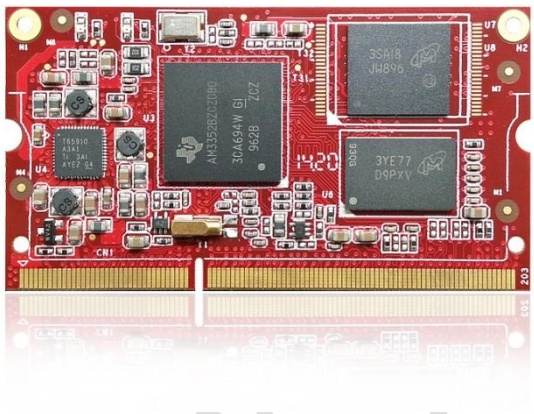
Linux U-Boot 中 GPIO 的控制 v1.0

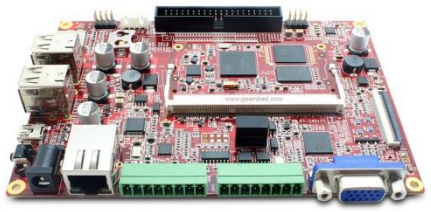
基于 TI AM335x 核心平台



免责声明

本文档是作者对 GOEMBED 产品进行实际操作和测试后，自我心得总结。建议读者具备一定的计算机基础和基本软件操作能力，如在操作过程中，遇到疑问和错误，欢迎加 QQ 群(462424566)交流，或发厂商技术支持邮箱进行咨询: support@goembed.com

操作环境配套说明：

硬件	详细介绍链接
SBC3358-B1A 单板机	
串口调试器：COM10U	

软件	详细介绍链接
Ubuntu 版本：12.04 LTS（64bit）	http://www.ubuntu.org.cn/download/desktop
Linux 版本：3.11.0-15-generic	
gcc 版本：4.6.3	

SBC3358-B1A 单板机软件特性

1、BootLoader 版本：u-boot-2013.01.01

2、内核版本：linux-3.2.0

- LCD 驱动
- LCD 背光驱动
- 电阻式触摸屏驱动
- VGA 驱动
- HSMMC/SD/MMC/SDIO 驱动
- IIC 驱动
- SPI 驱动
- 音频驱动
- DMA 驱动
- RTC 实时时钟驱动
- 电源管理
- USB HOST/DEVICE 驱动
- USB OTG 驱动
- DEBUG 驱动
- 以太网驱动
- TF 卡驱动
- CAN 驱动
- 串口驱动
- WG 驱动

3、交叉工具链：arm-linux-gnueabihf-gcc

SBC3358-B1A 单板机资源分配特性

1、emmc 空间分配

Partition	Size	Description
BootLoader	**200MB**	**FAT32 格式分区**
rootfs	**约 1500MB**	**EXT3 格式分区**

一、准备工作

1、参考《TI AM335x 搭建 Linux 开发环境 v1.0.docx》和《TI AM335x Linux 系统编译 v1.0.docx》把开发环境搭建好。

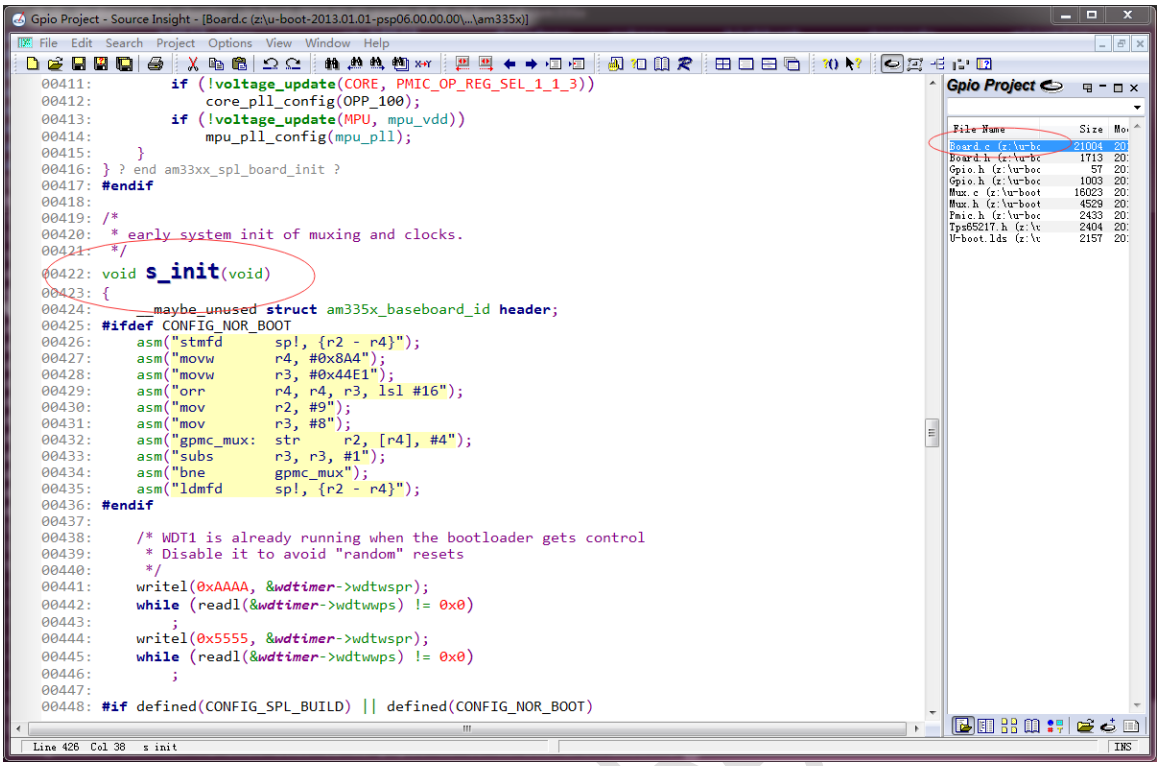
2、为了方便阅读和修改代码，在这里我使用的是 Source Insight(一个代码编辑工具)对代码进行修改。用户可以直接在终端使用 VI 编辑器编辑代码，结果是一样的，这里是为了阅读方便。

3、本文以控制一个 LED 为例子进行说明。

二、修改代码

1、s_init()函数分析

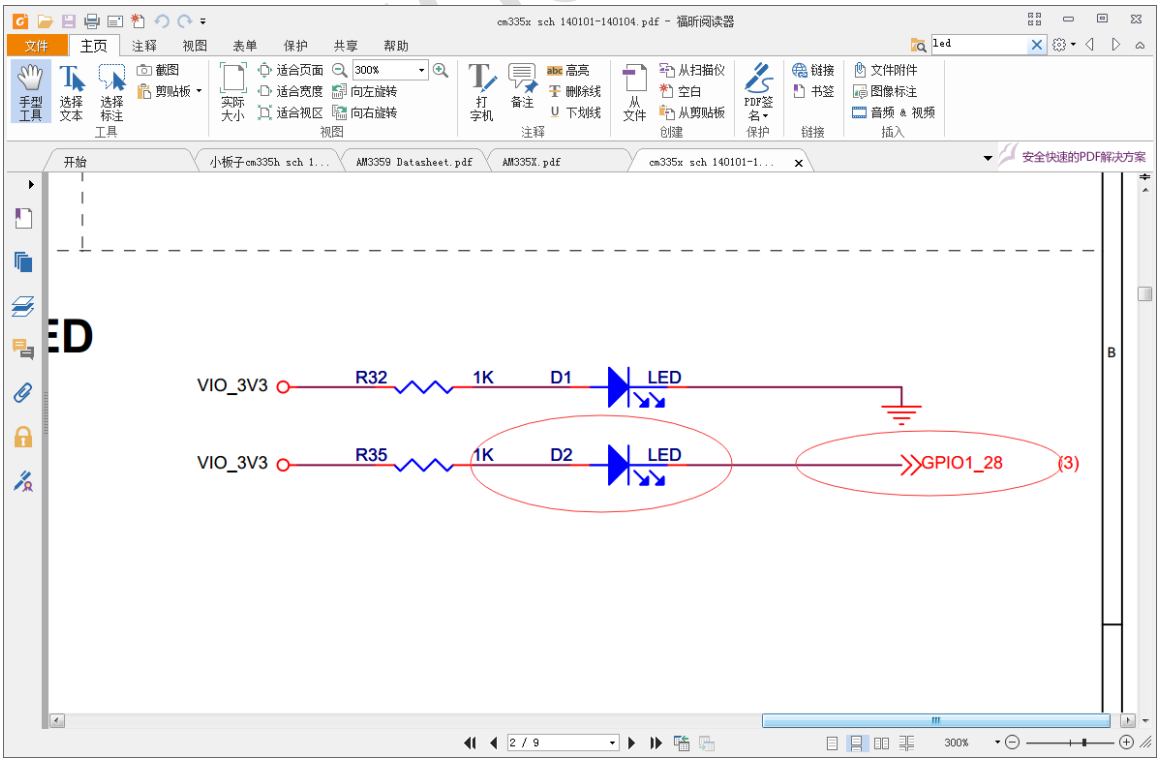
s_init()函数位于 u-boot-2013.01.01-psp06.00.00.00\board\ti\am335x\Board.c 中，这是一个很重要的函数，完成了时钟、串口等设置，当它调用 preloader_console_init()函数之后串口才能打印信息。下面我们在代码中通过添加打印信息来进行调试。s_init()函数如下图所示：



