TIMELINE

# Dual Wire Transfer

Our first task was to implement transfer of bits between two arduinos using a combination of two wires. One wire transferred a stream of bits and the other one maintained a clock to notify an incoming bit at the receiver end. At the receiver end the bits were received in packets of 8 and were converted into their decimal equivalent.

# Single wire transfer

Here we tried to transfer data bits using a single wire, because ultimately for using Li-Fi we had to achieve a method for one way transfer of data. Since we didn’t have a clock at the receiver’s end, we encoded each data bit in the form of three bits. We encoded 1 as 110 and 0 as 100 at the transmitter arduino. At the receiver’s end our final data was decoded from the input stream by measuring the delay between a HIGH and corresponding LOW signal. For instance , if our data was bit 1 the delay was twice as much as that in bit 0. Here too, our data stream was decoded into packets of 8 bits and was subsequently converted to corresponding decimal.

# Emulating Li-Fi

In order to emulate actual transfer through light and corresponding receiving through a photosensor, we used a CNY70 (a reflective optical sensor) available in the proteus library . Though this step was far from the actual conditions (lack of real world noise and many other factors), but still helped us understand the basic circuit requirements of light transfer. This was similar to the single wire transfer, except we used an amplifier and comparator setup at the receiver’s end to recover the original digital wave, from the photosensor output. Theoretical speed upto 8000-8500 bytes per second was achieved using this method.

# Modulation

Owing to the practical limitations , we decided to explore ways to make our data transfer effective. In order to increase the speed and bandwidth of transfer and to enable frequency channels in our setup, we tried to modulate our data through a carrier wave. It was decided to proceed with the Phase Shift Keying modulation (PSK) technique.

# 

PSK MODULATION

# Theory

PSK is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine and cosine inputs at a particular time. PSK technique is widely used for wireless LANs, biometric, contactless operations, along with RFID and Bluetooth communications.

In PSK, the [constellation points](https://en.wikipedia.org/wiki/Constellation_diagram) chosen are usually positioned with uniform [angular](https://en.wikipedia.org/wiki/Angle) spacing around a [circle](https://en.wikipedia.org/wiki/Circle). This gives maximum phase-separation between adjacent points and thus the best immunity to corruption.

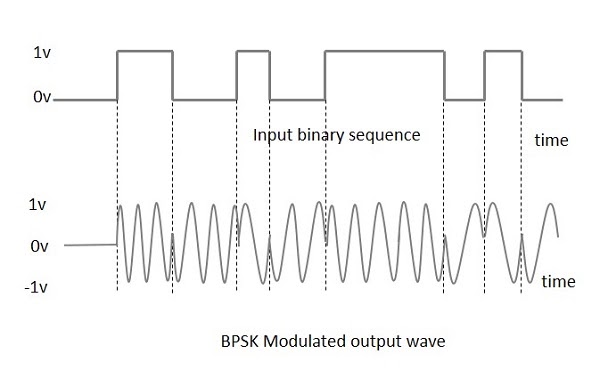
Two common examples are "binary phase-shift keying" ([BPSK](https://en.wikipedia.org/wiki/Phase-shift_keying#Binary_phase-shift_keying_(BPSK))) which uses two phases, and "quadrature phase-shift keying" ([QPSK](https://en.wikipedia.org/wiki/Phase-shift_keying#Quadrature_phase-shift_keying_(QPSK))) which uses four phases, although any number of phases may be used. Since the data to be conveyed are usually binary, the PSK scheme is usually designed with the number of constellation points being a [power](https://en.wikipedia.org/wiki/Power_(mathematics)) of two.

# BPSK

BPSK is the simplest form of phase shift keying (PSK). It uses two phases which are separated by 180° and so can also be termed 2-PSK. Therefore, it handles the highest noise level or distortion before the [demodulator](https://en.wikipedia.org/wiki/Demodulator) reaches an incorrect decision. That makes it the most robust of all the PSKs. It is, however, only able to modulate at 1 bit/symbol and so is unsuitable for high data-rate applications.

**BPSK Modulator**

The modulation of BPSK is done using a balance modulator, which multiplies the two signals applied at the input. For a zero binary input, the phase will be 0° and for a high input, the phase reversal is of 180°.The output sine wave of the modulator will be the direct input carrier or the inverted 180°phaseshifted ,180° phaseshifted input carrier, which is a function of the data signal.



# QPSK

The **Quadrature Phase Shift Keying**QPSK.QPSK is a variation of BPSK, and it is also a Double Side Band Suppressed Carrier DSBSC modulation scheme, which sends two bits of digital information at a time, called as **bigits**.

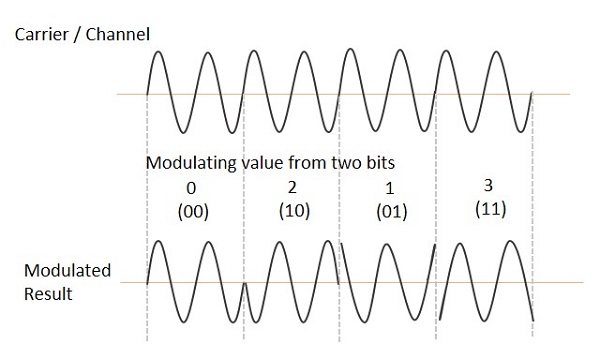
Instead of the conversion of digital bits into a series of digital stream, it converts them into bit pairs. This decreases the data bit rate to half, which allows space for the other users. In simpler terms the bandwidth of QPSK is double that of BPSK, which increases transfer rate.

**QPSK Modulator**

The QPSK Modulator uses a bit-splitter, two multipliers with local oscillator, a 2-bit serial to parallel converter, and a summer circuit.

