



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



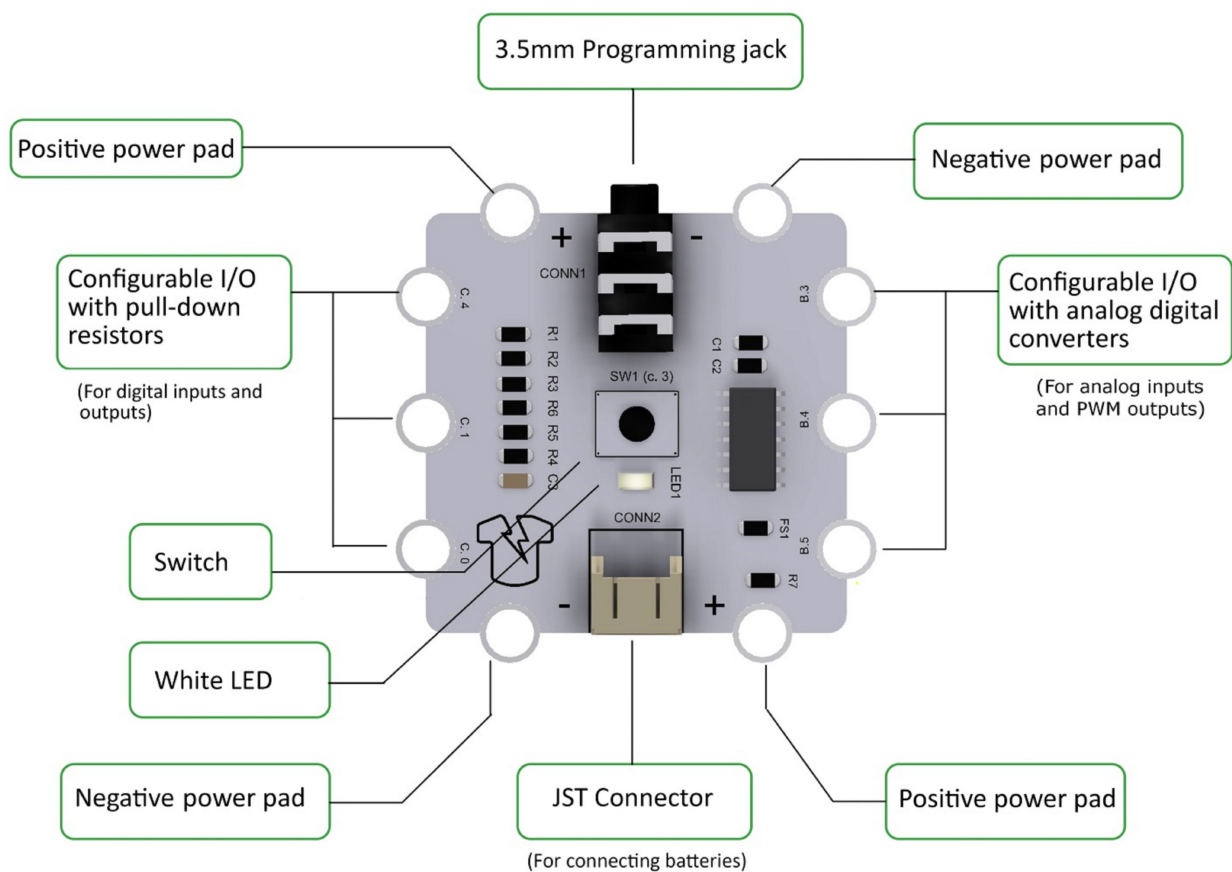
## Electro-Fashion: Igloo, PICAXE wearable module Starter Guide

