Комитет по образованию Правительства Санкт-Петербурга

Государственное бюджетное профессиональное учреждение среднего профессионального образования

«Санкт-Петербургский Колледж Информационных Технологий»

**КУРСОВОЙ ПРОЕКТ**

По теме: «Проектирование нестандартного периферийного устройства»

По дисциплине: МДК 03.01 Техническое обслуживание и ремонт компьютерных систем и комплексов.

Специальность: 09.02.01

«**КОМПЬЮТЕРНЫЕ СИСТЕМЫ И КОМЛЕКСЫ»**

Преподаватель: Некрасов Д.В.

Выполнил: Тимченко Д.А.

Студент группы № 481

Подпись преподавателя\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2020

**Оглавление**

[Задание 2](#_heading=h.gjdgxs)

[Введение 2](#_heading=h.30j0zll)

[1. Проектирование периферийного устройства 3](#_heading=h.1fob9te)

[1.1 Техническое задание 3](#_heading=h.3znysh7)

[1.2 Этапы проектирования 3](#_heading=h.2et92p0)

[2. Макетирование устройства 5](#_heading=h.3dy6vkm)

[2.1 Arduino Mega 5](#_heading=h.1t3h5sf)

[2.1.1 Общие сведения 5](#_heading=h.4d34og8)

[2.1.2 Характеристики отладочной платы Arduino Mega 2560 6](#_heading=h.2s8eyo1)

[2.1.3 Питание 9](#_heading=h.17dp8vu)

[2.1.4 Память 10](#_heading=h.3rdcrjn)

[2.1.5 Входы и Выходы 10](#_heading=h.26in1rg)

[2.2 Инфракрасный порт KY-022 12](#_heading=h.lnxbz9)

[2.3 Датчик освещённости LM393 14](#_heading=h.35nkun2)

[2.4 Нагрузка 15](#_heading=h.1ksv4uv)

[2.4.1 Коллекторный двигатель постоянного тока 15](#_heading=h.44sinio)

[2.4.2 Драйвер L298N 17](#_heading=h.2jxsxqh)

[2.4.3 Характеристики драйвера мотора L298N 18](#_heading=h.z337ya)

[3 Реализация проекта 19](#_heading=h.3j2qqm3)

[3.1 Управление коллекторным двигателем с помощью инфракрасного сигнала и датчика освещённости 19](#_heading=h.1y810tw)

[3.2 Написание управляющей программы в Arduino IDE 22](#_heading=h.4i7ojhp)

[3.3 Построение виртуального прибора в LabView 26](#_heading=h.2xcytpi)

[Вывод 28](#_heading=h.1ci93xb)

[Источники 29](#_heading=h.3whwml4)

1

# Задание

Разработка нестандартного периферийного устройства по индивидуальному заданию.

# Введение

В данной курсовой работе по модулю МДК 03.01 «Техническое обслуживание и ремонт компьютерных систем и комплексов» рассматривается взаимодействие различных профессиональных модулей: МДК 02.01 «Микропроцессорные системы», МДК 01.02 «Проектирование цифровых устройств», МДК 02.02 «Установка и конфигурирование периферийного оборудования».

Периферийное устройство — аппаратура, которая позволяет вводить [информацию](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) в компьютер или выводить её из него.

Периферийные устройства являются необязательными для работы системы и могут быть отключены от компьютера. Однако большинство компьютеров используются вместе с теми или иными периферийными устройствами.

Также, периферийные устройства делятся на устройства ввода, вывода и хранения.

Устройства ввода - устройства, использующиеся для ввода информации в ПК: клавиатура, мышь, графический планшет.

Устройства вывода - устройства, служащие для вывода информации из ПК: монитор, наушники, акустическая система.

Устройства хранения - устройства, служащие для хранения информации, которую обрабатывает ПК: жесткий диск, USB-флеш-накопитель.

2

# 1. Проектирование периферийного устройства

## 1.1 Техническое задание

Данная курсовая работа направлена на разработку нестандартного периферийного устройства, которое состоит из отладочной платы Arduino Mega 2560, датчика освещенности с фоторезистором LM393 и ИК приёмник KY-022. Необходимо разработать данное устройство, собрать его и запрограммировать в программе Arduino IDE. Также необходимо создать виртуальный прибор в программе LabView и подключить к нему созданное периферическое устройство.

## 1.2 Этапы проектирования

Для выполнения технического задания, необходимо выполнить следующие этапы проектирования:

1) Выбор компонентов, входящих в состав устройства:

Для создания периферийного устройства нам потребуется: мощная нагрузка (коллекторный двигатель), датчик освещённости (датчик освещённости с фоторезистором LM393), инфракрасный приёмник (ИК приёмник KY-022) и управляющее устройство (Arduino Mega 2560).

2) Разработка структурной схемы и блок-схемы устройства:

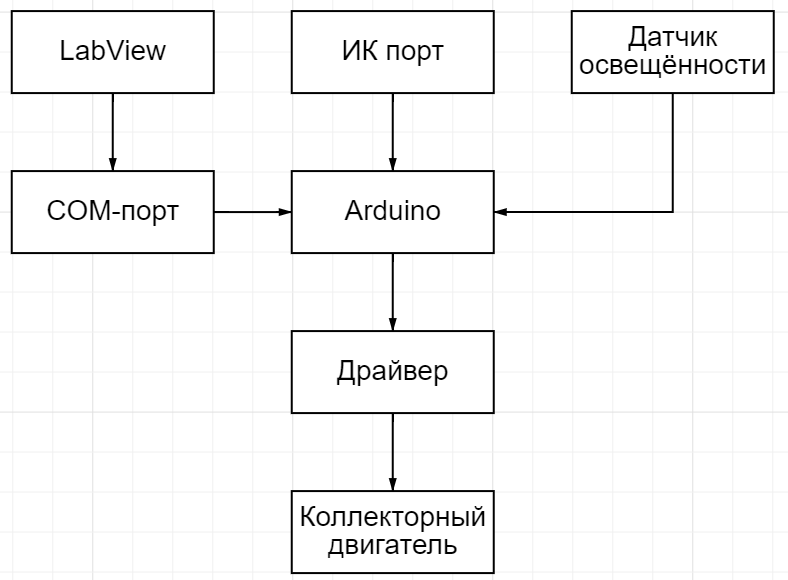


Рисунок 1 – Структурная схема периферийного устройства.

3

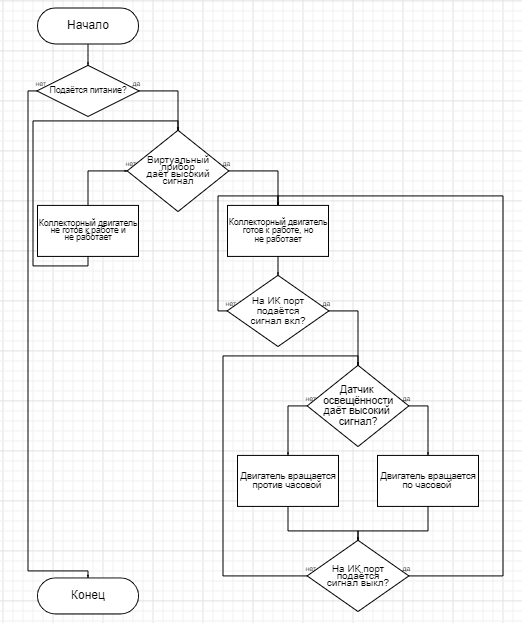


Рисунок 2 – Блок-схема периферийного устройства

3) Разработка алгоритма работы устройства:

С помощью пульта подаётся инфракрасный сигнал Вкл. или Выкл. на ИК порт. После чего управляющее устройство считывает данные с датчика освещённости и отправляет их на драйвер коллекторного двигателя. Так же нужно создать второй канал связи с помощью LabView и подключить его к управляющему устройству (в случае с Arduino, через COM порт).

4) Разработка программного обеспечения для устройства:

Необходимо написать код для Arduino Mega в программе Arduino IDE, затем собрать виртуальный прибор в программе LabView. Программа будет обеспечивать взаимосвязь датчиков и коллекторного двигателя через драйвер.

