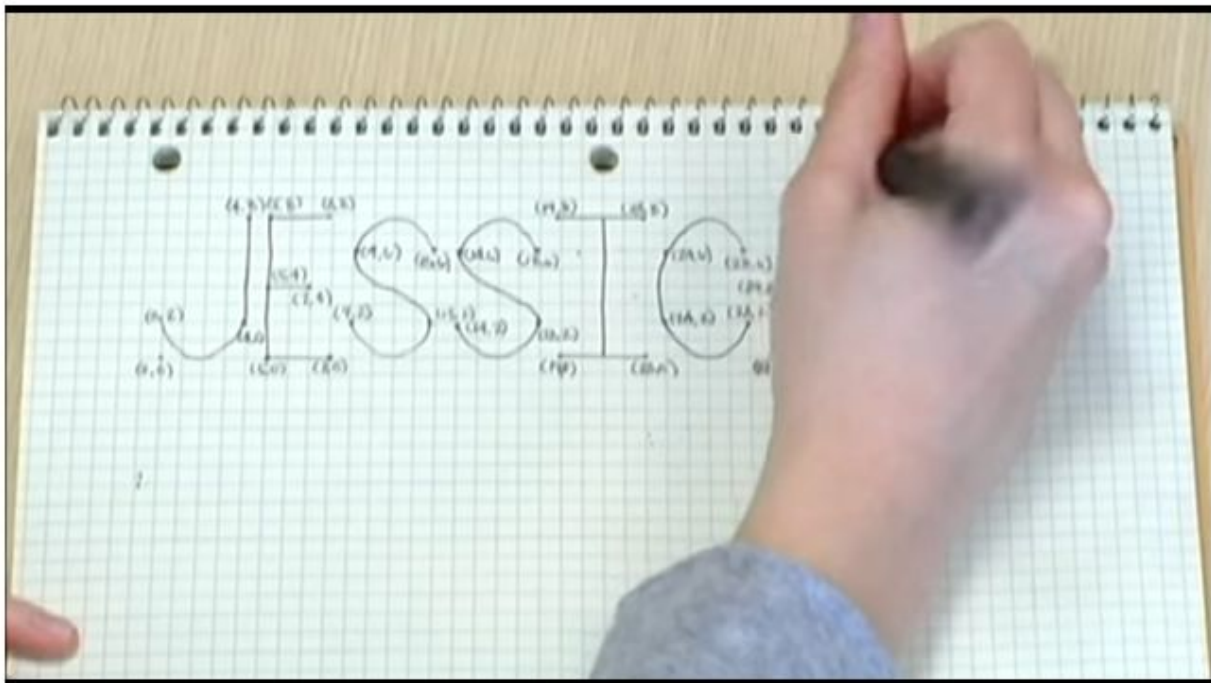


Spin Project: Scribble Your Name with the S2

Skill Level: Intermediate

Hours to Complete: 1-2

Skills Required: Spin programming, basic geometry



View this [project's YouTube video](#) on the ParallaxInc YouTube channel.

Time to brush off those geometry skills because you're going to learn how to draw your own name using the S2 Robot, some graph paper, and the `s2.spin` object written by Phil Pilgrim! Included in this object are two methods: `move_to(x, y)` and `arc_to(x, y, radius)`. As the variables `x`, `y` and `radius` suggest, these methods move the S2 Robot to a predefined coordinate location, or arc by a certain radius to a specific coordinate location.

By plotting what we would like to draw on graph paper and calculating the radius of each of the arcs in our drawing, we should be able to plot out the coordinates we want our S2 to move to, and then easily transfer those ordered pairs into code, right?

Right! Let's get started!

Starting with Spin

This tutorial does assume that you have some familiarity with programming in Spin. If you never have before, or need a refresher, it would be a good idea check out one of these great beginner tutorials before continuing:

- [Propeller Education Kit Labs](#) – Written by Andy Lindsay, these labs are a great introduction to the Propeller microcontroller. Previous electronics and programming experience is required.
- [Programming & Customizing the Multicore Propeller Microcontroller: Official Guide \(#32316\)](#) – This book published by McGraw-Hill contains an introduction to the Propeller chip's architecture and Spin programming language from Parallax's Jeff Martin, as well as debugging techniques, and sensor interfacing by Andy Lindsay.

