Be Artists of Robotics &
Advanced Micro Intelligence

바라미

Since 1994

ARDUINO

바라미 아두이노 세미나

5.C언어 문법, 함수, Keyboard/Mouse 라이브러리

김정현
kimdictor@gmail.com

박제윤
jeyunp@naver.com

C언어 배경지식

변수형

자료형(data type)		할당되는 메모리 크기	표현 가능한 데이터의 범위
정수형	char	1 바이트	-128 ~ +127
	short	2 바이트	-32768 ~ + 32767
	int	4 바이트	-2147483648 ~ + 2147483647
	long	4 바이트	-2147483648 ~ + 2147483647
실수형	float	4 바이트	$3.4 \times 10^{-37} \sim 3.4 \times 10^{+38}$
	double	8 바이트	$1.7 \times 10^{-307} \sim 1.7 \times 10^{+308}$
	long double	8 바이트 혹은 그 이상	차이를 많이 보임

출처:

C언어 배경지식

한정자

const

수정이 불가능한 읽기 전용 변수로 만든다.

Ex)const int var

Extern

다른 파일에 선언된 변수를 참조(공유)

C언어 배경지식

한정자

Static

함수/전역변수 앞
=>해당 함수/변수는 다른 파일에서 보이지 않음

지역변수 앞
=>함수호출에서 리턴해도 값이 사라지지 않음.

Volatile

컴파일러의 최적화 대상에서 제외
Ex)메모리 맵 입출력

```
static volatile int foo;

void bar (void)
{
    foo = 0;

    while (foo != 255);
}
```

C언어 배경지식

문자열 관련 함수: string.h 내에 존재

- size_t strlen(const char* str)
- char* strcpy(char* destination,const char* source)
- char* strncpy(char* destination, const char* source, size_t num)
- char* strcat(char* destination, const char* source)
- int strcmp (const char* str1, const char* str2)
- 이 외에 strncat, strncmp, strchr, strstr 등 다양한 함수 존재

C언어 배경지식

수학 관련 함수: string.h 내에 존재

- double cos (double x)
- double pow (double base, double exponent)
- double sqrt (double x)
- double fmax (double x, double y)
- double ceil (double x)
- 기타 다른 삼각함수, hyperbolic, 최대/최소값 구하는 함수들 존재.

C언어 배경지식

난수 생성

- stdlib.h, time.h 이용
- int rand() : 시드 값을 기반으로 랜덤 값 생성
- void srand(unsigned int seed) : rand 함수의 seed 변경
- time_t time(time_t* timeptr) : 1970년부터 흐른 시간 초 단위 반환
- srand(time(NULL));
  rand();

Keyboard 라이브러리

Keyboard 클래스에 속한 메서드

- begin() : 아두이노를 컴퓨터에 연결된 키보드로 에뮬레이트
- end() : 연결된 컴퓨터로의 에뮬레이션 종료
- press() : 키보드의 키가 눌린 것으로 처리
- release() : 눌린 특정 키를 손을 뗀 것으로 처리
- releaseAll() : 모든 눌린 키를 release

Keyboard 라이브러리

Keyboard 클래스에 속한 메서드

- print() : 연결된 컴퓨터로 키 보냄
- Println() : 키 보내고 줄바꿈
- write() : 연결된 컴퓨터로 keystroke 전송. 키를 눌렀다 떼는 것과 유사.

Keyboard.begin()/end()

Keyboard.begin()

구문: Keyboard.begin()
입력: none
반환: none
기능: 컴퓨터에 연결된 키보드 에뮬레이션 시작

Keyboard.end()

구문: Keyboard.end()
입력: none
반환: none
기능: 연결된 컴퓨터로의 키보드 에뮬레이션 중단

Keyboard.press()

Keyboard.press()

구문: Keyboard.press(key)
입력: char형. 누를 키를 의미
반환: size_t형. Key press가 전송된 횟수.
기능: 키가 키보드에서 계속 눌려 있는 것으로 처리

Keyboard.print()

Keyboard.print()

구문: Keyboard.print(character)
Keyboard.print(characters)
입력: character->char/int 전송.
characters->string 전송
반환: 전송된 바이트 개수 반환.
기능: 연결된 컴퓨터로 키 눌림 전송.

