```
 // MPU-6050 Short Example Sketch
// By Arduino User JohnChi
// August 17, 2014
// Public Domain
#include<Wire.h>
#include "MegunoLink.h"
#include "Filter.h"

//Megunolink objects
TimePlot AccGraph;
ExponentialFilter<float>xFilter(60,0),yFilter(60,0),zFilter(60,0);


const int MPU_addr=0x68;  // I2C address of the MPU-6050

 float AcX,AcY,AcZ; //Incoming accel data
 float AccX[10], AccY[10], AccZ[10]; //Accel data storage
 float meanX=0, meanY=0, meanZ=0; //mean of 30 data points
 float varX=0, varY=0, varZ=0;    //variance of 30 data points
 float xThresh = -.5, xThresh2 = -10, xThresh3 = .2, xThresh4 = 2, yThresh = .1,
yThresh2 = 1, yThresh3 = -1, yThresh4 = -6, zThresh = .150; //Threshold will be
compared to variance to determine valid gestures
 int16_t xmindex, xmaxdex, ymindex, ymaxdex, zmindex, zmaxdex;
int16_t datasample=0; //Accel data storage uses datasample to store Incoming
acceldata
const int16_t samplesize = 11;

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

//Graph setup
AccGraph.SetTitle("X, Y, and Z");
AccGraph.SetXlabel("Index");
AccGraph.SetYlabel("g exeperienced");
AccGraph.SetSeriesProperties("x-data", Plot::Blue, Plot::Solid, 2, Plot::Square);
AccGraph.SetSeriesProperties("y-data", Plot::Red, Plot::Solid, 2, Plot::Square);
AccGraph.SetSeriesProperties("z-data", Plot::Green, Plot::Solid, 2, Plot::Square);
AccGraph.SetSeriesProperties("X-variance", Plot::Blue, Plot::Dashed, 2,
Plot::Square);
AccGraph.SetSeriesProperties("Y-variance", Plot::Red, Plot::Dashed, 2, Plot::Square);
AccGraph.SetSeriesProperties("Z-variance", Plot::Green, Plot::Dashed, 2,
```

```
Plot::Square);

}



float mean(float *data, int16_t maxdex, int16_t mindex)
{
  float sum = 0, mean;
  float sizesample = abs(maxdex-mindex) + 1;
  if (mindex < maxdex)
  {
  for(int i =mindex; i < maxdex; i++)
  {
    sum = data[i] + sum;
  }
  }

    if (maxdex < mindex)
  {
  for(int i = maxdex; i < maxdex; i++)
  {
    sum = data[i] + sum;
  }
  }

  mean = sum/sizesample;
  return mean;
}




float var(  float *data,  float mean,  int16_t maxdex,  int16_t mindex)
{
 float indexSample;
  indexSample = maxdex-mindex;
  indexSample += 1;
 float sum = 0, var = 0;

 if ( mindex < maxdex )
 {
  for(int i = mindex; i< maxdex; i++)
  {
    sum += (data[i]*data[i]);
  }
```

```
  var = sum /indexSample;
  var = var - (mean*mean);

}


 if ( maxdex < mindex )
 {
  for(int i = maxdex; i< mindex; i++)
  {
    sum += (data[i]*data[i]);
  }
  var = sum /indexSample;
  var = var - (mean*mean);

}




return var;

}
int findMax(float *data)
{
  float maxVal;
  int index = 0, maxdex;
  maxVal = data[index];
  index++;

  for(index; index < samplesize; ++index)
  {
    if (data[index] > maxVal )
    {
      maxVal = data[index];
      maxdex = index;
    }
  }
  return maxdex;
}

int findMin(float *data)
{
  float minVal;
  int index = 0, mindex;
```

```
  minVal = data[index];
  index++;

  for(index; index < samplesize ; index ++)
  {
    if (data[index] < minVal)
    {
      minVal = data[index];
      mindex = index;
    }
  }
return mindex;
}


void loop(){

 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

  //reading data into varaibles
  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)
 //log filter for each axis---smoothing out the numbers
  xFilter.Filter(AcX);
  yFilter.Filter(AcY);
  zFilter.Filter(AcZ);
  //Storing data into arrays
  AccX[datasample] = xFilter.Current()/10000;
  AccY[datasample] = yFilter.Current()/10000;
  AccZ[datasample] = zFilter.Current()/10000;

//Priniting Data
//Serial.print("X:");Serial.println(abs(AccX[datasample]));
//Serial.print("Y:");Serial.println(AccY[datasample]);
// Serial.print("Z: "); Serial.println(AccZ[datasample]);

//Plotting Data
//AccGraph.SendData(F("x-data"), AccX[datasample]);
//AccGraph.SendData(F("y-data"), AccY[datasample]);
//AccGraph.SendData(F("z-data"), AccZ[datasample]);

//increasing index for future storage
```

```
 datasample += 1;
//are arrays full? If yes begin data analysis
 if (datasample==samplesize){
  datasample = 0;




//Data Analysis//

 //***Finding min and max for each axis****//
 xmaxdex = findMax(AccX);
 delay(5);
 xmindex = findMin(AccX);
delay(5);
 ymaxdex = findMax(AccY);
delay(5);
 ymindex = findMin(AccY);
delay(5);
 zmaxdex = findMax(AccZ);
delay(5);
 zmindex = findMin(AccZ);
delay(5);




//-----------------------------------//




//****Calculating mean for each Axis***//
     meanX = mean(AccX, xmaxdex, xmindex);
     meanY = mean(AccY, ymaxdex, ymindex);
     meanZ = mean(AccZ, zmaxdex, zmindex);



//-----------------------------------//
```

```
//***Calculating var for each Axis***//
  varX = var(AccX, meanX, xmaxdex, xmindex);

 varY = var(AccY, meanY, ymaxdex, ymindex);
//delay(5);



 //------------------------------//



//**********************Compare Varaince with
Thresholds*********************************//
if((varX <= xThresh) && (varX >= xThresh2))
 {
  if (xmindex > xmaxdex)
  {
    Serial.println("Hand is up");
   xmaxdex = 0; //reset max and min
    xmindex = 0;
 }

 }
 if((varX >= xThresh3) && (varX <= xThresh4)){

  if (xmaxdex > xmindex) {
    Serial.println("Hand is down");
    xmaxdex = 0;
    xmindex = 0;
  }

 }


if ((varY >= yThresh) && (varY <= yThresh2))
{
  if (ymaxdex > ymindex)
  {
    Serial.println("Hand is tilted left");
    ymaxdex = 0; // reset max and min
    ymindex = 0;
  }
}

 if (varY <= yThresh3 && varY >= yThresh4)
```

```
  {
    if(ymindex > ymaxdex)
    { Serial.println("Hand is titled right");
    ymaxdex = 0;
    ymindex = 0;
    }

    }









//}

//}

//-------------------------------------------------------------------------------
----//









}
}
```