UPPSALA
UNIVERSITET

Report for 1TE663

Project: Midi Controller

Group 28

Christos Magiras Nusrat Hossain

March 10, 2020

1 Abstract

The goal of the project was to implement a communication between the microcontroller ATmega328P and the synthesizer Korg Minilogue XD through the MIDI protocol. This provide more flexibility to use all the functionalities of the synthesizer programmatically. The required hardware for the implementation of the project includes: a microcontroller, a midi socket, a synthesizer with midi-in socket and some more electronics devices (Section 5) that would be described below. This paper includes the overview of MIDI protocol, the hardware design, software descriptions, future goals and the documentation of the source code. The aim of the project is to benefit the people who are interested in programming and music creation.

2 Introduction

This project describes the creation of a Midi Controller based on the microcontroller ATmega328p. The ATmega328P sends data to the synthesizer Korg Minilogue XD through a MIDI-OUT socket that is connected to a MIDI-IN socket of the synthesizer. The design on figure 1 demonstrates the logic.

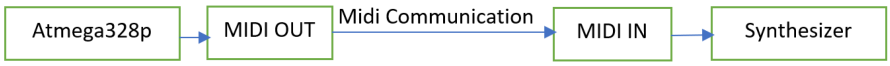

Figure 1: Project overview

3 Midi Protocol

MIDI (Musical Instrument Digital Interface) is a communications protocol for different types of sound generator devices i.e. synthesizers, sound cards, computer software like CUBASE, FL studio, guitar pro etc. It is a serial communications protocol which is functioning at 31,25 kb/sec. Each byte has 8 bits, plus a start bit and a stop bit. It runs at 5 volts DC. [1] One important thing is that the MIDI protocol does not send the soundwave itself, but instead it is sending the commands of how the synthesizer should perform. For example, if a change sound command to position 0 of the sound bank is sent, the synthesizer would change the current sound to position 0. The midi protocol uses 2 kinds of bytes, status bytes and data bytes. Status byte always have their Most Significant Bit set to 1, while Data Bytes have their Most Significant Bit set to 0. [2] Figure 2 describes a note on message, while figure 3 describes a note off message. Also, it uses 16 MIDI channels. In this project, only 1 channel is used (channel 1).

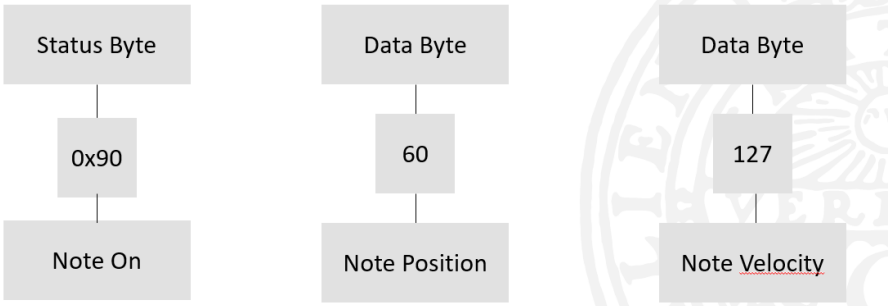

Figure 2: Note on message

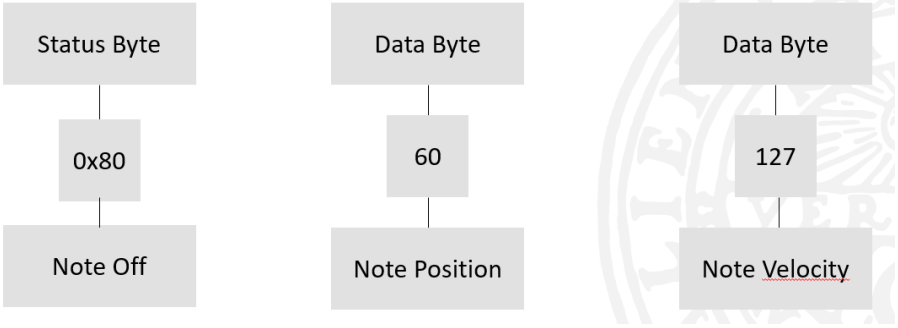

Figure 3: Note off message

4 ATmega328p

The ATmega328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48P/88P/168P/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed. [3]

