Scope International Journal of Science, Humanities, Management and Technology. ISSN : 2455-068X
Vol.2 Issue 2 (2016) 80-84. Submitted 28/03/2016. Published 05/04/2016

DESIGN OF WIRELESS SPY LANDROVER CONDROLLED BY SMARTPHONE

[1]A. Roshan Ranjan, [2]P. Sivaprasath, [3]S. Santhanalakshmi, [4]M.Sudhakaran,
[1,2],Dept of EEE, GTEC Engg. College, Vellore,
[3]Asst. Prof, Dept of EEE, GTEC Engg. College, Vellore,
[4]Associate Prof, Dept of EEE, GTEC Engg. College, Vellore.

Abstract:
The objective of this paper is to show that it is possible to create a single Android application capable of working with a number of electronic devices typically used within the hobby and armature robotics field, without the devices creator having to know anything about developing an Android application. To do this, a standard communication protocol must be established between Android powered devices and other electronic devices. To limit the scope of this task, this paper considers communication between an electronic device powered by a typical microcontroller and an Android 4.0 (Jelly Bean) or later powered device. Additionally communication between the two devices takes place over Bluetooth communication channels V2.1 or later.

Keywords – Android, Robotics, Scope, Microcontroller.

1. INTRODUCTION

In this project we are trying to establish both wireless communication between the mobile Robot and the remote Base Station, and serial communication between the remote Base Station and the GUI Application. The Base Station requires the serial communication with the GUI Application and also needs to be hardwired with the radio packet controller, FRPC2 for wireless control. Our aim is to be able to command and control the Robot wirelessly by the GUI Application.

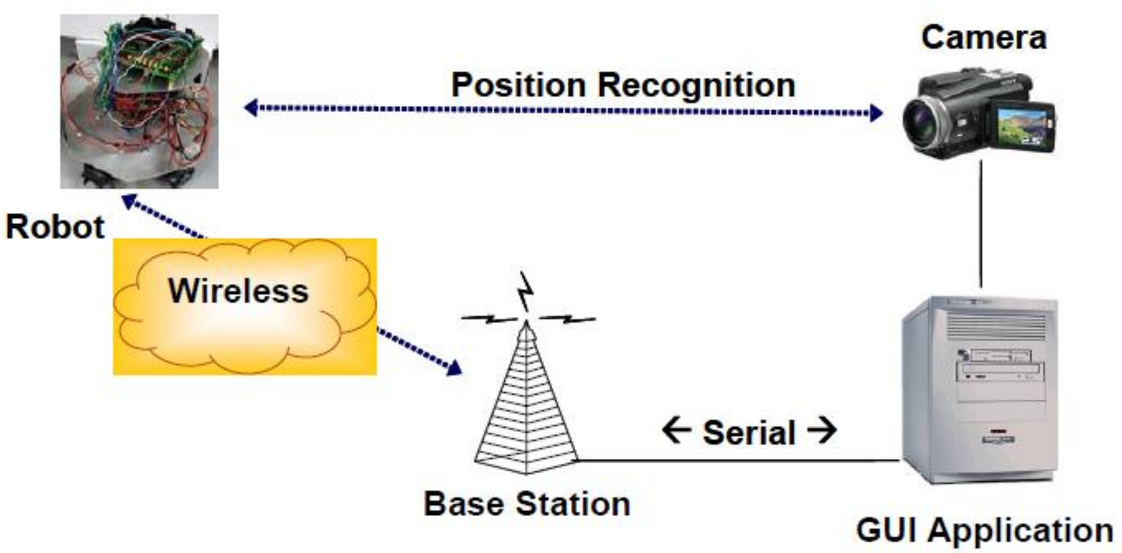

Fig.1.System structure

The main task of this project is two parts: (1) to program the AVR microcontroller on both the Base Station and the Robot interfaced to the radio packet controller module which would enable us to wirelessly control the Robot; (2) to program the GUI Application which would enable us to serially control the Base Station. Theoretical system limitation for the packet transmission is evaluated and analyzed. We tested packet stress to the wireless module while varying the number of Robots and the payload data. The wireless parts were evaluated with CRC error checking. As a result, we achieved control both wireless communication between the mobile Robot and the remote Base Station, and serial communication between the remote Base Station and the GUI Application. This level of completely was successfully tested on groups at up to four Robots. Hence the wireless communication and the serial communication were successful in the downlink. The Base Station requires serial communication with the GUI Application and also needs to be hardwired with the radio packet controller, FRPC2 for wireless control. Our aim is to be able to command and control the Robot wirelessly by the GUI Application

2. DESIGN REQUIRMENTS

The previous system using the RPC is susceptible to interferences caused by the sharing of same frequencies. Hence we need to find a new system that is not only preventing such occurrences but also reliable during the uncertain conditions. The system should also have error-checking capabilities. With regard to Performance, since our Robotic system is real-time, there is a need for a low latency system, which is affected by the delay in processing and transmission.

