We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



8

A Reusable UART IP Design and its Application in Mobile Robots

Ching-Chang Wong & Yu-Han Lin
Department of Electrical Engineering, Tamkang University, Tamshui, Taipei 25137, Taiwan

1. Introduction

A Universal Asynchronous Receiver and Transmitter (UART) is an integrated circuit which plays an important role in serial communication. The UART is a standard communication component that is provided by most of the available microprocessors. In general, the number of the UART in a microprocessor is limited and not enough for robot applications. In this chapter, a reusable Intellectual Property (IP) of UART design method by using VHISC Hardware Description Language (VHDL) is proposed and realized on a Field Programmable Gate Array (FPGA) chip. FPGA implementation is flexible because it can be easily reconfigured by the end user and reused for different designs. The proposed UART IP is composed of a baud rate generator, a receiver module, and a transmitter module. These modules are reusable and synthesizable. The popular N-8-1 (No parity (N), eight (8) data bits, and one (1) stop bit) data format is implemented in the proposed UART IP, but the parity setting, the number selection of data bits, and stop bits are user-enhancements. In this chapter, the proposed UART IP is applied to be a data detector of Infra-red (IR) ranging system to receive the distance information of objects, a data detector of digital compass to receive the head direction of robot, and a transceiver of a wireless modem to communicate with the other robots and a host computer. These application circuits have been implemented on a FPGA chip (Altera EP20K200EFC484-2X) to illustrate the robot can detect data from the IR ranging system and the digital compass exactly and communicate with the host computer and other robots successfully.

The importance for System-on-Chip (SOC) using a reusable IP is increasing in modern design methodology. The old design method is not suitable to design chip which operates on a required function in a given time. Therefore, using proper IP for requested specifications can reduce the design time and cope with time-to-market (Delforge, 1998). In this chapter, a reusable UART IP is designed for applications in the serial communication. A UART (Harvey, 1999; Michael, 1989) is an integrated circuit, which plays an important role in the serial communication. The UART contains a receiver (serial-to-parallel converter) and a transmitter (parallel-to-serial converter). It handles the conversion between serial and parallel data. Serial communication reduces the distortion of a signal, therefore makes data transfer between two systems separated in great distance possible. In most computer systems, the UART is connected to circuitry that generates signals that comply with the EIA (Electronic Industries Alliance) RS232-C specification. The advantages of UART systems are the simplicity of interconnection wiring and character transmission protocol and formats. Starting with the original IBM Personal Computer, IBM has selected the National Semiconductor INS8250 UART for use in the IBM PC parallel/serial

Open Access Database www.i-techonline.com

Source: Mobile Robots, Moving Intelligence, ISBN: 3-86611-284-X, Edited by Jonas Buchli, pp. 576, ARS/plV, Germany, December 2006

adapter. Subsequent generations of compatible computers from IBM and other vendors continued to use the INS8250 or improved versions of the National Semiconductor UART family. The 16550A and its successors have become the most popular UART design in the PC industry, mainly due to its ability to handle higher transfer rates of data reliably on operating systems with sluggish interrupt response times.

Recently, a reconfigurable computing system that uses the reconfigurable aspects of the FPGA to implement an algorithm has been attractive because it offers a compromise between special-purpose ASIC hardware and general-purpose processors. The FPGA system is flexible because it can be easily reconfigured by the end user and reused for many different designs. It also provides a rapid prototyping by synthesizing the desired system with an appropriate Electronic Design Automation (EDA) tool. The FPGA-based system is also useful because it can reduce development time greatly. In designing hardware, many design processes entail modelling hardware in varied levels of detail. Designs are described in a hardware description language like VHDL (Navabi, 1998; Roth, 1998) and are verified by simulation. Due to these useful features, we chose the reconfigurable FPGA system as the implementation platform of the mobile robot. The reusable UART IP designed in this chapter is composed of a baud rate generator, a receiver module, and a transmitter module. These modules are applied to the application of mobile robots, including the distance information detector of the IR ranging system, heading output detector of the digital compass module, and communications among soccer robots and a host computer.

This chapter is organized as follows: The concept of the UART and simulation of its individual modules are presented in Section 2. The applications of the reusable IP in autonomous robot soccer system are described in Section 3. The conclusions are given in Section 4.

