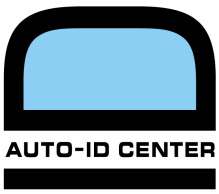
Published June 1, 2002. Distribution restricted to Sponsors until September 1, 2002.



WHITE PAPER

Multi-Band, Low-Cost EPC Tag Reader

Matthew Reynolds, Joseph Richards, Sumukh Pathare, Harry Tsai, Yael Maguire, Rehmi Post, Ravikanth Pappu, Bernd Schoner

AUTO-ID CENTER MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 77 MASSACHUSETTS AVENUE, BLDG 3-449, CAMBRIDGE, MA 02139-4307, USA

ABSTRACT

In collaboration with the Auto-ID Center, ThingMagic LLC has developed a unique multi-band RFID tag reader reference platform. This reader has been designed to read tags conforming to the the Auto-ID Center's emerging EPC specifications at both the 13.56MHz (HF) and 902 – 928MHz (UHF) frequency bands. The hardware architecture consists of a general purpose analog front end up/downconverter for each band, followed by a software radio architecture allowing easy adaptation to new frequencies and protocols. The reader's modular software architecture allows easy expansion while at the same time providing sophisticated networking capabilities including Web configurability, dynamic firmware update, and a TCP/IP reader interface by means of an embedded SQL-compatible database engine. This design offers excellent scalability and flexibility allowing rapid deployment and an in-situ upgrade path.

Published June 1, 2002. Distribution restricted to Sponsors until September 1, 2002.

WHITE PAPER

Multi-Band, Low-Cost EPC Tag Reader

Biography

Matt Reynolds
Partner, ThingMagic

Matt Reynolds is an electrical engineer specializing in wireless communication systems. He has designed remote sensing and communication systems that have been deployed successfully at the south summit of Mt. Everest, on the Embree Glacier in Antarctica, in rural South America, and underwater in MIT's ORCA robotic submarine. Matt's research interests include low power wireless systems, embedded communications and networking, radiolocation systems, electronic circuits and devices, and digital signal processing. He is a Ph.D. candidate and Motorola Fellow at the MIT Media Lab, and holds S.B. and M.Eng. degrees in electrical engineering and computer science from MIT.

Joseph Richards
Project Engineer, ThingMagic

Joey Richards received the Master of Engineering degree in electrical engineering and computer science from MIT. While at MIT, Joey studied communication systems, RF design and modeling, and nonlinear dynamics. His engineering experience includes developing GPS signal processing firmware, designing real-time sensor networks, and programming at all levels from hand-coded assembly for embedded processors to PC applications. He also holds Bachelor's degrees in physics and electrical engineering and computer science from MIT.

Sumukh Pathare
Project Engineer, ThingMagic

Sumukh Pathare holds a M.S. degree in Mechanical Engineering from the University of Massachusetts, Amherst and a B.Tech. degree in Engineering Physics from the Indian Institute of Technology, Bombay. His principal field of expertise is embedded hardware and firmware design. He has developed digital and analog hardware and embedded firmware for various applications including robotics, telephony, optical networking and most recently for RFID.

WHITE PAPER

Multi-Band, Low-Cost EPC Tag Reader

Biography

Yael Maguire
Partner, ThingMagic

Yael Maguire is interested in the fundamental ties between information processing and physics, signal processing and nontraditional computing devices. He has modeled oil pipeline robots and worked on software and electronics for aided inertial guidance systems. Recently, Yael worked on advanced web programming and sensor fusion in collaboration with the UnPrivate House exhibit at the Museum of Modern Art in New York. He has an undergraduate degree in Engineering Physics from Queen's University in Canada and holds a master's in Media Arts and Sciences from MIT.

Harry Tsai
Project Engineer, ThingMagic

Harry Tsai holds Bachelor and Master of Engineering degrees in Electrical Engineering and Computer Science from MIT. He did his graduate work at the MIT Artificial Intelligence Laboratory and previously worked for an AI Lab spinoff specializing in resource allocation software for the airport industry.

Rehmi Post
Partner, ThingMagic

Rehmi Post's research interests are in inertial sensing, dynamics of micro- and mesoscale systems, and MEMS. At the MIT Media Lab, where Rehmi is currently a PhD candidate, he also earned an M.Sc. for the development of e-broidery, a means of fabricating electronic circuitry on wearable textile substrates. Rehmi also holds a B.Sc. in Physics from the University of Massachusetts, where he studied condensed-matter systems and worked with the Tuominen Nanostructures Lab developing superconducting single-electron devices.

WHITE PAPER

Multi-Band, Low-Cost EPC Tag Reader

Biography

Ravi Pappu
Partner, ThingMagic

Ravi Pappu received his Ph.D. from the Physics and Media Group at the MIT Media Lab for his work on designing and implementing inexpensive systems for cryptographic authentication. While at MIT, he co-created the first dynamic holographic video system with haptic interaction. His technical interests are in physical cryptography, optical engineering, and display technology. Ravi holds a B.S. in electronics and communication engineering from Osmania University, India, an M.S. in electrical engineering from Villanova, and a M.S. in Media Arts and Sciences from MIT.

Bernd Schoner
Managing Partner, ThingMagic

Elektrizitätskünstler Bernd Schoner's expertise includes time series prediction, nonlinear estimation, stochastic processes, machine learning, neural networks, and audio processing. His research has led to devices and software applications as unique as the Marching Cello, a wearable instrument providing the functionality of a cello, and a giant polyphonic floorboard for the Flying Karamazov Brothers. Bernd holds a Diplom-Ingenieur from RWTH Aachen, Germany, and an Ingénieur des Arts et Métiers from Ecole Centrale de Paris, France. He received his Ph.D. from the MIT Media Laboratory in 2000.

WHITE PAPER

Multi-Band, Low-Cost EPC Tag Reader

Contents

1. INTRODUCTION

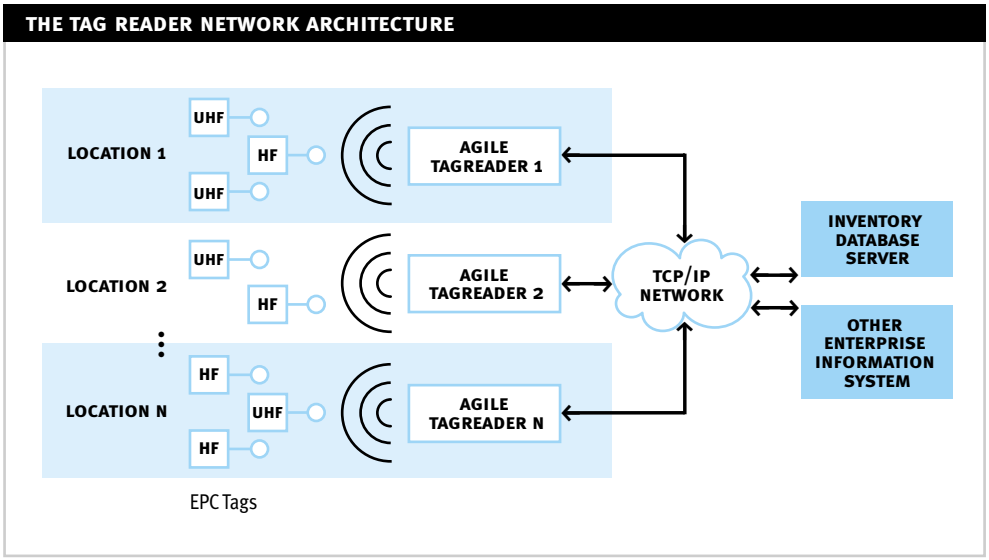
ThingMagic LLC has been working cooperatively with the Auto-ID Center and its members to design and prototype a new breed of RFID reader based on the Auto-ID Center's open-standards philosophy. The architecture of this reader is based on the realization that the RFID reader of the future is not merely a radio transceiver with a serial port; instead, the Auto-ID Center's vision of a supply chain managed with the help of RFID technology requires a fundamental change in both the hardware and the software capabilities of an RFID reader. Under the Auto-ID Center's vision, RFID readers will be installed on every factory floor, warehouse shelf, retail shelf, entry point and exit point to track every product through its entire pre-consumer life cycle. Most existing readers fail the crucial tests of scalability for these applications. We must therefore draw a distinction between the RFID readers of the past and the new generation of RFID readers required to meet these very important supply chain challenges.

