Proceedings of the International Multiconference on Computer Science and Information Technology pp. 869–872
ISBN 978-83-60810-27-9
ISSN 1896-7094

Wireless Transceiver for Control of Mobile Embedded Devices

Jan Kordas, Petr Wagner, Jiri Kotzian
Department of Measurement and Control, Faculty of Electrical Engineering and Computer Science
VŠB – Technical University of Ostrava
17. listopadu 15, Ostrava-Poruba 708 33, Czech Republic
{jan.kordas, petr.wagner, jiri.kotzian}@vsb.cz

***Abstract*—This article deals with the control of mobile embedded devices via wireless transceiver. The only way how to control the mobile devices such as robots, is to use a wireless data transfer. The possible wireless transceiver solution using the nRF24L01 transceiver by Nordic Semiconductor, which works in license-free worldwide 2.4 GHz ISM frequency band, is presented. Overview of this chip is included at the beginning of this article. The main part of this article deals with the design of a communication protocol. Then some optimizations of this protocol to improve its performance and determinism are discussed. In the last part, the results of measurement of some data transfer characteristics are presented.**

I. TIntroduction

HE wireless data transfer has been a very popular topic in recent years. It replaces the wire or optical connection in some applications. However, this article is focused on applications where a wire is impossible to use. The wireless connection is the only possibility in applications such as the control of the mobile embedded devices [1], e. g. robots. We can distinguish the two basic types of communication. In the first configuration of the wireless network, one stationary device is controlling one or more mobile devices. In the second configuration, all the devices are mobile.

This article discusses possible approaches of the control of the mobile embedded devices by the one stationary control system. This is intended to be used in the real-time application where a group of five robots is controlled by the stationary control system.

II. Design of Wireless Transceiver

The designed wireless transceiver is based on the nRF24L01 transceiver [3] which ensures the data transmission over the air. The nRF24L01 provides SPI (Serial Peripheral Interface) for connection with a target device. SPI is appropriate for the connection of the nRF24L01 with a microcontroller of the mobile device which has this interface also (this is the case of the mobile robots). However, it is not the suitable interface for the connection of control computer. So that SPI has to be converted to another more suitable interface for the connection of the nRF24L01 with a personal computer. USB (Universal Serial Bus) was chosen because each modern personal computer has available at least a few unused USB ports. The conversion from SPI to USB is done in two steps. At first, SPI is converted to UART. The conversion is made by the Freescale MC9S12D64 microcontroller which controls the nRF24L01 also. Any other microcontroller with the same or higher performance can be used. Then UART is converted to USB by the FTDI FT232RL converter. The block diagram of the wireless transceiver is shown in Fig. 1.

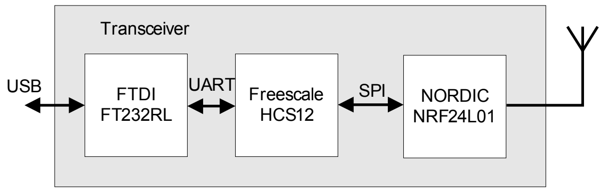

Fig. 1 Block diagram of wireless transceiver

I. Nordic nRF24L01

The nRF24L01 is a single chip 2.4GHz transceiver with an embedded baseband protocol engine (version with integrated MCU based on i8051 is also available), designed for ultra low power wireless applications. The nRF24L01 is designed for operation in the worldwide 2.4 GHz ISM frequency band. The air data rate is configurable to either 1 Mbps or 2 Mbps. The radio front end uses GFSK modulation. The chip has user configurable parameters like frequency channel, output power and air data rate.

1) Enhanced Shock-Burst™

The nRF24L01 provides Enhanced Shock-Burst™ mode beside manual operation mode. This mode allows 1 to 32 bytes dynamic payload length, automatic packet handling, auto packet transaction handling, data pipe MultiCeiver for 1:6 star networks.

2) MultiCeiver™

MultiCeiver™ is a feature used in receive mode that contains a set of 6 parallel data pipes with unique addresses. A data pipe is a logical channel in the physical RF channel. The nRF24L01 configured as primary receiver can receive data addressed to six different data pipes in one frequency channel. Each data pipe has its own unique address and can be configured for individual behavior. All data pipe address-

This work was supported by the GAČR project 102/08/1429 "Safety and security of networked embedded system applications" and by the internal project SP/2010165 "Development and using new methods for design and realization of embedded systems"

es are searched for simultaneously. The length of the messages can vary among data pipes [3].

