



Politechnika
Śląska

POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI
KIERUNEK ELEKTRONIK I TELEKOMUNIKACJA

Projekt inżynierski

Licznik rowerowy STM z lokalizatorem GPS

Autor: Adam Skóbel

Kierujący pracą: dr inż. Andrzej Kukiełka

Gliwice, styczeń 2022

Spis treści

Streszczenie	1
Rozdział 1 Wstęp.....	3
1.1. Wprowadzenie	3
1.2. Zakres i podział pracy	4
Rozdział 2 Analiza rynkowa	6
Rozdział 3 Wymagania i narzędzia	7
3.1. Wymagania projektowe	7
3.2. Schemat blokowy urządzenia	8
3.3. Schemat ideowy urządzenia.....	10
3.4. Narzędzia programistyczne.....	13
3.5. Narzędzia testowe	13
3.6. Narzędzia konstrukcyjne.....	14
Rozdział 4 Specyfikacja zewnętrzna	15
4.1. Instrukcja obsługi licznika rowerowego	15
4.2. Zasilanie	16
4.3. Instalacja oraz zalecenia użytkowe	16
4.4. Zabezpieczenia i zagrożenia	17
Rozdział 5 Specyfikacja wewnętrzna	18
5.1. Opis problemu.....	18
5.2. Ważniejsze elementy biblioteczne	20
5.3. Optymalizacja kodu źródłowego	26
Rozdział 6 Weryfikacja i walidacja.....	27
6.1. Prototyp urządzenia	27
6.2. Rozwiązanie docelowe	28
6.3. Kalibracja.....	30
6.4. Bilans energetyczny	30
6.5. Testy funkcjonalności	31
6.5.1. Warunki	31
6.5.2. Godzina.....	31
6.5.3. Data.....	31
6.5.4. Długość i szerokość geograficzna	32
6.5.5. Prędkość	32
6.6. Dokładność pomiarów	33
6.6.1. Warunki	33
6.6.2. Godzina.....	33
6.6.3. Data.....	33
6.6.4. Długość i szerokość geograficzna	33
6.6.5. Prędkość	33
Rozdział 7 Podsumowanie.....	34
Bibliografia.....	i
Spis skrótów i symboli	v
Lista dodatkowych plików uzupełniających tekst pracy	vi
Spis rysunków	vii
Spis tabel	viii

Streszczenie

Celem pracy jest zaprojektowanie i wykonanie licznika rowerowego, wykorzystującego w swoim działaniu moduł GPS. Głównym komponentem wchodzącym w skład urządzenia jest mikrokontroler STM32. Licznik rowerowy powinien zawierać pewne funkcjonalności, które wyświetlane będą na wyświetlaczu LCD. Podstawową funkcją eksponowaną na wyświetlaczu ma być aktualna prędkość. Dodatkowe informacje, które ma otrzymać użytkownik licznika rowerowego to m.in. aktualne dane pozycji geograficznej (szerokość oraz długość geograficzna), aktualna data oraz godzina pomiaru.

Urządzenie powinno posiadać mały rozmiar, aby umożliwić dogodny montaż na kierownicy rowerowej. W projekcie powinien być również zawarty zewnętrzny moduł zasilania.

Urządzenie projektowane jest z myślą o umieszczeniu go we własnoręcznie przygotowanej obudowie, chroniącej urządzenie przed uszkodzeniami mechanicznymi i nadającej projektowi walory estetyczne.

Słowa kluczowe: moduł GPS, mikrokontroler STM32, wyświetlacz LCD, moduł zasilania.

Rozdział 1

Wstęp

1.1. Wprowadzenie

W 1973 roku Amerykański Departament Obrony zapoczątkował projekt ukierunkowany na stworzenie satelitarnego systemu nawigacji [1][2][3]. System ten opierał się na technologii umożliwiającej namierzenie dowolnego punktu na ziemi z wykorzystaniem odpowiednio przystosowanego do tego odbiornika. Początkowo technologia ta znalazła użycie w zastosowaniach militarnych, jednak później udostępniona została także wszystkim innym użytkownikom na całym świecie, działając w sposób ciągły oraz nie nakładając na użytkowników żadnych dodatkowych opłat. Podstawą działania systemu nawigacji są satelity, które łączą się w konstelacje. Najbardziej znaną konstelacją jest GPS. Opracowana ona została przez Stany Zjednoczone i obejmuje swoim działaniem 31 satelitów krążących w przestrzeni średniej orbity okołoziemskiej, czyli przynajmniej na wysokości 20200km.

Istnieją również 3 inne konstelacje:

- GLONASS opracowana i obsługiwana przez Rosję,
- GALILEO opracowana i obsługiwana przez Unię Europejską,
- BEIDOU opracowana i obsługiwana przez Chiny.

Satelity te krążąc wokół planety nieustannie transmitują sygnał radiowy, który dociera do odbiorników umieszczonych na powierzchni Ziemi. Im więcej satelit nawiąże łączność z odbiornikiem GPS tym bardziej precyzyjne są odbierane dane.

System GPS jest dzisiaj podstawą działania wszelkich urządzeń elektronicznych i systemów umożliwiających określenie m.in. położenia oraz prędkości obiektu z dokładnością do metrów (odległość) czy też metrów na sekundę (prędkość). Technologia ta znajduje zastosowanie w wielu gałęziach gospodarki, pozwalając m.in. na określenie aktualnej pozycji geograficznej, prędkości, czy też wyznaczenie wysokości n.p.m.

W niniejszej pracy podjęto temat nawiązujący do technologii pozycjonowania dla samochodów osobowych, lecz w uproszczonej formie i ograniczonej skali. Praca dotyczy budowy licznika rowerowego, bazującego na module GPS, umożliwiającego pomiar aktualnej prędkości użytkownika, odczytanie aktualnej pozycji geograficznej oraz aktualnej daty i godziny pomiaru. Urządzenie ma być wyposażone w zasilane zewnętrzne. Możliwa jest także ewentualna rozbudowa urządzenia z uwagi na zaimplementowany port asynchroniczny UART, pozwalający na dołączenie dowolnego elementu elektronicznego, posługującego się tym właśnie portem komunikacyjnym.

Urządzenie ma na celu wspomóc użytkowników w wykonywaniu ćwiczeń rekreacyjnych, motywując poprzez dynamiczne przedstawianie rezultatów swojej aktywności fizycznej w prostej oraz przystępnej formie.

1.2. Zakres i podział pracy

Zakres prac związanych z wykonaniem urządzenia bazuje na dobraniu odpowiednich komponentów (tj. mikrokontrolera STM, modułu GPS, wyświetlacza LCD), zrealizowaniu modułu zasilania (którego podstawą będzie akumulator litowo-polimerowy) oraz zaimplementowaniu programu, realizującego wymagania projektowe, tj. wyświetlanie aktualnej prędkości, położenia geograficznego oraz aktualnej daty i godziny pomiaru. W zakres prac wchodzi również testy oraz ewentualna kalibracja urządzenia.

