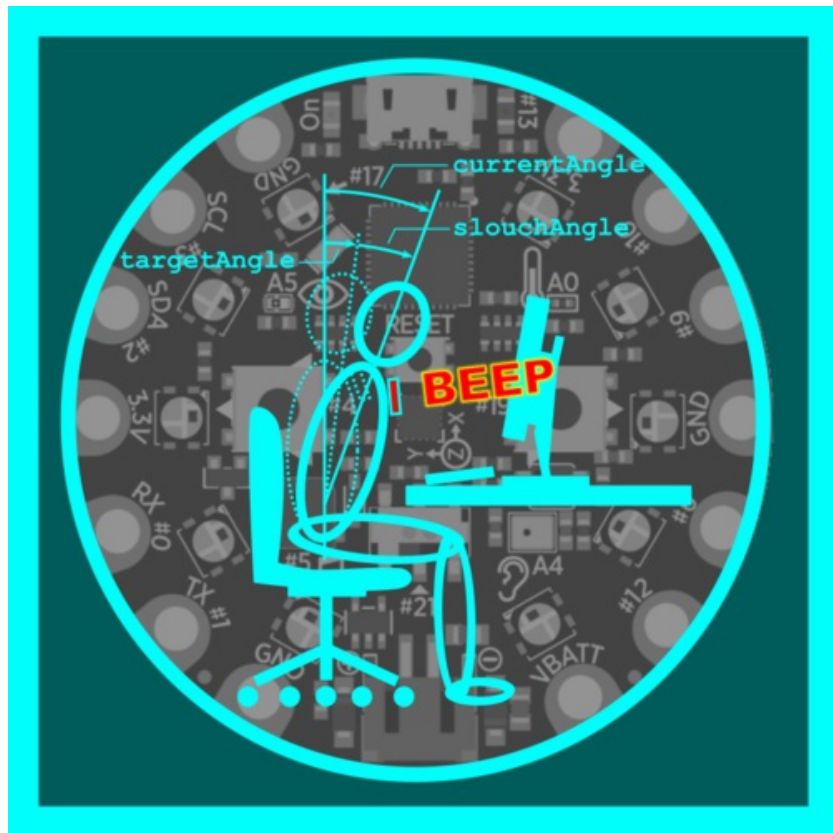


□

## Circuit Playground Slouch Detector

Created by Carter Nelson

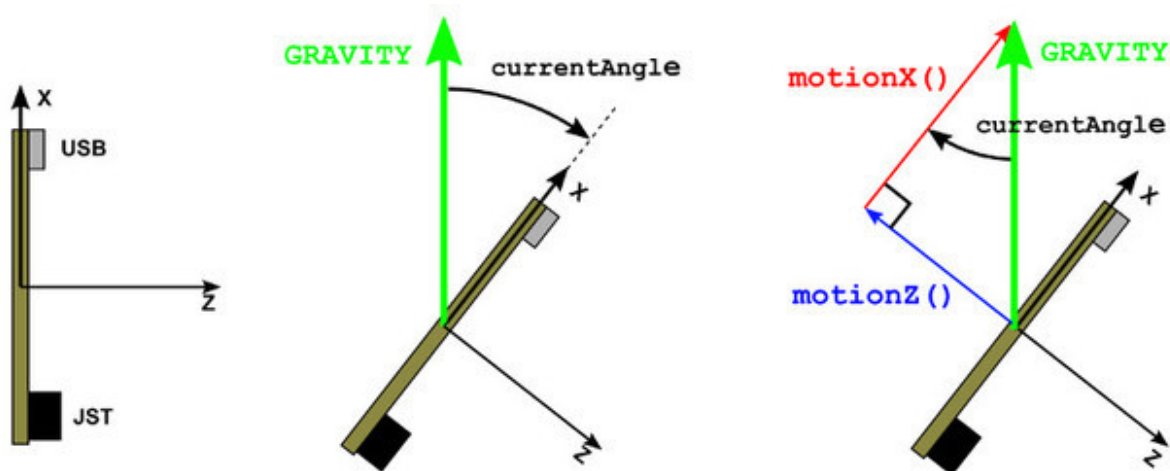


Last updated on 2016-12-21 11:25:27 PM UTC

## Guide Contents

Guide Contents	2
Overview	3
Required Parts	3
Optional Parts	4
Before Starting	4
Hello Slouch	6
Measuring Slouch Angle	8
Arctan Vs. Arcsine	12
Slouch Detector v1	14
Slouch Detector v2	16
Slouch Detector v3	19
Make It Wearable	22
Pocket It	22
Silver Pin	22
Magnetic Pin	23
Locating and Activating	24
Questions and Code Challenges	26
Questions	26
Code Challenges	26

# Overview



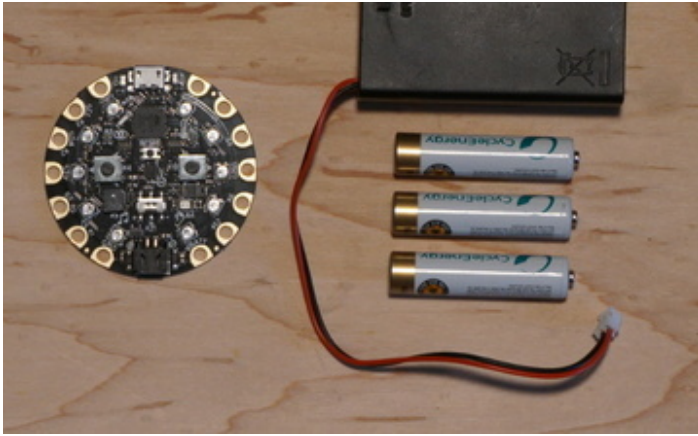
In this guide we will go through the steps to turn your Circuit Playground into a slouch detector. Using one of several pinning options along with various battery supply options, you can then wear your Slouch Detector while sitting at a desk. It will provide a warning if you, well, get a little slouchy - hopefully with reminders, you can keep your good posture and maintain a healthy back!

## Required Parts

This project uses the sensors already included on the Circuit Playground so no additional electronics or soldering are required. You will also need some batteries and a holder for the batteries.



