

## **User Manual**

# **CC400DB Demonstration Board**

Rev. 2.0

## Table of contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>3</b>
<b>2</b>	<b>CC400DB DEMONSTRATION BOARD.....</b>	<b>4</b>
2.1	BEFORE STARTING .....	4
2.2	DEMONSTRATOR OPERATION.....	5
2.3	SYSTEM PARAMETERS .....	5
2.4	CONFIGURATION REGISTER DATA .....	6
2.5	MICRO-CONTROLLER EEPROM MAPPING.....	7
2.6	TROUBLE-SHOOTING.....	8
<b>3</b>	<b>DEMONSTRATION BOARD SOFTWARE DESCRIPTION.....</b>	<b>9</b>
3.1	SLEEP MODE .....	9
3.2	POLLING MODE .....	9
3.3	DATA PACKET DESCRIPTION .....	9
3.4	DATA TRANSMISSION PROTOCOL .....	10
3.5	DATA RECEPTION.....	11
3.6	DATA DECODER ALGORITHM .....	11
<b>4</b>	<b>DEMONSTRATION BOARD HARDWARE DESCRIPTION .....</b>	<b>12</b>
4.1	RF-SECTION.....	12
4.1.1	<i>PLL loop filter.....</i>	<i>12</i>
4.1.2	<i>IF filter.....</i>	<i>12</i>
4.1.3	<i>The modulation input/output.....</i>	<i>12</i>
4.1.4	<i>LNA/PA matching .....</i>	<i>13</i>
4.1.5	<i>Voltage Controlled Oscillator (VCO).....</i>	<i>13</i>
4.1.6	<i>Crystal oscillator .....</i>	<i>13</i>
4.1.7	<i>Antenna .....</i>	<i>13</i>
4.2	CONTROLLER SECTION.....	14
4.2.1	<i>RC oscillator .....</i>	<i>14</i>
4.2.2	<i>Micro-controller I/O .....</i>	<i>14</i>
4.2.3	<i>In-system-programming.....</i>	<i>14</i>
4.3	LAYOUT SKETCHES .....	16
4.4	CIRCUIT DIAGRAM .....	17
4.5	BILL OF MATERIALS.....	18
<b>5</b>	<b>USING THE DEMONSTRATOR BOARD AS A PROTOTYPE MODULE... </b>	<b>19</b>

## 1 Introduction

The CC400 single chip RF transceiver includes many features and great flexibility, which makes the chip suitable for a very large number of applications and system requirements. The CC400DB Demonstration Board is designed to demonstrate how the CC400 chip can be used in a simple two-way remote control. Full documentation is provided for this module, including schematic, layout and micro-controller code. This design will be a starting point and make it very easy for the users to evaluate transceiver performance, and in a short time develop their own applications based on this reference design.

The kit includes two CC400DB Demonstration Boards with a complete CC400 transceiver, micro-controller, integrated PCB antenna and battery. The Demonstration Boards are pre-programmed to serve as a simple demonstrator. By pushing the “ON” button, the device will be awaked. Both modules should be awaked before a transmission takes place. A transmission is initiated by pushing the “TX” button. A blinking sequence of approximately one-second duration is then transmitted and shown on the receivers LED. The same “message” is returned from the receiver to the initial transmitter and shown on its LED as an “acknowledge”. The modules will enter sleep mode after being left without operation for one minute.

Technical features:

- 0dBm RF power
- Up to 150 meter range (depending on antenna orientation)
- Small integrated shorted-stub antenna
- <1uA power-down current in sleep mode
- 3.6V Lithium battery supply (~1000mAh)

From SmartRF<sup>®</sup> Studio it is possible to reconfigure the CC400DB system parameters by inserting the Demonstration Board into the CC400EB Evaluation Board edge connector. It is also possible to upgrade the Demonstration Board with new micro-controller program code from SmartRF<sup>®</sup> Studio.

The CC400DB is also designed to work together with the Atmel AVR micro-controller STK500 development board, the in-system programmer ATAVRISP or similar. At the rear side of the board there is a 10 pin connector foot-print which fits a 10-pin ISP cable.

This user manual describes how to use the Demonstration Board and how to develop new applications. You will also find a detailed description of the circuit board and the micro-controller code.

