BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Tomul LX (LXIV), Fasc. 2, 2014
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

TRANSMITTING DATA PACKETS USING THE CC1000 DEVICE

BY

PETRUȚ DUMA*

"Gheorghe Asachi" Technical University of Iaşi
Faculty of Electronics, Telecommunications and Information Technology

Received: May 15, 2014
Accepted for publication: June 26, 2014

Abstract. This paper presents the hardware interface for the CC1000 transceiver, which sends data using a development system equipped with a microcontroller from the ATMEL family. The command program schedules the transceiver and assures the transmission of the data packets, along with other general or communication commands with a personal computer.

Key words: transmitting data packets; interfacing the CC1000 transceiver; ATMEL microcontroller; command program.

1. Introduction

The interconnecting of processes is done by wired connections, through which various commands are being sent and received. If the distance between processes is increased to tens/hundreds of meters, we get greater costs because of the increase in length of the cables and the required interfaces at the ends. When connecting a new machine or reconnecting a repositioned installation, we get higher costs and connection problems that can lead to the interruption of processes for a period of time. Connection equipment designers, with the goal to eliminate the need for wiring, have implemented the wireless technology, which offers greater flexibility and a larger operating radius.

* *e-mail*: pduma@etti.tuiasi.ro

A wireless module for the command and control of user processes contains a transceiver for radio communication, a microcontroller and an interface for receiving and sending data. In order to meet these demands, it is used the CC1000 device made by Chipcon from Texas Instruments, a device which is designed for low-power and reduced voltage applications.

The basic schematics of a wireless module that manages a user-run process is illustrated in Fig.1; the notes mean: T_CC1000 – CC1000 transceiver used for sending data (*T*x) or for receiving data (*R*x); TIM – transceiver interface module; DS/AS – development system/ application system equipped with a microcontroller from the ATMEL family; UPI – user process interface; UP – user process; SI_RS232 – RS232 serial interface; PC – personal computer; SFM – serial flash memory; DDC – data display console (LCD).

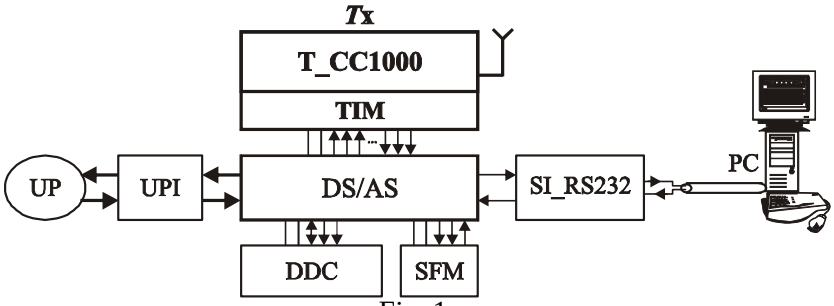

Fig. 1.

The sent data packets are stored in the serial FLASH memory, along with the status of the user process. The data display console is optional and shows permanently on an LCD the real time clock software, the status of the user process and the data packets.

This paper covers the CC1000 transceiver for sending data. Following papers will present the reception of data and other necessary functions for realizing a complete system.

2. Sending Data with the CC1000 Circuit

The CC1000 is a single chip transceiver, which works in the UHF band and is used in low-power wireless applications. The circuit is designed for ISM (Industrial, Scientific, Medical) and SRD (Short Range Device) frequency bands, which work on 315 MHz, 433 MHz, 868 MHz and 915 MHz frequencies, but can be used in any frequency from 300 MHz to 1 GHz. The transceiver command is made with a dedicated programming serial bus, while the data transmission/reception is made through another data serial bus. The circuit is used with a microcontroller and the interfacing requires a minimum of external components.

The main characteristics of the transceiver are: a programmable output

power which ranges from –20 dBm to +10 dBm; it does not require an RF switch and an IF filter; it uses a single antenna port when operating for data transmission/reception; it uses the FSK modulation with a transfer rate up to 76.8 Kbit/s; the frequency of the oscillator is programmable in 250 Hz steps; it is used at frequency hopping protocols, etc.

