

Beanie Turns Brainwaves Into Light

Update December 2017: We've noticed a lot of people trying to contact us with questions or debugging through comments or private messages. Please email us directly at illumino.eeg@gmail.com or else we may unfortunately not answer you on time!

Ever wanted to visualize your brain activity in real-time? Move an object on a screen with your mind? This tutorial shows you how to make a recreational EEG beanie that turns brainwaves into light, using a microcontroller, LEDs and a special chip that processes brain activity. It also demonstrates how to use the hat to control audiovisual interfaces in Processing.

To make this toy EEG device, I connected the [ThinkGear ASIC Module](#) (a chip from [Neurosky](#) which processes real-time brain activity) to an Arduino. The custom Arduino firmware then converts these brain signals into light of different colors, intensities and brightness, which change according to the mental state of the wearer. Neopixel LEDs are integrated into the hat's pompom, however if you prefer you can place LEDs elsewhere on the hat, or on an accessory (for this you will need to include bluetooth). All the small electronic components are hidden in the fold of the hat, so it looks and feels as though you're just wearing a comfortable beanie.

The finished hat:

...and here's a video of the testing of the hat (before putting the LEDs into the pompom and securing the battery onto the hat yet, hence the hanging wires). Turn on the captions for explanations!

Connecting the beanie to Processing

The Arduino is easily accessible via a micro USB connector hidden inside the hat's pompom. This allows you to create live interactive "brain art" with the hat, using a programming language such as [Processing](#). Simply connect the micro USB on the pompom to your computer's USB port, and import the real-time serial stream into Processing (or wirelessly if you make the hat with bluetooth). Then customize your Processing sketches so that the levels of "attention" and "meditation" control the sketch variables (such as color, opacity, movement, coordinates, rotation speed, etc). To get you started, I've included in step 13 the snippet of the Processing code you will need. There are endless possibilities: you can draw, write words, control moving blobs, play games. Here are two simple examples, full screen in HD for best effect!

Fly a bird using your brainwaves: In this example, the real-time amount of "attention" controls the vertical flight of the bird. The more I concentrated, the higher the bird flew. It takes some training to get used to it! You could also control, if you wanted, the speed of the wings flapping, or the trajectory of the flight, etc. Script was adapted from [the original sketch by Liana](#).

Brainwaves controlling the color of a blob: In this example, the real-time levels of "attention" (green) and "relaxation" (turquoise) control the color of the blob. Script was adapted from [the original sketch by Oggy](#).

Enjoy building your brainwave hat! If you know teachers or classrooms who would love to make an EEG hat, please share this project!

If you write about this project, thank you very much for sharing, and please link to this instructables tutorial.

Step 1: A Bit of Science

How can we measure brain activity using a wearable hat?

Electroencephalography (EEG) is the measurement of neural activity through sensors (electrodes) placed against the scalp. These electrodes can measure the tiny electrical changes that occur when neurons fire. By amplifying these signals through a computer, we can observe a person's brain activity in real-time. Here is a great video that explains the fundamentals of EEG (University of Waterloo):

Cleaning up the signal

When a brain signal is recorded from a person's head using EEG sensors, it picks up a bunch of information other than brain signals: muscle and eye movements from the person, and especially interference noise from that old refrigerator in the corner of the lab. Before the brain signal can be interpreted, we need to clean it up or "pre-process" it: filter out the noise, remove all the muscle, eye movement and blinking from the signal.

We then do a couple of other mathematical manipulations with the data to estimate the amount of different brainwaves by frequency of their oscillations (there's not only one type of brainwave). Neural oscillations can thus be classified into "frequency bands": **Delta** (0-4Hz), **Theta** (~4-8Hz), **Alpha** (~8-12Hz), **Beta** (~12-30Hz) and **Gamma** (>30Hz).

In research experiments, the signal cleaning is done manually after the experiments, using special software. Commercial EEG headsets contain chips that pre-process the signal in real-time. This tutorial uses Neurosky's ThinkGear™ ASIC Module, which filters out HF noise and muscle movements from the real-time brain signals, and applies custom algorithms designed by Neurosky to detect approximate levels of "relaxation" and "attention", as well as the levels of delta, theta, low alpha, high alpha, low beta, high beta and gamma waves.

