Firmware Specification

**GEN2, allears**

**Ver 0.1**

# **Revision History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **History** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **Preface**

이 문서는 allears Gen2 프로젝트 Firmware 명세서이다.

# **Terms and Definition**

## **Overview**

* 이 문서에서 사용되는 용어들을 정의한다.

## **Terms**

|  |  |
| --- | --- |
| **Term** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **Introduction**

1. Gen2는 allears의 2세대 버전이다.
2. 해당 실험 장치는 미주신경계를 자극하는 실험용 장치다.
3. UART로 명령을 하여 자극의 세기나 설정 값을 변경하고 상태를 확인한다.
4. 실험용 장치의 메인 컨트롤러는 Nucleo-64 형식의 STM32 L412RB MCU를 사용한다.

# **Version Format**

## **Format**



## **Number Meaning**

1. Major number
   * Numbering in order
2. Minor number

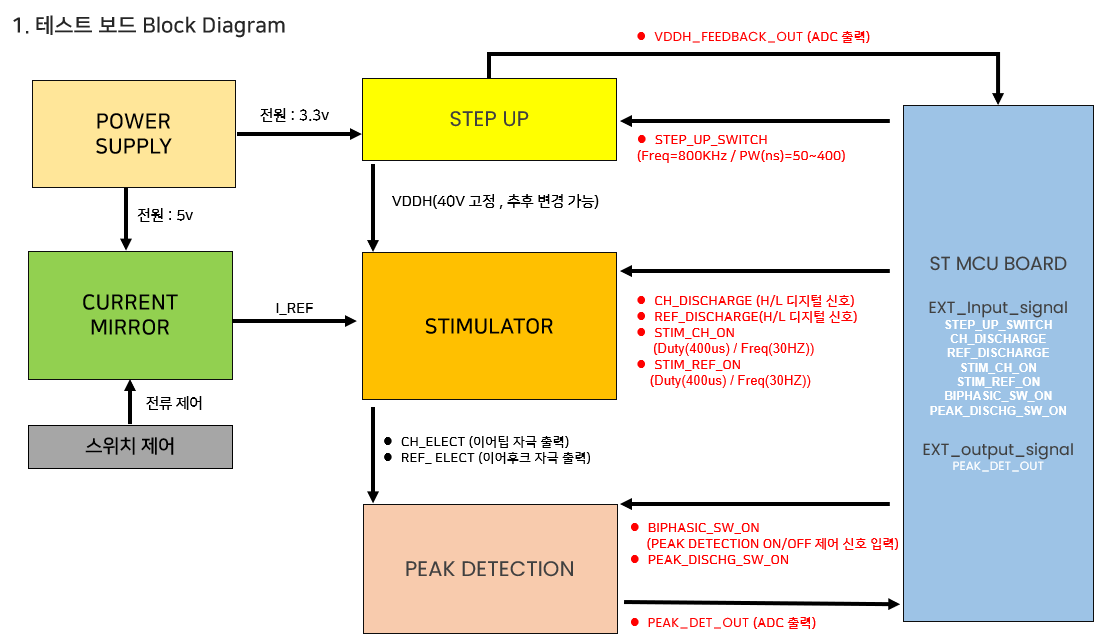
* Numbering in order

1. Subminor number
   * Numbering in order
2. ASCII ‘B’ or ‘R’
3. B: Beta version. Test or experimental version
4. R: Release version Officially

# **Basic Concept**

1. PC와 메인 MCU는 UART 통신으로 데이터를 주고받는다.
2. PC와 주고받은 데이터를 기반으로 MCU는 자극 회로를 제어한다.

# **Hardware Basic Diagram**

텍스트, 장치, 측정기, 게이지이(가) 표시된 사진

자동 생성된 설명

1. ST MCU 보드는 STEP UP에 PWM 출력의 전압을 제어하며 전압 데이터 값을 피드백 받는다.
2. ST MCU 보드는 STIMULATOR에 자극 파형을 출력한다.
3. STIMULATOR의 파형 출력이 PEAK DETECTION 블록의 입력으로 들어가며 PEAK DETECTOIN 블록에서 피크 값을 ST MCU 보드에 보내준다.
4. ST MCU 보드는 PEAK DETECTION에서 나오는 피크의 출력 유무를 제어할 수 있다.

