# ISAN - Integrated System for Autonomous Navigation

Collective Public Transmission

| | |
|---|---|
| Document author | : <D:\Solonerus>, <C:\Strikeeaglechase>, <C:\Lumi Virtual> |
| Cowriters | : <F:\Zaff>, <H:\Archduke>, <C:\Nordwolf> |
| Licensed under | : GNU General Public License v3.0 or later |
| Date of creation | : 20.06.2020 |
| Version number | : 1.3 |

---

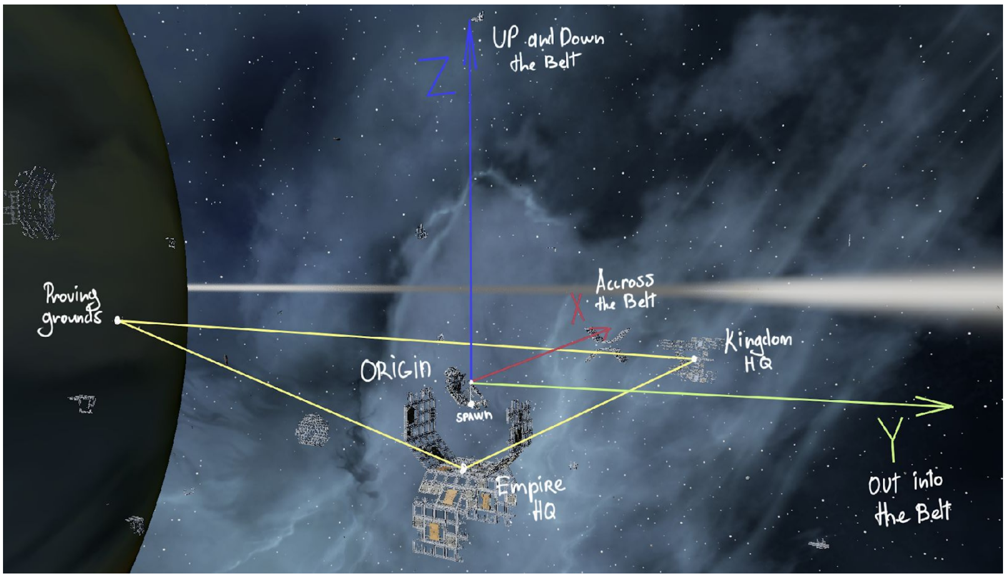
<< PUBLIC RELEASE >>

# TABLE OF CONTENTS

# >> INTRODUCTION

Introduction Video

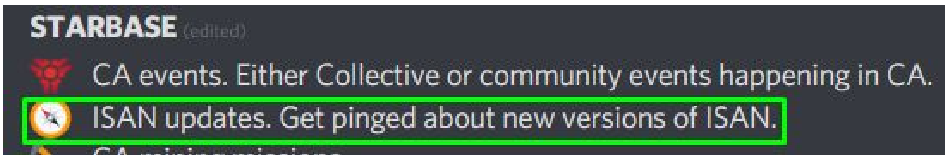
ISAN is a navigation system within Starbase, developed by Collective. When installed on a ship, it calculates your X, Y, and Z coordinates in space, relative to Origin station. Below is a diagram of the coordinate axes.



In the future, we plan to create the necessary infrastructure for ISAN in at least Collective space, for ours, and the Starbase community's continued use. In addition, we are also currently working on:

| | |
|---|---|
| *Universal Orientation Calculation (UOC)* | Complete |
| *Universal Autopilot (UA)* | In Development |
| *Online Starmap (Starmap)* | Complete |

From everyone here at Collective R&D, particularly the ISAN development team, we hope you enjoy ISAN. Subscribe to updates in the **#notifications** channel in the Collective Discord to get notified when a new version of ISAN is released. You can also follow the GitHub to follow development, or contribute to the project!



STARBASE (edited)

CA events. Either Collective or community events happening in CA.

ISAN updates. Get pinged about new versions of ISAN.

CA mining missions

# >> MODULARITY

ISAN is designed to be extendable using addons. To accomplish this, ISAN is split into one **core module**, and any number of optional **addons**. These communicate to each other using a standardised API, allowing you to use any core with any number of addons, you can even create your own! The core module you should use depends on how much space you have on your ship, and how much you care about maintaining coordinate accuracy while moving.

