

Введение

С развитием цивилизации человек пытается облегчить себе жизнь используя технику. Но техника работает на топливе, оно сгорает и появляется углекислый газ. Что приводит к таянию ледников и к увеличению уровня воды в океане, который нас когда нибудь затопит. Ежегодно мы выбрасываем 55 миллиардов тонн углекислого газа. Виной этому 4 сектора экономики: энергетика на ископаемом топливе, транспорт, здания (отопление), промышленность.

В нашей работе мы рассмотрим общественный транспорт. В России на 2020 год было 96 тыс. автобусов. Из них электробусов — 611. Это чуть больше 0,6% от общего количества. Основная доля электробусов приходится на Москву — в столице их 592. В Москве на электробусы приходится 5,5% всех автобусов. Это примерно соответствует европейскому уровню использования такого транспорта: в 2020 году в ЕС электробусы занимали долю в 6,1% от общего числа автобусов. В нашем городе Екатеринбурге электробусов к сожалению нет вообще. Мы решили изучить, с чем связана эта проблема.

Самое сложное в обеспечении электроэнергией общественного транспорте – сложность хранения электричества. Нужно использовать либо сложные сети которые доставят электрический ток в транспорт (как провода у трамвая и троллейбуса или контактный рельс в метро), либо очень дорогие и порой небезопасные аккумуляторы с большой ёмкостью.

Для решения этой проблемы в общественном транспорте можно менять аккумуляторы на каждой остановке, ведь от одной остановки до другой в городе автобус едет примерно от 5 до 15 минут. Тогда можно будет использовать недорогие ионисторы, которые заряжаются очень быстро и за время проезда между остановками не успеют полностью разрядиться. Это значительно снизит затраты на создание и использование электробусов.

Для работы над проектом мы распределили усилия следующим образом:

Общая работа: придумывание концепции проекта, написание текста защиты и этого описания.

Закиров Тимур Разработка модели автобуса и остановки (модели в среде Компас 3D, 3D-печать, программирование автобуса на языке Python и программирование остановки на языке EV3-G)

Ковалюк Алексей Разработка системы распознавания посылок, интерфейса терминала остановки (Язык программирования C# и Arduino IDE). Разработка дизайна (поле для проекта, плакат, панель остановки в среде Adobe Illustrator).

Разработка модели электробуса

Аккумуляторы очень долго заряжаются и у них есть склонность к саморазряду хоть и небольшая, но главное они очень дорогие – иногда стоят больше половины стоимости всего электробуса. И для решения проблемы долгой зарядки и высокой стоимости мы предлагаем ионисторы. Плюсы ионисторов: они быстро заряжаются и намного дешевле, чем аккумуляторы. Минусы: ионисторы быстро разряжаются, а также у них большой саморазряд от бездействия. Но для коротких интервалов между остановками на маршруте это не очень важно.

Модель ионистора разработана в компас 3D. Для корпуса используется круглая деталь из пластика, чтобы электромагнит мог захватывать деталь также используется металлическая пластина.

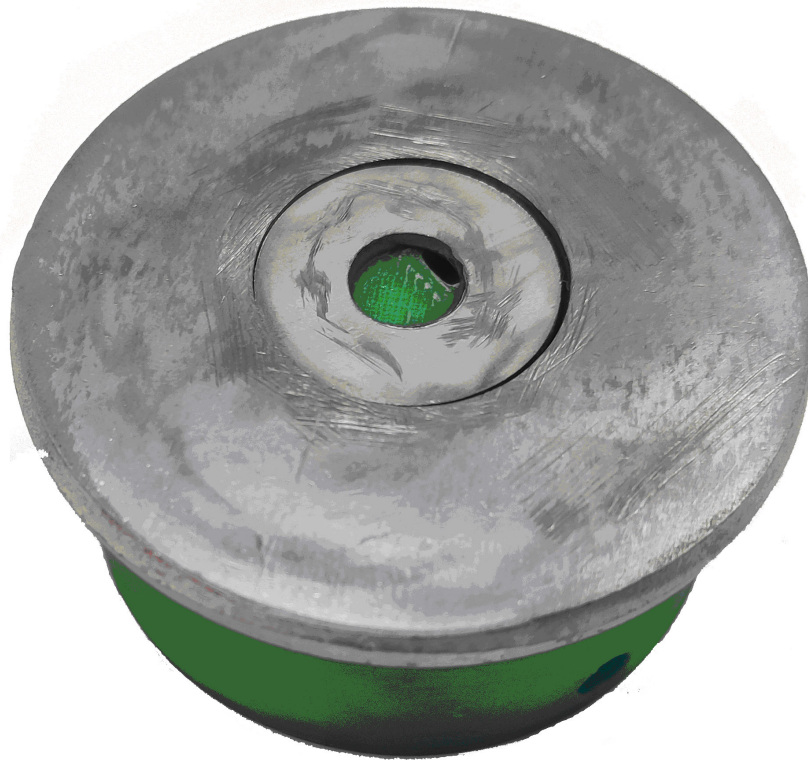


Рисунок 1. Ионистор.

