

UROP(Undergraduate Research Opportunity Program) 프로그램

[K-board] 과제 수행 결과보고서

1. 과제 현황

팀 명	K-board	지도교수	성 명	유진우
수행학기	2019학년도 하계방학		소 속	자동차융합대학
과 제 명	전동킥보드 안전성 향상 아이디어 및 시제품 개발			
과제유형	<input checked="" type="checkbox"/> 시제품 <input checked="" type="checkbox"/> 아이디어 <input type="checkbox"/> 소프트웨어 <input type="checkbox"/> 논문			

2. 참여 학생 현황

구분	학부(과)	학번	이름	휴대전화	E-Mail
대표	자동차공학과	20153357	서덕민	010-7582-8911	kookmin623@naver.com
팀원	자동차공학과	20153370	박재형	010-9990-9748	jjangbrett@kookmin.ac.kr
팀원	자동차공학과	20153343	류정연	010-5225-1498	jyr099@kookmin.ac.kr
팀원	자동차IT융합학과	20153409	명다울	010-9227-5236	daul@kookmin.ac.kr

3. 과제 결과 (혹은 학기 예상 결과 기입)

주요 성과	항 목	참가 내용 및 결과
	공모전/학회 참가	공모전/학회 참가 x
과제 결과 및 개선방안	과제 결과 1. 댐퍼 장착 시 Slide_crank 형식의 link에서 착안한 방식의 댐퍼 장착을 하였으나 회전시의 2차원 회전이 아닌 3차원 회전을 고려하지 않은 설계로 인하여 damper를 통한 실험 중 damper를 고정하는 부분과 link가 파손이 되었다. 그로 인해 damper를 장착한 후의 충분한 실험 결과를 제대로 얻지 못하였다. 2. Side LED 장착의 경우 또한 설문조사를 통한 결과로 안정성에 있어서 장착 할 경우 t-test를 통한 가설검정 결과 유의미한 결과를 얻었다고 판단할 수 있었다.	

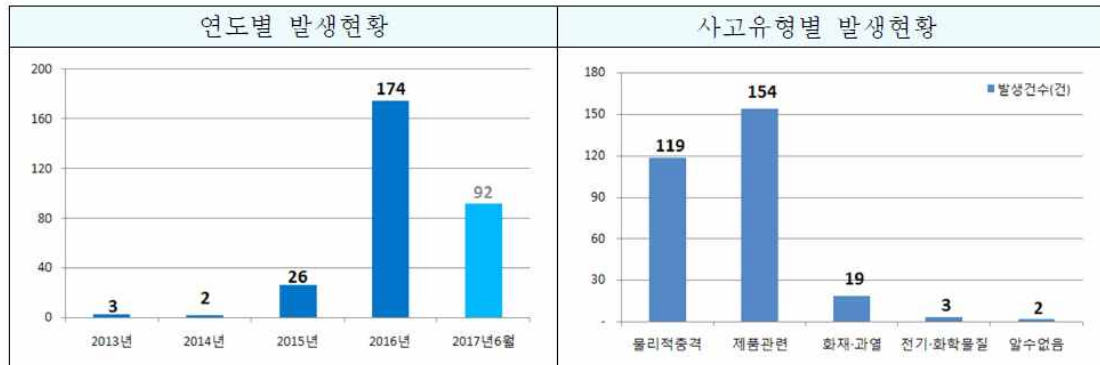
<p>과제 결과 및 개선방안</p>	<p>개선방안</p> <ol style="list-style-type: none"> 1. 댐퍼 장착 시 링크와 링크의 연결을 보다 부드럽게 연결하기 위한 베어링의 장착 시 1번 링크와 3번링크의 베어링의 회전에 대한 마찰을 다르게 하였다면 보다 좋은 결과를 얻을 수 있을 것이다. 2. 댐퍼 장착 시 body의 배터리를 분리하지 못하여 고정을 정확히 하지 못하여 파손이 일어났으므로 배터리의 위치를 정확히 파악하여 나사를 통한 고정을 한다면 댐퍼를 통한 진동감소의 효과를 보다 정확한 실험값으로 확인 할 수 있었을 것이다. 3. 제품 개발 시 시간 및 예산이 부족하여 CATIA를 통한 모델링과 댐퍼 장착, LED 구입 , 설문조사 등 에 생각 이외의 시간과 예산을 뺏겨서 3D 프린터를 통한 센서 case를 부착하는 것을 구현하지 못하였다. 실험시간과 예산이 조금 더 있었다면 보다 완성된 제품을 개발 할 수 있었을 것이다. 4. 제품 구매 후 DC 컨버터를 통한 배터리에서의 직접적인 전원공급 장치를 만들려고 했으나 배터리 분해 방법을 몰라 건전지를 이용하여야 했었다. 제품 구매 전 배터리의 분해방법을 정확히 조사를 하였다면 보다 완성된 제품을 개발 할 수 있었을 것이다. 5. 구조해석을 진행할 때 여러 환경에 대한 조건을 설정하여 시뮬레이션을 진행하고 싶었으나 시간의 제약과 프로그램의 한계로 인해 모든 실험결과를 얻지 못하였다. 좀 더 시간이 있고 프로그램 숙련도를 높힌다면 지면의 구조, 마찰 및 날씨 등에 따른 다양한 결과를 얻을 수 있었을 것이다. 6. 진동 측정을 위한 실험 시 아두이노와 컴퓨터 간의 시리얼 통신을 위해 센서, 아두이노, 노트북을 유선으로 연결한 상태로 노트북을 가방에 넣고 실험자가 가방을 맨채로 주행 및 실험을 진행하였다. 이 과정에서 excel의 data를 6만9천개 까지 밖에 기록을 하지 못하여 1회의 주행이 끝날 때 마다 노트북을 꺼내 확인해야 하는 등 불편함이 있었다. 노트북을 다른 노트북으로 원격 제어 하거나 아두이노와 노트북 간의 통신을 유선이 아닌 블루투스나 와이파이를 이용한 무선통신 시스템을 구축했다면 더욱 빠르고 편리하게 실험을 진행할 수 있었을 것이다.
------------------------------------	---

