**Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования**

**«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

ИНСТИТУТ ФИЗИКИ

КАФЕДРА РАДИОАСТРОНОМИИ

Направление: 03.03.03. — «Радиофизика»

КУРСОВАЯ РАБОТА

**Heatmap by WebGazer**

**Работа завершена:**

"\_\_\_"\_\_\_\_\_\_\_\_2020 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(A.A Ямолдин)

**Работа допущена к защите:**

Научный руководитель

ассистент

"\_\_\_"\_\_\_\_\_\_\_\_\_\_\_2020 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(К.?. Скобельцин)

Заведующий кафедрой

канд. физ.-мат. наук, доцент

"\_\_\_"\_\_\_\_\_\_\_\_\_\_\_2020 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(А.Д. Акчурин)

Казань — 2020

**СОДЕРЖАНИЕ**

Введение……………………………………………………………………….….3

ГЛАВА 1. Свободный браузерный айтрекер WebGazer.js……………………..……………………4

1.1. Особенности айтрекера WebGazer.js……………………..……..........…..4

1.2. Достоинства и недостатки айтрекера WebGazer.js……..........……...…10

ГЛАВА 2. Свободная библиотека hetmap.js…………………………………………..…………………………….13

2.1 Особенности библиотеки heatmap.js

2.2 Достоинтсва и недостатки heatmap.js

ГЛАВА 3.ОСНОВНЫЕ СВЕДЕНИЯ О СОЗДАНИИ ОДНОСТРАНИЧНЫХ WEB-ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ФРЕЙМВОРКА ANGULARJS…………………………………………..…………………………….13

2.1. AngularJS и совместимые языки программирования…………………….13

2.2. Среда разработки WebStorm IDE……………………………………….….14

2.3. Структура AngularJS приложений в среде WebStorm IDE……………………………...15

2.4.Библиотеки и функции……………………………………………...…….16

2.4.1. Библиотеки………………………………………………………….16

2.4.2. Функции…………………………………………………………….18

ГЛАВА 3. ПРАКТИЧЕСКАЯ ЧАСТЬ…………………………..….…………..21

3.1. Создание AngularJS приложения ……………………………21

3.2. Запуск айтрекера WebGazer.js……………………………………...……………22

3.3. Обмен данными между скриптами внутри web-приложения …………………………………………………26

3.3 Создание тепловой карты с помощью heatmap.js

3.4 Деплой приложения на хостинг GitHub Pages

Заключение…………………………………………………………….……..…..30

Список использованных источников………………………………….…….….31

Приложение………………………………………………………………………32

**ВВЕДЕНИЕ**

При разработке современных web-приложений часто требуется сбор данных о действиях пользователя в режиме реального времени. В настоящий момент для сбора данных существует несколько готовых решений, такие как JavaScript-библиотека TensorFlow(https://www.tensorflow.org/js/), JavaScript-библиотека WebGazer.js(https://webgazer.cs.brown.edu/#home). При работе таких скриптов часто требуется передача цифровых данных с различных датчиков, либо приборов на персональный компьютер. Это можно сделать по различным проводным и беспроводным цифровым интерфейсам, например Ethernet, Wi-fi, Bluetooth, USB. Каждый из них требует отдельного подхода и специализированного оборудования. Актуальным является применение интерфейса Ethernet в связи с его высокой стабильностью и скоростью передачи данных.

Поэтому перед нами была поставлена цель осуществить передачу данных на компьютер с помощью созданного нами AngularJS web-приложения, вебкамеры, дальнейшей их обработки с помощью свободной библиотеки WebGazer.js и наглядное представление этих данных в виде тепловой карты построенной посредством JavaScript-срипта heatmap.js. Для достижения поставленной цели потребовалось решить следующие задачи:

* ознакомление со свободной библиотекой WebGazer.js;
* ознакомление со свободным скриптом heatmap.js
* изучение языка программирования JavaScript;
* изучения языка разметки HTML;
* Изучение языка таблиц стилей CSS
* Изучение фреймфорка AngularJS
* Работа с git и деплой приложения на хостинг GitHub Pages
* написание программы для внедрения библиотеки WebGazer.js в AngularJS web-приложение, а также для передачи данных в библиотеку отображения heatmap.js для посроения тепловой карты движения глаз по web-странице.

**ГЛАВА 1. Свободный браузерный айтрекер WebGazer.js**

**1.1.** **Особенности айтрекера WebGazer.js**

WebGazer.js - это свободно-предоставляемая JavaScript-библиотека для отслеживания местоположения взгяда пользователя на web-странице в режиме реального времени, использующая данные, полученные с помощью обычных веб-камер.