Praca została podzielona na 7 rozdziałów:

- pierwszy rozdział stanowi wstęp tematyczny, określenie wymagań oraz przedstawienie celu oraz zakresu prac,
- drugi rozdział skupia się na przedstawieniu problemu oraz skonfrontowaniu go z innymi rozwiązaniami rynkowymi,
- trzeci rozdział skupia się na przedstawieniu wymagań projektowych, wstępnych rozwiązaniach oraz zastosowanych komponentach, narzędziach programistycznych i testowych,

- czwarty oraz piąty rozdział pracy polegają na przedstawieniu: funkcjonalności, metodyki pracy, zasad i sposobu działania oraz obsługi urządzenia, kwestii związanych z bezpieczeństwem użytkowania urządzenia, oraz aspektów sprzętowych/programowych (tj. struktura programu wsadowego, przedstawienie ważniejszych fragmentów programu, czy też szczegóły implementacji poszczególnych komponentów),
- szósty rozdział poświęcony jest prezentacji prototypu oraz rozwiązania docelowego. Zawarty jest w nim również sposób testowania, ewentualna kalibracja, czy też analiza dokładności wykonanych pomiarów,
- ostatni rozdział to podsumowanie związane z przebiegiem prac oraz przedstawienie ewentualnych wniosków, wynikających z uzyskanych rezultatów.

Podstawową literaturą wykorzystaną do realizacji projektu była książka *Mikrokontrolery STM32 dla początkujących* [4] dotycząca podstaw programowania mikrokontrolerów STM32.

Praca inżynierska realizowana jest przez jedną osobę, będącą pod opieką oraz znajdującą się w ścisłej współpracy z osobą kierującą niniejszym projektem.

Rozdział 2

Analiza rynkowa

Z uwagi na to, że głównym zadaniem urządzenia będzie odczytywanie pomiarów pochodzących z modułu GPS, należy na samym początku ustalić pewne standardy, według których realizowany projekt będzie oscylował. Będą nimi: wygląd urządzenia, sposób przedstawiania parametrów wynikowych, kształt obudowy (w tym typowe wymiary urządzenia), rodzaj zasilania i na koniec dodatkowe funkcjonalności.

W tym celu porównano parę urządzeń komercyjnych [5][6] możliwych do kupienia na stronie internetowej firmy KROSS produkującej akcesoria rowerowe (w tym liczniki rowerowe). Urządzenia te nie różnią się znacznie między sobą pod względem zasilania, gdyż licznik rowerowy bez wbudowanego zasilania akumulatorowego nie zapewnia wygodnego użytkowania. Różnią się one jednak rodzajami zastosowanych czujników oraz modułów, umożliwiających pomiar takich parametrów jak: czas podróży, siła sygnału satelitarnego (w przypadku użycia modułu GPS), wysokość nad poziomem morza, puls użytkownika, czy też wariacje dotyczące pomiaru prędkości (tj. prędkość maksymalna, prędkość minimalna, czy też prędkość średnia). Urządzenia umieszczone są w podobnych do siebie (wodoszczelnych) obudowach z dopasowanym wycięciem na wyświetlacz. Długość pracy z użyciem wspomnianego zasilania akumulatorowego (podczas pełnej eksploatacji urządzeń) wynosi przynajmniej 12 godzin. Zużycie to jest zależne od ilości zastosowanych modułów, a przez to realizowanych funkcji urządzenia.

Na podstawie specyfikacji referencyjnych urządzeń komercyjnych określono wymagania urządzenia projektowego, które opisano w rozdziale trzecim.

Rozdział 3

Wymagania i narzędzia

3.1. Wymagania projektowe

Najważniejszym wymogiem, jaki musi spełniać urządzenie projektowe jest posiadanie zewnętrznego, wymiennego zasilania akumulatorowego. W tym celu wybrano akumulator litowo-polimerowy [7] o napięciu 3.7V oraz pojemności 750mAh oraz połączono go z modułem zasilania TP4056 [8]. Moduł przekazuje niezmienny sygnał zasilający do przetwornicy MT3608 [9], jednocześnie pozwalając na ładowanie akumulatora poprzez podłączenie zasilania za pomocą przewodu micro USB typu B. Poprzez przetwornicę zasilanie jest dostarczane do wszystkich pozostałych podzespołów urządzenia. Dodatkowo dobrany akumulator zapewnia stosunkowo długi czas pracy oraz prosty montaż (dzięki niewielkiemu rozmiarowi oraz wyprowadzonemu złączu zasilania).

Drugim wymogiem było dobranie odpowiedniego modułu lokalizatora GPS, zapewniającego dobrą funkcjonalność pod względem dokładności odbieranych danych lokalizacyjnych. W tym celu wybrany został moduł GY-NEO6MV2 [10] umożliwiający ustalanie aktualnej pozycji geograficznej z dokładnością wynoszącą 2.5m. Ponadto moduł umożliwia komunikację za pomocą portu UART, SPI czy też USB. W tym projekcie wybrano komunikację UART.

Urządzenie musi także zapewniać czytelny odczyt mierzonych parametrów. Zapewnia to wyświetlacz LCD 2x8 ART BIG ze sterownikiem HD44780 [11]. Wskazany wyświetlacz umożliwia odczyt w postaci dużych znaków zarówno w pełnym świetle słonecznym jak i nocą, dzięki wykorzystaniu wbudowanego podświetlenia.

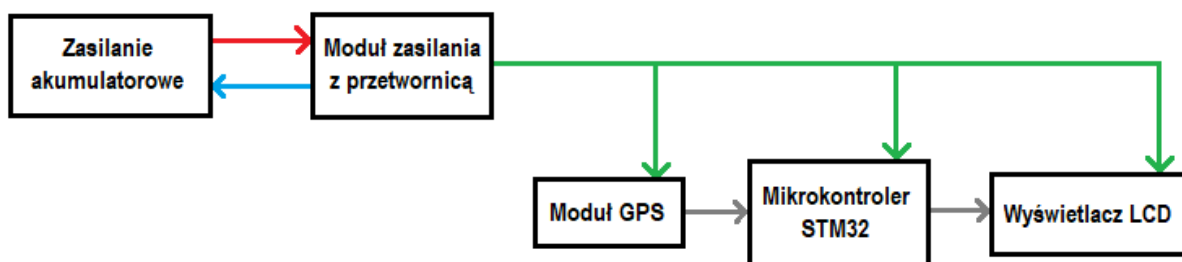
Prawidłowe działanie wszystkich komponentów ma zapewnić mikrokontroler STM32. Na potrzeby projektowe wybrany został układ NUCLEO-32 z mikroprocesorem STM32F303K8 [12]. Posiada on wbudowaną pamięć FLASH wynoszącą 64kB, odpowiednią ilość portów wyjściowych GPIO (w tym porty wyjścia UART umożliwiające komunikację z modułem GPS oraz dodatkowymi peryferiami). Dodatkowym atutem jest sam rozmiar układu, pozwalający na zaoszczędzenie miejsca na układzie drukowanym PCB.

Narzuconym (dodatkowym) wymogiem jest zastosowanie obudowy dopasowanej do układu licznika rowerowego, pozwalającej na stabilne umieszczenie urządzenia na kierownicy rowerowej. Mocowanie powinno zapewnić bezpieczeństwo użytkowania oraz łatwy sposób montażu i demontażu. W obudowie powinny się znaleźć dodatkowe otwory na ewentualne zastosowane porty, takie jak np. port ładowania akumulatora lub otwór na złącze zasilania. Obudowa powinna dodatkowo zapewniać stabilną pozycję akumulatora oraz anteny odbiorczej modułu GPS, która podatna jest na uszkodzenia wskutek nadmiernego uginania się przewodu.

Podczas dobierania elementów potrzebnych do skonstruowania urządzenia, kierowano się dostępnością oraz ceną poszczególnych komponentów.