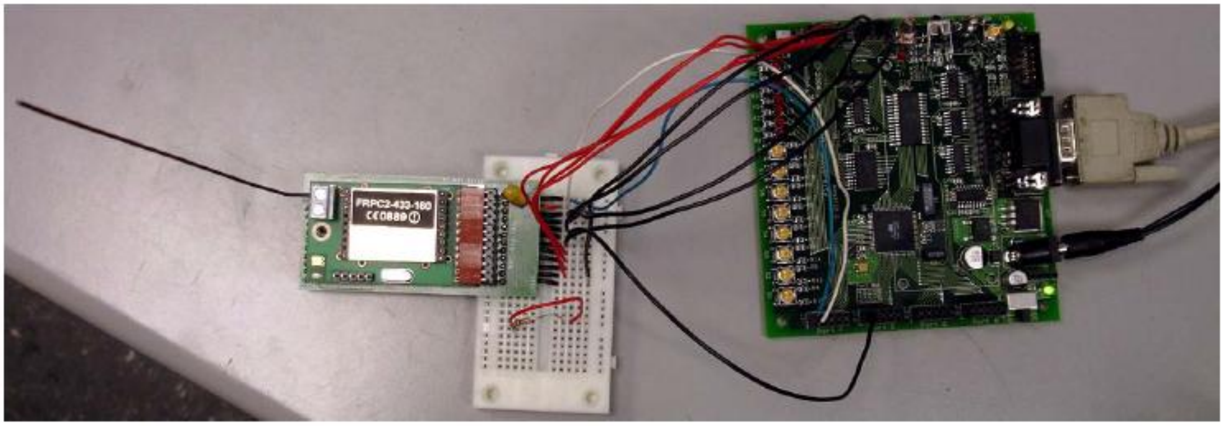

Fig.2.Base station

Latency in processing is defined to be the time taken to process the signal and high transmission rate is defined as the time taken for actual transmission. With regard to Ease of Implementation, the new system should be easy to integrate into the current Robotic unit so that no major redesigning of the boards etc is needed. It should also easy for the team to pick up and learn without needed a lot of expertise since student projects are time-constrained. Its implantation should not delay the development of other parts of the project. With regard to Cost, we need to consider the cost of new modules and new development kits. It has been suggested by past teams that full-duplexity in the communication link is desired. The advantage of having a full-duplex system is that information can be transmitted and received simultaneously from the Robot to the GUI Application through the Base Station, which might be useful in Robots' control. However, through benchmarking other wireless modules shown in Table 3.1, we decided that first we adopt the Fast RPC (FRPC2) whose performance is better than RPC. The advantage choosing the FRPC2 is the low cost, better performance, error-checking available, and ease of implementation. Later on, we plan to upgrade the wireless module to a system even which will be better than FRPC2. The chip with the FRPC2 is the BiM which is more reliable, easy to establish a communication link and it has lower latency. In addition, the BiM is easy to change Frequency without code compilation. In general, the BiM outperforms high-throughput. Other consideration to choose the wireless module is that it should be individually addressable, and have high resistance to interference.

3. SIGNALS

In short, the pin TXR signals a data transfer request from the Microcontroller to the FRPC2. The TXA pin accepts the handshake from the Microcontroller. Similarly, the RXR pin signals for a transfer request at data from the FRPC2 to the Microcontroller. And the RXA pin is used by the Microcontroller to accept the handshake from the FRPC2. D0 to D3 indicates four bit bi-direction data bus which has tri-state between packet transfers, When not in transmit mode, the FRPC2 continuously searches the radio noise for valid preamble. On detection of a preamble, the FRPC2 synchronizes to the in-coming data stream, decodes the data and validates the check sum. The Microcontroller is then signaled that a valid packet is waiting to be unloaded. The format of the packet is entirely of the users' determination except the 1st byte (the Control Byte shown in Fig. 3.6.1) which must specify the packet type (control or data) and the packet size.

