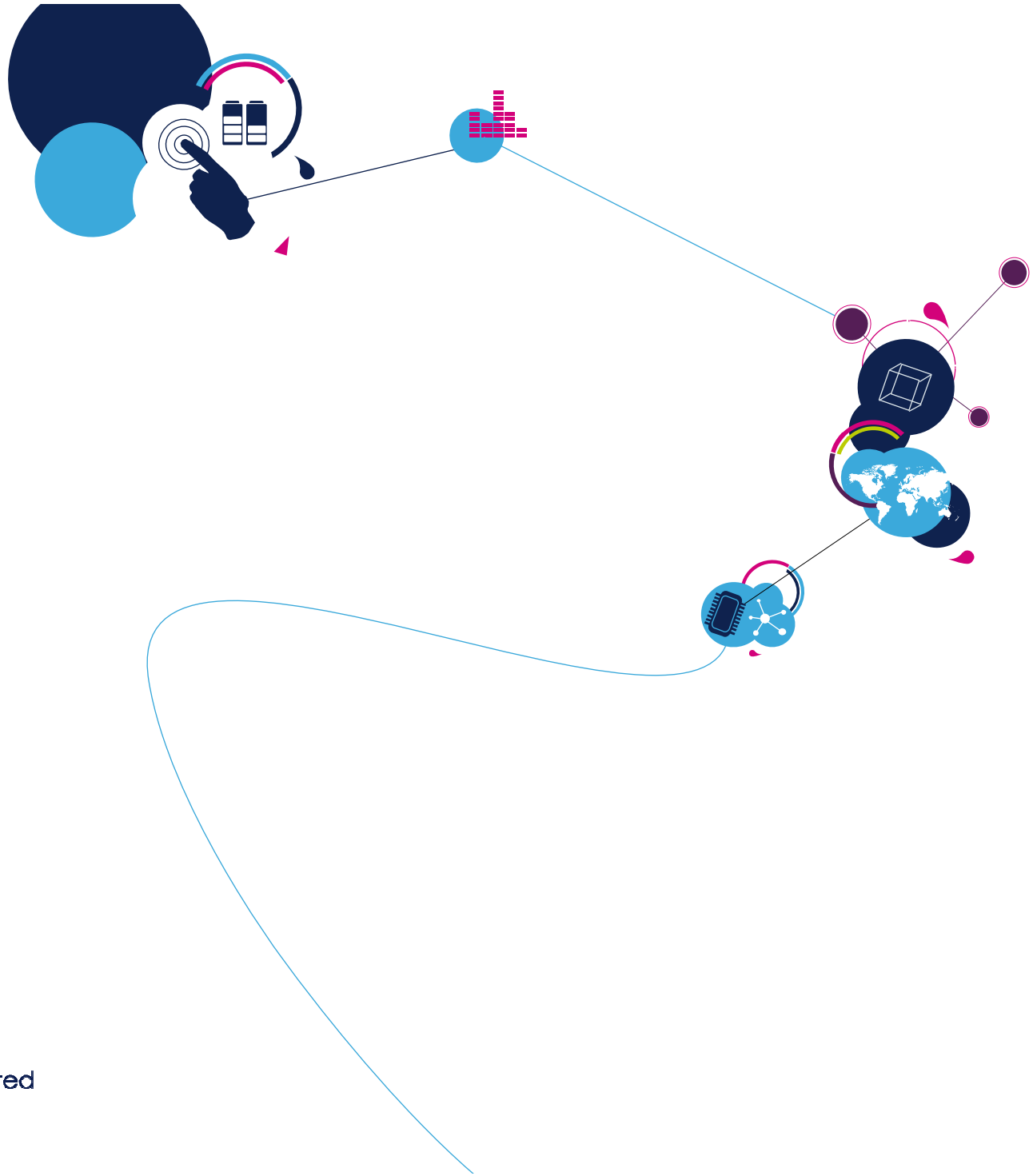


STemWin

life.augmented

CONTENTS

OBJECTIVES

- Provide a high level graphical library which:
  - Is easy to use
  - Is flexible to customize
  - Provides graphical objects from simple 2D elements to complex window driven objects
  - Benefit from Chrome-ART HW features without knowing detail
  - Is the industrial GUI stack in embedded field
  - Suits to project of every size → STemWin is free of charge!

Supported STM32 Target Board

Table 3. Supported EVAL boards and examples

STM32 Series/EVAL board		LCD interface	LCD driver
STM32F1	STM3210B-EVAL	SPI	FlexColor
	STM3210C-EVAL	SPI	
	STM3210E-EVAL	FSMC	
	STM32100B-EVAL	SPI	
	STM32100E-EVAL	FSMC	
STM32F2	STM322xG-EVAL	FSMC	
STM32F3	STM32303C-EVAL	SPI	
	STM32373C-EVAL	SPI	
STM32F4	STM324xG-EVAL	FSMC	
	STM324x9I-EVAL	LTDC	Lin
STM32L1	STM32L152-EVAL	SPI	FlexColor
	STM32L152D-EVAL	FSMC	

life.augmented

Working with Simulator

- Recommend to use simulator:
  - Only PC is needed
  - Easy to discuss with customer
  - https://www.segger.com/downloads.html?pid=26

SEGGER Downloads

The following list contains the latest SEGGER software and documentation downloads.
Please select a product below to view the available files in the appropriate download section.

