

## PY32T09x 的应用 注意事项

### 前言

PY32T09x 系列微控制器采用高性能的 32 位 ARM® Cortex®-M0+ 内核，宽电压工作范围的 MCU。嵌入最高 256Kbytes Flash 和 最高 32 Kbytes SRAM 存储器，最高工作频率 72MHz。包含多种不同封装类型多款产品。

本应用笔记将帮助用户了解 PY32T09x 各个模块应用的注意事项，并快速着手开发。

表 1. 适用产品

类型	产品系列
微型控制器系列	PY32T09x

## 目录

1	SRAM 使用注意事项.....	3
2	PWR 使用注意事项.....	3
3	ADC 使用注意事项.....	4
4	LPTIM 配置注意事项.....	5
5	I2C 注意事项.....	7
6	SPI 使用注意事项.....	7
7	USART 使用注意事项.....	7
8	LPUART 使用注意事项.....	8
9	BOR 使用注意事项.....	9
10	HSI 使用注意事项.....	9
11	GPIO 使用注意事项.....	10
12	版本历史.....	11

## 1 SRAM 使用注意事项

- 建议设置 IRAM 为芯片的 SRAM 起始地址和 SRAM 空间容量。如图 1-1 举例说明，使用 PY32T090xB 系列的芯片时，IRAM 的配置应为该芯片的 SRAM 起始地址和空间容量。PY32T090 系列的 SRAM 空间容量请参考《PY32T090 系列数据手册》表 1-1:

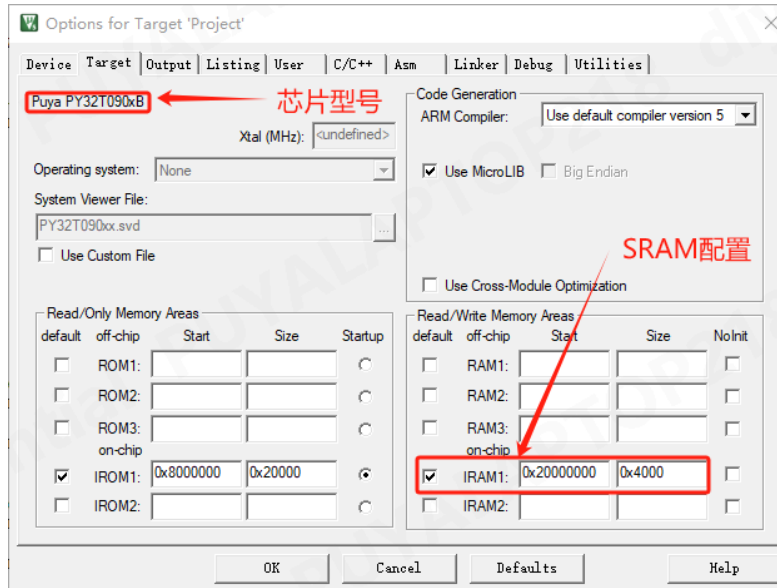


图 1-1

- 如果配置 IRAM 的 Size 小于芯片 SRAM 的空间容量时，需要手动将 IRAM 配置以外的 SRAM 空间初始化为零。

## 2 PWR 使用注意事项

### 2.1 LPRUN/LPSLEEP 模式操作注意事项

- 复位源限制：仅支持 PIN 复位 和 IWDG 复位，禁止开启以下系统软件复位：
  - 窗口看门狗复位 (WWDG)
  - 选项字节加载复位 (OBL)
  - SYSRESETREQ 软件复位
  - SRAM 奇偶校验错误复位
- LPRUN/LPSLEEP 切换模式前（如：RUN/STOP/STANDBY）需满足以下条件：
  - 禁止使能 IWDG 或手动刷新 IWDG\_KR\_KEY[15:0] 为 0xAAAA，确保 IWDG 在切换时不产生复位；
  - 手动清零 SYSCFG->SCSR\_PERR\_RSTEN，使 SRAM 奇偶校验出错时产生 NMI 中断而不

是系统复位;

## 2.2 STOP 模式操作注意事项

- 时钟限制：仅支持 8MHz HSI 进入 STOP 模式
- 切换模式前需满足以下条件：
  - 手动刷新IWDG\_KR\_KEY[15:0]为0xAAAA，确保 IWDG 在切换模式时不产生复位
  - 手动刷新WWDG\_CR\_T[6:0]为预设的计数值，确保WWDG在切换模式时不产生复位
  - 手动清零SYSCFG\_SCSR\_PERR\_RSTEN，使SRAM奇偶校验出错时产生NMI中断而不是系统复位
  - NRST Pin的复位信号需保持低电平  $\geq 80\mu\text{s}$
- 唤醒后处理：若使用 WFE 指令 进入 STOP 模式，需在指令后插入  $\geq 10$  个 NOP，确保时钟门控同步。

