Department of Computer Science, UTSA

Technical Report: CS-TR-2008-007

RF Communication for LEGO/Handy Board with Tmote *

Dakai Zhu and Ali Tosun

Department of Computer Science

University of Texas at San Antonio

San Antonio, TX, 78249

{dzhu,tosun}@cs.utsa.edu

Abstract

LEGO robots with Handy Board controllers have been widely used in upper-division Computer Science courses, such as Robotics *and* Embedded Systems*. However, due to the limited communication capability available on the Handy Board controller, only a single individual LEGO robot was exploited for each student group to complete a certain mission in most existing courses. In this paper, we describe our experience on enhancing the Handy Board controller with Tmote for* radio-frequency (RF) *communication capability. We discuss the hardware connections and sample codes for communication between Handy Board and Tmote via serial communication ports. With such enhancement, we can build LEGO robots that have wireless communication capability. Future courses that could benefit from such enhanced LEGO robots are also discussed.*

*The work of Dr. Zhu was supported in part by NSF award CNS-0720651. The work of Dr. Tosun was supported in part by NSF award CCF-0702728.

1 Introduction

Computing education is currently going through difficult times. The enrollment in bachelors of computer science programs at the universities nationwide has declined significantly in recent years (about 70% compared to the peak enrollment in 2000), according to the Computing Research Association [12]. However, a greater problem is the growing gap between the number of jobs requiring high-level computing skills and the number of graduates who are prepared to take those positions. Computer science departments throughout the country are investigating innovative teaching techniques and approaches to attract new students to computer science, to retain existing students and to train them to be the qulified workforce [16].

One approach that received significant attention is to use embedded devices (such as LEGO® robots [13]) in computer science education [16]. LEGO robots provide excellent platforms for teaching various subjects of computer science. For instance, LEGO robots have been exploited to teach introductory programming [8], java bytecode [5], graphical user interfaces [7], image processing [6], systems related courses [4] and robotics [11, 10].

As one advanced LEGO robot controller, MIT Handy Board [14] has been widely used [11, 10] due to its simple interface with LEGO components and rich functionalities of connecting other digital/analog I/O devices (see Section 2.1 for more details). In this paper, we consider LEGO robots equipped with Handy Board controllers. However, the *wireless* communication capability of Handy Board is rather limited. With such limitation, most of the existing courses have only projects that involve individual LEGO robots. For more interesting projects that involve collaboration between multiple LEGO robots (such as Robot Cup), flexible wireless communication capability is needed for LEGO robots to communicate with each other (or with the host PC).

Although previous effort has addressed the *radio-frequency (RF)* communication enhancement for Handy Board [17], such enhancement is limited to communication between different Handy Boards and it involves building the RF communication circuits. In this paper, we describe one approach of enhancing Handy Board for RF communication with Tmote [15], which has very simple hardware connections. Moreover, as one popular sensor node, Tmote has been used extensively in wireless sensor networks (WSN), which enables the wide range of applications of the enhanced LEGO robots.

The remainder of the paper is organized as follows. The background knowledge on Handy Board and Tmote is presented in Section 2. Section 3 describes our experience on enhancing Handy Board with Tmote and shows the hardware connection and sample software codes. Future courses that could benefit from such wireless-capable LEGO robots are discussed in Section 4 and Section 5 concludes the paper.

2 Background

Before presenting our experience on enhancing the MIT Handy Board (i.e., the LEGO robot controller) for wireless communication with Tmote sensor node, we first review in this section the features and specifications of LEGO robots, MIT Handy Board and Tmote.

2.1 LEGO Robots and MIT Handy Board

As the excellent teaching platforms that are widely exploited in high school, colleges and universities, LEGO robots are built from electric motors, sensors, LEGO bricks and LEGO Technic pieces (such as gears, axles, beams, and pneumatic parts) [13]. Although the original programmable controller (powered by a Renesas H8/300 microcontroller and known as the RCX brick) for LEGO robots is good enough for simple applications, it has only three sensor input ports and three motor output ports, which limit its usage in complex projects involving mulple different sensors and motors. In what follows, we will focus on discussing one advanced LEGO controller, the MIT Handy Board [14].