Product: emWin

emWin

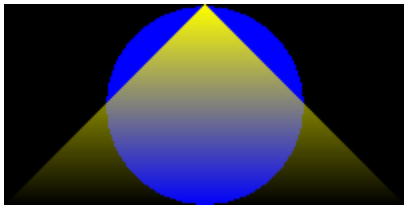
Product manuals and software

PDF	Manual	Version	Size
	emWin Custom Widget Type Creation Guide emWin Custom Widget Type Creation Guide Rev. 0	Rev.0	191.77 KB
	emWin Manual emWin Manual V5.28 Rev. 1	5.28 Rev. 1	13.03 MB
ZIP	**PC Software**	**Version**	**Size**
	emWin Bitmap Converter, Demo version emWin Bitmap Converter, Demo version	5.28	125.85 KB
	emWin Font Converter, Demo version emWin Font Converter, Demo version	5.28a	435.17 KB
	emWin Simulation & Trial version emWin Simulation & Trial version	5.28	14.16 MB

CONTENTS

Basic 2D library – Vector Objects

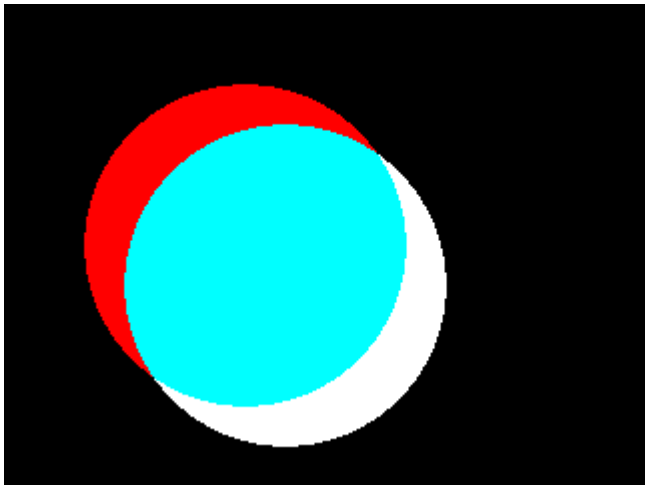
- With STemWin you can easily draw basic vector objects as
  - Lines, rectangles, arcs, polygons, graphs, bitmaps, JPEGs, PNGs and more
- Objects are drawn by a foreground color
  - `GUI_SetColor(GUI_RED);`
- STemWin supports Alpha blending – objects can overlay each other.



- Text and values can be easily written on the screen

Basic 2D library - examples


```
GUI_SetColor(GUI_RED);
GUI_SetDrawMode(GUI_DRAWMODE_NORMAL);
GUI_FillCircle(120, 120, 80);
GUI_SetDrawMode(GUI_DRAWMODE_XOR);
GUI_FillCircle(140, 140, 80);
```

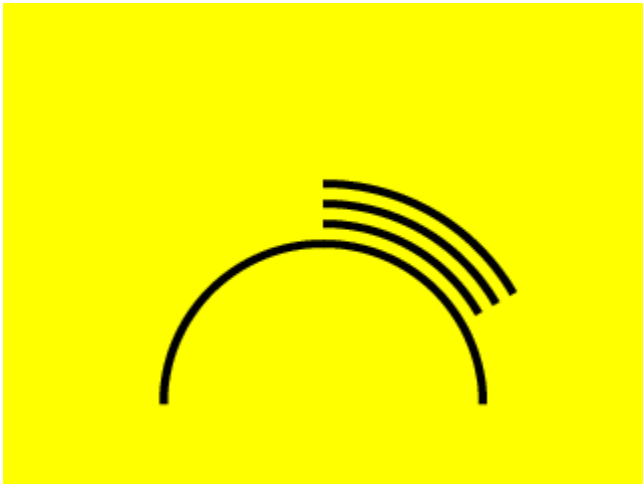


Basic 2D library - Antialiasing

- Antialiasing smoothens curves and diagonal lines by "blending" the background color with that of the foreground.
  - Text
    - Font converter is required for creating AA fonts.
  - Circles
  - Arcs
  - Lines
  - Polygons

```
GUI_AA_SetFactor(4);
GUI_AA_DrawArc(160, 200, 80, 0, 0, 180);
GUI_AA_DrawArc(160, 200, 90, 0, 30, 90);
GUI_AA_DrawArc(160, 200, 100, 0, 30, 90);
GUI_AA_DrawArc(160, 200, 110, 0, 30, 90);
```

The higher the number of shades used between background and foreground colors, the better the antialiasing result (and the longer the computation time).

