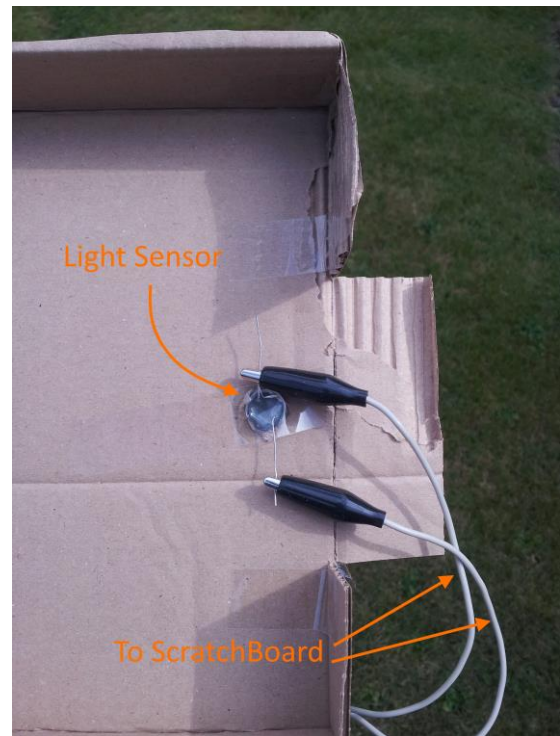


# How to use a Wind Speed Sensor

## Introduction:

The Wind Speed Sensor has a light sensor with two metal pins (legs) coming off it. To connect it first get your crocodile clip lead and plug it into one of the external sensor sockets of the Picoboard / Scratchboard. For this worksheet I'll assume you plugged it into socket A. Then take these crocodile clips and attach them to the two different legs of the light sensor. You have now successfully connected your sensor to the Picoboard, now let's learn how to use it.



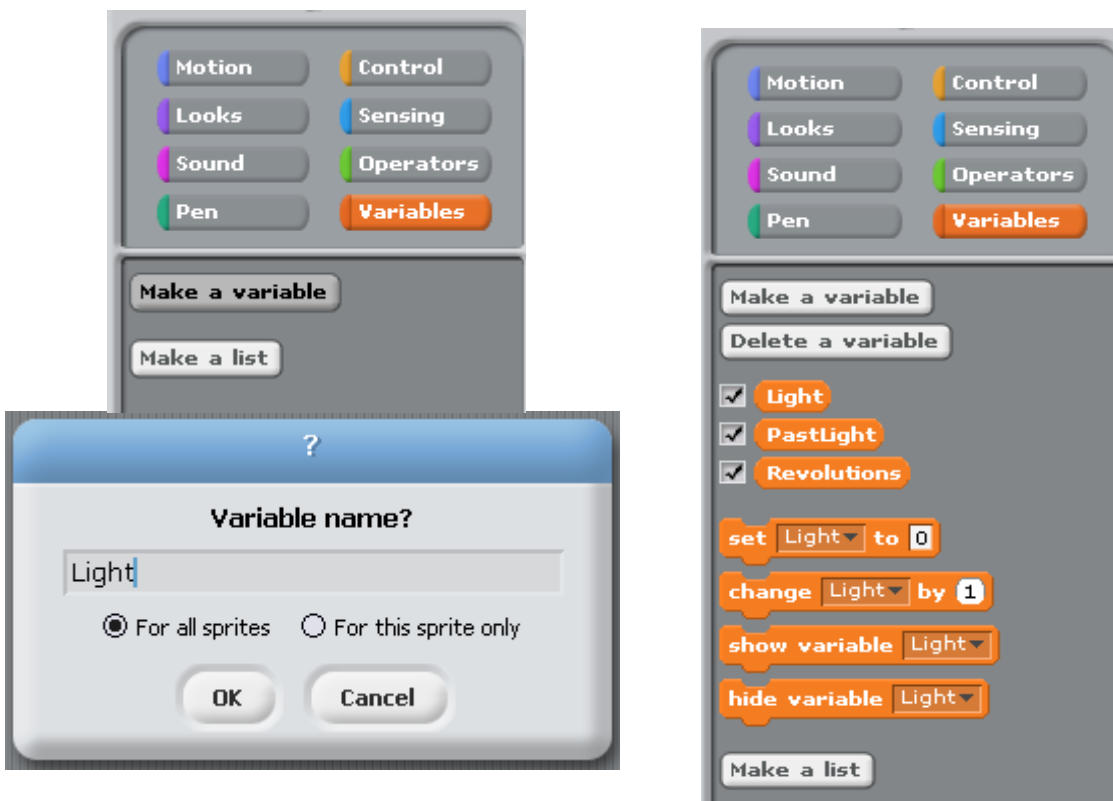
## How It Works:

The light sensor attached to your Wind Sensor gives a value between 0 and 100 depending on how much light falls on it. We use our shade wheel on top of the light sensor to limit the amount of light it receives depending on what colour is on top of the light sensor. So the darker the colour is above the sensor the more light we block from it and hence the lower the number it returns. Therefore black returns a low number, white a high number and a shade in between returns a number in between the black and white numbers. In this way we can detect when the shade wheel changes from white to black and hence detect when a complete revolution of the cups has been made. By measuring the number of revolutions in a minute we can see how fast the wind is going.