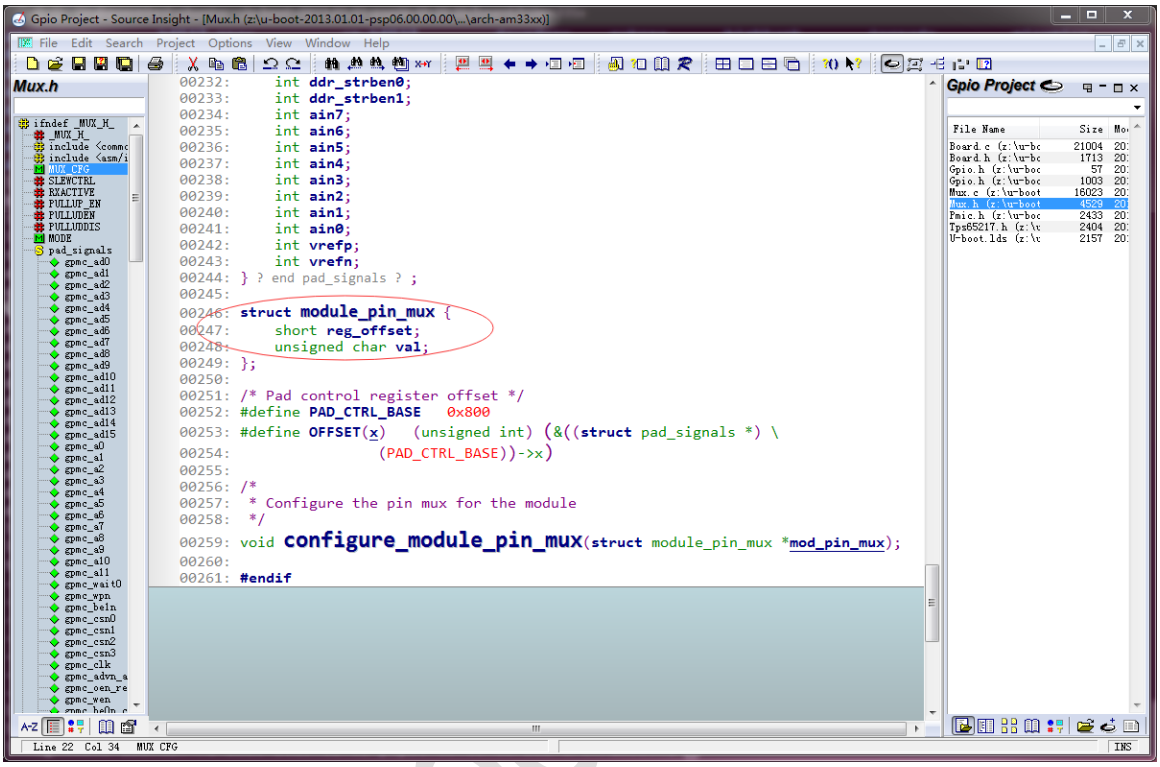

```
00411:          if (!voltage_update(CORE, PMIC_OP_REG_SEL_1_1_3))
00412:              core_pll_config(OPP_100);
00413:          if (!voltage_update(MPU, mpu_vdd))
00414:              mpu_pll_config(mpu_pll);
00415:      }
00416: } ? end am33xx_spl_board_init ?
00417: #endif
00418:
00419: /*
00420:  * early system init of muxing and clocks.
00421:  */
00422: void s_init(void)
00423: {
00424:      maybe_unused struct am335x_baseboard_id header;
00425: #ifdef CONFIG_NOR_BOOT
00426:     asm("stmfd      sp!, {r2 - r4}");
00427:     asm("movw       r4, #0x8A4");
00428:     asm("movw       r3, #0x44E1");
00429:     asm("orr        r4, r4, r3, lsl #16");
00430:     asm("mov        r2, #9");
00431:     asm("mov        r3, #8");
00432:     asm("gpmc_mux:  str     r2, [r4], #4");
00433:     asm("subs       r3, r3, #1");
00434:     asm("bne        gpmc_mux");
00435:     asm("ldmfd      sp!, {r2 - r4}");
00436: #endif
00437:
00438:     /* WDT1 is already running when the bootloader gets control
00439:      * Disable it to avoid "random" resets
00440:      */
00441:     writel(0xAAAA, &wdtimer->wdtwspr);
00442:     while (readl(&wdtimer->wdtwwps) != 0x0)
00443:         ;
00444:     writel(0x5555, &wdtimer->wdtwspr);
00445:     while (readl(&wdtimer->wdtwwps) != 0x0)
00446:         ;
00447:
00448: #if defined(CONFIG_SPL_BUILD) || defined(CONFIG_NOR_BOOT)
```

2、配置 pinmux

在原理图中可以看到，GPIO1_28 连接 的是一个 LED 灯，我们使用该 GPIO：



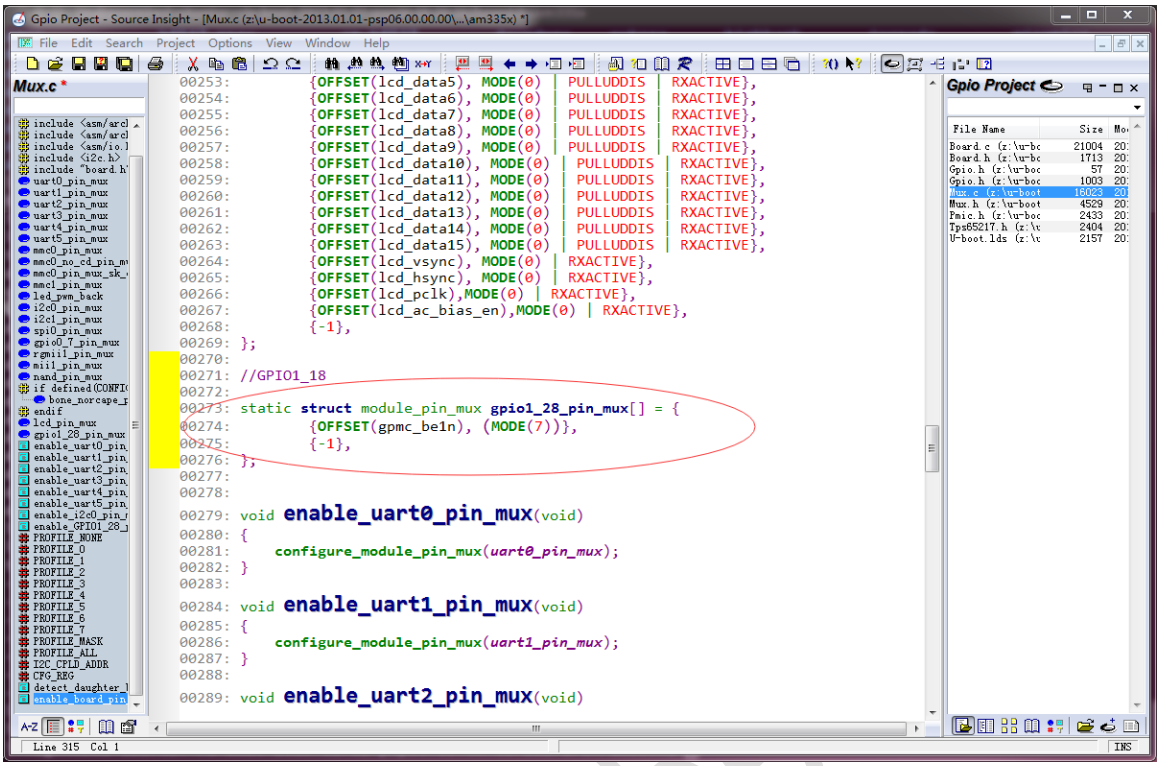
（1）新增一个 struct module_pin_mux 结构体变量，该结构体位于 u-boot-2013.01.01-psp06.00.00.00\arch\arm\include\asm\arch-am33xx\Mux.h 中：



