

ЭЛЕКТРОННЫЙ ПРЕДСКАЗАТЕЛЬ ПОГОДЫ

Гаевский М.С.

г. Керчь, МБОУ г. Керчи РК «Школа № 23», 10 класс

Руководитель: Семик Е.А., г. Керчь, МБУДО г. Керчи РК «Центр научно-технического творчества», педагог дополнительного образования секции «Радиотехническое творчество»

Настоящая работа посвящена изготовлению и исследованию электронного предсказателя погоды.

Актуальность работы заключается в необходимости таких приборов в домашних условиях для людей пожилого возраста, которые не имеют персональных компьютеров и интернета, где можно посмотреть прогноз погоды. А также не могут позволить приобрести себе приборы для измерения погодных показателей. Особенно это касается метеозависимых людей, у которых состояние здоровья зависит от атмосферного давления, влажности воздуха или температуры.

Цель исследования:

- изучение схем электронных барометров, термометров и измерителей влажности воздуха;
- изучение принципа действия датчиков давления, температуры и влажности воздуха
- изучение алгоритма работы микроконтроллера «Arduino NANO»;
- изучение программного обеспечения «Arduino»;
- изучение технологии программирования;

В результате проделанной работы изготовлен прибор «Электронный предсказатель погоды» (ЭПП). Стоимость данного прибора ≈ 700 рублей. Стоимость аналогичного прибора в интернете ≈ 3000 рублей.

Людам очень важно узнавать, как в ближайшее время изменится погода. Ведь она значительно может повлиять на их планы на день и на состояние здоровья. Сильному влиянию на самочувствие влажности воздуха и атмосферному давлению подвержены метеозависимые люди: астматики и сердечники. Поэтому необходимо следить за прогнозом погоды. Для этого используются различные источники информации (телевидение, радио) и в частности Интернет.

Но люди пожилого возраста зачастую не могут пользоваться глобальной сетью. Поэтому для них предназначен ЭПП – электронный предсказатель погоды. Данный прибор малогабаритный и поэтому он так же может применяться в походах для отслеживания ухудшения погоды, что очень поможет туристам.

Прибор выполнен на базе платформы Arduino «NANO» с микроконтроллером ATmega328P.

В ЭПП используются датчик температуры и влажности DHT22, датчик давления BM3180 и дисплей для вывода данных о погоде. Стоит отметить, что домашних метеостанций на основе Arduino и барометра на просторах Интернета достаточно много, но они показывают всего лишь текущее значение и не предсказывают изменение погоды в будущем. Погода от атмосферного давления и температуры воздуха. Чтобы предсказать изменение погоды заранее нужно следить за изменением давления по времени, то есть будет показываться не текущее давление, а «дельта» (насколько изменилось давление за последующие несколько часов). Сравнение давления будет идти каждые 10 минут, учитывая предыдущее. Для этого будут применены математические формулы в прошивке микроконтроллера. Так же для максимальной доступности будут применены формулы для перевода «дельты» в проценты. То есть на дисплее будет выводиться шанс дождя в процентах. Рядом будут выводиться показатели текущей температуры и влажности воздуха.

Так же упор данного проекта будет сделан на энергосбережение и максимально длительную работу ЭПП. Для этого в плате Arduino будут вырезаны некоторые ненужные детали. А так же дисплей не будет светиться постоянно, его можно будет включать вручную при помощи кнопки. То есть значения будут выводиться на дисплей постоянно, но посмотреть их можно будет, только если нажать на кнопку.

Таким образом, его работа рассчитана примерно на 3-4 месяца на 1 аккумуляторе (9900 mAh). Так как аккумулятор не будет выдавать напряжение больше 4 V, ЭПП нужно подключать через повышающий преобразователь напряжения. Датчики и микроконтроллер по стоимости обходятся намного дешевле, чем 3 прибора (анероид, термометр и гигрометр).