Webgazer.js разработана на основе исследования, проведеного Университетом Брауна[WebGazer: Scalable Webcam Eye Tracking Using User Interactions]. Работа над файлом калибровки была разработана в контекте курсового проекта с целью улучшения обратной связи WebGazer.js, предложенного доктором Джеральдом Веббером и его командой: доктором Клеменсом Зейдлером и Кай-Чунг Леуном.

Модель отслеживания взгяла, используемая в этой библиотеке, самокалибруется путём наблюдения за взаимодействиями веб-пользователя с веб-страницей и оттачивает зависимости между особенностями положения глаз пользователя и его местоположением на мониторе.

WebGazer.js - библиотека, написанная полностью на JavaScript и с помощью всего нескольких строк кода может быть интегрирована в лбьой веб-сайт. Webgazer работает полностью на клиентской стороне браузера, что гарантирует безопасность и сохранность персональных данных, так как никая информация не сможет быть отправлена на сервер без согласия пользователя на предоставление прав доступа к пользовательской веб-камере.

**1.2.** **Достоинства и недостатки айтрекера WebGazer.js**

В данной работе была выбрана именно эта библиотека в силу общедоступности программного обеспечения и средств разработки. Кроме того, WebGazer.js делает простой работу с пользоваельскими данными для разработчика, но, в отличие от иных систем, даѐт ряд превосходств для преподавателей, студентов и радиолюбителей:

* **Легкость в использовании.** Самое большое преимущество

библиотеки Webgazer.js - это готовая к использованию структура.Всё, что требуется для работы библиотеки это влючение нескольких строк кода в шапку HTML страницы.

* + **Широкая браузерная пожержка**.Еще одно большое преимуществоWebGazer.js-это возможность работы в большинстве современных браузеров за счет использования getUserMedia/Stream API[https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia] для доступа к веб-камере. Этот метод поддерживается большинством современных браузеров[https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia#Browser\_compatibility]
    - **Легкий доступ к выходным данным библиотеки.** Для получения предсказанных данных местоположения пользовательского взгляда достаточно добавить несколько строк JavaScript кода в тело HTML страницы.
    - **Программное обеспечение с открытым исходным кодом.**  WebGazer.js это библиотека с открытым исходным кодом, что дает большое преимущество для разработчка в плане понимания механизмов предсказания местоположения взгяда пользователя, реализованных в программе посредством просмотра “внутренностей” библиотеки.
  + **Нет необходимости в специальном программном обеспечении**. Большинство веб-камер поддерживаются данной библиотекой.

Но у WebGazer также есть и недостатки:

* **Структура.** СтруктураWebGazer.jsтакже является её недостатком.Присоздании проекта требуется совершать межскриптовые и межфункциональные переходы, а громоздкость библиотеки не позволяет осуществлять это быстро, а главное, зачастую теряется логическая связь между функциями, переменными и областями видимости.
* **Малочисленное сообщество**.В связи с малой извесностью данной библиотеки в сети практически нет примеров использования WebGazer.js в веб-разработке, в связи с этим во многих аспектах программы приходилось быть пионером.
* **Скудность документации.** Официальная документация описывает лишь малочисленные аспекты работы данной библиотеки.

**1.2.** **ETHERNET SHIELD**

Рис.1.Ethernet Shield P4S-348[5].

Ethernet Shield – это особая плата расширения, которая делает возможным удалѐнное управление. Данная плата делает возможным удалѐнное управление объектами через html-страницу браузера со своего компьютера, телефона, планшета. Она предоставляет разные сетевые функции в соответствии со средой и подключает Arduino к Интернету, распространяя при этом действие Arduino на огромные расстояния.

10

Подключение Ethernet Shield к аппаратной платформе Arduino осуществляется через длинные штыревые разъемы, кроме того на плате предусмотрены разъемы для подключения аналогичных ей плат. Для хранения файлов в плату вмонтирован разъѐм MicroSD. Его настройка реализовывается библиотекой SD. Взаимодействие платы Arduino с Shield и SD-картой осуществляется через шину SPI. Вывод 10 используется для выбора Shield, а вывод4 для выбора SD-карты. По причине того, что шина SPI одновременно используется и Shield и SD-картой, одновременно можно использовать только одно устройство. Стандартный разъѐм RJ45 установлен для подключения к маршрутизатору или компьютеру. Кнопка RESET, так же как и аналогичная кнопка на Arduino, перезагружает Shield и плату Arduino.