4

# 2. Макетирование устройства

## 2.1 Arduino Mega

Основой разрабатываемого нестандартного периферийного устройства является отладочная плата Arduino Mega (Рисунок 2).

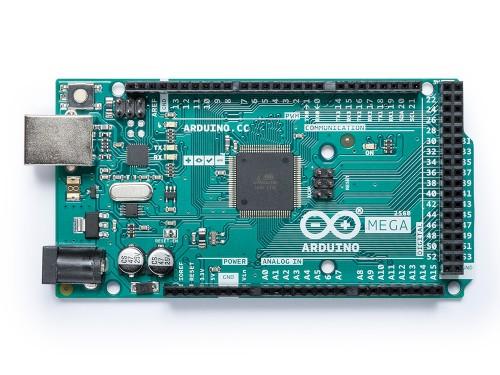


Рисунок 2 – Отладочная плата Arduino Mega 2560.

### 2.1.1 Общие сведения

Arduino Mega построена на микроконтроллере ATmega2560 (техническое описание). Плата имеет 54 цифровых входа/выходов (14 из которых могут использоваться как выходы ШИМ), 16 аналоговых входов, 4 последовательных порта UART, кварцевый генератор 16 МГц, USB коннектор, разъем питания, разъем ICSP и кнопка перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB или подать питание при помощи адаптера AC/DC, или аккумуляторной батареей. Arduino Mega 2560 совместима со всеми платами расширения, разработанными для платформ Uno или Duemilanove.

5

### 2.1.2 Характеристики отладочной платы Arduino Mega 2560

На таблице 1 показаны технические параметры отладочной платы Arduino 2560.

Таблица 1 – Технические параметры отладочной платы Arduino Mega 2560

| Микроконтроллер | ATmega2560 |
| --- | --- |
| Рабочее напряжение | 5 В |
| Входное напряжение (рекомендуемое) | | 7-12 В | | --- | |  | |
| Входное напряжение (предельное) | 6-20 В |
| Цифровые Входы/Выходы | 54  (14 из которых могут работает также как выходы ШИМ) |
| Аналоговые входы | 16 |
| Постоянный ток через вход/выход | 40 мA |
| Постоянный ток для вывода 3.3 В | 50 мA |
| Флеш-память | 256 KB  (из которых 8 КB используются для загрузчика) |
| ОЗУ | 8 KB |
| Энергонезависимая память | 4 KB |
| Тактовая частота | 16 МГц |

Также нам предоставлена подробная электрическая принципиальная схема отладочной платы Arduino Mega 2560 (см. рисунок 3).

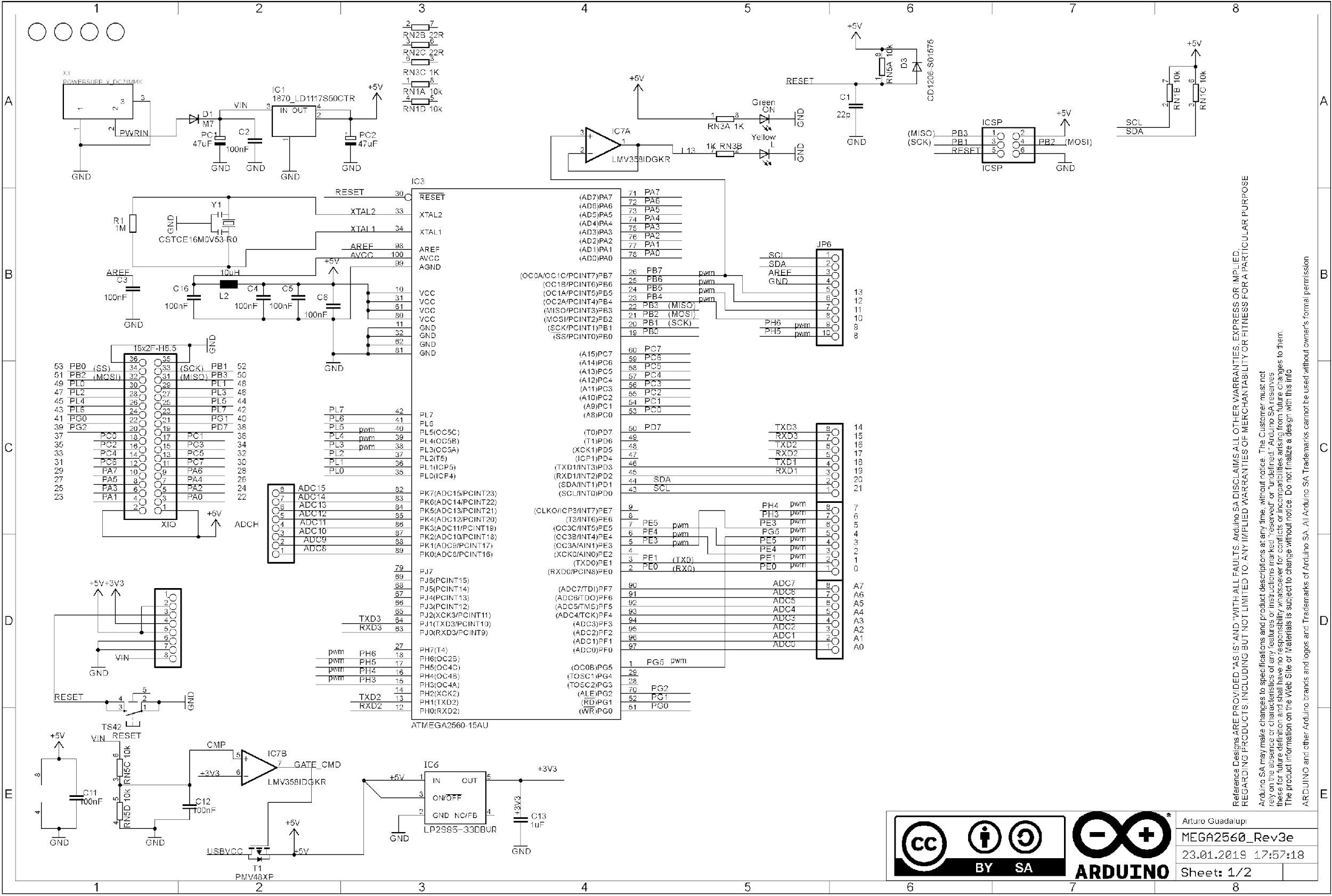


Рисунок 3 - Электрическая принципиальная схема отладочной платы Arduino Mega 2560

6

Для подключения устройств к Arduino, необходимо знать назначение и расположение выводов, для правильного подключения.

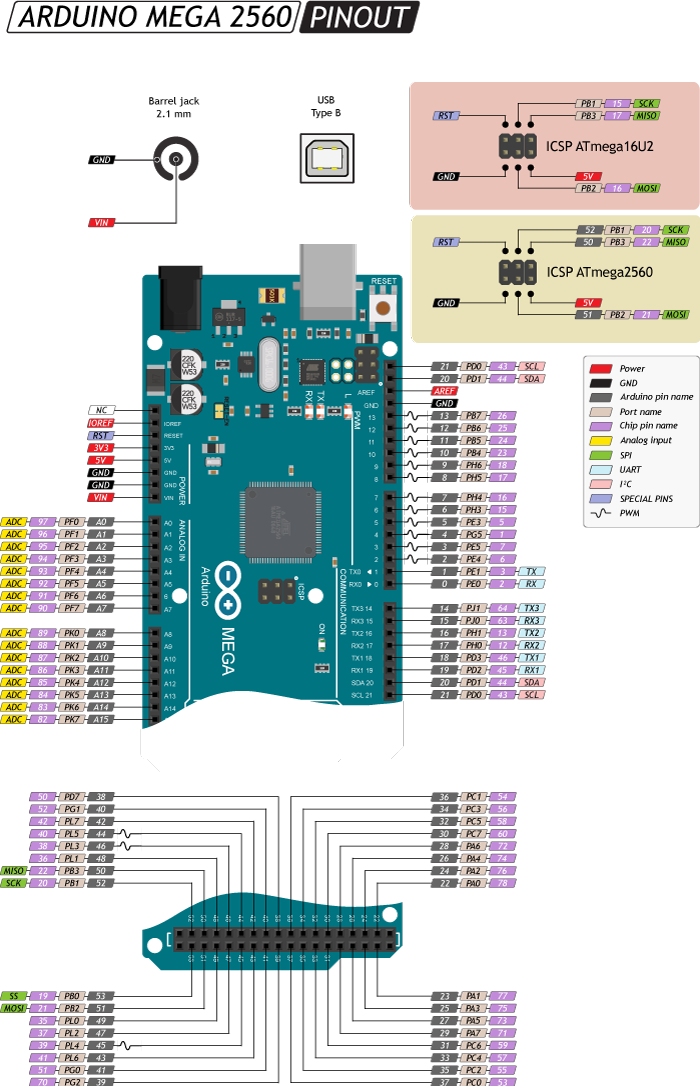


Рисунок 4 – Расположение выводов Arduino Mega 2560

7

### 2.1.3 Питание

Arduino Mega может получать питание как через подключение по USB, так и от внешнего источника питания. Источник питания выбирается автоматически.

