

대한기계학회 주최

제10회 전국학생설계경진대회(2020년)

설계 최종 보고서

참가부	고등부 () / 대학부 ()				
참가분야	공모주제 () / 자유주제 ()				
참가팀명	아보카도				
설계제목	경사로 및 계단 보행 안전성을 고려한 보행보조기와 IoT 기반 연계 어플리케이션				
지도교수/교사	(소속) 인천과학예술영재학교 (성명) 김시영 (이메일) ksy235@ice.go.kr				
대표자 (신청인)	성명	소속	연락처 (휴대폰)	E-mail	주소
	왕수진	인천과학예술영재학교			

참가팀원 인적사항

NO	성명	소속 / 학년	E-MAIL
1	이재은	인천과학예술영재학교 / 2학년	inthemode@naver.com
2	왕수진	인천과학예술영재학교 / 2학년	wsj0317@naver.com
3			
4			
5			
6			

설계 요약문

참가분야	공모주제 (○) / 자유주제 ()
참가팀명	아보카도
설계제목	경사로 및 계단 보행 안전성을 고려한 보행보조기와 IoT 기반 연계 어플리케이션
대표자명	왕수진
요약문	고령화 시대로 들어서면서 실버복지를 위한 시장도 빠르게 확대되고 있다. 그중에서도 노년층의 이동을 돕는 보행기는 전 노인층에 걸쳐서 사용되고 있다. 실제로 보행 중 교통사고의 절반이 노년층인 만큼 실버복지에서 보행이 중요한 요소로 작용한다. 그러나 시중의 보행기는 계단을 오르내릴 수 없는 점, 오르막, 내리막길에서 속도 조절을 하기 힘들다는 등의 단점이 존재한다. 따라서 본 팀은 쉽게 계단을 오르내릴 수 있고, 다리의 길이 조절을 통해서 경사로에서 이동이 쉽고, 가격과 보행보조기를 경량화시켜야 한다는 점을 고려하였다. 또한, IOT 기반 시스템 구축을 통해 추가적인 편의, 안전 기능을 갖춘 보행기를 개발하고자 한다. 1) 바퀴 부 : 계단이나 턱을 오를 때 편리성과 안전성을 높이기 위해서 보행기에 앞바퀴 양쪽에 각각 바퀴를 4개씩 달아서 제작한다. 바퀴 4개는 계단의 표준 높이에 맞춰 원형으로 나란히 배치하여서 보행기를 몸으로 밀면 자연스럽게 계단이나 턱을 오를 수 있는 구조이다. 2) 길이 조절부 : 경사진 길을 오르내릴 때의 위험성을 줄이기 위해서 보행기 앞의 두 다리의 길이를 조절 가능한 형태로 설계하였다. 보행기에 자이로 센서를 부착하여서 사용자가 서 있는 지면의 기울어진 정도를 측정하여서 그에 따라 보행기 다리 길이가 조절되어서 보행기 자체는 수평한 상태를 유지할 수 있도록 한다. 3) 종합 실버 복지 어플리케이션 : 실버 복지 복합 어플리케이션을 개발하고, 보행보조기에 부착한 센서의 정보를 어플리케이션과 연계하는 IoT 시스템을 구축한다. gps센서로 위치를 추적하여 노인분들의 사고의 위험성을 줄여주고 안전을 지켜주는 역할을 한다. 본 장치를 노년층에 보급하여 고연령 치매 환자들의 실종률을 줄이고, 더 안전하게 하여 보행 중 사고율을 낮출 수 있을 것이다. 또한, 개발한 어플리케이션은 실버 복지에 도움이되는 기능들이 포함되어 있어 노인분들을 보조하는 기능을 할 것이다. 더 나아가서, 본 장치를 더 응용시켜서 노년층에서 그치지 않고 다양한 사이즈로 제작하거나 치수를 조절할 수 있도록 하여 보행에 어려움이 있는 누구나 사용할 수 있을 것으로 기대된다.

1. 설계의 필요성 및 목적

고령화 사회로 들어서면서 실버복지는 새로운 사회문제로 부상하였다. 노년층 대부분이 보조기 없이 스스로 이동하는 데에 어려움을 겪고 있고, 이 때 가장 흔하게 사용하는 것이 보행보조기이다. 본 팀이 설계한 장치의 궁극적인 목적은 노인분들이 보행보조기를 이용하여서 혼자 이동할 때 더 안전하고 편리하도록 하는 것이다. 특히 경사면에서 보행기를 이용할 때 위험할 수 있다는 점과 계단이나 턱을 오를 수 없다는 점에 주목하여 이를 해결할 수 있는 설계를 한다.

