RADIOMETRIX

Hartcran House, 231 Kenton Lane, Harrow, Middlesex, HA3 8RP, England
Tel: +44 (0) 20 8909 9595, Fax: +44 (0) 20 8909 2233, www.radiometrix.com

CNM2

Issue 1, 06 June 2024

Economical Narrow Band UHF Radio Modem

The CNM2 is an economical easy to use low power narrow band GFSK radio modem module that offers up to 50mW RF power 4800 baud rate data link with 3.3V TTL UART interface.

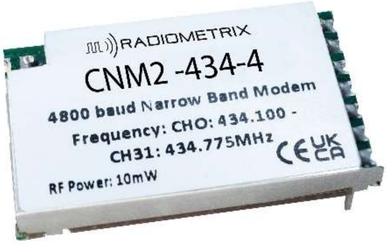


Figure 1: CNM2-434-4

Features

- Standard 433.05-434.79MHz (EU), 458.5-459.1MHz (UK)
- Available from 420MHz to 470 MHz
- Frequency Programmable
- 25kHz channel spacing Narrow Band Multichannel Frequencies
- TCXO referenced Fractional-N PLL Frequency Synthesiser
- ETSI EN 300 220-2 and FCC Part 15.249, FCC Part 90 Emission Mask D compliant
- <3ms Transmit Switching Time
- 4800 baud rate Transparent (streaming) or
  direct control over packet operation (command mode)

Technical Summary

- Supply range: 3.4-15V DC
- Current consumption: 28mA TX (10mW) or 52mA TX (50mW)
  17mA RX or 10μA (idle)
- RF baud rate: 6.6kbps GFSK (recommended for streaming) or 1.5kbps GFSK
- User baud rate: 4800bps (streaming mode) or 9600bps (command mode)
- Transmit power: 10mW default. 1mW to 50mW (adjustable)
- Receive Sensitivity: -115dBm for 0.1%PER @ 6.6kbps
- 26MHz TCXO 1ppm high stability reference
- Size: 32 x 21 x 5mm
- 0.1” pitch 0.65mm square pins or castellated SMD pads

Operational Description

The CNM2 radio modem can be used as either a packet orientated system or serial data streaming mode using programmable speed (1.5 or 6.6kbps GFSK) over-air link data rate. The relatively low data rates ensure optimum sensitivity and range for simple control applications.

Data packets are either short (1-8 bytes) or long (9-16 bytes) long, depending on the amount of user payload data uploaded (Packet length is selected automatically)

A straightforward “inverted RS232”, or UART type asynchronous baud (1 start, 8 data bits, no parity, 1 stop bit) serial port communicates with the user, employing a simple command protocol to both send and receive data, and to control critical RF parameters (such as channel frequency and RF power level)

The default operating state for the module is always receive. The CNM “listens” for a valid data packet (correct framing sequence, packet length and checksum) and in the event of finding one: decodes it, loads the payload data into an internal FIFO buffer and begins streaming it out via the serial port.

A CNM can receive and decode a new packet while still outputting a previous one.

