Jayhawk Engineering Company
3005 Eaton Hall, Lawrence, Kansas 66045

September 2024

FROM: Conner Glazner, Evan Day, Max Djafarov, Vincent Dick

TO: JECO Director of Engineering

SUBJECT: Documenting of IR Transmitter and Receiver Design

1. Statement of Purpose

Our team's goal for this project was to produce an inexpensive wireless communication system that would be suitable for an industrial environment. To accomplish this, our team designed an IR transmitting and receiving system with accompanying software for sending and receiving messages. We've designed this system because there's a demand for a wireless communication system in industrial sectors of the market. It's possible to develop such a system with readily available, cheap parts, and with a simple circuit design, so it's easy to transition into mass production. We first start with understanding the theory behind how the system should work, then design the circuits for the system, construct initial prototypes for testing and to start software development, and finally construct the final prototype with finalized software.

2. Theory and Design

2.1. Theory

We have two major circuits in our design, the transmitting circuit and the receiving circuit. For the transmitting circuit we have three major parts: the Arduino to send the data bitstream, an op-amp to power the IR LEDs, and a bipolar junction transistor (BJT) to control when the LEDs are powered. Originally, we planned to use the power from the Arduino to drive the LED but to improve the transmitting signal strength we used three LEDs in series, which required more voltage, so we accomplished this through the usage of a non-inverting op-amp circuit. To calculate the voltage gain we wanted, we used the following equations for non-inverting op-amp circuits:

$$V_s = V_e\left(1 + \frac{R_1}{R_2}\right) \tag{1}$$

$$\frac{R_1}{R_2} = \frac{V_s}{V_e} - 1 \tag{2}$$

In equation (1), we want V_s to be 12 volts, and V_e is known to be 5 volts, so we rearrange the equation to match equation (2), where we can now set the gain of the op-amp with resistor values. To transmit data, we turn our data into a bitstream which can then be sent through one of the Arduino Nano's data pins and used to switch the BJT. When the BJT isn't powered, the LEDs are grounded, which represents low voltage or a zero in the bitstream, and when the BJT is powered, the LEDs are powered by the op-amp circuit, which represents high voltage or a one in the bitstream. For the voltage switching part of the circuit, we used the following two equations to find our resistor values:

$$R_1 = \beta\left(\frac{V_{max} - 0.7}{I_F}\right) \tag{3}$$

$$R_2 < \frac{V_{DD}}{I_F} \tag{4}$$

For the receiving circuit there's only two major parts, the phototransistor and the Arduino. The phototransistor acts like a transistor that allows current through the circuit when exposed to IR light. The circuit is set up so that when light hits the phototransistor, we read voltage at one of the Arduino's analog input pins. For the phototransistor circuit, we wanted to use a mode known as switch mode, which can be designed based off the following equations:

$$V_{DD} < R_L I_C \tag{5}$$

$$R_L > \frac{V_{DD}}{I_C} \tag{6}$$

2.2. Design

One of the main constraints of this design was that we had to choose parts that were readily available and that we were to use infrared light for wireless communication. For required specifications, our circuit was to be designed to transmit at a baud rate of 1500 bits per second, and to transmit at a distance of at least one meter. For our designs, we started first with a circuit diagram for both the transmitter and receiver. In the transmitting circuit, we have two main parts of the circuit, the LED driver and the non-inverting op-amp. For the LED driver, we used two resistors, the 2N2222 BJT, and three TSAL6100 LEDs. Using equations (3) and (4) from the above theory section we can calculate the resistors needed for this part of the circuit:

$$R_1 = 100\left(\frac{5V - 0.7}{100mA}\right) = 4300\Omega \tag{7}$$

$$R_2 < \frac{5V}{100mA} = 50\Omega \tag{8}$$

When we originally designed this circuit our V_{DD} was set for a single LED, but this changed to 12V since we added more LEDs in series. Because the resistor value we chose was already below the threshold from the equation we didn't need to change our resistor. For the op-amp part of the transmitting circuit we wanted to boost the 5V output of the Arduino Nano up to 12V to power the LEDs, so we found our gain using equation (2) from the above theory section:

$$\frac{R_1}{R_2} = \frac{12V}{5V} - 1 = 1.4 \tag{9}$$

With this calculated ratio we can pick any two resistors that match that value, so we went with R_1 being 1.5kΩ and R_2 being 1kΩ since these resistor values are very common.