Interpreting the brain signal

Now that we have cleaned up the brain signal, we can start to interpret it. Changes over time in the levels of the different frequency bands reveal important information about the mental states of a person, for instance: if they are asleep, concentrating on a difficult mental exercise, relaxing, have their eyes open or closed, etc. Certain brain signals can also be indicative of clinical conditions (epileptic seizures or sleep disorders).

In a typical research experiment, the EEG is recorded and then averaged for each participant (over many, many trials), for a large group of participants. Scientists study recurrent patterns of neural responses to visual/auditory/multisensory stimuli to understand how the brain processes and encodes information. For example: based on a person's EEG, can we predict whether they are looking at a photo of Times Square at rush hour or an eagle flying against a blue sky? It appears that we can, by the way.

Brain-computer interfaces

Recurrent patterns in brain activity can also be used for writing algorithms for brain-computer interfaces (BCI). A brain computer interface is a computer system that allows a person to control an object using only their brain. Several BCIs are currently being developed to assist persons with restricted mobility and/or communication, allowing them to write or control objects on a computer screen. Two methods commonly used in developing such BCIs are motor imagery (imagining that you are executing a specific body movement) and the P300 event related potential (a positive change in voltage that roughly starts around 250ms after stimulus onset). A few spelling systems use these methods to allow people to communicate with their brainwaves. However, they require substantial training and they don't allow us to simply think of words and subsequently see them appear on a computer screen. The easiest signals to use for controlling objects with a BCI would be facial muscle activity, which is detected by sensors placed in

the frontal area of the head. These signals are easily recognizable (left/right eye movements, blinking, jaw movements, clenching teeth, etc). There's a lot of exciting progress being done in BCI research. However, keep in mind that BCIs do not (yet) allow us to do "mind-reading". The closest thing I have seen to "mind-reading" would have to be [this very cool experiment conducted at Berkeley](#), using fMRI (not EEG).

University of Minnesota's mind-controlled quadcopter:

Portable EEG vs Clinical/Research EEG

EEG equipment used in research labs and hospitals provide robust, high quality recordings of brain activity. Unfortunately, the wires connecting the headset to amplifiers restrict the mobility of the wearer. Wireless EEG headsets are easy to put on and nowadays many of these use "dry" electrodes (electrodes that don't require gooey gel or saline solution). The main limitation in using portable EEG devices for research purposes is the spatial & temporal resolution (1-14 electrodes; average sampling rate 128 Hz), which is nowhere as good as that of research EEG headsets (32-256 electrodes; sampling rates up to 20 kHz). This means that it is much more difficult to extract and accurately map the source of a brain signal with a portable device than it is with advanced equipment. Also, in a portable device the signal is more noisy as there's a lot more muscle activity due to the person moving around (even though a lot of the noise is filtered out during signal pre-processing). However, even with lower resolution, portable EEG headsets provide accurate readings of frequency band power, and can be used for different applications such as developing simple non-invasive brain computer interfaces, and for certain clinical practices (neurofeedback). Wireless EEG headsets are fairly recent, and their efficacy will improve along with technology.

- Research/Clinical EEG systems: EGI, Biosemi, BrainProducts, G.tec

- Wireless & portable EEG devices: Emotiv, IMEC, Muse, Melon, Neurosky
- Open Source EEG: OpenBCI

Step 2: Design Carefully

When I started this project, the design consisted of two hats on top of each other: one with all the electronics sewn on, and over it another thin knitted hat. Although it worked fine, it was too warm to wear, not very user-friendly, and the electronics got caught up in the "overhat" resulting in breaking points that needed constant repairing. Most of all, the several layers of hats made my head look a bit swollen.

Bottom line: design is as important as electronics. You are not going to want to wear a hat that is not as comfortable and normal looking as it is sturdy and functional. Choose small electronic components, and ensure that they are imperceptible. Put your soldering points on places of the hat that are not likely to get a lot of tension (the flat surfaces of your head are best). Wrap everything in isolating tape, sew down the wires, all this will really prevent any wear and tear.

Step 3: What You Will Need

The total cost of all materials and electrical components should be around \$100-\$130 (this is what it cost me to make this hat, including the Neurosky chip). This is relatively cheap for a decent quality (very comfy/discreet) one-electrode EEG wearable device. I've provided links for the materials. Most of these you can find online at [Adafruit](#) & [Sparkfun](#), for a bargained price on ebay shops, or at your local hardware store.

