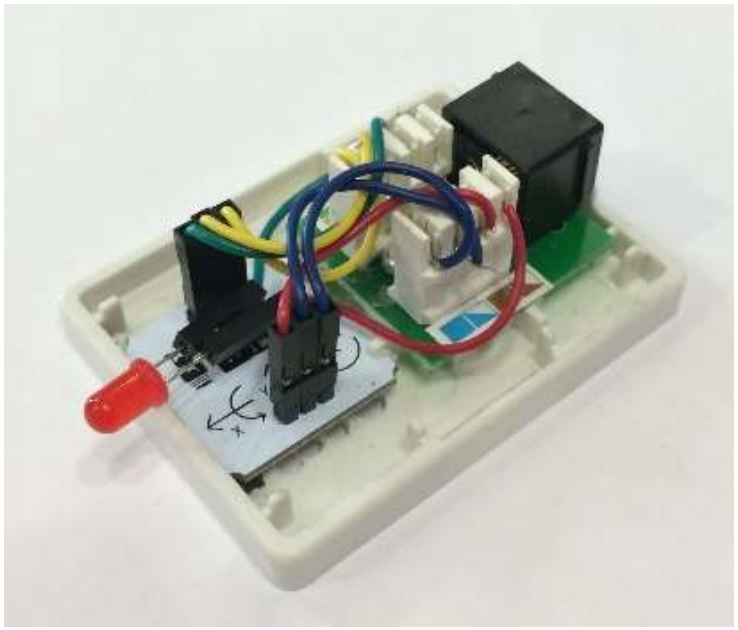
МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ «ДОМ ДЕТСКОГО ТВОРЧЕСТВА»

# «АВТОМАТИЧЕСКАЯ СИСТЕМА МЕТЕОРОЛОГИЧЕСКИХ НАБЛЮДЕНИЙ»





ПРОЕКТ ПОДГОТОВИЛ:
***Лапошин Никита Михайлович*** *19.07.2008 г.р.*
научное общество учащихся

РУКОВОДИТЕЛИ:
*педагоги дополнительного образования*
***Романенко Игорь Николаевич***
***Романенко Руслана Александровна***

г. ВИЛЮЧИНСК КАМЧАТСКИЙ КРАЙ

2024 г.

**СОДЕРЖАНИЕ**

# 1. ВВЕДЕНИЕ.

Важность и значимость наблюдений в ходе которых анализируется качество окружающей, включая такие параметры как температура, влажность, сила ветра и многое другое сложно переоценить. Метеорологические наблюдения позволяют обеспечить информацией отрасли хозяйственной деятельности человека, с целью наиболее полного и эффективного использования благоприятных условий погоды и климата и сокращения ущерба от опасных метеорологических явлений, а также что особенно ценно для комфорта каждого человека мониторить погоду.

Погода – уникальное явление. С самых древних времён, погода, несомненно, влияла на развитие истории человечества. Людям приходилось подстраиваться под неё. Дожди вынуждали людей прятаться в пещеры, морозы заставляли их переселяться, по поверхности земного шара. В экстремальных погодных условиях люди учились выживать, они развивались, придумывая одежду, различные орудия труда. Люди нуждались в тепле, возможно, это привело к тому, что человек получил огонь. Позже, с появлением земледелия, люди стали полностью зависеть от погоды. В засушливое время людям от безысходности приходилось молиться высшим силам, чтобы пошел дождь. Они приносили дары, жертвы шаманам, потому что считали, что они умеют управлять погодой, однако, поняв обратное, людям пришлось научиться предсказывать погоду. Необходимость заботиться о своей безопасности, и о своем благополучном существовании заставляла людей, а прежде всего рыбаков, крестьян, моряков самим следить за состоянием погоды и водных объектов, пытаться обобщать и накапливать прогностические признаки надвигающегося ненастья. Для этого, прежде всего, обращали внимание на облака, ветер, цвет зари и заката. По поведению животных, растений, птиц люди учились предсказывать погоду. На основе накопленных знаний и с течением времени, люди создали специальные устройства – метеостанции, по показаниям которых они научились предсказывать погоду с большей вероятностью. Но прогресс не стоит на месте.

