Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 934896, 8 pages
doi:10.1155/2010/934896

Research Article

A Software-Defined Radio System for Intravehicular Wireless Sensor Networks

Xiangming Kong, Deying Zhang, and Mohin Ahmed

Information & System Sciences Lab, HRL Laboratories, LLC, Malibu, CA 90265-4797, USA

Correspondence should be addressed to Xiangming Kong, ckong@hrl.com

Received 8 April 2010; Accepted 10 August 2010

Academic Editor: Ronan Farrell

An intra-vehicular wireless sensor network is designed and implemented on a software-defined radio system. IUWB signal is chosen to carry the data packets. The MAC layer of the system follows the specification of the IEEE802.15.4 standard. The transceiver design, especially the receiver design, is detailed in the paper. The system design is validated through lab test setup.

1. Introduction

Modern vehicles are equipped with a multitude of sensors to collect data that are essential for proper operation of the vehicle. These sensor data include vital information such as exhaust quality and lateral velocity which affect driving safety and environment friendliness, as well as information for improving the comfort and convenience of passengers. As a trend, more and more sensors will be added in the future to make automobiles "smarter". Currently, these sensors are connected to the Electronic Control Unit (ECU) via cables. By 2002, the length of these cables has already added up to over 1000 meters and weighed more than 50 kg, and the cable length still increases rapidly [1]. These cables not only adversely affect the cost and fuel economy, but also increase the complexity of the vehicle design. Another major drawback brought by the cables is the scalability issue. Whenever a new sensor is added to a car, a cable has to be added and routed properly inside the car body. The cable may interfere with other components. In this case, much more redesign work is necessary than simply adding the sensor alone. One way to reduce the undesirable effect caused by the bundle of wires is to connect sensors with the ECU wirelessly. Hence, ElBatt et al. proposed a wireless sensor network for intra-vehicular sensor data transmission [2].

Since some sensor data are essential for the safety of the vehicle, it is important for the intra-vehicular wireless sensor network (IV-WSN) to provide the same level of reliability and transmission latency as offered in the current wired system. This requirement differentiates the IV-WSN from the existing sensor networks and imposes a big challenge in the design of the WSN, especially at the physical layer, which determines the data integrity and data rate. Another factor to be emphasized in the WSN design is the power consumption. Removing wires means removing not only the data cables but also power supply cables. The sensors have to rely on other power sources. Vibration energy can be one viable source [3]. Energy harvesting for wireless sensors is by itself a hot research topic and will not be discussed further here. We only note that since the existing energy harvesting techniques cannot provide a large amount of power to the sensors, there is a strict limit on the power budget for transceivers.

In addition to the two major requirements mentioned above, many other issues are needed to be addressed in the design, including, but not limited to, cost, sensor heterogeneity, wireless channel variation, modulation waveform, and networking. Without prior experience, a flexible platform such as a software defined radio (SDR) system is proper in the early design phase to carry out different experiments.

The importance of removing the cables in vehicles has begun to attract attention only recently. Although some research has been done to study the viability of an IV-WSN [4, 5], to our knowledge, our system is the first SDR for IV-WSN. It will be very useful for future development of IV-WSN. Moreover, our SDR can handle very broadband signals and will be found useful in many applications.

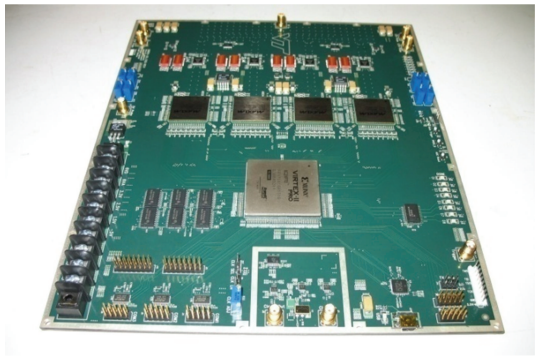

Figure 1: The transceiver board.

2. System Overview

The IV-WSN has to coexist with narrowband communication devices such as cell phones and should not add much interference to the narrowband system nor should it be susceptible to the interference from the narrowband systems. Moreover, some sensors have to send data at a rate that can only be serviced by relatively large bandwidth. Several short range wireless technologies are compared as the first design step, including RFID, Bluetooth, Zigbee, and UWB. The data rate and transmission range of RFID technology are both small and not suitable for our application. A Bluetooth network supports up to seven slave devices. This network size is too small to handle many sensors. Furthermore, Bluetooth and Zigbee both use the crowded ISM band of 2.4 GHz and are vulnerable to interference from signals coexisting in the same band, reducing the system reliability. To reduce interference, a low-power wideband technique is much more preferable. One choice is the ultrawideband (UWB) technology.

Two important types of UWB techniques exist: impulse UWB (IUWB) and OFDM UWB. IUWB signals are easily generated with simple hardware and widely used in WSN. Another valuable feature of the IUWB technology is its high time resolution. Like many indoor environments, the intravehicular communication channel is rich in multipaths [6]. An IUWB signal can easily differentiate the line-of-sight path and the reflection paths. This feature greatly simplifies the receiver design and reduces hardware cost. Since one receiver is needed for each sensor or at least each group of sensors that are positioned together, a low-cost receiver is highly desirable for the IV-WSN. Based on the above consideration, IUWB is our first choice as the media to carry information.

