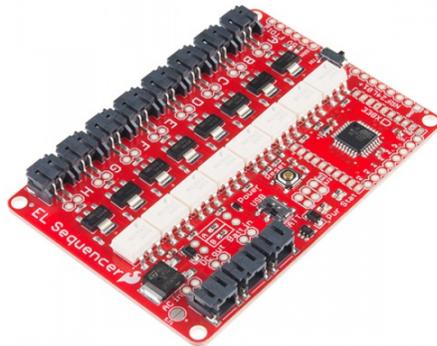




EL Sequencer/Escudo Dos Hookup Guide

Introduction

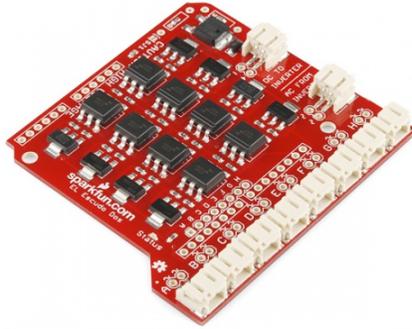
The SparkFun EL Sequencer is an Arduino-compatible microcontroller, with circuitry for controlling up to eight strands of electroluminescent (EL) wire.



SparkFun EL Sequencer

© COM-12781

The Arduino shield version of the Sequencer is the SparkFun EL Escudo Dos, and it can be used with any Arduino Uno footprint-compatible microcontroller.



SparkFun EL Escudo Dos

COM-10878

★★★★☆ 5

Materials Required

To follow along with this tutorial, we recommend you have access to the following materials in addition to your Sequencer or Escudo Dos.

EL Sequencer Hookup Guide Wishlist SparkFun Wish List	
	<p>SparkFun FTDI Basic Breakout - 5V DEV-09716 This is the newest revision of our [FTDI Basic](http://www.sparkfun.co...</p>
	<p>EL Inverter - 12v COM-10469 EL inverters allow you to drive EL (electroluminescent) wires. This par...</p>
	<p>EL Wire - Blue 3m COM-10195 EL wire, or electroluminescent wire, is a flexible wire coated in phosp...</p>
	<p>EL Wire - White 3m COM-10197 EL wire, or electroluminescent wire, is a flexible wire coated in phosp...</p>
	<p>EL Wire - Red 3m COM-10191 EL wire, or electroluminescent wire, is a flexible wire coated in phosp...</p>
	<p>Wall Adapter Power Supply - 12VDC 600mA TOL-09442 This is a high quality AC to DC 'wall wart' which produces a regulated...</p>
	<p>SparkFun USB Mini-B Cable - 6 Foot CAB-11301 This is a USB 2.0 type A to Mini-B 5-pin cable. You know, the mini-B...</p>

Suggested Reading

If you aren't familiar with the following concepts, we recommend reviewing them before beginning to work with the EL Sequencer or Escudo Dos.

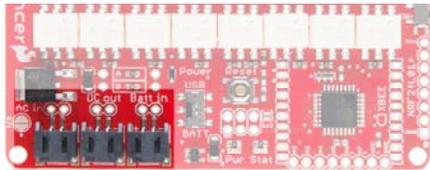
- Installing the Arduino IDE
- How to Power Your Project
- Battery Technologies

- How to Solder
- AC vs. DC Current

Hardware Overview - Similarities

This section will cover the features that are shared between both the EL Sequencer and the EL Escudo Dos.

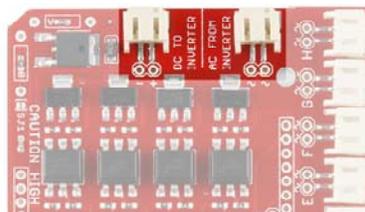
External Power Headers



External Power Headers on the EL Sequencer.

There are three power connection points on the Sequencer. The first is the AC In connection, which is the default connection point for any external inverter. The second connection is the DC out line, which will connect to any inverter that does not have an external power supply. This provides a DC power supply to the inverter from the EL Sequencer itself. The final power connection is the Batt In connection. This allows the user to provide a power supply from a 3.7V lipo battery, or other power supply of 3.5V-16V.