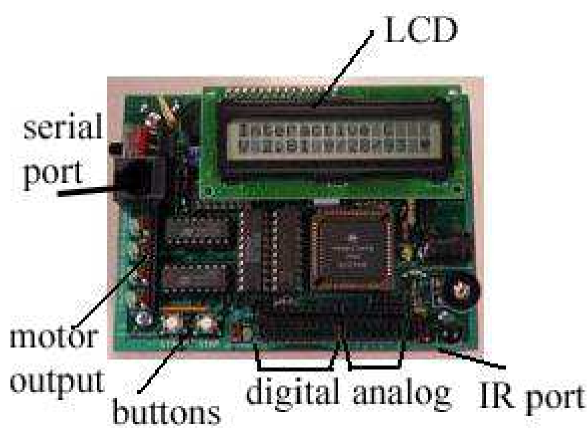


Figure 1. The block diagram of MIT Handy Board

Based on the Motorola MC68HC11 microprocessor, the Handy Board was originally designed by Dr. Fred G. Martin at MIT for experimental mobile robotics projects [14]. The wealthy instruction set of MC68HC11 microprocessors provides a rich set of functionalities for the Handy Board. Moreover, there are 32K battery-backed static RAM and a built-in nicad rechargable battery pack, which can hold and execute the control program of LEGO robots for hours. In addition, as illustrated in Figure 1, Handy Board is featured with an LCD screen, four DC motor outputs, seven analog inputs, nine digital inputs, infrared (IR) input sensor and output circuit and two function buttons. For more complex projects, an expansion board can be connected via the expansion interface, which will provide additional analog inputs, digital outputs and outputs for servo motors.

The ’computer connector’ on Handy Board is actually a RJ11 phone cable connector. Using a phone cable, the Handy Board can communicate with a host PC through a separate serial interface/charger board following standard serial communication protocols. Through this connection, Handy Board can download control programs that are developped on the host machine. In theory, it is possible to connect two Handy Boards using appropriate phone cable and make them communicate with each other. However, for collaborative mobile LEGO robots, wireless/cordless communication is desired.

Exploiting the built-in IR detector sensor and driver interface for IR emitter, it is possible for multiple Handy Boards to interact with each other via IR communication. However, the strict requirements of IR communication (i.e., direct pointing and short distance) limit its application. Moreover, IR communication drivers may consume significant portion of the precious computation power on Handy Boards.

Radio-frequency (RF) communication could be another possible approach for multiple Handy Boards communicating wirelessly. The enhancement for Handy Board with RF communication has been explored previously [17]. In this paper, we explore an alternative solution that exploits Tmote sensor node [15]. In addition to wireless communication capability, as discussed in Section 4, the data sensing ability of Tmote could enable a wide range of interesting applications of the enhanced LEGO robots.

2.2 Tmote Sky Sensors

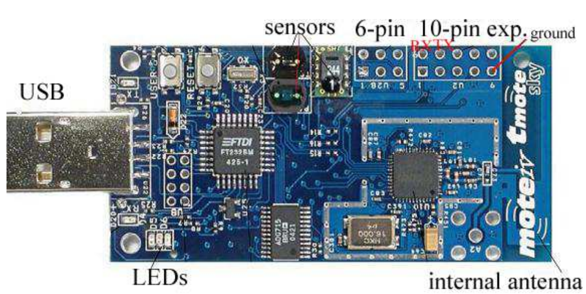


Figure 2. Tmote Sky Sensor

Tmote Sky has been developed as the next-generation platform for extremely low power and high data-rate sensor network applications [15]. Featured with the first available IEEE 802.15.4 radio communication, Tmote Sky boasts an integrated on-board antenna which can provide up to 125 meter communication range. For easy development, Tmote Sky uses the USB protocol to connect with a host machine for programming, debugging and data collection. In addition, Tmote Sky has a number of integrated peripherals (such as 12-bit ADC/DAC, Timer, I2C, SPI, and UART bus protocols). Through the two expansion connectors (i.e., 6-pin and 10-pin connectors as shown in Figure 2), Tmote Sky can connect/communicate with additional devices, such as analog sensors, digital peripherals and GPS receivers.

In this paper, exploiting the UART0 serial communication port available on the 10-pin connector (RX/TX and ground pins as shown in Figure 2), we focus on the communication between Tmote Sky and Handy Board through their serial communication interfaces. Then, messages can be relayed by Tmotes and transmitted between different Handy Boards through wireless communication.