- 1 double-layer hat with a wide foldover flap. No need to buy the most expensive hat: as long as it has a wide foldover flap, and most

importantly a double layer, it will do. You will hide all the electronics in the foldover part. The pompom wires will be run through the double layer of the hat.

- Weather stripping adhesive tape, wide (~1.5 inch)
- Thread, and a needle to sew with
- Glue which we will use for the pompom lights (all purpose metal & fabric clear glue).
- Kapton tape or normal isolating electrical tape
- Heatshrink for your wires
- [A 3.7V lithium ion battery](#) (500 mAh: less than this might heat up too much, and more than this is bulky).
- [1 JST battery extension](#)
- Thin and flexible but sturdy wires (~0.5mm). I like these [silicone wires](#).
- [2 shielded wires for the electrodes \(RG174 wires are fine\)](#)
- [2 4-pin connectors \(Male and female\)](#)
- 1 [tactile pushbutton switches with 2 leads](#), for the "Mode switch" (switch that allows you to select colors and brightness).
- [Conductive lycra fabric](#).
- High density foam, to make the electrode bump
- [Copper tape](#)
- Solder
- A soldering iron
- White thick yarn for the pompom (preferably white for best light diffusion/visibility).
- [Thin copper foil sheet](#)
- 1 [ECG snap lead](#) (for the reference electrode).
- [Small adhesive transparent one-time use gel electrodes with snap](#), for the reference electrode. You can also [get them from BITalino](#).
- A multimeter
- A wire stripper

Step 4: Electrical Components

The main components you will need are:

- a [lithium battery](#).
- a [JST breakout board with an on/off switch](#)
- an Arduino microcontroller, for the custom firmware that interprets the brain signal and turns it into light. A USB adapter is attached to the Arduino, to allow easy access to the software from a computer. I used the [Tinycircuits mini FTDI USB adapter and TinyLily Arduino](#). I like the TinyLily because its tiny size is very handy. Alternatively, you can use an Arduino that has an onboard USB slot, such as the LilyPad Arduino (the LilyPad328 and the Uno are compatible with the software). If you're going to implement bluetooth, you'll need an Arduino with more RAM.
- Brain signals are detected by the EEG electrode attached on the forehead area, on the inside of the hat. The tutorial shows you how to make your own dry EEG electrode. However, if you prefer, you can buy electrodes.
- [Neurosky's EEG TGAM](#). You can find this chip inside Neurosky's Mindflex headset. Although this product has been discontinued, you can still buy it on the Neurosky store, or second hand for a bargained price on ebay. The Neurosky chip is a component present in several of the company's commercial EEG headsets. It amplifies and pre-processes the incoming neural data, outputting real-time estimated levels of attention, of relaxation, and frequency band power, using custom algorithms. Since I do not have access to the details of this software (proprietary information of Neurosky), I can't give you a better explanation of how these algorithms work. You can read more about this in the attached document "Attention_relax_bands.pdf". If you wish to use your illumino hat to develop your own non-commercial brain-

computer interface applications beyond generating light or make interactive visual/acoustic art, I suggest you take the time to also read the attached "TGAM_Datasheet" pdf which explains the specifications of the Neurosky chip.

- Since illumino is programmed to turn the real-time brain signals into colorful light, **4 RGB LEDs** are connected to the Arduino. I use [Adafruit's Flora Neopixel LEDs](#), but any small RGB LEDs will do the trick.
- You will also need a [USB charger](#) to recharge the lithium battery via your laptop computer.

Note: To connect the Neurosky EEG TGAM to the Arduino, you need to first remove it from the plastic casing of the Mindflex headset. No modifications are made to the software or hardware of the TGAM. Rather, you integrate the chip into the circuit via a simple mechanical adaptation that allows you to connect it to an Arduino to provide a lighting function for the hat.

Step 5: EEG Electrode

An EEG device has two types of electrodes: the EEG electrode, which records brain signal from the scalp; and the reference electrode, which informs the EEG device what "no brain signal" resembles (baseline). **This is the most important step of the tutorial:** if you build the electrodes carelessly, your signal will be terrible, and it will pick up a lot of noise and interference. Take the time to refine the design if necessary.

I built the hat with a dry electrode, which provides a decent signal quality and is less messy to apply than traditional gel EEG electrodes. You will need to ensure a constant contact of the electrode with the skin, and to keep this area sweat free.

