Motion Handling

Pseudocode

struct axis {

int SPIcode;

int threshold;

int[] data;

int currentIndex;

int didBreak;

int numBreaks;

int sum;

int localMax;

}

void readAxis(struct\* axis) {

sendSPI(SPIcode);

data[index]=receiveSPI();

index = (index+1) % data.size;

if (data > threshold && didBreak == 0)

didBreak = 1;

}

void monitorBreakthrough(struct\* axis) {

if(didBreak)

sum += data[index];

numBreaks++;

else if(sum != 0)

int result = badData(axis);

if(result == 0)

localMax = findMax(axis);

sum = 0;

numBreaks = 0;

didBreak = 0;

}

int findMax(struct\* axis) {

int average = sum/numBreaks;

return 2\*average;

}

int badData(struct\* axis) {

int numExtraPoints = [[[???]]];

float ratio = [[[???]]]?

int count = 0;

int moreSum = 0;

int i;

for(i=0; i<extraPoints; i++)

readAxis(axis);

if(didBreak)

count++;

moreSum += data[index];

if(count/extraPoints > ratio)

sum += moreSum;

numBreaks += numExtraPoints;

return 1;

else

return 0;

}

void main\_accelerometer() {

int throwResult;

init\_accelerometer();

while(1) {

readAxis(X1);

monitorAxis(X1);

readAxis(X0);

monitorAxis(X0);

readAxis(Y1);

monitorAxis(Y1);

readAxis(Y0);

monitorAxis(Y0);

readAxis(Z1);

monitorAxis(Z1);

readAxis(Z0);

monitorAxis(Z0);

throwResult = checkMaximums();

if(throwResult == 1)

goodThrow();

else if (throwResult == -1)

badThrow();

else;

}

}