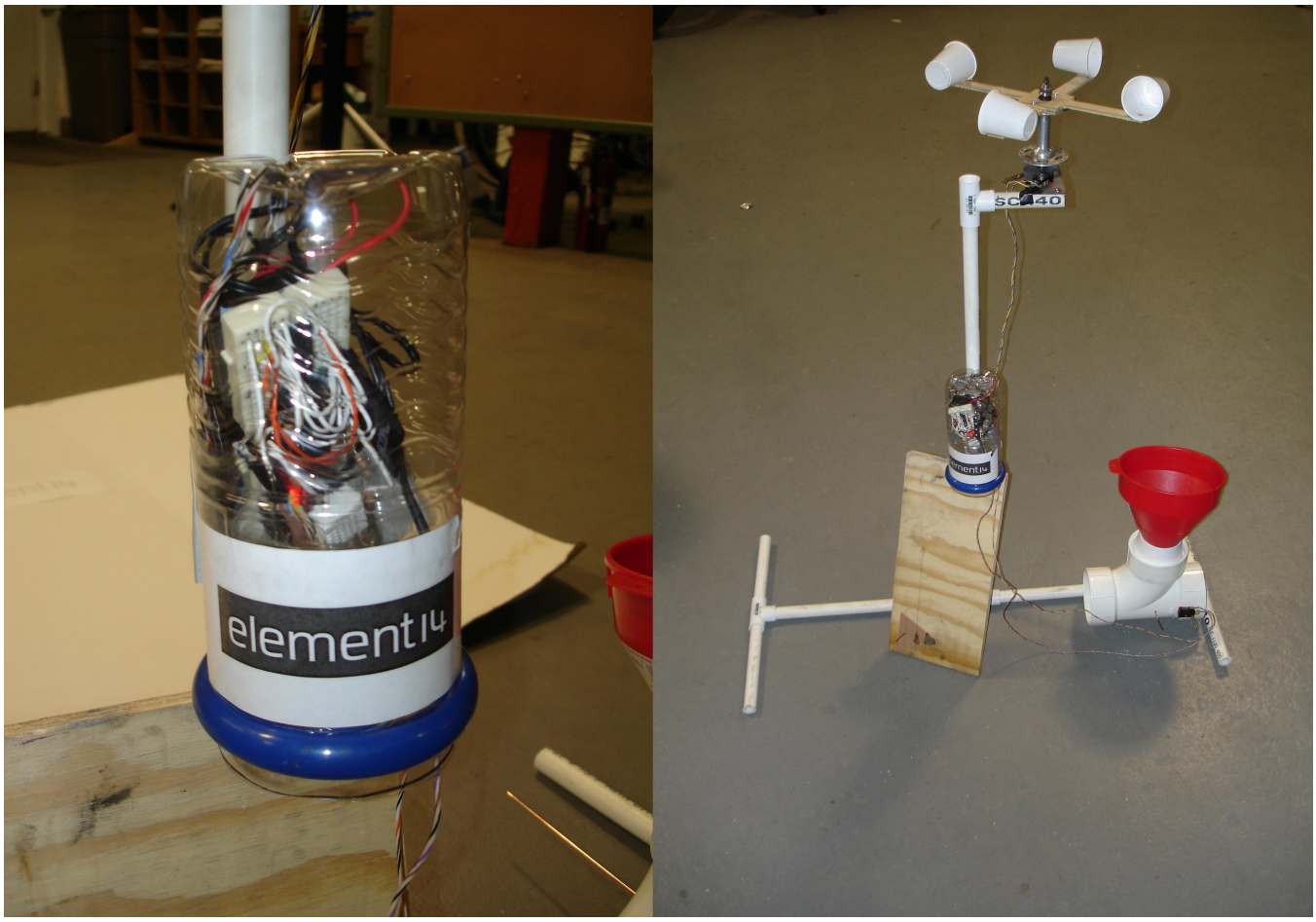
The Weather Window



Synopsis: The Maker Movement is about exploring your immediate world, learning on your own how it works and then tinkering with it as you see fit. Sector67 set out to embody as much of this spirit as possible in its creation of the Weather Window. We called it the Weather Window because when looking “through” it, a user will see opportunities and adventure!

The primary function of the Weather Window is to have users learn to gather and process data sets gathered from an omnipresent source, the weather!

The device is designed to have out-the-box functionality and independence from computers for use as an invitation for users to get busy exploring their world. In line with the Maker attitude, we wanted to empower the user to make the device their own by including “modular opportunities” in 5 concise lesson plans that expand understanding and instigate energy. Core goal of ours were to include opportunities for exploration for all levels of ability.

Data Logger/Weather Station

I. **Primary Objectives**
   a. Provide an economical device that enables the creation of relatable data sets based upon collection from a universal source, the weather.
   b. Facilitate the understanding and processing of data sets.
   c. Allow for multiple iterations and modular opportunities to engender further exploration and personalization of the device and experience.

II. **Basic Device**
   a. Functional weather station with the capability to measure and log temperature, light, humidity, barometric pressure all powered by a battery pack.
      i. All sensors are optional
      ii. Arduino Powered
      iii. Globally sourcable parts
      iv. Many options for each type of sensor
      v. Requires no end user configuration

III. **Powering the Device**
   a. **Scavenged, Rechargeable Power Supply:**
   Sector67 hacked a pair of Nokia cell phone batteries to provide a rechargeable and portable power source for the GHC device. Nokia's 1100/3200 series cellular handsets utilize the Nokia BL-5C battery, a single cell, 3.7V 850mAh Li-Ion cell. These handsets and batteries are widely available world-wide as the best selling cellular handsets in history, with over 400 million units shipped worldwide. Thus, they should be readily source-able as discarded or surplus items. Should you desire to source them new, the batteries are currently available on Amazon.com for $4 with free shipping.

