S32G ICU GPIO Input

by John Li (nxa08200)

本文说明如何配置MCAL ICU模块为GPIO Input。

默认的MCAL ICU模块是使用FTM输入为示例的。

本文采用软件版本为MCAL RTD 4.0.2。

历史	说明	作者
V1	● 创建本文	John.Li

目录

1 背景与资料说明

1.1 背景说明

目前 MCAL 的 ICE 模块驱动示例为 FTM 输入，参考文档：C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Icu_TS_T40D11M40I2R0\examples\EBT\S32G3\Icu_CheckNofi_S32G399A_M7\readme.txt 说明。

而在文档《S32G3_Standby_QSPIPowerDown_Demo_V*》说明了 WakeUp Input 事件的配置方法。

而 ICU 一般支持：

- FTM
- WakeUp
- GPIO

三种输入方法，所以现在还缺少 GPIO 输入的 ICE 示例说明，所以本文及本工程创建此工程示例。

1.2 所需资料说明

软件/工具	名称	如何获得
MCAL RTD	RTD_4.0.2	从www.nxp.com/s32g 个人帐号下载
M7 STBY AppNote	《S32G3_Standby_QSPIPowerDown_Demo_V*》	从https://community.nxp.com/t5/NXP-Designs-Knowledge-Base/S32G3-Standby-QSPIPowerDown/ta-p/1824728 下载
S32G3 Refercence Manual	《S32G3RM》	从www.nxp.com/s32g 下载
M7 STBY AppNote	《S32G3_IOMUX.xlsx》	S32G3RM.pdf 附件

2 创建 ICU 工程

2.1 打开工程

打开 EB tresos Studio 27.1 后 File->Import->General->Existing Projects into Workspace->Select root directiory Browse-> C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Icu_TS_T40D11M40I2R0\examples\EBT\S32G3\Icu_CheckNofi_S32G399A_M7。

选中：Copy projects into workspace。

点击 Finish，导入工程。

在 Icu_CheckNofi_S32G399A_M7 上右击->Properties->Configuration Project->Code Generator->Default Generation Patch=..\..\generate。

所以工程文件是在：C:\EB\tresos\workspace\Icu_CheckNofi_S32G399A_M7。

在 Icu_CheckNofi_S32G399A_M7 上右击->Generate Project 后，生成的配置文件在：

C:\EB\tresos\generate。将此目录拷贝到 C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Icu_TS_T40D11M40I2R0\examples\EBT\S32G3\\Icu_CheckNofi_S32G399A_M7\generate 下。

2.2 编译与运行

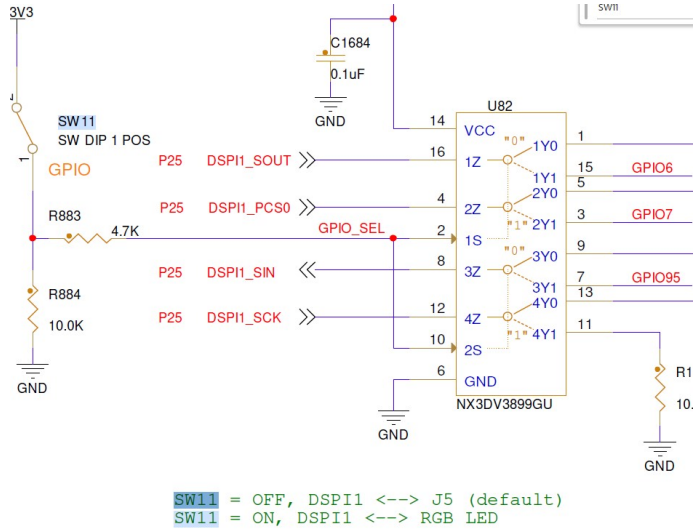
参考文件：C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Ocotp_TS_T40D11M40I2R0\examples\EBT\S32G3\Ocotp_Example_S32G399A_M7\readme.txt，来修改编译配置文件：C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Icu_TS_T40D11M40I2R0\examples\EBT\S32G3\Icu_CheckNofi_S32G399A_M7\project_parameters.mk

```
GCC_DIR = C:/NXP/S32DS.3.5/S32DS/build_tools/gcc_v9.2/gcc-9.2-arm32-eabi
TRESOS_DIR = C:/EB/tresos
T32_DIR = C:/T32
```

