Journal of Informatics and Interactive Technology
Volume 1, Number 3, Desember 2024
e-ISSN : 3048-1023

MIDI Controller Design Using Arduino Leonardo as a Digital Audio Workstation (DAW) Management Tool

Irfan Santiko[1], Darso[2]
[1,2]Department of Informatic, Faculty of Computer Science, Amikom Purwokerto University, Indonesia
email:[1]irfan.santiko@amikompurwokerto.ac.id, [2]darso@amikompurwokerto.ac.id

Article Info

Article history:

Received 24 06 2024
Revised 31-07-2024
Received 30-08-2024

Keywords:

Audio;
Workstation;
Microcontroller;
Arduino

ABSTRACT

It is very possible for a musician who works in the modern technology era to be familiar with digital sounds or tones. They even use technology in music in the modern era. Currently, there are many models of digital music devices, both physical and non-physical. One of them is a digital effect for bass and guitar. The size of the device ranges from large to simple. The obstacle faced is that musicians, especially those who are still new to utilizing technology, often have difficulty controlling it. This is due to the simple design of the device, which makes it more difficult to use on stage because of the many preset tones and controls available. Because digital effects require large storage capacity, their function is similar to the processor in the CPU. Technology devices such as the Digital Audio Workstation (DAW) feature, allow users to produce their own music without having to rent a recording studio. On this occasion, this article will discuss the design of a MIDI manager model as a tool in carrying out the audio processing process. In the design we use Arduino Leonardo technology with an ATmel Mega 32u4 processor as the control engine. The result is a device that can control features in DAW which makes it easier for musicians to manage audio both during the recording process and on stage.

Corresponding author:

Irfan Santiko
Department of Informatic, Faculty of Computer Science, Amikom Purwokerto University, Indonesia
Email: irfan.santiko@amikompurwokerto.ac.id

1. INTRODUCTION

DAW or Digital Audio Workstation is a software or hardware used to record, edit, arrange, and produce audio, including music, sound effects, and podcasts[1]. DAW is the main tool for musicians, producers, sound engineers, and content creators to create and manipulate audio in a digital environment. DAW allows users to record audio from multiple sources, such as microphones, musical instruments, or MIDI devices. Users can record multiple tracks simultaneously[2]. Users can edit audio with precision, such as cutting, splicing, adjusting timing, and adding effects[3]. DAW also allows for pitch correction and dynamic editing. DAW allows users to arrange various audio elements in a timeline, making it easier to create music compositions, podcasts, or other audio projects[4]. DAWs usually come with plugins and effects such as reverb, delay, equalizer, compression, and more that allow users to process audio as needed[5]. DAW also supports MIDI, which allows users to control virtual instruments, synthesizers, and other MIDI devices. This is very useful for electronic music creation or computer-based production[6]. Once recording and editing is complete, a DAW allows users to adjust levels, pan, and effects on each track, as well as perform mastering to ensure the audio is of optimal sound quality[7].

DAWs are used by musicians and producers to create, arrange, and refine musical works. Artists and producers use DAWs to record vocals, instruments, and other sounds in high quality. DAWs are used to precisely edit recordings, remove noise, and adjust other audio elements. Composers use DAWs to write and arrange music with the help of MIDI and virtual instruments. Audio engineers use DAWs to mix various audio elements into one balanced, high-quality output. Popular DAWs such as Ableton Live, FL Studio, Logic Pro, and Pro Tools have become the standard in the music and audio production industry[8]. MIDI, short for Musical Instrument Digital Interface, is a standard communications protocol that allows electronic musical instruments, computers, and other devices to communicate with each other[9]. MIDI does not transmit sound directly, but sends digital messages that describe aspects of music such as pitch, velocity, intensity, and duration. These messages can be used to trigger sounds on synthesizers, control effects, adjust stage lights, and more[10].