*Входы и выходы*

* помощью специальных функций библиотеки Ethernet цифровые выводы могут быть использованы как вход ил выход.
* Ethernet Shield есть 6 аналоговых входов (A0 - A5), каждый из которых может представить аналоговое напряжение в виде 10-битного числа (1024 различных значения). По умолчанию, измерение напряжения реализуется относительно диапазона от 0 до 5 В. Но вывод AREF и функцию analogReference() могут помочь изменить верхнюю границу этого диапазона.

*Программирование*

Есть два способа программирования Ethernet: первый способ осуществляется через 6-контактный разъем, другой через внешний ISP-программатор.

6-контактный программируемый разъем совместим с кабелями USB и другими похожими USB-UART переходниками от Sparkfun и Adafruit,

включая USB-Serial адаптер от Arduino. Разъем предназначен для обеспечения возможности автоматического сброса микроконтроллера во

11

время прошивки - это освобождает пользователя от необходимости самому нажимать кнопку сброса. При подключении к адаптеру USB плата питается от адаптера.

Если в среде разработки установить режим прошивки программатором, то Ethernet прошивается и с помощью внешнего программатора, такого как AVRISP mkII или USBTinyISP. Стоит учитывать, что его использование сотрѐт загрузчик во flash - памяти микроконтроллера.

*Физические характеристики*

Максимальная длина платы Ethernet составляет 6.9 см, ширина 5.4 см. Здесь учитывается тот факт, что разъемы RJ45 и разъемы питания выступают за пределы платы. Так же на плате на плате есть четыре крепежных отверстия, которые позволяют прикреплять плату к поверхности или корпусу. Расстояние между цифровыми выводами 7 и 8 и составляет 4 мм[6].

12

**ГЛАВА 2.ОСНОВНЫЕ СВЕДЕНИЯ О ПРОГРАММИРОВАНИИ**

**ARDUINO**

**2.1. Arduino и совместимые языки программирования**

В программировании у Arduino не очень много возможностей. Главным языком программирования является C/C++. От обычного C/C++ отличается только некоторыми доработками и упрощениями. Он используется в стандартной программной оболочке. Данный язык предпочтительно используется теми, кто уже знаком с другими языками программирования, и серьезно занимается робототехникой.

После установки специального плагина на стандартную среду Arduino IDE можно использовать язык программирования Scratch, а вернее его адаптацию — Ardublock. С помощью цветных блоков и конструктора становится возможным управление Arduino для людей, менее осведомленных о платформе.

Более «взрослая» версия данного языка – язык Snap! Наличие дополнительных возможностей, блоков, списков и функций делает его более сложным, но, всѐ же, недостаточным для разработки серьезных проектов. Чаще всего к нему обращается младшая возрастная группа и те, кто давно не занимался программированием. Для использования языка Snap! также нужно установить дополнительные компоненты.

Ещѐ один популярный язык программирования – Python. Это язык, с практически оптимальным уровнем сложности и возможностями в

программировании. Для его использования понадобятся библиотеки PySerial и vPython, которые доступны для скачивания в Интернете. К тому же через библиотеку PySerial Arduino взаимодействует с Go, Java и с HTML[7].

В данной работе кроме языка С используется язык программирования PHP - широко используемый язык cценариев общего нaзначения с открытым исхoдным кoдом. PHP расшифровывается как «Hypertext Preprocessor

13

(Препроцессор Гипертекста)». Этот язык программирования имеет возможность общаться с сервером и создавать динамическую веб-страницу для пользователя. PHP позволит вам создавать динамические веб-страницы на основе внешнего ввода, обычно от пользователя. Также важно отметить, что PHP является объектно-ориентированным языком программирования[8].

**2.2. СРЕДА РАЗРАБОТКИ ARDUINO IDE**

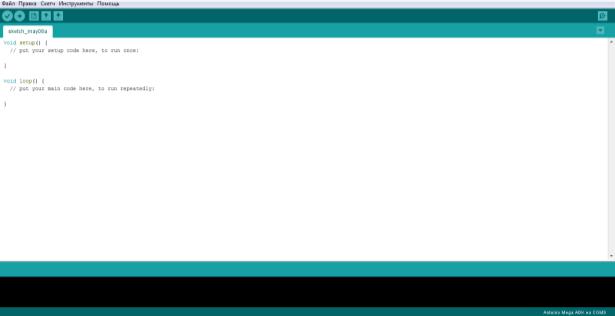


Рис. 4. Стартовое окно Arduino IDE.

Интегрированная среда разработки Arduino (IDE) - это кроссплатформенное приложение (для Windows, macOS, Linux), написанное на языке программирования Java. Он используется для написания и загрузки программ на Arduino-совместимые платы.

