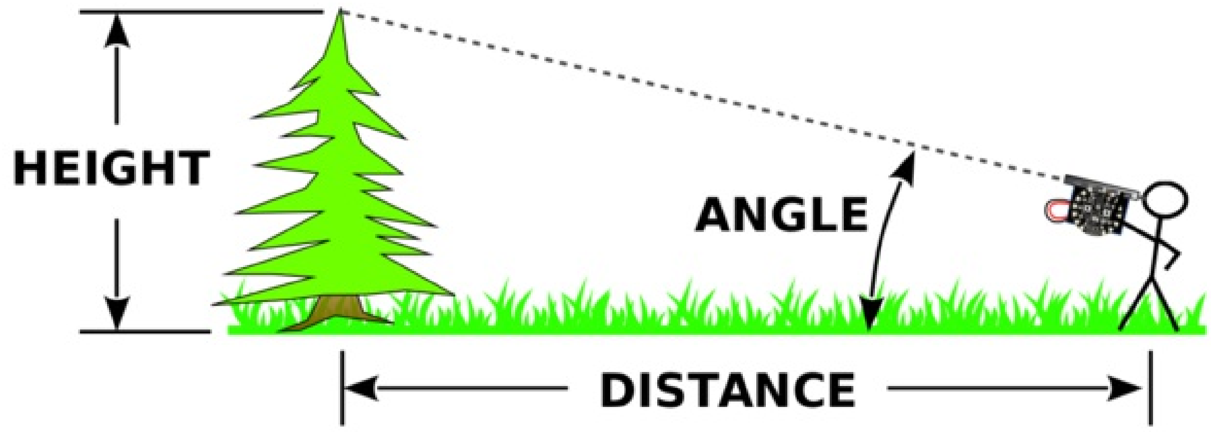
How Tall Is It?

Created by Carter Nelson

Last updated on 2016-11-11 04:40:33 PM UTC

Guide Contents

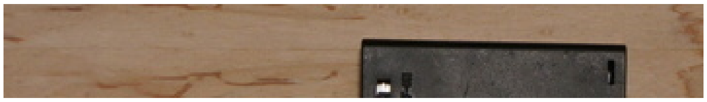
Overview



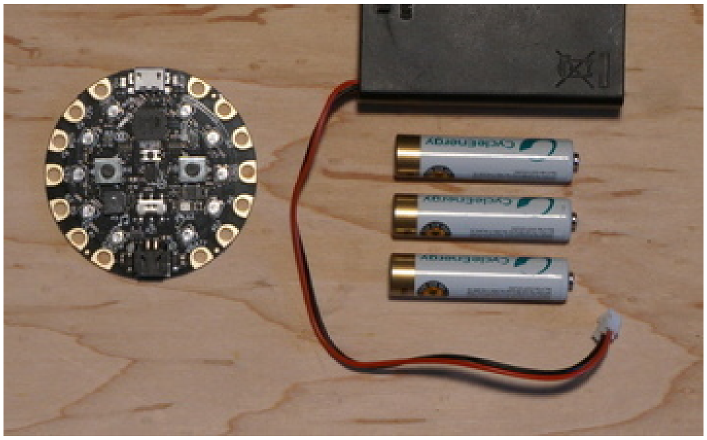
This is a fun and simple beginner project that turns the Circuit Playground into an inclinometer. An inclinometer is a device that can determine the angle of something relative to the horizontal. With this information and some right angle math, the height of tall objects can be determined. Don't worry, all the math is provided and use of a calculator is allowed.

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)

- 
- 3 x AAA Batteries (NiMH work great!)

Other Items

To build the inclinometer, you will need a few other items. A plastic tube is used to create a sight. This could be a drinking straw, the barrel of an old writing pen, or anything similar. Some rubber bands are used to hold the sight to the battery pack. Double backed tape provides a secure way of mounting the Circuit Playground to the battery pack.

-
- Plastic straw or other tube
- Rubber bands
- Double backed tape

Before Starting

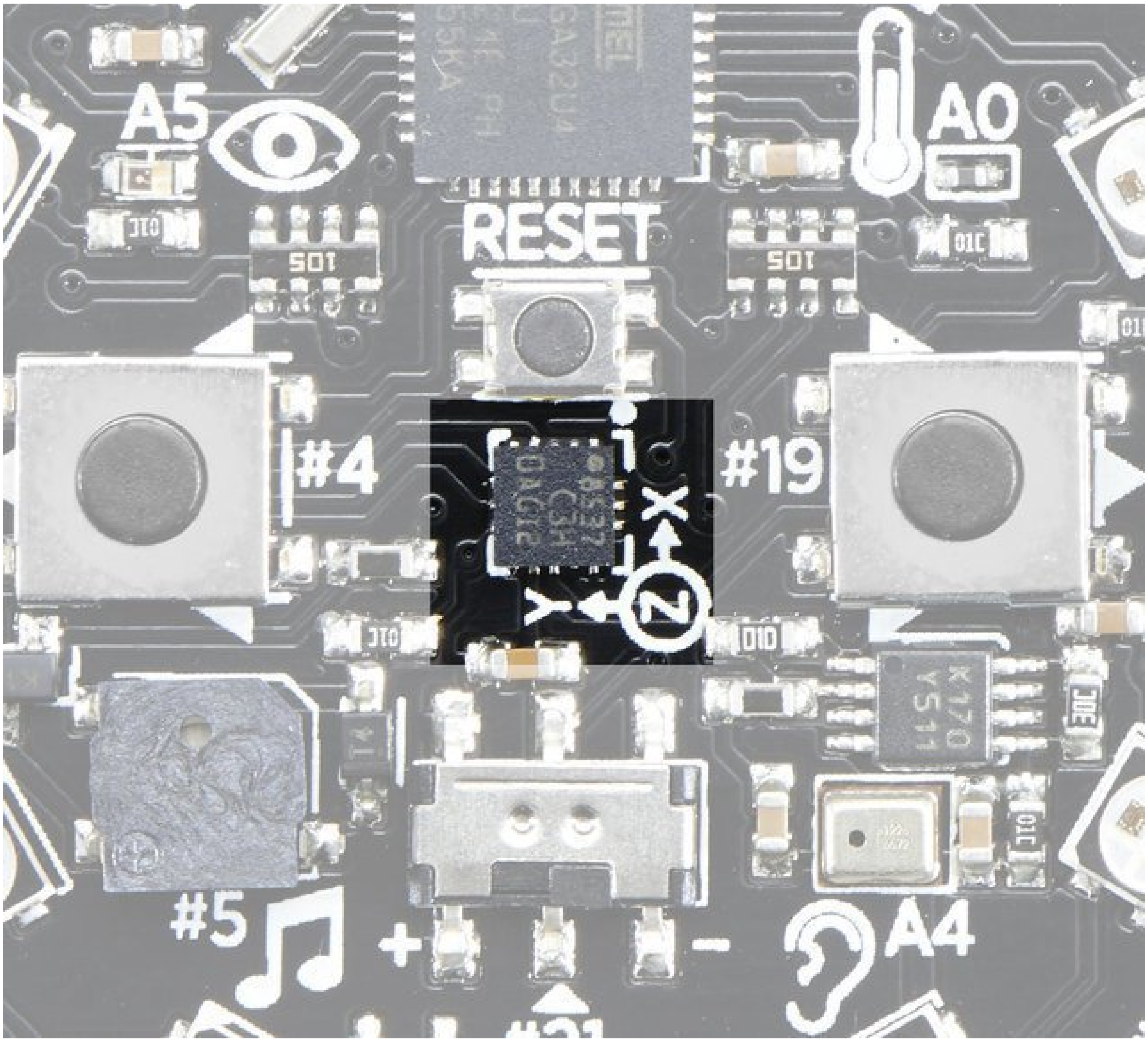
If you are new to the Circuit Playground, you may want to first read these overview guides.

- Overview (http://adafru.it/ncG)
- Lesson #0 (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 Accelerometer

Since this project uses the accelerometer, let's start by exploring how it works and behaves. You can read some technical details in the Lesson #0 Guide (http://adafru.it/s5A). And for reference, here is where the accelerometer is located on the Circuit Playground.



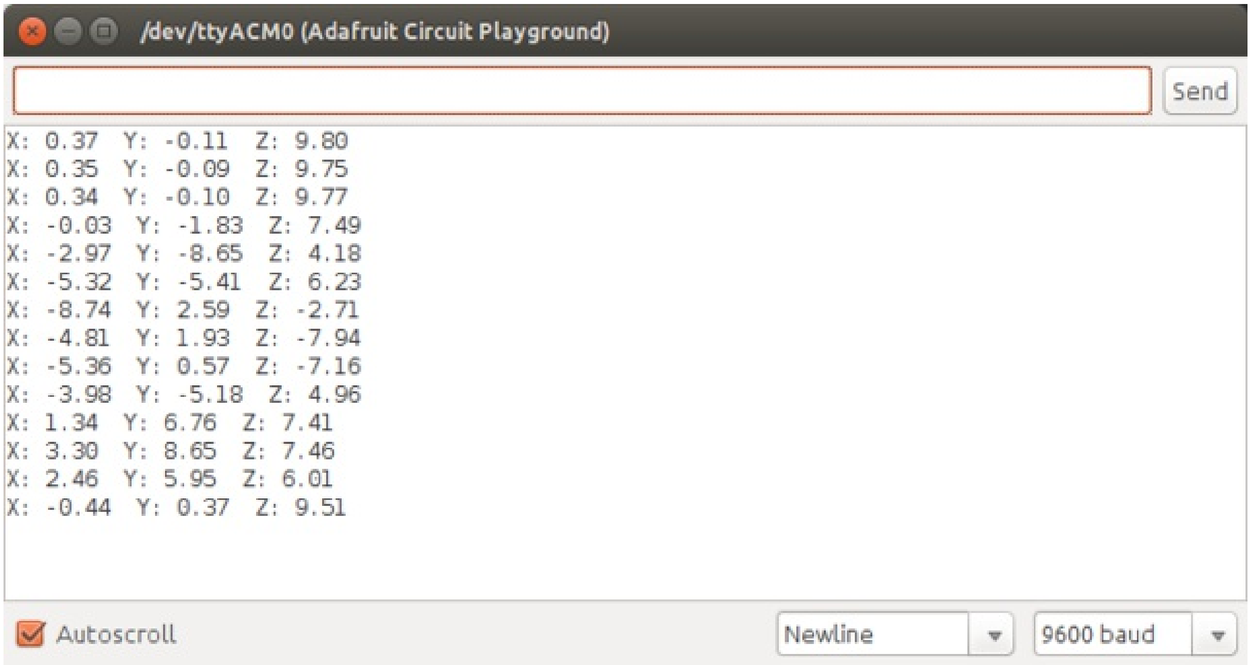
There is an example sketch included with the Circuit Playground library that we can use to play around with the accelerometer. It can be found in the following location:

File -> Examples -> Adafruit Circuit Playground -> Hello_CircuitPlayground -> Hello_Accelerometer

With this sketch loaded and running on the Circuit Playground, open the Serial Monitor.

