Good Display' co.,



E-ink 电子纸开发板使用说明手册

Edition 1.1.0

大连佳显电子有限公司

大连市沙河口区工华街 17 号

T: 86 411 84619565 DD: +86 411 84526151 E: sales@good-display.com

www.good-display.cn

摘要

大连佳显电子有限公司 2013 年推出的 2.1 寸、3.5 寸、 4.3 寸、6 寸和 8 寸的 E-ink 电子纸开发板是让用户简单上手 容易使用电子纸为目的而研制的,对接触过二代计算器语言 和 C 语言环境开发的工程师,可以很快了解并应用,此外针 对 C 语言的系统函数它可以非常灵活地在该产品上进行控制 和操作。Kiel 软件是在很多系统底层编程时需要用到的流行 软件,所以我们建议使用该软件进行编译,整个编译和下载 过程借助 Debugger 和 Kiel3 的结合完成。同时大连佳显电子 有限公司也提供相关的程序例程供客户参考,用户可以向与 你联系的销售工程师索取。

针对不同尺寸的电子纸,我们有不同的 Demo kit 可以提供,使用 Arm 单片机存储可以容纳多张图片交替刷新,我们以 2.1 寸的 Demo 开发板为例来说明此类 Demo 开发板的使用方法,其它的 Demo 开发板的使用都与此类似。

目录:

产品摘要	•••••	02
结构部分	•••••	04
设置程序	•••••	07
编译过程	•••••	12
下载	••••••	13
调试程序	••••••	14
生成一个图片	••••••	14

结构部分:

如果仅仅是想显示内容,只要将 Demo 开发板和电子纸在未上电情况下连接,之后将 Demo 开发板 的 USB 供电插入 PC 的 USB 总线进行供电就可使用了。



每一个 Demo kit 上留有一枚工作电源指示灯,在上电工作情况下 它会保持绿灯常亮,断电后瞬间熄灭,而且在供电源区域内可见 一个跳线,用户可以选择使用电源适配器或 USB 供电。



Demo kit 上的 USB 连接线不参与编译后下载,只能供电。下载接口部分是采用标准 JTAG 接口,此处需要与 Debugger 的 JTAG 连接,再用 Debugger 的另一端用 USB 连接计算器的 COM 口。

JTAG 接的连接方向如图所示,将凹槽对像连接线的凸边:



就下载和调试阶段而言,您需要具备一台安装了 Keils3 及以上版本(已注册)的计算机,除此之外还需要连接 Debugger 和 Demo 开发板,最后需要 Demo kit 和 E-ink 电子纸的正确连接。(如图)



设置程序:

打开我们的 Kiel 软件后

打开工程>光标右击左侧工程>选择 Options for Target >进入
 Debug 选项卡,找到 JTAG debugger,选择 settings。

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject	Pl <u>a</u> sh <u>D</u> ebug Peripherals <u>T</u> ools <u>S</u> VCS <u>1</u>	Window <u>H</u> elp
	1 つで (+ +) 作 時 時 時 () 律	i 11: 11: 11: Ma
۵ 🕮 🧼 🗮 🛍 🕲	📥 Right click 🔊 🚠 🗟	
roject	🗴 🏹 misc. c 🏦 stm32f10x_iwd	<mark>g. c 🏾 💭 core_cm3. c</mark>
E C FwLib	Options for Target 'mdk' Alt+F7	2); C4);
	Open File Click	1
± stm32f10x_flash.	Open List File	
	Open .\Debug\list\mdk.Map	
🛨 💽 stm32f10x_tim.c	Rebuild all target files	1
+ 🔆 stm32f10v judg.	Build target F7	1
E 🔄 StartUp	Tr <u>a</u> nslate File	1
🛨 📩 system_stm32f10x	Stop build	
t is the state of	Add Grown	10);//exit deep
startup_stm32f10) Idd Pilas to Group	1);//data enter
🖻 🔄 User 🕀 🔝 stm32f10x_it.c	Remoge Item	:03); :4);//set RAM x
🛨 📩 main. c	Anage Components	00);//RAM x add A(Ox1f);
	Show Include File Dependencies	:11);//RAM x add :5);//set RAM v
	2227 SPI4W_WRITEDATA() 2228 // SPI4W_WRITEDA	DxOO);//RAM y add ATA(Oxb3);