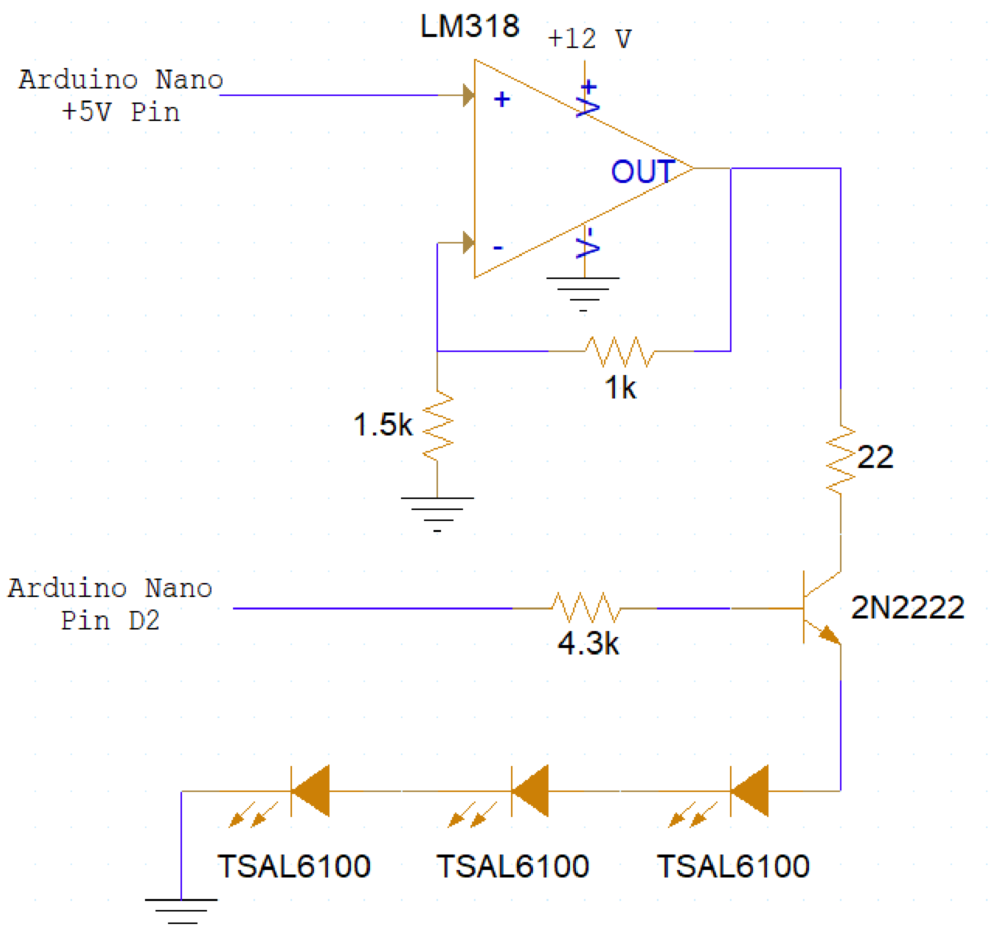

Figure 1 – Transmitter Circuit Diagram

For our receiver circuit we only have three components, the Arduino, a phototransistor and a resistor. We only really need to calculate the resistor value in equation (6) for this part of the circuit since we want to be in the switch mode:

$$R_L > \frac{5V}{29mA} = 173\Omega \tag{10}$$

Because our resistor value needs to be greater than 173Ω, we rounded up to 200Ω since this is a common resistor value.