The internal block schematics of the CC1000 transceiver, illustrated in a simple way, is shown in Fig. 2; the notes used are: PA – power amplifier; *S* – switch for sending/receiving data; VCO – voltage controlled oscillator; LPF – low pass filter; CP – charge pump; PD – phase detector; FD – frequency divisor; RD – reference divisor; Osc – quartz oscillator circuit (for data transmission); TCCB – transceiver command and control block; SPI – serial programming interface; SDI – serial data interface; BIAS – circuit polarization block (for the transceiver's command); LNA – low noise amplifier; *M* – mixer; IFS – intermediary frequency stage; DM – demodulator (for data reception).

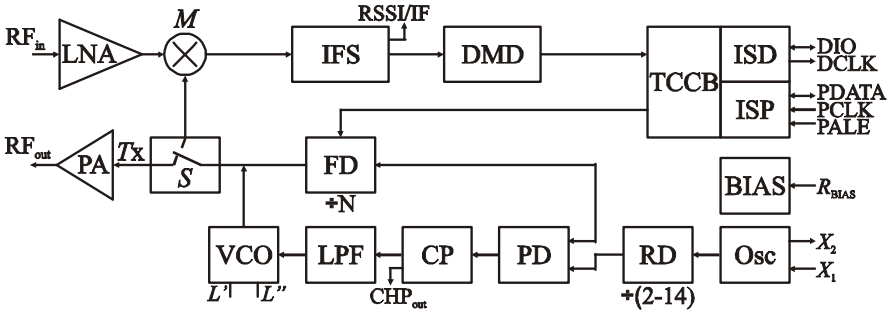

Fig. 2.

In data sending operating mode, the signal at the output of the voltage controlled oscillator is connected to the power amplifier through the transmission switch, which allows for simple interfacing and adaptation with the antenna. The modulation of the output signal of radio frequency is FSK and it is applied to the string of bits that are provided by the microcontroller at data input.

The frequency synthesizer consists of an oscillator driven by a quartz oscillator divided by 2…14 in order to obtain the signal with the reference frequency. The phase detector has, at one input, the reference signal connected and, at the other, the signal from the output of the voltage controlled oscillator through the frequency divisor. The output of the phase detector commands the charge pump whose output is passed through a low pass filter. Finally, the output filter signal commands the voltage control oscillator in order to generate the radio frequency signal.

The transceiver's serial programming interface is managed by the microcontroller for selecting the operating mode (transmitting data), the radio frequency, the output power, the parameters of the frequency synthesizer, the data transfer rate, the format of sent data, the reference frequency, etc.

The transmission of data from the microcontroller to the transceiver is made through a two-wire serial interface. The data is provided at the data input in the rhythm of the clock signal provided by the transceiver.

3. Interfacing the CC1000 Transceiver

The command and control of the CC1000 transceiver, used in the first stage for testing, checking and developing the application, is done with a development system equipped with the AT89S8253 microcontroller. In Fig.3 is presented the interface used for the management of this transceiver, which runs at a frequency of 868 MHz.

