

Wariometr dla paralotniarzy

Wariometr – znany również jako wskaźnik prędkości wznoszenia i zniżania, wskaźnik prędkości pionowej (VSI) – jest jednym z przyrządów pokładowych w pojazdach latających (i nie tylko – oprócz samolotów czy szybowców, znajdziemy go także w motolotniach, u paralotniarzy, skoczków spadochronowych etc). Przyrząd ten informuje pilota o prędkości zniżania lub wznoszenia się. Zazwyczaj jest on podłączony do zewnętrznego źródła ciśnienia, np. poza kadłubem statku.

W locie pojazdem z napędem pilot często korzysta z wariometru, aby upewnić się, że utrzymany jest lot poziomy, szczególnie podczas manewrów. Podczas szybowania przyrząd jest używany w sposób ciągły podczas normalnego lotu, często z dodatkowymi informacjami dźwiękowymi, aby poinformować pilota np. o opadaniu. Podobnie jest w przypadku paralotni, gdzie ciągłe monitorowanie prędkości pionowej jest gwarantem bezpiecznego lotu.

Autor projektu, Thomas Laireiter, paralotniarz i programista, kilka lat temu skonstruował swój pierwszy wariometr, opierając się na konstrukcji innego twórcy z portalu Instructables. Urządzenie działało bez zarzutu, ale autor poniższej konstrukcji postanowił zmienić kilka rzeczy. System zasilany był baterią 9 V, która zajmowała dużo miejsca, przez co obudowa układu była nieporęczna. Często w najbardziej obiecujący dzień lotów bateria się wyczerpywała i zawsze trzeba było mieć przy sobie zapasową baterię, inaczej urządzenie nie działało.

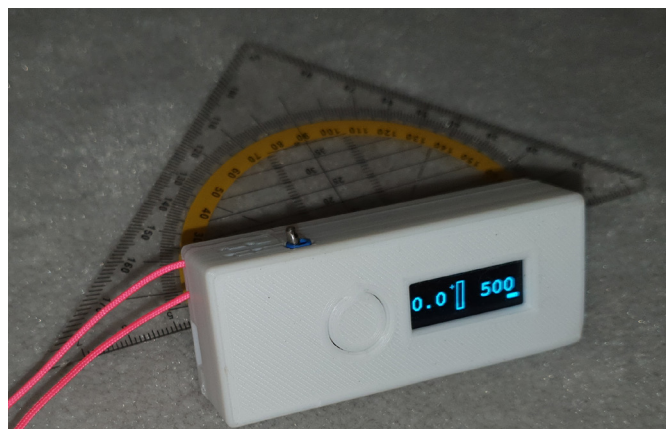
Dlatego też Thomas postanowił zaprojektować własny wariometr, inspirowany innymi konstrukcjami tego rodzaju urządzeń. Głównym celem projektu było zmniejszenie urządzenia i dodanie akumulatora z układem ładowania, zamiast zasilania baterijnego. Zmieniony został także wyświetlacz (na moduł OLED ze sterownikiem), co oznaczało, że konieczne jest napisanie nowego oprogramowania układowego dla mikrokontrolera.

Rezultatem tych prac jest kompaktowy wariometr z niewielkim wyświetlaczem i zintegrowanym akumulatorem, który ładować można poprzez standardowy port microUSB. Moduł mierzy zaledwie 8×3×2 cm, a dzięki jasnemu wyświetlaczowi graficznemu OLED, wskazania układu widoczne są nawet w słabych warunkach oświetleniowych – w ciemności, jak i w mocnym słońcu.

Potrzebne elementy

Do zbudowania opisywanej konstrukcji potrzebne są następujące elementy i moduły:

- Arduino Nano,
- moduł z TC4056A (ładowarka ogniwa li-poly),
- piezo buzzer,
- rezystor 10 kΩ,
- przełącznik dwupozycyjny, bistabilny,
- przycisk chwilowy (microswitch),
- moduł z czujnikiem ciśnienia BMP280,

