

<https://doi.org/10.7236/IIIBC.2017.17.5.151>

IIIBC 2017-5-21

사물 인터넷 기반 스마트 응용의 설계

Design of the Smart Application based on IoT

오선진*

Sun-Jin Oh*

요 약 무선 기반 네트워크와 인터넷 기술의 급속한 발전과 더불어, 주변의 모든 사물들이 인터넷으로 연결되어지는 초 연결 사회로 발전해 가고 있으며, 최근 사물인터넷에 기반한 스마트 응용 개발이 활발하게 이루어지고 있다. 사물인터넷은 사람, 사물, 공간, 데이터 등이 유무선 네트워크로 연결되어, 정보의 생성, 수집, 공유 및 활용을 할 수 있는 초 연결망을 구축하는 것으로 여기에 다양한 센서들과 클라우드 기술을 이용하여 실시간 상황정보를 수집 저장하여 스스로 판단하고 결정할 수 있는 지능을 부여하여 서비스화 하는 스마트 응용 개발이 주류를 이루고 있다. 본 연구에서는 이러한 시대적 조류에 맞춰 사물인터넷을 기반으로 하는 모바일 컴퓨팅 환경에서 센서와 클라우드 등 첨단 ICT 기술을 이용하여 특정 상황정보의 실시간 탐지와 인식을 바탕으로 현재 상황에 맞는 정확한 판단과 결정을 내릴 수 있는 지능을 기반으로 실시간 제어와 통제 및 처리를 할 수 있는 스마트 응용을 설계하고자 한다.

Abstract With the rapid growth of the up-to-date wireless network and Internet technologies, huge and various types of things around us are connected to the Internet and build the hyper-connected society, and lots of smart applications using these technologies are actively developed recently. IoT connects human, things, space, and data with various types of networks to construct the hyper-connected network that can create, collect, share and applying realtime information. Furthermore, most of the smart applications are concentrated on the service that can collect and store realtime contexts using various sensors and cloud technology, and provide intelligence by making inferences and decisions from them nowadays. In this paper, we design a smart application that can accurately control and process the current state of the specific context in realtime by using the state-of-the-art ICT techniques such as various sensors and cloud technologies on the IoT based mobile computing environment.

Key Words : Internet of Things(IoT), Sensor & Cloud Technology, Smart Application

1. 서 론

최근 무선기반 네트워크를 중심으로 하는 통신기술과 인터넷 응용기술의 급속한 발전과 스마트 폰 및 태블릿 PC와 같은 휴대용 단말기술의 대중화와 더불어, 정보화 사회는 주변의 모든 사물들이 인터넷을 중심으로 연결되

어지는 초 연결 사회로 급속히 발전해 가고 있으며, 인간 간의 소통뿐 아니라 인간과 사물 그리고 인간을 배제한 사물 간 직접 소통 형태의 새로운 패러다임으로 그 변화가 현실화되고 있다. 이와 같이 인터넷을 중심으로 “초 연결 사회”를 구성하고 서로 다양한 형태의 정보를 생산·가공·처리·유통과정을 통해 기존 재화의 가치를 높이는

*종신회원, 세명대학교 정보통신학부
접수일자 : 2017년 7월 13일, 수정완료 : 2017년 10월 10일
게재확정일자 : 2017년 10월 13일

Received: 13 July 2017 / Revised: 10 October, 2017 /

Accepted: 13 October, 2017

*Corresponding Author: sjoh@semyung.ac.kr

Dept. of Computer & Information Science, Semyung University, Korea

새로운 유형의 ICT 기술을 “사물 인터넷”이라 한다.^[1] 이러한 변화는 우리의 기본적인 삶의 모습뿐만 아니라 주변의 환경, 안전, 공공 서비스 등 다양한 분야의 모습을 바꾸고 있다.^[2] ITU-T는 차세대 네트워크 기술을 서술한 문서인 Y.2060 “Overview of the Internet of Things”에서 사물 인터넷은 이미 존재하거나 향후 등장할 상호 운용 가능한 정보기술과 통신기술을 활용하여 다양한 상대 및 가상사물 간 상호연결을 통해, 진보된 서비스를 제공하는 Global Infrastructure라고 정의하고 있다.^[3]

