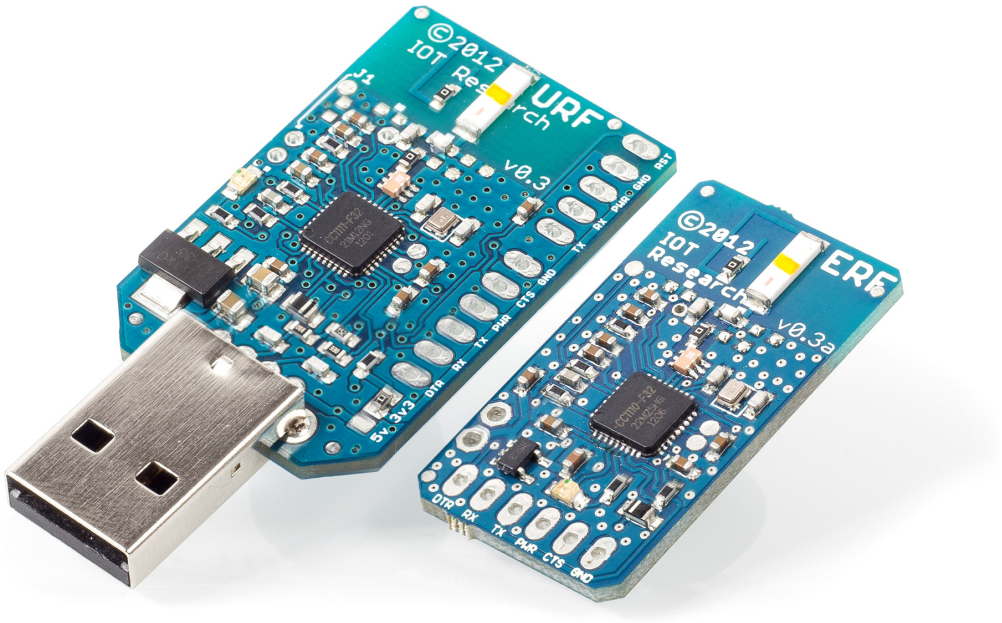
Which Wireless Radio Module for my PICAXE Project?



The ERF, URF and XRF modules are designed to provide 'out-of-the-box' instant wireless bi-directional serial communication between a computer and/or multiple PICAXE project boards. All 3 modules are designed around the Texas Instruments CC111x combined micro & RF transceiver integrated circuit.

It is possible to communicate wirelessly between PICAXE chips using these modules by simply using the standard 'serout' and 'serin' PICAXE commands. All 3 modules can communicate with each other. However for new projects we recommend use of the ERF and URF modules.

A URF and ERF pair may be used for serial communication between a computer and a PICAXE project. As a unique feature this pair also supports 'over-the-air' remote wireless PICAXE programming/debugging.

Two (or more) PICAXE project boards can communicate with each other via multiple ERF modules. These modules can communicate up to 500m via line of sight (obstacles such as walls and trees will decrease range).

It is also possible to connect an ERF to the computer via an AXE027 PICAXE USB cable (i.e. using an ERF at both computer and PICAXE end of a link). However this combination can only be used for serial communications, it will not work for wireless PICAXE programming.

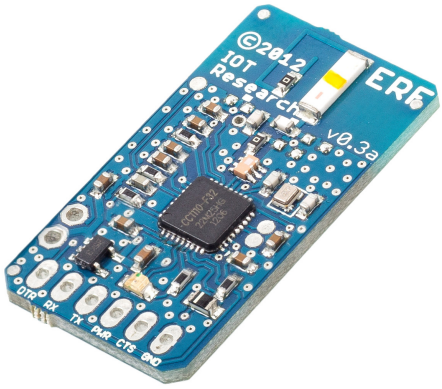
Module Comparison:

Module	**URF**	**ERF**	**XRF**
Connection Style	USB	6 pin 0.1"	2 x 10 pin 2mm (XBee style)
Operating Voltage	USB	3 to 5V	3V only
PCB Size (mm)	39 x 25 (exc. connector)	35 x 19	28 x 25
Weight (g)	5	2	4
Default Baud Rate	n/a	N9600 (auto*)	T9600
Optional Baud Rate	n/a	N1200 to N115200	T1200 to T115200
Antenna Style	Chip	Chip	Wire
Default Frequency (MHz)	868.3	868.3	868.3
Optional Frequencies (MHz)	915, 903, 868, 433, 315	915, 903, 868, 433, 315	915, 903, 868, 433, 315
Wireless Serial Communication	Yes	Yes	Yes
PICAXE Programming	Yes	Yes	No

When the ERF is paired with a URF the correct baud rate will be set automatically on the ERF. On all other pairings the default baud rate is used. The default baud rate can be reconfigured if desired via an ATBD command.