# **State Diagram**



* INIT 상태
* 시스템에서 전원을 넣고 넘어오는 상태다.
* 해당 상태에서는 내부 메모리에 저장한 데이터를 불러오며, 이를 실패할 경우 기본 값으로 설정하게 된다.
* IDLE 상태
* INIT 상태가 끝나고 아무것도 하지 않는 상태다.
* 버튼 입력 이벤트 또는 UART 시작 커맨드로 RUN 상태에 진입할 수 있다.
* UART 명령어를 통해 파형의 주기, 펄스 폭, Dead Time을 **설정하고 저장하며, 볼 수 있다.**
* RUN 상태
* 자극 파형이 나오는 상태다.
* 버튼 입력 이벤트 또는 UART 정지 명령어로 IDLE 상태에 진입할 수 있다.
* **설정한 값을 볼 수 있으며, 해당 상태에서는 값 설정변경이 불가능하다.**
* ERROR 상태
* 칩 자차에서 발생한 오류로 인해 아무런 동작이 되지 않는 상태다. 해당 상태에서는 시스템 리셋을 실행한다.

# **Electrical Stimulation**

## **Concepts**



## **Electrical Stimulation Parameters**



1. 하나의 자극 펄스는 하나의 Cathode와 하나의 Anode로 이루어진다.
2. 1s / Period = stimulation pulse Frequency
3. Cathode와 Anode의 Pulse width는 같다.
4. Cathode가 먼저 출력 되고, Dead Time 이후에 Anode를 출력한다.
5. 전기 자극이 출력되는 동안에는 충전 전압을 방전하고, 전기 자극이 멈추는 동안에 전압을 충전한다.

## **Electrode Charging**

1. 전기 자극에서 사용되는 전압을 충전한다.
2. 충전되는 전압은 일정하게 유지되어야 한다.
3. Control points

|  |  |
| --- | --- |
| **Control points** | **Description** |
| volt\_ctrl | 전압 수준을 제어 |
| monitor\_volt | 충전된 전압 수준을 확인/현재 전압 출력 수준 |
| stimul\_volt | 전기 자극으로 출력되는 전압. 방전시 전압 출력 목표 전압 |

## **Electrical Stimulation**

1. 전기 자극의 수준들 제어한다.
2. 출력되는 전기 자극의 정도는 착용 상태를 확인하는데 사용된다.
3. Control points

|  |  |
| --- | --- |
| **Control points** | **Description** |
| ca\_ctrl | cathode 전극의 파형 출력을 제어 |
| an\_ctrl | anode 전극의 파형 출력을 제어 |
| cathode | 출력 되는 전기 자극의 전극의 하나 |
| anode | 출력 되는 전기 자극의 전극의 하나 |

## **Electrical Stimulation Level**

|  |  |  |
| --- | --- | --- |
| **자극 레벨** | **출력 전압 (V)** | **펄스 폭(us)** |
| Level 0 | 0 | 0 |
| Level 1 | 40 | 50 |

## **Peak Detection**

1. 자극의 피크 감지기능의 작동 유무를 제어한다.
2. 출력된 피크의 정보를 기반으로 임피던스를 측정하는데 사용한다.
3. Control points

|  |  |
| --- | --- |
| **Control points** | **Description** |
| peak\_val | 입력되는 Peak 값 |
| peak\_pwr\_on | Peak 감지 기능의 전원부를 제어 |
| peek\_on | Peak 감지 기능의 출력부를 제어 |

## **Peak Detection Level**

|  |
| --- |
| **Voltage (V)** |
| 0 ~ 42 (여유2V 줌) |

# **Electrical Stimulation Control Command**

자극 관련 명령어와 응답 메시지를 UART를 통해 송수신한다.