Внешнее питание (не USB) может подаваться через преобразователь напряжения AC/DC (блок питания) или аккумуляторной батареей. Преобразователь напряжения подключается посредством разъема 2.1 мм с  положительным полюсом на центральном контакте. Провода от батареи подключаются к выводам Gnd и Vin разъема питания (POWER).

Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

Плата Mega2560, в отличие от предыдущих версий плат, не использует FTDI USB микроконтроллер. Для обмена данными по USB используется микроконтроллер Atmega8U2, запрограммированный как конвертер USB-to-serial.

Выводы питания:

**VIN**. Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого регулируемого источника питания). Подача напряжения питания происходит через данный вывод. Если питание подается на разъем 2.1mm, то на этот вход можно ёзапитаться.

**5V**. Регулируемый источник напряжения, используемый для питания микроконтроллера и компонентов на плате. Питание может подаваться от вывода VIN через регулятор напряжения, или от разъема USB, или другого регулируемого источника напряжения 5 В.

**3V3.**Напряжение на выводе 3.3 В генерируемое микросхемой FTDI на платформе. Максимальное потребление тока 50 мА.

**GND.**Выводы заземления.

8

### **2.1.4 Память**

Микроконтроллер ATmega2560 имеет: 256 кБ флеш-памяти для хранения кода программы (4 кБ используется для хранения загрузчика), 8 кБ ОЗУ и 4 Кб EEPROM (которая читается и записывается с помощью [библиотеки EEPROM](http://arduino.ru/Reference/Library/EERPOM)).

### 2.1.5 Входы и Выходы

Каждый из 54 цифровых выводов Mega, используя функции [pinMode()](http://arduino.ru/Reference/PinMode), [digitalWrite()](http://arduino.ru/Reference/DigitalWrite), и[digitalRead()](http://arduino.ru/Reference/DigitalRead), может настраиваться как вход или выход. Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (стандартно отключен) 20-50 кОм и может пропускать до 40 мА. Некоторые выводы имеют особые функции:

**Последовательная шина: 0 (RX) и 1 (TX); Последовательная шина 1: 19 (RX) и 18 (TX); Последовательная шина 2: 17 (RX) и 16 (TX);**Последовательная шина 3: 15 (RX) и 14 (TX). Выводы используются для получения (RX) и передачи (TX) данных TTL. Выводы 0 и 1 подключены к соответствующим выводам микросхемы последовательной шины ATmega8U2.

**Внешнее прерывание: 2 (прерывание 0), 3 (прерывание 1), 18 (прерывание 5), 19 (прерывание 4), 20 (прерывание 3), и 21 (прерывание 2).**Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции attachInterrupt().

**PWM: 2 до 13 и 44-46.** Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции [analogWrite()](http://arduino.ru/Reference/AnalogWrite).

**SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).**Посредством данных выводов осуществляется связь SPI, например, используя [библиотеку SPI](http://arduino.ru/Reference/Library/SPI). Также выводы SPI могут быть выведены на блоке ICSP, который совместим с платформами Uno, Duemilanove и Diecimila.

9

**LED: 13.** Встроенный светодиод, подключенный к цифровому выводу 13. Если значение на выводе имеет высокий потенциал, то светодиод горит.

**I2C: 20 (SDA) и 21 (SCL).** Посредством выводов осуществляется связь I2C (TWI). Для создания используется [библиотека Wire](http://out.arduino.ru/?redirect=http%3A%2F%2Fwiring.org.co%2Freference%2Flibraries%2FWire%2Findex.html&baseU=http%3A%2F%2Farduino.ru%2FHardware%2FArduinoBoardMega2560) (информация на сайте Wiring). Расположение выводов на платформе Mega не соответствует расположению Duemilanove или Diecimila.

На платформе Mega2560 имеется 16 аналоговых входов, каждый разрешением 10 бит (т.е. может принимать 1024 различных значения). Стандартно выводы имеют диапазон измерения до 5 В относительно земли, тем не менее имеется возможность изменить верхний предел посредством вывода AREF и функции analogReference().

Дополнительная пара выводов платформы:

**AREF.** Опорное напряжение для аналоговых входов. Используется с функцией [analogReference()](http://arduino.ru/Reference/AnalogReference).

**Reset.** Низкий уровень сигнала на выводе перезагружает микроконтроллер. Обычно применяется для подключения кнопки перезагрузки на плате расширения, закрывающей доступ к кнопке на самой плате Arduino.

10

## 2.2 Инфракрасный порт KY-022

Инфракрасный приемник - простой и недорогой вариант для приема инфракрасного сигнала. Идеально подходит для дистанционного управления устройствами в зоне прямой видимости на небольшом расстоянии. Позволяет принимать сигналы с пультов управления различной бытовой техники большинства производителей.

Модуль инфракрасного приемника удобен для использования в проектах для соединения без пайки. Модуль содержит светодиод - индикатор получения данных.



Рисунок 5 - Инфракрасный порт KY-022

Модуль “KY-022” ИК-приемника выполнен на базе датчика “VS1838B” (рабочая частота 38 кГц) и предназначен для приема данных по инфракрасному каналу от пультов дистанционного управления телевизором и от другого оборудования в доме.

Таблица 2 – Технические параметры инфракрасного порта KY-022

| Потребляемый ток | 0,5 мА |
| --- | --- |
| Рабочая частота | 38 КГц |
| Расстояние приема сигнала | 20 м |
| Эффективный угол приема сигнала | 90˚ |
| Рабочее напряжение (приемник) | 2,7 - 5,5 В |
| Рабочее напряжение (модуль) | 2,7 - 5,5 В |

11

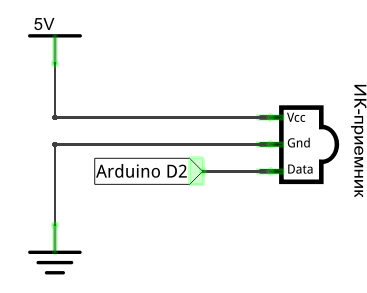


Рисунок 6 – Принципиальная схема инфракрасного порта KY-022

Инфракрасный датчик подключается к Arduino по следующей схеме (см. рисунок 7).

