

Цифровой термометр

Максим Хусаинов

Август 2024

1 Описание

Цифровой термометр состоит из двух частей, устройство, или прибор, и сервис НТТР. Устройство «разговаривает» с сервисом посредством запросов НТТР, передавая на сервис текущую температуру. Сервис собирает и сохраняет данные: время и температуру.

Прибор содержит экран, кнопку и датчик температуры. Датчик температуры подключается к прибору через провод. Сервис находится на сервере и имеет доступ к системе архивов.

При включении цифрового термометра, прибор каждую секунду считывает температуру с датчика температуры и отображает её на экране. Также прибор подключается к беспроводной сети. При нажатии на кнопку, начинается опыт: прибор отправляет запрос на сервис о начале опыта и в дальнейшем отправляет каждую секунду запросы с текущей температурой. Повторное нажатие на кнопку останавливает опыт, т. е. прибор перестаёт посылать запросы на сервис с текущей температурой.

Сервис, при запросе о начале опыта, создает новый архив. При запросе с температурой, сервис сохраняет её вместе со временем получения запроса в созданный архив.

Датчик температуры работает в диапазоне от -55°C до 125°C . Точность $\pm 0,5^{\circ}\text{C}$ в диапазоне -10°C до 85°C . Среднее потребление прибора 0,07 А или 0,23 Вт.

Далее документ содержит четыре раздела. В разделе 2 «Устройство» описывается прибор и делится на два подраздела. В подразделе 2.1 «Материальная часть» дается детальный разбор составных частей прибора и схема их соединений. Дополнительно, в подподразделе 2.1.1 «Аккумуляторный блок питания» предоставляется вариант автономии устройства в плане энергии. В подразделе 2.2 «Программная часть» описывается алгоритм работы прибора и прилагается блок-схема этого алгоритма. В раздел 3 «Сервис НТТР» описывается алгоритм работы сервера. В разделе 4 «Установка» даются инструкции по установке программной части прибора и сервиса. В подразделе 4.1 «Прошивка прибора» содержится инструкция по прошивке прибора и в подразделе 4.2 «Запуск сервиса» инструкция по запуску сервиса. В разделе 5 «Заключение» приводятся возможный вариант использования прибора, а также некоторые идеи для его усовершенствования.

2 Устройство

2.1 Материальная часть

Прибор цифрового термометра состоит из следующих составных частей, или компонентов:

1. датчика температуры DS18B20;
2. микроконтроллера ESP8266-12E;
3. экрана OLED (128 × 64 px);
4. стабилизатора напряжения с 5 В до 3,3 В или блока питания с аккумулятором;
5. резисторов и нажимной кнопки.

Также, для прошивки и программирования микроконтроллера, требуется преобразователь CP2102 USB-UART.

