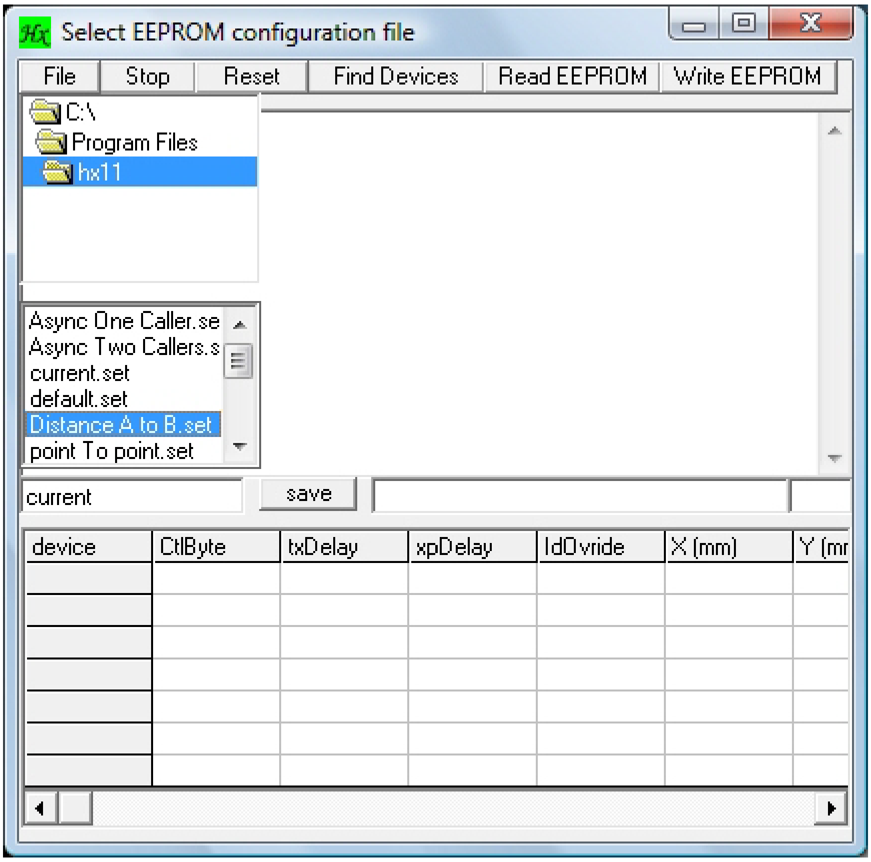
Using the configuration facility

There are two ways of changing the Hx11 parameters. Click on **Find Devices**, the program will search for all the devices connected on the network; this can take a few seconds. Once it can be acknowledged that all the devices have been accounted for then click on **Stop**. Note that if devices are active it may be necessary to click on **Stop** also before **Find Devices**. After all the devices have been found, click on **Read EEPROM**. This action should bring up the current configuration of the devices on line. The parameters can be modified and then transmitted back to the device (click **Write EEPROM**). Or the parameters can be saved on a file (change the name of **Current** to your own file name).

The other way of configuring the devices is by using a preset configuration. Double click on the proper file.

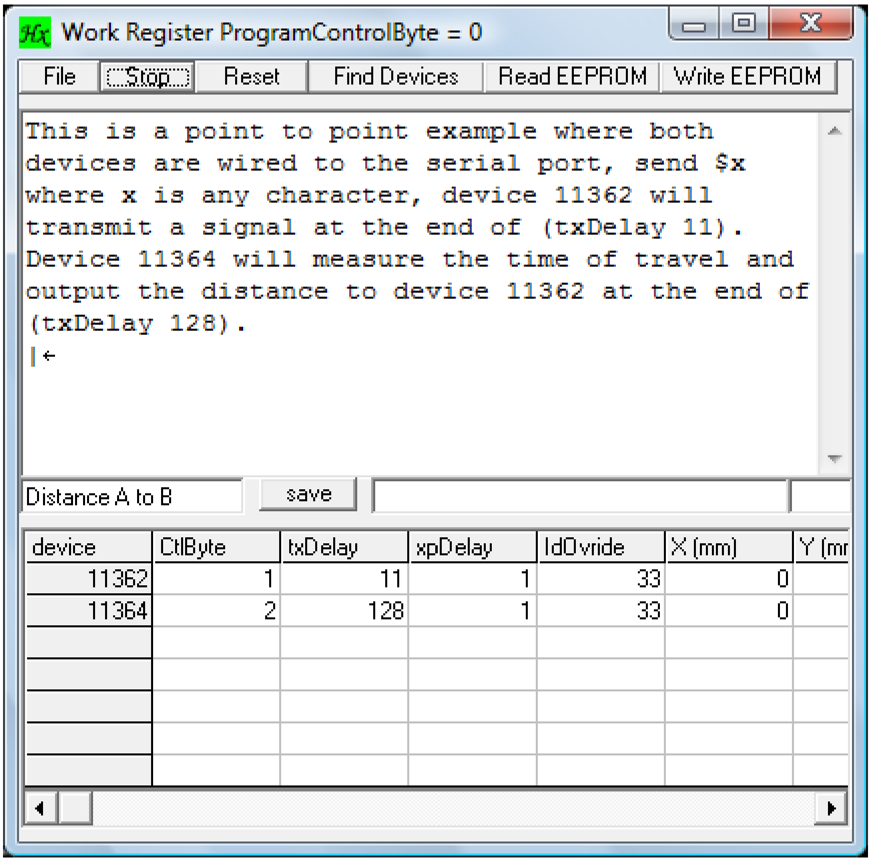


Async One Caller.se
Async Two Callers.s
current.set
default.set
Distance A to B.set
point To point.set

