ECE353 Lab 2 Report
SuoAn Gao, Siyu Wu, Yujia Huo

Report sections

• List contributions of each group member

SuoAn Gao: Construct the program structure with help of Siyu and Yujia. Assemble the breadboard. Write code for playback (including reading data in eeprom) and debug it.

Siyu Wu: Write the code for interrupt and modify then debug it. Debug the breadboard circuit.

Yujia Huo: Write the code for recording and debug it. Debug the breadboard circuit. Help others to debug their code.

• Paragraph on hardest bug faced and how it was debugged

The hardest bug was when it was the second time of recording, there was one leftover note from previous recording.

Firstly, we thought when starting the second recording, the writing address did not clear to zero. But it was not the situation that was in our code. Then we found something about USART online, and then we changed our hypothesis to "We might have something remain in USART". We assured that by having a couple of experiments and then eliminate all of other factors. Then we viewed our code, and figured out that we didn't have anything to clear USART buffer. We have the intuition to clear cache memory in computer, so we decide to find a function to clear the USART buffer. There was a function called flush the UDR. This one should be used to clear remain data in UDR. We added this function at the beginning of the USART receive, then our bug solved.

• Paragraph describing how the timing specs are satisfied

There two timing specifications. One is that we might have 10 s between notes. It means the timer should overflow before 10s. There should be some prescale of the timer 1.

The clock is 4MHz. It is a 16-bit timer.

The time that it overflow: $2^{16}*1/(4*10^6)=16.4ms$

$10s/(16.4ms)=610$, $256<610<1024$

So we choose the prescale to be 1024 which is to set the last three bits of TCCR1B to be 101.

The other one is about interrupt. In this case, the interrupt shoud be the compare match interrupt. There are normal mode and CTC mode which both can achieve compare match interrupt. Since the timer should start when a group of 3 bytes arrives rather than right after the interrupt happens, and the CTC mode will clear timer zero, we should use normal mode interrupt rather than CTC mode to achieve this. Also, in compare match interrupt, we need to set a top value stored in OCR1A. The function is :

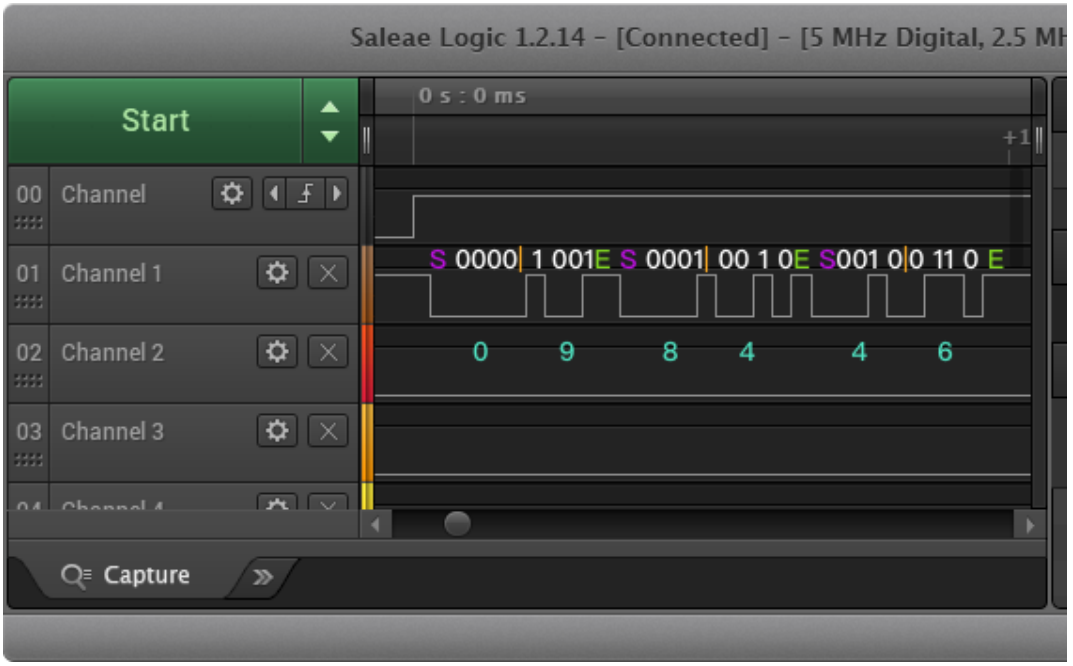
$$f_{OCRnA} = \frac{f_{CLK-I/O}}{N \cdot OCRnA}$$

To set 500ms, we need to calculate the value for OCR1A=0x7A1. When the OCR1A =0x7A1, the T_{OCRnA}= 499.968ms. Thus the error is 0.032 ms.

Annotated logic analyzer printouts included in report

• A single printout that shows playback switch turning on, and the first three bytes that are transmitted out of the USART

• Annotations on the printout that state the value of each of the bytes



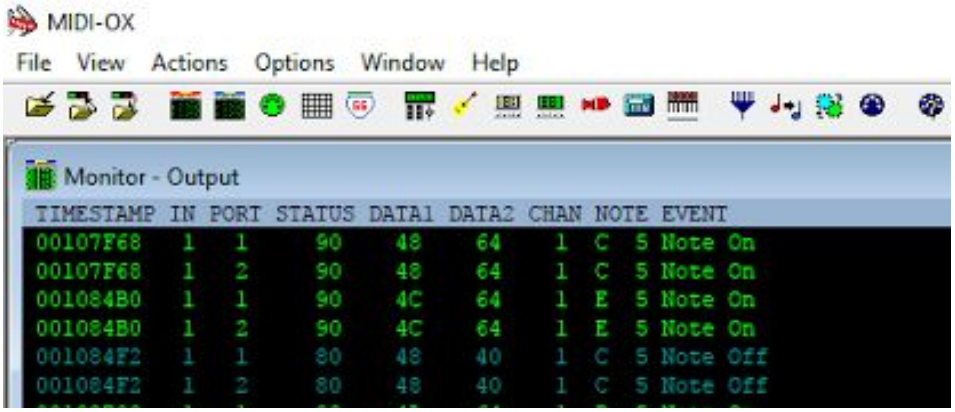
S: start bit

E: stop bit

First byte 1001000=0x90
Second byte: 01001000=0x48
Third byte: 01100100=0x64

• Short paragraph of text explaining why printout is showing correct behavior or not (note that you can get these points even if your printout is wrong, as long as you can show why it is wrong)

The first channel is the voltage on pin that the playback switch controls, it changes from low to high states that the playback mode is turned on. After the playback mode is turned on, the second channel, channel 1 is the data transmission at pin TXD. As you can see in the Midi output window, the first line indicates the first note containing 3 bytes that pin TXD outputs(90, 48, 64). We also have a start-bit and a stop bit to wrap-up each of byte. Notice that the output series should be from the least significant bit to the last significant bit (you can see that from picture above).We thus can observe these 3 bytes agree to pulses in channel 1which state the start 3 bytes of the first note.



Questions from lab description

• Analysis of timing precision is correct

Given your scheme for measuring timing between notes :

When USART receives first group of 3 bytes, timer starts. And when USART receives second group of 3 bytes, read the TCNT1 value which is the measured timing between 3 bytes and clear the timer. And then read the value of timer after receiving the third group of three bytes and then clear the timer……

Above that we measure is the time interval between every group of 3 bytes rather than the time between notes. To get the timing between notes, we need to add two time intervals between groups of 3 bytes.

What is the maximum imprecision in timing interval between two notes?(In other words, if A is the actual time between the start of two notes in a song, and B is the actual time between when you play back the same notes, what is the maximum difference between A and B?)

Our prescale of the timer is 1024. So timer value changes every 1024*1/(4*10^6)=0.256ms. The error between every group of three bytes:

If the actual time interval is right before the timer next change but the timer only represents the time right after this timer change. For example, if a actual time interval is 0.5119999999ms, the timer value at this time is the same as the timer value at 0.256ms. So the max difference is 0.256ms.
The max error between two notes:

0.256*2=0.512ms

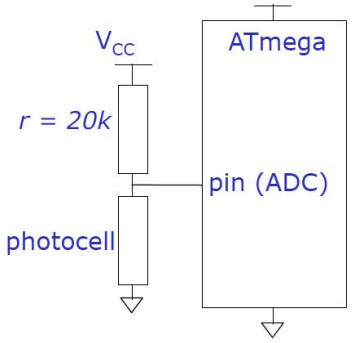
How much of that difference is caused by imprecision in the recording timing, and how much is caused by imprecision in the playback timing?

In our case, the imprecision is only cased by the recording timing. During the playback, the the delay between transmission of two groups of 3 bytes are implemented by

```
while(TCNT1<TimeDel) ;
```

So what we do is only reading the time that is stored in the EEPROM and compare it with the timer. This will get out of the while right after TCNT1 changes to the measurement time. So there is no error during the playback.

• Correct calculation of photocell resistance under three lighting conditions (show your work, starting with ADC values)



20k resistance and photocell are a voltage divider. knowing the pin ADC voltage, we can get the resistance of photocell.
For shining, the ADC value is around 7. Vin=ADC*V_{ref}/1024=0.034V. The curent in photocell is 0.2483mA. The resistance of photocell is 0.137 kohm.
For normal, the ADC value is around 145. Vin=0.71V. The current in photocell is 0.2145mA. The resistance of photocell is 3.3 kohm.
For coverd, the ADC value is around 1020. Vin=4.98V. The current in photocell is 0.001mA. The resistance of photocell is 498 kohm.

In addition, we used Multi-meter to measure the resistance of the photocell under these three condition. The measurement value agrees to our calculation.