임베디드 시스템 설계 및 실험

004분반 – 2조 - 3주차

실험 보고서

**GPIO 조작**

|  |  |
| --- | --- |
| **실험자** | 202055606 주우성  202055623 허치영  202255632벌드 바타르 아마르투브신  201724637 오치어 자미안퓨레브 |
| **실험날짜** | 2024-09-19 |
| **제출날짜** | 2024-09-20 |

**1. 실험 제목**

GPIO 조작

**2. 실험 목적**

* 임베디드 시스템의 기본 원리 습득
* 레지스터와 주소 제어를 통한 임베디드 펌웨어 개발 이해

**3. 세부 목표**

* 개발 환경 구축
* IAR Embedded Workbench에서 프로젝트 생성 및 설정
* Datasheet 및 Reference Manual을 참고하여 해당 레지스터 및 주소에 대한 설정 이해
* GPIO(general-purpose input/output)를 사용하여 LED제어
* 오실로스코프에 대한 이해와 DebugPin 설정

**3. 실험 장비**

* STM32F107VCT6
* IAR Embedded Workbench (EW)
* 오실로스코프

**4. 세부 실험 내용**

**1. 레지스터 및 주소에 대한 설정 이해**

Schematic 참고하여 LED 1,2,3,4 KEY 1, 2, 3, 4 포트 확인

|  |  |
| --- | --- |
| A diagram of a led  Description automatically generated | A diagram of a button  Description automatically generated |

Datasheet 참고하여 Memory Map, base address 확인

* Port A: 0x4001 0800
* Port B: 0x4001 0C00
* Port C: 0x4001 1000
* Port D: 0x4001 1400
* RCC: 0x4002 1000

Reference Manual 참고하여 offset 확인

* APB2 peripheral clock enable register (RCC\_APB2ENR): 0x18
* Port configuration register low (GPIOx\_CRL) (x=A..G): 0x00
* Port configuration register high (GPIOx\_CRH) (x=A..G): 0x04
* Port input data register (GPIOx\_IDR) (x=A..G): 0x08
* Port bit set/reset register (GPIOx\_BSRR) (x=A..G):0x10
* Port bit reset register (GPIOx\_BRR) (x=A..G): 0x14

**2. IAR EW에서 프로젝트 생성 후 설정**

**3. 버튼을 이용한 LED 제어 (main.c 작성)**

시스템 초기 설정

* KEY1,2,3,4: Input with pull-up / pull-down 모드 설정
* Port A, B, C, D 클럭 인가 (RCC\_APB2ENR)
* LED 끄기: GPIOD\_BSRR로PD2, PD3, PD4, PD7 ODR bit set

while loop 로직

* KEY1,2,3,4 input 감지 (GPIOx\_IDR)
* KEY1, 3 감지 시
  + GPIOx\_BRR 로 PD2, PD3 / PD4, PD7 ODR bit reset (켜기)
* KEY2,4 감지 시
  + GPIOx\_BSRR로 PD2, PD3 / PD4, PD7 ODR bit set (끄기)

**4. 정상 동작 유무 확인**

**5. 오실로스코프를 이용한 디버깅**

LED1 (PD2) -> D0 연결

KEY1 (PC4) -> D1 연결

Digital pin의 trigger 이용 KEY1 눌렀을 때 D0, D1 신호 확인

**4. 실험 결과**

**1. 버튼과 LED의 상호작용 및 정상 동작 여부**

|  |  |  |  |
| --- | --- | --- | --- |
| A hand holding a green light on a circuit board  Description automatically generated | A hand holding a circuit board  Description automatically generated | A hand holding a green light on a black surface  Description automatically generated | A hand holding a circuit board  Description automatically generated |
| KEY1  LED1 / LED2 ON | KEY2  LED1 / LED2 OFF | KEY3  LED3 / LED4 ON | KEY4  LED3 / LED4 OFF |

**2. 오실로스코프 측정 결과**

A screen with colorful lines on it

Description automatically generated

KEY1 눌렀을 때 D1입력 신호 가해진 직후 LED1(D0) 신호 발생

KEY1 에서 손을 뗀 후 D1 신호 없어짐

LED1 신호는 계속 유지

**5. 분석 및 고찰**

**레지스터 및 주소 설정**

임베디드 시스템에서는 하드웨어 자원을 제어하고 상호작용하기 위해 레지스터와 메모리 매핑 주소를 사용하여 데이터를 조작한다. CPU는 메모리 맵을 따라 각 레지스터의 주소를 참조하고, 해당 비트를 읽거나 쓰는 방식으로 하드웨어 동작을 제어한다. 이 과정에서 각 장치나 포트는 고유한 메모리 주소를 가지고 있으며, 레지스터를 통해 데이터를 주고받는다. 따라서 해당 보드에서 레지스터와 주소가 어떻게 구현되어 있는지 명확히 이해하는 것이 필요하다.

**정확한 레지스터 설정의 중요성**

이번 실험 과정 중에Port A 의 base address 값을 0x4001 0900 으로 잘못 설정하는 실수를 범했다. 그 결과 Key3을 눌렀다 뗐을 때KEY4를 누르지 않았음에도 LED3,4가 바로 꺼지는 현상이 발생했다. while loop 로직에서 GPIOA\_IDR 값을 확인하는 과정이 있는데, 잘못된 주소의 값을 확인하여 KEY4가 항상 눌러져 있는 것과 같은 효과가 난 것으로 추측된다. 간단한 실험이었기에 비교적 쉽게 디버깅을 할 수 있었지만 복잡한 시스템에서는 이러한 행운을 기대하기 어려울 것이다. 이러한 실수를 통해 임베디드 시스템에서 정확한 레지스터 설정의 중요성을 체감하였으며 이를 위해 Reference Manual 및 Datasheet를 철저히 참고하는 습관을 길러야 할 것이다.

**GPIO(general-purpose input/output)**

GPIO는 임베디드 시스템에서 외부 장치와 상호작용하기 위한 가장 기본적인 입출력 제어 방법을 제공한다. 이를 통해 센서 데이터의 수집, LED 제어, 버튼 입력 처리 등 다양한 작업을 수행할 수 있다. 이번 실험에서는 KEY와 연결된 포트를 pull-up / pull-down mode로 설정하여 입력 신호를 받고 LED와 연결된 포트를 output mode로 설정하여 KEY입력으로 LED를 제어하였다.

**오실로스코프**

오실로스코프는 신호의 변화를 시각적으로 확인할 수 있는 장비로서, 전기 신호의 변화를 정확히 분석해야 하는 임베디드 시스템 설계 및 실험에 필수적인 장비임을 확인했다. 이번 실험에서는 KEY1을 눌렀을 때 D1 입력 신호가 발생하고, 곧바로 LED1의 D0 신호가 변화하는 것을 통해, 버튼 입력과 LED 제어가 실시간으로 상호작용하는 것을 확인할 수 있었다.

**6. 참고자료**

* STM32107VCT6 schematic
* stm32 Datasheet
* stm32 Reference Manual