3-1) 내용

제품의 용도 및 필요성 :

출, 퇴근 및 등, 하교시의 전동킥보드 사용 인원 및 사고증가에 대한 보도자료(뉴스와 보고서)를 통해서 전동킥보드 안전성을 높이는 연구가 필요하다고 판단.

< 개인형 이동수단 안전 사고 발생현황 >



[출처 : 한국소비자원 소비자위해감시시스템]

판매중인 전동킥보드를 사전 조사해본 결과 법안 제한 속도 25km/h 이지만 대부분 개조를 통해서 제한속도를 어기는 사람이 많은 편이고, 주행 시 조향장치의 진동을 줄여주는 기본적인 댐퍼가 장착되지 않은 전동킥보드가 대부분으로 파악됨.

이로 인해 발생하는 사용 후기 대부분이 빠르지 않은 주행 속도 (약,20km/h)에서도 핸들의 떨림이 손에 많이 들어와 주행 시 불안하다는 후기가 많다는 사전조사 결과를 얻음, 따라서 행정안전부, 한국소비자원이 발표한 사고유형 중 제품관련을 제외한 사고원인 1순위인 물리적 충격중 주요원인인 미끄러짐 및 넘어짐, 충돌(물리적 충격의 96.4% 차지)을 개선 및 예방 할 수 있는 프로젝트를 진행함

□ 위해내용

- (위해원인) 전동형 개인 이동수단 위해사례 중 제품 관련(제품 성능, 화재 등) 건을 제외한 253건을 분석한 결과, 운행 중 미끄러지거나 넘어진 사고가 228건(90.1%)으로 가장 많았고, 충돌사고 16건(6.3%), 기타 9건(3.6%) 순임.

[전동형 개인 이동수단 위해사례 원인별 구분]

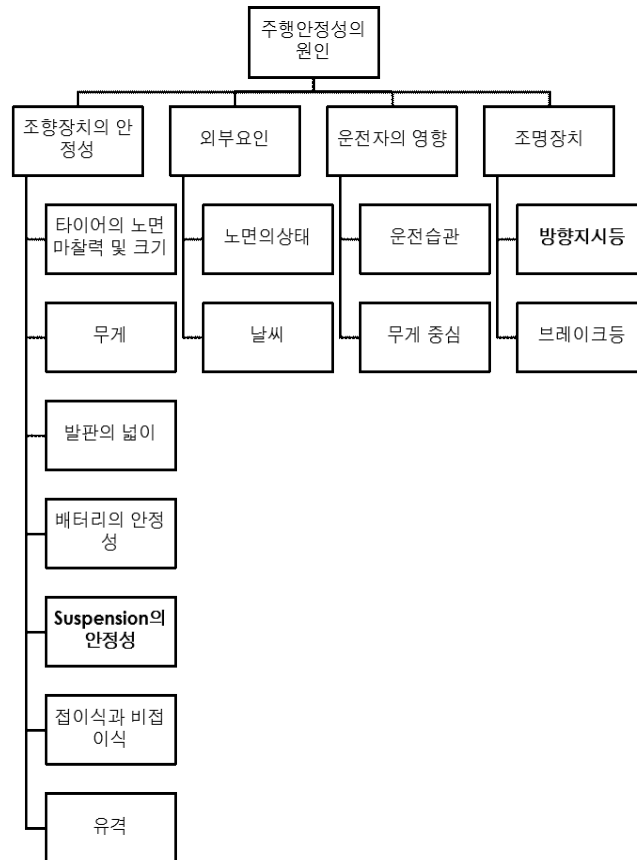
위해원인		건수	비율
제품 관련	제품결함	388건(87.8%)	442건(63.6%)
	기타(화재발생 등)	54건(12.2%)	
물리적 충격	미끄러짐, 넘어짐	228건(90.1%)	253건(36.4%)
	충돌	16건(6.3%)	
	기타	9건(3.6%)	
합계		695건(100.0%)	

제품의 기능 및 특성 :

1. K-board damper : 속도의 비례하는 힘을 발생시켜($\vec{F} = -c\vec{v}$) 주행 시 원하지 않는 타이어의 회전에 의해 발생하는 속도($\vec{v} = \vec{w} \times \vec{r}$)에 따라서 반대방향으로 힘을 발생시켜서 진동을 감소시킨다.
2. K-board Side LED : 아두이노를 이용하여 기울어짐을 측정하는 센서를 통하여 일정각도 이상 전동킥보드가 기울어질 경우 전동킥보드의 하부측면 쪽에서 자동적으로 기울어진 쪽에 LED가 들어와 접근 차량 및 보행자에서 주행방향을 표시한다.(본인의 이름을 표시하거나 점멸을 하는 등 customizing 가능)