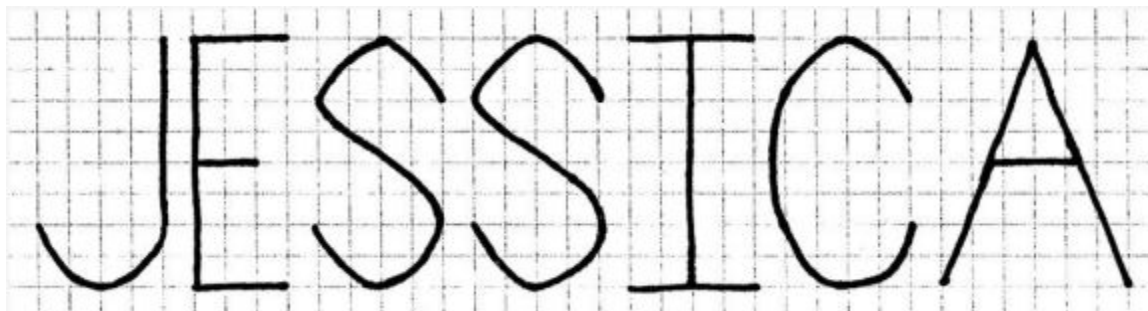
What's Required

- S2 Robot (not written for the S3)
- Graph paper
- Poster board
- Large Sharpie Marker
- s2.spin object (included with this project download, for S2 only)

Plotting Your Name

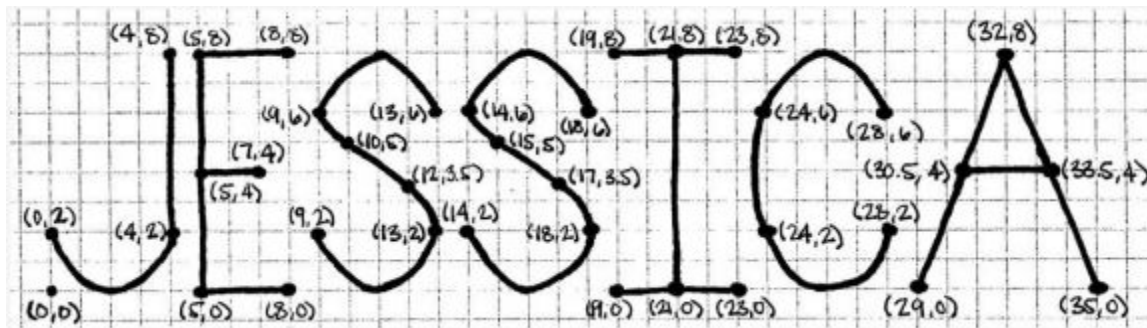
The first step is to plot your name on a sheet of graph paper. As you're drawing, be sure to keep in mind that this will serve as a template for what we'll be programming the S2 to draw later. To keep things simple, we'll want the ends of all of our letters to end at cross-sections on the graph paper, and we'll want each letter to be spaced evenly apart.

I found that keeping the letters in my name eight blocks in height and four in width with a one square spacing in between worked out pretty well. When you're done your name should look something like this:



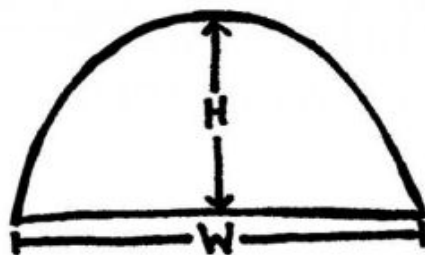
Next, we'll need to define a origin, point (0,0). Again to keep things simple, I decided to skip dealing with negative numbers, and defined my origin to align with the starting point of the first letter of my name, where no other letters would cross below.

After the origin is defined, we're ready to map out the major coordinate locations. These should be points where the S2 will begin, end or change direction on a straight path; or begin, end or change arc radius. After, your name should look something like this:



Note: As you are plotting your ordered pairs, try to think about how you S2 will move as it's scribbling your name. Where will it need to change direction? How will it move on to the next letter? By keeping these questions in mind, you'll more easily be able to pinpoint the major coordinate locations in each letter of your name.

Since the method `arc_to(x, y, radius)` requires us to provide the radius of the arcs we'd like to draw, we need to calculate these radii. The formula for calculating the radius of an arc with a known width and height is:

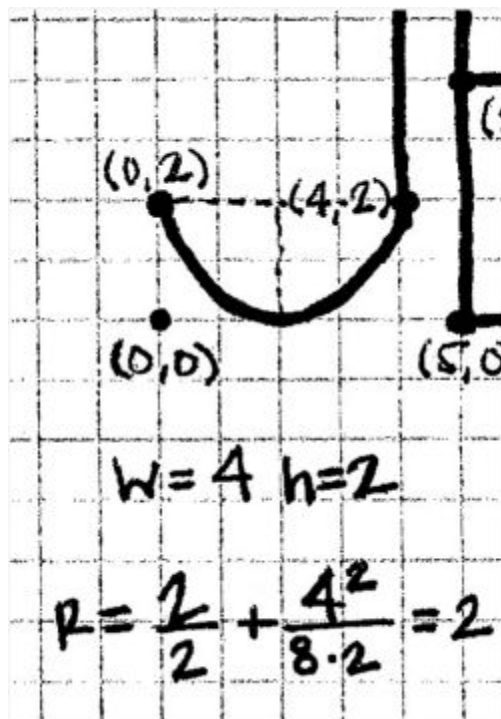


W = width of the base of the arc

H = height measured at the midpoint of the base

$$\text{radius} = \frac{H}{2} + \frac{W^2}{8H}$$

We can then apply this formula to letters in the name. For example, the arc in the letter J has a width of 4 and a height of 2, so its radius would be calculated as follows:



Scaling Letters

According to the documentation for the `move_to` and `arc_to` methods, each unit of movement is approximately 0.5 mm. With the letters in my name being about 8 blocks high, this means that when the S2 draws them, they'll only be about 4 mm in height.

