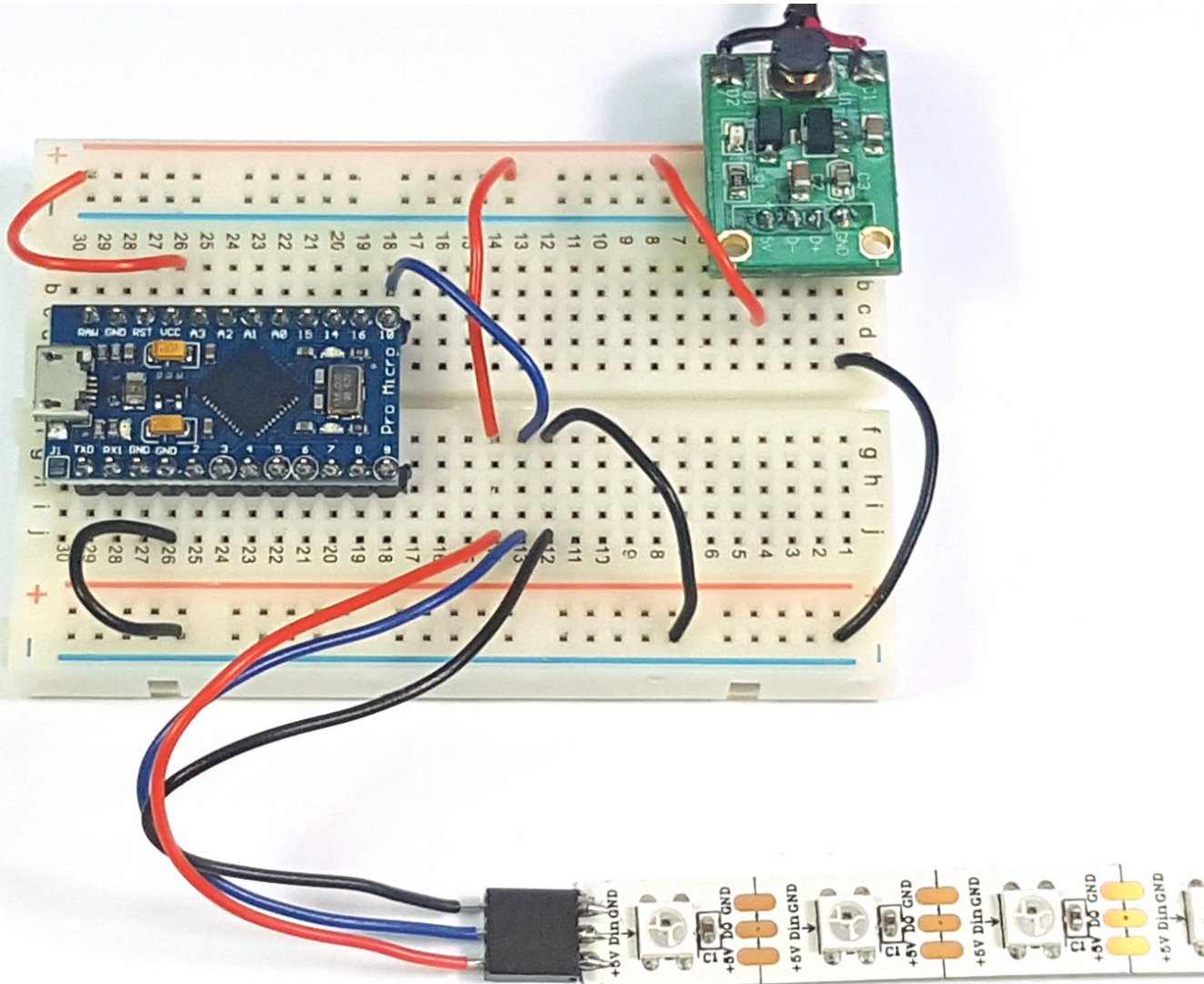


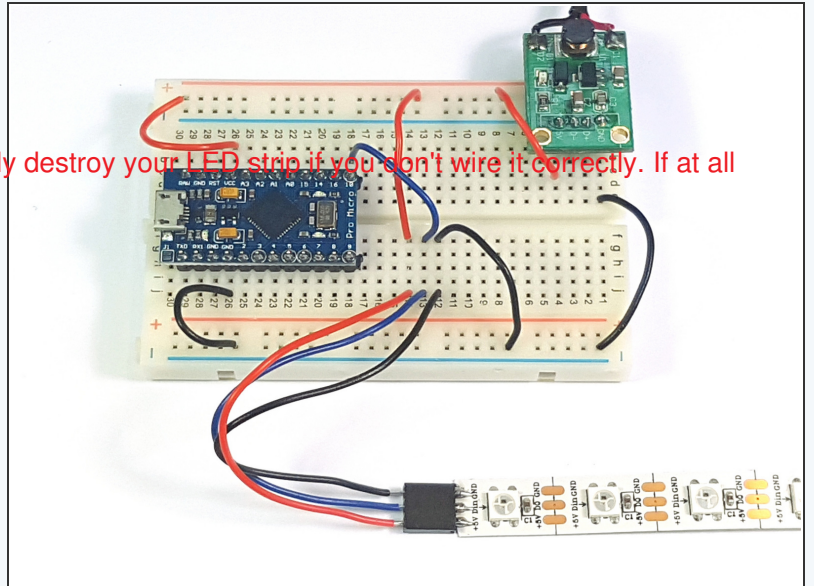
Inventor School Session 7 - LED strip



— Wiring the LED strip