## 3 ADC 使用注意事项

### 3.1 ADC 硬件使用注意事项

- 时钟约束：AD\_CLK 频率 $\leq$ PCLK，可通过寄存器 ADC\_CCR.CKMODE 选择 PLCK 或 ADC 异步时钟

### 3.2 ADC 软件使用注意事项

- ADC 配置为非连续模式，注入转换序列的情况下，禁止使用软件触发，仅支持外设触发，禁用 AUTODLY 模式。具体代码如下：

```
ADC_HandleTypeDef *hadc;

/* ADC Init Code Here*/
/* Discontinuous Conversion */
SET_BIT(hadc->Instance->CFGR, ADC_CFGR_DISCEN);
CLEAR_BIT(hadc->Instance->CFGR, ADC_CFGR_CONT);

/* Disable Autodelay */
CLEAR_BIT(hadc->Instance->CFGR, ADC_CFGR_AUTDLY);

/* Software Trigger is Forbidden*/
SET_BIT(hadc->Instance->JSQR, ADC_CFGR_JEXTEN);
```

- ADC 配置为单次转换模式，注入转换序列的情况下，禁止使用软件触发，仅支持外设触发。具体代码如下：

```
ADC_HandleTypeDef *hadc;

/* ADC Init Code Here*/
/* Single Conversion */
```

```

CLEAR_BIT(hadc->Instance->CFGR, ADC_CFGR_DISCEN);
CLEAR_BIT(hadc->Instance->CFGR, ADC_CFGR_CONT);

/* Software Trigger is Forbidden*/
SET_BIT(hadc->Instance->JSQR, ADC_CFGR_JEXTEN);

```

- ADC 配置为注入转换，软件触发加等待模式的情况下，需要等待注入通道序列结束，即 ADC\_ISR\_JEOS = 1，并将其软件清零后,启动 ADC，具体代码如下：

```

ADC_HandleTypeDef *hadc;
ADC_InjectionConfTypeDef *pConfigInjected

/* Injected Conversion Configure*/
pConfigInjected->ExternalTrigInjecConv = ADC_INJECTED_SOFTWARE_START;

/* Other Injected Conversion Configure Here*/

/* Enable Autodelay */
SET_BIT(hadc->Instance->CFGR, ADC_CFGR_AUTDLY);

While(1)
{
    /* Start ADC Conversion */
    SET_BIT(hadc->Instance->CR, ADC_CR_JADSTART);

    /* Wait for JEOS is set, then clear JEOS */
    while(__HAL_ADC_GET_FLAG(hadc, ADC_FLAG_JEOS) == 0);
    CLEAR_BIT(hadc->Instance->ISR, ADC_FLAG_JEOS);
    /* User Code here */
}

```

## 4 LPTIM 配置注意事项

- 关闭编码器模式之前需要清零所有标志位，特别是 UP 和 DOWN 标志位；
- LPTIM 配置操作模式步骤，下列代码展示两种不同 kernel 时钟的配置步骤：
  - 设置LPTIM的kernel时钟为PCLK/LSI
  - 使能LPTIM1/LPTIM2的时钟
  - 配置LPTIM触发方式
  - 等待5\*LPTIM kernel CLK，切换LPTIM的kernel时钟为低频时钟
  - 配置单次/连续计数模式。具体代码实现如下：

```

/* Select PCLK as LPTIM1 Clock source */
__HAL_RCC_LPTIM1_CONFIG(RCC_LPTIM1CLKSOURCE_PCLK);

__HAL_RCC_LPTIM1_CLK_ENABLE();

```

```

/* LPTIM Configure Code: */
/* LPTIM kernel Clock Enable Code Here */
/* LPTIM Configure Code Here */

/* If Select PCLK as LPTIM1 Clock source, Delay 5 NOP */
__NOP();
__NOP();
__NOP();
__NOP();
__NOP();

/* Select LPTIM source kernel clock as LSI*/
__HAL_RCC_LPTIM1_CONFIG(RCC_LPTIM1CLKSOURCE_LSI);

/* Set LPTIMx Single/Continuous mode */
hlptim->Instance->CR |= LPTIM_CR_CNTSTRT;

