Wireless RS232 Link

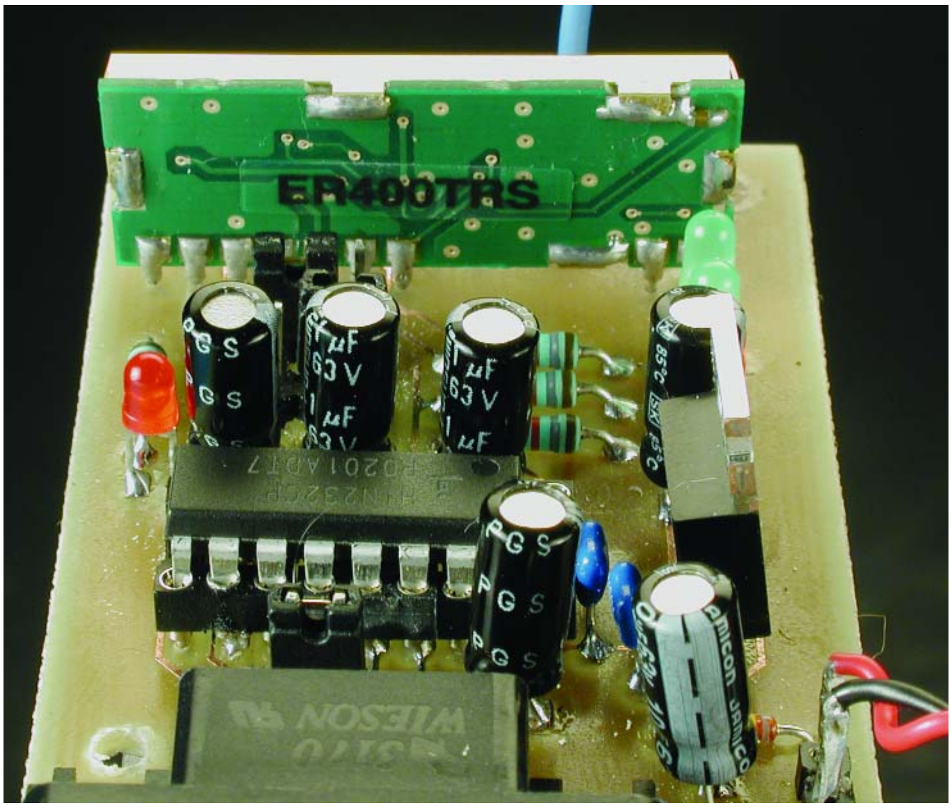
Using licence-exempt Short Range Devices (SRDs)

Design by D. Langwald, P. Groppe and B. vom Berg

As we show in this article, off the shelf radio modules called Short Range Device (SRDs) with an integrated microcontroller are here, making home construction of a wireless RS232 link a reality for DIY constructors.

Wireless transmission of 'information', be it digital data, music or video, is sure to find wide application areas. Today, those of you with an active interest in 'over-the-air' technology have a large number of options to choose from, ranging from transmission of infrared light pulses (remote controls, IrDA) right up to more challenging fields like RF applications called DECT, Bluetooth, Wireless LAN and so on. For 'large' Windows computers you can buy ready-made modules which are not only easy to add to existing equipment, but also certified and complete with a manufacturer's product warranty, not forgetting a suitable driver for the software link.

The alternative road is a bit more exacting. For example, if an automation component or a measurement system (sporting just about any microcontroller) is to be upgraded with a radio link, a custom solution involving RF components is sure to demand a lot of expertise, time and money, mostly due to three aspects:



– Developing an RF front end and transmitter for the relevant frequency range, observing the legal limits for bandwidth, spurious radiation and output power.
– Certification or type approval of the RF parts by an appointed national or international regulating authority, and paying the bill for the paperwork.
– The design of a suitable data transmission protocol to make sure that interference due to external noise and interference, moving objects, reflections, indirect paths etc., can be overcome by the system itself — the requirements in this respect are much higher than with, say, an RS232 link using the traditional cable.

An interesting alternative to the above is available in the form of cheap, off the shelf, type approved and multi-purpose radio modules operating in one of the allocated ISM band (Industrial / Scientific / Medical) and having a UART interface. In the case of the module used in our project, the 'firmware' for the data transmission process is available from the SRD manufacturer. Admittedly, controlling the SRD in software using the available firmware is not child's play, but it's not rocket science either!

