

낙상 감지 기술 조사

조사 목적 및 할일 :

목적 : 낙상 감지 기술 도입 가능 여부 검토

할일

- 도입 가능 여부 검토
 - 낙상 감지 센서 활용 방안
 - 필요 센서 구매 가능 여부
 - 레퍼런스/튜토리얼

1. 낙상 감지 센서 활용 방안

- 기울기 센서
 - 아두이노 우노 키트 포함
 - 구조: 단순한 **볼 타입** 또는 **수은 스위치**로 구성.
 - 원리: 기울어지면 내부 볼이 접점에 닿아 회로가 연결(ON) 또는 분리(OFF).
 - 출력: 단순 **디지털 값(0/1)** → 기울어짐 여부만 감지.
 - 장점: 저렴, 간단.
 - 단점: 방향·속도·회전량 측정 불가, 오작동률 높음.
- MPU 자이로 센서
 - 구매 필요
 - 구성: **3축 가속도계** + **3축 자이로스코프**(회전속도) → 6축 IMU.
 - 출력: X/Y/Z 축 가속도(m/s^2), 회전속도($^{\circ}/s$), 중력 벡터까지 포함.
 - 장점: 낙상 시 충격, 방향변화, 가속도 변화량을 수치로 정확히 계산 가능.
 - 단점: 프로그래밍 복잡, 약간 비쌘.

2. 낙상 감지 로직

- 기울기 센서
 - ```
1 if (센서_출력 == 넘어짐 상태) && (10초 이상 변화 없음):
2 낙상 경고 발생
```
  - 단점: 앉아있거나 누워있는 상태를 낙상으로 오인 가능.
  - 장점: 구현 간단.
- MPU 자이로

## ▪ MPU 자이로 센서를 활용한 낙상 감지 로직:

1. **가속도 변화 감지:** 센서의 **\*\*총 가속도( $TotalAcceleration = \sqrt{a_x^2 + a_y^2 + a_z^2}$ )\*\***를 지속적으로 측정합니다.
2. **낙상 초기 단계 (충격 감지):** 사람이 넘어질 때 발생하는 **\*\*갑작스러운 충격(높은 가속도 값)\*\***을 감지합니다.
3. **낙상 후 단계 (자세 변화 감지):** 충격 감지 후, 센서의 **자세(기울기)**가 **장시간(예: 15초) 동안 수평에 가깝게 유지**되는지 확인합니다. 사람이 넘어지면 일반적으로 바닥에 누워있는 자세가 되므로, 가속도계의 Z축 값이 0에 가까워집니다.
4. **일상 동작과 구분:**
  - 일반적인 앉기, 눕기, 서기 등은 가속도 변화가 크지 않거나, 충격 이후 곧바로 자세가 바뀌는 패턴을 보입니다.
  - 낙상 시에는 **\*\*'높은 가속도 변화(충격)'\*\***와 **'수평 자세 유지'** 두 가지 조건이 연속적으로 충족됩니다.
5. **알림 발송:** 두 가지 조건이 모두 충족되면 '낙상'으로 판단하고 알림을 보냅니다.

MPU 자이로 센서는 가속도와 각속도 데이터를 함께 분석하므로, 일상적인 동작과 낙상을 훨씬 정교하게 구분할 수 있어 신뢰도가 높습니다.

### ◦ MPU 3축 자이로(가속도계 + 회전속도)

#### ▪ 측정값

- 3축 가속도(X, Y, Z)
- 3축 중력 벡터에서 갑작스러운 변화 → 충격 감지

#### ▪ 낙상 패턴 예시

1. **자유낙하 구간:** 가속도 합( $|a|$ )이 **~0g**(중력 상쇄) 근처로 감소 (0.2~0.4g)
2. **충격 구간:**  $|a|$  급상승(>2g~3g)
3. **정지 구간:** 이후 가속도 변화 미미(0.9~1.1g 유지)

- 1 **if** (자유낙하 감지) and (충격 감지) and (이후 10초 정지):  
2 낙상 경고 발생

### ◦ MPU 6축 자이로(3축 가속도계 + 3축 자이로스코프)

#### ▪ 측정값

- 3축 가속도 + 3축 회전속도(°/s)

#### ▪ 추가 가능 로직

- 회전 패턴 분석 → 낙상 시 비정상 회전량(roll/pitch/yaw) 급격 변화를 감지.
- 예: 앞/뒤/측면 넘어짐 구분 가능.