Recommended PCB hole size 1mm after plating

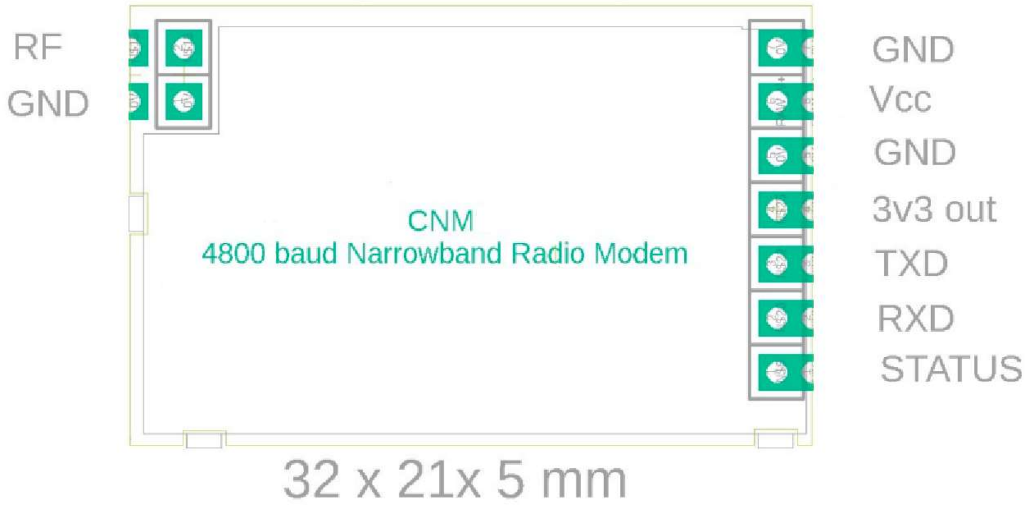


Figure 2: CNM2 Pinout and Dimensions

PIN	DESCIPTION
GND	Supply Ground
VCC	3.4V-15V DC Supply Voltage
GND	Supply Ground
3.3VOUT	3.3V 50mA max output; can also be used as regulated 3.3V input
Serial in (TXD)	3.3V TTL level UART Transmit Data Input Series 100kΩ “stopper” but NO pullup (tie to 3.3VOUT if unused)
Serial out (RXD)	3.3V TTL level UART Received Data Output Idles high. 330Ωseries resistor
STATUS	Active High when unit is transmitting. 330R series resistor
RFGND	RF ground
RF	50Ω RF Input / Output Antenna Port

General	min.	typ.	max.	unit	notes
DC supply					
Supply voltage (or 3.3V Regulated)	3.4		15	V	1
TX Supply current @ 1mW/10mW/50mW	19	28	60	mA	2
RX Supply Current		17		mA	
Idle State Current			10	µA	
RF					
Operating Frequency Band f_c		Programmable 420 – 470 MHz			
Centre Frequency Accuracy			1	ppm	3
Channel width (frequency step)		25		kHz	
RF Output Power	1	10	50	mW	2
Harmonic Emissions		-50		dBm	4
Receive Sensitivity (@ 6.6kbps)		-115		dBm	5
Adjacent Channel Rejection	-55			dB	
Occupied Band Width (OBW)		15		kHz	
RF link data rate			1.5	kbps	
			6.6	kbps	
Preprogrammed Frequency Channels			32+1		6
RF pin / Antenna Port Impedance		50		Ω	
Dynamic timing					
Transmit on switching time			3	ms	

Notes:

1. 3.3V regulated supply into 3.3VOUT instead of VCC
2. Transmit current depends on RF power setting for the country specific band
3. Frequency stability over -20˚C to +70˚C (and supply voltage variations.)
4. Meets EN300-220-2
5. Receive Sensitivity is defined at the 0.1%-bit error rate point.
6. Selected by serial command; Channel frequencies are factory pre-programmed according to country specific band.

For UK version the following channel assignment is supplied:

Chan	Frequency	Chan	Frequency	Chan	Frequency	Chan	Frequency
	(MHz)		(MHz)		(MHz)		(MHz)
0	434.100	8	434.300	16	434.500	24	434.700
1	434.125	9	434.325	17	434.525	25	434.725
2	434.150	10	434.350	18	434.550	26	434.750
3	434.175	11	434.375	19	434.575	27	434.775
4	434.200	12	434.400	20	434.600	28	434.775
5	434.225	13	434.425	21	434.625	29	434.775
6	434.250	14	434.450	22	434.650	30	434.775
7	434.275	15	434.475	23	434.675	31	434.775

Operating Modes (in detail)

The method used to transmit data depends on which operating mode the unit has been set to. There are two operating modes: **Transparent** and **Command.**

The operating mode is determined by the value of the MODE byte (specifically bit one)

This will be dealt with in more detail later

Transparent Mode

This is the default (as supplied) operating mode for a CNM, and is probably easiest to use. In this mode the user interface port runs (usually) at 4800 baud

In "**transparent**" mode the CNM will buffer and transmit bytes arriving over the serial port without any further input from the user.