	👯 🛛 🕅 Options for Target 'mdk'	
2206 2207 _gpio. c 2208) _rcc. c 2209 v _flash. c 2210 (_spi. c 2212 _tin. c 2213 _iwdg. c 2213 _iwdg. c 2214 _iwdg. c 2215 c 2216 #1 c 2218 _conf. h 2219 tu32f10x_hd. 2220 2221	SP SP SP Device Target Output Listing User C/C++ oid I C Use Simulator Settings C Limit Speed to Real-Time C Load Application at Startup Ru ton in() Initialization File: C Restore Debug Session Settings C Breakpoints T Toolbox SP SP SP SP SP SP SP	Asm Linker Debug Utilities 2 Image: Settings ULINK Cortex Debugger Settings ULINK Cortex Debugger FDI Interface Driver Settings Image: Signum Systems JTAGjet Debugger Debugger Image: Signum Systems JTAGjet Edk ULINK Procord Cortex Debugger Image: Strain Cortex Debugger ULINK Procord Image: Strain Cortex Debugger Edk Image: Strain Cortex Debugger Image: Strain Cortex Debugger Image: Strain Cortex Debugger Image: Strain Cortex Debugger
_it.e 2222 2223 2224 2225 2226 2227 2228 2229	SP CPU DLL: Parameter: // [SARMCM3.DLL]	Driver DLL: Parameter: SARMCM3.DLL Dialog DLL: Parameter: TARMSTM.DLL -pSTM32F103RB

N.		e Chain		Move
	TD0			Up
W:dl				Tr-un
w: 🔽 J-L	ink V4.74b Firmwar	e update		
)ebug	New features / improve	ements may not be avail	able without	a firmware updat
)ebug Connect & R Treset Reset after Connec	New features / improve	ements may not be avail 是(Y) 否(Y) 了 UY Cache <u>Coue</u> V Cache <u>M</u> emod	able without	a firmware updat code powwnoad pad to <u>F</u> lash
)ebug Connect & R <u>Heset</u> after Connect nterface	New features / improve	ements may not be avail 差(1) 子 (1) 子 Cache <u>Coue</u> 子 Cache <u>M</u> emor	able without i	a firmware updat coue Download pad to <u>F</u> lash Misc
Debug Connect & R Reset after Connect nterface	TCP/IP Network Settings	ements may not be avail 是①	able without i	a firmware updat coue Download pad to <u>F</u> lash Misc JLink Info
Debug Connect & R I Beset after Connect I SB ← TCP/IP Reconnect	New features / improv	ements may not be avail 是(1) 下 Cache <u>M</u> emor Port (Auto: 0) :	able without y	a firmware updat



上传完成后需要对内容进行确定保存。

2. 然后在 Utilities 选项卡中同样选择 JTAG 进行类似设定,进入

settings 选项。(此步骤同上)

gure Flash Menu Command	
Ise Target Driver for Flash	
Cortex-M/R J-LI	INK/J-Trace Settings IV Update Target before Debugging
	Cortex JLink/JTrace Target Driver Setup
Ise External Tool for Flas	Debug Trace Flash Download
ommand:	Download Function RAM for Algorithm
guments:	LOAD C Erase Full Chip V Program
F Run Indepr	C Do not Erase C Reset and Run
	Programming Algorithm
	Description Device Type Device Size Address Range
	STM32F10x Med-density Flash On-chip Flash 128k 08000000H - 0801FFFFH
1	Start: 0x08000000 Size: 0x00020000
	Add Bemove

注意:完成后请检查 Debug 和 Utilities 的 FlashDownload 选项 卡如果没有正确添加芯片请手动添加 STM32F10x 128K。

Device 选项卡中找出我们所用的单片机规格,完毕后将 User 选项卡中 Run user pro 去掉勾选并在 Output 选项卡中勾选上产 生 Hex 十六进制文件。

