AUTOMATIC BIKE CRASH DETECTION

*INTRODUCTION:* According to World Health Organisation a whopping 1.35 million people face a risk of life due to road traffic crash!! Many valuable lives are risked just because the help can’t reach in time. Imagine what if we reduce the time constant for the whole process of calling help and rescue many lives!!

*What is the use of developing a quick response system?*

Nick and Rick are two friends. Once while walking together by the road they saw the crowd gathering at a single point besides the road. They saw a man lying on the ground who faced an accident, asking for help. Nick stepped ahead and helped the person by calling the emergency services while Rick was silently watching everything. Nick will be appreciated for his move but the world is full of people like Rick and waiting for a Nick to step up takes too long. Rather than depending on the third source between a victim and emergency service, how convenient will it be if we directly connect both them!!

*How to implement a direct connection between the victim and emergency services?*

A victim is never in full sense to ask for help after an accident, so here comes an automatic crash detection device that will report a crash and details of the same on the webpage that the emergency services will have access to.

*COMPONENTS:* The device will consist of an MPU6050 sensor (accelerometer and gyro), Wi-Fi Module, GSM module, Arduino Uno, switch, LED, batteries, breadboard, GPS module.

*IMPLEMENTATION:*

• This saviour system is comprised of two devices. Device1 is then placed inside a black box and fitted on the vehicle. Device2 will be placed in a location with a 24x7 internet connectivity to receive the data from Device1 and upload the data on the website.

*• DEVICE1:*

• Device1 comprises of an Arduino board, GSM module, GPS module and MPU6050 sensor.

• The MPU6050 sensor will be connected to Arduino, it will measure the tilt of the box in which it is firmly placed on the vehicle, along with the orientation of the device it is also sensitive to acceleration and measures sudden changes in the acceleration.

• If the MPU6050 detects abnormal changes in orientation or acceleration it will trigger the glowing of the LED. The LED will glow for 2 minutes, for the demonstration purpose the snooze time is set to 30 seconds. This way the user gets time to switch off the device if it has generated a false alarm. After 2 minutes if the device is not shut off both the LEDs will continuously glow indicating the signal being sent for help.

• The GSM module connected with the Arduino board will send a message to the Device2, the message will include the coordinates of the accident obtained from GPS module, the vehicle number which will be already installed in the -device.

*• DEVICE2:*

• Device2 comprises of an Arduino board, GSM module, and an esp8266 Wi-Fi module.

• The signal received from Device1 will be received by the GSM module and will be passed on to the Wi-Fi module. The Wi-Fi module will upload the data on the website as it is being connected to internet 24x7.

• The webpage will display the location and details of the vehicle. The emergency service will act accordingly and update the webpage if the rescue has been sent or not.

For demonstration purpose only one device is shown attached with all the sensors and the corresponding output will be screened, as only one device is used GSM communication is disabled and direct Wi-Fi communication is established.

*CONCLUSION:*

With the use of this device the reaction process can be made faster, as the device is self-detecting it is independent of the third source between the victim and the emergency services. Still more people like Nick are encouraged in real life. So here it is “The Bike Crash Detection device” which will be your Nick forever!!

The code for the Arduino board is as follows:

#include <Wire.h>

#include <SoftwareSerial.h>

//Declaring some global variables

int killswitch=A0;

int reset=A1;

int killvalue;

int resetvalue;

const int MPU\_addr=0x68; int16\_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

int minVal=265; int maxVal=402;

SoftwareSerial Serial1(2,3);

String A="GET https://api.thingspeak.com/update?api\_key=W435O1P6X76Q2VJ4&field1=";

String B="GET https://api.thingspeak.com/update?api\_key=W435O1P6X76Q2VJ4&field2=";

String C="GET https://api.thingspeak.com/update?api\_key=W435O1P6X76Q2VJ4&field3=";

String Z=" HTTP/1.1 \nHOST: api.thingspeak.com \r\n\r\n";

double x; double y; double z;

double gx,gy,gz,AcX1,AcY1,AcZ1;

int flagax=0,avgax=0,sumax=0,countax=0;

int flagaz=0,avgaz=0,sumaz=0,countaz=0;

int flagacx=0,avgacx=0,sumacx=0,countacx=0;

int flagacy=0,avgacy=0,sumacy=0,countacy=0;

void setup() {

pinMode(13,OUTPUT);

pinMode(8,OUTPUT);

pinMode(9,OUTPUT);

pinMode(10,OUTPUT);

Wire.begin(); Wire.beginTransmission(MPU\_addr); Wire.write(0x6B); Wire.write(0); Wire.endTransmission(true); Serial.begin(115200);

Serial1.begin(115200);

String connection="AT+CWJAP=\"IITDH-Wireless-1\",\"!!tdh@KA25#\"";

Serial1.print(connection);

Serial.print(connection);