How MIDI Works, sends information when a note is played or stopped. This message contains information about which note is played and how hard (velocity) the note is pressed[11]. The Control Change (CC) function in this message controls certain parameters, such as volume, panning, modulation, or filters on the device[12]. MIDI can also be used to change sounds or presets on synthesizers or other devices. Then it can also measure the pressure on the keys after they are played, which can be used to control other effects such as vibrato. MIDI has 16 channels that can be used to send messages to different devices or to different parts of a device. For example, channel 1 can be used for piano, channel 2 for drums, and so on[13].

One of the functions of a MIDI device is to be able to manage a DAW (Digital Audio Workstation). Software that can receive, send, and process MIDI messages to trigger virtual instruments, set controls, and automate effects[14]. Because MIDI only sends data, not audio, MIDI files are very small and can be easily changed without changing the sound quality. Compatibility: MIDI is an industry standard, so various devices from various manufacturers can communicate with each other. MIDI allows precise control over various musical parameters, from dynamics to modulation. With MIDI, musicians and producers can create and arrange music in an efficient and creative way, utilizing technology to produce complex and dynamic sounds[15].

Currently, there are many models of digital music devices, both physical and non-physical. One of them is a digital effect for bass and guitar[16]. The size of the device ranges from large to simple. The obstacle faced is that musicians, especially those who are still new to utilizing technology, often have difficulty controlling it. This is due to the simple design of the device, which makes it more difficult to use on stage because of the many preset tones and controls available. Because digital effects require large storage capacity, their function is similar to the processor in the CPU[17].

Technology devices such as the Digital Audio Workstation (DAW) feature, allow users to produce their own music without having to rent a recording studio[18]. On this occasion, this article will discuss the design of a MIDI manager model as a tool in carrying out the audio processing process. In the design we use Arduino Leonardo technology with an ATmel Mega 32u4 processor as the control engine[19]. The result is a device that can control features in DAW which makes it easier for musicians to manage audio both during the recording process and on stage.

In this paper, we will discuss how to design the MIDI Controller using Arduino. This design is designed so that musicians can easily manage audio both during recording and while on stage during their performances. The Arduino used in this design is the Leonardo Atmel 32u4, the use of the Atmel 32u4 is because the chipset used supports both traditional and USB MIDI-based MIDI ports.

2. METHOD

To facilitate the discussion and design of this project, in the design planning we use the prototyping method. Prototyping is an approach to software or product development that involves creating an early version of a product, called a "prototype", to explore basic ideas, concepts, or functionality before producing a final product. These prototypes typically do not have all the features or level of completion that will be in the final product, but are sufficient to demonstrate the core idea or functionality. The prototype method used has the following stages:

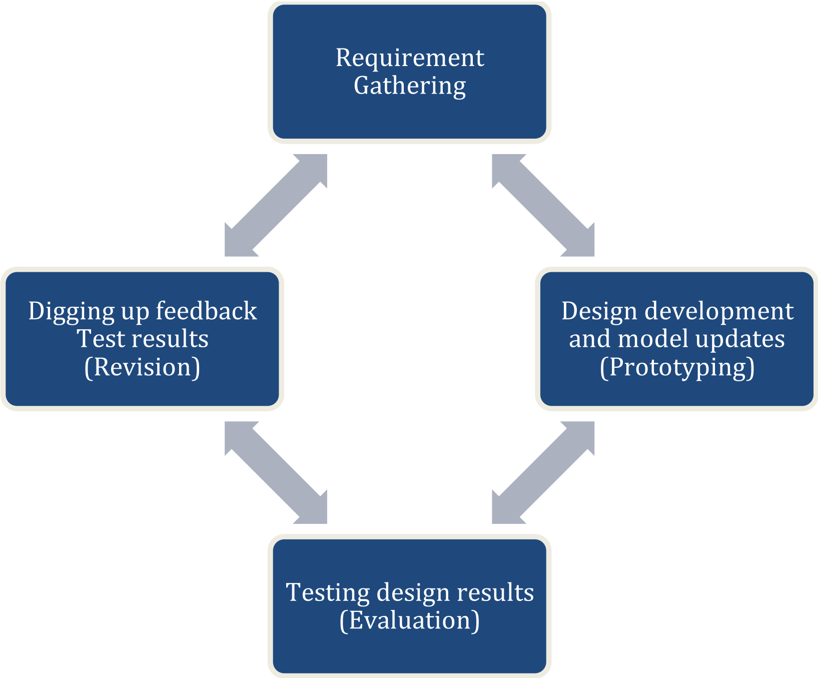

Figure 1, Design phase using the Prototype Method.