SP) void I (#if 0 // SP er	Device Target Output Li Database: Generic O Vendor: STMicroelectronics Device: STM32F103RB Toolset: ARM STM32F102R4 STM32F102R4 STM32F102R8	sting V PU Data B	ARM 320th Cottex-M3 Microcontroller, 72MHz, 128kB Flash, 2kB SRAM, ARM 320th Cottex-M3 Microcontroller, 72MHz, 128kB Flash, 2kB SRAM, ARM Stand 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD, Arm	
<pre> > void I (#if 0</pre>	Database: Generic C Vendor: STMicroelectronics Device: STM32F103RB Toolset: ARM STM32F102R4 STM32F102R4 STM32F102R6 STM32F102R8	PU Data B	ARM 32 bit Cottex-M3 Microcontroller, 72MHz, 128kB Flash, 21kB SRAM, APLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD,	• Terraria
void I (#if 0 // SP er	Database: Generic C Vendor: STMicroelectronics Device: STM32F103RB Toolset: ARM STM32F102CB STM32F102CB STM32F102R4 STM32F102R6 STM32F102R8	PU Data B	ARM 32 bit Cortex-M3 Microcontroller, 72MHz, 128kB Flash, 2kB SRAM, APLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD,	• •
(#if 0 // SP	Vendor: STMicroelectronics Device: STM32F103RB Toolset: ARM STM32F102CB STM32F102CB STM32F102R4 STM32F102R6 STM32F102R8	~	ARM 32 bit Cortex-M3 Microcontroller, 72MHz, 128kB Flash, 21kB SRAM, APLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD,	
#1f 0 // SP	Vendor: STMicroelectronics Device: STM32F103RB Toolset: ARM STM32F102CB STM32F102CB STM32F102R4 STM32F102R6 STM32F102R8	^	ARM 32 bit Cortex-M3 Microcontroller, 72MHz, 128kB Flash, 21kB SRAM, APLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD,	• •
#if 0 // SP	Device: STM32F103RB Toolset: ARM STM32F102CB STM32F102R4 STM32F102R6 STM32F102R6 STM32F102R8	~	ARM 32 bit Cortex-M3 Microcontroller, 72MHz, 128kB Flash, 24kB SRAM, APLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD,	
#1f 0 // SP	Toolset: ABM	^	ARM 32 bit Cortex-M3 Microcontroller, 72MHz, 128kB Flash, 24kB SRAM, APLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD,	
#if 0 // SP	- STM32F102CB - STM32F102CB - STM32F102R4 - STM32F102R6 - STM32F102R8	^	ARM 32 of Cortex-M3 Microcontroller, 72MHz, 128kB Flash, 24kB SRAM, APLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SWD,	
#if 0 // SP	- STM32F102CB - STM32F102R4 - STM32F102R6 - STM32F102R6 - STM32F102R8	~	ARM 32 of Cortex-M3 Microcontroller, 72MHz, 128kB Flash, 2 kB SRAM, PLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and SW 9,	
#if 0 // SP	- CI STM32F102R4 - CI STM32F102R6 - CI STM32F102R8		PLL, Enbedded Internal RC 8MHz and 32kHz, Real-Time Clock Nest d Interrupt Controller, Power Saving Modes, JTAG and Sw.D,	
SP	CI STM32F102R6		Nested Interrupt Controller, Power Saving Modes, JTAG and SWU,	
SP	STM32F102R8		A Stoch The bit Timers with Input Laplure Thutput Lompare and WM	
CD			16 pit 6-ch Advanced Timer, 2 16-bit Watchdog Timers, SysTick Timer,	
SP	STM32F102RB		2 PI, 212C, 3 USART, USB 2.0 Full Speed Interface, CAN 2.0 Active,	
SP	STM32F103C4		2 2-bit 16-ch A/D Converter, Fast I/D Ports	
SP	STM32F103C6			
SP	STM32F103C8			
SP	STM32F103CB			
CD.	STM32F103R4			4 12/101 -> 761
SP	STM32F103R6			12 (10) -> /0
JF	STM32F103H8			
11		~	· · · · · · · · · · · · · · · · · · ·	
SP	<	>	3	801
SP				and the second sec
SP -				
	SP SP SP // SP SP SP SP	SP SP SP SP SP SP SP SP SP SP SP SP SP S	SP STM32F103C4 SP STM32F103C6 SP STM32F103C8 SP STM32F103C8	SP SP SP SP SP SP SP SP SP SP SP SP SP S