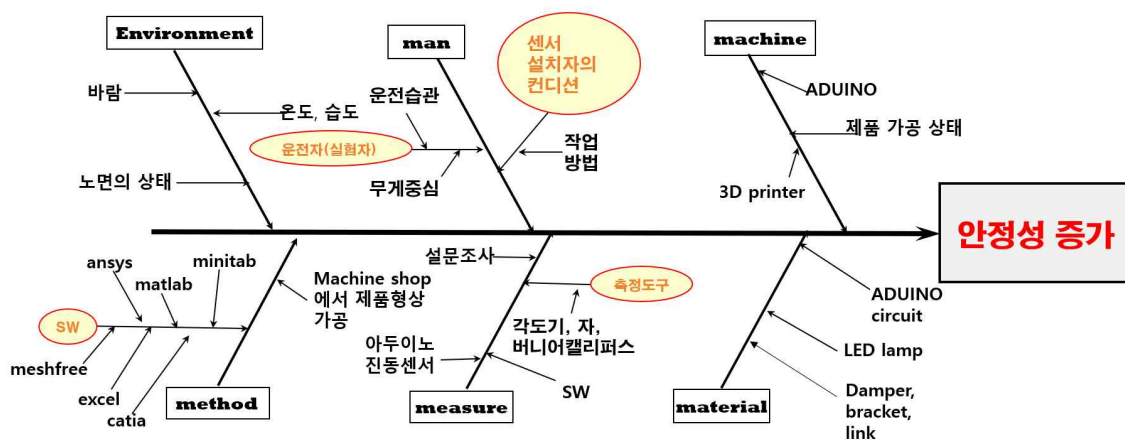
제품의 개발방법 및 과정 :

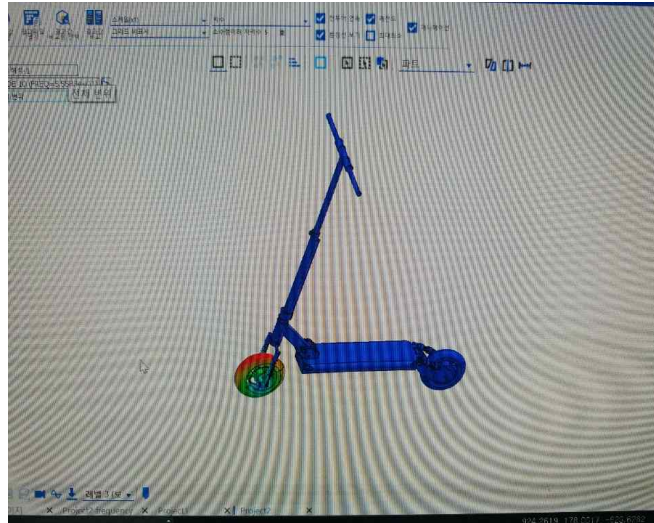
6sigma의 단계인 define 과정에서의 현 수준 및 문제점 파악으로 안정성의 주요원인 파악 및 catia 모델링을 통한 주행 시의 취약부분 파악 후 5why를 통하여 slide-crank 방식에서 착안한 형태의 유압댐퍼의 결정과 brain storming을 통하여 기울어짐을 통한 자동적인 side LED 제품의 개발로 진행방향을 결정



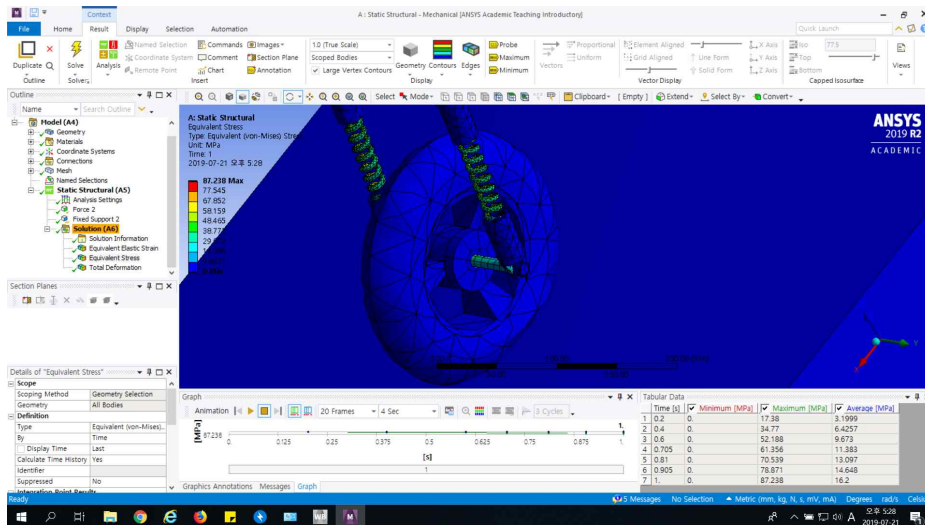
CTQ-tree 전개

Fish-bone diagram





stress analysis 1



stress analysis 2

해석조건
결과

전통 정적 해석-1
전체 변위



변위
전통 정적 해석-1
전체 변위

strain analysis

킥보드 보관상의 이유로 많은
사람들이 접이식 킥보드를 선
호한다.

킥보드 주행 분석시 바퀴의
축방향 진동이 가장 크게 측
정된다.

접이식 킥보드 특성상, 조향
장치 부근에 유격이 많이 발
생하며 접지면과의 거리가 팔
터 작은 각변위에도 큰 선변
위가 생성된다.

바퀴의 축회전을 감지할 수 있
도록 감쇄시키는 것은 안정성
에 도움이 된다.

damper 5why

뒷발을 이용하여 브레이크 또는 가속장치를
작동시, 자세가 불안정
하며 사고위험이 높아
진다.