   To create the GHC device power supply, take two BL-5C batteries and wire them in series. Run the connection wires along the front of the battery (the side with the +/- terminal markings.) Take care while soldering so that the battery will still fit into the handset body. The spring loaded battery connection pins in the handset will accommodate the new power leads. You can charge one battery at a time, using a single handset, or both batteries simultaneously using two handsets. The new battery pack is a 7.4V Li-Ion cell that you can pair with a standard 5V linear regulator to power the GHC device. We use the common 7805 linear regulator in our design. Pictures of the completed design follow on the next page.

The Completed Battery Pack, with Barrel Plug

Insert each battery into a handset to charge them

b. **Alternatives, if Nokia parts aren't available:**
If you don't have Nokia parts conveniently available you can source almost any surplus cell phone for this purpose. Check the battery voltage to make sure that it is 3.7V and make sure that the handset can accommodate battery leads after soldering.

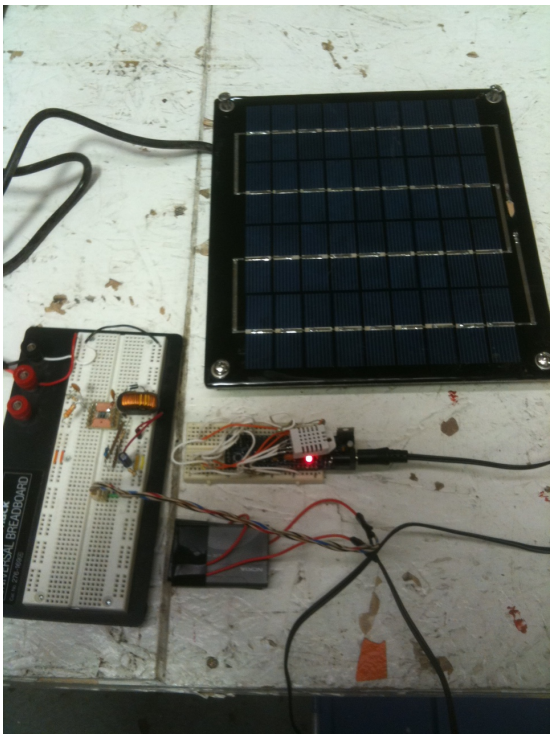
If no cell phones are available, get a 9V AA battery holder from an electronics supply house and power your project with alkaline or rechargeable batteries.

c. **Solar Power:**

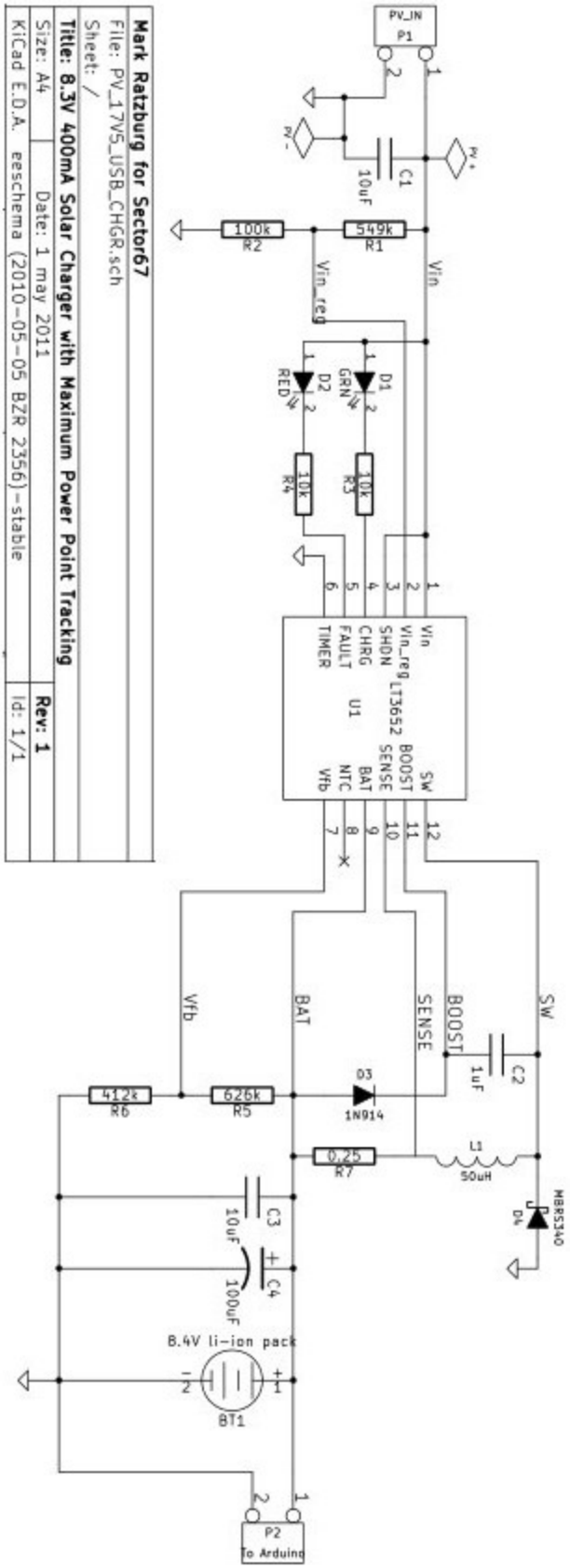
You may wish to leave the weather station unattended for an extended period while logging data. If the components are available, you can build a safe solar charging circuit for the lithium ion battery pack. Solar charging allows the weather window to log data autonomously for weeks at a time.