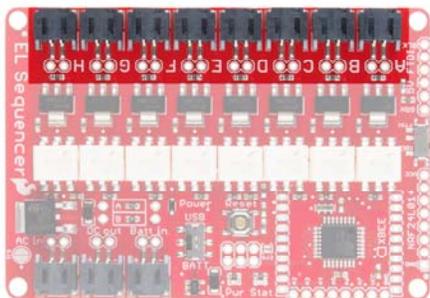
These can be connected via a standard 0.1" 2-pin header, or via the JST connector available on each line. If you ever forget the purpose of these connections, they are clearly labeled on the bottom of the board.



External Power Headers on the EL Escudo Dos.

The power headers on the EL Escudo Dos have the same functionality as the EL Sequencer. However, there is no Batt In header. This is because the power for the Escudo comes from the VRAW line on the connected Arduino or RedBoard.

Channel Headers (HIGH VOLTAGE)

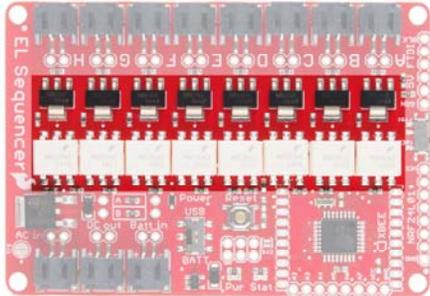


Channel Headers on the Sequencer

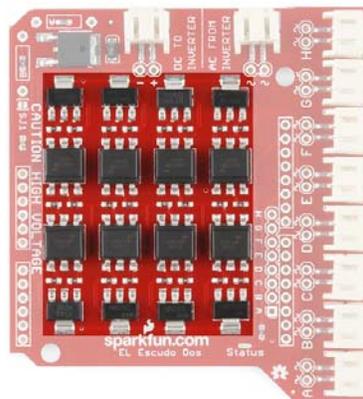
Escudo.

TRIACs and Optoisolators

The main functionality of the Sequencer/Escudo comes from the TRIACs (Triode for Alternating Current) on-board. The MO3063S TRIAC Optocoupler takes the logic output from the associated ATmega328 digital pin which triggers the Z0103MN TRIAC to open the flow of the AC current from the inverter through the gate to drive the EL channel. This keeps the AC side of the circuit optically isolated from the DC portion of the circuit.



EL Sequencer TRIACs and Optoisolators

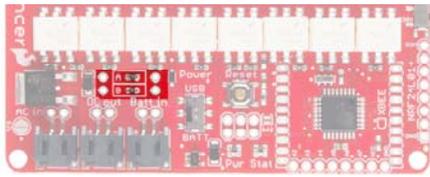


EL EscudoDos TRIACs and Optoisolators

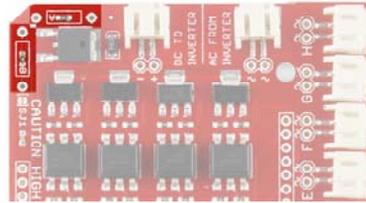
Keep in mind that when the boards are powered and connected to an inverter, touching these areas can lead to an unpleasant shock. It won't hurt you, but it doesn't feel good.

Adjustable Voltage Regulator

The LM317 voltage regulator on this board is adjustable based on the resistor values applied to it. By default, this board comes shipped with a 390Ω resistor on R1 and a 240Ω on R2. However, footprints for thru-hole resistors (A and B) have been added for the user to adjust the voltage out to their project needs. **You'll need to remove the SMD resistors first.** Then add thru-hole resistors of your choice. See the LM317 Datasheet for more information on which resistor values correspond to which voltage output.