Исходный код для IDE выпущен под общедоступной лицензией GNU. Arduino IDE поддерживает языки C/C ++, используя специальные правила структурирования кода. Arduino IDE предоставляет библиотеку программного обеспечения из проекта Wiring, которая дает в распоряжение множество общих процедур ввода и вывода. Для написанного пользователем кода требуются только две основные функции - для запуска эскиза и основного цикла программы, которые компилируются и связываются с

14

заглушкой программы main () в исполняемую циклическую исполнительную программу с помощью цепочки инструментов GNU, также включенной в дистрибутив IDE. Arduino IDE использует программу avrdude для преобразования исполняемого кода в текстовый файл в шестнадцатеричной кодировке, который загружается в плату Arduino программой-загрузчиком во встроенном программном обеспечении платы[9].

**2.3. СТРУКТУРА ПРОГРАММ В СРЕДЕ ARDUINO IDE**

Программы в среде программирования Arduino IDE называются скетчами. Скетчи состоят из двух частей: setup и loop.

Функция setup() является обязательным разделом программы Arduino.

Объявление функции установки выполняется следующим образом:

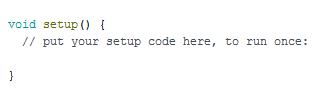


Рис.5. Вид функции setup().

Любой код, который находится внутри setup(), то есть между фигурными скобками ({}), выполняется только один раз в начале вашей программы.

Эта функция очень полезна для настройки Arduino. Именно здесь вы помещаете начальные настройки, такие как объявление входных и выходных пинов, открытие необходимых протоколов связи, установление соединения с дополнительными модулями и настройка подключенных библиотек.

Так же, как и функция setup(), loop() также является обязательной в программе Arduino. Большая часть кода помещается внутри этого раздела. Еѐ объявление производится следующим образом:

15

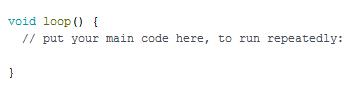


Рис.6. Вид функции loop().

Программа запускается сразу после открывающей фигурной скобки ({),

* процессор выполняет строки кода, пока не достигнет закрывающей фигурной скобки (}). В конце он возвращается к первой строке цикла и начинается заново.

Функция loop() будет выполняться вечно или до тех пор, пока вы не загрузите новый код, перезапустив процесс. Он также может быть перезапущен путем сброса Arduino (например, с помощью кнопки сброса)[10].

**2.4. БИБЛИОТЕКИ И ФУНКЦИИ**

**2.4.1. БИБЛИОТЕКИ**

Среда Arduino, как и большинство платформ программирования, может быть расширена за счет использования различных библиотек. Библиотеки предоставляют дополнительную функциональность для использования в

скетчах, например, работа с оборудованием или манипулирование данными. Чтобы использовать библиотеку в скетче, в среде Arduino IDE нужно выбрать Sketch> Import Library. Ряд библиотек предоставляется вместе с IDE, но вы также можете скачать или создать свою собственную.

*Библиотека Wire*

Эта библиотека позволяет вам общаться с устройствами I2C / TWI. На платах Arduino с разметкой R3 (вывод 1.0), SDA (линия передачи данных) и SCL (линия синхронизации) находятся на разъемах выводов рядом с выводом AREF.

Начиная с Arduino 1.0, библиотека наследует функции Stream, что делает ее совместимой с другими библиотеками чтения / записи. Из-за этого функции send() и receive() были заменены на read() и write().

16

*Библиотека SPI*

Последовательный периферийный интерфейс (SPI) - это протокол

синхронной последовательной передачи данных, используемый микроконтроллерами для быстрой связи с одним или несколькими периферийными устройствами на коротких расстояниях. Он также может быть использован для связи между двумя микроконтроллерами. При подключении SPI всегда есть одно главное устройство (обычно микроконтроллер), которое управляет периферийными устройствами. Данная библиотека позволяет общаться с устройствами SPI, с Arduino в качестве главного устройства.

*Библиотека SD*

Библиотека SD позволяет считывать и записывать на карты SD, например, на Arduino Ethernet Shield. Библиотека поддерживает файловые системы FAT16 и FAT32 на стандартных картах SD и SDHC. Имена файлов, передаваемые функциям библиотеки SD, могут включать в себя пути, разделенные прямой косой чертой /, например "Каталог / filename.txt". Поскольку рабочий каталог всегда является корнем SD-карты, имя относится к одному и тому же файлу независимо от того, содержит ли он косую черту (например, «/file.txt» эквивалентно «file.txt»). Начиная с версии 1.0, библиотека поддерживает открытие нескольких файлов.

Связь между микроконтроллером и картой SD осуществляется посредством SPI на цифровых пинах 50, 51 и 52 (Arduino Mega). Кроме того, для выбора SD-карты необходимо использовать другой пин. Это может быть аппаратный вывод SS - контакт 10 (на большинстве плат Arduino) или контакт 53 (на Mega) - или другой контакт, указанный в вызове SD.begin().