#### ▪ 낙상 패턴 예시

1. 자유낙하 감지 (3축 가속도 감소)

2. 회전속도 급증(>300°/s)
3. 충격 감지(가속도 급상승)
4. 이후 무동작 상태 유지

- 1 if (자유낙하) and (회전속도 급증) and (충격) and (무동작):
- 2     낙상 경고 발생 + 넘어짐 방향 분석

#### • 센서 도입 요소 비교 표

| 구분        | 기울기 센서          | MPU 3축 | MPU 6축            |
|-----------|-----------------|--------|-------------------|
| 측정 값      | ON/OFF(기울어짐 여부) | 3축 가속도 | 3축 가속도 + 3축 회전 속도 |
| 구현 난이도    | 매우 쉬움           | 중간     | 높음                |
| 낙상 감지 정확도 | 낮음              | 중간~높음  | 높음                |
| 방향 분석     | 불가              | 제한적    | 가능                |
| 패턴 분석     | 불가              | 가능     | 가능                |

#### 3. 필요 센서 구매 가능 여부

##### ◦ MPU 3축 : 가능

- 구매 URL : [아두이노 3축 자이로 + 가속도 센서 MPU-6050 GY-521, MPU6050 GY..., 1개](#)



The screenshot shows a Coupang product page for the MPU6050 GY-521 sensor module. The product is an Arduino-compatible 3-axis gyroscope and accelerometer module. The price is 1,590원, and the shipping cost is 2,500원. The page also shows the product's specifications and a link to the product page.

#### 4. 레퍼런스/튜토리얼

##### ◦ Tutorial

- [MPU6050 Sensor Arduino Tutorial](#)

- [📺 mpu 6050 arduino tutorial for beginners](#)

## ◦ Reference

- [📺 노인 낙상 감지\(App + 아두이노\)](#)
- [📺 아두이노 가속도센서를 활용한 낙상사고 알림전송](#)

## ◦ Code Example

- MPU-6050 기반 낙상 감지 (가속도+자이로)

