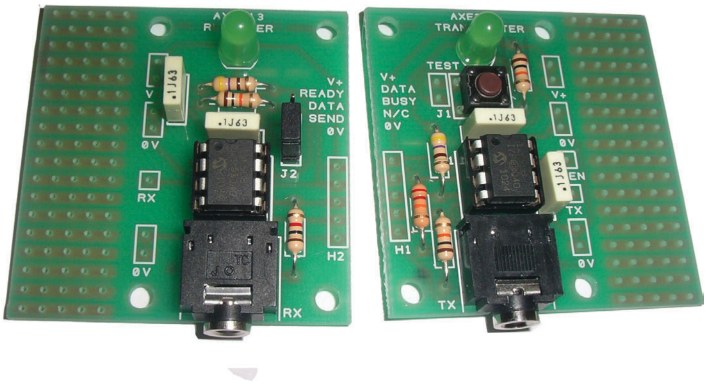
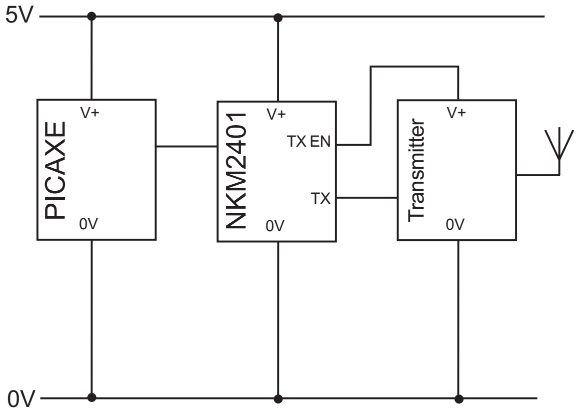
PICAXE RF Connect Kit (AXE213)

Kit Contents:

PCB	AXE213 Transmitter & Receiver PCB Pair
R1-3	10k resistor (brown black orange gold)
R4-5	470 resistor (yellow violet brown gold)
R6	22k resistor (red red orange gold)
R7	1k resistor (brown black red gold)
C1-4	100nF capacitor
LED1-2	LEDs
SW1	6mm push Switch
J1-2	jumper links
J1	2 pin header * *(optional)*
J2	3 pin header *
H1-2	5 pin header * *(optional)*
CON1-2	3.5mm Download socket
IC1-2	8 pin IC socket
IC1-2	NKM2401 IC *(PIC12F508 preprogrammed with NKM2401 firmware)*

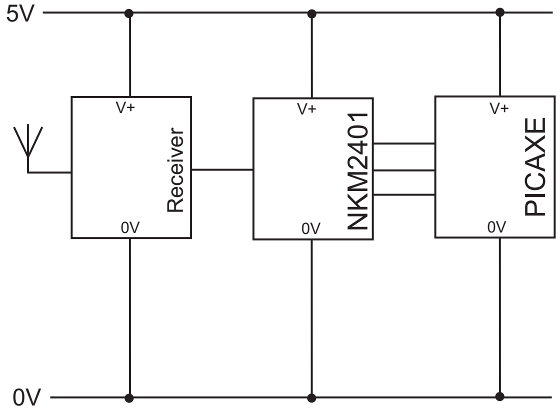


* *supplied as 2 off 10 pin headers which can be snapped to length*



AXE213 Transmitter Board Block Diagram

AXE213 Receiver Board Block Diagram



Overview:

The AXE213 is a pair of boards which allow simple PICAXE to PICAXE (or PICAXE to PC) interfacing using low cost 'license exempt' AM and FM radio modules (e.g part RFA001, purchase separately). The special encoder/decoder chip on the boards provide a much more reliable serial wireless communication system than that which can be achieved when using serin/serout commands with the radio modules connected directly to the PICAXE chip.

Please download the full assembly instructions and datasheet from this web link:

www.rev-ed.co.uk/docs/axe213.pdf

Contents

Description:

The AXE213 is a pair of boards which allow simple PICAXE to PICAXE (or PICAXE to PC) interfacing using low cost 'license exempt' radio modules (e.g part RFA001, purchase separately). The special encoder/decoder chip on the boards provide a more reliable serial wireless communication system than that which can be achieved when connecting the radio modules to the PICAXE chip directly.

The transmitter board, when fitted with an AM or FM transmitter module, allows a PICAXE or PC to send data to the receiver board (which must be fitted with a matching receiver module). The received data is buffered and stored, and then can be read by a PICAXE and/or a PC. Both ASCII data and raw byte data can be sent across the wireless link.

The boards can also be used to create a wireless LCD display, where the AXE033 Serial LCD is directly controlled to the receiver board without any PICAXE microcontroller.

Both transmitter and receiver boards are compatible with the full range of PICAXE controllers.

The transmitter board is fitted with a test button to allow standalone testing. The receiver has a number of operational modes which can be configured as required.

Each transmitter and receiver board includes a 3.5mm jack socket which can be used with AXE027 and AXE026 PICAXE download cables to connect the board directly to a PC for testing purposes. Solder pads allow direct connection to the PICAXE project. Both boards incorporate an LED which indicates when data is being transmitted and when data has been received.

To provide maximum flexibility, and cater for the wide range of wireless modules available and their differing pin-outs, both boards have a 'prototyping area' to which the wireless modules can be soldered. This allows almost any module to be used with the boards. If you do not already have radio modules part RFA001 is a low-cost matched pair of 433MHz AM modules which may be used with these boards.

The boards are designed around the NKM2401 Radio Data Encoder/Decoder IC (developed by Nikam Electronics) which is used to for both transmitting and receiving data. The NKM2401 'Manchester encodes' all data and automatically sends out the required RF preamble, data, checksum etc. so the user is left to simply send bytes of data and receive bytes of data. Data can be reliably sent with just a single 'serout' command and received with just a single 'serin' command.

Because the NKM2401 provides preamble, Manchester Encoding and uses CRC error checksum codes, the wireless link will be more reliable and give greater range than a direct 'serout' control of a wireless module.

Overall data throughput can be up to 60 bytes per second. This includes the time taken to send data from a PICAXE to the transmitter, the time to send data 'over the air' plus the time taken to pass received data to the receiving PICAXE.

Notes

The wireless connection established using an AXE213 transmitter and receiver board pair is one-way (simplex) system - from transmitter to receiver. There is no return path for data or acknowledgement, although this could be achieved by using two pairs of boards.

The communications is also 'point-to-multi-point'; anything sent by a transmitter will be received by all receivers. Only one transmitter should be operated at a time or communications may be unreliable and one transmitter may 'drown-out' another. The transmitter board has a 'transmit enable' connection which should be used to enable the wireless transmitter module only when data is to be sent. This minimises interference and complies with good radio etiquette.

Most wireless modules operate in the unlicensed radio spectrum so there is no guarantee of absolutely reliable operation at all times and communications may be affected by other transmitters which may be operating on the same frequencies in the vicinity and by other interference.

The 3.5mm jack sockets on the boards are serial connections and not download connection ports – it is not possible to reprogram the NKM2401. No damage will be done if an accidental attempt is made to download a PICAXE program into either board.

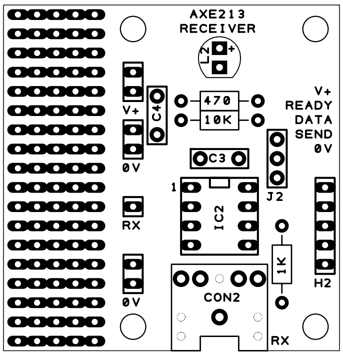
The NKM2401 is a pre-programmed Microchip PIC12C508A and is labelled as such. Take care not to confuse it with any other PIC you may have or with a PICAXE-08 or 08M. The 12C508A is not reprogrammable.

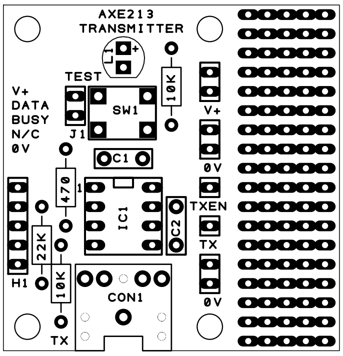
.

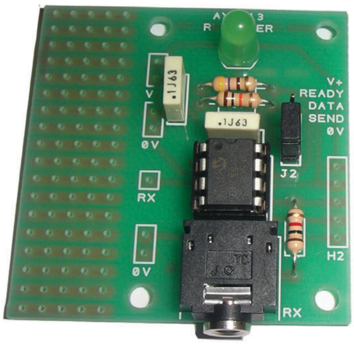
Assembly Instructions:

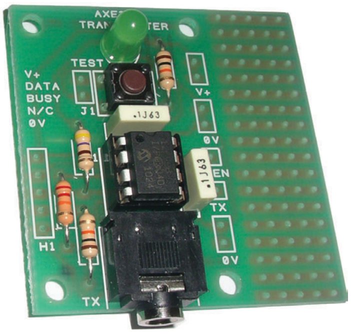
1. Snap the two PCBs apart. Solder the resistors in position.
2. Solder the IC socket, download socket and switch in position.
3. Solder the capacitors (not polarised) and 2 LEDs in position, ensuring correct polarity of the LEDs (align flat edge with diagram on PCB).
4. Snap a 3 pin length of the header strip. Solder in position J2. Optionally repeat with a 2 pin length in jumper position J1. If not using direct wire connections, solder 2 off 5 pin lengths in positions H1 and H2.