- [Circuit Playground](#) (<http://adafru.it/3000>)
- [3 x AAA Battery](#)



- [Holder](http://adafru.it/727) (<http://adafru.it/727>)
- 3 x AAA Batteries (NiMH work great!)

## Optional Parts

These items help make this project more wearable. There are a couple ways to pin the Circuit Playground in the right location. Additionally, there are a few smaller sized battery options to make it more pocketable. A battery extension cable allows for placing the battery in a different pocket.



- [Silver Pin](#)  
[Back](http://adafru.it/2838) (<http://adafru.it/2838>)
- [Magnetic Pin](#)  
[Back](http://adafru.it/1170) (<http://adafru.it/1170>)
- [JST Extension](#)  
[Cable](http://adafru.it/1131) (<http://adafru.it/1131>)
- [LiPo](#)  
[Battery](http://adafru.it/1578) (<http://adafru.it/1578>)
- [Coin Cell Battery](#)  
[Holder](http://adafru.it/783) (<http://adafru.it/783>)
- [Coin Cell](#)  
[Batteries](http://adafru.it/654) (<http://adafru.it/654>)

## Before Starting

If you are new to the Circuit Playground, you may want to first read these overview guides.

- [Overview](http://adafru.it/ncG) (<http://adafru.it/ncG>)
- [Lesson #0](http://adafru.it/rb4) (<http://adafru.it/rb4>)

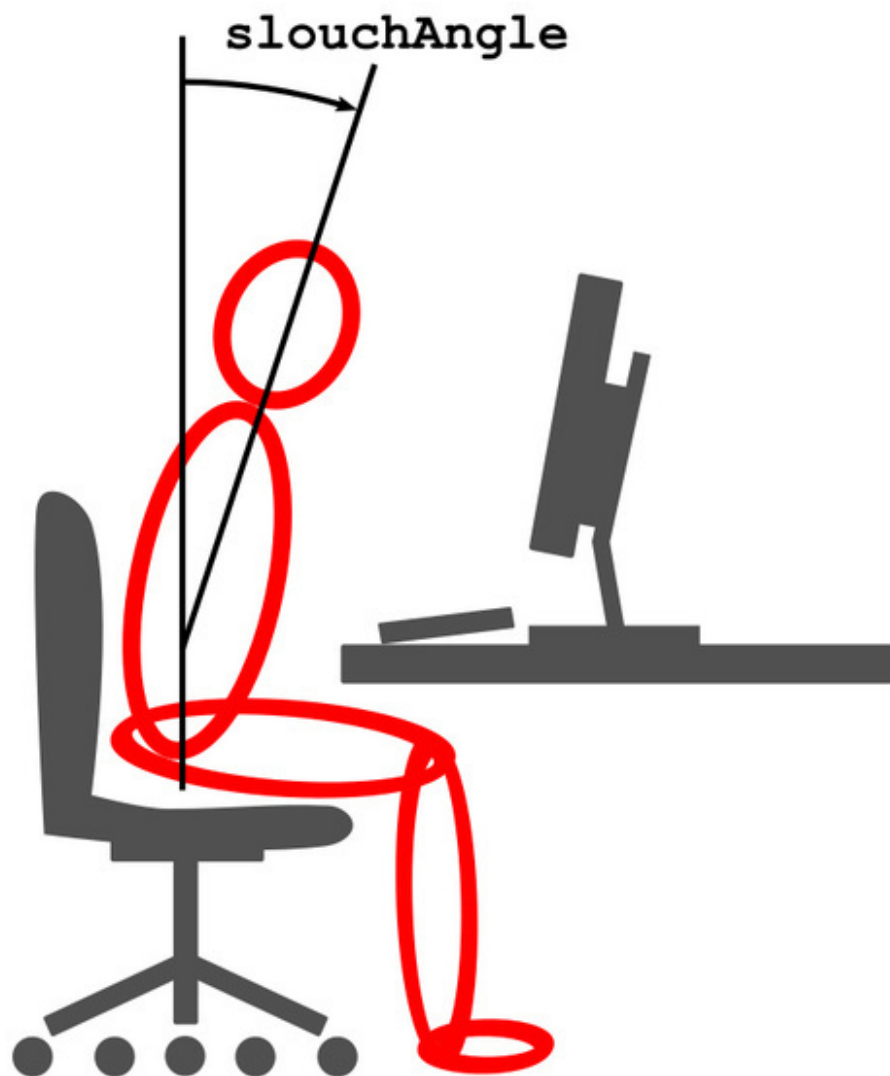
This project will use the Arduino IDE. Make sure you have added the board support for the Circuit Playground as well as installed the Circuit Playground library. **MUST DO BOTH.** This is covered in the guides linked above.

# Hello Slouch

First, what is slouching? It has a few meanings, but we are concerned about how it applies to posture. We'll mainly focus on sitting posture, so let's introduce a reference standard sitting **Stick Figure of Engineering** (SFoE), as shown below.



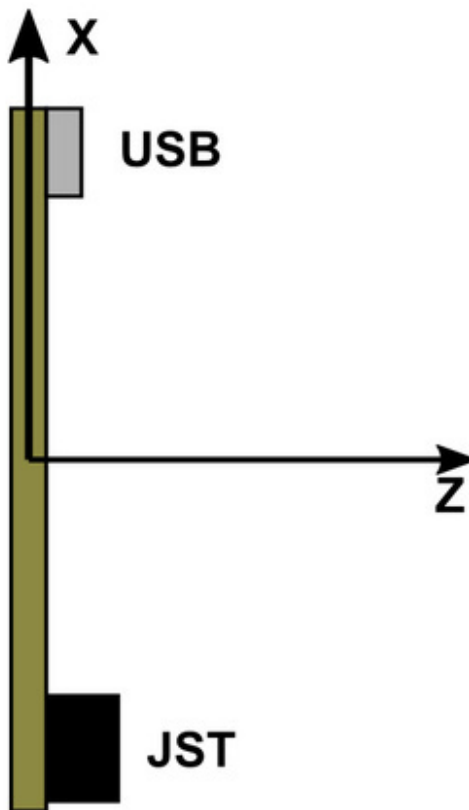
The SFoE is sitting nice and straight with good posture. If we imagine a line going up through the body of the SFoE, then we can think of slouch as an angle between this line and the vertical as shown in the figure below.