current

device	CtlByte	txDelay	xpDelay	IdOvride	X (mm)	Y (mr

Following example shows file Distance A to B, click on **Write EEPROM** to download the parameters to the devices on line.



```
This is a point to point example where both
devices are wired to the serial port, send $x
where x is any character, device 11362 will
transmit a signal at the end of (txDelay 11).
Device 11364 will measure the time of travel and
output the distance to device 11362 at the end of
(txDelay 128).
|←
```

Distance A to B

device	CtlByte	txDelay	xpDelay	IdOvride	X (mm)	Y (mr
11362	1	11	1	33	0	
11364	2	128	1	33	0	

The file “Distance A to B.set” looks like this and a plain text editor can be used to edit the file.

```
This is a point to point example where both devices are wired
to the serial port, send $x where x is any character, device
11362 will transmit a signal at the end of (txDelay 11). Device
11364 will measure the time of travel and output the distance
to device 11362 at the end of (txDelay 128).
|
11362          1              11            1             33
11364          2             128            1             33
```

The | sign indicates end of comments and start of data. If “Write EEPROM” is selected, the data in the first line is written to the EEPROM of device 11362 and the second line is written to the EEPROM of 11364.

Operation

Hx11 configure can be used to configure the HX11 devices either one by one, or many of them connected on a network. It should be able to configure networks connected to a HX11C (controller) or not. There are two ways to configure the device, i.e. by using a file from the file menu, or by finding the devices on the network, read their (use read EEPROM), configure the settings and reload by pressing "write EEPROM".

To edit the configuration table once the values have been loaded, click the desired (row,column), this will bring the value in the column to the right most text window, where it is highlighted, edit this value. Repeat this until all the values in the configuration table are as desired, then press either save to save on a file, or press "Write EEPROM". The file name can be edited by changing the content of the left most text box holding the current file name.

To modify the devices connected press the "Find Devices" button, then wait for a few seconds until the addresses of all the devices on the network have been listed. Once they are all listed press the "Stop" button, this will bring the listed primary device addresses in the configuration table. Once the device addresses are in the configuration table, then press "Read EEPROM", this will bring the parameters as they are currently configured into the configuration table. These parameters can be edited and changed, and then reloaded onto the devices on the network. This is done by pressing "Write EEPROM".

To bring a file into the configuration table, press the "File" button and select the configuration file, these have the extension SET. The SET file is really a text file and can be edited using any ordinary text editor.

TxDelay

The constant txDelay is in terms of 4.096 milliseconds, txDelay of 128 equals roughly 524 milliseconds. To put all of this into perspective, suppose txDelay of 11362 was set to 1 and 11364 was set at 10. Then the sonic signal leaves 11362 4mS after sync, given that the signal must travel maximum 4 meters, it takes the signal which is 15mS long, about 12 mS to travel this distance. It takes another 10mS to transmit the results over the serial wire; hence the value 10 x 4 (40mS) is barely adequate. If the value is less than what is required, the signal may not be fully in the round buffer when the device is required to empty its contents. Therefore next time around there may be two values in the round buffers making the sampling erratic.

XpDelay

The xpDelay is a delay introduced between the instant the transponder receives a call, to the instant the transponder responds to the call. It can be useful to allow some of the residual signal from previous transmission to decay at least partially. The xpDelay is in terms of 4.096 milliseconds.

IdOverride

Devices will respond to a transmitted Id under 16. So if one device sends a signal with an identification code less than 16, then any device receiving the code will respond by sending back a signal following xpDelay. The calling device will yield the time tag of the signal, (i.e. time of flight).

Hx11 Receiver Background operation

When the device receives $, it clears its timers upon the rising edge of the start-bit belonging to the character following the $. The latency of this synchronization is 200nS, and the timers overflow every 1.048576 seconds.

The device listens for ultrasonic signals and logs the identity and time of signal arrival in a ring (round) buffer. Data stored is referred to as the time-tag, where the time of arrival equals the time elapsed from synchronization, until a fixed point in the ultrasonic signal is detected.

If the device receives its own device ID (16bit) address over the serial lines, it will transmit all the data in the ring buffers. The ring buffer overflows for every 32 ultrasonic time tags received and the oldest time-tag gets overwritten

CtlByte = 1 (mode bit 0 set)

Once the time elapsed from synchronization equals time specified by the txDelay, then the device transmits an ultrasonic ID. This ID value is specified on the HX11 label. If a value less than 16 has been written into the override buffer, then this value becomes the transmitted ID, and is referred to as caller ID.

