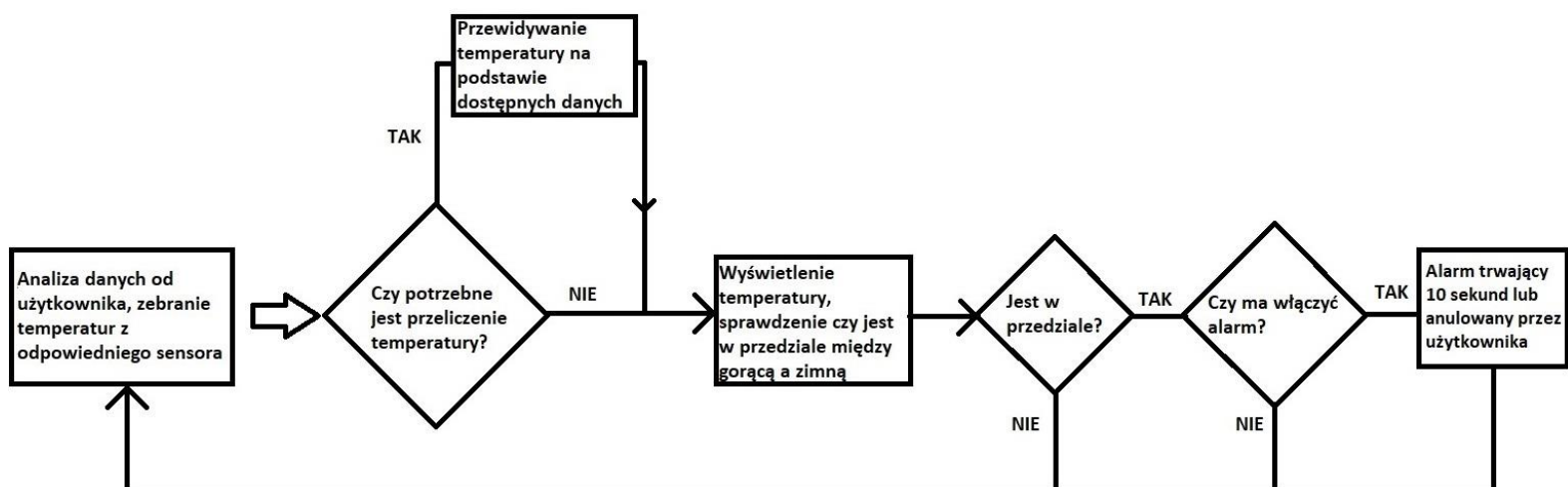


Specjalizowany czujnik temperaturowy - mikroprocesorowy analizator cieplny



To urządzenie zostało stworzone z myślą o osobach, które nie chcą czekać aż ich gorący napój będzie dobry do picia. Wystarczy włączyć urządzenie, postawić napój i... poczekać!

Zasada działania (schematy uproszczone):

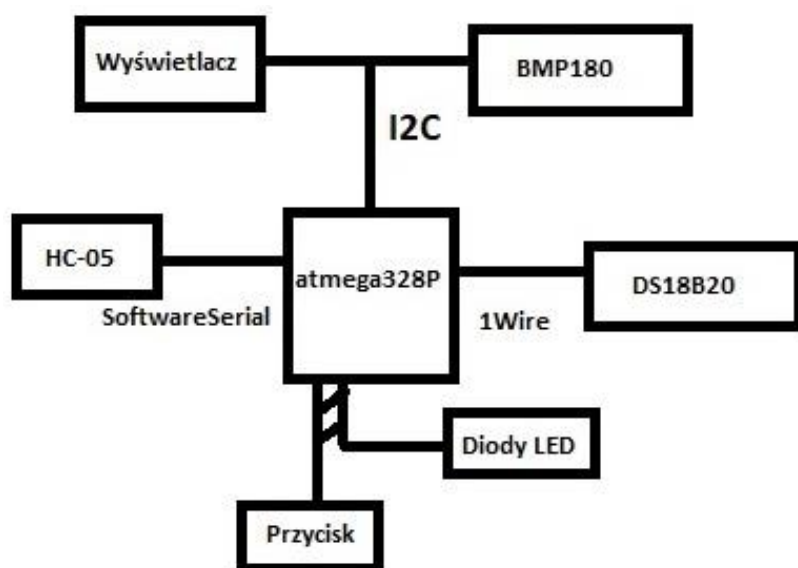


Oczywiście ten schemat jest jedynie uproszczeniem, a każda akcja wykonywana w blokach rozdzielona jest na kilka podfunkcji.