Within the boundaries of the ISM bands (for example the pan-European 433.05 – 434.79 MHz band) it is possible (in many, but not all countries) to employ, licence-free, a type-approved SRD with an ERP (effective radiated power) not exceeding 10 mW.

As a matter of course, the manufacturer (but also the end user) of SRDs has to ensure that the product is compliant with rather strict regulations. In the Europe, for example, the standards for unlicensed use of SRDs in the 433/434 MHz band are laid down in EN300-220-3 and EN301489-3. Those of you interested in the legal/technical aspects of SRD type approval standards are referred to the Low Power Radio Association (LPRA) which excels in supplying information on SRDs at all levels as well as in publishing a magazine and a website, the latter being posted at www.lpra.org.

Most SRD-based radio systems employed in remote controls convey data, telemetry or alarm information across relatively small distances (usually limited to a few hundred metres 'line of sight' — LOS). In some ISM bands, the transmission of audio and video signals is allowed, too. In everyday life, SRDs are used in appliances like vehicle central locking systems, outdoor wireless thermometers, cordless mice and keyboards for the PC, wireless headphones, security cameras, and so on. The internationally used abbreviation for such radio systems used to be LPD (low-power devices) but that seems to have evolved into the more modern SRD (short range device).

The UK-based firm Low Power Radio Systems (LPRS) has a number of 'embedded-software' SRDs in its impressive range of licence-exempt radio modules. One of these, the ER400TRS 'Easy Radio' transceiver, is used in our project — its main specifications and internal organisation are given in the inset.

Little more than the module

The circuit diagram of a complete wireless data transmission system as shown in **Figure 1** proves that no special RF experience is required to be able to build this interesting project. The four low-power (diagnostic) LEDs in the circuit indicate its main functionality:

LED D1: supply voltage
LED D2: data transfer: transmit
LED D3: data transfer: receive
LED D4: state of BUSY line

The ER400TRS is driven with TTL signals, which forces us employ a level translator (here, a MAX232) if the radio module is to be connected to the COM port of an ordinary PC. The MAX232 (IC2) is not required in a microcontroller system if you are absolutely sure the system UART operates with TTL levels only. The transmitted and received data are quite ordinary UART characters as they occur in a standard serial asynchronous (V24) interface like a COM port on a PC or a UART in a microcontroller system.

The ER400TRS offers two additional hardware handshake control signals, whose use is not mandatory:

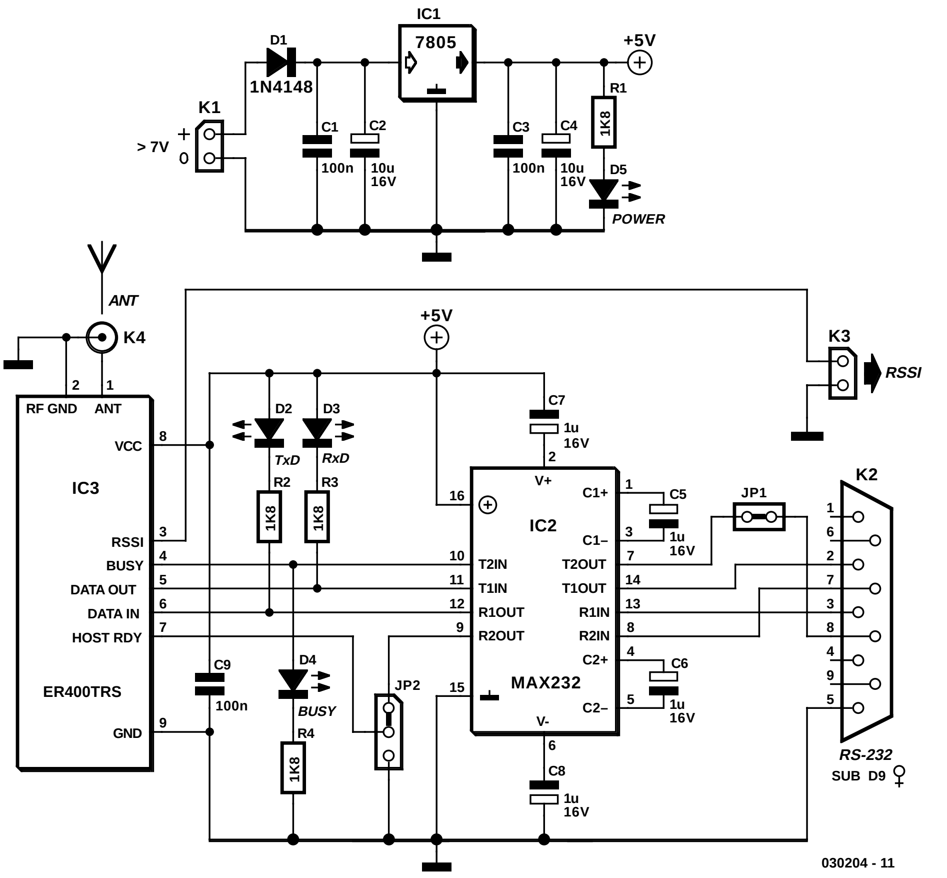


Figure 1. The external circuitry around the radio module comprises just a few parts, really.

