

캡스톤디자인 과제 결과보고서

과제명 : IoT 시스템 기초설계

캡스톤디자인 과제 결과보고서를 붙임과 같이 제출합니다.

지도교수 : 이규만(인)

대표학생 : 김지원(인)

참여인원 : 김지원 외 1 명



건양대학교 산학협력선도대학육성사업단



목 차

제1장. 과제내용 및 목표

제1절. 현황

제2절. 목적 및 필요성

제3절. 과제의 목표

제4절. 기대효과 및 활용방안

제2장. 개념설계 및 상세설계

제1절. 개념설계

제2절. 상세설계

제3장. 제작

제1절. 공정도

제2절. 제작

제4장. 운용 및 실험

제1절. 운용 및 실험 요구조건

제2절. 운용 및 실험결과

제5장. 결론

제1절. 문제점 분석 및 처리결과

제2절. 기업연계과제 결과 기술

제3절. 총평

부록(시제품 사진)

제1장. 과제내용 및 목표

제1절. 현황

			※ 접수번호		
캡스톤디자인 교과목 운영지원 사업 현황					
교과목명	IoT시스템 기초설계				
팀 명	6조				
팀 주제	safety necklace and badge				
지출액					
작품작동여부	<input type="checkbox"/> 작동 <input type="checkbox"/> 비작동 <input type="checkbox"/> 기타				
특허.실용신안					
지도교수	소 속	융합IT	휴대전화		
	성 명	이규만	E-mail		
대표학생	소 속	융합IT	휴대전화	010-6665-4202	
	성 명	김지원	E-mail	rlawldnjs433@naver.com	
참 여 학 생					
No.	소속학과	학년	학번	이름	
1	융합IT	3	14681004	김지원	
2	융합IT	3	14781010	김지은	

제2절. 목적 및 필요성

사물인터넷은 이미 우리 실생활에 깊숙이 스며들어 있다. 사물인터넷이란 인간과 사물, 서비스 세 가지 분산된 환경요소에 대해서 인간의 명시적인 개입 없이 센싱, 네트워킹, 정보 처리 등 지능적 관계를 형성하는 사물 공간 연결망을 의미한다. 이번 과목을 통해 사물인터넷 시스템 설계를 위해서 필요한 기본적인 개념부터 구현에 이르기 까지 필요한 소프트웨어, 하드웨어적인 지식을 학습했다. 이론에 그치지 않고, 매 프로그램마다 아두이노 및 IDE 플랫폼, 각종 센서와, 통신 인터페이스, 무선 통신 모듈들을 이용하여 실제 구현해보며 실생활에 적용해 볼 수 있는 아이디어를 제시하고 실제로 구현해 봄으로써 메이커의 역량을 기를 수 있다.

제3절. 과제의 목표

최근 들어 우리사회에서 남녀노소를 불문하고 '묻지마 범죄'의 대상이 되고 있으며, 범죄의 수준 또한 나날이 악화되고 있다. 이에 따라 사회적 문제에 대한 심각성을 인지하고 위험성을 줄일 수 있는 방안으로 호신용품과 사물인터넷을 접목한 방범용 목걸이와 호환되어 사용자에게 정보를 전달해주는 배지를 만들고자 한다. 늦은 귀갓길에 걸어가다 보면 뒤에 어떤 위험요소가 존재하는지 모른 채 지나다닌다. 이런 상황에 사용할 수 있는 IoT 제품을 생각해본 결과 일정 거리 이내에서 작동하는 적외선 거리센서를 통해 타인의 접근 여부를 감지, 목걸이와 배지 두 사물간의 무선 연결 통신을 이용한 정보 전송, 두 사물간 통신의 결과로 스피커와 LED 동작의 기능을 갖춘 하나의 제품을 만들고자 한다.

이를 통해서 여러 개의 센서들을 이용하여 하나의 완성된 제품을 구현할 수 있는 하드웨어 설계 능력과 각각의 센서들을 논리적으로 작동시키는 소프트웨어 능력을 기를 것이다.

제4절. 기대효과 및 활용방안

1. 기대효과

사용자가 방범용 목걸이와 배지를 착용함으로써 요즘 들어 더욱 자주 일어나는 범죄를 예방할 수 있도록 도와준다. 특히 주로 백팩 가방을 착용하는 어린아이들과 야간자율학습과 학원으로 인해 밤에 자주 돌아다니는 학생들에게 유용하게 사용 될 수 있다. 사용자의 가방에 방범용 배지를 달게 되면 인지하기 쉽지 않은 뒤에서 사람의 접근을 방범용 목걸이를 통해 알 수 있다. 사람의 접근뿐만 아니라 방범용 목걸이에서 나오는 경고음과 빨간색 조명으로 인해 범죄자들에게 위압감을 줄 수 있으며, 주위 사람들에게 자신이 위험한 상황에 놓여있다는 도움을 요청할 수 있는 매개체 역할을 할 수도 있다. 이러한 아이디어를 바탕으로 하나의 제품으로 구현해 낸다면 위험상황을 미리 인지하여 위험상황에서 좀 더 빨리 벗어날 수 있어 범죄가 줄어드는 효과를 기대해 볼 수 있을 것이다.

2. 활용방안

<Necklace>

- 방법모드

1. 목걸이의 1번 버튼과 배찌의 버튼을 누르게 되면 각각 빨간색 LED가 들어오고 서로 무선 통신을 시작하며 방법모드가 시작된다.
2. 배찌의 적외선 거리 센서가 20~80cm 범위 내에서 작동해 위험한 상황이라 인지하여 Xbee를 통해 데이터를 목걸이로 전송한다.
3. 목걸이의 스피커에서 경고음이 출력되며 빨간색 NeoPixel이 전체적으로 반짝거린다. 주변 사람들에게는 자신이 위험한 상황에 처해있다는 것을 알려줌과 동시에 범죄자를 위축하게 만든다.
4. 목걸이의 1번 버튼을 한 번 누를 때 작동을 멈추어 방법모드를 종료하며, 빨간색 LED가 꺼진다.

- 일반 모드

1. 목걸이의 2번 버튼을 누르면 후레시 기능을 하는 네오픽셀의 흰색 조명이 켜진다.
2. 후레시가 켜진 상태에서 목걸이 3번 버튼을 누르면 불이켜진 상태에서 스피커에서 경고음이 약 6초 동안 출력된다. (인터럽트)
후레시가 꺼진상태 에서는 스피커의 출력은 없다.
3. 다시 버튼 1을 한 번 누르면 후레시가 꺼진다.

<Badge>

- 방법 모드

1. 뒤에 있는 가방에 배찌를 달아 뒤에서 인지 할 수 없는 상황을 배찌를 통해 인지 할 수 있다.
2. 배찌에 있는 버튼을 누르면 방법모드가 시작되며, 빨간색 LED가 들어온다. Xbee를 통해 방법모드에서의 목걸이와 무선 통신을 시작한다.
3. 배찌에는 적외선 거리센서가 부착되어 있어 사람이 일정거리 이하로 접근이 된다면 측정된 거리를 SD카드에 저장하며, Xbee를 통해 목걸이 Xbee로 데이터를 전송한다. 전송된 데이터를 통해 목걸이가 반응하게 된다.
4. 배찌에 있는 버튼을 다시 한 번 누르면 방법모드가 종료되며, 빨간색 LED가 꺼진다.

