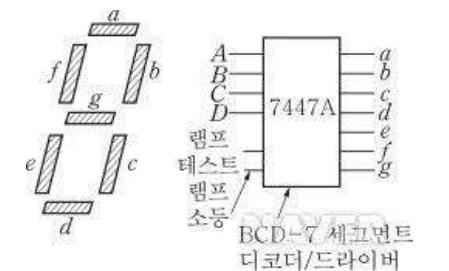

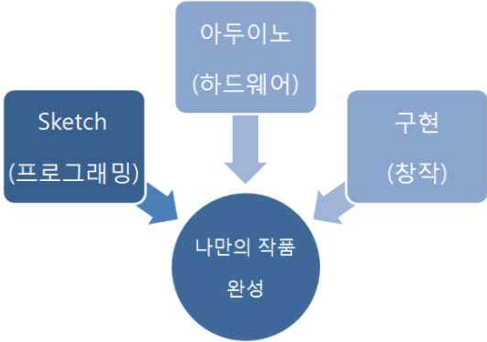


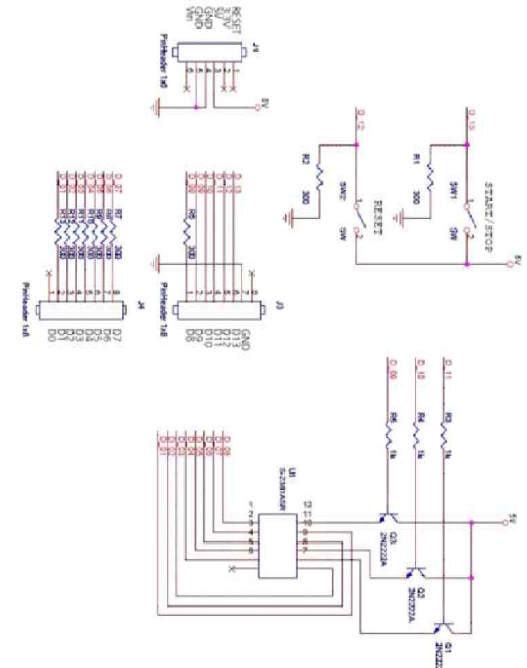
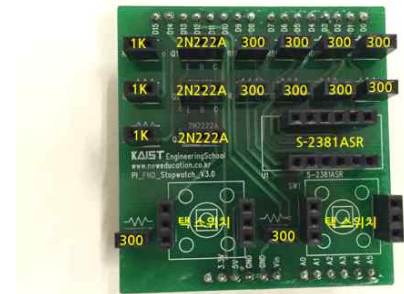
교사용 지도안

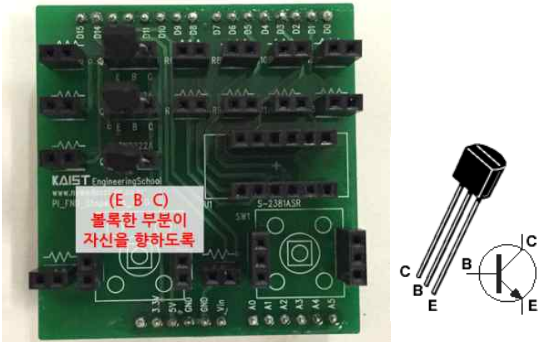
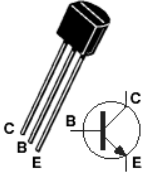
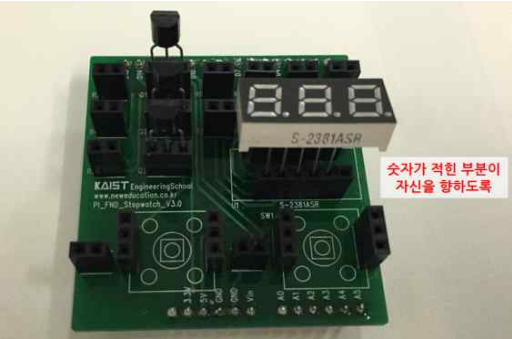