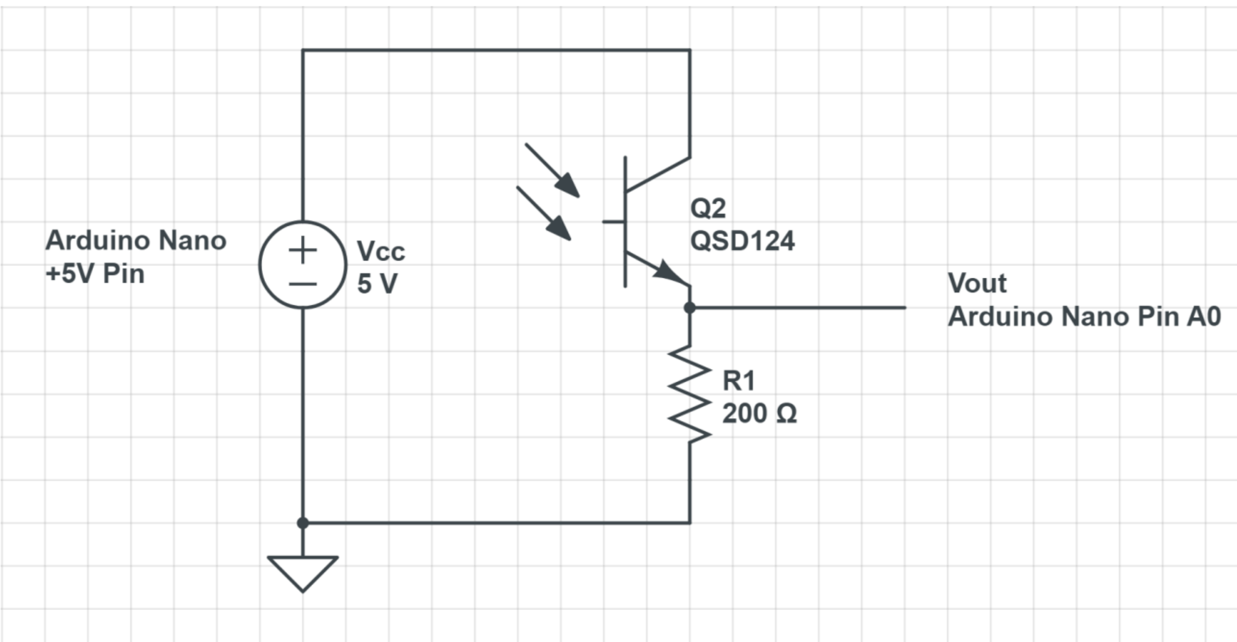

Figure 2 – Receiver Circuit Diagram

3. Implementation

3.1. Hardware

First, we constructed our circuits using all the components needed and a breadboard for the transmitter and receiver. With this breadboard we began coding rudimentary software to see if the circuits were working as we expected. In our original design we only had one LED and no op-amp in the transmitter circuit. When we learned that our signal strength wasn't very strong, we added 2 more LEDs and the op-amp to increase the voltage available to the LEDs. After these changes and some more software testing, we got to a point where we believed our circuits were ready to be turned into a prototype. Using vector boards, we recreated the circuits we had created on the breadboards and soldered the parts in place. For the transmitter circuit we have an Arduino data pin controlling the switching on the BJT, with the op-amp supplying the voltage necessary to power all the LEDs in series. Using software, we can transmit a message by converting data into a bitstream, which can be represented by high and low voltages that can then be sent via strobing the LEDs.