사물 인터넷기반 스마트 응용은 센서기술, 상황인식기술, 칩 디바이스 기술, 무선기반의 통신기술, 경량화된 임베디드 네트워크 기술, 자율적·지능형 플랫폼 기술, 클라우드 기술과 빅 데이터 기술, 데이터 마이닝 기술, 사용자 중심의 웹 서비스 기술 그리고 보안 및 프라이버시 보호 기술 등 다양한 융합 기술들이 요구된다.^[4] 사물 인터넷의 주요 개념은 주변 사물이나 장치들을 네트워크로 연결하여 커넥티드 디바이스를 구성하고 여기에 다양한 센서들과 클라우드 기술을 이용하여 실시간 상황정보를 수집 저장하여 스스로 판단하고 결정할 수 있는 지능을 부여하여 서비스화 함으로써 새로운 가치를 창출하고자 하는데 있다.^[1, 2]

본 연구에서는 사물인터넷을 기반으로 하는 모바일 컴퓨팅 환경에서 센서와 클라우드 등 첨단 ICT 기술을 이용하여 특정 상황정보의 실시간 탐지와 인식을 바탕으로 현재 상황에 맞는 정확한 판단과 결정을 내릴 수 있는 지능을 기반으로 실시간 제어와 통제 및 처리를 할 수 있는 스마트 응용을 설계하고 구현하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 IoT 기반 스마트 응용을 위한 시스템 모델을 소개하고, 3장에서는 스마트 응용의 모듈별 알고리즘을 설계하였으며, 4장에서는 구현 결과를 고찰하였고, 마지막으로 5장에서 향후 연구내용과 함께 결론을 맺는다.

II. 시스템 모델

본 연구에서 고려한 사물 인터넷기반 스마트 응용은 자이로 센서와 중력 가속도 센서를 이용한 주행하고 있는 이동체의 평형유지와 자세 제어를 실시간으로 하는 것으로 조선·자동차·항공 등 기계공학(메카트로닉스) 분야에서 주로 연구되고 응용되는 분야라 할 수 있다. 일정한 방향과 속도로 진행되는 이동체의 수평 균형을 유

지하고 외부 환경요인에 의해 변형이 발생했을 때 이를 실시간으로 탐지하고 다시 수평 균형을 유지하기 위해 복원과정을 거쳐 자세 제어를 하는 스마트 응용이다.

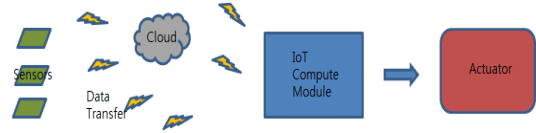


그림 1. 시스템 모델
Fig. 1. System Model.

그림 1은 본 연구에서 설계한 자세 제어를 위한 스마트 응용의 시스템 모델을 보여준다. 그림에서 보인바와 같이 이동체의 정확한 자세 제어를 위해 각속도계 자이로(Rate Gyro)센서와 3축 중력가속도(Accelerometer) 센서 등을 사용하여 이동체의 대용량의 자세 정보를 실시간으로 클라우드에서 모니터링하고, IoT 컴퓨팅 모듈에서 이 정보를 기초하여 현재 이동체의 자세 상황을 실시간으로 파악해 불균형이 포착되면 그 정도를 계산하여 평형 복원을 위해 Actuator를 구동하게 된다. 여기서 IoT 컴퓨팅 모듈로는 Arduino나 인텔 Edison 보드 또는 ARM Cortex M4 보드나 Raspberry Pi2 보드 등을 고려할 수 있고, Actuator는 이동체 종류에 따라 상이하게 적용될 수 있으며 전자 변속기(ECS)가 장착된 BLDC 모터 등이 주로 사용된다.