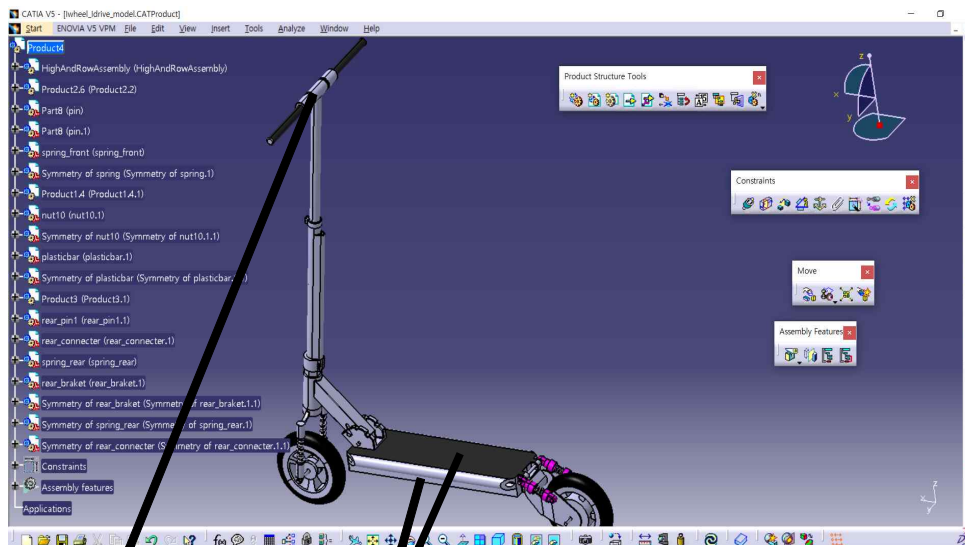
킥보드 주행시 양손
모두 브레이크 또는
가속장치에 위치해야
하며 움직이는 것에
제한이 있다.

가을기를 감지하여 사
이드 램프를 점등하는
것은 안정성을 위해
매우 좋다.

시중 전동 킥보드 구
조상 한 손잡이에는
브레이크 다른 한 손
잡이에는 가속장치가
있는 것이 일반적이다.

킥보드 주행중 방향지
시등을 직접 조작하는
것은 어렵고 불안정한
일이다.

Side LED 5why



기울어짐 측정 센서
부착

LED 장착

SIDE LED 장착 예시



Vibration's Cause & Effect Matrix

		7.5		7.5			
	떨림의 요소	가속도		진폭		합	순위
7	댐퍼의 유무	7	52.5	8.5	63.75	116.25	1
9	주행속도	7	52.5	7.5	56.25	108.75	2
1	노면의상태	3	22.5	9	67.5	90	3
6	실험자(운전자)	6.5	48.75	5.5	41.25	90	3
5	무게	6	45	5	37.5	82.5	5
2	전고	4	30	5	37.5	67.5	6
3	진동측정장비	2.5	18.75	2	15	33.75	7
8	센서부착위치	1.5	11.25	2.5	18.75	30	8
4	날씨	1.5	11.25	1	7.5	18.75	9



주요 인자 CNX 분석 후 제어 가능한 댐퍼의 유무, 주행속도, 실험자를 바꿔가며 실험결정



Constant - 날씨, 전고



X - 댐퍼의 유무, 주행속도



Noise - 날씨, 노면의 상태, 실험자

가속도 측정 시의 matlab FFT 코드

```
function [freq,xfft]=myfft(t,x)

Max_Number=max(size(t));
Max_Number=Max_Number(1);
dt=(t(Max_Number,1)-t(1,1))/Max_Number;

N=fix(log10(Max_Number)/log10(2));

xfft=fft(x(1:2^N));
xfft=abs(xfft(1:2^N/2))*dt;

freq=0;
freqf=(1/dt)/2;
df=freqf/(2^N/2);
freq=0:df:freqf-df;
freq=freq';
```

가속도 측정 시의 아두이노 코드

```
#include<Wire.h>

const int MPU_addr=0x68; // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
int16_t AcXi,AcYi,AcZi
void setup(){
  pinMode(2, INPUT_PULLUP);
  Wire.begin();

  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0);    // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);

  Wire.beginTransmission(MPU_addr);
  Wire.write(0x1C); // Accelerometer Configuration register
  //Wire.write(0x00); // AFS_SEL=0, Full Scale Range = +/- 2 [g]
  //Wire.write(0x08); // AFS_SEL=1, Full Scale Range = +/- 4 [g]
  Wire.write(0x10); // AFS_SEL=2, Full Scale Range = +/- 8 [g]
  //Wire.write(0x18); // AFS_SEL=3, Full Scale Range = +/- 16 [g]
  Wire.endTransmission(true);

  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
  AcXi=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcYi=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZi=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

  Serial.begin(9600);
  Serial.println("CLEARDATA"); // excel
  Serial.println("LABEL,Time,X,Y,Z");
}

void loop(){
  int btn = digitalRead(2); //스위치

  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
  AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

  if (btn==1){
    Serial.print("DATA, TIME,");
    Serial.print(float(AcX-AcXi)/4096*9.81);
    Serial.print(",");
    Serial.print(float(AcY-AcYi)/4096*9.81);
    Serial.print(",");
    Serial.println(float(AcZ-AcZi)/4096*9.81);
  }
  else{
    Serial.print("DATA, TIME,");
    Serial.print("0");
    Serial.print(",");
    Serial.print("0");
    Serial.print(",");
    Serial.println("0");
  }
  delay(150);
}
```