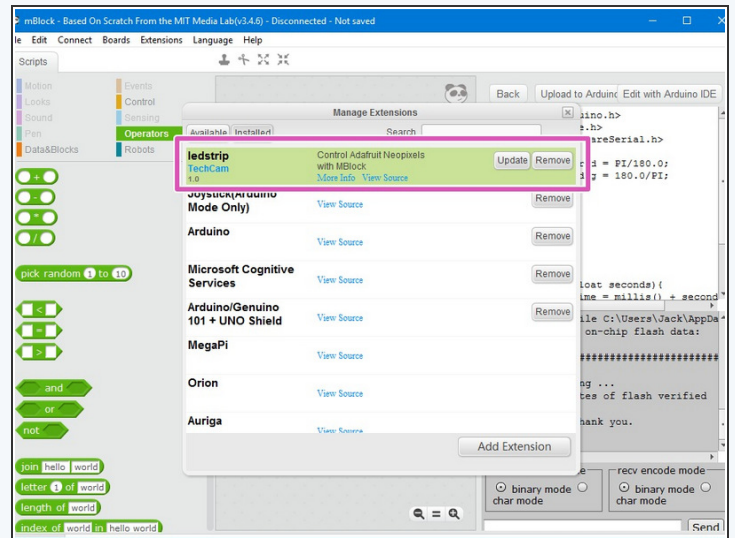
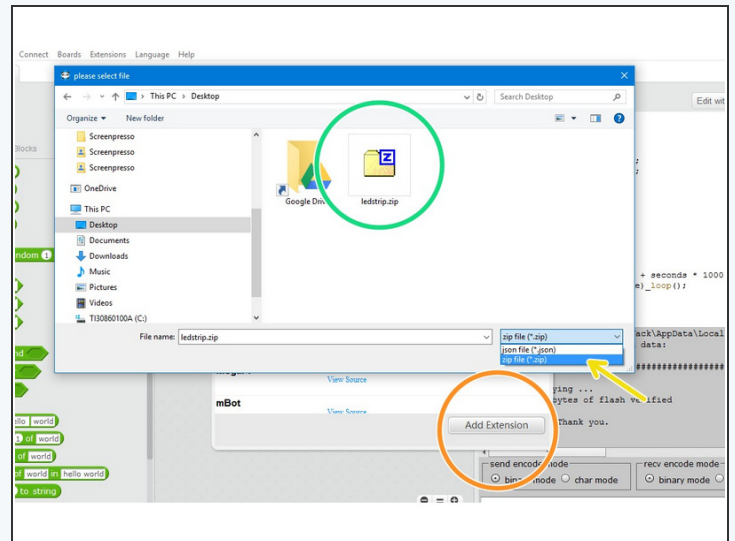
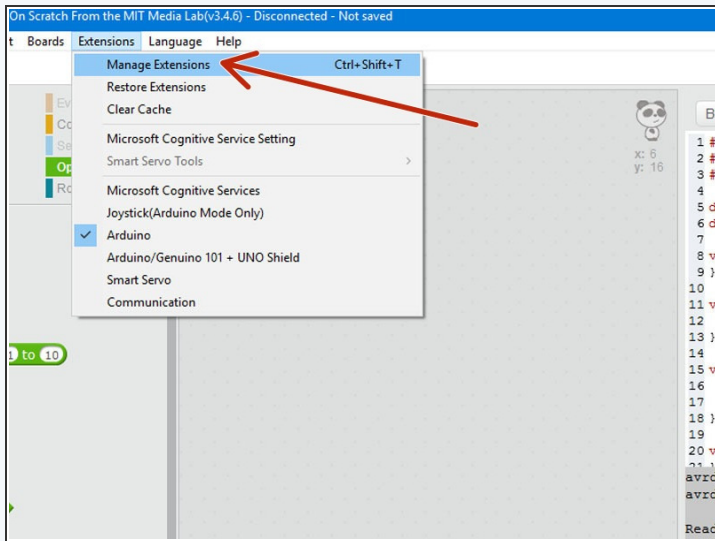
- Wire up your LED strip like the diagram

