

CC1020 Microcontroller Interfacing

By T. Sundet and A. Rogndalen

Keywords

- *CC1020*
- *Microchip PIC*
- *Example code*
- *Configuration interface*
- *Data interface*
- *Calibration*

Introduction

The CC1020 RF transceiver is very easy to interface with a microcontroller.

The chip is configured using a four-wire bus, comprising of PCLK, PDI, PDO and PSEL signals.

Data interfacing is done via the DCLK and DIO pins.

Data decision and synchronisation is done by the CC1020, easing the workload of the microcontroller considerably. Using the synchronous Manchester or NRZ modes,

the CC1020 supplies a synchronous clock, which can be used for triggering interrupts in the microcontroller, or for interfacing with a synchronous serial interface.

This application note covers interfacing the CC1020 to microcontrollers. Information is given about both the configuration and data interfaces of the CC1020.

Microcontrollers in general

A vast variety of microcontrollers (MCUs) exist today, from small 4-bit controllers up to powerful 32-bit controllers with processing power rivalling modern PCs. Digital signal processors (DSPs) are also used as controllers in some applications.

The CC1020 can be easily interfaced with any microcontroller on the market. The only requirement is to have enough I/O pins. To configure the CC1020, four I/O pins are required (one input and three output pins). The pins connected to PDI, PDO and PCLK can be shared with other circuitry, provided these circuits are not active when the configuration interface is active. The PSEL signal must be driven by pin dedicated only to interfacing the CC1020. For the data interface, two I/O pins are required, one bi-directional for DIO and one input for DCLK. The pin used to interface with DCLK should be able to generate an interrupt on signal edges.

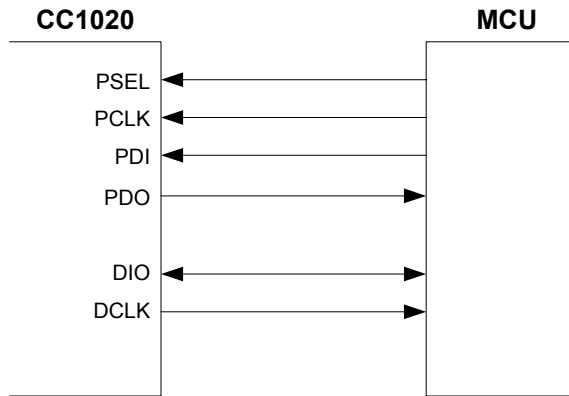


Figure 1 CC1020 – MCU hardware interface

In power-down mode, the CC1020 pins assume the following states:

Pin	Pin state	Note
PDI	Input	
PDO	Output	
PCLK	Input	
PSEL	Input with internal pull-up resistor	Should be driven high or high-impedance to minimize power consumption
DIO	Input	
DCLK	High-impedance output	

Figure 2: CC1020 pins in power-down mode

Configuration interface

The CC1020 is configured using the PCLK, PDI, PDO and PSEL signals. This is discussed in detail in the CC1020 data sheet. The configuration registers are also readable, so that the user can verify settings and read out status bits.

Using general-purpose I/O pins to handle an interface in this way is often called “bit banging”. This approach is very flexible, as the user is free to use any I/O pins on the microcontroller, but can be slow when working with a slow microcontroller.

The alternative is to use a synchronous serial interface to interface with the CC1020. An SPI master interface or some types of USART (Universal Synchronous/Asynchronous Receiver/Transmitter) are suitable. A free general I/O pin must be used to interface with the PSEL pin of the CC1020. Please note that some microcontroller manufacturers use different terminologies when naming the signals. MOSI stands for master out, slave in, while MISO stands for master in, slave out. The other standard SPI signals are SCK (Serial Clock) and SS (Slave select).

