Low Power Pulse-Based Communication

Santiago Bortman and Paresa Modarres

***Abstract*— When designing small, autonomous micro-robotic systems, minimizing power consumption by the system's components is an essential design consideration. Mindful of above-referenced consideration, a low power pulse-based communication system is proposed. Transmitting a pre-determined sequence of pulses corresponding to every binary "one" or "zero" enables the receiver to identify the transmitted sequence and recover the data in spite of noise. This approach completely eliminates the need for an RF carrier, which allows the transmitter to consume very little power when idle. The overall system design consists of a pattern and pulse generator for the transmitter, and a comparator and a RAKE receiver for pattern recognition and data recovery. We thoroughly simulated the circuit and then built prototypes using discrete components which were wirewrapped together into a functional circuit. The transmitter was able to generate a particular sequence of pulses corresponding to every transmitted bit. The receiver was able to detect the pulses and decode them back into the original bits of data.**

***Index Terms*—Comparator, RAKE receiver, pattern generator, pulse generator**

I. Introduction

Short range wireless communications often utilize high data rate transfer. Larger data rate communication increases power consumption and might not be ideal for some applications. Applications such as medical sensing inside the human body or small scale robotic communication require minimal power consumption but only modest data rates.

Conventional communication systems modulate data onto a radio frequency (RF) carrier which itself transmits no data. This radio frequency carrier requires constant power consumption, even when the communication channel is idle. However, pulse-based communication systems overcome the necessity of maintaining a carrier frequency by encoding each data bit into a sequence of electromagnetic (E.M.) pulses. Most pulse-based communication systems utilize large data rate by implementing very narrow pulses with ultra-wide bandwidth. In this work, we explore a pulse-based communication system with narrower bandwidth which provides the desired data rate at a much lower power cost. As Fig. 1 illustrates this system incorporates a pattern and pulse generator on the transmitter side and a comparator and RAKE receiver. The transmitter generates a pattern of pulses corresponding to each bit of data. The receiver recognizes the right pattern of pulses and recovers the digital data as previously demonstrated [1].

Design objectives for this system included a carrier-less (baseband) communication, ensuring low power consumption; generation of narrow pulses; encoding of data into sequences of E.M. pulses; and obtaining accurate reception of transmitted data.

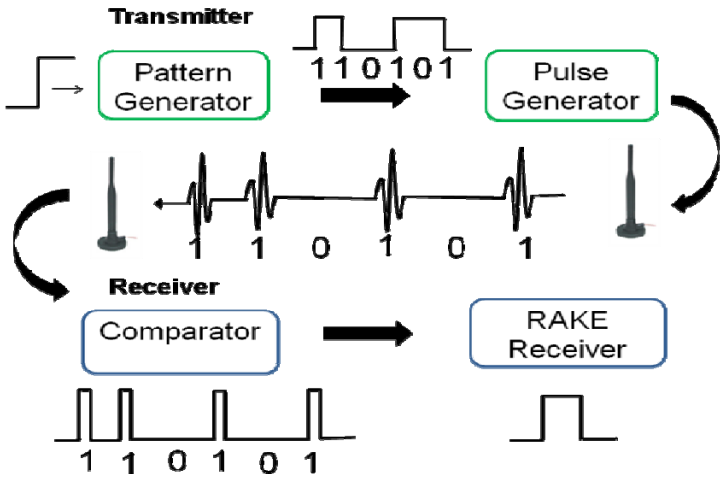

Fig. 1. Design Overview

II. Transmitter

The transmitter is responsible for generating a pattern of pulses corresponding to each bit of data. There is an implicit clock signal every 72ns defining the unit time step. Within each time step there could be a pulse or not. The presence of a pulse will symbolize a “1” and the absence of one would represent a 0. The pattern generator issues a sequence of 1’s and 0’s corresponding to each bit of data. This sequence will be converted into a stream of pulses. Encoding data in this way increases the immunity of the system to noise, even when the noise is higher than transmitted signal. When the receiver operates at a similar clock period of 72ns, it is able to ignore noise, which usually falls in between pulses.