并在 cygwin 中编译：

```
make build
…
Compiling ../../../../..//Platform_TS_T40D11M40I2R0/startup/src/m7/gcc/startup_cm7.s
Linking main.elf
```

将S32G3 RDB3板设置为下载模式，连接上电源和Lauterbach，SW11切换为ON，如下设置：



将GPIO6/7/95 LED灯切换为GPIO模式(注意这个不是工程要求的测试方法)。

然后使用Lauterbach加载脚本：

C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Icu_TS_T40D11M40I2R0\examples\EBT\S32G3\Icu_CheckNofi_S32G399A_M7\debug\run.cmm

并运行，可以看到LED蓝灯闪烁。

2.3 默认工程说明

Icu_CheckNofi_S32G399A_M7->someId(…)->Icu(…)->Icu->IcuChannel->IcuChannel_0->General：

- IcuChannelRef = /Icu/Icu/IcuConfigSet/IcuFtm_0/IcuFtmChannels_0 //IcuChannel 指向一个 FTM channel.
- IcuDefaultStartEdge = ICU_BOTH_EDGES
- IcuMeasurementMode= ICU_MODE_SIGNAL_EDGE_DETECT

->IcuSignalEdgeDetection： //由于之前选择的 measurement mode is "IcuSignalEdgeDetection"

- IcuSignalNotification= User_EdgeDetect0 //中断通知函数名称。

Icu_CheckNofi_S32G399A_M7->someId(…)->Icu(…)->Icu->IcuFtm 可以自行分析。

主测试函数：

```
void User_EdgeDetect0(void)
{
  /* increment IRQ counter */
  UserCountIrqCH0++;
}
int main(void)
```

```
{…
  Icu_EnableEdgeDetection(IcuChannel_0);
  Icu_EnableNotification(IcuChannel_0);

  while (UserCountIrqCH0 < 10U)
  {
    if(PulseCount % 2 == 0)
    {
      Dio_WriteChannel(DioConf_DioChannel_Port_pin_InputForIcuChannel0, STD_HIGH);
    }
    else
    {
      Dio_WriteChannel(DioConf_DioChannel_Port_pin_InputForIcuChannel0, STD_LOW);
    }
    delay();
    PulseCount++;
  }
```

参考文档 C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Ocotp_TS_T40D11M40I2R0\examples\EBT\S32G3\Ocotp_Example_S32G399A_M7\readme.txt 说明：

```
Main application is toggling an output pin connected to an input pin routed to FTM.
      Interrupt will be generated for FTM channel.
      User can exercise in debug mode the toggling of pin and generation of interrupt.
…
2.1.2 Connections
      Wire connection is required as follows:
        - SW11 - switch
        - Pin J5.16 (PA_06 - gpio output for Icu_Ftm channel) to J5.15 (PF_15 - FTM_1_CH_1 input)
        - SW11 = OFF, DSPI1 <--> J5(default).
```

其原理是使用了 PA_06 做 GPIO6 输出，回环到 PF_15 做为 FTM 输入，然后触发中断处理函数 User_EdgeDetect0 对 UserCountIrqCH0++；而主函数中不断翻动 GPIO 去触发 FTM，计算到 10 后退出。

文中要求是 SW11=OFF，然后连接上 J5 外部连接线，将 PA_06 到 PF_15 连接后，用 Lauterbach 跟踪来测试。

实际测试中使用 SW11=ON，然后用 LED 灯来 Demo 也可以起到类似效果。

3 增加 GPIO 输入支持

3.1 修改说明

本节的目的是说明如何将一下 FTM input channel 修改为 GPIO input channel，然后由于我们要选择有 EIQ 输入功能的管脚，所以对输入的 GPIO 管脚需要重新选择，根据文档：S32G3_IOMUX.xlsx：

	SIUL_OFFCC	EIRQ[0]	0000_0000	-	disable_low
			0000_0001	-	disable_high
			0000_0010	IO_PAD	PB_03
			0000_0011	IO_PAD	PJ_07