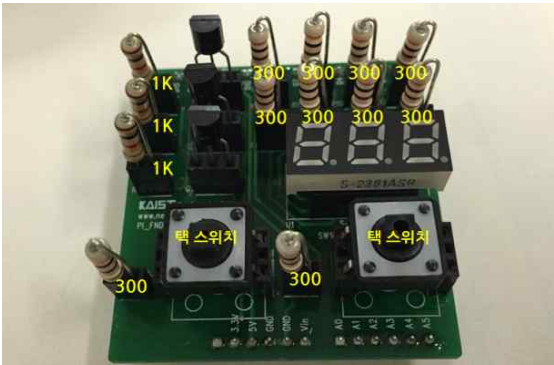
차 시	PS_09	대상	Planner(초등 6학년, 중등 1학년)	
교육주제	FND스탑위치 프로젝트	교육시간	150분	
교육목표	1. 아두이노의 개념과 작동 방식에 대해 알 수 있다. 2. FND의 개념과 작동 방식에 대해 알고, FND 회로를 제작할 수 있다. 3. 아두이노 프로그램을 이용하여 스탑위치 프로그래밍을 할 수 있다.			
교육자료 (준비물)	*장비류는 미리 배분하고, 재료는 각 실험 단계에서 배분한다.			
	Arduino	1.0	노트북 어댑터	1.0
	FND 스탑위치 프로그래밍 예제파일	1.0	마우스	1.0
	멀티탭	0.5	USB 케이블 (A-B)	1.0
	노트북	1.0	아두이노보드	1.0

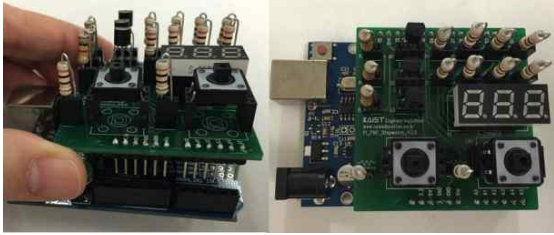

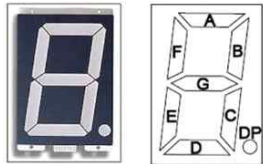
단계	교육내용	교육활동	시간	지도상의 유의점
도입	교육배경	<p>자동차에는 몇 개나 되는 마이크로컴퓨터가 들어간다고 생각하는가? 도요타의 대형 고급차 '크라운'에 들어있는 마이크로프로세서의 경우, 1991년 20개 미만이던 것이 2003년에는 50개로 늘어났다. 우리 주변에는 이처럼 수많은 마이크로프로세서들이 존재한다. 개인용 컴퓨터나 산업용 로봇·자동차 엔진·가정용 기기 등에서 제어장치로 이용되고 있다. 이렇게 활용가능성이 높은 마이크로프로세서가 기술이 발달해 가면서 점점 입문자를 위한 또는 비전문가를 위한 마이크로컴퓨터로서 나오기 시작하였는데 그 중 하나가 디자이너를 위한 아두이노이다. 우리는 아두이노를 조작하면서 기초적인 전자지식과 전산지식을 획득할 수 있고, 우리 나름의 작품을 만들 수 있다. 우리는 아두이노를 통해 FND스탑위치를 만드는 방법을 통해서 LED를 제어하는 법을 배우고 코드를 직접 짜볼 것이다.</p>		
도입	배경지식	<p>*마이크로프로세서 컴퓨터의 산술논리연산기, 레지스터, 프로그램 카운터, 명령디코더, 제어 회로 등의 연산장치와 제어장치를 1개의 작은 실리콘 칩에 모아놓은 처리 장치를 말한다. 주기억장치에 저장되어 있는 명령을 해석하고 실행하는 기능을 한다. 즉, 연산을 미리 확립된 순서에 의해 체계적으로 실행할 뿐만 아니라 컴퓨터의 각 장치에 제어 신호를 제공하는 제어장치를 1개의 작은 실리콘 칩에 집적시킨 초대규모 집적회로로 이루어진 처리장치이다. 마이크로프로세서는 주기억장치에 저장되어 있는 명령어를 인출하여 해독하고, 해독된 명령어를 실행하며 실행 결과를 다시 주기억장치에 저장할 수 있는 기능 등을 자동으로 수행함과 동시에 입출력 장치들과도 데이터 교환을 수행한다.</p>	20분	