II. Freescale MC9S12D64

The MC9S12D64 microcontroller [4] is a 16-bit device composed of standard on-chip peripherals including a 16-bit central processing unit, 64 Kbytes of Flash EEPROM for user program, 4 Kbytes of RAM for application data, 1 Kbyte of EEPROM as persistent data storage, two UARTs, one SPI, 29 discrete digital I/O channels, and so on. The inclusion of a PLL circuit allows power consumption and performance to be adjusted to suit operational requirements.

III. FTDI FT232RL

The FT232R is a single chip USB to asynchronous serial data transfer interface. UART interface supports 7 or 8 data bits, 1 or 2 stop bits and odd, even, mark, space, or no parity. Data transfer rates from 300 baud up to 3 Mbaud at TTL levels are supported. The entire USB protocol is handled on the chip, so that no USB specific firmware programming is required. The FT232RL supports bus powered, self powered, and high power bus powered USB configurations. For complete specification of the FTDI FT232RL please refer to [5].

Fig. 2 Realized transceiver

III. Communication Protocol

For systems where the real-time requirement should be met a deterministic access method should be used for channels which are not dedicated for the only two communicating devices [2]. The master-slave access method is appropriate for the robot control application because the prevailing data transfer direction is from the stationary device to the mobile devices. So the stationary device is the master and the mobile devices are the slaves.

We can distinguish the three types of messages which are sent from the personal computer to the wireless transceiver and then to the remote device:

1. Data for the mobile device.
2. Request for the mobile device status data.
3. Service messages to the wireless transceiver (these data are not sent over the air).

The data for mobile device are sent from the computer via the USB serial port to the microcontroller. This data are encapsulated by some extra information such as the type of the message, its length, CRC, etc. The structure of the message is shown in Fig. 3.

Type 1B	Length 1B	Address 3B	Data 1 to 32B	CRC8 1B

Fig. 3 Message structure

The data that are received and successfully checked by the microcontroller are unpacked (the header and footer are removed) and then they are sent directly to the nRF24L01. The nRF24L01 adds the destination address and CRC to the message and after that it sends this message over the air. A device that receives a correct message with its address sends the received message to the connected microcontroller on its request. The nRF24L01 is configured to send data with the 3 bytes address (the shortest possible option) and to secure the data by CRC16.

I. Protocol optimization

In the robot control application, two data pipes with different addresses can be used to distinguish the unicast and broadcast messages. The first data pipe is used for the unicast messages and the second data pipe is used for the broadcast messages. The unicast address is unique for each device in the net. The broadcast address is the same for all devices in the net.

The broadcast messages can be used only for the control messages where a response is not expected. The use of the broadcast messages instead of the unicast messages eliminates the delivery time delay and improves the determinism of the transmission time. The determinism is improved because all robots receive the message in the exactly same time. The improvement was tested and the results are presented in the next paragraph of this article.

The particular structure of the broadcast message is in Fig. 5. This message is inserted to the data part of the message shown in Fig. 3. The broadcast message contains data for up to 6 mobile robots. The data for each robot are identified by its address. The left and right wheel revolutions are the typical data for the robots. The command for each robot is the same hence it appears only once in the message. The counter is increased for the every new message. For better reliability of transmission, each message can be sent twice or more times without increasing of this counter so the remote device can recognize if the message has been sent before.

IV. Testing of wireless transceivers

Two wireless transceivers developed according to the previous design were tested. Two similar transceivers with the same configuration of the wireless part and the same antenna with gain 2 dBi were used for testing. The first wireless

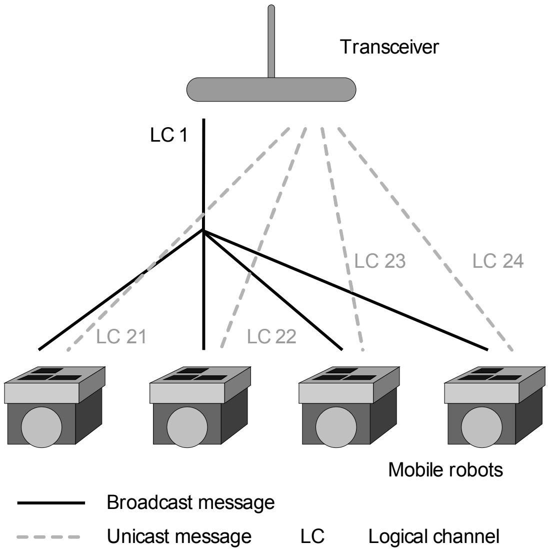

Fig. 4 Unicast and broadcast configuration

<table>
<tr><td rowspan="2">Counter
1B</td><td rowspan="2">Command
1B</td><td colspan="2">1 to 6 times</td></tr>
<tr><td>Address
1B</td><td>Data
4B</td></tr>
</table>