2. UART IP Design

A UART frame consists of 1 start bit, a number of data bits, an optional parity bit and 1, 1.5, or 2 stop bits. The start bit goes low for one bit time, then a number of data bits are transmitted, least significant bit first, the number of data bits ranges is typically 5, 6, 7, or 8.

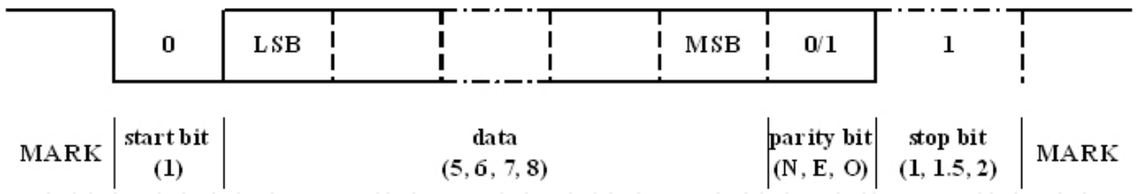


Fig. 1. Standard serial data format.

When no data is being transmitted, a logic 1 must be placed in the transmitted data line. Fig. 1 shows the standard format for serial data transmission. The number of data bits, the parity bit, and the number of stop bits must be set a priori in all communication partners. The design is minimalist, and no error checking logic is present by default. All of these features are to become user-enhancements. A frame format of 1 start bit, 8 data bits, 0 parity bit, and 1 stop bit is supported by all examined UART. In this chapter, the N-8-1 data format is implemented in the proposed UART IP, but the parity setting, the number selection of data bits, and stop bits are user-enhancements. Many UART perform multiple sample points to detect a bit cell and decide on a majority vote. This method affords a multiple of the sampling frequency for single bit detection but provides immunity to shorts spike disturbances on the communication line. The

UART IP is composed of a Baud Rate Generator (BRG), a receiver module, and a transmitter module. The design methods of these modules are described as follows.

2.1. Baud Rate Generator (BRG)

The number of bits transmitted per second is frequently referred to as the baud rate. The BRG divides the system clock by a divisor to provide standard RS-232C baud rate clock (bclk) and 8 times the data rate clock (bclkx8) for general purpose system use when using any one of three industry standard baud rate crystals (1.8432MHz, 2.4576MHz, and 3.072MHz). The divisor can be calculated as follows:

$$n = \frac{f_{clk}}{BR_{max} \times C \times 2} \tag{1}$$

where f_{clk} is the system clock frequency. The integer constant C represents the number of samples per bit cell. BR_{max} is the maximum baud rate frequency. We assumed that the system clock is 1.8432MHz and we want baud rates 300, 600, 1200, 2400, 4800, 9600, 19200, and 38400. The number of samples per bit cell is eight. Then the divisor can be calculated as 3 ($1843200/38400 \times 8 \times 2 = 3$). The architecture of the BRG is shown in Fig. 2. The f_{clk} is first divided by n using a counter. This counter output goes to an 8 bits binary counter. The outputs of the flip-flops in this counter correspond to divided by 2, 4, …, and 256. One of these outputs is selected by a multiplexer. The simulation result of the BRG is shown in Fig. 3.

sel[2..0]	bclk	bclkx8
000	38400	307200
001	19200	153600
010	9600	76800
011	4800	38400
100	2400	19200
101	1200	9600
110	600	4800
111	300	2400

Table 1. Frequencies generated table.

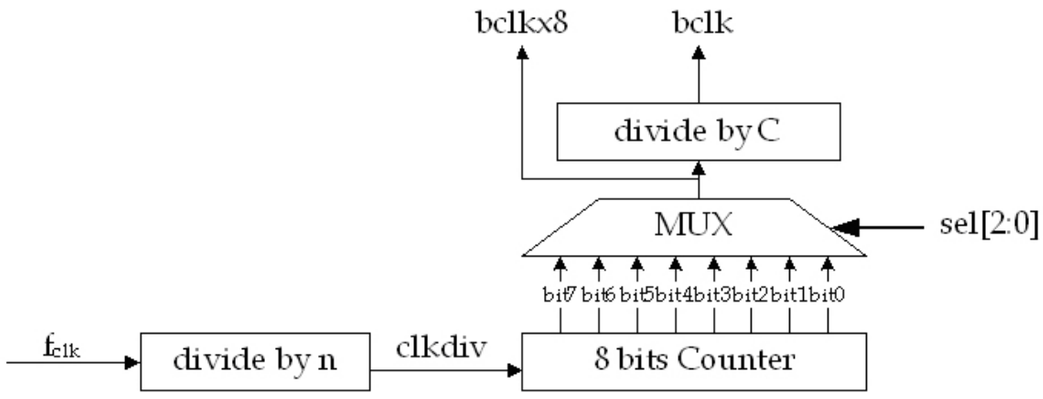


Fig. 2. Architecture of the baud rate generator.

