Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра «ЭВМиС»

Отчет по лабораторной работе №2

«Создание базы считывания данных с сканера кода O2I50x»

Выполнили:

Студенты 4-го курса

группы ПЭ-18

Мулярчик П.С.

Попко Д.П.

Краснов Р.А.

Мендельчук К.В.

Лавренюк И.А.

Подгорный А.В.

Проверил:

старший преподаватель

кафедры ЭВМ и С

Лапич С. В.

Брест 2020

Подключение сканера штрих кода осуществляем через кабель Ethernet для сканера O2I50x.

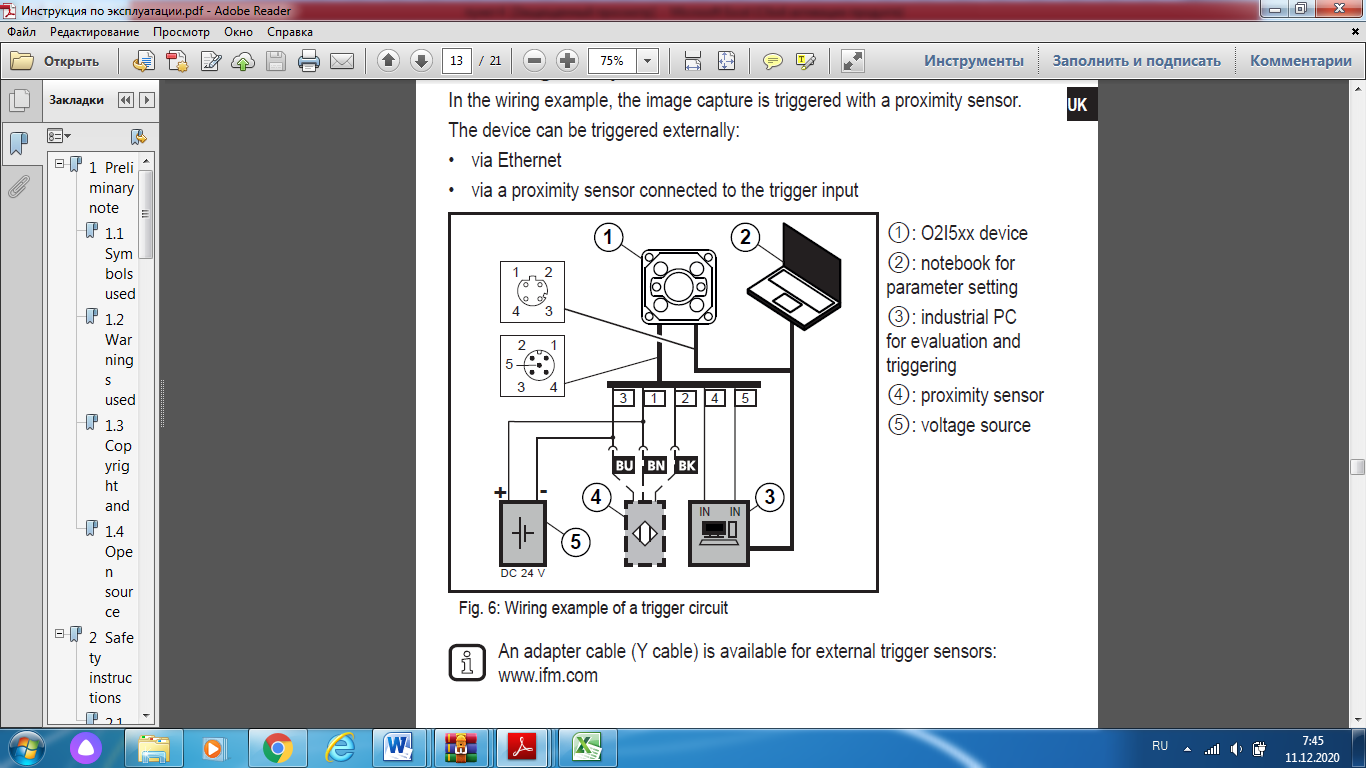


Рис.1 Схема подключения

1. Камера; 2- персональный компьютер для настройки параметров;
2. Промышленный компьютер для вычислений и переключений;
3. Датчик расстояния; 5- Источник питания.

В нашем случае нет промышленного компьютера для вычислений и переключений.