6. Insert the microcontrollers into the sockets, ensuring correct orientation of pin 1 on both chips.









Connecting a Transmitter Module

The transmitter board has a number of solder pads adjacent to the prototyping area to which the wireless transmitter module will be attached.

V+ The board power supply which connects to the wireless transmitter module to keep it permanently powered.

TXEN When the transmitter module has a transmit enable pin, use this connection to enable the transmitter.If the transmitter module does not have a transmit enable pin, use this pin to power the module (instead of V+). This pin has 20mA current capability.

TX The data signal to be transmitted from the NKM2401 to the transmitter module

0V The power supply 0V connection

If the wireless transmitter module does not have separate power and enable pins it is not recommended to power the module from V+, as this will keep it continuously transmitting which may interfere with other radio users. In these situations it is recommended to power the module from the TXEN connection so it is only powered when required.

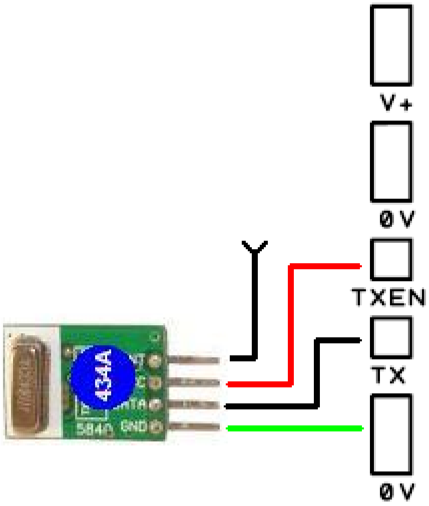
Notes

The wireless transmitter module must be capable of operating at 5V. The transmitter board does not directly support transmitter modules which require greater than 5V to operate.

The TXEN pin can supply 20mA of current for a wireless transmitter module. Exceeding this limit may cause permanent damage to the NKM2401. If more current is required for a wireless module (or it is required to operate the module from a supply greater than V+) then an appropriate 'transistor power switch' controlled by the TXEN pin must be implemented.

Example of Transmitter Module Connections

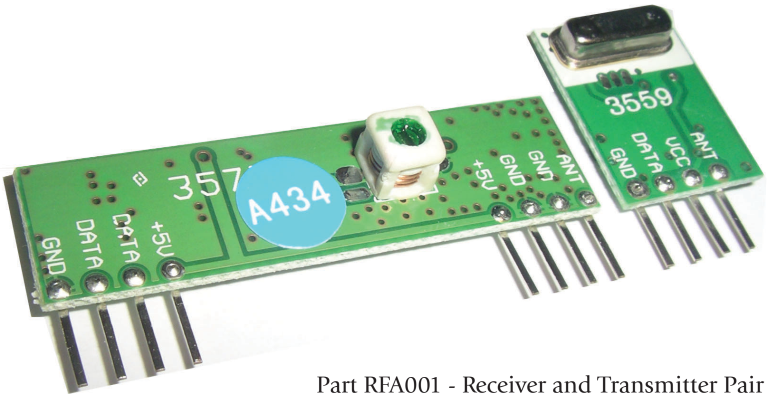
The following shows a typical connection of a transmitter module to the transmitter board connection points. The transmitter module pictured is as supplied in part RFA001.



On this transmitter the pin connections are physically printed on the transmitter module PCB; if the pins on your unit are not labelled you will have to consult the manufacturer's datasheet to determine the pin-out.

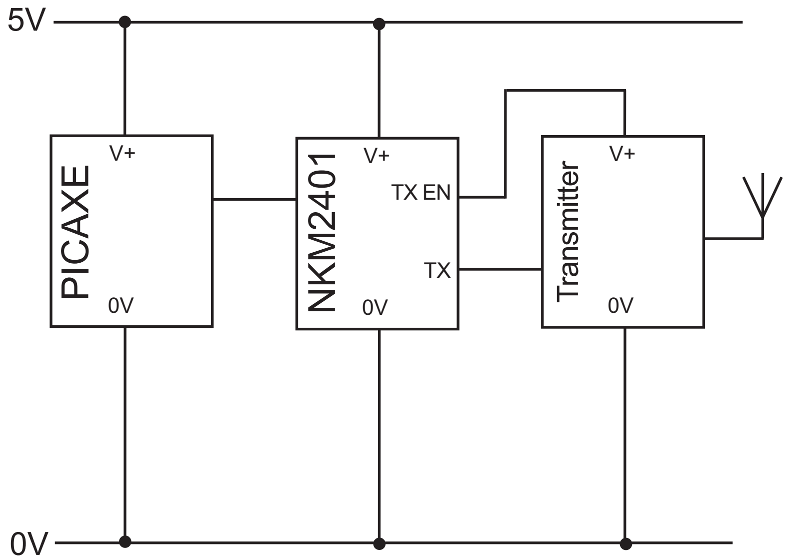
Note that the VCC for the transmitter module is connected to TXEN rather than V+. This means the transmitter module will only be powered when it needs to be and is recommended practice.

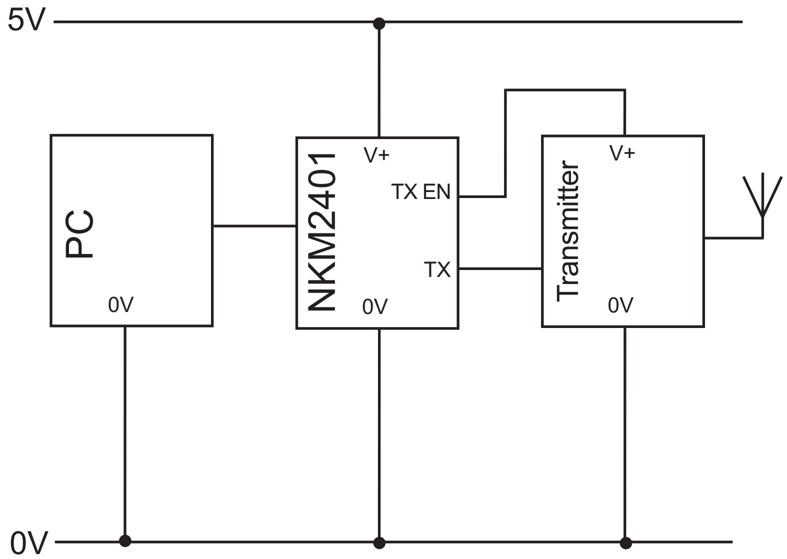
The transmitter module may be soldered directly to the transmitter board 'prototyping area' or suitable sockets may be soldered into which the module is then inserted. Connections between module or socket and transmitter board connection points are made with wire links.



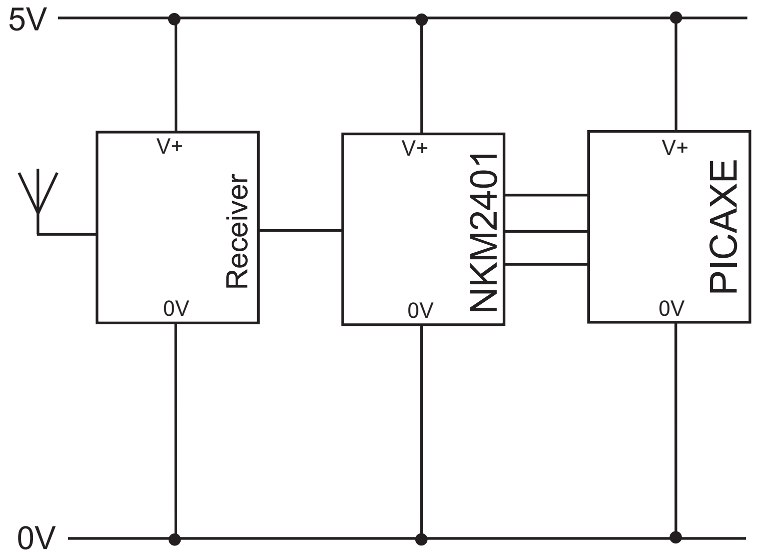
Part RFA001 - Receiver and Transmitter Pair
(purchase separately)

