|  |  |
| --- | --- |
| **Program / Project Name:** | Programming |
| **Checklist / Template Completed by:** | TBD |
| **Date Completed:** | Click here to enter a date. |

**Programming Guide**

Revision: 0.1

**Detailed Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Editor(s)** | **Description of change** |
| 0.1 | 09-JUN-2019 | Sang-Gu Kang | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1 API Library 4](#_Toc11008651)

[2 Todo 4](#_Toc11008652)

[2.1 Todo 4](#_Toc11008653)

[2.1.1 Todo 4](#_Toc11008654)

[3 Todo 4](#_Toc11008655)

[3.1 Todo 4](#_Toc11008656)

# API Library

## 문자열 처리 함수

### strcpy vs strcpy\_s

가급적 strcpy\_s, sprintf\_s, strcat\_s와 같은 안전한 스트링 함수를 사용해야 한다. Visual studio에서 strcpy와 같은 함수를 사용하면 error로 처리된다. 이 함수들의 원형을 살펴보면 두 번째 인자가 size\_t numberOfElements임에 유의한다. Sizeof(buffer)가 아니라 \_countof(buffer)로 계산해야 한다.

errno\_t strcpy\_s(char \*strDest, size\_t numberOfElements, const char \*strSrc);

안전한 스트링 함수는 strDest 버퍼를 넘어서는 메모리 복사를 방지하는 함수이다. 기존의 strcpy같은 함수를 사용할 때에는, 잘못된 메모리 할당으로 어느 순간에는 문제가 없는 것처럼 돌아가다가 나중에 다른 곳에서 메모리 할당시 엉뚱하게 문제가 발생할 수 있고, 이는 디버깅 자체를 어렵게 만든다. strcpy\_s 함수는 메모리 에러를 발생시켜 주므로 메모리를 안정적으로 사용할 수 있도록 해준다.



# 수식과 입출력

## Type Cast

### static\_cast<type>(value)

Todo

# Todo

## Todo