Для написания базы данных (кода) мы использовали :

Редактор - visual studio code

Среду разработки - node js

Сборщик - cmder mini

**Редактор visual studio code** нужен нам как платформа для написания кода на языке Java Script. Мы пишем код, сохряем его в папке в формате .json.

**Среда разработки Node.js** — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Грубо говоря, сам Node является приложением C++, которое получает на входе JavaScript-код и выполняет его.

Node.js просто содержит базу команд, инструкции по компиляции, и дает сборщику понять как именно нужно собирать проект.

**Сборщик cmder mini** или эмулятор другими словами - терминал для Windows. Cmder создан из-за отсутствия хороших эмуляторов консоли в Windows. Это бесплатная, не встроенная в операционную систему программа, которая упрощает работу с интерфейсом командной строки Windows.

Для создания базы считывания данных с сканера кода O2I50x среде разработки Node.js нужен ещё один код для настройки, указания версии программы, разработчика, названия программы.

{

"name": "free\_ifm",

"version": "0.0.1",

"author": "Mularchick Pavel",

"scripts": {

"start": "node index.js"

},

"dependencies": {

"express-generator": "^4.16.1",

"prompt-sync": "^4.2.0"

}

}

**Текст программы**

var net = require("net"); /\* Обращаемся к своей сети. Опрашиваем сетевые

порты **esernet \*/**

/\*var ip1;

var port1;\*/ // Перечисление переменных ip1, port1

var IFM\_Camera = new net.Socket(); // Объявление нового сетевого подключения

var conveyormoves = true; //Переменная типа булин. Если конвеер движется то true

function makeReq(ticket, message) { /\* Функция запроса на камеру,

let length = "" + (message.length + 6); запрашивает данные с камеры. Делаем запросы

while (length.length < 9) { определённой длины данных с этой

length = "0" + length; камеры. Пока длина данных меньше девяти, то мы заполняем длину до 9 битов \*/

}

IFM\_Camera.write(ticket + "L" + length + "\r\n" + ticket + message + "\r\n");

} //Отправляем запрос на получение данных

var fs = require('fs'); //Объявление переменной fs на работу с файлами fs, запрашивает.

var logger = fs.createWriteStream('Base of Data.txt', { // Инициализация базы данных

flags: 'a' /\* Флаг доступа,

}); чтобы поддерживалась синхронизация \*/

const prompt = require('prompt-sync')(); // const prompt присваиваем запрос ('prompt-sync')

const ip1 = prompt('Enter ip1: '); //Запрос на порт ip1 камеры

const port1 = prompt('Enter port1: '); // Запрос на порт port1 камеры

let date\_ob = new Date(); /\* Мы создаём объект класса Date. Наши данные будут храниться в new Date\*/

logger.write('\n'); ); /\* Это то, что мы записываем в базу данных и переходим на новую строку \*/

IFM\_Camera.connect(port1, ip1, function() { /\* Подключаем камеру ч/з порт и console.log("Camera#1 connected"); f-ию подключения в камере\*/

});