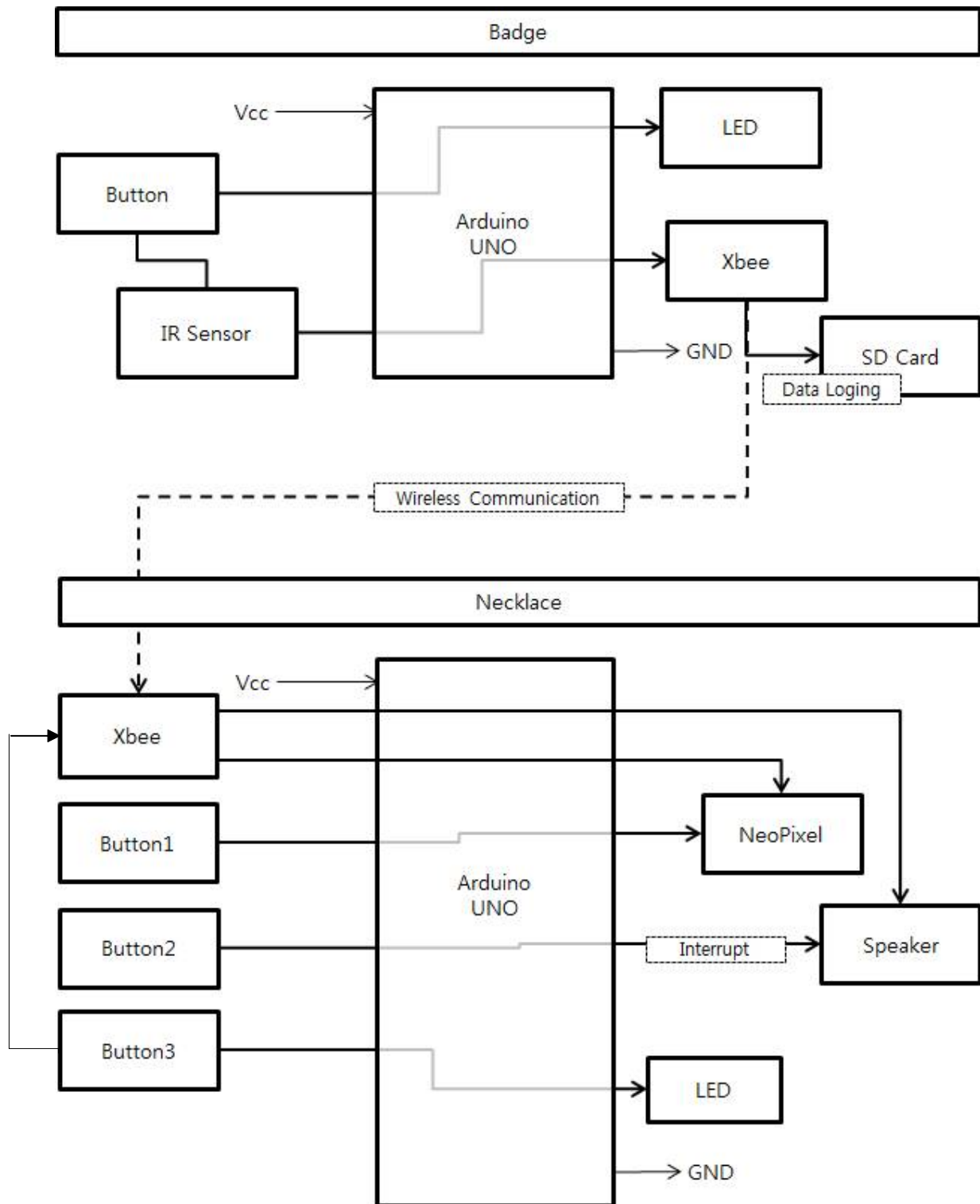
- 일반 모드

악세사리

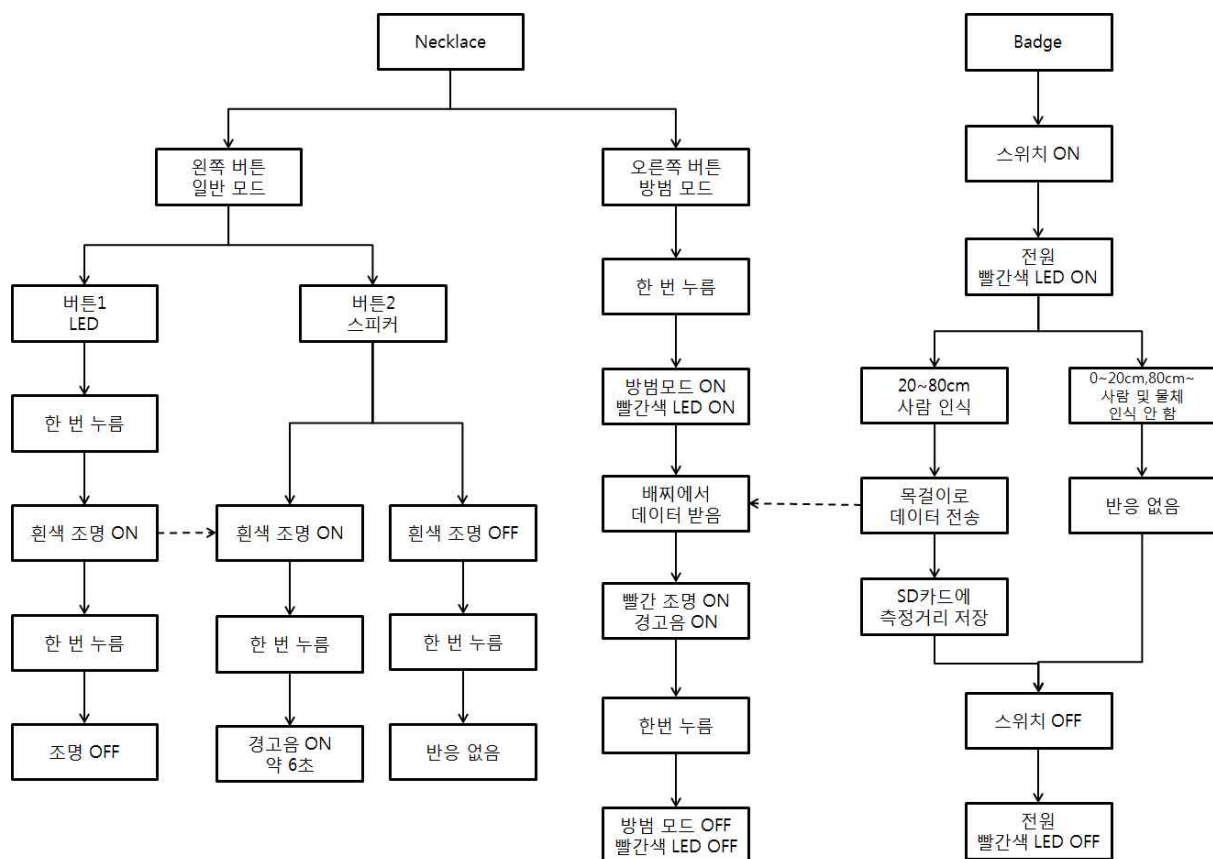
제2장. 개념설계 및 상세설계

제1절. 개념설계

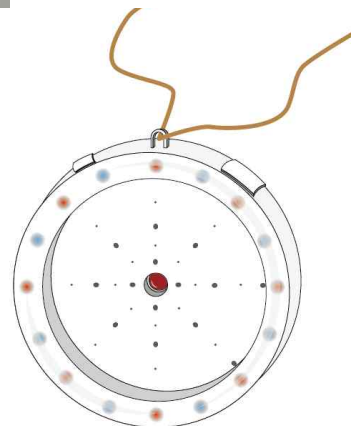
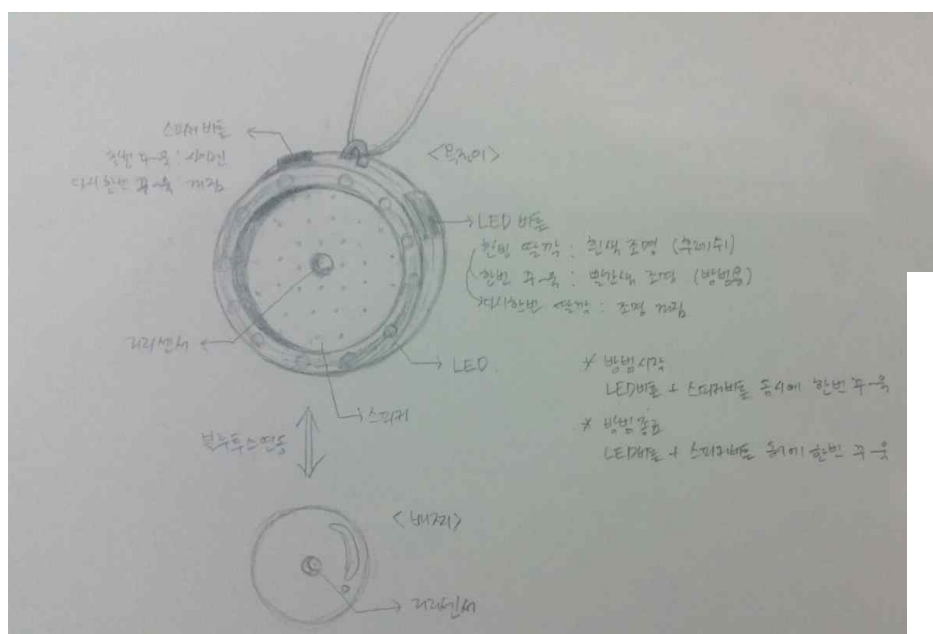
1. Block Diagram