*Библиотека Ethernet*

Эта библиотека предназначена для работы с Arduino Ethernet Shield, Arduino Ethernet Shield 2, Leonardo Ethernet и любыми другими устройствами на базе W5100 / W5200 / W5500. Библиотека позволяет плате Arduino подключаться к Интернету. Плата может служить либо сервером,

17

принимающим входящие соединения, либо клиентом, создающим исходящие. Библиотека поддерживает до восьми (W5100 и платы с SRAM <= 2 кБ ограничены четырьмя) одновременными подключениями (входящими, исходящими или комбинацией).

Плата Arduino взаимодействует с Shield по шине SPI. Это на цифровых контактах 11, 12 и 13 на Uno и на контактах 50, 51 и 52 на Mega. На обеих платах контакт 10 используется в качестве SS. На Mega аппаратный вывод SS 53 не используется для выбора микросхемы контроллера Ethernet, но он должен быть сохранен в качестве выхода, иначе интерфейс SPI не будет работать.

**2.4.2. ФУНКЦИИ**

*Функция digitalWrite()*

Вид функции: digitalWrite (pin, value)

* pin: цифровой вывод, в который записывается значение.
* value: может быть высоким или низким.

Эта функция используется для записи значения HIGH или LOW на цифровой вывод.

*Функция digitalRead()*

Вид функции: digitalRead (digital\_pin)

* digital\_pin: цифровой вывод, который должен быть прочитан.

Эта функция используется для считывания цифрового сигнала с указанного цифрового контакта (digital\_pin). Функция возвращает HIGH или LOW[11].

*Функция delay()*

Если функция delay() вызывается в виде delay(2000), тогда это вызывет период ожидания 2 секунды (2000 миллисекунд - в одной секунде 1000 миллисекунд, также записывается 1000 мс).

18

Время задержки можно изменить, передав функции delay() другое значение. 3000 вызовет 3-секундную задержку: delay (3000).

*Функция Serial.begin()*

Вид функции: Serial.begin (baud\_rate)

* baud\_rate: скорость передачи данных, которая будет использоваться для последовательной связи. Может быть 4800, 9600, 14400, и т. д.

Эта функция используется для определения скорости передачи, которая будет использоваться для последовательной связи. Для связи с конкретными устройствами необходимо использовать скорость передачи данных устройства.

*Функция Serial.available()*

Эта функция используется для получения количества байтов, доступных для чтения из последовательного порта. Она дает количество байтов данных, которые поступили и хранятся в буфере последовательного приема.

*Функция Serial.print()*

Вид функции: Serial.print(value)

* значение: символ, строка.

Эта функция используется для печати данных на последовательный порт в форме, удобной для чтения (символы, строки, числа).

*Функция Serial.println()*

Вид функции: Serial.println(value)

* значение: символ, строка, номер для печати.

Эта функция используется для печати данных на последовательный порт в форме, удобной для чтения человеком (символ, строки, числа), за которой следует возврат каретки (\ r) и символ новой строки (\ n).

19

*Функция Serial.read ()*

Эта функция возвращает символ, полученный на пине Rx Arduino. Пример: char read\_byte

read\_byte = Serial.read ()

Байт прочитанных данных сохраняется в read\_byte.

*Функция println()*

Функция println() отправляет текст из последовательного / USB-порта Arduino и отображает в окне последовательного монитора.

Функция println() отличается от функции delay() тем, что в ней есть Serial и точка (.) перед ней: Serial.println («Text to print.»);

Причина этого обозначения (Serial.function\_name ()) заключается в том, что функция действует на последовательный порт(serial port) или последовательный объект(Serial object) [11].

20

**ГЛАВА 3. ПРОГРАММНАЯ ЧАСТЬ**

**3.1. Сборка экспериментального оборудования**



Рис.7. Arduino Mega ADK с установленным на ней Ethernet Shield.

Для выполнения поставленной задачи понадобилось следующее оборудование:

* плата Arduino Mega ADK;
* плата расширения Ethernet Shield;
* Сетевой кабель;
* USB кабель;
* Датчик IMU 10DOF;
* Провода папа-папа;

Непосредственная сборка установки:

1. На плату Arduino сверху c помощью коннекторов устанавливается

Ethernet Shield;

1. C помощью USB кабеля Arduino подключается к компьютеру, таким образом осуществляя питание и Arduino и Ethernet Shield;
2. C помощью сетевого кабеля Ethernet Shield подключается к компьютеру.