IFM\_Camera.on("data", function(data) { /\* Если камера включена и передаёт

console.log("Received: " + data); данные, то записываются данные в базу

logger.write('Item number: '+' Data: '+data+'\n'}); данных компьютера\*/

IFM\_Camera.on("error", function(err) { /\* Если сама камера передаст

console.log(err); ошибку, то тебя оповестят, что тут ошибка\*/

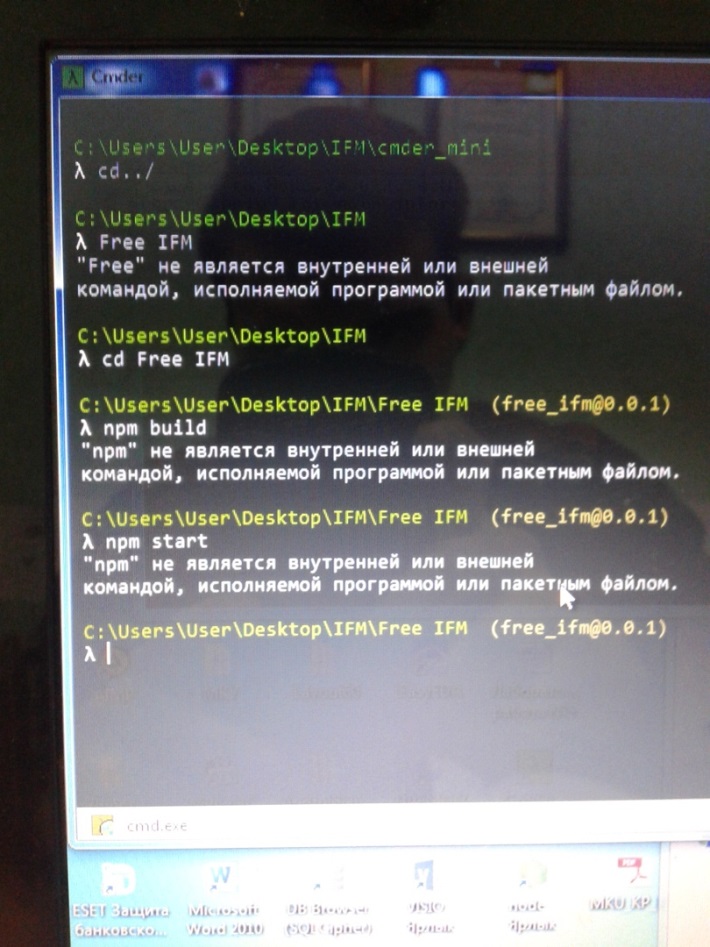
});

IFM\_Camera.on("close", function() { /\* Если камера передаст о закрытии

console.log("Connection closed"); подключения, то оповестят\*/

});

Это коды созданные в visual studio code.

Далее мы прописываем следующие команды в cmder mini. (Рис.2, рис.3)