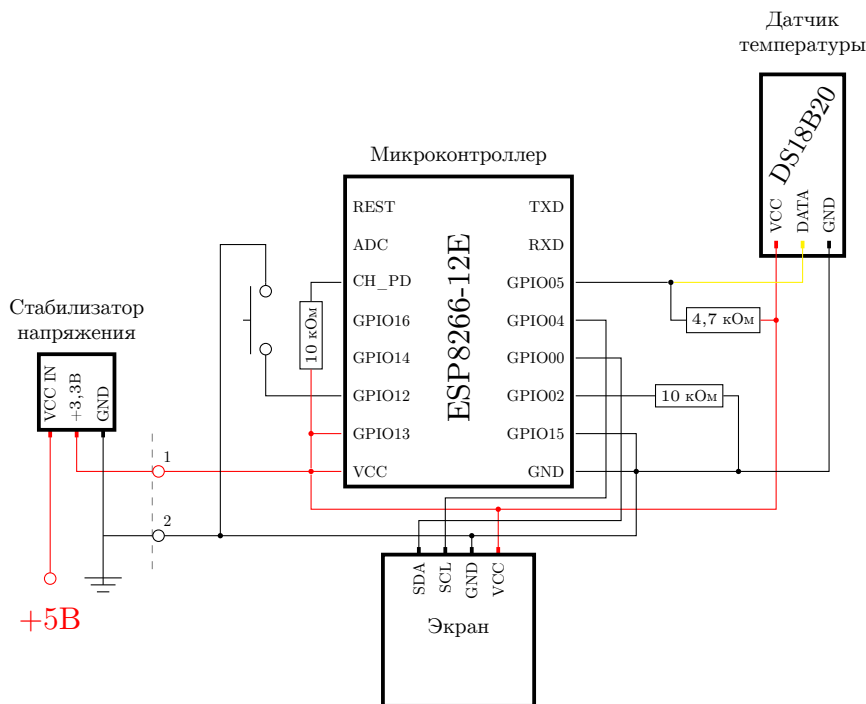


Рис. 1: Схема компонентов прибора цифрового термометра.

На рис. 1 показаны компоненты прибора цифрового термометра и их соединения. Стабилизатор напряжения получает 5 В, например от источника питания порта USB, и преобразует в напряжение в 3,3 В, от которого питаются все остальные компоненты прибора, контакты 1 и 2. Стабилизатор можно заменить аккумуляторным блоком питания, который описывается ниже в подразделе 2.1.1.

Для запуска микроконтроллера ножка CH_PD связывается через резистор 10 кОм с напряжением 3,3 В, ножка GPIO13 связывается с напряжением 3,3 В, ножка GPIO15 связывается с землей и ножка GPIO02 связывается через резистор 10 кОм с землей.

Микроконтроллер соединяется с выводом DATA датчика температуры через ножку GPIO05. Вывод DATA датчика температуры связывается с напряжением 3,3 В резистором 4,7 кОм. Микроконтроллер связывается с ножкой SCL экрана через ножку GPIO04, с ножкой SDA экрана через ножку GPIO00. Нажимная кнопка подает отрицательный сигнал на ножку GPIO12 микроконтроллера.

2.1.1 Аккумуляторный блок питания

Для того, чтобы устройство было автономным в плане энергии, вместо стабилизатора напряжение можно использовать аккумуляторный блок питания.

Аккумуляторный блок питания состоит из следующих трёх компонентов:

1. литий-полимерного аккумулятора с номинальным напряжением 3,7 В;
2. стабилизатора напряжения TPS63020 с выходным напряжением 3,3 В и диапазоном входного напряжения 1,8 – 5 В;
3. модуль зарядки аккумулятора TP4056 (входное напряжение 5 В от USB, максимальный заряд напряжения батареи 4,2 В).

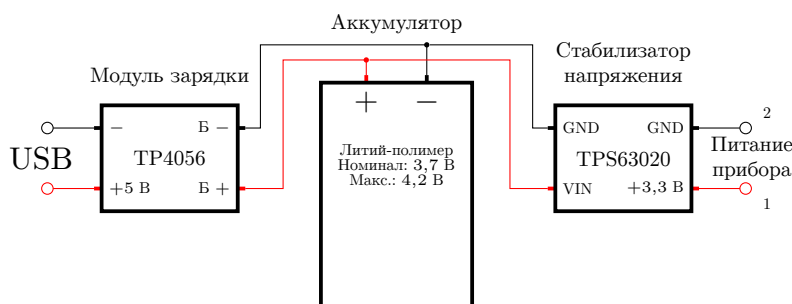


Рис. 2: Схема компонентов аккумуляторного блока питания.

На рис. 2 показаны компоненты аккумуляторного блока питания и их соединения. Модуль зарядки получает внешнее питание (5 В) от порта USB. Выход модуля зарядки соединен с аккумулятором и входом стабилизатора напряжения.

