AFRL-IF-RS-TR-2002-248
Final Technical Report
September 2002

DISTRIBUTED POSITION LOCATION

AEther Wire & Location, Incorporated

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. G333

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-248 has been reviewed and is approved for publication

APPROVED: Richard C. Butler
RICHARD C. BUTLER
Project Engineer

FOR THE DIRECTOR:
WARREN H. DEBANY, Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
SEPTEMBER 2002

3. REPORT TYPE AND DATES COVERED
Final Jul 98 – Jul 01

4. TITLE AND SUBTITLE
DISTRIBUTED POSITION LOCATION

5. FUNDING NUMBERS
C - F30602-98-C-0184
PE - 63739E
PR - G333
TA - 00
WU - 01

6. AUTHOR(S)
Robert Fleming, Cherie Kushner, Uday Nandiwada, and Vincent Coli

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AEther Wire & Location, Incorporated
5950 Lucas Valley Road
Nicasio California 94946

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203

AFRL/IFGA
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2002-248

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Richard C. Butler/IFGA/(315) 330-1888/ Richard.Butler@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

AEther Wire had, prior to this project, developed the base technology to radiate and receive ultra-wideband signals. During this project, AEther Wire advanced this base technology to demonstrate a network-based localization system with eight nodes based on our fourth generation Localizer. Information regarding the embedded software, hardware development, antenna design, and integrated circuit design are presented in this report. Specific tasks included:

1. Development of networking, control, and signal processing software for AEther Wire's fourth generation Localizer.

2. Research and development of improved transmit antennas and drivers, receive antennas, and amplifiers for ultra-wideband signals.

3. Revision of custom Driver and Receiver chips to implement circuit enhancements and accommodate plastic packaging.

4. Exploration of the design of a single chip solution using System-on-a-Chip (SoC) methodology.

14. SUBJECT TERMS
Ultra-Wideband, UWB, RF, Location, Position, Wireless Networks

15. NUMBER OF PAGES
34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

TABLE OF FIGURES

INTRODUCTION

Knowing position location is of immense value to many military and commercial applications. The significant cost and size reduction of GPS receivers brought this fact into focus in recent years. Currently available localization systems give absolute position on the geoid (*i.e.* GPS), location relative to fixed beacons (*e.g.* LORAN), or location relative to a starting point (*i.e.* inertial platforms). For most applications, what is really desired is location relative to other people or objects, whether moving or stationary, or the location within a building or an area. Since these requirements are not met by existing systems, Æther Wire has developed a position location and communication system that fills this gap and opens up new applications. The range and resolution of the position location are proportional to the scale of the objects being located, thus enhancing the utility.

Æther Wire's system provides relative position location within a network of RF transceivers (Localizers) distributed in the environment. Our technology is capable of localization to *centimeter* accuracy, and unlike GPS, can operate within buildings, urban areas, or forests. In addition, Localizers inherently share position location information throughout the network, while most other localization systems require a separate communication channel.

Position location is determined by sharing range information within a network of transceivers that resolve their separation by *cooperatively* exchanging an electromagnetic signal. The accuracy of this range determination is a function of the **bandwidth** of the exchanged signal. With conventional sinewave technology, the bandwidth of the signal relative to the carrier frequency is very small — at most a few percent using spread spectrum. Æther Wire's Localizers transmit and receive bursts of ultra-wideband signals consisting of many Gaussian impulses of ~1 nsec duration. These baseband signals occupy the frequency range from approximately 100 MHz to 1 GHz without any carrier frequency. Ultra-wideband radiation combines the advantages of having gigahertz bandwidth without resorting to high carrier frequencies. Gigahertz bandwidth gives centimeter range resolution for position location, the possibility of high data rate, and the ability to discriminate multipath signals. Operation at low frequencies gives better propagation characteristics, like the ability to penetrate walls, and the opportunity to use slower, cheaper (*i.e.* CMOS) circuits. Also, ultra-wideband radiation does not suffer from the deep fading nulls (~30 dB) that plague sinewave signals in the presence of multipath.

The most significant aspect of Æther Wire's technology is the level of integration that can be achieved. Localizers can be totally integrated in CMOS (Figure 1), because they require less aggressive technology and no reactive components. Transmission and reception of ultra-wideband signals does not require high-Q inductors nor transistors with f_T's 5 to 10 times the transmission frequency to ensure linearity. The position location capability, coupled with built-in communication mechanism in devices that are very low cost, opens up a host of applications. A sampling of military and commercial applications includes:

- Monitoring large numbers of sensors dispersed over an area to counter nuclear, biological, or chemical threats.
- Geospatial registration for warfighter visualization.
- Survey and construction.

- Keeping track of mines, armaments, equipment, vehicles, etc.
- Keeping track of personal items, such as one's children, pets, car, purse, luggage, etc.
- Inventory control in stores, warehouses, shipyards, railroad yards, etc.
- Safety – Finding fire fighters in a burning building, police officers in distress, or injured skiers on a ski slope.
- Synthesis of large aperture antennas for tight beam communication, using scattered transceivers that know their precise relative location and synchronization.
- Sports – Arbitrating rules in a game, playback of motions for coaching, or viewing the re-creation of an event.
- Home automation – Keyless locks and rooms that adjust the light, temperature, and music sound level.
- Motion pictures – Automatically adjusting camera focus and motion tracking for matching digital effects.

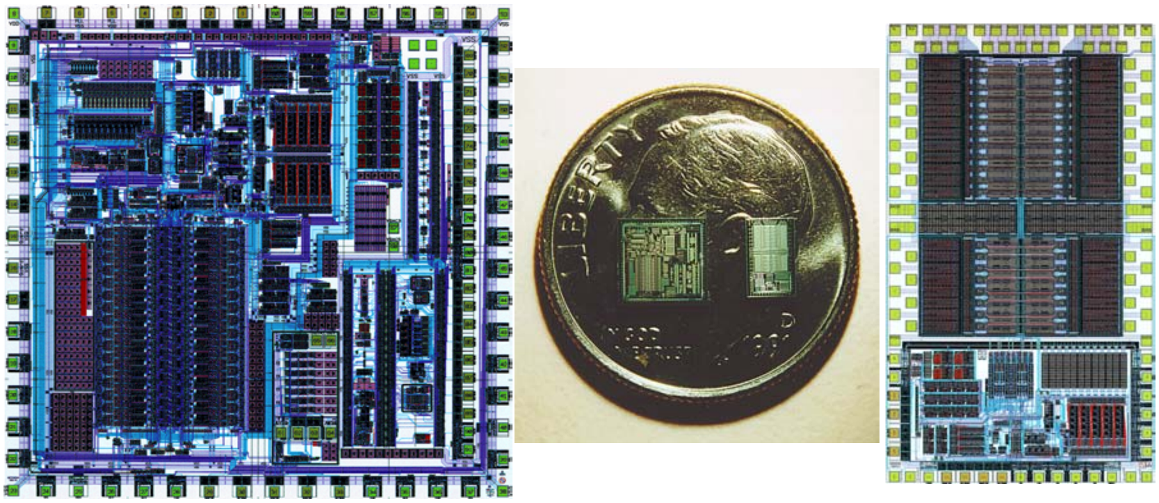


FIGURE 1. Plots of the Aether4 and Driver2 chips, and the actual chips on a dime. Shown on the left is the Receiver/control chip Aether4. Its size is 4.6 x 4.5 mm. The 32 Correlators form the major structure in the lower left quadrant. Above the Correlators are the Phase Locked Loop and Real Time Clock, with the Tx and Rx Linear Feedback Shift Registers to the right. Shown on the right is the transmit antenna driver chip Driver2. Its size is 2.4 x 4.1 mm. The H-Bridge for driving the antenna leads is the major structure filling the top two thirds of the chip.

ULTRA-WIDEBAND SIGNALS

Æther Wire's Localizer system uses ultra-wideband signals consisting of ~1 nsec Gaussian impulses, which have approximately 1 GHz of bandwidth. Important features and benefits of ultra-wideband signals include:

- Most of the energy of a nonsinusoidal 1 nsec wide Gaussian impulse is spread over frequencies below 1 GHz, whereas pulse-modulated sine waves require a carrier frequency of at least 30–60 GHz to get this bandwidth. Expensive microwave (GaAs MMIC) technology is needed for *sinusoidal* transmission at these carrier frequencies to offer sufficient bandwidth. Nonsinusoidal impulses are baseband signals, which means there is ***no carrier frequency***. Hence, they can be easily integrated in CMOS.

- The antennas can be small and can be driven directly by CMOS, because they are non-resonant, current-mode, and low voltage.
- Ultra-wideband signals form a shadow spectrum which can coexist, and does not interfere, with the sinewave spectrum. The transmitted power is spread over such a large bandwidth that the amount of power in any narrow frequency band is very small.
- Ultra-wideband signals do not suffer from deep fading nulls caused by multipath, and the multipath signals can be RAKED together with the direct signal.
- Ultra-wideband signals have very good penetrating capabilities due to bandwidth at low frequencies. Transceivers can operate within buildings, urban areas, and forests.
- Ultra-wideband signals have a tolerance of interference from other radio sources, and inherent privacy from eavesdropping (low probability of intercept).

POSITION DETERMINATION

A Localizer system is illustrated in Figure 2. This system is comprised of an interacting network of Localizers. In this illustration, the position of Localizer D is being determined with respect to the local reference frame. To achieve this, the remaining four localizers must be at known locations with respect to the reference frame. In practice, there could be more than 4 “fixed” Localizers, and the range measurements between pairs of Localizers would have varying degrees of noise. Consequently, the position calculation is performed using least squares techniques very similar to those used in GPS calculations.

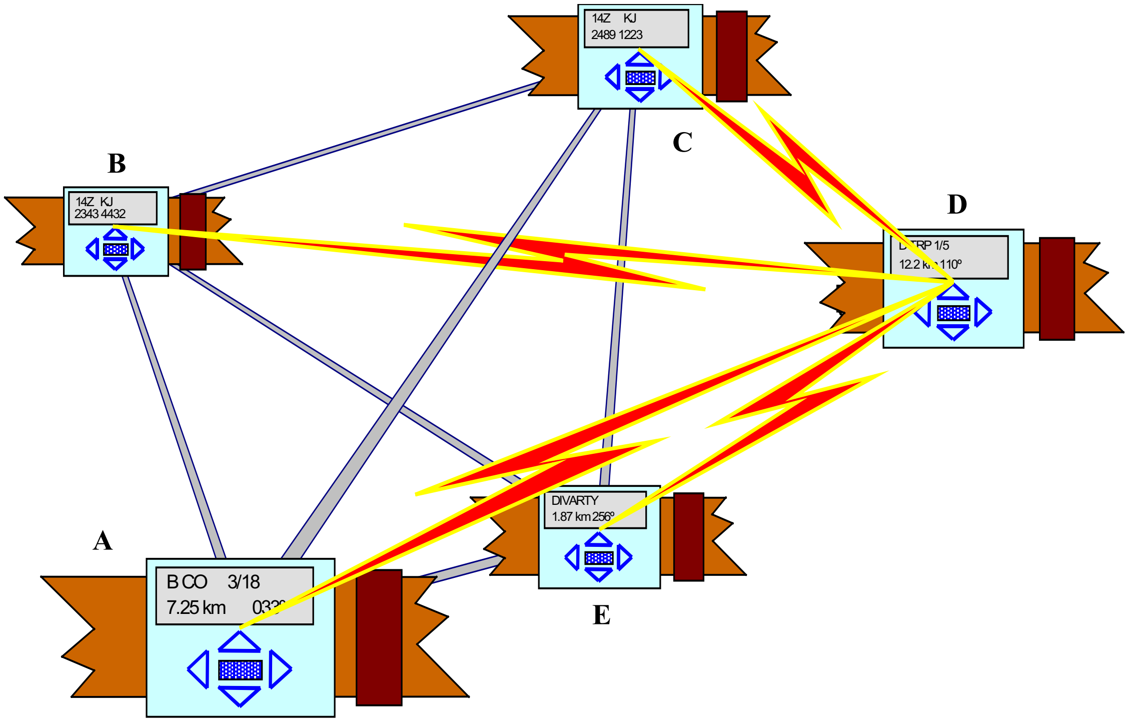


FIGURE 2. This diagram shows a network of 5 Localizers, where each Localizer determines the range to every other Localizer and then shares the information with members of the network. In the diagram, Localizer D is ranging between itself and A, B, C, and E. With 4 Localizers knowing all the ranges between them, a rigid tetrahedral structure is determined (assuming no 3 Localizers are collinear). Each Localizer can then be in one of two 3-dimensional locations with respect to the other 3 Localizers. Having a 5th Localizer resolves this ambiguity.