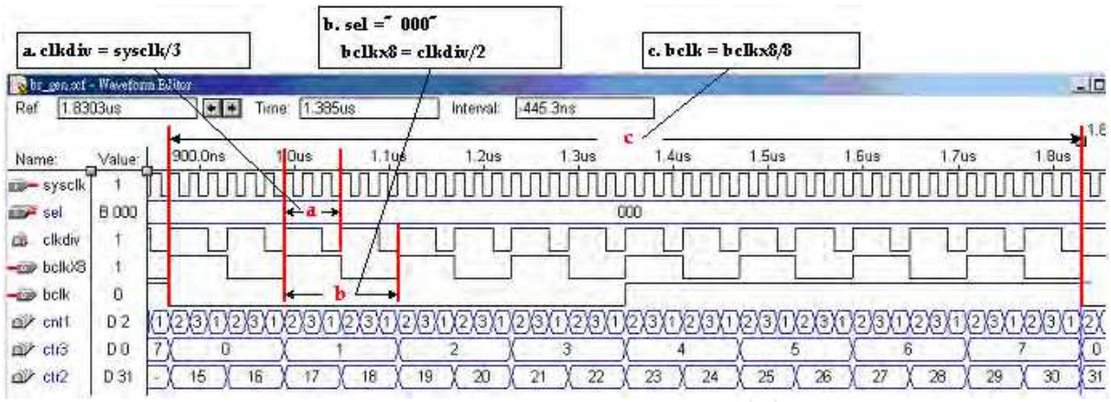


Fig. 3. Simulation result of the baud rate generator.

2.2 Receiver Module

The task of the receiver is to receive a serial bit stream in the form; start bits, data, parity information, and stop bits and store the contained data. To avoid setup and hold time problems and reading some bits at the wrong time, we sampled serial data eight times during each bit time. That is, we sampled on the rising edge of bclkx8. The arrows in Fig. 4 indicate the rising edge of bclkx8. When rxd first goes to 0, we will wait eight more bclkx8 periods, and we should be near the middle of the start bit. Then we will wait eight more bclkx8 periods, which should take us near the middle of the first data bit. We continue reading once every eight bclkx8 clocks until we have read the stop bit.

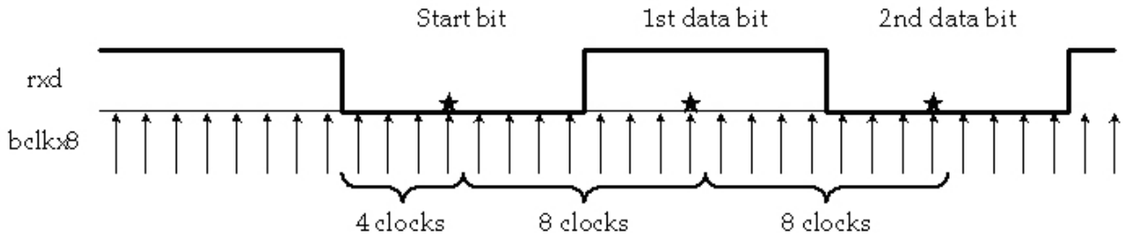


Fig. 4. Data sampling points by the UART receiver.

The receiver of the UART is composed of a state machine, a de-serializer, and a support logic. The main goal of the receiver is to detect the start-bit, then de-serialize the following bit-stream, detect the stop-bit, and make the data available to the host. Fig. 5 illustrates the functional block diagram of the receiver. The design is minimalist, and no error checking logic is presented by default. All of these features are to become user-enhancements.