2206	misc.c	🛚 Options for Target 'mdk'	<u> </u>
2200	SP	Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
2208	>	and the second	
2209	void I	Run User Programs Before Compilation of a C/C++ File	
2210	(□ Run #1: □ □ DOS16	
2211			
2212		1 Hun #2.	
2214			
2215		Run User Programs Before Build/Rebuild	
2216	#if O		
2217	11		
2218	SP	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
2219	SP		
2220	SP		
2221	SP	Run User Programs After Build/Rebuild	
2222	SP	「 lun #1: 「 DOS16	
2223	SP		
2224	//	1 Run #2:1 DUS16	
2225	SP	Reen When Complete Start Debugging	ia 12(18)
2226	SP		
2221	SP //		
2220	CD CD		901
2223	SP		001
2230	SP		
4	1.1		

SP 1	Device Target Output Listing Vser C/C++ Asm Linker Debug Utilities
roid I (Select Folder for Objects Name of Executable: mdk
	Create Executable: \Debug\obj\mdk
	✓ Debug Information Create Batch File
-	Create HEX File
if 0	
// @D	Browse Information
SP	C Create Library: .\Debug\obj\mdk.LIB
SP	
SP	
SP	
SP	
11	
SP	
SP	
1/	
SP	
SP	
SD _	

编译过程:

编译你的程序可以在左上角点击 build 按键,正式编译后会看到窗口下方的 Build Output 窗口内显示成功编译字样,如果存在错误,也会提示错误所在的命令行及函数体,且 Hex 文件不会被成功创建。



<u>下载</u>:

请再一次确认 Debugger 和 Demo kit 正确连接且都已上电,工作 指示灯闪烁或常亮。点击 Start/Stop Debugger session 按钮后自 动将已编译好的文件下载到 Arm 单片机中去。



下载只可能在编译后执行。

调试程序:

可以使用双击命令行前端设置一个节点,用 step 功能检查工作情况,建议每次调试后都编译后下载再使用,这是为了保证单片机中的程序和工程同步。

生成一个图片:

借助任意一款网络上的图片转码软件都可以,这里建议使用我们 的小软件 BitmapCovert,只要将.bmp 格式的绘图加载到软件中进 行格式化处理,再将图片转存.c 文件,记住勾选 without template 选项可以确保产生的.c 文件内没有重复的混淆使用的数据。将图 片部分产生的十六进制数组全部拷贝到工程内就可以正确的替换 一张内置图片,编译下载完成。

File Edit View Image Help Resolution: 99 Iransparency Colors: 4 Convert Into Zoom: 2.0 Flip Transparent - Gray16 Eile E Invert Info Gray256 S bit Color 111 (4 BFP) 6 bit Color 222 (6 BFP) 8 bit Color 233 (6 BFP) 8 bit Color 332 (6 BPP) 8 bit Color 3666 (6 BPP)	
Resolution: 99 Iransparency Colors: 4 Zoom: 2.0 Transparent Eile BW (1 BFP) BW (1 BFP) Gray4 (2 BFP) Gray16 (4 BFP) Gray256 (8 BFP) Gray256 (8 BFP) Stit Color 111 (4 BFP) 6 bit Color 222 (8 BFP) 8 bit Color 222 (8 BFP) 8 bit Color 233 (8 BFP) 8 bit Color 233 (8 BFP) 8 bit Color 332 (8 BFP) 8 bit Color 3666 (8 BFP)	
Colors: 4 Convert Into BW (1 BFP) Zoom: 2.0 File Bit Gray4 (2 BFP) Fransparent Rotate Gray16 (4 BFP) Gray16 (4 BFP) File Invert Gray256 (3 BFP) Gray256 (3 BFP) Stit Color 111 (4 BFP) Gray256 (3 BFP) Debug Peripherals Tools SVCS Window Bit Color 222 (8 BFP) 8 bit Color 233 (8 BFP) 8 bit Color 332 (8 BFP) 8 bit Color 3666 (8 BFP) 8 bit Color 3666 8 BFP) 8 bit Color 3666 8 BFP) 8 bit Color 3666 8 BFP) 8 bit Color 3666 8 BFP)	
Eile Invert Gray64 (6 BFF) Info Gray256 (3 BFF) Image: State of the state of t	
Image: Solution of the	07
Image: Solution of the	Ξ
P/P (24 PPP)	
Project Best palette Tisc. c Stm32f10z_iwdg. c	100
Imalk Custom palette Imalk Imalk Imalk Image: Company of the standard standa); 4)