# >> THE CORES

ISAN has two core modules; ISAN Mono and ISAN Quad. Mono is smaller, but inaccurate while moving, while Quad is larger, but more accurate.

| ISAN Core | **Mono** | **Quad** |
|---|---|---|
| YOLOL Chips | **1** | **1** |
| Radio Receivers | **1** | **4** |
| Refresh rate | **~1.4 seconds** | **0.8 seconds** |
| Accuracy while moving* | **~±20%** | **~±1%** |
| Accuracy while stationary* | **~±1%** | **~±1%** |
| Max range from Origin | **~1,000,000 meters** | **~1,000,000 meters** |

**Accuracy decreases with distance from Origin. Within 100,000 meters, accuracy is around ±0.1%.*

# >> SETTING UP ISAN MONO

Tutorial Video

You need a **receiver** (either size is fine), placed anywhere on your ship, an **advanced** quality or better **YOLOL chip**, a **memory chip**, and a **text panel**. Make sure they are all connected to the same cable network.

Rename these three fields in the **receiver**:

- “Message” to **N1**
- “SignalStrength” to **R1**
- “TargetMessage” to **M1**

Set the value of:

- “ListenAngle” to **180**

Include these six fields in the **memory chip**:

- **XX**, **YY**, **ZZ**, **CL**, **SD**, **Pos**

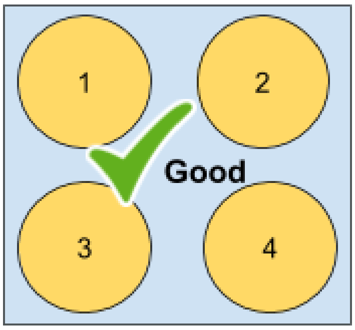
Open the device fields of the **text panel** and rename:

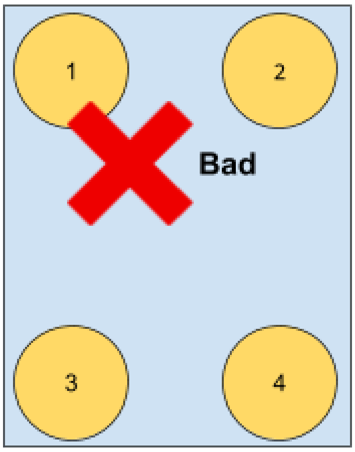
- “PanelValue” to **Pos**

Next open the YOLOL chip and, one line at a time, paste in the code below. Once done press enter to save, and you’re all set! *(Make sure you are on line 1 before pasting)*

```
1   a=19832.688 b=29157.219 c=-50445.701 d=9472.525 e=-4904.478 f=-801.149
2   k="station_hq_imperial_a" l="station_hq_kingdom_a"
3   m="station_proving_grounds" n="station_capital_imperial_a"
4   er="\nX:Loss\nY:of\nZ:Signal" ax="\nX:" ay="\nY:" az="\nZ:" o=10^6-1
5   ox=-9938.401 oy=4904.714 oz=0
6   :CL=0 g=o-:R1 :SD=(g<o)+(h<o)+(i<o)+(j<o) :M1=l u=0 v=0 w=0
7   :M1=m h=o-:R1 u++ x=(g^2-h^2+a^2)/2/a u/=(u<5)/(:N1!=l) :M1=l goto7
8   :M1=n i=o-:R1 v++ ya=g^2-i^2+b^2+c^2 v/=(v<5)/(:N1!=m) :M1=m goto8
9   :M1=k j=o-:R1 w++ y=(ya-2*b*x)/2/c w/=(w<5)/(:N1!=n) :M1=n goto9
10  :XX=x+ox :YY=y+oy :ZZ=(g^2-j^2+d^2+e^2+f^2-2*d*x-2*e*y)/2/f+oz
11  :CL=1 :Pos=ax+:XX+ay+:YY+az+:ZZ if :SD<4 then :Pos=er end goto6
12  // ISAN by Collective.
13  // YOLOL: Strikeeaglechase, Lumi Virtual, Solonerus; Coords: Nordwolf
14  // Documentation: https://tinyurl.com/collective-isan // MONO 1.1
```

# >> SETTING UP ISAN QUAD





You need **four receivers** (either size is fine). Place them anywhere on your ship as long as they're as **close together as possible and even distances**. Deviation will lead to inaccuracy. See figures: ⟶

You need an **advanced** quality or better **YOLOL chip**, a **memory chip**, and a **text panel**. Make sure they are all connected to the same cable network.