The data shift register is a simple serial-to-parallel shift register. It has one control input shiftRSR from the state machine. When this signal is active high, the data shift register shifts the data over by 1 bit. The received data loader is used to update the parallel data RDR by the load_RDR signal from the state machine. The received bit counter is used to keep track of the number of data bits cumulated so far. This counter has 2 control inputs: inc1 and clr1. When the former is active high, the counter is advanced by 1. When the latter is active high, the counter is cleared to 0. The bit cell counter is used to generate a delay in units of bclkx8. This is an up counter controlled by inc2 and clr2. The control method is the same with the received bit counter. When the receive operation is completed and no errors have occurred, the rxd_readyH will be a high signal for a system clock period.

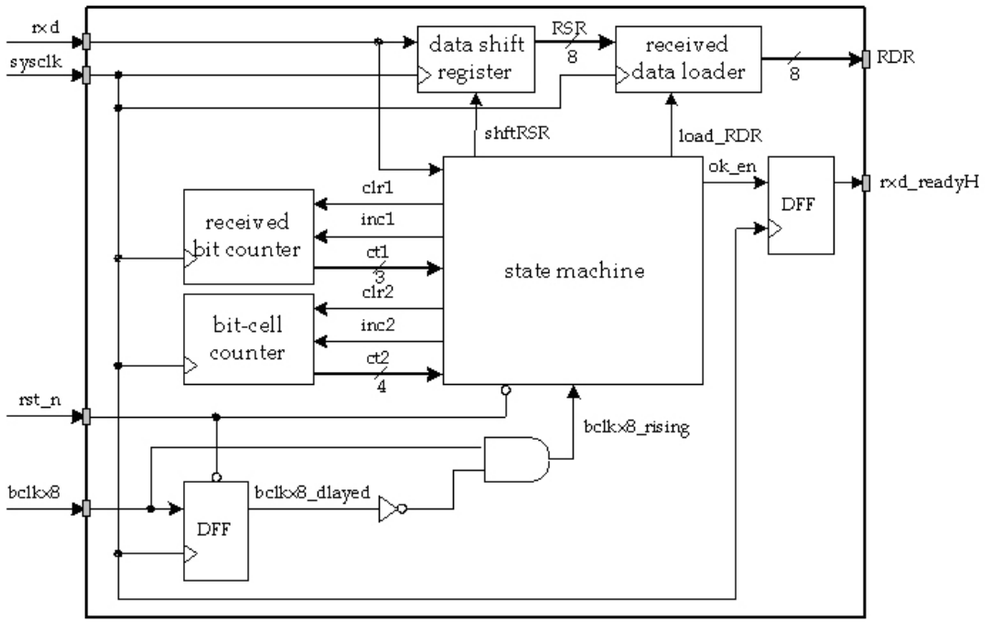


Fig. 5. Functional block-diagram of the UART receiver.

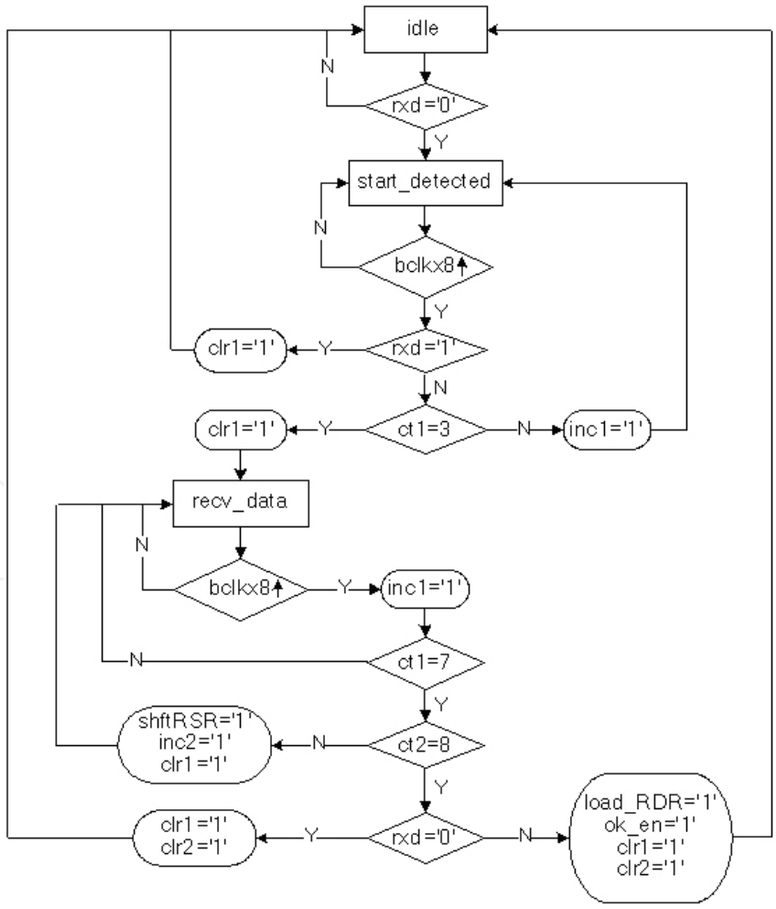


Fig. 6. SM chart for UART receiver.

