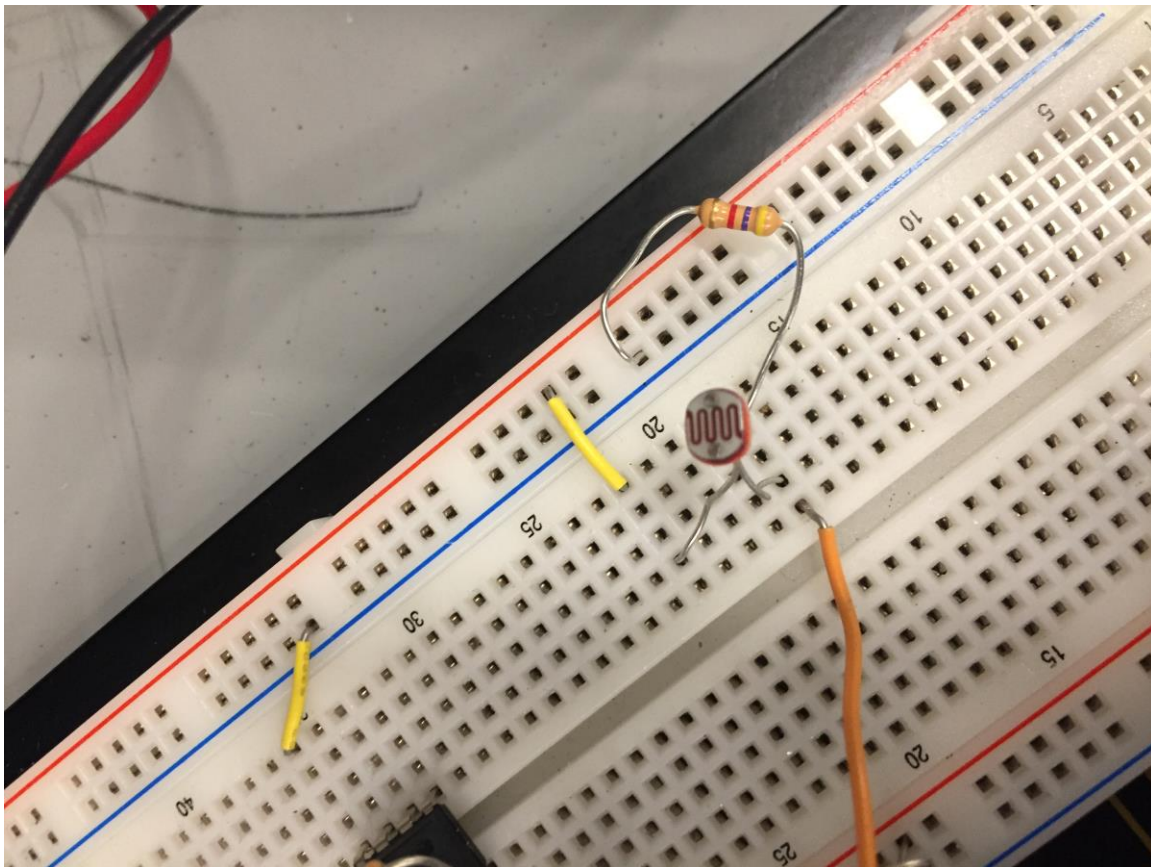
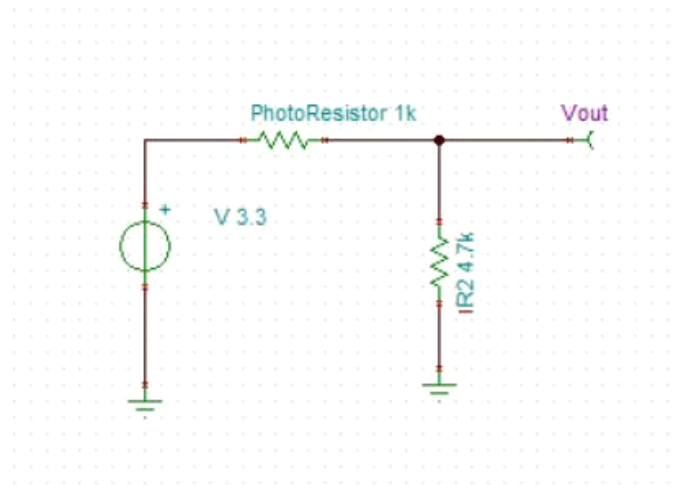