At the modulator’s input, the message signal’s even bits (i.e., 2nd bit, 4th bit, 6th bit, etc.) and odd bits (i.e., 1st bit, 3rd bit, 5th bit, etc.) are separated by the bits splitter and are multiplied with the same carrier to generate odd BPSK (called as **PSKI**) and even BPSK (called as **PSKQ**). The **PSKQ** signal is anyhow phase shifted by 90° before being modulated.

The QPSK waveform for two-bits input is as follows, which shows the modulated result for different instances of binary inputs.

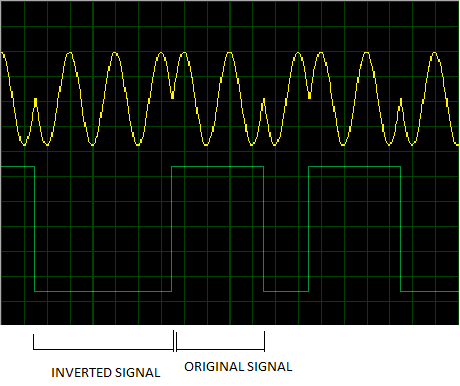


# IMPLEMENTATION

Since the QPSK is just an extension of BPSK, with a bit splitter being an additional requirement, we decided to go on with BPSK first, for the sake of simplicity.

**TRANSMITTER**

BPSK modulation was achieved using a multiplexer( 74HC4051) available in proteus. An carrier wave generated from an oscillator was fed into the X1 channel of multiplexer, while an inverted version of the same wave was fed into the X0 channel. Pin A (representing the leftmost bit of the channel decider) was fed with the data stream. All other pins of the multiplexer were grounded. Thus the output of the multiplexer was either the original carrier wave (if data bit was 1), or a 180° shifted signal (if data bit was 0). This comprised the final modulated signal.



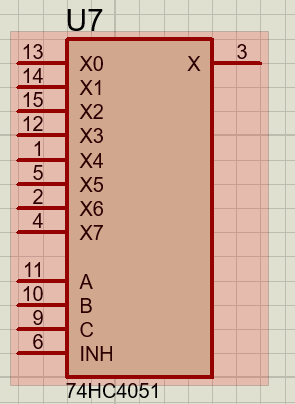
HARDWARE

# ARDUINO

The arduino is the brain of the setup. Encoding of data at the transmitter side , and subsequent decoding at the receiver end is done by arduino. It generates a digital stream of encoded bits which is fed into the multiplexer. To decrease the load on arduino and improve its performance tasks other than encoding and decoding are performed by IC’s.

# MULTIPLEXER

74HC4051 multiplexer is available in the proteus library. It consists of 8 input channels numbered from 0 to 7. Additional three pins are available to feed a three bit binary number which decides the output of the multiplexer. For details refer BPSK modulation.

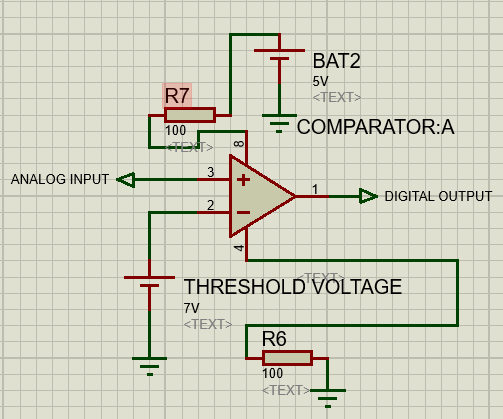


# COMPARATOR

The data received from the photo-sensor is in the form of an analog signal. This needs to be converted to a digital signal before feeding it into the arduino. However the inbuilt ADC of the arduino is too slow for our needs. Hence we need to use a dedicated ADC. Since the data is encoded purely in terms of the frequency of the signal, we do not care about any intermediate values of the amplitude. We just

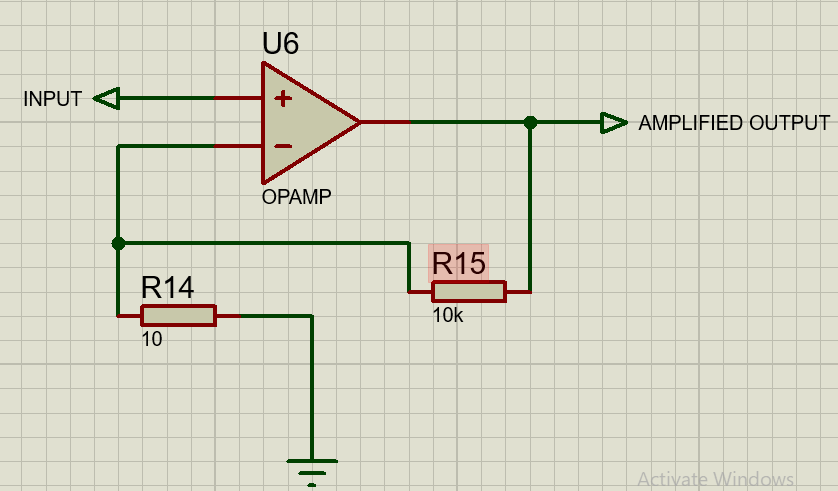
need to distinguish between the HIGH and LOW cases. This is achieved by using

a Op-Amp in a Comparator configuration.



# AMPLIFIER

The strength of the signal received from the photo-sensor is considerably low. For further processing this incoming analog signal needs to be amplified using an OP-AMP in a non inverting configuration mode.



# INVERTER

For implementation of the PSK , we need to have an inverted signal of the original carrier wave. This is obtained using OP-AMP as an inverter.