If, however, data is sent at 2400 baud, then a CNM (in transparent mode) will interpret it as a "command" string, and attempt to parse it accordingly. Yes. It's a 2 speed port

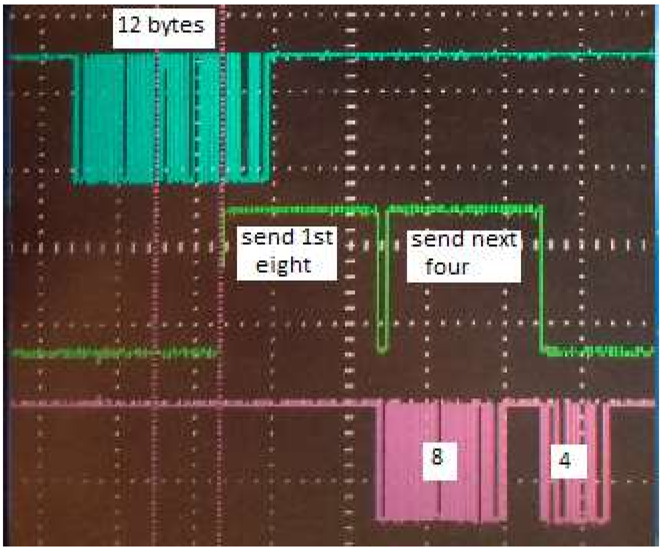
So how does a CNM handle data in Transparent mode?

CNM is a packetised data radio modem. This means that all data transmitted over the radio link is formatted into discrete messages (or "packets") consisting of preamble, a framing sequence, the data payload and a checksum. At the receiving end the preamble allows the receiver to "lock onto" the data stream, the framing sequence identifies a valid packet, and the checksum confirms it to be free of data errors. A CNM receiver "seeing" a packet with either a framer or a checksum error will reject the packet and output nothing. Unlike a "raw baseband interface" radio, a CNM in idle/receive mode outputs nothing in the absence of a valid packet. It does not throw out random "line noise" bytes.

To activate the transmitter and send a packet (remember: the default state for a CNM is receive/idle) the user just sends data to the serial input at 4800 baud.

Even if it is sent a constant stream of bytes, the unit will transparently buffer them, format them into packets and transmit a succession of RF bursts, each one containing a packet.

The amount of data in each packet is variable (otherwise if you sent the unit a number of bytes less than the pre-set payload length in a packet, it would wait indefinitely before transmitting).