⚠ As always check your wiring carefully - you could easily destroy your LED strip if you don't wire it correctly. If at all unsure, ask your tutor to check.



— Installing the Library

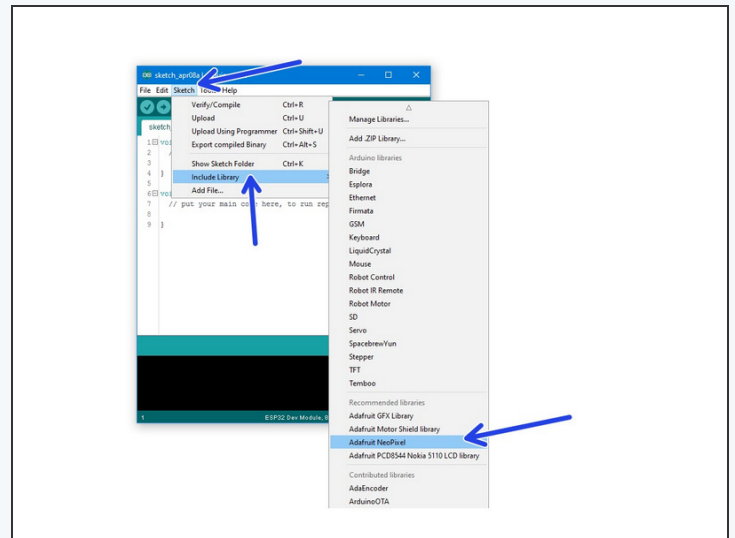
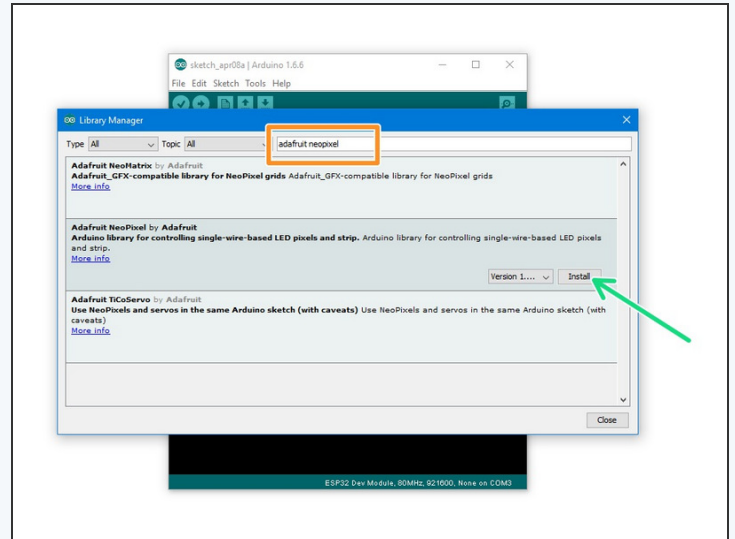
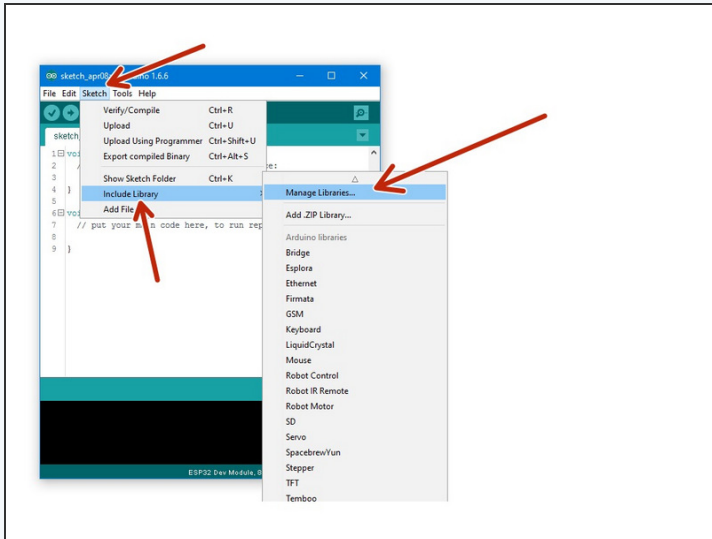
- (M)