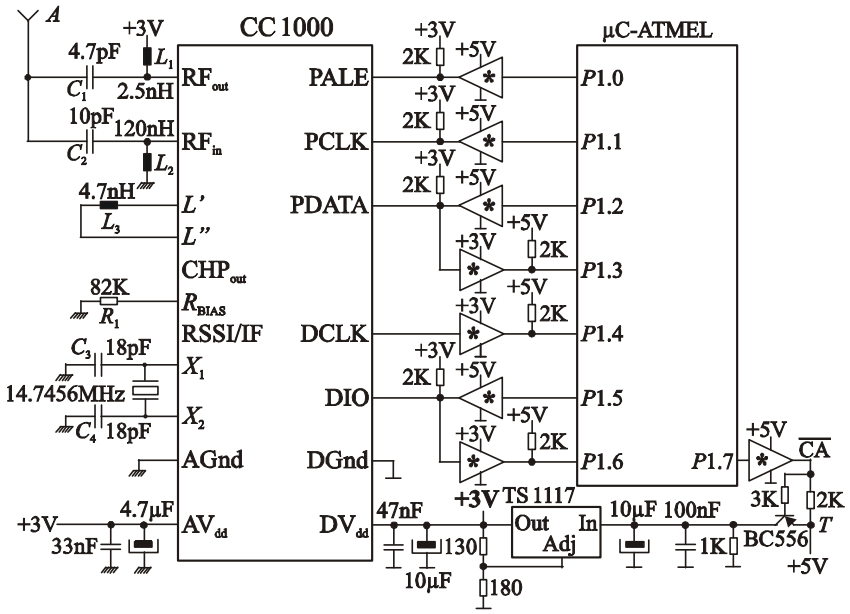

Fig. 3.

The C_1 capacitor (4.7 pF) and L_1 inductor (2.5 mH) are used to adapt the transmission impedance to 50 Ω. Based on an internal switch that selects the operating mode for transmitting or receiving data, it is possible to connect the RF output with the RF input and to adapt to 50 Ω in both cases. The C_2 capacitor (10 pF) and L_2 inductor (120 nH) form the adapting circuit for the receiver; furthermore the L_2 inductor is CC shock for the biasing current.

The voltage controlled oscillator is integrated in the transceiver, but the connection of an L_3 inductor (4.7 nH) to the external connections L' and L'' is needed; this inductor determines the interval of operating frequencies of the transceiver and it must have a high quality factor, a low tolerance and be placed closer to the external connections to reduce the effect of parasite inductance.

The CC1000 circuit has a quartz crystal driven oscillator with controlled amplitude. In order to start the oscillator a greater current is used, and as the amplitude of oscillations increases, the current is reduced until it is maintained at 600 mVpp. A quartz crystal XTAL and two capacitors (C_3, C_4 of 18 pF) are connected at the external connections of the oscillator (X_1 and X_2). The frequency of the oscillator is used as a reference frequency for the data transfer rate and for other internal functions; this frequency must be situated in the 3…4 MHz, 6…8 MHz or 9…16 MHz intervals. In this application, a quartz crystal with a frequency of 14.7456 MHz ensures a standard data transfer rate.

The CC1000 transceiver is powered in the DC voltage range from 2.1 V to 3.6 V, but a standard voltage of 3 V is recommended. In the built interface, the transceiver is powered at 3 V from the 3 point integrated adjustable positive voltage stabilizer (TS 1117), which requires at the external connections two resistors for establishing the output voltage and some filtering-decoupling capacitors. The powering command is done through an open drain buffer and a transistor *T* (BC556); the $\overline{\text{CA}}$ signal for powering the circuit is active on "0" logical level and it is supplied by *P*1.7 line of the microcontroller through software.

The filtering and decoupling of the supply are strictly needed in order that the transceiver be capable to reach a high operating performance.

The serial bidirectional PDATA line, the clock input PCLK and the address latch enable input PALE (for the transceiver programming), along with the bidirectional DIO data line and the DCLK clock output, are connected through open drain buffers at *P*1 port lines of the microcontroller (Fig. 3). These buffers are necessary for the logical level translation between the microcontroller of the development system supplied at +5 V and the transceiver, supplied at +3 V. The final hardware structure of the application will be managed by an application system with an ATMEL family microcontroller supplied at +3 V, in which case the level translation buffers will be no longer necessary.

4. Monitoring the CC1000 Device for Data Transmission

The programming of the CC1000 transceiver consists of the sending several data frames (Fig. 4), each having 16 bits, of which the first seven are address bits, then a read/write bit and finally 8 data bits.

