241213

생성자	하
생성 일시	@2024년 12월 13일 오전 9:05
태그	Firmware

GPIO 입력제어

- **풀업 저항**: 3.3v 일 경우 대게 10kohm(47k)정도 쓴다

풀업 저항을 사용하여 버튼을 연결할 때, 버튼이 눌리지 않으면 입력이 **High** 상태로 유지되고, 버튼을 누르면 입력이 **Low** 상태로 바뀌게 됩니다.

  - **스위치가 눌리지 않았을 때**: 논리 1 (High) 상태
  - release (high-impedance) : z = 1
  - **스위치가 눌렸을 때**: 논리 0 (Low) 상태
  - push = 0
  - 스위치가 **눌리지 않으면** 상위 전압(3.3V 또는 5V)으로 끌어올려지고, **눌리면** 하위 전압(0V)으로 내려갑니다.
- **풀다운 저항**:

풀다운 저항을 사용하여 버튼을 연결할 때, 버튼이 눌리지 않으면 입력이 **Low** 상태로 유지되고, 버튼을 누르면 입력이 **High** 상태로 바뀌게 됩니다

  - **스위치가 눌리지 않았을 때**: 논리 0 (Low) 상태
  - release(high-impedance) : z =0
  - **스위치가 눌렸을 때**: 논리 1 (High) 상태
  - push=1

- 스위치가 **눌리지 않으면** 하위 전압(0V)으로 끌어내려지고, **눌리면** 상위 전압(3.3V 또는 5V)으로 올라갑니다.

버튼인식

-inter lock

```
void Main(void)
{
    Sys_Init();
    Uart_Printf("KEY Input Toggling #1\n");


    // KEY[1:0], GPB[7:6]을 GPIO 입력으로 선언
    GPIOB->CRL = ((GPIOB->CRL) & ~(0xff<<24))|(0x44<<24);

    int interlock = 1;


    for(;;)
    {
        // KEY0가 눌릴때마다 LED0의 값을 Toggling

        // if(Macro_Check_Bit_Clear(GPIOB->IDR,6))
        // {
        //  Macro_Invert_Bit(GPIOB->ODR,8);
        // }
        if((interlock !=0) &&Macro_Check_Bit_Clear(GPIOB->IDR
        {
            Macro_Invert_Bit(GPIOB->ODR,8);
            interlock = 0;
        }
        else if((interlock ==0) &&Macro_Check_Bit_Set(GPIOB->
        {

            interlock = 1;
        }
```

```
    }
}
```

-chattering 문제를 해결하기위해

schmitt Trigger Input

&

sw적인 chattering 문제해결

연속된 신호의 변화를 일정 시간 간격으로 샘플링하여, 일정 시간이 지난 후 신호를 확인. 여러 번 발생한 신호 변화는 무시하고, 한 번의 안정된 상태만을 기록

```
#include "device_driver.h"

void Key_Poll_Init(void)
{
    Macro_Set_Bit(RCC->APB2ENR, 3);
    Macro_Write_Block(GPIOB->CRL, 0xff, 0x44, 24);
}

/* 0: 디버깅용 설정 */
/* 1: 정상 동작용 설정 */

static int Key_Check_Input(void)
{
    return Macro_Extract_Area(~GPIOB->IDR, 0x3, 6);
}

int Key_Get_Pressed(void)
{
    unsigned int i, k;

    for(;;)
    {
```

```
        k = Key_Check_Input();

        for(i=0; i<N; i++)
        {
            if(k != Key_Check_Input())
            {
                break;
            }
        }

        if(i == N) break;
    }

    return k;
}

void Key_Wait_Key_Released(void)
{
    while(Key_Get_Pressed());
}

int Key_Wait_Key_Pressed(void)
{
    int k;

    while((k = Key_Get_Pressed()) == 0);
    return k;
}
```

버튼

```
#include "device_driver.h"
```

```
void Key_Poll_Init(void)
{
    Macro_Set_Bit(RCC->APB2ENR, 3);
    Macro_Write_Block(GPIOB->CRL, 0xff, 0x44, 24);
}

int Key_Get_Pressed(void)
{
    return Macro_Extract_Area(~GPIOB->IDR, 0x3, 6);
}

void Key_Wait_Key_Released(void)
{
    while( !(Key_Get_Pressed() == 0) );

    // while( Key_Get_Pressed() != 0 );

    // for(;;)
    // {
    //  if( Key_Get_Pressed() == 0 ) return;
    // }
}

int Key_Wait_Key_Pressed(void)
{
    int r;
    while( !((r = Key_Get_Pressed()) != 0) );
    return r;

    // for(;;)
    // {
    //  int r = Key_Get_Pressed();
    //  if(r != 0) return r;
    // }
}
```

11.UART & RS232 통신

OSI 7계층 모델

- **응용 계층 (Application Layer)**
  - **기능**: 사용자와 직접 상호작용하는 계층으로, 네트워크 서비스나 애플리케이션을 제공합니다. 데이터가 사용자와 상호작용할 수 있도록 최종적인 서비스를 제공합니다.
  - **예시**: HTTP, FTP, SMTP, DNS.
- **표현 계층 (Presentation Layer)**
  - **기능**: 데이터의 형식, 표현 방식, 인코딩 등을 다루며, 송수신하는 데이터의 형식을 변환하거나 압축, 암호화 등의 처리를 합니다.
  - **예시**: JPEG, ASCII, SSL/TLS.
- **세션 계층 (Session Layer)**
  - **기능**: 애플리케이션 간의 통신 세션을 설정, 유지, 종료합니다. 또한, 데이터 교환을 동기화하고 관리하는 역할을 합니다.
  - **예시**: NetBIOS, RPC(Remote Procedure Call).
- **전송 계층 (Transport Layer)**
  - **기능**: 두 호스트 간의 데이터 전송을 관리합니다. 데이터의 오류 검출 및 수정, 흐름 제어, 데이터 세그먼트화 등을 처리합니다.
  - **예시**: TCP(Transmission Control Protocol), UDP(User Datagram Protocol).
- **네트워크 계층 (Network Layer)**
  - **기능**: 패킷을 출발지에서 목적지까지 라우팅하고, 논리적인 주소(IP 주소)를 사용하여 네트워크 간의 경로를 설정합니다.
  - **예시**: IP(Internet Protocol), 라우터.
- **데이터 링크 계층 (Data Link Layer)**
  - **기능**: 데이터를 물리적으로 전달할 때 오류를 처리하고, 데이터 프레임을 구성하여 상대방과 신뢰성 있는 연결을 제공합니다. 물리 계층과 네트워크 계층 사이에서 오류를 감지하고 수정합니다.
  - **예시**: MAC 주소, 스위치, 이더넷 프로토콜.

- **물리 계층 (Physical Layer)**
  - **기능**: 데이터가 실제로 전송되는 물리적 매체를 다룹니다. 전압, 케이블, 커넥터와 같은 하드웨어적 요소들이 포함됩니다.
  - **예시**: 이더넷 케이블, 광섬유, 무선 신호 등.

uart 근거리 저속 저량 통신

비동기 직렬 통신 장치

modulater

demodulater　　modem

RS232

data terminal equipment DTE 데이터단말장치　master manager

data communication Equipment DCE 데이터회선종단장치 slave subordinate