또한, 가격을 낮추는 것과 사용을 편하게 하도록 경량화하는 것에 중점을 뒀다. 부가적으로 보행보조기와 연동된 IoT 종합 실버 복지 어플리케이션을 개발하여서 노인분들을 다양한 측면에서 보조할 수 있는 기능들도 추가하려 한다.

2. 설계 핵심 내용

(1) 설계 문제의 정의

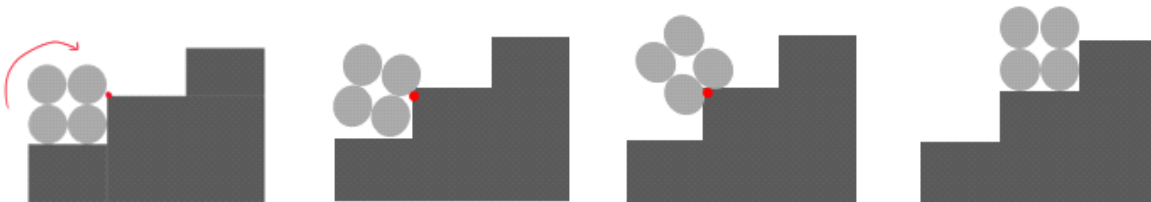
본 설계 팀이 생각한 보행보조기의 가장 중요한 목적은 노인분들이 이동할 때 더 안전하고 편리하도록 하는 것이다. 기존의 보행보조기는 계단을 오르내릴 수 없는 점, 오르막길, 내리막길에서 속도 조절을 하기 힘들다는 점 등 여러 가지 한계점이 존재한다. 따라서, 본 팀은 특히 계단과 경사로를 오르내리는 경우에서의 안전성과 편리성에 중점을 맞추어서 설계하였다. 또한, 가격과 보행보조기를 경량화시켜야 한다는 점을 고려하였다. 따라서, IoT 기반 시스템 구축을 통해 추가적인 편의를 갖춘 보행보조기이다.

(2) 설계의 독창성 및 접근 방법

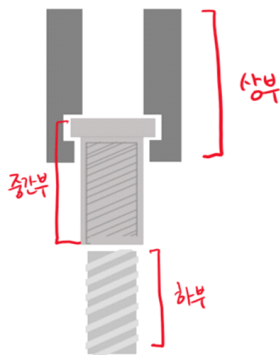
1) 설계 방법 및 배경

(1) 바퀴 부

계단이나 턱을 오를 때 편리성과 안전성을 높이기 위해서 보행기에 앞바퀴 양쪽에 각각 바퀴를 4개씩 달아서 제작한다. 바퀴 4개는 계단의 표준 높이에 맞춰 원형으로 나란히 배치하여서 보행기를 몸으로 밀면 자연스럽게 계단이나 턱을 오를 수 있는 구조이다. 3D 모델링을 통해서 축에 바퀴가 4개 붙어있는 형태를 구현하고, 직접 계단의 높이를 측정하여서 적절한 바퀴 높이를 계산해본다. 즉, 부착된 바퀴 4개는 계단의 표준 높이에 맞춰 원형으로 나란히 배치하여서 보행기를 몸으로 밀면 자연스럽게 계단이나 턱을 오를 수 있는 구조이다.



바퀴가 계단을 오르는 모습



보행보조기 다리 길이
조절부 형상화

(2) 길이 조절부

기존의 보행 보조기는 오르막, 내리막길에서 사용하기에 매우 위험하고 불편한 구조이다. 그래서 보행 보조기의 네 기둥 중 앞의 두 기둥의 길이를 조절함으로써 내리막, 오르막을 더 안정적으로 오르내릴 수 있도록 한다. 보행 보조기에 자이로 센서를 부착하여 높이 변화를 측정하고, 초깃값에 대한 높이 변화에 따라 그에 보행보조기의 길이가 조절된다. 길이 조절은 무게를 견디는데 더 유리하도록 나선 구조로 하여 접촉면을 높여 힘을 분산해 사람의 체중 정도의 큰 압력도 버틸 수 있도록 한다.