The detailed explanation is as follows:

1. Requirements Gathering: Identifying the basic needs of the user or project. This includes the main features that must be present in the prototype. At this stage, data collection takes place through user information.
2. Prototyping: Building an early version of the product with a focus on the core features or most important aspects. This prototype can be a sketch, model, or code that is functional but not yet fully developed.
3. User Evaluation: The prototype is given to users or stakeholders to be tested and evaluated. User feedback is very important at this stage to find out whether the prototype meets the needs and expectations.
4. Revision and Improvement: Based on feedback, the prototype is refined and improved. This cycle can occur several times until the prototype is satisfactory enough and can be the basis for developing the final product.

Considerations for using this Prototype because it has advantages including, Risk Reduction, By identifying problems early on, the risk of errors in the final product can be minimized. Early Feedback, Users can provide input at an early stage, so that the final product is more in line with needs. Time Savings, Potential problems are found faster, avoiding major changes in the final stages of development.

3. RESULTS AND DISCUSSION

Here is a discussion of the results we did. In accordance with the phase we use, namely using a prototype, then in this discussion it will be reviewed in stages. The stages of discussion are as follows:

3.1. Requirement Gathering.

In this data acquisition, we use a questionnaire model to obtain data results from how much digital audio technology is used by musicians. The objects collected are used randomly through the questionnaire. There are 4 main questions asked, including:

1. Does your audio project processing involve information technology? (Yes/No/Rarely)
2. Which one do you prefer to use in processing audio? (Analog/Digital/Hybrid)
3. What technology do you use? (Physical Media/Software Media)
4. What difficulties do you often complain about when using technology? (Accuracy/Buffer)

From the 4 questions, the following matrix was obtained:

Table 1, User questionnaire matrix.

User	Answer Q1	Answer Q1	Answer Q1	Answer Q1
Responden 1	Yes	Hybrid	Physical	Buffer
Responden 2	Rarely	Digital	Software	Buffer
Responden 3	Yes	Hybrid	Software	Buffer
…	…	…	…	…
Responden 20	Yes	Digital	Software	Accuracy

The conclusions that can be obtained from the questionnaire matrix are:

1. 80% of musicians currently use technology in processing audio, meaning they rely more on the output of digital filters that are considered to meet standards.
2. On average, 85% of users use hybrid, which means that the use of media is combined.
3. The technology used on average uses hybrid, meaning that the use of physical media is still needed to support the output results through digital.
4. The difficulties faced are balanced between accuracy and buffer, meaning that the comfort in playing music should be real-time and not delayed. While currently the constraints are in the synchronization between physical and non-physical, or between analog and digital.

The solution to the existing problems, then we try innovation using the technology offered through Ardunio technology. Where Arduino can also be a medium for sending digital signals or called a microcontroller.

3.2. Prototyping.

In the initial design we design the control pattern that will be used by adopting the system from the PC keyboard. Where the keyboard on the PC is a physical media that functions to control all digital functions in the computer.