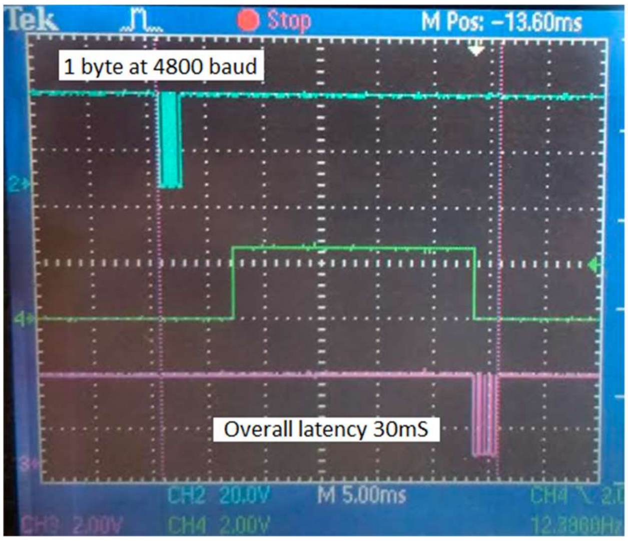


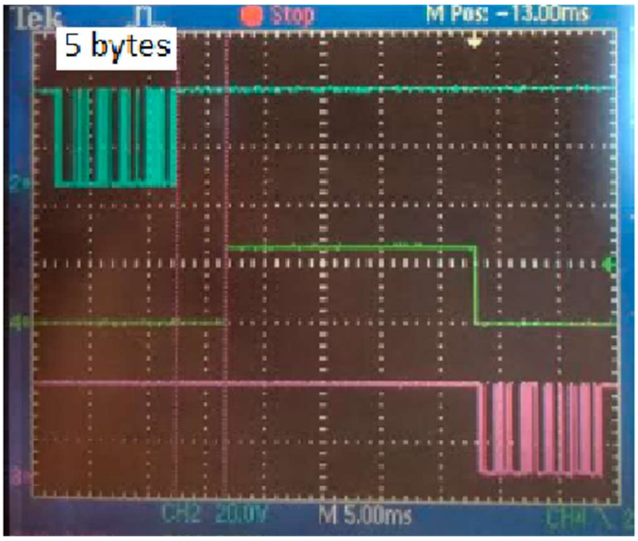
In the case of the CNM packets can either be "short" (containing 1-8 bytes of payload) or "long"(containing 9-16 bytes). The decision as to which length of packet to send is made automatically by the firmware, based on the contents of the transmit buffer and the timing of the data arriving at the port

This is handled by a simple set of rules:
If data is streamed (characters have no, or minimal, delay between them) then the unit waits until it has 8 bytes in the buffer, and then starts with a short packet. Assuming the data keeps coming then the next packet will be a "long" containing the data received while the first packet was being sent (between 10 and 12 bytes) ... then another "long" containing the data received while the second packet was transmitting (between 15 and 16 bytes) ... and so on.

If, however, the unit is not already transmitting AND it detects a 3mS or longer gap in the input data then it will initiate an "early" short packet, containing the data so far buffered

This allows the user to trigger short packet transmissions of up to 8 bytes in a highly predictable manner (useful for "control" applications) while also avoiding excessive latency on data-streams with inconsistent inter-byte-delays (on a "send what you've got, then get some more" basis)





The inquisitive reader might be wondering why the CNM uses fixed length packets and then "wastes" payload throughput (as, for example, a single byte is still sent as a short packet with the other 7 bytes in the payload as nulls) ?

There are actually a few reasons. Simplicity (in the design execution) is the obvious one, but there are also engineering reasons

Firstly, the fixed length packets makes it much easier to set up deterministic timing when using CNMs in simple networks (if you know exactly how long a burst is, and when it will be sent, then things get a whole lot easier)

Secondly, and possibly more importantly, it reduces false packet detection and spurious data output at low signal levels on short bursts. The effectiveness of data integrity detection methods (and CNM uses both a checksum and “forbidden codes” in the formatted bitstream) depends on the overall packet length, and can become ineffective if the number of decoded bits falls much below 32.

The radio link can also operate at one of two different speeds (set by MODE bit 0).

1) 6.6kbps (default)
2) 1.5kbps

The faster (default) 6.6kbps rate is intended for use with “transparent” mode to send and receive data at 4800bps, while the slower 1.5kbps speed optimises range at the price of longer data packet timings, and is usually used for industrial control applications in “command” mode (it is possible to use the slower link in transparent mode, but the 64 byte Transmit buffer will fill up faster than it can empty and data overrun errors may occur, so the user needs to carefully watch the data throughput)

Command Mode (and “commands” in general)

In command mode, the user port operates at 9600 baud, and the unit will only respond to correctly formatted “command strings”. It will not (in this mode) just “stream” data

In order to make a CNM set to command mode send data over the radio interface, the user must employ a specific command string (the @PKT command)

NOTE:

There is potential for confusion in the difference between “command mode” and “commands” in general. The exact same commands (bytes or strings) are used in both transparent mode and in command mode

In transparent mode these commands are sent at 2400 baud (while 4800 baud data would go straight to the radio link) but in command mode they are sent at 9600 baud

Command strings have four different formats:

Single byte commands
Two byte command
Command packet send
Other “long” commands

All long commands start with an ‘@’ character (0x40), and terminate with an ‘]’ (0x5D)

Following the reception of the @ character at the start of a command, the unit will stop receiving, and any data left in the RX buffer will be lost.