To construct the EEG electrode, you will need foam, copper or silver tape (conductive!), a copper foil sheet, silver conductive fabric (but not woven, it should be soft and flexible), weather stripping adhesive tape (wide), and a shielded wire. Carefully choose the right shield; a foil shield may not work well when the wire is bent or moves around (this may introduce high frequency noise into the signal). An RG174 wire is perfect.

- Make a rectangle out of the copper foil. Cover it with conductive fabric and then with copper or silver tape.
- Make a bump shaped mountain out of foam. Add some more conductive tape if you wish.
- Solder a wire to the base of the copper plate. Since there are two pins to input the electrode wire onto the Neurosky chip, the wire should branch out (it should resemble a "Y"). These branches are also made of shielded wire. The EEG electrode wire should be short: this greatly improves the quality of the signal. Make a small incision in the hat so the electrode wire comes out on the other side.
- Run the electrode wire through this hole. The end of the wire that branches out should stick out on the front of the hat. This is the extremity at which you will attach the Neurosky chip. The other side of the wire is attached to the electrode, on the inside of the hat (in the front). Look at the photos if this sounds confusing.
- Solder the wire branches to the Neurosky chip where it says "EEG". Doesn't matter which branch goes onto which pin.
- Position the electrode on the inside of the hat. Be sure that it is placed smack in the middle of the forehead (in EEG we call this position "FP", stands for "frontal parietal"). It should be in direct contact with your skin, well below your hairline. Any contact with hair will contaminate the

signal. However beware that if you place the electrode too low, it will be visible.

- Cut a small incision in the middle of the weather stripping, to allow only the electrode to pass through. Make sure the electrode is in place, then stick the weather strip onto it and press down. The adhesive weather strip should normally stick well onto the fabric. To improve the overall comfort, you can sew some microfiber cloth over the area. Try and keep this area flat. If it's too bulky it will look suspicious when you are wearing the hat.

Safety note: This electrode is perfectly safe to put on your forehead. There is no electricity running through it, unless you've really soldered the wires wrong :(. Still, in the worst case you would get a discharge of 3.7 volts (can't even feel that).

Step 6: Right Ear, Reference Electrode

There are two places you can place the reference electrode:

- **On the mastoid** = the bone behind your ear
- **On the earlobe** = this is mainly skin and fatty tissue

Placing the reference on the mastoid is invisible. When you are wearing the hat no one can see it. The earclip, on the other hand, is entirely visible. For the mastoid reference, you will need a snap sensor wire, and adhesive electrode patches. These patches are very cheap. It is best to use a new one each time. They are harmless when removed, and will not leave any gel on your skin.

- Solder a snap ECG sensor head (or an earclip, up to you which one you use) to a shielded wire (important!).

- Branch out the other extremity of this wire into 3 parts.
- Solder these branches to the "REF" pins on the Neurosky chip.

Step 7: Back of Hat, Battery and Power Switch

The hat runs on a lithium battery which is rechargeable through a USB connector. The back of the hat is the best place to put the battery and JST breakout board.

- Place the battery below the fold of the hat flap. Cut a rectangle out of felt or microfiber, slightly larger than the battery. Sew it on, to make a pouch for the battery.
- Place the JST 2-pin breakout board on one side of the battery.
- Solder on two leads: one into the "SW" pin, and the other into the "GND" pin. These leads should be long enough to go around to the other side of the hat. Later on, in the tutorial, you will connect these wires to the GND and SW (+) of the Arduino.
- Secure the switch in place by sewing or gluing it onto the hat.

Step 8: Arduino Software

Update October 2016: Several people have emailed me about missing libraries, so I have added a bunch of them to the zip file. Let me know if things still break so I can repair the code if needed!

You can upload the software into your hat either before constructing the LED structure that contains the Arduino, or afterwards when everything is finished. As you prefer.

Function of the software

The Arduino code converts the data string being received from the Neurosky chip into light of different colors, brightness and intensity: all of which are programmed to change according to your mental state. The pompom color is always be a combination of the real-time percentages of "attention" and "relaxation". For example, if a user has chosen yellow as his "relaxation color" and blue as the "attention" color, then the pompom color will be a mix of yellow and blue (100% attention = 100% blue, 50% attention, 50% relaxation = 50% blue 50% yellow, etc). If you achieve a high level of relaxation or attention over time, then the pompom will shine in "attention achieved color" or "relaxation achieved color". These colors are the same as the "attention color" and "relaxation color", only brighter (one Neopixel is white).

