

Infra-Red Proximity Sensor Part 1

Posted on January 22, 2008, by Ibrahim KAMAL, in [Sensor & Measurement](#), tagged

Based on a simple basic Idea, this proximity sensor, is easy to build, easy to calibrate and still, it provides a detection range of 35 cm (range can change depending on the ambient light intensity).

This sensor can be used for most indoor applications where no important ambient light is present. For simplicity, this sensor doesn't provide ambient light immunity, but a more complicated, ambient light ignoring sensor should be discussed in a coming article. However, this sensor can be used to measure the speed of object moving at a very high speed, like in industry or in tachometers. In such applications, ambient light ignoring sensor, which rely on sending 40 Khz pulsed signals cannot be used because there are time gaps between the pulses where the sensor is 'blind'... The solution proposed doesn't contain any special components, like photo-diodes, photo-transistors, or IR receiver ICs, only a couple of IR leds, an Op amp, a transistor and a couple of resistors. In need, as the title says, a standard IR led is used for the purpose of detection. Due to that fact, the circuit is extremely simple, and any novice electronics hobbyist can easily understand and build it.

Object Detection using IR light

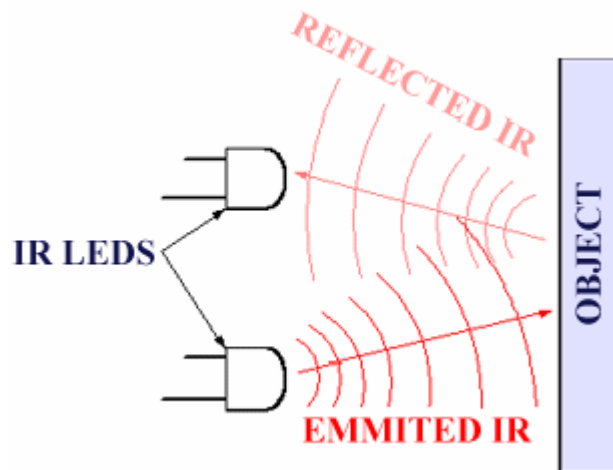


figure 1

It is the same principle in ALL Infra-Red proximity sensors. The basic idea is to send infra red light through IR-LEDs, which is then reflected by any object in front of the sensor (**figure 1**).

Then all you have to do is to pick-up the reflected IR light. **For detecting the reflected IR light, we are going to use a very original technique: we are going to use another IR-LED**, to detect the IR light that was emitted from another led of the exact same type! This is an electrical property of Light Emitting Diodes (LEDs) which is the fact that a led produce a voltage difference across its leads when it is subjected to light. As if it was a photo-cell, but with much lower output current.

In other words, the voltage generated by the leds can't be – in any way – used to generate electrical power from light, it can barely be detected. that's why as you will notice in the schematic, we are going to use a Op-Amp (operational Amplifier) to accurately detect very small voltage changes.

The electronic Circuit

Two different designs are proposed, each one of them is more suitable for different applications. The main difference between the two designs is the way infra-red (IR) light is sent on the object. The receiver part of the circuit is exactly the same in both designs. **Note: Both the sender and the receiver are constructed on the same board. They are separated in the schematics for simplification.**

Design 1: Low range, Always ON

As the name implies, the sensor is always ON, meaning that the IR led is constantly emitting light. This design of the circuit is suitable for **counting objects**, or **counting revolutions** of a rotating object, that may be of the order of 15,000 rpm or much more. However this design is more power consuming and is not optimized for high ranges. in this design, range can be from 1 to 10 cm, depending on the ambient light conditions.