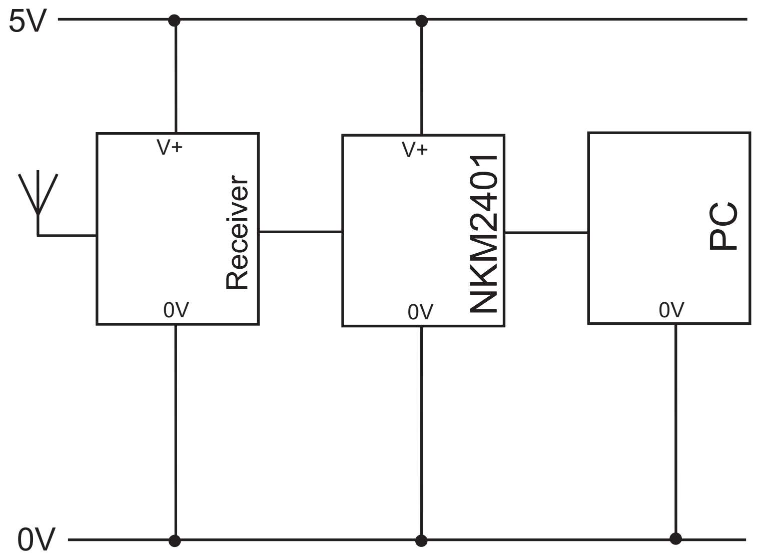
Transmitter Board Block Diagrams

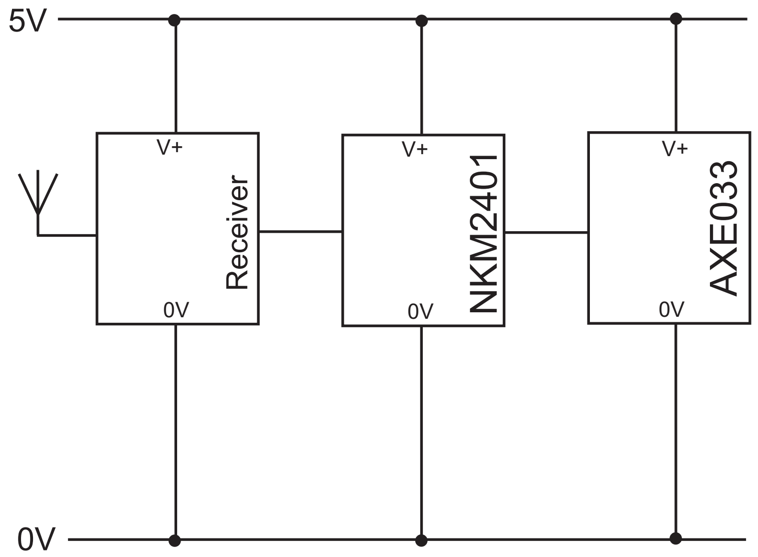




Receiver Board Block Diagrams







Connecting a Receiver Module

The receiver board has a number of solder pads adjacent to the prototyping area to which the wireless receiver module will be attached –

V+ The board power supply which connects to the wireless receiver module to keep it permanently powered

RX The data signal received by the receiver module and transferred to the NKM2401chip.

0V The power supply 0V connection

Note

The wireless receiver module must be capable of operating at 5V. The receiver board does not directly support receiver modules which require greater than 5V to operate.

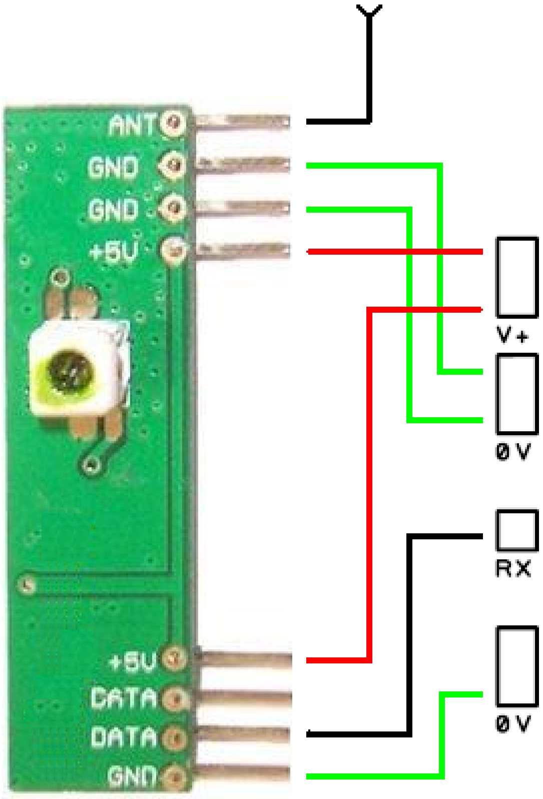
Example of Receiver Module Connections

The following shows a typical connection of a receiver module to the receiver board connection points. The receiver module pictured is as supplied in part RFA001.

In this example the pin connections are printed on the receiver module PCB; if they are not you will have to consult the manufacturer's datasheet to determine the pin-out.

Note that some receiver modules may have more than one DATA output. These may be 'digital' and 'analogue' and it will be necessary to determine which to connect to the receiver board. Some published circuits using the receiver module chosen may show both DATA outputs connected together; this is not always correct or most appropriate and some experimentation may be required to determine which DATA output to use. With the module supplied in part RFA001 only use one connection as shown.

The receiver module may be soldered directly to the receiver board 'prototyping area' or suitable sockets may be soldered into which the module is then inserted. Connections between module or socket and receiver board connection points are made with wire links.



Aerial Choice and Connection

Most wireless modules will require an aerial (also known as an antenna) to be connected. Aerials should be connected to the appropriate wireless module pin by soldering to the prototyping area.

There are a wide range of aerials which can be used with wireless modules which may be as simple as a single length of wire, single core or stranded. It is most common that ¼ wavelength aerials are used but other wavelength aerials may be considered; the advantage of single wire ¼ wavelength aerials is that they are relatively short so easy to manage. For 433MHz operation an appropriate aerial length of a single wire would be 173.2mm and for 868MHz operation the length would be 86.4mm. On the AXE213 take into account the length of PCB track from antenna to the module for optimum performance, but satisfactory results can usually be obtained even if the aerial length is not perfectly optimal.

Alternatively a short wire can be taken to a 'stubby' or 'whip' aerial designed for 433MHz or 868MHz operation. These may provide a more robust and aesthetically pleasing solution than a simple length of wire.

Appropriate connectors and external aerials may also be used with the boards.

Aerial Length Calculation

The length of aerial of a quarter wavelength for a particular operating frequency can be calculated as follows –

length of aerial (mm) = 75000 / frequency (MHz)

e.g. for 433MHz modules: *75000/433 = 173.2 mm*

The following tables gives the length of aerials for 433MHz and 868MHz operation

433 MHz

1/4 Wavelength	173.2 mm	6.82 inches
1/2 Wavelength	346.4 mm	13.64 inches
5/8 Wavelength	433.0 mm	17.04 inches
3/4 Wavelength	519.6 mm	20.45 inches
Full Wavelength	692.8 mm	27.27 inches

868 MHz

1/4 Wavelength	86.4 mm	3.40 inches
1/2 Wavelength	172.8 mm	6.80 inches
5/8 Wavelength	216.0 mm	8.50 inches
3/4 Wavelength	259.2 mm	10.20 inches
Full Wavelength	345.6 mm	13.60 inches

Transmission Distance

The maximum transmission distance achievable will depend on operating frequency, the transmitting module's output power, the receiving module's sensitivity, appropriateness and quality of aerials, power supply, radio interference levels and other environmental conditions.

Maximum output power of a transmitter will usually be dictated and limited by local legislation and set by an appropriate radio authority in the region. Output power of transmitters and receiver sensitivity will normally be included in the wireless module's datasheet. It is the responsibility of the user to ensure that the modules selected comply with and are operated in accordance with local laws and regulations.

Unobstructed line of sight (LOS) transmission distances outside will usually be greater than distances achieved indoors where there may be internal reflections of radio signals and the signal passes through internal walls and other radio signal reducing materials. Some materials may attenuate the radio signal strength and reduce range, including human bodies, while some materials may completely block radio signals. Metal supports, foil-lined walling and other building materials are things to consider when installing a wireless link.

Operation within electrically noisy environments may interfere with transmissions. Operating near motors, power supplies, TVs and computer monitors and other radio transmitters may cause intermittent loss of data or complete loss of communications.

If the transmitter or receiver boards are operated from 4.5V battery supplies with wireless modules specified for 5V use then operational range may be reduced.

Minimum Separation

As well as a maximum transmission range over which radio signals can be reliably received there is often a minimum distance which may be needed between transmitter and receiver. Usually this at least 1 metre. If the transmitter is placed too close to the receiver the transmitted signal may swamp the receiver and it may be unable to determine the actual data sent.

If the wireless connection between a transmitter and receiver does not appear to be working then try with a greater distance between them. Moving both transmitter and receiver may take them out of signal 'dead spots'.

Do not connect the transmitter module aerial output to the receiver module aerial input directly (e.g. by joining both aerials) as this may cause permanent damage to the modules.