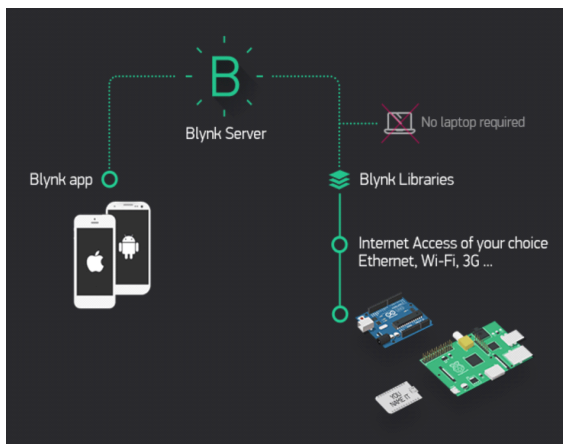
다리는 하부, 중간부, 상부로 구성되어 있다. 하부에는 나사의 양각, 중간부에는 나사의 음각 부분이 있어서 중간부가 돌아가면서 보행보조기의 길이를 조절한다. 상부와 하부는 회전하지 않고, 중간부가 회전한다. 따라서 모터로 인해 중간부에 토크를 준다. 상부와 중간부는 연결되어있으나 상부에 홈으로 연결하여 회전은 가능하다. 이때 보행보조기의 앞의 있는 두 다리는 바퀴를 달지 않고 마찰이 비교적

강한 고무 등의 재질로 제작하여 경사진 길에서 쉽게 미끄러지지 않도록 보조한다.

실제 보행기를 절단하기 시간적, 공간적 제약이 있어 SOLIDWORKS를 사용하여 3D로 설계하고 시뮬레이션하는 것으로 대체하였다.

기울어지는 정도를 알기 위해서는 자이로센서와 서브 모터를 제어하기 위해서 아두이노를 활용한다. 자이로센서 중에서도 6축 위치를 감지할 수 있는 모형을 사용하여, 급정지 등의 상황에도 대비할 수 있도록 한다.

(3) 종합 실버 복지 어플리케이션



- 종합 실버 복지 어플리케이션은 노인분들이 이동할 때 발생할 수 있는 각종 사고의 위험을 사전에 방지하기 위한 목적이 있다. 이를 위해서 보행보조기에 부착하면 앱을 통해서 위치를 수신받을 수 있도록 아두이노를 통해서 제작하였다. GPS 통신을 위해서 아두이노의 GPS 수신 모듈인 NEO-6M을 이용하였는데 이는 시리얼 통신으로 GPS 신호를 읽어 들일 수 있는 GPS 모듈이다. GPS 수신 모듈의 동작 원리는 해당 모듈이 GPS 위성으로부터 수신받은 신호를 처리한 후 아두이노를 통해 신호 값을 확인할 수 있는 것이다. 확인된 신호를 지도로 위치를 확인하거나, 값을 분석해 현재 시각, 이동 거리, 속도 등을 다양한 데이터를 얻을 수 있다.

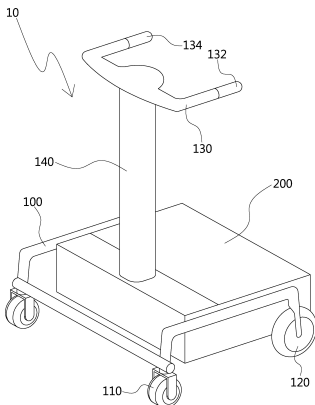
이때 받아온 값들은 아두이노의 시리얼 모니터를 통해서 표시되는 위도와 경도 값을 구글 지도에 넣어서 확인하는 것도 가능하지만, 이 방식은 실질적으로 보행보조기의 위치를 실시간으로 확인할 방법으로 적합하지 않다. 따라서, 아두이노에서 자체적으로 IoT 어플리케이션을 만들 수 있도록 지원하는 앱인 'Blynk'을 이용해서 제작하였다. 'Blynk' 어플리케이션은 원하는 프로젝트에 대해서 다양한 위젯을 이용해서 여러 인터페이스를 꾸미는 것이 가능하게 되어 있고, 'Blynk' 서버는 휴대전화와 하드웨어 사이에서 통신하는 역할을 한다.

2) 설계의 독창성

보행 중 교통사고 사망의 56%가 노인일 정도로(중앙일보, 강갑생 기자) 노인들은 다른 연령대에 비해 보행

시간이 길고, 그에 따른 위험 상황에 노출되어있다. 현재 보행보조기는 노인들에게 중요한 물품으로 움직일 때 지지를 해주는 역할은 하지만, 그 이외에 안전성에 대한 충분한 고려가 들어가지 않았다. 특히 우리는 계단과 경사로를 오르내리는 경우에서의 안전성과 편리성에 중점을 맞춰서 설계하였다. 이 장치를 통해 이동하는데 어려움을 갖는 노인들의 불편함을 해소하고 안정성을 높일 수 있다.

또한, 현재 대부분의 실버 복지 보조기구의 가격이 높은 가격이 책정되어 있다. 대부분 실질적으로 보조기구가 필요한 사람은 저소득층이기 때문에 아무리 좋은 기능이 있더라도 가격이 높다면 무용지물이다. 따라서 실제로 개발되어 상용화되었을 때를 충분히 고려하여 가성비를 좋게 하는 것은 중요하기 때문에 본 설계가 경제성이 있다고 볼 수 있다.

