CML Microcircuits

COMMUNICATION SEMICONDUCTORS

Application Note

Simple Data Communications with the CMX469A

Publication: AN/WData/469AFW/1 September 2006

1 Introduction

The CMX469A FFSK/MSK[1] modem IC provides designers with reliable and spectrally efficient data communications. The simplicity of this general-purpose modem, coupled with its flexible data rates (1200bps/2400bps/4800bps), has propelled it into many designs all over the world.

One of the most popular applications of the CMX469A is the wireless transfer of information over voice-grade radios. The CMX469A itself does not require microcontroller configuration, but a host processor is typically used with the device to handle the data flowing in and out of the modem.

The purpose of this document is to illustrate the ease with which the CMX469A and a microcontroller can add data communications to many applications. There are other modem devices in the CML product range, including the FX829 and FX429A but for the purpose of this application note the CMX469A is an ideal choice.

The CMX469A datasheet should be consulted while reviewing this document.

Firmware
Also available from the CML website [www.cmlmicro.com/Products/ . . Application Notes section] for use with this document, is the firmware listing referred-to in the text: AN_469AFW_x.txt.

[1] FFSK/MSK corresponds to Fast Frequency Shift Keying/Minimum Shift Keying, equivalent terms that refer to a spectrally efficient form of frequency shift keying.

TABLE OF CONTENTS

2 Application Description

The CMX469A is commonly used with voice-grade radios to relay data between remote locations, such as shown in the following illustration:

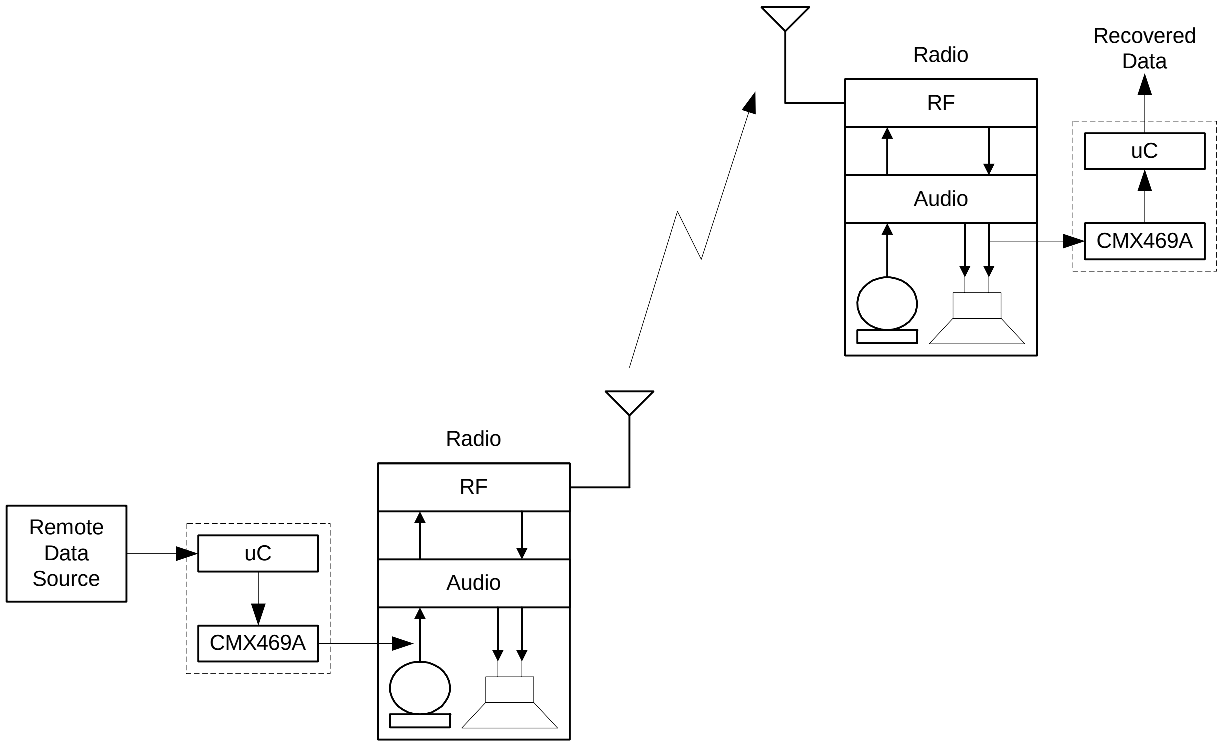

Figure 1: Typical CMX469A Application

There are many types of applications for which this diagram holds true. Any parameter that can be measured (temperature, pressure, level, flow rate, light level, etc.) can also be transmitted to a remote location via a radio. Similarly, control signals can be transmitted back to the remote site to affect changes in those parameters. The amount of data transmitted in both directions is typically small, so a high-speed modem is usually "overkill" for such an application.

