BEGV641A USER MANUAL

LCD Embedded System, Atmel ATmega644p MCU, Graphic 240x128 STN LCD, LED/white backlight, RS232/RS422/RS485, I²C /SPI 64KB in-system programmable Flash No Operating System required



CONTENTS

Precaution Packing Contents				P3 P4
Chapter 1 Introduction				P5
-	1-1	Features		P6
	1-2	Board Layout		P6
	1-3	Block Diagram		P7
	1-4	Mechanical Dimension		P7
	1-5 1 e	Board Specifications		P8
Chapter 2 Installation	1-0	Ordening mormation		Po P9
	2_1	lumpers		. 0 P10
	2-1	Jumpers	2.1.1 Contract adjust	
			2-1-1 Contrast adjust	P10
			2-1-2 Frame ground	P10
			2-1-3 RS-422/485 VDD/ground	P10
	2_2	Connectors	2-1-4 Screw hole ground	P11 P12
	2-2	Connectors	2-2-1 Connector & Pin Definition	P12
			2-2-2 Pin vs. Function Diagram	P15
Chapter 3 MCU Port Mapping				P18
	3-1	MCU Pin Configuration		P19
	3-2	MCU Port Mapping		P20
			3-2-1 LCD Controller	P20
			3-2-2 Touch Panel	P20
			3-2-3 RS-232/RS-422/RS-485	P20
			3-2-4 Enable Backlight	P20
			3-2-5 EEPROM/I ² C	P20
			3-2-6 2-wire serial port	P20
			3-2-7 SPI	P21
			3-2-8 General I/O	P21
Chapter 4 Software Development To	ol &	Utility		P22
	4-1 A	ATMEL ATmega644p Tool		P23
			4-1-1 Download software from AVR studio website	P23
			4-1-2 Additional tool for C language	P31
	4-2 E	Execute AVR studio 4.16 on	designer PC	P35
	4-3 I	n-system programmer AVR	ISP mk II	P43
	4-4 E	Bolymin Free Software Utilitie	es	P61
			4-4-1 Website links	P61
			4-4-2 Introduction of BOLYMIN software utilities	P62
			4-4-3 Software utilities function description	P70
Appendix A: LCD Controller Specific	catio	on		

Appendix B: EEPROM Specification

Appendix C: ATMEL ATmega644p MCU Specification



Precaution



This device is designed to meet the requirement in part 15 of the FCC rules. Operation is subject to conditions ruled under FCC part 15.

Please check packing content upon receiving BEGV641A parcel, make sure that all materials and options are packed inside parcel according to your order.

BEGV641A LCD Embedded module

- Touch panel
- Software Utility Disc
- $\hfill\square$ ISP Cable (option)
- □ ATMEL software development tool



BOLYMIN

Chapter 1 Introduction



This chapter is to offer you basic information regarding BEGV641A, to help you incorporate BEGV641A into your system.

Contents include:

1-1 Features

1-2 Board Layout

1-3 Block Diagram

1-4 Mechanical Dimension

1-5 Board Specifications

1-6 Ordering information

BOLYMIN



1-1 Features

This BEGV641A is designed based on ATmega644p microprocessor, which requires no operating system to run on. Together with a 240x128 STN LCD and LED backlight built-in, this all-in-one LCD embedded system BEGV641A help designer enhance a compact design with cost saving, space saving, and design phase saving.

Armed with RS232, RS422/485, I²C and SPI interface port, this BEGV641A communicates many devices and peripherals. The BEGV641A is therefore suitable to sit as a industrial control panel for factory automation equipment, electronics instrument, HMI (human-machine interface), office automation equipment, medical equipment, parking system, ticketing system.. and so on.

There are five LCD colors among choices: STN/gray, STN/yellow-green, STN/blue, FSTN/gray, and FSTN/black. All comes with LED/white backlight.

64KB in-system self-programmable Flash offers sufficient ROM size for designer to develop software, further to control LCD, touch panel, and interfaces.

Designer may simply design this BEGV641A into your application as you are designing a ATMEL CPU board, without worrying LCD module and other interfaces, since they are all on one board.

1-2 Board Layout

This layout shows the location of each important IC, connector and jumper. Please refer to chapter 2 for further information on jumper and connector.





1-4 Mechanical Dimension

(Drawing 1.4)





1-5 Board Specifications

(Table 1.5)

MCU	High-performance, Low-power AVR [®] 8-bit microprocessor ATMEL ATmega644P	
Memory	64K Bytes In-System Self-Programmable Flash 2K Bytes Internal EEPROM 4K Bytes Internal SRAM 3 x 64K Bytes External EEPROM(optional)	
Display	Support 8-bit single-scan resolution 240 x 128 monochrome STN LCD, with edge LED white backlight only	
Touch Panel(optional)	Support four-wired resistive touch panel	
Serial Ports	Support 1 x RS232 port, and 1 x RS232/RS422(isolated)/RS485(isolated) co-shared port Support 1 x full-duplex, three-wired synchronous data transfer SPI port Support 1 x two-wired serial interface to 250 KHz data transfer speed	

1-6 Ordering Information

(Table 1.6)					
Part No.	Description	RS232-A	RS232-B	RS422	RS485
BEGV641A	Dual RS232	X	Σζ		
BEGV641A1	One RS232	₩ Z			
BEGV641A2	One RS232/One RS422	₩ Z		Σ_{i}^{i}	
BEGV641A3	One RS232/One RS485	\$			Δ

Display: FSTN/Gray LCD,LED/White Backlight(Default)

Chapter 2 Installation



Abstract

This chapter is to offer designer fundamental information of BEGV641A jumpers and connectors, in order to help designer configure correct setting and connection between BEGV641A and system application.

Contents include:

2-1 Jumpers

2-2 Connectors



2-1 Jumpers

This section is to indicate location and function of each jumper on BEGV641A, which user can arrange according to the needs of different application desired. Be careful when setting jumper, user maybe need tool such as needle-nose pliers to help setting. Please note, jumpers not described here are intended to keep as factory default setting. Please consult Bolymin before trying to change default setting.

The table listed below describes location and function of each available jumper.

Jumpers: (Table 2.1)

Label	Function
VR	LCD contrast control
JFG	Frame ground
JG	RS-422/RS-485 VDD/ground
JHG	Screw hole ground

Detail location and function of each jumper is illustrated below.

2-1-1 Contrast Adjust

(Drawing 2.1.1)



VR	Contrast adjust	
10K ohm	Inside	default
N/A	Outside	

2-1-2 Frame Ground





JFG	Frame ground	
short	Connect metal frame with GND	
open	Not connected metal frame with ground	default

BOLYMIN

2-1-3 RS-422/RS-485 VDD/Ground

(Drawing 2.1.3)



S1	S2	RS-422/RS-485 VDD/Ground	
short	short	internal VDD/ground for isolated RS-422/485	
open	open	External VDD/ground for isolated RS-422/485	default

2-1-4 Screw hole ground



JHG	Frame ground	
short	Connect screw hole with GND	
open	Not connected screw hole with ground	default



2-2 Connectors

Connectors are the key link between BEGV641A and external devices. Detail locations and functions of available connectors are tabled and illustrated below.

