GPIO Driver Sample Code Reference Guide

V1.00.001

Publication Release Date: Sep. 2011

Support Chips:

ISD9160

Support Platforms:

NuvotonPlatform_Keil

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.

Table of Contents

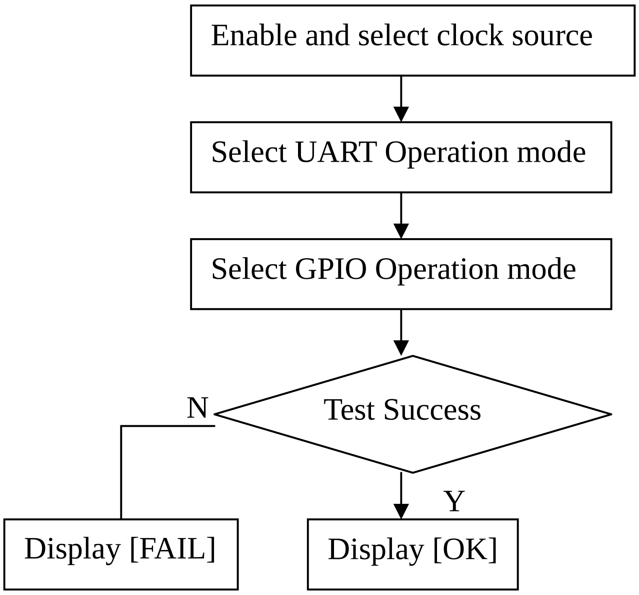
1. GPIO Driver Introduction

This sample code will demo GPIO IP on ISD9160 chip.

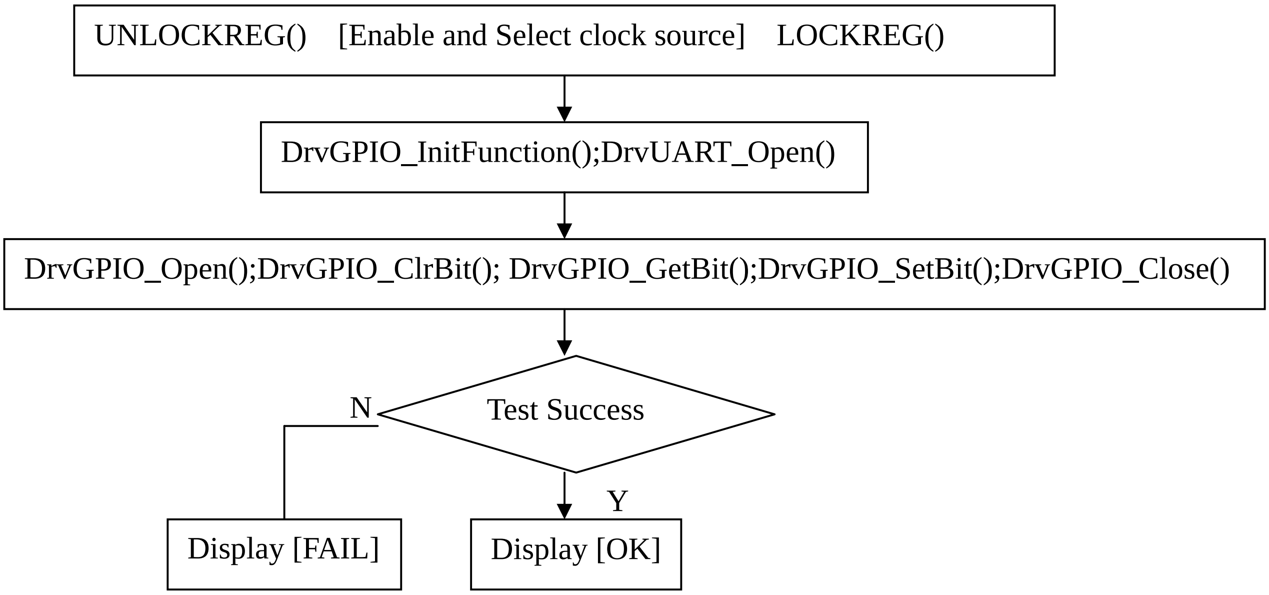
1.1 Feature

Up to 24 General Purpose I/O pins are available on the ISD91xx series. These are shared peripheral special function pins under control of the alternate configuration registers. These 24 pins are arranged in 2 ports named with GPIOA, and GPIOB. GPIOA has sixteen pins and GPIOB has eight. Each one of the 24 pins is independent and has corresponding register bits to control the pin mode function and data.