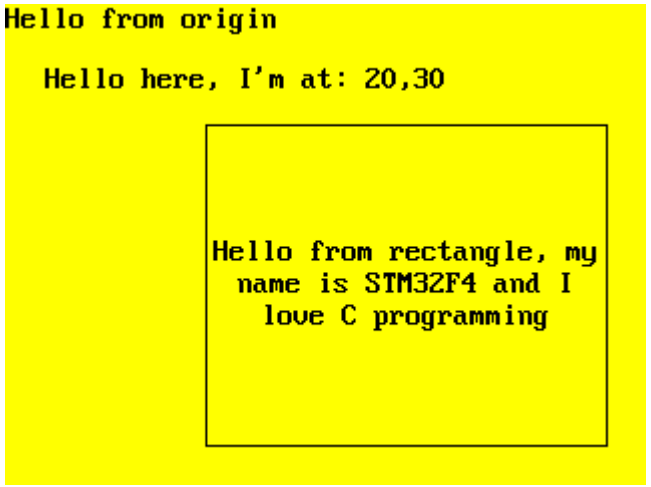


life.augmented

Basic 2D library – Text

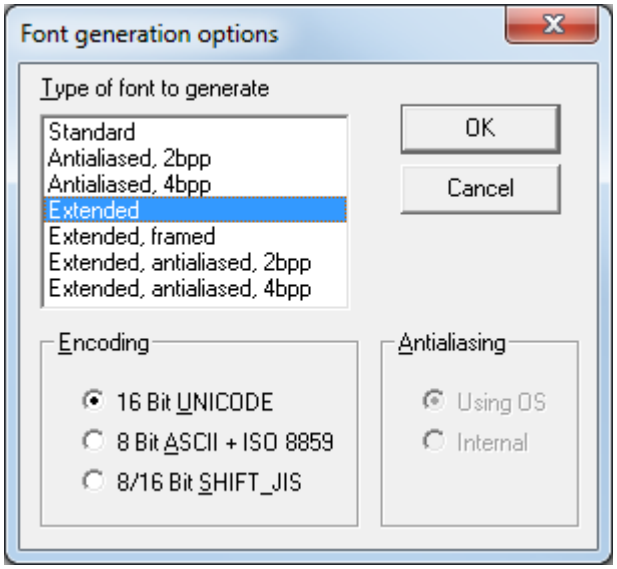
- STemWin enables you to add text of any font into your GUI
- Several API functions are available to ease text use
  - Display text at specific position
  - Manage text inside a rectangle

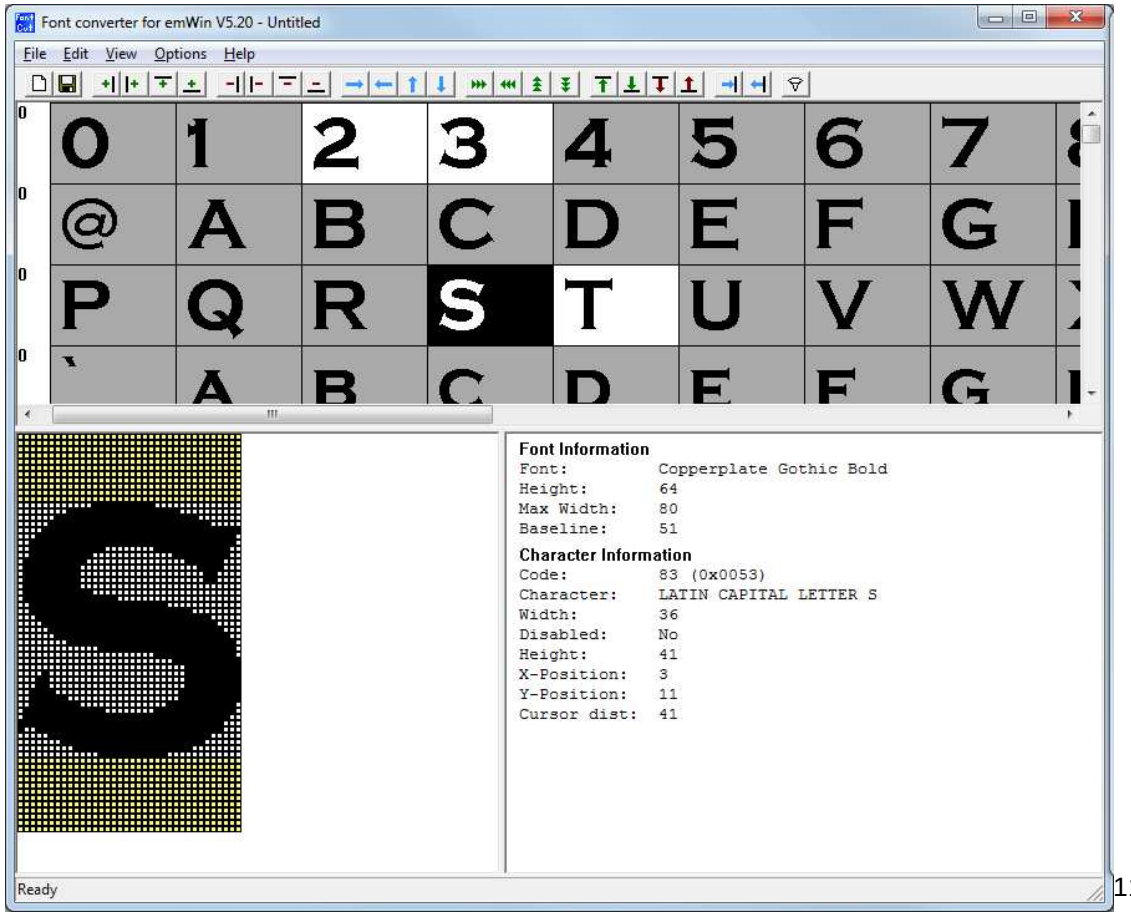
```
GUI_DispString("Hello from origin");
GUI_DispStringAt("Hello here, I'm at:
20,30", 20,30);
{
GUI_RECT pRect = {20, 60, 220, 160 };
GUI_DrawRect(20, 60, 220, 160);
GUI_DispStringInRectWrap("Hello from
rectangle, my name is STM32F4 and I love
C programming", &pRect, GUI_TA_VCENTER |
GUI_TA_HCENTER, GUI_WRAPMODE_WORD);
}
```