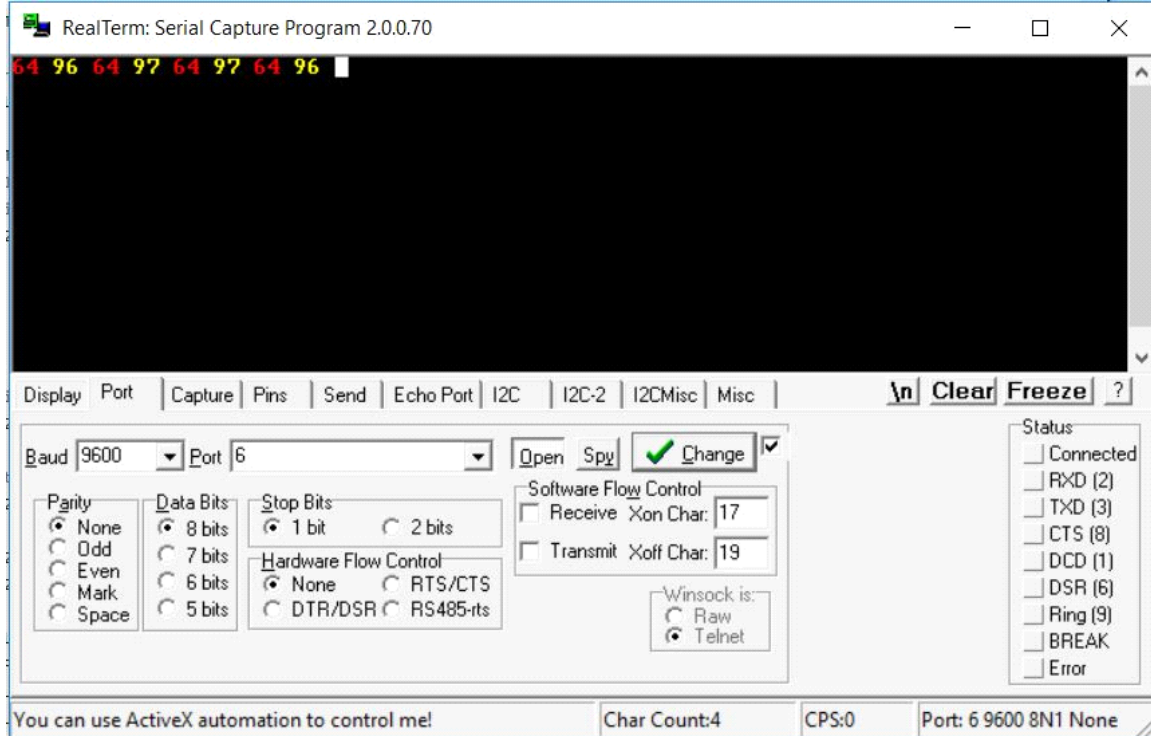
# Photoresistor

The first component tested was a photoresistor. A photoresistor is a light dependent resistor, in other words, it is a light controlled variable resistor. As incident light intensity intensifies, the resistance of the photoresistor decreases. This means that as incident light intensifies, the recorded output voltage should be higher. Below is a schematic of the built circuit