Вход стабилизатора напряжения получает напряжение в диапазоне от 1,8 В до 5 В и преобразует его, повышая или понижая, в 3,3 В на своем выходе. Выход стабилизатора напряжения соединяется с контактами 1 и 2 рис. 1, через которые запитывает прибор.

При подключении внешнего питания к модулю зарядки, модуль начинает заряжать батарею и одновременно питать стабилизатор напряжения. Когда батарея заряжена и модуль зарядки отключен от внешнего питания, то стабилизатор напряжения питает батарею.

В случае, если ёмкость батареи 1800 мА час, то полного заряда хватает на 7 часов работы прибора.

2.2 Программная часть

Программная часть касается программирования микроконтроллера. Микроконтроллер использует платформу Espruino, которая предоставляет среду выполнения JavaScript-кода.

На рис. 3 показан алгоритм программной части. Прибор может находиться в двух режимах: ожидания и опыта. При запуске микроконтроллер считывает значение режима из хранилища и, в случае, если значение равно «опыт», то устанавливает режим опыта. Иначе устанавливает режим ожидания. Установление режима означает присваивание соответствующего значения переменной «режим». Далее в микроконтроллере можно рассмотреть следующие потоки:

- Подключение микроконтроллера к сети WIFI.
- Соединение микроконтроллера с датчиком температуры.
- Соединение микроконтроллера с экраном.

После подключения к экрану запускаются два дополнительных потока:

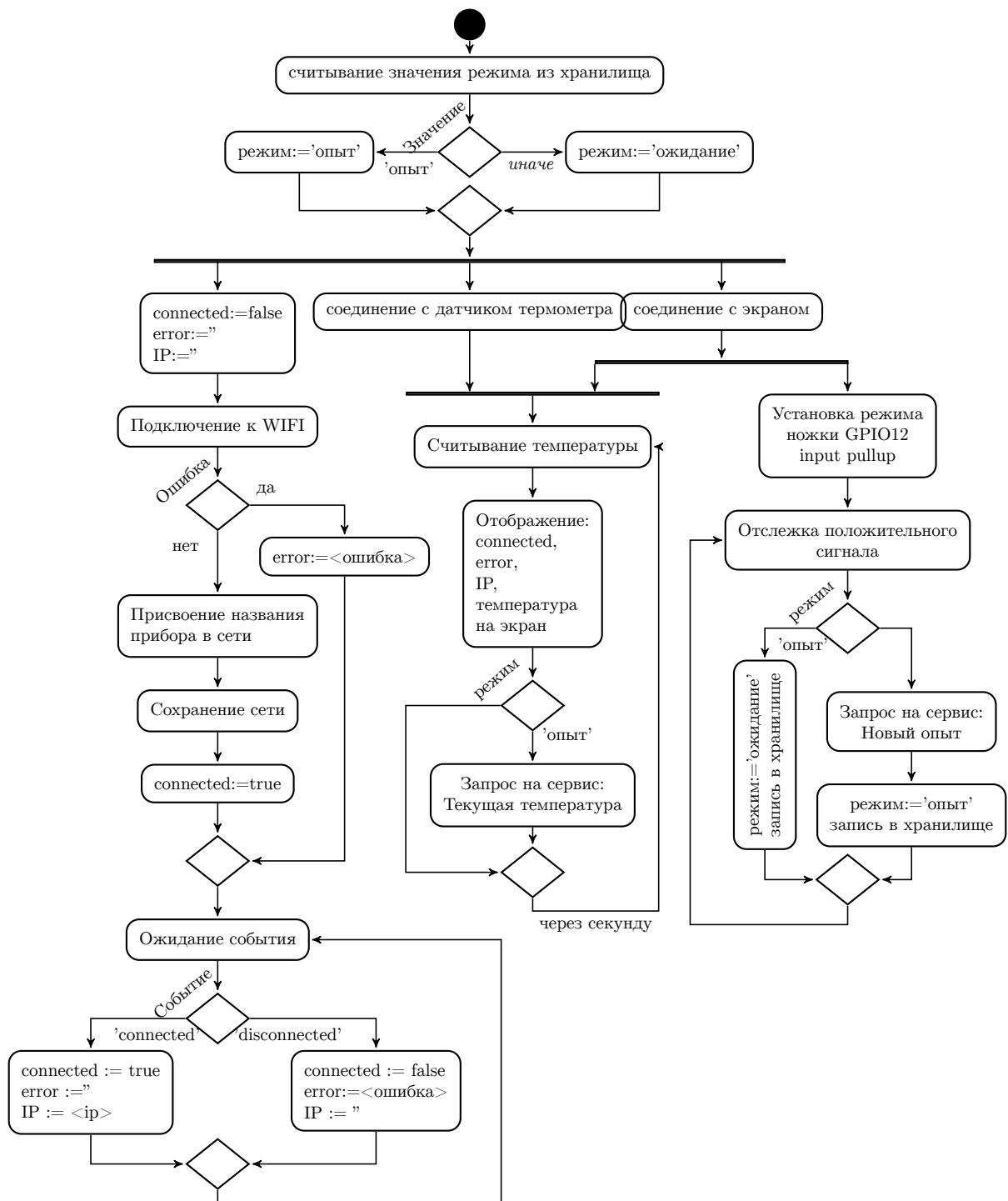


Рис. 3: Блок-схема алгоритма прибора.

- Первый поток касается ежесекундного считывания температуры.
- Второй поток касается отслеживания нажатия кнопки.

В потоке подключения к сети WIFI присваиваются значения переменным 'connected', 'error' и 'IP'. Данные подключения к сети WIFI, т. е. название точки доступа и пароль, прописаны в самом коде. Переменная 'connected' содержит логическое значение и при успешном подключении к сети принимает значение истинности. Переменная 'error' содержит строку ошибки в случае ошибки подключения. Переменная 'IP' содержит строку с адресом IP, который получил прибор при успешном подключении к сети. Эти переменные используются в потоке ежесекундного считывания температуры.

В потоке соединения с датчиком температуры микроконтроллер подключается к ножке GPIO05, используя протокол OneWire (однопроводной интерфейс). При успешном подключении, поток сливается с потоком соединения с экраном и переходит в поток ежесекундного считывания температуры.

Микроконтроллер соединяется с экраном, используя протокол I2C (Inter-Integrated Circuit), подключая линию данных (SDA) экрана к ножке GPIO00, а линию тактирования (SCL) экрана к ножке GPIO04. При успешном подключении, поток сливается с потоком соединения с датчиком температуры и переходит в поток ежесекундного считывания температуры.

В потоке ежесекундного считывания температуры микроконтроллер с интервалом в одну секунду считывает информацию с датчика температуры. Далее, микроконтроллер отображает на экране считанную температуру, переменные потока подключения к сети WIFI, а также режим опыта. Температура отображается крупным шрифтом в центре экрана. Переменные подключения к сети WIFI отображаются в начале верхней строки: успешное подключение к сети и адрес IP прибора, или же строка ошибки подключения к сети. Режим опыта отображается в конце верхней строки словом "Op". Если микроконтроллер находится в режиме опыта, то он посылает запрос на сервис с текущей температурой. Таким образом, в режиме опыта микроконтроллер ежесекундно посылает запрос на сервис с текущей температурой. Ссылка на сервис HTTP прописана в коде.

В потоке отслеживания нажатия кнопки микроконтроллер отслеживает отрицательный сигнал на ножке GPIO12. Для этого он устанавливает ножку в режиме input pullup и прерывание (watch). При нажатии кнопки, на ножку поступает отрицательный сигнал и срабатывает прерывание. Для избежания ложных срабатываний прерывания, например придребезге контактов кнопки, указывается параметр "debounce". Далее, рассматриваем не ложное прерывание. Когда микроконтроллер находится в режиме опыта, то он устанавливает режим ожидания. Когда микроконтроллер находится в режиме ожидания, то он посылает запрос «Новый опыт» на сервис и устанавливает режим опыта. В этих случаях установление режима означает не только присваивание соответствующего значения переменной «режим», но и запись этого значения в хранилище. Таким образом, нажатие кнопки является интерфейсом для переключения режима прибора с ожидания на опыт и наоборот.