OVERVIEW OF LOCALIZER TECHNOLOGY

The design of Æther Wire's Localizer system is based on the following key features:

- Episodic transmission of coded sequences of Impulse Doublets.
  A typical 1023-sequence of 10 nsec chips lasts 10 µsec.
- Coherent reception using Correlation for Channelization and Time Compression.
  Typical Process Gain with a 1023-sequence is 30 dB.
- Post-processing of Correlation Patterns.
  High speed processing is done in hardware, and complex processing done in software.
- Precise timing for Cooperative Ranging and Aggregate Network Capabilities.

The resolution for scheduling of transmissions and receptions is 10 psec, which allows sub-centimeter ranging. The Consensus Clock within a network of Localizers achieves part-per-billion relative stability.

In the following paragraphs, we elaborate on the concepts used in the Localizers and describe the hardware platform.

CODE DIVISION MULTIPLE ACCESS

The Localizer system makes use of Code Division Multiple Access (CDMA) to allow multiple Localizers to communicate at the same time in the same area. The Gaussian impulses are transmitted in a coded sequence, which are pseudo-random codes like those used for direct-sequence spread spectrum systems such as GPS. Different sequences provide separate channels roughly equivalent to frequency bands.

Correlation is used to discriminate a particular code sequence from other signals, both nonsinusoidal and sinewave frequency-based. When a coded sequence of Gaussian impulses is received by a Localizer, it is compared to a locally generated reference sequence in a Time-Integrating Correlator. The output from the correlator is a function of the relative shift between the received signal and the reference signal. When the shift of the input signal matches that of the reference code, a correlation peak is produced. Consequently, the separation distance between Localizers can be determined from the time domain output of each correlator.

Ideally, codes should be chosen so that the correlation of a code sequence against itself will have a single peak, making it easy to determine when the proper sequence has arrived. Maximal Sequence codes and Complementary codes approach this ideal. The cross-correlation of one code sequence with a different sequence should *not have correlation peaks* in order for multiple Localizers to operate at the same time. Fortunately, families of codes exist with tens of thousands of members that have both good autocorrelation and good cross-correlation properties. One important advantage of this correlation-based detection is that the receiver is able to detect signals that are buried in high levels of random noise. Consequently, the power requirements for the transmitters are not significant. Furthermore, the equipment is fully functional in high-noise environments such as cities.

DOUBLETS

A series of impulses can be launched by stepping the current through the transmit antenna up or down. The minimum total current occurs when, starting from zero current, a negative impulse immediately follows a positive impulse, and vice versa. We refer to such pairs of impulses as “doublets” shown in Figure 3. Doublets can be generated from a single supply by using switching circuits to control the direction of current flow through the transmit antenna (Figure 7).

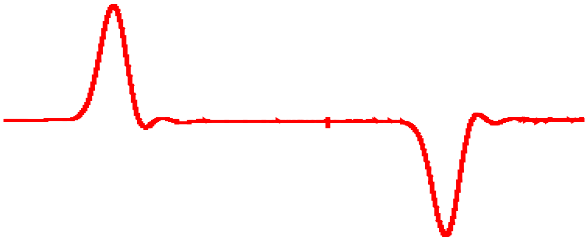


FIGURE 3. A Gaussian impulse Doublet, typically with 5 ns impulse spacing.

For each **bit** in any code sequence, we generate a doublet that starts with either a positive impulse or a negative impulse. An example for a sequence encoded with doublets is shown in Figure 4. In spread spectrum terminology, a doublet is our “**chip**”. If the autocorrelation of a sequence of impulses has a single correlation peak, then the autocorrelation of the same sequence encoded using doublets has a central peak bracketed by two negative peaks (Figure 5). This complex pattern is much easier to recognize than a single peak, especially when the signal is corrupted with noise.

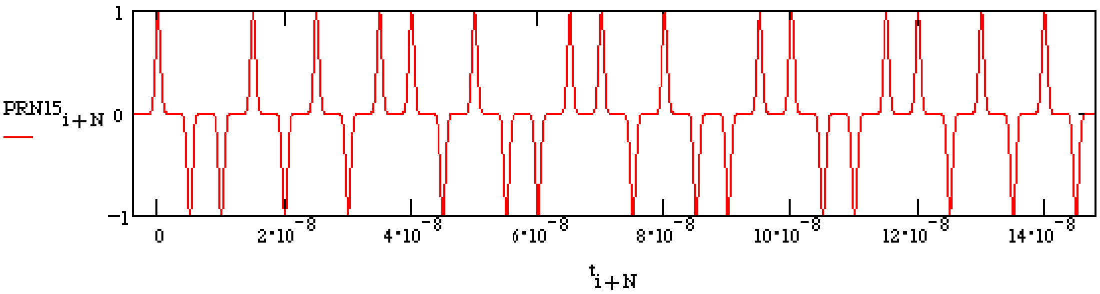


FIGURE 4. 15-bit Maximal Sequence “100011011010111” encoded using Doublets. Here the impulses are 1 nsec wide and separated by 5 nsec. Thus, most of the energy is below 1 GHz, with nulls every 200 MHz. Note that we use much longer impulse code sequences than this; typically sequences of 1023 doublets (2046 impulses) from families with excellent cross-correlation and autocorrelation properties.

TIME-INTEGRATING CORRELATOR

We have chosen to use the *dual form* of the usual Sliding Correlator implementation, known as a Time-Integrating Correlator (TIC). The reference code sequence is shifted past the changing analog input signal and the product of the code and signal is summed in a set of analog integrators. The output of each integrator represents a different alignment (“phase”) between the reference code sequence and the input signal (Figure 5). The outputs of the TIC correspond nearly to the sampled output of a Sliding Correlator as a function of time.

A key feature of a Time-Integrating Correlator is that the process of detecting a correlation peak does *not* have to be done in real time. The correlation results are saved in the outputs of the integrator phases, which we then digitize. The complex TIC *patterns* that result can subsequently be analyzed with a microprocessor over a substantial period of time. This allows sophisticated pattern recognition processes to be used, such as neural nets and maximum entropy.

CLOCKS AND TIMING

Localizers can use *ordinary crystal oscillators* for two reasons:

- Localizers can exchange signals even if their clocks differ, as long as the differences do not accumulate over the short duration of an exchange. A good crystal oscillator (about 1 part-per-million accuracy and stability) is more than sufficient.
- In the process of exchanging signals, cooperating Localizers can determine the amount by which their clocks differ, and can compensate for the different clock rates. This is possible because the short-term stability of a crystal oscillator can be much greater than its absolute accuracy.

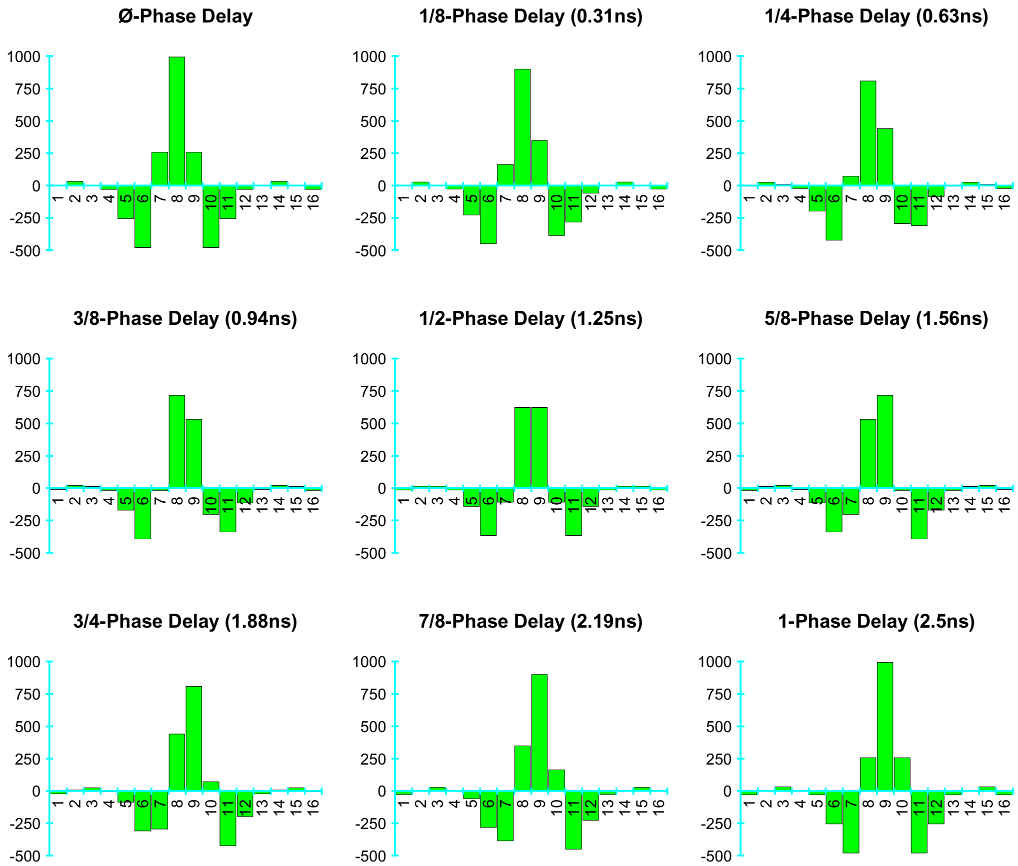
For example, assume two Localizers have determined the relative clock rate of each other's clocks. Each Localizer can then measure range with a fractional error given by the fractional error in its clock during an exchange of signals. Within a network of Localizers, only one Localizer needs to have a very accurate clock for other Localizers to absolutely calibrate their clocks. Independently knowing the distance between two Localizers can also be used for calibrating the clocks of all communicating Localizers.

Even with a perfectly accurate clock, a Localizer will have unknown circuit delays that will affect the measured time-of-flight delay for a signal. As long as these delays are relatively stable, they can be measured and factored out of the range computation. This self-calibration technique requires a Localizer to receive the same signal it sends. All the measured delay will then be due to circuit delays.

RAPID SYNCHRONIZATION

Localizers use correlation for reception of different code sequences in a manner analogous to filtering for reception of different sine wave frequencies (*i.e.* channels). The code sequences are pseudo-random codes like those used for direct-sequence spread spectrum systems such as GPS. Families of codes exist that have both good autocorrelation and cross-correlation properties, such as Kasami codes. We have discovered a technique for constructing a Beacon code that has low cross-correlation with other members of an orthogonal code family. Moreover, the Beacon code has special autocorrelation properties that allow for a ~30 times speedup in the process of synchronization.