The basic idea is to monitor this angle. We will constantly compare it to a preset value. If the angle exceeds this value, then we'll say the SFoE is slouching. Let's see how we can measure this angle with the Circuit Playground.

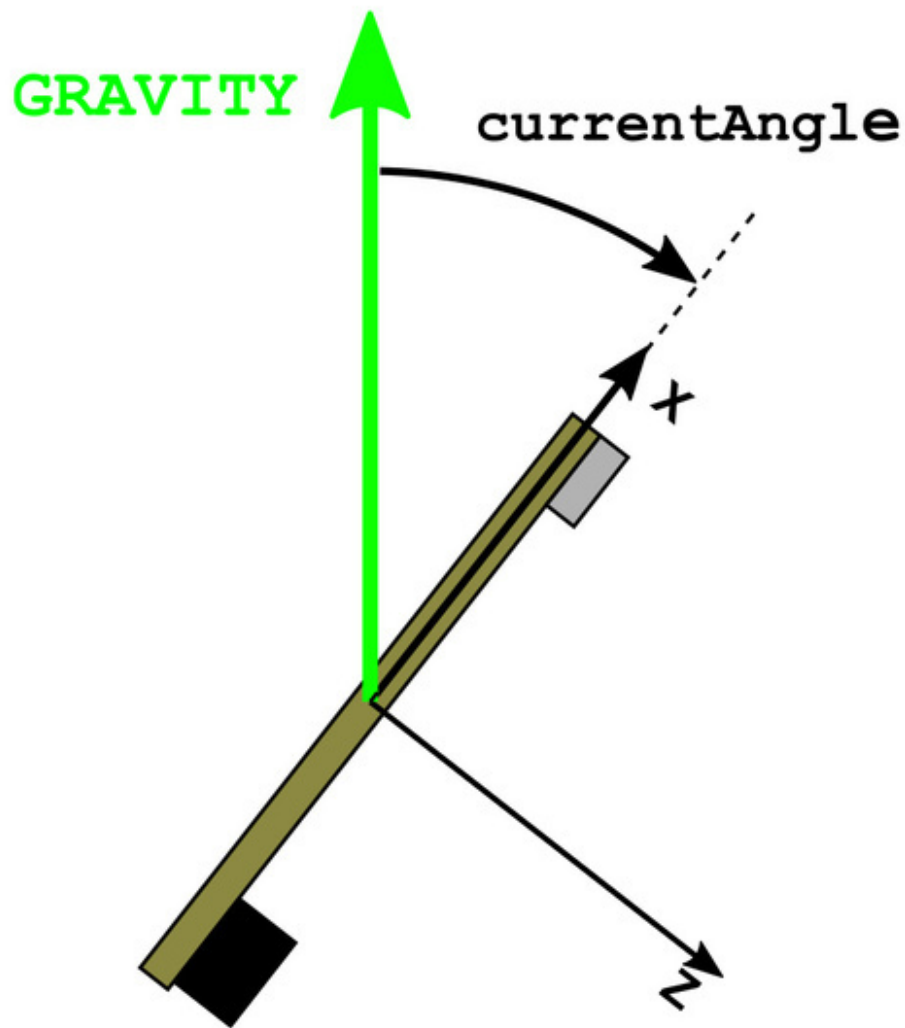
# Measuring Slouch Angle

We will use the accelerometer on the Circuit Playground to measure the slouch angle. A more in depth overview of the accelerometer can be found in the [How Tall Is It? Guide](http://adafru.it/t9f) (<http://adafru.it/t9f>). If you look at the Circuit Playground sideways, the coordinate system of the accelerometer is defined as shown in the figure below.

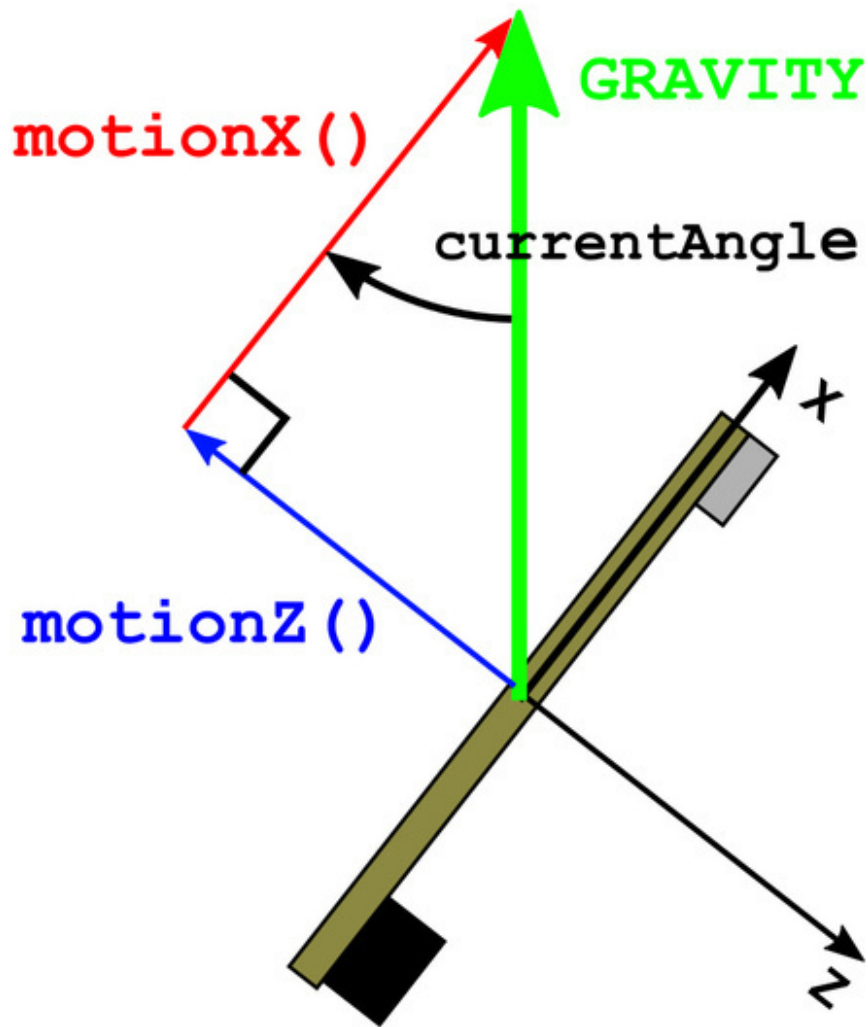


In the figure below, the Circuit Playground is tilted away from the vertical by an amount we are calling **currentAngle**. The big green arrow is the acceleration due to gravity. It is always present and always points vertically. ([see How Tall Is It? Guide for more info](http://adafru.it/tcA) (<http://adafru.it/tcA>))





