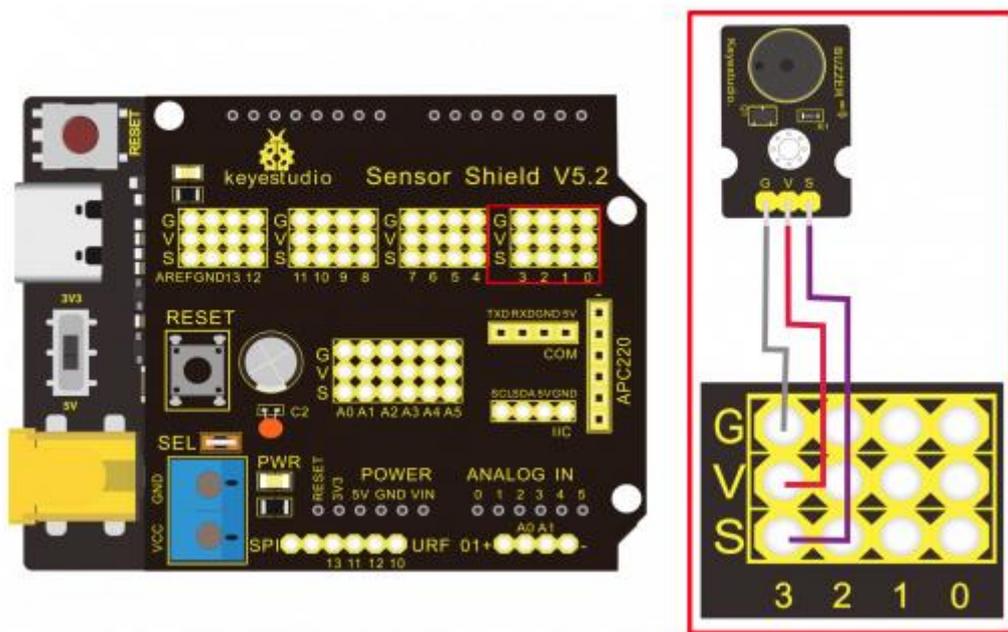
PLUS control board*1	Sensor shield*1	Passive buzzer*1	USB cable*1	3pin F-F Dupont line*1
				

Diagrama de conexión:



Nota: Los puertos G, V y S del módulo del zumbador pasivo están conectados por separado a G, V y 3, en el escudo, enciéndalo.

Código de prueba:

```

/*
Keystudio smart home kit for Arduino
Proyecto 3.1
Buzzer
http://www.keystudio.com
*/
int tonepin = 3; // Establezca el pin del zumbador en el digital D3
void setup() {
  pinMode(tonepin, OUTPUT); // Establezca el modo de pin de E/S digital en salida
}

void loop() {
  int i, j;
  while (1)
  {
    for(i=0; i<80; i++) // Emite un sonido de frecuencia
    {
      digitalWrite(tonepin, HIGH); // Sonido
      delay(1); // Retado 1 ms.
      digitalWrite(tonepin, LOW); // No sonido
      delay(1);
    }
  }
}

```

```

for(j = 0; j < 100; j++) // Salida de sonido con otra frecuencia
{
    digitalWrite(tonepin, HIGH); // Sonido
    delay(2); // Retado 1 ms.
    digitalWrite(tonepin, LOW); // No sonido
    delay(2);
}
}
}

```

Resultado de la prueba:

Del código anterior, 80 y 100 deciden la frecuencia de la declaración "for". Delay controla la duración, como el ritmo en la música.

Tocaremos música fabulosa si controlamos bien la frecuencia y los ritmos, así que averigüemos la frecuencia de los tonos. Como se muestra abajo.

Baja:

Tone Note	1#	2#	3#	4#	5#	6#	7#
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	175	196	221	248	278
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371

Alto:

Tone Note	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	350	393	441	495	556
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742

Triplicar:

Tone Note	1#	2#	3#	4#	5#	6#	7#
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	700	786	882	990	1112
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

Después de conocer la frecuencia del tono, luego controlar el tiempo de reproducción de la nota. La música se producirá cuando cada nota toque una cierta cantidad de tiempo. El ritmo de la nota se divide en un tiempo, medio tiempo, $\frac{1}{4}$ tiempo, $\frac{1}{8}$ tiempo, estipulamos que el tiempo para una nota es 1. Medio tiempo es 0,5, $\frac{1}{4}$ de tiempo es 0,25, $\frac{1}{8}$ es 0,125, por lo tanto, se reproduce la música. Tomaremos ejemplo de "Himno a la alegría".

Ode to joy

Beethoven

1=D $\frac{4}{4}$

3 3 4 5 | 5 4 3 2 | 1 1 2 3 | 3 . 2 2 - |
 Joy-ful, joy- ful, we a- dore thee, God of glo- ry, lord of love.

3 3 4 5 | 5 4 3 2 | 1 1 2 3 | 2 . 1 1 - |
 Heart un-frod like flowers be-fore thee, O-pening to the sun a -bove.

2 2 3 1 | 2 3 4 3 1 | 2 3 4 3 2 | 1 2 5 3 |
 Melt the clouds of sin and sad-ness. rieve the dark of doubts a- way. Giv-

3 3 4 5 | 5 4 3 4 2 | 1 1 2 3 | 2 . 1 1 - :|
 -ver of im - mor-tal glad-ness. full us with the light of day.

Según la notación, la música tiene un compás de 4/4,

Hay notas especiales que debemos explicar.

1. La nota normal, como la primera nota 3, corresponde a 350 (frecuencia), ocupa 1 tiempo.
2. La nota subrayada significa 0,5 tiempos.
3. La nota con punto (3.) significa que se suma 0,5 tiempos, es decir 1+0.5 tiempos.
4. La nota con “—” representa que se suma 1 tiempo, es decir 1+1 tiempos.
5. Las dos notas sucesivas con arco implica legato, puede modificar ligeramente la frecuencia de la nota detrás de legato (necesita depurarlo usted mismo), como reducir o aumentar algunos valores, el sonido será más suave.

```
/*
Keystudio smart home kit for Arduino
Proyecto 3.2
Buzzer music
http://www.keystudio.com
*/
```

```
#define NTD0 -1
```

```

#define NTD1 294
#define NTD2 330
#define NTD3 350
#define NTD4 393
#define NTD5 441
#define NTD6 495
#define NTD7 556

#define NTDL1 147
#define NTDL2 165
#define NTDL3 175
#define NTDL4 196
#define NTDL5 221
#define NTDL6 248
#define NTDL7 278

#define NTDH1 589
#define NTDH2 661
#define NTDH3 700
#define NTDH4 786
#define NTDH5 882
#define NTDH6 990
#define NTDH7 112

// Listar todas las frecuencias sintonizadas en D

#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.25
#define SIXTEENTH 0.625

// Listar todos los latidos

int tune [] = // Enumere cada frecuencia de acuerdo con la notación
{
    NTD3, NTD3, NTD4, NTD5,
    NTD5, NTD4, NTD3, NTD2,
    NTD1, NTD1, NTD2, NTD3,
    NTD3, NTD2, NTD2,
    NTD3, NTD3, NTD4, NTD5,
    NTD5, NTD4, NTD3, NTD2,
    NTD1, NTD1, NTD2, NTD3,
    NTD2, NTD1, NTD1,
    NTD2, NTD2, NTD3, NTD1,
    NTD2, NTD3, NTD4, NTD3, NTD1,
    NTD2, NTD3, NTD4, NTD3, NTD2,
    NTD1, NTD2, NTD5, NTD0,
    NTD3, NTD3, NTD4, NTD5,

```

```

    NTD5, NTD4, NTD3, NTD4, NTD2,
    NTD1, NTD1, NTD2, NTD3,
    NTD2, NTD1, NTD1
};

float durt[] = // Enumero los tiempos de acorde con la notación
{
    1,1,1,1,
    1,1,1,1,
    1,1,1,1,
    1+0.5, 0.5, 1+1,
    1,1,1,1,
    1,1,1,1,
    1,1,1,1,
    1+0.5, 0.5, 1+1,
    1,1,1,1,
    1,0.5,0.5, 1,1,
    1,0.5,0.5, 1,1,
    1,1,1,1,
    1,1,1,1,
    1,1,1,0.5,0.5,
    1,1,1,1,
    1+0.5, 0.5, 1+1
};
int length;
int tonepin = 3; // Usar interfaz 3
void setup() {
    pinMode(tonepin, OUTPUT);
    length = sizeof(tune) / sizeof (tune[0]); // Calcular longitud
}

void loop() {
    for(int x = 0; x < length; x++)
    {
        tone(tonepin, tune[x]);
        delay(350 * durt[x]); // Esto se usa para ajustar el retraso de
                                // con el ritmo, 350 puede ajustarlo uste
                                // mismo
        noTone(tonepin);
    }
    delay(2000); // Retardo de 2 segundos.
}

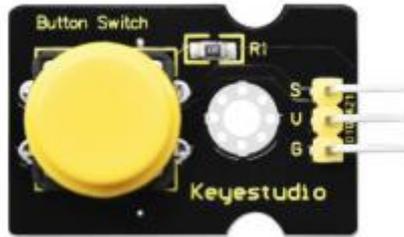
```

Cargue el código de prueba en la placa de desarrollo, ¿escucha “Himno a la alegría”?



Proyecto 4: Control de LED por módulo de botón

Descripción:



En este proyecto, controlaremos el encendido y apagado de la luz LED a través del módulo de botones. Cuando se presiona el botón, el extremo de la señal emite un nivel bajo (0); cuando se suelta, el extremo de la señal del sensor mantiene un nivel alto (1).

Equipo:

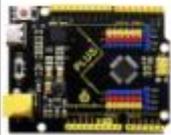
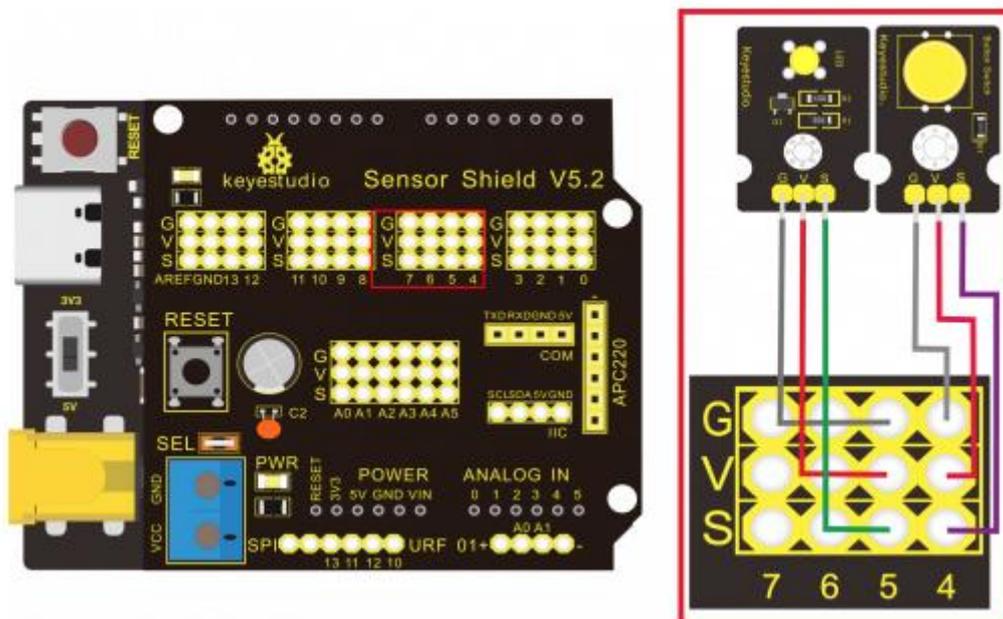
PLUS control board*1	Sensor shield*1	Yellow LED module*1	Button sensor*1	USB cable*1	3pinF-F Dupont line*2
					

Diagrama de conexión:



Nota: Los pines G, V y S del módulo del sensor de botón están conectados por separado a G, V y 4 en el escudo y los pines G, V, y S del módulo LED amarillo están conectados a G, V y 5 en el escudo.

Código de prueba:

Después de diseñar el programa, hacemos que el LED se encienda por el botón. En comparación con experimentos anteriores, agregamos una declaración de juicio condicional. Usamos la instrucción if. Las oraciones escritas de Arduino se basan en el lenguaje C, por lo tanto, la declaración de juicio de condición en C es adecuada para Arduino, como while, swich, etc.

Para esta lección, tomamos la declaración simple “if” como ejemplo para demostrar.

Si se presiona el botón, el nivel digital 4 es bajo, luego hacemos que la salida digital 5 sea de nivel alto, luego el LED se encenderá; por lo contrario, si se suelta el botón, el digital 4 es de nivel alto, hacemos que la salida digital 5 sea de bajo nivel, luego el LED se apagará.

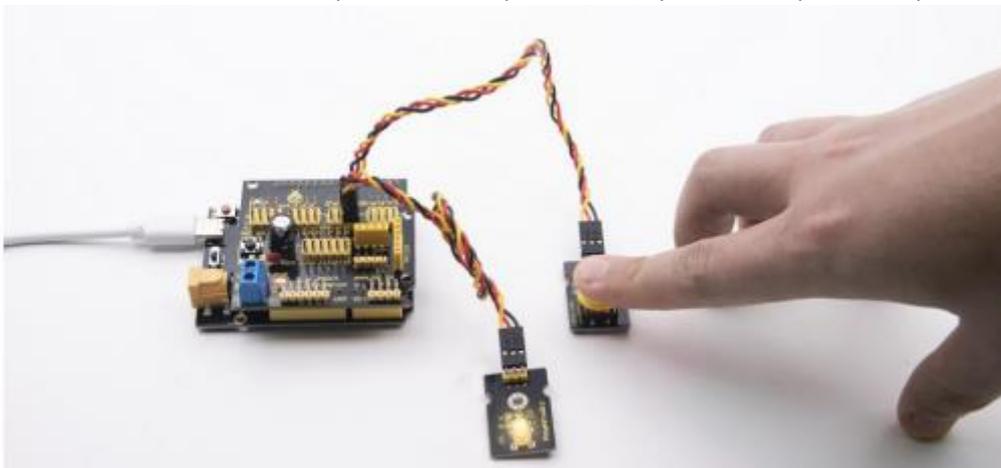
En cuanto a su referencia:

```
/*
Keyestudio smart home kit for Arduino
Proyecto 4
Botón
http://www.keyestudio.com
*/
int ledpin = 5; // Define la luz del en D5
int inpin = 4; // Define el botón en D4
int val; // Define la variable val
void setup() {
  pinMode (ledpin, OUTPUT); // La interfaz de luz LED se define como salida.
  pinMode (inpin, INPUT); // Define la interfaz del botón como entrada.
}

void loop() {
  val = digitalRead(inpin); // Lea el valor del nivel digital 4 y asígnelo a val.
  if(val == LOW) // Si se presiona el botón, la luz se encenderá.
  {digitalWrite(ledpin, HIGH);}
  else
  {digitalWrite(ledpin, LOW);}
}
```

Resultado de la prueba:

Este experimento es bastante simple y se aplica ampliamente a varios circuitos y aparatos eléctricos. En nuestra vida, puede encontrar este principio en cualquier dispositivo, como que la luz de fondo se enciende cuando presiona cualquier botón, que es el dispositivo típico.



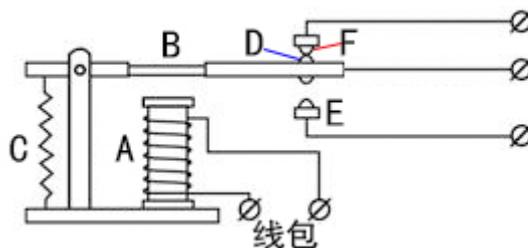
Proyecto 5: Módulo de relé de 1 canal

Descripción



Este módulo es un módulo dedicado Arduino y es compatible con la placa de expansión del sensor Arduino. Tiene un sistema de control (también llamado bucle de entrada) y un sistema controlado (también llamado bucle de salida). Comúnmente utilizado en circuitos de control automático, el módulo de relé en un “interruptor automático” que controla una corriente más grande y un voltaje más bajo con una corriente más pequeña y un voltaje más bajo.

Por lo tanto, desempeña el papel de ajuste automático, protección de seguridad y circuito de conversión en el circuito. Permite que Arduino maneje cargas de debajo de 3^a, como tiras de luz LED, motores de CC, bombas de agua de miniatura, interfaz enchufable de válvula solenoide. Los principales componentes internos del módulo de relé son el electroimán A, la armadura B, el resorte C, el contacto móvil D, el contacto estático (contacto normalmente abierto) E y el contacto estático (contacto normalmente cerrado) F (como se muestra en la figura).



Siempre que se aplique cierto voltaje a ambos extremos de la bobina, cierta corriente fluirá a través de la bobina para generar efectos electromagnéticos, y la armadura atraerá el núcleo de hierro contra la fuerza de tracción del resorte bajo la acción de la fuerza electromagnética, atracción impulsando así el contacto móvil y el contacto estático (contacto normalmente abierto) para atraer. Cuando se desconecta la bobina, la succión electromagnética también desaparecerá y la armadura volverá a la posición original bajo la fuerza de reacción del resorte, liberando el contacto móvil y el contacto estático original (contacto normalmente cerrado). Este tira y suelta, logrando así el propósito de encender y apagar en el circuito. Los contactos “normalmente abiertos y cerrados” del relé se pueden distinguir de esta manera: los contactos estáticos en estado desconectado cuando la bobina del relé esta apagada se denomina “contactos normalmente abiertos”, los contactos estáticos en estado conectado se denominan “contacto normalmente cerrado”. El módulo bien con 2 orificios de posicionamiento para que se pueda fijar el módulo a otros equipos.

Especificaciones:

- Voltaje de trabajo: 5V (CC)
- Interfaz: interfaz G, V, S.
- Señal de entrada: señal digital (nivel alto 1, nivel bajo 0)

- Contactos: contactos estáticos (contactos normalmente abiertos, contactos normalmente cerrados) y contactos móviles.
- Corriente nominal: 10ª (NA) 5V (NC).
- Tensión máxima de conmutación: 150V (CA) 24 V (CC).
- Corriente de descarga eléctrica: menos de 3ª.
- Peso: 15g.
- Tiempo de acción de contacto: 10 ms.

Equipo:

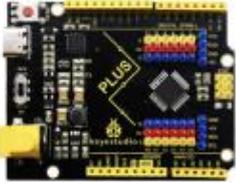
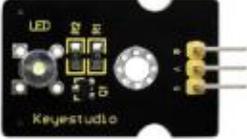
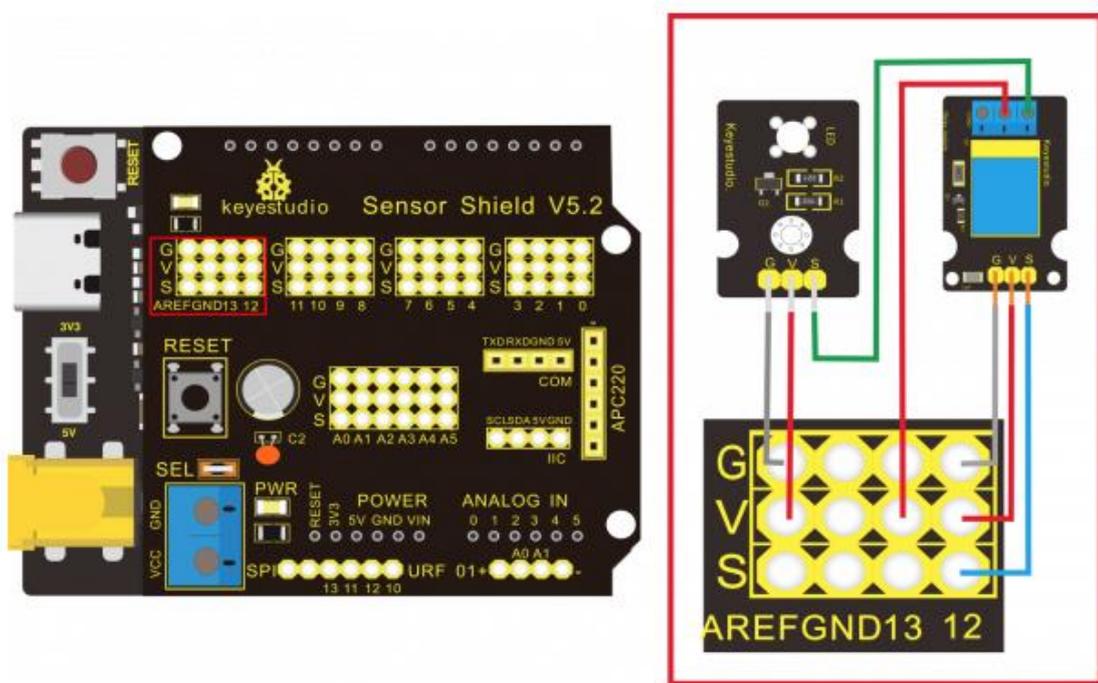
PLUS control board*1	Sensor shield*1	USB cable*1
		
Relay module*1	White LED*1	3pin F-F Dupont Line*1
		
Female to Female Dupont Lines*2	Male to Female Dupont Line*2	
		

Diagrama de conexión:



Nota: En el escudo, los pines G, V y S del módulo de relé de 1 canal están conectados a G, V y 12 respectivamente. El NO está vinculado con V; Los pines G, V y S del Led blanco están conectado respectivamente a G, V y al contacto estático de No en el módulo de relé.

Código de prueba:

```

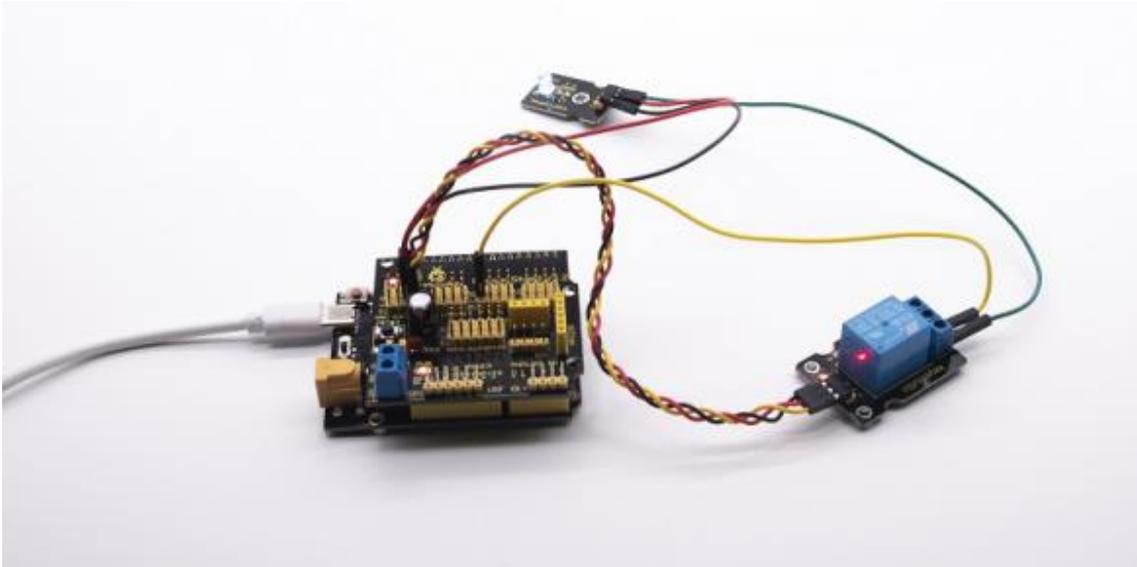
/*
Keystudio smart home Kit for Arduino
Proyecto 5
Relé
http://www.keystudio.com
*/
int Relay = 12; // Define el pin del relé en D12
void setup() {
  pinMode(13, OUTPUT); // Establece el Pin13 como salida.
  digitalWrite(13, HIGH); // Establece el Pin13 en ALTO.
  pinMode(Relay, OUTPUT); // Establece el Pin12 como salida.
}

void loop() {
  digitalWrite(Relay, HIGH); // Apague el relé.
  delay(2000); // Retardo de 2 segundos.
  digitalWrite(Relay, LOW); // Encienda el relé.
  delay(2000); // Retardo de 2 segundos.
}

```

Resultado de la prueba:

Conecte, encienda y cargue el código de prueba. El relé está conectado (“NO” está encendido, NC está apagado) durante 0,5 s. luego se desconecta durante 0,5 s (NC está encendido, NO está apagado) y alternativamente. Cuando el relé está conectado, el LED blanco estará encendido, por el contrario, el LED blanco se apagará.



