Morse Code Decoder

A 6.111 Final Project
by
Christopher Stephenson
Why choose Morse Code?

- Interesting to get a digital system to read
- Relatively easy to get digital system to read
- Good opportunities for interesting input / output
- Gives something to play with
Morse Code Specification

- Timings based off dots which are 1 unit
- Dash is 3 units
- Pause between dots and dashes 1 unit
- Pause between letters 3 units
- Pause between words 7 units

<table>
<thead>
<tr>
<th>A</th>
<th>--</th>
<th>N</th>
<th>--</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>---</td>
<td>O</td>
<td>---</td>
</tr>
<tr>
<td>C</td>
<td>---</td>
<td>P</td>
<td>---</td>
</tr>
<tr>
<td>D</td>
<td>---</td>
<td>Q</td>
<td>---</td>
</tr>
<tr>
<td>E</td>
<td>.</td>
<td>R</td>
<td>---</td>
</tr>
<tr>
<td>F</td>
<td>---</td>
<td>S</td>
<td>---</td>
</tr>
<tr>
<td>G</td>
<td>---</td>
<td>T</td>
<td>--</td>
</tr>
<tr>
<td>H</td>
<td>---</td>
<td>U</td>
<td>---</td>
</tr>
<tr>
<td>I</td>
<td>--</td>
<td>V</td>
<td>---</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>W</td>
<td>---</td>
</tr>
<tr>
<td>K</td>
<td>--</td>
<td>X</td>
<td>---</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Y</td>
<td>---</td>
</tr>
<tr>
<td>M</td>
<td>--</td>
<td>Z</td>
<td>---</td>
</tr>
</tbody>
</table>

Image Copyright British Library
System Overview

Morse Code Source → Morse Code Decoder → Screen Driver → Screen

Keyboard → Morse Code Encoder → Screen Driver

Audio Output
Screen Driver

Text Buffer

Circular Buffer containing Text

Output to Screen

Next Character from Encoder
Next Character from Encoder Available
Next Character from Decoder
Next Character from Decoder Available

Character to draw at position

Character Position
Encoder

Keyboard — ASCII Characters — Morse Code Encoder

Lookup Table Containing ASCII to Morse Mappings

Morse Code Encoder — Beeps — Audio Output

Screen — Text
Decoder

Demodulated and de-noised 1 bit Morse Code Signal

Symboliser
- Deals with turning beeps into numbers
- Serial stream of times

Converter
- Processes the numbers to produce known tokens
- Serial stream of dots, dashes and spaces

Matcher
- Matches the tokens to produce ASCII Characters
- Stream of ASCII Characters
Converter in Detail

- Must determine the "Clock"
- Must classify pulse lengths into dots and dashes
- Must classify gaps as inter character or inter word spaces
- Uses Moving Average
- Determines threshold lengths from previous stats
- Allows it to be robust to change of symbol rate
Extensions

- Using an FFT, detect what part of an audio spectrum contains a Morse signal
  - Requires DSP to clean up and demodulate signal

- Process “Non Ideal” Morse – i.e. Human tapped Morse
  - Requires that the Converter be made more robust

- Recognize a tapper's Fist
  - Not sure if this is actually possible given the time, but might be worth a shot!
# Timeline

<table>
<thead>
<tr>
<th>Week</th>
<th>Task Complete</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Screen Driver Complete</td>
<td>11/18 - 11/24</td>
</tr>
<tr>
<td>2</td>
<td>Encoder Complete</td>
<td>11/25 - 12/01</td>
</tr>
<tr>
<td>3</td>
<td>Main Decoder Complete</td>
<td>12/02 - 12/08</td>
</tr>
<tr>
<td>4</td>
<td>Frequency Scanner and Final Report Complete</td>
<td>12/09 - 12/12</td>
</tr>
</tbody>
</table>

The aim is to get everything up to main decoder done

If the project slips, the frequency scanner will be dropped

If the project under runs, a more robust converter will be added
Summary

- Produce a Morse Code decoder
- Decoder outputs to a screen
- Can also produce Morse from Keyboard input
- All done by the 12th of December

Samuel Morse, Inventor of the Morse Code