ISOBUS Virtual Terminal

ISO 11783-6

Contents

Virtual Terminal (VT)

- A graphical display that is able to control different ISOBUS compliant implements from several manufacturers
- The implement ECU stores and downloads the user interface
- Shows information received from the implement ECU
- Transmits the commands from the operator to the implement ECU, such as changing a parameter value

http://ien.kvernelandgroup.com/var/kv/storage/images/kvernelandgroup-corporate-site/brands-and-products/im-farming/isobus/isobus-intelligence/424015-47-eng-GB/ISOBUS-Intelligence.jpg
http://www.prefarm.it/pfp/wp-content/uploads/2012/02/VirtualTerminalJD2630.jpg

Terminology and Versions

- ISOBUS standard defines different versions for Virtual Terminal features
- VT versions are backward compatible
  - Features from previous versions are supported in the later versions
- ISOBUS terminals can be referred as
  - **Universal Terminal (UT), by AEF** or
  - **Virtual Terminal (VT), by ISO standard**

Terminology and Versions

Epec's VT client support	Virtual Terminal – VT (ISO standard)	Universal Terminal – UT (AEF)
Yes	VT 2	UT 1.0
Yes	VT 3	UT 2.0
Partly	VT 4	-
-	VT 5	-
-	VT 6 (draft)	UT 3.0 (draft)

- It is recommended to use VT 3 (UT 2.0) features
  - the most common version at the moment
- Object pool adapts according to VT versions

EPEC

Communication Example

I am a VT →

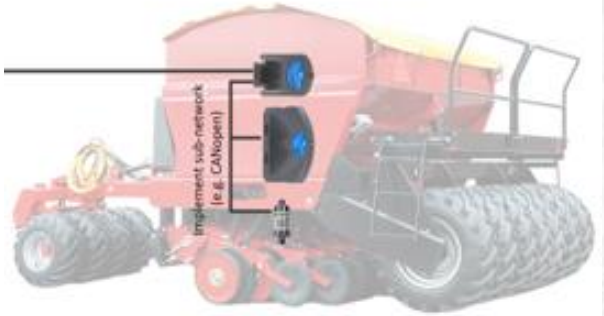
← I am a seeder, what kind of terminal are you?

I speak english and have 10 physical keys, 400x400 px data mask area.. →

← OK, here is my user interface

Operator input (e.g. left marker down) →

← Implement action (left marker down)

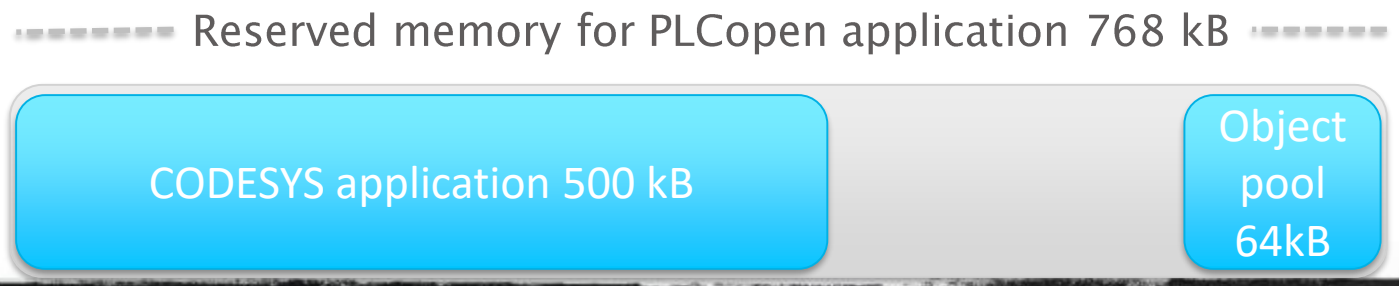


Object Pool

- Is a set of objects that defines the user interface of an implement
  - Consists of, for example, softkeys, buttons, numeric inputs and outputs
  - The more complex the VT design, the bigger the object pool
- Each object has a unique object ID in the object pool → uniquely addressable
- The object pool is transferred to the VT at initialization by the Working Set Master
  - If the pool already exists in VT, it is not transferred