Connectors: (Table 2.2)

Label	Pin No.	Function	Item No.	
CN1	32	Connector to connect via	Hirose DF11C-32DP-2V	
		Bolymin-defined adaptor to		
		Atmel AVR ISP		
CN2	4	FPC connector to connect touch	Molex 52207-0417	
		panel (T/P)		

2-2-1 Connectors & Pin Definition



• Pin Definition: BEGV641A-Dual RS232 (Table 2.2.1a)

Pin No.	Signal	Pin No.	Signal
1	GND	2	EEPSDA
3	VDD	4	EEPSCL
5	NC	6	EEPWP
7	NC	8	IOB
9	/Reset	10	NC
11	/SS	12	NC
13	MOSI	14	NC
15	MISO	16	NC
17	SCK	18	NC
19	RX0	20	NC
21	TX0	22	NC
23	RX1	24	NC
25	TX1	26	NC
27	SDA	28	NC
29	SCL	30	NC
31	IOA	32	NC



• Pin Definition: BEGV641A1-One RS232 (Table 2.2.1b)

Pin No.	Signal	Pin No.	Signal
1	GND	2	EEPSDA
3	VDD	4	EEPSCL
5	NC	6	EEPWP
7	NC	8	IOB
9	/Reset	10	NC
11	/SS	12	NC
13	MOSI	14	NC
15	MISO	16	NC
17	SCK	18	NC
19	RX0	20	NC
21	TX0	22	NC
23	NC	24	NC
25	NC	26	NC
27	SDA	28	NC
29	SCL	30	NC
31	IOA	32	NC

• Pin Definition: BEGV641A2-One RS232/One RS422 (Table 2.2.1c)

Pin No.	Signal	Pin No.	Signal
1	GND	2	EEPSDA
3	VDD	4	EEPSCL
5	NC	6	EEPWP
7	NC	8	IOB
9	/Reset	10	NC
11	/SS	12	422RP
13	MOSI	14	422RN
15	MISO	16	422TP
17	SCK	18	422TN
19	RX0	20	VDD2
21	TX0	22	VDD2
23	NC	24	GND2
25	NC	26	GND2
27	SDA	28	NC
29	SCL	30	NC
31	IOA	32	NC

• Pin Definition: BEGV641A3-One RS232/One RS485 (Table 2.2.1d)

Pin No.	Signal	Pin No.	Signal
1	GND	2	EEPSDA
3	VDD	4	EEPSCL
5	NC	6	EEPWP
7	NC	8	IOB
9	/Reset	10	NC
11	/SS	12	NC
13	MOSI	14	NC
15	MISO	16	485P
17	SCK	18	485N
19	RX0	20	VDD2
21	TX0	22	VDD2
23	NC	24	GND2
25	NC	26	GND2
27	SDA	28	EN485
29	SCL	30	NC
31	IOA	32	NC

2-2-1-1 Power & Ground (Table 2.2.1.1)

Signal	Туре	Pin No.	Description
GND	Р	1	Logic power supply (ground)
VDD	Р	3	Logic power supply (+5V)
VDD2	Р	20	External Power for isolated RS-422/485(+5V)
VDD2	Р	22	External Power for isolated RS-422/485(+5V)
GND2	Р	24	External Ground for isolated RS-422/485(ground2)
GND2	Р	26	External Ground for isolated RS-422/485(ground2)

2-2-1-2 Serial I/O (Table 2.2.1.2)

Signal	Туре	Pin No.	Description
/SS	U	11	SPI Chip select
MOSI	U	13	MOSI is the mater data output line, when SPI on module is configured as a master. When SPI is configured as a slave. This pin reverse the role.
MISO	U	15	MISO is the master data input line, when SPI is configured master. When SPI Is configured as a slave. This pin reverse the role.
SCK	U	17	SPI clock
RX0		19	Receiver of first RS232 with driver
TX0	0	21	Transmitter of first RS232 with driver
RX1		23	Receiver of second RS232 with driver
TX1	0	25	Transmitter of second RS232 with driver
SDA	U	27	Data of 2-wire serial interface, it can be programmed as IO.
SCL	U	29	Clock of 2-wire serial interface, it can be programmed as IO.
EEPSDA	Bi	2	Data of 2-wire serial interface for additional EEPROM update.
EEPSCL	Bi	4	Clock of 2-wire serial interface for additional EEPROM update
EEPWP		6	Write protect of additional EEPROM
422RP		12	no inverting receiver of RS422
422RN		14	inverting receiver of RS422
422TP/485P	Bi	16	When it configured as RS422 it act as no inverting transmitter, When is configured as RS 485 it acts as positive differential IO.
422TN/485N	Bi	18	When it configured as RS422 it acts as inverting transmitter, When is configured as RS 485 it acts as negative differential IO.
EN485	0	28	Enable RS-485

2-2-1-3 General I/O (Table 2.21.3)

Signal	Туре	Pin No.	Description
/Reset		5	Auxiliary moment reset for external input
IOA	U	31	I/O port (ATmega644P portA.4)
IOB	U	8	I/O port (ATmega644P portA.7)



2-2-2 Pin vs. Function Diagram

2-2-2-1 Power/LCD/Backlight- The following function block illustrates system power supply, contrast adjustment, LCD display driver, and the single chip.



2-2-2-2 In-System Programming(ISP) using AVRISP MKII by Atmel

BEGV641A supports ISP operation, which allows designer to write software into ATmega644p via ATMEL AVR ISP MKII writer without removing ATmega644p from board. Further info about ATMEL MKII can be found in http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3808. Both Bolymin ISP adaptor board (part # MGI01-0\$) and ATMEL MKII included set (part # OP-641A001) is available for purchase.



2-2-2-3 RS-232

For serial communication, BEGV641A support RS-232 communication through an ICL232 compatible serial UART.





2-2-2-4 SPI

BEGV641A offers SPI port. Designer has to define this port as SPI by software, or to use Bolymin SPI driver (free utility).



2-2-2-5 I²C/EEPROM

BEGV641A offers I²C port. Via this I²C port, designer may control 64Kbytes x 3 EEPROM in-system and external I²C devices.



2-2-2-6 RS-422

BEGV641A offers 1 x RS-422(isolated) port.



BOLYMIN

2-2-2-7 RS-485

BEGV641A offers 1 x RS-485(isolated) port.



Chapter 3 MCU port mapping



This chapter explains ATmega644p MCU pin configuration and port mapping toward key elements such as LCD, Touch Panel, RS-232, RS-422, RS-485, LED Backlight, EEPROM/I²C, 2-wire serial port, SPI, and General I/O. BOLYMIN

3-1 MCU Pin Configuration

(Drawing 3.1, ATmega644p MCU)



BOLYMIN



3-2 MCU Port Mapping

3-2-1 LCD Controller

(]	ſable 3.2a)	
-	MCU ATmega644P	LCD Controller
	PORTB.0	DB0(LSB)
	PORTB.1	DB1
	PORTB.2	DB2
	PORTB.3	DB3
	PORTD.4	DB4
	PORTD.5	DB5
	PORTD.6	DB6
	PORTD.7	DB7(MSB)
	PORTC.7	LCD /RESET
	PORTC.5	LCD RD
	PORTC.4	LCD WR
	PORTC.6	LCD C/D

3-2-2 Touch Panel

(Table 3.2b)

MCU ATmega644P	Touch Panel
PORTA.0	X1
PORTA.1	Y1
PORTA.2	X2
PORTA.3	Y2

3-2-3 RS-232/RS-422/RS-485

(Table 3.2c)

MCU ATmega644P	RS-232/422/485
PORTD.0	RX0
PORTD.1	TX0
PORTD.2	RX1
PORTD.3	TX1

(Table 3.2d)

MCU ATmega644P	RS-485
PORTA.5	Enable RS-485

3-2-4 Enable Backlight

(Table 3.2e)

MCU ATmega644P	LED backlight
PORTA.6	Enable Backlight

3-2-5 EEPROM/I²C

(]	Table	e 3.2	2f)
· / '			/

MCU ATmega644P	EEPROM/ I ² C
PORTC.2	EEPROM SDA
PORTC.3	EEPROM SCL

3-2-6 2-wire serial port

(Table 3.2g)

MCU ATmega644P	2-wire serial port
PORTC.1	SDA
PORTC.0	SCL

BOLYMIN

3-2-7 SPI

(Table 3.2h)

MCU ATmega644P	SPI
PORTB.4	/SS
PORTB.5	MOSI
PORTB.6	MISO
PORTB.7	SCK

3-2-8 General I/O

(Table 3.2i)

MCU ATmega644P	General I/O
PORTA.4	IOA
PORTA.7	IOB



Abstract

This chapter explains ATmega644p MCU software development tool AVR ISP mkII writer and Bolymin free software utilities.



4-1 ATMEL ATmega644p Software Development Tool

Designers may download software development tool from AVR Studio website <u>http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725</u>, or from BOLYMIN utility disk.