CtlByte = 2 (mode bit 1 set)

Once the time elapsed from synchronization equals time specified by the txDelay, the device transmits the contents of its own ring-buffer over the serial lines.

CtlByte = 3 (mode bit 0 and 1 set)

Once the time elapsed from synchronization equals time specified by the txDelay, then the device transmits an ultrasonic ID. This ID value is specified on the HX11 label. If a value less than 16 has been written into the override buffer, then this value becomes the transmitted ID, and is referred to as caller ID. The device will also clear its own timers, and transmit the contents of its own ring buffer over the serial lines. This operation repeats itself at interval delay equal to txDelay.

CtlByte: Mode bit 2 set

Transponder Disabled. The device will not transpond to any caller.

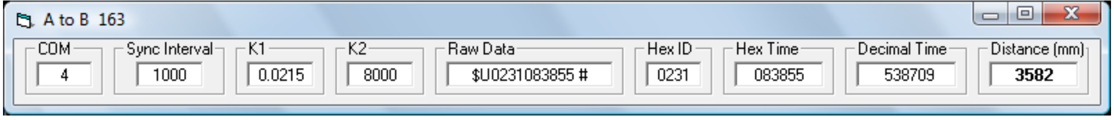
Using the A to B program

The A to B program is provided to help explain how to use the hx11 devices. It is designed for the basic point to point operation, in two cases.

Case one device A and device B must both to be wired (RS232) to a computer or a microcontroller. This is a synchronized operation where the A to B program synchronizes the devices periodically, at rates defined by the Sync Interval.

Case two the devices must be in a transponder mode. In this case one device is not connected to a controlling device (it is free moving). In transponder mode synchronization is not used or required.

Com, Sync Interval, K1 and K2 are all user-defined variables.



A to B 163

COM	Sync Interval	K1	K2	Raw Data	Hex ID	Hex Time	Decimal Time	Distance (mm)
4	1000	0.0215	8000	$U0231083855 #	0231	083855	538709	**3582**

Sampling rate and Sync Interval (Case one only)

The Sync Interval is entered in terms of milliseconds. Once the proper value has been selected, the user can double click on the Sync Interval value window and the sampling rate will change.

Synchronization character for the Hx11 is $ followed by any unreserved character.

For the wireless (transponder) operation the Sync Interval should be 0 (no sync).

Data Logging

To store the distance value on a file, double click on the Distance Value Window.

Calibration

Both txDelay and xpDelay (see the Config. program) are embedded in the time of flight of the signal. I.e. the Hex, Decimal time will be altered by these two constants. Since these are constants they can be calibrated out of the system as follows.

Simple Calibration

Given that the clock rate of the hx11 is 16000000 counts/sec and sound travels 344000 millimeters/second. We have 344/16000 or 0.0215 millimeters/count. This value varies slightly with temperature. The distance time relationship is as follows.

Distance = K1*time – K2

Set K2 = 0

Set the devices A and B at a known distance from each other and note the (K1*time) in the distance window.

Hence:

K2= Distance Window – Known Distance

More elaborate Calibration

Set A and B apart by say 3 meters

Note the Decimal Time Window value (DTWV) and Known Distance (KD)

KD=3000 and DTWV(1) = #1

Set A and B apart by say 6 meters

Note the Decimal Time Window value (DTWV) and Known Distance (KD)

KD=6000 and DTWV(1) = #2

Using the relationship **Distance = K1*time – K2** we now have enough information to calculate K1 and K2.

K1 = (6000 –3000) / (#2 – #1)

Knowing K1 then K2 can be calculated

K2 = K1 * #1 – 3000

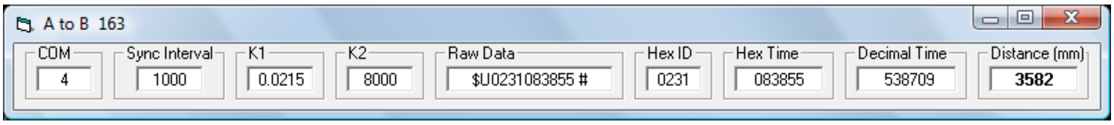
Enter K1 and K2 into their respective windows and your distance value should reflect the true distance from A to B.