Tools -> Serial Monitor

The current values from the accelerometer will be printed once a second.



```
X: 0.37  Y: -0.11  Z: 9.80
X: 0.35  Y: -0.09  Z: 9.75
X: 0.34  Y: -0.10  Z: 9.77
X: -0.03  Y: -1.83  Z: 7.49
X: -2.97  Y: -8.65  Z: 4.18
X: -5.32  Y: -5.41  Z: 6.23
X: -8.74  Y: 2.59  Z: -2.71
X: -4.81  Y: 1.93  Z: -7.94
X: -5.36  Y: 0.57  Z: -7.16
X: -3.98  Y: -5.18  Z: 4.96
X: 1.34  Y: 6.76  Z: 7.41
X: 3.30  Y: 8.65  Z: 7.46
X: 2.46  Y: 5.95  Z: 6.01
X: -0.44  Y: 0.37  Z: 9.51
```

Try rotating the Circuit Playground around in various orientations and watch how the values change. The values are in units of m/s^2 (meters per second squared). If you are doing this on planet Earth, you will see a value of 9.8 m/s^2 showing up when an axis is aligned vertically. This value will be there even if the Circuit Playground is held still and does not move. More on this later.

Another way to watch the values of the accelerometer is to use the Serial Plotter. To do this, let's first modify the code slightly. The code below changes the format of the output to work with the Serial Plotter and increases the rate at which the values are displayed.

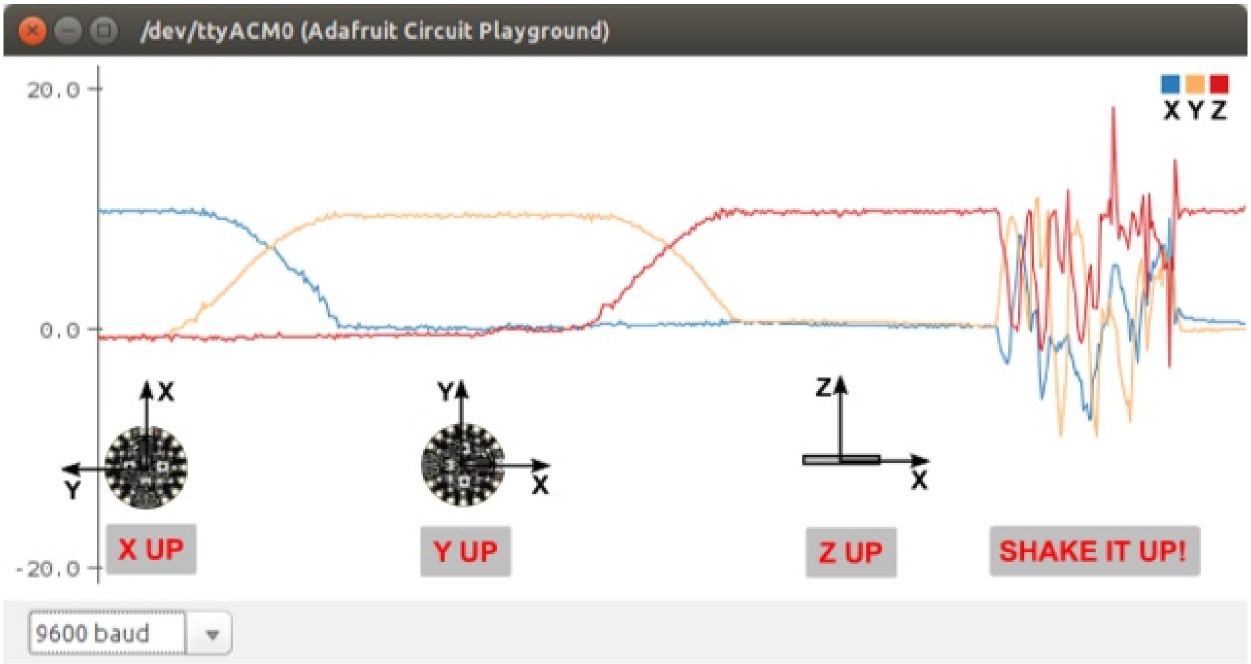
```
#include <Adafruit_CircuitPlayground.h>
#include <Wire.h>
#include <SPI.h>

float X, Y, Z;

void setup() {
  Serial.begin(9600);
  CircuitPlayground.begin();
}

void loop() {
  X = CircuitPlayground.motionX();
  Y = CircuitPlayground.motionY();
  Z = CircuitPlayground.motionZ();
  Serial.print(X);
  Serial.print(",");
  Serial.print(Y);
  Serial.print(",");
  Serial.println(Z);

  delay(100);
}
```

With this sketch loaded and running on the Circuit Playground, open the Serial Plotter.

Tools -> Serial Plotter

The accelerometer values will now be plotted like a strip chart as shown below. There should be 3 lines, one each for the X, Y, and Z values. Again, play around with rotating the Circuit Playground and watch the values change.

In the example below, the Circuit Playground was slowly rotated so that each axis aligned with the vertical, held there for a bit, and then rotated again. Finally, it was given a bit of a shake.



9600 baud

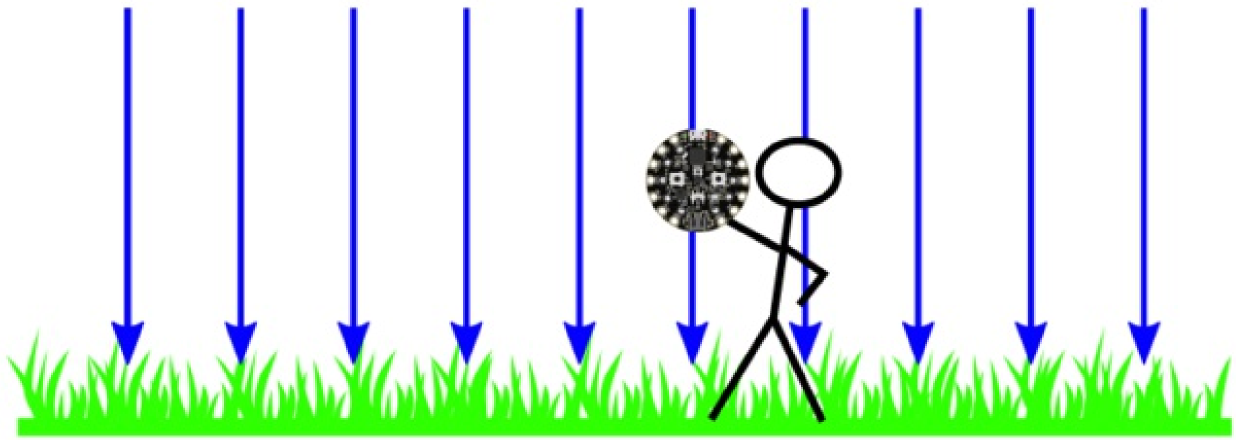
Gravity and Acceleration

As the name implies, an accelerometer measures acceleration. Acceleration happens when the velocity of something changes, like a Circuit Playground at rest which is suddenly moved. A Circuit Playground sitting still on your desk is obviously not accelerating. So why does the accelerometer still read 9.8 m/s^2? The answer is gravity.

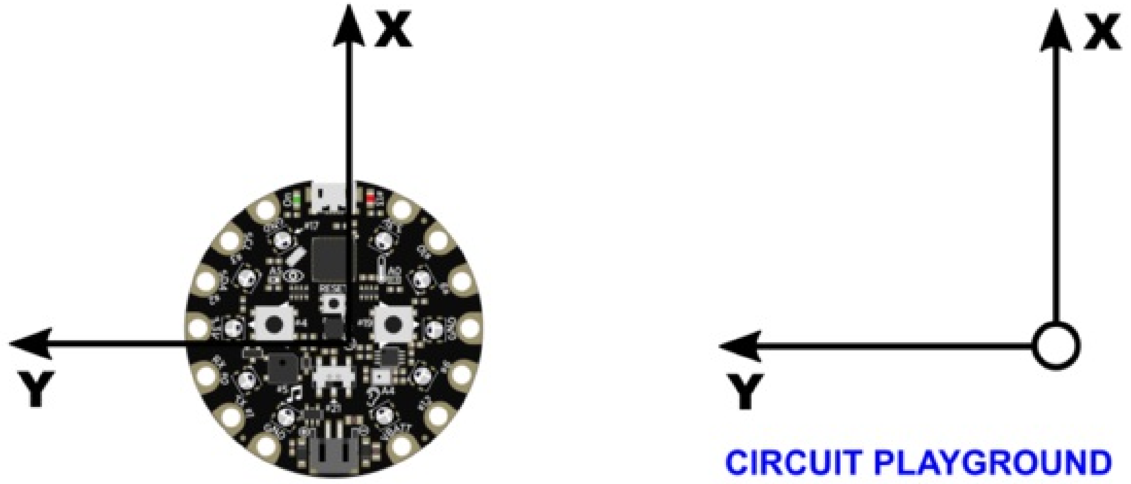
Gravity is what makes apples fall from trees and holds us to the Earth. If you looked at the planet from far away, you could think of gravity looking something like the figure below. All the arrows point in towards the planet. That's how you can stand at the South Pole and not fall off.