PJ_07	151	SIUL_OFFCC	0000_0000	0x4401049C	GPIO[151]
PJ_07			0000_0001		LLCE_CAN4_TX
PJ_07			0000_0010		FTM0_CH0_O
PJ_07			0000_0011		PFE_MAC0_PPS[0]
PJ_07	655	SIUL_OFFCC	0000_0100	0x44010C7C	FTM0_CH0_I
PJ_07	910	SIUL_OFFCC	0000_0011	0x44011078	EIRQ[0]

所以选择 PJ_07，SIUL2，GPIO151， EIRQ[0]，LLCE_CAN4_TX 作为输入 GPIO，然后输出依然使用 PA_06，SW11=ON，用灯来显示。

注意要选用具有 EIRQ 输入功能的 GPIO 管脚，他们全在 SIUL2_1 上。

3.2 修改 Port 模块

Icu_CheckNofi_S32G399A_M7->someId(…)->Port(…)->Port->Portcontainer->PortContainer_0->PortPin：修改 FTM1_CH1：

- Name= GPIO_EIRQ0
- PortPin SIUL2 Instance*= SIUL2_1
- PortPin Id*=1//自动计算
- PortPin Mscr (dynamic range)*=151 //见上表
- PortPin Mode= SIUL_OFFCC_EIRQ_0 //注意不要配置成 GPIO //
- PortPin Level Value= PORT_PIN_LEVEL_HIGH
- PortPin Pull Enable*= checked

- PortPin Pull Select=checked //以上三项，由于我们需要配置成下降沿触发，所以我们将此管脚默认配置成上拉。

->General :

- PortNumberOfPortPins =2//自动计算

3.3 修改 ICU 模块

1. 删除之前的 ftmchannel：

Icu_CheckNofi_S32G399A_M7->someId(…)->Icu(…)->Icu->IcuFtm：

将 IcuFtm_0 删除掉。

2. 增加 IcuSiul2 配置：

Icu_CheckNofi_S32G399A_M7->someId(…)->Icu(…)->Icu->IcuSiul2，点击+增加 IcuSiul2_0：

->SIUL2 Modules：

- SIUL2 instance (dynamic range) =1//默认 Select the hardware instance of SIUL2.
  Only support interrupt in SIUL2_1 instance. SIUL2_0 has no interrupt.
- ICU External Interrupt Filter Clock Prescaler (0 -> 15) = 2// Configure the clock prescaler which is used to select the clock for all digital filter counters in the SIUL2

  寄存器：IFCPR:

3-0 IFCP	Interrupt Filter Clock Prescaler setting Prescaled Filter Clock period is T_{IRC} × (IFCP + 1), where: • T_{IRC} is the internal oscillator period. • IFCP is 0 to 15.

->IcuSiul2Channels：点击+号增加：IcuSiul2Channels_0

- Siul2 Channel (dynamic range)* =0 //EIRQ0
- ICU External Enable Interrupt Filter = checked ： // Enable external digital filter counter on the interrupt pads to filter out glitches on the inputs

  寄存器：IFER0:

0 IFE0	Enables digital glitch filter on the interrupt pad input. 0b - Disabled 1b - Enabled

- ICU External Interrupt Filter setting (0 -> 15)=3 // Maximum Interrupt Filter Counter setting.

  寄存器：IFMCR0：

3-0 MAXCNT	Maximum Interrupt Filter Counter setting MAXCNT can be 0d to 15d. • A value of 0d, 1d, or 2d sets the filter as an all pass filter. • A value of 3d to 15d sets the filter period to TCK × MAXCNT + *n* × TCK, where: — *n* is 1, 2, 3, or 4. — TCK is the prescaled filter clock period, which is the IRC clock prescaled to the IFCP value specified in IFCPR. n accounts for the uncertainty factor in filter period calculation.

3. 修改 IcuChannel_0：

Icu_CheckNofi_S32G399A_M7->someId(…)->Icu(…)->Icu->IcuChannel-> IcuChannel_0->General：

- IcuChannelRef = /Icu/Icu/IcuConfigSet/IcuSiul2_0/IcuSiul2Channels_0 //从 FtmChannel_0 修改过来。
- IcuDefaultStartEdge = ICU_FALLING_EDGE //使用 GPIO 下降沿触发 Icu 事件。

4. 修改 IcuHwInterruptConfigList：

Icu_CheckNofi_S32G399A_M7->someId(…)->Icu(…)->Icu->IcuHwInterruptConfigList->IcuHwInterruptConfigList_0：

- ICU Peripheral ISR Name=IRQ_CH_0 //从 FTM1 修改过来。
- IcuIsrEnable =checked.