4-1-1 Download software from AVR Studio website

step1: Download design software

dister	AVR Studio 4.16 (build 628) (126 MB, updated 2/09)
Rea	AVR Store: 4.16 includes new device support and numerous overall enhancements; AVR QTouch Studio; Section active Section activ

First llame *		
Last Name*		
Title		
Job Function*	Select	
Company*		
Address 1*		
Address 2		
City*		
State/Province* (US and Canada only)	Select	
Province* (only if outside US and Canada)		
Postal Code*		
Country*	Select	
Telephone*		
E-mail Address*		
Fax	7	
What is the annual unit quantity usage?*	Fill in necessary information and	
What is the estimated timeframe for need?*	click to download	
Would you like to receive news about Atmel p		
	Click to Download	

Step2: install AVR studio on designer PC





AVRStudio4 - InstallShield Wizard Welcome to the InstallShield Wizard for AVRStudio4 The InstallShield Wizard will install AVRStudio4 on your computer. To continue, click Next. InstallShield Velcome to the InstallShield Wizard will install AVRStudio4 on your computer. To continue, click Next.



AVRStudio4 - InstallShield Wizard License Agreement Please read the following license agreement carefully. Welcome to AVR Studio from Atmel Corporation. ~ AVR Studio is a Development Tool for the entire AVR family of microcontrollers, including tinyAVR, megaAVR, picoPower, and XMEGA AVR devices. The AVR Studio is free of charge and may be freely copied and distributed in its original form. AVR Studio enables the user to fully control execution of programs on the AVR In-Circuit Emulator or on the included AVR Instruction Set Simulator. AVR Studio supports source level execution of Assembly and C/C++ programs assembled with the Atmel Corporation's included AVR Assembler or tools from 3rd party vendors. AVR Studio runs under Microsoft Windows 98, Windows NT, Microsoft Windows 2000, Windows XP and Windows Vista. AVR Studio is continously developing. In order to get latest upgrades of AVR Studio, please visit our web site www.atmel.com and check out the AVR page. V. I accept the terms of the license agreement <u>Print</u> I do not accept the terms of the license agreement. InstallShield < <u>B</u>ack Cancel Next > Accept terms to continue













AVRStudio4 - InstallShield Wi	izard	
Setup Status		
	AVRStudio4 is configuring your new software installation.	
	Installing	
InstallShield		Cancel
	Wait for seconds for installation	



AVRStudio4 - InstallShield Wizard





4-1-2 Additional tool for C language

For C language designers, additionally please download and install AVR gcc from <u>http://sourceforge.net/project/downloading.php?group_id=68108&filename=WinAVR-20090313-install.exe&a=6759369</u>





Close other applications to continue

WinAVR 20090313 Setup	
L icense Agreement Please review the license terms before installing WinAVR 20090313.	
Press Page Down to see the rest of the agreement.	
WinAVR Licensing Information	
GNU Binutils is distributed under the GNU GPL License. GNU Compiler Collection (GCC) is distributed under the GNU GPL License. avr-libc is distributed under a BSD License. GNU Debugger (GDB) is distributed under the GNU GPL License. Insight is distributed under the GNU GPL License SimulAVR is distributed under the GNU GPL License AVARICE is distributed under the the GNU GPL License. If you accept the terms of the agreement, click I Agree to continue. You must accept agreement to install WinAVR 20090313.	the
< Back LAgree Click I agree t	Cancel
WinAVR 20090313 Setup	
Choose Install Location	
Choose the folder in which to install WinAVR 20090313.	
Setup will install WinAVR 20090313 in the following folder. To install in a different fold Browse and select another folder. Click Next to continue.	er, click
Destination Folder C:\WinAVR-20090313 Browse	
Space required: 122.6MB Space available: 51.7GB	
< <u>B</u> ack <u>N</u> ext >	Cancel
Choose folder	to continue

l

Choose Components	
Choose which features of Win4	AVR 20090313 you want to install.
Check the components you wa install. Click Install to start the	nt to install and uncheck the components you don't want to installation.
Select components to install:	 Install Files Add Directories to PATH (Recommended) Install Programmers Notepad
Space required: 122.6MB	
	< <u>B</u> ack Install Cancel
	Choose components to install

WinAVR 20090313 Setup	
Installing Please wait while WinAVR 20090313 is being installed.	
Extract: libc.a Show <u>details</u>	
< <u>B</u> ack <u>N</u> ext >	Cancel
Wait for seconds for installation]



BOLYMIN

** AVR Studio and AVR gcc software are also available on Bolymin utility disk or Bolymin website.

4-2 Execute AVR Studio 4.16 on designer PC

Start→All programs→Atmel AVR Tools→AVR Studio 4

BOLYMIN



Welcome to AVR Stud	io 4		
15	New Project	Open Open	
A Strain	Recent projects		Modified
5			
		\frown	
Ver 4.16.628 🔽 Show of	dialog at startu		
	Click Ne	w Project to start	

Create pew rolect type	project s: Project name:
Atmel A	AVR Assembler CC C C C Create initial file Create folder Initial file:
Location: D:\AVR\BI	E240128_demotest
Ver 4.16.628 Ver 4.16.628	
Choose Atm	el AVR Assembler (assembly language),
or AVR GCC	(C code)

Image: Constraint of the system Create new project Project type: Atmel AVR Assembler Image: AVR GCC AVR GCC Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of the system Image: Constraint of the system Constraint of th	Project name: BEGV641A_demotest Create initial file Create folder Initial file: Main .c
Ver 4.16.628 🔽 Show dialog at startup	Next >>FinishCancelHelp
Choose AVR GCC here (<i>i</i> designed with C code), ar Initial file.	All Bolymin utility drivers are nd key in Project Name and

BOLYMIN

Welcome to AVR Studi	ο 4	
	Select debug platform and device Debug platform: Device:	
	AVB Dragon ATmega48P AVB ONE! ATmega48PA AVR Simulator ATmega64 AVR Simulator 2 ATmega64	
Studie	ICESO ATmega644 ICE 40 ATmega644 ICE 50 ATmega645 JTAG ICE ATmega6450 JTAGICE mkII ATmega649 ATmega6490 ATmega6430	
	Open platform options next time debug mode is entered	
Ver 4.16.628		
Choose AVR Simulator, ATmega644P, and click		
	Finish to continue	
		BOLYMIN
--	--	--
(— <i>I</i> /
AVR Studio - D:\AVR\BEGV641A\BEGV64	41A_demotest.c	
File Project Build Edit View Icols Debu		
	- Marina 1 / 2 / 2 / 2 / 2 / 2 / 2 / 2 / 2 / 2 /	
: Inace Disabled ・ ス 次 : 上 不		IN Way
Ava Gov C BEGV641A_demoist (default) G G Source Fales C Bedver Fales C Bewmal Dependencies C Other Fales C Other Fales	DAAYRABEGV641A_demotest.c	Name Value D AD_CONVERTER DAD_CONVERTER D AD_CONVERTER DAD_CONVERTER D AD_CONVERTER DAD_CONVERTER D BOOT LOAD EEFROM D EFFROM EEFROM D TRA FORTA D ORTA FORTA D ORTD FORTA D TIMER_COUNTER_0 FORTA D TIMER_COUNTER_1 FORTA D TIMER_COUNTER_2 FORTA D TIMER_COUNTER_2 FORTA D TARDOG Mame
	D:\&VR\BEGV641A\BEGV641A_demotest.e	4 Þ
Build		▼ X
Tal name 🧠 messafe 🦓 num n nines 🚺 Dieskbor	a no one sectored a	AVE Simulator Auto In La L Col 1 CAP NUM OVE

Here you can see software design screen, yet no hurry to start software design yet.