Resistors A and B on the EL Sequencer

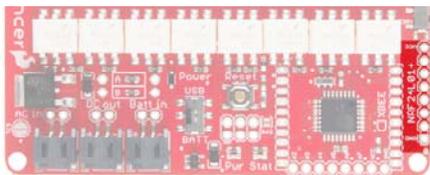


Resistors A and B on the EL Escudo Dos

Hardware Overview - EL Sequencer

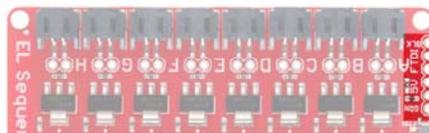
This section will cover the unique features of the EL Sequencer.

NRF24L01+ Header



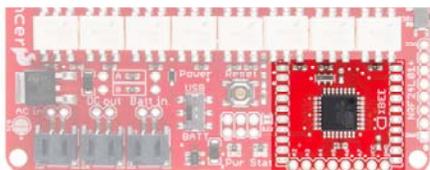
This header is designed to interface with an NRF24L01+ module breakout. This provides a 2.4GHz signal to the EL Sequencer, which provides remote control options for the Sequencer at up to a 100m range.

FTDI Header



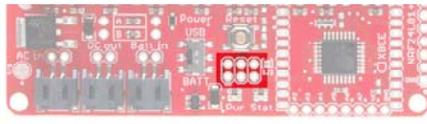
This is a 5V FTDI header, which is given to provide simple access for reprogramming the ATmega328 via serial connections. This can also be used to have the EL wire triggered via serial data in to the system.

XBee Header



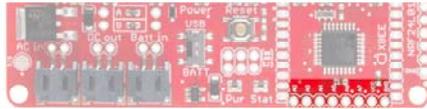
This connection is provided to give users the option of wirelessly connecting the EL Sequencer via XBee. This can be used for mesh networking of several different Sequencers together (this is great for group costume projects), or remote control of the Sequencer.

ICSP Header



This 6-pin header is provided to allow the user to reprogram the ATmega328 on board via an external programmer. This also allows you to reprogram the EL Sequencer without the requirement of the Optiboot loader from the Arduino IDE.

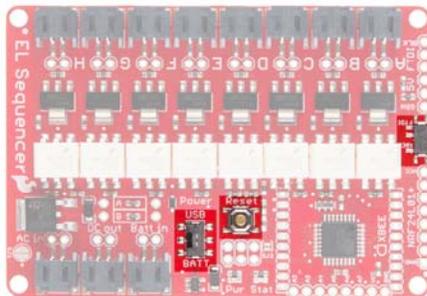
Analog Input Header



The additional analog pins on the ATmega328 are broken out to a standard 0.1" header on the EL Sequencer. This also gives the user the capability to trigger the EL wires from analog inputs (such as a photocell, temperature sensor, or a line sensor).

Pins A6 and A7 are analog input only. Pins A0-A5 can be used as an analog input or as a digital I/O.

Switches and Reset Button

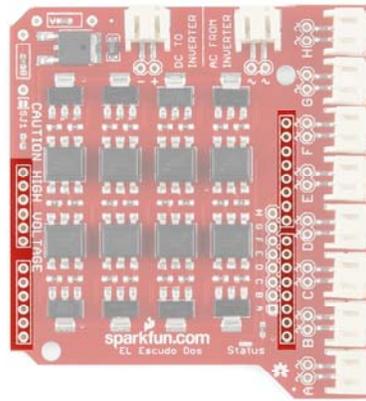


- **XBee to FTDI** (far right, center) - This switch changes the RX line from being supplied via the FTDI header or the XBee. If attempting to program the Sequencer via FTDI, **this switch must be set to the FTDI side**.
- **USB/BATT** (bottom, center) - This switch changes the power source for the ATmega328. The USB setting allows the ATmega328 to be powered via the FTDI connection, while the BATT setting allows the Arduino part to be powered off of a lipo battery instead.
- **Reset Button** - Resets the Arduino sketch running on the ATmega328. It pulls the RESET line low, allowing the system to reset.

Hardware Overview - EL Escudo Dos

This section will cover the unique features of the EL Escudo Dos Arduino Shield.