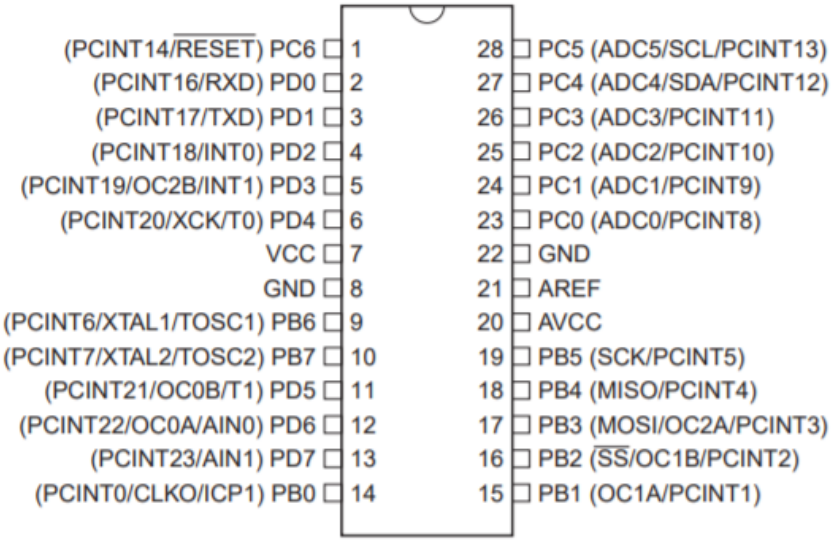

Figure 4: Pinout ATMega328P

In this project, PD1 is used for the MIDI communication between synthesizer and microcontroller. Port B was used for the button press tasks.

5 Hardware Design

Hardware parts:

- $1 \times ATmega328P$
- $1 \times BC377Transistor$
- $1 \times Breadboard$
- $1 \times HexInverter$
- $1 \times KorgMinilogueXD$
- $1 \times LCD24 - pinadapter$

- $1 \times Midicable$
- $1 \times Midisocket$
- $2 \times 220\Omega Resistor$
- $5 \times 10k\Omega Resistor$

Circuit diagram:

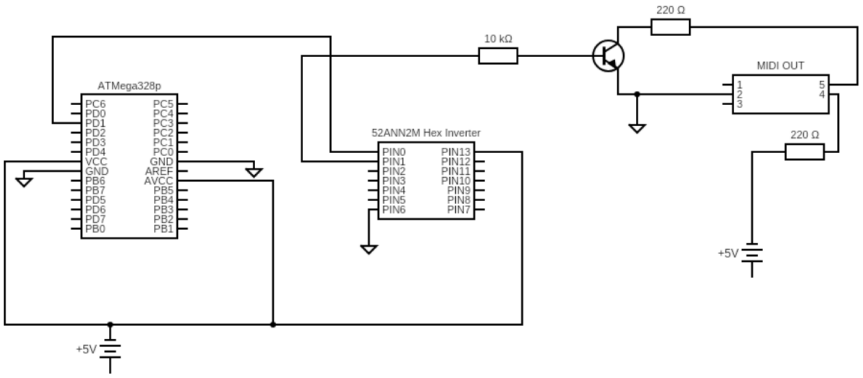

Figure 5: Overview circuit

This diagram only describes how the microcontroller is connected with the midi socket. The midi socket pins resemble the following midi diagram:

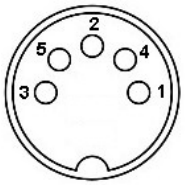

Figure 6: MIDI Socket

Microcontroller's serial output is connected to a hex inverter. This hex inverter is used to invert the data bit as ATmega328p does not perform the inversion by itself. The NPN transistor's collector is connected to pin 5 of midi socket through a 220Ω resistor while the emitter is grounded, and the base is connected to the hex invertor through one $10k\Omega$ resistor. Pin 2 of MIDI socket needs to be grounded and pin 4 is connected to 5V dc through one 220Ω resistor.
The LCD screen is used for displaying some messages and operating note on and note off status of the synthesizer. Two switches are connected, S1 and S2. In this project, only the S1 is used. Figure 7 describes how the LCD screen is connected to ATmega328P:

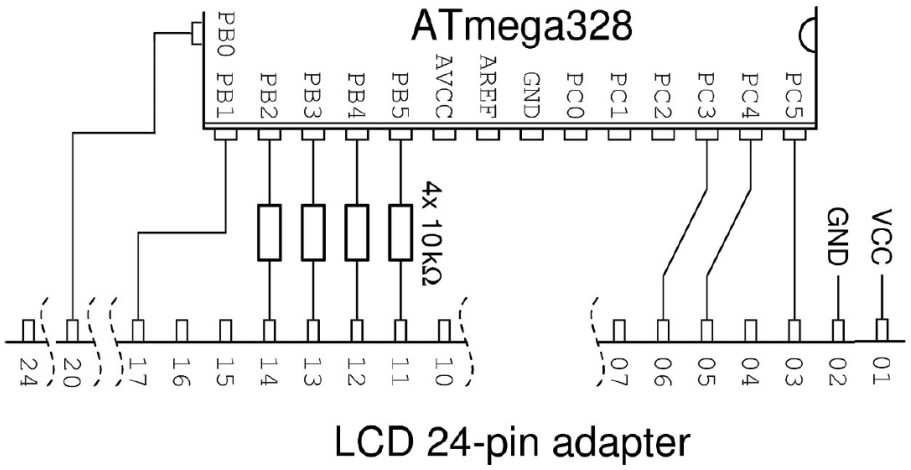

Figure 7: LCD adapter 24-pin

6 Software Design

The "uart.h" library is utilised for implementing the serial communication of microcontroller and synthesizer. Pin PD1 is being used. The command "uart_putc()" is being applied for sending one byte to the synthesizer. PortB is used for implementing the button pressed of the LCD screen. The delay until function is implemented to get the absolute wake up time interval since the delay function that it is provided by Atmel studio will cause a delay for a specified time interval. The implementation of delay until was applied through timer interrupt service routine that fires up every 10 ms. There are some declarations of variables and functions that makes the creation of music easier. The source code describes how the MIDI controller is functioning in this project.

7 Future Goals

- One desired goal is to apply the MIDI IN functionalities which runs from synthesizer to microcontroller. This would make it possible to even read the sound names, sound settings etc. of the synthesizer.
- The Korg Minilogue XD manual has a MIDI implementation chart that defines different functionalities of the synthesizer. Another anticipated goal of the project is to implement all the functionalities from this chart. [4]
- Since the note on messages have as an input field a velocity value, it is possible to show the velocity of the message through leds. For example, the higher the velocity, brighter the leds. If the project has dynamic leds, it would be more fascinating during the playing phase.

8 Conclussions

Deeper knowledge of embedded C programming is achieved through this project. It was possible to learn different operations of microcontroller especially the interrupt service routine. During the hardware design phase, we learned different internal definitions of hardware. Equally, it was possible to acquire knowledge with debugging in both hardware and software. Since the code is understandable, it is possible that other people can also experiment using their own MIDI devices.

9 References

References

[1] Tigoe
https://www.tigoe.com/pcomp/code/communication/midi/

[2] Midi Protocol Information
http://www.personal.kent.edu/ sbirch/Music_Production/MP-II/MIDI/midi_protocol.htm

[3] ATmega328p-datasheet
https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf

[4] Minilogue XD manual
https://cdn.korg.com/us/support/download/files/1362ee55daa0ec780da684b9ad9ad99b.pdf

[5] Different labs of Microcontroller programming course
https://www.uu.se/en/admissions/master/selma/kursplan/?kKod=1TE663