III. IoT기반 응용 설계

본 연구에서 설계한 IoT기반 응용은 입력 센서모듈, 상황정보 분석 처리모듈 그리고 actuator 제어모듈 등 세 부분으로 구성된다. 입력 센서모듈은 3축 중력 가속도계, 각속도계, 자자기계, 기압고도계 등을 사용하며, 이들 센서들이 수집하는 이동체의 실시간 자세 상황정보를 바탕으로 현재 이동체의 자세를 판단하는 근거가 된다. 주행하는 이동체의 정확한 자세 상황정보를 추출하기 위해서는 주로 중력 가속도 센서와 자이로 센서를 이용, 그 측정값을 조합하여 정확한 자세 상황정보를 결정하게 된다. 여기서 가속도 센서는 중력 가속도를 이용하여 이동체가 어느 정도 기울어졌는지에 대한 가로, 세로, 높이 등 3축의 각도를 측정하도록 되어 있다. 그 원리는 연직 방향으로 작용하는 중력을 이용하여 그와 기울어진 정도를 측

정하는 원리로 처음 가속도 센서에 전원이 인가된 시점의 위치를 기준점으로 삼아 기울어진 정도를 계산하도록 되어 있어 정확한 기울기 각도계산을 위해서는 전원 인가 전 센서의 정확한 수평 균형 보정이 매우 중요하다. 자이로 센서는 각속도를 측정하는 센서로 가로, 세로, 높이 축에 대해 회전하는 속도를 측정하여 각속도를 측정하는데 이때 측정되는 각속도를 적분하면 회전각을 얻을 수 있다. 가속도 센서와 자이로 센서를 조합하여 자세 상황정보의 가로축 X를 중심으로 회전각인 Roll, 세로축 Y를 중심으로 회전각인 Pitch, 그리고 높이축 Z를 중심으로 회전각 Yaw의 각도를 이용하여 정확한 이동체의 자세 상황정보를 구하게 된다.

센서로부터 실시간 센싱되는 자료들은 그 종류와 양이 매우 방대하여 효율적인 실시간 처리를 위해 클라우드 환경을 이용하게 된다. 특히 사물 인터넷기반 스마트 응용의 경우 서버, 스토리지, 네트워크 장비, 소프트웨어, 애플리케이션 등 주요 IT자원의 사용을 소유가 아닌 임대로 전환하는 패러다임의 변화가 본격화되고 있어 인터넷을 통해 가상의 클라우드 공간에서 원하는 서비스를 제공하는 클라우드 컴퓨팅 기술이 매우 유용하다.

평형을 유지하는 이동체에 외부요인에 의해 불균형이 초래되면 입력 센서모듈의 가속도와 자이로 센서에 의해 불균형 정도를 연직방향으로부터 벗어난 가로, 세로, 높이의 기울기 각각 기준 각도로부터 회전한 가로, 세로, 높이 축을 중심으로 회전각 등 총 6가지의 값으로 측정되며, 처리모듈에 의해 그 불균형 정도를 계산하여 actuator의 자세 제어모듈에서 불균형을 복원하기 위한 작업이 이루어진다. 이를 효율적으로 수행하기 위해 메카트로닉스 분야에서 사용되는 대표적인 제어방식으로 PID 제어가 있다.^[5] PID 제어기는 비례, 적분, 미분항 제어를 말하며 제어 대상의 측정된 출력값과 설정값 간의 오차를 이용하여 적절한 제어에 필요한 output을 계산하게 된다.

$$C_i(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

수식 (1)은 표준 PID 제어기 수식을 보여주며 비례항, 적분항 그리고 미분항 등 3개의 항을 더하여 제어값을 계산하도록 구성되어 있다.^[6] 여기서 K 항은 상수이고 $e(t)$ 항은 각 항에 대한 오차를 나타낸다. 보다 안정적인 자세 제어를 위해 경험적 방법을 통해 gain값을 계산하는 Ziegler-Nicholas 튜닝방법을 적용하기도 한다.^[5]