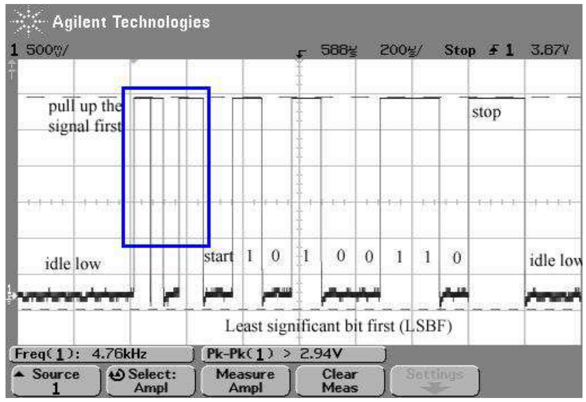
3 Communication Between Tmote and Handy Board

3.1 Serial Communication Signals

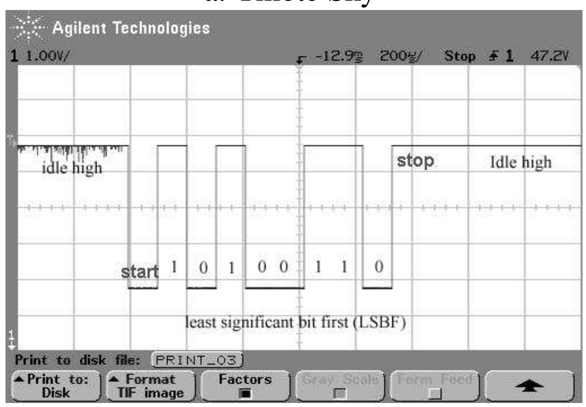
From the specification of Tmote Sky [15], we know that the UART0 serial communication port available on the 10-pin expansion connector shares the bus with radio communication port on Tmote Sky. Therefore, to avoid interference between the serial and radio communication, bus arbitration is needed. A sample program to illustrate the usage of UART0 together with radio communication is available on the webpage for Tmote Sky, where a specific data value is sent out via both UART0 and radio communication periodically [15]. However, after connecting the serial communication ports of Tmote and Handy Board as shown later in Figure 4, we found that Handy Board cannot receive the data sent out by Tmote.

To figure out the exact problem for the communication, we have diagnosed the signals of the serial communication port on both Handy Board and Tmote using an oscilloscope. For such purpose, we program both Handy Board and Tmote send out a data value 0x65 (i.e., the binary value of 01100101) periodically. Here, both Handy Board and Tmote are configured to have 9600 baud rate, eight data bits, no parity bit and one stop bit. Figure 3 shows the annotated signals from the TX pins of Tmote and Handy Board.

From the figure, we can see that both Tmote and Handy Board send out the data following the rules of one start bit, one stop bit and the least significant bit first (LSBF) data sequence. Moreover, the signal on TX pin of Handy Board (Figure 3b) follows the standard for UART closely with



a. Tmote Sky



b. Handy Board

Figure 3. Signals for the TX pins on both Tmote and Handy Board when 0x65 is sent periodically.

high voltage when idle (i.e., idle high). However, the signal on TX pin of Tmote Sky (Figure 3a) has low voltage on the TX pin when idle (i.e., idle low), which is different from the standard for asynchronous serial communication. Therefore, to emulate the standard, Tmote pulls up the voltage level of TX pin right before the communication (as indicated by signals within the rectangle) and then sends out the start bit (with low voltage level) followed by the data bit sequence.

To make the signal of UART0 port on Tmote follow the standard, we need to reset the defaul voltage level of TX pin of UART0 to be idle high. For such purpose, the following codes are needed at the beginning of the program:

```
TOSH_MAKE_UTXD0_OUTPUT();
TOSH_SET_UTXD0_PIN();
```

which set the TX pin as output with idle being at high voltage level. In what follows, we will show the hardware connections between Handy Board and Tmote and then show the sample codes for exchanging data between them.