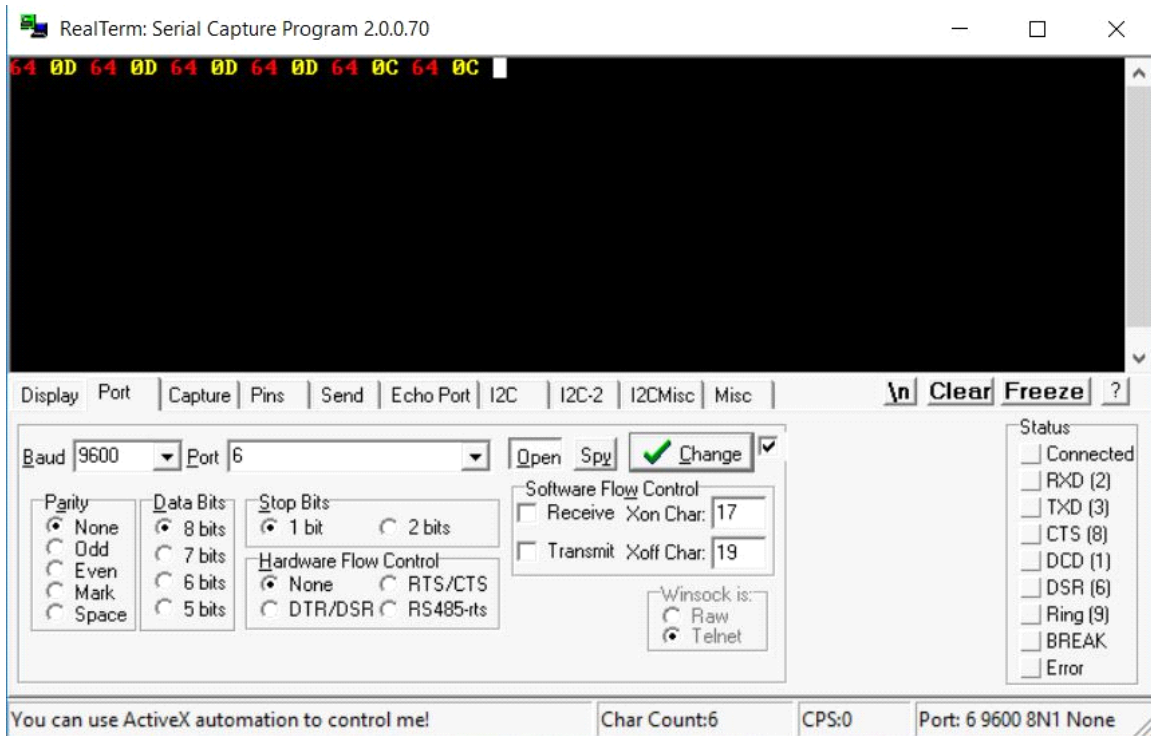


The 4.7k resistor is placed there to act as a voltage divider so that, even when the resistance of the photoresistor is low, the entire 3.3V from the source is not present at Vout.

### Light On:



### Light Off:

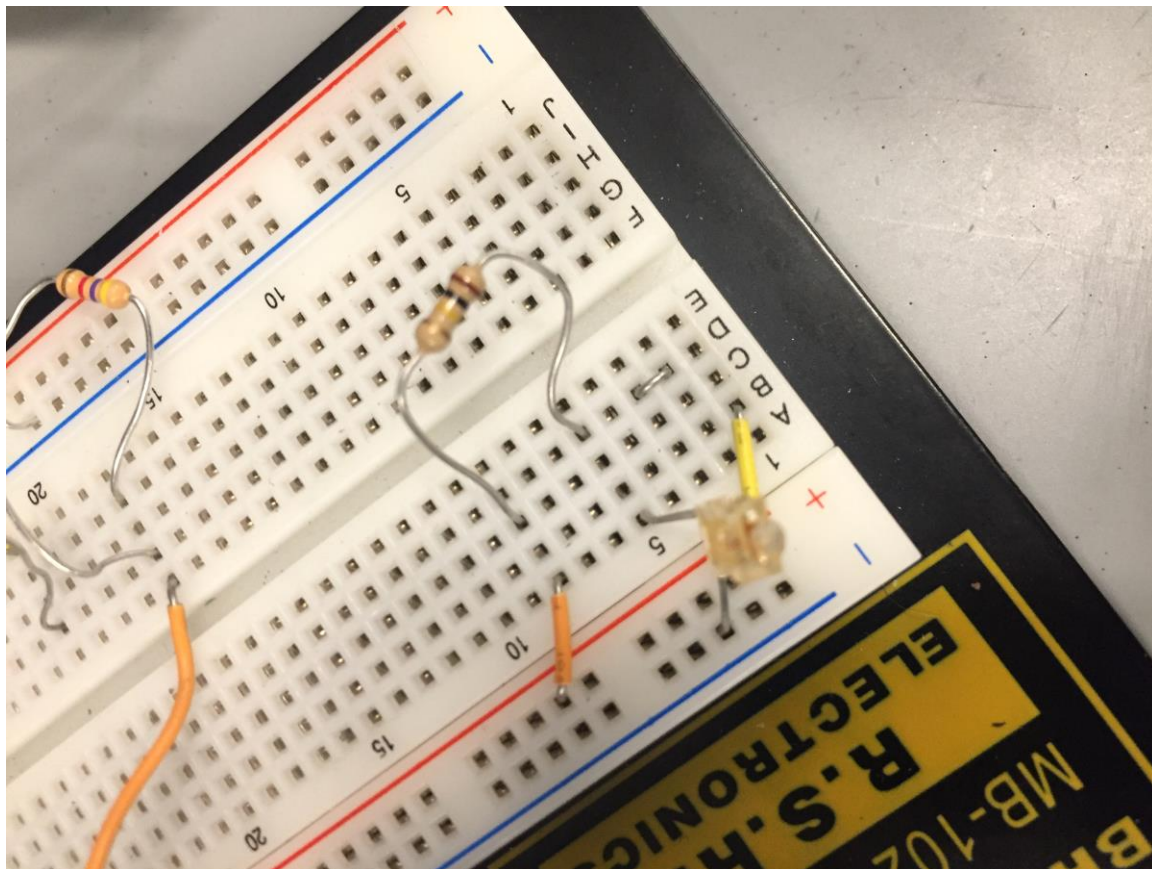
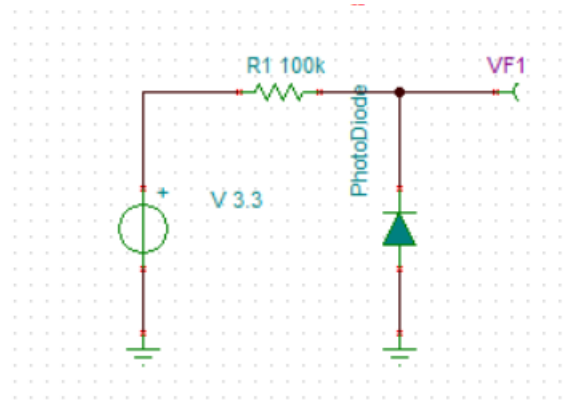