The color and brightness settings or "modes", can be controlled using the "Mode switch" (the black pushbutton switch on the hat). This switch allows you to select a specific color to represent the level of "attention". The "relaxation" color is automatically chosen for you by the software, as an opposite color of the "attention" color. For example, if you choose green for the attention color, the software will assign purple as your relaxation color. You can also adjust the maximum brightness of the Neopixels, check the signal quality, or choose to use the pompom as a beacon light without the EEG. Although I have not done it, you can write additional code that allows you to visualize a color that represents a mix of, or dominant frequency band power (alpha, beta, delta, theta, gamma).

Uploading the software for

The software is in the attached file `illumino_updatedOct_2016.zip`. `Awell.ino` is the main Arduino script. All code is commented to be self explanatory. Please don't hesitate to ask if you have any questions. A special thanks to Kurt Olsen for his dedication in assisting with the firmware!

- Mount the FTDI USB onto your Arduino.
- Plug the Arduino into your computer. Always be very careful doing so, as the mini USB component is delicate.
- Install the Arduino IDE 1.0.6 (<http://arduino.cc/en/main/software>) and appropriate drivers on your computer. Great websites for learning how to program an Arduino: arduino.cc, [electronhacks @.youtube](https://www.youtube.com/channel/UCyOYU0811333333333333333), [adafruit](https://www.adafruit.com).
- Select the right board and serial port (COM). This is usually the lowest number.
- Press the "verify" button.
- Press the "upload" button.
- Now, open the serial port. There should be a stream of numbers flowing down fast. This is the brain data incoming from the Neurosky chip.

Step 9: Visual Feedback, NeoPixel LEDs

The hat is designed to work with special RGB LEDs called NeoPixels. In this tutorial, I integrate the Neopixels into a pompom on top of the hat. If you choose to integrate a bluetooth component to your hat (see diagram in the next step), you can sew the Neopixels onto clothing, or place them in a lamp/bracelet/necklace, instead of a pompom. Be creative!

To make this LED component, you will need 4 RGB Neopixels, wire, solder, the TinyLily USB connector, the TinyLily Arduino, strong glue (on the inside) and Sugru (on the outside for the finishing touch).

- Connect the Neopixels to each other. 3 should be as close as possible, with the 4th a bit farther off.
- Connect the Neopixels to the Arduino, as shown in diagram (+, -, LED).
- Connect the Arduino to the rest of the circuit.
- Run 4 leads through the double layer from the top incision to the incision below the flap limit, where they should emerge.
- You should have 4 leads connected to your Arduino: GND, SW+, MODE, and TX. Label these leads with color heatshrink so you know which is which.
- Also label the leads with color heatshrink at the bottom extremities, where they exit from the incision in the hat. You will connect these to the rest of the circuit in the next step.
- Now plug the USB FTDI into the Arduino, being extremely careful since if you bend the pins too far it will break easily. Apply strong glue on the basis to hold these together. This is a bit tricky, so you might want to use some crocodile pins to help hold everything together. This is one of the most delicate parts of the hat components, and it will break easily if not connected well. Be sure you take your time and let the glue solidify before moving on to the next steps.
- Shape the Neopixel structure in a circular structure, around the mini FTDI USB adapter. Continue adding glue until the whole thing looks like a "cocoon". Do not cover the center of the Neopixels in glue!
- Put it out in the sun to dry for a couple hours. Stay vigilant: birds steal Neopixels for birdnests!

- When the "cocoon" is dry, add Sugru to the outside of the structure, to give it a clean finishing touch, being very careful not to put any on the Neopixels themselves or inside the USB slot.

Don't want lights at all?

The hat functions perfectly as an EEG device without the lights.

Step 10: Front of Hat, Connecting the Components

Turn the hat around, to its front, to connect all the wires and components.

Diagram A allows to see what our electrical circuit should look like (including how you would integrate Bluetooth if you wanted to). Because this is a bit difficult to visually project onto the hat, I've included **diagramB**. Makes it look as easy as it is to make!

- Make a small incision at the top of the hat, approximately 3cm wide. Carefully burn the edges with a lighter or stitch them, to avoid fraying of the fabric.
- Make a small incision under the top level of the hat flap, so it remains invisible.
- Place the Neurosky chip on the right or left side of the front of the hat, as you feel works best. The electrode wire should not be bent too much.
- Connect the incoming GND lead that is soldered onto the JST breakout board on the back of the hat to the GND lead coming from the pompom. Solder a new wire to the SW +, which you then connect to the "square pin" of the Neurosky chip (see photo).