In general it is only necessary to use the older XRF module when you are using an existing PCB or 'shield' that was originally designed for the 2mm spaced 'XBee' style footprint. Note also that the XRF is 3V only, whereas the ERF is more versatile at it can be used at 3 to 5V and also has 'standard' 0.1" (2.54mm) spaced pads.

ERF Module (part RFA020)



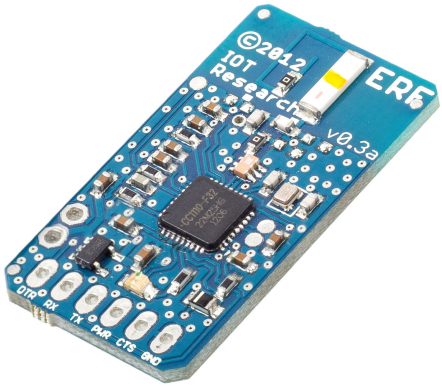
The ERF is the easiest and most convenient way to wirelessly communicate between a PICAXE chip and a computer, or between 2 (or more) PICAXE chips.

The serial and power connections are via standard 0.1" headers, making this module ideal for breadboard and stripboard work. Unlike most competitive radio modules the ERF can also be run at 5V (operates from 3 to 5V). The ERF transparently passes serial characters to and from individual or multiple radio devices. There is no need to use complex encoding or error checking as it is all done for you - simply send raw text/data via 'serout', and receive text/data via 'serin'.

When paired with a URF module you can also wirelessly reprogram a PICAXE chip 'over-the-air'. Simply replace your download cable with a URF/ERF pair for instant wireless PICAXE programming and debugging.

Key Features:

- Supplied preprogrammed with customised firmware to enable support of wireless PICAXE programming and debugging.
- Defaults to 'N9600' baud rate/polarity (can be reconfigured).
- 3 to 5V operation.
- 6 pin 0.1" (2.54mm) header connection
- Communicates with XRF, URF and ERF modules.
- Inbuilt chip antenna for neat robust finish, up to 500m range (LOS)
- Point to point, point to multipoint, or custom addressing available
- Can specify a network identifier PANID to separate communications into separate networks if desired
- Based on the Texas Instruments CC1110 combined micro & RF transceiver.
- Runs as standard at 868.3MHz, but 6 different frequencies can be configured (e.g. to use 6 different pairs in the same classroom). The chip antenna is optimised for 868-915MHz, but will also work at 315MHz and 433MHz with a reduced range.



Pinout *(6 pin 0.1" (2.54mm) header, suitable matching socket part CON047):*

```
01 - DTR -> Connect to LED via 1k resistor for RX/TX activity indicator
02 - RX  <- ERF data in, connect to PICAXE serial output pin
03 - TX  -> ERF data out, connect to PICAXE serial input pin
04 - PWR :  3 to 5V power supply
05 - CTS <- Chip enable, active low, connect to 0V to enable module
06 - GND :  0V power supply
```

If using the ERF as a 'radio receiver only' connect RX to 0V. Do not ever leave the RX pin floating (unconnected). If RX is connected to a PICAXE pin make sure that pin is actively driven as a low output (via a 'low' command).

If using the ERF as a 'radio transmitter only' TX may be left unconnected.

Kindly note that the module will not operate unless the CTS 'chip enable' pin is connected to 0V. If there is no heartbeat LED indicator after connecting power please double check the CTS to 0V connection!

The factory set default baud rate for the ERF module is N9600_8 (which requires the PICAXE chip to be operating at 8MHz).