Single and two byte commands … are pretty much what they sound like

Single Byte Commands (mostly test or diagnostic functions)

Command	Hex	Dec	Description
!	0x21	33	Puts unit into TEST transmission (CW by default)
$	0x24	36	Puts the CNM into SLEEP mode (low Iq, no RX)
&	0x26	38	Execute a single RSSI read
%	0x25	37	Internal EEPROM status read-out

The TEST and the SLEEP modes are terminated by any activity on the serial input (a space (0x20) byte is recommended)

The RSSI command initiates an immediate RSSI read, and outputs a single character (raw binary) reply over the serial port

The status read-out command % triggers a (human readable ascii) hex dump of the internal EEPROM (locations 0x10 to 0x18) separated by spaces. See below

0x10	Version number
0x11	power setting word
0x12	frequency trimming offset (bit 7 is the sign bit)
0x13	frequency word LSB
0x14	“
0x15	“
0x16	frequency word MSB
0x17	“mode” byte 0x00 = default mode (transparent, fast)
0x18	Channel pointer number

Two Byte Command (channel select)

At this point a digression into the frequency selection functions of the CNM would probably be a good idea.

The CNM has one ‘default’ frequency store (selected if the channel pointer is set to a number >31) and thirty two ‘channel frequency’ stores, selected by **channel pointer** values from 0 through 31.
While the two byte command can only make temporary changes to the current channel, there are long commands which can overwrite the EEPROM values in the channel pointer, the default store, and the channel stores. (detailed later)

Most users will never need to go beyond a “channel change” command, as the module is usually supplied with the table of frequencies that the customer needs already programmed in

	Byte	Hex	Command	
1	#	0x23	Channel Select	
2	(varies)		raw binary channel number	(2 byte command)

This is a volatile set-up. After a power cycle (or reset) the channel pointer value reverts to the last value written into the EEPROM, and not the last “two byte” command value

Example: # followed by binary 00000100 will select channel 4 (which is 434.200MHz)

The two byte command also has a number of subsidiary functions:
1. Programming a value of 0xFE into the channel pointer results in a hard RESET
2. Programming a value of 0xFD causes the unit to reply with an ascii + byte

Command mode packet send

This is a special case “long” command, which allows a CNM set in “command” mode to transmit a (defined) data packet over the radio link. It is probably the most commonly used of the long sequences. Like the other long commands, it starts with an **@** and ends with **]**

@PKT <LEN> <payload bytes>]

The <LEN> value is a single byte (1-16) representing the number of payload bytes sent
It can EITHER be raw binary (0x01-0x10) OR an ASCII A through P

The <payload bytes> field is simply one to sixteen bytes of user data

This process will take approximately 20/86ms (1-8 bytes) or 36/133ms (9-16 bytes)
While in transmit mode (STATUS pin HIGH) it will ignore any other commands.
If used in “command” mode it will send a single ‘>’ character (0x3E) over the serial port at the end of the operation before returning to receive operation.

Critical timings
No more than 20ms is permitted BETWEEN characters after the initial @ character, or the CNM will empty its internal buffers and return to default (receive) mode
Between command strings (following the ‘]’ character (0x5D) a pause of at least 10ms is required (and, more following an EEPROM write, which needs a longer idle period, of 25ms at least)

Other long commands

The remaining command strings permit the complete reprogramming of the CNM.