A portion of the green arrow will be sensed by the X axis and returned by `motionX()`. Similarly, a portion will be sensed by the Z axis as returned by `motionZ()`. All together, these values form a right triangle as shown in the figure below. This let's us use the basic right angle functions shown to compute the value for **currentAngle**.



$$\text{currentAngle} = \text{atan}\left(\frac{-\text{motionZ}()}{\text{motionX}()}\right)$$

$$\text{currentAngle} = \text{asin}\left(\frac{-\text{motionZ}()}{\text{GRAVITY}}\right)$$

Note that the portion of gravity sensed by the Z axis is in the negative direction. Therefore, a - sign shows up in the equations.

Two ways are shown above for computing **currentAngle**. One uses both the `motionX()` and `motionZ()` values in the arctangent function, `atan()`. The second one only uses the `motionZ()` value in

the arcsine function, `asin()`. **GRAVITY** is a constant which, on Earth at sea level, is about  $9.80665 \text{ m/s}^2$ .

So which equation should we use? We'll let's take a look at how they behave.

# Arctan Vs. Arcsine

You can use the sketch below to observe the behavior of computing the Circuit Playground tilt angle via the two different methods.

```
////////////////////////////////////
// Circuit Playground Slouch Detector Angle Demo
//
// Compute current angle using accelerometer.
// Compare asin() to atan2().
//
// Author: Carter Nelson
// MIT License (https://opensource.org/licenses/MIT)

#include <Adafruit_CircuitPlayground.h>

#define GRAVITY      9.80665 // standard gravity (m/s^s)
#define RAD2DEG      52.29578 // convert radians to degrees

float currentAngle1;
float currentAngle2;

////////////////////////////////////
void setup() {
  Serial.begin(9600);

  // Initialize Circuit Playground
  CircuitPlayground.begin();
}

////////////////////////////////////
void loop() {
  // Compute current angle
  currentAngle1 = RAD2DEG * asin(-CircuitPlayground.motionZ() / GRAVITY);
  currentAngle2 = RAD2DEG * atan2(-CircuitPlayground.motionZ() ,
                                   CircuitPlayground.motionX() );

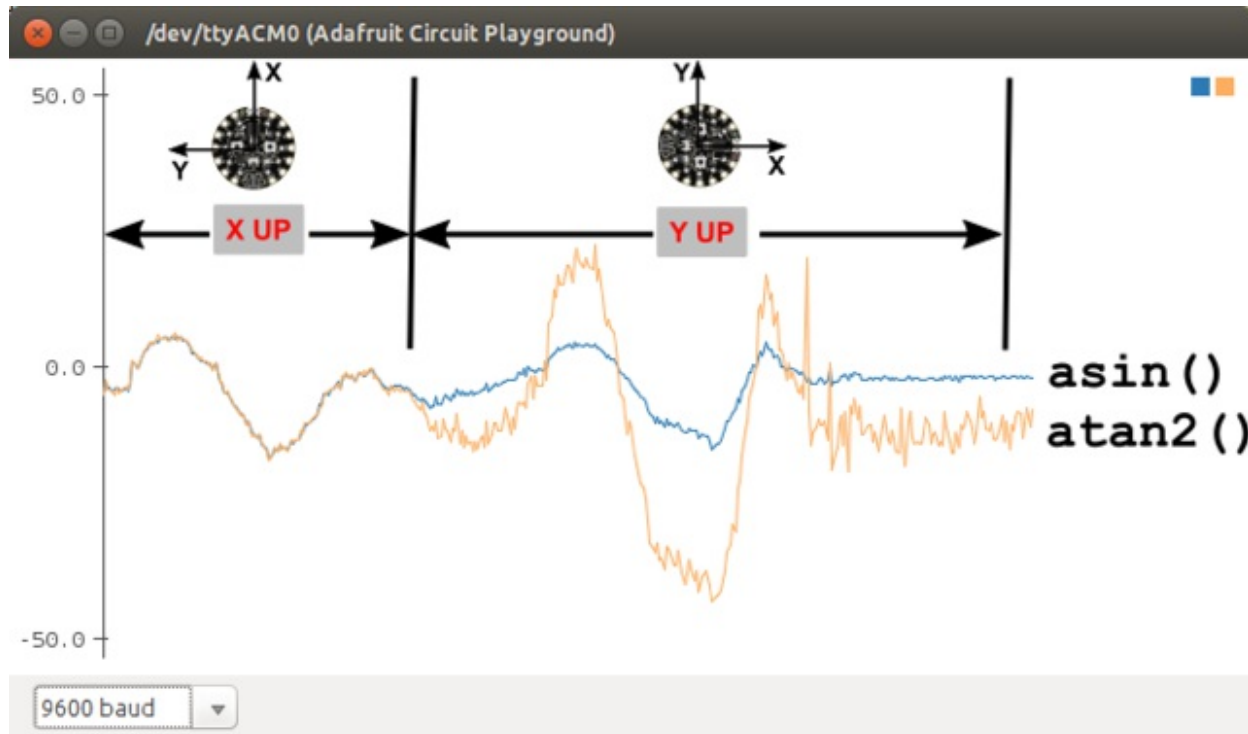
  // Print current angle
  Serial.println(currentAngle1);
  Serial.print(",");
  Serial.println(currentAngle2);

  // But not too fast
  delay(100);
}
```

Copy and paste this code into the Arduino IDE and upload it to the Circuit Playground. Then open the serial plotter:

## Tools -> Serial Plotter

First, try holding the Circuit Playground with the micro USB connector pointed up. This makes the X axis point up as shown in the figures in the previous section. Now tilt the Circuit Playground a little and watch the two lines. They should follow each other pretty closely. Now rotate the Circuit Playground so the micro USB connector is pointed to the side. Tilt the Circuit Playground again and you should see the lines behaving very differently. Something like what is shown in the figure below.