```

```

/* Select LSI as LPTIM1 Clock source */
__HAL_RCC_LPTIM1_CONFIG(RCC_LPTIM1CLKSOURCE_LSI);
__HAL_RCC_LPTIM1_CLK_ENABLE();

/* LPTIM Configure Code: */
/* LPTIM kernel Clock Enable Code Here */
/* LPTIM Configure Code Here */

/* If Select LSI as LPTIM1 Clock source, Delay 5 LSI Clock */
LPTIM_DelayLSI();
LPTIM_DelayLSI();
LPTIM_DelayLSI();
LPTIM_DelayLSI();
LPTIM_DelayLSI();

/* Select LPTIM source kernel clock as LSI*/
__HAL_RCC_LPTIM1_CONFIG(RCC_LPTIM1CLKSOURCE_LSI);

/* Set LPTIMx Single/Continuous mode */
hlptim->Instance->CR |= LPTIM_CR_CNTSTRT;

```

- 对 CFGR 寄存器写入时，推荐寄存器一次性写入配置值，否则两次写入操作之间必须间隔至少 3\*LPTIM\_kernel\_CLK。具体代码操作参考以下：

```

uint32_t tmpcfgr;
/* If Select LSI as LPTIM Clock source, Delay 3 LSI Clock */
/* Example: Set initialization parameters separately */
hlptim->Instance->CFGR |= hlptim->Init.Clock.Source |
                        hlptim->Init.Clock.Prescaler;

/* Delay */
DelayLSI();
DelayLSI();
DelayLSI();

```

```

hlptim->Instance->CFGR |= hlptim->Init.OutputPolarity | hlptim->Init.UpdateMode);
/* Delay */
DelayLSI();
DelayLSI();
DelayLSI();
/* Example: Set initialization parameters at once */
hlptim->Instance->CFGR = tmpcfgr;

void DelayLSI()
{
    /* Calculate the value required for a delay of macro-defined(Delay) us */
    uint32_t RatioNops = 40 * (SystemCoreClock / 1000000U) / 5;
    for(uint32_t i=0; i<RatioNops;i++)
    {
        __NOP();
    }
}

```

## 5 I2C 注意事项

- 不支持多主机仲裁
- 进入 I2C 中断后需要手动清除 I2C 中断标志位，推荐客户使用以下例程自带中断程序进行处理：

```

/**
 * @brief This function handles I2C1 interrupt.
 */
void I2C1_IRQHandler(void)
{
    HAL_I2C_IRQHandler(&I2cHandle);
}

```

## 6 SPI 使用注意事项

- 使用 TI 模式时，如需关闭 SPI 功能，则需要同时关闭 TI 模式。具体代码实现如下：

```

/* DeInit SPI1 */
if (HAL_SPI_DeInit(&Spi1Handle) != HAL_OK)
{
    APP_ErrorHandler();
}
/* Disable SPI */
CLEAR_BIT(Spi1Handle->Instance->CR1, SPI_CR1_SPE);
CLEAR_BIT(Spi1Handle->Instance->CR2, SPI_CR2_FRF);

```

## 7 USART 使用注意事项

- 自动波特率检测过程中，禁止清零 USART\_CR3\_ABREN、USART\_CR1\_RE 标志位

- 若对 IDLE 帧有精确要求，Stop 位需配置为 1 位

## 8 LPUART 使用注意事项

- 对 CR3 寄存器写入时，推荐寄存器一次性写入配置值，否则两次写入操作之间必须间隔 5\*LPUART\_kernel\_CLK。具体代码实现如下：

```

LPUART_HandleTypeDef *hlpuart;
uint32_t tmpcr3;

/* Enable the Driver Enable mode by setting the DEM bit in the CR3 register */
/* If Select LSI as LPUART Clock source, Delay 5 LSI Clock */
SET_BIT(hlpuart->Instance->CR3, LPUART_CR3_DEM);
DelayLSI();
DelayLSI();
DelayLSI();
DelayLSI();
DelayLSI();

/* Set the Driver Enable polarity */
MODIFY_REG(hlpuart->Instance->CR3, LPUART_CR3_DEP, Polarity);
DelayLSI();
DelayLSI();
DelayLSI();
DelayLSI();
DelayLSI();

/* Enable the Driver Enable mode by setting the DEM bit in the CR3 register and Set the Driver
Enable polarity */
/*at once*/
tmpcr3 |= LPUART_CR3_DEM;
tmpcr3 |= LPUART_CR3_DEP;
hlpuart->Instance->CR3 = tmpcr3;
/**
 * @brief Delay 1 LSI
 * @retval None
 */
void DelayLSI()
{
    /* Calculate the value required for a delay of macro-defined(Delay) us */
    uint32_t RatioNops = 40 * (SystemCoreClock / 1000000U) / 5;
    for(uint32_t i=0; i<RatioNops;i++)
    {
        __NOP();
    }
}