Сегодня, каждый может сам создать простую метеостанцию, например, на основе платы-микроконтроллера Adruino UNO, назначение которой не только измерение погоды, но и исследовательская, а также учебная деятельность.

**Цель проекта:** создать компактную метеостанцию, способную отслеживать в реальном времени такие параметры как: температура, атмосферное давление и высота над уровнем моря на основе платы-микроконтроллера Arduino UNO а также прикладную программу для Windows в виде гаджета рабочего стола, на который будут выводиться показания.

**Задачи проекта:**

- Понять и изучить принцип работы с платой-микроконтроллера Arduino;
- Ознакомиться и изучить язык программирования Arduino;
- Научиться подключать и управлять датчиками для Arduino;
- Научиться выводить показания, снятые с COM-порта Arduino и основам создания html-приложения для рабочего стала Windows;

**Актуальность проекта:** современное развитие общества и использование информационных технологий во всех сферах человеческой жизни, заставляет нас изучать возможности различных программных и технических продуктов для решения конкретных задач. Современные конструкторские, инженерные и программные технологии невозможны без применения даже простейшей робототехники с которой мы сталкиваемся всё чаще в повседневной жизни. Для создания роботов и «умных» датчиков используются самые разные программные и технические решения. Я решил начать изучение азов робототехники на основе платы-микроконтроллера Arduino UNO.

Во-первых, работа с платой-микроконтроллера Arduino интересна и увлекательна, во-вторых, обширные возможности платы-микроконтроллера Arduino позволяют создавать самые различные технические устройства и модели: от простых светодиодов, до «умных датчиков», с которыми мы всё чаще сталкиваемся, а также сложные роботизированные системы.

**Новизна:** в настоящее время, научно-техническая сфера очень чётко делится на прикладную и фундаментальную. Множество современных научных открытий находятся на границе между фундаментальной и прикладной частями науки.

Робототехника – одно из тех направлений, которое объединяет в себе глубокие изыскания в фундаментальной сфере и ярко выраженное прикладное применение в современных технологиях.

Мой проект наглядно демонстрирует совмещение системного программирования микроконтроллеров с прикладным программированием для операционных систем Windows. На примере моего проекта можно сделать вывод, что автоматизированные системы на основе микроконтроллеров весьма полезны, и востребованы на современном этапе развития технологий робототехники.

## 2. ОСНОВНАЯ ЧАСТЬ.

### 2.1. Описание проекта.

В ходе работы над проектом была выдвинута и проверена следующая **гипотеза**: используя возможности платы-микроконтроллера Arduino UNO, навыков программирования, технического моделирования, можно создать компактную метеостанцию для измерения температуры, атмосферного давления и высоты над уровнем моря. А также было создано html-приложение для рабочего стола Windows, в окне которого будут выводиться показания. Проект представляет собой компактную метеостанцию на основе платы-микроконтроллера Arduino UNO и гаджет рабочего стола Windows на который выводятся показания. Проект разработан с применением ***платы Arduino UNO***, ***датчика AMP-B034 Troyka*** и программ: ***текстовый редактор NotePad++, редактор кода Arduino Studio, компилятор языка программирования C (tdm-gcc-4.8.1).*** В папке «Troyka» содержатся библиотеки и «скетчи» для модуля AMP-B034 Troyka, и схема подключения его к Arduino. В папке «Гаджет» находится рабочая папка с html-приложением для рабочего стола Windows, внутри которой находится инструкция по установке.

### 2.2. Порядок выполнения проекта.

#### 2.2.1. Разработка идеи и постановка целей и задач.

Размышления о том, какой должна быть модель, что она должна из себя представлять, являются наиважнейшим этапом создания, проекта. Решение

создания метеостанции исходит из её функциональности, и, непосредственной пользы применения в быту. Целью было создание метеостанции, которая измеряет атмосферное давление, температуру и высоту над уровнем моря и выводит показания на, созданное к ней приложение рабочего стола Windows.