Transmitter Board - PICAXE and PC Connections

The transmitter board has a five way row of solder pads (H1) to which power and PICAXE connections are made. The pin-out, with the board held so the 3.5mm jack socket is to the bottom of the board, and the connections to the left, is as follows from top to bottom –

1 V+ 5V supply into board
Do not connect a 9V battery or greater than 5V
2 DATA Data to be transmitted (from a PICAXE output pin to the transmitter board)
3 BUSY An optionally used ‘transmitter is busy’ signal (from transmitter board to a PICAXE digital input pin). The signal is low when the transmitter is not busy and high when the board is busy and transmitting data. Do not send data when the BUSY line is high.
4 N/C Not connected
5 0V 0V power supply connection

The DATA signal (pin 2 of header H1) is switched through the 3.5mm jack socket. When no cable is inserted the DATA signal is connected to the NKM2401 IC as expected. When a cable is inserted the DATA signal is automatically disconnected, as the data comes from the cable instead. Therefore always remove the cable when not in use.

This arrangement means that data from a PC (via the cable) will override data sent from a PICAXE; this can be useful for testing. The jack socket is compatible with use of the AXE027 and AXE026 PICAXE download cables. Note the 3.5mm jack socket is a serial connection and not a download port – it is not possible to reprogram the NKM2401. No damage will be done if an accidental attempt is made to download a PICAXE program into the NKM2401.

The DATA input is configured for PICAXE N2400 communication and for 2400,N,8,1 communications from a PC. The input is fitted with a 10k/22k interface and supports both RS232 levels and TTL levels. No voltage level translation or polarity inversion is required.

The optionally used BUSY signal is low when the transmitter is not busy (data can be sent to the board) and high when the board is busy transmitting. When connecting the BUSY output to a PICAXE operating at 3V (or below the board’s V+ supply) the signal should be current limited by an in-line 1k resistor.

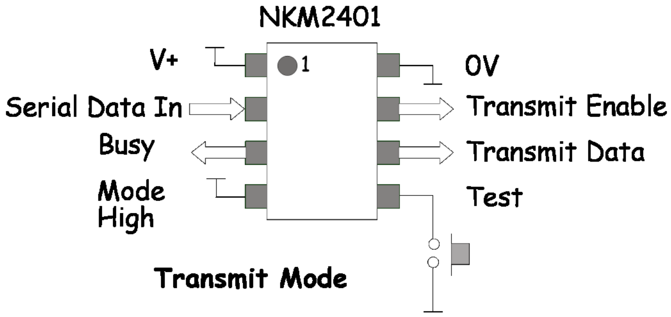
To transmit data it is simply a matter of checking the board is currently capable of receiving data (by checking the BUSY line is low or simply waiting an appropriate time, at least 130ms), and then sending eight bytes of data to the board. Those eight bytes of data will then be encoded and transmitted.

The rate at which bytes are sent to the board does not matter; eight bytes will be buffered up and when all eight have been received they will be transmitted. If less than eight bytes are to be sent then additional ‘padding bytes’ must be used to make up the eight bytes. If more than eight bytes are to be sent they must be sent as multiple eight byte chunks and it may be necessary to use one or more of those bytes to indicate which packets are which.

If more than eight bytes of data are sent when the transmitter is not ready to receive the bytes may be lost. Ensure that the transmitter is ready to receive data bytes before sending them.

The transmitter board's LED will flash while data is being transmitted. The LED on the receiver board should also flash as it receives the data sent.

NKM2401 Transmitter Configuration



Test Button

The transmitter board is fitted with a test button which causes an eight byte test message to be transmitted every half second whilst the button is pushed.

This is useful for checking receiver operation and, when the transmitter board is powered from a portable supply, in performing range testing.

The test message is the eight ASCII characters "Test $*xx*" where '$*xx*' is an incrementing hexadecimal number from $00 to $FF.

A jumper link (J1) can be fitted in parallel with the push button which will have the effect of permanently enabling the test message. This is equivalent to permanently holding down the test switch.

PICAXE Transmitter Example – Without Using BUSY signal

```
; DATA to transmitter board from Output Pin B.7
; BUSY from transmitter board not used, simply use delay between
; transmissions

main:
      pause 500                          ; ensure ready to send
      serout B.7, N2400, ( "12345678" )  ; send eight bytes of data
      goto main                          ; repeat
```

PICAXE Transmitter Example – Using BUSY signal

```
; DATA to transmitter board from Output Pin B.7
; BUSY from transmitter board to Input Pin C.1, low when ready to send

main:
      if pinC.1 = 1 then main            ; wait while busy
      serout B.7, n2400, ( "12345678" )  ; send eight bytes of data
      goto main                          ; repeat
```

PICAXE Transmitter Example – Byte at a Time

Data does not need to be sent in a single 'serout' command but can be split across a number of 'serout' commands. After eight bytes have been sent to the transmitter board it will then transmit them to the receiver.

```
; DATA to transmitter board from Output Pin B.7
; BUSY from transmitter board not used, high when busy, low when not busy

main:
      pause 500                          ; ensure ready to send
      serout B.7, n2400, ( "1" )         ; send first byte of data
      serout B.7, n2400, ( "234" )       ; send three bytes of data
      serout B.7, n2400, ( "56" )        ; send two bytes of data
      serout B.7, n2400, ( "78" )        ; send final two bytes of data
      goto main                          ; repeat
```

Receiver Board - PICAXE and PC Connections

The receiver board has a five way row of solder pads (H2) to which power and PICAXE connections are made. The pin-out, with the board held so the 3.5mm jack socket is to the bottom of the board, and the connections to the right, is as follows, from top to bottom –

1 V+ 5V supply into board
Do not connect a 9V battery or greater than 5V
2 READY An optional 'data is ready' signal (from the receiver board to a PICAXE digital input pin). This signal is active low – it goes low to indicate data has been received and is currently buffered.
3 DATA Received data (from the board to a PICAXE digital input pin).
4 SEND An optional 'send data' signal (from a PICAXE output pin to the board). This signal should be set high to enable the board to send its buffered data to the PICAXE
5 0V 0V power supply connection

The DATA signal passed to the PICAXE input pin is simultaneously passed to the 3.5mm jack socket (RX). This allows a PC to monitor what the PICAXE is receiving. Both cable and PICAXE receive the same data at the same time. The jack socket is wired as per a standard PICAXE download and serial connection and is compatible with use of the AXE027 and AXE026 cables.

Note the 3.5mm jack socket is a serial connection and not a download port – it is not possible to reprogram the NKM2401. No damage will be done if an accidental attempt is made to download a PICAXE program into the NKM2401.

The DATA output from the board is TTL level which is usable with PICAXE and AXE027 cable and should be usable with most RS232 serial port interfaces. The output is configured for PICAXE N2400 reception and 2400,N,8,1 communications with a PC. When connecting the DATA output to a PICAXE operating at 3V (or below the board's V+ supply) the DATA line should be current limited by an in-line 1k resistor.

The receiver board's LED will light when data has been received and buffered, and will stay lit until all data bytes buffered have been sent to the PICAXE or PC. In most cases the LED will be seen to flash as data is received. If the PICAXE is accepting bytes one at a time from the receiver at a slow rate the LED will remain on for longer. If the LED stays permanently lit this may be an indication that the PICAXE has not recognised that data has been received or is not requesting the buffered data.

Receiver Configuration Options

The receiver board has a three-pin header (J2) which is used to select three modes of operation.

With the 3.5mm jack socket to the bottom of the board, and header pins to the right of the chip, the modes of operation are as follows –

One-Wire Mode
Link bottom two pins of jumper J2
All data received is passed automatically to the PICAXE or PC as soon it is received.
Require ones PICAXE input pin.

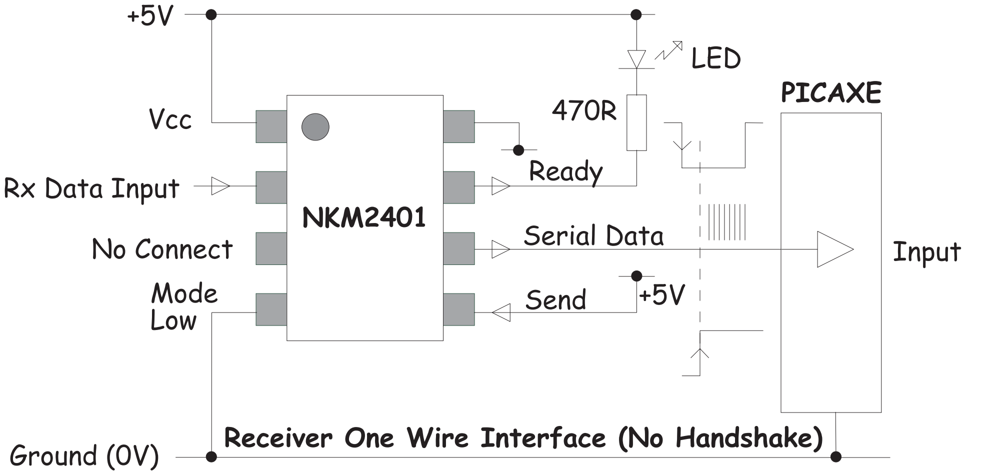
Three-Wire Mode
No link fitted on jumper J2
All data received is internally buffered. The READY pin is set low to indicate that data is available and the buffered data will only be passed to the PICAXE (and PC) when the SEND line from the PICAXE is set high.
Requires two PICAXE inputs and one PICAXE output.

