ISSN NO: 9726-001X
Volume 11 Issue 02 April 2023

The Use of a Bipedal Robot for Wireless BAN

Dr.G Sankar , Mr.D Sanjeeva Reddy , Mr.P Viswanatha Reddy
Associate Professor[1] , Assistant Professor[2,3]
Department of CSE,
Viswam Engineering College (VISM) Madanapalle-517325 Chittoor District, Andhra Pradesh, India

Abstract

In this research, we provide a way for "wiring" a two-legged robot without the need of actual cables. Significant weight reductions are possible and cable failures caused by moving parts are eliminated when a "body area network" (BAN) is built using radio frequency (RF) technology. The resulting system is fast to update and has low latency and great throughput, all while being wireless. Energy consumption has utilized single-chip transceivers that are readily accessible on the market. Two different network topologies have been evaluated and compared for their efficacy. To continually gather and analyze 3D acceleration information from sensors mounted on the robot's seven limbs, a custom-built FPGA multi-core CPU using BAN technology. The FPGA might potentially use the same BAN to send instructions back to the limbs.

1. Introduction

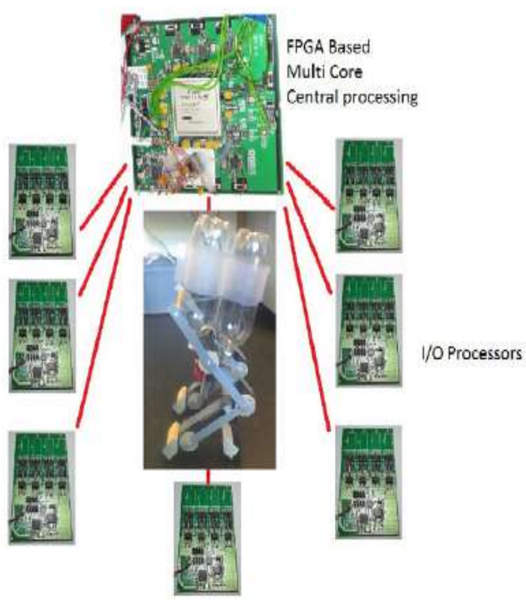
Playwright Karel Capek's brother Josef first used the word robot [1] in a play he wrote and published in 1920; the term is taken from the Slavic word robot, which means forced labor. Although not all modern artificial robots have a humanoid form, walking on two legs is the most natural mode of mobility for them to use. Where machines must function side-by-side with people. However, bulk (or weight) continues to be an issue; the most current iteration of Honda's renowned ASIMO [2] is just 48 kilograms, whereas Boston Dynamics' PETMAN [3] is 80 kilograms (or the equivalent of two adults and a toddler). A robot weighing more than 50 kg poses safety concerns in a human setting due to the extra energy it needs to simply move its own weight. Given this, it's easy to see why our strategy of adopting a very minimal framework is so appealing. Our robot, dubbed "Monkey," is a lightweight and inexpensive biped capable of running at speeds similar to human jogging. Only a handful of other running robots have been built so far, and none of them adhere to our basic design ethos. Uses for Monkey range from supporting people with their everyday lives to medical diagnosis and treatment.