BUSY (output)
This signal indicates that the module is processing commands (reception of data, error checking) and not capable of loading transmitter data. In RS232 parlance, this corresponds to the CTS (clear to send) signal.

The following states apply:

BUSY = 1: the ER400TRS is busy and no transmit data should be conveyed to it on penalty of being ignored (= lost).

BUSY = 1: The ER400TRS is ready to accept data and actually transmit them.

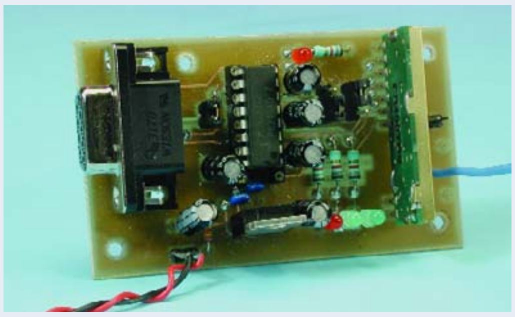
Jumper JP2 allows the BUSY signal to be applied to the RS232 converter and from there, to the host for processing by the PC software. Alternatively, the signal may be just visually signalled by LED D4. If this control signal is not actually processed by software or hardware, then provision must be made to ensure the ER400TRS is really ready to load and subsequently transmit data. In some cases, this means that delays following transmission and reception should be built into the software, and be strictly maintained until the next data block is sent.

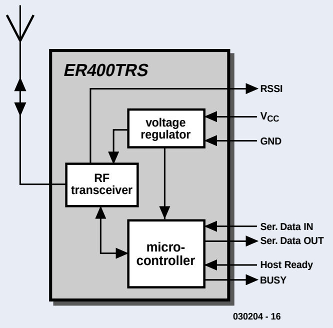
Jumper JP3 enables the BUSY signal to be

ER400TRS Quick Data

Datasheet at: http://www.lprs.co.uk/main/viewdatasheet.php?datasheetref=112

- Half-duplex FM transceiver
- 10 programmable channels within 433.25 MHz and 434.35 MHz(70 cms pan-European ISM allocation)
- 50-Ω antenna connection
- RF output power programmable in 10 steps between 1 mW and 10 mW
- Range 250 m line of sight (LOS) or 30 m in buildings
- Data speed on host link: 2.4 to 38.4 kbits/s, adjustable in five steps
- Fixed 19.2 kbits/s data speed on RF link
- Two hardware handshaking signals
- RSSI (received signal strength) analogue output
- Supply voltage 3.3 V to 5.5 V
- Current consumption at 5.5 V supply;
  - transmitting (10 mW): 23.0 mA
  - receiving: 17.0 mA
  - standby: 2.0 mA
- Dimensions: 37.5 x 14.4 mm





removed from the sub-D plug pin, which may be necessary to prevent signal conflicts or contention.

Host Ready (input)
This signal tells the module whether or not the host is ready to load data from the receive buffer in the ER400TRS. Again, in RS232 terms this equates to an RTS (ready to send) signal, with the following logic states:

Host Ready = 0: Host CPU is ready to load and process the data that was just received. The module should respond by releasing the data to the host.

Host Ready = 1: Host CPU not ready to load and process the data that was just received. In this case the radio module should keep the data in its internal receive register, taking into account that the data are erased no sooner than 2.5 seconds after their reception. Within that period, the host must have fetched the data (i.e., pulled Host Ready to '0'), else they are lost.

When the host can be relied on to immediately fetch and process the received data, or if it has a sufficiently large receive buffer, it is safe to permanent tie the Host Ready input to ground using jumper JP1. In this way the ER400TRS is encouraged to immediately transmit received data. If, on the other hand, the host is not quick to process received data, it may control this module input via its serial interface and jumper JP1 in accordance with its timing requirements.

Finally, the ER400TRS puts analogue information regarding received signal strength at your disposal.