3.2. Schemat blokowy urządzenia

Po określeniu podstawowych wymagań dotyczących budowy i działania licznika rowerowego należało w kolejnym kroku sporządzić schemat blokowy urządzenia, w celu wizualizacji wstępnych połączeń pomiędzy poszczególnymi zastosowanymi podzespołami. Chodzi tutaj o połączenia odpowiadające za zasilanie oraz za przesyłanie sygnałów nadawczo-odbiorczych. Schemat w minimalnym stopniu powinien także wizualizować rozmieszczenie elementów na przygotowanej podstawie (w postaci układu stykowego), a w późniejszej fazie projektu na układzie drukowanym. Schemat blokowy przedstawiono na rysunku nr 1.



Rysunek 1: Schemat blokowy licznika rowerowego

Pierwszy blok widoczny na rysunku nr 1, odpowiada za moduł akumulatora litowo-polimerowego. Jest on połączony z blokiem modułu zasilania z przetwornicą za pomocą linii zasilającej, co obrazuje czerwona strzałka. Niebieska strzałka odpowiada za sygnał zwrotny w postaci linii odpowiedzialnej za ładowanie akumulatora. Jest to możliwe dzięki zastosowanemu modułowi zasilania posiadającemu port micro USB typu B, pozwalającemu na podłączenie zewnętrznego źródła zasilania. Moduł ten przekazuje także niezmienny sygnał zasilający z akumulatora do przetwornicy, zwiększając jego wartość. Wzmocniony sygnał zasilający dociera do każdego pozostałego użytego w projekcie podzespołu, co obrazują wyprowadzenia w postaci zielonych strzałek. Podstawą pracy licznika rowerowego jest moduł GPS. Odbiera on dane z systemu satelitarnego i przekazuje je mikrokontrolerowi STM32, który przetwarza je i eksponuje na wyświetlaczu LCD w odpowiedniej formie. Kierunek przesyłanych danych przedstawiony został na rysunku za pomocą szarych strzałek.

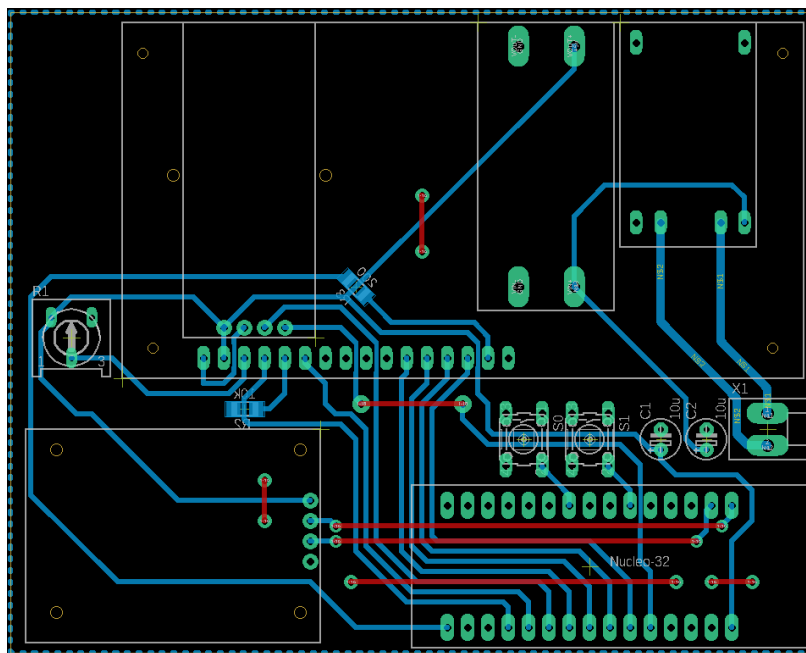
Po pełnej wizualizacji połączeń występujących pomiędzy użytymi podzespołami, jak i zastosowanych dodatkowych elementów w postaci półprzewodników, należało przystąpić do realizacji schematu ideowego.

Na rysunku nr 2 pokazany został schemat ideowy zaprojektowanego urządzenia.



W prawym górnym rogu rysunku nr 2, zauważyć można moduł zasilania z przetwornicą zasilającą pozostałą część układu. Można tu zaobserwować wystąpienie wyprowadzeń z portów B+ oraz B-. Są to wyprowadzenia dla złącza służącego do podłączenia zasilania akumulatorowego. W centralnej części schematu znajduje się układ rozwojowy NUCLEO-32. Do portów PB0 oraz PB1 układu podłączono dwa przyciski funkcjonalne opisane jako S0 oraz S1, umożliwiające przełączanie się pomiędzy danymi wyświetlanymi na ekranie wyświetlacza LCD. Wspomniany wyświetlacz umiejscowiony jest nad układem rozwojowym, podłączonym do niego w sekwencji 4-bitowej poprzez porty PA4, PA5, PA6 oraz PA7. Na lewo od wyświetlacza LCD znaleźć można moduł GPS. Połączony jest on z układem rozwojowym za pomocą portów PA9 oraz PA10, zapewniającymi wyprowadzenia układu NUCLEO dla portu UART, stanowiącemu połączenie transmisyjne dla wyprowadzeń RX oraz TX modułu GPS. W dolnej części schematu zauważyć można ostatni komponent urządzenia o nazwie HC-06 ZS-040 [13]. Jest to moduł Bluetooth zaimplementowany w celach ewentualnej przyszłej rozbudowy funkcjonalności urządzenia. Sposób sterowania urządzeniem przedstawiony został w rozdziale 4-tym.

W oparciu o schemat ideowy został utworzony także projekt układu drukowanego PCB. Układ modelowany był w oparciu o koncept układu jednostronnego, zawierającego wyprowadzenia dla złącza zasilania akumulatora. Projekt układu drukowanego PCB przedstawiony został na rysunku nr 3. Rzeczywiste wymiary układu to 10.3cm szerokości i 8.2cm długości.



Rysunek 3: Projekt układu drukowanego PCB

Zarówno schemat ideowy, jak i projekt układu drukowanego PCB, utworzony zostały w aplikacji EAGLE, posiadającą licencję umożliwiającą wykorzystanie edukacyjne. Wszelkie biblioteki elementów użytych w procesie projektowania zarówno schematu ideowego jak i układu drukowanego zostały wykorzystane w oparciu o biblioteki wbudowane programu EAGLE [14] (licencja edukacyjna), lub utworzone własnoręcznie zachowując oryginalny rozmiar i rozmieszczenie portów oraz otworów mocujących zastosowanych komponentów. Do określenia parametrów takich jak wymiary komponentu czy też rozmieszczenie otworów montażowych, tj. średnica otworu i pozycja względem krawędzi bocznych użyto not katalogowych wykorzystywanych podzespołów, a także dokonano pomiarów ręcznych w celu skorelowania wartości podanych przez producenta z rzeczywistymi.

Rezystor R3 widoczny między złączem PB3 mikrokontrolera STM32, a złączem LEDA wyświetlacza LCD dobrany został na podstawie obliczeń (prawo Ohma). Obliczenia przedstawione zostały za pomocą wzoru nr 1.

$$R3 = \frac{V_{CC} - V_d}{I_d} = \frac{3.3V - 2.5V}{10mA} = 80\Omega \quad (1)$$

V_{CC} – napięcie zasilania

V_d – napięcie diody

I_d – prąd diody

Wartości elementów tj. R1, R2, C1 i C2 dobrane zostały doświadczalnie. Urządzenie w początkowej fazie projektowania zostało zaimplementowane oraz uruchomione w konfiguracji, której podstawą był układ stykowy.

Wartość parametru R1, czyli potencjometru wynosi w tym przypadku 6.63kΩ, natomiast wartości kondensatorów C1 oraz C2 wynoszą 10μF.