If we zoomed way in to just the local area where you might be, say standing outside with a

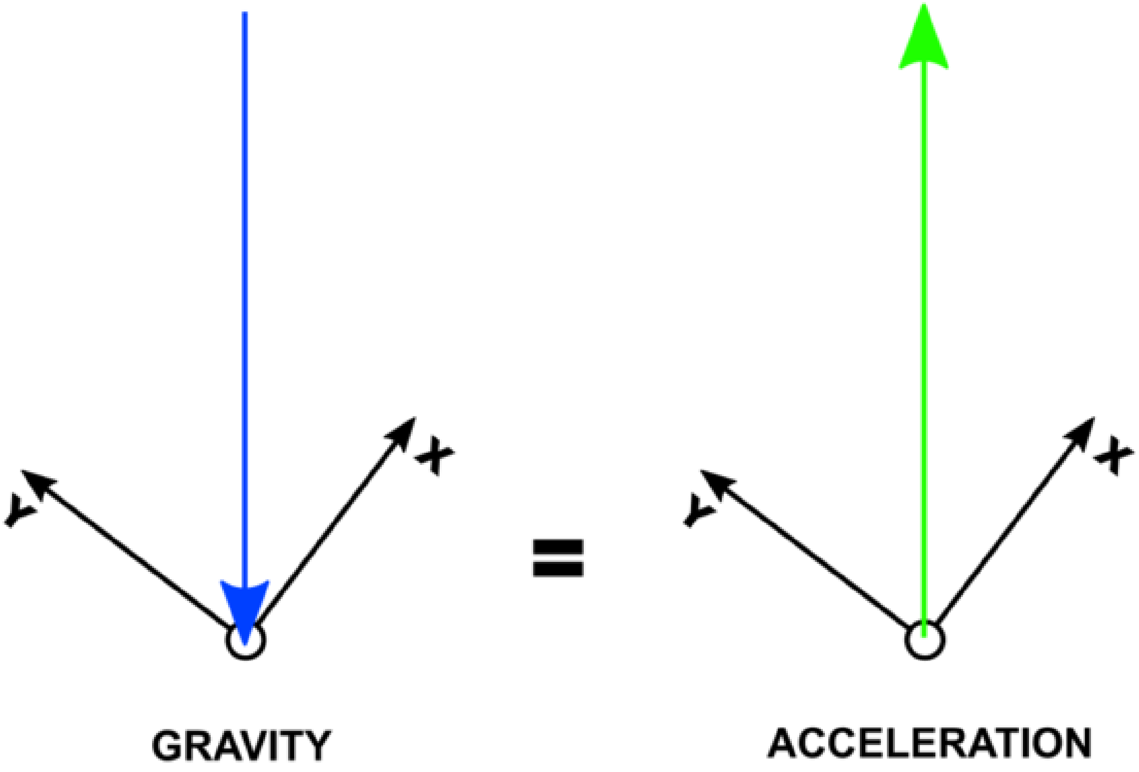
Circuit Playground, it would look more like the figure below. The Earth is so big that close up it looks flat. Now all the lines are straight and point down.



Just like gravity happens in a direction (down), acceleration also happens in a direction. In order to determine the direction, the Circuit Playground uses an XYZ coordinate system. For reference, there's a little diagram printed on the silk screen next to the accelerometer chip. A larger version is shown in the figure below. Z points out.



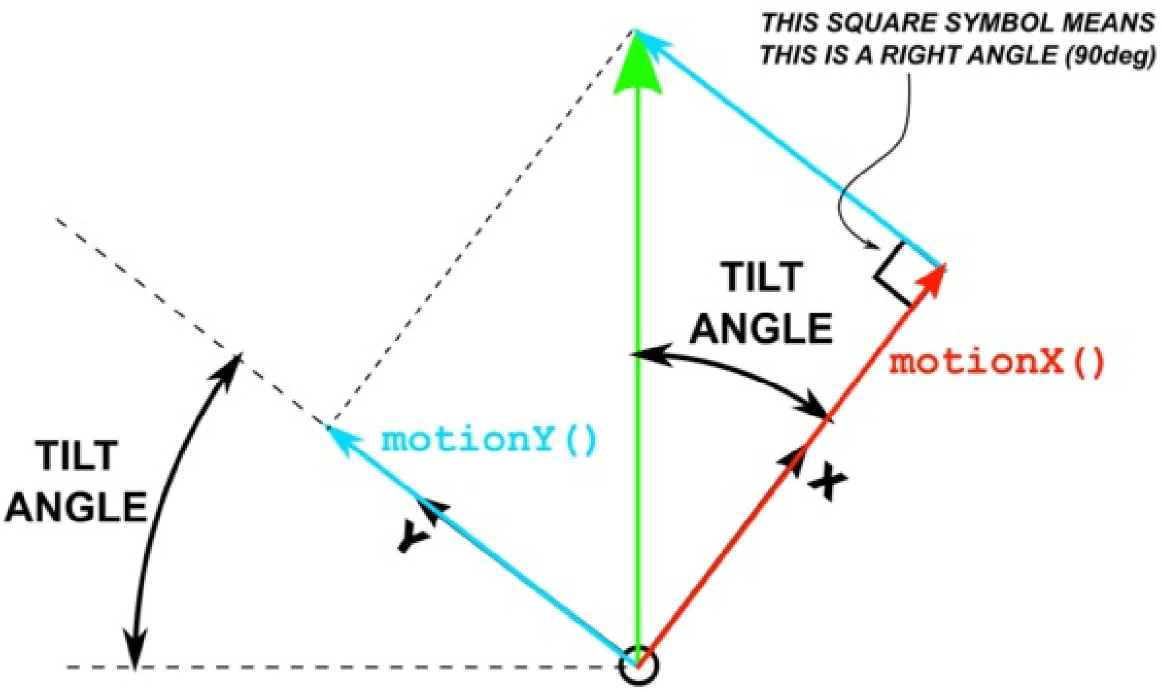
So let's say you held the Circuit Playground at a little tilt angle as shown by the coordinate system below. Gravity, as shown by the blue arrow, will still point down. In this case, the accelerometer will sense some of it in the X direction and some of it in the Y direction. However, from the point of view of the Circuit Playground, it feels like it's accelerating in the direction of the green arrow. You can test this yourself by placing the Circuit Playground flat on your desk. The Z axis will be pointing up and gravity will be pointing down. However, the value returned from `motionZ()`is positive. The Circuit Playground feels like it's accelerating up off your desk!



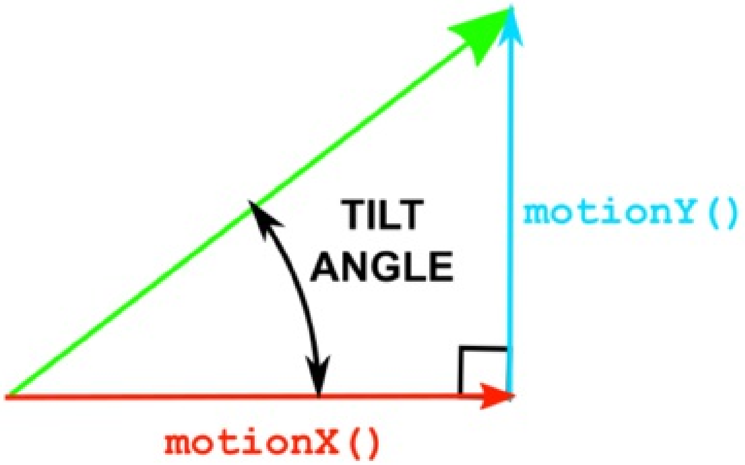
OK, now for the math. Sorry, but this is important. It's how we can use the accelerometer to measure angles and turn the Circuit Playground into an inclinometer. Also, rockets are built using math like this.

The same tilted axis and green arrow from above are shown again below. The red arrow represent the amount that the X axis will sense and returned by motionX(). Similarly, the blue arrow represents the amount the Y axis will sense and returned by motionY(). All together they form a special kind of triangle called a "right triangle", where one of the angles is a "right angle", which means it equals 90 degrees.

The tilt angle is shown in two places. The one on the left is what we are really interested in. However, it turns out it is the same as the one on the right, inside our right triangle. That's cool, since it let's us use motionX() and motionY() to compute the tilt angle.



Now let's focus on that right triangle. It is shown below with the other stuff removed and rotated to make it easier to look at. Also, the equation needed to compute the tilt angle is shown.

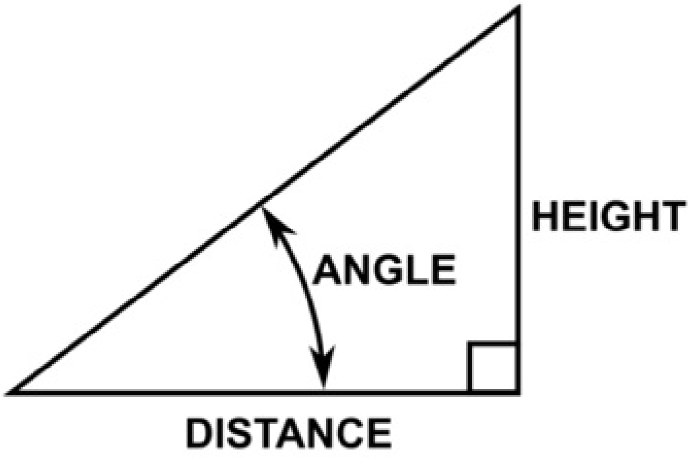


$$\texttt{TILT ANGLE} = \texttt{atan}\left(\frac{\texttt{motionY()}}{\texttt{motionX()}} \right)$$

The function **atan** is called the arctangent, or inverse tangent. It does the opposite of what the tangent function does. They are both part of a whole discipline of math called trigonometry. We will use the tangent function later when we go outside to measure the height of things. However, we will let either the Circuit Playground or a calculator do the math for us.

Measuring Height

OK, a little more math. But don't worry, this is pretty much the same thing as the previous page, just with different values. There's our friend the right triangle shown again in the figure below. This time, we want to compute the **HEIGHT**. To do so, we use the tangent function and do a little rearranging. The resulting equation is shown below.

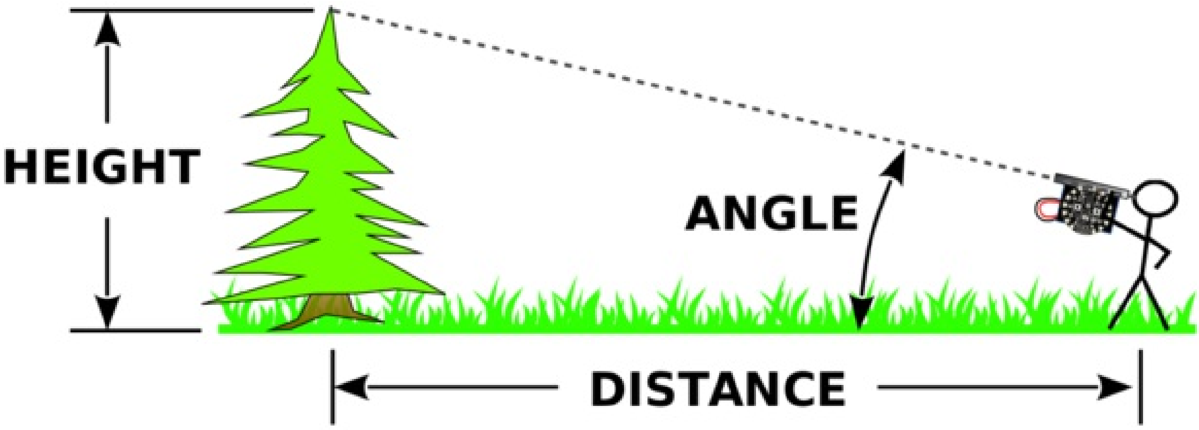


$$\tan(\text{ANGLE}) = \frac{\text{HEIGHT}}{\text{DISTANCE}}$$

MULTIPLY BOTH SIDES BY DISTANCE

$$\text{DISTANCE} \times \tan(\text{ANGLE}) = \text{HEIGHT}$$

So all we need to know is the **DISTANCE** and the **ANGLE** and we can compute the **HEIGHT**. The figure below shows how this would work for measuring the height of a tree.