5	Active	default		Edit Config	gurations
General	1. Target name r 2. Clean/rebuild 3. Makefile and	Makefile nust equal project name. support requires "clean" target must exist in the sa	target. me folder		
elude Directories	Output File Name Output File	e: BEGV641A_demo default/	test.elf		
Libraries	Device: Frequency Optimization:	atmega644p 11059200 hz -00 💌	Unsigned (Unsigned) Pack Struc Short Enw	Chars (-funsigned-c Bitfields (-funsigned ture Members (-fpa ms (-fshort-enums)	har) 1-bitfields ck-struct)
2	Create Hex F	ïle 🦵 Genera	ate Map File	🗖 Generate	List File
			確定	取消	說明
	Choose Genera continue	al and key in necessar	y information to		















[2]. Program editing window



BOLYMIN	

I/O View ▼ X 🐔 🛛 🔚 🔚 AD_CONVERTER • Name Value DAD_CONVERTER ∃ DANALOG_COMPAR.... 🗉 🖹 BOOT_LOAD 🛨 📑 CPU 🗄 🧮 EEPROM ■ SEXTERNAL_INTERR.... ■ JIAG PORTA PORTB PORTB PORTC PORTD 🛨 🚼 SPI 🖅 🛞 TIMER_COUNTER_0 H S TIMER_COUNTER_1 🗄 🚫 TIMER_COUNTER_2 TWI USARTO USARTI Name Address Value Bits H DADC na (0x78) na (Ox7A) na (0x7B) **∃**DADMUX na (0x7C) DIDR0 na (0x7E)

[3]. I/O view window

[4]. Message window



[5]. Toolbars

:	<u>F</u> ile	<u>P</u> roject	<u>B</u> uild	<u>E</u> dit	∐iew	Tools	Debug	$\underline{W} indow$	Help																
: [) 💕	.	U X	i 1	1	1 C	°- 🌳	÷ #4 /	6 % (% %	5 (F 1	e : 👸	×	1	π		t 11	S	• • • [,	1	*{}	63	周	× 📰	(j. 11
; T	race Disa	bled		• %	Ж,	<u>6</u> *	不 🕴 🖸	5 EVB	ک <mark>ر ا</mark>	₽ ^{a auto}	1	🚯 🚳	×	0											



4-3 In-System Programmer AVR ISP mkll



** Inside mkII box(1), designer will get DVD(2), mkII device(3), and USB cable(4). Please be sure that all parts are packed inside.



Steps: 1.Get mkII connected to PC

★¥R Studio - D:\A¥R\BE2401	28_ d	lemotest\BE240128_de	mot	est/uart.c				
<u>File Project Build Edit V</u> iew	Tool	ls <u>D</u> ebug <u>W</u> indow <u>H</u> elj	P					
i 🗋 💕 🗟 🖉 🙂 🐰 🖻 🛍 🤅		AVR Pilog	%	%%;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;				
; Trace Disabled 🔹 🔩 😒		ICESO Upgrade	42	No 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
AVR GCC		ICESO Selftest						
BL12864G_AVR (default)		JTAGICE mkII Upgrade	E24	0128_demotest\BE24012				
±		AVR ONE! Upgrade	R\E	BE240128_demotest\BE24				
🗉 🔄 External Dependencies		AVRISP mkII Upgrade		R\BF240128_demotest\B				
		AVR Diagon Upgiade						
		STK600 Upgrade		:\AVR\BE240128_demote				
		<u>C</u> ustomize		D:\AVR\BE240128_den				
		Options		}				
		Show Key Assignments		//! gets a byte (if				
	<u>P</u> lug-in Manager			uint8_t uartReceive				
	RWB	Program AVR 🕨	Con	Connect d != uT				
	dana Tara	FLIP 3.1.0	RWR	Auto Connect				
	×	AVR Wiseless Studio	-	Write Flash uTail >				
	1	AVR Battery Studio	YE2	Write EEPROM In TRUE				
	ar	AVR QTouch Studio	E	Read EEPROM				
			RUTO	Start Auto				
//! flush all data								
Choose Tools→Program AVR→Connect to continue								

Select AVR Programmer	
Platform:	Cancel Baud rate:
Tip: To auto-connect to the programmer used last time, press the 'Programmer' button on the toolbar.	Baud rate changes are active immediately.
Note that a tool cannot be used for programming as long as it is connected in a debugging session. In that case, select 'Stop Debugging' first.	
Disconnect to continue Choose AVRISP mkII→USB→Connect to continue]

RISP mkII in ISP mode with ATmega644P	
fain Program Fuses LockBits Advanced HW Settings HW Info Auto Device	
Flash	
 Use Current Simulator/Finulet - PEASH Memory Input HEX File D:\BE240128\BE240128\default\BE240128\hex 	
Program Verify Read	
 Use Current Simulator/Empleter EEEECM Memory Input HEX File D:\BE240128\BE240128\default\BE240128\default\BE240128.eep 	
Program Verify Read	
ELF Production File Format Input ELF File D:\BE240128\BE240128\default\BE240128.elf	
Fuses and lockbits as Program Save be specified before a ELF	ttings must aving to
etting revisions HW: 0x01, FW Major: 0x01, FW Minor: 0x0c OK	
Choose Hex file for ATmega644p flash → Program	
Choose Hex file for ATmega644p EEPROM →Program	
Choose ELF file for fuses and lockbits → Program	

Start software burning

AVRISP mkII in ISP mode with ATmega644P	
Main Program Fuses LockBits Advanced HW Settings HW In	nfo Auto
 Erase Device Check signature Program FLASH Verify FLASH Read FLASH Program EEPROM Verify EEPROM Read EEPROM Write osc. cal. byte Program fuses Verify fuses Read fuses Program lock bits Verify lock bits Read lock bits Fnable fuse warnings 	<u>S</u> tart ■ Log to file © Overwrite © Append. Browse
Detecting on USB' AVRISP mkII with serial number 000200002421 found. Getting isp parameter SD=0x06 OK	<u>~</u>
Choose Auto and necessary configure and click Start to program	

VRISP mkII in ISP mode with ATmcgn644P Main Program Finase Device Check signature Check signature Start Program FLASH Start Program EEPROM Start Program lock bits Log to file Program lock bits Coverwrite Program lock bits Append Read FLASH Browse Program lock bits Log to file Program lock bits Coverwrite Program lock bits Browse Verify lock bits Browse Verify tuse saddres 0 to 2. 0xEF, 0xD9, 0xFF 0K1 Reading tuses addres 0 to 2. 0xEF, 0xD9, 0xFF 0K1 Verify tuse saddres 0 to 2. 0xEF, 0xD9, 0xFF 0K1 Programming mode 0K1			BOLYMIN
VRISP mkII in ISP mode with A Imega644P Main Program Fuses LockBits Advanced HW Settings HW Info Auto France Device Check signature Program FLASH Verify FLASH Read FLASH Read FLASH Read FLASH Program lock bits Verify fuses Read fuses Verify fuses Read fuses Program lock bits Program lock bits Verify fuses Read lock bits Read lock			
Main Program Fuses LockBits Advanced HW Settings HW Info Auto Erass Device Check signature Program FLASH Verify FLASH Program EEPROM Write oss. cal. byte Program fuses Verify fuses Read fuses Program lock bits Overwrite Append Browse Keth contents is equal to file. OK Writing fuses address to 12. OxEF, OxD9, 0xFF OK! reading fuses address to 12. OxEF, 0xD9, 0xFF OK! reading fuses address to 12. OxEF, 0xD9, 0xFF OK! reading fuses address to 2 OxEF, 0xD9, 0xFF OK! reading fuses address to 2 OxEF, 0xD9, 0xFF OK! reaving programming mode OK! 	VRISP II	akII in ISP mode with ATmega644P	
Main Program Fuses LockBits Advanced HW Settings HW Info Auto Image: Check signature Check signature Start Start Image: Check signature Program FLASH Start Start Image: Check signature Program FLASH Start Start Image: Check signature Program FLASH Image: Check signature Start Image: Check signature Program FLASH Image: Check signature Start Image: Check signature Program FLASH Image: Check signature Start Image: Check signature Program EEPROM Image: Check signature Image: Check signature Image: Check signature Image: Check signature Program lock bits Image: Check signature Image: Check signature Image: Check signature Image: Check signature Program lock bits Image: Check signature Image: Check signature Image: Check signature Image: Check signature Image: Check signature Program lock bits Image: Check signature Image: Check signature Image: Check signature Image: Check signature Image: Check signature Image: Check signature Image: Check signature			
Enase Device Start Check signature Program FLASH Verify FLASH Program EEPROM Program EEPROM Verify EEPROM Write osc. cal. byte Program fuses Program lock bits C Overwrite Program lock bits C Overwrite Verify lock bits C Append Browse Browse	Main	Program Fuses LockBits Advanced HW Settings HW Info Auto	
✓ Enable fuse warnings LASH contents is equal to file OK Virting fuses address 0 to 2 0xEF, 0xD9, 0xFF OK! teading fuses address 0 to 2 0xEF, 0xD9, 0xFF OK! use bits verification OK use bits verification OK Good job! Software burning is done!	 Erass Che Prog Veri Reaa Prog Veri Reaa Writ Prog Veri Reaa Prog Veri Reaa Prog Exact Prog Veri Reaa Prog Reaa Prog Reaa Reaa Reaa Reaa Reaa Reaa Reaa 	* Device Start <p< td=""><td></td></p<>	
Good job! Software burning is done!	✓ Enal LASH con Vriting fus leading fu use bits vi eaving pr	Browse ble fuse warnings ntents is equal to file OK ses address 0 to 2 0xEF, 0xD9, 0xFF OK! uses address 0 to 2 0xEF, 0xD9, 0xFF OK! terification OK rogramming mode OK!	
Good job! Software burning is done!			
Good job! Software burning is done!			
	[Good job! Software burning is done!	