W międzyczasie program sprawdza czy oczekuje jakaś komenda i zbiera akcje przycisku.

Sercem urządzenia jest mikrokontroler atmega328P (Atmel), który mimo małych rozmiarów posiada wiele możliwości.

Schemat budowy:

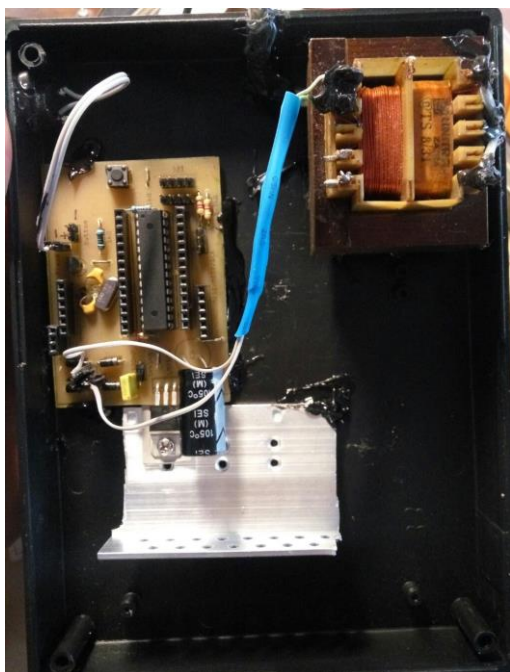


Zastosowane komponenty (widoczne na schemacie) i zastosowanie:

- Wyświetlacz HD44780 z interfejsem I2C – wyświetlanie danych
- Czujnik temperatury BMP180 firmy Bosch (I2C) – główny czujnik
- Moduł Bluetooth HC-05 – komunikacja z PC
- Czujnik DS18B20 – sonda pomocnicza

Zasilanie doprowadzane jest z transformatora 230-30V i stabilizowane do napięcia 5V które atmega toleruje.

Dodatkowo urządzenie posiada około dwumetrowy przewód zasilający z włącznikiem. Dzięki temu można urządzenie kompletnie wyłączyć.

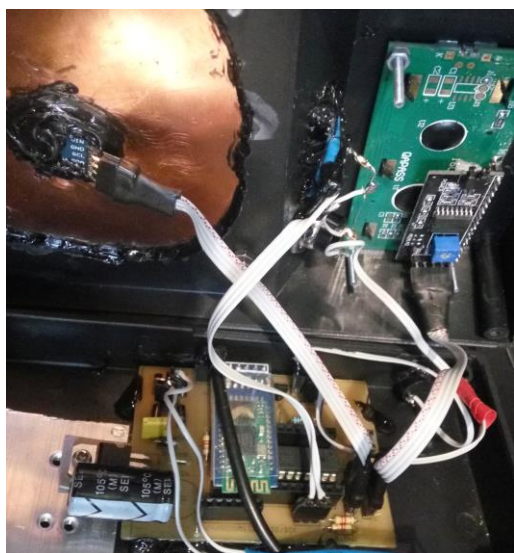


Komponenty zostały przytwierdzone na gorący klej, ponieważ obudowa i konstrukcja płytki nie pozwalały umieścić jej w przewidzianym przez producenta obudowy miejscu.

Wyprowadzenia transformatora zostały zalane klejem aby uniknąć ewentualnego porażenia podczas pracy z otwartą obudową.

Specjalnie przygotowane przewody połączeniowe pozwalają w prosty sposób połączyć płytę główną z komponentami. Płyta została wyposażona w dedykowane gniazdo dla modułu Bluetooth, sondy pomocniczej oraz interfejs SPI.