Object Pool

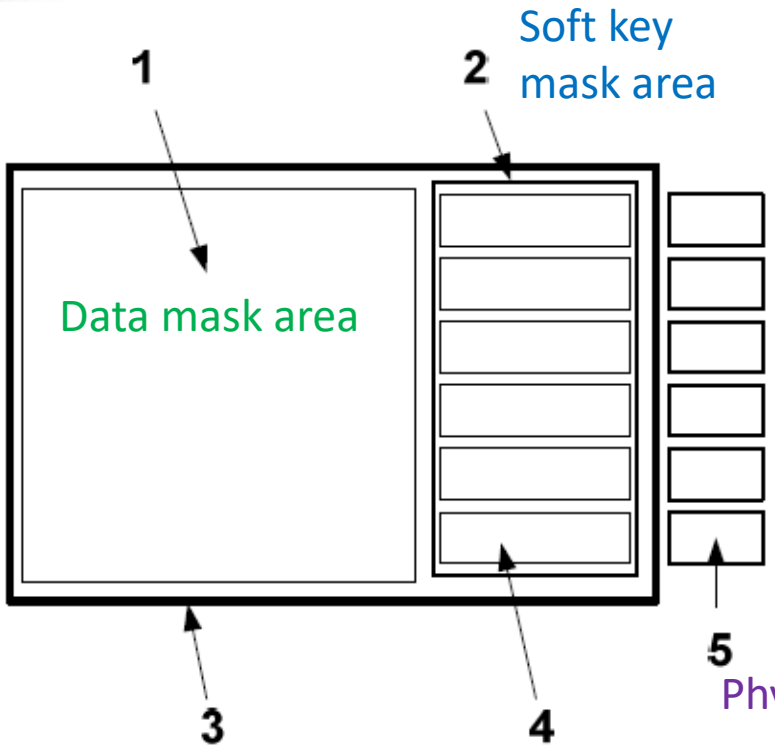
- The object pool is downloaded to the same memory area than the PLCopen application
  - Reserved memory for PLCopen application (3000 series) → 768 kB
  - The object pool size is given in MultiTool (range 64 kB – 256 kB)
  - The downloaded binary file needs to fit into the reserved space