### Introduction

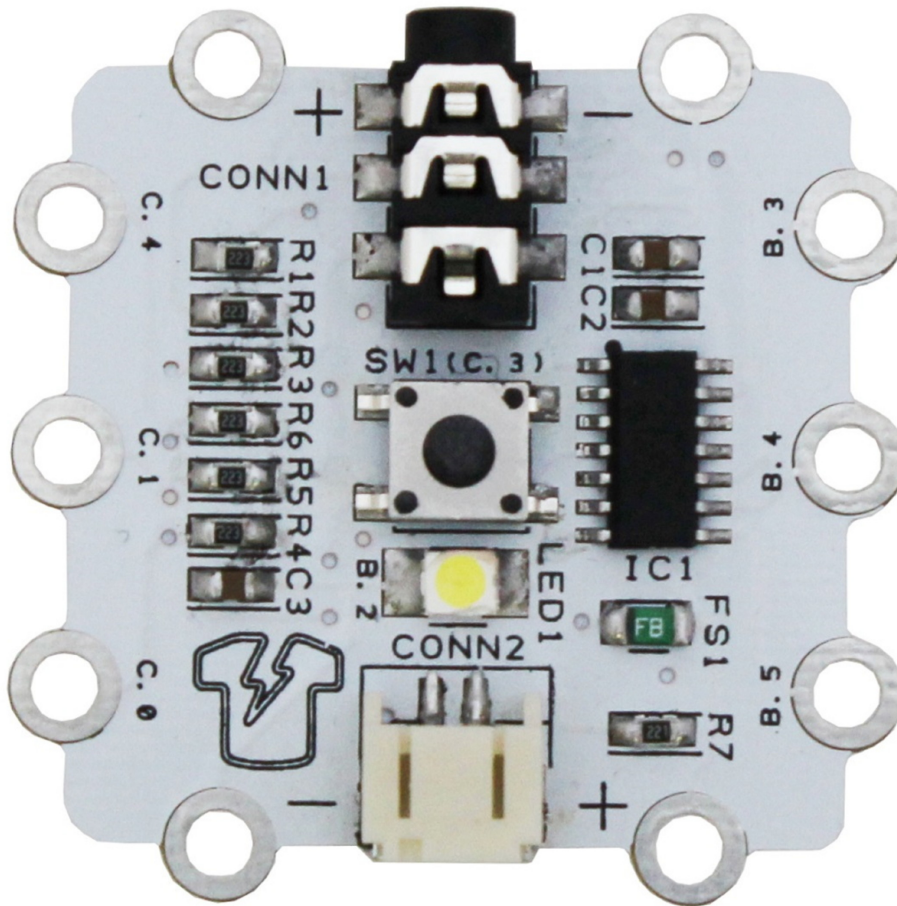
The Igloo: PICAXE wearable module has been designed specifically for use in E-textiles projects and provides a cost effective, fun way to learn about electronics and programming in a textiles environment.

The board uses a PICAXE14M2 microcontroller. This is a chip that can be programmed using the free PICAXE Programming Editor software to respond to various inputs like switches and sensors and control outputs like LEDs or buzzers.

This guide will provide all the information you need to get started with the Igloo PICAXE wearable module.



## Hardware



The Igloo PICAXE wearable module has 6 I/O pins.

The three on the left (C.4, C.1 and C.0) have pull-down resistors on the board. This means you can attach a switch between the “+” pad at the top or bottom of the board and any of those inputs and have a switch which goes high when closed and low when opened.

The three on the right (B.3, B.4 and B.5) have no pull-down resistors so they can be connected to analogue inputs such as our Light Sensor or a potentiometer.

All six can also be used as outputs if required.

The processor input C.3 is connected to the on-board push switch. The pin is normally low, going high when the switch is pressed.

The processor pin B.2 is connected to the on-board LED.



## Connections

Name	Description	Additional Features
C.0	I/O pin with 10K pull-down resistor	ADC, Touch, PWM, hPWM
C.1	I/O pin with 10K pull-down resistor	hPWM
C.3	On-board switch with built in pull-down	
C.4	I/O pin with 10K pull-down resistor	Touch, ADC
B.2	White LED with current limit resistor	PWM
B.3	I/O pin with ADC, no pull-down	ADC, Touch
B.4	I/O pin with ADC, no pull-down	ADC, PWM, Touch
B.5	I/O pin with ADC, no pull-down	ADC, Touch, hPWM
"_"	Negative/GND from battery	
"+"	Positive voltage from battery	Thermal Reset fuse hold current = 100mA
CONN1	3.5mm Programming Jack	
CONN2	Power connector	2mm JST compatible

## Electrical Information

Operating Voltage	2.5-5.5V
Max current per pin (sink or source)	20mA
Max current total for all pins	90mA
Max current draw by on-board LED	20mA @5.5V
Max input impedance for ADC pins	20K Ohms
Min input voltage - Logic HIGH	0.8 x Operating Voltage
Max input voltage - Logic LOW	0.2 x Operating Voltage

## Glossary of Terms

ADC – Analog Digital Conversion. This means the pin can read in a voltage and provide a number between 0 and 255.