단계	교육내용	교육 활동	시간	지도상의유의점
도입	배경지식	<p>*7-segment-diplay (FND)</p>  <p>7개의 조명 조각을 그림과 같이 배치하고, 그 몇 개를 골라서 빛을 냐으로써 0부터 9까지의 10진 digit을 표시할 수 있도록 한 표시 장치.</p> <p>*Arduino란? 아두이노, 혹은 아르두이노라고 읽는다. 이탈리아어로 '강력한 친구'라는 뜻. 자바기반의 오픈소스 프로그래밍 언어인 Processing과 AVR 기반의 마이크로컨트롤러 보드인 Wiring에서 파생한 프로젝트로, 2005년 이탈리아의 Massimo Banzì와 David Cuartielles가 처음 개발하였다. Arduino(이하 아두이노)는 오픈소스기반의 피지컬 컴퓨팅 플랫폼으로, AVR을 기반으로 한 보드와 소프트웨어 개발을 위한 통합 환경(IDE)을 제공한다. 쉽게 말하면, 흔히 마이크로컴퓨터, 마이크로프로세서, 마이크로컨트롤러라고 불리는 종류에 속한다. 세상에는 수많은 기계들이 있고 그 안에는 대부분 마이크로컴퓨터가 들어가 있다. 자동차에도 천개가 넘는 마이크로컴퓨터가 들어가 있으며, 최근에는 그 경향이 더욱 심해지고 있다. 그러나 공학자가 아니면 이런 마이크로컴퓨터를 다루기가 쉽지 않은 것이 사실이다. 미디어아트와 같은 일들을 하는 데에 있어서 모두 엔지니어의 자문을 구할 수는 없으므로 보다 다루기 쉬운 마이크로컨트롤러를 만들자는 움직임이 생겨났다. 그래서 시각적 효과에 최적화되어있고 비교적 쉬운 컴퓨터 언어인 프로세싱을 이용하는 아두이노가 제작이 되었다. 그 넓은 창작범위와 활용범위에 의해서 급속도로 널리 퍼져 수많은 작품들에 활용되고 있다.</p>  <p><아두이노 UNO> <아두이노 로고></p>	20분	

단계	교육내용	교육 활동	시간	지도상의유의점
도입	배경지식	<p>*Arduino의 작동방식</p>  <p>‘아두이노’라는 명칭은 하드웨어와 하드웨어를 프로그래밍하기 위한 소프트웨어 또한 지칭한다. 이때 나만의 작품을 만들기 위해서는 세가지 부분이 필요하다.</p> <p>첫 번째로 하드웨어인 아두이노가 필요하다. 두 번째로 아두이노 기능을 제어하는 컴퓨터 명령이 포함된 소스코드인 스케치 (sketch), 세 번째로는 그 외의 부분 (예를 들어 미디어아트의 경우 재료가 되는 영상 소스들)이 필요하다. 사람들은 이 부분들의 조합을 통해 다양한 작품을 만들어 낸다.</p>	20분	<p>아두이노에 대한 설명은 앞으로의 과정에 대한 간략한 소개와도 같다. 지루하지 않은 범위에서 적당히 소개하도록 하자.</p> <p>필요하다면 아두이노를 활용한 영상(교육정리에 첨부)를 통해 흥미를 일으키도록 한다.</p>
	교육개요	<p>1. FND스탑위치 제작(30분) 회로와 구성에 대한 설명, PCB기판 조립</p> <p>2. 스톱위치 프로그래밍(90분) FND에 대한 이해와 프로그래밍 2-1. 디스플레이 프로그래밍 2-2. 스톱위치 프로그래밍</p> <p>3. 조별게임시간(30분)</p>		

단계	교육내용	교육 활동	시간	지도상의유의점
전개	1. FND 스톱위치 제작	<p>FND 스톱위치의 회로도는 다음과 같다.</p> 	30분	<p>교사는 지도안을 참고하여 학생들이 원활히 따라할 수 있도록 지도한다. 학생들이 제작 시에 ppt의 사진을 보고 참조할 수 있도록 하며, 교사는 지속적으로 학생들의 결과물을 체크하도록 한다.</p>
		<p>FND의 각 자리수를 동작할 트랜지스터 3개와 저항 3개 그리고 FND안의 총 8개의 LED(a,b,c,d,e,f,g,dp(점))을 동작제어할 저항 8개와 기판이 동작하는지를 확인하기위한 LED와 저항이 있다. 마지막으로 각각의 START/STOP과 RESET스위치와 저항이 있는 것을 확인할 수 있다.</p> <p>#1. 보드와 부품을 나누어준다. ▶확인!</p>  <p>기판위에 그려진 도형을 관찰한다. 저항은 저항의 표시가 되어있고 LED는 LED표시가 되어있으며 트랜지스터는 방향성을 알기위한 CBE표시가 되어있다. 각각을 콧을 위치를 미리 잘 살펴본다.</p>		

단계	교육내용	교육 활동	시간	지도상의유의점
전개	1. FND 스탑위치 제작	<p>#2. 2N222A를 방향에 맞추어 꽂는다. ▶확인!</p>  <p>(E B C) 복록한 부분이 자신을 향하도록</p> 	30분	교사는 지도안을 참고하여 학생들이 원활히 따라할 수 있도록 지도한다. 학생들이 제작 시에 ppt의 사진을 보고 참조할 수 있도록 하며, 교사는 지속적으로 학생들의 결과물을 체크하도록 한다.
		<p>#3. FND(S-2381ASR)를 다음과 같이 꽂는다. ▶확인!</p>  <p>숫자가 적힌 부분이 자신을 향하도록</p>  <p>#4. 1K 저항과 300 저항, 택 스위치를 다음 위치에 꽂는다. ▶확인!</p>  <p>복록한 부분이 자신을 향하도록</p>		

