# Smart Cooling Fan with Arduino to combat heatwaves

The UK spends most of the year in cold or moderate temperature so it's quite a shock to get sudden increases of temperature. Recently there's been frequent heatwaves in the UK and since it used to be quite uncommon, most people are unprepared for it and to make matters worse most British homes are designed to keep heat in. My partner and I's aim when doing this project was to design a fan that automatically turns on as it senses your presence and the light levels in its surroundings. This would be using a motion sensor and a photo-resistor.

We approached this problem by firstly connecting a button to LED lights to test whether we had a correct button-pushing system, then replaced the LED with a motor so it could power the blades of the fan. Once we had this working correctly, we connected both an LED and a motor to the button so both the motor and the LED would simultaneously turn on for the duration that the button is pushed. However, it wasn't possible for us to reach our aim of the project due to its complexity and time limitations so the final product consisted of a working button connected to a motor to power the blades of the fan and a LED to signal that the fan is on.

Similar solutions to this problem:
For portable fans, users have the choice between multiple fan types. Tower Fans are becoming one of the most popular styles of portable fans due to their design being space-saving which makes them a great option for small and tight spaces. Their design is also very aesthetically pleasing. Floor Fans are the most common type of fan and due to their design, they can be placed in front of areas with high air flow like windows to promote air circulation through a room. However, a disadvantage is that they can be quite noisy. Pedestal Fans often come with multiple settings and are sometimes programmable for when you sleep. This makes them quite versatile but the disadvantage is that they're quite bulky. Tabletop Fans are extremely common in home-settings because of their sturdy bases, several speed settings and the fact that they're relatively quiet and are often programmable. Lastly, the most similar to my fan design is Personal Fans. Personal Fans are the smallest and most portable of fans and are usually battery operated and often rechargeable. What makes them attractive to many users is their inexpensive price tag.

The similar solutions to this problem, such as oscillating tower fans and table fans, have the advantage of being larger which allows for more opportunity to be more customisable to the user. This could include having a bright, programmable LED display or having an easily removable grille for quick cleansing. An example of a fan like this is the Dylan Cool Tower Fan which has these features and has no blades so it gives an uninterrupted stream of airflow and it can be cleaned easily. In addition to this, having a larger fan means it can have bigger batteries which would last more hours of continuous operation than a smaller handheld portable fan.

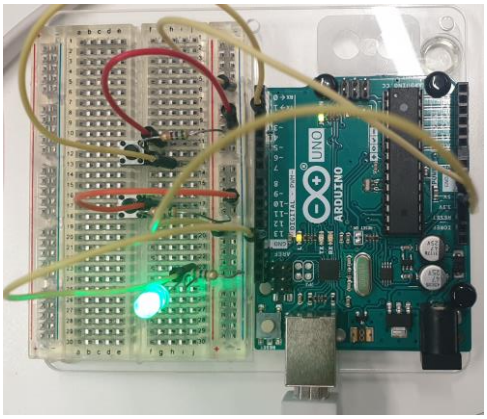
The design idea:
However, my design has an extreme benefit of being compact, portable and has the ability to be handheld. This means it can be stored in small areas or tight spaces and still be able to provide cooling airflow. In addition to this, it's more energy efficient due to its smaller size. Another added benefit is that my design can push cool air to spots that larger fans might not be able to reach and because it's compact and inexpensive, a user could have multiple of the portable fans just for that purpose.

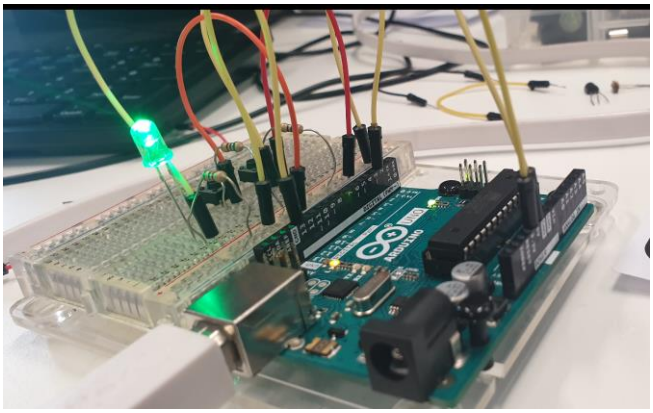
What differentiates it from typical personal fans is the fact that it would automatically turn on in the presence of the user which would be more convenient to the user, especially if they have multiple of these fans set up, since they wouldn't have to spend time turning on each individual fan.

Prototype:

We used built the prototype with 4 steps in order to differentiate between what was working and what wasn't when building the prototype and making the code as well as to ensure that there's room to add more features to the fan if we had spare time once completing the prototype.

