Complete ARM Solutions
Design, Development and Manufacturing
Expertise on Embedded Linux, Android, WindowsCE

Tiny210/Smart210/Mini210S User's Manual

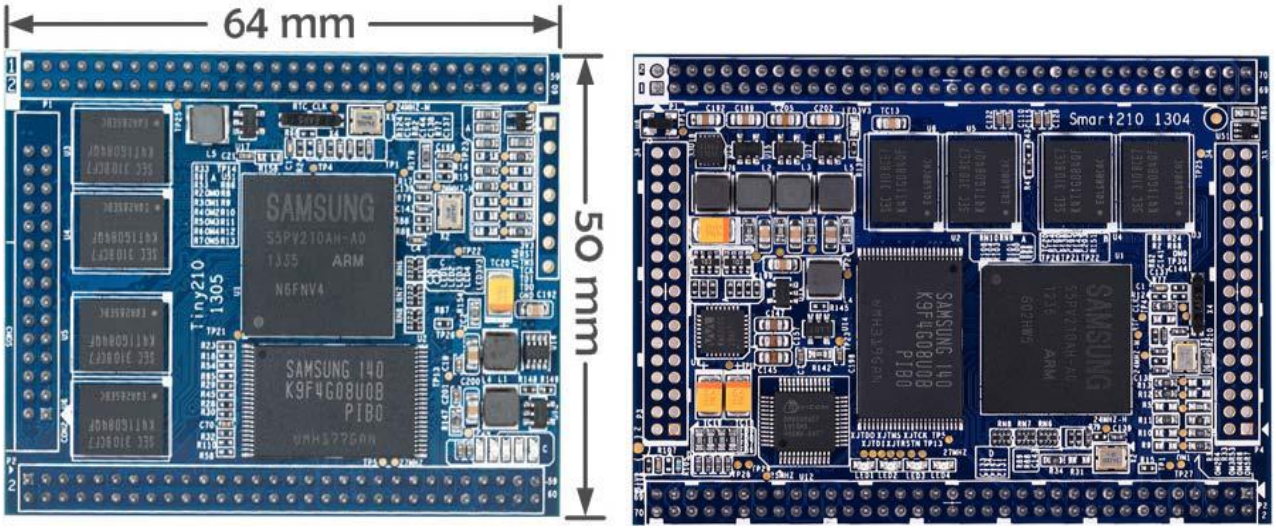
REVISION	ORIGINATOR	SCR	REV DATE
0.1.0	Guangzhou FriendlyARM Co., Ltd		August 11st, 2014
Guangzhou FriendlyARM Computer Tech Co., Ltd Confidential: This document and information contained in it shall not be reproduced by, used by, or disclosed to others except as expressly authorized in writing by Guangzhou FriendlyARM Co., Ltd.			**Guangzhou FriendlyARM Computer Tech Co., Ltd Guangzhou, China**

Address: Room 1705,Block A1, Longyuan Plaza, Longkouxi Road, Guangzhou, China, 510640 Website: http://www.arm9.net
Sales: +86-20-85201025 Tech Support: +86-13719442657 Fax: +86-20-85261505
Email for Business and Cooperation: capbily@163.com Email for Tech Support: dev_friendlyarm@163.com

COPYRIGHT STATEMENT

Index

1 Introduction to Tiny210/Smart210



Tiny210 CPU Board vs Smart210 CPU Board

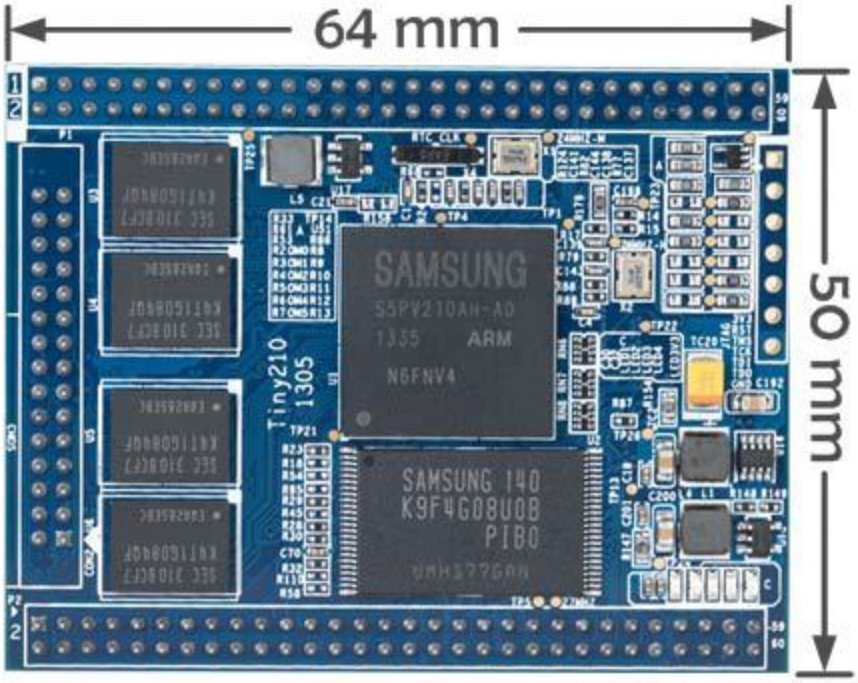
The Tiny210 CPU board and Smart210 CPU board are both Cortex-A8 embedded processing board that uses the Samsung S5PV210 System On Chip (SOC). Its maximum frequency is up to 1GHz. The S5PV210 integrates the PowerVR SGX540 graphic engine with hardware support for 3D and can drive video playing on screens up to 1080P.

1.1 Tiny210 CPU Board

The Tiny210 CPU board has 2.0 mm spacing double row pitch headers which connect the Tiny210 to a carrier board and extend most of the CPU's pins and work with both the Tiny210 and Tiny6410 carrier boards. The standard version integrates 512M DDR2 RAM and 512M Flash (1G Flash optional) and can run Android, Linux and WinCE6. These features make it easily and widely used in MID development,

Android notepads, auto electronic devices, industrial applications, GPS systems and multimedia systems.

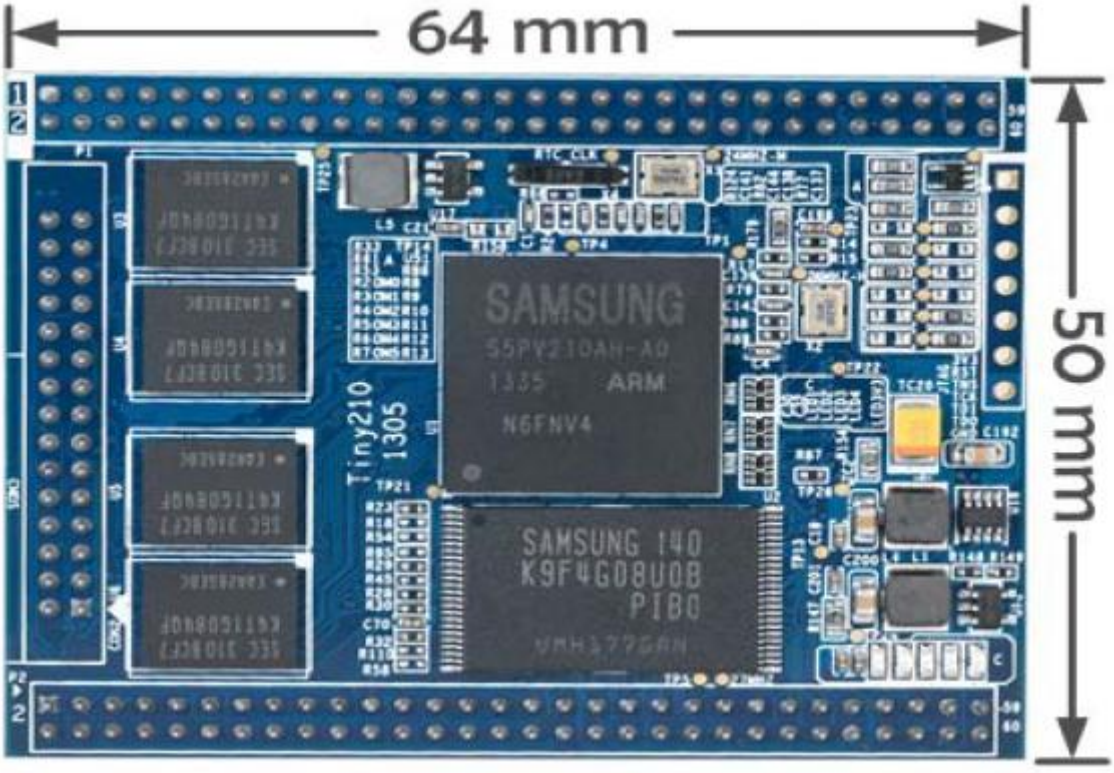
1.1.1 Tiny210 CPU Board Hardware Feature



CPU	● Samsung S5PV210, based on CortexTM-A8, 1GHz ● Integrated PowerVR SGX540 graphic engine ● Elegent 2D/3D graphic accelaration ● Up to 1080p@30fps hard decoded video playing, support MPEG4, H.263, H.264 etc ● Up to 1080p@30fps hard decoded (Mpeg-2/VC1) video input
DDR2 RAM	● 512M ● 32bit data bus, single channels ● 200 M Hz
FLASH	● SLC NAND Flash: 512M, optional 1GB

Connector	● 2 x 60 pin 2.0 mm pitch header ● 1 x 30 pin 2.0 mm pitch header
On Board Hardware Resource	● 4 x LED (Green) ● 1 x Power LED (Red)
Power	● 4.7V to 5.6V (support sleep mode)
PCB Dimension	● Eight layered board ● Dimension: 64 x 50 x 11(mm)

1.1.2 Tiny210 CPU Board Pin Spec



P1	Spec	P1	Spec
P1.1	VDD_5V	P1.2	DGND
P1.3	XvVD23	P1.4	XvVD22
P1.5	XvVD21	P1.6	XvVD20
P1.7	XvVD19	P1.8	XvVD18
P1.9	XvVD15	P1.10	XvVD14
P1.11	XvVD13	P1.12	XvVD12
P1.13	XvVD11	P1.14	XvVD10
P1.15	XvVD7	P1.16	XvVD6
P1.17	XvVD5	P1.18	XvVD4
P1.19	XvVD3	P1.20	XvVD2
P1.21	XvVDEN	P1.22	XEINT10
P1.23	XvVSYNC	P1.24	XvHSYNC
P1.25	XvVCLK	P1.26	XpwmTOUT1
P1.27	XuoVBUS	P1.28	XuoDRVVBUS
P1.29	XuoID	P1.30	XEINT8
P1.31	XuoDM	P1.32	XuhDM
P1.33	XuoDP	P1.34	XuhDP
P1.35	XadcAIN9_XP	P1.36	XadcAIN8_XM
P1.37	XadcAIN7_YP	P1.38	XadcAIN6_YM
P1.39	XadcAIN0	P1.40	XadcAIN1
P1.41	WIFI0_PD_GPIO	P1.42	WIFI0_RESET_GPIO
P1.43	Xmmc2CLK/SPI_CLK2	P1.44	Xmmc2CMD/SPI_CSn2
P1.45	Xmmc2CDn/SPI_MISO2	P1.46	XEINT11
P1.47	Xmmc2DATA0/SPI_MOSI2	P1.48	Xmmc2DATA1
P1.49	Xmmc2DATA2	P1.50	Xmmc2DATA3
P1.51	XdacOUT	P1.52	XpwmTOUT0
P1.53	XEINT16/KP_COL0	P1.54	XEINT17/KP_COL1
P1.55	XEINT18/KP_COL2	P1.56	XEINT19/KP_COL3
P1.57	XEINT24/KP_ROW0	P1.58	XEINT25/KP_ROW1
P1.59	XEINT26/KP_ROW2	P1.60	XEINT27/KP_ROW3
P2	**Spec**	**P2**	**Spec**
P2.1	OM1	P2.2	GND
P2.3	M_nRESET	P2.4	OVDD_RTC_AP
P2.5	XuCTSn1	P2.6	XuRTSn1
P2.7	XuTXD0	P2.8	XuRXD0
P2.9	XuTXD1	P2.10	XuRXD1

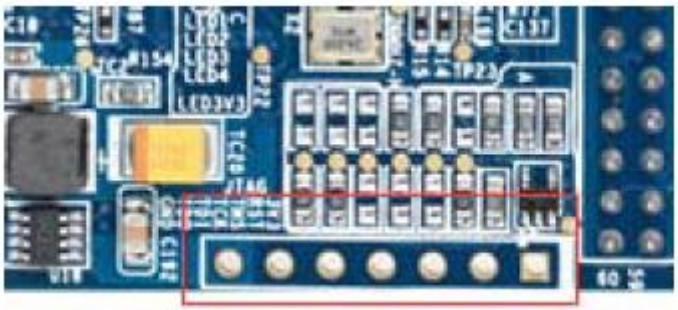
P2.11	XuTXD2/UART_AUDIO_TXD	P2.12	XuRXD2/UART_AUDIO_RXD
P2.13	XuTXD3/RTSn2/UART_AUDIO_RTSn	P2.14	XuRXD3/CTSn2/UART_AUDIO_CTSn
P2.15	XspiMISO1	P2.16	XspiMOSI1
P2.17	XspiCLK1	P2.18	XspiCS1
P2.19	Xi2cSCL0	P2.20	Xi2cSDA0
P2.21	XmmcCLK0	P2.22	XmmcCMD0
P2.23	Xmmc0CDn	P2.24	XEINT6_SD0_nWP
P2.25	Xmmc0DATA0	P2.26	Xmmc0DATA1
P2.27	Xmmc0DATA2	P2.28	Xmmc0DATA3
P2.29	Audio_Xi2sSCLK0	P2.30	Audio_Xi2sCDCLK0
P2.31	Audio_Xi2sLRCK0	P2.32	Audio_Xi2sSDO0_0
P2.33	Audio_Xi2sSDI0	P2.34	XEINT9
P2.35	Xm0ADDR0	P2.36	Xm0ADDR1
P2.37	Xm0ADDR2	P2.38	Xm0ADDR15
P2.39	Xm0CSn1	P2.40	XEINT7
P2.41	Xm0WAITn	P2.42	XnRSTOUT
P2.43	Xm0WEn	P2.44	Xm0OEn
P2.45	Xm0DATA0	P2.46	Xm0DATA1
P2.47	Xm0DATA2	P2.48	Xm0DATA3
P2.49	Xm0DATA4	P2.50	Xm0DATA5
P2.51	Xm0DATA6	P2.52	Xm0DATA7
P2.53	Xm0DATA8	P2.54	Xm0DATA9
P2.55	Xm0DATA10	P2.56	Xm0DATA11
P2.57	Xm0DATA12	P2.58	Xm0DATA13
P2.59	Xm0DATA14	P2.60	Xm0DATA15
CON1	**Spec**	**CON1**	**Spec**
CON1.1	VDD_5V	CON1.2	VDD_5V
CON1.3	XmmcCLK1	CON1.4	VDD_SYS_3.3V
CON1.5	XmmcCMD1	CON1.6	VDD_SYS_1.8V
CON1.7	XmmcCDn1	CON1.8	DGND
CON1.9	Xmmc1DATA0	CON1.10	CAM_B_D0
CON1.11	Xmmc1DATA1	CON1.12	CAM_B_D1
CON1.13	Xmmc1DATA2	CON1.14	CAM_B_D2
CON1.15	Xmmc1DATA3	CON1.16	CAM_B_D3
CON1.17	Xmmc3CLK	CON1.18	CAM_B_D4
CON1.19	Xmmc3CMD	CON1.20	CAM_B_D5
CON1.21	Xmmc3CDn	CON1.22	CAM_B_D6
CON1.23	Xmmc3DATA0/Xmmc2DATA4	CON1.24	CAM_B_D7
CON1.25	Xmmc3DATA1/Xmmc2DATA5	CON1.26	CAM_B_PCLK

CON1.27	Xmmc3DATA2/Xmmc2DATA6	CON1.28	CAM_B_VSYNC
CON1.29	Xmmc3DATA3/Xmmc2DATA7	CON1.30	CAM_B_HREF
CON1.31	Xi2sCLK1/PCM_SCLK1/AC97_BITCLK	CON1.32	CAM_B_FIELD
CON1.33	Xi2sCDCLK1/PCM_EXTCLK1/AC97_RESETn	CON1.34	CAM_B_CLKOUT
CON1.35	Xi2sLRCK1/PCM_FSYNC1/AC97_SYNC	CON1.36	CAMERA_B_GPIO0
CON1.37	Xi2sSDI1/PCM_SIN1/AC97_SDI	CON1.38	CAMERA_B_GPIO1
CON1.39	Xi2sSDO1/PCM_SOUT1/AC97_SDO	CON1.40	CAMERA_B_GPIO2
CON1.41	XpcmSCLK0/SPDIF_OUT0/Xi2sSCLK2	CON1.42	CAM_B_RESET
CON1.43	XpcmEXTCLK0/SPDIF_EXTCLK/Xi2sCDCLK2	CON1.44	Xi2cSCL1
CON1.45	XpcmFSYNC0/LCD_FRM/Xi2sLRCK2	CON1.46	Xi2cSDA1
CON1.47	XpcmSIN0/Xi2sSDI2	CON1.48	XpwmTOUT2
CON1.49	XpcmSOUT0/Xi2sSDO2	CON1.50	XpwmTOUT3/PWM_MIE
CON1.51	GND		
CON2	**Spec**	**CON2**	**Spec**
CON2.1	Xi2cSDA0	CON2.2	Xi2cSCL0
CON2.3	CAMERA_A_GPIO2	CON2.4	CAM_A_RESET
CON2.5	XciCLKenb	CON2.6	XciHREF
CON2.7	XciVSYNC	CON2.8	XciPCLK
CON2.9	XciYDATA7	CON2.10	XciYDATA6
CON2.11	XciYDATA5	CON2.12	XciYDATA4
CON2.13	XciYDATA3	CON2.14	XciYDATA2
CON2.15	XciYDATA1	CON2.16	XciYDATA0
CON2.17	VDD_SYS_3.3V	CON2.18	VDD_CAM_2.8V
CON2.19	VDD_CAM_1.8V	CON2.20	DGND
CON2.21	Xi2cSDA2/IEM_SCLK	CON2.22	XEINT15
CON2.23	Xi2cSCL2/IEM_SPWI	CON2.24	XEINT14
CON2.25	XvVD0	CON2.26	XvVD1
CON2.27	XvVD8	CON2.28	XvVD9
CON2.29	XvVD16	CON2.30	XvVD17

1.1.3 Tiny210 CPU Board Interface and Port

1.1.3.1 JTAG

The Tiny210 CPU board has test points for JTAG as follows: GND, TDO, TDI, TCK, TMS, TSRT and 3.3V:



1.1.3.2 LED

LED is a commonly used status indication device. The Tiny210 has four programmable LEDs which are directly connected to GPIO and are on at a low level voltage.

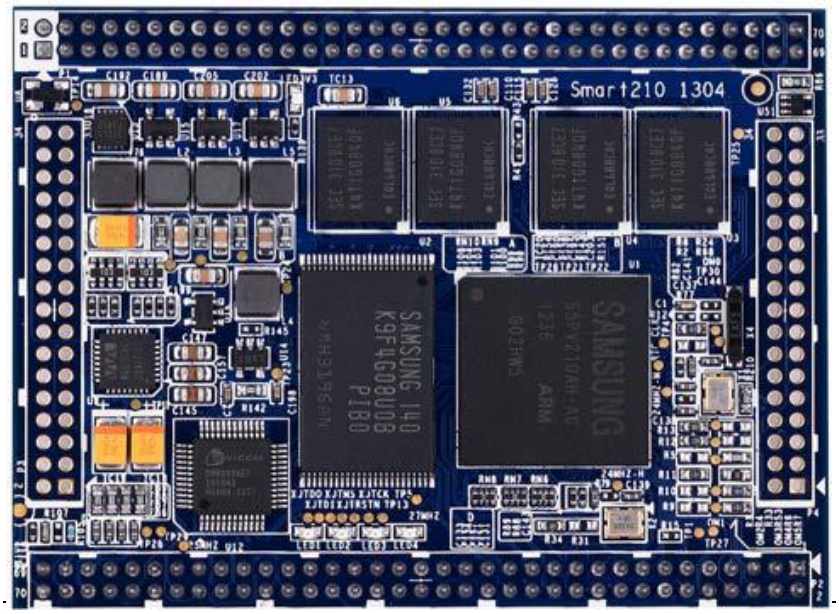
	LED1	LED2	LED3	LED4
GPIO Pins	GPJ_0	GPJ_1	GPJ_2	GPJ_3

1.2 Smart210 CPU Board

The Smart210 CPU board has 2.0 mm spacing double row pitch headers (P1, P2, P3 and P4) which connect the Smart210 to a carrier board and extend most of the CPU's pins. Its size (74 x 55 mm) is bigger than the Tiny210 CPU board. P1 and P2 are standard configurations, and P3 and P4 are left for users. The standard version integrates 512M DDR2 RAM and 512M Flash (1G Flash optional) and can run Android, Linux and WinCE6. These features make it easily and widely used in MID development, Android notepads, auto electronic devices, industrial applications, GPS systems and

multimedia systems.

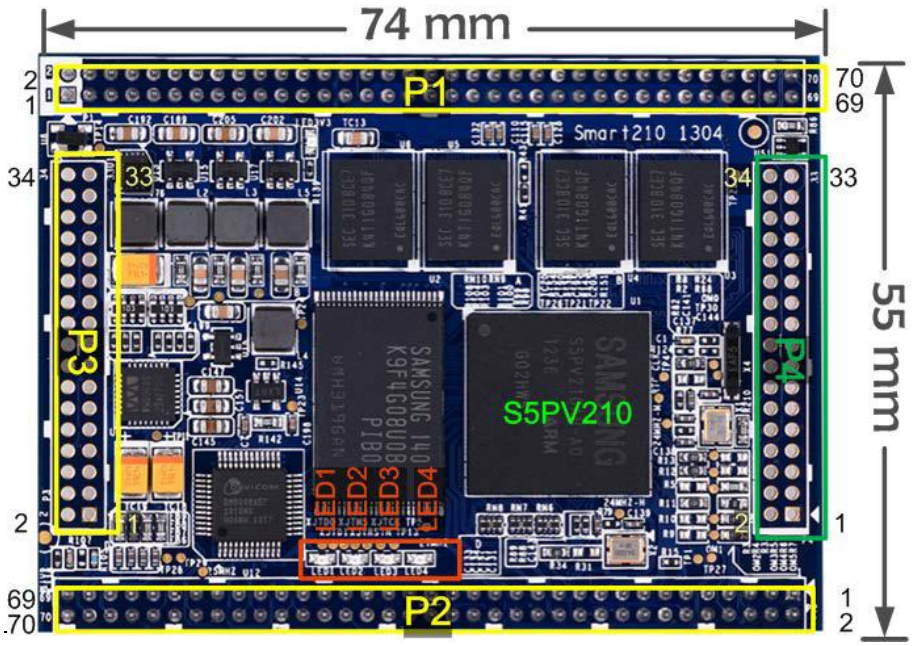
1.2.1 Smart210 CPU Board Hardware Feature



CPU	● Samsung S5PV210, based on CortexTM-A8, 1GHz ● Integrated PowerVR SGX540 graphic engine ● Elegent 2D/3D graphic accelaration ● Up to 1080p@30fps hard decoded video playing, support MPEG4, H.263, H.264 etc ● Up to 1080p@30fps hard decoded (Mpeg-2/VC1) video input
DDR2 RAM	● 512M ● 32bit data bus, single channels ● 200 M Hz
FLASH	● MLC NAND Flash: 512M SLC Flash optional 1G SLC Flash
Connector	● 2 x 70 pin 2.0 mm DIP connector ● 2 x 34 pin 2.0 mm DIP connector

On Board Hardware Resource	● 4 x LED (Green) ● 1 x Power LED (Red) ● 1 x Audio(WM8960) ● 1 x Ethernet(DM9000)
Power	● 4.7V to 5.6V (support sleep mode)
PCB Dimension	● Six layered board ● Dimension: 74 x 55 x 11(mm)

1.2.2 Smart210 CPU Board Pin Spec



P1	Spec	P1	Spec
P1.1	5VDC Input	P1.2	GND
P1.3	VDD_RTC	P1.4	WIFI0_RESET_GPIO
P1.5	Manual Reset Input(Active Low)	P1.6	WIFI0_PD_GPIO
P1.7	XmmcCMD0	P1.8	Xmmc2CMD
P1.9	XmmcCLK0	P1.10	Xmmc2CLK

P1.11	XmmcODATA0	P1.12	Xmmc2DATA0
P1.13	XmmcODATA1	P1.14	Xmmc2DATA1
P1.15	XmmcODATA2	P1.16	Xmmc2DATA2
P1.17	XmmcODATA3	P1.18	Xmmc2DATA3
P1.19	XmmcOCDn	P1.20	Xmmc2CDn
P1.21	XEINT16/KP_COL0	P1.22	XEINT24/KP_ROW0
P1.23	XEINT17/KP_COL1	P1.24	XEINT25/KP_ROW1
P1.25	XEINT18/KP_COL2	P1.26	XEINT26/KP_ROW2
P1.27	XEINT19/KP_COL3	P1.28	XEINT27/KP_ROW3
P1.29	XEINT10	P1.30	XEINT14
P1.31	XEINT11	P1.32	XEINT15
P1.33	XpwmTOUT0	P1.34	Xi2cSCL0
P1.35	XpwmTOUT1	P1.36	Xi2cSDA0
P1.37	XuhDM	P1.38	Xi2cSCL2
P1.39	XuhDP	P1.40	Xi2cSDA2
P1.41	VD0(Blue LSB)	P1.42	VD1
P1.43	VD2	P1.44	VD3
P1.45	VD4	P1.46	VD5
P1.47	VD6	P1.48	VD7(Blue MSB)
P1.49	VD8(Green LSB)	P1.50	VD9
P1.51	VD10	P1.52	VD11
P1.53	VD12	P1.54	VD13
P1.55	VD14	P1.56	VD15(Green MSB)
P1.57	VD16(Red LSB)	P1.58	VD17
P1.59	VD18	P1.60	VD19
P1.61	VD20	P1.62	VD21
P1.63	VD22	P1.64	VD23(Red MSB)
P1.65	Vertical Synchronous Signal	P1.66	Horizontal Synchronous Signal
P1.67	RGB Data Sampling Clock	P1.68	RGB Data Enable
P1.69	GND	P1.70	Boot Mode Select
P2	**Spec**	**P2**	**Spec**
P2.1	XuTXD0	P2.2	XuRXD0
P2.3	XuTXD1	P2.4	XuRXD1
P2.5	XuTXD2	P2.6	XuRXD2
P2.7	XuTXD3	P2.8	XuRXD3
P2.9	XuCTSn1	P2.10	XuRTSn1
P2.11	CAM_A_D0	P2.12	CAM_A_D1
P2.13	CAM_A_D2	P2.14	CAM_A_D3
P2.15	CAM_A_D4	P2.16	CAM_A_D5

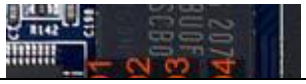
P2.17	CAM_A_D6	P2.18	CAM_A_D7
P2.19	CAM_A_PCLK	P2.20	CAM_A_VSYNC
P2.21	CAM_A_HREF	P2.22	CAM_A_FIELD
P2.23	CAM_A_CLKenb	P2.24	XhdmiTX1P
P2.25	XhdmiTX0P	P2.26	XhdmiTX1N
P2.27	XhdmiTX0N	P2.28	XhdmiTXCP
P2.29	XhdmiTX2P	P2.30	XhdmiTXCN
P2.31	XhdmiTX2N	P2.32	XEINT13/HDMI_HPD
P2.33	Xi2cSDA1	P2.34	Xi2cSCL1
P2.35	XuoID	P2.36	XspiCS0
P2.37	XuoDM	P2.38	XspiMISO0
P2.39	XuoDP	P2.40	XspiMOSI0
P2.41	XuoVBUS	P2.42	XspiCLK0
P2.43	XmipiMDPCLK	P2.44	XuoDRVVBUS
P2.45	XmipiMDNCLK	P2.46	GND
P2.47	XmipiMDP0	P2.48	Ethernet PHY RX-
P2.49	XmipiMDN0	P2.50	Ethernet PHY RX+
P2.51	XmipiMDP1	P2.52	Ethernet PHY TD-
P2.53	XmipiMDN1	P2.54	Ethernet PHY TD+
P2.55	XmipiMDP2	P2.56	Ethernet Link / Active LED
P2.57	XmipiMDN2	P2.58	Ethernet Speed LED
P2.59	XmipiMDP3	P2.60	GND
P2.61	XmipiMDN3	P2.62	Head Phone Detect
P2.63	Head Phone Output R	P2.64	Head Phone Output L
P2.65	SPK_OUT_LP	P2.66	SPK_OUT_LN
P2.67	SPK_OUT_RP	P2.68	SPK_OUT_RN
P2.69	Mic Input P	P2.70	Mic Input N
P3	**Spec**	**P3**	**Spec**
P3.1	5VDC Input	P3.2	GND
P3.3	XEINT0	P3.4	XEINT1
P3.5	XEINT2	P3.6	XEINT3
P3.7	XEINT4	P3.8	XEINT5
P3.9	XEINT6	P3.10	XEINT8
P3.11	Xm0ADDR0	P3.12	Xm0ADDR1
P3.13	Xm0ADDR2	P3.14	NC
P3.15	Xm0CSn0	P3.16	XnRSTOUT(System Reset Output)
P3.17	Xm0WEn	P3.18	Xm0OEn
P3.19	Xm0DATA0	P3.20	Xm0DATA1
P3.21	Xm0DATA2	P3.22	Xm0DATA3

P3.23	XmODATA4	P3.24	XmODATA5
P3.25	XmODATA6	P3.26	XmODATA7
P3.27	XmODATA8	P3.28	XmODATA9
P3.29	XmODATA10	P3.30	XmODATA11
P3.31	XmODATA12	P3.32	XmODATA13
P3.33	XmODATA14	P3.34	XmODATA15
P4	**Spec**	**P4**	**Spec**
P4.1	CAM_B_D0	P4.2	XEINT20/KP_COL4
P4.3	CAM_B_D1	P4.4	XEINT21/KP_COL5
P4.5	CAM_B_D2	P4.6	XEINT22/KP_COL6
P4.7	CAM_B_D3	P4.8	XEINT23/KP_COL7
P4.9	CAM_B_D4	P4.10	XEINT28/KP_ROW4
P4.11	CAM_B_D5	P4.12	XEINT29/KP_ROW5
P4.13	CAM_B_D6	P4.14	XEINT30/KP_ROW6
P4.15	CAM_B_D7	P4.16	XEINT31/KP_ROW7
P4.17	CAM_B_PCLK	P4.18	XEINT12/HDMI_CEC
P4.19	CAM_B_VSYNC	P4.20	GND
P4.21	CAM_B_HREF	P4.22	XadcAIN0
P4.23	CAM_B_FIELD	P4.24	XadcAIN1
P4.25	CAM_B_CLKOUT	P4.26	XadcAIN6_YM
P4.27	XspiMOSI1	P4.28	XadcAIN7_YP
P4.29	XspiMISO1	P4.30	XadcAIN8_XM
P4.31	XspiCS1	P4.32	XadcAIN9_XP
P4.33	XspiCLK1	P4.34	GND

1.2.3 Smart210 CPU Board Interface and Port

1.2.3.1 LED

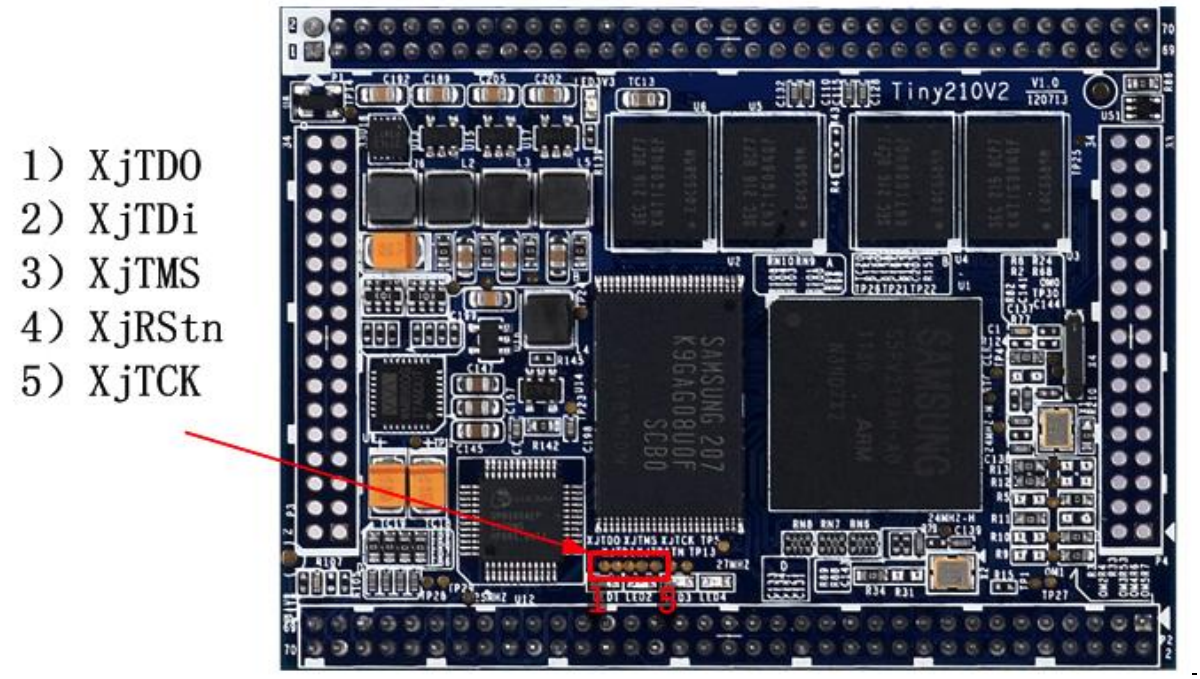
LED is a commonly used status indication device. The Smart210 has four programmable LEDs which are directly connected to GPIO and are on at a low level voltage.

		LED1	LED2	LED3	LED4

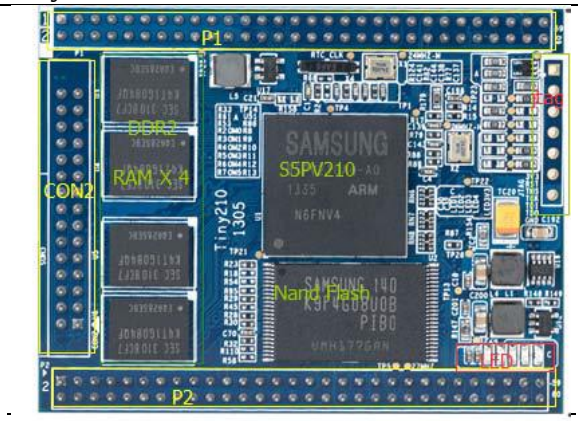
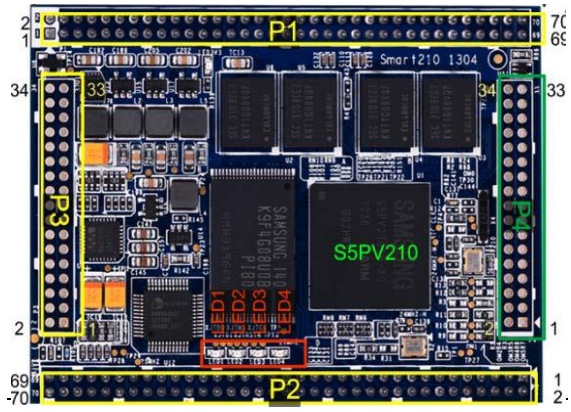
	GPIO Pins	GPJ_0	GPJ_1	GPJ_2	GPJ_3

1.2.3.2 Jtag

The Smart210 CPU board has five Jtag test points from which users can extend their applications.

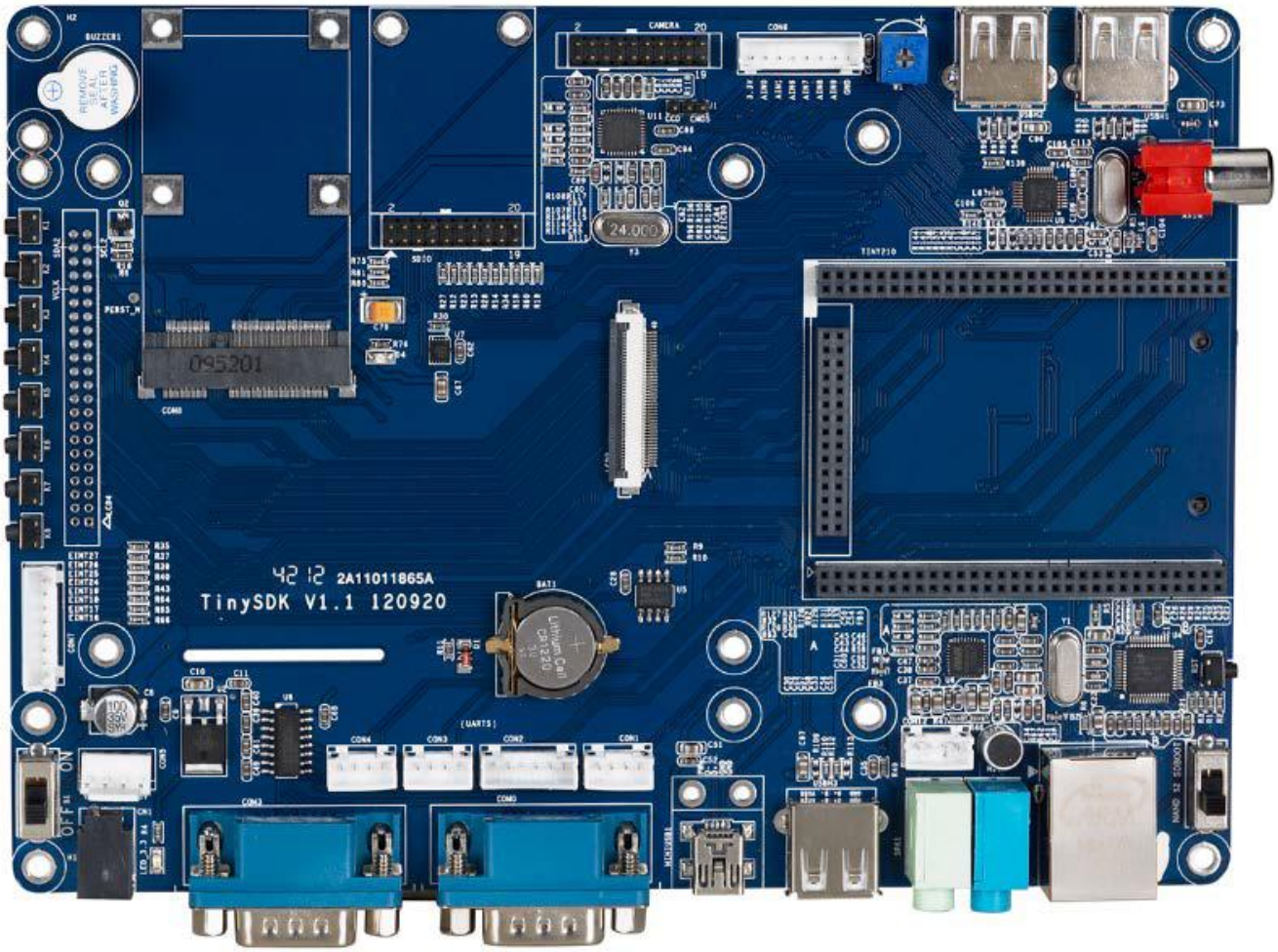


1.3 Comparison of Smart210 and Tiny210 CPU Board

	Tiny210 CPU Board	Smart210 CPU Board
		

CPU	● Samsung S5PV210, based on CortexTM-A8, 1GHz ● Integrated PowerVR SGX540 graphic engine ● Elegent 2D/3D graphic accelaration ● Up to 1080p@30fps hard decoded video playing, support MPEG4, H.263, H.264 etc ● Up to 1080p@30fps hard decoded (Mpeg-2/VC1) video input	
DDR2 RAM	● 512M ● 32bit data bus, single channels ● 200 M Hz	
Flash	● 32bit data bus, sin MLC NAND Flash: 512M SLC Flash optional 1G SLC Flash	
Connector	● 2 x 60 pin 2.0 mm pitch header ● 1 x 30 pin 2.0 mm pitch header	● 2 x 70 pin 2.0 mm DIP connector ● 2 x 34 pin 2.0 mm DIP connector
On Board Hardware Resource	● 4 x LED (Green) ● 1 x Power LED (Red) ● 1 x J-TAG	● 4 x LED (Green) ● 1 x Power LED (Red) ● 1 x Audio(WM8960) ● 1 x Ethernet(DM9000) ● 1 x J-TAG Test Points
Power	● 4.7V to 5.6V (support sleep mode)	
PCB Dimension	● Eight Layer Board ● 64 x 50 x 11 mm	● Six Layer Board ● 74 x 55 x 11 mm

1.4 TinySDK1312B Carrier Board



1.4.1 TinySDK 1312B Carrier Board Hardware Feature

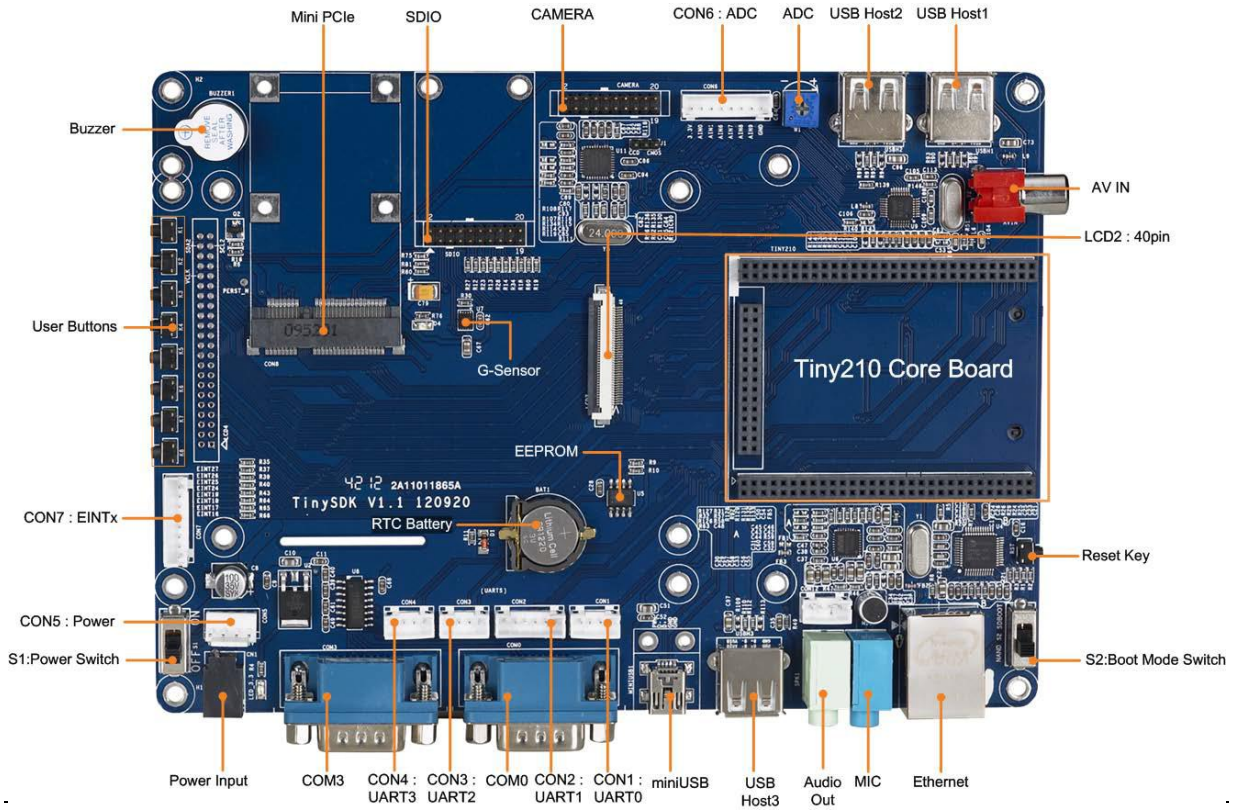
The TinySDK carrier board is a two-layer circuit board that demonstrates user-friendly reference designs with all the common interfaces. Components and interfaces (except the SD socket) are all located on one side for easy use.

LCD	● LCD1 interface (on the reverse): 45Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching ● LCD2 interface (on the obverse): 40Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching

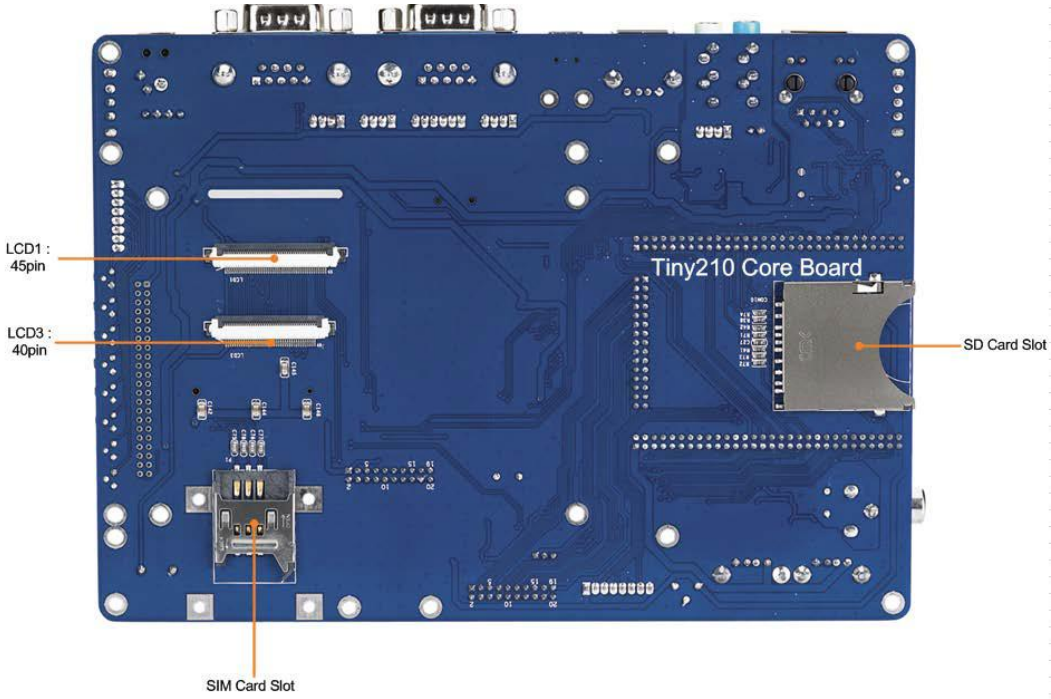
	● LCD3 interface (on the reverse): 40Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching ● LCD4 interface (reserved): 40Pin, 2mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching ● miniHDMI high definition interface (Type C) ● LCDs supported from 3.5" to 12.1" up to maximum resolution of 1024x768
Network	● 1 x 10/100M Ethernet interface(RJ45) using DM9000AEP
Standard Configuration	● 1 x miniPCIe interface ● 2 x DB9 RS232 serial port ● 1 x miniUSB Slave-OTG 2.0 which can be extened via a 2.0mm socket ● 1 x 3.5mm stereotype audio output ● 1 x integrated microphone ● 3 x USB Host 2.0 ● 1 x standard SD card socket ● 1 x 5V power input (DC-23B)
On Board Hardware Resource	● 1 x I2C-EEPROM (256byte) for I2C bus test ● 8 x Interrupt Style Push Button on module ● 1 x test pot for ADC testing ● 1 x PWM buzzer ● 1 x backup battery for on board real time clock ● 1 x G-sensor chip
External Resource	● 4 x TTL ● 6 x channels AD Input ● 1 x SDIO ● 1 x CMOS camera interface ● 1 x AV IN for CCD camera

PCB Dimension	● Two Layered Board ● Dimension: 180 x 130(mm)
Software	● Superboot ● Android 4.0 ● Android 2.3 ● Linux-3.0.8 + Qt4.8.5/Qtopia2/Qtopia4 ● Windows CE6.0

1.4.2 TinySDK 1312B Carrier Board Layout



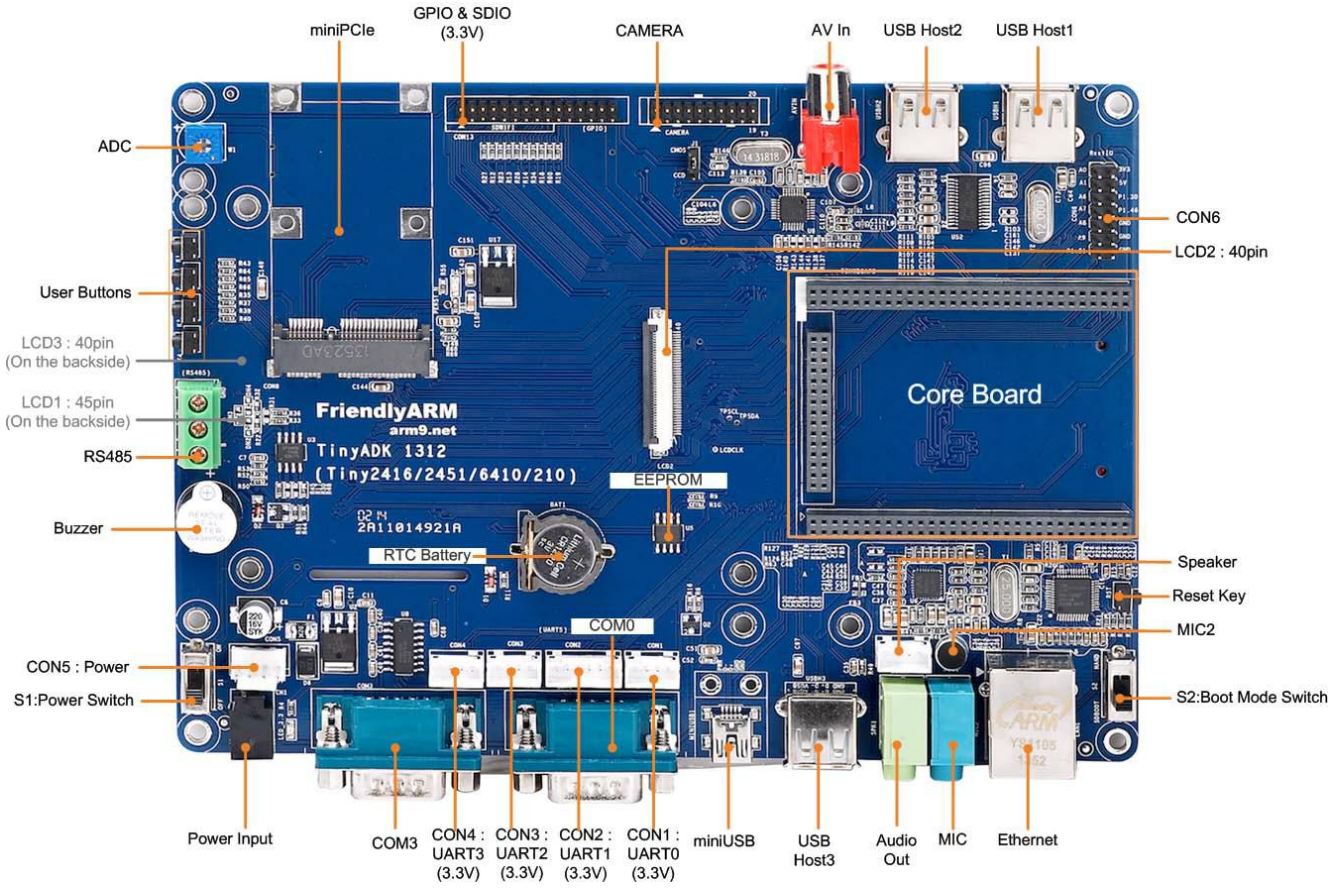
Obverse



Reverse

1.5 TinyADK1312B Carrier Board

The TinyADK is a common carrier board that works for the Tiny2416 CPU board, Tiny2451 CPU board, Tiny6410 CPU board and Tiny210 CPU board.



1.5.1 TinyADK 1312B Carrier Board Hardware Feature

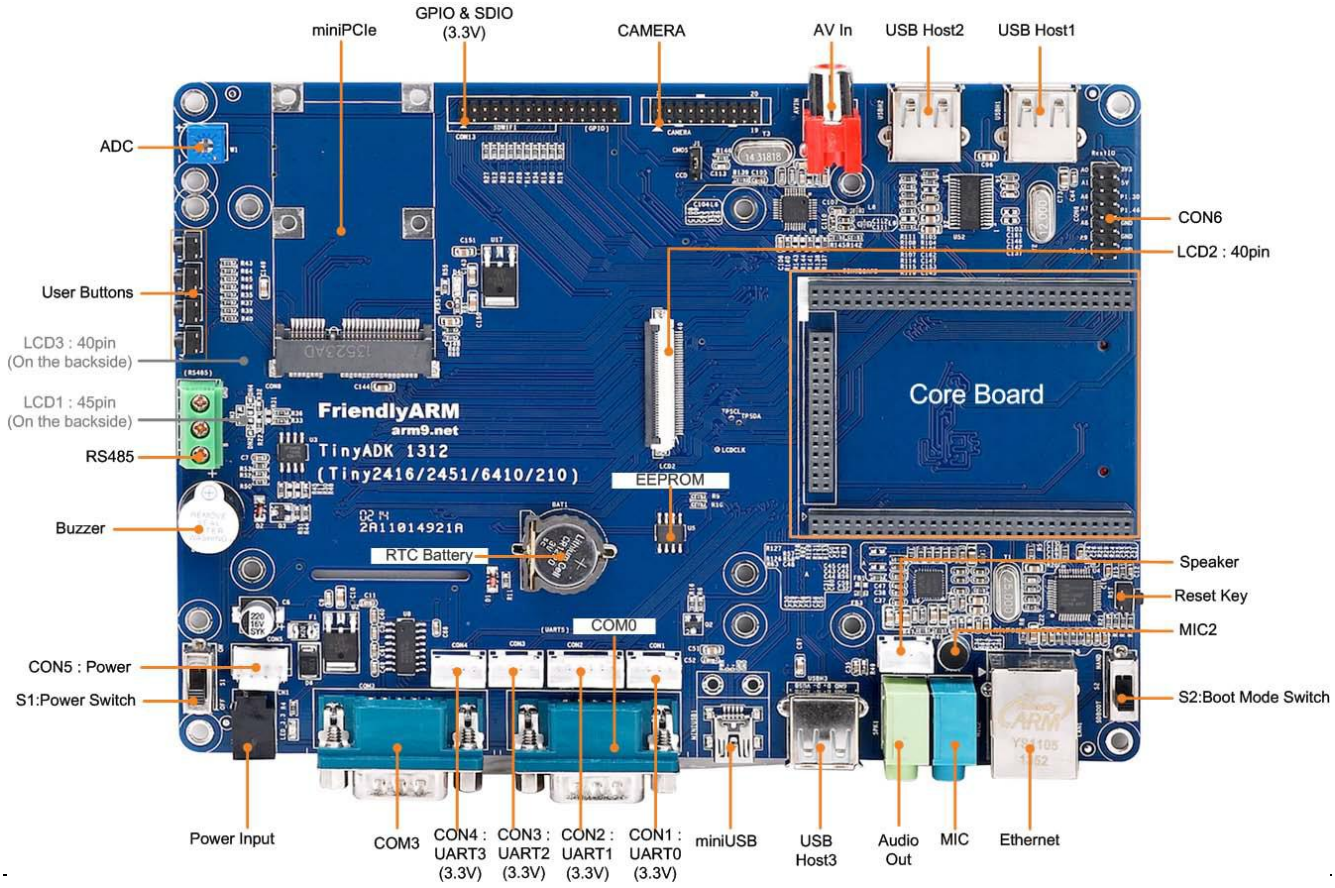
The TinyADK carrier board is a two-layer circuit board that demonstrates user-friendly reference designs with all the common interfaces. Components and interfaces (except the SD socket) are all located on one side for easy use.

LCD	● LCD1 interface (on the reverse): 45Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching ● LCD2 interface (on the obverse): 40Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching ● J-TAG: test points on the CPU board ● LCDs supported from 3.5” to 12.1” up to maximum resolution of 1024x768

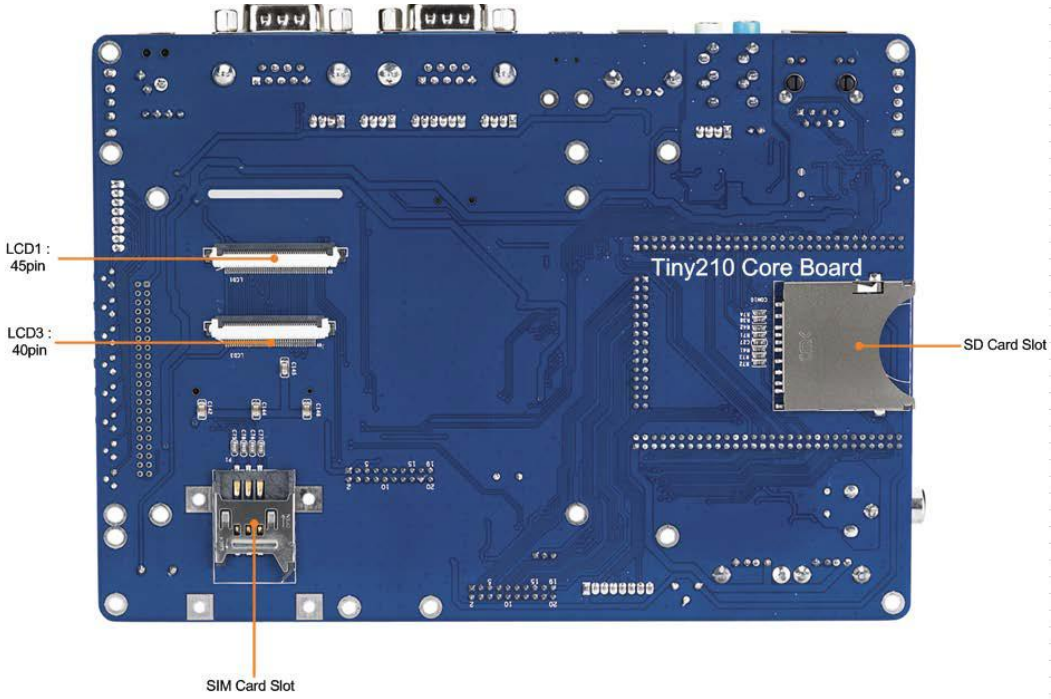
Network	● 1 x 10/100M Ethernet interface(RJ45) using DM9000AEP
Standard Configuration	● 1 x miniPCIe interface ● 2 x DB9 RS232 serial port ● 1 x miniUSB Slave-OTG 2.0 which can be extened via a 2.0mm socket ● 1 x 3.5mm stereotype audio output ● 1 x integrated microphone ● 3 x USB Host 2.0 ● 1 x standard SD card socket ● 1 x 5V power input (DC-23B)
On Board Hardware Resource	● 1 x I2C-EEPROM (256byte) for I2C bus testing ● 4 x Interrupt Style Push Button ● 1 x test pot for ADC testing ● 1 x PWM buzzer ● 1 x backup battery for on board real time clock
External Resource	● 4 x TTL ● 6 x channels AD Input ● 1 x SDIO ● 1 x CMOS camera interface ● 1 x AV IN for CCD camera ● 1 x RS485
PCB Dimension	● Two Layered Board ● Dimension: 180 x 130(mm)
Software	● Superboot ● Android 4.0

	• Android 2.3 • Linux-3.0.8 + Qt4.8.5/Qtopia2/Qtopia4 • Windows CE6.0

1.5.2 TinyADK 1312B Carrier Board Layout

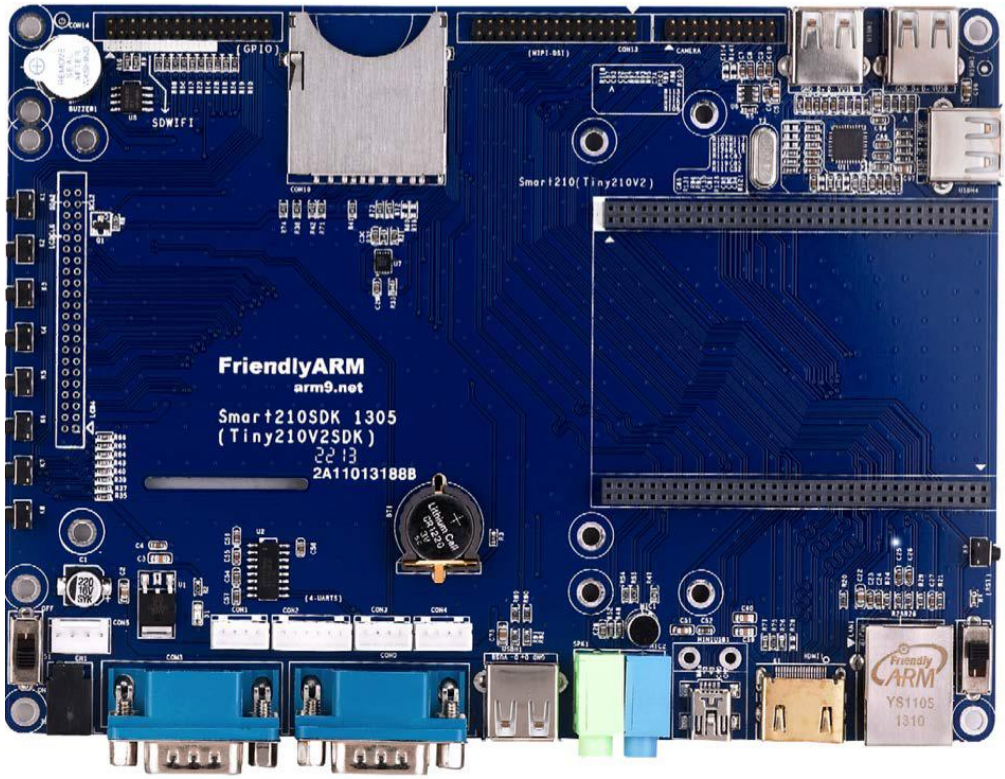


Obverse



Reverse

1.6 Smart210 1305 Carrier Board

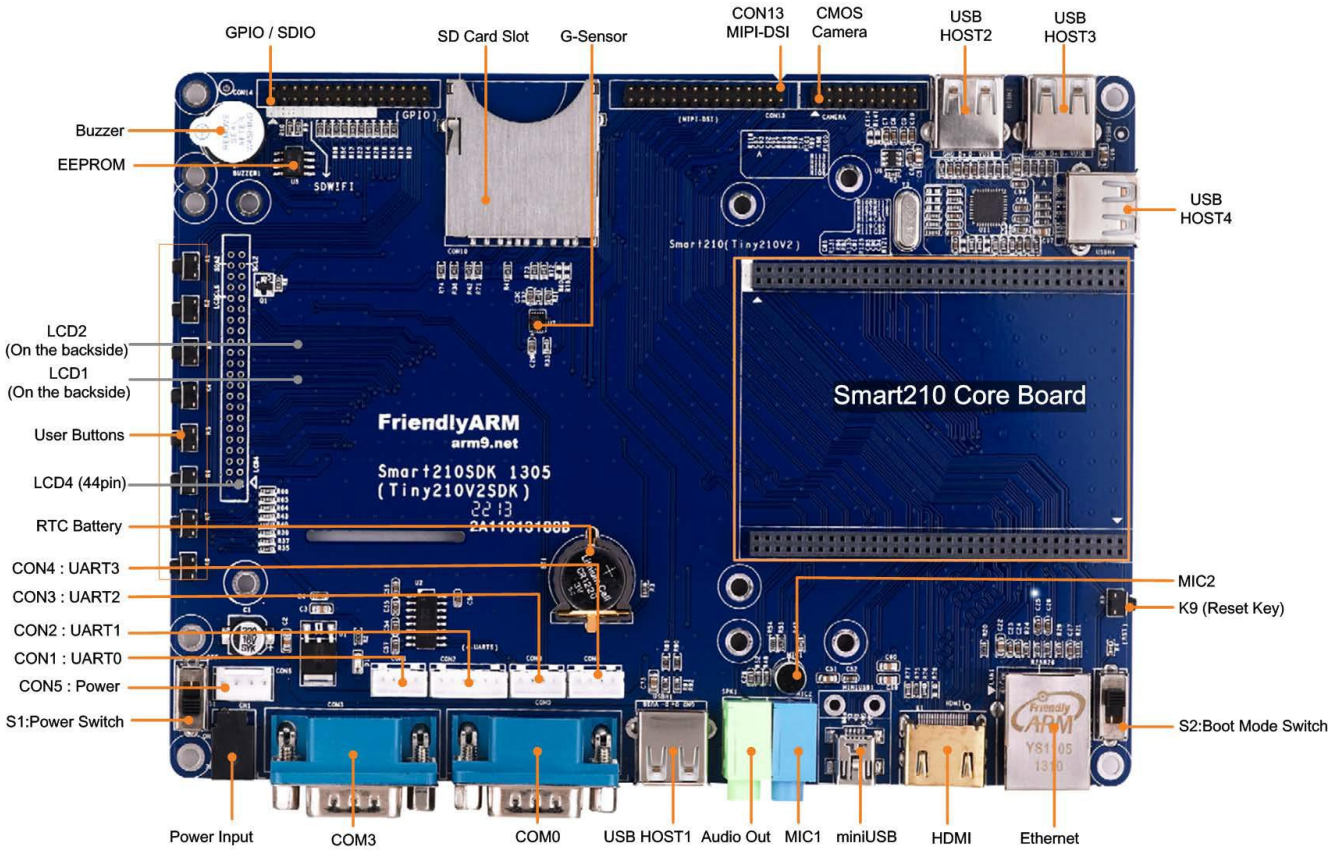


1.6.1 Smart210 1305 Carrier Board Hardware Feature

LCD	LCD1 interface: 45Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching LCD2 interface: 40Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching LCD4 interface: 44Pin, 0.5mm spacing, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching and capacitive touch HDMI high definition interface (Type A) LCDs supported from 3.5" to 12.1" up to maximum resolution of 1024x768
Network	1 x 10/100M Ethernet interface(RJ45) using DM9000AEP
Standard Configuration	2 x DB9 RS232 serial port 1 x miniUSB Slave-OTG 2.0 which can be extened via a 2.0mm socket 1 x 3.5mm stereotype audio output 1 x integrated microphone 4 x USB Host 2.0 1 x standard SD card socket 1 x 5V power input (DC-23B)

On Board Hardware Resource	1 x I2C-EEPROM (256byte) for I2C bus test 8 x Interrupt Style Push Button on module 1 x PWM buzzer 1 x backup battery for on board real time clock 1 x G-sensor chip
External Resource	4 x TTL socket 1 x GPIO 1 x CMOS camera interface 1 x MIPI
PCB Dimension	Two Layered Board Dimension: 180 x 130(mm)
Software	Superboot Android 4.0 Android 2.3 Linux-3.0.8 + Qt4.8.5/Qtopia2/Qtopia4 CE6

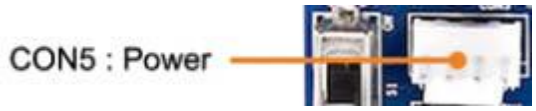
1.6.2 Smart210 1305 Carrier Board Layout



1.7 Tiny210/Smart210 Carrier Board Interface and Port

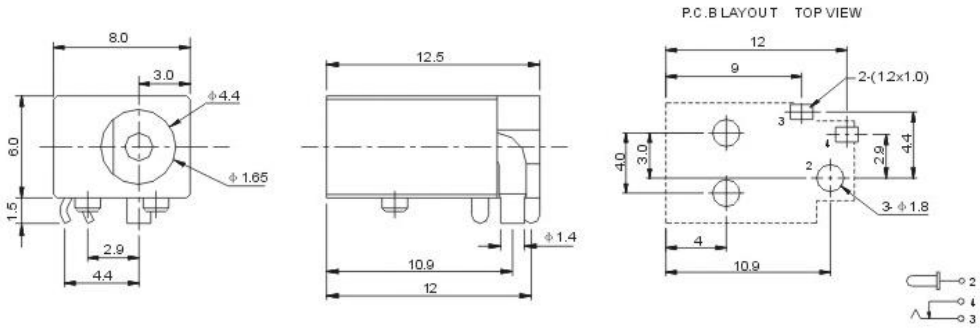
1.7.1 Power

The carrier board requires 5V DC and has two power input ports. CN1 is the barrel jack for the 5V / 2A PSU included with SDK kits. The 4 pin white CON5 takes a connector with a “click in place” for secure power in enclosures or commercial applications.

CON5	NO.	Pin Spec
	1	VDD5V
	2	GND
	3	GND
	4	VDDIN
	Note: this layout is convenient for users to connect S1 when it is extended	

Power Socket:

Type: DC023B

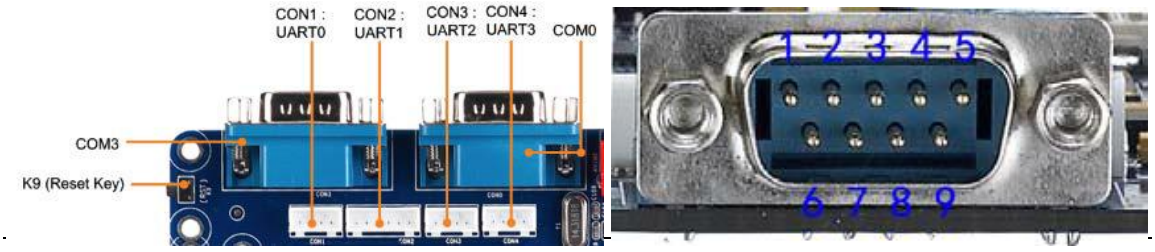


1.7.2 Serial Port

The S5PV210 has four serial ports: UART0、1、2 and 3. The UART1 is a four wire serial port and UART0, 2 and 3 are two wire serial ports.

The UART0 and 3 are converted to RS232 (COM0 and COM3). You need to use the shipped cross serial cable to connect the board to a PC.

Details of CON1，CON2，CON3, CON4 are as follows:



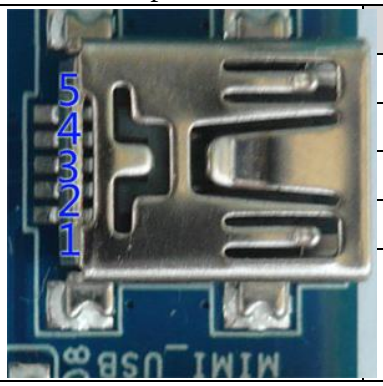
CON2	Pin Spec(TTL)	CON1, 3, 4	Pin Spec(TTL)	COM0	Pin Spec(RS232)
1	RTSn	1	NC	1	NC
2	CTSn	2	NC	2	RSRXD
3	TXD	3	TXD	3	RSTXD
4	RXD	4	RXD	4	NC
5	5V	5	5V	5	GND
6	GND	6	GND	6	NC
				7	NC
				8	NC
				9	NC
Note: NC means no connection					

1.7.3 USB

This board has two types of USB interfaces: one USB Host (2.0) and one USB Slave (2.0). The USB Host functions the same as on PC's. You can access USB camera, USB keyboard, USB mouse, USB flash drives and other common USB peripherals. You can also use USB Hub extensions. Each OS has native USB Hub drivers. The USB Slave (MiniUSB 2.0) is generally used to download from a host. If you use WinCE, connecting from a PC to this USB port will invoke ActiveSync and the 210 can be viewed as a volume, and data and programs dragged and dropped between host PC and

the 210 and the other miniUSB (2.0)

miniUSB Spec:



miniUSB	Pin Spec
5	GND
4	OTGID
3	D+
2	D-
1	Vbus

USB Host Pin Spec

USB Host	Pin Spec
1	5V
2	D-
3	D+
4	GND

1.7.4 Network Interface

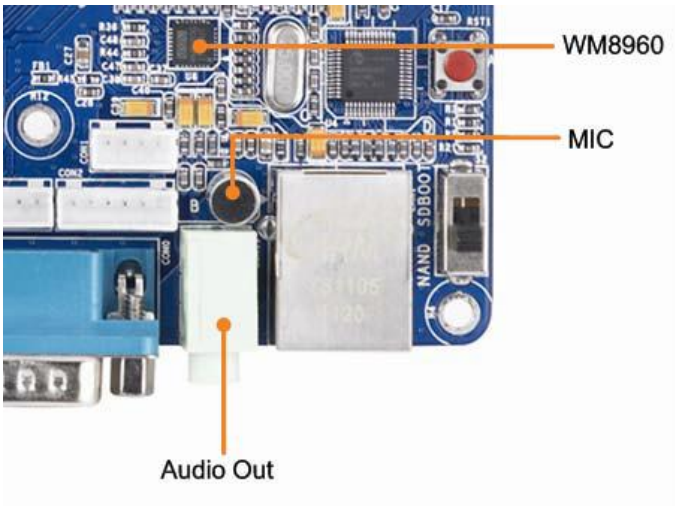
The 210 carrier board has a DM9000 LAN chip for adaptive 10/100M Ethernet. The RJ45 connector includes the magnetics. You can use ordinary Cat5 with RJ45 to connect to your router or switch.

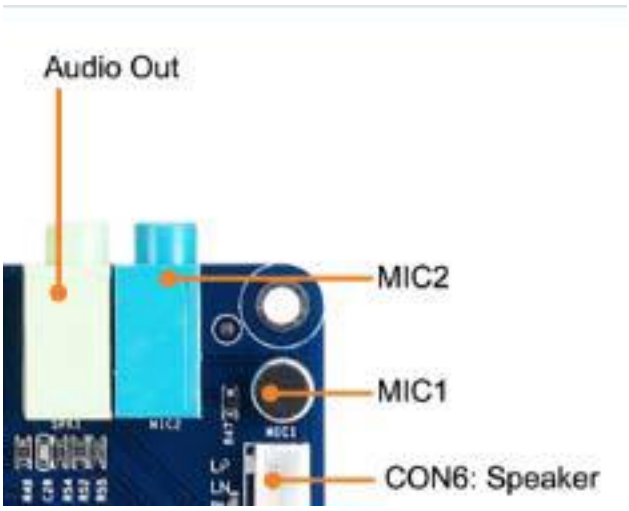
1.7.5 Audio

The S5PV210 supports I2S/PCM/AC97 audio interfaces. The 210 board uses the I2S0 interface with the WM8960 CODEC.

The audio output is on the 3.5mm green jack. The WM8960 has a D type amplifier. The board also has a microphone input. When recording we suggest users to move the

microphone close to the audio source





1.7.6 User Button

The Smart210 1305 carrier board has 8 buttons. The TinySDK/ADK 1312B has four user buttons connected directly to the CPU's interrupt pins. They trigger low when closed.

Button	K1	K2	K3	K4
Interrupt	EINT16	EINT17	EINT18	EINT19
GPIO	GPH2_0	GPH2_1	GPH2_2	GPH2_3
Notes: 1. The Smart210 1305 carrier board's CON14 has all these buttons' pins 2. The TinySDK/ADK 1312B's CON13 has all these buttons' pins. Please refer to the schematic for more details				

1.7.7 LCD Interface

For convenience in mounting various displays the 210 has three LCD connectors one of which is a 45pin and can work with both one wire touch LCDs and capacitive LCDs.

The LCD control signals are the same on all connectors with horizontal and vertical scan, clock, enable, disable etc., and 8:8:8 models of RGB data. It has a PWM output and a reset signal (nRESET). LCD_PWR is the backlight switch signal.

Since we apply the one wire precise touch technology neither LCD interface has the four wire resistor touch pins that the CPU uses by default (LCD1-37, 38, 39 and 40). This way gives us flexibility to connect capacitive screens. The LCD2 has more CPU pins

LCD1 & LCD2	Pin Spec	LCD1 & LCD2	Pin Spec
1	VDD_5V	2	VDD_5V
3	VD0	4	VD1
5	VD2	6	VD3
7	VD4	8	VD5
9	VD6	10	VD7
11	GND	12	VD8
13	VD9	14	VD10
15	VD11	16	VD12
17	VD13	18	VD14
19	VD15	20	GND

21	VD16	22	VD17
23	VD18	24	VD19
25	VD20	26	VD21
27	VD22	28	VD23
29	GND	30	PWM1/GPD0_1
31	XEINT10/GPH1_2	32	nRSTOUT
33	VDEN	34	VSYNC
35	HSYNC	36	VCLK
37	I2CSCL2	38	XEINT14/GPH1_6
39	I2CSDA2	40	XEINT15/GPH1_7
41	GND		
Note: the S5PV210 has three I2Cs and here we use I2C2. Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference			

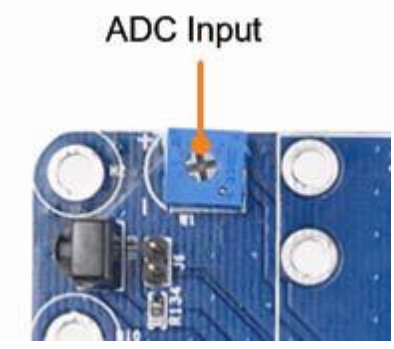
1.7.8 ADC

The Smart210 SDK 1305 doesn't have ADC inputs.

The TinySDK/ADK has 6 ADC channels for different purposes:

- AIN0 is connected to a variable resistor W1 for testing
- AIN0,1,4, 5, 6 and 7 are connected to CON6

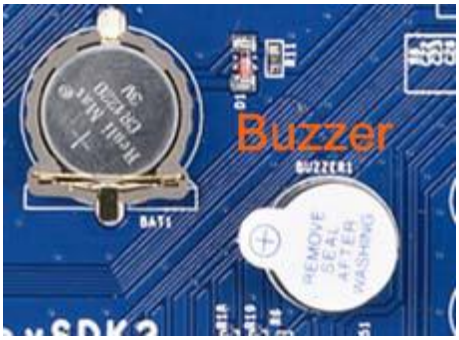
The S5PV210's AD conversion can be configured to 10-bit or 12-bit data for 0V ~ 3.3V



Note: the Smart210 SDK 1305 doesn't have this

1.7.9 PWM Buzzer

The on-board buzzer is controlled by PWM0, the diagram is shown below. PWM0 corresponds to GPD0_0 which can be configured as PWM output via software or used as a GPIO.



1.7.10 I2C-EEPROM

The 210 has a direct connection to an AT24C08 – an I2C EEPROM which has a capacity of 256 bytes and is mainly for testing I2C bus.

Note: the S5PV210 has three I2Cs and here the 210 uses I2C0

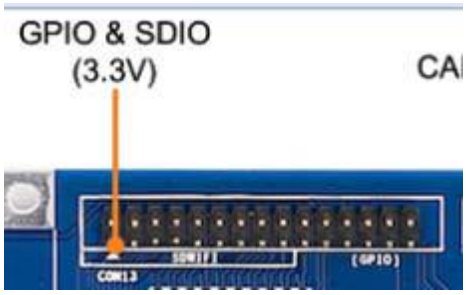
1.7.11 SD Card Socket

The 210 has two SDIO interfaces. The SDIO0 is used for ordinary SD cards. The interface can support SDHC, that is, high-speed large-capacity cards up to 32G bytes.

1.7.12 SDIO/GPIO Interface

The S5PV210's third SDIO is connected to CON13 on the TinySDK/ADK 1312B and to CON14 on the Smart210 SDK 1305, which is a 2.0mm spacing 20Pin header. It has SPI, I2C and four GPIOs

The SDIO is usually used for SD-WiFi.

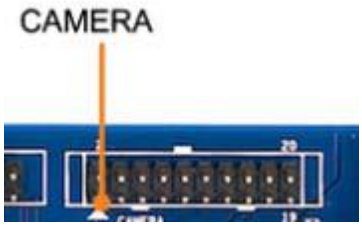


CON13/CON14	Pin Spec	CON13/CON14	Pin Spec
1	3.3V	2	GND
3	TXD2	4	RXD2
5	I2CSCL0	6	I2CSDA0
7	SPIMOSI	8	SPIMISO
9	SPICLK	10	SPICS
11	GPIO (WIFI Power Down)	12	GPIO
13	SD1_CLK	14	SD1_CMD
15	SD1_nCD	16	XEINT9/XEINT11
17	SD1_DAT0	18	SD1_DAT1
19	SD1_DAT2	20	SD1_DAT3
21	XEINT16/KP_COL0	22	XEINT17/KP_COL1
23	XEINT18/KP_COL2	24	XEINT19/KP_COL3
25	XEINT24/KP-ROW0	26	XEINT25/KP_ROW1
27	XEINT26/KP_ROW2	28	XEINT27/KP_ROW3
29	5V	30	GND
Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference			

1.7.13 CMOS Camera

The TinySDK/ADK 1312B and Smart210 SDK 1305 have a CMOS camera interface which is a 2.0mm spacing 20 pin header. Users can use our CAM130 cameras by connecting it to this header. Actually the CAM130 doesn't have any circuits and it is a conversion board which applies the ZT130G2 module.

Note: the CAMER interface is a multiplexed port which can be used as a GPIO by setting corresponding registers. The table below shows its GPIO pins. In addition the CMOS Camera interface is also connected to a TVP5150 chip. Users can select either CCD or CMOS by jumping J3 to different positions.