}

void loop() {

killvalue=analogRead(killswitch);

if(killvalue<900)

{digitalWrite(13,1);

Wire.beginTransmission(MPU\_addr); Wire.write(0x3B); Wire.endTransmission(false); Wire.requestFrom(MPU\_addr,14,true); AcX=Wire.read()<<8|Wire.read(); AcY=Wire.read()<<8|Wire.read(); AcZ=Wire.read()<<8|Wire.read(); int xAng = map(AcX,minVal,maxVal,-90,90); int yAng = map(AcY,minVal,maxVal,-90,90); int zAng = map(AcZ,minVal,maxVal,-90,90);

x= RAD\_TO\_DEG \* (atan2(-yAng, -zAng)+PI); y= RAD\_TO\_DEG \* (atan2(-xAng, -zAng)+PI); z= RAD\_TO\_DEG \* (atan2(-yAng, -xAng)+PI);

AcX1=abs(map(AcX,-20000,20000,0,1000)-400);

AcY1=abs(map(AcY,-20000,20000,0,1000)-400);

AcZ1=abs((map(AcZ,-20000,20000,0,1000)/2)-400);

gx=170-sqrt((180-x)\*(180-x));

gy=180-sqrt((180-y)\*(180-y));

gz=abs((180-sqrt((180-z)\*(180-z)))-125);

/\*Serial.print("AngleX= "); Serial.println(gx);Serial.println(AcX1);

Serial.print("AngleY= "); Serial.println(gy);Serial.println(AcY1);

Serial.print("AngleZ= "); Serial.println(gz);Serial.println(AcZ1); Serial.println("-----------------------------------------");\*/

if(countax==0&&flagax==0){

if(gx>50)

countax++;}

if(countax>0){

sumax=sumax+gx;

countax++;

if(countax==10)

{avgax=sumax/10;

countax=0;}

if (avgax>45)

flagax=1;}

else{

avgax=0;sumax=0;countax=0;

}

if(countaz==0&&flagaz==0){

if(gz>50)

countaz++;}

if(countaz>0){

sumaz=sumaz+gz;

countaz++;

if(countaz==10)

{avgaz=sumaz/10;

countaz=0;}

if (avgaz>40)

flagaz=1;}

else{

avgaz=0;sumaz=0;countaz=0;

}

if(countacx==0&&flagacx==0){

if(AcX1>300)

countacx++;}

if(countacx>0){

sumacx=sumacx+AcX1;

countacx++;

if(countacx==10)

{avgacx=sumacx/9;

countacx=0;}

if (avgacx>250)

flagacx=1;}

else{

avgacx=0;sumacx=0;countacx=0;

}

if(countacy==0&&flagacy==0){

if(AcY1>300)

countacy++;}

if(countacy>0){

sumacy=sumacy+AcY1;

countacy++;

if(countacy==10)

{avgacy=sumacy/10;

countacy=0;}

if (avgacy>275)

flagacy=1;}

else{

avgacy=0;sumacy=0;countacy=0;

}

if((flagaz+flagax+flagacx+flagacy)>1)

{Serial.println("crash");

digitalWrite(9,1);

delay(10000);

if(analogRead(A0)<900)

{float la,lo,bn;

//get the reading from gps

bn=5638;

char la\_buffer[16];

char lo\_buffer[16];

char bn\_buffer[16];

String lafinal=dtostrf(la,4,5,la\_buffer);

String lofinal=dtostrf(lo,4,5,lo\_buffer);

String bnfinal=dtostrf(bn,4,5,bn\_buffer);

String str=lafinal+lofinal+bnfinal;

String postStr1=A+lafinal+"&field2="+lofinal+"&field3="+bnfinal+Z;

Serial1.println("AT");

Serial.println("AT");

delay(2000);

Serial1.print("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n");

Serial.print("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n");

delay(3000);

String ciplength1="AT+CIPSEND="+String(postStr1.length())+"\r\n";

Serial1.print(ciplength1);

Serial.print(ciplength1);

delay(3000);

Serial1.print(postStr1);

Serial.print(postStr1);

delay(3000);

Serial1.print("AT+RST\r\n");

Serial.print("AT+RST\r\n");

delay(3000);

digitalWrite(9,0);

exit(0);

/\*resetvalue=analogRead(reset);

if(resetvalue>900)

{

flagaz=0;flagax=0;flagacx=0;flagacy=0;}

\*/}

}

delay(200);

}

else

{

digitalWrite(13,0);

flagax=0;avgax=0;sumax=0;countax=0;

flagaz=0;avgaz=0;sumaz=0;countaz=0;

flagacx=0;avgacx=0;sumacx=0;countacx=0;

flagacy=0;avgacy=0;sumacy=0;countacy=0;

digitalWrite(9,0);

}

// put your main code here, to run repeatedly:

}