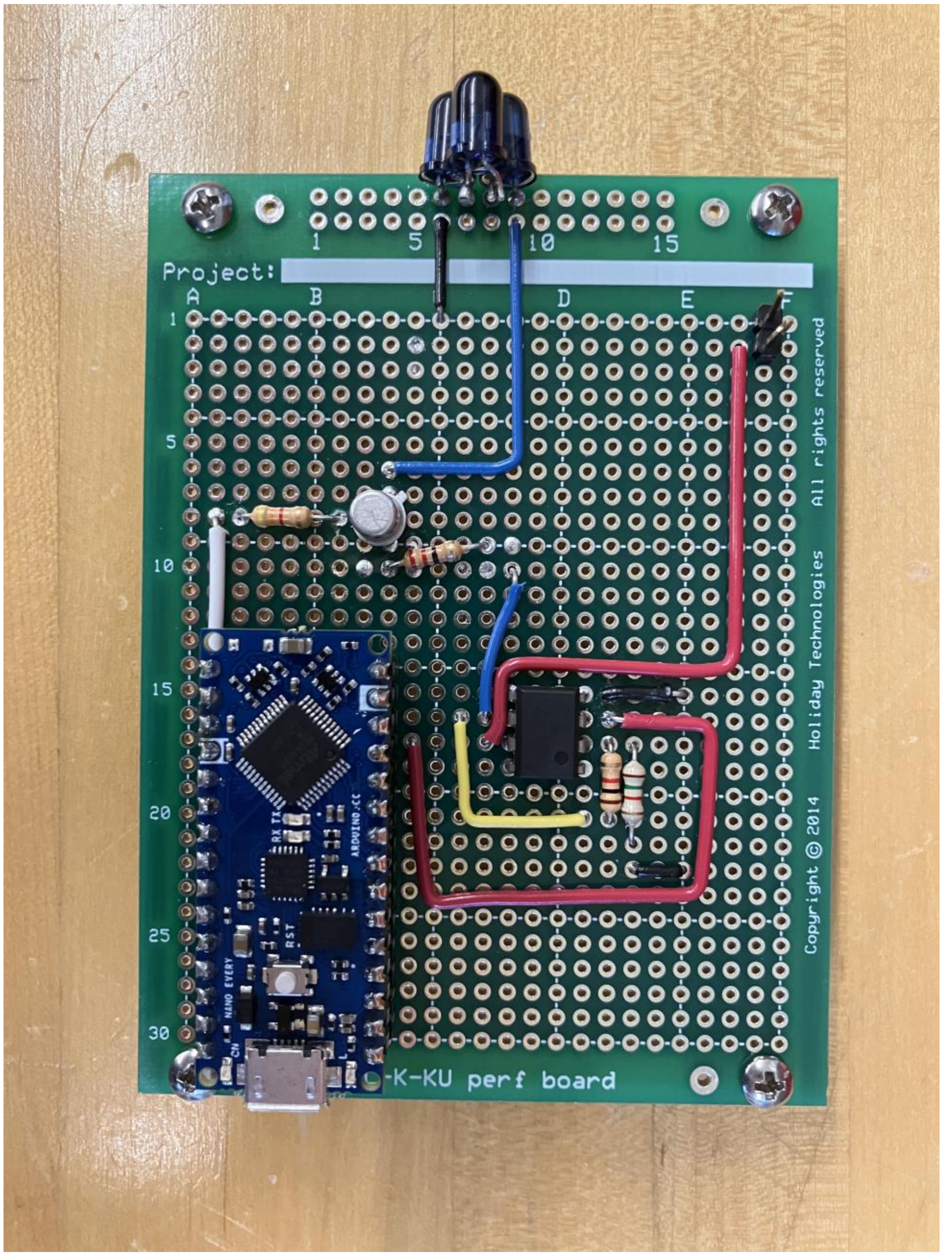

Figure 3 – Photo of Transmitter Prototype

For the receiver we used the Arduino power pin to send current into the circuit, and an output pin after the phototransistor that goes into an analog data pin on the Arduino. When IR light hits the phototransistor, the circuit can be completed, and the Arduino can read the voltage through the analog pin. Using this data input, we can read the transmitted bitstream and decode the message through software.