CAMERA			
CAMERA	**Pin Spec**	**CAMERA**	**Pin Spec**
1	I2CSDA0	2	I2CSCL0
3	XciFIELD	4	CAM_A_RESET/GPJ3_1
5	CAM_A_CLK	6	CAM_A_HREF
7	CAM_A_VSYNC	8	CAM_A_PCLK
9	CAM_A_DATA7	10	CAM_A_DATA6
11	CAM_A_DATA5	12	CAM_A_DATA4
13	CAM_A_DATA3	14	CAM_A_DATA2
15	CAM_A_DATA1	16	CAM_A_DATA0
17	VDD_3.3V	18	VDD_2.45-2.8V
19	VDD_1.8V	20	GND
Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference			

2 Introduction to Mini210S

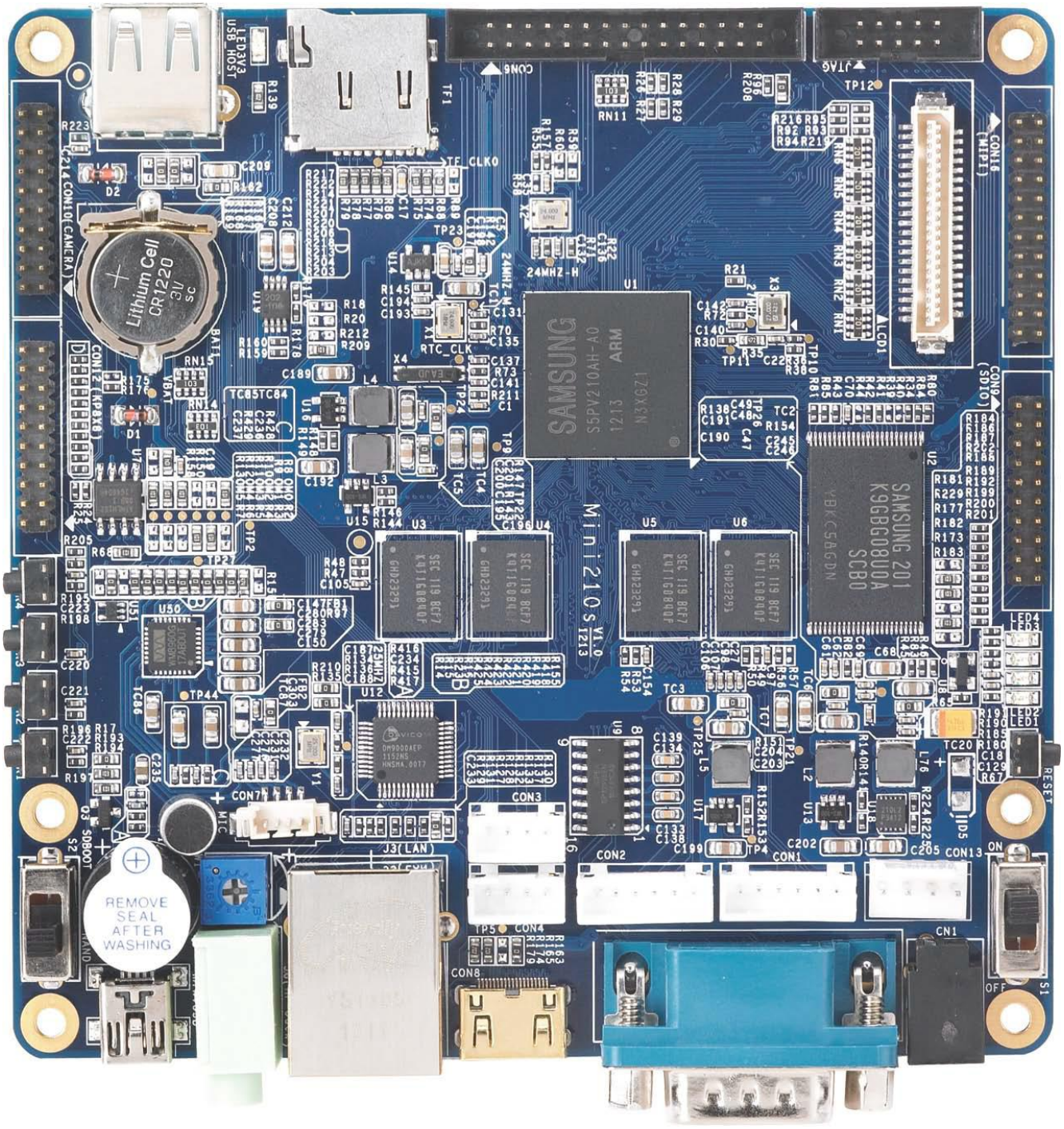
The Mini210S development board is a powerful Cortex-A8 board offering a comprehensive solution integrating both hardware and software. It is designed, developed and distributed by FriendlyARM. It incorporates Samsung's S5PV210 microprocessor whose maximum frequency is up to 1GHz. The S5PV210 integrates the PowerVR SGX540 graphic engine, supports 3D and can drive video playing on screens up to 1080P. It is equipped with a 5" LCD, 512M DDR2, 1G SLC NAND Flash, SD WiFi, D type WM8960 audio which supports 8Ω 1W speakers. In addition it has a miniHDMI output, USB2.0 camera and 8 x 8 matrix keyboard. It also supports power idle mode. These features make it easily and widely used in MID development, Android notepads, auto electronic devices, industrial applications, GPS systems and multimedia systems.

It is very easy and convenient for users to refresh the system with various OS via a TF card with our specially developed Superboot.

2.1 Mini210S Board

2.1.1 Mini210S Overview

Here is an overview of the Mini210S board.



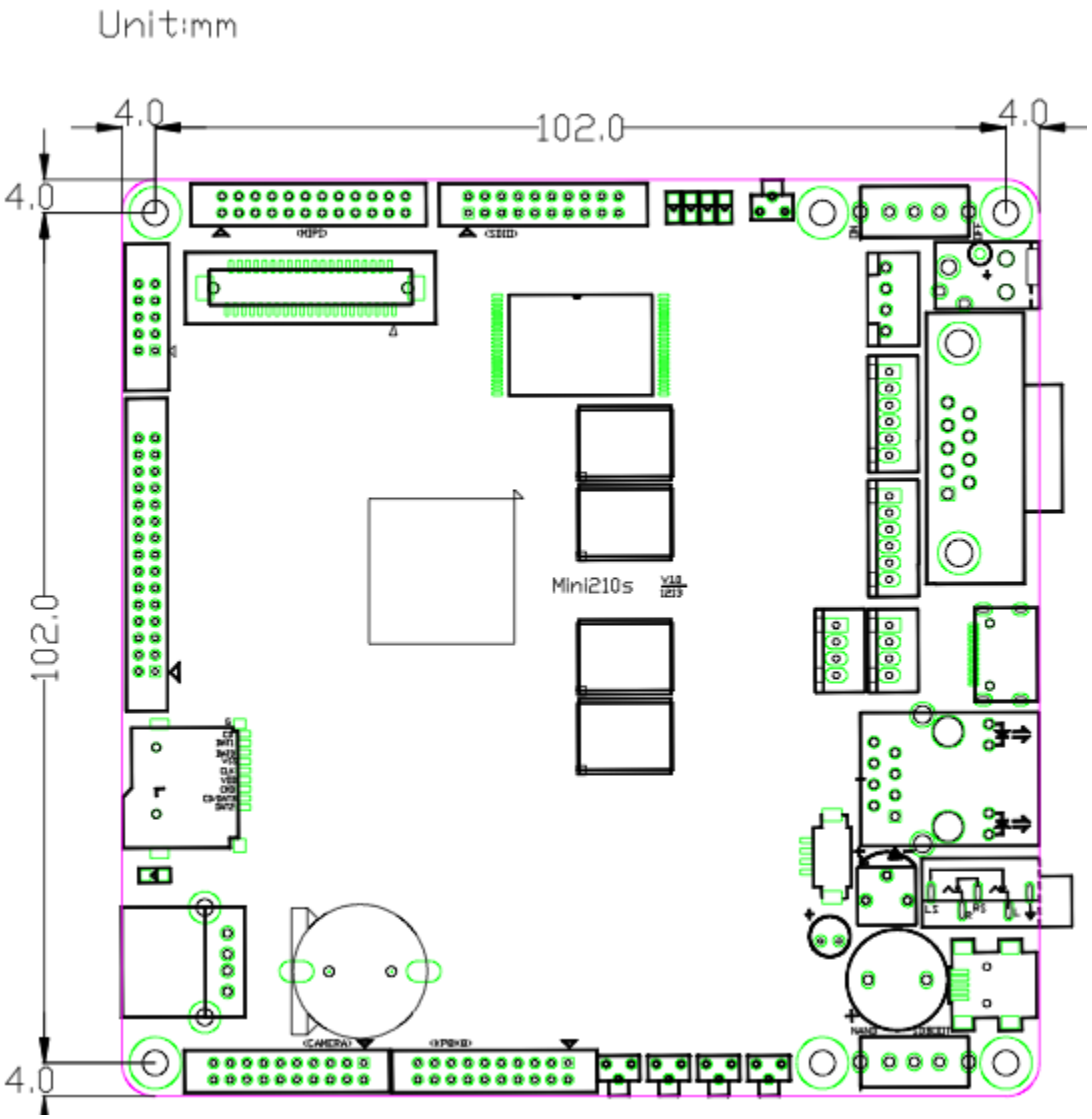
2.1.2 Mini210S Hardware Features

CPU	● Samsung S5PV210, based on CortexTM-A8, 1GHz

	● PowerVR SGX540 graphic engine ● 2D/3D graphic accelaration ● Up to 1080p@30fps hard decoded video playing, support MPEG4, H.263, H.264 etc ● Up to 1080p@30fps hard decoded (Mpeg-2/VC1) video input
DDR2 RAM	● 512M ● 32bit data bus, dual channels
FLASH	● SLC NAND Flash: 1GB
LCD	● LCD1 interface: 41Pin, 1.0mm spaced, compatible with Mini2440/Mini6410 LCD, supports one wire precise touching including an I2C, three interrupts and a PWM output ● miniHDMI high definition interface (Type C) ● LCDs supported from 3.5" to 12.1" up to maximum resolution of 1024x768
Network	● 10/100M Ethernet interface(RJ45) using DM9000AEP
Standard Configuration	● 1 x DB9 RS232 serial port ● 4 x TTL socket ● 1 x miniUSB Slave-OTG 2.0 which can be extened via a 2.0mm socket ● 1 x 3.5mm stereotype audio output ● 1 x integrated microphone ● 1 x one speaker port which can drive an 8Ω 1W speaker ● 1 x USB Host 2.0 ● 2 x standard TF card socket ● 1 x 5V power input (DC-23B)

On Board Hardware Resource	● 1 x I2C-EEPROM (256byte) for I2C bus test ● 4 x LEDs (green) ● 4 x User's Button (interrupt pin) ● 1 x adjustor resistor for ADC test ● 1 x PWM buzzer ● 1 x backup battery for on board real time clock
External Resource	● 4 x TTL ● 1 x JTAG ● 1 x LCD interface ● 1 x SDIO ● 2 x CMOS camera interface ● 1 x matrix keyboard interface: 20pin ● 1 x GPIO
PCB Dimension	● Four layered baord ● Dimension: 110 x 110 x 1.6(mm)
Bootload + OS	● Superboot ● Android 4.0 ● Android 2.3 + Linux-2.6.35(kernel) ● Linux-3.0.8 + Qt4.8.5/Qtopia2/Qtopia4 ● WindowsCE 6.0

2.1.3 Mini210S Board Dimension



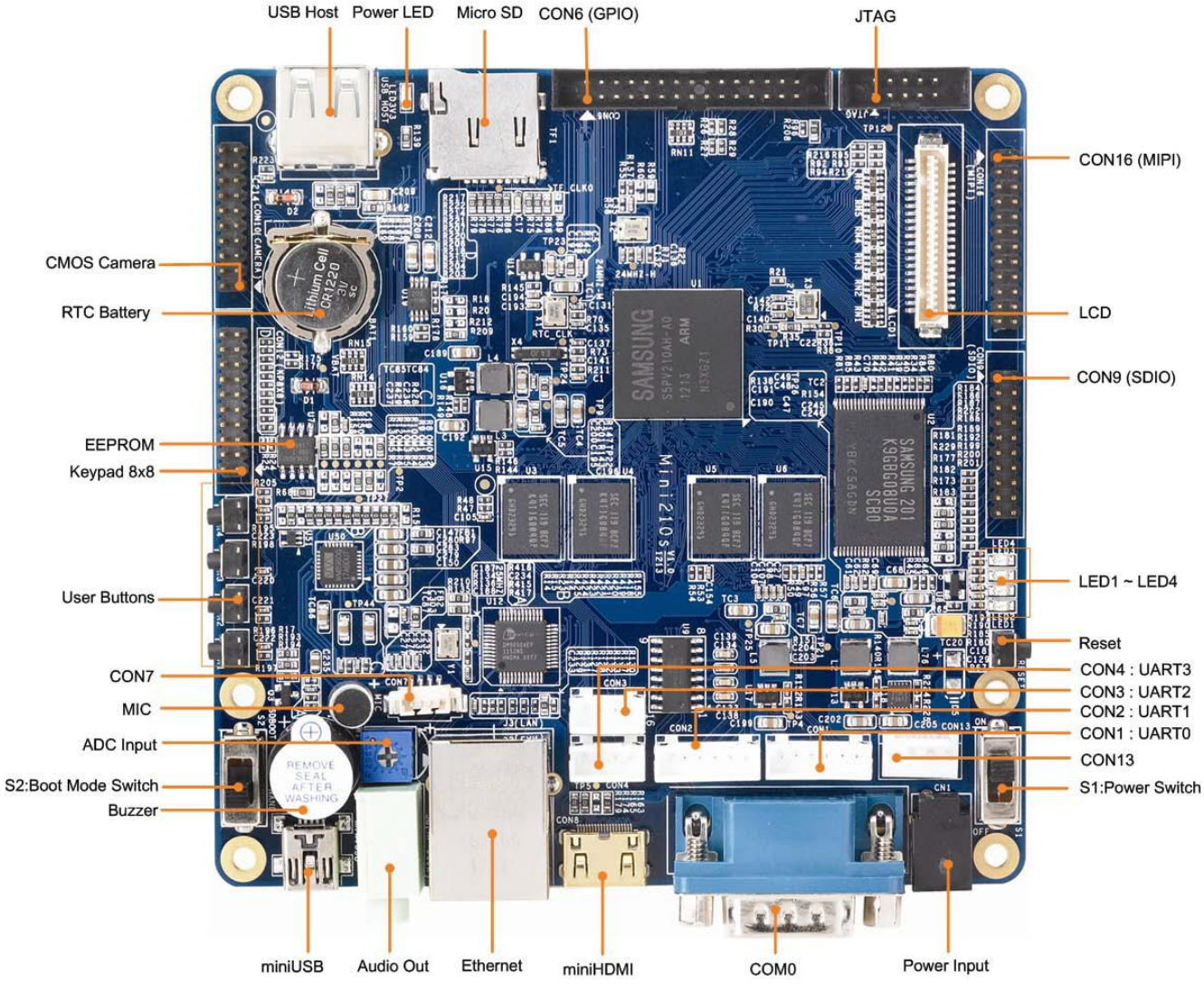
2.2 Mini210S Board Jumpers and Diagram

2.2.1 Jumpers

The Mini210S doesn't have jumpers.

2.2.2 Board Diagram

Here is the Mini210S' diagram

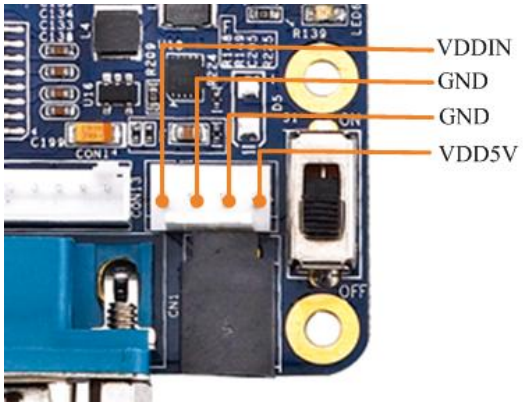


2.3 Mini210S Ports and Interfaces

This section describes in detail each interface/port on the board. For more details please refer to the complete schematics (in PDF and Protel99SE) in the CDs shipped together with this product.

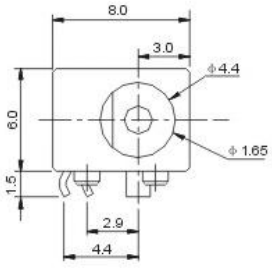
2.3.1 Power

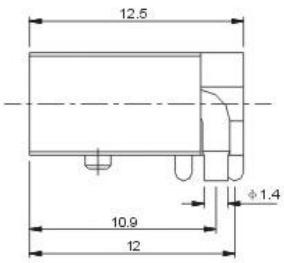
The Mini210S is powered by an external 5V power supply. It has two power inlets: CN1 is for 5V power adapter and the white CON13 is a 4 pin socket used to connect an external power supply when the board is embedded in a closed box.

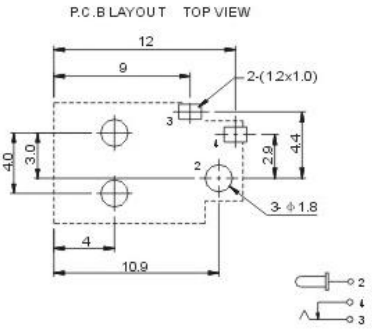
CON13	NO.	Pin Spec
	1	VDD5V
	2	GND
	3	GND
	4	VDDIN
	Note: this layout is convenient for users to connect S1 when it is extended	

Power inlet type and dimension:

Type: DC023B





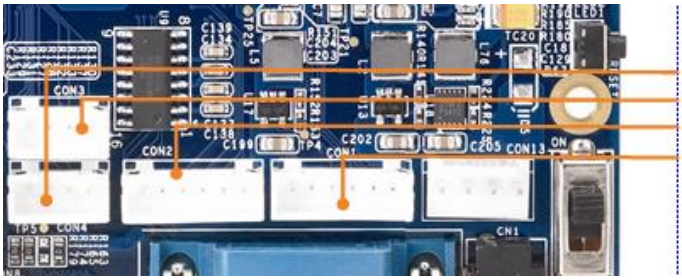


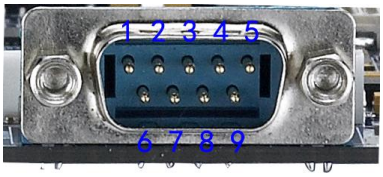
2.3.2 Serial Port

The S5PV210 has four serial ports: UART0, 1, 2 and 3. The UART0 and 1 are four wire serial ports and UART2 and 3 are two wire serial ports.

For the Mini210S the UART0 is converted to RS232 (COM0). You need to use the shipped cross serial cable to connect the board to a PC.

Details of CON1，CON2，CON3, CON4 are as follows:





CON1、2	Pin Spec(TTL)	CON3、4	Pin Spec(TTL)	COM0	Pin Spec(RS232)
1	RTSn	1	NC	1	NC
2	CTSn	2	NC	2	RSRXD
3	TXD	3	TXD	3	RSTXD
4	RXD	4	RXD	4	NC
5	5V	5	5V	5	GND
6	GND	6	GND	6	NC
				7	RSCTSn
				8	RSRTSn
				9	NC
Note: NC means floating					

2.3.3 USB

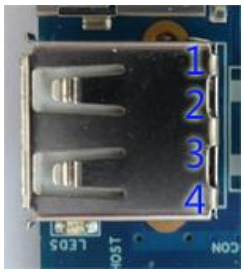
This board has two types of USB interfaces: one USB Host (2.0) which can be used to connect USB cameras, USB keyboard, USB mouses and flash drives; and the other miniUSB(2.0), which is OTG and usually used for ADB in Android

miniUSB Spec:

miniUSB	Pin Spec
5	GND
4	OTGID
3	D+

	2	D-
	1	Vbus

USB Host Pin Spec

	USB Host	Pin Spec
	1	5V
	2	D-
	3	D+
	4	GND

2.3.4 Ethernet

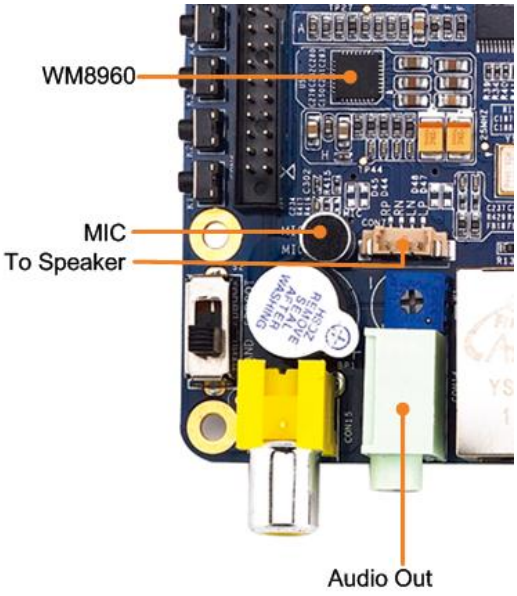
The Mini210S integrates a RJ45 (10M/100M) Ethernet interface which uses the DM9000 chip. Users can connect with a standard Ethernet cable to browse the internet.

2.3.5 Audio

The S5PV210 supports I2S/PCM/AC97 audio interfaces. The Mini210S uses the I2S0 interface, and integrates a WM8960 CODEC module.

The audio output is a 3.5mm green jack. The WM8960 has a D type amplifier therefore the Mini210S extends a Speaker socket(CON7) which can be connected to a 8 Ω 1W speaker.

The Mini210S also has a microphone input. When recording we suggest users to move the microphone close to the audio source.



2.3.6 HDMI

The Mini210S has a mini HDMI output. The Mini210S extends the output to a Type C miniHDMI. Users can connect the board to an HDMI monitor or TV via a standard HDMI cable.

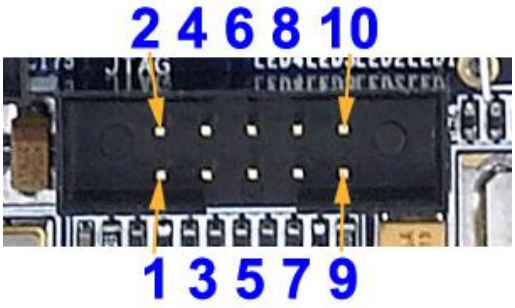
Note: Android supports simultaneous output to both LCD and HDMI.

2.3.7 JTAG

When a board just comes off from production lines it is just a bare board without any data and we usually have to burn the first program to it through the JTAG interface.

However since the S5PV210 supports booting from the TF card the JTAG is not significant to users any more. Now the JTAG is more often used for debugging. In fact, most of the widely used utilities in markets like JLINK, ULINK and other simulators actually work via the JTAG interface. A standard JTAG has 4 signals :TMS, TCK, TDI and TDO which are test mode select input, test clock, test data input and test data output. These 4 signal lines plus a power line and a ground line form 6 lines in total. In order for testing, most simulators even have a reset signal. Therefore, a standard JTAG is supposed to have those signal lines, and it does not specify whether it is 20Pin or 10Pin. As long as a JTAG interface has those signal lines it will be a standard JTAG interface. The Mini210 has a 10Pin JTAG interface which has complete standard JTAG signals.

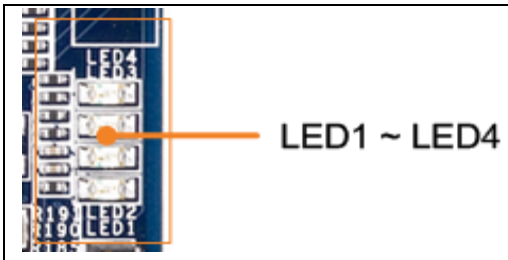
Notes: for beginners who just want to focus on Linux or WinCE development, the JTAG interface has no significance because most development boards already have a complete BSP which includes commonly needed serial port, network port and USB port.When a board runs with Linux or WinCE installed, users can fully utilize more convenient functions and utilities provided by the operating system to debug. They do not need a JTAG. Even if you can trace your programs it will be extremely tough to step debug because it will go into the operating system. This is not an easy job.



JTAG Interface				
2	4	6	8	10
3.3V	nRESET	TDO	GND	GND
1	3	5	7	9
3.3V	nTRST	TDI	TMS	TCK

2.3.8 LED Indicator

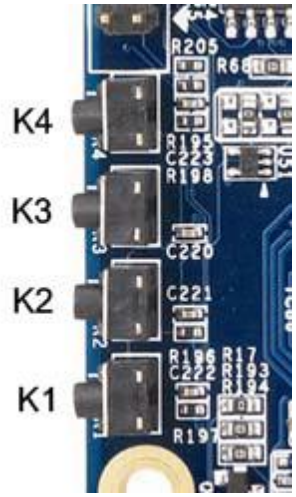
LED is a commonly used status indication device. The Mini210S has four programmable LEDs which are directly connected to GPIO and are on at a low level voltage.



	LED1	LED2	LED3	LED4
GPIO Pins	GPJ_0	GPJ_1	GPJ_2	GPJ_3

2.3.9 User Button

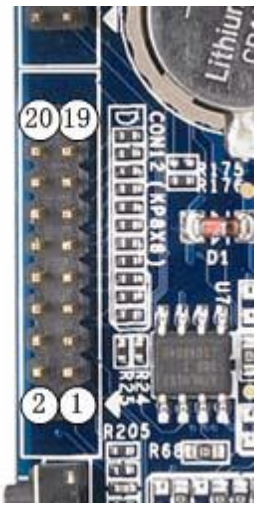
The Mini210S has four user buttons all of which are extended from CPU's interrupts and are triggered at a low level voltage.



Key	K1	K2	K3	K4
Interrupt	EINT16	EINT17	EINT18	EINT19
Multiplexed GPIO	GPH2_0	GPH2_1	GPH2_2	GPH2_3

2.3.10 Matrix Keyboard

The S5PV210 supports 8x8 keyboards. The Mini210S extends the pins to CON12:



CON12 Pin	Pin Spec	CON12 Pin	Pin Spec
1	XEINT16/KP_COL0	2	XEINT24/KP_ROM0
3	XEINT17/KP_COL1	4	XEINT25/KP_ROW1
5	XEINT18/KP_COL2	6	XEINT26/KP_ROW2

7	XEINT19/KP_COL3	8	XEINT27/KP_ROW3
9	XEINT20/KP_COL4	10	XEINT28/KP_ROW4
11	XEINT21/KP_COL5	12	XEINT29/KP_ROW5
13	XEINT22/KP_COL6	14	XEINT30/KP_ROW6
15	XEINT23/KP_COL7	16	XEINT31/KP_ROW7
17	VDD_3.3V	18	VDD_3.3V
19	GND	20	GND

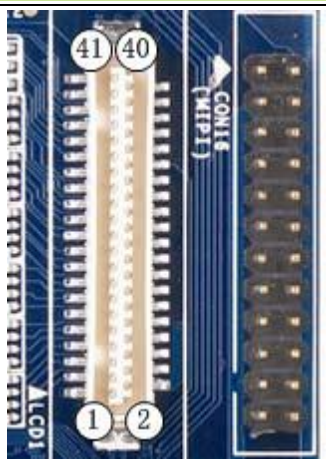
Note:

1. CON12 is a standard IDC 2.0mm 20Pin socket.
2. XINT16/KP_COL0 means the pin can be multiplexed to interrupt XEINT16. This specification applies to other pins too.
3. Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference

2.3.11 LCD Interface

The LCD interface is a 1.0mm spaced 41 pin connector. It has most of the commonly used control signals (line scan, clock, enable/disable) and complete RGB data signals (RGB output is 8:8:8 and can support LCDs up to 16M pixels). It has a PWM output and a reset signal (nRESET). LCD_PWR is the backlight switch signal.

Since we apply the one wire precise touch technology the LCD interface doesn't have the four wire resistor touch pins that the CPU uses by default (LCD1-37, 38, 39 and 40). This way gives us flexibility to connect capacitive screens.



LCD1 & LCD2	Pin Spec	LCD1 & LCD2	Pin Spec
1	VDD_5V	2	VDD_5V
3	VD0	4	VD1
5	VD2	6	VD3
7	VD4	8	VD5
9	VD6	10	VD7
11	GND	12	VD8
13	VD9	14	VD10
15	VD11	16	VD12
17	VD13	18	VD14
19	VD15	20	GND
21	VD16	22	VD17
23	VD18	24	VD19
25	VD20	26	VD21
27	VD22	28	VD23
29	GND	30	PWM1/GPD0_1
31	XEINT10/GPH1_2	32	nRSTOUT
33	VDEN	34	VSYNC
35	HSYNC	36	VCLK
37	I2CSCL2	38	XEINT14/GPH1_6
39	I2CSDA2	40	XEINT15/GPH1_7
41	GND		

Note: the S5PV210 has three I2Cs and here we use I2C2.

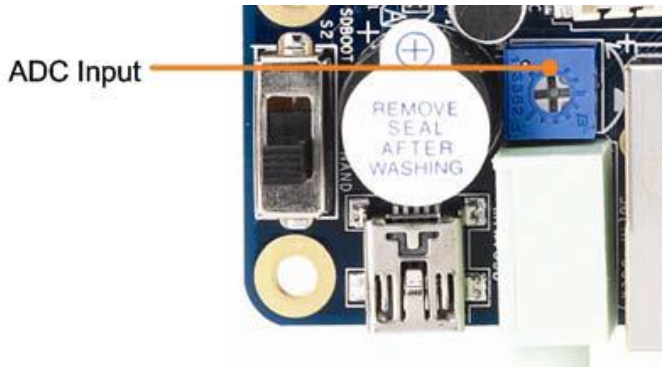
Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference

2.3.12 ADC

The Mini210S utilizes 6 ADC channels that the S5PV210 has for different purposes:

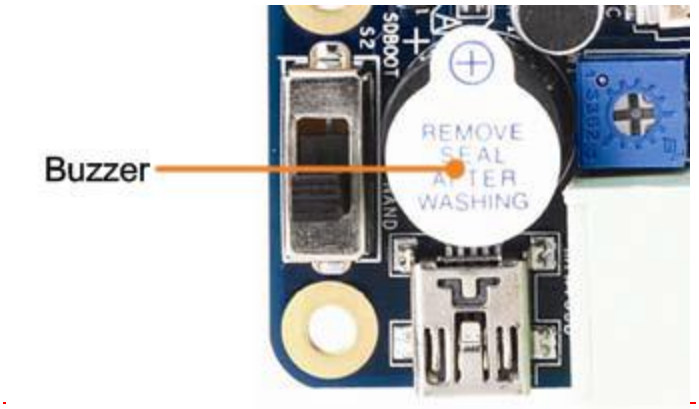
- AIN0 is connected to an adjustable resistor W1
- AIN1, 2, 3, 4 and 5 are connected to CON6 which includes other GPIOs as well

The S5PV210's AD conversion can be configured as either 10-bit or 12-bit.



2.3.13 PWM

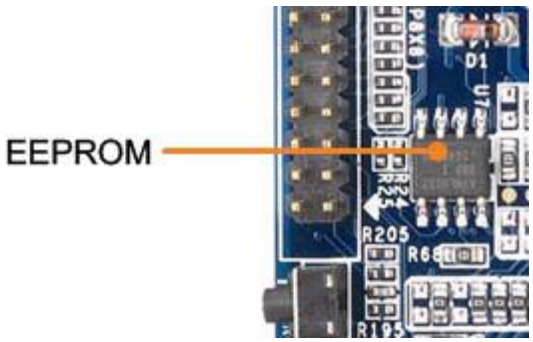
The on-board buzzer is controlled by PWM0, the diagram is shown below. PWM0 corresponds to GPD0_0 which can be configured as PWM output via software or used as a GPIO.



2.3.14 IIC-EEPROM

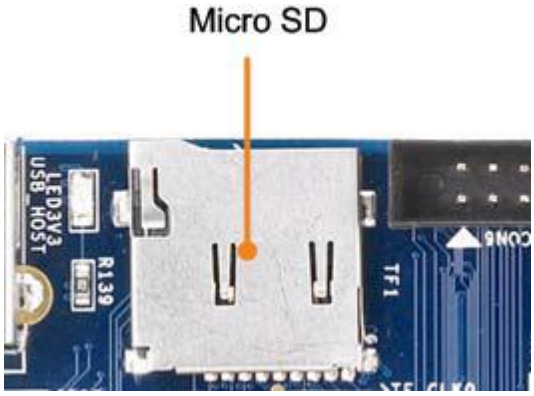
The Mini210S has an EEPROM AT24C08 connected to CPU's I2C0. It has 256 bytes memory and is mainly for testing I2C bus.

Note: the S5PV210 has three I2Cs and here the Mini210 uses I2C0



2.3.15 Micro SD/TF Card

The S5PV210 has four SDIO interfaces. The SDIO0 is extended to a Micro SD/TF socket.



2.3.16 SDIO/SD-WIFI

The S5PV210's third SDIO is extended to CON9 which is a 2.0mm 20Pin socket. It contains an SPI, an I2C and four GPIOs.

CON9	Pin Spec	CON9	Pin Spec
1	VDD_3.3V	2	GND
3	WIFI1_RST_GPIO/GPJ4_4	4	WIFI1_PWR_ONOFF/JPJ4_2
5	I2CSCL0	6	I2CSDA0
7	SPI0_MOSI0	8	SPI0_MISO0
9	SPI0_CLK0	10	SPI0_CS0
11	WIFI1_IO/GPJ4_1	12	WIFI1_PD_GPIO/GPJ4_3
13	MMC3_CLK	14	MMC_CMD2
15	MMC_CDn2	16	WIFI1__nWP
17	MMC3_DAT0	18	MMC3_DAT1
19	MMC3_DAT2	20	MMC3_DAT3
Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference			

2.3.17 CMOS Camera

The CMOS camera interface is extended to CON10. It is a 2.0mm 20 pin connector. Users can directly connect to a CAM130 camera. Actually the CAM130 doesn't have any circuits and it is a conversion board which is mounted with a ZT130G2 module.

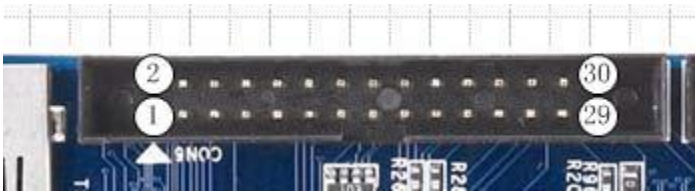
Note: The CMOS camera interface is a multiplexed port which can be used as a GPIO by setting corresponding registers. The table below shows its GPIO pins.

CAMERA			
CON10	**Pin Spec**	**CON10**	**Pin Spec**
1	I2CSDA0	2	I2CSCL0
3	XciFIELD	4	CAM_A_RESET/GPJ3_1
5	CAM_A_CLK	6	CAM_A_HREF
7	CAM_A_VSYNC	8	CAM_A_PCLK
9	CAM_A_DATA7	10	CAM_A_DATA6
11	CAM_A_DATA5	12	CAM_A_DATA4
13	CAM_A_DATA3	14	CAM_A_DATA2
15	CAM_A_DATA1	16	CAM_A_DATA0
17	VDD_3.3V	18	VDD_2.45-2.8V
19	VDD_1.8V	20	GND
Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference			

2.3.18 GPIO

GPIO is the abbreviated form of “General Purpose Input Output”. The Mini210S has a 30 Pin 2.0mm spaced GPIO interface, i.e. CON6.

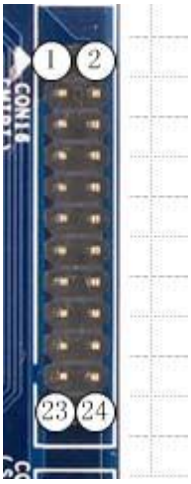
In fact, CON6 has not only quite a few GPIO pins but also some CPU pins such as AD input, SPI, I2S, PCM and so on. Most of them can be multiplexed by setting corresponding registers.



CON6	Pin Spec	CON6	Pin Spec
1	VDD_3.3V	2	GND
3	ADCAIN1	4	EINT0/GPH0_0
5	ADCAIN2	6	EINT1/GPH0_1
7	ADCAIN6	8	EINT2/GPH0_2
9	ADCAIN7	10	EINT3/GPH0_3
11	ADCAIN8	12	EINT4/GPH0_4
13	ADCAIN9	14	EINT5/GPH0_5
15	SDA0	16	EINT6/GPH0_7
17	SCL0	18	EINT9/GPH1_1
19	SDA1	20	SPICLK1/GPB4
21	SCL1	22	SPICS1/GPB5
23	SDA2	24	SPIMISO1/GPB6
25	SCL2	26	SPIMISOI1/GPB7
27	nRSTOUT	28	PWM2/GPD0_2
29	XPWRRGTON	30	PWM3/GPD2_3
Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference			

2.3.19 MIPI

The MIPI interface is extended to CON16. The following table lists the signals' specifications



CON6	Pin Spec	CON6	Pin Spec
1	VDD_3.3V	2	GND
3	VDD_5V	4	GND
5	mipiSDPCLK	6	mipiMDPCLK
7	mipiSDNCLK	8	mipiMDNCLK
9	mipiSDP0	10	mipiMDP0
11	mipiSDN0	12	mipiMDN0
13	mipiSDP1	14	mipiMDP1
15	mipiSDN1	16	mipiMDN1
17	mipiSDP2	18	mipiMDP2
19	mipiSDN2	20	mipiMDN2
21	mipiSDP3	22	mipiMDP3
23	mipiSDN3	24	mipiMDN3
Please refer to the schematics in the shipped CD for the exact connection details between each pin and CPU. The information provided here is for reference			

3 Software Features

3.1 Android 2.3.1 Features

Cross-compiler	arm-linux-gcc-4.5.1-v6-vfp	Same as Mini6410, by default it compiles with armv7 command set. It supports hard floating point arithmetic
Superboot-210	It supports SD card system burning and can install (YAFFS2) systems within 1.8 seconds.	Superboot is especially developed for enterprise users
	It has a graphic interface and can display LCD info, hardware configurations, installation process	
	It can automatically detect MMC/NAND booting mode.	
	It needs the SD-Flasher burning utility which runs on WindowsXP/Vist/Windows7	
Android kernel	Kernel version: Linux-2.6.35	Complete BSP
	It works with YAFFS2/CRAMFS/NFS/UBIFS/NFS/FAT32.	Open source
	Watchdog	Open source, provided by Samsung
	RTC driver	Open source, provided by Samsung
	LED driver	Open source, migrated by FriendlyARM
	User button driver	Open source, migrated by FriendlyARM
	SPI driver	Open source,it comes with the kernel (2011.1.16) but is not verified by FriendlyARM
	I2C-EEPROM driver	Open source, provided by Samsung
	PWM buzzer driver	Open source, migrated by FriendlyARM

	ADC driver (channel: AIN0)	Open source, migrated by FriendlyARM
	Touch screen coontroller driver which comes with CPU by default	Open source, provided by Samsung(we don't use it and use one wire precise touching instead)
	One wire precise touch driver	Open source, migrated by FriendlyARM
	LCD back light driver: it allows users to adjust the board's backlight up to 127 levels	Open source, migrated by FriendlyARM
	LCD driver(4.3", 5", 7" etc): it supports screen rotation	Open source, migrated by FriendlyARM. It drives screens based on the initialization parameter "lcd="
	USB Host driver: it supports flash drives, blue tooth and so on.	Open source, provided by Samsung
	USB Device driver: it supports USB ADB	Open source, provided by Samsung
	SD card driver	Open source, provided by Samsung
	Serial port driver	Open source, provided by Samsung
	On board SD WiFi driver(Marvell8686)	Open source, migrated by FriendlyARM
	USB WiFi driver: it comes with the kernel but can only drive limited types	Open source. It comes with the kernel(we don't use it)
	USB WiFi driver: it supports more types	Open source. It comes with the kernel but not configured by default
	Audio driver(WM8960: it supports audio recording and playing, ALSA API and type D amplifier）	Migrated by FriendlyARM
	Ethernet driver(DM9000)	Open source, migrated by FriendlyARM
	FIMC driver	Open source, provided by Samsung
	JPEG driver	Open source, provided by Samsung
	MFC multi-media driver	Open source, provided by Samsung

	CMOS camera driver(OV9650)	Migrated by FriendlyARM
	HDMI driver	Open source, provided by Samsung
	3D accelerator	Open source, provided by Samsung
	2D accelerator	Open source, provided by Samsung
	USB to serial driver	Open source, it comes with the kernel
	3G driver	Open source, migrated by FriendlyARM. It is a USB to Serial driver.
Android System	Version: Android 2.3.1	Open source, Samsung BSP + FriendlyARM customization
Application	2D/3D Acceleration	Good for 2D/3D games
	GPS	External serial port GPS devices
	WiFi	
	CMOS	
	3G Wireless	
	3G Messaging	
	USB Flash Drive	Up to 32G
	USB Blue Tooth	File Transfer
	One Wire Precise Touching	
	Back Light Adjusting up to 127 Levels	
	GUI for Network Configuration	Automatic IP Allocation

3.2 Android 4.0.3 Features

Cross-compiler	arm-linux-gcc-4.5.1-v6-vfp	Same as Mini6410, by default it compiles with armv7 command set. It supports hard floating point arithmetic

Superboot-210	It supports SD card system burning and can install (YAFFS2) systems within 1.8 seconds.	Superboot is especially developed for enterprise users
	It has a graphic interface and can display LCD info, hardware configurations, installation process	
	It can automatically detect MMC/NAND booting mode.	
	It needs the SD-Flasher burning utility which runs on WindowsXP/Vist/Windows7	
Android kernel	Kernel version: Linux-3.0.8	
	It works with YAFFS2/CRAMFS/NFS/UBIFS/NFS/FAT32.	
	Watchdog	
	RTC driver	
	LED driver	
	User button driver	
	SPI driver	
	I2C-EEPROM driver	
	PWM buzzer driver	
	ADC driver (channel: AIN0)	
	Touch screen coontroller driver which comes with CPU by default	
	One wire precise touch driver	
	LCD back light driver: it allows users to adjust the board's backlight up to 127 levels	
	LCD driver(4.3", 5", 7" etc): it supports	

	screen rotation	
	USB Host driver: it supports flash drives, blue tooth and so on.	
	USB Device driver: it supports USB ADB	
	SD card driver	
	Serial port driver	
	On board SD WiFi driver(Marvell8686)	
	USB WiFi driver: it comes with the kernel but can only drive limited types	
	USB WiFi driver: it supports more types	
	Audio driver(WM8960: it supports audio recording and playing, ALSA API and type D amplifier）	
	Ethernet driver(DM9000)	
	FIMC driver	
	JPEG driver	
	MFC multi-media driver	
	CMOS camera driver(OV9650)	
	HDMI driver	
	3D accelerator	
	2D accelerator	
	USB to serial driver	
	3G driver	
Android System	Version: Android 4.0.3	
Application	2D/3D Acceleration	Good for 2D/3D games
	WiFi	
	CMOS	

	3G Wireless	
	3G Messaging	
	HDMI Audio and Video Output	Up to 1080p
	GSM Telephoning	Tested with Huawei EM310
	USB Flash Drive	Up to 32G
	One Wire Precise Touching	
	Back Light Adjusting up to 127 Levels	
	GUI for Network Configuration	Automatic IP Allocation

3.3 Linux Features

Cross-compiler	arm-linux-gcc-4.5.1-v6-vfp	Same as Mini6410, by default it compiles with armv7 command set. It supports hard floating point arithmetic
Superboot-210	It supports SD card system burning and can install (YAFFS2) systems within 1.8 seconds.	Superboot is especially developed for enterprise users
	It has a graphic interface and can display LCD info, hardware configurations, installation process	
	It can automatically detect MMC/NAND booting mode.	
	It needs the SD-Flasher burning utility which runs on WindowsXP/Vist/Windows7	
Linux kernel	Kernel version: Linux-3.0.8	Complete BSP
	It works with YAFFS2/CRAMFS/NFS/UBIFS/NFS/FAT32.	Open source
	Watchdog	Open source, provided by Samsung

	RTC driver	Open source, provided by Samsung
	LED driver	Open source, migrated by FriendlyARM
	User button driver	Open source, migrated by FriendlyARM
	SPI driver	Open source,it comes with the kernel (2011.1.16) but is not verified by FriendlyARM
	I2C-EEPROM driver	Open source, provided by Samsung
	PWM buzzer driver	Open source, migrated by FriendlyARM
	ADC driver (channel: AIN0)	Open source, migrated by FriendlyARM
	Touch screen coontroller driver which comes with CPU by default	Open source, provided by Samsung(we don't use it and use one wire precise touching instead)
	One wire precise touch driver	Open source, migrated by FriendlyARM
	LCD back light driver: it allows users to adjust the board's backlight up to 127 levels	Open source, migrated by FriendlyARM
	LCD driver(4.3", 5", 7" etc): it supports screen rotation	Open source, migrated by FriendlyARM. It drives screens based on the initialization parameter "lcd="
	USB Host driver: it supports flash drives, blue tooth and so on.	Open source, provided by Samsung
	USB Device driver: it supports USB ADB	Open source, provided by Samsung
	SD card driver	Open source, provided by Samsung
	Serial port driver	Open source, provided by Samsung
	On board SD WiFi driver(Marvell8686)	Open source, migrated by FriendlyARM
	USB WiFi driver: it comes with the kernel but can only drive limited types	Open source. It comes with the kernel(we don't use it)
	USB WiFi driver: it supports more types	Open source. It comes with the kernel but not configured by default

	Audio driver(WM8960: it supports audio recording and playing, ALSA API and type D amplifier）	Migrated by FriendlyARM
	Ethernet driver(DM9000)	Open source, migrated by FriendlyARM
	FIMC driver	Open source, provided by Samsung
	JPEG driver	Open source, provided by Samsung
	MFC multi-media driver	Open source, provided by Samsung
	CMOS camera driver(OV9650)	Migrated by FriendlyARM
	HDMI driver	Open source, provided by Samsung
	3D accelerator	Open source, provided by Samsung
	2D accelerator	Open source, provided by Samsung
	USB to serial driver	Open source, it comes with the kernel
	3G driver	Open source, migrated by FriendlyARM. It is a USB to Serial driver.
GUI System	Qtopia-2.2.0	Open source for both x86 and arm
	QtEmbedded-4.8.5	Open source for arm
	Qt-Extended-4.4.3	Open source for Cellphone based Qtopia, also called Qtopia4
Application	The following programs are developed by FriendlyARM and are not open source	
	3G Communication	1) Support more than 100 USB cards for WCDMA, CDMA2000 and TD-SCDMA. 2) Support auto dialing on system startup
	GPRS Messaging	Support both GPRS Modem or USB 3G messaging
	HDMI	Support HDMI resolution settings, auto HDMI output on system startup
	ADC Conversion	

	LED Control	
	User Button Test	
	I2C-EEPROM reading and writing	
	LCD Test Utility	
	Ping Test Utility	
	USB Camera Utility	
	CMOS Camera Utility	
	Audio Recorder	
	Web Browser	
	Watchdog	
	Network Configuration Utility	
	Backlight Control Utility	
	Language Setting	
	Calibration Utility	
	Qt4 Switch Utility	
	Qtopia4 Switch Utility	
	SMPlayer	

3.4 WinCE6 Features

Version	WindowsCE Embedded 6.0	
Superboot-210	It supports SD card system burning and can install (YAFFS2) systems within 1.8 seconds.	Superboot is especially developed for enterprise users
	It has a graphic interface and can display LCD info, hardware configurations, installation process	

	It can automatically detect MMC/NAND booting mode.	
	It needs the SD-Flasher burning utility which runs on WindowsXP/Vist/Windows7	
BSP Feature	Rapid System Booting (in 8 seconds)	
	Use Configurable bootlogo with SD card buring	
	RTC driver	
	LED driver	
	User button driver	
	PWM buzzer driver	
	Touch screen controller driver which comes with CPU by default	
	One wire precise touch driver	
	LCD back light driver: it allows users to adjust the board's backlight up to 127 levels	
	LCD driver(4.3", 5", 7" etc)	
	USB Host driver: it supports flash drives, blue tooth and so on.	
	SD card driver	
	Serial port driver	
	Audio driver(WM8960）	
	Ethernet driver(DM9000)	
	HDMI driver	
Application	The following programs are developed by FriendlyARM and are not open source	
	HDMI	Support HDMI resolution settings, auto HDMI

		output on system startup
	LED Control	
	User Button Test	
	Audio Recorder	
	Watchdog	
	Network Configuration Utility	
	Backlight Control Utility	
	PWM Buzzer	
	Serial Assistant	

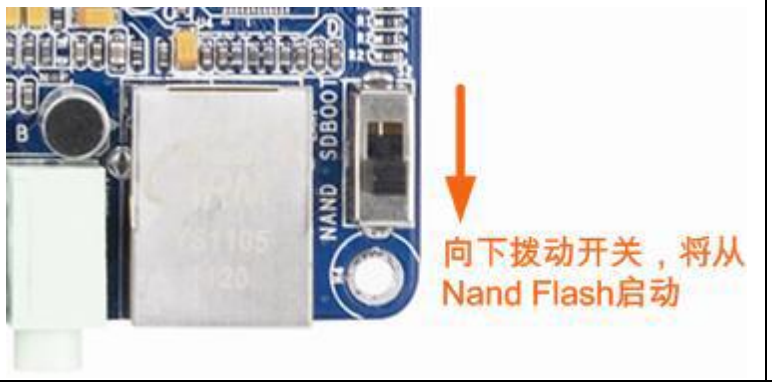
4 Getting Started

By default, all our systems have been preinstalled with Android 4.0 (located in the shipped CDs' directory /images/Android are superboot, zImage, root_android.img and so on) therefore you can easily boot the board and play.

4.1 System Setup and Configurations

4.1.1 Boot Option

The 210 boards support booting from either SD card or Nand Flash. Users can switch between by toggling the S2 switch:

Screenshot	Operation	Comment
	Toggle S2 to "SDBOOT" and the board will boot from the **SD card**	This is for system booting or burning
	Toggle S2 to "NAND" and the board will boot from the **Nand Flash**	Default and Standard booting

Usually, S2 is switched to the Nand Flash side unless users need to boot from the SD card or reflash the system.

4.1.2 Hardware Connection

Please follow the steps below to hook up the board:

- Connect a 210 board's serial port0 (Debug Serial Port) to a PC's serial port with the shipped crossover serial cable (blue one) in the package
- Connect the 210 board's Ethernet interface to a PC with the shipped crossover cable (this step can be skipped if you don't need to connect to the internet)
- Connect the shipped 5V power supply adapter to the 5V power supply interface on the board (do it with care to prevent damaging the interface)
- Connect a headphone or speaker to the audio input(green) on the board
- Connect an LCD touch screen (if the user has one) to the LCD interface on the board following the data bus' arrow

4.1.3 Setting up Super Terminal

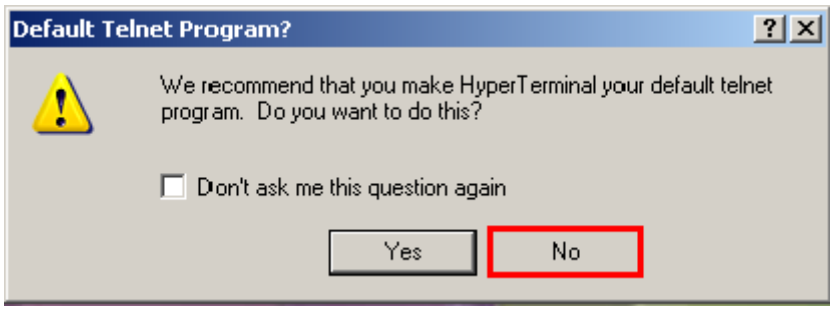
Note: some users attempt to expand the board's serial ports by using a USB to serial port cable. Sometimes this operation would cause error codes. This might be an indication that the cable doesn't work. Most of our agents have these conversion cables for sale. Users can contact them. In addition we strongly recommend users to use our shipped crossover serial cable. Other serial cables might not work properly.

To connect the 210 board to a host PC via a serial cable, you should use a simulation terminal. There are many tools available. A most widely used one is the MS-Windows'

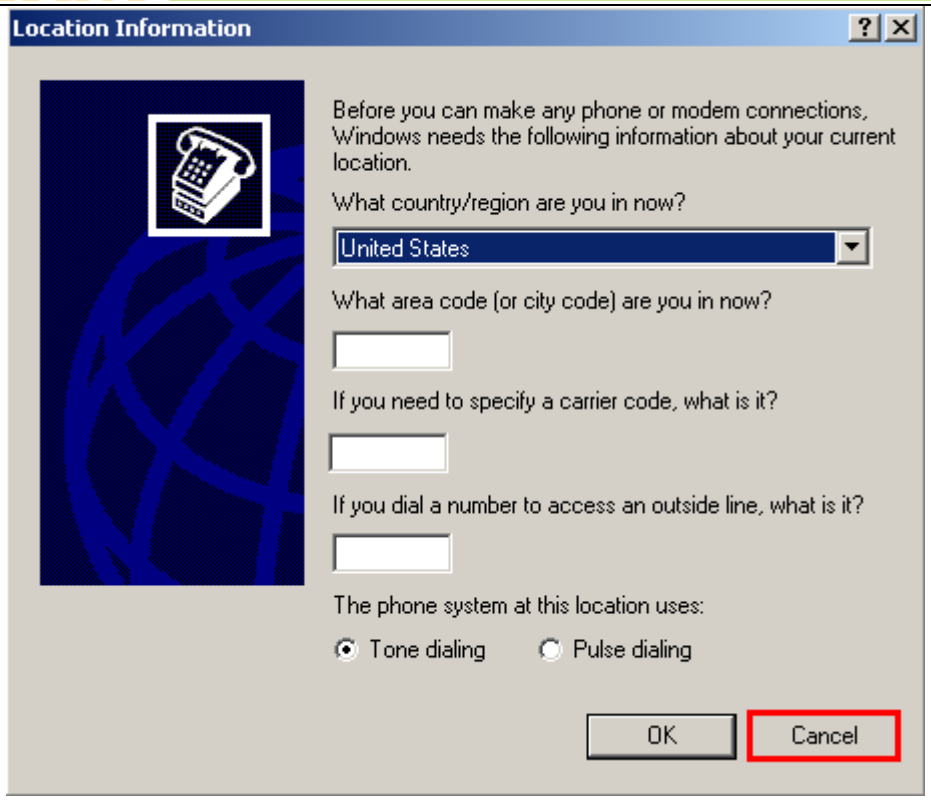
super terminal. In Windows9x, you need to install it by checking that option during installation. Windows2000 and later

A common Linux desktop version has a similar terminal too and it is minicom. It is a command line utility which may not be easy for beginners. Interested users can search the internet for more resources.

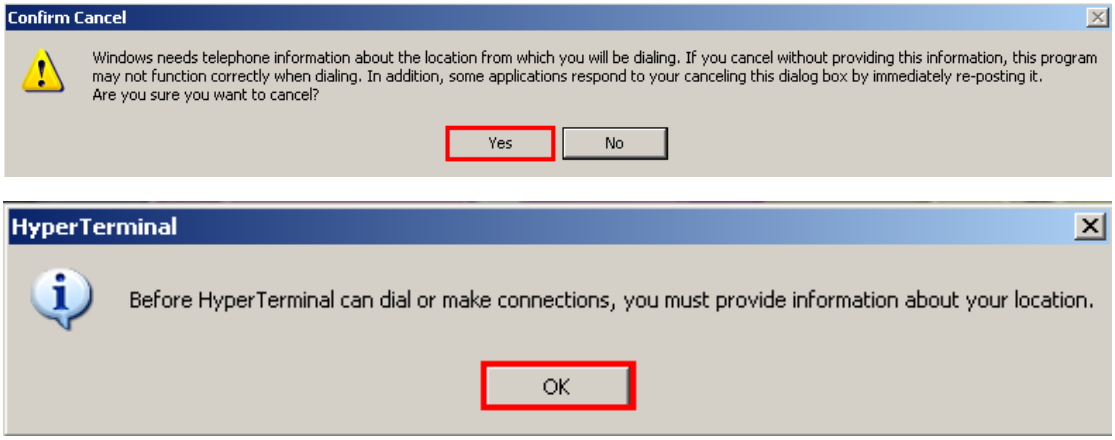
We take WindowsXP's super terminal for instance. You can find it by going to "Start->Programs->Accessories->Communications". After it starts the following dialog will pop up, please click on the "No" button



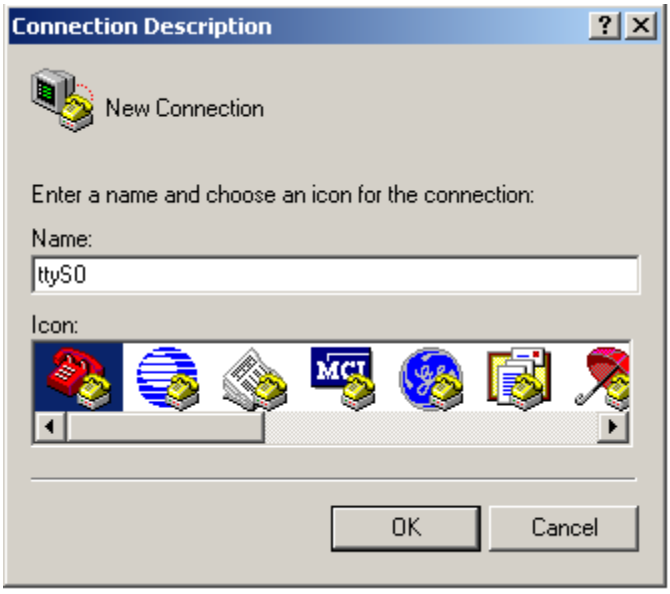
Click on the "Cancel" button on the following dialog



Click on the “Yes” button and the “OK” button to the next step

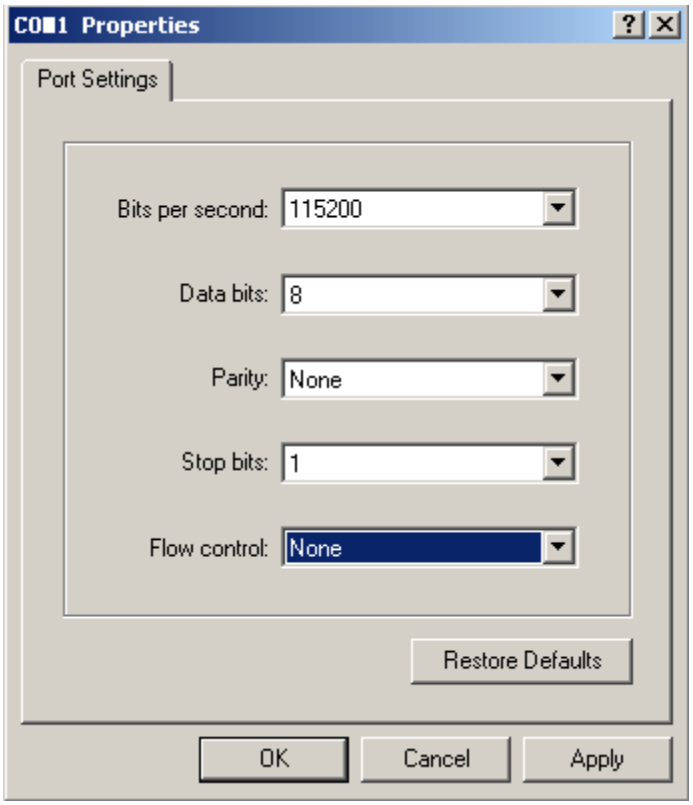


A popup window will require you to name this connection. In this example we typed “ttyS0”. Windows does not accept names like “COM1” that have already been used by the system.



After naming this connection another window will require you to select a serial port that will be used to connect the Tiny210 board. Here we selected COM1:

Lastly, also the most important step is to set up the port properties. Note: you must select "No" in the data flow control field otherwise you will only be able to see outputs. In addition the bits per second should be set to 115200.



After setting up all properties, turn on the board's power supply, if the connection gets set properly, you will see a bootloader startup interface. If everything runs fine please save this connection for later use.

4.2 Burning Superboot to SD Card

In order to boot from an SD card, you need to burn BIOS to it. FriendlyARM offers a flashing utility: SD-Flasher.exe which can burn our Superboot-210 to an SD card. Since Superboot-210 can detect an SD card and a NAND Flash it can boot both the SD card and the NAND Flash.

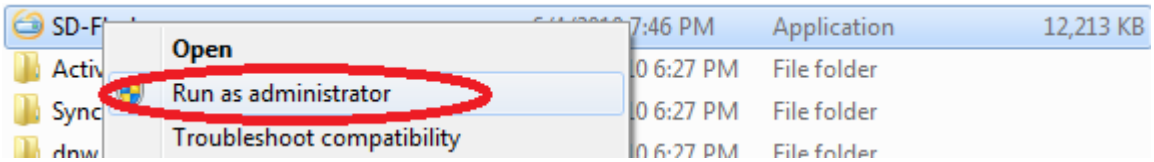
4.2.1 Burning Superboot to SD Card

We tested the following steps on Windows7

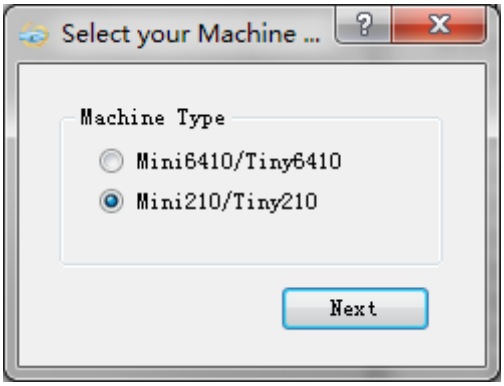
Note: users complained that some notebook's integrated SD card reader cannot work properly with card burning or reading. So far we haven't encountered this issue and we suggest that you should try a common card reader in this case.

Our SD-Flasher.exe formats a 130M space for the bootloader therefore an SD card whose memory is less than 256M cannot work and we recommend using one whose memory is at least 4G

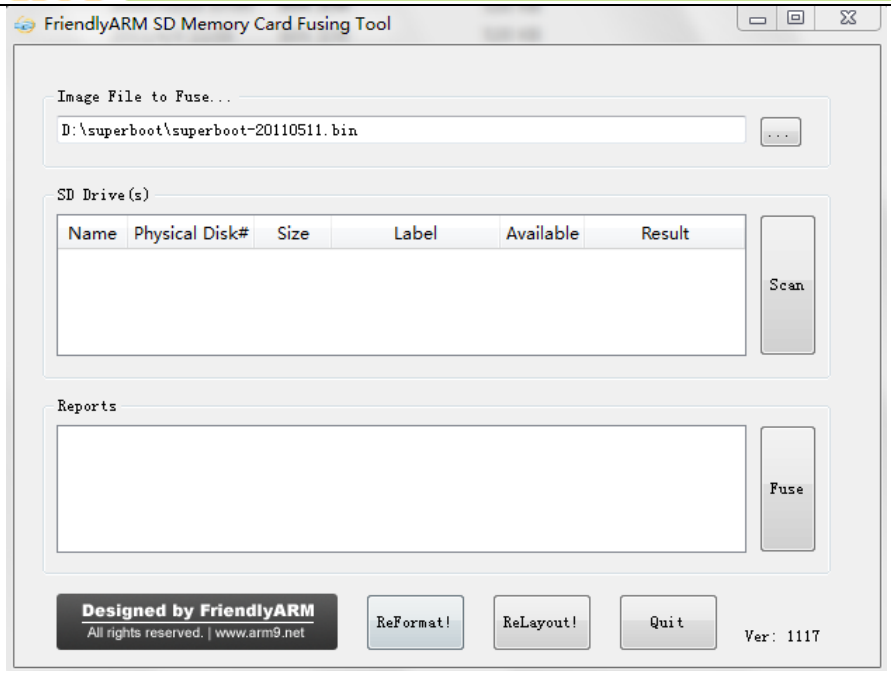
Step1: launch the SD-Flasher.exe in your shipped CD (under "\tools\"). Note: this program should be run under "administrator"



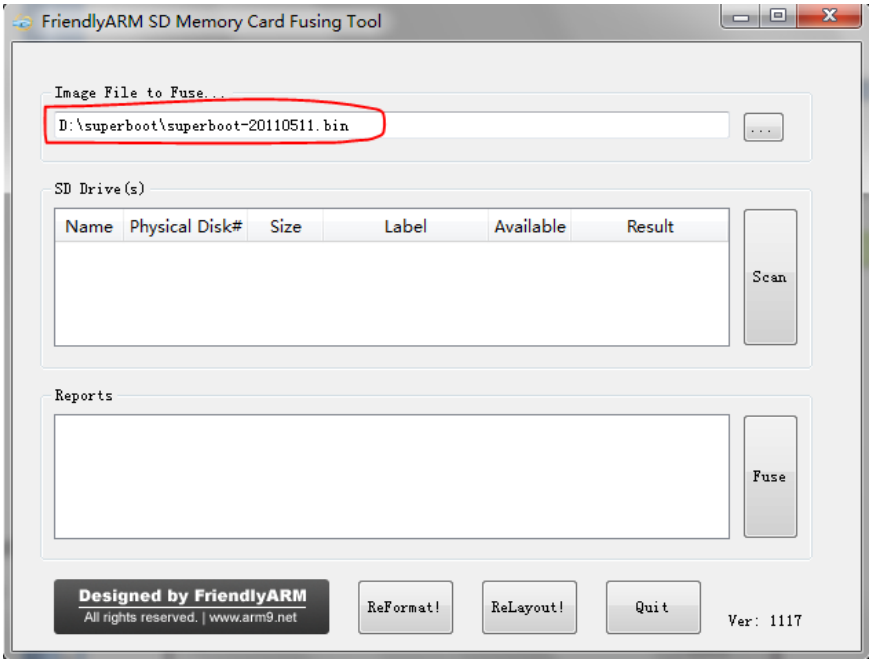
When the utility is launched a message box will pop up "Select your Machine…", please select "Mini210/Tiny210":



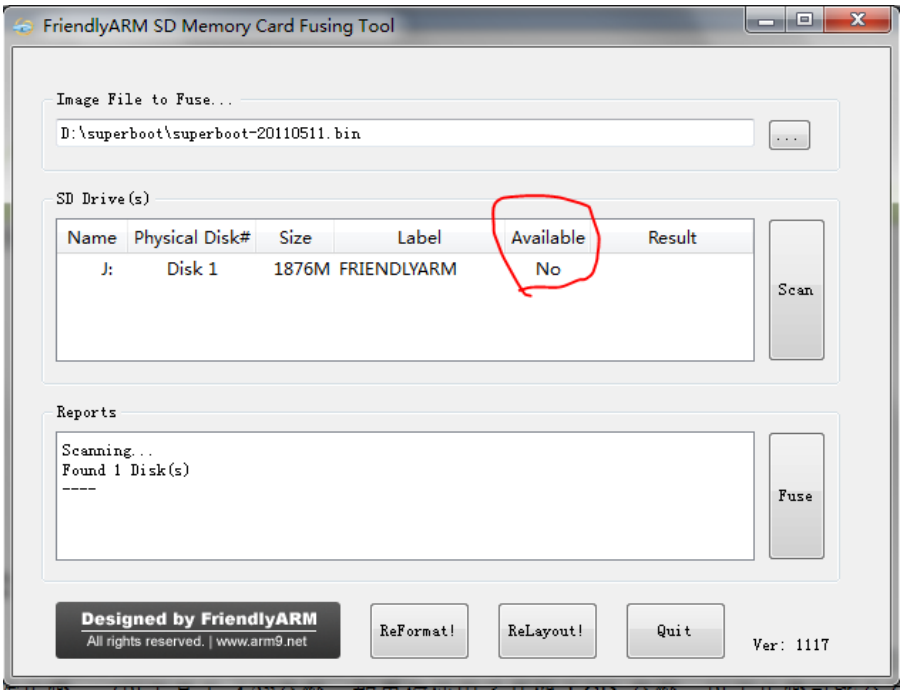
Below is the dialog you will see after it is started. Note: the "ReLayout" is enabled and we will format the SD card with this function.



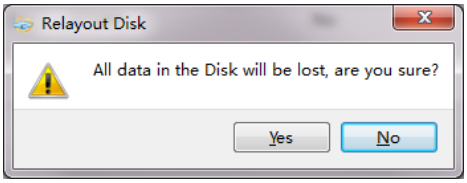
Step2: click on  to select your Superboot file



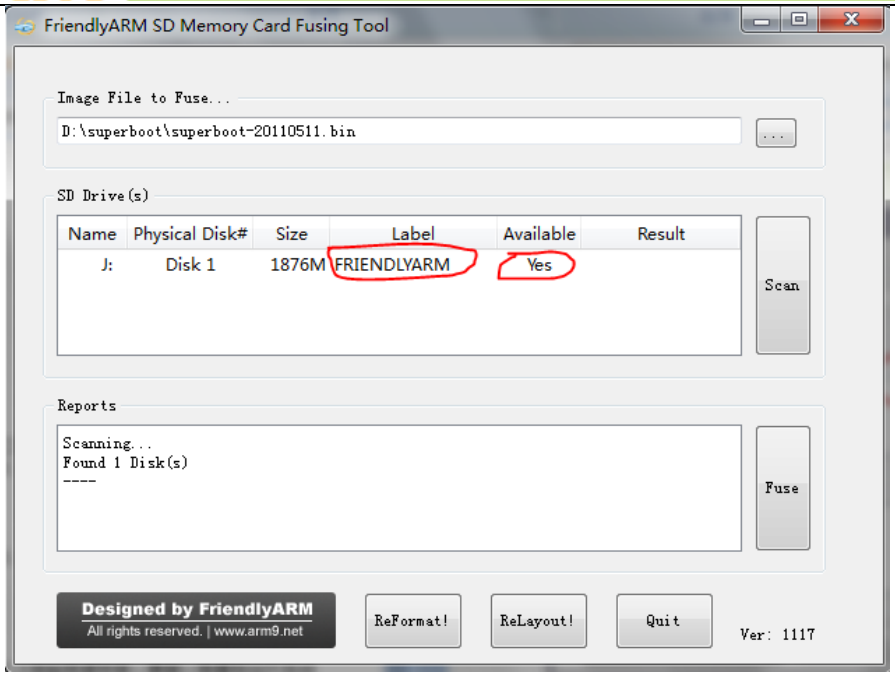
Step3: insert a FAT32 SD card into your host's SD card socket (you can also use a USB card reader to connect to a PC), backup your data in the card and click on "Scan", all recognized SD cards will be listed. For now, the SD card cannot be burned (circled by red)



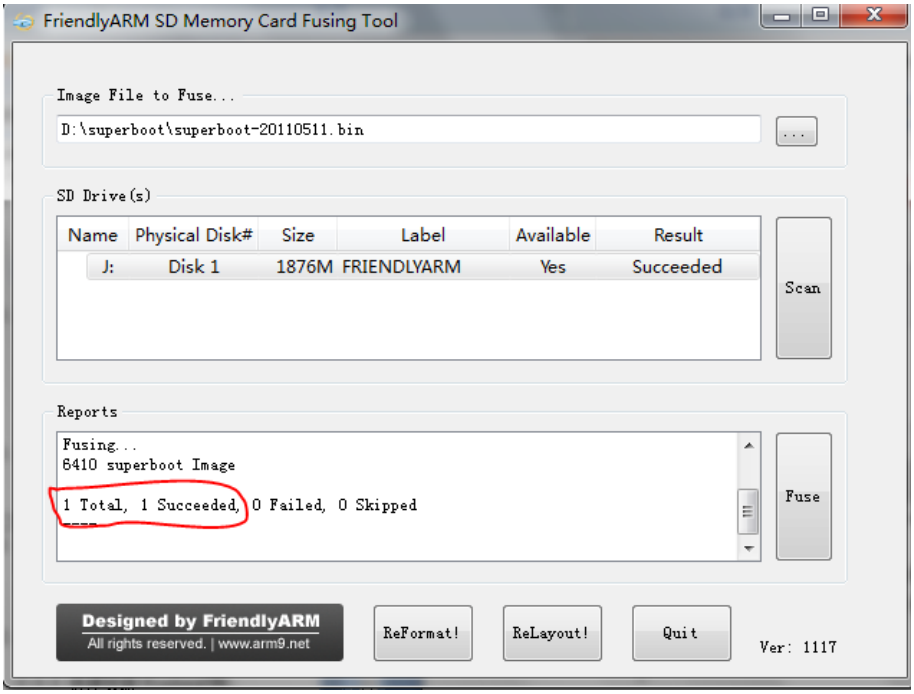
Step4: click on "ReLayout", the following dialog will pop up prompting you that the data in your card will be lost. Just click on "Yes"



After formatting is done you will be directed back to the main menu. Click on "Scan", you will see that a "FriendlyARM" section available.



Step5: click on “Fuse”, Superboot will be safely burned into the SD card. You can burn this card in WindowsXP without worrying about its FAT32 data being lost or damaged.



The Superboot in your SD card is invisible. To verify it you can insert your SD card

into your board's SD card socket and switch S2 to the "SDBOOT" mode, reboot your board and if LED1 is flashing it is indicating that your Superboot is functioning.

If you don't see LED1 flashing or any output from your serial port it may indicate your burning was not successful. The following cases could result in this failure:

1. You might use a notebook and the notebook's the card reader might not work. We suggest using an external usb card reader.

2. You might use a bad SD card. We suggest using one whose memory is at least 4G or SDHC

3. You might use an SD card. Please use a standard SD card which can be directly connected to your card reader

4. The SD card booting function is integrated in Samsung's chip and the ROM is preinstalled. It might not recognize some cards. In this case we suggest you try some different cards

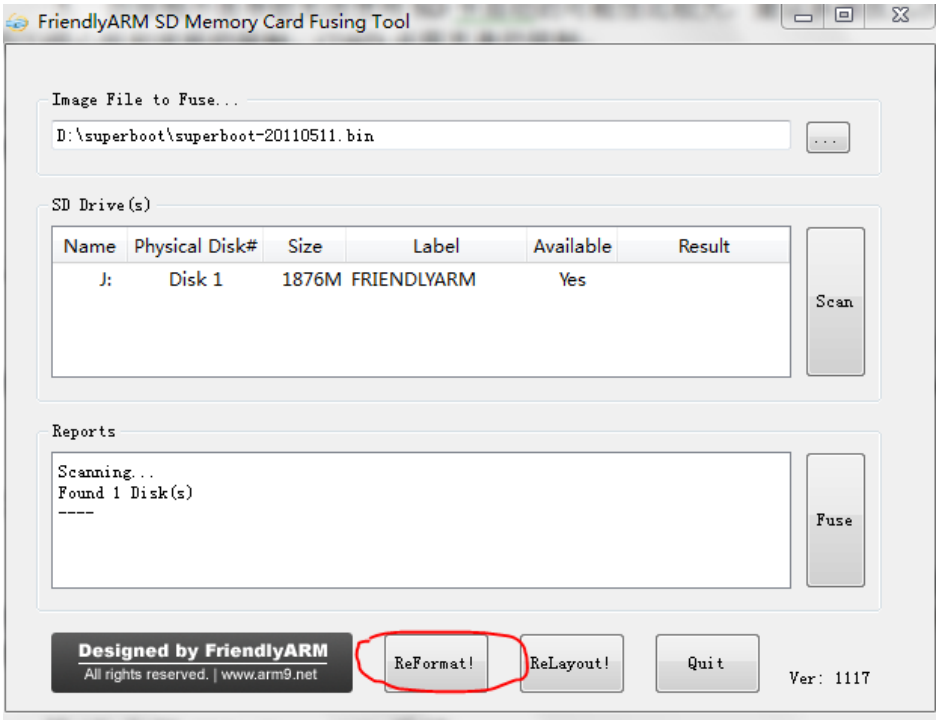
5. Poor contact might be another reason. In this case you could try a few more times: by unplugging and plugging the core board and the base board (if your board is a tiny board) and unplugging and plugging the SD card

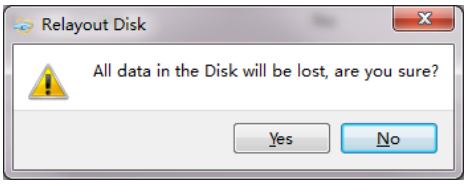
4.2.2 Restore SD Card

Note: we tested the following steps on Windows7

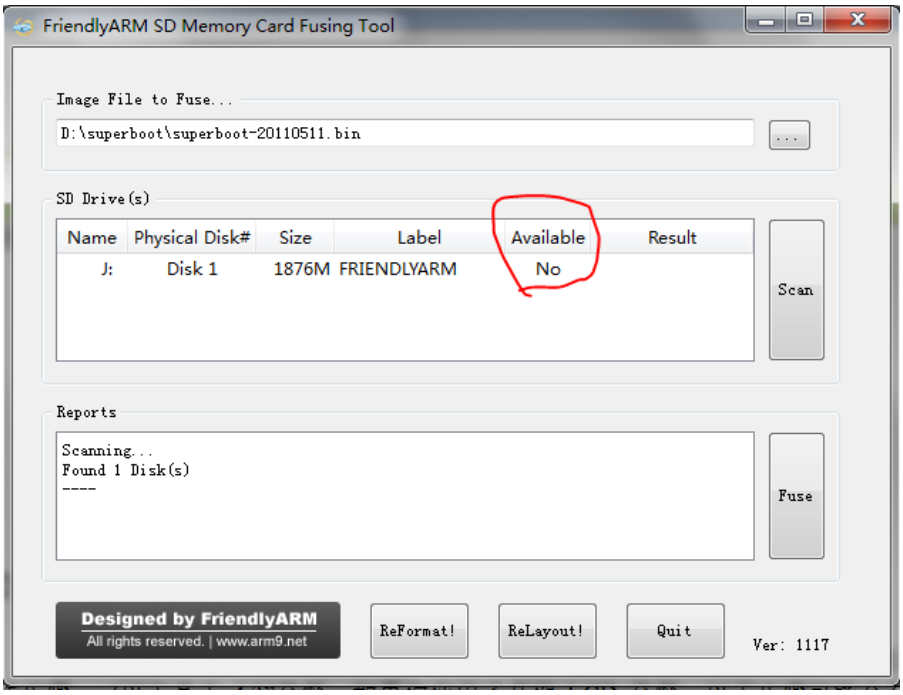
Using SD-Flasher.exe will reserve 130M memory for Superboot. When you no longer need your SD card for system burning you might want to restore your card to what it

was. You can do it this way: launch SD-Flasher.exe as an administrator; click on “scan” and “ReFormat” you will see the following dialog





Click on “Yes”. A moment later click on “Scan” again, you will find your card becomes “no” available and your card is restored successfully.



4.2.3 Notes to Users

Common SD cards are used as storage cards therefore on Vista/Windows7 SD-Flasher automatically formats an SD card to two sections: one is FAT32 (named “FriendlyARM”) and the other (by default 130M) reserved for the bootloader.

In fact, Vista/Windows7’s system security policies don’t permit unauthorized users to start auto burning an SD card thus common users need to format the SD card first and then burn data into it. For WindowsXP users we just set the burning mode to auto burning, the same as what Samsung does

4.2.4 Configuring FriendlyARM.ini

When installing systems you will need the “FriendlyARM.ini” file. Its content is as follows:

FriendlyARM.ini File

```
#This line cannot be removed. by FriendlyARM(www.arm9.net)

CheckOneButton=No
Action = Install
OS = Android
LCD-Mode = No
LCD-Type = S70
LowFormat = Yes
VerifyNandWrite = No
CheckCRC32=No
StatusType = Beeper | LED
#################### Android 4.0.3 ####################
Android-BootLoader = Superboot210.bin
Android-Kernel = Android/zImage
Android-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc
androidboot.console=s3c2410_serial0
Android-RootFs-InstallImage = Android/rootfs_android.img
#################### Android 2.3.1 ####################
#Android-BootLoader = Superboot210.bin
#Android-Kernel = Android2.3.1/zImage
#Android-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc
androidboot.console=s3c2410_serial0
#Android-RootFs-InstallImage = Android2.3.1/rootfs_android.img
#################### Linux ####################
Linux-BootLoader = Superboot210.bin
Linux-Kernel = Linux/zImage
Linux-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc
Linux-RootFs-InstallImage = Linux/rootfs_qtopia_qt4.img
#################### Windows CE6.0 ####################
WindowsCE6-Bootloader = Superboot210.bin
WindowsCE6-BootLogo = WindowsCE6\bootlogo.bmp
WindowsCE6-InstallImage = WindowsCE6\NK.bin
WindowsCE6-RunImage = WindowsCE6\NK.bin
```

We listed the details of each item in the table below:

Item	Comment: the default configurations are different for different systems
CheckOneButton	= “Yes”, users need to press any button to resume system booting = “No”, system will reboot completely after it is reset or powered on. For mass burning

	this item is usually set to "No" The default setting is "No"
Action	Set actions: Install/Run/Null Install – Install to the NAND Flash Run – Run from SD card Null – No action The default setting is "Install"
OS	Operating system to be loaded: Linux/WindowsCE6/Ubuntu/Android/UserBin ; "UserBin" means independent programs or single file image such as uCos2 and Rt-Thread. The default setting is "Android"
VerifyNandWrite	= "yes", system will verify after burning is done. This is more reliable ; = "No", system will not verify, this takes less time. The default setting is "No"
LowFormat	Perform low level formatting on the NAND Flash The default setting is "Yes"
StatusType	Status of the burning process: "LED","Beeper" and "LED|Beeper" The default setting is "LED|Beeper"
Items to specify Android images, they can include directories and "/" or "\"	
Android-BootLoader	Bootloader file The default setting is Android-BootLoader =superboot-210.bin
Android-Kernel	Kernel image The default setting is Android-BootLoader=Android/zImage
Android-CommandLine	Boot arguments - When using the yaffs2 system the suggested (default) commandline is: Android-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc androidboot.console=s3c2410_serial0 Note: if you want to skip calibration you can add "skipcali=yes" in the command line - When running from SD card use the default setting - When you want to run your board with a capacitive LCD you need to specify "skipcali=yes" and "ctp=n". The value "n" can be either 0 or 1, 2, and 3. If n=0 it means no capacitive LCD will be connected. If n=1 it means a GT80X based 7' capacitve touch panel will be connected. If n=2 it means a FT5206 based 7" capacitive touch panel will be connected. If n=3 it means a FT5406 based 7" capacitive touch panel will be connected.
Android-RootFs-InstallImage	File system image used to be installed, now only the yaffs2 system is provided. The default setting is Android-RootFs-InstallImage = Android/rootfs_android.img
Android-RootFs-RunImage	File system image used to be run directly from SD card The default setting is Android-RootFs-RunImage = Android/rootfs_android.ext3
Items to specify Linux images, they can include directories and "/" or "\"	
Linux-BootLoader	Bootloader file

	The default setting is Linux-BootLoader =superboot-210.bin
Linux-Kernel	Kernel image The default setting is Linux-BootLoader=Linux/zImage
Linux-CommandLine	Boot arguments - When using the yaffs2 system the suggested (default) commandline is: Linux-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc Note: if you want to skip calibration you can add "skipcali=yes" in the command line - When running from SD card use the default setting - When you want to run your board with a capacitive LCD you need to specify "skipcali=yes" and "ctp=n". The value "n" can be either 0 or 1, 2, and 3. If n=0 it means no capacitive LCD will be connected. If n=1 it means a GT80X based 7' capacitve touch panel will be connected. If n=2 it means a FT5206 based 7" capacitive touch panel will be connected. If n=3 it means a FT5406 based 7" capacitive touch panel will be connected.
Linux-RootFs-InstallImage	File system image used to be installed, now only the yaffs2 system is provided. The default setting is Linux-RootFs-InstallImage = Linux/rootfs_android.img
Linux-RootFs-RunImage	File system image used to be run directly from SD card The default setting is Linux-RootFs-RunImage = Linux/rootfs_android.ext3
Items to specify WindowsCE6 images, they can include directories and "/" or "\"	
WindowsCE6-BootLoader	Bootloader file The default setting is Linux-BootLoader =superboot-210.bin
WindowsCE6-BootLogo	Boot Logo, BMP file, 24 bit color The default setting is WindowsCE6-BootLogo=WindowsCE6\bootlogo.bmp
WindowsCE6-InstallImage	File system image used to be installed The default setting is WindowsCE6-InstallImage = WindowsCE6\NK.bin
WindowsCE6-RunImage	File system image used to be run directly from SD card The default setting is WindowsCE6-RunImage = WindowsCE6\NK.bin

Notes:

1. Statements after "#" will not be executed by Superboot. Actually any character except key words can be used to comment. "#" is just widely accepted

2. To prevent our Superboot from being illegally copied we make it a rule that the first line of the ini file cannot be edited or deleted. It is:

#This line cannot be removed. by FriendlyARM(www.arm9.net)

Note: no space or any other character after the last ")" is allowed

4.3 Install Systems with Minitools

The Minitools utility is a FriendlyARM developed USB download tool which allows users to install systems more easily and conveniently. It has the following features:

- Only need a USB cable: with the Minitools users only need a USB cable to install systems
- One key action: no need to type any command.
- Works with both 32/64-bit OS: it can be installed on both 32-bit and 64-bit Windows systems
- Cross platform: it can be installed on both Windows and Linux systems

4.3.1 Install Minitools

4.3.1.1 Install on Windows

Double click on the "MiniToolsSetup.exe" icon in the tools directory in your shipped DVD and you will be guided to install it. Just follow the prompts and take the default options. When it asks whether you want install the driver please go by "continue anyway". After installation is done please unplug and plug the USB cable and Windows will prompt that it is updating drivers. After Windows' updating is done you can continue

If your installation is successful there will be an icon on your desktop. You can double click on it to run:

The minitools' main window is shown below:

4.3.1.2 Install on Linux

We tested installing the Minitools on Fedora9/Fedora15/Ubuntu12.04 64-bit systems. Please login and execute the installation as root. Please copy the "MiniTools-Linux-YYYYMMDD.tgz" in the "tools" directory from your DVD to your PC and untar the ball and run the "./start.sh" command to the installation.