2. Block Diagram


Enable and select clock source
Select UART Operation mode
Select GPIO Operation mode
Test Success
N
Y
Display [FAIL]
Display [OK]

3. Calling Sequence


UNLOCKREG() [Enable and Select clock source] LOCKREG()
DrvGPIO_InitFunction();DrvUART_Open()
DrvGPIO_Open();DrvGPIO_ClrBit(); DrvGPIO_GetBit();DrvGPIO_SetBit();DrvGPIO_Close()
Test Success
N
Y
Display [FAIL]
Display [OK]

4. Code Section –Smpl_DrvGPIO.c

```
/*---------------------------------------------------------------------------------------------------------*/
/*
*/
/* Copyright(c) 2011 Nuvoton Technology Corp. All rights reserved.
*/
/*
*/
/*---------------------------------------------------------------------------------------------------------*/
#include <stdio.h>
#include "Driver\DrvGPIO.h"
#include "Driver\DrvUART.h"
#include "Driver\DrvSYS.h"
/*---------------------------------------------------------------------------------------------------------*/
/* MAIN function
*/
/*---------------------------------------------------------------------------------------------------------*/
int main (void)
{
        STR_UART_T param;
        int32_t i32Err;

        /* Step 1. Enable and Select clock source*/
        UNLOCKREG();
        SYSCLK->PWRCON.OSC49M_EN = 1;
        SYSCLK->PWRCON.OSC10K_EN = 1;
        SYSCLK->PWRCON.XTL32K_EN = 1;
        SYSCLK->CLKSEL0.STCLK_S = 3; //Use internal HCLK

//      SYSCLK->CLKSEL0.HCLK_S = 0; /* Select HCLK source as 48MHz */
        SYSCLK->CLKSEL0.HCLK_S = 1; /* Select HCLK source as 32KHz */
        SYSCLK->CLKDIV.HCLK_N    = 0;   /* Select no division                  */
        SYSCLK->CLKSEL0.OSCFSel = 0;    /* 1= 32MHz, 0=48MHz */
        LOCKREG();

        /* Step 2. Select UART Operation mode */
        DrvGPIO_InitFunction(FUNC_UART0);
  //      param.u32BaudRate               = 115200;
        param.u32BaudRate               = 2400;
        param.u8cDataBits              = DRVUART_DATABITS_8;
        param.u8cStopBits              = DRVUART_STOPBITS_1;
        param.u8cParity                = DRVUART_PARITY_NONE;
        param.u8cRxTriggerLevel    = DRVUART_FIFO_1BYTES;
        param.u8TimeOut                  = 0;
```

```
    DrvUART_Open(UART_PORT0, &param);

    printf("\n\n");
    printf("+----------------------------------------------------------------------+\n");
    printf("|                      GPIO Driver Sample Code                         |\n");
    printf("+----------------------------------------------------------------------+\n");

/*---------------------------------------------------------------------------------------------------------*/
/* Basic Setting                                                                                           */
/*---------------------------------------------------------------------------------------------------------*/

/*-----------------------------------------------------------------------------------------------------*/
/* Configure Bit0 in GPIOA to Output pin and Bit1 in GPIOA to Input pin then close it*/
/*-----------------------------------------------------------------------------------------------------*/
    printf("    >> Please connect GPA0 and GPA1 first <<\n");

    /* Step 3. Select GPIO Operation mode */
    DrvGPIO_Open(GPA,0, IO_OUTPUT);
    DrvGPIO_Open(GPA,1, IO_INPUT);

    DrvGPIO_ClrBit(GPA,0);

    i32Err = 0;
    printf("    GPIO Input(GPA[1])/Output(GPA[0]) test ................................ ");

    if(DrvGPIO_GetBit(GPA,1)!=0)
    {
        i32Err = 1;
    }
    DrvGPIO_SetBit(GPA,0);
    if(DrvGPIO_GetBit(GPA,1)!=1)
    {
        i32Err = 1;
    }

    DrvGPIO_Close(GPA,0);
    DrvGPIO_Close(GPA,1);

    if(i32Err)
    {
        printf("[FAIL]\n");
        printf("\n    (Please make sure GPA0 and GPA1 are connected)\n");
    }
    else
    {
        printf("[OK]\n");
```

5. Execution Environment Setup and Result

- Prepare a ISD9160 board.
- Compile the sample code.
- Connect GPA0 and GPA1.
- If the setting is right, the terminal will display [OK] not [FAIL].

6. Revision History

Version	Date	Description
V1.00.01	Sep. 2011	Created