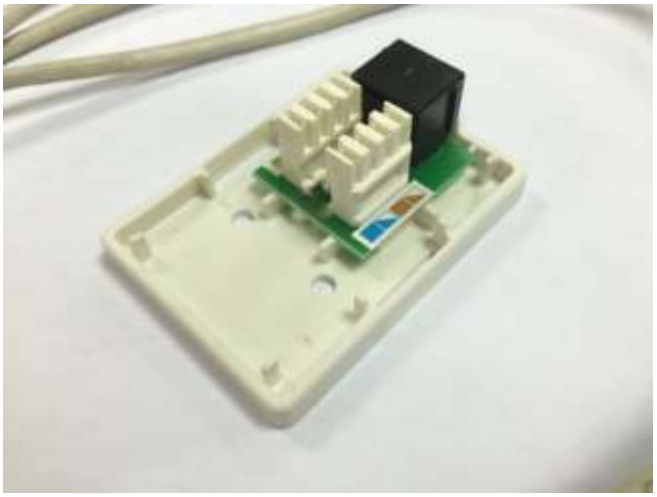
### 2.2.2. Схема подключения и внешний вид использованного датчика.

Создание станции началось с поиска схемы, по которой я бы мог её собрать. Мне потребовалась лишь схема подключения модуля Troyka к Arduino так как он уже содержит в себе температурный датчик:

Схема подключения

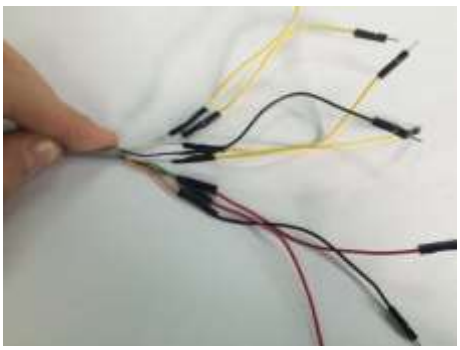
### 2.2.3. Конструктив. Подборка корпуса и кабеля.

Сама метеостанция располагается на некотором расстоянии от Arduino, так как должна находиться за окном. Для этих целей я решил использовать сетевой коммутатор и сетевой кабель.



### 2.2.4. Конструктив. Пайка проводов.

Я подготовил сетевой кабель и провода Arduino для пайки, а затем спаял их и изолировал места пайки

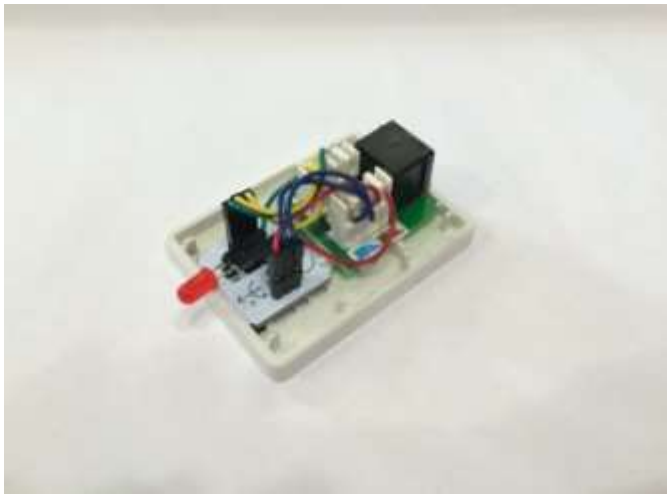


### 2.2.5. Конструктив. Подключение проводов к коммутатору

Я подключил провода, с выходами для подключения к датчику Troyka к коммутатору в соответствии с их расцветкой. Можно заметить, что по схеме подключения датчика, приведённой выше задействованы только шесть проводов. Оставшиеся провода понадобились на подключение светодиода, показывающего что модуль работает.

### 2.2.6. Конструктив. Интеграция модуля Troyka в корпус и его подключение.

Я поместил модуль в корпус и подключил провода и светодиод в соответствии со схемой. Затем, сделал отверстия, для контакта датчика с окружающей средой.