## **메지지 및 명령어 규칙**

1. 모든 명령어는 접두어로 “#”을 사용한다.
2. 시스템의 중요한 명령어 5개는 “#”을 제외한 접두어가 없다.

* 시작, 정지, 저장, 공장초기화, 버전정보

1. 값을 세팅할 때는 접두어로 “#set” 가 붙는다.

* Ex) #setDT

1. 설정한 값을 읽어올 때는 접두어로 “#get” 가 붙는다.

* Ex) #getDT

1. #set, #get에 대한 응답 메시지는 접두어로 “#res”가 붙어 나온다.

* Ex) #resDT

1. 에러 코드는 접두어로 “#Error” 가 붙으며 에러 코드번호는 “-“로 구분한다.

* Ex) #Error-1

1. 그 외 시스템 메시지는 형식이 없다.

## **Command and Response Format**

### **Admin Command**

|  |  |
| --- | --- |
| **Command type** | **Description** |
| #start | 자극 시작 명령어 |
| #stop | 자극 중지 명령어 |
| #save | 설정한 펄스 폭, 주기, Dead Time 등의 데이터를 저장하는 명령어 |
| #factoryreset | 설정된 값을 초기 값으로 설정하는 공장초기화 명령어 |
| #version | 버전 정보를 확인 |

### **Parameter Command**

|  |  |
| --- | --- |
| **Parameter type** | **Description** |
| #setDT,xxx | 입력범위 1 ~ 100us, 10us 단위  펄스와 펄스 폭 사이의 간격 시간을 설정하는 명령어  EX) #setDT,100 => Dead Time 100us 로 설정함 |
| #setPW,xxx | 입력범위 1 ~ 1000us, 100us 단위  펄스 폭을 설정하는 명령어  EX) #setPW,1000 => 펄스 폭 1000us 로 설정함 |
| #setPER,xxx | 입력범위 1 ~ 30Hz, 1단위  펄스의 주기를 설정하는 명령어  EX) #setPER,30 => 30Hz 설정함 |
| #getDT | 설정된 Dead Time 출력 |
| #getPW | 설정된 펄스 폭 출력 |
| **#**getPER | 설정된 펄스 주기 출력 |
| #getALLPRM | 설정된 모든 값 출력 |

### **Parameter Response Message**

|  |  |
| --- | --- |
| **Message type** | **Description** |
| #resDT: XXX | #setDT, #getDT 출력에 대한 형식이다.  XXX는 설정된 값 |
| #resPW: XXX | #setPW, #getPW 출력에 대한 형식이다.  XXX는 설정된 값 |
| #resPER: XXX | #setPER, #getPER 출력에 대한 형식이다.  XXX는 설정된 값 |
| #resALLPRM  DT: XXX  PW: XXX  PER: XXX | #getALLPRM 출력에 대한 형식이다.  XXX는 설정된 값 |

### **Error Codes**

|  |  |
| --- | --- |
| **Error code** | **Description** |
| #Error-1 | System Fault Exception  에러 처리: 이후 시스템을 리셋을 한다. |
| #Error-2 | 설정한 값을 내부 메모리에 저장 실패  에러 처리: 사용자가 저장을 재 시도한다. |

# **Button**

|  |  |  |
| --- | --- | --- |
| **Button Event** | **State** | **Button action** |
| Pressed | IDLE | Start stimulation |
| Pressed | RUN | Stop |
| Pressed | ERROR | No action |

# **LED Indication**

|  |  |  |
| --- | --- | --- |
| **LED** | **State** | **Indication** |
| Green | IDLE | LED OFF |
| Green | RUN | LED ON |
| Green | ERROR | LED blinks 10 times in 3sec |