The blue line is from the `asin()` function and continues to behave correctly. However, the orange line, which comes from the `atan2()` function only works when the X axis is pointing up. For our slouch detector, we can't assume this to be true. Therefore, we will use the `asin()` equation to compute the value for **currentAngle**.

# Slouch Detector v1

The idea of the slouch detector is simple. If the current angle sensed by the Circuit Playground (**currentAngle**) exceeds a preset value (**SLOUCH\_ANGLE**), then we are slouching and should sound an alarm. We can achieve this with a simple modification of our previous sketch. Here's the code.

```
////////////////////////////////////
// Circuit Playground Slouch Detector v1
//
// Compute current angle using accelerometer and compare
// to preset slouch angle. Sound alarm if slouching.
//
// Author: Carter Nelson
// MIT License (https://opensource.org/licenses/MIT)

#include <Adafruit_CircuitPlayground.h>

#define SLOUCH_ANGLE    10.0    // allowable slouch angle (deg)
#define GRAVITY         9.80665 // standard gravity (m/s^s)
#define RAD2DEG         52.29578 // convert radians to degrees

float currentAngle;

////////////////////////////////////
void setup() {
  // Initialize Circuit Playground
  CircuitPlayground.begin();
}

////////////////////////////////////
void loop() {
  // Compute current angle
  currentAngle = RAD2DEG * asin(-CircuitPlayground.motionZ() / GRAVITY);

  // Check if slouching
  if (currentAngle > SLOUCH_ANGLE) {
    // Sound alarm
    CircuitPlayground.playTone(800, 500);
  }
}
```

The preset angle is defined globally at the top of the sketch.

```
#define SLOUCH_ANGLE    10.0    // allowable slouch angle (deg)
```

And then a simple if statement is used to check for slouching.

```
// Check if slouching
```

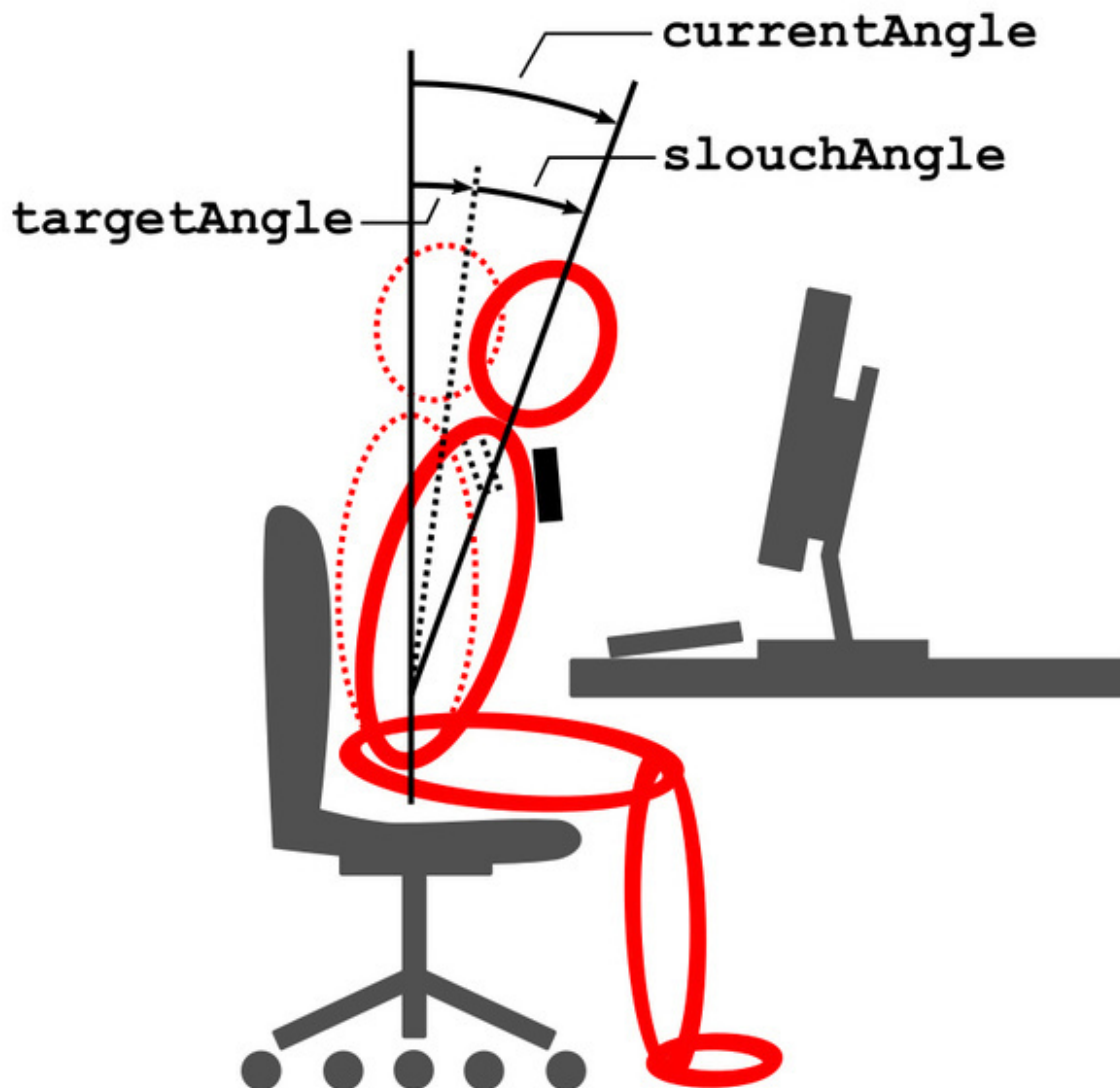
```
if (currentAngle > SLOUCH_ANGLE) {  
  // Sound alarm  
  CircuitPlayground.playTone(800, 500);  
}
```

With the above sketch loaded and running on the Circuit Playground, you should hear the alarm sound if the tilt (slouch) angle is greater than 10 degrees.

## Slouch Detector v2

While the v1 version of our slouch detector works, there are a few features we need to add. The first is to account for the fact that the Circuit Playground may not sit perfectly vertical when being worn.