Consider the two setup files below

Device Id 11362 … Transmitter ID 231 … Transponder ID 31
Device Id 11364 … Transmitter ID 232 … Transponder ID 32

Distance A to B.set (see application notes)

```
This is a point to point example where both devices are wired
to the serial port, send $x where x is any character, device
11362 will transmit a signal at the end of (txDelay 11). Device
11364 will measure the time of travel and output the distance
to device 11362 at the end of (txDelay 128).
|
11362         1            11          1           33
11364         2           128          1           33
```

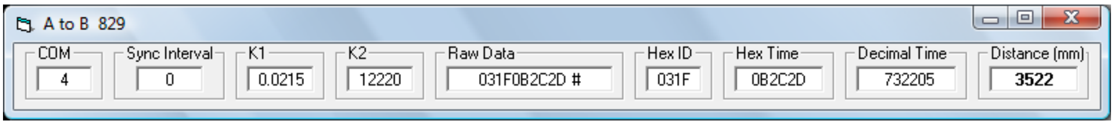


A to B 163

COM	Sync Interval	K1	K2	Raw Data	Hex ID	Hex Time	Decimal Time	Distance (mm)
4	1000	0.0215	8000	$U0231083855 #	0231	083855	538709	**3582**

Distance A to B wireless.set (see application notes)

```
This is a point to point example where A and B are not wired
together. Device 11362 transponds to 11364 call.
|
11362         0            80          1           33
11364         3           255          1           15
```



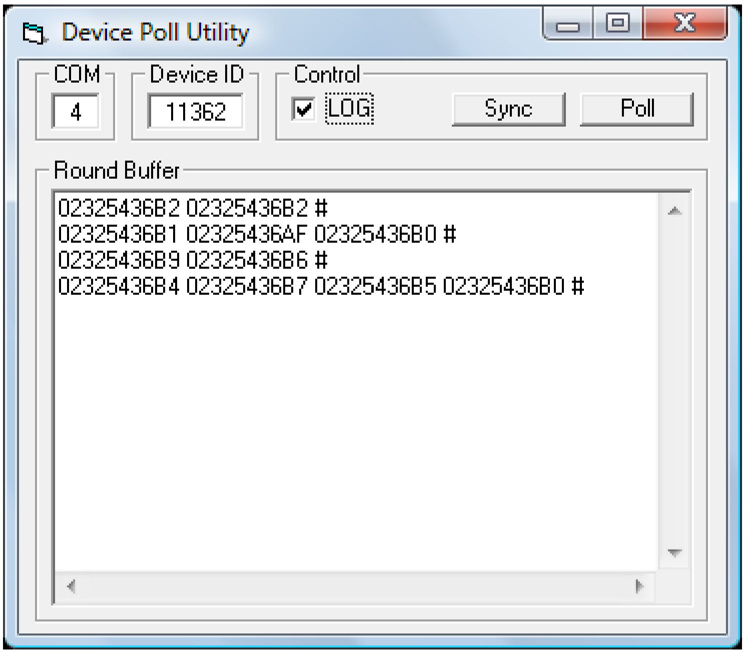
A to B 829

COM	Sync Interval	K1	K2	Raw Data	Hex ID	Hex Time	Decimal Time	Distance (mm)
4	0	0.0215	12220	031F0B2C2D #	031F	0B2C2D	732205	**3522**

Other than one configuration using sync, and the other one not, the major difference is in the resulting Hex ID of these two configurations. Note that in the latter configuration 11364 has been given the caller ID 15, or F in hexadecimal terms. The Hexadecimal tag ID of device 11362 is 231(hex) and its transponder Id is 31. In the wireless configuration, device 11364 reports that its call signified as F, has been bounced back by transponder 31. This allows the user to single out the response, even if there are many other signals detected from other transponders.

The Device Poll Utility

The devicePoll program is a utility for inspecting the ring buffer of a device. A feature that allows the contents of the ring buffer to be logged on a file is provided. The name of the file is devicePoll.txt, and it can be viewed using a normal text editor.



COM: 4

Device ID: 11362

☑ LOG

Round Buffer

```
02325436B2 02325436B2 #
02325436B1 02325436AF 02325436B0 #
02325436B9 02325436B6 #
02325436B4 02325436B7 02325436B5 02325436B0 #
```

Program datalog.exe (network controller required)

This program reads the data stream directly from the network controller. And stores the data on file. This file can then later be converted to timeline tag positioning data, using the program xyz in mode 1. If the program data is running; then xyz.exe running in mode 3 can tap into the data stream real time, while data from the network controller is being stored on file monData.hxm.

Please refer to the Data display window below. The first 4 digits in the data string are the hexadecimal representation of the monitor's ID. Each tag is represented with a total of 10 hexadecimal digits. The first 4 contain the tag's ID number and the remaining 6 represent the position timing data. If the same tag exists in both upper and lower acquisition cycles (see the data program window), then the tag transmission rate is to high. If the xyz.exe program is used to extract 3D position out of the monData.hxm where this rule is violated, it will probably crash. Sometimes a monitor can pick up the same tag twice in the same acquisition cycle, the xyz program treats the latter pickup as echo of the first and it is ignored.

CRITICAL OPERATION RULE

A monitor or monitors should not pick up the same tag in two consecutive acquisition cycles. Run the tags at the slowest transmission rate at first, and then gradually increase the rate, to find the limit.