RSSI (received signal strength indicator)
The transceiver has a built-in RSSI (Received Signal Strength Indicator) that provides an analogue output voltage that is inversely proportional to the RF energy present within the pass band of the receiver. It ranges from 0 V (maximum signal, –50dBm) to 1.2 V (minimum signal, –105 dBm) and has a slope of approximately 50 dB/Volt. This analogue output signal should only be connected to a high impedance load (>100 kΩ) and can be used to provide a measure of the signal strength and any interfering signals (noise) within band during the installation and operation of systems.

The hardware links between the host(s) and the radio modules are depicted in **Figure 2**. Because the ER400TRS operates in half-duplex bi-directional mode, both Host A and Host B can alternately transmit and receive.

Easy Radio embedded software

The firmware running inside the ER400TRS module, and in particular the routines that look after the data transmission protocols, are proprietary, i.e., have been developed by LPRS themselves for their own products, hence do not reflect any kind of international standard. The software that does it all is remarkably simple, as well as easy to use from the outside. After all, when you are looking at 'just' a fast connection using simple microcontroller systems, no complex memory-hungry protocol stack is required like the one required by TCP/IP.

The embedded software of the ER400TRS is dubbed 'Easy Radio'. It handles three important functions:

1. parameter setting on the interface to the host (PC);
2. parameter setting on the RF parts;
3. executing the data transmission protocol.

The various parameters are loaded using ASCII command sequences listed in the Easy Radio software documentation. For example, the transmitter output power is stepped down to 5 mW by sending the command string `ER_CMD#P5` to the radio module. An overview is given in the text box 'Easy Radio Functionality'. On www.lprs.co.uk/download/ LPRS kindly supply a Windows-savvy software tool that allows all ER400TRs parameters to be given the desired values, and convey them to the transceiver module.

Host interface parameters
The serial asynchronous interface may be used at one of five different (but very common) baud rates (2,400 to 38,400 bits/s). Characters are invariably transmitted and received in UART format. This is only applicable to the relevant host, with the ER400TRS running at a factory-determined speed of 19,200 bits/s. Hence the data speed on the radio link will always be 19.2 kbits/s.

RF section parameter
Here users have the opportunity to select one of ten available radio channels (frequencies) between

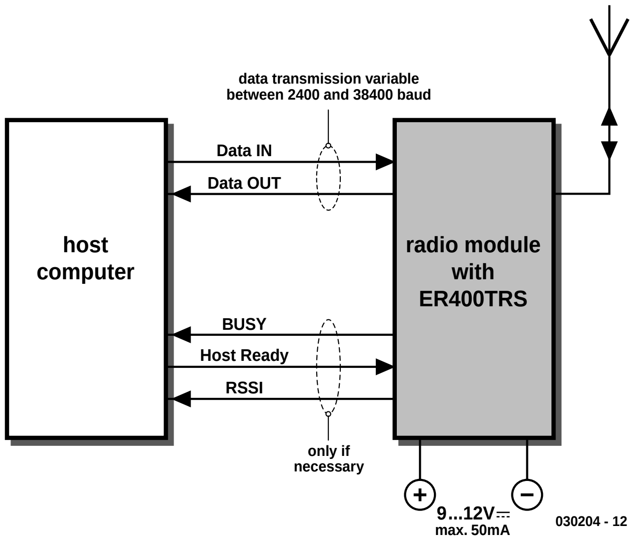


Figure 2. Basic configuration of a radio station for data transmission and reception.

433.25 MHz and 434.35 MHz, where the transmit and receive frequencies are always equal (i.e., no offset or shift).

The transmitter output power can be adjusted between 1 mW and 10 mW in ten steps. This may be useful to conserve battery energy when the module is used to cover a relatively small distance.

Data transmission protocol execution

The ER400TRS comprises a shared 128-byte transmit/receive buffer, via which the traffic is handled as follows:

First, the Host transmitter CPU checks if BUSY (= CTS on an RS232 link) is Low. Alternatively, it should idle a fixed time after the last transmission, until the ER400TRS flags that it has finished its internal tasks, thus allowing the host CPU to release its data. Next, the data are written into the module's internal buffer, where they remain latched.

The data are transmitted when either the buffer is full after continuous reception of 128 bytes (if more are sent to the chip on one go, the excess ones are discarded), or if a pause occurs in the datastream after a byte is conveyed where the pause length is greater than two byte lengths. In this way, individual bytes may be 'broadcast'.