문단 시작 아래 코드는 자유낙하 → 충격 → 무동작 + 자이로 급증을 조합해 간단 판정합니다. (I<sup>2</sup>C, 기본 레지스터 사용)

- ```
1 // Arduino C++ 예제: MPU-6050 낙상 감지 (간단 임계값 버전)
2 // 하드웨어: Arduino Uno, MPU-6050 (VCC=5V 가능 모듈 or 3.3V, SDA=A4, SCL=A5)
3 // 라이브러리 없이 최소 레지스터 접근 (실무에선 MPU6050/MPU_DMP 라이브러리 권장)
4
5 #include <Wire.h>
6
7 const uint8_t MPU_ADDR = 0x68;
8
9 // --- 튜닝 파라미터 (사용자 환경에 맞게 조정) ---
10 const float G = 9.80665;
11 const float FREEFALL_G = 0.4; // 자유낙하 임계(|a| < 0.4g)
12 const float IMPACT_G = 2.5; // 충격 임계(|a| > 2.5g)
13 const float GYRO_SPIKE = 300.0; // 회전속도 임계(°/s)
14 const unsigned long INACT_MS = 10000; // 무동작 시간 10초
15
16 // 상태 추적
17 bool freefallDetected = false;
18 bool impactDetected = false;
19 unsigned long impactTime = 0;
20
21 void mpuWrite(uint8_t reg, uint8_t val){
22   Wire.beginTransmission(MPU_ADDR);
23   Wire.write(reg); Wire.write(val);
24   Wire.endTransmission();
25 }
26
27 void readMPU(int16_t &ax,int16_t &ay,int16_t &az,int16_t &gx,int16_t &gy,int16_t &gz){
28   Wire.beginTransmission(MPU_ADDR);
29   Wire.write(0x3B); // ACCEL_XOUT_H
30   Wire.endTransmission(false);
31   Wire.requestFrom(MPU_ADDR, 14, true);
32   ax = (Wire.read()<<8)|Wire.read();
33   ay = (Wire.read()<<8)|Wire.read();
34   az = (Wire.read()<<8)|Wire.read();
35   Wire.read(); Wire.read(); // temp skip
36   gx = (Wire.read()<<8)|Wire.read();
37   gy = (Wire.read()<<8)|Wire.read();
38   gz = (Wire.read()<<8)|Wire.read();
39 }
40
41 void setup(){
42   Serial.begin(115200);
43   Wire.begin();
```

```

44 mpuWrite(0x6B, 0x00); // PWR_MGMT_1: wake up
45 mpuWrite(0x1B, 0x00); // GYRO_CONFIG:  $\pm 250^\circ/\text{s}$ 
46 mpuWrite(0x1C, 0x00); // ACCEL_CONFIG:  $\pm 2g$ 
47 delay(100);
48 Serial.println("MPU6050 fall detection start");
49 }
50
51 float norm3(float x, float y, float z){ return sqrtf(x*x + y*y + z*z); }
52
53 bool isInactive(){
54     // 간단 무동작: 최근 N 샘플의 가속도 변화가 작으면 true
55     static float prevA = 0;
56     static unsigned long stableSince = millis();
57     int16_t ax, ay, az, gx, gy, gz;
58     readMPU(ax, ay, az, gx, gy, gz);
59     // 스케일 변환: 데이터시트 기준( $\pm 2g$ ,  $\pm 250^\circ/\text{s}$ )
60     float aX = ax/16384.0f, aY = ay/16384.0f, aZ = az/16384.0f; // 단위 g
61     float aNorm = norm3(aX, aY, aZ);
62
63     if (fabs(aNorm - prevA) < 0.05) { // 변화 0.05g 이내면 안정으로 간주
64         if (millis() - stableSince > INACT_MS) return true;
65     } else {
66         stableSince = millis();
67     }
68     prevA = aNorm;
69     return false;
70 }
71
72 void loop(){
73     int16_t ax, ay, az, gx, gy, gz;
74     readMPU(ax, ay, az, gx, gy, gz);
75
76     float aX = ax/16384.0f, aY = ay/16384.0f, aZ = az/16384.0f; // g
77     float gX = gx/131.0f, gY = gy/131.0f, gZ = gz/131.0f; //  $^\circ/\text{s}$ 
78     float aNorm = norm3(aX, aY, aZ);
79     float gNorm = norm3(gX, gY, gZ);
80
81     // 1) 자유낙하
82     if (!freefallDetected && aNorm < FREEFALL_G) {
83         freefallDetected = true;
84         Serial.println("[DEBUG] Free-fall detected");
85     }
86
87     // 2) 충격 + 자이로 스파이크
88     if (freefallDetected && !impactDetected && (aNorm > IMPACT_G || gNorm >
GYRO_SPIKE)){
89         impactDetected = true;
90         impactTime = millis();
91         Serial.println("[DEBUG] Impact/gyro spike detected");
92     }
93
94     // 3) 무동작 유지 → 낙상
95     if (impactDetected && (millis() - impactTime > INACT_MS) && isInactive()){
96         Serial.println(">>> FALL DETECTED (MPU) <<<");
97         // TODO: 알림 발송 / 부저 / MQTT publish
98         // 상태 초기화(원한다면 유지)
99         freefallDetected = impactDetected = false;
100     }

```

```
101  
102     delay(20); // 50Hz  
103 }  
104
```