21

**3.2. Создание веб-сервера**

Первая задача состоит в использовании Arduino Ethernet Shield в качестве веб-сервера для загрузки HTML страницы.

Во вкладке «Панель управления» --> «Центр управления сетями и общим доступом» в списке активных сетей нужно найти «Неопознанную сеть». Это и есть подключенный Arduino Ethernet Shield. Далее во вкладке «Подключение по локальной сети» --> «Свойства» необходимо прописать следующие настройки:

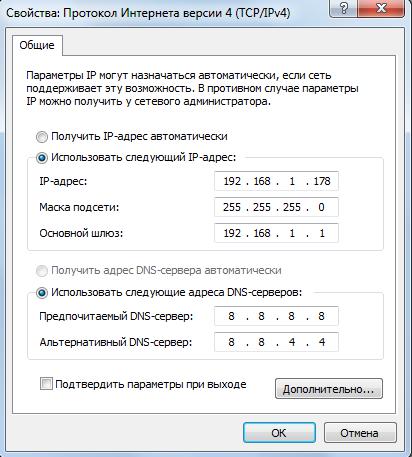


Рис.8. Настройки соединения Ethernet.

После используется программа PHPoc Debugger. PHPoC Debugger - это программное обеспечение, используемое для разработки и управления продуктами PHPoC. Его можно использовать для загрузки или выгрузки файлов между PHPoC и ПК, для добавления/редактирования/удаления файлов, отладки скрипта PHPoC

* реальном времени, настройки параметров PHPoC. Эта программа является вторым шагом в настройке веб-сервера.

22

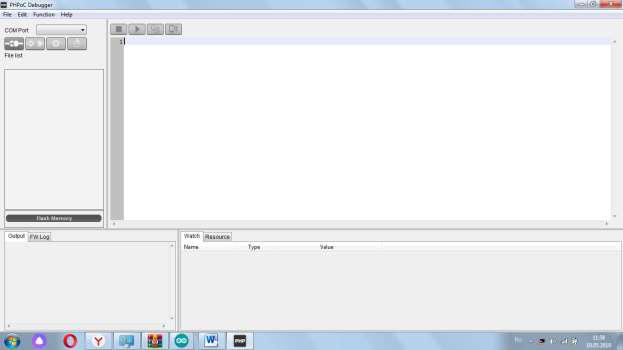


Рис.9. Стартовое окно PHPoc Debugger.

Во вкладке «COM Port» нужно выбрать тот порт, к которому подключен Ethernet Shield. В данном случае это «USB – COM4».



Рис.10. Вкладка COM Port.

* помощью кнопки «Connect» (цифра 1 на рис.7) нужно запустить отладчик PHPoc. Во вкладке «Settings» --> «Network»

прописываются те же настройки, что в Панели управления компьютера:

23

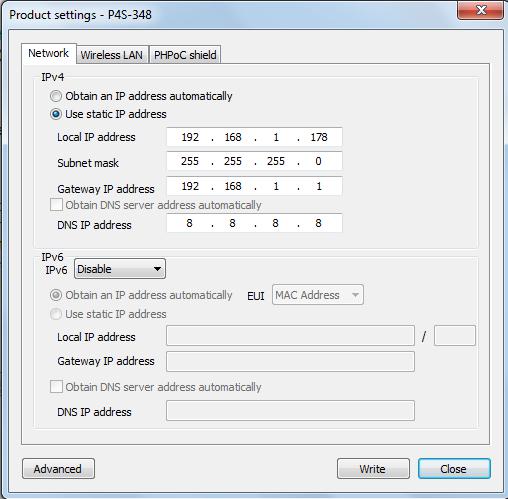


Рис.11. Настройки сети в программе PHPoc Debugger.

Таким образом, создаѐтся сервер, на который можно зайти в браузере, если ввести установленный IP-адрес: 192.168.1.178

Проверка работы веб-сервера:

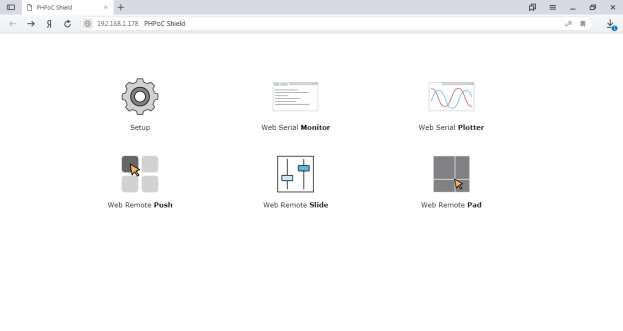


Рис.12. HTML страница Ethernet Shield.