Before the data leave the transmitter a CRC (cyclic redundancy check) is run to generate a CRC byte which is added to the 'message' bytes, together with a preamble and some more additional information (for example, the number of data bytes). Next, all bytes are Manchester-encoded and finally transmitted. During these activities, the BUSY line of the ER400TRS is logic High.

At the receiver side the data are Manchester-decoded, the extra information is separated and an error check is performed. During these activities, the BUSY line of the ET400TRS is logic High. In case data corruption is detected, all data is rejected and the receiving host does not get any data.

However, the transmitting host also does not get acknowledge information about the rejected 'telegram'. If it is required for the host to have confirmation that all data has been correctly received, then an acknowledge process has to be implemented using the ER400TRS firmware.

If all data has been received okay, the data are copied to the transmit/receive buffer in the receiving ER400TRS. Next, the chip waits for the receiving host to pull Host Ready Low (= RTS in RS232 terms). Following this the data are automatically and continuously fed to the host via the serial link. If this type of handshaking is not required, the Host Ready input may be permanently tied to ground using jumper JP2.

In case the Host Ready signal is not pulled low within 2.5 s from reception of the data, all received data are wiped from the buffer and the ER400TRS is ready for transmission or reception.

The data processing inside ER400TRS modules is invisible to the two hosts — all they need to do is supply and receive data via their RS232 ports.

Antennas

No specialist knowledge is required on RF technology if you want to set up SRD-based data link like the one described in this article. There's one exception, though: the antenna.

The transceiver can be used with the various common types of antenna that match the 50 Ω RF Input/Output such as a whip, helical or PCB/Wire loop antennas — see **Figure 3**.

Whip antennas are resonant with a length corresponding to one quarter of the electrical wavelength (λ/4). They are very easy to implement and can simply be a 'piece of wire' or PCB track, which at 433 MHz should be about 15.5 cms in length. This should be straight, in 'free space' (kept well away from all other circuitry) and should be connected directly to the Antenna pin of the transceiver. If the antenna is remote, it should be connected via a 50-Ω coaxial feeder cable (RG58U, RG174U). The quarter-wave antenna has the widest range of the three types dis-

Easy Radio Functionality

The embedded microcontroller looks after the following Easy Radio functions:

- processing input and output data using a standard UART format (1 start bit, 8 databits, no parity, 1 stop bit)
- Manchester encoding/decoding of data for/from radio link
- Calculation and comparison of CRC checksum
- Radio protocol monitoring and control: transmission of preamble and sync bytes, removal of these bytres when receiving
- Programming of receive/transmit synthesizer for channel selection
- RF output power programming
- Operation of the external host interface at the desired baud rate
- Storage of up to 128 bytes for receive/transmit buffer
- Operation of the two handshaking signals
- Operational parameter back up in on-board EEPROM

cussed here, but only if it ‘sees’ a sufficiently large ground plane beneath it. Its disadvantage is mainly mechanical vulnerability and sensitivity to large metal objects or surfaces in its vicinity. It is also rather cumbersome in semi-portable equipment.

PCB Loop antennas are the most compact antennas but are less effective than the other types. They are also more difficult to design and must be carefully ‘tuned’ for best performance. At 433 MHz the loop should cover an area of 400-1000 mm^2 and be tuned to resonance using a small (1.5-5 pF) capacitor. The PCB track acting as the loop may be as narrow as 1 mm. The feeder point should be at 15-25% of the overall length of the loop.

Helical antennas are also resonant and generally chosen for their more compact dimensions. They are more difficult to optimise than quarter wave antennas and are critical with regard to surrounding objects that can easily ‘de-tune’ them. They operate most efficiently when there is a substantial ground plane for them to radiate against. For our project, a helical antenna can be made from 0.5-mm dia. (26 SWG) enamelled copper wire (ECW). Suitable dimensions are:
17 turns, 5 mm internal diameter, stretch to 34 mm length;
24 turns, 3.2 mm internal diameter, stretch to 19 mm length.

As to range and susceptibility to

An Application using the Elektor 80C537 Board

Two built up radio modules are needed to realise the example application depicted in **Figure A**. At one side we find Host A, an 8051 controller system based on the 1997 Elektor Electronics 80C537 Microcontroller Board connected to an SRD radio module via the second serial interface of the 80C537 micro. A MAX232 level converter is not required here. On the radio module, the relevant pins are simply interconnected with wires (pin 11 to pin14; pin 12 to pin 13). The development PC allows the software for the 80537 to be set while also acting as a display device for the received characters.