3.4. Narzędzia programistyczne

Oprogramowanie dla urządzenia docelowego zostało stworzone dzięki środowisku STM32CubeIDE, udostępnionego przez firmę STMicroelectronics na licencji studenckiej (edukacyjnej) oraz w środowisku Atollic TrueSTUDIO, będącym prekursorem dla programu STM32CubeIDE, również należącym do firmy STMicroelectronics, lecz udostępnionym przez firmę do użytku bez przyznawania wyspecjalizowanej licencji. Aplikacje te są zalecane do programowania mikrokontrolerów STM32, zapewniając szereg funkcjonalności. Taką przykładową funkcjonalnością jest np. zaimplementowany w środowisku interfejs graficzny umożliwiający inicjalizację portów, przydzielenie im indywidualnej nazwy, włączenie specjalnych funkcji mikrokontrolera, czy też uruchomienie specjalnych cech poszczególnych portów takich jak obsługa interfejsu UART, SPI czy I2C. Środowisko STM32CubeIDE zawiera również uproszczenia, umożliwiające programiście sprawować lepszą kontrolę nad pisanymi fragmentami kodu poprzez możliwość dynamicznego tworzenia plików bibliotek dla elementów będących w trakcie konfiguracji.

3.5. Narzędzia testowe

Do sprawdzenia poprawności wysyłanych przez moduł GPS danych, użyta została przeglądarka internetowa w postaci Google Maps [15] oraz program U-center [16].

Poprawność programu testowano przy użyciu konwertera USB UART [17], umożliwiającego wysyłanie własnych komunikatów z poziomu komputera osobistego na port UART licznika rowerowego, wykorzystując do tego aplikację PuTTY [18]. Stanowiło to adekwatne rozwiązanie w stosunku do funkcjonalności modułu GPS, który pozostawiony na parapecie w trakcie testów, nie potrafił zalogować się do systemu satelitarnego, a tym samym niemożliwym było otrzymywanie aktualnych oraz poprawnych danych satelitarnych. Użycie wspomnianych dwóch aplikacji testowych zapewniło ciągłość prac, gdyż możliwe było natychmiastowe podawanie pożądaných komunikatów bez oczekiwania na zalogowanie się modułu GPS do systemu satelitarnego.

W przypadku poprawnego działania modułu GPS, znakomicie sprawdzała się aplikacja U-center, gdyż oprogramowanie to pozwala na sprawdzenie szeregu funkcji (takich jak wykaz ilości satelitów połączonych z modułem, uogólnione dane pomiarowe, wykaz wiadomości przychodzących, mapa ze wskazaniem aktualnej lokalizacji).

Aplikacja posiada również funkcję konfigurowania modułu GPS oraz możliwość ponownego jego uruchamiania w odpowiednim trybie:

- zimny start – uruchamianie modułu trwa ok. 27 sekund (jest to najwolniejszy tryb w jakim może uruchamiać się moduł GPS),
- gorący start – uruchamianie modułu trwa ok. 3 sekundy a dane odbierane są prawidłowo w przypadku zakończenia i rozpoczęcia pracy w tej samej pozycji geograficznej.

Jedyną właściwością, której nie da się przydzielić modułowi GPS jest wskazywanie godziny pomiaru w standardzie UTC+1. Aby to zrobić, należy wprowadzić poprawki w kodzie programu obsługującego działanie modułu.

3.6. Narzędzia konstrukcyjne

Do budowy projektu wykorzystano stację lutowniczą Zhaoxin 936D wraz z wyposażeniem pozwalającym na dogodną instalację złącz oraz portów na płycie drukowanej, a także bezpośrednie przylutowanie komponentów w postaci modułu zasilania, przetwornicy oraz modułu bluetooth do układu drukowanego.

Dodatkowe narzędzia, których użyto do montażu podzespołów oraz w procesie lutowniczym to kombinerki oraz obcęgi.

Narzędziem przydatnym do wykonania pomiarów wielkości poszczególnych modułów była suwmiarka. Pozwoliła ona na bardzo dokładne określenie rozmiarów układów oraz odległości od krawędzi otworów zapewniających komfortowe mocowanie. Możliwe było także zmierzenie średnic wspomnianych otworów montażowych.

Przyrządem, którego użyto do pomiaru poziomów sygnałów zasilających oraz do sprawdzania poprawności połączeń pomiędzy elementami był multimetr uniwersalny DPM DT9205A, którego dokładności dla konkretnych typów pomiarów przedstawione są w instrukcji dołączonej do urządzenia [19].

Ostatnim narzędziem użytym w celach projektowych była drukarka 3D. Pozwoliła ona na zaprojektowanie oraz wydrukowanie obudowy dla licznika rowerowego STM.

Nagrania oraz zdjęcia wykonane zostały za pomocą telefonu osobistego.

Rozdział 4

Specyfikacja zewnętrzna

4.1. Instrukcja obsługi licznika rowerowego

Do obsługi urządzenia służą dwa przyciski funkcyjne. Wybór prawego przycisku (S1) spowoduje przełączenie się wyświetlanego obrazu między pięcioma dostępnymi stronami (menu) o jeden stopień w górę, natomiast wybór lewego (S0) spowoduje cofnięcie się w obszarze menu o jeden stopień w dół. W przypadku przytrzymania któregośkolwiek z przycisków nastąpi nieustanna, dynamiczna inkrementacja lub dekrementacja (w zależności od przytrzymywanego przycisku) strony. Menu składa się z pięciu stron: aktualna godzina (strona 1), data (strona 2), długość geograficzna (strona 3), szerokość geograficzna (strona 4) oraz prędkość (strona 5). Dane wyświetlane są na wyświetlaczu LCD ze znakami w formule ART BIG, co oznacza, że wyświetlane znaki będą dobrze widoczne dzięki ich dużemu rozmiarowi oraz zielonemu tłu, które stanowi kontrast dla znaków czarnego koloru. Po podłączeniu zasilania akumulatorowego na ekranie wyświetlać się będą informacje w dwóch wierszach - w pierwszym wierszu ukazywane są danych pomiarowe, a w drugim wyświetlany jest numer strony. Dla strony nr 3 oraz 4, tj. szerokość oraz długość geograficzna, nie wyświetlany jest numer strony, gdyż dane pomiarowe zajmują całą przestrzeń wyświetlacza. W przypadku ciągłego braku łączności modułu z systemem satelitarnym, a co za tym idzie brakiem uzyskiwania poprawnych danych, na ekranie wyświetli się komunikat „LOADING!”. Należy wtedy poczekać aż moduł GPS zacznie odbierać informacje prawidłowo, co zasygnalizowane będzie miganiem niebieskiej diody LED, umieszczonej na module. Szacowany czas do rozpoczęcia poprawnego działania modułu GPS od momentu włączenia zasilania to okres ok. 27 sekund w przypadku dobrego zasięgu.

4.2. Zasilanie

Możliwe jest zaoszczędzenie energii elektrycznej pobieranej z akumulatora poprzez odłączenie go od układu, wyciągając złącze umieszczone na prawo od przycisków funkcyjnych, na obrzeżach układu drukowanego. Ponowne podłączenie akumulatora, tym samym automatyczne uruchomienie urządzenia, następuje poprzez umiejscowienie złącza w porcie opisanym jako „+ -”, podłączając czerwony przewód do złącza „+” a czarny przewód do złącza „-”. W przypadku wystąpienia braku zasilania może to oznaczać rozładowanie się akumulatora.