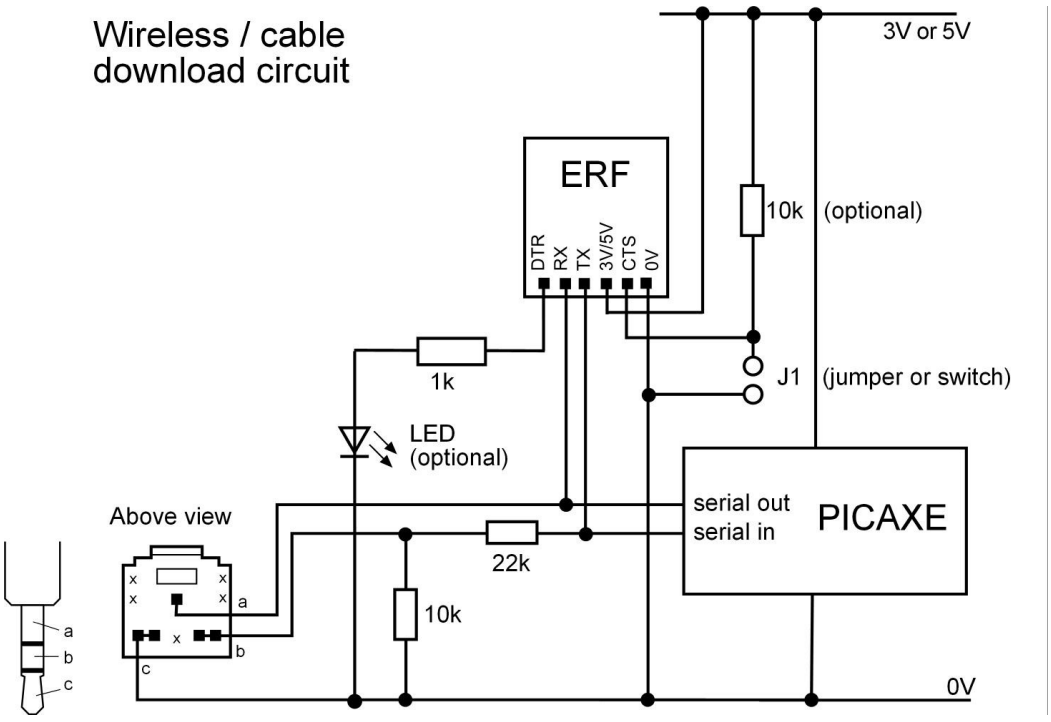
Connecting an ERF to a PICAXE for Wireless Programming:

The ERF to PICAXE programming circuit is shown below. The ERF can coexist with the normal 3.5mm socket for direct AXE027 cable programming. In most cases it can also be retrofitted to an existing circuit.

When the jumper (or switch) J1 is open the 10k resistor pulls the CTS 'chip enable' pin high. This places the ERF in a suspend mode so that it has no effect on the circuit. Therefore the AXE027 USB cable can be used as normal.

When the jumper (or switch) J1 is closed the ERF becomes active and wireless programming is then possible. If both AXE027 and ERF are accidentally both connected at the same time the 22k resistor prevents physical damage (although naturally neither system will successfully program).

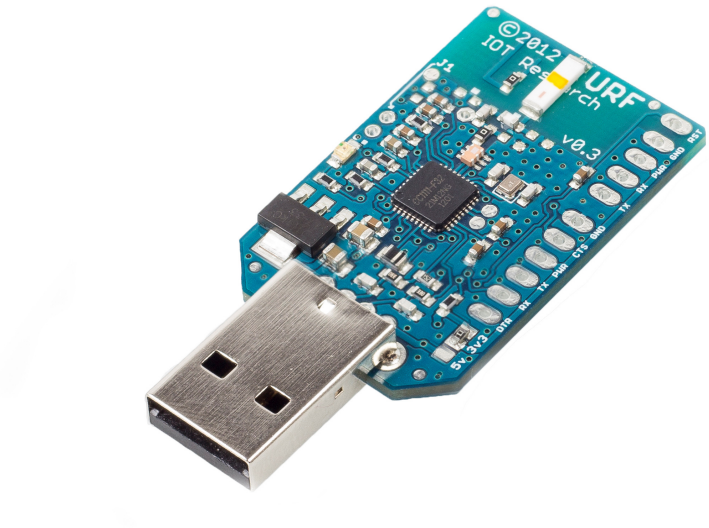
An optional 'activity' LED can connect to the ERF DTR pin. This is a useful testing feature as it flashes when there is any RX/TX activity on this ERF module. Note this is a different feature to the power on 'heartbeat' LED visible on the ERF module itself.