PWM – Pulse Width Modulation. The pin can pulse on and off very quickly and allows the user to vary the length of the pulses and also the time between pulses. This is useful for dimming LEDs.

hPWM – Hardware Pulse Width Modulation. The same as PWM but it is capable of pulsing much more quickly, making it good for sending data or dimming LEDs.

Pull-down resistor – A resistor between the pin and 0V/GND. This pulls the pin to 0V when it isn't being pulled high by an input. This ensures the pin is always held in a valid logical state.

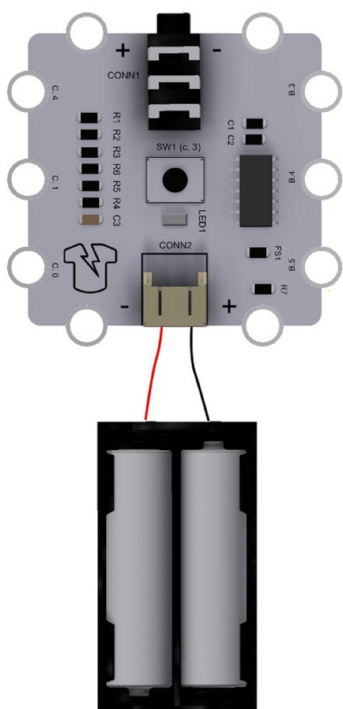


## Power

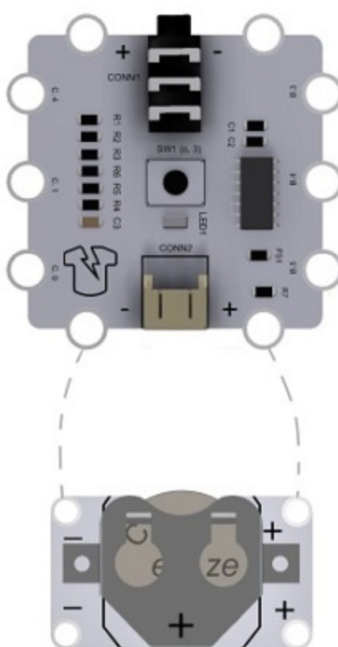
There are several ways you can power your Igloo PICAXE wearable module:

- 1) The simplest way is with a 2xAAA alkaline batteries in a battery cage with JST connector. Zinc carbon or rechargeable NiMH batteries will not work well as they drop in voltage very quickly, meaning any LEDs attached will start to dim.
- 2) Alternatively you can sew a [coincell power board](#) with [battery](#) to the "+" and "-" pads at the top or bottom of the board using our [conductive thread](#).
- 3) Or a 3.7V rechargeable LiPo battery with a 2mm JST connector will also work.

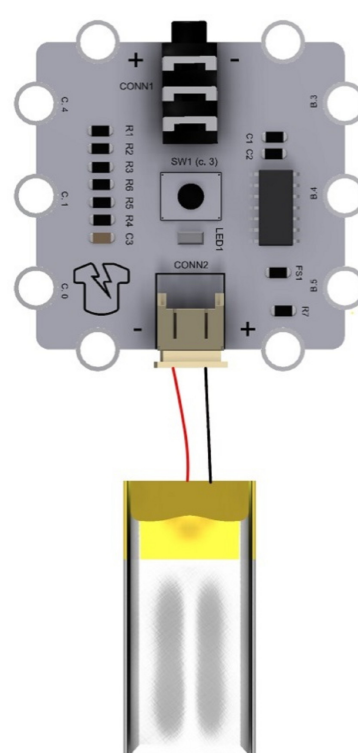
### 2xAAA



### Sewable Coincell



### Rechargeable LiPo



<ul style="list-style-type: none"> <li>+ Longer battery life</li> <li>+ Low cost</li> </ul>	<ul style="list-style-type: none"> <li>+ Slim profile</li> <li>+ Lightweight</li> </ul>	<ul style="list-style-type: none"> <li>+ Long battery life</li> <li>+ Slim Profile</li> </ul>
<ul style="list-style-type: none"> <li>- Bulky</li> <li>- Heavy</li> </ul>	<ul style="list-style-type: none"> <li>- Shorter Battery Life</li> <li>- Sewing required</li> </ul>	<ul style="list-style-type: none"> <li>- Special charger required</li> <li>- Higher initial cost</li> </ul>