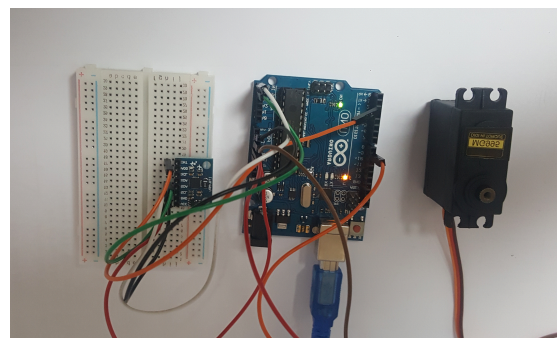
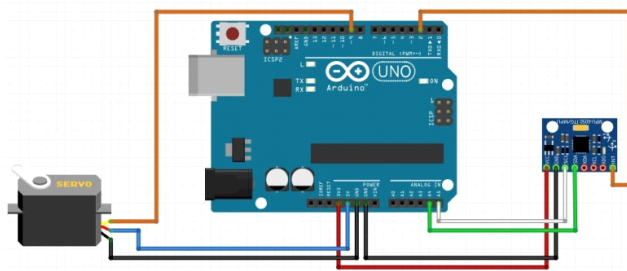


더 나아가서, 'Kipris'에서 경사로나 계단에서의 안전성을 높여주는 보행보조기를 검색한 결과, 보행보조기(1020190010865)를 찾을 수 있었다. 해당 발명품은 경사진 비탈길에서 자동으로 저속 모터에 의해 바퀴가 회전하도록 하여서 과속을 방지한다. 경사로에서의 안전성을 높여준다는 점에서 본 설계와 발명의 의도는 유사하지만, 해당 발명품은 더 기계적인 장치와 모터의 힘을 많이 빌리기 때문에 비효율적인 측면이 있다. 또한, 본 설계는 경사로에서의 안전성 이외에도 노인분들에게 큰 불편함을 가져다줄 수 있는 계단에서의 안전성도 고려하였고, 더 나아가서 IoT 기반의 어플리케이션을 이용하여서 보호자도 노인분들의 상태를 수시로 파악할 수 있도록 하였다는 점이 차별성이 크다.

3) 설계의 제약조건 및 문제 해결 방법

길이 조절부를 제작할 때 경사로의 기울기 변화에 따라서 미세한 길이 변화가 가능하도록 해야 하므로 단순히 힘으로 누르거나 버튼 제어 방식 등으로는 어려움이 있을 수 있다. 또한, 어르신분들은 힘을 내기 어렵다는 점을 고려하여서 사람의 힘이 있어야 하는 제어 방식은 문제점이 많다. 따라서, 기울기 변화에 따라서 모터가 회전해야 하는 정도를 계산 값으로 구해서 모터를 미세하게 조절하는 방식으로 설계하였다. 또한, 보행보조기의 위치를 추적하는 것도 중요하지만, 이를 어플리케이션으로 연동할 수 있게 하여서 보호자가 실질적으로 정보를 받아야 하므로 IoT 기반의 어플리케이션을 제작하였다.

(3) 설계 내용



가장 먼저 아두이노 회로를 어떻게 구상할 것인지 AUTODESK THINKERPAD를 사용해서 미리 설계하고, 시뮬레이션한 뒤, 실제로 구현하였다.

아두이노를 사용해서 자이로센서를 사용해서 모터-를 움직일 수 있게 하도록 자이로센서(MPU 6050), 서브모터(MG995 Servo)를 사용하였다. 자이로센서의 높이에 따라서 모터가 움직이는 정도를 다르게 하여 보행자의 관점에서 높이 변화의 폭이 작도록 조정하였다. 코드는 다음과 같다.

```
#define INTERRUPT_PIN 2 // use pin 2 on Arduino Uno & most boards

// MPU control/status vars
bool dmpReady = false;
uint8_t mpuIntStatus;
uint8_t devStatus;
uint16_t packetSize;
uint16_t fifoCount;
uint8_t fifoBuffer[64];

Quaternion q; // [w, x, y, z]      quaternion container
VectorFloat gravity; // [x, y, z]  gravity vector
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

|
volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has gone high
void dmpDataReady() {
    mpuInterrupt = true;
}
```