All available documentation can be downloaded from our web site. That includes:

- CC400DB User Manual (this document)
- CC400DB micro-controller code and configuration data
- CC400 data sheet

## 2 CC400DB Demonstration Board

### 2.1 Before starting

The CC400DB runs on a 3.6V Lithium battery. If the battery is not mounted and soldered to the board, insert the battery at the rear side of the board. Observe the LED. If the LED is turned on continuously, push the “ON” button. For proper operation the LED shall be pulsing with a period of approximately one second. If not, remove the battery momentarily and reconnect. The battery terminals should be soldered for proper operation.

**Important:** Be sure not to short-circuit the battery terminals during soldering or at any point of use.

**Before using the CC400DB:** The use of radio transceivers is regulated by international and national rules. Before transmitting a RF signal, please contact your local telecommunication authorities to check if you are licensed to operate the transceiver.

## 2.2 Demonstrator operation

The CC400DB demonstrates a very simple two-way remote control system.

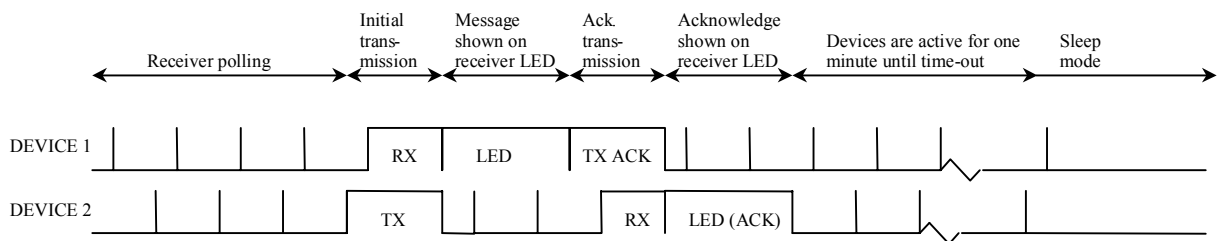
To “wake” the module from power-down mode, simply press the “ON” button for a short time. The LED will then start blinking with a period of approximately 1 second. This is the “heart-beat” that shows the module is operating. During the short time the LED is lit, the module is in active receive mode, and listens for a data packet. If a data packet is not found, it goes to a “rest” mode for approximately 1 second. Hence, the module listens for a data packet every second. This polled receiver algorithm is implemented in order to save battery power.

Both modules should be awaked before a successful transmission can take place. A transmission is initiated by pushing the “TX” button. A data packet of approximately 1.5 second duration is then transmitted. The data packet consists of a long preamble and a short data field. The pre-amble is longer than 1 second to ensure that the transmission intercepts with the polled receiver.

The data packet transmitted consists of a blinking sequence of approximately 2 seconds duration, and will be shown on the receiver’s LED if the data packet was received free of errors. The same “message” with an acknowledgement flag is then returned from the receiver to the initial transmitter. Again, the blinking sequence will be shown on the LED as a receipt acknowledge.

The modules will enter sleep mode after being left without operation for about one minute. The CC400 and the micro-controller are then in power-down mode, ensuring minimum power consumption.

The timing diagram below illustrates a packet transmission sequence when the transmission is initiated from device number 2.



## 2.3 System parameters

The system parameters used to generate the configuration data for the CC400DB is shown below. This is a printout from the SmartRF® Studio software.

System parameters:  
X-tal frequency: 12.000000 MHz  
X-tal accuracy: +/- 50 ppm

RF frequency: 433.920 MHz  
IF stage: 200 kHz  
Frequency separation: 40 kHz  
Data rate: 1.2 kbps  
Power amplifier class: Class B  
RF output power: 0 dBm  
Mode: TX  
Receiver mode: Optimum sensitivity  
LOCK indicator: Continuous

## 2.4 Configuration register data

Based on the system parameters referred above, the following configuration data is generated. This is a printout from the SmartRF<sup>®</sup> Studio software.

