

Xilinx Zynq FPGA, TI DSP, MCU

프로그래밍 및 회로 설계 전문가 과정

김현, 최준호

LiDAR App 문서화

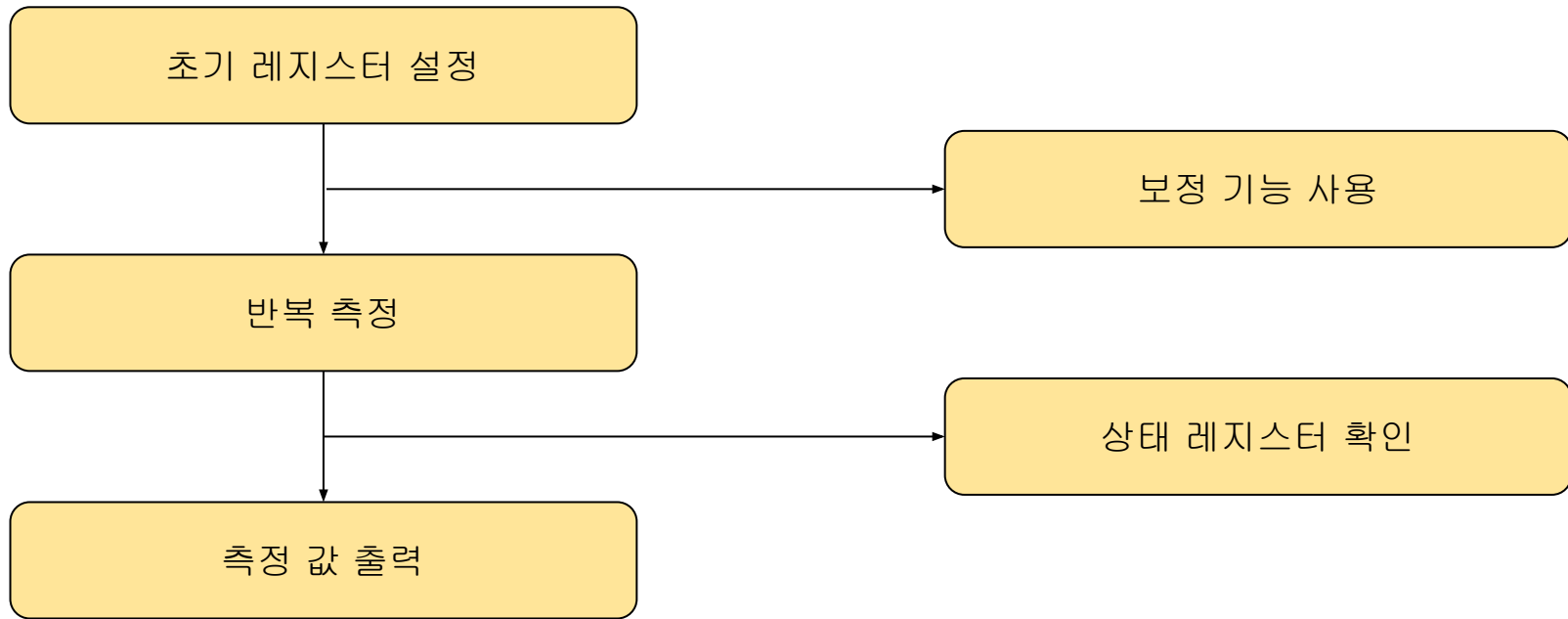
목차

- LiDAR Lite v3
- 앱 코드 흐름 설명
- 초기 레지스터 설정
- 보정 기능 설정
- 상태 레지스터 확인
- 측정 값 출력

LiDAR App

LiDAR Lite v3

Description LiDAR App Flow



초기 레지스터 설정

Register Definitions Control Register List

Address	R/W	Name	Description	Initial Value
0x00	W	ACQ_COMMAND	Device command	--
0x01	R	STATUS	System status	--
0x02	R/W	SIG_COUNT_VAL	Maximum acquisition count	0x80
0x04	R/W	ACQ_CONFIG_REG	Acquisition mode control	0x08
0x09	R	VELOCITY	Velocity measurement output	--
0x0c	R	PEAK_CORR	Peak value in correlation record	--
0x0d	R	NOISE_PEAK	Correlation record noise floor	--
0x0e	R	SIGNAL_STRENGTH	Received signal strength	--
0x0f	R	FULL_DELAY_HIGH	Distance measurement high byte	--
0x10	R	FULL_DELAY_LOW	Distance measurement low byte	--
0x11	R/W	OUTER_LOOP_COUNT	Burst measurement count control	0x01
0x12	R/W	REF_COUNT_VAL	Reference acquisition count	0x05
0x14	R	LAST_DELAY_HIGH	Previous distance measurement high byte	--
0x15	R	LAST_DELAY_LOW	Previous distance measurement low byte	--
0x16	R	UNIT_ID_HIGH	Serial number high byte	Unique
0x17	R	UNIT_ID_LOW	Serial number low byte	Unique
0x18	W	I2C_ID_HIGH	Write serial number high byte for I2C address unlock	--
0x19	W	I2C_ID_LOW	Write serial number low byte for I2C address unlock	--
0x1a	R/W	I2C_SEC_ADDR	Write new I2C address after unlock	--
0x1c	R/W	THRESHOLD_BYPASS	Peak detection threshold bypass	0x00

0x02 : 피크를 찾기 위해
신호를 쏘고 받는 최대
횟수를 정한다.(범위 $n^{(1/4)}$)

0x04 : 3번 비트가 0이면
신호 피크가 최대일 것이라고
예상될 때 빠르게 종료하여
성능은 유지하면서도 빠른
측정이 가능(덜 정확)

0x1C : 감도를 줄이고
높인다.

default는 신호 세기, 잡음,
피크 값을 기본 설정으로
한다.

