SUMMER 2021

#GIRLSINSTEM

GIRLS IN STEM

TUTORIALS HANDBOOK

E-Textiles Toolkit

GIRLS IN STEM

EU PROJECT

The aim of this project, funded by the European Commission through the Erasmus+ programme, is to empower girls to pursue their interests in Science, Technology, Engineering and Maths (STEM) subjects.

The project will provide young girls and women with support and opportunities to experience the potential of studying STEM subjects and embarking on a fulfilling career in these disciplines.

The project has 4 main actions.

The project team will develop three toolkits for educators, focusing on three subjects; reverse engineering, e-textiles and digital art. These toolkits will provide all the information and resources for educators to be able to facilitate activites with young people on these subjects, in their own environments.

These toolkits will form the basis of three bootcamps which will take place during the Summer of 2021. These are international bootcamps bringing together young girls and women from across Europe. Each bootcamp will focus on one subject and provide the opportunity for participants to engage with the activities being developed.

The project will also lead a mentoring programme for young females to deepen their knowledge for one specific STEM area. The programme will be implemented in partnership with local NGOs, enterprises and institutional environments, where volunteers from such organisations can accompany the youth to discover more about the areas they work in.

Find out more about the project through the website at:
GIRLSINSTEM.EU

Above: Digijeunes' Youth Workshop at Luchon MJC, France. Feb 2021.

CONTENTS

Before getting started..

This toolkit aims to allow any youth worker to facilitate activities with young people on the subject of E-Textiles. For this, introductions are provided on the; equipment, techniques and components, that will be used throughout the activities and tutorials.

Get making!

With the background knowledge ready, it's time to get making! These sections provide detailed tutorials on how to complete E-Textile activities. We have aimed to provide a range of activites that vary in length of time to complete, complexity and required equipment. For each section; Build the components, Mini projects and Project challenges, there is a page detailing this information for facilitators. Enjoy!

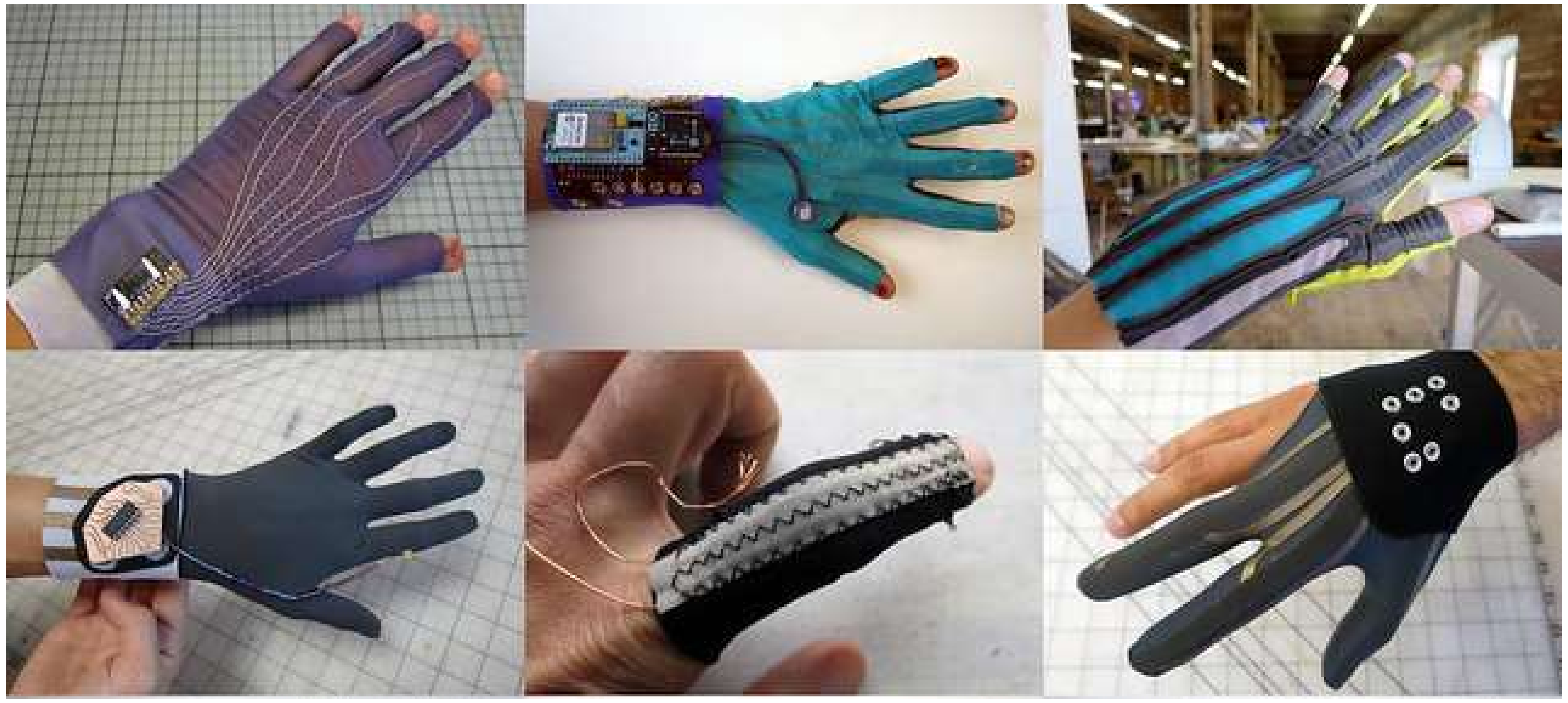
These tutorials focus on building electrical components out of textiles. These are the simplest tutorials, requiring less time and equipment.

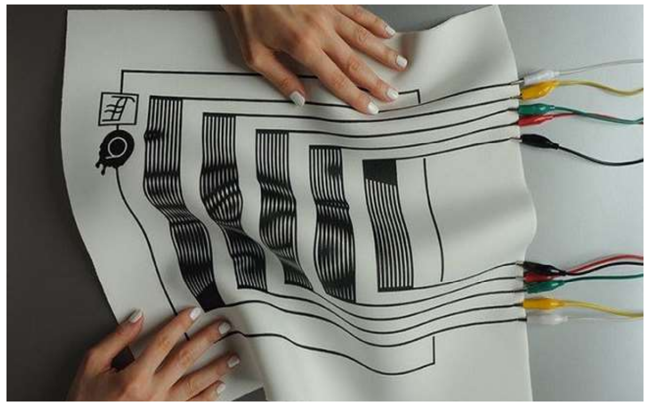
Mini projects provide an opportunity to create something that tests skills in electronics, sewing and soldering. These projects will engage young people for over multiple sessions.

For the Porject Challenges, it is necessary to have a basic understanding of Arduino. This section briefly explains what Arduino is and how it can be setup and used. A few starter activities are also provided to help beginners get to grips with it.

These are complex projects which will challenge a participant's skills and technical understanding.

INTRODUCTION TO E-TEXTILES





Top: E-Textile data gloves by Kobakant (Kobakant.at) .
Bottom left: The Language Game Series Costumer by Kobakant (Kobakant.at).
Bottom Right (top): Flexible electronics and smart textiles (hyperegio.eu).
Bottom Right (bottom): All-Inkjet-Printed Conductive Electronic Textiles (Manchester University).

"What makes smart fabrics revolutionary is that they have the ability to do many things that traditional fabrics cannot."

E-Textiles (Electronic Textiles) is the combination of textiles with electrical components. E-Textiles is the embedding of electrical components, such as batteries, lights or sensors, into textiles so that fabric and sewing techniques can be used alongside electronics to enhance a product, piece of art or item of clothing.

E-textiles can be found in an array of contexts, including smart clothing and wearable technologies. However, e-textiles do not have to be wearable, as they have also been seen in other contexts such as interior design or interactive installations.

The phrase 'smart textiles' or 'smart clothing', is used to define items that have the ability to assist their wearer. Smart textiles are likely to use a range of integrated sensors which receive information from their surroundings. Common inputs can include temperature, movement or light. These inputs can then be analysed to trigger the textiles, item of clothing or integrated components to react and behave in response.

As E-Textiles have the ability to identify a change in the environment, they can be incredibly useful in contexts where monitoring of a person, their movements, physical health or environment is important. Common areas of research for E-textiles and wearable technologies at this time include sports and fitness, supporting older generations and helping individuals with severe health conditions. Integrated wearable sensors are less intrusive to the wearer, therefore less likely to impact their movements or everyday activities. At the same time, they are able to trigger warnings or actions which can relieve pain or guide their actions and routine.

In this handbook, activities are provided to demonstrate how electronics can be integrated with textiles and how sensors or actuators can be produced. The activities involve E-textile equipment and materials such as conductive thread, conductive fabric and sewable electronic components. The aim is for these activities to be interesting and informative for young females, as well as capture their imagination about what is possible when using electronics outside of more commonly known contexts.

INFORMATION FOR FACILITATORS

"We want people to be able to reach a point where they become independent thinkers and self sufficient builders."
Fran Sanchez - Beachlab

This handbook hopes to provide all the information and resources for any educator, with a background in electronics, textiles or neither, to be able to facilitate activities on the subject of E-textiles.

Step by step detailed instructions are provided for each activity. With the support of colourful and clear illustrations, which can also be shared with the young participants.

This booklet provides supporting information on techniques and components, to help a facilitator to answer any questions that may arise from curious young minds. Having a greater understanding of the tools and equipment, such as the electrical items and the Arduino board, will help facilitators in feeling confident to lead successful and engaging workshops.

Going through the booklet, the complexity of the projects increases. It is advised to work through the booklet section by section as this allows participants' understanding and confidence with different techniques to increase in a natural manner.

The tutorials are split into three sections. For each section there is an introductory page for facilitators. These pages provide details on the activities such as; their level of difficulty, their cost and time requirements.

The first group of activities is based on building components. The second are mini projects which produce an end item and finally the third section gives advanced project challenges that include an element of programming with Arduino. All the activities are built upon the basics of physics, electronics and programming .

The integration of such skills with textiles hopes to bring these subjects into light for young females who may typically find such subjects to be 'not for them'. The activities also provide an element of creativity and design, which hopes to appeal to a greater audience of young people.

PREPARING A WORKSHOP

When setting up a workshop, it is important to prepare and know the objectives of the activity that you will be undertaking with the group. Before the workshop make sure to have:

- All the equipment available.
- Adequate space.
- Have defined and achievable learning outcomes for the session.
- Have ran through the activity to check that you can follow the steps and have identified where common errors may be made by the participants.
- Have an example of what they will create (either physically or through images) to show them before starting.

Above: Digijeunes workshop with Youth Group in Luchon, France. Feb 2021.

DEALING WITH FRUSTRATION

Working with electronics can be frustrating. It is small, fiddly and not always obvious why something doesn't work as it should. Working with textiles also requires patience and a lot of concentration. For some participants, this may be challenging, and their attention may not remain focussed throughout the whole session. To aid participants with this:

- Break the activity down into small manageable steps. The bigger the steps, the more likely it is that a participant will feel overwhelmed and give up on the task in hand.
- Provide positive encouragement. If participants are struggling on a task, reflect back on the steps that they have already achieved and encourage them to continue trying.
- Encourage teamwork. If a participant has managed to complete a challenging task, ask them to explain to others how they managed to do it and work together for others to complete the step too.

Frustration, especially when trying new things, is common. It is important for facilitators to help participants accept the frustration they feel and be able to manage it, and still carry on. This is a great skill for participants to learn, especially for later in their studies and future careers.

DEALING WITH QUESTIONS

It is not necessary for the facilitator to understand everything about the activity, the background knowledge and equipment. It can be very empowering for young people to also have the facilitator to be learning alongside them. This booklet aims to provide the minimal amount of knowledge to be able to execute activities, but there will still be room to learn and problem solve together. When questions arise that are outside of the information in this book, see it as a challenge for the participant to find out and discover the answer for themselves. If you don't know the answer, it's best to be honest and participate in the research!

HANNAH PERNER-WILSON

ARTIST AND DESIGNER

Hannah Perner-Wilson develops new ways to integrate electronics, conductive materials, textiles and craft techniques. Along with Mika Satomi, she has created KOBAKANT which is an collective which shares all their discoveries and projects, openly online.

She received a BA in Industrial Design from the University for Art and Industrial Design Linz and an MA in Media Arts and Sciences from the MIT Media Lab, where she was a student in the High-Low Tech research group.

Many tutorials in this kit have been inspired by her work. You can discover more through their website.

WWW.KOBAKANT.AT

WOMEN IN E-TEXTILES

Inspirational women who have carved a career in E-Textiles. Hannah, Vanessa and Becky are all at the top of their respective fields using E-Textiles in various different ways. They present how broad the subject is, how complex and how influential it could be in the future.

VANESSA SANCHEZ

MATERIAL AND FABRIC SCIENTIST

Vanessa is a researcher in Material Science and Engineering at Harvard University in America. Her research is focused on developing robotic fabrics for patients who are ill or disabled. She has created robotic fabric leg sleeves that treat deep-vein thrombosis, as well as soft robotic fabrics with built-in feedback control that can help users have better mobility.

Vanessa is at the forefront of discoveries for Electronic textiles and how they can be integrated in everyday life to support people. Find our more about Vanessa, her research and her work, through her website.

WWW.V-SANCHEZ.COM

RESEARCHER AND MUSIC ENGINEER

Becky is a lecturer at the Design School of Design Engineering at Imperial College in England. She has worked on many Engineering and consultancy projects, mostly around the subject of e-textiles and signal processing to build interactive, body-centric wearable computing systems.

She was also the Co-Founder of Codasign, a social enterprise focused on creative technology education. They ran workshops for children and adults about integration of technology and creative subjects such as music, performance and crafts. Find our more about Becky and her work on her website.

BECKY STUART

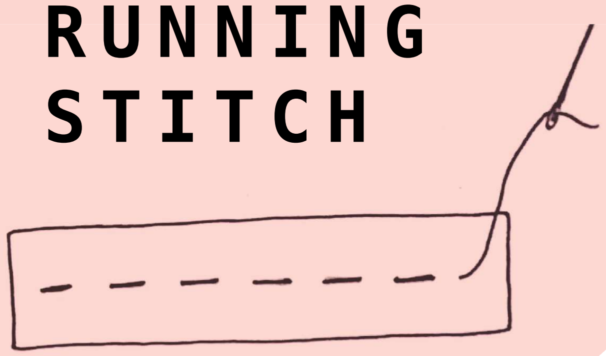
THELEADINGZERO.COM

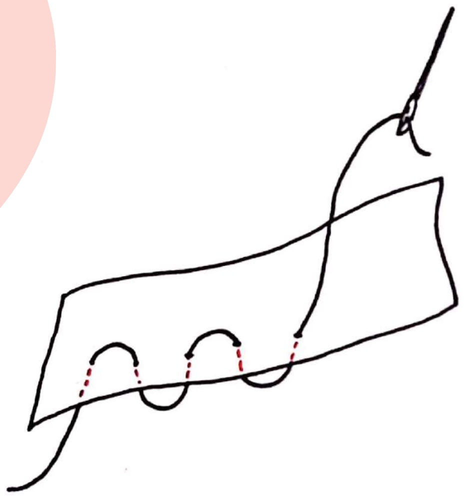
SEWING TECHNIQUES

Here are some basic stitches that will be useful during these E-Textile tutorials. They are very commonly used and videos of such titches can be found on YouTube.

BASIC TECHNIQUES

RUNNING STITCH

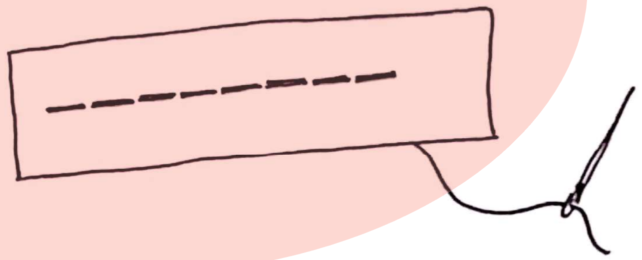


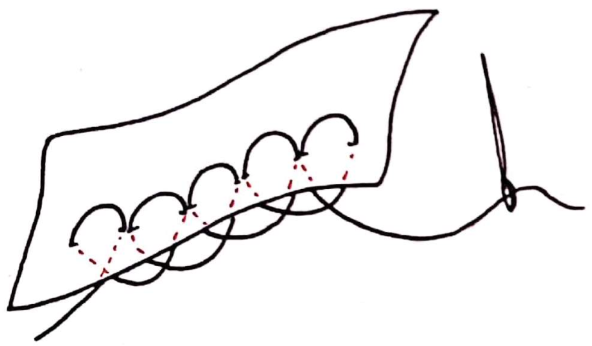


The simplest stitch to sew two pieces of fabric together.

BACK STITCH

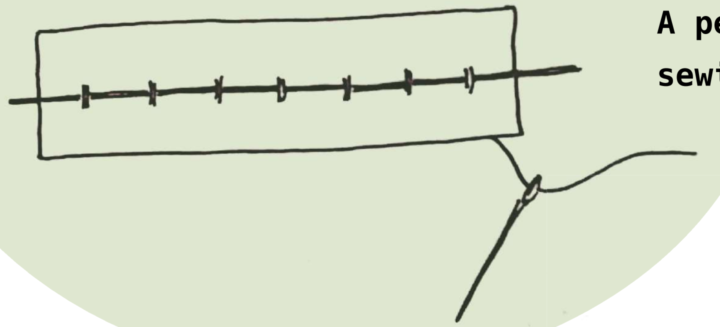
The most common stitch to use when sewing two pieces of fabric together. It’s neater and stronger than the running stitch.



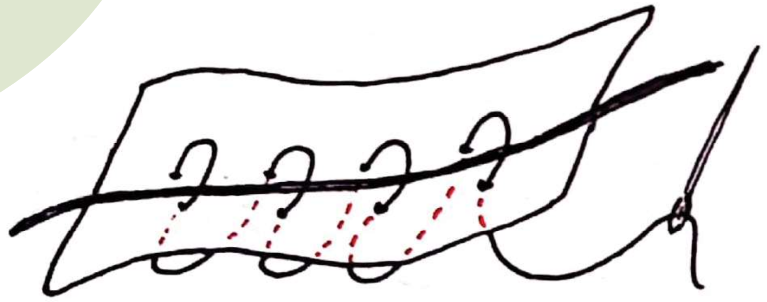


USEFUL TECHNIQUES TO KNOW

COUCHING STITCH

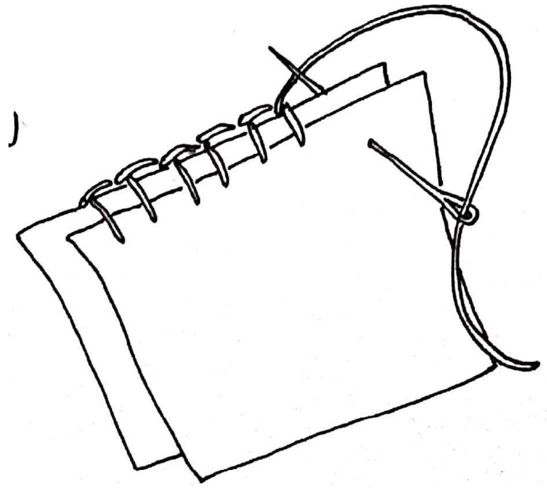


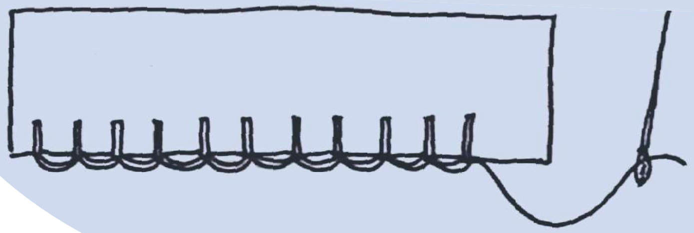
A perfect technique to use when sewing thin wire onto fabric.



The blanket stitch is great to connect the edges of two pieces of fabric. It is relatively quick and easy but forms a neat and decorative finish.

BLANKET STITCH

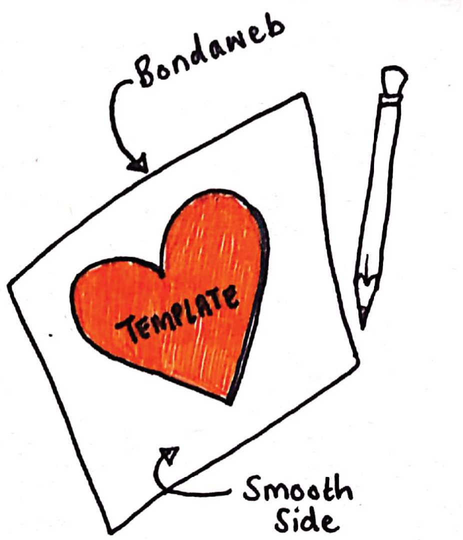




SEWING TECHNIQUES

APPLIQUÉ

Applique is a method that is used to sew patterns, shapes and images made from fabric, onto a second piece of fabric.



1. CREATE A TEMPLATE

Create a template of the shape you want to sew onto the base fabric.

Draw round the template onto the SMOOTH side of the Bondaweb

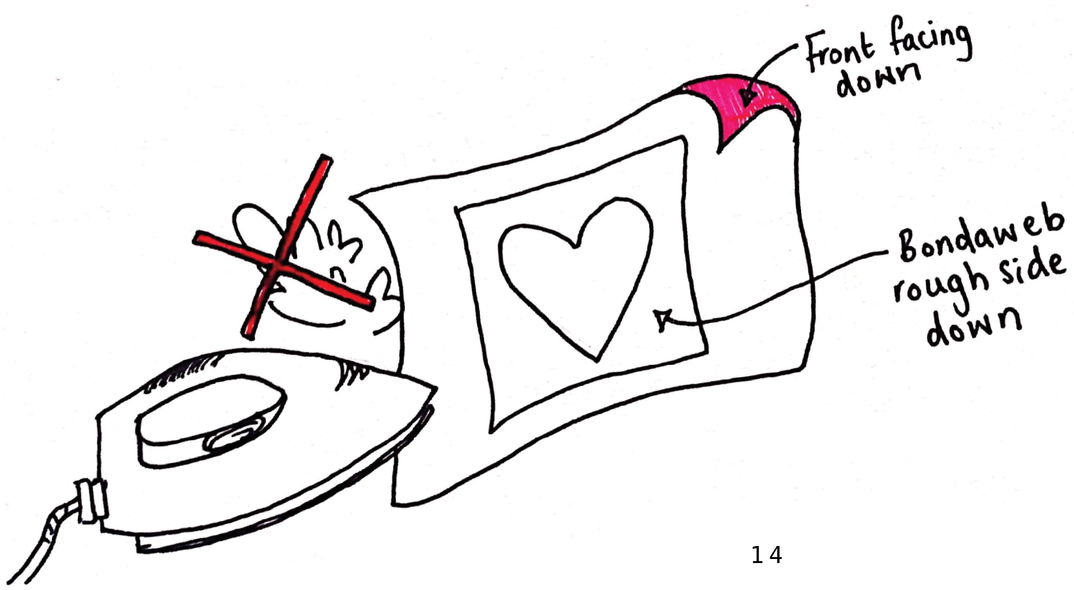
(If you are doing letters or numbers, make sure they are back to front so they are the correct way round later on).

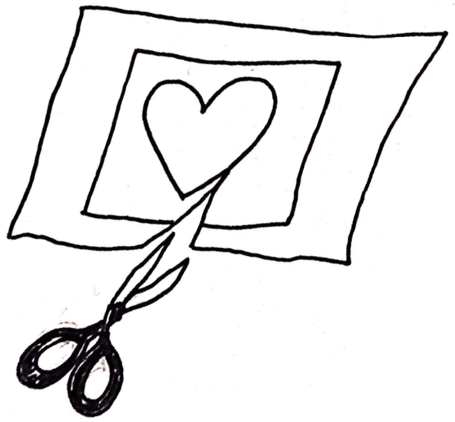
2. MAKE THE DESIGN

Take the fabric you want the shape to be made out of.