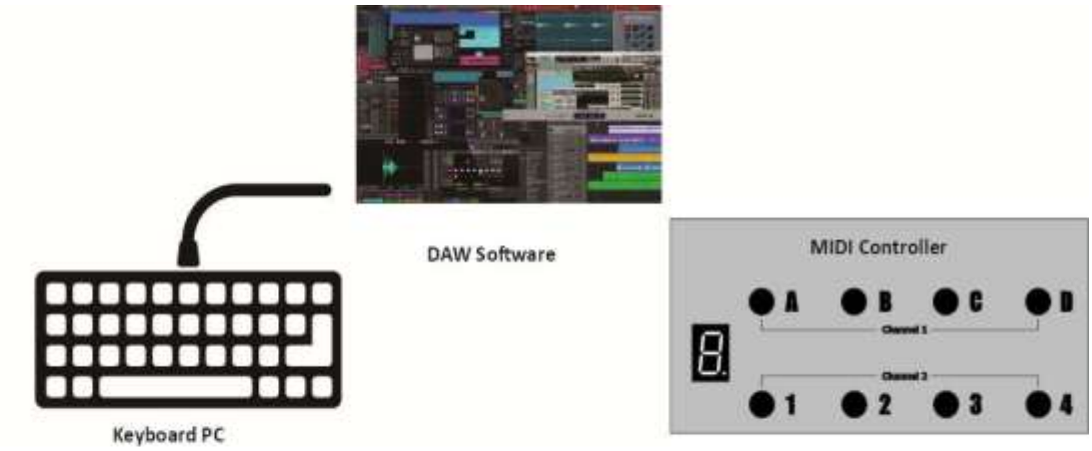

Figure 2, MIDI Controller design principles

After finding the design principles, we then prepare the materials for the design, including the following:

Table 2, List of equipment and materials to create a MIDI Controller project.

Device	Name	Function
	Board ATMel Mega 32u4 - Arduino	As a microcontroller processing media in MIDI programs.
	Push On Button	Devices to carry out controlling commands into the MIDI system.
	Seven Segmen LED	Indicators to provide information on messages managed by MIDI.
	MIDI and USB Port	Devices for connections into digital systems.
	Hardcase	Places to arrange MIDI device layouts.

After the materials are obtained, the next step is to program using Arduino IDE, the program used uses IDE 2.3 and the board according to the materials, namely Leonardo Atmega 32u4. The code in the IDE programming is as follows:

```
#include <MIDIUSB.h>
#include <SEVSEG.h>
const int buttonPins[8] = {2, 3, 4, 5, 6, 7, 8, 9};
bool buttonState[8] = {false, false, false, false, false, false, false, false};
bool lastButtonState[8] = {false, false, false, false, false, false, false, false};
void setup() {
  for (int i = 0; i < 8; i++) {
    pinMode(buttonPins[i], INPUT_PULLUP);
  }
}
void loop() {
  for (int i = 0; i < 8; i++) {
    buttonState[i] = digitalRead(buttonPins[i]) == LOW;
    if (buttonState[i] && !lastButtonState[i]) {
      sendMIDIControlChange(i);
    }
    lastButtonState[i] = buttonState[i];
  }
}
void sendMIDIControlChange(int buttonIndex) {
  byte channel = 1;
  byte controlNumber = 20 + buttonIndex;
  byte controlValue = 127;
  midiEventPacket_t event = {0x0B, 0xB0 | (channel - 1), controlNumber, controlValue};
  MidiUSB.sendMIDI(event);
  MidiUSB.flush();
  delay(10);
```

}

After the coding is declared to have passed verification, it is then uploaded to the previously determined board. The upload process is carried out using the IDE mentioned earlier.

3.3. Evaluation.

Next is testing. Where in this test we use the MIDI OX device to prove the connection results and message delivery run as expected. In the test we simulate using a hybrid device where the physical device is connected through a non-physical device media. In the results obtained through the Alpha test, the results of the feasibility test and the results of the comparison test were obtained. For the feasibility test, the following results were obtained:

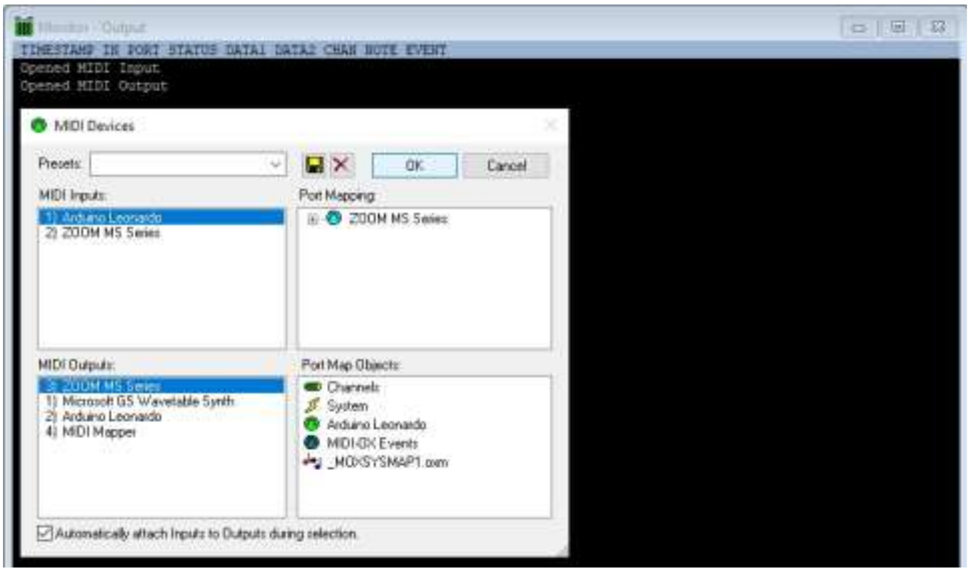

Figure 3, Simulation using MIDI OX

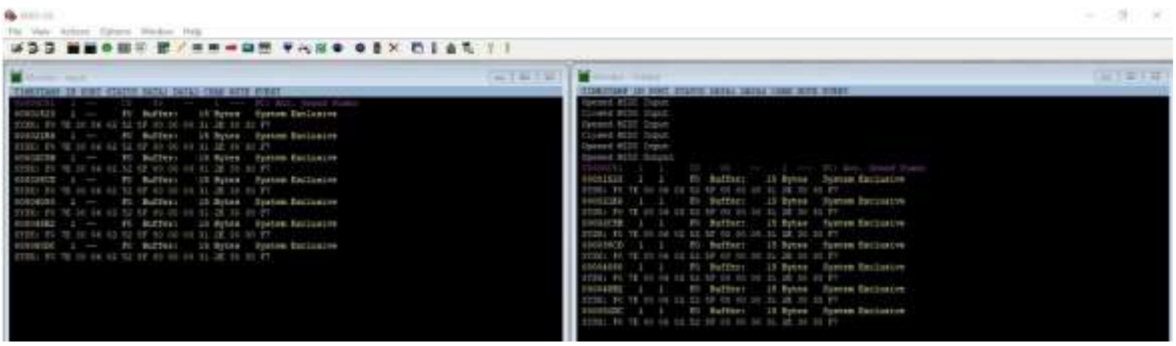

Figure 4, Test results using MIDI OX.

After being declared feasible, then an analysis is carried out based on the MIDI test. From Figure 4, it can be seen that the response to the signal delivered via the Atmel Mega 32u4 is classified as very fast, as seen from the message delivered having a buffer value of 0 for each 15 byte message, meaning 0 seconds/15 bytes. This message is classified as very fast and can be said to match the response speed of the keyboard on a PC/Computer.

3.4. Improvement.

Because the MIDI function designed is a push on command, where if the button is pressed it will give a signal, then what has been created can be used as a data/byte management media. Furthermore, it can be developed according to certain functions such as the "Program Change" or "Control Change" events. This is also needed for advanced testing, such as how precise the control is given.

4. CONCLUSION

In information technology innovation, especially in audio management, it is important to pay attention to the functionality aspect. Of course, technology is expected to be very helpful in the management process. Arduino is a technological innovation that is highly recommended for the development of this audio technology. Apart from this project, there will be many DIY products that use this Arduino technology. The microcontroller project is one of the innovations that can be used to help manage Audio. By creating this MIDI Controller project, it will become a preference for musicians to make it easier to process audio, especially on the Digital Audio Workstation (DAW). Real-time audio signal processing is needed to provide MIDI audio control that sounds like digital tones. This is an interesting project, but it can be difficult to complete. Although Arduino's audio processing capabilities are limited, there are a few methods that can be used to optimize this. It will be easier for artists, especially music composers, to regulate the tone when recording or performing live if more media is created for that purpose. This is undoubtedly very beneficial. Please note that this Arduino media should only be used by devices that can support USB MIDI. We utilized an Arduino Pro Micro, however it would be essential to add a conventional MIDI tool as well.

