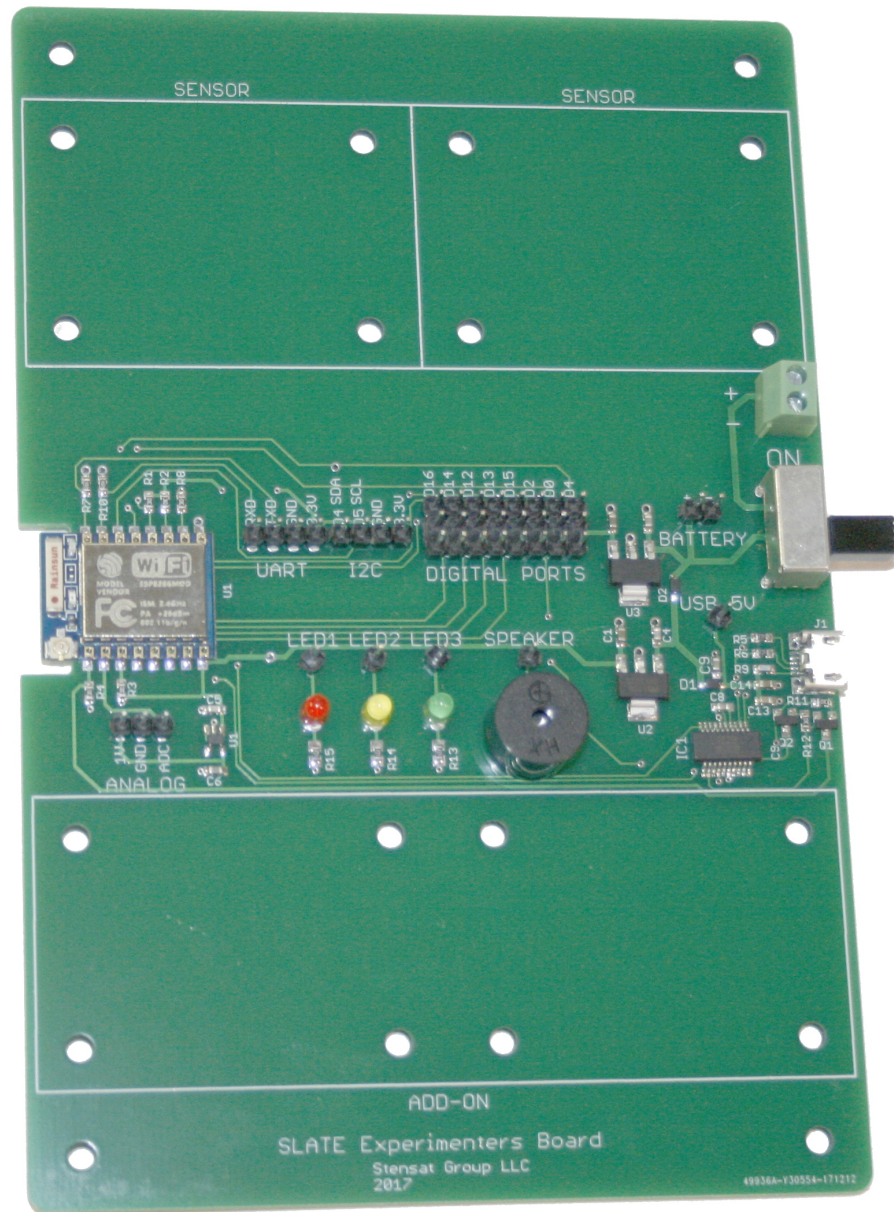


Sten-SLATE ESP Kit

Controlling Hardware



Controlling Hardware

Controlling an LED

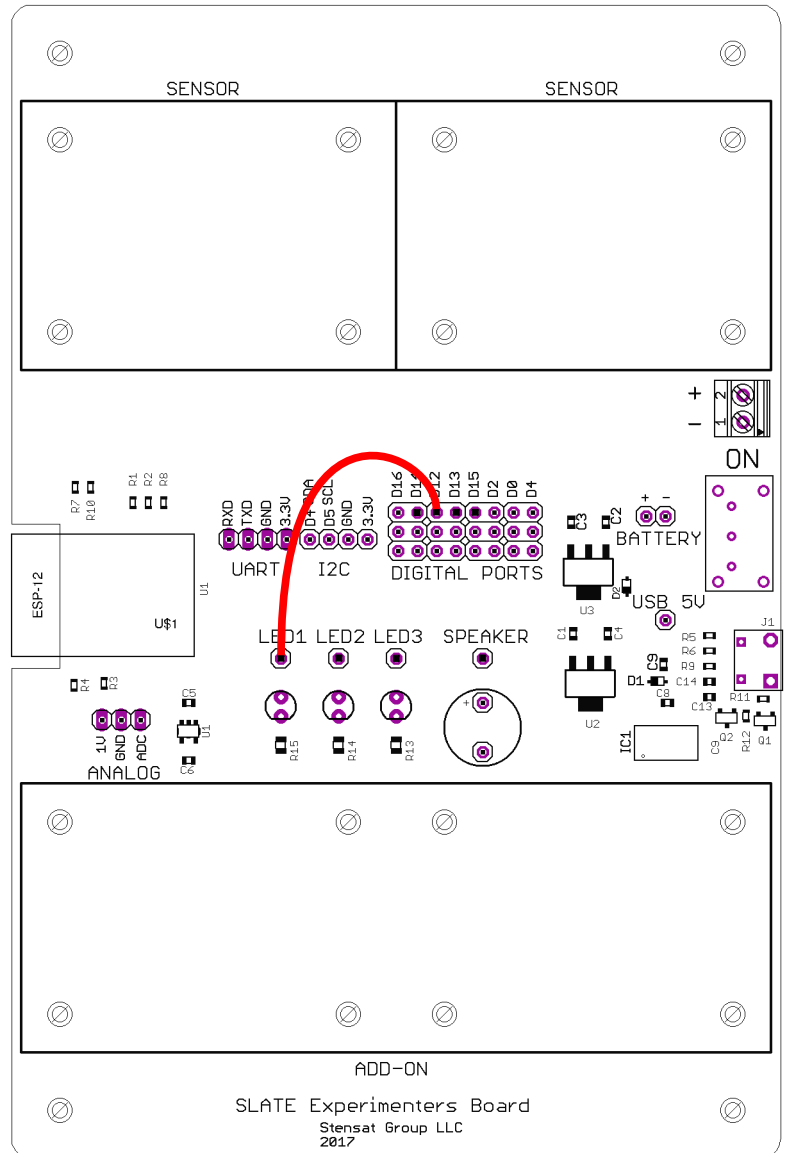
Take a jumper wire and insert one end into pin **D12**. Insert the other end into pin **LED1** as shown in the picture.

Open a new program and enter the code to the right. Click on the **Upload Code** button. You will be asked to save the program to a file. Pick a name like **blinky** and save. The software will then compile and upload the code.

The LED should start blinking.

```
void setup()
{
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(12, HIGH);
  delay(500);
  digitalWrite(12, LOW);
  delay(500);
}
```



How the Code Works

In the `setup()` function, the function **`pinMode(12,OUTPUT)`** is used to configure the digital pin 12 to be an output so it can control the LED. You always need to configure the digital Pin to be an output when using it to control something like the LED.

In the `loop()` function, digital pin 12 is set high which causes the pin to generate 5 volts. The LED turns on. The **`delay()`** function halts the program for 500 milliseconds. The next **`digitalwrite()`** command sets digital pin 12 to 0 volts turning off the LED.

You can think of the digital pin as a light switch. It can be turned on and off. The computer language uses **HIGH** for the **on** state and **LOW** for the **off** state.

Experiment and change the delay settings to blink at different rates. How fast can you make the LED blink before you cannot see if blink?

What is an LED?

LED stands for Light Emitting Diode. It is a semiconductor device that generates light when an electric current passes through it. Current can only travel through the LED in one direction. In the wrong direction, the LED does not light up.

LEDs need the current flowing through it limited otherwise it would pull a lot of current on its own and burn up. A resistor is used to reduce the current flowing through the LED. A resistor is a device that limits current flowing through a circuit. A circuit diagram to the right shows the LED connected to 5 volts.

The LED has an anode and a cathode. When the anode is at a higher voltage than the cathode, the LED will conduct current and light up. Do the reverse and the LED does not light up.

```
void setup()
{
    pinMode(12,OUTPUT);
}

void loop()
{
    digitalWrite(12,HIGH);
    delay(500);
    digitalWrite(12,LOW);
    delay(500);
}
```

Two LEDs

Using the same **blinky** program, you will add a second LED. Connect D13 to LED 2, the yellow LED. Modify your **blinky** program to include the extra code in bold shown to the right.