4.3.2 Flash Superboot to SD Card

In order to work with the Minitools you need to get an SD card and flash our superboot to it. Please follow the steps below:

1. Please flash the superboot to an SD card with “SD-Flasher”
2. Please copy the whole “images” directory from your DVD to the root directory of your SD card
3. Open the “images/friendlyARM.ini” and add the following line

   USB-Mode = yes

Please follow the steps below to connect your board to your PC

1. Switch the S2 on your board to “SD”
2. Power on the board and you will see the LCD showing “USB Mode:Waiting” if everything works correctly
3. Please connect your board to your PC via a USB cable
4. If the connection is successful the LCD will show “USB Mode:Connected”

Now you can start installing systems with the Minitools

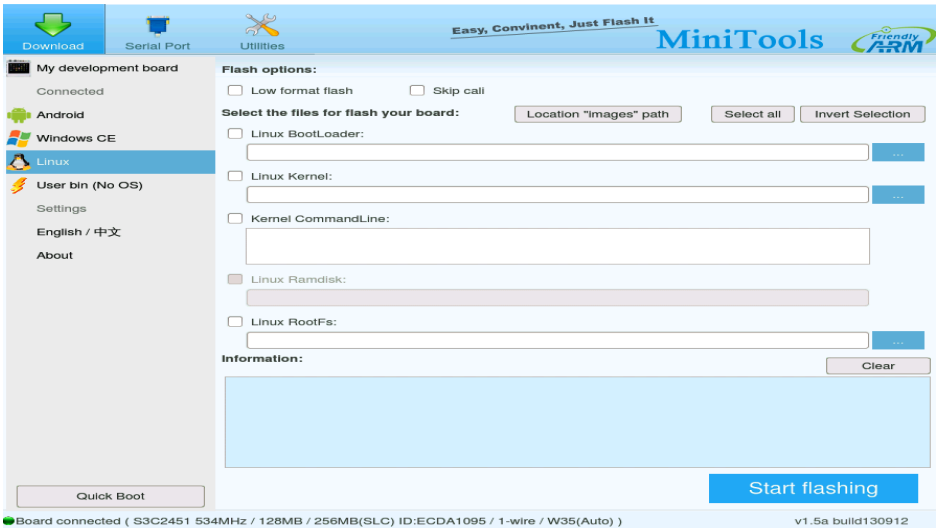
To change the installation method back to SD card installation you just need to change the “USB-Mode = yes” to “USB-Mode=no”.

4.3.3 Install Systems with Minitools

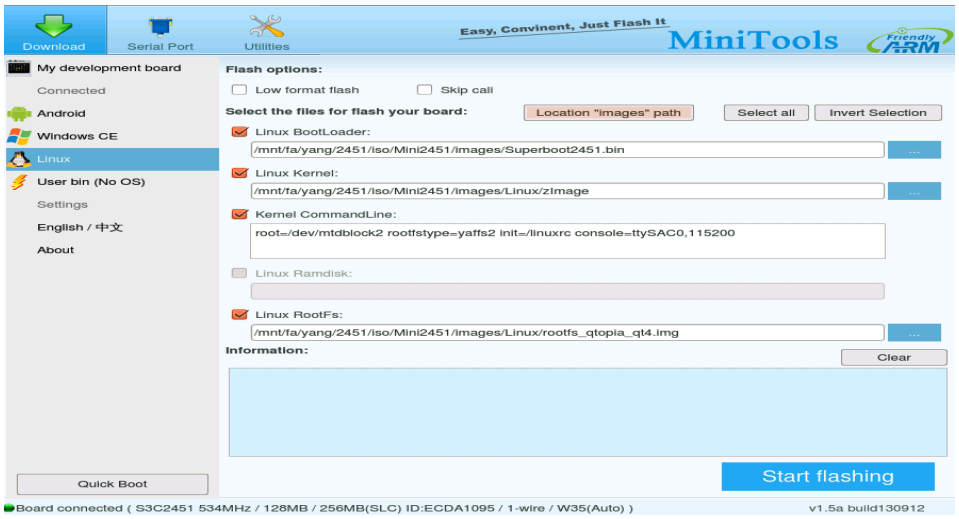
Please enter the USB download mode and connect your board to your PC which runs the Minitools via USB

On the left bottom of the window there is an LED which is green indicating the board is connected successfully. On the left bottom there is a button which can start your board directly without switching to NAND.

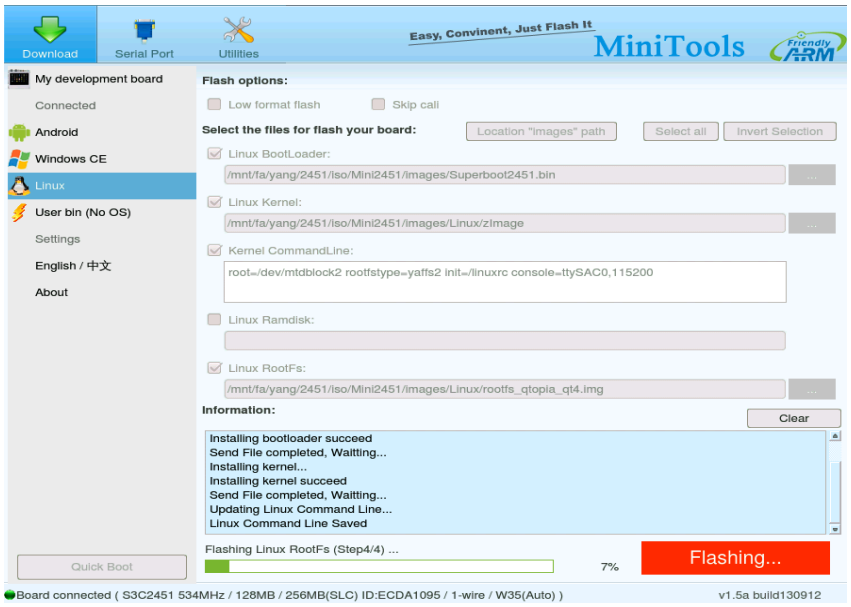
Before install systems please select the system you want to install e.g. Linux and then its configuration will be presented as follows:



You can just click on the "images" button to select an "images" directory which contains complete installation files for all systems and the Minitools will show all the info listed in the FriendlyARM.ini.

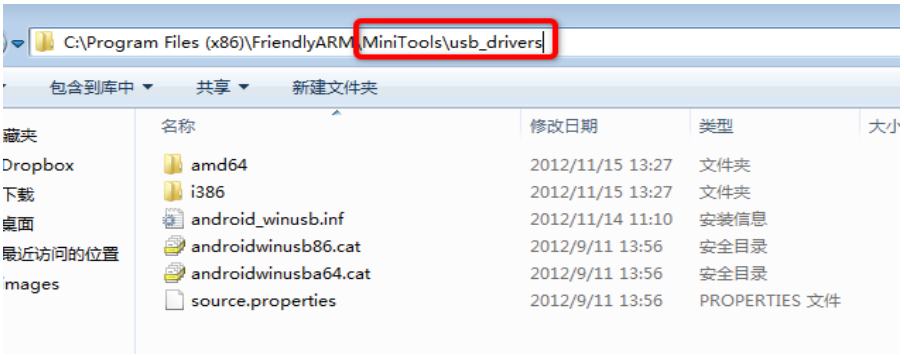


With the Minitools utility you can update either the whole system (all image files) or individual image files e.g. the kernel image file. After you are done with your installation configuration please click on "Start flashing"



After installation is done you can boot your board and enter your system.

Note: sometime users complain that Minitools shows the board isn't connected to PC. It is very likely that the USB download driver is not properly installed on your PC and you can try manually install the USB download driver which is under the Minitools directory in the shipped DVD



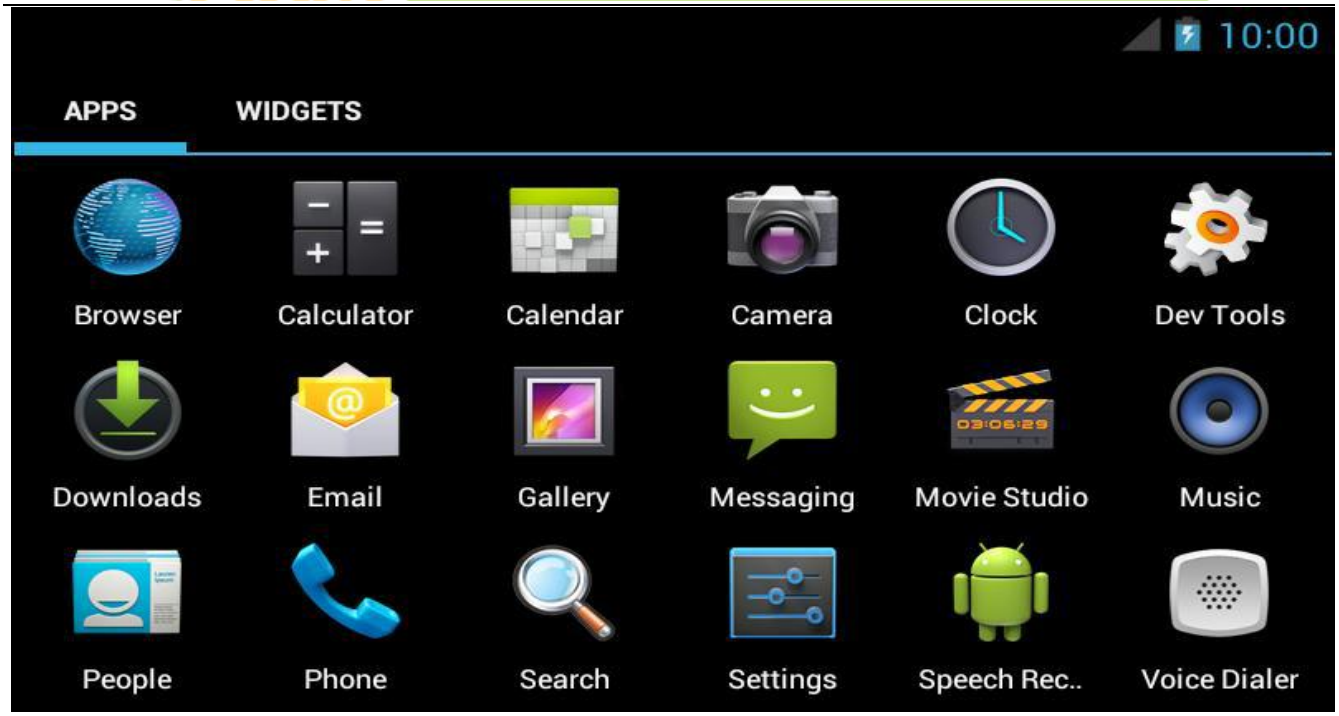
5 Android Installation and Navigation

The 210 boards can run both Android2.3 and Android4. We migrated all the utilities we developed for the Mini6410 to the 210 boards. This not only meets most customers' requirements but also enables users to focus on application development.

Here is a table which lists all the software features the Tiny210 offers:

2D/3D Acceleration	1080P High Definition Video Playing	HDMI Output	3G Dial-Up	3G Messaging
CMOS Camera	Bluetooth	SD-WiFi	USB WiFi	Ethernet
GPS	iTest	Low Level Hardware Access Libraries	ADB Debug	Flash Drive Plug and Play
Backlight Control	Flash Wallpaper	Audio Recording/Playing		

Android 4.0.3 Screenshot



5.1 Installing and Playing with Android

5.1.1 Installing Android 4.0.3

Note: before read the following sections please burn Superboot to your SD card and copy corresponding installation files to your card.

Step1: insert the SD card to a PC, open the "images\FriendlyARM.ini" file and modify it as follows:

```
#This line cannot be removed. by FriendlyARM(www.arm9.net)

CheckOneButton=No
Action=Install
OS=Android
VerifyNandWrite=No
low-format=No
LCD-Mode = No
LCD-Type = S70
Check CRC32=No
Status Type = Beeper| LED
```

```
#################### Android 4.0.3 ####################
Android-BootLoader = superboot210.bin
Android-Kernel = Android/zImage
Android-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc
androidboot.console=s3c2410_serial0
Android-RootFs-InstallImage = Android/rootfs_android.img
```

(Note: in "Android-CommandLine" the "lcd" parameter specifies the LCD type which currently has: H43, W50, A56, S70, A70, L80 and G10)

Step2: make sure your card has the following files (actually you can copy the whole image directory to your SD card's root directory)

File	Comment
images\superboot210.bin	Bootloader. It can boot Android and other OS such as Linux and WinCE. It can be run from an SD card.
images\Android\zImage	Android kernel. It can automatically detect LCD types
images\Android\rootfs_android.img	Android file sysem image
images\FriendlyARM.ini	Configuration file

Step3: insert the SD card to the board's **SD** socket and switch S2 to the SD side. Power on the board and you will hear a beep and see a progress bar on the LCD.

Step4: after system burning is done you will hear two continuous beepings and the LCD will show the burning status. Switch S2 to the Nand Flash side, reboot the system and

you will see Android loads.

5.1.2 Installing Android 2.3

Note: before read the following sections please burn Superboot to your SD card and copy corresponding installation files to your card.

Step1: insert the SD card to a PC, open the "images\FriendlyARM.ini" file and modify it as follows:

```
#This line cannot be removed. by FriendlyARM(www.arm9.net)

CheckOneButton=No
Action=Install
OS=Android
VerifyNandWrite=No
low-format=No
LCD-Mode = No
LCD-Type = S70
Check CRC32=No
Status Type = Beeper| LED

#################### Android 2.3.1 ####################
Android-BootLoader = superboot210.bin
Android-Kernel = Android2.3.1/zImage
Android-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc
```

```
androidboot.console=s3c2410_serial0
Android-RootFs-InstallImage = Android2.3.1/rootfs_android.img
```

(Note: in "Android-CommandLine" the "lcd" parameter specifies the LCD type which currently has: H43, W50, A56, S70, A70, L80 and G10)

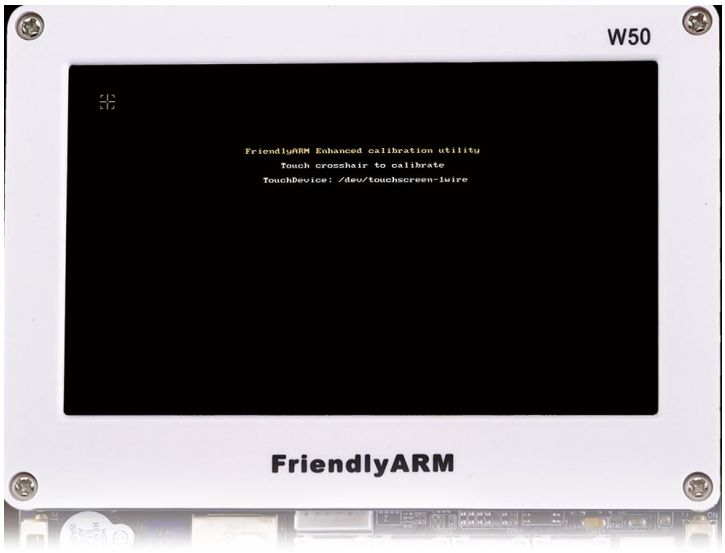
Step2: make sure your card has the following files (actually you can copy the whole image directory to your SD card's root directory)

File	Comment
images\superboot210.bin	Bootloader. It can boot Android and other OS such as Linux and WinCE. It can be run from an SD card.
images\Android2.3.1\zImage	Android kernel. It can automatically detect LCD types
images\Android2.3.1\rootfs_android.img	Android file sysem image
images\FriendlyARM.ini	Configuration file

Step3: insert the SD card to the board's **SD** socket and switch S2 to the SD side. Power on the board and you will hear a beep and see a progress bar on the LCD.

Step4: after system burning is done you will hear two continuous beepings and the LCD will show the burning status. Switch S2 to the Nand Flash side, reboot the system and you will see Android loads.

If you are running Android for the first time you will see the following calibration screen:



Click on "+", follow it till the end position and Android will resume. After it is completely booted you will see the following screen:

User button layout:

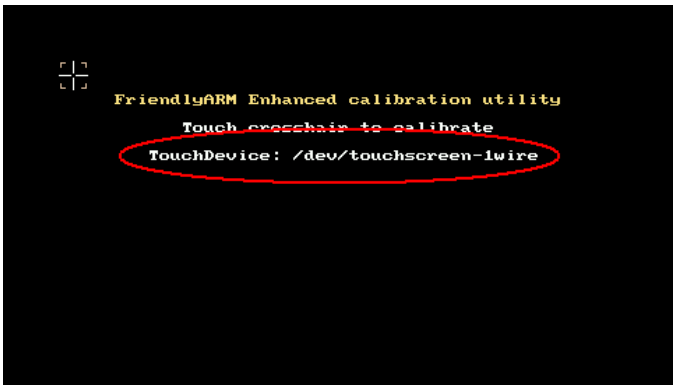
The 210 boards has 4 buttons:

Key	Function
K1	**Back**
K2	**Home**
K3	**Menu** and **Screen rotation**
K4	**OK**

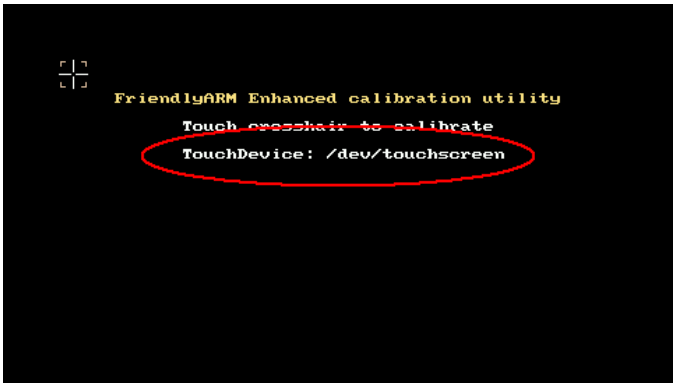
5.2 Playing with Android

5.2.1 Calibrate Touch Screen

After you burn Android into your board you will see a calibration screen on the very first system boot. The following screen shows the system uses a one wire precise touching LCD: /dev/touchscreen-1wire (marked in red).



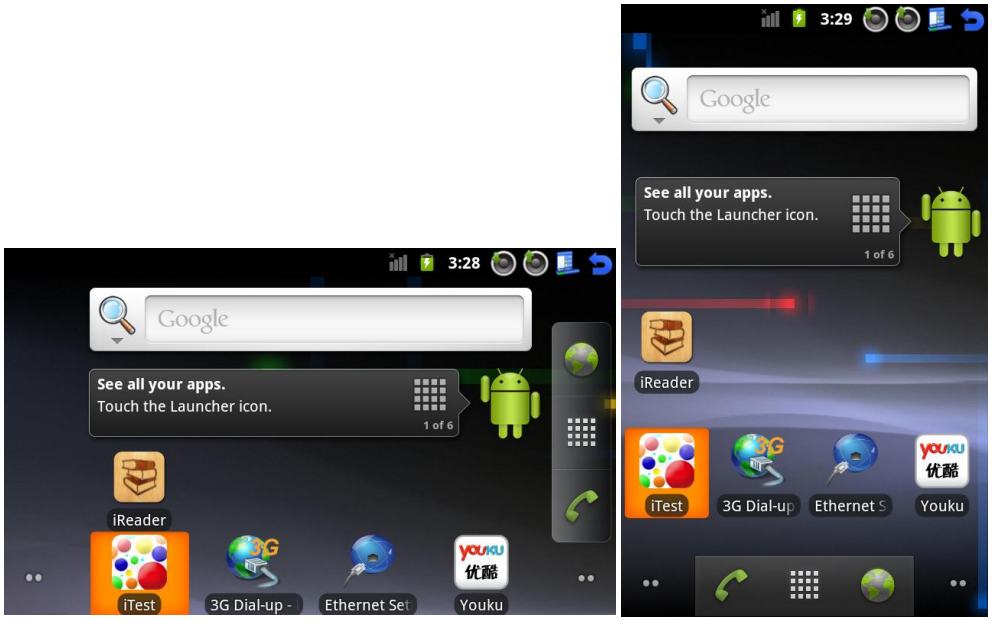
The following screen shows the system uses an ARM LCD: /dev/touchsreen (marked in red)



Click on "+" and follow it to calibrate and you will enter the system after your calibration is done. If you don't position your pen properly the calibration process will restart until you are done successfully.

5.2.2 Rotate Touch Screen

After Android 2.3 is loaded by default it will display horizontally. To switch to the vertical presentation please long-press the menu key (k3) and it will rotate.



5.2.2.1 Vertical Display/Horizontal Display

By default the board displays horizontally or vertically. Users can change that by following the steps below.

Firstly please open the “init.rc” in the root directory, search for “ro.sf.hwrotation” and remove the “#”. This sets the rotation angle to 270 degrees.

setprop ro.sf.hwrotation 270

If your OS is Android 2.3.1 after you make that change it will take into effect on system rebooting.

If your OS is Android 4.0 you need one more step: open the file “/system/build.prop” change the following line:

ro.sf.auto_lcd_density=yes

to:

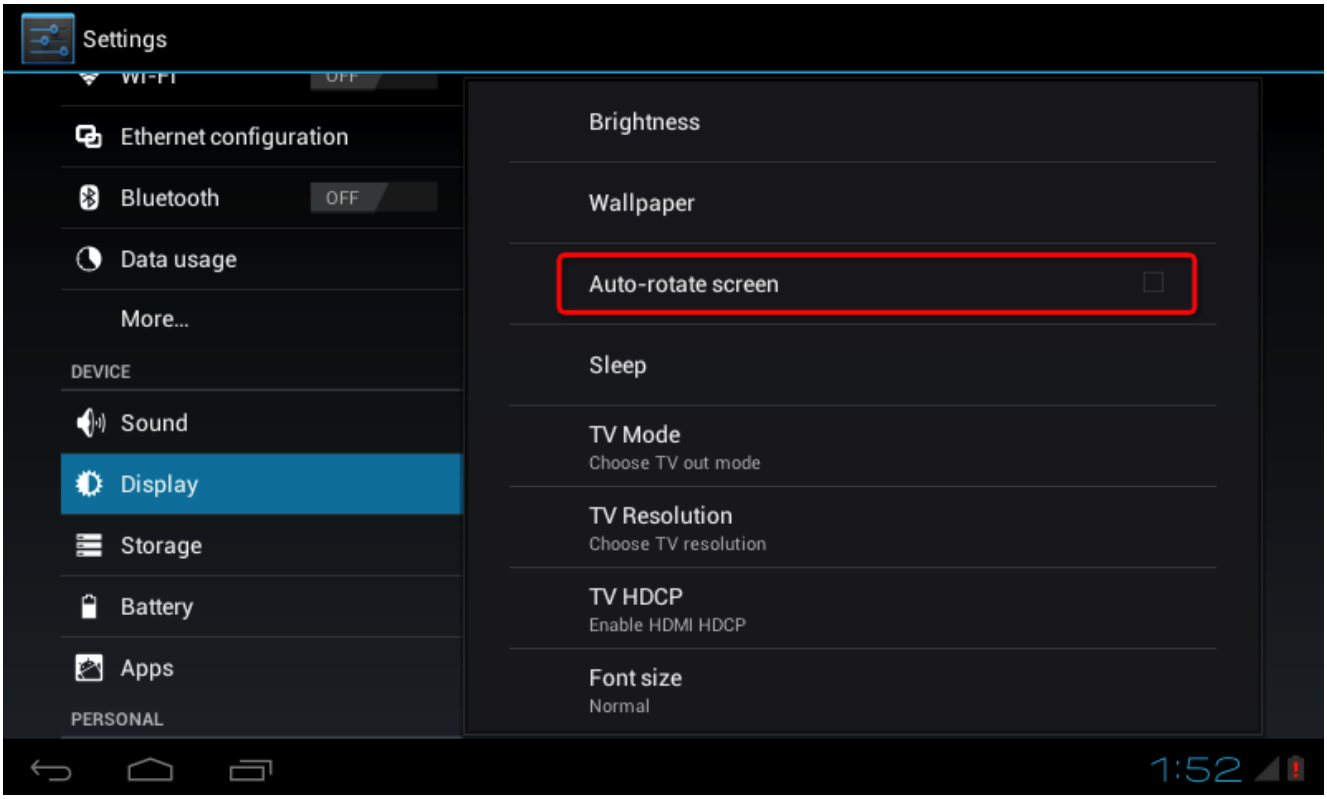
ro.sf.auto_lcd_density=no

ro.sf.lcd_density=200

After you make the changes it will take into effect on system rebooting.

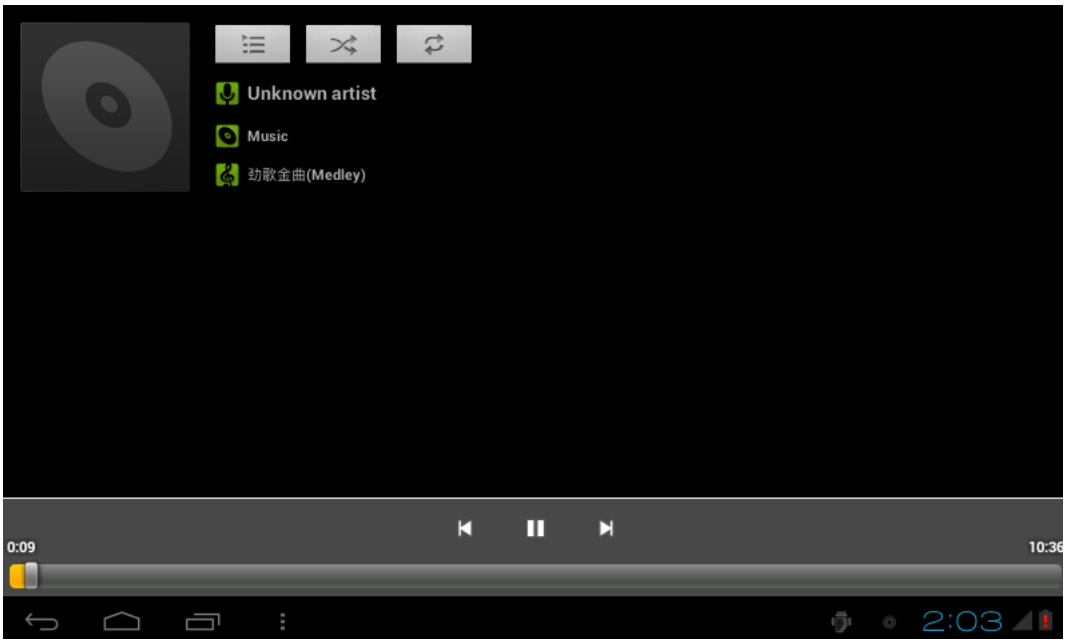
5.2.2.2 Display Auto Rotation

If your 210 board has a g-sensor you can enble this function by setting up the following option.



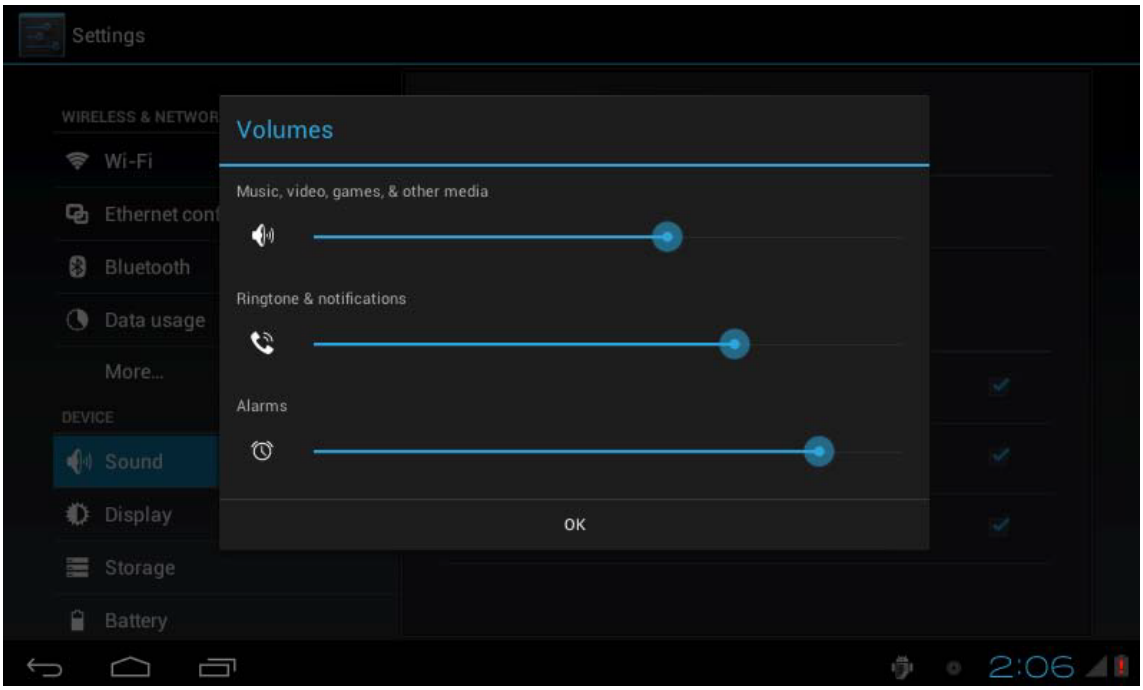
5.2.3 Play MP3

Android can automatically detect MP3 files in the SD card.



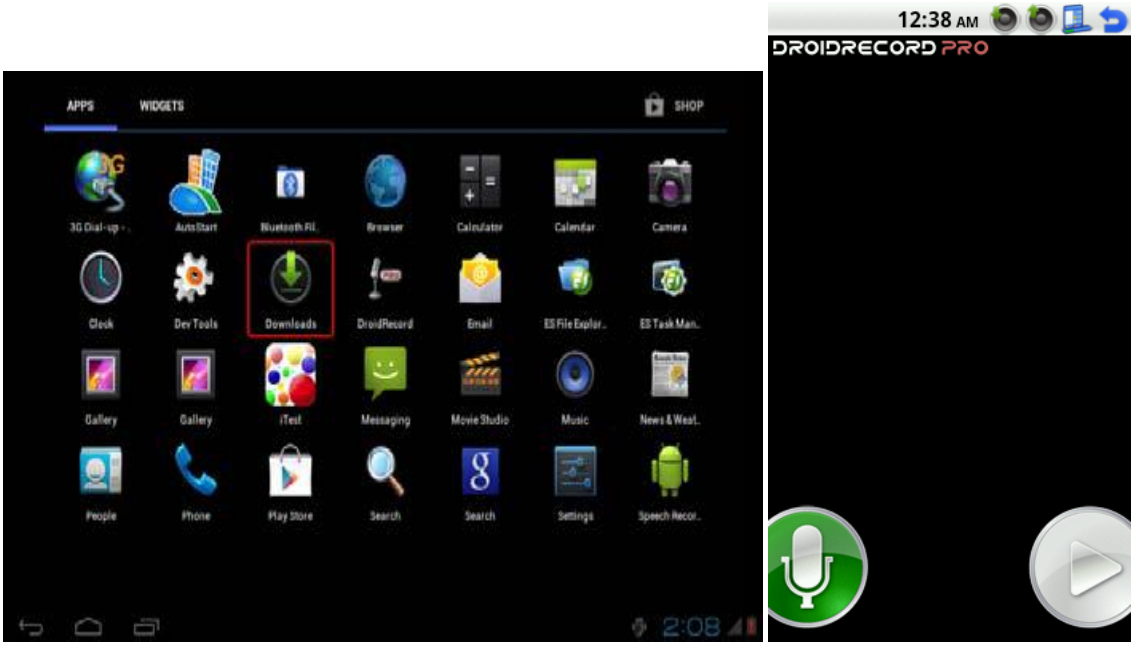
5.2.4 Adjust Volumn

Users can go to “Setting” -> “Sound” to adjust the volumn



5.2.5 Audio Recording

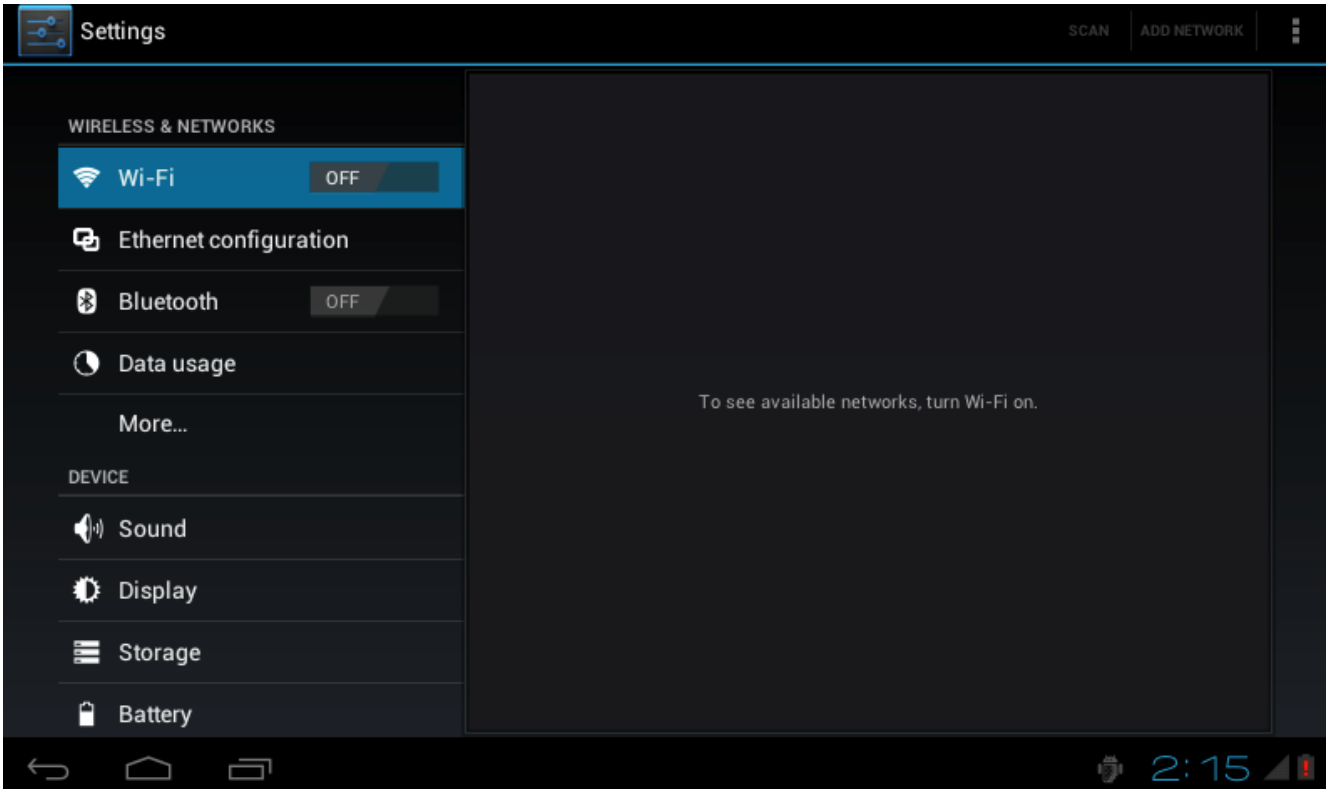
The DroidRecord utility can record and play audio. Double click on the icon to launch it.



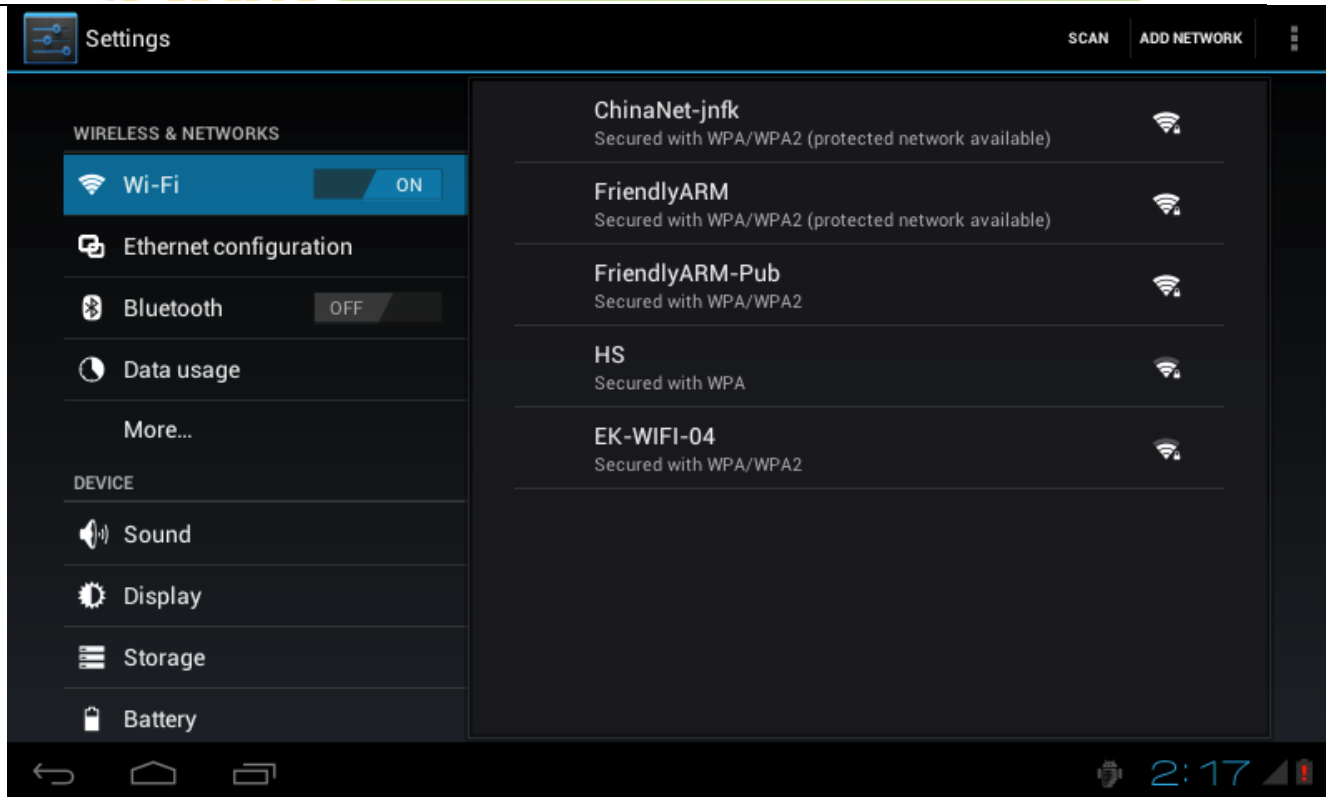
Please follow the screenshots below to start recording and play:

5.2.6 SD WiFi

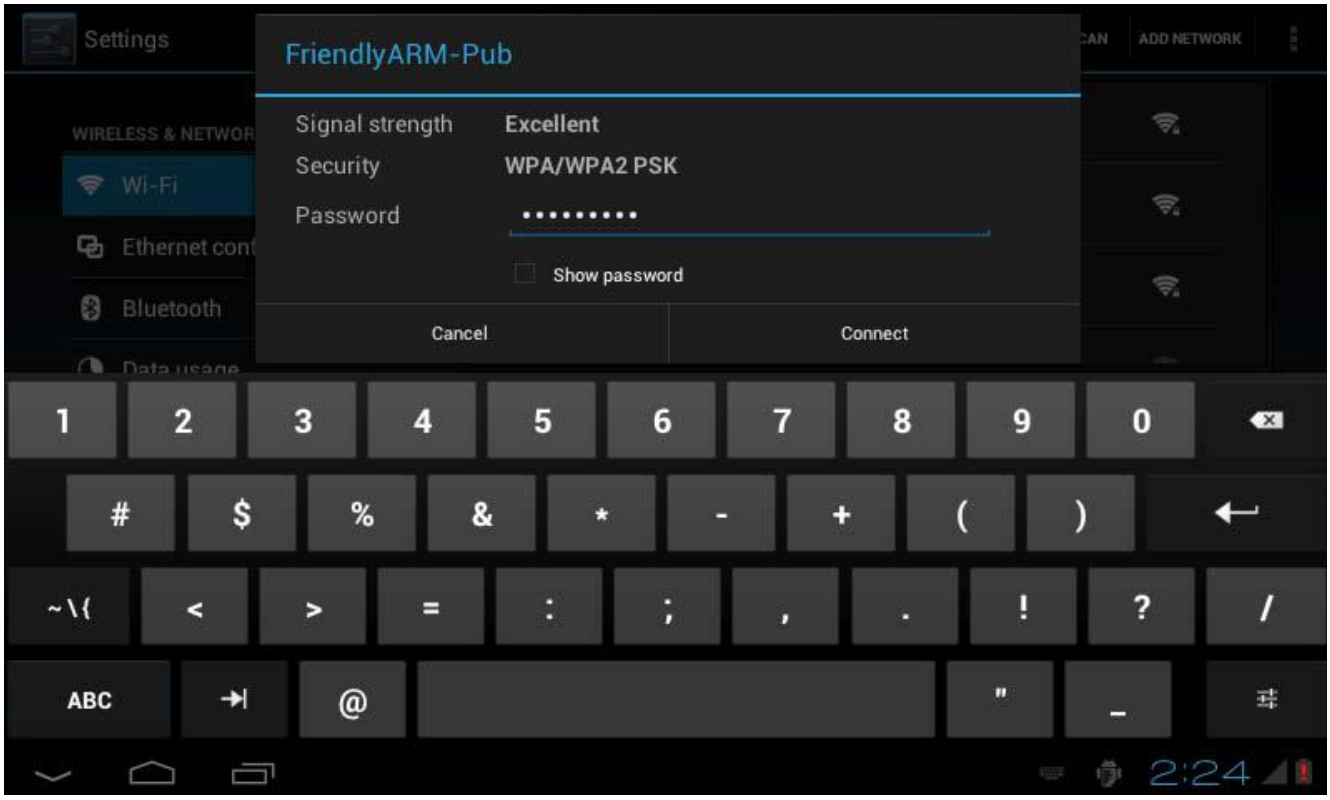
The 210 boards can work with external SD-WIFI or USB WIFI modules. After Andoid loads please go to “settings”.



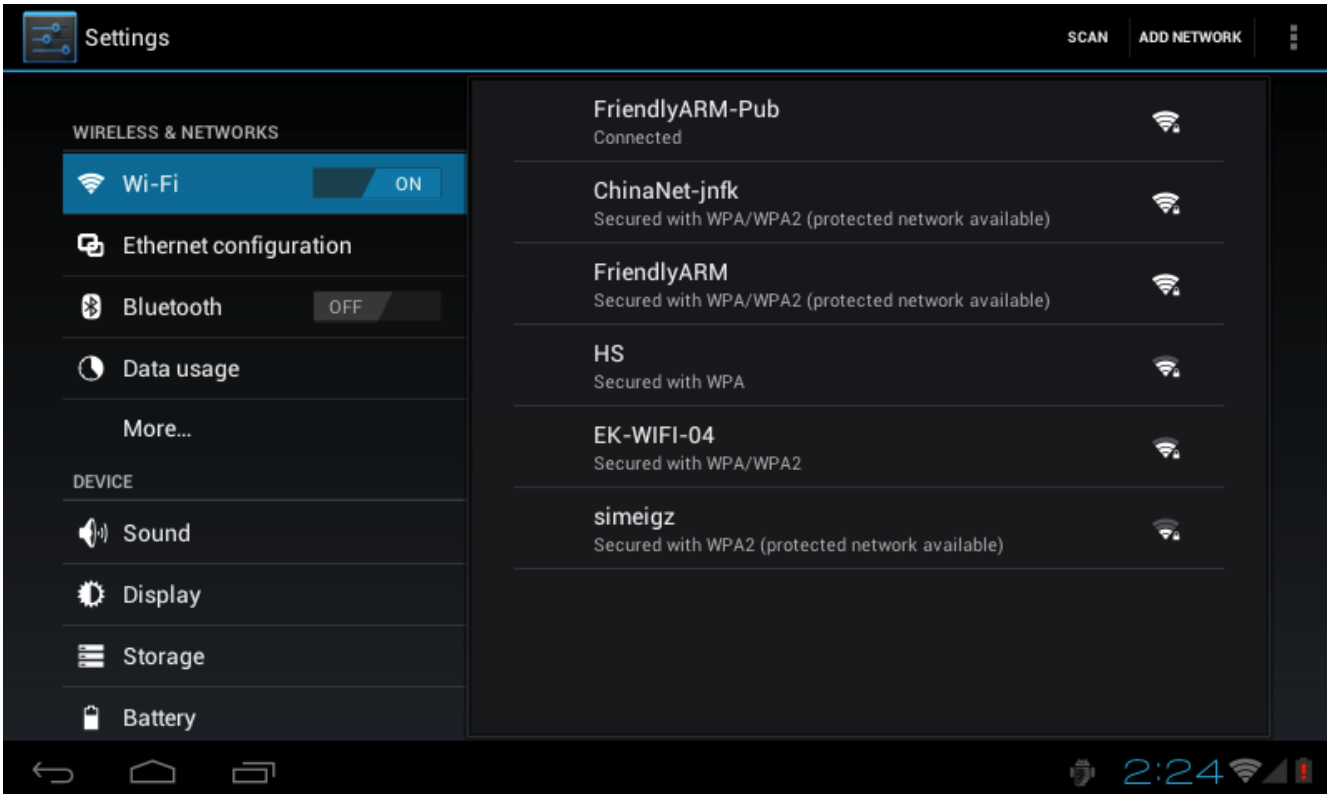
Click on “Wi-Fi” and its “ON/OFF” button.



Select a source and type your username and password



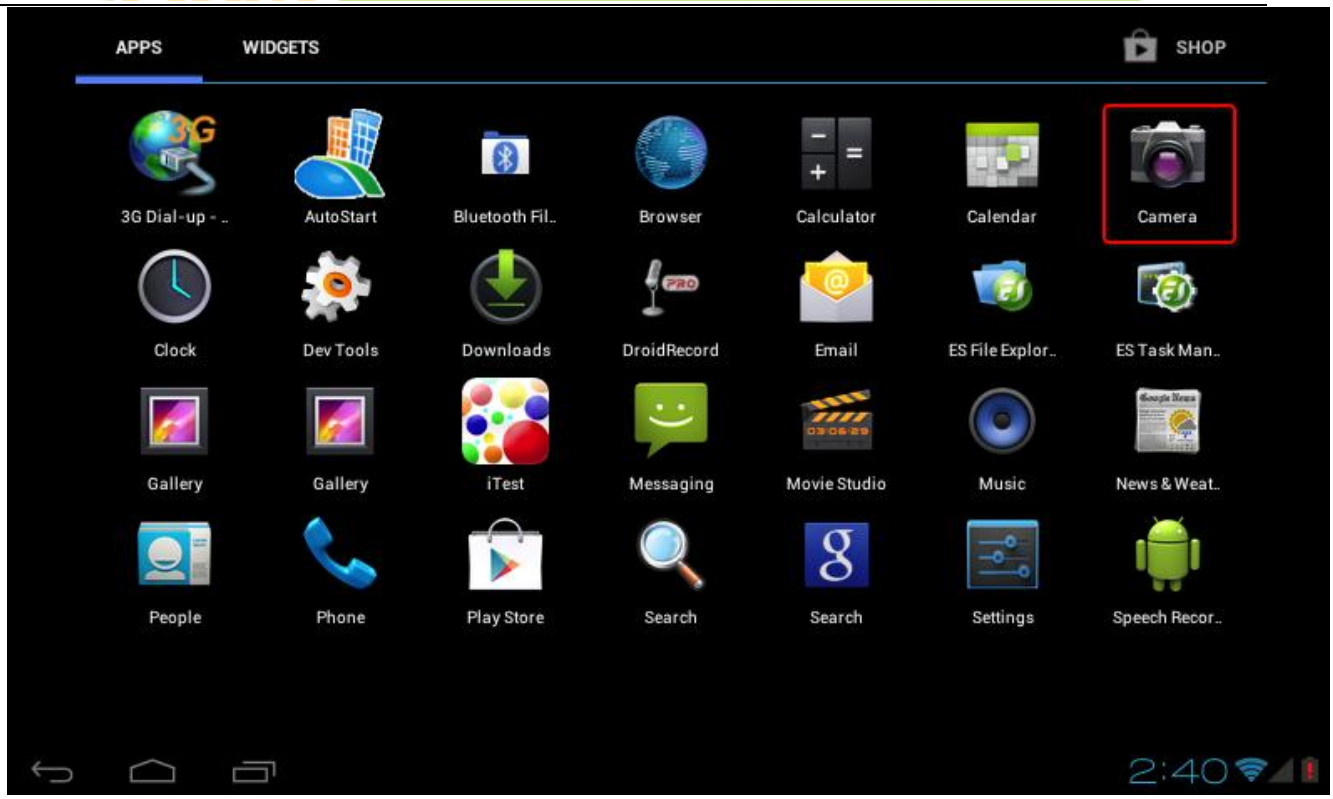
If your connections is successful you will see the following window.



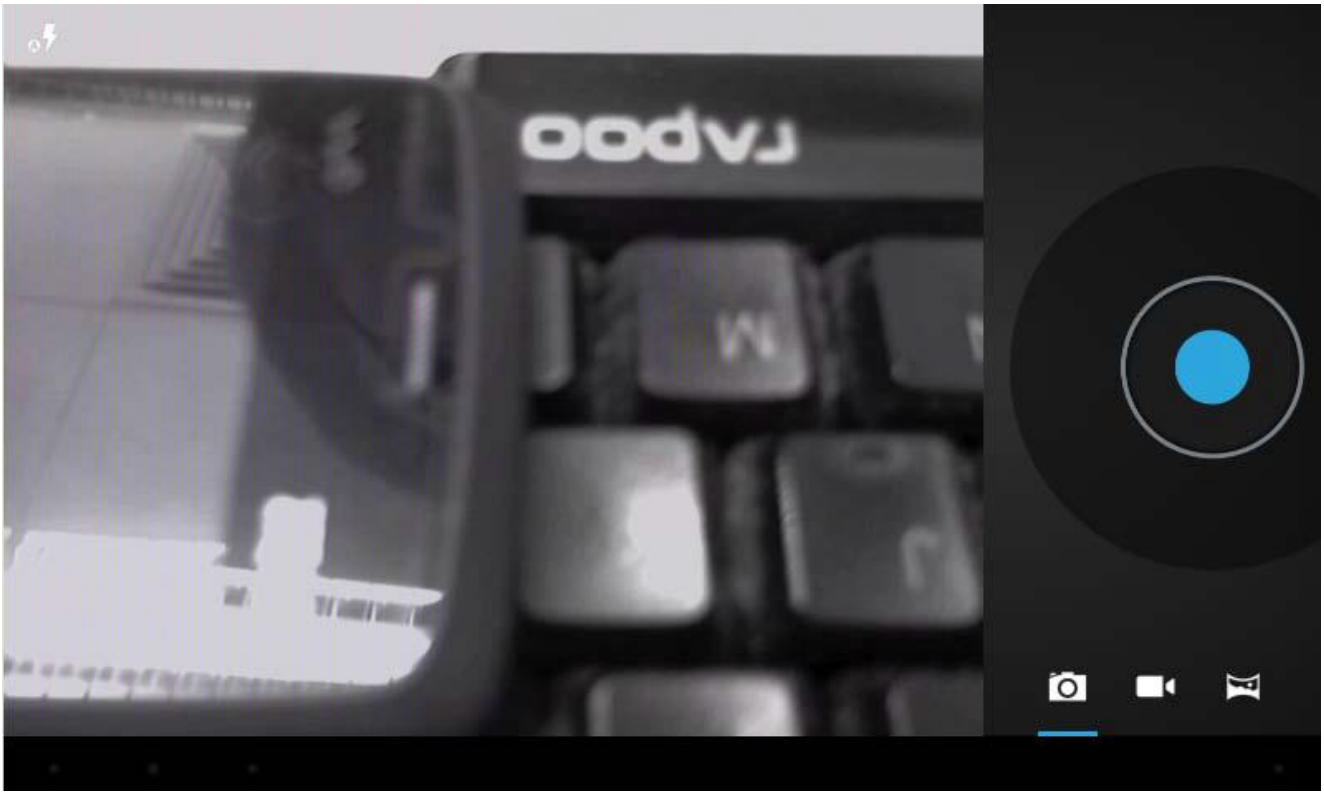
5.2.7 CMOS/USB Camera

The 210 boards can work with both USB cameras and CMOS cameras.

Double click on the “Camera” icon



Double click on it you will see the following window



The round button on the bottom right is the snapshot button. Click on it you will see a picture taken. To browse all your pictures click on the top right button.

We provide two HALs under Android 4 for CMOS, CCD and USB cameras. The CCD and CMOS cameras share the same HAL.

1) camera.cmos.so – this is for the CCD and CMOS cameras
2) camera.usb.so – this is for the USB cameras

When Android 4 is loading it will automatically check whethere a USB camera is connected. If a USB camera is connected Android will call "camera.usb.so" to load the USB camera otherwise it will call "camera.cmos.so".

If you need to load your own HAL or stop the above behavior you can edit item "ro.sf.auto_detect_camera" in the file "/system/build.prop". Setting it to "no" is to stop that behavior.

In our system Android 4 will load "camera.mini210.so" under "/system/lib/hw" when it is loading.

Notes to users as for development of USB camera applications:

1) We tried Logitech's C270 USB camera and it worked. Usually if a camera's data format is YVYV/YUY2 it should work with the 210 boards. If a camera's data format is different you might need to change the value of the item "ro.kernel.android.cam_yuy2=n" in the file "/system/build.prop".

2) Usually a USB camera displays horizontally. When you take pictures with a USB camera your pictures will be saved in the SD card if an SD card is inserted into your board.
3) By default the USB camera's resolution is set to 544 x 288. Users can change the value of the item "ro.kernel.android.cam_def_size" in the file "/system/build.prop". the Logitech C270's resolution can be set to 1280 x 720, 1184 x 656, 960 x 720, 960 x 544, 864 x 480, 800 x 448, 544 x 288, 352 x 288 and 320 x 176.
4) The resolution values that the USB camera's resolution can be set can set in "ro.kernel.android.cam_s_sizes".
5) Here is the section that covers the USB camera's settings in "/system/build.prop"

```
#
# USB Camera Preview and Picture Size (for Logitech C270
webcam)
#

ro.kernel.android.cam_def_size=544x288
#
# USB Camera Supported Size (for Logitech C270 webcam)
#
ro.kernel.android.cam_s_sizes=1280x720,1184x656,960x720,960x544,864x480,800x448,640
x480,544x288,352x288,320x176
#
# USB Camera Using YUY2 ColorSpace (Set to n will support more usb camera model)
#
ro.kernel.android.cam_yuy2=y
```

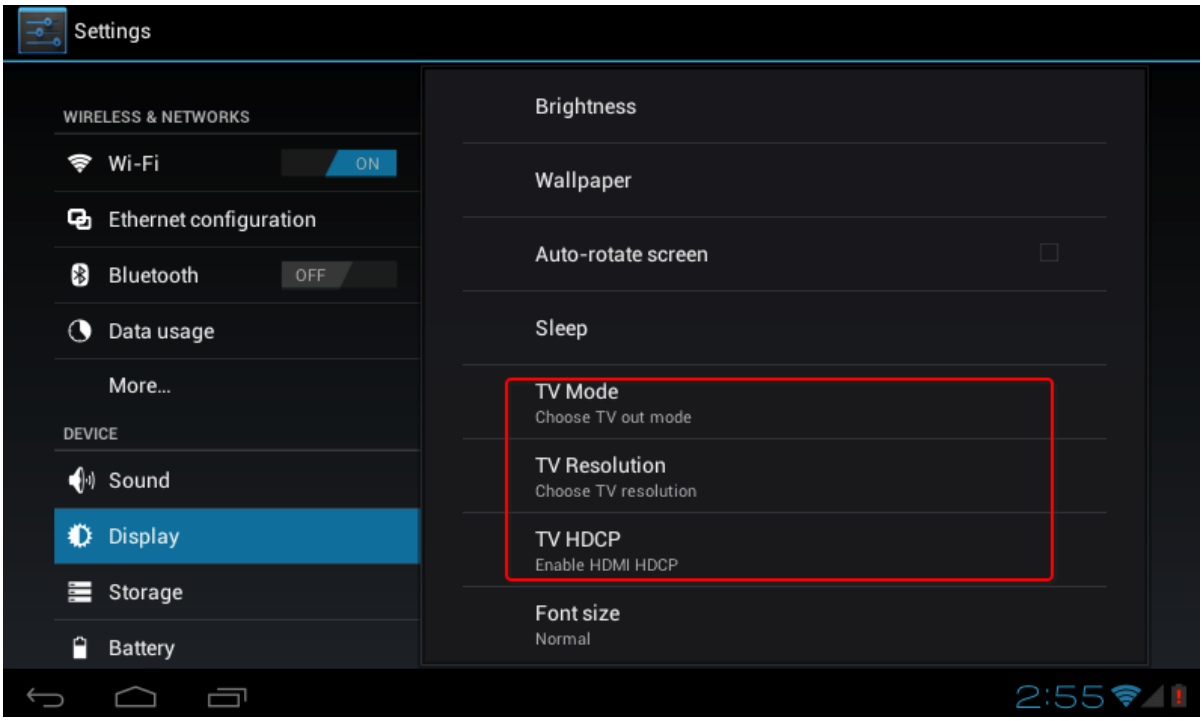
5.2.8 HDMI Output

Connect a 210 board to a TV with a standard HDMI cable the LCD and audio output

will be simultaneously output to your TV.

You can configure the HDMI output format by following the steps below:

1) Go to “Settings” to enter the “Display” menu



2) You can set your resolution to 480p/720p/1080p in “TV Resolution”

5.2.9 HDMI Output Without Connecting LCD

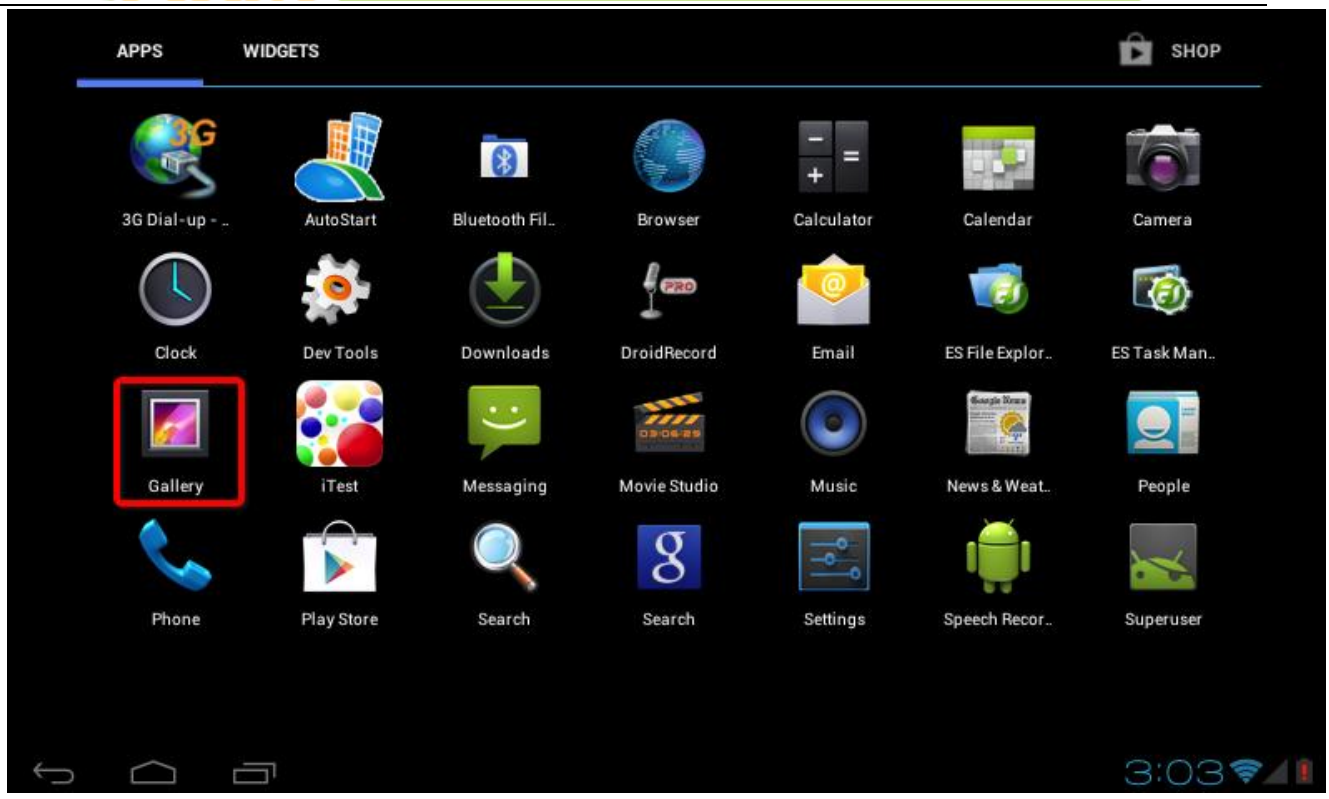
The 210 boards can output to HDMI monitors without connecting an LCD.

If you want to connect your 210 board only to an HDMI monitor without connecting an LCD you need to specify “LCD-Type” e.g. “LCD-Type=HDMI720P60”. You need to reflash your board with this new definition without connecting an LCD.

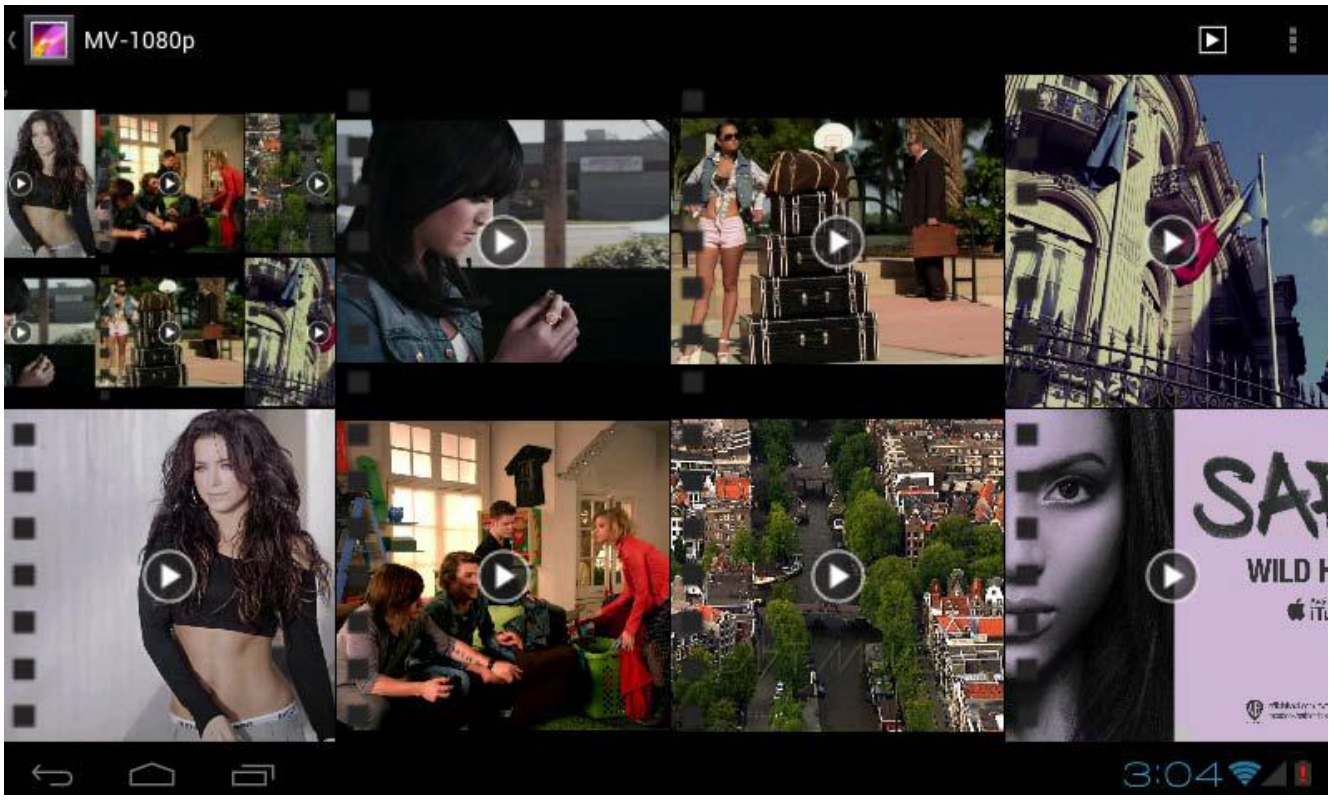
After an OS is installed please power on the board without connecting an LCD you will be able to see HDMI output.

5.2.10 Play High Definition Video

Copy your video files to an SD card (note: they must be mp4 and the audio data is in “aac” format. We provide some test files in the “Test Video” directory) and click on “Gallery”:



You will see all the video files.



When playing video in full screen you will see similar effects like the following screenshot:

When connecting your board to a TV with an HDMI cable the audio and video output will be simultaneously output to your TV.

5.2.11 Play Flash

Playing Flash on a web page needs flash plugins. Users can download and install them on your board. After that your board will be able to play Flash as follows:

5.2.12 GPS

FriendlyARM specially developed utilities for serial port based GPS devices in Android. Theoretically it should support USB based GPS devices as well. If your board connects to a USB GPS you need to change your GPS device to ttyUSB0 by editing the init.rc file to add a property "ro.kernel.android.gps" to your GPS device such as "set prop ro.kernel.android.gps ttyUSB0". The default device is s3c2410_serial1.

We take a serial GPS device as an example to show how to use GPS:

1) Connect your GPS device to your board's CON4.
2) Add one line in "init.rc" : "setprop ro.kernel.android.gps s3c2410_serial3"
3) Save and reboot

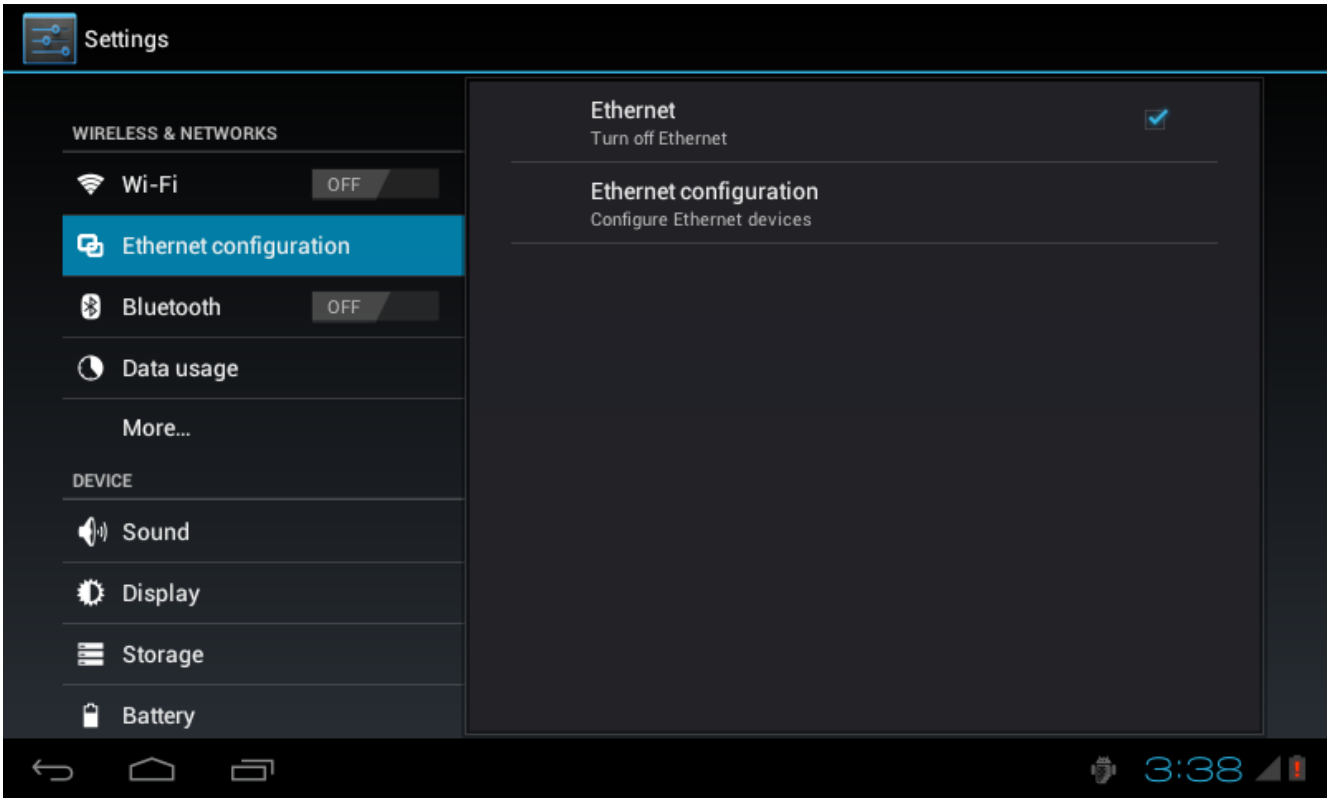
If you want to use google maps or baidu maps you need to connect your board to the internet. We used baidu map in the following example. We connected the board to the

internet, opened baidu's map and used our GPS device to position:

Note: to get better signals we suggest testing this function outdoors.

5.2.13 Configure Ethernet

Android has an ethernet configuration utility

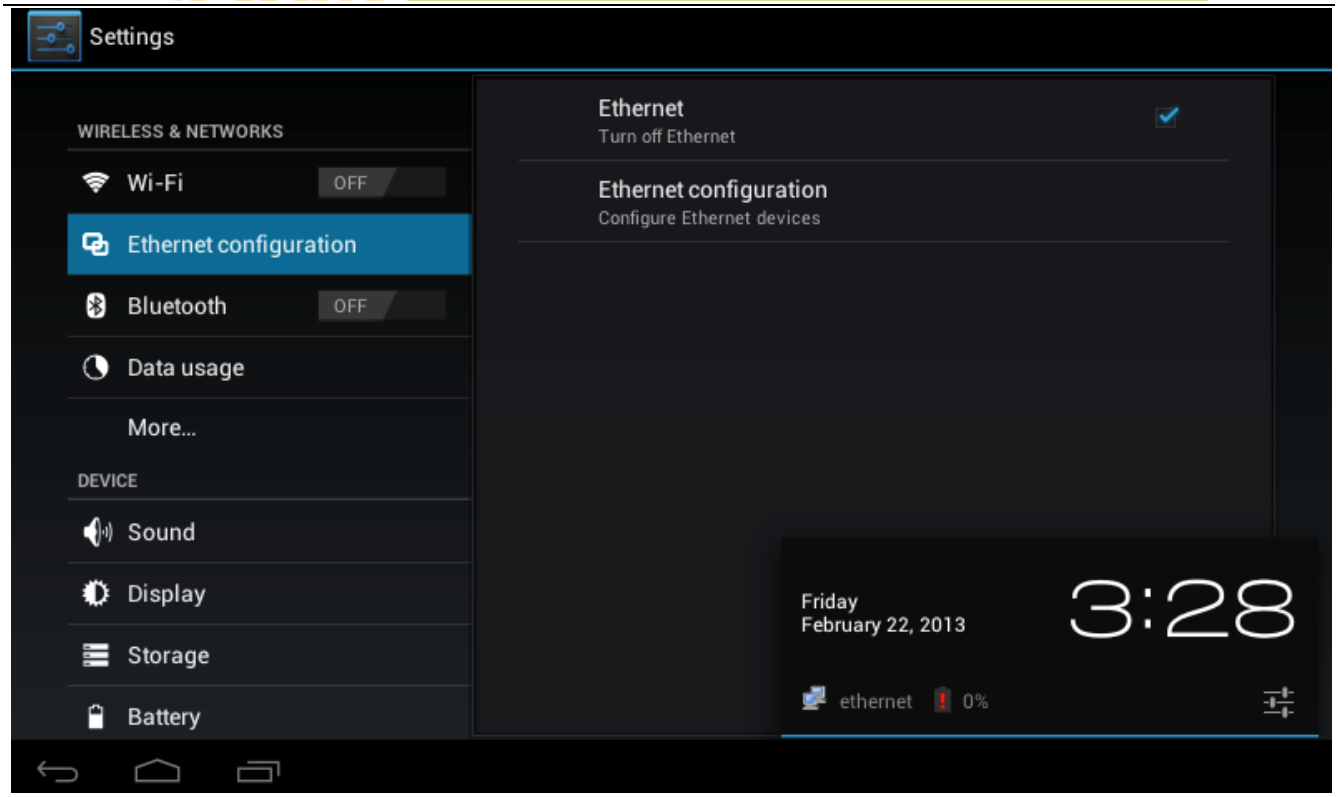


Click on “Ethernet Configuration” you can do either of the two settings:

1) Ethernet: turn on/off the Ethernet

2) Ethernet configuration: setup DHCP or static IP

After Ethernet connection is successfully set you will see an icon as follows:



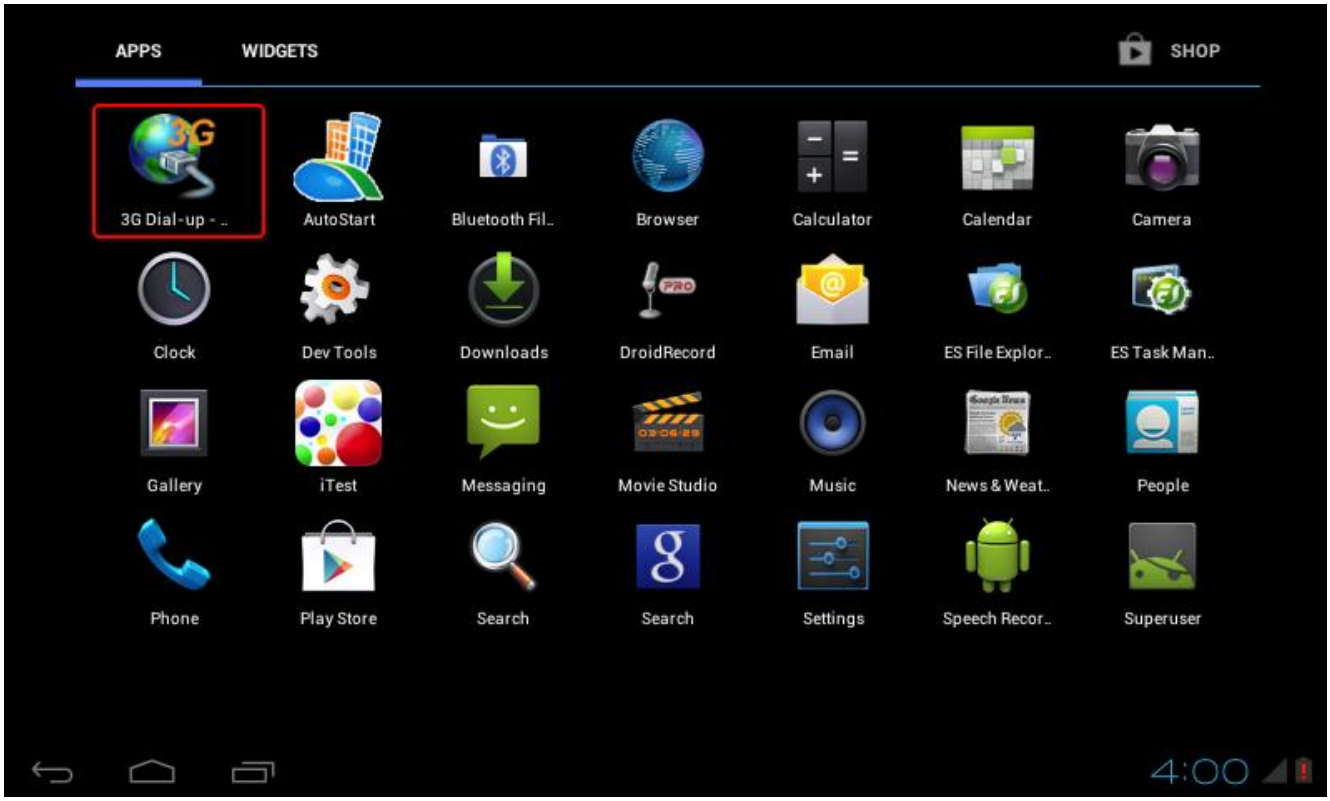
5.2.14 3G Dial-Up

5.2.14.1 3G Manual Dial-Up

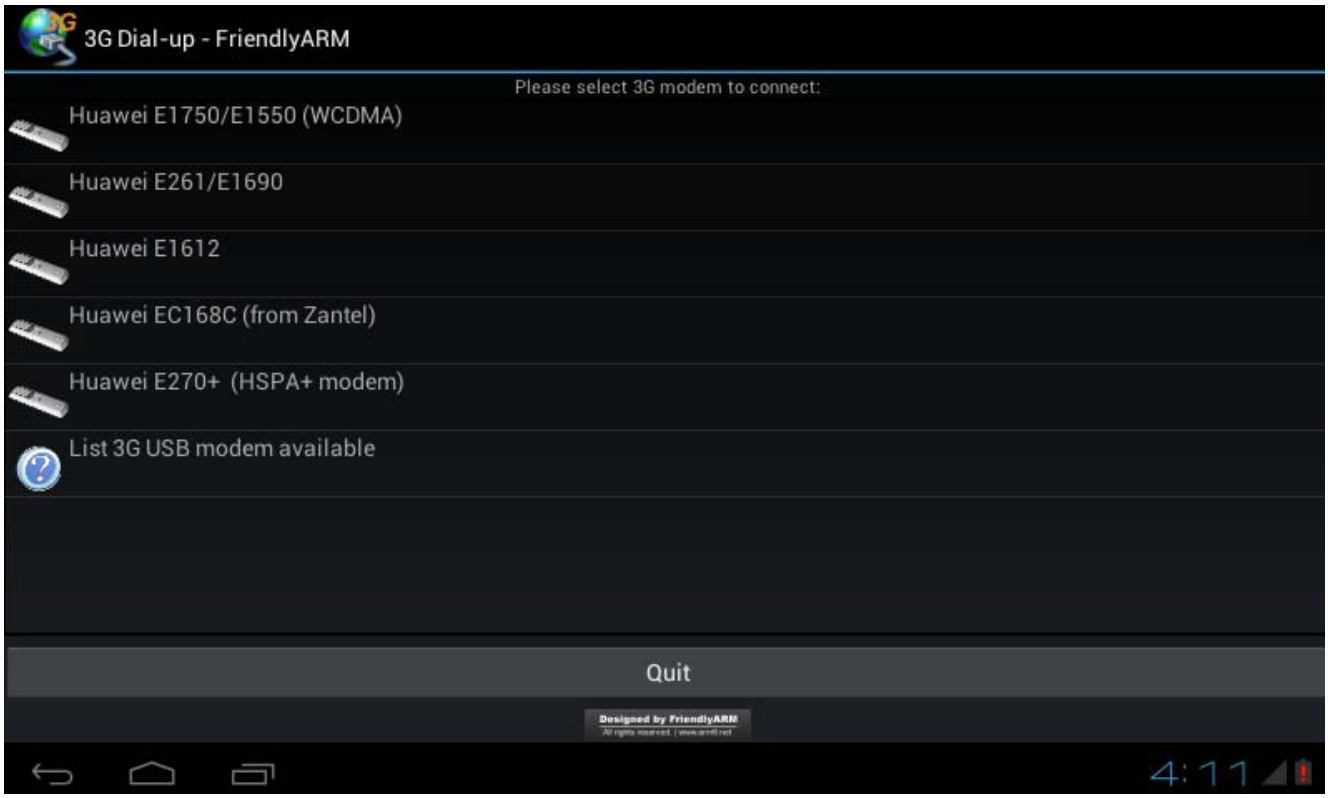
We specially developed a 3G network utility for Android. It can automatically detect and support up to more than one hundred USB network cards for all these systems: WCDMA, CDMA2000 and TD-SCDMA. We have a list of the USB 3G cards that are supported (listed in later sections).