Таким образом, можно сказать, что ЭПП является понятным, доступным и полезным, как для людей пожилого возраста, так и для туристов.

Обзор барометров, термометров, гигрометров

1. Аналоги барометра

Одним из аналогов барометра является **анероид** – прибор для измерения атмосферного давления, в отличие от ртутного барометра, действующий без помощи жидкости. Приемной частью анероида служит цилиндрическая металлическая коробочка с концентрически-гофрированными основаниями, внутри которой создано разрежение. При повышении атмосферного давления коробочка сжимается и тянет прикрепленную к ней пружину, а при понижении давления коробочка раздувается, толкая пружину. Перемещение конца пружины через систему рычагов передается на стрелку, перемещающуюся по шкале.

Электронный барометр – современная разновидность данного прибора, линейные показатели обычного барометра-анероида преобразовываются в электронный сигнал, который обрабатывается микропроцессором и выводится на жидкокристаллический экран. Имеет компактные размеры, прост и удобен в использовании, например, для туризма или как дачный вариант; ≈ 2000 р.

2. Аналоги датчика температуры

Термометры для измерения температуры окружающей среды чаще всего спиртовые и ртутные. Они основаны на принципе изменения объема жидкости, которая залита в термометр, при изменении температуры окружающей среды. Так же есть механические, оптические, газовые, инфракрасные и электронные. ≈ 250 р.

Датчик температуры работает на основе встроенного металлического датчика (платина, никель, медь). В ЭПП используется медный датчик. Принцип работы основан на изменении сопротивления.

Для большинства металлов температурный коэффициент сопротивления положителен – их сопротивление растет с ростом температуры. Для полупроводников без примесей он отрицателен – их сопротивление с ростом температуры падает.

3. Аналоги датчика влажности

Гигрометр – прибор, который определяет уровень влажности воздуха в окружающем пространстве. Чаще всего используются механические, конденсационные и электронные. ≈ 500 р.

Механический гигрометр – прибор, в который входит пористая или твердая керамическая масса, в состав которой также входят металлические элементы имеет электрическое сопротивление. Его уровень

напрямую зависит от влажности. Для правильного его действия керамическая масса должна состоять из некоторых окислов металла. В качестве основы используется каолин, кремний и глина.

Конденсационный гигрометр – прибор, принцип которого основывается на использовании встроенного зеркала. Температура этого зеркала изменяется вместе с температурой воздуха в окружающем пространстве. Определяется его температура в первоначальный момент измерения. Далее на поверхности зеркала появляются капли влаги либо небольшие кристаллы льда. Температура измеряется еще раз. С помощью разницы температур, определенных конденсационным гигрометром, и определяется влажность воздуха.

Электронный гигрометр – прибор имеющий пластинку из стекла или другого подобного электроизоляционного вещества, на которое наносят слой хлорида лития. Меняется влажность – увеличивается или уменьшается концентрация и сопротивляемость хлористого лития. Стоит отметить, что на показания электронного (электролитического) гигрометра может оказывать незначительное влияние температура воздуха, поэтому он часто оборудован встроенным термометром. Такой гигрометр предельно точен и дает показания с минимальной погрешностью.

Выводы

Приборы для измерения давления, температуры и влажности могут не требовать электроэнергии, но они достаточно габаритные и дорогие. Электронный барометр BMP180 размером 12×9 мм ($=75$ рублей); датчик температуры и влажности размером $20 \times 15 \times 13$ мм ($=140$ рублей). Использование датчиков является более рациональным решением, чем использование готовых приборов.

Функциональная схема ЭПП

На рисунке 1 представлена функциональная схема ЭПП.

Барометр является аналоговым датчиком, поэтому его сигналы идут в аналогово-цифровой преобразователь (АЦП) для привращения аналогового сигнала в цифровой. Датчик температуры и влажности является цифровым, со встроенным АЦП поэтому преобразование его сигналов не требуется. Далее уже цифровые сигналы идут в микропроцессор, где написанная программа выполняет с ними то, что в ней указано. В данном случае параметры с датчика температуры и влажности выводятся на дисплей, а параметры с барометра подвергаются математическим действиям для вывода на дисплей значений в ином виде (в процентах).