The other party is formed by Host B, a regular PC connected to a radio module via its serial port. At this side of the radio link, a MAX232 level converter is required. The PC initially runs an ordinary terminal emulation program (like HyperTerminal) just to display received characters and to send out (individual) characters.

Different applications were realised using this configuration, and the relevant software (C51 for the 80537 and Visual BASIC for the PC/Notebook) may be downloaded free of charge from the Elektor Electronics webserver at www.elektor-electronics.co.uk/dl/dl.htm (select this issue).

1. Individual key presses are transmitted by Host B, received by the 80C537 system and get displayed on the monitor of the development PC.

2. Host A sends individual characters or whole text strings — these appear on the monitor connected to Host B.

3. The 80C537 system is turned into a small weather station by extending it with sensors and/or actuators and an LC display. It employs the radio link to convey to Host B meteorological data like air pressure, temperature, relative humidity, etc. complete with time stamps. Host B runs a small Visual BASIC program displaying received data and writing an Excel compatible (log) file for inspection/evaluation at a later time. The system also allows texts to be sent from Host B to the LC display connected to the 80C537 system. **Figure B** shows a screendump of the program running on Host B. The hardware should also allow the control (by radio) of just about any actuator at the ‘remote’ site.

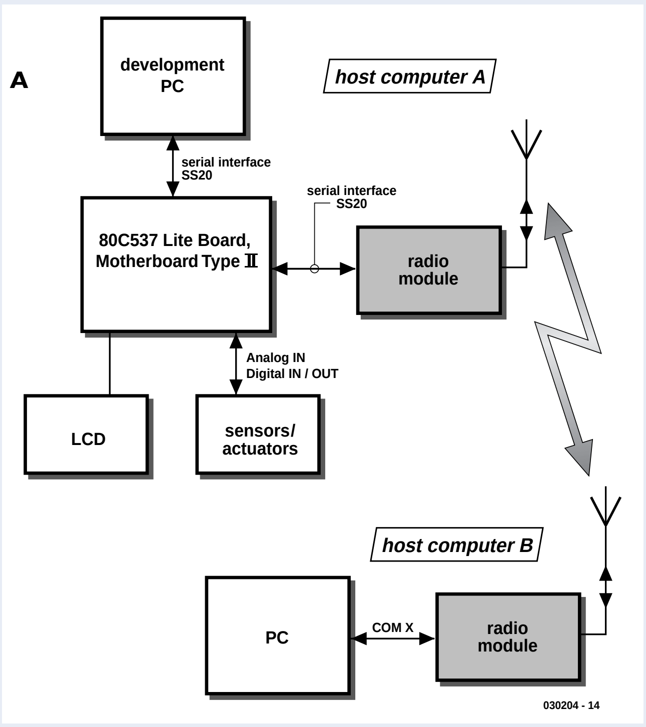
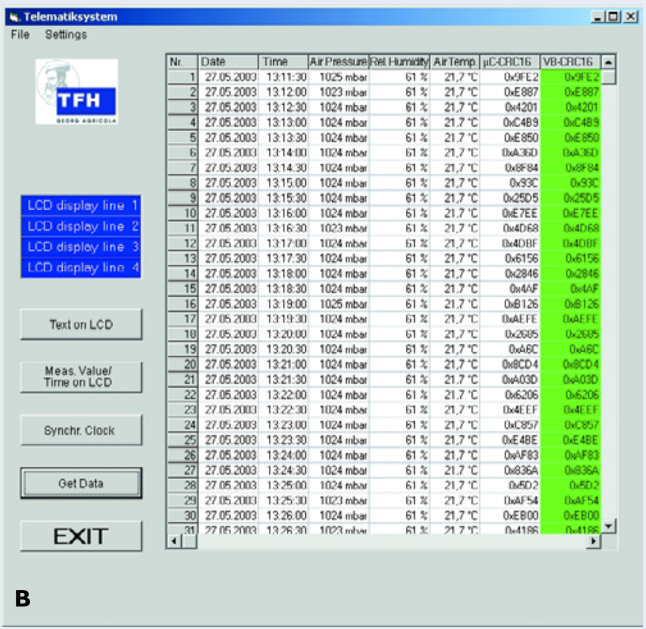


Figure A. A PC and a microcontroller system communicating over a radio link.