When the light is turned off the output voltage drops from 96 (150 in decimal) to 0D (13 in decimal). This shows that the phototransistor circuit is working properly and the output voltage is successfully being sent and received by UART.

## Photodiode

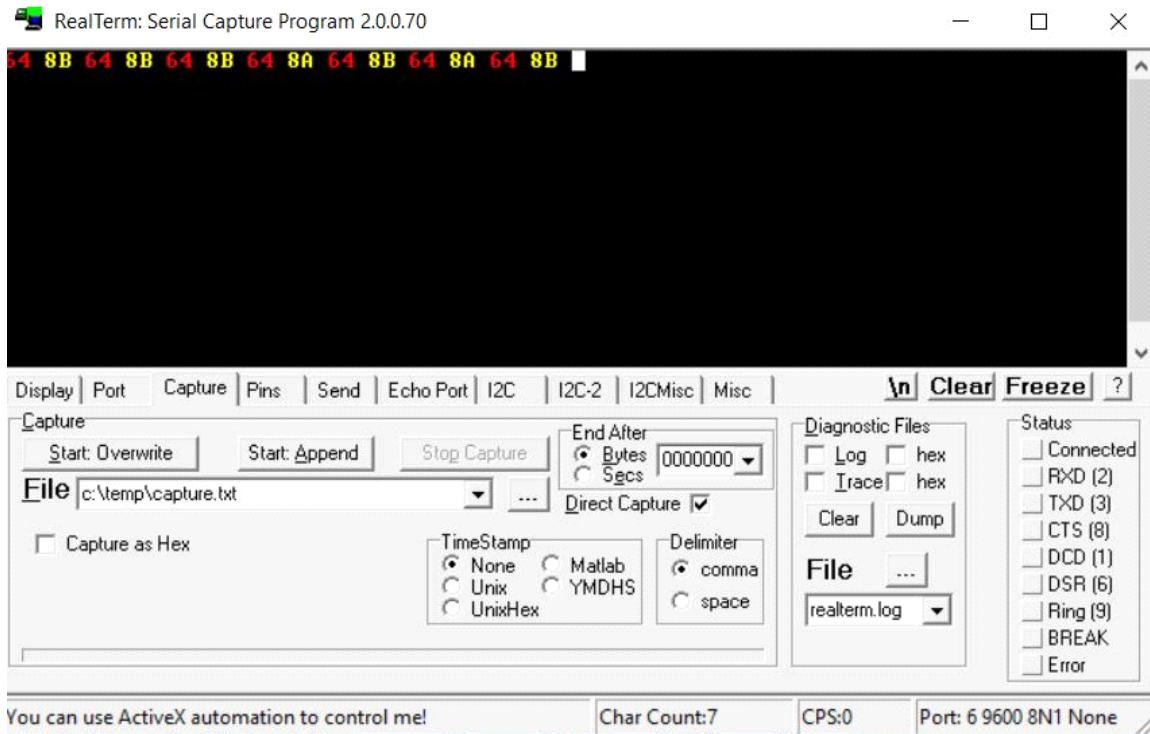
A photodiode is a semiconductor device that converts photons into an electrical current. A photodiode is a p-n junction diode. When a photon of sufficient light enters the diode, 1.1eV (the bandgap of silicon), an electron-hole pair is created. This process is also called the photoelectric effect.