3.2 Connections and Sample Codes

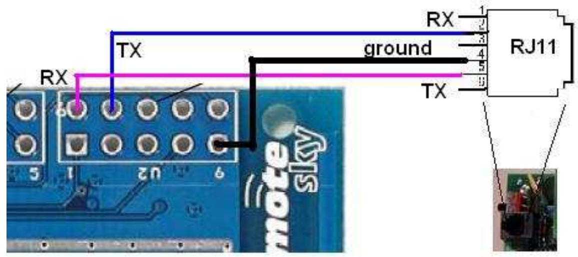


Figure 4. The hardware connection between Handy Board and Tmote

As shown in Figure 4, the connections between Handy Board and Tmote are pretty simple and straightforward. The RX pin of Tmote (pin 2 of the 10-pin connector) needs to be connected to TX pin of RJ11 for Handy Board; similarly, the TX pin of Tmote (pin 4 of the 10-pin connector) needs to be connected to RX pin of RJ11 for Handy Board. The last and important step is to connect the ground pins together (pin 9 of the 10-pin connector on Tmote and pin 4 of RJ11 connector) as illustrated in Figure 4.

```
//hb-test.c
void main(){
  int tx_char=100, rx_char;
  disable_pcode_serial();
  while (!stop_button()) {
    tx_char ++;
    if (tx_char > 110) tx_char = 100;
    serial_putchar(tx);
    rx_char = serial_getchar(rx_char);
    printf("send \%d; receive \%d\n",
           tx_char, rx_char);
    sleep(1.0);
  }
  enable_pcode_serial();
}
```

Figure 5. Sample codes for Handy Board

After proper connections between Handy Board and Tmote, we can program them separately for exchanging data. Using the functions defined in *serialio.c* (which is available in [1]), Figure 5 shows the sample codes for Handy Board. The function *disable_pcode_serial()* is first used to disable the protocol between Handy Board and the host machine for downloading codes. Then, using the functions *serial_putchar()* and *serial_getchar()*, the codes send an integer value (from 100 to 110), receive data and display the values on LCD periodically (for every one second). At the end, the function *enable_pcode_serial()* resets the protocol between Handy Board and the host machine.

```
//TestUART0P.nc
... ...
uint8_t data_in, data_out=0;
uint8_t first_time = 1;
... ...
void SendAndReceiveData(){
  ... ...
  //send out a value
  data_out++;
  if (data_out>10) data_out = 0;
  if (data_out==0) {call Leds.greenOn();}
  if (data_out==6) {call Leds.greenOff();}
  call UartControl.tx( data );
  while (! call UartControl.isTxEmpty() ) ;

  //receive data
  data_in = call UartControl.rx();
  if (data_in==100) {call Leds.yellowOn();}
  if (data_in==106) {call Leds.yellowOff();}
  call Leds.redOff();
}
event void ResourceCmd.granted(uint8_t rh){
  SendAndReceiveData();
}
event void Timer.fired() {
  call Leds.redOn();
  if (first_time ==1){
    first_time =0;
    call ResourceCmd.request(RESOURCE_NONE);
  }else SendAndReceiveData();
}

command result_t StdControl.start() {
  //reset TX pin of UART0 as default high
  TOSH_MAKE_UTXD0_OUTPUT();
  TOSH_SET_UTXD0_PIN();

  call Timer.startPeriodic( 1024);
  return SUCCESS;
}
... ...
```

Figure 6. Sample codes for Tmote

Figure 6 shows the corresponding codes for Tmote, which is modified from the example available on Tmote Sky webpage [15]. The program first resets the default voltage level of TX pin of UART0 to be high (lines 35 and 36). Then it starts a timer (with one second interval). When the timer expires, the event function *Timer.fired()* will be called automatically to periodically send and receive data (line 38).

To illustrate the communication between Handy Board and Tmote, the shared bus resource on Tmote is requested for serial communication when the timer expires the first time (line 29) and is held for UART0 all the time, which will block radio communication but is needed to successfully receive data sent out by Handy Board. More complex bus resource access protocol is needed on Tmote to receive data from UART0 and send data out via radio communication, which is omitted due to space limitation.

Here, we use the red LED to indicate the timer event (line 26). The green LED is used to indicate the data sent out by Tmote: it is turned on when value 0 is sent out (line 11) and remains on until value 6 is sent out (line 12), at that time it is turned off. Note that the output data increases by 1 every time the timer is fired and is reset to 0 when it is more than 10. Thus, the green LED will be on during the period of sending from 0 to 5, and off from 6 to 10. Similarly, the yellow LED is used to indicate the data received by Tmote.