RX mode:

Frame Addr/Data (hex)

A	002A
B	330B
C	4041
D	7EE1
E	8A00
F	BA02
G	CB6C
H	<b>E040</b>

TX mode:

FrameAddr/Data (hex)

<b>A</b>	<b>082A</b>
<b>B</b>	<b>330B</b>
<b>C</b>	<b>4041</b>
<b>D</b>	<b>7F9A</b>
<b>E</b>	<b>8A29</b>
<b>F</b>	<b>BA02</b>
<b>G</b>	<b>CB6C</b>
<b>H</b>	<b>E040</b>

PD osc. on mode:

FrameAddr/Data (hex)

A	182A
B	330B
C	4041
D	7EE1
E	8A00
F	BA02
G	CB6C
H	E040

PD osc. off mode:  
 FrameAddr/Data (hex)  
**A 182A**  
 B 330B  
**C 4051**  
 D 7EE1  
 E 8A00  
 F BA02  
 G CB6C  
 H E040

RX Precharge mode:  
 FrameAddr/Data (hex)  
**A 002A**  
**B 330B**  
**C 4041**  
**D 7EE1**  
**E 8A00**  
**F BA02**  
**G CB6C**  
**H E248**

In this application we have decided to use RX precharge, RX, TX and PD osc. off modes. Shown in bold in the listing above is the configuration data that is used, and stored in the micro-controller EEPROM.

Note: The frame address is combined with the data to get a two-byte word consisting of 3 address bits (as MSBs) and the 13 data bits (as LSBs). The combined value is shown in these listings.

## 2.5 Micro-controller EEPROM mapping

The CC400 configuration data are stored in the micro-controller EEPROM. The following map shows the data location. To customise the module for a different frequency or power level, the EEPROM data can be replaced by new values found using the SmartRF® Studio software.

EEPROM Address (hex)	Data (hex)	Reference	Comment
00	082A	TX frame A	
02	330B	TX frame B	
04	4041	TX frame C	
06	7F9A	TX frame D	
08	8A29	TX frame E	
0A	BA02	TX frame F	
0C	CB6C	TX frame G	
0E	E040	TX frame H	
10	E040	RX frame H	
12	002A	RX PC frame A	

14	330B	RX PC frame B	
16	4041	RX PC frame C	
18	7EE1	RX PC frame D	
1A	8A00	RX PC frame E	
1C	BA02	RX PC frame F	
1E	CB6C	RX PC frame G	
20	E248	RX PC frame H	
22	182A	PD (osc. Off) frame A	
24	4051	PD (osc. Off) frame C	

## 2.6 Trouble-shooting

If the module do not operate as described in this document, try the following.

<b>Problem</b>	<b>Possible solution</b>
LED is continuously on	Push the “ON” button once more. If this does not work, reset the micro-controller (see below).
LED does not start blinking after “ON” is pressed	Check battery voltage. If less than 2.7 Volt, replace battery. Only replace with a battery of the same type. If battery voltage is OK, reset the micro-controller (see below).
Receiver does not receive packet from closely located transmitter.	Use a Spectrum Analyser at 433.92 MHz to check the module during transmission. Connect the analyser to the external antenna connector foot-print and measure the output power and frequency using the “max hold” function. The output power shall be approximately 0dBm (1mW) with a peak at 433.92MHz +/-10kHz.  Test the receiver the same way in transmit mode.

The micro-controller can be reset by shorting the reset line to ground for a short moment. The reset line is located at the edge connector pin 6. Ground is pin 1. The orientation mark is between pin 2 and 3.



### 3 Demonstration Board software description

The demonstration board software is implemented in assembly and runs on the Atmel AVR 8-bit micro-controller ATtiny12L.

The micro-controller handles the CC400 configuration, data encoding and decoding using Manchester code, and implements a simple packet protocol.

A program listing with the assembly source code is available for reference and can be used as a starting point for your own applications using the AVR series of controllers from Atmel.

#### 3.1 Sleep mode

When the CC400DB is left inactive for approximately one minute the CC400 is set to power down mode, and the controller goes to “sleep”. The controller is awaked by an interrupt signal at INT0 (pin 6).

In sleep mode the current consumption is less than 1  $\mu$ A.

#### 3.2 Polling mode

In polling mode the CC400 is used for receive polling. Once every second the CC400 is set to RX mode, and the micro-controller starts searching for a preamble. The micro-controller reads two data bits, if these are not Manchester coded 1's, the CC400 is set to power down mode again. The LED is lit during the pre-amble search and will appear as a “heartbeat” signal.