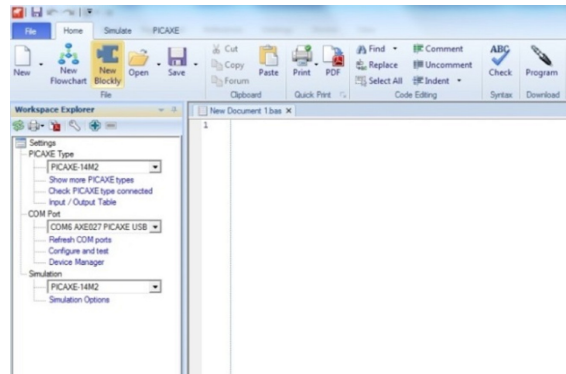


## Programming

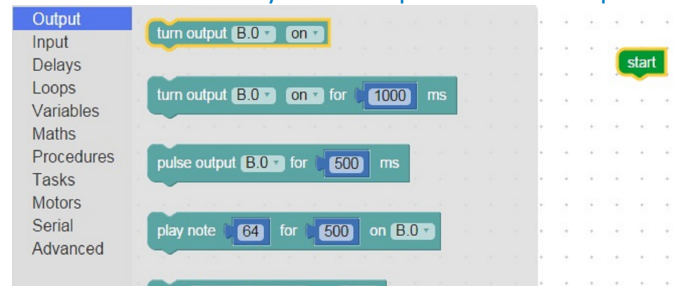
**Important:** To program the board you will need a USB PICAXE Programming Cable (available [here](#)).

You will also need to install drivers and software available from our [PICAXE quick start webpage](#). Installation details can be found on that page.

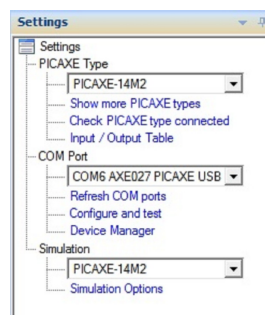
The images in the examples below are taken from PICAXE Programming Editor 6. If you are using the Blockly App or the Chrome Blockly extension your screen may appear slightly different.



Install and run the program, then click “New Blockly” in the top left corner to open the Blockly interface.



From here you can browse the available blocks in the categories down the left hand side and arrange them into programs in the space to the right. Remember to connect to the green “start” block.

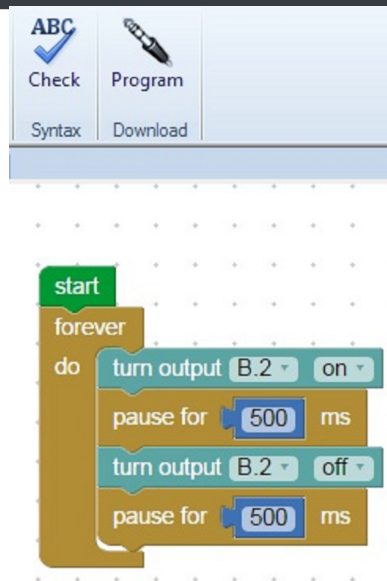


Once your code is complete, check the settings at the far left of the screen. Make sure you have the correct COM port selected from the dropdown menu (it will be the one that says PICAXE USB). Also check you have the correct chip type selected (PICAXE 14M2).



# Igloo: PICAXE wearable module Starter Guide

www.kitronik.co.uk



Check the programming cable is fully inserted into the Igloo PICAXE wearable module and that the board has a power supply attached then click the “Program” icon near the top of the screen. A green progress bar will appear at the bottom right of the screen and it will inform you when the code has successfully been uploaded.

The program shown above can be used to check the set-up and will flash the on-board LED.



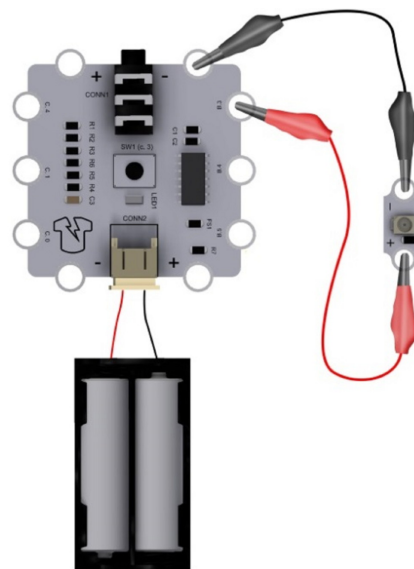
## Connecting Components

There are two ways to connect the PICAXE Control Board to your other e-textiles components.

