



RF RSSI Sniffer Tool for Car Access Systems

Chris Wunderlich, George Rueter

Introduction

The ability to monitor the received signal strength in a radio receiver is a valuable development and diagnostic tool for the engineer. In an automotive application the ability to monitor this received signal strength indicator (RSSI) value represents what a radio receiver will experience in service, including the effects related to antenna and mounting location. The article describes a simple method to re-configure the Atmel® ATA5830N device, utilizing its Flash program capability, to create an RSSI monitoring tool for laboratory and in-vehicle use.

The ATA5830N UHF transceiver chip incorporates a high-performance UHF transceiver and a low-power 8-bit AVR® microcontroller on a single die. Figure 1 shows a simplified block diagram of the ATA5830N device. Included in the device is 6KB of Flash memory available to the end user. This article describes an application that utilizes this Flash memory space to develop an application for an RSSI monitor that generates universal synchronous/asynchronous receiver transmitter- (USART-) formatted messages containing RSSI data. From an operational perspective this application lends itself well to several common RF engineering tasks such as: a) RF environment analysis, b) performance tuning of the

receiver section, c) RF component selection and d) antenna performance evaluation.

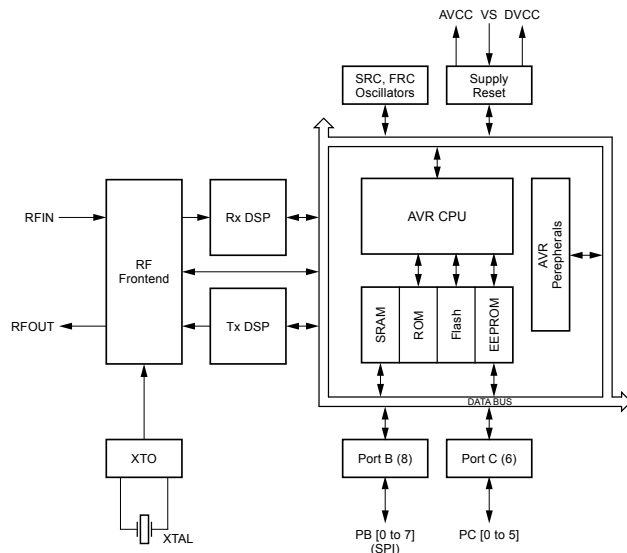


Figure 1. ATA5830N Block Diagram

Flash-based RSSI Application

RSSI data is a critical radio reception metric that reflects the level of RF signal energy at a given frequency channel and is useful in evaluating an RF environment. The Flash memory application uses the internal RSSI measuring function of the chip and communicates this value using a general-purpose I/O pin using standard USART communication protocol timing. This article describes how a Flash program application is implemented and demonstrates the flexibility of the internal AVR controller. The Flash application provides a small flexible RSSI measurement tool that can be easily and quickly configured to operate across a wide range of frequencies, providing a simple way to measure RSSI.

The software for this demo was developed using the ATAK51002-V1 evaluation kit. The application operates with no external inputs required other than 5V DC power and the RF signal source. The desired radio parameters, such as frequency and intermediate frequency bandwidth (IFBW), are programmed into the part using the EEPROM configuration file. The user selects these values by using an Excel spreadsheet tool that automatically generates the EEPROM file. Once these values are programmed into the EEPROM, the application of power will automatically initiate the self-configuration and execution of the Flash application program. It begins measuring and outputting the 8-bit RSSI data values on pin 17 (PC3) of the device. USART communication parameters are 38400 Baud, 8 data bits,

1 start bit and no parity bit (38400, 8-1-N). Each data byte output requires about 260µs to transmit and occurs about every 50ms. The output data is at logic value voltage levels (about 0V to 5V) and, while not compliant with RS-232 standards, most PCs have no problem in receiving and displaying this data stream using a terminal program.

RSSI Measurement Details

Received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal. In the ATA5830N chip, this function is built into the receiver's digital signal processing (Rx DSP) section and occurs completely in the digital signal domain, allowing for high relative and absolute RSSI accuracy.