2. 제품 순서도



2. 제품 스케치_초안



4. 전체일정

P	프로그램명	담당 교수	교수-학습 계획						
			주요 학습 내용	주요 학습 활동					
				Pre Class	In-Class				Post Class
				일 Act.	개인 Act.	실험 실습	강의		
P1	Internet of Things	이규만	<ul style="list-style-type: none"> 수업 오리엔테이션 오물 소개 및 일 발당 Internet of Things란 무엇인가? 	●	●			●	
P2	Internet of Things의 플랫폼	이규만	<ul style="list-style-type: none"> Internet of Things 국제표준 및 S/W 플랫폼 IoT 기초 실습 프로젝트 진행I 	●	●			●	
P3	Internet of Things의 시스템 기본구조와 실제 응용	이규만	<ul style="list-style-type: none"> IoT를 구성하고 있는 Hardware 시스템 분석과 실제 ATmega 데이터시트와 아두이노 회로도 분석 IoT 기초 실습 프로젝트 진행II 	●	●			●	
P4									
P5	오디오 제어	이규만	<ul style="list-style-type: none"> 음파의 속성 이해 아두이노 IDE 스피커를 이용한 소리 출력 일 기반 문제 해결 및 발표 	●	●			●	
P6	USB와 시리얼 통신	이규만	<ul style="list-style-type: none"> USB, Serial 통신 기능의 이해 데스크톱 응용 프로그램 제어 키보드, 마우스 애플리케이션 	●	●			●	
P7	시프트 레지스터	이규만	<ul style="list-style-type: none"> 시프트 레지스터의 이해 많은 수의 디지털 출력이 요구하는 프로젝트 구현 	●	●			●	
P8	통신 인터페이스(I): I2C 통신	이규만	<ul style="list-style-type: none"> I2C 프로토콜의 이해 시프트 레지스터와 시리얼통신 일 기반 문제 해결 및 발표 온도 모니터링 시스템 구현 	●	●			●	

P	프로그램명	담당 교수	교수-학습 계획						
			주요 학습 내용	주요 학습 활동					
				Pre Class	In-Class				Post Class
					팀 Act.	개인 Act.	실험 실습	검의	
P9	통신 인터페이스(2): SPI 통신	이규안	<ul style="list-style-type: none"> • SPI 프로토콜의 이해 • MCP4201 (디지털 가변저항) 응용 • 팀 기반 문제 해결 및 발표 • SPI 디지털 가변저항 시스템 구현 	●	●			●	
P10	Liquid Crystal Display 제어	이규안	<ul style="list-style-type: none"> • LCD의 이해 • LCD 라이브러리 사용 • 팀 기반 문제 해결 및 발표 • 자동 온도 조절 시스템 	●	●			●	
P11	무선 통신 시스템 응용	이규안	<ul style="list-style-type: none"> • 802.15.4 통신 표준 이해 • XBee 무선 모듈 설정 • 팀 기반 문제 해결 및 발표 • 무선 센서 네트워크 시스템 구현 	●	●			●	
P12	비동기 하드웨어/타이머 인터럽트	이규안	<ul style="list-style-type: none"> • 타이머와 인터럽트의 이해 • 반전 슈미트 트리거 회로 이해 • 비동기 시운트 시스템 구현 	●	●			●	
P13	데이터 로깅 시스템 구현 프로젝트	이규안	<ul style="list-style-type: none"> • 팀 기반 문제 해결 및 발표 준비 • 팀별 결과 발표 • 기말시험 (50분) • 모듈 팀 프로젝트 결과 발표 • 팀별 프로젝트 수행 결과 발표 • 아이디어 보완을 위한 토론 • 모듈세미나 						
P14					●			●	
P15									

제2절. 상세설계

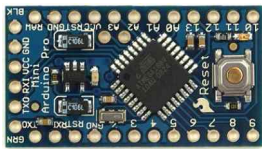
1. 상세설계도면 및 부품설명

1) 아두이노 YUN, NANO _1개



Arduino YUN

방법용 목걸이에 사용한다. 목걸이의 두 가지 기능 일반용과 방법용을 구현하기 위해서 선택하였다. YUN에는 와이파이 모듈이 내장되어 있기 때문에 와이파이를 통한 통신이 가능하다. 이 기능을 이용하여 방법용 모드가 작동하면 지정해놓은 스마트폰으로 사용자의 거리 값이 기록되고 위험하다는 정보를 전송할 수 있다.



Arduino NANO

방법용 배찌에 사용한다. 목걸이보다 비교적 크기가 작기 때문에 NANO를 통해 구현할 수 있다.

2) 만능기판_1개



제작할 제품의 크기 조절과 제품의 무게를 가볍게 하기 위해서 브레드보드보다는 만능기판의 사용이 더 용이하다.

3) 저항_6개



LED - 180Ω _2개 (목걸이 , 배찌)

스위치 - 91Ω _3개

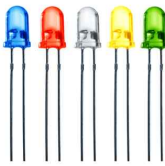
$4.7K\Omega$ _1개

인터럽트 핀 - 91Ω _1개

4) 리드선



5) 기본 LED_2개, NeoPixel_1개



기본 LED

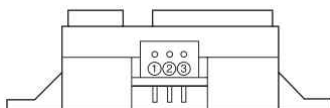
목걸이에서는 방법모드인지 일반모드인지 확인할 때 사용한다.
배지에서 배지의 전원 여부를 확인할 때 사용한다.



NeoPixel

목걸이에서 일반모드 경우에는 흰색 조명으로 방법모드 경우에는 빨간색 조명으로 사용된다.

6) 적외선 거리센서_1개



PIN	SIGNAL NAME
①	V _{CC}
②	GND
③	V _{CC}

적외선 거리센서_GP2Y0A21

뒤에서의 거리 측정 즉, 배지에서만 거리를 입력받아, Xbee를 활용해 목걸이로 무선으로 데이터를 전송하고 전송받은 데이터에 따라 목걸이가 반응하도록 한다.

적외선센서는 일정 주파수의 빛을 발산하는 발광부와 발산하는 빛을 받아들이는 수광부로 구성되어있다. 발광부에서 발생된 적외선이 물체에 부딪혀 반사되고, 반사된 빛을 수광부에서 감지하여 물체의 유무와 거리등을 측정할 수 있다. 또한 아날로그방식의 거리 측정 센서로 주변 환경(센서 내부에서 전압변화, 노이즈, 높은 온도 등)에 따라 측정 가능한 거리에 영향을 준다. 보통 30cm정도의 측정이 가능하다. 적외선을 방출하는 Light emitter부분과 쏘아 올린 적외선이 다시 되돌아올 때 그 빛을 다시 감지하는 Light detector부분으로 구성되어 있다. 빨강 선은 VCC에, 검은 선은 GND에, 노란 선은 S(Sensor)에 연결한다.

