Project Final Report

Voice Recorder

May 01, 2008

Thadeous A. Goodwyn
tgoodwyn@sparta.com

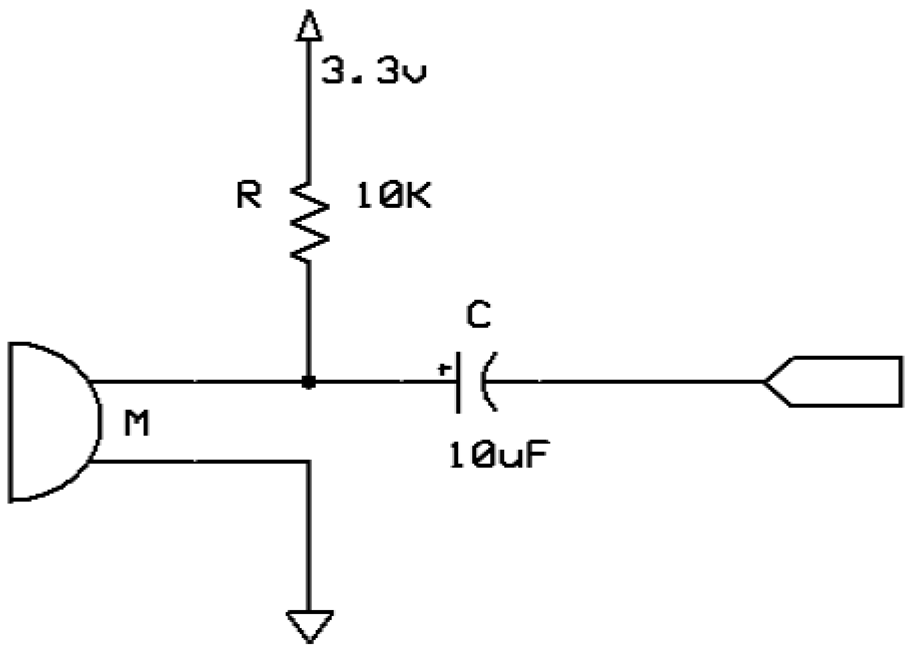
Project Abstract
For my final project I created a digital voice recorder using my Z8 encore evaluation board. The problem of digital voice recording and playback introduces an engineer to many different concepts. The overall process of digitizing an analog signal, storing a digital representation and finally reconstructing an analog signal can be attacked with more than one technique. I look to explore more than one approach. Mainly, I look to examine pulse width modulation (PWM) for sound creation and 1-bit audio. For both of these techniques there are many parameters that can be tweaked to increase performance. This project can also be a fun test bed for simple RC filters.

Status
A poor quality voice recorder is implemented with two different techniques. The *Issues* section contains more information concerning outstanding problems. My PWM implementation currently works better; but for a good reason (analysis in *Issues* section).

Introduction
As stated earlier I attacked this problem with two techniques. The first technique is probably the most obvious. The technique involves directly storing and using the values calculated by the analog to digital converter (ADC). Then, a PWM timer is used to produce the sound. The second technique evaluated is referred to as 1-bit audio. Instead of simply storing and directly using the values calculated by the ADC, the current reading is compared with the previous reading. If the difference is greater than some threshold a '1' is stored. If not, a '0' is stored. Basically 1-bit audio stores a stream of 1-bit values. The magic of 1-bit audio is in the calculation of the threshold (or better referred to as step size). The value is a function of the playback rate and the values of the electronics components used in the RC filter. I will examine this further later in my paper. Note, the later technique requires no PWM timer.

Digitalization
To provide audio to the Z8, a microphone is connected to an analog input port on the Z8. I will initially skip over the circuit connecting the microphone to the Z8. A simple circuit diagram is included below.

We will address characteristics of the circuit later. The Z8 microprocessor, like any other microprocessor, deals in ones and zeros. In order for the Z8 microprocessor to "read" microphone input, the input must be digitized. The signal coming from a microphone is an analog signal with voltage differences representing sound waves. In my project, the very first required task for both techniques is to convert the analog signal into a a collection of digital values (samples). For this task I used the on-board Z8 ADC converter. This piece of hardware takes in an analog signal and returns a digital representation of the voltage magnitude of the signal at a given time. First, it is important to consider characteristics of the ADC in the context of its use. The on-board ADC is a 10-bit linear converter. This means that a sample reading has a precision of 10 bits and preforms a linear mapping from the input value to the output value. An ADC using a reference voltage to determine the maximum value and "scale" the samples. For my project, I used a reference voltage of 2.0 volts. Since the converter is 10 bits, there are 2^10 possible values. This results in a step size of 2/2^10 which equals 1.95 millivolts. With the microphone circuit I used, the microphone provided values between 170-260 which is consistent with what I observed on the oscilloscope.

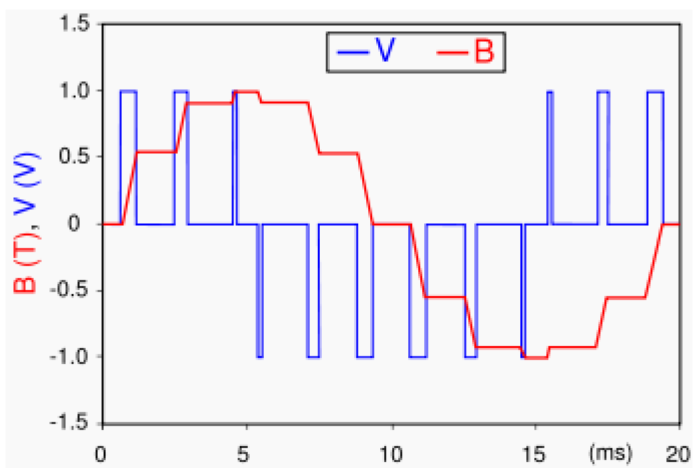


Following analog to digital conversion, the two techniques differ. The first technique involved simply storing the values returned which is directly used to generated sound. The second technique involves comparing the current value with the previous value. If the difference is over some threshold value, a '1' is stored. The threshold value comes from some simple algebra. A capacitor is commonly used to "smooth" a signal. This is because a capacitor has the ability to store and release current. Changing (or actually keeping the same output value over time) results in a different voltage across the capacitor. It is the difference in voltage that is used in the threshold value for determining if a one or zero should be stored during the recording phase. This means the ADC value difference between consecutive samples should be multiplied by the ADC step size which gives the difference in volts. That number should be compared with the threshold value. A great reference can be found at http://centauri.ezy.net.au/~fastvid/picsound.htm .

Playback

The first section described the process used to digitize the input signal. In this section I will describe the process of generating sounds from the digital representation of the sound signal. Like I mentioned earlier, I attempted to solve the problem with two different techniques. With the first technique I simply stored the values calculated by the ADC. To generate an analog-like output signal a PWM clock is used. Generating an analog-like signal using PWM is not very difficult but depending on the problem domain certain issues need to be addressed. A pulse width modulation clock has two important parameters. The first is the reload time. The second is the PWM value. A PWM clock is very simple. It starts at an initial clock value and increases value until it hits the reload time. Upon hitting the reload time, the timer starts over. Everything just described is consistent with a normal continuous clock. The remaining functionality makes it a PWM clock. The timer initially outputs a low signal on the output pin. When the counter reaches the PWM value, the output switches to a high value. The value is maintained high until the clock is restarted.

The process of creating an analog-like signal from a PWM clock may not initially be obvious. The technique involves manipulating the PWM value over time. This results in the output pin being in the high state for different periods of time. As a general rule of thumb, the PWM value should increase as the ADC value increase. Since I was interested in a pretty narrow frequency range, I preformed some simple multiplication emphasizing my range of interest. It is important to note that the ADC range should not be the driving force for determining the PWM clock reload value. A mapping function should be used to map the ADC output values to PWM values. Again, the important rule of thumb is 'a larger ADC value should result in a larger PWM value'. A capacitor can be used to smooth the output signal. Also, a speaker has internal characteristics that will smooth the signal.



You can think of (B) as the analog signal generated
by the PWM signal (V). Thanks to Wikipedia for the picture.

There are two issues that need to be addressed with audio generated with PWM. First, the carrier frequency. The clock reload frequency can be thought of as a carrier frequency. If this frequency makes it into the human audible range[1], noise will be sent to the speaker. A low pass filter can be used but if the designer is not careful the frequency may lay in the desired playback frequency range. The second issue can be observed when a quick clock is used, or better stated, a small reload value is configured. If your PWM clock cycles are very short then the highs and lows are generated and switched very quickly. If they are too quick, the volume produced with not be loud enough.

For the second technique used, 1-bit audio, sound is generated very differently. Instead of using a

1 Technically the lab speaker I am using doesn't support the full human range so the speaker its self filters frequencies close to 20khz and lower frequencies (but obviously your carrier frequency probably isn't that low).

PWM clock, 1-bit values (1 or 0) are stored with the knowledge of the capacitor, resister and sampling values used. The characteristics (mainly voltage across the capacitor) of the RC circuit can be modeled algebraically such that the recorder code can be aware of the effect of writing a low or high to the output port.

Issues

First, I found the on-board ADC slower than my initial estimation. It is not quite capable of converting at a 1khz sampling frequency. From the Nyquist sampling theorem it is known that the sampling rate needs to be at least as large as twice the highest frequency of the original analog signal. For a normal human conversation the maximum frequency is around 300hz. This means I must sample at a frequency of at least 600hz. Really, for the result I want, I needed more. To get the results I want, I need to use an external ADC. I would want something that supports 8khz sampling. Given my limited microphone output range I could possibly compromise with an 8 bit converter for a faster conversion rate. To make matters worse, the voltage discharge/charge across the capacitor diverges from the modeled value as time steps increase for 1-bit audio. The model assumes a linear function but as my playback frequency decreases my step sizes increase and the actual voltage diverges from the modeled voltage difference.

Second, I probably need a better microphone amplification circuit. The voltage range or interest really isn't that great. It would be nice if the microphone circuit varied the voltage a little more.

Third, for my PWM implementation I used really low clock reload time. I believe this results in a better sound but it definitely results in a lower volume. To increase the volume, the reload time could be increased which results in the output pin staying high (or low) for longer periods of time. The mapping function from ADC value to PWM value would have to be updated of course. In my project I could only increase it by a couple of factors before I would need to filter the carrier frequency out. The volume gain was not great enough to justify changing the reload time.

Finally, my resister selection was limited. As a result I was trying to use a 100kohm potentiometer to construct a 7.7kohm resister. This is not a good idea. To get better results I needed to use a lower resistance potentiometer. This would have increased the “smoothness” of my 1-bit audio.

Conclusion

I learned a lot about digital sound playback. The concept of PWM for analog signal generation is a pretty interesting concept. I was more impressed with the 1-bit audio solution. The simplicity and elegance of the solution makes it a very attractive approach for low to medium quality audio recorders. I listened to samples on the Internet with a sampling rate of 8khz. The samples were very impressive. If I had a faster ADC I feel the 1-bit audio solution would have given me great results. With the slow ADC I found better results with the PWM technique.