

< 센서 >

1) LED 구조 : 빛의 색깔은 크리스탈 도핑(Crystal doping)의 양과 종류에 따라 달라진다

-10~20mA의 전류, 1.5V~2.5V의 전압강하

-다리가 긴 부분이 양극(Anode), 짧은 쪽이 음극(Cathode)이다.

-정적 구동방식

-동적 구동방식(매트릭스형) ex) 전광판

<3주차 소리 센서>

<조도 센서> : 주변 환경의 밝기를 측정할 수 있는 센서

광에너지(빛)를 받으면 내부에 움직이는 전자가 발생하여 전도율이 변하는 광전효과를 가진 소자활용

-극성이 존재하지 않는(+ - 한 개만 존재) 가변저항

-cds센서가 저항이기 때문에 빛의 양이 매우 많으면 저항수치가 과전류가 흐를 수 있음

-1옴 : 1V의 전압으로 1A의 전류가 흐를 때의 저항

-밝은 곳에서는 내부 저항이 작아지는 광 가변 저항기이다.

<ULTRASONIC SENSOR 제어 프로그램 > - 초음파 센서

-먼지, 연기, 수증기 등과 같은 환경 조건이 센서에 영향을 줄 수 있는 산업분야에서 비 접촉식 위치 및 거리 측정

-초음파 센서는 인간의 가청 범위 이상의 고주파를 사용

-초음파가 센서와 측정 물체 사이를 통과하는데 소요되는 시간으로 거리 또는 위치 값을 계산

-초음파 센서는 파장이 짧고 주파수가 높다 (20kHz ~ 200Mhz)

-trigger핀 센서가 신호를 발생하면 8개의 구형파에 의해 초음파가 발생한다. Echo핀을 LOW에서 HIGH로 변경하고 송신했던 초음파가 물체에 반사되어 돌아올 때까지 기다린다. 초음파 신호가 되돌아오면 Echo핀은 신호를 다시 LOW로 변경한다. HIGH로 유지하고 있었던 시간이 얼마정도 인가?에 따라 거리를 측정한다.

<가변 저항>

-스위퍼가 달려 있는 가변 저항은 반시계 방향으로 회전하면 저항이 커지고 시계방향으로 돌리면 저항이 작아진다.

-저항은 극성이 없다. (전원과 GND를 바꿀 순 있지만 값이 반대로 바뀐다.

-

<블루투스 제어 프로그램 실습>

-목표 : 저 전력 ↔ wifi- CSMA/CA가 전력을 많으쓰는 것과 반대임

-일반적으로 저전력의 목표에 맞게 데이터 채널, 음성 채널에 쓰인다.

-전력 모드

1) standby mode : 대기상태로 블루투스유닛이 피코넷 범위에 들어 왔을 때(미접속상태)의 상태이다.

2) page mode : 상대방 MAC 주소를 알 경우 1.28초 간격으로 page 패킷 보냄

3) inquiry mode : 마스터가 기기의 주소를 알지 못할 때 주변장치에게 문의한다.(디바이스 정보 획득

4) active mode : 마스터가 슬레이브가 연결을 설정하고 데이터 송수신 하는 것

5) park mode : 전력소비를 줄이기 위한 것, 마스터와 동기화 되어있지만 데이터 주고받지 않는 상태

6) hold mode : 소비전력을 줄이기 위해 데이터 전송을 중지 하는 상태

7) sniff mode : 슬레이브 유닛에만 적용되며 데이터를 전송하지는 않고 송신되는 데이터만 점검한다.

-기기간의 연결은 paging과 inquiry 절차로 이루어 진다.

-상대방 기기의 고유주소(BD_ADDR)를 알 경우 paging만으로 연결이 성립이 되나 그렇지 않으면 inquiry로 상대방 기기의 고유주소를 우선 확인한 후 paging으로 연결을 성립한다.

-최대 전송 속도 10Mbps, 최대 전송 거리 10m~ 100m

-2.4GHz의 주파수 사용

-주파수 호핑 방식의 스펙트럼 확산(FHSS) 기술 사용

(1Mhz폭의 채널 79개를 사용, 초당 최대 1600회까지 채널 바꿈) - 스펙트럼을 주파수(1MHZ),채널(79개)로 분할한다.

➔ 마스터 기기가 생성하는 주파수 호핑 방식에 슬레이브 기기를 동기화 시켜야 한다.