8) 스피커_1개, 스위치_4개



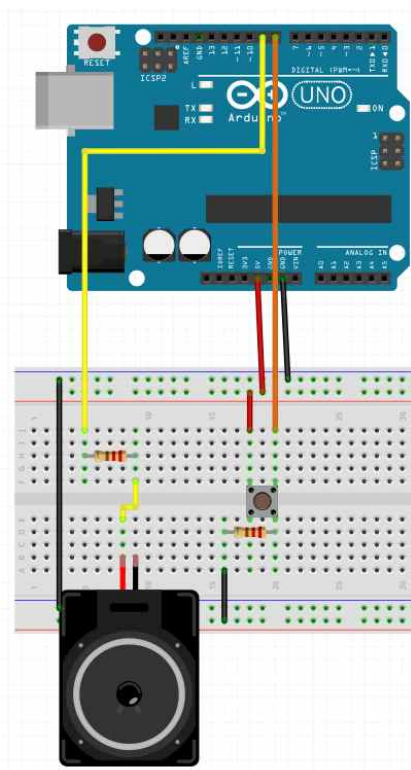
스피커

경고음을 출력할 때 사용한다.



스위치

목걸이에서 일반모드와 방법모드의 작동 스위치로 사용한다. 배찌에서는 전원 스위치로 사용한다.



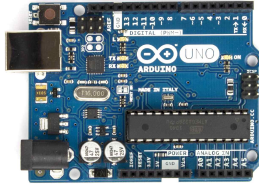
1. 스피커의 (+),(-) 두 개의 선을 브레드보드에 연결할 수 있도록 리드선과 납땜한다.
2. 옴의 법칙을 이용하여 회로에 최소한으로 필요한 저항 값을 계산하여 회로를 구성한다.
3. 스위치는 풀다운 저항으로 연결한다. 풀다운 저항이란 GND에 묶어둔다는 뜻으로 스위치를 누르지 않은 경우에는 전류가 GND로 흐르게 되어 0V가 되고 스위치를 누른 경우에는 전류가 Vcc로 통하기 때문에 핀은 5V를 갖게 되며 1을 입력한 것으로 인식된다.
4. 스위치를 누르게 되면 소리가 출력되고 누르지 않으면 소리가 출력되지 않는다.

9) 목걸이와 배찌 형태 구체화

Evolve를 통해 목걸이와 배찌의 모양을 만들고 3D print를 통해 제작

2. 도면 또는 부품 변경내역

1) 아두이노 YUN, NANO --> UNO_2개



Arduino UNO

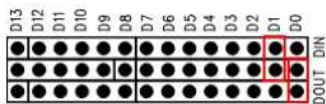
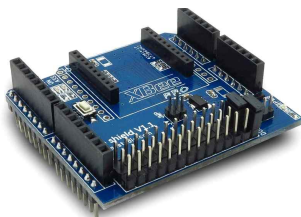
Xbee의 와이파이 실드를 사용하기 위해 UNO를 사용하며, 전지 커넥터를 사용하기 위해 UNO를 선택하였다.

2) 수은 건전지 3V --> 1.5V전지_4개, 커넥터_2개



Arduino UNO에 전력을 공급하기 위해 사용한다.

3) Xbee_2개, WiFi Shield_2개



배찌와 목걸이의 무선통신을 위해 사용한다. 배찌가 사용자 뒤에서 거리를 감지하게 되면 무선으로 목걸이와 통신하여 목걸이에서 경고음과 빨간빛이 나오게 한다.

XBee 모듈은 IEEE 802.15.4 표준에 부합하고 저비용, 저전력 무선 센서 네트워크의 고유한 요구를 지원하기 위해 설계되었다. 이 모듈은 최소한의 전력을 필요로 하고, 장치 사이의 데이터의 안정적인 전송을 제공한다. 또한 ISM 2.4 GHz 주파수 대역에서 작동하며 pin-for-pin 방식으로 서로 호환된다.

예를 들어 WiFi Shield에서 점퍼연결을 D1을 DIN에다 연결하고, D0을 DOUT에 연결하기 위해서 위의 그림처럼 점퍼연결을 하면 된다.

4) SD 카드_1개, SD 카드 Shield_1개



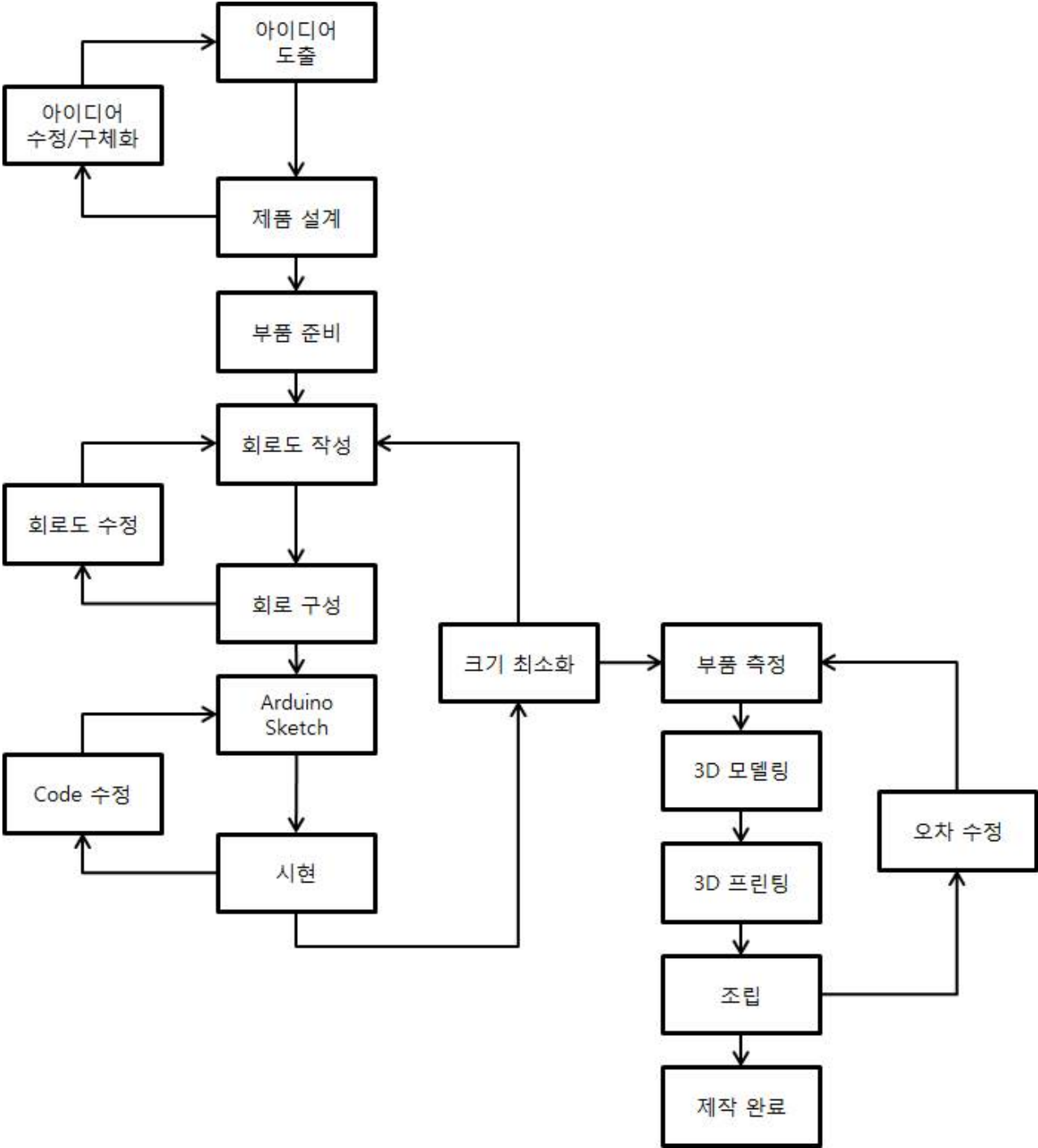
배씨의 적외선센서에서 측정된 거리 값을 SD카드에 저장하기 위해 사용한다.

SD카드 실드는 논리 레벨을 3.3V로 조정하고 3.3V의 전원 공급을 처리할 수 있어야 한다. SD카드와 아두이노는 SPI 통신을 통해 데이터를 주고받는다.

CS_PIN에서는 실제 사용하는 SD실드에서 사용하는 CS 핀 번호를 지정한다. 기본 CS_PIN 번호는 10이지만, 아두이노 공식 이더넷 실드를 사용하면 CS_PIN은 4이다.