3 Сервис HTTP

Сервис написан на языке Python и реализует простой сервис HTTP, в котором обрабатываются два запроса типа GET: «Текущая температура» и «Новый опыт».

Сервис запускается функцией 'run', которая принимает аргументы 'serverClass' и 'handlerClass'. По умолчанию аргумент 'serverClass' имеет значение 'HTTPServer',

Для того, чтобы обрабатывать полученные запросы, сервис расширяет класс 'BaseHttpRequestHandler' классом 'HttpGetHandler', в котором, в методе 'doGET', реализуется обработка запросов типа GET. Метод 'doGET' обрабатывает полученные запросы, выделяя из них необходимые параметры «температура» или «опыт». Если в запросе присутствует параметр «опыт», то выполняется метод 'novijOpyt'. Иначе, если в запросе присутствует параметр «температура», то выполняется метод 'pishiOpyt'.

Метод 'pishiOpyt' записывает в созданный ранее архив время получения запроса и полученную температуру в соответствующие столбцы.

4.1 Прошивка прибора

Микроконтроллер

ESP8266-12E

Стабилизатор напряжения

Преобразователь USB

5 B
GND
TXO
RXI
DTR

CP2102

+5 B

10 кОм

10 кОм

REST
ADC
CH_PD
GPIO16
GPIO14
GPIO12
GPIO13
VCC

TXD
RXD
GPIO05
GPIO04
GPIO00
GPIO02
GPIO15
GND

VCC IN
+3,3 B
GND

Программа микроконтроллера написана на языке JavaScript и работает через платформу Espruino. Это значит, что на микроконтроллере должна присутствовать прошивка платформы Espruino (espruino 2v21).

- компьютер с операционной системой Windows 10,
- архив с прошивкой платформы Espruino ('espruino_2v21.zip'),
- интерпретатор Python (версия 3.8.9),

- архив с драйвером преобразователя ('CP210x_Universal_Windows_Driver.zip').

Далее следует пошаговая инструкция:

1. Скачиваем нужные архивы на компьютер.
2. Устанавливаем на компьютер драйвер преобразователя из архива 'CP210x_Universal_Windows_Driver.zip'.
3. Устанавливаем на компьютер Python (если он не установлен).
4. Устанавливаем на компьютер модуль питона "esptool" командой:

```
pip install esptool
```

5. Распаковываем прошивку платформы Espruino в какую-нибудь папку на компьютере (например в папку c:\espruino\).
6. Подключаем компьютер к преобразователю (через USB кабель). В диспетчере устройств на компьютере в портах (COM и LPT) следует обнаружить порт подключения (например COM6).
7. На микроконтроллере замыкаем ножку GPIO00 с землей.
8. На компьютере через командную строку переходим в подпапку "espruino_2v21_esp8266_4mb" папки, куда распаковали прошивку платформы Espruino, и задаём следующую команду:

```
c:\espruino\espruino_2v21_esp8266_4mb>esptool --port COM6 --baud 460800 write_flash  
↪ --flash_freq 80m --flash_mode qio --flash_size 4MB-c1 0x0000 boot_v1.6.bin 0x1000  
↪ espruino_esp8266_user1.bin 0x3FC000 esp_init_data_default.bin 0x3FE000 blank.bin
```