단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스탑위치 프로그래밍	<p>#5. 보드와 아두이노를 연결한다. 보드와 아두이노의 1~13 번호는 같은 번호끼리 맞추어 연결하고, 보드의 5V와 GND는 아두이노의 GND에 연결한다. ▶확인!</p> 	90분	교사는 지도안을 참고하여 학생들이 원활히 따라할 수 있도록 지도한다. 학생들이 제작 시에 ppt의 사진을 보고 참조할 수 있도록 하며, 교사는 지속적으로 학생들의 결과물을 체크하도록 한다.
		<p>[2-1. 디스플레이 프로그래밍]</p> <p>#1. 아두이노를 케이블을 이용하여 컴퓨터와 연결한 후 arduino를 실행한다. ▶확인!</p>  <p>#2. '바탕화면\교육자료\EX1'을 열어준다. ▶확인! (화면의 크기가 작으면 가장자리를 드래그해서 크기를 늘린다.)</p> <p>#3. 빈 화면에 먼저 FND의 각 부분이 연결된 핀 숫자를 적어준다. FND는 a부터 g까지의 부분을 가지고 있으며, 각 부분이 연결된 핀 번호를 아래와 같으므로 동일하게 적어준다. ▶확인!</p> <pre> EX1 //FND의 각 LED의 핀번호를 입력한다. int a = 1; int b = 3; int c = 5; int d = 7; int e = 8; int f = 2; int g = 4; //자리점의 핀번호를 입력한다. int p = 6; //자리수의 핀번호를 입력한다. int d3 = 11; int d2 = 10; int d1 = 9; </pre> 		

단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스텝워치 프로그램 명	<p>(이때 int는 각 부분이 어디 핀에 연결되었는지를 설명하기 위한 함수이며 그 이후 그림과 같이 설명한다. a = 1의 의미는, a 부분이 아두이노 기판에서 1번 핀에 연결되었다는 의미이다. 마지막으로 줄을 바꾸기 전 설명을 끝내야 하므로 세미콜론(;)을 붙인다. 하단의 d1은 LED에서의 숫자 기판의 자릿수를 의미한다. 3칸의 LED창이 있으므로 d3까지 존재한다.)</p> <p>#4. LED창에 값을 출력할 것인지 입력할 것인지를 선언해야 한다. 만약 버튼이라면 입력(INPUT)이겠지만 출력(OUTPUT)으로 숫자를 표시해야 하므로 아래와 같이 각 부분을 출력한다는 것을 선언한다. ▶확인!</p> <pre> EX1 void setup() { //입력된 아두이노 핀을 모두 아웃쪽으로 설정하여 LED를 켤 수 있게 한다. pinMode(d1, OUTPUT); pinMode(d2, OUTPUT); pinMode(d3, OUTPUT); pinMode(a, OUTPUT); pinMode(b, OUTPUT); pinMode(c, OUTPUT); pinMode(d, OUTPUT); pinMode(e, OUTPUT); pinMode(f, OUTPUT); pinMode(g, OUTPUT); pinMode(p, OUTPUT); } </pre> <p>(이때 void setup()을 쓰고, 중괄호({ })를 입력한 후 중괄호 사이에 pinMode(부분, 입출력 선언)을 쓴다. 이렇게 쓰는 이유는 중괄호 안의 선언들이 setup에 엮인 선언임을 설명하는 것이다. pinMode는 위에서 선언한 요소(위에서 각 요소가 몇 번 핀에 연결되었는지 선언했으므로 다시 말해 아두이노의 핀 번호)가 입력값으로 받을 것인지 출력값으로 내보낼 것인지를 말한다.)</p>	90분	<p>아두이노를 연결 시에 드라이버를 자동으로 설치하지 못한다면 c드라이브의 program 폴더의 Arduino폴더를 드라이브 설치시 선택지정하여 설치한다.</p> <p>중괄호와 소괄호로 시작했다면 반드시 소괄호나 중괄호로 닫아야 하며, 콤마나 세미콜론을 반드시 확인하여 빼먹지 않는다.</p> <p>대소문자는 구분해서 정확히 입력한다.</p>

