/\*

 \* Project homehack

 \* Description: Laundry basket that determines when it is full and publishes event

 \* Author: Ben Fisher

 \* Date: 1/29/18

 \*/

 int ledPin = D0; // Establish pin for the LED

 int proxPin = A0; // Establish input pin for PIR sensor [PROX SENSOR]

 //int pirState = LOW; // Assume no motion detected at start

 int buzzPin = D2;

 int val = 0; // Variable for reading the pin status

 int score=0; // Counter variable for shots made

 int other\_sensor=0; // Used for delay between both sensors [BETWEEN CHECKING THE SAME SENSOR IN MY CASE]

 int this\_sensor=0; // Used for delay between both sensor

 int proxTriggerFlag=0;

 int binIsFullTimeDelay=2000; // how many milliseconds the sensor stays active before reporting that the bag is full

/\*void motion\_track\_score(const char \*event, const char \*data){

 other\_sensor=millis();} //Used to track time of motion event from top PIR

\*/

// setup() runs once, when the device is first turned on.

void setup() {

 // Put initialization like pinMode and begin functions here.

 pinMode(ledPin, OUTPUT); // declare LED as output

 pinMode(buzzPin, OUTPUT); // declare sensor as input

 pinMode(proxPin, INPUT); // declare sensor as input

 //eventually, add a reset button here (define int above)

 //Particle.subscribe("Motion\_PIR\_Top",motion\_track\_score); //Subscribe to motion of top PIR

 Serial.begin(9600);

}

// loop() runs over and over again, as quickly as it can execute.

void loop() {

 // The core of your code will likely live here.

 val = analogRead(proxPin); // read input value

Particle.variable("val",&val,INT);

Particle.variable("this\_sensor",&this\_sensor,INT);

// blink new LED for 100 ms to indicate that sensor data was published

delay(100);

 if (val <= 2000) { // check if the input is HIGH

 //worksdigitalWrite(ledPin, HIGH); // turn LED ON

 //worksdelay(250); //led test only, remove later

 /\*

 // implement a wait to ensure that any falling clothes don't trigger "full"

 // just because they pass the sensor

 \*/

 //How many millis have passed?

 proxTriggerFlag=1;

 //Particle.variable("this\_sensor",&this\_sensor,INT);

 // Is this the first time that this\_sensor was triggered?

 if (this\_sensor==0){

 this\_sensor = millis(); //now we know when it is first active.

 }

 //first half means it won't trigger the first time thanks to previous loop

 //since it takes .000000001 seconds between these two lines, not 2 seconds

 if (abs(millis()-this\_sensor)>=binIsFullTimeDelay && this\_sensor!=0){

 //then it's not the first time that the sensor was triggered

 digitalWrite(ledPin, HIGH);

 delay(5000);

 // sound the alarm!

 // publish an event?

 Particle.publish("Laundry is Full!", PRIVATE); //Publish event (console)

 this\_sensor=0; //Used to reset timer for next proximity sensor trigger

 // wait 1s before retrieving data again (per assignment)

 delay(1000);

 }

 } else {

 proxTriggerFlag=0; //prevents two completely separate occurrences from

 // triggering the "laundry is full" loop

 digitalWrite(ledPin, LOW); // turn LED OFF

 //wait 1 second before retrieving data again per assignment requirement

 delay(1000);

 }

}