The auto-correlation of the Beacon signal has multiple sidelobes with a triangular shape to the envelope (Figure 10). This allows a Listener to search for a Beacon signal by shifting its reception window in steps of 10 μsec instead of 40 to 80 nsec. If it detects any of the sidelobe peaks, it can search for nearby sidelobe peaks and use the shape of the envelope to find the central peak. A Listener and a Pinger Localizer can establish a duplex communication using just the Beacon code. Then they switch to one of the other codes in the family. Note that this technique is comparable to the protocol of using certain hailing frequencies for maritime radio.


Ø-Phase Delay

1/8-Phase Delay (0.31ns)

1/4-Phase Delay (0.63ns)

3/8-Phase Delay (0.94ns)

1/2-Phase Delay (1.25ns)

5/8-Phase Delay (1.56ns)

3/4-Phase Delay (1.88ns)

7/8-Phase Delay (2.19ns)

1-Phase Delay (2.5ns)

FIGURE 5. Time-Integrating Correlator output Phases. This diagram shows the evolution of the pattern of output phases of the Time-Integrating Correlator as the alignment of the received signal and reference code shifts in increments of 1/8-phase. These are for a 31-doublet maximal sequence with phases (*i.e.* time shift bins) separated by 2.5 nsec, and each graph representing an additional 312 psec relative time shift between received signal and reference code. The pattern differences are quite easily detected using neural networks in software. Note that a time shift of 2.5 nsec produces the same pattern in the lower right graph as in the upper left graph, except the pattern is centered in bin 9 instead of bin 8, (which represents a shift of 8•312 psec = 2.5 nsec).

RANGE DETERMINATION

As previously illustrated, the fundamental problem in determining position is to obtain a measurement of the range between Localizers in the system. The range between any two Localizers is determined by measuring the round-trip transit time for a signal and multiplying by the speed of light. Figure 14 shows a typical ranging transaction. Ranging requires sending an encoded signal from a first Localizer to a second one, and then getting a different encoded signal back. Note that unlike radar, the return signal is not an echo, but a retransmission of a new code.

HARDWARE

Æther Wire has taken a systems approach to the design of Localizers. Only the most time-critical and processing-intensive functions are performed in hardware, and most of the system complexity is handled in software. As much as possible, the sequential operation of the hardware is controlled by a general purpose processor. This way, the details of operation can be worked out over time, and can be quickly changed if there are unanticipated problems. For transmission, the basic task is *sending coded sequences of Gaussian impulses at precise times*. For reception, the basic task is *correlating the antenna signal against a known code sequence* with precise alignment in time. A Block diagram of the Localizer hardware is shown in Figure 6.

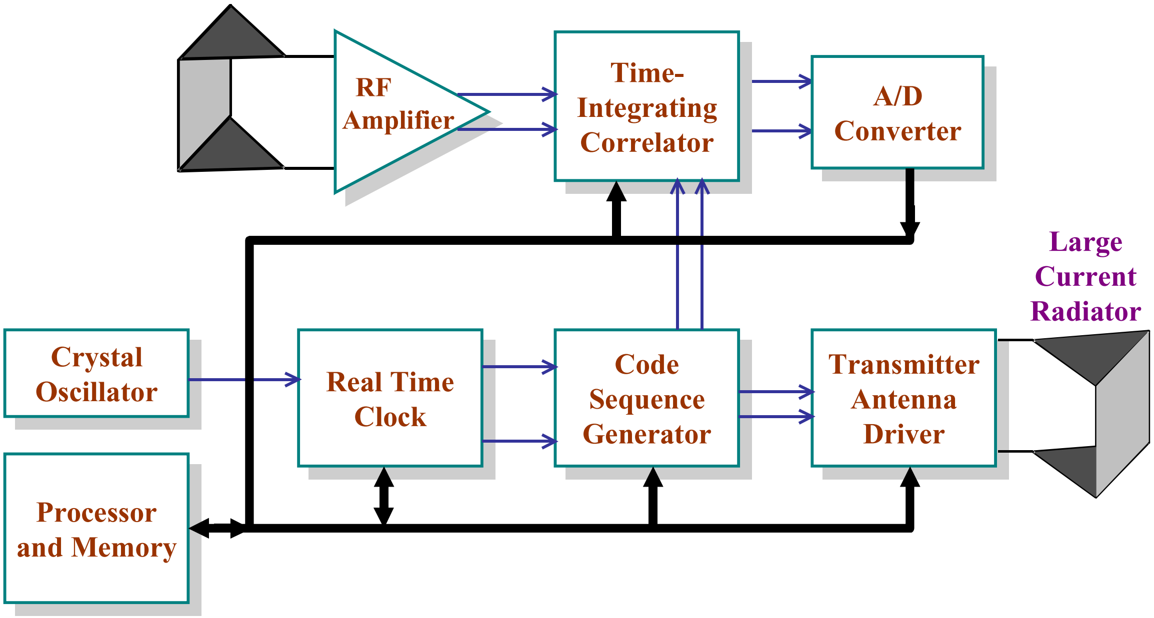


FIGURE 6. Block diagram of the Localizer hardware. For transmission, the basic task is *sending coded sequences of Gaussian impulses at precise times*. For reception, the basic task is *correlating the antenna signal against a known code sequence* with precise alignment in time.

RECEIVER SECTION

The receiver section includes the receiving antenna amplifier, the Code Sequence Generator, and the Time-Integrating Correlator (TIC) with 32 integrators. The receiving antenna amplifier and the Code Sequence Generator feed the Time-Integrating Correlator (TIC) with the amplified received signal and the reference code respectively.

The same code sequence is fed to each integrator in the TIC, but delayed by a different amount for each "phase". The analog received signal is multiplied by the delayed digital code sequence and summed at the output of each integrator. (A phase corresponds to an integration window 5 nsec wide, but the phases are spaced 2.5 nsec apart, so that their windows overlap.) Each integrator output is sampled by a Sample-and-Hold (S/H) circuit at the end of the code sequence. The outputs of the S/H circuits are multiplexed onto a common analog bus to be digitized by the

A/D converter, which is read by the processor. Thus, after a reception event, the processor has 32 values that are equivalent to sampling the output of a sliding correlator during an 80 nsec window.

ANTENNA

The development of the ultra-wideband Large Current Radiator (LCR) antenna by Dr. Henning Harmuth has made it possible to radiate nanosecond wide impulses with inexpensive CMOS chips. The LCR is a current-mode antenna which radiates outwards from the surface of a flat square conductor. The only restriction on its size is that it cannot be made *larger* than the equivalent width of the impulse being transmitted.

Applying a step change in the current through an LCR causes an impulse to be radiated, since radiation power launched is proportional to the square of the *derivative* of current flow. The impulse is narrower and has more radiated power when the current can be changed more quickly. The sign of the derivative of current determines the polarity of the impulse. Thus, turning **off** the current through an LCR generates an impulse which has the opposite polarity of the impulse generated when the current is turned **on**. The transmitter section uses the same circuitry as the receiver section for generation of the code sequence at a nominal 100 MHz chip rate (200 MHz Gaussian impulse rate).

TRANSMITTER SECTION

The antenna driver section of the transmitter looks like a standard H-bridge, as is commonly used to drive stepping motors (Figure 7). Each bit in the code sequence determines whether the current initially flows one way or the other through the bridge. The current through the antenna is turned **on** by connecting opposite sides of the antenna to V_{DD} and V_{SS}. The current is turned **off** by connecting both sides of the antenna to the same supply, V_{DD} *or* V_{SS}, which means there is always a closed path for current to flow. In other words, we generate a step change in current through the transmit antenna by causing a step change in voltage across the antenna. Each of the four arms of the 'H' bridge has two programmable delay elements (one for each edge) to adjust making and breaking the switch connection in the respective output transistor. This can compensate for the different switching delays of the P-channel versus N-channel transistors, and for other circuit mismatches. Each arm is also divided into multiple (≥ eight) sections which can be individually enabled for power control.

REAL TIME CLOCK

Localizers can use *ordinary crystal oscillators* because: a) Localizers can exchange signals even if their clocks differ over the short duration of an exchange; and b) Cooperating Localizers can determine the amount by which their clocks differ, and can compensate for the different rates. We achieve 10 psec timing resolution without using a 50 GHz clock. A stable crystal oscillator constantly clocks the coarse resolution part of a Real Time Clock counter, and a Phase-Locked Loop (PLL) derived from this provides the 200 MHz clock. The 5 nsec period of this oscillator is further divided by selecting a tap around the ring, and by using a programmable delay generator with 10 psec (or finer) resolution.

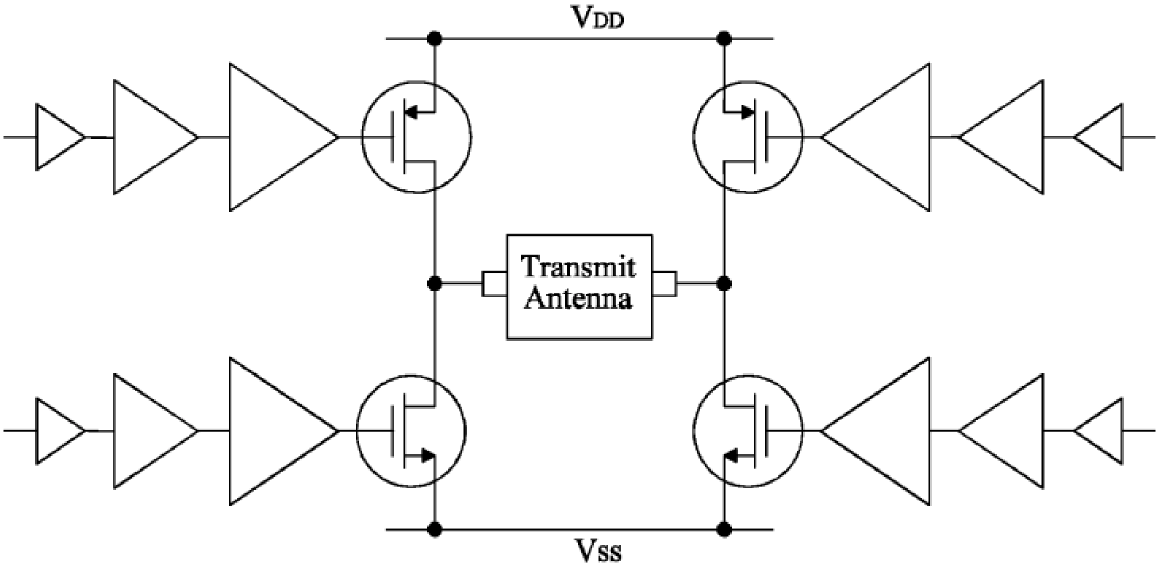


FIGURE 7. Block diagram of the Transmit antenna drivers. A typical section is shown in detail to illustrate the 'H' bridge configuration of output transistors which drive the transmit antenna. There are eight to twelve sections which can be separately enabled for adaptive power control. A step change in current is generated by causing a step change in voltage across the antenna. The current can be turned on in either direction through the antenna, and turned off by switching both sides of the antenna to either VDD or VSS. An electromagnetic impulse is generated for each on and an opposite polarity impulse for each off.

CODE SEQUENCE GENERATOR

The same type of pseudo-noise (PN) code sequence generator is used for both the transmitter and receiver sections. The design is a fully programmable 25-stage Linear Feedback Shift Register (LFSR). It is a universal code generator capable of generating Maximal Sequences, Kasami Small codes, Kasami Large codes, Gold and Gold-like codes, and more. To generate a code sequence, the processor initializes the generator with the seed value, the polynomial, and the sequence length.