### 2.2.7. Программирование. Скетч и библиотеки.

Для работы с модулем Troyka необходимы библиотеки, которые подключаются к скетчу Arduino. В библиотеках прописаны вычисления и параметры для считывания показаний с датчика. Для полноценной работы платы-микроконтроллера Arduino UNO и датчиков, необходимо написать, так называемые «Скетчи», которые представляют собой обычный программный код. Скетчи пишутся в среде Arduino Studio, после чего загружаются в память платы Arduino и осуществляют управление работой микроконтроллера. Здесь прописаны обозначения велечин, ссылка для обращение к порту Arduino и подклечение к библиотекам.

### 2.2.8. Html-приложение. Решение о создании гаджета.


При использовании метеостанции, не всегда представляется удобным снимать показания с помощью программы Arduino Studio, поэтому, я решил сделать очень компактное и удобное приложение для рабочего стола Windows на которое будут выводитьься показания снятые с метеостанции. Гаджет представляет собой папку со скриптами, файлом манифеста и html файлом, а так же с использованными изображениями. С порядком работы над созданием гаджета можно ознакомиться в приложении. Дополнительно я написал программу для считывания данных с COM порта Arduino, которая будет выводить считанные данные в xml файл откуда гаджет берёт показания.

## 3. ЗАКЛЮЧЕНИЕ.

В заключении, хотелось бы сказать, что работая над проектом я изучил основы робототехники, базовые принципы работы с микроконтроллером Arduino UNO и возможности Web-программирования, а так же расширил свои знания и навыки в области программирования на языках «C», «Java Scripts» и «Visual Basic». Итогом работы стала портативная метеостанция, собранная на основе платы-микроконтроллера Arduino UNO и модуля AMP-B034 Troyka, содержащего в себе температурный датчик и барометр, а также html-приложение (гаджет) для рабочего стола Windows в окне которого выводятся показания метеостанции. Мой проект наглядно демонстрирует область практического применения системного программирования микроконтроллеров и прикладного программирования для операционных систем Windows.

В ходе работы над проектом мною была подтверждена гипотеза, использовав возможности платы-микроконтроллера Arduino UNO и навыков программирования.

## 4. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ И ИСТОЧНИКОВ ИНФОРМАЦИИ


1. Стернзат М.С. Метеорологические приборы и измерения. – Л.: Гидрометиздат, 1978 г.
2. Городецкий О.О. и др. Метеорология, методы и технические средства наблюдений. – Л.: Гидрометиздат, 1984 г.
3. [http://arduino-project.net/pogodnaya-stantsiya-na-arduino/].
4. [https://www.arduino.cc]

# 4. ПРИЛОЖЕНИЕ. ПОРЯДОК РАБОТЫ НАД СОЗДАНИЕМ ГАДЖЕТА РАБОЧЕГО СТОЛА.

**ШАГ 1:** *Разработка идеи.*

Размышления о том, каким должен быть гаджет, что он должен из себя представлять, являются наиважнейшим этапом его создания. Решение создания гаджета рабочего стола исходит из необходимости удобного вывода метео-показаний на основе данных, полученных с различных погодных ресурсов.

**ШАГ 2:** *Постановка целей и задач*.

Целью было создание гаджета на который выводятся показания атмосферного давления и температуры снятые с погодного интернет-ресурса, либо с одноплатного компьютера Arduino.

**ШАГ 3:** *Структуризация проекта.*

Для удобства реализации проекта я поместил его в каталог **гаджет,** где находятся папки с двумя версиями гаджета: **weather_Nikitop.gadget -** для работы, непосредственно с файловой системой и **weather_Nikitop_web.gadget** – для получения погодных показаний с сайта. Вторая версия отличается тем, что вместо атмосферного давления и высоты над уровнем моря - выводит уровень влажности и атмосферного давления, а так же показывает состояние погоды. В обоих директориях лежат файлы манифеста гаджета, Visual Basic и Java скрипты, текстовый документ: «Установка гаджета» где прописана инструкция по его установке, а также папка **images** с необходимыми для гаджета изображениями. В директории с первой версией гаджета находится программа **write.exe** для записи показаний в файл **weather.xml**, а также папка **images** с необходимыми для гаджета изображениями.