As incident light intensifies, current is able to run through the diode. Because of the configuration of the circuit, as seen below, this means that as incident light intensifies, current will be able to run to ground, so the output voltage will decrease. Alternatively, as incident light decreases, the output voltage will be greater.

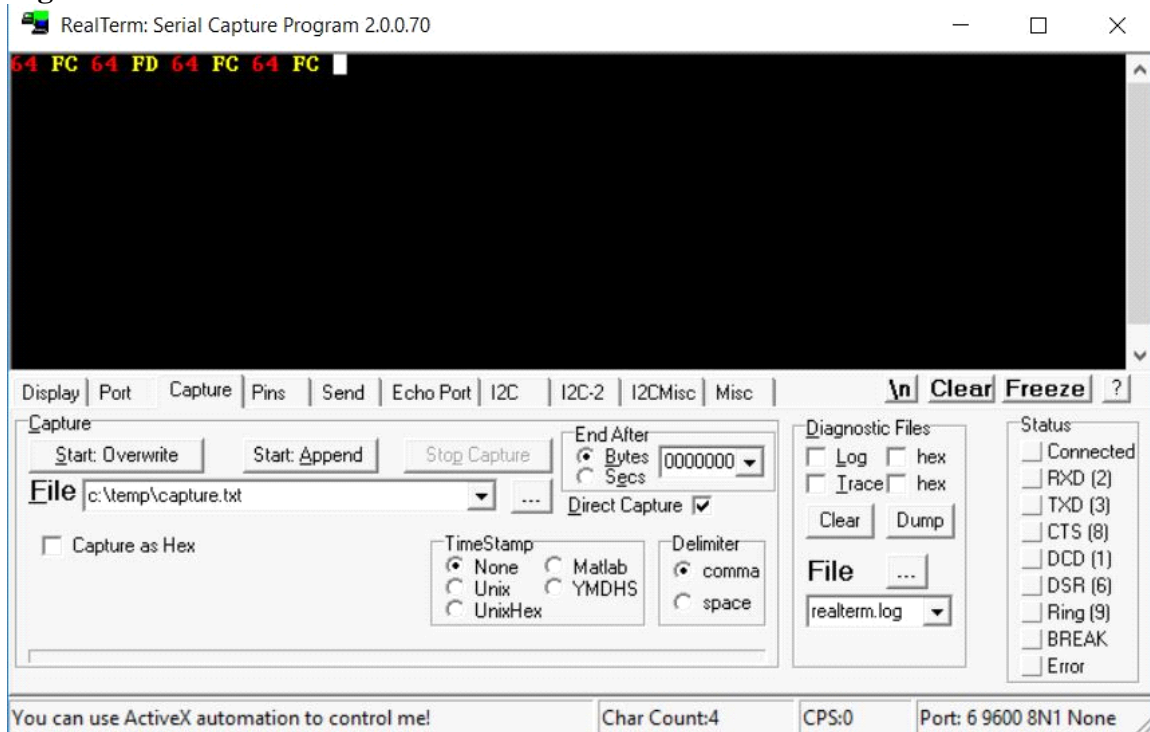


**Light On:**





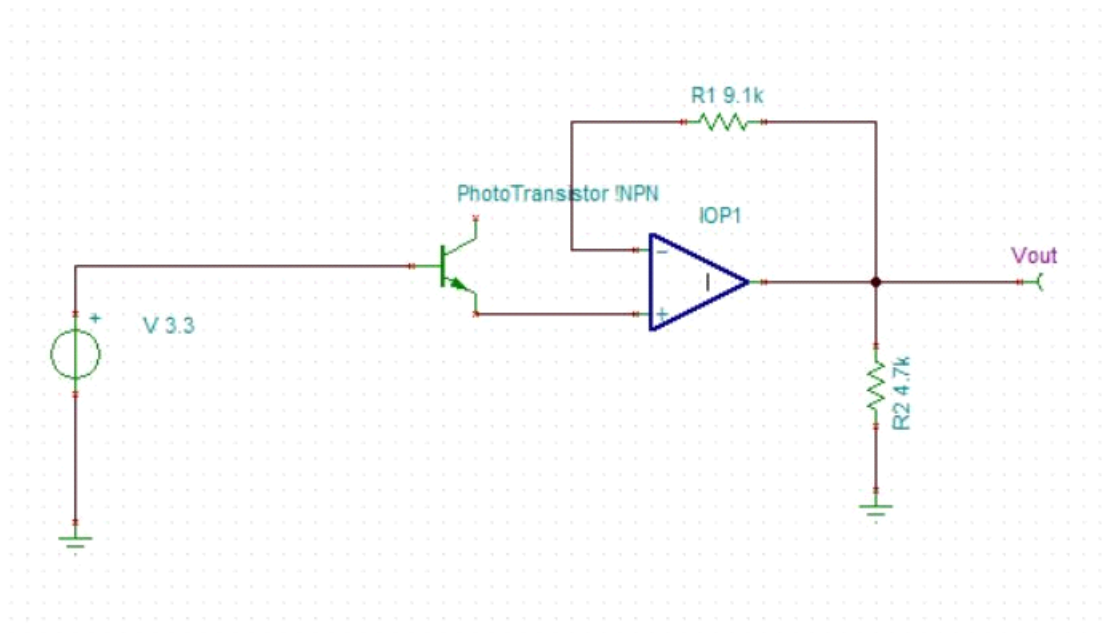
## Light Off:

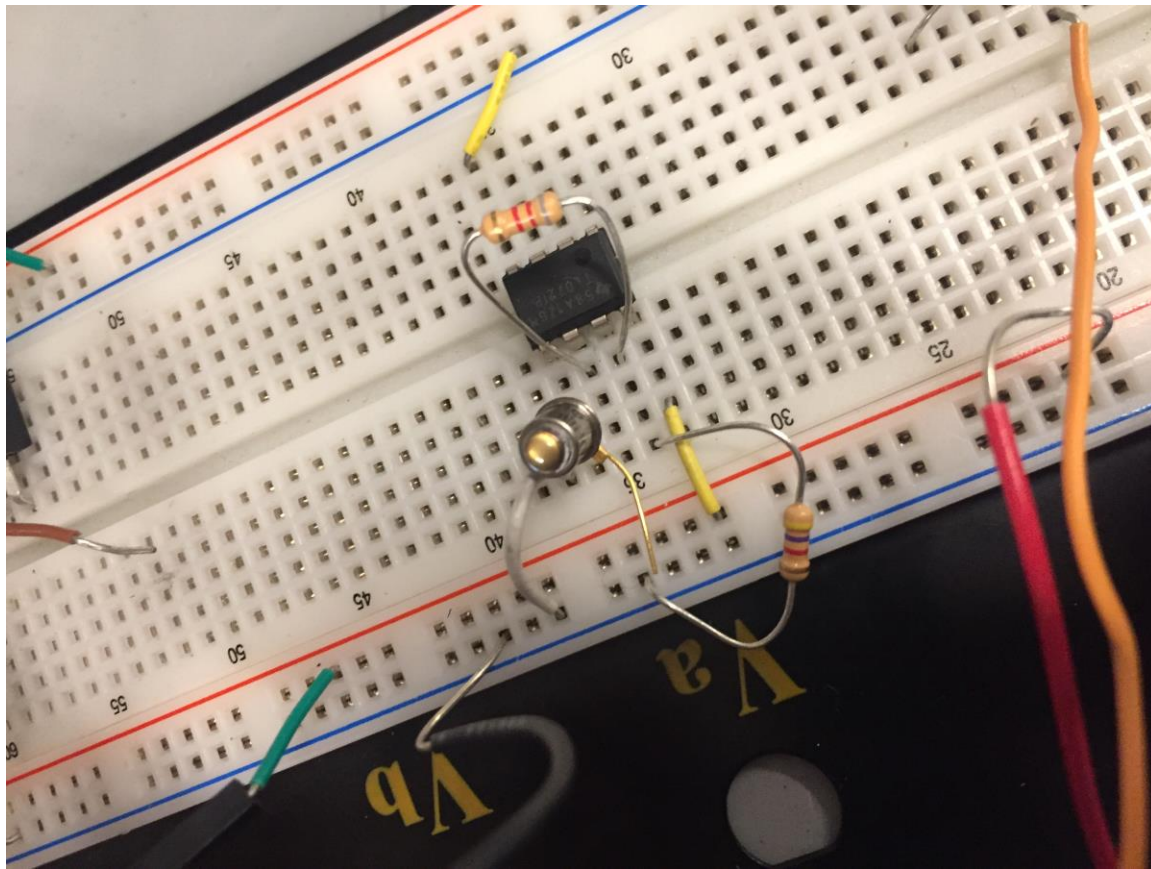


When the light is turned off the output voltage increases from 8B (139 in decimal) to FC (252 in decimal). This shows that the photodiode circuit is working properly and the output voltage is successfully being sent and received by UART.