PROJECT ACCOMPLISHMENTS

Æther Wire had, prior to this project, developed the base technology to radiate and receive ultra-wideband signals. During this project, Æther Wire advanced this technology to demonstrate a network-based localization system with eight nodes based on our fourth generation Localizer. Information regarding the embedded software, hardware development, antenna design, and integrated circuit design are presented in this report. Specific tasks included:

- Development of networking, control, and signal processing software for Æther Wire's fourth generation Localizer
- Research and development of improved transmit antennas and drivers, receive antennas, and amplifiers for ultra-wideband signals
- Revision of custom Driver and Receiver chips to implement circuit enhancements and accommodate plastic packaging
- Exploration of the design of a single chip solution using System-on-a-Chip (SoC) methodology

During the course of this project, several improvements were made to the hardware and software. From our experience with the design of early versions of Localizers, we realized that we needed the control and discipline offered by a Real Time Operating System (RTOS). Hence, the emphasis has been on the design of structured software based on an RTOS. We ported the basic Localizer design to the new hardware and software platforms. We designed, simulated and implemented a variety of control and communication algorithms for Localizers. Each of the algorithms was tested on the prototypes and validated. The software was developed with self-configuring and distributed networking in mind. Some of the important results were in the field of pseudo-random code sequences with special properties. We addressed the long-term clock drift of the individual localizers and its effect on the communication between Localizers.

On the hardware side, we fabricated newer versions of the Æther Wire chips and assembled prototypes based on these new designs. Since the antenna is a very important component of a nonsinusoidal system, the antenna radiation patterns were thoroughly investigated. We combined our experience in antenna design with the state-of-the-art simulation tools using Finite Difference Time Domain (FDTD) techniques. As a result, we developed antennas with enhanced radiation characteristics. Starting with a network of two Localizers, we established and maintained communication between them to prove the cooperative networking concept. The concept was then extended to a larger set of Localizers.

Several demonstrations and presentations were made describing the progress made during the research.

LOCALIZER SOFTWARE DEVELOPMENT

The software was designed to execute under the μC/OS real-time kernel running on a Motorola ColdFire MCF5204 processor. The software is written primarily in the C programming language with the time-critical tasks coded in assembly language. The control software for the Localizer was designed with a top-down approach. The control software supervises the overall initialization, acquisition, network communications, and ranging for each Localizer. The software includes:

- Control logic
- Synchronization algorithms
- Ranging and communication algorithms
- Signal detection and identification
- Kalman filter for clock tracking
- Enhanced debug and calibration monitor
- PC-based real time display of Localizer positions
- Error control coding to improve the BER
- Low-level software modules for the ColdFire processor

The salient features of the software development are described below.

SYSTEM LEVEL TASKS

The system software for the Localizer is organized in terms of tasks. Each task has a different priority level. The processor executes the tasks on the basis of the priority level. Each task may consist of several processes which execute at the priority level of that task. Of these tasks, only the epoch time management (for the time line) is of higher priority than the message exchange task. The priority for the tasks is as shown below:

	Priority	Description
High	5	Epoch Time Management
	10+	Node Message Exchange
	45	Node Search
	46	Input Message Management
	47	Output Message Management
	48	Node-to-Node Messages
Low	49	Lab Data Display

Epoch time management is the high priority task that updates the timeline. The epoch number and start time are updated every epoch. All message exchanges are relative to the start of the epoch and are based on the calculated receive (Rx) times (The Rx operation is the first operation in every epoch). The timeline for the Localizer is explained in the next section.

Node Message Exchange task is responsible for the exchange of information with all nodes in contact. This executes as a single task with multiple copies at different instants of time, one for each node in contact. Ranging operations and general message interface are managed by this task. The high priority messages are executed at the priority level of this task. Low priority messages are processed by the Input Message Management task. This task is explained in detail elsewhere in the report.

Node Search is the task responsible for searching for nodes. This is a low priority task. Since the RTOS allows only one task per priority level, successive searches execute with decreasing priority. This is an important part of the Localizer software. A detailed description of this task is given later in the report.

Input Message Management task processes the low priority messages that are not processed by the node message exchange task. This task decodes and assigns a priority to the message and places it in the node's input message queue.

Output Message Management task is a low priority task which collects information about all other nodes that are in contact and generates the messages to be sent out. These messages are placed in an output message queue. The Message Exchange task sends these messages out if no other higher priority messages are waiting.

Node-to-Node Messages is a task that allows an operator to send text messages to any node in the network. This task has the necessary logic to support broadcast of text messages.

Lab Data Display task allows the background transmission of data to a PC for display. The data transmitted to the PC includes the correlator output for one bit exchange, information about the nodes that are in contact, and the Pinger/Listener status. The data sent to the PC also includes the ranges between pairs of Localizers for displaying a 3-D view of the Localizer network.

COMMUNICATION PROTOCOL

The Localizer undergoes an initialization sequence at Power-On and then searches for other nodes within its cluster. The initialization process consists of the following steps:

- initialization of the operating system
- estimation of the background noise
- initialization of the Localizer time line
- search for other nodes within the cluster

The startup process initiates the Localizer correlators and calibrates them. Since the correlator outputs drift with time, the average correlator offset is determined to compensate for this drift. Next, the background noise for all communications codes – the small Kasami codes, the beacon code, an extra Kasami code, and maximal sequence codes – is measured. Then, the operating system tasks are initialized, and the Localizer time line is started.

The Localizer timeline is as shown in Figure 8. The timeline is divided into eras. Each era comprises 16 packets and each packet is divided into 32 epochs. An epoch has 32 time slots of 31.25 µsec each. The 31.25 µsec slot accommodates one 10 µsec transmit pulse train, one 10 µsec receive pulse train, speed of light delay time, and guard time. Communication between Localizers takes place during the 10 µsec intervals. There is a guard epoch on either side of the packet structure. In addition, one epoch is reserved for the initialization of the Localizer Shift Registers – the code seed value, polynomial, and sequence length. Hence, the standard packet length for information exchange between nodes is 29 bits. A Localizer transmits a code sequence for each bit of the message to be exchanged during the 10 µsec transmit (Tx) slot in each epoch. The single bit exchange continues until the message exchange is completed. Between bit exchanges, the Localizer correlators are powered off to conserve power.

A Localizer has two distinct functions which it can perform in each epoch – the node search (acquisition) process or the information exchange process. Other miscellaneous functions such as background noise estimation are lower priority tasks. Each packet in the packet table is reserved for one of the two primary tasks. At Power-On, all the packets are available for the node search process. In addition, each node randomly configures itself to be a *Pinger* or a *Listener* at Power-On. After a prescribed period of time, the nodes reassign themselves to the other function and continue the search process until another node is found. Once a node is acquired, they negotiate to select a suitable packet from the packet table for further communication. All other available packets may be used for searching.

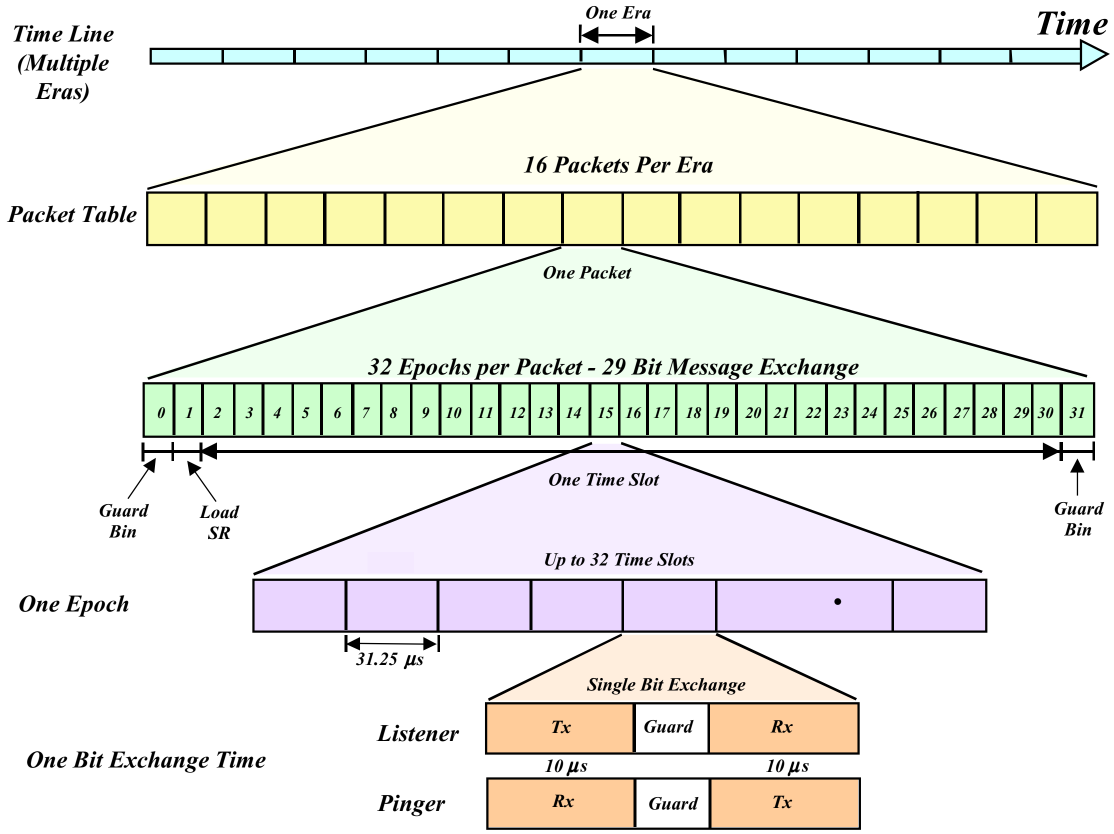


FIGURE 8. The Localizer timeline is described in this figure. The Localizers exchange one packet every era. The initialization and the guard bins take three epochs every packet. The communication is based on TDMA and, within a time slot, on CDMA. Within every time slot, there is a transmission and reception with guard time in between.

NODE SEARCH PROCESS

Acquisition of other nodes is required before any ranging or communication can be attempted with those nodes. As can be seen from the description of the timeline, the communication between Localizers is episodic with a low duty cycle. This implies that the Listener has to scan through time until it acquires the Pinger. This is a time consuming process because of the low duty cycle of transmissions. Hence, the acquisition process is a crucial part of the Localizer scheme.

A Localizer configured as a Pinger transmits a known beacon code for 10 µsec every epoch. Once the Localizer attempting acquisition (the Listener) detects this beacon code, it transmits the beacon code back to inform the Pinger of its presence. The Pinger, which always listens before a transmission, will acknowledge this communication from the Listener once it receives it. This closes the loop between the Pinger and Listener.