입력 센서모듈에서는 3축 중력 가속도계와 각속도계 센서를 이용하여 주행하는 이동체의 자세 상황정보를 수집하고, 처리모듈에서 측정된 이동체의 자세정보를 바탕으로 실시간 균형 이탈정도를 계산하여 actuator 모듈에 자세제어를 위한 균형 복원작업을 수행하도록 한다. 이 과정은 실시간으로 이루어지며 외부요인으로 인한 이동체의 일시적 불균형을 즉시 해소할 수 있다.

```

Input Sensor Module Operation Algorithm
#include <Wire.h>
#include <Servo.h> // import Header files
-중략-
void readSensors() { // 실시간 센서값 읽기
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr, 14, true);
  AcX = Wire.read() << 8 | Wire.read(); // 가속도 센서 X
  AcY = Wire.read() << 8 | Wire.read(); // 가속도 센서 Y
  AcZ = Wire.read() << 8 | Wire.read(); // 가속도 센서 Z
  Trmp = Wire.read() << 8 | Wire.read(); // 온도 센서
  GyX = Wire.read() << 8 | Wire.read(); // 자이로 센서 X
  GyY = Wire.read() << 8 | Wire.read(); // 자이로 센서 Y
  GyZ = Wire.read() << 8 | Wire.read(); // 자이로 센서 Z
}
-중략-
void CalcSensor() {
float s_AcX=0, s_AcY=0, s_AcZ=0; // 미세 떨림 오차 측정값 합 변수 초기화
float s_GyX00, s_GyY=0, s_GyZ=0; // 미세 떨림 오차 측정값 합 변수 초기화
  readSensors(); // 가속도와 자이로 센서 측정값 읽기
  for(int i=0; i<10; i++) { // 10번 반복해서 측정값을 읽어 평균값 구하기
    readSensors(); // 센서 측정값 읽기
    s_AcX += AcX;
    s_AcY += AcY;
  }
-중략-
  baseAcX = s_AcX/10; // 기본 센서 값과 평균값 구하기
  baseAcY = s_AcY/10;
-중략-
}
    
```

그림 2. 입력 센서모듈 구동 알고리즘
 Fig. 2. Input Sensor Module Operation Algorithm.

그림 2는 입력 센서모듈의 구동 알고리즘 일부를 보여 준다. 그림에서 보인바와 같이 readSensors() 함수를 호출하여 총 10번 반복하여 가속도 센서와 자이로 센서의 실시간 측정값을 읽어 그 평균값을 계산하여 사용하는데 그 이유는 센서의 미세한 떨림으로 인한 오차를 보정하기 위함이다.

```

Sensor Process Module Algorithm
void calcAcc() { // 중력 가속도 센서 처리 루틴
float accX, accY, accZ; // X, Y, Z 축에 대한 보정각도 저장 변수선언
float accXZ, accYZ;
#define RAD2DEG 57.32 // Radian to degree 값 180/3.14
  accX = AcX - baseAcX;
  accY = AcY - baseAcY;
  accZ = AcZ + (16384 - baseAcZ); // 이상적 중력 가속도 값 16384
  accYZ = sqrt(pow(accY,2) + pow(accZ,2));
  acc_Y = atan(-accX/accYZ)+RAD2DEG; // roll 값 계산
  accXZ = sqrt(pow(accX,2) + pow(accZ,2));
  acc_X = atan(accY/accXZ)+RAD2DEG; // pitch 값 계산
  acc_Z = 0;
-중략-
void calcGyro() { // 자이로 센서 처리 루틴
#define AVEL 131 // 각속도 저장 매크로 변수
  gyroX = (GyX - baseGyX) / AVEL; // 보정 roll 계산
  gyroY = (GyY - baseGyY) / AVEL; // 보정 pitch 계산
  gyroZ = (GyZ - baseGyZ) / AVEL; // 보정 yaw 계산
}
-중략-
}
    
```

그림 3. 센서 처리모듈 알고리즘
 Fig. 3. Sensor Process Module Algorithm.

그림 3은 가속도 센서와 자이로 센서로부터 측정된 값을 처리모듈에서 균형 이탈각도를 구하는 알고리즘 일부를 보여준다. 여기서 calcAcc()함수는 가속도 센서로 이동체의 기울어진 각도를 계산하고, calcGyro()함수는 자이로 센서를 이용하여 Roll, Pitch, Yaw를 구한다.

```

Actuator Control Module Algorithm
void calcMotorSpeed() { // 현재 모터의 스피드 계산
int throttlePower = 819 + throttle 875;
if(throttlePower > MaxSIG;
throttIrPower = MaxSIG;
if(throttlePower < MinSIG;
throttIrPower = 900;
motorA_speed = throttlePower;
}
void updateMotorSpeed() { // 설정된 모터 스피드의 갱신
A.writeMicroseconds(motorA_speed);
}
int p = pthread_create(&Recv_Thread, NULL, RECVFunction,(void*)0); //pthread
pthread_detach(Recv_Thread);
void loop() { // POSIX pthread를 이용한 실시간 병행 처리
calcMotorSpeed();
updateMotorSpeed();
}
-종략-
    
```

그림 4. Actuator 제어모듈 알고리즘
Fig. 4. Actuator Control Module Algorithm.