	Format specification 🛛 🔀	ph
	Select format:	
	2 bit per pixel 4 bit per pixel 8 bit per pixel Compressed, RLE4 Compressed, RLE8 High color (565) High color (565), red and blue swapped	1
1	₩ithout palette	. c
1	OK Cancel	
	OK Cancel	-

这里的选项请注意是 per pixel「每像素比特数」,所以请 了解产品可支持的灰阶等级后进行转换。 文件(F) 编辑(E) 格式(0) 查看(V) 帮助(H)

```
/*
                Palette
The following are the entries of the palette table.
Every entry is a 32-bit value (of which 24 bits are actually used)
the lower 8 bits represent the Red component,
the middle 8 bits represent the Green component,
the highest 8 bits (of the 24 bits used) represent the Blue component
                                               ØxBBGGRR
as follows:
*/
static GUI_CONST_STORAGE GUI_COLOR Colors新建BMP图像[] = {
                  0x000000,0x555555,0xAAAAAA,0xFFFFFF
};
#endif
static GUI_CONST_STORAGE GUI_LOGPALETTE Pal新建BMP图像 = {
      4,
                            /* number of entries */
                            /* No transparency */
       0,
      NULL
};
static GUI_CONST_STORAGE unsigned char ac新建BMP图像[]
                             0xFF, 0x
                             0xFF, 0x
                            0xFF, 0xFF,
```

0xF0.

