

Popup book play mp3 on page turn

Suggested parts:

- Arduino Nano
- DF Robot mp3 player (See https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299 for docs)
- mini speaker
- 1K resistors x 3
- 10K resistors x number of pages in the book
- Breadboard and jumper wires for prototyping
- 3mm neodymium magnets: 2 per page

Sensing the active page:

In most popup books, each page is constructed from two layers or thin card (or thick paper) glued together. We can sandwich the circuitry for each page in between these two layers. We'll use copper tape running close to the spine. Make sure to leave enough gap for binding (Fig 1).

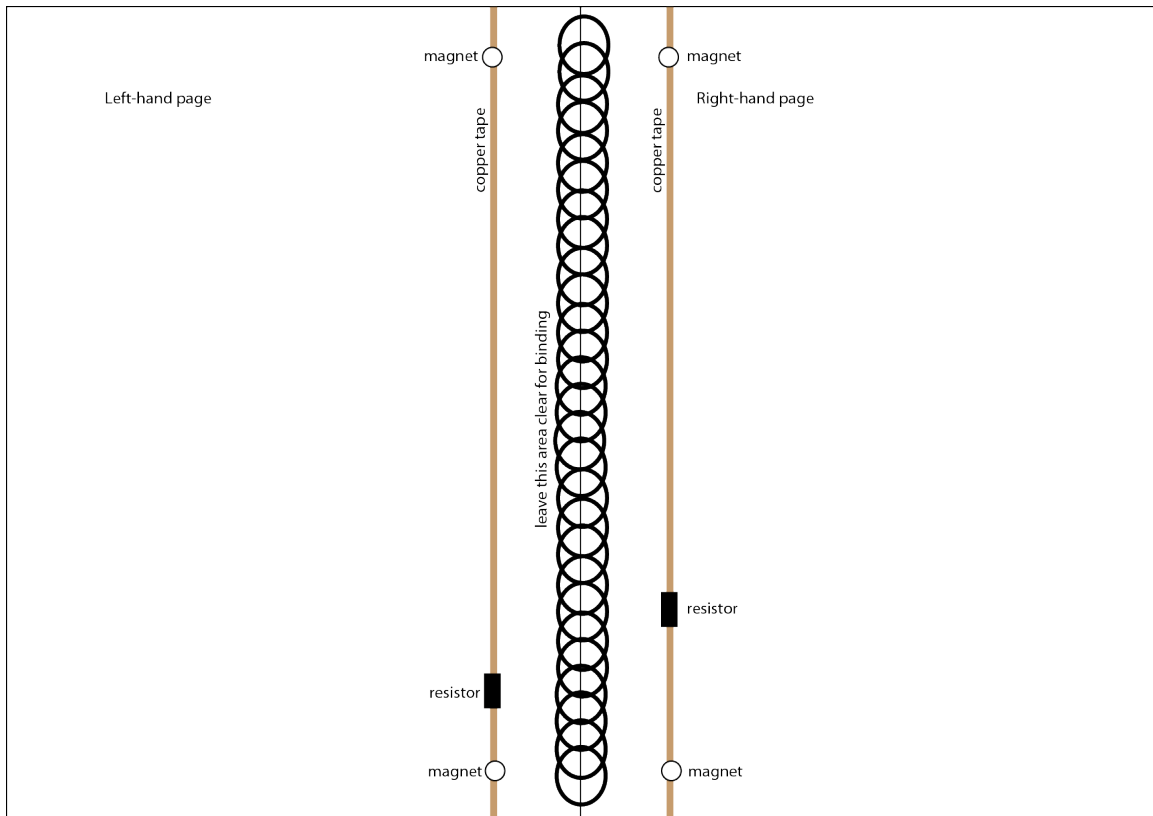


Fig 1. Page layout

First, tape 3mm neodymium magnets on the two inner corners between the layers to keep the pages snapped tightly together (Fig 2). This will ensure a good electrical connection between the copper tape tabs on the edges of the pages even if the pages are lightweight - the magnets keep the popup parts secured (otherwise they push the pages apart when they are not folded down all the way). (You could use insulated wire instead of copper tape if you prefer).