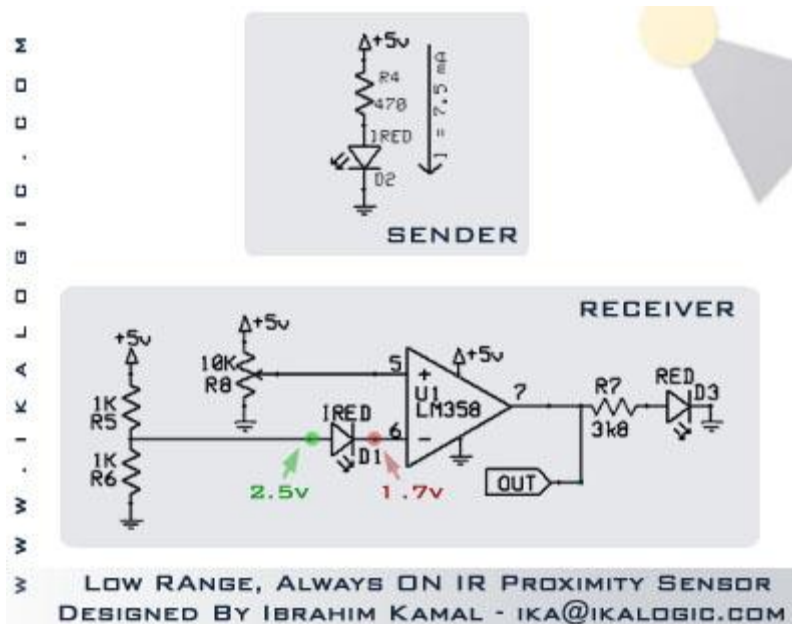


figure 2.A

As you can see the schematic (**figure 2.A**) is divided into two parts the **sender** and the **receiver**. The **sender** is composed of an IR LED (D₂) in series with a 470 Ohm resistor, yielding a forward current of **7.5 mA**.

The receiver part is more complicated, the two resistors R₅ and R₆ form a voltage divider which provides 2.5V at the anode of the IR LED (here, this led will be used as a sensor). When IR light falls on the LED (D₁), the voltage drop increases, the cathode's voltage of D₁ may go as low as 1.4V or more, depending

on the light intensity. This voltage drop can be detected using an Op-Amp (operational Amplifier **LM358**). You will have to adjust the variable resistor (POT.) R_8 so the the voltage at the positive input of the Op-Amp (pin No. 5) would be somewhere near 1.6 Volt. if you understand the functioning of Op-Amps, you will notice that the output will go High when the volt at the cathode of D_1 drops under 1.6 V. So the output will be High when IR light is detected, which is the purpose of the receiver.

In case you're not familiar with op-amps, here is shortly and in a very simplified manner, what you need to know to understand how this sensor functions: The op-amp has two input, the **+ve** input, and the **-ve** input. ***If the +ve input's voltage is higher than the -ve input's voltage, the output goes High (5v, given the supply voltage in the schematic), otherwise, if the +ve input's voltage is lower than the -ve input's voltage, then the output of the Op-Amp goes to Low (0V)***. It doesn't matter how big is the difference between the **+ve** and **-ve** inputs, even a 0.0001 volts difference will be detected, and the the output will swing to 0v or 5v according to which input has a higher voltage.

Some applications of the 'low range Always ON' Design:

Notice how in both devices, the IR leds are encapsulated to protect them from ambient light. This kind of encapsulation was totally sufficient to overcome all noise due to ambient light for indoor applications.

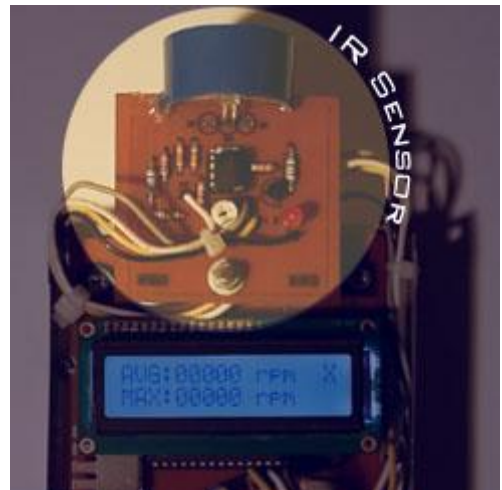
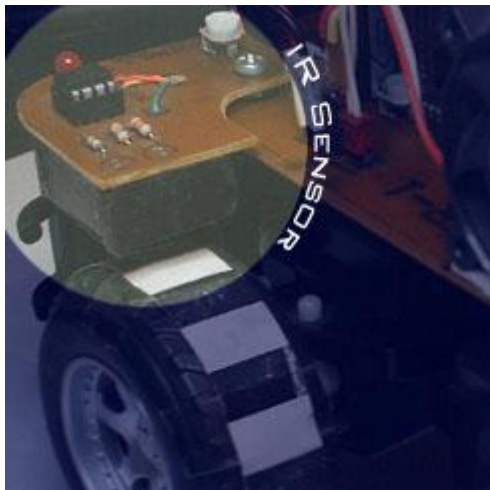
Wheel Encoder