가장 먼저 핀을 지정한 뒤 quaternion container, gravity vector 변수, 각각 yaw/pitch/roll container and gravity vector 값을 담는 배열을 선언한다.

```
void setup() {
    myservo.attach(servoPin);

    Wire.begin();
    Wire.setClock(400000); // 400kHz I2C clock. Comment this line if having compilation difficulties

    Serial.begin(115200);
    Serial.println("Initializing I2C devices...");
    mpu.initialize();
    pinMode(INTERRUPT_PIN, INPUT);
    // verify connection
    Serial.println("Testing device connections...");
    Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));

    // load and configure the DMP
    Serial.println("Initializing DMP...");
    devStatus = mpu.dmpInitialize();

    // supply your own gyro offsets here, scaled for min sensitivity
    mpu.setXGyroOffset(220);
    mpu.setYGyroOffset(76);
    mpu.setZGyroOffset(-85);
    mpu.setZAccelOffset(1788); // 1688 factory default for my test chip

    // make sure it worked (returns 0 if so)
    if (devStatus == 0) {
        // turn on the DMP, now that it's ready
        Serial.println("Enabling DMP...");
        mpu.setDMPEnabled(true);

        // enable Arduino interrupt detection
        Serial.println("Enabling interrupt detection (Arduino external interrupt 0)...");
        attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), dmpDataReady, RISING);
        mpuIntStatus = mpu.getIntStatus();

        // set our DMP Ready flag so the main loop() function knows it's okay to use it
        Serial.println("DMP ready! Waiting for first interrupt...");
        dmpReady = true;

        // get expected DMP packet size for later comparison
        packetSize = mpu.dmpGetFIFOPacketSize();
    } else {
        // ERROR!
        // 1 = initial memory load failed
        // 2 = DMP configuration updates failed
        // (if it's going to break, usually the code will be 1)
        Serial.print(F("DMP Initialization failed (code ") + devStatus);
        Serial.print(")");
        Serial.println(F(""));
    }
}
```

setup 부에서는 시리얼 모니터로 자이로센서가 잘 작동하는지 확인할 수 있도록 작성했다. 또한, 자이로센서의 초깃값을 설정하였다.

```
void loop() {
  // if programming failed, don't try to do anything
  if (!dmpReady) return;

  // wait for MPU interrupt or extra packet(s) available
  while (!mpuInterrupt && fifoCount < packetSize);

  // reset interrupt flag and get INT_STATUS byte
  mpuInterrupt = false;
  mpuIntStatus = mpu.getIntStatus();

  // get current FIFO count
  fifoCount = mpu.getFIFOCount();

  if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    // reset so we can continue cleanly
    mpu.resetFIFO();
    Serial.println(F("FIFO overflow!"));
  } else if (mpuIntStatus & 0x02) {

    // wait for correct available data length, should be a VERY short wait
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

    // read a packet from FIFO
    mpu.getFIFOBytes(fifoBuffer, packetSize);

    // track FIFO count here in case there is > 1 packet available
    // (this lets us immediately read more without waiting for an interrupt)
    fifoCount -= packetSize;

    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
  }
}
```

loop 부에서는 계속해서 자이로센서의 값을 받고, 계산식을 활용해서 모터를 제어하였다.

```
//소숫점 제거하여 프린트 찍어봄
int yaw = ypr[0] * 180.0/M_PI;
int pitch = ypr[1] * 180/M_PI;
int roll = ypr[2] * 180/M_PI;
Serial.print(yaw);
Serial.print("\t");
Serial.print(pitch);
Serial.print("\t");
Serial.println(roll);

/*****
/*****
//pitch 데이터가 -90 ~ 90까지 들어오기 때문에 그것을
//map 함수를 사용하여 0 ~ 180으로 변경한 것이다.
servoVal = map(pitch, -90, 90, 0, 180);
myservo.write(servoVal);
```

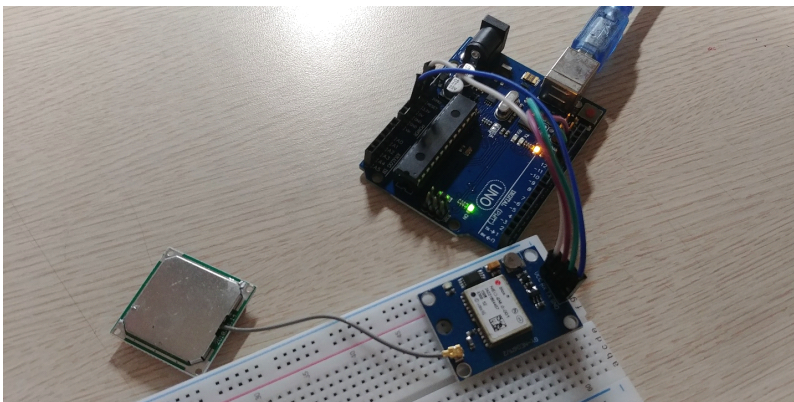
loop 코드 중 이 부분이 핵심적이다. 자이로센서의 각 값에 따라서 모터의 각도를 계산하는 코드이다. 실제 모델을 제작한 후 사용자에게 맞게 이 값을 조정하여 높이 보정 정도를 결정할 수 있도록 하였다.