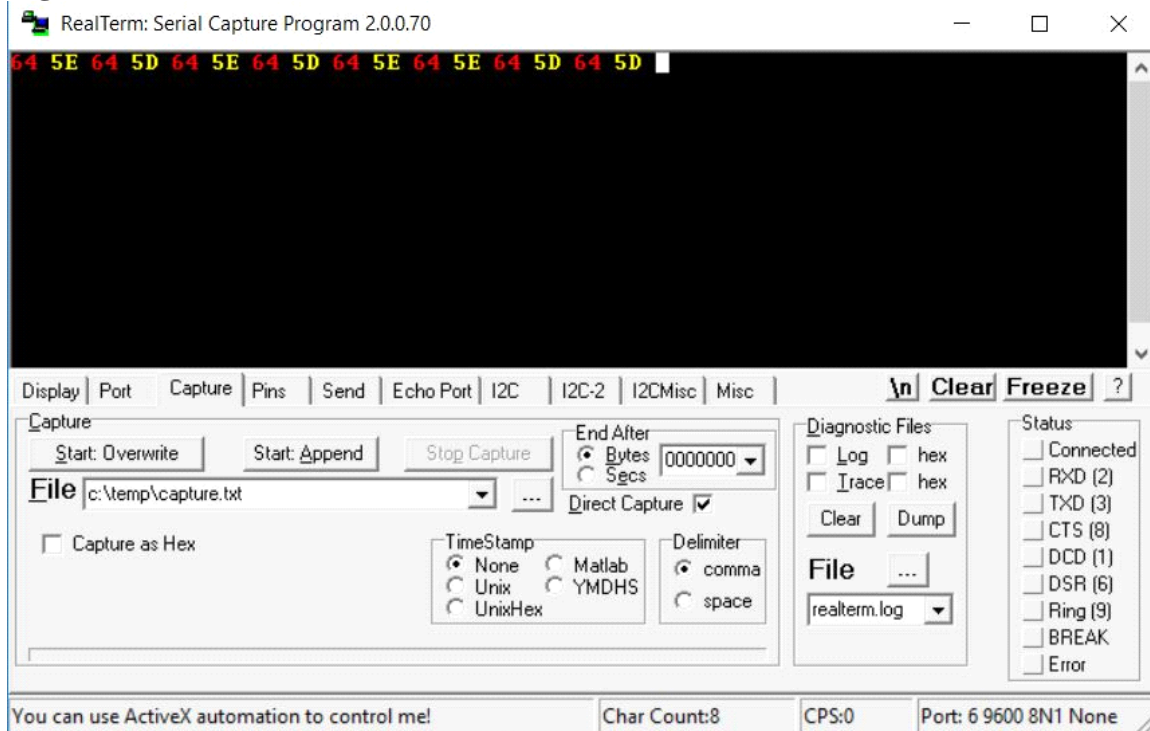
## Phototransistor

A phototransistor is simply a light-sensitive transistor and is very similar to a photodiode. As photons hit the phototransistor, electrons are generated in the base collector junction and are injected into the base. As this occurs, the current is amplified by the transistors at a magnitude equal to the phototransistors gain. Phototransistors are no more efficient at detecting low levels of light as opposed to photodiodes but have a significantly longer response time. The schematic for the phototransistor diode may be viewed below. Because only the gate and drain pins were used, the phototransistor is acting as a photodiode. Further, the circuit has an op amp in a noninverting configuration with a gain equal to 3. This will amplify the signal and reduce noise.