This is a simple wheel encoder based on the idea that white stripes will reflect IR light, while black ones will absorb it. This will result in a series of electrical pulses as the wheel is rotating, providing the microcontroller with precious information that can be used to calculate displacement, velocity or even acceleration. It is now clear that this kind of sensor has to be Always ON, to detect every single white stripe passing in front of it, to achieve accurate results.

Contact-Less tachometer

This is a tachometer, that counts the revolutions per min a rotating object, given that the object has a reflective glued on it, that will pass in front of the IR sensor for each every revolution, giving a pulse per revolution. Ag microcontroller will have to be used to 'understand' the provided by the sensor and display it. Many comm contact-less tachometers, that are sold for more than \$20 on this simple idea!

[Build your own one for less than \$20 in [this article...](#)]



Design 2: High range, Pulsed IR

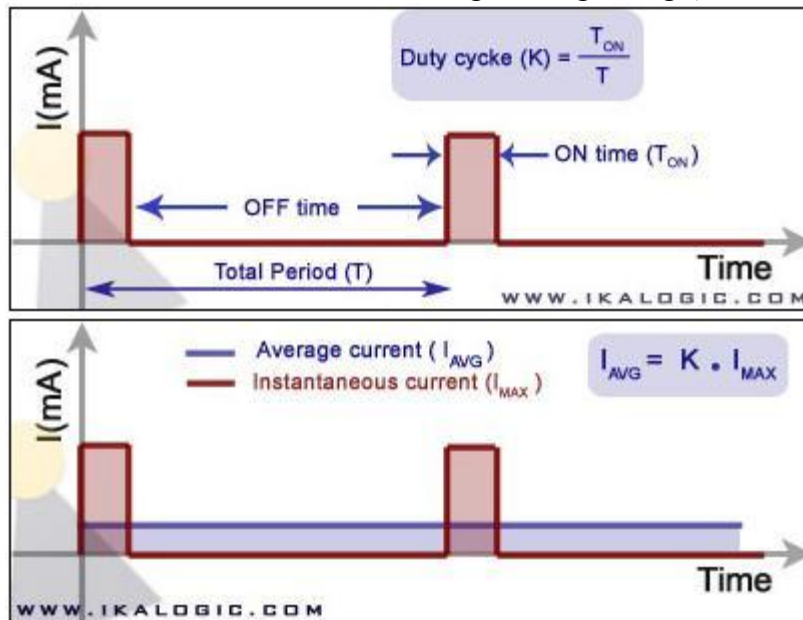


figure 3.A

In this design, which is oriented to obstacle detection in robots, our primary target is to reach high ranges, from 25 to 35 cm, depending on ambient light conditions. The range of the sensor is extended by increasing the current flowing in the led. This is a delicate task, as we need to send pulses of IR instead of constant IR emission. The duty cycle of the pulses turning the LED ON and OFF have to be calculated with precision, so that the average current flowing into the LED never exceeds the LED's maximum DC current (or 10mA as a standard safe value). The duty cycle is the ratio between the ON duration of the pulse and the total period. A low duty cycle will enable us to inject in the LED high instantaneous currents while shutting it OFF for enough time to cool down from the previous cycle.

