2DT901 : Lab 4

Group 1 : Samuel Berg & Jesper Wingren

Task 1

A

Code:

```c
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"

#define LED_R 0
#define ON 1
#define OFF 2
#define GPIO_OUT 1
#define GPIO_IN 0

int main() {
    stdio_init_all();

    gpio_init(LED_R);
    gpio_set_dir(LED_R, GPIO_OUT);

    gpio_init(ON);
    gpio_set_dir(ON, GPIO_IN);

    gpio_init(OFF);
    gpio_set_dir(OFF, GPIO_IN);

    while (1) {
        if (gpio_get(ON)) {
            gpio_put(LED_R, 1);
        } else if (gpio_get(OFF)) {
            gpio_put(LED_R, 0);
        }
    }

    return 0;
}
```

Video

B

Code:

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"

#include "hardware/regs/addressmap.h"
#include "hardware/regs/sio.h"
#include "hardware/regs/io_bank0.h"
#include "hardware/regs/pads_bank0.h"

#define LED_R 0
#define ON 1
#define OFF 2
#define GPIO_OUT 1
#define GPIO_IN 0

#define SIO_BASE_ADR 0xd0000000

int get(int gpio);
void put(int gpio, bool val);

int main() {
    stdio_init_all();

    gpio_init(LED_R);
    gpio_set_dir(LED_R, GPIO_OUT);

    gpio_init(ON);
    gpio_set_dir(ON, GPIO_IN);

    gpio_init(OFF);
    gpio_set_dir(OFF, GPIO_IN);

    while (1) {
        if (get(ON)) {
            put(LED_R, 1);
        } else if (get(OFF)) {
            put(LED_R, 0);
        }
    }

    return 0;
}

int get(int gpio) {
    uint32_t bitmask = 1 << gpio;
    uint32_t input = *(volatile uint32_t *)(SIO_BASE_ADR +
SIO_GPIO_IN_OFFSET);
    return (input & bitmask) != 0;
}

void put(int gpio, bool val) {
    if (val) {
        uint32_t bitmask = 1 << gpio;
```

```
        *(volatile uint32_t *)(SIO_BASE_ADR + SIO_GPIO_OUT_SET_OFFSET) |=
bitmask;
    }else {
        uint32_t bitmask = 1 << gpio;
        *(volatile uint32_t *)(SIO_BASE_ADR + SIO_GPIO_OUT_CLR_OFFSET) |=
bitmask;
    }
}
```

Video

C

Code:

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"

#include "hardware/regs/addressmap.h"
#include "hardware/regs/sio.h"
#include "hardware/regs/io_bank0.h"
#include "hardware/regs/pads_bank0.h"

#define LED_1 0
#define LED_2 6
#define ON 1
#define OFF 2
#define GPIO_OUT 1
#define GPIO_IN 0

#define SIO_BASE_ADR 0xd0000000

int get(int gpio);
void put(int gpio, bool val);

int main() {
    stdio_init_all();

    gpio_init(LED_1);
    gpio_set_dir(LED_1, GPIO_OUT);

    gpio_init(LED_2);
    gpio_set_dir(LED_2, GPIO_OUT);

    gpio_init(ON);
    gpio_set_dir(ON, GPIO_IN);

    gpio_init(OFF);
    gpio_set_dir(OFF, GPIO_IN);
```

```
    while (1) {
        if (get(ON)) {
            put(LED_2, 1);
            put(LED_1, 1);
        } else if (get(OFF)) {
            put(LED_1, 0);
            put(LED_2, 0);
        }
    }

    return 0;
}

int get(int gpio) {
    uint32_t bitmask = 1 << gpio;
    uint32_t input = *(volatile uint32_t *)(SIO_BASE_ADR +
SIO_GPIO_IN_OFFSET);
    return (input & bitmask) != 0;
}

void put(int gpio, bool val) {
    if (val) {
        uint32_t bitmask = 1 << gpio;
        *(volatile uint32_t *)(SIO_BASE_ADR + SIO_GPIO_OUT_SET_OFFSET) |=
bitmask;
    }else {
        uint32_t bitmask = 1 << gpio;
        *(volatile uint32_t *)(SIO_BASE_ADR + SIO_GPIO_OUT_CLR_OFFSET) |=
bitmask;
    }
}
```

Video

Task 2

Code:

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"

#define LED_1 1 // 000X     represents to 0 or 1 at idx X
#define LED_2 2 // 00X0
#define LED_4 3 // 0X00
#define LED_8 4 // X000
#define LED 25  // used for debbuging
#define INC 5
#define DEC 6
#define GPIO_OUT 1
#define GPIO_IN 0
```

```
volatile int counter = 0;

void display();
void reset();
void inc();
void dec();

int main() {
    stdio_init_all();

