

스마트시스템입문

2018년 1학기

프로젝트 조편성

- “조편성 결과” 게시판 사용

강의 개요



과목공지



질의응답



조편성 결과

중간고사 공지 (A)

- 일시: 4월 25일
 - 필기고사 없음
 - 오후 3시-4시: 326호
 - 아두이노 반드시 지참
- 범위: 아두이노 관련 수업 자료 (이번주 수업 포함)
- Cheat sheet : A4 1장 (앞뒷면)의 cheat sheet 작성가능.
단, 복사, 출력 금지, 반드시 필기로 직접 작성할 것.
시험 시간 중 검색이 불가할 수 있음.

중간고사 공지 (B)

- 일시: 4월 25일
 - 필기고사 없음
 - 오후 5시-6시: 326호
 - 아두이노 반드시 지참
- 범위: 아두이노 관련 수업 자료 (이번주 수업 포함)
- Cheat sheet : A4 1장 (앞뒷면)의 cheat sheet 작성가능.
단, 복사, 출력 금지, 반드시 필기로 직접 작성할 것.
시험 시간 중 검색이 불가할 수 있음.

5월 1일 발표

- 이러닝 게시판 참조
- 요령은 동일

PWM

- Advantage of PWM
 - Power loss
 - Potentiometer: always on
 - PWM: no energy loss during off state
 - Accuracy
 - PWM resolution: the number of different steps you can have from zero to full power.
 - Arduino default = 8 bit
 - 8 bit resolution = 256 steps

PWM from Arduino

```
int pin = 11;  
int brightness = 10; // 0~255
```

```
void setup() {  
  pinMode(pin, OUTPUT);  
}
```

```
void loop() {  
  analogWrite(pin,brightness);  
}
```

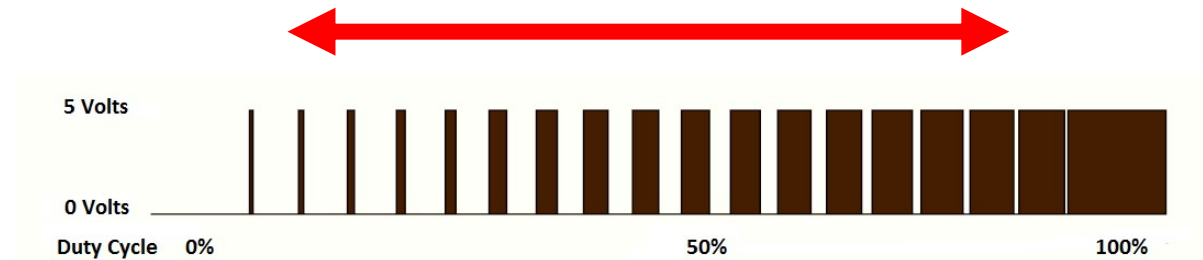
- “brightness”를 바꿔가면서 실행

PWM from Arduino

```
int pin = 11;
```

```
void setup() {  
  pinMode(pin, OUTPUT);  
}
```

```
void loop() {  
  for (int i=0;i<=255;i++){  
    analogWrite(pin,i);  
    delay(10);}  
  
  for (int i=255;i>=0;i--){  
    analogWrite(pin,i);  
    delay(10);}  
  
}
```



<https://electrosome.com/pwm-pulse-width-modulation/>

Simulate PWM

```
int pin = 11;
int Delay = 5;

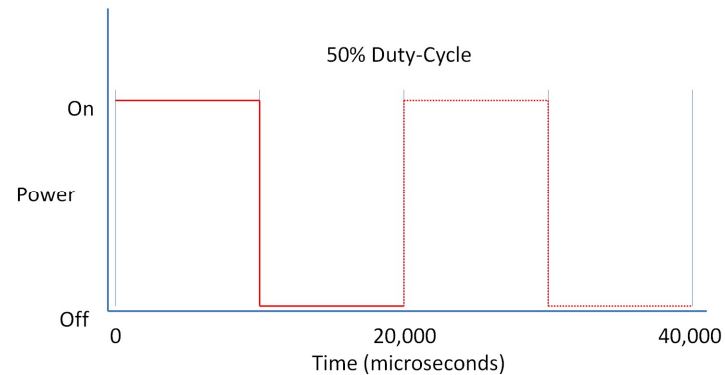
void setup() {
  pinMode(pin, OUTPUT);
}

void loop() {

  digitalWrite(pin,HIGH);
  delayMicroseconds(Delay);
  digitalWrite(pin,LOW);
  delayMicroseconds(1000 - Delay);

}
```

- Repeat HIGH and LOW
- Change “Delay”



<http://www.waitingforfriday.com/?p=404>

Control LED with potentiometer

```
int POT_pin = 0;  
int LED_pin = 11;
```

```
void setup() {  
  pinMode(POT_pin, INPUT);  
  pinMode(LED_pin, OUTPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  
  int POT = analogRead(POT_pin);  
  int PWM_OUT = map(POT, 0, 1023, 0, 255);  
  
  Serial.print(POT);  
  Serial.print("\t");  
  Serial.print(PWM_OUT);  
  Serial.print("\n");  
  
  analogWrite(LED_pin, PWM_OUT);  
}
```

- Serial monitor 실행

map(value, fromLow, fromHigh, toLow, toHigh)

Description

Re-maps a number from one range to another. That is, a value of fromLow would get mapped to toLow, a value of fromHigh to toHigh, values in-between to values in-between, etc.