We will use the Circuit Playground to give us the**ANGLE** value. We will then use a calculator to compute the tangent of that angle. There are various ways to come up with the **DISTANCE** to the object, which you will see later in some example use cases.

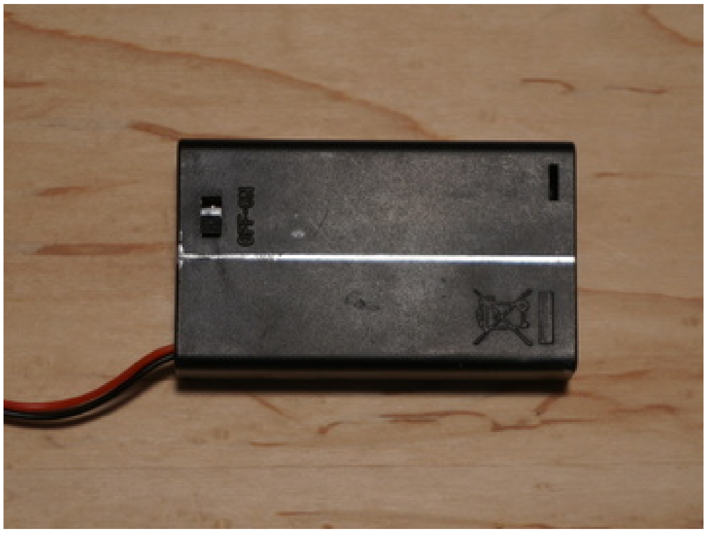
Download and print the following worksheet to help guide you through the process. It also has the decoder ring for reading the angle value off the Circuit Playground.

How_Tall_Is_It_Worksheet.pdf
http://adafru.it/s5B

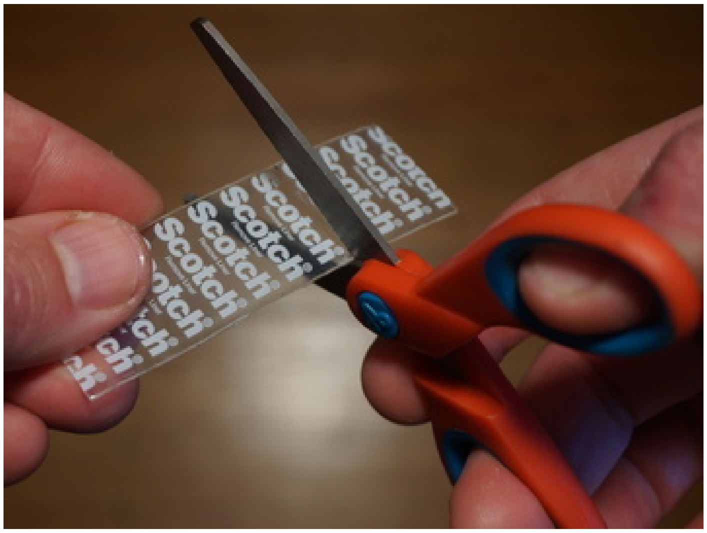
Great! Now let's turn the Circuit Playground into an inclinometer.

Assembling the Inclinometer

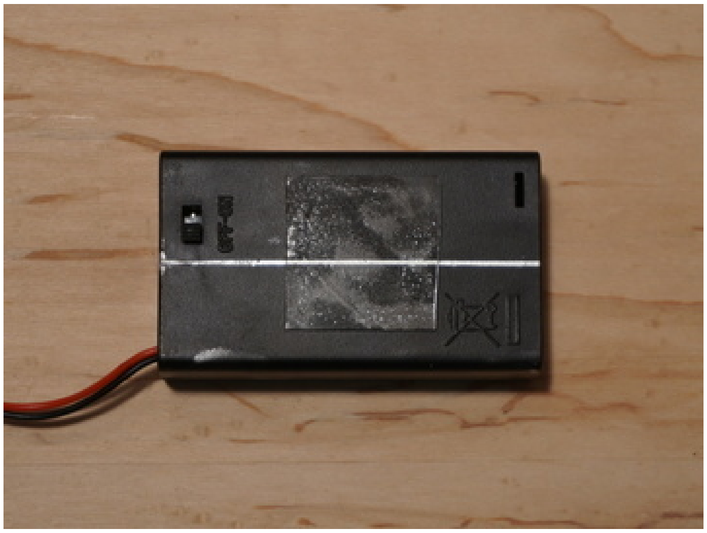
To assemble the inclinometer, channel your inner MacGyver, and follow the steps below.

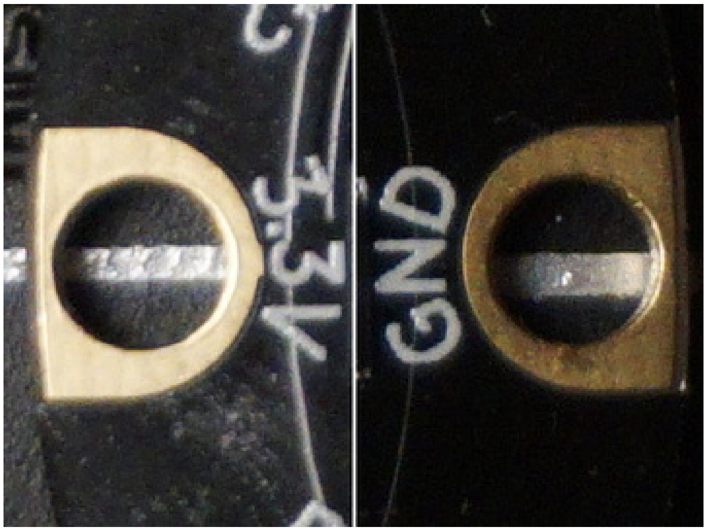
- 

  Draw a reference line 3/4" below the top of the battery pack. This will be used to help align the Circuit Playground when we mount it.

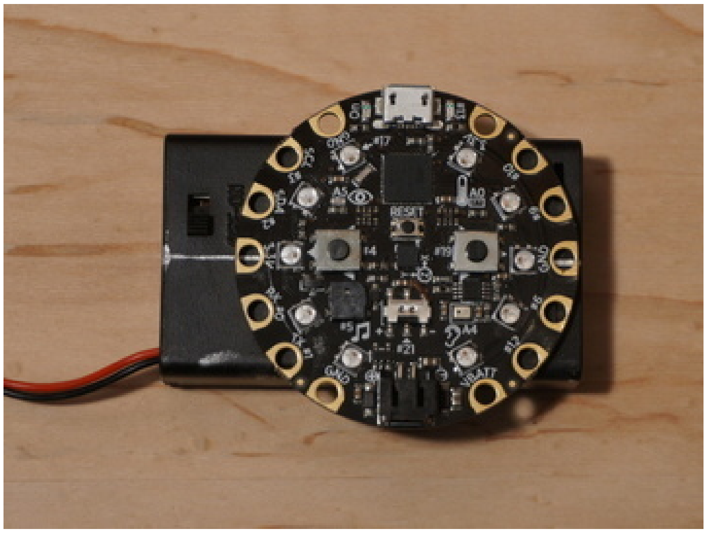
- 

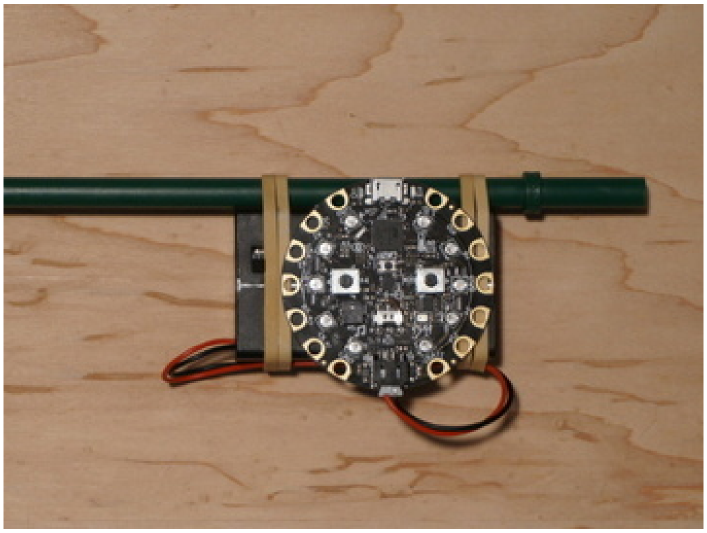
  Cut off a piece of double backed tape. About 1" x 1" is good.

- 
  Apply the double backed tape to the battery pack.