기울기 측정 및 SIDE LED 점멸시의 아두이노 코드

```
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"

#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

// class default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for SparkFun breakout and InvenSense evaluation board)
// AD0 high = 0x69
MPU6050 mpu;
//MPU6050 mpu(0x69); // <-- use for AD0 high

#define OUTPUT_READABLE_YAWPITCHROLL

#define INTERRUPT_PIN 2 // use pin 2 on Arduino Uno & most boards
#define LED_PIN 13 // (Arduino is 13, Teensy is 11, Teensy++ is 6)
bool blinkState = false;

// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, !0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorInt16 aa; // [x, y, z] accel sensor measurements
VectorInt16 aaReal; // [x, y, z] gravity-free accel sensor measurements
VectorInt16 aaWorld; // [x, y, z] world-frame accel sensor measurements
VectorFloat gravity; // [x, y, z] gravity vector
float euler[3]; // [psi, theta, phi] Euler angle container
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

// packet structure for InvenSense teapot demo
uint8_t teapotPacket[14] = { '$', 0x02, 0,0, 0,0, 0,0, 0x00, 0x00, '\r', '\n' };

// =====
// === INTERRUPT DETECTION ROUTINE ===
// =====

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

// =====
// === INITIAL SETUP ===
// =====

void setup() {
    pinMode(12, OUTPUT);
    pinMode(8, OUTPUT);
    digitalWrite(12, LOW);
    digitalWrite(8, LOW);
    // join I2C bus (I2Cdev library doesn't do this automatically)
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin();
        Wire.setClock(400000); // 400kHz I2C clock. Comment this line if having compilation difficulties
    #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
    #endif

    // initialize serial communication
    // (115200 chosen because it is required for Teapot Demo output, but it's
    // really up to you depending on your project)
    Serial.begin(115200);
    while (!Serial); // wait for Leonardo enumeration, others continue immediately

    // NOTE: 8MHz or slower host processors, like the Teensy @ 3.3v or Arduino
    // Pro Mini running at 3.3v, cannot handle this baud rate reliably due to
    // the baud timing being too misaligned with processor ticks. You must use
    // 38400 or slower in these cases, or use some kind of external separate
    // crystal solution for the UART timer.

    // initialize device
    Serial.println(F("Initializing I2C devices..."));
    mpu.initialize();
    pinMode(INTERRUPT_PIN, INPUT);

    // verify connection
    Serial.println(F("Testing device connections..."));
}
```

```

Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));

// load and configure the DMP
Serial.println(F("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(F("Enabling DMP..."));
  mpu.setDMPEnabled(true);

  // enable Arduino interrupt detection
  Serial.println(F("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(F("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 ");
  Serial.print(devStatus);
  Serial.println(F(")"));
}

// configure LED for output
pinMode(LED_PIN, OUTPUT);

mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
Serial.print("ypr\t");
Serial.print(ypr[0] * 180/M_PI);
Serial.print("\t");
Serial.print(ypr[1] * 180/M_PI);
Serial.print("\t");
Serial.println(ypr[2] * 180/M_PI);
int xi = ypr[0];
int yi = ypr[1];
int zi = ypr[2];
}

// =====
// ===                MAIN PROGRAM LOOP                ===
// =====

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) {
    // other program behavior stuff here
    // .
    // .
    // .
    // if you are really paranoid you can frequently test in between other
    // stuff to see if mpuInterrupt is true, and if so, "break;" from the
    // while() loop to immediately process the MPU data
    // .
    // .
    // .
  }

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

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

  // check for overflow (this should never happen unless our code is too inefficient)
  if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    // reset so we can continue cleanly