제3장. 제 작

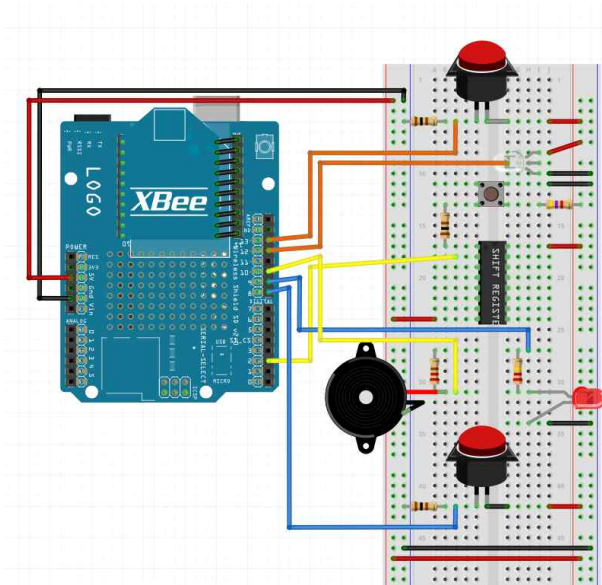
제1절. 공정도



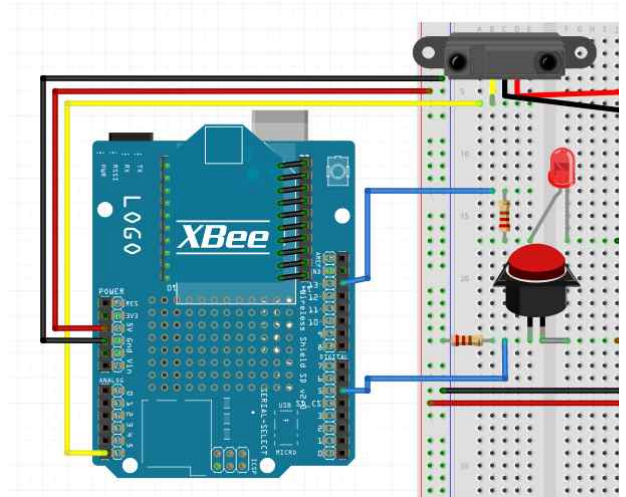
제2절. 제작

1. 제작과정

[회로도]

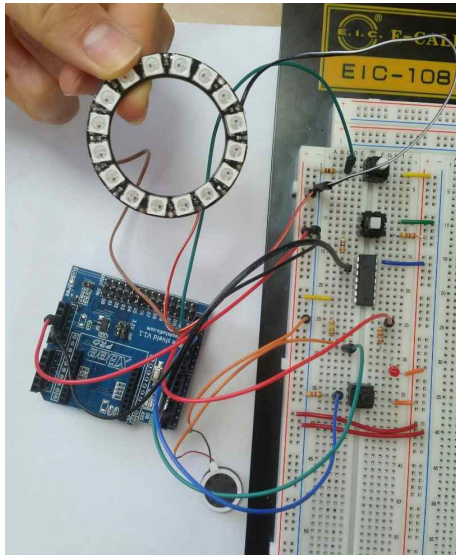


[목걸이]

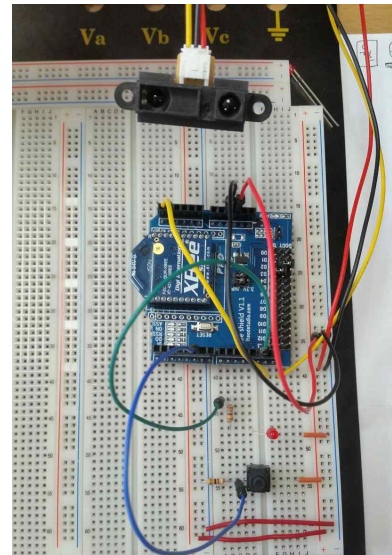


[배찌]

[회로 구성]

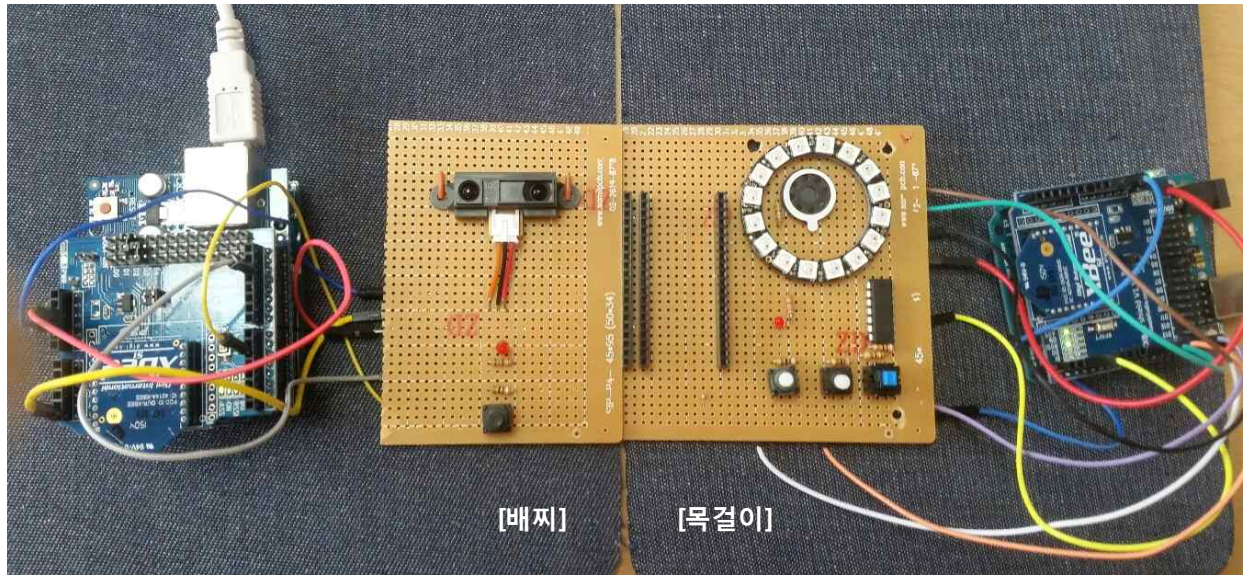


[목걸이]



[배찌]

[PCB판 구성]



브레드보드를 이용해 초기회로를 구성하였고 이후 PCB판을 이용한 회로구성은 목걸이와 배찌를 어떻게 제품화 할지에 대해 고민해보고 구성해보았다.

제4장. 운용 및 시험

제1절. 운용 및 시험 요구조건

1. IR센서

- ① 측정 밖의 거리
- ② 방향
- ③ 오차범위

이번실험에서 사용하게 될 IR센서는 GP2Y0A21이다. datasheet의 분석결과 위 센서의 측정범위는 10~80cm 이며 4.5V~5V의 전압이 사용되어야 한다.

하지만 이번에 만들고자 하는 제품의 특성상 거리측정에 대한 오류가 없어야 하며, 다각도로 측정이 가능해야 한다. 적외선 센서가 사람을 인지할 수 있는 거리의 범위와 정면이 아닌 측면에서도 결과값을 도출해 낼 수 있는지에 대해 시험해 봐야한다. 또한 오차범위를 파악하여 소프트웨어적으로 제어가 가능해야 한다.

2. 전압공급

회로에 사용하는 스위치, LED, 적외선센서, 스피커, 네오픽셀, XBee, SD card 모든 센서에 충분한 전원을 공급해주기 위해서 9V전지를 사용할 것이며, 이를 토대로 만들고자 하는 제품의 구현이 9V전지 2개의 전압공급으로 가능한지에 대해 시험해 봐야한다.