Those two graphs (**figure 3.A**) shows the meaning of the duty cycle, and the mathematical relations between the ON time, the Total period, and the average current.

In the second graph, the average current in blue is exaggerated to be visible, but real calculations would yield a much smaller average current.

Now, hands on the circuit that will put all this theory into practice. The CTRL input in the figure, stands for **Control**, and this pin should be connected to the source of the low duty cycle pulses discussed above, whether it is a microcontroller or an LM555 timer that generates the pulses.

The calculations yielded that a 10 ohm resistor is series with the LED D_2 , would cause a current of approximately 250 mA to flow through the LED. A current this high, would destroy the LED if applied for a long period of time (some dozens of seconds), this is why we have to send low duty cycle pulses.

The first Op-amp, the **sender part (figure 3.B)**, will provide voltage buffer, to enable any kind of device to control the sensor, also, it will provide the 30 mA base current required to drive the base of the transistor. The calculation of the the base resistor R_3 depends on the type of transistor you use, thus on how much current you need on the base to drive the required collector current. The **receiver part (figure 3.B)** of this schematic functions in the exact same way as in the first design, refer to the first, 'ALLWAYS ON' design for a detailed description.

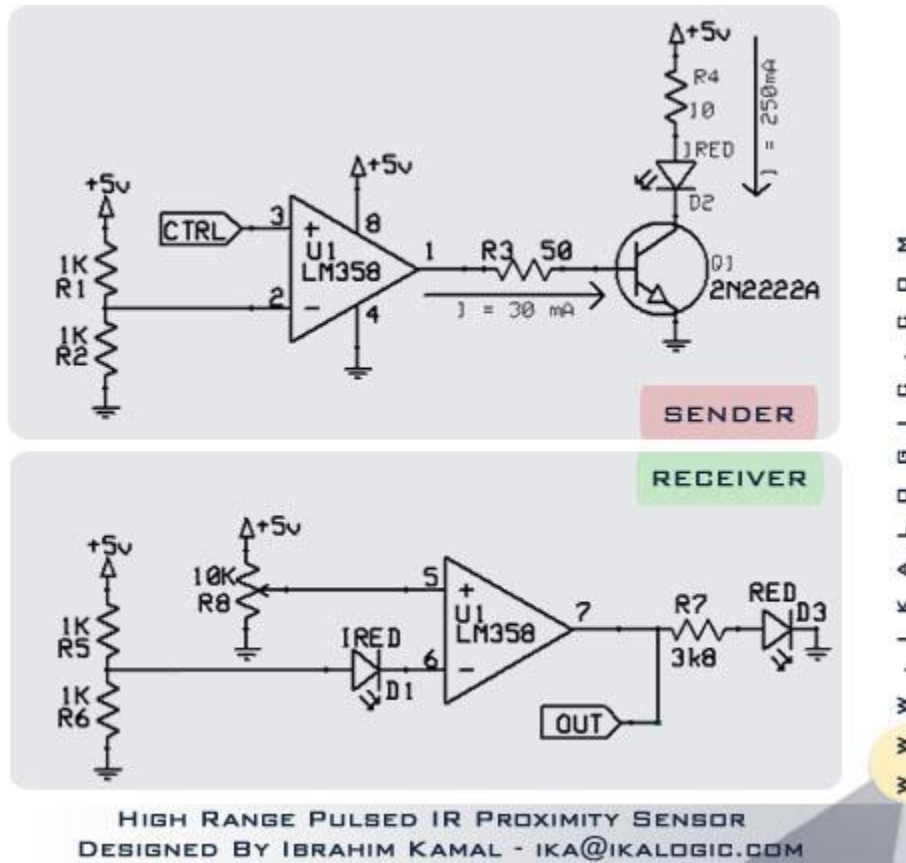


figure 3.B

Software based ambient light detection

When the sensor is controlled by a microcontroller to generate the low duty cycle pulses, you can benefit from the High and Low pulses to be able to detect any **false readings** due to ambient light. This is done by recording two different outputs of the sensor, one of them during the ON pulse (the sensor is emitting infra red light) and the other during the OFF time and compare the results.