A software defined system is called so because many fixed hardware components in it are replaced either by software or reconfigurable firmware running on computing devices. Unlike the analog devices that handle analog signals directly, computing devices can only process digital signals. Hence, the analog signal received by an antenna has to be digitized before it can be further processed. Due to the limitation imposed by current analog-to-digital converters (ADCs), most existing SDRs only replace the baseband or at most intermediate-frequency (IF) hardware components and leave the radio frequency (RF) front end components unchanged. This configuration limits the system in two ways. First, we have to restrict ourselves to baseband experiments. Any change made in the RF components has to be done in the traditional way, that is, designing a brand-new RF board. Secondly, and more importantly, we cannot try out the IUWB signal most suitable for our system through those SDRs. Therefore, an SDR that digitizes the signal in the RF band is necessary for our system. To address this requirement, we adopt the testbed from Virginia Polytechnic Institute and State University in our system [7, 8].

For a simple WSN, a network stack made of a physical (PHY) layer and a media access control (MAC) layer is enough. There has been a lot of effort devoted to the MAC layer protocol design [9, 10]. IEEE802.15.4 standard also specifies the control and implementation of the low-rate Wireless Personal Area Network, which can also be applied to a WSN. To simplify the system, we follow the IEEE802.15.4 specification for MAC layer design. However, to accommodate the high data rate required by some sensors in our application, we replace the OQPSK modulation specified in the standard by IUWB in the PHY layer. Hence, the system is not compatible with IEEE802.15.4 devices.

3. System Architecture

3.1. Transceiver System. The major components of the transceiver include a simple RF front end, eight ADCs that can take up to 10^9 samples per second each, a Field-Programmable Gate Array (FPGA) embedded with a CPU, and some peripherals. The transceiver board is shown in Figure 1. Its corresponding functional block diagram is given in Figure 2.

The CPU is responsible for data handling, MAC layer processing, and hardware control. To keep the system flexible, the RF front end only consists of the most basic elements: a transmitter/receiver switch, filters, and amplifiers. At the transmitter side, a transmission controller reads in the data packetized by the MAC layer and generates a positive/negative pulse for each 1/0 bit accordingly through a diode. The pulses are then passed to the RF front end and sent out through an antenna. At the receiver side, the signal from the RF front end is first digitized by eight MAX104 ADCs. Each ADC offers an input bandwidth of 2.2 GHz with a maximum sampling rate of 1 GHz. Interleaving sampling techniques as in [7, 8] are applied to combine the samples from different ADCs. The total maximum sampling rate is thus 8 GHz. The samples from the ADC are fed into the FPGA. The transmission controller and the remaining receiver blocks are all designed using VHDL in RTL level and implemented in Xilinx FPGA (xc2vp70ff1704-7).

3.2. Signal Structure. Before jumping to the design detail, it is useful to first explain the signal structure because some design issue is closely related to the signal structure.

The data from sensors are first wrapped into a packet with the following structure see Table 1.

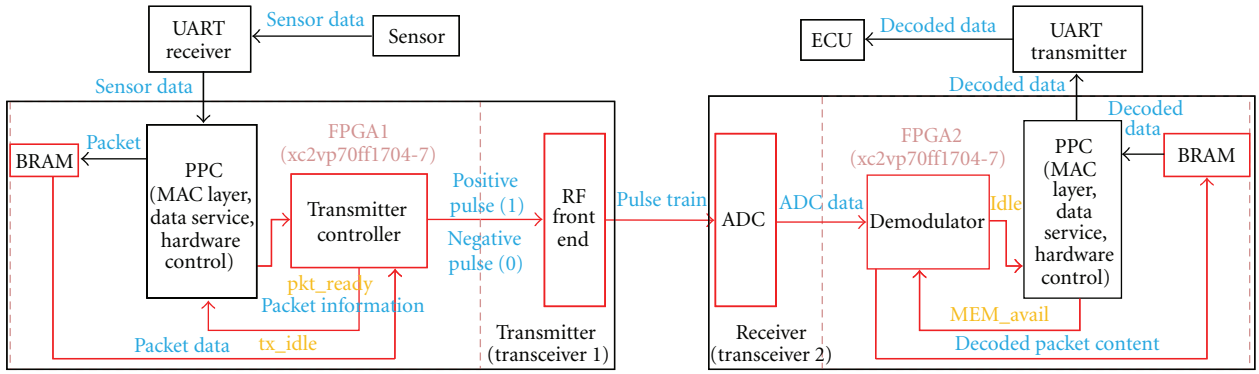

Figure 2: Transceiver block diagram.