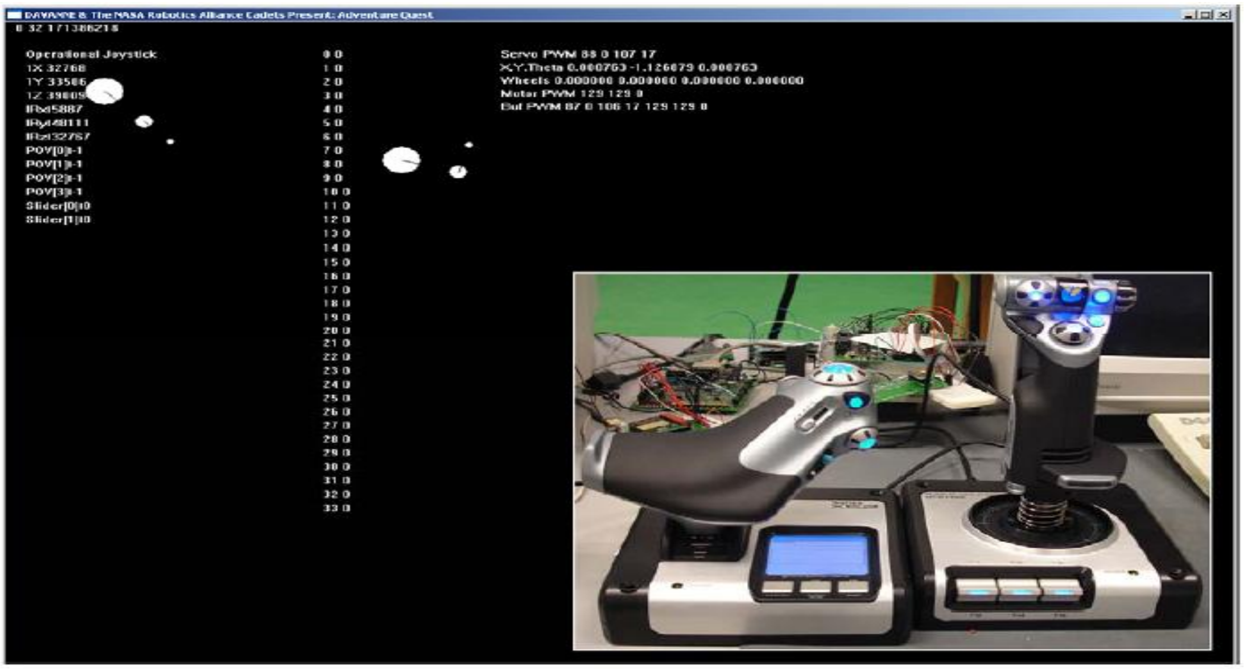

Fig.3.GUI Application

To preserve versatility, the FRPC2 does not generate routing information (i.e. source/ destination addresses) nor does it handshake packets. These network specific functions are left for the Microcontroller to perform. Additional features of the FRPC2 include extensive diagnostic/debug functions for evaluation and debugging of the radio and Microcontroller driver software, a built in self test function and a sleep mode / wake-up mechanism which may be programmed to reduce the average current to less than 100μA. The operating parameters are fully The IDLE state is the quiescent/rest state of the FRPC2. In IDLE the FRPC2 enables the receiver and continuously searches the radio noise for message preamble. If the power saving modes have been enabled, the FRPC2 will pulse the receiver on, check for preamble and go back to SLEEP if nothing is found. The 'ON' time is 2.5ms, OFF time is programmable in the FRPC2's EEPROM and can vary between 22 ms and 181ms. The TX Request line from the Microcontroller is constantly monitored and will be acted upon if found active (low). A TX Request will immediately wake the FRPC2 up from SLEEP mode.

4. ANALYSIS

The proposed protocol defines a simple packet syntax that is independent of the physical transport medium. The protocol is designed to support up to 15 uniquely addressed nodes where a node can be a Robot or a Base Station. Within the network, all nodes are peers, and therefore any node can communicate with any other node. Although the proposed protocol can be used in a number of network topologies, it is optimized for use in a peer, asynchronous, bussed network. Every node in the network must have a unique, nonzero address.