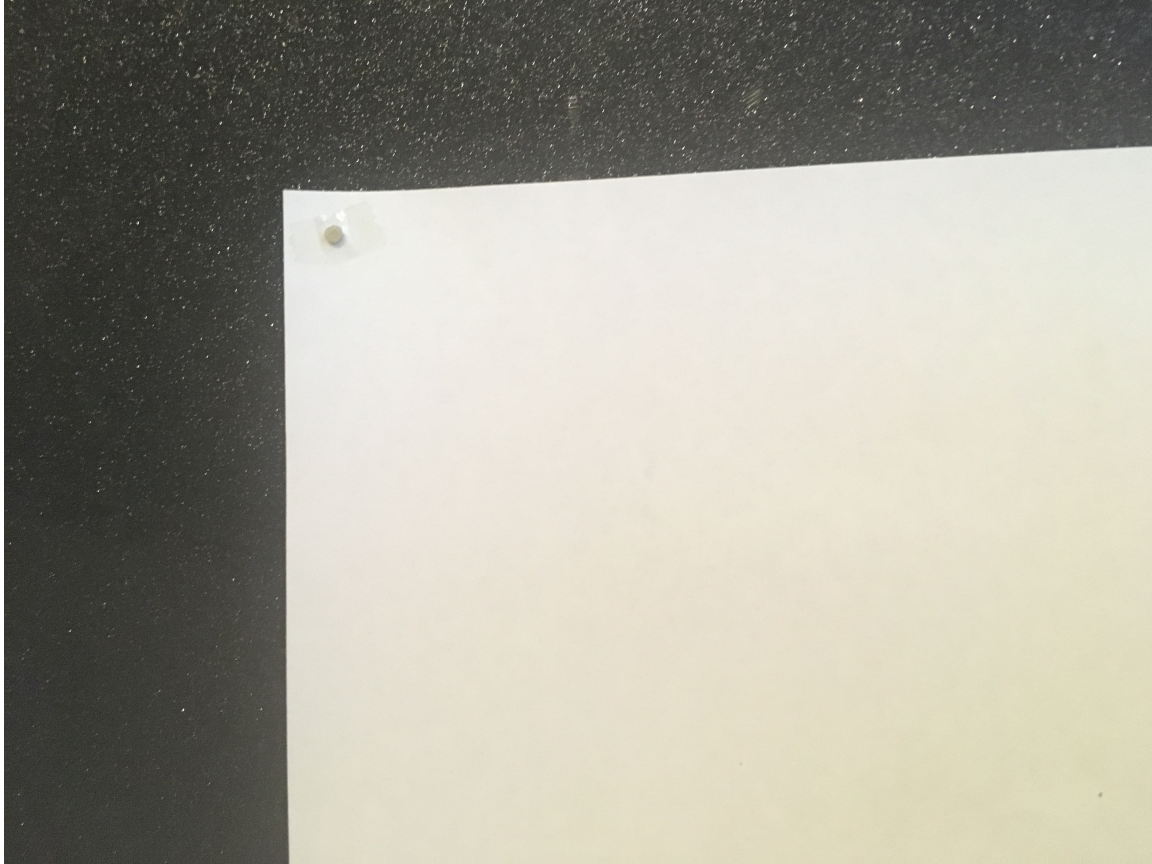


Fig 2: Tape magnet on back of page corner

Use conductive tape (e.g. 5mm copper tape) to run along the edge of each page, with a small gap, where the resistor will be connected inline. This could be an SMD resistor glued in place using conductive glue for least obvious components, however this is fiddly and prone to bridging the resistor if you slip or use too much glue, so soldering a through-hole resistor to the copper tape would likely be more reliable and if the card is thick enough should not be very noticeable. A small amount of the copper tape folds over onto the front at the top and bottom of the page so that the pages are all connected (the resistors for all of the pages are connected in parallel) (Fig 3). Run the tape along the entire length of the page (Fig 4) leaving a small gap for the resistor (Fig 5).