Модель автобуса собрана на наборе Lego Spike Prime. Крыша, боковые стенки и передняя часть смоделированы в программе Компас 3D и распечатаны на 3D-принтере.

Скриншоты некоторых частей корпуса на рисунках ниже.

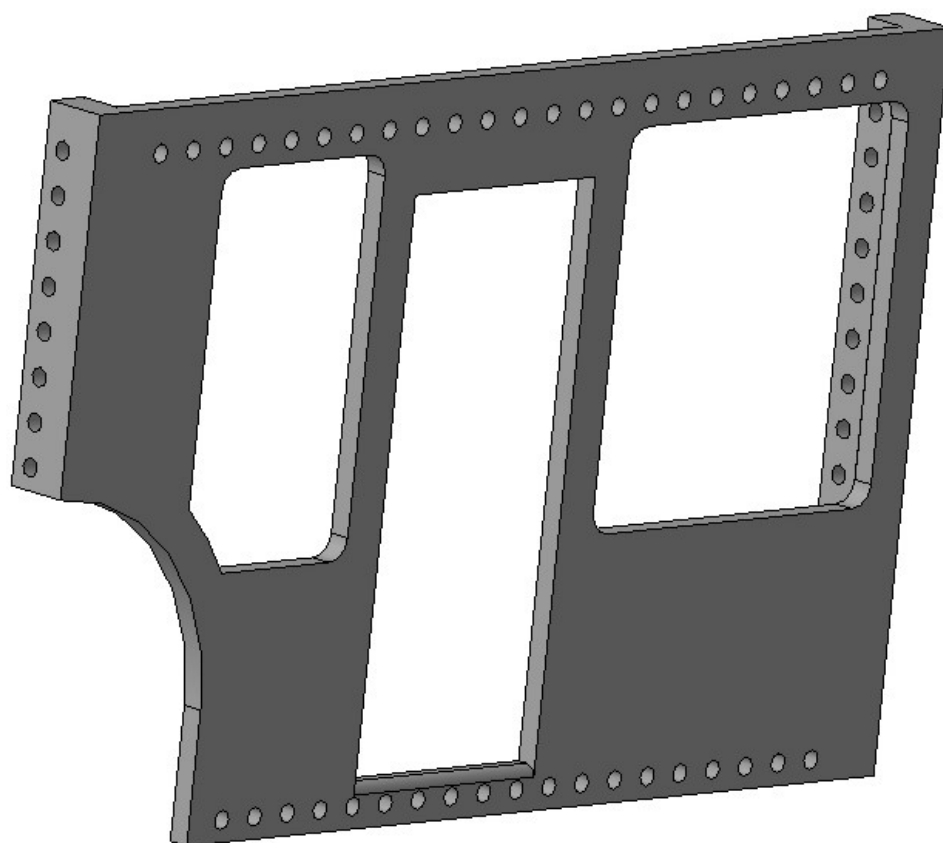


Рисунок 2. Боковая часть автобуса

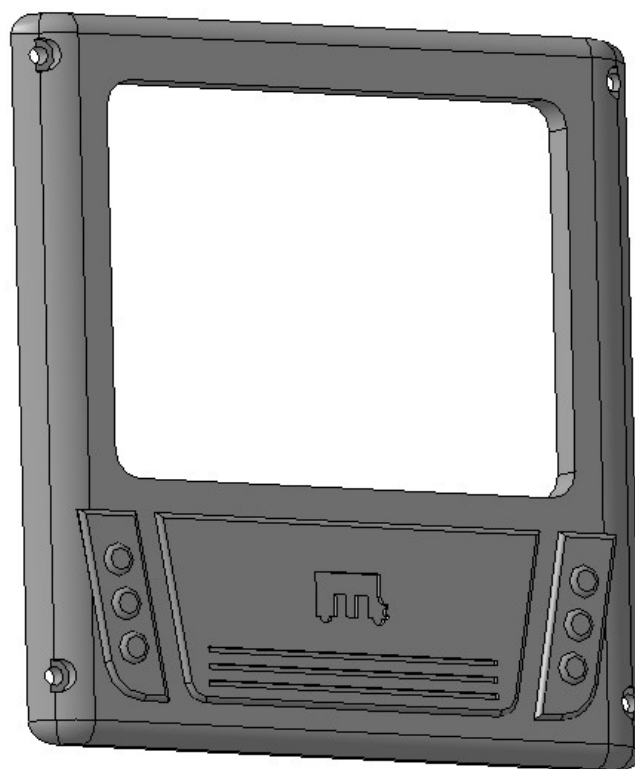


Рисунок 3. Передняя часть автобуса

Программа для автобуса написана в среде Lego Spike Prime. Рассмотрим некоторые её части. Вначале инициализации датчиков чтобы понимать какой как мы назвали. Здесь ещё значения переменных – серое для проезда по линии, средняя скорость и коэффициенты для регуляторов.

```
# Инициализация датчиков цвета
cross = ColorSensor ('F')
line = ColorSensor('D')
# Инициализация моторов
motorLeft = Motor('B')
motorRight = Motor ('C')
motorDoor = Motor ('E')
grayline = 75
pk = 0.55
pd = 5
speed = 30
olderror = 0
```