These MUST be used with great care, as it is possible to disable the unit by their incautious use

The *generic* format of a long command is this

@	prefix character
SET or RAM	the EEPROM or non volatile select descriptor
c, m, f, p, or t	the actual function code
1 or 4 bytes	of “operand” data
checksum byte	(see below)
]	suffix character

The checksum is a single (raw binary) byte. It is the **8 bit truncated binary sum** of all operand bytes and the (ascii value of the) function code

A **RAM** command will set up the CNM accordingly, but will put the programmed operand into volatile memory. Resetting the unit will return it to the previous default value
A **SET** command **also** overwrites the default value in EEPROM. There is no way of “un doing” a command of this type

Long commands (that can be generally used, without special restrictions)

`@SETc <chan byte> <checksum>]`	These commands program the channel pointer
`@RAMc <chan byte> <checksum>]`	(@RAMc is equivalent to the two byte command)
`@SETm <MODE byte> <checksum>]`	Sets up the all important MODE byte
`@RAMm <MODE byte> <checksum>]`	**(see below)**

The Mode Byte

(Editing this value will be fairly common, to set operating mode and radio link data rate)

Bit	IF a ZERO	IF a ONE
Bit 7	Normal operation	Constant CW
Bit 6	Normal operation	CW TX replaced with 1200 bps PN9 data (3kHz dev.)
Bit 5	Reserved	
Bit 4	Normal operation	Long (pager type) 254 byte preamble
Bit 3	Reserved	
Bit 2	Reserved	
Bit 1	Transparent mode	Command Mode
Bit 0	Fast (6600bps) radio data	Slow (1500bps) radio link data

The default MODE byte value for a CNM2 is ZERO (transparent mode, fast data)

Long commands (that require a fair bit more consideration)

@SETf <four byte> <checksum>]	These commands program the default frequency
@RAMf <four byte> <checksum>]	

The operand in this case is a four byte (32 bit raw binary) number
You calculate it as:

$$ProgrammedNumber = f_c(MHz) \times \frac{2^{19}}{6.5MHz}$$

400MHz bands UHF Version

If the three most significant bytes of this operand zero then a "**transfer current default to channel map**" is executed, using the LSB as the channel number (valid channels: 0 to 31)

Long commands (that need “RF development lab” level test equipment)

(These are commands which make fine adjustments to the module’s internal parameters, and therefore need RF test equipment to confirm, for example, output power and exact frequency. These should only be approached by experienced users)

@SETc <trim byte> <checksum>]	These commands fine tune the frequency
@RAMc <trim byte> <checksum>]	Operand is a signed 7 bit integer
@SETm <power> <checksum>]	Sets the transmit output power
@RAMm <power> <checksum>]	Allowable operand range 0-200

Example: setting the default frequency to 433.975MHz

@SETf<4 bytes raw binary, msb sent first><checksum>]

$$ProgrammedNumber = f_c(MHz) \times \frac{2^{19}}{6.5MHz}$$

(433975000 x 524288) / 6500000 = 35,004,289.969230
So, programming number = 35,004,290 (or 2161F82 in hex)
(round to the nearest whole number)

Then, we know that the checksum is the **8 bit truncated binary sum** of all payload bytes and the function code (f) … So the checksum would be,
66+02+16+1F+82 = 0x11F
Lowest 8 bits ...0x1F

So, the command string would be:
@SETf
followed by the 4 bytes **0x02161F82**
followed by the single byte **0x1F**
and finally an **]** character

