

## Zenotroupe code

```
//This sketch demonstrates controlling a motor using either a
temperature sensor or a PIR sensor

#include <SparkFun_TB6612.h> // This library adds more commands for
the motor

//Motor 1
int pinAIN1 = 13; //Direction
int pinAIN2 = 12; //Direction
int pinPWMA = 11; //Speed

//Temperature Sensor
int pinTemp = A1; // Temperature sensor uses analog pin A1 on the
Arduino board

//PIR Sensor
int pinPIR = 2; // PIR sensor uses digital pin 2 on the Arduino board

//Standby for motor
int pinSTBY = 10; //Standby connection on the Arduino
Motor motor1 = Motor(pinAIN1, pinAIN2, pinPWMA, 1, pinSTBY);
//intializing the motor

void setup()
{
    Serial.begin(9600);
    pinMode(pinPIR, INPUT_PULLUP); //setup the PIR as an input
}

void loop()
{
    int sensorInput = analogRead(pinTemp);
    double temp;
    temp = (double)sensorInput / 1024; //find percentage of input
reading
    temp = temp * 5; //multiply by 5V to get voltage
    temp = temp - 0.5; //Subtract the offset
    temp = temp * 100; //Convert to degrees C

    // print the value to Serial Monitor
    // helpful for troubleshooting
    Serial.print("temperature: ");
    Serial.print(temp);
}
```

```

Serial.println(" C");

int proximity = digitalRead(pinPIR); //check for motion

if( proximity == LOW && temp < 27 )
{
    motor1.drive(30); // spin motor at a slower speed when motion
is detected
}
else if( temp >= 27 )
{
    motor1.drive(150); // spin motor at a faster speed when
temperature increases
}
else
{
    motor1.brake(); // when neither is true turn the motor off
}
}

```

### Color coding

- Comment lines are grey. They provide useful descriptions but do not alter the functionality of the code.
- Command lines which relate to the motor are red
- Command lines which relate to the temperature sensor are blue
- Command lines which relate to the PIR sensor are purple
- A serial monitor is used for troubleshooting; it provides the values measured and calculated. Those lines are green.
- Command lines used for decision making based on light levels are orange.
- Command lines which are black are essential for every Arduino sketch