Эта часть программы для проезда по линии:

```
while cross.get_color() != 'yellow':
    error = line.get_reflected_light() - grayline
    u = pk * error + pd * (error - olderror)
    motorLeft.start_at_power(-1*(int(speed + u)))
    motorRight.start_at_power(int(speed - u))
    olderror = error
    #wait_for_seconds(0.01)

motorLeft.stop()
motorRight.stop()
olderror = 0
```

А здесь мы немного доезжаем чтобы встать точно в том месте, где удобно заменить аккумулятор:

```
motorRight.set_degrees_counted(0)

while motorRight.get_degrees_counted() < 200 :
    error = line.get_reflected_light() - grayline
    u = pk * error + pd * (error - olderror)
    motorLeft.start_at_power(-1*(int(speed + u)))
    motorRight.start_at_power(int(speed - u))
    olderror = error

motorLeft.stop()
motorRight.stop()
```

Далее посмотрим некоторые элементы конструкции остановки.

На рисунке ниже пример ножки. Сверху и снизу видны выемки под гайку. Также видно отверстие, в которое вставляется направляющая оси Y. На этой ножке видны крепления под мотор, остальные 3 ножки не содержат этих креплений.

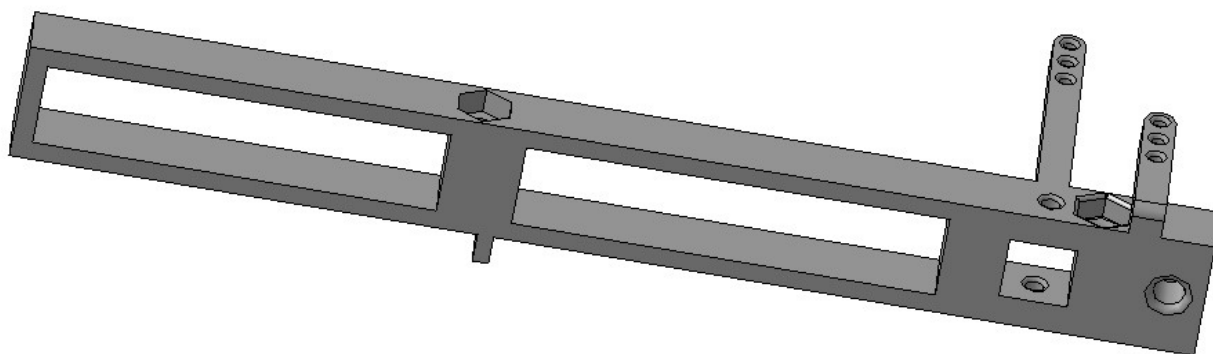


Рисунок 4. Ножка остановки.

На направляющих осях Y передвигаются на подшипниках LM8UU каретки, они собраны из двух частей. В верхнюю часть потом вставляем мотор от EV3. Сбоку отверстия направляющих оси X:

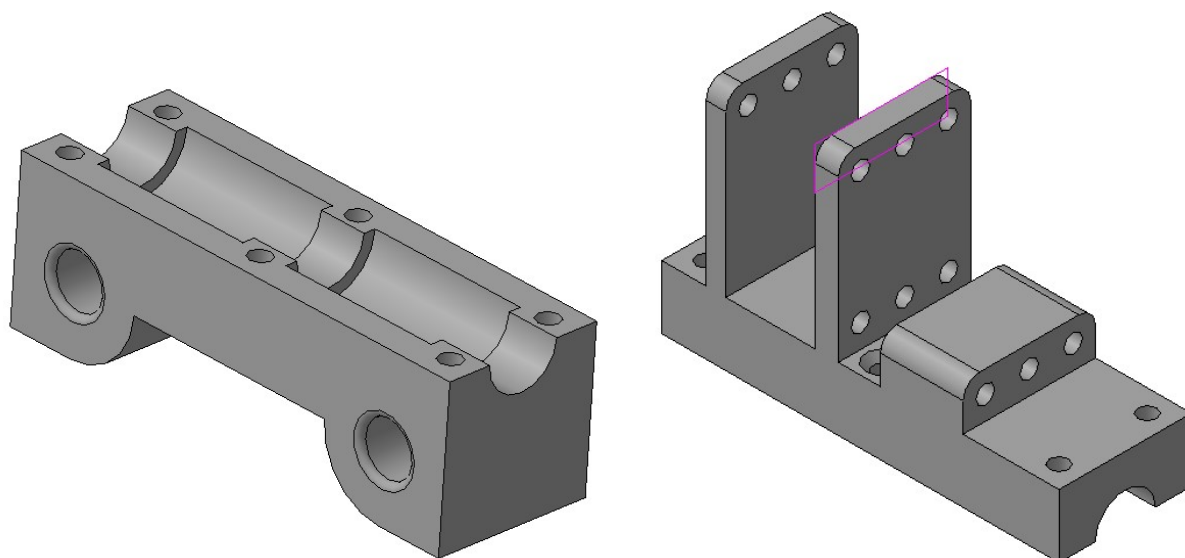


Рисунок 5. Каретка оси Y.

Вот так выглядит деталь, которая перемещается по оси X и как бы является верхней частью оси Z. В неё вставляются подшипники и крепится мотор с лебёдкой.

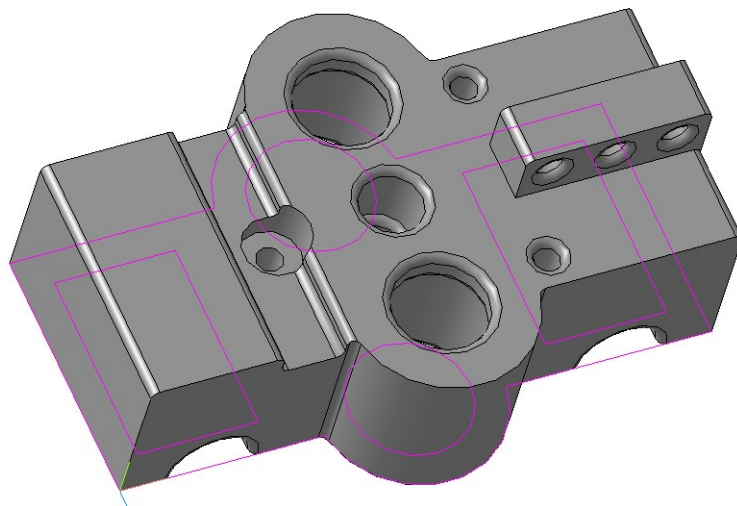


Рисунок 6. Каретка оси X.

Снизу деталь для электромагнита. По бокам потом прикручиваются зажимы для того чтобы держать направляющие. Вырез сделали чтобы было куда деть провода.

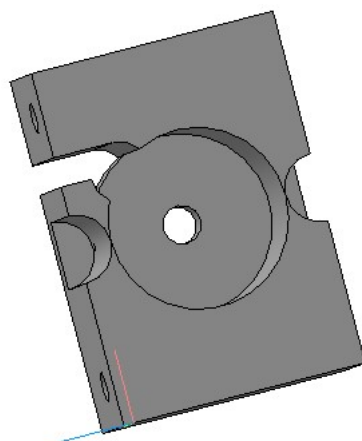


Рисунок 7. Деталь с электромагнитом – нижняя часть оси Z.

Это только часть деталей остановки, всего остановка собирается из 17 деталей.

На следующей странице приведён скриншот программы для остановки. Поднимание и опускание в ней настраивается блоками подпрограмм Give и Drop.