Sector67 designed a circuit utilizing the LT3652 solar charger IC with maximum power point tracking. Maximum power point tracking allows the solar panel to charge the battery under partial light conditions by regulating the output current to ensure the solar cell is supplying maximum power for the given light conditions. Solar chargers without maximum power point tracking cannot charge in non-optimal light conditions.

The circuit utilizes a solar cell with a maximum power point at 17.5V, and supplies a maximum of 400mA of current to the li-ion cells and the Weather Window. The completed circuit is shown in the picture below; the LT3652 is breadboarded using a surface mount breakout board. Additionally, the schematic is included on the following page.



The solar charger connected to a 17.5V solar cell



Solar battery charger schematic

Lesson 1 : Gathering Data

Why Data?

Data is the engine of innovation in this day and age. In using a familiar source of data (the weather), data can be collected, recorded and examined in a manner that will expand your understanding of information and how it is used.

Why Weather?

What is one thing you can count on every day? One thing that is always present at the exact moment you wake up in the morning? The weather, of course! It doesn't matter where you live or how you live, weather is always there, and it is something you are familiar with. And it is because of this familiarity and consistency we are embarking on a journey to explore data through the observation and documentation of weather.

Activity: Device Deployment

Deployment of the device is important to ensuring quality data is collected. Let's examine temperature as an example. If you place the device in direct sunlight, the device will record not just the air temperature, but also the heat generated by the sunlight striking the device. These types of data corruption are best avoided so the gathered data is as accurate as possible.

> *Follow-up Questions: What other issues can you imagine arising when trying to gather accurate data about weather? Where would you place a rain gauge?*

Placement for temperature and humidity:

- Flat, relatively open ground.
- A location that is representative of the area. Find the most common landscape in your area and use that.
- 1.5 meters/5 feet above the ground.
- Away from local heating and cooling sources.
- At least 6 meters/18 feet from buildings or other large structures.
- At least 30 meters/100 feet from large areas of concrete and/or asphalt.
- Avoid sites where water or snow collects.
- DO NOT install the sensors under the shade of trees or vegetation.
- Shaded from the sun at all times of day, away from objects the sun shines on.
- Hang sensors in the air 10 cm/4 inches from any object. (Objects retain heat.)

These suggestions will give you the most accurate data for air temperature and humidity in your area. It is difficult to follow them all. Do the best you can. Of course, they may not apply if you intend to measure a specific location (inside a structure, in a tree, under a rock).

Placement for barometric pressure:

- In a location with minimal air flow. *(This will be an issue when we get to sensing wind, but you'll figure it out!)*
- Relative pressure can be recorded without calibration. Simply log data, and then determine the range of that data. For an absolute pressure reading, you have to consider that altitude (height above sea level) affects pressure.

Placement for light:

- The most important factor in getting a light measurement is to be sure the light is not interfered with over time. As an example, placement under a tree would make the measurements susceptible to interference by wind shifting leaves, which in turn would affect the light reading.

Step 1: Determine how frequently data will be collected. We suggest collecting data every hour for this activity. Set the device to the appropriate setting.

Step 2: Assess the potential locations for the device and determine which is most ideal.

Step 3: Note the time of day and begin data collection.

Step 4: Return the following day after a 24 hour period has passed to collect the device and/or data.

Lesson 2 : Plotting Data

Why Plot Data?

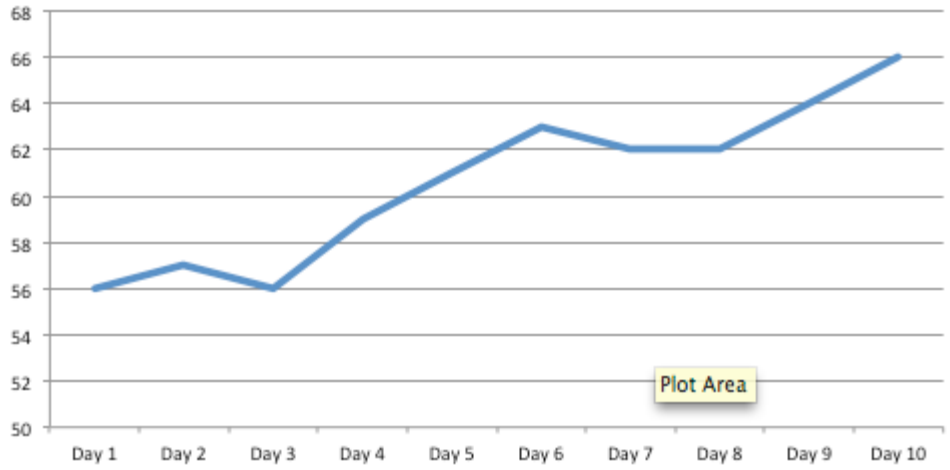
Plotting data makes it easier to understand. The visualization of information, which is what a plot does, offers varied perspectives and allows us see different characteristics of a data set.

As an example, let's say we have a data set of average daily temperatures collected over a period of 10 days. Our goal is to make a prediction as to what Day 11's average temperature will be. Compared to the table in Figure 1, the plot in Figure 2 makes the trend very apparent.