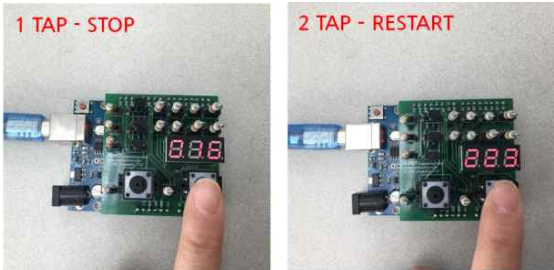
단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스텝워치 프로그램 명	<p>#5. 중괄호가 끝난 부분에 엔터를 치고 그 하단에 새로운 함수를 선언한다. ▶확인!</p> <pre> EX1 void loop() { //FNO가 모두 정상작동하는지를 테스트 하기위한 간단한 소스 테스트 //먼저 LED를 모두 끈 후 clearLEDs(); //2번째 자리수에 digitalWrite(d2, HIGH); //숫자 8을 LED내도록 한다. pickNumber(8); } </pre> <p>→ 0-9까지 원하는 숫자로 입력 가능</p> <p>#6. pickNumber 정리 ▶확인! #7. 자리점 선언 ▶확인!</p> <pre> EX1 //입력된 값에 (0과) 각 숫자의 함수를 설정시켜 숫자를 표시한다. void pickNumber(int x) { switch(x) { default: zero(); break; case 1: one(); break; case 2: two(); break; case 3: three(); break; case 4: four(); break; case 5: five(); break; case 6: six(); break; case 7: seven(); break; case 8: eight(); break; case 9: nine(); break; } } EX1 //자리점을 표시하는 함수 void display(int x) { digitalWrite(p, LOW); } </pre> <p>(이 함수는 pickNumber(숫자를 입력받음)이라는 함수를 만들어서 입력받은 숫자를 화면에 표시하는 역할을 수행하게 된다.)</p> <p>#6. 이제 LED창에 숫자를 표시할 것이다. 숫자가 바뀌기 이전에 입력된 숫자를 지워야 그 잔상이 남지 않을 것이다. 따라서 이전에 기록된 LED숫자를 지우는 함수를 아래 그림처럼 새롭게 만든다. 이 함수는 pickNumber의 중괄호가 끝나는 그 하단에 엔터를 쳐서 기록한다. ▶확인!</p> <pre> EX1 //모든 LED를 끄는 함수. LED의 잔상을 제거하는 목적으로 사용된다. void clearLEDs() { digitalWrite(a, HIGH); digitalWrite(b, HIGH); digitalWrite(c, HIGH); digitalWrite(d, HIGH); digitalWrite(e, HIGH); digitalWrite(f, HIGH); digitalWrite(g, HIGH); digitalWrite(p, HIGH); } </pre>	90분	<p>setup 이후의 하단괄호안의 설명은 질문이 들어오면 설명해주도록 하며, 학생들이 이해하지 못했을 경우 우선 따라할 것을 지시한다.</p> <p>따라한 이후에 PPT에서 각 부분을 짚어주면서 간단하게 설명하는 것으로 설명을 마무리한다.</p> <p>중괄호와 소괄호로 시작했다면 반드시 소괄호나 중괄호로 닫아야 하며, 콤마나 세미콜론을 반드시 확인하여 빼먹지 않는다.</p> <p>대소문자는 구분해서 정확히 입력한다.</p>