In the model of the past, an RFID reader is an isolated object that uses its radio frequency (RF) channel to read a tag and transmit its ID string over a serial port or a rudimentary network interface to a nearby PC, whereupon the PC will interact with a company's enterprise systems. Such last-generation systems are currently deployed in the form of access control systems, simple warehouse logistics systems, toll collection, and other applications. This "dumb reader" solution relies too heavily on a multitude of unreliable, insecure PCs running consumer grade operating systems that require constant human intervention. This solution cannot provide for the realization of the Auto-ID Center's vision from the standpoints of cost, scalability, maintenance cost, installation cost, and power consumption.

[1] The Savant is a distributed database system jointly developed by the Auto-ID Center and OATSystems, Inc.

Instead, future RFID readers will be part of a large, distributed and dynamic system in which each reader is **responsible for the management of its local population of tags**. In this type of truly distributed system, RFID readers act as a gateway between a relatively "dumb" tag and a very "smart" distributed information system which is in turn gatewayed into enterprise software applications using a system like the **Savant** distributed database system [1]. Thus the RFID reader takes on an increasing amount of responsibility beyond that of a simple interrogator; the reader is responsible for all aspects of the management of a local population of tags that is changing dynamically to reflect the motion of tagged objects through the supply chain. The reader described in this paper has been designed to live in the context of a large network, where it provides the functionality of a specialized network gateway with an RF air interface to the tags on one side of the device, and a database server with a TCP/IP network interface on the other side, ready to be part of a distributed data aggregation and analysis system.

Figure 1: The tag reader network architecture



Different applications requiring RFID tagging place vastly different demands on the RF channel of the tag/reader system. Even if the logical data structure and requirements are fixed, as in the case of systems conforming to the EPC specification, the requirements for the air interface tend to vary with the application because of the fundamental physics of antennas and radio propagation at different frequency bands. We believe that different frequency bands, for example 13.56MHz and 915MHz, and coupling technologies (near-field inductive or capacitive coupling and far-field radiation) provide different benefits and functionality trade-offs and we therefore expect that the use of at least two different frequency bands in the RFID marketplace will remain a reality for the foreseeable future. Therefore our reader is designed around the notion of simple analog band modules which can be mixed and matched to support different frequencies within the same reader.

Furthermore, the use of RFID systems in the supply chain requires technology platforms that can be standardized globally, so that tags can travel around the world and be read independently from the specific regulatory requirements in different countries. Consequently the RFID readers in this effort need to be able to read tags without regard to their frequency of operation. Additionally, since the expected lifetime of readers in supply chain management and warehouse management is as long as ten to twenty years, while the effective lifetime of a tag may be only a few days to a few weeks, readers need to be easily software-reconfigurable to support frequency bands and protocols that will become available later. Since new tags are constantly flowing into the supply chain and old tags are constantly flowing out of it, we need to allow for constant system innovation to take place on the condition that an existing reader infrastructure can support new tag technologies.

In short, we believe that "smart" RFID readers deployed in supply-chain applications should:

- operate at multiple frequency bands,
- speak Internet protocols natively,
- be part of a client-server system,
- and incorporate agent-like behavior.

The reference implementation described in this paper is the first step toward a multi-frequency, multi-protocol reader based on open standards. It is designed to communicate with a new generation of RFID tags currently being developed by the Auto-ID Center and its members based on the Electronic Product Code (EPC). The first two species of the new family of tags operate at 915MHz and 13.56MHz. Further development and production of tags at other frequency bands allocated to RFID technology worldwide are in progress. The logical data structure of these tags is made scalable across the family; at this time all of them are designed to store an EPC consisting of a 64 bit or 96 bit number. The reader functions as a translator that makes the specific air interface of the individual tag transparent to the back-end software infrastructure.

Key features of the new design include:

- frequency agility by means of modular analog signal chains,
- protocol agility by means of a DSP-based software radio design,
- standards compliant TCP/IP networking by means of a Linux-based back-end using an Ethernet network,
- low cost because most elements of the system are reused across different frequencies and protocols,
- network-driven protocol upgrades by means of firmware upgrades over an intranet or the Internet,
- interoperability between passive tags, semi-passive backscatter tags, and active tags.

By minimizing the hardware requirements for these different protocols and by implementing software modules that abstract away the differences between protocols, this Reader is superior to most other multi-band solutions in terms of hardware cost and software flexibility. It is the goal of the collaboration between ThingMagic LLC and the Auto-ID Center to make this design accessible on the same open basis as the Auto-ID Center's tag specification efforts, allowing both the end user community and the vendor community to benefit from our efforts.

2. DESIGN OVERVIEW

2.1. Design Philosophy

In designing this Reader we were guided by a small number of key principles:

1. Protocol and frequency abstraction
The physical tag technology, including carrier frequency and communication protocol, should be abstracted away from the network device talking to the reader. The reader communicates the logical properties of the EPC protocol, while hiding the specific physical transport mechanism. This requires that the most common RFID frequencies be supported by the reader so that a single reader can speak to all tags.

2. Scalability
The reader has to be designed in such a way that it scales with the amount of traffic required both on the air interface and on the network interface.

3. Ease of Deployment
The network interface has to enable easy installation, using existing networking infrastructure as much as possible. Therefore the Reader's primary interface is by means of an Ethernet based TCP/IP network.

4. Ease of Maintenance
The reader must be easily remotely maintainable by the information systems staff of an organization so that a separate maintenance staff is not needed.

5. Network Device Metaphor
The reader has to behave like a well understood network device (e.g. a router, network file server, etc) to enable large-scale deployment, configuration, and interoperability.

Guided by these basic principles, we believe that the reader and its population of tags become a natural extension to a company's general purpose Internet or intranet as we know it today (Figure 1). Note that there is no need for an intermediate PC in this architecture; all interaction with tag readers is handled on a peer-to-peer network server basis.

2.2. DSP-based Architecture

A key observation about the design of hybrid analog and digital systems (for example an RFID reader) is that the digital portion of the device can be expected to follow Moore's Law, resulting in rapidly increasing capability and decreasing cost, while the cost and functionality of the analog portion of the device can be expected to be relatively constant over time. Therefore the reader is designed around a powerful Digital Signal Processor (DSP), which handles all the modulation, demodulation and anti-collision search functionality in software. By providing most key elements of the signal chain and

related functionality digitally, the DSP provides a maximum of flexibility since the hardware can be kept constant across different protocols while the firmware is easily adapted, changed and updated.

The analog hardware of the Reader consists of a separate band module for each band, connected to the DSP system. These band modules are simple up/down converters that convert signals from the operating frequency to baseband, so that the DSP system's analog to digital converter can digitize the signal in preparation for digital demodulation.

The Reader is implemented in the form of four separate printed circuit (PC) boards which take advantage of the natural moduarity of the system:

Bamboo-DSP
The Bamboo-DSP board hosts the Bamboo Linux Server, a Digital Signal Processor (DSP), and an Analog-to-digital Converter (ADC).

13.56MHz Band Module
The 13.56MHz band module hosts the analog processing chain of the 13.56MHz signal. The board receives digital control data from the Bamboo-DSP board, connects to the 13.56MHz antenna port and provides both a thresholded digital signal as well as analog outputs to the Bamboo-DSP board for decoding. This module is capable of delivering up to 7W of RF power at a frequency adjustable in software between 13.553 – 13.567MHz.

900MHz Band Module
The 900MHz band module hosts the analog processing chain of the 900MHz signal. The board receives digital control data from the Bamboo-DSP board, connects to the 900MHz antenna port and provides two analog signals to the Bamboo-DSP board for analog-to-digital conversion and decoding. The 900MHz module is tunable in software between 902 – 928MHz and in practice is used in a frequency hopping mode, with power adjustable in software up to +28dBm.

Front Panel Module
The front-panel PC board connects to the Bamboo-DSP board. The front panel board receives user input for configuration and testing by means of four buttons and provides user feedback by means of five LEDs. Additionally the front panel module has a beeper for power on self-test (POST) codes.

2.3. Design Elements

2.3.1. Antenna Unit

The antenna unit of the Reader needs to support multiple frequency bands, with two to four orders of magnitude difference in frequency. Beyond mere differences in resonant frequencies the different bands operate in different physical regimes and hence require different coupling technology.

The current antenna unit uses a planar geometry consisting of a combination of a micropatch element for the UHF band and a coil element for the HF band. The two elements connect to the reader through two independent RF cables. It was initially expected that a single cable connection for the two bands would prove desirable, but the two bands tend to require different antenna placement since the read range at 13.56MHz is considerably less than that at 915MHz. Therefore a separate cable connection for each band is used providing maximal flexibility.

