1. Introduction

In this lab you will build and program a MIDI (Musical Instrument Digital Interface) controller, the GME (General MIDI Explorer). A MIDI controller is a device which sends MIDI messages to a synthesizer. In Lab 4 we will send messages to the General MIDI-compliant synthesizer in the PC soundcards and listen to the resulting sounds.

The MIDI standard was established by a consortium of manufacturers of electronic musical instruments over 20 years ago. The companies Sequential Circuits, Roland Corporation and Oberheim Electronics were major players in establishing MIDI as an international standard. General MIDI (GM) is an extension of the original MIDI specification (MIDI 1.0). Whereas 1983's MIDI 1.0 did not tie certain musical instrument sounds to certain MIDI codes, GM did just that. Established in 1991, GM assigned 128 musical instrument sounds (timbres) their own 7-bit codes. SoundBlaster-compatible soundcards the world over support GM. [GM Instruments listed at http://www.midi.org/about-midi/gm/gm1sound.shtml]

The Lab 4 device, the GME, will allow the user to "explore" the sounds available in General MIDI. The GME will also record General MIDI notes coming into the device and replay the notes on command. The user will optionally interact with the GME by changing the amount of light falling on three optical sensors. The light levels will determine the instrument/note/volume combinations to be transmitted to the sound card.

2. Project Description

In the required part of the lab, the user will send MIDI notes into the GME from the computer. The GME will compress and store the sequence and the notes that are received. Once the GME is done recording, it will hold the notes in memory until a new recording is made. The notes will be played back once the user enables a playback. Two switches provide inputs to the GME that determine when it is recording MIDI messages, playing back MIDI message, or remaining idle.

In the optional part of the lab, students will manipulate the playback of notes by changing the amount of light falling on three optical sensors and the state of a 4-bit rotary hexadecimal switch. The on-chip ADC converts an analog voltage from a photo sensor into a digital value which should modify the MIDI message that will be played back. Groups are encouraged to have fun and play with these inputs to come up with their own ideas of how to modify the playback. For example, the digitized sensor values could modify the instrument/note/velocity, the playback speed, or the number of times each note is repeated during playback.

3. Design Specification

The system will be built around the ATmega32 AVR microcontroller from Atmel. The AVR will be programmed in C or AVR assembly language. You will program the device in C using the AVR Studio Integrated Development Environment (IDE). You will use some subset of the following AVR on-chip subsystems: Parallel I/O Ports, 10-bit Analog to Digital Converter (ADC), Timer1 (a 16-bit timer) and the USART (Universal Synchronous/Asynchronous Receiver/Transmitter) Serial Port. The two most important
Atmel documents you will need to refer to are the 300+ page ATmega32 datasheet and the 100+ page 8-bit AVR Instruction Set document. The GME can be based on a loop program structure with timing based on counting of 16-bit Timer1 overflows.

**Input/Output signals on the ATmega32 AVR microcontroller:**

**Play Mode – 1 Bit Input Switch** – Turns off and on Playback Mode. The play mode switch is required to play back any stored MIDI information in the order in which it was received. It is required that it plays both the note on and note off in a quick interval (around 1ms, can vary by preference), and play each Note On and Off packet in a timely manner. It can either be predefined, half a second per Note On and Off packet, or varied by a hex switch. This mode must be enabled when recording mode is disabled.

**Record Mode – 1 Bit Input Switch** – Turns off and on Record Mode. The recording mode switch is required to record MIDI notes being sent by the computer and store it in the AVR’s EEPROM data. The recording mode should keep count of how many notes have been stored and know when to stop when it has reaches its maximum storage capabilities.

**MIDI Input – 1 Bit Input Line** – MIDI Information. The MIDI input sends the notes into the AVR to store into the AVR’s EEPROM memory. The MIDI input will be received via the USART serial port (pin 14, RXD/PD0). The messages will only consist of Note On and Note Off messages.

*(Optional) Rotary hex switch input* – Provides 4-bit input selected by rotary knob. Groups can use switch to modify playback in ways of their choosing, such as repetition, changing speeds, etc.

*(Optional) Analog input from optical sensors* – analog voltage from a simple voltage divider circuit as shown in figure. Each of the three analog input signals will be connected to one of the eight available Analog to Digital Converter (ADC) inputs on the AVR, say ADC0 (pin 40), ADC1 (pin 39), and ADC2 (pin 38).

![Diagram of optical sensor setup](image)

**MIDI Output – 1 Bit Output Line** - The MIDI messages will be sent out of the AVR via the USART serial port (pin 15, TXD/PD1).
4. **Project Demonstration and Report**

You will submit your code online by the designated due date and sign up for a demo slot. At the start of the demo the oscilloscope should be displaying the USART output. MIDI-OX should be displaying the incoming messages from your board. MIDI-OX should be configured to send the messages to the PC soundcard.

In the report, provide a complete description of your design and its components. Report should be clear and concise including the following parts.

- Code should be included as a separate file with your report.
- Annotated logic analyzer printouts indicating that you analyzed the results and whether the design functions correctly or not.
- A brief explanation of the debugging and any problems encountered.
- Include a copy of the lab4 assessment form as part of your report. A second copy of the assessment will be completed by the TA at the demo.
- Include a description of your compression, with analysis showing what expected compression ratio for MIDI.
- Describe the contributions of each group member.

See website for due date. Late submissions will not be accepted.

**EXTRA CREDIT:** Extra credit up to 20% of lab 4 grade will be given for clever designs that utilize the analog and rotary inputs to modify playback in interesting ways. Credit can be used to top-off scores on labs 3 and 4, but students cannot exceed full credit on these 2 labs in aggregate.