Data display window

Accepted	Banned	Accepted
4E50 000218355B #	4E50 0002618BCC #	4E50 00031399AC 00021507DB #
4E52 0002186B6F #	4E52 000261C1E8 #	4E52 000313D987 0002153DE9 #
4E54 0002185317 #	4E54 000261A980 #	4E54 000313BD38 0002152578 #
4E56 0002184403 #	4E56 0002619A75 #	4E56 0003139A33 000215167B #
4E50 000348DBFA #	4E50 0002103663 #	4E50 #
4E52 0003491BC6 #	4E52 0002106C6C #	4E52 #
4E54 000348FF83 #	4E54 000210540E #	4E54 #
4E56 000348DCC6 #	4E56 0002104500 #	4E56 #

If the xyz program cannot evaluate a position it returns 0 for the scanned coordinate. Setting the reference coordinates at a nonzero value, and making (0,0,0) an impossible position, evaluation failures can be ignored from a plot.

If ? follows the monitor rather than #, then that specific monitor is not replying to the polling of the Network Controller. In this case make sure the connection is good and the monitor is turned on. **NOTE: the network controller polls only the receivers specified in the layout.hxm file.**

Note that the rule of operations specified above, also applies to the xyz program in all modes of operation.

The XYZ program

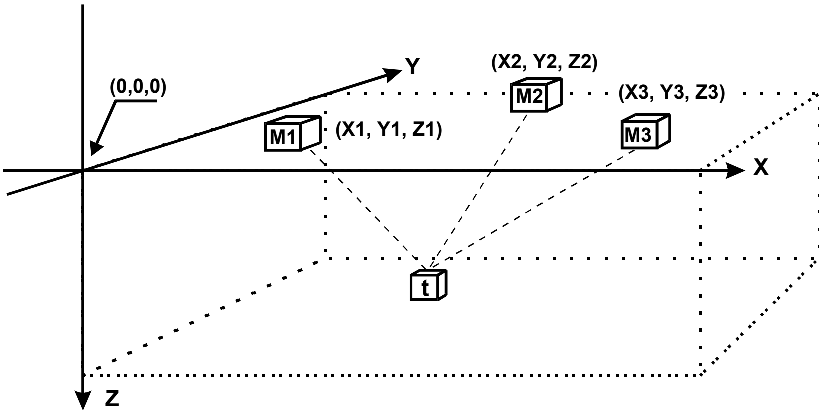
For real time operation a network controller must be used for both datalog, XYZ and the uTrack programs. The XYZ program converts the position of multiple transmitters into actual 3D coordinates (X,Y,Z). It reads the time-tag of the transmitters, either directly from the HX11C (network controller) or from a file with the time-tag information such as monData.hxm. This file is created by a program called datalog. The XYZ program can also gather time-tag information through DDE and convert to 3D coordinates. It does this by comparing time-tags from several monitors (receivers), and computing the X, Y, Z coordinates from the time difference.

This data tag id, (X,Y,Z), time of arrival, is stored on a file named by the time and date the file was created. The user can tap into all data as it is being created by the xyz.exe program; for real time monitoring. This is done using DDE (Dynamic Data Exchange) see (DDE example) in the software section.

Using the XYZ program

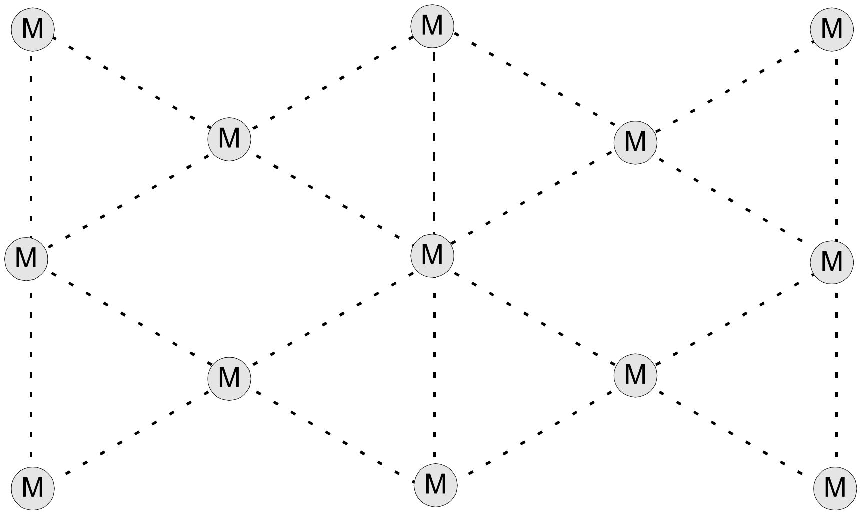
The XYZ program requires the (x,y,z) coordinates to be specified for each receiver, on the network. These coordinates can be entered into a text file (layout.hxm). The xyz program reads this file, and relative to origin (0,0,0) placed at the users convenience, it calculates the position of the transmitters in range based on these coordinates. All positions should be in the positive x and y quadrant. The tag coordinates and time of encounter are stored in directory [xyz Data]; on a file with a name given by time and data. Other program applications have access to the xyz data stream via DDE (dynamic data exchange).

The image below, suggests how overhead monitors can be set up for example in a room where the 0 (xy) plane is the ceiling.