Proyecto 6: Sensor de fotocélula

Descripción:



El sensor de fotocélula (fotorresistencia) es una resistencia hecha por el efecto fotoeléctrico de un semiconductor. Es muy sensible a la luz ambiental, por lo que su valor de resistencia varía con diferentes intensidades de luz. Usamos sus características para diseñar un circuito y generar un módulo sensor de fotorresistencia. El extremo de la señal del módulo está conectado al puerto analógico del microcontrolador. Cuando la intensidad de la luz aumenta, la resistencia y el voltaje del puerto analógico aumenta, es decir, el valor analógico del microcontrolador también aumenta. De lo contrario, cuando la intensidad de la luz disminuye, la resistencia aumenta y el voltaje del puerto analógico disminuye. Es decir, el valor analógico del microcontrolador se vuelve más pequeño. Por lo tanto, podemos usar el módulo sensor de fotorresistencia para leer el valor analógico correspondiente y detectar la intensidad de luz en el ambiente. Se aplica comúnmente a la medición, el control y la conversión de luz, así como el circuito de la luz.

Especificaciones:

- Voltaje de funcionamiento 3,3 V – 5 V (CC)
- Interfaz: Interfaz 3PIN
- Señal de salida: señal analógica.
- Peso 2,3 g.

Equipo

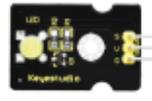
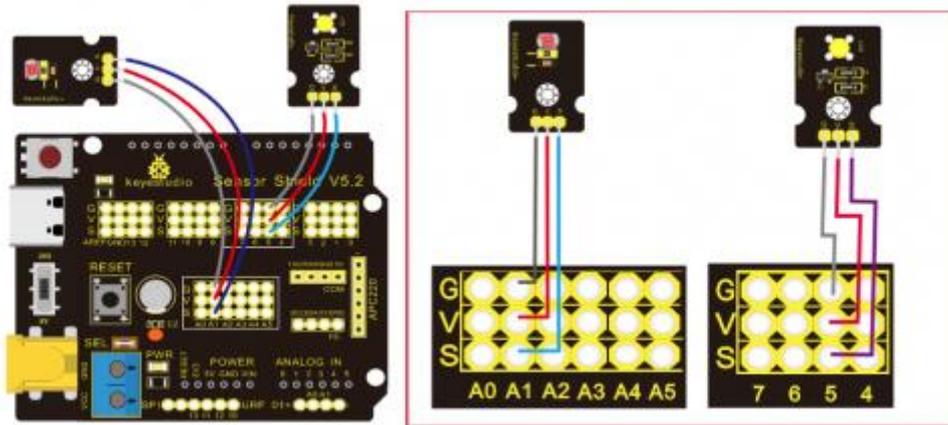
PLUS control board*1	Sensor shield*1	Photocell sensor*1	Yellow LED module*1	USB cable*1	3pin F-F Dupont line*2
					

Diagrama de conexión:



Nota: En la placa de expansión, los pines G,V y S del módulo sensor de fotocélula están conectados a G, V y A1; los pines G, V y S del módulo LED amarillo están vinculados con G, V y 5 por separado.

Código de pruebas:

```

/*
Keystudio smart home kit for Arduino
Proyecto 6
Fotocélula
http://www.keystudio.com
*/
int LED = 5; // Establece pin LED en D5
int val = 0; // Lee el valor del voltaje del fotodiodo
void setup() {
  pinMode(LED, OUTPUT); // LED es de salida
  Serial.begin(9600); // La tasa de baudios del puerto serie
  | | | | | | | | // se establece en 9600
}

```

```

void loop() {
  val = analogRead(A1); // Lee el valor voltaje del Pin A1
  Serial.println(val); // Puerto serie para ver el cambio
  | | | | | | | | | | // de valor del voltaje
  if(val < 900)
  {
    // Menos de 900, la luz LED está apagada
    digitalWrite(LED, LOW);
  }
  else
  {
    // De lo contrario el LED se enciende
    digitalWrite(LED, HIGH);
  }
  delay(10); // Retardo 10 ms
}

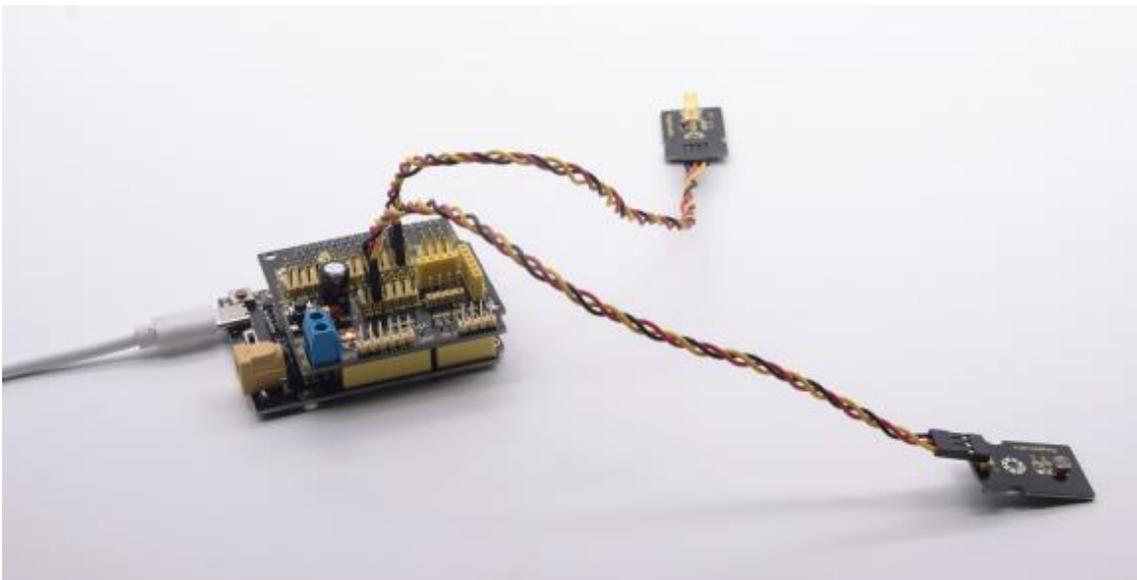
```

Resultado de la prueba:

El LED se encenderá después de cargar el código de prueba, apunte al sensor de la fotocélula con la linterna (o el flash del teléfono), verá que el LED se apaga automáticamente. Sin embargo, retire la linterna, el LED encenderá nuevamente.

Revisión Para esta cadena de código, es simple. Leemos el valor a través del puerto analógico, tenga en cuenta que la cantidad analógica no necesita modo de entrada y salida. Lea el valor analógico del sensor de fotocélula por el puerto analógico.

El valor analógico disminuirá gradualmente una vez que haya luz, el valor es de hasta 1000, este valor se puede elegir según el brillo que necesite. Seleccione el método: coloque todo el dispositivo en el entorno donde el LED está apagado, abra el monitor en serie para verificar el valor mostrado, reemplace 1000 con este valor. El valor de lectura del monitor en serie es una buena manera de modular el código.



Proyecto 7: Ajuste del ángulo del servomotor

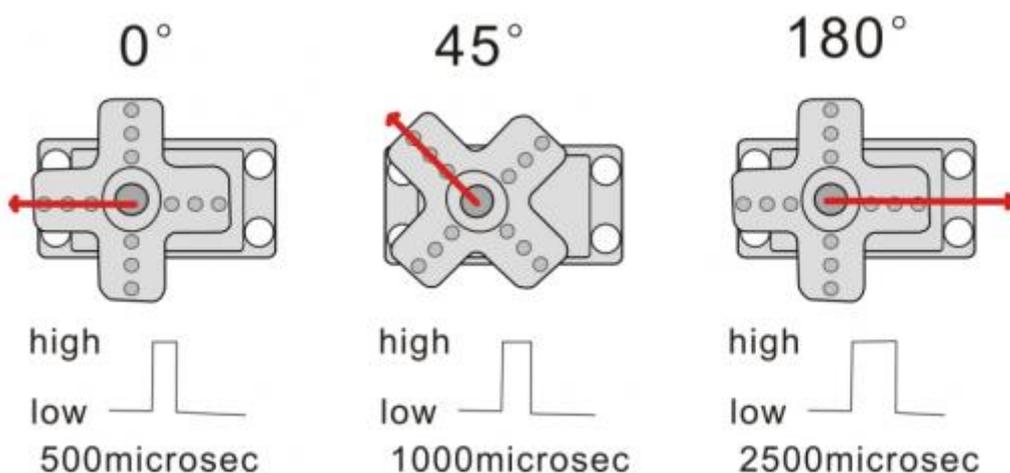
Descripción:



Cuando hacemos este kit, a menudo controlamos puertas y ventanas con servos. En este curso, presentamos su principio y cómo usar los servomotores. El servomotor es un actuador rotativo de control de posición. Se compone principalmente de carcasa, placa de circuito, motor sin núcleo, engranaje y sensor de posición. Su principio de funcionamiento es que el servo recibe la señal enviada por MCU o receptor y produce una señal de referencia con un período de 20 ms y un ancho de 1,5 ms, luego compara el voltaje de polarización de CC adquirido con el voltaje del potenciómetro y genera una diferencia de voltaje.

El servomotor viene con muchas especificaciones. Pero todos ellos tienen tres cables de conexión, que se distinguen por los colores marrón, rojo y naranja (diferentes marcas pueden tener diferentes colores). El marrón es para GND, el rojo para la potencia positiva, el naranja para línea de señal.

El ángulo de rotación del servomotor se controla regulando el ciclo de trabajo de la señal PWM (Modulación de ancho de pulso). El ciclo estándar de la señal PWM es de 20 ms (50Hz). En teoría, el ancho se distribuye entre 1 ms y 2 ms., pero en realidad está entre 0,5 ms. y 2,5 ms. El ancho corresponde al ángulo de rotación de 0° a 180°. Pero tenga en cuenta que para motores de diferentes marcas, la misma señal puede tener un ángulo de rotación diferente.



Hay dos formas de controlar un servomotor con Arduino. Una es usar un puerto de sensor digital común de Arduino par producir una onda cuadrada con un ciclo de trabajo diferente para simular la señal PWM y usar esta señal para controlar el posicionamiento del motor. Otra forma es usar directamente la función Servo del Arduino para controlar el motor. De esta manera, el programa será más fácil, pero solo puede controlar un motor de dos contactos porque para la función servo, solo pueden usar los pines digitales 9 y 10.

La capacidad de la unidad Arduino es limitada. Entonces, si necesitas controlar más de un motor, necesitarás alimentación externa.

Especificaciones:

Voltaje de trabajo: CC 4,8 V ~ 6 V Rango de ángulo de funcionamiento: alrededor de 180° (a 500 → 2500 µseg) Rango de ancho de pulso: 500 → 2500 µseg Velocidad sin carga: 0,12 ± 0,01 s/60 (CC 6 V) Corriente sin carga: 200 ± 20 mA (CC 4,8 V) 220 ± 20 mA (CC 6 V) Par de parada: 1,3± 0,01 kg cm (CC 4,8 V) 1,5±0,1 kg cm (CC6V) Corriente de parada: ≤ 850 mA (CC 4,8 V) ≤1000 mA (CC 6 V) Corriente en espera: 3 ± 1 mA (CC 4,8 V) 4 ± 1 mA (CC 6V) Longitud del cable: 250 ± 5 mm. Tamaño de apariencia: 22,9 * 12,2 * 30 mm. Peso: 9 ± 1 g (sin bocina de servo) Temperatura de almacenamiento -20°C ~ 60°C Temperatura de funcionamiento: -10°C ~ 50°C.

Equipo de experimentación:

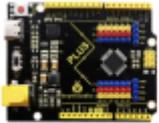
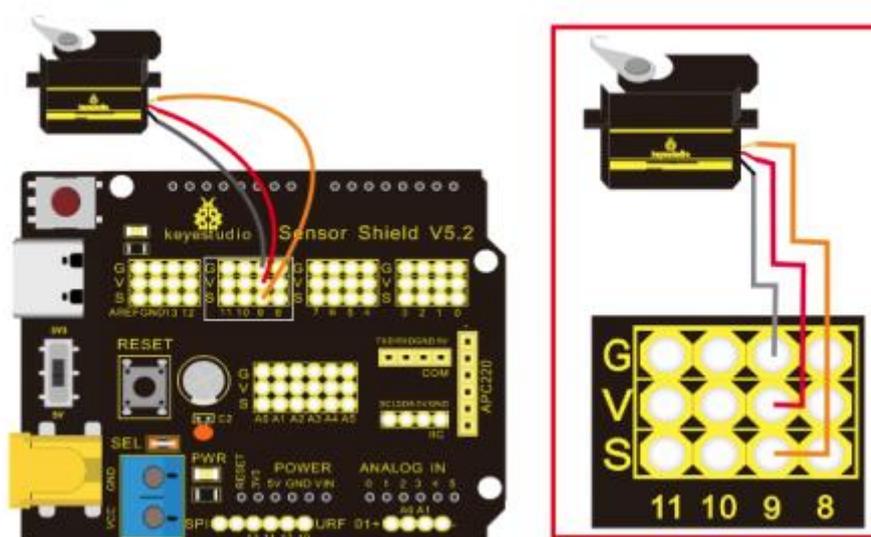
PLUS control board*1	Sensor shield*1	Servo*1	USB cable*1
			

Diagrama de conexión:



Nota: El servo está conectado a G (GND), V (VCC), 9. El cable marrón del servo está conectado a Gnd (G), el cable rojo está conectado a 5v (V) y el cable naranja está conectado al pin digital 9.

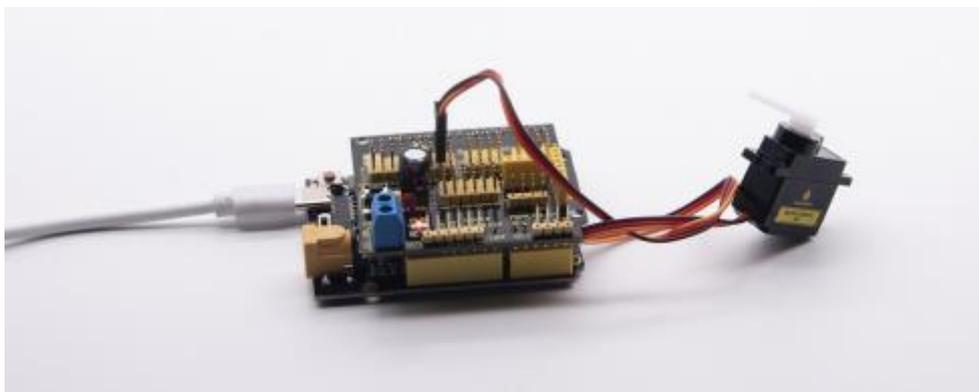
Código de prueba:

```
/*
Keyestudio smart home Kit for Arduino
Proyecto 7
Servo
http://www.keyestudio.com
*/
#include <Servo.h> // Biblioteca de funciones de servo
Servo myservo;
int pos = 0; // Ángulo de inicio del servo
void setup() {
  myservo.attach(9); // Define la posición del servo en D9
}

void loop() {
  for(pos = 0; pos < 180; pos += 1) // Ángulo de 0 a 180 grados
  {
    myservo.write(pos); // El ángulo del servo en pos.
    delay(15); // Retardo de 15 ms.
  }
  for(pos = 180; pos >= 1; pos -= 1) // Ángulo de 180 a 0 grados
  {
    myservo.write(pos); // El ángulo del servo en pos.
    delay(15); // Retardo de 15 ms.
  }
}
```

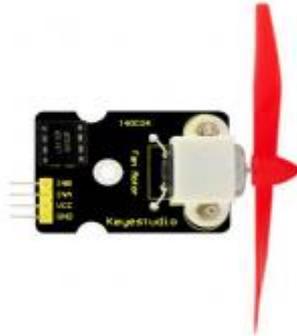
Resultado de la prueba: a 180

Cargue el código, conecte según el diagrama de conexión y enciéndalo. El servo gira de 0° a 180° y luego de 180° ~ 0°.



Proyecto 8: Módulo de ventilador

Descripción:



El módulo de ventilador L9110 adopta el chip de control del motor L9110, puede controlar la dirección de rotación y la velocidad del motor. Además, este módulo es eficiente y cuenta con un ventilador de alta calidad, que puede apagar la llama a una distancia de 20 cm.

Del mismo modo, también es una parte importante del robot de fuego.

Especificaciones:

- Diámetro del ventilador: 75 mm.
- Voltaje de trabajo: 5V.

Equipo:

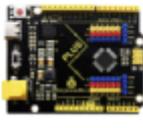
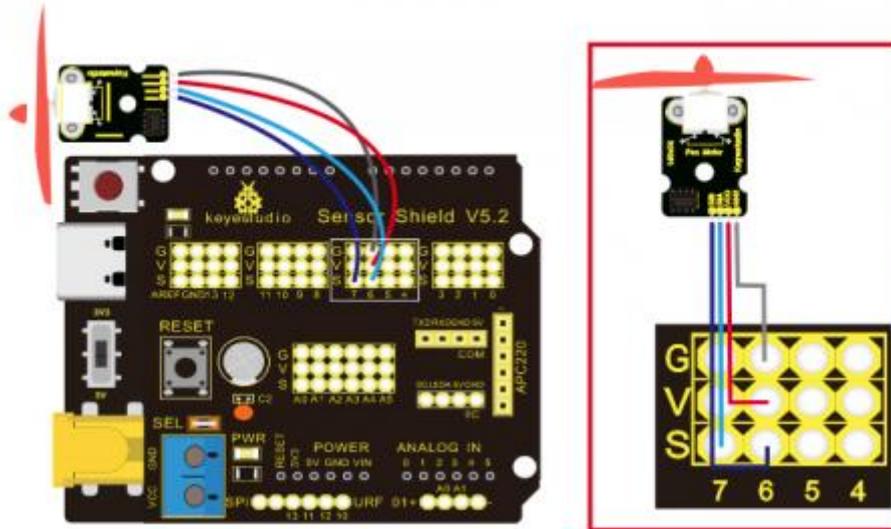
PLUS control board*1	Sensor shield*1	Fan module*1	USB cable*1	Female to Female Dupont lines*4
				

Diagrama de conexión:



Nota: En el escudo, los pines GND, VCC, INA e INB del módulo de ventilador están conectados respectivamente a G, V, 7, 6.

Código de prueba:

```

/*
Keystudio smart home kit for Arduino
Proyecto 8
Ventilador
http://www.keystudio.com
*/

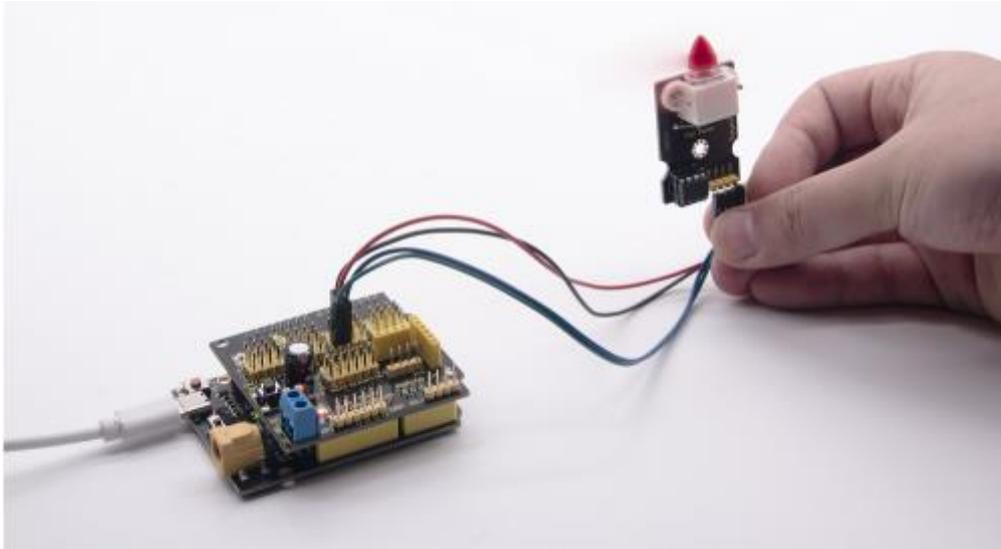
void setup() {
  pinMode(7, OUTPUT); // Define pin D7 como de salida
  pinMode(6, OUTPUT); // Define pin D6 como de salida
}

void loop() {
  digitalWrite(7, LOW);
  digitalWrite(6, HIGH); // Rotación inversa del motor.
  delay(3000); // Retardo de 3 segundos
  digitalWrite(7, LOW);
  digitalWrite(6, LOW); // El motor deja de girar.
  delay(1000); // Retardo de 1 segundos
  digitalWrite(7, HIGH);
  digitalWrite(6, LOW); // El motor gira en la dirección de avance.
  delay(3000); // Retardo de 3 segundos
}

```

Resultado de la prueba:

Cargue el código de prueba, cablee de acuerdo con el diagrama de conexión, el interruptor DIP se marca en el lado derecho y enciende. El ventilador gira en sentido contrario a las agujas del reloj durante 3000 ms, se detiene durante 1000 ms y luego gira en sentido de las agujas del reloj durante 3000 ms.



Proyecto 9: Sensor de vapor

Descripción:



Este es un sensor de vapor de uso común. Su principio es detectar la cantidad de agua mediante líneas paralelas impresas desnudas en la placa de circuito. Cuanta más agua haya, más cables se conectarán. A medida que aumenta el área de contacto conductor, el voltaje de salida aumentará gradualmente. También puede detectar vapor de agua en el aire. El sensor de vapor se puede utilizar como detector de agua de lluvia e interruptor de nivel. Cuando aumenta la humedad en la superficie del sensor, el voltaje de salida aumentará.

El sensor es compatible con varias placas de control de microcontroladores, como los microcontroladores de Arduino, proporcionamos la guía para operar el sensor de vapor y el tablero de control Arduino. Conecte el extremo de la señal del sensor al puerto analógico del microcontrolador, detecte el cambio de valor analógico y muestre el valor analógico correspondiente en el monitor serie.

Nota: La parte de conexión no es resistente al agua, lo la sumerja en el agua, por favor.

Especificaciones:

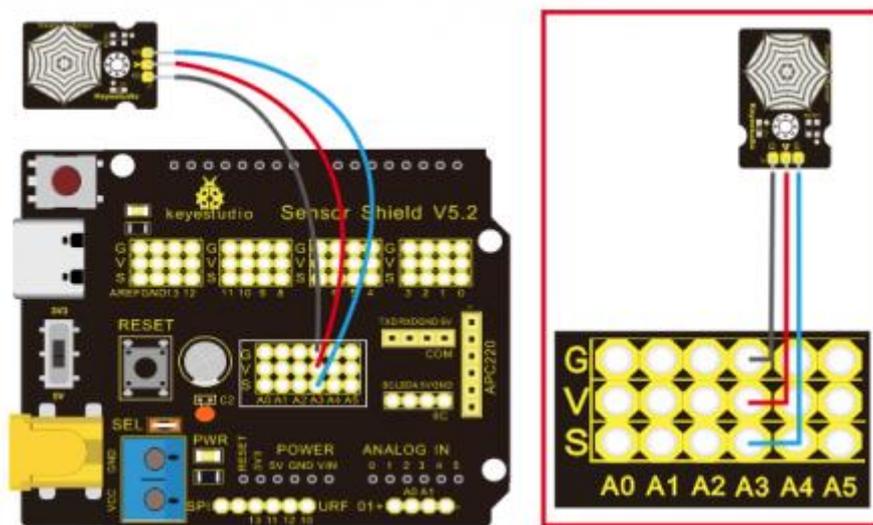
- Voltaje de funcionamiento: CC 3,3 – 5V.
- Corriente de trabajo: <20mA.
- Rango de temperatura de funcionamiento: -10 °C ~ +70°C.
- Señal de control: Salida de señal analógica.
- Interfaz: interfaz de 3 pines de 2,54 mm.

- Tamaño: 35 * 20 * 8 mm.
- Peso: 2,2 g.
- S: Salida de señal.
- V (+): Fuente de alimentación (VCC)
- G (-): Tierra (GND)

Equipo:

PLUS control board*1	Sensor shield*1	Steam sensor*1	USB cable*1	3pinF-F Dupont line*1
				

Diagrama de conexión:



Nota: En el escudo del sensor, los pines G, V y S del sensor de vapor están conectados a G, V y A3; los pines G, V y S del sensor de fotocélula están conectados a G, V y A1; los pines G, V y S del led amarillo están vinculados con G, V y 5; la línea Marrón está conectada con G, el cable rojo con V, el cable naranja con 9.