Two-Wire Mode
Link top two pins of jumper J2
This is similar to the Three-Wire Mode but uses a single bi-directional I/O line to read the READY pin signal and control the SEND line
Requires one PICAXE input and one PICAXE bidirectional (input/output) pin.

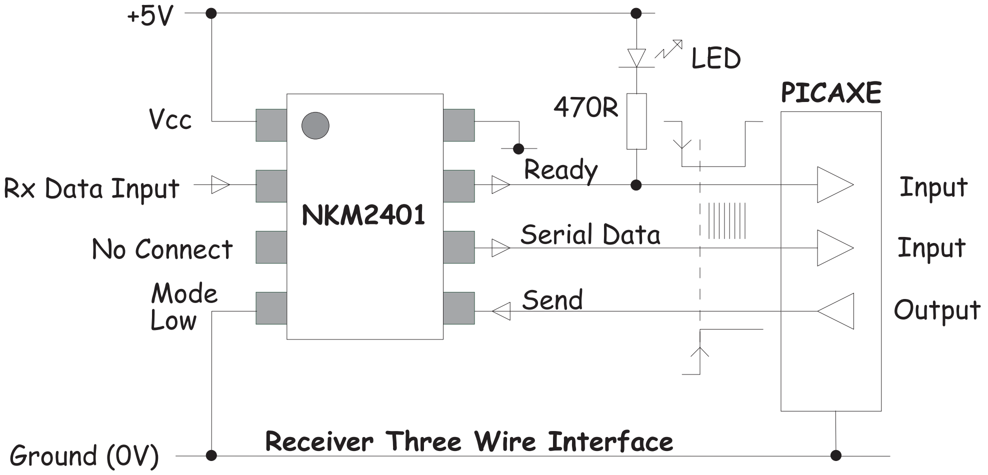
See the diagrams overleaf for more details.

The required configuration must be selected before powering the receiver board. If received data is to an AXE033 LCD or PC only the board should be operated in One-Wire Mode (link bottom two pins of jumper J2).

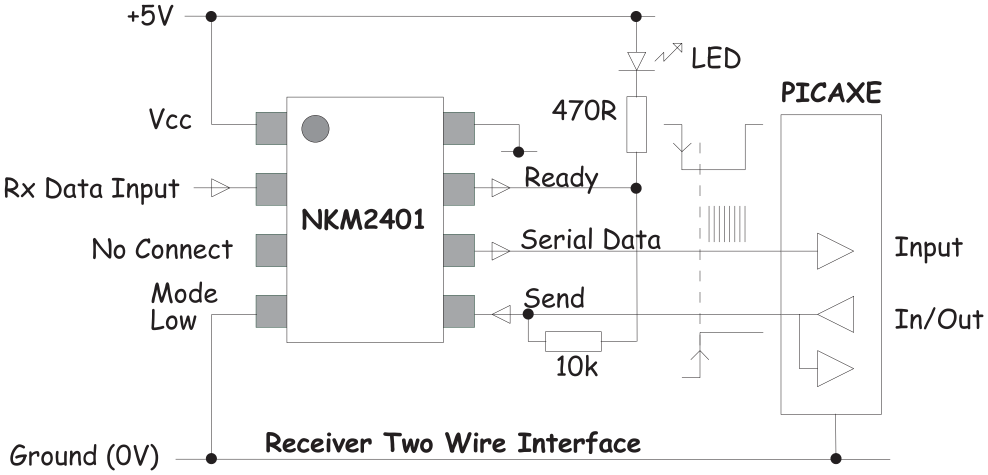
NKM2401 Receiver Configurations



Receiver One Wire Interface (No Handshake)



Receiver Three Wire Interface



Receiver Two Wire Interface

PICAXE Receiver Examples - One-Wire Mode (J2 link bottom two pins)

This mode is most appropriate when the PICAXE is always ready to receive eight bytes of data (e.g. when background serial receive is being used with a PICAXE X2 part). It is also the mode which should be used when received data is being passed to a PC and AXE033 serial LCD module.

The NKM2401 will wait for a data packet to arrive and then the eight data bytes will be immediately sent to the PICAXE and/or PC.

Once the eight bytes have been sent to the PICAXE and/or PC the NKM2401 will be ready to receive the next data packet.

PICAXE Receiver Example 1a – One-Wire Mode

```
; DATA from receiver board connected to Input Pin C.7
; READY from receiver board not used
; SEND to receiver board not used

main:
      serin C.7, n2400, b0,b1,b2,b3,b4,b5,b6,b7
                              ; read eight bytes of data into b0 to b7
      debug                   ; show the data received in b0 to b7
      goto main               ; repeat
```

PICAXE Receiver Example 1b – One-Wire Mode (with qualifier)

```
; DATA from receiver board connected to Input Pin C.7
; READY from receiver board not used
; SEND to receiver board not used

main:
      b0 = "T"                ; set qualifer so sync on the letter "t"
      serin C.7, n2400, (b0),b1,b2,b3,b4,b5,b6,b7
                              ; read eight bytes of data into b0 to b7
      debug                   ; show the data received in b0 to b7
      goto main               ; repeat
```

PICAXE Receiver Examples - Three-Wire Mode (J2 no link fitted)

This mode is most appropriate when the PICAXE wants to be informed that data has been received and buffered, so that the PICAXE can read the data as and when it has time to do so.

When data is received the READY line will be set low. The PICAXE then, when ready, sets the SEND line high and then the first byte of data is sent after 1ms.

If, within 1ms of receiving a data byte, the SEND line is set low, and kept low, no further data bytes will be sent until SEND is again set high. This gives the PICAXE time to process single bytes at a time if required. Once a byte has been processed the SEND line should be set high to receive the next byte and this process repeats until all eight data bytes have been read.

If the SEND line is kept high, all 8 buffered bytes will be sent. The SEND line should be set low again after the eighth data byte has been read.

When connecting the DATA output to a PICAXE operating at 3V (or below the board's V+ supply) the DATA line should be current limited by an in-line 1k resistor.

Note that the NKM2401 will not be able to receive any additional data packets until all eight buffered data bytes have been read.

PICAXE Receiver Example 2a – Three-Wire Mode (no link fitted)

```
; DATA from receiver board connected to Input Pin C.7
; READY from receiver board connected to Input Pin C.1
; SEND to receiver board connected to Output Pin B.2

main:
      low B.2                     ; don't accept data from module yet
      if pinC.1 = 1 then main     ; wait until data available
      high B.2                    ; tell receiver to send data received
      serin C.7, n2400, b0,b1,b2,b3,b4,b5,b6,b7
                        ; read eight bytes of data into b0 to b7
      low B.2           ; tell receiver not to send any further bytes
      debug             ; show the data received in b0 to b7
      goto main         ; repeat
```

PICAXE Receiver Example 2b – Three-Wire Mode using Interrupt (no link fitted)

```
; DATA from receiver board connected to Input Pin C.7
; READY from receiver board connected to Input Pin C.1
; SEND to receiver board connected to Output Pin B.2

main:
      low B.2                       ; don’t accept data from module yet
      setint %00000000, %00000010   ; interrupt when input pin 1 goes low

donothing:                          ; do nothing until data available
      goto donothing

interrupt:
      high B.2                ; tell receiver to send data received
      serin C.7, n2400, b0,b1,b2,b3,b4,b5,b6,b7
                              ; read eight bytes of data into b0 to b7
      low B.2                 ; tell receiver not to send any more bytes
      debug                   ; show the data received in b0 to b7
      setint %00000000, %00000010        ; interrupt when 1 goes low
      return                             ‘ return to ‘donothing’ loop
```

PICAXE Receiver Example 2c – Three-Wire Mode, Byte at a Time (no link fitted)

```
; DATA from receiver board connected to Input Pin C.7
; READY from receiver board connected to Input Pin C.1
; SEND to receiver board connected to Output Pin B.2

main:
      low B.2                            ; don’t accept data yet
      if pinC.1 = 1 then main            ; wait until data available
      gosub readintob13 : b0 = b13       ; read byte into b13 then b0
      gosub readintob13 : b1 = b13       ; read byte into b1
      gosub readintob13 : b2 = b13       ; read byte into b2
      gosub readintob13 : b3 = b13       ; read byte into b3
      gosub readintob13 : b4 = b13       ; read byte into b4
      gosub readintob13 : b5 = b13       ; read byte into b5
      gosub readintob13 : b6 = b13       ; read byte into b6
      gosub readintob13 : b7 = b13       ; read eighth byte into b7
      debug                         ; show the data received in b0 to b7
      goto main                     ; repeat

readintob13:                        ; read a byte from receiver into b13
      high B.2                      ; tell receiver to send one byte
      serin C.7, n2400, b13         ; read one byte into b13
      low B.2                       ; tell receiver not to send any more
      return
```

PICAXE Receiver Example - Two-Wire Mode (J2 link top two pins)

The Two-Wire Mode is very similar to the Three-Wire Mode but a single bi-directional I/O line is used for both the READY signal and control the SEND line. In many cases operation in Two-Wire Mode will be no harder to achieve than operation in Three-Wire Mode.