The CMX469A FFSK/MSK modem is ideal for this situation. Its modest data rates of 1200bps, 2400bps, and 4800bps, combined with its support for voice-grade radio transmission, allow it to serve in low-cost solutions for remote data collection projects. A complete modem can be cased in a small plastic box and conveniently linked to a low-cost voice radio using the mic and earphone connections.

For this application note, the "remote data source" is the DS1620 digital temperature sensor from Dallas Semiconductor (Maxim). Temperature data from the DS1620 is collected once per second by the host microcontroller, an inexpensive Atmel 8051-class AT89C2051 device. The microcontroller creates a simple data frame consisting of a preamble, "sync word", and temperature data. The data frame is then sent to the CMX469A for transmission. Since many different radios can be used with the CMX469A, no particular radio model was selected for this document. Instead, the transmit path ends with properly formatted CMX469A data signals ready for insertion into the "microphone path" of a typical voice-grade radio.

As would be the case with the output of a radio's discriminator in a finished product, the recovered "audio" (MSK signals) is applied to the receive port of a second CMX469A. When the CMX469A's carrier detect asserts, the receiving microcontroller executes a detection routine that looks for preamble and sync word. If both preamble and sync word are correctly detected, the temperature data is extracted, processed, and sent to a display. An error signal is generated if both preamble and sync word are not correctly detected.

3 CMX469A FFSK/MSK Wireless Modem

The CMX469A FFSK/MSK wireless modem is a flexible device that offers the following functions:

- Spectrally efficient MSK modulation/demodulation
- 1200bps, 2400bps, or 4800bps data rates
- Pin selectable crystal inputs (1.008MHz or 4.032MHz)
- Tx and Rx data clocks provided for microcontroller use
- Carrier Detect circuitry with adjustable detection times
- Vdd=2.7V – 5.5V

The CMX469A is configured for 1200bps in this application by applying a logic high to its 1200/2400 BAUD pin.

The CMX469A is a synchronous modem, so no start and stop bits are required for each data character. The lack of start and stop bits requires that other steps be taken to ensure successful data transfer. Transmit data and receive data clock signals are provided to the host processor from the CMX469A to ensure that data is presented to, and taken from, the CMX469A at the appropriate time. A sixteen-bit preamble of alternating ones and zeros (0xAAAA) is transmitted before any data to ensure the receiving modem can properly train on the incoming signal. A sync word is also transmitted to allow the receiving modem to establish frame synchronization. Both the preamble and sync word are determined by the host processor firmware; the CMX469A does not autonomously generate these sequences.

3.1 CMX469A Interconnection to Radio Module

Radio modules commonly provide a connector to allow interconnection to other equipment. A typical connection port of a radio module is provided below:

Pin Number	Pin Function
1	Audio Input
2	Audio Output
3	PTT (Push To Talk)
4	GND
5	Vout (12VDC)
6	Carrier Detect
7	N/C
8	Switch
9	Speaker (PA output)

Table 1: Typical Radio Module DB-9 Connector Pinout

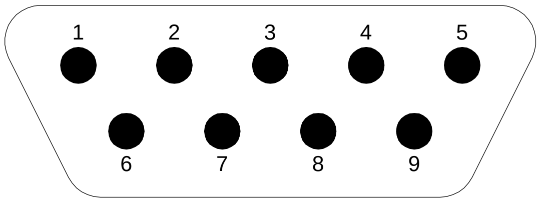

Figure 2: Typical Radio Module DB-9 Connector

Using the above DB-9 pinout as an example, the CMX469A modem circuit could connect to the radio module using the following pins:

Radio Module DB-9 Connector		**Possible CMX469A Circuit Connections**	
Pin Number	Pin Name	Connection Node	Notes
1	Audio Input	CMX469A "Tx Signal O/P"	Typical Tx level out of CMX469A is 775mVrms with V_{DD}=5.0V (465mVrms with V_{DD}=3.0V). Verify that this level is suitable for the selected radio module and scale as necessary with an external amplifier.
2	Audio Output	CMX469A "Rx Signal I/P"	Acceptable Rx input levels for the CMX469A are 100-1000mVrms, and optimal BER is achieved with Rx input level of approximately 230mVrms. See Figure 6 of the CMX469A datasheet for more information on variation of BER with Rx input level. The designer must consider the audio output level from the radio module, as drive levels suitable for speaker reproduction are usually too high for the CMX469A.
3	PTT (Push To Talk)	Microcontroller "Tx Enable" pin	Microcontroller can activate this line when data transmission is required.
4	GND	GND	
5	Vout (12VDC)	Input to voltage regulator	Ensure that DC voltage provided by radio module is within input range of chosen voltage regulator in modem circuit.
6	Carrier Detect	Microcontroller "CD" pin	Depending on its response time and noise immunity, this signal may be preferred to start the receive process. If this is done, the CMX469A carrier detect function will no longer be required and the CMX469A Rx circuit can be powered down when not in use.

4 Firmware Description

The state flow diagram of this simple application is illustrated in the following figure:

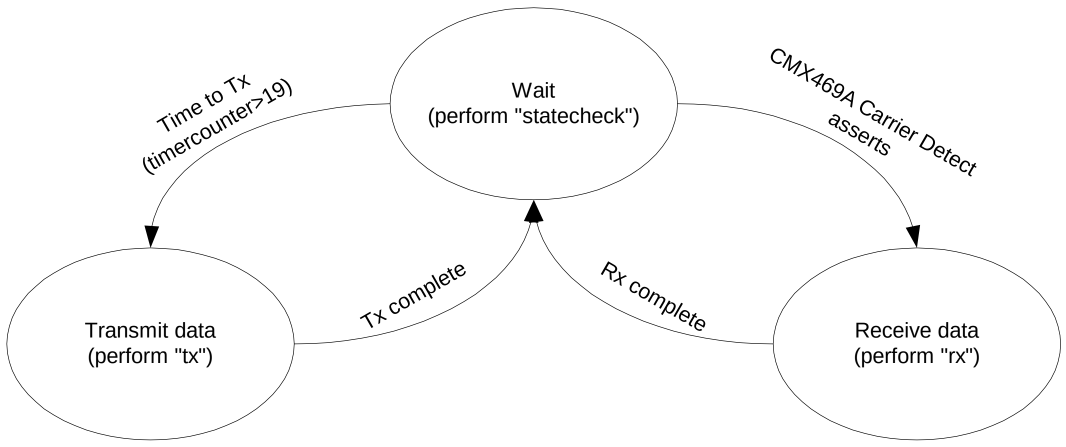

Figure 3: State Flow Diagram

The basic premise of this application is that of a remote station (sensor/controller) that transmits its data, at regular time intervals, to a collecting station. Occasionally the collecting station may need to send a control signal back to the remote station (e.g. to close a valve or turn on a pump), so the capability to send and receive data is required. Therefore, the remote station must also receive data transmissions.

As can been seen from the above figure, the microcontroller in this scenario spends most of its time waiting for either:

- The delay between transmissions to end, indicating the need to transmit data, or…
- Carrier detect to assert, indicating an incoming data transmission.

Since many off-the-shelf voice-grade radios are half-duplex (i.e. push to talk, release to listen), the data communications used in this scenario are also half-duplex.

The code for this project was written in C and compiled using the Keil C compiler (V7.5). The code, as written, occupies 832 bytes of the AT89C2051's 2048 bytes of program memory space. Much of that space, however, is occupied by routines that likely would not be used in an end product, such as the display driver routines. When the display driver and "parse_temperature" routines and corresponding variables are removed, the required code space is reduced to approximately 497 bytes, which would leave 1551 bytes of program space for an actual application. Other microcontrollers with additional program memory space can be used with the CMX469A if required.

4.1 Microcontroller Initialization

The "config_8051" function, called from the main routine, is responsible for configuration of the host processor, the AT89C2051 microcontroller.

Within this function, logic ones are written to port pins that need to be used as inputs, such as the pin used for the "CARRIERDETECT" function. The two external interrupts provided by the AT89C2051 are used in falling-edge triggered mode to detect transitions on the CMX469A's TX SYNC O/P and RX SYNC O/P pins.

The AT89C2051 provides two 16-bit timers that can be used as timers or event counters. In this application, Timer1 and Timer0 are both configured as 16-bit timers (Mode 1). Timer1 is used for the 5ms delay at the end of the "rx" routine, and Timer0 is used for the one second delay associated with the "tx" routine. The high and low Timer0 special function registers (TH0 and TL0) are loaded with 0x4C00 to create a timer overflow after 50ms. The 11.0592MHz crystal used with the AT89C2051, coupled with its 12 clock cycles per instruction cycle, translates to (46080 counts x 1.08us/count ≈ 50ms).