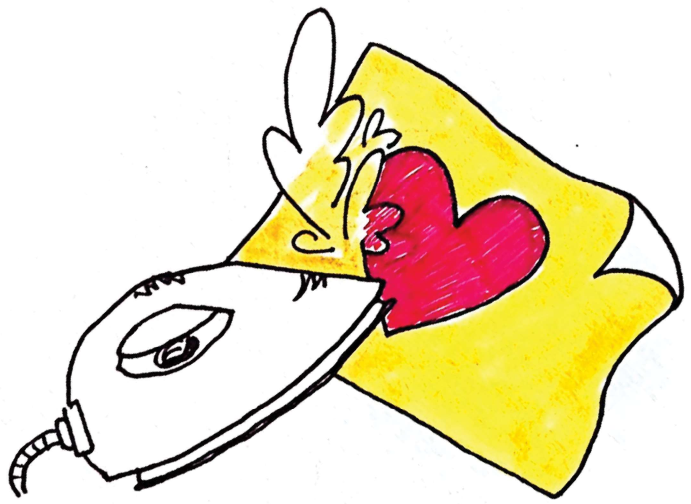
Dry iron the Bondaweb piece to this fabric (don't use any steam). The rough side is down and the smooth side, with the tracing, is facing up).

Then, using the pencilled trace, made at the beginning, cut out the desired shape





3. IRON ONTO THE BASE FABRIC



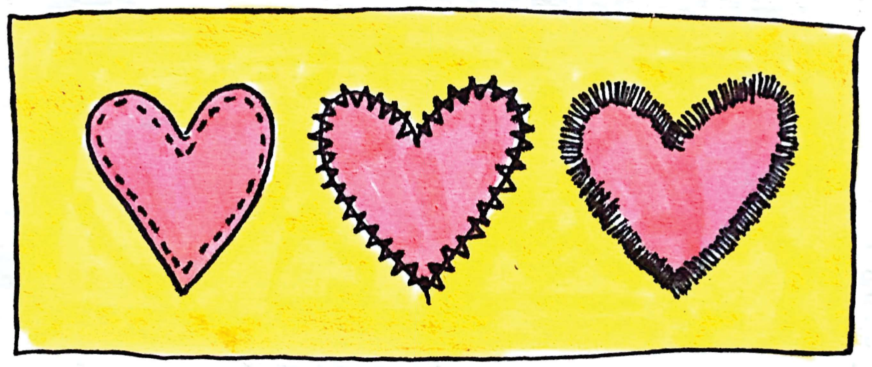
Peel of the back layer of the Bondaweb and position your piece onto the base fabric with this side facing down. (If it is a letter it should now be the correct way around).

Iron over this piece so that it sticks to the base piece. (This time you can use steam whilst ironing)

4. SEW ON THE DESIGN

Using a sewing machine or by hand stitching, stitch around the outside of the shape. This helps it to stay in place and will prevent the edges from fraying.

This also gives a decorative finish, with each of the three stitches commonly used, giving a different look. These are shown below:

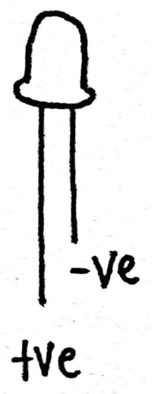


RUNNING STITCH | ZIG ZAG STITCH | TIGHT ZIG ZAG STITCH

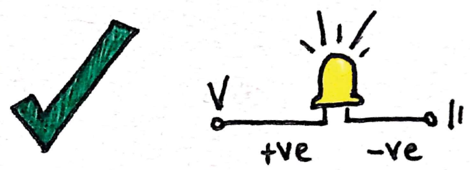
UNDERSTAND YOUR COMPONENTS

LED

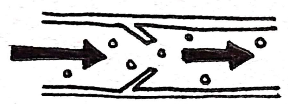
LIGHT EMITTING DIODE

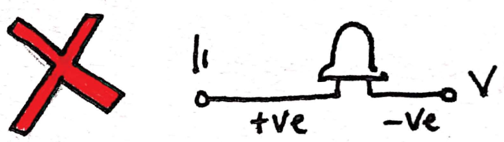


The current can only flow in one direction through an LED. This means that the positive and negative terminals need to be connected correctly.

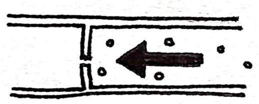


When the legs of the LED are connected in the right way, the current can flow through causing the LED to light up.



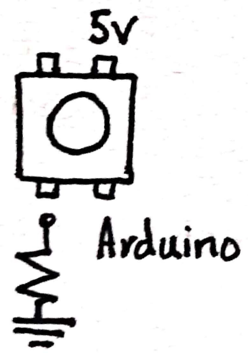


Connected incorrectly, and the LED won't light up as the current will be travelling in the opposite direction and won't be able to pass through.

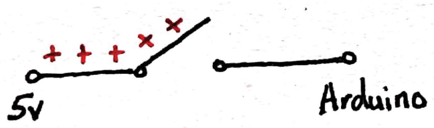


BUTTON

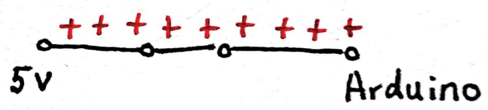
ELECTRICAL SWITCH



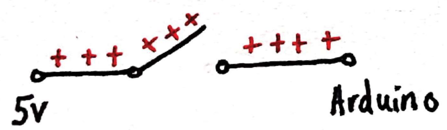
The charge is unable to pass through if the button is not pushed. In this state, no input signal will be recieved by the Arduino.



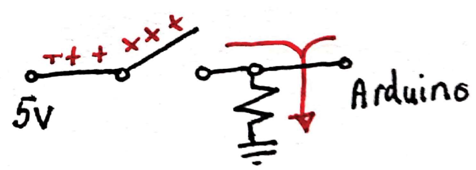
When pushed the charge can flow through to the Arduino connection



When released, the circuit breaks but the charge is still held within the circuit, still providing an input to the Arduino

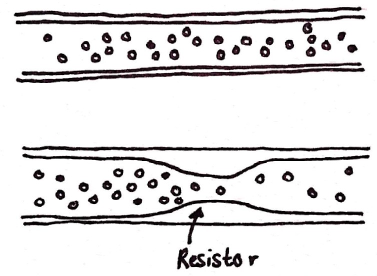


Therefore putting a grounded resistor between the button and Arduino means that the charge can be drawn away when the button is released.

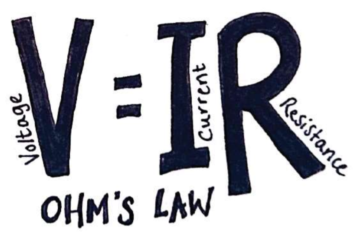


RESISTOR

A resistor is an electronic component that restricts the flow of electric current (by resisting the flow of electrons around the circuit). The most common type of resistors do this by either being made from a less conductive material, making the conductive material thinner, or increasing the length of the conductive material.

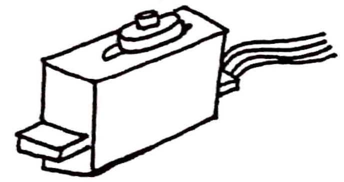


By reducing the current that flows through the circuit, the resistor also reduces the voltage across the components in the circuit. This is thanks to voltage, current and resistance being related to each other through Ohm's law.



Resistors can have a resistance value which remains the same or varies due to an additional variable (e.g time or temperature) . A common variable resistor is a potentiometer.

SERVO MOTOR



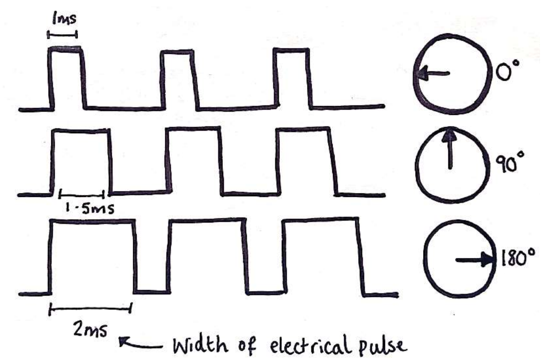
Servo motors are known for being small in size but very energy efficient. They are commonly used to operate remote controlled toy cars and are also used in industrial applications, such as with robotics.

Inside a servo motor there is;

- a small DC motor,
- a potentiometer (a variable resistor),
- a control circuit.

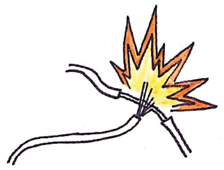
As the motor rotates, the resistance of the potentiometer changes. This allows the control circuit to know exactly how much the motor has turned by, and in which direction.

Servos are controlled by sending an electrical pulse of variable width, also known as a pulse width modulation (PWM). The PWM, which is sent to the motor, determines the position of the shaft based on the duration of the pulse. The diagram gives an example below.



UNDERSTAND YOUR EQUIPMENT

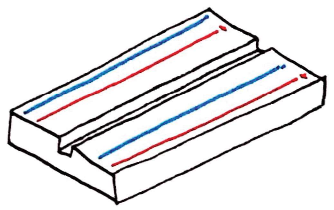
SHORT CIRTCUITING



Short circuiting can occur when wire connections in a circuit are damaged, loose or exposed. The current in the circuit, always seeks to flow through the least resistant pathway. Shortcircuiting happens when there is a less resistance route available. This can be very dangerous, as the least resistant pathway could be through a flammable substance or human body.

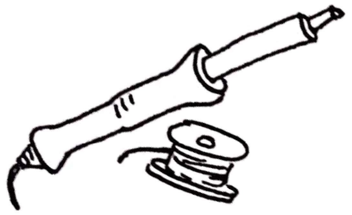
In these tutorials we are working with very small voltages, so a short circuit is unlikely to be dangerous but it can damage the circuit. or components. The low resistance connection can cause an excessive current flow. This makes it important that no wires are left exposed, and if they are (or not insulated), they should not cross or have the possibility of touching.

BREADBOARD



The breadboard is used for electronic prototyping. It allows circuits to be built without any soldrering. On the central part of the breadboard, connections are made horizontally and not vertically. Where as along the sides, the connections are made vertically, following the coloured lines.

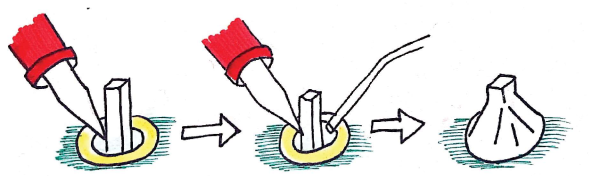
SOLDERING



Soldering is the process of joining two metals together, using solder, to make an electrical connection.

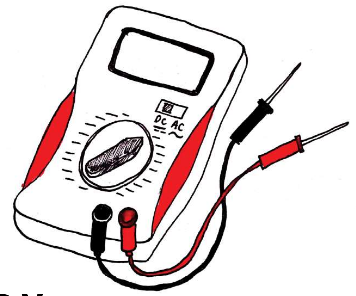
1. Make sure all parts are clean,
2. Clean the tip of the hot soldering iron on a damp sponge,
3. Add a very small amount of solder to the tip of the hot soldering iron,
4. Heat all the parts which want to be joined together,
5. Once all the parts have reached a high enough temperature, keep the heat there and apply the solder to the components (not to the iron)
6. Allow the solder to melt onto the components and form the joint
7. Remove the heat
8. Make sure the solder has cooled before moving the components

A good solder connection will have formed a pointed mountain shape, rather than a bubble of solder as shown below.

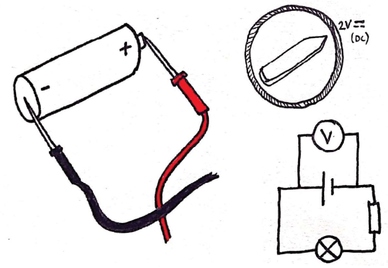


MULTIMETER

Common uses for a multimeter are to learn about electronic components, see if a circuit is working or test a battery.
The selection knob allows the user to define what the multimeter will read.
The two probes, one black (negative/ground) and one red (positive) and use to connect the multimeter into the circuit.

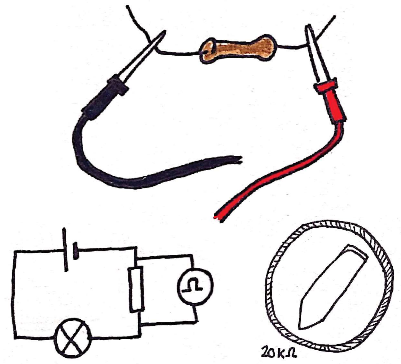


MEASURING VOLTAGE ACROSS A BATTERY



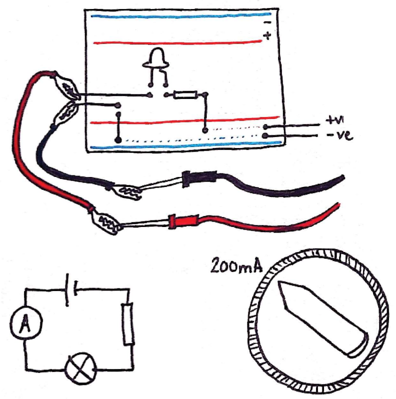
Set the multimeter to 2V in the DC range (batteries use DC - Current). If you are testing a battery greater than 2V, select 20V.
Connect black probe to the battery's ground (-) and the red probe to the power (+).
With a fresh AA battery you should get around 1.5V on the screen.

MEASURING RESISTANCE



You can check the value of a resistor or see if a resistor is damaged by testing it‘s resistance with the multimeter.
Select the 20kΩ setting and press the probes against the resistor legs.
If the meter reads a number such as 0.46 this means the resistor has a value of 460Ω (you need to multiply the result by 100 as the result is being shown in kΩ).
If the meter reads 1 or OL, it‘s overloaded - you need to try a higher mode such as 200kΩ. If the meter reads 0.00, you need to try a lower mode such as 2kΩ or 200Ω

MEASURING CURRENT



You have to measure current in series which makes it a bit trickier!
There needs to make an interruption in the circuit where the current can be measured.
To do this, it is best to use additional wires and crocodile clips to pinch the probes.
Like with reading voltage and resistance, you then need to set the multimeter to the correct range. It‘s usually best to start with 200mA and building up if needed as overloading the circuit with a current can result in damaging the circuit rather than just an overload (OL) display like before.

Get Making!

WHAT'S IN THE KIT?

ELECTRONICS

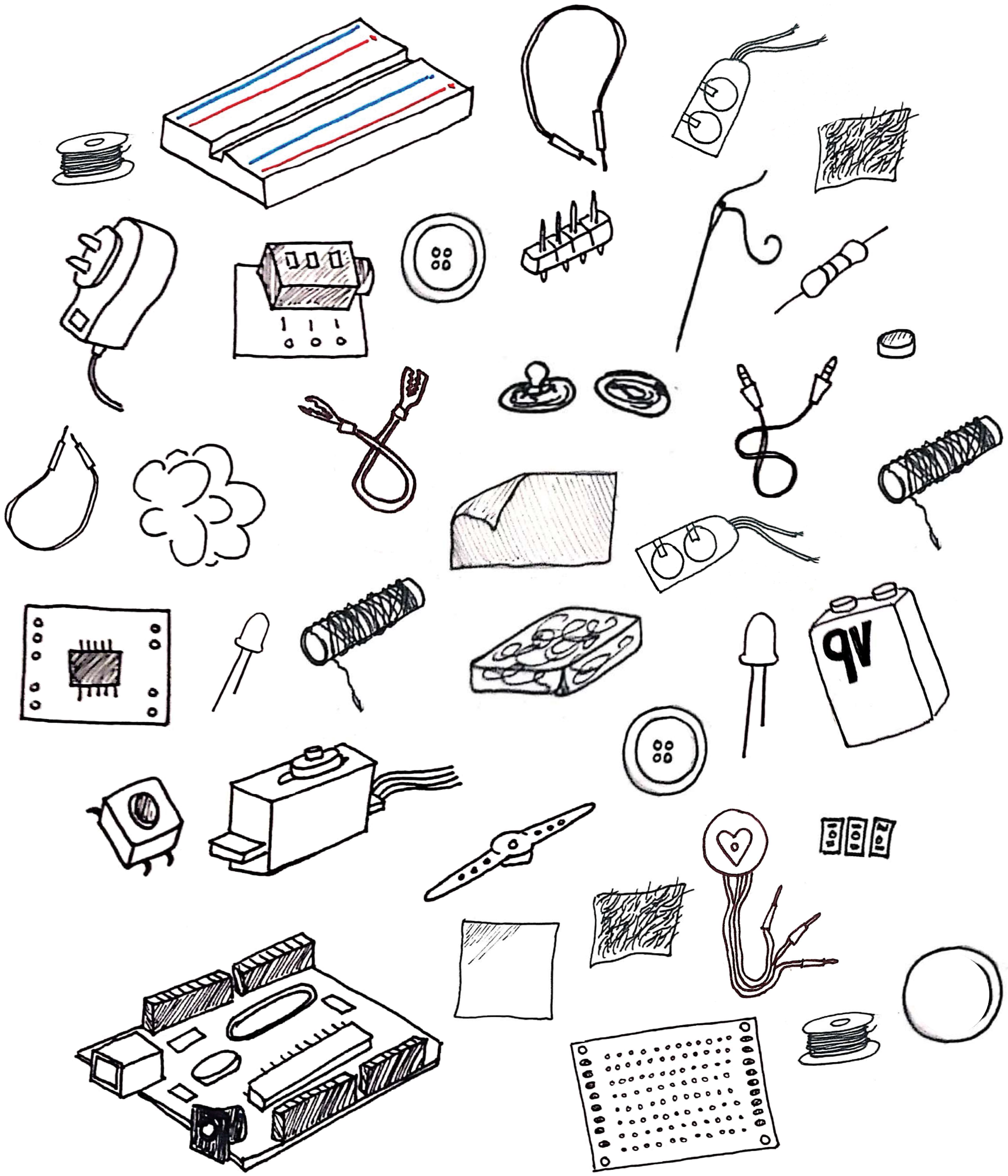
9V BATTERIES
AA BATTERIES
COIN BATTERIES
5V POWER SUPPLY
BATTERY CONNECTORS
SEWABLE BATTERY POUCHES

ARDUINO
PCB BOARDS
BREADBOARDS
JUMPER CAMBLES
CROCODILE CLIPS
PIN HEADERS

RESISTORS
BUZZERS
PHOTOSENSORS
COLOURED LEDS
MINI LEDS
BUTTONS
PULSE SENSOR

SPEAKER
AMPLIFIER CHIPS
AUX CABLE
AUDIO JACK PORT

THIN INSULATED WIRE
THIN COPPER WIRE
MAGNETS



TEXTILES

CONDUCTIVE FABRIC
CONDUCTIVE THREAD

COTTON FABRIC
FELT
THREAD
BONDAWEB
STUFFING

POLYMORPH
PLASTIC BUTTONS
METALLIC BUTTONS
POPPERS
NEEDLES
VELCRO
VINYL STICKERS

ALSO REQUIRED

SOLDER
SOLDERING IRON
GLUE GUN
PAINT
GLUE, FLOUR, WATER
NEWSPAPER
CELLOTAPE
SCISSORS
STANLEY KNIFE
PLIERS

WEBSITES

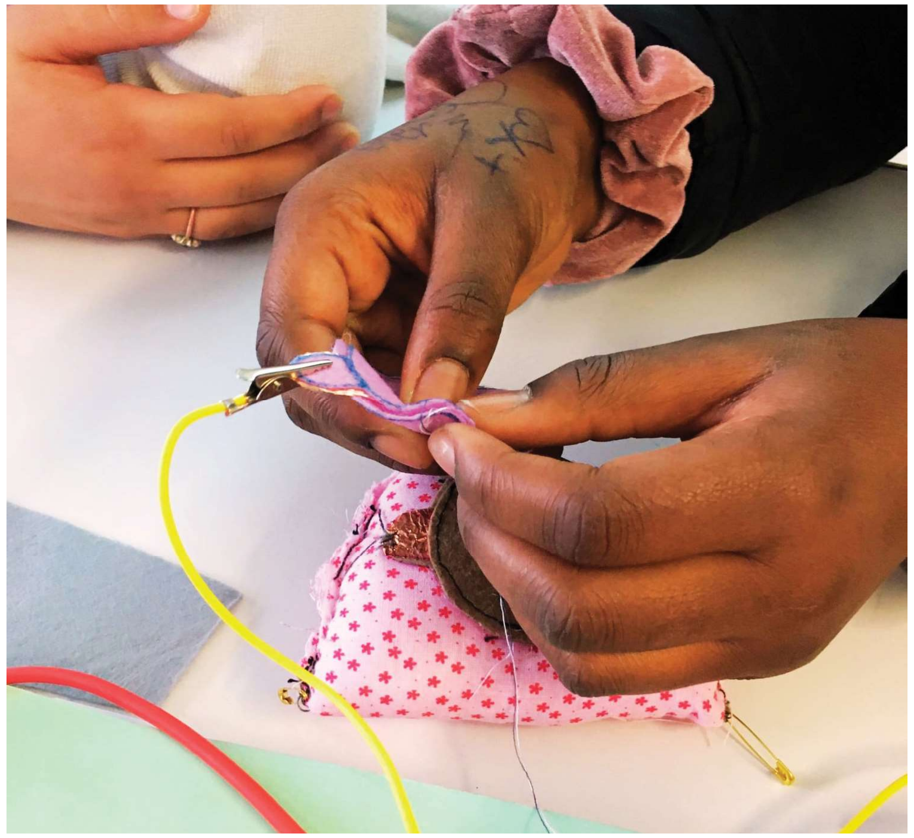
Good websites for odering components include:

WWW.BANGGOOD.COM
WWW.KITRONIK.CO.UK
WWW.RS-ONLINE.COM
WWW.ALIEXPRESS.COM

BUILD THE COMPONENTS

TEST CIRCUIT
FABRIC BUTTON
BUTTON BUTTON
BUTTON SWITCH
STROKE SENSOR
FLIP DOT EYE
FLAPPING WING

FOR THE FACILITATORS



This section focuses on building individual components from textile materials and sewing techniques. This is the best place to start if you are new to sewing!

Participants can build these components, play with them and see how they work. They are a great introduction to circuitry and understanding the basics about complete circuits and how current flows.

DIFFICULTY

These activities are aimed to be suitable for beginners. They are appropriate for children aged 11+.
For participants younger than 11, it would be advised to prepare the components and allow them to play and create circuits with them.

COST

All these activities are relatively cheap. They require minimal fabric and can be hand sewn. The most expensive material includes the conductive fabric and conductive thread. However the amount they use is minimal therefore only one piece of conductive fabric could be purchased for every 10 participants.

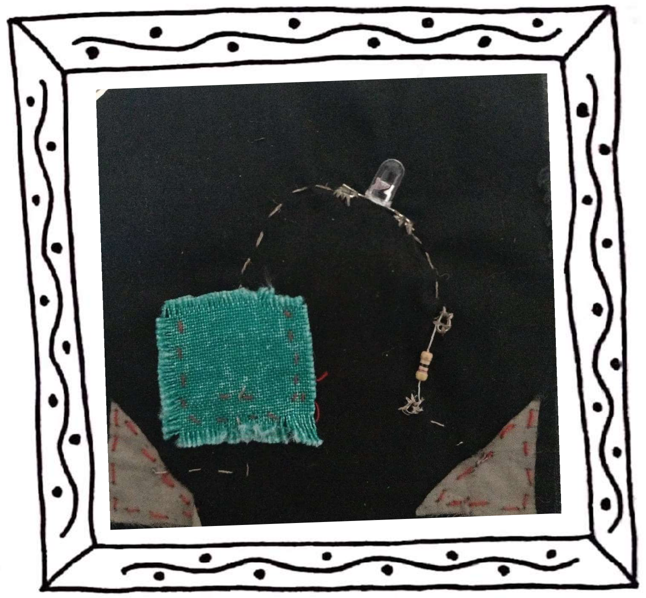
TIME

These activities are short and would be suitable for workshop or youth centre sessions. One component can be made in 20 to 40 minutes, making it great for one hour sessions or a walk in activity (if some preparations are made).
Together, if participants were to create multiple components each, this activity could be good for a half-day activity.

After building and playing with individual components, they could be used in bigger projects. Some ideas for projects which use these e-textile components include:

- Fabric game controller using Makey Makey
- Fabric keyboard using Makey Makey
- Interactive quilt (With different patches which you can interactive with in some way)
- Interactive piece of textiles art
- Interactive puppet (Shape of a puppet makes this quite complex)

Test Circuit



This circuit is to be used with E-Textile components such as the Button Button and Stroke Sensor. It allows you to see if your component works correctly.