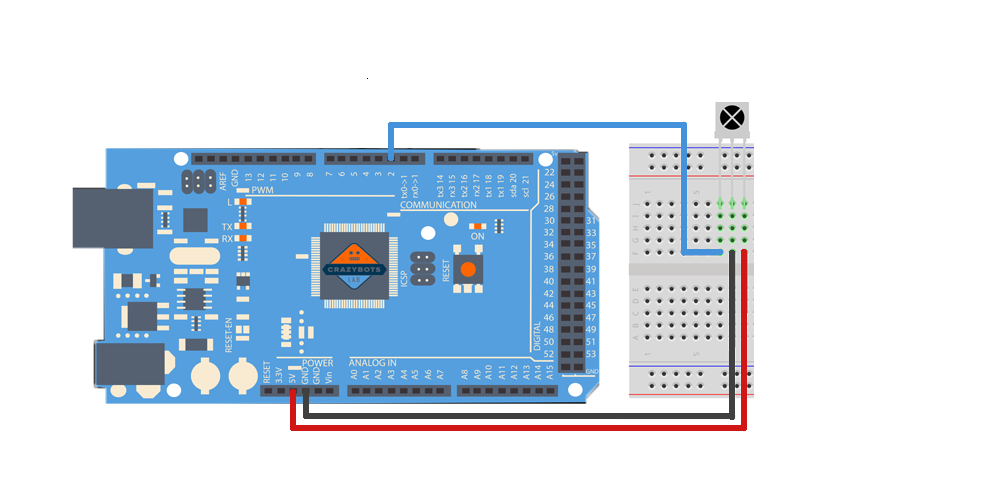


Рисунок 7 – Схема подключения к Arduino

12

## 2.3 Датчик освещённости LM393

Модуль освещенности на LM393, используется для измерения интенсивности света в различных устройствах, таких как, автоматизация света, роботах и приборов контролирующих уровень освещенности. Измерения осуществляется с помощью светочувствительного элемента, который меняет сопротивление в зависимости от освещенности.



Рисунок 8 – Датчик освещённости LM393 с фоторезистором

Таблица 3 – Технические параметры освещённости LM393

| Напряжение питания | 3.3 В ~ 5.5 В |
| --- | --- |
| Потребляемый ток | 10 мА |
| Цифрового выход | TTL (лог 1 или лог 0) |
| Аналогового выход | 0 В … Vcc |
| Диаметр монтажного отверстия | 2.5 мм |
| Выходной ток | 15 мА |
| Габариты | 42мм х 15мм х 8мм |

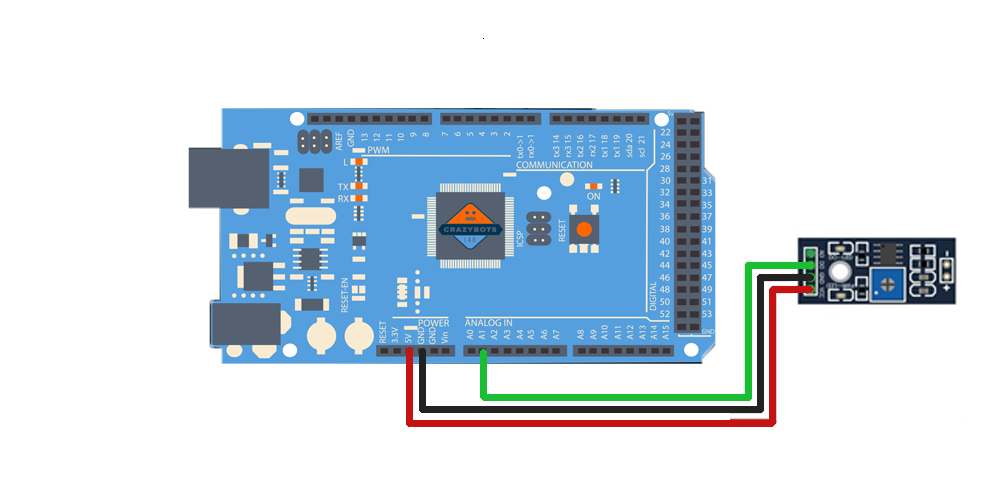


Рисунок 9 – Схема подключения к Arduino

13

## 2.4 Нагрузка

### 2.4.1 Коллекторный двигатель постоянного тока

**Коллекторный электродвигатель** — электрическая машина, в которой датчиком положения ротора и переключателем тока в обмотках является одно и то же устройство — щёточно-коллекторный узел.

**Щёточно-коллекторный узел** — узел электрической машины, обеспечивающий электрическое соединение цепи ротора с цепями, расположенными в неподвижной части машины. Состоит из коллектора (набора контактов, расположенных на роторе) и щёток (скользящих контактов, расположенных вне ротора и прижатых к коллектору).

В коллекторном электродвигателе щёточно-коллекторный узел одновременно выполняет две функции:

* является датчиком углового положения ротора (датчик угла) со скользящими контактами;
* является переключателем направления тока со скользящими контактами в обмотках ротора в зависимости от углового положения ротора.

**Принцип работы:**

При подаче питающего напряжения ток проходит сквозь обмотку возбуждения. У смежных полюсов возникает противоположная полярность, из-за чего образуется своеобразный магнит с постоянным магнитным полем. На якорь двигателя через коллектор подаётся постоянный ток, на который со стороны магнитного поля статора действует электромагнитная индукция. В результате создаётся вращающий момент, поворачивающий ротор на 90 электрических градусов. После этого щёточно-коллекторный узел коммутирует обмотки ротора, и вращение продолжается.

14



Рисунок 10 – Простой коллекторный двигатель

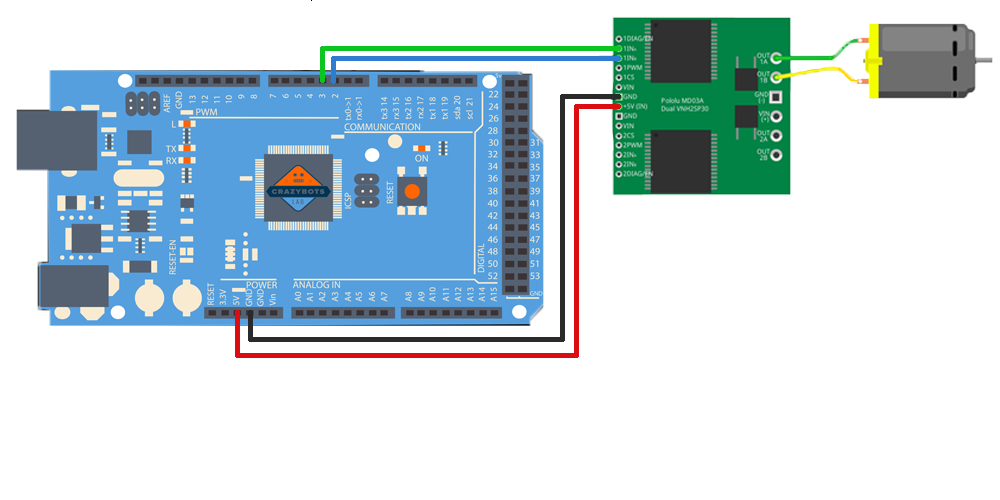


Рисунок 11 – Схема подключения коллекторного двигателя через драйвер

15

### 2.4.2 Драйвер L298N

Напрямую подключить электродвигатель к выводам Arduino нельзя, есть риск сжечь вывод, к которому подключён двигатель. Для безопасного подключения электродвигателей разных типов к Arduino необходим драйвер двигателей. Драйверы бывают разные, для их работы часто используются микросхемы типа HG788, L9110S, L293D, L298N и другие. Драйверы двигателей имеют выводы подачи питания, выводы для подключения электродвигателей, а также управляющие выводы.

Драйвер L298N используется радиолюбителями для многофункционального управления двигателями постоянного тока. Схема модуля, состоящая из двух H-мостов, позволяет подключать к нему один биполярный шаговый двигатель или одновременно два щёточных двигателя постоянного тока. При этом есть возможность изменять скорость и направление вращения моторов. Управление осуществляется путём подачи соответствующих сигналов на командные входы, выполненные в виде штыревых контактов.