Basic 2D library – Text

- Manage your fonts
  - Create fonts from any windows font
  - Mange the number of characters so that you save only those you need → safe ROM
  - Export as .c files





Basic 2D library – Text

- Using the font we have generated is very easy
  - Include the generated .c file into the project
  - Include the extern font declaration to all modules which will use it
  - Use GUI_SetFont() function to point STemWin to this font

```
extern GUI_CONST_STORAGE GUI_FONT GUI_FontCopperplateGothicBold64;
extern GUI_CONST_STORAGE GUI_FONT GUI_FontCopperplateGothicBold64_aa;
..
GUI_RECT pRect1 = {0, 60, 319, 119};
GUI_RECT pRect2 = {0, 120, 319, 180};
GUI_SetFont(&GUI_FontCopperplateGothicBold64);
GUI_DispStringInRect("STM32", &pRect1, GUI_TA_VCENTER | GUI_TA_HCENTER);
GUI_SetFont(&GUI_FontCopperplateGothicBold64_aa);
GUI_DispStringInRect("STM32", &pRect2, GUI_TA_VCENTER | GUI_TA_HCENTER);
```

Drawing anti-aliased text takes much more time! (and memory as well)

STM32

STM32

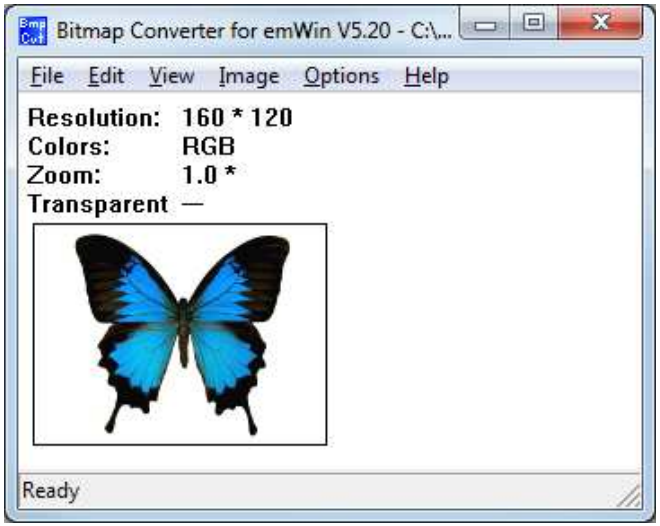
life.augmented

CONTENTS

- Objectives of the STemWin solution
- Basic 2D library
- PC SW Tools
  - FontCvt
  - BmpCvt
  - GUIBuilder
- Window Manager
- Other Features

Basic 2D library bitmaps

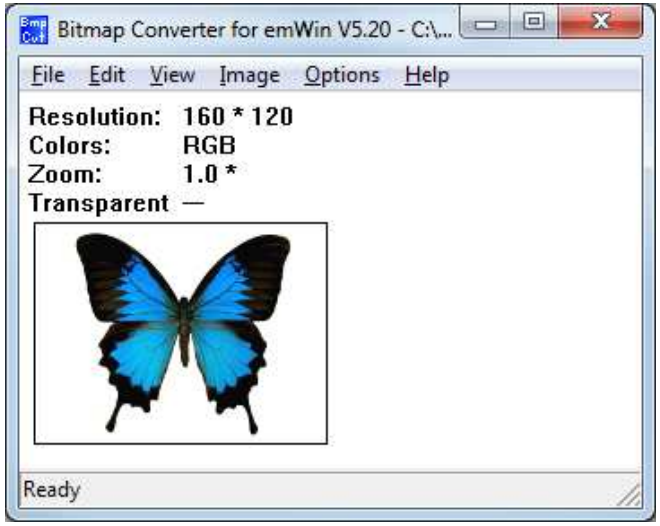
- Supported formats
  - BMP
  - JPEG
  - PNG
  - GIF

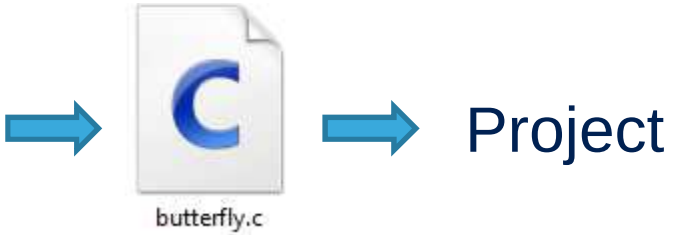


- BmpCvt.exe can be used for bitmap conversion and storage to .c files
- Bin2c.exe can be used to store any binary in .c form, e.g. complete bitmap files