Nr.	Date	Time	Air Pressure	Rel. Humidity	Air Temp.	µC-CRC16	VB-CRC16
1	27.05.2003	13:11:30	1025 mbar	61 %	21,7 °C	0x9FE2	0x9FE2
2	27.05.2003	13:12:00	1023 mbar	61 %	21,7 °C	0xE887	0xE887
3	27.05.2003	13:12:30	1024 mbar	61 %	21,7 °C	0x4201	0x4201
4	27.05.2003	13:13:00	1024 mbar	61 %	21,7 °C	0xC4B9	0xC4B9
5	27.05.2003	13:13:30	1024 mbar	61 %	21,7 °C	0xE850	0xE850
6	27.05.2003	13:14:00	1024 mbar	61 %	21,7 °C	0xA36D	0xA36D
7	27.05.2003	13:14:30	1024 mbar	61 %	21,7 °C	0x8F84	0x8F84
8	27.05.2003	13:15:00	1024 mbar	61 %	21,7 °C	0x93C	0x93C
9	27.05.2003	13:15:30	1024 mbar	61 %	21,7 °C	0x25D5	0x25D5
10	27.05.2003	13:16:00	1024 mbar	61 %	21,7 °C	0xE7EE	0xE7EE
11	27.05.2003	13:16:30	1023 mbar	61 %	21,7 °C	0x4D68	0x4D68
12	27.05.2003	13:17:00	1024 mbar	61 %	21,7 °C	0x4DBF	0x4DBF
13	27.05.2003	13:17:30	1024 mbar	61 %	21,7 °C	0x6156	0x6156
14	27.05.2003	13:18:00	1024 mbar	61 %	21,7 °C	0x2846	0x2846
15	27.05.2003	13:18:30	1024 mbar	61 %	21,7 °C	0x4AF	0x4AF
16	27.05.2003	13:19:00	1025 mbar	61 %	21,7 °C	0xB126	0xB126
17	27.05.2003	13:19:30	1024 mbar	61 %	21,7 °C	0xAEFE	0xAEFE
18	27.05.2003	13:20:00	1024 mbar	61 %	21,7 °C	0x2605	0x2605
19	27.05.2003	13:20:30	1024 mbar	61 %	21,7 °C	0xA6C	0xA6C
20	27.05.2003	13:21:00	1024 mbar	61 %	21,7 °C	0x8CD4	0x8CD4
21	27.05.2003	13:21:30	1024 mbar	61 %	21,7 °C	0xA03D	0xA03D
22	27.05.2003	13:22:00	1024 mbar	61 %	21,7 °C	0x6206	0x6206
23	27.05.2003	13:22:30	1024 mbar	61 %	21,7 °C	0x4EEF	0x4EEF
24	27.05.2003	13:23:00	1024 mbar	61 %	21,7 °C	0xC857	0xC857
25	27.05.2003	13:23:30	1024 mbar	61 %	21,7 °C	0xE4BE	0xE4BE
26	27.05.2003	13:24:00	1024 mbar	61 %	21,7 °C	0xAF83	0xAF83
27	27.05.2003	13:24:30	1024 mbar	61 %	21,7 °C	0x836A	0x836A
28	27.05.2003	13:25:00	1024 mbar	61 %	21,7 °C	0x5D2	0x5D2
29	27.05.2003	13:25:30	1023 mbar	61 %	21,7 °C	0xAF54	0xAF54
30	27.05.2003	13:26:00	1024 mbar	61 %	21,7 °C	0xEB00	0xEB00
31	27.05.2003	13:26:30	1023 mbar	61 %	21,7 °C	0x4186	0x4186

Figure B. Screendump of the weather station software running on Host B.

Literature and Internet addresses:

[1] LPRS: www.lprs.co.uk

[2] ER400TRS hardware manual:
http://www.lprs.co.uk/main/viewdatasheet.php?datasheetref=112

[3] Easy Radio software manual:
http://www.lprs.co.uk/pdf_directory/23861055851852.pdf

[4] 80C537 Microcontroller Board, Elektor Electronics June 1997.

[5] Portal website for the SRD industry and radio regulation authorities:
www.lpra.org

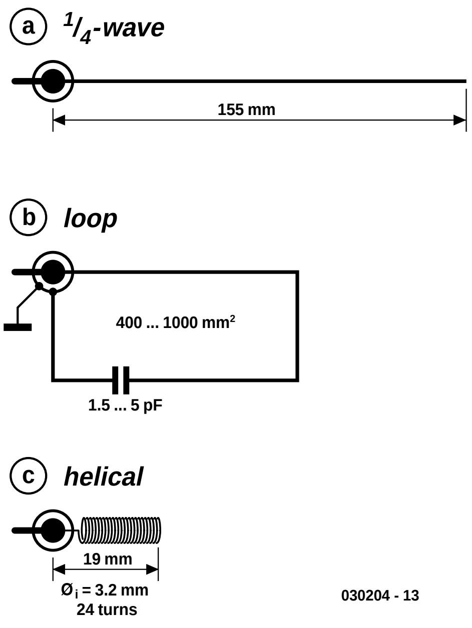


Figure 3. Pick your antenna.

noise, the performance of the helical antenna lies roughly between that of the whip and the loop. It is, however, the least space consuming of the three. Moreover, it is easily tuned by stretching and compressing the coil until the greatest range is obtained.