Código de prueba:

```

/*
Keystudio smart home Kit for Arduino
Proyecto 0
Vapor
http://www.keystudio.com
*/

```

```

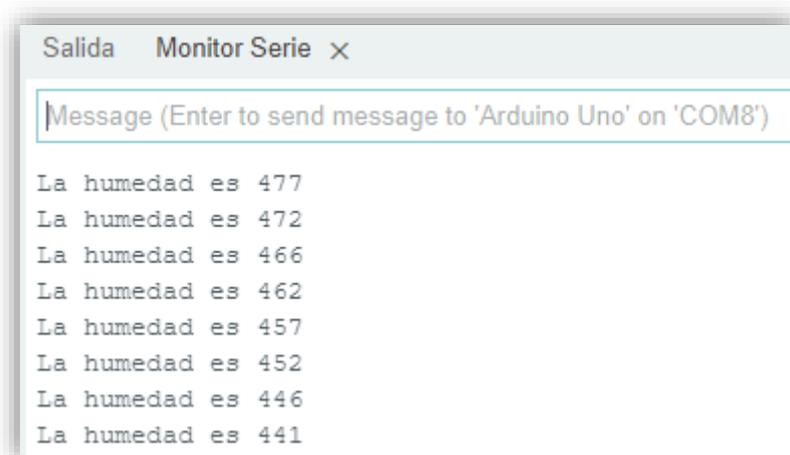
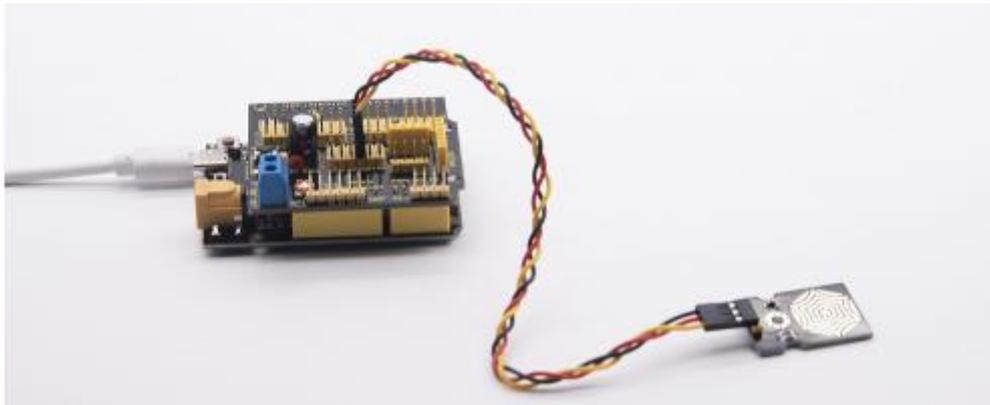
void setup() {
  Serial.begin(9600); // Abra el puerto seri y establezca
  // la velocidad de transmisión en 9600 bps
}

void loop() {
  int val;
  val = analogRead(A3); // Enchufe el sensor de vapor en el puerto
  // analógico A3
  Serial.print("La humedad es ");
  Serial.println(val, DEC); // Lee valor analógico a través del
  // puerto serial.
  delay(100); // Retardo 100 ms
}

```

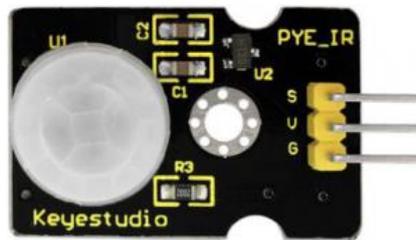
Resultado de la prueba:

Al detectar diferentes grados de humedad, el sensor obtendrá la retroalimentación de un valor actual diferente. Se muestra como la siguiente imagen. Cuando el sensor detecta el vapor de agua hervida, el valor de humedad se muestra en el monitor de serie del software Arduino.



Proyecto 10: Sensor de movimiento PIR

Descripción:



El sensor de movimiento infrarrojo piroeléctrico puede detectar señales infrarrojas de una persona o animal en movimiento y emitir señales de conmutación. Se puede aplicar a una variedad de ocasiones para detectar el movimiento del cuerpo humano. Los sensores infrarrojos piroeléctricos convencionales son mucho más grande, con circuitos complejos y menos confiabilidad. Ahora lanzamos este nuevo sensor de movimiento infrarrojo piroeléctrico, especialmente diseñado para Arduino. Este sensor integra un sensor infrarrojo piroeléctrico digital integrado, y los pines de conexión. Presenta mayor confiabilidad, menor consumo y un circuito periférico más simple.

Especificaciones:

- Voltaje de entrada: CC 3,3 V ~ 18 V.
- Corriente de trabajo: 15uA.
- Temperatura de trabajo: -20 ~ 85 grados centígrados.
- Voltaje de salida: Alto 3 V, bajo 0 V.
- Tiempo de retardo de salida: (nivel alto): alrededor de 2,3 a 3 segundos.
- Ángulo de detección: alrededor de 100°.
- Distancia de detección: 3-4 metros.
- LED indicador de salida (luz de alto nivel)
- Corriente límite de pines: 100mA.

Nota especial:

- 1.- La distancia máxima es de 3-4 metros durante la prueba.
- 2.- Al realizar la prueba, primero abra la lente blanca, puede ver la parte de detección rectangular. Cuando la línea larga de la parte de detección rectangular es paralela al suelo, la distancia es mayor.
- 3.- Al realizar la prueba, el sensor debe cubrirse con una lente blanca; de lo contrario, afectará la distancia.
- 4.- La distancia en mejor a 25°C, y la distancia de detección se acorta cuando supera los 30°C.
- 5.- Terminado de encender y cargar el código, debe esperar de 5 a 10 segundos ky luego comenzar a probar, de lo contrario, no es sensible.

Equipo:

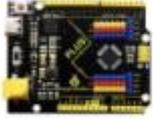
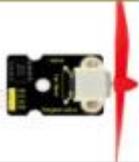
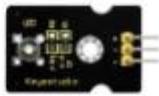
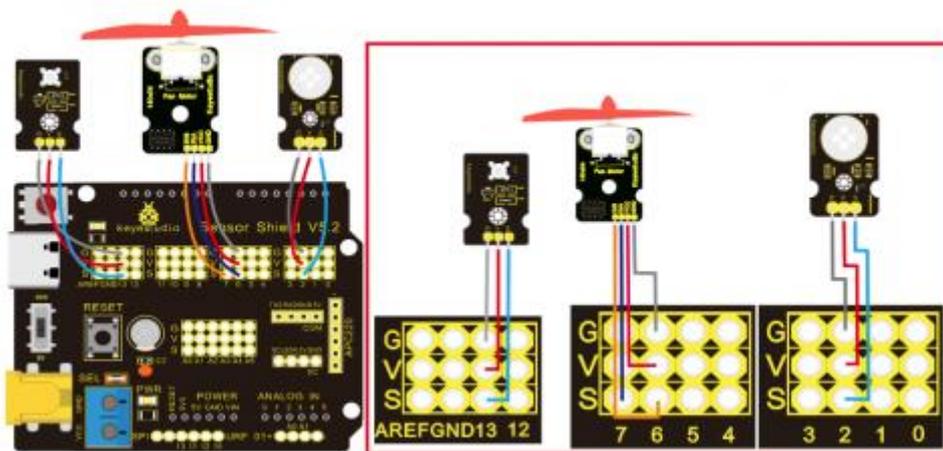
PLUS control board*1	Sensor shield*1	PIR Motion Sensor*1	Female to Female Dupont lines*4
			
Fan module*1	White LED module*1	USB cable*1	3pinF-F Dupont line*2
			

Diagrama de conexión:



Nota: En el escudo la G, V y S del sensor de movimiento PIR están conectados a G, V y 2; GND, VCC, INA e INB del módulo de ventilador están conectado a G, V, 7, 6. Los pines del módulo LED vinculados con G, V y 13.

Código de pruebas:

```

/*
Keystudio smart home Kit for Arduino
Proyecto 10
Sensor de movimiento PIR
http://www.keystudio.com
*/

```

```

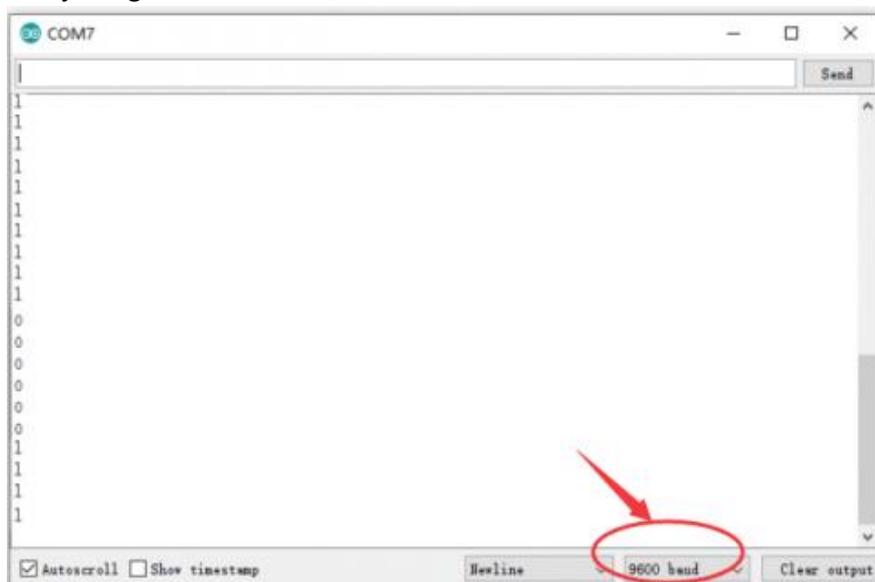
void setup() {
  Serial.begin(9600); // Abra el puerto serie y establezca la
  // velocidad de transmisión en 9600 bps
  pinMode(2, INPUT); // Definir PIR como entrada en D2
  pinMode(13, OUTPUT); // Define LED como salida de D13
  pinMode(7, OUTPUT); // Definir D7 como salida
  pinMode(6, OUTPUT); // Definir D6 como salida.
}

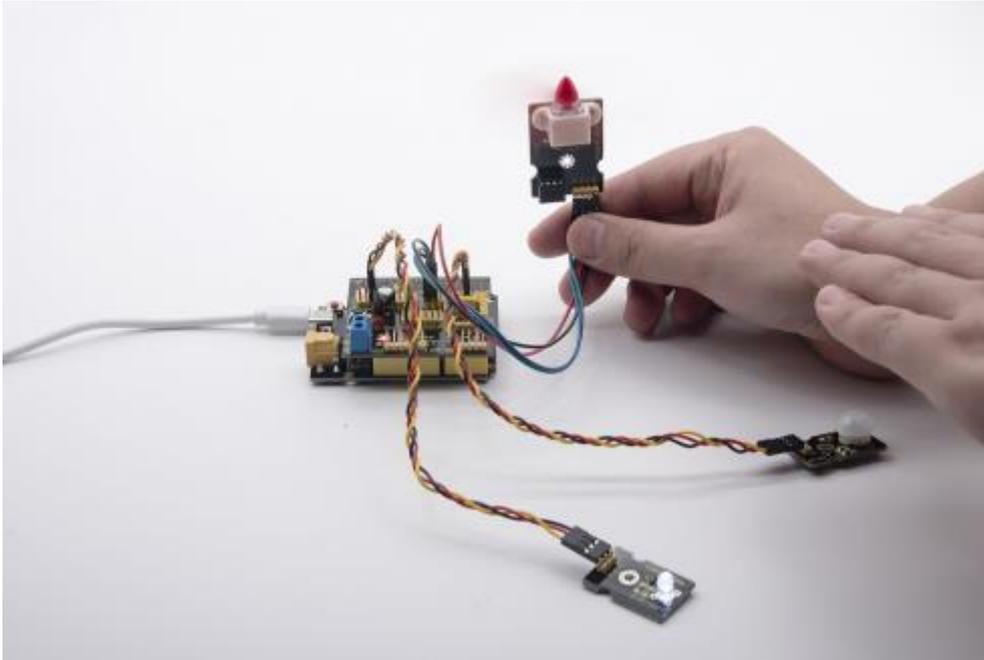
void loop() {
  Serial.println(digitalRead(2));
  delay(500); // Retardo de 500 ms.
  if(digitalRead(2)== 1) // Si alguien es detectado caminando
  {
    digitalWrite(13, HIGH); // La luz LED está encendida
    digitalWrite(7, HIGH);
    analogWrite(6, 150); // El ventilador gira
  }else // Si no detecta persona caminando
  {
    digitalWrite(13, LOW); // La luz LED está apagada
    digitalWrite(7, LOW);
    analogWrite(6, 0); // El ventilador no gira
  }
}
}

```

Resultado de la prueba:

Cargue el código de prueba, abra el monitor en serie y configure la velocidad de transmisión en 9600. Si el sensor de movimiento PIR detecta a las personas alrededor, el monitor en serie muestra "1", el D13 y el LED blanco se encienden al mismo tiempo, el ventilador gira. Si no hay nadie alrededor, el monitor serie muestra "0", el indicador D13 y el LED blanco están apagados. El ventilador deja de girar.





Proyecto 11: Sensor analógico (MQ-2)

Descripción:



Este sensor de gas se utiliza para alarmas de fugas de gas domésticas, alarmas de gas combustibles industrial e instrumentos portátiles de detección de gas. Y es adecuado para la detección de gas licuado, benceno, alcano, alcohol, hidrógeno, etc., y se usa ampliamente en varios sistemas de alarmas contra incendios. El sensor de humo MQ-2 puede ser con precisión un detector de gases múltiple y tiene la ventaja de alta sensibilidad, respuesta rápida, buena estabilidad, larga vida útil y circuito de accionamiento simple. Puede detectar la concentración de gas inflamable y humo en el rango de 300~10000ppm. Mientras tanto, tiene una alta sensibilidad de gas natural, gas licuado de petróleo y otros humos, especialmente al humo de los alcanos. Debe calentarse durante un período de tiempo antes de usar el sensor de humos, de lo contrario, la resistencia de salida y el voltaje no son precisos. Sin embargo, el voltaje de calentamiento no debe ser demasiado alto.

Pertenece al material sensible de gas semiconductor de dióxido de estaño y pretende al semiconductor de tipo N de tipo ion de superficie. A cierta temperatura, el dióxido de estaño adsorbe oxígeno en el aire y forma adsorción de iones negativos de oxígeno, reduciendo la densidad de electrones en el semiconductor, aumentando así su valor de resistencia. Cuando está en contacto con gas inflamable en el aire y la niebla tóxica, si la niebla tóxica se ajusta a la barrera potencial en el límite del grano, hará que cambie la conductividad de la superficie. Con esto se puede obtener información sobre la presencia de humo o gas inflamable. Cuanto mayor sea la concentración de humo o gas inflamable en el aire, mayor será la conductividad y menor la resistencia de salida, mayor será la salida de la señal analógica. El sensor bien con un orificio

de posicionamiento, lo cual es conveniente para usted para fijar a otro dispositivos. Además, la sensibilidad se puede ajustar girando el potenciómetro.

Especificaciones:

- Voltaje de funcionamiento: 3,3~5 V (CC)
- Interfaz: 4 pines (VCC, GND, D0, A0)
- Señal de salida: señal digital y señal analógica.
- Peso: 7,5 g.

Equipo:

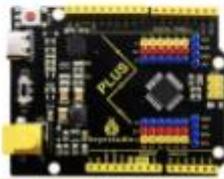
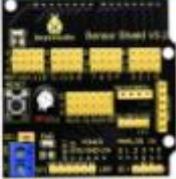
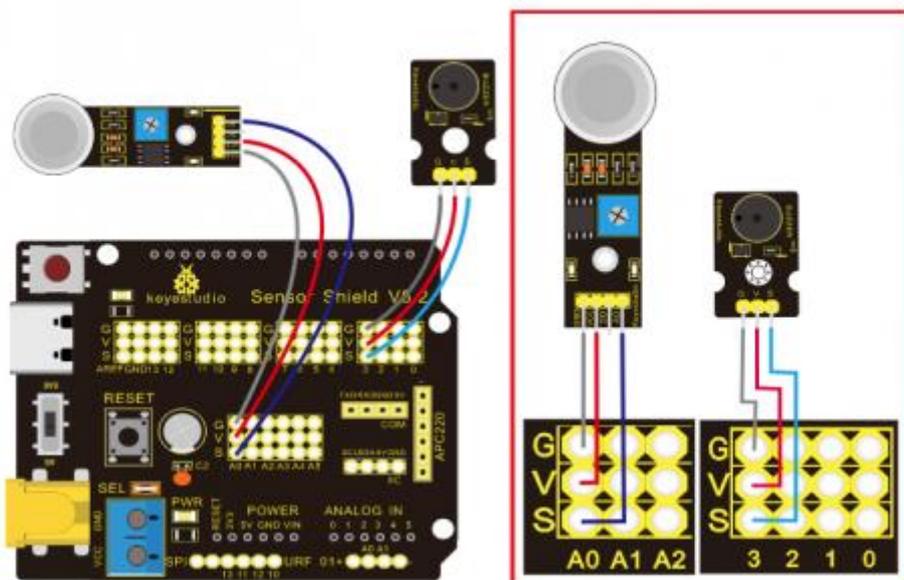
PLUS control board*1	Sensor shield*1	MQ-2 gas sensor*1	3pinF-F Dupont line*1
			
Passive buzzer*1	USB cable*1	F-F Dupont line*3	
			

Diagrama de conexión:



Nota: En el escudo, los pines GND, VCC, D0 y A0 del sensor de gas están vinculados con los pines GND, VCC, INA e INB del módulo de ventilación están conectados a G, V, 7 y 6. Los pines G, V y

S del zumbador pasivo están conectados a G, V y 3; el pin G, V y S del LED amarillo están conectados a G, V y 5.

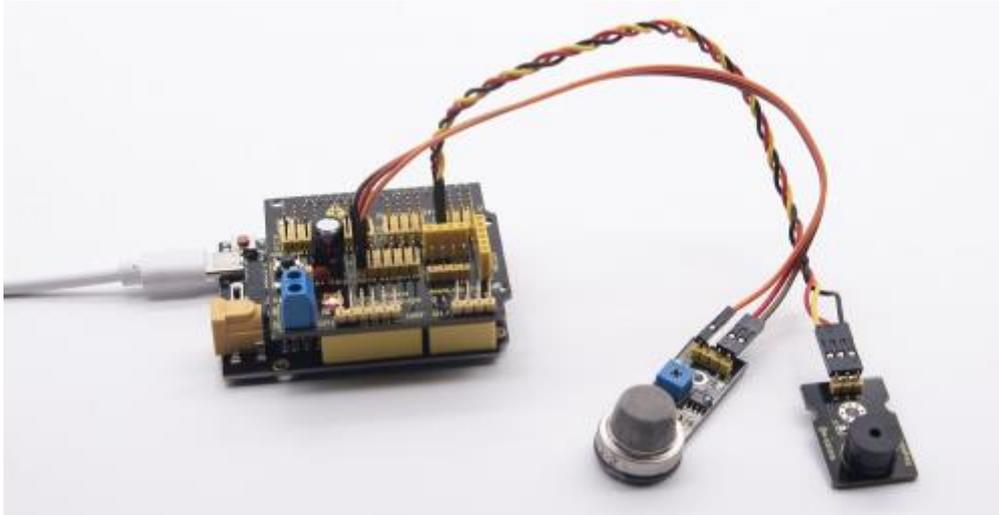
Código de prueba:

```
/*
Keyestudio smart home Kit for Arduino
Proyecto 11
Gas
http://www.keyestudio.com
*/
int MQ2 = A0; // Definir MQ2 gas sensor en el pin A0
int val = 0; // Declarar variable
int buzzer = 3; // Definir buzzer pin D3
void setup() {
  pinMode(MQ2, INPUT); // MQ2 sensor de gas como entrada
  Serial.begin(9600); // Establecer la tasa de baudios del
  // puerto serie en 9600
  pinMode(buzzer, OUTPUT); // Establecer el mode de pin de E/S
  // digital para la salida
}

void loop() {
  val = analogRead(MQ2); // Leer el valor del voltaje del puerto A0
  // y se lo asignamos a la variable val
  Serial.println(val); // Serial port envia el valor de val.
  if(val>450)
  {
    tone(buzzer, 589);
    delay(300);
  }
  else
  {
    noTone(buzzer);
  }
}
```

Resultado de la prueba:

Cargue el código de prueba, cablee de acuerdo con el diagrama de conexión y enciéndalo. Cuando el sensor de gas detecta el gas inflamable, sonará el zumbador pasivo, el ventilador girará y el LED amarillo se encenderá, cuando no hay gas inflamable, el zumbador pasivo no sonará, el ventilador no girará y el LED amarillo estará apagado.



Proyecto 12: Pantalla LCD 1602

Descripción:



Con el módulo de comunicación I2C, este módulo de visualización que puede mostrar 2 líneas con 16 caracteres por línea. Muestra un fondo azul y una palabra blanca y se conecta a la interfaz I2C de MCU, lo que ahorra mucho de los recursos de MCU. En la parte posterior de la pantalla LCD, hay potenciómetro azul para ajustar la luz de fondo. La dirección de comunicación por defecto es 0x17. La pantalla CLD 1602 original puede iniciarse y ejecutarse con 7 puertos IO, pero la nuestra está construida con interfaz Arduino IIC/I2C, ahorrando 5 puertos IO. Alternativamente, el módulo bien con 4 orificios de posicionamiento con un diámetro de 3 mm, lo que es conveniente para que lo fije en otros dispositivos.

Tenga en cuenta que cuando la pantalla se vuelve más brillante o más oscura, los caracteres se vuelven más o menos visibles.

Especificaciones:

- Dirección I2C: 0x27
- Luz de fondo (azul, blanco)
- Tensión en alimentación: 5V
- Contraste: ajustable
- GND: un pin que conecta a tierra.
- VCC: un pin que conecta a la fuente de alimentación de +5V
- SDA: un pin que se conecta al puerto analógico A4 para la comunicación IIC.
- SCL: un pin que se conecta al puerto analógico A5 para la comunicación IIC.

Equipo:

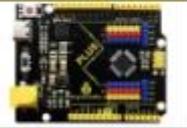
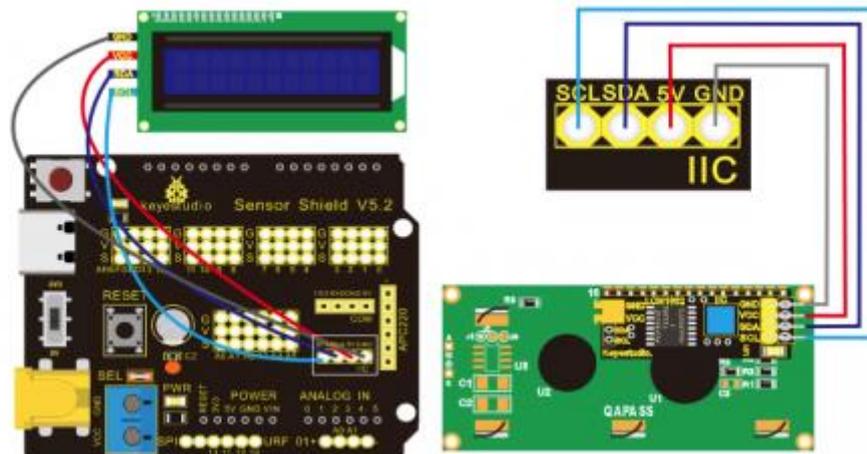
PLUS control board*1	Sensor shield*1	1602 LCD Display*1	USB cable*1	4pinF-F Dupont line*1
				

Diagrama de conexión:



Nota: hay pines GND, VCC; SDA y SCL en el módulo 1602LCD. GND está vinculado con GND (-) de comunicación IIC, VCC está conectado a 5V (+), SDA a SDA, SCL a SCL.