During a writing frame of a command word, the PDATA line is only input. The command register is addressed using the first seven A_6 - A_0 bits of the frame; then, the read/write $\overline{R}/W$ bit is sent, having logical value "1" for writing and logical "0" for reading. During the transmission of the address bits and of the read/write bit, the PALE input must be kept in logical "0". Afterwards, the eight D_7,…,D_0 command bits are transmitted when the PALE input is kept in logical "1". At the end of a command frame reception operation, the addressed

register is loaded with the sent data. The data for configuring the transceiver are stored in an internal RAM memory and maintained during the supply of the circuit.

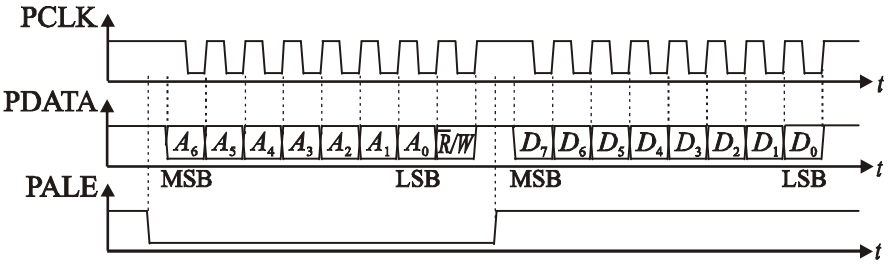

Fig. 4.

The structure of the reading frame of a command register is similar with the above described writing frame. In the first half of the frame the address bits are sent in the same way, as well as the read/write $\overline{R}/W$ bit which has logical value "0" (PDATA is input and PALE is logical "0"). In the second half of the frame in which PALE has the logical value "1", the PDATA line becomes an output and supplies the $D_7,\ldots,D_0$ data bits with the contents of the addressed register.

The clock signal, PCLK, is active on the descending edge. The bits from the frame structure have to stay the same for 10 ns (t_{SD}) before and, respectively, at least 10 ns (t_{HD}) after the descending edge. Moreover, PALE must be stable for at least 10 ns before and, respectively, after the clock signal edges (t_{SA}, t_{HA}).

The CC1000 transceiver has 21 configuration registers and 7 testing registers. In Table 1 are shown the numerical values of all frames with the first byte (address and $\overline{R}/W$ =1) and second byte, which are transmitted to the device for its programming and calibration in the data sending data operating mode, along with a brief description of the selected indicators.

The CC1000 circuit can be configured by the user for sending data in one of the following three ways: NRZ synchronized mode; synchronized Manchester encoding mode and transparent asynchronous UART mode.

For sending data in the NRZ synchronized mode, the CC1000 circuit supplies the clock signal at DCLK, while the data is supplied by the microcontroller to the DIO data entry. The data is loaded on the ascending edge of the clock signal (Fig. 5 *a*). In this operating mode, the data is modulated without encoding. The data transmission is made at a standard rate from 600 bit/s to 76.8 Kbit/s; in order to use the 38.4 Kbit/s and 76.8 Kbit/s, a 14.7456 MHz quartz crystal is required.