In this application, RSSI values are returned as 8-bit values and the signal power at the matched 50Ω antenna input can be calculated in dBm as:

$$\text{RF Power In (dBm)} = \text{RSSI}/2 - 155 \text{ (154 in 868-928MHz band)}$$

The ATA5830N datasheet notes the following RSSI measurement specifications:

- Absolute RSSI accuracy $\pm 5.5\text{dBm}$
- Relative RSSI accuracy $\pm 1\text{dB}$
- RSSI resolution 0.5dB

The typical RSSI range is a function of the IFBW selected:

- IFBW = 25kHz >>> RSSI = -115dBm to -25dBm
- IFBW = 80kHz >>> RSSI = -110dBm to -25dBm
- IFBW = 165kHz >>> RSSI = -105dBm to -25dBm
- IFBW = 366kHz >>> RSSI = -100dBm to -25dBm

Software Operational Flow

On power-up and after the hardware-generated power-on reset is complete, the AVR controller will execute its main() function in ROM, where the EEPROM configuration settings are loaded and checked. This EEPROM configuration defines all operational parameters of the device, including that a Flash application program is present. When this EEPROM setting is encountered, the application flow jumps to the Flash main(), and the application flow begins. The Flash main() function can be very simple since the ROM software functions will perform most of the work. In this application, the Flash program function loops indefinitely, first requesting ROM-code-based RSSI measurements and then calling Flash code functions that format and output the data. The following is the Flash program loop that drives this application:



```
for(;;){
    _WDR;
    ATA_makeGlobalSystemFlags();
    RDCR |= (1 << RQSS);
    rssi=RSSI;
    if (rssiCnt==100){
        RSSI_flash();
        RSSI_TransmitValue(rssi);
        while ((T4CR & (1<<T4ENA)));
        PORTC &= ~(1<<PORTC3);
        TMCR = (TMCR & 0xF8) | SSI_PC3_OUTPUT;
        ATA_timer4Close();
    }
    rssiCnt++;
}
```

```
// ROM code check and execution of mode changes
// Register bit set to request RSSI measurement
// Store the RSSI register value in SRAM variable
//Every 100th reading output data to FLASH program (~50mS)
// FLASH program to initialize required timers
// FLASH program to format and output data
// Wait for data transmission to complete
// Set output port value
// Set the modulator MUX to output at PORTC
// Close the timer used by the FLASH functions

// increment counter
```

Operation and Use

Hardware Platform

The ATA5830N can be used in several frequency bands: 310MHz to 318MHz, 418MHz to 477MHz and 836MHz to 928MHz, operating from a single crystal. The highly integrated radio architecture is optimized for minimal external part count. See Figure 2 for the schematic of a low-band Flash-based application such as the one described in this article.

Setup

The only required external signals are a +5V power supply and the RF signal input. The reference design included in the ATAK51002-V1 evaluation kit consumes about 9mA when running this application, which is sufficiently low to enable battery-powered use. Once the reference design is powered up, the ATA5830N device must be awakened by momentarily connecting any of the "npwron" pins (15, 16, 17, 18, 19 and 29) to ground or the "pwron" pin to +5V. After the part is awake and active, no other input is required and the RSSI data will be available at PC3 pin 17.

RF Considerations

In addition to the EEPROM configuration settings, the hardware in the RF path must be optimized or "matched" to the specific frequency being used. The default configuration frequency for the ATAK51002-V1 evaluation kit is 433.9MHz,

so the application kit hardware is shipped with this hardware installed. If another frequency is selected, the matching component values in the RF signal path need to be changed.

The ATA5830N device includes two separate LNA inputs, one for low-band (315MHz and 433MHz) and one for high-band (869MHz to 902MHz) operation. These LNA inputs are provided since the chip is capable of multi-band operation and these separate ports provide optimum performance matching for each frequency range. The hardware configuration as shipped utilizes the low-band input port. The hardware must be modified accordingly to measure a high-band frequency.

A list of component values for several common radio frequency bands is available upon request and included in the evaluation kit. Installation of the corresponding hardware component values is required for optimal performance at the desired frequency.