(3) 종합 실버 복지 어플리케이션

- 보행보조기에 부착하면 앱을 통해서 위치를 수신받을 수 있도록 하는 시스템을 아두이노를 통해서 제작하였다. 이를 위해서 'NodeMCU ESP8266 Breakout Board'와 'NEO-6M V2 비행 제어 GPS모듈'을 이용하였고, 앱과의 연동을 위해서는 아두이노에서 IoT 어플리케이션을 제작할 수 있도록 도움을 주는 어플리케이션인 'Blynk'를 이용하였다.

가장 먼저, GPS 모듈의 기능을 테스트해보기 위해서 아두이노의 시리얼 모니터를 통해서 값을 받아온 뒤 구글 지도를 통해서 위치를 확인해 볼 방법을 시도해보았다.

아두이노 우노	GPS 수신 모듈	
5V	VCC	<pre>#include <SoftwareSerial.h> SoftwareSerial gps(11, 12); void setup() { Serial.begin(9600); gps.begin(9600); } void loop() { if(gps.available()){ Serial.write(gps.read()); } }</pre>
GND	GND	
D12	RX	
D11	TX	



위에 보이는 코드를 동작시키고 나면 시리얼 모니터를 통해서 GPS 수신 모듈의 신호를 받을 수 있다. 이때 확인되는 여러 가지 값 중에서 GPS 신호의 위치, 시간, 속도 등 다양한 정보를 담고 있는 GPGAA의 값을 이용해서 이 위도와 경도 값을 얻어내기 위해서 'Gonmad'라는 사이트에서 GPS NMEA data를 위도, 경도 값으로 바꾸어준다. 그 뒤에 위도와 경도 값을 구글 지도에 붙여넣으면 해당하는 위치를 찾을 수 있다.

더 나아가서, 해당하는 위치만 확인하는 것이 아니라 날짜, 시간, 고도 진행 방향, 이동 속도 등을 파악하기 위해서 다음과 같은 코드를 이용하여 얻어낸 데이터를 GPS 로그 파일인 gpx 파일로 변환을 한 뒤에 구글어스에 파일을 불러오면 이동한 길이 화면상에 표시되는 것을 확인할 수 있다.

```

#include <SoftwareSerial.h>
#include <TinyGPS.h>
#define RXPIN 6
#define TXPIN 5
#define GPSBAUD 9600

TinyGPS gps;

SoftwareSerial uart_gps(RXPIN, TXPIN);

void getgps(TinyGPS &gps);

void setup()
{
  Serial.begin(9600);
  uart_gps.begin(GPSBAUD);

  Serial.println("");
  Serial.println("GPS Shield QuickStart Example Sketch v12");
  Serial.println("...waiting for lock...");
  Serial.println("");
}

void loop()
{
  while(uart_gps.available())
  {
    int c = uart_gps.read();
    if(gps.encode(c))
    {
      getgps(gps);
    }
  }
}

void getgps(TinyGPS &gps)
{
  float latitude, longitude;
  gps.f_get_position(&latitude, &longitude);
  Serial.print("Lat/Long: ");
  Serial.print(latitude,5);
  Serial.print(", ");
  Serial.println(longitude,5);

  int year;
  byte month, day, hour, minute, second, hundredths;
  gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths);
  Serial.print("Date: "); Serial.print(month, DEC); Serial.print("/");
  Serial.print(day, DEC); Serial.print("/"); Serial.print(year);
  Serial.print("   Time: "); Serial.print(hour, DEC); Serial.print(":");
  Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);
  Serial.print("."); Serial.println(hundredths, DEC);

  Serial.print("Altitude (meters): "); Serial.println(gps.f_altitude());
  Serial.print("Course (degrees): "); Serial.println(gps.f_course());
  Serial.print("Speed(kmph): "); Serial.println(gps.f_speed_kmph());
  Serial.println();

  unsigned long chars;
  unsigned short sentences, failed_checksum;
  gps.stats(&chars, &sentences, &failed_checksum);
  delay(10000);
}

