

Mp3Flower is an MP3 player for small children from 2 to 5 years old.

I present my work in a little details so that it can be useful to you.

My son Quentin wanted me to make an MP3 player for his little boy Tom who is 2 and a half years old. It involves making available a large number of MP3s containing nursery rhymes, but also stories, music ...

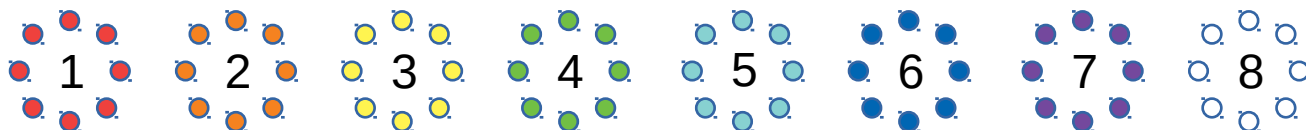
The devices that I saw on the internet did not satisfy me. Indeed, the large number of MP3s does not allow the use of RFID tags which would eventually get lost. A keyboard with a large number of keys did not seem suitable either.

For display, the ideal is a graphic display with the name of MP3 ... But Tom will not be able to read for 5 or 6 years.

For ergonomics I made the following choices: a button to turn the device on and off. With automatic shutdown if no MP3 has been played for 5 minutes. So he can fall asleep while listening to a story, the device will turn off automatically. A button to choose the color (the directory) and a button to choose the MP3 in the directory. Finally a button to listen to or stop the MP3. And then a rotary knob for the volume (limited by the parents).

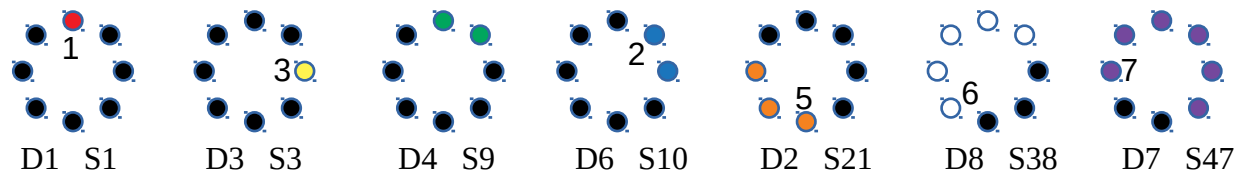
My idea was to make a display with a ring of multicolored LEDs. The color and position of the lit LEDs allow you to know which MP3 will be played.

As I had on hand a ring of 8 LEDs, I chose to distribute its MP3s in 8 categories each associated with a clearly identifiable color.



1 Nursery rhymes, 2 music, 3 sounds of nature, 4 stories read by her grandmother, 5 stories to learn, 6 stories to learn, 7 stories to sleep and 8 Grandpa to the piano. Obviously these choices are modifiable.

Then in each category (or directory) the LEDs displayed inform of the MP3 which is selected by the right button on the front of the device. Obviously this solution does not allow to have too many MP3 per directory. Theoretically 256, but Tom does not yet know binary. I limited myself to 55 MP3 per directory as follows:



D_x for the directory number and S_y for the number of the MP3 to be played. The number in the ring is the number of the first (clockwise) LED on.

The number of the song S in the directory D is $S = N \circ \text{FirstLed} + 8 \times (\text{Number of Leds On} - 1)$

But Tom will not do any calculation! For the first 8 MP3s in a directory, the display will be simple: a led = a MP3. It will already have access to $8 \times 8 = 64$ MP3. If the number of MP3s in a directory exceeds 8 but is less than 17, it will quickly understand the meaning of the fact that two LEDs are on. Parents will just have to be reasonable about the number of MP3s per repertoire and gradually increase it, if necessary, beyond 8, then 16, then 24 ...

-091.mp3 directory 1 home audio
...
-098.mp3 directory 8 home audio

12-001xxzz.mp3 The hidden directory 12 contains the mp3 numbered with 3 digits.
-002yyzzyy.mp3
- ...

A hole in the back of the case gives access to the SD card. Because of the thickness of the case, you have to have a pair of tweezers and handling is not so easy. That's fine because I'm sure Tom's not going to get the SD card

The case is made of wood and Quentin still has to paint it. So it is completely removable apart from the charger for the LiPo battery which is stuck in the case.

I had fun making the knob of the volume potentiometer, and the other buttons too, out of wood. That of the volume potentiometer is connected to the potentiometer via a transparent garden tube with an internal diameter of 6mm. For push buttons, they slide into a hole in the front and press on recovered keyboard switches. To prevent them from leaving the facade, on the inside, they are surrounded by a copper ring with an internal diameter of 10mm and a height of 3 to 4 mm.
The electronics are based on the DFPlayer an MP3 player incorporating serial communication, an SD card holder, an equalizer and a small audio amplifier. I was pleasantly surprised at the quality of the rendering as long as you use a suitable speaker and choose the EQ settings appropriately.