In polling mode the “TX” button input (PB2) is also monitored. If this line is pulled low by the push button, the micro-controller will configure the CC400 to TX mode, and send a predefined data packet at the DIO line (PB1), before returning to receiver polling.

#### 3.3 Data packet description

The data packet consists of a pre-amble, a start-of-frame (SOF), and a data field.

The pre-amble is a train of ‘1’s. The train is longer than 1 second in order to ensure that it intercept with a polled receiver.

The SOF consists of two consecutive ‘0’s. The SOF marks the end of the preamble, and the start of the actual data.

The data field consists of ID (identification), TYPE (packet type) and DATA (user data, 3 bytes). No FEC (forward error correction) or CRC (cyclic redundancy check) is used in order to keep it as simple as possible.

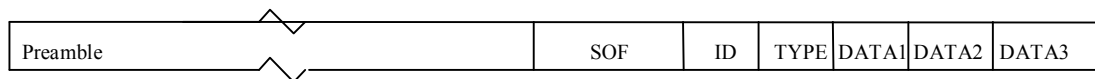
The ID is intended to be used for unique identification of different modules. However, this is not exploited in this software revision. All modules are addressed using the public address (0x00).

The TYPE is used to separate between an original message (0x80) and an acknowledge (0x81).

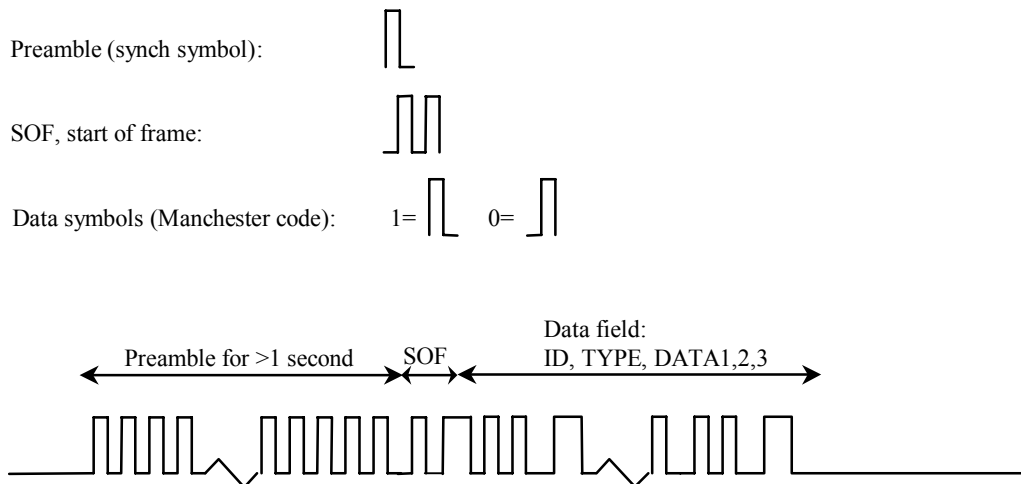
The 3 DATA bytes are used to code the blinking sequence. Every bit represents a 0.1 seconds period approximately.

No end of packet is implemented.

The data packet is shown in the figure below.



The figure below shows the symbol coding and an example of a packet with the preamble and SOF timing.



### 3.4 Data transmission protocol

The data transmission protocol is very simple, using only packet acknowledgement.

When a data package is received without errors, the packet type is checked. If it is an original message, it is shown on the LED, and an acknowledgement is sent as a reply.

However, if it is an acknowledgement packet that is received, the message is only shown on the LED without any reply.

### **3.5 Data reception**

The data reception algorithm is as follows:

1. Enter RX precharge mode, wait
2. Enter RX mode
3. The receiver shall receive synchronisation pattern (pre-amble) for some periods before it is allowed to detect SOF
4. Wait for SOF
5. Receive and decode data
6. Display data on LED

### **3.6 Data decoder algorithm**

The decoder algorithm implemented is based on oversampling. Each bit is sampled 8 times, and the data decision is based on a majority vote. The use of oversampling makes the data decision more immune to noise and duty cycle variations than a data decision algorithm based entirely on detecting edges.

The oversampling algorithm synchronises on the preamble, and thereafter continues to synchronise itself by detecting edges.

For more information on oversampling, see application note AN008 : Oversampling and data decision.