```

```

mpu.resetFIFO();
    Serial.println(F("FIFO overflow!"));

    // otherwise, check for DMP data ready interrupt (this should happen frequently)
} else if (mpuIntStatus & 0x02) {
    // wait for correct available data length, should be a VERY short wait
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

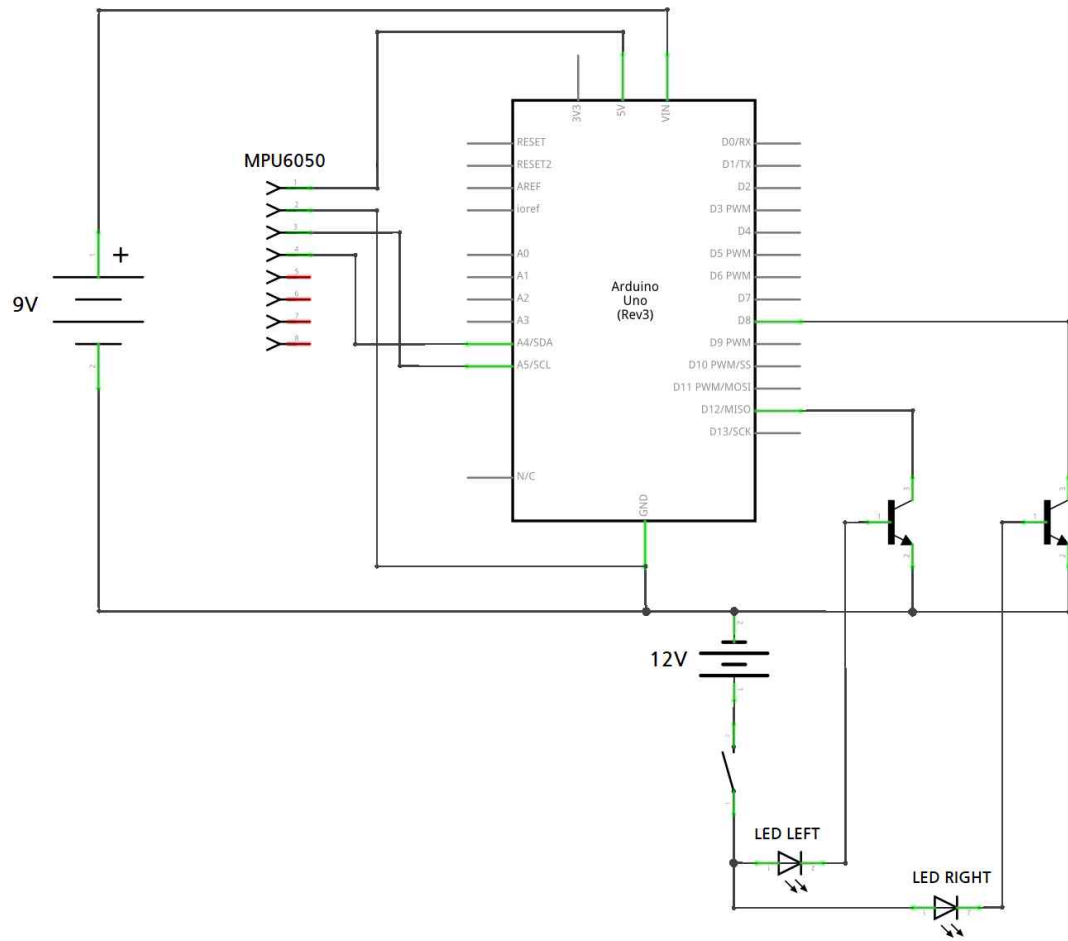
    // read a packet from FIFO
    mpu.getFIFOBytes(fifoBuffer, packetSize);

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

#ifdef OUTPUT_READABLE_YAWPITCHROLL
    // display Euler angles in degrees
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    Serial.print("ypr\t");
    Serial.print(ypr[0] * 180/M_PI);
    Serial.print("\t");
    Serial.print(ypr[1] * 180/M_PI);
    Serial.print("\t");
    Serial.println(ypr[2] * 180/M_PI);
#endif
    if (ypr[2] * 180/M_PI > 10){
        Serial.println("Left");
        digitalWrite(12, HIGH);
    } else {
        digitalWrite(12, LOW);
    }
    if (ypr[2] * 180/M_PI < -10) {
        Serial.println("Right");
        digitalWrite(8, HIGH);
    } else {
        digitalWrite(8, LOW);
    }

    // blink LED to indicate activity
    blinkState = !blinkState;
    digitalWrite(LED_PIN, blinkState);
}
}

```



SIDE LED 회로도

가속도 측정 matlab code

```
for i=1:4
j=int2str(i);
filename=sprintf('르즈ㅇ 25 %s.xlsx',j);
sheet=1;
xlRange='B:B';
data(i).exp25=xlsread(filename,sheet,xlRange);
s=size(data(i).exp25(:,1));
k=1;
    while (1)
        if data(i).exp25(k)==0
            data(i).exp25(k)=[];
            s(1)=s(1)-1;
        else
            k=k+1;
        end
        if k>s(1)
            break
        end
    end
end
for i=1:4
j=int2str(i);
filename=sprintf('류정연 10-%s.xlsx',j);
sheet=1;
xlRange='B:B';
data(i).exp10=xlsread(filename,sheet,xlRange);
s=size(data(i).exp10(:,1));
k=1;
    while (1)
        if data(i).exp10(k)==0
            data(i).exp10(k)=[];
            s(1)=s(1)-1;
        else
            k=k+1;
        end
        if k>s(1)
            break
        end
    end
end
for i=1:3
j=int2str(i);
filename=sprintf('ㅁ즈ㅎ 10-%s.xlsx',j);
sheet=1;
xlRange='B:B';
data(i+4).exp10=xlsread(filename,sheet,xlRange);
s=size(data(i+4).exp10(:,1));
k=1;
    while (1)
        if data(i+4).exp10(k)==0
            data(i+4).exp10(k)=[];
            s(1)=s(1)-1;
        else
            k=k+1;
        end
        if k>s(1)
            break
        end
    end
end
end
```

가속도 측정 matlab code

```
fs=1344;
for i=1:7
    s=size(data(i).exp10(:,1));
    data(i).time10=(linspace(0,s(1)/fs,s(1)))';
    [data(i).freq10,data(i).fftax10]=myfft(data(i).time10,data(i).exp10(:,1));
end

for i=1:4
    s=size(data(i).exp25(:,1));
    data(i).time25=(linspace(0,s(1)/fs,s(1)))';
    [data(i).freq25,data(i).fftax25]=myfft(data(i).time25,data(i).exp25(:,1));
end

subplot(2,2,3);

for i=1:7
    plot(data(i).freq10,data(i).fftax10*10);
    hold on;
end
legend();
xlim([1,200]);
grid on;
xlabel 'Hz';
ylabel 'cm/s^2';
title '10km/h accelation FFT No damper';

subplot(2,2,4);
for i=1:4
    plot(data(i).freq25,data(i).fftax25*10);
    hold on;
end
legend();
xlim([1,200]);
grid on;
xlabel 'Hz';
ylabel 'cm/s^2';
title '25km/h accelation FFT No damper';
```