Figure 1

DAY	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11
TEMP	56	57	56	59	61	63	62	62	64	66	?

Figure 2



Activity: Plot Data for 10 days to make predictions of temperature, amount of daylight, humidity, and pressure for Day 11 and Day 12.

Step 1: Place weather station/data logger according to the guidelines provided in Lesson 1. We suggest you collect one data point per hour for this activity.

Step 2: Verify device is operating after first 24-hour period. *(It is good practice to double check that your tools are functioning.)*

Step 3: After the 10-day period is up, collect data from the device and document it. *(Either plug in the LCD screen and copy data to paper or plug the device/SD card into a computer to transfer data.)*

Step 4: Calculate daily averages for temperature, humidity and pressure. These calculations are dependent upon the number of data points you collected per day. If you collected one per hour *(as we suggest for this activity)*, you will add up the 24 data points you collected for a day and divide that number by 24.

Day 1 Average Temperature = (Temp1 + Temp2 + Temp3+...+...+Temp24) / 24

NOTE: A spreadsheet program (see Step 5) can calculate this with its AVERAGE function.

Step 5: Now we will plot the data.

With Computer: Import the data into a spreadsheet program such as Open Office (http://www.openoffice.org/) or Microsoft Excel, and then use the plot data functions to create plots. Tutorials are readily available online.

By Hand: Get a pencil and enough paper so you can plot your data in a clear fashion. A ruler or straightedge will be helpful as well.

Label the horizontal axis with the date or day the data for each average was collected.

For temperature, label the vertical axis “Average daily temperature, ° F”. (Fahrenheit or Celsius degrees can be selected in the embedded software in the device.)

The light sensor is not calibrated to a particular scale. Label the vertical axis “Average daily sunlight”.

For humidity, label the vertical axis “Humidity, %”. The values on the vertical axis will be 0 to 100.

For pressure, label the vertical axis “ Average daily barometric pressure, hPa”. (This unit is hectoPascals.)

Additional Resources:
http://nces.ed.gov/nceskids/createagraph/default.aspx

Lesson 3 : Long-Term Data Collection

Why Long-Term?

Collecting data over a long period of time is generally a good idea because doing so expands the total information set. More data means you will have more to work with. Longer-term data sets also minimizes days when weird things happened, making the spotting of trends and predictions more accurate.

Activity: Long-Term Data Collection

Step 1: Set the observation interval on the device. For this activity, we recommend once a minute, once an hour, or something in between.

Step 2: Deploy your device according to the guidelines outlined in Lesson 1.

Step 3: Verify device is fully functioning after the first 24-hour period.

Step 4: Retrieve data every few days or once a week, depending on how long your battery lasts.

Step 5: Continue to collect data. For this activity, we recommend a period of one month, to have sufficient data to compare to other data sets.

Step 6: Compare data to public information, like: http://www.weatherbase.com/

Lesson 4 : Rain Gauge

Why Rain?

Precipitation is critical to the cycle of life. Water is needed by all life forms.

Activity: Build a Rain Gauge

(1 inch = 2.54 cm)

Materials:

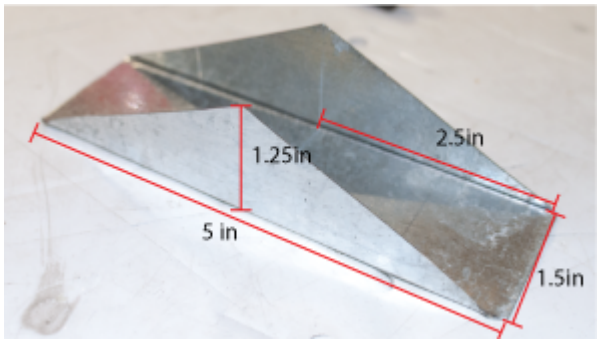
- 3 inch PVC pipe T-junction
- funnel (calculate surface area by diameter*π)
- 4 x 5 inch sheet metal or flashing
- 1.75 x 2.5 inch sheet metal or flashing
- 5 inch stiff wire
- 2 inch piece of 3 inch PVC pipe, or 1 Styrofoam cup
- Caulk or sealant glue of some variety
- micro-switch sensor (momentary straight lever switch)

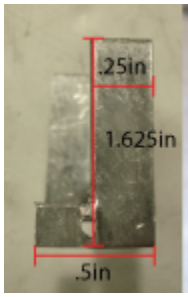
Tools:

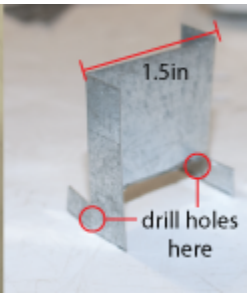
- Knife
- Hammer
- 1.5 x 1.5 x < 5 inch piece of wood
- drill and bit (size is relative to the thickness of the stiff wire)

Construction:

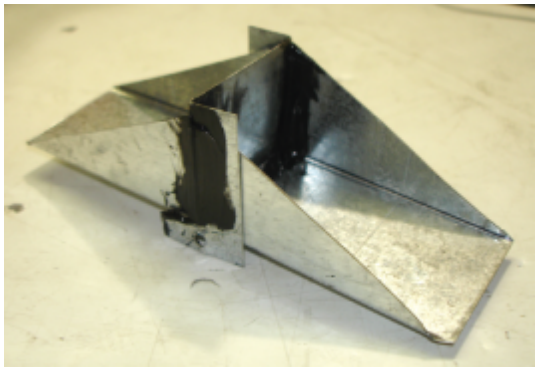
Step 1: Create the two pieces needed to make the tip bucket in the rain gauge. A sharp knife can be used to cut metal stock if it is thin enough. The bends are all 90 degrees and can be achieved with a hammer and the piece of wood. **NOTE:** Prior to bending the middle divider piece, do not forget to drill the mounting holes on the two tabs that extend below the tip bucket.