- Solder a new lead to the GND coming from the pompom, and connect this lead to the pin on the Neurosky indicate GND or "V pin" in the pic above.
- Solder the TX from the pompom to the "T" pin on the Neurosky chip.

Mode Switch

- Place the mode switch on the side of the hat. Attach one lead of the tactile push-button to the "GRD" node, and the other to the mode lead coming out of the pompom.

Finishing touches

- Sew all wires onto the hat, securing them in place. Sew on some small snaps on the inside of the hat flap so that it doesn't open when you are wearing it.

Step 11: Testing Functionality

Once the "Neopixel cocoon" has dried and is completely solid, connect the JST pin female connector to the male connector at the top of the hat. Test the overall resistance of your frontal electrode, then the power outputs of the voltage converter and battery. Check if the Arduino and Neurosky chip receive the proper voltage (you don't want to fry anything after your hard work!). If everything looks normal then you are all set to go! You can now run some functionality tests to check whether everything is running as it should be.

!!! As a safety precaution, never, EVER wear the hat when it's plugged into your laptop AND the laptop is plugged into the wall socket. Keep the laptop running on its batteries. Please be careful :)

- Turn on the hat via the switch on the JST breakout board.
- Put on the hat, and make sure there is good contact between your forehead and the EEG electrode.
- The lights in the pompom should start blinking in red at this point, indicating that your reference electrode is not attached.
- Place an adhesive electrode over the mastoid bone behind your reference ear.
- Click on the sensor wire head onto the electrode's snap. If you are using an earclip, simply clip it to your earlobe. The lights should stop blinking red, and instead go off for a few seconds. The Neurosky chip is, at this point, receiving the brain signal and starting to process it.
- Observe as the light begins to flicker in a specific color. Take slow deep breaths, relax, and watch as this color changes. The intensity and brightness of this color indicate the amount of relaxation.
- Try focusing hard on a mental math exercise. Watch as the color turns the opposite of the relaxation color!

This is the part of the tutorial where you get to do mental partial integration while listening to Tibetan throat singing, while wearing a hat with blinking neon Neopixels on your head. Can't skip this unique experience :) Turn on the captions in the video for explanations!

Debugging... maybe the hat is trying to tell you something?

If the hat doesn't seem to be functioning correctly, check all the electrodes: are they well connected? Is the battery recharged (basic question but I made the mistake too)?

Flashing red lights = Please connect the reference! If the signal quality from the reference electrode is bad or absent, the Neopixels will blink in red. Check that you have well connected the reference electrode to your mastoid or earlobe.

Flashing yellow lights = No incoming data string. The Arduino permanently tracks the serial output of the Neurosky chip. If at any point there is no incoming string of data, it will complain by flashing all Neopixels in yellow color. This means that there is either no EEG signal, or something is wrong in the communication between the Neurosky chip and the Arduino. Check whether there is proper contact between the EEG electrode and your forehead.

Still doesn't work? Check all your soldering points: is one broken? Use and abuse the heatshrink to protect these fragile points! If you're really in a crisis, you can send me an email or a message and I will try to help you.

Step 12: Putting the Neopixels Into a Pompom

Now that your hat is working, you can finish the pompom. If you wish, you can 3D print a capsule as a casing for the "Neopixel cocoon". White yarn is best, as it diffuses light well. I've tried with other colors, but you won't get the same beautiful effect. There are many youtube tutorials on [how to make a yarn pompom](#). Before you start, remember that the central hole during the construction of the pompom has to fit the cocoon structure, so that when you cut the cardboard, it holds the Neopixel cocoon in place. Don't make the pompom too fat, don't make it too thin. It's better to use a lot of yarn, so that the pompom is dense. The Neopixel glue cocoon should not be clearly visible between the yarn. Of course, I learned this by trial and error, trust me you don't want to have to make a pompom ten times in a row.

- Once you've got your cardboard circle full of wound up yarn, secure the Neopixel component inside the pompom.