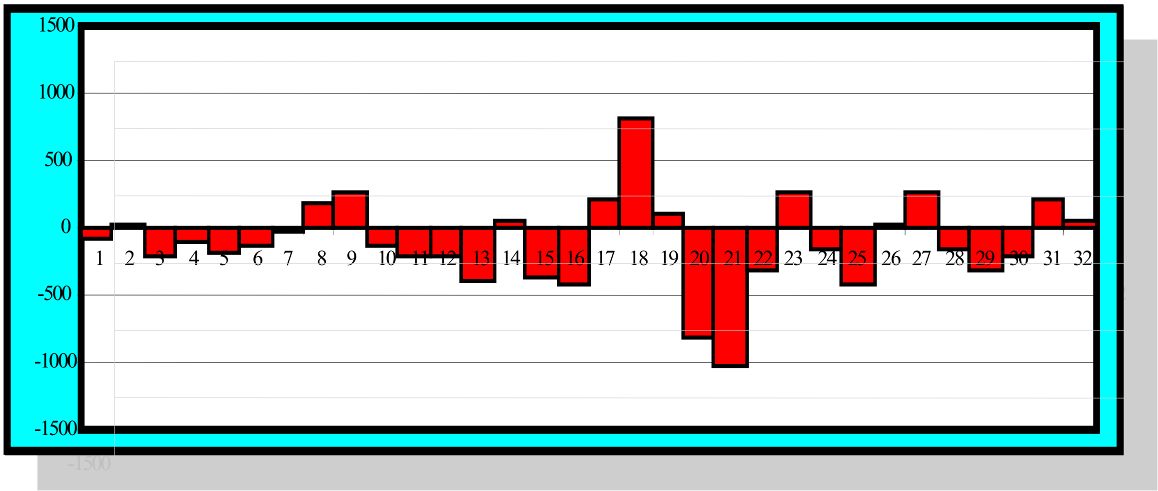


FIGURE 9. The 80 nsec correlator reception window shows the output of a correlator.

RAPID ACQUISITION SCHEME

Localizers, like other spread spectrum radios, have to find the precise alignment of the received signal with the reference sequence to detect a correlation peak. Without any prior knowledge or an alternate back-channel, a spread spectrum receiver has to search through all possible times that a transmission may occur. Using exhaustive search, the worst case time required for synchronization grows as the square of the transmission repetition interval. Thus, a repetition interval ten times as large requires 100 times the acquisition time. Also, the acquisition time is proportional to the fraction of each interval that the receiver can examine (*i.e.* 100 repetitions are required if 1% of the transmission interval can be examined each time). For low duty-cycle communication, this can lead to unacceptably large acquisition times.

Localizers need a rapid acquisition technique because the transmissions / receptions are spaced apart (episodic), and the correlation window spans a small fraction of the interval between transmissions. The latter limitation comes from what can be practically integrated on-chip. The reason Localizers use episodic communications is to support very low power operation and precise position location in a multipath environment.

Extremely low power consumption for distributed unattended sensor networks ultimately comes from low duty-cycle episodic operation. This means reception as well as transmission is powered and active only during predetermined periods.

Accurate ranging for determining precise position location requires detecting the direct path signal sent between Localizers. Other radios typically detect the strongest signal, which can be a multipath reflection. Localizers use episodic transmissions spaced further apart than the typical delay spread. Thus, the multipath from a previous transmission will have dissipated before the next transmission, and the direct path signal will not be obscured by multipath.

The PN (pseudo-noise) code sequences that we use for ranging and communication need to have certain autocorrelation and cross-correlation properties. Specifically, the autocorrelation of a given code must have a single correlation peak with minimal sidelobes, and the cross-correlation with other members in a family can be no higher than the sidelobes of the autocorrelation. Fortunately, Kasami codes have large families that match these requirements. However, these properties mean that a transmission can only be detected if it falls within the correlation window.

In the current generation of Localizer chips, the correlation window is only 80 nsec in length, and consists of 32 correlator bins. Thus, if the Pinger transmitted once every epoch (*i.e.* once every millisecond), then the Listener would need to search (receive) as many as 16,667 times (includes 20 nsec window overlap). Acquisition in this manner would require nearly **17 seconds** (one millisecond per receive).

The germ of the idea for our rapid acquisition process was finding a way to examine a larger fraction of the interval between transmissions with each reception. We imagined that there exists a PN code sequence that has autocorrelation sidelobes that can be detected with only partial overlap of the transmission and reception code sequences. Then the lack of any detectable signal would mean that the reception time could be advanced with a step size much larger than the width of our correlation window. The other important property of this imagined PN code sequence is that its cross-correlation with the codes used for ranging and communication be no worse than other code family members.

After extensive simulations and studies, we discovered a *Beacon* code with the necessary properties! This Beacon code is formed from $2^{n/2}+1$ repetitions of the maximal sequence of length $2^{n/2}-1$, which is one of the generator polynomials for a Kasami code of length $2^{n}-1$. Our application typically uses $n=10$ to yield a code length of 1023 Doublets, or 33 repetitions of a 31-length code. The worst-case cross-correlation of this code with members of the Kasami family is the same as the rest of the family members (which is approximately $N^{1/2}$). Figure 10 illustrates the autocorrelation function of the Beacon code, which is transmitted over approximately 10 µsec, as would be seen by a series of Localizer receptions spanning 20 µsec. Also shown is an expanded view of the sidelobes, which occur at precisely-spaced intervals of 310 nsec. Note that the amplitude of the correlation increases until the received code aligns with the expected code, which is a key component of the rapid acquisition process.

Use of this Beacon code affords the following advantages:

- Low cross-correlation (so other local receivers are minimally affected).
- Significant cyclic autocorrelation sidelobes (for easy detection of the beacon).
- A well-defined autocorrelation sidelobe structure that enables the rapid detection of the signal peak.

Since a sidelobe detection occurs every 310 nsec (with increasing/decreasing amplitude) in a 20 µsec window of time, it is only necessary to receive data for a contiguous 310 nsec period once every 20 µsec. However, since the sidelobes are low at the edges, and may not be detectable through the noise, the search algorithm examines a 310 nsec window once every 10 µsec. Since the correlator window size is 80 nsec, and some overlap is desired, this sidelobe search is accomplished by increasing the time (modulo 1 msec) in 60 nsec increments for five consecutive windows. Figure 11 illustrates the construction of a 310 nsec window using 80 nsec wide correlator windows. Each of these five windows is sampled over five contiguous epochs, which spans 5 msec. The time is then incremented by 10 µsec, and the process is repeated. This search process is shown in Figure 12. Thus, 100 windows, spaced 10 µsec apart, are required as a maximum to detect one of the sidelobes of the Beacon signal. Five receives are performed during each of the 100 windows, resulting in 500 receives by the Listener. Since each receive requires one epoch (1 msec), the maximum time for the Listener to detect the Pinger is **500 msec**. This is faster by a factor of **33** over conventional search techniques.

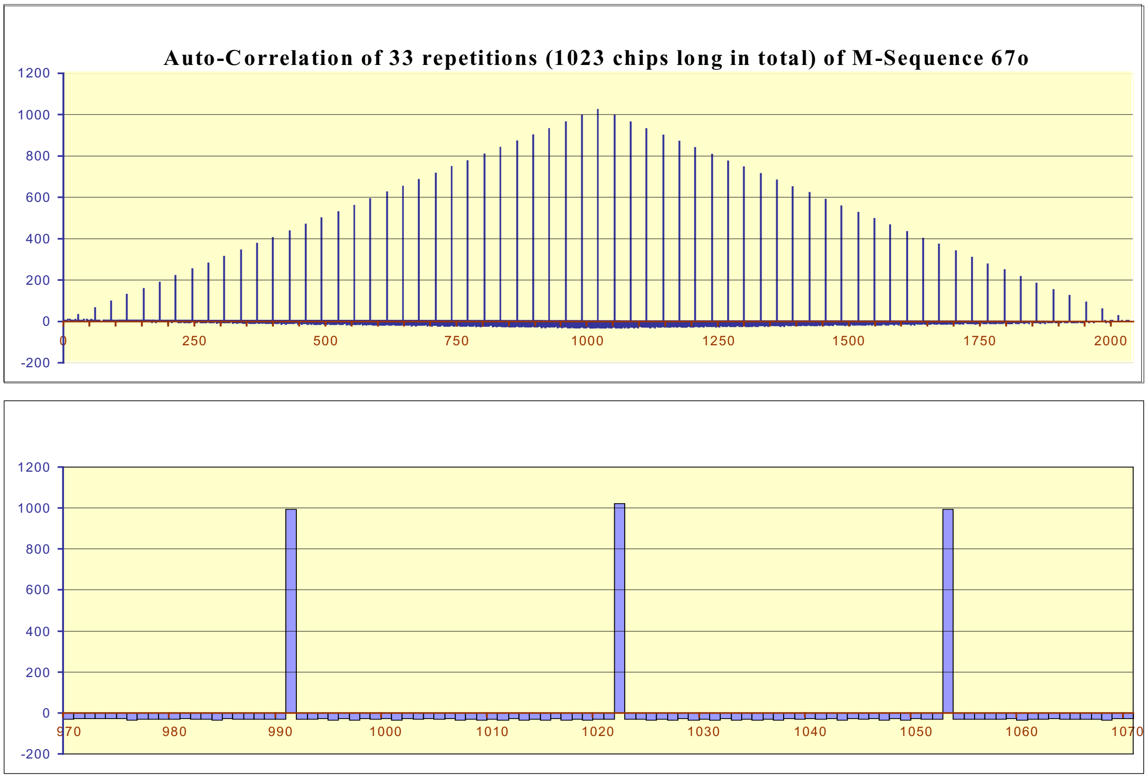


FIGURE 10. The structure of the autocorrelation function for the special Beacon code is shown here. We can see that there are correlation peaks every 310 nsec. The peak amplitudes increase as perfect alignment in time is reached. An expanded view of the 310 nsec period is shown to explain the periodicity of autocorrelation It can be seen that the correlation peaks are very sharp and are well separated from the noise floor. This structure makes it easy to detect the peaks.

Once the Listener finds the exact location of the beacon code, it has to transmit a response which will be received at the exact (advance) time that the Pinger is listening. The transit time for a signal between the Pinger and Listener is unknown because the separation is yet to be determined. So the Listener must scan through the range of possible Rx/Tx delay times while transmitting a ONE in a loop. It starts with a nominal 5 µsec offset between the time of its reception (Rx) and transmission (Tx). During each subsequent loop, it increments this time by 30 nsec. Meanwhile, the Pinger continues to beacon, not knowing that the Listener is trying to send it a bit. When the Listener uses an Rx/Tx delay time close enough to the range between the two nodes, the Pinger makes the first detection. At this time, the Pinger flips the transmitted bit from a ONE to a ZERO. The Listener knows that the Pinger detected the Listener's signal from this polarity change. The Listener and Pinger then go through an exchange process until the Rx/Tx delay is accurate enough to center the detection peak in the middle of the Pinger's correlator window. The node acquisition process is now nearly complete.

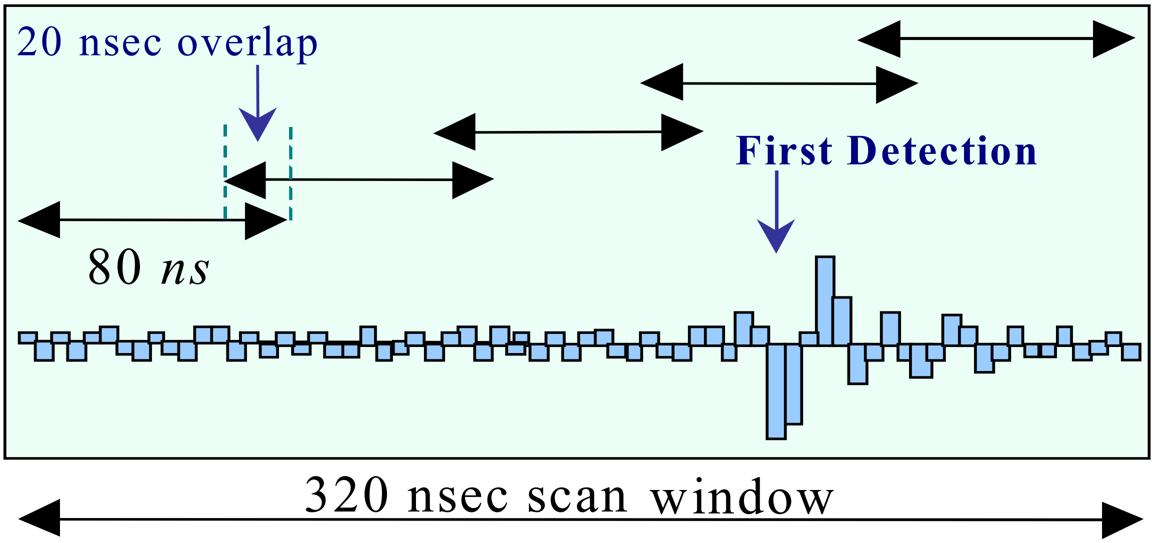


FIGURE 11. The construction of a 310 nsec scan window from 80 nsec wide correlator reception windows is shown. A 20 nsec overlap is allowed between windows. We can see that the output remains low until we come across one of the correlation peaks. The peaks are much above the noise floor and can be easily detected.