The AT89C2051 provides five different interrupt sources, four of which are used in this application. (The serial port and its interrupt are not used in this scenario.) The following interrupts are enabled within “config_8051”:

- Timer1
- EX1 (external interrupt 1, used to detect transitions on CMX469A RX SYNC O/P pin)
- Timer0
- EX0 (external interrupt 0, used to detect transitions on CMX469A TX SYNC O/P pin)

The “config_8051” function concludes by turning on Timer0, which initiates the one-second delay until data transmission.

4.2 Main Routine

The firmware begins by initializing the LED display, temperature sensor, and the microcontroller. Once the initializations are complete, an infinite loop is entered and the microcontroller continually checks to see if either data transmission or data reception is required.

The “statecheck” function is called within the main routine’s infinite loop to determine if data transmission or data reception is required. The variable returned from “statecheck” is passed to a switch expression that calls the required transmit (“tx”) or receive (“rx”) function.

4.3 Data Transmission

The general flow of the transmit routine is shown in the following figure:

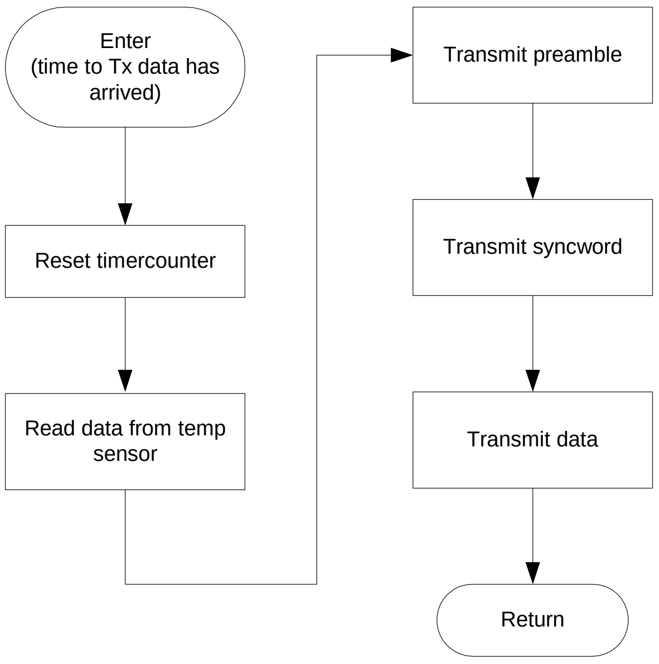

Figure 4: Transmit Operation (“tx” Function)

Data is collected from the temperature sensor once per second. To accomplish this delay, Timer0 of the AT89C2051 microcontroller is configured to issue interrupts once every 50ms. A “timercounter” variable is incremented each time Timer0 overflows, and after 20 such interrupts, one second has elapsed and the time for transmission has arrived.

When the one second delay has elapsed the “timercounter” variable is incremented past a preset value. The “statecheck” function detects the excessive “timercounter” value and determines that data transmission is required; control is then passed to the “tx” function.

The “tx” function begins by resetting the “timercounter” variable to facilitate the next one second delay. Next, nine bits of data corresponding to the Celsius temperature are read out of the DS1620 temperature sensor, least significant bit first. The microcontroller then creates a simple data frame as follows:

Preamble 16 bits (0xAAAA)	Sync Word 16 bits (0xCB23)	Data 9 bits

Figure 5: Data Frame Structure

(The CMX469A is capable of transmitting any amount of data required for the application, but in the interest of simplicity, only a single 9-bit data payload is used in this scenario. The designer is encouraged to experiment with sync word values and data lengths as needed for their particular design requirements.). In addition as there is no means of bit stuffing (the process of inserting temporary bits in transmit to avoid replication of data and increase mark/space transitions) or FEC (Forward Error Correction) present to increase data integrity there is scope for further research.

The preamble allows the receiving modem to develop a sense of signal timing and signal level, both of which are required to allow distinction between a “one” and a “zero”. The sync word is used to allow the receiving modem to establish frame synchronization.

The data frame is passed to the CMX469A for transmission, most significant bit first. After the complete frame has been transmitted, control shifts back to the infinite loop within the main routine.