가속도 측정 matlab code

```
subplot(2,2,1);

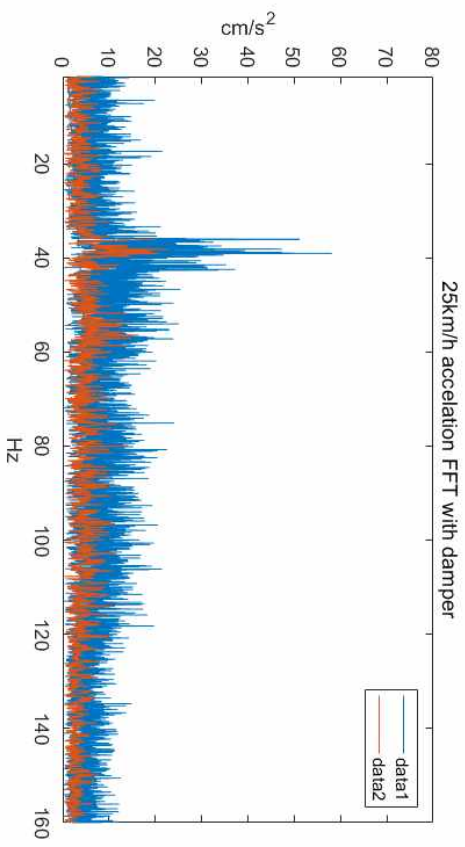
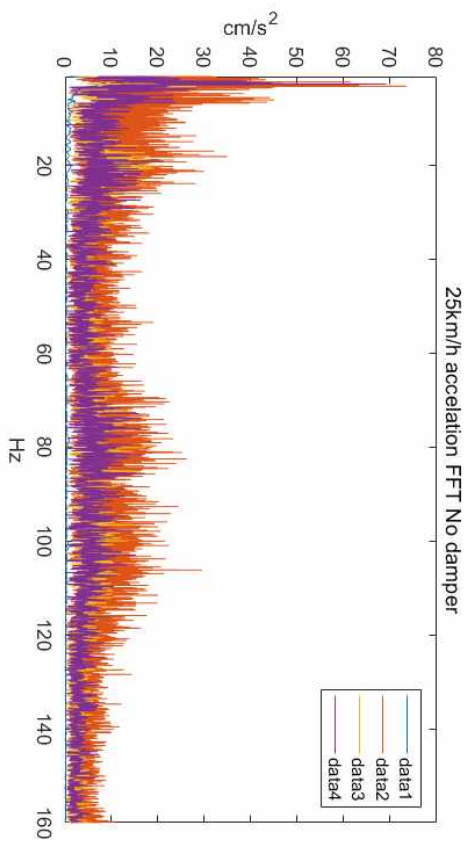
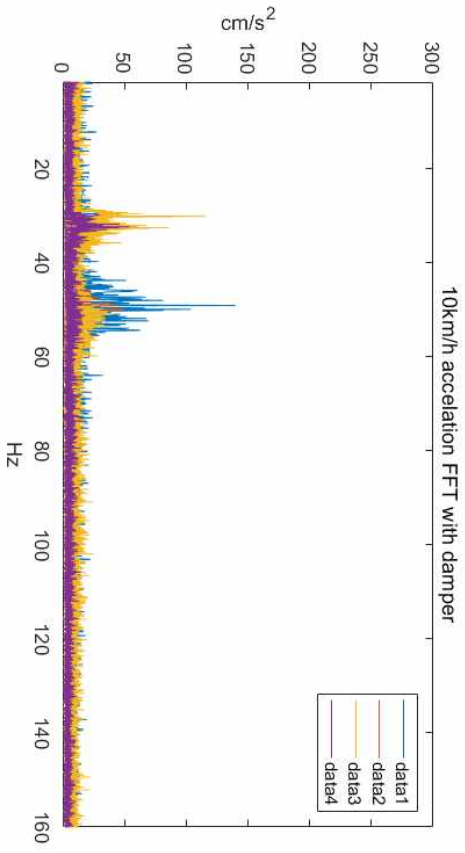
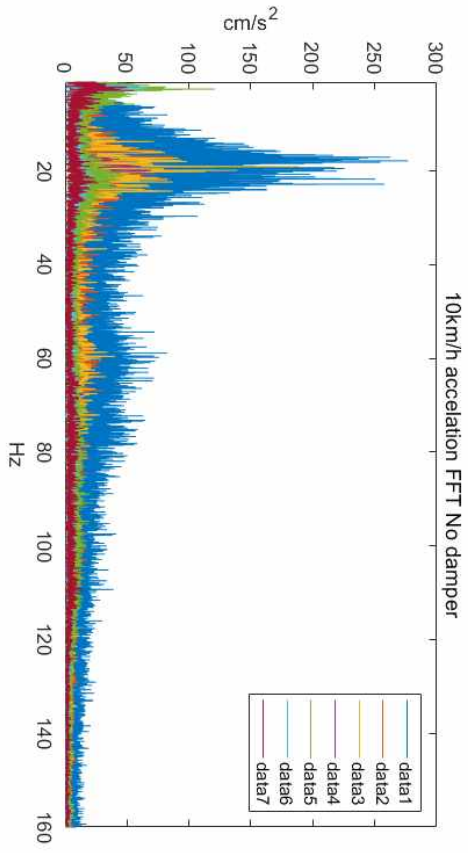
for i=1:7
    plot(data(i).freq10,data(i).fftax10*10);
    hold on;
end
legend();
xlim([1,160]);
grid on;
xlabel 'Hz';
ylabel 'cm/s^2';
title '10km/h accelation FFT';

subplot(2,2,2);

for i=1:4
    plot(data(i).freq25,data(i).fftax25*10);
    hold on;
end
legend();
xlim([1,160]);
grid on;
xlabel 'Hz';
ylabel 'cm/s^2';
title '25km/h accelation FFT';
subplot(2,2,3);

for i=1:7
    data(i).fftx10=data(i).fftax10./(data(i).freq10).^2;
    plot(data(i).freq10,data(i).fftx10*10);
    hold on;
end
legend();
xlim([1,5]);
% ylim([0,5]);
grid on;
xlabel 'Hz';
ylabel 'cm';
title '10km/h distance FFT';

subplot(2,2,4);
for i=1:4
    data(i).fftx25=data(i).fftax25./(data(i).freq25).^2;
    plot(data(i).freq25,data(i).fftx25*10);
    hold on;
end
legend();
xlim([1,5]);
% ylim([0,5]);
grid on;
xlabel 'Hz';
ylabel 'cm';
title '25km/h distance FFT';
```