Código de prueba:

```

/*
Keyestudio smart home Kit for Arduino
Proyecto 12
1602 LCD
http://www.keyestudio.com
*/

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x27, 16, 2); // Establece la dirección LCD en 0x27
// para pantalla de 16 caracteres y 2 líneas
void setup() {
  lcd.init(); // Inicializa el lcd
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("Hello, world!"); // Muestra en el lcd "Hola, mundo!"
  lcd.setCursor(2,1);
  lcd.print("keyestudio!"); // Muestra en el lcd "keyestudio!"
}

void loop() {
}

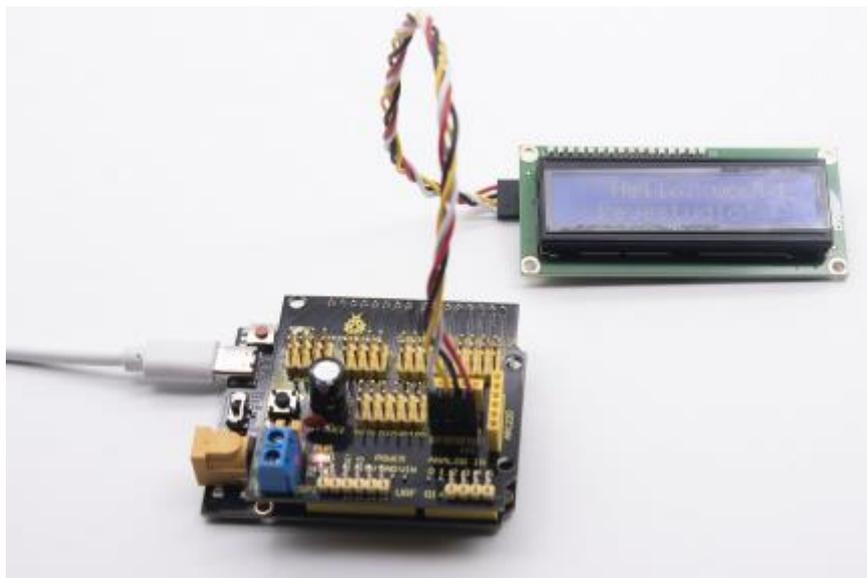
```

Resultado de la prueba:

Después de conectar y cargar el código de muestra, la primera línea en pantalla LCD imprime “¡Hola mundo!”, la segunda línea imprime “Keystudio!”, con un potenciómetro para ajustar la luz de fondo de la pantalla LCD.



Nota. Cablee de acuerdo con el diagrama de conexión, cargue el código y después del encendido, cuando la pantalla no muestre caracteres, puede ajustar el potenciómetro detrás del 1602LCD y la luz de fondo para que el 1602LCD muestre la cadena de caracteres correspondiente.



Proyecto 13: Sensor de humedad del suelo

Descripción:



Este es un sensor de humedad del suelo simple que tiene como objetivo detectar la humedad del suelo. Si el suelo carece de agua, el valor analógico emitido por el sensor disminuirá; de lo contrario, aumentará. Si usa este sensor para hacer un dispositivo de riego automático, puede detectar si su botánica tiene sed para evitar que se marchite cuando salga. El uso del sensor con

el controlador Arduino hace que su planta sea más cómoda y su jardín más inteligente. El módulo sensor de humedad del suelo no es tan complicado como podría pensar, y si necesita detectar el suelo en su proyecto, será su mejor opción. El sensor se configura con dos sondas insertadas en el suelo, luego, con la corriente que atraviesa el suelo, el sensor obtendrá el valor de resistencia al leer los cambios de corriente entre las dos sondas y convertir dicho valor de resistencia en contenido de humedad. A mayor humedad (menor resistencia), mayor conductividad tiene el suelo. Insértelo en el suelo y luego use el convertidor AD para leerlo. Con ayuda de este sensor, la planta puede recordarte: necesito agua.

Especificación:

- Voltaje de fuente de alimentación: 3,3 V o 5 V.
- Corriente de trabajo: $\leq 20\text{mA}$.
- Voltaje de salida: $\approx 2,3\text{ V}$ (cuando el sensor está totalmente sumergido en agua, el voltaje será de 2,3 V) cuanto mayor sea la humedad, mayor será el voltaje de salida.
- Tipo de sensor: Salida analógica.
- Definición de interfaz: señal S, G-GND, V – VCC.
- Empaque: sellado electrostático de bolsas.
- Tamaño_ 63 * 20 * 8 mm.
- Peso: 2,5 g.

Equipo:

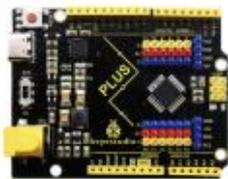
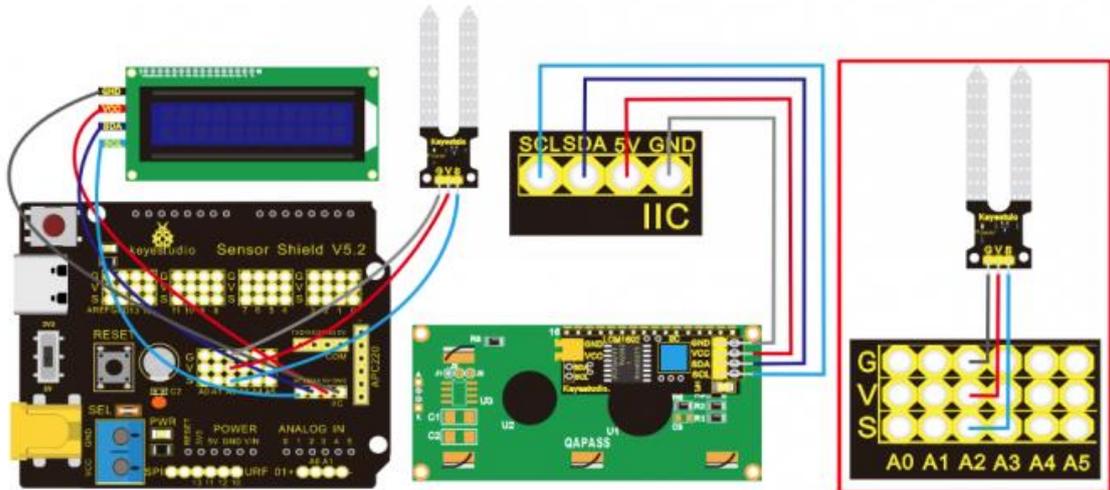
PLUS control board*1	Sensor shield*1	Soil humidity sensor*1	1602 LCD display*1
			
USB cable*1	4pinF-F Dupont line*1	3pinF-F Dupont line*1	
			

Diagrama de conexión:



Nota: En el escudo, los pines G,V y S del sensor de humedad del suelo están conectados a G, V y A2; GND de 1602LCD está vinculado con GND de comunicación ICC , VCC está conectado a 5V (+), SDA a SDA, SCL a SCL.

Código de prueba

```

/*
Keyestudio smart home Kit for Arduino
Proyecto 13
Sensor de humedad
http://www.keyestudio.com
*/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
volatile int value;
LiquidCrystal_I2C mylcd (0x27, 16, 2); // Establece la dirección LCD en 0x27
| | | | | | | | | | | | | | | | // Pantalla de 16 palabras y 2 líneas.
void setup() {
  Serial.begin(9600); // Establecer una comunicación puerto serie de 9600
  value = 0;
  mylcd.init();
  mylcd.backlight(); // Enciende la luz de fondo.
  mylcd.clear(); // Borra pantalla
  pinMode(A2, INPUT); // Sensor de suelo en A2, se ingresa en modo entrada
}

void loop() {
  Serial.print("Valor de humedad del suelo: "); //muestra en el monitor serie.
  Serial.println(value);
  delay(500); // Retención de 500 ms.
  value = analogRead(A2); // Lee el valor del sensor de humedad
  if(value < 300) // Si value es menor a 300.
  {
    mylcd.clear(); // Borra pantalla
    mylcd.setCursor(0,0);
    mylcd.print("value:");
    mylcd.setCursor(6,0);
  }
}

```

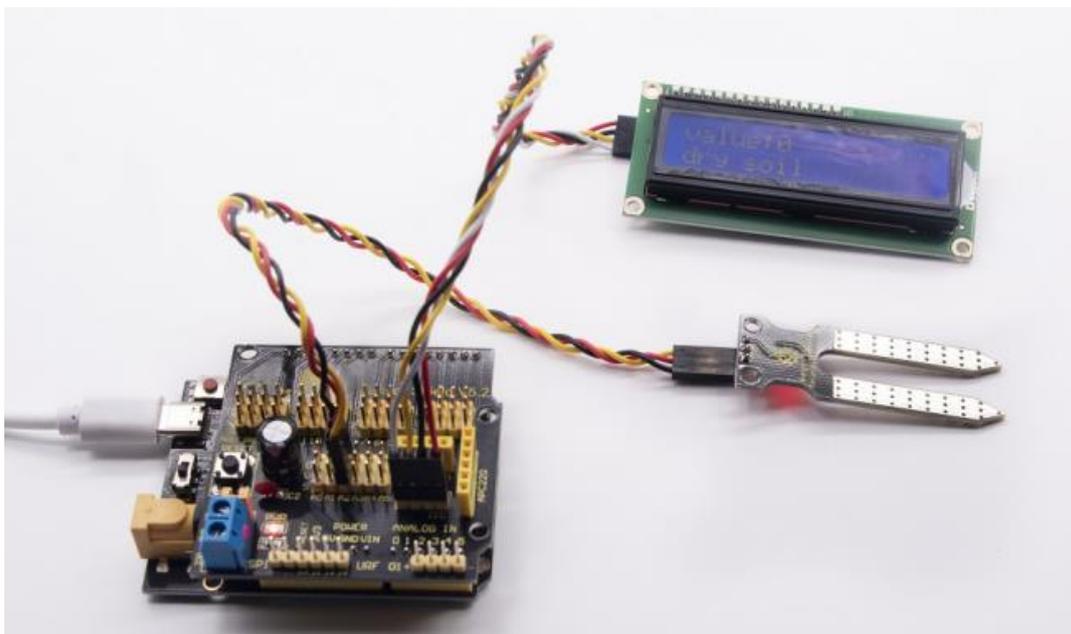
```

mylcd.print(value);
mylcd.setCursor(0,1);
mylcd.print("suelo seco"); // Muestra texto en lcd
delay(2000);
}
else if ((value >= 300) && (value <= 700)) // Si esta entre un valor de 300 hasta 700
{
mylcd.clear(); // Borra pantalla
mylcd.setCursor(0,0);
mylcd.print("value:");
mylcd.setCursor(6,0);
mylcd.print(value);
mylcd.setCursor(0,1);
mylcd.print("suelo humedo"); // Muestra texto en lcd
delay(2000);
}
else // si está en un valor mayor 700
{
mylcd.clear(); // Borra pantalla
mylcd.setCursor(0,0);
mylcd.print("value:");
mylcd.setCursor(6,0);
mylcd.print(value);
mylcd.setCursor(0,1);
mylcd.print("en agua"); // Muestra texto en lcd
delay(2000);
}
}
}

```

Resultado de la prueba:

Conecte de acuerdo con el diagrama de cableado, grabe el programa y enciéndalo. Abra el monitor en serie e inserte el sensor de humedad del suelo en el suelo. Cuando mayor es la humedad, mayor es el número, en el rango de 0-1023, El sensor de suelo se inserta en el suelo y el agua con diferente humedad y el 1620LCD muestra el valor correspondiente.



Proyecto 14: Prueba de Bluetooth

En el siglo XX, la tecnología ha cambiado nuestra vida. Las personas pueden trabajar en casa con dispositivos inalámbricos como mouse, impresora y altavoz, lo que mejora enormemente nuestro nivel de vida. Bluetooth puede hacer que el trabajo en casa sea fácil, así como el entretenimiento. Los usuarios pueden controlar de forma inalámbrica el archivo de audio desde el PC o Apple iPod dentro de 30 pulgadas. La tecnología Bluetooth también se puede usar en adaptadores, lo que permite a las personas compartir su vida diaria con amigos de Internet y las redes sociales.

Mando a distancia bluetooth



La tecnología Bluetooth es una tecnología inalámbrica estándar que permite el intercambio de datos a corta distancia entre dispositivos fijos, dispositivos móviles y la construcción de redes de área personal (utilizando ondas de radio UHF en banda ISM de 2,4 a 2,485 GHz). Este kit está equipado con el módulo Bluetooth HM-10, que es una máquina maestro-esclavo. Cuando se usa como host, puede enviar comandos al esclavo de forma activa; cuando se usa como esclavo, solo puede recibir comandos del host. El módulo Bluetooth HM-10 es compatible con el protocolo Bluetooth 4.0, que no solo es compatible con dispositivos móviles Android, sino que también es compatible con el sistema iOS. En el experimento, usamos por defecto el módulo Bluetooth HM-10 como esclavo y el teléfono celular como Anfitrión. Instalamos la APP Bluetooth en el teléfono móvil, conectando el módulo; finalmente use la aplicación Bluetooth para controlar las partes del kit de la casa inteligente.

Parámetros del módulo Bluetooth HM-10:

Pines	Descripción
BRK	Como pin de entrada, control de presión corta o pulso único de entrada de 100 ms bajo. Nivel para archivar las siguientes funciones: 1.- Cuando el módulo está en estado de reposo: El módulo se activa en estado normal, si está abierto AT+NOTI, el puerto enviará OK+WAKE. 2.- Cuando está en estado conectado: El módulo solicitará activamente desconectarse. Cuando está en modo de espera: El módulo estará en estado inicial.
RXD	Entrada de datos en serie.
TXD	Salida de datos en serie.
GND	Cable de tierra.
VCC	Polo positivo de potencia, entrada 5 V.
STATE	Como pin de salida, muestra el estado de funcionamiento del módulo Flash lentamente en estado de espera – repetir pulso de 500 ms; Siempre la luz está en estado de conectado – nivel alto. Puede establecer que no parpadee en el estado de espera, siempre que se encuentre en el estado conectado.

- Protocolo Bluetooth: Especificación Bluetooth V4.0 BLE.
- Sin límites de bytes en el puerto serie Transceptor.
- En un entorno abierto, realice una comunicación de ultra distancia de 100 m con iphone4s.
- Protocolo USB: USB V2.0.
- Frecuencia de trabajo: banda ISM de 2,4 GHz.
- Método de modulación: -23dbm, -6dbm, 0dbm, 6dbm, se puede modificar mediante el comando AT.
- Sensibilidad: $\leq -84\text{dbm}$ a 0,1% VER.
- Tasa de transmisión: Asíncrona: 6K bytes; Sincrono: 6K Bytes.
- Características de seguridad: Autenticación y encriptación.
- Servicio de soporte: UUID central y periférico FFE0, FFE1.
- Consumo de energía: modo de reposo automático, corriente en espera 400uA~800uA, 8,5 mA durante la transmisión.
- Fuente de alimentación: 5V CC.
- Temperatura de trabajo: -5 a +65 Centígrados.

Uso de la aplicación Bluetooth

Descripción

En la lección anterior, presentamos el principio básico de los parámetros del módulo Bluetooth HM-10. En este proyecto, le mostraremos cómo usar el módulo Bluetooth HM-10. Para controlar de manera eficiente este kit mediante el módulo Bluetooth HM-10, diseñados especialmente una aplicación, como se muestra a continuación.



Hay doce botones de control y cuatro controles deslizantes en la aplicación. Cuando conectamos el módulo Bluetooth HM-10 y la aplicación, solo presione el botón de control de la aplicación, y

el Bluetooth del teléfono celular envía un carácter de control. El módulo Bluetooth recibirá un carácter de control correspondiente. Al programar, configuramos la función correspondiente de cada sensor o módulo de acuerdo con el carácter de control clave correspondiente. A continuación, probemos los 12 botones de la aplicación.

Aplicación para móvil Android:

Nota: Permita que la aplicación acceda a la “ubicación” en la configuración del teléfono celular cuando se conecte al módulo Bluetooth; de lo contrario, es posible que Bluetooth no esté conectado.

Ingresa a Google Play, busque “keys IoT”, si no puede buscarlo en la tienda de aplicaciones, descargue la aplicación en el siguiente enlace:

https://play.google.com/store/apps/details?id=com.keyestudio .iot_keys

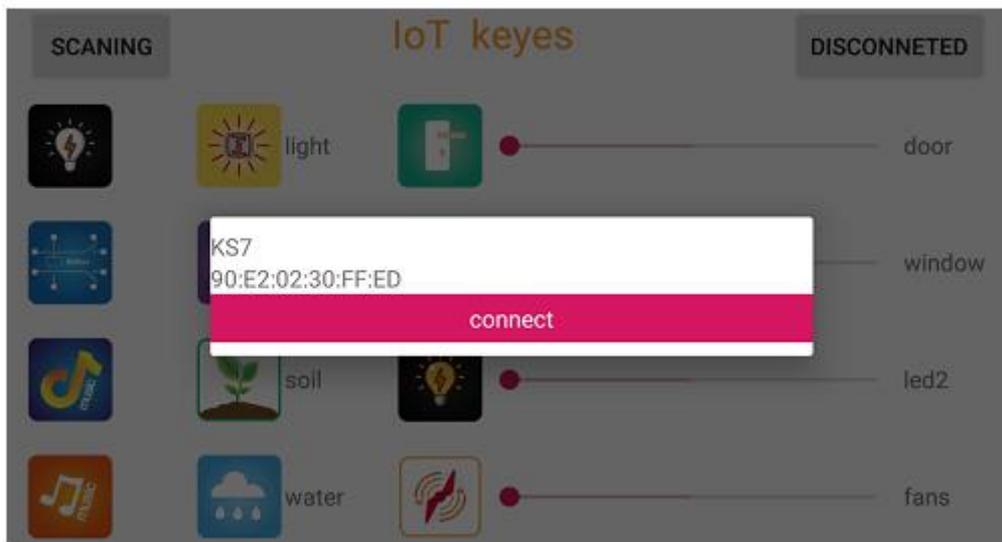
Después de instalar y abrir la aplicación.



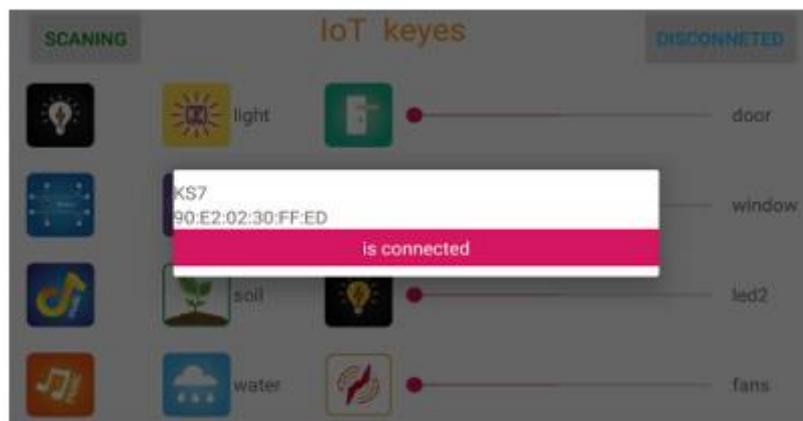
La interfaz aparece como se muestra a continuación.



Cargue el código y enciéndalo, el LED del módulo Bluetooth parpadea. Inicie Bluetooth y abra la aplicación par hacer clic en “CONECTAR” para conectarse.

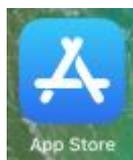


Haga clic en “Conectar”, Bluetooth está conectado con éxito. Como se muestra a continuación, el LED del módulo Bluetooth normalmente está encendido.

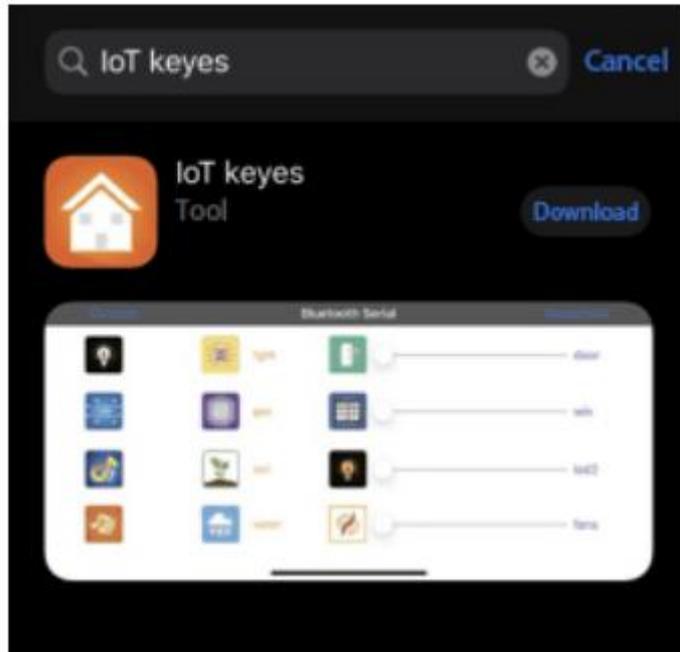


Para el sistema iOS:

1.- Abrir tienda de aplicaciones



2.- Busque “IoT keys” en la tienda de aplicaciones, luego haga clic en “download”.

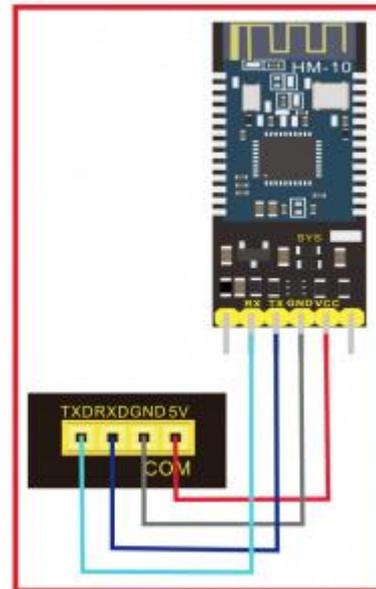
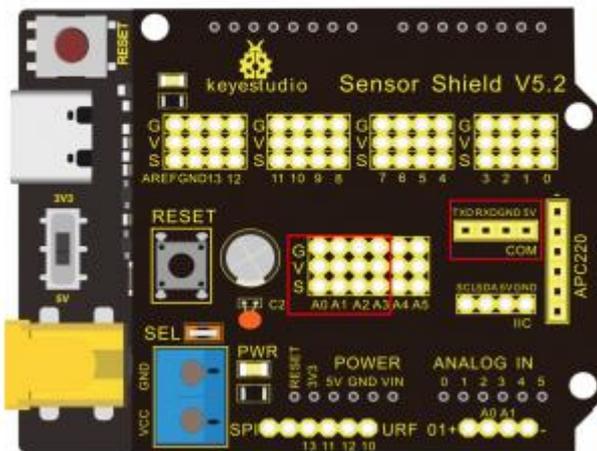


3.- Después de instalar correctamente y abrir la aplicación se muestra el siguiente interfaz:



4.- Cargue correctamente el código de prueba, inserte el módulo Bluetooth y enciéndalo. El LED del módulo parpadea. Inicie Bluetooth en el teléfono celular, luego haga clic en “conectar” a la izquierda para buscar Bluetooth y emparejar. Después de emparejar con éxito, el LED del módulo Bluetooth está encendido. Nota: Retire el módulo Bluetooth, por favor, cuando cargue el código de prueba. De lo contrario, el programa no se podrá cargar. Conecte el módulo Bluetooth y Bluetooth para emparejar después de cargar el código de prueba.

Diagrama de conexión:



Nota: En la placa de expansión del sensor, RXD, TXD, GND y VCC del módulo Bluetooth están conectados respectivamente a TXD, RXD, GND y 5V, y no es necesario conectar los pines STATE y BRK del módulo Bluetooth. Conecte la fuente de alimentación.