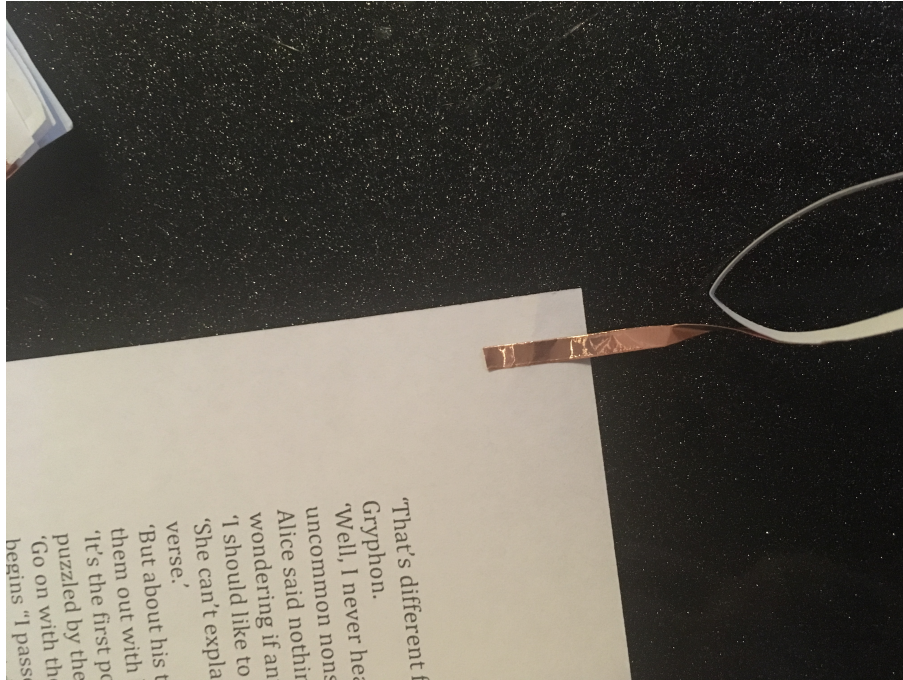


Fig 3. Copper tape folded over in front of the magnet on the front of the page

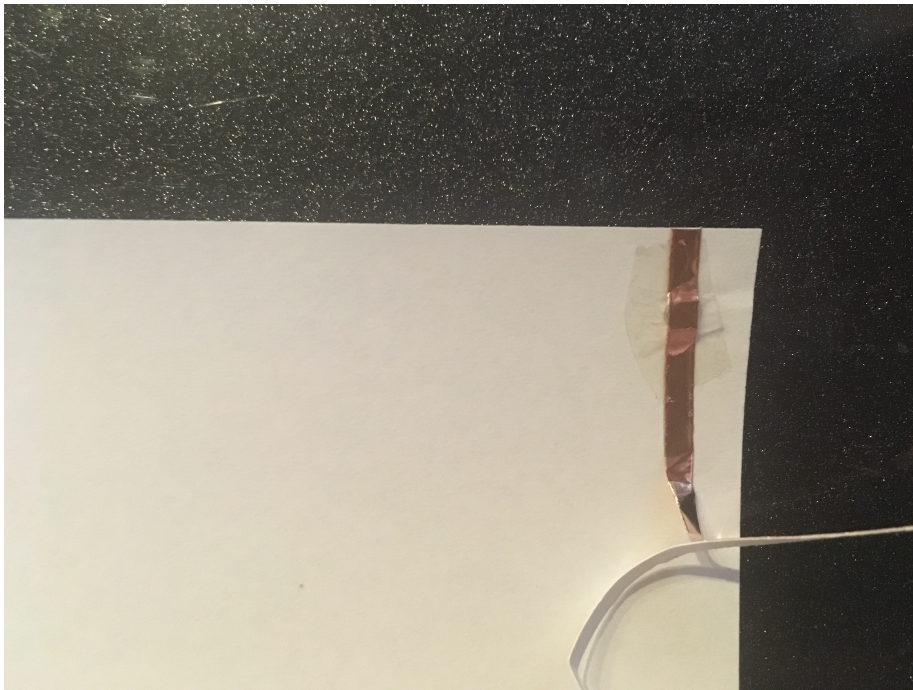


Fig 4. Copper tape running along the back of the page (this will be sandwiched between the card).