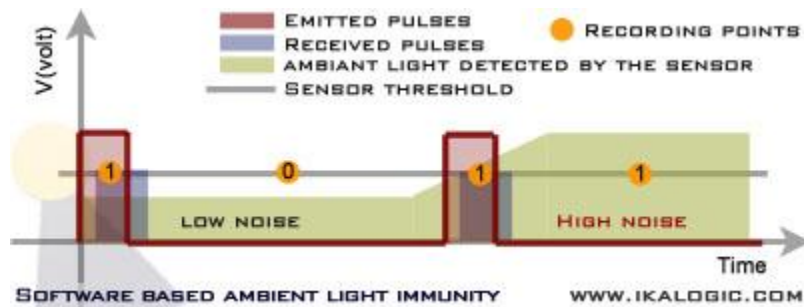


figure 4

The Idea is enlightened by this graph, where in the first period (**figure 4**), there is low ambient noise, so the microcontroller records a "1" during the on cycle, meaning that an object reflected the emitted IR Light, and then the microcontroller records a "0" meaning that during the OFF time, it didn't receive anything, which is logic because the emitter LED was OFF. But study the second period of the graph (**figure 4**), where the sensor is put in a high ambient light environment. As you can see, the the microcontroller records "1" in both conditions (OFF or ON). This means that we can't be sure whether the sensor reception was caused by an object that reflected the sent IR light, or it is simply receiving too much ambient light, and is giving "1" whether there is an obstacle or not.

The following table show the possible outcomes of this method.

Output recorded during:		Software based deduction
On pluse	Off time	
1	0	There is definitely an Obstacle in front of the sensor
1	1	The sensor is saturated by ambient light, thus we can't know if there is an obstacle
0	0	There is definitely Nothing in front of the sensor, the way is clear
0	1	This reading is un logical, there is something wrong with the sensor.

Example C Code for 8051 microcontrollers

```
#include <REGX51.h>

#include <math.h>

unsigned char ir;          // to store the final result
bit ir1,ir2;             // the 2 recording point required for our algorithm

delay(y) {               // simple delay function
    unsigned int i;
```

```

    for(i=0;i<y;i++){;}
}

void main(){
//P2.0 IR control pin going to the sensor
//P2.1 IR output pin coming from the sensor

while(1){
    P2_0 = 1;          //send IR
    delay(20);
    ir1 = P2_1;
    P2_0 = 0;          //stop IR
    delay(98);
    ir2 = P2_1;

    if ((ir1 == 1)&(ir2 == 0)){
        ir = 1;          // Obstacle detected
        P2_3 = 1;          // Pin 3 of PORT 2 will go HIGH turning ON a LED.
    if ((ir1 == 1)&(ir2 == 1)){
        ir = 2;          // Sensor is saturated by ambient light
    }else{
        ir = 0;          // The way is clear in front of the sensor.
    }
}
}
}

```

Components positioning

The correct positioning of the sender LED, the receiver LED with regard to each other and to the Op-Amp can also increase the performance of the sensor. First, we need to adjust the position of the sender LED with respect to the receiver LED, in such a way they are as near as possible to each others , while preventing any IR light to be picked up by the receiver LED **before it hit and object** and returns back. The

easiest way to do that is to put the sender(s) LED(s) from one side of the PCB, and the receiver LED from the other side, as shown in the 3D model below.