Step1 insert a SIM card to your 3G card

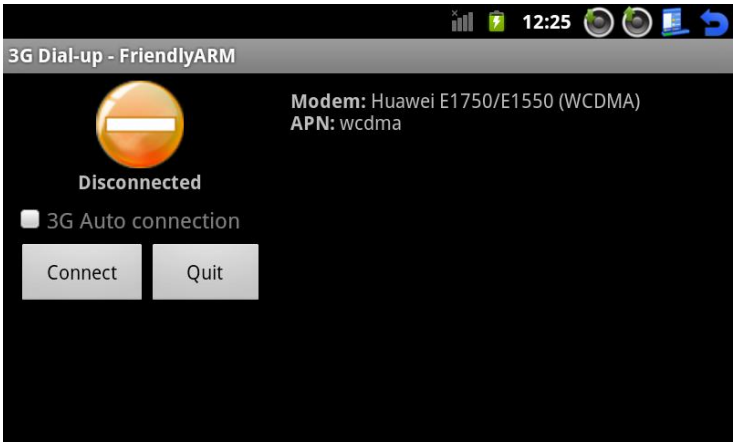
Step2 connect your USB card to the board and start the 3G utility



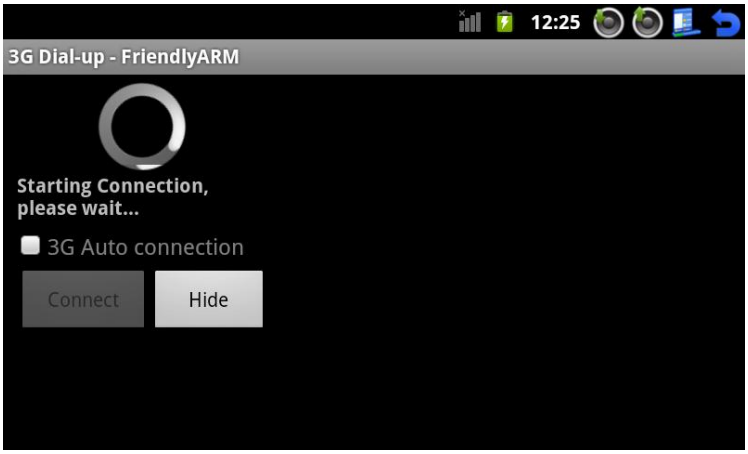
Step3 after the 3G card is detected please click on its icon



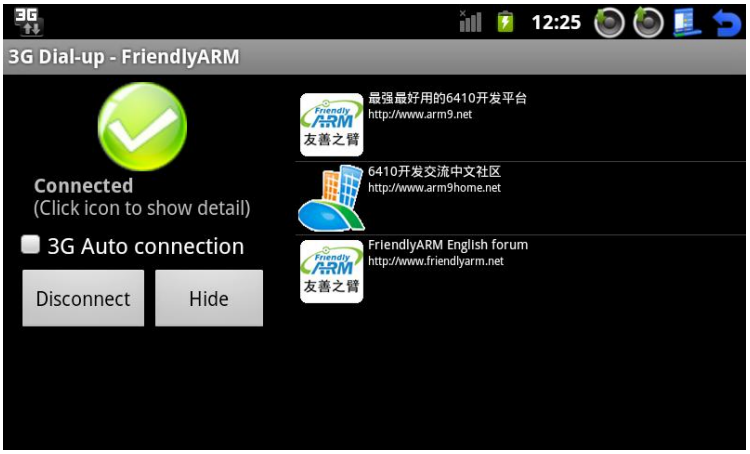
Step4 in the dialog shown below there is an orange icon with a "-" in the center. This means no network is connected. Click on "Connect" to start connection



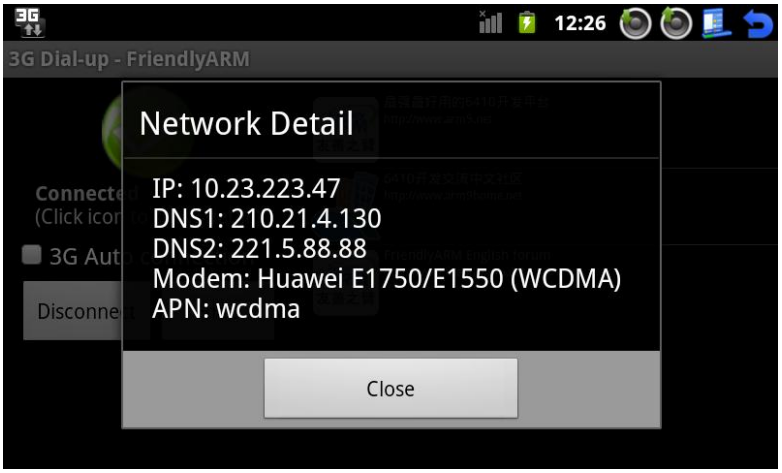
Step5 the connection may take a while



Step6 if the connection is a success the orange icon will turn green and shows "Connected" and meanwhile FriendlyARM's websites will be listed and a "3G" icon will show up on the upper left of the screen.



Step7 Click on the green icon you will see the current network information

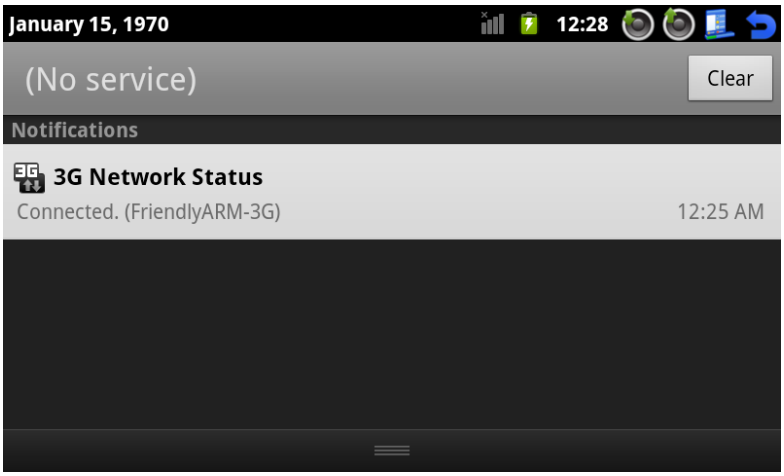


Step8 you can click on "Hide" to run it on background

Step9 try youku.com

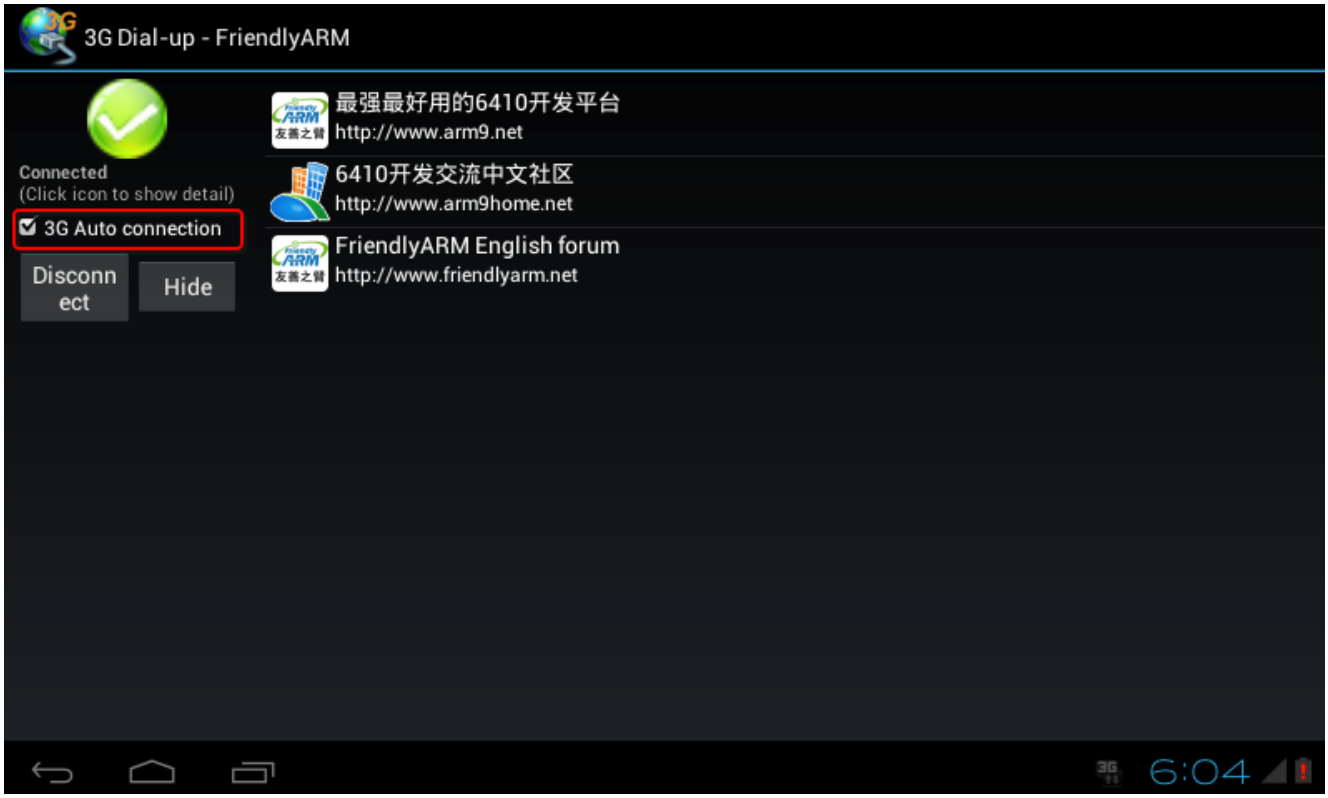
Try QQ browser

Step10 to close the connection click on the “3G Network Status” icon to return to the main menu and click on “Disconnect”

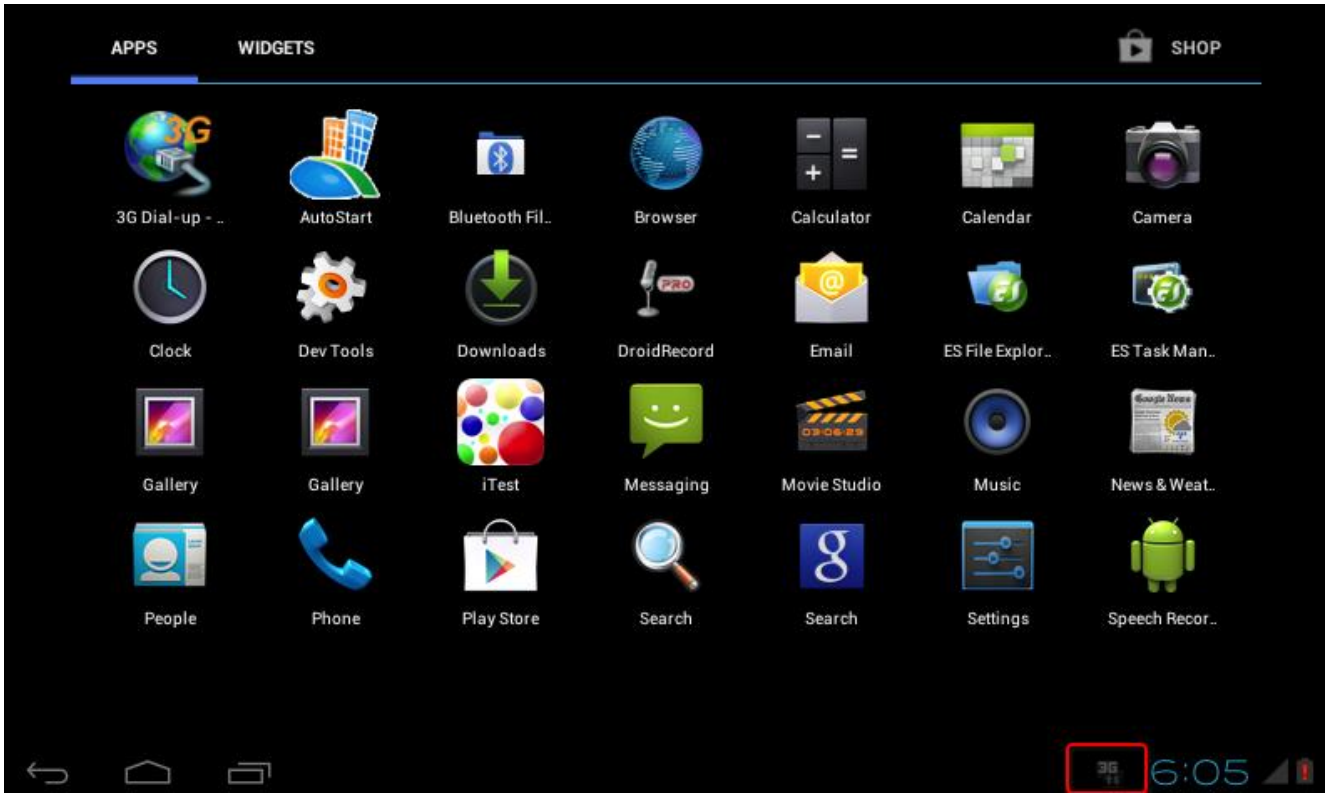


5.2.14.2 3G Auto Dial-Up

This utility also supports auto dial-up on system startup. Make sure your board connects to the internet and click on "3G Auto connection" in the dialog shown below. 3G auto dial-up will be effective on system reboot.



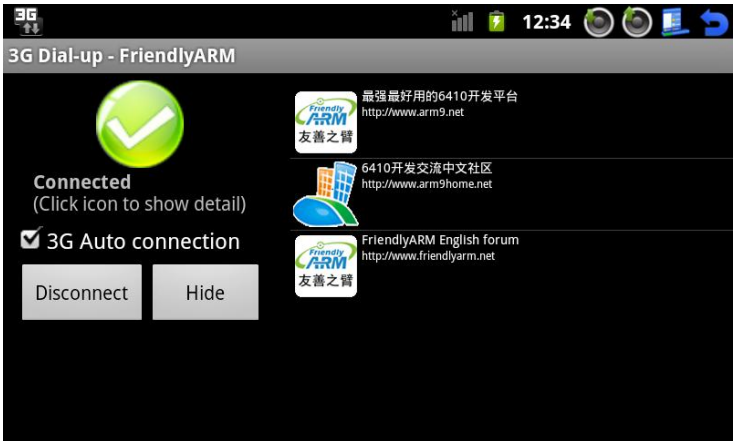
On system startup, if there is a 3G icon shown on the top left it indicates that 3G auto dial-up is on.



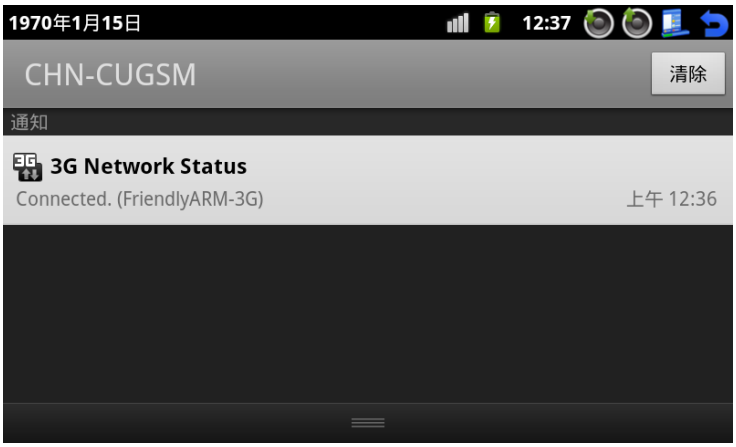
(Note: 3G auto dial up by default uses the USB card you used before you set it therefore if you use a different card you need to reset it)

5.2.15 3G Dial-up and Messaging

To use 3G to send and receive messages, you can set up 3G auto dial up and do it:

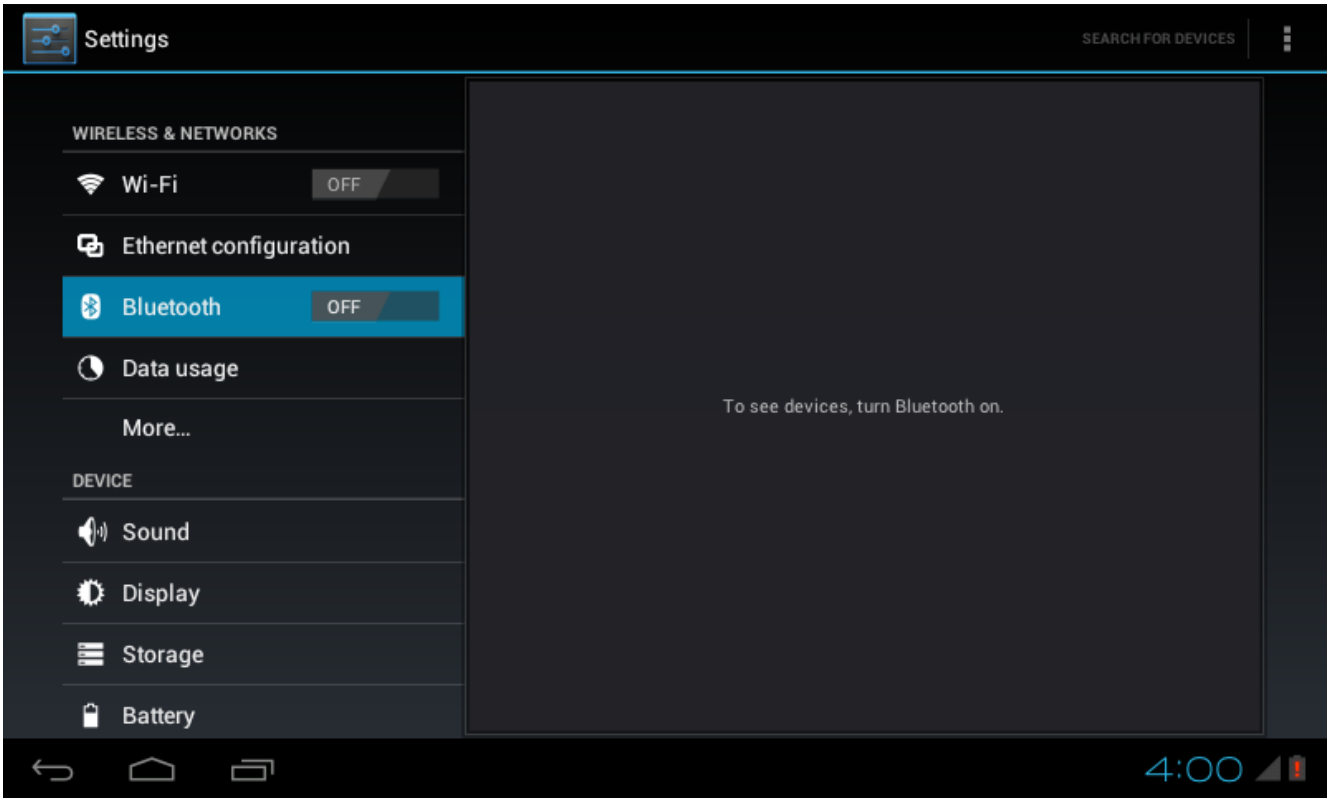


If 3G auto dial up is set your USB device will act as a Modem. Please pull down the status bar if it has the service provider's information it means you can send and receive messages

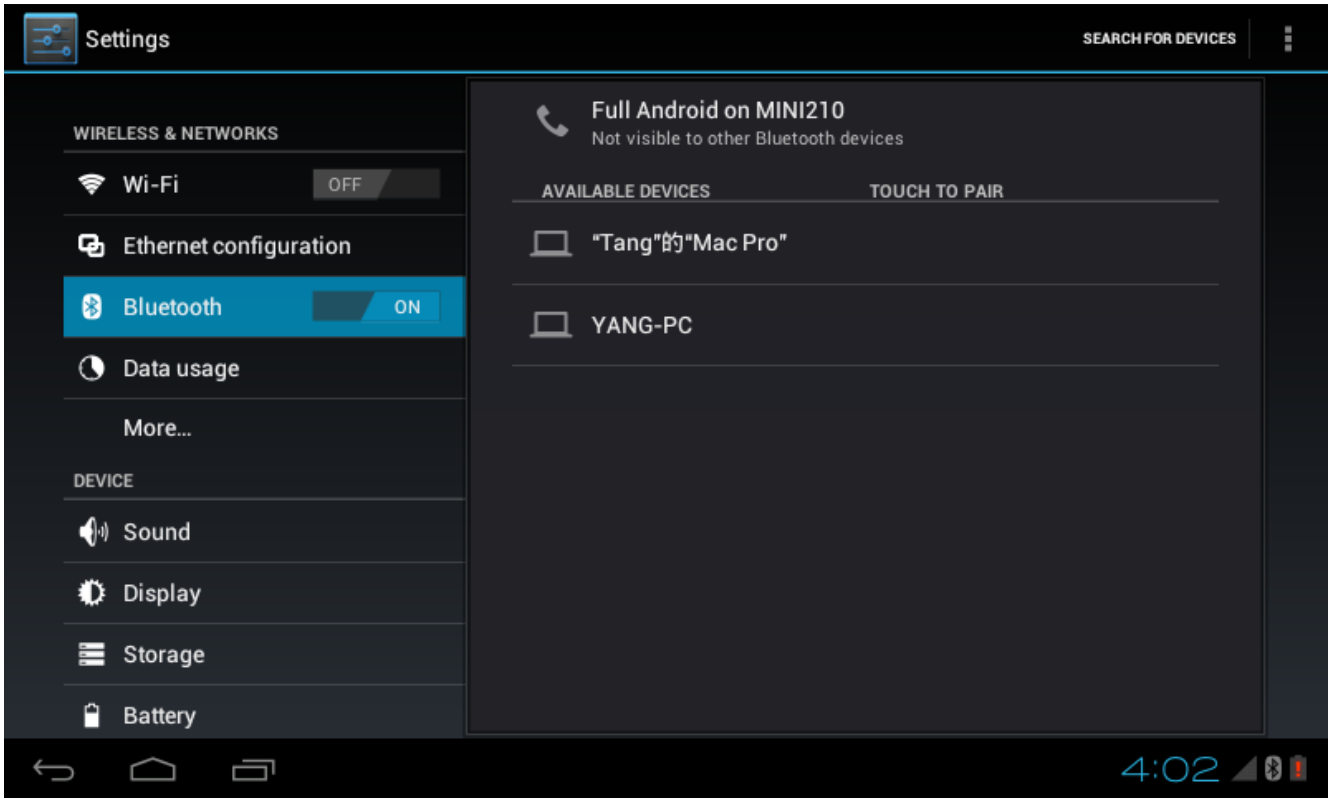


5.2.16 USB Bluetooth

Android supports various USB bluetooth devices. Please connect your USB Bluetooth card to the USB host on the board, press the K3 button and click on "Settings" to enter the main menu:

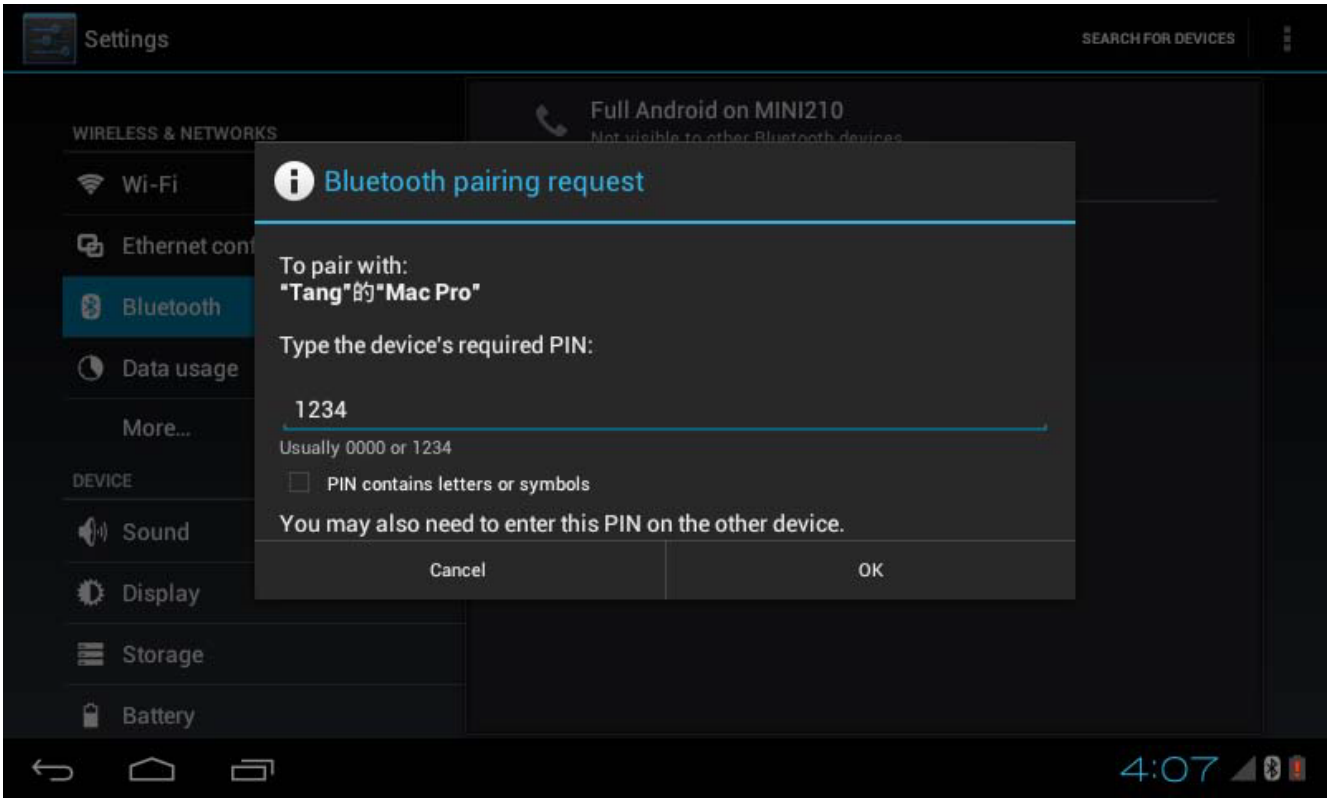


After click on “Bluetooth” to turn it on it will search for nearby bluetooth devices and list them



5.2.16.1 Bluetooth Communication

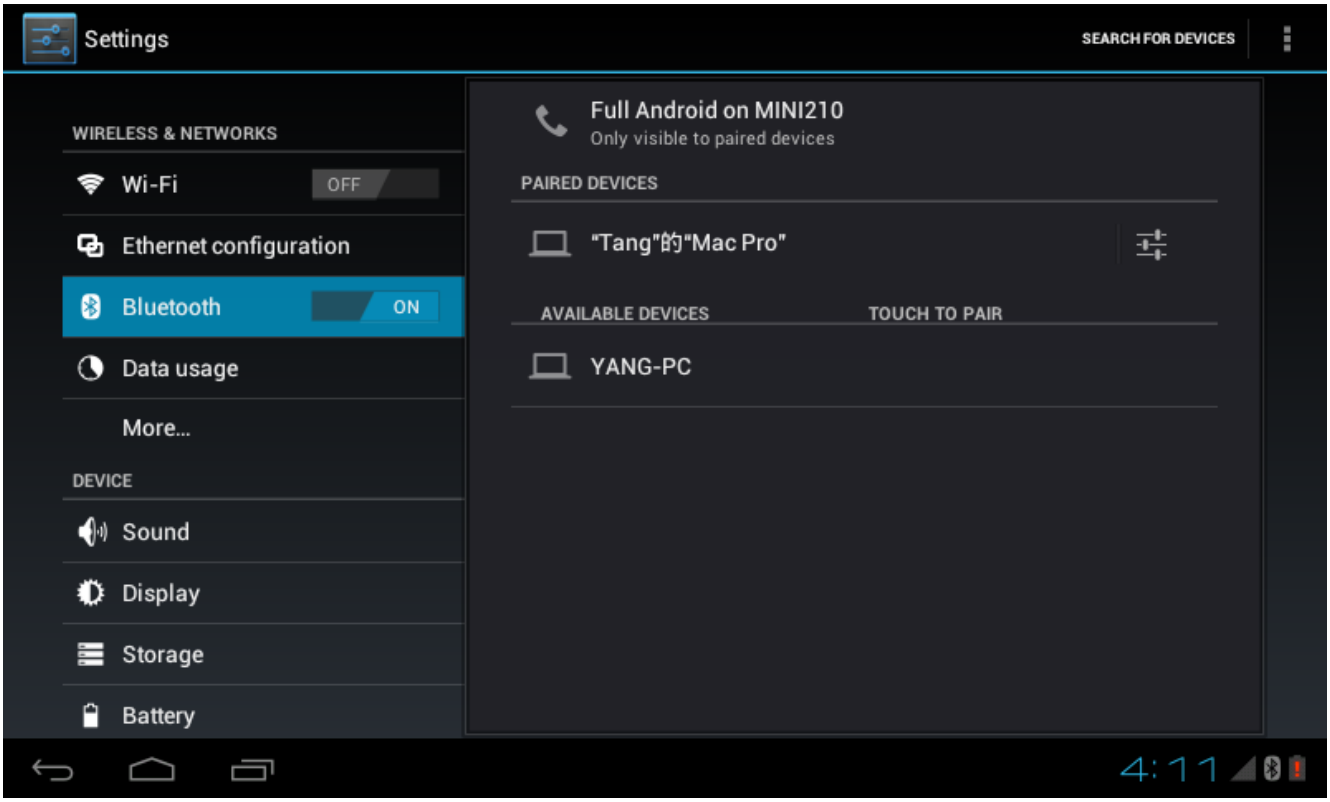
Please get a cell phone which supports bluetooth and start the Bluetooth service. Boot your board with Android, go to “Bluetooth settings”, click on “Scan for devices” and it will find your cell phone.



Click on the cell phone name, type the password and click on “OK”

At the same time there is a dialog shown on your cell phone prompting you to input a password. Type the same one you did on the board.

If the connection is a success you will see the following window.

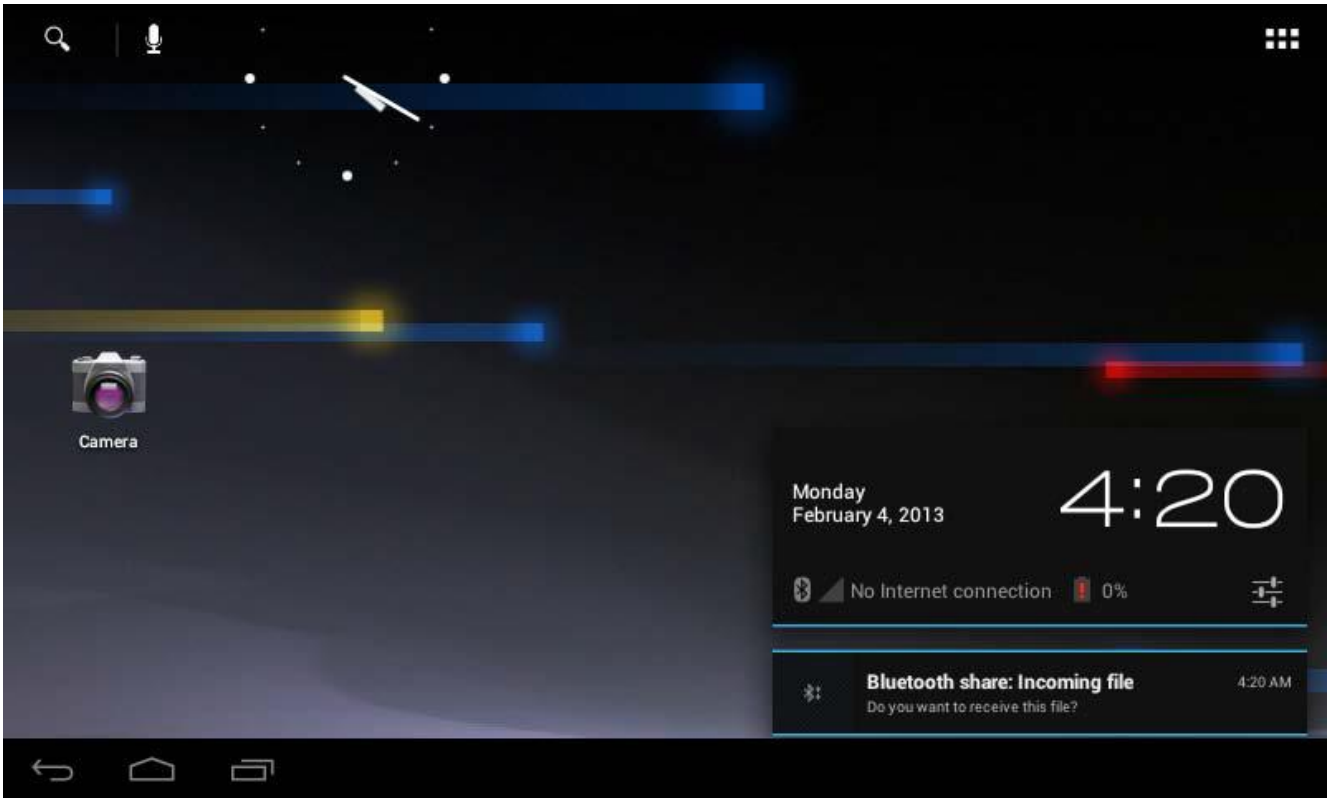


5.2.16.2 Transfer Files to Cell Phone

Please follow the steps described in the previous section to connect your board to a cell phone. Power on your board, open the “Gallery”, click a picture you want to transfer, click on the top right icon and then select “Bluetooth” to start transfering.

5.2.16.3 Transfer Files to 210 Board

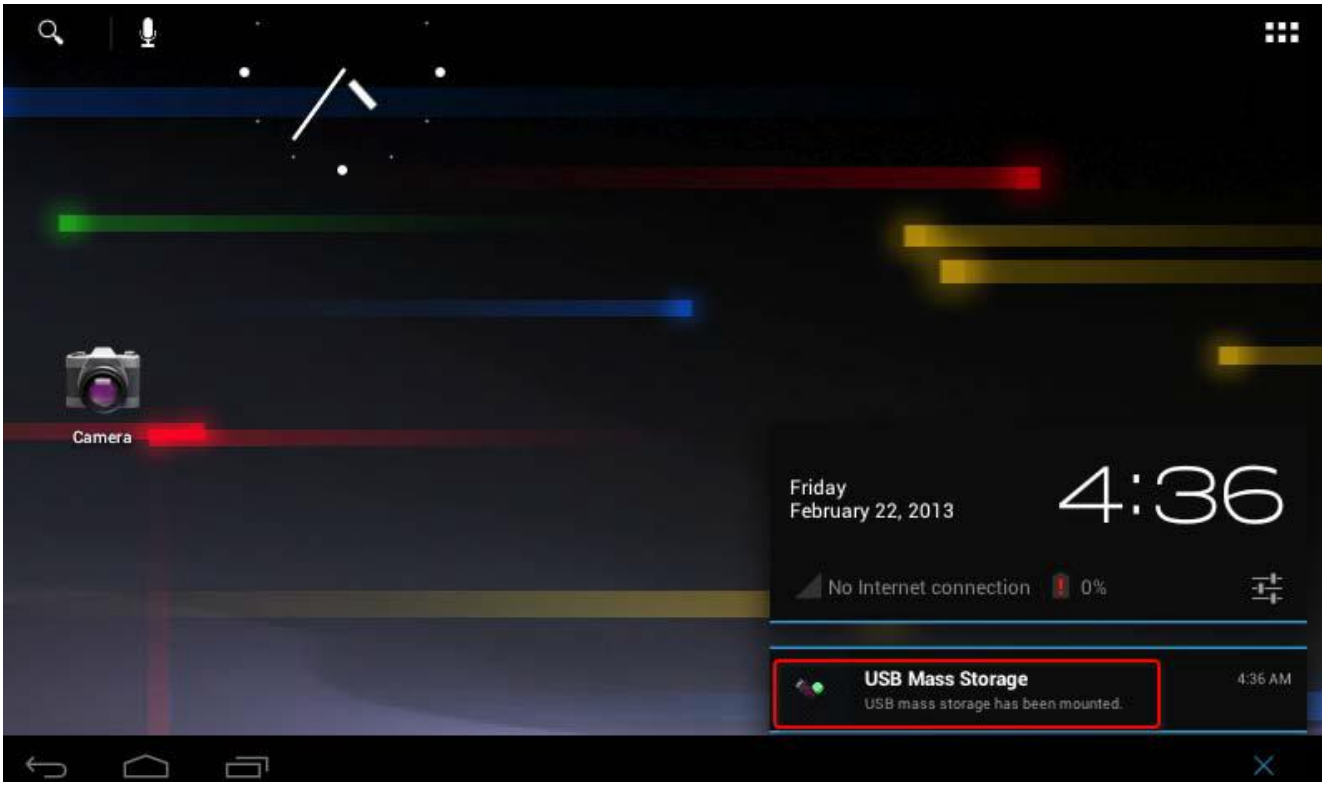
Please follow the steps described in the previous section to pair your board with a cell phone. After you send a file from your cell phone to your 210 board you will see the following window on your board



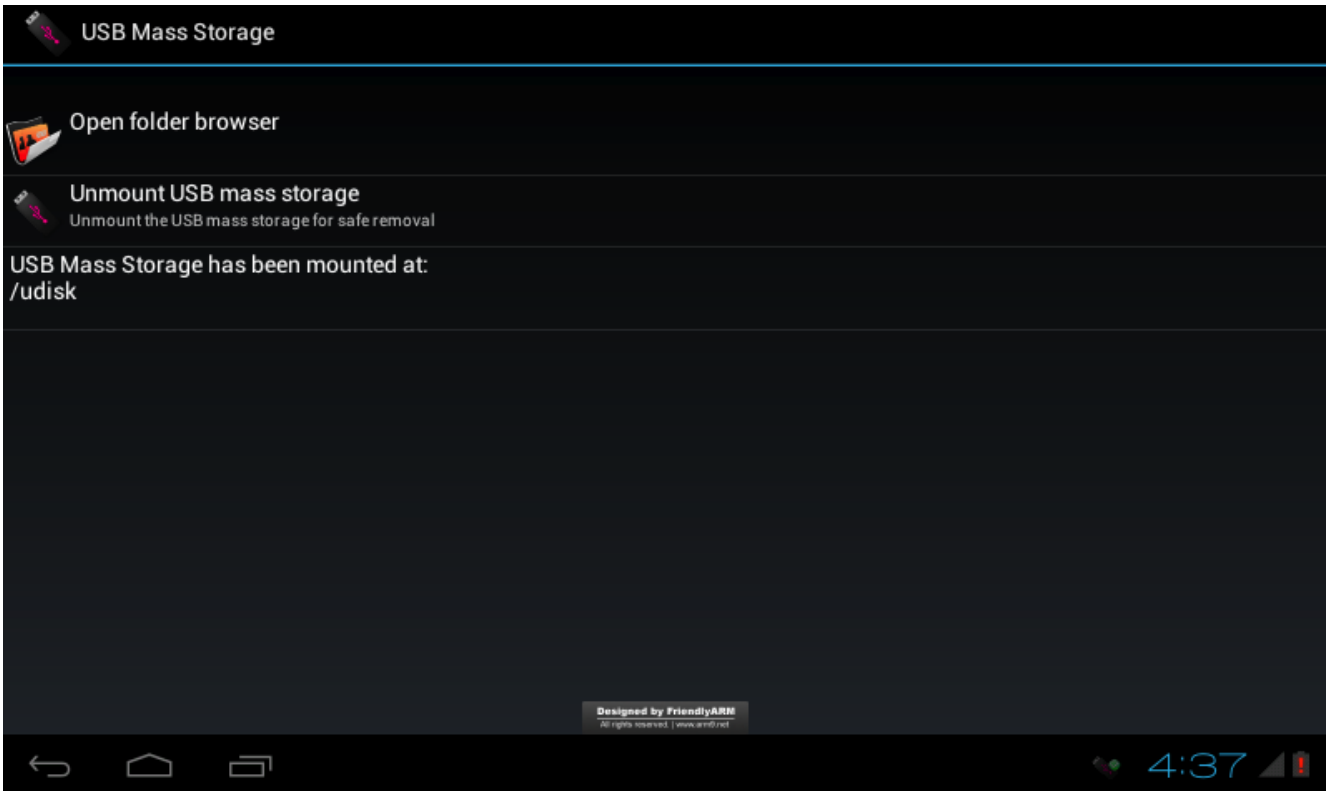
5.2.17 USB Flash Drive

Android supports plug and play of USB flash drives up to a maximum of 32G (note: the drive should be formatted to FAT32).

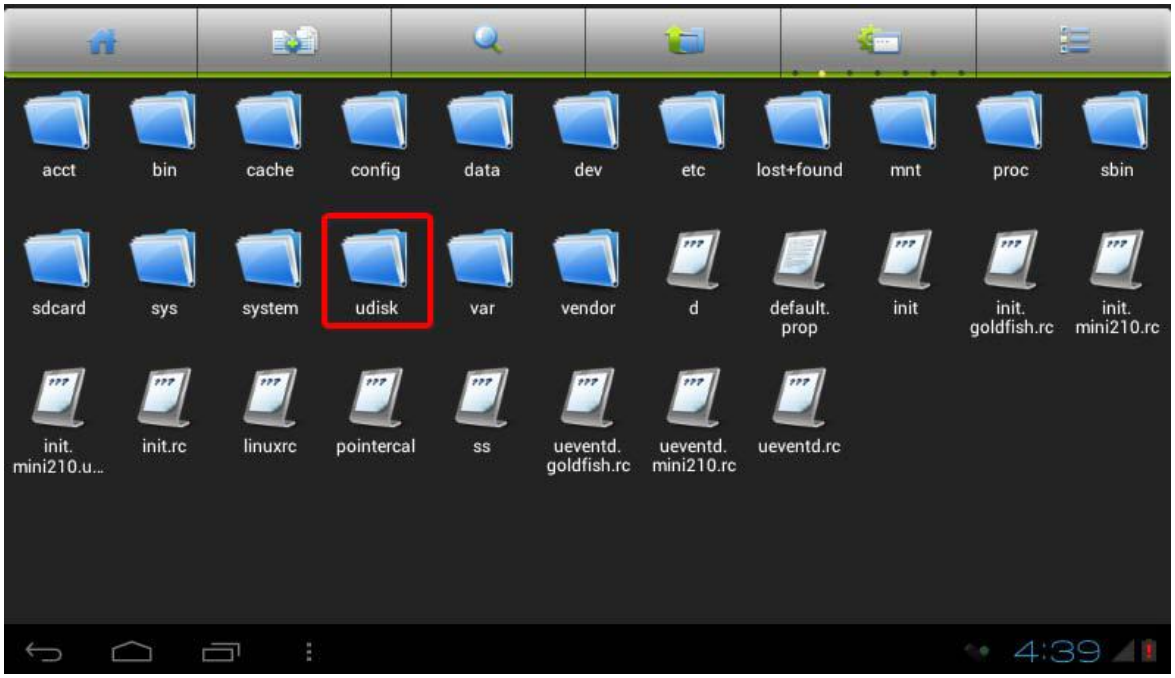
Insert your drive to the USB host and a flash drive icon will apprear on the bottom right of the screen

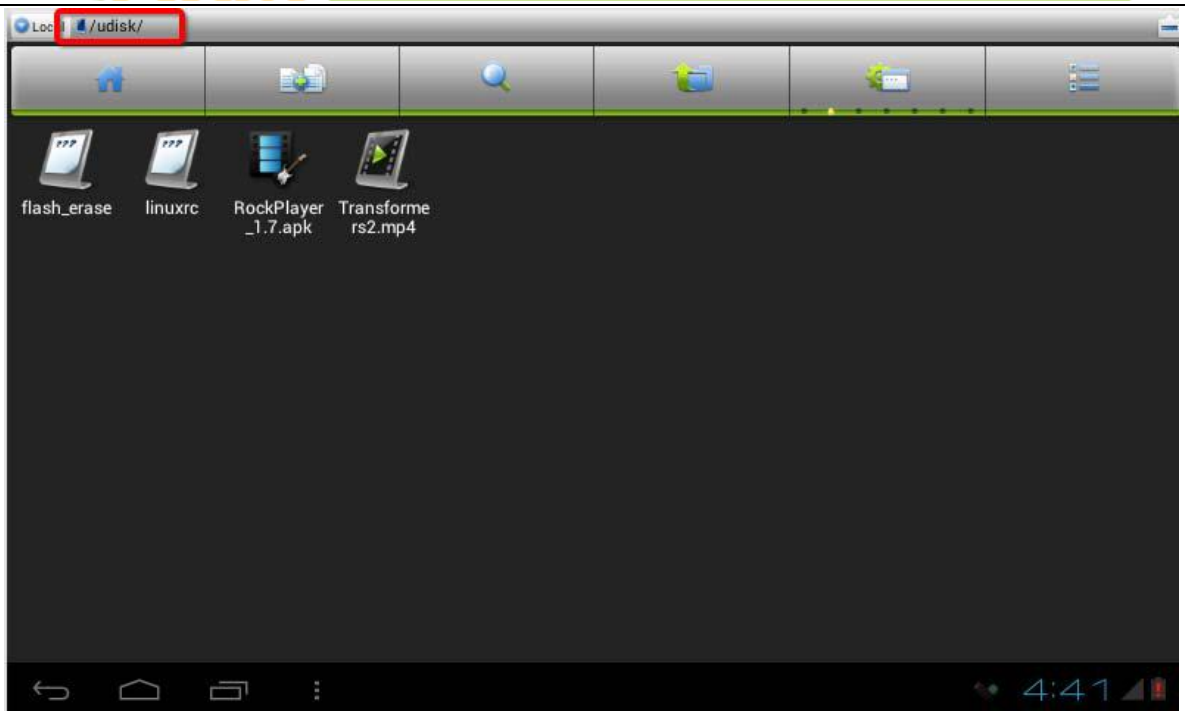


Click on the icon you will see the following window



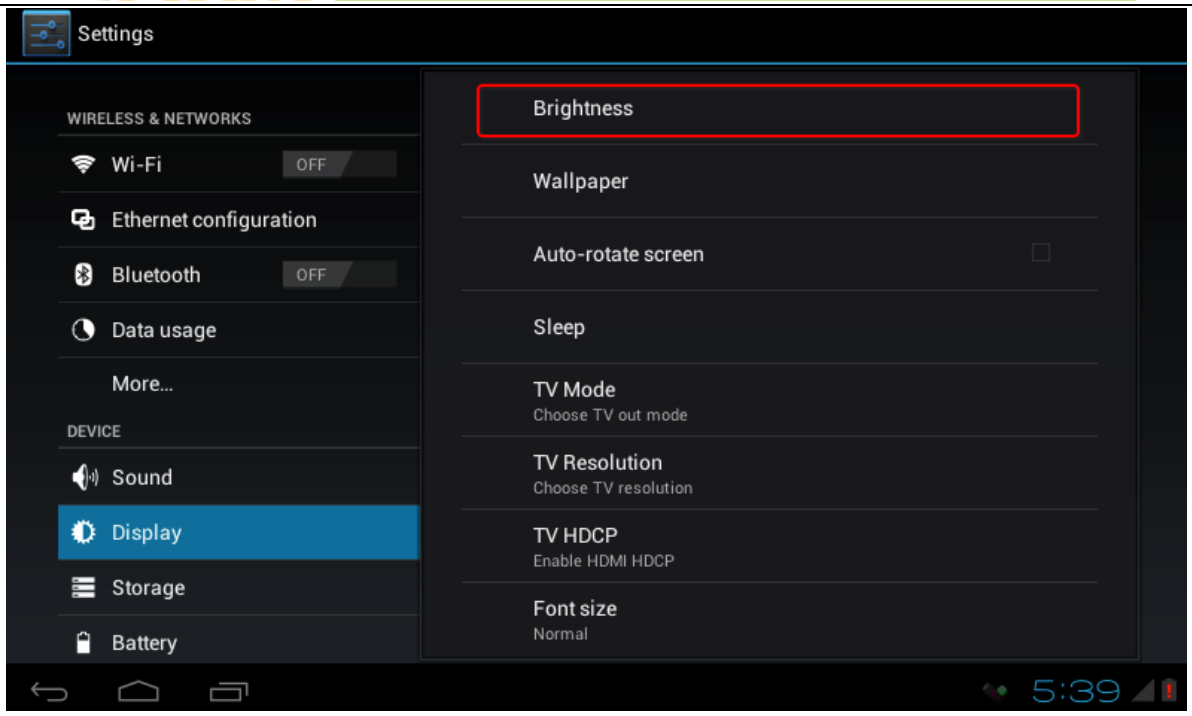
If you click on “Umount USB mass storage” you will safely umount your USB drive. If you click on “Open folder browser” you will lauch the ES file manager. By default the file manager lists all the files under “/sdcard”. You can go back to the root directory and then to “udisk” to browse your USB drive:



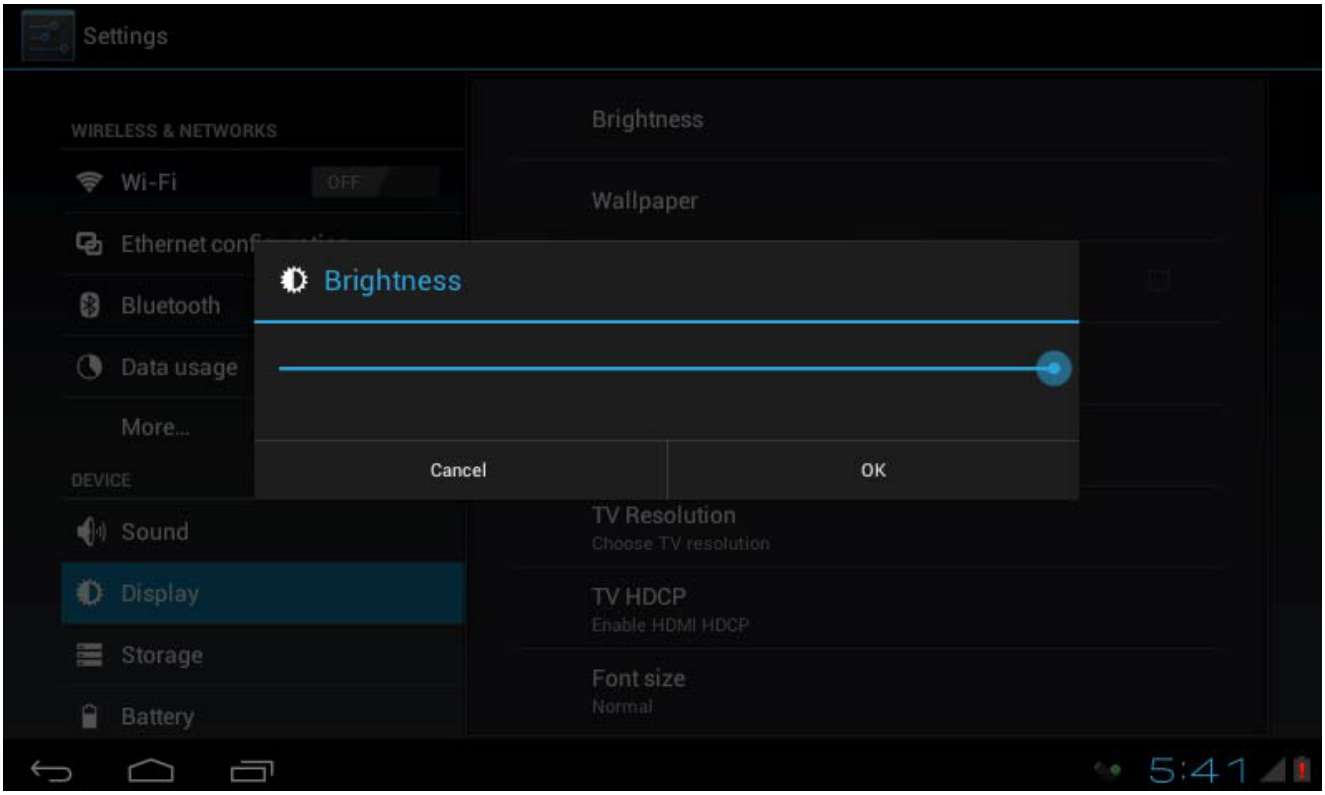


5.2.18 Backlight Control

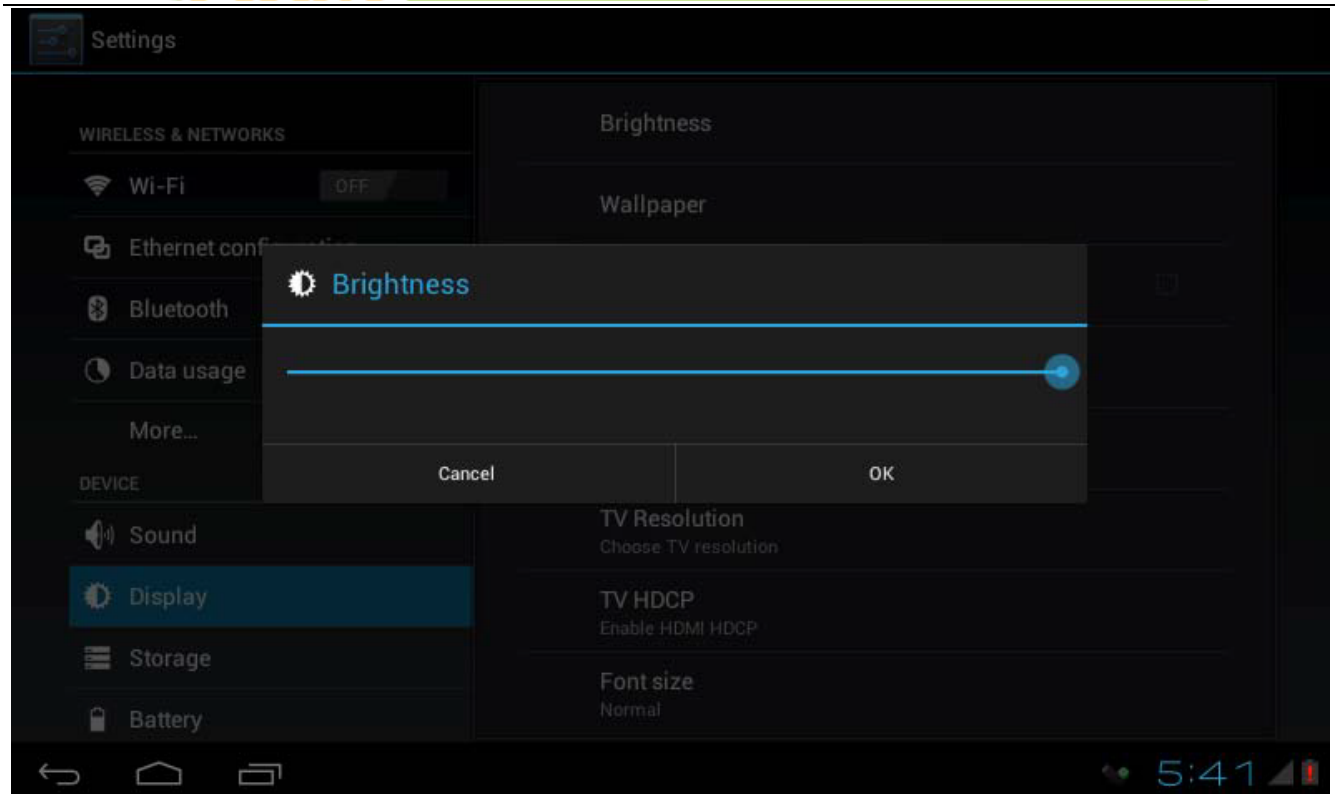
You may have noticed that after the system boots the backlight will turn off gradually if the touch screen doesn't receive any touch. This is manipulated by the backlight control. Please go to "Display" -> "Brightness".



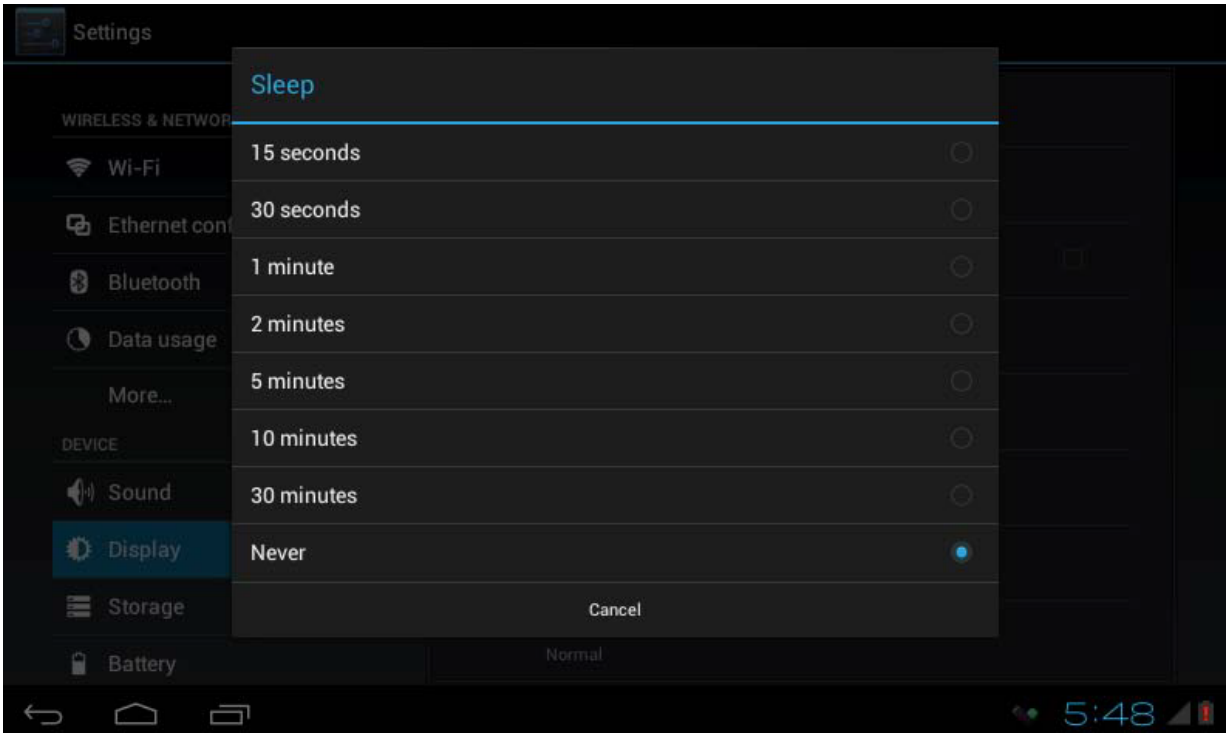
Click on “Brightness” you will get the following window



Click on “Brightness” you can set its brightness

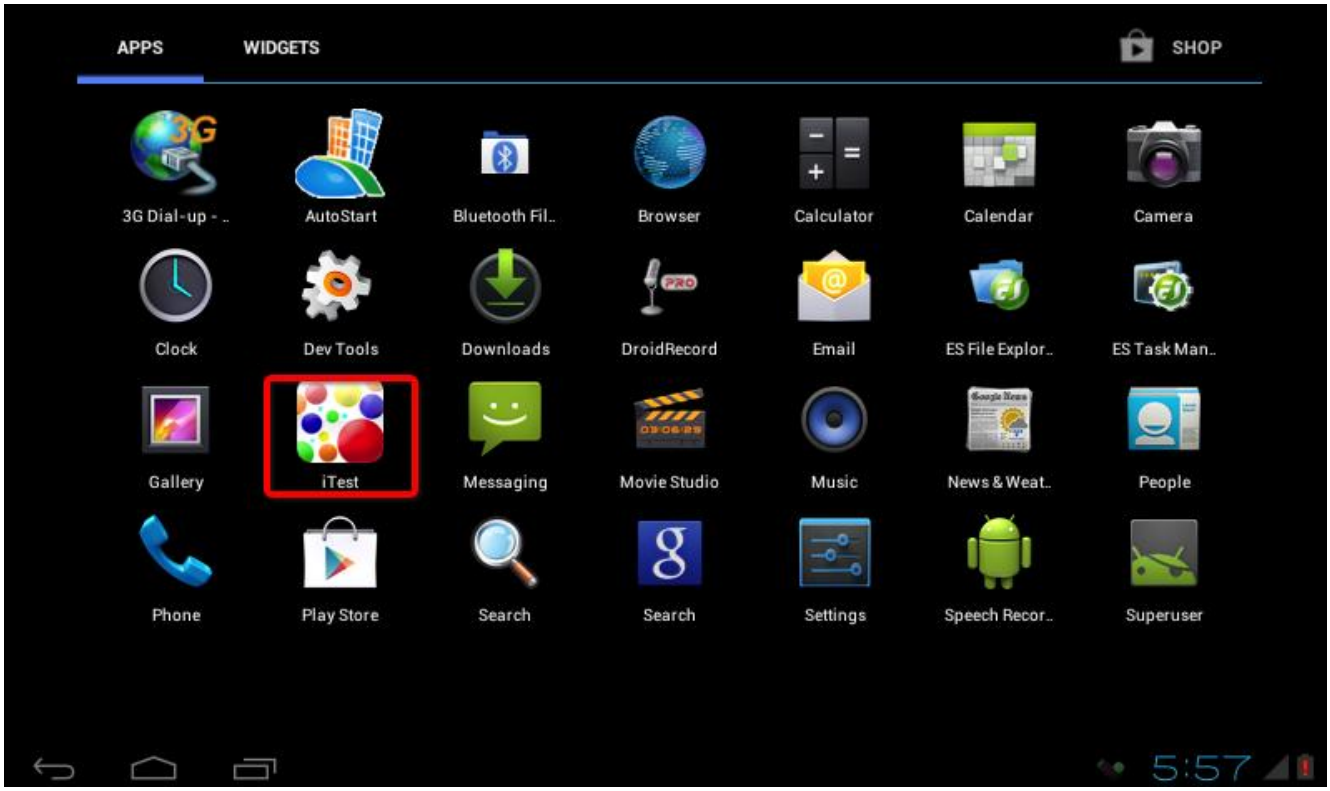


By clicking on “Sleep” you can set a time period after which the backlight will be turned off.

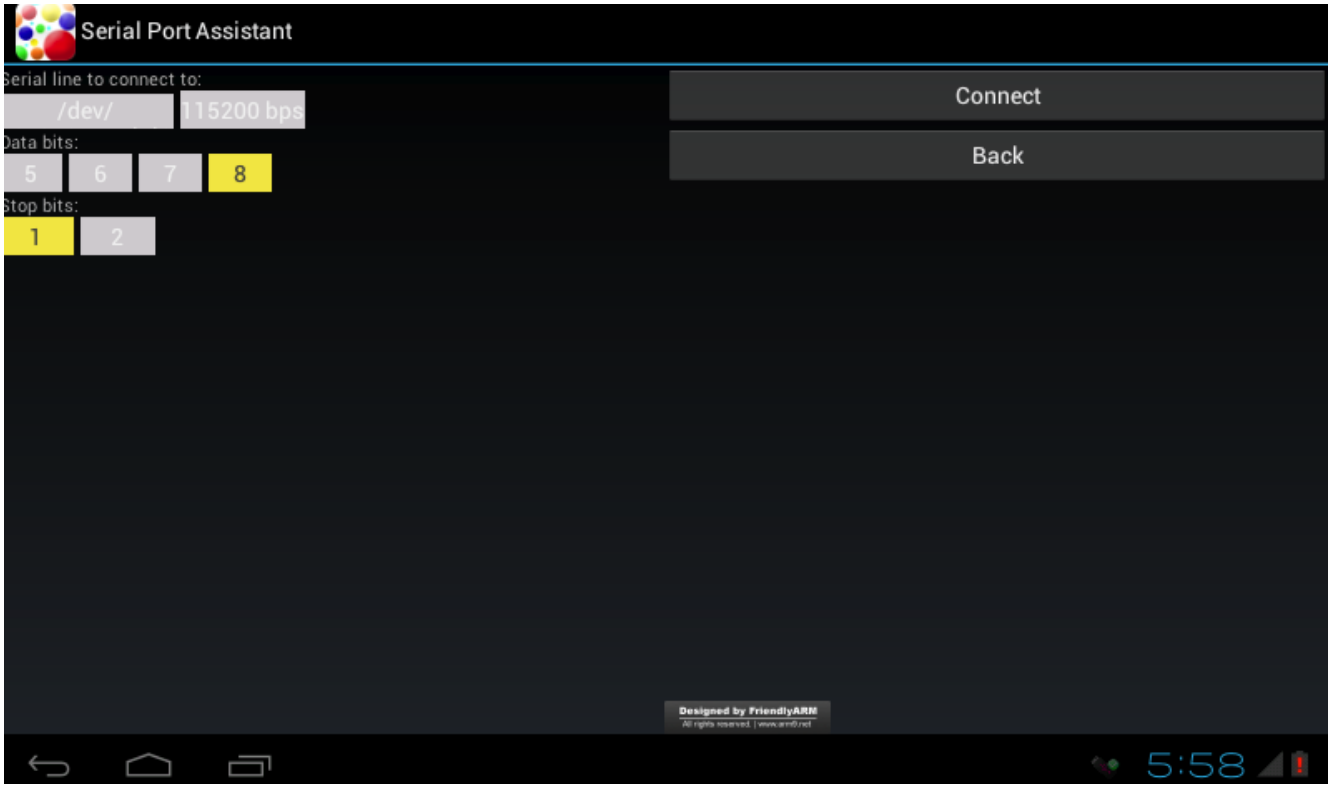


5.2.19 Serial Port Assistant

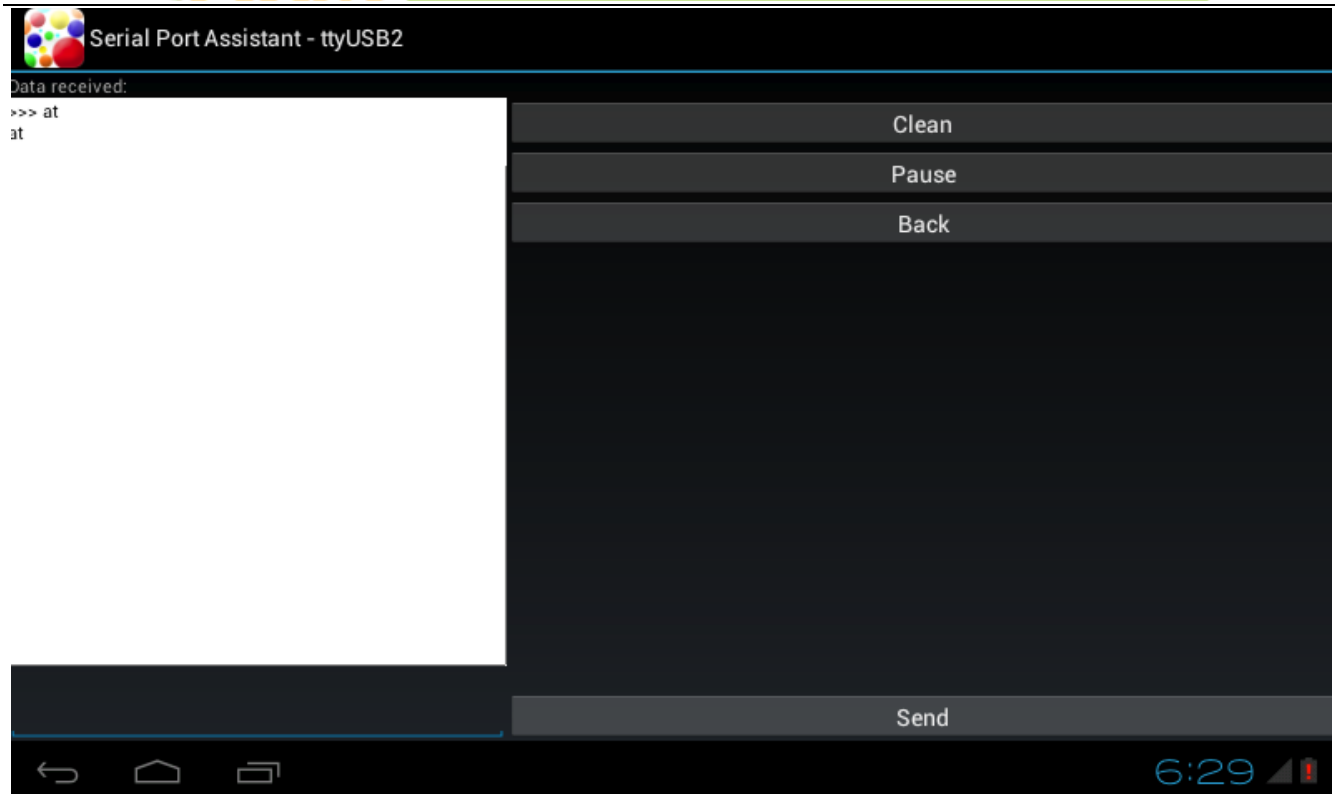
To launch our serial port assistant utility, you can click on the “iTest” icon



Click on “Serial Port Assistant” and you can set its parameters as follows:



After setup is done, click on “Connect” and if the connection is successful you will see the following messages from the serial port



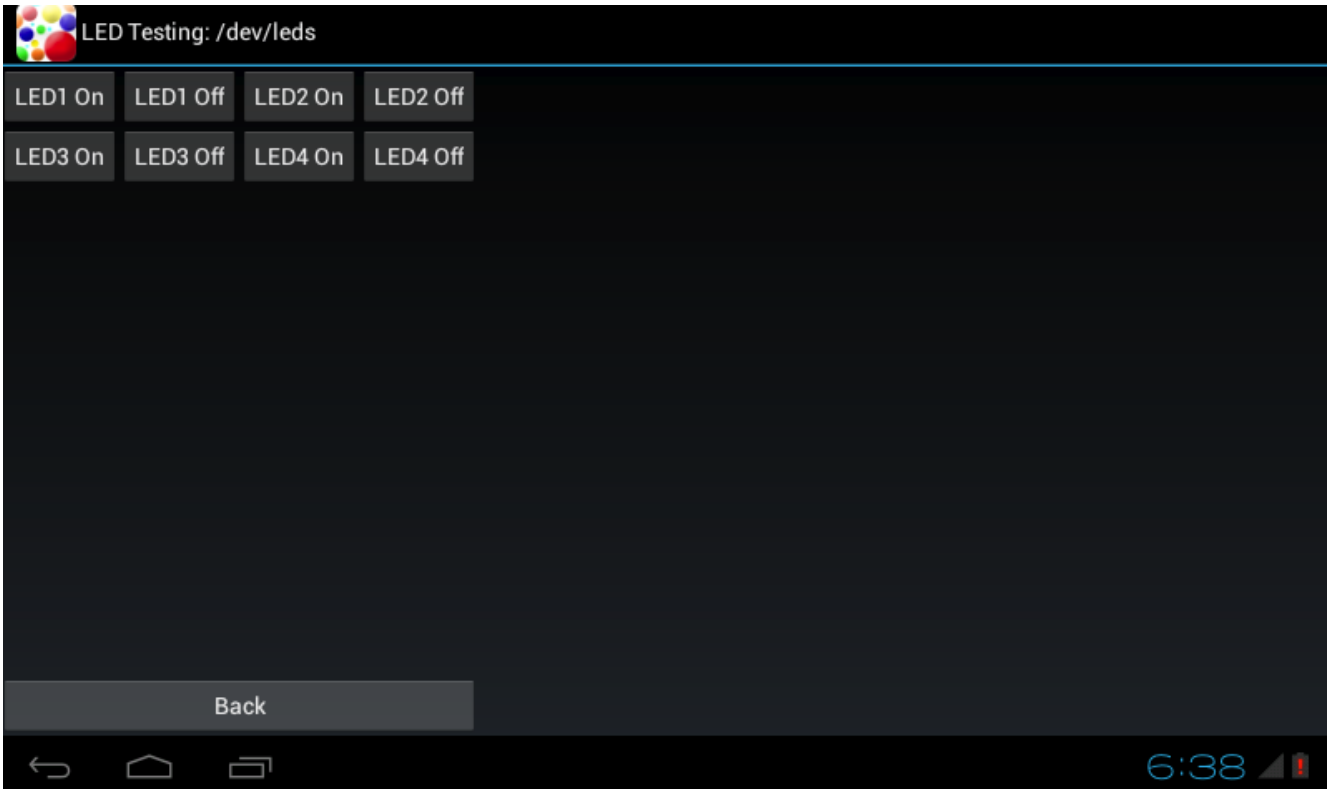
To send data to the serial port, you can type your messages in the left text box and click on “send”. Clicking on “Pause” pauses message sending and “Clean” removes all the received messages

Note:

1) If a serial port doesn’t work you can check whether it is occupied by other applications by commanding “fuser filename”.
2) If a serial port is not occupied and doesn’t work you can check whether it has read and write access by commanding “ls -l” and “chmod 777” to set its access.
3) By default “s3c2410_serial0” is associated with COM1 which is for debugging therefore it is not commended to use this serial port

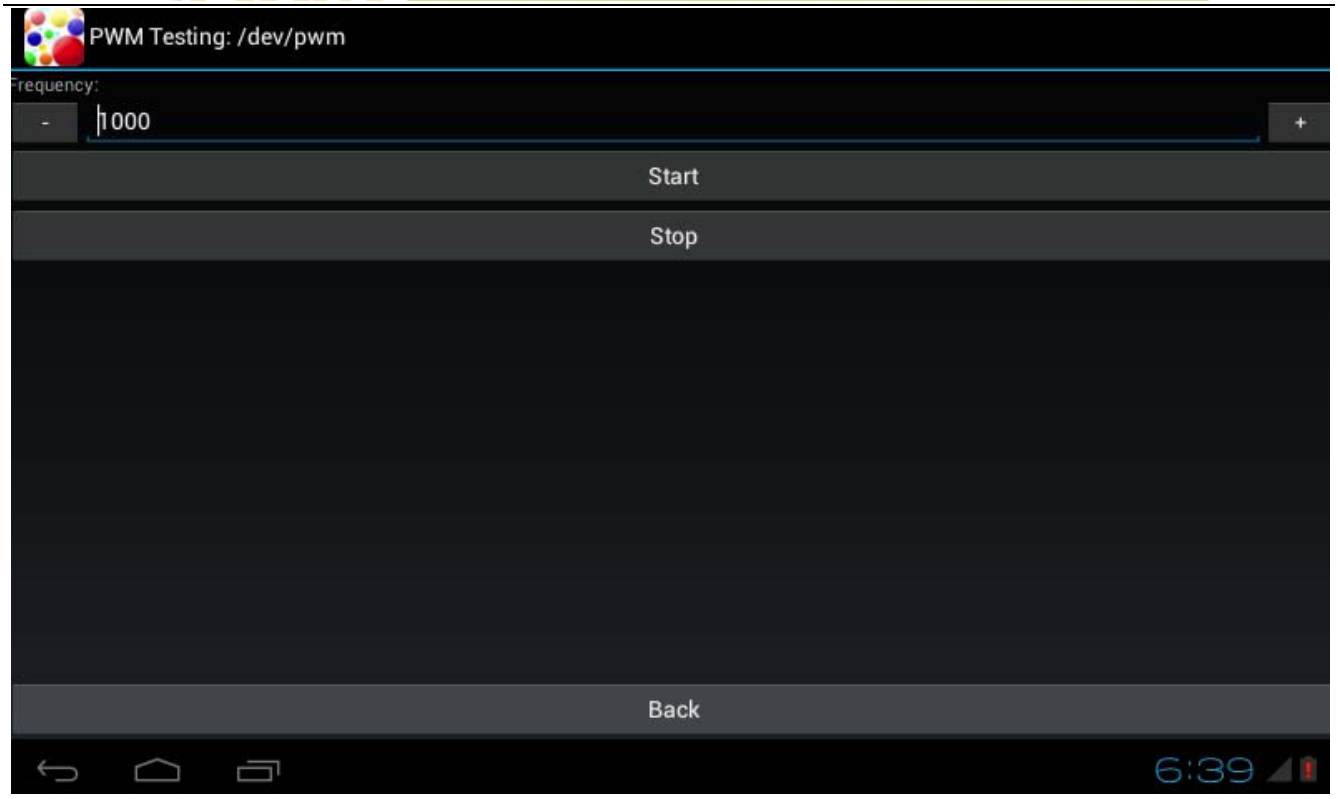
5.2.20 LED Testing

To test LEDs, please click on the “iTest” icon



5.2.21 PWM Buzzer

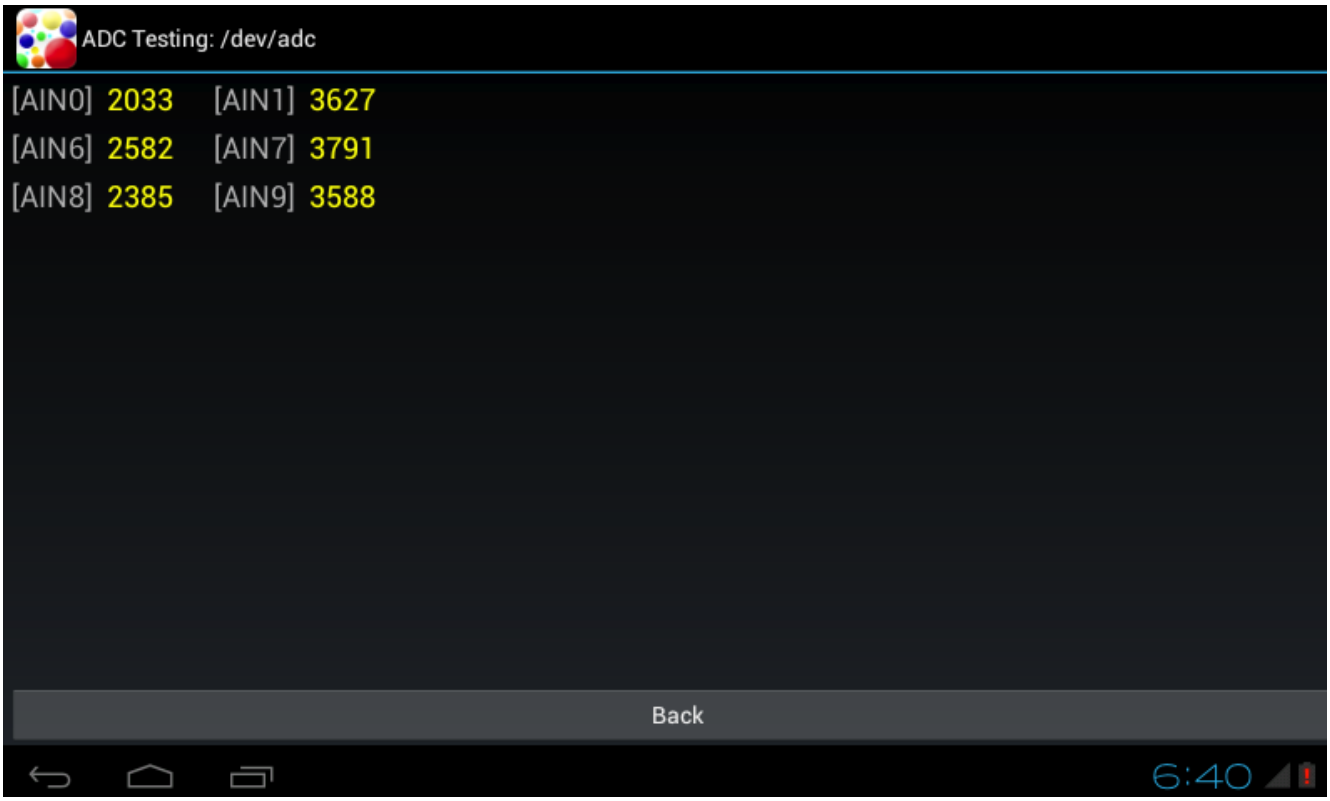
To test PWM, please click on the iTest icon



On the window, you can type a frequency and “start” or adjust the frequency by clicking on “+” and “-”. To stop it you can click on “stop”.

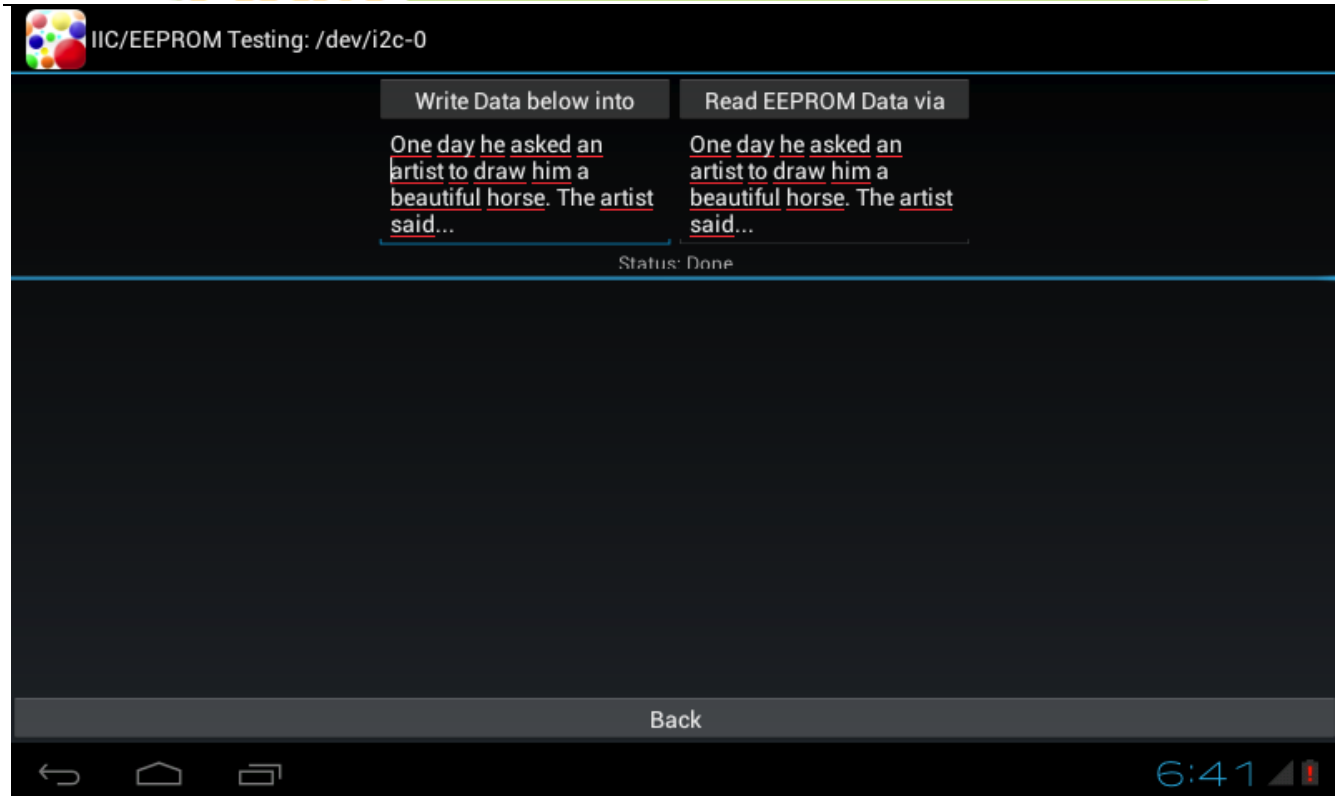
5.2.22 ADC Testing

To test ADC, please click on the “iTest” icon



5.2.23 I2C-EEPROM Testing

To test “I2C-EEPROM” please click on the “iTest” icon



Click on "Write Data below into EEPROM" to write your data on the left to "EEPROM" and then clickon "Read EEPROM Data via IIC" to read it from EEPROM to the right area

5.2.24 Setup Auto Run Utility

Users can set a utility to be automatically started on system boot in "/data/system/autostart_config.xml". Here is a sample of this file:

```
<?xml version="1.0" encoding="UTF-8"?>
<appConfigs>
<appConfig id="1">
<packageName>com.friendlyarm.net3gdialup</packageName>
<mainActivityName>com.friendlyarm.net3gdialup.ActivityMain</mainActivityName>
<autoStart>false</autoStart>
</appConfig>
```

```
<appConfig id="2">
<packageName>com.android.mms</packageName>
<mainActivityName>com.android.mms.ui.ConversationList</mainActivityName>
<autoStart>true</autoStart>
</appConfig>
</appConfigs>
```

The utility's name is bold is the one that will be automatically started on system boot.