Fig. 6 illustrates the SM chart of the UART receiver. The state machine is the Mealy machine and composed of three states (idle, start_detected, and recv_data). Two counters are used. ct1 counts the number of bclkx8 clocks. ct2 counts the number of bits received after the start bit. In the idle state, the SM waits for the start bit and then goes to the stsrt_detected state. The SM waits for the rising edge of bclkx8 and then samples rxd again. Since the start bit should be '0' for eight bclkx8 clocks, we should read '0'. ct1 is still 0, so ct1 is incremented and the SM waits for bclkx8↑. If rxd='1', this is an error condition and the SM clears ct1 and resets to the idle state. Otherwise, the SM keeps looping. When rxd is '0' for the fourth time, ct1 = 3, so ct1 is cleared and the state goes to recv_data state. In this state, the SM increments ct1 after every rising edge of bclkx8. After the eighth clock, ct1=7 and ct2 is checked. If it is not 8, the current value of rxd is shifted in to RSR, ct2 is incremented, and ct1 is cleared. If ct2=8, all 8 bits have been read and we should be in the middle of the stop bit. If rxd ='0', the stop bit has not been detected properly, the SM clear ct1 and ct2 and resets to the idle state. If no errors have occurred, RDR is loaded from RSR and two counters are cleared and ok_en is set to indicate that the receive operation is completed. The simulation result of the UART receiver receiving 0x53 is shown in Fig. 7.

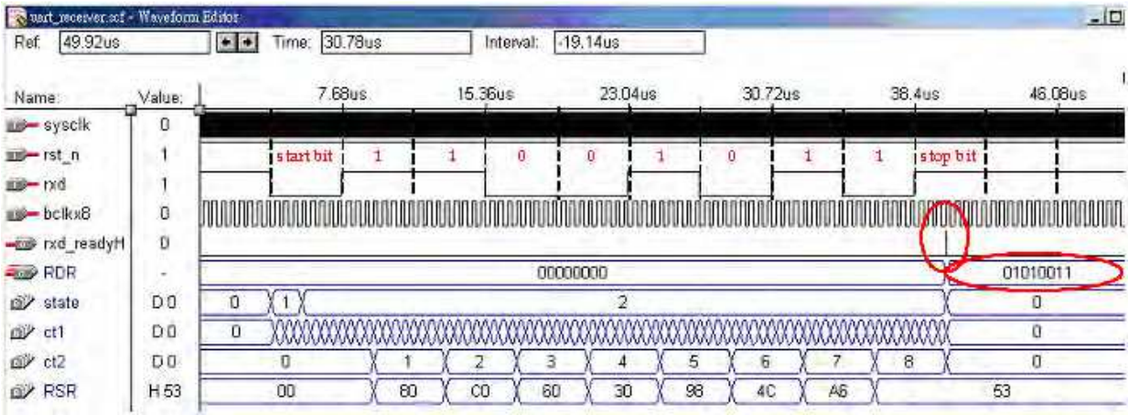


Fig. 7. UART receiver receiving 0x53.

2.3 Transmitter Module

The transmitter circuitry converts a parallel data word into serial form and appends the start, parity, and stop bits. The transmitter of UART is composed of transmitted bit counter, a data shift register, a state machine and support logic. Fig. 8 illustrates the functional block diagram of the UART transmitter. The transmitted bit counter has the same function and implementation as that of the receiver, only the signal name have changed slightly.

The data shift register is an 8 bits parallel-in-serial-out shift register. It has 3 control inputs: loadTSR, start and shiftTSR. An active high on the first signal loads the parallel data into the shift register. An active high on the second signal transmits a start-bit (logic 0) for one bit time. An active high on the last signal shifts the loaded data out by 1 bit. When the transmit operation is completed, the txd_doneH will be a high signal for a system clock period.