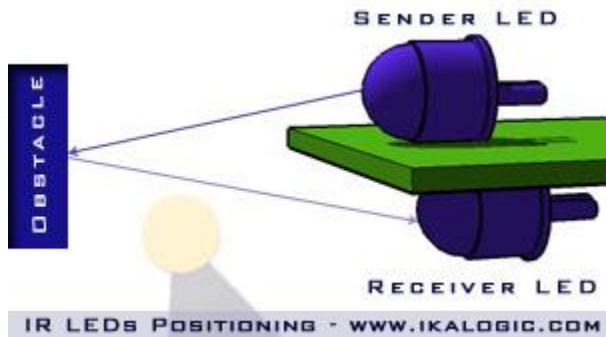


figure 5.A

This 3D model (**figure 5.A**) shows the position of the LEDs. **The green plate** is the PCB holding the electronic components of the sensor. you can notice that the receiver LED is positioned under the PCB, this way, there wont be ambient light falling directly on it, as ambient light usually comes from the top. It is also clear that this way of positioning the LEDs prevent the emitted IR light to be detected before hitting an eventual obstacle.

Another important issue about components positioning, is the distance between the receiver LED and the Op-Amp. which should be as small as possible. Generally speaking, the length of wires or PCB tracks before an amplifier should be reduced, otherwise, the amplifier will amplify – along with the original signal – a lot of noise picked up form the electromagnetic waves traveling the surrounding.

Here is an example PCB (**figure 5.B**) where the distance between the LED and the Op-Amp is shown. Sure this distance is not as critical as you may think, it can be up to 35mm without causing serious problems, but trying to reduce this distance will Always give you better results.

Actually, when I design the PCB, I start by placing the receiver LED and the Op-Amp, as near to each others as possible, then continue the rest of the design.