    // init
    gpio_init(LED);
    gpio_set_dir(LED, GPIO_OUT);

    gpio_init(LED_1);
    gpio_set_dir(LED_1, GPIO_OUT);

    gpio_init(LED_2);
    gpio_set_dir(LED_2, GPIO_OUT);

    gpio_init(LED_4);
    gpio_set_dir(LED_4, GPIO_OUT);

    gpio_init(LED_8);
    gpio_set_dir(LED_8, GPIO_OUT);

    gpio_init(INC);
    gpio_set_dir(INC, GPIO_IN);
    gpio_set_irq_enabled(INC, GPIO_IRQ_EDGE_RISE, true);
    irq_set_enabled(IO_IRQ_BANK0, true);
    gpio_add_raw_irq_handler(INC, &inc);

    gpio_init(DEC);
    gpio_set_dir(DEC, GPIO_IN);
    gpio_set_irq_enabled(DEC, GPIO_IRQ_EDGE_RISE, true);
    irq_set_enabled(IO_IRQ_BANK0, true);
    gpio_add_raw_irq_handler(DEC, &dec);

    // loop
    while (1) {
        display();

        tight_loop_contents();
    }
    return 0;
}

void display() {
    reset();
    gpio_put(LED_1, (counter & 0x01));
    gpio_put(LED_2, (counter & 0x02) >> 1);
    gpio_put(LED_4, (counter & 0x04) >> 2);
    gpio_put(LED_8, (counter & 0x08) >> 3);
```

```
}

void inc() {
    if (gpio_get_irq_event_mask(INC) & GPIO_IRQ_EDGE_RISE) {
        gpio_acknowledge_irq(INC, GPIO_IRQ_EDGE_RISE);
        if (counter < 15) {
            counter++;
        }
    }
}

void dec() {
    if (gpio_get_irq_event_mask(DEC) & GPIO_IRQ_EDGE_RISE) {
        gpio_acknowledge_irq(DEC, GPIO_IRQ_EDGE_RISE);
        if (counter > 0) {
            counter--;
        }
    }
}

void reset() {
    gpio_put(LED_1, 0);
    gpio_put(LED_2, 0);
    gpio_put(LED_4, 0);
    gpio_put(LED_8, 0);
    gpio_put(LED, 0);
}
```

Video

Task 3

Code:

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"
#include "hardware/timer.h"
#include "pico/time.h"
#include "hardware/irq.h"

#define LED_1 1 // 000X     represents to 0 or 1 at idx X
#define LED_2 2 // 00X0
#define LED_4 3 // 0X00
#define LED_8 4 // X000
#define LED 25  // used for debbuging
#define RESET 0
#define GPIO_OUT 1
#define GPIO_IN 0

volatile int counter = 0;
```

```
void display();
void gpio_cb(uint gpio, uint32_t events);
void reset();
int64_t timer_cb(alarm_id_t id, void *unused);

int main() {
    stdio_init_all();

    // init
    gpio_init(LED);
    gpio_set_dir(LED, GPIO_OUT);

    gpio_init(LED_1);
    gpio_set_dir(LED_1, GPIO_OUT);

    gpio_init(LED_2);
    gpio_set_dir(LED_2, GPIO_OUT);

    gpio_init(LED_4);
    gpio_set_dir(LED_4, GPIO_OUT);

    gpio_init(LED_8);
    gpio_set_dir(LED_8, GPIO_OUT);

    gpio_init(RESET);
    gpio_set_dir(RESET, GPIO_IN);
    gpio_set_irq_enabled_with_callback(RESET, GPIO_IRQ_EDGE_FALL |
GPIO_IRQ_EDGE_RISE, true, &gpio_cb);

    add_alarm_in_us(1000 * 1000, timer_cb, NULL, true);

    // loop
    while (1) {
        display();

        tight_loop_contents();
    }

    return 0;
}

void display() {
    reset();
    gpio_put(LED_1, (counter & 0x01));
    gpio_put(LED_2, (counter & 0x02) >> 1);
    gpio_put(LED_4, (counter & 0x04) >> 2);
    gpio_put(LED_8, (counter & 0x08) >> 3);
}

void gpio_cb(uint gpio, uint32_t events) {
    if (gpio == RESET && events & GPIO_IRQ_EDGE_FALL) {
        counter = 0;
    }
```

```
}

void reset() {
    gpio_put(LED_1, 0);
    gpio_put(LED_2, 0);
    gpio_put(LED_4, 0);
    gpio_put(LED_8, 0);
    gpio_put(LED, 0);
}

int64_t timer_cb(alarm_id_t id, void *unused) {
    if (counter < 15) {
        counter++;
    }
    return 1000 * 1000;
}
```

Video