PC Connection

Data is output at pin 17 (PC3), which can be accessed at one of the I/O pins on the hardware evaluation board. This signal can be connected directly to pin 2 of a PC-connected RS-232 port. While the voltage level is not compliant with RS-232 standards, most PCs will successfully receive this data. The data can then be displayed using a terminal emulation program such as "Realterm", available at no cost at <http://realterm.sourceforge.net/>.

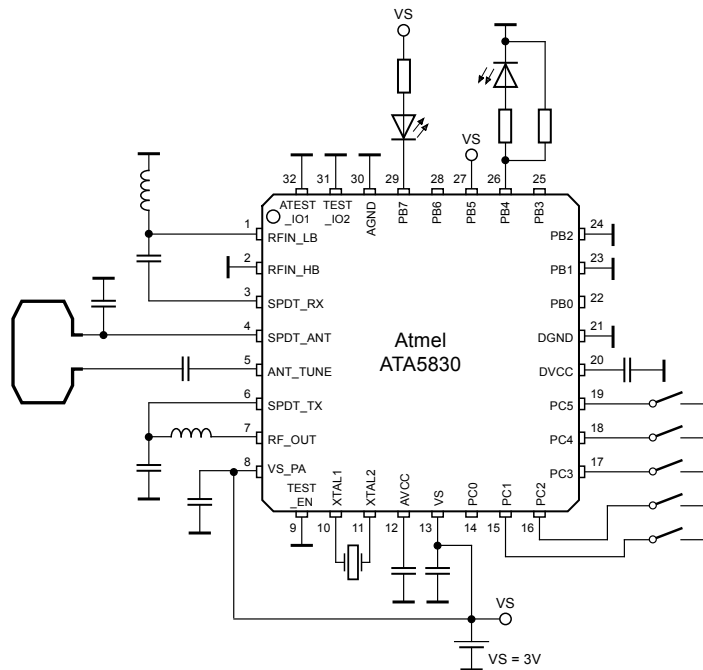


Figure 2. ATA5830N Flash Application Schematic

Data Collected

Figure 3 shows sample RSSI data collected using the described Flash application with the ATA5830N chip configured to operate at a center frequency of 314.9MHz with an IFBW of 165kHz.

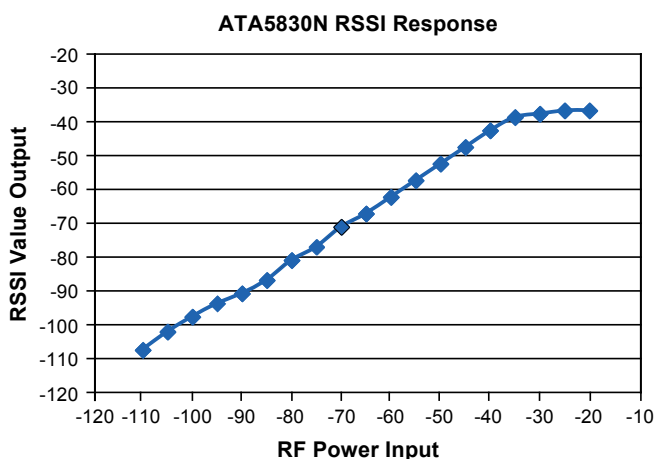


Figure 3. Sample Data Collected Using Flash Application

Summary and Results

This article highlights the capability and flexibility of the ATA5830N radio transceiver. By generating a program that runs in the Flash memory space, and leveraging the available

ROM-based functions, end users can customize the chip for their individual needs. The described application, for example, reports measured RSSI values every 50ms using standard USART timing. This tool would be useful in RF environment analysis such as choosing an operating band or frequency. While not quite as easy to program as a commercial spectrum analyzer, the tool does provide a means to test the RF energy content of a specific frequency and measurement bandwidth.

Perhaps the most useful application for this tool is the optimization of external components for an automotive car access system using the ATA5780N or ATA5830N device. Traditional fine-tuning of a receiver requires a method to extract received data or RSSI, and is typically done with an external microcontroller. Using this application, the external microcontroller and its programming can be eliminated—all that is needed is a PC.

External References

The software and documentation, incl. an application note with even greater detail for this application, is available by contacting automotive@atmel.com.

The Flash software program was generated using Atmel Studio 6, IAR Embedded Workbench for AVR and JTAGICE mkII.