Table 1

Name	A_6-A_0	First Byte	Second Byte	Description
MAIN	00H	01H	BAH	*R*x*T*x = 1(*T*x), F_REG = 0(A), Rx_PD = 1, *T*x_PD = 1, FS_PD =1, CORE_PD = 0, BIAS_PD = 1, RESET_N=0(on)
MAIN	00H	01H	A1H	*R*x*T*x = 1(*T*x), F_REG = 0(A), Rx_PD = 1, *T*x_PD = 0, FS_PD = 0, CORE_PD = 0, BIAS_PD = 0, RESET_N= 1(off)
		FFH	02H	Wait 2 ms (not necessary)
FREQ_2A	01H	03H	58H	FREQ_*A* = 583313H = 5780243, f_{RF} = 868.2972 MHz
FREQ_1A	02H	05H	33H	
FREQ_0A	03H	07H	13H	
FREQ_2B	04H	09H	75H	FREQ_*B* = 75A000H = 7708672, f_{RF} = 868.2972 MHz
FREQ_1B	05H	0BH	A0H	
FREQ_0B	06H	0DH	00H	
FSEP1	07H	0FH	01H	FSEP = 1ABH = 427, Δf = 64.05 KHz
FSEP0	08H	11H	ABH	
CURRENT	09H	13H	F3H	VCO_CURRENT=*F*(2.55mA *T*x), LO_DRIVE=0(0.5 mA, *T*x), PA_DRIVE = 3(4 mA, *T*x)
FRONT_END	0AH	15H	32H	BUF_CURRENT=1(0.69mA), LNA_CURRENT=2(1.8mA), IF_RSSI=1(on), XOSC_BYPASS=0(on)
PA_POW	0BH	17H	00H	PA_HIGHPOWER=0, PA_LOWPOWER=0(PA off)
PLL	0CH	19H	30H	EXT_FILTER=0, REFDIV=6(÷6), ALARM_DISABLE=0, ALARM_*H*=0, ALARM_*L*=0
LOCK	0DH	1BH	10H	LOCK_SELECT=1(LOCK_CONTINUOUS), PLL_LOCK_ACCURACY=0, PLL_LOCK_LENGTH=0, LOCK_INSTANT=0, LOCK_CONTINUOUS=0
CALL	OEH	1DH	26H	CAL_START=0(off), CAL_DUAL=0, CAL_WAIT=1, CAL_CURRENT=0, CAL_COMPLETE=0, CAL_ITER=6
MODEM2	0FH	1FH	90H	PEAKDETECT=1(on), PEAK_LEVEL_OFFSET=10H
MODEM1	10H	21H	6FH	MLIMIT=3, LOCK_AVG_IN=0, LOCK_AVG_MODE=1, SETTLING=3, MODEM_RESET_*N*=1
MODEM0	11H	23H	47H	BAUDRATE=4 (9600Kbit/s), DATA_FORMAT = 1 (Manchester), Xosc_FREQ = 3(12…16 MHz, req. 14.7456 MHz)
MATCH	12H	25H	10H	RX_MATCH=1(*f*>500MHz), *T*x_MATCH=0
FSCTRL	13H	27H	01H	FS_RESET_*N*=1(off)
PRESCALER	1CH	2DH	00H	PRE_SWING=0, PRE_CURRENT=0, IF_INPUT=0, IF_FRONT=0
MAIN	00H	01H	A1H	*R*x*T*x=1(*T*x), F_REG=0(A), *R*x_PD=1, *T*x_PD=0, FS_PD=0, CORE_PD=0, BIAS_PD=0, RESET_*N*=1(off)
Calibration				
CALL	0EH	1DH	A6H	CAL_START=1(on), CAL_DUAL=0, CAL_WAIT=1, CAL_CURRENT=0, CAL_COMPLETE=0, CAL_ITER=6
		FFH	22H	Wait 34 ms (or test CAL_COMPLETE)
CALL	0EH	1DH	26H	CAL_START=0(off), CAL_DUAL=0, CAL_WAIT=1, CAL_CURRENT=0, CAL_COMPLETE=0, CAL_ITER=6
PA_POW	0BH	17H	80H	PA_HIGHPOWER=8, PA_LOWPOWER=0, PA on,16.8 mA

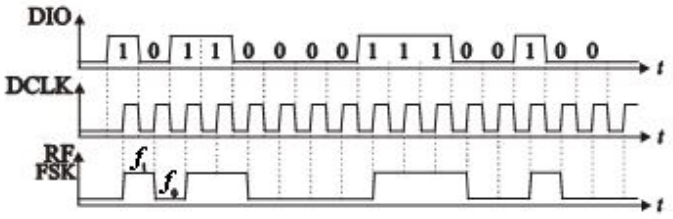

Fig. 5 *a*.