4.4 Data Reception

The general flow of the receive routine (“rx”) is depicted in the following figure:

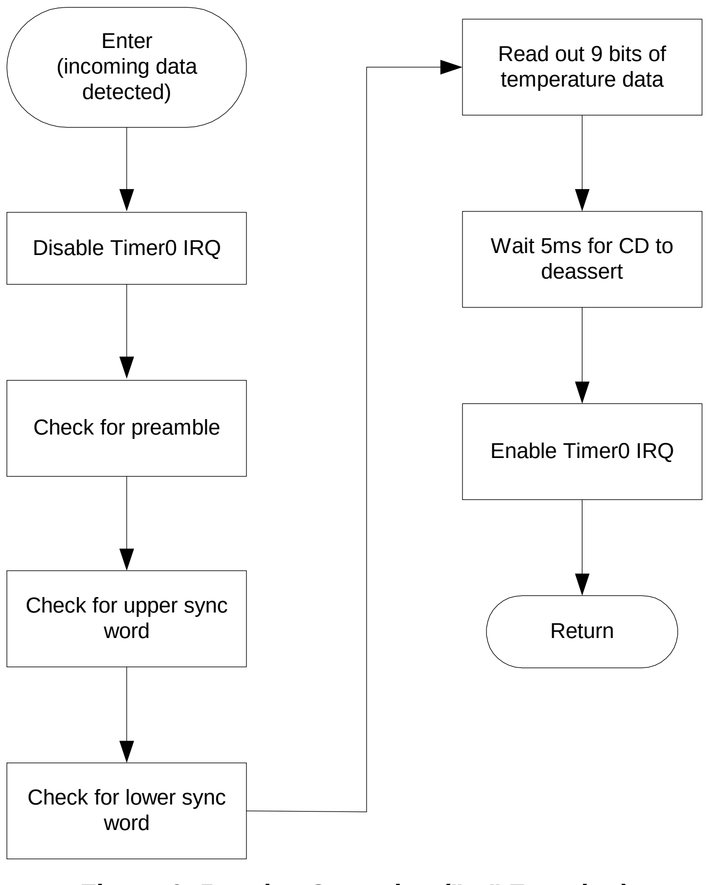
Enter (incoming data detected)
Disable Timer0 IRQ
Check for preamble
Check for upper sync word
Check for lower sync word
Read out 9 bits of temperature data
Wait 5ms for CD to deassert
Enable Timer0 IRQ
Return

Figure 6: Receive Operation ("rx" Function)

When an incoming data transmission is detected, the CMX469A CARRIER DETECT O/P pin will transition to a logic high, and this causes the “CARRIERDETECT” variable to be set to 1. The “statecheck” function will detect the “CARRIERDETECT” variable change and call the “rx” function to extract the data from the incoming transmission.

The “rx” function begins by disabling the Timer0 IRQ to ensure that the receive routine flows without interruption from the transmit routine’s one second timer. The “checkfor” function is then called to search the received bit stream for the preamble (0xAA), upper sync word (0xCB), and the lower sync word (0x23). If any of these components are not detected, the “checkfor” function generates an error message and prevents the erroneous characters from disturbing the display.

Once the preamble and sync words are detected, the actual temperature data is extracted from the CMX469A. The host processor formats the temperature data and transmits it to the LCD display driver IC for display.

A five millisecond delay is then executed to ensure that the received signal has ended. This delay helps ensure that the CMX469A CARRIER DETECT O/P pin has deasserted, thereby preventing an incorrect “rx” function call when the “statecheck” routine is next performed.