Notes on using the ERF for PICAXE programming

1) When used with a URF module the baud rate for downloads/debug/serial terminal etc. is set automatically on the ERF (via an automatic wireless command from the URF to the ERF) when the URF COM port is opened by the computer. Therefore there is no need to configure the baud rate manually on the ERF (i.e. no need to set the baud rate via AT commands when paired with a URF).
2) When an ERF resets it will always use the default baud rate stored in it's EEPROM (factory setting 9600) until the URF instructs it to change. Therefore if, for instance, the PICAXE Editor 'Serial Terminal' opens the URF COM port at 4800 the ERF will then automatically change to 4800 and start transmitting data at 4800 as expected. However if the ERF is powered down, and then re-powered whilst the Serial Terminal on the computer remains open the ERF will default back to 9600 baud and the transmitted data will no longer be at the expected 4800. In this situation it is necessary to simply close and re-open the COM port (Serial Terminal) to force the URF to instruct the ERF to change its baud rate again.
3) To perform a 'power-on' wireless hard-reset of the PICAXE chip the ERF must still remain powered. In other words only reset the power to the PICAXE chip itself, not the whole circuit.
4) The DTR pin can supply an absolute maximum of 4mA to drive the optional activity LED. Therefore an external 1k resistor is recommended in series with the LED. Drawing a larger current may cause unreliable operation.
5) The CTS 'chip enable' pin has a 'weak' internal pull up resistor, and so the external 10k pull-up is optional. However we recommend adding it to provide a 'strong' pull-up in the circuit.
6) To use two or more URF/ERF pairs in close proximity (e.g. in a classroom for 2 remote robots) each pair should be configured to use a different frequency via the URF configuration wizard, which is a free download from www.picaxe.com/downloads/urf.zip. The default frequency is 868.3MHz and this is the frequency the antenna is optimised for. Other available frequencies that work well with the on-board chip antenna are 868, 915 and 903. However in a short range (e.g. 10m or so within a classroom) very different frequencies such as 315 and 434 will also function, giving up to 6 default pair options.

URF Module (part RFA021)



The URF is the easiest and most convenient way to wirelessly communicate between a PICAXE chip and a computer. Simply connect the URF directly to the USB port of your computer and an ERF module to the PICAXE chip.

Power is supplied from the USB port, and the URF transparently passes serial characters to and from individual or multiple ERF radio devices. There is no need to use complex encoding or error checking as it is all done for you - simply send raw text/data form your favourite Serial Terminal application, and receive text/data via the 'serin/serout' commands. Even the baud rate is set automatically on the remote ERF!

When paired with a URF module you can also wirelessly reprogram a PICAXE chip 'over-the-air'. Simply replace your download cable with a URF/ERF pair for instant wireless PICAXE programming and debugging. All baud rate settings are fully automatic when used in this way.

Key Features:

- Plugs into the computer's USB port to provide a virtual COM port.
- When paired with an ERF module auto-sets the ERF baud rate to match the computer baud rate. When paired with an XRF uses the XRF default baud, regardless of computer baud rate.
- Communicates with XRF, URF and ERF modules.
- Inbuilt chip antenna for neat robust finish, up to 500m range (LOS)
- Point to point, point to multipoint, or custom addressing available
- Can specify a network identifier PANID to separate communications into separate networks if desired
- Based on the Texas Instruments CC1110 combined micro & RF transceiver.
- Runs as standard at 868.3MHz, but 6 different frequencies can be configured (e.g. to use 6 different pairs in the same classroom). The chip antenna is optimised for 868-915MHz, but will also work at 315MHz and 433MHz with a reduced range.
- Supports Windows, Linux, Mac and Android with OTG support

Installing the URF USB Driver:

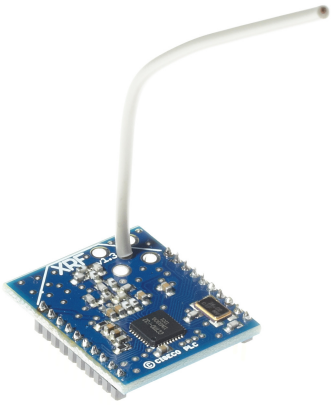
1) Download the USB driver file from www.picaxe.com/downloads/urf.zip
2) Unzip this file and drag out the contained file (URF_CC1111_USB_Driver.inf) onto your desktop.
3) Insert the URF into the USB port of your computer.
4) When the Windows 'New Hardware Wizard' automatically starts point the Wizard to the URF_CC1111_USB_Driver.inf file when prompted. Accept any security warnings and let the wizard complete.
5) Once completed Windows will allocate a 'COM port' number to the URF device. You can view which COM port number this is within the Windows Device Manager (in the Ports (COM & LPT) section).
6) Within the PICAXE software select the appropriate COM port. Note that the port may be simply named 'Communication Port' within the PICAXE software.
7) Connect the ERF module to the PICAXE circuit (see ERF section of this datasheet), ensure the ERF CTS pin is tied to 0V, and download as normal.

Increasing URF Range:

All computers emit quite a lot of local RF noise and also act as an increased ground plane. Therefore although the URF will work when inserted directly into the computer's USB port, the range of the URF can often be greatly improved by lifting it slightly above and away from the computer via use of a short USB extension cable (e.g. part CAB033).

The optimum chip antenna orientation is when laid flat, so best results are generally obtained when the URF PCB is laid flat (parallel with the floor).

XRF Module (part RFA022)



The XRF is an XBee footprint compatible module to allow wirelessly communication between a PICAXE chip and a computer, or between 2 (or more) PICAXE chips. The XRF transparently passes serial characters to and from individual or multiple radio devices. You don't need to use complex encoding or error checking as it is all done for you - simply send raw text/data via 'serout', and receive text/data via 'serin'.

Please note this module does not support over-the-air PICAXE programming - you will need the ERF module instead to achieve this. For new projects we would recommend the more versatile ERF module instead of this older XRF module.

Key Features:

- 3V operation, but has 5V tolerant i/o for direct connection to 5V circuits.
- 10 pin (2mm) header connection, XBee footprint compatible.
- Communicates with XRF, URF and ERF modules.
- Defaults to 'T9600' baud rate/polarity (can be reconfigured).
- Wire antenna for increased range (up to 500m or more). PICAXE users have achieved over 3km line-of-sight connection across water with two of these modules!
- Point to point, point to multipoint, or custom addressing available
- Can specify a network identifier PANID to separate communications into separate networks if desired
- Based on the Texas Instruments CC1110 combined micro & RF transceiver.
- Runs as standard at 868.3MHz, but 6 different frequencies can be configured (e.g. to use 6 different pairs in the same classroom).

***XRF Pinout** (2x10 pin 2mm header, suitable socket part CON046):*

```
01 - PWR +3V                20 - (Factory Configuration Reset)
02 - TX Data Out            19 - (Factory Configuration Reset)
03 - RX Data In             18 - Future use
04 - Future use             17 - Future use
05 - Reset                  16 - Future use
06 - Heart Beat LED         15 - Future use
07 - Future use             14 - Future use
08 - Future use             13 - Future use
09 - Sleep                  12 - Future use
10 - GND                    11 - Future use
```

TX (pin 2) connects to the PICAXE input RX pin.
RX (pin 3) connects to the PICAXE output TX pin.

If using the XRF as a 'radio receiver only' connect RX (pin 3) to +3V. Do not ever leave the RX pin floating (unconnected). If RX is connected to a PICAXE pin make sure that pin is actively driven as a high output (via a 'high' command).

If using the XRF as a 'radio transmitter only' TX (pin 2) may be left unconnected.

The factory set default baud rate for the XRF module is T9600_8 (which requires the PICAXE chip to be operating at 8MHz).

Appendix 1 - ERF Module – Custom firmware for PICAXE support

ERF modules supplied by PICAXE distributors are pre-programmed with a special custom firmware to enable over-the-air PICAXE programming. These modules can be identified by a yellow band half way down the white chip antenna next to the 'ERF' label.

This special firmware adds 5 unique additional features to the standard ERF system:

1) Inverts the polarity of the serial signal as required by the PICAXE system, so that it now defaults to N9600 (idle low, active high polarity).
2) Adds support for automatic baud-rate setting whilst programming/debugging with a wireless link to a URF module.
3) Adds support for the RS232 break signal as required by the PICAXE system.
4) Adds TX/RX activity LED function to DTR pin.
5) Adds active low 'chip enable' function to CTS pin.