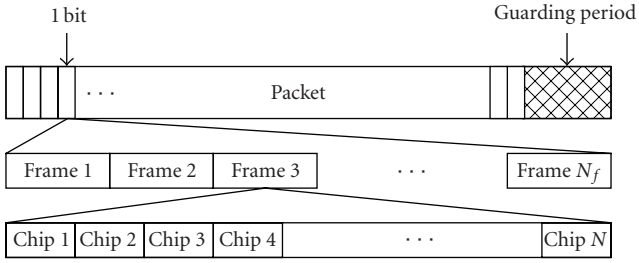

Figure 3: Signal structure.

Table 1

Bits: 32	8	8	Variable
Preamble	SFD	Frame Len	Payload

The preamble bits are all "1"s and are known at the receiver. The preamble is used as a pilot to learn the wireless channel. The Start of Frame Delimiter (SFD) is a byte of fixed value. The frame length byte records the length of the payload in bytes. The payload comes from the MAC layer which contains the sensor data and the MAC header.

At the signal level, each bit is made of N_b frames. Each frame is further divided into N_f time slots called chips. The duration of a chip equals the length of a transmitted pulse. Within each frame, only one chip contains a pulse. Other chips are empty. The index of the chip containing the pulse varies from one frame to the next and follows the pattern of a time-hopping (TH) sequence of length N_{th}. N_f can be equal to or greater than N_{th}. The large N_f is useful for removing any intersymbol interference (ISI). The signal structure is given in Figure 3. According to this structure, the transmitted signal can be written as

$$s(t) = \sum_i b_{\lfloor i/N_b \rfloor} p\left(t - iT_f - c_{[i/N_f]} T_c\right), \tag{1}$$

where b_n is the bit sequence, $p(t)$ is the transmitted waveform, T_f is the frame duration, T_c is the chip duration, and c_n is the time-hopping sequence.

3.3. Receiver Design. Each multipath attenuates and delays the transmitted signal differently. Assuming there are L paths, the received signal can be represented as

$$r(t) = \sum_{l=1}^{L} a_l s(t - \tau_l), \tag{2}$$

where a_l and τ_l represents the path amplitude and delay, respectively.

We use a coherent receiver in our system. A coherent receiver demodulates the signal by correlating the received signal with a local template and determining the bit value based on the correlation result. Letting $g(t)$ be the local template, the correlation result is

$$R(\tau) = \int_0^{T_f} r(t) g(t, \tau) dt. \tag{3}$$

As explained in the next two sections, different templates are used in the timing acquisition phase and the demodulation phase.

The digitized signal from the ADCs passes through five-phase processing: energy detection, timing acquisition, bit demodulation, packet synchronization, and packet demodulation. When the receiver powers up, after initialization, it enters an idle state. Once the CPU informs it that the upper layer is ready, it begins to detect energy on the channel by comparing the square of the received signal with a threshold value. This procedure is realized through a correlator where the received signal itself is used as the local template. If energy is detected on the channel, the receiver enters timing acquisition phase. The timing acquisition process determines the channel delay. If correct delay is acquired, future absolute correlation value should be larger than a given threshold. Only if this condition is met can the receiver move on to the next phase. Otherwise, it returns to the energy detection phase. In the next phase, the receiver first generates the local template for demodulation purpose. Then, it demodulates each bit and begins to search for the SFD pattern from the demodulated bits. When the SFD field is found, it means that the packet level synchronization is reached. The next byte after the SFD should be the payload length. This value dictates the number of bytes the receiver should demodulate

before it notifies the upper layer of the incoming packet and returns to idle state, waiting for a new signal from the upper layer. The whole process is illustrated in Figure 4.

When the demodulator receives a command from the CPU to begin receiving packets and enters the energy detection state, there may be a node already sending on the channel. Even if the receiver is not receiving the preambles, it may still pass the timing acquisition phase and begin to demodulate bits. However, since there is no byte in the payload that has the same pattern as SFD (this is ensured by the software), the demodulator never reaches packet synchronization and may get stuck. To avoid this scenario, a guarding period is added at the end of a packet. Since a packet sent from one node will be received by all nodes, every node knows when the guarding period is. During this period, no sender will transmit a packet. While the demodulator searches for the SFD, it also monitors the absolute correlation value. If this value drops below a threshold, it means that the end of a packet has been reached. The demodulator automatically stops SFD search and jumps back to the energy detection phase. (In fact, even if a node fails to detect the previous packet and transmits during the guarding period, the channel it goes through will be different from the previous sender. Since the demodulation template is one frame of the signal from the previous sender, with high probability, this template would not have high correlation with the current received signal and the correlation value test would fail. The receiver then goes back to the energy detection phase).