24

* программе PHPoc Debugger уже существуют стартовые программы, написанные на языке php, которые открывают на странице некоторые приложения для контролирования Ethernet Shield(см.

рис.9). Мы будем использовать приложение «Web Serial Monitor».



Рис.13. Вкладка Web Serial Monitor

При нажатии кнопки «Connect» осуществляется подключение к портам Ethernet Shield и данные с подключенных датчиков выводятся в окошко. После нажатия кнопки «Connect» будет активирована

кнопка «Disconnect», с помощью которой вывод данных останавливается. С помощью кнопки «Clear» можно очистить окошко от данных.

25

**3.3.Приѐм данных с датчика**

Задача состоит в сборе данных с датчика Grove - IMU 10DOF.

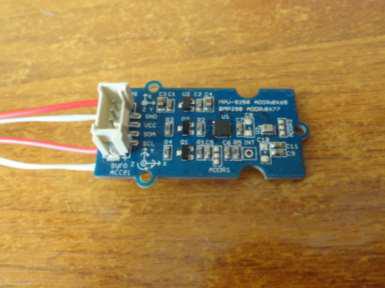


Рис.14. Датчик IMU 10DOF.

Grove - IMU 10DOF представляет собой комбинацию Grove - IMU 9DOF и Барометрический датчик BMP180. Этот модуль базируется на MPU-9250 и BMP180, MPU-9250 и является 9-осевым устройством «захвата движения», которое объединяет 3-осевой гироскоп, 3-осевой акселерометр, 3-осевой магнитометр и цифровой процессор движения (DMP), а также BMP180. Это высокоточные цифровые датчики давления сверхнизкой мощности для бытовых применений.

Подключение этого датчика осуществляется с помощью соединительных проводов мама-папа.

Таблица.1. Подключение IMU 10DOF.

|  |  |  |
| --- | --- | --- |
| **Grove - IMU 10DOF** |  | **Ethernet Shield** |
|  |  |  |
| GND |  | GND |
|  |  |  |
| VCC |  | 3.3V |
|  |  |  |
| SDA |  | SDA |
|  |  |  |
| SCL |  | SCL |
|  |  |  |
|  | 26 | |

В данной работе будут собираться следующие данные: температура, давление и высота над уровнем моря, которая рассчитывается по барометрической формуле. Таким образом, мы воспользуемся датчиком BMP280, встроенным в IMU 10DOF.

После подключения датчика согласно вышеприведенной таблице, нужно воспользоваться программой Arduino IDE. Для сбора данных понадобится специальная библиотека «BMP280\_DEV».

Для большей наглядности были построены графики изменения данных от времени. Для этого использовалось программа SFMonitor. Данная программа позволяет в реальном времени читать и визуализировать потоки данных, отправленные по последовательному порту.

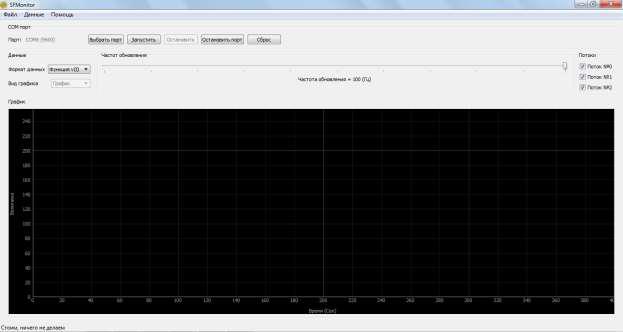


Рис.15. Стартовое окно SFMonitor.

Для настройки программы нужно выбрать порт, по которому отправляются данные и скорость порта. В данном случае это COM3(9600). Формат данных: функция v(t).

Для отправки пакета данных в программе Arduino IDE была использована библиотека «SerialFlow.h».

27

После компиляции программы (*Приложение*) в Arduino IDE, полученные данные можно посмотреть в веб-приложении «Web Serial Monitor»:

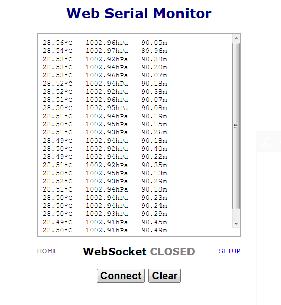


Рис.16. Приложение «Web Serial Monitor» с собранными данными. На рис.16. слева направо можно увидеть данные: температура,

давление, высота над уровнем моря.