5.3 Set up Android Development Environment

What we mean by "set up Android Development Environment" includes the following three steps:

Step1: install Fedora14 (32bit)

Step2: install cross compilers

Step3: install mktools tool chains

(Note: the development environment was tested on both Fedora14 and Fedora15 and it worked on both systems. The following steps were for Fedora14. We didn't test it on other operating systems and we recommend Fedora14 and Fedora15)

5.3.1 Install Fedora14

Please download a Fedora14 installation image and install Fedora14. Fedora's official website is: http://fedoraproject.org/. We strongly suggest users install all software components except server components such as DNS, DHCP and so on.

Note: **Fedora14** should be a **32bit** version, please don't install a 64bit system.

The difference between Fedora14 and Fedora9 is that in Fedora14 users by default

don't login GUI as a root. This causes some inconveniences. Please follow the steps below to enable root login:

In Fedora14 open a terminal and type the command below:

```
# sudo vim /etc/pam.d/gdm
```

Open the gdm file and locate the following line and comment it by adding a "#"

```
#auth       required    pam_succeed_if.so user != root quiet
```

Save it and edit "/etc/pam.d/gdm-password"

```
sudo vim /etc/pam.d/gdm-password
```

Locate the following line and comment it by adding a "#"

```
#auth       required    pam_succeed_if.so user != root quiet
```

Save it, reboot the system and select "root" when login

5.3.2 Setup Android Compiler

We used arm-linux-gcc-4.5.1 and it by defauly supports armv6 command sets. The following steps will introduce how to build a compile environment.

Step 1: copy the compressed file "arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz" in the shipped CD into a system's directory, e.g "tmp\", enter this directory and execute the following commands:

#cd /tmp
#tar xvzf arm-linux-gcc-4.5.1-v6-vfp-20101103.tgz –C /

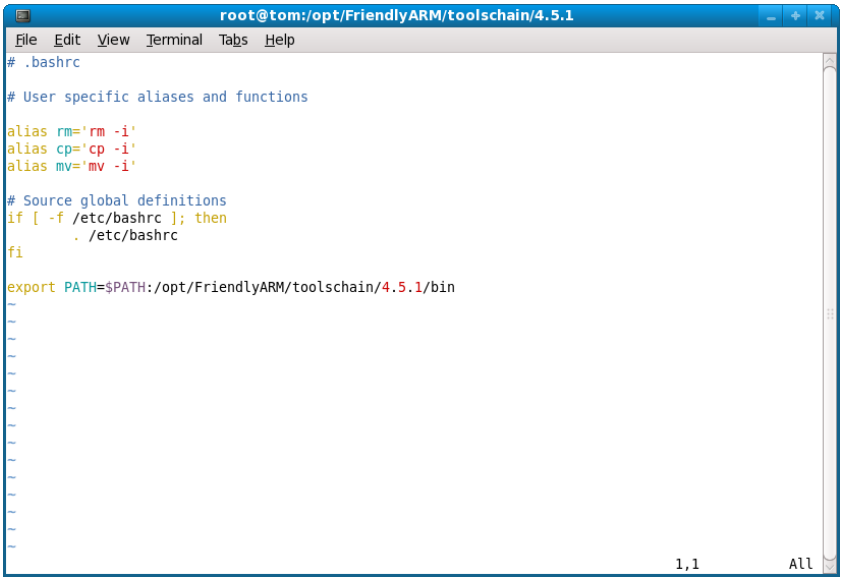
Note: there is a space after "C" and "C" is a capital letter.

These commands will install "arm-linux-gcc" in the "**/opt/FriendlyARM/toolschain/4.5.1**"

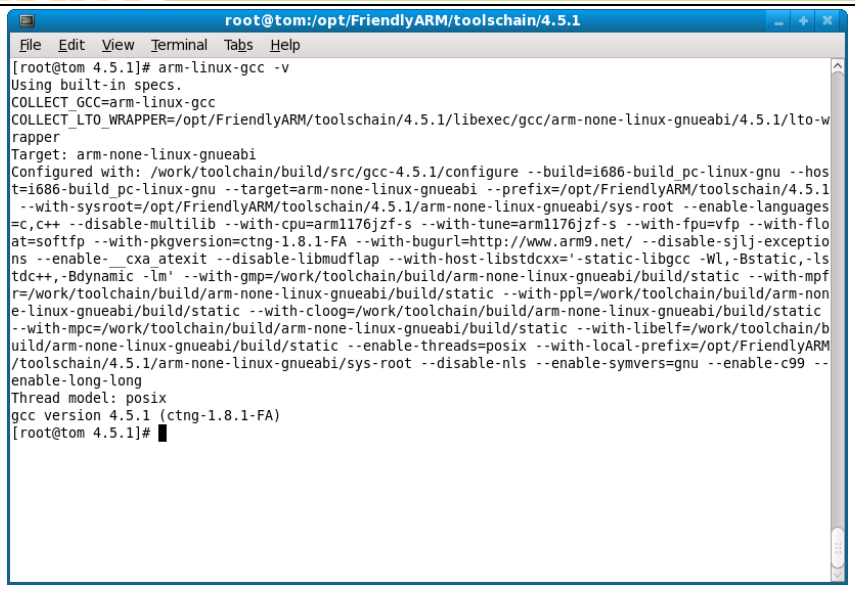
Step 2: run the command below to add the compiler's path to system variables:

#gedit /root/.bashrc

This is to edit the "/root/.bashrc" file (there is a "." before "bashrc"). Update the last line with "**export PATH=$PATH:/opt/FriendlyARM/toolschain/4.5.1/bin**" in the opened file, save and exit the file



Logout and login the system again (no need to reboot the system, just go to "start"-> "logout"), the above settings will take into effect. Type "arm-linux-gcc –v", if the messages depicted in the screen shot below appear, it indicates the compile environment has been set up successfully.



5.3.3 mktools tool chain

To burn a target file system to the board you need to make an image first. The "mkyaffs2image-128M" is for for this. It is for creating an yaffs2 image whose page format is "1 Page= 2K Byte" and block format is "1 Block=128K" for SLC Nand Flash such as K9F2G08, K9F4G08 and K9K8G08.

You can copy "mktools-YYYYMMDD.tar.gz" in the Android directory in the shipped CD to /tmp and run the following command to install the mktools tool chain.

#tar xvzf /tmp/mktools-YYYYMMDD.tar.gz –C /

This will create tools chain in the "/usr/sbin" directory.

Note: "C" is capitalized and means "change". If your system has been installed a Mini2440 or Mini6410's mkyaffs2image it will be overwritten. But you don't need to worry about it since they are identical

5.3.4 Uncompress Source Code and Install Application Utilities

Firstly, create a working directory: /opt/FriendlyARM/mini210/android by running the following command

#mkdir –p /opt/FriendlyARM/mini210/android

All the source code in the following steps will be uncompressed in this working directory

(1) Get a Copy of Android Souce Code Package

Create a temporary directory "**/tmp/android**" in Fedora9

#mkdir /tmp/android

Copy all the files under "Android" from the shipped CD to "**/tmp/Android**"

(2) Uncompress Android Kernel

Execute the commands below in "/opt/FriendlyARM/mini210/android"

```
#cd /opt/FriendlyARM/mini210/android
#tar xvzf /tmp/android/ linux-2.6.35.7-android-2011-09-30.tgz
```

This will create a "linux-2.6.35-android" directory which contains a complete copy of source code

Note: 2011-09-30 is the date when we released it

(3) Uncompress Android System

Execute the commands below in "/opt/FriendlyARM/mini210/android"

```
#cd /opt/FriendlyARM/mini210/android
#tar xvzf /tmp/android/android-2.3.1-fs-20110925.tar.gz
```

This will create an "Android-2.3.1" directory

Note: 20110925 is the date when we released it. This source code contains a copy of Android-2.3.1 source code and compiling scripts

(4) Uncompress Android File System

Execute the command below in "/opt/FriendlyARM/mini210/android"

```
#cd /opt/FriendlyARM/mini210/android
#tar xvzf /tmp/android/ rootfs_android-20110925.tar.gz
```

This will create a rootfs_android directory

Note: 20110925 is the date when we released it.

5.4 Configure and Compile Linux Kernel

Android's Linux kernel is a little bit different from a standard one but its configuration utility is the same. If you are not familiar with configuring a Linux kernel we suggest you use our default configuration file

To compile a kernel for an N43 LCD system please follow the steps below

```
#cd /opt/FriendlyARM/mini210/android/ linux-2.6.35.7-android
#cp mini210_android_defconfig    .config    ; note: there is a "." before "config"
```

You can run "make menuconfig" to configure the kernel and run "make" to compile":

```
#make
```

This will generate a zImage under "arch/arm/boot".

5.5 Create Android

Compiling Android may not be an easy task for beginners. Therefore we have a complete copy of the source code and two compiling scripts: build-android and genrootfs.sh.

Execute the commands below:

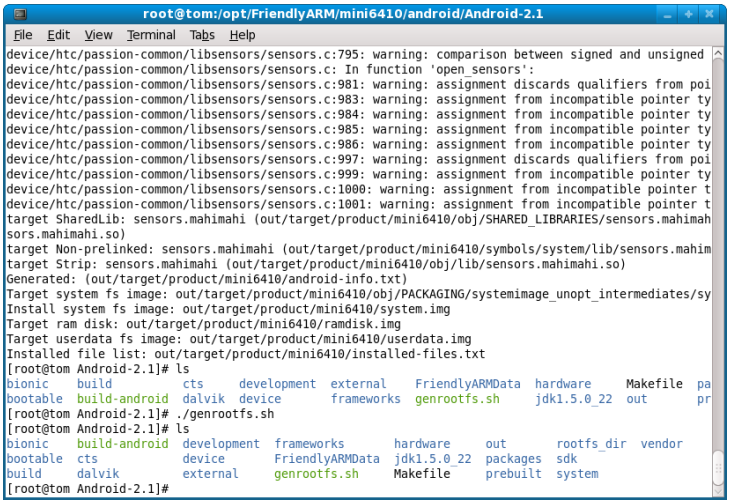
```
#cd /opt/FriendlyARM/mini210/android/Android-2.3.1
#./build-android
```

This will begin to compile Android-2.3.1. This process may take a while. We recommend users to use a multi-core CPU and Linux instead of using a simulator.

After it is done, run the following script:

#./genrootfs.sh

This will create a target file system we need and a "rootfs_dir" directory. It is the same as "rootfs_android".



Run the command below to create a yaffs2 image file:

 # mkyaffs2image-128M rootfs_dir rootfs_android.img

5.6 Create and Run File System

With the **mkyaffs2image-128M** utility, you can make a yaffs2 image. The Android kernel by default supports this file system. Please run the commands below:

#cd /opt/FriendlyARM/mini210/android/Android-2.3.1
#mkyaffs2image-128M rootfs_dir rootfs_android.img

6 Android Application Development

This chapter introduces how to install Android SDK and Eclipse and debug programs for the 210 boards. We wish this could help Android beginners.

6.1 Set up Development Environment for Android Applications

This section will introduce how to setup Android development on **Windows 7** and how to use ADB. Please use **Android2.3 or above. If your Android version is older than 2.3 we recommend you to download and install Android2.3 from www.arm9.net.**

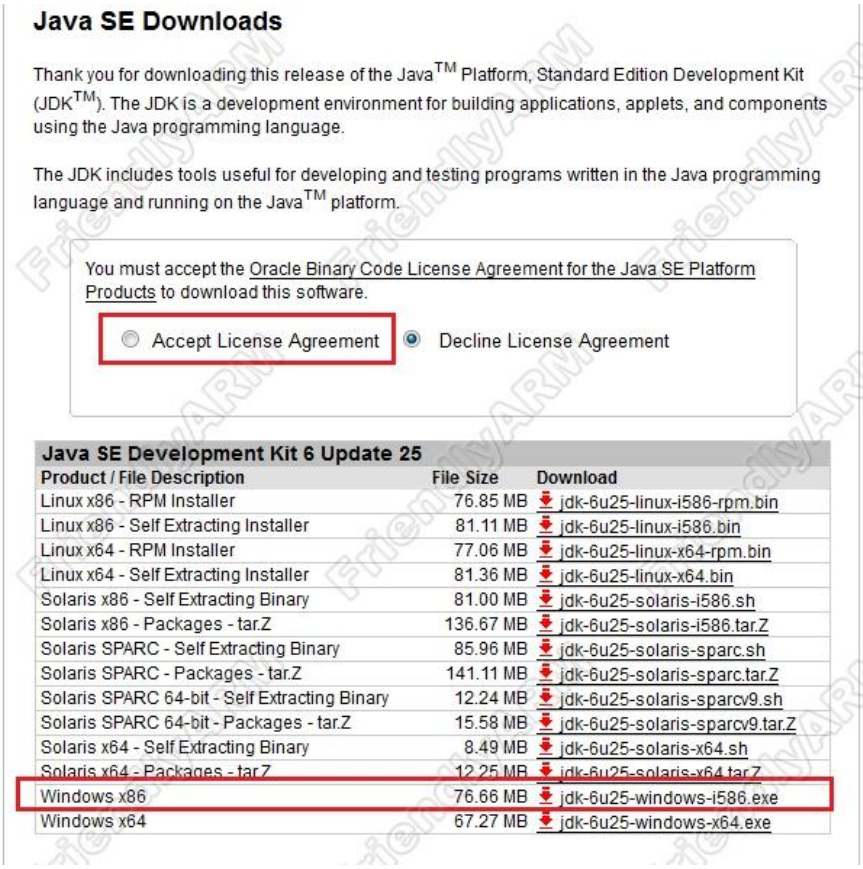
The steps described in this section were tested on **Fedora14**, and it may not apply to Fedora9. All the files can be found in the Android directory in the shipped CDs

6.1.1 Download and Install JDK (Java SE Development Kit)

Since Android SDK and Eclipse are both written in Java we strongly recommend to install this JDK on Windows 7. Please follow the steps below:

Please visit oracle's official site to download a JDK：

Check “Accept License Agreement”



Java SE Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java™ platform.

You must accept the Oracle Binary Code License Agreement for the Java SE Platform Products to download this software.

Accept License Agreement Decline License Agreement

Java SE Development Kit 6 Update 25

Product / File Description	File Size	Download
Linux x86 - RPM Installer	76.85 MB	jdk-6u25-linux-i586-rpm.bin
Linux x86 - Self Extracting Installer	81.11 MB	jdk-6u25-linux-i586.bin
Linux x64 - RPM Installer	77.06 MB	jdk-6u25-linux-x64-rpm.bin
Linux x64 - Self Extracting Installer	81.36 MB	jdk-6u25-linux-x64.bin
Solaris x86 - Self Extracting Binary	81.00 MB	jdk-6u25-solaris-i586.sh
Solaris x86 - Packages - tar.Z	136.67 MB	jdk-6u25-solaris-i586.tar.Z
Solaris SPARC - Self Extracting Binary	85.96 MB	jdk-6u25-solaris-sparc.sh
Solaris SPARC - Packages - tar.Z	141.11 MB	jdk-6u25-solaris-sparc.tar.Z
Solaris SPARC 64-bit - Self Extracting Binary	12.24 MB	jdk-6u25-solaris-sparcv9.sh
Solaris SPARC 64-bit - Packages - tar.Z	15.58 MB	jdk-6u25-solaris-sparcv9.tar.Z
Solaris x64 - Self Extracting Binary	8.49 MB	jdk-6u25-solaris-x64.sh
Solaris x64 - Packages - tar.Z	12.25 MB	jdk-6u25-solaris-x64.tar.Z
Windows x86	76.66 MB	jdk-6u25-windows-i586.exe
Windows x64	67.27 MB	jdk-6u25-windows-x64.exe

For Windows7 32bit systems you need to select “jdk-6u25-windows-i586.exe” to download the JDK. After download is completed, double click on it to install.

After installation is completed you need to **add the JDK command path to the Path environment variable**. Please follow the steps below:

1) Right click on “My Computer” -> Property and select “Advanced System Settings”.

2) Click on “Environment Variables”

3) Find the Path variable in “System Variables”, double click on it and add “C:\Program Files\Java\jdk1.6.0_25\;” at the beginning.

4) Click on “OK” to complete

6.1.2 Download and Install Android SDK

Please go to http://developer.android.com/sdk/ to download the latest Android SDK for windows. Please download the program marked in red in the screenshot shown below:

(Note: up till May 18, 2011, the latest version is **installer_r11-windows.exe**)

From the website users can get the latest version. You can also just use the one in the shipped CD, which is "installer_r11-windows.exe" in the Android directory.

Double click on "installer_r11-windows.exe", follow the prompt to install the program in "C:\Program Files\Android\android-sdk" with the default options. After it is completed the SDK manager will automatically start up.

Note: during the installation if it prompts that JDK cannot be found, you can go "back" and "Next" to try again.

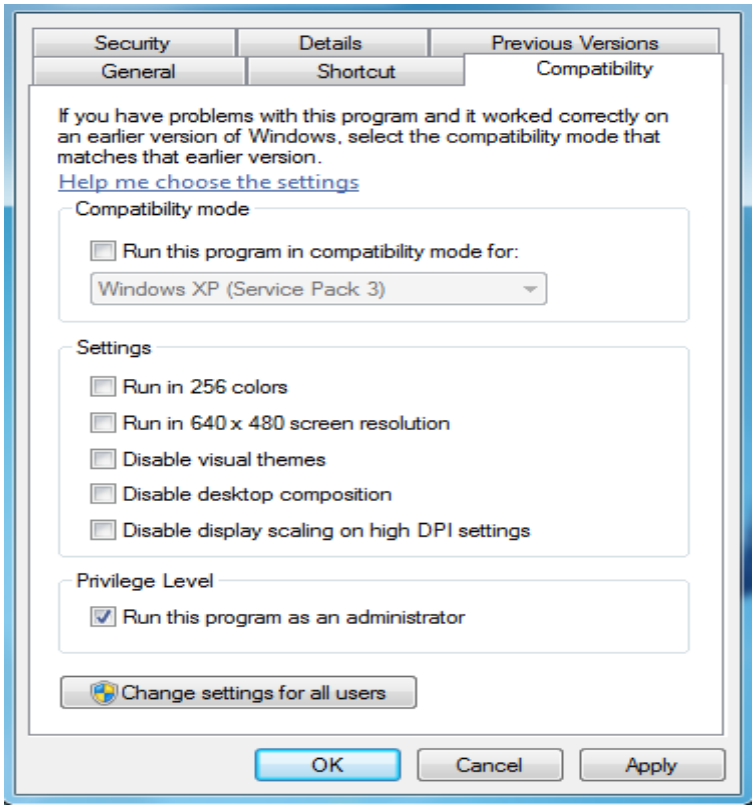
When the SDK Manager starts for the first time it will pop up a "Choose Packages to Install" dialog asking users if there are available Packages to install. Just click on

"Cancel" to ignore it.

Then you can find "Android SDK Tools" in the "Start" menu and click on "SDK Manager" to start it:

By default the SDK is installed in drive C it is better to run it as an administrator otherwise data wrting will fail due to limited access to drive C. Please follow the steps below to set up the SDK:

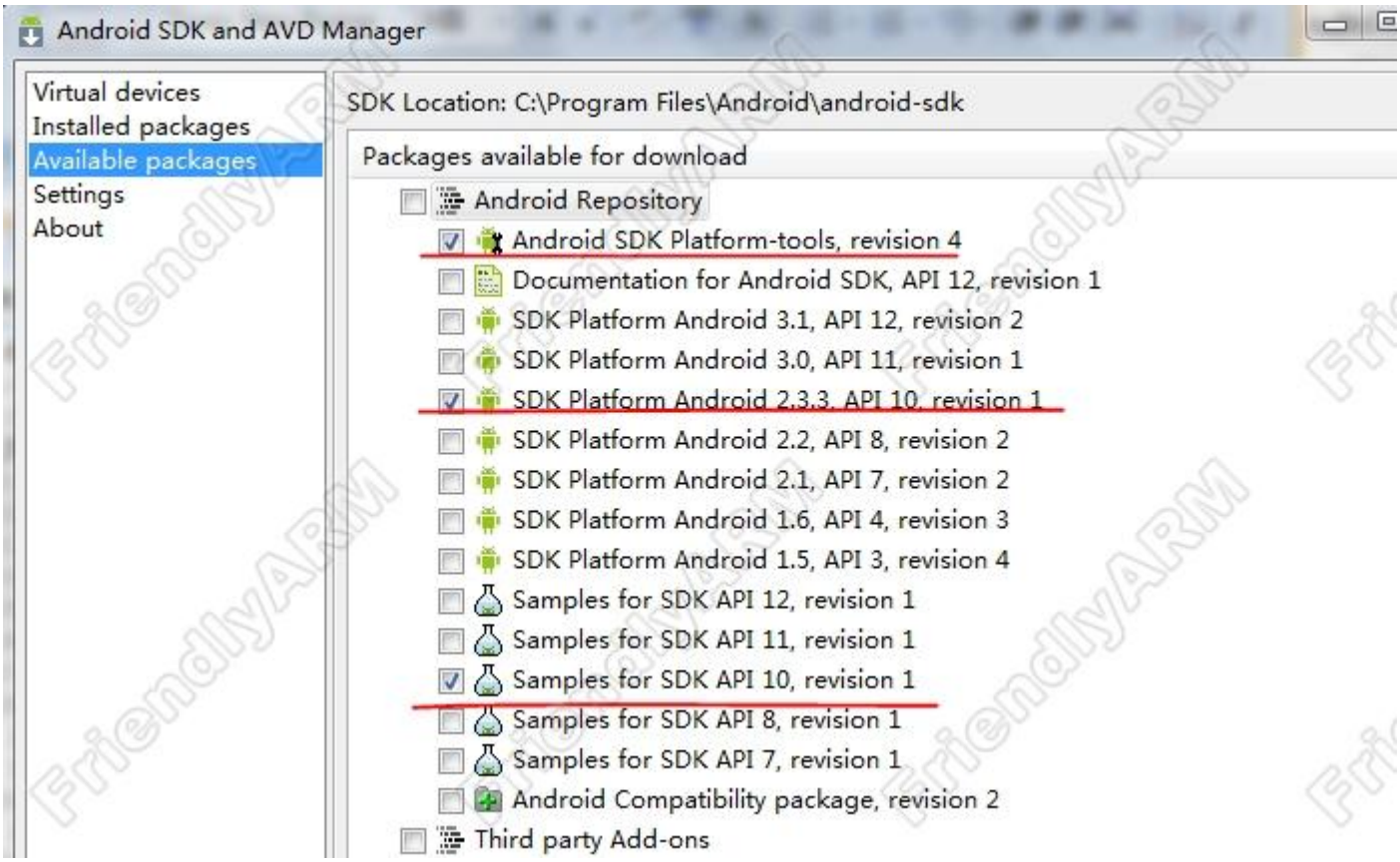
1) Right click on "SDK Manager" in the start menu and click on "Property"
2) Click on "Compability", check "Run this program as an administrator" and click on "OK"



Now you can run the SDK you will be prompted to confirm if you want to run it as an administrator. Please click on "Yes" to go on.

6.1.3 Download and Install Android 2.3 Packages

Run the SDK Manager as an administrator, select "Available Packages", and click on the ">" icon beside "Android Repository" to expand it and check the options shown below:



Click on "Install Selected", select "Accept All" in the "Choose Packages to Install" dialog and click on "Install" to begin download. This may take a while.

After download is completed a message box will pop up asking you whether you

want to restart ADB, just click on “Yes” to continue.

6.1.4 Install Eclipse

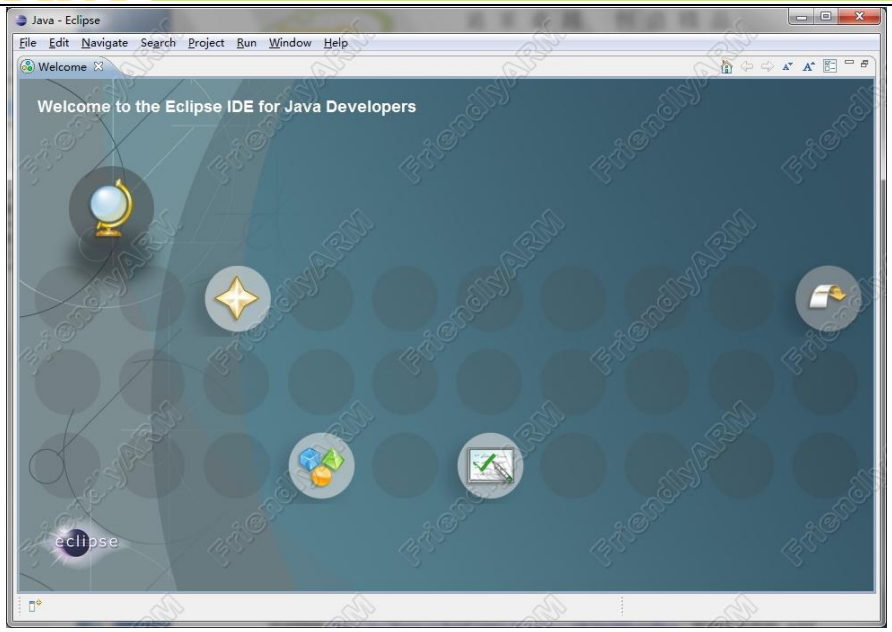
Eclipse is a very popuplar IDE for Android. Please visit http://www.eclipse.org/downloads/ to download it.

On the download page select “Eclipse IDE for Java Developers”

You can also use the version “eclipse-java-helios-SR2-win32.zip” in the Android directory in the shipped CD.

Please unzip your package (ours was “eclipse-java-helios-SR2-win32.zip”) in drive D:\, go to the “eclipse” directory and double click on “eclipse.exe” to start it.

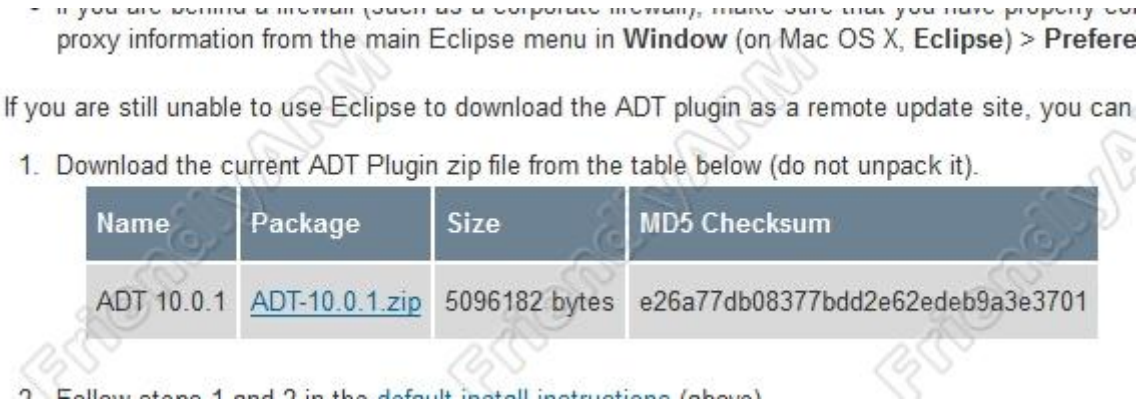
On the initial startup it will ask you to set up the “Workspace” path in which all your created projects’ source code will be saved. After Eclipse starts it will show the following main window:



6.1.5 Install Android Plugins

By following the above steps we have installed an Android SDK and Eclipse. In order to use the Android SDK in Eclipse you need to install an ADT plugin as well:

Please visit http://developer.android.com/sdk/eclipse-adt.html#installing, find the link for ADT and download it.

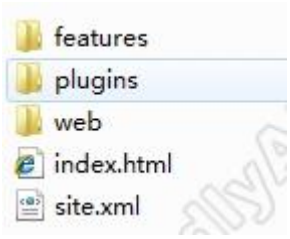

proxy information from the main Eclipse menu in **Window** (on Mac OS X, **Eclipse**) > **Prefere**

If you are still unable to use Eclipse to download the ADT plugin as a remote update site, you can

1. Download the current ADT Plugin zip file from the table below (do not unpack it).

Name	Package	Size	MD5 Checksum
ADT 10.0.1	ADT-10.0.1.zip	5096182 bytes	e26a77db08377bdd2e62edeb9a3e3701

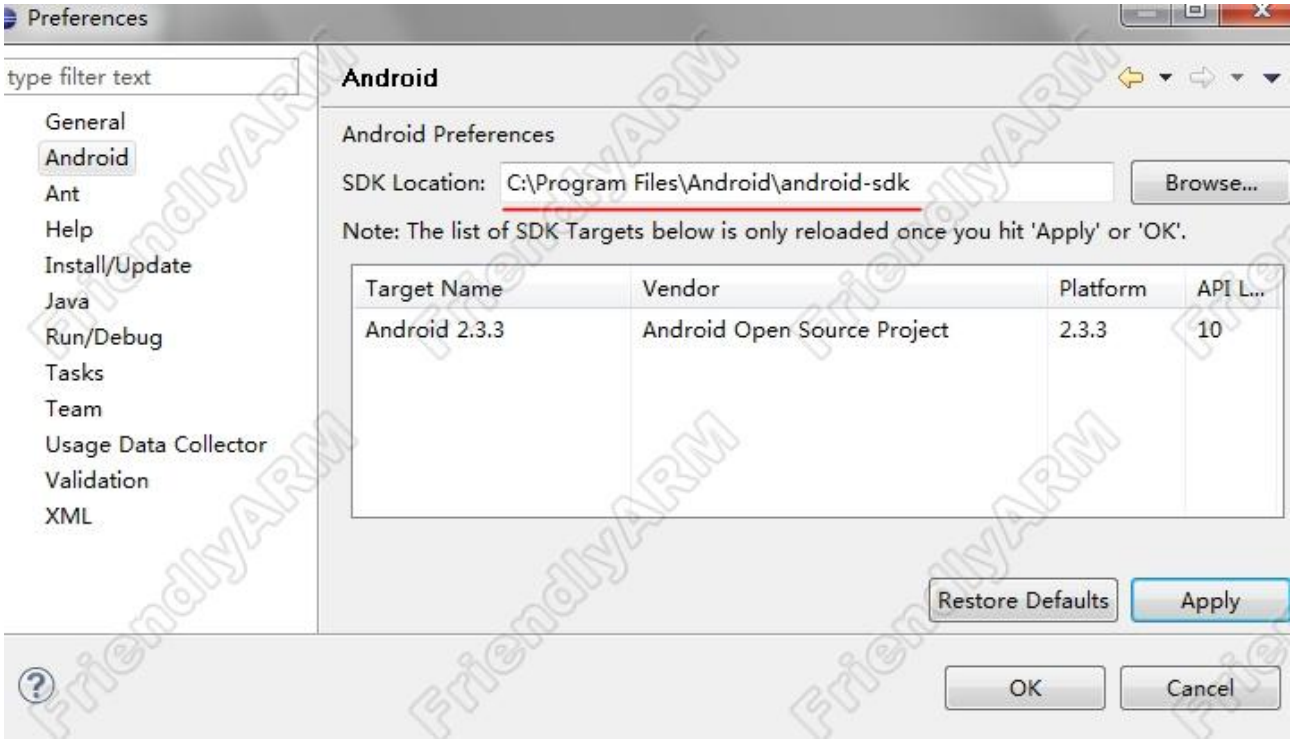
After download is completed uncompress it you will get the following files:



Exit Eclipse, replace the Eclipse directory with these files to complete the installation.

6.1.6 Configure Eclipse

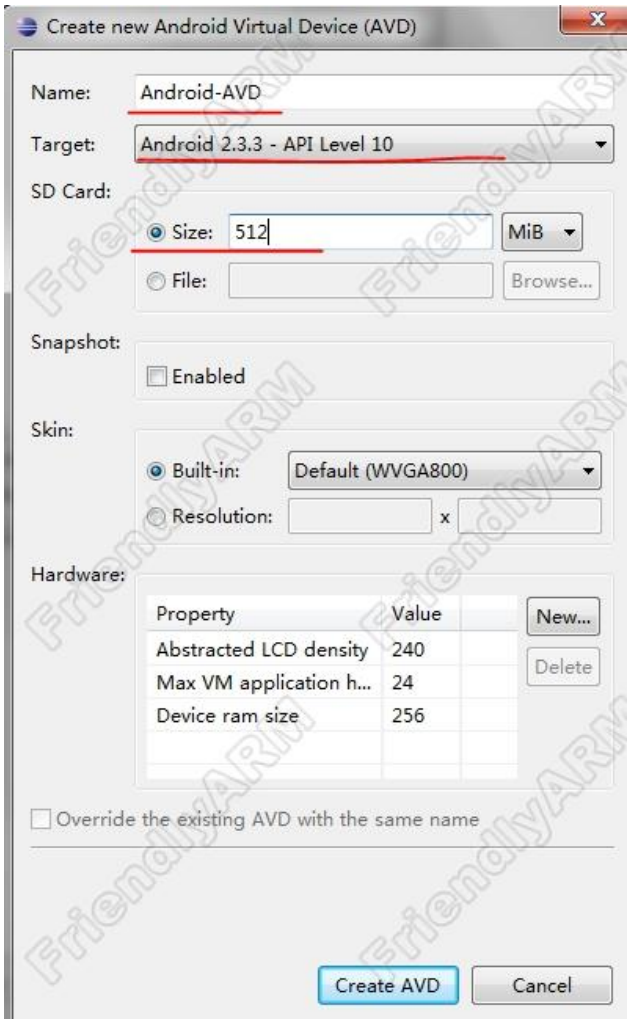
Start Eclipse, click on “Window” on the main page and select “Preferences”. On the left side select “Android” and click on “Browser” to point to the installation path. By default it is “C:\Program Files\Android\android-sdk”:



Click on “OK” to complete.

6.1.7 Set up Android Simulator

Move to “Android SDK Tools” in the start menu and click on “SDK Manager”. Click on “New…” in the “Android SDK and AVD Manager” dialog and a “Create new Android Virtual Device(AVD)” dialog will pop up. Input “Android-AVD” in the Name field, select “Android 2.3.3 - API Level 10” in the “Target” field, input “512” in the “SD Card” field, keep all the other settings with default options and click on “Create AVD”.



After the configuration is done a simulator will be listed as follows:

List of existing Android Virtual Devices located at C:\Users\tzs\.android\avd

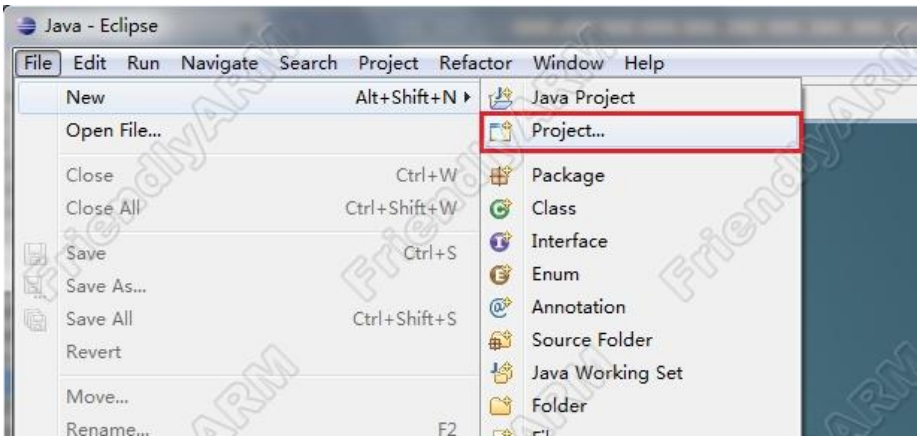
AVD Name	Target Name	Platform	API Level
✓ Android-AVD	Android 2.3.3	2.3.3	10

Select your simulator, click on "Start" and "Launch" in the "Launch Option" dialog to start it:

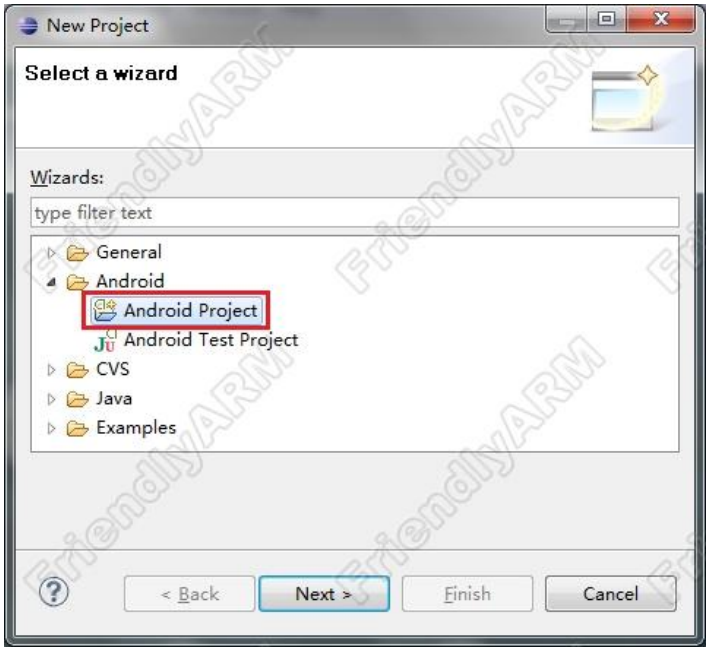
6.1.8 Create Android Program

6.1.9 Create HelloMini210 Project

In this section we will create an Android project "HelloMini210" to test our development environment. Start Eclipse and go to "File->New->Project...".



In the “New Project” dialog, select “Android->Android Project” and click on “Next”



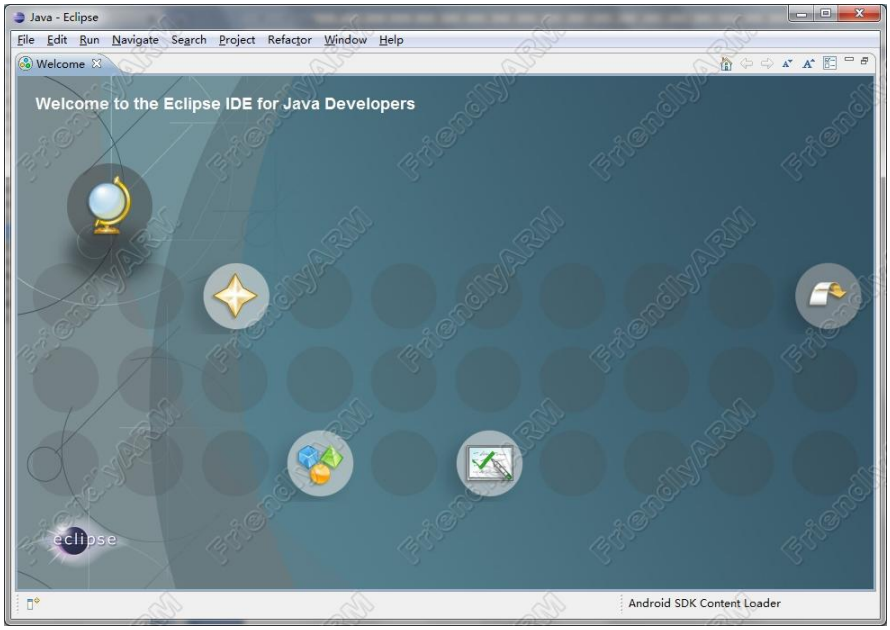
In the “New Android Project” dialog type the following information or check the following options:

1) Project Name: HelloMini210
2) Build Target: Android 2.3.3
3) Appication name:HelloMini210

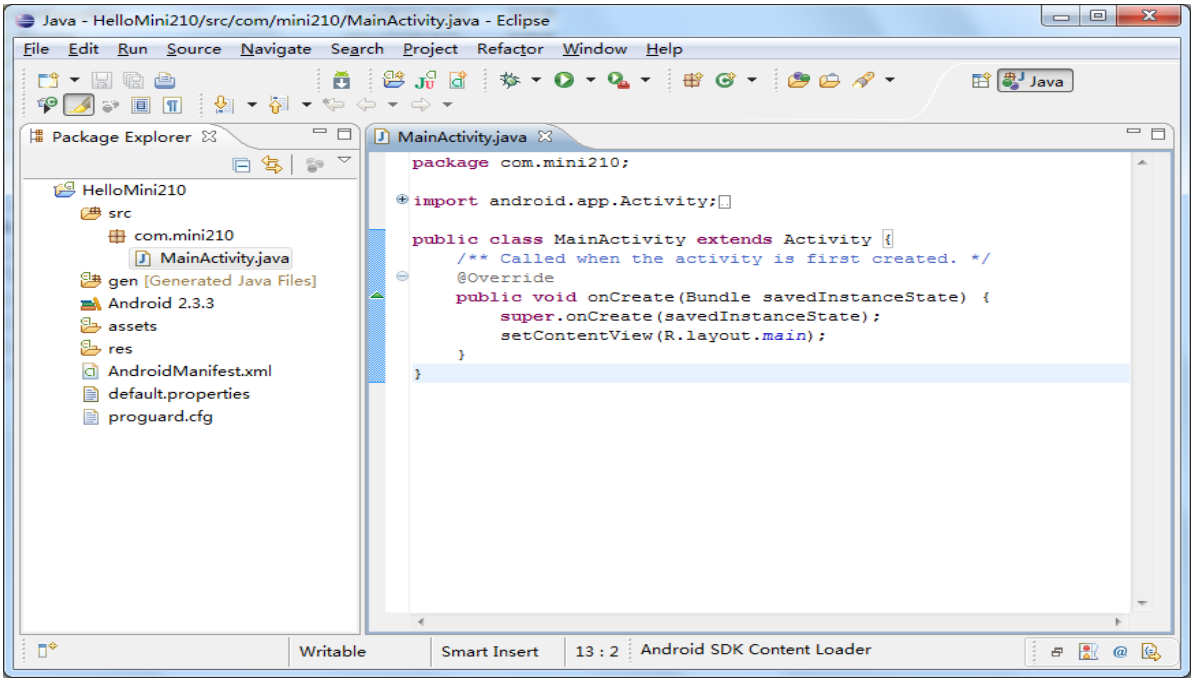
4) Package name:com.mini210

5) Create Activity:MainActivity

Click on “Finish” to complete the wizard and return to the main window:



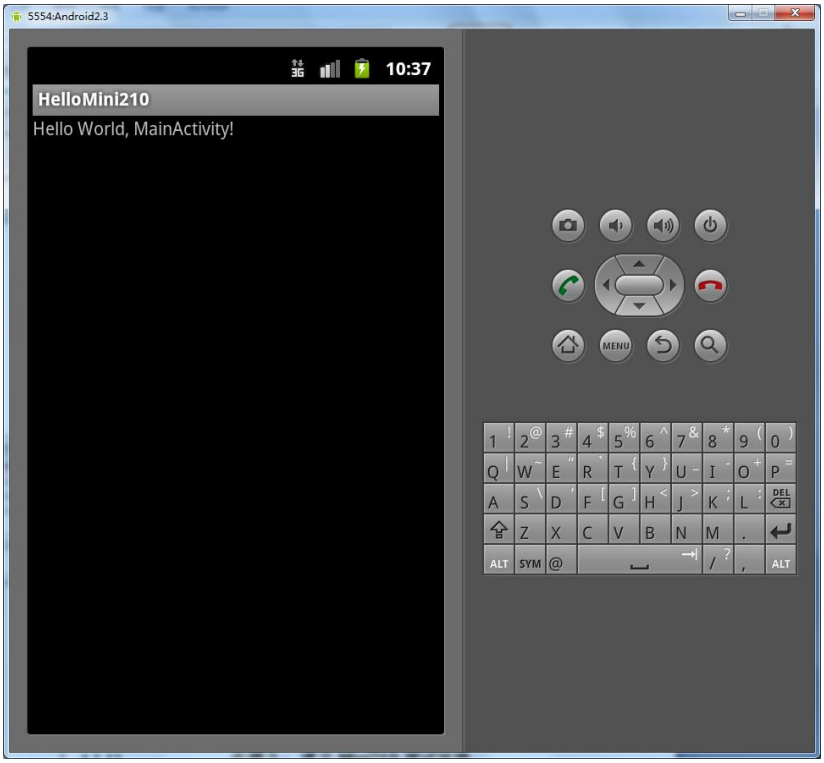
Close the “Welcome” page and the project view will be presented. On the left side click on “src -> com.mini210 -> MainActivity.java” you will see the source code:



6.1.10 Run HelloMini210 in Simulator

To compile and run “HelloMini210” please select the “HelloMini210” project in “Package Explorer”and then click on the “run” button or go to “Run->Run As->Android Application”.

The Android simulator will automatically start and it may take a while. After it fully loads it will run “HelloMini210”:



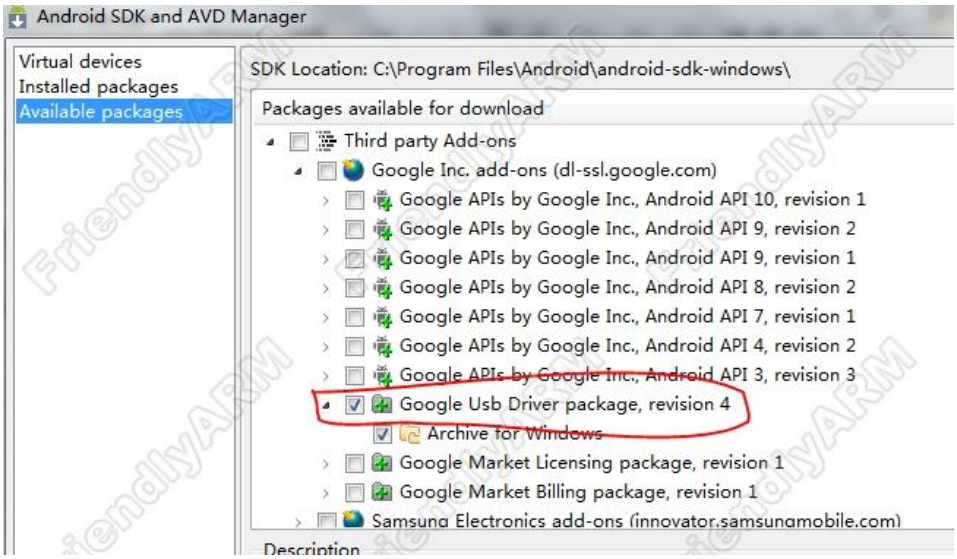
In the following sections we will debug and run this program on the Tiny210

6.1.11 Set up Android Debug Environment

6.1.11.1 Install USB ADB Driver

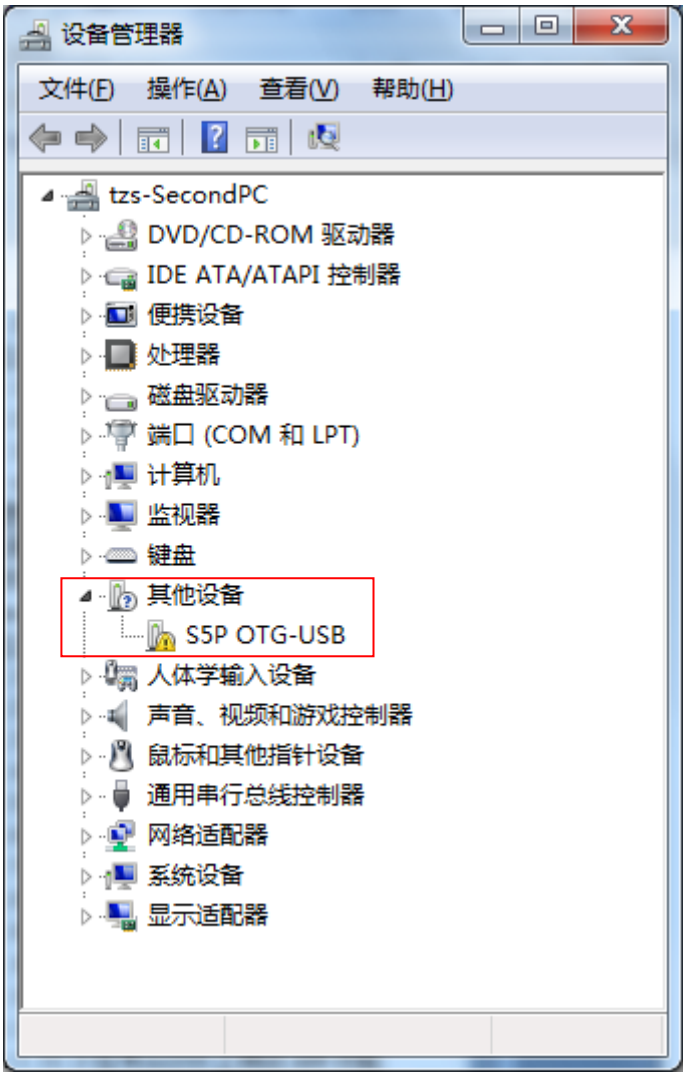
Run the SDK Manager as an administrator, select “Available Packages” on the

"Android SDK and AVD Manager" page, go to "Third party Add-ons" click on ">" to expand the list and check "Google Usb Driver package" as follows:

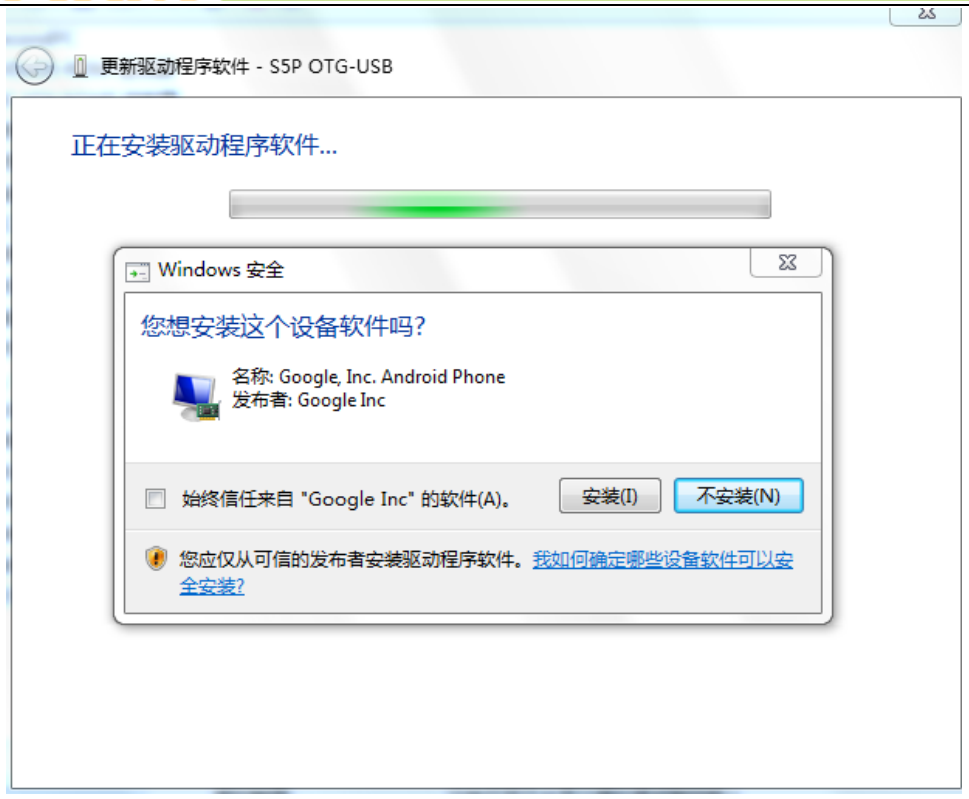


Click on the "Install Selected" button, check "Accept All" in the "Choose Packages to Install" dialog and click on "Install" to begin download. It may take a while.

After download is completed power on a 210 board and wait until Android fully loads. Connec the board to your PC via the shipped MiniUSB cable you will see Windows7 prompt that it is installing the driver. Later it will show "driver installation failed" and you can right click on "My Computer" and select "Property" you will see a S5P OTG-USB device:



Right click on “S5P OTG-USB”, select “Update Driver Software”, click on “Browse my computer for driver software” and “Browse”to select a USB driver. By default it is “C:\Program Files\Android\android-sdk\extras\google\usb_driver”. Click on “Next” to continue.



Click on the “Install” button on the above dialog and a moment later you will see the following information which indicates that your installation is completed.

6.1.11.2 Test ADB on 210

6.1.11.3 Add ADB Commands to Path Environment Variable

Please follow the steps below to add the ADB path to the Path variable

1) Right click on “My Computer” -> “Property”, on the left side click on the “Advanced System Settings” button

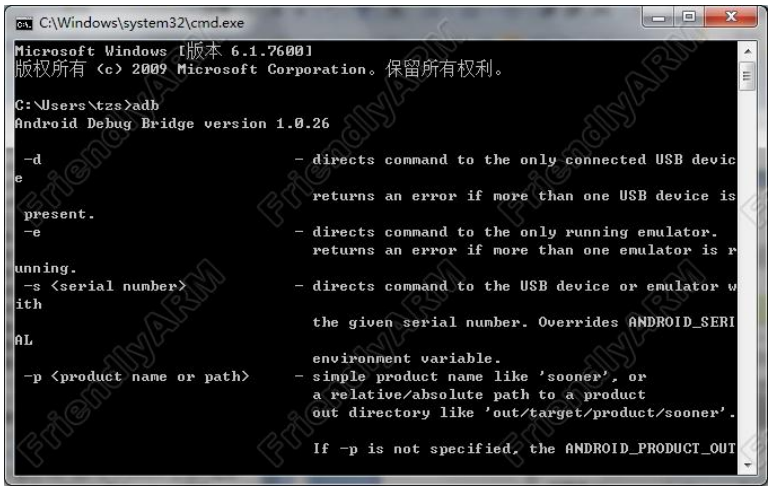
2) Click on the “System Variables” button

3) Go to “System Variables”, double click on the “Path” variable and add “C:\Program Files\Android\android-sdk\platform-tools;” at the very beginning. Note: there is a “;”.

4) Click on “OK” to save the changes

Verify ADB commands

Click on the Start menu, launch the commandline utility, type “adb” and return. If it is successfully installed you will be able to see the following information.



6.1.11.4 Test ADB

Check device status

Power on your 210 board and connect it to your PC via the mini USB cable. Please type the command below in your DOS commandline utility

```
# adb devices
```

If you can see the following information it means your device is successfully connected to your PC:

Enter the ADB Shell

Please run the command below to enter the terminal:

```
# adb shell
```

Type “exit” to return to DOS

Install Software with ADB

Let’s take “D:\sinaweibo_2.0.4.apk” as an example. Typing “**adb install D:\sinaweibo_2.0.4.apk**” in the DOS commandline will install the package.

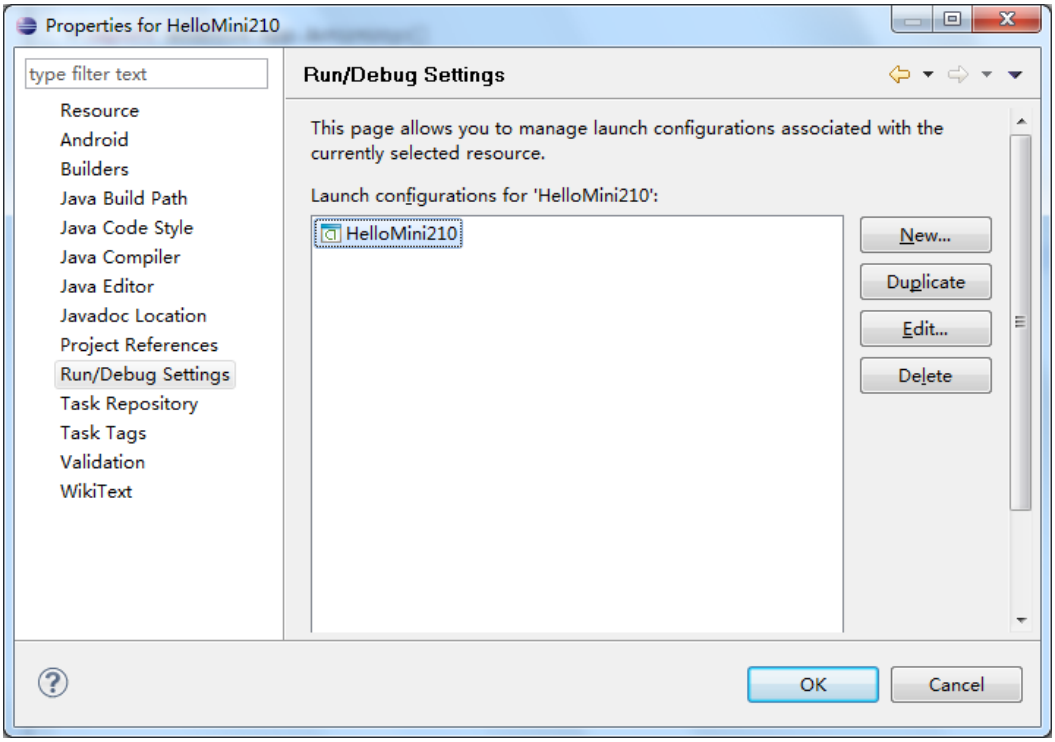
Other Functions

ADB is very powerful. Besides installation, debugging and Shell it can also transfer files to the 210 board.

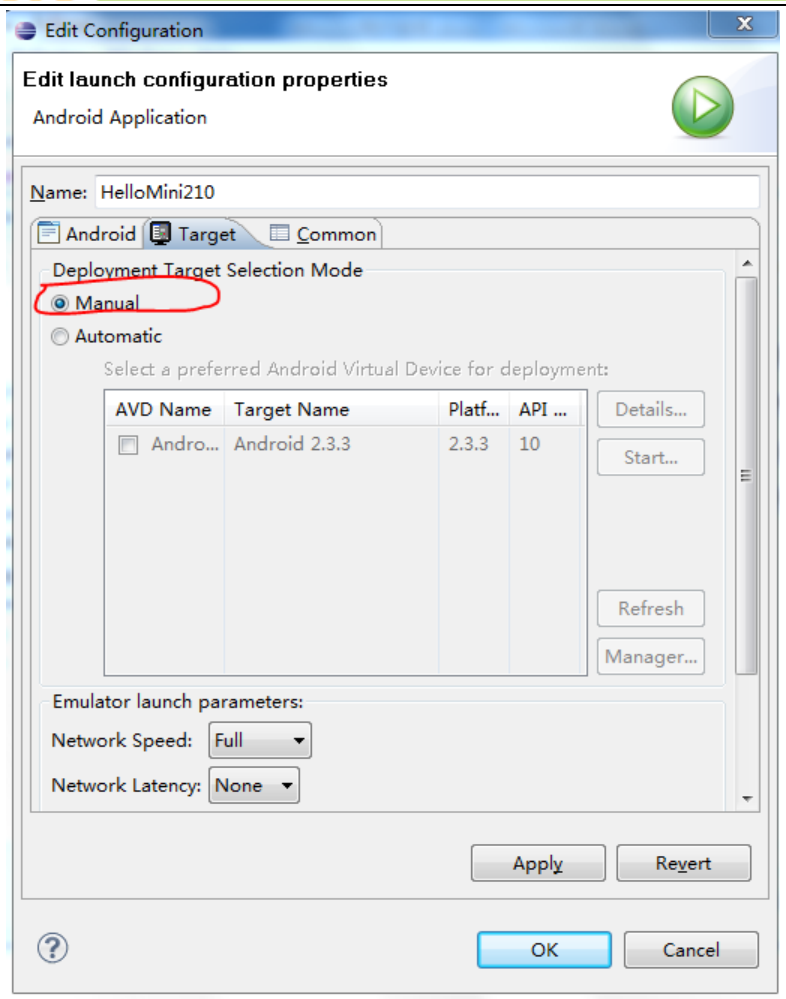
6.1.11.5 Run Programs on 210 Board with USB ADB

Start Eclipse and open the “HelloMini210” project.

On the left side click on the “HelloMini210” project in the Package Explorer, click on “Properties” and the “Properties for HelloMini210” dialog will pop up:

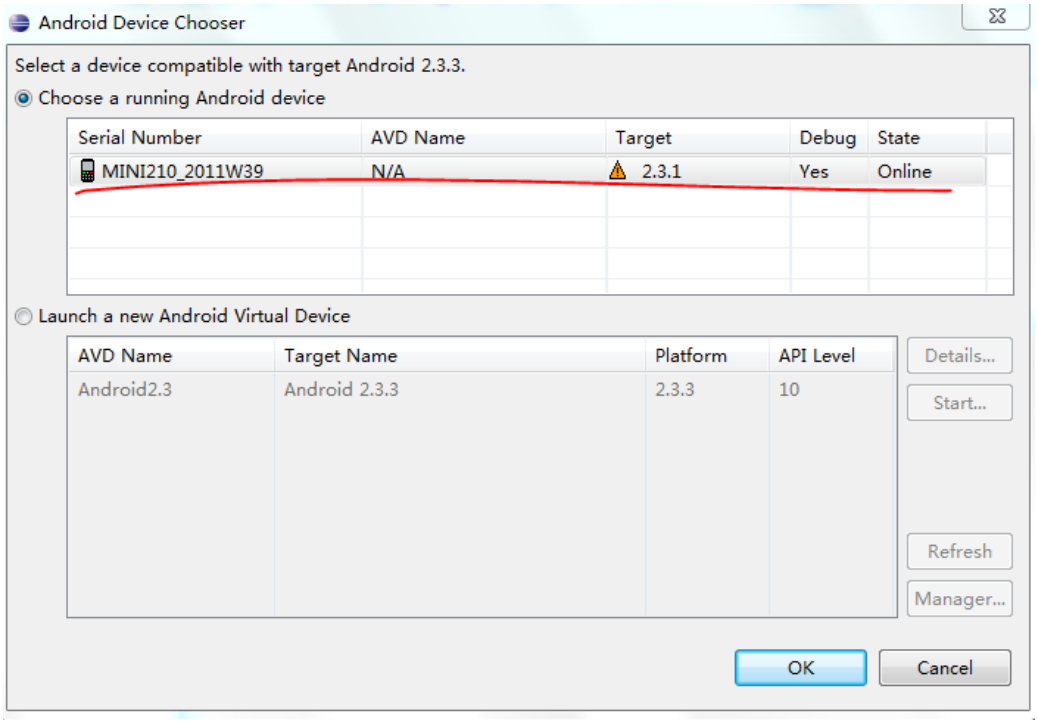


Click on “Run/Debug Settings”, select “HelloMini210” and click on “Edit…”. Click on “Target” on the Edit Configuration window and select “Manual” on the Deployment Target Selection Mode dialog

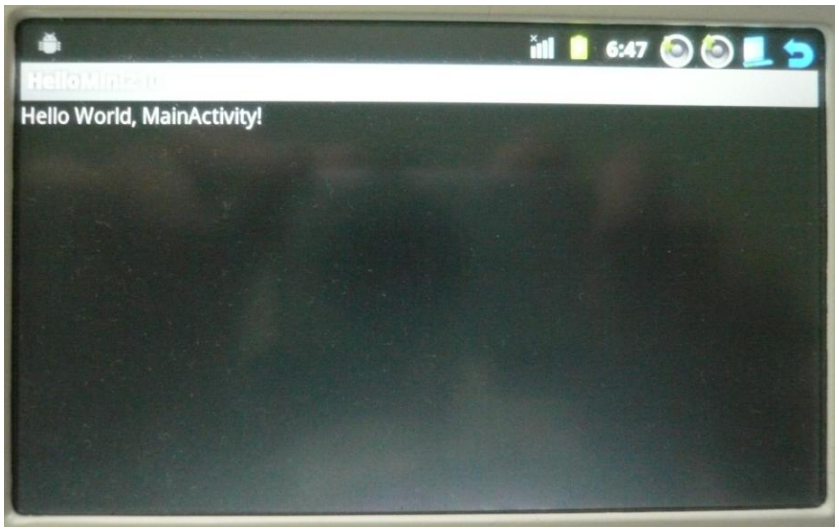


Click on “OK” to save

Now select “HelloMini210” and click on the “Run” button on the tools bar or go to “Run->Run As->Android Application”. On the “Android Device Chooser” dialog select “Choose a running Android device” and select the “2.3.1” target device (Tiny210), and click on “OK”



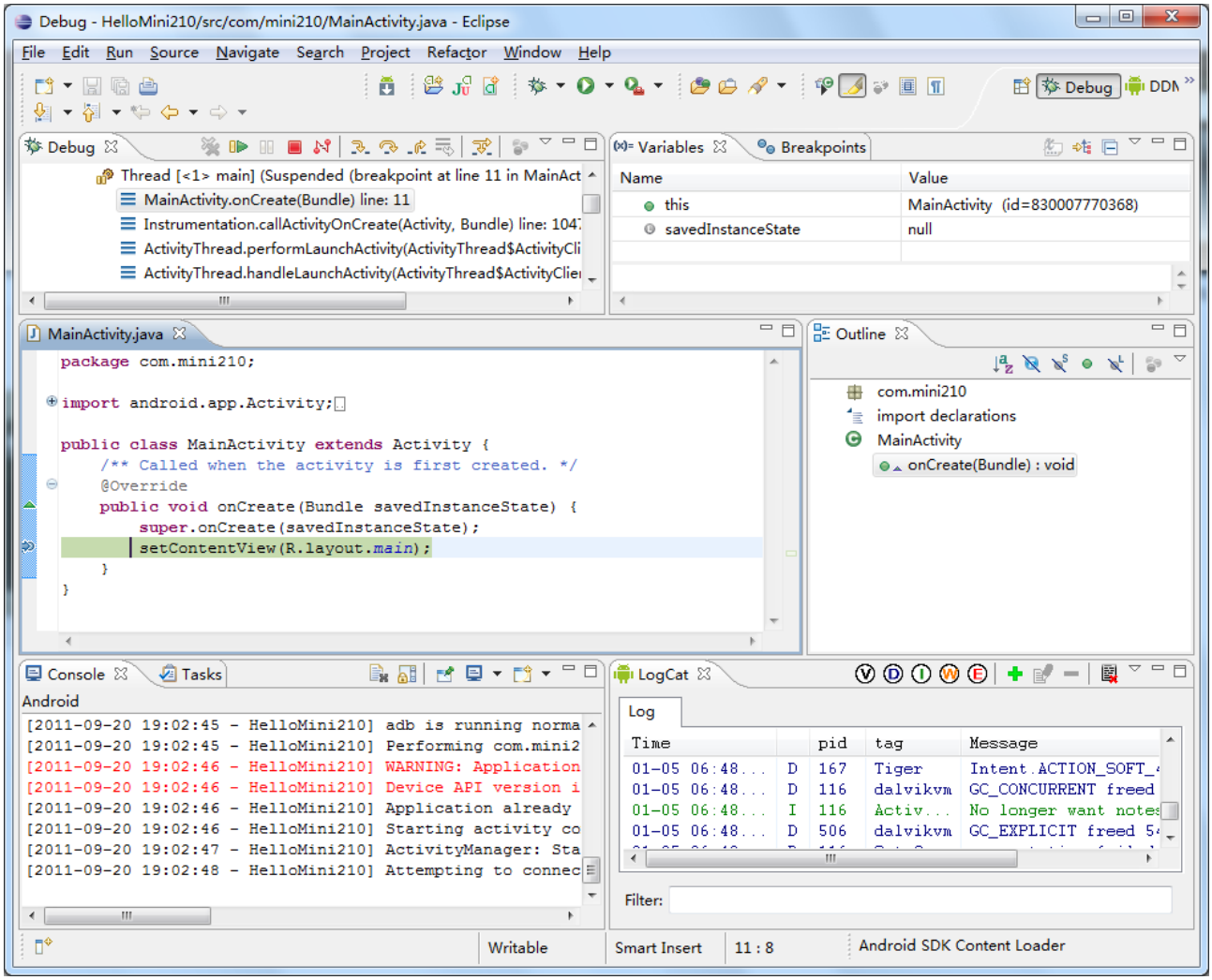
A moment later "HelloMini210" will be running on the Tiny210:



6.1.11.6 Debug Android Programs on 210

Exit the HelloMini210 if you are running it. Go to "Open Perspective->Debug" on the Eclipse main window you will enter the debug view. We can set a break point and

go to “Run->Debug” to run the program. The program will run to the break point and pause.



6.2 Access Hardware in Andorid

For users to fully use and acces the 210 hardware resources FriendlyARM developed a library named “libfriendlyarm-hardware.so” which can be used to access and operate the hardware resources on the 210 including serial port, buzzer, EEPROM and ADC.

The iTest utility is based on this library and you can run the iTest utility to learn this

library.

In this section we will describe how to use the libfriendlyarm-hardware.so library.

6.2.1 How to Use "libfriendlyarm-hardware.so"

FriendlyARM has included the libfriendlyarm-hardware.so library in Android since the March 2011 version. It is in the following directory:

```
device/Samsung/smdkv210/prebuilt/libfriendlyrm-hardware.so
```

On the 210 Android system it is in the "/system/lib/libfriendlyarm-hardware.so" directory.

If you develop Android applications with Eclipse you can follow the steps below to call libfriendlyarm-hardware.so APIs:

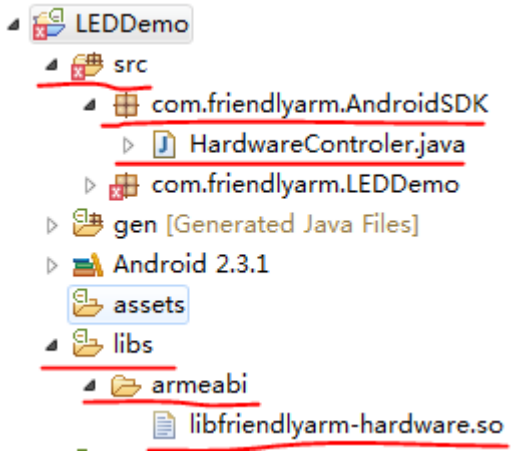
1) Go to your Android program's directory, create a "libs" directory, enter it and create an "armeabi" directory and copy "libfriendlyrm-hardware.so" to this directory.

2) Go back to your program's directory, enter the "src" directory and create a "com\friendlyarm\AndroidSDK" directory and create a "HardwareControler.java" file and type the following code:

```
package com.friendlyarm.AndroidSDK;
import android.util.Log;

public class HardwareControler
{
    /* Serial Port */
    static public native int openSerialPort( String devName, long baud, int dataBits,
```

```
int stopBits );

    /* LED */
    static public native int setLedState( int ledID, int ledState );

    /* PWM */
    static public native int PWMPlay(int frequency);
    static public native int PWMStop();

    /* ADC */
    static public native int readADC();

    /* I2C */
    static public native int openI2CDevice();
    static public native int writeByteDataToI2C(int fd, int pos, byte byteData);
    static public native int readByteDataFromI2C(int fd, int pos);

    /* 通用接口 */
    static public native int write(int fd, byte[] data);
    static public native int read(int fd, byte[] buf, int len);
    static public native int select(int fd, int sec, int usec);
    static public native void close(int fd);

    static {
        try {
        System.loadLibrary("friendlyarm-hardware");
        } catch (UnsatisfiedLinkError e) {
            Log.d("HardwareControler", "libfriendlyarm-hardware library not
found!");
        }
    }
}
```

Start Eclipse and select your project list and “Refresh” it now you will see the following information:



To use the HardwareControler APIs you need to add the following line to your code which introduces the HardwareControler class:

```
import com.friendlyarm.AndroidSDK.HardwareControler;
```

Now you will be able to call HardwareControler APIS and we will show you some examples in the following sections:

6.2.2 libfriendlyarm-hardware.so APIs

We will list some HardwareControler APIs here.

6.2.2.1 Serial Port API

Serial Port APIs:

APIs	Parameters and Return Value	Comment
int openSerialPort(String devName, **long** baud, **int** dataBits, **int** stopBits)	devName: device name, the following devices are available: /dev/s3c2410_serial1 /dev/s3c2410_serial2 /dev/s3c2410_serial3 /dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 baud: baud rate dataBits: in general we use 8 stopBits: in general we use 1	Open a serial device and return its file descriptor.

	Return Value: When the device is opened successfully it will return a file descriptor which can be used to read、write and select the device otherwise it will returns -1.	
int write(**int** fd, **byte**[] data)	fd: file descriptor data: data to write to the device Return Value: When the operation succeeds it will return the number of characters written otherwise it will return -1.	Write data to an opened device.
int read(**int** fd, **byte**[] buf, **int** len)	fd: file descriptor buf: data buffer len: number of characters to read Return Value: When the operation succeeds it will return the number of characters read otherwise it will return -1. If before the read function is called the file pointer already reaches the end of the device it will return 0	Read data from an opened device.
int select(**int** fd, **int** sec, **int** usec)	fd: file descriptor sec: seconds allowed to wait usec: useconds allowed to wait(1ms = 1000us) Return Value: If the device has data it will return 1 otherwise it will return 0. If this operation fails it will return -1.	Query whether an opened device has data for reading.
void close(**int** fd)	fd: file descriptor Return Value: No	Close a device

Notes:

First you need to open a serial device with “openSerialPort”, then your can call “select” to query if is has available data. When it has data you can call “read” to read data.

To write data to a device you can call “write”. If you don’t need to use a device remember to “close” it.

6.2.2.2 LED APIs

LED APIs:

APIs	Parameters and Return Value	Comment
int setLedState(**int** ledID, **int** ledState)	ledID： LED you want to access (0~3) ledState: 1 is on， 0 is off Return Value: If this operation succeeds it will return 0 otherwise it will return -1	Open an LED

6.2.2.3 PWM APIs

PWM Buzzer APIs:

APIs	Parameters and Return Value	Comment
int PWMPlay(**int** frequency);	frequency: frequency of sound Return Value: If this operation succeeds it will return 0 otherwise it will return -1	Play a Buzzer with the specified frequency
int PWMStop();	Return Value: If this operation succeeds it will return 0 otherwise it will return -1	Stop a buzzer

6.2.2.4 ADC APIs

ADC APIs:

APIs	Parameters and Return Value	Comment
int readADC()	Return Value: If this operation succeeds it will return the conversion result otherwise it will return -1	Read an ADC conversion result

6.2.2.5 EEPROM APIs

EEPROM APIs:

APIs	Parameters and Return Value	Comment
int openI2CDevice();	Return Value: If this operation succeeds it will return a file descriptor otherwise it will return -1.	Open an IIC device and return a file descriptor. After an IIC device is opened successfully you can call “writeByteDataToI2C” and “readByteDataFromI2C” to operator the EEPROM.
int writeByteDataToI2C(	fd: file descriptor	Write data to EEPROM (one byte on each

int fd, **int** pos, **byte** byteData);	pos: position where data to be written (0~255) byteData: data to be written Return Value: If this operation succeeds it will return the number of characters written otherwise it will return -1.	write operaration). Note: this operation is time consuming. It costs about 10 ms.
int readByteDataFromI2C(**int** fd, **int** pos);	fd: file descriptor pos: position where data to be read (0~255) Return Value: If this operation succeeds it will return data it reads otherwise it will return -1. If before this function is called the file pointer has reached the end of the device it will return 0. The return value's type is int you need to convert it to a byte value.	Read data from EEPROM. Note: this operation is time consuming. It costs about 10 ms.
void close(**int** fd)	fd: file descriptor Return Value:No	Close a device

Notes:

First you need to open a serial device with "openI2CDevice", then your can call "writeByteDataToI2C" to write data and "readByteDataFromI2C" to read data. These operations are time consuming and will usually cost 10ms therefore it is better to call them in a new thread.

The EEPROM device can store 256 bytes data so the position parameter's value ranges from 0 to 255 and each time it can only read/write one byte

If you don't need to operate the device you need to "close" it.

6.2.3 Code Samples

In the "Android" directory in the shipped CO there is a LED Demo program you can open it in Eclipse and learn how to use the libfriendlyarm-hardware.so library.

You can debug, download and run it on the board via the shipped miniUSB cable.

7 Linux Installation and Navigation

7.1 Linux GUIs

The Linux image we prepare for the 210 includes Qtopia2.2.0, QtE4.7 and Qt Extended 4.4.3. Users can switch among all these three GUIs freely. By default the Linux GUI is Qtopia 2.2.0.

To get the latest QtE, please go to http://qt.nokia.com/.

7.2 Install and Play with Linux

7.2.1 Install Linux

Note: before read the following sections please burn Superboot to your SD card and copy corresponding installation files to your card.

Step1: insert the SD card to a PC, open the "images\FriendlyARM.ini" file and modify it as follows:

```
#This line cannot be removed. by FriendlyARM(www.arm9.net)

CheckOneButton=No
Action=Install
OS=Linux
VerifyNandWrite=No
lowformat=No
LCD-Mode = No
LCD-Type = S70
Check CRC32=No
Status Type = Beeper| LED

################### Linux ####################
Linux-BootLoader = superboot210.bin
Linux-Kernel = Linux/zImage
```

```
Linux-CommandLine = root=/dev/mtdblock4 console=ttySAC0,115200 init=/linuxrc
Linux-RootFs-InstallImage = Linux/rootfs_qtopia_qt4.img
```

(Note: the image currently supports these LCDs: H43, W50, A56, S70, A70, L80 and G10)

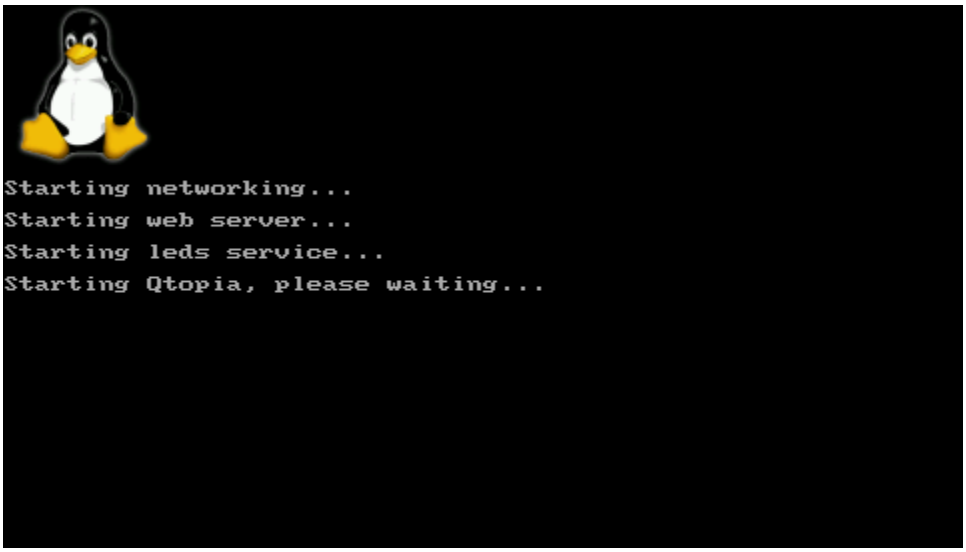
Step2: make sure your card has the following files (actually you can copy the whole image directory to your SD card's root directory)

File	Comment
images\superboot210.bin	Bootloader. It can boot Linux and other OS such as Android and WinCE. It can be run from an SD card.
images\Linux\zImage	Linux kernel. It can automatically detect LCD types
images\Linux\rootfs_qtopia_qt4.img	Linux file sysem image
images\FriendlyARM.ini	Configuration file

Step3: insert the SD card to the board's **SD** socket and switch S2 to the SD side. Power on the board and you will hear a beep and see a progress bar on the LCD.

Step4: after system burning is done you will hear two continuous beepings and the LCD will show the burning status. Switch S2 to the Nand Flash side, reboot the system and you will see Android loads.

If you are running Linux for the first time you will see the following calibration screen:



7.2.2 Calibrate Touch Screen

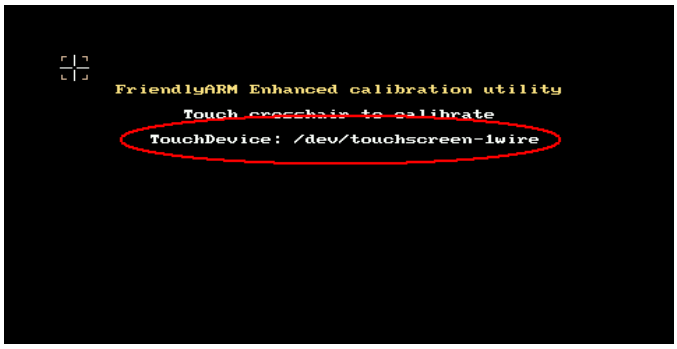
Note: if you didn't calibrate well you can delete the "etc/pointercal" and reboot the system, or reflash the board or use a USB mouse to calibrate after Linux is loaded.

In the following two scenarios the calibration screen will be brought up:

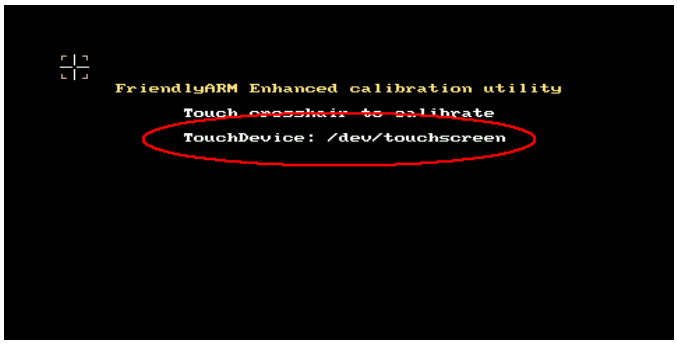
1. You reflash your board and restart your board for the first time

Click on "+", follow it till the end position and Android will resume. After it is

completely booted you will see the following screen:



The following screen shows the system uses an ARM LCD: /dev/touchsreen (marked in red)



2. After enter the system go to “start->setting” and click on the “calibrate” icon you will see the above screen too.

7.2.3 Introduction to Main Pages

After Linux qtopia is loaded you will see the following screen.

There are five pages which represent five categories of software and documents. Click on the “start” on the left bottom you will see five sub-menus which are the same as these five pages. The “FriendlyARM” page contains software utilities that are all developed or migrated by us. All the other utilities and documents in other pages are open source

7.2.4 SMPlayer

Mplayer is an open source player which supports output to various devices such as X11, Framebuffer and so on. Here we use its output to framebuffer and integrate the SMPlayer.

