# Celebrate.h

Text

Description automatically generated

The “danceMoves()” function will make the robot dance.

This function is used together with the “timer” function and therefore needs to return a boolean. This will determine if the timer needs to run this function again (which in this case is always false). We also need to use the “void\*” (any) type as a parameter, since the timer can pass an optional parameter along to the function, even if we don’t use a parameter (like we do here).

The dance itself takes ~4s, the song takes 14 seconds, so we will do the dance 4 times using a for loop.

“for(int i=0; i<=3 i++)” will run the code within the loop for 4 times.

This loop uses the following functions:

“changeLED()” defined in “LedControl.h”, it accepts the color we want to set as a parameter.

“drive()” defined in “Motor.h”, it accepts the direction, (optionally) the amount of time to go in this direction in ms, and the speed to go in the given direction.

“MOTOR\_SPEED\_SLOW” is defined at the top of “Motor.h” and is a pre-defined slower speed that is a perfect use for dancing.

These are the directions used for the “drive()” function:

Graphical user interface, text, application

Description automatically generated

1. We first turn the led blue.
2. We slowly drive forwards for 400ms
3. We stop all together
4. We slowly drive backwards for 400ms
5. We turn the led orange
6. We slowly drive to the left for 400ms
7. We slowly drive to the left for 400ms
8. We turn the led green
9. We slowly move forwards for 400ms
10. We slowly move backwards for 400ms
11. We turn the led purple
12. We slowly drive left for 1000ms
13. We turn the led red
14. We slowly drive right for 1000ms
15. We turn the led black (or “off”)

When the dance is done, we will stop the the robot all together using “drive(0)”.

Text

Description automatically generated

This function helps us to start the dance with a delay, this can help us to trigger the dance at a later time without blocking the rest of the code.

We use a parameter called “timeDelay” (in ms) which defaults to 0.

We trigger the “playMP3()” function defined in “MP3Player.h” to play the dance MP3 file (track 10). We pass the delay as a second parameter to this function to tell the playMP3() function that it has to play the MP3 file with a delay.

We use the “timer” library to trigger the “danceMoves()” function with the given delay. This is done with “timer.in(timeDelay, danceMoves)”. Note: this is a cheat way to run parallel actions. Since the Arduino is one core, we can only run one thing at a time. The timer.in() allows us to run the dancing, without interrupting the rest of the loop from executing.

In case the timeDelay is 0, it will simply play the MP3 file and start dancing directly.

Graphical user interface, text

Description automatically generated

This function is triggered from the main “loop()” function in case we won.

We set the global “hasWon” variable to true, so we wont be able to trigger this function untill the game is restarted.

We set the “voiceControlled” to false, this variable is used in the code to determine if we are currently playing the game (which is now no longer true, since we just won.).

We then send a “2” to the countdown timer using “Serial2.println(“2”)”. This uses Serial2 (**PIN 16** + **PIN 17**) to communicate with the BLE bluetooth module, which in turn sends the data to the countdown timer.

We then drive a tiny bit further forward (since this function will already be triggered before we are fully on the “winner plate”. We do this by triggering the “drive(1, 100)” which will make the robot drive forwards for 100ms.

Now we bring the robot to a full stop, using “drive(0)” and play the victory sound MP3 file (track 8).

After that we will play the “*We did it!”* MP3 file (track 9), however we want to play this after the victory sound is done, so we pass in a 3500ms delay as the second parameter (as was explained before).

Now we can trigger the dance, however we again want to start with a delay since we want to wait till both the victory sound AND the “We did it!” MP3 files are done. We do this by passing a 5000ms delay to the “robotDance()” function (explained before).

After all this is done, we will play the “*Time for a new round. Give me directions when you are ready.”* MP3 file (track 13). This, however, we want to play after all the previous things are done. This is done by passing a delay as a second parameter of 18000ms (18s).

These timings where all figured out by checking on the play time of the MP3 files, and testing if all the things where triggered in a natural order.