ISSN NO: 9726-001X
Volume 11 Issue 02 April 2023

Figure 1. The bipedal robot Monkey, which may be used for sports practice. Pneumatic actuators are described for a bipedal robot in [4]. However, we are optimistic that our new Ribbon Air actuators would outperform their McKibben actuators in terms of contraction ratio. Proteins that build muscle (RAM) [5]. Powered by a 6l air reservoir and 1Ah 11.1V lithium-ion battery, Monkey (Fig.1) is a self-contained device. At a modest 76 centimeters in height, the monkey weighs in at a little 4 kilograms. A microcontroller with the ability to transmit radio frequency (RF) signals is mounted on a tiny printed circuit board (PCB) at the end of each limb. The MCU's analog-to-digital converter (ADC) is hooked to a three-axis accelerometer, and the MCU's digital output pins are connected to output driver ICs and fly back diodes, which power the valves. Each of the robot's seven legs has a printed circuit board (PCB) attached to it: the left shin, the right shin, the right shank, the left shank, the left hip, the right hip, and the pelvis. From here on out, we'll refer to these PCBs as nodes or radio frequency nodes (RF-nodes). Each limb is powered by a pair of opposing air muscles. While each muscle is coordinated by its own node, the FPGA multi-core CPU is responsible for the overall coordination of the Monkey system. Our team has created this unique network-on-a-chip architecture [6] as part of a larger Microsoft-funded, high-level hardware/software co-design initiative.
The co-design tool-chain takes the form of Active Cells, a high-level programming language designed to specify all three architectural levels of a multi-core system (a) the entire chip-architecture, (b) the processor-core architecture, and (c) the code running on the individual cores) in a single, coherent program text. Tiny Register Machine (TRM) is a lightweight, custom-designed processor-core that may be used in conjunction with dedicated hardware components to perform time-critical aspects of our project. In the following, we will elaborate on the role that the communication system plays in this overarching architecture.

2. Communication

Our primary concerns for the networked-communications infrastructure are:
Six-legged angular measurements (strain sensor/hall sensor/potentiometer) Two-foot pressure sensing With a low latency and a refresh rate of at least 100 hertz, a three-axis accelerometer on all six limbs, and at least two drive signals per limb (antagonistic air muscle pair),



As shown in Fig. 2, the robot's BAN consists of seven nodes and a single FPGA.
More than 60 cables would be needed to link everything in a wired setup. (In a related research, our colleagues at ETH Zurich have constructed a quadruped [7]. To get a refresh rate of 100 hertz, they ran two CAN-buses in simultaneously. Rate and only the wiring add 5kg to the whole weight.) Our robot architecture is set apart by its a) distributed computing and b) wireless communication features. The following advantages result from these choices:

ISSN NO: 9726-001X
Volume 11 Issue 02 April 2023

Benefits from distributed computing

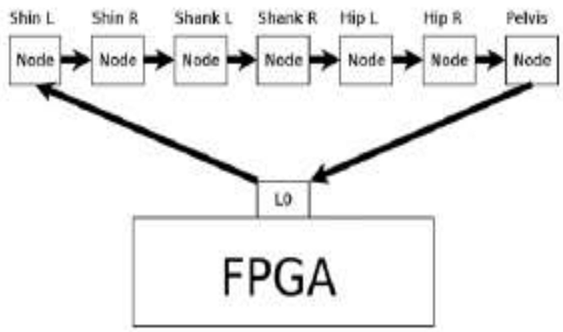
• The data is handled and preprocessed locally at the same location where it is measured. This eliminates the potential for signal distortion during wiring and eases the load on the central processor.

• The distributed architecture provides a measure of freedom to the appendages. Based on biological principles, in the sense that it incorporates into the walking pattern movements that are directly processed by the spinal cord (i.e., reflexes) [8].

Benefits from wireless communication

• The data is handled and preprocessed locally at the same location where it is measured. This eliminates the potential for signal distortion during wiring and eases the load on the central processor.

• The distributed architecture provides a measure of freedom to the appendages. Based on biological principles, in the sense that it incorporates into the walking pattern movements that are directly processed by the spinal cord (i.e., reflexes) [8].



3 BAN in a ring topology illustration

The following RF-specifications are used by us:

CRC checksum or cyclic redundancy check

• Positively acknowledged packets, or ACKs

Retransmission Occurs Immediately Throughput of 2 Mbps per channel, with a maximum payload size of 32 bytes Each RF-node is managed by a distinct core inside the FPGA multicourse processor through an SPI link. A pattern technique is implemented on top of the communications architecture to control the behavior of the nodes (limbs). Each node has its own clock yet all nodes are powered by the same battery. All nodes are activated simultaneously when battery power is turned on, and they all operate in sync after being activated. (The startup time of each node is the same.) Movement commands consequently may relate to the status of an internal clock counter. Each time slice in a pattern lasts 10 milliseconds. If a node receives the commands "30 flex; 100 relax; 150 extend... ", it will do the following series of operations indefinitely, with the length of the loop set dynamically. For 300 milliseconds, flex the corresponding joint; for 700 milliseconds, relax it; for 500 milliseconds, extend it again. Without any input from the main processor, the nodes will keep cycling through the established pattern indefinitely. The pattern technique relieves the main processor from the burden of maintaining a constant motion. Instead, its computing capacity may be put to work learning and optimizing the patterns of the various limbs. We are well aware that the clocks will eventually go out of sync. The tests revealed a discrepancy of around one second every 24 hours. However, this is not a concern given that nodes only operate for brief periods of time. The FPGA can shut off the nodes whenever the robot is idle. We haven't gotten into the specifics of the Monkey robot's network structure just yet. In the following sections, we will describe and contrast two tested and implemented alternatives: a ring topology and a star topology.

3. Ring Topology

Each node in the BAN on Monkey only communicates with its immediate neighbors through a ring-shaped network topology (Fig.3). In this scenario, just one node has to be linked to the central computer. An off-the-shelf FPGA-development board was used to realize this architecture. Using spare FPGA pins to attach an RF interface (Nordic L01).

ISSN NO: 9726-001X
Volume 11 Issue 02 April 2023

Accelerometer data is continually retrieved and stored by the nodes. (ADC values). As soon as a matching RF-packet arrives, the appropriate node updates the packet with the most current full set of ADC data. There are two buffers involved in this system. All of the nodes in Fig. 4 have the same payload.

Fig. 4. The graphic demonstrates how the acceleration readings are kept in the payload. The X-axis (blue), Y-axis (red), and Z-axis (magenta) all use 12 bit measurements. There are six blue-red-magenta-blocks related to six sensors. Data from the seventh sensor (green) is striped throughout the whole payload in order to fit under the payload's 32-byte restriction. Only four bits out of the original 32 bytes remain.

Figure 5. The "monkey" suspended by a steel rope

All seven nodes' three 12-bit measurements (X, Y, Z) just fit into a single packet, but there's nearly no space left for more data due to the hardware constraint of the packet size to 32 bytes. The system is only good enough to create a basic kind of walking based on patterns of length 4; however this constraint is shared with the pattern size. Monkey is hanging by a pulley and moving with elegance thanks to this mechanism. Monkey is able to walk back and forth along the rope because the pulley moves along a horizontal steel rope (Fig. 5). Due to the automated retransmission, data loss during regular environment testing (including in the presence of other wireless networks) was less than 1%. (The nodes had to be enclosed in metal bowls to simulate packet loss.) A node resends the packet to the next node when it hits its retransmission limit, in this case 15. For the duration of one cycle, the failing node's sensors will not send any data. A temporary loss of RF connection prevents the delivery of any new orders to a node, which instead keeps repeating the last pattern it received. Due to the frequent updates, lost data packages are seldom a major issue.

The ring structure is supported by an FPGA design (Fig. 6) with four TRM cores, each of which performs a distinct function:

The Assignment component specifies the subsequent steps to be done. Switched on and off by the user.

• Radio frequency (RF) communication: The "Monkey" processor core's GPIO pins link to a single Nordic L01-chip that handles wireless I/O. It's reading 3D acceleration values and transmitting motion patterns to the robot.

• Data processing: Future work will make use of the clustering-core for data processing. By aggregating sensor data into clusters, dimension reduction occurs. The connection back to the assignment hub is not active at the moment. Because of this, the walking cycle is open-loop regulated rather than closed-loop.

• Display — Provides feedback to the user through an LCD and LED combination. Integrated to show the current packet count and the user's choice of limb angles.