When the input of the pattern generator is toggled high and then low, the edge will ripple through a five stage delay line. To generate a pattern of 1’s and 0’s, each edge corresponding to a “1” is collected using Exclusive OR gates and each edge corresponding to a zero is ignored. Therefore, in the output signal of the pattern generator, each edge corresponds to a “1”. The timing between consecutive 1’s and 0’s is defined by the delay through one stage of the delay line, which is implemented by sequential NAND gates wired as invertors. The output of the pattern generator is sent to a pulse generator.

The pulse generator produces a pulse corresponding to each “1” sent by the pattern generator. The pulse generator consists of four similar branches where each branch is separated by a

delay stage comprising of two inverters. In each branch, the pulse generator creates narrow Gaussian-shaped spikes (named Spike in Fig. 2) corresponding to each edge of the incoming signal from the pattern generator. These spikes are then amplified using inverting and non-inverting op amp configurations to produce signals that are negative or positive, respectively. Then, the four spikes from the four branches of the pulse generator are coupled together to create a pulse as shown in Fig. 2 [2]. The circuit which generates Gaussian-shaped spikes is shown in Fig. 3. In order to generate a Gaussian-shaped spike, the delayed versions of the incoming signal from pattern generator is sent to an AND gate. When the two inputs of the AND gate are simultaneously high, a narrow spike is created [3].

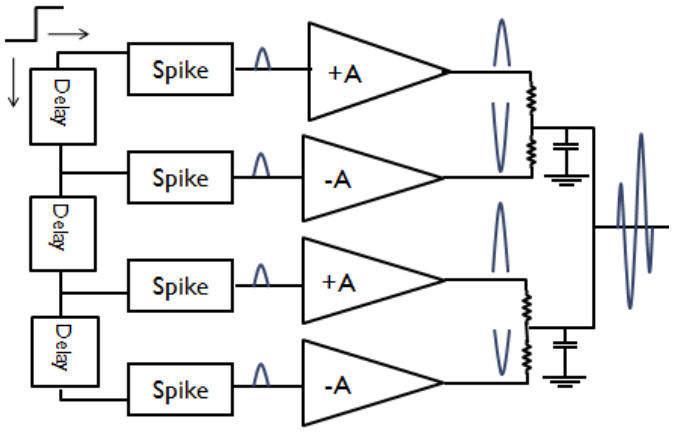

Fig. 2. Pulse Generator: The circuit above generates a sequence of pulses corresponding to each edge of the incoming signal. The Spike circuit is shown in Fig. 3. Each triangle represents an op-amp. Symbols +A and -A represent non-inverting and inverting configurations, respectively.

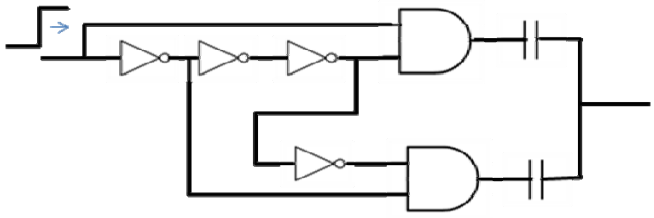

Fig. 3. Spike circuit: The circuit above will generate narrow spikes corresponding to each edge of the incoming signal.

III. .RECEIVER

Data reaches the antenna with a high level of noise and distortion from the environment. It is the receiver's responsibility to interpret the incoming signal and recover the original data. The two main sources of distortion are: 1) white noise which originates from cell phones, appliances, and a large number of electrical devices or natural phenomena; and 2) reflections from walls, trees, etc. known as multipath effect. Multipath could cause a transmitted pulse to appear twice or more in the received signal stream. The problem of white noise can be overcome with the implementation of a comparator. This is a device that sets a threshold for the incoming signal and digitizes it according to whether or not the signal goes over the set threshold assigning a logic value of "1" to all signals above threshold and logic value of "0" to all signals below threshold. Fig. 4 depicts this thresholding operation on an incoming signal.