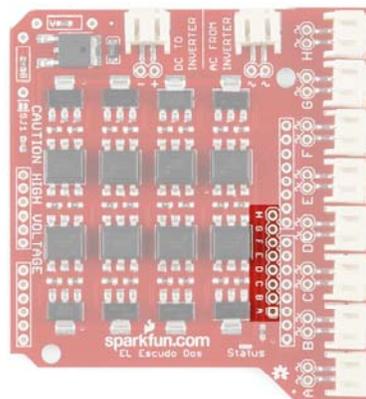
Shield Headers



These headers allow the shield to be connected to any Arduino Uno/Uno R3 compatible board. We will be using the SparkFun RedBoard for this example.

Currently, the EL Escudo Dos Shield is designed for the original Arduino Uno footprint (not the R3 footprint). However, this shield can be used with Arduino Uno R3 boards without issue.

Channel Headers



The EL Escudo also provides a row of 0.1" spaced headers to allow the user to read the signals from the microcontroller to the EL wires. This is beneficial for debugging.

Hardware Hookup

To get started using the EL Sequencer or EL Escudo Dos, there are a few basic steps that must be completed.

Solder Headers

EL Sequencer

At an absolute bare minimum, you must solder 6 pins on the FTDI header. For simplicity's sake, we recommend using male headers (either straight or right angle as these are compatible by default with the FTDI breakout). This will enable you to program the EL Sequencer via an FTDI breakout board.

If you plan on using an XBee unit with your Sequencer, you will also want to solder XBee Sockets to the board. If you plan on using any additional connectors on the board, you will also want to solder those headers on now. Keep in mind you may also want to trim off any excess header on the bottom of the board if you intend to use this in a wearable application.

EL Escudo Dos

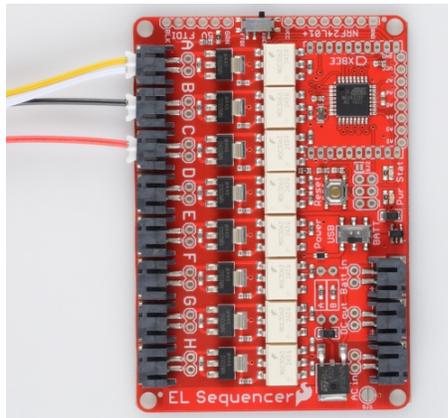
Thanks to the simplified design, you only need to solder Arduino shield headers onto the Escudo. If you want to use any additional components in your project (such as wireless modules or external sensors), you will want to use stackable headers.

Connect the EL Wire

You have 8 channels available on the Sequencer to which you can plug in EL wire, chasing EL wire, or EL panels.

Please check out this tutorial on working with EL Wire specifically, and the current requirements for different types of materials.

Note: If you plan on using **Chasing EL wire**, keep in mind you will need an adapter cable to make it compatible with the EL Sequencer/EL Escudo Dos. You also will need 3 channels available on the EL Sequencer for the chasing wire, in order to get the proper 'chasing' effect to display. This means you will only have 5 additional channels available (or 2, if you plan to use 2 chasing EL wire strands).



Connect the Inverter

Once you have hooked all of your EL wires up, the next step is to hook up the inverter.

- **3V Inverter** - If you are using the 3V inverter, you will need to hook up both wire pairs from the inverter to the Sequencer. The **black/black** wire pair will plug into the AC In JST connector, while the **red/black** wire pair plugs into the DC Out JST connector.
- **12V Inverter** - While using the 12V inverter, there is only one wire pair that will plug into the Sequencer. The red/black wire pair will plug into the AC In JST connector. You will need a separate power supply to run this inverter (such as the 12VDC 600mA Wall Adapter).

Note: Make sure all power has been removed from the system before plugging in any inverter. This will help prevent any inadvertent shocks to the user.

Connect the Power Supply

EL Sequencer

If you are using a battery for the power supply, you can connect this to the `Batt In` JST header without altering the board. Using an external power supply with the 3V inverter **will require closing SJ1**. You will need to connect that supply to the PTH headers labeled `Batt In`. Make sure the power supply falls in the range of 3.5V-16V for an external power supply.