To deal with this, we need to generalize the problem a little further. Here is another version of our SFoE with some more angle definitions.



The dashed lines represent the SFoE sitting with good posture, which we are defining as the **targetAngle**. We want to determine if the SFoE exceeds the **slouchAngle** (**SLOUCH\_ANGLE**), but the Circuit Playground only reports the angle relative to the vertical line, **currentAngle**.



If we could determine the **targetAngle**, we could compute the **slouchAngle** from the **currentAngle** as follows:

$$\text{currentAngle} = \text{targetAngle} + \text{slouchAngle}$$

(rearrange)

$$\text{slouchAngle} = \text{targetAngle} - \text{currentAngle}$$

So how do we come up with the **targetAngle**? We will make this settable by using the buttons on the Circuit Playground. The idea will be:

- Attach Circuit Playground to clothing.
- Sit with good posture.
- Press either button to set **currentAngle** as **targetAngle**.
- Do math shown above to compute **slouchAngle**.
- Compare **slouchAngle** to preset value, **SLOUCH\_ANGLE**.
- Sound alarm if value exceeded.

Here is the code that adds this feature:

```
/////////////////////////////////////////////////////////////////
// Circuit Playground Slouch Detector v2
//
// Push button(s) to set a target angle.
// Compute current angle using accelerometer and compare
// to preset slouch angle. Sound alarm if slouching.
//
// Author: Carter Nelson
// MIT License (https://opensource.org/licenses/MIT)

#include <Adafruit_CircuitPlayground.h>

#define SLOUCH_ANGLE    10.0    // allowable slouch angle (deg)
#define GRAVITY         9.80665 // standard gravity (m/s^s)
#define RAD2DEG         52.29578 // convert radians to degrees

float currentAngle;
float targetAngle;

/////////////////////////////////////////////////////////////////
void setup() {
  // Initialize Circuit Playground
  CircuitPlayground.begin();

  // Initialize target angle to zero.
  targetAngle = 0;
}

/////////////////////////////////////////////////////////////////
void loop() {
```

```

// Compute current angle
currentAngle = RAD2DEG * asin(-CircuitPlayground.motionZ() / GRAVITY);

// Set target angle on button press
if ((CircuitPlayground.leftButton()) || (CircuitPlayground.rightButton())) {
  targetAngle = currentAngle;
  CircuitPlayground.playTone(900,100);
  delay(100);
  CircuitPlayground.playTone(900,100);
  delay(100);
}

// Check if slouching
if (currentAngle - targetAngle > SLOUCH_ANGLE) {
  // Sound alarm
  CircuitPlayground.playTone(800, 500);
}
}

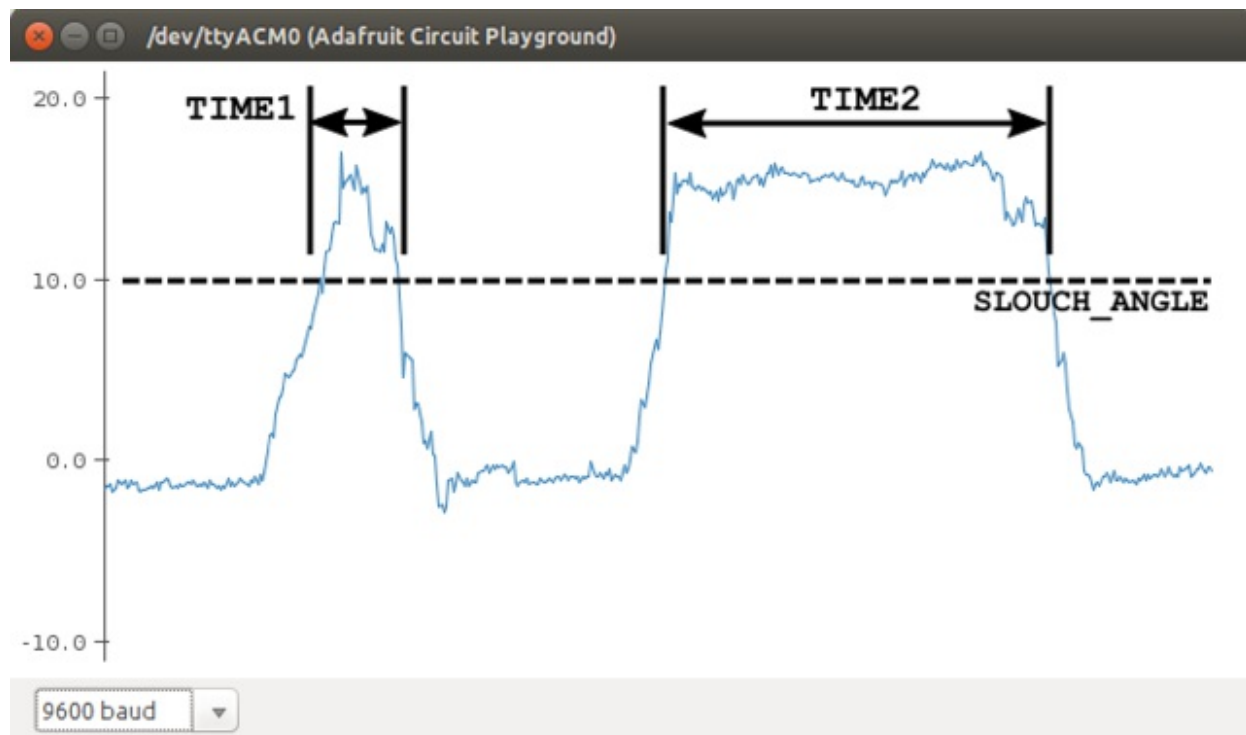
```

Now you can press either button to set the **targetAngle** to the **currentAngle**. Two small beeps will sound to confirm this.

# Slouch Detector v3

The next feature we need to add is a time delay for the alarm. Currently, it sounds the instant the slouch angle is exceeded. But what if you were just leaning over to look at something? We should give you a chance to straighten back up before sounding the alarm.

This idea is shown in the figure below, which shows two cases where the current angle exceeded the slouch angle.



We don't want to sound an alarm for the first case (TIME1). That was a brief event like someone bending over to look at something. However, in the second case, the angle has been exceeded for a longer period of time (TIME2). Either the person is slouching or has fallen asleep. In either case, we want to sound the alarm.