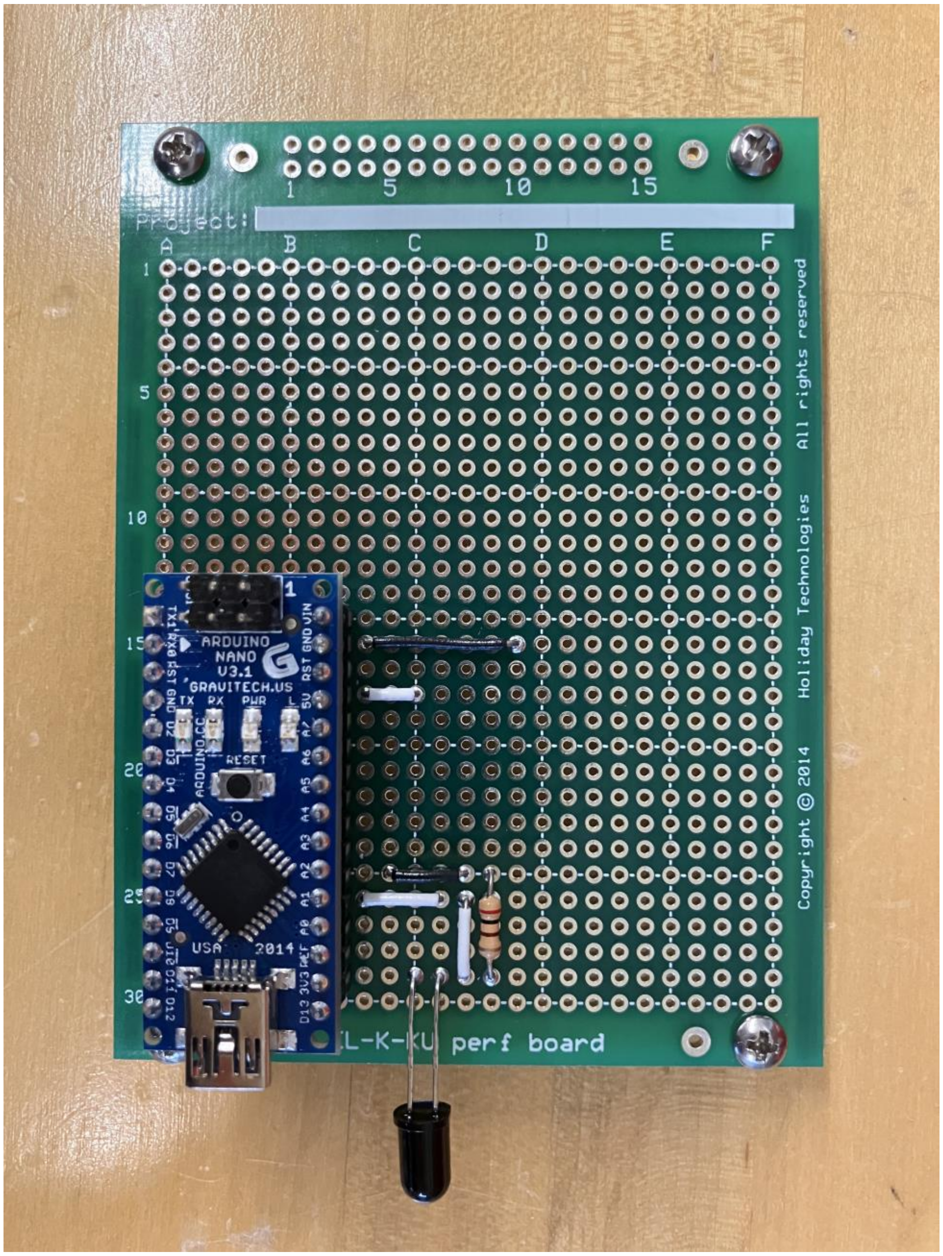

Figure 4 – Photo of Receiver Prototype

3.2. Software

The software for the transmitter circuit relies on a function serialByte(), which is called repeatedly and allows any message to be typed in the serial monitor and sent through the IR LEDs. When nothing is being transmitted, the function has a default low state. If a message is sent through the serial monitor, a high start bit is sent in order to alert the receiver circuit of the incoming message. Each ASCII character is converted into its 8-bit binary equivalent, and the IR LED is flashed on and off accordingly. The timing of each bit is calculated for a baud rate of 1500 and is managed by delayMicrosecondsCustom() function, which ensures that each bit is sent for approximately 667 microseconds. A parity bit is also calculated and added at the end of each byte for error detection. Before each character is sent, it initializes evenParity variable to 0. While it iterates through each bit, it checks whether each bit is ‘1’ using bitRead(). If bitRead is ‘1’ it increments the evenParity counter. Then it checks whether the evenParity count is even or odd and sets the parityBit. “*parityBit = (evenParity % 2 == 0) ? LOW: HIGH;”*. After sending the data bits and the parity bit, a stop bit is transmitted to show the completion of the message, so that the receiver gets ready for the next incoming message. The stop bit is characterized by LOW state when the LED is off, and it is held for a longer duration (10milliseconds) to ensure the receiver has enough time to process the incoming bits. Additionally, “bit rate” command was added to the program which acts as a trigger to execute a function to measure the time taken for messages to execute. When entered in both the transmitter and receiver’s Serial Monitors, this command initiates a process that counts the transitions between the HIGH and LOW states. It was added to provide real-time feedback and to help with transmission error issues.

The software for the receiver circuit relies on continuously monitoring the incoming IR signal. It utilizes an analog pin (A0) to detect incoming bits, which are stored in a FIFO buffer for processing. The code first waits for a start bit (a HIGH signal) and then reads a complete message consisting of 8 data bits, 1 parity bit, and 1 stop bit. After reading the bits, it verifies the message for transmission errors and converts the data into an ASCII character if valid. Additionally, the code includes a functionality to calculate the bit rate upon receiving the command "bit rate" via the Serial Monitor. It counts transitions and errors over a 1-second interval, reporting the total transitions and the Bit Error Rate (BER). The implementation features custom timing functions to ensure precise bit duration handling.