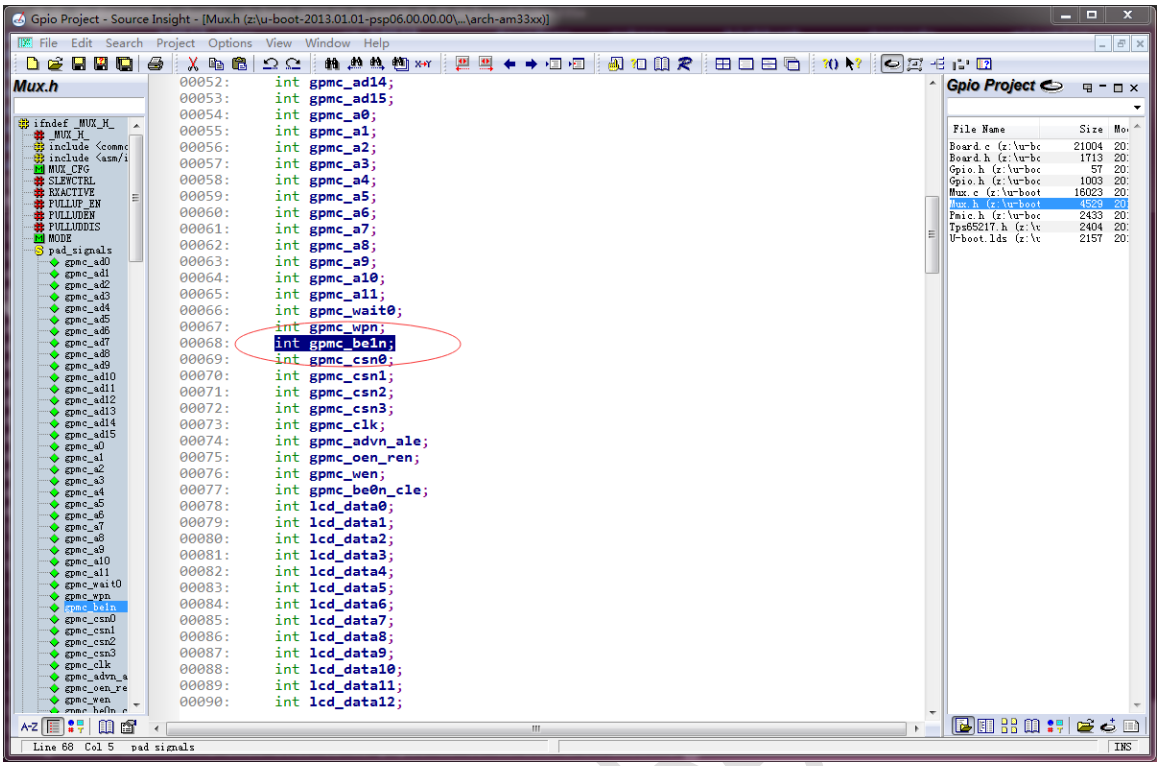
如果要在 U-Boot 中新增一个 GPIO 口，则需要在 u-boot-2013.01.01-psp06.00.00.00\board\ti\am335x\Mux.c 中新增一个对应的结构体：

比如在这里新增 GPIO1_28，则新增的结构体如下：

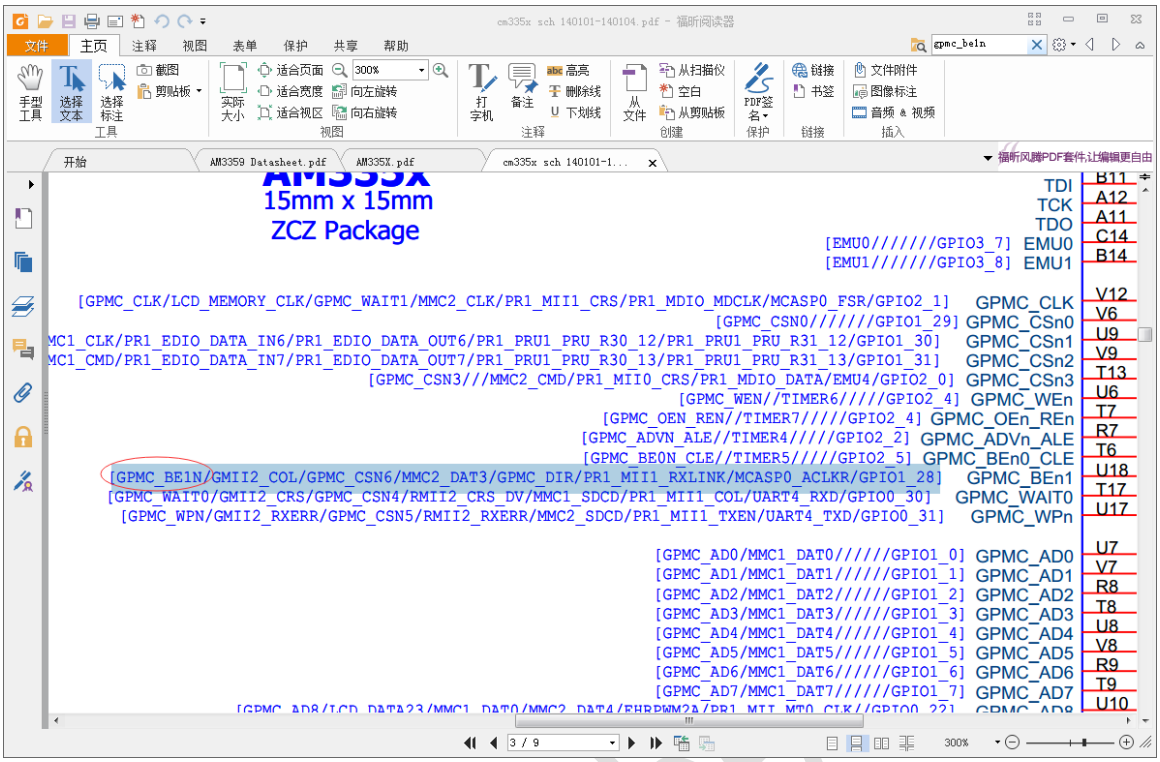
```
static struct module_pin_mux gpio1_28_pin_mux[] = {
        {OFFSET(gpmc_beln), (MODE(7))},
        {-1},
};
```



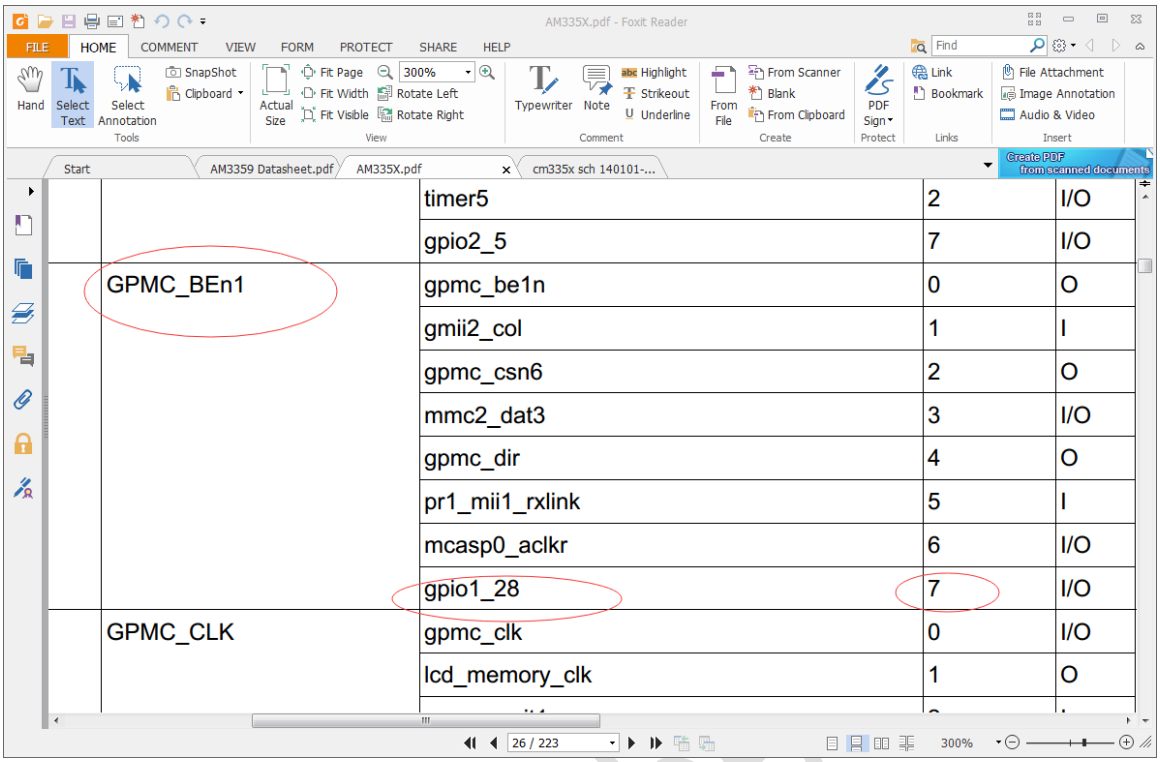
需要注意的是，OFFSET(gpmc_be1n)中的 gpmc_be1n 对应的是该引脚的第一功能，在 struct pad_signals 结构体中定义，该结构体位于 \u-boot-2013.01.01-psp06.00.00.00\am335x\include\asm\arch\Mux.h 中：



如果使用别的引脚，一定要记得把 OFFSET(x)中的参数更换为对应的第一功能，比如 GPIO3_8 则应改为 OFFSET（emu1）。那么，怎么知道引脚的第一功能是什么呢？我们可以查看原理图，我们知道很多引脚都有复用功能，在原理图上该引脚的多种功能都会列出，最前面的就是第一功能。