- **Step 1 (using a push-button to activate a LED):**

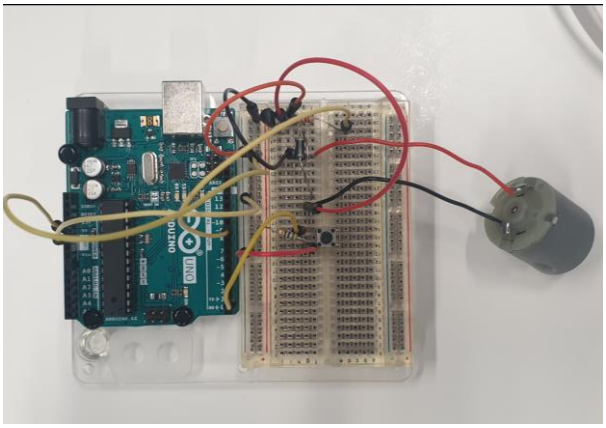


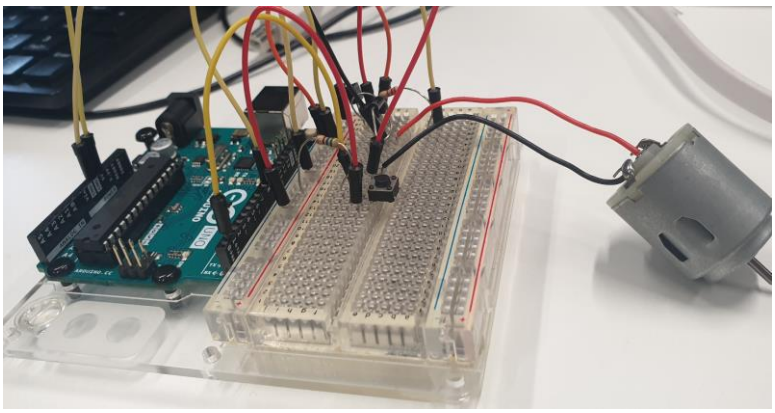


| **Components Used:** |
|---|
| Arduino Board |
| Push button |
| 10K Ohm Resistor |
| Hook up wires |
| Breadboard |
| LED |

Using a built-in Arduino example, we built a model where when the push-button is pressed, the LED is activated.
The code we used at this stage initially sets the PIN numbers of the buttons and the LED and sets a variable for reading the push-button status. Then, the LED pin is initialised as an output and the push-button pin is initialised as an input. Lastly, we have a loop where the state of the push-button is read: if the push-button is pressed, the button State is high and so turn on the LED and if not, turn the LED off.

- **Step 2 (using a push-button to activate a motor):**



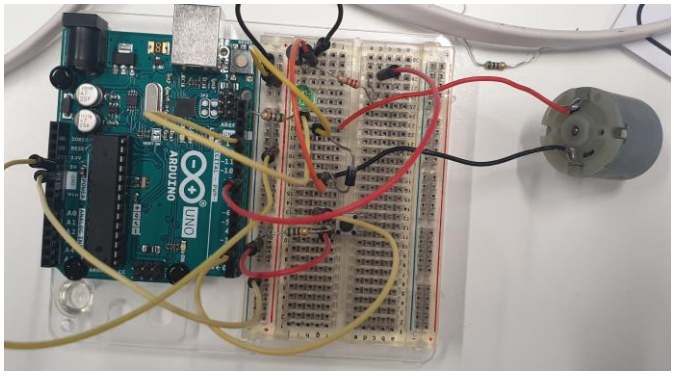


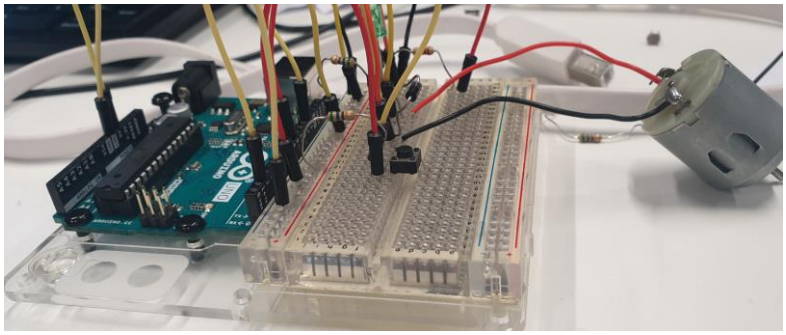
| **Components Used:** |
|---|
| Arduino Board |

| |
|---|
| Push button |
| 10K Ohm Resistor |
| Hook up wires |
| Breadboard |
| Transistor |
| Diode |
| Motor |

The code differs at this stage to account for the addition of the motor and the removal of the LED. In addition to this, using the code we determined the speed of the motor.