Rename these three fields in each of your **receivers** (1 through 4, replacing # with the receiver number):

- "Message" to **N#**
- "SignalStrength" to **R#**
- "TargetMessage" to **M#**

Set the value of:

- "ListenAngle" to **180**

Include these six fields in the **memory chip**:

- **XX**, **YY**, **ZZ**, **CL**, **SD**, **Pos**

Open the device fields of the **text panel** and rename:

- "`PanelValue`" to **Pos**

Next open the YOLOL chip and, one line at a time, paste in the code below. Once done press enter to save, and you're all set! *(Make sure you are on line 1 before pasting)*

```
1   a=19832.688 b=29157.219 c=-50445.701 d=9472.525 e=-4904.478 f=-801.149
2   :M1="station_hq_imperial_a" :M2="station_hq_kingdom_a"
3   :M3="station_proving_grounds" :M4="station_capital_imperial_a"
4   er="\nX:Loss\nY:of\nZ:Signal" ax="\nX:" ay="\nY:" az="\nZ:" o=10^6-1
5   ox=-9938.401 oy=4904.714 oz=0
6   :CL=0 g=o-:R1 h=o-:R2 i=o-:R3 j=o-:R4 :SD=(g<o)+(h<o)+(i<o)+(j<o)
7   x=(g^2-h^2+a^2)/2/a y=(g^2-i^2+b^2+c^2-2*b*x)/2/c
8   :XX=x+ox :YY=y+oy :ZZ=(g^2-j^2+d^2+e^2+f^2-2*d*x-2*e*y)/2/f+oz
9   :CL=1 :Pos=ax+:XX+ay+:YY+az+:ZZ if :SD<4 then :Pos=er end goto6
10  // ISAN by Collective.
11  // YOLOL: Strikeeaglechase, Lumi Virtual, Solonerus; Coords: Nordwolf
12  // QUAD 1.1 Documentation: https://isan.to/doc/
```

# >> VELOCITY ADDON

The velocity addon calculates the speed of your ship when moving in a straight line.

You need to add a new **professional YOLOL chip**, and a new **text panel**. Make sure they are both connected to the same cable network as the core module.

Include these two fields in a **memory chip**:

- **VV, Vel**

Open the device fields of the **text panel** and rename:

- “PanelValue” to **Vel**

Next open the YOLOL chip and, one line at a time, paste in the code below. Once done press enter to save, and you’re all set!

```
1  av="\nV:" e="\nV:LOS"
2  :VV=sqrt((xl-:XX)^2+(yl-:YY)^2+(zl-:ZZ)^2)*(1/(i*0.2)) xl=:XX yl=:YY
3  i=2 zl=:ZZ vd=:VV/vl if ((vd>1.5)+(vd<0.5))>0 then :VV=vl end vl=:VV
4  i++ if :CL>0 then :Vel=av+:VV if :SD<4 then :Vel=e end goto2 end goto4
5  // ISAN by Collective.
6  // YOLOL: Strikeeaglechase, Lumi Virtual, Solonerus; Coords: Nordwolf
7  // Velocity 1.1 Documentation: https://isan.to/doc
```

# >> UOC-MOMENTUM ADDON

Universal Orientation Calculation-Momentum (UOC-M) calculates your heading and pitch while moving in a straight line.

**Heading** is your **angle through the X-Y plane**, from **0 to 360**, increasing as you yaw to the right. **0** is **away from Eos, 180** is **toward Eos**. Here's a diagram.

**Pitch** is your **angle off from the X-Y plane**, from **-90 to 90**. **0** is **perpendicular to the X-Y plane**, **90** is **perpendicular and up towards positive z**, **-90** is **perpendicular and down towards negative z**. Here's a diagram.

You need to add a new **professional YOLOL chip**, and a new **text panel**. Make sure they are both connected to the same cable network as the core module.

Include these three fields in a **memory chip**:

- **HH**, **PP**, **Orn**

Open the device fields of the **text panel** and rename:

- "PanelValue" to **Orn**

Next open the YOLOL chip and, one line at a time, paste in the code below. Once done press enter to save, and you're all set!

```
er="\nH:LOS\nH:LOS" ah="\nH:" ap="\nP:" f=360 q=180 p=0.5
x=xl-:XX y=:YY-yl z=:ZZ-zl h=sqrt(x^2+y^2) xl=:XX yl=:YY zl=:ZZ
:HH=atan(x/y)+q+(y>0)*((x<0)-p)*f :PP=atan(z/h) o=ah+:HH+ap+:PP
if :CL>0 then :Orn=o if :SD<4 then :Orn=er end goto2 end goto4
// ISAN by Collective.
// YOLOL: Strikeeaglechase, Lumi Virtual, Solonerus; Coords: Nordwolf
// ORN 1.3 Documentation: : https://isan.to/doc
```