第二个问题，module_pin_mux 结构体中的 MODE（x）是怎么确定的呢？答案在 AM335X.pdf 技术手册和数据手册可以看出，每个管脚有 8 种模式，即 MODE0-MODE7，其中 MOED0 是主模式。在 AM335x.pdf 中搜索“gpmc_be1n”可以看到，MODE7 对应的是 gpio1_28 模式。所以可以确定 MODE 为 7。其他引脚方法类似。

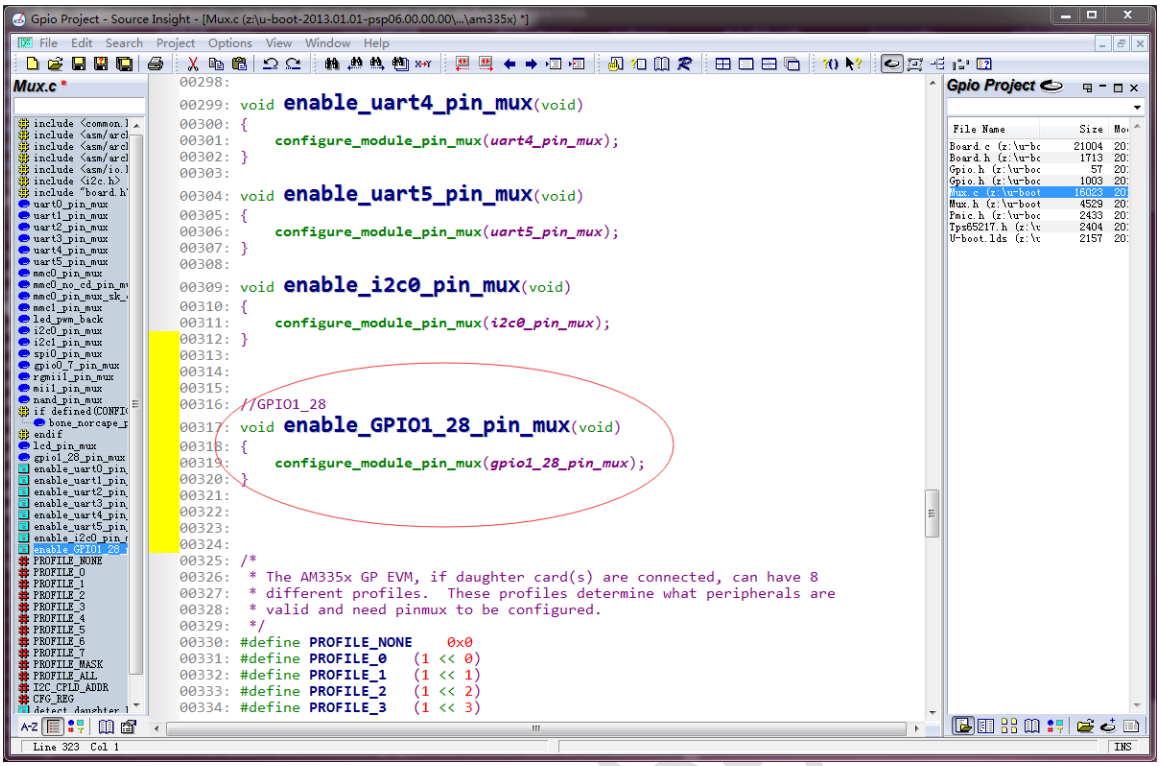


		timer5	2	I/O
		gpio2_5	7	I/O
	GPMC_BEn1	gpmc_be1n	0	O
		gmii2_col	1	I
		gpmc_csn6	2	O
		mmc2_dat3	3	I/O
		gpmc_dir	4	O
		pr1_mii1_rxlink	5	I
		mcasp0_aclkr	6	I/O
		gpio1_28	7	I/O
	GPMC_CLK	gpmc_clk	0	I/O
		lcd_memory_clk	1	O

最后，结构体 module_pin_mux 最后的“-1”作为结束的判断标记。

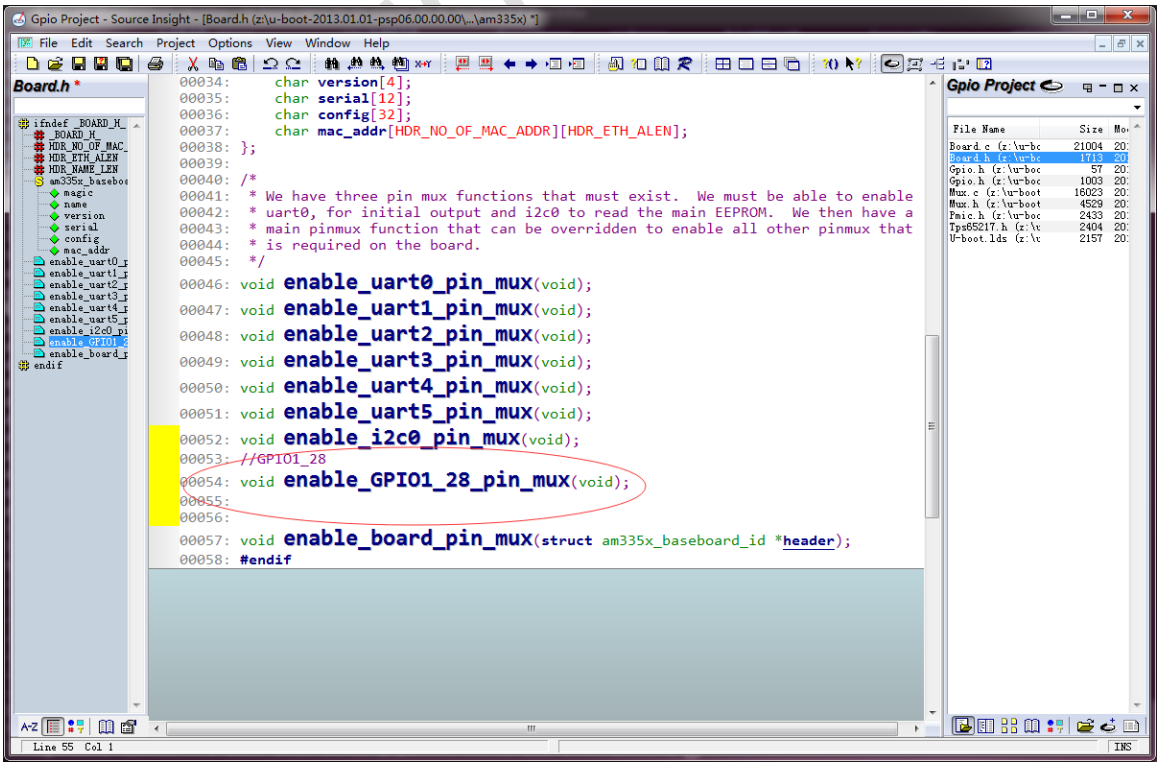
（2）、调用 void configure_module_pin_mux(struct module_pin_mux *mod_pin_mux);该函数位于 u-boot-2013.01.01-psp06.00.00.00\arch\arm\include\asm\arch-am33xx\Mux.h 中，我们需要调用它并传入新增的结构体变量 gpio1_28_pin_mux 对 GPIO1_28 进行配置。在这里有几个地方需要改，在 u-boot-2013.01.01-psp06.00.00.00\board\ti\am335x\Mux.c 中新增

```
void enable_GPIO1_28_pin_mux(void)
{
    configure_module_pin_mux(gpio1_28_pin_mux);
}
```