Product Function Verify



BOLYMIN

On PC: Start→All programs→Telecommunication→Hyper Terminal

連線描述	?	\mathbf{X}
戦 新的連続	Į	
諸爲這個連線輸 名稱(N):)入名稱並選擇圖示:	
圖示(I):		
Please	define a name and choose a icon for th	is connection.
and cli	ck Yes to continue.	
連線到	? 🞽	
🧞 test		
請輸入要撥號的	1電話號碼詳細資料:	
國家(地區)(<u>C</u>):	中華民國 (886) 🔹 🗸	
區碼(E):		
電話號碼(P):		
使用連線(N):	COM1	
	Tervide	
Choose	a suitable COM port, and click Yes to c	ontinue.

			BOLYMIN
COM1 內容		? 🔀	
連接埠設定			
每秒傳輸位元(B):	115200	~	
資料位元(D):	8	~	
同位檢查(P):	無	~	
停止位元③:	1	~	
流量控制①:	無	✓	
	「漂盾」) 預設(店(P)	
	Jac John Par		
 	取消	<u> </u>	
	Key in COM port setti	ng:115200/8/No/1/No,	
	and click Yes to conti	nue.	



After seeing above screen, designer can operate on PC & BEGV641A:

- A). Enter number "1": for I²C EEPROM function test (1st 2nd 3rd EEPROM) B). Enter number "2": for 2nd UART function test
- C). Enter number "3": for ATmega644P internal EEPROM function test
- D). Enter number "4": for 4-wire SPI Master function test
- E). Enter number "5": for 4-wire SPI Slave function test
- F). Enter number "6": to touch panel function test
- G). Enter number "7": for touch panel calibration (5 point)
- H). Enter number "8": for LCD function test
- I). Enter number "9": for backlight brightness adjustment function test



A). Enter number "1": for I²C EEPROM function test (1st 2nd 3rd EEPROM)



- Enter number "1": for I²C EEPROM function test (1st EEPROM)
 Enter number "2": for I²C EEPROM function test (2nd EEPROM)
 Enter number "3": for I²C EEPROM function test (3rd EEPROM)
- 4). Enter number "0": to return to main screen

 ● text - 超数终端线 ● 体式 - 超数终端线 ● 論報 (2) 時叫(2) 轉送(1) 説明(1) ● 論 / (2) 論 	
12C - A2 $0000 - FF F$	
0 - EXIT P - PREV N - NEXT 1 - 55 2 - AA 3 - 00 4 - FF 連線000141 自動値測 1152008-N-1 SCROLL CAPS NUM 題 7	I 2C - A2 I T XREXEL T NLR SASE

- Enter number "1": to write 55 on current screen
 Enter number "2": to write AA on current screen
- 3). Enter number "3": to write 00 on current screen
- 4). Enter number "4": to write FF on current screen
- 5). Enter "P": to switch to previous page6). Enter "N": to switch to next page
- 7). Enter number "0": to return to the previous page

B). Enter number "2": for 2nd UART function test



BOLYMIN

- 1). Enter number "1": to send 1 byte (0x55)
- 2). Enter number "2": to send 1 byte (0xAA)
- 3). Enter number "3": to send 1 word (0x5555)
- 4). Enter number "4": to send 1 byte (0xAAAA)
- 5). Enter number "0": to return to main screen

Auto receive byte or word

C). Enter number "3": for ATmega644P internal EEPROM setting

● 1031 - 超数終肩機 検索(m) 絶報(m) 検報(m) 随助(m) 施祥(m) 貸明(m)		
[
EEPROM		
0000 - 55 AA F5 6D 5E B7 4F 14 97 BE C8 1	9 87 43 47 AF	
0010 - 46 BE 9F E0 5D BB 2D EC 25 43 C0 B	3 25 52 FF FF - FF FF FF FF	
0030 - FF F	FF FF FF FF	
0040 - FF F	· FF FF FF FF - FF FF FF FF	
0060 – FF F	FF FF FF FF	
0070 - FF F	· FF FF FF FF - FF FF FF FF	
0090 - FF F	E FE FE FE FE	
00A0 - FF F	·	
ŎŎĊŎ - FF F	E FE FE FE FE	
0000 - FF F	·	
ŎŎĔŎ – FF F	· FF FF FF FF _	
	1032000	
1 - 33 2 - HH 3 - 00 4 - FF	0000 - 5500571B4B9	EEPROM
	0020 - FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	EXIT
	0030 - EFFEFEFEFEFE	PREY
	0040 - FFFFFFFFFFFF	NEXT
	N 0060 - FFFFFFFFFFFFFFFFF	🦷 🕅 🥂 🧰 🤁
	2 0010 - FFFFFFFFFFFFF	55
	<u> 0090 - FEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</u>	ĂĂ I-A
	BORO - FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	
	00C0 - FFFFFFFFFFFFFF	F F
	00D0 - FFFFFFFFFFFFF	
	00F0 - FFFFFFFFFFFFFF	

** Address 0x0000~0x001D is for touch panel calibration data

- Enter number "1": to write 55 on current screen
 Enter number "2": to write AA on current screen
 Enter number "3": to write 00 on current screen
 Enter number "4": to write FF on current screen
- 5). Enter "P": to switch to previous page6). Enter "N": to switch to next page
- 7). Enter number "0": to return to main screen



- Enter number "1": to send 1 byte (0x55)
 Enter number "2": to send 1 byte (0xAA)
 Enter number "3": to send 1 word (0x5555)
 Enter number "4": to send 1 byte (0xAAAA)
 Enter number "0": to return to main screen



- Enter number "1": to receive 1 byte
 Enter number "2": to receive 1 word
 Enter number "0": to return to main screen



240128 - 超銀終端機		
檔案 [E] 編輯 [E] 檢視 [V] 呼叫 (C) 轉送 (I) 說明 [H]		
C 📽 🚿 🖞 🍟		
X=661, Y=653, DX=158, DY=85 X=535, Y=574, DX=136, DY=63 X=547, Y=580, DX=137, DY=65 X=559, Y=458, DX=102, DY=67 X=561, Y=455, DX=101, DY=67 X=561, Y=454, DX=101, DY=67 X=696, Y=407, DX=87, DY=91 X=343, Y=737, DX=184, DY=29 X=341, Y=740, DX=184, DY=29 X=340, Y=740, DX=184, DY=28 X=339, Y=737, DX=184, DY=28 X=436, Y=819, DX=207, DY=45		
X=483, Y=810, DX=204, DY=54 X=598, Y=790, DX=198, DY=74 X=596, Y=782, DX=195, DY=73 X=595, Y=786, DX=197, DY=73 -		
連線 00:31:34 自動偵測 115200 8-N-1 SCROLL CAPS		
	о и	
	EXIT	

- Using light pen to touch a point on touch panel
 X, Y values are touch panel coordinates
- 3). DX, DY values are LCD coordinates (DX=0/DY=0 if touch panel is not calibrated.)
- 4). Enter number "0": to return to main screen



- 2). Please repeat above process to calibrate 5 points.
- 3). Touch panel calibration is finished.