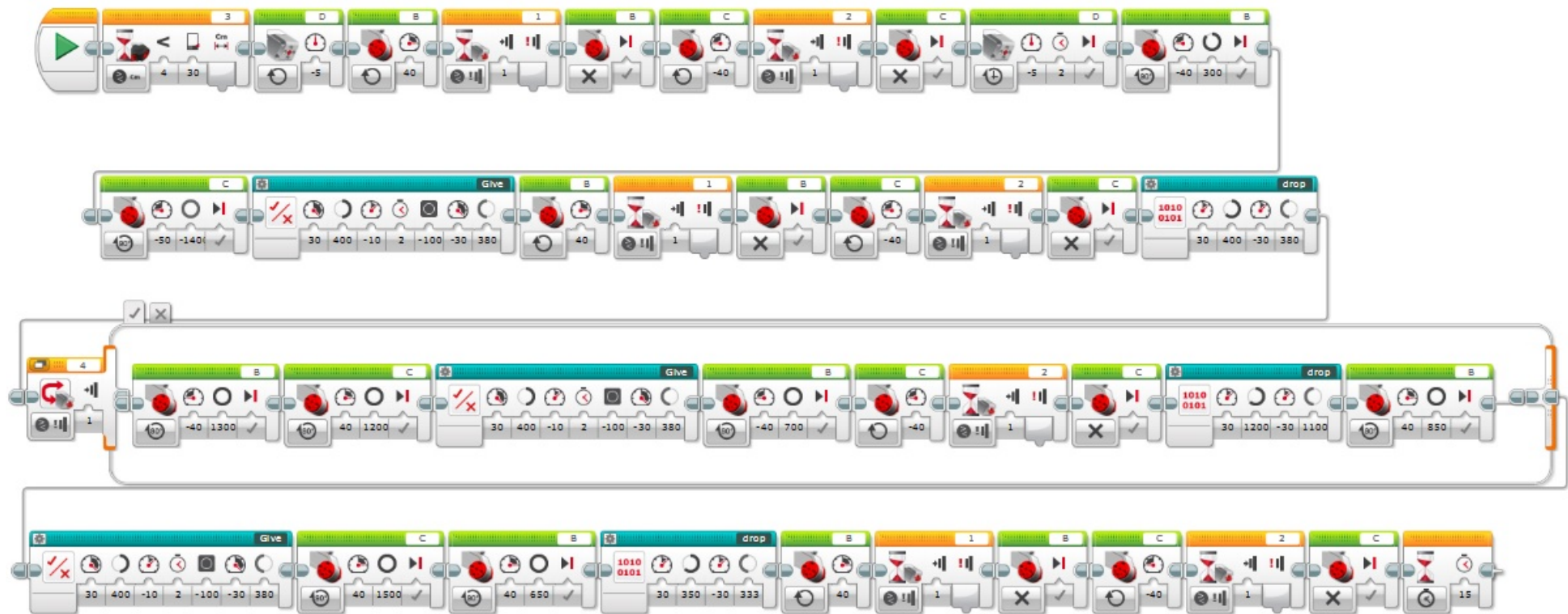


Рисунок 8. Скриншот программы для остановки.

Разработка системы отправки и распознавания посылок.

Отправка посылок с остановки будет осуществляться с помощью машинного зрения. Когда вы что-то заказываете в интернет магазине, курьер относит вашу посылку на остановку. Кладёт посылку в специальную ячейку с уникальным QR кодом, на каждой ячейке он разный. В удобном и простом интерфейсе выбирает остановку на которую нужно отправить посылку. Закрывает дверцу на на электрозамке.

Когда ближайший автобус подходит к остановке, камера закреплённая на остановке видит QR код и включает захват который во время замены аккумуляторов автобуса, загрузит на него ячейку с посылкой и автобус едет с ней до нужной остановки. На нужной остановке камера видит посылку и снимает её с автобуса и отправляет получателю код от ячейки с посылкой. Потом заказчик вводит в интерфейсе код который пришёл ему по SMS и забирает посылку.

Основная часть программы выполняется на компьютерном терминале остановки и написана на языке C# с использованием библиотек AForge.Video и ZXing. Для обмена данных с остановкой мы будем использовать Arduino с шилдом Eryoma и переходником Eryoma - EV3. Соединение с Arduino происходит через интерфейс COM-порта с использованием библиотеки System.SerialPort.

Рассмотрим основные алгоритмы программы для распознавания ячейки с посылкой. При запуске программы через использование события Load подгружается список видеокамер:

```
//загружаем список видеокамер
VideoCaptureDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
//очищаем список устройств
comboBox1.Items.Clear();
//если ни одно устройство не найдено генерируем ошибку
if (VideoCaptureDevices.Count == 0) throw new ApplicationException();
//переносим список устройств в список comboBox1 в текущем окне
foreach (FilterInfo device in VideoCaptureDevices)
{
    comboBox1.Items.Add(device.Name);
}
```

Также подгружается список доступных COM портов:

```
//получаем список доступных COM портов в массив строк
string[] ports = SerialPort.GetPortNames();
//переносим строки из массива в список comboBox2 в текущем окне
foreach (string port in ports)
{
    comboBox2.Items.Add(port);
}
```


Далее программа открывается и доступна для использования. Скриншот программы приведён на рисунке 1.

