



**THE UNIVERSITY  
OF ADELAIDE**  
AUSTRALIA

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING  
ADELAIDE, SOUTH AUSTRALIA, 5005

## **ELEC ENG 4039A/B**

### **Honours Project**

A Radio Relay System for Remote Sensors in the Antarctic

Supervisors: Dr. Chris Coleman, Dr Said Al-Sarawi

## **Final Report**

**Written by:**

Mark Jessop

1163807

Date Submitted: October 22, 2010

Signature of Supervisor:

## **Abstract**

A common problem with remote sensor systems is the retrieval of data. Satellite-based systems are expensive, as is travelling to the sensor. HF propagation provides an inexpensive alternative. Radio signals below 30MHz can easily bounce off the ionosphere, travelling thousands of kilometres using only a few watts of transmit power. Based around an Atmel XMeta Micro-Controller and using Direct Digital Synthesis techniques, this project aims to provide a reliable low power HF telemetry system, usable in a variety of remote telemetry applications. By making use of the XMeta's power-save modes and using high-efficiency RF amplifiers, power consumption is minimised, allowing months of operation from battery power.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Hardware</b>	<b>5</b>
2.1	CPU . . . . .	5
2.2	Signal Generator . . . . .	6
2.3	Pre-Amplifier . . . . .	6
2.4	Class E Power Amplifier . . . . .	6
<b>3</b>	<b>Software</b>	<b>6</b>
3.1	Hardware Libraries . . . . .	6
<b>4</b>	<b>Testing</b>	<b>6</b>
<b>5</b>	<b>Project Management</b>	<b>6</b>
<b>6</b>	<b>References</b>	<b>6</b>
<b>Appendix A</b>	<b>Software</b>	<b>6</b>
A.1	The Arduino Project and the XMega . . . . .	7

# 1 Introduction

Many scientific experiments require data logging over a long time period, and in remote locations. To further research progress, it is desirable for some of the collected data to be accessible throughout the run of the experiment.

A number of solutions exist to obtain data from a remote sensor unit. The first is to physically access the remote sensor station, and copy data from whatever storage may exist. For very remote sensor units visiting the site may be impractical or too costly, so some form of wireless communication is often used.

Satellite data transmission is a reliable way of retrieving data, but comes at a high cost. For example, using the Iridium Satellite constellation to transmit data would cost approximately USD\$2.50 per minute, with a 2400 baud data rate<sup>[1]</sup>.

The other option is to other methods of radio communication. For reasonably short ( $<50\text{km}$ ) distances VHF or UHF telemetry can be used. For example, the Australian Bureau of Meteorology uses a network of VHF transmitters<sup>[2]</sup> and repeaters to obtain data from weather stations around Adelaide. VHF and UHF can work for longer distances ( $>50\text{km}$ ), but with less reliability. Transmission at these distances relies on tropospheric ducting, a phenomenon that is not always present. To obtain high reliability long distance transmission, we must move further down the electromagnetic spectrum, to the HF ('high frequency') band.

## The Ionosphere

Radio waves in the HF band propagate mainly by two means: ground-wave and sky-wave. Ground-wave, as the name suggests, travel along the surface of the earth. Ground-wave signals are attenuated as they travel along the earth's surface, limiting their distance. Sky-waves however, refract off a charged region of the earth's atmosphere called the ionosphere, and can travel extremely long distances.

The ionosphere is a complex and ever-changing layer (actually multiple layers) of ionized particles that surrounds our planet. It consists of multiple layers, given the letters D, E and F. The D-layer, which is the lowest and is strongest during the day, attenuates (absorbs) RF energy below a certain frequency. The E and F layers, however, *reflect* RF energy below a certain *critical frequency*,  $f_c$ . This property enables HF radio signals to be 'bounced' off the ionosphere, even multiple times, to communicate over long distances. Losses from the D-layer, and the critical frequencies of the E and F layers determine the Lowest Usable Frequency (LUF) and Maximum Usable Frequency (MUF) for long distance communication. Ionospheric prediction services, such as the one provided by the Australian government<sup>[3]</sup>, provide tools to predict these frequencies, and hence determine the optimal frequency

for reliable communication.