```

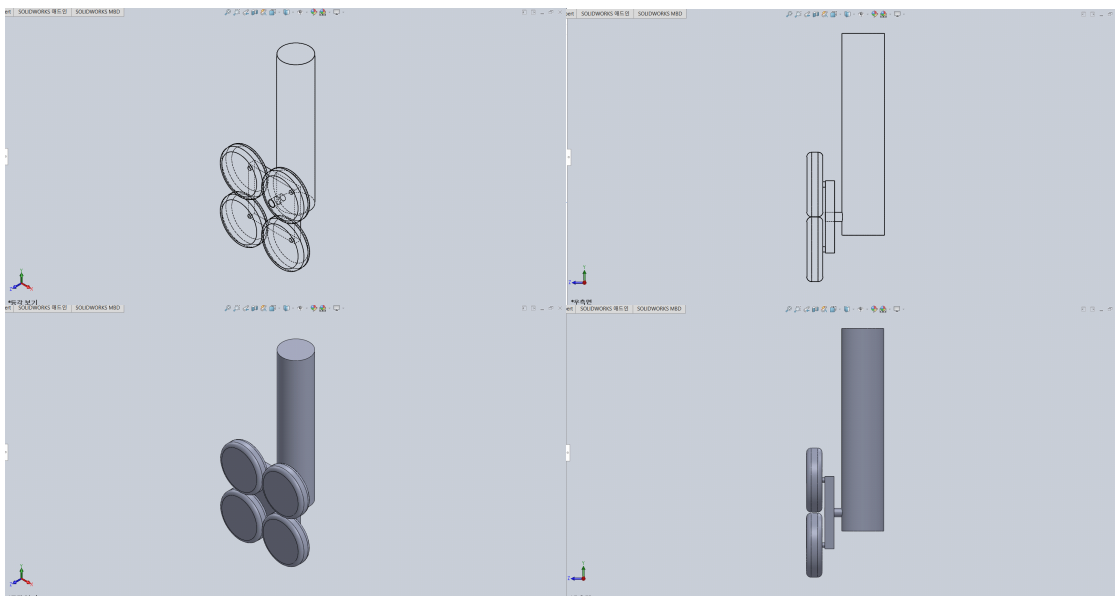
앞서 살펴본 GPS 모듈의 기능을 실제로 보행보조기에 부착시킬 아두이노에 대해서 앱과 연동을 시키기 위해 아두이노에서 자체적으로 IoT 어플리케이션을 만들 수 있도록 지원하는 앱인 'Blynk'을 이용하여서 제작하였다.

3. 설계 수행 일정

설계 진행 내용	4월	5월	6월	7월	8월	9월	
주제 선정 및 설계제안서							
설계 <u>구체화</u> 및 3D 모델링							
설계도 완성							
1차 모델 제작							
앱 개발							
최종보고서 작성							
개발 마무리 및 개선							