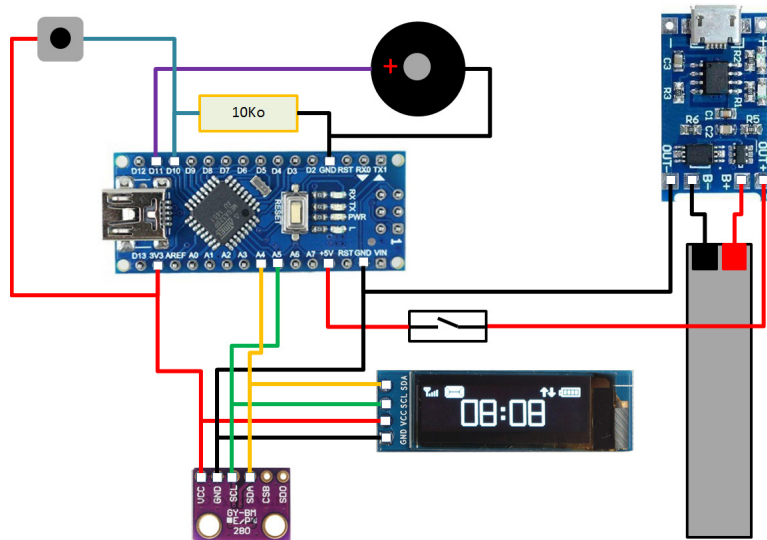


- wyświetlacz OLED 32×128 pikseli z zintegrowanym sterownikiem SSD1306,
- akumulator litowo-polimerowy 1S,
- opornik 4...10 kΩ SMD o rezystancji zależnej od prądu ładowania ogniwa litowo-polimerowego (patrz w tekście poniżej).

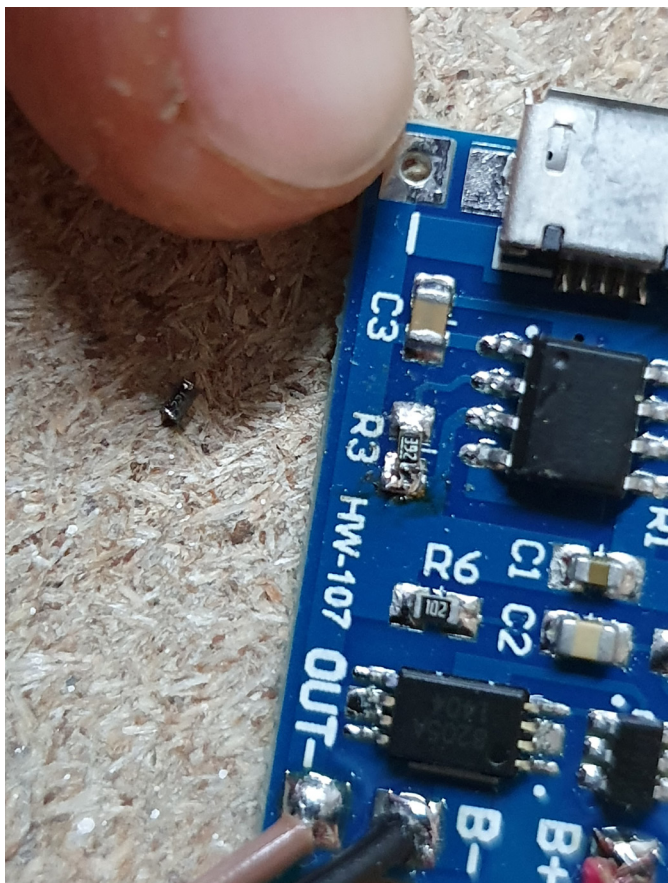
Na **rysunku 1** zaprezentowano uproszczony schemat połączenia wszystkich modułów układu. Zasilanie 5 V podawane jest z modułu ładowarki (do której podłączone jest ogniwo Li-poly) do modułu Arduino. Linia połączona jest poprzez przełącznik, który umożliwia włączanie i wyłączanie urządzenia. Drugi przycisk (microswitch) podłączony jest do linii cyfrowej D10 Arduino, która z pomocą opornika 10 kΩ ściągnięta jest do masy. Kolejna linia cyfrowa – D11 – steruje działaniem buzzera.

Moduł barometru oraz wyświetlacz podłączone są do układu poprzez interfejs I²C. Nie ma konieczności podciągania linii SDA i SCL tego interfejsu do zasilania (3,3 V), gdyż odpowiednie elementy podciągające znajdują się na płytce modułu Arduino.

Uwaga! Jak widać na schemacie, Arduino zasilone jest przez pin 5 V, a całe zasilanie 3,3 V pochodzi z tej płytki. Nie jest to zalecane rozwiązanie i może powodować niestabilność procesora lub problemy



Rysunek 1. Uproszczony schemat połączenia modułów w systemie



Fotografia 1. Położenie rezystora programującego prąd ładowania ogniwa litowo-polimerowego w module z TC4056A

z pracą stabilizatora 3,3 V w przypadku przeciążenia. Napięcie wychodzące z modułu ładowarki jest równe napięciu z ogniwa litowo-polimerowego i wynosi od 3,7 V do 4,2 V, zależnie od poziomu naładowania akumulatora. Przez większość tego zakresu daje wystarczający zapas napięcia, by liniowy stabilizator 3,3 V na pokładzie modułu działał poprawnie. Jeśli sprawia to jednak jakikolwiek problem, można umieścić pomiędzy akumulatorem, a płytką Arduino przetwornicę impulsową step-up, która podniesie napięcie do 5 V, niezależnie jakie napięcie dawać będzie ogniwo.