To make this work, we need to keep track of time. Specifically, how long we've been slouching, and compare that to a preset value (**SLOUCH\_TIME**). We only sound the alarm if (a) we are slouching, and (b) we've been doing so for longer than this preset value.

Here is the code that adds this feature:

```
//////////////////////////////////////////  
// Circuit Playground Slouch Detector v3  
//  
// Push button(s) to set a target angle.
```

```

// Compute current angle using accelerometer and compare
// to preset slouch angle. Sound alarm if slouching after
// a preset period of time.
//
// Author: Carter Nelson
// MIT License (https://opensource.org/licenses/MIT)

#include <Adafruit_CircuitPlayground.h>

#define SLOUCH_ANGLE    10.0    // allowable slouch angle (deg)
#define SLOUCH_TIME     3000    // allowable slouch time (secs)
#define GRAVITY         9.80665 // standard gravity (m/s^s)
#define RAD2DEG         52.29578 // convert radians to degrees

float currentAngle;
float targetAngle;
unsigned long slouchStartTime;
bool slouching;

////////////////////////////////////
void setup() {
  // Initialize Circuit Playground
  CircuitPlayground.begin();

  // Initialize target angle to zero.
  targetAngle = 0;
}

////////////////////////////////////
void loop() {
  // Compute current angle
  currentAngle = RAD2DEG * asin(-CircuitPlayground.motionZ() / GRAVITY);

  // Set target angle on button press
  if ((CircuitPlayground.leftButton()) || (CircuitPlayground.rightButton())) {
    targetAngle = currentAngle;
    CircuitPlayground.playTone(900,100);
    delay(100);
    CircuitPlayground.playTone(900,100);
    delay(100);
  }

  // Check for slouching
  if (currentAngle - targetAngle > SLOUCH_ANGLE) {
    if (!slouching) slouchStartTime = millis();
    slouching = true;
  } else {
    slouching = false;
  }

  // If we are slouching
  if (slouching) {
    // Check how long we've been slouching
    if (millis() - slouchStartTime > SLOUCH_TIME) {

```

```

    // Play a tone
    CircuitPlayground.playTone(800, 500);
  }
}
}

```

Our new preset value is again set globally at the top of the sketch.

```

#define SLOUCH_TIME      3000    // allowable slouch time (secs)

```

We add a few variables to keep track of how long we've been slouching.

```

unsigned long slouchStartTime;
bool slouching;

```

We replace the alarm with time tracking logic. If we aren't slouching, then we set that value false and nothing else happens. However, if we are currently slouching, we set the value to true and note the time when this started.

```

// Check for slouching
if (currentAngle - targetAngle > SLOUCH_ANGLE) {
  if (!slouching) slouchStartTime = millis();
  slouching = true;
} else {
  slouching = false;
}

```

And then move the alarm to a new set of if statements. If we are currently slouching, then check for how long. If it's been too long, sound the alarm.

```

// If we are slouching
if (slouching) {
  // Check how long we've been slouching
  if (millis() - slouchStartTime > SLOUCH_TIME) {
    // Play a tone
    CircuitPlayground.playTone(800, 500);
  }
}

```

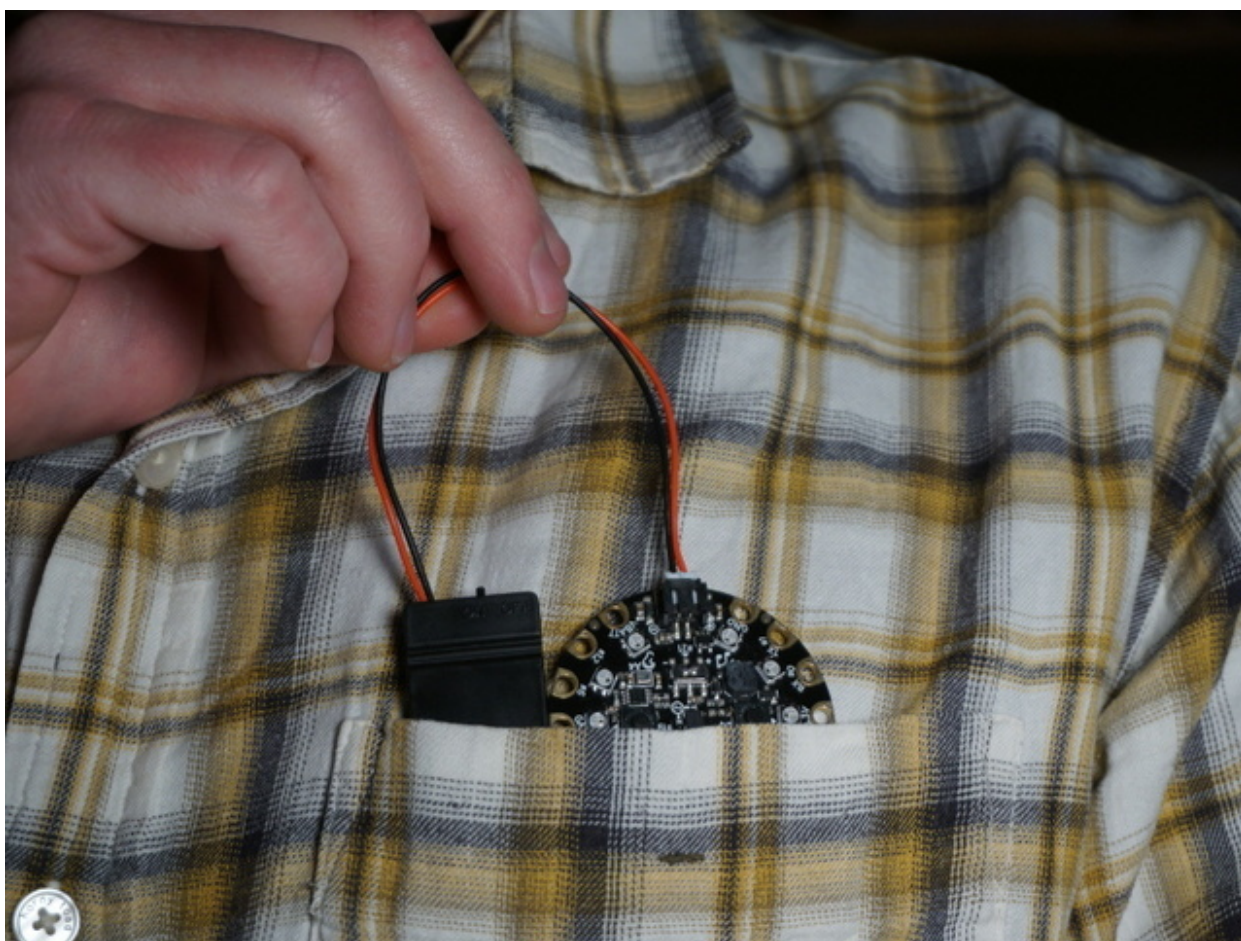
And that's it. Now let's see how we can make this thing wearable.

