

Circuit Playground – Hardware Reference

<https://learn.adafruit.com/introducing-circuit-playground> or <https://learn.adafruit.com/circuit-playground-lesson-number-0>

Processor & Programming:

- ATmega32u4
- Running at 8Mhz (3.3v)
- Program with Arduino
- Reset but no power button
- Program via micro USB (take it easy on that cable!)

Power, Input:

- Max of USB (5v) or Battery
- Protected and regulated
- Battery 3.5-6.5v via JST

Power, Output:

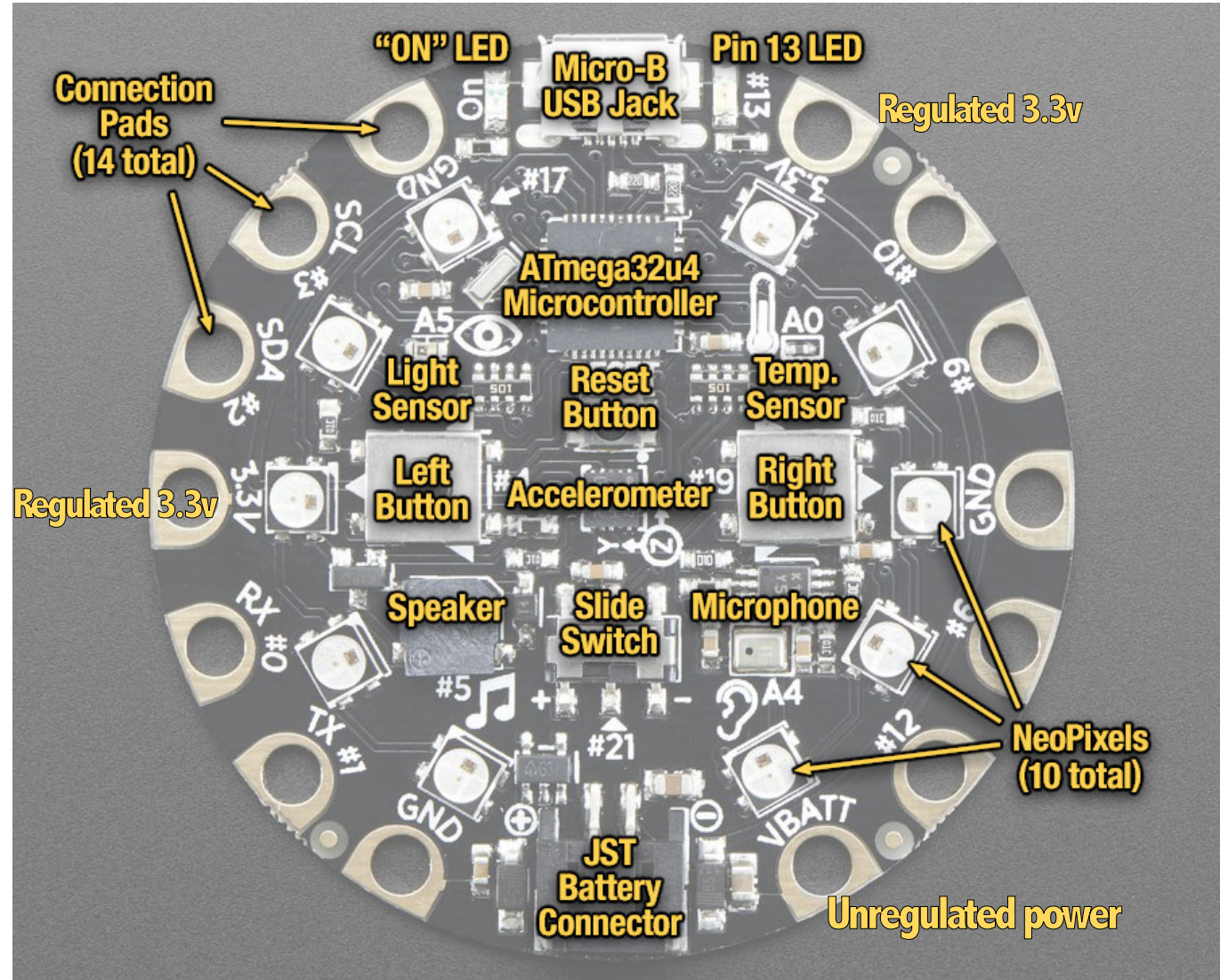
- 3.3v regulated & 3-6v raw
- 500ma max, 175ma over USB