Keyboard.println()

Keyboard.println()

구문: Keyboard.println()
Keyboard.println(character)+
Keyboard.println(characters)
입력: character->char/int 전송.
characters->string 전송
반환: 전송된 바이트 개수 반환.
기능: 연결된 컴퓨터로 키 눌림 전송, 개행.

Keyboard.println()

Keyboard.println()

구문: Keyboard.println()
Keyboard.println(character)+
Keyboard.println(characters)
입력: character->char/int 전송.
characters->string 전송
반환: 전송된 바이트 개수 반환.
기능: 연결된 컴퓨터로 키 눌림 전송, 개행.

Keyboard.release()

Keyboard.release()

구문: Keyboard.release(key)
입력: char형. 누를 키를 의미
반환: release된 key의 개수 반환.
기능: 에뮬레이션된 키보드의 특정 키에서 손을 뗌.

Keyboard.releaseAll()

Keyboard.releaseAll()

구문: Keyboard.releaseAll()
입력: none
반환: none
기능: 눌린 모든 key를 release.

Keyboard.write()

Keyboard.write()

구문: Keyboard.write(character)
입력: char/int 형
반환: size_t형 보내진 바이트 개수
기능: 연결된 컴퓨터로 keystroke 보냄. 키를 눌렀다 떼는 것과 같음.

