|  |
| --- |
|  |
| Lab 5: Design Project |
| Acoustic Data Modem |
|  |
| **Greg Malysa (gjm76) and Arseney Romanenko (asr96)** |
| **5 May 2010** |

# Introduction

For our final design project, we designed and built a prototype acoustic modem to serve as a physical transport layer for digital communications. It converts between a digital communications scheme (RS-232) and an acoustically coupled communications scheme of our own design. Our project consists of a pair of such modems to operate as transmit/receive pairs and supports duplex communications. Although our modem operates in air, it is a proof-of-concept experiment in encoding, decoding, and data transmission techniques that will be used in the following years by the CUAUV team to design a system capable of communicating over some distance underwater.

# High Level Design

### Rationale

We chose this product because of its relevance to a student project team we are both members of, CUAUV (Cornell University Autonomous Underwater Vehicle team). Usually, remote-controlled vehicles use radio frequency (RF) wireless communications to transmit data between the device and its operator. However, due to the nature of water, electromagnetic waves do not propagate well, with an effective range of about a foot, depending on frequency. This makes them unsuitable for communicating with AUVs. Acoustic data, however, can propagate very far underwater and acoustic underwater communications is currently an important area of research at Woods Hole Oceanographic Institute (WHOI). Inspired by their success, our team has as a long term goal the creation of an acoustic modem that can be used to communicate with the vehicle while it is in the water without a tether. Our project serves as a prototype to help us develop our own algorithms and techniques for encoding and processing data acoustically.

### Background Math

Because we used cheap audio-range speakers and microphones, our project was limiting to transmitting within the range of human hearing, ~20 Hz – 20 kHz. Therefore, we selected an encoding scheme that minimized per-byte bandwidth utilization while retaining simplicity, so that it was possible to implement decoding. The two most intuitive and basic digital encoding schemes are frequency-shift keying, where data is transmitted on a pair of frequencies, each representing a distinct digital value, and on-off keying, where the presence or absence of a single frequency is used to encode a digital value. We chose the second (OOK) because it requires half the acoustic bandwidth (uses only one frequency per channel, rather than two) to transmit a particular amount of data. We also chose to use an asynchronous design in order to avoid paying for the overhead of having an additional clock frequency, instead breaking the transmissions into a series of known-width pulses. Transmitted data is broken into chunks of 64 samples so as to conveniently fit into a power of two size buffer so that it can efficiently be implemented as a circular buffer. Each physical frame consists of a start bit (S1, a one bit), eight data bits, most-significant-bit first, and a stop bit (S2, a zero bit). At a sampling rate of 40 kHz (chosen to avoid aliasing across the entire 20 Hz-20 kHz range), 64 samples corresponds to a pulse width of 1.6 ms, which limits each frequency channel to transmitting 62.5 complete frames per second as a theoretical max.

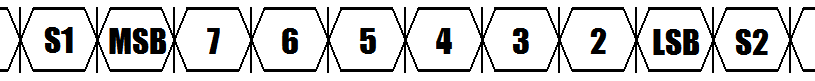


Figure - Frame Decomposition

Next, as we are limited by the computational power of our Atemga644 microcontroller, and we would ideally transmit and receive from a single microcontroller, we designed a hybrid algorithm for detecting transmitted data using FIR filters to estimate the magnitude of specific frequency components over time.