The data shift register is an 8 bits parallel-in-serial-out shift register. It has 3 control inputs: loadTSR, start and shiftTSR. An active high on the first signal loads the parallel data into the shift register. An active high on the second signal transmits a start-bit (logic 0) for one bit time. An active high on the last signal shifts the loaded data out by 1 bit. When the transmit operation is completed, the txd_doneH will be a high signal for a system clock period.

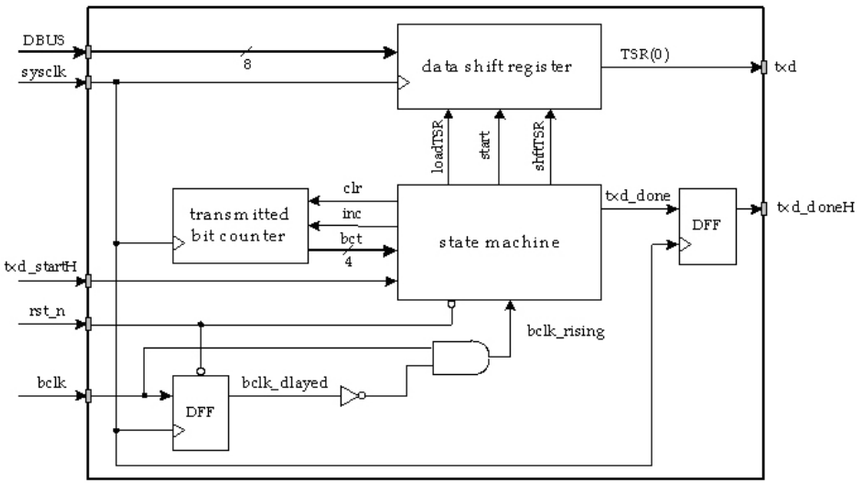


Fig. 8. Functional block-diagram of the UART transmitter.

Fig. 9 illustrates the SM chart of the UART transmitter. The state machine is the Mealy machine and composed of three states (idle, synch, and tdata). In the idle state, the SM waits until txd_startH has been set and then loads DBUS data into higher eight bits of TSR register. The TSR register is a nine bits data register and the low-order bit is initials to '1'. In the synch state, the SM waits for the rising edge of the bit clock (bclk↑) and then clears the low-order bit of TSR to transmit a '0' for one bit time. In the tdata state, each time bclk↑ is detected, TSR is shifted right to transmit the next data bit and the bit counter (bct) is incremented. When bct=9, 8 data bits and a stop bit have transmitted, bct is then cleared and tx_done is set to indicate that the transmit operation is completed. The simulation result of the UART transmitter transmitting 0xa5 is shown in Fig. 10.

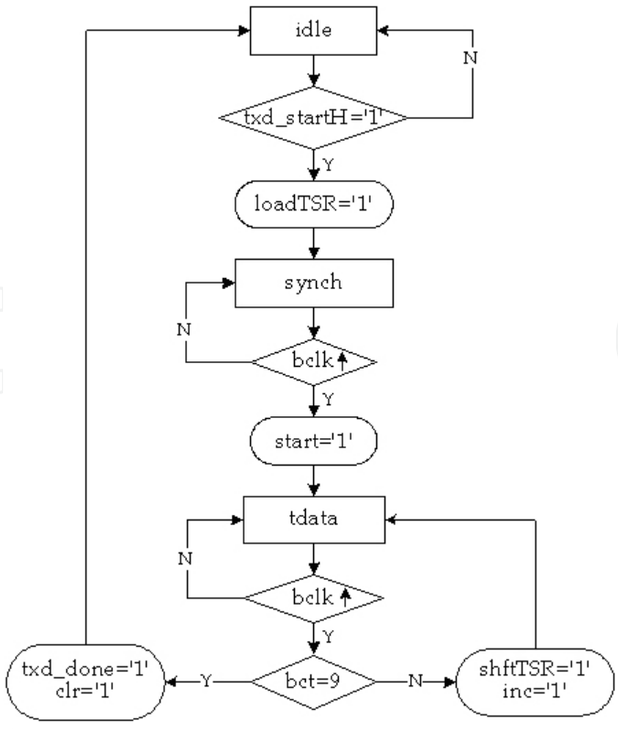


Fig. 9. SM chart for UART transmitter.