在 u-boot-2013.01.01-psp06.00.00.00\board\ti\am335x\Board.h 中新增函数声明，即添加：

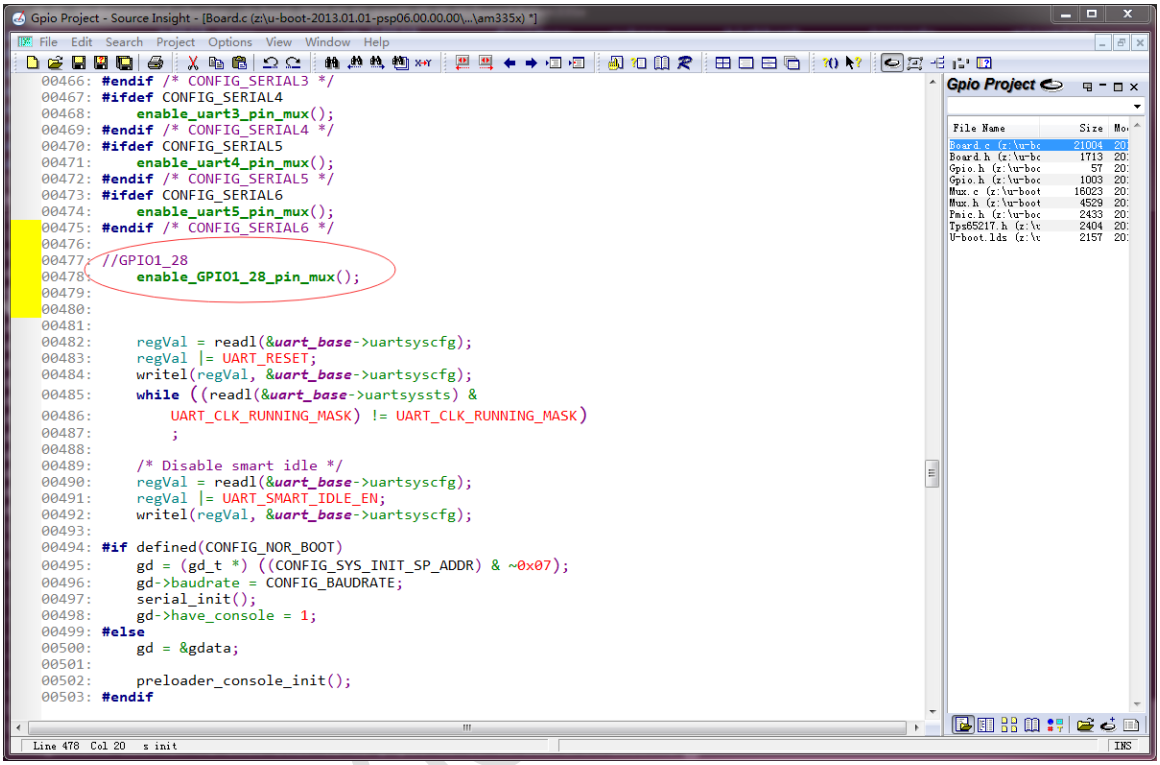
```
void enable_GPIO1_28_pin_mux(void);
```



然后在 u-boot-2013.01.01-psp06.00.00.00\board\ti\am335x\Board.c 中的 s_init 函数调用它，把

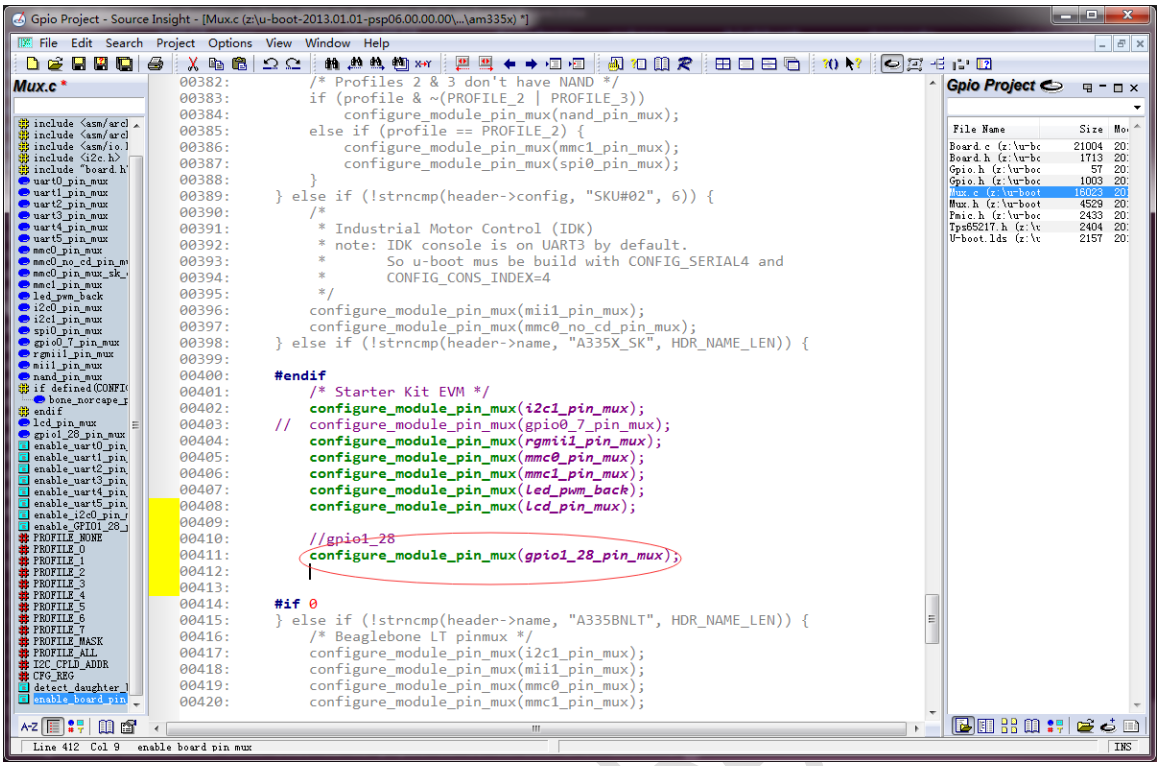
```
enable_GPIO1_28_pin_mux();
```

添加到最后面即可:



（3）、在 s_init 中的函数 “enable_board_pin_mux(&header)（位于 u-boot-2013.01.01-psp06.00.00.00\board\ti\am335x\Mux.c)”函数中新增:

```
configure_module_pin_mux(gpio1_28_pin_mux);
```



（4）、我们需要知道以下几个函数的功能：

- gpio_request(60, "gpio1_28");这个函数的作用是申请 GPIO，检查该 GPIO 口是否可用，如果可用则返回 0，不可以则返回-1。这里的“60”是怎么得到的呢？我们知道AM335x有128个GPIO口，也就是说GPIO1_28是第 32*1+28=60 个，而“gpio1_28”没什么实际用处，可以为空。
- gpio_direction_output(60, 1);将 GPIO1_28 设置为输出并置为高电平：。
- gpio_set_value（60，0）;将 GPIO1_28 拉低；
- gpio_set_value（60，1）;将 GPIO1_28 拉高；
- gpio_direction_input（60）;设置为输入引脚；
- gpio_get_value（60）;获取 GPIO1_28 的电平。

(5)、加入测试代码：

我们可以在 board_late_init()函数中测试该 GPIO 口，用户也可以在别的初始化函数中使用。这里以 board_late_init()中测试为例：

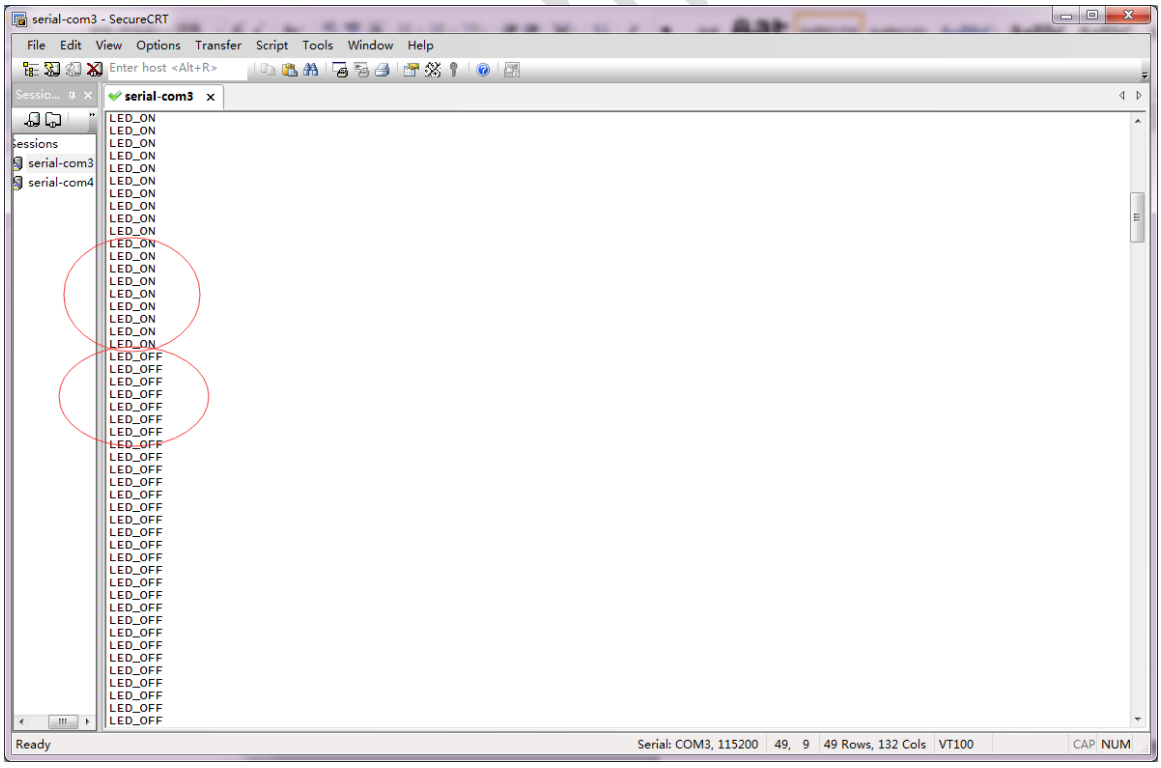
```
int board_late_init(void)
{
...
//前面的代码不要动
//GPIO1_28 test
   int i=0;
   if(gpio_request(60,"gpio1_28")==0)
   {
      gpio_direction_output(60,1);
      while(1)
         {
            gpio_set_value(60,0);
            for(i=0;i<1000;i++)
               printf("LED_ON\n\r");

            gpio_set_value(60,1);
            for(i=0;i<1000;i++)
               printf("LED_OFF\n\r");

         }
```

```
    }
    return 0;
}
```