Wide Area Coverage

The illustration below, suggests how a large area can be covered, using monitors. Any formation can be chosen. But the following formation will provide one of the best coverage per unit cost. In the following the D distances between monitors are everywhere the same. The monitors form 60-degree triangles. Depending on the density of the network, many monitors may contribute to the positioning of a given tag. The higher then number of monitors that detect the tag, the more accurate and stable will be the positioning.



Setting up for Z evaluation

The HX11 operates on equidistance principle. It assumes there exist equidistance lines between the monitors (receivers) which all cross where the tag or transmitter is located. Imagine the receivers set up in a triangle, and the transmitter located at equal distance from them all. In this case equidistance exists along a line right through the triangle. Therefore the fourth receiver is needed to fix Z to a point rather than a line. If Z needs to be tracked, don't set the receivers all on the same plane.

Data Evaluation and Errors

If the xyz program is unable to calculate coordinates, some of the X, Y or Z coordinates will read zero. Setting the receiver layout so that zero is an impossible coordinate (i.e.) at zero the transmitter is out of range, will help programs disregard unsuccessful scans. The last number in the data string i.e the result from the xyz program's position scan indicates how many lines were available to construct the coordinate. If this number is less than 3, what ever coordinates resulted should not be considered valid. Note that tags can collide, data from two or more tags may intercept and cause an erroneous reading. If a tag isn't picked up by more than two monitors (lines available number < 3); and it is picked up in one acquisition cycle (only) over a period of some time, it should be disregarded.

XYZ Version 2. program description

The XYZ needs four files to run, **these files must be located in the directory where the xyz.exe is located**. These files are:

MonData.hxm	(mode 1 only)
Layout.hxm	(must access in all modes)
Parameters.hxm	(must access in all modes)
Sspeed.hxm	(must access in all modes)

MonData.hxm is created by a program called data.exe. All files with hxm extension are text files which can be read and modified using Notepad.exe.

Layout.hxm, can be created by the user, or it can be created using the layout.exe program. First column is the monitor's X coordinate, the second column is the Y coordinate and the third column is the Z coordinate. The fourth column is the tag's ID.

File: Sspeed.hxm

XYZ looks for a speed of sound parameter in a file called sspeed.hxm on startup. In the sample file the speed of sound is set at 340 (m/s). Changing this value, will effect the outcome of the (xyz) coordinate calculation

File: Layout.hxm

```
2040    1440    0    20048
2950    1000    0    20050
2180     450    0    20052
1000    1000    0    20054
```

Coordinates are always entered in millimeters; the above translates to:

Monitor 20048 location: (x,y,z) (2040,1440,0)
Monitor 20050 location: (x,y,z) (2950,1000,0)
Monitor 20052 location: (x,y,z) (2180, 450,0)
Monitor 20054 location: (x,y,z) (1000,1000,0)

Suggestion

For testing using the parameter file below, set the monitors on a XY cross with about 2 meters between monitors on the XY axis. This is just a suggestion; any formation can be evaluated.

File: Parameters.hxm

The numbers in the following are arbitrary setting, these numbers should be set to suit the setup.

```
5           "Line1: Total Number of Monitors on this network "
1           "Line2: delay between updates"
1           "Line3: Communications Port 1,2,3,4,5,6 and etc"
1           "Line4: 1 = high and 0 = low communication speed"
1           "Line5: Mode 0 through 2"
256         "Line6: Scan Step"
1000.5      "Line7: Scan Tolerance"
10000       "Line8: Maximum Distance Between Monitors"
-200        "Line9: X scan range start in mm"
+200        "Line10: X scan range end in mm"
-200        "Line11: Y scan range start in mm"
+200        "Line12: Y scan range end in mm"
2000        "Line13: Z scan range start in mm"
4000        "Line14: Z scan range end in mm"
3           "Line15: viewing 1=on 0=off"
4000        "Line16: Viewing window full scale X"
4000        "Line17: viewing window full scale Y"
6000        "Line18: viewing window full scale Z"
0           "Line19: viewing window offset X
0           "Line20: viewing window offset Y"
0           "Line21: viewing window offset Z"
```

Parameters Description

Line1: Total number of monitors on the network

This value is crucial, and must be updated if a monitor is added or removed from the network. The xyz program and data both rely on this number.

Line2: Sampling Rate Control (whole network)

For the standard systems the Interval between updates is calculated as follows:

Interval Between Updates= 0.131072s * (Sampling Rate Control Value)

The (Sampling Rate Control Value) must be an integer number from 1 to 255. If faster or slower update rates are desired, contact Hexamite; the HX11NC network controller needs to be modified or replaced. When the interval times out, data is recovered from all monitors on the network.

Line3: Communication Port

The value of this line selects the serial communications port used, Com1, Com2 …, ComN. This parameter is an integer value representing the value of a free comport on the monitoring computer.

Line4: 1 = high and 0 = low communication speed

Not applicable, doesn't matter what you put here.

Line5: xyz Operation Mode

The value of this line can be 0, 1 or 2; these values select one of the following modes.