5 Schematic

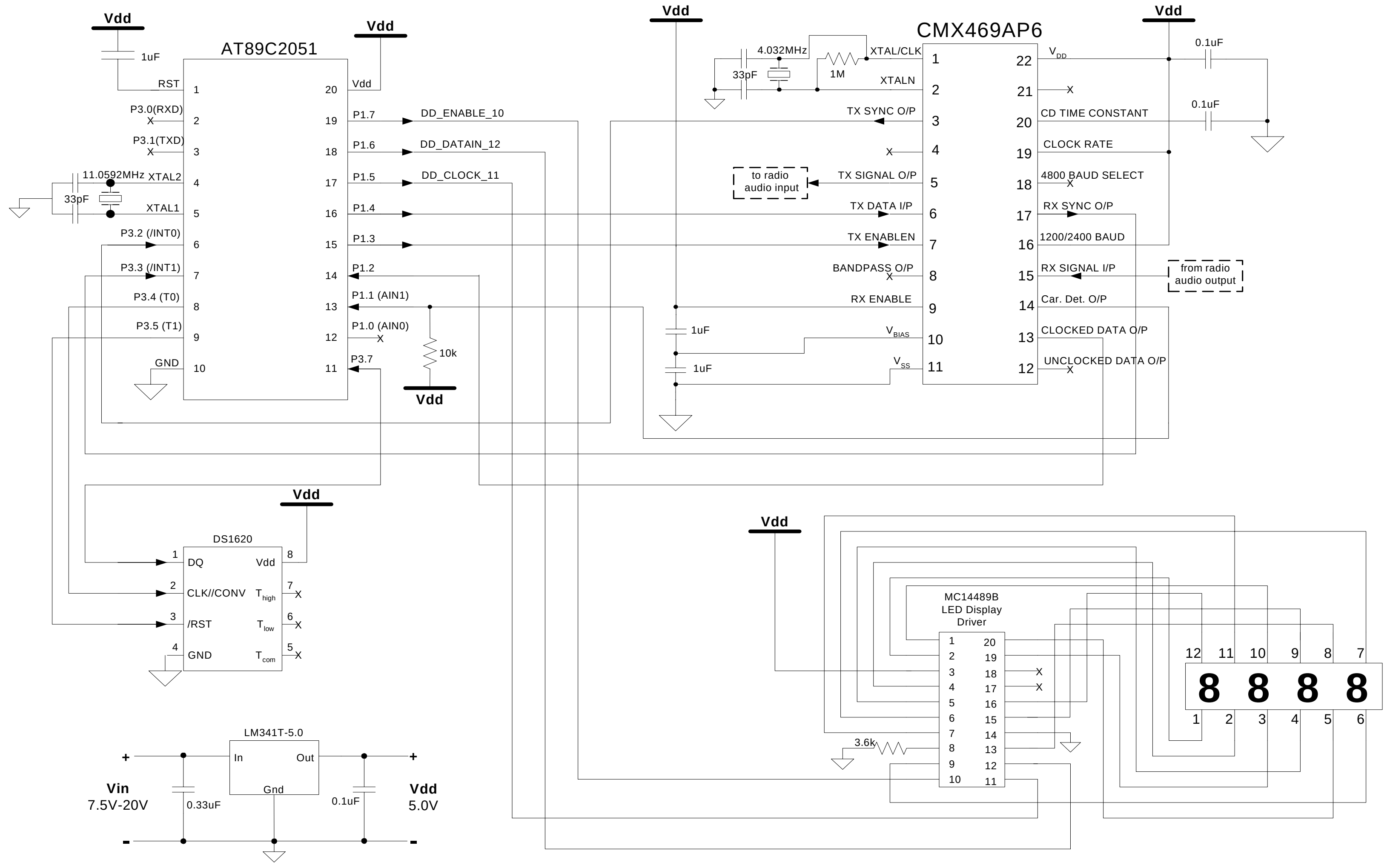

Figure 7: CMX469A Circuit Interconnections

6 Possible Improvements

The following sections address potential improvements to the circuitry depicted in this application note.

6.1 RS-232 Data Communications

Some applications may interface to a data source that conforms to the RS-232 standard. The CMX469A can be used to transfer RS-232 data, but additional steps are required to do this. For additional information on this topic, please review the following application note available from the CML website: "Using an FX469 FFSK Synchronous Modem with an Asynchronous Data I/O". With regards to that application note, the presence of the microcontroller should eliminate the need for the synchronous-to-asynchronous converter IC, as the microcontroller controls the pacing of data in the application. The RS-232 transceiver IC will still be required for voltage translation, however.

6.2 Faster Microcontroller Operation

The slow data rate used in this application, coupled with the simple structure of the routines, allowed the use of:

- A classic 8051 microcontroller (12 clock cycles per instruction cycle).
- A slower crystal (11.0592MHz) for the microcontroller.

A faster data rate and/or a more rigorous application may require the use of a processor that can perform instructions faster than the AT89C2051 used here. Options for this situation include:

- Use of a higher frequency microcontroller crystal. The AT89C2051 can operate with crystal frequencies up to 24MHz.
- Use of a microcontroller that requires fewer clock cycles per instruction cycle, such as a "single-cycle" microcontroller.

6.3 Faster Data Rate

The CMX469A is capable of operation at 1200bps, 2400bps, or 4800bps. The choice of CMX469A data rate is controlled through:

- Crystal frequency selection.
- Logic level on CMX469A 1200/2400 BAUD pin.
- Logic level on CMX469A 4800 BAUD SELECT pin.