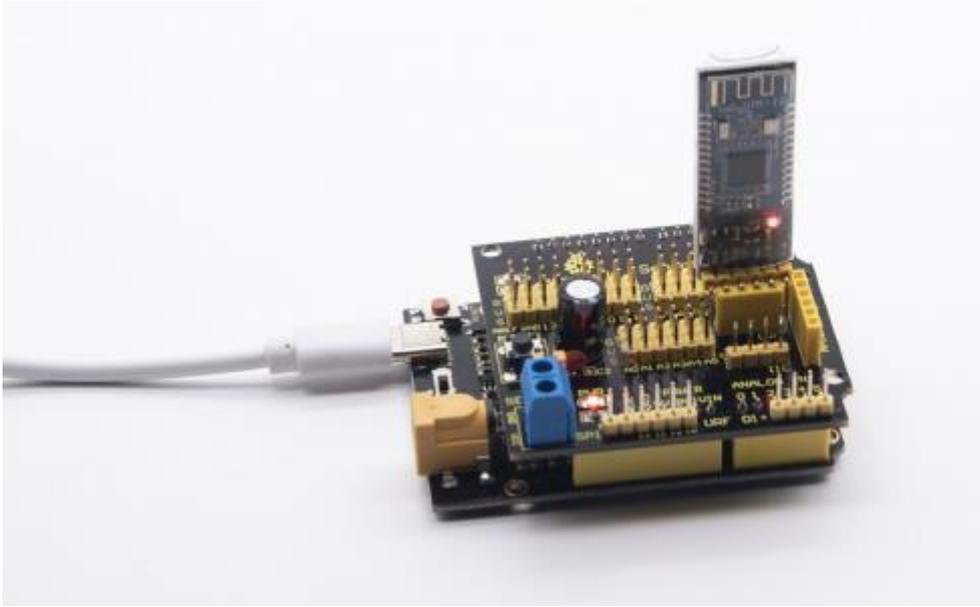
Código de prueba:

```

/*
Keyestudio smart home Kit for Arduino
Proyecto 14
Bluetooth
http://www.keyestudio.com
*/
char val;
void setup() {
  Serial.begin(9600); // Establezca la tasa de baudios
  // del puerto serie en 9600
}

void loop() {
  while (Serial.available() > 0)
  {
    val = Serial.read(); // Leer el valor enviado por Bluetooth
    Serial.print(val); // El puerto serial imprime el valor leído
  }
}

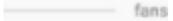
```



Funciones clave en la aplicación:

No.	Botón	Control Carácter	Función	No.	Botón	Control Carácter	Funcion
1	SCANNING		Empaje y conéctese al módulo Bluetooth HM-10.	2	DISCONNECT		Desconectar Bluetooth.
3			Haga clic para enviar "a", haga clic de nuevo para enviar "b"	4			Haga clic para encender el LED blanco, haga clic nuevamente para apagar el LED.
5			Mantener y presionar para enviar "e", liberar para enviar "g"	6			Haga clic para enviar "c", haga clic de nuevo para enviar "d"
7			Mantener y presionar para enviar "e", liberar para enviar "g"	8			Haga clic para enviar "i", haga clic de nuevo para enviar "S"
9			Haga clic para enviar "h", haga clic de nuevo para enviar "s"	10			Haga clic para enviar "j", haga clic de nuevo para enviar "S"
			Haga clic para reproducir música.				Haga clic para encender el sensor de fotocélula, la luz muestra los datos; haga clic de nuevo para apagar el sensor de fotocélula.
			Haga clic para reproducir música.				Haga clic para encender el sensor de gas, el gas muestra los datos detectados; haga clic de nuevo para apagar el sensor de gas.
			Haga clic para reproducir música.				Haga clic para encender el sensor de vapor, el agua muestra los datos detectados; haga clic de nuevo para apagar el sensor de vapor.

11		Haga clic para enviar "l", haga clic de nuevo para enviar "m"	Haga clic para abrir la puerta; haga clic de nuevo para cerrar la puerta.	12		Arrastre el control deslizante para enviar "t 50 #", 't' representa el carácter inicial; 50 en el ángulo del servo1; '#' implica carácter de terminación.	El control deslizante controla el ángulo del servo 1 para controlar la puerta, la pantalla de la puerta y luego el valor del ángulo del servo 1.
13		Haga clic para enviar "n", haga clic de nuevo para enviar "o"	Haga clic para abrir la ventana; haga clic de nuevo para cerrar la ventana.	15		Arrastra el deslizador para enviar "u 34 #" 'u' representa el carácter inicial; 34 en el ángulo del servo 2; '#' representa el carácter de terminación.	El control deslizante controla el ángulo del servo 2 par controlar la ventana, muestra el valor del ángulo del cero 2.
15		Haga clic para enviar "p", haga clic de nuevo para enviar "q"	Haga clic para encender el LED; haga clic de nuevo para apagar el LED.	16		Arrastra el control deslizante para enviar "v 100 #", 'v' representa el carácter inicial; 100 es el valor PWM de led2; '#' significa el carácter de terminación.	El control deslizante controla el brillo del LED, el valor de brillo de la pantalla led2.

17		Haga clic para enviar "r", haga clic de nuevo para enviar "s"	Haga clic para encender el ventilador; haga clic de nuevo para apagar el ventilador.	18		Arrastra el control deslizante para enviar "w 153 #"; 'w' representa el carácter inicial; 153 el valor PWM del ventilador; '#' representa el carácter de terminación.	El control deslizante controla la velocidad de rotación, los ventiladores indican el valor de la velocidad de rotación.
----	---	---	--	----	---	---	---

10.- Guía de montaje

Compruebe primero la placa A~I y las piezas.

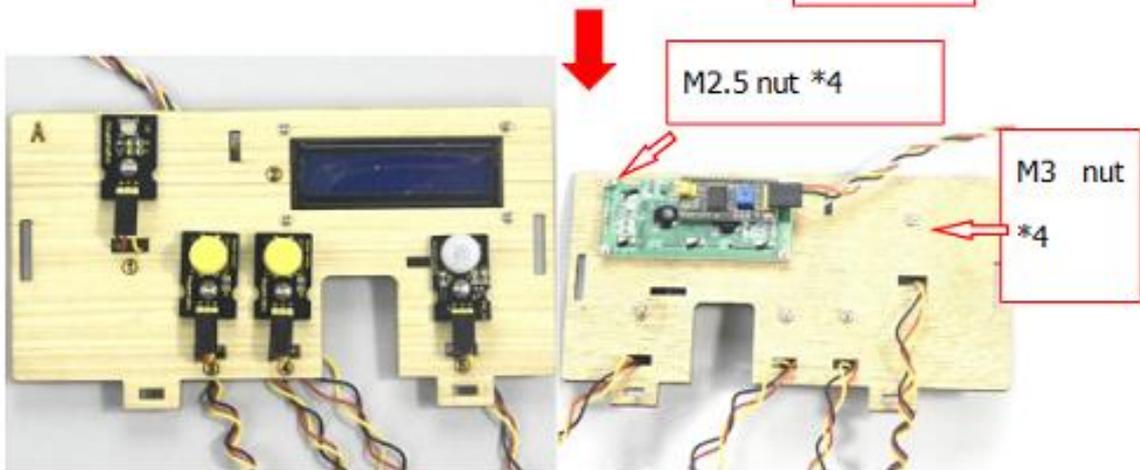
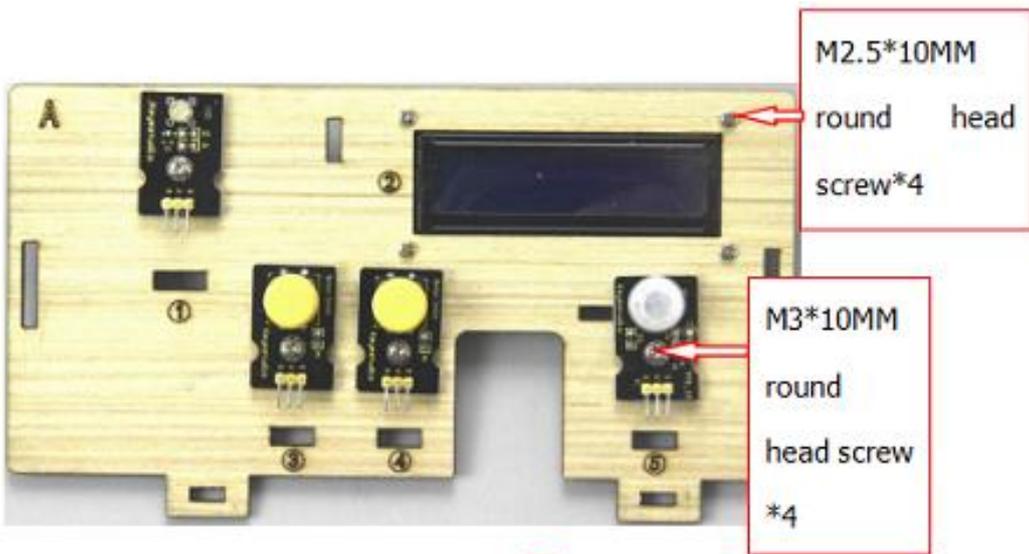
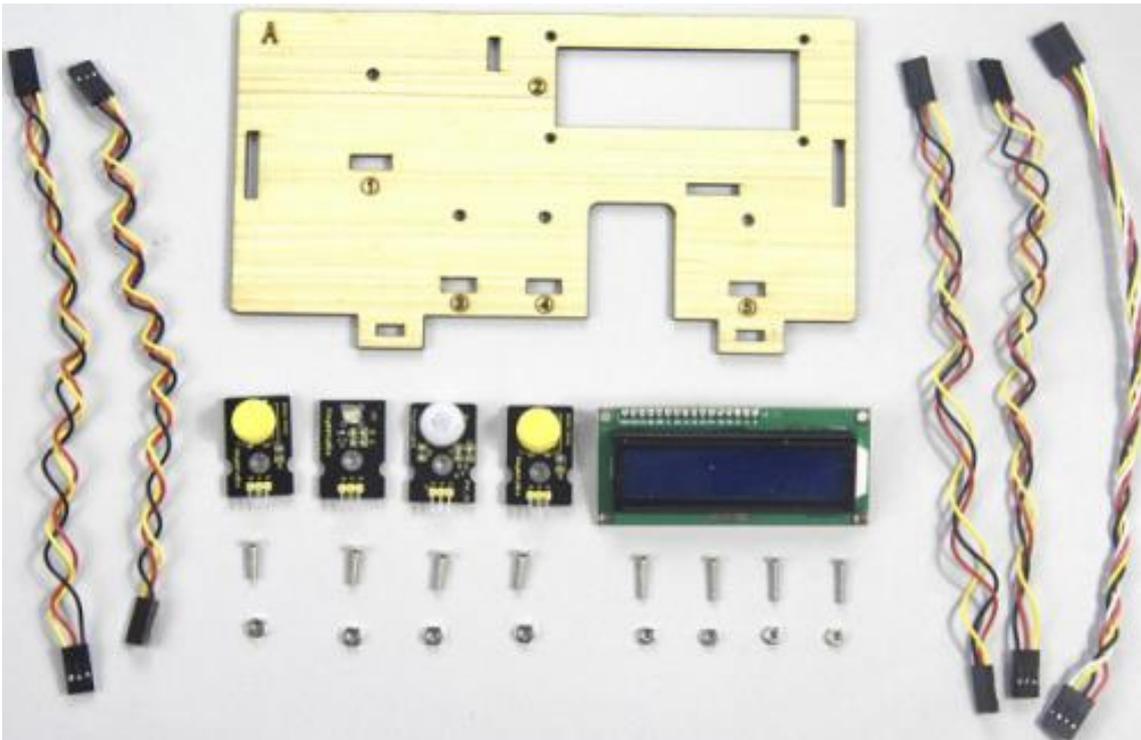


Paso 1: Instalar sensores en la placa A.

Prepare la placa * 1, tornillo redondo M3 * 100MM *4, tuerca níquelada M3 *4, tornillo redondeo M2.5MM * 4, sensor de botón * 2, LED blanco *1, sensor de movimiento PIR * 1, pantalla LCD1602 * 1, Línea dupont FF de 4 pines * 1, Línea dupont FF d e3 pines * 4.

A Board*1	Button module*2	White LED*1	PIR motion sensor*1	LCD1602 Display*1	4pin F-F Dupont line*1
M2.5 Nickel plated nut*4	M3 Nickel plated nut*4	M2.5*10MM Round head screw*4	M3*10MM Round head screw*4	3pin F-F Dupont line*4	

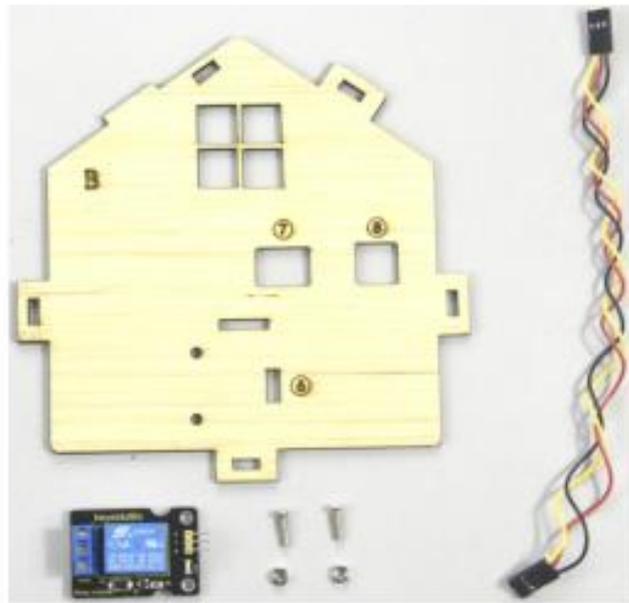
- A. Fije el LED blanco, los sensores de 2 botones y el sensor de movimiento PIR en el área correspondiente de la placa A con los tornillos de cabeza redonda M3*10MM y 4 tuercas M3.
- B. A continuación, instale la pantalla LCD1602 en la placa con 4 tornillos de cabeza redonda M2.5*10MM y tuercas M2.5.
- C. Conéctelos con líneas dupont de 3 pines y 4 pines.



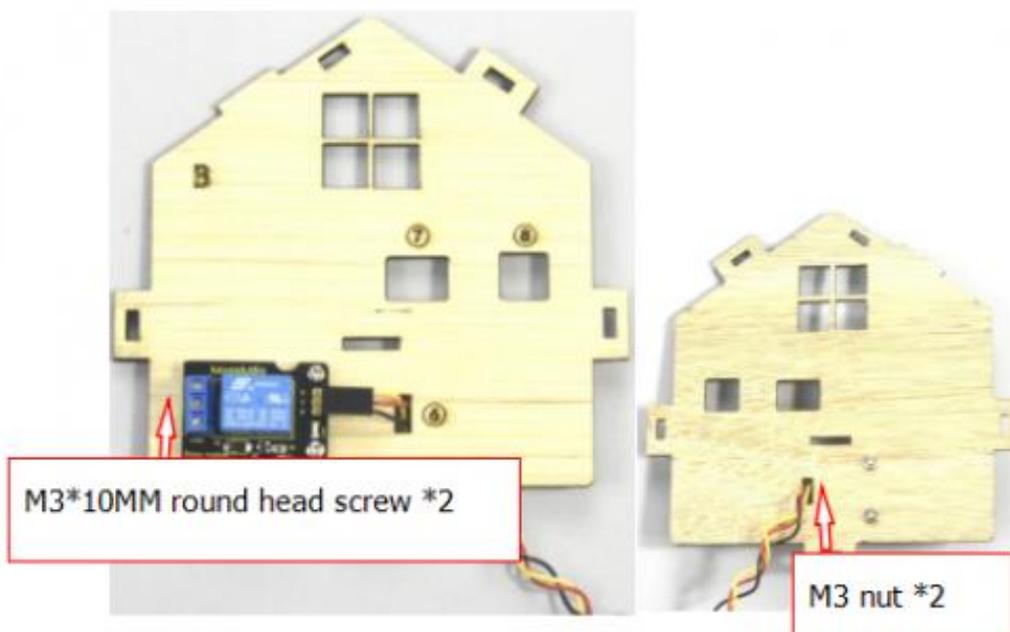
Paso 2: Instale el sensor de la placa b

Prepare una placa B, una línea FF Dupont de 3 pines, 2 tornillos de cabeza redonda M3 * 10MM, 2 tuercas niqueladas M3 y un módulo de relé.

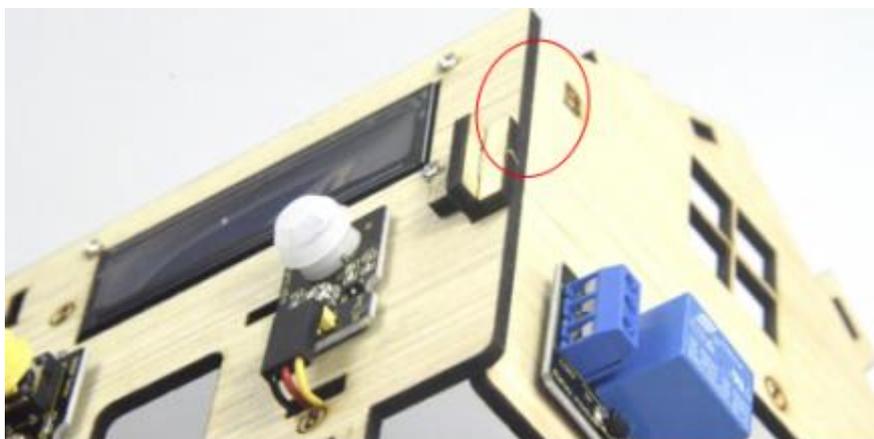
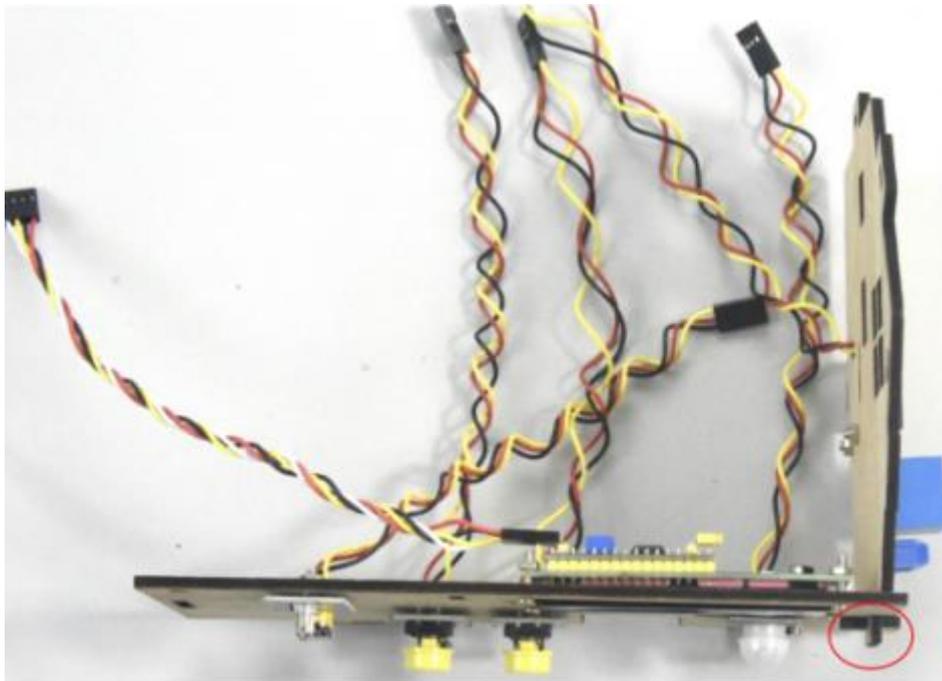
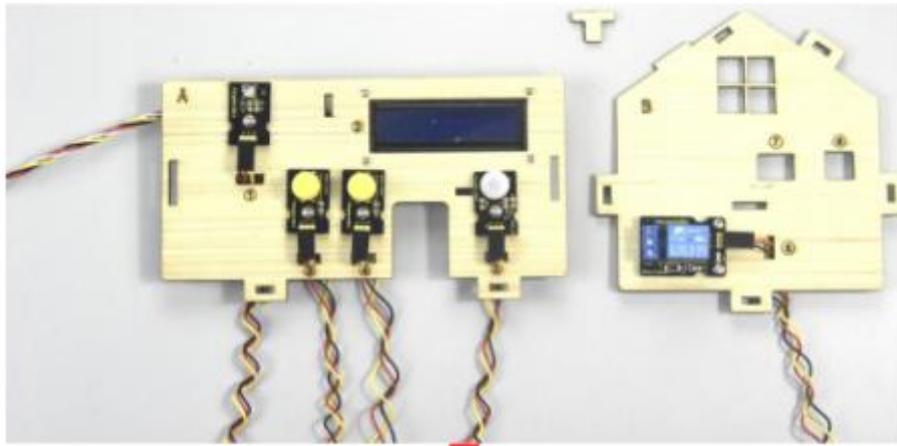
B Board*1	Relay module*1	M3 Nickel plated nut*2	M3*10MM Round head screw*2	3pin F-F Dupont line*1
				



Ensamble el módulo de relé en la placa B con 2 tornillos M3 * 10MM y 2 tuercas M3, conéctelos con una línea dupont de 3 pines.

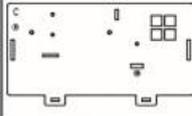


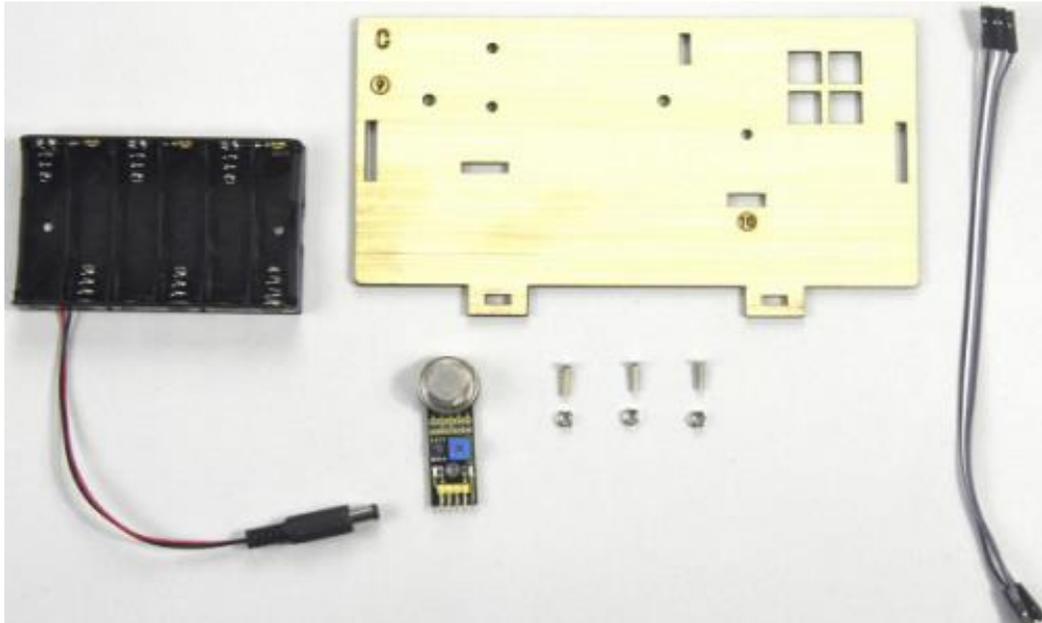
Paso 3: Fije las tablas A y B juntas con el perno en "T".



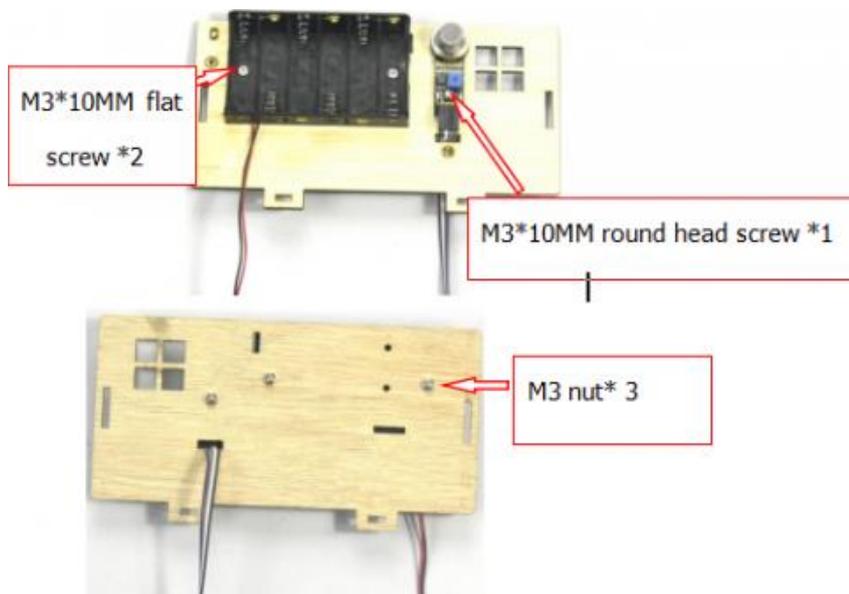
Paso 4: ensamble los sensores y el soporte de la batería de la placa c

Prepare una placa C, sensor de gas MQ-2, soporte de batería, 2 tornillos de cabeza plana M3 * 10MM, un tornillo de cabeza redonda M3 * 100MM, 3 tuercas niqueladas M3 y 4 líneas Dupont FF.