life.augmented

Basic 2D library – bitmaps example



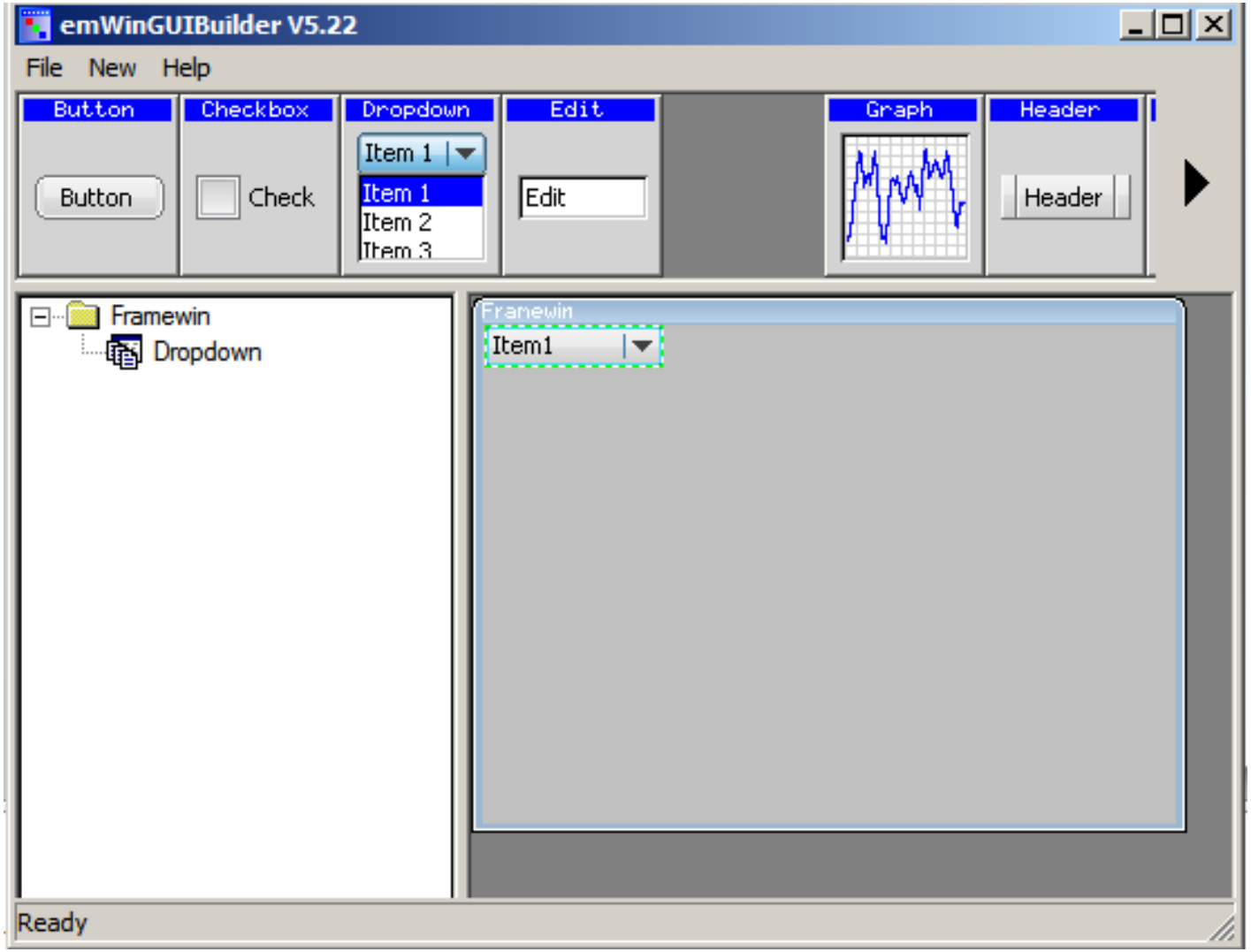



```
extern GUI_CONST_STORAGE GUI_BITMAP bmbutterfly;
GUI_DrawBitmap(&bmbutterfly, 30, 30);
```

Storing bitmap in the same format as the LCD increases drawing performance → no pixel conversion needed. But may need more storage space.

life.augmented

GUI Builder


emWinGUIBuilder V5.22
File
New
Help
Button
Checkbox
Dropdown
Edit
Graph
Header
Button
Check
Item 1
Item 2
Item 3
Edit
Header
Framewin
Dropdown
Framewin
Item1
Ready

CONTENTS

- Objectives of the STemWin solution
- Basic 2D library
- PC SW Tools
  - FontCvt
  - BmpCvt
  - GUIBuilder
- Window Manager
- Other Features

Windows Manager

What is the **W**indow **M**anager?

- **Management system for a hierarchic window structure**

  Each layer has its own desktop window. Each desktop window can have its own hierarchic tree of child windows.

- **Callback mechanism based system**

  Communication is based on an event driven callback mechanism. **All** drawing operations should be done within the WM_PAINT event.

