.

Using Arduino Uno

You can use the light sensor to measure the ambient light level and use that information to control the LEDs. Specifically, you can set a threshold value for the ambient light level, and when the measured light level falls below that threshold, turn on the LEDs.

To control the LED light intensity based on the number of objects detected by the IR sensor, you can use the digital output of the IR sensor to increment a counter variable, and then use that variable to set the brightness of the LEDs. For example, you can set the brightness to be a percentage of the maximum brightness, where the percentage is determined by the value of the counter variable.

You will need to write code using the Arduino programming language to control the sensor and LEDs according to your desired behavior.

Using Arduino Uno and PIC

If you're using both an Arduino Uno and a PIC microcontroller, you'll need to use them in conjunction to accomplish your goals. The Arduino can be used to read the input from the light sensor and IR detector, and the PIC can be used to control the LEDs.

You can use the Arduino to measure the ambient light level using the light sensor and use that information to send a signal to the PIC. Specifically, you can set a threshold value for the ambient light level, and when the measured light level falls below that threshold, the Arduino sends a signal to the PIC to turn on the LEDs.

To control the LED light intensity based on the number of objects detected by the IR sensor, you can use the digital output of the IR sensor to increment a counter variable on the Arduino, and then use that variable to send a signal to the PIC to set the brightness of the LEDs. For example, you can set the brightness to be a percentage of the maximum brightness, where the percentage is determined by the value of the counter variable.

You will need to write code for both the Arduino and the PIC to control the sensor and LEDs according to your desired behavior. It is also important to establish communication between the two microcontroller, it can be done using UART, I2C or SPI protocols.

.

Code

I'd be happy to help you write the code for your project, but it's important to note that code will depend on the specific components you're using and the communication protocols you've established between the Arduino and PIC.

Here is an example of how you might write the code for the Arduino to read input from the light sensor and IR detector, and send signals to the PIC:

```
int lightSensorPin = A0;
int irSensorPin = 2;
int threshold = 500;
int irObjectCounter = 0;

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

void loop() {
  int lightLevel = analogRead(lightSensorPin);
  int irSensor = digitalRead(irSensorPin);
  if (lightLevel < threshold) {
    Serial.write("Lights On");
  }
  if (irSensor == HIGH) {
    irObjectCounter++;
    Serial.write(irObjectCounter);
  }
}
```

The example above reads the input from light sensor, compares it with the threshold value, if the light level is below the threshold, it sends "Lights On" signal via serial communication to the PIC. It also reads the input from IR sensor, if the sensor detects any object, it increments the counter variable and sends the value to PIC.

And here is an example of how you might write the code for the PIC to receive the signals from the Arduino and control the LEDs:

.

```
int ledPin = 13;
int ledBrightness = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    char incomingByte = Serial.read();
    if (incomingByte == "Lights On") {
      analogWrite(ledPin, ledBrightness);
    } else {
      ledBrightness = incomingByte;
      analogWrite(ledPin, ledBrightness);
    }
  }
}
```

The code above sets the LED pin as output, then it waits for incoming serial data, if the data is "Lights On" it turns on the LED with the brightness level set by Arduino, if the data is a number, it sets the led brightness with that value.
It is important to note that this is an example, and it may need to be adjusted depending on your specific components and the communication protocols you've established.