When using a PICAXE operating at 3V (or below the board’s V+ supply) the Three-Wire Mode should be used rather than the Two-Wire Mode.

The PICAXE I/O pin controlling handshaking, connected to the SEND line, should initially be set as an input which will be pulled low by the NKM2401 READY signal when data becomes available. The NKM2401 SEND signal will also be pulled low at the same time so no data will be sent from the board to the PICAXE.

When the I/O pin detects a low input the pin is then made an output high. The NKM2401 READY signal remains low but the I/O pin is current protected by the resistor to NKM2401 READY. As the I/O pin goes high, this also pulls the NKM2401 SEND signal high which instructs the board to send a data byte to the PICAXE.

When the I/O pin is again made an input, the NKM2401 READY signal being low will pull the NKM2401 SEND signal low and prevent further bytes being sent until SEND is made an output high again.

This appears to be more complicated than it is in practice which simply requires the switching of an I/O line between input (while waiting for data to become available) and setting output high to receive a byte and input to stop further bytes being sent.

PICAXE Receiver Example 3 – Two-Wire Mode (link top two pins of header J2)

```
; DATA from receiver board connected to Input Pin C.7
; READY from receiver board not connected
; SEND to receiver board connected to Input/Output Pin B.2

main:
      input B.2                     ; don’t accept data from module yet
      if pinB.2 = 1 then main       ; wait until data available
      high B.2                      ; tell receiver to send data received
      serin C.7, n2400, b0,b1,b2,b3,b4,b5,b6,b7
                        ; read eight bytes of data into b0 to b7
      input B.2         ; tell receiver not to send any further bytes
      debug             ; show the data received in b0 to b7
      goto main         ; repeat
```

Quickstart Configuration and Testing

For most first-time users the following steps will be useful in checking that your boards are working.

Initial Configuration

Select and purchase appropriate wireless transmitter and receiver modules and any aerials you will be using.

Review the wireless transmitter module datasheet to determine connections to the transmitter board. Make appropriate connections with wire links. Fit the transmitter module to the board. Attach aerial. Attach power connections.

With power applied to the transmitter board, press the Test button and check that the LED flashes when pressed, and flashes every half second whilst held.

Review the wireless receiver module datasheet to determine connections to the receiver board. Make appropriate connections with wire links. Fit the receiver module to the board. Attach aerial.

With the receiver board so the 3.5mm jack is to the bottom, link the bottom two pins of the header which is to the right of the board. This will put the receiver into One-Wire Mode so all data received is automatically passed on to the receiving device.

Connect and AXE027 or AXE026 cable to the 3.5mm jack on the receiver board and connect the cable to your PC. Attach power connections to the receiver board.

Launch Programming Editor or AXEpad, select the serial port to which the AXE027 or AXE026 cable is attached, open the Terminal display (PICAXE>Terminal menu) and select 2400 baud.

With the both transmitter and receiver boards powered, pressing the Test button on the transmitter should cause the LED on the transmitter to flash, the LED on the receiver to flash, and the test message "Test $xx" will be displayed in the Terminal display; the '$xx' will be an incrementing hexadecimal number from $00 to $FF, then back to $00 again. A test message will be sent and should be shown every time the Test button is pressed and repeated every half second while the Test button is held.

If everything is as described you have the transmitter board successfully communicating with the receiver board.

PICAXE Transmitter Testing

Connect your PICAXE transmitter circuit to the 0V and DATA connections of the transmitter board and download the following program into the PICAXE –

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( "12345678" )
      goto main
```

You will need to edit the symbol definition to be the pin number or pin identifier of the PICAXE output pin which connects to the transmitter board DATA connection.

With the receiver board so the 3.5mm jack is to the bottom, link the bottom two pins of the header pins which are to the right of the board. This will put the receiver into One-Wire Mode so all data received is automatically passed on to the receiving device.

Connect and AXE027 or AXE026 cable to the 3.5mm jack on the receiver board and connect the cable to your PC.

With the transmitter and receiver boards plus PICAXE transmitter powered, the LED on the transmitter board should flash once a second, the LED on the receiver should flash once a second, and the text "12345678" should be displayed in the Terminal display and repeated every second.

If everything is as described you now have a PICAXE transmitter communicating with the receiving board.

PICAXE Receiver Testing

Disconnect power from the receiver board. With the receiver board so the 3.5mm jack is to the bottom, link the bottom two pins of the header pins which are to the right of the board. This will put the receiver into One-Wire Mode so all data received is automatically passed on to the receiving PICAXE. Reconnect power.

Connect your PICAXE receiver circuit to the 0V and DATA connections of the receiver board and download the following program into the PICAXE.