4.3. Instalacja oraz zalecenia użytkowe

Urządzenie docelowo instaluje się na ramie rowerowej za pomocą umieszczonych na odwrocie obudowy zaczepów. Są to zaczepy, które swoją zasadą działania przypominają opaskę zaciskową, lecz umożliwiają także poluzowanie uścisku oraz całkowite zdjęcie obudowy. Nie zaleca się użytkowania urządzenia w budynkach lub pod zadaszeniami, gdyż funkcjonalność urządzenia znacznie wtedy spada. Objawia się to niemożnością zalogowania się nadajnika do systemu satelitarnego, a zatem pobrania danych satelitarnych. Spowodowane jest to zastosowanym modulem GPS, który ma trudności z łącznością satelitarną, gdy znajduje się pod zadaszeniem lub w głębi budynku. Gdy urządzenie działa poprawnie na ekranie wyświetlają się dane pomiarowe lub w przypadku ładowania danych satelitarnych napis „LOADING!”. Dodatkową informacją o działaniu urządzenia jest dioda LED umieszczona na module mikrokontrolera, która zmienia swój stan logiczny na przeciwny za każdym razem, gdy otrzymuje komunikat od modułu GPS. Sam moduł wysyła również informacje o poprawności swojego działania poprzez nieustanne zaświecanie oraz gaszenie niebieskiej diody LED umieszczonej na układzie drukowanym modułu. Świecenie diody oznacza, że moduł zalogował się do systemu satelitarnego i wysyła poprawne komunikaty na porty RX oraz TX mikrokontrolera STM. W przypadku wymiany akumulatora należy pamiętać o usunięciu zaślepki umieszczonej pomiędzy spodem akumulatora a obudową urządzenia. Dla ułatwienia montażu akumulatora należy w pierwszej kolejności umieścić przewód zasilania w specjalnie przygotowanym otworze, a dopiero później umieścić resztę komponentu.

4.4. Zabezpieczenia i zagrożenia

Urządzenie zabezpieczone jest częściowo przed zewnętrznymi uszkodzeniami mechanicznymi obudową wykonaną z wykorzystaniem drukarki 3D, a niektóre komponenty urządzenia umieszczone na układzie drukowanym wsparte są dystansami, które wzmacniają konstrukcję urządzenia oraz zabezpieczają podzespoły.

Podczas korzystania z licznika rowerowego w dniach wzmożonej wilgotności powietrza atmosferycznego należy uważać, gdyż obudowa nie chroni całkowicie urządzenia przed wilgocią oraz nie izoluje całkowicie części sprzętowej od czynników zewnętrznych. Testy środowiskowe licznika rowerowego odbywały się w sezonie zimowym, zatem nie udało się doświadczyć wpływu wysokiej temperatury na pracę urządzenia. Obudowa nie zapewnia całkowitej szczelności.

W przypadku demontażu obudowy, należy zwrócić szczególną uwagę podczas wyciągania anteny ze specjalnego podestu. Należy lekko podeprzeć antenę od strony podstawy sztywnym oraz w miarę szerokim narzędziem. Przed przystąpieniem do podważania elementu zaleca się w pierwszej kolejności odłączenie przewodu anteny od modułu GPS, gdyż przy nagłym usunięciu odbiornika przewód może ulec uszkodzeniu.

W przypadku niezastosowania dystansów dołączonych do obudowy licznika rowerowego drgania pojawiające się podczas użytkowania urządzenia mogą zniszczyć komponenty niezabezpieczone wspornikami.

Rozdział 5

Specyfikacja wewnętrzna

5.1. Opis problemu

Głównym problemem do rozważenia podczas pisania kodu programu obsługującego licznik rowerowy była interpretacja protokołów NMEA wysyłanych do modułu GPS przez system satelitarny Ziemi oraz przekazywanych do mikrokontrolera za pośrednictwem portu UART, w postaci informacji zapisanych w kodzie ASCII. Wykaz protokołów możliwy jest do znalezienia pod adresem umieszczonym w bibliografii [20].

Protokoły te zawierają informacje dotyczące konkretnego typu pomiaru. Przykładowo komunikat \$GPGLL zawiera dane dotyczące aktualnej lokalizacji geograficznej, a komunikat \$GPGSV zawiera informacje dotyczące aktualnej liczby satelitów dostępnych w zasięgu urządzenia. Każdy komunikat zawiera ponadto własną, unikatową oraz niezmienną liczbę przecinków. Pomiędzy konkretnymi przecinkami znajdują się informacje dotyczące specyficznych parametrów. Nawiązując do komunikatu \$GPGSV, pomiędzy 8 a 9 przecinkiem znajduje się informacja dotycząca aktualnej liczby satelitów w zasięgu urządzenia. Za pomocą tego komunikatu można w ten sposób określić siłę sygnału przychodzącego, gdyż im większa liczba satelitów w zasięgu urządzenia, tym dokładniejszy i szybszy jest pomiar wartości mierzonych.

Po analizie listy protokołów wysyłanych przez system satelitarny, wyznaczono dwa protokoły, z których pobierane będą dane dotyczące lokalizacji geograficznej, daty, godziny oraz prędkości. Za odbiór informacji o aktualnej godzinie oraz aktualnej dacie pomiaru, długości, oraz szerokości geograficznej odpowiada komunikat \$GPRMC, natomiast za informacje dotyczące pomiaru prędkości lądowej odpowiada komunikat \$GPVTG. Wspomnieć trzeba, że dane dotyczące godziny odbierane są w formacie UTC,

co powoduje, że godzina wyświetlana na wyświetlaczu LCD będzie nieaktualna względem czasu dla szerokości geograficznej Polski (dla Polski: UTC+1). Problem ten rozwiązano poprzez zastosowanie odpowiedniej funkcji w kodzie programu, obsługującego pracę modułu GPS.

Dane pomiarowe dotyczące położenia geograficznego podawane są w zapisie, który umożliwia natychmiastowe wyświetlenie na wyświetlaczu LCD tylko stopni oraz minut kątowych. Podczas jazdy rowerem główna zmiana parametrów lokalizacji dotyczy sekund kątowych i taką zmianę w większości będzie obserwował użytkownik urządzenia. Pokazywanie zmiany sekund kątowych powoduje zarazem zwiększenie dokładności pomiaru, przez co użytkownik przepisując współrzędne do wyszukiwarki Google, będzie mógł odczytać wskazaną pozycję z dokładnością do paru metrów.

Aby uzyskać pożądaną dokładność należy w pierwszej kolejności zrozumieć jakie informacje wchodzi w skład wybranych protokołów oraz w jaki sposób należy interpretować składnię przychodzących danych. Poniżej przedstawiono zawartość wybranych na potrzeby projektu komunikatów wraz z ich przykładem i opisem, co do informacji znajdujących się po kolejnych przecinkach komunikatu.

\$GPRMC

Przykład: *\$GPRMC,124717.00,A,3506.52397,N,12214.15914,E,0.028,,011221,,,A*72*

Po pierwszym przecinku znajduje się informacja dotycząca aktualnej godziny występującej w formacie UTC. Symbol „A” występujący po przecinku to flaga poprawności. W przypadku wystąpienia symbolu „V” komunikat nie zawierałby żadnych danych, co jednocześnie oznaczałoby brak zalogowania do systemu satelitarnego przez moduł GPS. Po symbolu flagi występuje długość geograficzna. Pierwsze 4 znaki to stopnie (2 znaki) oraz minuty (2 znaki). W przypadku długości kątowej dla stopni przysługują 2 znaki, gdyż maksymalny kąt odchylenia wynosi 90 stopni. Wartość po kropce to zakodowane sekundy kątowe. Aby je zdekodować należy wartość po przecinku (w tym przypadku 0.52397) przemnożyć przez liczbę 60. W ten sposób powstaje wartość 31.43 co oznacza się odpowiednim symbolem ("). Symbol następujący po długości geograficznej to oznaczenie długości północnej „N” lub oznaczenie długości południowej „S”. Dla następującej później szerokości geograficznej sytuacja wygląda analogicznie z wyłączeniem liczby znaków dla stopni, gdyż dla szerokości geograficznej maksymalne odchylenie wynosi już 180 stopni. Przedostatnią informacją jest prędkość lądowa w węzłach (0.028). Ostatnią informacją jest data wykonania pomiaru. Komunikat kończy się trzema przecinkami oraz sumą kontrolną.