此段数组就是转换成的代码,需要将程序部份替换。

<u>File Edit View Project Fla</u> sh	Debug	Pe <u>r</u> ipherals	<u>T</u> ools S	WCS Window	r <u>H</u> el	p					
0 🗃 🖬 🕼 🖌 🖻 🛍 🔊	(n. da		图图	*F *F //	≡ <i>II</i> ¥	2		× 🔊	🥐 🔍 🖣	000	- 3
😂 🔛 🖾 🧼 🚉 🕌 mdk		× 🕅	A 5			FL	Inctio	n name	The co	net char	table
roject 🥑 🗙		misc.c 💽	stm32f1	Ox_iwdg.c		core_cm3.c	📩 🚥	in.c	The co	inst chai	lable
🛨 🔛 system_stm32f10x. c 📩	1922	OXFF,OXFF	,OXCF,O	XFF,OXC3	,OXFF	,OXCF,OKF	F,OXFF,	OXFF, OXFF	,OXFF,OXF1	,OXFF,OXFF,	OXFF,
🛨 🚆 core_cm3. c	1923	OXFF, OXFF	,OXFF,O	XFF, OXFF	,OXFF	,OXFF, XF	F, OXFF,	OXFF, OXFF	,OXFF,OXF	, OXFF, OXFF,	OXFF,
stm32f10x_conf. h	1924	OXFF, OXFF	,OXFF,O	XFF, OXFF	,OXFF	,OXFF OXF	F, OXFF,	OXFF, OXFF	,OXFF,OXIF	,OXFF,OXFF,	OXFF,
🚬 💒 startup_stm32f10x_	1925	OXFF, OXFF	,OXFF,O	XFF, OXFF	, OXFF	,OXT,OXF	F, OXFF,	OXFF, OXFF	,OXFF,OX F	,OXFF,OXFF,	OXFF,
🖃 🍓 User	1926	OXFF, OXFF	,OXFF,O	XFF, OXFF	,OXFF	,OXL,OXF	F,};				
± stm32f10x_it.c	1927	const uns	igned c	<mark>har gIma</mark>	ge SU	NING[3096] = { /	* 0X81,0X	02, OXAC OX	00,0X48,0X0	0, */
E i main. c	1928	0x00,0x00	,oxoo,o	x00,0x00	, OXFF	,OXFF,OXF	F, OXFF,	OXFF, OXFF	,OXFF,CXFF	, OXFF, OXFF,	OXFI,
stm32f10x.h	1929	DXFF, OXFF	,0x00,0	x00,0x00	, 0X00	,OXOO,OXF	F, OXFF,	oxoo,oxoo	, oxoo xoo	,0X00,0X00,	oxoc,
core_cm3.h	1930	DX00,0X00	,OXOF,O	XFF, OXOO	, oxoo	,0X00,0X0	0,0X00,	OXFF, OXFF	,OXFF	, OXFF, OXFF,	OXF1,
stdint. h	1931	DXFF, OXFF	,OXFF,O	XFF, OXFF	,OXFF	,0X00,0X0	0,0X00,	oxoo, oxoo	,OXFF,OXFF	,0X00,0X00,	oxoc,
system_stm32f10	1932	DX00,0X00	,0X00,0	x00,0x00	, oxoo	,OXOF,OXF	F, OXOO,	0X00,0X00	,0x00,0x00	, OXFF, OXFF,	OXF1,
stm32f10x_conf.	1933	DXFF, OXFF	,OXFF,O	XFF, OXFF	,OXFF	,OXFF,OXF	F, OXFF,	OXFF, OXOO	,0x00,0x00	,0x00,0x00,	OXF1,
stm32f10x_flash	1934	DXFF, OXFF	,OXFF,O	XFF, OXFF	, OXFF	,OXFF,OXF	F, OXFF,	OXFF, OXFF	,OXFF,OXOO	,0X00,0X00,	oxoc,
stm32f10x_gpio.	1935	DX00, OXFF	,OXFF,O	x00,0x00	, oxoo	,oxoo,oxo	0,0X00,	oxoo, oxoo	,OX3F,OXFF	, OXFF, OXOO,	oxoc,
stm32f10x_iwdg	1936	DX00,0X00	,0x00,0	XFF, OXFF	,OXFF	,OXFF,OXF	F, OXFF,	OXFF, OXFF	,OXFF,OXFF	, OXFF, OXFF,	OXF1,
stm32f10x_rcc.h	1937	DX00,0X00	,0X00,0	x00,0x00	, OXFF	,OXFF,OXO	0,0X00,	oxoo, oxoo	,0x00,0x00	,0X00,0X00,	0X31 <mark>,</mark>
stm32f10x_spi.h	1938	DXOO, OX3F	,0X00,0	x00,0x00	, oxoo	,OXOO,OXF	F, OXFF,	oxoo, oxoo	,0x00,0x00	,0X00,0X00,	oxoc,
stm32f10x_tim.h	1939	DX00,0X3C	,OXFC,O	XCF, OXOO	, oxoo	,oxoo,oxo	0,0X00,	OXFF, OXFF	,0x00,0x00	,0X00,0X00,	oxoc,
stm32f10x_usart	1940	DX00,0X00	,0X00,0	X3C,OXFC	,OXCF	,0X00,0X0	0,0X00,	0X00,0X00	,OXFF,OXFF	,OXFF,OXFF,	OXF1,
🛄 misc. h 🔍	1941	DXFF, OXFF	,OXFF,O	XFF, OXFF	,OXFC	,OXFC,OXC	F, OXOO,	0X00,0X00	,0x00,0x00	,OXFF,OXFF,	oxoc,
< >	1942	pxoo,oxoo	,0x00,0	x00,0x00	, 0X00	,0X00,0X3	F, OXO3,	OX3F,OX00	,0X00,0X00	,0X00,0X00,	OXFI,
E P {} F 0, I	1943	DXFF.OXOO	.nxnn.n	100.000	. nxnn	.nxnn.nxn	n.nxnn.	NX3F.NXFF	.OXFF.OXOO	.nxnn.nxnn.	nxnr.
uild Output											