Рис. 1.

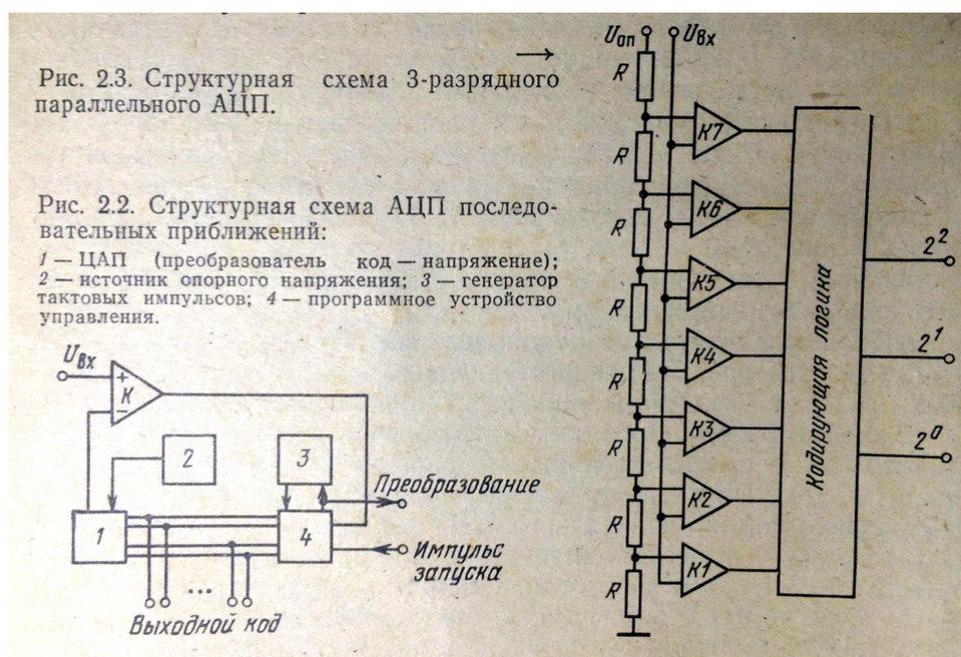


Рис. 2.

1. Устройство АЦП

Аналого-цифровые преобразователи (АЦП) являются устройствами, которые принимают входные аналоговые сигналы и выдают на выходе соответствующие им цифровые сигналы, пригодные для работы с ЭВМ и другими цифровыми устройствами. Самым простым примером АЦП является Параллельные АЦП. Преобразователи этого типа осуществляют одновременно квантование сигнала с помощью набора компараторов, включенных параллельно источнику сигнала (рис. 2.3). Пороговые уровни компараторов установлены с помощью резистивного делителя в соответствии с используемой шкалой квантования. При подаче на такой набор компараторов исходного сигнала на выходах последних будет иметь место проквантованный сигнал, представленный в унитарном коде. Для преобразования этого кода в двоичный, используются логические схемы, называемые обычно кодирующей логикой.

2. Технические показатели составляющих ЭПП

Устройство	Характеристика
Барометр BMP180	Датчик аналоговый; потреб. напряжение – 3,3V; потреб. ток – 3...32 мкА; диапазон давления: 300...1100 гПа.
Датчик температуры и влажности DHT22	Датчик цифровой; потреб. напряжение – 5V; потреб. ток – 1...2.5мА; диапазон влажности от 0% до 100%; диапазон температуры: -40°C до 125°C;
Arduino «Nano»	
Тип микроконтроллера	ATmega328P
Напряжение питания	5V