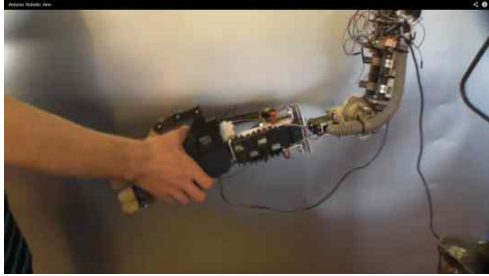

단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스탑워치 프로그래밍	<p>(여기서 digitalWrite(부분, 표시설정)함수는 LED창에 화면을 표시하는 것에 대한 설정을 해주는 함수이다. 하단의 LED창 요소에서 각 부분이 표시가 될지(LOW) 되지 않을지(HIGH)를 지정하는 것이다. 이때 콤마와 세미콜론(;)을 반드시 붙인다.)</p> <p>#7. 숫자 zero를 시작으로 숫자함수를 선언한다. 형식은 아래와 같이 하며, zero()를 시작으로 nine()까지 선언한다. ▶확인!</p> <pre>void zero() { digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, LOW); digitalWrite(f, LOW); digitalWrite(g, HIGH); }</pre>  <p>(0이므로 가운데 g를 제외한 요소가 보여야 한다. 고로 g를 제외한 나머지는 LOW로, g는 HIGH로 선언하였다. 이와 같은 방법으로 void one(), void two().. 순서대로 nine()까지 선언한다. 숫자 영어는 위에서 쓴 pickNumber()의 영어와 동일하게 적어야한다.)</p> <pre>void zero() { digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, LOW); digitalWrite(f, LOW); digitalWrite(g, HIGH); } void one() { digitalWrite(a, HIGH); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, LOW); digitalWrite(g, HIGH); } void two() { digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, HIGH); digitalWrite(d, HIGH); digitalWrite(e, LOW); digitalWrite(f, LOW); digitalWrite(g, HIGH); } void three() { digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, HIGH); digitalWrite(f, HIGH); digitalWrite(g, HIGH); } void four() { digitalWrite(a, HIGH); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, HIGH); digitalWrite(f, LOW); digitalWrite(g, HIGH); } void five() { digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, HIGH); digitalWrite(f, LOW); digitalWrite(g, HIGH); } void six() { digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, HIGH); digitalWrite(f, HIGH); digitalWrite(g, HIGH); } void seven() { digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, HIGH); digitalWrite(f, LOW); digitalWrite(g, HIGH); } void eight() { digitalWrite(a, LOW); digitalWrite(b, LOW); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, HIGH); digitalWrite(f, HIGH); digitalWrite(g, HIGH); } void nine() { digitalWrite(a, LOW); digitalWrite(b, HIGH); digitalWrite(c, LOW); digitalWrite(d, LOW); digitalWrite(e, HIGH); digitalWrite(f, HIGH); digitalWrite(g, HIGH); }</pre> <p>#8. 숫자를 다 기록했다면 상단의 업로드 버튼을 누른다. 잠시 뒤 FND에 출력되는 숫자를 확인한다. ▶확인!</p> <pre>void loop() { //FND가 모두 정상작동하는지를 테스트 하기위한 간단한 소스 테스트 //먼저 LED를 모두 끈 후 clearLEDs(); //2번째 자리수해 digitalWrite(d2, HIGH); //숫자 값을 나타내도록 한다. pickNumber(0); }</pre>  <p>0-9까지 원하는 숫자로 입력 가능</p>	90분	<p>중괄호와 소괄호로 시작했다면 반드시 소괄호로 중괄호로 닫아야 하며, 콤마나 세미콜론을 반드시 확인하여 빼먹지 않는다.</p> <p>대소문자는 구분해서 정확히 입력한다.</p>


단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스탑워치 프로그래밍	<p>[2-2. FND 스탑워치 프로그래밍(1)]</p> <p>#1. 바탕화면 \ 교육자료 \ EX2.ino을 연다. ▶확인! (EX1보다 추가된 소스의 의미를 이해한다.)</p> <p>#2. 앞에서 배운 내용은 모두 설정되었으나 가장 중요한 스탑워치를 실행할 소스가 없는 파일이다. 먼저 다음 위치에 스위치의 핀 번호 아래에 다음을 설정한다. ▶확인!</p> <pre>//자리수의 핀번호를 입력한다. int d3 = 11; int d2 = 10; int d1 = 9; //스위치의 핀번호를 설정한다. int switch1 = 12; //n 변수는 시간값을 표시하기 위해 사용한다. //ratio는 n변수가 너무 크기때문에 비례적으로 줄이기 위해 사용한다. //delayTime은 LED를 표시하는 시간을 설정한다. //button 변수는 초기화 버튼이 눌렸을 때의 상황을 표시하기 위해 사용한다. long n = 0; int ratio = 10; int delayTime = 1; int button = 0;</pre> <p>(int button = 0 이라 하였으므로, button이라는 변수를 설정함과 동시에 이 변수의 값을 0으로 지정하였다. 이렇게 생성한 button 변수는 버튼을 눌렀을 때의 경우를 판별할 수 있는 기준이 될 것이다.)</p> <p>#3. 다음 줄에서 입력된 아두이노 핀을 모두 아웃풋으로 설정하여 LED를 켜다. ▶확인!</p> <pre>void setup() { pinMode(d1, OUTPUT); pinMode(d2, OUTPUT); pinMode(d3, OUTPUT); pinMode(a, OUTPUT); pinMode(b, OUTPUT); pinMode(c, OUTPUT); pinMode(d, OUTPUT); pinMode(e, OUTPUT); pinMode(f, OUTPUT); pinMode(g, OUTPUT); pinMode(p, OUTPUT); pinMode(switch1, INPUT); }</pre> <p>첫 번째 스위치는 신호를 받아들이는 역할을 수행하도록 설정한다.</p>	90분	<p>중괄호와 소괄호로 시작했다면 반드시 소괄호로 중괄호로 닫아야 하며, 콤마나 세미콜론을 반드시 확인하여 빼먹지 않는다.</p> <p>대소문자는 구분해서 정확히 입력한다.</p> <p>트랜지스터의 다리가 얇아 접촉이 불량하여 FND가 밝게 표시되지 않거나 안보일 수도 있다. 그럴때 다리를 조금 휘어서 접촉이 잘 되도록 한다.</p>