Результат команды должен выглядеть следующим образом:

```
esptool.py v4.7.0  
Serial port COM6  
Connecting....  
Detecting chip type... Unsupported detection protocol, switching and trying again...  
Connecting...  
Detecting chip type... ESP8266  
Chip is ESP8266EX  
Features: WiFi  
Crystal is 26MHz  
MAC: 58:bf:25:d7:5c:89  
Uploading stub...  
Running stub...  
Stub running...  
Changing baud rate to 460800  
Changed.  
Configuring flash size...  
Flash will be erased from 0x00000000 to 0x00000fff...  
Flash will be erased from 0x00001000 to 0x0008dfff...  
Flash will be erased from 0x003fc000 to 0x003fffff...  
Flash will be erased from 0x003fe000 to 0x003fefff...  
Flash params set to 0x006f  
Compressed 3856 bytes to 2763...  
Wrote 3856 bytes (2763 compressed) at 0x00000000 in 0.1 seconds (effective 254.1 kbit/s)...  
Hash of data verified.  
Compressed 577188 bytes to 369191...  
Wrote 577188 bytes (369191 compressed) at 0x00001000 in 9.3 seconds (effective 496.7 kbit/s)...  
Hash of data verified.  
Compressed 128 bytes to 75...  
Wrote 128 bytes (75 compressed) at 0x003fc000 in 0.1 seconds (effective 18.4 kbit/s)...  
Hash of data verified.  
Compressed 4096 bytes to 26...  
Wrote 4096 bytes (26 compressed) at 0x003fe000 in 0.1 seconds (effective 633.4 kbit/s)...  
Hash of data verified.
```

Таким образом мы прошили микроконтроллер прошивкой платформы Espruino.

Чтобы запрограммировать микроконтроллер, понадобится среда Espruino Web IDE, которую следует установить на компьютер. Запускаем среду, открываем архив с программным кодом, подключаемся к микроконтроллеру (предварительно подключив компьютер к преобразователю через USB кабель) и посылаем программный код на микроконтроллер. При успешной прошивке прибор постарается подключиться к беспроводной сети WIFI, а при нажатии на кнопку отослать запрос на сервис. Данные подключения к сети и ссылки на сервис необходимо изменить в начале программного кода прошивки, где они выражены переменными WIFI_NAME, WIFI_OPTIONS и serviceHost соответственно.

4.2 Запуск сервиса

Сервис запускается на сервере. На сервере необходимо установить интерпретатор Python версии 3 (если он не установлен). Установка сервиса – это помещение кода сервиса, архив 'Сервис HTTP.py', в какую-нибудь папку на сервере, в которой пользователь имеет доступ записи.

Запуск сервиса осуществляется следующей командой:

```
python "Сервис HTTP.py"
```

Сервис работает постоянно, т. е. запуск сервиса не заканчивается, если его не закончить принудительно. Сервис создает и записывает архивы опытов в папку, из которой он запущен.

5 Заключение

В этом документе описываются способ создания прибора «Цифровой термометр». Прибор можно использовать в лабораторных физических опытах, таких как кристаллизация и плавление жидкостей, в частности воды. У прибора выносной термометр, позволяющий поместить его в жидкость, заморозить вместе с жидкостью и, подключив термометр к прибору, провести опыт плавления. Так как прибор может питаться от аккумулятора, его можно поместить в морозильную камеру и провести опыт кристаллизации. Данные, собранные прибором, можно использовать для построения графиков изменения температуры во времени.

Прибор можно усовершенствовать, добавив ему функционал режима настройки данных подключения к сети WIFI и ссылки на сервис. Этого можно достичь следующим образом: при включении прибора и одновременном нажатии на кнопку устанавливать микроконтроллер в режиме точки доступа беспроводной сети WIFI. В этом режиме микроконтроллер будет выполнять роль сервиса HTTP и предоставлять страницу с формой настроек. В таком режиме пользователь подключается к точке доступа (без пароля), переходит на страницу с формой настроек и указывает настройки. Также можно добавить настройку интервала периодического считывания температуры или интервала отправки запроса.