For transmitting data in the synchronized Manchester encoding mode, the circuit supplies the clock signal at DCLK, and the NRZ data is sent by the microcontroller to the DIO input and are loaded on ascending edge of the clock signal (Fig. 5 *b*). In this operating mode, the data is modulated using the Manchester code where a logical "0" bit is encoded as a rising edge in the middle of the bit interval, while a logical "1" bit is encoded on the descending edge in the middle of the bit interval. Sending data in this case is done with a standard transfer rate of 300 bit/s, up until 38.4 Kbit/s; the halving of the data transfer speed is due to transitions in the Manchester code.

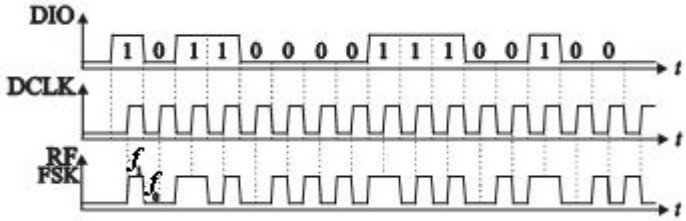

Fig. 5 *b*.

For transmitting data in the transparent asynchronous UART mode, the circuit does not supply the clock signal at DCLK, but data is supplied by the microcontroller to the DIO (Fig. 5 *c*).

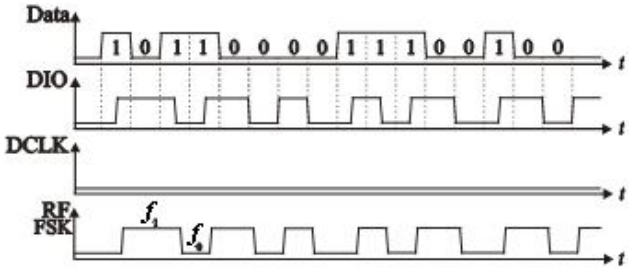

Fig. 5 *c*.

In this operating mode, the transceiver performs neither the bit synchronization, nor the data encoding. These operations must be accomplished

by the microcontroller through software. Sending data is carried out with a standard data transfer rate of 600 bit/s to 76.8 Kbit/s.

The data packets sent by the microcontroller to the CC1000 device consist of many fields and their structure is shown in Fig 6. The notes have the following meaning: MP – message prefix (for the locking of the averaging filter and for the preamble detector); BS – Barcker sequence (ensures the byte synchronization at reception); MH – message header (has the same numerical value for a certain group of user processes); SA – source address (the address of the data packet sending process); DA – destination address (the address of the process which is going to receive the data packets); MT – message type (specifies using a numerical value the function which needs to be performed at the receiving end); DBL – data block length; MS – message separator (separates the header of the data packet from the numerical values of the data block); DB – data block; CS – checksum (performs the logical function XOR between all the data block bytes); EM – end of message (a terminator is sent at the end of the data packet).

MP	BS	MH	SA	DA	MT	DBL	MS	DB	CS	EM
8/16 bytes	7 bits	2 bytes	2 bytes	2 bytes	1 byte	1 byte	1 byte	Max256 bytes	1 byte	2 bytes
8/16x55H	72H	FEFDH	xxxx	xxxx	xx	xx	AAH	….	xx	AAAAH

Fig. 6

The command program consists of a segment of initializations which loads the variables of the application for sending data, erases the console screen, then shows a startup message. Then the program reaches a keyboard interrogation loop through which the user sends different specific commands for the application. The meaning of the data sending commands, implemented through software, will be described briefly.

a) The command (**A.**) for powering the transceiver in DC; the command produces the connection ($\overline{CA}$ =0) / disconnection ($\overline{CA}$ =1) of the power source.

b) The command (**S.**) for entering low power consumption mode; it has the role of connecting/disconnecting from this operating mode.