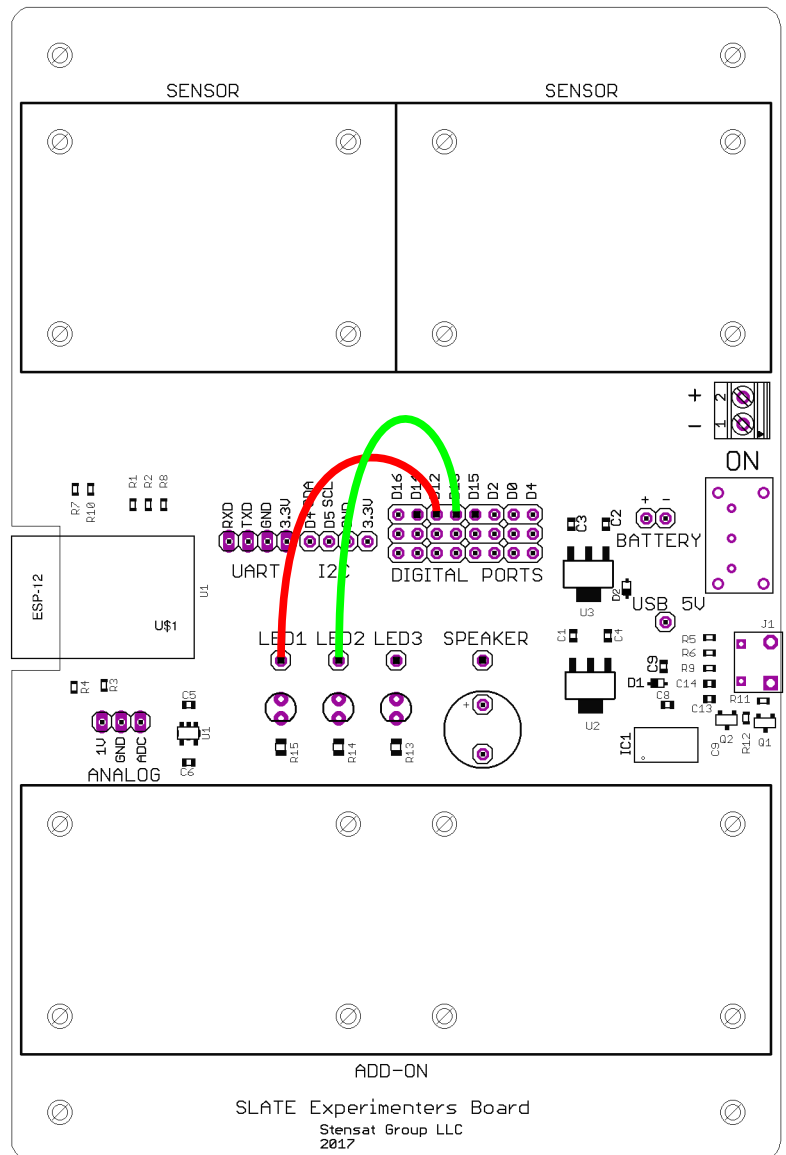
Click on the **Upload Code** button. Once the program uploads, you should have the red and yellow LEDs blinking back and forth.

On your own, add the third LED. Pick a digital pin and connect it to the green LED.

Change the program to blink each LED in sequence.

```
void setup()
{
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
}

void loop()
{
  digitalWrite(12,HIGH);
  digitalWrite(13,LOW);
  delay(500);
  digitalWrite(12,LOW);
  digitalWrite(13,HIGH);
  delay(500);
}
```

