자율주행 자동차 변수 및 함수 이름 지침서

목차

[1. 공통 사항 2](#_Toc487920670)

[A. 방향 관련 2](#_Toc487920671)

[B. 변수 및 함수 이름 관련 2](#_Toc487920672)

[C. 변수 선언 2](#_Toc487920673)

[2. Servo Motor 3](#_Toc487920674)

[A. 변수 선언 3](#_Toc487920675)

[1. 방향 제어 3](#_Toc487920676)

[2. 기타 서보 3](#_Toc487920677)

[3. DC Motor 3](#_Toc487920678)

[A. 변수선언 3](#_Toc487920679)

[B. 구동 관련 3](#_Toc487920680)

[4. 함수 4](#_Toc487920681)

[A. 반복적인 기능, 기본 기능은 모두 함수로 만듦 4](#_Toc487920682)

[B. 함수, 제어문 및 반복문 사용시 4](#_Toc487920683)

[C. Servo 함수 5](#_Toc487920684)

[D. 초음파 함수 5](#_Toc487920685)

[E. Motor 함수 6](#_Toc487920686)

[1. 속도 6](#_Toc487920687)

[2. 기본 기능 6](#_Toc487920688)

# 공통 사항

## 방향 관련

* + 1. Left, Center, Right 등을 사용
       1. L, C, R로 앞 글자만 대문자로 사용
    2. Front, Rear, Left Side, Right Side
       1. Front, Rear로 사용
       2. L\_Side, R\_Side
    3. 방향을 앞에 표시
       1. Ex) L\_Motor, L\_Side\_Echo

## 변수 및 함수 이름 관련

* + 1. 단어간 연결은 \_(Underline)을 사용
    2. 단어의 첫 글자는 대문자 사용
    3. 변수 선언 시 무엇에 관련된 변수 인지 주석 달기
    4. 함수 이름은 하나의 단어로 구성될 경우 소문자로만 작성
    5. Set, get 등으로 시작하는 함수에는 \_없이 사용

## 변수 선언

* + 1. 변수 선언 시 같은 묶음 단위 변수 선언 후 공백
    2. 3글자 이하인 센서는 모두 대문자로 선언

# Servo Motor

## 변수 선언

### 방향 제어

* + - 1. Steer\_Servo

### 기타 서보

* + - 1. (기능명 또는 센서 이름)\_Servo
      2. Ex) UV\_Servo(적외선 센서와 관련 있는 서보)

# DC Motor

## 변수선언

* + 1. L\_Motor, R\_Motor, C\_Motor[[1]](#footnote-1)

## 구동 관련

* + 1. Forward, Backward, Velocity[[2]](#footnote-2), Stop 을 사용
    2. 각각의 앞 글자만 따서 F, B, V, S로 사용
       1. L\_Motor\_F //좌측 모터 전진
       2. R\_Motor\_B //우측 모터 후진

# 함수

## 반복적인 기능, 기본 기능은 모두 함수로 만듦

* + 1. forward(), backward(), turn\_Left(), turn\_Right(), setSpeed()[[3]](#footnote-3), getDistamce()[[4]](#footnote-4)등등

## 함수, 제어문 및 반복문 사용시

* + 1. 수정 용이성과 { } 가독성을 위해 BSD 또는 GNU 방법으로 코딩

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6 | if(...)      {          처리();      }    GNU | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6 | if(...)  {          처리();  }    BSD | [cs](http://colorscripter.com/info#e) |

## Servo 함수

* + 1. turn\_Right : 우회전
       1. turn\_Right\_R : 후진 우회전
    2. turn\_Left : 좌회전
       1. turn\_Left\_R : 후진 좌회전
    3. turn\_U

## 초음파 함수

* + 1. 거리 구하기

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9 | float / int getDistance  {      digitalWrite(TRIG,HIGH);      delayMicroseconds(10);      digitalWrite(TRIG,LOW);      distance = pulseIn(ECHO,HIGH)/58.2;        return distance;  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |

## Motor 함수

### 속도

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5 | void setSpeed(char speed)  {      analogWrite(R\_Motor\_V, speed);      analogWrite(L\_Motor\_V, speed);  } | [cs](http://colorscripter.com/info#e) |

### 기본 기능

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | void forward()  {      Steer\_Servo.write(정면 각도);      digitalWrite(L\_Motor\_F, HIGH) ;      digitalWrite(L\_Motor\_B, LOW);      digitalWrite(R\_Motor\_F, HIGH) ;      digitalWrite(R\_Motor\_B, LOW);  }    void backward()  {      Steer\_Servo.write(정면 각도);      digitalWrite(L\_Motor\_F, LOW) ;      digitalWrite(L\_Motor\_B, HIGH);      digitalWrite(R\_Motor\_F, LOW);      digitalWrite(R\_Motor\_B, HIGH);  }    void stop()  {      digitalWrite(L\_Motor\_F, HIGH) ;      digitalWrite(L\_Motor\_B, HIGH);      digitalWrite(R\_Motor\_F, HIGH);      digitalWrite(R\_Motor\_B, HIGH);  } | [cs](http://colorscripter.com/info#e) |

1. 모터 하나로 두 바퀴를 제어 할 경우 [↑](#footnote-ref-1)
2. Speed, Stop 모두 S로 시작해 혼동을 막기 위해 변수 이름엔 Velocity 사용 [↑](#footnote-ref-2)
3. 모터 구동 속도 설정 [↑](#footnote-ref-3)
4. 초음파 센서 사용시 거리 구하는 함수 [↑](#footnote-ref-4)