To facilitate disassembly, I chose not to stick the speaker. It is held in the following way: an 8mm nut is glued to the back of the speaker, a bolt completely screwed into the nut just allows the assembly to slip into the housing. Once in the case, the speaker is positioned against two small wooden wedges glued to the case. It is then enough to unscrew the bolt from 1 to 2 turns so that its head bears against the face of the housing opposite the diaphragm of the speaker. A lock nut ensures that the bolt remains in compression.

The volume is adjusted as follows: the position of the potentiometer is read from an analog input on the Arduino. The value obtained is converted in such a way that the sound volume remains less than or equal to that fixed by the parents. The DFPlayer accepts levels between 0 and 30. If the parents have set the maximum level at 18, the child can vary the level from 2 to 18.
The DFPlayer is controlled by a classic Arduino 5V 16Mhz. I had a mini pro on hand which I removed the power regulator and the power led.

The display uses a ring of 8 WS2812 LEDs. Between the ring and the front of the device, I interposed a white plastic disc, 3mm thick and translucent, to diffuse the light emitted by the LEDs. The effect is prettier and less dazzling. In order to compensate for the thickness of the facade and to have a greater visibility angle, the hole in the facade is bevelled.

The charger is of type TP4056, I had it on hand, with the load limiting resistor modified to limit the charging current to around 200mA. A hole on the back of the case allows a connection for the charger in micro USB.

The auto power off circuit is optional, we can settle for a toggle switch. The TP4056 and the LiPo battery can also be replaced by a coupler of 3 x 1V5 batteries.

If you remove the auto power off circuit, small simple changes are possible in the program, but are not required. During programming, the auto power off does not work because the Arduino is directly supplied with the programming serial port. I strongly advise you to disconnect the LiPo battery during the development and programming phases.

The purpose of the 1000uF capacitor is to avoid having to keep the power button pressed for too long to start up. Indeed without this capacitor, as it takes a few seconds for the Arduino to start, you must keep the power button pressed for at least 3 to 5 seconds.

The PNP 2N2905 could have been replaced by an MOS but I did not have one on hand. The voltage drop would have been smaller. It is around 0.8V with 2N2905.

The program could be more elegant. It works perfectly thanks (see below my thanks) to some authors of articles and blogs which allowed me not to waste too much time of development. I have tried to make it as clear as possible so that you can change it at will. The program has around 600 lines, including over a hundred comments.

The wiring is not very pretty, but it is a prototype ...

Conclusion: this prototype is perfectly functional, its ergonomics and sound qualities are very satisfactory. It remains to be seen if the object will be adopted by Tom. I will only have this answer in several weeks due to confinement. When Quentin has had time to paint the object, I will add a photo for the final rendering and clarify Tom's reactions to the object.

Details of the equipment used:

- 1 DFPlayer for MP3 playback
- 1 Mini Arduino Pro 5V 16Mhz or other classic
- 1 ring WS2812 of 8 leds for display
- 1 speaker 4 or 8 Ohms of not too bad quality from 8 to 12 cm in diameter.
- 1 piece of translucent white plastic 4 cm in diameter and 3 mm thick.
- 4 momentary push buttons P (Power), G, C and D.
- 1 potentiometer 47K for adjusting the volume

- 1 LiPo 3V7 battery with 600mAH capacity or more
- 1 TP4056 LiPo battery charger with charge limited to 200mA

- 1 capacitor of 470uF 10V smoothing
- 1 Resistance of 1K towards the RX of the DFPlayer
- 2 resistors of 220K divider for Vcc and potar
- 1 capacitor of 47nF 10V filtering at the potar output
- 1 resistor of 47K divider for Vcc

For auto power off (otherwise a rocker switch ...)

- 1 PNP transistor 2N2905 acts as a switch
- 1 NPN 2N2222 transistor drives the switch
- 1 type 4148 diode to logically isolate the power ON / OFF button P
- 1 1000uF 10V capacitor for ON and OFF storage
- 1 resistance of 220 Ohms limits the current in the capacitor and the push button
- 1 resistor of 470 Ohms for driving the base of the 2905
- 2 resistors of 1K for piloting the base of 2222

3 resistors of 10K to recall to "0"

Thanks :

Thanks to x1sc0 who deserves a good beer for his work on driving the WS2812.

<https://www.instructables.com/id/Bitbanging-step-by-step-Arduino-control-of-WS2811-/>

I didn't want to use a library to drive the WS2812.

Thanks to Angelo qiao for his library for DFPlayer.

<https://github.com/DFRobot/DFRobotDFPlayerMini>

Thanks also to Byfeel for their work on the DFPlayer.

<https://byfeel.info/un-lecteur-mp3-pour-votre-esp8266/>

Thanks to Quentin Brun, my son and dad of Tom who takes care of painting the camera.

Thanks to Nathalie Brun, my wife, for the repeated tests on the use and ergonomics of the device.

Thanks to Pierre Brun, my son who corrected my translation of this text in English.

What could be done:

Manage the activity of the volume potentiometer with a hysteresis between Potar and Potar8. I didn't use it.

The same applies to the measurement of the battery level between ValTens and NivBatterie.