3.4. Timing Acquisition. The narrowness of a pulse and the low duty cycle of the pulse period of an IUWB signal impose a stringent requirement on the time accuracy at the receiver side. The fine timing resolution in IUWB systems results in a large search space, making timing acquisition a very challenging problem. Long spreading sequences employed in typical IUWB systems further complicate the acquisition problem. Acquisition can be achieved by a noncoherent energy detection scheme [11]. Since this scheme is more sensitive to noise, we focused only on coherent receivers. In general, the search space in the acquisition process is made up of two subspaces: the one for the delay of the pulse within a chip and the one for the phase of the spreading sequence. The existing paper addresses only one aspect of the problem. Either the spreading code is assumed to be absent [12], or the channel delay is assumed to be an integer number of pulse width [13]. There are two typical searching schemes: serial search and parallel search. Serial search scheme requires very simple hardware. However, its searching time is long. Multiple searches can be done in parallel to reduce the search time, but the hardware complexity increases a lot. Many techniques have been proposed to speed up the searching process without increasing hardware cost. However, they either require large computational complexity [14] or are not suitable for the environment rich in multipaths [15, 16]. In our system, a new acquisition scheme is developed to meet our needs [16].

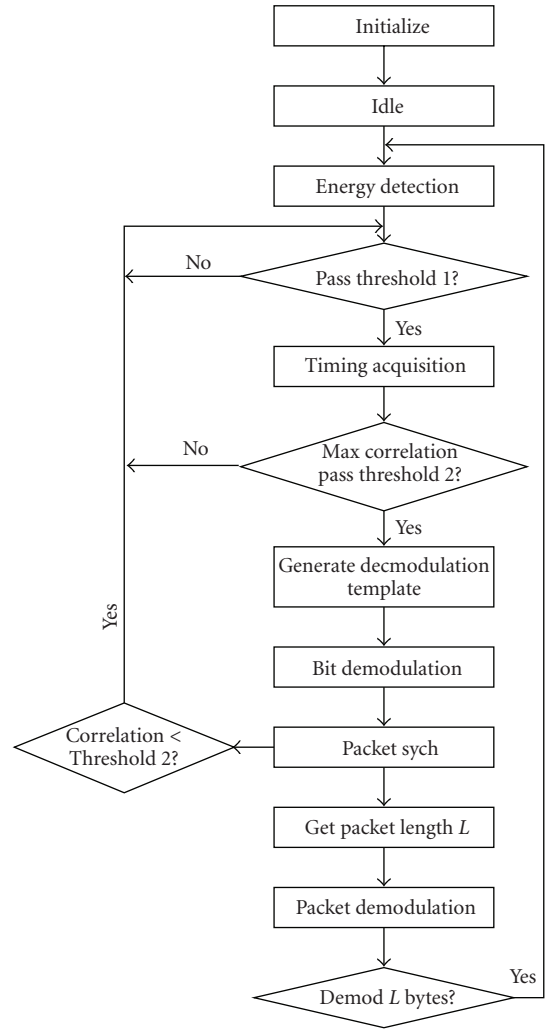

Figure 4: Receiver physical layer processing diagram.

The template used at the timing acquisition phase is the transmitted signal received in the free space. This template incorporates the characteristics of both transmitter antennas and receiver antennas without being affected by multipaths. Such a template can resolve the time delay of the channel very well. Since this template is fixed, it is hardcoded in the receiver.

In our new scheme, the timing acquisition process is carried out in two steps: delay search and code search. Delay search determines the delay at the chip level. The received signal is correlated with the local template and sampled at the end of each chip. After every $N_d = N_f + \max_{i,j} |c_i - c_j|$ sample has been obtained, the template is delayed by $\Delta\tau$, where $\Delta\tau$ is the predefined search resolution and $|c_i - c_j|$ represents the number of chips between two pulses in the transmitted signal. The delay search stops when the maximum value of the N_d samples is above a given threshold Th_1. This happens when the received signal and the template are aligned on the chip boundary.

The code search follows the delay search and further locates the phase of the TH sequence. When the code search

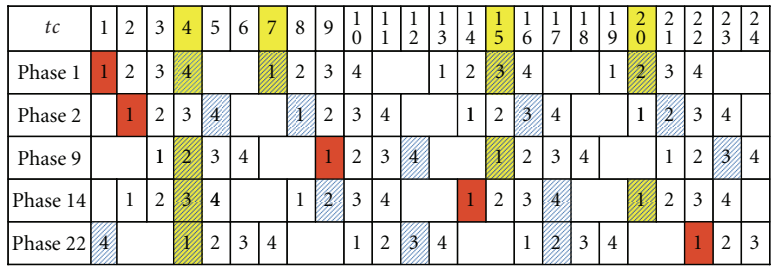

Figure 5: Illustration of the code search process. Assume that the TH sequence is $\{4, 1, 3, 2\}$, $N_{\text{th}} = 4$, and $N_f = 6$. Let the code search start from chip 1. The chips in yellow correspond to when there is a received pulse at the receiver. The shaded chips are the chips when there should be a pulse for the corresponding phase. The red chips represent the bit starting positions.

starts, the chip index is set to 1. At the end of each chip, the correlation between the received signal and the local template is sampled and compared with the threshold Th_1. When the sample value is above Th_1, a pulse is detected in that chip. The pulse count of the phase which allows a pulse in the chip is increased by 1. At the end of a bit, the phase having the maximum pulse count corresponds to the correct phase. In this way, many different phases are searched simultaneously. This process is illustrated in Figure 5. In this example, assume that a pulse is detected in chip 4. Phase 1 expects a pulse in this chip four, but not phase 2. Hence, the pulse count of phase 1 is increased by 1. To reduce the number of counters, pulse counts are only kept for the possible phases after the first pulse is detected. In the above example, the possible phases are 1, 9, 14, and 22.