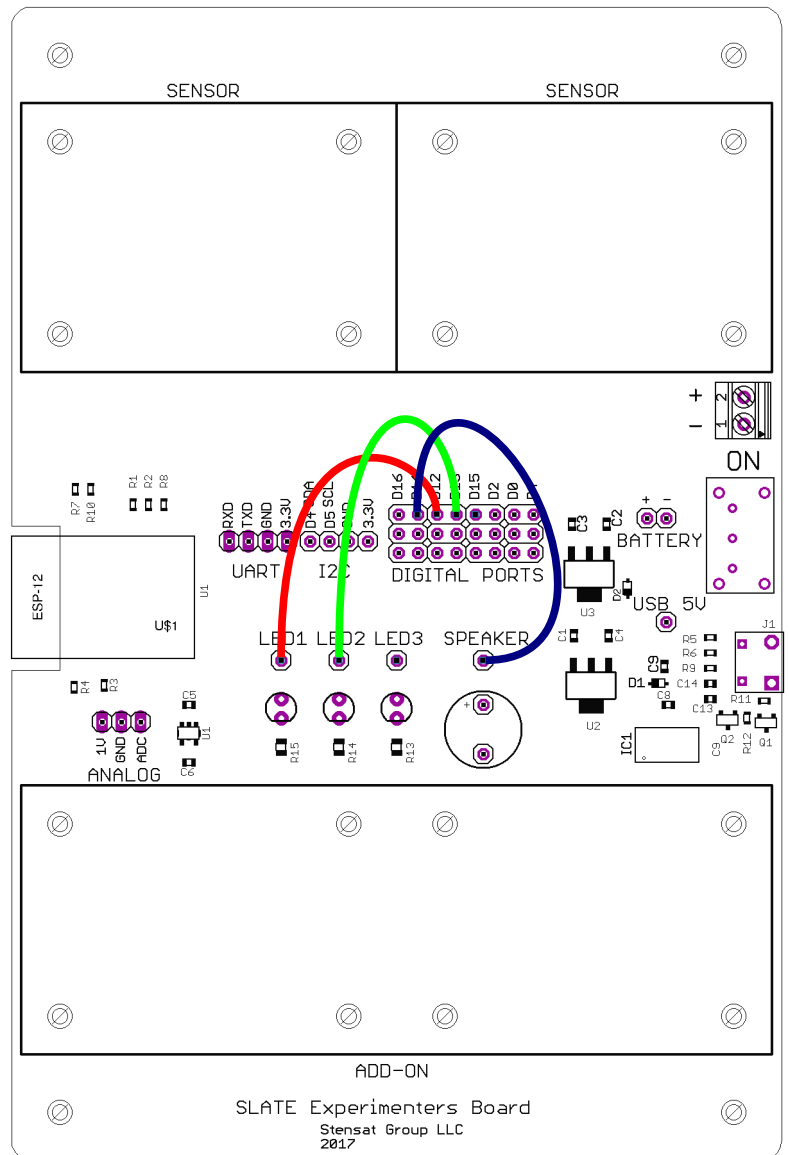


Making Sound

A speaker is a device that can make sounds. Connect digital pin **D14** to the **SPEAKER** pin. Start a new program and enter the code below. To make a sound, you use the **tone()** function. The tone function takes two numbers, the digital pin number and the tone frequency. Try the program below and change the tone frequency. Pick a number between 40 and 5000.

```
void setup()
{
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(14, OUTPUT);
}

void loop()
{
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  tone(14, 2000);
  delay(500);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
  tone(14, 1500);
  delay(500);
}
```



Musical Notes

Musical notes have specific frequencies. Since the `tone()` function uses frequency to set the pitch, you will need to translate notes to frequencies.

A C note has a frequency of 261 Hz. A D note has a frequency of 294 Hz.

In the code shown, there are these **#define** statements that are used to assign a value to a capital letter. These are not variables so they do not take up space. These are constants. Instead of entering numbers for tones, you can just enter the musical note. It makes the code more understandable. Try out the program. Add more notes and different delays.

```
#define C 161
#define D 294
#define E 329
#define F 349
#define G 392
#define A 440
#define B 493
#define C2 523

void setup() {
  pinMode(14, OUTPUT);
}

void loop() {
  tone(14, C);
  delay(500);
  tone(14, C2);
  delay(500);
}
```

Play Music

For this program, you will learn about data arrays and **for()** loops. A data array is a group of variables with the same name but uses a number to select which variable is used. Variable **note** is an array. It is indicated by the square brackets after the name. This bracket is used to specify how many elements or variables are in the array. For this program, there are 7 elements. You also see that the elements are set with the notes in the curly brackets. There can be no more than 7 values put into the array.

There is also the **dur** array. This array is used to determine how long the note plays. 4 is for quarter note, 8 is for 1/8 note. A whole note is 1.

The **for()** loop lets your program loop through a specific number of time. There are three parts to the **for()** loop. The first part sets the starting condition. It declares a variable **i** and sets it to 0. The second part tests the value of **i** and if it is not less than **p** which is 7, the program exits the for loop. The last part increments **i** by 1 each time it loops. All code between the curly brackets after **for()** is executed until **i** is not less than **p**.

Notice the **noTone()** function. This turns off the tone being generated. A short delay is added. This allows same notes playing in a row to be distinct.

Try the program out and see how it plays. Try other music. You can change the size of the array. Remember to change the value of **p**.

```
#define C 161
#define D 294
#define E 329
#define F 349
#define G 392
#define A 440
#define B 493
#define C2 523
#define p 7 // number of notes

int note[p] = { C,C,G,G,A,A,G};
int dur[p] = {4,4,4,4,4,4,1};
void setup() {
  pinMode(14,OUTPUT);
}

void loop() {
  for(int i=0;i<p;i=i+1) {
    int t = 1000/dur[i];
    tone(14,note[i]);
    delay(t);
    noTone(14);
    delay(20);
  }
}
```