### Using Crocodile Clips

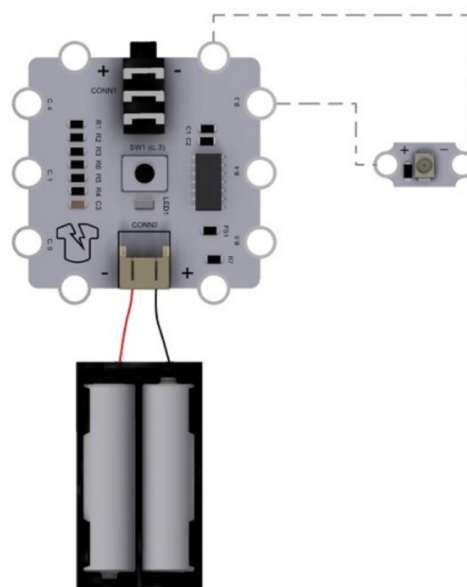
The fastest way to do this is by using crocodile leads. It is a good idea to test your circuit this way before sewing it into a project as it is much easier to make changes if your circuit is incorrect.

To connect an output such as an LED simply use one lead to connect the negative leg of your LED to either of the negative pads on the PICAXE Control Board, and another lead to connect the positive leg of the LED to the output pin you wish to use.



### Using Conductive thread

Once you are happy with your circuit you can use our conductive thread instead of the crocodile leads. The thread needs to be wrapped tightly around the metal pads on the E-textiles components several times to give a good connection. Detailed guides on using conductive thread can be found [here](#).



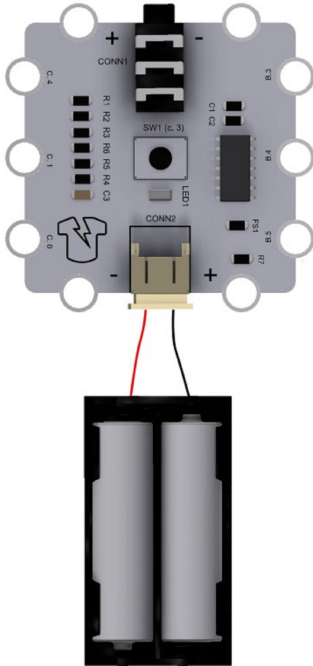


## Hook-up Guide

Below are some simple circuits with accompanying example code to help get you started. The code has been made using Blockly for PICAXE, but the circuits will work just as well with BASIC or the flowcharts in PICAXE Programming Editor.

### Switch (momentary) – On-board digital input and output

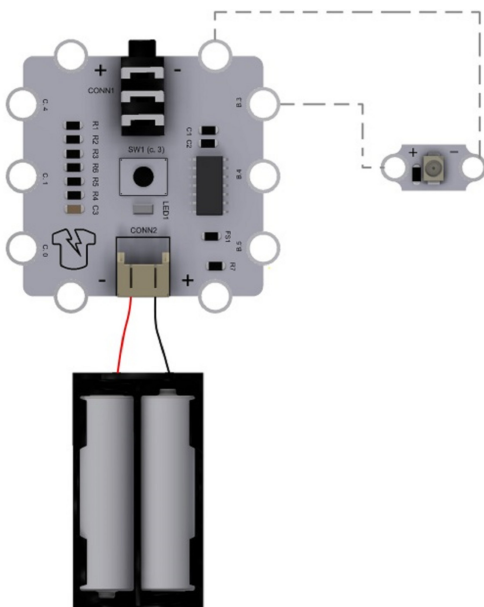
This example demonstrates how to use the on-board push button switch as a momentary switch to turn the on-board LED on and off. The LED will come on while the button is held down and turn off when the button is released.



```
start
forever
do
  if input C.3 is on
  then
    turn output B.2 on
  else
    turn output B.2 off
```

### Blink – External Digital Output

This example demonstrates how to use an Igloo PICAXE wearable module to make a [sewable E-textiles LED](#) flash on and off.

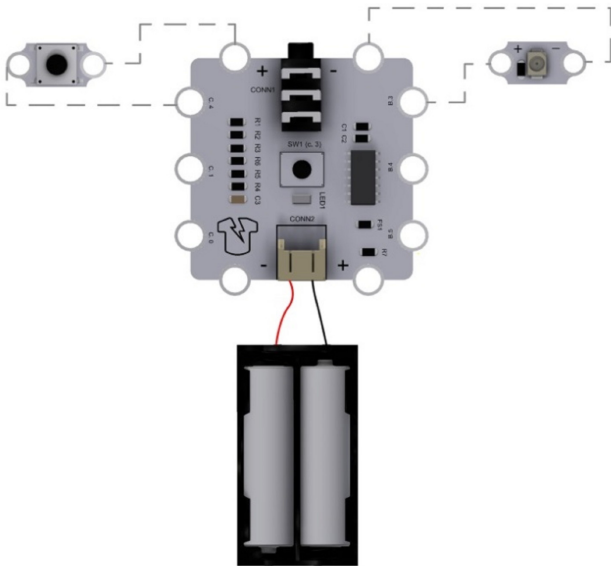


```
start
forever
do
  turn output B.3 on
  pause for 500 ms
  turn output B.3 off
  pause for 500 ms
```



