정적 분석을 이용한 인공심장 소프트웨어 분석

조성근, 이원찬

2010. 4. 12.

소프트웨어 무결점 연구센터, 서울대학교

발표 순서

1. 연구 목표
2. 분석 결과
   - Buffer overrun 경고
   - Divided by zero 경고
3. 분석 방법 / 분석기
   - 요약 해석(abstract interpretation)
   - Sparrow : C 프로그램 분석기
4. 결론 / 앞으로의 과제

연구 목표

- 인공심장 소프트웨어의 신뢰도 높이기
- 일반적인 프로그램 오류 분석
  - Buffer overrun `array[index]` `index > array_size`
  - Divided by zero `num / divider` `divider = 0`
- 프로그램의 기능성(functionality) 분석
  - 소프트웨어가 실행 중에 만족해야 하는 조건

분석 결과

- Buffer overrun
  - 2개의 참 경고(true alarm)
    (실제 프로그램 실행에서 발생할 수 있는 오류에 대한)
- Divided by zero
  - 2개의 참 경고
  - 1개의 허위 경고(false alarm)

Buffer Overrun

CTRL_Var.h

```
 170     Uint16  HS_Group[7] = {0, 5, 3, 4, 1, 0, 2};
```

CTRL_Sub.c

```
 252 void Position_Check(void)
 253 {
 257   // HSI_Pattern = GpioDataRegs.GPEDAT.all & 0x0007;
       // 0000 0000 0000 0 HA HB HC
 258   HSI_Pattern = rand() % 8;
 259
 260   Cur_HSI_SEQ = HS_Group[HSI_Pattern];
```

모터의 위치

배열 크기: 7

배열 접근: 0~7

Divided by Zero

CTRL_Var.h

```
 136     double  MC_Sample_Counter = 0;
```

CTRL_Sub.c

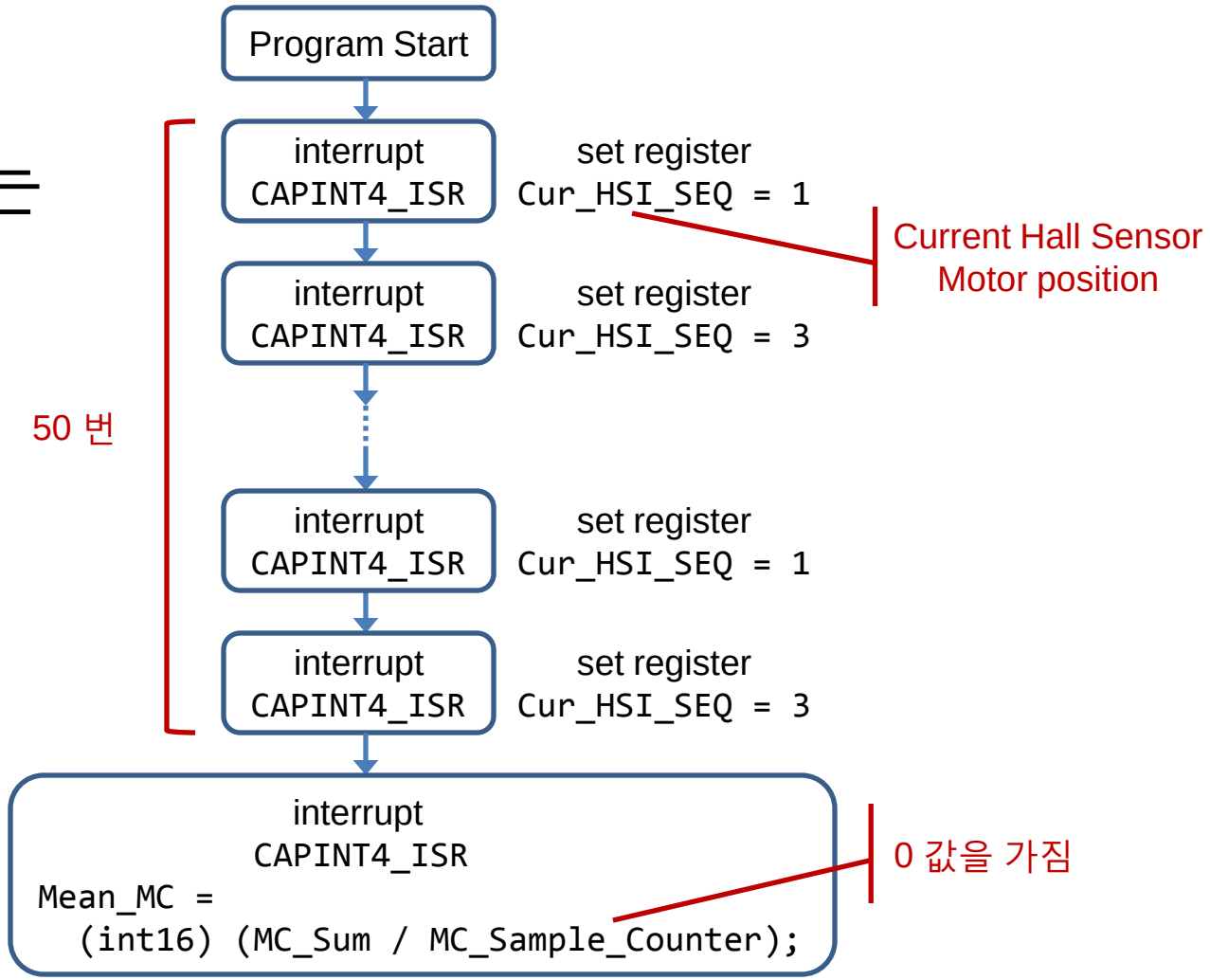
```
   1 void CAPTURE_Process(void)
   2 {
  96   switch( CHKBIT(Control_Flag, BIT02) ) {
  97     case B2SET:     // DIR->LEFT
 100       if( Current_Position >= Left_End_Position ) {
 110         Mean_MC = (int16) (MC_Sum / MC_Sample_Counter);

 825 void AD_Sampling(void)
 826 {
 864     MC_Sample_Counter++;
```

가질 수 있는 값: 0 ~ oo

Divided by Zero

- 오류를 발생시키는 시나리오



분석 방법 / 분석기

- 요약 해석 (abstract interpretation)
  - 프로그램의 실행의미를 안전하게 요약하는 이론

- 분석기: Sparrow*
  - C 프로그램을 정적으로, 자동으로 분석
  - buffer overrun, divided by zero,
    null dereference, use after free, double free,
    memory leaks, file handle leaks

* http://www.spa-arrow.com/

Sparrow 데모

결론 / 앞으로의 과제

- 찾아낸 소프트웨어의 오류
  - 2개의 buffer overrun
  - 2개의 divided by zero
- 프로그램의 기능성 분석
  - 소프트웨어 개발자의 도움 필요
- 인공심장 소프트웨어의 무결점 검증
  - 인공심장 소프트웨어의 신뢰도 향상

감사합니다.