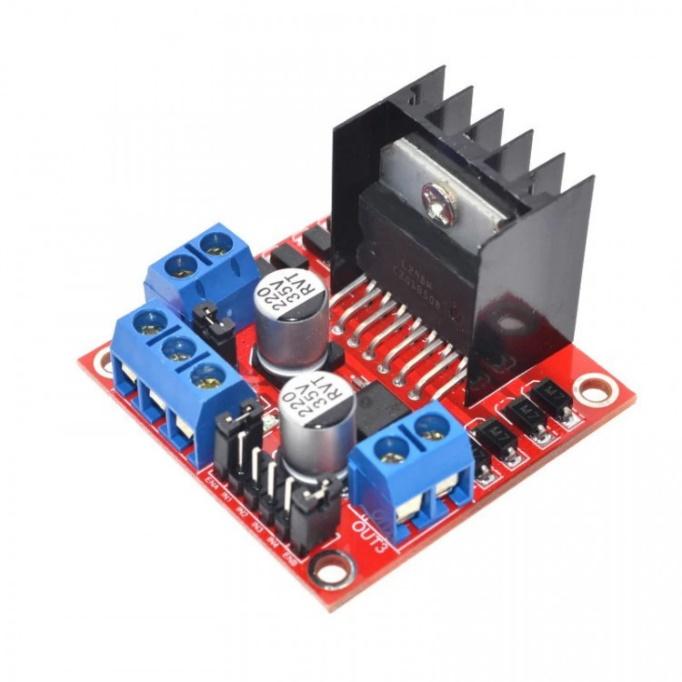


Рисунок 12 – Драйвер моторов L298N

16

### 2.4.3 Характеристики драйвера мотора L298N

Таблица 4 – Технические параметры драйвера L298N

| Напряжение питания логики модуля | 5 В |
| --- | --- |
| Потребляемый ток встроенной логики | 36 мА |
| Напряжение питания драйвера | 5 В – 35 В |
| Рабочий ток драйвера | 2 А (пиковый ток 3 А) |
| Габариты | 43.5 мм х 43.2мм х 29.4мм |

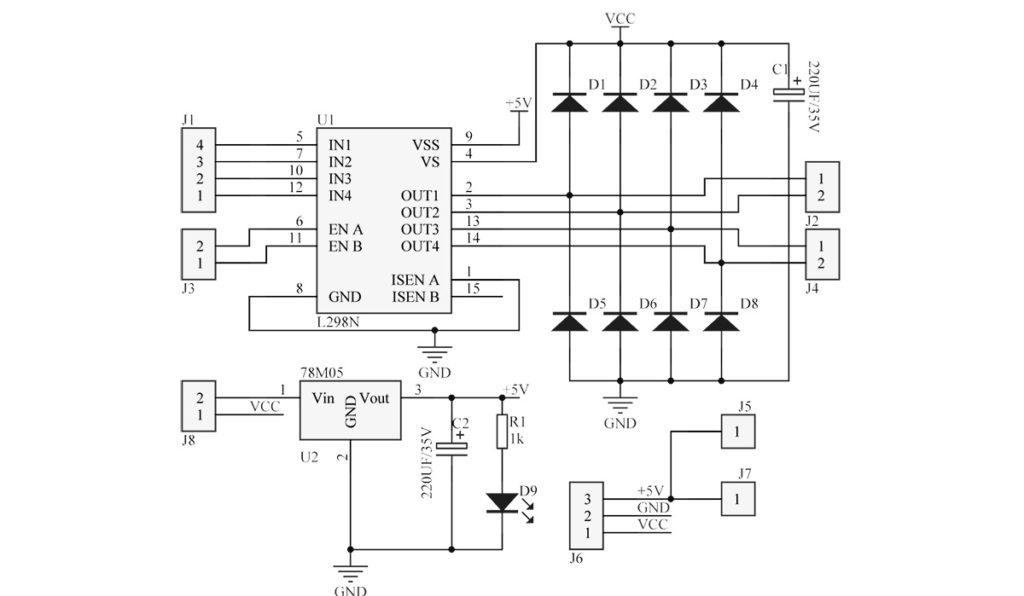


Рисунок 13 - Принципиальная схема модуля L298N

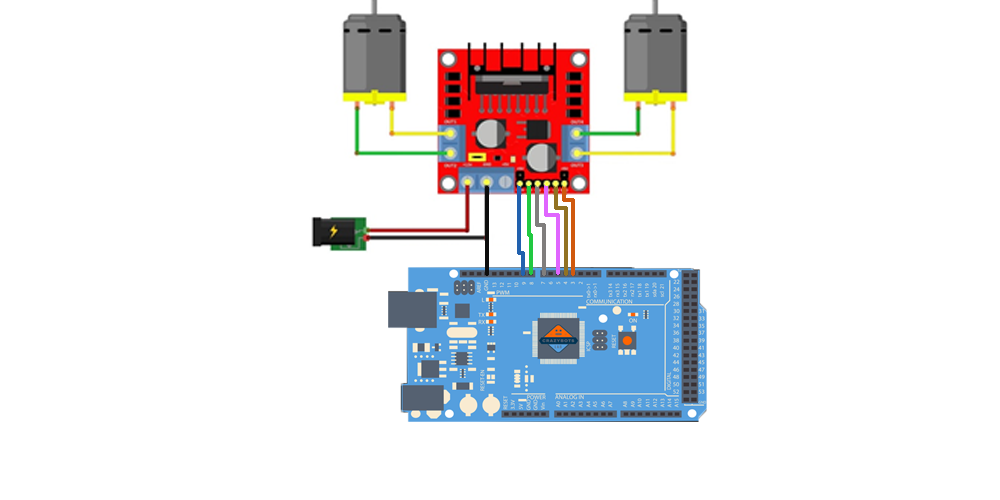
****

Рисунок 14 – Схема подключения драйвера с коллекторными двигателями к Arduino

17

# 3 Реализация проекта

## 3.1 Управление коллекторным двигателем с помощью инфракрасного сигнала и датчика освещённости

Для управления коллекторным двигателем нам необходимо подключить его к драйверу L298N, который, в свою очередь, необходимо подключить к Arduino (см. рисунок 14). Входные сигналы мы будем подавать с помощью, подключённых к Arduino (см. рисунок 7 и рисунок 9), инфракрасного сигнала, приходящий в инфракрасный порт, а регулировать направление вращения с помощью датчика освещённости.