2.3.2. UHF and HF Analog Signal Chains

The analog chain of the proposed reader design is intended to be as flexible as possible. In the current implementation we provide two independent band modules for the HF and the UHF signal chains. On both boards the transmit signal is generated by means of a programmable local oscillator (PLO) module and modulated by a control line coming from the DSP/CPLD unit. The received signal is mixed to baseband using IQ demodulation resulting in two signals. Each channel is digitized in a separate 12-bit Analog-to-Digital converter (ADC) channel and handed off to the DSP for demodulation.

The number of supported frequencies can be extended by adding more hardware modules to the design.

2.3.3. Bamboo Linux Platform

The Bamboo embedded Linux server, which ThingMagic has previously internally designed and developed, is a low-cost general-purpose Linux server that consists of a Motorola 68000-based processor, the MC68EZ328, along with 8MB DRAM, 4MB Flash memory, and integrated network connectivity by means of an SMSC LAN91C96 Ethernet interface chip. Bamboo runs a port of the Linux operating system, which is a free, open-source operating system that provides highly integrated network connectivity and that allows easy application development using free tools. Bamboo has been designed as a hardware/software core that is easily customizable for specific embedded and handheld applications.

Bamboo's network stack is fast and compatible with all the standard Internet protocols, including IP, TCP, UDP, HTTP, and others. The memory architecture of Bamboo allows a fast, parallel interface to the DSP's shared memory for communication and loading of the DSP firmware. These features enable a 'division of labor' where the fast DSP handles low-level computationally intensive protocol and tag processing, while Bamboo, running at a slower pace of 16MHz, collects data when needed, initiates tag reads, and provides tag database information on the network layer.

In addition to real time data handling, Bamboo hosts a web server, which serves the HTML-based query and configuration interface of the tag reader. The query interface enables a user to issue queries using the reader query language documented below. The Web based configuration system lets the user configure the network and RF settings of the Reader.

2.3.4. DSP Module

The DSP Module is responsible for the real-time signal processing tasks in the reader. This Module receives and transmits digital signals to the Band Modules to modulate and demodulate data to and from the tag. With the expectation that the DSP will follow Moore's Law we chose a fairly inexpensive DSP chip, the TI TMS320VC5410, quoted by its manufacturer at a $10 price point in volume. This choice was made primarily on price-performance grounds; other DSPs are certainly suitable but most are more expensive than the '5410. This DSP provides reasonable performance (160MHz clock rate, typically 160MIPS) in the reader application. The computational requirements of modulation and demodulation are not too demanding given the relatively simple AM, FSK, and PSK modulations likely to be used in tag systems.

The DSP itself does not include non-volatile memory. The DSP's firmware is stored in flash memory accessible from the Bamboo Linux processor. Our design makes use of the TI DSP's Host Port Interface (HPI) interface to provide a shared memory interface in to the DSP. At boot time the DSP's firmware is loaded from the Bamboo's filesystem into the DSP through the shared memory. Therefore DSP programming occurs after Bamboo has loaded its own firmware and booted. Because of this architecture the DSP firmware can be easily upgraded in the field and only a single flash memory chip is required for the entire system, keeping the cost low.

3. READER INTERFACES

3.1. UHF EPC Air Interface

[2] The authors thank Alien Technology and Rafsec Oy for their support and cooperation during the development of this specification.

The air interface at 915MHz/868MHz is designed to be compliant with the EPC UHF protocol specification as proposed by the Auto-ID Center in collaboration with its sponsoring institutions [2].

The goal of the UHF EPC interface is to provide an open standard interface that lets different manufacturers build devices that understand each other. While the protocol specifications itself are open, the specific physical implementation and manufacturing technology are left to the individual manufacturer. Hence contributing companies retain a competitive edge by developing proprietary manufacturing processes and device implementations.

The version of the UHF EPC protocol currently running on the Reader is specified in the document "Operational Specification for a Very Low Cost (VLC) Radio Frequency Identification (RFID) System. Part I. Class 1 Devices. Version 9.1" (1). The 915MHz EPC air interface has been designed to comply with Part 15 FCC regulations. A full EMC evaluation and Part 15 certification has not been attempted on the prototype hardware.

3.2. HF EPC Air Interface

[3] The authors would like to thank Philips Semiconductors, Rafsec Oy, and Peter Cole for their support and cooperation during the development of this specification.

The air interface at 13.56MHz is designed to be compliant with the EPC HF protocol specification as proposed by the Auto-ID Center in collaboration with its sponsoring institutions [3].

While the HF tag protocol is designed to largely the same functional requirements as the UHF design, the different physical boundary conditions and regulatory requirements impose a different air interface and logical level protocols. For example, the implementation of the anti-collision algorithm is guided by the available bandwidth for reader-to-tag and tag-to-reader communication. Since these parameters are significantly different for the two bands, a very different anti-collision algorithm was selected for the HF specification.

The version of the HF EPC protocol currently running on the Reader is specified in the document "Revised Draft Specification for an HF EPC Label" (2). The 13.56MHz EPC air interface has been designed to comply with European electromagnetic emission regulations (CE regulations) for the 13.56MHz band. A full EMC evaluation and Part 15 certification has not been attempted on the prototype hardware.

3.3. Reader Query Protocol

To enable a scalable client-server infrastructure between back-end software and the reader, an open and scalable protocol, SQL, was adapted to run across the TCP/IP interface of the tag reader's tag database server. The protocol is used in connection with the Savant hierarchical database software, which has been designed to connect to various different EPC tag readers.

The interface between the reader and the network is defined by a tag database server that speaks a variant of the extensible Structured Query Language commonly used in enterprise database systems. This language is derived from ANSI standards documents X3.135-1989 and X3.168-1989. The Reader

SQL is an extension of SQL specifically designed to access a variety of tags with different frequencies of operation and protocols. The SQL server communicates between the tag database stored in the DSP/ Bamboo shared memory and a host on a network. The host can make a structured query to retrieve arbitrary subsets of the tags in the field based on a number of criteria such as the ID of the tag, the protocol, the antenna and more. The server will request tags and return only those that match the structured query. The server can operate in a poll mode where tags are returned within a specified timeout or in streaming mode, where the tag database is reported at any integer number of millisecond intervals. The tag database can also be queried in a human readable format such as standard telnet (specified in RFC 0854).

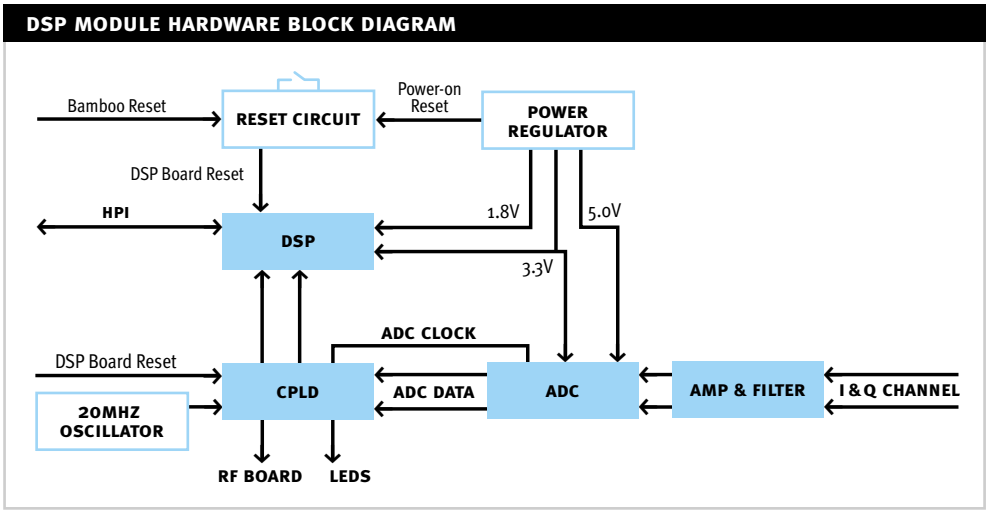
The query protocol is documented in (3).

4. HARDWARE DESIGN

4.1. DSP Board Design

The DSP Module as shown in Fig.2 consists of three main circuit blocks:

Figure 2: DSP module – hardware block diagram



1. **DSP (Digital Signal Processor) block** including a TMS320VC5410 DSP from Texas Instruments and its associated support circuitry such as clock oscillator and power-on reset circuit.

2. **CPLD (Complex Programmable Logic Device)** using the XC95144XL CPLD from Xilinx Inc.

3. **Analog circuit block** using the ADS2807 12-bit ADC from Texas Instruments and an amplifier/filter circuit using OPA2681 high speed operational amplifiers.