3. 스위치 사용

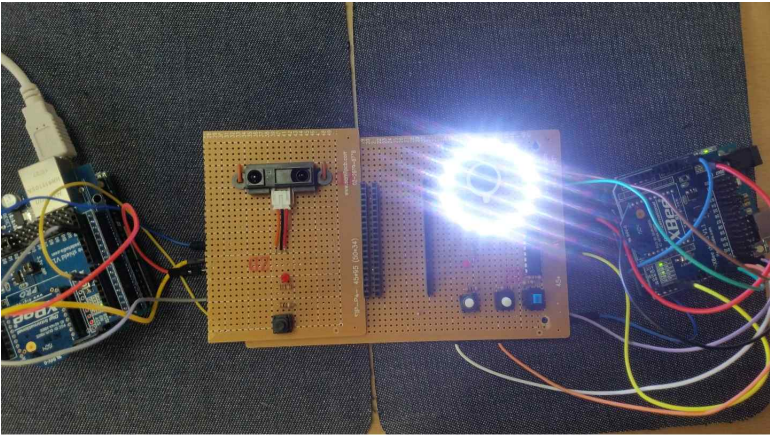
각 스위치별 출력 값도 다르며, 스위치별 동작 순서가 존재하기 때문에 이에 주의하여 코딩해야하고 디버깅 문제 또한 해결해야 한다.

4. Xbee무선 통신

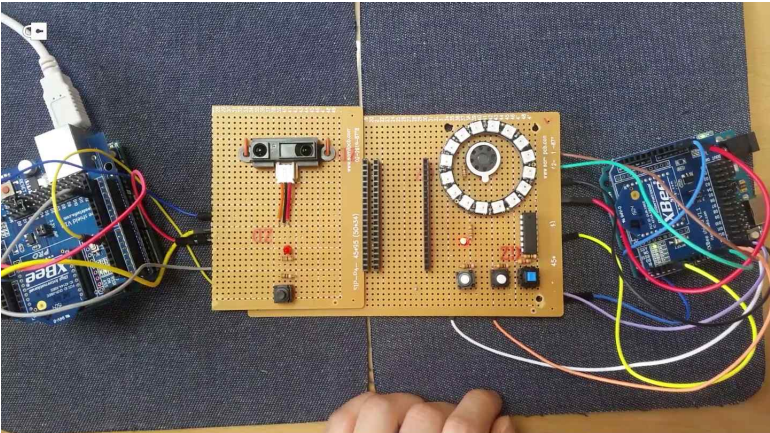
목걸이와 배찌가 Xbee로 무선으로 통신되어야 한다. 배찌에서 거리를 측정한 값으로 인해 목걸이가 작동해야하기 때문이다.

제2절. 운용 및 시험결과

[일반모드 후레시 작동]

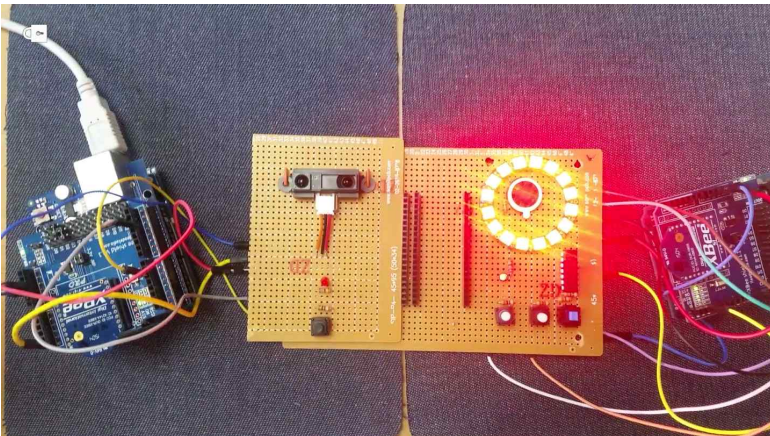


[방법모드 ON]



63	15299	11	15299,	11
64	15612	6	15612,	6
65	15927	19	15927,	19
<input checked="" type="checkbox"/> 자동 스크롤				

[20~80cm 사람 및 물체 인식]



58	13724	25	13409,	25
59	14039	24	13724,	25
60	14354	22	14039,	24
61	14669	22	14354,	22
62	14984	22	14669,	22
			14984,	22

[XBee 통신 및 SD카드 데이터 저장]

The screenshot displays a Windows XP desktop environment. In the background, a file explorer window shows the contents of a drive labeled '이동식 디스크 (G:)'. Overlaid on this is a Microsoft Excel spreadsheet titled 'LOG - Microsoft Excel'. The spreadsheet contains a table of data with columns A, B, C, and D. A third window, titled 'COM7 (Arduino/Genuino Uno)', is open in the foreground, showing a list of data received from the Arduino board. The data is presented in a two-column format, matching the structure of the Excel spreadsheet. The terminal window also includes settings for 'line ending' and 'baud rate'.

이름	수정된 날짜	유형	크기
LOG	2000-01-01 오전...	Microsoft Excel ...	1KB

	A	B	C	D
43	9003	18	9003,	18
44	9317	17	9317,	17
45	9630	17	9630,	17
46	9945	23	9945,	23
47	10259	28	10259,	28
48	10574	19	10574,	19
49	10888	28	10888,	28
50	11203	25	11203,	25
51	11517	25	11517,	25
52	11833	19	11833,	19
53	12146	18	12146,	18
54	12461	23	12461,	23
55	12776	16	12776,	16
56	13095	16	13095,	16
57	13409	25	13409,	25
58	13724	25	13724,	25
59	14039	24	14039,	24
60	14354	22	14354,	22
61	14669	22	14669,	22
62	14984	22	14984,	22
63	15299	11	15299,	11
64	15612	6	15612,	6
65	15927	19	15927,	19

COM7 (Arduino/Genuino Uno)
9003, 18
9317, 17
9630, 17
9945, 23
10259, 28
10574, 19
10888, 28
11203, 25
11517, 25
11833, 19
12146, 18
12461, 23
12776, 16
13095, 16
13409, 25
13724, 25
14039, 24
14354, 22
14669, 22
14984, 22
15299, 11
15612, 6
15927, 19

제5장. 결론

제1절. 문제점 분석 및 처리결과

1. 과제에 대한 문제 분석

가. 문제점

1) 각각의 센서들은 작동하지만 여러개의 센서들을 한꺼번에 작동시키면 제대로 작동하지 않음

2) PCB판의 회로구성의 문제 발생

나. 개선방안 및 처리결과

1) 회로상의 혼선을 없애고 적절한 저항을 사용

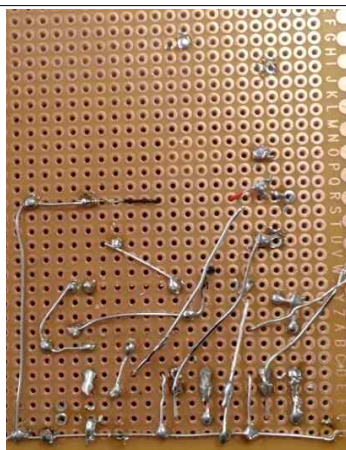
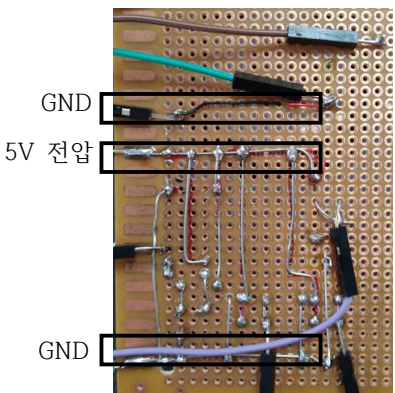
LED - 180Ω _2개 (목걸이 , 배찌)

스위치 - 91Ω _3개

$4.7K\Omega$ _1개

인터럽트 핀 - 91Ω _1개

2) PCB판 회로구성의 문제 해결

	전	후
회로구성		
설명	회로의 5V와 GND를 고려하지 않은 회로 구성으로 인해 다른 회선과의 합선이 일어나 각 센서들이 제대로 작동하지 않았다.	각 센서의 회로를 미리 구상하고 5V와 GND를 고려하여 회선을 구상하니 훨씬 더 깔끔하고 잘 작동되는 것을 확인할 수 있었다.