**ШАГ 4:** *Сбор и обработка информации.*

Информацию о работе с гаджетом, в частности с JavaScripts и VBScripts взял из официальных источников, и интернета.

**ШАГ 5:** *Написание манифеста гаджета*.

Файл **Gadget.xml** – манифест гаджета, является основным файлом гаджета.

Здесь собрана информация о названии гаджета, версии, об авторе гаджета, а так же его описание и информация об иконке.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<gadget>
 <name>Погода</name>
 <version>1.0</version>
 <hosts>
  <host name="wdonma">
   <base type="HTML" apiVersion="1.0.0" src="main.html" />
   <permissions>Full</permissions>
   <platform minPlatformVersion="1.0" />
  </host>
 </hosts>
 <icons>
   <icon width="128" height="128" src="icon.png" />
 </icons>
 <author name="Бабченко Никита"></author>
 <description>Гаджет, показывающий погоду на основе данных с метеостанции Arduino </description>
 <icons></icons>
</gadget>
```

**ШАГ 6:** *Написание main.html.*

**main.html** представляет собой обычный html файл, содержащий информацию о расположении изображений на гаджете, параметры его фона в режимах docked и undocked, а так же осуществление вызова работы скриптов **main.vbs**, и **main.js**

```
<SCRIPT Language="VBScript">
'Вызов функции JavaScript из области VBS
function isDocked
    isDocked = isDockedJS()
End Function
</SCRIPT>
<script src="main.vbs" type="text/vbscript"></script>
<SCRIPT Language="JavaScript">
    docked = 0;
    function isDockedJS() {
      return docked;
    }
</SCRIPT>
<script type="text/javascript" src="main.js"></script>
</head>
```

```
<style>
body{ width:150px; height:202px; padding: 0; margin: 0; font-family: Tahoma; font-size: 10px;  color: #fff; }

#small_mainbg { width: 148px; height: 201px; }
#small_maincloudyimg { position: absolute; top: 38px; left: 8px;  }
#small_maintemp { position: absolute; top: 35px; left: 65px; font-size: 28px; }
#small_maintempminmax { position: absolute; top: 73px; left: 65px; font-size: 10px; }
#small_maincloudy { position: absolute; top: 81px; left: 65px; font-size: 11px; }
#small_mainair { position: absolute; top: 92px; left: 65px; }
#small_mainhumidity { position: absolute; top: 101px; left: 65px; }
#small_mainwind { position: absolute; top: 110px; left: 65px; }

#big_mainbg { width: 230; height: 160; background: url(images/gadgeth.png) }
#big_maincloudyimg { position: absolute; top: 19px; left: 8px;   }
#big_header { position: absolute; left: 38; top: 1; font-size: 12px; }
#big_maintemp { position: absolute; top: 15px; left: 100px; width: 120; text-align: right; font-size: 28px; }
#big_maintempminmax { position: absolute; top: 52px; left: 100px; width: 120; text-align: right; font-size: 10px; }
#big_maincloudy { position: absolute; top: 62px; left: 60px; width: 160; text-align: right;  font-size: 11px; }

#big_mainhumidity { position: absolute; top: 75px; left: 100px; width: 120; text-align: right;  }
#big_mainwind { position: absolute; top: 85px; left: 100px; width: 120; text-align: right; }

a { color: #fff; text-decoration: none; }
a:hover { text-decoration: underline; }
</style>
```

**ШАГ 7**: *Написание main.js*

Файл **main.js** является Java скриптом, где прописаны параметры конкретного расположения элементов на фоне гаджета в состояниях docked и undocked.

