# Optical communication in free space

Department of Computer Science

Ariel University

*Samuel Schwarcz and Elad Havshush*

**Abstract**

This paper describes the current technologies in the optical communications field and our work building an FSO system. The goal of the project is to build an optical communication system based on LED flashes. This system can provide satellites the ability to transfer data without using to much space and resources. The system conclude both receiver and sender modules. The system operation include two phases:
1.acquiring connection between the sender and the receiver and tracking the sender and than sending the data using flashes from LED installed on the sender and converting the transferred bits of data to usable data(Images, Telemetry etc). The system is consisted of two parts - sender built on Arduino with LED flash and receiver which built on pc with camera. we used both c++ and pyhon to build the system. The system we built capable of transferring one byte of information in about 3 seconds. This result id due to technical limitations of the camera and the PC and can be improved using better camera or light sensor.

# Contents

Chapter 1

# Introduction

## §1.1. Optical Communications

### 1.1.1 *Overview of FSO communications*

In recent years, free space optical (FSO) communication has gained significant importance owing to its unique features: large bandwidth, license free spectrum, high data rate, easy and quick deployability, less power, and low mass requirements compared to the traditional radio-wave based communications. FSO communication uses optical carrier in the near infraredband to establish either terrestrial links within the Earth's atmosphere or inter-satellite/deep space links or ground-to-satellite/satellite-to-ground links. alongside the mentioned advantages, there are several limitations like the atmospheric environment and the high precision needed to transfer the data. There are several approaches to build FSO communication system like using laser, LED, and also several approaches to overcome the difficulties and to prevent data loss.

### 1.1.2 *System description*

Our proposition is to build an optical communication system based on LED flashes for satellites. Our approach uses visible light to transfer data from the satellite to ground station but can also be used to transfer data between different sources. our motivation is to provide micro-satellites the ability to transfer data without using to many resources and space on the satellite. The data is represented as arrays of bits. Before the data(Images, Telemetry etc.) transfer begins, the sender have to flash the light in particular known frequency so the receiver can detect the sender and track him to get the data. after pre-defined time where the receiver detects the flashing LED with the correct frequency it turn into receiving mode and gets the data.
after the connection is made the sender send the bits via flashing LED with on-off keying method - when the LED is on it gets the value 1 and when its off it gets 0. The receiver gets the bits and translate them back to usable data. The System provides one-way communication, the LED transmit data by flashlight and the camera receive this light. that mean that the LED device doesn't need any information or accreditation from the receiver to start or continue emitting data. it will continually transmit until it finishes and stop. that mean that if any problem interferes during the reception by the camera device, some data can be lost.

Chapter 2

# System Review

In this chapter we will review the Modules of the system and how they operate together in order to send and receive the data. First we will review the sender part and than The receiver and the combined process.

## §2.1. The Sender Module

The sender part is built on Arduino using C++. The board we used is AI Thinker ESP32-CAM - an Arduino Board with camera and LED flash module.

### 2.1.1 *Signaling Phase*

When the satellite is in range and need to transmit data, the module send signals to the ground station. The signal consist of LED flashes in frequency of 200 millisecond for 5 seconds so the ground station can recognize and track the satellite. After 5 seconds the LED stop flashing for 4 seconds and than flashes one more time for 200 milliseconds to indicate the data transmission.

### 2.1.2 *Data Transmitting Phase*

After signaling the receiver the module starts transmitting the data. The data is transferred in packets composed by: A single shorter bit "1" (ON) that indicate the start of the packet -8 bits that represent the data, it can indicate chars for a text or a RGB color for an image. The packets are being transferred using on-off keying when every 300 milliseconds of flash indicates the bit 1 and 300 millisecond without flash indicates the bit 0. After every packet except the last one, the module signals the Receiver that a packet was sent by setting the flash off for 100 milliseconds and than the next packet signals again with another 200 milliseconds flash. After the last packet sent, the module send another Byte(01111111) indicates End Of File so he receiver knows that all of the data has been transferred, and stops the transferring process. While the satellite is not transferring data the module is responsible for getting the data from other satellite modules and prepare it for sending.