H). Enter number "8": for LCD function test



BOLYMIN

- Enter number "1": to LCD show full on
 Enter number "2": to LCD show vertical line
 Enter number "3": to LCD show horizontal line
- 4). Enter number "4": to LCD show half
- 5). Enter number "5": to LCD show cross dot
- 6). Enter number "6": to LCD show character
- 7). Enter number "7": to LCD show picture 1
- 8). Enter number "8": to LCD show picture 2
- 9). Enter number "0": to return to main screen



H line

Half



Picture1

Picture2



- Enter number "1": to backlight brightness + 5
 Enter number "2": to backlight brightness 5
 Enter number "0": to return to main screen

4-4 Bolymin Free Software Utilities

4-4-1 Website Links

• Touch Panel Driver www.bolymin.com.tw/embedded/utility/BEGV641A/tpdriver.zip BOLYMIN

- UART Driver (RS-232/485/422) www.bolymin.com.tw/embedded/utility/BEGV641A/uartdriver.zip
- SPI Driver www.bolymin.com.tw/embedded/utility/BEGV641A/spidriver.zip
- I²C Driver

www.bolymin.com.tw/embedded/utility/BEGV641A/i2cdriver.zip

Backlight Driver
 www.bolymin.com.tw/embedded/utility/BEGV641A/backlightdriver.zip

• LCD Driver www.bolymin.com.tw/embedded/utility/BEGV641A/lcddriver.zip



4-4-2 Introduction of Bolymin software utilities

It is recommended to use Bolymin software utilities in order to speed up project development phase. However, designer may develop your own software utilities, if you find Bolymin utilities is not convenient to use.

On following paragraphs, we explain the way to add Bolymin software utilities into designer PC.



Note: Under AVR Studio4, Project→Configuration Options

BL12864G_AVR	Project Options 🛛 📉
2200	Include File Search Path:
General Include Directories	
Libuaries Memory Settings	
	Choose Include Directories, and Include File Search Path to continue 確定 取消 説明

Note: Adding into Header files

制管咨封床	
🗐 🦳 8328d37d388ca4a6ce622c4b	
ArcBackupDeviceInfo	
🗉 🧰 AUD-AM09MY-105	
AVL-2500-V01-emn&emp	
E Guid790	
awu≀o∋	
😑 🛅 BE240128_demotest	
🖨 🧰 BOL_lib_ver3	
BOL_lib	
🖅 🛅 default	
🖬 🧰 BE240128_test	
biosview	×



BE240128 Projec	t Options	
General	Library Search Path:	
Include Directories	Available Link Objects: libc.a libobjc.a libopintf_fit.a libprintf_min.a libscanf_fit.a libscanf_min.a	Add Library> Remove Object Add Object
		Move up Move down
		確定 取消 説明

Note: For touch panel calibration, Floating-point operations is necessary, so please add libm.a and libprintf_flt.a in WinAVR.

BE240128 Project	Options	
General General Include Directories Libraries Memory Settings	Library Search Path: Available Link Objects: libc.a libm.a libobjc.a libprintf_flt.a libprintf_flt.a libscanf_flt.a libscanf_min.a	Link with These Objects: Add Library
Curtur Curius		Move down
		確定 取消 説明

BE240128 Projec	t Options	
General	Library Search Path:	
Include Directories Libraries Memory Settings	Available Link Objects: libc.a libobjc.a libprintf_flt.a libscaaf_flt.a libscaaf_flt.a libscaaf_min.a	Link with These Objects: Add Library> libm.a Remove Object Move up Move up Move down

BE240128 Project	t Options 🛛 🔀
2000 Geneval	Library Search Path:
include Directonies	Available Link Objects:
Libuaries	libc.a Add Library
Memory Settings	hbscanf_lit.a hbscanf_min.a Add Object
- <u> </u>	Move down Move down 確定 取消 説明
	HEAVE TOURING BARYS

BE240128 Project Options X Library Search Path: General 44 E Include Directories Available Link Objects: Link with These Objects: libc.a libm.a libm.a libprintf_flt.a Add Library --> libobjc.a libprintf_flt.a libprintf_min.a libscanf_flt.a libscanf_min.a Remove Object Libraries Add Object . Memory Settings Move up Move down ⊡ 取消 說明 確定

Note: Adding into Obj

Add Link Object						? 🔀
查詢(1):	🗁 obj	•	4	<u>٤</u> (* 🎟	•
我最近的文件 反一 東面 教的文件 教的電腦 一 網路上的芳鄰	 ● 桌面 ● 我的交件 ● 我的变件 ● 我的電腦 ● ACER (C:) ● ACERDATA (D:) ● BE240128 ● BE240128 ● DY ● DYD-RAM 磁碟機 (E:) ● 卸除式磁碟 (F:) ● 卸除式磁碟 (G:) ● 卸除式磁碟 (I:) ● DYD 光碟機 (J:) ● DYD 光碟機 (J:) ● DYD 光碟機 (J:) ● 世俗式磁碟 (I:) ● DYD 光碟機 (J:) ● 如除式磁碟 (I:) ● DYD 光碟機 (J:) ● 新路上的芳鄰 ● My Documents ● My Documents ● My DxDM Places ● 網路上的芳鄰 ● My Private Folder ● 581 option ● 30304 ● AVL-2500 ● BEGX581A ● BIOS基本設定練習模擬器 ● CABLE 				•	開 <u>略()</u> 取消

Add Link Objec	t				? 🔀
查詢(]):	🗀 obj		•	🗢 🗈 💣 🎫	
我最近的文件 反正的文件 反正の 東面 教的文件 教的文件 一 教的文件 一 一 一 一 一 一 一 一 一 一 一 一 一	a2d.o i2ceeprom.o MainUtility.o spi.o f5963c.o TestParten.o touch.o vart.o				
	檔名(N): 檔案類型(T):	i2ceeprom.o All Files(*.*)		•	開啓(<u>0</u>) 取消

BE240128 Projec	t Options	
Geneval	Library Search Path:	
Include Directories	Available Link Objects: libc.a libm.a libobjc.a libprintf_flt.a libprintf_min.a libscanf_flt.a libscanf_min.a	Add Library> libm.a Add Diject D:\BE240128\BE240128\obj\\2ce Add Object Move up Move down Image: Comparison of the second
		確定 取消 説明

		BOLYMIN
BE240128 Project	Options	
Include Directories	Custom Compilation Options [All files] main.c DemoPrg.c demo.c [Linker Options] 	Remove
Memory Settings	External Tools Use WinAVR avr-gcc: C:\WinAVR-20090313\bin\avr-gcc.exe make: C:\WinAVR-20090313\utils\bin\make.exe	Edit Add

BE240128 Projec	et Options		
Include Directories Libraries Libraries Merroay Settings Custom Options	Custom Compilation Options [All files] main.c Demo Bus- demo.c [Linker Options] External Tools Vuse WinAVR avr-gcc: C:\WinAVR-20090313\bin\avr-gcc.exe make: C:\WinAVR-20090313\utils\bin\make.exe	Remove Edit Add	
	確定 取消 取消	說明]

Note: choose [Linker Options]

BE240128 Project	Options 🛛 🗙
Include Directories Libraries Memory Settings	Custom Compilation Options [All files] main.c DemoPrg.c demo.c [Linker Options] Fdit -Wl,-u,vfprintf] Add External Tools V Use WinAVR
	avr-gcc: C:\WinAVR-20090313\bin\avr-gcc.exe make: C:\WinAVR-20090313\utils\bin\make.exe
	確定 取消 說明

Note: key in -WI,-u,vfprintf, and then Add

BE240128 Projec	et Options 🔀
Libuaries Libuaries	Custom Compilation Options [All files] main.c DemoPrg.c demo.c [Linker Options] Edit Edit Edit Mdd
	avr-gcc: C:\WinAVR-20090313\bin\avr-gcc.exe make: C:\WinAVR-20090313\utils\bin\make.exe
	確定 取消 説明

4-4-3 Software Utilities Function Description 4-4-3-1 UART function

Header file : uart.h Object file : uart.o

uartInit Function: Initial UART.

Syntax	void uartInit(
-	uint8_t	byPort,
	uint32 t	uBaudrate.
	uint8 t	bvParity.
	uint8_t	uDatabit.
	uint8_t	uStopbit
	uint8_t	nTxMode
) [.]	
Parameters	byPort UA	RT_PORT0 – 1st Uart port
	UA	RT_PORT1 – 2nd Uart port
	nBaudrat Bau	id rate ex: 9600
	hyParity Par	ity Check 'N' - None 'E' - EVEN 'O' - ODD
	uDatabit dat	h bit $5 \sim 8$
		a dit, $1 \sim 2$.
	n IxMode Irai	nsmission Mode.
	0 or	FALSE – RS232.
	1 or	TRUE - RS485 or RS422.
Return value	None.	