CNM Evaluation Kit

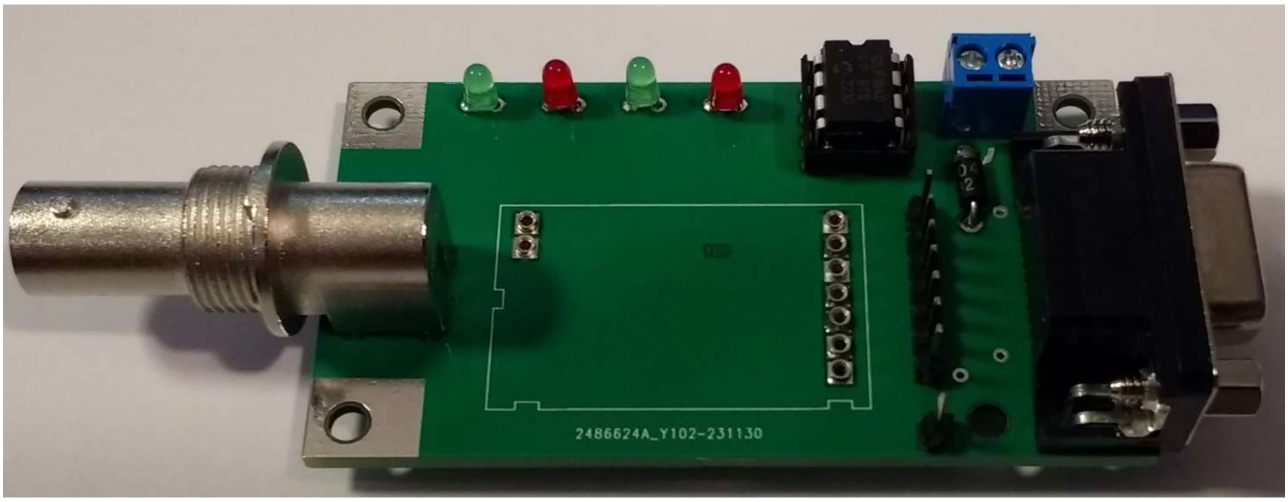


Figure 3: CNM2/CNM3 Evaluation Kit with RS232 interface and bus activity LEDs

Dimensions (excluding BNC and DE9F connector): 72 x 39 mm

Status LEDs from Left (BNC end)	Function
D1 Green	STATUS / TX ON
D2 Red	TXD Activity
D3 Green	RXD Activity
D4 Red	Power Applied

D4 flashes at power up, or when a CNM has just been inserted into a powered board.
If there is no TXD/RXD activity, the Eval Kit goes into sleep mode drawing <10µA

