

Sound on the iRobot create

Florian Bachinger
SICK Robots
TGM, HTL Vienna
Vienna, Austria
fbachinger@student.tgm.ac.at

Sundas Syed
SICK Robots
TGM, HTL Vienna
Vienna, Austria
ssyed@student.tgm.ac.at

Christian Moser
SICK Robots
TGM, HTL Vienna
Vienna, Austria
cmoser01@student.tgm.ac.at

Valentin Veluppillai
SICK Robots
TGM, HTL Vienna
Vienna, Austria
vveluppillai@student.tgm.ac.at

Melina Müller
SICK Robots
TGM, HTL Vienna
Vienna, Austria
mmueller2@student.tgm.ac.at

This paper is written for the European Conference of Educational Robotics (ECER) 2018. It is about the melody we programmed one of our robots to play at our school tournament at the TGM. The paper includes how we got to this idea and why we did it. Also how the programming itself worked exactly and how we put it into practice. In the end we will conclude how this idea turned out and our opinion about it.

melody, programming, tune, music, midi, audio

I. INTRODUCTION

At the beginning of this year, on January 18 2018, we had a botball tournament at our school, the TGM. It was hosted by PRIA for all robotic students. We had to build a botball robot with the set of last year's ECER tournament. In order to stick out from the other opponents, we decided to give our robot a special feature and let him play a melody before turning off. Because the controller we used was called "link", we let him play a melody from the video game the "Legend of Zelda", where the main character is also called "Link". In this paper we will write about the implementation of the melody.

II. CONCEPT/DESIGN

A. Theoretical function

Our main robot, the "create" from last year's tournament, features a simple square-wave sound generator, which can be controlled over the serial interface of the robot. It can only play one tone at a time, and after 16 tones it has to be restarted, but the frequency and duration of each tone is, to an extend, controllable. In order to play the melody, we first had to find the corresponding MIDI (Musical Instrument Digital Interface)-file, to know which notes to play.

The MIDI format is mostly used to play or export musical content, but it works very differently to, for example, the MP3 (MPEG-1/2 Audio Layer III) format. In MP3, the music is sampled at a fixed rate, for example 44.2kHz or 192kHz, depending on the level of quality. Through the use of different

compression techniques, the file size is reduced a lot, and with a minimal loss of audio quality.

MIDI, on the other hand, works by a different approach. It is not possible to record, for example, a human voice, as it would be with the MP3-format. Instead, in a MIDI-file, only single tones, and only their frequency and length, and their position in time in the song are saved.

Furthermore, when a MIDI-file is played, it is necessary to specify the instruments which were intended by the creator. So, one of the main advantages over the sampling-technique is that music can be saved completely lossless, and in comparison to MP3 with much less disk space, but the sound is not the same, because by changing the instrument playing, the sound changes with it.

After we had found the MIDI-file for the melody, we went on with trying out shifting the melody up and down to find the one that sounded the best with our "limited" setup. But it turned out the original melody was, to our surprise, the best sounding one.

After that, it was only a matter of specifying the notes to play and their duration in the code, so at the end, when we wanted to play the melody, we just had to enable the sound module and call a function which sent the tones to the robot, and we were finished.

B. Advantages and disadvantages

Before starting, we thought about whether it would be worth doing it. The music was not directly necessary for the robot's tasks. Therefore, we had to evaluate the advantages and disadvantages of this idea and decide if we should do it.

One of the larger negative domains of the music was the energy consumption. We had to always charge our robot and look for it to be full. We did not want to risk losing because the battery was empty. With the music, we had a higher risk of this to happen. The robot would need some of its precious energy to play the music, which he might have needed somewhere else.

Another disadvantage of this function was the unknown time it would take to program it. We had the idea on the same day as the competition was held. Hence, we needed as much time as possible to finish the robot and to correct bugs. The implementation would take time that we might not have.

Also we could not play any song, because the robot could only play single notes and no chords. So some melodies would just sound weird and people would not recognize them.

But of course there also advantages of this function. One of them, simple as it is, is that it is fun. The idea for the Zelda sound is a good pun that will be appreciated by the other contestants. More people will watch our robot, because they like the sound and want to hear it in the end.

Another advantage of the melody is its uniqueness, which further separates us from the other ones. Our robot would be the only one with this feature. Therefore, more people will remember it and will tell others about it.

Even though it will of course take time it not to hard to implement it. This feature will be easily programmed and realized.

At the end, we decided to do it. The output is far bigger than the input. And even though it would take time and energy, it will be worth it and we will have a lot of fun with it.

III. IMPLEMENTATION

After we decided that we will do it we just had to put our words into action. Mainly one of our team members dealt with the music and programmed it. Therefore his first act was to look up the tunes for the "Zelda sound". This took a lot of time because they were kind of hard to find, but finally we were successful. On a sheet of paper, he wrote down the 16 tunes we decided to take and the corresponding numbers which he had from a document which he found online. Afterwards he just had to program it.

We had a lot of luck and the sound worked on the first try. The robot played it in the last seconds before he turned off. We played a bit around in different pitches but then decided for the first one.

Because the tune was such a hit and the negative impact on power consumption also turned out to be much less than expected, we thought about letting the robot play another one as well. We wanted him to play the "Donau Walzer" every time he had to turn, which happened around three times. However there we had a lot of problems.

First, the notes we found included chords. We tried to simply ignore them and let the robot play it as a single tone, but the music it played was completely different from how it should have sounded and could not be recognized.

Another problem was that we weren't able to play sounds while controlling the rest of the robot, which obviously posed a problem when the only purpose of your robot consists in moving around. So in the end, we decided to delete the part of the code which played the waltz and just kept the "Zelda" sound.

IV. CONCLUSION

All in all, it can be said that it was the right decision. We had a lot of fun with the music and people just loved it. It was fast and easy to accomplish. The time would not have been put better into action, because in the end we won with our robot. A lot of people asked us how we programmed it and were big fans of this feature.

The only disadvantage was, that we could not play chords. We will definitely do it again the next time, for example at the ECER open tournament in Malta. Hopefully, a bit more advanced the next time, and with more sounds. Maybe we will even find a way to play the "Donau Walzer" at last.