Serial Communication

- Arduino \leftrightarrow PC two-way communication
- We've been using Arduino to PC only
- You can control the board with PC input
- Used functions
 - `Serial.available()`
 - `Serial.read()`

Serial.available()

- Serial.available()
- Parameters: none
- Returns: the number of bytes available to read
- Get the number of bytes (characters) available for reading from the serial port. This is data that's already arrived and stored in the serial receive buffer (which holds 64 bytes).
- 한 번에 여러 문자를 읽을 수 있음 → 연습

Serial.read()

- Serial.read()
- Parameters: none
- Returns: **the first byte** of incoming serial data available (or -1 if no data is available) - int.

```
int incomingByte = 0;    // for incoming serial data

void setup() {
    Serial.begin(9600);    // opens serial port, sets data rate to 9600 bps
}

void loop() {
    // send data only when you receive data:
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingByte = Serial.read();

        // say what you got:
        Serial.print("I received: ");
        Serial.println(incomingByte, DEC);
    }
}
```

Serial Communication

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  if (Serial.available() )  
  {  
    char val = Serial.read();  
    Serial.println(val);  
  }  
}
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  if (Serial.available() )  
  {  
    int val = Serial.read();  
    Serial.println(val);  
  }  
}
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  if (Serial.available() )  
  {  
    int val = Serial.read()-'0';  
    Serial.println(val);  
  }  
}
```

int vs. char?

Serial Communication: ASCII table

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Serial Communication: read string

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  if (Serial.available() )  
  {  
    String val = Serial.readString();  
    Serial.println(val);  
  }  
}
```

- 문자열 입력 가능
- 속도가 느리다
- 실행 속도를 어떻게 비교할 수 있을까?

Serial Communication: 속도 비교

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  if (Serial.available() )  
  {  
    unsigned long start_time = micros();  
    char val = Serial.read();  
    unsigned long end_time = micros();  
    Serial.println(val);  
    Serial.println(end_time - start_time);  
  }  
}
```

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  if (Serial.available() )  
  {  
    unsigned long start_time = micros();  
    String val = Serial.readString();  
    unsigned long end_time = micros();  
    Serial.println(val);  
    Serial.println(end_time - start_time);  
  }  
}
```

tone() , noTone()

Syntax

```
tone(pin, frequency)
```

```
tone(pin, frequency, duration)
```

Parameters

pin: the pin on which to generate the tone

frequency: the frequency of the tone in hertz - `unsigned int`

duration: the duration of the tone in milliseconds (optional) - `unsigned long`

- 한번에 한 개의 tone만 발생
- 다른 핀에서 tone이 실행되고 있으면 새로운 tone 명령은 실행이 안됨
- Arduino Uno: 31Hz - 65535Hz 주파수 생성 가능

Syntax

```
noTone(pin)
```

키보드 입력으로 buzzer 제어

```
int buzzer=7;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(buzzer, OUTPUT);  
}
```

```
void loop() {  
  if (Serial.available() )  
  {  
    char val = Serial.read();  
    Serial.println(val);  
    switch (val)  
    {  
      case '1':  
        {tone(buzzer, 20, 100);break;}  
      case '2':  
        {tone(buzzer, 40, 100);break;}  
      case '3':  
        {tone(buzzer, 80, 100);break;}  
      default:  
        {noTone(buzzer);break;}  
    }  
  }  
}
```

키보드 입력으로 LED On/Off 조절

- 조건: LED pin =11
- 1을 누르면 on
- 그외의 입력은 모두 off
- analogWrite을 사용하여 제어할 것

```
int pin = 11;
```

```
void setup() {  
  pinMode(pin, OUTPUT);  
  Serial.begin(9600);  
}
```

키보드 입력으로 LED On/Off 조절

- 조건: LED pin =11
 - 1을 누르면 on
 - 그외의 입력은 모두 off
 - analogWrite를 사용하여 제어할 것
-
- “123”을 입력하면 어떻게 되는가?
 - analogWrite 대신 digitalWrite를 사용하면 어떻게 되는가?

```
int pin = 11;

void setup() {
  pinMode(pin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() )
  {
    char val = Serial.read();
    Serial.println(val);
    if (val=='1')
    {analogWrite(11, 255);}
    else
    {analogWrite(11, 0);}
  }
}
```

키보드 입력으로 LED 밝기 조절?

- 조건: LED pin =11
 - 1을 누르면 on
 - 2를 누르면 밝기 증가 (PWM+25)
 - 3을 누르면 밝기 감소 (PWM-25)
 - 그외의 입력은 모두 off
-
- analogWrite을 사용하여 제어할 것
 - constrain 사용: PWM의 값을 0-255로 제한

키보드 입력으로 LED 밝기 조절?

```
int pin = 11;  
int brightness = 0;
```

```
void setup() {  
  pinMode(pin, OUTPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  analogWrite(pin, brightness);  
  if (Serial.available() )  
  {  
    char val = Serial.read();  
    switch (val)  
    {  
      case '1':  
        {brightness=255;break;}  
      case '2':  
        {brightness=brightness+25;break;}  
      case '3':  
        {brightness=brightness-25;break;}  
      default:  
        {brightness=0;break;}  
    }  
    brightness = constrain(brightness, 0, 255);  
    Serial.print(val);  
    Serial.print('\t');  
    Serial.println(brightness);  
  }  
}
```