제3절. 총평

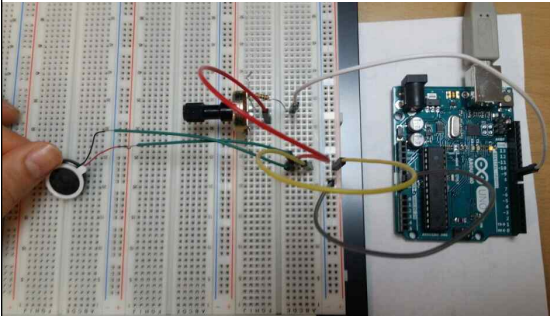
IoT는 모든사람이 언급하고 사용할 만큼 화두가 되고 있다. 이런 사물인터넷을 다양한 오픈소스와 센서, 하드웨어 플랫폼을 이용하여 계획해보고 구현해보았다. 더불어 IoT의 국제표준 s/w 의 플랫폼들에 대해 조사해보고 지금 현시점의 IoT에 대해 알아보면서 앞으로 메이커로써 어느 방향으로 나아가야 할지에 대해 조사해보고 알아보면서 충분한 역량을 길러 볼 수 있었다.

각 프로그램에서 스피커를 통한 소리출력, LCD화면을 이용한 문구출력, 유·무선통신들을 이용한 실험을 진행해보았고 각 조마다 프로젝트를 구상한 후 safety necklace and badge를 구상한 시나리오에 맞게 부품들과 통신방법을 선택해 프로젝트를 수행해 볼 수 있었다.

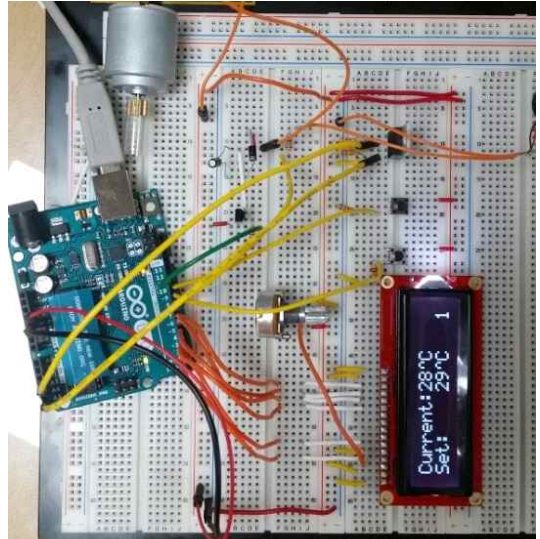
그동안의 실험들과 프로젝트 진행을 통해서 알게 된 점은 USB를 통해 어떻게 컴퓨터와 정보를 주고받을 수 있는지, WIFI를 통한 통신을 어떻게 이루어지는지와 필요한 것들이 무엇이 있는지를 알 수 있었다. 이들을 토대로 사용자의 입장에서는 작동여부에 대해서만 관심이 있었다면 이번 수업을 들으면서 모든 기기들의 작동원리를 생각해보고 응용 할 수 있는 부분들을 추가로 생각해 볼 수 있었으며, IoT는 사물과 인터넷의 연결이 반드시 이루어져야 한다고 생각했으나 인터넷의 연결 없이 사물과 사물간의 통신을 이루는 것도 IoT라는 것을 알 수 있었다. 회로를 구현하면서 통신과정을 직접 확인해 볼 수는 없지만 각 센서들이 어떤 통신을 이용하여 정보를 주고받는지, 주고받은 결과 값들이 무엇인지 시리얼 모니터를 통해 확인해 볼 수 있었다. 또한 마이크로 프로세서 강의 시간 때에는 하나의 아두이노와 몇 개 되지 않는 센서들을 이용하여 컨트롤 해봤다면 이번에는 두 개의 아두이노로 무선통신을 활용하여 서로 데이터를 주고 받으며 센서를 제어해 볼 수 있었다. 또한 아두이노에 쉴드를 여러 개 입혀서 많은 센서들을 활용해 보았다.

IoT가 우리의 실생활에 많은 부분을 차지하고 있다는 것을 체감하고 있는 요즘 직접 하드웨어를 컨트롤 해보고 코딩해봄으로써 좋은 기회가 되었던 것 같다. 이러한 기술이 더 많은 사람들이 사용하도록 상용화 되기까지는 아직도 많은 시간이 남았다고 생각하지만 그동안 계속해서 공부해보고 실험해봄으로써 더욱더 관심을 갖고 알아가려고 노력하는 자세를 갖추어야 겠다.

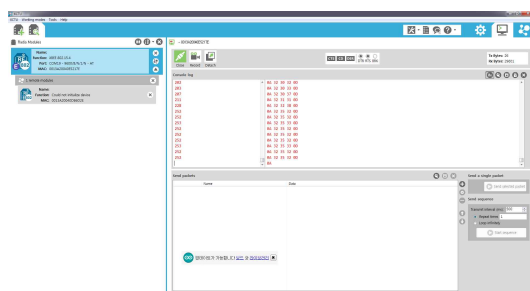
부록(시제품 사진)