```
symbol rxpin = C.7
#terminal 4800
main:
      serin rxpin, n2400, b0, b1, b2, b3, b4, b5, b6, b7
      sertxd (“Received= “, b0, b1, b2, b3, b4, b5, b6, b7, CR, LF)
      goto main
```

You will need to edit the symbol definition to be the pin number or pin identifier of the PICAXE input pin which connects to the receiver board DATA connection.

Leave the download cable connected to your PICAXE receiver circuit, open the Terminal display and select 4800 baud (or 9600 baud if using a PICAXE X2 part).

With the transmitter and receiver boards plus PICAXE transmitter and PICAXE receiver powered, the LED on the transmitter board should flash once a second, the LED on the receiver should flash once a second, and a line of text showing “Received = 12345678” should be displayed in the Terminal display and repeated every second. A line of text showing “Received = Test $xx” should be displayed when the Test button on the transmitter board is pressed, where ‘$xx’ is a hexadecimal number incrementing from $00 to $FF, then back to $00.

If everything is as described you now have one PICAXE communicating with the other wirelessly.

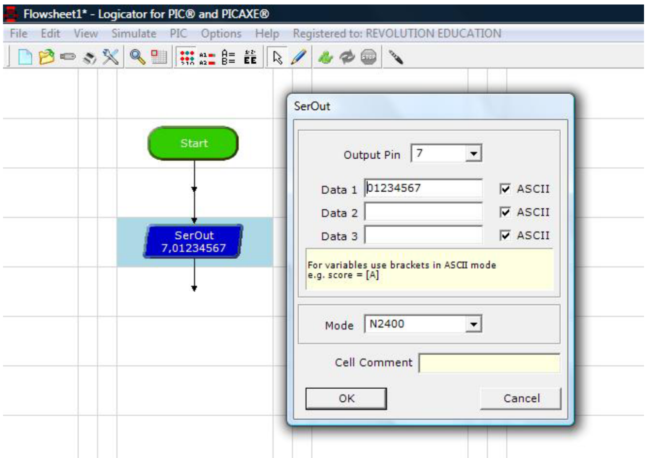
Using with Logicator Flowcharts

Transmitter

DATA from transmitter board connected to Output 7
BUSY not connected

For the transmitter simply use the 'serout' command cell.
Ensure the output pin is set correctly, that the mode is set to 'N2400' and that the data line contains exactly 8 characters. Note that the output connection should come directly from a PICAXE output pin, not from a Darlington (ULN2803A) output connection on a project board.

The BUSY signal is not used. Simply ensure there is at least 500m between transmissions.


Flowsheet1* - Logicator for PIC® and PICAXE®
File Edit View Simulate PIC Options Help Registered to: REVOLUTION EDUCATION
Start
SerOut 7,01234567
SerOut
OK
Cancel

Output Pin 7
Data 1 01234567 ☑ ASCII
Data 2 ☑ ASCII
Data 3 ☑ ASCII
For variables use brackets in ASCII mode e.g. score = [A]
Mode N2400
Cell Comment

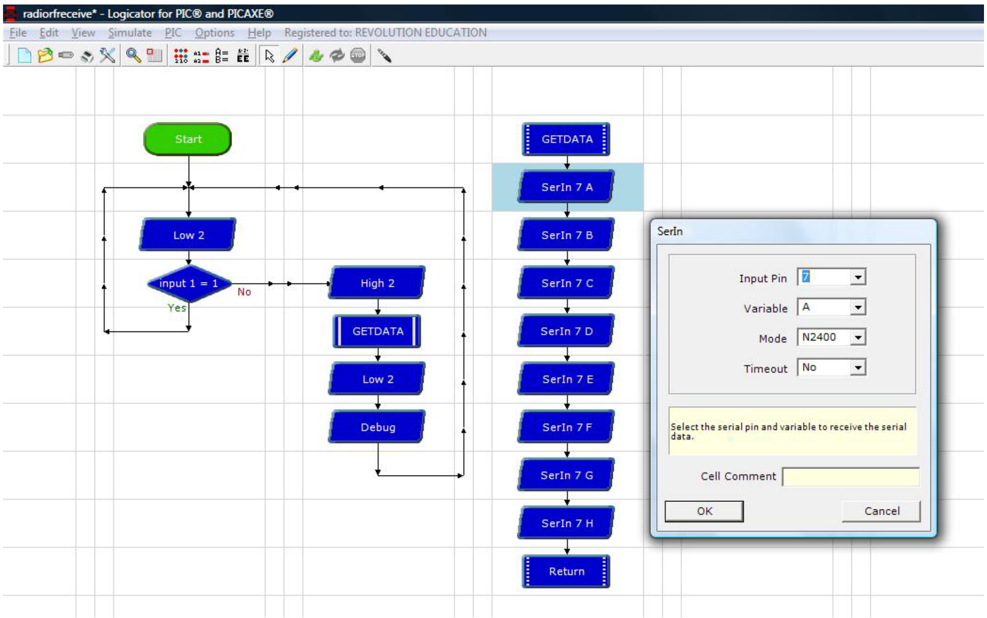
Receiver

It is best to use the receiver in Three Wire mode (no link fitted).

The following flowchart shows how to receive all 8 bytes of data in three wire mode.

DATA from receiver board connected to Input Pin 7
READY from receiver board connected to Input Pin 1
SEND to receiver board connected to Output Pin 2

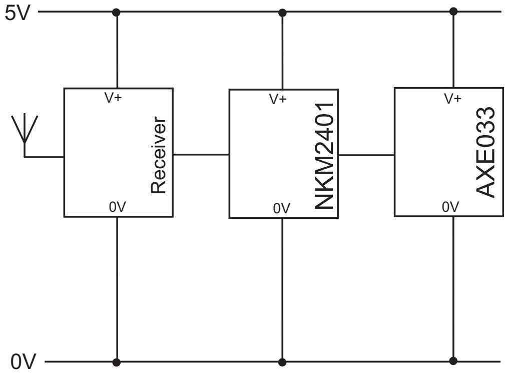
Eight 'serin' command cells receive the eight received bytes of data once the READY signal (input 2) shows that data has been received and buffered.



Input Pin 7
Variable A
Mode N2400
Timeout No
Select the serial pin and variable to receive the serial data.
Cell Comment

Wireless Control of an AXE033 LCD Module

This example application will show how a PICAXE can measure a sensor reading, and transmit it to a remote AXE033 LCD connected directly to the receiver.



Pre-requisites for the receiver in this application are –

An assembled receiver board with receiver module and aerial fitted
An assembled AXE033 LCD module
A 5V power supply

The receiver module should be configured for One-Wire Mode operation; with the 3.5mm jack socket to the bottom of the board the two lower pins of the 3-pin header connector (J2) linked.

The AXE033 module must be configured for correct operation; on the AXE033 remove the MODE (J1) and CLK jumper links if fitted. Remove the POWER (J2) link for 5V operation (J2 is linked for 4.5V operation).

The receiver board and the AXE033 V+ and 0V should be connected to the same 5V power supply.
The receiver board's DATA output should be connected to the AXE033 module's IN input connection.
No connection is required to the receiver board's READY output and SEND input connections.

Connecting the Transmitter Board to a PICAXE

Pre-requisites for the transmitter in this application are –
An assembled transmitter board with transmitter module and aerial fitted
An assembled PICAXE board with PICAXE fitted
A 5V power supply

The transmitter board's V+ and 0V should be connected to the power supply.
The PICAXE board's V+ and 0V should be connected to the power supply.
A PICAXE output pin should be connected to the transmitter board's DATA input.
Note that this connection should come directly from a PICAXE output pin, not from a Darlington (ULN2803A) output connection on a project board.
No connection is required to the transmitter board's BUSY output.

Example PICAXE Programs

In the following example programs Output Pin B.7 is used to send data to the transmitter board’s DATA input. If using a different pin you will need to edit the ‘txpin’ symbol definition to be the pin number or pin identifier of the PICAXE output pin which connects to the transmitter board DATA input.

Data sent via the transmitter board must always be sent eight bytes at a time. These eight bytes may be ASCII data (characters) or raw data bytes such as LCD control commands. Control commands are described in the AXE033 datasheet and include:

254, 1	Clear Display
254, 16	Move Cursor Left
254, 20	Move Cursor Right
254, 128	Move to line 1, position 1
254, y	Move to line 1, position x (where y = 128 + x)
254, 192	Move to line 2, position 1
254, y	Move to line 2, position x (where y = 192 + x)

The following program will send the control code to place the LCD cursor at the start of the first line (254,128), then send the six characters “Hello!”; the LCD should show “Hello!” on the first line -

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, “Hello!” )
      goto main
```

To display “Hello!” on the first line of the LCD and “World!” on the second line we can use the following program –

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, “Hello!” )
      pause 1000
      serout txpin, n2400, ( 254, 192, “World!” )
      goto main
```

To display “Hello Everyone” on the first line of the LCD we can use the following program (note the trailing space after “Hello”) –

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, “Hello “ )
      pause 1000
      serout txpin, n2400, (“Everyone” )
      goto main
```

To send less than six characters we still need to send eight bytes at a time. One way to achieve this is to send cursor positioning commands a multiple number of times. To display "Hi" on the first line of the LCD we can send the control code to place the LCD cursor at the start of the first line (254,128) three times, then send the two characters "Hi"; eight bytes sent in total –

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, 254, 128, 254, 128, "Hi" )
      goto main
```

To send an odd number of characters we can often add an extra space character to make the number of characters even with no adverse consequences or we can send control codes to move the cursor back a column (move cursor left) and resend the character which was already placed in that column. To display "Hi!" we can send "Hi", move back a column, then send "i!" -

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, "Hi", 254, 16, "i!" )
      goto main
```

Instead of using move cursor left (or right) commands it is also possible to use the character position control codes to select the exact character position to overwrite -

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, "Hi", 254, 129, "i!" )
      goto main
```

The clear screen command requires a delay after it is received while the AXE033 and LCD processes that command. To allow for that delay an eight byte packet can be sent consisting of three dummy cursor positioning commands and then the control code for clear screen. The following program will send three commands to set the cursor to the start of the first line then clear the screen before displaying "Hello!" then adding "World!" -