After the connections are set up as shown in Figure 4, we can download the above programs and run them on Handy Board and Tmote, respectively. The output displayed on the LCD of Handy Board and the LEDs on Tmote will confirm that data is exchanged successfully between them.

4 Discussion and Future Courses

4.1 Mobile Sensor Robot

With the successful communication between Handy Board and Tmote Sensor, we can build LEGO robots that have wireless communication capability, which what we called "mobile sensor robots". As shown in Figure 7, the left side shows the connection between the Tmote sensor and the Handy Board, and the right picture shows the working robot. Here, Tmote sensor is placed on the front of the robot and uses its own battery for power. For the cost of a mobile sensor robot, the LEGO set and Handy Board controller cost around $500, and one Tmote sensor costs about $130 including light, humidity and temperature sensors. Moreover, additional equipment used includes a few pins, a cable and costs a few dollars, which leads to the cost of one mobile sensor robot to be around $650.

Such mobile sensor robots can have a wide range of use in computer Science education and

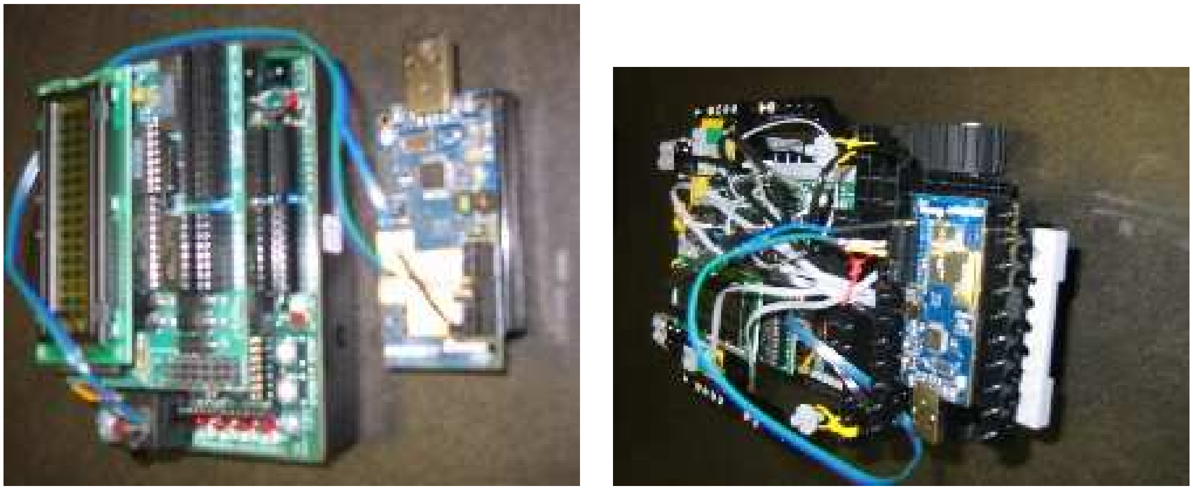


Figure 7. Robot using Handy Board and Tmote Sensor

research. For instance, many courses can benefit from them. Next, we briefly discuss how these robots can be used in wireless networks, sensor networks, network security, robotics and research experience for undergraduates.

4.2 Wireless Networks

Wireless adhoc networks maintain communication between mobile elements even when nodes move. Intermediate nodes serve as relays for communication. Simulation is the major tool used due to the lack of equipment for wireless ad hoc networks. Although it is possible to put together an adhoc network using laptops, adding mobility to the mix is difficult. Robots developed can be used to support mobility for experimental results. Multiple mobile robots can be placed in a room and programmed to move according to various mobility patterns. The sensors can be used for communication between the robots. This allows collection of experimental results that can be replicated.

4.3 Sensor Networks

Sensor networks are deployed in an ad hoc manner for collecting data from a region of interest over a period of time. Deployments include ecology monitoring, habitat monitoring and military surveillance. Communication power consumed is proportional to d^c where d is the distance between the nodes and c is a system dependent constant $2 < c < 3$. To increase the lifetime of sensor nodes mobile robots can be used. Robots can visit nodes and collect data instead of sensors passing data to each other for collection. Projects/assignments in sensor networks such as data aggregation and broadcast can use mobile robots for power efficient operation.