PCB and practical use

The printed circuit board for one radio module (**Figure 4**) is double-sided hence very compact. Populating the board should not present problems if you pay due attention to the correct fitting of the polarized parts.

To close off this article, we should reiterate that the use of SRDs is governed by strict regulations. On request, LPRS supply customers with a copy of the CE compliance document for their SRD modules, as well as pointers to legal information relevant to the use of SRDs in several countries.

(030204-1)

COMPONENTS LIST

Resistors:
R1-R4 = 1kΩ8

Capacitors:
C1,C3,C9 = 100nF
C2,C4 = 10μF 16V radial
C5-C8 = 1μF 16V radial

Semiconductors:
D1 = 1N4148
D2,D3,D5 = LED, green, 3mm, low current
D4 = LED, red, 3mm, low current
IC1 = 7805CP
IC2 = MAX232CP
IC3 = ER400TRS from LPRS (see text)

Miscellaneous:
JP1 = jumper with 2-way pinheader
JP2 = jumper with 3-way pinheader
K1,K3 = 2-way pinheader
K2 = 9-way sub-D socket (female), angled pins, PCB mount
K4 = BNC socket, 50Ω, PCB mount (Farnell # 365-0558)
RS232 cable (non crossing)
PCB, order code **030204-1** (see Readers Services page)

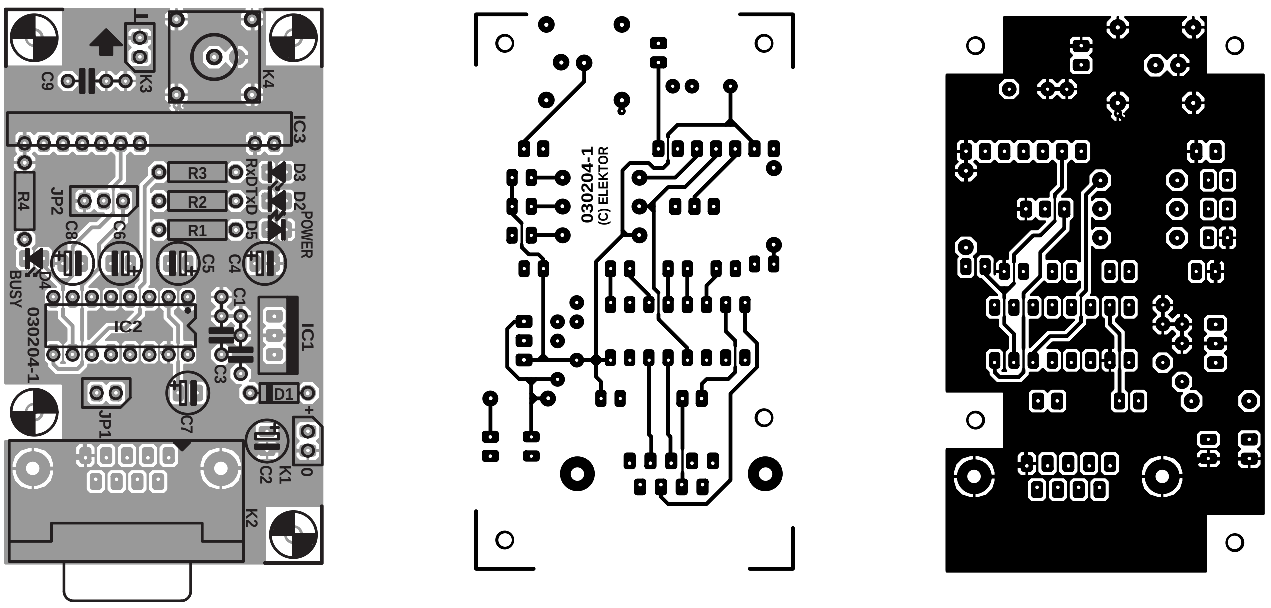


Figure 4. The PCB for the Wireless RS232 Link project is double-sided and easy to build up (board available ready-made).