三、编译并烧写新的 U-Boot

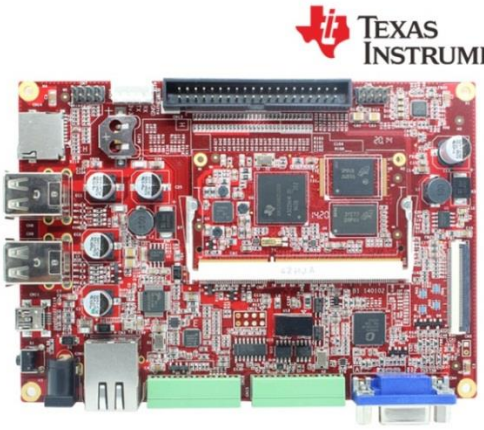
根据《TI AM335x Linux 系统编译 v1.0.docx》将 U-Boot 重新编译并烧写新的 U-Boot，开机后不断打印如下信息，并且可以看到核心板上 GPIO1_28 对应的 Led 灯一闪一闪(由于原代码在成功进入内核后 GPIO1_28 控制的 LED 灯也会一闪一闪，为了避免混淆，测试前请将 uImage 文件删除，这样系统就不会进入内核)。

U-Boot 打印信息如下：

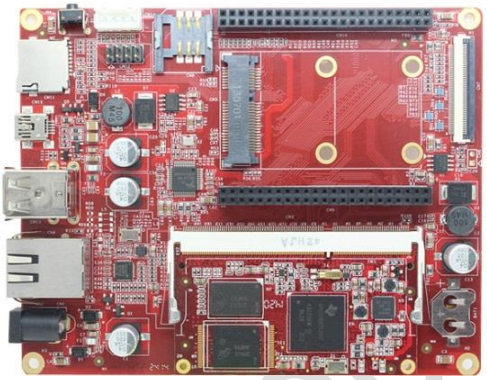


观察核心板可以看到灯一闪一闪，到这里我们成功实现在 U-Boot 中控制 GPIO。

附 相关 GOEMBED 产品介绍



SBC335x – B1A



SBC335x – B2A

The single board computer SBC335x-B1A/B2A which has an expansion board to carry the CM335X is one of our design of the base plate . The flexible design allows the fast and easy way of realizing and upgrading the controller's capabilities. In additional to those features offered by CM335X.

The B1A features 4 serial ports (including 2 RS232 and 2 TTL), 4 USB Host and 1 USB OTG, 1 Ethernet ports, CAN, RS485, Wiegand, VGA, LCD, Touch screen, Audio, ADC and more other peripherals.

The B2A features 4 USB Host and 1 USB OTG, 1 Ethernet ports, LCD, Touch screen,RTC, and more other peripherals.

The SBC board targets a wide range of applications, including: HMIs, Digital Signage, POS, Data Terminal, Medical Devices, Navigation, Industrial Automation, Entertainment system, Thin Clients, Robotics, Game Console and much more.

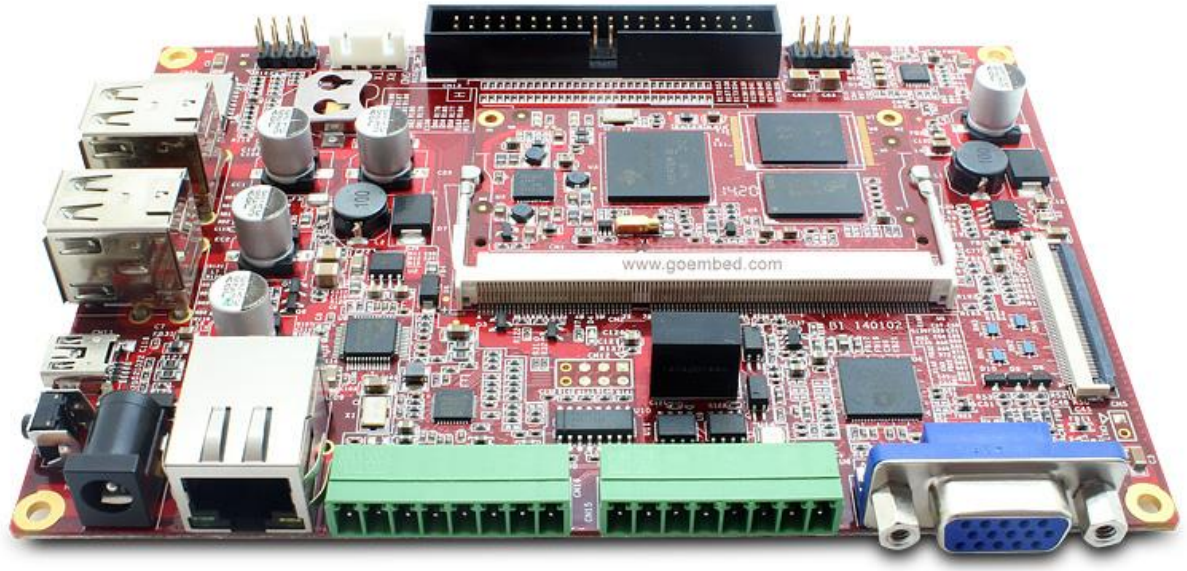
The SBC335x-B1A/B2A are ready-to-run platform to support Linux 3.x, Android 4.x and WinCE 7.0/6.0 operating systems.

If you want to support other Operating System, For more information to contact us.

Single Board Computer

SBC335x-B1A

A perfect solution for upgrading ARM9 or ARM11 devices



CM3352 ACW + B1A = SBC3352 ACW-B1A

SBC335x-B1A boards Description of part code:

Series	B1	B1	B1	B1
Part Code	SBC3352 ACW-B1A	SBC3352 BCW-B1A	SBC3358 ACW-B1A	SBC3358 BCW-B1A
Order Code	-	-	-	-
Core Module	CM3352 ACW -M51E20/08	CM3352 BCW -M51E40/08	CM3358 ACW -M51E20/10	CM3358 BCW -M51E40/10
CPU Type	ARM Cortex™-A8			
CPU Cores	1x			
CPU Clock	800MHz	800MHz	1.0GHz	1.0GHz
RAM DDR3	Micron 512MB@16bit*1			
eMMC Flash	2GB@8bit*1	4GB@8bit*1	2GB@8bit*1	4GB@8bit*1
PMU	TI TPS65910A3			
Supply Voltage	DC 9-14V			
Optimal Input	DC 12V,1.5A			
Size(L*W)	146 x 102 mm			
Temperature	0° to 70° C			
Support OS	Linux 3.x/ Android 4.x/ Ubuntu/ Angstrom/ Debian/ QT/ WinCE 6.0/7.0			
Inventory status	In Stock	Out of Stock Contact us	In Stock	Out of Stock Contact us
Minimum Availability	2022			