4. 설계 결과물

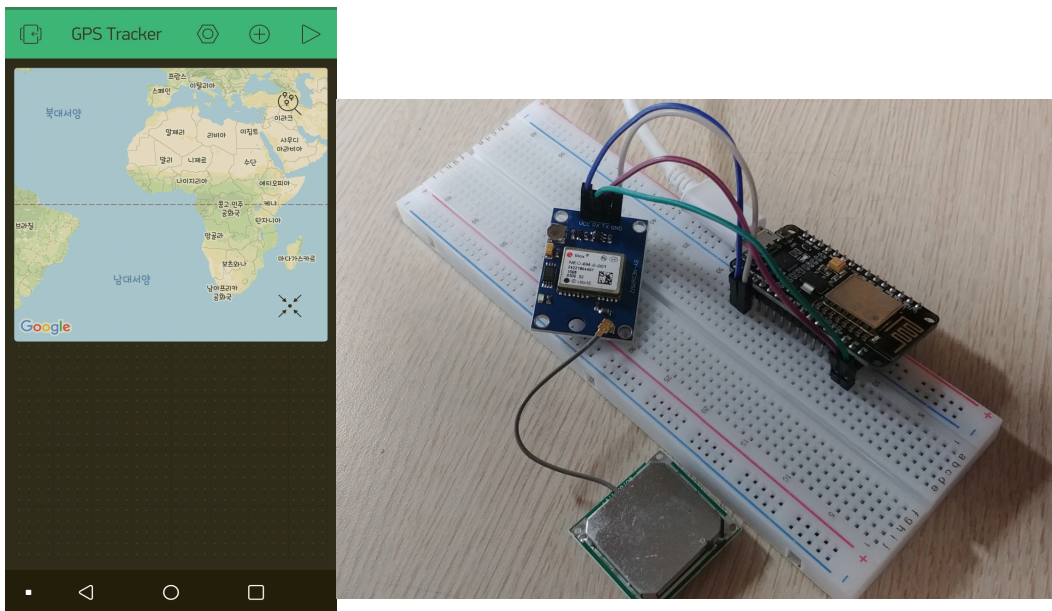
(1) 최종 결과물 형상 및 작동원리



▲ 바퀴 부 설계 모형



▲ 길이 조절부 설계 모형



▲ 어플리케이션 부 설계 모형

바퀴 모형, 길이 조절부를 3D 설계로 구현하였다. 바퀴는 4개의 바퀴 축이 판에 고정되어 있고, 그 판이 중심 축에 고정된 형태이다. 길이 조절부는 하부, 중간부, 상부로 구성되어 있다. 하부에는 나사의 양각, 중간부에는 나사의 음각 부분이 있어서 중간부가 돌아가면서 보행보조기의 길이를 조절한다. 상부와 하부는 회전하지 않고, 중간부가 회전한다.

아두이노를 사용해서 자이로센서를 사용해서 모터를 움직일 수 있게 하려고 자이로센서(MPU 6050), 서브모터(MG995 Servo)를 사용하였다. 입력 값에 따라서 자이로센서의 위치 변화 값에 따라 모터의 회전 각도와 방향을 조절할 수 있는 시스템을 구현했다. 이 시스템을 3D 모델링 한 것과 같은 보행기에 부착하면, 보행기가 덜컹거리거나 경사로에서 모터로 다리의 길이를 조절할 수 있다.

또한, 어플리케이션 부에서도 아두이노의 GPS 모듈과 wifi 보드를 이용해서 휴대전화와의 연동을 통해서 실시간으로 위치 정보를 휴대폰의 앱으로 보낼 수 있도록 하였다. GPS 모듈을 앱과 연동을 시키기 위해 아두이노에서 자체적으로 IoT 어플리케이션을 만들 수 있도록 지원하는 앱인 'Blynk'을 이용하여서 제작하였다.

(2) 최종설계 결과물의 장단점 및 의의

본 팀이 고안한 보행보조기로 경사로나 계단을 오를 때 발생하는 불편한 점과 안전에 대한 문제점들이 해소될 것이다. 센서의 가격이 저렴하므로 가격에서 기존 보행기와 차이 없이 보행에 어려움이 있는 누구에게나 보급할

수 있다. 또한, 기존에 사용하던 보행기에 감지기 시스템을 추가하면 업그레이드할 수 있다는 장점이 있다. 즉, 기존 보행보조기의 문제점들을 해결하면서 한계점들이 존재하는 핵심이 되는 부분들을 개선해 나간 설계라는 점에서 의미가 크다.

그러나 위치추적 기능은 어플리케이션을 사용하기 때문에 스마트폰을 가지고 있어야지만 사용할 수 있다는 한계가 있다.

5. 활용방안 및 기대효과

본 팀이 고안한 보행보조기를 노년층에 보급하여 고연령 치매 환자들의 실종률을 줄이고, 더 안전하게 하여 보행 중 사고율을 낮출 수 있다. 특히, 보행 중 노인 사고 비율이 줄어들 것으로 기대된다. 더 나아가서, 노년층에서 그치지 않고 다양한 크기로 제작하거나 크기를 조절할 수 있도록 하여 보행에 어려움이 있는 누구나 사용할 수 있도록 할 수 있다.

현재 개발한 것은 위치추적 기능까지이지만, 보행보조기의 센서외에도 다른 보조기구의 센서에도 연동하고, 돌보미 방문 체크 등의 실버 복지에 도움이 되는 기능을 추가하여 실제로 활용할 수 있을 정도의 복합 실버 복지 어플리케이션을 개발할 수 있다.

<참고문헌>

- 1) 최혜선, 김성민, 김은지, 이우란, 강희석. (2016). [한이음논문] 아두이노를 활용한 자전거 안전 주행 어플리케이션 ChildCycle. 한국정보과학회 학술발표논문집, (), 133-135.
- 2) 형준우, 서효동, 김태현, 이동우. (2017). 위치 기반의 버스 알림 어플리케이션. 한국정보기술학회 종합학술발표논문집, (), 251-252.
- 3) 리메이르, 정의철. (2017). 계단이 있는 환경에서 사용 가능한 보조보행기 디자인 제안. 한국자동차공학회 추계학술대회 및 전시회, (), 1438-1445.
- 4) 김현주, 이승민, 함소정, 강연호, 김기동, 김혜윤. (2018). 경사로 안전사고 방지 및 응급대처와 편리를 기반으로 한 노인을 위한 스마트 안전 보행기. 대한전기학회 학술대회 논문집, (), 173-175.