продолжение табл.	
Рекомендуемое напряжение питания	7-12V
Предельно допустимое напряжение	6-20V
Цифровые входы-выходы	14 (из них 6 поддерживают ШИМ)
Выходы ШИМ модуляции	6
Аналоговые входы	6
Допустимый ток цифровых выходов	20мА
Допустимый ток выхода 3,3V	50мА
Объем флэш памяти (FLASH)	32 кБ (0,5 кБ используется загрузчиком)
Объем оперативной памяти (SRAM)	2 кБ
Объем энергонезависимой памяти (EEPROM)	1 кБ
Частота тактирования	16 МГц
Дисплей	Символьный 16x02; Подсветка: Синяя с белыми символами Контраст: Настраивается потенциометром Напряжение питания: 5В Интерфейс: I2C I2C адрес: 0x27

Выводы

АЦП бывают разных видов и являются необходимым компонентом при работе с датчиками, так как все датчики вырабатывают аналоговый сигнал, но некоторые из них имеют встроенный АЦП. Без аналогово-цифрового преобразователя невозможно выводить значения, как на дисплей, так и на компьютер. Так же выяснили схему работы ЭПП и характеристики всех его составляющих.

Что такое «Arduino»

Arduino – это инструмент для проектирования электронных устройств более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Это платформа, предназначенная для физических вычислений с открытым программным кодом, построенная на простой печатной плате с современной средой для написания программного обеспечения. ≈ 160р.

Arduino применяется для создания электронных устройств с возможностью приема сигналов от различных цифровых и аналоговых датчиков, которые могут быть подключены к нему, и управления различными исполнительными устройствами. Проекты устройств, основанные на Arduino, могут работать самостоятельно или взаимодействовать с программным обеспечением на компьютере. Программная часть Arduino состоит из интегрированной программной среды (IDE), позволяющей писать, компилировать программы, а также загружать их в аппаратуру. Для программирования Arduino используется язык C++.

1. Программирование Arduino

Структура программы Arduino в минимальном варианте состоит из двух частей `setup()` и `loop()`.

Функция `setup()` выполняется один раз, при включении питания или сбросе контроллера. В ней происходят начальные установки переменных, регистров. Функция должна присутствовать в программе, даже если в ней ничего нет.

После завершения `setup()` управление переходит к функции `loop()`. Она в бесконечном цикле выполняет команды, записанные в между фигурными скобками. Собственно эти команды и совершают все алгоритмические действия контроллера.

Для программирования используются несколько типов данных:

Тип данных	Разрядность, бит	Диапазон чисел
boolean	8	true, false
char	8	-128 ... 127
unsigned char	8	0 ... 255
byte	8	0 ... 255
int	16	-32768 ... 32767
unsigned int	16	0 ... 65535
word	16	0 ... 65535
long	32	-2147483648 ... 2147483647
unsigned long	32	0 ... 4294967295
short	16	-32768 ... 32767
float	32	-3.4028235+38 ... 3.4028235+38
double	32	-3.4028235+38 ... 3.4028235+38

Объявление переменных. Указывается тип данных, а затем имя переменной.

Все переменные должны быть объявлены до того как будут использоваться. Переменная может быть объявлена в любой ча-

сти программы, но от этого зависит, какие блоки программы могут ее использовать. Т.е. у переменных есть области видимости. Переменные, объявленные в начале программы, до функции void setup(), считаются глобальными и доступны в любом месте программы. Локальные переменные объявляются внутри функций или таких блоков, как цикл for, и могут использоваться только в объявленных блоках. Возможны несколько переменных с одним именем, но разными областями видимости.