## 4 Demonstration Board hardware description

### 4.1 RF-section

The RF section consists of a CC400 chip with a few external components. The different parts of the circuit are explained below.

#### 4.1.1 PLL loop filter

The PLL loop filter contains the components C121-C123 and R121-R123. The SmartRF<sup>®</sup> Studio software program has been used to calculate the component values. Using the calculated component values for the loop filter gives an optimum loop bandwidth for the selected system parameters.

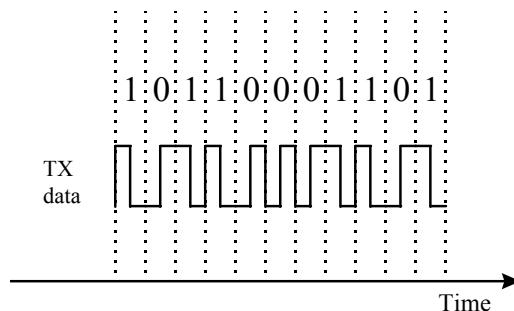
The transmitted frequency is FSK modulated, which means that the bits ‘0’ and ‘1’ is coded by jumping between two different frequencies. The loop filter bandwidth is optimised according to the frequency separation (difference between the two frequencies) and the data rate (the speed we jump between these two frequencies).

#### 4.1.2 IF filter

The Demonstration Board use the internal 200kHz IF filter. The 200kHz IF filter has a larger bandwidth than the 60 kHz IF filter, and therefor allows us to use a less accurate, and hence less expensive crystal. The sensitivity is slightly less than for the 60 kHz IF, but is considered adequate in this application.

#### 4.1.3 The modulation input/output

The modulation input/output (DIO) is connected to a micro-controller I/O pin. The data to be sent is Manchester encoded (also known as bi-phase-level coding) by the micro-controller. The Manchester code ensures that the signal has no DC component, which is necessary for the FSK demodulator to work in an optimal way. The Manchester code is based on transitions; a “0” is encoded as a low-to-high transition, a “1” is encoded as a high-to-low transition. This is illustrated in the figure below.



#### 4.1.4 LNA/PA matching

The input/output antenna matching network is optimised for 433.92MHz operation. The component values were calculated in the SmartRF<sup>®</sup> Studio software program, and consist of C51, C61, L51 and L61. To compensate for layout parasitics the component values were tuned to optimise for 50 Ohm input impedance in receive mode, and maximum output power in transmit mode.

#### 4.1.5 Voltage Controlled Oscillator (VCO)

The VCO tank is optimised for the 433MHz band, covering approximately 410 – 450MHz. The tank contains the components C91, C92, C93, L91 and the variable capacitance D2.

#### 4.1.6 Crystal oscillator

The crystal oscillator frequency is 12.000 MHz set by X1. We have chosen a low cost crystal for this application. The crystal is specified as 50/50/10/12, which means that it has a +/- 50ppm initial tolerance, and will drift less than +/-50ppm over a temperature range of –10 to 60°C. SmartRF<sup>®</sup> Studio specifies a total crystal accuracy of +/-50ppm for the parameters used here. This means that the total system is only specified for room temperature, as all the accuracy is “used up” for the initial tolerance. If a wider temperature range is required, a higher accuracy crystal should be used.

The loading capacitors, C151 and C161, together with the input capacitance and parasitics makes up the total 12pF load capacitance the crystal is designed for.

#### 4.1.7 Antenna

This antenna is an example of a PCB integrated antenna. Several antenna configurations were evaluated, and the short stub was found to be the best solution for this application. As the stub is shorter than one-quarter wavelength ( $\lambda/4 = 16.4$  cm at 433.92MHz), it has to be prolonged and matched using lumped elements. The short stub has a capacitive reactance and is thus prolonged using an inductor (L8) in series. Another inductor in shunt (L7) matches the antenna to 50 Ohm.

The antenna is fairly omni-directional, but have slightly more emission sideways. This follows from the fact that the antenna elements and the ground plane have the largest physical dimensions normal to this direction. Placing the board in a vertical direction, or placing two units parallel sideways will give the best range.