7.2.5 HDMI Output

By default Linux opens the HDMI output to TV. You need to turn on your TV fist and then connect it to your board via the HDMI cable.

You can even directly connect to a TV without using the LCD we usually provide. In this case since you don't need to do screen calibration you can add "skipcali=yes" in the Linux-CommandLine to skip the calibration step.

If you want to close the HDMI output or reset the HDMI resolution you can go to the "FriendlyARM" page and click on the HDMI icon. Uncheck "Auto start HDMI-output on boot" and check the resolution you want from the list and save it.

You can configure the HDMI setting in the "/root/Settings/HDMI.conf" file.

```
[HDMISetting]
AutoStart = yes
Resolution = 720P-60Hz
SupportResolution1=720P-60Hz
SupportResolution2=1080I-50Hz
```

"AutoStart" defines whether the HDMI output will be started on system boot. "Resolution" defines the HDMI resolution and the following values are available:

1080P-60Hz, 1080P-50Hz, 1080I-60Hz, 1080I-50Hz, 720P-60Hz, 720-50Hz, 576P-50Hz-16:9, 576P-50Hz-4:3, 480P-60Hz-16:9 and 480P-60Hz-4:3

HDMI's backend service is "hdmi-service" which can be started by calling "hdmi-service autostart" in the "/etc/init.d/rcS".

7.2.6 HDMI Output Without Connecting LCD

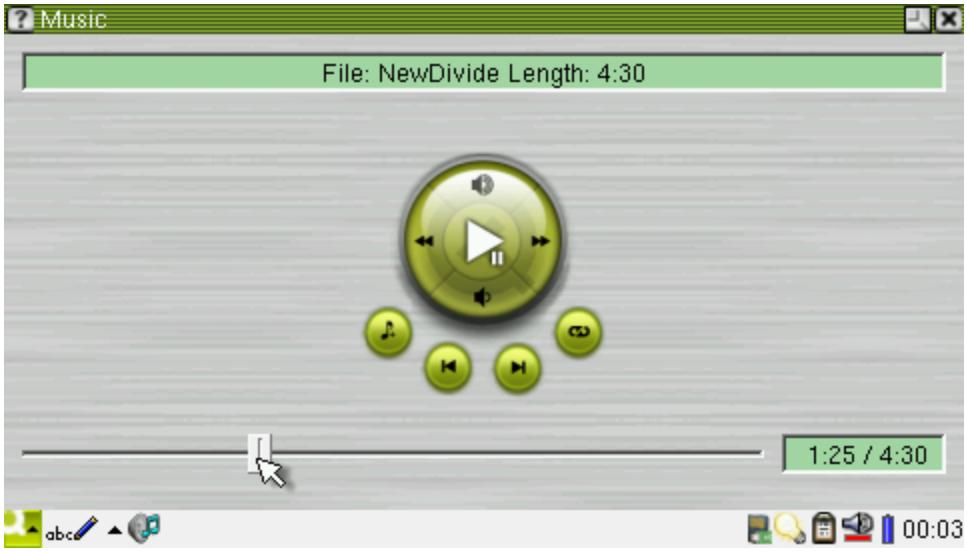
The 210 boards can output to HDMI monitors without connecting an LCD.

If you want to connect your 210 board only to an HDMI monitor without connecting an LCD you need to specify "LCD-Type" e.g. "LCD-Type=HDMI720P60". You need to reflash your board with this new definition without connecting an LCD.

After an OS is installed please power on the board without connecting an LCD you will be able to see HDMI output.

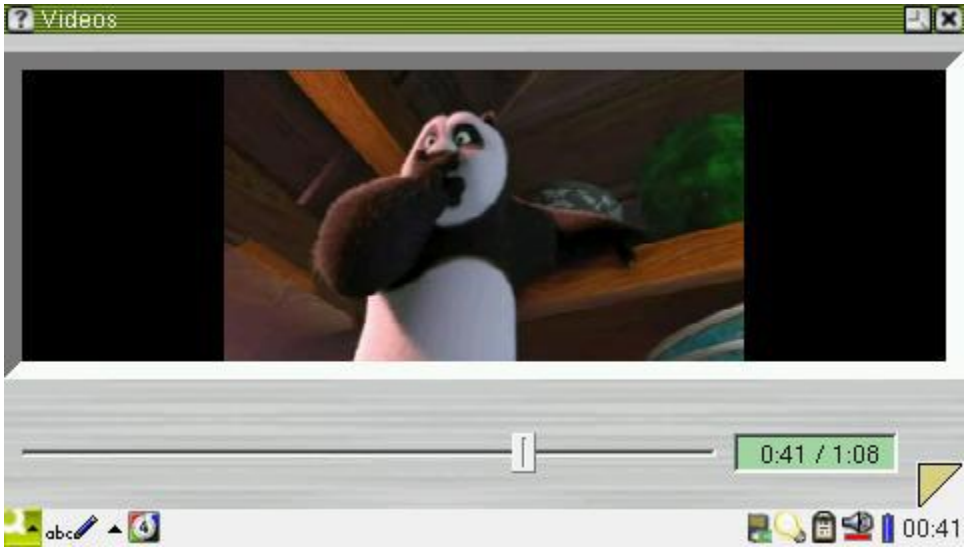
7.2.7 Play MP3

Go to the "Application" page, click on the "music" icon, select an mp3 and click on "play".



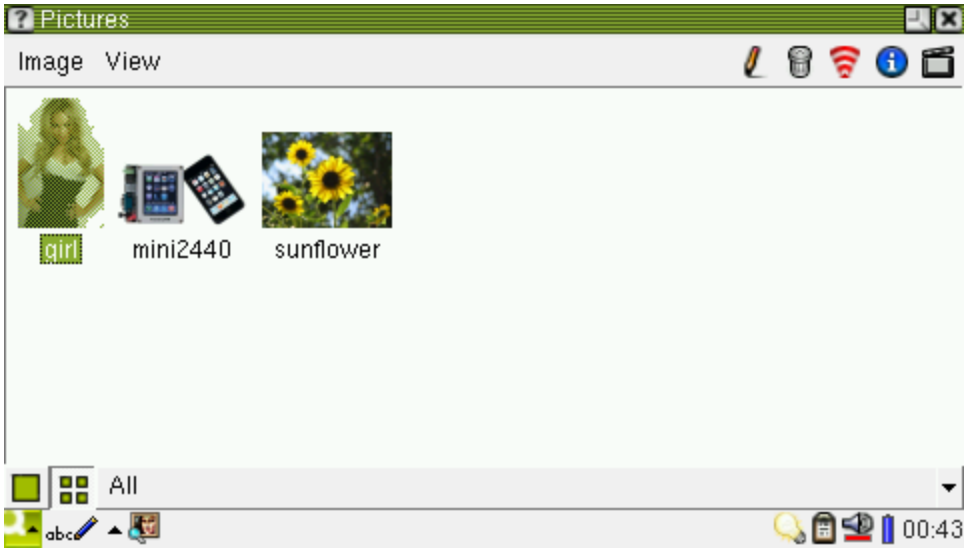
7.2.8 Play Video

Go to the “Application” page, click on the “video” icon, select a video file and click on “play”. This player can fluently play H.264/H.263/Mpeg4 files.



7.2.9 Image Viewer

Go to the Application page, click on the “pictures” icon and you will be able to browse pictures



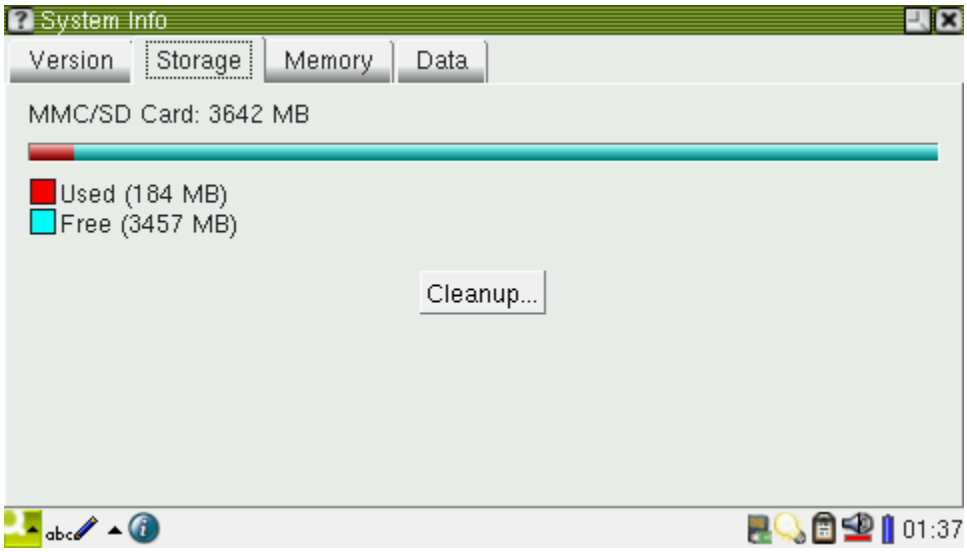
7.2.10 Auto Mount of SD Card

After system is loaded if you plug an SD card or a USB flash drive you will see a mobile storage device icon appear on the right bottom.

All files in the MMC/SD card will be listed in the “Documents” page.

Note: this auto mount function is developed by FriendlyARM and currently it can only recognize the card’s first section and formats of VFAT/FAT32/FAT16.

Click on the “Applications” -> “Storage” you will see the card’s data



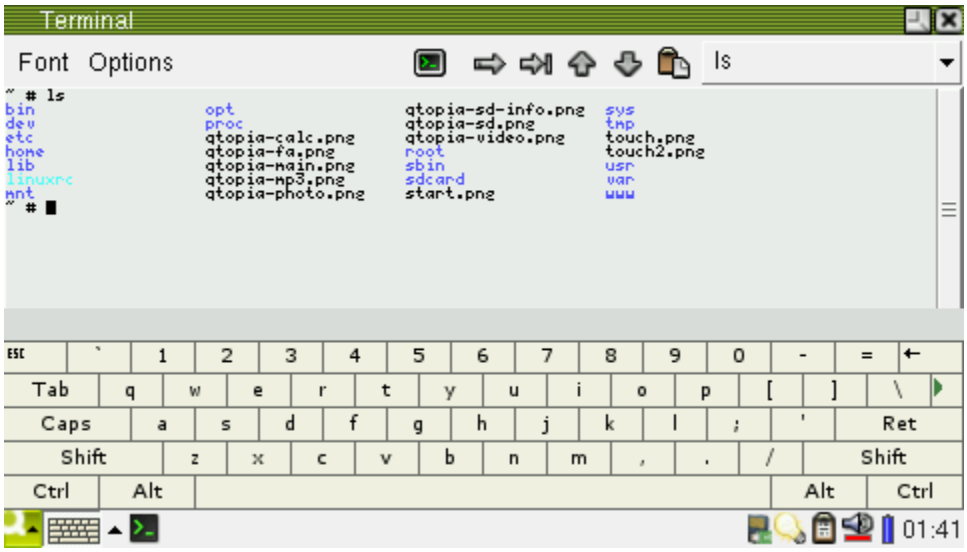
7.2.11 Calculator

Go to “Applications” and click on the calculator icon. You can select “Simple”, “Fraction”, “Scientific” and “Conversion”.

7.2.12 Terminal

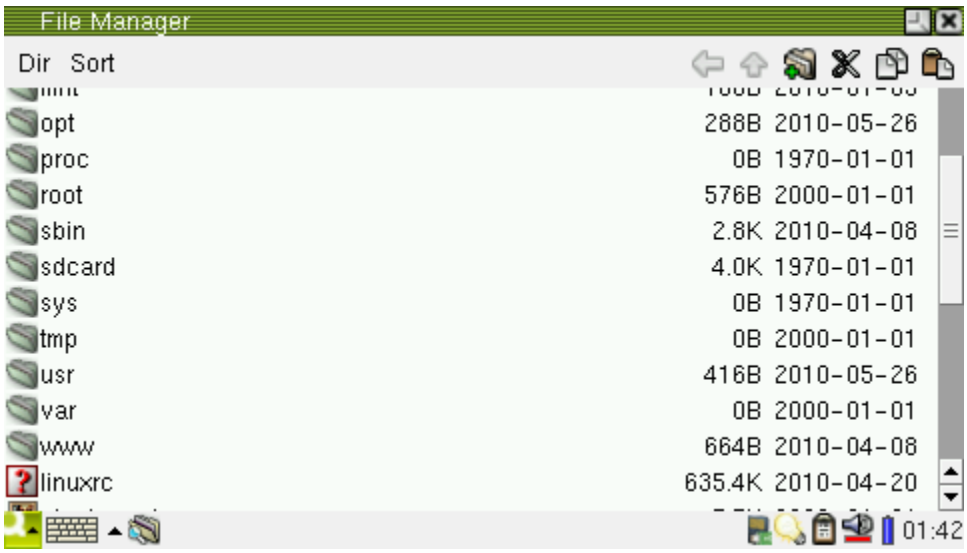
Go to “Applications”, click on the terminal icon and you will be able to type Linux

commands.



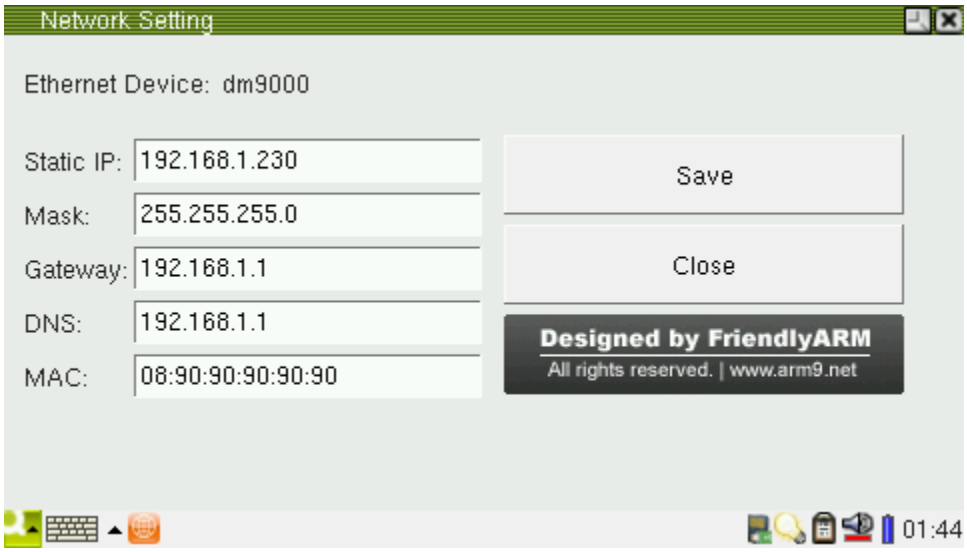
7.2.13 File Manager

Go to “FriendlyARM”, click on the file manager icon and you will see your system’s file structure:



7.2.14 Ethernet Setting

Go to “FriendlyARM”, click on the network setting and you will be able to see the following screenshot



You can set your network parameters and “save” it to the “/etc/eth0-setting”.

7.2.15 Wireless Network

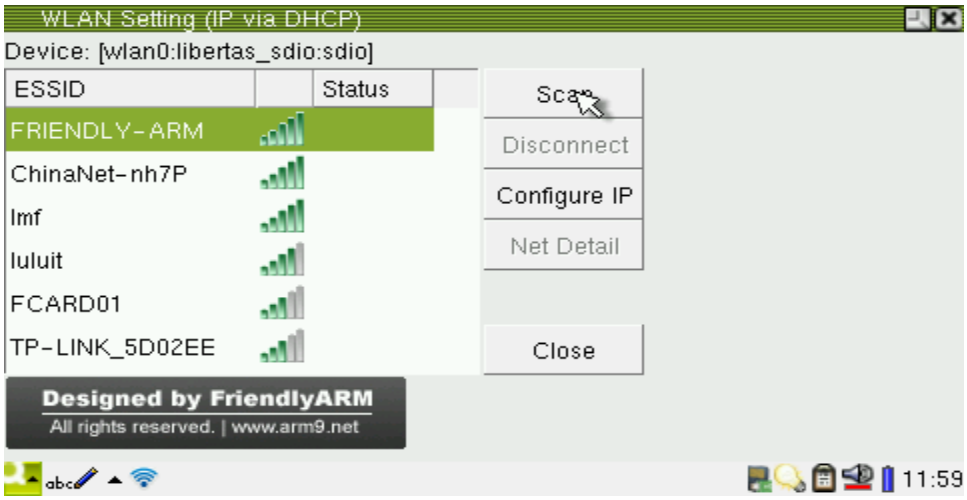
This section will introduce how to configure the SD WiFi and USB WiFi.

7.2.15.1 Wireless Utility

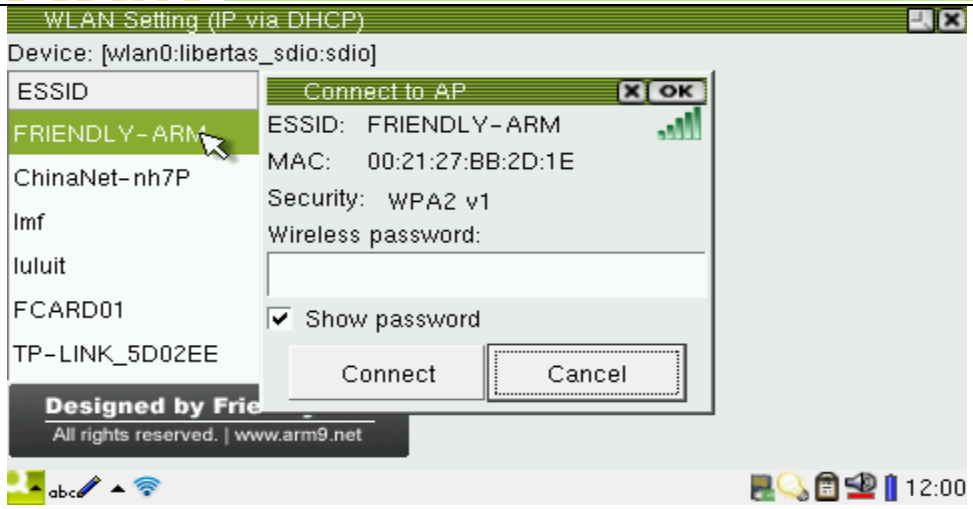
Go to the “FriendlyARM” page, click on the wireless setting icon

7.2.15.2 Wireless AP

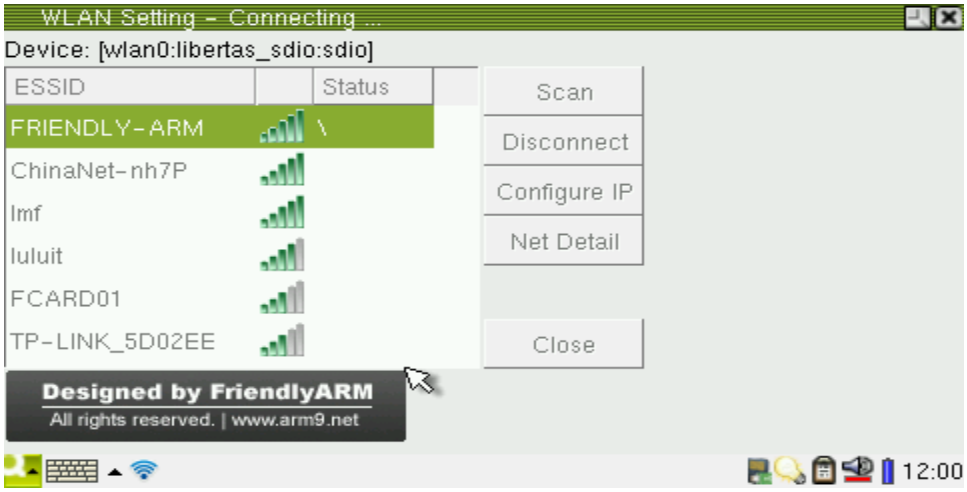
After launching the setting utility it will automatically search for an AP and list all SSIDs and their signal strengths.



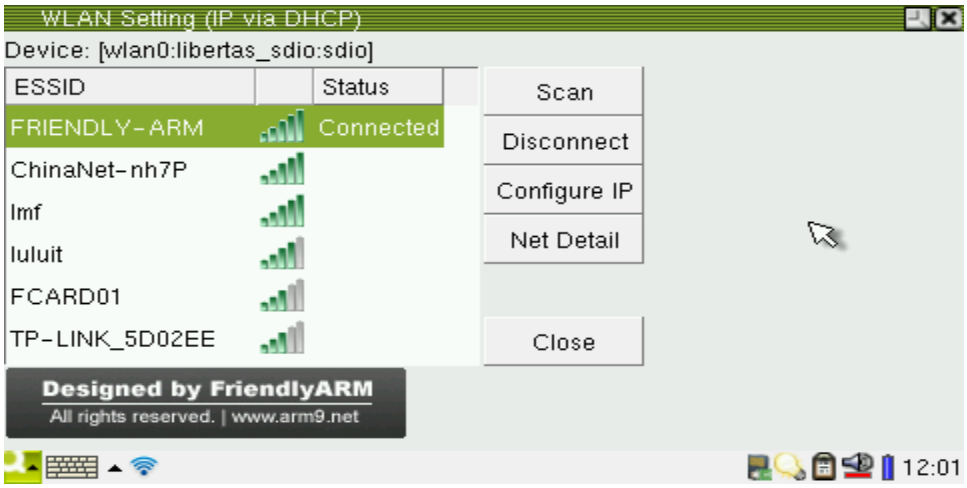
After an AP is found to connect to it you can click on its ESSID and input its password



Click on “connect”

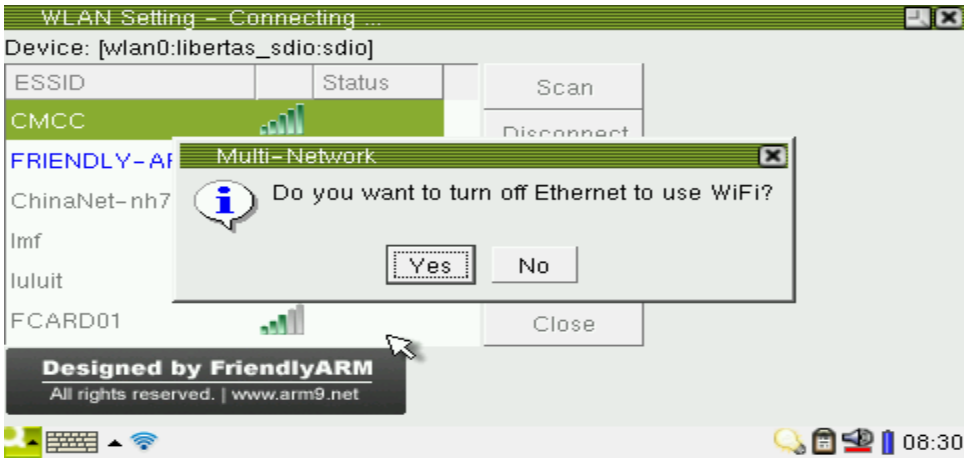


If the connection is successful it will show “Connected”

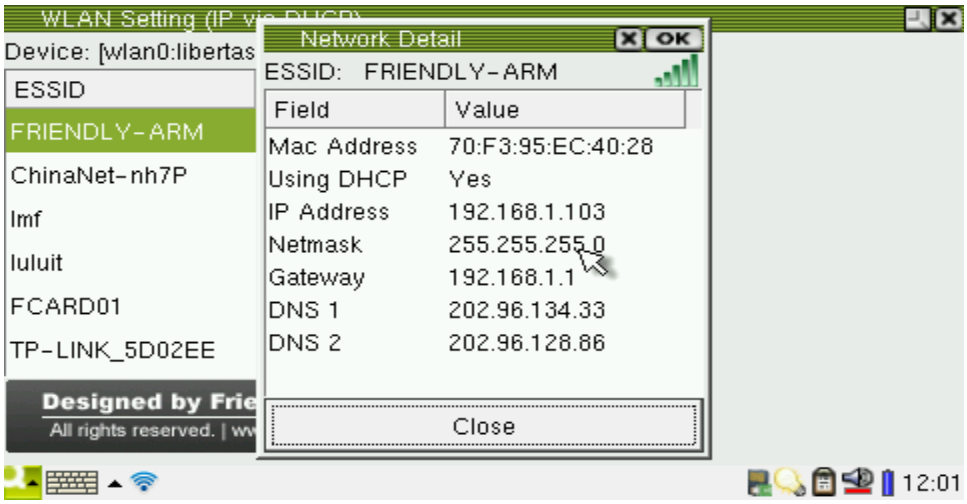


If you started the Ethernet before you start the wireless you will see the following dialog

which prompts you to close the Ethernet. You need to close the Ethernet.



Click on “Net Detail” you will see the wireless network’s details

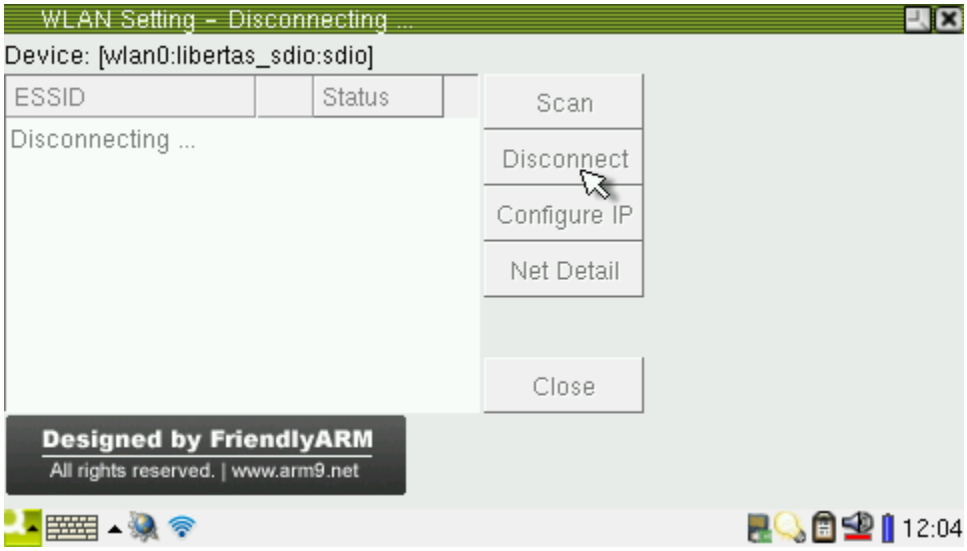


After your connection is successful, click on “close” to minimize the utility

Now you can surf the internet

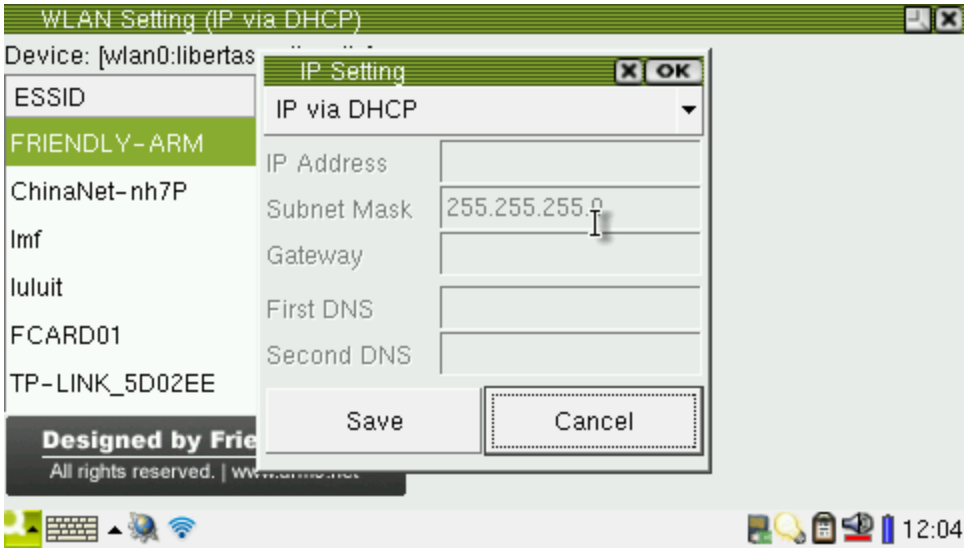
7.2.15.3 Disconnect Wireless Network

To disconnect the wireless network you can just click on “Disconnect”

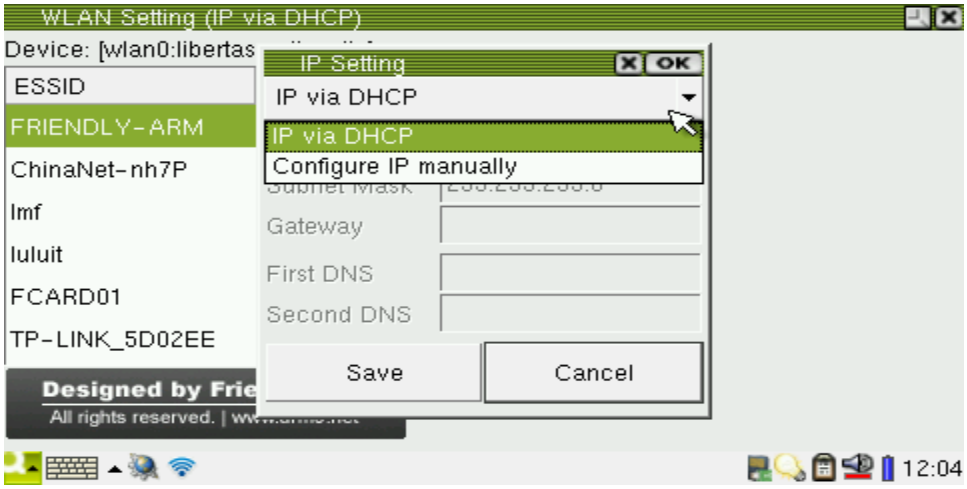


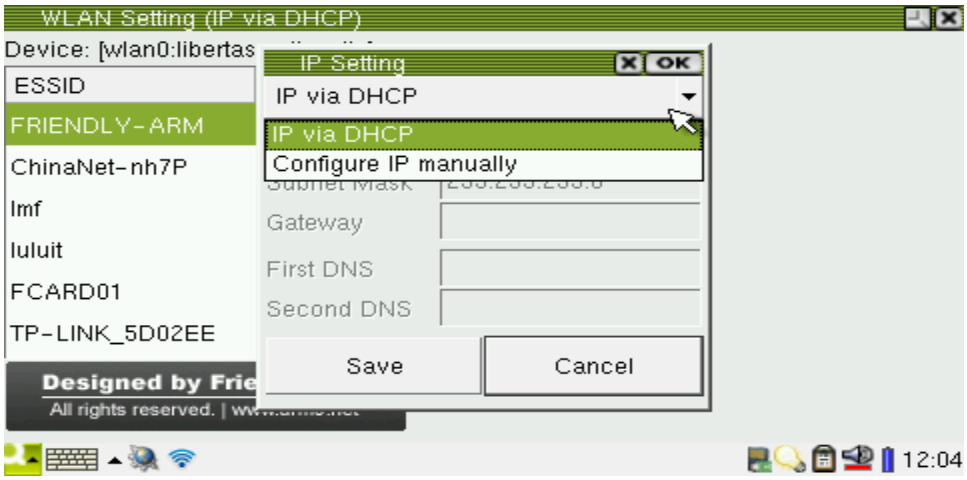
7.2.15.4 IP Configuration

On the wireless utility window click on “Configure IP” you will see the following dialog:



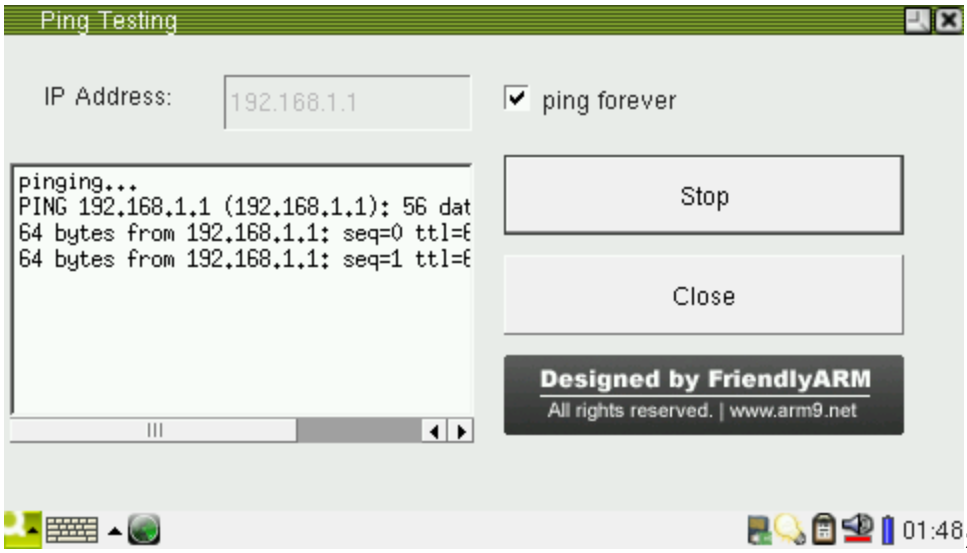
Click on the "IP Setting" pull-down list you will be able to select "DHCP" or "Configure IP Manually"





7.2.16 Ping Test

After configure your network please go to “FriendlyARM” and click on “Ping Testing”



7.2.17 Web Browser

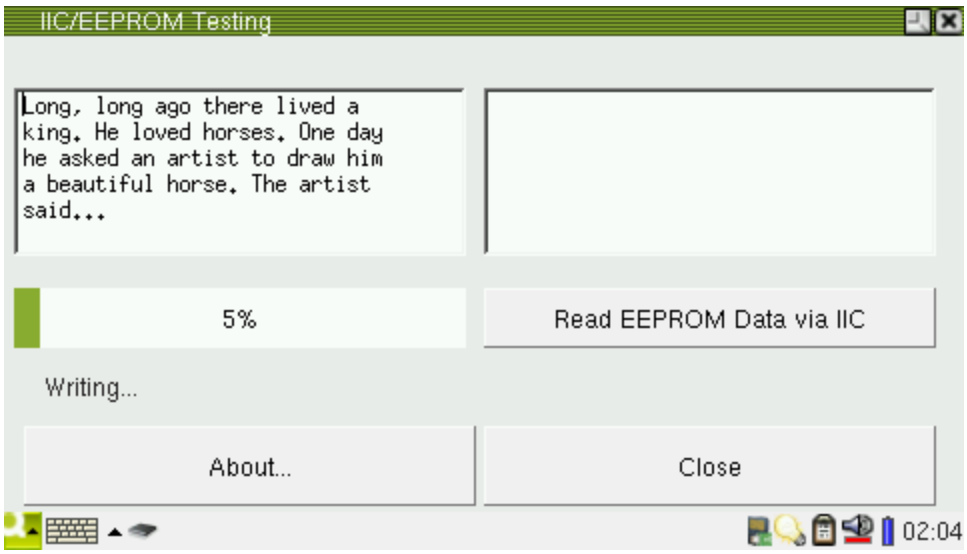
Go to “FriendlyARM”, click on “Browser”, open the soft keypad on the left bottom and you can type a website in the address bar.

7.2.18 LED Test

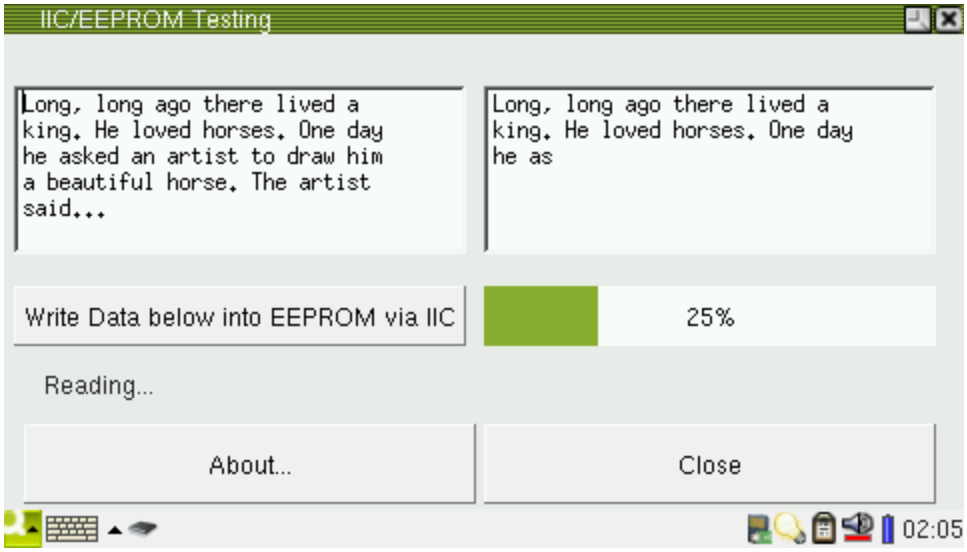
Go to “FriendlyARM” and click on “LED Testing”

7.2.19 EEPROM Reading and Writing

Go to “FriendlyARM” and click on “I2C-EEPROM”. Click on “Write Data below into EEPROM via IIC” you will see the writing process



Click on “Read EEPROM Data via IIC” you will see the reading process



8 Linux Application Development

We have another document which has very detailed information about how to do Linux development applications.

9 WindowsCE6 Installation and Navigation

9.1 Installaing and Playing with WindowsCE6

9.1.1 Installing WindowsCE6

Note: before read the following sections please burn Superboot to your SD card and copy corresponding installation files to your card.

Step1: insert the SD card to a PC, open the "images\FriendlyARM.ini" file and modify it as follows:

```
#This line cannot be removed. by FriendlyARM(www.arm9.net)

CheckOneButton=No
Action=Install
OS=CE6
```

```
VerifyNandWrite=No
low-format=No
LCD-Mode = No
LCD-Type = S70
Check CRC32=No
Status Type = Beeper| LED

#################### Linux ####################
WindowsCE6-BootLoader = superboot210.bin
WindowsCE6-BootLogo = WindowsCE6\bootlogo.bmp
WindowsCE6-InstallImage = WindowsCE6\NK.bin
WindowsCE6-RunImage = WindowsCE6\NK.bin
```

(Note: the image currently supports these LCDs: H43, W50, A56, S70, A70, L80 and G10)

Step2: make sure your card has the following files (actually you can copy the whole image directory to your SD card's root directory)

File	Comment
images\superboot210.bin	Bootloader. It can boot CE and other OS such as Android and Linux. It can be run from an SD card.
images\CE\bootlogo.bmp	CE start logo. You can replace it with your own. It should be a 24bit bmp.
images\CE\NK.bin	CE kernel and file sysem image
images\FriendlyARM.ini	Configuration file

Step3: insert the SD card to the board's **SD** socket and switch S2 to the SD side. Power on the board and you will hear a beep and see a progress bar on the LCD.

Step4: after system burning is done you will hear two continuous beepings and the LCD will show the burning status. Switch S2 to the Nand Flash side, reboot the system and you will see Android loads.

9.1.2 Running WindowsCE6 from SD Card

Note: before read the following sections please burn Superboot to your SD card and copy corresponding installation files to your card.

Step1: insert the SD card to a PC, open the "images\FriendlyARM.ini" file and modify it as follows:

```
#This line cannot be removed. by FriendlyARM(www.arm9.net)

CheckOneButton=No
Action=Run
OS=CE6
VerifyNandWrite=No
low-format=No
LCD-Mode = No
LCD-Type = S70
Check CRC32=No
Status Type = Beeper| LED

################### Linux ####################
WindowsCE6-BootLoader = superboot210.bin
WindowsCE6-BootLogo = WindowsCE6\bootlogo.bmp
```

```
WindowsCE6-InstallImage = WindowsCE6\NK.bin
WindowsCE6-RunImage = WindowsCE6\NK.bin
```

(Note: the image currently supports these LCDs: H43, W50, A56, S70, A70, L80 and G10)

Step2: make sure your card has the following files (actually you can copy the whole image directory to your SD card's root directory)

File	Comment
images\superboot210.bin	Bootloader. It can boot CE and other OS such as Android and Linux. It can be run from an SD card.
images\CE\bootlogo.bmp	CE start logo. You can replace it with your own. It should be a 24bit bmp.
images\CE\NK.bin	CE kernel and file sysem image
images\FriendlyARM.ini	Configuration file

Step3: insert the SD card to the board's **SD** socket and switch S2 to the SD side. Power on the board and you will hear a beep and see WinCE is being loaded.

10 WindowsCE6 Application Development

10.1 Set up Development Environment for WindowsCE Applications

Note: the 6410's WinCE development environment cannot co-exist with the 210's. If

your PC is installed with the Tiny6410's development environment you need to delete the following directory before you install the Tiny210 BSP

C:\WINCE600\PLATFORM\COMMON\SRC\SOC\S3C6410_SEC_V1

If you do want to install both the Tiny6410 and Tiny210's BSPs please try this way on VMWare

Note: the following software installation steps are based on MS Windows 7(Flagship). We haven't tested them on other systems. We suggest users to copy software to hard disk to install

Here are our PC host's configuration for reference:

CPU: Intel Core Duo E8400

RAM: DDR2 4GB

Hard Disk: 500GB

Here is our software list: (our company does'n provide Windows Embedded 6.0 CE 6

installation file, users need to go to MS' home site to download its trial version)

Visual Studio 2005 (trial version:

http://download.microsoft.com/download/e/1/4/e1405d9e-47e3-404c-8b09-489437b27fb0/En_vs_2005_Pro_90_Trial.img)

Visual Studio 2005 Service Pack 1(VS80sp1-KB926601-X86-ENU.exe)

http://www.microsoft.com/downloads/details.aspx?familyid=bb4a75ab-e2d4-4c96-b39d-37baf6b5b1dc&displaylang=en

Visual Studio 2005 Service Pack 1 Update for Windows Vista

(VS80sp1-KB932232-X86-ENU.exe)

http://www.microsoft.com/downloads/details.aspx?FamilyID=90E2942D-3AD1-4873-A2EE-4ACC0AACE5B6&displaylang=en)

Visual Studio 2005 Service Pack 1 ATL Security Update

(VS80sp1-KB971090-X86-INTL.exe)

http://www.microsoft.com/downloads/details.aspx?familyid=7C8729DC-06A2-4538-A90D-FF9464DC0197&displaylang=en

Windows Embedded CE 6.0

http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=7e286847-6e06-4a0c-8cac-ca7d4c09cb56

Windows Embedded CE 6.0 Platform Builder Service Pack 1

http://www.microsoft.com/downloads/details.aspx?FamilyId=BF0DC0E3-8575-4860-A8E3-290ADF242678&displaylang=en

Windows Embedded CE 6.0 R2

http://www.microsoft.com/downloads/details.aspx?FamilyId=F41FC7C1-F0F4-4FD6-9366-B61E0AB59565&displaylang=en

Windows Embedded CE 6.0 R3

http://www.microsoft.com/downloads/details.aspx?FamilyID=BC247D88-DDB6-4D4A-A595-8EEE3556FE46&displaylang=ja&displaylang=en

Tencent QQ (Third Party Software)

http://www.microsoft.com/downloads/details.aspx?FamilyID=527042f7-bb5b-4831-a6ad-5081808824ec&displaylang=en

WesttekFileViewers6.exe

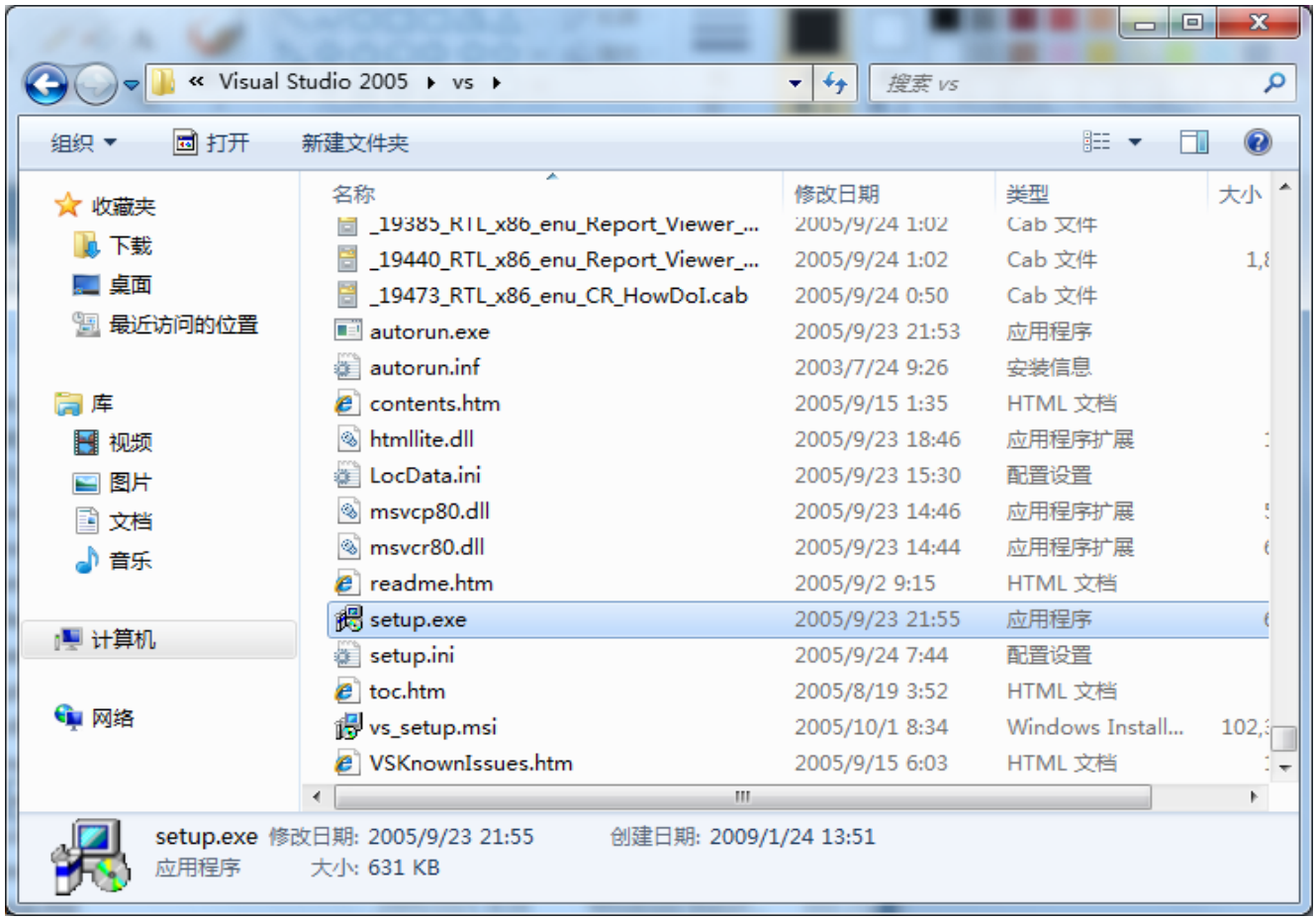
http://www.microsoft.com/downloads/details.aspx?FamilyID=d2fd14eb-7d5c-428b-951c-343f910047c1&displaylang=en

Please follow the order in the list to install these software components.

Note: WinCE6.0's Platform Builder is a plug-in of VS2005 therefore users need to install VS2005 first.

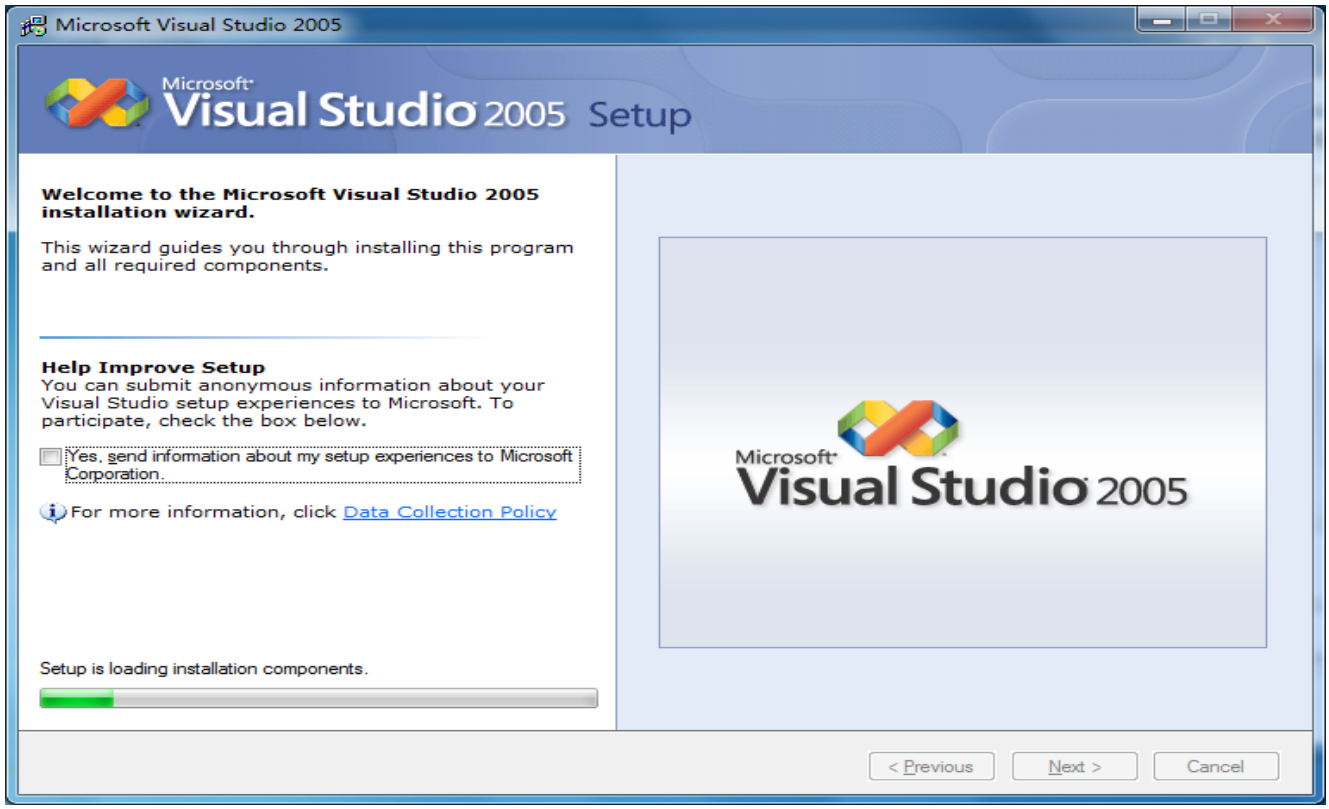
10.1.1 Install Visual Studio 2005 and Patches

Step 1: Open Visual Studio 2005 and find setup.exe, double click on it and install

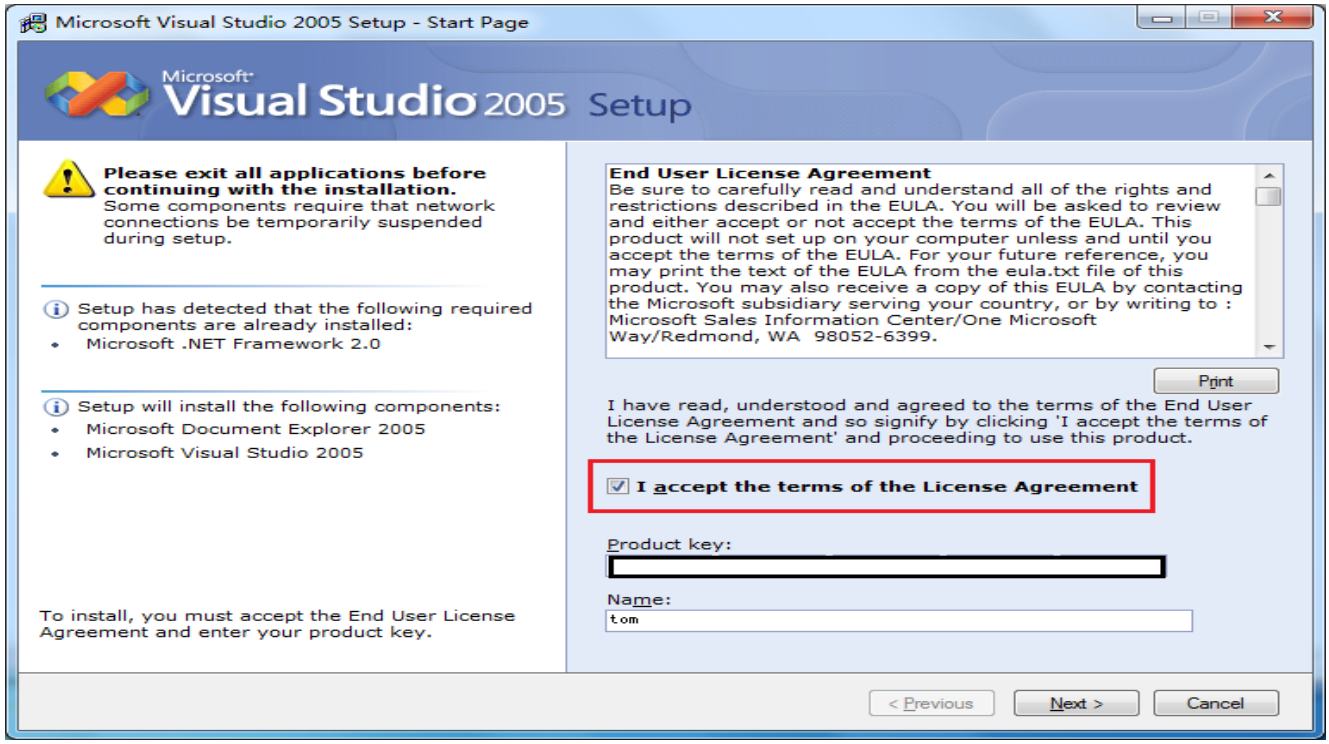


Step 2: In the following dialog click on “Install Visual Studio 2005”

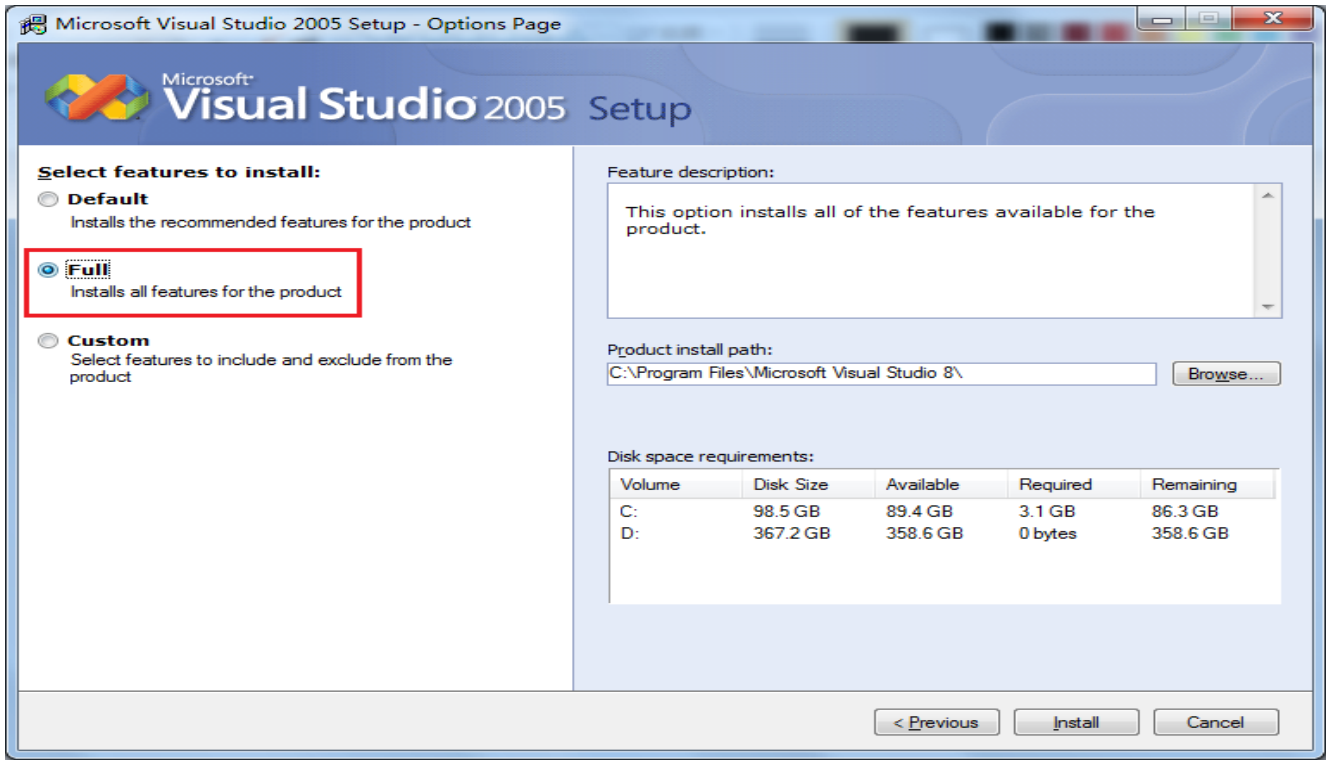
Step 3 Wait a while and click on “Next”



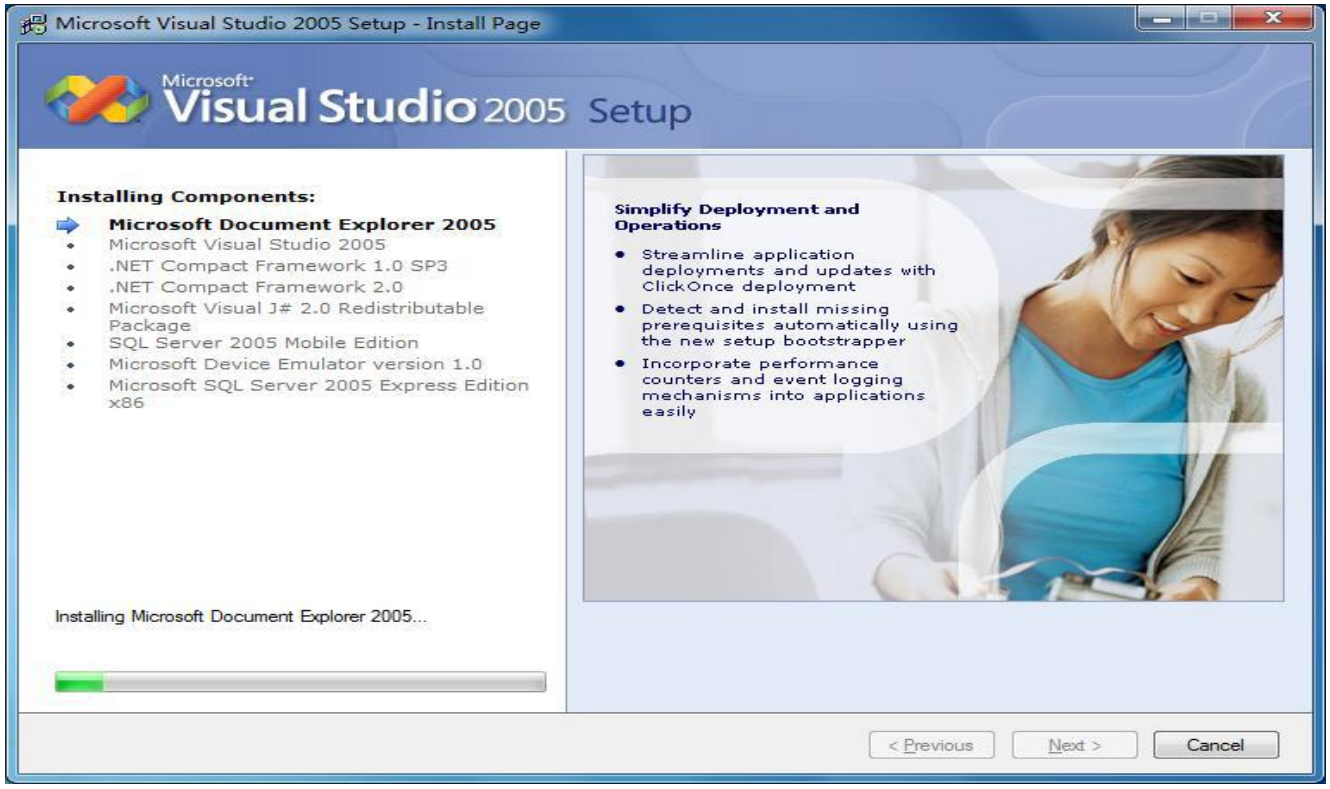
Step 4: type a serial number and click on “Next”



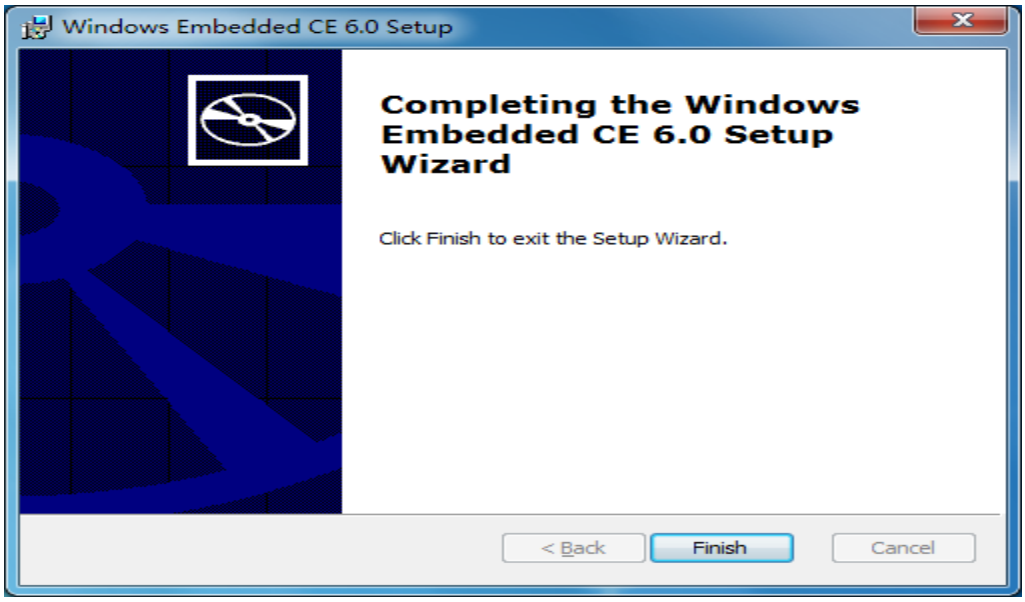
Step 5: Select “Full” and click on “Next”

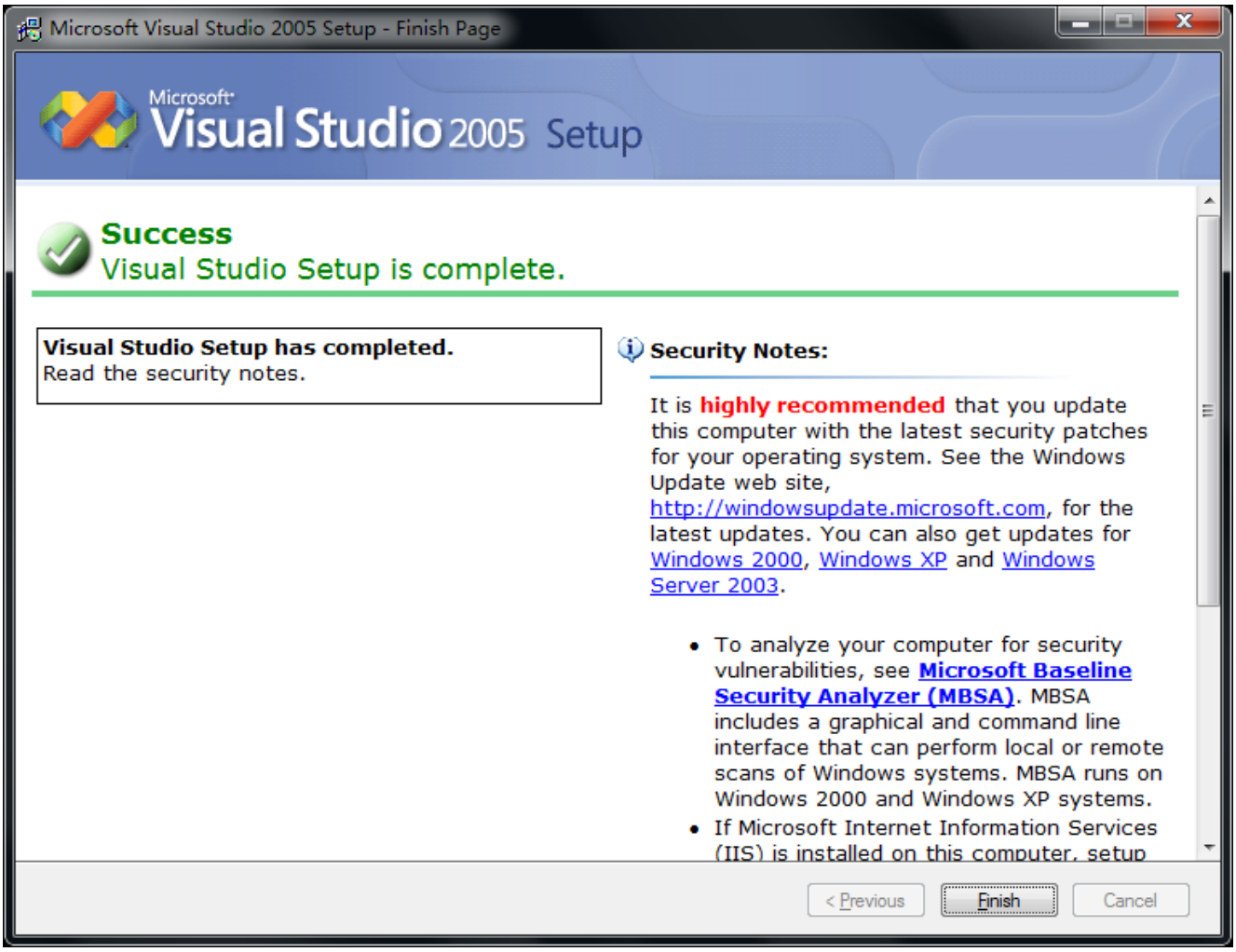


Step 6: Begin to install Visual Studio 2005



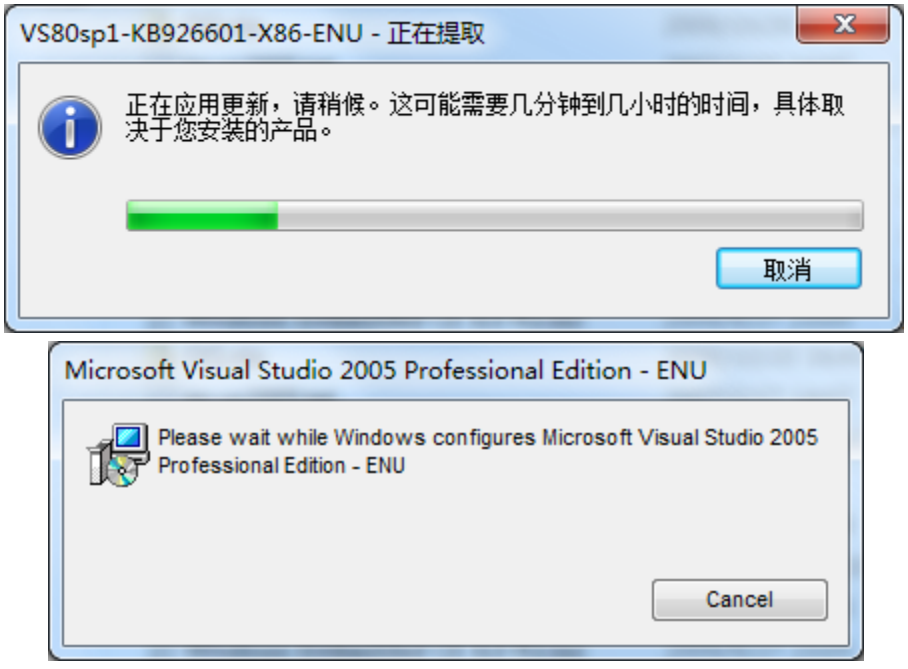
Step 7: After VS2005 installation is done, click on “Finish”





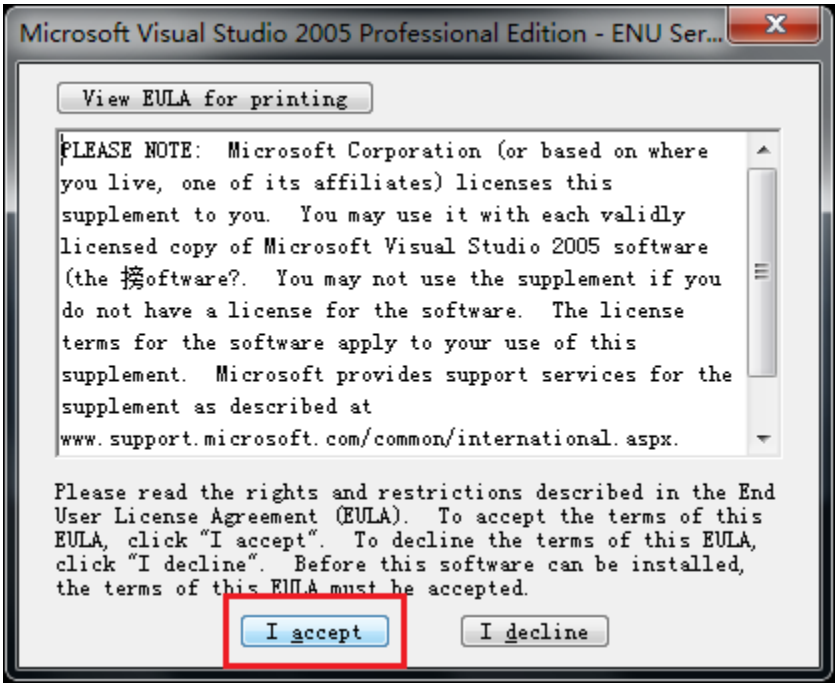
Click on “Exit” to complete

Step 8: Install Visual Studio 2005 Service Pack 1: double click on “VS80sp1-KB926601-X86-ENU.exe”

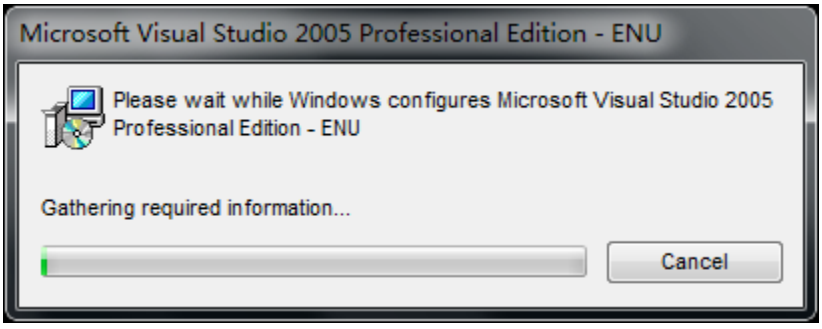


Step 9: wait a while and click on “OK” on the following dialog

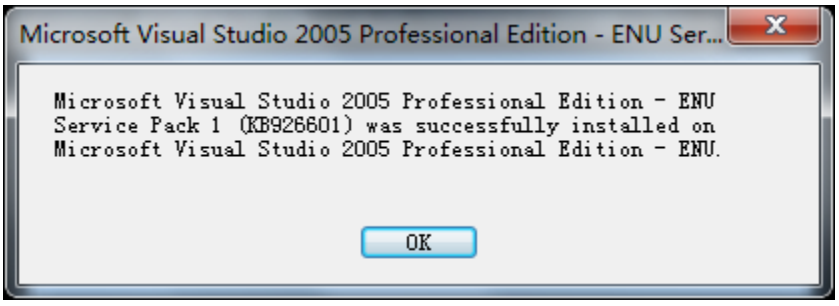
Step 10: Accept the aggrement



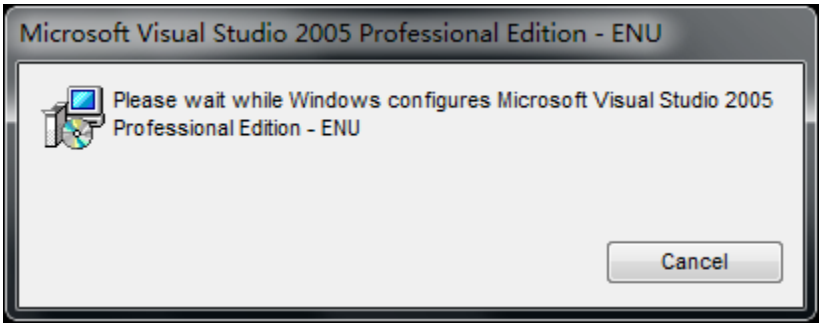
Step 11: Please wait for the installation



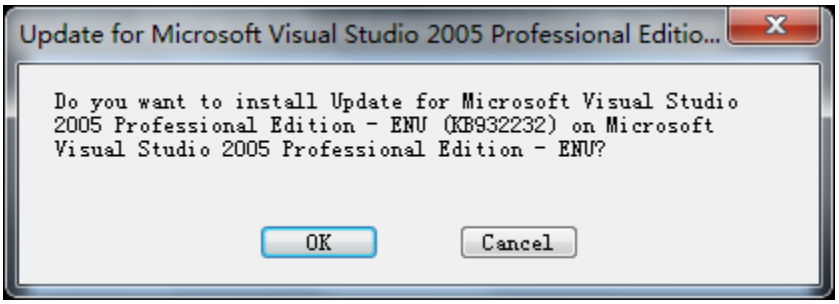
Step 12: Click “OK” to complete installation



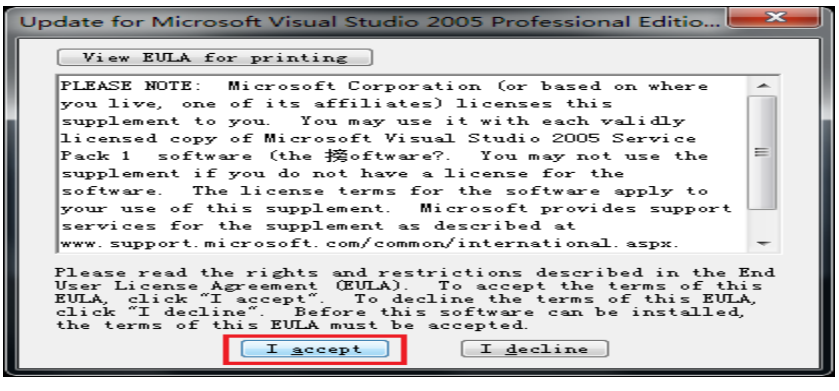
Step 13: Install Visual Studio 2005 Service Pack 1 Update for Windows Vista, double click on VS80sp1-KB932232-X86-ENU.exe



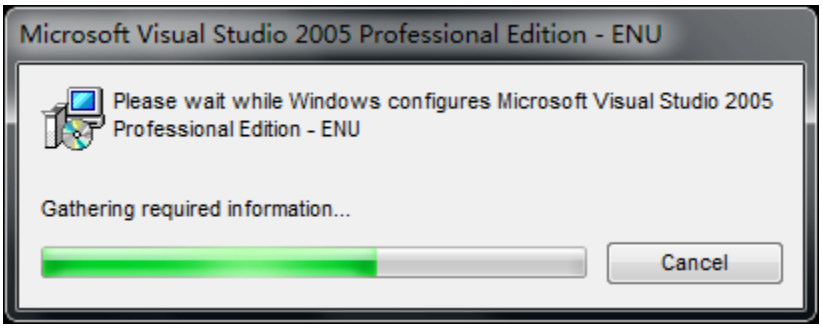
Step 14: Click on "OK" to continue



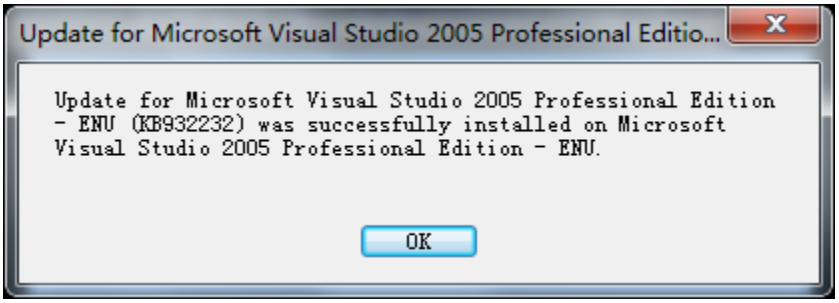
Step 15: accept the aggreement



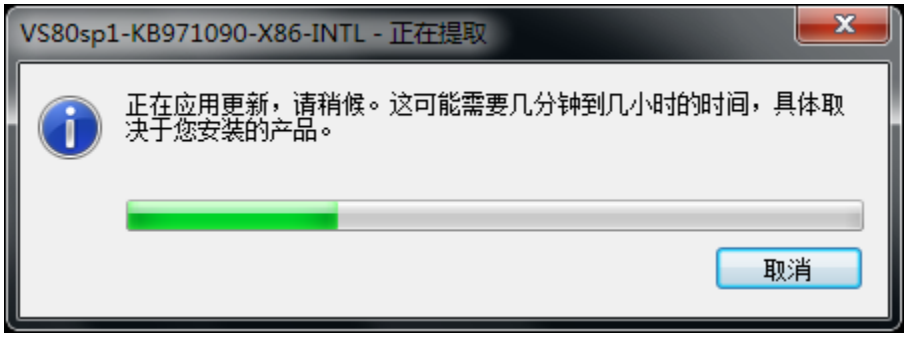
Step 16: Please wait for the installation

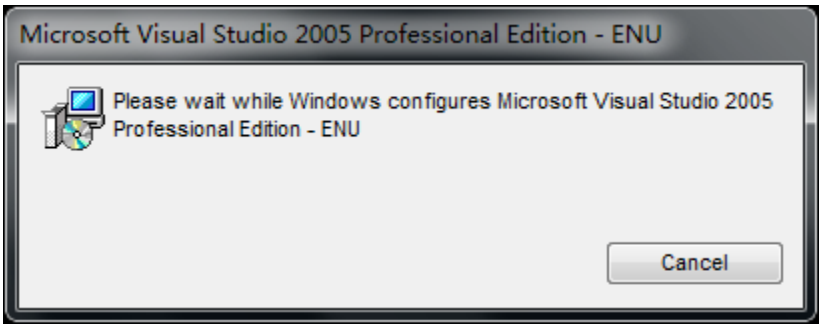


Step 17: After installation is done click on “OK” to complete the installation



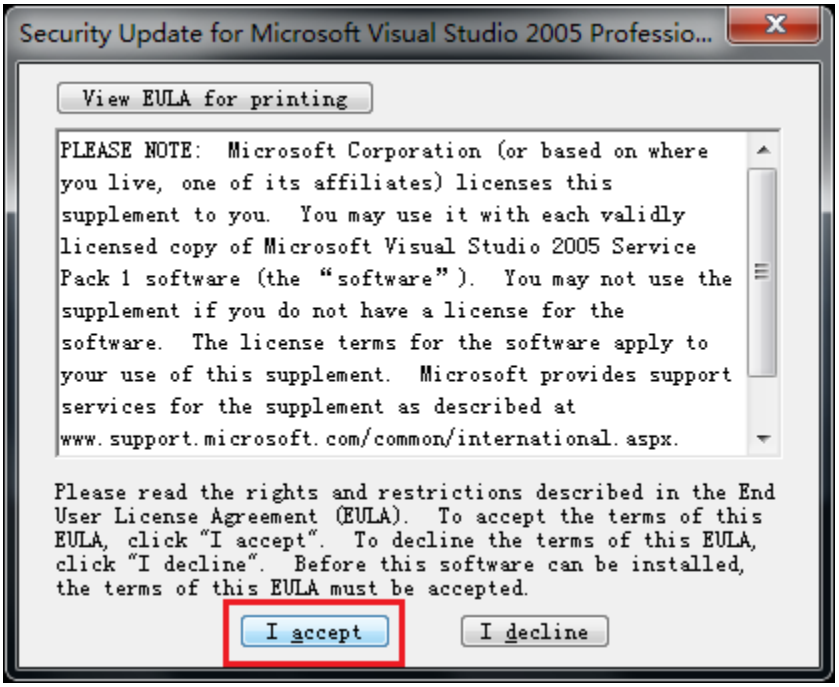
Step 18: Install Visual Studio 2005 Service Pack 1 ATL Security Update, double click on VS80sp1-KB971090-X86-INTL.exe



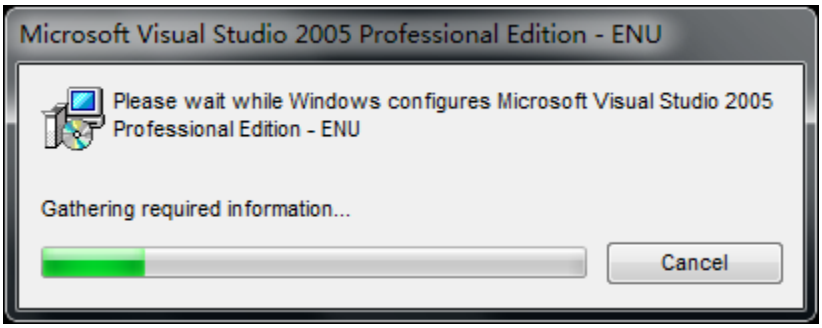


Step 19: Wait a while please click on “OK” to continue

Step 20: accept the agreement



Step 21: Wait for the installation



Step 22: After installation is done, click on “OK” to complete

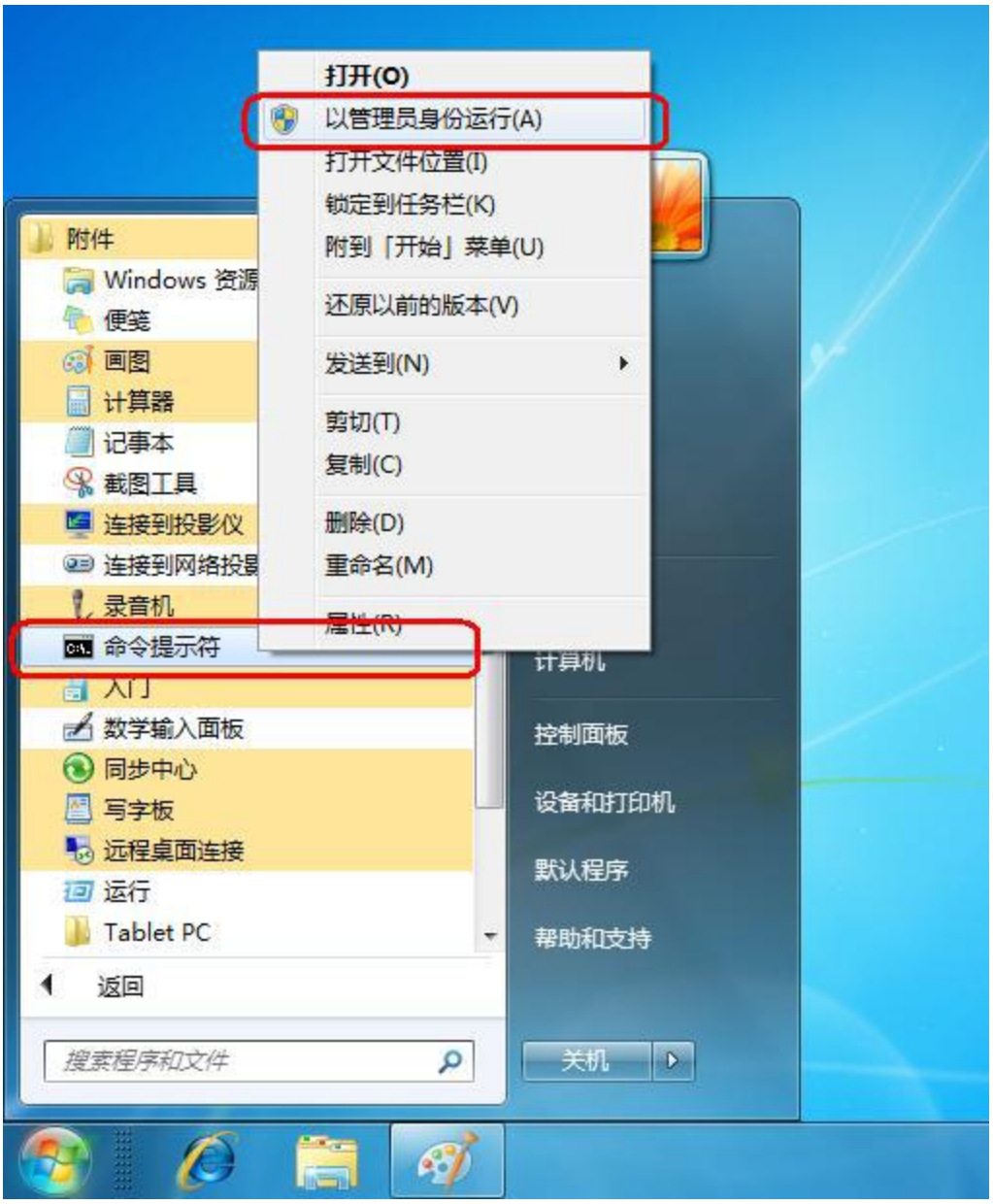
Now all Windows 7 and Visual Studio 2005 components have been installed

10.1.2 Install Windows CE 6 and Patches

In this section we will show you how to install Windows CE6 and Platform Builder.