- Unfortunately MBlock can't control our fancy LED strip without some **help**!
- We need to install an **extension** to allow the use of another block to control the LEDs.
- Download the extension from [this link \(https://drive.google.com/open?id=0B9fks-SkOzQYQTg1bTFUZmhpQzQ\)](https://drive.google.com/open?id=0B9fks-SkOzQYQTg1bTFUZmhpQzQ) and save it to your computer.
- Click on **Manage Extensions** in the **Extensions Menu**
- Click on **Add Extension**
- Change the file type to **ZIP**
- Find the file you downloaded and select it
- Click **open**, and the extension should be installed!

Step 3 — Installing the Library

(A)



- We need to install a **library** to make the LEDs work - a library is just some code someone else has written we can use to make controlling things **easier**.
- Click on the **Sketch Menu > Include Library > Manage Libraries**
- Search for "**adafruit neopixel**"
- Select the **Adafruit NeoPixel** library and install it
- Finally, we need to include the library so we can use it in the program.
- Include the library by going to the **Sketch Menu > Include Library, Adafruit NeoPixel**

Step 4

— Testing the Strip (M)

- Upload the programme shown here. All of the 5 lights should turn red, except the third one which will alternate green and blue.
- ① Can you see how this works? The block can be used in two ways - the first time it is used we set 'LED 0' - this is a short-hand way of saying to turn all of them the same colour
- ① The second time we use it, we only set 'LED 3' - this is the third one along
- ① Whenever we use this block, we must make sure the pin is set to '10' and the Qty to '5' (as there are 5 LEDs on the strip)
- ① Each of the five LEDs actually has three separate colours inside - a red, green and blue LED - and each of these can be set to a value between 0 and 255 (fully on) - with these three 'subpixels' we can make any colours that the eye can see!