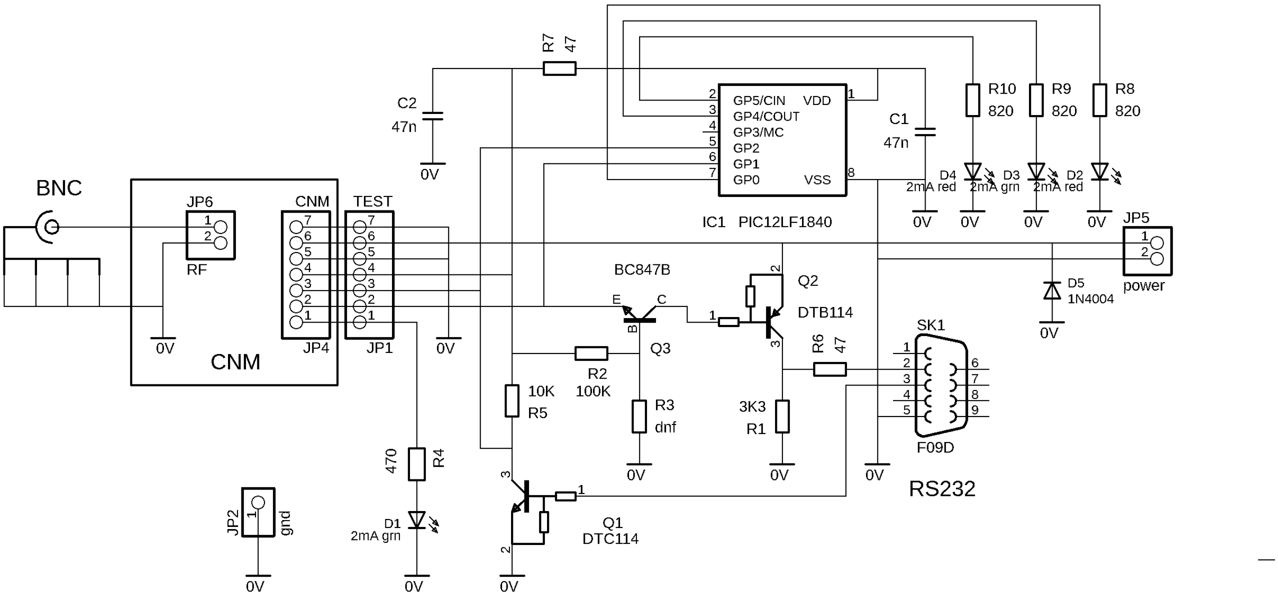


Figure 4: CNM Evaluation Kit Schematics

JP1 User Interface	Function
Pin 1	STATUS / Transmitter ON output
Pin 2	3.3V TTL level UART Received Data (RXD) Output
Pin 3	3.3V TTL level UART Transmit Data (TXD) Input
Pin 4	3.3V 50mA max output or regulated 3.3V input without VCC
Pin 5	GND
Pin 6	3.4V-15V DC Supply Voltage

Variants and ordering information

The CNM2 Modem is manufactured in several variants:

CNM2-434-4	10mW	Europe
CNM2-434-4-25mW	25mW	Australia
CNM2-458-4	50mW	UK

Please note it is the responsibility of end user to adjust RF Power and Transmit Duty cycle according to operating sub-band within 433.05-434.79MHz.

https://efis.cept.org/adhoc_grabber.jsp?annex=4
g1 433.05MHz - 434.79MHz 10 mW e.r.p. ≤ 10% duty cycle Not specified
g3 434.04MHz - 434.79MHz 10 mW e.r.p. No requirement ≤ 25 kHz

For other variants please contact the factory.

Other variants can be supplied to individual customer requirements at frequencies from 142MHz to 1050MHz

Radiometrix Ltd
Hartcran House
231 Kenton Lane
Harrow, Middlesex
HA3 8RP
ENGLAND
Tel: +44 (0) 20 8909 9595
Fax: +44 (0) 20 8909 2233
sales@radiometrix.com
www.radiometrix.com

Copyright notice

This product data sheet is the original work and copyrighted property of Radiometrix Ltd. Reproduction in whole or in part must give clear acknowledgement to the copyright owner.

Limitation of liability

The information furnished by Radiometrix Ltd is believed to be accurate and reliable. Radiometrix Ltd reserves the right to make changes or improvements in the design, specification or manufacture of its subassembly products without notice. Radiometrix Ltd does not assume any liability arising from the application or use of any product or circuit described herein, nor for any infringements of patents or other rights of third parties which may result from the use of its products. This data sheet neither states nor implies warranty of any kind, including fitness for any particular application. These radio devices may be subject to radio interference and may not function as intended if interference is present.
We do NOT recommend their use for life critical applications.
The Intrastat commodity code for all our modules is: 8542 6000

Radio Equipment Directive (RED)

Before it can be placed on the UK market, radio control equipment must first comply with the provisions of the Radio Equipment Directive 2014/53/EU.

To comply, all equipment has to meet a set of Essential Requirements that are based on voluntary Harmonised European Standards. Manufacturers can meet the essential requirements by ensuring equipment meets the applicable harmonised standards or by seeking the opinion of an Radio Equipment Directive Notified Body. Once this assessment has been carried out, the manufacturer can declare compliance, affix the CE mark to the equipment and then place it on the market anywhere in the European Community.

https://www.ofcom.org.uk/spectrum/information
https://www.ofcom.org.uk/spectrum/rules/
https://ec.europa.eu/docsroom/documents/33162