In general, small electrical antennas are not very efficient. Electrical stub antennas should be designed as long as possible, up to one-quarter wavelength. They should be kept away from the ground plane and other surrounding metallic bodies. For hand-held remote controls, the antenna should not be in close contact with the hand. If this can not

be avoided, a magnet loop antenna could be used. The loop is less sensitive to nearby objects, but suffers of very low efficiency due to very low radiation resistance. The stub antenna implemented at the Demonstrator PCB showed better efficiency than a loop of similar size, even when operated hand-held.

## 4.2 Controller section

The controller section consists of an Atmel AVR ATtiny12L micro-controller with a few external components.

### 4.2.1 RC oscillator

The internal RC oscillator is used as reference time base. The oscillator frequency is relatively independent of temperature and operating voltage. Its nominal frequency is 1.2 MHz, varying less than +/- 5% over the application temperature and supply voltage range. A calibration feature utilising a calibration word programmed into the micro-controller at the factory ensures that the oscillator frequency is equal from unit to unit.

### 4.2.2 Micro-controller I/O

As the controller is packed in an 8-pin package, it has room for only 5 external I/O lines. These are combined serving several functions in different operating modes.

The micro-controller has only one interrupt input, PB1/INT0. This input is shared between the “ON” button and the DIO signal to CC400. In sleep mode the controller will be awaked by an interrupt when this line is forced low by the push button. In receive mode the interrupt is used for data decoding. During transmit the pin is configured as an output.

The table below shows the function of all I/O pins in the various modes.

Signal/pin	Program loading	Demonstrator	Comment
PB0, pin 5	MOSI	LED	
PB1/INT0, pin 6	MISO	DIO “ON” in sleep mode	Interrupt input
PB2, pin 7	SCK	CLOCK “TX”	
PB3, pin 2		STROBE	
PB4, pin 3		PDATA	

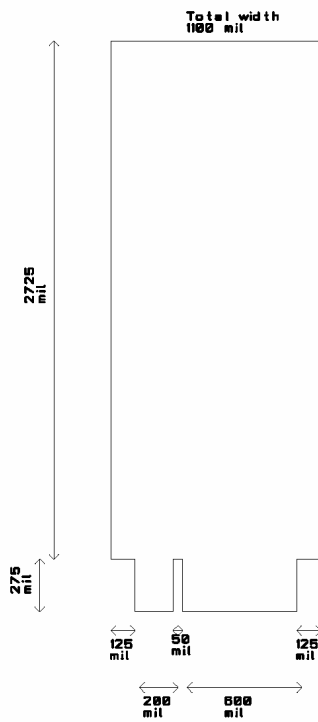
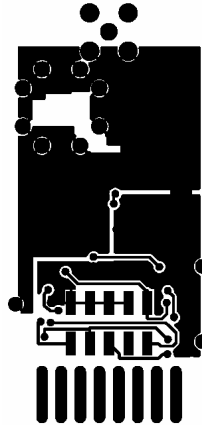
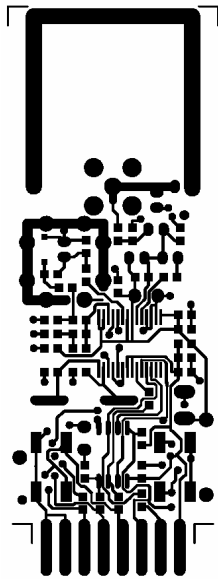
### 4.2.3 In-system-programming

Using the ATMEL AVR STK500 development board, ATAVRISP in-circuit programmer or similar devices, the micro-controller can be reprogrammed while in the circuit. Both the program code memory (flash) and the EEPROM can be reprogrammed.

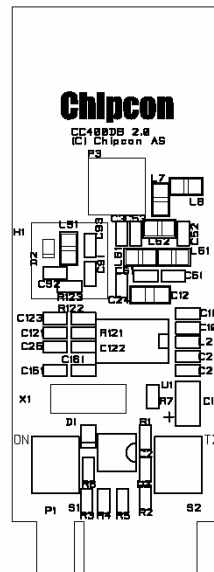
The device used for programming must be connected to the CC400DB programming socket or edge finger contact. The programming interface uses 4 signal lines. These four lines can be accessed at both the 10-pin connector footprint at the rear of the board, and at the edge contact. Pin-out is listed in the table below.