This isn't very big, so for my final code, I decided to scale the letters so they'd be almost 5 inches in height. To do this, I multiplied each ordered pair and each radius length by 30.

Changing from Points to Code

In addition to `move_to` and `arc_to`, we'll need to call some other methods from the `s2.spin` object in order to set some parameters. Remember, we'll first have to set our clock speed and declare the `s2.spin` object:

```

CON

_CLKMODE = XTAL1 + PLL16X
_CLKFREQ = 5_000_000

OBJ

s2 : "S2"

```

Then, we'll need to call some methods to initialize the S2 Robot:

```

PUB Main

S2.start
S2.button_mode(true, true)
S2.set_led(s2#POWER, s2#BLUE)
S2.start_motors

' Start S2 object
' Set button mode to display in LEDs and to reset
' Turn on the power LED.
' Start motor controller

```

Next, we'll set the motor speed to half of its maximum, which is a value of 7. And then we'll need to tell the S2 Robot what its current location is using the `here_is` method. This should be the first coordinate location of the first letter in your name. For my name, this is: 0, 60 (remember that we've scaled the letters by multiplying each ordered pair by 30).

```

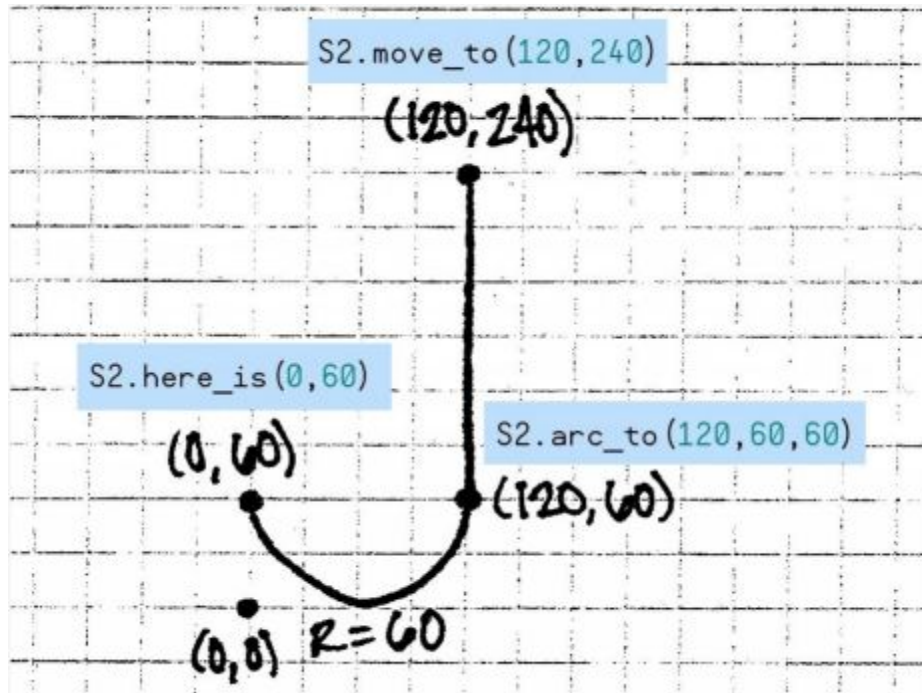
S2.set_speed(7)
S2.here_is(0,60)

' Set drawing speed to 50%
' Set current S2 coordinate location

```

Now we're ready to start porting our ordered pairs into code. When doing so, you may find that some values may need to be tweaked slightly – that's perfectly OK! It actually took a couple of iterations for me to be completely happy with my letter S's. Just keep playing around with your values until you're completely satisfied with each letter.

Here's an example of the coding sequence for the letter J:



And one last thing, by putting all of our `move_to` and `arc_to` method calls between the `begin_path` and `end_path` methods, our S2 Robot will execute smooth, connected movements.

When you're done, your name should look a little something like this:



Included with the S2 object is the source code I used for my name, although really it's not as cool to run it unless your name is Jessica too. Or I guess unless your significant other is named Jessica, or your mom, or your sister, aunt, BFF ... then you could just really impress some people.

But hopefully you can use the code as a template to code your own name.

Until next time...Happy Scribbling!