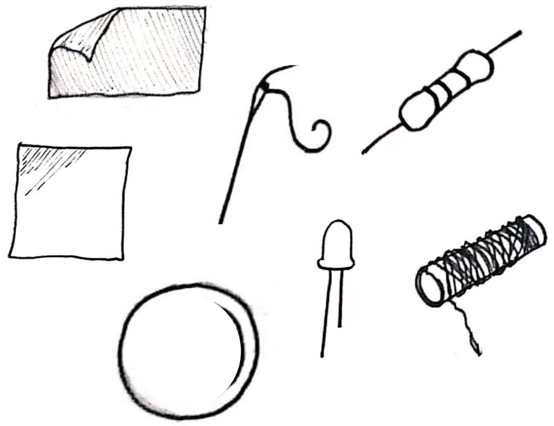
When connecting your component to the Test Circuit, if the component has been built right, the LED should light up when it is used.

What you need:

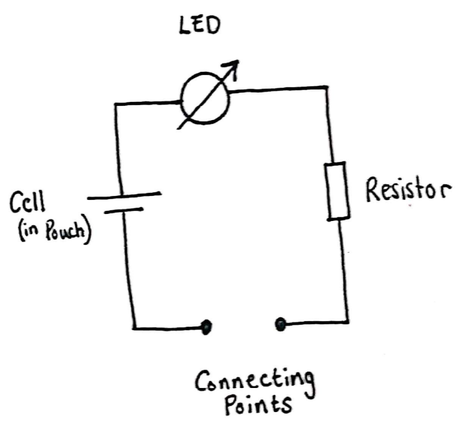
CONDUCTIVE THREAD
CONDUCTIVE FABRIC
NORMAL FABRIC
NEEDLE AND THREAD

LED
RESISTOR (47 OHM)

BATTERY CR2032



Circuit Diagram



How it Works

Test by touching the 2 triangles together and watch the LED light up!

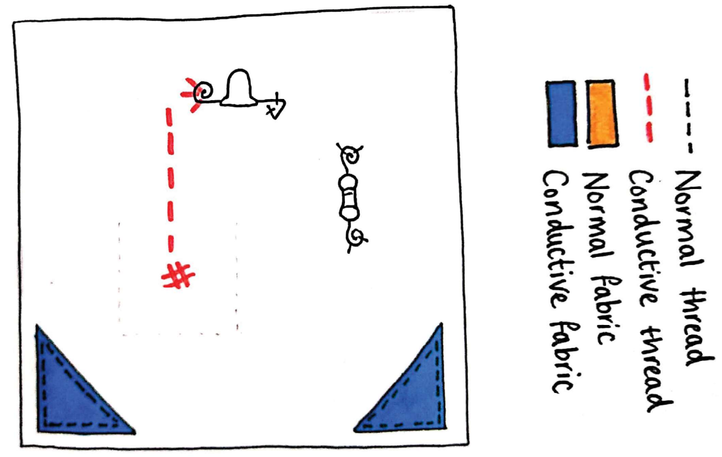
Now, using crocodile clips, connect the component you want to test to the triangles. This will complete the circuit and the LED will light up according to the component.

Build Illustration

STEP 1

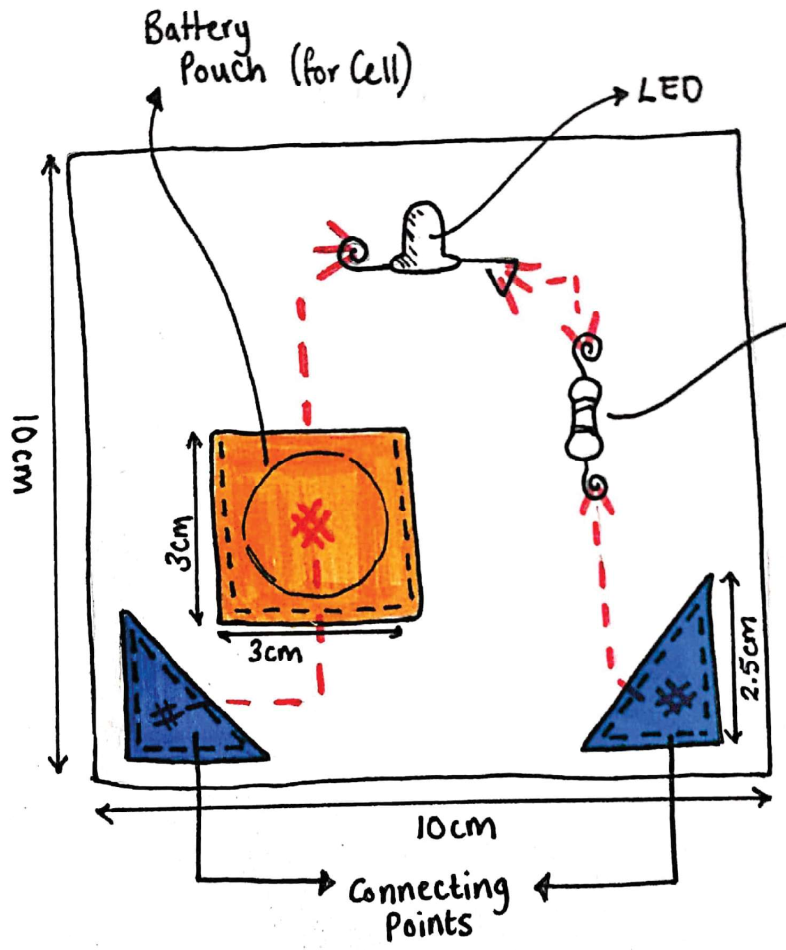
Cut out your piece of base fabric and two triangles of conductive fabric. Sew the triangles into position as shown in the image, using a running stitch.

Coil the ends of the LED and resistor. For the LED it is a good idea to coil the ends differently so that you know which side is positive and which is negative.



Sew the resistor and LED into place using normal thread. From one end of the LED use conductive thread to sew down and form a connecting point which will sit under the battery.

STEP 2



Using conductive thread, sew from the LED to the resistor. Then seperately sew from the resistor to the conductive fabric.

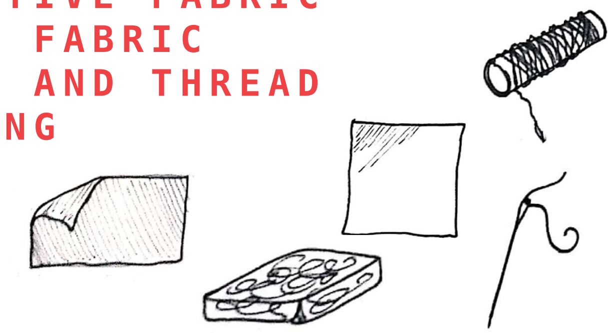
Place the coin battery on top of the connecting point previously stitched in Step 1. Make sure it is the correct way round for the LED. Cover the battery with a square piece of fabric and sew into place. It is best to sew precisely around the edge of the battery so that the fabric holds it firmly in position. Using conductive thread, sew from the top piece of fabric to the second peice of the conductive fabric.

Fabric Button

The fabric button is a simple two-state button which is made from textiles. It has two connecting points (positive and negative) which allows it to be integrated into a circuit.

What you need:

CONDUCTIVE THREAD
CONDUCTIVE FABRIC
NORMAL FABRIC
NEEDLE AND THREAD
STUFFING

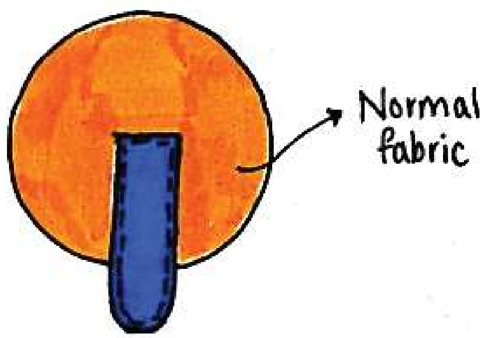


How it Works

Test by connecting the button to the tester circuit. The LED should light up when you push down on the button.
This is because, when pushed, the two pieces of conductive fabric make contact through the layer of stuffing. This completes the circuit so the current can run through and the LED can light up.

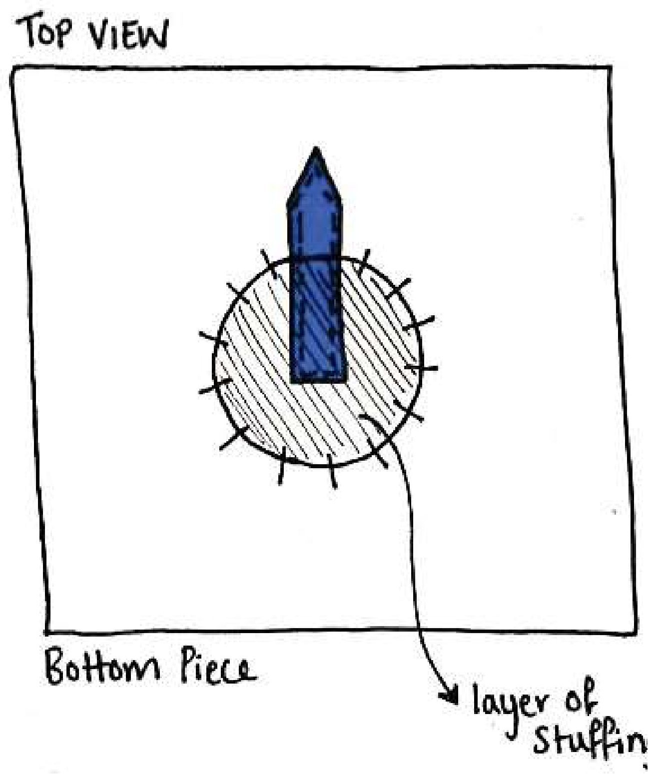
STEP 1

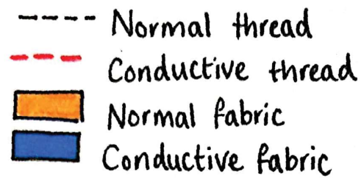
Cut a circle (or the shape you wish your button to be) out of normal fabric.
Cut a piece of conductive fabric in the shape as shown, which will reach the middle of the circle and the end poking out.
Sew or glue into place, on the back of the normal fabric.



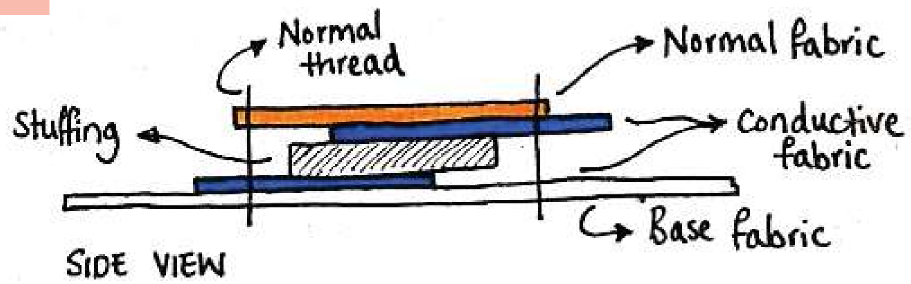
STEP 2

Next, cut another piece of conductive fabric in the shape shown.
Sew or glue where your button will be on your base fabric (the end will be in the centre of your button).
Cut out a circle of stuffing (very thin) which is slightly smaller than the size of the button you want.



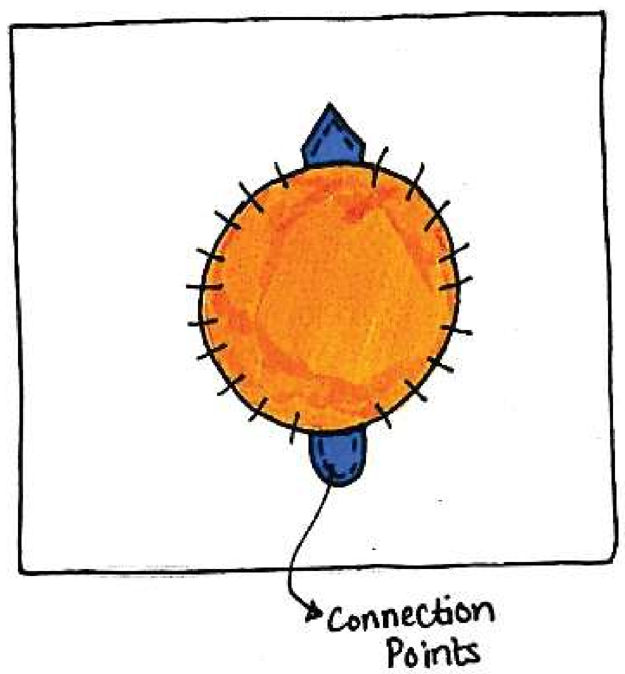


STEP 3



Position the piece you made in Step 1 on top of the stuffing.
Make sure that it is aligned correctly like shown. The ends of the conductive fabric should be pointing out at the ends.

Before sewing up around the edges, test that it works with your test circuit.
When you push on the button the LED should light up. If it doesn't your layer of stuffing may be too thick and you should remove some.
Sew up your button around the outside of the fabric.



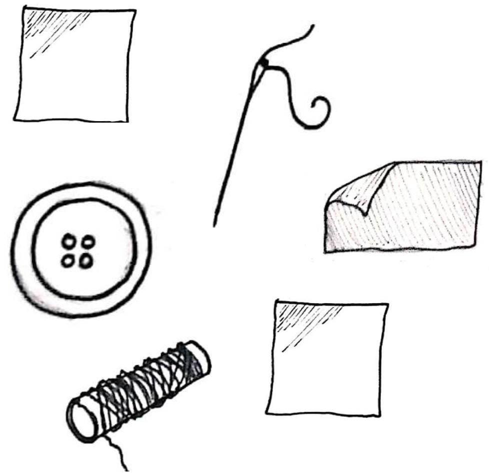
Button Button

The Button Button is another two-state (on and off) button made from textiles. It uses a plastic button, sewn in position with conductive thread, which than can be integrated into a circuit.

What you need:

CONDUCTIVE THREAD
CONDUCTIVE FABRIC
NORMAL FABRIC (X2)
NEEDLE AND THREAD

PLASTIC BUTTON
(MUST HAVE RAISED EDGES)



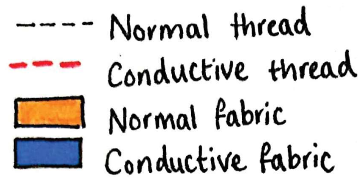
How it Works

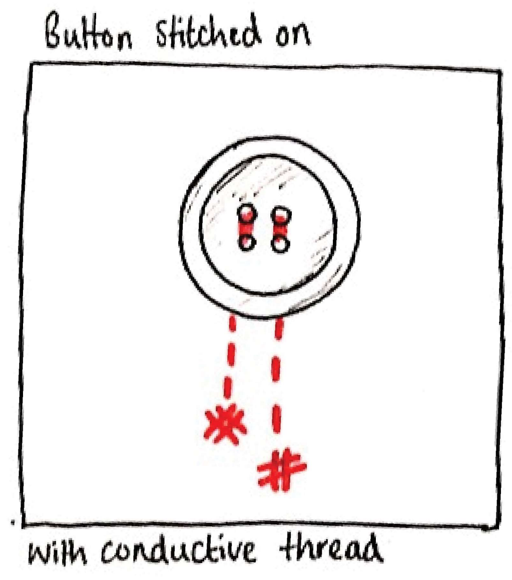
Test by connecting the button to the tester circuit. The LED should light up when you push down on the button.
This is because, when pushed, the conductive fabric forms a connection between the two bridges of conductive thread holding the button in place. This completes the circuit so the current can run through and the LED lights up.

Build Illustration

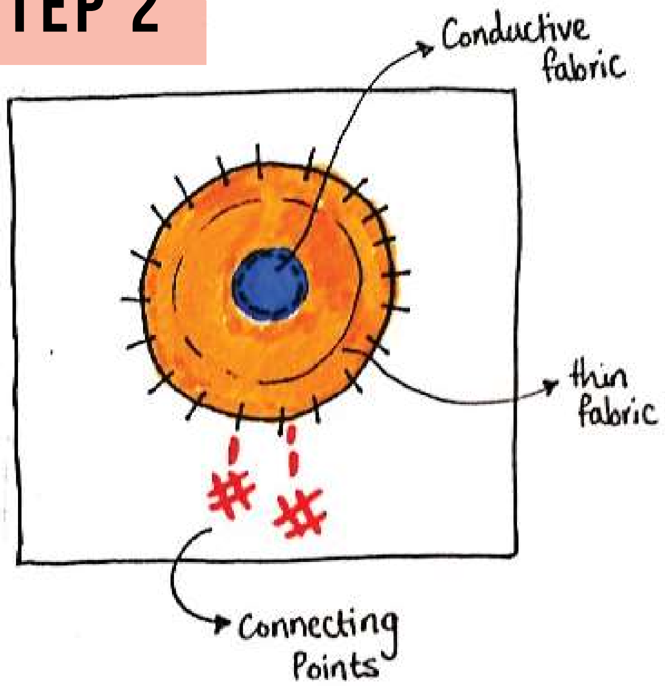
STEP 1

Sew the button into position, onto the base fabric, using conductive thread. Each side of the button should be sewn down seperately, with a seperate piece of thread. From each side, a connecting point can be made at the base of the fabric square.

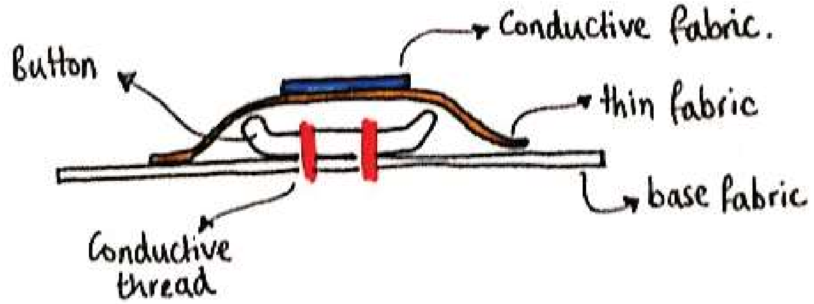




STEP 2



Cut out your second piece of fabric into a circle that will fit over the button, leaving enough excess to be able to sew around. Cut a small circle of conductive fabric to fit in the centre.
Sew the conductive fabric in the centre of the normal fabric. If the fabric is thin, it can sit on top, if not, it should be on the bottom side.
Sew this over the top of your button, making sure it is taught so it is not constantly touching the centre of the button.



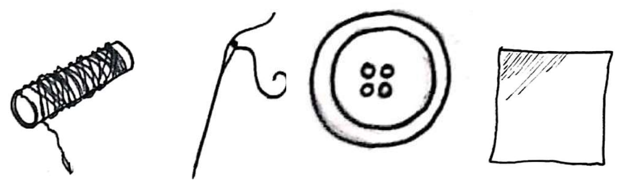
Button Switch

The Button Switch uses a button and hole to create an electrical switch. When open the switch is 'off' and when done up, the switch takes the 'on' state.

What you need:

CONDUCTIVE THREAD
NORMAL FABRIC (X2)
NEEDLE AND THREAD

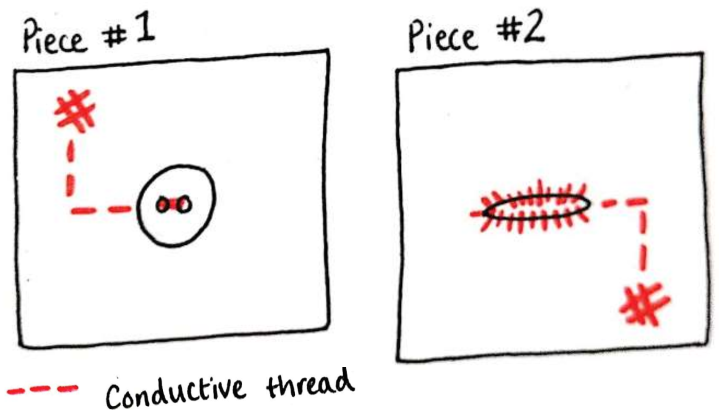
METALLICE BUTTON
(IT MUST BE CONDUCTIVE)



How it Works

Test by connecting the switch to the tester circuit. The LED should light up when the switch is 'closed'. This is because, when closed, the conductive metal back of the button comes in contact with the conductive thread that borders the button hole. This completes the circuit, allowing the current to run through and the LED to light up.

Build Illustration



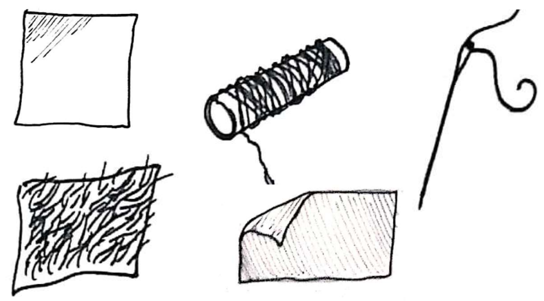
Sew the metallic button onto one piece of fabric using conductive thread. Still with this conductive thread form a connecting point, exiting the button from the top. For the second piece of fabric, cut a button hole the same size at the metallic button. Sew the connecting point exiting the hole from the bottom.

Stroke Sensor

The Stroke sensor is able to provide an output when a section of furry fabric is stroked.

What you need:

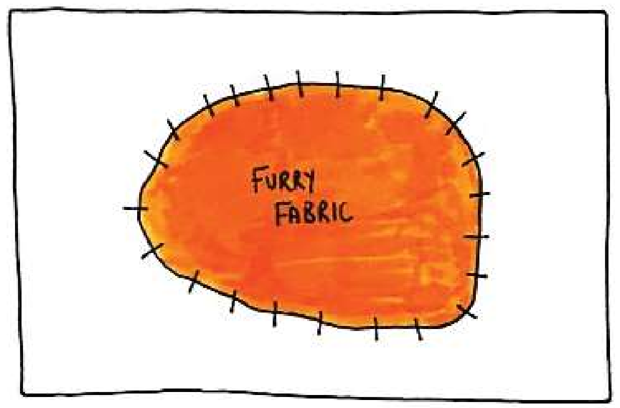
CONDUCTIVE FABRIC
CONDUCTIVE THREAD
NORMAL FABRIC (X2)
NEEDLE AND THREAD
FURRY FABRIC

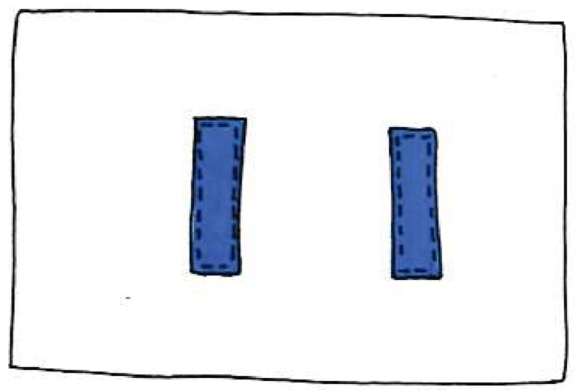


Build Illustration

STEP 1

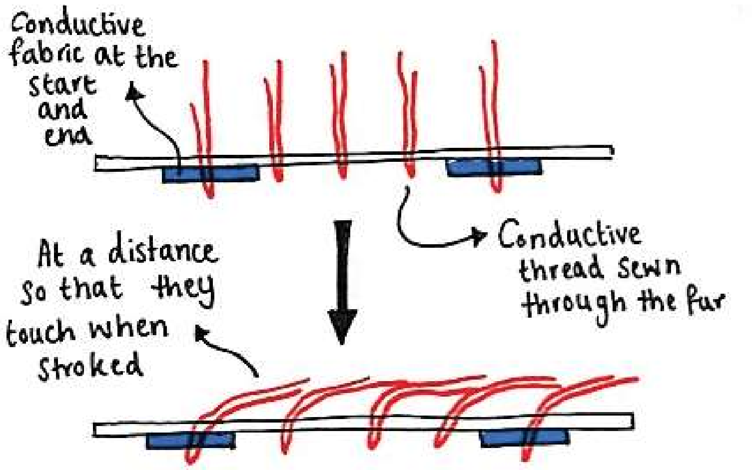
First, cut out a piece of furry fabric and sew into place on the base fabric.
Cut out two rectangular pieces of conductive fabric.
Turn the base fabric over, and sew on the two rectangular pieces.
Position the rectangular pieces so that they are at each end of the furry fabric.