```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, 254, 128, 254, 128, 254, 1 )
      pause 1000
      serout txpin, n2400, ( 254, 128, "Hello!" )
      pause 1000
      serout txpin, n2400, ( 254, 192, "World!" )
      goto main
```

The pause after issuing a clear screen needs to be 30ms for the AXE033 and LCD but we also have to take into account the transmission time of the data packet sent. The pause should therefore be at least the transmission time plus the pause time required.

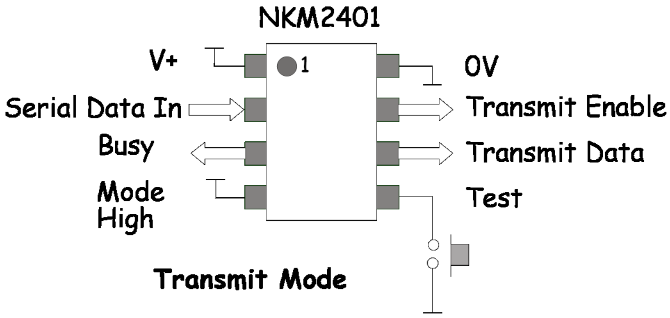
Our final example shows a ‘real world application’ which reads the analogue input from a potentiometer connected to ADC input 1 and shows its value on the LCD display with right alignment.

The first step is to clear the display and output an “ADC 1 = “ message (eight bytes) then repeatedly output the value read placed at the correct position on the display. We use the ‘bintoascii’ command to convert the analogue value read into easy to use individual digits to send –

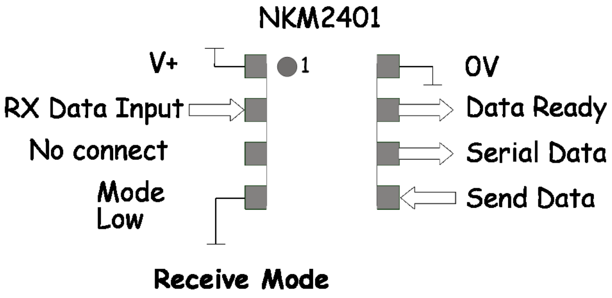
```
symbol txpin = B.7
main:
      pause 1000
      serout txpin, n2400, ( 254, 128, 254, 128, 254, 128, 254, 1 )
      pause 250
      serout txpin, n2400, ( "ADC 1 = " )
      pause 250
mainloop:
      readadc 1, b0
      bintoascii b0, b1, b2, b3
      serout txpin, n2400, ( 254, 135,  254, 135, " ", b1, b2, b3 )
      pause 250
      goto mainloop
```

AXE213 Schematic

NKM2401 Pin Out Schematic & Description



Pin	Function (Mode Pin High - Transmit)	
1	V+	+5V supply
2	Serial Data Input	2400bps
3	Busy Output.	Goes high whilst transmitting. Active High
4	Mode Pin	High (V+) selects transmit mode
5	Test Input	Sends test message. Active Low
6	Transmit Data	Output to RF transmitter
7	Transmit Enable	Output to RF transmitter. Active High
8	0V	Ground



Pin	Function - (Mode Pin Low - Receive)	
1	V+	+5V supply
2	Data input	From RF receiver
3	No connection.	Do not connect. Factory test point.
4	Mode Pin	Low (0V) selects receive mode
5	Send Data	Pull high to send the data in receive buffer
6	Serial Data Input	2400bps
7	Data Ready	Goes low when valid data in buffer
8	0V	Ground

NKM2401 Technical Specifications

Ambient Temperature	20° C
Supply Voltage	+5.0 V

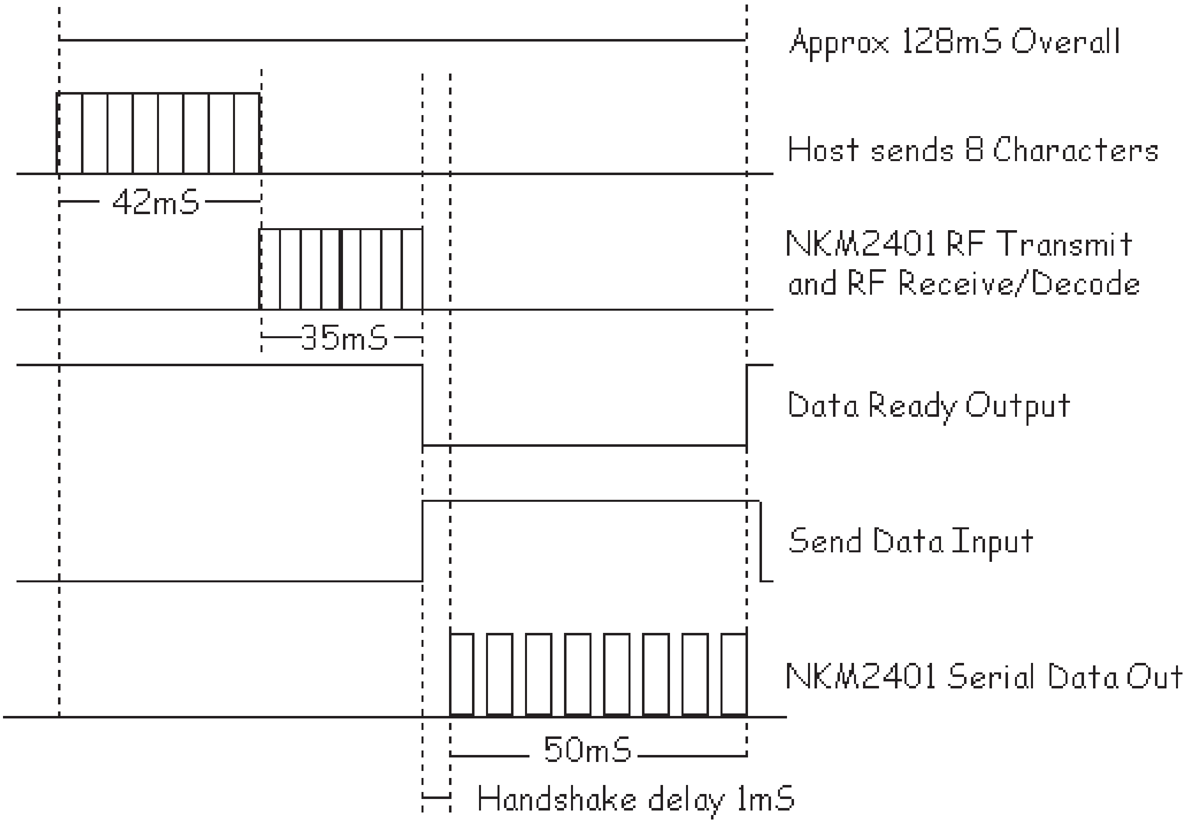
Parameter	Min	Typ.	Max	Units	Notes
Supply Voltage	3.0	5.0	5.5	Volts	
Supply Current	1.8		2.4	mA	1
TX enable source current			20	mA	
Serial In/Out Baud rate		2400		bps	2
Character time		4.16		ms	3
RF transmit bit time		200		us	4
RF transmit time		35		ms	
Receive & decode time		36		ms	
Handshake delay		1		ms	
Serial data output time		50		ms	5
Operating temperature	0	20	+40	deg. C	
Storage Temperature	-40		+85	deg. C	

Notes

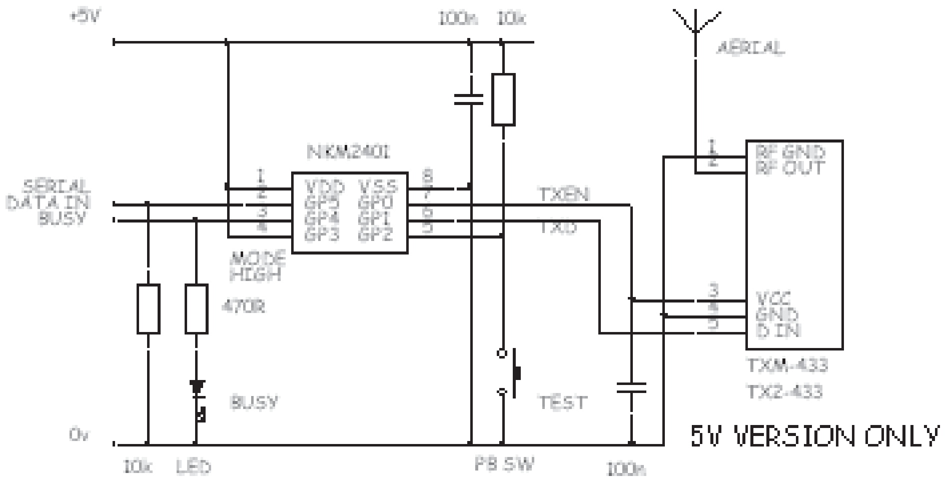
1) Excluding current required for wireless transmitter module through TX enable (TXEN)
2) Data signal polarity is always 'true' with the same polarity as RS232 +/-12V levels but at 0V to +5V (supply voltage) logic levels; start bit is logic high, stop bit is logic low. This is N2400 baud rate and polarity for PICAXE and 2400,N,8,1 format for PCs.
3) 1 start bit, 8 data bits, 1 stop bit = 10 bits at 416us per bit = 4.16ms.
4) Manchester encoded.
5) Eight bytes with handshake delay.

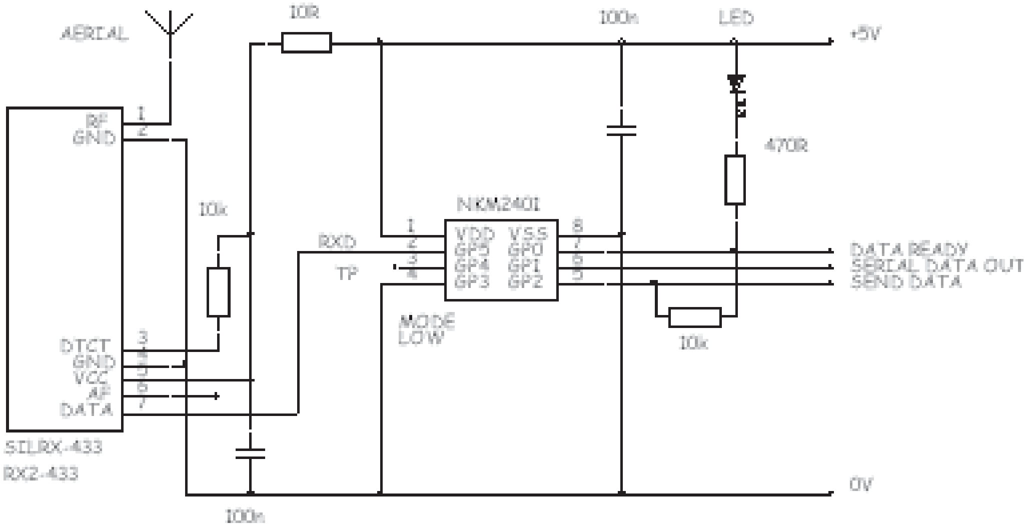
Note that the NKM2401 is supplied as a pre-programmed PIC12F508-I/P microcontroller. The firmware is provided free of charge by Nikam Electronics.
If you wish to make your own NKM2401 chips using a suitable PIC programmer, the hex file to program onto a new (blank) PIC12F508-I/P is available at this web link:
http://www.rev-ed.co.uk/software/NKM2401.HEX
All fuse configuration settings are embeded in the hex file.

NKM2401 System Timing Diagram



Typical NKM2401 Application Circuit





Manufacturer Website

Main website:	www.picaxe.co.uk
Forum:	www.picaxeforum.co.uk
VSM Simulator:	www.picaxevsm.com

PICAXE products are developed and distributed by
Revolution Education Ltd www.rev-ed.co.uk

Trademark

PICAXE® is a registered trademark licensed by Microchip Technology Inc. Revolution Education is not an agent or representative of Microchip and has no authority to bind Microchip in any way.

Acknowledgements

NKM2401 firmware, data and all related information kindly provided free of charge by Mike Meakin of Nikam Electronics and reproduced with his permission.