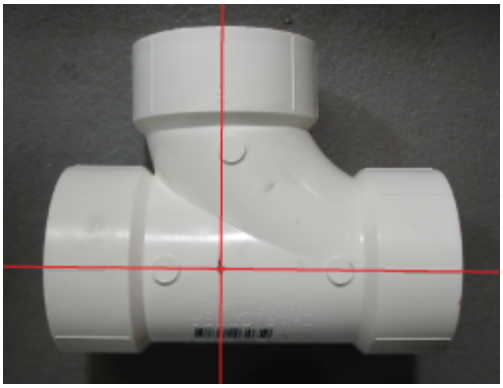




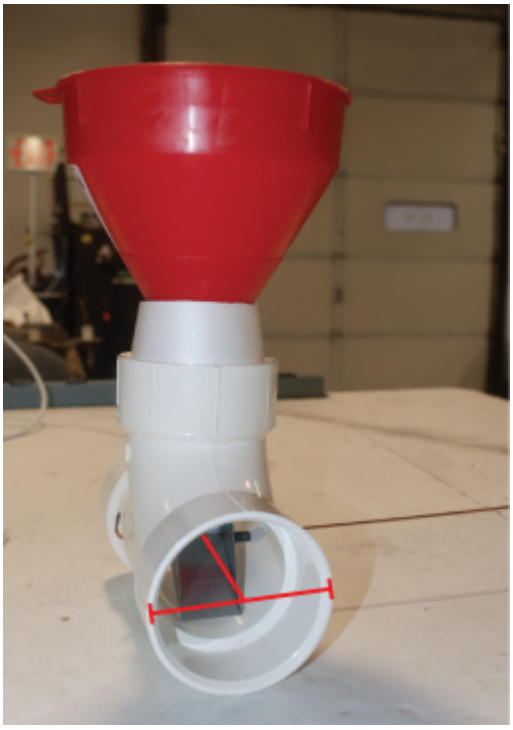
Step 2: Assemble the tip bucket and seal the seams with an adhesive (caulk).



Step 3: Drill a hole at the center point of the 3 inch PVC pipe T-junction

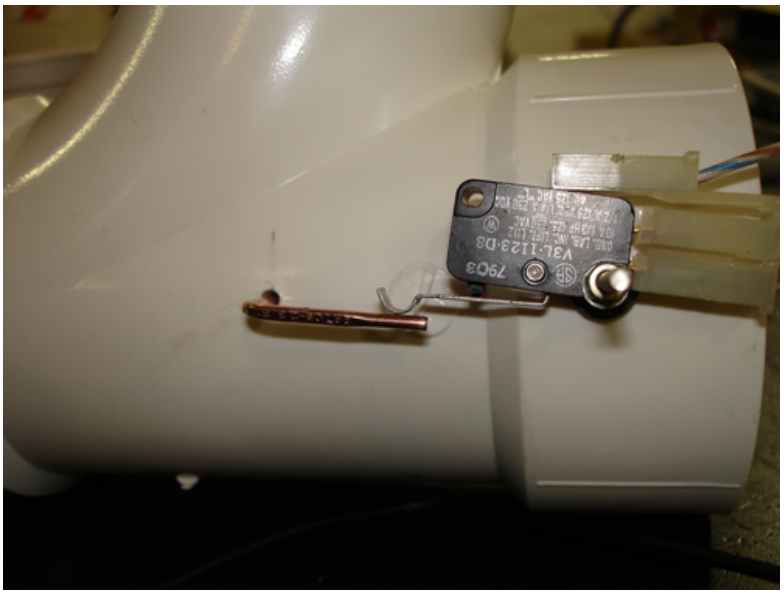


Step 4: Slide the 5 inch wire through the hole in the T-junction and the tip bucket construction as shown below. The bucket should be centered in reference to the horizontal axis of the T-junction (as shown in the image below) and fixed to the wire (adhesive works great!).

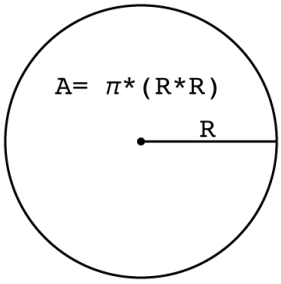


Step 5: The funnel is then attached to the upright outlet of the T-junction. A spacer may be needed to ensure clearance between the spout of the funnel and the tip bucket. In this case we used a Styrofoam cup with the bottom cut off, but a 2 inch piece of 3 inch PVC pipe would work, as would any number of things.

Step 6: Attach switch as shown in the image below. In this setup, the switch is tripped with every two fills of the bucket.



Step 7: Calibration of the device is done in two steps. First, calculate the area of the mouth of the funnel. In this case, the radius is 3.25 inches, so the area is 10.21 square inches.



Second, the volume of the tip bucket needs to be calculated. This can be done by using the dimensions of one side of the tip bucket (1.25 x 2.5 x 1.5/2) or by pouring a known amount of liquid into the tip bucket. Our tip bucket has a volume of 24 milliliters per side, so in order for the switch to be tripped each bucket will be filled once (48 milliliters).