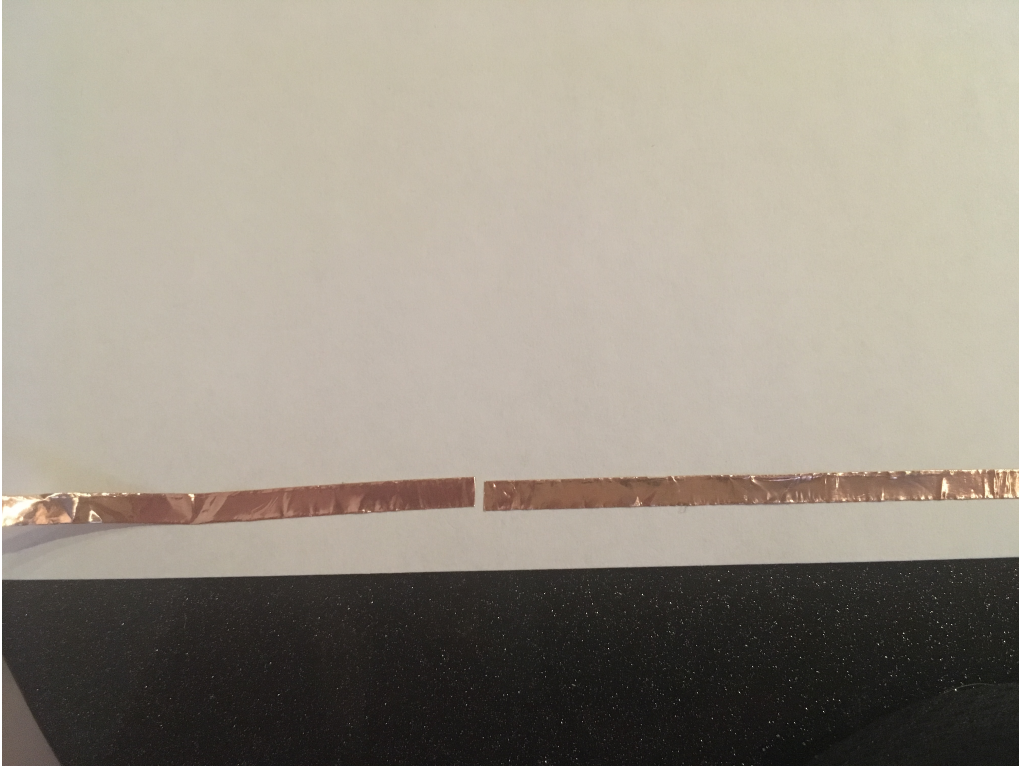


Fig 5. Leave a gap for the resistor

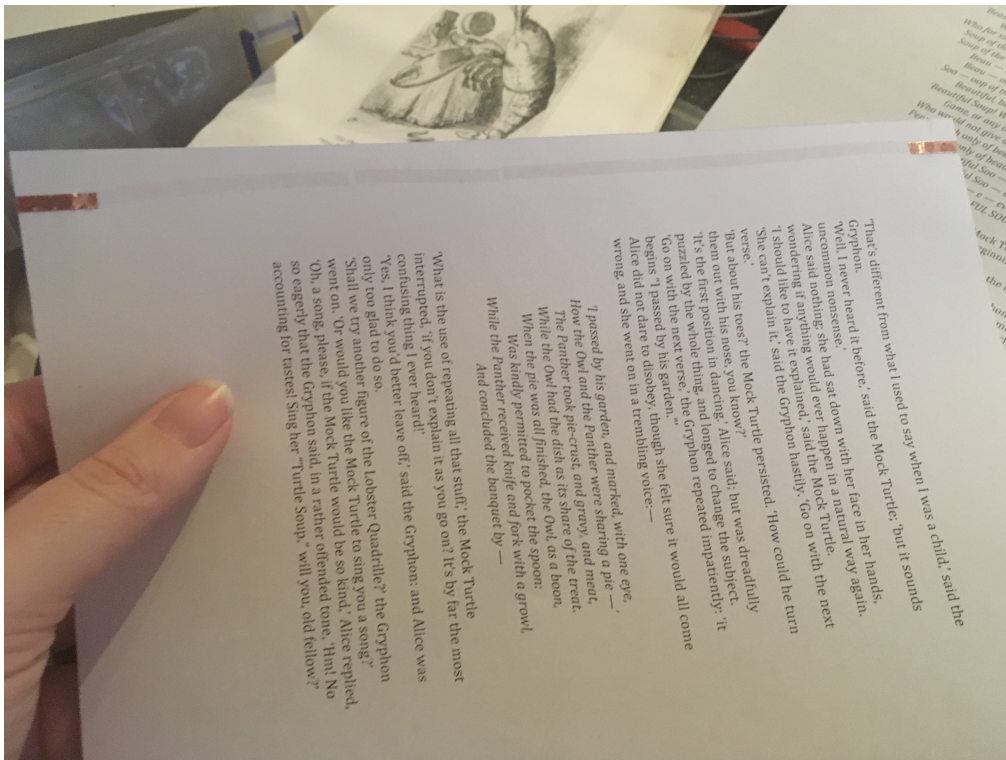


Fig 6. Copper tape runs from front top of magnet, along the back of the page (with resistor) and folds over again onto the front at

the bottom to cover the magnet

Stagger the placement of the resistors across all the pages to avoid bulking up the pages in one area. Solder or glue (with conductive glue) a resistor in place to bridge the tap in the tape (Fig 7). The page resistors can be any value but the maths is easier if you stick to the same value for each page. For example, use 10K resistors.

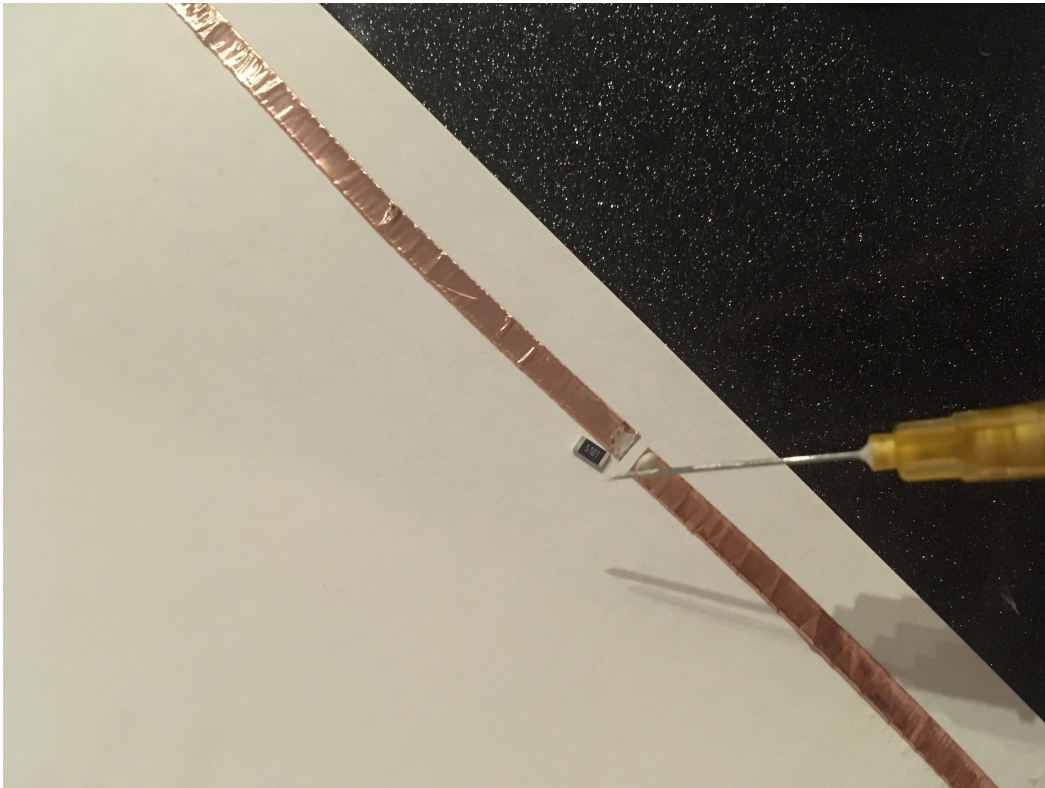


Fig 7. Attach resistor

On the other layer of the page, add copper tape tabs to the top and bottom of the page (just enough to fold over to cover where the magnet goes (Fig 8). This copper tape joins the pages into a single circuit when they are stacked together.