-피코넷 형성 : 한 개의 마스터가 7개까지의 active(동시 접속) 모드 슬레이브와 255개 까지의 park 모드 슬레이브로 구성

-스캐터넷 : 2개 이상의 피코넷으로 구성 (100개 까지의 피코넷 연결가능)

하나의 피코넷 마스터는 다른 피코넷의 슬레이브가 될 수 있음

-슬레이브는 time division 방식으로 다른 피코넷에 참여 가능

Time division : physical channel은 625us의 time slot으로 분할

-Master는 짝수 번째 slot으로 slave에 packet 송신

-slave는 홀수 번째 slot으로 master에 packet 송신

-패킷의 길이는 1, 3, 5 인데 1은 단순한 signaling할 때 쓰고, 3, 5는 데이터를 보낼 때 쓴다.

<블루투스 4 부터는 inquiry, paging 안한다>

Advertisement interval이 있어서 슬레이브 장치가 주기적으로 자신의 존재를 알리고 마스터는 스캔한다. (BLE 상태에서의 페어링)

-각 블루투스 기기는 고유의 시스템 clock을 가지고 있다.

-피코넷의 동기화는 마스터의 시스템 clock에 의해 이루어지고, 슬레이브는 자체 시스템 clock을 마스터의 clock에 일치시킨다.

§ 슬레이브는 time-division 방식으로 다른 피코넷에 참가 가능

§ Master는 짝수 번째 Slot으로 Slave에 Packet 송신

§ Slave는 홀수 번째 Slot으로 Master에 Packet 송신

Packet의 길이는 1, 3, 5 (Length of Time Slot)

§ 패킷 다중화 방식 : TDD(Time Division Duplex)

<라즈베리파이>

<시리얼 통신>

-SPI 통신

-1대 다수의 통신을 지원하는 동기식 통신 (다수의 통신을 위해서 다수의 선이 필요함)

-동시에 송수신이 가능함

-i2c에 비해 속도가 빠름

-컴퓨터 메인보드에 연결된 랜카드, 그래픽카드 등등 끼리의 1대 다수 통신이 spi 통신임

1) SCK : 클럭 전송을 위한 단자로, 마스터에서 슬레이브로 클럭을 전송

2) MOSI : Master->slave 데이터 전송을 위한 단자로, 클럭을 전송하면 마스터에서 슬레이브로 데이터를 보냄

3) MISO : Slave->Master 데이터 전송을 위한 단자로 클럭을 전송하면 슬레이브에서 마스터로 데이터를 보냄

4) SS : 마스터 장치에서 슬레이브 장치를 선택하기 위한 단자

(슬레이브의 개수만큼 SS 선이 늘어남)

-CAN 버스 보다 비교적 짧은 거리에서 동작함

CAN 통신 (Controller Area Network): 자동차에서 사용 - 직렬 통신

-I2C 통신 :

- 데이터 송수신을 위한 선 : SDA (Serial Data)
- 송수신 타이밍 동기화를 위한 클럭 선 : SCL(Serial Clock)
- 하나의 Master, 하나 이상의 Slave(slave는 최대 127개)

-라즈베리파이의 GPIO는 아두이노와는 다르게 디지털 입력만 가능하다

(아날로그는 디지털, 아날로그 핀이 둘다 있음)

-따라서 라즈베리파이는 ADC(Analog Digital 컨버터)를 사용해야 한다.

-SPI는 디지털만 인식할 수 있으므로, MCP3208이라는 ADC 컨버터를 SPI 인터페이스로 사용해서 아날로그 값을 디지털로 바꿔서 SPI로 전달한다.

-MCP3208 : 8 개의 채널을 가지고 있고 센서의 아날로그 출력 값을 12비트의 디지털 값으로 변환하는 ADC 칩이다. 예로, 0 ~ 3.3V의 센서 출력 값을 0 ~ 4095(12BIT)의 디지털 값으로 변환해준다.
(여기서 8개의 채널이란 의미는 CH0 ~ CH7 까지의 포트중에서 선택해서 Analog Inputs값을 받을 수 있다는 의미이다.)

-라즈베리파이는 GPIO, SPI, I2C, UART 통신 포트를 가지는데 ADC 기능이 없어서 MCP3208 ADC칩을 이용하여 외부에 ADC 기능을 가지도록 한다.

-라즈베리파이와 MCP3208간에는 SPI 통신으로 ADC 데이터를 가져온다.