4. Laboratory Circuit Evaluation

4.1. Configuration

For the transmitter circuit, we used an Arduino Nano as a central controller for generating and modulating the infrared signal. The Arduino Nano was then connected to the op-amp (LM318) to amplify the signal and configured it as a non-inverting op-amp. The op-amp included a 1.5kΩ input resistor and 1kΩ as a feedback resistor. The output of the op-amp was then connected in series with the 22Ω resistor, which then connected to the collector side of transistor(2N2222), which served as a switch to drive the IR LEDs. Three IR LEDs (TSAL6100) were connected in series between the collector of the transistor and the ground. The combined output of three IR LEDs in series greatly enhances the effective range of the infrared signals. The circuit was powered by a power supply with 12V input connected into the positive terminal and the ground to the negative terminal of the op-amp. The 5V pin of the Arduino Nano was then connected to the non-inverting input of the op-amp. Additionally, 4.3kΩ resistor was connected between the Pin D2 of the Arduino Nano and the base of the transistor.

For the receiver circuit another Arduino Nano was used to process the incoming infrared signals. The QSD124 phototransistor was connected to the 5V pin of the Arduino Nano to convert the incoming light into a signal. The cathode side of phototransistor was then linked to a 200Ω resistor, which limited the flow of current into the Arduino's analog pin A0. The ground connection was configured between the Arduino and the phototransistor.

4.2. Procedure

The transmitter and receiver were placed approximately one meter apart for testing. Then they were aligned so that the IR LED directly faced the phototransistor on the receiver circuit. Both circuits were then connected to a computer to send and receive data through the Arduino serial ports. Additionally, the transmitter circuit was connected to a 12 volt power supply. Data was then imported through the transmitter serial port.

5. Evaluation of Test Data

5.1. Data

The transmission distance between the transmitter and receiver circuits was just slightly over one meter. After this distance the bit error rate increased dramatically, and data could no longer be reliably sent anymore. The transmission baud rate was ~1450 bits per second, and the receiving baud rate was ~1350 bits per second. The bit error rate within the one-meter transmission distance was ~5%.

5.2. Interpretation

This transmission distance is acceptable for the purposes of this project. However, a higher transmission distance could be achieved using more IR LEDs in series and running 100 or more milliamps through them. This would require an alternative transmitter circuit, as an op amp cannot supply more than 25 milliamps. Additionally, more phototransistors could be used in parallel on the receiver circuit, increasing the angle of transmission.

A 1500 baud rate was aimed for in this project. As a result, the delay between bits was calculated to be 667 microseconds. The reason for the baud rate being slightly lower in practice is the result of the transmitter and receiver circuits each having their own delay when processing the signals. This is due to both the time it takes for the code to run on both of the Arduinos and the time it takes for the hardware to send and receive the signals.

5.3. Conclusions

In conclusion, our wireless infrared communication system was successful and demonstrated its ability to transmit and receive data over about a meter. The integration of the op-amp (LM318) on the transmitter side helped to achieve a higher output voltage which resulted in having three IR LEDs in series. This configuration significantly increased the signal's strength, helping achieve the range that we got. While achieving a little over a meter distance was satisfactory for this project, the signal started to diminish significantly beyond that range. Anything further than that resulted in all kinds of issues with transmission. One improvement that could have been made is to enhance the strength of the signal by utilizing more IR LEDs and achieving higher current values.

6. References

[1] Texas Instruments, “High-Performance Operational Amplifiers datasheet”, SLOS063B datasheet, June 1976 [Revised Dec. 2002]

[2] Vishay, “High Power Infrared Emitting Diode, 940 nm, GaAlAs, MQW”, 81009 datasheet, Mar. 2014

[3] onsemi, “Plastic Silicon Infrared Phototransistor”, QSD123/D datasheet, Aug. 2023

[4] onsemi, “Amplifier Transistors”, P2N2222A/D datasheet, Jan. 2013

[5] Arduino, “Arduino Nano Pinout”, A000005 pinout datasheet, June 2021

[6] Electronics Notes. *Phototransistor Applications & Circuit Configurations.* [Online]. Available: https://www.electronics-notes.com/articles/electronic_components/transistor/phototransistor-circuits-applications.php