Konfiguracja ładowarki ogniwa litowo-polimerowych

Moduł TC4056A został zaprojektowany do ładowania akumulatora prądem o natężeniu 1 A. Wartość może być za duża dla małych ogniw litowo-polimerowych, dlatego może wynikać potrzeba zmniejszenia prądu ładowania. Zgodnie z kartą katalogową modułu TC4056A

R_{prog} (k Ω)	I_{bat} (mA)
30	50
20	70
10	130
5	250
4	300
3	400
2	580
1,66	690
1,5	780
1,33	900
1,2	1000

można to zrobić, zmieniając rezystor R3 na płytce drukowanej. Lokalizacja tego opornika pokazana jest na **fotografii 1**. Należy usunąć znajdujący się opornik 1,2 k Ω i zmienić go na przykład na 4 k Ω czy dowolną inną wartość, odpowiednią dla zastosowanego ogniwa litowo-polimerowego. Maksymalną wartość prądu ładowania należy odczytać z karty katalogowej ogniwa a wartość opornika zgodnie z **tabelą 1**, skopiowaną z karty katalogowej kontrolera ładowania TC4056A. Należy pamiętać, aby nie przekraczać maksymalnego dopuszczalnego prądu ładowania ogniwa, gdyż może to grozić jego uszkodzeniem lub nawet katastrofalną awarią z wybuchem czy pożarem włącznie.

Uwaga! Zgodnie z kartą katalogową modułu ładowarki, zasilanie dla Arduino musi być odłączone podczas ładowania ogniwa litowo-polimerowego!

Oprogramowanie

Do kompilacji i wgrania kodu do modułu Arduino potrzebne jest oprogramowanie Arduino IDE. W pierwszej kolejności trzeba dodać do IDE dodatkowe biblioteki, z jakich korzysta kod. Można je zainstalować z poziomu środowiska deweloperskiego z menu zarządzania bibliotekami. Wyszukujemy i instalujemy następujące biblioteki (dwie ostatnie biblioteki pobrać trzeba ze strony projektu na portalu Instructables – <http://bit.ly/2NPdf6k>):

- Adafruit_SSD1306,
- Biblioteka Adafruit GFX,
- Biblioteka Adafruit BMP280,
- SBB_Click,
- Bounce2.

Po zainstalowaniu wszystkich potrzebnych bibliotek można skompilować szkic i wgrać go do modułu Arduino. Fragmenty głównego szkicu programu – pokazano na **listingu 1**. Jeśli podczas kompilacji wystąpi błąd, oznacza to, że trzeba dopasować bibliotekę Adafruit SSD1306 do poprawnego adresu wyświetlacza.

Uwaga! Upewnij się przed podłączeniem modułu do komputera, że Arduino jest zasilane tylko przez USB podczas przesyłania kodu. Wyjmij baterię przed podłączeniem kabla USB do portu programowania.

Działanie systemu

Po włączeniu wariometru pojawia się ekran inicjalizacyjny, a następnie ekran pozostaje czarny. Przez większość czasu do działania potrzebny jest tylko dźwięk. Jeśli nie chcesz, aby tak się stało, należy zmienić wartość zmiennej `display_on` w szkicu na wartość `true` (linia 30) oraz `menu = 1` (wiersz 26).

Przy jednokrotnym naciśnięciu przycisku na ekranie urządzenia wyświetli się pierwszy z ekranów (1 strona). Krótkie naciśnięcie przycisku umożliwi przełączanie między czterema głównymi stronami głównego menu, na których wyświetla się odpowiednio:



Fotografia 2. Układ w obudowie wykonanej w technologii druku 3D

Listing 1. Fragmenty głównego szkicu programu

```

void setup() {
  Serial.begin(9600);
  // Pomiar napięcia
  vRef.begin();
  // Inicjalizacja wyświetlacza na I2C pod adresem 0x3C
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  // Inicjalizacja BMP280
  bmp.begin();
  // Wyjście buzzera
  pinMode(buzzer, OUTPUT);
  // Czyszczenie bufora ekranu
  display.clearDisplay();
  // Czekamy aż sensor będzie gotowy
  initScreen();
  // Przyciski skonfigurowane jako linie wejścia
  pinMode(switchpin1, INPUT);
  // Konfiguracja podciągania w górę na liniach przycisków
  digitalWrite(switchpin1, HIGH);
  // Podłączenie przycisków do obiektu filtrującego
  btbounce1.attach(switchpin1);
  btbounce1.interval(5);
}

void loop() {
  // Tutaj można wstawić poprawki korygujące
  float Altitude = bmp.readAltitude(1013.25 + pressurecorrecture);
  // Obliczane prędkości pionowej
  VarioMSCalculation();
  // Próg alarmu dźwiękowego (działa tylko gdy dźwięki są włączone)
  if (beep_on){
    VarioBeep(0.2, -5.0);
  }
  ButtonFunctions(); // Obsługa przycisków
  if (display_on){ // Obsługa ekranu
    if (menu == 1) {
      M1_displayVarioScreen(Altitude, vario);
    } else if (menu == 2){
      M2_DisplayClimbBar(vario);
    } else if (menu == 3){
      M3_DisplayTempPressure(bmp.readTemperature(), bmp.readPressure()/100.0);
    } else if (menu == 4){
      M4_DisplayBattery();
    } else if (menu == 5){
      M5_DisplayPressureSetup(Altitude);
    } else if (menu == 6){
      M6_DisplayBeepSetup();
    } else if (menu == 7){
      M7_DisplayDisplay_OnOff();
    } else if (menu == 8){
      Mx_DisplayExit();
    }
  } else {
    // Ponieważ wyświetlanie ekranu zajmuje jakiś czas
    // konieczne jest dodanie pewnego opóźnienia
    delay(39);
  }
}

void VarioMSCalculation(){
  // Zmienna przechowuje czas od włączenia systemu,
  // pobrany z millis()
  tempo = millis();
  // zmiennie do uśredniania
  N1 = 0;
  N2 = 0;
  N3 = 0;
  D1 = 0;
  D2 = 0;
  vario = 0;

  // Uśrednianie
  for(int cc=1; cc<=maxsamples; cc++) {
    alt[cc-1] = alt[cc];
    tim[cc-1] = tim[cc];
  }
  alt[maxsamples] = bmp.readAltitude();
  tim[maxsamples] = tempo;
  float stime = tim[maxsamples-samples];

  for(int cc=(maxsamples-samples); cc<maxsamples; cc++) {
    N1+=(tim[cc]-stime)*alt[cc];
    N2+=(tim[cc]-stime);
    N3+=(alt[cc]);
    D1+=(tim[cc]-stime)*(tim[cc]-stime);
    D2+=(tim[cc]-stime);
  }
  // Wyliczenie wysokości
  vario=1000*((samples*N1)-N2*N3)/(samples*D1-D2*D2);
}

```

1. tempo zmiany wysokości w postaci cyfrowej i graficznej (belka) oraz aktualna wysokość i poziom naładowania baterii,
2. duża belka tempa wznoszenia/opadania (do pionowego montażu układu),
3. temperatura i ciśnienie otoczenia,
4. procent naładowania baterii.

Z kolei długim naciśnięciem przechodzi się od menu ustawień. Krótkie naciśnięcie umożliwia iterację pomiędzy poszczególnymi ustawieniami. Długie naciśnięcie ponownie pozwala wprowadzić określone ustawienie i zmieniać jego nastawy krótkim naciśnięciem. Długie naciśnięcie ponownie zapisuje. W menu ustawień znajdziemy strony ustawień:

1. Wysokość,
2. Dźwięk ON/OFF,
3. Ekran ON/OFF,
4. Wyjście.

Obudowa

Autor projektu zaprojektował obudowę dla opisanej elektroniki. Została wykonana w technologii druku 3D. Autor nie udostępnił plików tego druku, gdyż są w nim pewne błędy, które sprawiają, że wykorzystanie jej do urządzenia wymaga sporo dodatkowej pracy, by ją dopasować. Z uwagi na fakt, że układ wykorzystuje płytkę uniwersalną i połączenia kablami modułów elektronicznych, wymiary każdego z wykonanych według tego projektu urządzeń mogą się nieznacznie różnić. Jednocześnie mogą one mieć różne wymiary, więc ciężko jest opracować gotową, kompletną obudowę do tego systemu. Na **fotografii 2** pokazano urządzenie w wydrukowanej obudowie.

Podsumowanie

Przedstawiony powyżej wariometr jest prostym w wykonaniu urządzeniem, które z pewnością przyda się każdemu parolotniarzowi, skoczkiwi spadochronowemu i innym. Jakkolwiek nie zastąpi profesjonalnych systemów, chociażby z uwagi na brak odpowiednich certyfikatów to z pewnością będzie doskonałym uzupełnieniem wykorzystywanych przyrządów.

Nikodem Czechowski, EP

Źródło: <http://bit.ly/2NPdf6k>

REKLAMA

www.facebook.com/ElektronikaPraktyczna