<스텝 모터> ←->DC 모터는 : 강한 파워

-펄스 모양의 전압에 의해 일정 각도 회전하는 모터

(높은 정밀도를 요구하는 프린터나, 디스크드라이버, 테이프 드라이버, 로봇의 관절에 사용)

-회전 각도는 입력 펄스의 신호 수에 비례한다. 회전 속도는 입력 펄스 신호의 주파수에 비례

스텝 모터 구동 원리)

스테핑 모터 : 펄스 모양의 전압에 의해 일정 각도 회전하는 전동기

예) 스텝 각이 1,8도인 모터에 펄스를 100개 인가하면 모터는 180도 회전

1) single coil excitation (싱글 코일 여자 방식) : 한번에 코일 하나에만 전류를 공급

2) Full step drive(풀 스텝 구동)

스텝모터는 펄스를 각 각도로 변환하여 주는 부품이다.

스텝모터에 어떤 펄스 신호를 주게 되면 스텝모터는 그 펄스 신호에 맞게 각을 틀어 움직인다.

<IoT 개념 이론>

Ex. 구글 인터넷 지도 : 교통량

->휴대폰의 센서를 통해 데이터 수집한 후 클라우드 환경에서 분석하여 교통량을 나타냄
데이터를 수집하고 가공, 분석하는 것이 머신러닝 개념이다.

사물인터넷의 사례 : ex. 스마트 홈 , 스마트 팜

<기존의 센서 네트워크>

센서들 끼리 구성된 센서네트워크는 센서노드와 싱크노드로 구성이 되는데, 싱크노드는 일정량의 버퍼가 있다. 버퍼가 일정 부분 이상 차 버리면 버퍼 오버플로우가 일어난다. 그리고 싱크노드는 호스트 컴퓨터와 연결되어 있고, 컴퓨터는 데이터 베이스와 연동이 된다.

->사물인터넷(모든 디바이스가 직접적으로 네트워크에 연결됨) 과 센서 네트워크는 개념이 다르다

지금은 엣지 노드(컴퓨터)가 데이터를 감당을 못해서 '클라우드' 를 쓴다.

<소규모 IOT 시스템의 기능>

디바이스	<->	브로커	<->	대시보드
데이터 인식과 수집 (아두이노)		메시지 전송과 교환 MQTT 프로토콜		통제와 연계 (모니터링)

블루투스, Lora와 같은 NON-ip things는 IOT server와 서버와 통신하기 위해 중간에 Gateway가 필요하다. (ip가 없기 때문에)

사물인터넷 시스템 구성에 필요한 기술

- 1) 데이터 인식과 수집 기능 : Arduino
- 2) 메시지 전송과 교환 기능 : 브로커 (MQTT 프로토콜) – 약속되어 있는 규약대로 해야 해서 (MQTT : IoT 메시지 교환의 표준 – 다양한 디바이스가 존재하는 IoT 네트워크의 복잡성 극복위해)
- 3) 대시 보드 : 브로커에 접속된 디바이스들을 모니터링하고 통제하기 위해.

->ex. 클라우드 : 외부의 AI 엔진과 연계하여 지능화된 결과 추출

->외부의 IT 시스템과 연계, 빅 데이터 생성

4) 대시 보드의 플랫폼 : Node-RED

->디바이스의 메시지 발신을 노드를 통해 처리하고 대시보드에 보여줌

->팅커캐드 처럼 사용하면 됨.

<MQTT> : Message queue telemetry(원격) transportation : 메시지 원격 전송

아두이노 환경에서 사용할 수 있는 무선 통신은 와이파이, 지그비, 블루투스, 저전력 블루투스 통신 (BLE), LoRa 등이 있다.

와이파이 모듈인 ESP8266 보드는 와이파이의 Station(STA) 모드와 Access point(AP) 모드를 지원한다.

-ap 모드는 esp8266을 중심으로 다른 기기가 네트워크에 접속하도록 한다

-station 모드는 esp8266이 이미 구성되어 있는 네트워크에 접속하는 것이다.

MQTT :

-다양한 디바이스가 존재하는 IoT 네트워크의 복잡성 극복을 위해 간편한 IoT 기기용 프로토콜이다.

-인터넷 TCP/IP 방식으로 작동한다. (게이트 웨이가 필요없는 IP 방식)

-MQTT는 브로커 based이고, '허브와 스포크' 방식의 토폴로지를 사용한다. (중심에 브로커, 각각의 스포크에는 센서나 디바이스가 있다)

-중심에서 데이터의 송수신을 관장하는 서버는 MQTT 브로커라고 한다.

-MQTT는 송수신하는 정보를 토픽(데이터 이름)과 페이로드(데이터 내용)으로 구분하여 관리한다.

EX) 온도 센서가 달려 있는 마이크로컨트롤러는 temp라는 토픽에 "25"라는 페이로드를 묶어 MQTT 브로커에 송신할 수 있다.