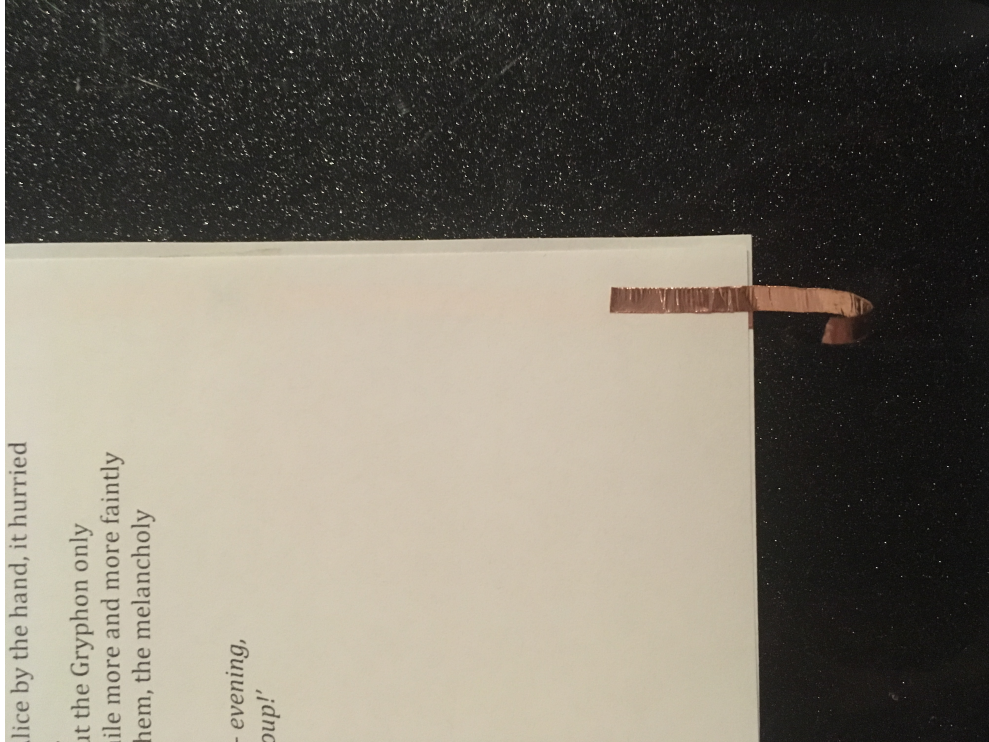


Fig 8. Fold small amount of copper tape over top and bottom of page

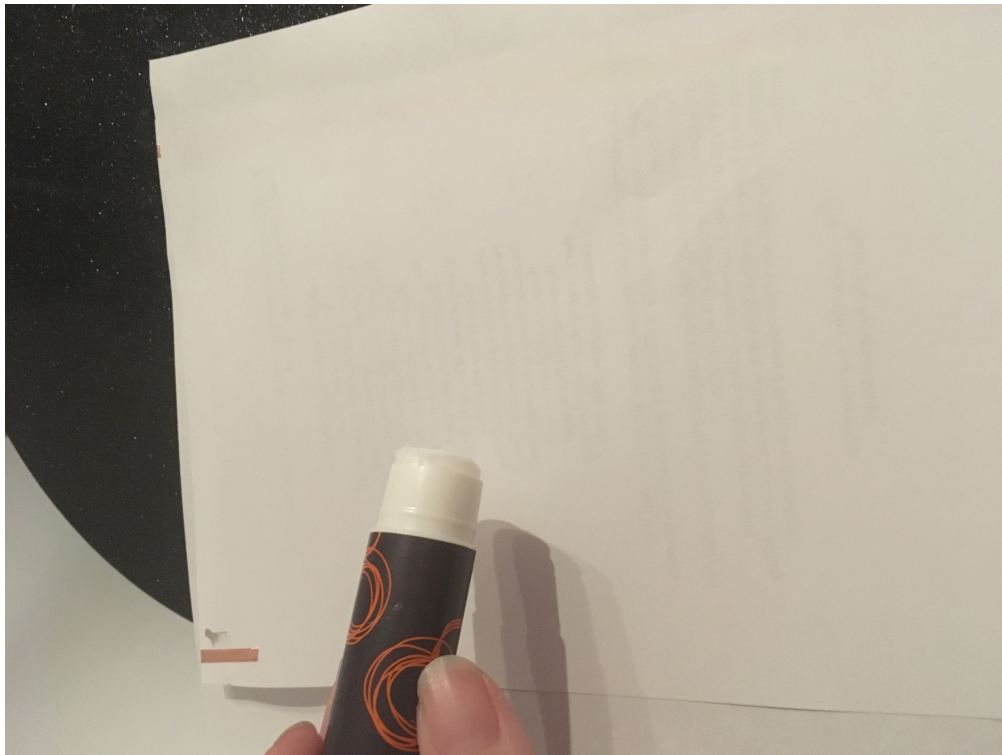


Fig 9. Glue both sides of the page together.