## Switch (latching) – External digital input and output

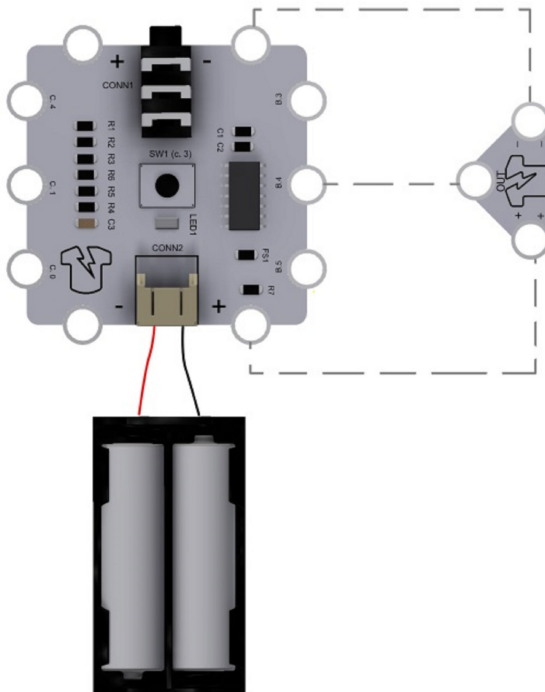
This example demonstrates how to use a [sewable push button switch](#) as a latching switch to turn a [sewable LED](#) on and off.



```
start
forever
do
  repeat
  until input C.4 is on
  turn output B.2 on
  pause for 100 ms
  repeat
  until input C.4 is off
  repeat
  until input C.4 is on
  turn output B.2 off
  pause for 100 ms
  repeat
  until input C.4 is off
```

## Light Sensor – External Analog Input and digital output

This example demonstrates how to use a [sewable light sensor](#) to turn the on-board LED off when it gets dark. The light sensor is facing downward so if you are sewing this into fabric you need to make a hole for the light sensor to poke through.



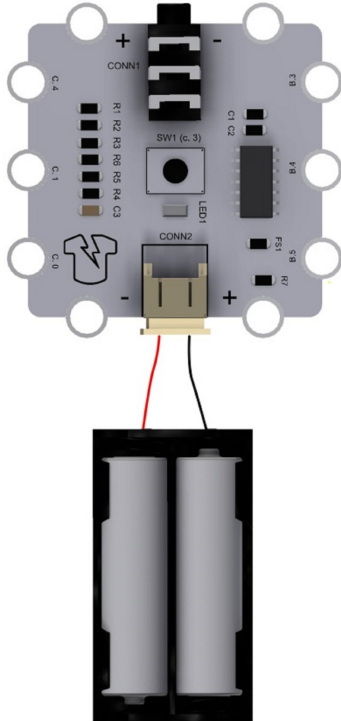
```
start
forever
do
  read analogue B.4 to varA
  if varA > 100
  then turn output B.2 on
  else turn output B.2 off
  pause for 50 ms
```

In this example we read the input and check to see if it is over 100. In total darkness the reading will be virtually 0 and under a very bright light the reading will be close to 255.



## Fading LED – Pulse Width Modulation

By rapidly turning the LED on and off we can control the brightness. By altering the amount of time the LED spends on (the duty) we can fade the LED.



```
start
set varA to 120
forever
do
repeat
set pwmout period 50 duty varA on B.2
pause for 1 ms
decrease varA by 1
until varA = 5
pause for 50 ms
repeat
set pwmout period 50 duty varA on B.2
pause for 1 ms
increase varA by 1
until varA = 120
pause for 500 ms
```

In our example the first segment of the code decreases the duty (pulse length) every millisecond until it reaches 5. Then the second segment does the reverse and increases it until it reaches its original value of 120.

The graphs below show what a dim PWM signal would look like (left) and a brighter PWM signal (right).