c) The command (**PT.**) for programming the transceiver for transmitting data; this command uses the programming table of the circuit with the specified numerical values. The microcontroller extracts from Table 1 the first and the second byte for each frame and programs the transceiver register with the specified address. If the first byte is FFH, then a delay is inserted with the duration specified by the second byte (in ms).

d) The command (**CT.**) for configuring the transceiver for transmitting data; it displays every indicator of the device and possible values, after which the user selects the convenient numerical value from the console's keyboard, then the microcontroller forms and sends the frame with the selected numerical

value. This command is useful in the stages of application development and performance testing of the circuit.

e) The data transmission command (**TX.**); it produces a software image of the data packet in the RAM memory of the microcontroller system. The data block can be specified by the user or can represent the data acquired from the current user process. After the data packet is sent using the CC1000, it is saved in the FLASH serial memory and it is displayed on the console.

5. Conclusions

The described hardware structure is built in practice, it is simple and consists of the CC1000 transceiver which requires some passive external components and a development system equipped with an ATMEL microcontroller. It is used in short distance (tens/hundreds of meters) data packet transmission applications between various user processes.

The written command program implements a few commands and subroutines, which perform the programming of the transceiver and the transmission of data packets. If the UART transparent asynchronous operating mode is used, the program performs the bit synchronization along with the Miller software encoding. The program is written in machine language, occupies 2.8 Kbytes of program memory and stands out by the small memory storage required compared to the offered facilities.

The wireless module communicates serially asynchronous with a PC in order to send and receive various commands and data, for displaying sent data in different formats, for displaying the state of the user process, for downloading the FLASH memory into the computer, etc.

REFERENCES

Bălan C., *Tehnica Radiocomunicațiilor* I. Edit. Acad. Tehn. Milit., București, 2009.

Bozomitu R.G., *Radioemițătoare și Radioreceptoare.* Edit. Performantica, Iași, 2012.

Duma P., *Microcontrolerul INTEL 8051. Aplicații*. Edit. TEHNOPRESS, Iași, 2004.

Hintz J.K., Tabak D., *Microcontrollers. Architecture, Implementation and Programming*. McGraw Hill, New York, 1993.

Mârza E., Alexa F., Simu C., *Radiocomunicații. Fundamente.* Edit. Vest, Timișoara, 2007.

Peatmann B.J., *Design with Microcontrollers*. McGraw Hill, New York, 1998.

* * * *ATMEL, Family Microcontroller*. Data Book, 1998.

* * * *ATMEL, AT89S8253 Microcontroller*. Data Sheet, 2005.

* * * *CHIPCON, Single Chip Very Low Power RF Transceiver*. CC1000 Data Sheet, 2007.

* * * *CHIPCON, CC1000PP Plug and Play Module*. User Manual, 2007.

* * * *CHIPCON, CC1000DK Development Kit 868/915 MHz.* User Manual, 2007.

* * * *CHIPCON, Antenna Selection Guide*. Application Note, 2007.

* * * *CHIPCON, RF Module Testing Using Smart RF Studio*. Design Note, 2007.

* * * *CHIPCON, Interfacing CC1100-CC2500 to the MSP430*. Design Note, 2009.
* * * *CHIPCON, Packet Transmission Basics.* Design Note, 2009.
* * * *MAXIM, Multichannel RS232 Drivers/Receivers*. MAX232A Data Sheet, 2006.
* * * *Taiwan Semiconductor*. TS1117 Data Sheet, 2003.
* * * *Texas Instruments*. Data Book, 1992.

TRANSMITEREA PACHETELOR DE DATE UTILIZÂND CIRCUITUL CC1000

(Rezumat)

Se prezintă interfaţa hardware pentru transceiverul CC1000 care transmite date, utilizând un sistem de dezvoltare echipat cu microcontroler din familia ATMEL. Programul de comandă realizat programează transceiverul şi asigură transmiterea pachetelor de date, alături de alte comenzi generale sau de comunicare cu un calculator personal.