\$GPVTG

Przykład: *\$GPVTG,,T,,M,0.009,N,0.016,K,A*2D*

Pierwsza informacja komunikatu dotyczy nawiązania punktu odniesienia do bieguna północnego Ziemi. Oznacza się to występowaniem symbol „T” po drugim przecinku komunikatu. Kolejną informacją jest symbol „M”, który oznacza poprawność odczytu pola magnetycznego. Wartość *0.009* to prędkość lądowa występująca w węzłach. Symbol „N” to jej oznaczenie. Ostatnią informacją jest prędkość lądowa występująca w wartościach kilometrów na godzinę. Oznacza się ją symbolem „K”. Komunikat kończy suma kontrolna.

Wiele protokołów odbieranych przez moduł GPS w standardzie NMEA zawiera te same dane. Dla przykładu użyty protokół \$GPRMC zawiera w sobie informacje dotyczące lokalizacji geograficznej. Te same informacje zawiera również protokół \$GPGLL, \$GPGAA, czy też \$GPTRF. Protokół \$GPRM został wybrany z uwagi na mnogość informacji w nim zawartych, tj. długość i szerokość geograficzna oraz data i godzina pomiaru. Zaoszczędzono w ten sposób pamięć mikrokontrolera STM32, przeznaczoną na zapis treści pochodzących z wybranych protokołów.

5.2. Ważniejsze elementy biblioteczne

Bibliotekę obsługi modułu GPS oparto o strukturę pokazaną na rysunku nr 4.

```
typedef struct{
    unsigned char string_data[70][buf_size];
    // rezerwacja miejsca w pamięci na bufor danych

    unsigned char receive;
    unsigned char inkrementator[buf_size];
    unsigned char string_flag[buf_size];
    unsigned char a[buf_size];
}gps;

gps GPS;
```

Rysunek 4: Struktura podstawowa programu

Jest to struktura, wykorzystywana w późniejszej funkcji *read_gps_data()* do odczytu oraz zapisu znaków, wybranych przez programistę protokołów standardu NMEA. Zawiera ona zmienne odpowiedzialne m.in. za przechowywanie ciągu znaków przechwytywanego komunikatu w tablicy dwuwymiarowej, czy też flagę zapisu.

Dodatkowo struktura przypisana została do nazwy GPS, więc przykładowe odniesienie się do jakiegokolwiek zmiennej struktury będzie następowało poprzez wywołanie tejże nazwy. Przykładowo dla chęci wywołania zmiennej *receive* będzie konieczne użycie zapisu *GPS.receive*.

Zawartość funkcji *read_gps_data()* bazującej na wspomnianej strukturze pokazano na rysunku nr 5.

```
void read_gps_data(unsigned char *string_data, unsigned char adr)
{
    if(GPS.string_flag[adr] == 1)    // rozpoczęcie zapisu z chwili wykrycia flagi
    {
        if(*(string_data + (GPS.inkrementator[adr])) != GPS.receive)
            && ((GPS.inkrementator[adr] < 5))
            // wykrywanie błędu w postaci niepoprawnego komunikatu
        {
            GPS.string_flag[adr] = 0;
            GPS.inkrementator[adr] = 0;
        }
        if((GPS.inkrementator[adr] >= 5 && GPS.receive != '\n')
            // sprawdzenie końca transmisji jednego ciągu znaków
        {
            GPS.string_data[GPS.a[adr]][adr] = GPS.receive;
            GPS.a[adr]++;
        }
        if((GPS.string_flag[adr] != 0)
            // przesuwanie się wzdłuż długości komunikatu
        {
            GPS.inkrementator[adr]++;
        }
        if(GPS.receive == '\n')
            // wykrywanie końca komunikatu i zerowanie zmiennych pomocniczych
        {
            GPS.inkrementator[adr] = 0;
            GPS.string_flag[adr] = 0;
            GPS.a[adr] = 0;
        }
    }
    if(GPS.receive == '$')GPS.string_flag[adr] = 1;
    // flaga wykrycia znaku $ komunikatu NMEA
}
```

Rysunek 5: Odczyt komunikatów z modułu GPS

Jak widać, programista (*read_gps_data()*) może wybrać nazwę komunikatu oraz wiersz tabeli, do którego chce zapisać dane. Możliwe jest również definiowanie liczby wierszy tabeli, przez co możliwe jest przechwytywanie większej liczby komunikatów. Dokonuje się tego poprzez zmodyfikowanie zmiennej globalnej o nazwie *buf_size*. Fragment kodu pokazanego na rysunku nr 1 powstał na bazie wskaźnika, odnoszącego się do adresu wybranego protokołu i porównującego jego zawartość ze znakiem przejętym przez stałą statyczną *receive* należącą do struktury GPS. Zmienna *receive* otrzymuje znak dzięki

przerwaniu zainicjowanego poprzez funkcję UART. Oznacza to, że funkcja porównuje nazwę wybranego przez programistę protokołu znak po znaku z protokołami przychodzącymi z sieci satelitarnej. Jeśli następuje zgodność, to zawartość przychodzącego komunikatu zapisywana jest do wybranego wiersza tablicy.

Wywołanie funkcji `read_gps_data()` zostało przedstawione na rysunku nr 6.

```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if(huart->Instance == USART1)
    {
        read_gps_data("GPRMC",0); // zapis danych do wiersza 0 tablicy
        read_gps_data("GPVTG",1); // zapis danych do wiersza 1 tablicy

        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_3);

        init_gps();
    }
}
```

Rysunek 6: Wywołanie funkcji oraz wybór przechwytywanych protokołów

Do odczytu danych wysyłanych przez moduł GPS wykorzystano funkcję UART mikroprocesora. Przy odbiorze danych towarzyszy także świecenie diody LED umieszczonej na układzie rozwojowym NUCLEO. Miganie diody oznacza, że do mikrokontrolera za pomocą przerwania przychodzą pojedyncze komunikaty NMEA, wysyłane przez moduł GPS.

Drugą najważniejszą funkcją znajdującą się w bibliotece `gps.c` jest `number_comma()`. Używana jest ona do sprawdzania liczby znaków znajdujących się za konkretnym przecinkiem. Funkcja zlicza liczbę znaków do następnego przecinka występującego od momentu rozpoczęcia zliczania. Ona również zbudowana jest o koncept działania na wskaźniku. W tym przypadku programista podaje numer przecinka oraz adres wiersza tabeli dwuwymiarowej. Funkcja opisana jest szczegółowo w przypisach znajdujących się w kodzie programu źródłowego.