키보드 만들기

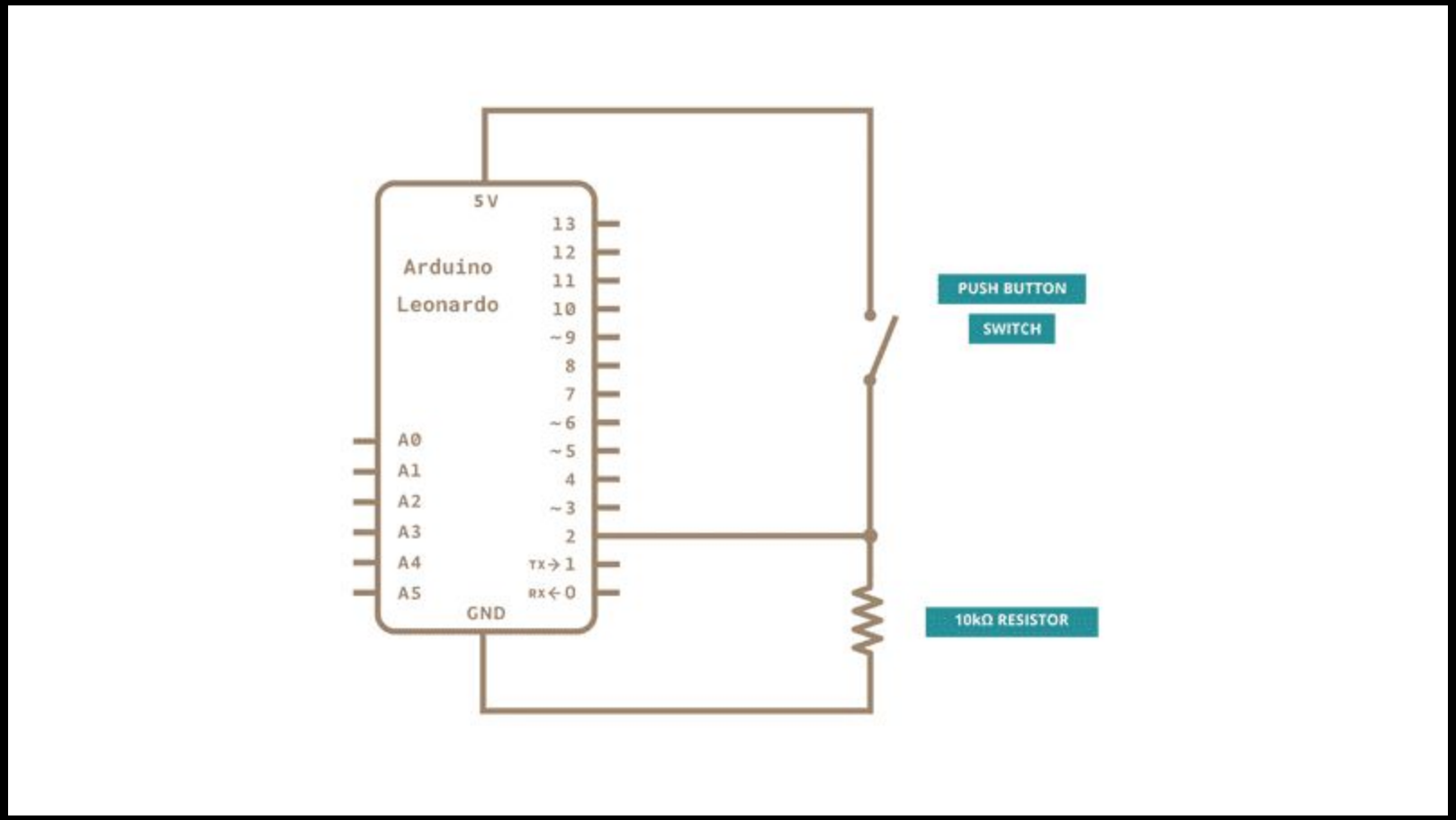
목적: 버튼을 누르면 컴퓨터에 " ㄹㅇ ㅋㅋ"문자열 출력

Keyboard 라이브러리 이용

다음 슬라이드의 코드를 업로드해 버튼을 누르면 “ㄹㅇ ㅋㅋ” 문자열이 출력되는 키보드를 만들 수 있다.

*programming port로 업로드 후 usb를 native port로 바꿔주세요. 키보드의 한영키로 한영변환합니다.

키보드 만들기


5V
13
12
11
10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
Arduino
Leonardo
A0
A1
A2
A3
A4
A5
GND
PUSH BUTTON
SWITCH
10kΩ RESISTOR

키보드 만들기

keyboardtest | 아두이노 1.8.13

파일 편집 스케치 툴 도움말

keyboardtest

```
#include "Keyboard.h"

const int buttonPin = 4;          // input pin for pushbutton
int previousButtonState = HIGH;   // for checking the state of a pushButton
int counter = 0;                  // button push counter

void setup() {
  // make the pushButton pin an input:
  pinMode(buttonPin, INPUT);
  // initialize control over the keyboard:
  Keyboard.begin();
}

void loop() {
  // read the pushbutton:
  int buttonState = digitalRead(buttonPin);
  // if the button state has changed,
  if ((buttonState != previousButtonState)
      // and it's currently pressed:
      && (buttonState == HIGH)) {
    // increment the button counter
    counter++;
    // type out a message
    Keyboard.println("fdzz");
  }
  // save the current button state for comparison next time:
  previousButtonState = buttonState;
}
```

Mouse 라이브러리

Mouse 클래스에 속한 메서드

- begin() : 아두이노를 컴퓨터에 연결된 마우스로 에뮬레이트
- end() : 연결된 컴퓨터로의 에뮬레이션 종료
- click() : 마우스 클릭
- move() : 연결된 컴퓨터에서 커서 움직임
- press() : 마우스 버튼 누름
- release() : 눌린 버튼 release
- isPressed() : 버튼의 현재 상태 반환

Mouse.begin()/end()

Mouse.begin()

구문: Mouse.begin()
입력: none
반환: none
기능: 컴퓨터에 연결된 마우스 에뮬레이션 시작

Mouse.end()

구문: Mouse.end()
입력: none
반환: none
기능: 연결된 컴퓨터로의 마우스 에뮬레이션 중단

Mouse.click()

Mouse.click()

구문: Mouse.click()
Mouse.click(button)
입력:
MOUSE_LEFT/MOUSE_RIGHT/MOUSE_MIDDLE
반환: none
기능: 커서의 위치에서 클릭 실행

Mouse.click()

Mouse.click()

구문: Mouse.click()
Mouse.click(button)
입력:
MOUSE_LEFT/MOUSE_RIGHT/MOUSE_MIDDLE
반환: none
기능: 커서의 위치에서 클릭 실행

Mouse.move()

Mouse.move()

구문: Mouse.move(xVal, yVal, wheel)

입력: xVal->x축 이동량. Signed char

yVal->y축 이동량. Signed char

wheel->scroll wheel 이동량. Signed char

반환: none

기능: 연결된 컴퓨터에서 커서 움직임. 현재 커서의 위치와 관련됨.

Mouse.press()

Mouse.press()

구문: Mouse.press()
Mouse.press(button)
입력:
MOUSE_LEFT/MOUSE_RIGHT/MOUSE_MIDDLE
반환: none
기능: 마우스 버튼 누름. 버튼을 계속 누르고 있는 것과 같다.