The final step is to connect the gauge to the Weather Window device. Wire the switch to pin 3 and +5v.

Lesson 5 : Anemometer (Wind Speed Gauge)

Why Wind?

Wind speed is a basic metric that is used in the power generation industry (wind turbines), aviation, transportation safety, and weather forecasting. The shifting of wind patterns over time can be an indicator of significant climate change.

Activity: Build an Anemometer

(1 inch = 2.54 cm)

Materials:

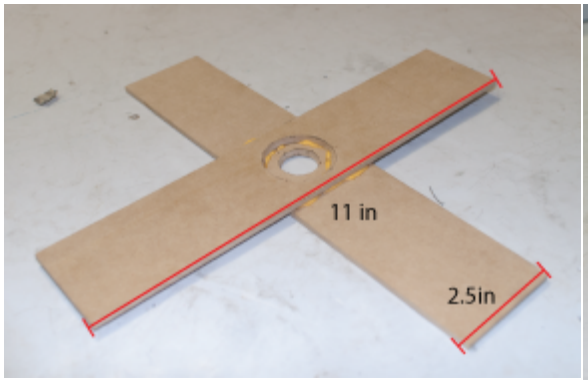
- 4 cups (bottoms of aluminum cans would work)
- 1 hub from bicycle wheel
- Adhesive
- 11 x 5 inch piece of thin, rigid wood (or metal or plastic)
- 5 small bolts
- 4 x 2 inch piece of sheet metal or flashing
- micro-switch sensor (momentary switch with a long straight lever)

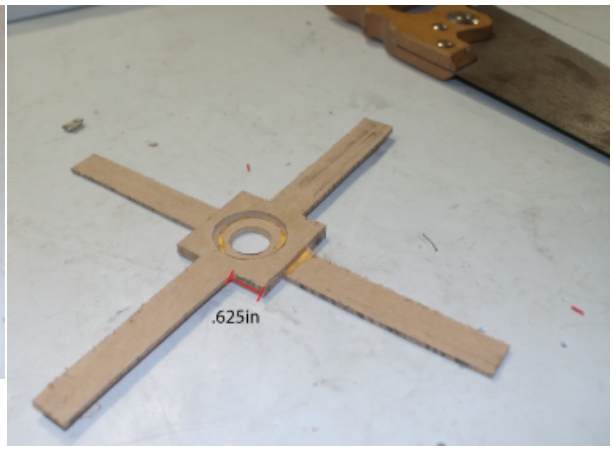
Tools:

- Knife
- Hammer
- Drill and bits (sizes of the diameter of bolts and hub axle)
- Razor blade, knife or small saw

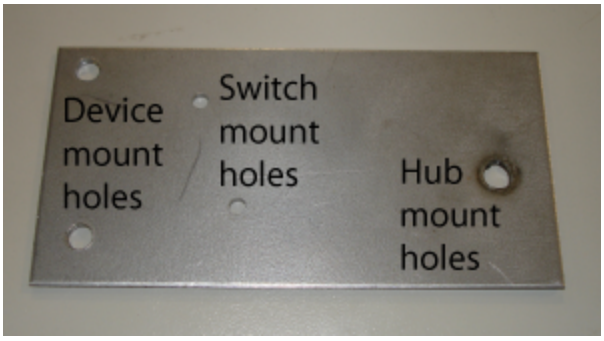
Construction:

Step 1: To build the rotor, cut the 11 x 5 inch piece of wood into two 11 x 2.5 inch pieces. Using the adhesive, mount the pieces perpendicular to each other as shown in the image. Drill a hole at the center of the intersection, to fit the axle of the bicycle hub. To reduce weight for better readings, cut away any unnecessary material.

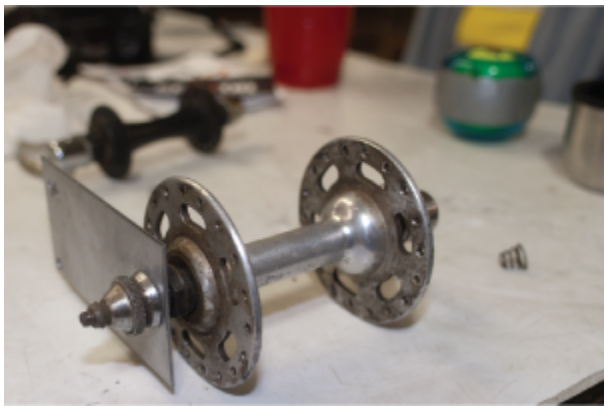




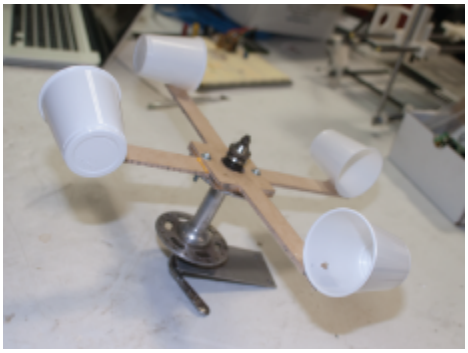
Step 2: Drill holes in the 4 x 2 inch sheet metal to attach the bicycle hub, the sensor switch, and to mount the data logging device.