Arduino Program

```
ledStrip pin 10 qty 5 LED 0 R 255 G 0 B 0
forever
  ledStrip pin 10 qty 5 LED 3 R 0 G 255 B 0
  wait 1 secs
  ledStrip pin 10 qty 5 LED 3 R 0 G 0 B 255
  wait 1 secs
```

Step 5

— Testing the strip (A)

- Try the programme shown. It should set the first three pixels to red, green and blue respectively.
- ① The 'strip.Color' code snippet takes three parameters between 0 (off) and 255 (fully on) for the red, green and blue channels respectively.

```
#include <Adafruit_NeoPixel.h>
Adafruit_NeoPixel strip = Adafruit_NeoPixel(5, 10, NEO_GRB + NEO_KHZ800);

void setup() {
  strip.begin();
  strip.setPixelColor(0, strip.Color(255, 0, 0));
  strip.setPixelColor(1, strip.Color(0, 255, 0));
  strip.setPixelColor(2, strip.Color(0, 0, 255));
  strip.show();
}

void loop() {
}
```

Step 6

— F1 Traffic Light

- Have a look at this video. It shows how formula 1 traffic lights work. They turn on one after the other and then the race starts.
- The lights are normally vertical, with the lights starting from the top.



— F1 Challenge

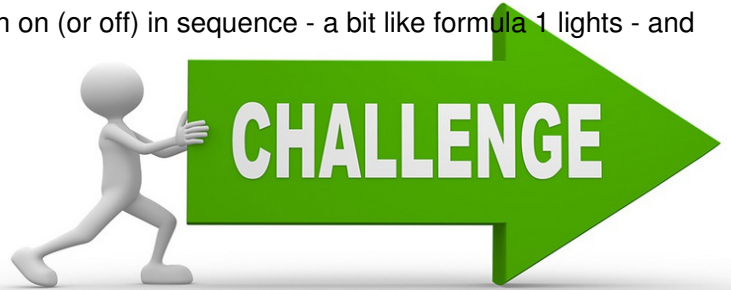
- Can you replicate these lights yourself using your LED strip?
- You should also add a buzzer that turns on after all the lights are on to give an audible indication the race has started.



Step 8

— Reaction Timer V2.0

- Do you remember how we made our reaction timer? Now repeat this concept by using your LED strip!
- ① You can decide how it works - you can have the lights turn on (or off) in sequence - a bit like formula 1 lights - and perhaps stop when you press the button!



Step 9

— Mood light (M)

- Try out the following programme
- You'll need to use the 'Make a Variable' command from the 'Data and Blocks' menu
- ① Can you see how it works? Every time through the loop we add different amounts on to the red, green and blue variables. If we added the same amounts on each time, then we wouldn't get any nice colours as when these three channels are set the same, we would just see a neutral (white) colour.



Step 10

Step 10

— Mood light (A)

- Try out the following programme.

- ① Can you see how it works? Every time through the loop we add different amounts on to the red, green and blue variables. If we added the same amounts on each time, then we wouldn't get any nice colours as when these three channels are set the same, we would just see a neutral (white) colour.
- ① The commands at the top such as 'int p;' define variables. Variables are like empty boxes that you can put in any number you want. Here we use 'p' for the pixel number (1-5) and r, g, b for the amounts of red, green and blue for the LEDs.
- ① The 'for' statement is a loop - it runs through every value of p from 0 to 4, incrementing by one each time ('p++').

```
#include <Adafruit_NeoPixel.h>

Adafruit_NeoPixel strip = Adafruit_NeoPixel(5, 10, NEO_GRB + NEO_KHZ800);

int p;
int r, g, b;

void setup() {
  pinMode(10, OUTPUT);
  r=0;
  g=0;
  b=0;
}

void loop() {
  for(p=0; p<=4; p++)
    strip.setPixelColor(p, strip.Color(r, g, b));
  delay(10);
  r=r+1;
  g=g+1;
  b=b+1;
  strip.show();
}
```

Step 11

— Improved mood light

- Can you make a better mood light?

- ① Ideally the colours would be more vibrant, they would change more slowly, and there shouldn't be sudden 'jumps' in the colours. These jumps are caused by the variables overflowing past 255
- ① Hint: You'll probably need to use some if statements to check if the variables are going to go over 255 (or under 0 if you are decrementing them)
- ① Hint: you might like to use a random number generator block/command to select either the colour channel you are changing (or the amount you are changing the colour channel by).

