

## Firmware Release Document

The software document is the sole property of Toptrend Technologies CORP. All rights, title, ownership, or other interests in the software remain the property of Toptrend Technologies CORP. This document and the corresponding software should only be used in accordance with the corresponding license agreement. Any unauthorized use, duplication, transmission, distribution, or disclosure of the software; the contents of this document is expressly forbidden.

This copyright notice should not be removed or modified without prior written consent of Toptrend Technologies CORP. Toptrend Technologies CORP. reserves the right to modify this software without notice.

<b>Project</b>	8535_SmartRF Firmware
<b>Version</b>	1.1
<b>Revision Date</b>	June/2002
<b>Classification</b>	Firmware Addendum
<b>Distribution</b>	AT86RF211
<b>Released By</b>	Mikeal Wang

### ***Introduction:***

This document describes the 8535-RF211 Development Kit for Toptrend Technologies CORP. The firmware provides isolated modules and several head files. User can take one out from this and doesn't modify it. Customer can include any module in his project only from this SC. All function just support RF211 h/w v2-04 please checks your h/w version. The project was developed with the ICCAVR V6.25 and used ANSI C code. The RF211 control code modules does refer to demo board, the first please read the AVR-RF211 v2-04 Software Description. All functions are base on AVR-RF211 v2-04.

### ***Feature:***

- 1.Support UART test function.
- 2.Support EEPROM r/w function.
- 3.Support TRX01 registers r/w function.
- 4.Support PD/TX/RX mode switch.
- 5.Support data carry/accept function.
- 6.Support data rate 50k/10k/9.6k/4.8k/2.4k/1k.
- 7.Support great data carry/accept function. (Disable)
- 8.Support just ANSI ASCII code (0-255).

### ***Project Code Modules:***

MainApp.c	The main function and control user interface from PC to AVR.
-----------	--

UARTControl.c	Includes UART controls functions.
EEPROMControl.c	Includes EEPROM controls functions.
TimeControl.c	Control the pauses and timeouts in applications with the AVR. <b>Please refer to AVR-RF211 v2-04 Software Description.</b>
TRX01Control.c	Control the RF211 functions. <b>Please refer to AVR-RF211 v2-04 Software Description.</b>

### ***Command Operate Descriptions:***

Please refer to SmartRF-AT90LS8535 Firmware Process V1.0 PDF file.

### ***Module Descriptions by Toptrend:***

- **Name:** Init\_AVR

**Include file:** BasicDef.h

**Syntax:** Void Init\_AVR(Void);

**Description:** Initial ATMEL AT90S8535 micro-controller. We just used six in/output pins to connect with RF211 chip and controls RF211's operation state. The other in/output pins that we do not use. The static of pins, please previews content of firmware or refer to descriptions as below. If you want to get detail, please read the RF211 datasheet. All I/O ports should be set pull-up if we do not mark.

---

PortC.0	In/Output Pin	DATAMSG (Default: Input)
PortC.1	Output Pin	SLE
PortC.2	Output Pin	SCK
PortC.3	In/Output Pin	SDATA (Default: Output)
PortD.2	Input (pull-up)	DATACLK (link to the TRX01)
PortD.3	Input (Not pull-up)	WAKEUP from the TRX01 or from the push button on board (Non pull-up input in order to easily be tied externally)

---

The UART baud rate was defined 19200 (Default) and set by manufacture before programming.

About AT90S8535 micro-controller please refer to datasheet.

- **Name:** Init\_TRX01

**Include file:** TRX01Def.h

**Syntax:** void Init\_TRX01(void);

**Description:** Initial RF211 chip with the configuration function defined in the file: "TRX01Conf.h". The item of setting as below:

**RESET\_TRX01();**

Reset RF211 chip.