Mode 0: (Real Time Mode)

In this mode the program taps directly into the data stream from the Network controller HX11NC, this data is converted to xyz time coordinates and made available via DDE. See visual basic sample program DDEexample below section DDE interface. Note that in real time mode the data is not time tagged. Since this is real time the computer clock can be used to log time. In case of many tags and monitors working at high sampling rates, the computer may not be able to keep up in this mode. Therefore the communication buffer will fill up and eventually overflow, this will crash the program. Operations should be restarted at lower sampling rate with fewer monitors and tags. In case high-speed real-time operations are needed, monitor networks should be divided among more high-speed computers. The computers can then be networked together for high speed and large system operations.

Mode 1: (Filing Mode)

Data from file called MonData.hxm is converted to tag positions (x,y,z) on a timeline and stored in a subdirectory [xyz Data]. If this subdirectory doesn't exist on your computer the xyz program creates it. The name of the processed positioning data becomes the time and date the data was acquired. For example:

Jun 09 040609143505.xyz is actually the date June 9th 2004 and the time is 14:35:05.

If the format of this name is changed, the layout program will not be able to display the tag and it's position on the background image.

Mode 2: (Real Time Mode.)

In this mode the xyz program gets it's data from another program called data. This program must be up and running before xyz is executed in this mode. And like in mode 0 the xyz coordinates are available through DDE using the DDEexample program. In this real time mode falling behind is no problem. Program data taps into the data stream from the network controller hx11nc and stores the data in raw form on a file called monData.hxm. While this data is being stored on the file; the xyz program can monitor the data, convert it to xyz coordinates and make these coordinates available via DDE to the user for real time applications.

Line6: Scan Step

The computer has to scan for the rough x,y,z positions. This number represents the size of the scan step in mm. If for example the largest distance between two monitors detecting a specific tag is 2000mm. Then selecting this value as 100mm means 20 scans will determine the position. But in this case if for example there are 3000mm on the X axes, 2000mm on the Y axis and 500mm on the Z axis then the total number of steps are 30x20x5 or 3000 steps total.

Once the rough position is found the computer iterates for the fine position 8 times cutting the scan step finer by the factor of two every time. For example if your scan step is 128 you have.

Iteration 1.	128/2 or	64mm
Iteration 2.	64/2 or	32mm
Iteration 3.	32/2 or	16mm
Iteration 4.	16/2 or	8mm
Iteration 5.	8/2 or	4mm
Iteration 6.	4/2 or	2mm
Iteration 7.	2/2 or	1mm
Iteration 8.	1/2 or	1/2mm

Experiment with this value to see what suits you best.

Line7: Scan Tolerance

The computer conducts minimum square error evaluation for the convergence of the coordinates. The resulting minima is ErrorX^2 +ErrorY^2 +ErrorZ^2. If the minima is greater than Scan Tolerance value that coordinate calculation is ignored.

Experiment with this value to see what suits you best.

Line8: Maximum Distance Between Monitors

This value tells the program the maximum allowable distance between the monitors which detect a given tag. For example. If there is a large setup and Tag 500 is detected by the monitors surrounding it. And it is detected by an error by a monitor elsewhere in the setup, perhaps 20 or more meters away. The XYZ program will attempt to include the error detection in the calculation for the position of Tag 500. This could delay the calculation for the point significantly, and give erroneous results. It is not likely that Tag 500 is picked up by an error, but it can happen.

Line9: X scan range start in mm

If for example two monitors pick up tag 100. Where one monitor is on the X axis at 20400 (mm) and the other at 22400 (mm). Given that this value equals (- 300) then the scan for the X position of tag 100 starts at 20100.

Line10: X scan range end in mm

Please refer to the example for line 9. This value determines where the scan for tag 100 ends. Given that this value equals (+200), the scan for tag 100 ends at X equals 22600 (mm)

Line11: Y scan range start in mm

If for example two monitors pick up tag 100. Where one monitor is on the Y axis at 20400 (mm) and the other at 22400 (mm). Given that this value equals (- 300) then the scan for the Y position of tag 100 starts at 20100.

Line12: Y scan range end in mm

Please refer to the example for line 9. This value determines where the scan for tag 100 ends. Given that this value equals (+200), the scan for tag 100 ends at Y equals 22600 (mm).

Line13: Z scan range start in mm

If for example two monitors pick up tag 100. Where one monitor is on the Z axis at 20400 (mm) and the other at 22400 (mm). Given that this value equals (- 300) then the scan for the Z position of tag 100 starts at 20100.

Line14: Z scan range end in mm

Please refer to the example for line 14. This value determines where the scan for tag 100 ends. Given that this value equals (+200), the scan for tag 100 ends at Z equals 22600 (mm).

All axis are treated equally. The setup in the parameters file is geared towards an office environment where the Z position of the monitors is 0. And the Z position for the tags can only be between 2 and 4 meters from the ceiling, this cuts down on processing time.

Line15: Viewing

This value enables graphic viewing of the xyz positioning in any mode. Please note that this feature slows the processing down considerably. The computer cannot process as many tag positions per second with this feature enabled, therefore real time operations risk falling behind.