Mouse.press()

Mouse.press()

구문: Mouse.press()
Mouse.press(button)
입력:
MOUSE_LEFT/MOUSE_RIGHT/MOUSE_MIDDLE
반환: none
기능: 마우스 버튼 누름. 버튼을 계속 누르고 있는 것과 같다.

Mouse.release()

Mouse.release()

구문: Mouse.release()

Mouse.release(button)

입력:

MOUSE_LEFT/MOUSE_RIGHT/MOUSE_MIDDLE

반환: none

기능: 눌린 마우스 버튼에서 손을 떼는 입력을 집어넣음.

Mouse.release()

Mouse.release()

구문: Mouse.release()

Mouse.release(button)

입력:

MOUSE_LEFT/MOUSE_RIGHT/MOUSE_MIDDLE

반환: none

기능: 눌린 마우스 버튼에서 손을 떼는 입력을 집어넣음.

Mouse.isPressed()

Mouse.isPressed()

구문: Mouse.isPressed()
Mouse.ispressed(button)
입력:
MOUSE_LEFT/MOUSE_RIGHT/MOUSE_MIDDLE
반환: 버튼이 눌렸는지 여부. Bool type
기능: 마우스 버튼이 눌렸는지 확인.

조이스틱 마우스 제작

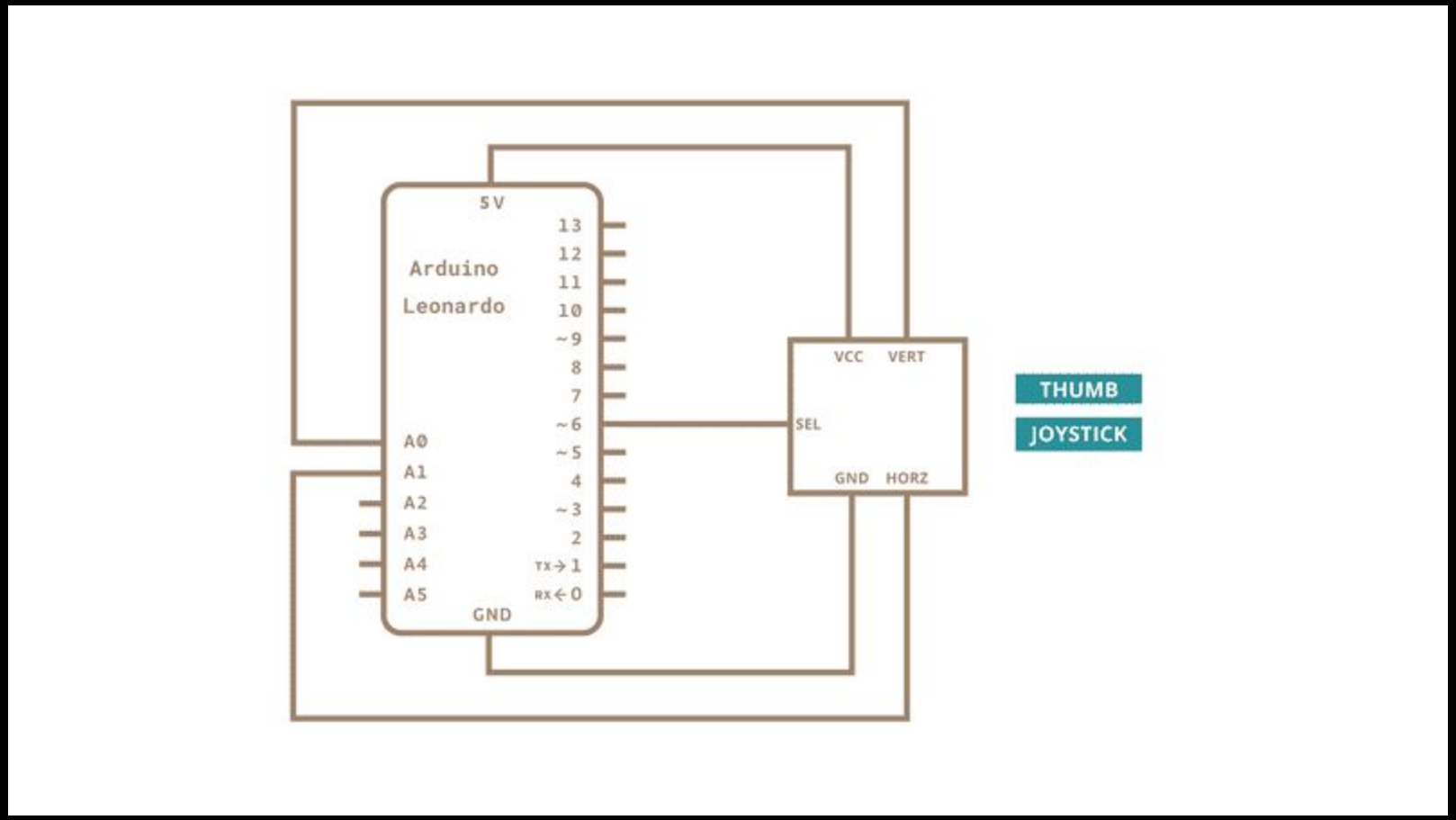
목적: 조이스틱 모듈을 연결해 마우스로 활용

Mouse 라이브러리 이용

다음 슬라이드의 코드를 업로드해 아두이노에 연결된 조이스틱 모듈이 마우스 기능을 하도록 만들 수 있다.

*programming port로 업로드 후 usb를 native port로 바꿔주세요.

마우스 만들기


5V
Arduino
Leonardo
13
12
11
10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
A0
A1
A2
A3
A4
A5
GND
VCC
VERT
SEL
GND
HORZ
THUMB
JOYSTICK

마우스 만들기

JoystickMouseControl §

```
#include "Mouse.h"

// set pin numbers for switch, joystick axes, and LED:
const int switchPin = 2;      // switch to turn on and off mouse control
const int mouseButton = 3;    // input pin for the mouse pushButton
const int xAxis = A0;         // joystick X axis
const int yAxis = A1;         // joystick Y axis
const int ledPin = 5;         // Mouse control LED

// parameters for reading the joystick:
int range = 12;               // output range of X or Y movement
int responseDelay = 5;        // response delay of the mouse, in ms
int threshold = range / 4;    // resting threshold
int center = range / 2;       // resting position value

bool mouseIsActive = false;    // whether or not to control the mouse
int lastSwitchState = LOW;        // previous switch state

void setup() {
  pinMode(switchPin, INPUT);       // the switch pin
  pinMode(ledPin, OUTPUT);         // the LED pin