- **Step 3 (using push-button to activate both the LED and the motor):**





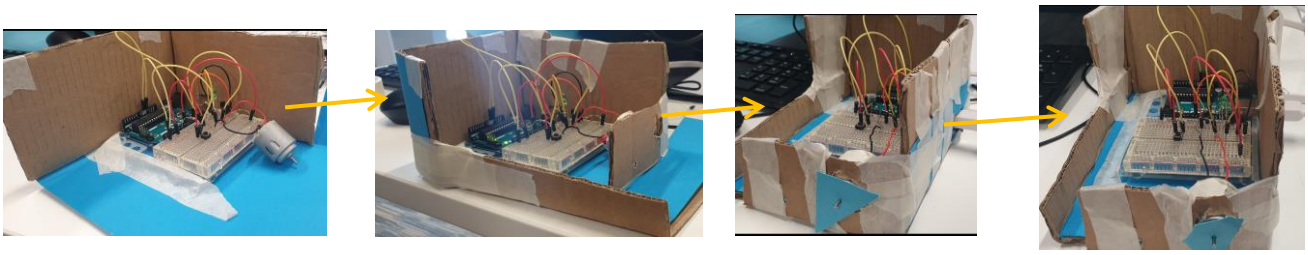
| **Components Used:** |
|---|
| Arduino Board |
| Push button |
| 10K Ohm Resistor |
| Hook up wires |
| Breadboard |
| Transistor |
| Diode |
| LED |

Here, the code implements elements from both Step 1 and Step 2 which allowed us to effectively simultaneously activate both the motor and LED with the push-button:

```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
const int motorPin = 9;

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialise the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialise the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
```

```
  //initialise the motor pin as an output:
  pinMode(motorPin, OUTPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == LOW) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    analogWrite(motorPin, 200);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    analogWrite(motorPin, 0);
  }
}
void motorOnThenOffWithSpeed(){

  int onSpeed = 200;  // a number between 0 (stopped) and 255 (full speed)
  int onTime = 2500;  //the number of milliseconds for the motor to turn on for

  int offSpeed = 50;  // a number between 0 (stopped) and 255 (full speed)
  int offTime = 1000; //the number of milliseconds for the motor to turn off for

  analogWrite(motorPin, onSpeed);   // turns the motor On
  delay(onTime);                    // waits for onTime milliseconds
  analogWrite(motorPin, offSpeed);  // turns the motor Off
  delay(offTime);                   // waits for offTime milliseconds
}

/*
 * motorAcceleration() - accelerates the motor to full speed then
 * back down to zero
*/
void motorAcceleration(){
  int delayTime = 50; //milliseconds between each speed step

  //Accelerates the motor
  for(int i = 0; i < 256; i++){ //goes through each speed from 0 to 255
    analogWrite(motorPin, i);   //sets the new speed
    delay(delayTime);           // waits for delayTime milliseconds
  }

  //Decelerates the motor
  for(int i = 255; i >= 0; i--){ //goes through each speed from 255 to 0
    analogWrite(motorPin, i);   //sets the new speed
    delay(delayTime);           // waits for delayTime milliseconds
  }
}
```

- **Step 4 (assembling the fan):**



When assembling the fan we taped the Arduino and breadboard to a cardboard base to ensure it won't move when the fan is switched on. We also boxed the whole fan with cardboard and left the top open to not only allow the user to access the button but also to allow for a stream of air to pass through from the top of the fan and come out from behind the blades of the fan in order to power the blades of the fan more.

Testing:
When testing the fan on users, our main feedback was that although the fan had a good cooling effect, the awkward placement of the button used to turn on the fan reduced the quality of the user's experience.

Future work:
In a future execution of this project, I would add photo resistor to detect light levels so the fan can turn on automatically at certain times of day. In order to improve upon the work from this project, I would add a WiFi Shield to the Arduino so it can connect to selected devices and deliver information to their health apps. In addition to this, I would implement solar panels to the fan as a way to extend the life of the fan by slowly recharging the original battery. I believe this would be especially beneficial to users if they were to use a fan outdoors. This is supported by a study which claims "solar cells cannot currently be a feasible power source for normal operation of handheld computers. However, from both modelling and experiment results, we learn that solar cells can be a partially supportive power source for the extension of the "standby time" of handheld computers." As said by Bai and Chen in 2005. In conjunction with this, I believe simply adding modes for different fan speeds will increase the quality of the user's experience since it would make the fan more customisable to them.

In conclusion, despite not fully meeting the aims of the project, the final product reached the efficiency I desired and the fan is able to be developed further. When completing this project, I learnt the importance of breaking down a problem into steps before tackling it. I saw a real benefit of building the prototype for this cooling fan in steps as it allowed for my partner and I to add more features to the fan depending on the time we had left. Furthermore, I believe implementing the features I mentioned in my future work would attract more users in the UK who are struggling with the constantly changing weather and would act as a unique solution to consumers for the heatwaves in the UK.

## References


- SM 2018, *Button Built-in Example*, viewed 17th August 2021, <https://www.arduino.cc/en/Tutorial/BuiltInExamples/Button>.
- Conan, L 2021, *Best Portable Fans: Stay Cool and Comfortable*, viewed 20th August 2021, < https://www.thedrive.com/reviews/35362/portable-fan-2>.
- Bai, Y, Chen, C 2005, *Using Solar Cells to Extend Standby Time of Handheld Computers*, viewed 19th August 2021, <https://www.researchgate.net/publication/245516848_Using_Solar_Cells_to_Extend_Standby_Time_of_Handheld_Computers>.