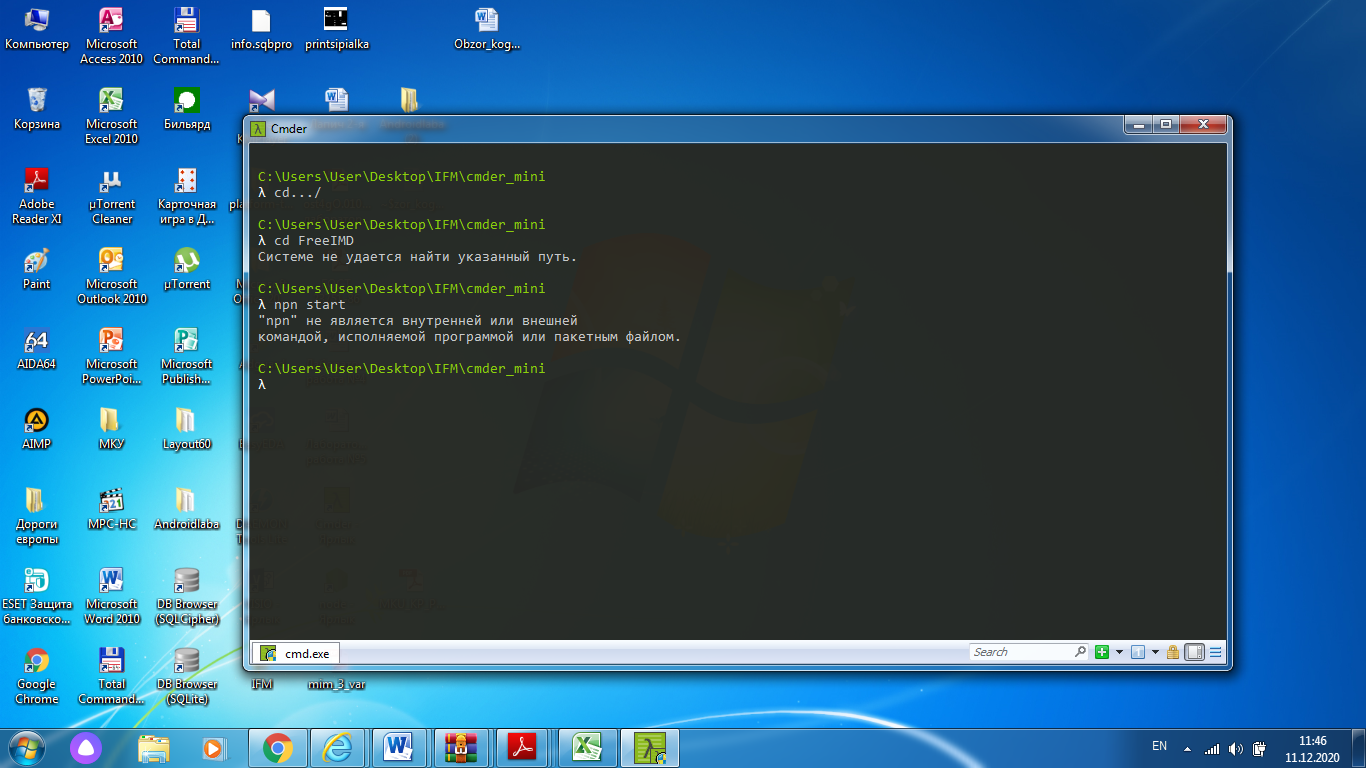


Рис.3

Первая команда cd.../ - это команда для того чтобы выйти из папки командера мини.

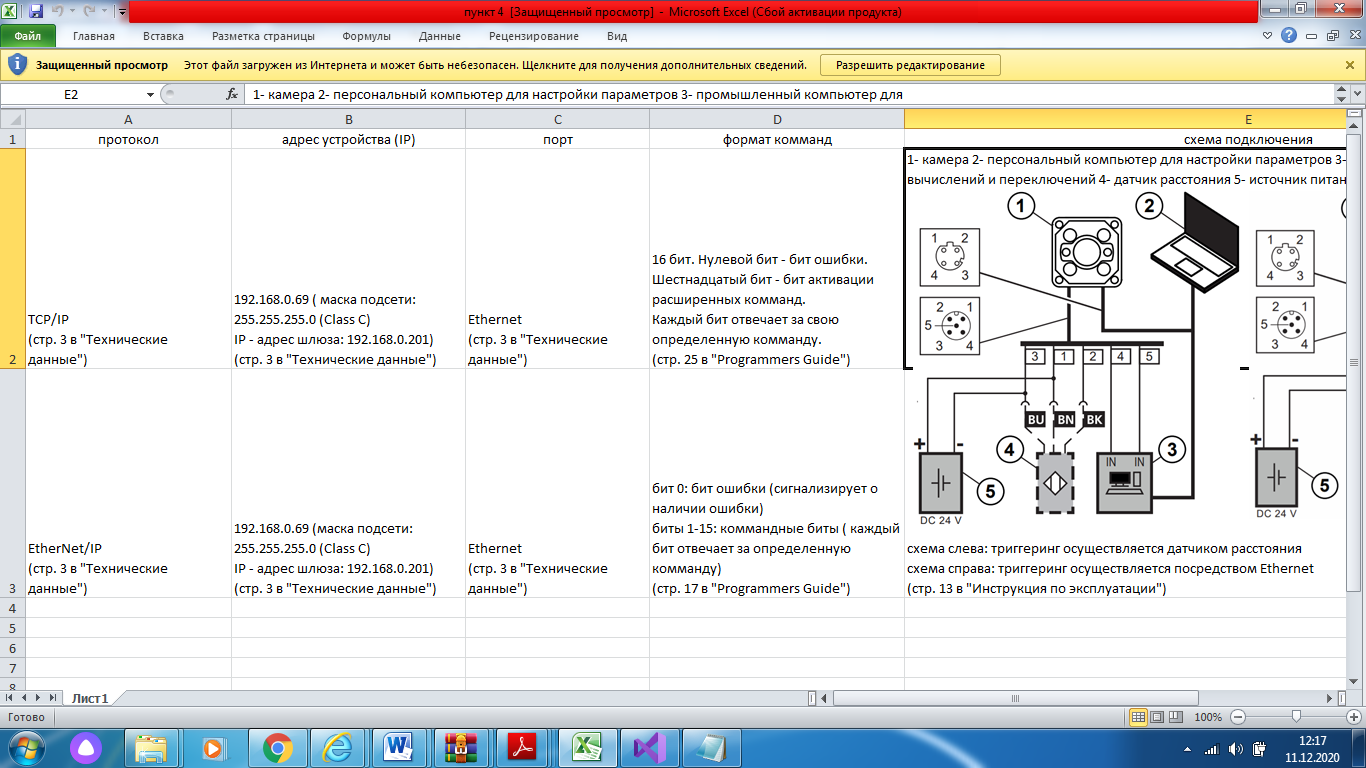
cd Free IFM команда- это обращение к папке Free IFM и его содержимому.

npm start - это запуск программы.