# Make It Wearable

There are a few options for attaching your Circuit Playground Slouch Detector to your garment. Some other items, like the [JST extension cable](http://adafru.it/1131) (<http://adafru.it/1131>), can also make locating the battery more convenient.

## Pocket It

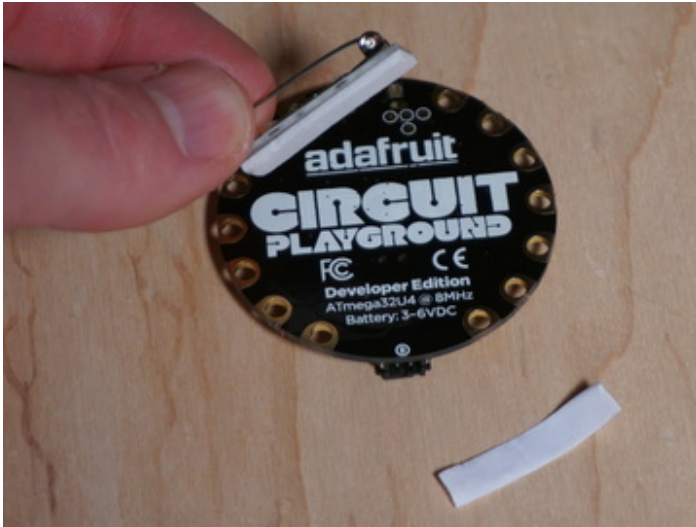
If you have a shirt with a front pocket, you can just drop it in there.



## Silver Pin

The [silver pin with adhesive foam back](http://adafru.it/2838) (<http://adafru.it/2838>) can be used to attach the Circuit Playground to most any garment.





Remove the protective film from the adhesive back and apply the pin to the back of the Circuit Playground.



Press it down firmly and you're good to go.

## Magnetic Pin

The [magnetic pin](http://adafru.it/1170) (<http://adafru.it/1170>) is a nice option if you don't want to stick holes in your shirt. Or just don't like sharp pointy things.



Remove the protective film from the adhesive back and apply the metal strip to the back of the Circuit Playground.

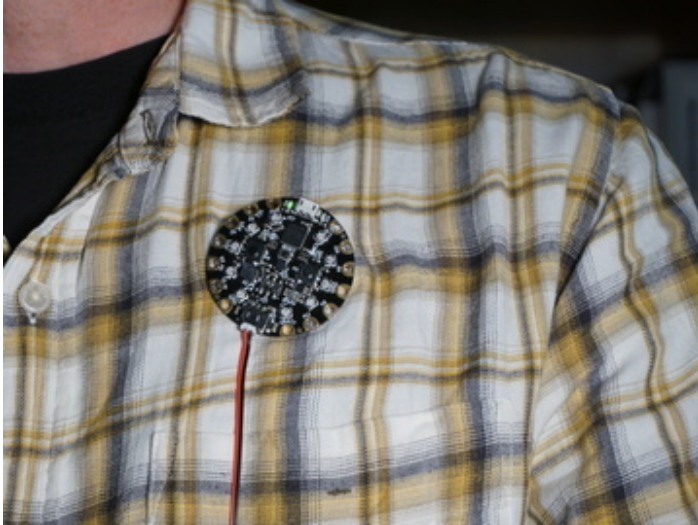


Just to be safe, apply the pin at the angle shown. It is metal and if you put it on horizontally, there's a slim chance it could short the 3.3V pad on one side to the GND pad on the other.

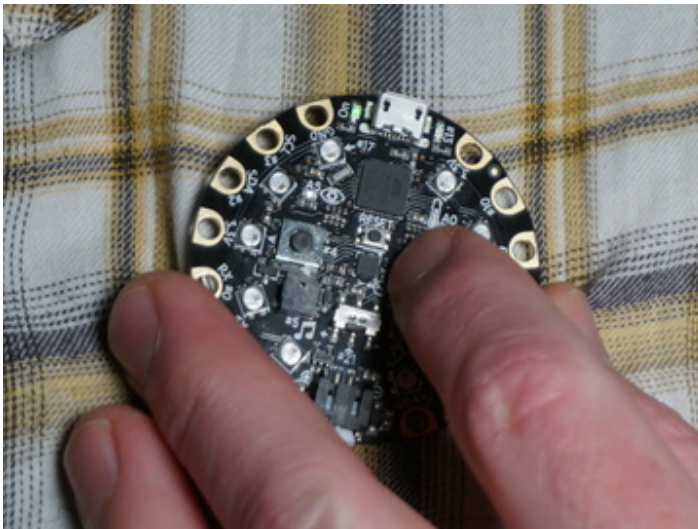
## Locating and Activating

Somewhere below your shoulder seems to work best.





The best location is in the general area below your shoulder.



Now sit up straight. Press either button to set the target angle. Two beeps should sound. **The slouch detector is now activated!**

# Questions and Code Challenges

The following are some questions related to this project along with some suggested code challenges. The idea is to provoke thought, test your understanding, and get you coding!

While the sketches provided in this guide work, there is room for improvement and additional features. Have fun playing with the provided code to see what you can do with it.

## Questions

- What are good values for **SLOUCH\_ANGLE** and **SLOUCH\_TIME**?
- Will the alarm sound if the wearer leans back too far?
- In the v3 code, what prevents slouchStartTime from constantly being reset?

## Code Challenges

- Use NeoPixels instead of the speaker to create a silent alarm.
- Use the slide switch to choose between silent or audible alarm.