Line 16 and 18: View Window Scaling

The value of line 4 and 5 is the full scale of the viewing window in terms of millimeters.

Lines 19 and 21: View Window Offset

The value of these lines will pan the viewing window left/right or up/down.

DDE Interface

Below is a simple DDE link program written for visual basic called DDEexample. This program reads results from the xyz program running in any of it’s three modes. The results from this example are stored on file called “myFile.xyz”. The scan quality indicates the number of equidistance lines extracted from the data program. If this number is under 3 the coordinates are invalid and will probably be all zeros.

```
Private Sub Form_Load()

  Text1.LinkTopic = "xyz|HX11"
  Text1.LinkItem = "text3"
  Text1.LinkMode = 1          'link mode must be set last
  Open "myFile.xyz" For Output As #1

End Sub

Private Sub Form_Unload(Cancel As Integer)

  Close #1
  End

End Sub

Private Sub Text1_Change()

  myTag = Val(Mid(Text1, 1, 4))
  myX = Val(Mid(Text1, 6, 6))
  myY = Val(Mid(Text1, 13, 6))
  myZ = Val(Mid(Text1, 20, 6))
  scanQuality = Val(Mid(Text1, 27, Len(Text1) - 1))
  Print #1, myTag, myX, myY, myZ, scanQuality

End Sub
```

Program Layout.exe

Layout requires the following file for operation

Layout.gif: This should be the image background (or floorplan) on which the tags and monitors are positioned. The user must provide this image, and it must reside in the same directory as layout.exe. This must be a gif format.

Layout creates the following files

Layout.scl: This file stores the scaling information. The image on the screen needs to be scaled, in order for the monitor layout and the position of the tags to be properly and proportionally displayed.

Layout.hxm: This file contains the layout of the monitors, and is used by the xyz program to calculate the position of the tags.

The Layout Program
Moving the Background Image

Double click on the image and hold the left mouse button down while moving the mouse.
Scaling

Before saving the scale parameters, the text boxes Origin and XY scale must be placed in the upper left hand corner and bottom right hand corner respectively. The XY distance from Origin to XY scale can be entered by imperial or metric notation. The values must be entered using the enter key. Please make sure the values are entered and the Origin and XY boxes are in place before saving the scale.
Erasing Monitors

Double click on the monitor to be erased and then press the Delete Key.
Moving Moveable Objects

Double click on the object and hold the left mouse button down while moving the mouse. Note that the <u>upper left hand corner of the object is the locus</u> of the object.
The Reference Object

This object can be placed anywhere on the image, and the relative coordinates in the status line on the bottom indicates the distance from the reference to the mouse.

Real Time Display on Background Image

The layout program can be used to display the position of the monitors real time. Positions are acquired via DDE from the xyz program. The xyz program must be running in the background for this feature to operate.

Monitor Formation (Setup Hints.)

The formation of the monitors should suit your application, if the application requires high sampling rates, precision and accuracy may have to be reduced. The higher the number of monitors picking up a tag (or the greater the number of lines plotted by the xyz), the longer will the xyz program take to calculate the position. For most 2D applications, only 3 monitors will be needed to pick up the tag. For high-speed 2D real time operation, the formation should tend to keep only 3 monitors in range. 3D operations will require more monitors to pick up the tag, and iteration scans will be needed to evaluate the Z position.

IMPORTANT

All files used by hx11 must reside in the same directory, 4 separate programs should be installed. Use install.bat it will guide you through the process. All files are installed in directory:
\program files\hx11

Caution:

Make sure either serial port 1 (com1) or serial port 2 (com2) is available for the hx11

Power the network controller and then network simultaneously. If the receivers are powered on and no controller is retrieving data from the receivers, the receiver buffers may overflow. If this unlikely condition occurs, power down the receivers and restart.

Helpful Hint:

The processing time of the xyz program is taken by code execution and number crunching rather than peripheral access. Increasing the speed of the microprocessor effectively increases processing of the xyz coordinates.

If the power is momentarily interrupted, (try to avoid this) the whole system will reboot, data will attempt to reconnect. If this happens it may become necessary to restart the whole operation.

HX11 Easy start

A. Make sure all the programs and files are in the same directory (hx11), and the network controller is connected to your computer and the receivers.
B. Place the receivers in a square formation one for each corner of the 2m x 2m square. Make sure there are close to 2 meters between the sensing elements on the receivers.
C. Place a tag somewhere in the square 2-4 meters away from the receiver plane (ceiling).
D. Set the tag at the second or third slowest transmission rate
E. Run the data program and collect positioning data for a few minutes.
F. Shut the data program down and run the xyz program. (during DDE both program run simultaneously)

The following files must be in the hx11 directory where all the programs should be.

USE A TEXT EDITOR TO OPEN layout.hxm MAKE SURE THE TAG IDENDIFICATION NUMBERS CORRISPOND TO YOUR RECEIVERS. AND THAT THE RECEIVER'S COORDINATES IN RELATION TO THE OTHER RECEIVERS ARE CORRECT.