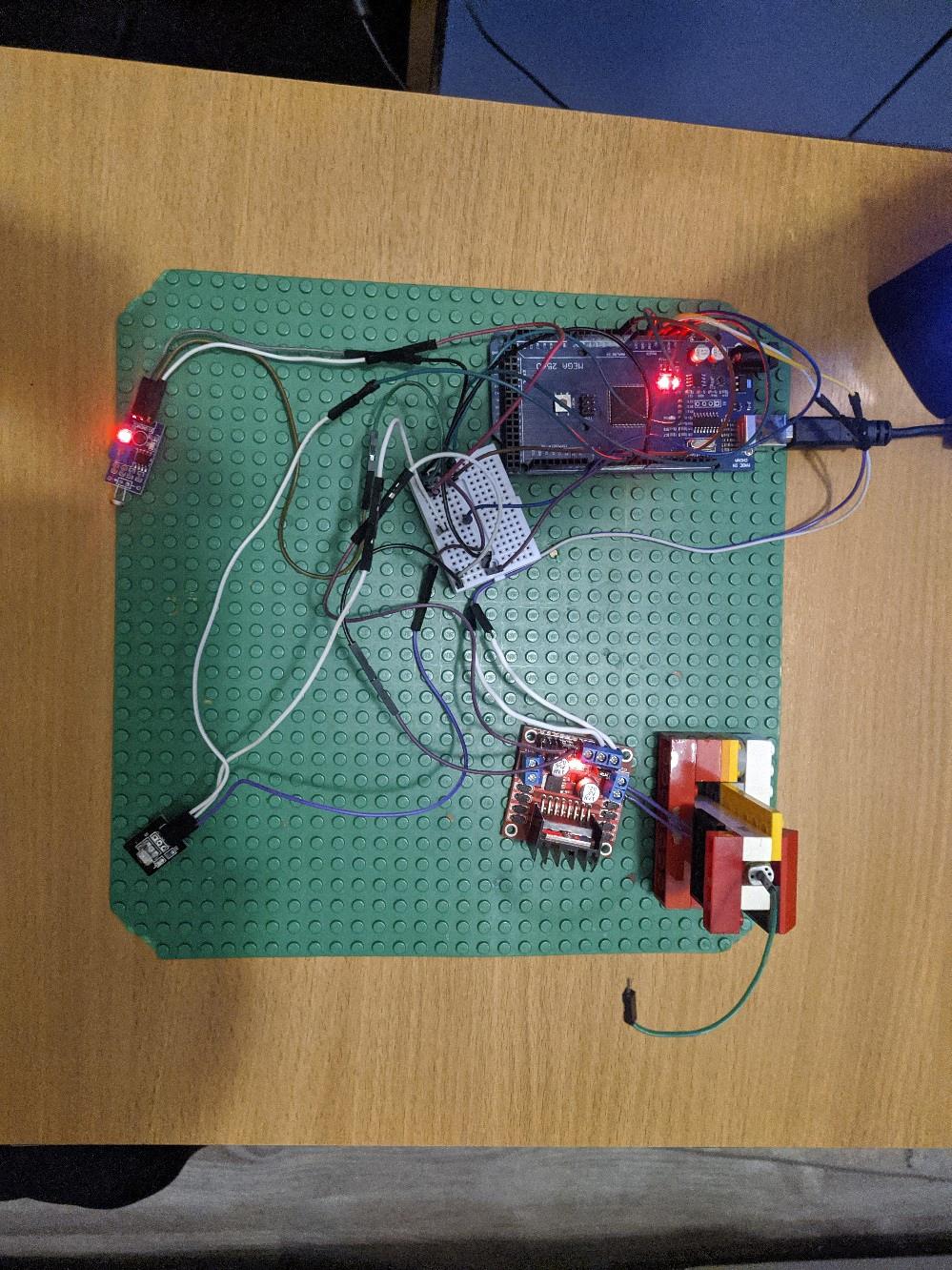


Рисунок 15 – Собранное периферийное устройство

При подаче инфракрасного сигнала под кодом 0xFF и при низком сигнале с датчика освещённости, двигатель должен будет вращаться по направлению часовой стрелки (см. рисунок 16).

18

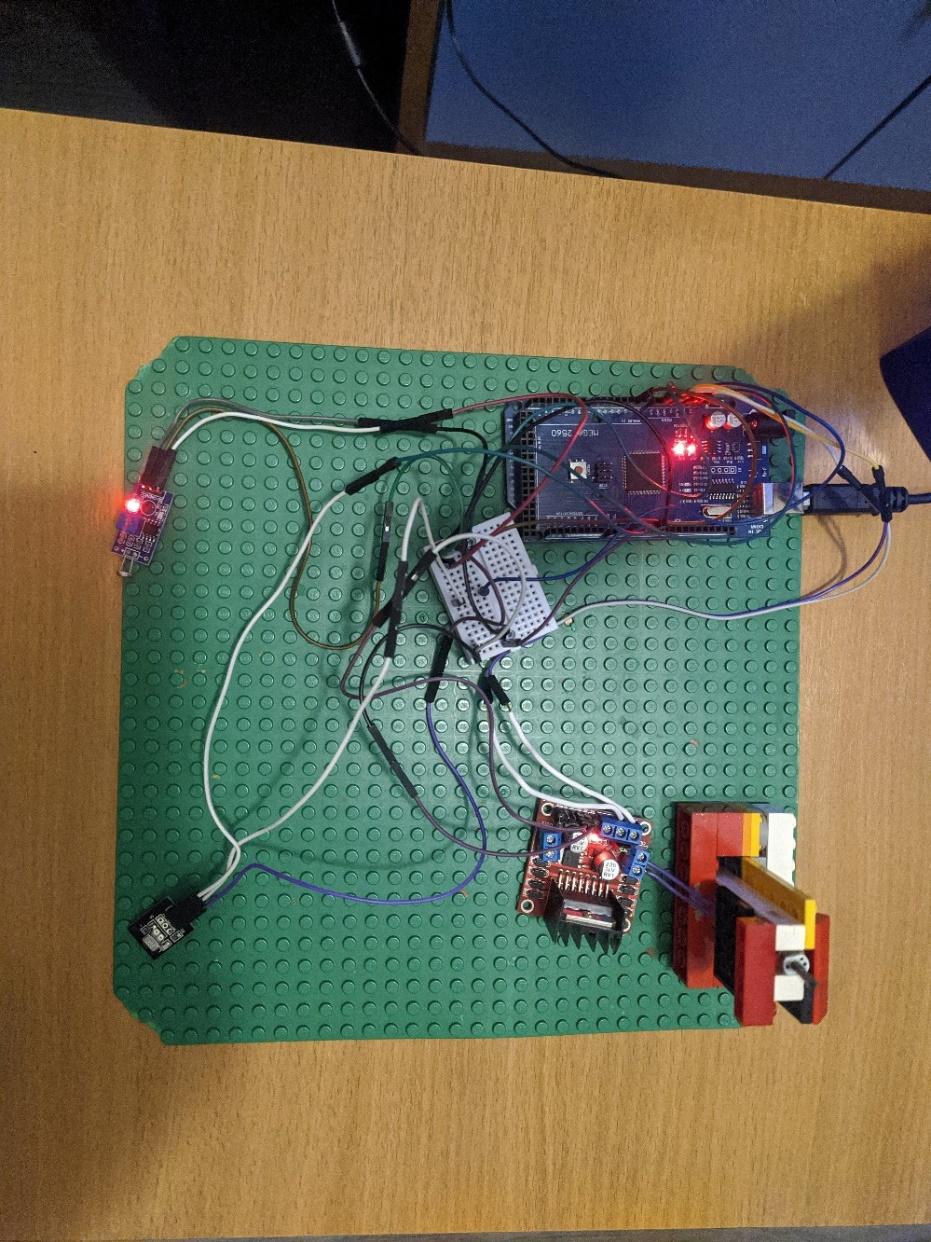


Рисунок 16 – Работа двигателя при низком сигнале с датчика освещённости

Если же датчик освещённости выдаёт высокий сигнал, что можно узнать по зелёному светодиоду на датчике, то при повторном нажатии на кнопку подачи инфракрасного сигнала, двигатель остановится (см. рисунок 17), после чего начнёт вращаться в противоположную сторону (см. рисунок 18).