Так же Arduino поддерживает некоторое число операций:

- Арифметические операции
 - = присваивание
 - + сложение
 - вычитание
 - * произведение
 - / деление
 - % остаток от деления
- Операции отношения
 - == равно
 - != не равно
 - < меньше
 - > больше
 - <= меньше или равно
 - >= больше или равно
- Логические операции
 - && логическое И
 - || логическое ИЛИ
 - ! логическое НЕ
- Операции над указателями
 - * косвенная адресация
 - & получение адреса переменной
- Битовые операции
 - & и
 - | или
 - ^ ИСКЛЮЧАЮЩЕЕ ИЛИ
 - ~ ИНВЕРСИЯ
 - << СДВИГ ВЛЕВО
 - >> СДВИГ ВПРАВО
- Операции смешанного присваивания
 - ++ + 1 к переменной
 - - 1 к переменной
 - += сложение
 - = вычитание
 - *= умножение
 - /= деление
 - %= остаток от деления
 - &= битовое И
 - |= битовое ИЛИ

Выбор вариантов, управление программой

Оператор IF проверяет условие в скобках и выполняет последующее выражение или блок в фигурных скобках, если условие истинно.

```
if (x == 5) // если x=5, то выполняется
z=0
```

```
z=0;
if (x > 5) // если x > 5, то выполняется
блок z=0, y=8;
{ z=0; y=8; }
```

IF ... ELSE позволяет сделать выбор между двух вариантов.

```
if (x > 5) // если x > 5, то выполняется
блок z=0, y=8;
```

```
{
z=0;
y=8;
}
else // в противном случае выполняется
этот блок
```

```
{
z=0;
y=0;
}
```

ELSE IF – позволяет сделать множественный выбор

```
if (x > 5) // если x > 5, то выполняется
блок z=0, y=8;
```

```
{
z=0;
y=8;
}
else if (x > 20) // если x > 20, выполняет
этот блок
```

```
{ }
else // в противном случае выполняется
этот блок
```

```
{
z=0;
y=0;
}
```

SWITCH CASE – множественный выбор. Позволяет сравнить переменную (в примере это x) с несколькими константами (в примере 5 и 10) и выполнить блок, в котором переменная равна константе.

```
switch (x) {
case 5 :
// код выполняется если x = 5
break;
case 10 :
// код выполняется если x = 10
break;
default :
// код выполняется если не совпало ни
одно предыдущее значение
break;
}
```

Цикл FOR. Конструкция позволяет организовывать циклы с заданным количеством итераций. Синтаксис выглядит так:

```
for ( действие до начала цикла;
      условие продолжения цикла;
      действие в конце каждой итерации ) {
    // код тела цикла
}
```

Пример цикла из 100 итераций.
for (i=0; i < 100; i++) // начальное значение 0, конечное 99, шаг 1

```
{
  sum = sum + I;
}
```

Цикл WHILE. Оператор позволяет организовывать циклы с конструкцией:

```
while ( выражение )
{
  // код тела цикла
}
```

Цикл выполняется до тех пор, пока выражение в скобках истинно. Пример цикла на 10 итераций.

```
x = 0;
while ( x < 10 )
{
  // код тела цикла
  x++;
}
```

DO WHILE – цикл с условием на выходе.

```
do
{
  // код тела цикла
} while (выражение);
```

Цикл выполняется пока выражение истинно.

BREAK – оператор выхода из цикла. Используется для того, чтобы прервать выполнение циклов for, while, do while.

```
x = 0;
while ( x < 10 )
{
  if ( z > 20 ) break; // если z > 20, то выйти из цикла
  // код тела цикла
  x++;
}
```

CONTINUE – пропуск операторов до конца тела цикла.

```
x = 0;
while ( x < 10 )
{
  // код тела цикла
  if ( z > 20 ) continue; // если z > 20, то вернуться на начало тела цикла
  // код тела цикла
  x++;
}
```

2. Программа ЭПП

Программа (прошивка) для Arduino называется **скетч**. Скетч ЭПП был составлен из:

- скетча для датчика температуры и влажности
- скетча для барометра

⇒ Скетчи были взяты из Интернета

Скетчи были объединены и изменены для вывода значений на дисплей иных показателей отличных от показателей в Интернете. Добавлены формулы для выведения дельты давления и вывода ее в процентах, показывая шанс дождя.