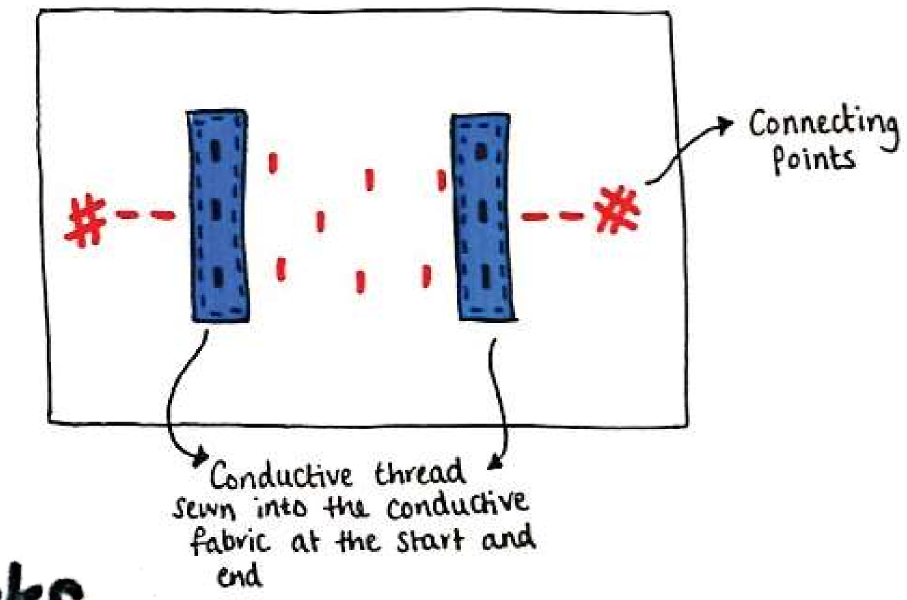
STEP 2

Using conductive thread, sew small tufts into the furry fabric. These should be the same height as the furry fabric so that they are not easily visible.
Make sure there are several tufts of thread at each end, sewn into the conductive fabric rectangles.
The aim is to sew enough tufts of thread into the fabric so that when the sensor is stroked, these tufts lie flat and touch one another, forming a bridge bewteen the two conductive rectangles.



STEP 3

Using conductive thread, sew two connecting points for the sensor. One coming from each of the rectangles made from conductive fabric.



How it Works

Test by connecting the sensor to the tester circuit. The LED should light up when you stroke along the furry fabric. It may take some tries to get enough tufts of thread in place. This works because, when stroked, the conductive thread weaved into the furry fabric flattens and comes in contact with each other. This forms a bridge between the 2 pieces of conductive fabric sewn at each end of the sensor, completing the curcuit and allowing current to run through to the LED.

Flip dot eye

This is an eye which flips when attached to a 9V battery. Thanks to creating an electromagnet, when electricity flows through the circuit, the eye either flips open or closed!

What you need:

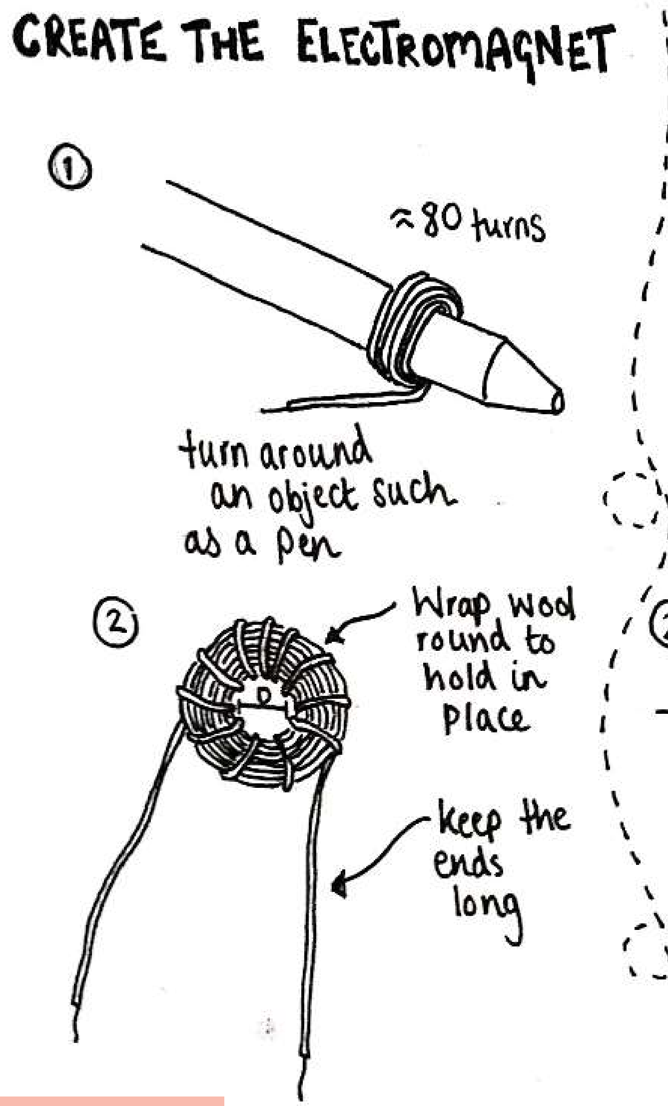
NORMAL FABRIC
NEEDLE AND THREAD
THICK THREAD OR WOOL
INSULATED WIRE
POLYMORPH
STRONG CIRCULAR TINY MAGNET (5MM DIAMETER)

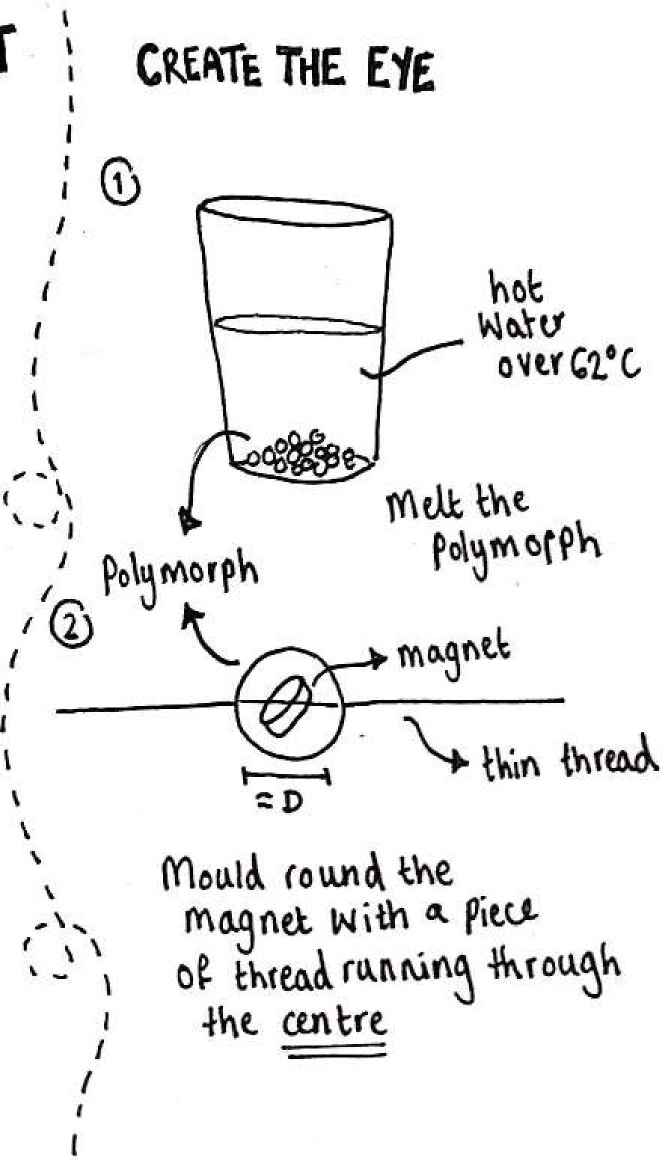
Build Illustration

STEP 1

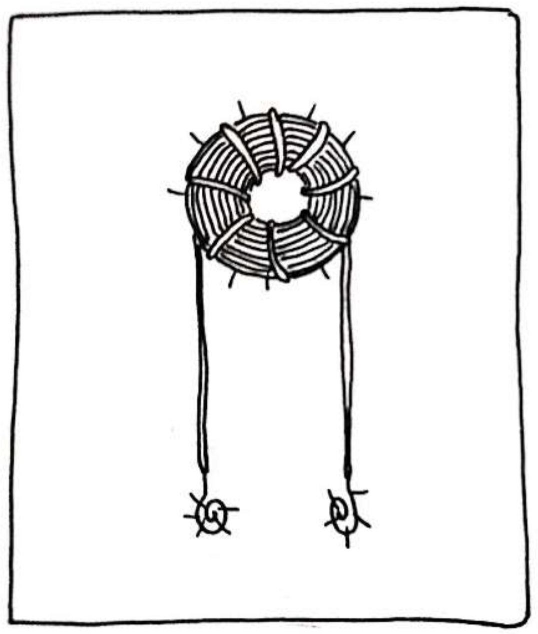
To create the electromagnet, you need a coil out of the insulated wire. The central hole of the coil needs to be around the same size as the eye you want. Here, we have used a pen. Wrap the wire around about 80 times, then hold in place by wrapping it in wool or thick thread. Make sure to leave excess wire at the beginning and end for connections.

To create the eye, melt polymorph in a small cup of water. It must be above 62 degrees celcius. Once melted, form the polymorph around the magnet into a sphere.
Add a thin peice of thread cross the centre.





STEP 2



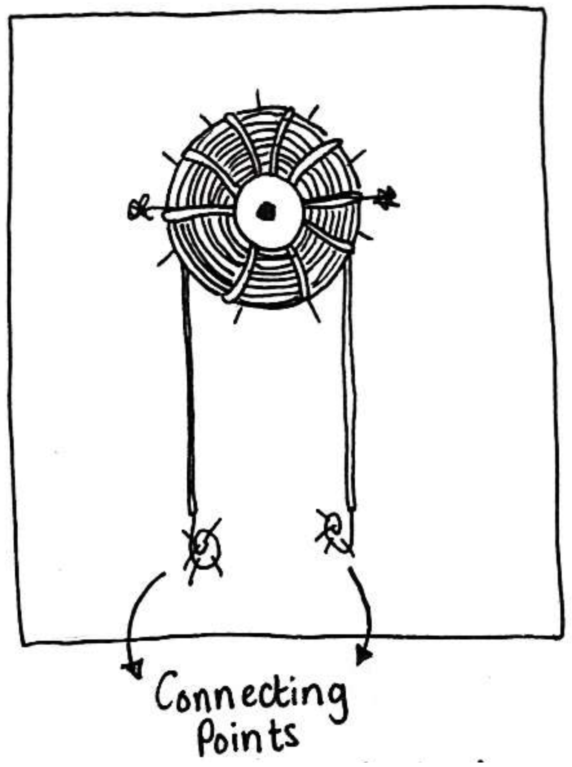
Once you have created the electromagnet, strip the two ends of the insulated wire and coil up the ends.

Now atttach the electromagnet coil to the base piece of fabric. Sew it into place, making sure that it is secure.

Then, attach each of the legs of the electromagnet to the base fabric, sewing each coil of exposed wire.

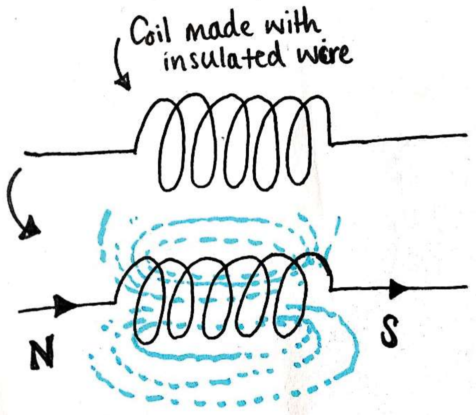
STEP 3

Attach the eye into position by placing it in the centre of the electromagnet. Sew it into place by sewing the ends of the thin thread into the base fabric. Make sure that the thread is tight enough that the eye will remain in place, but is slightly slack to allow the eye to rotate.

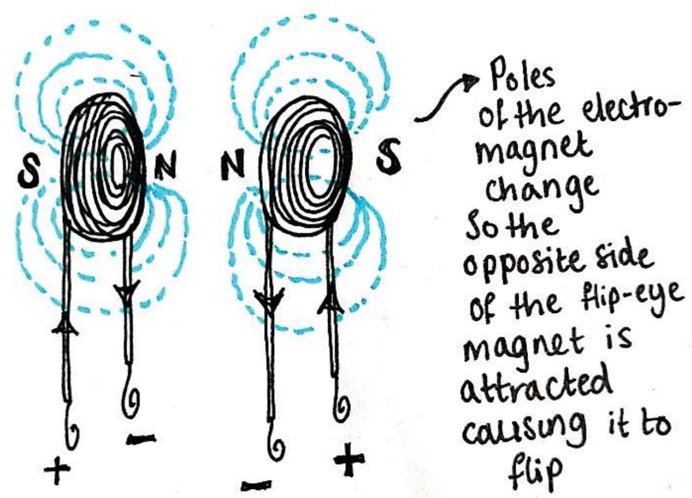


Connect one end to the +ve of the 9V battery and the other to -ve. Then swap to make the eye flip!

How it Works



An electromagnet is created when current runs through a coiled wire. This forms an electromagnetic field around the coil as shown above, with north and south poles. The more turns the coil has, the stronger the electromagnetic field and the electromagnet is. This is why around 80 turns of wire are needed.



When you swap the connection of the battery, the current flows in the opposite direction and the electromagnetic field flips. This causes the polarisation of the electromagnet (the sides of the magnet which are north and south poles) to switch.
When the polarisation of the electro-magnet swaps, the opposite side of the magnet (in the eye) is attracted, causing it to flip.

Flappin' Wing

Make a wing that can flap up and down when an electrical current is applied. The flapping wing uses electromagnetism to create movement in a similar way to the flip dot eye.

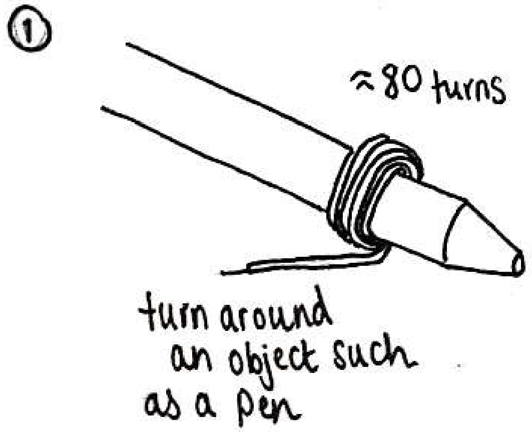
What you need:

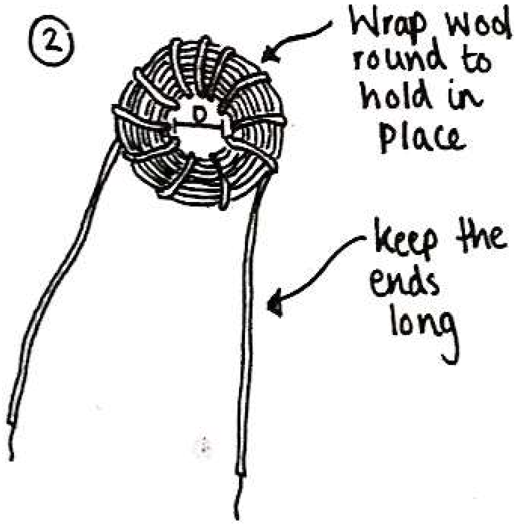
NORMAL FABRIC
NEEDLE AND THREAD
THICK THREAD OR WOOL
INSULATED WIRE
WADDING OR LIGHT FABRIC
STRONG SMALL CIRCULAR MAGNET (1CM DIAMETER)

Build Illustration

STEP 1

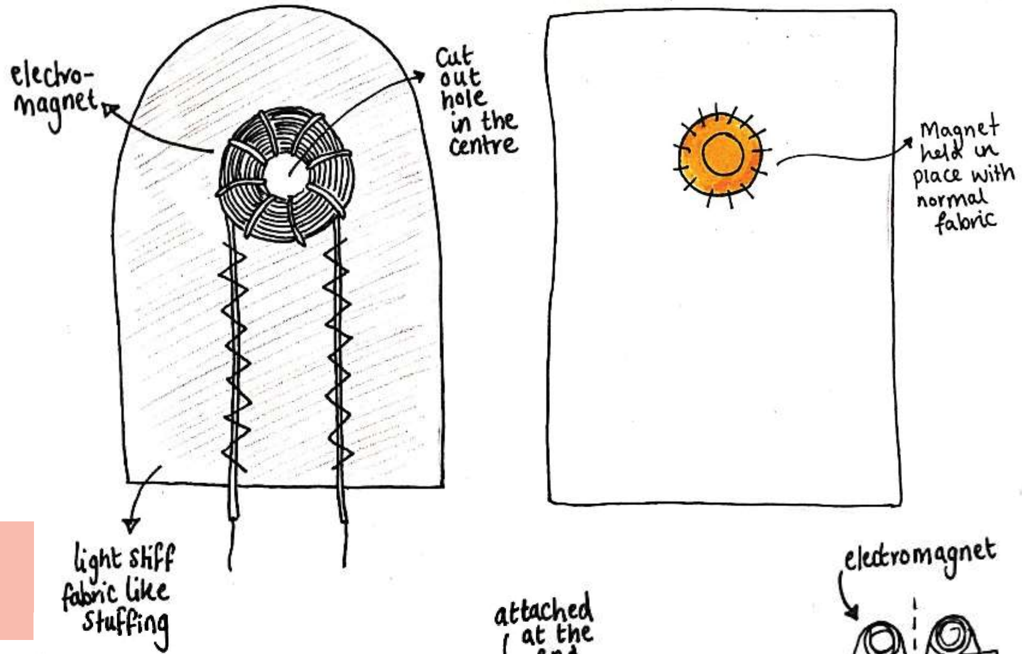
CREATE THE ELECTROMAGNET



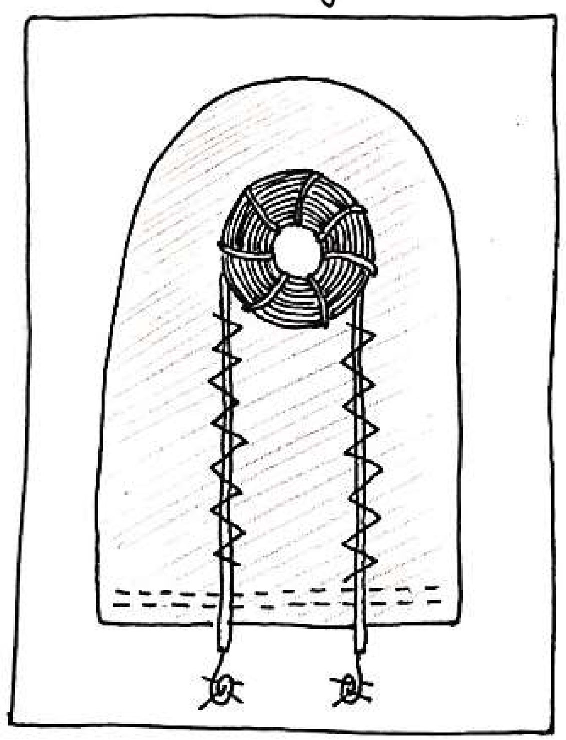


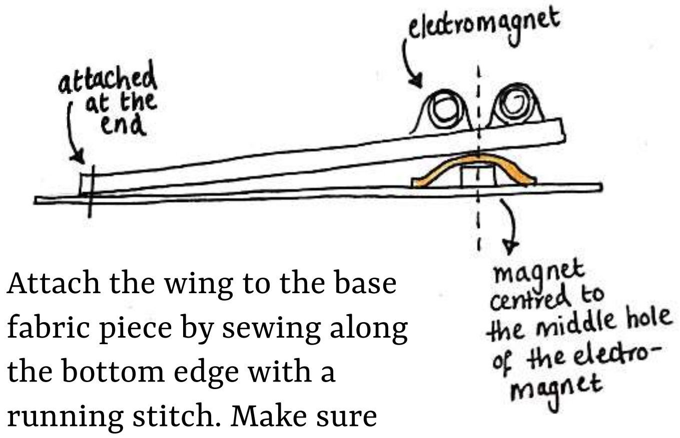
STEP 2

Attach the electromagnet into position on the wadding or light, stiff fabric that has been cut out into the shape for the wing. It's suggested to sew the the legs down using a zig zag or couching stitch. On the base fabric, sew the magnet into positon by covering it with a small circle of fabric and sewing around the edges. You need the magnet to line up with where the electromagnet on the wing will sit.



STEP 3





Attach the wing to the base fabric piece by sewing along the bottom edge with a running stitch. Make sure the electromganet is well positioned over the magnet like shown in the image above.

Strip the ends of the electromagnet, coil these up and sew into place on the base fabric.

Connect one end to the +ve of the 9V battery and the other to -ve. Then swap to make the wing flap!

How it Works

Check out the How it Works section for the Flip dot eye. The flappin' wing works on the same principles!

MINI PROJECTS

WOVEN SPEAKER
PLUSH MONSTER
LIGHT UP BRACELET

Right: Etiam egestas, dui vel facilisis consequat, massa nibh ultrices nisl, sed sollicitudin diam odio non ante

FOR THE FACILITATORS

These projects focus on participants creating end items which require multidisciplinary skills. They provide additional context to E-textiles by presenting how an object can be made which has a function through integrating electronics with textiles or textile methods.

THE WOVEN SPEAKER

This project helps provide a hands-on explanation of how speakers function using electromagnetism. It is a good introduction to soldering.

Difficulty: With the addition of soldering, the ideal age for this activity is 13.

Cost: Each speaker costs around 7 euros

Time: Depending on the age of the participants, this activity could take between half a day and a full day.

PLUSH MONSTER

This project focuses on textiles skills and integrating a simple circuit. It is a more creative and design based project.

Difficulty: This activity is not too complex, if the circuit remains simple. It is suitable for participants over the age of 11.

Cost: This is a cheap activity with the most expensive material being the batteries and conductive thread.

Time: Depending on the age of the participants, this activity could take between half a day and a full day. Some preparation could be made prior to the activity to make it shorter. However with older participants more complex designs and circuits could be created, extending the project to multiple days to include planning and testing.

LED FABRIC BRACELET

This project focuses on that can be worn. This project is quite complex due to the nature of wearables having to be small and compact. It uses soldering techniques as well as requiring a basic knowledge of circuitry.

Difficulty: This activity is of higher difficulty and will require a high level of patience and perseverance. It is good for participants to have some soldering experience and have basic knowledge of circuitry.

Cost: This is a cheap activity with the most expensive component being the batteries and conductive thread. Each bracelet can cost as little as 1,50 euro if multiple are made. It is beneficial to have a sewing machine but not compulsory.

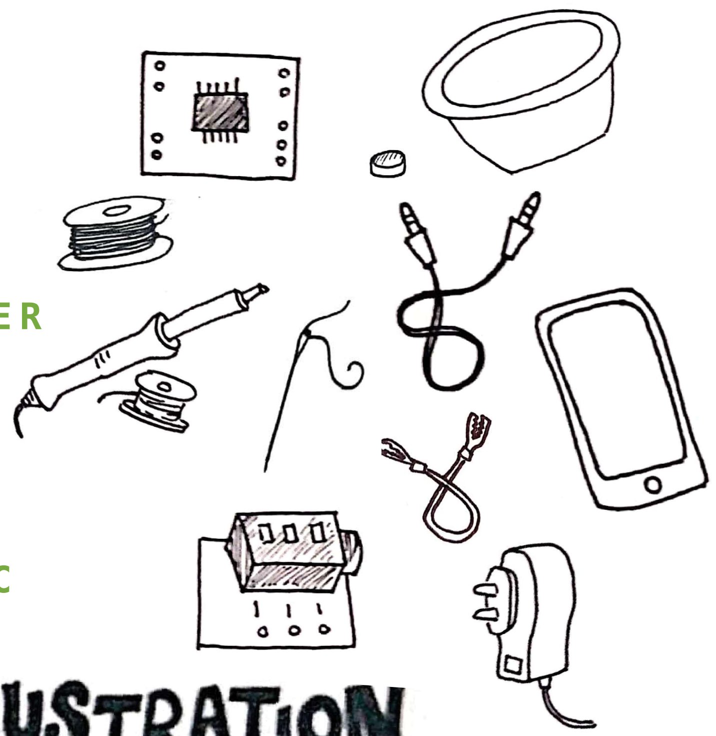
Time: It would be expected for this project to take a full day.