신제품 t-test 결과

키크보드 운전자의 떨림, 흔들림 뿐만 아니라 사고위험 등을 모두 고려한 전반적인 주행 안정성

t-검정: 쌍체 비교			
	변수 1	변수 2	
평균	4.327586	7.534483	
분산	5.382033	2.744404	
관측수	58	58	
피어슨 상관 계수	0.186451		
가설 평균차	0		
자유도	57		
t 통계량	-9.44016		
P(T<=t) 단측 검정	1.49E-13		
t 기각치 단측 검정	1.672029		
P(T<=t) 양측 검정	2.97E-13		
t 기각치 양측 검정	2.002465		

움직임 예측/파악 가능 여부

t-검정: 쌍체 비교		
	변수 1	변수 2
평균	4	7.586207
분산	5.754385965	3.790684
관측수	58	58
피어슨 상관 계수	0.037563572	
가설 평균차	0	
자유도	57	
t 통계량	-9.007256802	
P(T<=t) 단측 검정	7.51197E-13	
t 기각치 단측 검정	1.672028889	
P(T<=t) 양측 검정	1.50239E-12	
t 기각치 양측 검정	2.002465444	

차량 운전자 입장에서의 사고 안정성

t-검정: 쌍체 비교		
	변수 1	변수 2
평균	3.758621	7.344828
분산	5.835451	3.422868
관측수	58	58
피어슨 상관 계수	0.266256	
가설 평균차	0	
자유도	57	
t 통계량	-10.4137	
P(T<=t) 단측 검정	4.17E-15	
t 기각치 단측 검정	1.672029	
P(T<=t) 양측 검정	8.33E-15	
t 기각치 양측 검정	2.002465	

기대효과 및 활용방안 :

기대효과

- 본 제품의 과제개발 목표를 통하여 댐퍼의 경우 장착을 할 경우 전동킥보드의 조향장치의 불안성을 감소시킴으로 안정적인 전동킥보드 주행환경 조성을 한다, 또한 전동킥보드의 경우 손으로 엑셀, 브레이크, 방향전환, 자세의 유지를 하는 만큼 SIDE LED를 통해서 편안하고 안정적인 전동킥보드 주행환경 조성을 한다.

활용방안

- SIDE LED의 경우, 제품개발 시 실용성 및 가격에 초점을 두어서 상용화 될 경우 많은 사람들이 손쉽게 구매하고 이용하여 전동킥보드 주행 시의 사고를 줄인다.

4. 역할 분담표				
연번	학부(과)	학번	성명	역할
1	자동차공학과	20153357	서덕민	전제적인 일정 관리 및 팀원 업무 지시, 보고서작성, CATIA 모델링, 구조해석, 6sigma 작성, matlab code 작성, 제품구매 시 운행지원, 실험 수행
2	자동차공학과	20153370	박재형	CATIA 모델링, 6sigma 작성, 기타 업무 보조, 아두이노 기울기 감지 코딩 및 가속도 측정 코딩, 실험 수행
3	자동차공학과	20153343	류정연	CATIA 모델링, 구조해석, 실험 장소 예약 및 제품 장착, 실험 수행
4	자동차IT융합학과	20153409	명다울	아두이노 회로도 작성 및 설계, 아두이노 기울기 감지 코딩 및 가속도 측정 코딩, 제품 구매 및 경리 업무, 실험 수행

5. 예산 집행 내역			
예산항목	금액(원)	품목 및 내용	비고
재료비	403,000	전동킥보드	
재료비	20,800	LED형 원형 토글 스위치, 고휘도 LED 5파이(5mm), LED용 저항 220옴 10pcs, 빨/검 절연전선 22AWG 1.65mm, 점퍼케이블 수-수 20cm 40핀, 브레드보드 400홀(불투명), 기울기센서(6축 가속도+자이로) MPU-6050, 아두이노 우노용 USB B 케이블 50cm, 아두이노 우노 Uno R3 DIP 호환보드	
재료비	17,400	자전거헬멧	
재료비	29,800	무릎 팔꿈치 보호대 세트	
재료비	3,800	9V 배터리 홀더	
재료비	8,190	9V 건전지 5개	
재료비	33,000	LED바 3개	
재료비	15,400	건전지 소켓 2개, AA건전지 30개	
재료비	250,000	댐퍼	
재료비	2,200	트랜지스터	
재료비	5,750	절연테이프, 양면테이프	
합 계	789,340		

위와 같이 학부생 연구 프로그램(UROP) 결과를 보고합니다.

2019 . 08 . 21 .

대표학생 서덕민 (인)

담당교수 유진우 (인)