**SETR\_TRX01\_CTRL1(MY\_TRX01\_CTRL1\_INITVAL);**

To set RF211 Control Register 1. The content of setting as below:

PDN 0 (off);RXTX 1 (TX);DATACLK 0 (DTCLK off);TXLOCK 1 (on);PAPDN 0 (PA off);WUEN 0 (off);LNAGSEL 0 (max gain);MVCC 0 (MRSSI);TRSSI 010000 (RSSI level);HRSSI 010 (RSSI hyst.);TXLVL 111 (PA power);TXFS 00 (F0-F1);RXFS 10 (F2);XTALFQ 0 (div/1);FSKBW 0 (25kHz);FSKPOL 1 (direct);DSREF 0 (external).

In this case, when start to program RF211 firmware that will set on PD mode and maximum TX power. If you want to get detail, please read the RF211 datasheet.

**SETR\_TRX01\_CTRL2(MY\_TRX01\_CTRL2\_INITVAL\_96);**

To set RF211 Control Register 2. The content of setting as below:

DATARATE 42B (00010000101011);DATATOL 16 [3%](00010110);LDCK 0 (simple);N0LD2 0010 (unlock condition);N1LD2 10111 (lock condition).

The data rate is set 9600bps and 2% error rate. If you want to get detail, please read the RF211 datasheet.

**SET\_TRX01\_F0\_INIT;**

**SET\_TRX01\_F1\_INIT;**

**SET\_TRX01\_F2\_INIT;**

To set RF211's Frequency Registers on 915MHz  $\pm$  10 kHz deviation. These parameters must be set by manufacture before programming. User can adjust this value "TRX01\_FREQ" to change from this to the other frequency. This firmware provides three channels 433.5MHz, 868MHz and 915MHz, if you want to get detail please read the RF211 datasheet.

- **Name:** DWordToChar

**Include file:** No

**Syntax:** void DWordToChar(unsigned long ChangeData);

**Description:** The function provides DWORD data type transform into the CHAR data type. Where do you need to use this function? Because the Utility Serial Interface just supports ASCII code and the char must be smaller than 0x80 hex value, so we need this function to accept data from utility.

- **Name:** CharToDWord

**Include file:** No

**Syntax:** long CharToDWord(**unsigned char** ChangeData1, **unsigned char** ChangeData2,  
**unsigned char** ChangeData3, **unsigned char** ChangeData4,  
**unsigned char** ChangeData5);

**Description:** The function provides the CHAR data type transform into the DWORD data type.

This function does same as the DwordToChar function. Our data usually is DWORD from TRX01 reg. or CHAR from PC, so you need to use this function and to exchange data type thereafter let PC can get it.

All char must be ASCII code.

● **Name:** UARTReceiveByte

**Include file:** UARTDef.h

**Syntax:** **unsigned char** UARTReceiveByte( **void** );

**Description:** This function provides the base protocol for UART interface. All the command and data must be use this function for transmission and receiver from PC to AVR. If the received value is correct, the return is READY, else RESEND.

**Format:**

Name	HAND	COM	ADD	DATA0	DATA1	DATA2	DATA3	LEN	TAIL
Size (byte)	1	1	1	1	1	1	1	1	1

HAND: This field must be set UART\_HAND\_MASK.

COM: The command field has support some function as below:

Command	Description
UART_UART_TEST	UART test
UART_Eep_WRITE	Write data into the EEPROM
UART_Eep_READ	Read data from EEPROM
UART_TRX01_WRITE	Write data into the RF211's registers
UART_TRX01_READ	Read data from RF211's registers
UART_TRX01_SEND	Send standard package with RF211
UART_TRX01_RECEIVER	Receive RF211 data
UART_TRX01_HUGE_MODE	Into the great data mode for TX/RX
UART_TRX01_WAKEUP_MODE	Into the Wake-up mode (Not support yet)

ADD: The address field provides a parameter to control register address. When you r/w RF211 register, you need a parameter to let RF211 to know what register will be r/w.

Data0-3: The true data buffer for transmission or receiver.

LEN: The length same as ADD field. The program provides length field to control buffer size for r/w.

TAIL: This field must be set UART\_TAIL\_MASK.