WOVEN SPEAKER

The Woven Speaker works on the basis of electromagnetism. Using a simple weaving technique, you can make your own speaker which can transmit music from your phone!

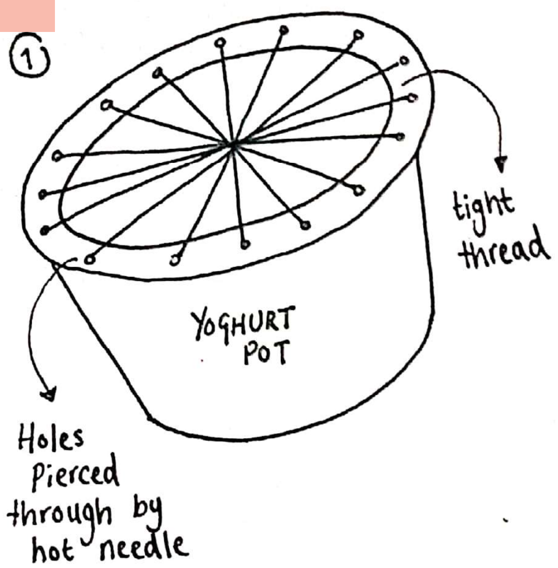
What you need:

YOGHURT POT
INSULATED WIRE
MAGNET
SOLDERING IRON AND SOLDER
NEEDLE AND THREAD
CROCODILE CLIPS
AUX CABLE
AUDIO JACK PORT
5V POWER SUPPLY
DEVICE FOR PLAYING MUSIC



BUILD ILLUSTRATION

STEP 1



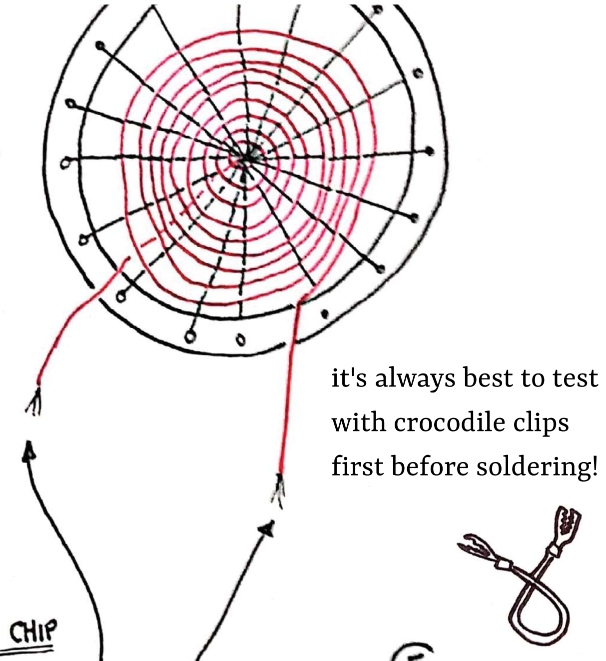
Prepare your yoghurt pot by piercing holes around the lid. You can do this by heating up a needle, using a small candle, and piercing through the plastic as it melts.

Create a web for weaving, by using thread and crossing it from one side to the other. tie it in position. Make sure that is is tight!

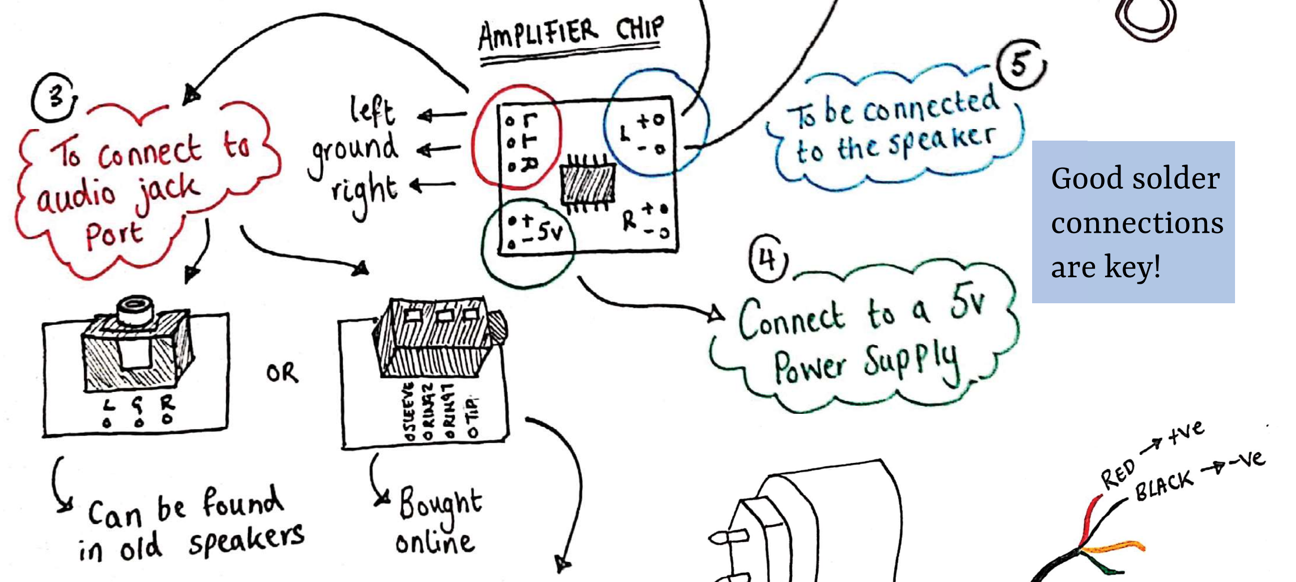
STEP 2

Weave the insulated wire around the web. You need to include as many turns as possible, making sure it's tight. Alternate going under and over the thread to weave into position. Make sure to leave excess at each end.

For the electronics, make the different connections for the Amplfier chip as shown in the diagram below. Use thin insulated wire.

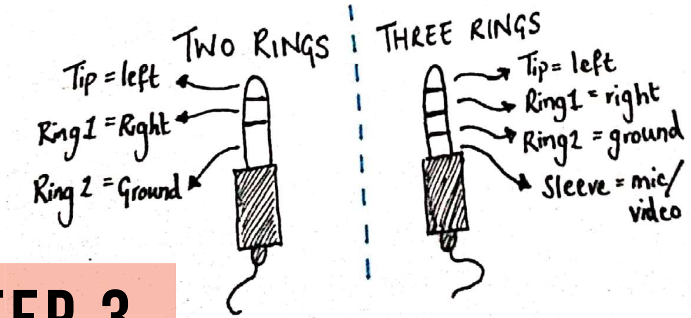


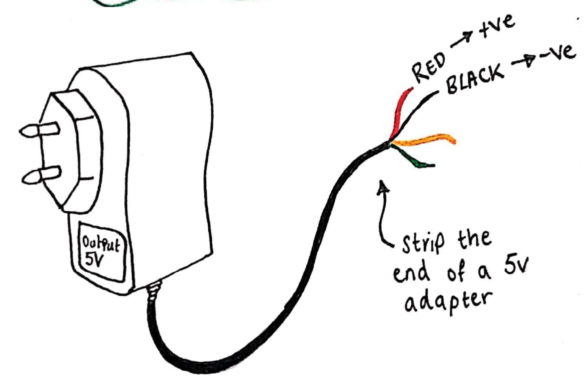
it's always best to test with crocodile clips first before soldering!



Good solder connections are key!

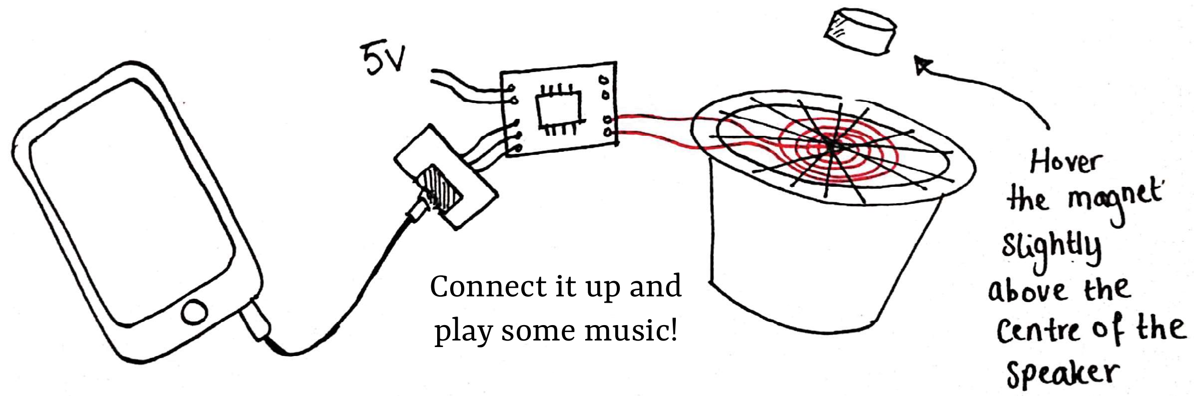
Know Your Aux





Make sure to use a 5V supply only. Any bigger and the Amplifier chip will be damaged.
Strip the end of the plug and make the correct connections by soldering in place.

STEP 3

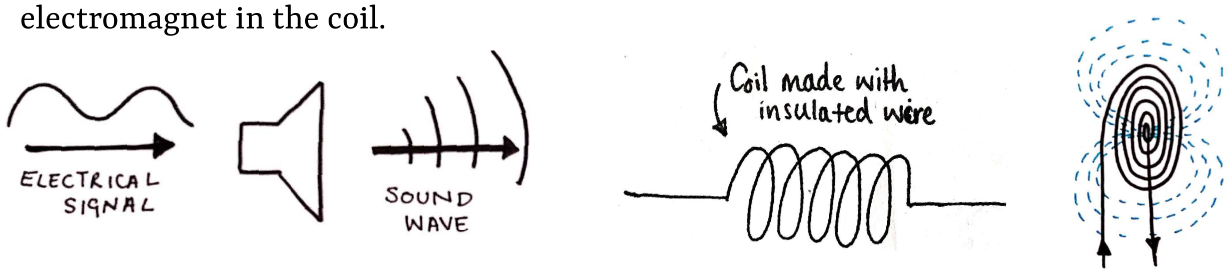


Connect it up and play some music!

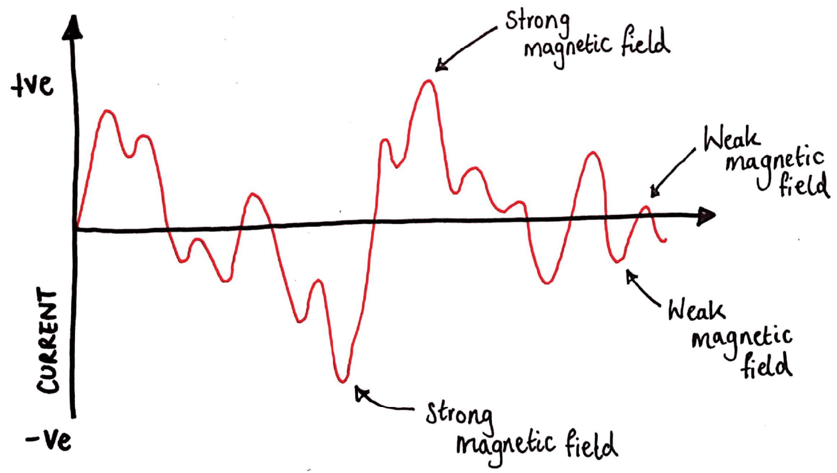
How it Works

Connect up your phone and play some music.
Hover the magnet over the top of the speaker. Listen very carefully and you should hear your music play!

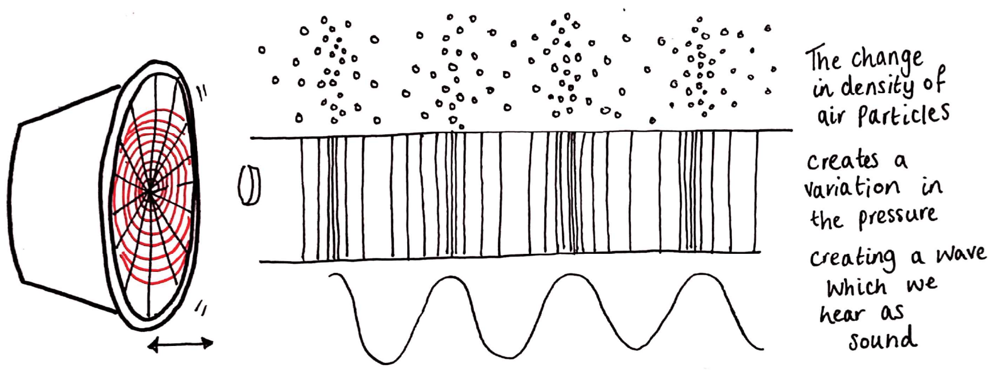
The woven speaker is an electromechanical transducer as it converts an electrical signal (from the phone) to sound waves.
The phone supplies an electric signal with a varying current, which runs through the coil of wire (which was created when weaving the spiral). This creates an electromagnet in the coil.



This electromagnet changes polarisation and strength in accordance with the electric current. As the signal provided by the phone varies, with a varying current, the properties of the electromagnet field also change.



This electromagnet is attracted to and repelled by the magnet which is held above the coil, causing the woven spiral to move up and down. This movement creates a pressure wave which allows us to hear sound.

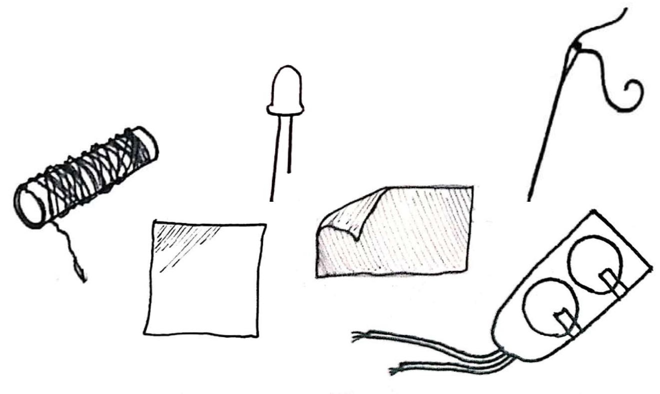


PLUSH MONSTER

Create a monster, alien or animal which can light up when you give it a squeeze! It can be made by hand sewing and using basic electronics.

What you need:

CONDUCTIVE FABRIC
CONDUCTIVE THREAD
NORMAL FABRIC (OR FELT)
NEEDLE AND THREAD
LED
COIN BATTERIES
COIN BATTERY HOLDER

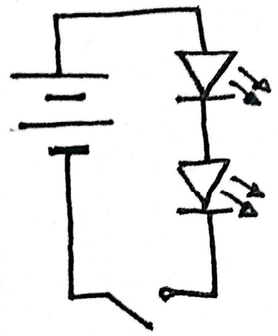


How it Works

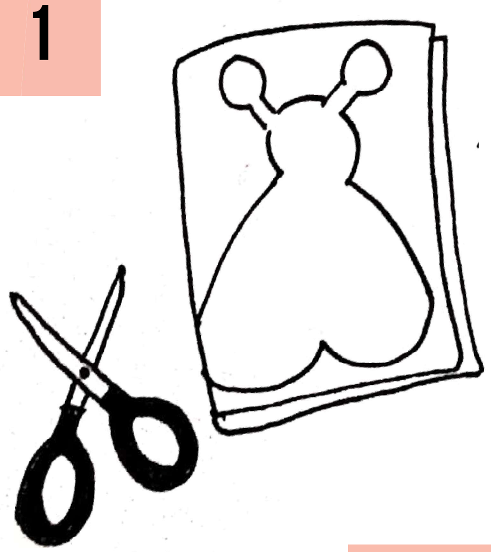
The patches of conductive fabric act as a switch. When stuffing is between them, they do not touch and the ciricuit is broken. When they are pushed together and they touch, current can run through and the LEDs can light up!

As the circuit diagram shows, the LEDs are connected in SERIES. This means that if one LED breaks or if it is connected the wrong way round, the current is unable to run through. This results in a break in the circuit, so that the 2nd LED will not be able to light up.

Circuit Diagram



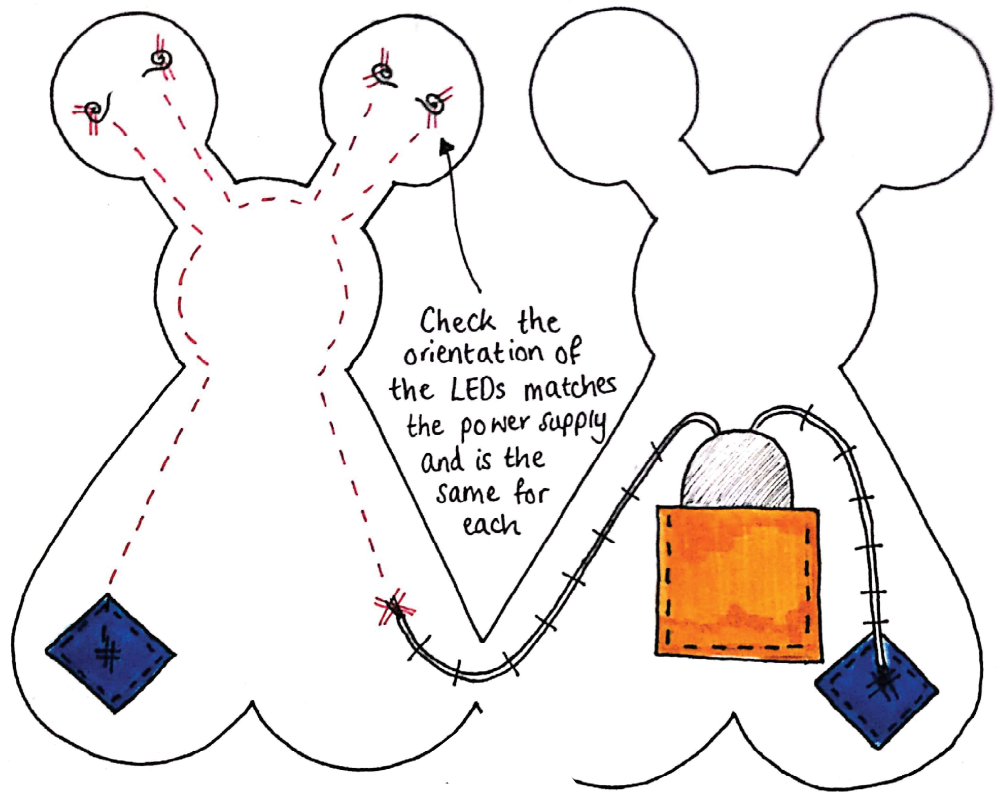
STEP 1



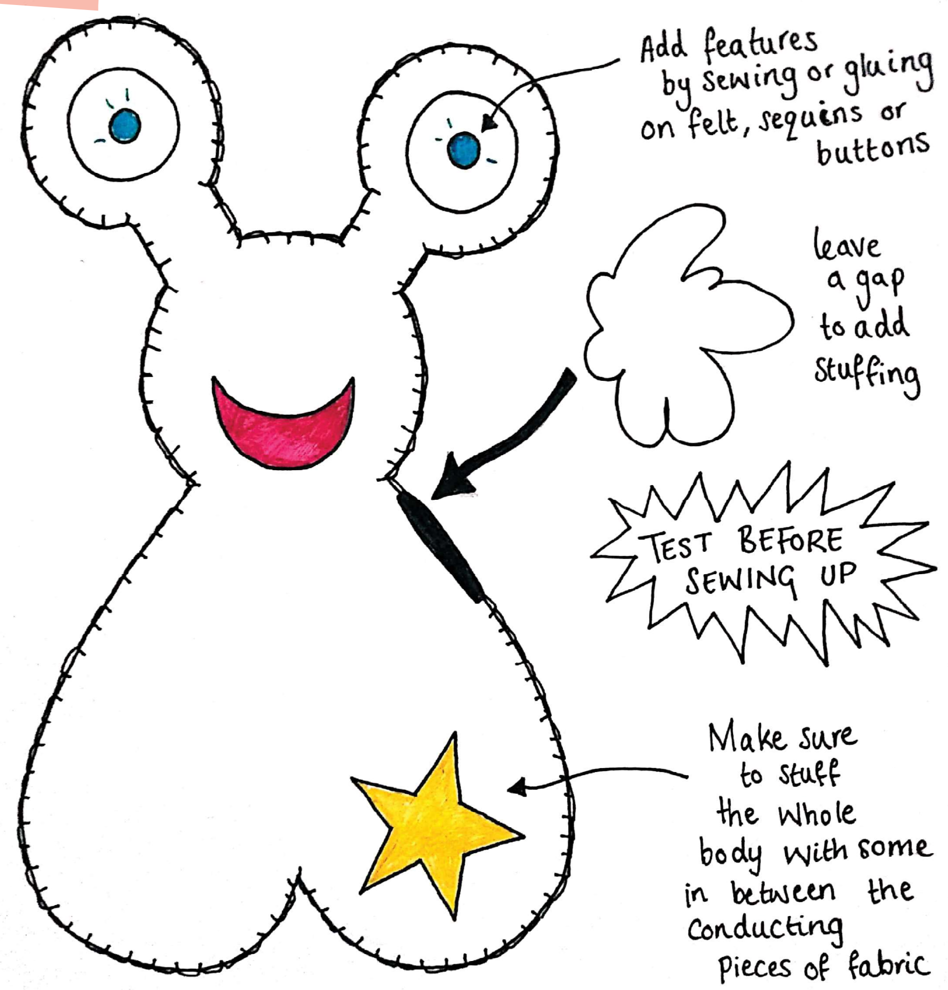
The first step is to cut out the shape of the creature you would like to make. You need a front and back piece that are connected. The easiest way is to fold a piece of fabric in two and cut the shape out, cutting through both pieces at the same time. If you leave a small section where there is the fold in the fabric, the two pieces will remain connected like shown below.

STEP 2

The next step is to create the circuit on the inside of your creature. Sew two squares of conductive fabric to the base corners with normal thread. When folded together, these two squares should line up with each other and touch. Sew a pouch for your battery holder to sit in. Connect the negative connection to the conductive fabric square and the positive connection to the positive leg of the first LED. The diagram below then shows two LEDs connected in series. Make sure that both LEDs are the same way around so the negative leg of the first LED connects with the positive leg of the second. Complete the circuit by sewing from the final negative leg of the 2nd LED to the second square of conductive fabric. These connections should be done in conductive thread, shown in red in the diagram

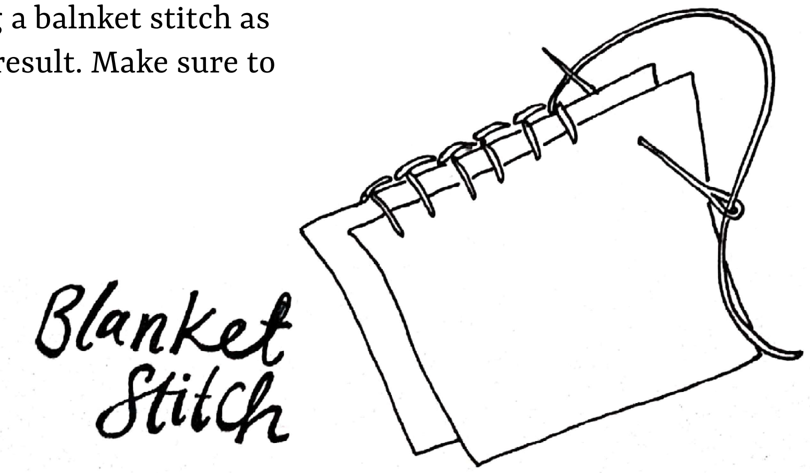


STEP 3



Test that your circuit works by touching the two conductive squares together - the LEDs should light up!
Sew on features to the front of your creature, such as eyes and a shape to indicate where it should be squeezed.
Knowing that it works, you can fold your creature and sew it together. We suggest using a balnket stitch as it is a fast way to give a tidy end result. Make sure to leave a small hole in the side.