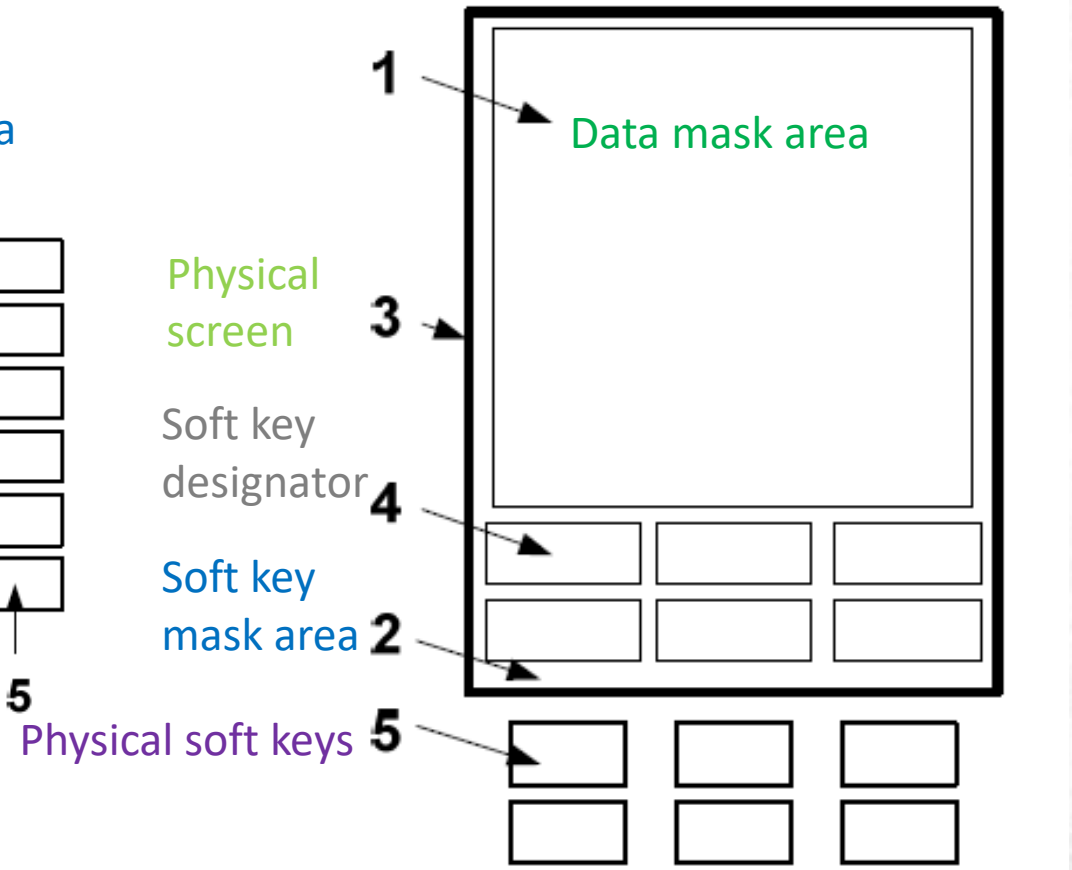


Working Set

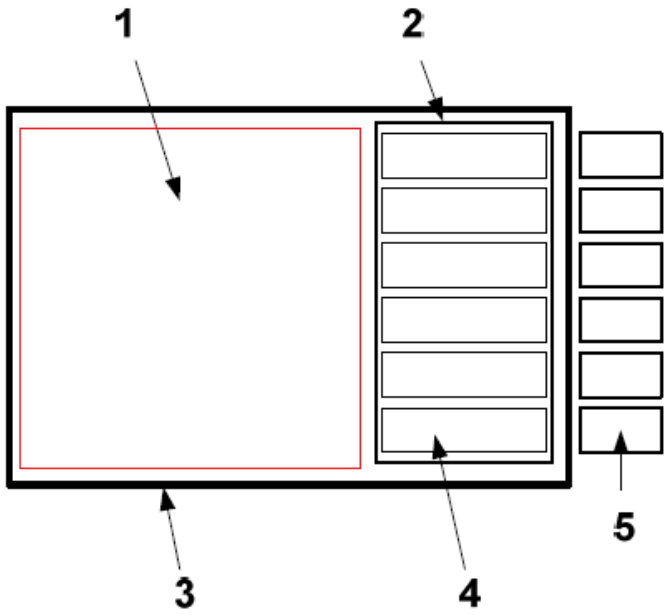
- One or several ISOBUS devices (ECUs) that control the implements functionality
  - Handles the input signals from the operator through the VT
  - Shows the process data to the operator through the VT
- Main device is called the Working Set Master
  - Stores and sends the object pool to the VT
- Working set can also have working set members
- At the moment, Epec libraries support Working Set Master

VT Design and Orientation Examples





Data Mask Area

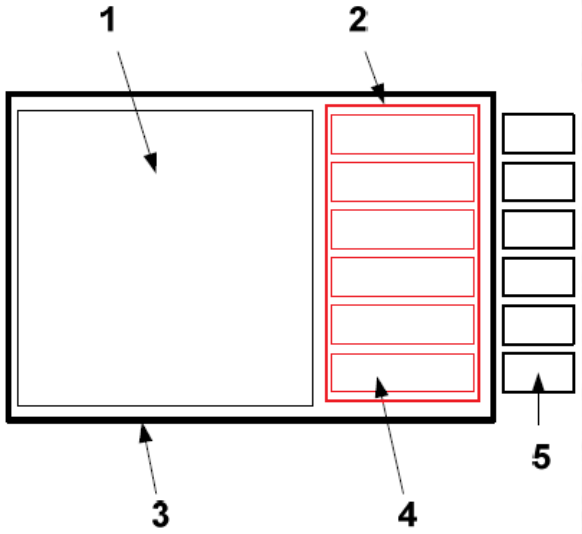


- A square area for displaying information using the data or alarm mask
- Examples of data mask areas
  - 200 x 200 px (minimum)
  - 240 x 240 px
  - 400 x 400 px
  - **480 x 480 px**