The DSP system uses the TMS320VC5410A DSP from Texas Instruments operating at 160 MHz. This chip includes 64 Kwords of on-chip RAM which is used for both program and data storage. Its core voltage is 1.8 Volts while the I/O supply voltage is 3.3 V. The DSP is configured in “Microcontroller Mode” by tying the MP/MC pin to ground, allowing it to boot from on board RAM. The DSP interfaces to Bamboo through its Host Port Interface (HPI). The processor can be reset either manually using a switch, by power-cycling,

or under the software control of the Bamboo. Following a reset, the DSP waits for the Bamboo to download the DSP's operating firmware into its on-chip memory through the HPI. The HPI is also used for transferring run time data between Bamboo and the DSP by means of a shared memory interface. Details of the data transfer are explained in the software section.

The DSP interfaces to a CPLD chip (XC95144XL from Xilinx Inc.) via the DSP's two Multi-channel Buffered Serial Ports (McBSP). The McBSP configuration is tabulated in Table 1. The CPLD interfaces with the ADC using a generic parallel port interface. One of the functions of the CPLD is to latch two streams of 12 bit ADC data into its internal registers and serially shift out this data on McBSP0 and McBSP1 to the DSP.

Table 1: DSP – McBSP Configuration

MCBSP	DIRECTION (RELATIVE TO DSP)	BIT RATE	DATA
0	Tx	24 Mbps	Register Settings for CPLD
0	Rx	24 Mbps	I channel ADC data
1	Tx	1 Mbps	Bit sequence for RF transmission
1	Rx	24 Mbps	Q channel for ADC data

The CPLD has various internal 8bit registers to which the DSP can write using the McBSP. The register addresss and functions are tabulated in Table 2. The DSP uses the McBSP0 Tx line to write into the CPLD registers. The McBSP data is based on a 12 bit mixed address-data format; the first 4 bits designate the address of the CPLD register (LSB first), while the remaining 8 bits designate the data to be latched into the CPLD register (LSB first).

Table 2: CPLD register addresses and function

REGISTER ADDRESS	REGISTER NAME	DESCRIPTION
0	RESERVE_REG	This is reserved
1	VERSION_REG	This stores version of the CPLD code
2	ADC_DIV_REG	Division factor for ADC clock (CPLD clock is divided by this factor and given to ADC)
3	RF_GIO1_REG	Digital I/O lines for RF board (GPIO lines 0..7)
4	RF_GIO2_REG	Digital I/O lines for RF board (GPIO lines 8..15)
5	LED_REG_REG	LED state. (bit0 = LED1, bit1 = LED2, bit2 = LED3)
6	SWRESET_REG	This is reserved
7	DIAG_REG	When LSB in this register is set, ADC output is not given to DSP, instead a repetitive test pattern is generated internal to CPLD and given to DSP.
8	DDS_REG0	This is reserved
9	DDS_REG1	This is reserved
10	DDS_REG2	This is reserved
11	DDS_REG3	This is reserved
12	DDS_REG4	This is reserved
13	MBSP_PASS_REG	When a bit 0..7 is set corresponding GPIO line 8..15 reflects logic state of McBSP Tx1 line.

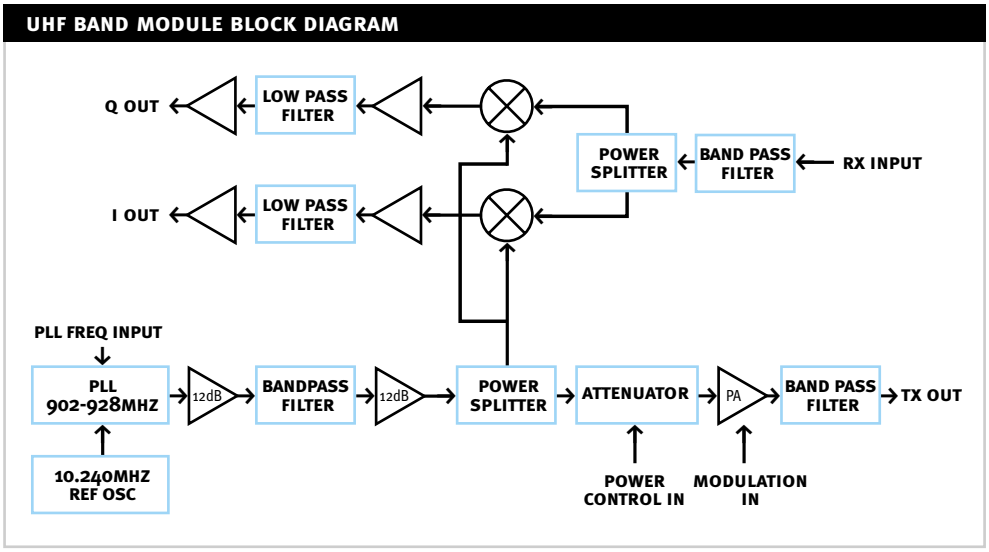
Internal timing of the CPLD is driven by a 20Mhz crystal oscillator. This 20Mhz frequency is divided and supplied to the ADC for sampling the analog data. The division factor can be programmed by the DSP by writing to one of the registers of the CPLD. The CPLD also includes a register which controls the state of four system LEDs on the front panel.

The DSP board contains a small block of analog circuitry for conditioning two incoming analog signals prior to conversion in the ADC. The input signals are filtered and amplified by a factor of two. An offset of 2.5V is added to the signals to match the dynamic range of the ADC. The ADC uses an analog power supply at 5V with the dynamic range of the input signal set to 1.5V to 3.5V. The signals are low-pass filtered with a cut-off frequency of 450kHz, and are sampled at 1.67MHz.

4.2. UHF Band Module

The UHF band module is a channelized 902/928MHz-to-baseband downconverter designed for frequency hopping operation under the FCC's Part 15.247 rules. These rules specify that a maximum output power of 1W may be used in a frequency hopping system using at least 50 channels, with maximum dwell time of 400mS at any given frequency. The band module was therefore subject to to the limitations of PLL lock time and receiver T-R recovery time. A conscious trade-off was made between channel utilization and cost; a single synthesizer design was chosen because of its lower cost even though the synthesizer lock time would result in "dead time" in which the reader field would be off during channel transitions.

Figure 4: UHF Band Module – Block Diagram



4.2.1. Local Oscillator

The operating frequency is generated by a phase locked loop synthesizer module (Z-Comm Inc PSN0930A), integrating a VCO and a National Semiconductor LMX2316 PLL IC. This inexpensive module generates +3dBm output power with phase noise specified at –100dBc/Hz at 10KHz. Significant harmonic energy is present at the VCO output port. In order to increase VCO load isolation, a 6dB attenuator pad is used between the VCO and the first MMIC amplifier (a Mini-Circuits ERA-3SM). This amplifier is biased from the +12V supply with a standard L/R bias network. The amplifier's output power is approximately +8dBm at this point. The amplifier's output is filtered by a two-pole ceramic monoblock bandpass filter centered at 915MHz to remove the second harmonic and other spurious outputs. A second MMIC amplifier and 3dB power splitter split this local oscillator signal into two paths, one for transmit and one for receive.

4.2.2. Transmit Chain

The transmit signal comprises a Hittite Microwave 3-bit digitally controlled step attenuator (for power control) and an RF Micro Devices GSM/AMPS GaAs power amplifier IC, followed by a second ceramic monoblock bandpass filter for harmonic and spurious output suppression. This chain is capable of delivering up to +28dBm at 915MHz. The transmitter can be amplitude modulated by means of the power amplifier's power control input; this is accomplished under digital control from the 900MHz chain's CPLD. While closed loop power control was initially designed into past prototypes, the difficulty of achieving sufficient power control bandwidth at reasonable cost and complexity led to the present open loop design, which has been found to be sufficient for this application.

4.2.3. Receive Chain

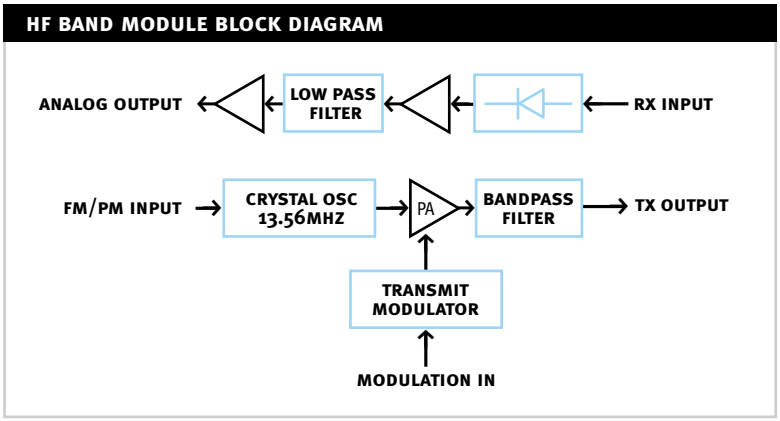
The majority of the receive chain is implemented in software on the DSP. Therefore the analog receive section is very simple. The incoming RF signal is filtered by a ceramic bandpass filter and split into two signal paths for quadrature (IQ) demodulation. This signal is fed to two Mini-Circuits double-balanced mixers; their local oscillator signals are generated by splitting and phase delaying the receive LO path to generate a 90 degree lag in the Q path. No front end RF preamplifier is used because large signal handling capability is more important than receive-noise figure.

The I and Q channel outputs are amplified and filtered by a 4-pole Bessel antialiasing filter with its cutoff frequency at 450KHz. These two signals are then applied to the analog to digital converter on the Bamboo-DSP board. Further signal processing is accomplished digitally; this is the most general approach possible.

4.3. HF Band Module

The HF band module is designed to receive inductively coupled, load modulated signals from an EPC compliant label. It is capable of generating a variable transmit power of up to 7W at a software controlled frequency between 13.553–13.567MHz. The receive section is similar to the UHF band module in that the majority of the signal processing tasks are handled in software on the DSP.

Figure 5: HF Band Module – Block Diagram



4.3.1. Local oscillator

Since the 13.56MHz band is only 14KHz wide, a variable crystal oscillator can be employed. In this design a varactor "pulls" the crystal oscillator; the nominal crystal frequency is 13.560MHz, and at the extrema of the tuning range about 14KHz of tuning range can be achieved under the control of a 12 bit DAC. This capability is useful mainly for output spectrum control during transmit modulation, if desired.

4.3.2. Transmit Amplifier

The transmit power amplifier is a switched mode design operating in a nearly class E mode. This amplifier is designed around an inexpensive logic-level drive MOSFET, the IRL510. Gate drive is supplied by an HCMOS logic IC driving a pair of high speed, high gain bipolar transistors. The gate is driven at 50% duty cycle, while transmit power is modulated by drain voltage derived from a linear amplifier driven by another section of the 12 bit DAC. Thus the transmitter power may be adjusted dynamically and separately for nominal and dip modulation intervals. The output is filtered by the normal series-resonant network and is matched from an internal 12.5 Ohm target load impedance to the 50 Ohm output impedance by means of the same network.

4.3.3. Receive Chain

The 13.56MHz receive chain is based on the standard voltage doubling AM detector circuit, followed by a 13.56MHz trap circuit. An antialiasing filter identical to that used at 915MHz is used before amplification and analog-to-digital conversion on the Bamboo-DSP board. In addition to this output, an otherwise unused opamp section is employed as a comparator to provide a thresholded digital output to the CPLD for testing bit-level demodulation options.

5. SOFTWARE DESIGN

5.1. General Software Architecture

5.1.1. Query Processing Chain

A tag read is exclusively initiated by a client software or user connected to a TCP/IP network. The query client is either embodied in a browser-based Java query interface hosted by the reader (Figures 6 and 7) or it is itself part of a higher-level data handling infrastructure like the Savant. However, it can also be queried simply by a user manually typing requests through a telnet program. The protocol is an SQL-like language carried over a standard internet TCP stream connection (see below).

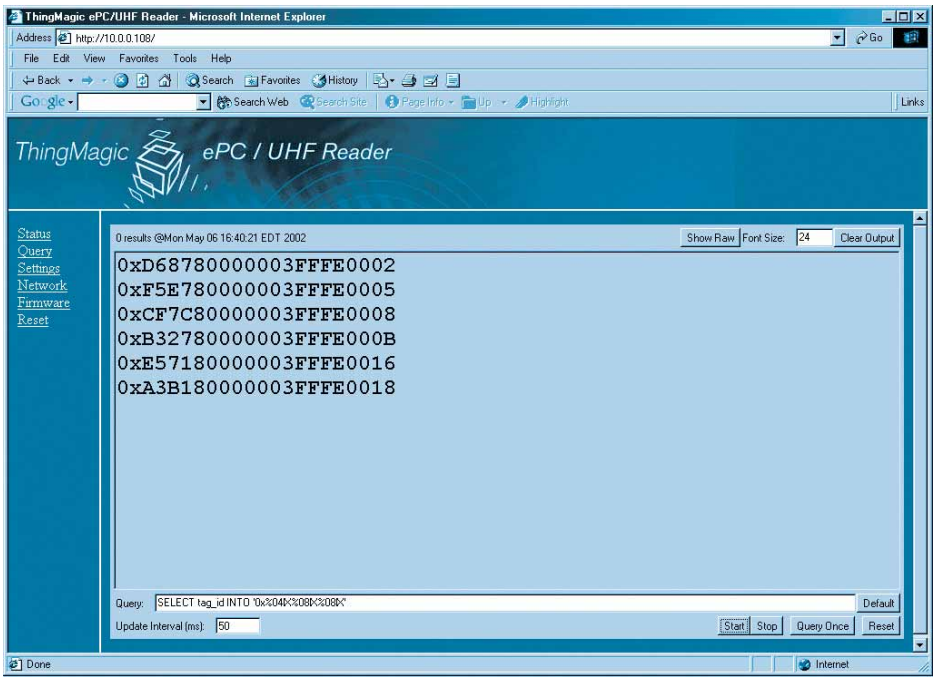


Figure 6: Java Reader interface – Query Page: Six aggregated EPC numbers are displayed in the browser window.

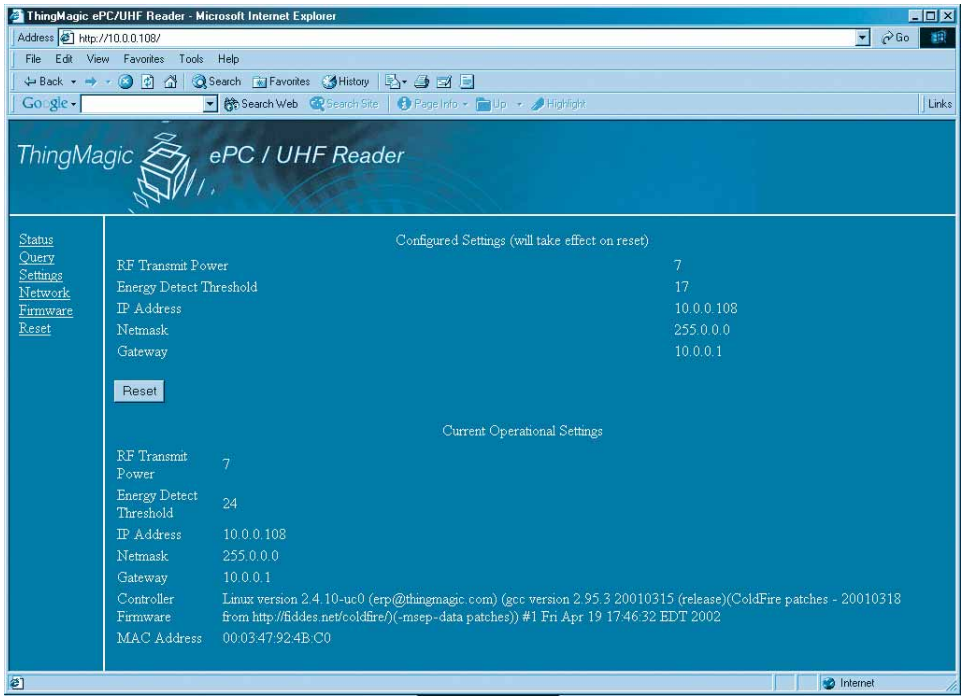


Figure 7: Java Reader interface – Configuration Page: the interface to select the network configuration, reset the device to the factory settings, or select the RF properties.

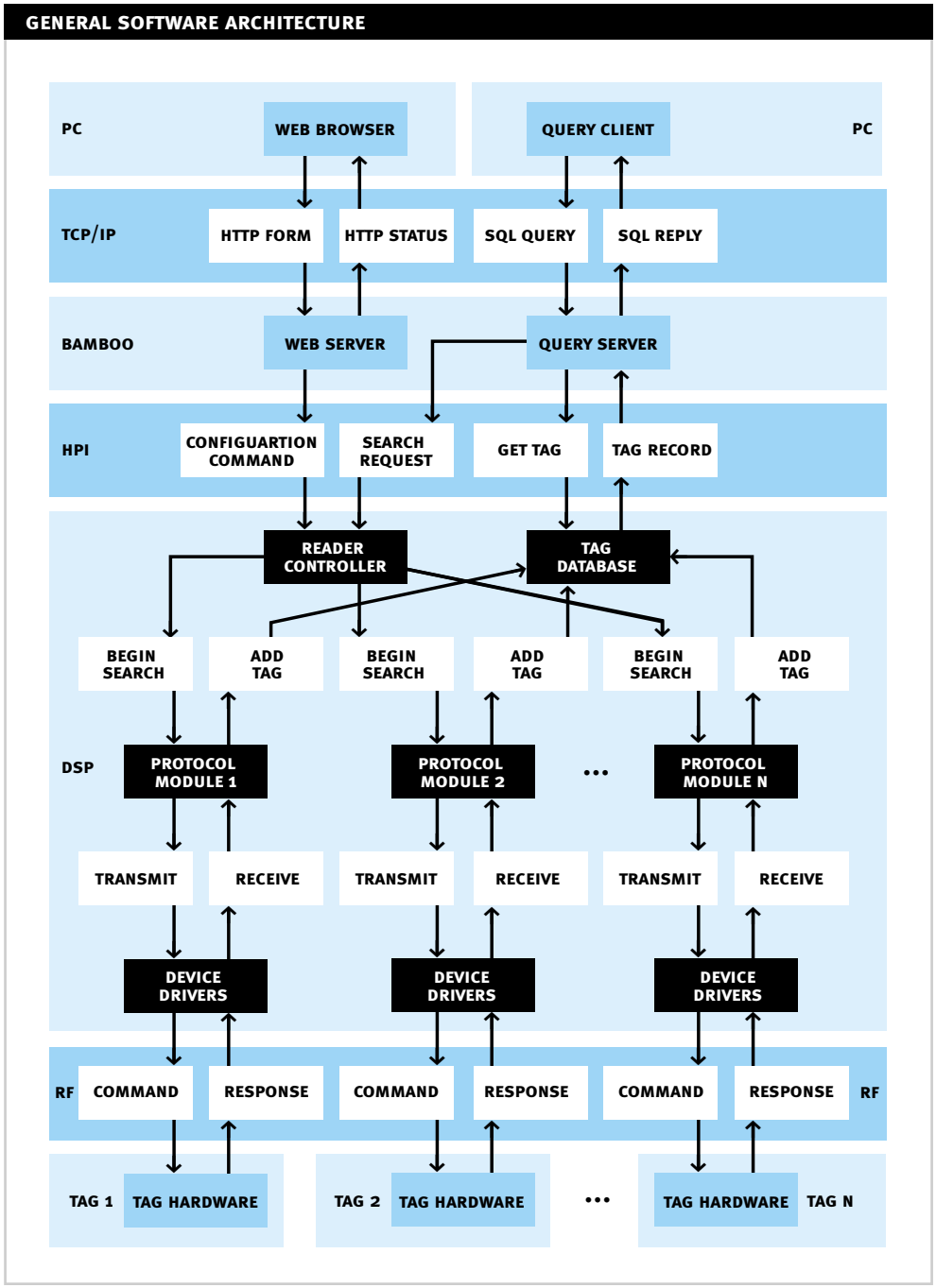
The query server resides on Bamboo, a general-purpose, Linux-based embedded processor. It receives SQL requests from the other end of the TCP connection and interprets them into a series of actions for the DSP. Once the result is received from the DSP, the SQL server forwards the results to the network client. Communication between Bamboo and the DSP occurs through a shared-memory mechanism which is physically connected through the DSP's HPI port.

The query server instructs the DSP to run a structured tag search based on the parameters contained in the SQL query. Within the DSP, the top-level control software requests a search from a protocol module, which encapsulates the particulars of a tag protocol. The protocol module communicates with the device drivers, which manipulate the DSP hardware to send and receive radio signals to and from the tag.

As tag responses are received, the protocol module stores them in a tag database which is shared between all the protocol modules running on the tag reader. After completion of the search, the SQL server reads the contents from the tag database. The tag records are collated and packaged into a SQL reply to send back to the client.

Figure 3 summarizes the software architecture of the Reader.

Figure 3: General software architecture – block diagram



5.1.2. Multi-Protocol Capability

Because of the functional abstraction of the reader's software system, changes to the system to support additional protocols are limited to the protocol modules and their device drivers. Adding support for more protocols involves little change at the higher levels of the system. At the client level, users or software infrastructure are given additional options for new protocols, but the network interface remains unchanged. Similarly, at the Bamboo level, new protocol options are processed but the basic software structure is not changed.

The tag database associates a particular protocol with a particular tag record through a protocol ID field (a "magic number"). The Query Server communicates the protocol ID of a specific tag to the client if so requested.

5.1.3. Reader Configuration

Bamboo hosts a Web server which provides an interface to the Reader configuration settings. Using standard HTML pages and form submissions, the web server reports status and allows configuration of parameters, including power level and network configuration settings such as the IP address.

5.2. Device Drivers

The interface from the DSP firmware to the hardware is abstracted into a set of device drivers. As shown in Figure 8, the device drivers separate the high-level firmware and protocol modules from the low-level hardware interfaces. Device drivers are provided for the transmit (TX) and receive (RX) chains of each RF module, as well as for other hardware functions such as the LED front-panel display.

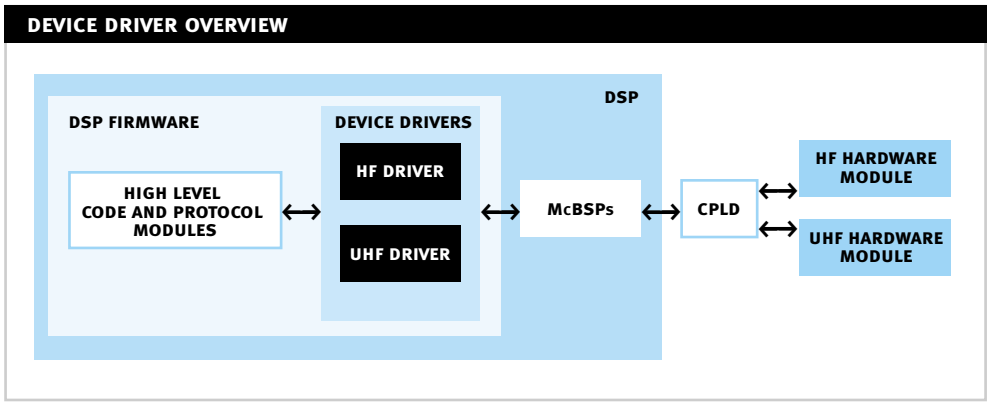


Figure 8: Device Driver Overview: The device driver are a set of software modules that abstract the hardware interface for the high-level DSP code and protocol modules

The device driver code translates the function calls to the device into hardware operations to perform the desired function. The device drivers abstract the hardware interface both by managing on-chip DSP peripherals (e.g., serial ports and DMA controllers) and low-level details of the external hardware devices. Details of the hardware interfaces are discussed in Section 4. The RF device drivers are designed to provide access to the hardware in a protocol-independent manner in order to allow all protocol modules supported by the hardware to operate on a small total number of device drivers. The device driver application program interface (API) consists of a set of C-callable functions for writing data to or reading data from the device or for configuring device parameters. This API mimics the POSIX file I/O interface, using `write`, `read`, and `ioctl` functions for these procedures. A device driver is made active by calling `open` and released by calling `close`.

The `ioctl` call provides an interface to device-specific configuration functions. Table 3 shows the list of configuration functions for the UHF TX and RX devices as an example.

The precise meaning of writing to or reading from a device depends on the nature of the device, and in some cases may be an illegal operation (e.g., writing to an RX device or reading from a TX device). Some devices (e.g., the LED front-panel display device) provide neither a `write` nor a `read` function. These devices are entirely controlled by `ioctl` functions.

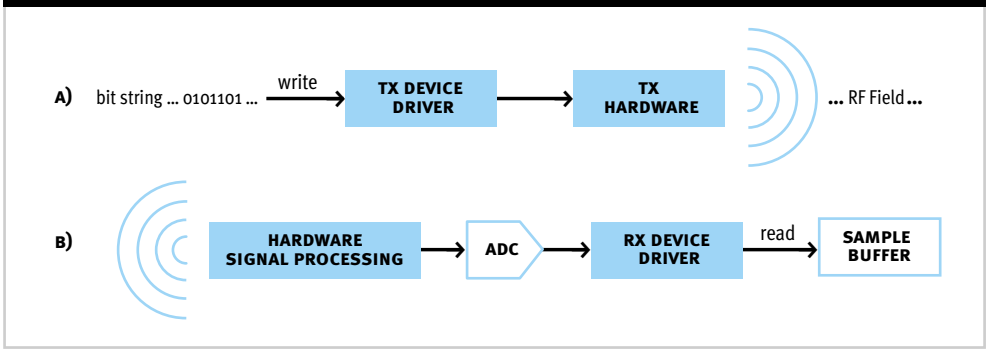
Table 3: Configuration functions for UHF TX and RX

CONFIGURATION FUNCTIONS FOR UHF TX AND RX	
UHF TX DEVICE	
IOCTL FUNCTION	**EFFECT**
set_passthru	When passthru is set to 1, bits written to the device are transmitted over RF as on-off keyed chips at the chip rate set by the set_rate ioctl function. When set to 0, the RF output is held steadily on or off as specified by a call to set_RF_state and writes to the device are ignored.
set_rate	Sets RF chip rate for bits written to the device.
set_RF_state	Sets RF field to be on or off. The state is only used when passthru is set to 0.
freq_hop	Yields for a frequency hop.
set_RF_power	Sets RF output field strength.
UHF RX DEVICE	
IOCTL FUNCTION	**EFFECT**
set_rate	Sets the sampling rate used to collect samples for a device read operation.

Writing to a TX driver typically causes data to be modulated over the output field. Reading from the RX driver fills an input buffer with samples from an ADC. Generally, these samples will represent a partially-demodulated data stream requiring further signal processing. The operations are illustrated in Figure 9.

Figure 9: Typical RF device driver interfaces are through the (a) `write` and (b) `read` operations. These calls use the hardware to transmit and receive data over the RF channel as shown

TYPICAL RF DEVICE DRIVER INTERFACE



Because the various devices share hardware resources (e.g., both the HF and UHF TX devices send data over the same hardware serial port), some form of resource management is required. In this system the high-level firmware manages the resources to avoid conflicts. In general, this is accomplished by only keeping one device driver active at a time. For situations where multiple device drivers must be active simultaneously (e.g., for timing-critical coordination of transmission and reception), safe combinations of device driver function calls are specified.

5.3. UHF Software Module

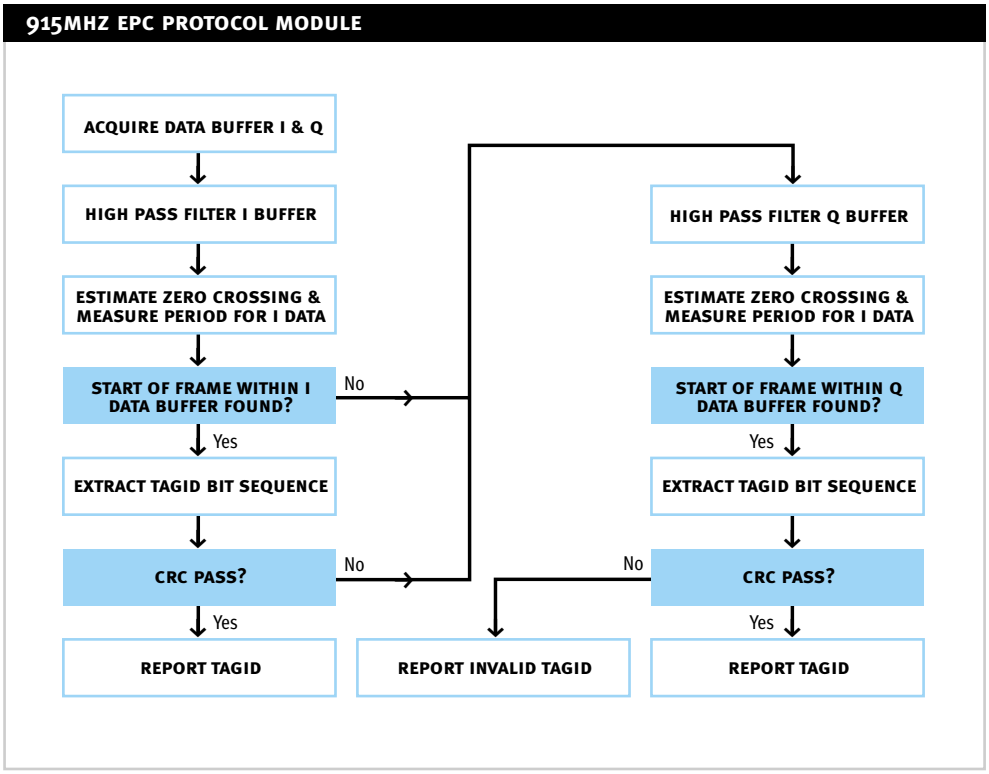
5.3.1. Command Structure

The UHF software module is implemented in its entirety on the DSP. It supports the following calls:

1. `AC_search_start` (anti-collision_search_start) initiates the anti-collision search. Any tags found in the field are reported by registering them in the tag data base. The data base is queried by Bamboo.
2. `AC_search_stop` halts the AC search initiated by the `AC_start_command`.
3. `AC_search_step` steps through the AC search one tag query command at a time.
4. `ping` makes the DSP issue a Ping 0 or Ping 1 command to which tags in the field can be expected to respond.
5. `scroll_start` makes the DSP issue Global scroll continuously. Any tags found are reported to the tag database.
6. `scroll_stop` stops the scroll initiated by `scroll_start` command.
7. `set_ping_threshold` sets the power threshold for detection of tag response to ping command.

The details of various commands and tag responses are defined in detail in (1).

Figure 10: 915MHz EPC module – receive signal, signal processing flowchart



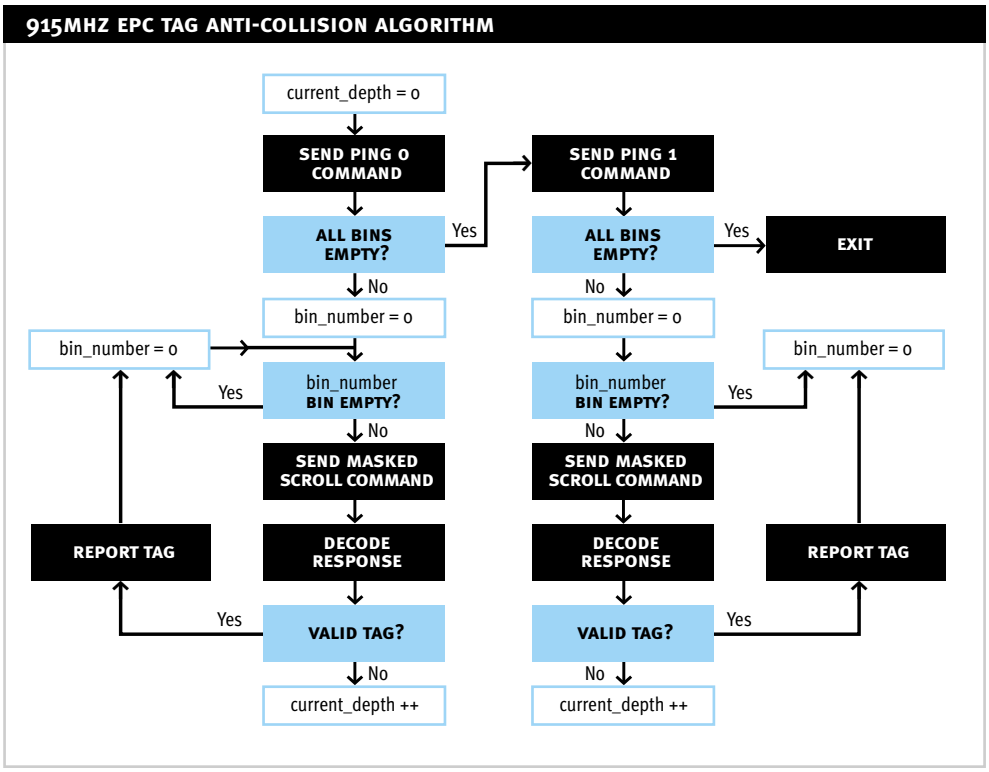
5.3.2. Anti-Collision Algorithm

The UHF EPC tags respond to three commands: `Ping`, `Masked Scroll` and `Global Scroll`. A `Global Scroll` will make all the tags in the field respond at the same time, causing a collision if there is more than one. Hence it is necessary to implement an anti-collision search (AC search) which makes only certain tags respond at a time based on a systematic use of `Ping` and `Masked Scroll` commands.