- **Foundation of widget library**

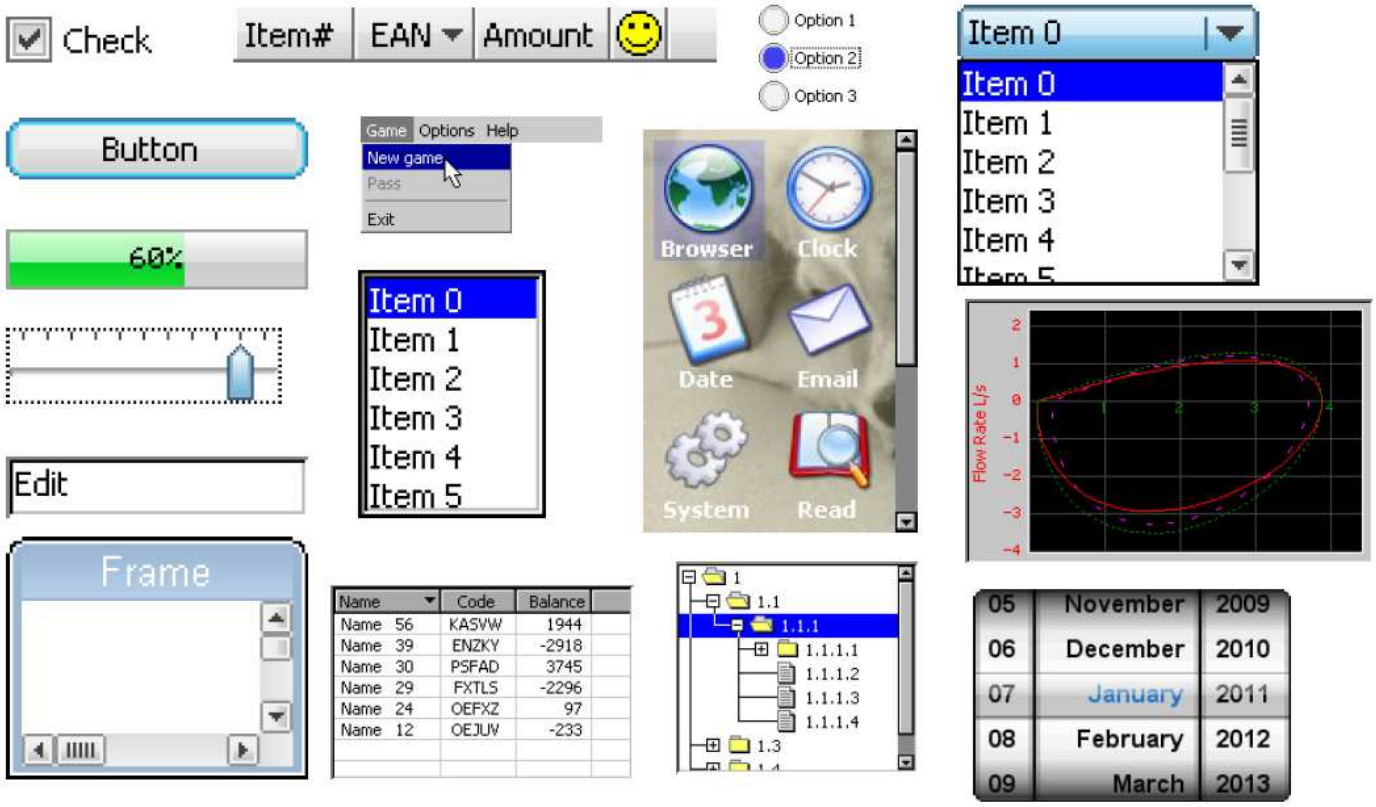
  All widgets are based on the functions of the WM.

WM – Widget library

Widget = **Wi**ndow + Ga**dget**

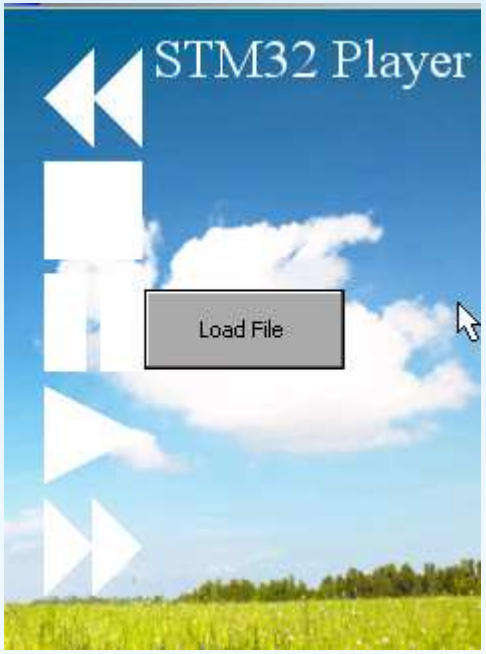
Currently the following widgets are supported:

- Button, Checkbox, Dropdown, Edit, Framewin, Graph, Header, Iconview, Image, Listbox, Listview, Listwheel, Menu, Multiedit, Progbar, Radio, Scrollbar, Slider, Text, Treeview



Add Widget

- After click on button show a
- LISTVIEW widget with selection of songs




```
#include "BUTTON.h"

#define BUTTON_WIDTH            100
#define BUTTON_HEIGHT           40

static BUTTON_Handle hButton;

hButton = BUTTON_Create( 240/2 - BUTTON_WIDTH / 2, 320/2 - BUTTON_HEIGHT / 2,
BUTTON_WIDTH, BUTTON_HEIGHT, GUI_ID_BUTTON1, WM_CF_SHOW);

BUTTON_SetText(hButton, "Load File");
```

Add Widget

- Add ListView widget

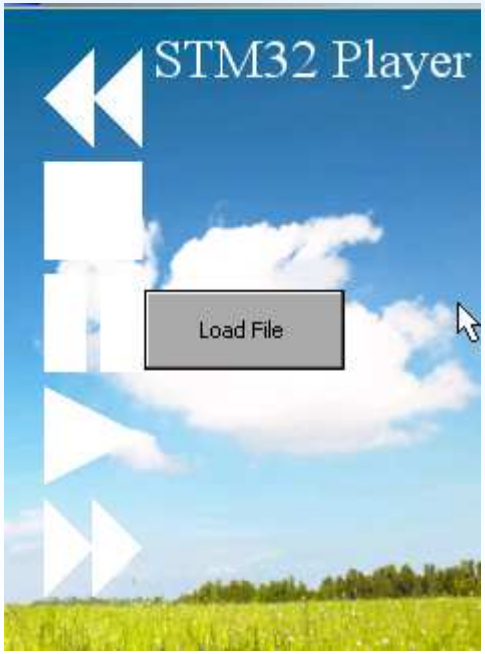

```
#include "LISTVIEW.h"

#define LISTVIEW_WIDTH 220
#define LISTVIEW_HEIGHT 300

static LISTVIEW_Handle hListView;
static const char * myFileTable[][1] =
{"Miley Cyrus","Katy Perry",
"Robin Thicke","Lady Gaga"};

int i;

hListView = LISTVIEW_Create( 240/2 - LISTVIEW_WIDTH / 2, 320/2 - LISTVIEW_HEIGHT / 2,
LISTVIEW_WIDTH, LISTVIEW_HEIGHT, WM_HBKWIN, GUI_ID_LISTVIEW0, 0, 0);

// fill in the data
LISTVIEW_AddColumn(hListView, 100, "File Name", GUI_TA_CENTER);
for (i = 0; i < GUI_COUNTOF(myFileTable); i++)
 LISTVIEW_AddRow(hListView, myFileTable[i]);
```