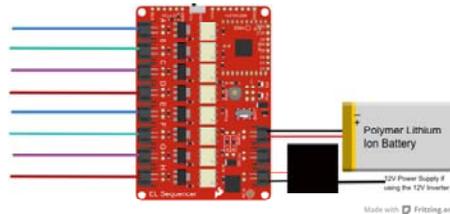
EL Escudo Dos

The power supply for the EL Escudo Dos actually comes from the Arduino. We recommend using a 12V wall adapter for powering the Arduino.

Alternatively, if you would like to go wireless, you could use a male barrel jack adapter along with a 12V battery. Typically, however, the EL Sequencer will be a better option for you for wireless applications.

Final Circuit

Once everything is hooked up, your circuit should look something like the following:



Arduino Code

Once you have all of your hardware hooked up properly, it's time to program your Sequencer to run the EL Wire display as you want. You will need to download the Arduino code in order to follow along with the example.

[DOWNLOAD EL SEQUENCER EXAMPLE FIRMWARE](#)

You can also download the most up-to-date code from the EL Sequencer GitHub repository.

We will be working with the "SparkFun_EL_Example.ino" example. If you need a review on how to upload Arduino code to your board or using the Arduino IDE, please check our tutorial here. Also, if you are unsure about installing the FTDI drivers, please check out this tutorial.

If using the EL Sequencer:

Plug in your FTDI board to the computer via USB, and insert the FTDI onto the EL Sequencer. Make sure you match up the `BLK` to `BLK` and `GRN` to `GRN` labels.

When uploading your code, please be sure to select the following settings in the IDE:

- **Board:** LilyPad Arduino
- **Processor:** ATmega328
- **COM port:** Whichever port the FTDI Basic appears on for your operating system.

If using the EL Escudo Dos + RedBoard:

Plug in the RedBoard to the computer via USB. Select the following settings

in the IDE:

- **Board:** Arduino Uno
- **COM port:** Whichever port the FTDI Basic appears on for your operating system.

Demo Code

This basic demo will cycle through each channel on the Sequencer, one by one, repeating until the system is either reprogrammed or turned off.

In the first section, we must define each pin that we will be using on the ATmega328.

```

/*****Setup Loop*****/
void setup() {
  // The EL channels are on pins 2 through 9 on the ATmega328
  // Initialize the pins as outputs
  pinMode(2, OUTPUT); // channel A
  pinMode(3, OUTPUT); // channel B
  pinMode(4, OUTPUT); // channel C
  pinMode(5, OUTPUT); // channel D
  pinMode(6, OUTPUT); // channel E
  pinMode(7, OUTPUT); // channel F
  pinMode(8, OUTPUT); // channel G
  pinMode(9, OUTPUT); // channel H

  //Uncomment the following line if using this on the EL Escudo
  Dos.
  // Pin 10 is a status LED on the EL Escudo.
  // pinMode(10, OUTPUT);

  //Pin 13 is the status LED on both the EL Sequencer and Arduin
  o for use with the EL Escudo Dos.
  pinMode(13, OUTPUT);
}

```

Each pin is declared as an output, which will allow us to drive the channel HIGH or LOW. We will cycle through each channel, turning them on individually, waiting 1/10 second, turning off, and moving to the next channel.

```

/*****Main Loop*****/
void loop()
{
  int x,status;

  //Step through all eight EL channels (pins 2 through 9)
  for (x=2; x<=9; x++)
  {
    digitalWrite(x, HIGH); // turn the EL channel on
    delay(100);           // wait for 1/10 second
    digitalWrite(x, LOW); // turn the EL channel off

    //digitalWrite(10, status); //Uncomment this line if using t
    he EL Escudo Dos
    digitalWrite(13, status); // blink status LEDs
    status = !status;
  }
}

```

Each time a channel is cycled through, the status LEDs on the board will blink.

Once the code is uploaded, you should see each strand of EL wire light up one by one. If you aren't seeing that, verify that you have the correct power settings on the USB/Batt switch, and that the inverter is connected properly, switched on (if using the 12V inverter), and that the power supply to the board is properly charged and/or connected.