W razie potrzeby można połączyć się z mikrokontrolerem przez programator.



Krótko o kodzie źródłowym dla mikrokontrolera

Program został napisany używając kompilatora Arduino. Używa standardowych bibliotek Arduino oraz delikatnie zmodyfikowanej przeze mnie biblioteki ClickButton (na licencji GPLv3, autor: raronzen@gmail.com).

Program (ogółem) działa tak jak na schemacie wyżej, ale w rzeczywistości jest jeszcze kilka skomplikowanych funkcji.

Została zaimplementowana funkcja nazwana btctl() która zarządza poszczególnymi elementami zasilania a także przekazywaniem danych. Przykładowo możemy za jej pośrednictwem wyłączyć urządzenie, zmienić temperaturę graniczną dla gorącej albo odczytać obecne ustawienia.

Obsługa komend specyficznych dla modułu Bluetooth (komend AT) również została dodana aby w razie potrzeby była możliwość zmiany jego ustawień (np. nazwa, pin parowania etc. Więcej informacji znajduje się w nocie producenta).

Kod został podzielony na dwa pliki: KubkoinoDNO.ino oraz vars.h. Nie ma to żadnego praktycznego powodu, ale przeniesienie deklaracji zmiennych (tj. tablica String'ów zawierająca tryby) zwiększyło czytelność kodu.

Na chwilę obecną Arduino trzyma wszystkie dane w pamięci RAM, przez co po odłączeniu go od zasilania wszystkie dane znikają.

Aby uprościć konfigurację urządzenia (i ułatwić przywracanie danych) stworzony został program Kubkuino (wymaga bluetooth i .NET Framework w wersji >= 3.5).

Krótki spis funkcji programu Kubkuino

- Ustawianie temperatur granicznych,
- Zmiana trybu pomiaru
- Zmiana materiału i rodzaju dna naczynia
- Włączanie i wyłączanie alarmu
- Zapis ulubionego ustawienia
- Reset fabryczny, zmiana nazwy i kodu parowania BT
- Konsola do własnej komunikacji z urządzeniem

Lista komend do komunikacji z urządzeniem

X - Pierwszy argument

Y - Drugi argument

e; - Nawiązanie połączenia. Urządzenie zwróci „HELLO\r\n” (znaki nowej linii są potrzebne do prawidłowej komunikacji!). Nie jest szczególnie potrzebne, ale urządzenie zasygnalizuje że jest połączone.

qquit; - Rozłączenie. Musi zostać wysłane, w innym razie komputer może nie zamknąć lokalnego portu COM i uniemożliwi następne połączenia.

CXX.X; - Ustawienie temperatury zimna*

HXX.X; - Ustawienie temperatury ciepła*

MX; - Ustawia materiał*:

S[komenda AT]; - Przesyła komendę AT do modułu Bluetooth. Zwraca odpowiedź modułu.

d; - Reset modułu Bluetooth, ustawienie domyślnej nazwy i hasła (określonej w kodzie źródłowym)

r0; - Zwraca temperaturę czujnika dla bieżącego trybu

r1; - Zwraca poprawioną temperaturę dla bieżącego trybu

t; - Zwraca stan jako char (H - Hot, R - Ready, C - Cold)

DXY; - Moduł zarządzania komponentami:

- 0Y* - LCD (1 - włącz, 0 - wyłącz)
- 1Y* - Podświetlenie (1 - włącz, 0 - wyłącz)
- 2Y - BT (1 - włącz, 0 - wyłącz)
- 3Y* - LED (1 - włącz, 0 - wyłącz)
- 4Y* - Wyciszenie (1 - wyciszony, 0 - głośny)
- 5Y* - Rodzaj dna (1 - płaskie, 0 - wklęsłe)
- 6 - Status LCD, Podświetlenie, LED, Alarm, Dno (płaskie) (1 - włączony, 0 - wyłączony)
- 7Y - Tryb pomiaru:
 - 0 - automatyczny wybór czujnika (wklęsłe dno gdy inny tryb niż aluminium)

- 1 - pomiar temperatury dna bez poprawki (wkłęśłe dno, każdy tryb, temperatura na oko)
- 2 - tylko zewnętrzny czujnik (DS18B20)
- 3 - pomiar temperatury dna
- ? - zwraca aktywny tryb (0-3)
- 8 - Przejście w tryb uśpienia

*Jak jako argument poda się znak zapytania (?) zostanie zwrócony stan pojedynczego modułu, np. H? / D0? / D3? / M?