BOLYMIN

uartSetBaudRate Function: to set up baud rate for assigned UART port

Syntax	void uartSetBa	audRate(
-		uint8_t byPort,
		uint32_t uBaudrate
)	·, —
Parameters	byPort	UART_PORT0 – 1st Uart port
	-	UART_PORT1 – 2nd Uart port
	nBaudrate	Baud Rate, ex: 9600.
Return value	None.	

uartSendByte Function: To send 1 byte from assigned UART port

Syntax	void uartSendByte(
	uint8_t byPort,
	uint8_t txData
);
Parameters	byPort UART_PORT0 – 1st Uart port
	UART_PORT1 – 2nd Uart port
	txData byte to be sent.
Return value	None.



uartDisablePort Function: to stop operation of assigned Uart port

Syntax	void uartDisablePort (
-	uint8_t byPort
);
Parameters	byPort UART_PORT0 – 1st Uart port
	UART_PORT1 – 2nd Uart port
Return value	None.

uartSendString Function: to send 1 string from assigned UART port

Syntax	void uartSendString(
	uint8_t byPort
	uint8_t* str
);
Parameters	byPort UART_PORT0 – 1st Uart port
	UART_PORT1 – 2nd Uart port
	str Index of string to be sent, ending with "0".
Return value	None.

uartSendBuffer Function: to send buffer from assigned UART port

Syntax	void uartSendBu	uffer(
-		uint8_t byPort,
		uint8_t* buffer,
		uint16_t nBytes
);	
Parameters	byPort	UART_PORT0 – 1st Uart port
	-	UART_PORT1 – 2nd Uart port
	buffer	index of buffer to be sent
	nBytes	bytes of buffer to be sent
Return value	None.	

uartReceiveByte Function: to read 1 byte data from assigned UART port

Syntax	uint8_t uartReceiveByte(
	uint8_t byPort,		
	uint8_t* rxData		
);		
Parameters	byPort UART_PORT0 – 1st Uart port		
	UART_PORT1 – 2nd Uart port		
	rxData index of data to be received		
Return value	TRUE – rxData is true data		
	FALSE – UART port no data		

uartReceiveBufferIsEmpty Function: to check if there is data in assigned UART port

Syntax	void uartReceiveBufferIsEmpty(
-	uint8_t byPort		
);		
Parameters	byPort UART_PORT0 – 1st Uart port		
	UART_PORT1 – 2nd Uart port		
Return value	TRUE – There is data on assigned UART port		
	FALSE – There is no data on assigned UART port		



uartFlushReceiveBuffer Function: to clear receiving buffer on assigned UART port

Syntax	void uartFlushReceiveBuffer(
-	uint8_t byPort		
);		
Parameters	byPort UART_PORT0 – 1st Uart port		
	UART_PORT1 – 2nd Uart port		
Return value	None.		

uartEnableTx Function: to Enable or Disable UART transmitter. (When UART port is applied on RS485 or RS422, transmitter must be set to Disable, and to Enable transmitter only when sending data.)

Syntax	void uartEnableTx(
);
Parameters	bEnable TRUE – Enable transmitter
	FALSE – Disable transmitter.
Return value	None.


4-4-3-2 I²C function

Header file : i2ceeprom.h object file : i2ceeprom.o

i2clnitial Function: Initial I2C functions. User should call this function before using I2C functions.		
Syntax	void i2clnitial(
);	
Parameters	None.	
Return value	None.	

100 0

i2cReadByte Function: to read 1 byte data from I^2C

Syntax	uint8_t i2cReadByte(
	uint8_t uDevAddr,	
	uint16_t nAddr	
);	
Parameters	uDevAddr I ² C device address. (address of three 24c512 on board are:	
	A2 _{hex} , A4 _{hex} , A6 _{hex} .	
	nAddr address to write in .	
Return value	Data reading from I ² C	

i2cWriteByte Function: to write 1 byte data from I^2C

Syntax	void i2cWriteByte(
-	uint8_t uDevAddr,
	uint16 t nAddr,
	uint8 t byData
);
Parameters	uDevAddr I ² C device address.(address of three 24c512 on board are: A2 _{hex} ,
	A4 _{hex} ,A6 _{hex} .
	nAddr address to write in .
	byData data to write in .
Return value	None.
i2cSetSpeed Function	to set baud rate for I2C
Syntax	void i2cSetSpeed(
-	uint16 t nSpeed,
);
Parameters	uSpeed I ² C baud rate , 0 – 100K, 1 – 250K.
Return value	None.

4-3-3-3 SPI function

Header file : spi.h object file : spi.o



BOLYMIN

spiMaster Function: Initial SPI to master mode.



spiSlave Function: Initial SPI to slave mode.

Syntax	void spiSlave(
	uint8_t mode,
);
Parameters	mode SPI mode.
	0 – CPOL=0, CPHA=0 Sample (Rising) Setup (Falling)
	1 – CPOL=0, CPHA=1 Setup (Rising) Sample (Falling)
	2 – CPOL=1, CPHA=0 Sample (Falling) Setup (Rising)
	3 – CPOL=1, CPHA=1 Setup (Falling) Sample (Rising)
Return value	None.

spiSetBitrate Function: to set SPI baud rate

Syntax	void spiSetBitrate(
		uint8_t spr
)	,
Parameters	spr	SPI baud rate. system OSC is 11.0592MHz.
		0 – OSC / 4.
		1 – OSC / 16.
		2 – OSC / 64.
		3 – OSC / 128.
Return value	None.	

spiSetDataOrder Function: to set SPI order when sending data, LSB first or MSB first.

Syntax	void spiSetDataOrder(
		uint8_t order
);
Parameters	order	LSB first or MSB first. DATA_LSB_FIRST–LSB of the data word will be transmitted first. DATA_MSB_FIRST–MSB of the data word will be transmitted first.(Default)
Return value	None.	

spiSendByte Function: to send 1 byte data from SPI port

Syntax	void spiSendByte(
	uint8_t data		
);		
Parameters	data byte to be sent		
Return value	None.		

spiRecvByte Function: to receive 1 byte data from SPI port

Syntax	uint8_t spiRecvByte(
);	
Parameters	None.	
Return value	To read 1 byte data from SPI	



spiTransferByte Function: to send and read 1 byte data from SPI port

Syntax	uint8_t spiTransferByte(
-	uint8_t data		
);		
Parameters	data byte to be sent		
Return value	1 byte data reading from SPI		

spiTransferWord Function: to send and read 1 word from SPI port

Syntax	Uint16_t spiTransferByte(
	uint16_t data		
);		
Parameters	data word to be sent		
Return value	1 word data reading from SPI		

4-3-3-4 E²PROM function

E²PROM Function is built inside WinAVR, so users need only to include eeprom.h in program to call E²PROM Function.

BOLYMIN

For exmaple :

#include <avr/eeprom.h>

eeprom_write_byte Function: to write 1 byte to MCU E²PROM.

Syntax	void eeprom_w	vrite_byte(uint8_t* address, uint8_t value
Parameters	address value	E ² PROM address to write in, range 0 ~ 0x7FF E ² PROM data to write in
Return value	None.	

eeprom_read_byte Function: This function read one byte from EEPROM.

Syntax	uint8_t eeprom_read_byte (
	uint8_t* address		
);		
Parameters	address E^2 PROM address to read, range 0 ~ 0x7FF		
Return value	Data reading from E ² PROM		

eeprom_write_block Function: to write block data to E2PROM.

Syntax	void eeprom_write_l	byte (
		void* const void*	pointer_ram, pointer_eeprom,
		size_t	n
);	• •	
Parameters	pointer_ram	index of block	data to write in .
	pointer_eeprom	E ² PROM addr	ess to write in, range 0 ~ 0x7FF.
	n	length of E ² PR	OM data to write in
Return value	None.		

eeprom_read_block Function: to read block data from E2PROM

Syntax	uint8_t eeprom_read_block (
		void* const void* size_t	pointer_ram, pointer_eeprom, n
),		
Parameters	pointer_ram	index of block	data to read .
	pointer_eeprom	address of E ² F	PROM to read, rnage 0 ~ 0x7FF.
	n – .	length of E ² PR	OM data to read
Return value	Data block read to .		



Header file : touch.h, a2d.h object file : touch.o, a2d.o

touchInit Function: Initial Touch panel.

Syntax	void touchInit(
);
Parameters	None.
Return value	None.