Fig. 5 Structure of broadcast message

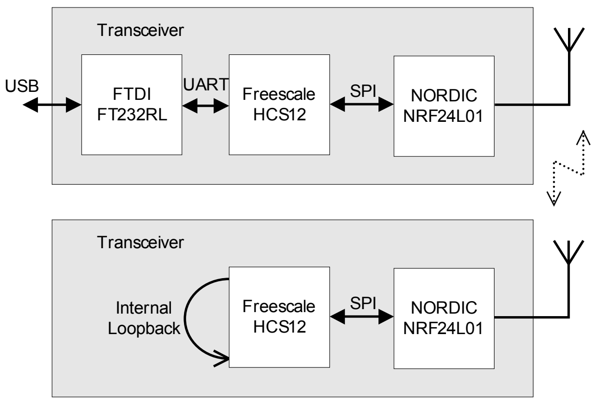

Fig. 6 Configuration of the experiment

transceiver had implemented the internal loopback on wireless interface while the second sent the data received from the wireless interface to USB and vice versa. The test configuration is shown in Fig. 6.

I. Delivery time

The first of all, the message delivery times via wireless interface were tested. For this measurement, the oscilloscope was used because very short delivery times were expected. The microcontroller, which is on wireless transceiver, sets (for 3 milliseconds) one of its outputs on the data sent event and the other output on the data received event. These events were measured by oscilloscope. The measurement was performed for various packet lengths and for the both air data rates supported by the nRF24L01. The results are in Table 1.

TABLE I.
COMPARISON OF 1 MBPS AND 2 MBPS

number of bytes	**1 Mbps**	**2 Mbps**
5	224 µs	202 µs
10	289 µs	220 µs
15	325 µs	239 µs
20	367 µs	259 µs
25	406 µs	277 µs
30	448 µs	297 µs

Notice that the delivery times for 2 Mbps are not a half of the delivery times for 1 Mbps. This is caused by the constant time that is necessary to set up the nRF24L01 to the transmit mode. This setup takes 150 microseconds according to the performed test. It is possible to send more than one message after setting the nRF24L01 to the transmit mode but the nRF24L01 must not be in the transmit mode for more than 4 milliseconds [3].

One byte at 1 Mbps is transmitted in 8 microseconds. At 2 Mbps, this time is a half. According to the test, there is 7 bytes overhead on the wireless interface. This overhead includes 3 bytes of address, 2 bytes of CRC, and 2 extra bytes (probably preamble).

1) Unicast vs. broadcast messages delivery time

From the results in Table 1, we can calculate the difference in the delivery time DT between the unicast and broadcast messages. For this calculation, we use equation (1), where the setup_time is 150 us, the overhead_length is 7 and the byte_time is 8 us for 1 Mbps and 4 us for 2 Mbps.

When we use the unicast messages we have to send five messages with 5 bytes payload. This takes 630 us at 1 Mbps and 390 us at 2 Mbps. When we use the broadcast messages we have to send one message with 32 bytes. This takes 462 us at 1 Mbps and 306 us at 2 Mbps.

$$DT = setup_{time} + msgs_{cnt} \cdot ¿ ¿¿(overhead_{length} + paylod_{length}) \cdot byte_{time} \quad (1)$$

The calculated unicast delivery times are only theoretical because all messages must be sent together after one setup of the nRF24L01 to the transmit mode. In reality, each message is send separately and this takes 1.2 ms at 1 Mbps and 1.0 ms at 2 Mbps.

The theoretical difference between the unicast messages and the broadcast messages is not significant but the practical difference is considerable. The real delivery time is three times greater for the unicast messages than for the broadcast messages.