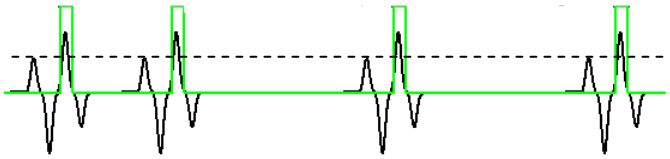

Fig. 4. A sequence of pulses and corresponding digitized output. In this example, the dashed line is the threshold level.

If the amplitude of the pulses is comparable to the amplitude of the noise, it is likely that noise signals will most likely be digitized to logic value of "1" together with the pulses and create false positives. This means that the receiver will have a stream of digital data corresponding to the transmitted pulses plus high amplitude noise. In order to discard these false positives appearing in the recovered data, we implement a RAKE receiver. This device consists of a delay line, which serves to provide a history of the signal's progression through time by taking multiple synchronized snapshots of the signal. The delay line is sampled at every step by a latch that will serve to store the value at that point in the delay line on every positive clock edge. The sampling clock period should match the period between pulses (72ns), which should also be the same as the total delay through the delay line. That way the entire signal will be captured every clock period (72ns) with intermediate time steps sampled along the different fingers of the RAKE receiver (see Fig. 5). Matching these parameters allows the clock to fall out of phase with relation to the incoming pulses without having pulses pass through the delay line and not be sampled and stored by any latch. The output of every latch in the first column will feed into the input of a latch in the next column. In this way, the data will move from one column of latches to the next on every clock edge, as the first column captures a new set of data and the last column discards the oldest set of data.

With this configuration, every pulse (or high amplitude noise event) will be captured by a latch and passed on to subsequent latches. Since the clock period and the time between pulses match, every sent pulse will be sampled by the same row of latches (called a finger) in the delay line. This means that the digital data corresponding to the transmitted pulses will all be captured in one row of latches and noise will most likely fall in the others. Once the data is captured in this structure the values stored in every row of latches are compared with a pattern stored in memory in order to detect a bit; if there is a match, then the data received by that latch is indeed data sent by the transmitter and not noise. Each correct pattern is then converted back into the binary bit it represents, and the data is recovered. At the same time, if the values stored in the latches do not match the pattern stored in memory, the data is taken to be noise. No action is taken in this case; newer data will enter the RAKE pushing the values that fail to match to the next latch until they are flushed out the end of the rake.

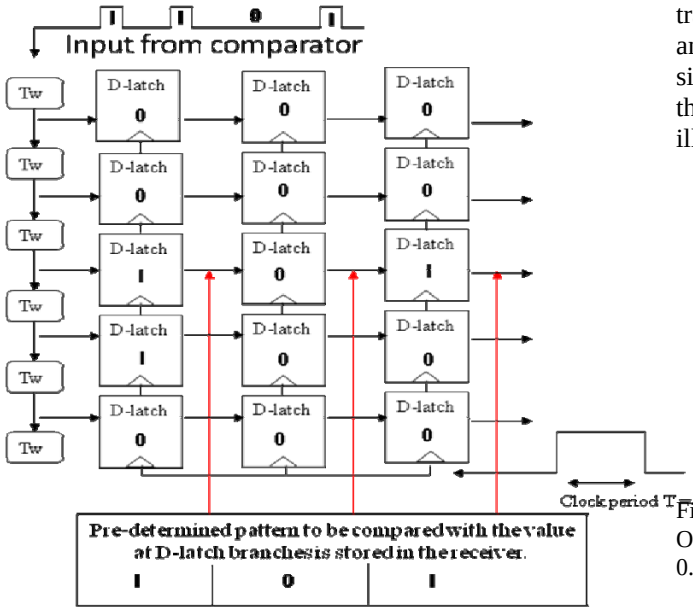

Fig. 5. RAKE receiver. The rake receives digital inputs from the comparator. These inputs move down the delay line on the left and are sampled by latches. The information moves right until it reaches the end and is discarded. There is an ongoing comparison to check if at any time data present in the latches matches the pattern. If so, the system will output logic value of '1', if not logic value of '0'.