  + Any other square dimensions are acceptable

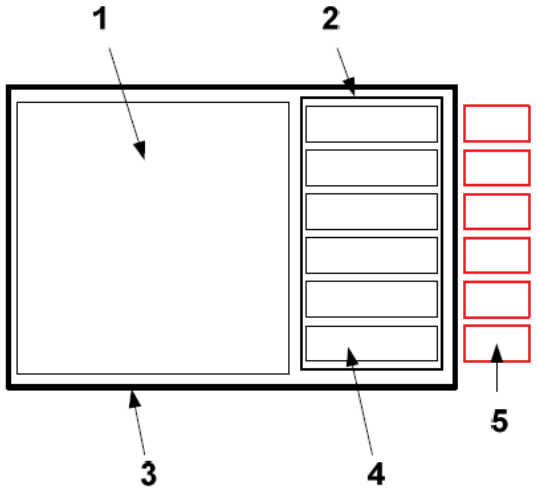
Soft Keys

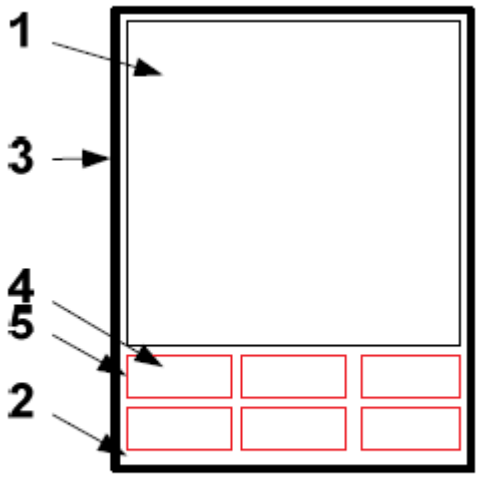
- The *Soft Key Mask* area (2) is reserved for displaying Soft Key labels (designators)
- Separate from data mask area
- Soft key designators (4) may contain text, graphics or both
- Soft keys can be used to, for example, activate events and change the shown mask



Soft Keys

- Physical soft keys
  - the amount of permanently dedicated keys that the VT makes available
  - may be located directly on the touch screen



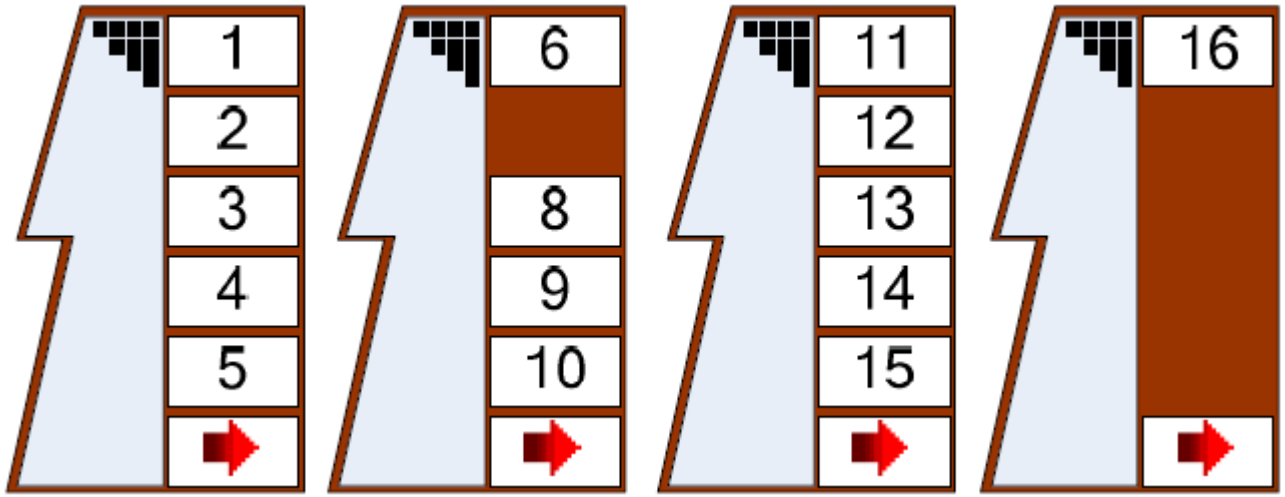


Touch Screen,
Portrait Orientation

Navigation

- If the working set provides more soft keys on a soft key mask than the VT has reported in the number of physical Soft Keys