Going Wireless

If you would like to add wireless capabilities to your project, there's just a few more steps to get up and running.

- **Insert a radio unit:** You can either use a paired XBee unit or an NRF24L01+ module. You will need to make sure headers are soldered on for either device. If you need additional information on adding an XBee module, please check out our tutorial [here](#). On the EL Sequencer, you will also need to flip the XBee to FTDI switch to XBee to ensure XBee communication is working properly.
- **Update the Code**

If using a Series 1 XBee:

Upload the 'SparkFun_XBee_EL_Sequencer.ino' sketch to your board. In the first section, the status LED is declared as an output, and the Serial connection is opened at 9600bps.

```
//Declare character 'val'
char val;

/*****Setup Loop*****/
void setup(){
  Serial.begin(9600); //Begin Serial communication for debugging
  pinMode(13, OUTPUT); //Set pin mode as output for status LED
  val = 'A'; // button pressed, therefore sending letter A

  Serial.println("EL Sequencer's XBee is Ready to Receive Characters");
  digitalWrite(13,LOW); //set Status LED off
  delay(1000); //Wait 1 second
}
```

The main loop of the code simply waits for the XBee unit to receive the character 'A' from a remote XBee unit. If the character is received, the status LED is turned on, and a message is output to the serial terminal.

```
/******Main Loop*****/  
void loop(){  
  
  //Check if XBee is receiving data from other XBee  
  if(Serial.available()){  
    val = Serial.read();  
  
    //Check to see if character sent is letter A  
    if(val == 'A'){  
      digitalWrite(13,HIGH); //turn ON Status LED  
      delay(50);  
      Serial.println("Character Received");  
    }  
  
    else{  
      digitalWrite(13,LOW); //turn OFF Status LED  
    }  
  }  
}
```

If using an NRF24L01+:

You will need to use the RF24 library with the nRF24L01+ boards. We have a tutorial with more information regarding installing this library [here](#).

We recommend using the `RF24/examples/Usage/led_remote.ino` example sketch. This is already configured to be compatible with the EL Sequencer with 6 channels of EL wire attached and will trigger each channel based on buttons hit on the remote unit.

The EL Sequencer will trigger the channels on and off based on the LED role in the code.

```

    //
    // LED role. Receive the state of all buttons, and refl
ect that in the LEDs
    //
    if ( role == role_led )
    {
        // if there is data ready
        if ( radio.available() )
        {
            // Dump the payloads until we've gotten everything
            while (radio.available())
            {
                // Fetch the payload, and see if this was the las
t one.
                radio.read( button_states, num_button_pins );

                // Spew it
                printf("Got buttons\n\r");

                // For each button, if the button now on, then tog
gle the LED
                int i = num_led_pins;
                while(i--)
                {
                    if ( button_states[i] )
                    {
                        led_states[i] ^= HIGH;
                        digitalWrite(led_pins[i],led_states[i]);
                    }
                }
            }
        }
    }
}

```

The code waits until the receiver gets data coming in and then swaps the LED states accordingly (in this case, the LEDs will be replaced with the EL channels).

That's it!

That's all you need to do to add wireless capabilities to your EL project. You can now remotely control your EL Sequencer project.

Resources and Going Further

Now that you have your EL Sequencer or EL Escudo Dos up and running, you can start creating your own awesome art displays, interactive costumes, or colorful installations. Making something really neat? Let us know - we'd love to see it!

If you have any feedback, please visit the comments or contact our technical support team at TechSupport@sparkfun.com.

Check out these additional resources for more information and other project ideas.

- [MOC3063 Datasheet](#)
- [Z0103MN 4Q Triac Datasheet](#)
- [LM317 Adjustable Regulator Datasheet](#)
- [EL Sequencer GitHub Repository](#)
- [EL Escudo Dos GitHub Repository](#)
- [EL Wire Burning Man Sign](#)
- [EnLightningment](#)
- [Setting up an EL display with the EL Escudo Dos](#)

- Heartbeat Straight Jacket
- Safe Jacob's Ladder