<b>Signal/pin</b>	<b>AVR programming socket, pin</b>	<b>Edge connector, pin</b>
MOSI	1	4
MISO	9	5
SCK	7	3
RESET	5	6
GROUND	4,6,8,10	1,2

### 4.3 Layout sketches

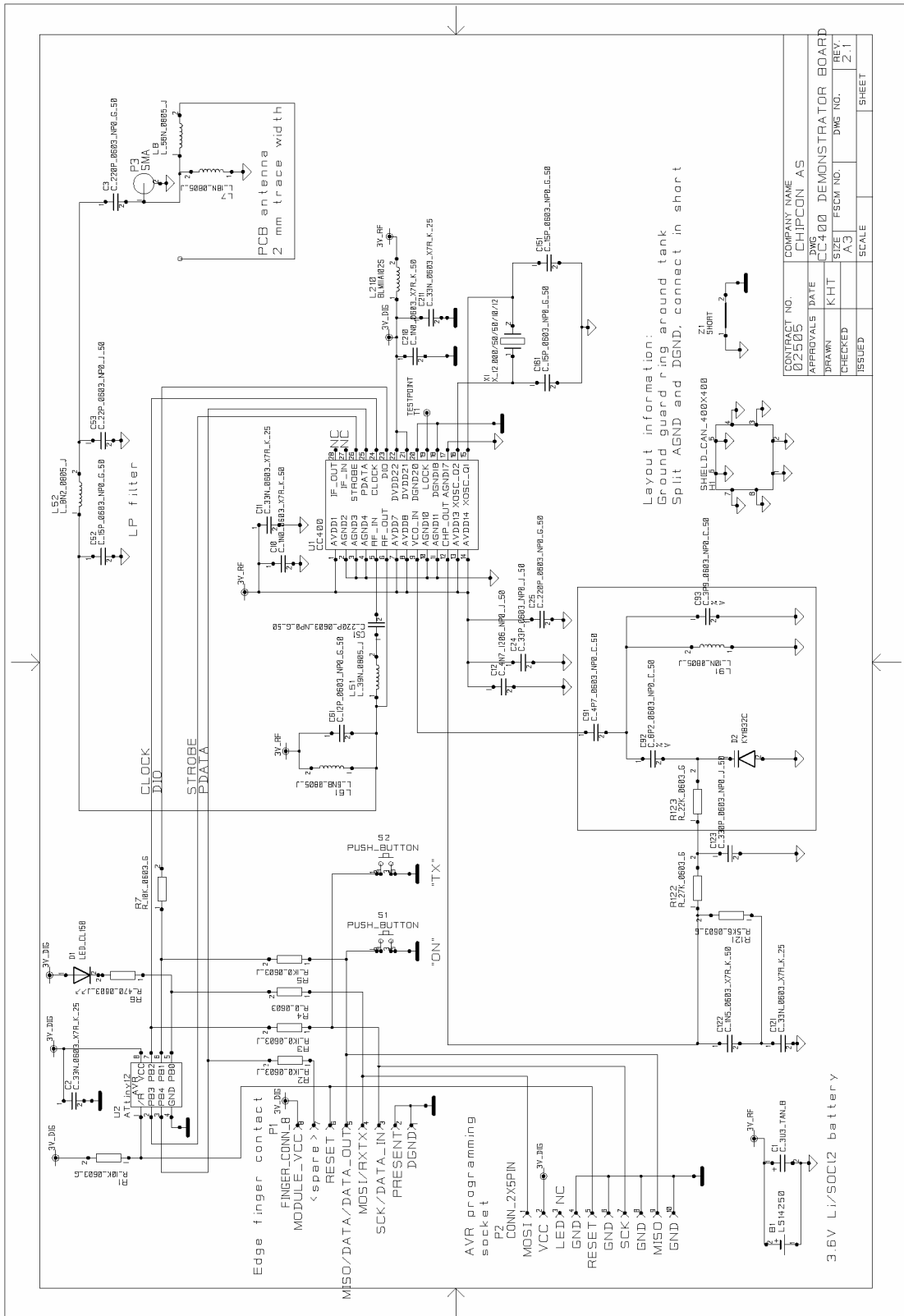


Total height  
3888 mil





### 4.4 Circuit diagram



## 4.5 Bill of materials

CC400DB			
Reference	Description	Value	Part
B1	Battery	3.6V	LS14250 3PF, Lithium Thionyl Chloride
C1	Capacitor, tantal	3.3 $\mu$ F	C_3U3_TAN_B
C2	Capacitor 0603	33nF	C_33N_0603_X7R_K_25
C3	Capacitor 0603	220pF	C_220P_0603_NP0_G_50
C10	Capacitor 0603	1nF	C_1N0_0603_X7R_K_50
C11	Capacitor 0603	33nF	C_33N_0603_X7R_K_25
C12	Capacitor 1206	4.7nF	C_4N7_1206_NP0_J_50
C24	Capacitor 0603	33pF	C_33P_0603_NP0_J_50
C25	Capacitor 0603	220pF	C_220P_0603_NP0_G_50
C51	Capacitor 0603	220pF	C_220P_0603_NP0_G_50
C52	Capacitor 0603	15pF	C_15P_0603_NP0_G_50
C53	Capacitor 0603	22pF	C_22P_0603_NP0_J_50
C61	Capacitor 0603	12pF	C_12P_0603_NP0_G_50
C91	Capacitor 0603	4.7pF	C_4P7_0603_NP0_C_50
C92	Capacitor 0603	8.2pF	C_8P2_0603_NP0_C_50
C93	Capacitor 0603	3.9pF	C_3P9_0603_NP0_C_50
C121	Capacitor 0603	33nF	C_33N_0603_X7R_K_25
C122	Capacitor 0603	1.5nF	C_1N5_0603_X7R_K_50
C123	Capacitor 0603	330pF	C_330P_0603_NP0_J_50
C151	Capacitor 0603	15pF	C_15P_0603_NP0_G_50
C161	Capacitor 0603	15pF	C_15P_0603_NP0_G_50
C210	Capacitor 0603	1nF	C_1N0_0603_X7R_K_50
C211	Capacitor 0603	33nF	C_33N_0603_X7R_K_25
D1	LED	Red	CL150, Citizen
D2	Varactor diode		KV1832C, Toko
L7	Inductor 0805	18nH	L_18N_0805_J
L8	Inductor 0805	56nH	L_56N_0805_J
L51	Inductor 0805	39nH	L_39N_0805_J
L52	Inductor 0805	8.2nH	L_8N2_0805_J
L61	Inductor 0805	6.8nH	L_6N8_0805_J
L91	Inductor 0805	10nH	L_10N_0805_J, Murata LQN21A
L210	EMI filter bead		BLM11A102S, Murata