  → the VT provides navigation



Color Palette

- VTs use the standard 216 color “web browser safe” palette (used by Internet browsers)
- Three color modes
  - 2 color, monochrome (1 bit)
    - (not supported by Epec)
  - 16 color (4 bit)
  - 256 color (8 bit)
- Colors 232 to 255 are proprietary to the VT design to extend the color palette
  - Not recommended to use in implement application

Index	R,G,B value
0 (Black)	00,00,00[a, b]
1 (White)	FF,FF,FF [a, b]
2 (Green)	00,99,00[b]
3 (Teal)	00,99,99[b]
4 (Maroon)	99,00,00[b]
5 (Purple)	99,00,99[b]
6 (Olive)	99,99,00[b]
7 (Silver)	CC,CC,CC[b]
8 (Grey)	99,99,99[b]
9 (Blue)	00,00,FF[b]
10 (Lime)	00,FF,00[b]
11 (Cyan)	00,FF,FF[b]
12 (Red)	FF,00,00[b]
13 (Magenta)	FF,00,FF[b]
14 (Yellow)	FF,FF,00[b]
15 (Navy)	00,00,99[b]

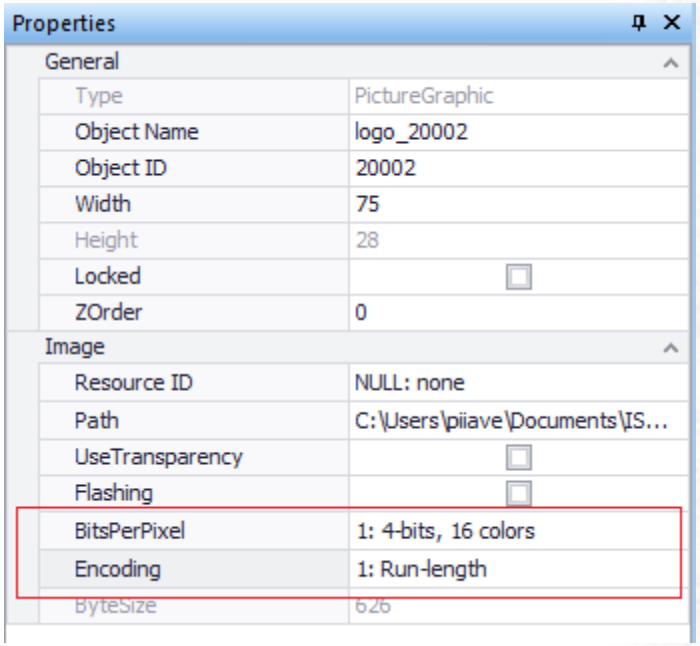
Font Types

Font Type	VT 3 and prior	VT 4 and later
0 = ISO8859-1 (ISO Latin 1)	X	X
1 = ISO8859-15 (ISO Latin 9)	X	X
2 = ISO8859-2 (ISO Latin 2)		X
3 = Reserved		X
4 = ISO8859-4 (ISO Latin 4)		X
5 = ISO8859-5 (Cyrillic)		X
6 = Reserved		X
7 = ISO8859-7 (Greek)		X
8 – 239 Reserved		X
240 – 254 = Proprietary		X
255 = Proprietary	X	X

Images (Picture Graphic)

- The VT scales the picture graphic (bitmap) from the actual width and height to the target width and calculated target height
- 16 color bitmaps are recommended

