



Los periféricos como mouse, teclados, entre otros, cada día son más comunes y la transmisión de datos para estos por medio de puerto USB se ha convertido en un estándar. Para poder conectar periféricos a Arduino se diseñó la board host shield, la cual tiene un socket USB hembra tipo A donde podemos conectar periféricos con puerto USB.

ARDUINO HOST SHIELD

Guía

Práctica: "Control de LED por medio del mouse"

Objetivo de la práctica:

Encender y apagar un led usando los botones de un mouse.

Elementos a usar

- Host USB
- Arduino UNO
- Cable USB
- Mouse USB
- Fuente de alimentación

Se recomienda Usar

- Librería max3421e.h
- Librería usb.h
- Librería Max3421e_constants.h



Hardware y Conexiones

¿Cómo funciona?

La USB host shield usa el integrado Max3421e, el cual funciona como un conversor de SPI a host USB, (el mismo que usa la Arduino ADK). se estuviera conectando a una computadora. datos o autenticaciones, los mismos deben mediante la interface SPI. De esta reconocimiento automático del Arduino sino que, por el contrario, adecuada el uso de cualquier

La Host shield, al igual que muchos otros sistemas open hardware viene sin soldadura en los pines de conexión, esto se debe a que se espera que el usuario decida qué tipo de conector es mejor para su aplicación, recomendamos soldar regletas tipo Arduino para tener accesos a los pines de salida desde la parte superior.

La Host shield para Arduino se comunica con ésta a través de una comunicación SPI, y se usan los siguientes pines en Arduino UNO **D10** – CS, **D11** – MOSI, **D12** – MISO, **D13** – SCK

Conexiones

- 1. Conectar la host shield a la board Arduino Uno
- 2. Conectar una fuente de alimentación por el jack de conexión

3. Conectar el mouse USB al puerto USB que se encuentra en la Host shield.



Se requiere una fuente de alimentación externa debido a que se va a energizar además de la Arduino VNO y la Shield. el mouse. La board tiene un switch de ON/OFF al lado del conector VSB, asegúrese de que esté en posición ON y que el LED de color rojo este encendido.

Cuando se adquiere datos usando la USB host shield, se debe tener en cuenta que la velocidad de transmisión de datos de USB es mayor que la SPI, por lo tanto no se recomienda enviar gran cantidad de datos (USB a SPI) debido a que la board Arduino no va a ser capaz de capturarlos y procesarlos.



Programación

Se requiere descargar la librería de la VSB host shield (<u>click aquí</u>). Luego de descomprimirla, se guarda el archivo en la carpeta de librería de Arduino

4	libraries *	Nombre	Fecha de modifica	Tipo	Tamaño
Þ	BluetoothShieldDemoCode	🕌 examples	06/02/2013 02:48	Carpeta de archivos	
	CMUcam4	, aitianore	12/06/2012 05:34	Archivo GITIGNORE	1 KB
	DS1307RTC	Ch9.h	12/06/2012 05:34	VisualStudio.h.10.0	10 KB
⊳ 🕛	EasyVR	Max LCD	12/06/2012 05:34	Archivo CPP	8 KB
	EEPROM	Max LCD.h	12/06/2012 05:34	VisualStudio.h.10.0	3 KB
P	Esplora	• Max3421e	12/06/2012 05:34	Archivo CPP	10 KB
Þ 🕕	Ethernet	Max3421e.h	12/06/2012 05:34	VisualStudio.h.10.0	2 KB
Þ	Firmata	Max3421e constants.h	12/06/2012 05:34	VisualStudio.h.10.0	7 KB
Þ	Gameduino		12/06/2012 05:34	Archivo	1 KB
	herkulex	Usb	12/06/2012 05:34	Archivo CPP	19 KB
	HerkuleXCtrl	Usb.h	12/06/2012 05:34	VisualStudio.h.10.0	11 KB
⊳ 🕛	LCDcolor				
Þ 🔒	LCDLibversionv1				
Þ	LiquidCrystal				
▷ 🚺	PN532_SPI				
Þ	PString				
Þ	SD				
Þ]]	SdFat				
	SeeedTFTv2				
Þ 🚺	SeeedTouchScreen				
Þ]	Servo				
Þ	SoftwareSerial				
Þ 🚺	SPI				
Þ	Stepper				
Þ	Streaming				
Þ]]	Time				
Þ	TimeAlarms				
Þ]]	USB_Host_Shield				
Þ	WiFi				
Þ	WifiShield				





Tenga en cuenta que cuando conecte un dispositivo USB debe tener una librería apropiada para cada dispositivo, esto debido a que cada uno tiene datos de configuración diferente



Haga <u>Click Aquí</u> para descargar el siguiente ejemplo de programación.