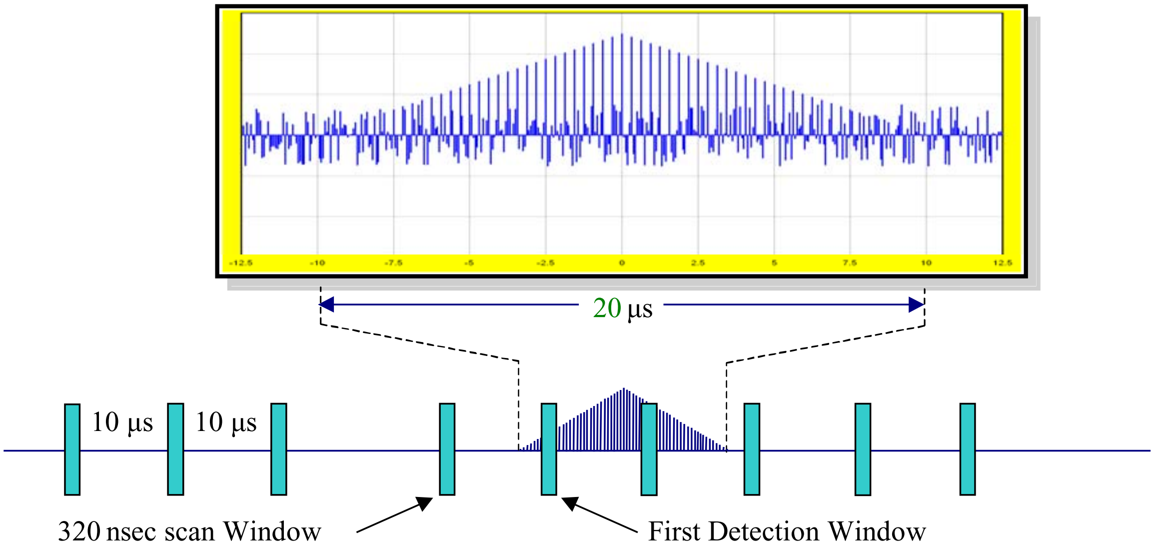


FIGURE 12. The Rapid acquisition process is demonstrated in this figure. The 320 nsec scan windows and the 20 µsec autocorrelation function are shown here. The Listener scans for 320 nsec and looks for a correlation peak. If it does not find a correlation peak in a 320 nsec scan window, it implies that there is no beacon signal in that 10 µsec period. Hence, it moves the reception window ahead by 10 µsec and repeats the process.

The next step is to fine tune synchronization. The two nodes switch to a special (*extra*) Kasami code and adjust their Rx times until a detection is made. Recall the overlap process only estimates the center of the beacon peaks. Because of noise and other factors, the estimate may be off by an integer multiple of 310 nsec. If the estimate of the peak position is not exact, then the special Kasami code will not correlate to the peak value. The Pinger continues to transmit a ONE until the Listener locates the main peak. The Listener moves its reception window by 310 nsec forward and backward progressively until the correlation peak is found.

The two nodes next exchange IDs. It is possible that the two nodes have already acquired each other. If this is the case, then each lets the contact drop. Otherwise, the two enter the ongoing message exchange process. An example of the message exchange is shown in Figure 13.

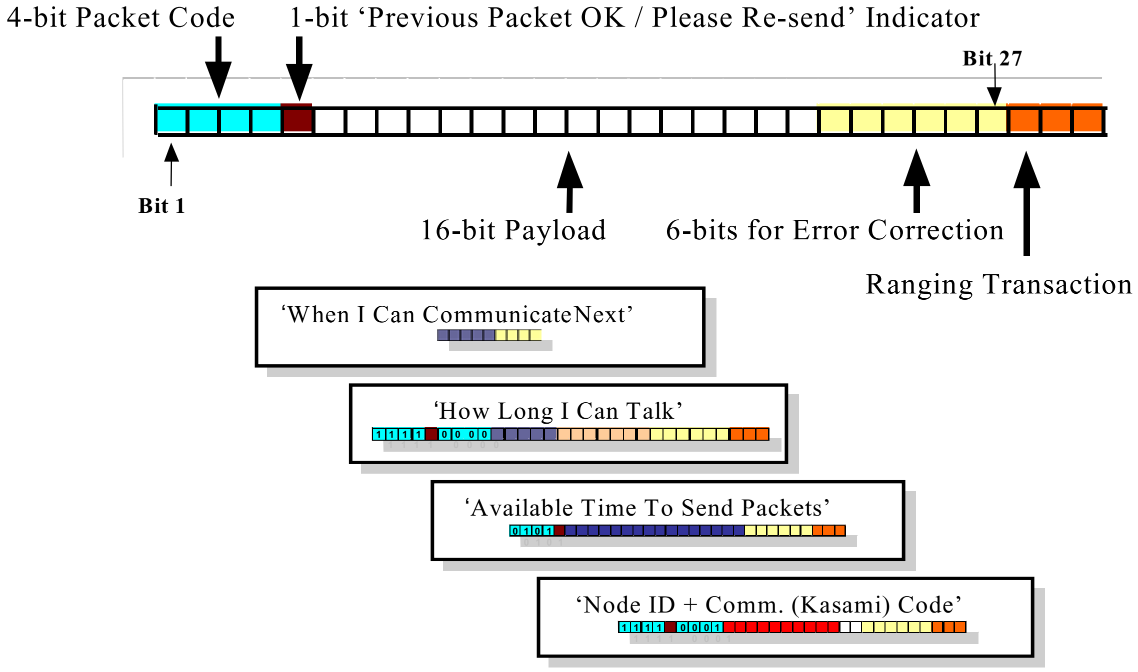


FIGURE 13. Completion of the acquisition process. Listener and Pinger Localizers exchange packets necessary for scheduling future exchanges, including the slot and epoch times that are available and which ID and code sequence to use.

MESSAGE EXCHANGE PROCESS

The nodes exchange a variety of messages with different priorities. The priority of the message depends upon the message content. Some of the messages are processed immediately while others are processed as time permits. The different messages that are exchanged are shown below:

- High priority messages
  - Range Sync
  - Range Correction
  - Node ID
  - Change Kasami Code
- Low priority messages
  - Node to node message
  - Cluster range information – cluster node IDs, range and time of range
  - Time base
  - Node configuration – Pinger or Listener

The messages are processed by the Localizer message handler system. The high priority messages are processed in the order of arrival. The low priority messages are queued and processed as time permits.

Once contact has been established, the nodes exchange a packet once every era. The nodes must communicate every era; the lack of communication is interpreted as lost contact. Hence, if a node does not have any message to send, it sends out a null message to keep up the contact. A range Sync message is sent at regular intervals, (currently set at 20 eras) which provides timing information required for a full ranging operation. Immediately after the range Sync message, the nodes exchange a single bit using the information from the Range Sync message. Since their clocks drift, the peak may not be at the center of each node's correlator window. During the next iteration, each node calculates the offset it required to position the peak exactly at the center and transmits this as the Range Correction message. This gives the nodes a measure of the difference between their clock rates and enables them to cancel out the difference. Now, the two nodes have sufficient information to measure the range between them. The ranging protocol is shown in Figure 14. The rest of the messages exchanged pass information between the nodes. During each of these messages, the range is re-calculated using the last received Range Correction information. Thus, the range information is updated every era.

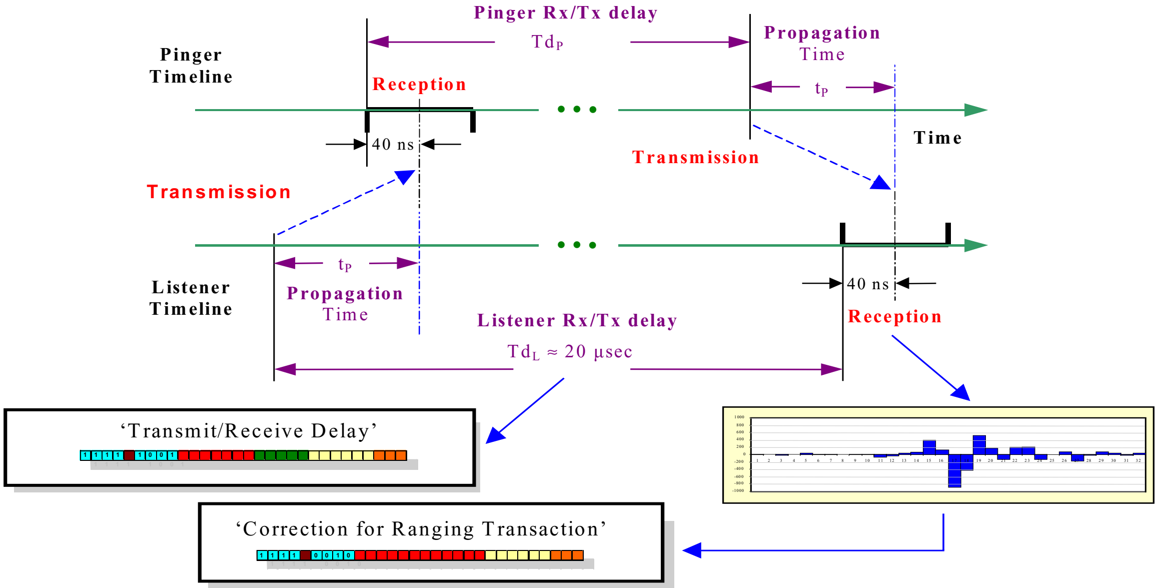


FIGURE 14. Timeline representation of the ranging protocol. Listener transmits and receives while the Pinger first listens and then transmits. This two-way exchange occurs over a small interval of time ~30 μsec. The propagation time t_p is calculated from the Pinger Rx/Tx delay Td_P, the Listener Tx/Rx delay Td_L and the position of the correlation peak within the reception window of both Pinger and the Listener.

LOCALIZER SUPPORT SOFTWARE

The support software includes several functions that are executed as extensions of the calling task. These functions are:

- Node tracking
- Consensus clock
- Background noise updates
- Correlator residuals updates

Node Tracking: Each node has a fairly stable crystal oscillator. These clocks drift only by a few microseconds over tens of seconds. However, because of the timing precision needed to accurately determine range (to the picosecond level), this drift becomes intolerable over a relatively brief period of time. Since clock drifts are independent, some may be moving at a faster rate while others move at a slower rate. In addition, the nodes themselves may be moving. The Kalman Tracker employed by the Localizer software is designed to track both clock drift and node movement. Every single bit exchange results in information which is supplied to the Tracker so that it can track the other nodes' clock drift and movement.