Объединенный скетч для вывода текущих значений с датчиков:

```
#include <Wire.h> // Добавляем необходимые библиотеки
#include <LiquidCrystal_I2C.h>
#include <BMP085.h>
#include <DHT.h>
#define DHTPIN 2 // пин для сигнала поступающего с датчика
#define DHTTYPE DHT22 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
const int SVET = 9;
BMP085 dps = BMP085();
long Temperature = 0, Pressure = 0;
byte degree[8] = // кодируем символ гра-
```

```
дуса
{ B00111,
  B00101,
  B00111,
  B00000,
  B00000,
  B00000,
  B00000, };
LiquidCrystal_I2C lcd(0x27, 16, 2); // Задаем адрес и размерность дисплея
void setup()
{
  Serial.begin(9600);
  Wire.begin();
  delay(1000);
  dps.init();
  lcd.begin(); // Инициализация lcd
  lcd.createChar(5, degree);
  pinMode(SVET, OUTPUT);
  analogWrite(SVET, 255);
  dht.begin();
}
void loop() {
  dps.getPressure(&Pressure);
  dps.getTemperature(&Temperature);
  Serial.print("Pressure(mm Hg:");
  Serial.print(Pressure/133.3); //Выводим давление в мм.рт ст
  Serial.print("Temp:");
  Serial.println(Temperature*0.1); // Температуру в градусах Цельсия
  delay(2000);
  analogWrite(SVET, 100);
```

```

delay(2000);
float h = dht.readHumidity();
float t = dht.readTemperature();
lcd.setCursor(11, 0); // Выводим показани
ния влажности и температуры
lcd.print(«F»); lcd.setCursor(12, 0);
lcd.print(«=»); lcd.setCursor(13, 0); lcd.
print(h,1); lcd.setCursor(15, 0); lcd.
print(«%»); lcd.setCursor(10, 1); lcd.
print(«t»); lcd.setCursor(11, 1); lcd.
print(«=»); lcd.setCursor(12, 1); lcd.
print(t,1); lcd.setCursor(14, 1); lcd.
write((byte)5); lcd.setCursor(15, 1); lcd.
print(«C»); lcd.setCursor(0,0); lcd.print(«P»);
lcd.setCursor(1,0); lcd.print(«=»); lcd.
print(Pressure/133.3); //Выводим давление в
мм.рт ст); }
Формулы для вывода дельты:
void loop() {
if (wake flag) {
delay(500);
pressure = aver_sens(); // найти текущее
давление по среднему арифметическому
for (byte i = 0; i < 5; i++) // счетчик от 0 до
5 (да, до 5. Так как 4 меньше 5)
{
pressure_array[i] = pressure_array[i + 1];
// сдвинуть массив давлений кроме
последней ячейки на шаг назад
}
pressure_array[5] = pressure; // последний
элемент массива теперь – новое давление

```

```

sumX = 0;
sumY = 0;
sumX2 = 0;
sumXY = 0;
for (int i = 0; i < 6; i++) { // для всех эле-
ментов массива
sumX += time_array[i];
sumY += (long)pressure_array[i];
sumX2 += time_array[i] * time_array[i];
sumXY += (long)time_array[i] * pressure_
array[i]; }
a = 0;
a = (long)6 * sumXY; // расчет коэффи-
циента наклона прямой
a = a - (long)sumX * sumY;
a = (float)a / (6 * sumX2 - sumX * sumX);
delta = a * 6; } // расчет изменения давления
Формулы для выведения дельты
в процентах:
{ if (delta (0) = 0%) ;
delta ((300) = 100%) ;
float chance = bmp.readdelta(%) ;
lsd.Print (chance.1) ; }

```

Выводы

Разобравшись с методами программирования микроконтроллера, были выведены формулы для выведения показаний на дисплей, преобразования значений в дельту и проценты для предсказания ухудшения погоды заблаговременно и выведения шанса выпадения осадков.