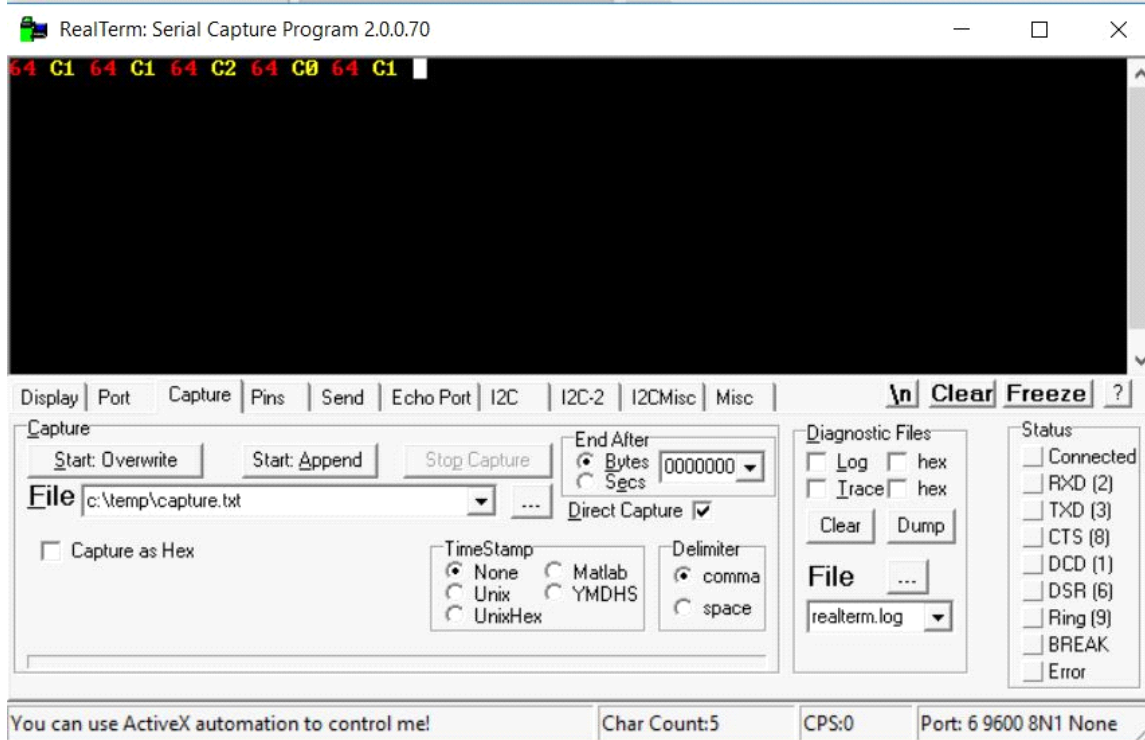




## Light On:



## Light Off:



When the light is turned off the output voltage increases from 5E (94 in decimal) to C1 (193 in decimal). This shows that the phototransistor circuit is working properly and the output voltage is successfully being sent and received by UART.

## How to Implement Code

1. Gather all three sensors.
2. Construct circuits using schematics as a reference.
3. Copy main.c code and copy it into Code Composer.
4. Debug and program your MSP430F5529.
5. Connect all wires from MSP430F5529 to circuit. (3.3V to power rail, GND to ground rail, P6.0 to Vout of circuit being tested).
6. Remove headers from RTS>> and CTS<< of the MSP430F5529. This is because UART1 is being utilized not UART0.
7. Open RealTerm and change "Display As" to "Hex[space]", check "Half Duplex", change the baud to 9600 and the Port to whichever USB port the MSP430F5529 is plugged into. Now begin typing "d" into the command window and collect the transmitted data.



## ADC10

After the ADC12 was finished for the MSP430F5529, an ADC10 code was written to be used with the MSP430G2553. The code was written, debugged, and tested to ensure full functionality. It has been confirmed that all of the circuits functioned properly and successfully communicated over UART to ADC12, therefore, it was decided to not repeat the same measurements with ADC10 as the same results are to be expected, just collected differently.

There are some key differences between ADC12 and ADC10 that will require different implementation in the code. To begin, ADC12 is a 12 bit signal whereas ADC10 is a 10 bit signal. The voltage being collected will still only be an 8 bit signal (FF is max) and therefore we need to shift the signal 2 bits for ADC10 whereas the signal was shifted 4 bits for ADC12. Further, initializing UART needed to be changed because a different board is being used, however, the same 9600 baud rate is used and 2 pins, RX and TX, are still initialized. Initializing ADC10 requires some different steps as opposed to initializing ADC12. Initializing ADC10CTL0 requires the extra commands SREF\_0 which will enable the ADC10 external reference, as well as REFON which will turn the reference on. Finally, the pin that collects the voltage is now P1.7 as opposed to P6.0 for ADC12. To continue, when utilizing ADC10, a timer, in this case Timer A0, needs to be initialized as well as a clock. For the Timer A0, TA0CCR0 was set to a period of 50000 while TA0CCR1 was set to a period of 30000. SMCLK was used. For the clock, the lowest settings for DCOx and MODx were selected. BC1 was set to 1Mhz and DC0 was set to 1Mhz.

Besides the items listed above, implementing ADC10 is very similar to implementing ADC12.