사진내용 : 스피커를 통한 소리출력



사진내용 : LCD를 통한 문자출력

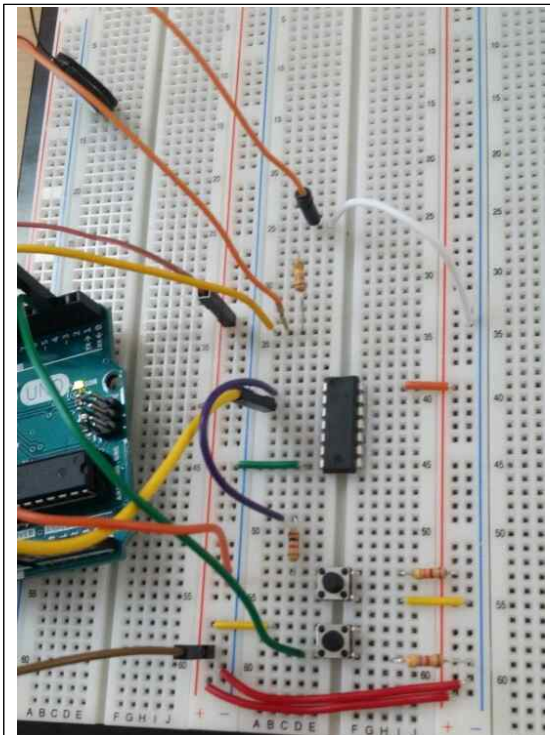


사진내용 :
무선통신을 하기 위한 프로그램 시현

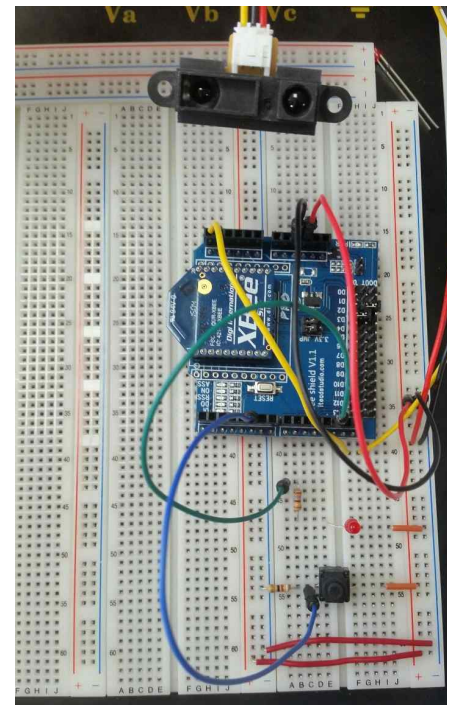


지으남아 마녕우와hdkssydhahahahaha

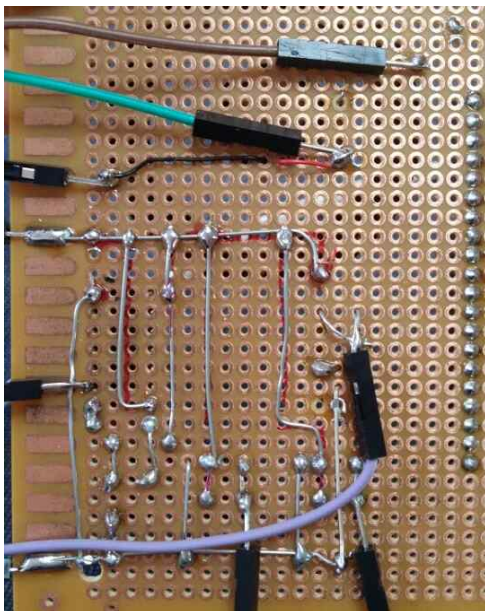
사진내용 :
시리얼 모니터를 통한 결과값 출력



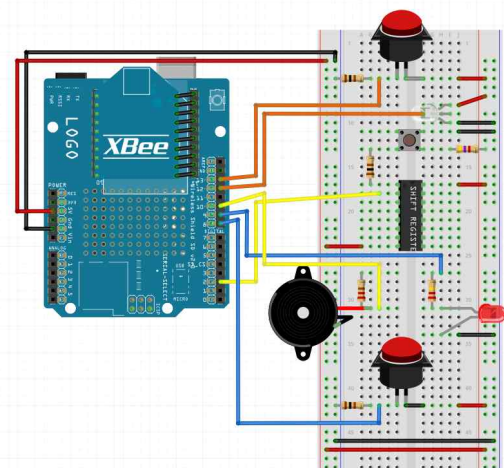
사진내용 : 브레드보드 회로구현 _목걸이



사진내용 :
XBee를 통한 무선통신 회로구현 _ 배찌



사진내용 : 목걸이와 배찌를 제품화하기
위해서는 회로를 단순화 해야 한다.
납땀을 통해 회로선들을 연결 한 후
센서들을 작동시킬 수 있었다.



사진내용 :
firtzing 프로그램을 이용한 회로구현 _
직접실험하지 못할 때 컴퓨터를 통한
회로구현

배찌 최종 코드

```
// We'll use SoftwareSerial to communicate with the XBee:
#include <SoftwareSerial.h>
#include <SD.h>

// XBee's DOUT (TX) is connected to pin 2 (Arduino's Software RX)
// XBee's DIN (RX) is connected to pin 3 (Arduino's Software TX)
SoftwareSerial XBee(2, 3); // RX, TX

const int ir=A5; //적외선
const int buttonPin = 5; //XBee 통신 버튼
const int ledPin = 13;
const int CS_PIN= 4;
const int POW_PIN=8; //전원을 공급

int buttonState = 0;
int state = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  Serial.println("Initializing Card");
  pinMode(CS_PIN, OUTPUT); //CS 핀을 출력으로 설정
  pinMode(POW_PIN, OUTPUT); //8번 핀을 전원으로 사용하므로 출력으로 설정
  digitalWrite(POW_PIN, HIGH); //아두이노 공식 이더넷 실드를 사용하면 주석 처리

  if(!SD.begin(CS_PIN))
  {
    Serial.println("Card Failure");
    return;
  }
  Serial.println("Card Ready");
}

void loop()
{
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH){
    state=1-state;
  }

  if(state==1)
  {
```

```

XBee.begin(9600);
digitalWrite(ledPin, HIGH); //XBee통신이 되고 있다는 것을 알려줌
int distance=analogRead(ir); //IR 센서에서 장애물 거리값을 읽음
int range = (6787 / (distance - 3)) - 4; //IR 센서 값을 CM로 변환
delay(100); //장애물을 감지하는 시간 간격 지정

long timeStamp=millis();
File dataFile=SD.open("log.csv", FILE_WRITE);

if(dataFile)
{
    dataFile.print(timeStamp);
    dataFile.print(",");
    dataFile.println(range);
    dataFile.close(); //close() 함수로 연결을 끊어야만 실제로 데이터가 입력

//디버깅을 위해 화면에 동일한 문자열 입력
    Serial.print(timeStamp);
    Serial.print(",");
    Serial.println(range);
}
else
{
    Serial.println("Couldn't open log file");
}

// 사람이 20~80CM의 범위로 다가오면
if (range>=20&&range<=80)
{
    int a=1;
    XBee.write(a);
}
delay(100);
}

else if(state==0)
{
    digitalWrite(ledPin, LOW); //XBee통신을 하고 있지 않다는 것을 알려줌
    int b=0;
    XBee.write(b);
    XBee.end();
}
delay(100);
}

```

목걸이 최종 코드

```
#include <Adafruit_NeoPixel.h>
#include <SoftwareSerial.h>

#define PIN 11 //NeoPixel 핀 번호
const int BUTTON_INT= 0; //interrupt를 위한 버튼
uint16_t i=15;
const int buttonPin = 10; //NeoPixel 전원 버튼
const int btn=12; //XBee통신 전원 버튼
const int led=13;
const int SPEAKER=9;
int btnstate=0;
int bstate=0;
int buttonState = 0;
int state = 0;
int a;

//네오픽셀을 사용하기 위해 객체 하나를 생성한다.
//첫번째 인자값은 네오픽셀의 LED의 개수
//두번째 인자값은 네오픽셀이 연결된 아두이노의 핀번호/
//세번째 인자값은 네오픽셀의 타입에 따라 바뀌는 flag
Adafruit_NeoPixel strip = Adafruit_NeoPixel(16, PIN, NEO_GRB + NEO_KHZ800);

SoftwareSerial XBee(4, 5); // RX, TX

void setup()
{
  pinMode(buttonPin, INPUT);
  pinMode(led, OUTPUT);
  pinMode(btn, INPUT);
  strip.begin(); //네오픽셀을 초기화하기 위해 모든LED를 off시킨다
  strip.show();
  attachInterrupt(BUTTON_INT, speaker, RISING); //interrupt
  Serial.begin(9600);
}

//interrupt 함수
void speaker()
{
  //state가 1이면 경고음을 약 5초 동안 출력
  if(state==1)
  {
    tone(SPEAKER, 400, 5000);
  }
}

void loop()
{
  btnstate = digitalRead(btn);
```

```

    if (btnstate == HIGH)
    {
        bstate=1-bstate;
    }

//XBee 통신 시작
    if (bstate==1)
    {
        XBee.begin(9600);
        digitalWrite(led, HIGH);

        if (XBee.available())
        {
            a=XBee.read();
            if(a==1)
            {
                Serial.println(a);
                tone(SPEAKER, 400);
                colorWipe(strip.Color(255, 0, 0), 50); //빨간색 출력
                colorWipe(strip.Color(0, 0, 0), 50); //조명 꺼짐
                theaterChase(strip.Color(255, 0, 0), 50); //빨간색 출력
                colorWipe(strip.Color(255, 0, 0), 50); //빨간색 출력
            }
        }
        noTone(SPEAKER);
        delay(100);
    }

//XBee 통신 종료
    else if (bstate==0)
    {
        Serial.println(a);
        XBee.end();

        buttonState = digitalRead(buttonPin);
        digitalWrite(led,LOW);

        if (buttonState == HIGH)
        {
            state=1-state;
        }

//state가 1이면 NeoPixel에서 흰색 조명 출력
        if(state==1)
        {
            for( i=0; i<strip.numPixels(); i++){
                strip.setPixelColor(i, 255,255,255);
                strip.show();
            }
        }
        delay(100);
    }

```

```

}

//state가 0이면 NeoPixel 조명 꺼짐
else if(state==0)
{
    for( i=0; i<strip.numPixels(); i++){
        strip.setPixelColor(i, 0,0,0);
        strip.show();
    }
    delay(100);
}

}

delay(100);
}

//NeoPixel에 달린 LED를 각각 주어진 값 색으로 채워나가는 함수
void colorWipe(uint32_t c, uint8_t wait)
{
    for(uint16_t i=0; i<strip.numPixels(); i++){
        strip.setPixelColor(i, c);
        strip.show();
        delay(wait);
    }
}

//입력한 색으로 LED를 깜빡거리며 표현하는 함수
void theaterChase(uint32_t c, uint8_t wait)
{
    for (int j=0; j<10; j++){
        for (int q=0; q < 3; q++){
            for (int i=0; i < strip.numPixels(); i=i+3)
            {
                strip.setPixelColor(i+q, c);
            }
            strip.show();
            delay(wait);
            for (int i=0; i < strip.numPixels(); i=i+3){
                strip.setPixelColor(i+q, 0);
            }
        }
    }
}

```