● **Name:** UARTTransmitByte

**Include file:** UARTDef.h

**Syntax:** void UARTTransmitByte( **unsigned char** hand,**unsigned char** command,\n**unsigned char** address,**unsigned char** data1,\n**unsigned char** data2,**unsigned char** data3,\n**unsigned char** data4,**unsigned char** length,**unsigned char** tail );

**Description:** Refer to UARTReceiveByte function descriptions.

● **Name:** CheckUARTCommand

**Include file:** UARTDef.h

**Syntax:** **unsigned char** CheckUARTCommand(**unsigned char** getdata);

**Description:** This function is support customer to check command, if the command is fails return fail,  
else true.

● **Name:** TEEPROMReady

**Include file:** EEPROMDef.h

**Syntax:** **unsigned char** TEEPROMReady(**void**);

**Description:** This function is test EEPROM state if the EEPROM is ready for r/w operation, returns non-zero if ready, else not ready.

● **Name:** TEEPROMwrite

**Include file:** EEPROMDef.h

**Syntax:** void TEEPROMwrite(**int** locationH, **int** locationL, **unsigned char** databyte);

**Description:** To writes data to the specified location. At this part of AVR, EEPROM supports high bit and low bit to control offset.

● **Name:** TEEPROMread

**Include file:** EEPROMDef.h

**Syntax:** **unsigned char** TEEPROMread(**int** locationH, **int** locationL);

**Description:** To reads a byte from the specified location. Return data within EEPROM.

● **Name:** UARTReceive30Byte

**Include file:** No

**Syntax:** void UARTReceive30Byte(**void**);

**Description:** The function provided us receive 30 bytes once. In this program that will always wait for data, so if data from PC not correct transmit or receive, micro-controller will be stop and

knot and wait next data. We used the standard method by ATMEL's C code to receive data from PC.

**Format:**

Name	DATA[0..29]
Size (byte)	30

**Frame: (Receive loop once)**

```
while ( !(USR & (1<<RXC)) ) ; // wait for incoming data
unsigned char parameter = UDR; // return the data
```

- **Name:** UARTTransmit30Byte

**Include file:** No

**Syntax:** void UARTTransmit30Byte( unsigned char TransBit[30], unsigned char workstate );

**Description:** This function provides us total 33 bytes data include head byte, tail byte and state byte.

When we received data 30 bytes and need to return Receiver State, user can use this function transmitting to the PC. The parameter "workstate" that can help us to know the Receiver State. We used the standard method by ATMEL's C code to transmit data to the PC.

**Format:**

Name	HAND	DATA[0..29]	WORKSTATE	TAIL
Size (byte)	1	30	1	1

**Frame: (Transmit loop once)**

```
while ( !(USR & (1<<UDRE)) ) ; // wait for empty transmit buffer
UDR = 0x40; // start transmission
```

***Reference Documents:***

AT86RF211(TRX01) Datasheet Rev. 1942B-11/00

AT90LS8535 Summary Rev. 1041GS-09/01

AVR-RF211 v2-04 Software Description Rev. date: 17/07/01

User Guide – Annex2 Sleep Mode implementation 23/08/01 Rev. 2.0

Application Note - AT86RF211 (TRX01) Bill-Of-Material vs application requirements 17/04/01

Application Note – How to save power with TRX01 Wake-Up Protocol? 23/10/00

---

# TOPTREND

## TOPTREND TECHNOLOGIES CORP.

8F-2, No. 15, Lane 360, Sec. 1, Nei-Hu Road,  
Taipei, Taiwan, R.O.C.

Tel : 886-2-2659-3636 Fax : 886-2-2659-1518

E-mail : support@toptrend.com.tw

---

### ***Contact Us:***

Toptrend Technologies CORP.

TEL: +886-2-26593636

FAX: +886-2-26591518

Address: 8F-2, No. 15, Lane 360, Sec. 1, Nei-Hu Road, Taipei, Taiwan, R.O.C.

E-mail: Support@toptrend.com.tw

Web Site: <http://www.toptrend.com.tw/>