Uwaga! Każda komenda jest zakończona średnikiem ';'!

W przypadku łączenia się innym programem należy ustawić znaki końca linii na CRLF (\r\n).

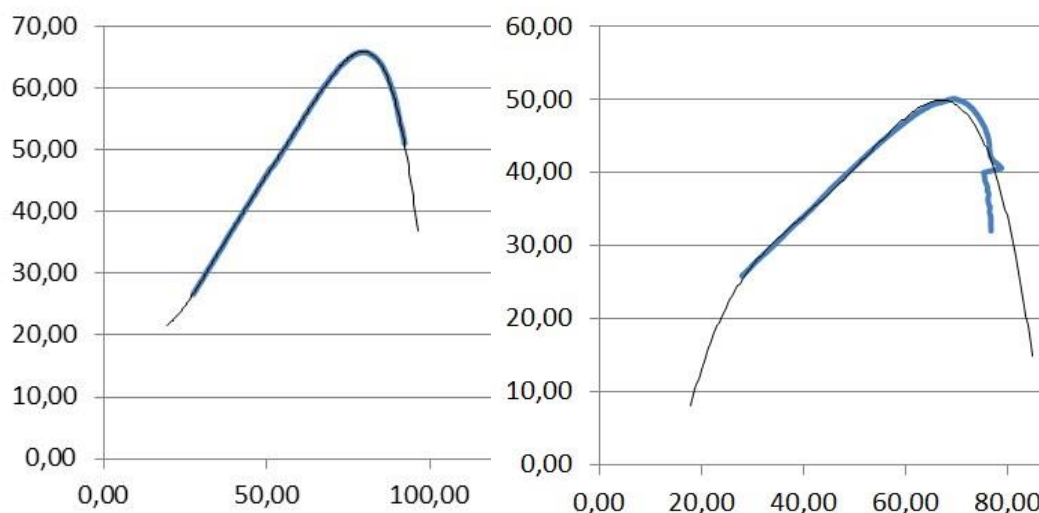
Przykładowa komenda zmiany pinu parowania modułu BT za pośrednictwem programu Kubkuino: SAT+PSWD=1234;

Problemy napotkanie w trakcie rozwoju projektu

Urządzenie jest zaprojektowane aby odczytywać temperaturę dna i szacować temperaturę napoju.

Już w pierwszych testach ukazał się problem – bezwładność cieplna naczynia i często niewystarczające jego właściwości termoprzewodzące.

Tutaj znajdują się dwa wykresy zależności temperatury dna od temperatury napoju (czas nieokreślony).



Wykres po lewej stronie przedstawia średnią z 5 pomiarów dla naczynia aluminiowego o płaskim dnie, a z prawej – ceramicznego również o płaskim dnie.

Biorąc ten fakt pod uwagę nie da się jednoznacznie określić temperatury napoju, ale tylko ją oszacować.

Aby otrzymać wzór funkcji do szacowania temperatury, w programie Excel wygenerowałem wykres (oś x – napój, y – dno) na który nałożyłem wielomianową linię trendu. Stopień wielomianu dopasowałem patrząc na prognozę (10 okresów w tył i 10 w przód). W ten sposób wygenerowany został wzór funkcji.

Aby wzór działał poprawnie, jego dokładność została zwiększona do 10 znaków po przecinku.

Przy obecnej konstrukcji metoda ta daje najlepsze rezultaty.

Informacje dodatkowe

Istnieje możliwość pobrania kodu źródłowego oraz materiałów pomocniczych.

Strona domowa projektu:

<https://github.com/marecl/Kubkuino>

Załączniki:

1. Instrukcja obsługi specjalizowanego czujnika temperatury
2. Instrukcja obsługi programu Kubkuino