ISSN NO: 9726-001X
Volume 11 Issue 02 April 2023

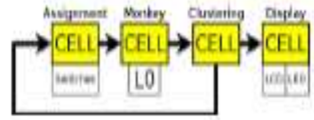
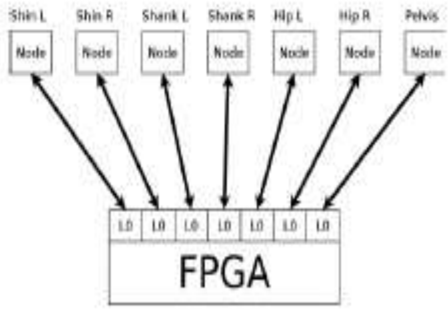


Fig. 6. Core structure for the ring communication system



Star-shaped BAN topology (Figure 7)

It's worth noting that there's a big difference between FPGA-to-FPGA connection and external communication. All of our examples and descriptions include data transfer that occurs outside of the FPGA. On the subject of conversation, read on! The FPGA (linking the CPUs) is described in [10]. A stop-and-go protocol, in which the FPGA repeatedly transmits a packet and waits for it to return after passing each node, was first created. The maximum refresh rate of 72 Hz was reached in the default configuration.

It is possible to increase the refresh rate to 90 Hz by using pipelined mode (sending many packets before the first one comes back). According to the profiling results, the SPI connection between the FPGA and the RF-node is rather time consuming. A final refresh rate of 166 Hz was attained by raising the SPI speed.

4. Star-Shaped Topology

We redesigned the BAN on Monkey to have a star-shaped topology (Fig. 7) in which every node is in direct contact with the FPGA. To do this, a special FPGA board with seven integrated L01 chips was developed. It's easy to see why this topology might be advantageous: The whole payload is now available to each node. Size, which enables the transfer of intricate designs and supplementary measures. Six more cores (Fig. 8) were installed in the FPGA to accommodate the L01-chips. There are now 10 TRM cores in the main processor of the FPGA. The RS232 interface and the LEDs are used to provide feedback to the user from the display core. The custom-made board is smaller, lighter, and cheaper than the development board we were using previously, but it lacks an LCD display. Up to 30 TRM cores may be installed in the FPGA. Higher refresh rates, up to 313 Hz using a stop-and-go protocol, are possible because to the shorter transmission cycles in a star-shaped topology. Each RF-node chooses its own channel to listen on from a total of 63. With the help of the parallel channels, we were able to increase the data transfer rate from the FPGA to the nodes to 1961 packets (32Bytes each) per second. Additionally, fault-tolerance is enhanced by the availability of channel hopping. If retransmission fails, a node might dynamically switch to a different channel to try sending the packet again. From zero to fifteen retries may be set before a failure is notified. If a node detects a packet loss, it will retransmit the failed packet to a different RF-node (on the same FPGA-PCB) and will also transmit the new channel number it will listen for. After the FPGA receives the packet, it notifies the appropriate core of the channel-hop. Each node knows about every hop, thus it can keep track of all the active channels.

ISSN NO: 9726-001X
Volume 11 Issue 02 April 2023

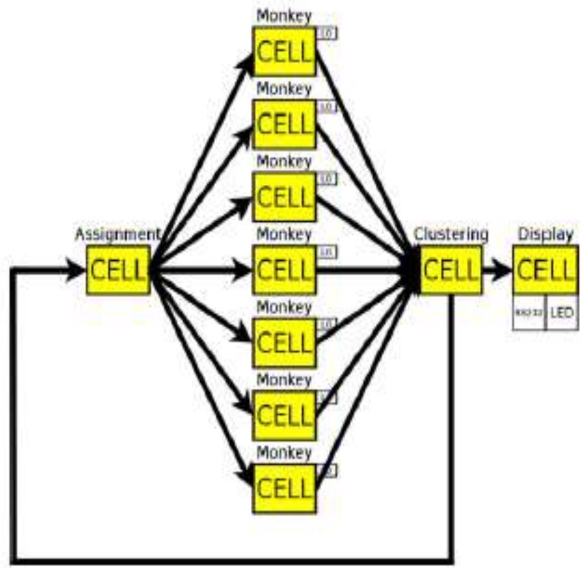


Fig. 8. Core Structure on the FPGA for the star topology

5. Results

The following contrast summarizes the acquired findings so far:

	Ring	Star
Refresh rate	166Hz	313Hz
Packets/s	166	1961
Pro	• Simple hardware • Simple RF handling • Low power consumption	• Low Latency • Full payload (32Bytes) available for each node • 7 channels in parallel • High throughput • Nodes can dynamically hop to another channel in the case of packet loss
Contra	• Higher Latency • Payload shared between all nodes • Very limited amount of data • Low throughput	• FPGA has to handle 7 connec-tions • Complex hardware • Danger of RF interferences

6. Summary

We have developed a cheap and lightweight bipedal robot that operates only using wireless technologies. A multitude of peripherals and a custom-built FPGA multiprocessor (the "brain") make up the hardware system.

Figure 9: Node board for a custom FPGA (the "arms").

Based on a pattern approach, a hybrid centralized-distributed control architecture has been built. Each limb node operates independently while the FPGA controls the entire system. Processes data in its own unique way. The ring topology and the star topology are two examples of communication structures that have been built and compared. For our future work, we find that the star-shaped form is the best option, since its usage of parallel RF-channels allows for a refresh-rate of more than 300Hz and a quick and effective fault tolerance mechanism. However, adding an RF-chip to the FPGA-PCB for each FPGA-to-node connection limits the scalability of the tight star architecture. A viable solution would be to merge the two architectures, with many node rings linked to the FPGA in a star configuration. Given this, we conclude that 49 (7 7) nodes, each with a refresh rate of 166Hz and a payload of 36Bits, would be optimal. This would amount to an RF data-rate of 716631.5Bytes/s, or 35.6KB/s.

References

[1] K. Cˇ apek, R.U.R (Rossum's Universal Robots), Echo Lib, 2010.
[2] T. Yoshikawa, O. Khatib, Compliant motion control for a humanoid robot in contact with the environment and humans, in: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 2008, pp. 211 –218. doi:10.1109/IROS.2008.4651155.
[3] J. Hodgins, M. Raibert, Adjusting step length for rough terrain locomotion, Robotics and Automation, IEEE Transactions on 7 (3) (1991) 289 –298. doi:10.1109/70.88138.
[4] K. Hosoda, T. Takuma, A. Nakamoto, S. Hayashi, Biped robot design powered by antagonistic pneumatic actuators for multimodal locomotion, Robot. Auton. Syst. 56 (2008) 46–53. doi:10.1016/j.robot.2007.09.010.
[5] D. Majoe, L. Widmer, J. Gutknecht, Pneumatic air muscle and pneumatic sources for light weight autonomous robots, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 3243 –3250. doi:10.1109/ICRA.2011.5979796.
[6] L. Liu, F. Friedrich, O. Morozov, Y. Han, J. Gutknecht, P. R. Hunziker, Automatic soc design flow on many-core processors: a software hardware co-design approach for FPGAs, in: FPGA, 2011, pp. 37–40.
[7] M. Hoepflinger, C. Remy, M. Hutter, L. Spinello, R. Siegwart, Haptic terrain classification for legged robots, in: Robotics and Automation (ICRA), 2010 IEEE International Conference on, 2010, pp. 2828 –2833. doi:10.1109/ROBOT.2010.5509309.
[8] C. Torres-Huitzil, On the implementation of central pattern generators for periodic rhythmic locomotion, in: Reconfigurable Computing and FPGAs, 2009. ReConFig '09. International Conference on, 2009, pp. 422 –426. doi:10.1109/ReConFig.2009.81.
[9] The official Bluetooth sig member website. URL https://www.bluetooth.org/Technical/Specifications/adopted.htm
[10] F. Friedrich, L. Liu, and J. Gutknecht, Active cells: A computing model for rapid construction of on-chip multi-core systems, in: ACIS-ICIS'12, 2012, pp. 463–469.