```
if (System.Gadget.docked) {
  // docked
   bd.width=146;
   bd.height=122;
   bd.background='url(images/gadget.png) no-repeat';
   document.getElementById("small_needupdate").innerHTML = document.getElementById("big_needupdate").innerHTML;
   document.getElementById("big_needupdate").innerHTML = '';

   var text = document.getElementById("big_maincloudyimg").innerHTML;
   var newString = text.replace("b.png", "m.png");

   document.getElementById("small_maincloudyimg").innerHTML = newString;//document.getElementById("big_maincloudyimg").innerHTML;
   document.getElementById("small_maintemp").innerHTML = document.getElementById("big_maintemp").innerHTML;
   document.getElementById("small_maintempminmax").innerHTML = document.getElementById("big_maintempminmax").innerHTML;

   var text = document.getElementById("big_maincloudy").innerHTML;
   var arr = text.split(/[,]/);

   document.getElementById("small_maincloudy").innerHTML = arr[0];//document.getElementById("big_maincloudy").innerHTML;
   if (arr[1] != undefined) document.getElementById("small_mainair").innerHTML = arr[1];
   //document.getElementById("small_maincloudy").innerHTML+= ', '+document.getElementById("big_mainair").innerHTML;
   document.getElementById("small_mainhumidity").innerHTML = document.getElementById("big_mainhumidity").innerHTML;
   document.getElementById("small_mainwind").innerHTML = document.getElementById("big_mainwind").innerHTML;

   document.getElementById("big_header").innerHTML = '';
   document.getElementById("big_maincloudyimg").innerHTML = '';
   document.getElementById("big_maintemp").innerHTML = '';
   document.getElementById("big_maintempminmax").innerHTML = '';
   document.getElementById("big_maincloudy").innerHTML = '';
   document.getElementById("big_maincloudy").innerHTML = '';
   document.getElementById("big_mainhumidity").innerHTML = '';
   document.getElementById("big_mainwind").innerHTML = '';
```

```
   // big state
   bd.width=227;
   bd.height=98;
   bd.background='url(images/gadgetb.png) no-repeat';
   docked=0;
   document.getElementById("big_needupdate").innerHTML = document.getElementById("small_needupdate").innerHTML;
   document.getElementById("small_needupdate").innerHTML = '';
   document.getElementById("big_header").innerHTML = 'Вилючинск, Камчатский край'; //отображение региона и города
   var text = document.getElementById("small_maincloudyimg").innerHTML;
   var newString = text.replace("m.png", "b.png");

   document.getElementById("big_maincloudyimg").innerHTML = newString;//document.getElementById("small_maincloudyimg").innerHTML;
   document.getElementById("big_maintemp").innerHTML = document.getElementById("small_maintemp").innerHTML;
   document.getElementById("big_maintempminmax").innerHTML = document.getElementById("small_maintempminmax").innerHTML;
   document.getElementById("big_maincloudy").innerHTML = document.getElementById("small_maincloudy").innerHTML;
   if (document.getElementById("small_mainair").innerHTML != '') document.getElementById("big_maincloudy").innerHTML+= ', '+document.getElementById("small_main
   document.getElementById("big_mainhumidity").innerHTML = document.getElementById("small_mainhumidity").innerHTML;
   document.getElementById("big_mainwind").innerHTML = document.getElementById("small_mainwind").innerHTML;

   document.getElementById("small_maincloudyimg").innerHTML = '';
   document.getElementById("small_maintemp").innerHTML = '';
   document.getElementById("small_maintempminmax").innerHTML = '';
   document.getElementById("small_maincloudy").innerHTML = '';
   document.getElementById("small_mainhumidity").innerHTML = '';
   document.getElementById("small_mainwind").innerHTML = '';
   document.getElementById("small_mainair").innerHTML = '';
```

**ШАГ 8:** *Написание main.vbs.*

Файл **main.vbs** является скриптом, написанным на языке Visual Basic. Служит, так называемым парсером – то есть скриптом, для считывания данных из соответствующего xml файла, а также для последующего вывода этих данных на гаджет. Содержимое этого файла в папке со второй версией гаджета отличается, так как «web» версия гаджета скачивает xml файл с интернет ресурса и «парсит» его из другой директории.