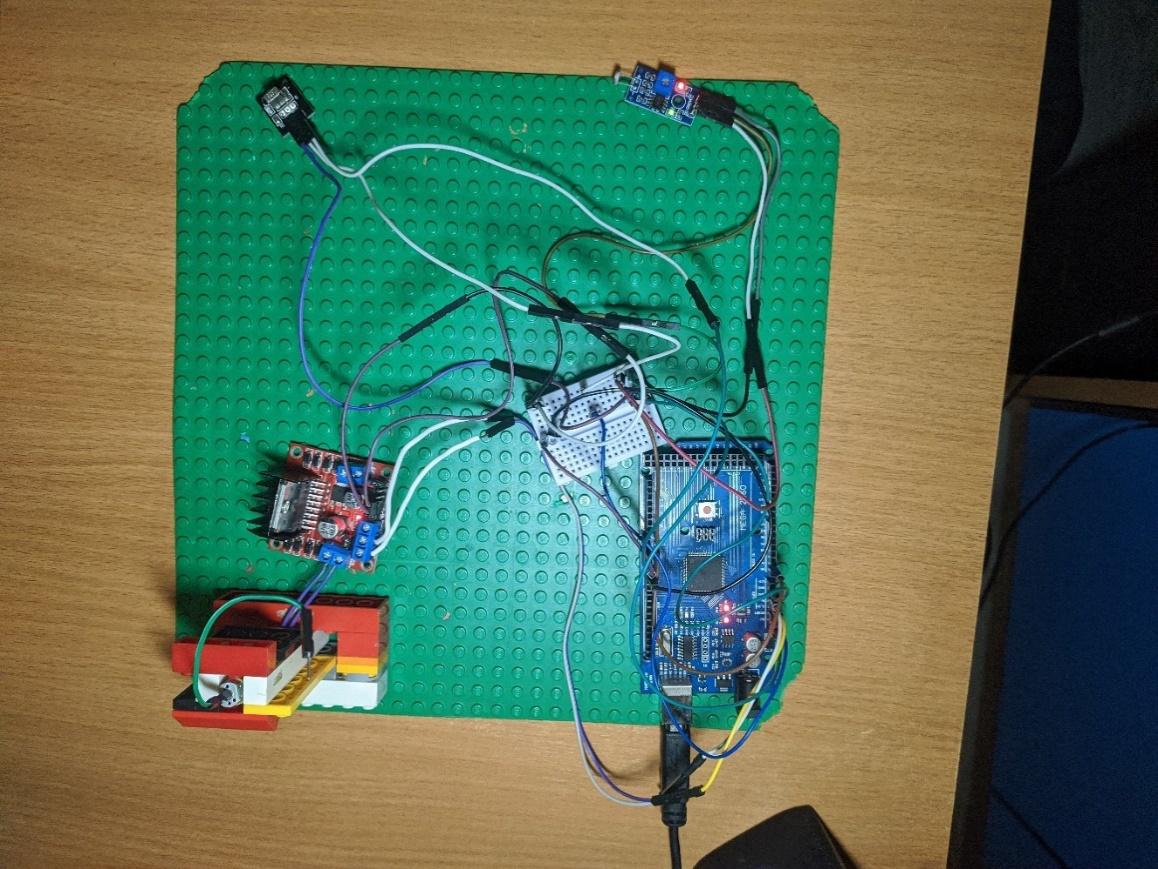


Рисунок 17 – Выключенное устройство с датчиком освещённости, подающий высокий сигнал

19

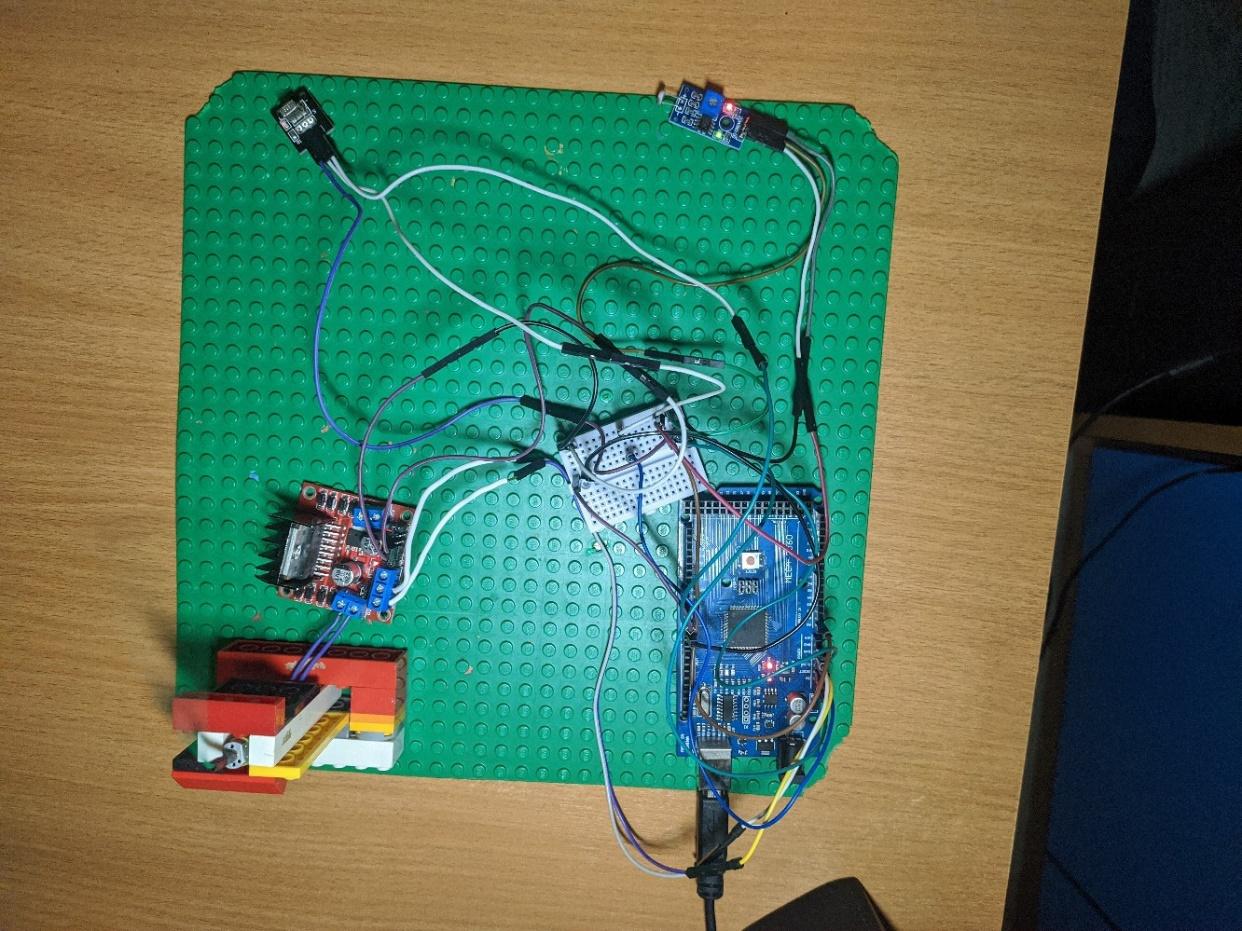


Рисунок 18 – Работа двигателя при высоком сигнале с датчика освещённости

Для выключения устройства требуется нажать на кнопку на пульте, которая подаст инфракрасный сигнал, под кодом 0x20DF, на инфракрасный приёмник.

Программа для работы Arduino, написана в рабочей среде Arduino IDE и дополнительно управляется с помощью виртуального прибора, построенного в LabView.

20

## 3.2 Написание управляющей программы в Arduino IDE

Для создания программы для Arduino используется среда разработки Arduino IDE (см. рисунок 19).

**Arduino IDE** — интегрированная среда разработки для Windows, MacOS и Linux, разработанная на Си и C ++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей.

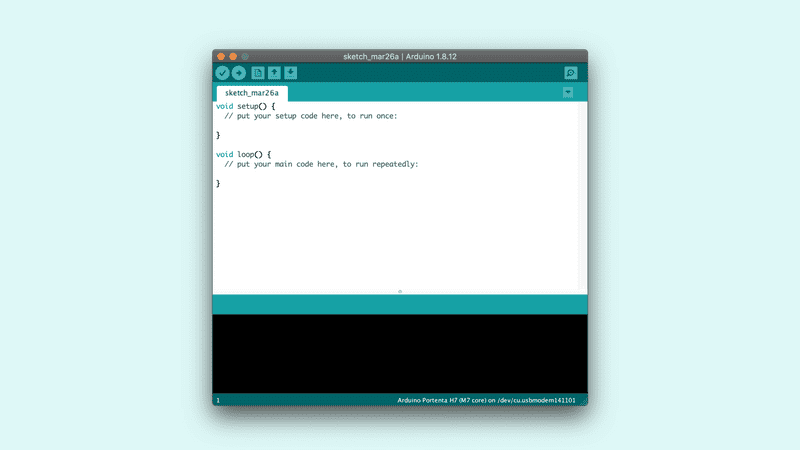
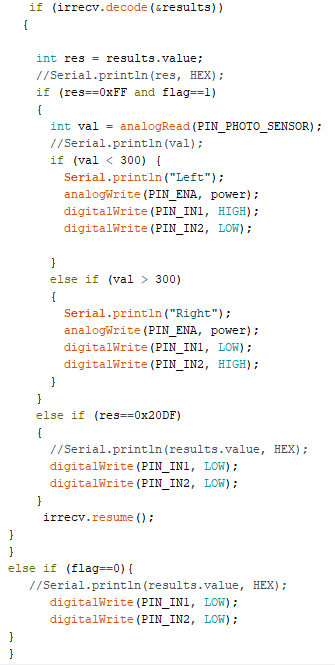
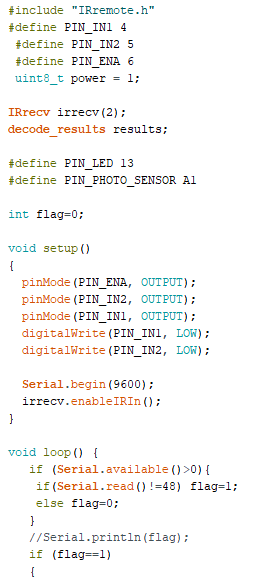


Рисунок 19 – Начальный экран Arduino IDE

20

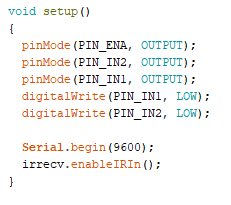
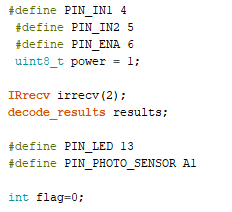
В данной программе была написана управляющая программа для разрабатываемого периферийного устройства (см. рисунки 20, 21). Для написания данной программы использовалась библиотека IRremote, для получения и использования кодов с кнопок пульта для инфракрасного порта.