Na podstawie funkcji `number_comma()`, zaproponowana została funkcja `MENU()`, w której wykonano uwarunkowane wyświetlanie danych pochodzących z wybranych przychodzących komunikatów. Tutaj został zamieszczony również algorytm dekodujący dane komunikatu dla lokalizacji geograficznej w celu uzyskania sekund kątowych oraz korekta formatu wyświetlanej godziny. Algorytm obliczający sekundy kątowe przedstawiony został na rysunku nr 7, natomiast algorytm nanoszący poprawki godziny na rysunku nr 8.

```
// zapisanie do zmiennej tymczasowej liczby po przecinku
uint32_t temporary = (GPS.string_data[znak+6][0] & (0x0F))*1000
                    + (GPS.string_data[znak+7][0] & (0x0F))*100
                    + (GPS.string_data[znak+8][0] & (0x0F))*10
                    + (GPS.string_data[znak+9][0] & (0x0F));

// wartość pożądana to 60-krotność wartości liczby po przecinku
// wartość jest mnożona ze względu na potrzeby wyliczenia danych
// dotyczących sekund dla położenia geograficznego
temporary = (6 * temporary)/10;

write_number(temporary/1000);
write_number(temporary/100);
write_sign('.');
write_number(temporary/10);
write_number(temporary%10);
write_sign(0x22);
write_sign(' ');
write_sign(GPS.string_data[znak+12][0]);
// znak kierunkowy dla długości geograficznej (N/S)
```

Rysunek 7: Algorytm na obliczanie sekund kątowych

```
// złączenie 2 znaków godziny w liczbę dwocyfrową
uint8_t temporary = (GPS.string_data[znak+1][0] & (0x0F))* 10
                    + (GPS.string_data[znak+2][0] & (0x0F));
temporary += 1; // poprawka dla UTC+1
if(temporary >= 24) // warunek dla północy
{
    temporary = 0;
}
```

Rysunek 8: Algorytm dla poprawy formatu wyświetlanej godziny

W projekcie została wykorzystana biblioteka dla wyświetlacza LCD w formacie 2x8, napisana na podstawie już istniejącej biblioteki dla wyświetlacza w formacie 2x16, tj. HD_44780 [21] na licencji umożliwiającej wykorzystanie w celach naukowych oraz noty katalogowej użytego wyświetlacza [11]. Wszelkie opisy dotyczące interpretacji ważniejszych elementów biblioteki można znaleźć w przypisach umieszczonych w plikach kodu źródłowego.

Na potrzeby biblioteki wyświetlacza LCD wykonano bibliotekę opóźnień (w formacie mikrosekund) o nazwie waitus. Biblioteka opóźnień została napisana w oparciu o funkcję NOP, użytą w kodzie nadzorującym pracę wyświetlacza. Funkcja *waitus()* została przedstawiona na rysunku nr 9. Służy ona do wywoływania opóźnień w mikrosekundach.

```

void waitus(uint32_t us) // funkcja generowania opoznienia w 'us'
{
    uint32_t i;
    for(i = 0; i < us; i++)
    {
        // x32 NOP [NOP - no operation]
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP");

        // x32 NOP [NOP - no operation]
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP"); asm("NOP");
        asm("NOP"); asm("NOP");
    }
}

```

Rysunek 9: Funkcja waitus dla biblioteki opóźnień

Obliczono, że wykonanie operacji NOP zajmuje procesorowi czas 15.625 nanosekund. Na zdjęciu nr 9 pokazano, że operacja NOP wykonana została 64 razy. Powoduje to, że zadeklarowanie argumentu funkcji waitus jako 1 spowoduje opóźnienie wynoszące 1 mikrosekundę. Obliczenia wykonane podczas realizacji funkcji zawarte zostały w komentarzach plików źródłowych.

Należy wspomnieć, że operacja opisana jako NOP jest możliwa do użycia tylko dla mikrokontrolerów z procesorem o architekturze ARM, występującym w wersji nowszej niż szósta [22].

Na koniec pozostawiono zaprogramowanie przycisków umieszczonych na układzie drukowanym PCB, służącym do zmiany stron zaprojektowanego menu.

Na rysunku nr 10 przedstawiono fragment kodu, który przedstawia obsługę przycisku opisanego na schemacie ideowym jako S0, służącego do zmienienia strony menu o 1 stopień w dół.

```
if(HAL_GPIO_ReadPin(BUTTON_S0_GPIO_Port, BUTTON_S0_Pin) != 1)
    // jeśli wcisnięty S0
{
    time0++;
    if(time0 > 70)        // programowa niwelacja drgań
    {
        time0 = 0;
        menu--;          // zmniejszenie poziomu menu
        if(menu < 0)
        {
            menu = 4;    // zabezpieczenie przed wyjściem z obszaru menu
        }
    }
}
else time0 = 0;
```

Rysunek 10: Deklaracja przycisku S0

Obsługę przycisków oparto o zegar zaimplementowany jako jedna z funkcji mikrokontrolera STM32. Zastosowany tutaj został także fragment kodu umożliwiający programową niwelację drgań styków podczas wciskania przycisków.

Na rysunku nr 11 przedstawiona została również obsługa przycisku S1, służącego do zmiany strony menu o 1 stopień w górę.

```
if(HAL_GPIO_ReadPin(BUTTON_S1_GPIO_Port, BUTTON_S1_Pin) != 1)
    // jeśli wcisnięty S1
{
    time1++;
    if(time1 > 70)        // programowy eliminator drgań
    {
        time1 = 0;
        menu++;          // zwiększenie poziomu menu
        if(menu > 4)
        {
            menu = 0;    // zabezpieczenie przed wyjściem z obszaru menu
        }
    }
}
else time1 = 0;
```

Rysunek 11: Deklaracja przycisku S1

Wszelkie opisy ważniejszych fragmentów kodu, zawarte są w komentarzach umieszczonych w poszczególnych plikach bibliotek.

5.3. Optymalizacja kodu źródłowego

Kod programu realizowany był z początku w inspiracji o kod zewnętrzny [23], udostępniony w Internecie na prawach pozwalających ulepszenie, czy też wykorzystywanie go w celach naukowych. Kod ten zamieszczony jest w bibliografii niniejszej pracy. Oparty był on jednak o bibliotekę „string”, udostępnioną za darmo przez firmę STMicroelectronics - co uniemożliwiało dalsze prace, gdyż pamięć FLASH mikroprocesora przepełniała się w przypadku załadowania wspomnianej biblioteki. W tym celu przystąpiono do tworzenia własnego kodu, inspirowanego jedynie własnymi przemyśleniami oraz informacjami dotyczącymi działania używanego modułu GPS. W późniejszych pracach wymieniono mikrokontroler na inny (STM32F303K8) wyposażony w większą pojemność pamięci FLASH (umożliwiającą wykorzystanie pierwszego konceptu rozwiązania), jednak prace projektowe były w wystarczająco zaawansowanym stadium, aby zrezygnować z przebudowywania projektu.

W zamieszczonej bibliotece znajduje się nieaktywna funkcja *check_position()*. Wykorzystywana była ona przy pierwszych próbach wykrywania danych dla konkretnych pozycji w komunikacie. Każdy komunikat zawiera bowiem zawsze tyle samo przecinków. Jedynymi danymi ulegającymi dynamicznej zmianie w treści komunikatu są informacje zawarte pomiędzy konkretnymi przecinkami np. informacje dotyczące aktualnej godziny zamieszczone są pomiędzy pierwszym a drugim przecinkiem komunikatu \$GPRMC. Stan tych informacji zależy jedynie od poprawności działania modułu GPS, tj. zalogowanie się do systemu satelitarnego i odbieranie poprawnych, pełnych danych. Z użycia napomnianej funkcji ostatecznie zrezygnowano i skupiono się na stworzeniu nowej, opartej na podobnym konceptie funkcji (*number_comma()*), mającej za zadanie wykryć konkretny przecinek komunikatu i określenie liczby znaków po nim następujących, do czasu wystąpienia kolejnego znaku przecinka.

