**PROGRAM CODE:**

//Put the ADXL345 into +/- 4G range by writing the value 0x01 to the DATA\_FORM writeRegister(INT\_ENABLE, 0x04) ; // Enable interrupts for FRE//Add the SPI library; to communicate with the ADXL345 sensor #include <SPI.h>

#include <SoftwareSerial.h> SoftwareSerial SIM900(7, 8);

//Assign the Chip Select signal to pin 10 int CS=10;

const int redLEDPin = 4; // L E D connected to digital pin 4 const int greenLEDPin = 5; // L ED connected to digital pin 5 const int blueLEDPin = 6; // L ED connected to digital pin 6

//ADXL345 Register Addresses

#define DEVID 0x00 //Device ID Register #define THRESH\_TAP 0x1D //Tap Threshold

:

//This buffer will hold values read from the ADXL345 registers char values[10];

char output[20];

//These variables will be used to hold the x, y and z axis accelerometer values int x,y,z;

double xg, yg, zg;

char free\_fallEvent = 0;

void **setup**(){

SPI.begin(); //Initiate an SPI communication instance SPI.setDataMode(SPI\_MODE3); //Configure the SPI connection for the ADXL345 SIM900.begin(19200);

SIM900power();

delay(20000); // give time to log on to network

//Set up the Chip Select pin to be an output from the Arduino pinMode(greenLEDPin,OUTPUT); pinMode(redLEDPin,OUTPUT); pinMode(blueLEDPin,OUTPUT); digitalWrite(CS, HIGH);

//Create an interrupt that will trigger when an activity is detected attachInterrupt(0, fall, RISING);

EFALL writeRegister(POWER\_CTL, 0x08); }

//Equivalent to pressing the GSM shield "power" button void SIM900power(){

digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

delay(5000);}

void sendSMS(){

SIM900.print("AT+CMGF=1\r");

delay(100);

SIM900.println("AT + CMGS = \"+60162923680\"");

delay(100);

SIM900.println("Emergency! Falling incident happened."); delay(100);

SIM900.println((char)26); delay(5000); SIM900power(); }

void **loop**(){

digitalWrite(greenLEDPin, HIGH); digitalWrite(blueLEDPin, LOW); digitalWrite(redLEDPin, LOW);

:

//Reading 6 bytes of data starting at register DATAX0 readRegister(DATAX0, 6, values);

//The X value is stored in values[0] and values[1] x = ((int)values[1]<<8)|(int)values[0];

:

//Convert the accelerometer value to G's. xg = x \* 0.0078;

:

//Delay before next reading delay(100); if(free\_fallEvent == 1){

**Serial**.println("freefall"); **Serial**.print(x, DEC);

**:**

**Serial**.println(z, DEC); digitalWrite(greenLEDPin, LOW); do{

digitalWrite(redLEDPin, HIGH); delay(100); digitalWrite(redLEDPin, LOW); digitalWrite(blueLEDPin, HIGH); delay(100); digitalWrite(blueLEDPin, LOW); delay(100);

int picth = map(i,1000,1023,50,4000); tone(3,picth,10);

} while (a<=50); sendSMS(); free\_fallEvent = 0; }

detachInterrupt(0); delay(100);

attachInterrupt(0, fall, RISING); }

//This function writes a value to a register on the ADXL345 void writeRegister(char registerAddress, char value){

digitalWrite(CS, LOW); SPI.transfer(registerAddress); SPI.transfer(value); digitalWrite(CS, HIGH); }

void readRegister(char registerAddress, int numBytes, char \* values){ char address = 0x80 | registerAddress;

if(numBytes > 1)address = address | 0x40; digitalWrite(CS, LOW); SPI.transfer(address);

for (int i=0; i<numBytes; i++) { values[i] = SPI.transfer(0x00); } digitalWrite(CS, HIGH); }

void fall(void){

if((values[0] & 0x04)==0x04 ) free\_fallEvent = 1;

else free\_fallEvent = 0; readRegister(I NT\_SOURCE,1,values); }