In this loop just copy text array into the widget

Add Widget

- Set call back function by WM_SetCallback
  - WM_SetCallback(WM_HBKWIN, myCbWin);
  - In WM_PAINT, we only clear screen with color blue for it's background callback

```
static void myCbWin(WM_MESSAGE * pMsg) {
  int NCode, Id;
  switch (pMsg->MsgId) {
  case WM_NOTIFY_PARENT:
    Id    = WM_GetId(pMsg->hWinSrc);      /* Id of widget */
    NCode = pMsg->Data.v;                 /* Notification code */
    switch (Id) {
    case GUI_ID_BUTTON1:
      switch (NCode) {
      case WM_NOTIFICATION_RELEASED:
        WM_ShowWindow(hListView);
        break;
      }
      break;
    }
    break;
  case WM_PAINT:
    GUI_SetBkColor(GUI_BLUE);
    GUI_Clear();
    break;
  }
}
```



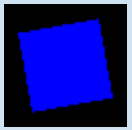
life.augmented

CONTENTS

- Objectives of the STemWin solution
- Basic 2D library
- PC SW Tools
  - FontCvt
  - BmpCvt
  - GUIBuilder
- Window Manager
- Other Features

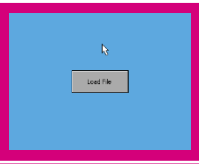
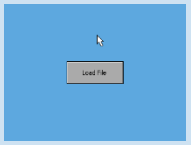
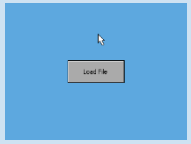
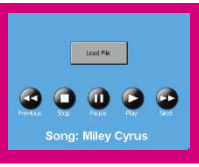
Other STemWin features

- Memory devices
  - Memory Devices can be used in a variety of situations, mainly to prevent the display from flickering when using drawing operations for overlapping items.
  - This requires additional RAM memory

API function	Without Memory Device	With Memory Device
Step 0: Initial state		
Step 1: GUI_Clear()		
Step 2: GUI_DrawPolygon()		
Step 3: GUI_DispString()		
Step 4: GUI_MEMDEV_CopyToLCD()		

