COMP 4550
Real-Time Systems
PWM and Sound Generation

Autonomous Agents Lab, University of Manitoba
jacky@cs.umanitoba.ca
http://www.cs.umanitoba.ca/~jacky
http://aalab.cs.umanitoba.ca
Sound

- Music consists of individual notes
- Mathematically a note can be defined as a sine wave function

\[ y(t) = a \sin(\omega t + \phi) \]

- Amplitude \( a \)
- Frequency \( \omega \)
- Phase \( \phi \)
- A-440 is the tuning standard since 1939
  - 440 Hz
Sine Wave

- Amplitude controls volume
- Frequency controls pitch
Sound Generation using a Microcontroller

- The speaker on the educational robot is connected to pin Port E3/OCR3A.
- Movement of speaker membrane moves the air, which is what you can hear.
- If pin Port E3 is
  - on = speaker magnet is on
  - off = speaker magnet is off
- Turning magnet on/off quickly
  - Digital control of the speaker
Sound Generation

- Some boards include dedicated digital to analog converters (DAC)
  - Extra cost for hardware
  - Component matching
  - High end audio devices
- Simplest DAC is a pulse width modulated signal into a low pass filter
  - Most micro-controllers and cheap MP3 players
- How to generate a sine wave using a single bit signal
Sound Generation

- Magnet pulls the membrane in only one direction
- Assume that we have a very fast clock frequency compared to the sound we want to generate
- Assume that output on pin is changed at 10 kHz
  - 10,000 times per second
- Behaviour of the speaker if we sent the following bitstream
  - 00001111,11110000
  - 00110011,00110011
  - 00000111,11100000
  - 00001111,11100000
Sound Generation

- 00001111,11110000
  - Speaker magnet is on for 8 ticks, off for 8 ticks
  - Frequency of the sound is 10KHz/16 = 625 Hz
  - Close to a D#

- 00110011,00110011
  - Speaker magnet is on for 2 ticks, off for 2 ticks
  - Frequency of the sound is 10 KHz/4 = 2500Hz

- Amplitude is determined by the amount of movement of the membrane

- 00000111,11100000
  - Same frequency as 00001111,11110000
  - Lower amplitude
  - Maximum amplitude is 1/2 * frequency
Implementation

- To generate sound, we want to generate a repeating stream of on/off
- Timing must be very exact to make sure that you do not get any clicks
- Pulse width modulation
  - Generate a variable on pulse
  - at a fixed frequency
Pulse Width Modulation (PWM)

- Simple and therefore very popular method for generating analog output
- Also used for encoding inputs
- Signal is repeated (Period)
- Signal stays on for some percentage (Duty cycle) 50%
Pulse Width Modulation (PWM)

- Duty cycle determines the voltage applied to the system
- Motor control
  - Analog voltage is not linear. Motor may not start if voltage too low
  - PWM turns the motor on/off with full voltage
- RC Servos using PWM input to control position
  - 20ms period
  - Pulse between 0.7ms to 1.9ms to control position of servo
H Bridge

- An H-Bridge allows us to control the direction that the motor is turning
- Four transistors/MOSFETs

<table>
<thead>
<tr>
<th>S0</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Brake All</td>
</tr>
<tr>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>Brake Left</td>
</tr>
<tr>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>Brake Right</td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>Off</td>
<td>Short Circuit Death S1-&gt;S3</td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>On</td>
<td>Motor Left</td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>On</td>
<td>Off</td>
<td>Short Circuit Death S1-&gt;S3</td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>Brake Right</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>Off</td>
<td>Off</td>
<td>Short Circuit Death S0-&gt;S2</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>Off</td>
<td>On</td>
<td>Short Circuit Death S0-&gt;S2</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>On</td>
<td>Off</td>
<td>Motor Right</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>On</td>
<td>On</td>
<td>Brake Left</td>
</tr>
<tr>
<td>On</td>
<td>On</td>
<td>Off</td>
<td>Off</td>
<td>Short Circuit Death All</td>
</tr>
<tr>
<td>On</td>
<td>On</td>
<td>Off</td>
<td>On</td>
<td>Short Circuit Death S0-&gt;S2</td>
</tr>
<tr>
<td>On</td>
<td>On</td>
<td>On</td>
<td>Off</td>
<td>Short Circuit Death S1-&gt;S3</td>
</tr>
<tr>
<td>On</td>
<td>On</td>
<td>On</td>
<td>On</td>
<td>Brake</td>
</tr>
</tbody>
</table>

![H-Bridge Circuit Diagram](image)
Phase and Frequency Correct PWM Mode

- Timer TCNT1 counts
  - up from BOTTOM=0 to ICR1
  - then down from ICR1 to BOTTOM
- Whenever the TCNT1 value matches OCR3 PIN OCR3A/Pin PORT E3 is toggled
- Normal and inverted outputs for OCR3A
- If TCNT1 matches OCR1 then
  - if counting up -> turn OCR3A on
  - if counting down -> turn OCR3A off
Phase and Frequency Correct PWM Mode

Fig. 48. Phase and Frequency Correct PWM Mode, Timing Diagram

- OCnA Interrupt Flag Set or ICFn Interrupt Flag Set (Interrupt on TOP)
- OCRnx/TOP Update and TOVn Interrupt Flag Set (Interrupt on Bottom)

TCNTn
OCnx
OCnx
Period

(COMnx1:0 = 2)
(COMnx1:0 = 3)
Phase and Frequency Correct PWM Mode

- If ICR3 is
  - increased then frequency/pitch is lower
  - decreased then frequency/pitch is higher
- If OCR3A is
  - increased then amplitude/volume is lower
  - decreased then amplitude/volume is higher
Phase and Frequency Correct PWM Mode

- Set PIN E3/OCR3A as output
  - Initialize DDRE.3 to 1
- TCCR3A
  - COM3A1 – select compare output mode toggle (01)
- TCCR3B
  - Select Phase and Freq. Correct PWM Mode (WGM=1000)
  - Select prescalar to 1 (CS1 = 001)
- OCR3A defines volume
- ICR3 defines frequency
Waveform Generation

- The previous code can generate sounds/melodies with a single tone/note
- Generate polyphonic sounds is not possible using this method
- Same principle can be applied to play a waveform
  - recorded/sampled from a microphone or other audio source
Waveform Generation

- 8 bit unsigned values sampled at 8000 Hz
- Difference encoding
  - Use a faster timer interrupt and
  - Turn speaker on if next sample has higher value
  - Turn speaker off if next sample has a lower value
- Integrate to arrive at the approximate speaker setting
- Note that this is not easily supported with a PWM function