Glue the pages together (Fig 8), making sure that the copper tape is lined up so that the front and back sides of the page circuit line up (Fig 9).

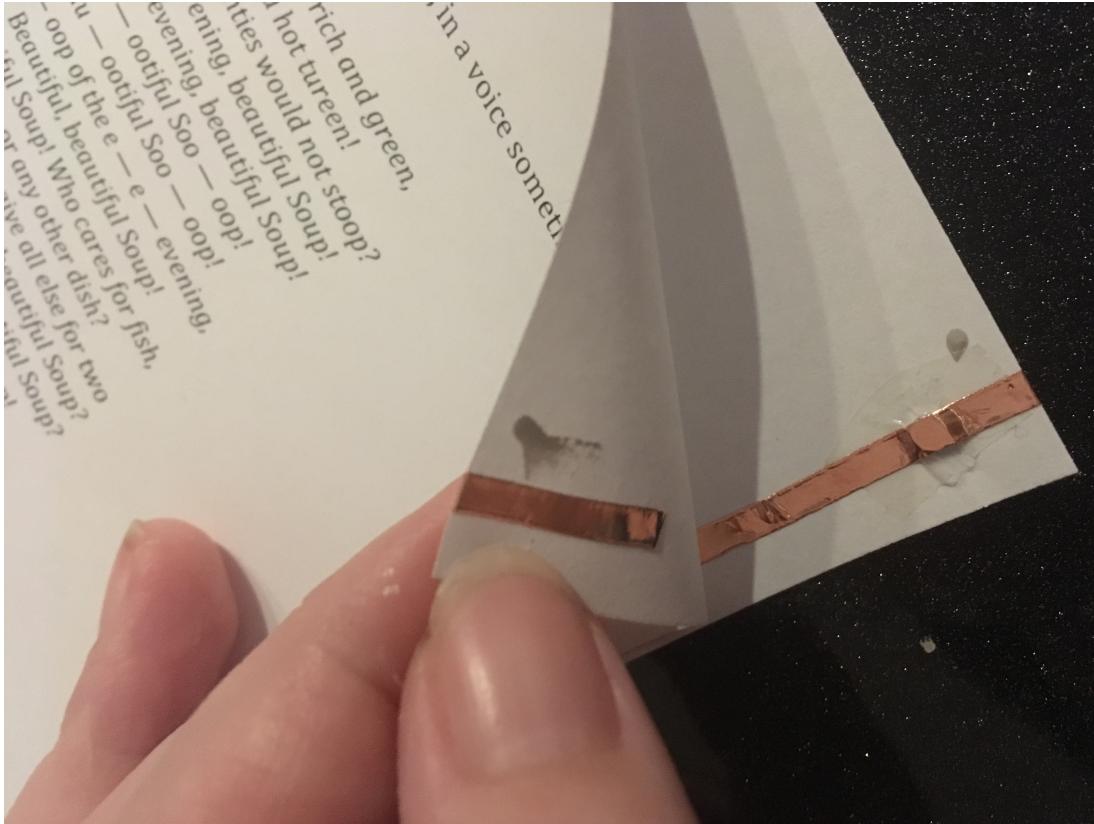


Fig 10. Tape on back side only needs to reach to the magnet (to ensure a good connection).

To calculate the total resistance of resistors in parallel:

$$R_{\text{total}} = 1/(1/R_1 + 1/R_2 + 1/R_3 \dots + 1/R_n)$$

e.g. book with 3 pages, using a 10K resistor per page, the resistance read on the analog pin when each page is open is as follows:

page 1: 3.3K

page 2: 5K

page 3: 10K

Book with 8 pages:

page 1: 1.25K

page 2: 1.43K

page 3: 1.67K

page 4: 2K

page 5: 2.5K

page 6: 3.3K

page 7: 5K

page 8: 10K

Book with 12 pages:

page 1: 833K

page 2: 909K

page 3: 1K

page 4: 1.1K

page 5: 1.25K

page 6: 1.43K

page 7: 1.67K

page 8: 2K

page 9: 2.5K

page 10: 3.3K

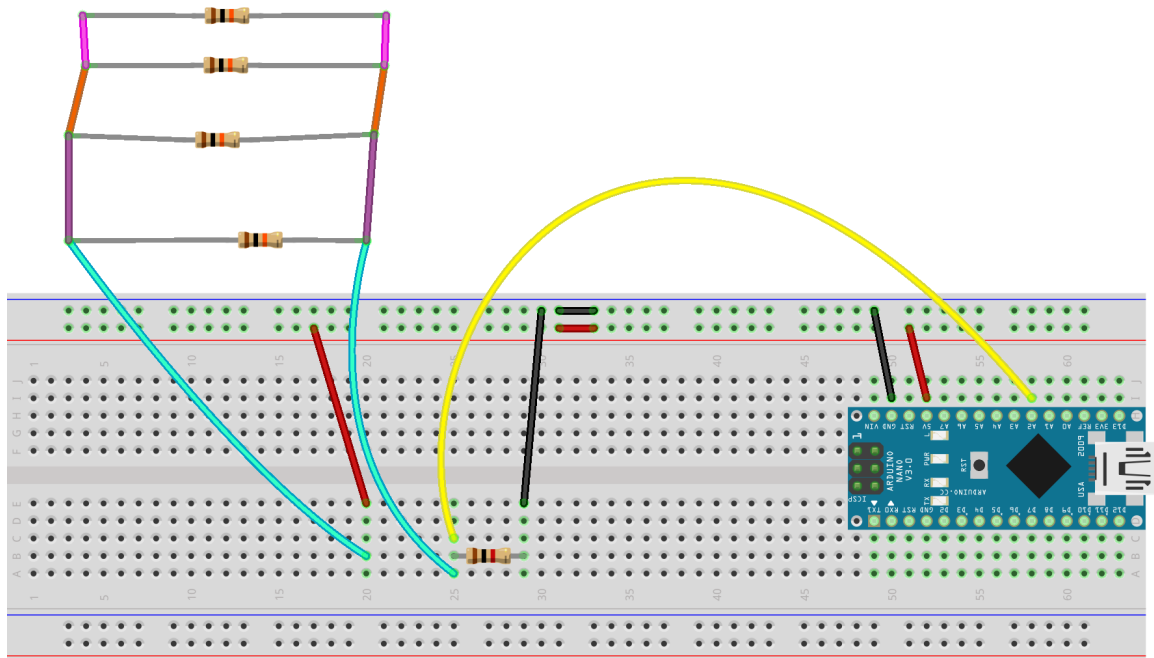
page 11: 5K

page 12: 10K

There is always some degree of variation in the resistance detected by the Arduino, so if you are planning on making books

with more than 12 pages, use a larger resistor (e.g. 100K) to allow more gap between the valid resistance ranges for the pages at the start of the book.

The pages are connected up to an analog pin (e.g. A2) with a voltage divider with another resistor of known resistance (in the circuit shown in Fig 10 this is a 1K resistor towards the center of the breadboard). The resistors connected in the top left of the diagram represent the pages – using 10K resistors for each page. We can convert the raw sensor reading into a resistance in ohms to work out which page is open at any given time.



fritzing

Fig 10. Pages circuit

A basic Arduino sketch for reading in the resistance and printing the calculated value is useful while building each page to check that the resistance being read correctly:

```
int analogPin = A2; // analog pin you have the page circuits on
int numPages = 8;

// these values are for up to 8 pages, 10K resistors on each page
int thresholds[] = {1250, 1429, 1666, 2000, 2500, 3333, 5000, 10000}; //
thresholds for each page
```

```

int rawValue = 0;
int vin = 5;
float vout = 0;
float r1 = 1000; // Known value resistor in the voltage divider (1K)

float currentPageResistance = 0; // the resistance we are reading from the
combined pages
float buff= 0;

int pageVal; // the current page

void setup() {
  Serial.begin(9600);
}

void loop() {
  // read the raw value from the analog pin
  rawValue = analogRead(analogPin);

  if (rawValue) {
    // convert it to a resistance value in ohms
    buff = rawValue * vin;
    vout = (buff)/1024.0;
    buff = (vin/vout) -1;
    currentPageResistance = r1 * buff;

    // calculate current page
    for (int p = 0; p < numPages; p++) {
      if (currentPageResistance >= thresholds[p]) {
        pageVal = p + 1;
      }
    }

    Serial.print("current page: ");
    Serial.println(pageVal);
    Serial.print("Resistance: ");
    Serial.println(currentPageResistance);
  }
  delay(1000);
}

```

Open Tools > Serial Monitor in Arduino IDE to see the resistance values (Fig 11):



Fig 11. Serial monitor

Prepare the audio files:

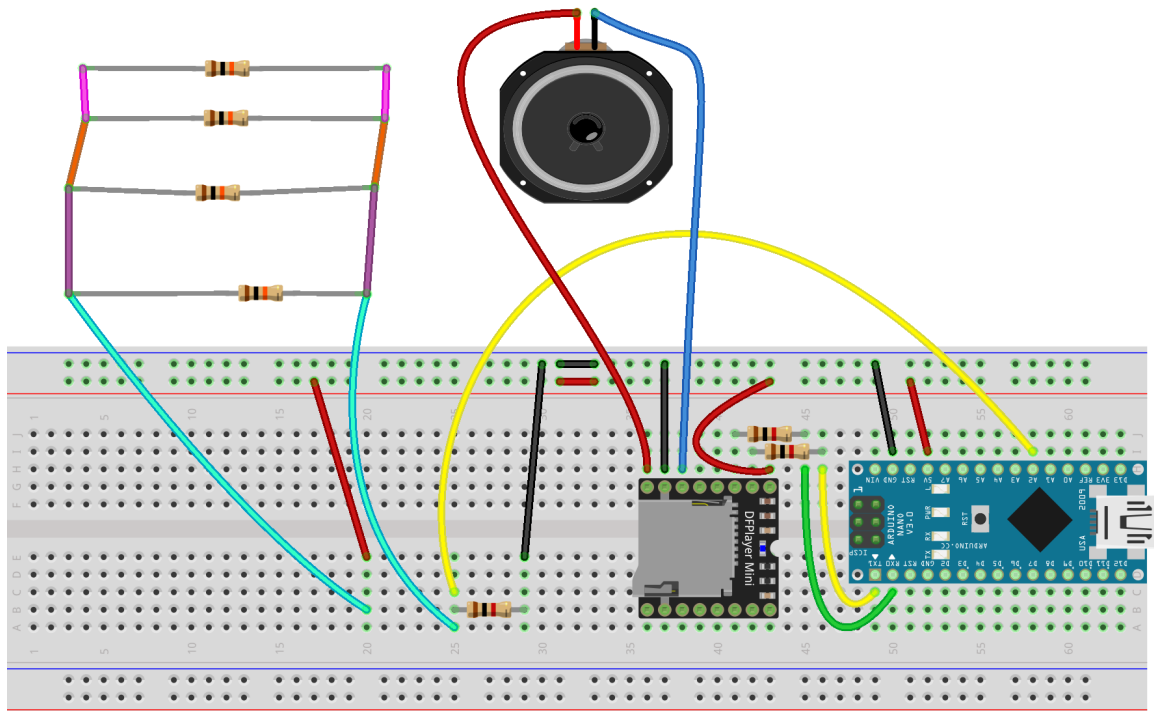
Create an 'mp3' folder on the microSD card. Copy one mp3 for each page into that folder. The filenames must begin with a 4 digit number but can have any title after that e.g.

0001FirstPage.mp3
0002music_for_page2.mp3
0003excellent-third-page.mp3

Using the DFRobot Mp3 player:

Install the DF Robot mp3 player library from:
(unzip and copy into ~/Documents/Arduino/libraries)
https://github.com/Arduinolib/DFRobot_Mini_Player/raw/master/DFPlayer_Mini_mp3.zip

Fig 12 shows the circuit with the mp3 player module and speaker added:



fritzing

Fig 12. Mp3 player circuit

To detect when the page is turned for triggering mp3 playback, we can keep track of the previous active page, and poll the analog pin every so often, and compare the current active page with the previous value - if it is different than the previous value then we know that the page has been turned and can trigger the mp3 for that page to play.

```
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h>

int analogPin = A2; // analog pin you have the page circuits on
int numPages = 8;

// these values are for up to 8 pages, 10K resistors on each page
int thresholds[] = {1250, 1429, 1666, 2000, 2500, 3333, 5000, 10000}; //
thresholds for each page

int rawValue = 0;
int vin = 5;
float vout = 0;
float r1 = 1000; // Known value resistor in the voltage divider (1K)

float currentPageResistance = 0; // the resistance we are reading from the
```

```

combined pages
float buff= 0;

int pageVal = 0; // the current page
int prevPageVal;

void setup () {
  Serial.begin (9600);

  mp3_set_serial (Serial);
  delay(1);
  mp3_set_volume (25);          // value 0~30
}

void loop () {
  // read the raw value from the analog pin
  rawValue = analogRead(analogPin);

  if (rawValue) {
    // convert it to a resistance value in ohms
    buff = rawValue * vin;
    vout = (buff)/1024.0;
    buff = (vin/vout) -1;
    currentPageResistance = r1 * buff;

    // calculate current page
    for (int p = 0; p < numPages; p++) {
      if (currentPageResistance >= thresholds[p]) {
        pageVal = p + 1;
      }
    }
    if (currentPageResistance < 1) {
      pageVal = 0;
    }
  } else {
    pageVal = 0;
  }
  if (pageVal != 0 && prevPageVal != pageVal){
    prevPageVal = pageVal;
    mp3_play(pageVal);
  } else if (pageVal == 0) {
    mp3_stop();
  }

  delay(100);
}

```

Operations supported by the mp3 player:

```

mp3_play (); //start play
mp3_play (5); //play "mp3/0005.mp3"
mp3_next (); //play next

```

```
mp3_prev (); //play previous
mp3_set_volume (uint16_t volume); //0~30
mp3_set_EQ (); //0~5
mp3_pause ();
mp3_stop ();
void mp3_get_state (); //send get state command
void mp3_get_volume ();
void mp3_get_u_sum ();
void mp3_get_tf_sum ();
void mp3_get_flash_sum ();
void mp3_get_tf_current ();
void mp3_get_u_current ();
void mp3_get_flash_current ();
void mp3_single_loop (boolean state); //set single loop
void mp3_DAC (boolean state);
void mp3_random_play ();
```