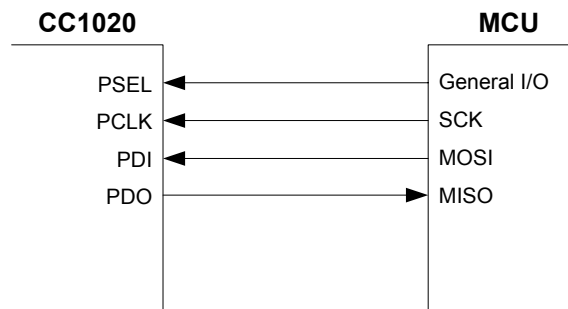


Figure 3: Suggested SPI configuration interfacing

A USART or other synchronous serial interface can be suitable for use as well, study the datasheet for the microcontroller to see if it can be configured to be compatible with the CC1020.

The biggest advantage of using a hardware interface is that it can be much faster; in many cases the configuration can be performed with a clock rate of several MHz. This is especially important in frequency hopping applications, where configuration must occur in the blanking interval, which should be as short as possible.

Configuration procedure

Chipcon recommends resetting the CC1020 (by clearing the RESET_N bit in the MAIN register) when the chip is first powered up. All registers that need to be configured should then be programmed (those which differ from their default values). Registers can be programmed freely in any order. The CC1020 should then be calibrated in both RX and TX mode. After this is complete, the CC1020 is ready for use. See the datasheet for detailed procedure flowcharts.

SmartRF[®] Studio and the example code written by Chipcon assumes that frequency register A is used for the RX frequency, and frequency register B is used for the TX frequency. This makes it easy to switch between RX and TX mode by programming the MAIN, ANALOG (and INTERFACE if squelch enabled) registers.

Data interface

When using the Synchronous Manchester or the Synchronous NRZ mode, the DCLK pin on the CC1020 should be connected to an MCU input pin that can generate an interrupt in the MCU. DIO should be connected to a bi-directional MCU I/O pin.

In TX mode, the interrupt should be triggered on the falling edge of DCLK. When the interrupt occurs, write the next bit to be transmitted to the I/O pin. In RX mode, the interrupt should be triggered on the rising edge of DCLK. When the interrupt occurs, read the data from the I/O pin.

Note that data transferred to/from the MCU is *always* NRZ coded, regardless of whether Synchronous NRZ or Synchronous Manchester mode is selected. The mode setting only affects the signal modulated onto the RF carrier. The Manchester encoding/decoding is performed by the CC1020.

A clock signal is output on DCLK as long as DCLK squelch is disabled on the CC1020. If transmissions are intermittent, the DCLK squelch function should be used to determine whether valid data is received. The squelch must be disabled in TX-mode in order NOT to mute the DCLK.

A preamble of alternate ones and zeros must be used regardless of which mode the CC1020 is in. See the datasheet for details regarding minimum length of the preamble.

Microchip PIC16 example

This example is written for the PIC16F876 microcontroller, using the RF Modem hardware. See [1] for more information about the hardware.

cc1020pic.c provides a library of routines for configuring the CC1020. cc1020.h contains definitions for all of the CC1020's registers. There are two versions of the low-level configuration routines, one using general-purpose I/O and one using the SPI interface built into the PIC. If the symbol 'SPI' is defined, the SPI versions are used, if not, the general-purpose I/O routines are used.

A short description of the routines in the library follows:

Function/routine name	Short description
<i>ConfigureCC1020()</i>	Takes an array and a count value as input parameters. The array should be filled with values of the <i>short int</i> type (16 bits) consisting of a valid CC1020 register address in the 7 highest bits, and the register value in the lowest 8 bits. The count value should be set equal to the number of values stored in the array.
<i>WriteToCC1020Register()</i>	Takes an address and a value as parameters, and writes to a single CC1020 register.
<i>ReadFromCC1020Register()</i>	Takes an address of a CC1020 register as parameter, and returns the value of that register.
<i>CalibrateCC1020()</i>	Initiates a calibration, and waits for it to finish. It also turns off the PA before initiating calibration, which prevents spurious transmission from occurring in TX mode. The PA is turned back on once calibration is completed.
<i>ResetCC1020()</i>	Resets the CC1020, clearing all registers.
<i>SetupCC1020PD()</i>	Puts the CC1020 into power down mode. Use <i>WakeUpCC1020ToRX</i> or <i>WakeupCC1020ToTX</i> to wake up from power down.
<i>SetupCC1020TX()/</i> <i>SetupCC1020RX()</i>	Puts the CC1020 into RX mode (from TX). When switching to RX from PD, use <i>WakeupC1020ToRX</i> first.
	Puts the CC1020 into TX mode (from RX). When switching to TX from PD, use <i>WakeupC1020ToTX</i> first.
<i>WakeUpCC1020ToTX()/</i> <i>WakeUpCC1020ToRX()</i>	When the CC1020 is awakened from PD mode, the power consumption can be minimised by only turning on the crystal oscillator at first, then waiting until it has stabilised (2-5ms) before turning the rest of the transceiver on. This is implemented in <i>WakeUpCC1020ToTX/RX</i> .

Table 1: Summary of software functions/routines

When the system is initially turned on, the CC1020 should first be reset. The registers can then be programmed to the settings used in the system. The CC1020 should be calibrated in both RX and TX modes before system operation takes place. For details, see the CC1020 data sheet.

All source code can be downloaded from the Chipcon website (www.chipcon.com).

References

- [1] Chipcon: *Application Note AN015 RF Modem Reference Design*. www.chipcon.com
- [2] Karl H. Torvmark: "Short Range Wireless Design". *Embedded Systems Programming*, October 2002 p. 20.

This application note is written by the staff of Chipcon to the courtesy of our customers. Chipcon is a world-wide distributor of integrated radio transceiver chips. For further information on the products from Chipcon please contact us or visit our web site.

Address Information

Web site: <http://www.chipcon.com>
E-mail: wireless@chipcon.com

Technical Support Email: support@chipcon.com
Technical Support Hotline: +47 22 95 85 45

Headquarters:

Chipcon AS
Gaustadalléen 21
NO-0349 Oslo
NORWAY

Tel: +47 22 95 85 44
Fax: +47 22 95 85 46
E-mail: wireless@chipcon.com

US Office:

Chipcon Inc.
19925 Stevens Creek Blvd.
Cupertino, CA 95014-2358
USA

Tel: +1 408 973 7845
Fax: +1 408 973 7257
Email: USSales@chipcon.com

Sales Office Germany:

Chipcon AS
Riedberghof 3
D-74379 Ingersheim
GERMANY

Tel: +49 7142 9156815
Fax: +49 7142 9156818
Email: Germanysales@chipcon.com

Chipcon AS is a ISO 9001:2000 certified company



Disclaimer

Chipcon AS believes the information contained herein is correct and accurate at the time of this printing. However, Chipcon AS reserves the right to make changes to this product without notice. Chipcon AS does not assume any responsibility for the use of the described product.; neither does it convey any license under its patent rights, or the rights of others. The latest updates are available at the Chipcon website or by contacting Chipcon directly.

As far as possible, major changes of product specifications and functionality, will be stated in product specific Errata Notes published at the Chipcon website. Customers are encouraged to sign up to the Developer's Newsletter for the most recent updates on products and support tools.

When a product is discontinued this will be done according to Chipcon's procedure for obsolete products as described in Chipcon's Quality Manual. This includes informing about last-time-buy options. The Quality Manual can be downloaded from Chipcon's website.

Trademarks

SmartRF[®] is a registered trademark of Chipcon AS. *SmartRF*[®] is Chipcon's RF technology platform with RF library cells, modules and design expertise. Based on *SmartRF*[®] technology Chipcon develops standard component RF circuits as well as full custom ASICs based on customer requirements and this technology.

All other trademarks, registered trademarks and product names are the sole property of their respective owners.

© 2003, Chipcon AS. All rights reserved.