- Make sure the Neopixel cocoon isn't visible from the outside, and that the mini/micro USB slot is accessible at the base.
- Apply a bit of glue to keep it snugly in place.
- Practice making a pompom first before you make the final one! Cut the hairs of the pompom evenly around the Neopixel structure. Since the Neopixel structure is glued in, if you get the hair cut wrong, there's no going back: you'll have to make an entire new Neopixel cocoon. If you cut more hairs on one side (like I did), the light will not diffuse evenly. The best way to do this is to run on the LEDs, so you can see how the light diffuses while you cut the yarn.

Congratulations! if you got this far you've made it! Now you can play with the hat and make the colors oscillate :)

Here's a short video demo. Brightness was not on maximum. Greenish-blue is attention color, orange is the relaxation color.

Safety Note: All the soldered bits are encased in glue, and then Sugru, which are isolating. There should be nothing but the USB slot and the Neopixel faces that are in direct contact with the yarn. The risk that your hair will burn off due to an pompom fire is quasi non-existent. Don't worry :)

Step 13: Mode Selection

The Mode selection

The push-button on the hat or "Mode switch" allows you to change the "attention color" and maximum brightness of the Neopixels. The "relaxation color" is automatically chosen by the software for you (as explained previously in step 8).

Click once to go to the next mode. At the end of the mode menu, the cycle will automatically go back to the beginning. If you do not click, the hat will simply stay in that mode. Watch the videos above, they will help you. Turn on the captions!

Step 14: Use the Hat to Control Graphics in Processing

Now that you've made your hat, you can play around with it in [Processing](#). Processing is a free and open source programming language which allows you to create beautiful visuals by using the brainwaves to change variables such as colors, shapes and movement of graphics. The website is well documented, with extensive tutorials and a support community. It's intuitive and fun to work with, even if you have little experience in coding. You can create fantastic visuals with minimalistic code.

- Make sure your laptop is running on batteries only, then plug the mini USB into the pompom of the hat. If you made your hat with bluetooth, you won't need a USB cable, however you will need to adapt the Arduino firmware to work with bluetooth (in this case you may need to use a different Arduino with more RAM).
- Switch on the hat.
- Wearing the hat, plug the other end of the USB cable into your computer.
- Open your Arduino IDE. Make sure you've chosen the right serial port (COM). It's most often the lowest number. Also choose the right board (for TinyLily select "Arduino Pro or Pro Mini 3.3V 8MHz ATmega328"). If you get an error, un-plug the USB and plug it back in again.

```
// serial import for Processing sketches.
// Copy this code into your sketch and customize it.

// at the very beginning of you sketch
import processing.serial.*;
Serial myPort;
String inString;
// The serial string contains the following information in the following
order:
// battery, relaxation percentage, attention percentage, delta, theta, low
alpha, high alpha, low beta, high beta, gamma

float batt; // battery level
float sig; // signal quality
float att; // attention percentage
float relax; // relaxation percentage
// you can also visualize the different frequency band activity, however be
sure to only compare these amounts relative to one another. These values are
outputted as "unitless" from the Neurosky chip.

// in the setup
void setup(){
  println(Serial.list());//
  myPort = new Serial(this, Serial.list()[0], 57600); //change this to your
serial port.
  myPort.bufferUntil('\n');
}

// include a serial event
void serialEvent(Serial myPort) {
  String inString = myPort.readStringUntil('\n');
  if (inString != null); {
    inString = trim(inString);
    float[] val = float(split(inString, ","));
    if (val.length >=3) {
      batt = val[0];
      relax = val[1];
      att=100-relax;
    }
  }
}
```

Step 15: Tweaks and Other Things

What's next?

I am continuing with several projects: instructables tutorials, lighting installations, and custom hardware. If you are interested in following these projects as they take form, you can visit [this website](#).

Build a hat with kids (or if you are a teacher, with your students)!

The illumino hat is a great for teaching kids about human neurophysiology in a fun and interactive way. Plus, if you build the hat with them, they get a hands on experience with DIY electronics. I suggest using a colorful hat that's not as boring as the gray beanie in this tutorial. If you're hesitant to put a lithium battery on a child's head (although lithium batteries are sturdy), you can make a shock proof plastic case for it.

And before you go, don't forget to take a look at these other DIY EEG Instructables tutorials:

- [Adding Bluetooth to an EEG headset for mind-controlled projects](#)
- [How to hack EEG toys with arduino](#)
- [DIY EEG \(and ECG\) Circuit](#)
- [Control any gadgets with your Brain!](#)

Share

Recommendations