- 

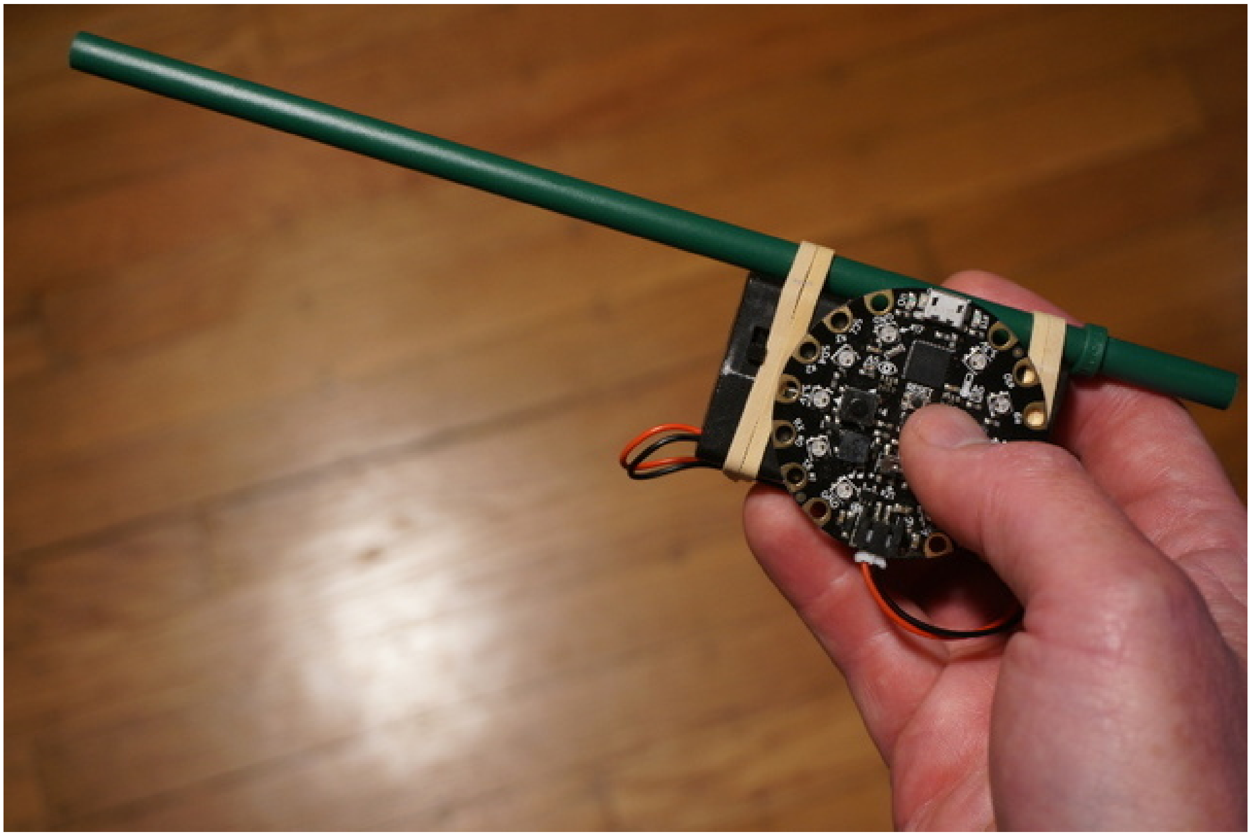
  When placing the Circuit Playground on the tape, try to keep the line centered in the 3.3V hole on the left and the GND hole on the right.

- 
  Apply the Circuit Playground to the double backed tape and press down firmly.

- 

  Attach the plastic straw to the top of the battery case using the rubber bands.

And here's the fully assembled inclinometer. Didn't use a paper clip, but still pretty MacGyvery.



Now let's load the inclinometer software.

Inclinometer Software

Here is the code to turn the Circuit Playground into an inclinometer. Copy and paste it into the Arduino IDE, load it on to the Circuit Playground, and go the next page to learn how to use it.

```
////////////////////////////////////////////////////////////////////////////////
// Circuit Playground How Tall Is It
//
// Uses the accelerometer to turn the Circuit Playground into an inclinometer.
// Can be used to determine the height of objects using a little right angle
// math.
//
// Author: Carter Nelson
// MIT License (https://opensource.org/licenses/MIT)

#include <Adafruit_CircuitPlayground.h>
#include <Wire.h>
#include <SPI.h>

float X, Y, Z;
float angle;
float totalAccel;
bool goodReading;
uint8_t angleDisplay;

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

  // Set the accelerometer range.
  CircuitPlayground.setAccelRange(LIS3DH_RANGE_2_G);

  // Indicate ready using NeoPixel 9.
  // Set it red to indicate no good reading yet.
  CircuitPlayground.setPixelColor(9, 255, 0, 0);
}

////////////////////////////////////////////////////////////////////////////////
void loop()
{
  // Only take action when either button is pressed.
  if ( (CircuitPlayground.leftButton()  == true) ||
       (CircuitPlayground.rightButton() == true) ) {

    // Average several readings.
    X = 0.0;
    Y = 0.0;
    Z = 0.0;
```

```
for (int i=0; i<10; i=i+1) {
  X = X + CircuitPlayground.motionX();
  Y = Y + CircuitPlayground.motionY();
  Z = Z + CircuitPlayground.motionZ();
  delay(10);
}
X = X / 10.0;
Y = Y / 10.0;
Z = Z / 10.0;

// Compute angle.
angle = atan2(Y, X);

// Compute total acceleration
totalAccel = sqrt(X*X + Y*Y + Z*Z);

// Initially assume the reading is good.
goodReading = true;

// Check for levelness.
// Ideally Z=0, but allow a small amount of Z.
if (abs(Z) > 1.0) {
  goodReading = false;
}

// Check for motion.
// Gravity (9.8 m/s^2) should be the only acceleration, but allow a small amount of motion.
if (totalAccel > 10.0) {
  goodReading = false;
}

// Indicate if reading was good.
if (goodReading == true) {
  // Green light.
  CircuitPlayground.setPixelColor(9, 0, 255, 0);
} else {
  // Red light.
  CircuitPlayground.setPixelColor(9, 255, 0, 0);
}

// Indicate sign of angle.
if (angle < 0) {
  // Blue light.
  CircuitPlayground.setPixelColor(8, 0, 0, 255);
} else {
  // Off.
  CircuitPlayground.setPixelColor(8, 0, 0, 0);
}

// Display angle magnitude, in degrees, on NeoPixels 0-7 as 8 bit value.
// 1 = NeoPixel ON, 0 = NeoPixel OFF
// First, convert the angle to degrees and make integer.
angleDisplay = uint8_t(abs(angle * 57.29578));
// Now display one bit at a time
```

```
    for (int p=0; p<8; p=p+1) {
      // Is the least signficant bit a 1?
      if (angleDisplay & 0x01 == 1) {
        // Turn on the NeoPixel
        CircuitPlayground.setPixelColor(p, 255, 255, 255);
      } else {
        // Turn off the NeoPixel
        CircuitPlayground.setPixelColor(p, 0, 0, 0);
      }
      // Shift the value down to the next bit.
      angleDisplay = angleDisplay >> 1;
    }

  }
}
```

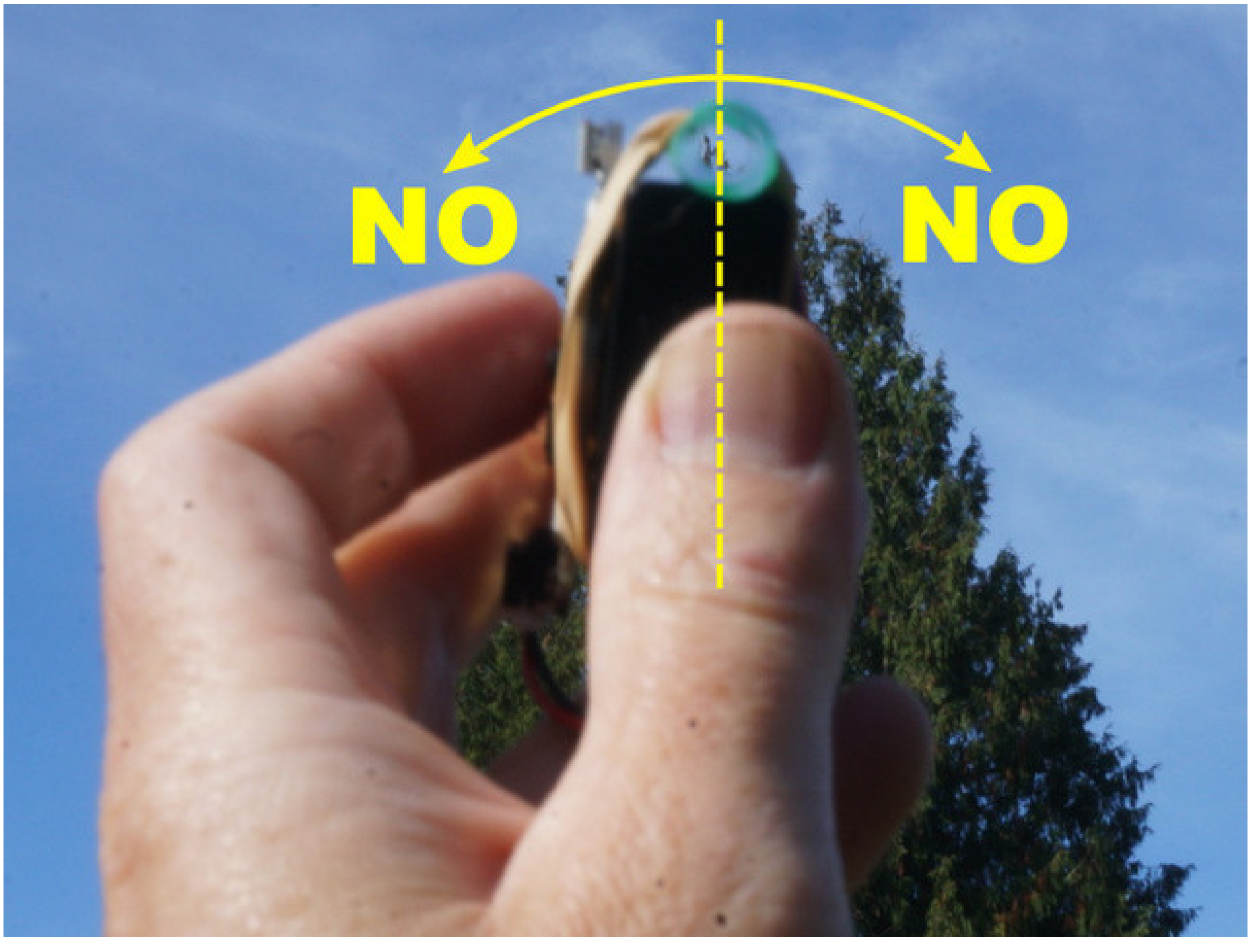
Using the Inclinometer

With the inclinometer sketch loaded and running on the Circuit Playground, you can now use it to make angle measurements. The general steps are:

1. Sight in the top of the object through the plastic tube.
2. Hold the Circuit Playground upright and still, then press either button to take a reading.
3. The results will be shown on the NeoPixels #0-#8.
4. If the #9 NeoPixel is red, try again. The Circuit Playground was either not upright or there was too much motion.