Sensors and Outputs:

- 10 'Neopixels' (pin 17)
- 2 Pushbuttons (pins 4 & 19)
- Slide Switch (pin 21)
- Sound Sensor, raw 100-10khz (pin A4)
- Light Sensor 0-1500 Lux (pin A5)
- Buzzer, square wave only (pin 5)
- Temperature Sensor (pin A0)
- 3 Axis accelerometer (SPI)
 - Adjustable 2-8g range
 - Tap detection
- Red LED (Pin 13)
- Green power LED (can't disable)
- 8 capacitive touch pads (tied to pin 30)
 - Double as I/O pins

I/O pins:

- Serial communication
- Hardware I2C
- 4 PWM / analog read



Circuit Playground – Library Command Reference

<https://learn.adafruit.com/circuit-playground-lesson-number-0>

LED: call `CircuitPlayground.redLED(true)` for on and `CircuitPlayground.redLED(false)` for off.

Slide Switch: call `CircuitPlayground.slideSwitch()`. Returns true for left and false for right.

Buttons: Call `CircuitPlayground.leftButton()` or `CircuitPlayground.rightButton()`. Return true if and only if the button in question is being pressed right when the function is called.

Neopixels: Change individual brightness/color with `CircuitPlayground.setPixelColor(n, red, green, blue)` Where *n* is between 0 and 9 and indicates which LED you are setting. *red/green/blue* vary between 0 (off) and 255 (full on). Turn *all* the LEDs off with `CircuitPlayground.clearPixels()`, Change the global brightness in setup by calling `CircuitPlayground.setBrightness(b)` where *b* varies from 0 to 255.

Temperature: `CircuitPlayground.temperature()` will return a floating point number in Centigrade. If you need Fahrenheit use `CircuitPlayground.temperatureF()`.

Light: request the raw analog reading (0 = no light, up to 1023 = ~1500 Lux) with `CircuitPlayground.lightSensor()`

Sound: read the raw analog value from the amplified microphone with `CircuitPlayground.soundSensor()`. This will give you just a value from between 0 and 1023 where the default 'quiet' voltage is ~330.

Buzzer: You can play basic square wave beeps/tones with `CircuitPlayground.playTone(frequency, duration_ms)`. Where *frequency* is the frequency in Hertz and *duration* is the time to play the beep for, in milliseconds.

Accelerometer:

You can ask the sensor for the amount of *g* detected in X, Y and Z directions. Positive values mean acceleration in the direction of the arrow on the silkscreen.

`CircuitPlayground.motionX()`, `CircuitPlayground.motionY()`, `CircuitPlayground.motionZ()` will each return a floating point value.

Set the range whenever you like using

`CircuitPlayground.setAccelRange(range)` where *range* can be `LIS3DH_RANGE_2_G`, `LIS3DH_RANGE_4_G` or `LIS3DH_RANGE_8_G`

Tap Detection

You can turn on tap detection by calling

`CircuitPlayground.setAccelTap(taps, threshold)`. Tap detection can detect single taps or 'double taps' (like a double-click).

If *taps* is 1 you will only detect single taps, one at a time.

If *taps* is 2, you will be able to detect both single taps and double taps.

You can detect taps in real time by calling

`CircuitPlayground.getAccelTap()`. It will return 0 if no tap is detected, 1 if a single tap is detected, and 2 or 3 if double tap is detected.

The problem with that is you need to query it within about 100 milliseconds of the tapping, which can be difficult to do. If you are comfortable with interrupts you can use the interrupt output to have a function called when a tap is detected.