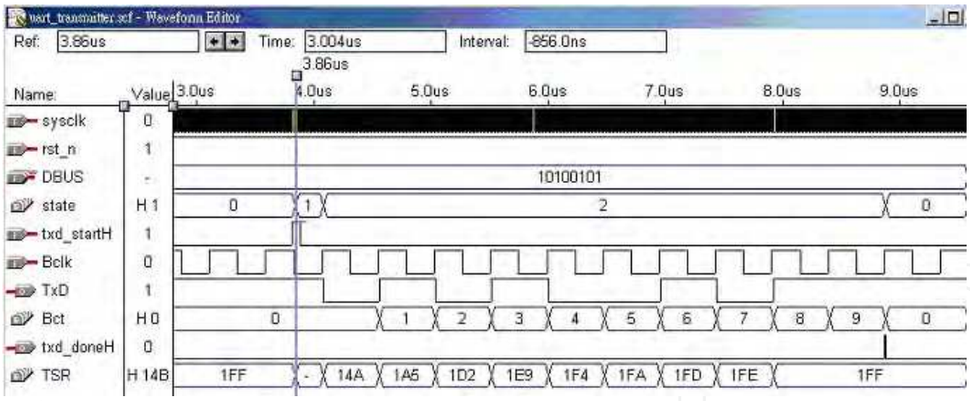


Fig. 10. UART transmitter transmitting 0xa5.

3. Application to Autonomous Soccer Robots

The reusable UART IP is used to detect the information of the IR ranging system and digital compass module of autonomous soccer robots. The communication between robots are using RF module.

3.1 IR Ranging System Data Detector

The IR ranging system of the autonomous robot soccer system we adopted is DIRRS+, which is manufactured by HVW Technologies Inc. (HVW Technologies Inc., 1998), and has a range of 10 to 80 cm. The actual photo of digital infra-red ranging system is shown in Fig. 11. The distance is output by the sensor on a single pin as a digital 8-bit serial stream. The specification of the serial stream is RS-232 8-N-1 at 4800 bps. A close object reports a larger value, and a distant object reports a smaller value. The BRG and receiver module of the UART IP can be used to receive the distance information of the IR ranging system.

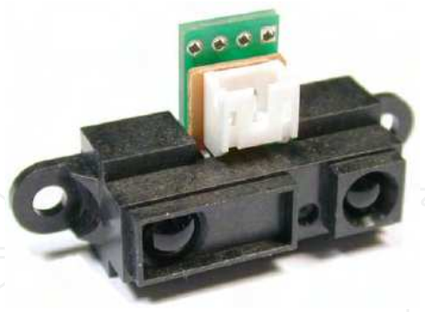


Fig. 11. Digital infra-red ranging system.

3.2 Digital Compass Data Detector

The digital compass module of the autonomous robot soccer system we adopted is TDCM3, which is manufactured by Topteam Technology Corp. (Topteam Technology Corp., 2001). The digital compass outputs compass heading via electronic interface to a host system. It outputs compass readings by a host request. The interface is RS-232 8-N-1 at 4 levels speed to transmit data, these are 2400, 4800, 9600, 19200 bps. When the host sends a pulse via RTS pin to the device it will output heading via TX pin to the host. Before the host sends a pulse to the device the RX pin needs to be kept in high level. The normal mode timing diagram is shown in Fig. 12. The data

format is Status, θ_{MSB} and θ_{LSB}. The "status byte" is a flag that shows the TDCM3 status. In normal state the "status byte" is equal to 80H when distortion is detected the "status byte" is equal to 81H. The compass heading can be calculated as follows:

$$\theta = (\theta_{MSB} \times 256 + \theta_{LSB})/2 \tag{2}$$

The BRG and receiver module can be used to receive the heading information of the digital compass.

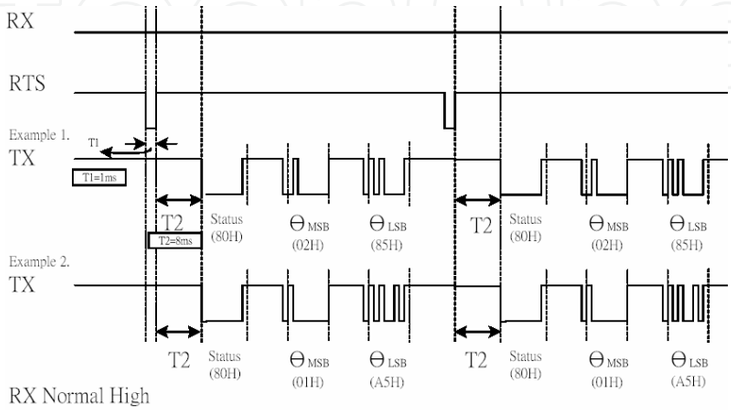


Fig. 12. Timing chart of normal mode.