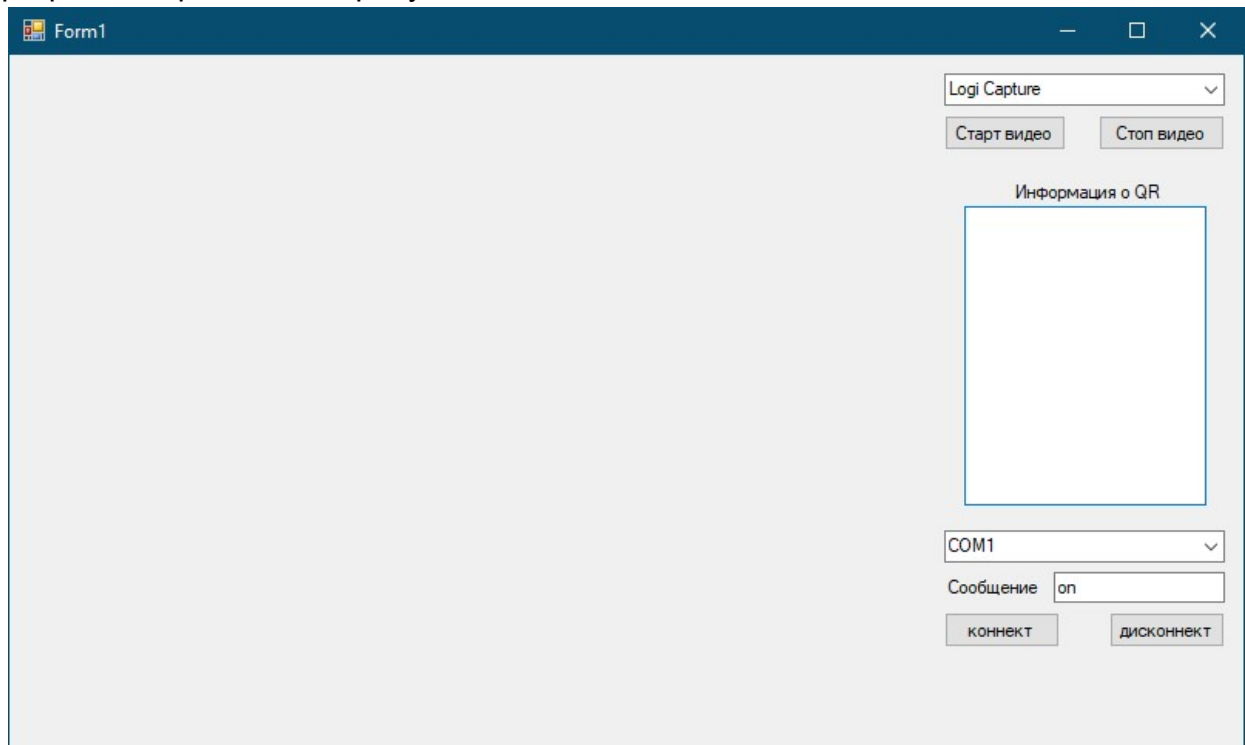


Рисунок 9.Скриншот программы при запуске

Для подключения камеры мы выбираем нужную в списке и нажимаем кнопку "старт видео". при этом выполняется захват потока видео

```
//подключаемся к выбранной видеокамере
FinalVideo = new VideoCaptureDevice(VideoCaptureDevices[comboBox1.SelectedIndex].MonikerString);
//при обновлении изображения с видеокамеры вызывается метод FinalVideo_NewFrame
FinalVideo.NewFrame += new NewFrameEventHandler(FinalVideo_NewFrame);
//начинаем работу с видеокамерой
FinalVideo.Start();
```

Для соединения с Arduino выбираем нужный COM порт: и нажимаем кнопку "коннект":

```
currentPort = new SerialPort(comboBox2.Text, 9600);
currentPort.BaudRate = 9600;
currentPort.DtrEnable = true;
currentPort.ReadTimeout = 1000;
try
{
    currentPort.Open();
}
catch {}
```

Дальше рассмотрим метод FinalVideo_NewFrame отвечающий за обработку кадров. Для начала мы создаём Bitmap который загружает видео изображение:

```
Bitmap video = (Bitmap)eventArgs.Frame.Clone();
video.RotateFlip(RotateFlipType.RotateNoneFlipX);
```

Далее мы пытаемся определить есть ли в кадре посылка с кодом:

```
BarcodeReader reader = new BarcodeReader();
Result result = reader.Decode(video);
if (result != null)...
```

При наличии посылки (кода на изображении) мы расшифровываем координаты и её номер:

```
string s = result.ToString() + Environment.NewLine;
string temp = result.ResultPoints[0] + " ";
temp = temp.Replace(" ", "r");
temp = temp.Replace("(", "");
temp = temp.Replace(")", "");
string sx = temp.Remove(temp.IndexOf("r"), temp.Length - temp.IndexOf("r"));
string sy = temp.Remove(0, sx.Length + 1);
if (sx.IndexOf(",") > 0) sx = sx.Remove(sx.IndexOf(","), sx.Length - sx.IndexOf(","));
if (sy.IndexOf(",") > 0) sy = sy.Remove(sy.IndexOf(","), sy.Length - sy.IndexOf(","));
int x = int.Parse(sx);
int y = int.Parse(sy);
```

Дальше мы добавляем к Bitmap кадрам с видеокамеры жёлтый прямоугольник в координатах метки:

```
Graphics fig = Graphics.FromImage(video);
Pen p = new Pen(Color.Yellow, 5.0f);
fig.DrawRectangle(p, x - 5, y - 5, 10, 10);
```

Далее мы отправляем информацию на COM порт:

```
if (currentPort.IsOpen && !isMessegeSend)
{
    currentPort.Write(textBox2.Text);
    isMessegeSend = true;
    timer1.Start();
}
```

Затем загружаем кадр с изображением в PictureBox:

```
pictureBox1.Image = video;
```

Пример обычного изображения и кадра с найденной Aztec меткой приведён на рисунках 2, 3.