보정 기능 설정

Address	R/W	Name	Description	Initial Value
0x00	W	ACQ_COMMAND	Device command	--
0x01	R	STATUS	System status	--
0x02	R/W	SIG_COUNT_VAL	Maximum acquisition count	0x80

0x00 : 하드웨어 초기화와
거리 측정 시 **bias**를 보정
할건지 말건지 정하는
레지스터

```
while(1) {  
    measurement(CORRECTION, options, receives);  
    for(i=0; i<99; i++) measurement(NO_CORRECTION, options, receives);
```

```
if(is_correction) i_write(ACQ_COMMAND, 0x04);  
else i_write(ACQ_COMMAND, 0x03);
```

100번 중 1번은 값을 보정해서 받고 나머지
99번은 그냥 받는 것이 예제 코드의 내용

반복 측정

```
27 #define AR_VELOCITY 0
28 #define AR_PEAK_CORR 1
29 #define AR_NOISE_PEAK 2
30 #define AR_SIGNAL_STRENGTH 3
31 #define AR_FULL_DELAY_HIGH 4
32 #define AR_FULL_DELAY_LOW 5
```

repeated start 방식이 아닌

라이다 자체적인 프로토콜에 의해
연속되는 여러 개의 값들을 가져온다.

0x09	R	VELOCITY
0x0c	R	PEAK_CORR
0x0d	R	NOISE_PEAK
0x0e	R	SIGNAL_STRENGTH
0x0f	R	FULL_DELAY_HIGH
0x10	R	FULL_DELAY_LOW

100번 중 1번은 값을 보정해서 받고 나머지
99번은 그냥 받는 것이 예제 코드의 내용

상태 레지스터 확인

```
unsigned get_status() {  
    return buf[0] & 0x01;  
}
```

```
while(busy_flag) {  
    busy_flag = get_status();  
    busy_counter ++;  
    if(busy_counter > 9999) {  
        printf("BUSY COUNT TIME OUT !\n");  
        return ;  
    }  
}  
if(!busy_flag) {
```

상태 레지스터의 값을 확인하여 현재 라이다가 수신한 신호를 처리하는 등의 이유로 바쁜지 확인하고 바쁘다면 측정 값 요청을 대기한다.

0x01

R/W	Name	Description	Initial Value
R	STATUS	System status	--

Bit	Function
6	Process Error Flag 0: No error detected 1: System error detected during measurement
5	Health Flag 0: Error detected 1: Reference and receiver bias are operational
4	Secondary Return Flag 0: No secondary return detected 1: Secondary return detected in correlation record
3	Invalid Signal Flag 0: Peak detected 1: Peak not detected in correlation record, measurement is invalid
2	Signal Overflow Flag 0: Signal data has not overflowed 1: Signal data in correlation record has reached the maximum value before overflow. This occurs with a strong received signal strength
1	Reference Overflow Flag 0: Reference data has not overflowed 1: Reference data in correlation record has reached the maximum value before overflow. This occurs periodically
0	Busy Flag 0: Device is ready for new command 1: Device is busy taking a measurement

측정 값 출력

```
150 void display(unsigned char options, unsigned char *buf) {
151     unsigned char i;
152     char* strings[5] = {"Velocity", "Peak value in correlation record"
153         , "Correlation record noise floor", "Received signal strength", "Distance"};
154     buf[AR_FULL_DELAY_HIGH] = buf[AR_FULL_DELAY_HIGH] << 8 | buf[AR_FULL_DELAY_LOW];
155
156     /*
157         AR_VELOCITY           0
158         AR_PEAK CORR         1
159         AR_NOISE PEAK        2
160         AR_SIGNAL STRENGTH   3
161         AR_FULL_DELAY_HIGH   4
162         AR_FULL_DELAY_LOW    5
163     */
164
165     switch(options) {
166         case OUTPUT_OF_ALL :
167             for(i=0; i<5; i++) printf("%s \t\t\t\t = %d\n", strings[i], buf[i]);
168             break;
169         case DISTANCE_ONLY :
170             printf("%s \t\t\t\t = %d\n", strings[4], buf[AR_FULL_DELAY_HIGH]);
171             break;
172         case DISTANCE_WITH_VELO :
173             printf("%s \t\t\t\t = %d\n", strings[0], buf[AR_VELOCITY]);
174             printf("%s \t\t\t\t = %d\n", strings[4], buf[AR_FULL_DELAY_HIGH]);
175             break;
176         case VELOCITY_ONLY :
177             printf("%s \t\t\t\t = %d\n", strings[0], buf[AR_VELOCITY]);
178             break;
179     }
180     printf("\n");
181 }
```

출력된 값들 보여줄
부분

옵션에 따라 여러 값들을 출력한다.