Hold It Upright

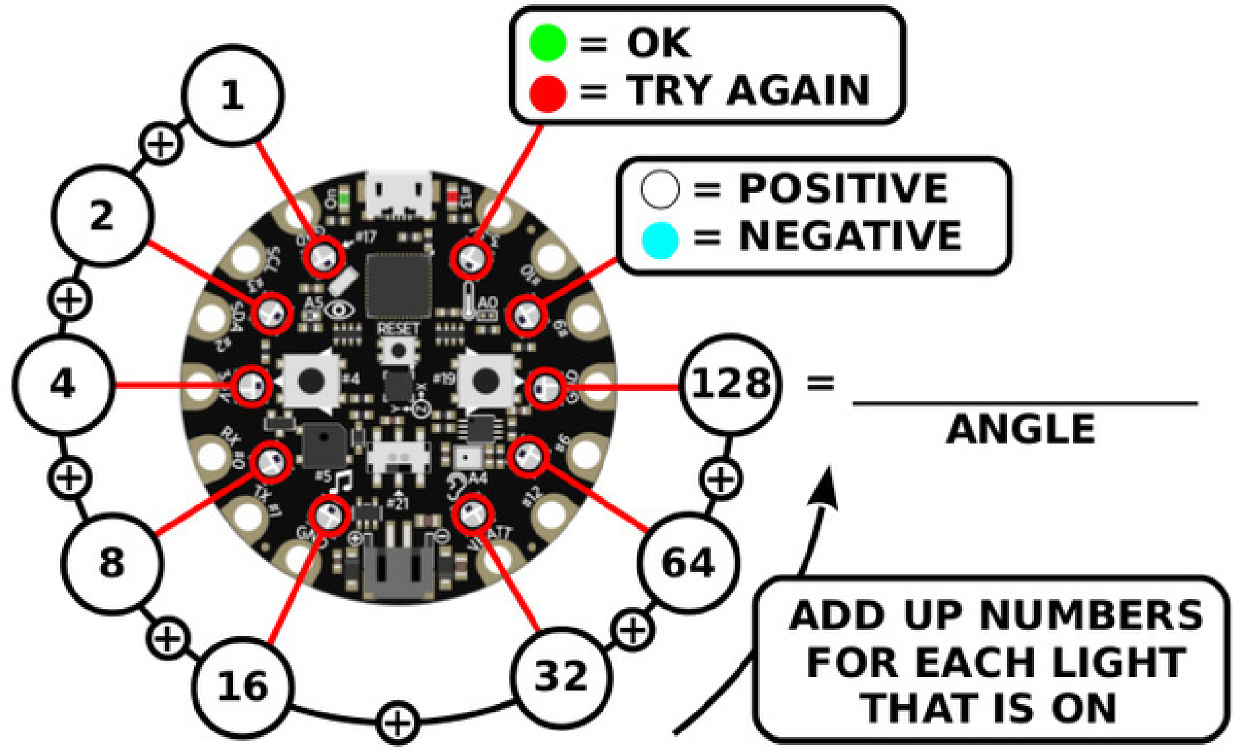
In order for the angle measurement to be accurate, gravity should be sensed only in the X and Y directions. This is done by holding the Circuit Playground upright as shown in the figure below. Don't let it tilt too much to the left or right.



Reading the Angle

If there were some kind of text display on the Circuit Playground, we could use that to display the angle directly. However, there isn't one, so we need to do something else. The approach taken here is to use the first 8 NeoPixels, #0-#7, to indicate the magnitude of the angle. The sign (+ or -) is indicated on the #8 NeoPixel.

The [worksheet](http://adafru.it/s5C) (http://adafru.it/s5C) includes the following diagram to help determine the angle from the sequence of lit NeoPixels.



To figure out the angle, simply add up each number shown for each NeoPixel that is lit. If the NeoPixel is not lit, do not add in that number.

For example, if these NeoPixels were lit (remember numbering starts with #0):

#1, #3, and #5

We would add up the corresponding numbers:

2 + 8 + 32 = 42

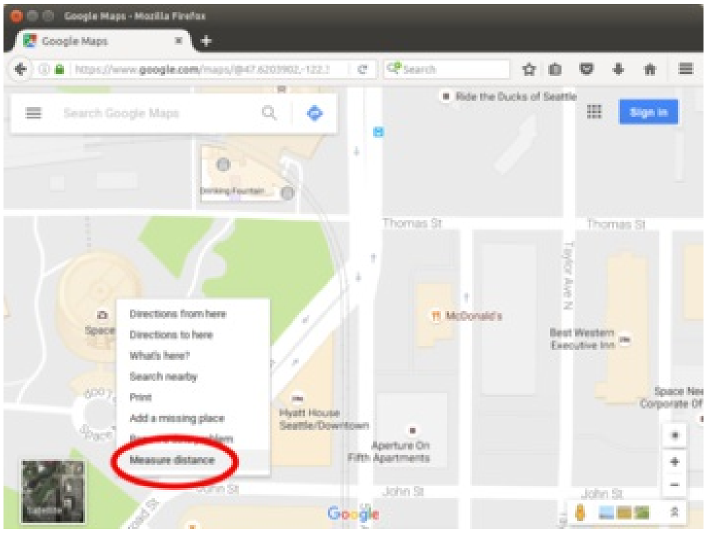
To come up with the value**42**. If the #8 NeoPixel had been lit blue, then the value would have been **-42**.

Example - Space Needle

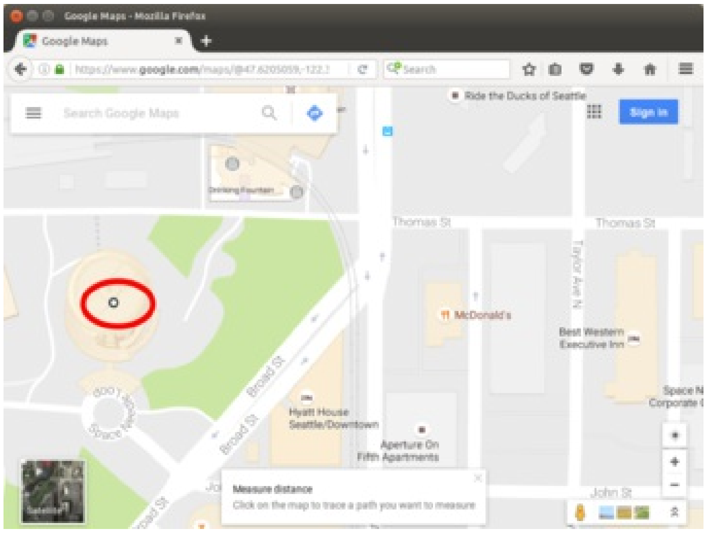
How tall is the Space Needle? Sure, The Google could tell you in an instant, but let's see what the Circuit Playground says.

First thing is to find a safe location with a good view. This street corner works.

Next we need to know the distance from here to the Space Needle. For this we will ask The Google for help. You can use Google Maps to measure the distance between two locations. Here are the steps used to determine the distance from the Space Needle to the street corner. This can be done for other locations as well.

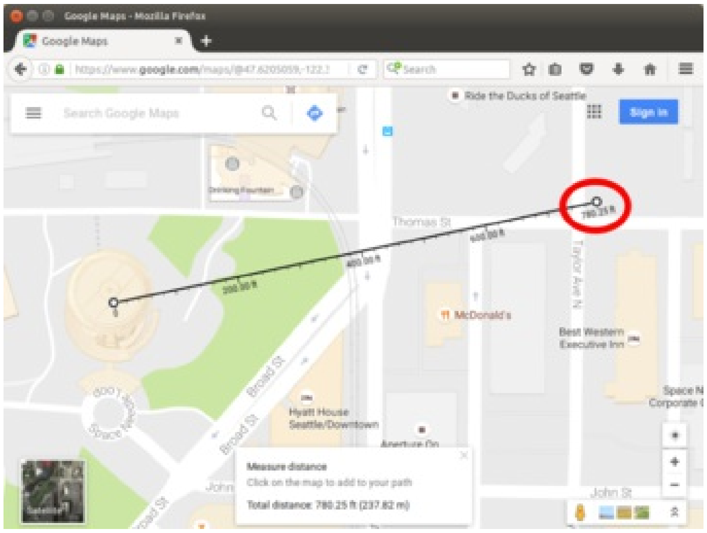
- 

Move mouse pointer to the location of the object (in this case, the Space Needle) and right click to bring up menu. Select **Measure distance**.

- 
Google Maps - Mozilla Firefox
Google Maps
Thomas St
McDonald's
Best Western Executive Inn
Broad St
Hyatt House Seattle/Downtown
Aperture On Fifth Apartments
John St
Sign in

Measure distance
Click on the map to trace a path you want to measure

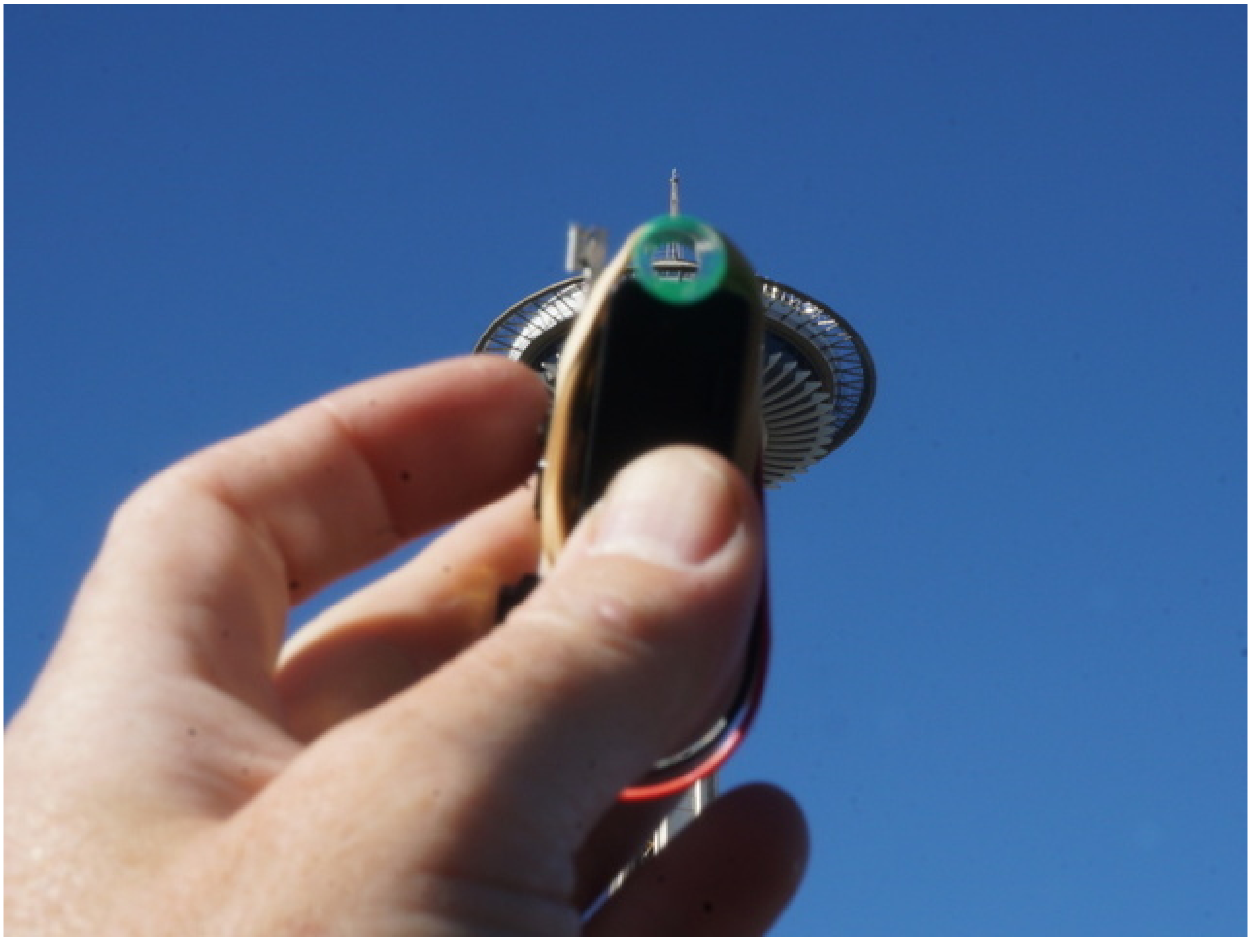
A small round circle will be placed to indicate the starting location. This can be relocated by grabbing and moving it with the mouse.