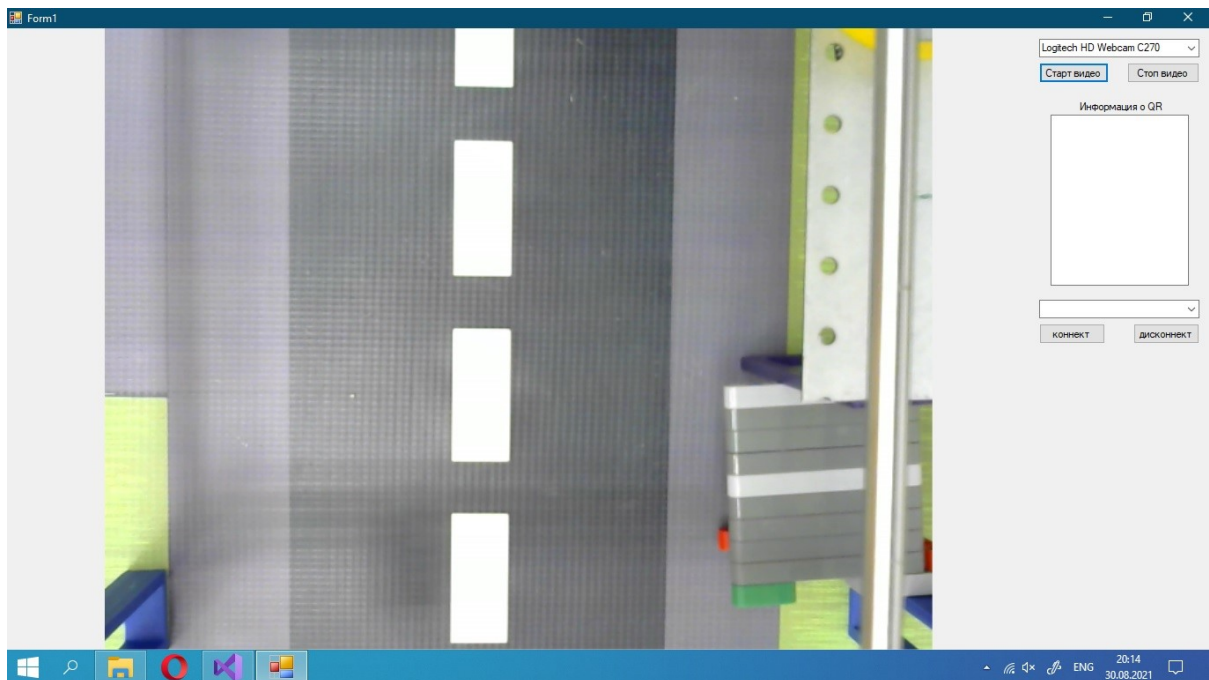


Рисунок 10. Скриншот программы с кадром без метки.