**Safety:** Please note that you use these resources at your own risk. Correct use of some components requires care.

**Step 1:** Start Scratch and get the Picoboard working correctly with Scratch

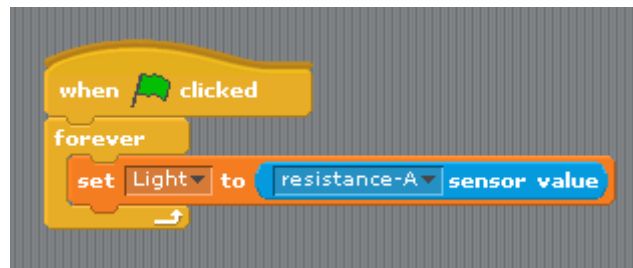
**Step 2:** Now make three new variables, one called Light (we will set this to contain the 0-100 light sensor value mentioned above), another called PastLight (which holds which side of the wheel we saw last with our light sensor) and the other called Revolutions (which will be the number of rotations the cups have made). To create a variable go to the Data tab (Variables tag in Scratch V1.4), click 'Make a Variable', type in a name and make sure 'For all sprites' is selected, then click OK (as demonstrated in pictures blow).



**Step 3:** Make sure that the Light, PastLight and Revolutions variables have ticks next to their names as in the picture to the right above.

**Safety:** Please note that you use these resources at your own risk. Correct use of some components requires care.

**Step 4:** Create the following block in the Stage sprite. You will find the blocks in 'Events', 'Control', 'Data' and 'More Blocks' tabs. ('Control', 'Sensing' and 'Variables' in Scratch V1.4)



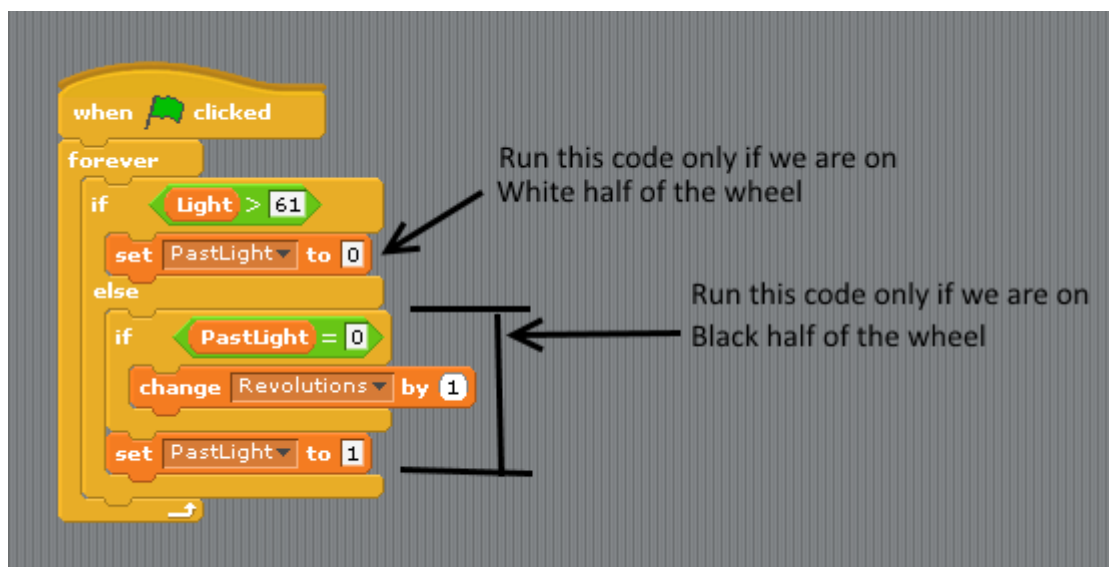
**Step 5:** Now it is time to calibrate our sensor. Press the Green Flag to start. You should now see the live values from your sensor in the Light value box on your stage. See the image to the right below. Turn your cups so that the shade above the light sensor is a shade between black and white and note down the value in Light. I got 61 but you may get a different value.



**Safety:** Please note that you use these resources at your own risk. Correct use of some components requires care.

**Step 6:** So now we can split our shade wheel into two sections if  $Light < 61$  it's on the half with the black section, if  $Light > 61$  it's on the half with the white section. In every complete turn the wheel makes, the light sensor goes from detecting the white half ( $Light > 61$ ) to detecting the black half ( $Light < 61$ ) exactly once so we can use this to count revolutions of the wheel.

Create the blocks below in your Stage Sprite. Replacing the value 61 with your value found in step 5.



Notice how when we are on the white side of the wheel we set PastLight to 0 and when we are on the black side of the wheel we set PastLight to 1. Also notice that the 'if PastLight = 0' block is only executed if we detect we are on the black side of the wheel. So basically what this is saying is if we last detected we were on the white side of the wheel and we are now on the black side of the wheel, add one to revolutions.

**Step 7:** Take your sensor outside or blow on it and see if it's working correctly, you may need to make some adjustments to make it work better but I'll leave that and making the rest of your game up to you!

**Safety:** Please note that you use these resources at your own risk. Correct use of some components requires care.