C Board*1	MQ-2 Gas sensor*1	Battery holder*1	M3*10MM Flat head screw*2	M3*10MM Round head screw*1	M3 Nickel plated nut*3	F-F Dupont line*4
						



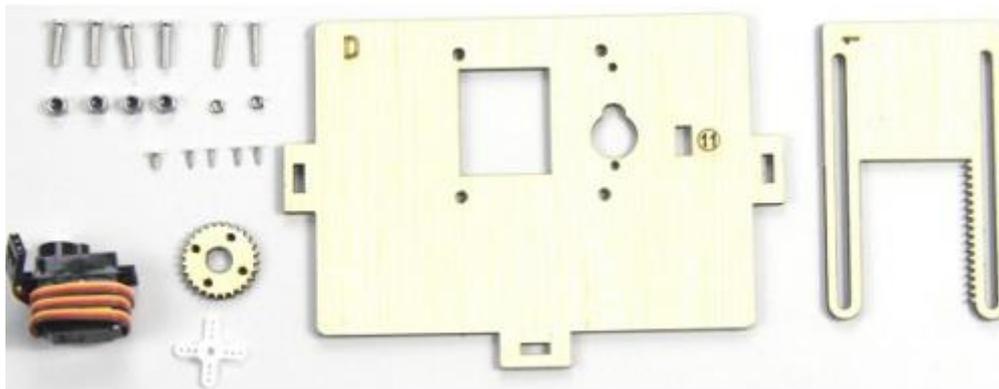
- A. Fije el soporte de la batería en la placa C con 2 tornillos de cabeza plana M3*10MM y 2 tuercas M3.
- B. Luego instale el sensor de gas MQ-2 en el área correspondiente de la placa C con un tornillo de cabeza redonda M3*100MM y una tuerca de M3.
- C. Conéctelos con 4 líneas dupont hembra a hembra.



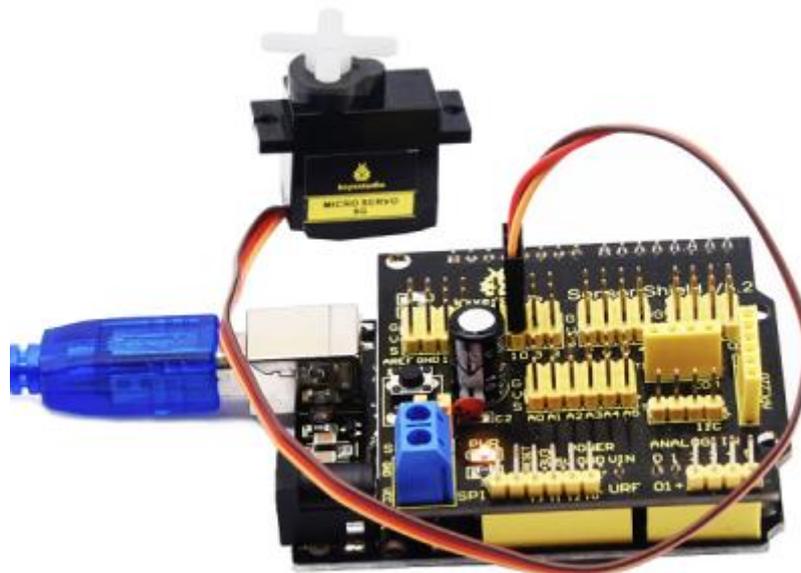
Paso 5: Instale los sensores y las partes de la placa D

Prepare un servo, 4 tornillos autorroscantes M12 * 5, un soporte cruzado blanco (incluido en el servo), un tornillo de cabeza redonda M2 * 5 (incluido en el servo), 2 tornillos de cabeza redonda M2 * 12MM, 2 tornillos M2 niquelados tuerca, 4 tornillos de cabeza redonda M3 * 12MM, 4 tuercas autoblocantes de acero inoxidable M3, una placa D, un engranaje, una placa 1.

D Board*1	Board 1*1	Gear*1	Servo motor*1	White cross mount*1	M2*5 Round head screw*1
M2 Nickel plated nut*2	M3 Stainless self-locking nut*4	M3*12MM Round head screw*4	M2*12MM Round head screw*2	M1.2*5 Self-tapping screw*4	



Gire el servo a 90° antes de instalarlo, conecte el servo a la placa de control keystudio PLUS; cargue el código de prueba en el tablero de control y haga que el servo gire 90°.



Servo Motor	
Cable marrón	GND
Cable rojo	5V
Cable naranja	S (10)

Código de prueba:

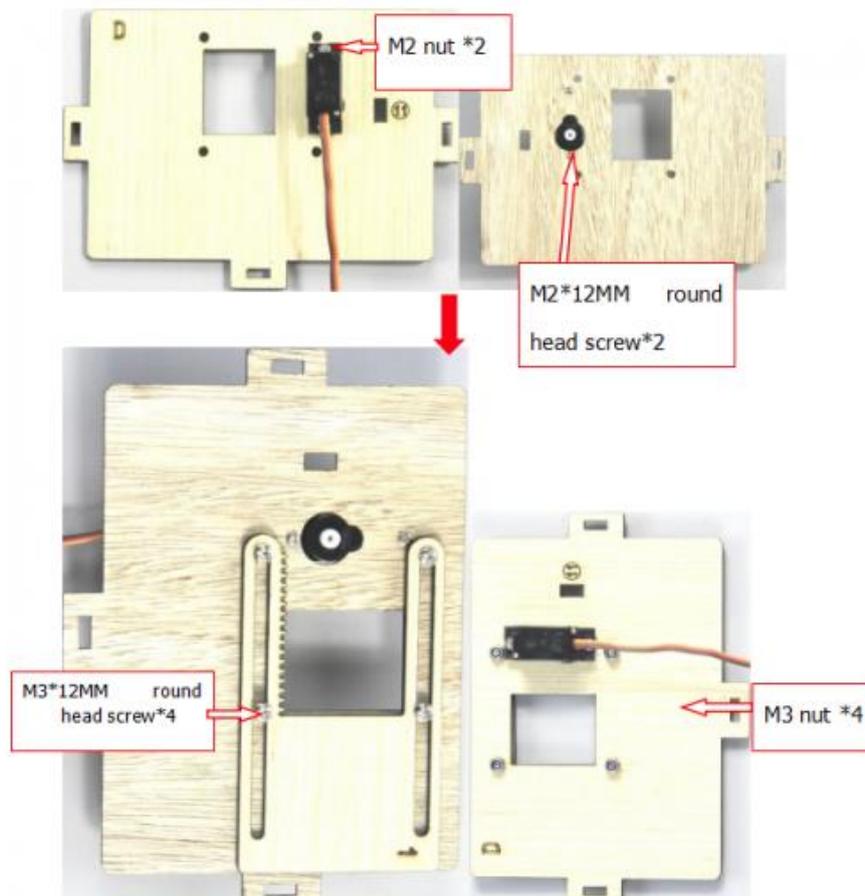
```
#include <Servo.h>
Servo servo_10;

void setup(){
  servo_10.attach(10);
}
void loop(){
  servo_10.write(90);
  delay(500);}

```

Cargue el código de prueba con éxito, el servo gira a 90°.

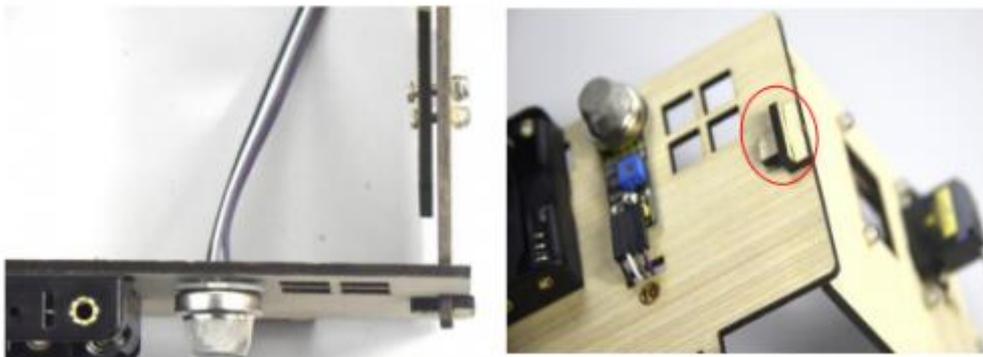
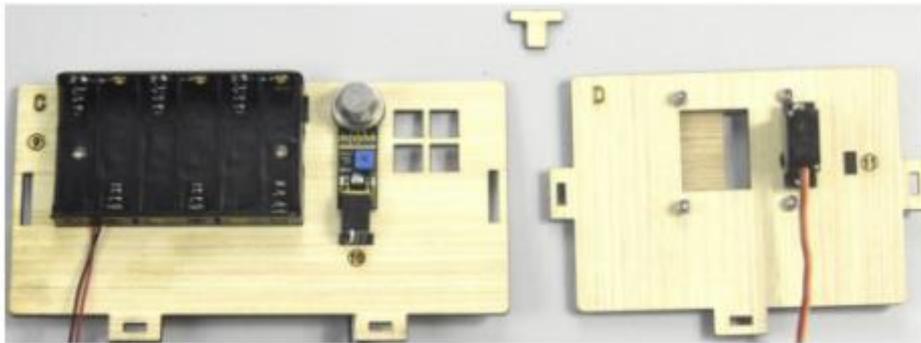
- A. Fije el servo en el área correspondiente de la placa D dos tornillos de cabeza redonda M2 * 12MM y 2 tuercas M2.
- B. Luego instale la placa cuadrada 1 en la placa D con 4 tornillos de cabeza redonda M3 * 12MM y 4 tuercas autoblocantes M3.



Fije el soporte cruzado blanco con el engranaje con 4 tornillos autorroscables M1.2 * 5MM y monte el engranaje en el servomotor con 1 tornillo de cabeza redonda M2 * 5MM.

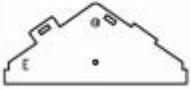


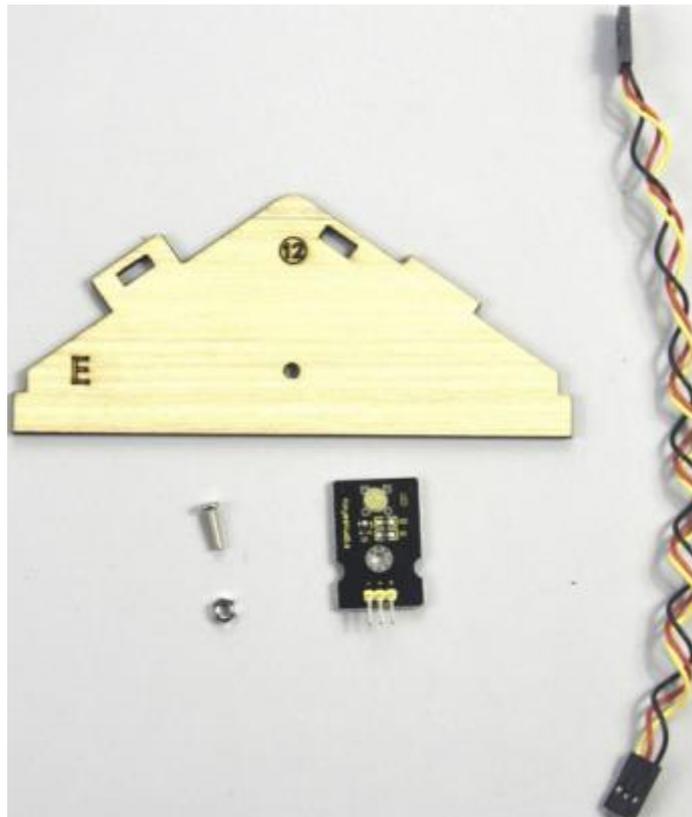
Paso 6: Ensamblaje la placa C con la placa D mediante un perno de tipo "T".



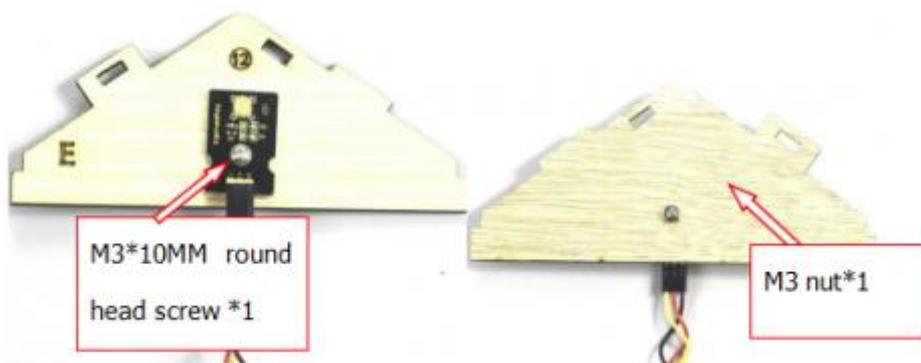
Paso 7: Instale el sensor de la placa E

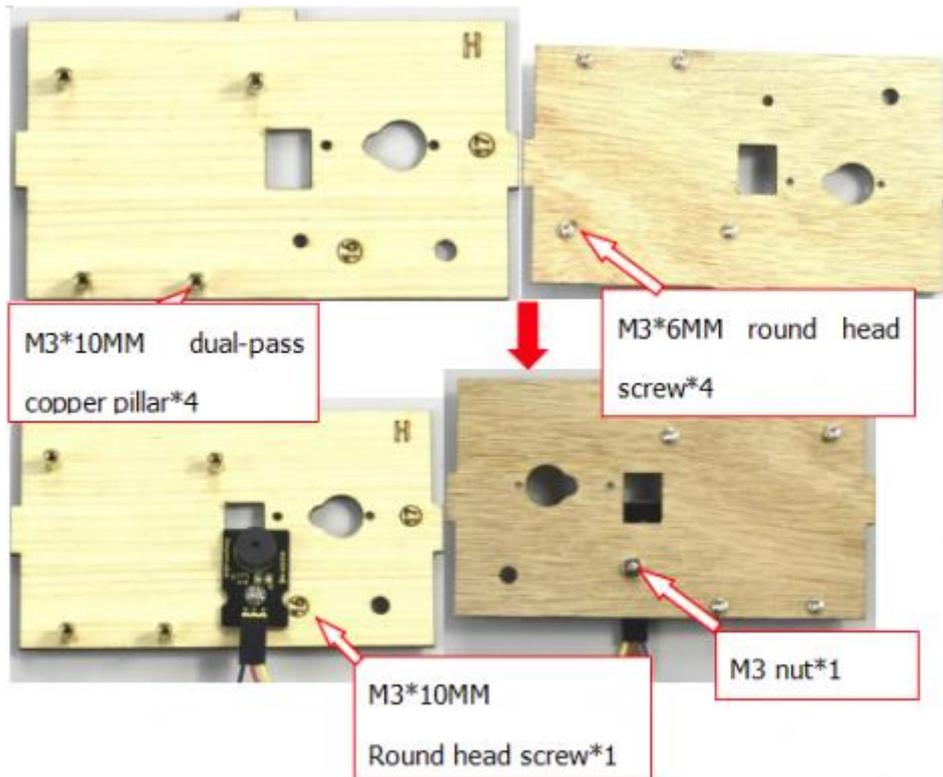
Prepare un módulo LED amarillo, una placa E, un tornillo de cabeza redonda M3 * 10MM, una tuerca niquelada M3 y una línea FF Dupont de 3 pines.

E Board*1	Yellow LED*1	M3 Nickel plated nut*1	M3*10MM Round head screw*1	3pin F-F Dupont line*1
				

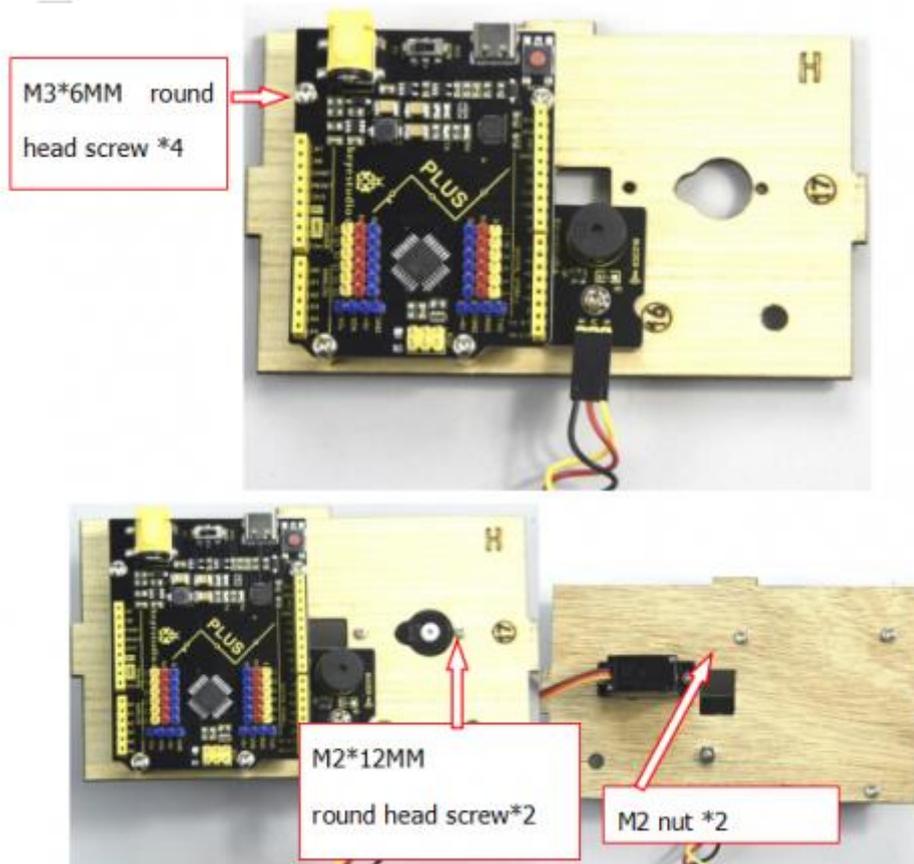


Monte el LED amarillo en el área correspondiente de la placa E con 1 tornillo de cabeza redonda M3 * 10MM y 1 tuerca niquelada M3, luego conéctelo con una línea dupont de 3 pines.

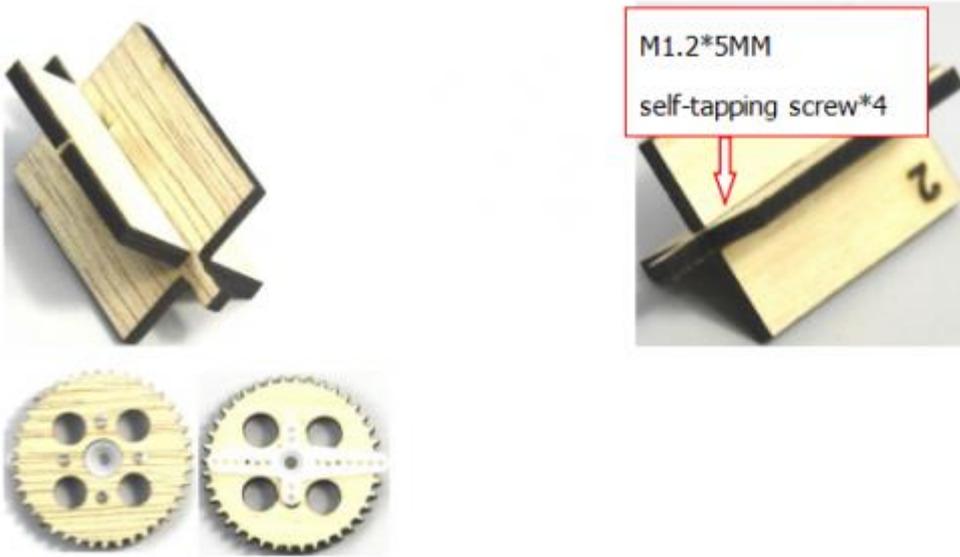




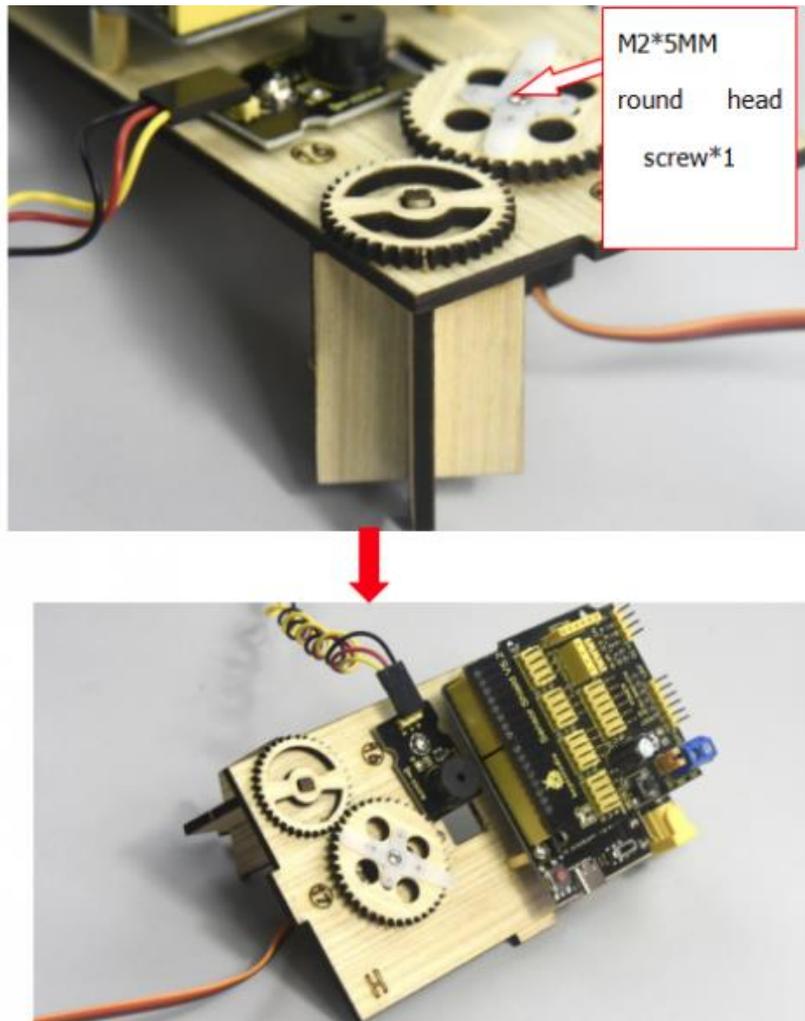
Gire el servo 90° antes de instalarlo, el método es el mismo que en el paso 6. Fije los pilares de cobre M3 * 10MM en la placa de control Keystudio PLUS con 4 tornillos de cabeza redonda M3 * 6MM, luego fije el servo en el área correspondiente de Placa H con 2 tornillos de cabeza redonda M2 * 12MM y 2 tuercas M2.



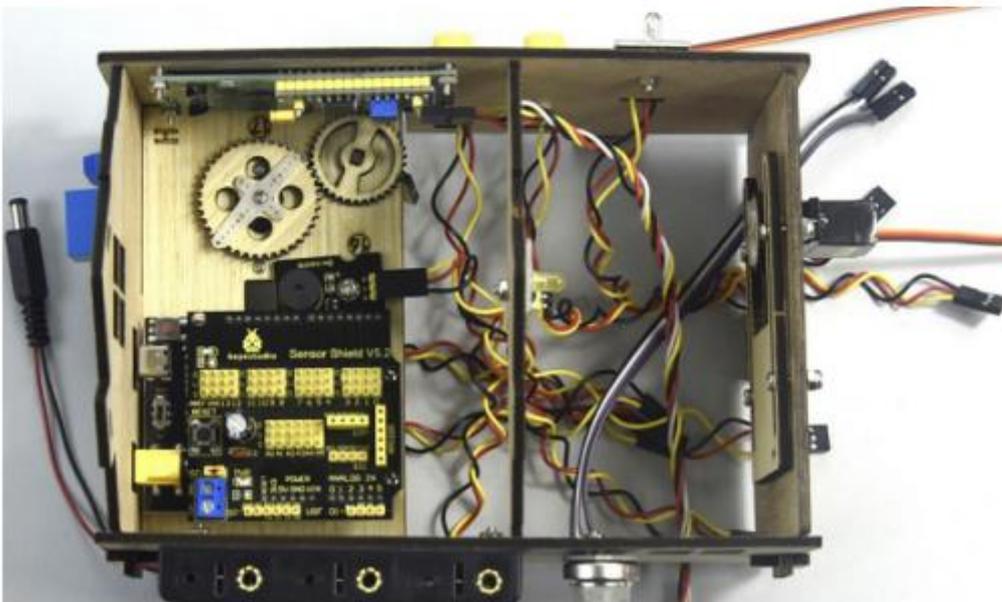
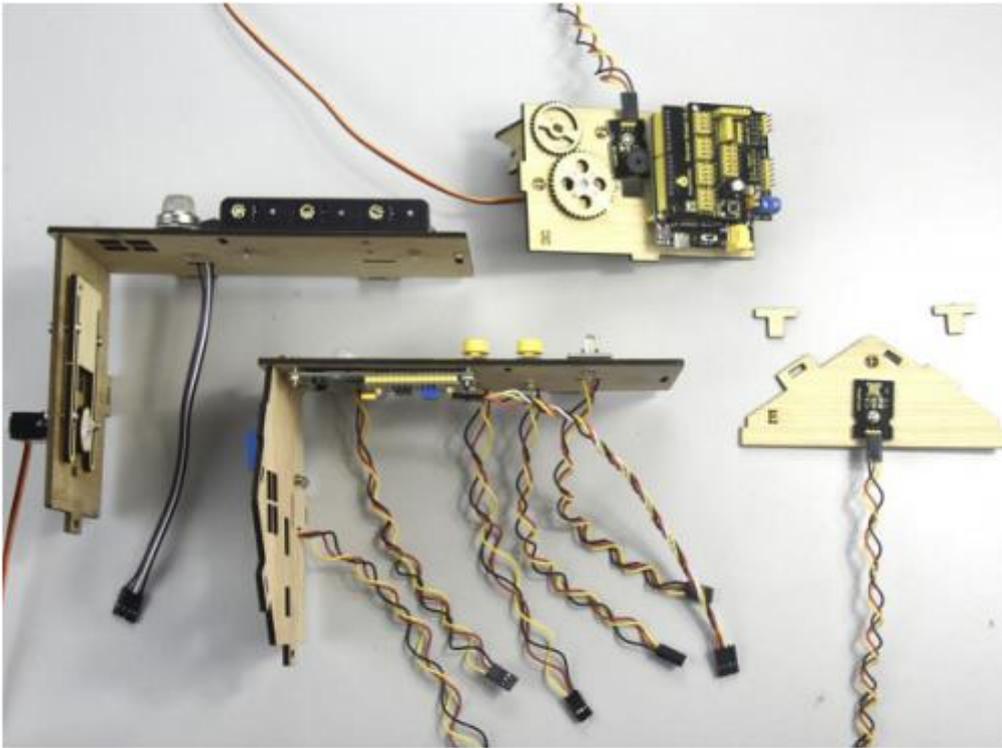
Ensamble 2 piezas de la placa 2 juntas, luego fije el soporte cruzado blanco en el engranaje con 4 piezas de M1.2 * 5 tornillos autorroscantes.