10-3 17분 그림 참고

<클라우드>

센서들이 수집하는 데이터의 양이 너무 많아서 개인의 PC에서 처리하기가 어려워서 클라우드에서 처리한다. 하지만, 클라우드까지 가서 처리하는 시간이 너무 많이 걸려서,

Things->Edge cloud -> Fog cloud -> cloud 로 구성되어 있다.

클라우드 컴퓨팅의 3모델

1) IaaS (infrastructure as a service) : 네트워킹 기능, 컴퓨터 및 데이터 스토리지 공간을 제공

(가상화된 서버, 스토리지, 네트워크)

2) paas (platform as a service) : 서비스로서의 플랫폼을 사용하면 조직은 기본 인프라를 관리할 필요가 없어 애플리케이션 개발과 관리에 집중할 수 있다.

3) saas : 대시보드 처럼 플랫폼 위에 서비스만 관리를 한다.

(서비스로서의 어플리케이션 - 소프트웨어를 가상화하여 제공)

(11주차 1강 16분 그림)

-AWS 형태 (22분 그림)

1) Device gateway : MQTT와 HTTP와의 커뮤니케이션을 위한 GATEWAY

<클라우드 제공 형태>

1) public clouds : 서비스 - 인터넷

2) private clouds : 데이터 센터 - 인트라넷

3) Hybrid clouds : private, public

<최종 정리>

아두이노 : 하드웨어 모듈

라즈베리 파이 : 소형 컴퓨터를 대신하는 하나의 OS

wiringpi.h : gpio input의 번호를 정확히 달기 위해

thinger/thinger.h : cloud를 쓰기 위해

wiringpispi.h : 시리얼 통신

pthread.h : cloud를 하면서 사용

스레드

-프로세스는 적어도 하나의 스레드를 가짐 (ex. main)

-여러 개의 프로세스를 동시에 쓰는 것을 멀티프로세스라고 한다. 그리고 하나의 프로세스는 멀티 스레드를 가지고 있다(ex. 하나의 프로세스의 code내에서 main함수와 다른 함수간(cloud를 정의한 함수)의 관계

시험

-coding은 없다

아두이노는 OS가 아니라서 혼자서 동작하는 것이 불가능하다 (OS가 없어서 단일 작업을 한다)

->따라서 PC와 USB Serial 통신을 통해서 통신해야 한다.

->Serial 통신은 2개의 통신 선 (TX - digital 0번 핀, RX -digital 1번 핀)를 사용

라즈베리파이는 OS 라서 혼자서 동작하는 것이 가능하다

->CPU, 메모리, 입출력 포트, 네트워크 장치를 내장하고 있다.

->라즈베리파이는 운영체제를 통해 다수의 작업이 멀티태스킹 방식으로 동시에 진행

->라즈베리파이에는 ADC가 내장되어 있지 않기 때문에 아날로그 입력을 위한 센서 등을 장착하기 위해 별도의 칩이 필요 (MCP3208)

시리얼 모니터 (IDE)

-아두이노 보드에서 PC로 시리얼 통신을 하는 것이다 (TX, RX)

-아두이노는 baud rate를 쓴다

*bit rate : 초당 데이터 비트(0, 1)을 얼마만큼 전송할 수 있는지 ? bps임

*baud rate : 초당 펄스 수

-즉, 초당 심볼(의미 있는 데이터 묶음)을 얼마만큼 전송할 수 있는지?

ex) 4bit 데이터(하나의 의미있는 심볼) 를 초당 200개 보내는 경우

bps : 4bit * 200 = 800bps , baud rate : 200bd

Serial.begin(speed) : 시리얼 통신을 통해 연결된 두 장치는 같은 통신 속도를 가져야 함

Serial.print(" ") , Serial.println(" ") : serial.print랑 동일하지만 줄바꿈한다.

단일 저항값 = (입력 전압 - 구동 전압) / (구동 전류 - A)

ex) 아두이노는 5v를 공급해주고, LED는 2V, 20ma를 소비

=> (5 - 2) / 0.02 = 150옴

-라즈베리파이와 MCP3208간에는 SPI 통신으로 ADC 데이터를 가져온다 (8개 채널, 12비트 해상도)

-라즈베리파이는 GPIO를 WiringPi와 연결을 시킨다. (핀 번호가 다름)

mcp3208그 복잡한 코드는 시험에 안나온다.

-블루투스 : CSMA/CA를 쓰지않고 주파수 호핑 방식의 스펙트럼 확산(FHSS)를 쓴다.

