Kneron Inc

Document Name: **Kneron GPIO Driver API**

GPIO Driver API
Kneron Inc
Engineering Design Document

Table of Contents

1 Introduction

This document describes Kneron GPIO driver API usage, there are only one GPIO controller with maximum 32 GPIO pins can be utilized. It supports pins to be configured as digital outputs, digital inputs and interrupts separately; while configured as interrupt pin, it can be set up as level trigger or edge trigger.

One thing should be noted that pin internal pull-up or pull-down is supported, however it is not part of GPIO API but PIN CONFIGURE API, please refer to the corresponding API doc for more details.

2 Reference

Kneron Mozart Design Specification, Rev. 0.5, Faraday, Feb. 2019
FTGPIO010_Block_Data_Sheet_v1.17.pdf, Faraday, October. 2017

3 Acronyms, Abbreviations, Definitions

GPIO – General Purpose Input / Output
Interrupt – A signal to the processor emitted by external hardware

4 API description

kdp_status_e kdp_gpio_init(void)

Return	Description
KDP_STATUS_OK	Successful to initialize the GPIO controller

This function initializes the GPIO controller including clock enable and interrupt handler set up.

kdp_status_e kdp_gpio_deinit(void)

Return	Description
KDP_STATUS_OK	Successful to execute the call

This function disables the corresponding clock and frees resources allocated for GPIO operations.

```
kdp_status_e kdp_gpio_set_pin_attribute (
            kdp_gpio_pin_e pin,
            uint32_t attributes)

typedef enum{
   KDP_GPIO_PIN_0,
   KDP_GPIO_PIN_1,
   KDP_GPIO_PIN_2,
   …
   KDP_GPIO_PIN_31
}kdp_gpio_pin_e;

typedef enum{
   KDP_GPIO_DIR_INPUT = 0x1,
   KDP_GPIO_DIR_OUTPUT = 0x2,
   KDP_GPIO_INT_EDGE_RISING = 0x4,
   KDP_GPIO_INT_EDGE_FALLING = 0x8,
   KDP_GPIO_INT_EDGE_BOTH = 0x10,
   KDP_GPIO_INT_LEVEL_HIGH = 0x20,
   KDP_GPIO_INT_LEVEL_LOW = 0x40,
}kdp_gpio_attribute_e;
```

Parameters	Description
pin	After configuring the desired pin as a GPIO pin, the corresponding GPIO pin name should be used as kdp_gpio_pin_e indicated
attributes	This is to specify the function of specified GPIO pin, for digital output, set only DIR_OUTPUT, for digital input for read, set only DIR_INPUT, for interrupt usage, set DIR_INPUT and one of EDGE or LEVEL trigger attributes, this implies pin is used as an interrupt input

Return	Description
KDP_STATUS_OK	Successful to execute the call

This function is to specify functions of one GPIO pin, it must be well set up before GPIO pin to be used.

kdp_status_e kdp_gpio_set_notify (
kdp_gpio_pin_e pin,
osThreadId_t tid,
uint32_t tflag)

Parameters	Description
pin	GPIO pin name, as described previously
tid	CMSIS-RTOSv2 thread ID
tflag	CMSIS-RTOSv2 thread flag

Return	Description
KDP_STATUS_OK	Successful to execute the call

If the specified pin is configured as interrupt mode, user can use this function to set thread ID and flag to get notified (awakened) while the interrupt is triggered in a wanted condition by external hardware sources.

It will not work if the pin is not configured as interrupt mode.

kdp_status_e kdp_gpio_set_interrupt_enable (
kdp_gpio_pin_e pin,
kdp_bool_e isEnable)

Parameters	Description
pin	GPIO pin name, as described previously
isEnable	Enable or not

Return	Description
KDP_STATUS_OK	Successful to execute the call

This function enables/disables the interrupt detection of the specified GPIO pin.
It should be the last call after all corresponding configuration is done for interrupt mode.

kdp_status_e kdp_gpio_set_debounce_enable (
kdp_gpio_pin_e pin,
kdp_bool_e isEnable,
uint32_t debounce_clock)

Parameters	Description
pin	GPIO pin name, as described previously
isEnable	Enable or not
debounce_clock	The debouncing clock frequency in Hz

Return	Description
KDP_STATUS_OK	Successful to execute the call

This can enable internal debouncing hardware for interrupt mode to eliminate the switch bounce.
It is very useful for connecting devices like a switch button or a keypad thing.

kdp_status_e kdp_gpio_set_pin_value (
kdp_gpio_pin_e pin,
kdp_bool_e value)

Parameters	Description
pin	GPIO pin name, as described previously
value	Output value as digital high or digital low

Return	Description
KDP_STATUS_OK	Successful to execute the call

This function writes a high or low value to a digital pin.
The specified pin must be configured as digital output.

kdp_status_e kdp_gpio_get_pin_value (
kdp_gpio_pin_e pin,
kdp_bool_e *pValue)

Parameters	Description
pin	GPIO pin name, as described previously
pValue	A pointer to data to be written to for input value

Return	Description
KDP_STATUS_OK	Successful to execute the call

This function read a high or low value from a digital pin.
The specified pin must be configured as digital input and not in interrupt mode.

5 Example code

```c
/*
 * Kneron peripheral driver test code for GPIO
 *
 * Copyright (C) 2019 Kneron, Inc. All rights reserved.
 *
 */

#include "cmsis_os2.h"
#include "test_peripheral.h"

// including kdp-system.lib API
#include "kdp_peripheral.h"

static osThreadId_t tid_user;
static void gpio_test_thread(void *argument);

#define FLAG_WAIT_GPIO_24_INTERRUPT    0x01
#define FLAG_WAIT_GPIO_25_INTERRUPT    0x02

void gpio_main(void)
{
    printf("Creating a user thread for interrupt notification\n");

    // create a thread to wait for interrupt notifications
    tid_user = osThreadNew(&gpio_test_thread, NULL, NULL);

    printf("Configuring pins to GPIO function, GPIO22(SD_CLK), GPIO23(SD_CMD), GPIO24(SD_DAT0), GPIO25(SD_DAT1)\n");

    /*
       configure selected pins as GPIO function (pinmux)
       users must check pinmux table to configure specified pins
       as desired peripheral functions properly
    */
    kdp_pinconfig_set_pin_mode(KDP_PIN_SD_CLK, PIN_MODE_3, PIN_PULL_NONE);  // as GPIO 22
    kdp_pinconfig_set_pin_mode(KDP_PIN_SD_CMD, PIN_MODE_3, PIN_PULL_UP);    // as GPIO 23
```