IV. Experimental Results

The transmitter and receiver were built using discrete devices wire-wrapped on a circuit board. One important issue in simulations and experimental results of the transmitter was the timing between generated pulses. Since in most devices, the pull-up and pull-down times may be different, the periods between consecutive pulses may differ by a few nanoseconds. This issue could affect the receiver, since the RAKE is sampling the digitized pulses at the same period as the period between pulses, mismatch in periods could cause the receiver not to detect a pulse; or cause consecutive pulses to fall into different rows of latches. To overcome this timing issue, the digitized pulses were made wider by the comparator stage in order to allow them to drift slightly in time without failing to appear at a given stage in the delay line.

Another minor issue encountered in testing of the transmitter was the timing of the delay line. The delay time of the delay line was three times faster than what was expected from simulation results. To solve this problem, the components (74F logic series) used to build the delay line were replaced with slower devices. In addition, the pulse generator was configured in simulations to produce narrow spikes with a width of about 5 ns. However, generating these spikes was infeasible using discrete devices. To solve this problem, minor changes were made so that the delay between the two inputs of the AND gate (see Fig. 3) responsible for producing the spikes would be longer. Correspondingly, the generated spikes have longer width of about 8 ns.

The rake was wired using 74F series devices such as Inverters, Exclusive OR and AND gates. The comparator was implemented using a fast operational amplifier, two transistors, a capacitor and a resistor. The operational amplifier was used to set the threshold and compare incoming signals while the rest of the elements where used to modify the width of the obtained pulse to the desired size, as illustrated by Fig. 6.

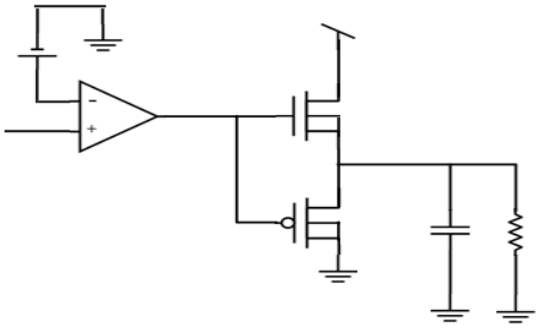

Fig. 6. Implementation of the comparator. The negative side of the Op-Amp is the threshold voltage which will be swept from 3 volts to 0.2 volts.

Under low noise conditions, the device was able to detect incoming pulses with ease and store the values with almost 0% false positive readings appearing in the RAKE. Comparisons between values present in the latches and stored patterns are performed using Exclusive OR gates. Each gate compares the data in a specific latch and the complement of the stored data corresponding to that column. If there is a match it will output "1". If all the gates in a given row detect a match, the whole pattern must have appeared in that row. A simple 6 input AND gate serves to detect this event and output one when a whole row matches. By joining all the outputs of the AND gates with an OR gate, a single node is obtained that goes high when the pattern is detected anywhere in the RAKE. Fig. 7 shows that the comparator can digitize the signal with ease and the rake is able to decode the digital signal back into the original binary data when a clean signal with correct patterns is fed into the input of the receiver.