그림 4는 이동체의 불균형을 복원하기 위해 모터를 구동시키는 제어모듈 알고리즘의 일부를 보여준다. 그림에 보인바와 같이 현재 구동중인 모터의 속도를 계산하는 함수와 설정된 모터 속도로 갱신하는 함수의 실행이 반복문 내에서 POSIX Thread를 이용하여 실행된다. 이렇게 함으로써 동시에 병행처리가 가능하여 실시간 불균형 해소를 위한 복원 처리가 즉시 이루어지게 된다.

IV. 구현 결과 및 고찰

이 장에서는 본 논문에서 제안한 IoT기반 스마트 응용의 성능실험을 위한 시스템 구현과 구동 결과를 소개한다. 그림 5는 IoT기반 응용을 위한 시스템의 주요 컴포넌트 구성도를 보여준다.

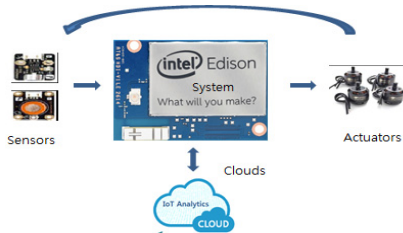


그림 5. 시스템 컴포넌트 구성도
Fig. 5. System Component Diagram

표 1. 시스템 환경

Table 1. System Environment

Component	Description
Processor	Intel Edison board with dual-core ATOM CPU at 500 MHz & 32 bit
Sensors	MPU6050 (Accelerometer & Gyro Sensor)
Actuator	Motor (Brushless DC Motor) MT2206-2300KV
Clouds	CleanFlight & Ubidots IoT Cloud Services

그림에서 보인바와 같이 주요 컴포넌트는 시스템 처리부, 센서부, 구동 제어부 그리고 클라우드부로 구성된다. 표1에서는 각 컴포넌트에서 사용된 주요 시스템 환경을 보여준다. 표에 보인바와 같이 시스템 처리기는 인텔 에디슨 보드를 사용하였고, 센서는 MPU6050, 구동장치는 BLDC 모터, 그리고 실행 결과 확인을 위해 CleanFlight와 Ubidots 클라우드 서비스를 사용하였다.

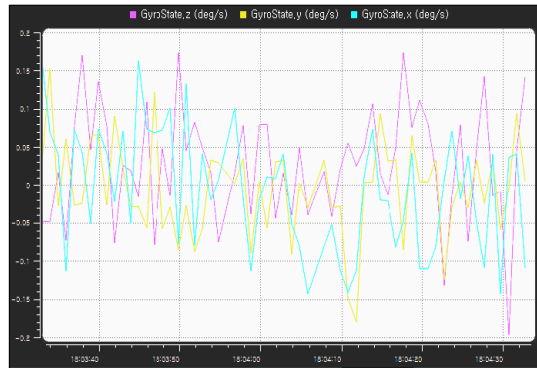


그림 6. 자이로 센서의 계측기 측정
Fig. 6. Scope Measurement of the Gyro Sensor.