- 
Google Maps - Mozilla Firefox
Google Maps
200.00 ft
400.00 ft
600.00 ft
780.25 ft
Thomas St
McDonald's
Best Western Executive Inn
Broad St
Hyatt House Seattle/Downtown
Sign in

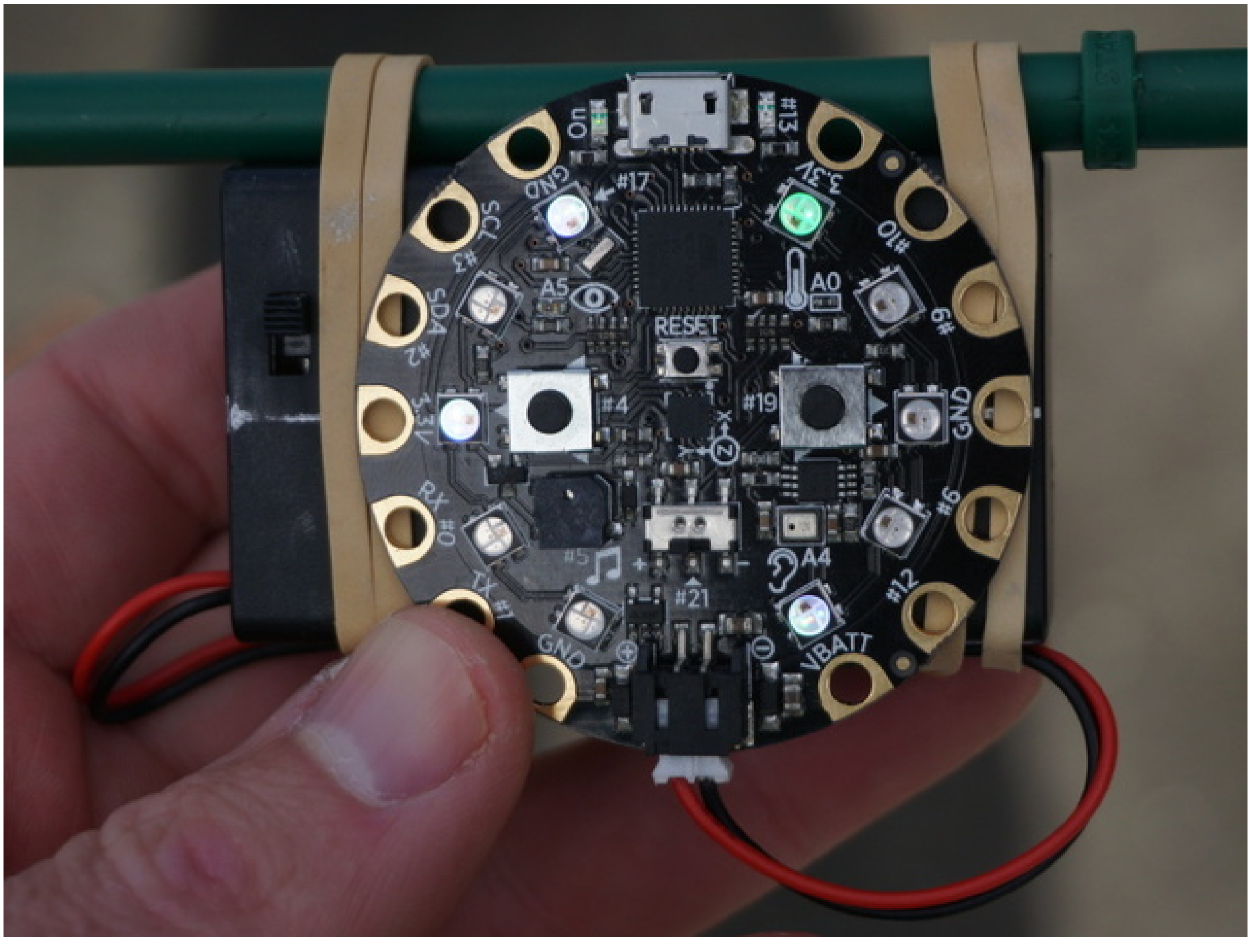
Measure distance
Click on the map to add to your path
Total distance: 780.25 ft (237.82 m)

Click on the location where you are taking the angle measurement. A line will be drawn between the two points and the distance will be shown. Rounding to the nearest foot is fine, so in this case **780 ft**.

Now we use our Circuit Playground inclinometer to measure the angle to the top of the Space Needle. Sight in the top through the plastic tube and press either button to take a measurement.



Here is the resulting display on the Circuit Playground. The green light on NeoPixel #9 means the reading is good, the #0, #2, and #5 NeoPixels are lit indicating the angle, and since the #8 NeoPixel is not lit, the value is positive.



Now we can use the worksheet (http://adafru.it/s5C) to determine what angle is being indicated. In this case:

1 + 4 + 32 = 37 degrees

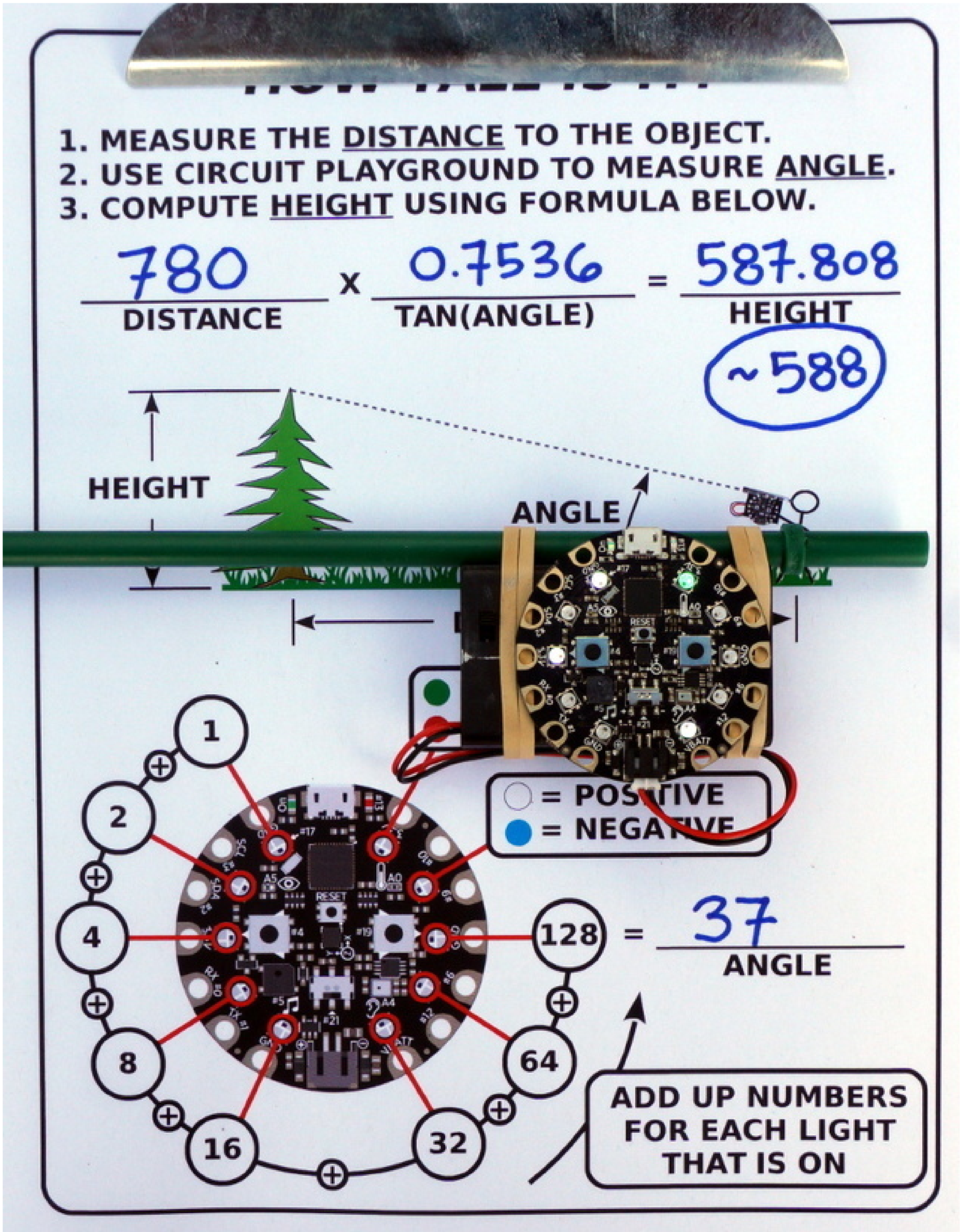
Then use a calculator to get the tangent of this angle:

tan (37 deg) = 0.7536

We determined our distance was 780 feet above, so just need to multiply to get the final answer:

780 x 0.7536 = 587.808

Let's just call it **588 feet**.