Both 1200bps and 2400bps operation can typically be achieved by connecting the CMX469A Tx and Rx ports to the radio's audio path. Operation at 4800bps via a typical wireless audio path connection is impractical, however, due to the filtering in the voice path (typically 300-3000Hz). Access to the radio's modulator and discriminator is required for CMX469A operation at 4800bps. The designer should check to see if there are any license restrictions in place locally that may disallow the use of data over audio radio channels.

6.4 Lower Power Consumption

The core circuit of CMX469A modem, AT89C2051 microcontroller, and DS1620 temperature sensor consume approximately 9.6mA during normal operation with V_{DD}=5.0V. This current consumption results from:

- Transmission once per second.
- Continuous receive mode.
- No idle mode used by the microcontroller.
- Room temperature operation.

The V_{DD}=5.0V value was chosen to satisfy the LED display used in this project, but in an application where the LED display isn't required, the AT89C2051 and CMX469A are both capable of operation down to V_{DD}=2.7V. When the power supply voltage is reduced to V_{DD}=3.0V, the core circuit current consumption is reduced to approximately 4.2mA (previous assumptions are used).

The use of a polled system will dramatically reduce power consumption. In this approach, the remote sensor circuit is normally powered down to the maximum extent possible. When the collecting station requires data from the remote sensor circuit, it will transmit a data request to the remote station. The receive circuitry detects this incoming signal, enables its remote sensor circuit, and interprets and acts on the incoming request.

Powerdown mode will reduce power consumption of this circuit from milliamps to microamps. For example, the powerdown current consumption of the CMX469A, microcontroller and temperature sensor is approximately 321µA. If low-current consumption solid-state switches are used to gate the power supply to the CMX469A and temperature sensor, the circuit current consumption can be reduced to approximately 20µA (V_{DD}=3.0V). One caveat of this approach is the "warm-up time" of the AT89C2051 microcontroller's crystal oscillator circuit. Once the microcontroller exits from powerdown mode, the crystal oscillator must start and generate stable oscillations before the microcontroller will perform any instructions. This warm-up time, which can easily be on the order of 10ms, may cause many bits of preamble to be received before the microcontroller and CMX469A are properly enabled. This can be resolved by extending the length of preamble so that the microcontroller and CMX469A are fully operational before the entire preamble has been received. (The CMX469A and microcontroller requires 16 bits of preamble in order to train properly.)

The carrier detection function of the CMX469A is disabled when its receive circuit is powered down, so to ensure detection of incoming signals, the CMX469A receive function is continuously enabled in this application. The CMX469A consumes approximately 3.6mA in receive mode and 1.6mA in transmit mode (V_{DD}=5.0V), so overall power consumption can be improved by reducing the amount of time spent by the CMX469A in receive mode.

Many radio modules provide a carrier detect function that is accessible by external circuitry, such as a carrier detect signal that is brought out on a radio module's DB-9 connector. This signal can be sensed by the microcontroller and used to initiate the reception process, and this will allow the CMX469A receive circuit to be powered down when not in use. The designer should consider the response time and noise immunity of such a signal to ensure it will meet the system requirements.

Another possible solution would be to develop a carrier detection circuit, using external components, that notified the microcontroller when the received signal exceeded a preset threshold. The microcontroller could then enable the CMX469A receive circuit and begin the receive process. Please note that noise immunity will have to be considered with such an approach, and any such circuit is application dependent.

The following additional options can be considered if lower power consumption is required:

- Operate at lower V_{DD}.
- Implement "idle mode" in the microcontroller. One potential approach would involve invoking idle mode for a short amount of time if the "statecheck" function determined that neither transmit nor receive routines were required. Once the delay had elapsed, the microcontroller could resume normal powered operations and perform "statecheck" again.
- Transmit data to collecting station less often, and power down CMX469A transmit section when not in use. (CMX469A transmit section is powered down in this application when not in use.)

6.5 Carrier Detect Qualification

The CMX469A carrier detect output is checked by the microcontroller during the "statecheck" function. It is conceivable that noise could cause the CMX469A carrier detect output to temporarily transition to a logic high as the "statecheck" function checks its level. In other words, noise could falsely cause the firmware to enter the receive routine.