Ошибка выскакивающая при npm start, означает об отсутствии связи с средой разработки - node js, на которой написанный код переводился в машинный код.

Не забываем ввести необходимый IP и маску подсети камеры. IP компьютера должно отличаться от IP камеры.

По заводским условиям для камеры O2I50x



При подключении нашего компьютера к сканеру кодов мы в этой же программе cmder mini будем наблюдать данные считывания со сканера, которые будут заноситься в базу данных сборочных документов.

В данном случае в папке Bast of Data. Рис.5

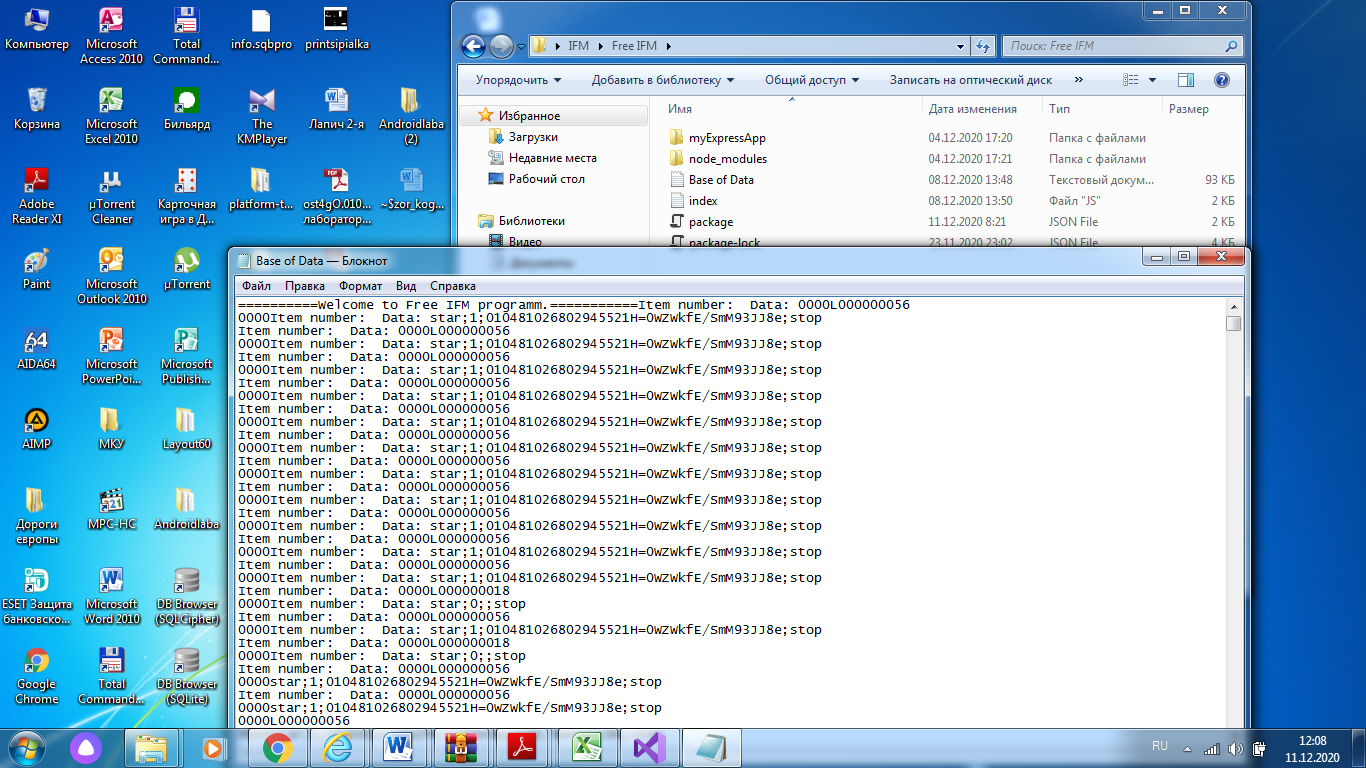


Рис.5

Вывод: создали базу считывания данных с сканера кода O2I50x, используя такие приложения как visual studio code, node js, cmder mini.