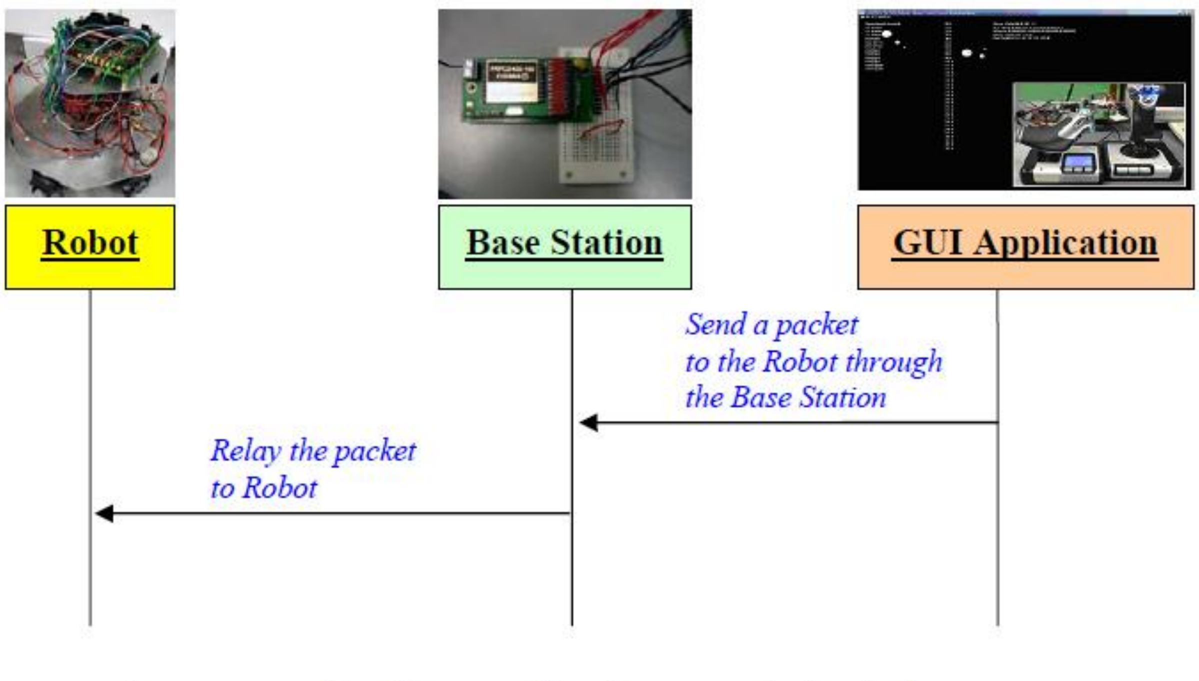

Fig.4.Analysis

This allows every packet to carry both source and destination information. Therefore, by default, any node can send a packet to any other, n order to be as flexible as possible, the packet is byte aligned, with 8bit bytes. This makes implementation easy and efficient on a number of platforms, including the 8bit AVR microcontroller used as the main processing unit on the Robots. Byte alignment also makes the packet easily compatible with a number of physical transport layers The HEADER section contains all important information about a packet. It contains four nibbles split between two bytes that collectively define the packet. The first byte contains source and destination address information (SRC, DST). The source address must always be the unique address of the originating node. Note that address 0 is reserved for future use and thus cannot be used as an originating address. The destination address can either be the address of the intended recipient or 0 for a general call. Every node is required to accept a general call broadcasting of a message from one node automatically.

CONCLUSION

In this project, we achieved control both wireless communication between the mobile Robot and the remote Base Station, and serial communication between the remote Base Station and the GUI Application. The main task of this project was two parts: to program the AVR microcontroller on both the Base Station and the Robot interfaced to the radio packet controller module which would enable us to wirelessly control the Robot and to program the GUI Application which would enable us to serially control the Base Station. This level of completely was successfully tested on groups at up to four Robots. Hence the wireless communication and the serial communication were successful in the downlink.

REFERENCES

[1] Definition of a communication protocol
http://www.webopedia.com/TERM/C/communications_protocol.html
[2] Rover Revolution™ App-Controlled Wireless Spy Vehicle
http://www.brookstone.com/rover-revolution-wireless-spy-vehicle
[3] Parallax RN-42 Bluetooth to PC demo
http://learn.parallax.com/project/rn-42-bluetooth-pc-demo
[4] MicroTronics Technologies Mobile Controlled Robot
http://www.projectsof8051.com/mobile-controlled-robot/
[5] Mobile Operating System World Wide Market Share – International Business Times
http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892
[6] Android platform names, API levels, and Code names
http://developer.android.com/guide/topics/manifest/uses-sdk-element.html
[7] Bluetooth Adapter Class
http://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html