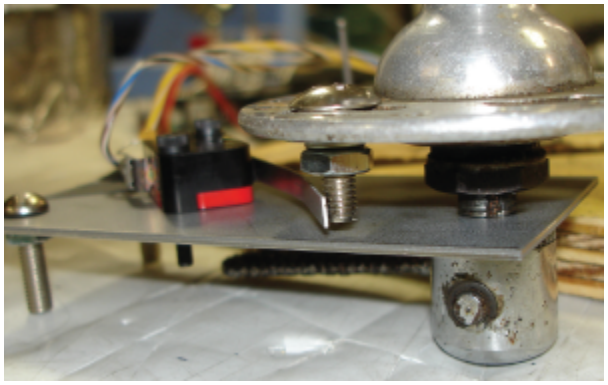
Step 3: Attach the hub to the mounting plate.



Step 4: Attach the cups to the rotor. We use a hot glue gun, but any strong adhesive will work. Then attach the rotor and cups to the bicycle hub.



Step 5: Attach the switch to the mounting plate. Here we have used a bolt to trip the switch on every rotation. The unit of measure is revolutions per second.



The final step is to connect the switch to digital pins 2 and 3.

END USER CONFIGURATION GUIDE

The Weather window datalogger is designed for the easiest user experience possible. The only end-user field configurable option is the amount of time between data samples. All other functions are automatic.

SELECTING LOGGING INTERVAL

On bootup, the weather window will look for a file on the SD card, in the root directory, called “interval.txt”. This file must contain one line, that only contains a number. The number is the number of seconds between logging points. If no file is found, the station will default to one second intervals

EXAMPLE VALUES:

1 hour=3600
1 day=86400

GHC: Sourcing/Building the Device

The datalogging device is made of several groups of components. the primary ones being:

-The Arduino core
-SD card interface
-Sensors
-Software

This document will discuss how to source each of the components, and various options for each. The scope of this project is focused on using the device, these notes assume a level of familiarity with electronics that the rest of the lesson plan does not. If you are an educator, this guide should be all someone with basic electronics experience would need to build the device for you.

Great care has been taken in the design of this hardware to ensure that each component is as easy to obtain as possible. If you find that any components are not easily sourced in your area, please let us know at team@sector67.org.

The Arduino Core

Items required: 1 Arduino core platform

The datalogger is built around an arduino core. http://arduino.cc In our schematic, we have used the "standard" arduino uno. but have tried several variants with similar success. The overall load on the processor for this project is extremely low, as are demands for input and output pins. the arduino was chosen for its global availability, and excellent documentation. The most practical and best working variant we used was the "boarduino" (http://www.ladyada.net/make/boarduino/) The boarduino has a much lower overall cost, uses all through hole parts, and leaves the expensive USB programming connector as a detachable component. USB programming of the device /should/ only be required once. For a classroom environment only one USB programmer would be required for an entire classroom supply of devices.

None of the arduino devices are required to be purchased from any particular vendor. The design is open hardware, and the code is open source software. All of the components should be extremely easy to source anywhere in the world for the entire arduino core platform.

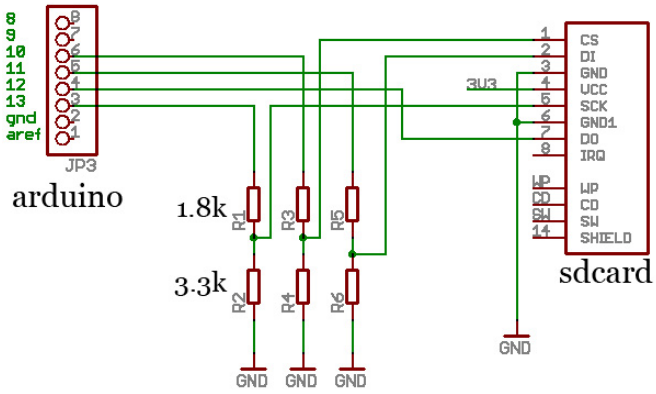
SD card interface

SD/MMC was chosen for a storage format because the prevalence of mobile phones around the world has made them very accessable. Both regular size, and the smaller "transflash" or "micro SD" size

can be used for this interface. Data logging requires very little in the way of file size, so even a 64MB/ 128MB SD card would be more than useful for this project. It is more cost effective to use the microsd cards, if you can find them, because the hardware for the microsd slot is much easier to come by than the SD card slot itself.

The mechanical interface for the SD card can be handled a number of ways. By far the easiest way, is to take advantage of a SD->MicroSD card adapter. As shown below, a simple modfication with 0.1" header pins results in a very reliable and cost effective SD card interface. The plastic may meltt, but will work to keep the solder traces separate! of course, any type of SD card slot can be used.

The electrical interface for the SD card is slightly complicated by the 3.3v requirement to talk to an SD card, and the arduinos 5v core. The provided schematic solves this the "correct" way using a **SN74AHC125N,** but using a voltage dividing network for each of the 3 data pins on the SD card, is a very cost effective way of accomplishing the conversion. The arduino can read the 3.3V from the card on its 5v input, but trying to access the card with 5v can permanently damage it. You will not be able to access the full command set for interfacing with the card, but for the purposes of the datalogger, only writing one block at a time. it is more than adequate.



Sensor hardware

For useful data gathering, at least the temperature and light sensor must be used. These two are considered our "core" sensors to get the device going. Other sensors are noted, but are considered "optional" due to the difficulty of globally sourcing parts.