/*www.dynamoelectronics.com Ejemplo de uso Host shield, usando las librerías max3421e.h usb.h, Max3421e_constants.h con Arduino UNO, tenga en cuenta que esta board solo funciona con Arduino UNO

Pines de Arduino

- ** MOSI pin 11
- ** MISO pin 12 ** CLK - pin 13
- ** CLK pin 13
- ** Led pin 6 conexión externa

Este ejemplo muestra en el monitor serial la posición relativa del mouse y el estado de los botones, adicionalmente al presionar el botón 1 se enciende un led conectado al puerto 6 y presionando el botón 2 se apaga

Este ejemplo es de dominio público Creado por Fabio Eraso - Dynamo Electronics Basado en el ejemplo de Oleg Mazurov, Circuits At Home

*/

<pre>#include <max3421e.h> #include <usb.h> #include <usb.h> #include <max3421e_cou #define="" 1="" confvalue="" devaddr="" lab;<="" max3421e="" max;="" pre="" usb=""></max3421e_cou></usb.h></usb.h></max3421e.h></pre>	//incluyendo libreria IC //incluyendo libreria USB nstants.h> //Constantes //Constantes //Constantes	
void setup()	// Condiciones iniciales	
<pre>{ Serial.begin(115200); Serial.println("Start"); Max.powerOn(); delay(200); pinMode(6,OUTPUT); }</pre>	// Inicio interfase 115200 // imprime el incio // inicio de potencia en el Max // Retardo 200ms // Pin de led	K
void loop() { byte rcode; Max.Task();	// Variable rcode	
		www.dynamoelectronics.com



```
Usb.Task();
 if(Usb.getUsbTaskState() == USB STATE CONFIGURING) { // configurando mouse
    mouse0_init();
 if( Usb.getUsbTaskState() == USB_STATE_RUNNING ) { // tomando datos
    rcode = mouse0_poll();
    if(rcode) {
     Serial.print("Datos de mouse Error "); // si rcode existe entonces que imprima el tipo de error
     Serial.println( rcode, HEX );
                                        // impirmiendo el error
void mouse0_init( void )
                                //Rutina inicio mouse
byte rcode = 0;
                                           //Varaiable rcode variable de error
Usb.setDevTableEntry(1, Usb.getDevTableEntry(0,0)); //copy device 0 endpoint information to device 1
/* Configurando dispositivo */
rcode = Usb.setConf( DEVADDR, 0, CONFVALUE );
                                                            // obteniendo datos Rcode
if(rcode) {
                                       // si Rcode existe imprima el error
 Serial.print("Error configurando el mouse: ");
                                                   // imprime una señal de error
 Serial.println( rcode, HEX );
                                              // imprime el Rcode
 while(1);
                                       // si hay error se queda
Usb.setUsbTaskState( USB_STATE_RUNNING );
                                                             //USB corriendo
return:
byte mouse0 poll(void) /* toamando datos del mouse usando Get Report e imprimiendo los resulatados */
byte rcode,i;
                                               // Vector buffer datos mouse
char buf[ 4 ] = \{0\};
                                                  // Vector buffer ultimos datos mouse
static char old_buf[ 4 ] = { 0 };
                         /* obteniendo datos mouse */
 rcode = Usb.getReport( DEVADDR, 0, 4, 0, 1, 0, buf );
 if(rcode) {
                                              //error
  return( rcode );
 for( i = 0; i < 4; i++) {
                                              //buscando nuevos datos
  if( buf[ i ] != old_buf[ i ] ) {
                                              //comparando datos nuevos y viejos
    break:
  }
 if(i == 4) {
  return(0);
                                             //Si todos los datos son = retorna 0
                      /* Muestra de datos en la consola serial*/
 if(buf[0]&0x01){
  Serial.print("Buton 1 presionado "); // Se pulso el boton 1
   digitalWrite(6,HIGH);
 if( buf[ 0 ] & 0x02 ) {
  Serial.print("Buton 2 presionado "); // Se oprimio el boton 2
   digitalWrite(6,LOW);
```



```
}
 if( buf[ 0 ] & 0x04 ) {
   Serial.print("Buton 3 presionado "); // Se oprimio el boton 3
}
 // se imprime las posiciones del mouse
 Serial.println("");
 Serial.print("Eje X: ");
 Serial.println( buf[ 1 ], DEC);
 Serial.print("Eje Y: ");
  Serial.println( buf[ 2 ], DEC);
 Serial.print("Rueda: ");
 Serial.println( buf[ 3 ], DEC);
 for(i = 0; i < 4; i++) {
  old_buf[ i ] = buf[ i ];
                                   //copia en buffer
 Serial.println("");
 return( rcode );
}
```

Pruebas

	🖆 COM10
1. Conecte una fuente externa	Enviar
2. Conecte el mouse a la host shield	Start
3. Mueva el interruptor a On y el led rojo se encenderá	
4. Abra el monitor serial, donde debe aparecer datos como los	Eje X: 1
5 Pulse el botón derecho y el led se encenderá	Eje Y: O
 Pulse el botón Izquierdo y el led se apagará 	Rueda: 0
	Eje X: 1
Video de funcionamiento <u>Click aquí</u>	Eje Y: 1
	Rueda: 0
	Desplazamiento automático Nueva Línea 🗸 115200 baud 🗸



Links de interés

- 1. Llibreria y otros archivos <u>https://github.com/felis/USB_Host_Shield_2.0</u>
- 2. Ejemplos de uso <u>http://www.circuitsathome.com/communicating-arduino-with-hid-devices-part-1</u>
- http://www.circuitsathome.com/arduino_usb_host_shield_projects
- 3. esquematico <u>http://www.sparkfun.com/datasheets/DevTools/Arduino/VSBHostShield-v13.pdf</u>
- 4. Archivos Eagle <u>http://www.sparkfun.com/datasheets/DevTools/Arduino/USBHostShield-v13.zip</u>
- 5. Hoja de datos http://www.sparkfun.com/datasheets/DevTools/Arduino/MAX3421E.pdf