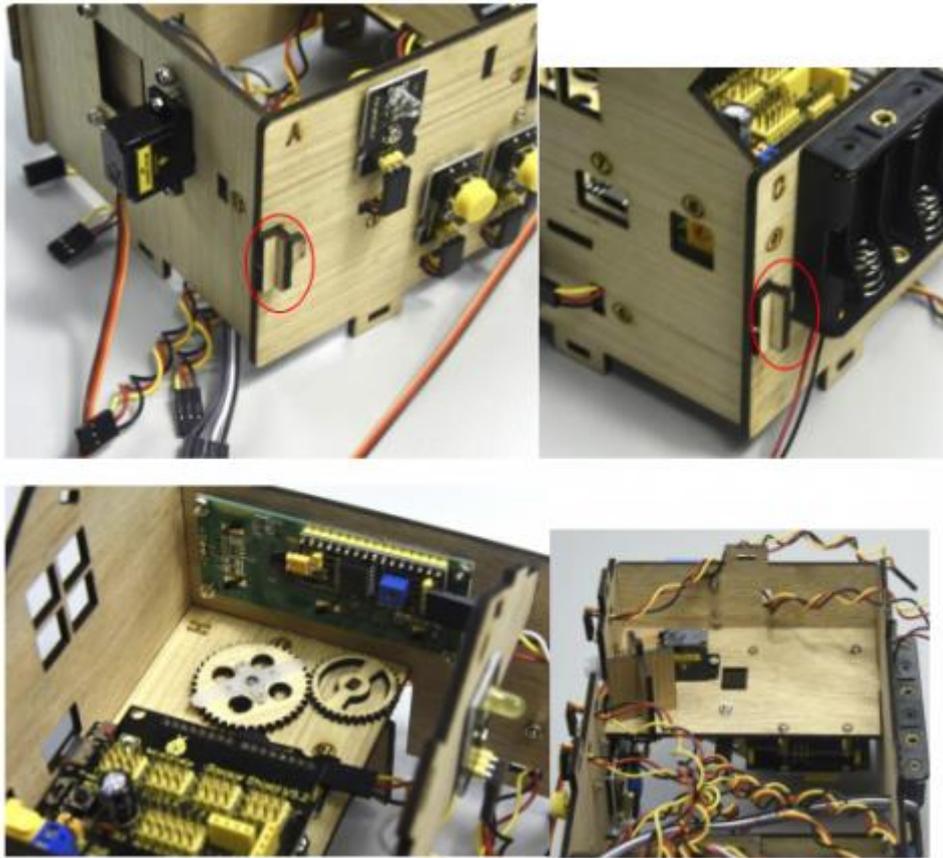


Fije el engranaje con el montaje cruzado blanco en el servo negro con un tornillo M2 * 5MM (incluido en el servo), luego instale la combinación de 2 piezas de la placa 2 y otro en el área correspondiente a la placa H, finalmente apile el escudo del sensor en el Keystudio tablero de control PLUS.



Paso 9: Ensamble las tablas A, B, C, D, E y H juntas, luego fíjelas con 2 pernos tipo "T"
(Nota: la interfaz de alimentación de la placa de control PLUS está alineada con el orificio (8) en la placa B, y la interfaz del cable USB está alineada con el orificio (7) en la placa B)

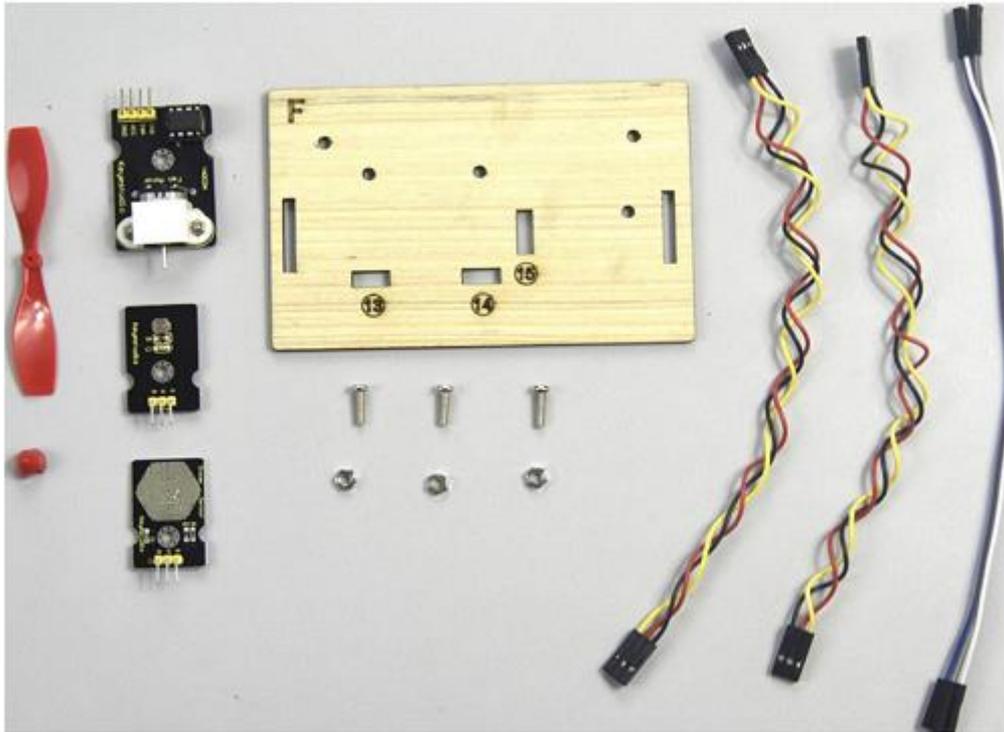




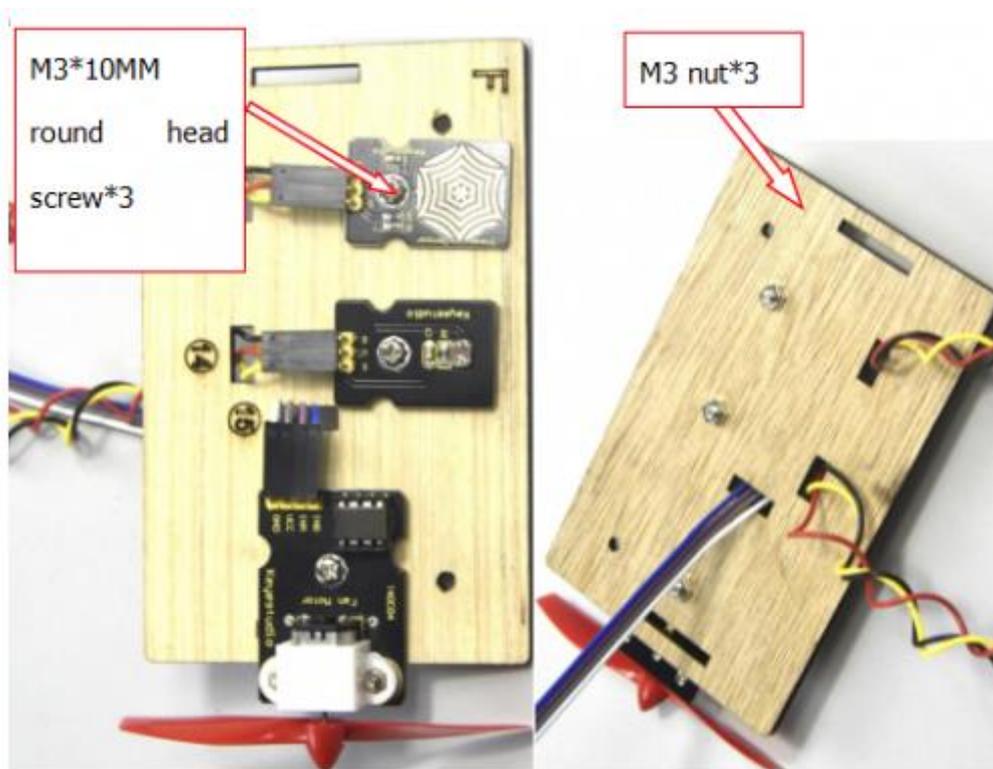
Paso 10: Instale los sensores de la placa F

Prepare un sensor de vapor, un sensor de fotocélula, un módulo de ventilador (con ventilador), una placa F, 2 líneas Dupont de 3 pines F-F, 4 líneas Dupont FF, 3 tornillos de cabeza redonda M3 * 10MM y 3 turecas niqueladas M3.

F board*1	Steam sensor*1	Photocell sensor*1	Fan module*1	M3*10MM Round head screw*3	M3 Nickel plated nut*3	F-F Dupont line*4	3pin F-F Dupont line*2



Fije por separado el sensor de vapor, el sensor de la fotocélula y el módulo del ventilador en la placa F con 3 tornillos de cabeza redonda M3 * 10MM y 3 tuercas M3, luego conéctelo con líneas dupont de 3 pines y 4 pines.

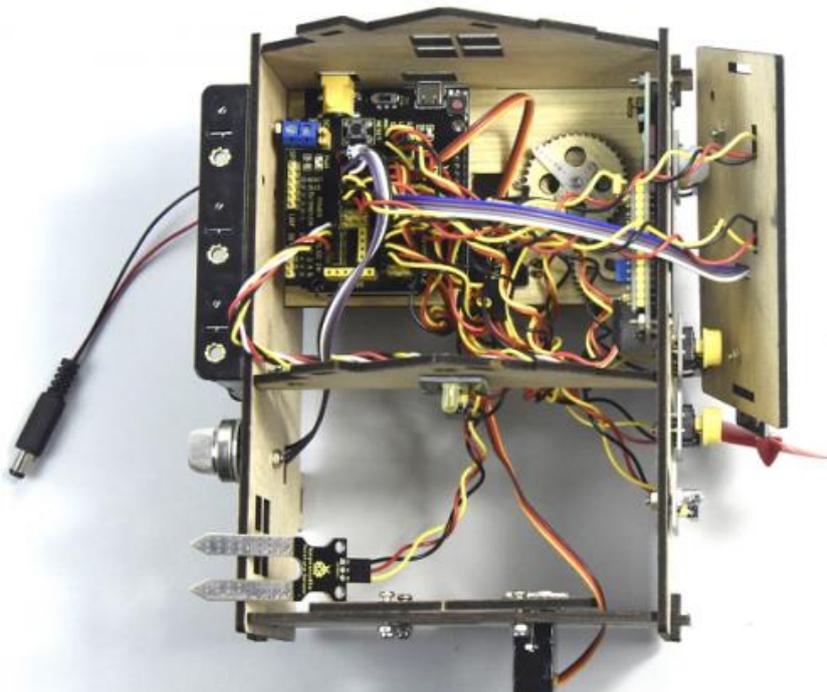


Paso 11: Conecte el sensor/módulo

Conecte un extremo de la línea dupont de 3 pines al pin del sensor de humedad del suelo, luego conecte todos los sensores al protector del sensor. (arregle 2 servos y haga que el cable dupont pase por los agujeros del tablero).

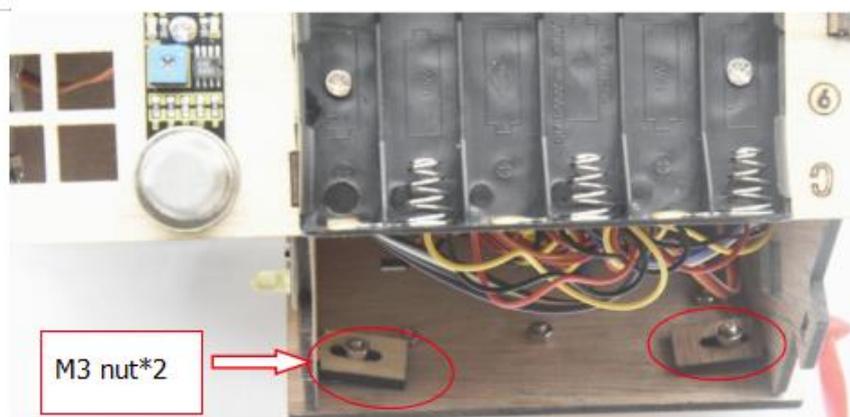
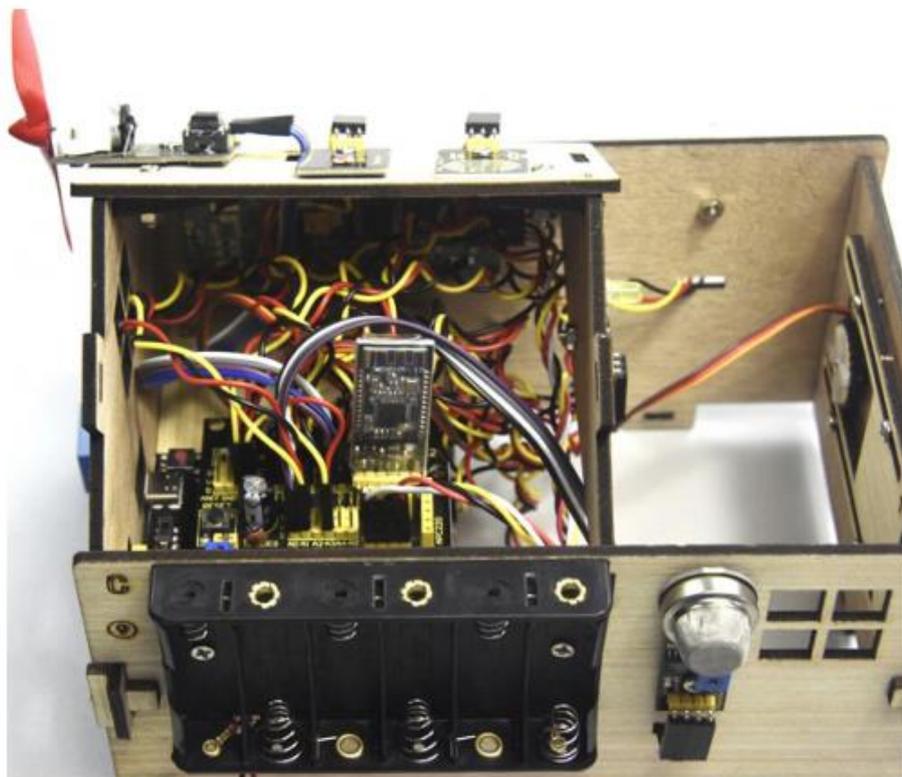


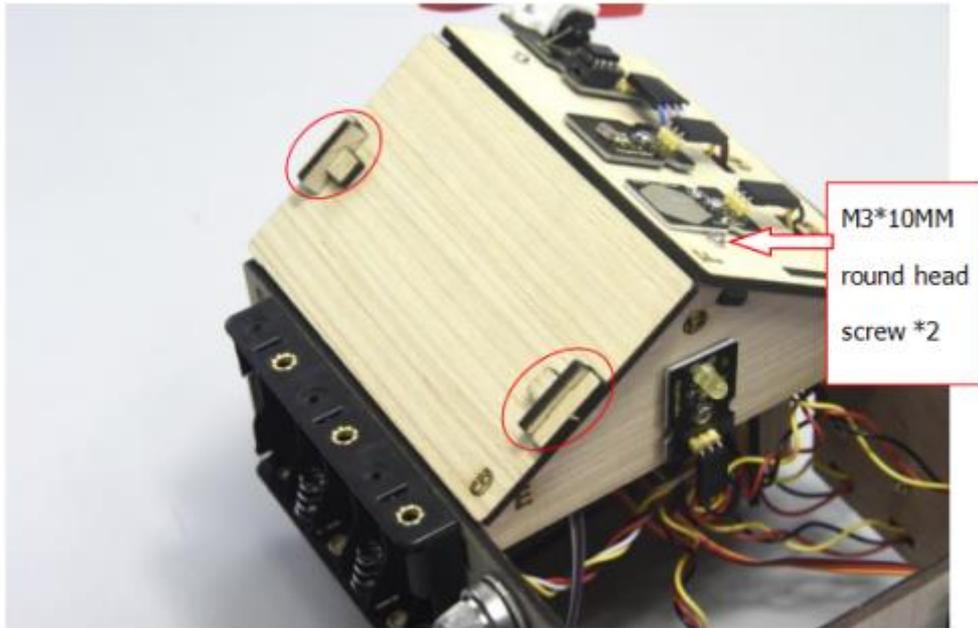
Name	The corresponding interfaces of sensors and sensor shield		The corresponding installed area on the board
PIR Motion Sensor	G/V/S	G/V/2	⑤
Passive buzzer	G/V/S	G/V/3	⑥
Button module 1	G/V/S	G/V/4	③
Yellow LED	G/V/S	G/V/5	⑫
Fan module	GND/VCC/INA/INB	G/V/7/6	⑤
Button module 2	G/V/S	G/V/8	④
Servo 1 controlling the door	Brown/Red/Orange wire	G/V/9	⑰
Servo 2 controlling the windows	Brown/Red/Orange wire	G/V/10	①
MQ-2 Gas Sensor	GND/VCC/D0/A0	G/V/11/A0	⑩
Relay Module	G/V/S	G/V/12	⑥
White LED	G/V/S	G/V/13	①
LCD1602 Display	GND/VCC/SDA/SCL	GND/5V/SDA/SCL	②
Photocell Sensor	G/V/S	G/V/A1	⑭
Soil humidity sensor	G/V/S	G/V/A2	
Steam sensor	G/V/S	G/V/A3	⑬



Inserte el módulo Bluetooth en el protector del sensor, luego fije la placa F con 2 tornillos de cabeza redonda M3 * 10MM, 2 tuercas M3 y 2 piezas con orificios en el medio, monte bien la placa G con 2 pernos tipo "T".

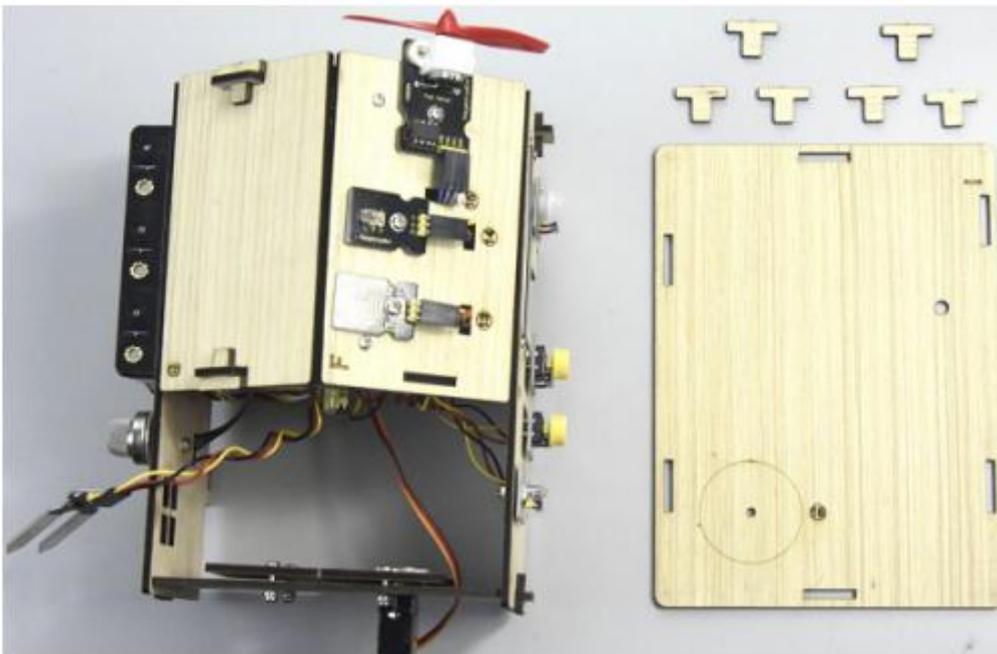
Bluetooth Module	Sensor Expansion Board
VCC	5V
GND	GND
TXD	RXD
RXD	TXD

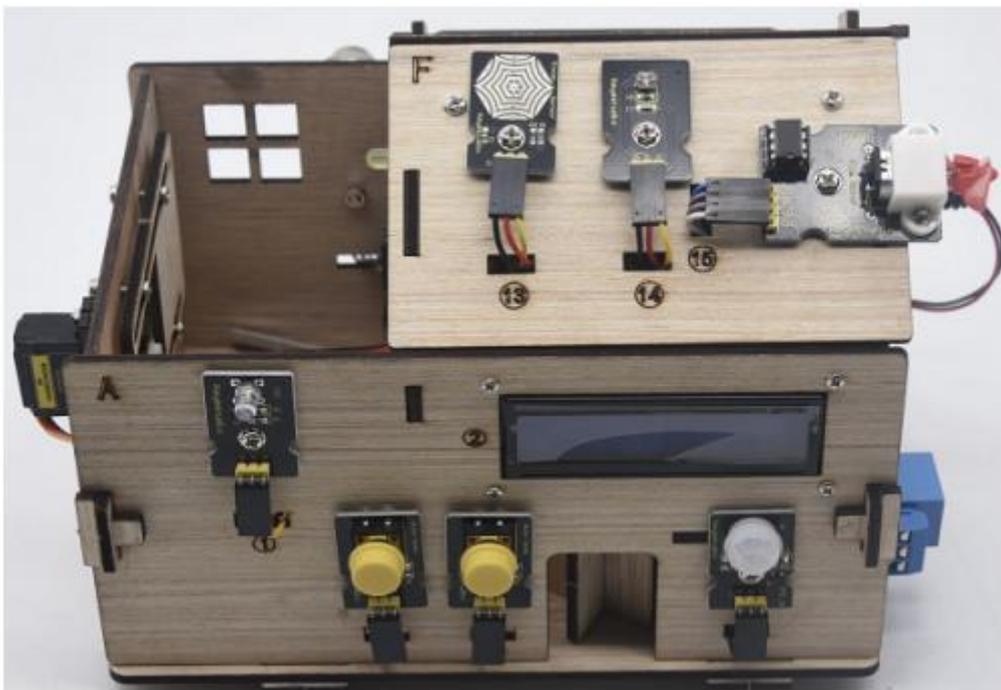
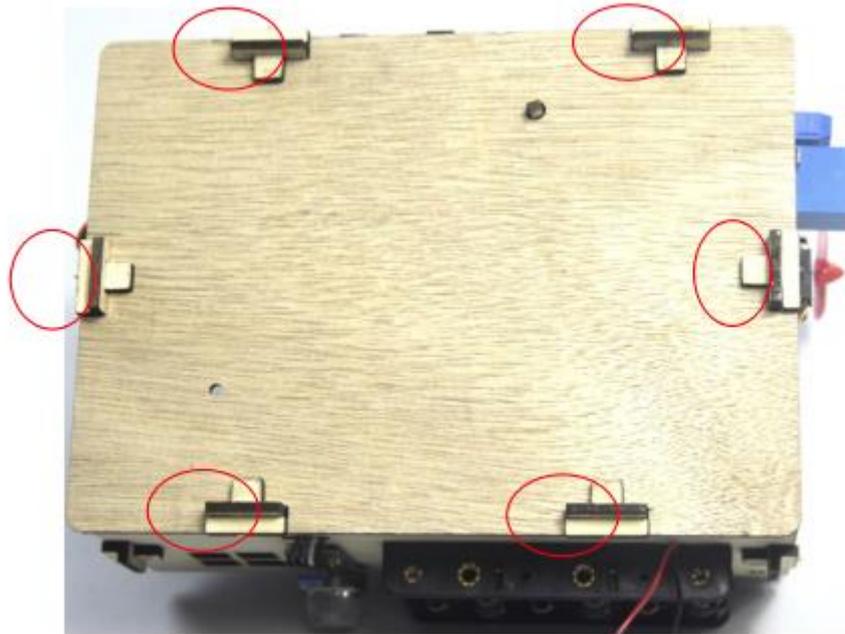




Paso 12: armar el kit

Fijar el tablero I con 6 pernos tipo "T".





Proyecto final: kit de hogar inteligente multiusos

Descripción:

En los proyectos anteriores, presentamos cómo usar sensores, módulos y el módulo Bluetooth HM-10, Para esta lección, realizaremos todas las funciones. Lograremos el efecto de la siguiente manera:

Sensor de fotocélula, sensor de movimiento PIR y LED. Cuando por la noche pasa alguien, el LED está encendido, no hay nadie alrededor, el LED está apagado.



Hay una pantalla LCD 1602, 2 botones, 1 servo en el tablero. Presione el botón 1 para ingresar la contraseña (puede configurar la contraseña en un código de prueba), el 1502LCD mostrará “*”, luego presione el botón 2 para “asegurarse”. Si la contraseña es correcta, el 1602LCD mostrará “abierto”, la puerta estará abierta. Sin embargo, si la contraseña es incorrecta, aparece el “error”, después de 2 segundos, el “error” se convertirá en “nuevamente”, puede ingresar la contraseña nuevamente.

La puerta se cerrará cuando el sensor de movimiento PIR no detecte personas alrededor. Además, presione y mantenga presionado el botón 2, sonará el zumbador, la pantalla LCD muestra “espera”. (Si la contraseña es correcta, el servo girará a 180°, de lo contrario es servo no girará).

Nota: la contraseña correcta es “. - - . - .” lo que significa que presione brevemente el botón 1, presione prolongadamente el botón 1, presione prolongadamente el botón 1, presione brevemente el botón 1, presione prolongadamente el botón 1, presione brevemente el botón 1.

“-“ Significa pulsación larga del botón 1, “.“ Significa pulsación corta del botón 1.

Inserte la humedad del suelo en la maceta, cuando el suelo esté demasiado seco, en timbre sonará y recibirá la notificación en la aplicación.



Cuando el sensor de gas detecta el gas con alta concentración, el zumbador emite un sonido de alarma “tick, tick”.



Cuando el sensor de vapor detecta lluvias, el servo 2 se activará, la ventana se cerrará automáticamente, de lo contrario, la ventana se abrirá.



Equipo:

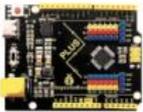
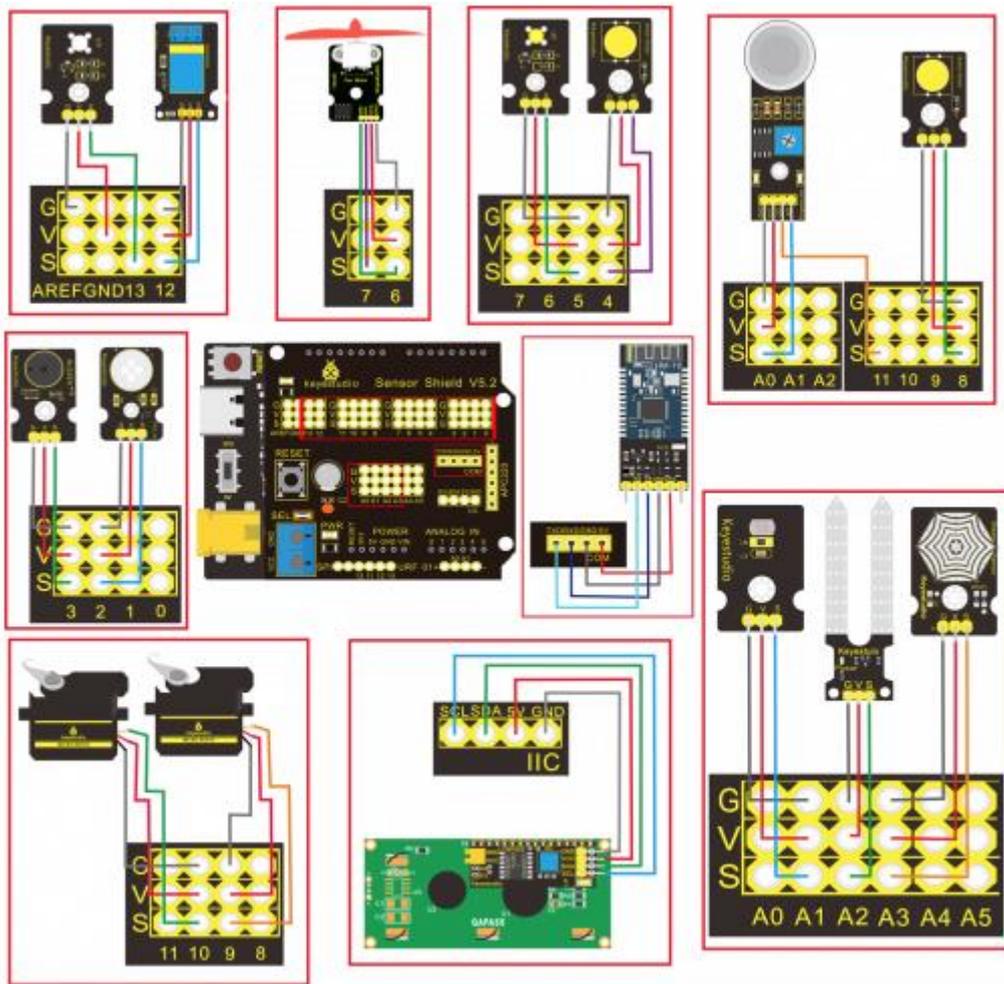
Keyestudio PLUS control board	Sensor shield	Fan module*1	Servo motor*2	LCD1602 display*1	Button sensor*2	White LED*1
						
Relay module*1	Passive buzzer*1	PIR motion sensor*1	Steam sensor*1	photo cell sensor*1	Bluetooth module*1	Yellow LED*1
						
Soil humidity sensor*1	MQ-2 Gas sensor*1	4pin F-F Dupont line*1	F-F Dupont lines	USB cable*1	3pin F-F Dupont line*10	
						

Diagrama de conexión:



Código de prueba:

```
//call the relevant library file
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//Set the communication address of I2C to 0x27, display 16
characters every line, two lines in total
LiquidCrystal_I2C mylcd(0x27, 16, 2);

//set ports of two servos to digital 9 and 10
Servo servo_10;
Servo servo_9;

volatile int btn1_num;//set variable btn1_num
volatile int btn2_num;//set variable btn2_num
volatile int button1;//set variable button1
volatile int button2;//set variable button2
String fans_char;//string type variable fans_char
volatile int fans_val;//set variable fans_char
volatile int flag;//set variable flag
volatile int flag2;//set variable flag2
volatile int flag3;//set variable flag3
volatile int gas;//set variable gas
volatile int infrar;//set variable infrar
String led2;//string type variable led2
volatile int light;//set variable light
String pass;//string type variable pass
String passwd;//string type variable passwd

String servo1;//string type variable servo1
volatile int servo1_angle;//set variable light
String servo2;//string type variable servo2
volatile int servo2_angle;//set variable servo2_angle

volatile int soil;//set variable soil
volatile int val;//set variable val
volatile int value_led2;//set variable value_led2
volatile int water;//set variable water

int length;
int tonepin = 3; //set the signal end of passive buzzer to digital
3
//define name of every sound frequency
#define D0 -1
#define D1 262
#define D2 293
#define D3 329
#define D4 349
#define D5 392
#define D6 440
#define D7 494
#define M1 523
#define M2 586
#define M3 658
#define M4 697
#define M5 783
```

```

#define M6 879
#define M7 987
#define H1 1045
#define H2 1171
#define H3 1316
#define H4 1393
#define H5 1563
#define H6 1755
#define H7 1971

#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.25
#define SIXTEENTH 0.625

//set sound play frequency
int tune[] =
{
    M3, M3, M4, M5,
    M5, M4, M3, M2,
    M1, M1, M2, M3,
    M3, M2, M2,
    M3, M3, M4, M5,
    M5, M4, M3, M2,
    M1, M1, M2, M3,
    M2, M1, M1,
    M2, M2, M3, M1,
    M2, M3, M4, M3, M1,
    M2, M3, M4, M3, M2,
    M1, M2, D5, D0,
    M3, M3, M4, M5,
    M5, M4, M3, M4, M2,
    M1, M1, M2, M3,
    M2, M1, M1
};

//set music beat
float durt[] =
{
    1, 1, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1 + 0.5, 0.5, 1 + 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1 + 0.5, 0.5, 1 + 1,
    1, 1, 1, 1,
    1, 0.5, 0.5, 1, 1,
    1, 0.5, 0.5, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 1,
    1, 1, 1, 0.5, 0.5,
    1, 1, 1, 1,
    1 + 0.5, 0.5, 1 + 1,
};

```

```

void setup() {
  Serial.begin(9600); //set baud rate to 9600

  mylcd.init();
  mylcd.backlight(); //initialize LCD
  //LCD shows "passcord:" at first row and column
  mylcd.setCursor(1 - 1, 1 - 1);
  mylcd.print("passcord:");

  servo_9.attach(9); //make servo connect to digital 9
  servo_10.attach(10); //make servo connect to digital 10
  servo_9.write(0); //set servo connected digital 9 to 0°
  servo_10.write(0); //set servo connected digital 10 to 0°
  delay(300);

  pinMode(7, OUTPUT); //set digital 7 to output
  pinMode(6, OUTPUT); //set digital 6 to output
  digitalWrite(7, HIGH); //set digital 7 to high level
  digitalWrite(6, HIGH); //set digital 6 to high level

  pinMode(4, INPUT); //set digital 4 to input
  pinMode(8, INPUT); //set digital 8 to input
  pinMode(2, INPUT); //set digital 2 to input
  pinMode(3, OUTPUT); //set digital 3 to output
  pinMode(A0, INPUT); //set A0 to input
  pinMode(A1, INPUT); //set A1 to input
  pinMode(13, OUTPUT); //set digital 13 to output
  pinMode(A3, INPUT); //set A3 to input
  pinMode(A2, INPUT); //set A2 to input

  pinMode(12, OUTPUT); //set digital 12 to output
  pinMode(5, OUTPUT); //set digital 5 to output
  pinMode(3, OUTPUT); //set digital 3 to output
  length = sizeof(tune) / sizeof(tune[0]); //set the value of
length
}

void loop() {
  auto_sensor();
  if (Serial.available() > 0) //serial reads the characters
  {
    val = Serial.read(); //set val to character read by serial
    Serial.println(val); //output val character in new lines
    pwm_control();
  }
  switch (val) {
    case 'a': //if val is character 'a', program will circulate
      digitalWrite(13, HIGH); //set digital 13 to high level, LED
      lights up
      break; //exit loop
    case 'b': //if val is character 'b', program will circulate
      digitalWrite(13, LOW); //Set digital 13 to low level, LED is
off
      break; //exit loop
    case 'c': //if val is character 'c', program will circulate
      digitalWrite(12, HIGH); //set digital 12 to high level, NO of
relay is connected to COM

```

```

        break;//exit loop
    case 'd'://if val is character 'd', program will circulate
        digitalWrite(12, LOW); //set digital 12 to low level, NO of
        relay is disconnected to COM

        break;//exit loop
    case 'e'://if val is character 'e', program will circulate
        music1();//play birthday song
        break;//exit loop
    case 'f'://if val is character 'f', program will circulate
        music2();//play ode to joy song
        break;//exit loop
    case 'g'://if val is character 'g', program will circulate
        noTone(3);//set digital 3 to stop playing music
        break;//exit loop
    case 'h'://if val is character 'h', program will circulate
        Serial.println(light);//output the value of variable light in
new lines
        delay(100);
        break;//exit loop
    case 'i'://if val is character 'i', program will circulate
        Serial.println(gas);//output the value of variable gas in new
lines
        delay(100);
        break;//exit loop
    case 'j'://if val is character 'j', program will circulate
        Serial.println(soil);//output the value of variable soil in
new lines
        delay(100);
        break;//exit loop
    case 'k'://if val is character 'k', program will circulate
        Serial.println(water);//output the value of variable water in
new lines
        delay(100);
        break;//exit loop
    case 'l'://if val is character 'l', program will circulate
        servo_9.write(180);//set servo connected to digital 9 to 180°
        delay(500);
        break;//exit loop
    case 'm'://if val is character 'm', program will circulate
        servo_9.write(0);//set servo connected to digital 9 to 0°
        delay(500);
        break;//exit loop
    case 'n'://if val is character 'n', program will circulate
        servo_10.write(180);//set servo connected to digital 10 to
180°
        delay(500);
        break;//exit loop
    case 'o'://if val is character 'o', program will circulate
        servo_10.write(0);//set servo connected to digital 10 to 0°
        delay(500);
        break;//exit loop
    case 'p'://if val is character 'p', program will circulate
        digitalWrite(5, HIGH); //set digital 5 to high level, LED is
on
        break;//exit loop

```

```

    case 'q'://if val is character 'q', program will circulate
        digitalWrite(5, LOW); // set digital 5 to low level, LED is
off
        break;//exit loop
    case 'r'://if val is character 'r', program will circulate
        digitalWrite(7, LOW);
        digitalWrite(6, HIGH); //fan rotates anticlockwise at the
fastest speed
        break;//exit loop
    case 's'://if val is character 's', program will circulate
        digitalWrite(7, LOW);
        digitalWrite(6, LOW); //fan stops rotating
        break;//exit loop
    }
}

////////////////////set birthday
song////////////////////////////////////
void birthday()
{
    tone(3, 294); //digital 3 outputs 294HZ sound
    delay(250); //delay in 250ms
    tone(3, 440);
    delay(250);
    tone(3, 392);
    delay(250);
    tone(3, 532);
    delay(250);
    tone(3, 494);
    delay(500);
    tone(3, 392);
    delay(250);
    tone(3, 440);
    delay(250);
    tone(3, 392);
    delay(250);
    tone(3, 587);
    delay(250);
    tone(3, 532);
    delay(500);
    tone(3, 392);
    delay(250);
    tone(3, 784);
    delay(250);
    tone(3, 659);
    delay(250);
    tone(3, 532);
    delay(250);
    tone(3, 494);
    delay(250);
    tone(3, 440);
    delay(250);
    tone(3, 698);
    delay(375);
    tone(3, 659);
    delay(250);
    tone(3, 532);
    delay(250);
}

```

```

tone(3, 587);
delay(250);
tone(3, 532);
delay(500);
}

//detect gas
void auto_sensor() {
  gas = analogRead(A0); //assign the analog value of A0 to gas
  if (gas > 700) {
//if variable gas>700
    flag = 1; //set variable flag to 1
    while (flag == 1)
      //if flag is 1, program will circulate
      {
        Serial.println("danger"); //output "danger" in new lines
        tone(3, 440);
        delay(125);
        delay(100);
        noTone(3);
        delay(100);
        tone(3, 440);
        delay(125);
        delay(100);
        noTone(3);
        delay(300);
        gas = analogRead(A0); //gas analog the value of A0 to gas
        if (gas < 100) //if variable gas is less than 100
          {
            flag = 0; //set variable flag to 0
            break; //exit loop exist to loop
          }
      }
  } else
    //otherwise
    {
      noTone(3); // digital 3 stops playing music
    }
  light = analogRead(A1); //Assign the analog value of A1 to light
  if (light < 300) //if variable light is less than 300
    {
      infrar = digitalRead(2); //assign the value of digital 2 to
infrar
      Serial.println(infrar); //output the value of variable infrar in
new lines
      if (infrar == 1)
        // if variable infra is 1
        {
          digitalWrite(13, HIGH); //set digital 13 to high level, LED
is on
        } else //Otherwise
        {
          digitalWrite(13, LOW); //set digital 13 to low level, LED is
off
        }
    }
}

```

```

    }
    water = analogRead(A3); //assign the analog value of A3 to
variable water
    if (water > 800)
        // if variable water is larger than 800
        {
            flag2 = 1; //if variable flag 2 to 1
            while (flag2 == 1)
                // if flag2 is 1, program will circulate
                {
                    Serial.println("rain"); //output "rain" in new lines
                    servo_10.write(180); // set the servo connected to digital 10
to 180°
                    delay(300); //delay in 300ms
                    delay(100);
                    water = analogRead(A3); //assign the analog value of A3 to
variable water
                    if (water < 30) // if variable water is less than 30
                        {
                            flag2 = 0; // set flag2 to 0
                            break; //exit loop
                        }
                }
        }
    } else //Otherwise
    {
        if (val != 'u' && val != 'n')
            //if val is not equivalent 'u' either 'n'
            {
                servo_10.write(0); //set servo connected to digital 10 to 0°
                delay(10);
            }
    }
    }
    soil = analogRead(A2); //assign the analog value of A2 to variable
soil
    if (soil > 50)
        // if variable soil is greater than 50
        {
            flag3 = 1; //set flag3 to 1
            while (flag3 == 1)
                //If set flag3 to 1, program will circulate
                {
                    Serial.println("hydropenia "); //output "hydropenia " in new
lines
                    tone(3, 440);
                    delay(125);
                    delay(100);
                    noTone(3);
                    delay(100);
                    tone(3, 440);
                    delay(125);
                    delay(100);
                    noTone(3); //digital 3 stops playing sound
                    delay(300);
                }
        }
    }
}

```

```

        soil = analogRead(A2); //Assign the analog value of A2 to
variable soil
        if (soil < 10) //If variable soil < 10
        {
            flag3 = 0; //set flag3 to 0
            break; //exit loop
        }
    }

} else //Otherwise
{
    noTone(3); //set digital 3 to stop playing music
}
door(); //run subroutine
}

void door() {
    button1 = digitalRead(4); // assign the value of digital 4 to
button1
    button2 = digitalRead(8); //assign the value of digital 8 to
button2

    if (button1 == 0) //if variable button1 is 0
    {
        delay(10); //delay in 10ms
        while (button1 == 0) //if variable button1 is 0, program will
circulate
        {
            button1 = digitalRead(4); // assign the value of digital 4 to
button1
            btn1_num = btn1_num + 1; //variable btn1_num plus 1
            delay(100); // delay in 100ms
        }

    }
    if (btn1_num >= 1 && btn1_num < 5) //1 ≤ if variable btn1_num < 5
    {
        Serial.print(".");
        Serial.print("");
        passwd = String(passwd) + String("."); //set passwd
        pass = String(pass) + String("."); //set pass
        //LCD shows pass at the first row and column
        mylcd.setCursor(1 - 1, 2 - 1);
        mylcd.print(pass);
    }
    if (btn1_num >= 5)
        //if variable btn1_num ≥ 5
    {
        Serial.print("-");
        passwd = String(passwd) + String("-"); //Set passwd
        pass = String(pass) + String("-"); //set pass
        //LCD shows pass at the first row and column
        mylcd.setCursor(1 - 1, 2 - 1);
        mylcd.print(pass);
    }

}
if (button2 == 0) //if variable button2 is 0

```

```

{
  delay(10);
  if (button2 == 0)//if variablebutton2 is 0
  {
    if (passwd == ".--.-.")//if passwd is ".--.-."
    {
      mylcd.clear();//clear LCD screen
      //LCD shows "open!" at first character on second row
      mylcd.setCursor(1 - 1, 2 - 1);
      mylcd.print("open!");
      servo_9.write(100);//set servo connected to digital 9 to
100°
      delay(300);
      delay(5000);
      passwd = "";
      pass = "";
      mylcd.clear();//clear LCD screen
      //LCD shows "password:"at first character on first row
      mylcd.setCursor(1 - 1, 1 - 1);
      mylcd.print("password:");

    } else //Otherwise
    {
      mylcd.clear();//clear LCD screen
      //LCD shows "error!"at first character on first row
      mylcd.setCursor(1 - 1, 1 - 1);
      mylcd.print("error!");
      passwd = "";
      pass = "";
      delay(2000);
      //LCD shows "again" at first character on first row
      mylcd.setCursor(1 - 1, 1 - 1);
      mylcd.print("again");
    }
  }
}
infrar = digitalRead(2);//assign the value of digital 2 to infrar
if (infrar == 0 && (val != 'l' && val != 't'))
  //if variable infrar is 0 and val is not 'l' either 't'
  {
    servo_9.write(0);//set servo connected to digital 9 to 0°
    delay(50);
  }
if (button2 == 0)//if variablebutton2 is 0
  {
    delay(10);
    while (button2 == 0) //if variablebutton2 is 0, program will
circulate
    {
      button2 = digitalRead(8);//assign the value of digital 8 to
button2
      btn2_num = btn2_num + 1;//variable btn2_num plus 1
      delay(100);
      if (btn2_num >= 15)//if variablebtn2_num ≥15
      {
        tone(3, 532);
        delay(125);
        mylcd.clear();//clear LCD screen

```

```

        //LCD shows "password:" at the first character on first row
        mylcd.setCursor(1 - 1, 1 - 1);
        mylcd.print("password:");
        //LCD shows "wait" at the first character on first row
        mylcd.setCursor(1 - 1, 1 - 1);
        mylcd.print("wait");
    } else//Otherwise
    {
        noTone(3);//digital 3 stops playing music
    }
}

}

btn1_num = 0;//set btn1_num to 0
btn2_num = 0;//set btn2_num to 0
}

// Birthday song
void music1() {
    birthday();
}
//Ode to joy
void music2() {
    Ode_to_Joy();
}
void Ode_to_Joy()//play Ode to joy song
{
    for (int x = 0; x < length; x++)
    {
        tone(tonepin, tune[x]);
        delay(300 * durt[x]);
    }
}

//PWM control
void pwm_control() {
    switch (val)
    {
        case 't'://if val is 't', program will circulate
            servol = Serial.readStringUntil('#');
            servol_angle = String(servol).toInt();
            servo_9.write(servol_angle);//set the angle of servo
connected to digital 9 to servol_angle
            delay(300);
            break;//exit loop
        case 'u'://if val is 'u', program will circulate
            servo2 = Serial.readStringUntil('#');
            servo2_angle = String(servo2).toInt();
            servo_10.write(servo2_angle);//set the angle of servo
connected to digital 10 to servo2_angle
            delay(300);
            break;//exit loop
        case 'v'://if val is 'v', program will circulate
            led2 = Serial.readStringUntil('#');
            value_led2 = String(led2).toInt();
            analogWrite(5, value_led2); //PWM value of digital 5 is
value_led2
            break;//exit loop
    }
}

```

```
case 'w': //if val is 'w', program will circulate
  fans_char = Serial.readStringUntil('#');
  fans_val = String(fans_char).toInt();
  digitalWrite(7, LOW);
  analogWrite(6, fans_val); //set PWM value of digital 6 to
fans_val, the larger the value, the faster the fan
  break; //exit loop
}
}
```

Sube el código y mira el resultado!

Nota: Retire el módulo Bluetooth, por favor, cuando cargue el código de prueba. De lo contrario, el programa no se podrá cargar. Conecte el módulo Bluetooth y Bluetooth para emparejar después de cargar el código de prueba.

Resultado de la prueba:

Cargue el código de prueba, apile la placa de expansión en la placa de control PLUS y enciéndala. Después de emparejar y conectar Bluetooth con éxito, podemos controlar la casa inteligente a través de la aplicación.

Recuerda que todos los recursos los tienes en: <https://fs.keyestudio.com/KS0085>

KS0085 Smart Home Kit for Arduino

desde keyestudio Robot (keyestudio)

Guardar en Dropbox

Descargar

Nombre

1. About keyestudio

2. Tutorial for Arduino

3. APP

4. Tutorial for Mixly

5. Installation guide

Install a driver

Russian Tutorial