3.4 Platform 模块

Icu_CheckNofi_S32G399A_M7->someId(…)->Platfrom(…)-> Platfrom->Interrupt Controller-> IntCtrlConfig->IntCtrlConfig：

->PlatformIsrConfig_56：FTM1_IRGn：

- Interrupt Enabled=unchecked//关掉 FTM1 中断。
- Handler = undefined_handler

-> PlatformIsrConfig_177：SIUL1_ORED_IRQn

- Interrupt Enabled=checked
- Handler = SIUL2_1_ICU_EIRQ_SINGLE_INT_HANDLER //参考源文件：icu_ts_xxxx/include/Siul2_Icu_Ip_Irq.h：

```
#if (defined(SIUL2_1_ICU_EIRQ_SINGLE_INT))
/**
 * @brief     Interrupt handler for SIUL2 instance 1.
 * @details   Process the interrupt of SIUL2 instance 1.
```

```
 * @isr
 * @note   This will be defined only if the single interrupt mode is configured.
 */
ISR(SIUL2_1_ICU_EIRQ_SINGLE_INT_HANDLER);
#endif /* SIUL2_1_ICU_EIRQ_SINGLE_INT */
|   |-> Siul2_Icu_Ip_ProcessSingleInterrupt(1U);
|   |    |-> |-> Siul2_Icu_Ip_ReportEvents
Siul2_Icu_Ip_aChannelState[instance][hwChannel].callback(Siul2_Icu_Ip_aChannelState[instance][hwChannel].callbackParam, FALSE); //从而调用到 IcuChannel 注册的 callback 函数：User_EdgeDetect0
```

特别注意：不要注册成此中断：ISR(SIUL2_EXT_IRQ_0_7_ISR);。

3.5 主测试程序修改

C:\NXP\SW32G_RTD_4.4_4.0.2\eclipse\plugins\Icu_TS_T40D11M40I2R0\examples\EBT\S32G3\Icu_CheckNofi_S32G399A_M7\src\main.c

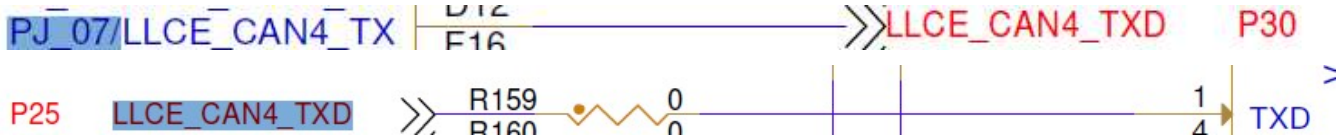
```
//johnli gpio volatile uint8 PulseCount;
void User_EdgeDetect0(void)
{
  /* increment IRQ counter */
  UserCountIrqCH0++;
    if(0xFF == UserCountIrqCH0)
    {
      UserCountIrqCH0 = 0U;
    }
}
int main(void)
{…
 //johnli gpioPulseCount = 0U;
   Icu_EnableEdgeDetection(IcuChannel_0);
  Icu_EnableNotification(IcuChannel_0);
#if 1 //johnli gpio //主函数逻辑，如果 Icu 中断触发，则操作 GPIO 翻转点灯
   while (1)
   {

          if(UserCountIrqCH0 % 2 == 0)
```

```
        {
            Dio_WriteChannel(DioConf_DioChannel_Port_pin_InputForIcuChannel0, STD_HIGH);
        }
        else
        {
            Dio_WriteChannel(DioConf_DioChannel_Port_pin_InputForIcuChannel0, STD_LOW);
        }

    delay();
  }

#else
…
#endif
```

3.6 测试结果

如之前测试方法，SW11 切换为 ON，打开 GPIO 点灯通道。

如下原理图设计：



所以将 R159 对地轻触，则会触发 GPIO151 的下降沿中断，从而调用到 ICU 注册的 Callback 函数中，导致 UserCountIrqCH0 开始向上计数，则调用 DIO 驱动翻转 GPIO6 的 LED 灯闪烁。