3.3 Communication of robots

Autonomous robot soccer system consists of a multi-agent system that needs to cooperate in a competitive environment. Thus, it is a crucial issue how to realize effective and real time communications among soccer robots. We use the SST-2400 radio modem which is developed by IPC DAS Corp. (ICP DAS Corp., 1999) to construct a wireless LAN. The actual photo of SST-2400 wireless radio modem is shown in Fig. 13. The SST-2400 is a spread spectrum radio modem with RS-232 interface port. Based on direct sequence spread spectrum and RF technology operating in ISM bands. The Frequency Range is 2400MHz~2483.5MHz. Considering about experiment requirements for multi-robots communication, we set the wireless modems in the operation mode2 which is point to multi-point, half-duplex, synchronous, the fixed data format 8-N-1, and 9600 baud rate. Three modules of the UART IP can be used to receiving and transmitting data among host computer and other robots.



Fig. 13. SST-2400 wireless radio modem.

4. Conclusions

A reusable UART IP design and its application in mobile robots are presented in this chapter. The UART IP is composed of a baud rate generator, a receiver module, and a transmitter module. These modules are reusable and synthesizable and they are used to be a data detector of IR ranging system to receive the distance information of objects, a data detector of digital compass to receive the head direction of robot, and a transceiver of a wireless modem to communicate with the other robots and host computer. These application circuits have been implemented on a FPGA chip to illustrate the robot can detect data from the IR ranging system and the digital compass exactly and communicate with the host computer and other robots successfully.

5. References

Delforge, P. (1998). IP Re-use Model, IP Market and IP Corporate Database, Proceedings of Intellectual Property System on Chip Conference, pp. 11-16, Mar, 1998.

Harvey, M.S. (1999). Generic UART Manual, Silicon Valley, December, 1999.

Michael, M.S. (1989). A Comparison of the INS8250, NS16450 and NS16550AF Series of UARTs, National Semiconductor Application Note 493, April, 1989.

Navabi, Z. (1998). *VHDL: Analysis and Modeling of Digital Systems*, 2nd Edition, McGraw-Hill, 0070464790, New York.

Roth, Charles H. (1998). *Digital System Design Using VHDL*, PWS, 053495099X, Boston, Mass.

Data Sheet DIRRS+, Alberta, Canada, HVW Technologies Inc., 1998.

Data Sheet TDCM3, Taiwan, Topteam Technology Corp., 2001.

User's Manual: SST-2400 Radio Modem Module, Taiwan, ICP DAS Corp., 1999.

IntechOpen



Mobile Robotics, Moving Intelligence

Edited by Jonas Buchli

ISBN 3-86611-284-X

Hard cover, 586 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers many aspects of the exciting research in mobile robotics. It deals with different aspects of the control problem, especially also under uncertainty and faults. Mechanical design issues are discussed along with new sensor and actuator concepts. Games like soccer are a good example which comprise many of the aforementioned challenges in a single comprehensive and in the same time entertaining framework. Thus, the book comprises contributions dealing with aspects of the Robotcup competition. The reader will get a feel how the problems cover virtually all engineering disciplines ranging from theoretical research to very application specific work. In addition interesting problems for physics and mathematics arises out of such research. We hope this book will be an inspiring source of knowledge and ideas, stimulating further research in this exciting field. The promises and possible benefits of such efforts are manifold, they range from new transportation systems, intelligent cars to flexible assistants in factories and construction sites, over service robot which assist and support us in daily live, all the way to the possibility for efficient help for impaired and advances in prosthetics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ching-Chang Wong and Yu-Han Lin (2006). A Reusable UART IP Design and its Application in Mobile Robots, Mobile Robotics, Moving Intelligence, Jonas Buchli (Ed.), ISBN: 3-86611-284-X, InTech, Available from: http://www.intechopen.com/books/mobile_robotics_moving_intelligence/a_reusable_uart_ip_design_and_its_application_in_mobile_robots

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2006 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen