
NodeJS와 MongoDB를 활용한 모터 동작 제어시스템 구현

강진영, 이영동

창신대학교 컴퓨터소프트웨어공학과

Implementation of motor control system using NodeJS and MongoDB

Jin Young Kang, Young-dong Lee

Department of Computer Software Engineering, Changshin University

E-mail : ydlee@cs.ac.kr

요 약

사물인터넷이 고도의 지능화가 됨에 따라 다양한 분야로 적용되고 있다. 그중에 센서-서버-DB를 결합해 하나의 플랫폼을 만들어서 다양한 분야에 접목하면 데이터를 쉽게 관리 할 수 있는 사물인터넷 기반의 시스템이 구현 가능하다. 이에 따라, 본 논문에서는 사물인터넷 기반 NodeJS와 MongoDB를 활용하여 스마트폰 기울기에 따른 서보모터 동작 제어 시스템 구현을 위해 Raspberry Pi, 서보모터, 스마트폰으로 시스템을 구성하고, NodeJS, MongoDB, npm모듈, HTML을 활용한 서보모터 센서 값이 발생하면 NodeJS로 전송하여 데이터를 저장 관리 할 수 있도록 하였다.

ABSTRACT

With the development of intelligent technologies, the Internet of Things(IoT) has been applied to various applications. A platform technology including a sensor-server-DB for easily managing data at a remote site is required. In this paper, we implemented a servo motor control system that moves by the smart phone tilt value using NodeJS and MongoDB. The system consists of Raspberry Pi, servo motor and smart phone and the servo motor sensor data is transmitted to NodeJS so that data can be stored in database.

키워드

Raspberry Pi, NodeJS, MongoDB, Motor control system

I. 서 론

사물인터넷이 고도의 지능화가 됨에 따라 다양한 기술력들이 발전하고 있다. 그중에 센서-서버-DB를 연결해 하나의 플랫폼을 만들면 다양한 아이디어와 결합 할 수 있고, 다양성 및 생산성이 증가함에 따라 많은 데이터들을 쉽게 관리 할 수 있다.

본 논문에서는 NodeJS[1,2]를 사용하여 서버를 구축하고 스마트폰 기울기센서를 이용해서 Servo Motor 센서의 방향을 조작함에 따라 발생한 데이터 값을 MongoDB[3]에 저장한다. 이로 인해 사물인사물인터넷 기반 서비스 플랫폼은 다양한 센서를 부착하여 데이터 값을 저장 시키거나 관리 할 수 있고, 웹 브라우저를 통한 원격조작이나 모

니터링 할 수 있게 개발이 가능하다.

II. 시스템 구성

2.1 Raspberry Pi

리눅스 기반으로 동작하는 오픈소스 하드웨어이며 브로드컴사에서 만든 BCM2835[4] 라고 불리는 고성능 칩이다. 키보드나 마우스를 연결 할 수 있는 USB포트가 있고 이더넷 포트, 스피커잭, HDMI포트, SD카드 소켓 등을 가지고 있다. Raspberry Pi에 모니터를 연결하면 소형 데스크탑 컴퓨터처럼 사용 할 수 있어 활용도가 높아 개발 및 학습용으로 많이 사용되어지고 있다.

2.2 NodeJS

NodeJS는 자바스크립트 언어를 사용 하였으며 서버구축이나 설계 및 개발을 위해 이벤트 처리 I/O 프레임워크로 웹 서버나 네트워크 프로그램을 제작하기 위해 개발 되었다. 기존에는 여러 스레드를 가지고 양이 많은 데이터를 처리 할 때 스레드를 증식 시켜 일을 처리 했지만 여러 가지 문제점들이 있었다. NodeJS는 이벤트 처리방식이며 스레드가 대기하고 있다가 처리 하라는 요청이 들어오면 그때 가서 처리하는 방식으로 자원 낭비 없이 효율적인 일 처리가 가능 하며 대규모 서버응에 적합하다.

2.3 MongoDB

데이터베이스의 한 종류로써 데이터 값을 저장하는 역할을 하는 도큐먼트 지향 데이터베이스 시스템이라 한다. JSON형식이며 NoSql이어서 다른 RDB와는 다르게 서로 유기적인 관계가 없다. MongoDB는 Table 대신 Collection을 사용하기 때문에, RDB방식의 테이블 속성 값에 맞는 데이터를 넣어야 하는 것과는 다르며, 제약 없이 마음대로 데이터를 넣을 수 있다.

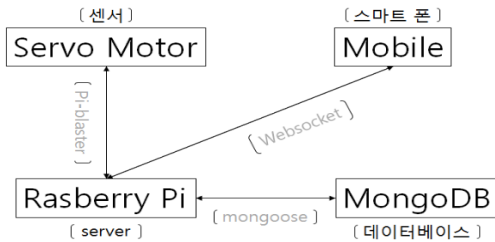


그림 1. 전체 시스템 구성도

III. 시스템 구현 및 결과

그림 1은 서비스 플랫폼을 구축하기 위한 시스템 구성도이다. Raspberry Pi에 NodeJS를 설치하여 서버로 사용하였고, 스마트폰에서 기울기 센서를 통해 발생한 데이터 값을 Websocket 모듈을 통해 서버(Raspberry Pi)로 전송 한다. 서버는 받은 데이터 값을 Pi-blaster 모듈과 mongoose모듈을 통해 각자 센서(Servo Motor)와 MongoDB로 데이터 값을 전송 한다. 이로 인해 센서가 동작 하게 되고 MongoDB는 시간과 데이터 값을 함께 저장한다. 사물인터넷 기반 서비스 플랫폼은 다양한 센서를 부착하여 데이터 값을 저장 시키거나 관리 할 수 있고, 웹 브라우저를 통한 원격조작이나 모니터링 할 수 있게 개발이 가능하다.

그림 2는 Express, HTTP 모듈을 사용하여 서버를 구축하였고, 스마트폰이 웹 브라우저에 접속

할 경우 Websocket[5]을 통하여 Raspberry Pi와 통신 할 수 있도록 하였다.

```

var express = require('express');
var app = express();
var server = require('http').createServer(app);
var fs = require('fs');
var WebSocketServer = require('websocket').server;
var piBlaster = require('pi-blaster.js');

var clients = [];

var mongoose = require('mongoose');
mongoose.connect('mongodb://localhost/appserve');

var db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function callback(){
  console.log("Connected to mongod server");
});

//Getting request from browser
//and send response back
app.get('/', function(req, res){
  fs.readFile('ws.html', 'utf8', function(err, text){
    res.send(text);
  });
});

//listening to Port 1337 for incoming messages
server.listen(1337, function (){
  console.log(new Date() + " Server is listening on port 1337... *");
});

websocket = new WebSocketServer({
  httpServer: server
});
    
```

그림 2. 서비스 플랫폼 모듈 및 MongoDB 연결

그림 3은 x, y, z축을 통해 스마트폰의 기울기 정도를 파악 할 수 있는 코드이며 스마트폰을 통하여 웹 브라우저에 접속하게 되면 Connecting...이라는 문구가 표시되고 스마트폰의 기울기 정도에 따라 생성되는 데이터 값이 달라진다.

```

<script type="text/javascript">
(function () {
  // Position Variables
  var x = 0;
  var y = 0;
  var z = 0;

  // Speed - Velocity
  var vx = 0;
  var vy = 0;
  var vz = 0;

  // Acceleration
  var ax = 0;
  var ay = 0;
  var az = 0;
  var a1 = 0;
  var aAlpha = 0;
  var aBeta = 0;
  var aGamma = 0;

  var delay = 200;
    
```

그림 3. 웹 브라우저 원격조작

그림 4는 스마트폰에서 발생한 데이터값을 라 Raspberry Pi로 입력을 받으면 pi-blaster 모듈을 통해 센서 역할을 하는 Servo Motor에 명령을 주어서 데이터가 2보다 크면 왼쪽(24%), 데이터가 0이면 중간(15%), 데이터 2보다 작으면 오른쪽(6%)으로 날개가 움직이게 된다.

그림5는 Raspberry Pi와 MongoDB가 서로 연결되어 있어 데이터 값이 흘러 들어오면 MongoDB에 저장 될 수 있게 하였다. 이는 데이터 관리를 쉽게 하고 다른 디바이스로 모니터링 할 때 편리하게 추출 하여 사용 할 수 있다.