Light sensor

The light sensor is a standard Photocell. any type will do, as we are interested in the relative light level, not the exact light level. as we are for most sensors. A simple resistor is all that is needed to interface to an analog input on the main board. One example is: http://www.newark.com/excelitas-tech/vt93n2/ldr-500kohm-80mw-vt900-series/dp/99F5218

Temperature sensor

The temperature sensor is a standard 3 pin analog temperature sensor. Any variant of analog temperature sensor will do. We are mostly interested in the change in temperature, but even the cheapest temperature sensors will have calibration values that can be accounted for in the code, or in the end application, that will give temperature readings within a degree or two. One example of this, used in our reference design, is a TMP36 http://www.newark.com/analog-devices/tmp36gt9z/ic-temperature-sensor-3-c-to-92/dp/19M9015 Which is handy because it requires no external components.

OPTIONAL SENSORS

Aside from the wind and rain gauge, which can be easily constructed with the supplimental materials, several other sensors are easily interfaced to the project. The code, by default, is accounting for a wide variety of optional sensors. Most of these sensors are much more complicated than simple analog voltage readings, but are still easily interfaced with a bit of code. Example code for the following, specific, sensors are available. but there are very few small sensors that can /not/ be interfaced with an arduino, but example generic code cannot be provided.

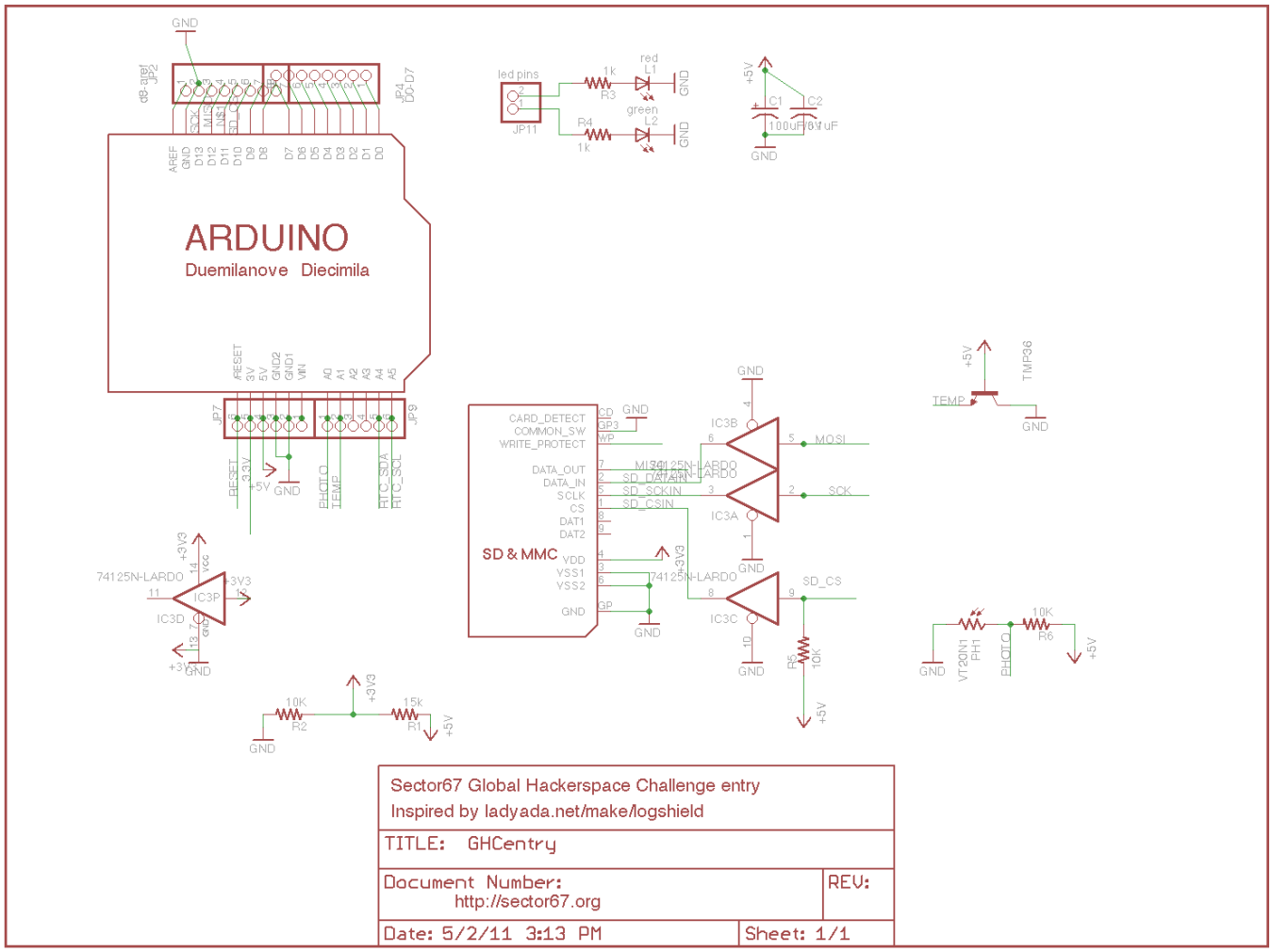
The specific sensors used in our code are:

Bosch BMP085 pressure sensor:
http://www.bosch-sensortec.com/content/language1/html/3477.htm

HIH-4030 humidiy sensor:
http://www.newark.com/honeywell-s-c/hih-4031-001/humidity-sensor/dp/62M4609?in_merch=Popular%20Products&in_merch=Popular%20Products&MER=PPSO_N_C_EverywhereElse_None

These sensors provide a very high level of functionality, at a very high cost so they were included to show an example of high level functionality on the back of the cheap core.

SCHEMATIC



SOURCE CODE

The source code is provided in the form of an “arduino sketch”, and is located in the google code repository at http://code.google.com/p/sector67-ghc