  → works best in different VTs
- Use correctly scaled images that are designed to fit to the data mask resolution.
- Scaled images also reduces the object pool size
- **Encoding type > Run-length**
  - Compresses the picture



Images (Picture Graphic)

- Epec's VT client library has a sending buffer that limits the image size that can be scaled on downloading
  - If the object size is bigger than the defined constant, the image is downloaded directly from flash without scaling it
- The default buffer size set to 3kB
  - the buffer size can be increased, but it'll require more RAM memory
  - global constant variable in ISOBUS VT library > *G_ISOBUS_VT_POOL_MAX_OBJECT_SIZE_FOR_ONLINE_CHANGE*

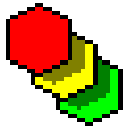
VT Project Workflow

Epec MultiTool

Configure

Jetter ISO-Designer

Design layout



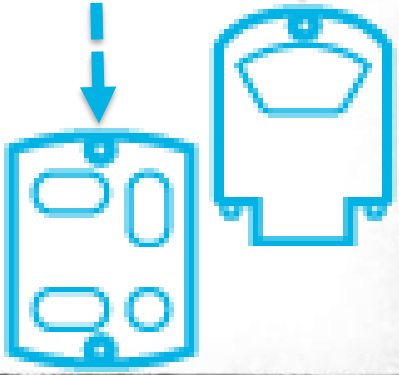
CODESYS

Application code

Epec CANmoon

Download software

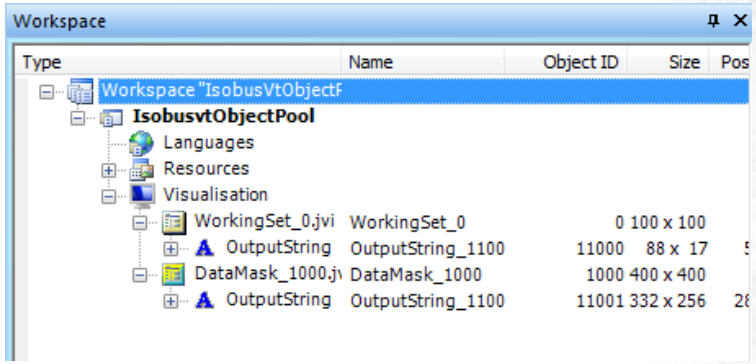
Diagnose



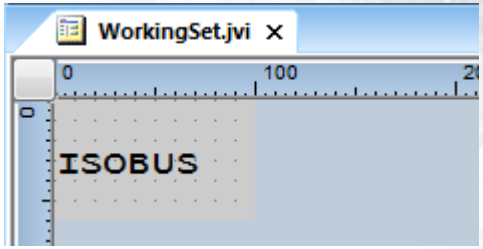
MultiTool

MultiTool creates a device folder including

1. CODESYS project file (.pro)
2. *Libraries* folder
3. *ISOBUS* folder including
   - *Jetter* folder
     - Empty ISO-Designer project template with a Working Set and a Data Mask
   - *Python* folder
     - Scripts that update ISO-Designer updates to CODESYS
     - **BinaryMaker** folder includes the object pool *downloaded.bin* (combined from IsobusVt, IsobusTc, Languages folder)



Design Tips



- Working set needs at least one object
- Soft Keys
  - Square soft keys are easy to scale to different VT sizes
  - To increase soft key size (from default), update
    - Epec MultiTool > ISOBUS Components > VT softkey width / height
    - ISO-Designer project

Design Tips

- Fill and font attributes
  - Defining a fixed set for each size/attribute is recommended
  - Changing attribute will update the change to every element in which the attribute is used
  - Font type **0 = ISO8859-1 (ISO Latin 1)** recommended

Design Tips

- Colors
  - The first 16 color indexes in ISO-Designer are recommended to be used
  - Avoid using the color indexes 232-255; they are proprietary and RGB code may differ between VT manufacturers

Design Tips

- Picture graphic object
  - 16 color bitmaps recommended
  - Run-length encoding for image size optimization
  - Scale bitmaps to correct size before adding them to ISO-Designer project
- Audio output is not always supported

  → VT pool should be usable also without sounds