SBC335x-B1A Block Diagram

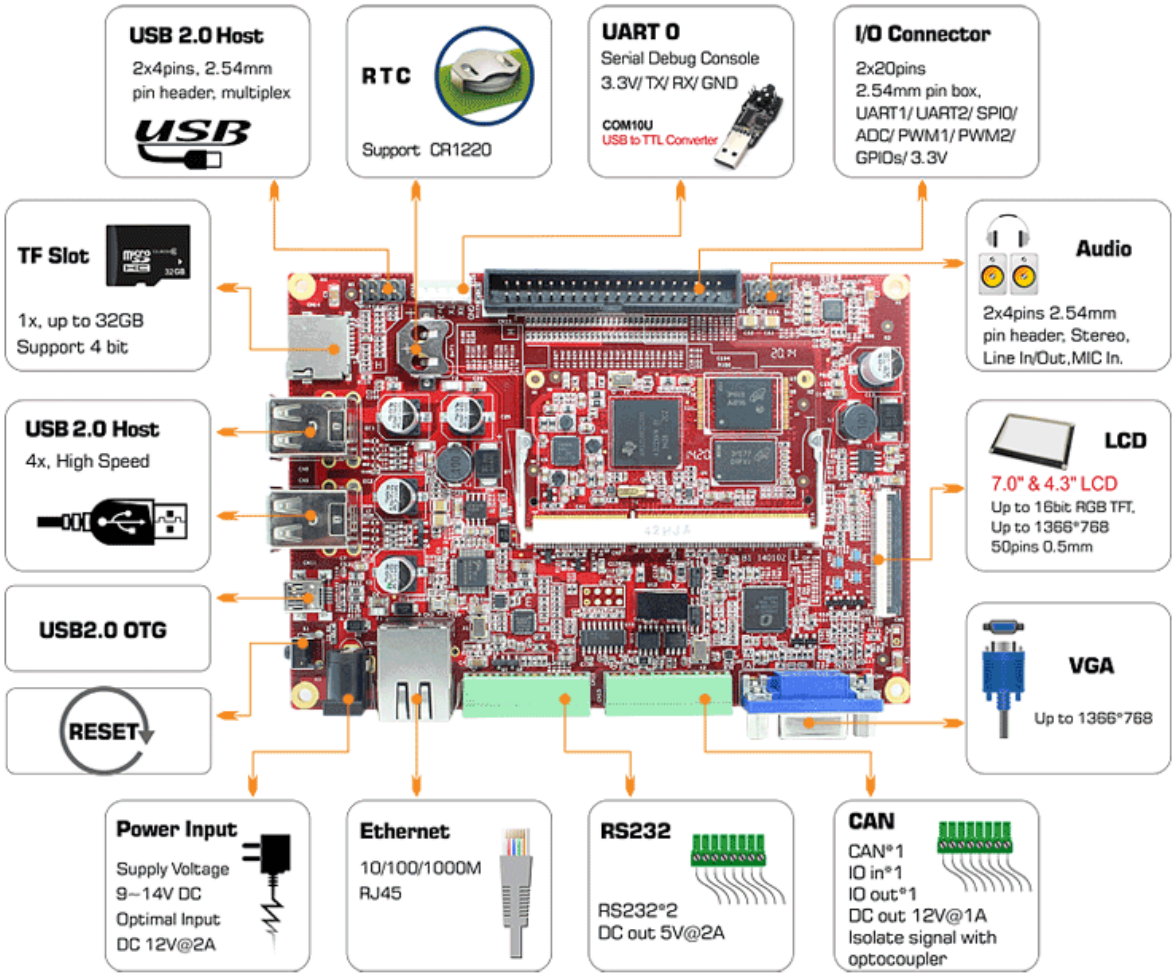


Figure 1 B1 Block Diagram

SBC335x-B2A

Single Board Computer

CM3358 ACW + B2A = SBC3358 ACW-B2A

SBC335x-B2A boards Description of part code:

Series	B2A	B2A	B2A	B2A
Part Code	SBC3352 ACW-B2A	SBC3352 BCW-B2A	SBC3358 ACW-B2A	SBC3358 BCW-B2A
Order Code	-	-	-	-
Core Module	CM3352 ACW -M51E20/08	CM3352 BCW -M51E40/08	CM3358 ACW -M51E20/10	CM3358 BCW -M51E40/10
CPU Type	ARM Cortex™-A8			
CPU Cores	1x			
CPU Clock	800MHz	800MHz	1.0GHz	1.0GHz
RAM DDR3	Micron 512MB@16bit*1			
eMMC Flash	2GB@8bit*1	4GB@8bit*1	2GB@8bit*1	4GB@8bit*1
PMU	TI TPS65910A3			
Supply Voltage	DC 9-14V			
Optimal Input	DC 12V,1.5A			
Size(L*W)	130 x 103.5 mm			
Temperature	0° to 70° C			
Support OS	Linux 3.x/ Android 4.x/ Ubuntu/ Angstrom/ Debian/ QT/ WinCE 6.0/7.0			
Inventory status	In Stock	Out of Stock Contact us	In Stock	Out of Stock Contact us
Minimum Availability	2022			

SBC335x-B2A Block Diagram

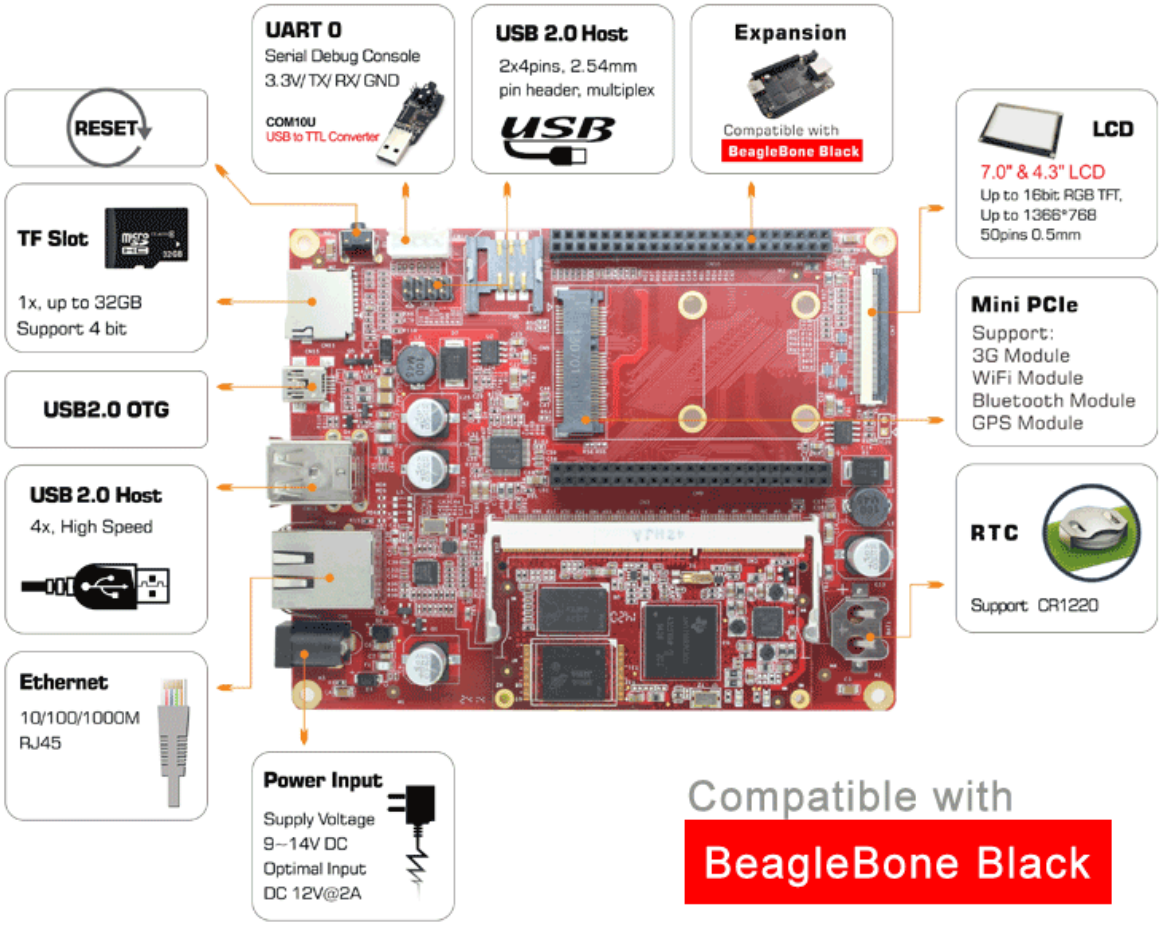


Figure 1 B2A Block Diagram

ABOUT GOEMBED

GOEMBED team with experienced embedded engineers who have been engaged in ARM hardware and software design for 10+ years.

Our products include single board computers and CPU core modules based on TI ® Sitara and Freescale ® i.MX Applications Processors based on ARM® Cores. Supported by Linux / Android / Debian / Ubuntu / QT / Angstrom / WinCE 7.0 & 6.0 / uCOS. We can redesign carrier boards and SBC as your idea quickly.

GOEMBED focus on Embedded Board Solutions, provide a complete new board for your specified requirement or even a turnkey solution to accelerate your new products to market.

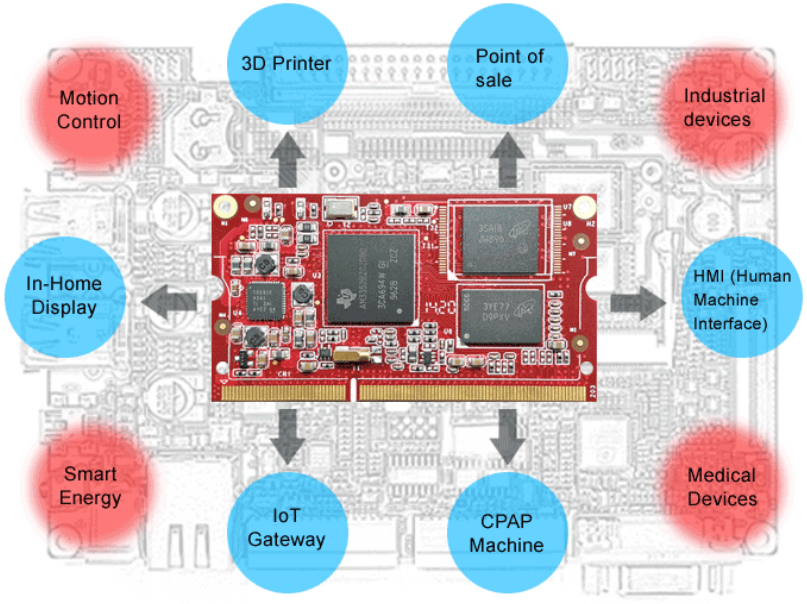
We are your trust worthy partner on ARM embedded design services and solutions.