```
Sub ParseXML
  Set fso = CreateObject("Scripting.FileSystemObject")
  filepath = ".\weather.xml"
  Set xmlDoc = CreateObject("Msxml2.DOMDocument")
  xmlDoc.async="false"
  xmlDoc.load(filepath)
  daycount = 0
  prefix = "small_"
  imgprefix = "n.png>"
  if isDocked() = 0 then
    prefix = "big_"
    imgprefix = "b.png>"
  end if

  'get main node <all>
  Set currNode = xmlDoc.documentElement
  'get other nodes <day>
  Set dayNode = currNode.firstChild
  While Not dayNode Is Nothing
    Set currNode = dayNode.firstChild
    While Not currNode Is Nothing
      if currNode.parentNode.getAttribute("id") = "today" then
      if currNode.nodeName = "temp" then document.getElementById(prefix+"maintemp").innerHTML = currNode.childNodes(0).text+Chr(176)
      if currNode.nodeName = "cloudyin" then document.getElementById(prefix+"maincloudying").innerHTML = "<img src=images/"+currNode.childNodes(0).text+imgprefi
      if currNode.nodeName = "humidity" then document.getElementById(prefix+"mainhumidity").innerHTML = "давление "+currNode.childNodes(0).text+" Hp" 'Давление
      if currNode.nodeName = "wind_direction" then document.getElementById(prefix+"mainwind").innerHTML = "высота "+currNode.childNodes(0).text+" м"  'высота
      if currNode.nodeName = "min" then document.getElementById(prefix+"maintempminmax").innerHTML = currNode.childNodes(0).text + +Chr(176)
      if currNode.nodeName = "max" then document.getElementById(prefix+"maintempminmax").innerHTML = document.getElementById(prefix+"maintempminmax").innerHTML
    end If
    Set currNode = currNode.nextSibling
    Wend
  Set dayNode = dayNode.nextSibling
```

**ШАГ 9:** *Написание программы для вывода данных в xml файл.*

Для снятия показаний и вывода их в читаемый гаджетом файл **weather.xml,** к которому ссылается парсер, мне потребовалось написать программу на языке “C”. На изображении слева представлен код программы, справа – созданный программой weather.xml с показаниями

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
FILE *f;
FILE *i;
i=fopen("input.txt","r");
f=fopen("weather.xml","w");

float t, p, h, min, max;

fscanf(i,"%f %f %f %f %f",&t,&h,&p,&min,&max);
fprintf(f,"<?xml version=\"1.0\"?>\n");
fprintf(f,"<all>\n");
fprintf(f,"<day id=\"today\">\n");
fprintf(f," <temp>%.1f</temp>\n",t);
fprintf(f," <cloudyim>34</cloudyim>\n");
fprintf(f," <humidity>%.1f</humidity>\n",p);
fprintf(f," <wind_direction>%.1f</wind_direction>\n",h);
fprintf(f," <min>%.0f</min>\n",min);
fprintf(f," <max>%.0f</max>\n",max);
fprintf(f,"</day>\n");
fprintf(f,"</all>");
fclose(i);
fclose(f);

return 0;
}
```

```xml
<?xml version="1.0"?>
<all>
<day id="today">
 <temp>15.0</temp>
 <cloudyim>34</cloudyim>
 <humidity>40.0</humidity>
 <wind_direction>788.0</wind_direction>
 <min>10</min>
 <max>17</max>
</day>
</all>
```

**ШАГ 10:** *Описание использованных мной изображений.*

В ходе работы над гаджетом, мной был использован ряд изображений. Гаджет принимает одно из двух состояний – docked и undocked. Фоновые изображения для этих состояний я заготовил с помощью графического редактора Adobe Photoshop CS6. Так же мной была выбрана иконка, и две картинки с изображением солнца, высвечиваемые при работе гаджета. «Web» версия гаджета отличается наибольшим набором изображений, так как способна показывать состояние погоды (пасмурно, солнечно и т.п.). Ниже приведён набор изображений из первой версии