BOLYMIN

touchGet Function: to read touch data from touch panel

Syntax	uint8_t touchG	et(
		int * pX,
		int * pY
);	
Parameters	pХ	to read X Coordinate from touch data
	pY	to read Y Coordinate from touch data
Return value	TRUE	data of pX and pY is true
	FALSE	data of pX and pY is false

touchDrawCalPoint Function: to draw Calibration cross Coordinate on LCD

Syntax	uint8 t touchDrawCalPoint (
	POINT* ptCal,		
	int n		
);		
Parameters	ptCal Calibration Coordinate.		
	n Calibration Coordinate No.		
Return value	None.		

setCalibrationMatrix Function: to set Calibration calculation matrix

Syntax	void setCalibrationMatrix(
-		POINT * ptDisplay,	
		POINT * ptTouch,	
		int n	
);	
Parameters	ptDisplay	LCD reference Coordinate for calibration.	
	ptTouch	Touch Coordinate for calibration	
	n	Coordinate No. for calibration	
Return value	None.		
getDisplayPoint Funct	ction: to change Touch Coordinate into LCD		
Syntax	void getDispla	yPoint(
		int x,	
		int y,	
		int * pX,	
		int * pY	
);	
Parameters	x Tou	ich X Coordinate.	
	y Tou	ich Y Coordinate.	
	pX LCI	O X Coordinate changed from Touch X Coordinate	
	pY LCI	D Y Coordinate changed from Touch Y Coordinate	
Return value	None.		



4-3-3-6 LCD control function

Header file : t6963c.h object file : t6963c.o

IcdInit Function: Initialize all parameters of LCD display. User should call this function before use functions of LCD display.

Syntax	void IcdInit (
);
Parameters	None.
Return value	None.

IcdDisplayClr Function: Clear screen (include graphic and text layer).

Syntax	IcdDisplayClr(
);
Parameters	None.
Return value	None.

IcdSetCursorPos Function: Set the position of cursor. (UNIT=character=8*8 pixel)

Syntax	void IcdSetCursorPos(
	uint8_t x,		
	uint8_t y		
);		
Parameters	x X coordinate of cursor		
	y Y coordinate of cursor		
Return value	None.		

IcdSwitchDisplay Function: ON/OFF.the display of cursor, text and graphic layer.

Syntax	void IcdSwitchDisplay(
-		uint8_t	display_s	witch
);			
Parameters	display_switch	Display s	witch flag	
		DS_DISF	PLAY_OFF	Turn off all display.
		DS_TEX	T_ON	Turn ON the display of text layer.
		DS_GRA	PHIC_ON	Turn ON the display of graphic layer.
		DS_CUR	SOR_ON	Turn ON the display of cursor.
		DS_BLIN	K_ON	Turn ON cursor blink. This flag is
				available while cursor is ON.
Return value	None.			
Example	// Turn OFF all displa	ay.		
	lcdSwitchDisplay(DS_DISPLAY_OFF);			
	// .Turn ON the displ	ay of text a	ind graphic	layer. Turn OFF cursor.
	lcdSwitchDisplay(DS_TEXT_ON DS_GRAPHIC_ON);			
	// .Turn ON the display of text layer and cursor without blink. Turn OFF graphic			
	layer.			
	IcdSwitchDisplay(DS_TEXT_ON DS_CURSOR_ON);			
	// .Turn ON the display of text layer, graphic layer and cursor with blink.			
	IcdSwitchDisplay(DS_TEXT_ON DS_GRAPHIC_ON DS_CURSOR_ON			
	DS BLINK ON);			



IcdSetDispMode Function: Set the display mode between graphic layer and text layer.

Syntax	void IcdSetDispMode (
-	uint8_t new_mode			
);			
Parameters	New_mode New display mode between graphic layer and text layer. Value : DM_OR_MODE, DM_XOR_MODE, DM_AND_MODE			
	Content on Graphic layer Content on Text layer			
	DM_OR_MODE DM_XOR_MODE DM_AND_MODE			
Return value	None			

IcdDraw Function: Draw input binary picture on specified area of graphic area.

Syntax	void lcdDraw (
_	uint8	_t x_start,
	uint8	t y_start,
	uint8	t x end,
	uint8	t y end,
	uint8	t * pic data.
	uint8	t mode
);	
Parameters	x_start	X coordinate of the top-left point of input picture. (UNIT=pixel)
	y_start	Y coordinate of the top-left point of input picture. (UNIT=pixel)
	x_end	X coordinate of the bottom-right point of input picture. (UNIT=pixel)
	y end	Y coordinate of the bottom-right point of input picture. (UNIT=pixel)
	pic data	Bit map data will be drawn. Input 0 will reverse pixels of specified
	· _	area.
	mode	DRAW NORMAL: Draw the picture normally.
		DRAW REVERSE : Reverse the picture and then draw the
		picture.
Return value	None.	

IcdFillByte Function: Fill input byte value on specified area of graphic layer.

Syntax	void IcdFillByte	(
		uint8_t x_start,
		uint8_t y_start,
		uint8_t x_end,
		uint8_t y_end,
		uint8_t data,
		uint8_t mode
);	
Parameters	x_start	X coordinate of the top-left point of specified area. (UNIT=pixel)
	y_start	Y coordinate of the top-left point of specified area. (UNIT=pixel)
	x_end	X coordinate of the bottom-right point of specified area.
		(UNIT=pixel)
	y_end	Y coordinate of the bottom-right point of specified area.
		(UNIT=pixel)
	pic_data	Byte value will be filled.
	Mode	DRAW_NORMAL: Fill input value normally.
		DRAW_REVERSE : Reverse the input value and then fill it on the
		specified area
Return value	None.	

IcdPrintString Function: Print input string to specified location of text layer.

Syntax	void IcdPrintString (
		uint8_t x_start,
		uint8_t y_start,
		char * string,
		uint8_t str_count
);	
Parameters	x_start	X coordinate of start location that input string will be printed.
		(UNIT= character=8*8 pixel)
	y_start	Y coordinate of start location that input string will be printed.
		(UNIT= character=8*8 pixel)
	string	string will be printed to LCD
	str_count	character count of input string
Return value	None.	

IcdDrawBit Function: ON/OFF the pixel on specified location of graphic layer.

Syntax	void IcdDrawBit (
	uint8_t x,
	uint8_t y,
	char bit_value
);
Parameters	x X coordinate of the location will be drawn. (UNIT=pixel)
	 Y coordinate of the location will be drawn. (UNIT=pixel)
	bit_value 1 : ON the pixel
	0 : OFF the pixel
Return value	None.

IcdDrawRect Function: Draw rectangle by single line on graphic layer.

Syntax	void IcdDrawR	lect (
-		uint8_t x_start,
		uint8_t y_start,
		uint8_t x_end,
		uint8_t y_end,
)	;
Parameters	x_start	X coordinate of the top-left point of rectangle. (UNIT=pixel)
	y_start	Y coordinate of the top-left point of rectangle. (UNIT=pixel)
	x_end	X coordinate of the bottom-right point of rectangle. (UNIT=pixel)
	y_end	Y coordinate of the bottom-right point of rectangle. (UNIT=pixel)
Return value	None	

4-3-3-7 Backlight PWM control function

Header file : bklight_pwm.h object file : bklight_pwm.o

Note:Backlight PWM control used TIMER0 and INTERRUPT0

bkIPWM_Init Function: Initialize all parameters of backlight PWM control function. User should call this function before use backlight PWM control functions.

BOLYMIN

Syntax	void bkIPWM_Init (
);
Parameters	None.
Return value	None.

bklSetBrightness Function: Set current brightness value of backlight.

Syntax	void bklSetBrightness (
		uint8_t brightness	
););	
Parameters	Brightness	New brightness value	
	-	0 – OFF backlight	
		1~100 – Control the brightness of backlight	
Return value	None.		

bklGetBrightness Function: Get current brightness value of backlight

Syntax	uint8_t bklGetBrightness (
);
Parameters	None.
Return value	Current brightness value of backlight. (0 ~ 100)



Appendix A: LCD Controller Specification

Appendix B: EEPROM Specification

Appendix C: ATMEL ATmega644p MCU Specification

Please download this specification from following ATMEL link: http://www.atmel.com/dyn/resources/prod_documents/doc8011.pdf



<END of BEGV641A User Manual>