Графики изменения данных от времени можно посмотреть в программе SFMonitor:

28

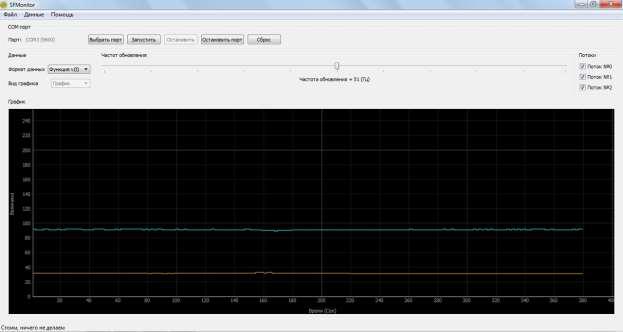


Рис.17. График изменения температуры (желтый цвет) и высота над уровнем моря (синий цвет) от времени.

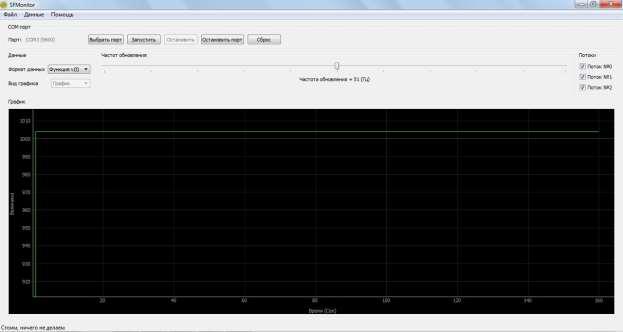


Рис.18. График изменения давления от времени (зеленый цвет).

29

**Заключение**

* + данной курсовой работе были осуществлены следующие

задачи:

1. Было проведено ознакомление с Ethernet Shield.
2. Было проведено ознакомление с аппаратной платформой Arduino Mega ADK.
3. Было проведено подключение Ethernet Shield к сети Интернет.
4. Была написана программа для передачи температуры, давления и высоты над уровнем моря с Arduino на компьютер по интерфейсу

Ethernet. Так же были построены графики изменения вышеперечисленных данных от времени.

Цель, поставленная в начале работы, достигнута, задачи выполнены.

30

**Список использованных источников**

1. Вид платы Arduio Uno[Электронный ресурс].-URL: http://robotech46.ru/shop/product/arduino-uno
2. Преимущества и недостатки платы Arduino[Электронный ресурс].-URL: http://engineerexperiences.com/advantages-and-disadvatages.html
3. Вид платы Arduino Mega ADK R3 [Электронный ресурс].-URL:http://doc.arduino.ua/ru/hardware/ADK
4. Память EEPROM в Arduino ADK [Электронный ресурс].-

URL:https://randomnerdtutorials.com/arduino-eeprom-explained-remember-last-led-state/

1. Вид платы Ethernet Shield PHPoc [Электронный ресурс].-URL: https://voltiq.ru/brand/sollae-systems/
2. Технические характеристики Ethernet Shield [Электронный ресурс].-URL:http://doc.arduino.ua/ru/hardware/Ethernet
3. Языки программирования Arduino [Электронный ресурс].-URL:https://geekbrains.ru/posts/arduino\_and\_languages
4. Язык программирования PHP [Электронный ресурс].-URL:http://howtostartprogramming.com/PHP/
5. О среде программирования Arduino IDE [Электронный ресурс].-URL: https://en.wikipedia.org/wiki/Arduino\_IDE
6. Структура программы в среде Arduino IDE [Электронный ресурс].-URL:https://www.circuitar.com/projects/setup-loop/index.html
7. Функции в языке программирования Arduino IDE[Электронный ресурс].-URL: https://startingelectronics.org/software/arduino/learn-to-program-course/02-main-loop-calling-functions/

31

**Приложение**

**Листинг программы**

#include <BMP280\_DEV.h>

#include <SerialFlow.h>

float temperature, pressure, altitude;

BMP280\_DEV bmp280;

SerialFlow rd(&Serial);

void setup()

{ rd.setPacketFormat(2, 3, 0); rd.begin(9600);

Serial.begin(9600);

bmp280.begin();

//bmp280.setPresOversampling(OVERSAMPLING\_X4);

//bmp280.setTempOversampling(OVERSAMPLING\_X1);

//bmp280.setIIRFilter(IIR\_FILTER\_4);

bmp280.setTimeStandby(TIME\_STANDBY\_2000MS);

bmp280.startNormalConversion();

}

void loop()

{

if (bmp280.getMeasurements(temperature, pressure, altitude))

{

Serial.print(temperature);

Serial.print(F("\*C "));

Serial.print(pressure);

Serial.print(F("hPa "));

32

Serial.print(altitude);

Serial.println(F("m"));

}

rd.setPacketValue(temperature);

rd.setPacketValue(pressure);

rd.setPacketValue(altitude);

rd.sendPacket();

delay(20);

}

33