Рисунки 20, 21 – Программа управления

21

Изначально необходимо установить переменные для выходов Arduino (см. рисунок 22), а также присвоить переменным power и flag значения 1 и 0 соответственно. Затем надо установить режимы работы выходов для драйвера (см. рисунок 23), а также установить скорость обмена данных с помощью Serial.begin.



Рисунки 22 / 23 – Программа присвоения переменных / Программа установки режимов работы выходов

Основная часть программы начинается с проверки сигнала из виртуального прибора LabView и, за счёт этого сигнала, присваивание значение 1 или 0 переменной flag (см. рисунок 24).

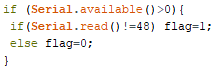


Рисунок 24 – Программа проверки сигнала из виртуального прибора LabView

Если переменной flag присвоен сигнал 1, то программа начинает считывать значения кодов, поданные на инфракрасный порт с помощью пульта (см. рисунок 25). Если подан сигнал под кодом 0xFF, то на драйвер поддадутся три сигнала переменных (см. рисунок 26): PIN\_IN1, PIN\_IN2 (сигналы пойдёт на контакты управления первым коллекторным двигателем), которые буду регулировать направление вращения двигателя, а также PIN\_ENA (сигнал пойдёт на контакт разрешения работы первого коллекторного двигателя).

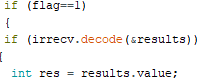


Рисунок 25 – Программа обработки сигналов и присвоения их кодов переменной res

22

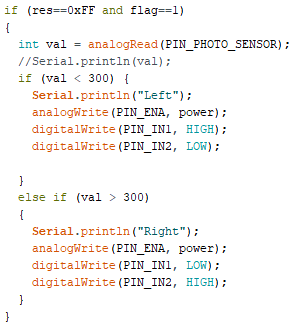


Рисунок 26 – Программа управления коллекторным двигателем

Направление вращения двигателя регулируется переменной val, значение которой изменяется в зависимости от сигнала датчика освещённости, который считывается с выхода, которому присвоена переменная PIN\_PHOTO\_SENSOR, с помощью функции analogRead (см. рисунок 26).

Если же с пульта подан сигнал под кодом 0x20DF, то на PIN\_IN1 и PIN\_IN2 поддадутся два низких сигнала, благодаря чему вращение двигателя остановится (см. рисунок 27).

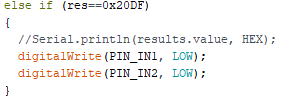


Рисунок 27 – Программа остановки двигателя

И конечно же программа не будет работать, если переменной flag будет присвоено значение 0 (см. рисунок 28).

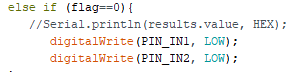


Рисунок 28 – Программа, запрещающая работу устройства

23

## 3.3 Построение виртуального прибора в LabView

Как уже было сказано, для начала работы основной части программы, нужно будет присвоить переменной flag значение 1, что будет сделано через виртуальный прибор, созданный в рабочей среде LabView.

**LabVIEW** (**Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench) — это среда разработки и платформа для выполнения программ, созданных на графическом языке программирования «G» фирмы National Instruments (США). Первая версия LabVIEW была выпущена в 1986 году для Apple Macintosh, в настоящее время существуют версии для Unix, Linux, Mac OS и Microsoft Windows. LabVIEW используется в системах сбора и обработки данных, а также для управления техническими объектами и технологическими процессами.

Виртуальный прибор, построенный в рабочей среде LabView (см. рисунок 30), должен будет отвечать за подачу сигнала управления, после подачи которого, устройство будет готова к использованию и управлению через пульт.



Рисунок 29 – Логотип LabView

Для подключения LabView к программе написанной в Arduino IDE, необходимо скачать и установить библиотеку NI-VISA.

**VISA** - высокоуровневый программный интерфейс (API), предназначенное для коммуникаций через инструментальные шины. VISA не зависит ни от платформы, ни от шины, ни от рабочего окружения. Другими словами, используется одно и то же API, независимо от того, создана ли программа для обмена данными между устройством USB и системой LabVIEW.

24

Первым делом, необходимо создать программу, благодаря которой мы сможем подавать сигнал с виртуальной кнопки и считывать это значение в программе Arduino, для присваивания переменой flag значения 1 или 0.

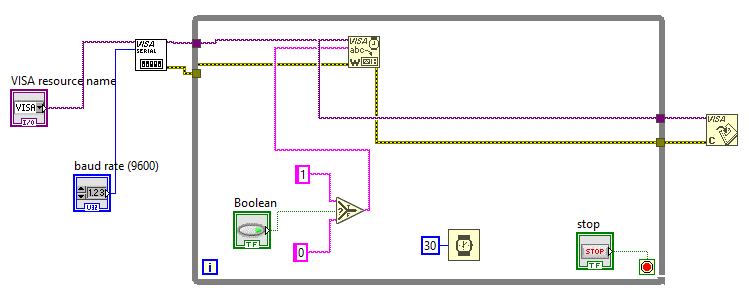


Рисунок 30 – Программа виртуального прибора

А с помощью автоматически созданной визуализации нужно выбрать соответствующий COM-порт и скорость обмена данных, а с помощью кнопки, управлять работой устройства (см. рисунок 31).

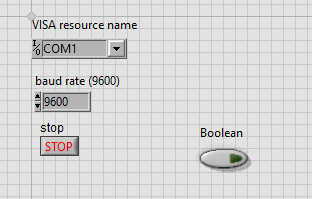


Рисунок 31 – Визуализация виртуального прибора

25

# Вывод

В ходе данной курсовой работы, было спроектировано, собрано и протестировано нестандартное периферийное устройство. Устройство состояло из отладочной платы Arduino Mega 2560, датчика освещенности LM393, инфракрасного приёмника KY-022 и коллекторного двигателя с драйвером L298N. Перед началом работы над устройством, были разработаны этапы проектирования, вместе с блок-схемой и структурной схемой устройства. Так же была разработана программа управления для Arduino в рабочей среде Arduino IDE и виртуальный прибор в программе LabView. После проверки правильности подключения и загрузки программы, периферийное устройство было неоднократно протестировано.

26

# Источники

* <http://arduino.ru/Hardware/ArduinoBoardMega2560>
* [http://wiki.amperka.ru/продукты:arduino-mega-2560](http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:arduino-mega-2560)
* <http://funnydiy.net/1007>
* [https://роботехника18.рф/ик-приемник-ардуино/](about:blank)
* [https://роботехника18.рф/датчик-освещенности-ардуино/](about:blank)
* <https://arduinomaster.ru/datchiki-arduino/photorezistor-arduino-datchik-sveta/>
* <https://robotchip.ru/obzor-modulya-osveshchennosti-lm393/>
* <https://cable.ru/articles/id-1939.php>
* <https://3d-diy.ru/wiki/arduino-moduli/drayver-dvigatelya-l298n/>
* <https://ru.wikipedia.org/wiki/LabVIEW>
* <http://microsin.net/programming/pc/ni-visa-usb-instrument-control-tutorial.html>
* <https://microcontrollerslab.com/program-arduino-labview-example/>
* <https://programforyou.ru/block-diagram-redactor>

27