The entire acquisition algorithm can be implemented through one correlator and N_{th} counters. However it finishes the acquisition process in N_f frames. This search time is the same as the traditional parallel search scheme, which requires $N_{\text{th}}N_f$ correlators in total. More detail about the acquisition algorithm can be found in [16].

3.5. Demodulation. While the template of one transmitted pulse is good for the timing acquisition purpose, it does not capture all the received energy and can result in poor signal-to-noise ratio (SNR). Therefore, we use the received signal within a frame period captured after timing acquisition phase as the template in the demodulation phase. As long as the wireless channel is stationary during a packet period, this template can combine the energy from multipaths coherently and improve the SNR. Such a receiver is suboptimal compared to the RAKE receiver. However, its hardware complexity is much lower too. Since the received signal is noisy, directly using the received signal as the template increases noise and degrades the receiver performance. To reduce the noise in the template, the received pulses from multiple frames are averaged and used as the template. When a frame is long compared to the chip duration, even in rich-multipath environment, the useful signal only occupies a small portion of a frame period. In this case, only the portion of the frame that contains the useful signal is used to generate the template.

3.6. Software Infrastructure. Since this is a data transfer system, the application layer is very simple. The major task at this layer is data handling, including collecting data from sensors and passing data to the ECU.

At the physical layer, the software is also relatively simple. In addition to driving transmitter and receiver hardware and handling the physical layer header in the packet, it also performs bit staffing/removal operation to ensure that no byte in the payload has the same pattern as the SFD.

The MAC layer follows the MAC sublayer specification in the IEEE802.15.4 standard. A superframe is composed of a mixture of contention access period (CAP) and collision free period (CFP). For sensors whose data are very delay-sensitive, one or more guaranteed time slots (GTS) are allocated to it. Other sensors will compete for slots in the CAP.

To reduce system complexity and cost, the software system is designed to be event driven without an operating system. When an event happens, such as when a packet is to be received or to be sent out, a task is created and added to the task queues. Tasks can have high, medium, or low priority. Tasks at one priority level are all finished before any task at the next lower priority level can be processed.

A main timer controls the handling of the tasks. It times out every one unit of the back-off period. Since the backoff period is aligned with the boundary of a slot, using this time period ensures that no slot will be missed. When the timer times out, tasks are processed following the order determined by their priority levels and the time when it is added to the queue. Once all the tasks in the queue are processed, the processor gets into the idle state again until the next timeout happens.

The entire software is partitioned into modules. There are four major modules: a timing module, a radio management module, a service module, and a device driver module. Depending on the complexity of each module, there may be one or more state machines maintained within the module. The timing module is the heart of the MAC sublayer. It fulfils the basic functionality of an operating system and controls the flow of the entire sublayer. The radio management module is responsible for communication handling and provides the interface between the service module and the physical layer. The service module includes two major

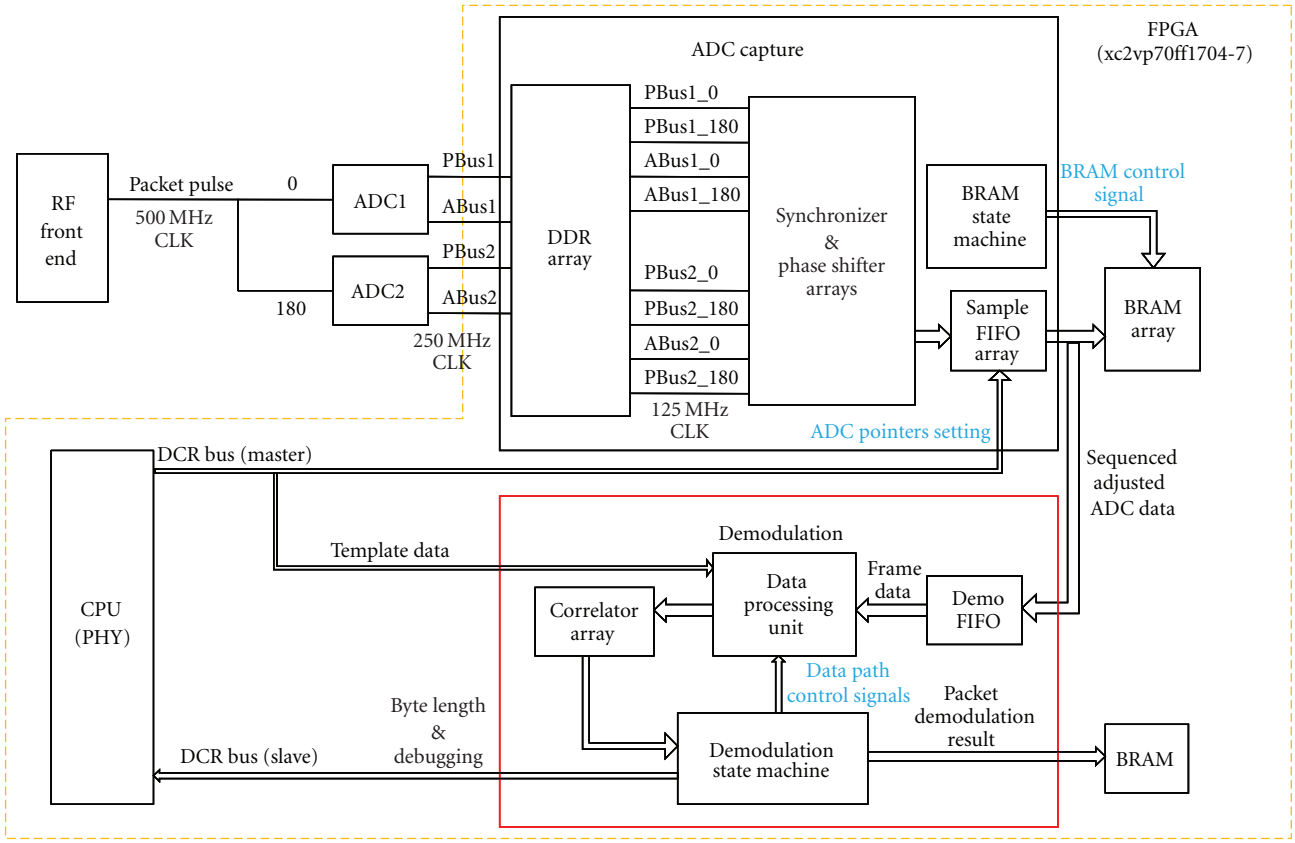

Figure 6: Receiver hardware blocks.

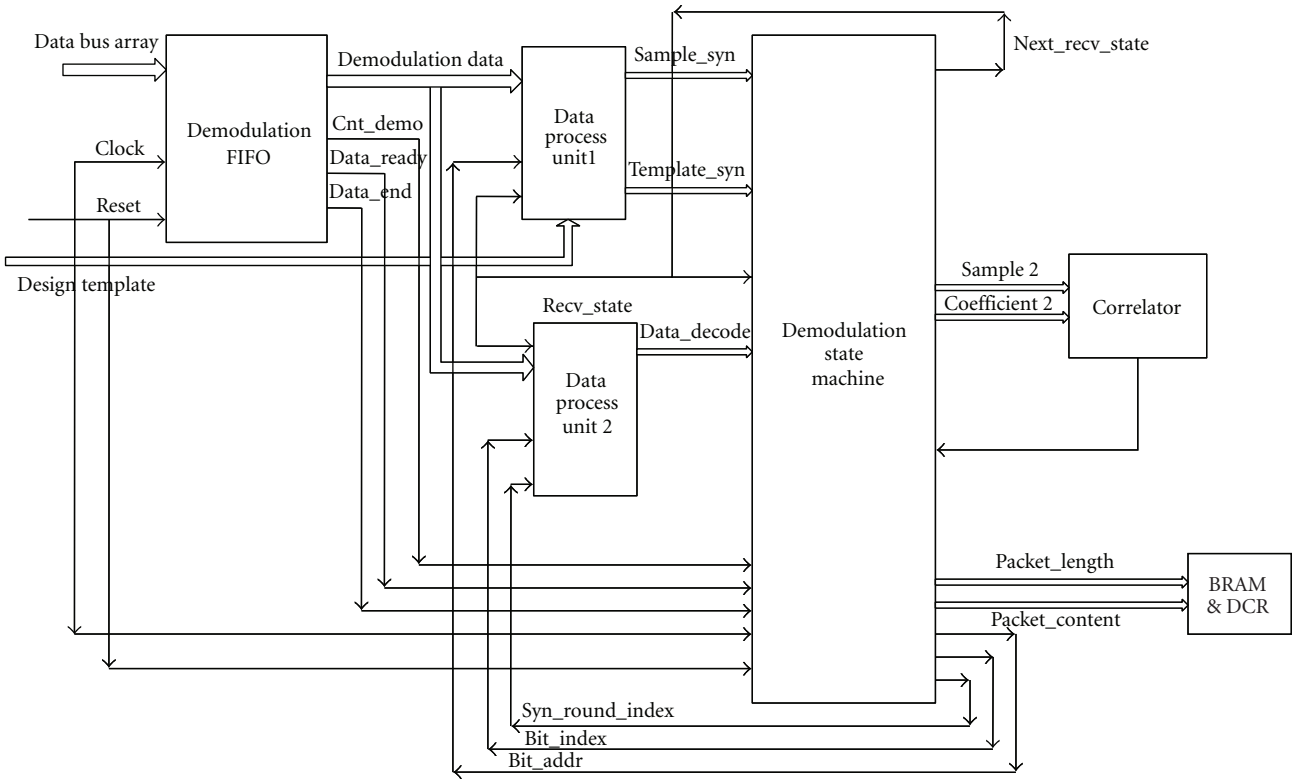

Figure 7: Demodulation unit internal structure.

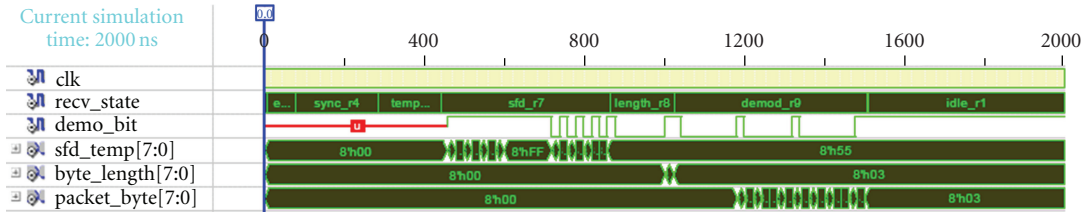

Figure 8: Timing sequence of signals in the receiver.

```
Packet Decoding Result
  BIT_SYNC: 0x 8
  bit_max_corr: 0xEC00
  syn_round_index: 0x 4
  bit_index: 0x 2
  Decoding Template:
  Template_decode[0]: 0x 1; Template_decode[1]: 0x 1;  Template_decode[2]: 0x 1;  Template_decode[3]: 0x 1
  Template_decode[4]: 0x 2; Template_decode[5]: 0x2B;  Template_decode[6]: 0x2B;  Template_decode[7]: 0x2F
  Template_decode[8]: 0x2F; Template_decode[9]: 0x2F;  Template_decode[10]: 0x2E; Template_decode[11]: 0x2E
  Template_decode[12]: 0x2E; Template_decode[13]: 0x2E; Template_decode[14]: 0x2C; Template_decode[15]: 0x 3
  Byte_length: 0x B
  Demodulation Result in BRAM5 --
  i = 00000000 D[02010000]=08070605  005  006  007  008  05  06  07  08
  i = 00000001 D[02010004]=04030201  001  002  003  004  01  02  03  04
  i = 00000002 D[02010008]=00000000  000  000  000  000  00  00  00  00
  i = 00000003 D[0201000c]=000b0a09  009  010  011  000  09  0a  0b  00
```

Figure 9: Packet data decoding result printed from HyperTerminal.

components: (1) the MAC common part sublayer (MCPS) responsible for the data handling of the upper layer, (2) the MAC layer management entity (MLME) that manages other services generally more internal to the MAC layer. The device driver module includes all the low-level interfaces to the physical devices and the I/O peripherals.

4. Receiver Hardware

The transmitter in an IUWB system is very simple. However, this often means more weight would be lifted by the receiver whose design is detailed next.

Since the design of eight-ADC channels is very similar to that of two-ADC channels, the design and implementation details of a two-ADC system are presented in this section. As shown in Figure 6, The ADC capture unit consists of a DDR array, a synchronizer, and phase shifter arrays, a sample FIFO array, and a BRAM state machine. The DDR array in the FPGA reduces the data rate in the ADCs by a factor of 2 to match that of the internal bus in the FPGA. The synchronizer and phase shifter arrays make sure that each internal bus in the FPGA is synchronized with the global clock and there is a fixed phase shift between two buses. The data on each internal FPGA bus will be combined sequentially to form the final data stream. These data are also stored into the BRAM array for debugging purpose. At power up, the order of the output data sequences from the ADCs does not necessarily follow the sequential order. For example, ADC2 may sample first while ADC1 may take the second sample. Through a testing signal, the CPU determines the order of the ADC output data sequence and assigns a pointer to data sequences for them to be combined correctly for future use.

The internal structure of the demodulation unit is given in Figure 7. The demodulation unit consists of a demodulation FIFO, a data processing unit, correlators, and a demodulation state machine. The demodulation FIFO serves as a data buffer and stores the samples from the PLB bus as well as distributes the stored data to data processing units. Data unit1 takes in the received data (from the demodulation FIFO) and the template data (from the memory) and adjusts the data format for the correlation procedure. Data unit2 handles the template reconstruction, SFD searching, and packet demodulation. The demodulation state machine generates the control signal for data processing and computation, as well as controlling the receiver to go through states based on the state machine diagram in Figure 4. The correlator is made of adder trees. It carries out the correlation computation for timing acquisition and bit demodulation.

To help readers understand the receiver state transition, the timing sequence of some signals inside the receiver is shown in Figure 8. The content of the packet is "FFFFFFFF5503010203", where "FFFFFFFF" is the preamble, "55" is the SFD, "03" is the byte length, and "010203" is the payload. All numbers are given in hexadecimal format. The "recv_state" signal tells the receiver state. Soon after the receiver finishes the template generation, valid result begins to appear on "demo_bit" (demodulation bit) and SFD search begins. Once the SFD byte shows up in the "demo_bit", the receiver finds it and the state transfers to the packet length-reading state. The packet length value is one byte

long and is correctly read as "03" in the "byte_length" signal. The packet demodulation phase then begins. After 3 bytes are demodulated, as shown in the "packet_byte" signal, the receiver transitions into the idle state again.