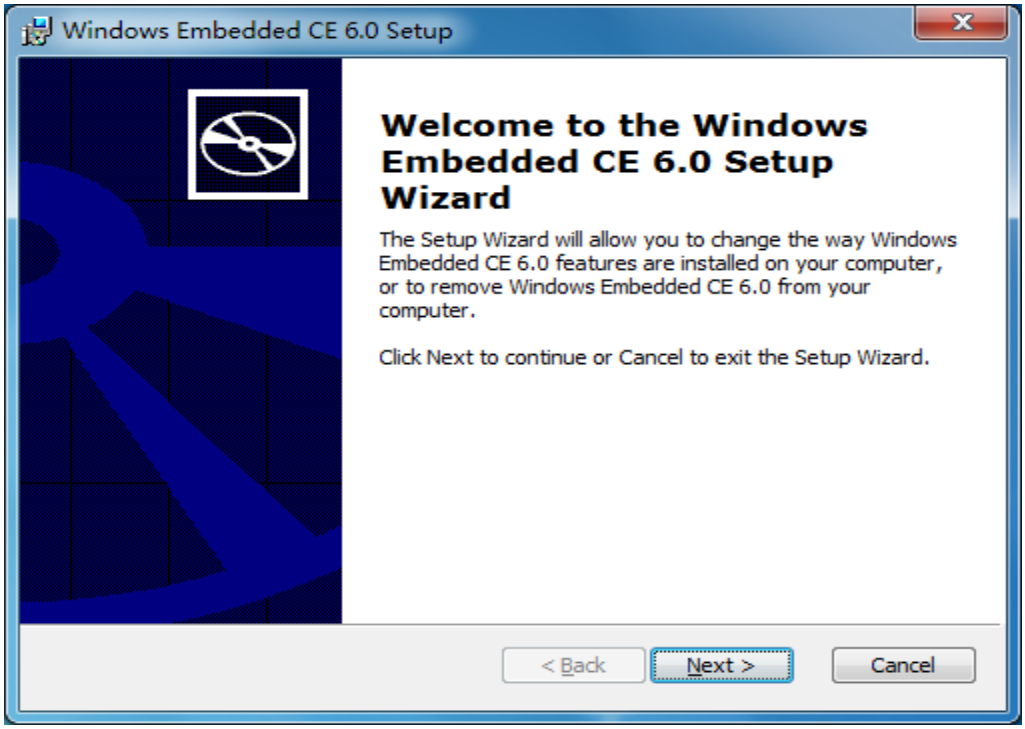
Note: to install Windows CE6 on Windows 7 you need to do it as administrator.

Step 1: go to “Start” -> “All Programs” -> “Accessories”, move to “command line”, right click and select “run as administrator”

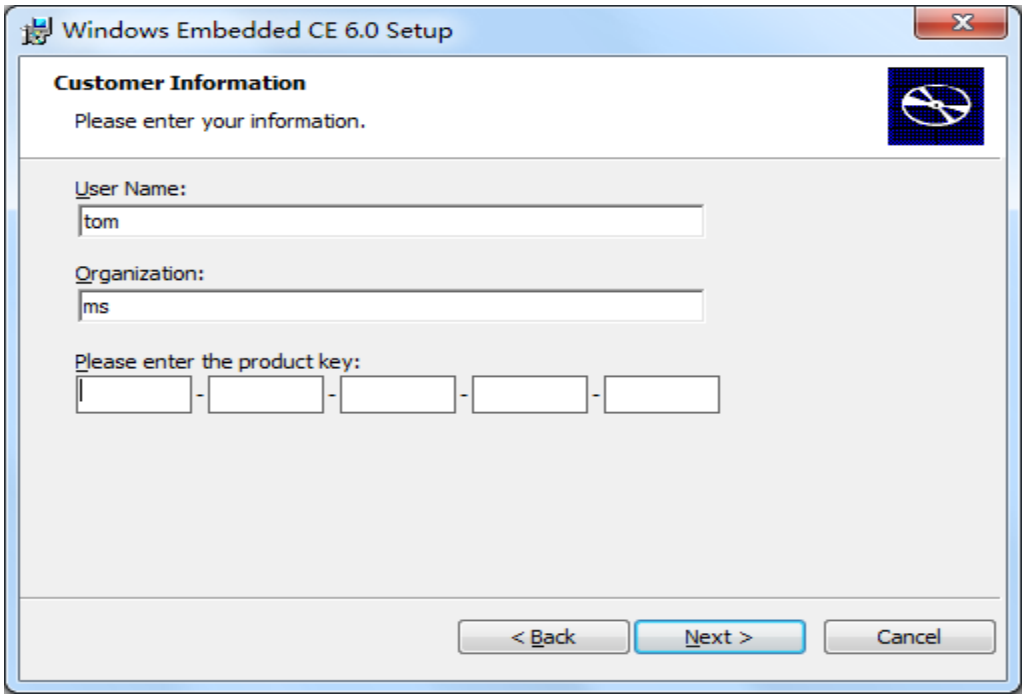


Step 2: in the command line window go to the installation directory type the installation program “Windows Embedded CE 6.0.msi” to begin installation

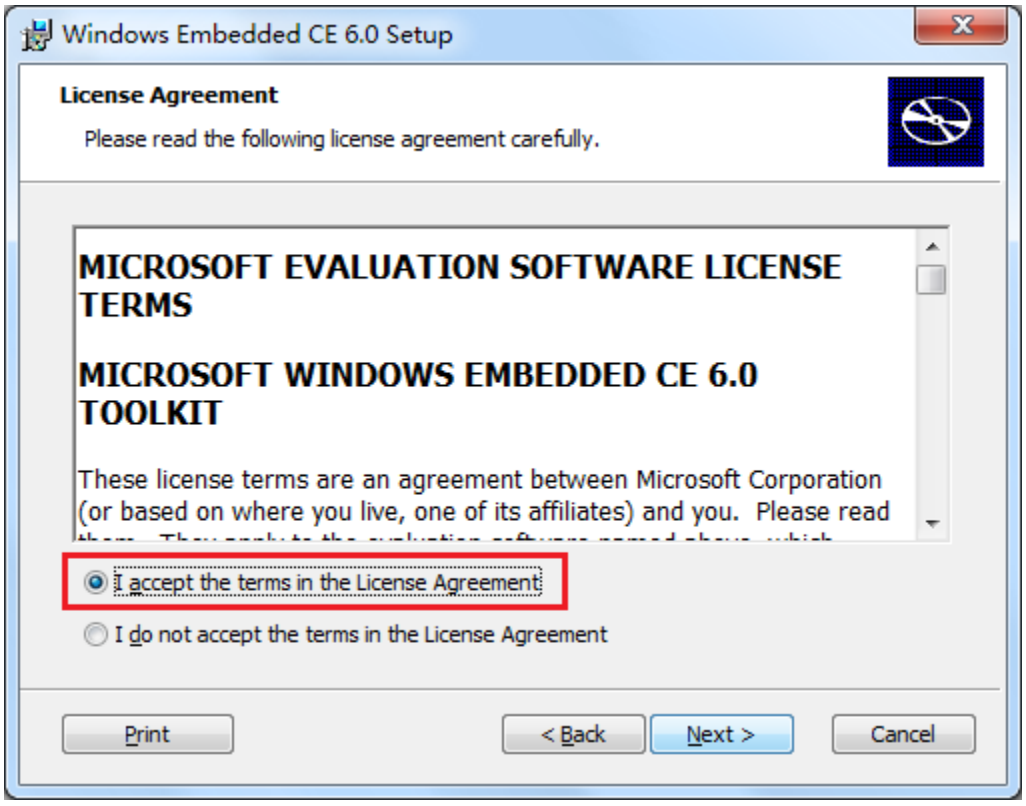
Step 3: click on “Next” to continue



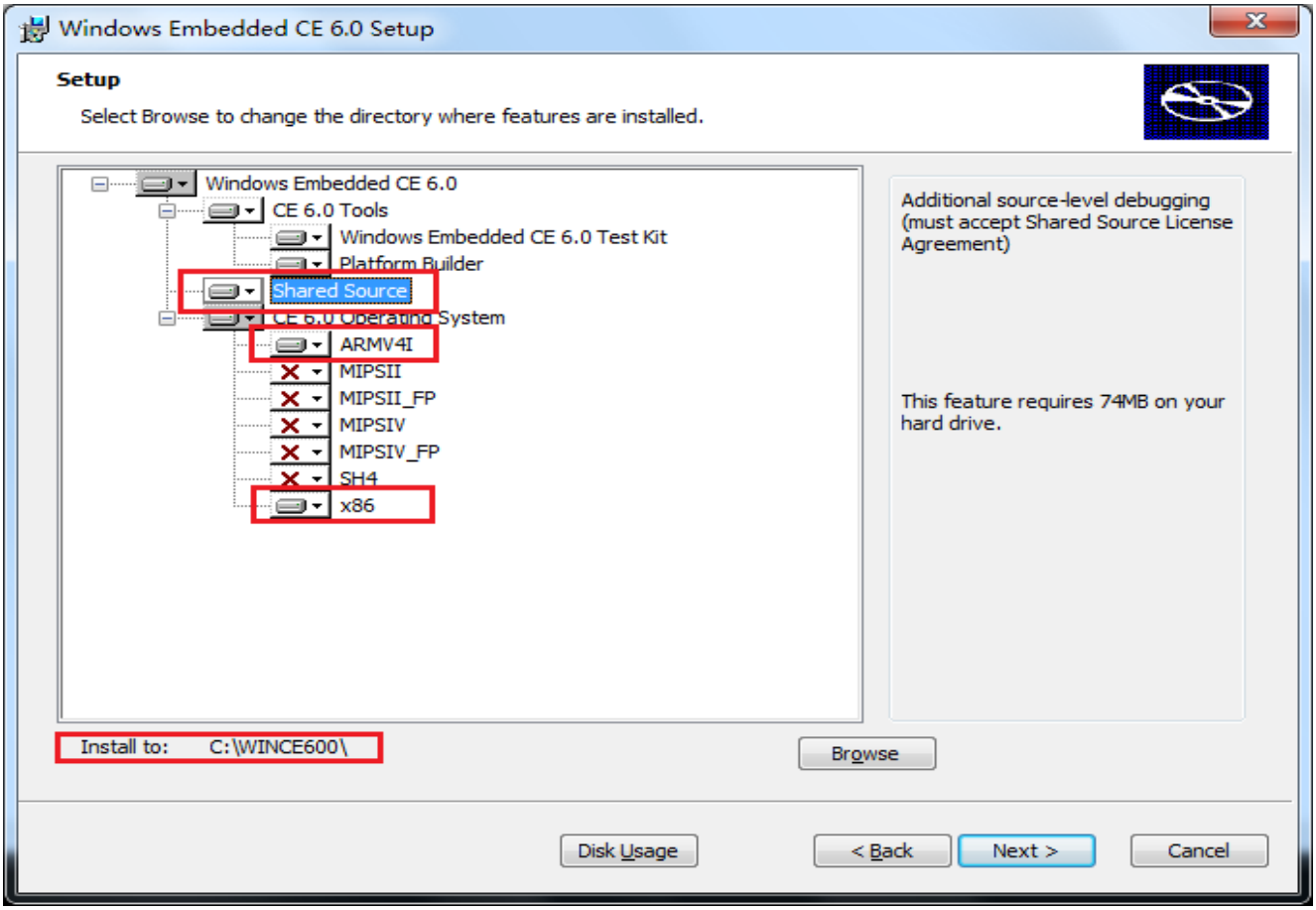
Step 4: type a serial number and click on “Next” to continue



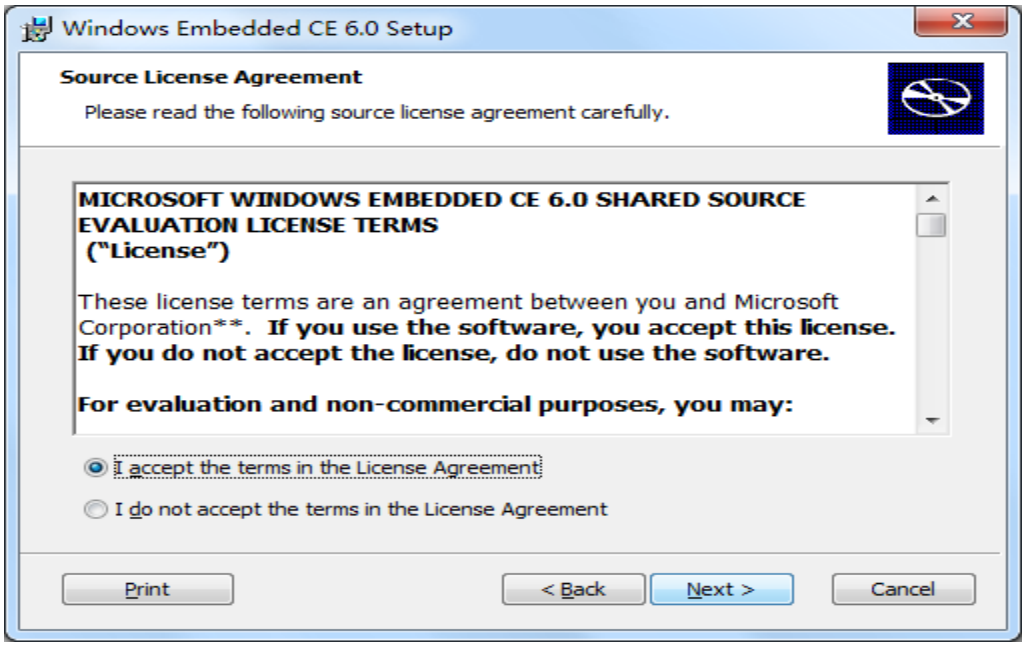
Step 5: accept the agreement and click on “Next” to continue



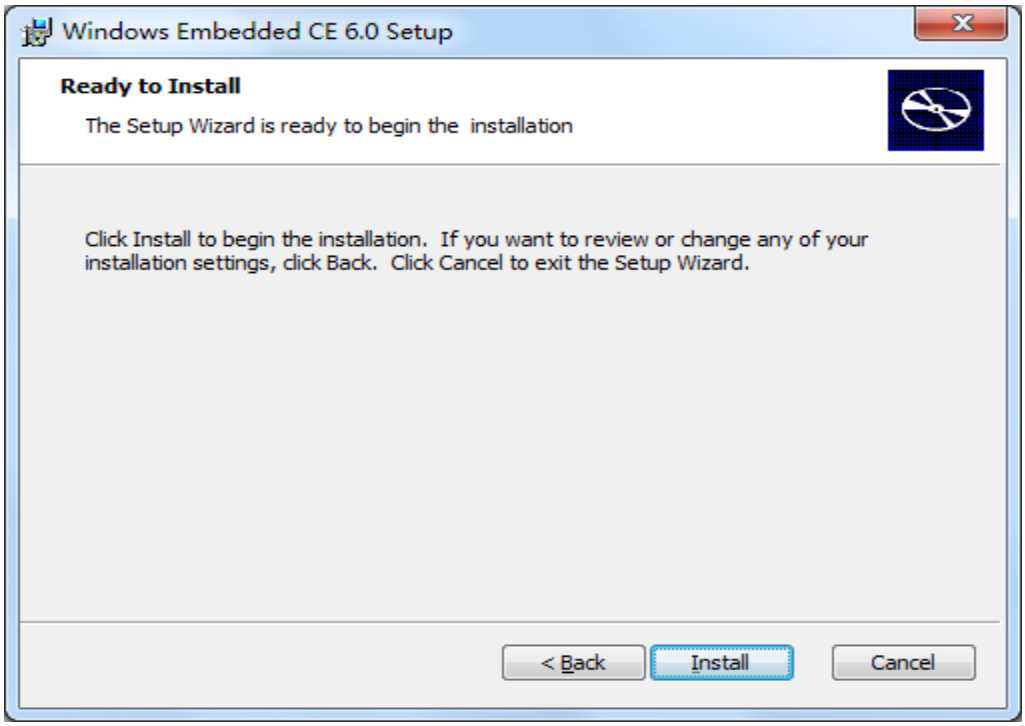
Step 6: select options as below and click on “Next” to continue



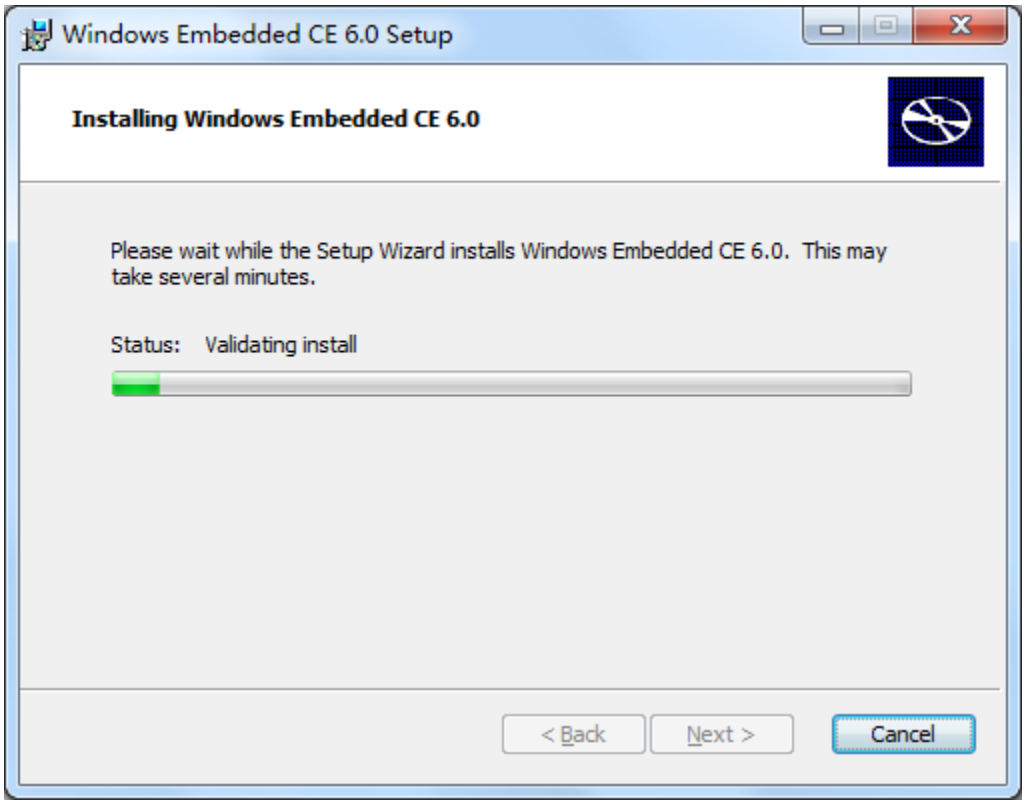
Step 7: click on “Next” to continue



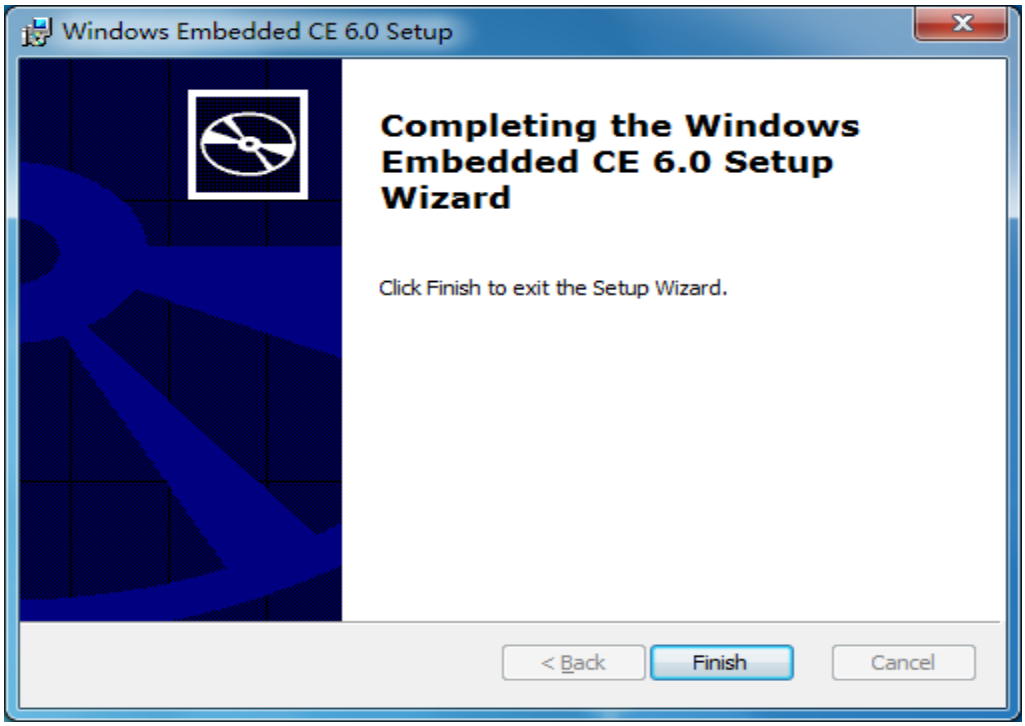
Step 8: click on “Install”



Step 9: Installation begins



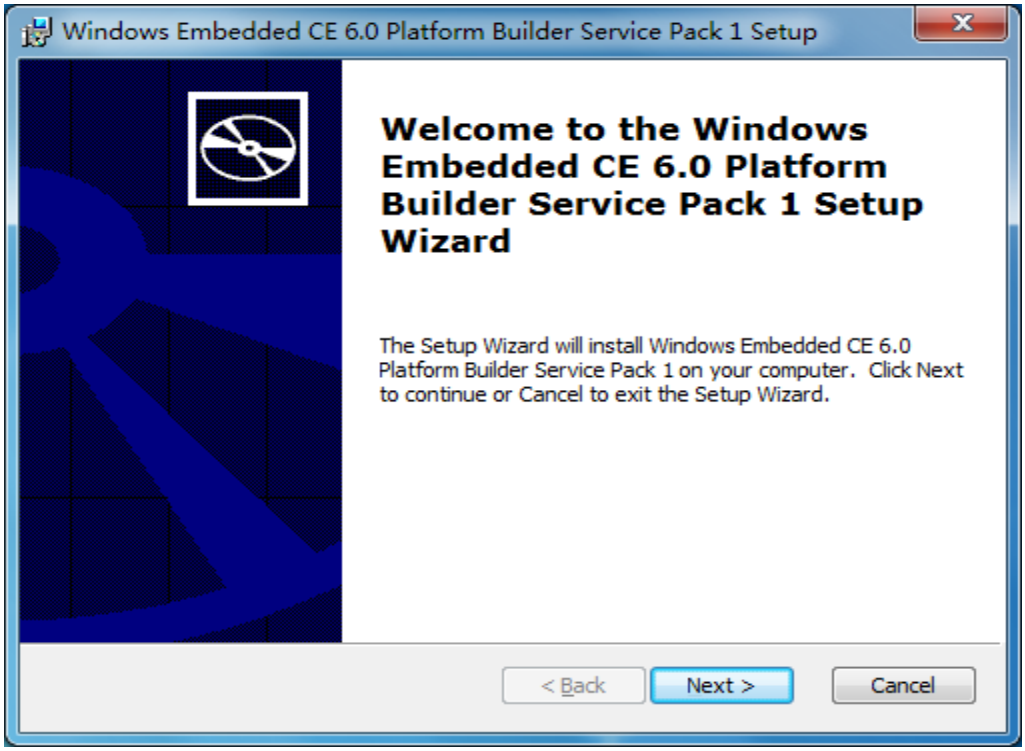
Step 10: after installation is done, click on “Finish”



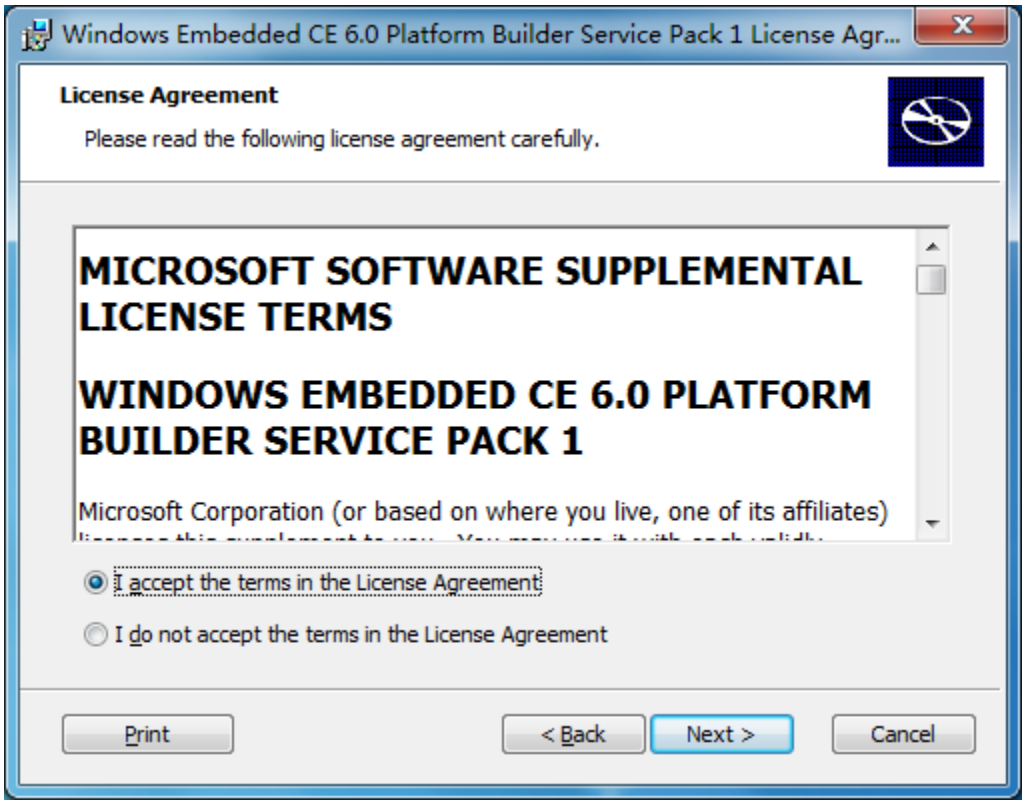
Step 11: Install “Windows Embedded CE 6.0 Platform Builder Service Pack 1.msi”.

Follow the instructions in Step 1 to install it.

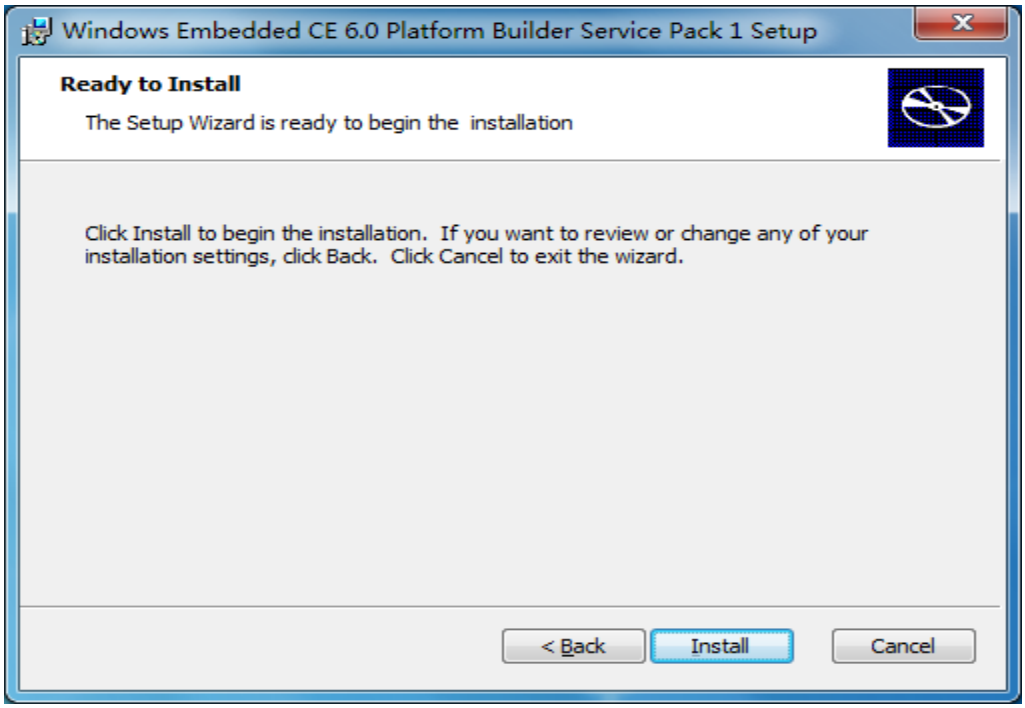
Step12: Click on “Next” to continue:



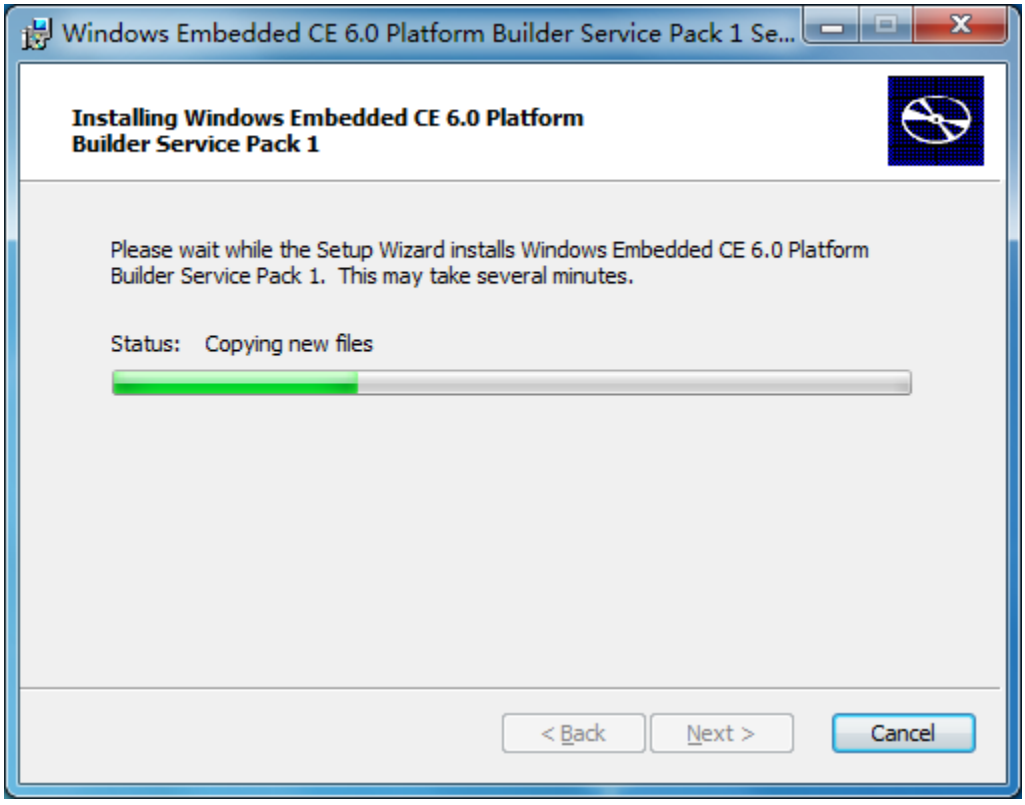
Step 13: Accept the agreement and click on “Next” continue



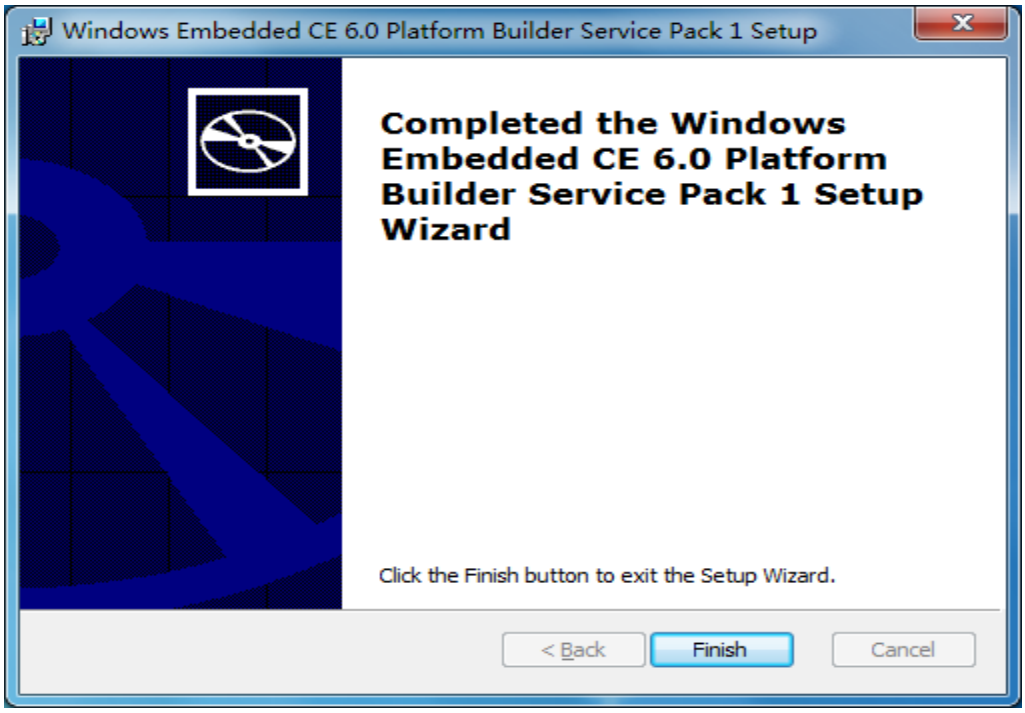
Step 14: click on “Next” to continue



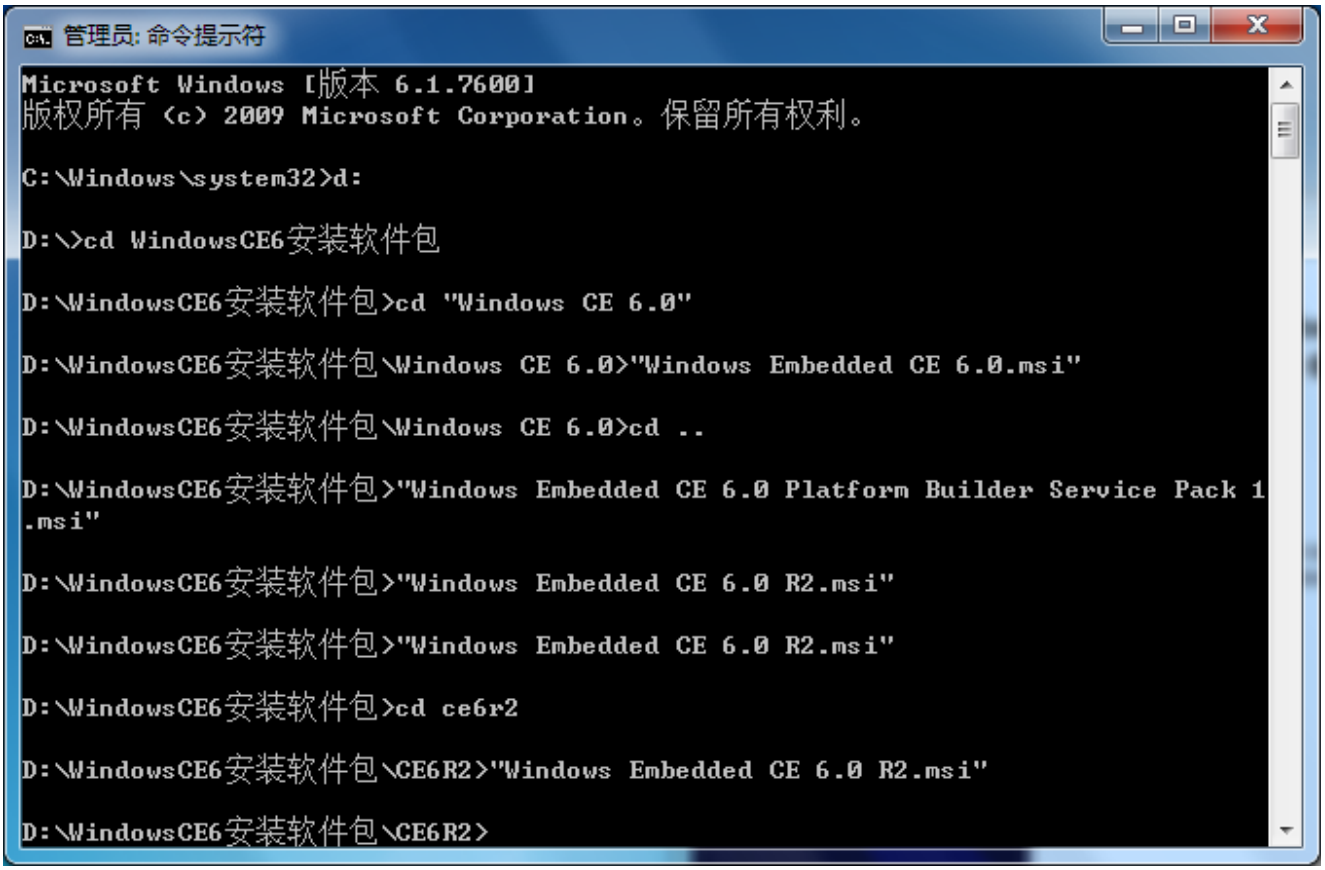
Step 15: wait for the installation process



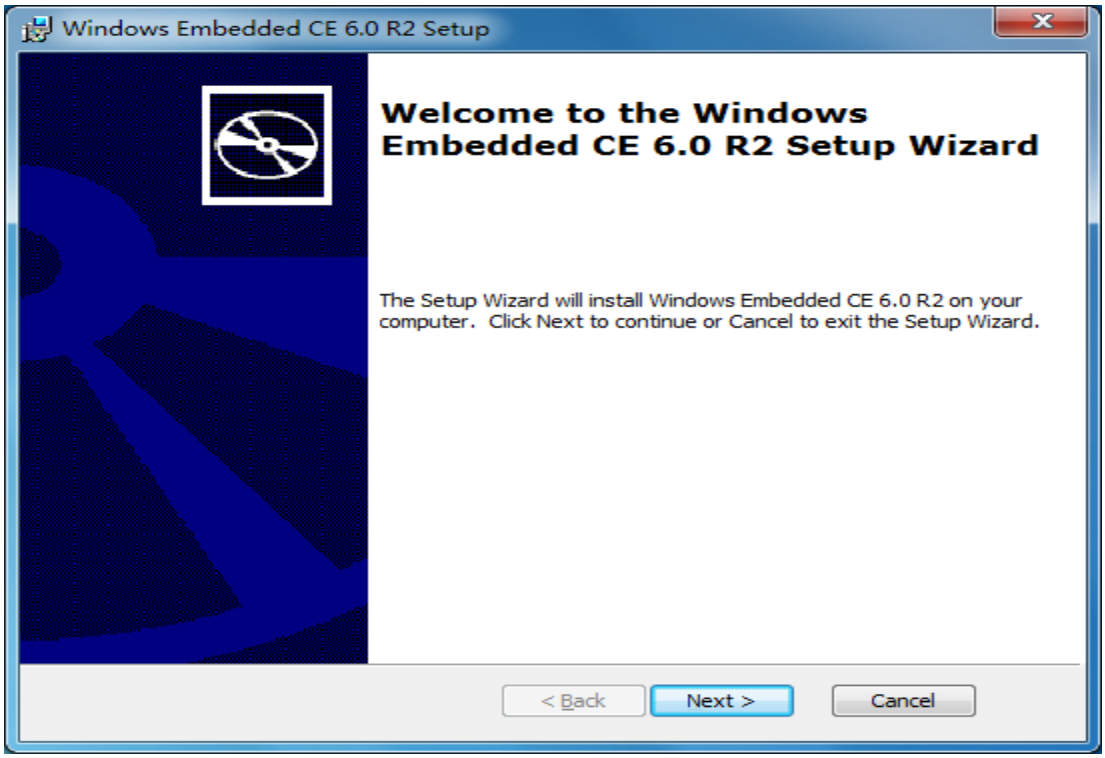
Step 16: after installation is done, click on “Finish”



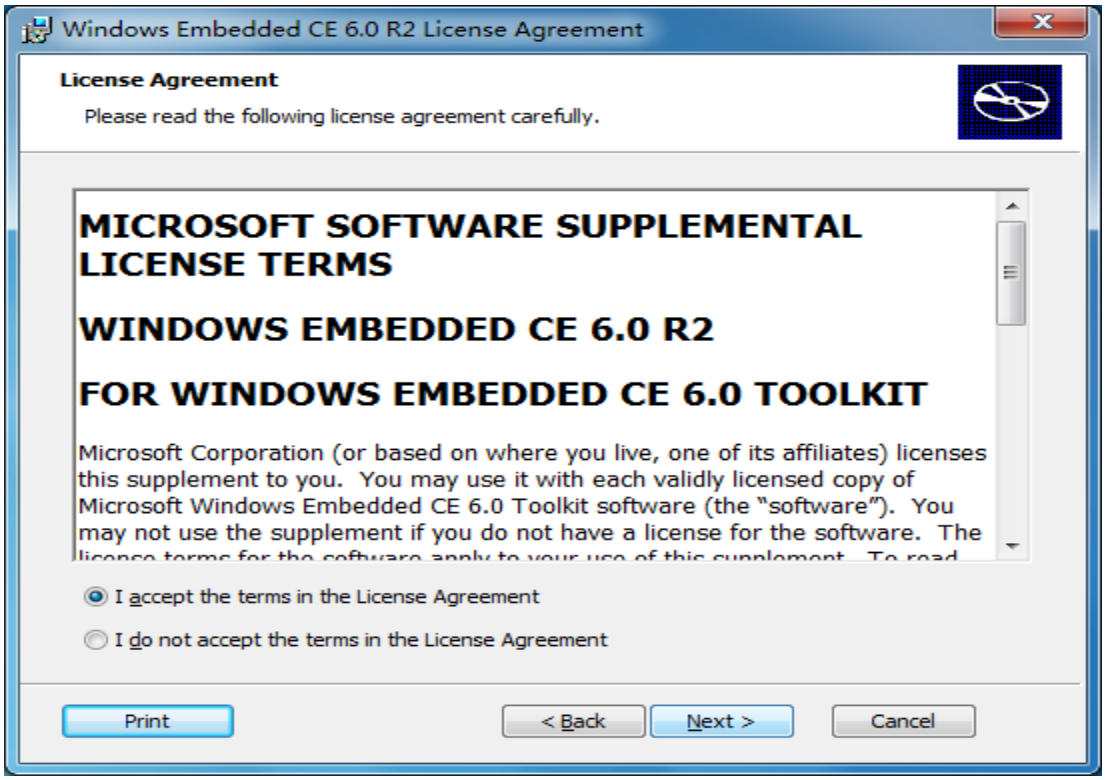
Step 17: Install “Windows Embedded CE 6.0 R2.msi”



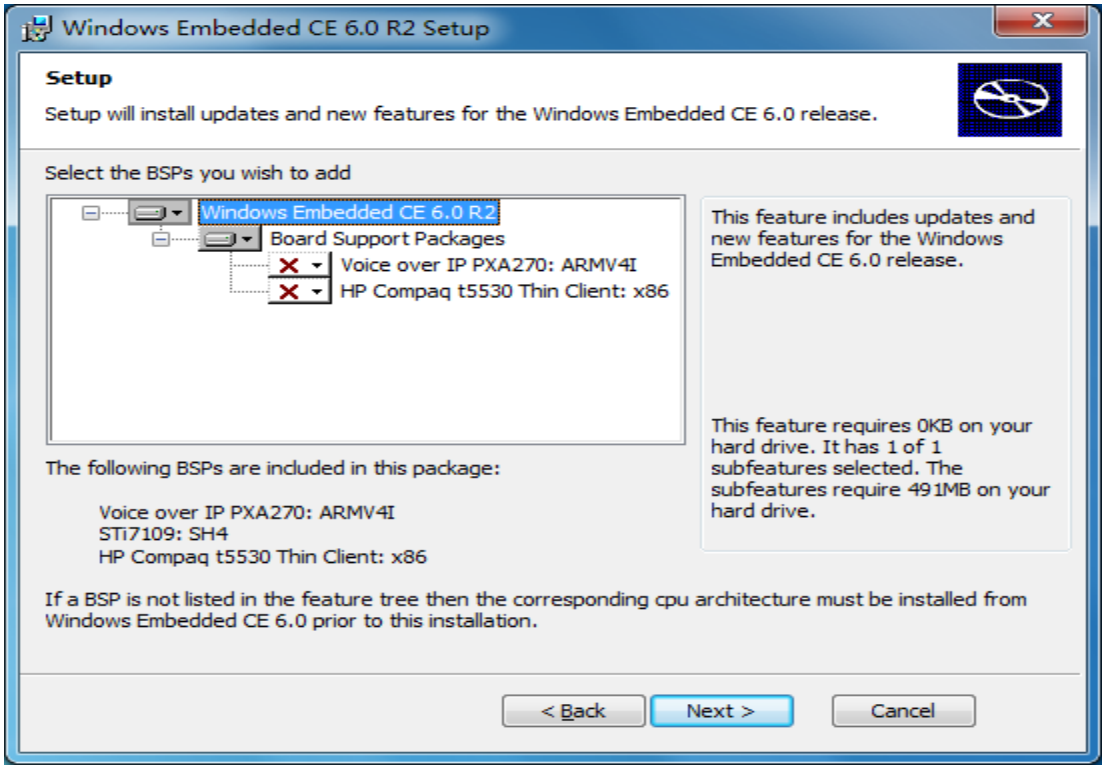
Step 18: Click on “Next” to continue



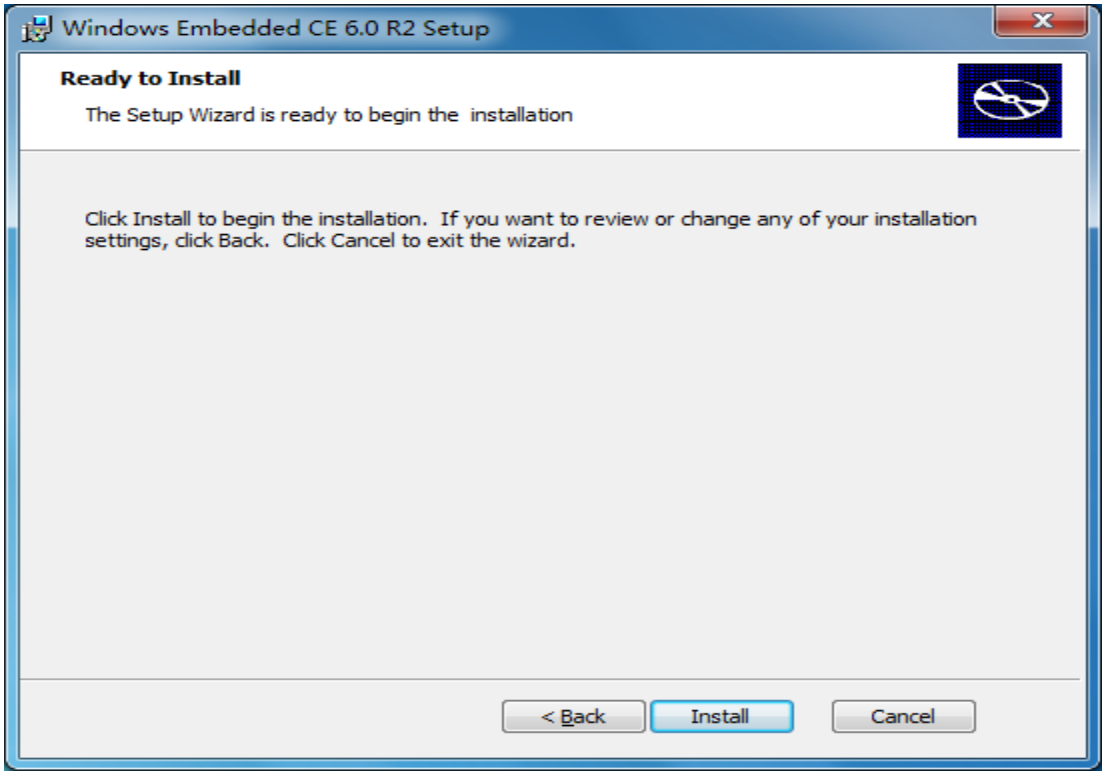
Step 19: accept the agreement and click on “Next” to continue



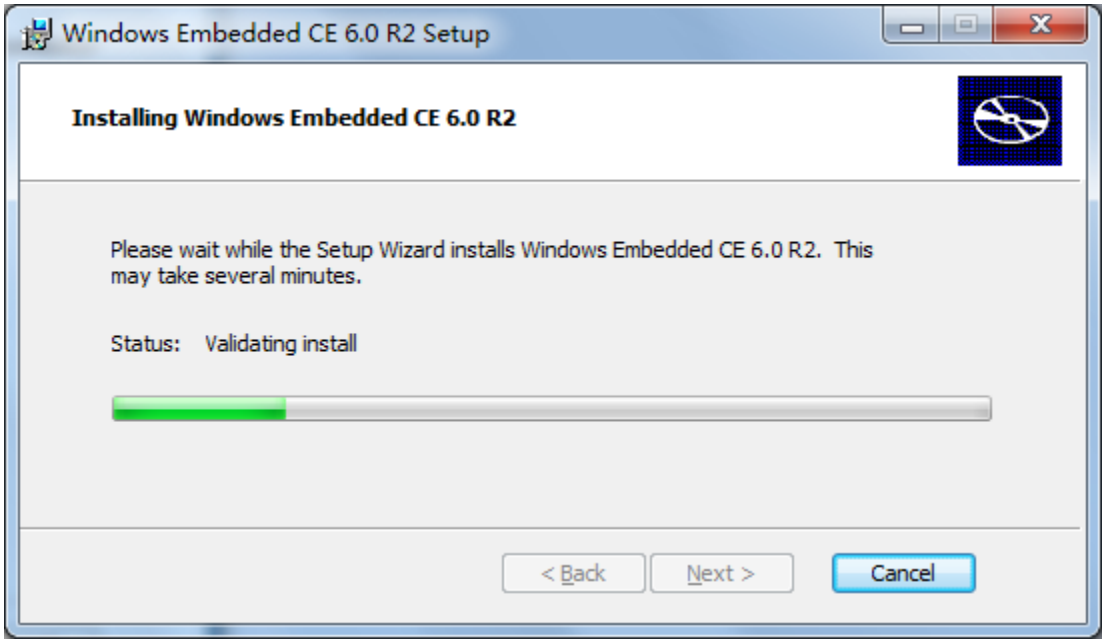
Step 20: Click on “Next” to continue



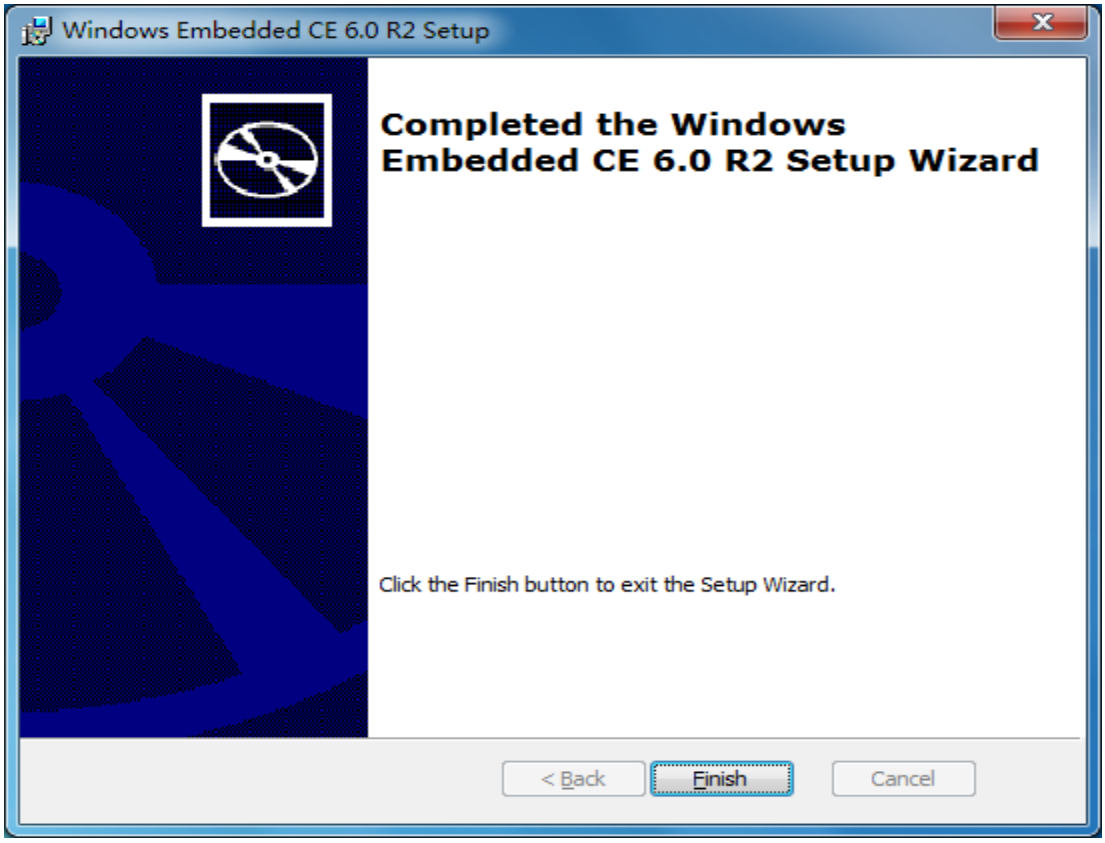
Step21: Click on “Next” to continue



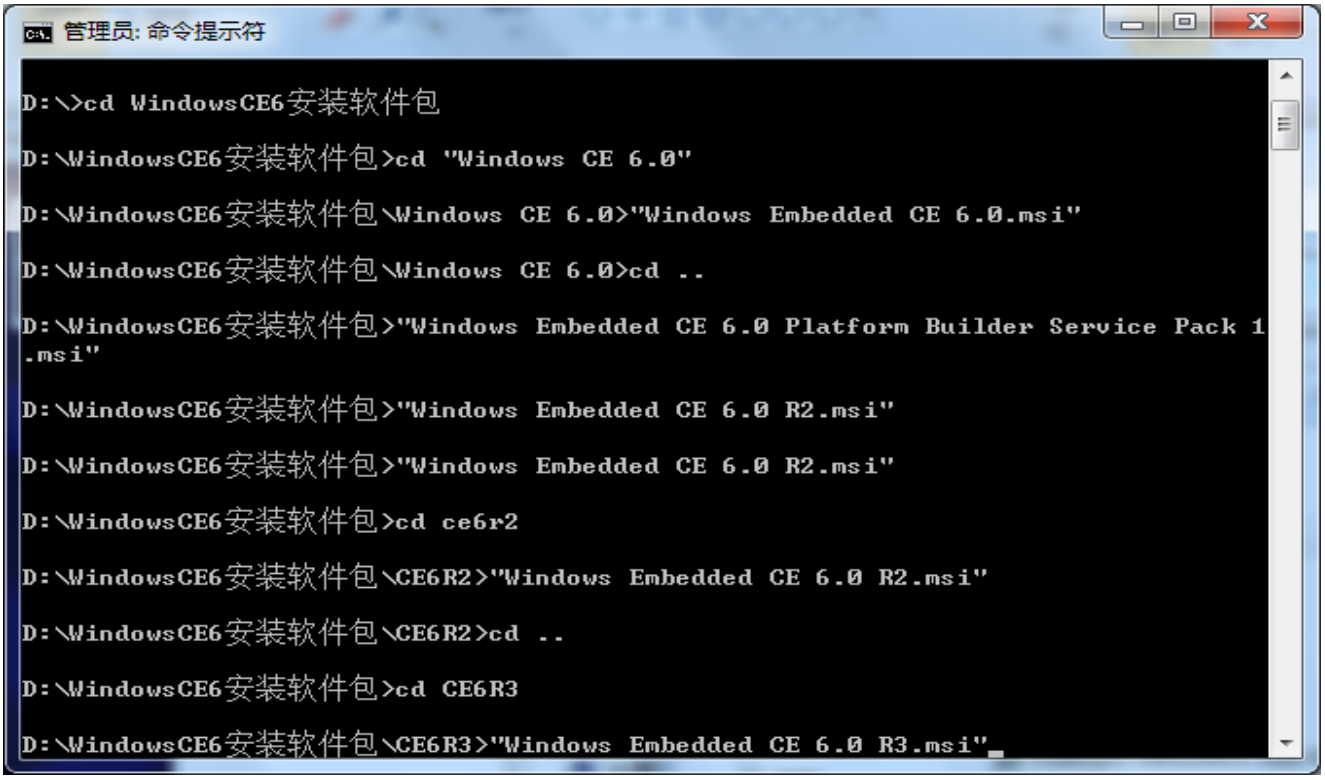
Step 22: Installation begins:



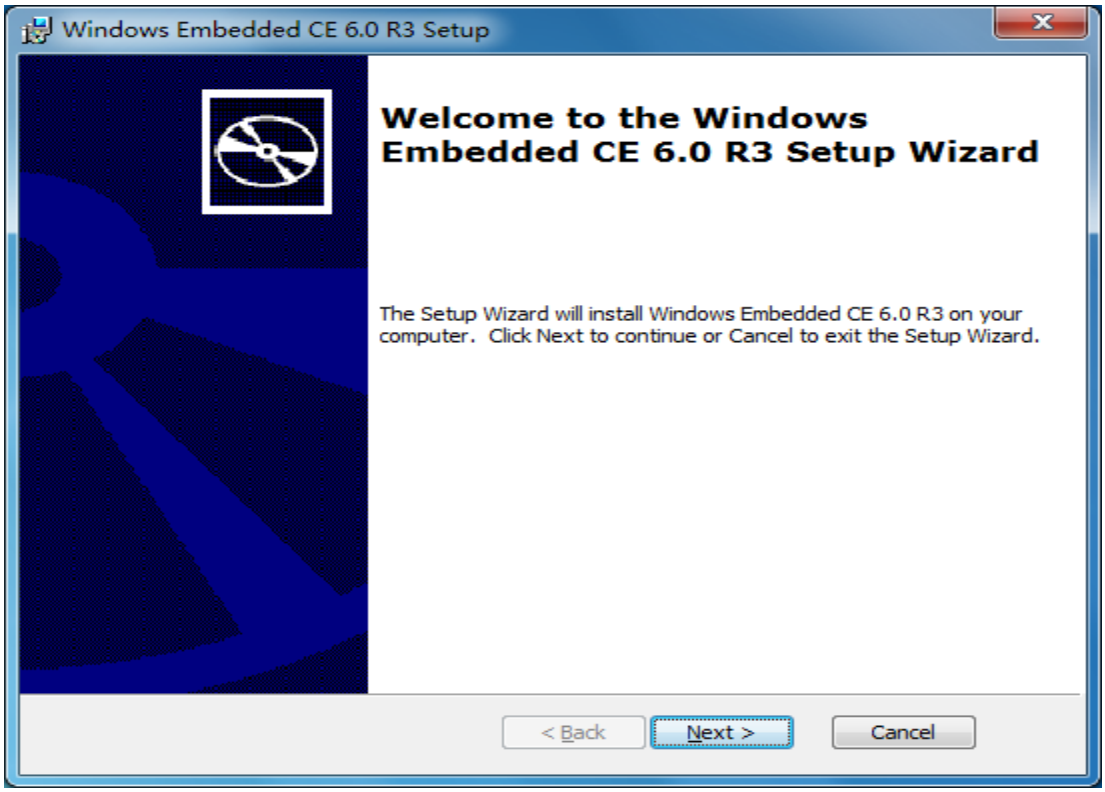
Step 23: After installation is done click on "Finish"



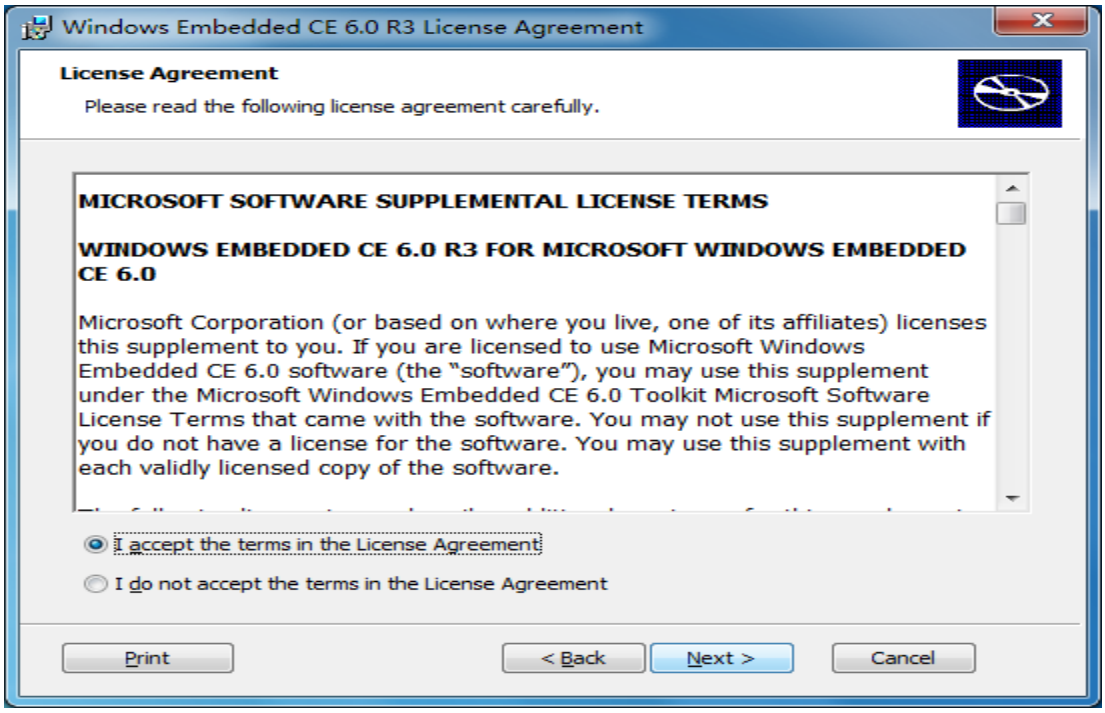
Step 24: Install “Windows Embedded CE 6.0 R2.msi”



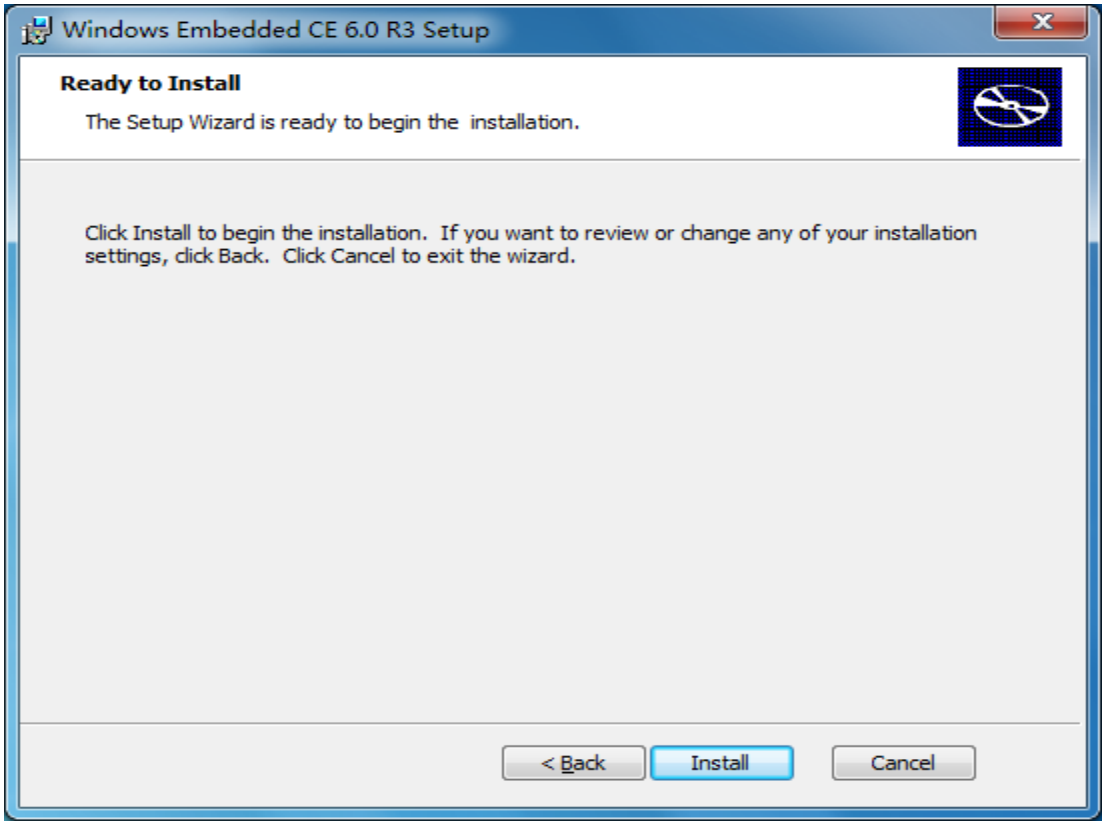
Step 25: Click on “Next” to continue



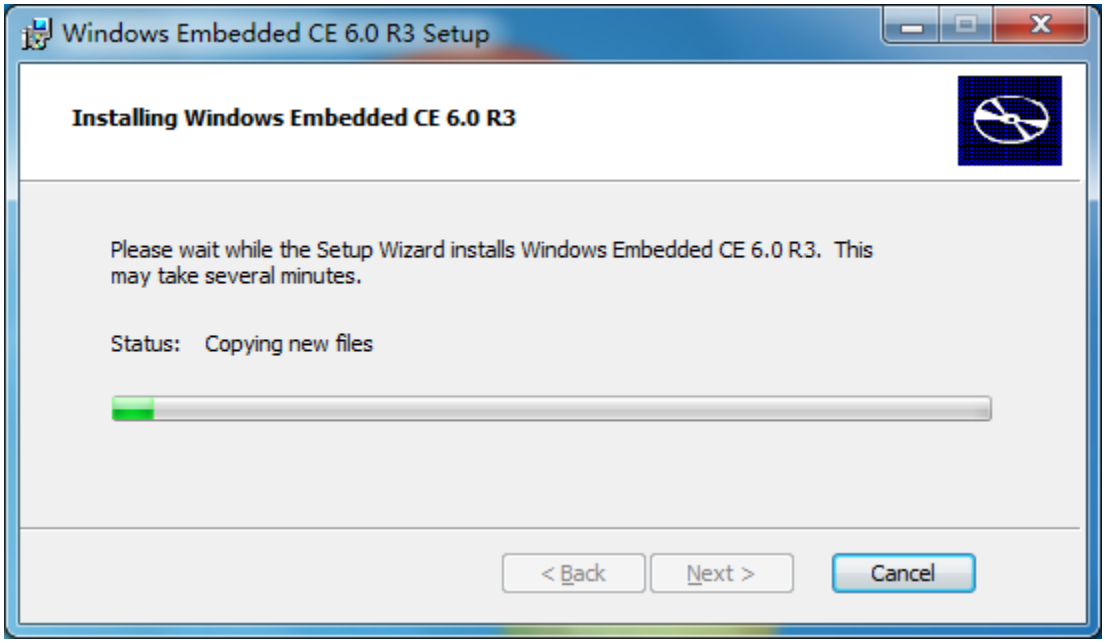
Step 26: accept the agreement and click on “Next” to continue



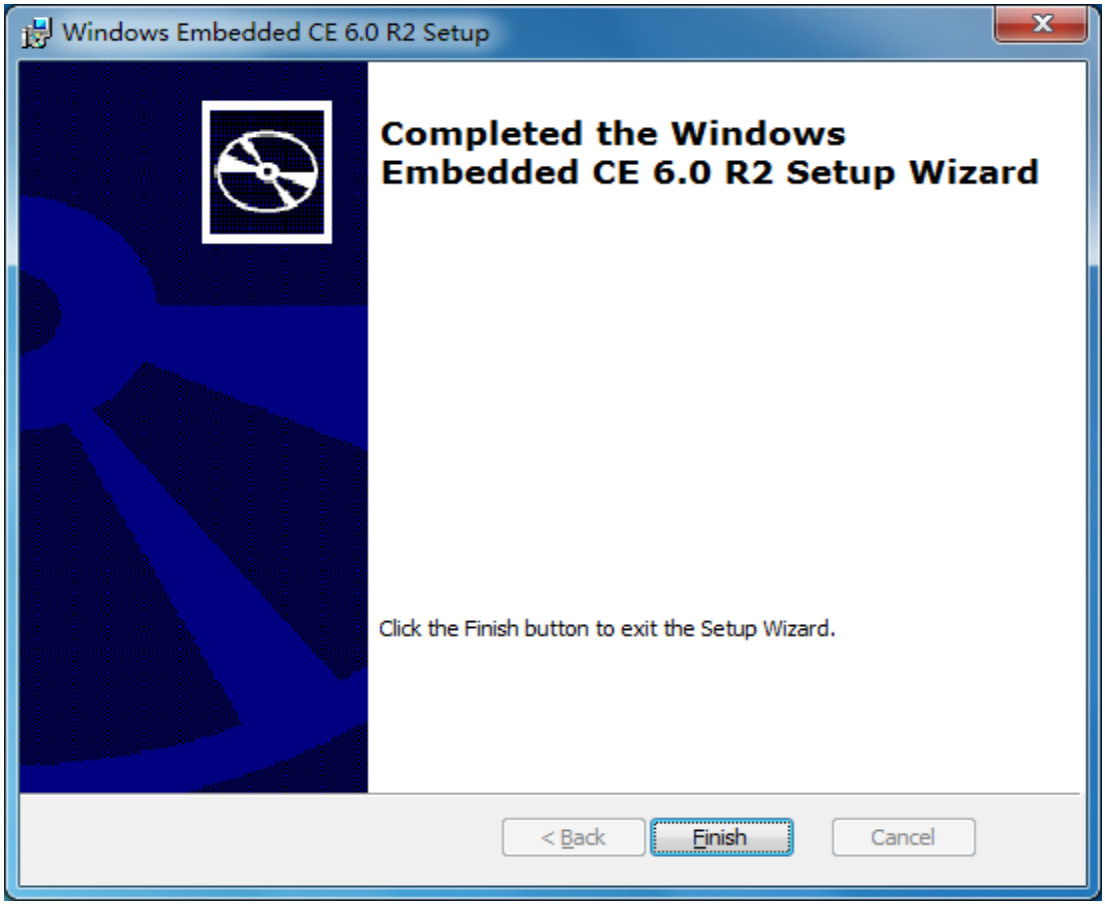
Step 27: Click on “Next” to continue



Step 28: Installation begins:



Step 29: After installation is done, click on "Finish"

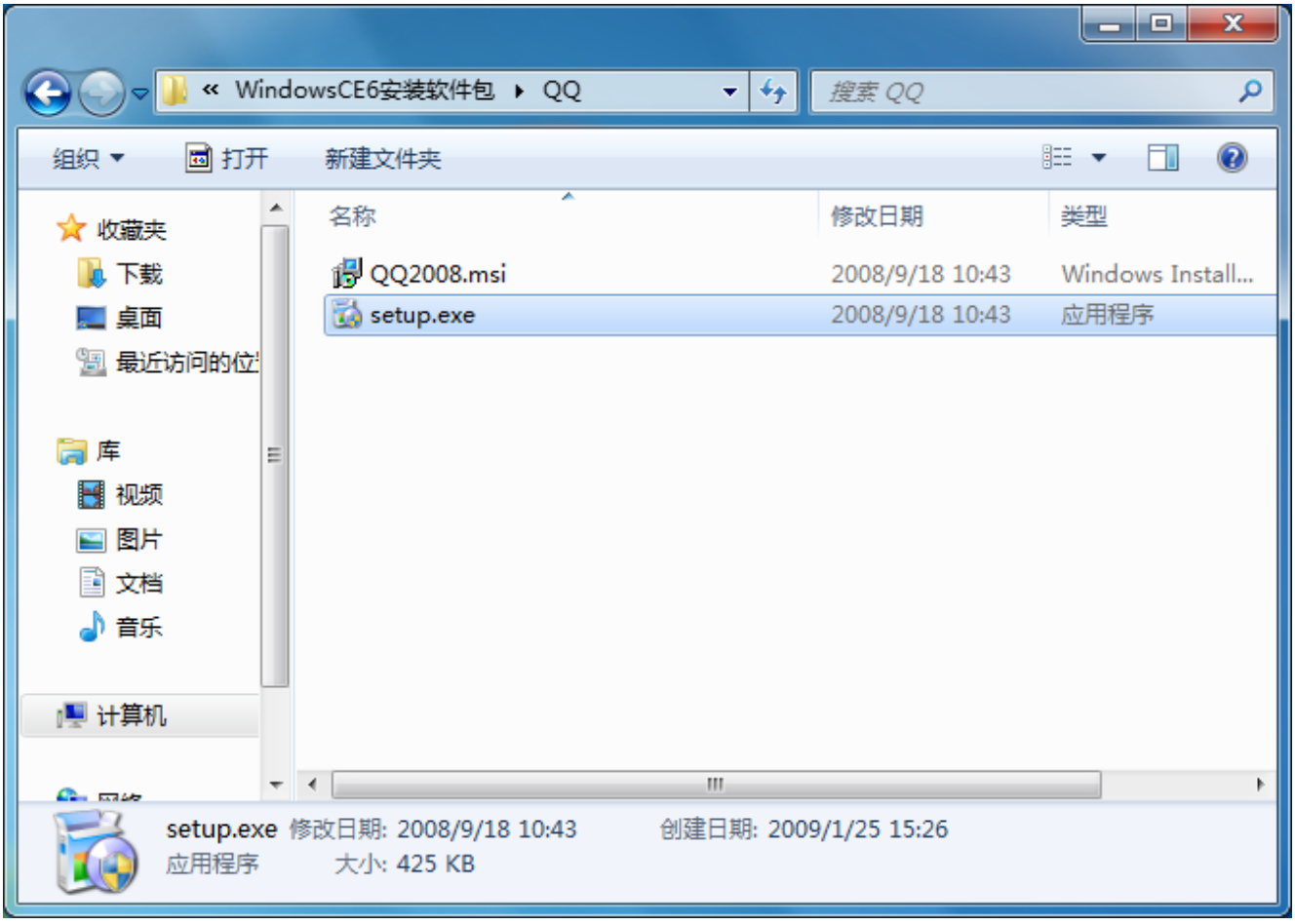


Step30: Install "WinCEPB60-101231-Product-Update-Rollup-Armv4I.msi". The download address is

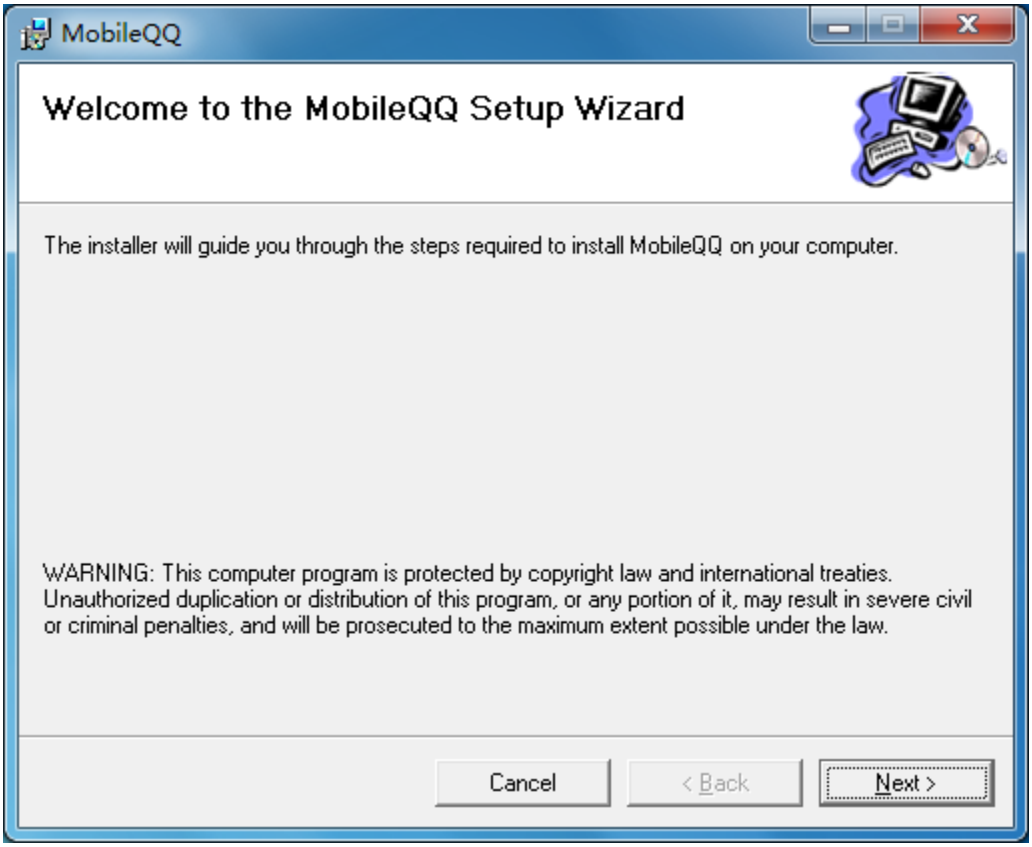
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=1127

10.1.3 Install Third Party Software Tencent QQ

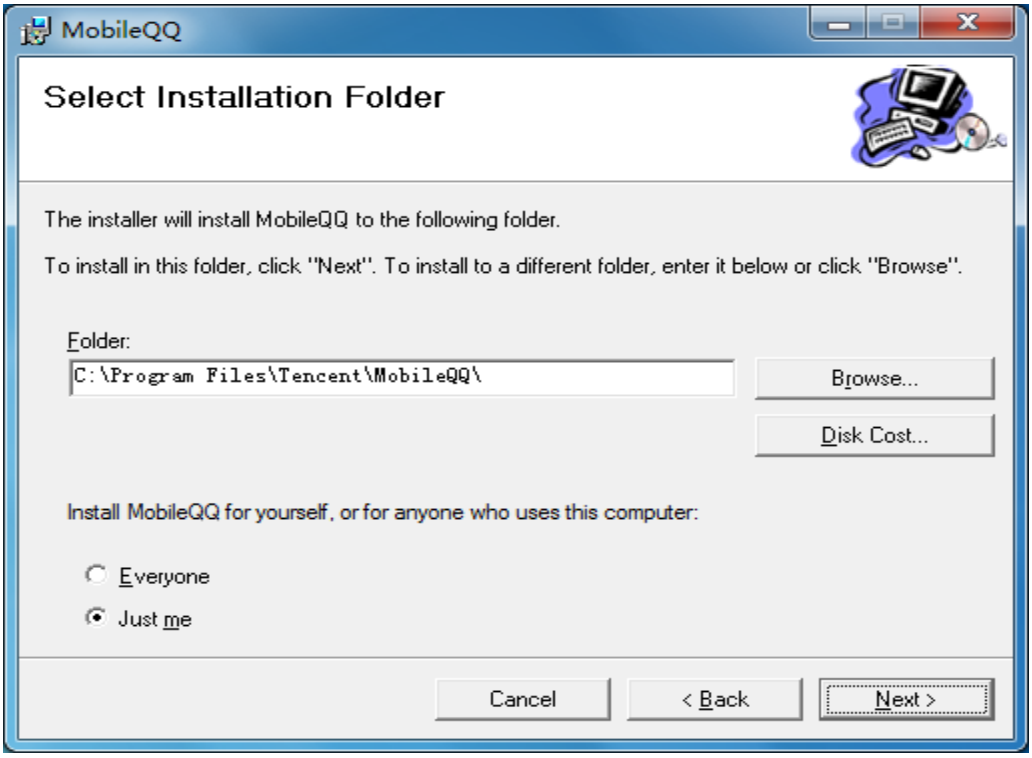
Step1: enter the QQ installation directory and double click on setup.exe to begin installation