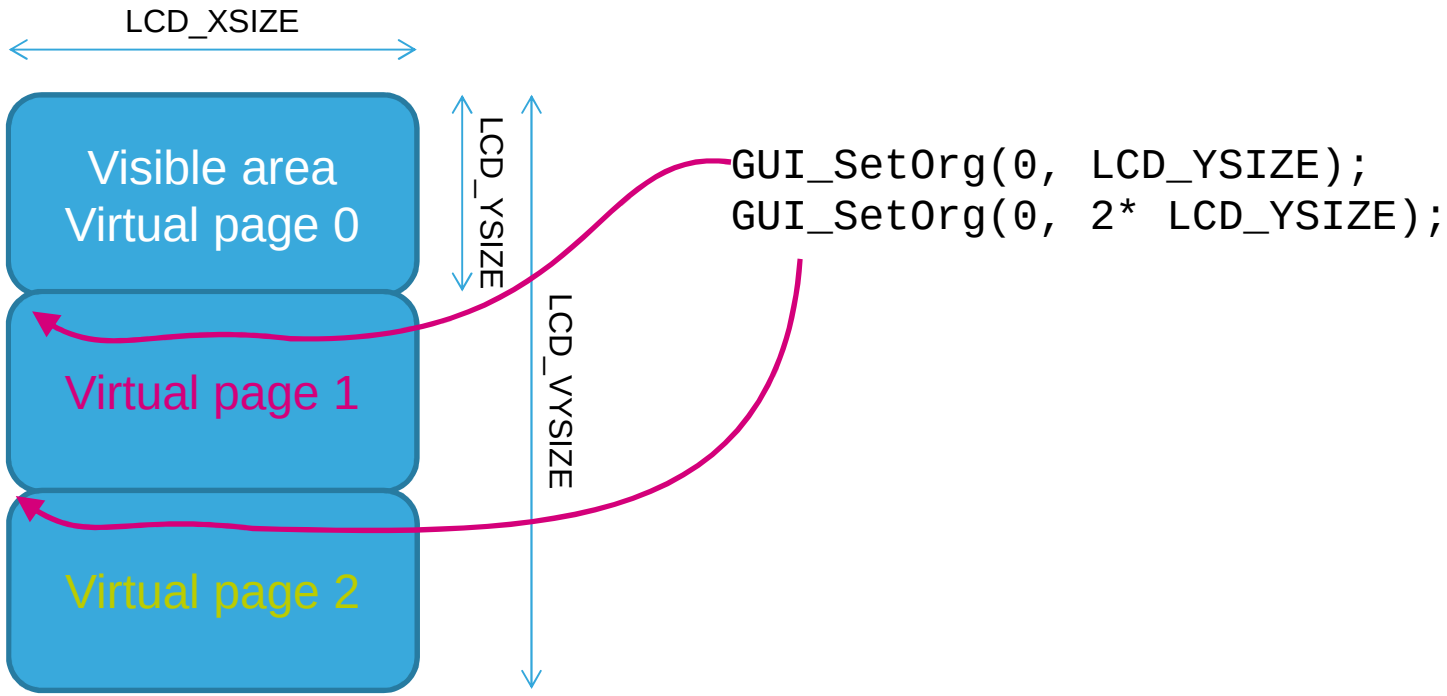
Other STemWin features

- Multiple buffering
  - With multiple buffers enabled there is a front buffer which is used by the display controller to generate the picture on the screen and one or more back buffers which are used for the drawing operations.
- In general it is a method which is able to avoid several unwanted effects:
  - The visible process of drawing a screen item by item
  - Flickering effects caused by overlapping drawing operations
  - Tearing effects caused by writing operations outside the vertical blanking period

API function	1st buffer	2nd buffer
1. Copy active buffer to inactive		
1. Drawing operations in inactive buffer		
3. Set 2nd buffer as active		

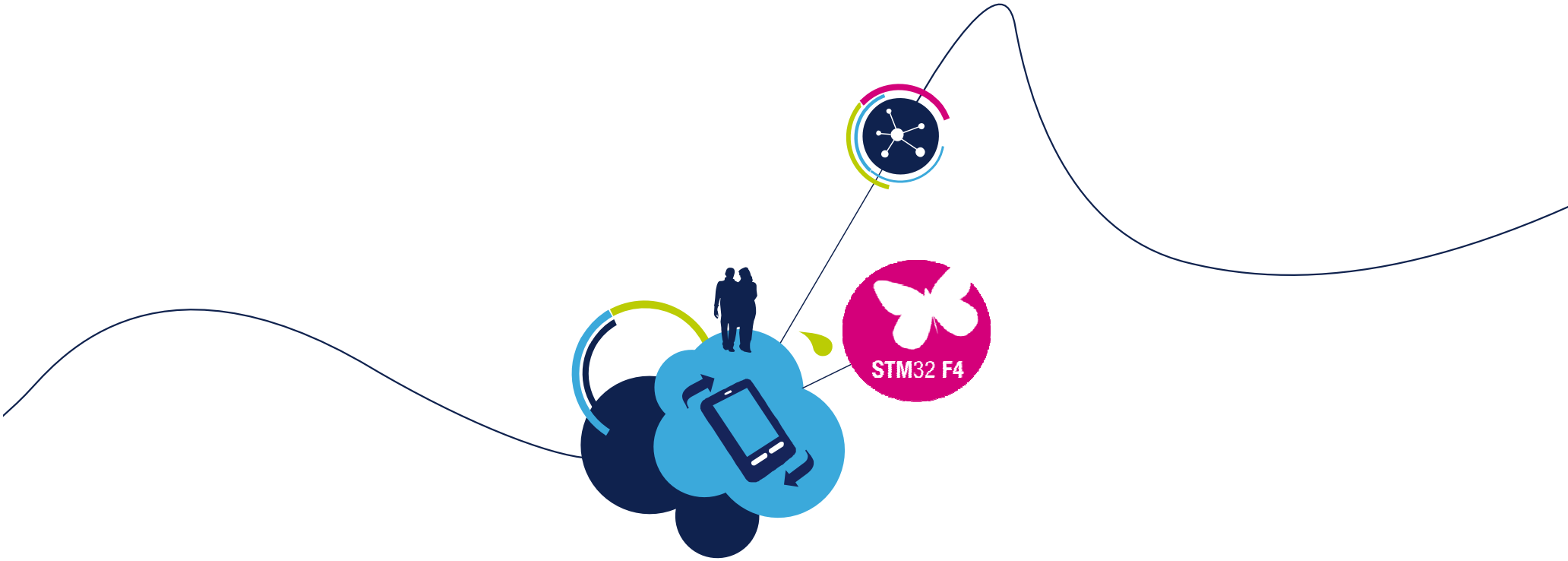
Other STemWin features

- Virtual screens
- Display area greater than the physical size of the display. It requires additional video memory and allows instantaneous switching between different screens.
- Similar to Multiple buffering, but no copy operation is performed.



Other STemWin features

- Video Play
  - Convert each movie frame to single JPEG file, and contains them with a EMF file
  - Compile EMF file in c array format or open it as file



Q&A

THANKS

CONFIDENTIALITY OBLIGATIONS:

THIS DOCUMENT CONTAINS SENSITIVE INFORMATION.

IT IS CLASSIFIED “MICROCONTROLLERS, MEMORIES & SECURE MCUs (MMS) RESTRICTED AND ITS DISTRIBUTION IS SUBMITTED TO ST/MMS AUTHORIZATION

AT ALL TIMES YOU SHOULD COMPLY WITH THE FOLLOWING SECURITY RULES:

- DO NOT COPY OR REPRODUCE ALL OR PART OF THIS DOCUMENT
- KEEP THIS DOCUMENT LOCKED AWAY
- FURTHER COPIES CAN BE PROVIDED ON A “NEED TO KNOW BASIS”, PLEASE CONTACT YOUR LOCAL ST SALES OFFICE OR DOCUMENT WRITTER.