REFERENCE

[1] H. D. Bastyan and P. F. Utami, “Micro Controller Modeling On Digital FX Pedal For Guitar And Bass Using Arduino Pro Micro,” *J. Multimed. Trend Technol.*, vol. 3, no. 2, pp. 122–129, 2024.

[2] O. Hödl and G. Fitzpatrick, “Exploring the design space of hand-controlled guitar effects for live music,” in *ICMC*, 2013.

[3] E. Himonides, “Music technology and response measurement,” *Routledge Companion to Music. Technol. Educ.*, pp. 451–460, 2017.

[4] C. S. Rosenberg, *Evaluating alternative controllers using the MIDI protocol for human-computer interaction*. University of Washington, 1994.

[5] K. Konovalovs, J. Zovnercuka, A. Adjorlu, and D. Overholt, “A wearable foot-mounted/instrument-mounted effect controller: Design and evaluation,” in *New Interfaces for Musical Expression 2017*, 2017, pp. 354–357.

[6] E. Singer, “Sonic Banana: A Novel Bend-Sensor-Based MIDI Controller.,” in *NIME*, 2003, pp. 220–221.

[7] J. Kurniawan, P. Tubagus, and R. Domingus, “An Examination of Audio-visual Effects on Memory Capacity of Production Workers Based on Educational Targets,” *J. Multimed. Trend Technol.*, vol. 3, no. 1, pp. 33–39, Apr. 2024, doi: 10.35671/jmtt.v3i1.49.

[8] L. F. Ludwig, “Tactile, visual, and array controllers for real-time control of music signal processing, mixing, video, and lighting,” *J. Acoust. Soc. Am.*, vol. 114, no. 6, p. 3000, 2003.

[9] D. Self, “MIDI: Handbook for Sound Engineers by David Huber,” in *Audio Engineering Explained*, Routledge, 2012, pp. 457–494.

[10] D. M. Huber, “Midi,” in *Handbook for Sound Engineers*, Routledge, 2013, pp. 1088–1117.

[11] F. Jean and A. B. Albu, “The visual keyboard: Real-time feet tracking for the control of musical meta-instruments,” *Signal Process. Image Commun.*, vol. 23, no. 7, pp. 505–515, 2008.

[12] A. Pejrolo and R. DeRosa, *Acoustic and MIDI orchestration for the contemporary composer: a practical guide to writing and sequencing for the studio orchestra*. Routledge, 2016.

[13] V. Genovese, M. Cocco, D. M. De Micheli, G. C. Buttazzo, and S. S. S. Anna, “MIDI Event Generator”.

[14] S. McGuire, *Modern MIDI: Sequencing and Performing Using Traditional and Mobile Tools*. Routledge, 2019.

[15] T. Todoroff, “Control of digital audio effects,” *DAFX Digit. Audio Eff.*, pp. 465–497, 2002.

[16] L. Mueth, “MIDI Technology for the Scared to Death: Understanding MIDI opens the door to a powerful

resource in music education. Larry Mueth guides music educators through the basics," *Music Educ. J.*, vol. 79, no. 8, pp. 49–53, 1993.

[17] M. Linder, "An Arduino-based MIDI Controllerfor Detecting Minimal Movementsin Severely Disabled Children." 2016.

[18] M. Maulana *et al.*, "Design of Midi Drum Controller Using Piezoelectric and Arduino-Based TCRT5000 Sensor to Facilitate Sound Engineer Performance in the Music Industry," *J. Renew. Energy, Electr. Comput. Eng.*, vol. 3, no. 2, pp. 69–75, 2023.

[19] R. Vieira and F. L. Schiavoni, "Fliperama: An affordable Arduino based MIDI Controller.," in *NIME*, 2020, pp. 375–379.