그림 6은 구현한 시스템의 자이로 센서로부터 계측기를 이용하여 실시간 측정되는 상황정보를 보여준다. 그림 7은 구현한 이동체가 이동하면서 불균형을 초래했을 때 이 상황을 실시간으로 인지하고 균형으로의 자세복원을 위해 Actuator인 모터를 구동시키고 있는 상황을 스톱샷한 실행결과를 보여준다. 그림에 보인바와 같이 좌측으로의 불균형의 정도가 심할수록 이를 복원하기 위해 반대쪽 모터의 회전수가 급격히 높아지고 있는 것을 알 수 있다. 그림 8은 클라우드 IoT 서비스인 CleanFlight로 이를 통해 실시간 이동체의 자세 상황을 모니터링하고, 불균형 상태에 대한 복원을 위한 각 모터들의 구동 상황을 실시간 확인할 수 있다.

```

Com4 -Putty
roll_output : -3.708917 pitch_output : 27.084194 yaw_output : 6.310498
motor1_speed:982, motor2_speed:1024, motor3_speed:1029, motor4_speed:962

roll_output : -4.234787 pitch_output : 27.847200 yaw_output : 3.881990
motor1_speed:980, motor2_speed:1028, motor3_speed:1027, motor4_speed:964

roll_output : -7.213301 pitch_output : -4.990145 yaw_output : -20.150478
motor1_speed:992, motor2_speed:1022, motor3_speed:967, motor4_speed:1017

roll_output : 1.497988 pitch_output : 2.633337 yaw_output : -80.878883
motor1_speed:914, motor2_speed:1082, motor3_speed:923, motor4_speed:1079

roll_output : 1.521367 pitch_output : -3.968770 yaw_output : 65.274879
motor1_speed:1067, motor2_speed:929, motor3_speed:1062, motor4_speed:940

roll_output : 0.594061 pitch_output : -0.732072 yaw_output : 69.120720
motor1_speed:1069, motor2_speed:929, motor3_speed:1068, motor4_speed:932

roll_output : -0.365723 pitch_output : -0.145851 yaw_output : -8.321955
motor1_speed:992, motor2_speed:1008, motor3_speed:991, motor4_speed:1008
    
```

그림 7. 실행 결과
 Fig. 7. Execution Results.



그림 8. 클라우드를 통한 실행결과
 Fig. 8. Execution Result through Clouds.

V. 결론

무선 인터넷 기술의 급속한 발전과 더불어, 주변의 모든 사물들이 인터넷으로 연결되는 초 연결 사회로 급속히 발전하고 있으며, 이를 이용한 사물인터넷기반 스마트 응용의 개발이 활발하게 이루어지고 있다. 최근 사물인터넷은 다양한 센서들과 클라우드 기술을 이용, 실시간 상황정보를 수집 가공하여 스스로 판단하고 결정할 수 있는 지능을 부여한 서비스화된 스마트 응용 개발이 주류를 이루고 있다. 본 논문에서는 사물인터넷기반 센서와 클라우드 등 첨단 ICT 기술을 접목하여 특정 상황정보의 실시간 탐지와 인식을 바탕으로 현재 상황에 맞는 정확한 판단과 결정을 내릴 수 있는 지능기반 실시간 제어와 통제 및 처리를 할 수 있는 스마트 응용을 설계하

고 제안한 알고리즘의 성능 확인을 위해 프로토타입 시스템을 구현하여 IoT 기반 클라우드 서비스를 통해 확인하였다. 향후 연구과제로는 본 논문에서 제안한 스마트 응용을 이용한 구현과 실세계 활용에 관한 것이다.

References

- [1] H. Y. Kim, *Internet of Things : Concept, Implementation, and Business*, HongReung Science Pub. Co., 2014.
- [2] M. Koster, *Data Models for the Internet of Things*, 2012.
- [3] P. Pahlevani, M. Hundeboll, M. Pedersen, D. Lucani, H. Charf, F. Fitzek, H. Bagheri, M. Katz, "Novel Concepts for Device-to-Device Communication Using Network Coding", *IEEE Communications Magazine*, pp. 32 - 39, 2014.
- [4] S. J. Oh, "Design of a Smart Application for Remote Diagnosis in Ubiquitous Computing Environment", *The Journal of the Institute of Internet, Broadcasting and Communication (IIBC)*, vol. 16, no. 4, pp. 81 - 87, August 2016.
- [5] Y. Kim, J. Shin, S. Lee, H. Lee, H. Lim, K. Kim, S. Lee, "Quadrotor Attitude Stabilization by Using PID Controller", *Journal of Aerospace System Engineering*, vol. 4, no. 4, pp. 18 - 27, 2010.
- [6] Y. W. Chang, S. T. Lim, T. H. Kim, K. H. Lee, S. J. Baek, K. T. Chong, "Design and Simulation Quadrotor Controller for Stable Positioning Control", *Proceedings of the Conference on Information and Control System*, pp. 205 - 206, 2011.

저자 소개

오 선 진(중신회원)



- 제 6권 제2호 참조
- 현재 세명대학교 정보통신학부 교수
 <주관심분야 : 스마트 응용, IoT, 빅데이터, 모바일컴퓨팅, USN 등>

※ 이 논문은 2016학년도 세명대학교 교내학술연구비 지원에 의해 수행된 연구임.