단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스택위치 프로그래밍	<p>#4. 다음 줄에서 timeNumber를 계산하는 식을 정리 ▶ 확인!</p> <pre> EX2 § void loop() { int timeNumber = n / ratio; clearLEDs(); pickDigit(1); pickNumber((timeNumber/100)%10); delay(delayTime); clearLEDs(); pickDigit(2); pickNumber((timeNumber/10)%10); delay(delayTime); clearLEDs(); pickDigit(3); pickNumber(timeNumber%10); delay(delayTime); } </pre> <p>n이 빠르게 증가하며 숫자가 커지기 때문에 ratio변수를 숫자 10으로 나누어 표시</p> <p>먼저 LED를 모두 끈 후, 첫 번째 자릿수에 timeNumber의 100의 자리를 표시 100으로 나눈 후, 10으로 나눈 후의 나머지를 표시 LED를 delayTime 값 만큼 표시</p> <p>두번째 자릿수에 timeNumber의 10의 자리를 표시 10으로 나눈 후, 10으로 나눈 후의 나머지를 표시 LED를 delayTime 값 만큼 표시</p> <p>두번째 자릿수에 timeNumber의 1의 자리를 표시 10으로 나눈 후, 10으로 나눈 후의 나머지를 표시 LED를 delayTime 값 만큼 표시</p> <pre> EX2 § //button의 값이 1이되면, n값을 계속 1씩 증가시킨다. //이후 시간을 정지시키는 상황을 연출하기 위해 필요하다. if(button == 1) { n++; } //첫 번째 버튼이 눌리면, n=0으로 만든다. //초기와 버튼 역할 if(digitalRead(switch) == HIGH) { n = 0; } //에러를 방지하기 위해 timeNumber 값이 10000이 되면 n을 다시 0으로 만든다. if(timeNumber == 10000) { n = 0; } </pre> <p>#5. 다음 줄에서 몇 번째 자리에 숫자를 표시할지 선택할 수 있는 함수를 입력한다. ▶ 확인!</p> <pre> void pickDigit(int x) { digitalWrite(d1, LOW); digitalWrite(d2, LOW); digitalWrite(d3, LOW); switch(x) { case 1: digitalWrite(d1, HIGH); break; case 2: digitalWrite(d2, HIGH); break; default: digitalWrite(d3, HIGH); break; } } </pre>	90분	<p>중괄호와 소괄호로 시작했다면 반드시 소괄호나 중괄호로 닫아야 하며, 콤마나 세미콜론을 반드시 확인하여 빼먹지 않는다.</p> <p>대소문자는 구분해서 정확히 입력한다.</p>

단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스택위치 프로그래밍	<p>#6. 입력된 값에 따라 각 숫자의 함수를 실행시켜 숫자를 표시한다.</p> <p>▶ 확인!</p> <pre> void pickNumber(int x) { switch(x) { default: zero(); break; case 1: one(); break; case 2: two(); break; case 3: three(); break; case 4: four(); break; case 5: five(); break; case 6: six(); break; case 7: seven(); break; case 8: eight(); break; case 9: nine(); break; } } </pre> <p>#7. 상단메뉴의 업로드를 눌러서 FND 숫자가 정상적으로 올라가는지 확인해본다. ▶ 확인!</p>	90분	<p>중괄호와 소괄호로 시작했다면 반드시 소괄호나 중괄호로 닫아야 하며, 콤마나 세미콜론을 반드시 확인하여 빼먹지 않는다.</p> <p>대소문자는 구분해서 정확히 입력한다.</p>

단계	교육내용	교육 활동	시간	지도상의유의점
전개	2. 스탱위치 프로그래밍	<p>[2-3. FND 스탱위치 프로그래밍(2)]</p> <p>#1. 바탕화면 \ 교육자료 \ EX3.ino을 연다. ▶확인! (EX2보다 추가된 소스의 의미를 이해한다.)</p> <p>#2. 스위치의 신호를 받아들이는 수식에 2번 스위치에 대한 수식을 추가한다. ▶확인!</p> <pre>//스위치는 신호를 받아들이는 역할을 수행하도록 설정한다. pinMode(switch1, INPUT); pinMode(switch2, INPUT); // 두 번째 스위치에 대한 신호 수식 추가 }</pre> <p>#3. 1번 버튼에 대한 수식 다음에 2번 버튼의 역할 설정 ▶확인!</p> <pre>//첫 번째 버튼이 눌리면, n=0으로 만든다. //초기와 버튼 역할 if(digitalRead(switch1) == HIGH) { n = 0; } //2번째 버튼이 눌리면 button 변수에 -1을 곱하여 상황이 변했음을 저장한다. //아두이노의 반응이 버튼이 눌리는 속도보다 빠르기 때문에 //100 밀리초 동안 딜레이를 주어 이를 해결한다. if(digitalRead(switch2) == HIGH) { button *= -1; delay(100); } //에러를 방지하기 위해 timeNumber 값이 1000이 되면 n을 다시 0으로 만든다. if(timeNumber == 1000) { n = 0; } }</pre> <p>#4. 상단메뉴의 업로드를 눌러서 FND에 숫자가 스위치에 따라 어떻게 변하는지 확인 ▶확인! 왼쪽 버튼이 RESET 오른쪽 버튼이 STOP / RESTART</p> 	90분	<p>중괄호와 소괄호로 시작했다면 반드시 소괄호나 중괄호로 닫아야 하며, 콤마나 세미콜론을 반드시 확인하여 빼먹지 않는다.</p> <p>대소문자는 구분해서 정확히 입력한다.</p>