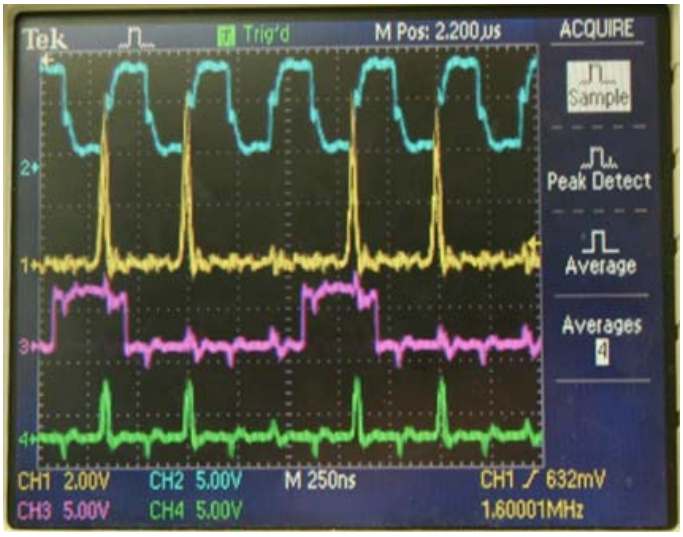

Fig. 7. A clean signal enters the receiver (110110). It is digitized and sent to the RAKE. A row of latches captures the data (not shown). The data is matched with the stored pattern and the RAKE outputs a

one every time it sees the pattern. Legend from top to bottom: clock, incoming signal, output, and intermediate step in the delay line.

The presence of noise in a signal introduces uncertainty as to the appropriate value for the threshold that should be applied to the comparator. If the threshold is set too high, the pulses will pass through without being detected; if it is set too low, random noise will appear as signal pulses.

When a noisy signal is fed into the input of the receiver, the comparator must sweep the threshold voltage downward starting from a high voltage. As the voltage is swept down, the tallest peaks in the incoming signal will be digitized to a logic value of "1". If the tallest peaks are the transmitted pulses, the rake will detect the right pattern appearing at the latches and the sweep will conclude. If no discernable patterns appear, the threshold voltage should be dropped further; this will cause the pulses to eventually be digitized to a one together with a series of false positives due to noise. The spikes due to noise will happen at random times, and will not follow the same period as the clock. By clocking the latches at the right frequency, we will ensure that all transmitted data falls in one row of latches (see Fig. 8) and that the noise is distributed amongst the other rows. The chance of spurious noise bits falling in the row that is capturing the data is then one over the length of the pattern times the total number of rows. Therefore, the larger the number of rows in the RAKE, the smaller the chance of data corruption.

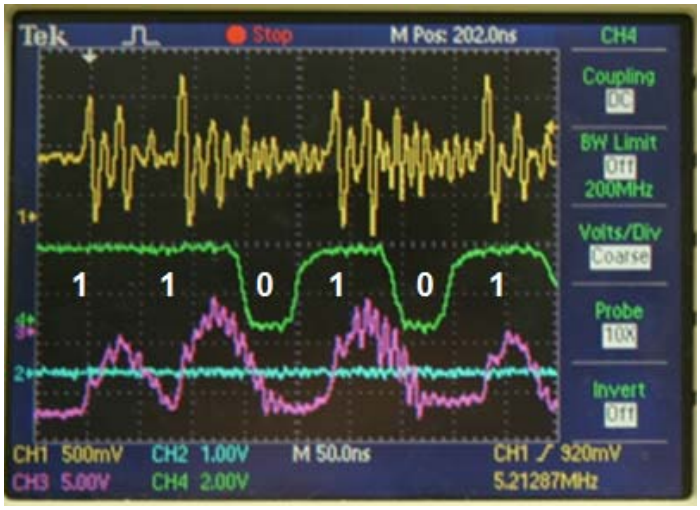

Fig. 8. The output of the transmitter is directly connected to the receiver. The RAKE is able to digitize the incoming signal to the correct pattern in spite of a high level of noise.
Legend from top to bottom: incoming noisy signal, recovered digital pattern (output node of one latch), output of the comparator circuit, and the flat line is the threshold.

V. Conclusion

Implementation of the device using discrete components proved that the desired operation of the system is possible. Data was transmitted using a 6 bit coding scheme for every bit of information. This scheme yielded pattern recognition and data recovery. The discrete components introduced speed limitations in many parts of the circuit, restricting generation of short pulses and high speed comparisons.

Further tests should be carried out to determine the interference and noise that transmission through antennas would add to the system, and determine a range of operation for the system. Work should also be done in trying to minimize the temporal litter between pulses, in order to increase the chances that a particular bit of transmitted data will be received and detected by the RAKE.