# >> WNS ADDON

Waypoint Navigation System (WNS) is an add-on that makes going to ISAN coordinates fast and easy. Using this system in conjunction with the UOC makes it easy to point toward a destination. To enter a destination, simply change the value of the **DX, DY** and **DZ** fields in the memory chip to the X, Y and Z of your destination. *It is recommended that you put the memory chip with DX, DY and DZ fields somewhere accessible.*

You need to add a new **professional YOLOL chip**, and a new **text panel**. Make sure they are both connected to the same cable network as the core module.

Include these seven fields in a **memory chip**:

- **DX**, **DY**, **DZ**, **DH**, **DP**, **DD**, **Dst**

Open the device fields of the **text panel** and rename:

- "PanelValue" to **Dst**

Next open the YOLOL chip and, one line at a time, paste in the code below. Once done press enter to save, and you're all set!

```
1  ax="\nX:" ay="\nY:" az="\nZ:" ah="\nH:" ap="\nP:" ad="\nD:"
2  e="\nX:"+:DX+"\nY:"+:DY+"\nZ:"+:DZ+"\nH:ERR\nP:ERR" f=360 q=180 p=0.5
3  x=:XX-:DX y=:DY-:YY z=:DZ-:ZZ h=sqrt(x^2+y^2) :DP=atan(z/h)
4  :DH=atan(x/y)+q+(y>0)*((x<0)-p)*f
5  :DD=sqrt(x^2+y^2+z^2) o=ax+:DX+ay+:DY+az+:DZ+ah+:DH+ap+:DP+ad+:DD
6  if :CL>0 then :Dst=o if :SD<4 then :Dst=e end goto3 end goto6
7  // ISAN by Collective.
8  // YOLOL: Strikeeaglechase, Lumi Virtual, Solonerus; Coords: Nordwolf
9  // WNS 1.0 Documentation: https://isan.to/doc
```

# >> API SPECIFICATIONS

If you plan on making your own addons, these are all the specs you need to know to create a fully compatible addon. Please contribute your module to the repo!

External variables created by a **core module** are:

- **:XX** , **:YY** , **:ZZ**
    - **X**, **Y**, and **Z coordinates** respectively.
- **:CL**
    - **Clock pulse**, which tells addons when to output their data and loop back to their beginnings. Set to **1** on the **last line** in core modules, and is set to **0** on the **first line** in core modules.
- **:SD**
    - **Signal detection**, a number from **0 to 4**, representing **the number of receiver signals that are being received**.
- **:Pos**
    - **Pre-formatted** output for text displays.
- **:R#**
    - **Signal strength** of the #th receiver.
- **:M#**
    - **Target message** of the #th receiver.
- **:N#**
    - **Received message** of the #th receiver.

Requirements:

- Addons can be **at most four lines** long. Additional lines are allowed that do not loop, such as lines dedicated to setting up local variables.
- Use **at least two** characters per external variable.
    - (Correct) **:FO**
    - (Incorrect) **:B**
- (Optional) Include a **pre-formatted text** display external variable, with a **three character variable**.
    - (Examples) **:Pos**, **:Vel**
- The **last line** of your addon should include a way to **loop continuously** on that line until **:CL = 1**, where it should then output the pre-formatted output, and **return to the start** of your addon. See the velocity addon for an example.
- **:XX**, **:YY**, and **:ZZ**, must be **read on or before** the **third line** of your addon.

# >> CREDITS

- **Solonerus** - *Project management*
- **Lumi Virtual** - *Development of the current version of ISAN 'ISAN alchemist'*
- **Strikeeaglechase** - *Development of offsets and ISAN code*
- **MuNk** - *Code consultation*
- **Nordwolf** - *Design of previous iteration of ISAN, coordinate system calculation and measurements*
- **Battle_Wrath** - *Various design ideas and general help*
- **Archduke** - *Invaluable support and document writeup*
- **Zaff** - *Security and usability consultation, documentation*
- **Meboy100** - *Le rubber duck*

# >> USEFUL LINKS

- You can reach this document in your browser by entering "`isan.to/doc`".
- The **ISAN Starmap**, introduced in this **video**.
- ISAN on the **Starbase Wiki**
- ISAN Open Source **Gitlab page**