```

## 9 BOR 使用注意事项

- 配置 User Option1 中的 BOR 相关选项时，如果选择使能 BOR\_EN，则需将 BOR\_LEV 配置为 BOR level1 及以上的选项。如图 9-1 所示，使用 PY32CubeProgrammer 时配置 BOR 相关选项。

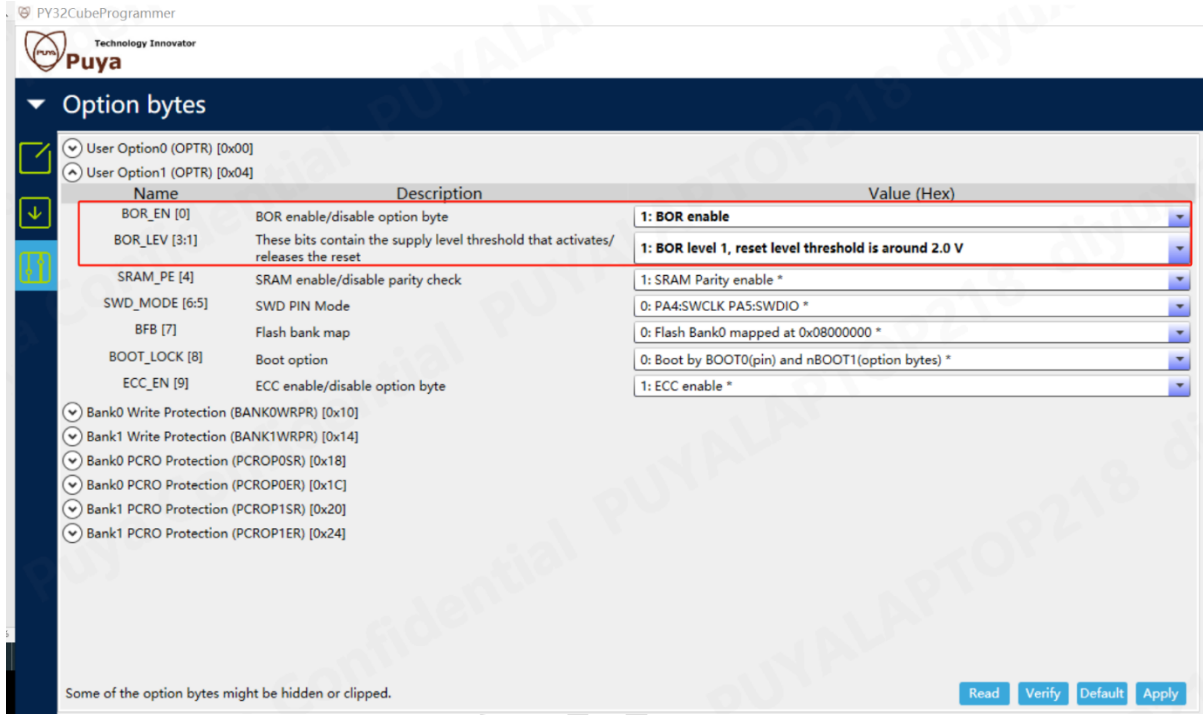


图 9-1

## 10 HSI 使用注意事项

- 表 10-1 为 PY32T090 的内部高频时钟源 HSI 的特性，请参考表 10-1，在相应的条件下选择合适的 HSI 频率。更详细的特性参考《PY32T090 系列数据手册》表 5.23 内部高频时钟源特性。