Add stuffing into your creature through the hole. Make sure it is enough to keep the two conductve fabric squares seperate, but still allows electricity to conduct through when they are pushed together.

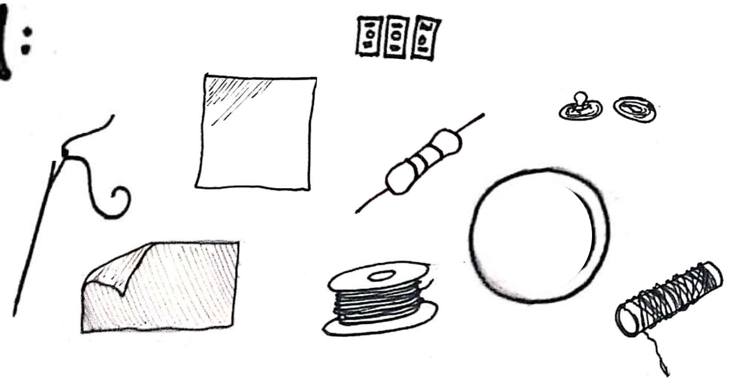


LED FABRIC BRACELET

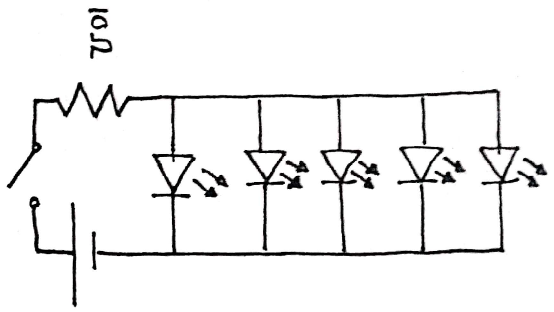
Create a bracelet that lights up when you put it on! A challenging project which integrates different elctronic and sewing techniques.

What you need:

CONDUCTIVE FABRIC
CONDUCTIVE THREAD
NORMAL FABRIC
NEEDLE AND THREAD
THIN COPPER WIRE
MINI LEDS
COIN BATTERY
METAL POPPERS
SEWING MACHINE
RESISTOR (10 OHM)



Circuit Diagram



How it Works

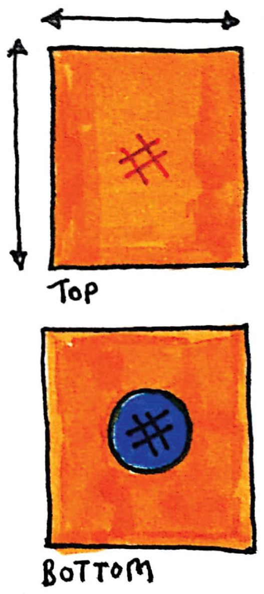
Test by touching the 2 parts of the popper together.
Here, the popper is acting as a switch. When undone, the circuit is not complete. When the popper is done up, the current is able to flow through as the popper is made from a conductive metal. This completes the circuit.

The LEDs are connected in PARALLEL as shown in the Circuit diagram. This means that the LEDs will continue to light up even if one or more of the other LEDs don't work. This is because the flow of electricity doesn't have to go through an LED to get to the next one.

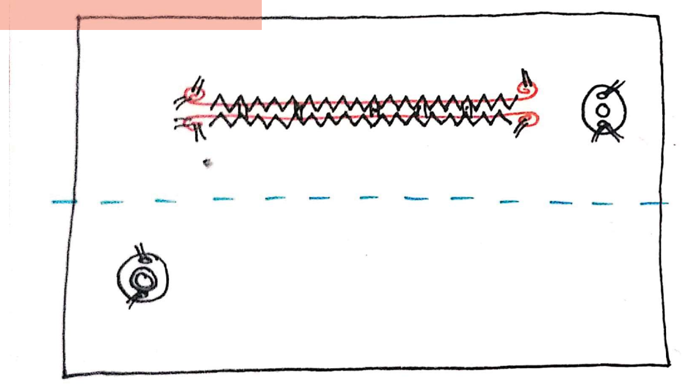
STEP 1

The first step is to cut out the correct pieces of fabric.
You will need one piece of fabric that will be the bracelet. This should be the distance to fit around you wrist + about 3 cm, and a height of 7cm.
(Detailed diagram is shown on the next page).

You will then need to cut out a square that is 2cm x 2cm. Cut out a small circle of conductive fabric and glue it to the centre of the back of this square.

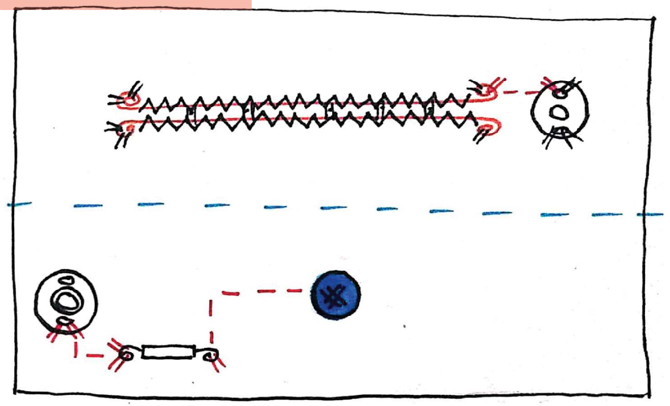


STEP 2



The second step is to sew down the first components. Take the fabric that will be your braclet and on the top half, sew down one half of the popper. On the bottom half, sew down the other half of the popper, so that they match up when the bacelet will be closed.
Sew down two pieces of thin copper wire. They should line up parallel to each other, with the distance of a mini LED between them. It is best to use a zig-zag stitch to hold them in position. This is best done using a sewing machine.

STEP 3



Solder the mini LEDs to the thin copper wire. Paying attention that each LED is the same way round. (Positvie connections are usually indicated by a green square on the back of the LED.)

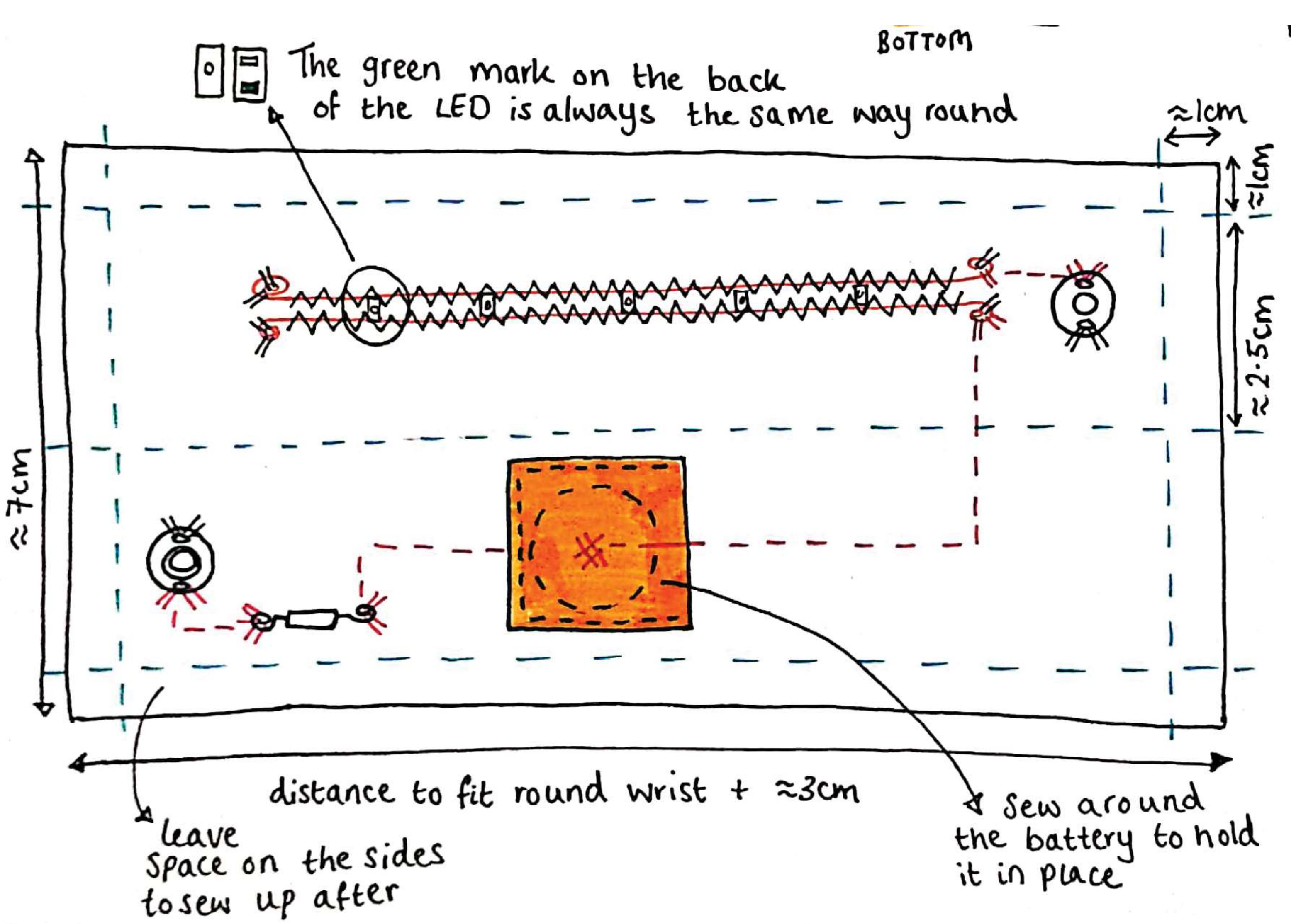
Next, on the bottom half of the fabric sew into position a resister and small circle of conductive fabric. This will be where the battery sits.
Join these 3 components with 2 seperate pieces of conductive thread, as shown.

STEP 4

Complete the circuit by taking the square piece of fabric cut in Step 1. Sew in a connection point using conductive thread. Keep this thread on the needle and sew the square down, sewing along 3 sides.

Insert the battery and test that it is the correct way round by closing the circuit using the needle still threaded with conductive thread. and touching it to the bottom piece of thin copper wire.
The LEDs should light up.
Once you know the battery is the correct way round, sew around the battery to hold it securely and tightly in place.

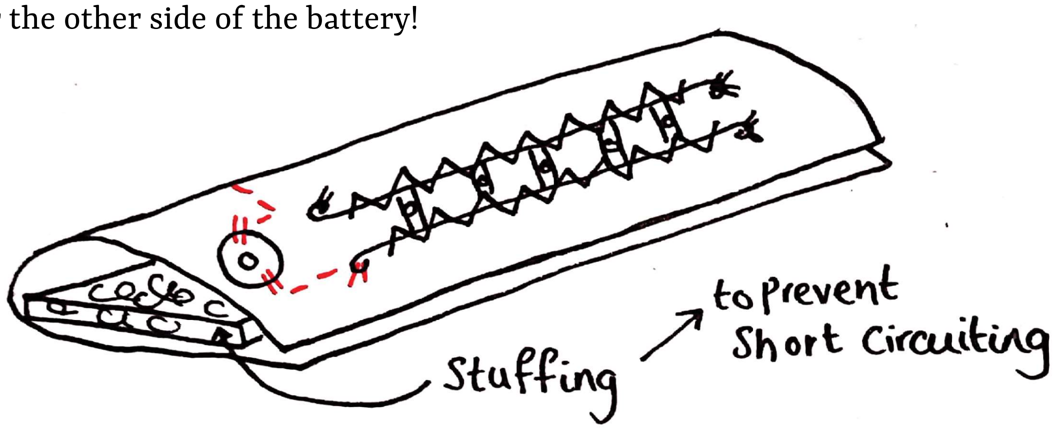
With the threaded needle with conductive thread, continue to form the connection between the battery and the thin copper wire as shown in the diagram.

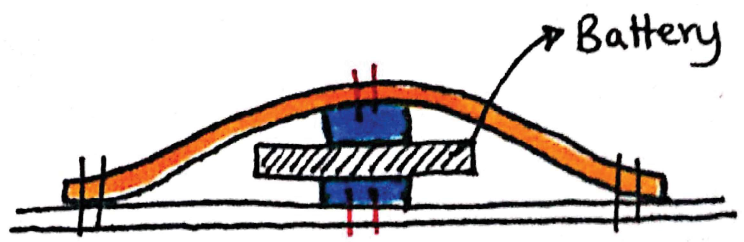


STEP 5

Check that your bracelet works by touching the two poppers together. The LEDs should light up. If not, there is probably something wrong with the battery pouch and it's connections. Make sure none of the conductive fabric or thread, that touches one side of the battery is touching one another or the other side of the battery!

Once working, cut a small thin piece of stuffing, that is the size of your bracelet.
place this on the inside and fold the top of your bracelet over.
This will help prevent short circuiting.

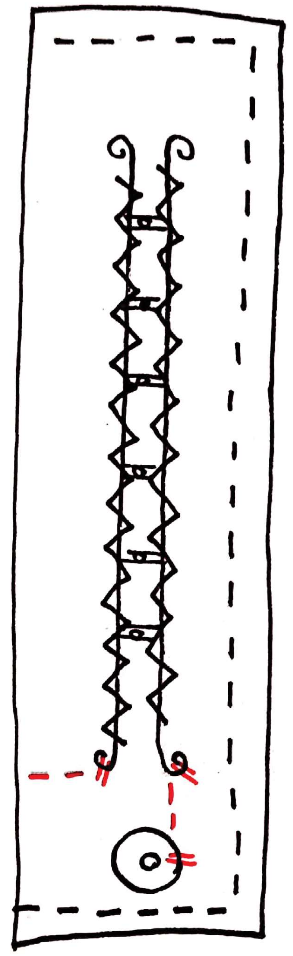




This diagram shows the side view of the battery pouch. From the bottom up, you should have a small circle of conductive fabric on the base fabric - this is connected to the resistor.
You should then have the battery.
Followed by the top piece of fabric which has a second piece of conductive fabric underneath - which is connected to the popper.

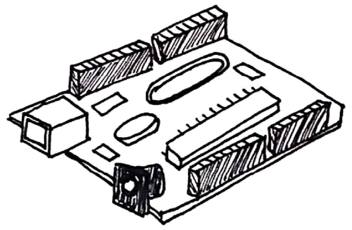
STEP 6

The final step is to sew up your bracelet. You can use a quick running stitch to hold the bracelet in place. Then go around the edge with a blanket stitch to give a clean and tidy finish.
Your bracelet should now be ready to wear!



INTRODUCTION TO ARDUINO

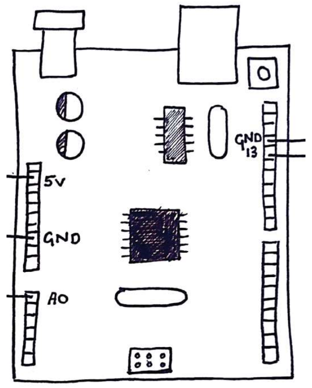
WHAT IS AN ARDUINO?



Arduino is an open-source hardware and software. The hardware is a board with a microcontroller, as shown in the illustration. It can take an input (such as a signal from a sensor) and create an output (such as make a motor turn or an LED flash). The board works with Arduino IDE software, which allows you to create the code that controls your board, the inputs it reads and the outputs it creates.

HOW TO USE AN ARDUINO

When looking at an Arduino board, there are many different ports. The illustration shows a top down image of the board. The ports are where components can be connected such as sensors or actuators, usually using jumper cables and a breadboard.
Each port has a label next to it. On the left there are ports related to power, including ground (GND) and different voltages (5V, 3.3V, VIN). Under these there are the analog ports (all starting with A; A0, A1..) These are used for components which give an analog signal such as a potentiometer. On the right, there are numbered ports which can be used for other components such as an LED. In addition, the numbered ports with a (~) mean that they can be used with Pulse Width Modulation components, such as a servo motor.
It is with these numbers that you can define which port your component is connected to in your code.



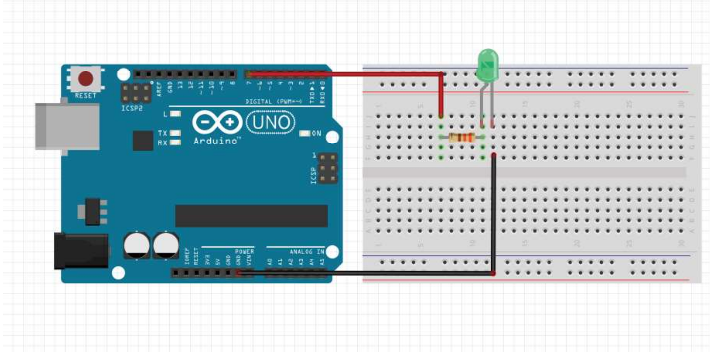
SETUP ARDUINO ON YOUR COMPUTER

To be able to use the Arduino Board, download the free Arduino IDE software from the Arduino website. This provides a programme where you can write the code to be uploaded onto the baord. When using the Arduino IDE, make sure you have all the necessary libaries downloaded for the code you are trying to run, and that it is connected to the correct COM port to which your Arduino is connected. For more detailed information on this, check out this instrucables:

WWW.INSTRUCTABLES.COM/INTRO-TO-ARDUINO

ARDUINO ACTIVITIES FOR BEGINNERS

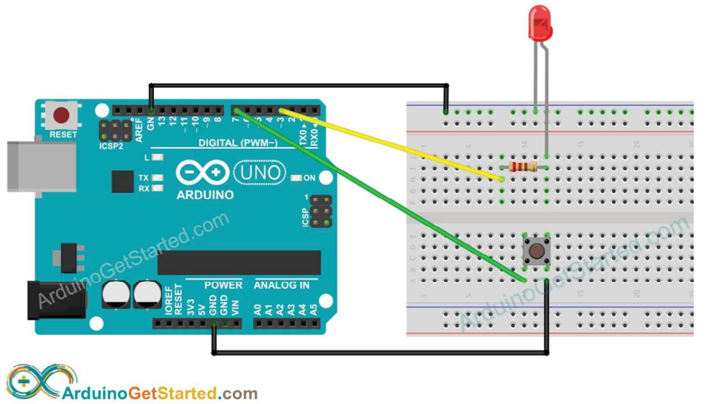
Here are a few great starting activities for Arduino Beginners. With Arduino, it's best to start very simple and to really understand whats going on. This way, it is easier to build up to more complicated circuits. These are three activities we always start with to give a good introduction to how to make a circuit and code work. Make sure to look through the code before running it, to be able to understand whats happening. Check out the Coding Dictionary on page 72 for additional information.



FLASHING LED

The best place to start: run an example code already provided by Arduino, which can make an LED flash.

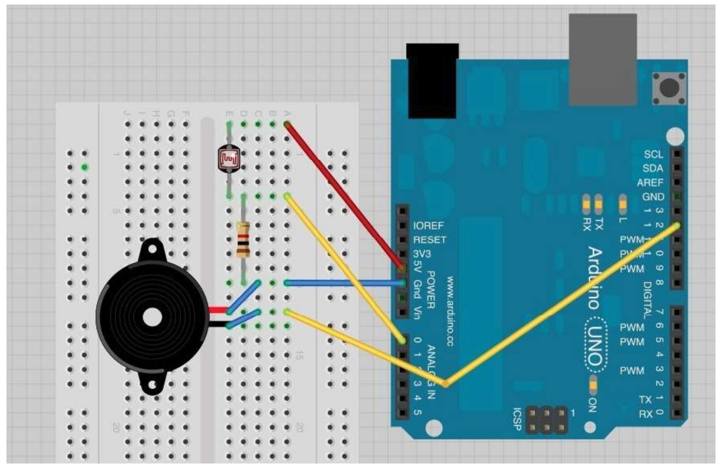
WWW.INSTRUCTABLES.COM/ARDUINO-BLINKING-LED/



BUTTON CONTROLLED LED

Now include a new component - a button. Write the code so that the LED lights up when the button is pressed.

HTTPS://ARDUINOGETSTARTED.COM/TUTORIALS/ARDUINO-



THEREMIN

Now change your input and outputs. Include a photosensor for the input and a buzzer to create sound as the output. The code causes the sound to vary with different amounts of light.

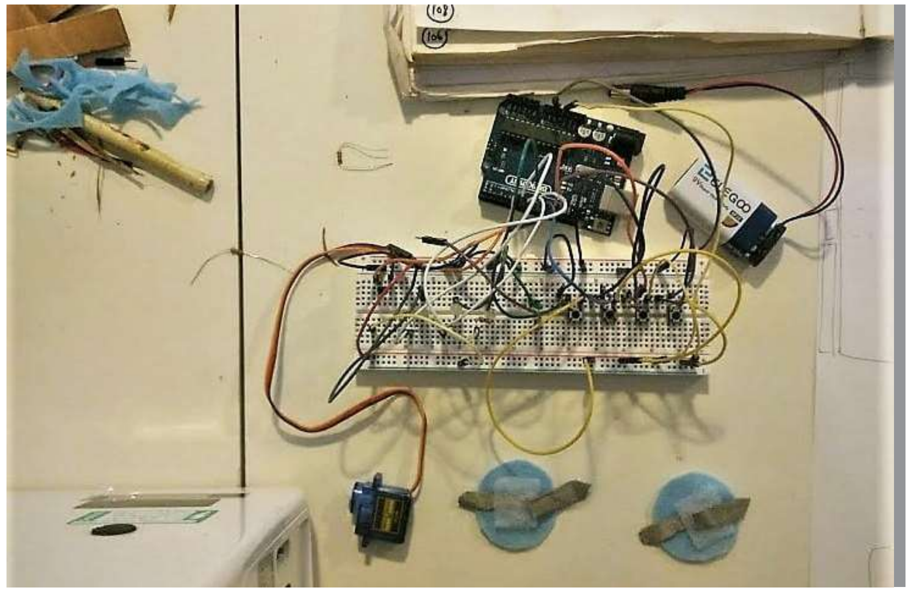
LEARN.ADAFRUIT.COM/ADAFRUIT-ARDUINO-LESSON-10-MAKING-SOUNDS/PSEUDO-THERAMIN

PROJECT CHALLENGES

LED FABRIC BRACELET WITH A PULSE SENSOR

SECRET CODED DIARY

FOR THE FACILITATORS



These projects are aimed to challenge participants. They bring together advanced skills and should be tackled with some previous knowledge of electronics and programming.
Both challenges introduce Arduino and programming to the participant, allowing E-Textile projects to become 'Smart textile' projects where there is a greater interaction between the item and the user.

Before starting these challenges, it is advised for both the facilitator and participants to have a basic understanding of electronic prototyping and using Arduino IDE. For these, follow the beginner activities provided on pages 52 and 53.

LED FABRIC BRACELET WITH A PULSE SENSOR

This project requires a LED fabric bracelet to have been created. Instructions can be found in the previous section.

The project aims to show how an E-textile item (in this case the bracelet), can be turned into a 'smart textile) with the introduction of a sensor.
Here a small pulse sensor is integrated into the circuit of the bracelet so that the lights flash in accordance to the wearer's heartbeat.

Difficulty: Along with the creation of the bracelet, this project would be suitable for participants who are 15+.

Cost: Arduino Uno board and accessories cost around 5 euros when purchased in bulk. The pulse sensor also costs 4 to 5 each. Therefore it is suggested that this project is completed in pairs or groups to reduce overall costs.

Time: This project could easily fill a day. With the creation of the bracelet and having time to explore and test how the pulse sensor works, this project could be part of a multiple day workshop.

SECRET CODED DIARY

This project provides an example of how to take a working circuit and integrate it into textiles. The project provides the opportunity to learn about electronic prototyping, Arduino programming, soldering as well as basic sewing techniques.

Difficulty: This project is complex and requires a lot of concentration from participants. It is best tackled in small groups with participants being 16+. The broken down steps help to aid participants in seeing the project as more manageable.

Cost: The Arduino and it's accessories are the most expensive component of this project, costing around 5 euros each when purchased in bulk.

Time: This project has been broken down into different steps, therefore can be spread over multiple sessions. It would be advised to expect that each step would take half a day to complete. Overall, it should be anticipated that such a project could be completed in around 3 days.

LED FABRIC BRACELET

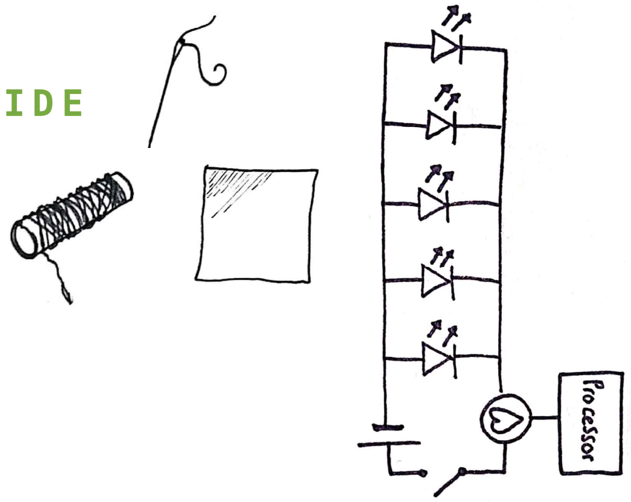
WITH A PULSE SENSOR

This tutotrial enhances the LED fabric bracelet from the previous tutorial, into a bracelet that responds to your heart rate! Using a pulse sensor and Arduino, you can create a bracelet that flashes at the same rate as your pulse!

What you need:

LED FABRIC BRACELET
ARDUINO UNO
COMPUTER WITH ARDUINO IDE
LED
CONDUCTIVE THREAD
NEEDLE AND THREAD
NORMAL FABRIC
CROCODILE CLIPS
PULSE SENSOR
VELCRO
CLEAR VINYL STICKER
GLUE GUN

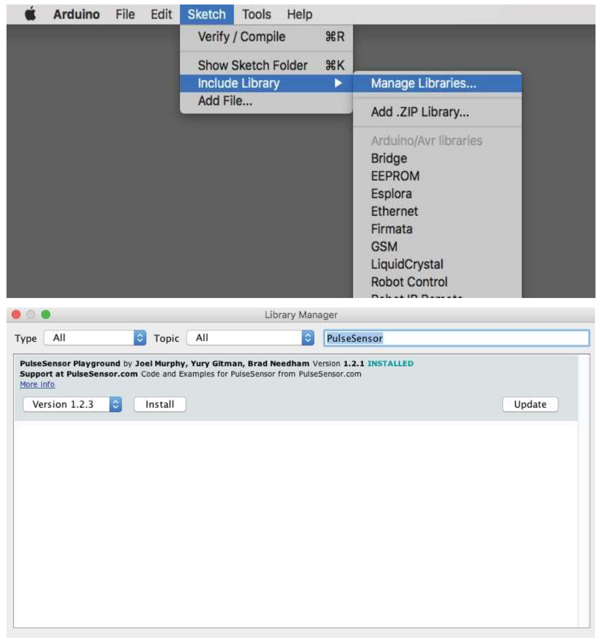
Circuit Diagram

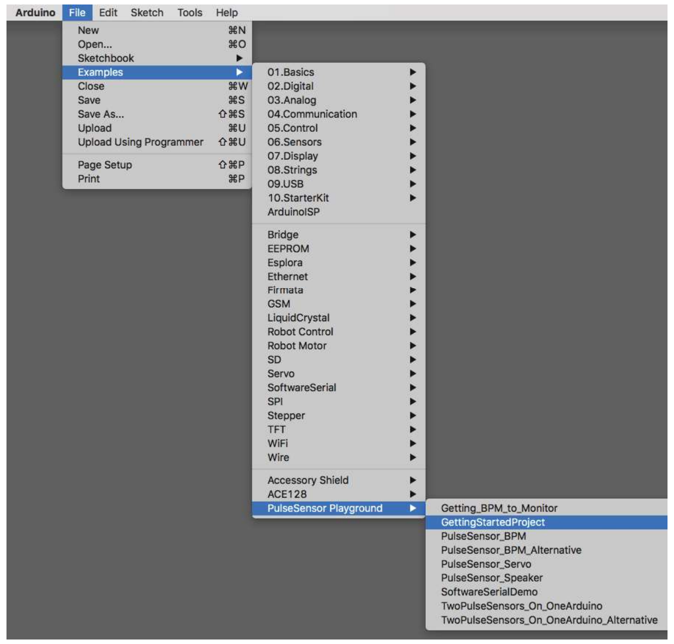


STEP 1

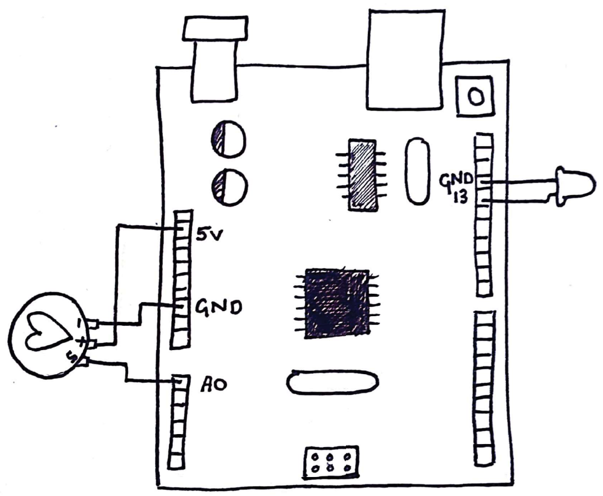
SETUP YOUR COMPUTER

The first step is to prepare your computer and the code that you will run. Open up Arduino IDE and download the PulseSensor Playground library.
As shown in the screenshots: Sketch > Include Library > Manage libaries > (Search for PulseSensor Playground) > Install.
Once installed open up the GettingStartedProject code.
Files > Examples > PulseSensor Playground > GettingStarted Project.



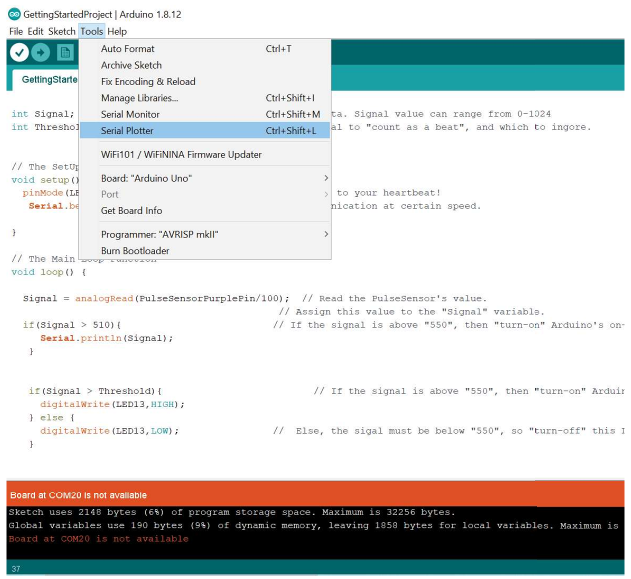


STEP 2



BUILD A TEST CIRCUIT

Now you need to test the code. To do this, the simplest way is to build a prototype circuit. With the Arduino, form the connections as shown in the illustration. This can be done by directly inserting the component into the ports or using jumper cables with a bread board.
LED negative leg, Pulse sensor negative = Ground
LED positive leg = 13
Pulse sensor positive = 5V
Pulse sensor signal (S) = A0



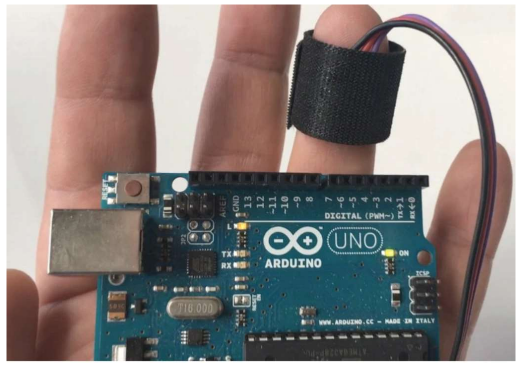
RUN THE CODE

Run the code by uploading it onto the Arduino board.
Make sure you have selected the correct board and COM port for your Arduino.
Tools > Board > Arduino Uno
Tools > Ports > COM...
When you run the code, open the serial plotter to see the graph which is produced from the pulse sensor signal. Make sure to select 9600 baud
Tools > Serial plotter > 9600 baud

STEP 3

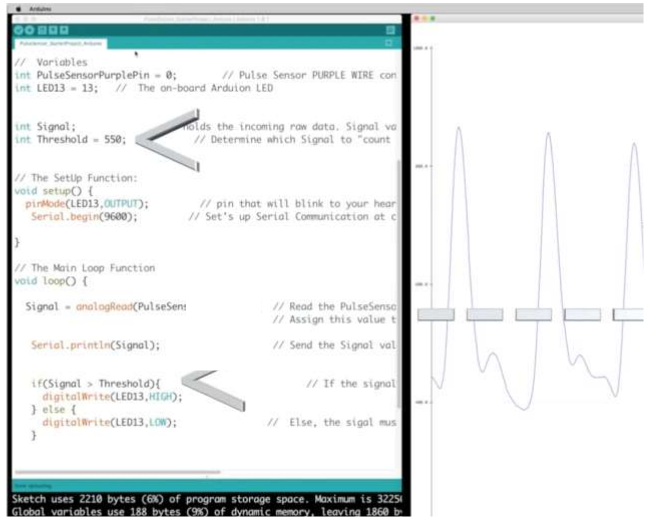
TEST AND MODIFY THE CODE

Set up the sensor as detailed in this document:



WWW.GENERATIONROBOTS.COM/MEDIA/DETECTEURDEPOULSAMPLIFIE/PULSESENSORAMPEDGETTINGSTARTEDGUIDE

Start with the sensor on your finger. Continue to watch the Serial plotter and once settled, move the sensor to find a good position on your wrist. This will be where the sensor sits when you wear your bracelet.



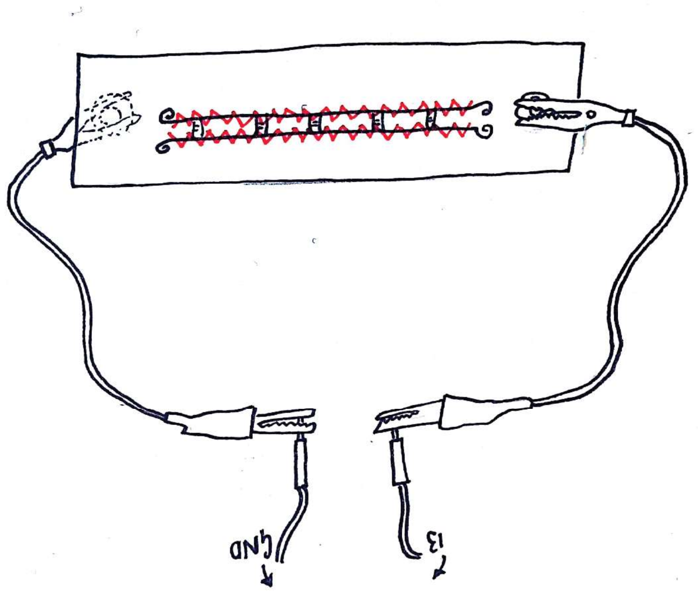
When moving the sensor there will be a lot of noise which will effect the results. Once it has settled again, you can modify the code to make it more accurate.

On the serial plotter, you should be able to see a spike for every heart beat. You can set a more accurate threshold value in the code as indicated by the arrow. Define the threshold value by reading off the serial plotter at which value is surpassed with every heart beat.

STEP 4

TEST WITH THE BRACELET

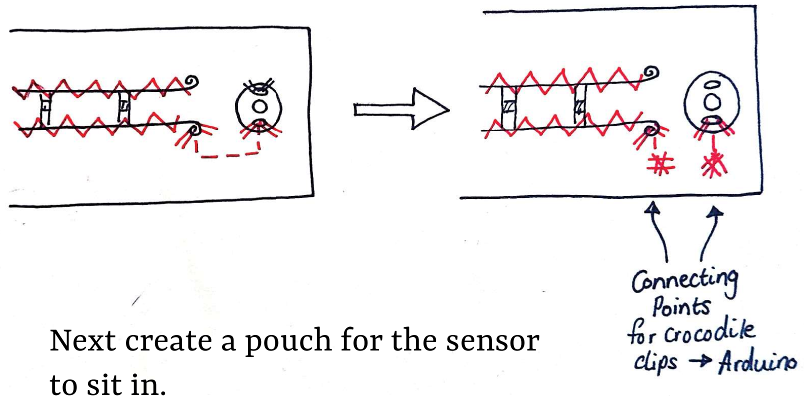
Now that the code is working well and the LED on the Arduino is flashing at the same rate as your pulse, you need to integrate the sensor with the LED bracelet.
First of all, test it will work using crocodile clips as shown in the illustration. Here you are completing the circuit between the two poppers with the Arduino circuit. Here, the bracelet is a replacement for the LED in the test circuit. This means that the crocodile clip connected to the positive popper needs to connect to pin 13 and the other popper to the ground using jumper cables.



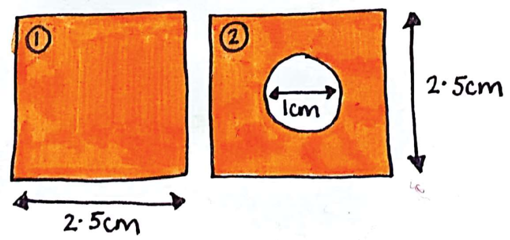
STEP 5

MODIFY THE BRACELET

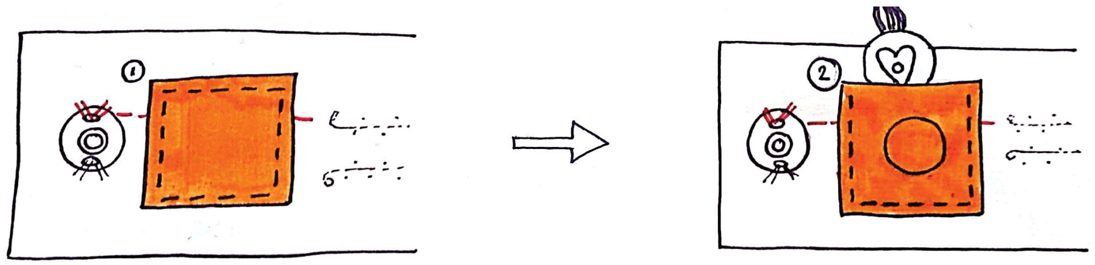
Now you know that it works, you can modify the bracelet. To do this, undo the conductive thread stitching between the thin copper wire and the popper.
Restitch two connecting points with conductive thread.



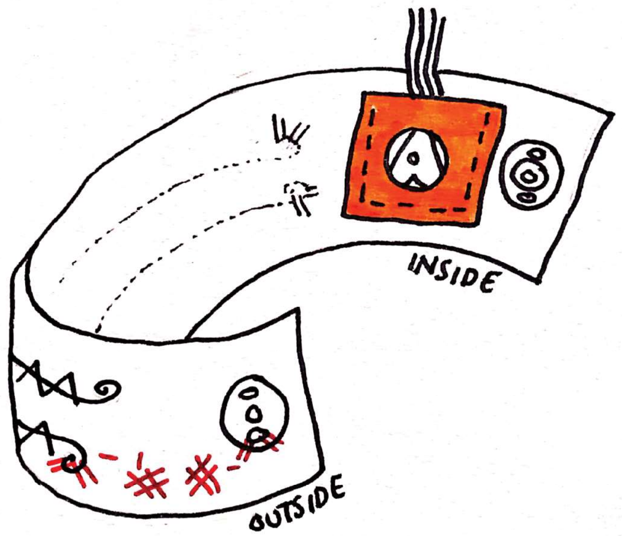
Next create a pouch for the sensor to sit in.



Cut out two squares of fabric (both around 2.5cm by 2.5cm). Cut a small circle in the centre of one of the pieces. This should be big enough for the sensor and light on the pulse sensor to be visible but not big enough for the sensor to fall through.



Sew down the full square piece first - this is to prevent short circuiting.
Then sew down the second piece with the circle cut out. Sew along only three sides as shown in the illustration. The pulse sensor should then be able to slide in when connected to the Arduino.

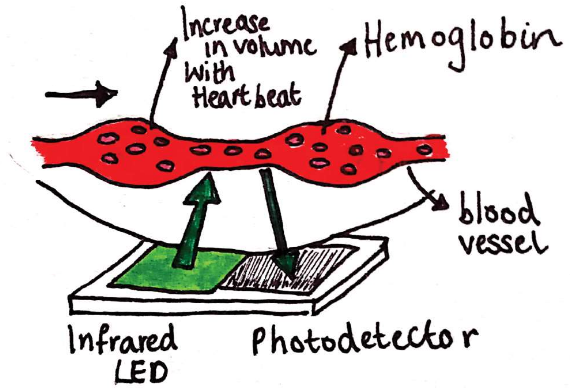


How it Works

In our blood there is oxygenated hemoglobin, which are types of cells that are carrying oxygen around our body. These can absorb light. This means that if there is more oxygenated hemoglobin in our blood more light is absorbed.

Your heart beat forces oxygenated blood around your body, so that the oxygen can be delivered to your organs. The blood is pushed round your body by blood vessels increasing and decreasing in volume.

With every heart beat, the vessels increase in volume, increasing the amount of oxygenated hemoglobin and increasing the amount of light that is absorbed.



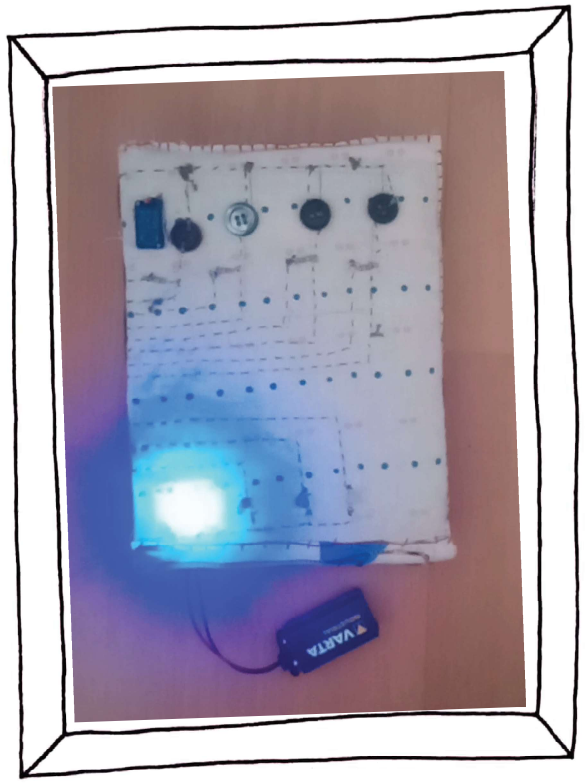
The pulse sensor uses a green Infrared LED (the light source) and a photodiode sensor (a light detector). The infared LED shines infared light and the reflected light is then detected by the photodiode sensor.

The photodiode sensor generates a small voltage and current when light is shone onto it (like a miniture solar panel). The size of the voltage is proportional to the amount of light detected.

Therefore when the volume of the blood vessel increases, with every heart beat there is more oxygenated hemoglobin at that point. This means that more light is absorbed and so less will be reflected. The photodiode will sense less light being reflected and a lower voltage will be given out. When the volume of the blood vessel reduces after the heart beat, the amount of light absorbed is also reduced. More light is then reflected and picked up by the photodiode, giving a larger voltage.

This small change in voltage with every heartbeat can then be amplified through the circuit. In our circuit, the signal from the sensor is continuously being sent to the Arduino, and the amplified result is plotted on the serial plotter. The graph that is shown shows regular peaks and troughs with the repetitive variation in the amount of light being absorbed when the heart beats and the blood vessel volume changes.

Secret Diary Box



Create a box which can only be accessed using a secret code! Using an Arduino and integrating the circuit onto fabric you can create a stylish box to safely store all your precious pocessions.

What you need:

CONDUCTIVE THREAD
NORMAL FABRIC
NEEDLE AND THREAD
SEWING MACHINE

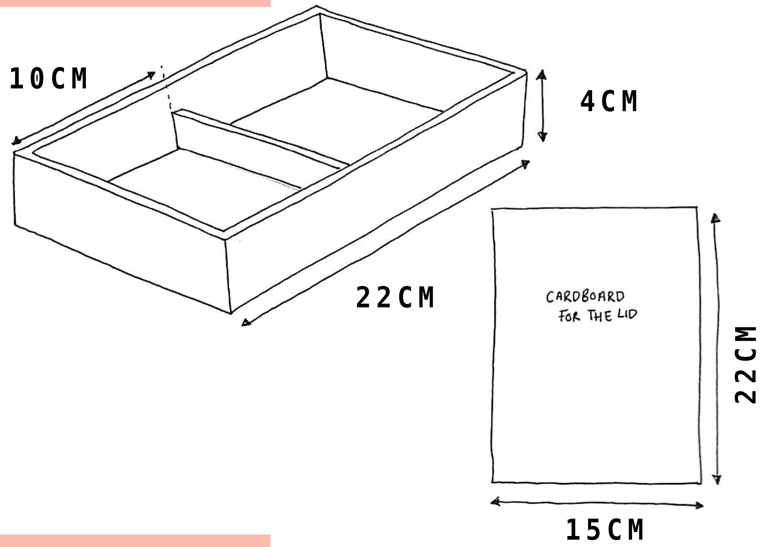
4 LEDS
(DIFFERENT COLOURS)
4 PLASTIC BUTTONS
4 ELECTRONIC BUTTONS
SERVO MOTOR
4 RESISTORS (1K OHM)

ARDUINO UNO
PCB BOARD
THIN INSULATED WIRE
BREADBOARD
JUMPER CABLES
PIN HEADERS
9V BATTERY AND CONNECTOR
SOLDERING IRON AND SOLDER

NEWSPAPER
GLUE, FLOUR AND WATER
PAINT
TAPE
CARDBOARD

STEP 1

CREATE YOUR BOX

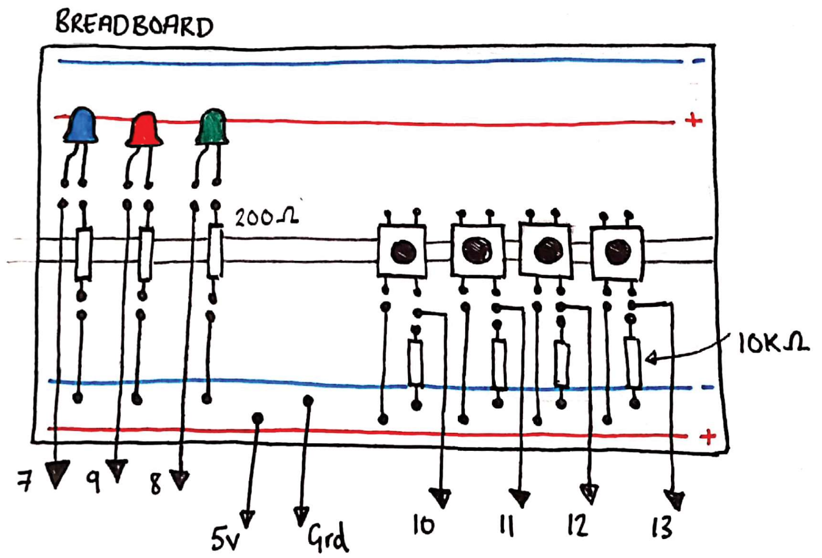


Create your box using cardboard. Cut out the 5 pieces (the base, 4 sides and the middle divider) to create the box on the left, as shown in the illustration. Hold the pieces together using tape. Using glue with a little bit of flour and water, create a paste. Paper mache the box to hold all the elements in place using pieces of ripped up newspaper. Leave to dry then paint. Cut a seperate piece of cardboard for the lid.

STEP 2

BUILD A TEST CIRCUIT

Create a prototype circuit using electronic components, a breadboard and jumper cables. Follow the diagram below, connecting the components with the Arduino ports as defined.



UNDERSTAND YOUR COMPONENTS

Details on how the components work, such as the breadboard, LEDs and buttons, can be found in the 'Understanding your components' section at the beginning of this booklet.

STEP 3

UNDERSTAND THE CODE

Take a look at the code that will be used to program the Arduino. This will create the system that will recognise when someone has tried to enter the box. It will cause the green LED to light up if the code entered is correct or flash red if it is wrong.
Open the file in Arduino IDE and take a look at the notes below. Make sure you understand the code before moving on to the next steps. The Coding Dictionary will help you!

LINK TO CODE FILE

This first section of the code defines all the elements that will be needed later on in the code. This includes any pins of the Arduino connected to a component and defining any initial variables with values.

Define the Arduino pin for each of the LEDS (the variable name which is linked to each LED takes the number of the pin it is connected to)

```
#define RED 8
#define GREEN 7
#define BLUE 6
```

Define your secret code as an Array*

```
const int code[] = {0, 1, 3, 2, 3};
```

Here it calculates how long your code is

```
const int codeLength = sizeof(code) / sizeof(int);
```

sizeof() command gives the size (how many values there are inside) of an Array*

The Arduino pin for each button is being define as part of an Array*

```
const int buttons[] = {13, 12, 11, 10};
```

Here it calculates how many buttons there are using the length of the Array* set above

```
const int buttonsLength = sizeof(buttons) / sizeof(int);
```

This command creates an Array called 'pressed' which has the same number of values in it as the number of buttons provided by 'buttonLength'. These values can either be true or false.

```
bool pressed[buttonsLength];
```

Index defines how many times the code has been run through. This is initially set to 0

```
int index = 0;
```

The bool variable* of correct is given the initial value of true.

```
bool correct = true;
```

The variable attempt defines the number of attempts that have been tried to crack the code. This is initially set to 0.

```
int attempt = 0;
```

The setup section of the code defines the components within the code, this gives the code an idea of what information needs to be received or given to make a component behave in the way we want it to.

```
void setup() {
```

Here we set the pinMode of the Arduino pins.
These 3 examples are for the LEDs. We want the LED to light up or turn off when it is told to do so. This is an output* as it gives an action in reaction to command or signal provided by the code.

```
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
```

Use the pinMode() command with Arduino.
The format is as follows:
pinMode (Ardunio pin number, INPUT or OUTPUT)

Here a for clause* is used to quickly define the pinMode for all of the buttons. This can be done because the buttons were defined in an Array* and all buttons provide an electrical signal when pressed which will be used in the code as an input.

```
  for (int i = 0; i < buttonsLength; ++i) {
    pinMode(buttons[i], INPUT);
  }
}
```

Learn more about this notation in the Code dictionary section.

ARDUINO COMMON ERRORS

- Make sure to have selected the correct device (in this case Ardunino Uno)
- Make sure to have selected the correct port that the Arduino will be connected to
- Make sure to have installed any necessary libaries that are being used in your code (this will be necessary when adding the servo motor.
- Make sure there is a ';' at the end of each command.
- Make sure the names of variables are the same throughout the code. The code is case sensitive!!

This final section of the code, the 'loop' is the main body. This is the part of the code which will continue to be repeated in a loop.

```
void loop() {
```

Here the Arduino is told to light up the blue LED if the value of 'index' is 0 (no attempts have been made)

```
  digitalWrite(BLUE, !index);
```

The digitalWrite() command with Arduino, uses the following format:
pinMode (Ardunio pin number, HIGH or LOW)

A for clause* is used here to go through each of the buttons to see if any have been pressed

```
  for (int i = 0; i < buttonsLength; ++i) {
```

An if clause* is used here saying that 'if button no.() has been pressed' then continue.

```
    if (digitalRead(buttons[i])) {
```

This checks if the same button has just been pressed and if so 'return' meaning ignore this input as it is likely to be an accidental double press.

```
      if(pressed[i]) return;
```

If the code does not 'return' it will complete this command, to make the equivalent entry in the array 'pressed' to be true. recording that the button has been pressed.

```
      pressed[i] = true;
```

The code assigns bool variable 'correct' to true if the number of the button pressed (identified here as 'i') matches (==) to the corresponding button number in the pin set at the beginning.

```
      correct = correct && i == pin[index];
```

This continues to run through until the number of pressed buttons is equal to the length of the code defined at the beginning. This is done using this if clause where index is the number of times it has reached this line of code therefore is equal to the number of buttons pressed successfully.

```
      if (++index == pinLength) {
```

Here, if the inputted pin matches the pin defined at the beginning, the code continues to flash the green LED.

```
        if (correct) {
```

Here the green LED turns on

```
          digitalWrite(GREEN, HIGH);
```

The number of attempts is reset to 0.

```
          attempt = 0;
```

The code then pauses here to make sure the green LED is visible to the user.

```
          delay(1000);
        }
```


else is used here to define the path which the computer will take if the pin entered was not correct.

```
        else {
```

The value of 'attempt' recording the number of attempts tried by the user, will increase by 1

```
        attempt++;
```

The code then checks if the number of attempts is within the limit of 5 attempts the user can make, which in this case is set to 5.

```
        for(int j = 0; j < attempt; j++) {

          for(int k = 0; k < 5; k++) {
```

If so, it will mean that the code was wrong but they can try again, so the device will flash the red LED using the digitalWrite command.

```
            digitalWrite(RED, HIGH);
            delay(100);
            digitalWrite(RED, LOW);
            delay(100);
          }
        }
      }
```

Once either of these two options have been taken and completed, before the code runs back through the loop again, it makes sure that the red and green LEDs are off.

```
              digitalWrite(GREEN, LOW);
              digitalWrite(RED, LOW);
```

It resets the index (the number keeping track of how many times a button was successfully pressed) is reset to 0.

```
              index = 0;
```

correct is reset to its default state of true

```
              correct = true;
            }
```

Finally, for the situation where no button is pressed in the first place, no true value is assigned into the pressed array.

```
            } else {
              pressed[i] = false;
            }
          }
```

A small delay is then included to counteract the incredibly fast speed that the arduino works at, before the loop starts again.

```
          delay(10);
        }
```

TEST RUN YOUR CODE

Complete a test run of your code,with you Arduino correctly connected to all the components in the bread board. C

STEP 3

MODIFY THE CODE

Now that we have understood the code and tested that it runs with the circuit prototype, we can modify the code to work as a secret diary box. We can continue to test it with our temporary circuit. First we will need to include a white LED which flashes each time a button has been pressed. Then we will add a servo motor, which will turn when the correct code has been entered, whcih will eventually act as the lock for our box.

ADD A WHITE LED

Hint #1: Steps to include

- Define the Arduino pin for the LED
- Set the pinMode for the LED
- Include the commands to make the white light flash when a button has been pressed.

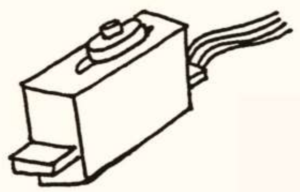
Hint #2: Lines of code to include

```
#define WHITE 9

pinMode(WHITE, OUTPUT);

digitalWrite(WHITE, HIGH);
delay(100);
digitalWrite(WHITE, LOW);

digitalWrite(WHITE, LOW);
```

Hint #3: Modified code

INSERT LINK TO MODIFIED CODE FILE

ADD A SERVO MOTOR



Hint #1: Steps to include

- Install and include the library for a servo motor
- Create an object recognised as 'servo motor'
- Define the Arduino pin for the servo motor (5)
- Define the Open and Closed states of the motor
- Attach the servo to the number corresponding to the Arduino pin
- Setup the servo to be initially closed
- Define the pinMode of the servo
- Make the servo turn when the correct code is given
- Make sure the servo is turned off before resetting for the next attempt

Hint #2: Lines of code to include

Below are the different lines of code that are needed to be added to the original code to make the servo motor work. Try to put them into the code in the right place.

```
#include <Servo.h>

Servo servo;

#define SERVO 5

const int Closed = 50;
const int Open = 100;

servo.attach(SERVO);

servo.write(Closed);

pinMode(SERVO, OUTPUT);

digitalWrite(SERVO, HIGH);
servo.write(Open);
delay(4000);
servo.write(Closed);
delay(10);
digitalWrite(SERVO, LOW);

digitalWrite(SERVO, LOW);
```

Hint #3: Modified code

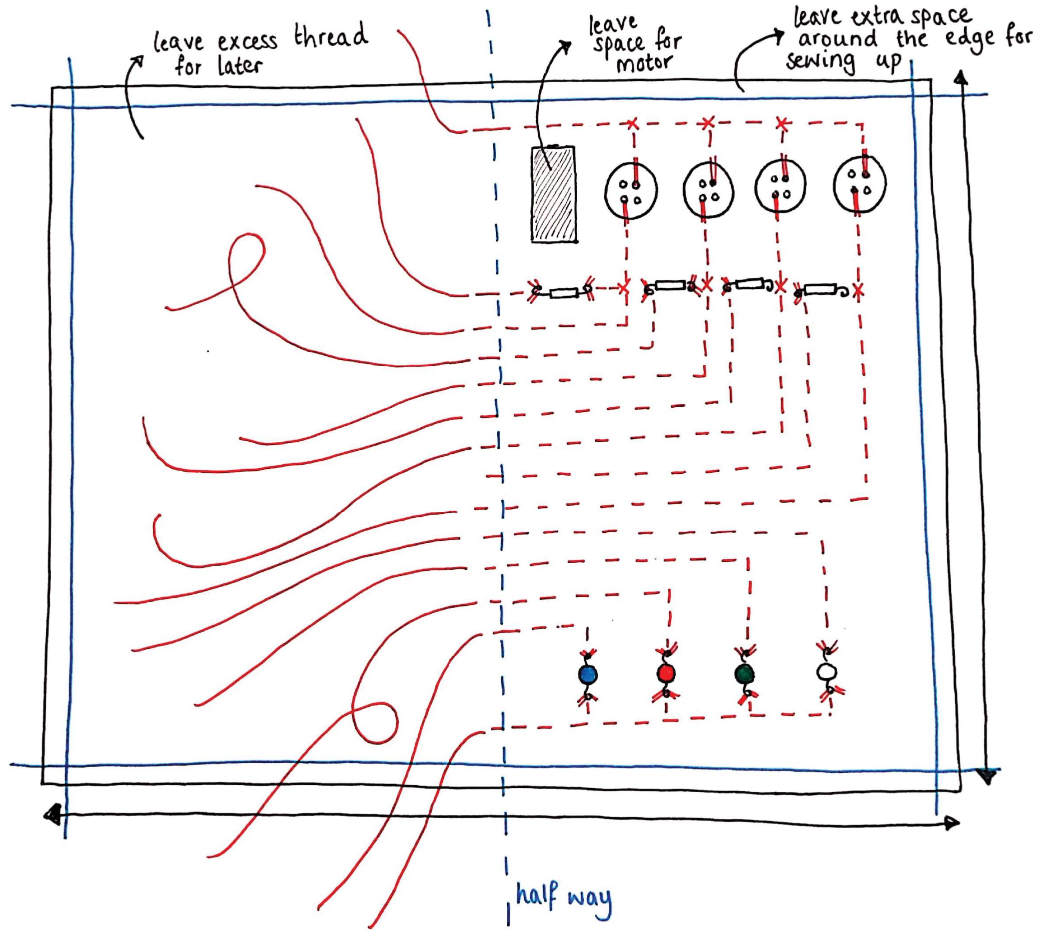
INSERT LINK TO MODIFIED CODE FILE

TEST RUN YOUR CODE

Complete a test run of your code,with you Arduino correctly connected to all the components in the bread board. Compare it to the video.

STEP 5 TRANSFER THE CIRCUIT ONTO FABRIC

Steps 5 to 7 are the most complicated steps for this project. Make sure to take your time and regularly test the circuit to make sure there are no errors or bad connections.
To transfer the circuit onto fabric, follow the illustration below. First, sketch out the stitching onto the fabric. You will then be able to identify where to sew on each of the buttons, resistors and LEDS. Do this with normal thread to hold them in place. Then using conductive thread make the connections.
Each connection should be made using a seperate piece of conductive thread.
Make sure to leave a lot of excess thread as shown in the illustration.



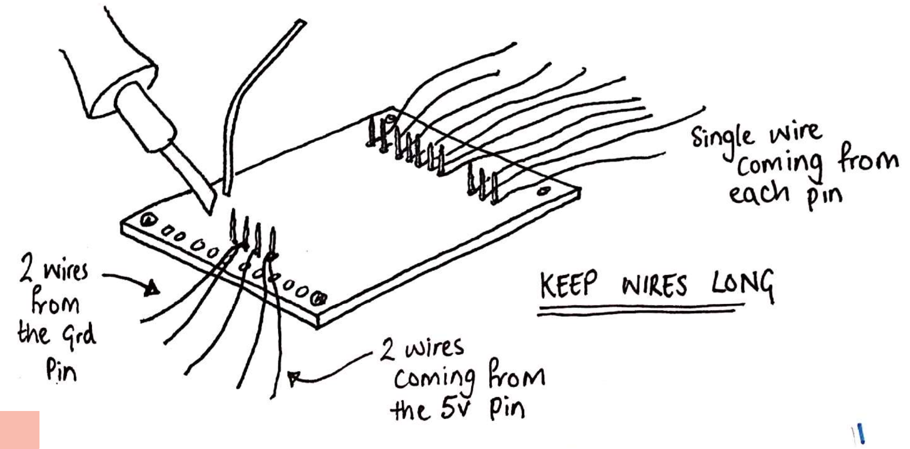
TEST RUN YOUR CODE

Test your sewn circuit by connecting your Arduino. Use crocodile clips to connect the conductive threads to the correct Arduino ports. Make sure the thread isn't touching any other thread which could cause a short circuit.
If it doesn't work, it could be because:

- Button connections are constantly in a state of 'ON' (by thread always touching)
- Resistors are not connected correctly (they must branch off from the button)
- LEDs are not connected the correct way round.

STEP 6 PREPARE THE PCB BOARD

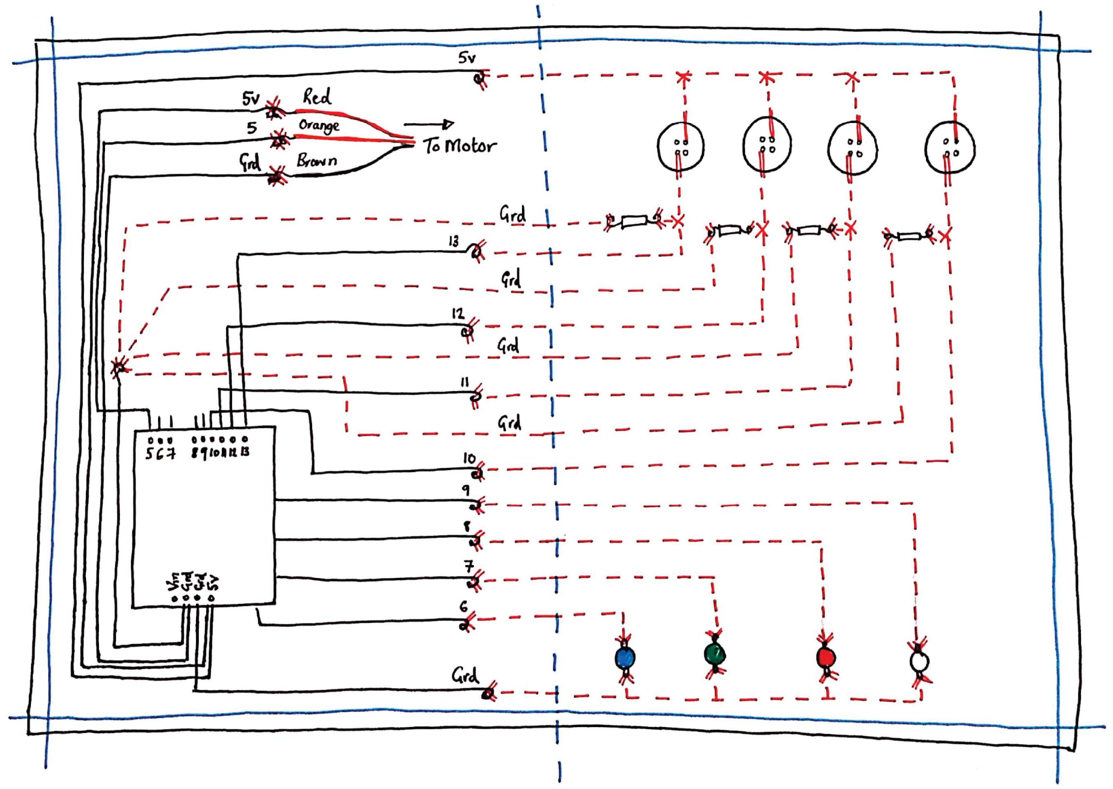
Using solder and a soldering iron, prepare your PCB board with pin headers in the correct locations to match up with the Arduino ports you are using. Connect long pieces of insulated wire to each. Make sure 2 wires come from the Ground and 5V pins.



STEP 7 CONNECT THE PCB BOARD

Follow the diagram below to connect the PCB board to the sewn circuit. Make sure to:

- Position the PCB so that the Arduino can be connected and will enter the diary box when closed and sit in the smaller section seperated by the divider.
- Sew down the insulated wire. We advise to use a sewing machine with a zigzag stitch. By hand you can use a couching stitch, but this will take a lot longer. It can be a good idea to use take to hold down the wire until it is stitched into position.
- Strip the end of the wire, curl it up and sew it down. Then connect the correct conductive thread to its corresponding pin by forming a connection at this spiraled end of the insulated wire. Make sure no conductive thread overlaps!

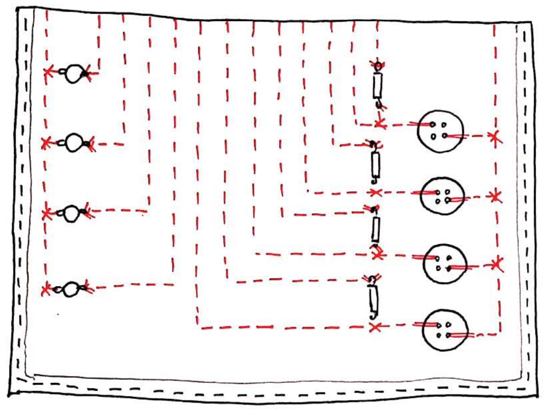


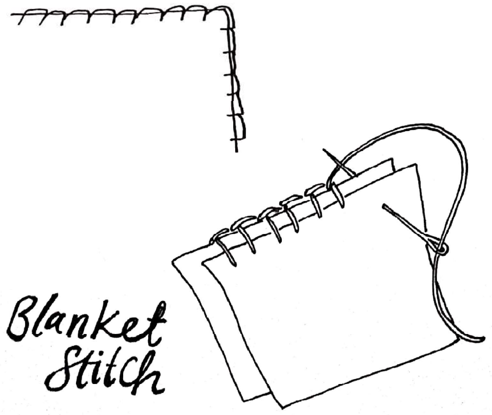
TEST RUN YOUR CODE

Connect the Arduino to the PCB board and a battery and check the circuit and code continues to work as expected.

STEP 8 PREPARE THE LID

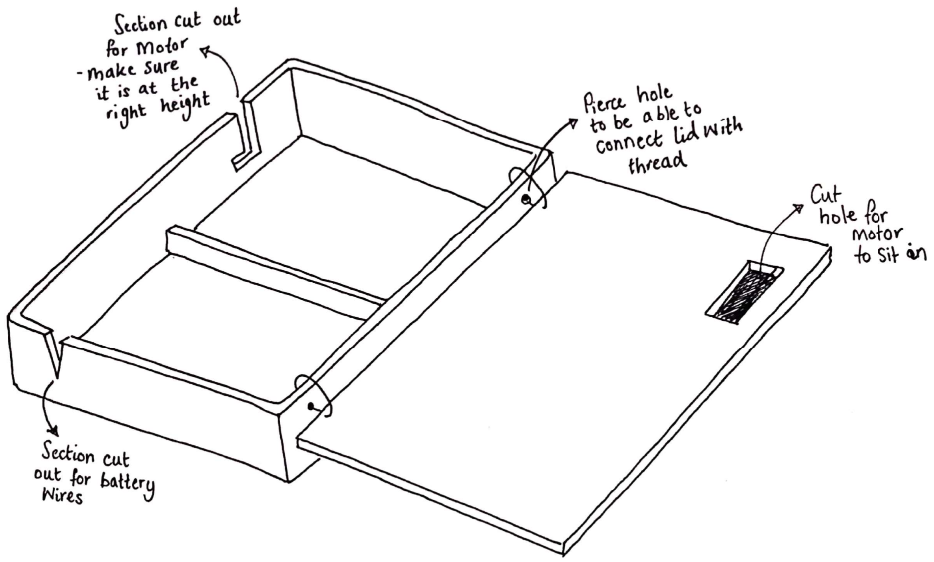
Line up the card board to sit between the front and back parts of the fabric. Hand stitch around the edges to enclose the cardboard. You can then fold over the rough edges and tidy them up using a blanket stitch.





ATTACH THE LID AND FINISH

One the lid, using a stanley knife, cut out a hole for the moto to sit in. Pierce two holes into the lid and two into the side of the box. Connect the lid to the box by threading thread between these holes as shown in the illustration. In the side of the box, cut out an 'L' shaped section. This is for the motor head to lock into when it turns. Check the height needed before cutting. The bottom of the L should be cut at the height of the motor arm when the lid is closed. Finally cut a slit in the bottom side of the box for the battery connection wire to go through.



GALLERY

CODING DICTIONARY

VARIABLES
A variable is data or information that is in a form that Ardunino can understand. It can hold a value in many forms such as a number, a series of numbers or text. There are different types of variable datatypes which are used in the context of Arduino programming. Here are some examples:

INTEGER (INT)
This is the primary datatype of number storage for Arduino. This is when the value is a whole number that can be positive, negative or zero (e.g 3).
An integer variable can be assigned like so: int Three = 3. Here, the variable 'Three' is assigned the value of 3. The command int() can convert the variable in the brackets to hold an integer value.

BOOL
A bool holds one of two values; true or false.
A bool datatype can be assigned like so: bool Running = true;
'true' and 'false' are predefined expressions in the Arduino language. They could be translated as false = 0 and true = 1.

ARRAY
An array is a collection of data that are accessed by a single variable name. The command 'int Numbers[] = {1, 2, 3, 4, 5, 6}' creates an array of 6 integer numbers that go under the variable name of Numbers.
A value within an array can then be changed using the command structure 'Numbers[0] = 8'. For example, this would change the array to {8, 2, 3, 4, 5, 6}. You can also create an array of a certain number of entries but not specify their values using the command 'int Numbers[9]'. This will create an array of 10 unspecified integer values.

When defining the pin of an Arduino through PinMode, you need to state whether it is an input or output.

INPUT
This is when an electrical signal is being received to an Arduino pin. For example, it may be the signal from a temperature sensor or a button. This signal will then in some way provide data that will be used in the code.

OUTPUT
This is when an electrical signal is given out by an Arduino pin for a circuit or component. For example, the code will result in an electrical signal being sent out so an LED lights up.

CONTROLS AND COMMANDS
In the Arduino language you can use different structures so that commands are completed in a certain way. Here are some common ones:

'FOR'
The for statement is used to repeat a block of statements enclosed in curly brackets. The for statement is useful for any repetitive operation that needs to be completed under certain conditions or for a certain number of times.

```
for (int i = 0, i < 5, i++)
{ACTION};
```

This for statement is saying that 'i' will start with the value of 0. The ACTION will then run whilst 'i' is less than 5. Each time it runs through, the value of 'i' will increase by 1 (this is what i++ means). This means that the ACTION will be completed 5 consecutive times in total.

'IF'
An if statement checks to see if a condition is met before executing a command.

```
if (Odd = true)
{ACTION};
```

This if statement is saying that if the variable Odd has the value of true, then the ACTION can be completed. Otherwise the ACTION will not be completed.

'ELSE'
Else can be used with 'if' statements, as it can define what happens if the 'if' action is not completed.
For example, the code might state that an action must happen if a condition is completed. Then an 'else' clause is used after to state that a second action will take place if the first action doesn't take place because the 'if' condition is not met.
Multiple else clauses can be used to define multiple conditions. This gives increased control over the direction of the code.

```
if (x < 5)
{ACTION 1}
else if (5 > = x > 10)
{ACTION 2}
else
{ACTION 3}
```

CC BY NC