1. MEASURE THE DISTANCE TO THE OBJECT.
2. USE CIRCUIT PLAYGROUND TO MEASURE ANGLE.
3. COMPUTE HEIGHT USING FORMULA BELOW.
780 DISTANCE x 0.7536 TAN(ANGLE) = 587.808 HEIGHT
~588
HEIGHT
ANGLE
○ = POSITIVE
● = NEGATIVE
1
2
4
8
16
32
64
128
37 = ANGLE
ADD UP NUMBERS FOR EACH LIGHT THAT IS ON

The actual height of the Space Needle is **605 feet**. So our reading was off by **17 feet**. Oh well, not perfect, but not bad for a plastic straw and some rubber bands.

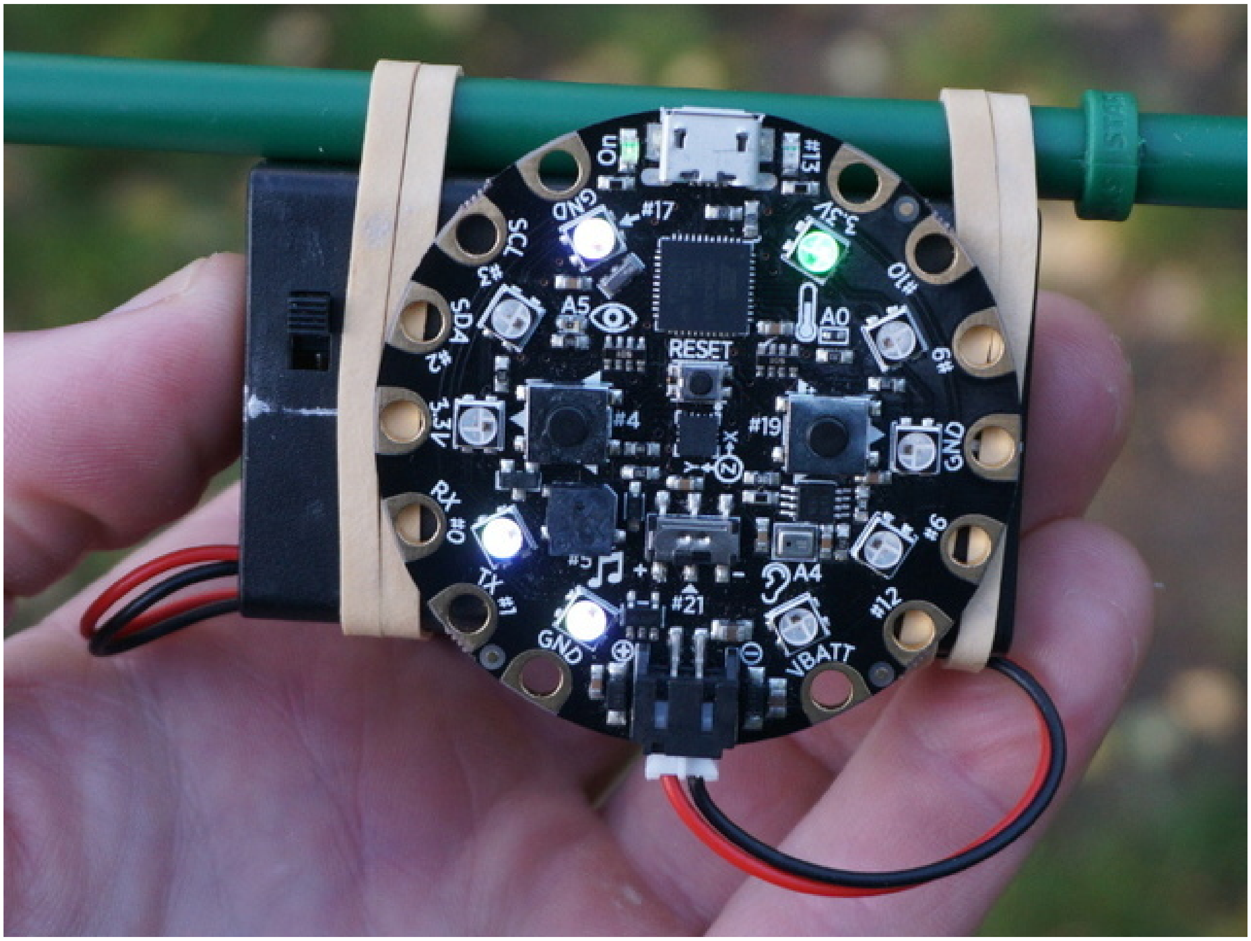
Example - A Tree

How tall is that tree? This time the Internet can't help us.

In order to determine the distance to the tree, a very simple method was used: paces. Paces are just slow deliberate steps that are as close to the same as possible each time. The actual size of a pace will be different for different sized people. For this case, the distance was **44 paces**.

Then, to get the angle, use the Circuit Playground and sight in the top of the tree. Press either button to get a reading.

Here is the resulting display on the Circuit Playground. The green light on NeoPixel #9 means the reading is good, the #0, #3, and #4 NeoPixels are lit indicating the angle, and since the #8 NeoPixel is not lit, the value is positive.



Now we can use the worksheet (http://adafru.it/s5C) to determine what angle is being indicated. In this case:

1 + 8 + 16 = 25 degrees

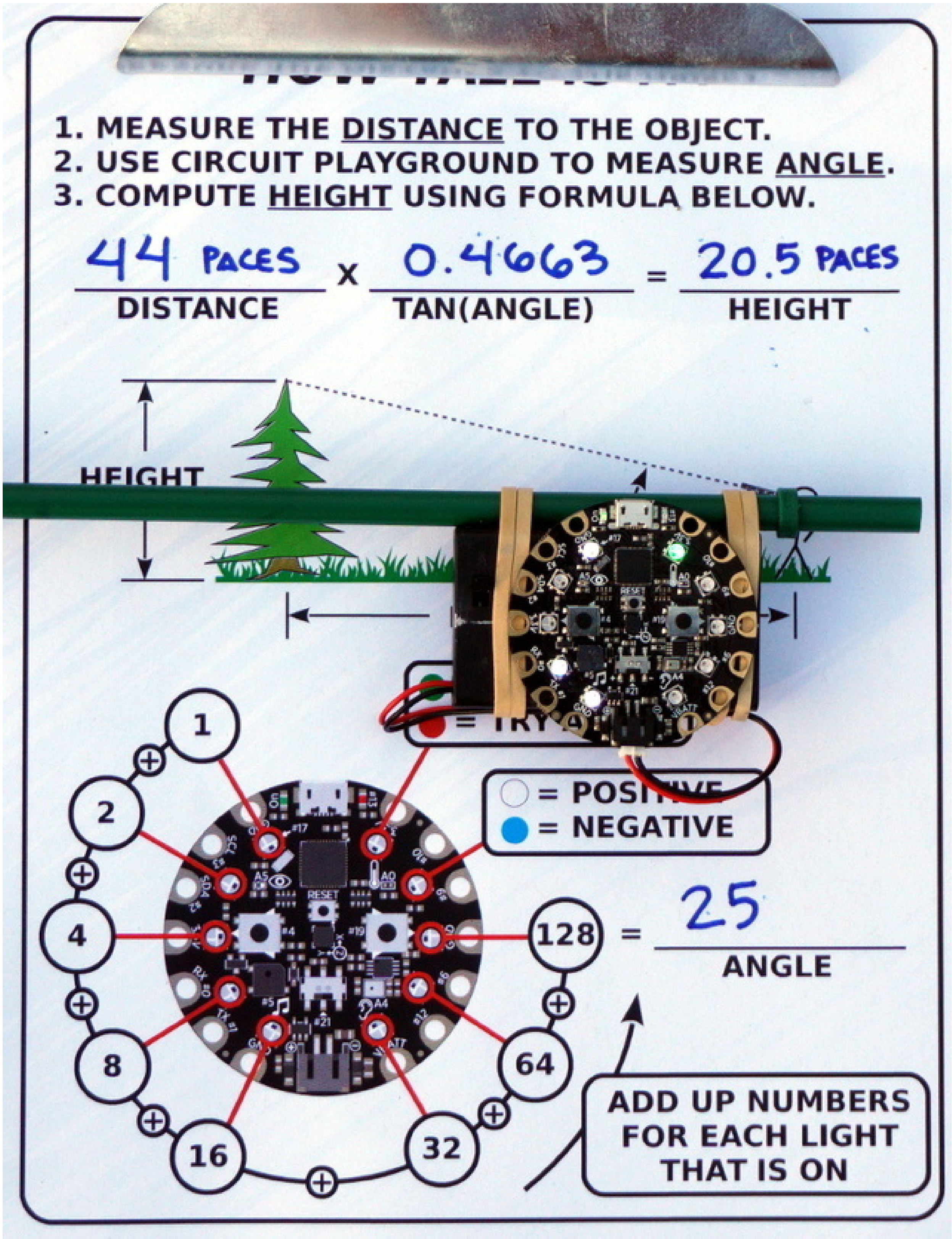
Then use a calculator to get the tangent of this angle:

tan (25 deg) = 0.4663

We determined our distance was 44 paces above, so just need to multiply to get the final answer:

44 x 0.4663 = 20.5

Let's just call it **21 paces**.



1. MEASURE THE DISTANCE TO THE OBJECT.
2. USE CIRCUIT PLAYGROUND TO MEASURE ANGLE.
3. COMPUTE HEIGHT USING FORMULA BELOW.

44 PACES x 0.4663 = 20.5 PACES

DISTANCE x TAN(ANGLE) = HEIGHT

○ = POSI
● = NEGATIVE

128 = 25 ANGLE

ADD UP NUMBERS FOR EACH LIGHT THAT IS ON

Great. But how tall is that? If you take this approach you will need to determine how big your paces are. You can place a tape measure on the ground and step next to it to get an idea. I came up with about 3 feet for my pace, which gives:

3 feet/pace x 21 paces = 63 feet

The tree is about **63 feet** tall.

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 inclinometer sketch provided works, there is room for improvement and additional features. Have fun playing with the provided code to see what you can do with it.

Questions

- Why is it important for the Circuit Playground to be upright and held still?
- Are there any issues with making the angle reading at eye level?
- Would this work on the moon?
- Would this work in space, for example, on the International Space Station?

Code Challenges

- Add an audio cue to indicate if the measurement is good or bad.
- Improve the resolution of the indicated angle measurement.
- Incorporate all the math into the Circuit Playground so it provides height directly.