ERF modules sourced elsewhere do not have this special firmware or features. Therefore although they may be still used for normal serial communication projects (at T9600) they cannot be used for wireless PICAXE programming.

It is not possible for the user to reprogram 'standard' ERF modules into 'PICAXE' ERF modules – the custom firmware is factory programmed onto new units only.

Module	**URF**	**ERF**	**ERF**	**XRF**
Firmware	**Standard**	**Custom**	**Standard**	**Standard**
Connection Style	USB	6 pin 0.1"	6 pin 0.1"	2 x 10 pin (2mm XBee)
Operating Voltage	USB	3 to 5V	3 to 5V	3V only
Default Baud Rate	n/a	N9600 (auto*)	T9600	T9600
Antenna Style	Chip	Chip	Chip	Wire
Default Frequency	868.3	868.3	868.3	868.3
Optional Frequencies	915, 903, 868, 433, 315	915, 903, 868, 433, 315	915, 903, 868, 433, 315	915, 903, 868, 433, 315
Wireless Serial Communication	Yes	Yes	Yes	Yes
PICAXE Programming	Yes	Yes	No	No
HeartBeat LED	Yes	Yes	Yes	Yes
Activity LED on DTR	n/a	Yes	No	No
Chip Enable on CTS	n/a	Yes	No	No

** When the PICAXE custom firmware version ERF is paired with a URF the correct baud rate will be set automatically on the ERF. On all other pairings the default baud rate is used. The default baud rate can also be reconfigured via an ATBD command.*

Appendix 2 - Sample ERF to ERF test programs (serout / serin on 28X2)

The following two programs can be used to test a wireless connection between two ERF modules. The transmitter sends test data once per second and the receiver displays the received data via sertxd commands to the PICAXE Serial Terminal.

Remember to maintain a reasonable distance (over 1m) between the two units.

Serout command (transmitter):

```
#picaxe 28X2
; ERF RX to PICAXE TX (C.1)
; ERF TX to PICAXE RX (C.7)
; ERF CTS to 0V

setfreq m8

low C.1
pause 1000

main:
  bintoascii b0, b4,b5,b6
  serout C.1, n9600_8, ("ABC", b4, b5, b6 )
  pause 1000
  b0 = b0 + 1
  goto main
```

Serin command (receiver):

```
#picaxe 28X2
#terminal 9600
; ERF RX to PICAXE TX (C.1)
; ERF TX to PICAXE RX (C.7)
; ERF CTS to 0V

setfreq m8

low C.1     ; NB! ensure ERF RX is held low
            ; even when not used in this program

main:
  serin C.7, n9600_8, b0,b1,b2,b3,b4,b5
  sertxd( b0,b1,b2,b3,b4,b5, cr, lf )
  goto main
```

Appendix 3 - Sample ERF to ERF test programs (hserout / hserin on 28X2)

The following programs can be used to test a wireless connection between two ERFs. The transmitter sends test data once per second and the receiver displays the received data via sertxd commands to the PICAXE Serial Terminal.

Remember to maintain a reasonable distance (over 1m) between the two units.

Hserout command (transmitter):

```
#picaxe 28X2
; ERF RX to PICAXE TX (C.1)
; ERF TX to PICAXE RX (C.7)
; ERF CTS to 0V

hsersetup b9600_8, %110
pause 1000

main:
  bintoascii b0, b4,b5,b6
  hserout 0, ("ABC", b4, b5, b6 )
  pause 1000
  b0 = b0 + 1
  goto main
```

Hserin command (receiver):

```
#picaxe 28X2
#terminal 9600
; ERF RX to PICAXE TX (C.1)
; ERF TX to PICAXE RX (C.7)
; ERF CTS to 0V

hsersetup b9600_8, %110

main:
  hserin 0, 6
  get 0, b0
  get 1, b1
  get 2, b2
  get 3, b3
  get 4, b4
  get 5, b5
  sertxd( b0,b1,b2,b3,b4,b5, cr, lf )
  goto main
```

Hserin command (background receiver):

```
#picaxe 28X2
#terminal 9600
; ERF RX to PICAXE TX (C.1)
; ERF TX to PICAXE RX (C.7)
; ERF CTS to 0V


hsersetup b9600_8, %111

main:
  if ptr = hserptr then main
  sertxd( @ptrinc )
  goto main
```