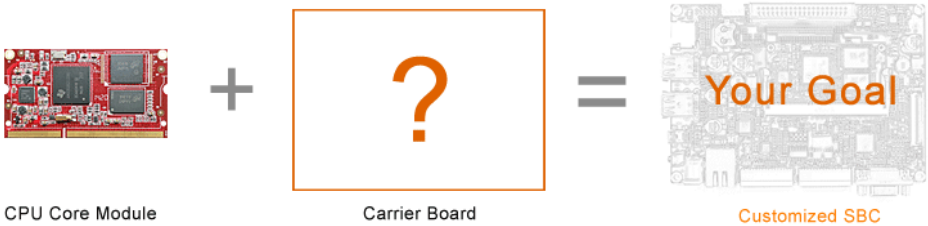
More Carrier Boards

Customized based on your needs!

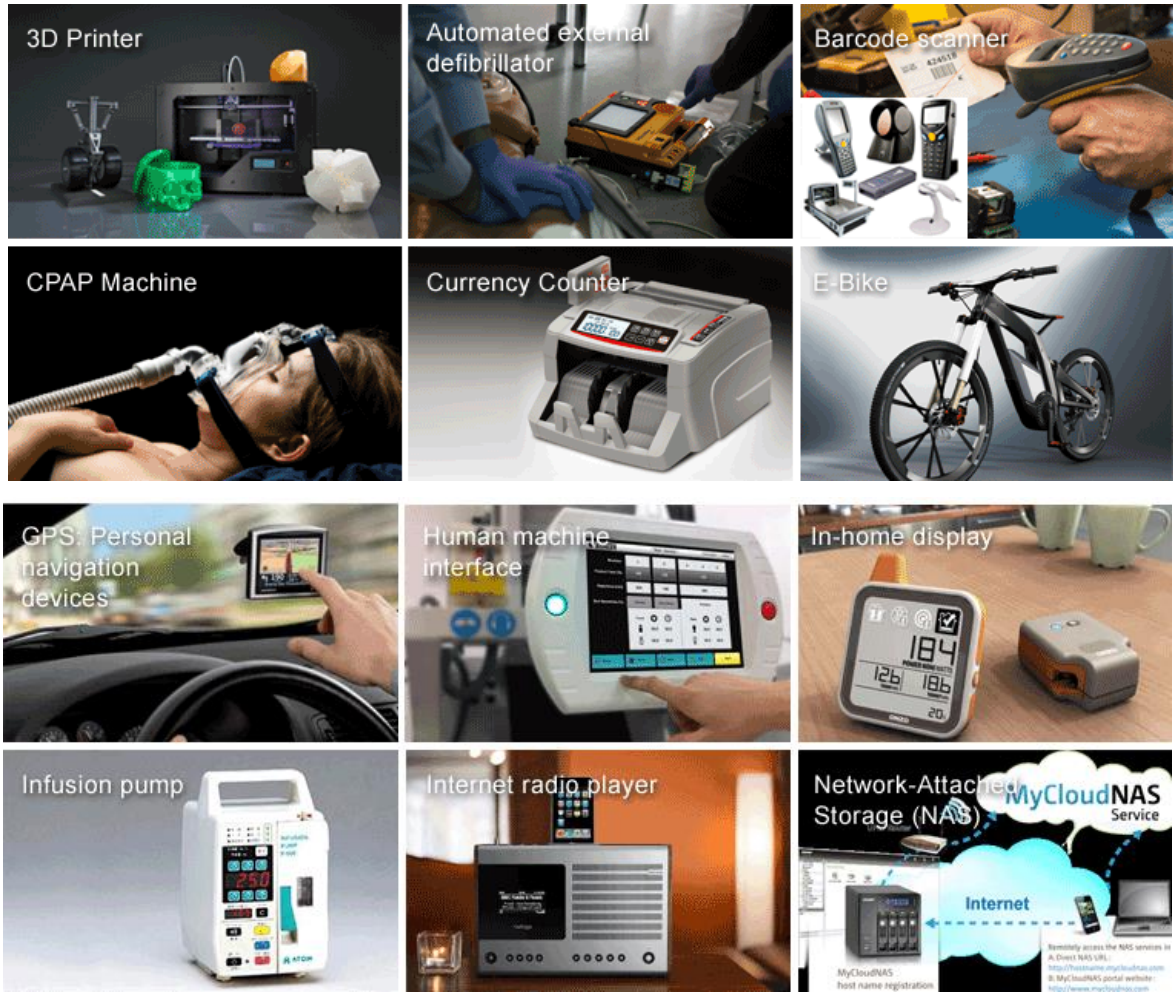
ODM / OEM Services

Bring your new products to market quickly



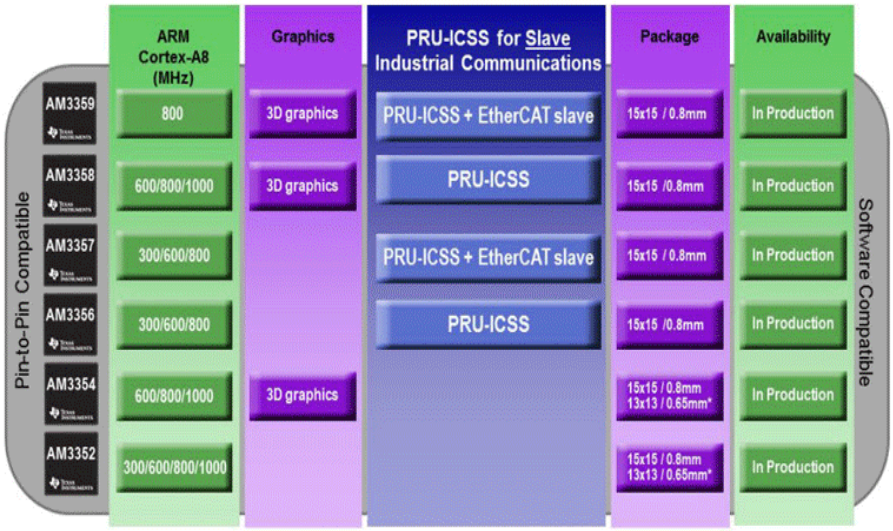


Related end equipment



Learn more applications please click http://www.ti.com/lsds/ti/apps/appshomepage.page

AM335x - A scalable platform with 6 pin-to-pin compatible devices



✓PRU-ICSS is used for slave industrial communication protocols such as Profibus, Profinet, Powerlink & Ethernet/IP

Package	15x15mm (ZCZ)	13x13mm (ZCE)
ARM speed	Up to 1000 MHz	Up to 600 MHz
USB 2.0 OTG + PHY	x2	x1
EMAC	2-port switch	Single port

TI Sitara ARM Cortex-A8 AM335x processors information (Content from TI):

AM335x Cortex™-A8 based processors

Benefits
- High performance Cortex-A8 at ARM9/11 prices
- Rich peripheral integration reduces system complexity and cost

Sample Applications
- Industrial / Home Automation
- Portable Navigation Devices
- Robotics
- Consumer electronics
- Smart Appliances
- Low power instrumentation
- Wireless Accessories
- Networking

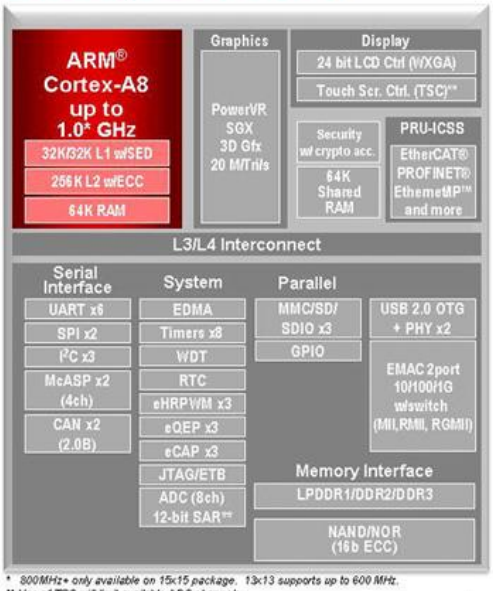
Software and development tools
- Free Linux and Android support packages direct from TI
- StarterWare enables quick and simple programming and migration among TI embedded processors
- WinCE and RTOS (QNX, Wind River, Mentor, etc.) from partners
- Full featured and low cost development board options

Power Estimates
- Total Power: 600mW-1000mW
- Standby Power: ~25mW
- Deep Sleep Power: ~5-7mW

Schedule and packaging
- Status: In production
- Dev. Tools: Available today
- Docs: Available today
- Packaging: 13x13, 0.65mm via channel array; 15x15, 0.8mm

More Information
- www.ti.com/am335x

Availability of some features, derivatives, or packages may be delayed from initial silicon availability
Peripheral limitations may apply among different packages
Some features may require third party support
All speeds shown are for commercial temperature range only



** 800MHz+ only available on 15x15 package. 13x13 supports up to 600 MHz.*
*** Use of TSC will limit available ADC channels.*
SED: single error detection/parity

1

Texas Instruments