Consensus Clock: It is necessary to keep the clock drift under control to stay in contact with other nodes. However, the uncoordinated adjustment of times within each node can cause the two nodes to drift apart in their time lines. Over a period of time, the epoch timing may not coincide as expected, and the epochs may gradually move away from each other. The purpose of the Consensus Clock logic is to drive all nodes in contact towards a common time line. It achieves this goal by monitoring the Node Tracker and other node information, and by periodically updating the epoch timing and length.

Background Noise Updates: The environment in which the nodes reside is dynamic. Nodes may be moving, passing momentarily through areas that contain more or less background noise. Because the node environment may change, the background noise is constantly updated. This task periodically records the correlator noise during a time when a transmission is not in progress, and exponentially averages the noise over time. The background noise statistics are vital in the detection of peaks, since the amplitude above the noise is the defining factor.

Correlator Residuals Updates: Ideally, the correlator outputs should be ZEROs in the absence of a signal if absolutely no noise is present. Instead, the correlator outputs have a small magnitude of error that cannot be attributed to background noise. This error is referred to as the *correlator residuals*. This task estimates the residuals, which are subsequently subtracted from correlator data before the data are processed by the software.

HARDWARE DEVELOPMENT

The Localizer hardware went through several revisions during this report period. The improvements range from integrated circuit design to antenna design. We designed, fabricated and debugged several prototypes as well as conducted pioneering research in antenna design. Some of the aspects of the design work are explained below.

ANTENNA DESIGN

The antenna development was very important since there had not been sufficient experimental work done on practical radiators for nonsinusoidal signals. We had early experience with the radiators for ultra-wideband signals based on the Large Current Radiator (LCR) invented by Dr. Henning Harmuth. These designs were empirical and were developed as proof of concept rather than to achieve efficient radiation. However, to take the system design further, we realized that the investigations had to be conducted with more theoretical and simulation backup, and validated by measurements for real world antenna design.

We were not able to realize the performance expected of the early LCR designs. There was severe ringing on the received signal by the Localizers using these antennas. It was not immediately clear whether the problem was caused by the transmit driver chip, the transmit antenna, the receive antenna or the receive amplifier. In order to isolate the problem, we investigated each of the possibilities by breaking the loop. We set up an antenna test range at our facility to conduct the experiments. We procured two TEM sensor ultra-wideband reference antennas from Farr Research and a PicoSecond Pulse Lab model 2600C pulse generator. The Farr antenna can be driven by the pulse generator for transmitting a calibrated impulse doublet.

We also used the Farr antenna for receive, and captured the signal with a Tektronix TDS694C 10 GS/s digital real time storage oscilloscope. With this combination we looked at the signal transmitted by our fourth generation Localizer, and verified that there were unexpected oscillations. The antenna design might have been the source of the oscillations, or there were extraneous factors affecting the radiation adversely. To determine the cause, we needed a method to verify the fundamental design of the antenna. To do that, it was clear that the antennas had to be modeled and simulated using sophisticated numerical electromagnetic simulators.

Since the LCR is not a simple electromagnetic structure, the simulator had to be able to model it accurately enough, to avoid any other anomalies. We investigated the features and applicability of several electromagnetic simulators that are available in the market. We evaluated the packages by simulating our antennas and verifying whether they had the required flexibility. The packages that we investigated were LC from Cray Research, IE3D from Zeland Software, Inc. and XFDTD from Remcom, Inc. We decided to use the XFDTD package since it allowed us to construct the antennas exactly as they are and drive them from a variety of sources.

We modeled the LCR with ideal conditions to see that the antenna design was fundamentally correct. The simulation had no ground plane since the LCR is differentially driven. The antenna modeled was a 4 cm LCR. The excitation waveform for a 4 cm LCR and the generated impulses are shown in Figure 15 on the left. On the right, a screen shot of the propagating electric field is shown. This picture shows the propagation of the electric field with time. The intensity of the field is the highest in the red colored region and progressively decreases as we go away from the antenna. The LCR is located at the center of the red region; it is covered by the propagating field plot. This plot shows that there is a strong near field component from the antenna.

For comparison, we simulated an electric dipole antenna of a comparable size. Since the 4 cm LCR fills the volume of a sphere with a diameter of 6 cm, we modeled a dipole 6 cm long. The radiation power versus frequency are shown in Figure 16 for the dipole and the LCR. It can be seen that the dipole has a very narrow useful bandwidth and the pattern has a 40 dB variation between 100 MHz and 2.2 GHz. Besides, the dipole differentiates the impulse shape producing a monocycle. The LCR, on the other hand, has a response that varies less than 6 dB over a frequency range of 100 MHz to 2.5 GHz. and preserves the impulse shape as well. The LCR requires a low impedance drive, of the order of 1 Ω, and must be driven directly from our chip without any transmission line.

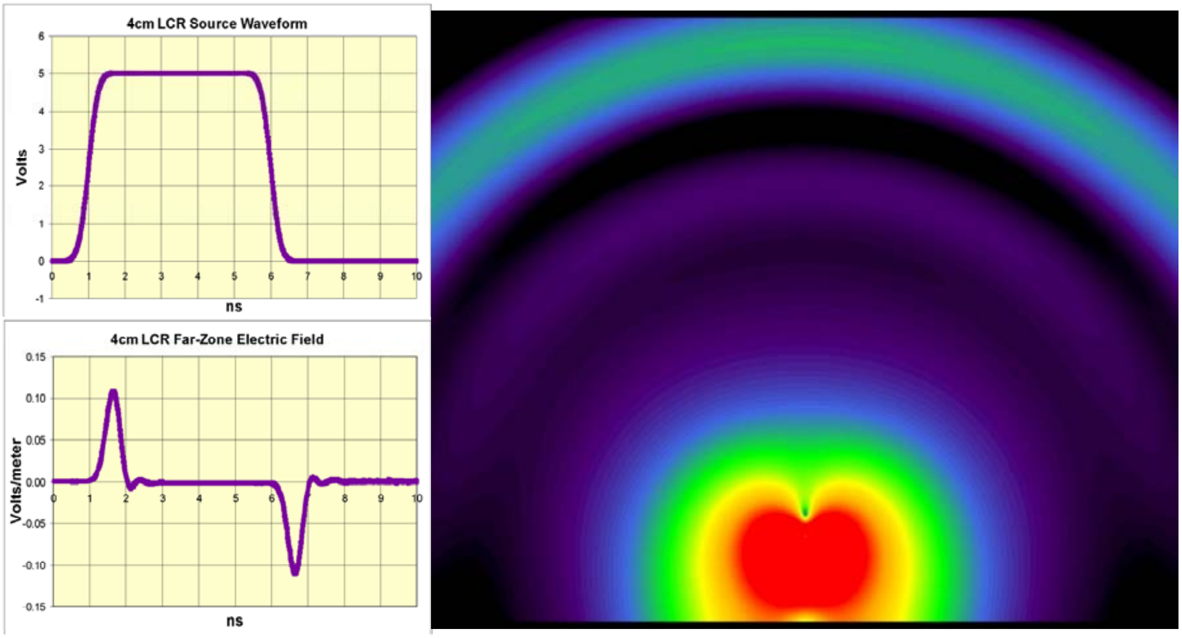


FIGURE 15. The FDTD simulation of the LCR antenna under ideal conditions (no ground plane) is shown here. The LCR source waveform and the resulting radiated electric field are shown on the left. A screen capture of the propagating electric field is shown on the right. The LCR is located in the red colored region. The energy of the field decreases as we move outwards from the red colored region. Thus, it can be seen that there is a significant near field radiation from the LCR.

The simulations proved that the basic antenna design is indeed correct. However, the LCR simulation done here assumes that there is no ground plane present, which is not true for any practical circuit. Besides, the LCR is a small antenna and even a small electrical circuit in the vicinity can present enough ground to it. The printed circuit board ground on the fourth generation Localizer is indeed very close to the antenna and might be the reason for the oscillations. To prove this point, we simulated the LCR with ground plane nearby, and we were able to produce the oscillations that we observed in the measurements.

ÆTHER WIRE CHIP DESIGN

The Æther Wire chips incorporated several significant improvements during the period of this project. Probably, the most important was the design of packaging for the chips. To support high-frequency operation, the chips on our third generation Localizer were mounted chip-on-board (COB). However COB's are difficult to debug and the PCBs are expensive to manufacture. Therefore, we decided to develop a custom package. We had to re-layout the die to simplify wire-bonding for the new package. We took the development one step further to reduce pin count while re-laying out the die, and to prepare the die for a lower-cost commercial plastic package. The Aether4 series chips have the custom packages as well as plastic packages. The Aether4C chip in a plastic package can be seen on the prototype board shown in Figure 17.

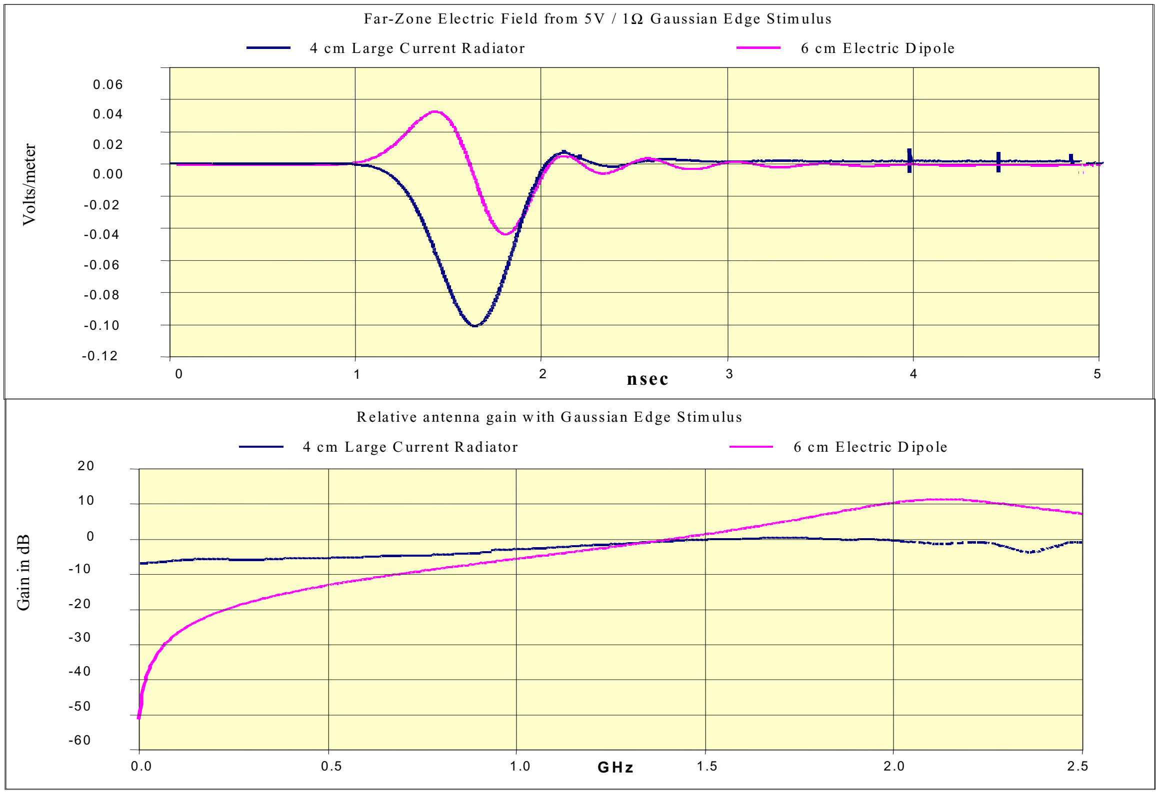


FIGURE 16. The plots of the radiated field versus time and antenna gain versus frequency are shown for the LCR and a comparable electric dipole antenna. The LCR was 4 cm in size and the dipole was 6 cm long. It can be seen that while the LCR preserves the shape of the impulse, the dipole differentiates and radiates a monocycle instead. Besides, the LCR has <6 dB gain variation over a 2 GHz band while the dipole shows about 40 dB variation over the same frequency range.