4.4 Network Security

With the increase in use of wireless and sensor networks, security in wireless networks has received a lot of attention. Suitable cryptographic primitives can be implemented with the limited resources of sensors. Robots can be used to demonstrate many protocols such as key exchange, authentication and encryption. LCD screen on the Handy Board can be used to display data and this data can be used to test whether the protocol works. Many cryptographic protocols assume a central authority trusted by all the nodes. Mobile robot can be programmed to serve as central authority in such protocols. When a sensor needs to contact the central authority it can send a message to the robot and the robot can move to the location of the sensor.

4.5 Robotics Course

Students learn better when they can see the result of their work. Proposed robots can be easily used in robotics courses for this purpose. Both the LEGO/Handy Board and Tmote Sky Sensors are easy to program and together provide a very rich set of functionalities and can be used for a large number of applications. For example, the temperature sensor on Tmote Sky node can be used to find the position of a heat source in a room.

4.6 Research Experience for Undergraduates

Undergraduate students who take any of the above described courses can do research to learn more about the topic. Over the years we have observed that many undergraduate students like to work on well-defined manageable research projects. Through independent studies mobile robots can be used to implement a large number of projects spanning various areas. For instance, in Fall 2007, two undergraduate students have successfully completed one research project involving one mobile sensor robot for data collection in wireless sensor networks. Moreover, students can enhance the capabilities of the mobile robot by adding a camera to the robot or program the robots so that multiple robots move in a room without colliding to each other.

5 Conclusion

We successfully connect Tmote Sensor and Handyboard to enhance LEGO robots with RF communication capability. The process is explained in detail and easy to replicate using a few simple tools. Developed mobile robot has a rich set of functionalities and can be used for various applications. Many courses including wireless networks, sensor networks, robotics, network security and distributed systems can benefit from the use of these robots.

References

[1] Fred G. Martin, *Robotic Explorations: A Hands-on Introduction to Engineering*, Prentice Hall, 2001.

[2] R. Hartenstein. The digital divide of computing. In *ACM International Conference on Computing Frontiers (CF 04)*, pages 357–362, Italy, April 2004.

[3] Department of Trade and Industry (DTI), London,UK. *no*, 2001.

[4] Frank Klassner and Christopher Continanza. Mindstorms without robotics: An alternative to simulations in systems courses. In *SIGCSE 2006*, 2006.

[5] Michael Jipping, Cameron Calka, Brian O'Neill, and Christopher Padilla. Teaching students java bytecode using lego mindstorm robots. In *SIGCSE 2007*, 2007.

[6] Daniel Stevenson and James Schwarzmeier. Building an autonomous vehicle by integrating lego minstorms and a web cam. In *SIGCSE 2007*, 2007.

[7] Judith Challinger. Efficient use of robots in the undergraduate curriculum. In *SIGCSE 2005*, 2005.

[8] Cynthia Hood and Dennis Hood. Teaching programming and language concepts using legos. In *ITiCSE 2005*, 2005.

[9] NSF. Women, minorities, and persons with disabilities in science and engineering: 2002. http://www.nsf.gov/sbe/srs/nsf03312/start.htm.

[10] Autonomous Robotics. http://www.eecs.case.edu/courses/lego375/, 2007.

[11] Introduction to Robotics. http://www-scf.usc.edu/ csci445/, 2007.

[12] Jay Vegso Interest in CS as a Major Drops Among Incoming Freshmen In *Computing Research News* Vol. 17, No. 3, 2005 available at *http://www.cra.org/CRN/articles/may05/vegso*

[13] LEGO Mindstorms, http://mindstorms.lego.com/, 2007.

[14] Handyboard *Website for Handyboard*. http://www.handyboard.com, 2007

[15] Moteiv Corp., Tmote Sky Develop Kits http://www.moteiv.com/products/tmotesky.php, 2007

[16] Joe Bergin and Raymond Lister and Barbara Boucher Owens and Myles McNally The first programming course: ideas to end the enrollment decline, In *SIGCSE Bull.*, Vol. 38, No. 3, 2006

[17] Kam K. Leang, RF Serial Communication for the MIT Handy Board. available at *http://www.leang.com/robotics/info/index.html*, 2007