In addition to validating the system through simulators, we also tested the entire system in the lab setup. The first test only checks the physical layer. One transmitter sends out the data packet containing "FFFFFFFF550B 0102030405060708090A0B". The demodulated packet from a receiver is printed out on a computer screen, as shown in Figure 9. (The data bus in the receiver is eight bytes wide, and the data bus in the memory is only four bytes wide. When the data is stored in the memory, the first four bytes and the second four bytes are stored in reverse order. MAC layer handles this problem. Since the MAC layer is not involved in this test, the data in the screen print is shown in reverse order). No MAC layer processing is involved in this test.

The second test checks both the PHY layer and the MAC layer. A hyperterminal simulates a sensor and generates data to be sent out. The data are correctly received at a receiver and printed out.

Since we still have some problem in the antenna design, the tests are carried out through two units connected by a cable. Hence, rich-multipath environments, low-SNR scenarios, and network handling functions in the MAC layer cannot be tested.

5. Conclusion

The design of an intra-vehicular wireless sensor network is presented in this paper. Due to the high reliability required by the automobile sensors and the lack of prior experience, we need a flexible system so that parameters, such as transmission waveforms, and power budgets, can be optimized through tests in real environments. Hence, an SDR system is chosen to implement the design. Our system also proves the concept that a software radio can be realized at the RF band without using any mixers. This proof is useful in the expansion of the application areas of SDR systems and hopefully can also promote the development of SDR systems.

Acknowledgments

This paper is supported by the General Motor's research fund. The authors wish to thank Cem Saraydar and Tim Talty for their support of the project, Daniel Zehnder for sharing his expertise in VHDL design, Hooman Shirani-Mehr for his simulation effort, and David Shu and Simon Han for many useful discussions. The authors would also like to thank the reviewers for their insightful comments.

References

[1] G. Leen and D. Heffernan, "Vehicles without wires," *Computing and Control Engineering Journal*, vol. 12, no. 5, pp. 205–211, 2001.

[2] T. Elbatt, C. Saraydar, M. Ames, and T. Talty, "Potential for intra-vehicle wireless automotive sensor networks," in *Proceedings of IEEE Sarnoff Symposium*, Princeton, NJ, USA, March 2006.

[3] S. Roundy, P. K. Wright, and J. Rabaey, "A study of low level vibrations as a power source for wireless sensor nodes," *Computer Communications*, vol. 26, no. 11, pp. 1131–1144, 2003.

[4] O. K. Tonguz, H.-M. Tsai, T. Talty, A. Macdonald, and C. Saraydar, "RFID technology for intra-car communications: a new paradigm," in *Proceeding of IEEE Vehicular Technology Conference (VTC '06)*, pp. 3008–3013, Montreal, Canada, September 2006.

[5] H.-M. Tsai, C. Saraydar, T. Talty, M. Ames, A. Macdonald, and O. K. Tonguz, "ZigBee-based intra-car wireless sensor network," in *Proceedings of IEEE International Conference on Communications (ICC '07)*, pp. 3965–3971, Glasgow, UK, June 2007.

[6] W. Niu, J. Li, and T. Talty, "Ultra-wideband channel modeling for intravehicle environment," *Eurasip Journal on Wireless Communications and Networking*, vol. 2009, Article ID 806209, 12 pages, 2009.

[7] C. Anderson, *A software defined ultra wideband transceiver testbed for communications, ranging, and imaging*, Ph.D. thesis, 2006.

[8] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pp. 171–180, 2003.

[9] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, p. 224, 2004.

[10] B. Miscopein and J. Schwoerer, "Low complexity synchronization algorithm for non-coherent UWB-IR receivers," in *Proceedings of the 65th IEEE Vehicular Technology Conference (VTC '07)*, pp. 2344–2348, April 2007.

[11] E. A. Homier and R. A. Scholtz, "Rapid acquisition of ultra-wideband signals in the dense multipath channel," in *Proceedings of IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 105–109, May 2002.

[12] S. R. Aedudodla, S. Vijayakumaran, and T. F. Wong, "Timing acquisition in ultra-wideband communication systems," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 5, pp. 1570–1583, 2005.

[13] H. Zhang, S. Wei, D. L. Goeckel, and M. Z. Win, "Rapid acquisition of ultra-wideband radio signals," in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, pp. 712–716, November 2002.

[14] S. Gezici, E. Fishier, H. Kobayashi, H. V. Poor, and A. F. Molisch, "A rapid acquisition technique for impulse radio," in *Proceedings of IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing*, vol. 2, pp. 627–630, 2003.

[15] L. Reggiani and G. M. Maggio, "A reduced-complexity acquisition algorithm for UWB impulse radio," in *Proceedings of IEEE Conference on Ultra Wideband Systems and Technologies*, pp. 131–135, November 2003.

[16] X. Kong and M. Ahmed, "A rapid acquisition method for impulse ultra-wideband signals," in *Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT '10)*, Nanjing, China, November 2010.