Implementing the circuit using integrated circuit technology would allow for faster, smaller, and more accurate systems. By correctly sizing the width of the transistors in the pattern generator, we could fix the problems of different pull-up and pull-down times, yielding a series of pulses with well matched periods. A longer delay line with additional rows of latches would allow for better noise and multipath immunity by lowering the chances of false positives falling in the same row as the data being collected.

Appendix

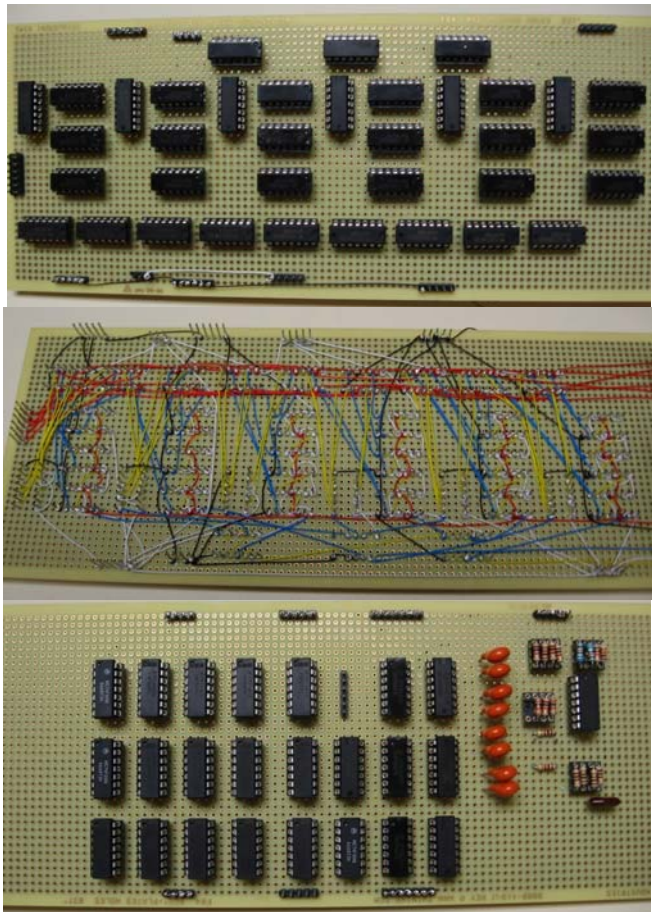

Fig. 9. The receiver and transmitter circuit board. Top, a portion of the RAKE receiver. Middle, the backside of the RAKE receiver. Bottom, the transmitter.

ACKNOWLEDGMENT

We would like to thank Dr. Pamela Abshire, our faculty advisor, and David Sander, our graduate advisor, for their technical help and support in different aspects of this project.

REFERENCES

[1] M.C Munshi, J. Jiang, Y. Xin, T.S. Lande, and Y. Lian, "An efficient synchronization scheme for digital UWB communication systems for biomedical applications," *in Proc. of Biomedical Circuits and Systems Conf. (BioCAS 2008) Nov.2008.*

[2] H. Kim, Y. Joo, and S. Jung, "A Tunable CMOS pulse generator," *in Proc. of 2006 IEEE Int'l Conf. on Ultra-Wideband, Waltham, Massachusetts, Sept. 2006.*

[3] B. Jung, Y. Tseng, J. Harvey, and R. Harjani, "Pulse Generator Design for UWB IR communication systems," *in Proc. of IEEE Int'l Symposium on Circuits and Systems (ISCAS), Kobe Japan, May 2005.*

[4] A.M. Shankiti and M. Leeser, " Implementing a RAKE receiver for wireless communications on an FPGA-based computer system," *in Proc. of the 2000 ACM/SIGDA eighth Int'l symposium on Field programmable gate arrays, Monterey, California, Feb.2000.*

[5] N. Xie and Y. Xhou, "An adaptive nonlinear rake receiver in UWB wireless communications," *in Proc. of IEEE Int'l Conf. on Networking, Sensing and Control (ICNSC 2006), Aug.2006.*