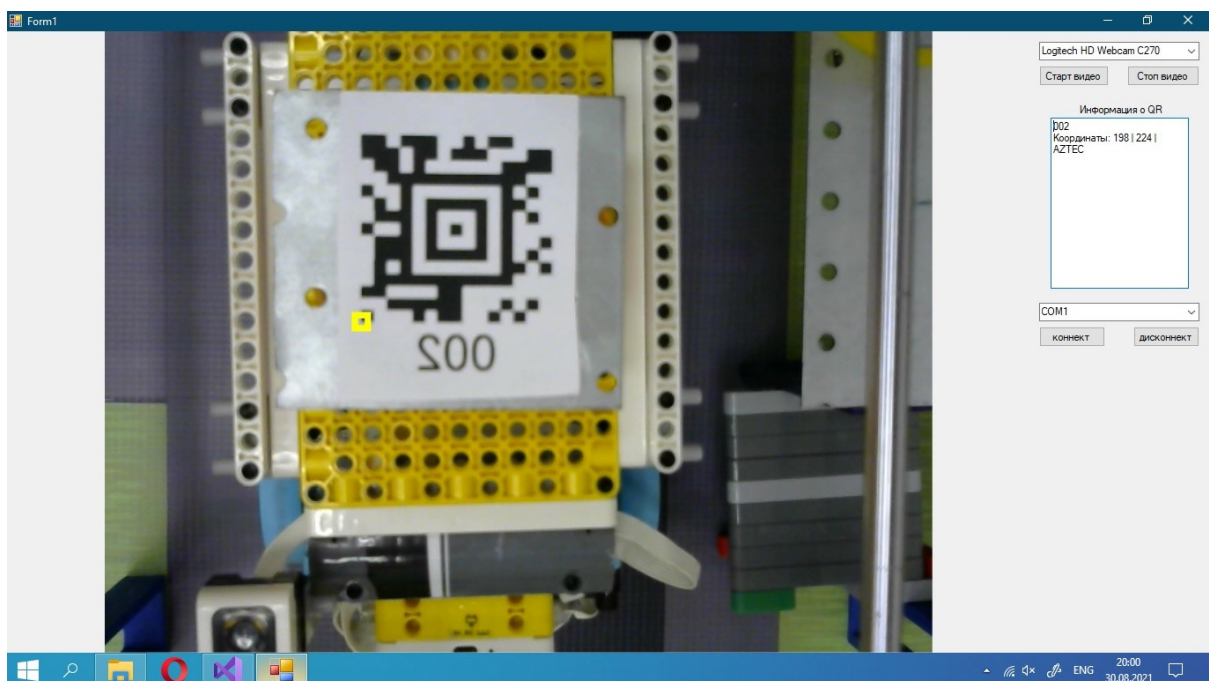


Рисунок 11. Скриншот программы с распознанной меткой.

На стороне Arduino мы принимаем сигнал и направляем его в EV3:

```
if (Serial.available() > 0) {  
  String str = Serial.readString();  
  if(str == "on") digitalWrite(A0, HIGH);  
  delay(1000);  
}  
digitalWrite(A0, LOW);
```

Также для терминала остановки написана отдельная программа на языке C#. В терминале отображается название остановки, текущее время и дата, погода на улице, время до прибытия транспорта. Также заложена возможность посмотреть карту маршрутов и самое главное – кнопки для получения и отправки посылок.

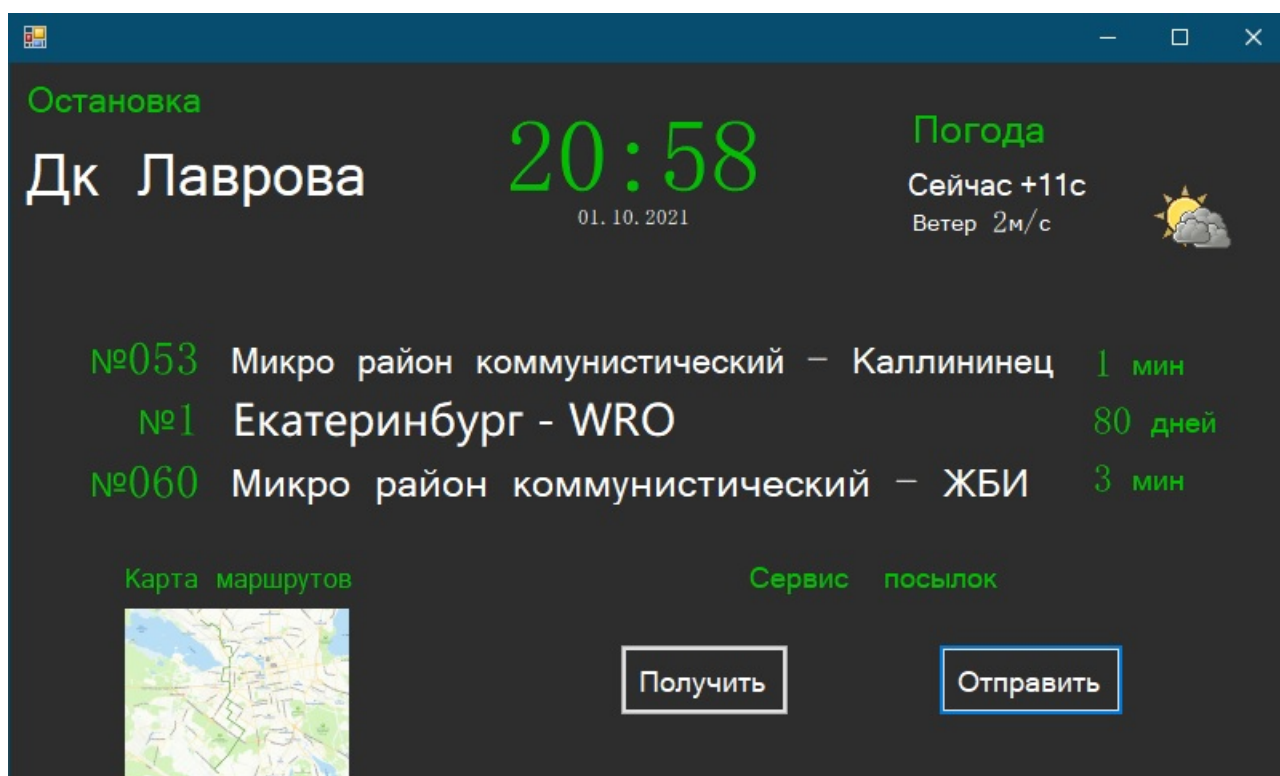


Рисунок 12. Скриншот терминала остановки.