## §2.2. The Receiver Module

The receiver device we used is an ordinary computer camera that can read until 30 frame per second (FPS) and the algorithm we write to activate the camera is written in Python language using libraries of OpenCV and NumPy.

### 2.2.1 *General overview of the algorithm*

The program activates the camera that captures and wait for an event to occur. When it detects a light (HSV [0,0,255] white color) it will create a new process that will check if in this region of the frame, Region Of Interest (ROI), there is a device that flashes in the correct frequency.The lifetime of the process is 10 seconds. If the flash in the ROI doesn't blink at the expected frequency the process will be killed and the location of this ROI we will be save in order to not being disturb in the later by the flash in this area. If the flashes are at the correct frequency, the process will be kept in life and the process will start to track him in order to maintain connection during the data transfer, and will wait for the device to start transmitting message. When the flashes stops, the process goes into reception mode and will wait for the message to start. the message will be transmitted by On Off Keying (OOK). The sender will send an End Of File (EOF) signal it will kill the process. The choice of using multi-processing programming instead of simple multi-threading programming is motivated by the demand to reduce the response time of the CPU, so it will not affect the main process that is responsible for the frame capture handling, that can lead loosing data. The main process communicates with the ROI process by the help of two queues. The first queue will handle the information that the main process gives to the child process, and the second from the child to the main proccess.

### 2.2.2 *First signal reception*

When the program detects a flash it automatically open a new process that checks if the flashes are in the right frequency and continually sends the frames with the coordinates of the region .In order to detect, the process crops the frame at the given coordinates and measures the time the flash was ON and OFF, saves them to an array and calculate the average of the ON/OFF duration in order to check if it matches the expected ON/OFF duration. We chose to expect 0.13 seconds of ON and 0.07 seconds OFF to complete a period of 0.2 seconds. because the Average is not an accurate data, we add a variance error of 0.03 seconds to prevent errors. If the frequency is correct the program starts tracking the light and wait until the flashes are lost, and expects the data transfer.

### 2.2.3 *Data reception phase*

After the first phase is done the process enters to the second phase - receiving the data. The sender and the receiver must use the same data rate and the same Encoding/Decoding algorithm in order to transfer the data. the data is decomposed in packages in order to minimize the loss of information, if a bit on one packet is lost, it doesn't affect the rest of the packets. The module detects the flashes and convert the flashes into bits in bitrate of 1 bit every 300 milliseconds. The receiver gets one flash shorter than 300 milliseconds that indicates the beginning of a packet and starts storing the next bits(The data). After receiving 8 bits the receiver decodes the data into the sent character and stores it. The Real-Time encoding is required in order to recognize an EOF character that is sent after the last packet and indicates that the data transfer is done. Between each packet, the queue is cleaned in order to avoid reading garbage from the last packet. The EOF signal used is a packet containing the char "01111111" If the EOF signal get lost the program will end after 5 seconds without any data transferred. The final message will be display after the EOF reception. We used a rate of 1 bit per 300 milliseconds and a packet size of 8 bits, in order to avoid data loss due to the camera FPS and limitations of computing power.

Chapter 3

# Conclusion

During the development we encountered two issues that can be addressed to improve the system. the first is related to data loss, some bits can be missed due to environment and atmospheric conditions or camera speed and limitations of computing power. this issue can be addressed using faster camera or light sensor. also the the system can be used to create two-ways communications using another LED on the ground station and detect data loss with checksum or other method and initialize re-transmission . the second issue the data transmitting rate . in our experience we transmit data at a rate of 0.3 sec/bit that mean 3.3bit/sec . for example a satellite orbiting at LEO that can transmit during 6 minutes, can transmit less than 150 bytes . the issue can be resolved using a faster camera or and faster computer.