The AC algorithm for decoding multiple tags in the field is implemented using a combination of `Ping` and `Masked Scroll` commands. The search follows an octal tree in a depth first approach. Figure 11 shows the flow chart for the AC search for the variable `current_depth` set to zero. This variable indicates the length of the bits in the Tag ID being queried and hence the current depth of the search tree. `current_depth` of zero indicates bits 2 to 4 are being queried. A `current_depth` of 1 indicates bits 5 to 7 are being queried and so on.

The AC search starts by sending out a `Ping 0` command. In response to a `Ping 0` command, tags with a least significant bit of zero will respond. The response is received in one of the eight bins. The bin number in which the tag response is received indicates the next three bits of the tag ID. For example, if the `Ping 0` command were sent and a response were received in Bin 3, then the first four bits of the tag ID of the tags which respond: `0 011`. `A Masked Scroll` command is then issued for all the bins in which tag responses were recorded.

Figure 11: 915MHz EPC tag – anti-collision flowchart



The `Masked Scroll` command makes use of the tag ID bits determined so far in the search. The tag responds with its entire tag ID. In case the decoding of the tag ID fails, it is concluded that more than one tag responded and it is necessary to differentiate the tag IDs further by sending out `Ping` commands with more bits. In this case the `current_depth` variable is incremented. If a valid tag ID is decoded, the tag ID is put into the database and the remaining bins that contain a `Ping` response are probed by sending a `Masked Scroll` command.

When the `current_depth` variable is incremented, a `Ping` command is sent with three additional bits in the bit mask. This process is repeated until there are no tag responses at a given `current_depth` or till `current_depth` reaches a maximum value based on the maximum number of bits in a tag ID.

After finishing the `Ping 0` tree, a `Ping 1` command is sent and the `Ping 1` tree is searched in the same fashion as the `Ping 0` tree.

5.3.3. Signal Flow and Demodulation

Each command sent to the tag has a unique command bit code which is detailed in (1). The DSP formats the bit sequence based on the command code and command parameters to be transmitted. The bit sequence includes appropriate CRC values required by the tag to validate the command. The composed bit sequence is transmitted from the DSP to the CPLD via the DMA and McBSP. The CPLD in turn lets the data pass through to the modulation input of the RF board.

In the return channel a down-converted RF signal is digitized by the ADC and given to the DSP through a CPLD. The data is of 12 bit dynamic range. The DSP filters the signal and then processes the data further to extract the data bits.

5.3.4. Scroll Processing

A tag responds to a `Scroll` command by sending out its Tag ID encoded in sidebands of the carrier frequency. Due to variation of the tag clock the bit period of the tag to reader signal can vary greatly. The bit period may also drift while the Tag is transmitting its response. The actual bit period is determined by the DSP by searching for a known preamble bit pattern transmitted by the Tag. Once this bit period is determined, the Tag ID is decoded by fitting the received signal to the pattern expected for bit one or bit zero. During this process the bit-period estimate is constantly being updated to account for the drift in the bit period. The signal processing steps are detailed in Figure 10.

Once the required number of bits is received the Tag ID is validated by checking the CRC. Once the CRC check is passed the Tag ID is reported to the database.

5.3.5. Ping Processing

A tag responds to a ping command by sending 8 bits of data in the appropriate `Ping` Bin (Bins 1 to 8). The DSP software module evaluates the energy in each Bin by comparing it to a reference Bin. If the power of the signal is more than an adjustable threshold times the power in the response-free reference signal it is concluded that at least one tag is present in the Bin.

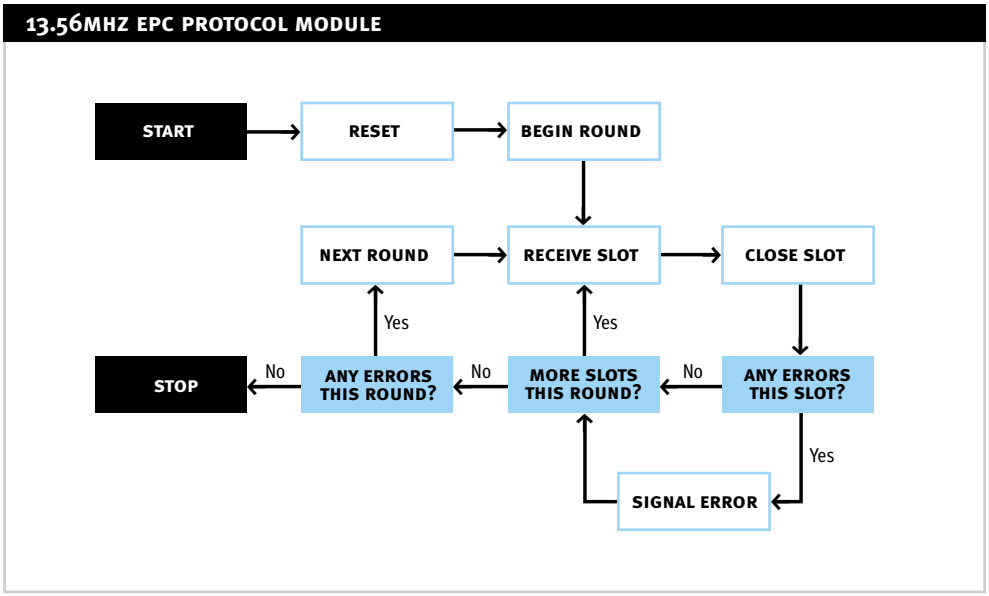
5.4.HF Software Module

Unlike the EPC UHF protocol, the EPC HF protocol and its anti-collision scheme is based on the idea of pseudo-random slotting, i.e. tags respond at different times thus avoiding collision. The basic transaction in the EPC HF protocol is a “slot” which, simply put, is a time slot in which a single tag is expected to reply. Slots are gathered into groups called “rounds,” each of which has a fixed number of slots, declared at the beginning of the round. A tag search consists of a series of rounds, one right after another, until all available tags have been heard from.

Before a search cycle begins, a `Reset` is issued to put all tags into a known starting state. The search is then started with a `Begin Round` command.

The inner loop of the search process processes a single slot. First, the Reader listens for the duration of the slot and attempts to decode a tag response from the received signal. After this processing is done, it closes the slot by sending a `Close Slot` signal.

Figure 12: 13.56MHz EPC tag – reader protocol module flowchart



In addition, an error signal is sent if the Reader believes that the previous slot contained a tag response but was unable to decode it properly. This error signal informs the tag that its response was not successfully received and that it should repeat its response later (during the next round.)

The inner loop repeats for every slot in the current round (a number which was predetermined at the beginning of the round, and should ideally be chosen to minimize the chance of collisions between tags while also minimizing the number of unused slots.)

When the round is completed, the Reader decides whether or not another round is necessary based on whether or not there were any collisions (decode errors) during the round. A collision implies that there is still at least one tag that is responding, but has not been successfully heard. In this case, a new round is initiated with the next round command, and the cycle begins again.

6. CONCLUSIONS

This paper puts into context and motivates the concept of a multi-frequency EPC tag reader. It further summarizes the basic design principles and choices for the reference implementation of such a reader, and finally explains that implementation in detail. In addition to this white paper, a set of hardware schematics and firmware source code will be made available under license by the Auto-ID Center. These documents in combination with this paper constitute the EPC Reader Reference Design, which will enable a skilled engineer to reproduce the design and manufacture a working reader, or to use this Reference Design as a starting point for another EPC reader design.

A systematic performance evaluation of the Reader will be undertaken over the coming months. We will examine read range and read pattern with varying antenna design and frequencies, read-rate variation in different geometries, throughput of the anti-collision search, and reliability of reads under real-world conditions.

The dual-band reader offers an outstanding opportunity to directly and simultaneously compare the performance of HF versus UHF RFID technology. With the back-end software interface and the digital

reader infrastructure remaining constant, a comparison can be made between the true performance differences due to the analog front-end circuitry and air-interfaces. It is the author's belief that such testing will reaffirm the need for multiple frequency bands in supply chain RFID deployment and hence the need for multi-band RFID readers.

7. ACKNOWLEDGEMENTS

The authors would like to thank Kevin Ashton, Sanjay Sarma, Peter Cole, Dan Engels, Silvio Albano, the Auto-ID Center, and the Auto-ID Center's Sponsor Community for making this work possible.

8. REFERENCES

1. **Auto-ID Center. Operational Specification for a Very Low Cost (VLC) Radio Frequency Identification (RFID) System. Part I. Class 1 Devices.**
   Version 9.1., 2002.

2. **Auto-ID Center. Revised Draft Specification for an HF EPC Label,**
   March 8, 2002.

3. **ThingMagic LLC. Reader Query Protocol.**
   Rev. 1.2, April 2002. To be published by the Auto-ID Center.

Designed and produced by Foxner. www.foxner.com