```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include "HX711.h"

#define pir 4
const int LOADCELL_DOUT_PIN = 2;
const int LOADCELL_SCK_PIN = 3;
float a;
float b=60;

HX711 scale;
//create an RF24 object
RF24 radio(9, 8);  // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

void setup()
{
  pinMode(pir,INPUT);
  radio.begin();

  //set the address
  radio.openWritingPipe(address);

  //Set module as transmitter
  radio.stopListening();

  Serial.begin(9600);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

  scale.set_scale(2280.f);
  scale.tare();
}
void loop()
{
  if (digitalRead(pir)==1){
  Serial.println("Motion detected");
  delay(1000);
}
```

```
  Serial.print("one reading:\t");
  Serial.print(scale.get_units(), 1);
  Serial.print("\t| average:\t");
  Serial.println(scale.get_units(10), 1);
  a = scale.get_units(10);

if (a>b && digitalRead(pir) == 1){
  Serial.print("Some one is on bed and moving\n");
  const char text[] = "ON bed moving"; //13 char
  radio.write(&text, sizeof(text));
  delay(1000);
}
else if (a<b && digitalRead(pir) == 1){
   Serial.print("Some one is NOT on bed and moving \n");
  const char text[] = "NOT bed moving"; //14 char
  radio.write(&text, sizeof(text));
  delay(1000);
}
else if (a>b && digitalRead(pir) !=1){
  Serial.print("Some one is on bed and not moving \n");
  const char text[] = "ON bed not moving"; //17 char
  radio.write(&text, sizeof(text));
  delay(1000);
}
else if(a<b && digitalRead(pir) !=1){
  Serial.print("Some one is NOT on bed and not moving \n");
  const char text[] = "NOT bed not moving"; // 18
  radio.write(&text, sizeof(text));
  delay(1000);
}


  scale.power_down();             // put the ADC in sleep mode
  delay(5000);
  scale.power_up();

}
```

DOOR BOARD

```
//Include Libraries
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//create an RF24 object
RF24 radio(9, 8);  // CE, CSN

//address through which two modules communicate.
const byte address[6] = "00001";

#define sensor1 5 //black In IR
```

```
#define sensor2 6 //white Out IR
#define alarm1 2
#define alarm2 3
#define sms 4
#define LED 7

char dir;

void setup()
{
  while (!Serial);
  Serial.begin(9600);
  radio.begin();

  //set the address
  radio.openReadingPipe(0, address);

  //Set module as receiver
  radio.startListening();

  pinMode(alarm1, OUTPUT);
  pinMode(alarm2, OUTPUT);
  pinMode(LED, OUTPUT);
  pinMode(sms, OUTPUT);
  pinMode(sensor1,INPUT_PULLUP);
  pinMode(sensor2,INPUT_PULLUP);

  dir='x';

  }

void loop(){

  digitalWrite(alarm1,HIGH);
  digitalWrite(alarm2,HIGH);
  digitalWrite(sms, HIGH);
  digitalWrite(LED, LOW);
  const char a[32]={"ON bed moving"}; //14
  const char b[32]={"NOT bed moving"}; //15
  const char c[32]={"ON bed not moving"}; //18
  const char d[32]={"NOT bed not moving"}; //19
  char text[32]={0};
  int w;
  //Read the ata if available in buffer
```

```
 if (radio.available())
 {
   radio.read(text, sizeof(text));
   //String mytext =String(text);
   Serial.println(text);
  //w= mytext.length();
 }


 if (digitalRead(sensor1)==0){//sensor 1 detects object

  if (dir=='x'){//if the last movement was unkown
   dir='o';//movement is out going
   //give 2 seconds before ignoring the direction
   Serial.println("direction is Out");
   delay(1000);

   if (dir =='o'&& (sizeof(text)== sizeof(b) || sizeof(text)== sizeof(c)) ){       //still not bed not
moving || not on bed while moving (18 || 14)
       digitalWrite(alarm1, LOW);
       digitalWrite(alarm2, LOW);
       digitalWrite(sms, LOW);
       delay(1000);

       }
     else if (dir =='o' && (sizeof(text)== sizeof(a) || sizeof(text)== sizeof(d))){    //still on bed not
moving && still on bed while moving (17 || 13)
       digitalWrite(LED, HIGH);
       delay(1000);

       }
       dir='x';
  }
 }

 if (digitalRead(sensor2)==0){//sensor 2 detects object

  if (dir=='x' && sizeof(text)==sizeof(a)){//if the last movement was unkown
   dir='i';//movement is out going
   //give 2 seconds before ignoring the direction
   Serial.println("direction is IN");
   dir='x';
   delay(1000);
   }
```

```
  }

}
```