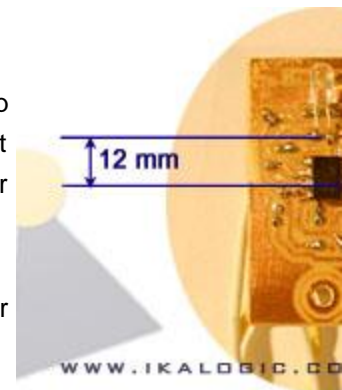


figure 5.B

An example PCB construction

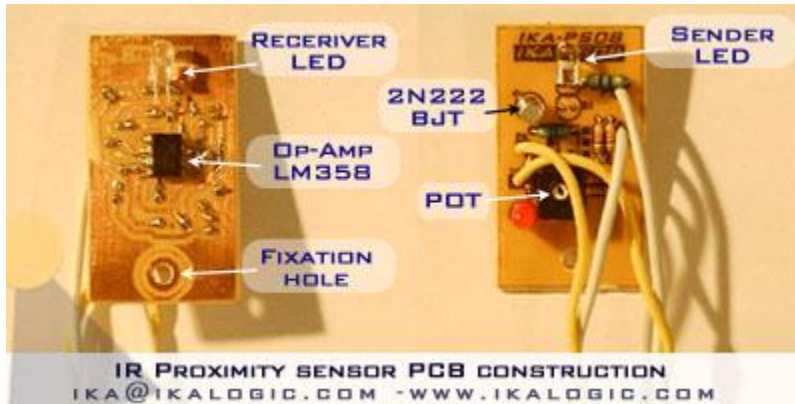


figure 6

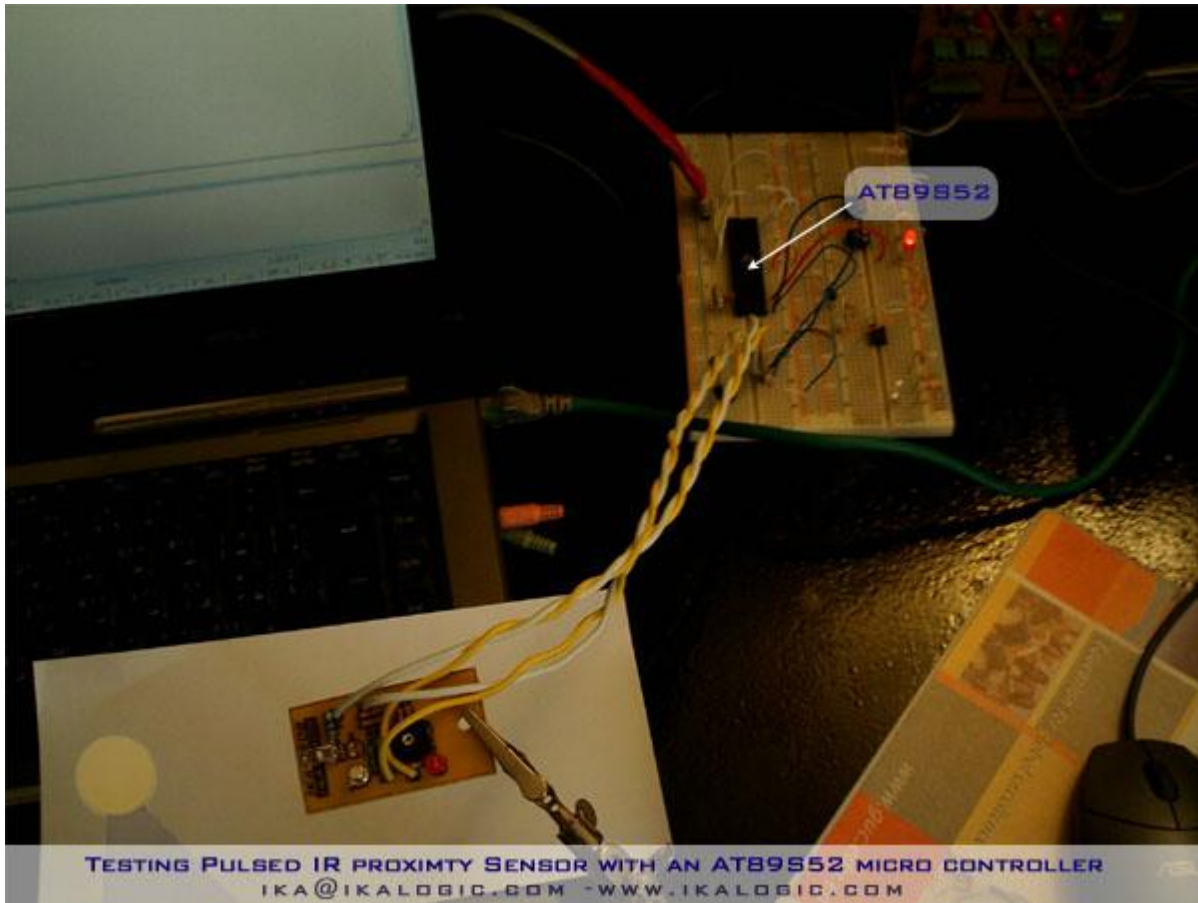
Here is an example construction of the PCB (*figure 6*) for the High Range, Pulsed IR proximity sensor. You can download here the project folder containing the schematic, the PCB design, and an example code for the microcontroller to send the low duty cycle pulses. In this design, the LM358 Op-Amp is mounted on the copper side to save some space. The POT is the potentiometer used to adjust sensitivity.

As explained before, the sender and receiver LEDs are placed on both sides of the PCB.

Testing the High range Pulsed IR sensor

The last step, is to test the performance of the pulsed IR proximity sensor. To do this, I connected the sensor to a 89S52 microcontroller, loaded with a program to generate pulses with a duty cycle of approximately 1.6% at a frequency of 3Khz. LEDs are designed to operate at very high frequencies, so you don't have to worry about the response time. To make sure your duty cycle calculations are correct, let the sensor running for a minute, and check with your fingers the temperature of the IR sender LED. If it's not hot, then everything is alright. On the other hand, if the LED is getting hot, to an extent that you can feel it, there is probably something wrong, you should then try to decrease the duty cycle, or increase the series resistor, in order to decrease the average current flowing into the LED.

Then, you can start testing the range of the sensor, and experiment it in different ambient light conditions, but the potentiometer may have to be adjusted carefully, to cope with ambient light.



In the example C code above, the final output of the sensor appears on the pin P2_3 of the microcontroller, as explained before.

Source: <http://www.ikalogic.com/infra-red-proximity-sensor-part-1/>