단계	교육내용	교육 활동	시간	지도상의유의점
정리	교육정리	<p>1. 아두이노로 또 어떠한 것을 만들 수 있을까?</p>  <p><Youtube동영상 Arduino Robotic Arm></p>  <p><Youtube동영상 Piano Stairs></p> <p>우리의 삶 구석구석에는 마이크로컴퓨터가 자리하고 있다. 그 기능을 아두이노를 이용하여 구현할 수도 있을 것이고, 마이크로 컴퓨터를 활용하여 할 수 있는 일을 이용해 새로운 형태의 창작을 해낼 수도 있을 것이다. 전문가가 아니면 만들기 어려웠던 위와 같은 로봇팔과 탱크를 만들 수도 있으며 수많은 미디어아티스트들의 사회운동에도 활용될 수도 있다. 'Piano Stairs'라는 캠페인에서 아두이노를 이용해 계단에 센서를 부착하여 '피아노 계단'을 만들어 사람들이 에스컬레이터보다 계단을 이용하게 하여 전기를 아끼는 사회적 움직임을 만들 수도 있을 것이다. 심지어 아두이노를 활용해 음악을 만드는 사람들도 있다. 당신의 머릿속에 무엇이 있든, 아두이노라면 그것을 가능하게 해줄 것이다.</p> <p>http://www.youtube.com/watch?v=gNeVxUAdv2o Youtube동영상 Arduino Tank(3분여)</p> <p>http://www.youtube.com/watch?v=ppMuu1X08T8 Youtube동영상 Arduino Robot Arm(6분여)</p> <p>http://www.youtube.com/watch?v=2lXh2n0aPyw 아두이노를 이용한 사회적 운동, Piano Stairs(2분여)</p> <p>http://www.youtube.com/watch?v=IY0mDRqcVU 아두이노를 이용한 음악, Still Alive(on Floppy Drives) (2분여)</p>	20분	<p>교육정리에서는 파워포인트에 있는 질문을 바탕으로 교육내용을 정리, 응용, 확장하는 시간이다.</p> <p>제작(전개)시간이 길어질 경우 교사의 재량에 따라 교육정리 시간은 단축해도 무방하다.</p>

단계	교육내용	교육 활동	시간	지도상의유의점
정리	교육정리	<p>2. FND 말고 다른 디스플레이를 이용할 수는 없을까?</p> <p>[그래픽 LCD 디스플레이 연결에 대하여] 아두이노의 라이브러리에는 GLCD 즉, 그래픽 LCD 디스플레이에 대한 라이브러리가 존재하여 제어할 수 있다. GLCD 라이브러리는 http://code.google.com/p/glkcd-arduino/downloads/list 에서 다운로드할 수 있다. GLCD 디스플레이를 연결하여 좀 더 화려한 그래픽을 표시할 수 있고 이를 위해서는 핀을 잘 확인하여야 한다. 특히 전원부가 잘못 연결될 시에는 디스플레이가 손상될 확률도 있음을 조심하자. 그래픽 디스플레이 연결 자체는 학생들의 수준에서는 보다 고난이도이며, 비트맵 이미지 등을 활용하여 직접 디스플레이에 표시하려면 프로세싱에 대한 이해도 필요하다. 더 자세한 설명을 위해서는 라이브러리의 설명서를 참조하도록 하자.</p>  <p><아두이노 UNO와 연결된 GLCD 디스플레이></p>	20분	<p>교육정리에서는 파워포인트에 있는 질문을 바탕으로 교육내용을 정리, 응용, 확장하는 시간이다.</p> <p>제작(전개)시간이 길어질 경우 교사의 재량에 따라 교육정리 시간은 단축해도 무방하다.</p>
	관련분야	<p>*아두이노 홈페이지 http://www.arduino.cc/</p> <p>*아두이노 개발 포럼 http://www.arduino.cc/forum/</p> <p>*고려대학교 컴퓨터시스템연구소 http://it.korea.ac.kr/ (02-3290-0896)</p> <p>*아두이노 개발 카페 아두이노 스토리 http://cafe.naver.com/arduinostory/</p>		