R1	Resistor 0603	10k $\Omega$	R_10K_0603_G
R2	Resistor 0603	1k $\Omega$	R_1K0_0603_J
R3	Resistor 0603	1k $\Omega$	R_1K0_0603_J
R4	Resistor 0603	0 $\Omega$	R_0R_0603
R5	Resistor 0603	1k $\Omega$	R_1K0_0603_J
R6	Resistor 0603	470 $\Omega$	R_470_0603_J
R7	Resistor 0603	10k $\Omega$	R_10K_0603_G
R121	Resistor 0603	5.6k $\Omega$	R_5K6_0603_G
R122	Resistor 0603	27k $\Omega$	R_27K_0603_J
R123	Resistor 0603	22k $\Omega$	R_22K_0603_G
S1	Push button		PUSH BUTTON
S2	Push button		PUSH BUTTON
U1	Single chip transceiver		CC400
U2	Micro-controller		ATtiny12L-4SC
X1	Crystal, HC-49-SMD		X_12.000000_MHz, 12pF load

## 5 Using the Demonstrator Board as a prototype module

The purpose of the Demonstrator Board is to show a simple transceiver module with battery supply and integrated PCB antenna. The demonstrator can be used “as is” for simple range and coverage measurements. But keep in mind that this module uses only 0 dBm and a small rather ineffective antenna. The range can be extended substantially using more power and a better antenna.

The Demonstrator Board can also be used as a prototype module when developing your own applications. You may:

- Reprogram the EEPROM configuration data for your own system parameters. SmartRF<sup>®</sup> Studio generates the configuration data values. See section 2.5 for EEPROM memory map.
- Rewrite the micro-controller code to use the edge connector to interface your host system
- Insert a standard 2.54mm pitch contact to interface a motherboard at the edge connector.
- Remove L7 and L8, and mount a coax connector at the antenna output. Thus an external antenna could be used. The antenna should be matched to 50 Ohm