  pinMode(mouseButton, INPUT_PULLUP);
  // take control of the mouse:
  Mouse.begin();
}

void loop() {
  // read the switch:
  int switchState = digitalRead(switchPin);
  // if it's changed and it's high, toggle the mouse state:
  if (switchState != lastSwitchState) {
    if (switchState == HIGH) {
      mouseIsActive = !mouseIsActive;
      // turn on LED to indicate mouse state:
      digitalWrite(ledPin, mouseIsActive);
```

마우스 만들기

확인

JoystickMouseControl §

```
      // turn on LED to indicate mouse state:
      digitalWrite(ledPin, mouseIsActive);
    }
  }
  // save switch state for next comparison:
  lastSwitchState = switchState;

  // read and scale the two axes:
  int xReading = readAxis(A0);
  int yReading = readAxis(A1);

  // if the mouse control state is active, move the mouse:
  if (mouseIsActive) {
    Mouse.move(xReading, yReading, 0);
  }

  // read the mouse button and click or not click:
  // if the mouse button is pressed:
  if (digitalRead(mouseButton) == HIGH) {
    // if the mouse is not pressed, press it:
    if (!Mouse.isPressed(MOUSE_LEFT)) {
      Mouse.press(MOUSE_LEFT);
    }
  }
  // else the mouse button is not pressed:
  else {
    // if the mouse is pressed, release it:
    if (Mouse.isPressed(MOUSE_LEFT)) {
      Mouse.release(MOUSE_LEFT);
    }
  }

  delay(responseDelay);
}
```

마우스 만들기

```
/*
  reads an axis (0 or 1 for x or y) and scales the analog input range to a range
  from 0 to <range>
*/

int readAxis(int thisAxis) {
  // read the analog input:
  int reading = analogRead(thisAxis);

  // map the reading from the analog input range to the output range:
  reading = map(reading, 0, 1023, 0, range);

  // if the output reading is outside from the rest position threshold, use it:
  int distance = reading - center;

  if (abs(distance) < threshold) {
    distance = 0;
  }

  // return the distance for this axis:
  return distance;
}
```