There are two measures used in this application to limit such noise falsing. The first involves the selection of the capacitor on the CMX469A "CD Time Constant" pin. This capacitor sets a time constant internal to the CMX469A that is used to determine what the logic condition of the CARRIER DETECT O/P pin should be. A larger value of capacitance on the CMX469A CD TIME CONSTANT pin means that the received input signal must be present for a longer period of time before the carrier detect asserts. The downside is that the longer time constant means "real" signal will pass through the CMX469A, before the microcontroller realizes that a valid signal is present, so some of the “real” signal will be missed. A tradeoff exists in the selection of the CMX469A CD TIME CONSTANT capacitor, and the optimal value for any application must be experimentally determined.

The second falsing deterrent is the “checkfor” function. When noise causes the CMX469A carrier detect to assert and launch a receive sequence, the “checkfor” function will monitor the bit stream and look for predetermined values of preamble and sync word. If the expected values aren’t found, an error message is generated and the receive function ends.

It is conceivable that these two techniques may not be adequate for noisy operating environments. The designer may then wish to perform multiple reads of the CMX469A CARRIER DETECT O/P pin, perhaps with a slight delay between each read, and only execute the receive sequence if the pin indicates logic high for multiple reads.

6.6 Test for Valid Data

In most applications, the expected range of data values to be received by the remote terminal is known in advance. For example, a remote pressure gauge may be known to have valid values between 1MPa – 3MPa. In this situation, the designer can insert a test within the receive sequence that only passes valid data to the microcontroller and indicates an error condition if invalid data is received. Such a test could easily be inserted after the “parse _temperature” function call within the “rx” routine.

7 Miscellaneous Components

A number of microcontrollers were considered for this project however the AT89C2051 was chosen. The AT89C2051 is a member of Atmel’s 8051-class product family, it was chosen for this project for its ease of use. The 2kbytes of on-chip FLASH contained in the AT89C2051 provides sufficient program memory for this application.

Noteworthy features of the AT89C2051 include:

- Vdd=2.7V – 6.0V
- Clock speeds up to 24MHz supported
- 15 programmable I/O lines
- Two 16-bit timer/counters
- Five-vector two-level interrupt structure
- Full-duplex serial port
- Precision analog comparator
- On-chip oscillator
- Idle and powerdown modes for reduced current consumption

An 11.0592MHz crystal provides the timing for the microcontroller. The datasheet for the AT89C2051 can be found by visiting Atmel’s website.

As mentioned previously there are many microcontrollers that could be used with the CMX469A for this application. It is hoped that the widespread deployment and familiarity of the 8051 microcontroller family, coupled with the extensively commented C code used in this project, will allow the information in this document to transcend microcontroller families and be useful to many readers.

The DS1620 temperature sensor from Dallas Semiconductor served as the “remote data source” in this scenario. A generic four-digit LED display was used to visually observe the results, and a Motorola MC14489B LED display driver IC served as the interface between the microcontroller and the LED display. Since these components are not expected to serve in an actual application, no further treatment of these devices will be provided in this document.

8 Conclusion

The CMX469A FFSK/MSK wireless modem IC offers reasonable data rates, good spectral efficiency, and its audio tone outputs allow the use of voice-grade radios to accomplish wireless communications. The CMX469A does not require microcontroller configuration, but a microcontroller is typically used to manage data traffic through the modem.

This application note has illustrated one method in which a microcontroller and the CMX469A can work together to perform half-duplex data communications. It is hoped that the information in this document will assist designers in integrating the CMX469A FFSK/MSK wireless modem IC into their designs.

www.cmlmicro.com

For FAQs see: http://www.cmlmicro.com/products/faqs/index.htm

For a full data sheet listing see: http://www.cmlmicro.com/products/datasheets/download.htm

For detailed application notes: http://www.cmlmicro.com/products/applications/index.htm

CML Microcircuits (UK) Ltd COMMUNICATION SEMICONDUCTORS	CML Microcircuits (USA) Inc. COMMUNICATION SEMICONDUCTORS	CML Microcircuits (Singapore) Pte Ltd COMMUNICATION SEMICONDUCTORS	
Tel: +44 (0)1621 875500 Fax: +44 (0)1621 875600 Sales: sales@cmlmicro.com Technical Support: techsupport@cmlmicro.com	Tel: +1 336 744 5050, 800 638 5577 Fax: +1 336 744 5054 Sales: us.sales@cmlmicro.com Technical Support: us.techsupport@cmlmicro.com	Tel: +65 7450426 Fax: +65 7452917 Sales: sg.sales@cmlmicro.com Technical Support: sg.techsupport@cmlmicro.com	Tel: +86 21 63174107 +86 21 63178916 Fax: +86 21 63170243 Sales: cn.sales@cmlmicro.com.cn Technical Support: sg.techsupport@cmlmicro.com