Besides these changes to the layout and packages, there were other important advances made in the integrated circuit design. One of them was the processor clock (PClock) Phase-Locked Loop circuit. The PClock PLL circuit was modified to operate with a larger selection of clock frequencies, forcing the clock to have a 50% duty cycle.

32 additional D/A converters were added to the Aether4 chips to provide offset correction for each of the 32 correlators. We tried using analog offset correction for the correlators initially, but it was difficult to control them to get precise correction. The DACs offered the advantage of stability even though they take up more space on the chip.

The low noise input RF amplifier was designed and characterized completely for the first time on Aether4 series chips. We added two power rings to the chip, one each for power and ground. With this design, the circuits could be connected to the power supply without long lead lines, thus reducing the inductance and the associated surges and noise on the power supply lines. In addition, bypass capacitors were added to reduce the power supply noise. The edge delay control circuits for the transmit driver were tested using software control and calibrated. Internal voltage DACs for setpoints were added to the design.

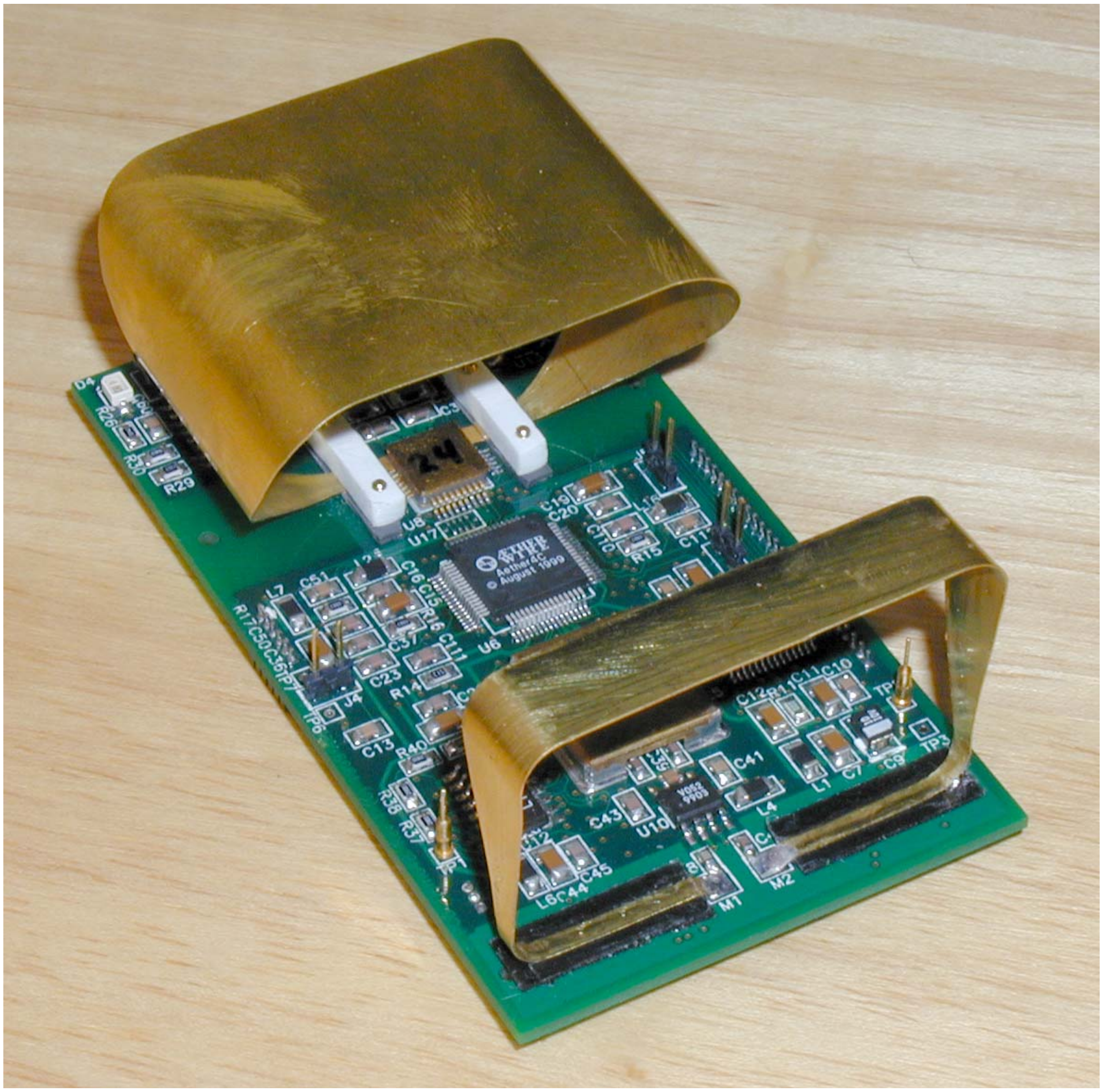


FIGURE 17. The fourth generation pager-sized Localizer prototype is shown here. The Aether4C chip in a plastic package can be seen here along with a driver chip in a custom package. The transmit antenna is connected directly to the driver chip and the receive antenna is in the foreground. The ColdFire processor is on the other side of the PCB.

SYSTEM ON CHIP INVESTIGATIONS

Our goal is to build low cost single chip Localizers. This requires that we integrate not only the Æther Wire chips but also the processor, DRAM and ROM on a single chip. The single chip design also requires certain semiconductor processes, like 0.18µ technology. Since the Æther Wire chips were made with 1.2µ technology, the feasibility of porting the design had to be examined.

FIGURE 18. A group of seven networked Localizers, with one of them interfaced to a PC, is shown. The monitor program output can be seen on the PC screen. A wire-grid model for the environment for real time position display is shown on the left on the screen. A sample of the correlator output can be seen on the right. This network demonstrates the data flooding and cooperative networking concepts

We conducted investigations of the processes from ST Microelectronics and Taiwan Semiconductor. We obtained the design rules for their processes and compared them with our present process, the AMI 1.2 µ available through MOSIS. The important factors examined were:

- Analog circuit capability – including isolated capacitors and high value resistors
- Higher voltage transistors (*i.e.* 5V) for the antenna drivers
- Availability of Dynamic RAM
- EPROM or Flash capability

Since the design rules were very different and the process of scaling the design was essentially nonlinear, and different for different parts of the circuit, it was necessary to get a comfort level for the options available. The goal was to translate just the Æther Wire chips alone to see how the translation works. This was an extremely difficult job since there is little direct correspondence between the current process and new processes. We evaluated various scaling factors that would violate the minimum number of rules in the new process, even if it was not the most efficient in terms of silicon.

LOCALIZER PROTOTYPES AND NETWORKS

Localizer prototypes based on Aether4 series chips were built and tested. The photograph of a fourth generation Localizer is shown in Figure 17. We started with a set of two Localizers, established communication between them, and obtained the ranging results. We extended the network by adding a third Localizer. Once this was achieved, it was relatively easy to extend the network to eight Localizers. A network of seven Localizers is shown in Figure 18. One of the Localizers was connected to a PC to display the results of the position location and communication. The display on the PC shows the wire-grid model for the real time position information and the correlator output for the Localizer interfaced to it.

RESULTS AND DISCUSSIONS

The concept of position location and communication through Localizers based on non-sinusoidal ultra-wideband technology has been successfully demonstrated. The software for the Localizers was developed on a Real Time Operating System (RTOS) platform. Improvements were made to the synchronization and networking protocols to achieve fast and accurate synchronization. The networking protocols were defined to permit easier expansion of the networks and addition or deletion of nodes. Notable improvements were achieved in the radiation patterns of the Large Current Radiator (LCR) antennas. We successfully reduced the size, improved the performance, and interfaced the antennas with the Æther Wire transmit chips. Newer versions of the Æther Wire chips were fabricated and tested. Localizer prototypes based on the new chips were assembled and the distributed networking concepts were proven. In the sections to follow, we discuss the developments and results obtained.

DEVELOPMENT OF SOFTWARE AND PROTOCOLS

The software was ported to the µC/OS RTOS running on a Motorola ColdFire MCF5204 processor. The software was tested using the fourth generation Localizer board. The monitor program for downloading the code was enhanced for greater flexibility and control. With the help of this monitor program, we have been able to debug and calibrate the Aether series chip on the fourth generation Localizer board.

The control software for the Localizer was redesigned with a top-down approach. The peer-to-peer network synchronization algorithms were implemented on the prototype Localizers for self-configuring, ad-hoc networks. Routines for the estimation and cancellation of the background noise, signal detection, and identification were developed. Error control codes were implemented to improve the BER performance. A clock tracking loop based on a Kalman filter was implemented to mitigate the effects of long term clock drift.

A rapid acquisition algorithm with special Beacon codes was implemented to speed up the acquisition process by a factor of approximately 30. We discovered, by a thorough search through code space, a new set of families of Kasami code sequences, verified their properties, and tabulated them.

A PC-based monitor program for the Localizers was developed for communicating with the Localizers and displaying the correlator outputs and the positions of the nodes. The program uses a wire-grid model for the environment for real time display of the Localizer positions.

DEVELOPMENT OF HARDWARE

The hardware developments were in integrated circuit design, antenna design, and the fabrication of prototypes. The Æther Wire chips underwent many major and minor improvements. The changes to the pinout and pad layout facilitated the PCB assembly and reduced the costs. The reduction of the pin count, coupled with the new pad layout, enabled us to design custom packages and even commercial plastic packages for the chips. This is a major achievement since the commercial packages make the fabrication process simpler. Digital-to-analog converters were added to each of the 32 correlators for offset correction. The problem of the PClock Phase-

Locked Loop was addressed and solved. The receive RF amplifier was designed and successfully tested. The power distribution through the chips was redesigned to reduce the surges on the power supply line and hence the noise on the supply line. Numerous bypass capacitors were added for the same purpose. There were many other minor changes and performance tweaks.

We conducted thorough investigations and gained insight into the design of LCRs. We solved the problems of oscillations on the antenna signal by testing against a reference ultra-wideband transmitter and receiver. Finite Difference Time Domain (FDTD) methods were used to accurately model the antenna and identify the problem areas. An antenna test range was set up to verify the simulation results.

Localizer prototypes, with optical interfacing to Windows NT-based workstations, were assembled and tested. The data exchange between Localizers through cooperative networking was verified. A network of eight Localizers was demonstrated, proving the cooperative networking concept.

PRESENTATIONS AND DEMONSTRATIONS

We participated in several DARPA-sponsored meetings and made presentations at various technical forums. The Localizer networks were demonstrated during the Warfighter Visualization PI Meeting in 1999 and to Dr. Norman Whitaker of DARPA during his visit to Æther Wire in 1999.

FUTURE WORK

We plan to develop a next generation system, Generation 5, with enhanced technical capabilities and modularity for experimentation and field testing. Part of this effort will include migrating the driver and receiver chips to 0.5 μ technology. We plan to investigate monopole antennas for the Localizers. We propose to explore new signal processing algorithms to improve the range and reliability of communication between Localizers.