符号	参数	条件	最小值	典型值	最大值	单位
f <sub>HSI</sub>	HSI 频率	V <sub>CC</sub> = 3.3 V, T <sub>A</sub> = 25 °C	-	8.0 16.0 24.0 48.0 64.0	-	MHz
Δ <sub>Temp</sub> (HSI)	HSI 16/24/48 MHz 频率温度漂移	V <sub>CC</sub> = 3.3 V, T <sub>A</sub> = 25 °C	-1 <sup>(2)</sup>	-	1 <sup>(2)</sup>	%
		V <sub>CC</sub> = 2.0 ~ 5.5 V, T <sub>A</sub> = -20 ~ 85 °C	-2 <sup>(2)</sup>	-	2 <sup>(2)</sup>	
		V <sub>CC</sub> = 1.8 ~ 5.5 V, T <sub>A</sub> = -40 ~ 105 °C	-3 <sup>(2)</sup>	-	3 <sup>(2)</sup>	
	HSI 8/64 MHz 频率温度漂移	V <sub>CC</sub> = 3.3 V, T <sub>A</sub> = 25 °C	-1 <sup>(2)</sup>	-	1 <sup>(2)</sup>	
		V <sub>CC</sub> = 2.0 ~ 5.5 V, T <sub>A</sub> = -20 ~ 85 °C	-2.5 <sup>(2)</sup>	-	2.5 <sup>(2)</sup>	
		V <sub>CC</sub> = 1.8 ~ 5.5 V, T <sub>A</sub> = -40 ~ 105 °C	-5 <sup>(2)</sup>	-	5 <sup>(2)</sup>	
f <sub>TRIM</sub> <sup>(1)</sup>	HSI 微调精度	-	0.02	0.1	0.2	%
D <sub>HSI</sub> <sup>(1)</sup>	占空比	-	45 <sup>(1)</sup>	-	55 <sup>(1)</sup>	%

符号	参数	条件	最小值	典型值	最大值	单位
t <sub>Stab</sub> (HSI)	HSI 稳定时间	-	-	3	4 <sup>(1)</sup>	μs
I <sub>CC</sub> (HSI) <sup>(2)</sup>	HSI 功耗	8 MHz	-	108	-	μA

表 10-1

## 11 GPIO 使用注意事项

- 使能 120mA 灌电流功能前，需确保 GPIO 为低电平输出状态；输出高电平前，必须预先关闭该功能。PY32T09x 系列中，具有 120mA 灌电流功能的引脚为 PB2~PB9，下列具体代码以配置 PB2 的灌电流功能为例：

```

GPIO_InitTypeDef GPIO_InitStructure = {0};
__HAL_RCC_GPIOB_CLK_ENABLE();

/* Initialize GPIO PB2 */
GPIO_InitStructure.Pin = GPIO_PIN_2;
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;          /* Push-pull output */
GPIO_InitStructure.Pull = GPIO_PULLUP;                  /* Enable pull-up */
GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_VERY_HIGH;  /* GPIO speed */

HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);             /* GPIO initialization */

/* Set GPIO PB2 Output Low Level*/
GPIOB->BRR = GPIO_PIN_2;
/* Then Enable LED Pin High Drive*/
SET_BIT(SYSCFG->LEDCFG, SYSCFG_EHS_PB2);

/* To Set GPIO PB2 Output High, LED Pin High Drive Should Be Disabled First*/
CLEAR_BIT(SYSCFG->LEDCFG, SYSCFG_EHS_PB2);
/* Then Set GPIO PB2 Output High Level*/
GPIOB->BSRR = GPIO_PIN_2;

```

## 12 版本历史

版本	日期	更新记录
V1.0	2025.01.26	初版
V1.1	2025.09.10	1. 添加第 9 点 2. 添加第 10 点 3. 修改文档中芯片系列名称 PY32T090_PY32T092 为 PY32T09x
V1.2	2026.03.25	1. 添加第 11 点
V1.3	2026.05.19	1. 修改第 3、4 点中部分寄存器名称 2. 更新表 10-1



Puya Semiconductor Co., Ltd.

### 声 明

普冉半导体(上海)股份有限公司 (以下简称: “Puya”) 保留更改、纠正、增强、修改 Puya 产品和/或本文档的权利, 恕不另行通知。用户可在下单前获取产品的最新相关信息。

Puya 产品是依据订单时的销售条款和条件进行销售的。

用户对 Puya 产品的选择和使用承担全责, 同时若用于其自己或指定第三方产品上的, Puya 不提供服务支持且不对此类产品承担任何责任。

Puya 在此不授予任何知识产权的明示或暗示方式许可。

Puya 产品的转售, 若其条款与此处规定不一致, Puya 对此类产品的任何保修承诺无效。

任何带有 Puya 或 Puya 标识的图形或字样是普冉的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代并替换先前版本中的信息。

普冉半导体(上海)股份有限公司 - 保留所有权利