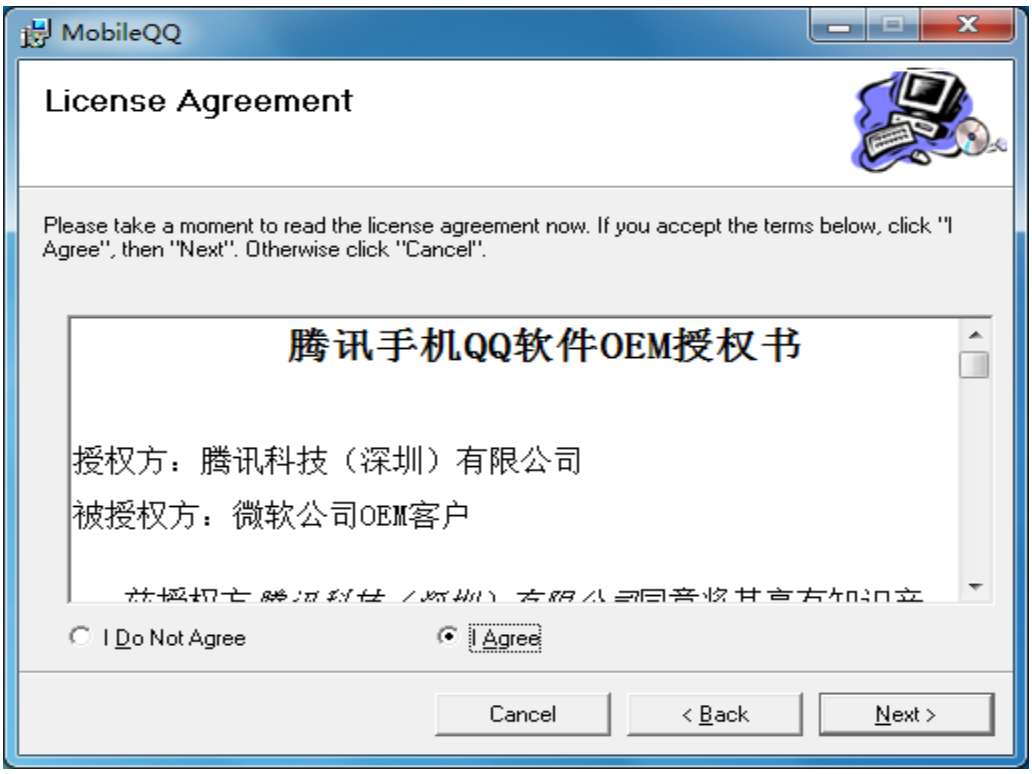
Step 2: click on "Next" to continue



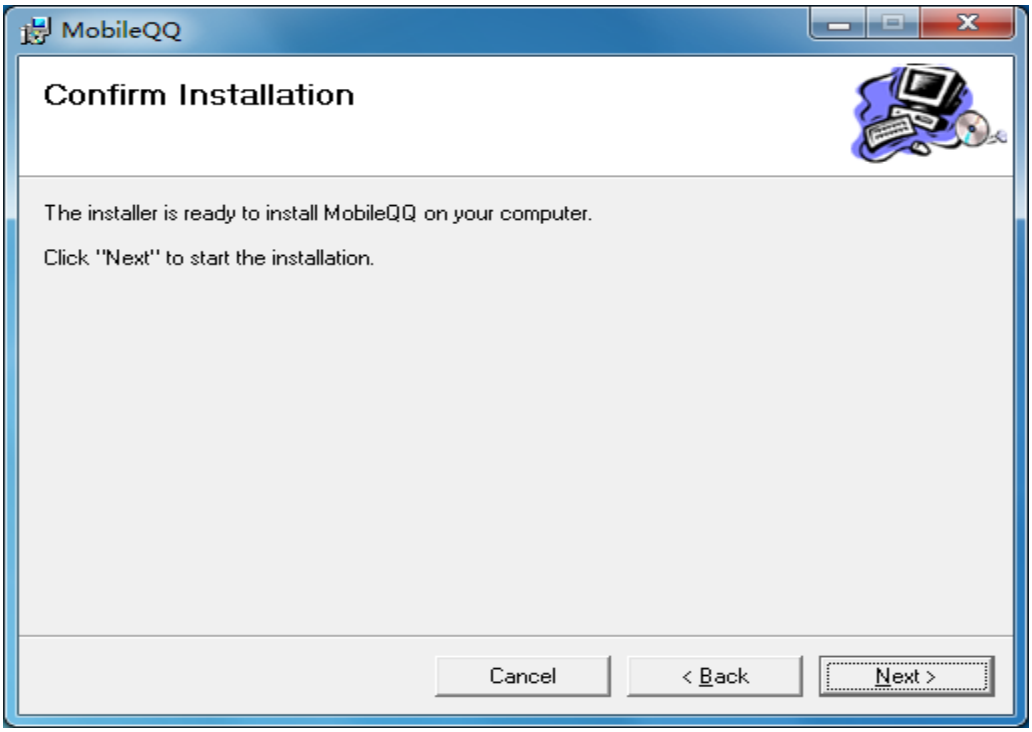
Step 3: Click on “Next” to continue



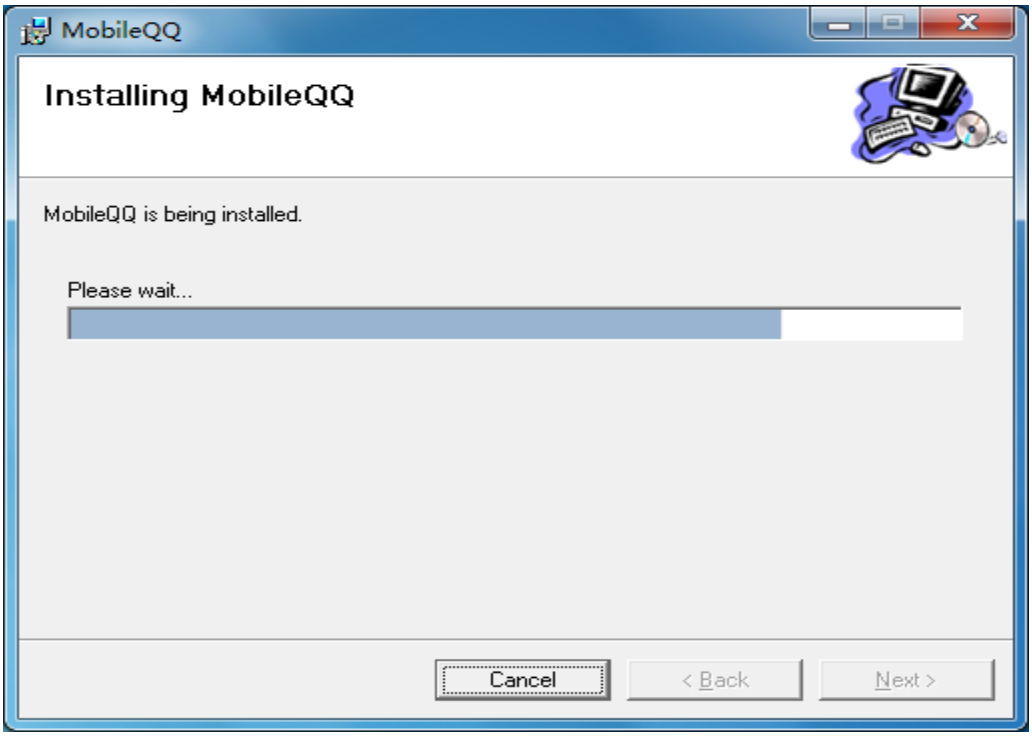
Step 4: Accept the agreement and click on “Next” to continue



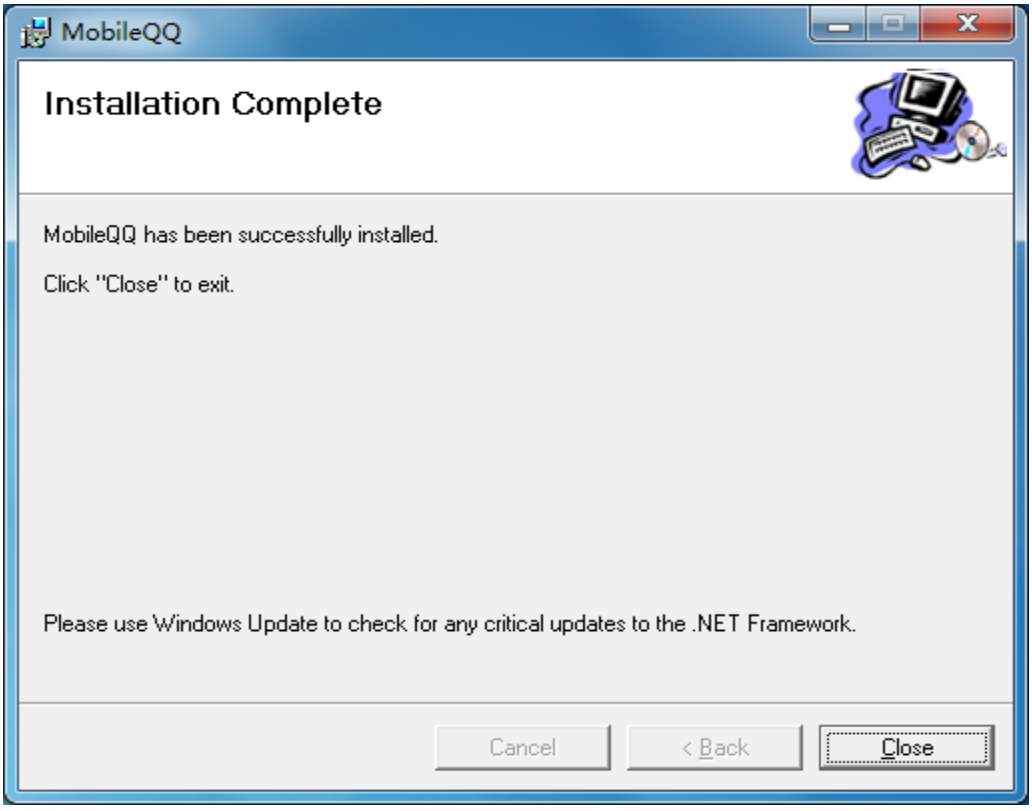
Step 5: click on “Next” to continue



Step 6: Wait a while



Step 7: click on “Close” to complete



10.1.4 Install BSP and Examples

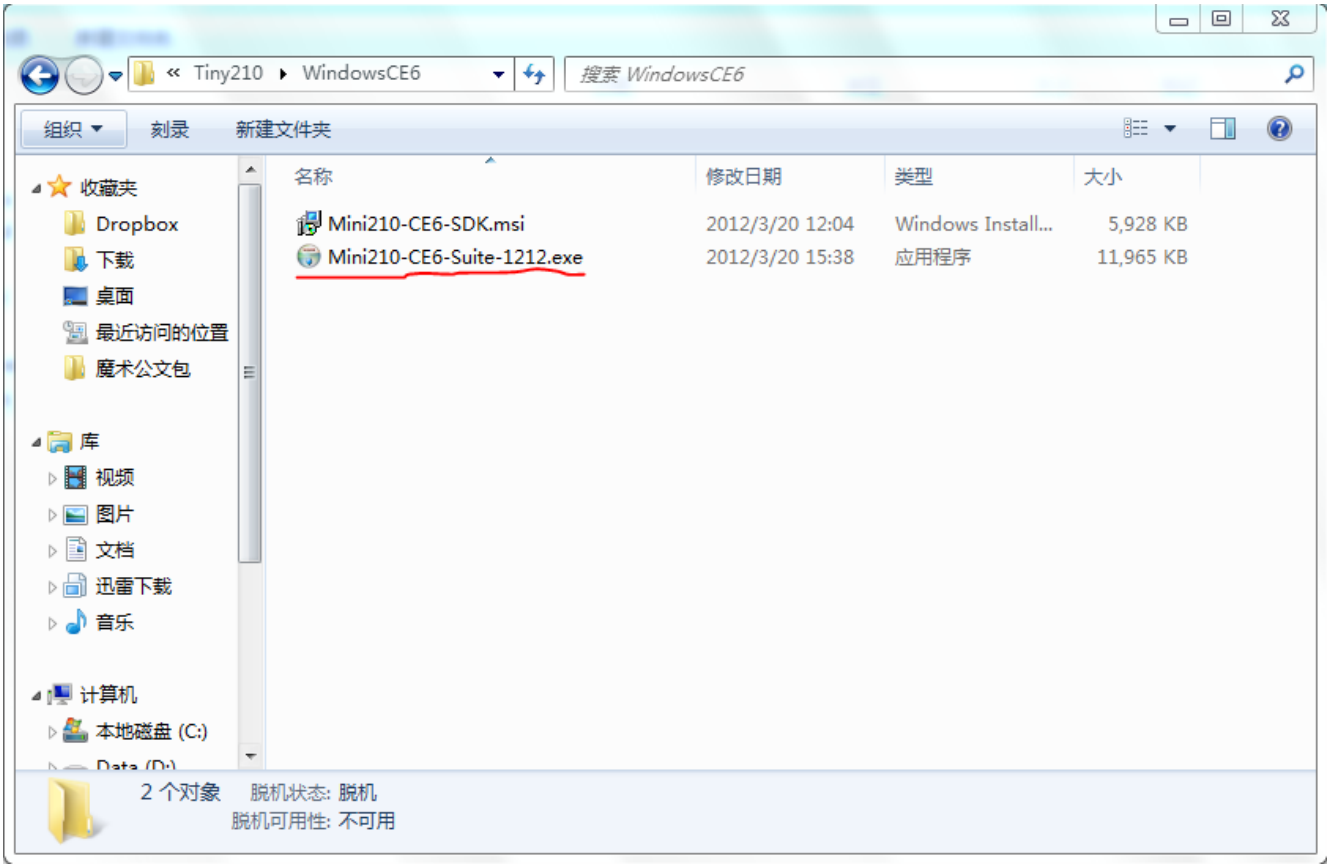
Note: the 6410's WinCE development environment cannot co-exist with the 210's. If your PC is installed with the Tiny6410's development environment you need to delete the following directory before you install the Tiny210 BSP

C:\WINCE600\PLATFORM\COMMON\SRC\SOC\S3C6410_SEC_V1

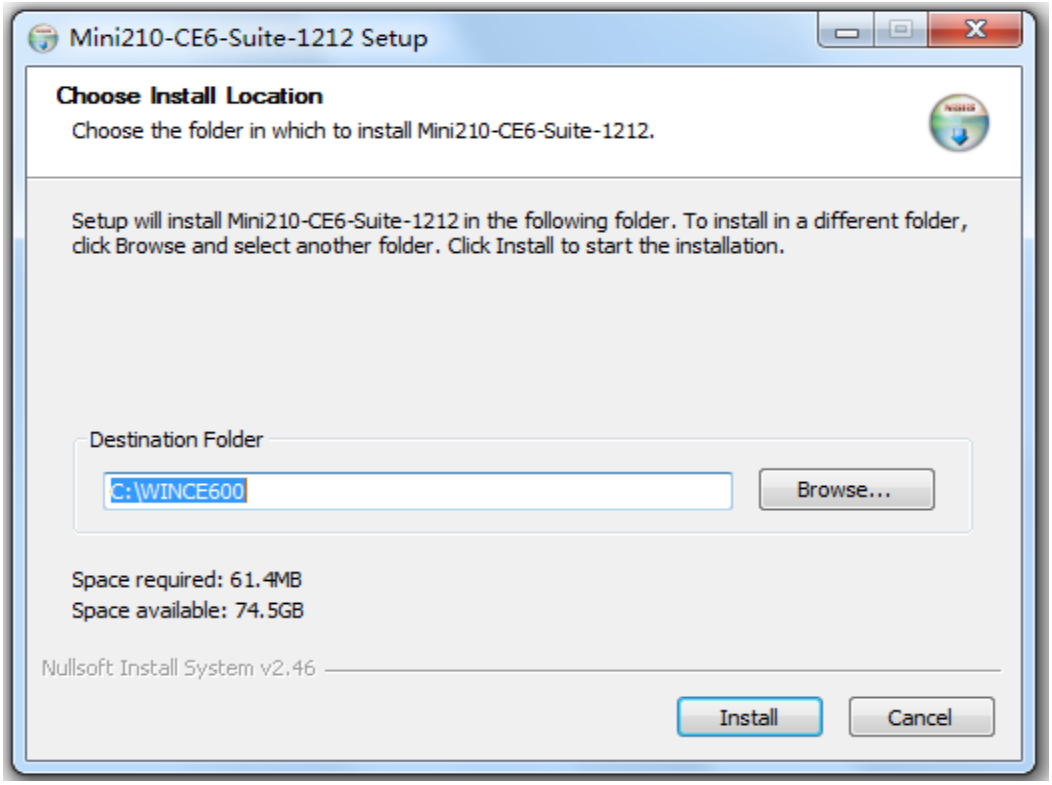
If you do want to install both the Tiny6410 and Tiny210's BSPs please try this way on VMWare

The Tiny210's BSP has only one installation file Mini210-CE6-Suite-1212. Belwo are the detailed installation steps:

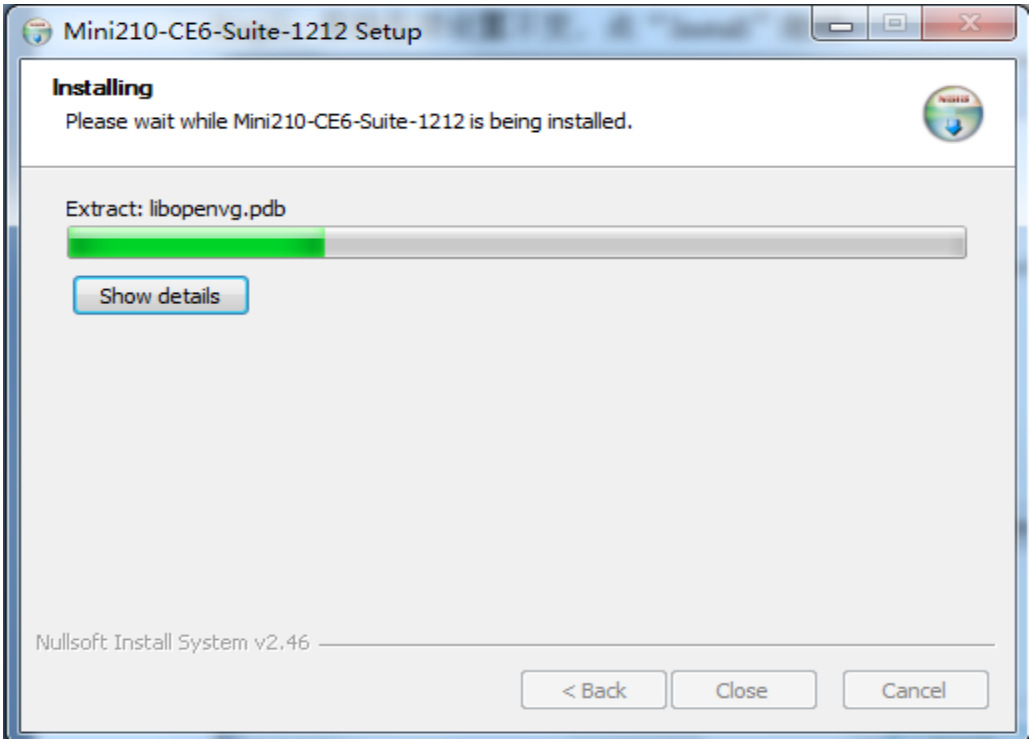
Step 1: double click on "Mini210-CE6-Suite-1212.exe"



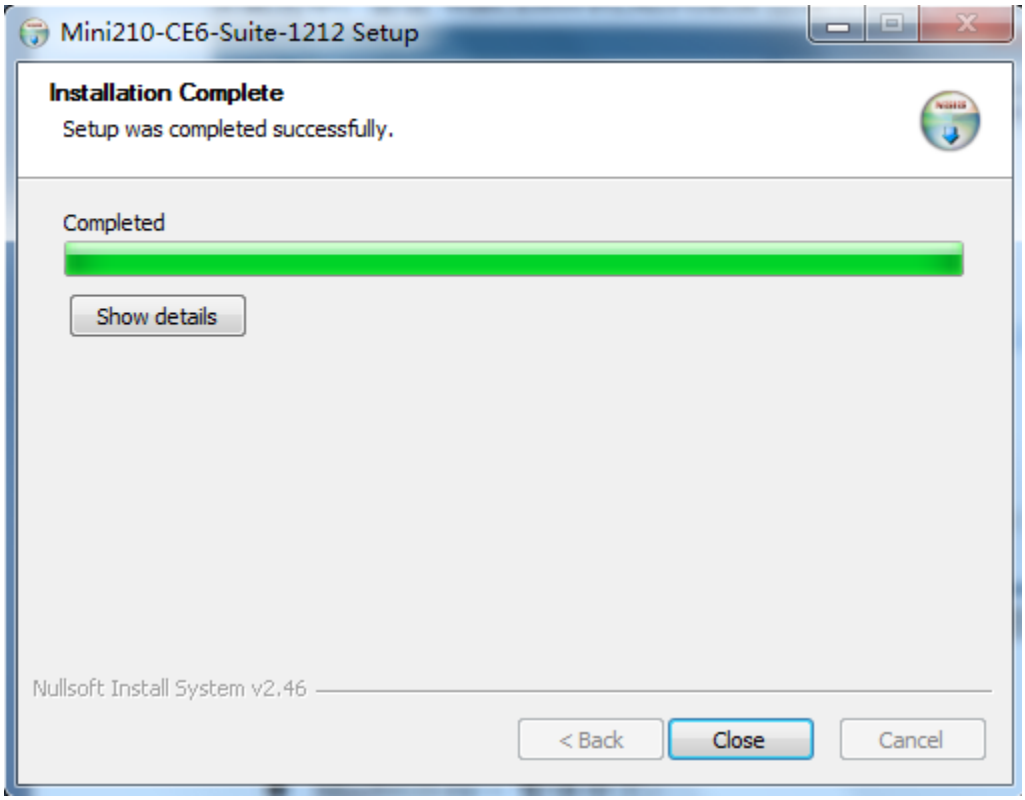
Step 2: click on “Install”



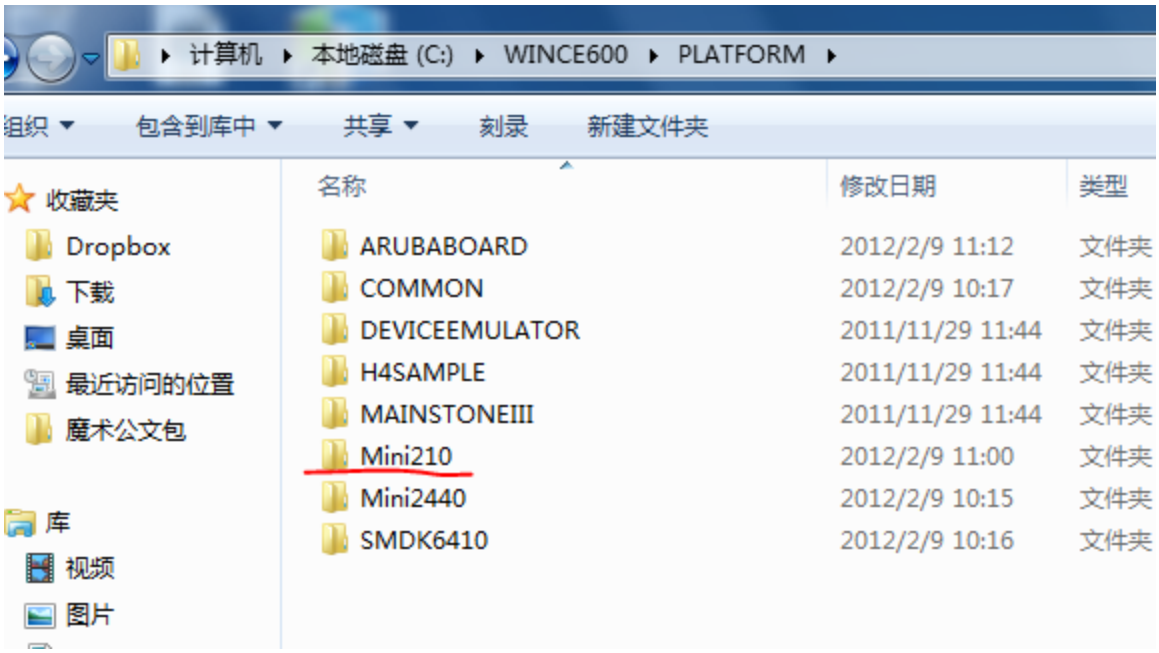
Step 3: Installation begins:



Step 4: After installation is done click on “Close” to complete



After installation is done a Mini210 BSP directory will be created under “WinCE600\PLATFORM”

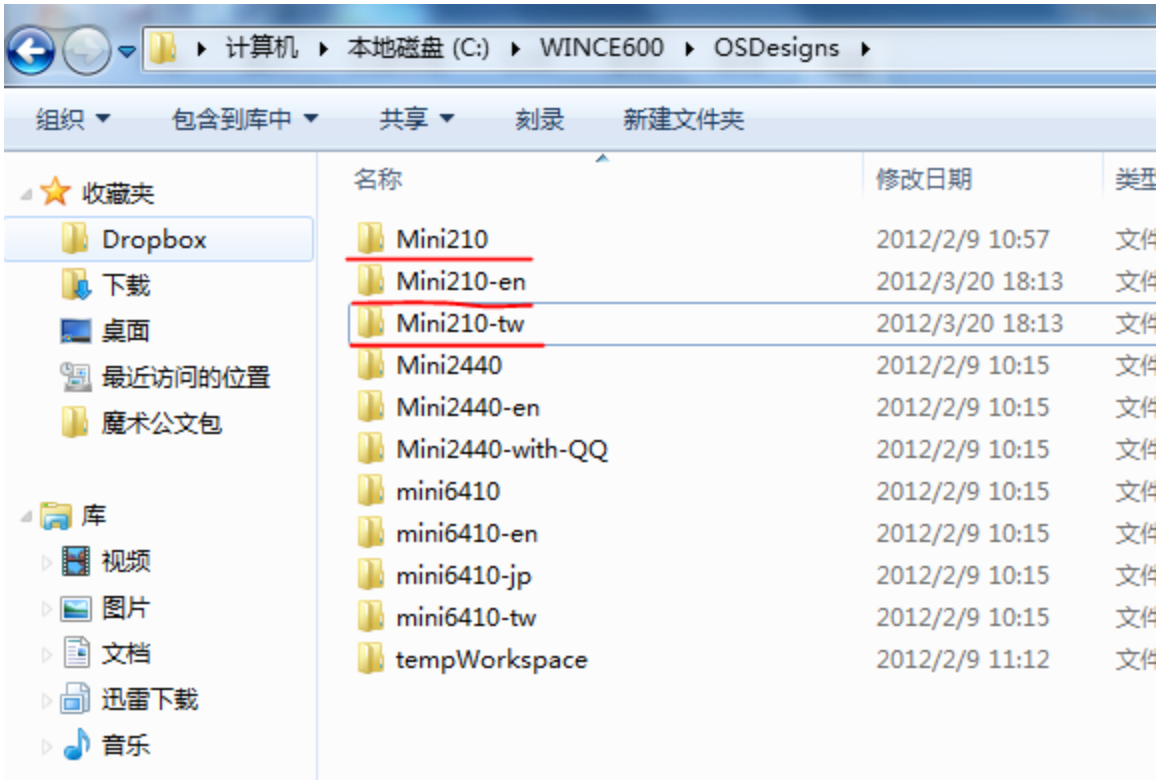


There are three directories in “WinCE600\OSDesigns”:

Mini210 – Simplified Chinese

Mini210-en – English

Mini210-tw – Traditional Chinese



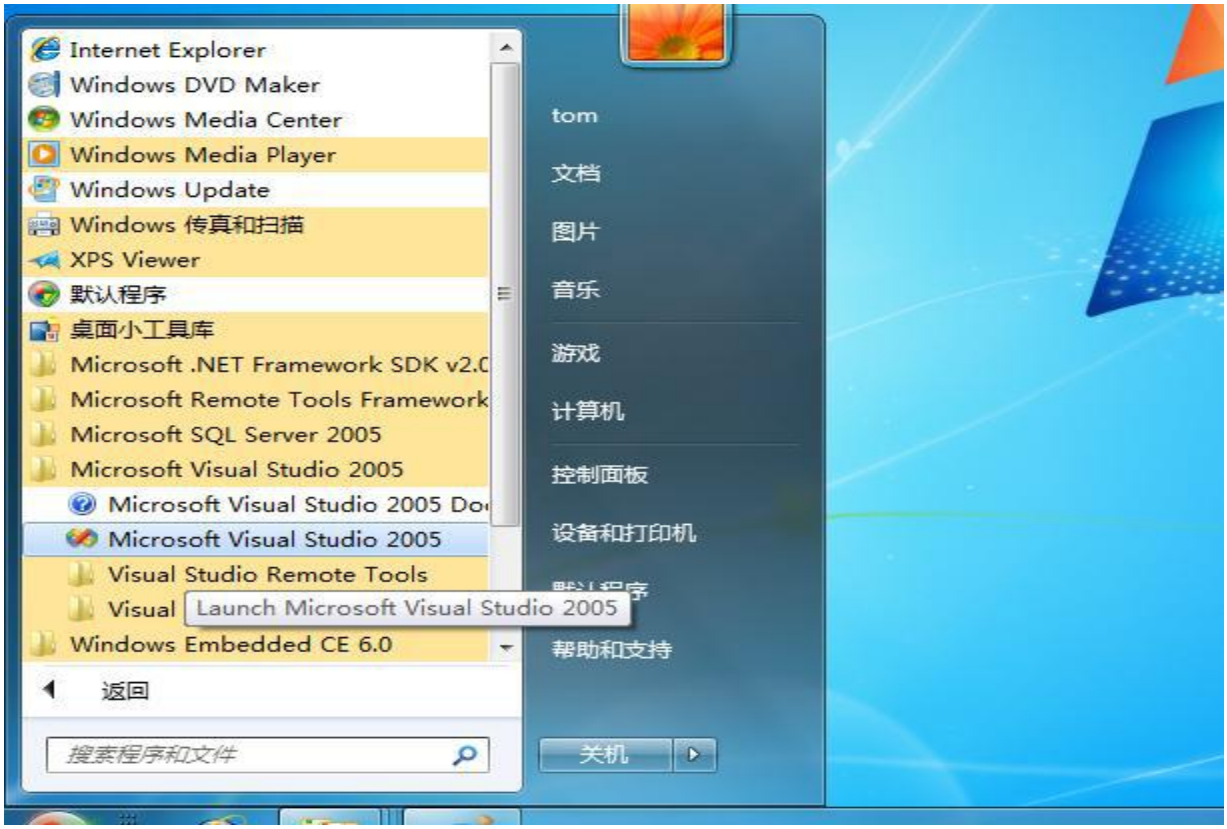
10.2 Compile WindowsCE 6 Kernel and Bootloader

It takes much time to compile a WinCE kernel therefore we provide a workable CE image in our DVD.

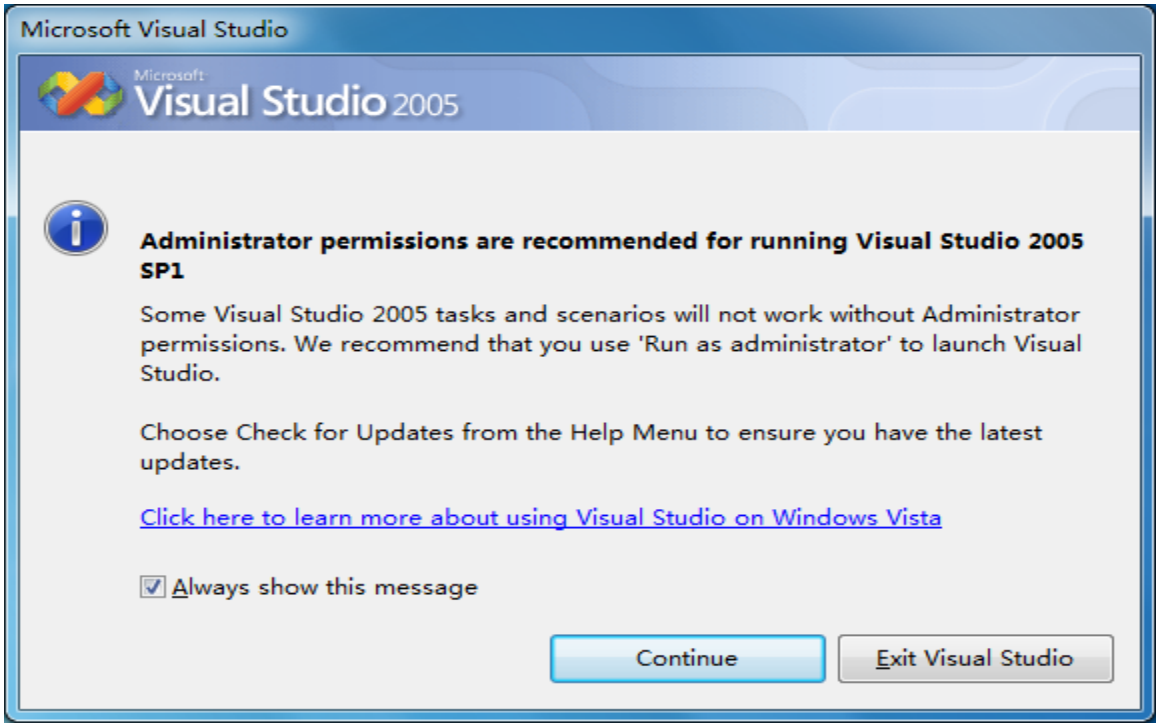
10.2.1 Compile Default Kernel Project

Now we will use VS2005 to compile our Mini210 BSP.

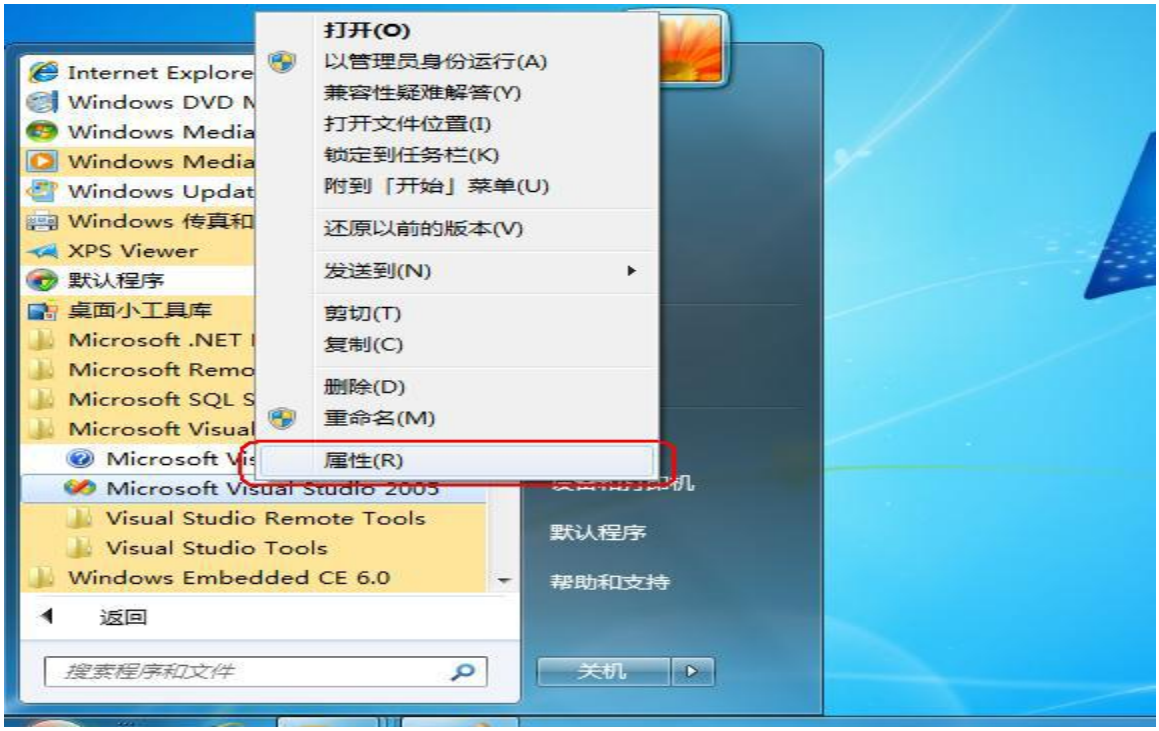
Step 1: go to “Start” -> “Programs” -> “Microsoft Visual Studio 2005” -> “Microsoft Visual Studio 2005” (abbreviated as VS2005)



Step 2: Click on “Exit Visual Studio”

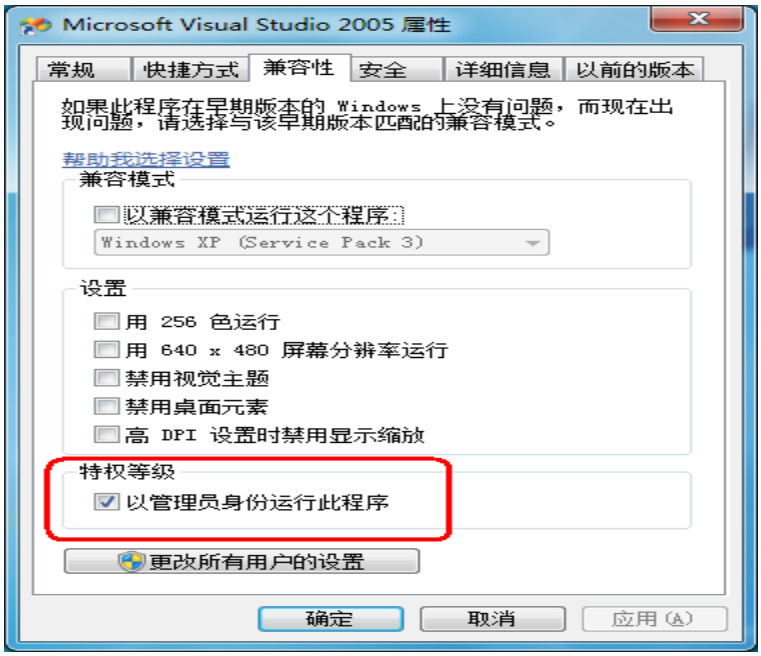


Step 3: Set up VS2005's property. Go to "Start" -> "Programs" -> "VS2005" -> "VS2005", right click and select "Property"



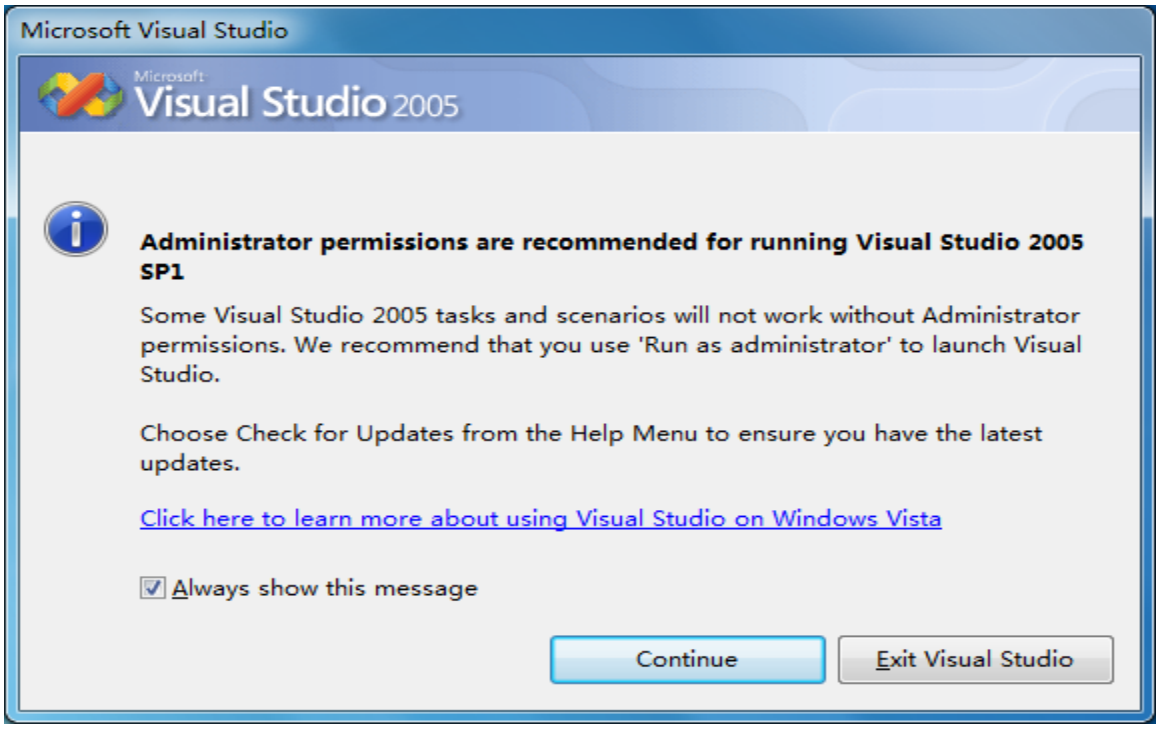
Step 4: On the pop up dialog, click on "Compatibility", check the following options and

save

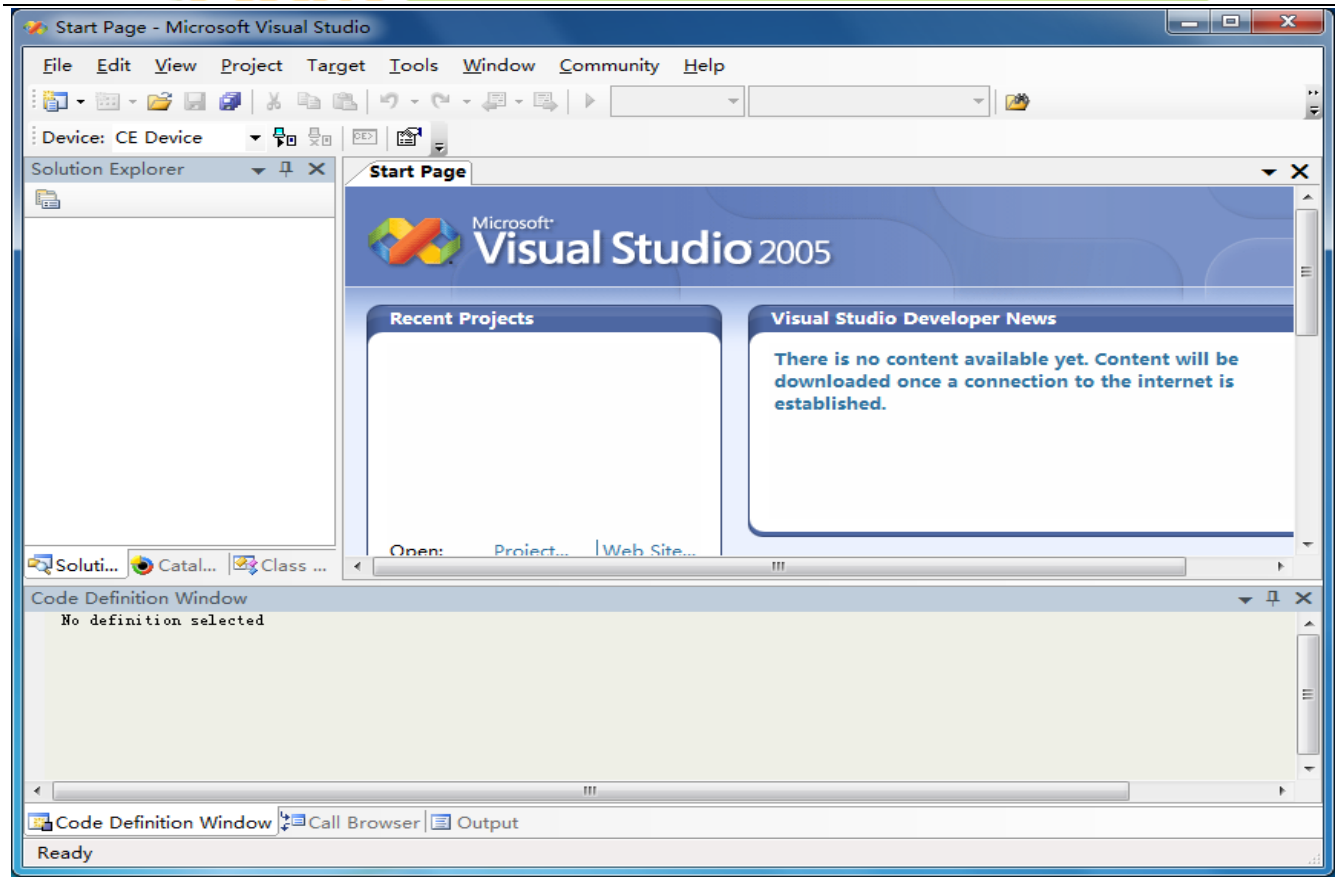


Step 5: go to “Start” -> “Programs” -> “Vs2005” -> “VS2005”, click on “Continue”.

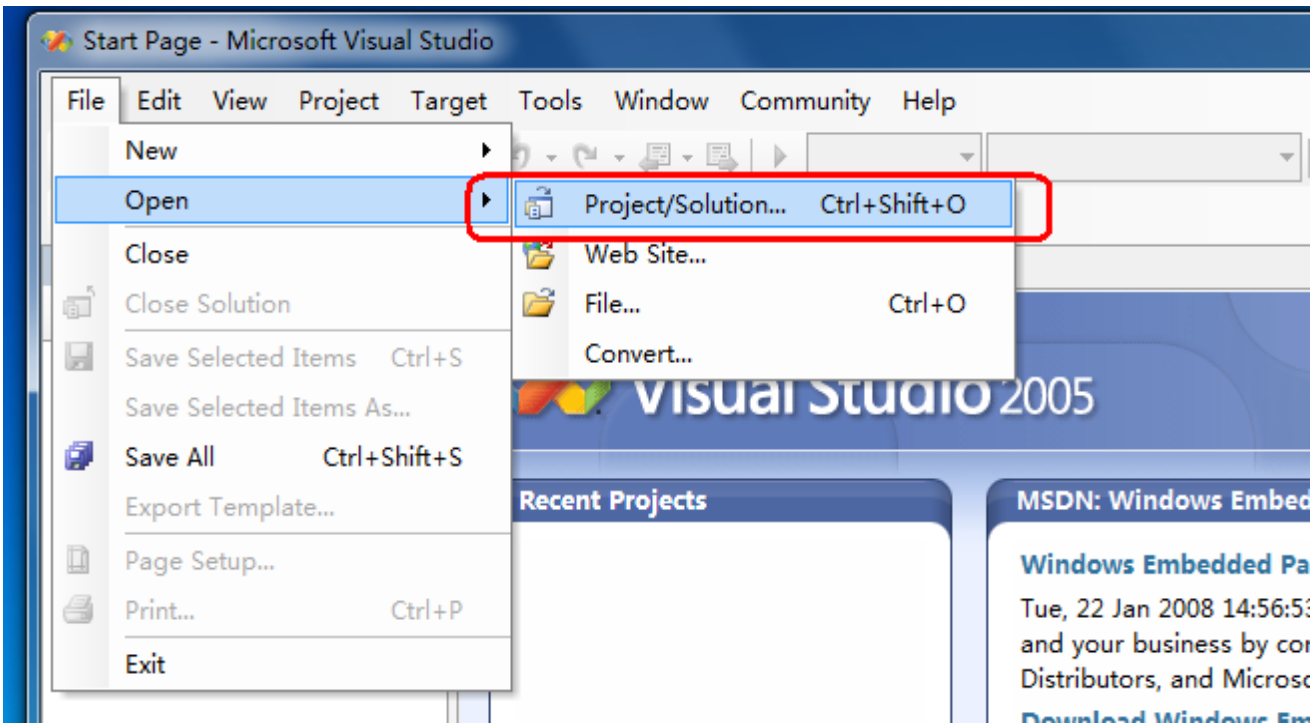
You will run VS2005 as administrator



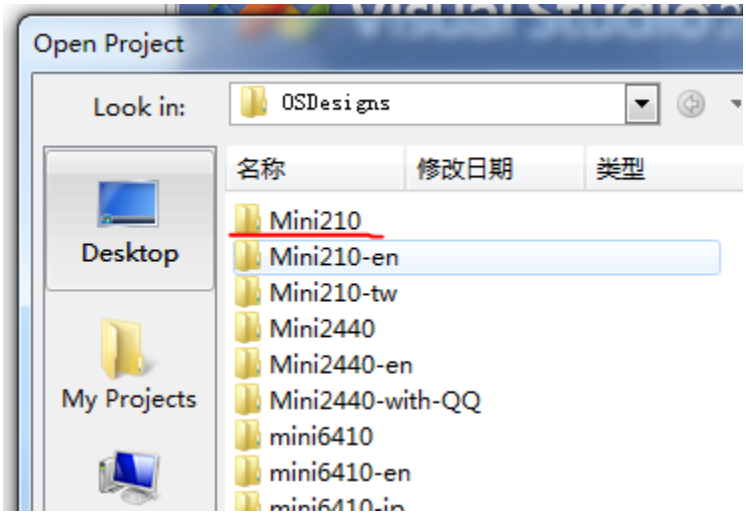
Step 6: you will see the following working window



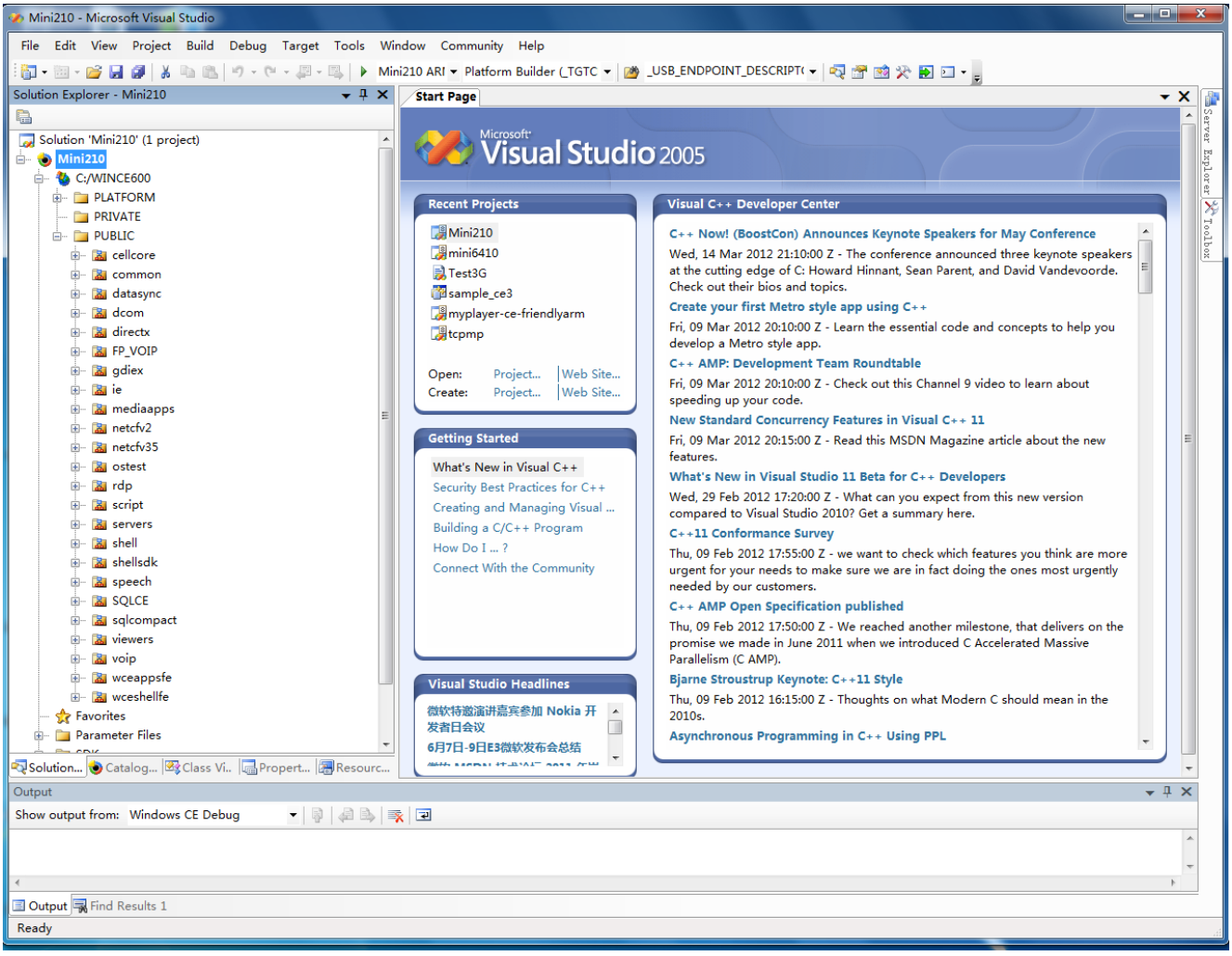
Step 7: go to “File” -> “Open” -> “Project/Solution…”



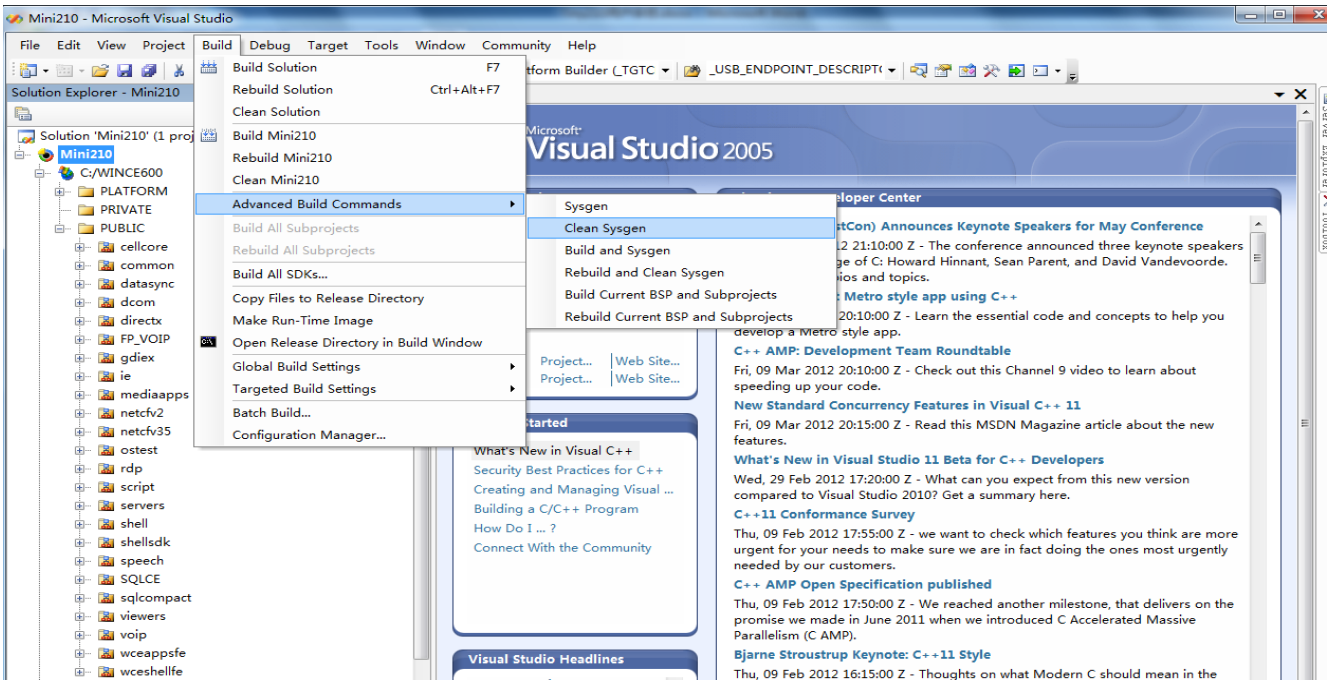
Step 8: Open the default Mini210 file "C:\WINCE600\OSDesigns\Mini210"



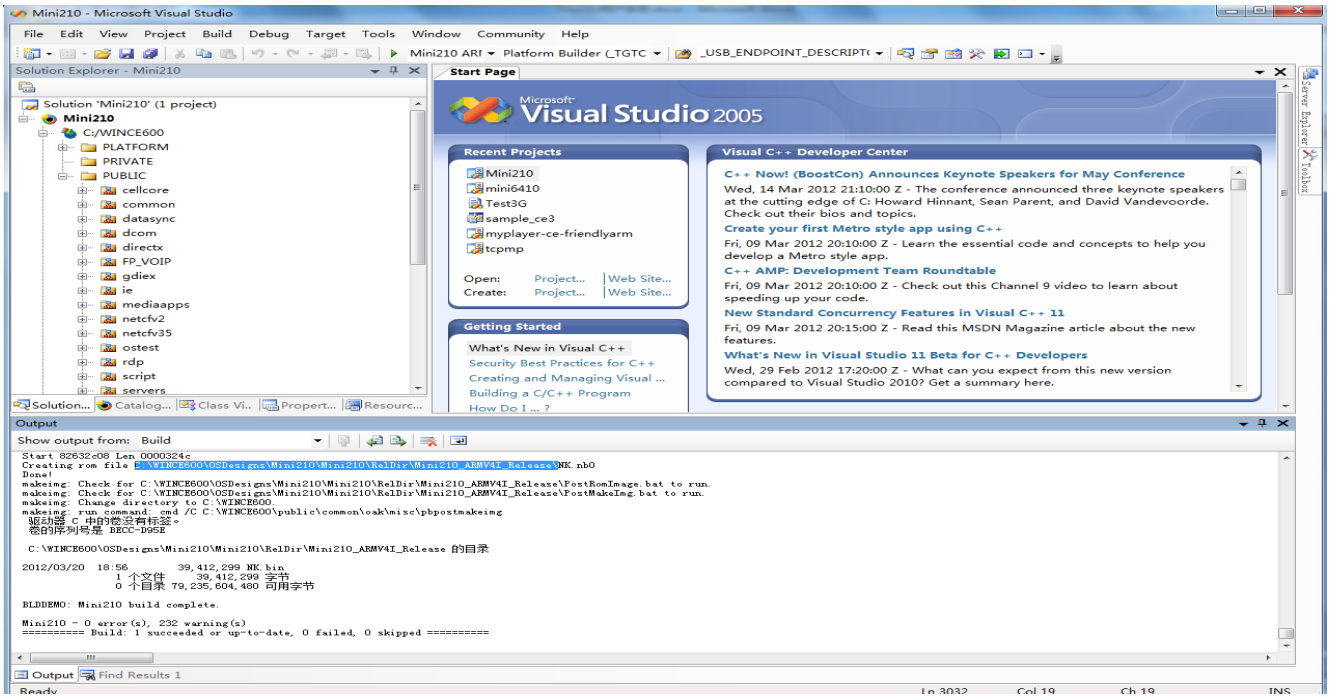
Step 9: After a while you will see the following dialog



Step 10: Go to “Build” -> “Advanced Build Commands” -> “Clean Sysgen” to compile the kernel.



Step 11: after compilation is done, an NK.bin and NK.nb0 will be created in “C:\WINCE600\OSDesigns\Mini210\Mini210\RelDir\Mini210_ARMV4I_Release\”



10.2.2 Change Serial Output

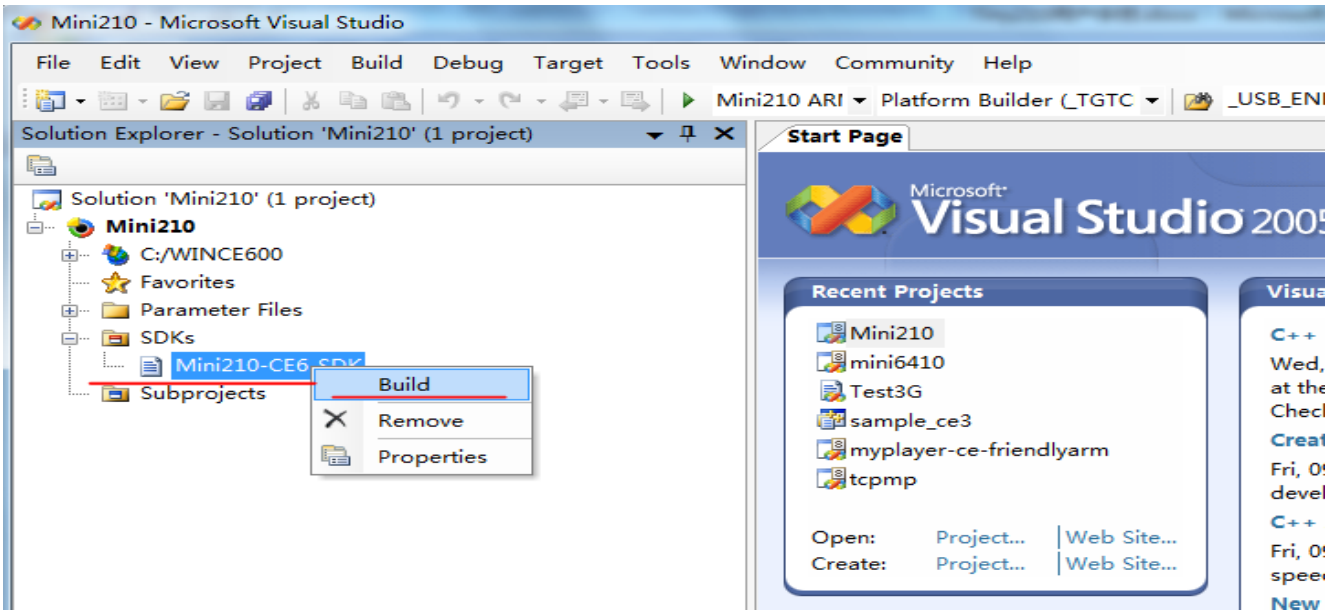
Open the options.h file you can define the serial ports' output:

#define DEBUG_PORT DEBUG_UART_NONE
The default definition doesn't output debug messages to any serial port. The DEBUG_PORTcan be set to the following values:
DEBUG_UART_NONE -> No output message
DEBUG_UART0 -> output to COM1
DEBUG_UART -> output to COM2
DEBUG_UART2 -> output to COM3
DEBUG_UART3 -> output to COM4

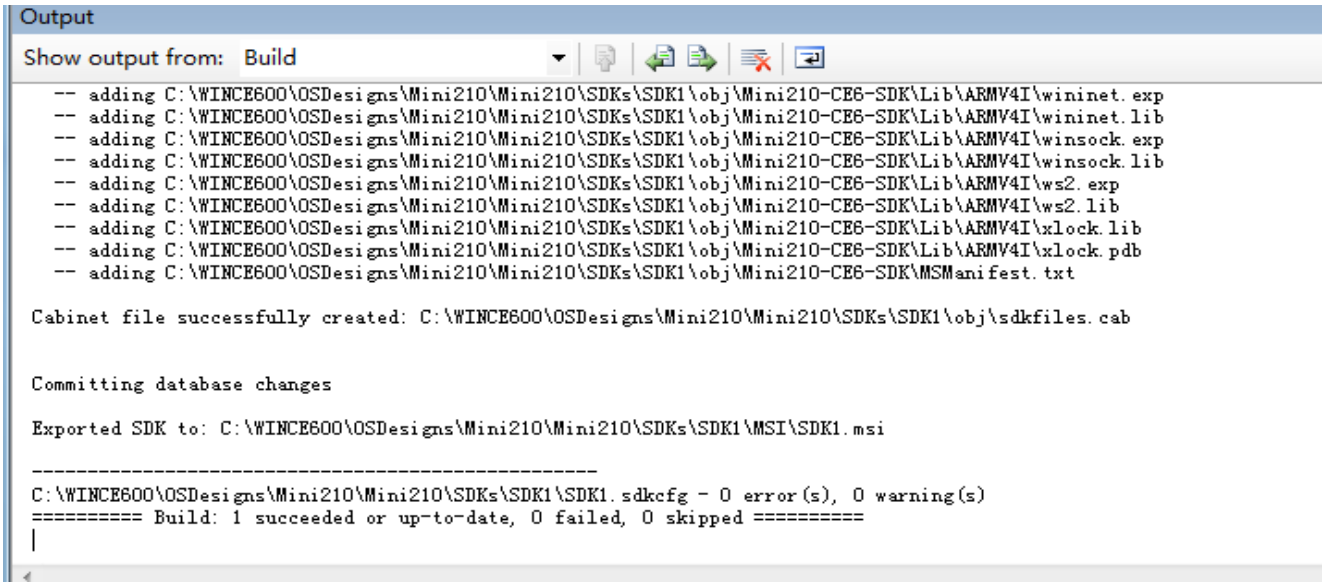
10.2.3 Create SDK

When your PC only has VS2005 but doesn't have Windows CE6 Platform Builder you still want to develop applications for the Tiny210 you need an SDK. After you compile a default kernel you can create an SDK via VS2005. Note: this SDK only applies to VS2005 and cannot work with EVC or VS2008. Here are the steps to create it:

Step1: start VS2005 and open the compiled project file Mini210, right click on "Mini210-CE6-SDK" and click on "Build"



Step 2: a while later an SDK will be created



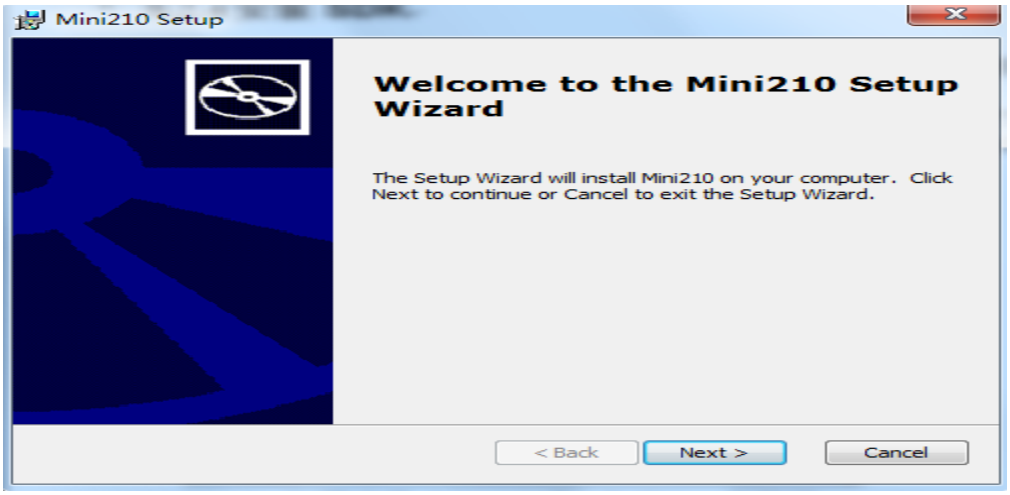
Step 3: an SDK1.msi will be created under "C:\WINCE600\OSDesigns\Mini210\Mini210\SDKs\SDK1\MSI"

10.2.4 Install SDK

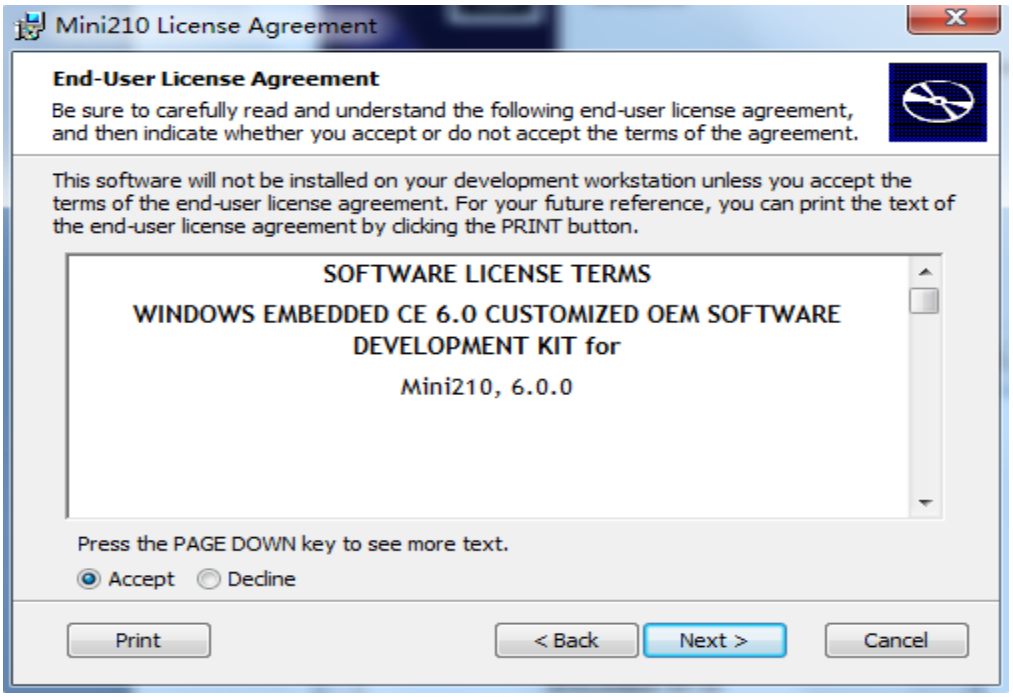
If you don't want to make your own SDK you can use the one included in the shipped

DVD. It is under "WindowsCE6\Mini210-CE6-SDK.msi". To develop applications for the Tiny210 in VS2005 you need to install an SDK. Here are the steps to install it.

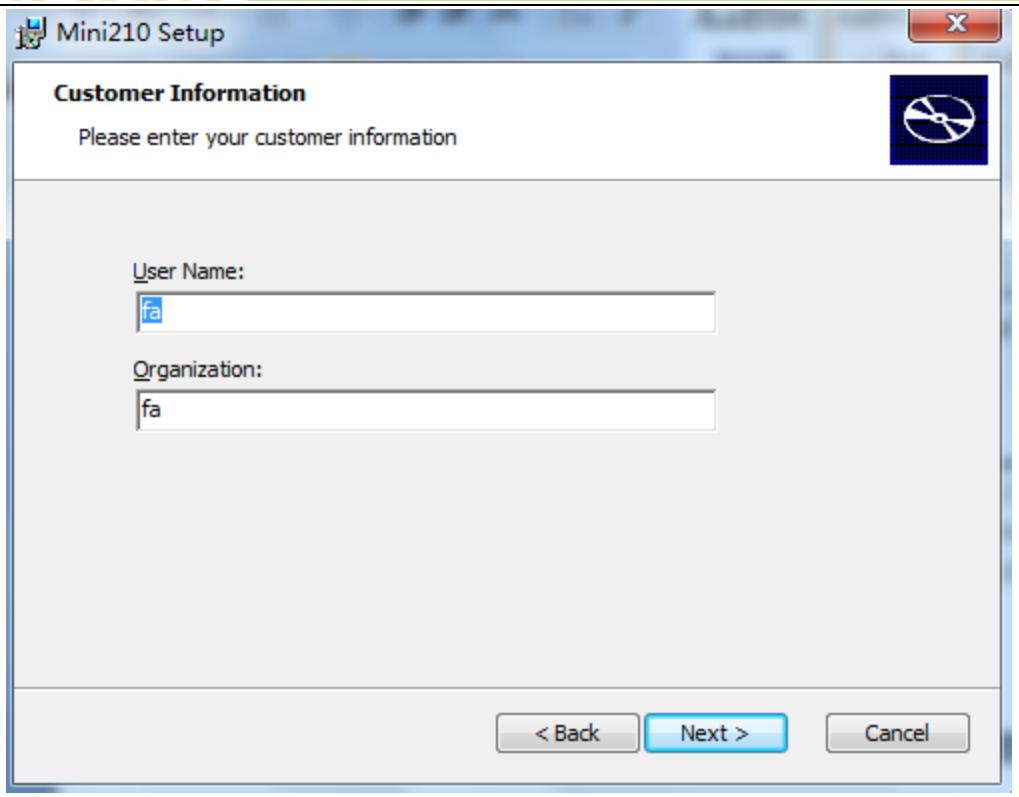
Step 1: double click on "Mini210-CE6-SDK.msi" and click on "Next"



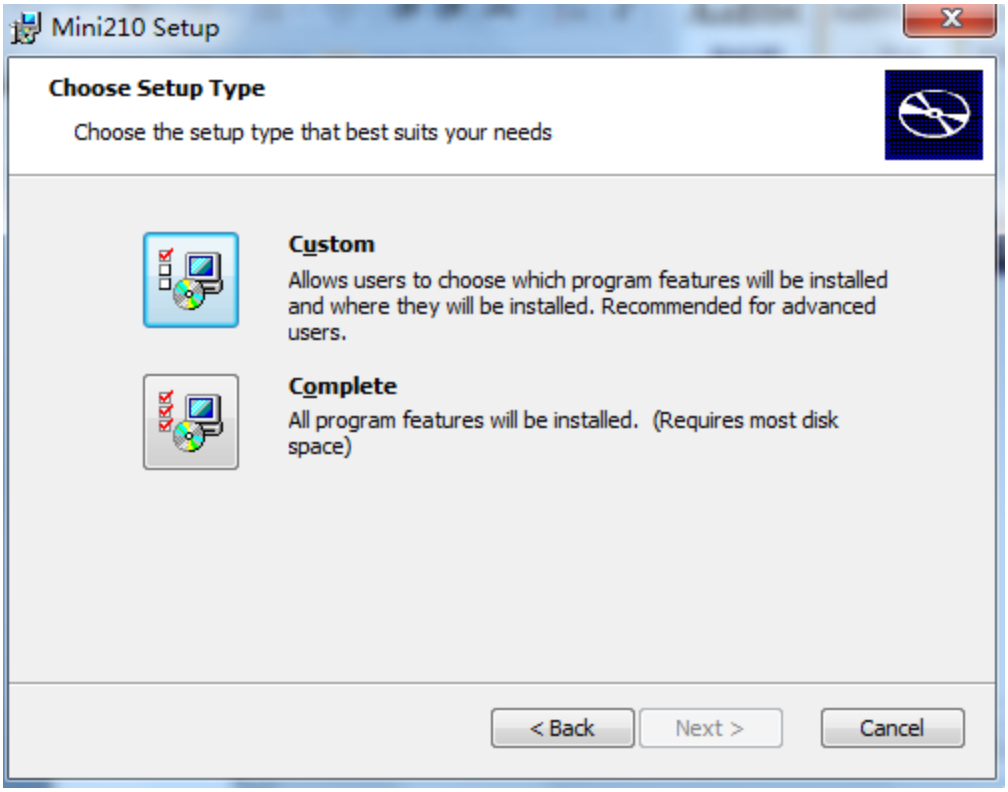
Step 2: accept the agreement and click on "Next"



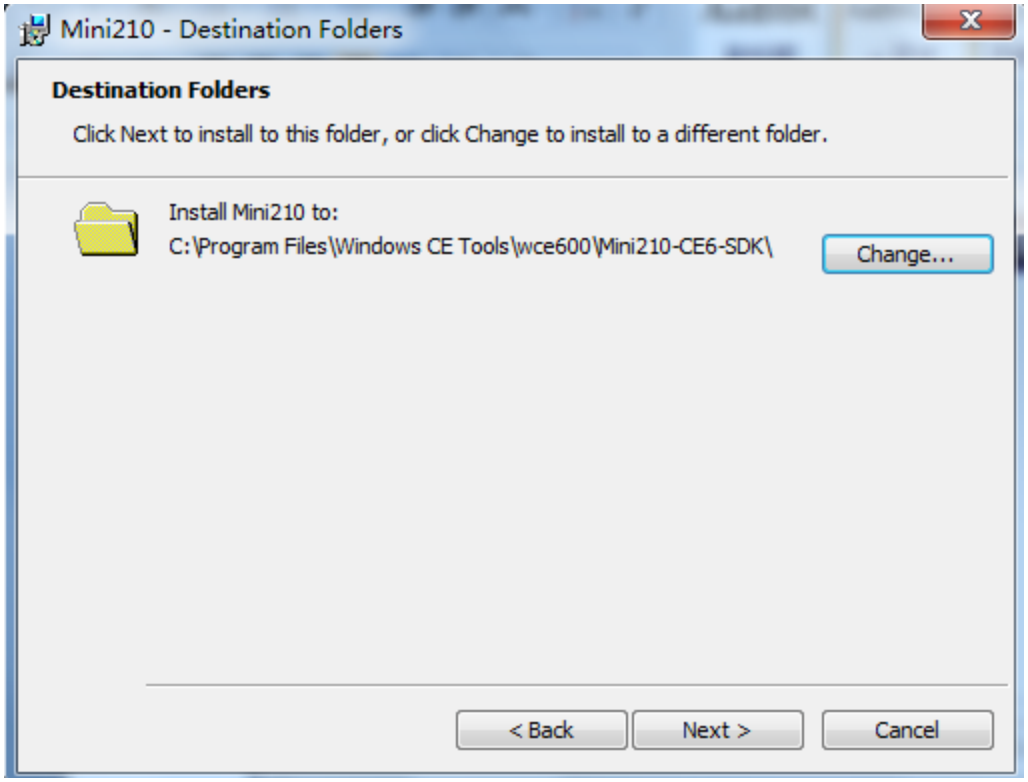
Step 3: type your company name and user name and click on "Next" to continue



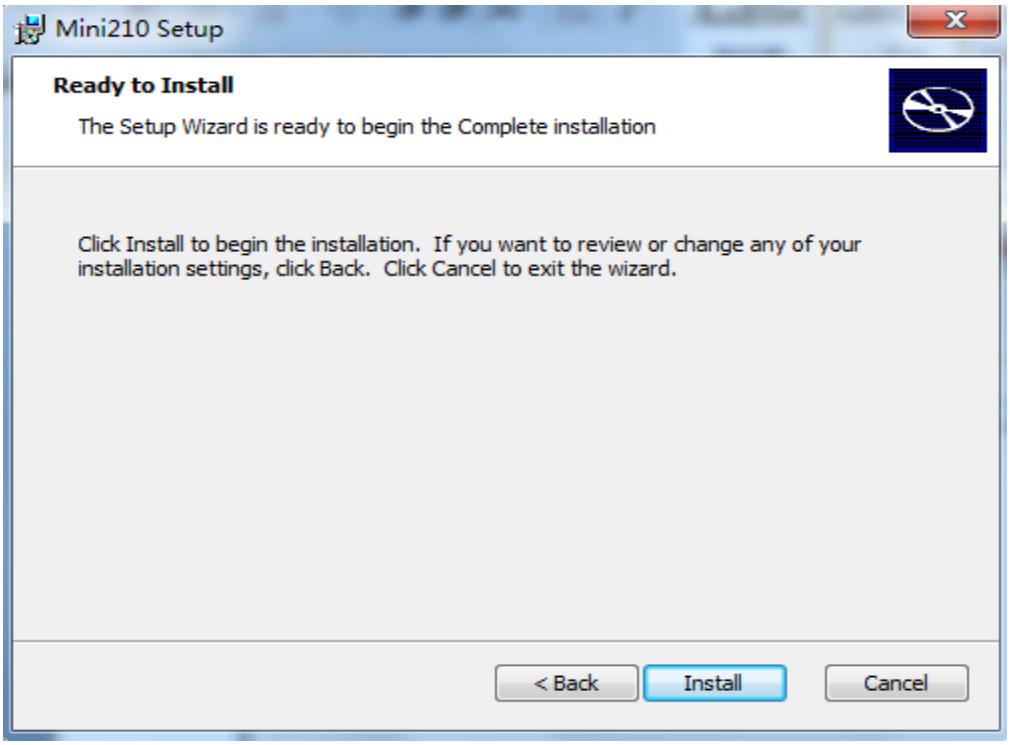
Step 4: Click on “Complete” to continue



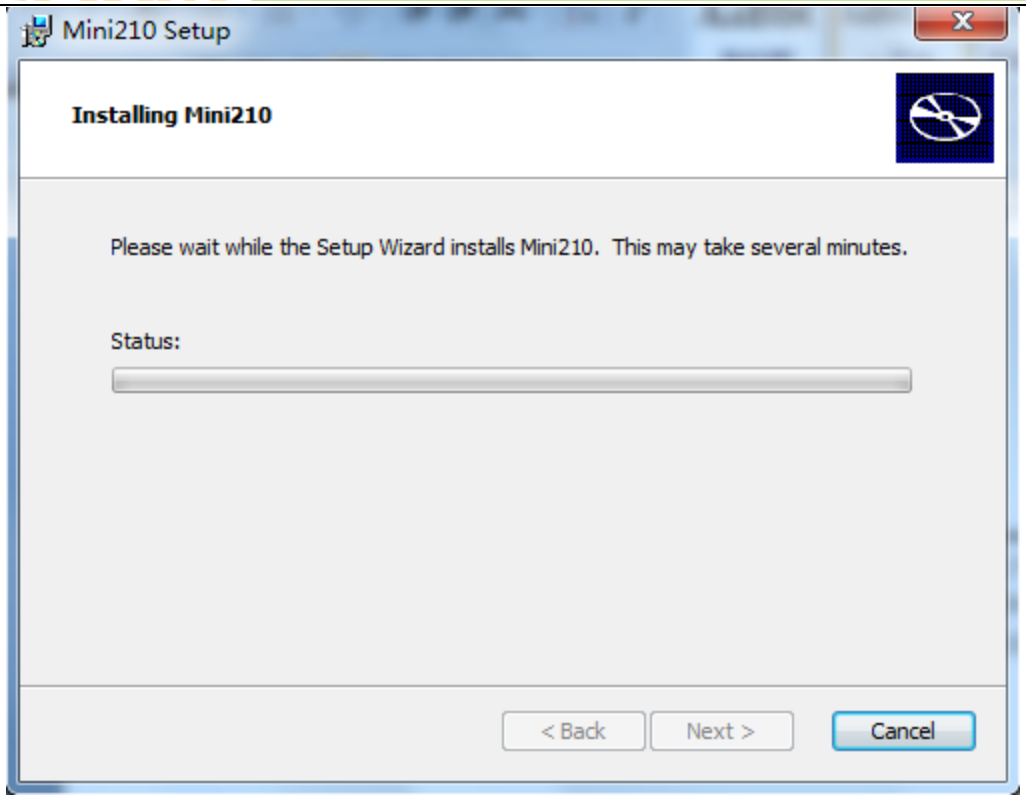
Step 5: click on “Next” to continue



Step 6: click on "Install" to continue



Step 7: Wait for installation



Step 8: After installation is done, click on “Finish” to complete