

# APPLICATION NOTE

## **ABSTRACT**

This application note describes how a SC16C650 or SC16C650B can be used to implement an IrDA interface. This application note is also applicable to SC16C654B, SC16C654DB, and SC16C652B.

## **AN10219**

**Using SC16C650/SC16C650B to  
implement an IrDA interface**

Author: Dong Nguyen  
Supersedes data of 2003 Jul 25

2004 Jul 12

# Using SC16C650/SC16C650B to implement an IrDA interface

AN10219

## INTRODUCTION

This application note describes how a SC16C650 can be used to implement an IrDA interface. There are two sections to this application note – hardware and software. The hardware section shows how the SC16C650 is electrically hooked-up to an IrDA transceiver module; the software section shows how the code can be written to configure the SC16C650 in IrDA mode (115.2 kbit/s max.).

## HARDWARE INTERFACE

The electrical connection between the SC16C650 and an IrDA transceiver module is very straightforward, no external component is required except an inverter to invert the IrDA output signal. Figure 1 shows this connection between the SC16C650 and an IrDA transceiver module made by Vishay Semiconductors. Besides the power supply noise-filtering components, the only other component required is a 14  $\Omega$  resistor (this value is recommended by the manufacturer). This resistor sets the current through the IR emitter, hence the power output of the transceiver (please consult your transceiver module manufacturer datasheet for other values).

## SOFTWARE CODING

Codes to be written to initialize the SC16C650 to operate in IrDA mode can be divided into two sections:

1. codes to set the communication parameters: baud rates, word length, parity and stop bits.
2. codes to turn on IrDA mode.

The following codes show, as an example, how the above two sections can be written:

```
/* This program configures the UART to operate in IrDA mode. It sits in a loop
waiting for a character to be received by polling the receive ready bit of the LSR
register, if the bit is set it will read the character and display it on the screen.
Otherwise it will poll the keyboard to see if there is a key struck, if there is
will send this key character out */
```

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>

#define PORT 0x2e8
/* Defines Serial Ports Base Address */
/* COM1 0x3F8 */
/* COM2 0x2F8 */
/* COM3 0x3E8 */
/* COM4 0x2E8 */

/*register addresses*/
#define RHR PORT+0 /* R ; Receive Hold Reg */
#define THR PORT+0 /* W ; Transmit Hold Reg */
#define IER PORT+1 /* R/W ; Interrupt enable reg */
#define FCR PORT+2 /* W ; FIFO Control reg */
#define IIR PORT+2 /* R ; Interrupt ident. reg */
#define LCR PORT+3 /* R/W ; Line Control Reg */
#define MCR PORT+4 /* R/W ; Modem Control Reg */
#define LSR PORT+5 /* R ; Line Status Reg */
#define MSR PORT+6 /* R ; Modem Status Reg */
#define SPR PORT+7 /* R/W ; Scratch reg */
```

## Using SC16C650/SC16C650B to implement an IrDA interface

AN10219

```

    /*ext reg addr accessible only when LCR bit-7 is 1 */
#define DLL      PORT+0      /* R/W      ; Divisor latch LSB*/
#define DLH      PORT+1      /* R/W      ; ,,      ,,  MSB */

    /*ext reg addr accessible only when LCR is set to 0xBF */
#define EFR      PORT+2      /* R/W; Enhanced Feature Reg */
#define XON1     PORT+4      /* R/W      */
#define XON2     PORT+5      /* R/W      */
#define XOFF1    PORT+6      /* R/W      */
#define XOFF2    PORT+7      /* R/W      */

int read(int);          /*read from register in UART*/
void write(int,int);   /*write to register in UART*/

void main(void)
{
    int c;
    unsigned int c, ch;
    int key;
    unsigned int n;

    write(MCR, 0x10);    /* put UART in loopback mode */
    while(read(LSR)&0x01) /* clear RX FIFO */
        read(RHR);

    write(IER, 0x00);   /* Turn off interrupts - Port1 */

    /*          PORT 1 - Communication Settings          */

    write(LCR, 0x80);   /* SET DLAB ON */
    write(DLL, 0x02);   /* Set Baud rate - Divisor Latch Low Byte */
                        /* Default 0x03 = 38,400 BPS */
                        /*      0x01 = 115,200 BPS */
                        /*      0x02 = 57,600 BPS */
                        /*      0x06 = 19,200 BPS */
                        /*      0x0C = 9,600 BPS */
                        /*      0x18 = 4,800 BPS */
                        /*      0x30 = 2,400 BPS */

    write(DLH, 0x00);   /* Set Baud rate - Divisor Latch High Byte */
    write(LCR,0xBF);    /* To access EFR reg. LCR must be set to BF */
    write(EFR, 0x10);   /* enable enhanced features: IrDA*/
    write(LCR, 0x03);   /* 8 Bits, No Parity, 1 Stop Bit */
    write(FCR, 0x0f);   /* FIFO reg.: DMA mode 1, clear TX and RX fifo */
    write(MCR, 0x4b);   /* turn on IrDA */

    do { c = read(LSR); /* Check to see if char has been received */
        if (c & 1) {ch = read(RHR); /* If so, then get Char */
                    printf("%c",ch); /* Print Char to Screen */

                    if (kbhit()){ch = getch(); /* If key pressed, get Char */
                                write(THR, ch); /* Send Char to Serial Port */
                    } while (ch !=27); /* Quit program when ESC(ASC 27) is pressed */
        }
}

```

## Using SC16C650/SC16C650B to implement an IrDA interface

---

AN10219

```
}  
  
int read(int regist)  
{  
    int rd;  
    rd=inportb(regist);  
    return (rd);  
}  
  
void write(int regist, int code)  
{  
    outportb(regist,code);  
}
```



# Using SC16C650/SC16C650B to implement an IrDA interface

AN10219

## REVISION HISTORY

Rev	Date	Description
_3	20040712	<b>Application note; third version (9397 750 13516). Supersedes data of 25 July 2003 (939775011631).</b> Modifications: <ul style="list-style-type: none"> <li>• Added SC16C650B to title.</li> <li>• 'Abstract' re-written to reference additional part-types.</li> </ul>
_2	20030725	<b>Application note; second version (9397 750 11631). Supersedes data of 24 April 2003 (939775011423)</b>
_1	20030424	<b>Application note; initial version (9397 750 11423).</b>

## Definitions

**Short-form specification** — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

**Limiting values definition** — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

**Application information** — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

## Disclaimers

**Life support** — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

**Right to make changes** — Philips Semiconductors reserves the right to make changes in the products—including circuits, standard cells, and/or software—described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

## Contact information

For additional information please visit  
<http://www.semiconductors.philips.com>. Fax: +31 40 27 24825

© Koninklijke Philips Electronics N.V. 2004  
 All rights reserved. Printed in U.S.A.

Date of release: 07-04

For sales offices addresses send e-mail to:  
[sales.addresses@www.semiconductors.philips.com](mailto:sales.addresses@www.semiconductors.philips.com)

Document order number:

9397 750 13516

*Let's make things better.*