Another consideration is the angle at which a radio wave interacts with the ionosphere. If a radio signal is aimed at the horizon it will hit the ionosphere at a shallow angle, resulting in very long propagation distances, possibly thousands of kilometres. For shorter distances the signal is aimed close to vertical, allowing much shorter hops, perhaps a few hundred kilometres. This method of transmission is called ‘Near Vertical-Incidence Skywave’ (NVIS) and is the primary method this project will use to transmit data.

## Project Applications & Aim

*Design and build a low power short wave data transmitter that can translate low rate data into a modulated short wave output with sufficient power to transmit the data over several hundred kilometres on an Antarctic communication path.* - Original Project Aim

This project was originally intended to be used by a researcher stationed in Antarctica, hence the project name. Throughout the course of the project, lack of information from the researcher prompted a widening of the project’s scope to include other telemetry applications. Examples include:

- Use in the Australian outback for energy research.
- Use on remote islands, for weather monitoring.
- As a telemetry system for a high-altitude weather balloon.

To keep the spirit of the original application it was decided to keep the constraints which Antarctic operation required. These were:

1. Must be able to operate at extremely cold temperatures ( $< -30^{\circ}\text{C}$ ), such as those commonly experienced in Antarctica.
2. Must be very energy efficient, to enable operation from battery power for long periods.

These constraints formed the guidelines for the design and construction of a prototype transmitter, which was successfully tested and then flown as a payload on a high-altitude weather balloon.

## 2 Hardware

### 2.1 CPU

The main CPU of the HF transmitter is an Atmel AT-XMega128A1

## 2.2 Signal Generator

## 2.3 Pre-Amplifier

## 2.4 Class E Power Amplifier

# 3 Software

## 3.1 Hardware Libraries

# 4 Testing

# 5 Project Management

# 6 References

- [1] Iridium Communications Inc. “Iridium Satellite Call Plans” <http://www.iridiumphones.com.au/Call%20Plan%20Brochure%20-%20Post-paid.pdf>, 2010
- [2] ACMA Register of Radio-communications Licenses, Met Bureau Site near Corkscrew Rd Montacute [http://web.acma.gov.au/pls/radcom/assignment\\_search.lookup?pACCESS\\_ID=1320447&pDEVICE\\_ID=1316356](http://web.acma.gov.au/pls/radcom/assignment_search.lookup?pACCESS_ID=1320447&pDEVICE_ID=1316356)
- [3] Australian Bureau of Meteorology, “IPS Online HF Network Frequency Selection Tool” [http://www.ips.gov.au/HF\\_Systems/7/1/10](http://www.ips.gov.au/HF_Systems/7/1/10), 2010

# Appendices

## A Software

All software for this project has been released under the GPLv3 license.

## A.1 The Arduino Project and the XMega

The Arduino is an

open-source electronics prototyping platform based on flexible, easy-to-use hardware and software.<sup>1</sup>

Since many users of the Arduino haven't had much experience in programming, many libraries have been created to fulfil various needs. It's fairly common to begin working on a bit of code to drive some chip, to find that someone has already written a library a few months previously. To quickly get sections of the XMega's codebase working, it was decided to port certain Arduino libraries to the XMega platform.

Arduino's IDE uses the 'wiring' programming language. Wiring is a 'C-like' language, following most of C's syntax, to the point that the Arduino IDE uses a collection of C++ libraries to 'convert' wiring to C++ for compilation. While these libraries could be ported to the XMega allowing usage of the Arduino IDE for this project, this would have required a lot of work and been out of scope. Instead, the Arduino libraries which were needed (OneWire, DallasTemperature, TinyGPS) were individually ported. This involved replacing various Arduino-specific function calls with generic AVR-C calls, and some other minor code changes.

---

<sup>1</sup><http://arduino.cc/>