II. Lost messages

The lost of the messages was tested by sending one message per second from the PC during 10 minutes to the wireless transceiver that sent this message to the second wireless

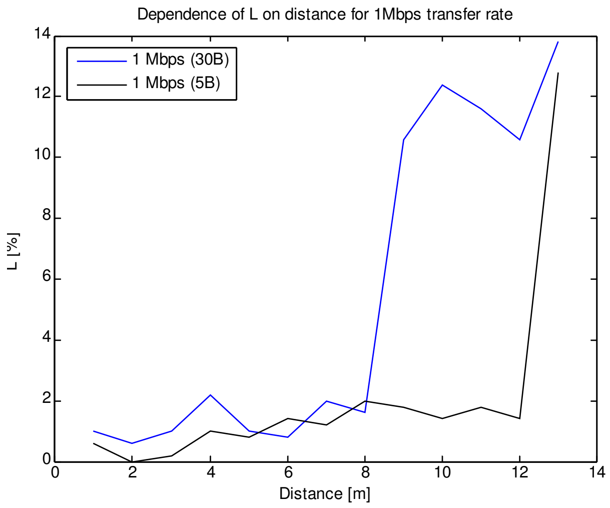

Fig. 7 Message lost ratio (1 Mbps)

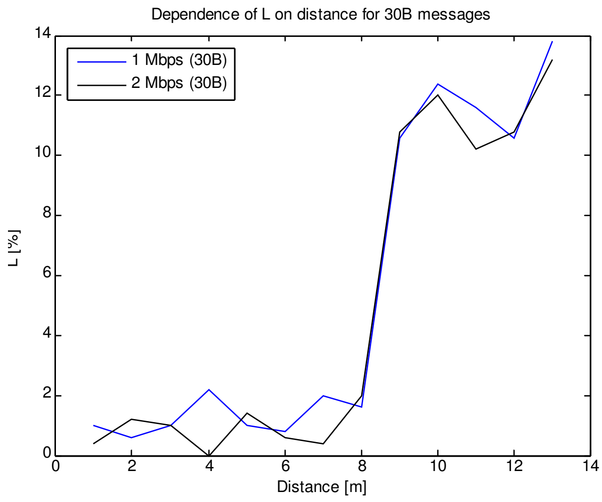

Fig. 8 Message lost ratio (30B messages)

transceiver with internal loopback. The second wireless transceiver only sent the received message back so sender should receive the sent message back. The message lost ratio L is given by equation (2).

$$L = 1 - \frac{received_{messages}}{sent_{messages}} \quad (2)$$

The lost of the messages was tested for both air data rates supported by the nRF24L01, for two lengths of message, and for various wireless transceiver distances. The dependence of the lost messages ratio on distance for 1 Mbps air data rate and for 5 and 30 bytes messages is shown in Fig. 7. And the dependence on distance for 30B messages for 1 and 2 Mbps air data rate is shown in Fig. 8. We can see that the dependence of L on the distance becomes significant for a distance over 8 meters for 30 bytes messages and for a distance over 12 meters of 5 bytes messages, respectively. The air data rate does not significantly affect the lost ratio.

V. Conclusion

The described wireless transceivers were constructed from components and then they were tested. According to the tests, the new design of the wireless system for the control of mobile embedded devices is suitable. Using the communication protocol optimization, the significant improvement has been done. The same information can be sent (using the broadcast messages and air data rate 2 Mbps) in one third time in comparison to the previous solution (using the unicast messages and air data rate 1 Mbps). Each of the controlled robots receives its piece of information in the same time as its team-mates. Furthermore, the use of the broadcast messages allows sending the same message more times so the information delivery probability can be increased. All of these optimizations help to control the mobile embedded devices more accurately which was the objective of our work.

References

[1] V. Srovnal Jr., Z. Machacek, V. Srovnal, "Wireless Communication for Mobile Robotics and Industrial Embedded Devices", in *Proc. Eighth International Conference on Networks*, pp. 253-258, 2009.

[2] S. Sliva, V. Srovnal, "Distributed processing applicable to embedded systems", in *Proc. 7th International Conference on Automatic Control, Modeling and Simulation*, pp. 75-58, 2005.

[3] *nRF24L01 Product Specification*, Nordic Semiconductor, rev. 2.0, 2007.

[4] *MC9S12DJ64 Device User Guide*, Freescale Semiconductor, rev. 1.20, 2005.

[5] *FT232R USB UART IC Datasheet*, Future Technology Devices International, rev. 2.02, 2009.