Zrezygnowano również z rozwiązania dotyczącego wyświetlania danych z komunikatów na podstawie stałych pozycji w ciągu znaków. Okazało się to wysoce nieefektywne, gdyż komunikaty nie posiadają w większości tej samej długości informacji zawartych pomiędzy przecinkami. Przykładowo, gdy prędkość poruszania się przekroczy wartość 9.99 km/h, informacja o wartości prędkości zmieni długość z 5 do 6 znaków pomiędzy 7 a 8 przecinkiem komunikatu \$GPVTG.

Rozdział 6

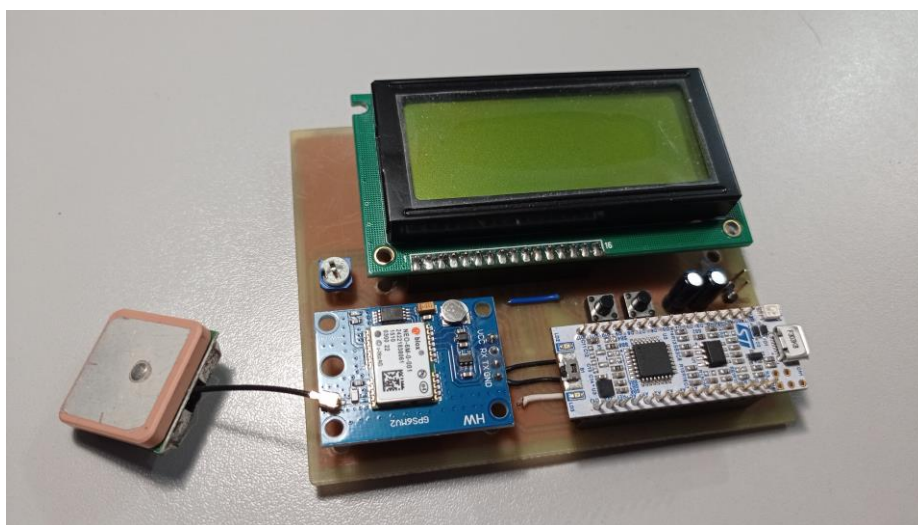
Weryfikacja i walidacja

6.1. Prototyp urządzenia

W pierwszych fazach projektowania urządzenia, za podstawę konstrukcyjną obrano układ stykowy o wymiarach zapewniających komfortowe umieszczenie komponentów oraz zorganizowanie połączeń. W pierwszych fazach projektowania oprogramowania licznika rowerowego badano poprawność działania użytego wyświetlacza LCD, czy też modułu lokalizatora GPS. Rozwiązanie sprawdzało się dla testowania biblioteki wyświetlacza, lecz w przypadku sprawdzania poprawności działania modułu GPS, nie zapewniała już poprawnej funkcjonalności układu. Przewody łączące lokalizator z układem rozwojowym NUCLEO nie zapewniały stabilnego połączenia, przez co moduł nieustannie inicjował swoje połączenie wydłużając czas walidacji układu. W celu dalszego testowania urządzenia (m.in. w warunkach środowiskowych, będąc w ruchu), zdecydowano się na przeniesienie układu na układ drukowany PCB (w celu zapewnienia stabilności połączeń oraz spójności układu). Prototyp urządzenia sprawdzał się jednak dla testowania poprawności działania układu z modułem GPS zastąpionym konwerterem UART USB. Połączenie nawiązywane przez konwerter nie ulegało nieustannym przerwaniom, a komunikaty wysyłane ręcznie pozwalały na kontrolowaną walidację zaprojektowanego układu.

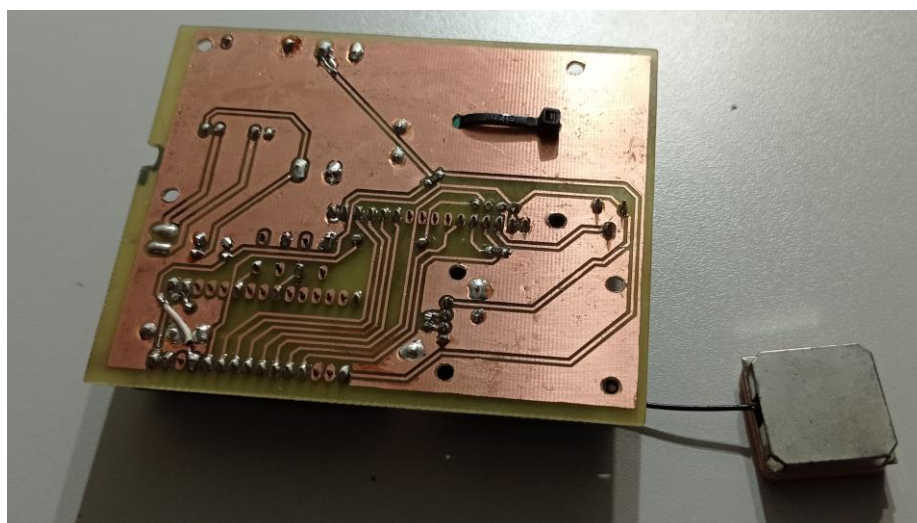
6.2. Rozwiązanie docelowe

Rozwiązanie docelowe obejmuje montaż komponentów w układzie drukowanym PCB oraz umieszczenie urządzenia w samodzielnie wykonanej obudowie. Na rysunku nr 12 pokazano układ docelowy nieumieszczony jeszcze w obudowie i nie zostało podłączone do zasilania akumulatorowego.



Rysunek 12: Układ docelowy

Rysunek nr 13 przedstawia układ drukowany, na którym zawarte są wszystkie ścieżki oraz pozostałe łączenia komponentów licznika rowerowego. Widoczna jest tutaj także opaska zaciskowa służąca do montażu modułu bluetooth.



Rysunek 13: Układ drukowany

Rysunek nr 14 przedstawia działające urządzenie umieszczone w obudowie, przytwierdzone do ramy rowerowej, wyświetlające godzinę w formacie UTC+1.



Rysunek 14: Obudowane urządzenie umieszczone na kierownicy rowerowej

Na rysunku nr 15 zaprezentowano również urządzenie znajdujące się w obudowie przytwierdzonej do kierownicy rowerowej bez zainstalowanego panelu przedniego obudowy, pokazujące zawartość strony nr 5, tj. aktualną prędkość.



Rysunek 15: Urządzenie bez panelu przedniego obudowy

W dodatkowych plikach [24] znajdują się zdjęcia (01 - 13) pokazujące wygląd oraz pracę licznika rowerowego.

6.3. Kalibracja

Kalibracja urządzenia miała nastąpić poprzez wykorzystanie oprogramowania producenta U-center. Oprogramowanie to umożliwia wgląd w dane odbierane przez moduł GPS oraz kalibrację modułu poprzez opcje konfigurowania urządzenia. Dokonano zmiany częstotliwości pracy urządzenia z domyślnej wartości 1Hz na 5Hz oraz ustawiono szacowany błąd pomiarowy pomiaru prędkości na wartość 0.1 metra na sekundę. Nie udało się zmienić sprzętowo formatu odbieranej wartości godziny, gdyż moduł działa tylko w formacie UTC. Korektę zastosowano jednak w kodzie programu, co przedstawione zostało w rozdziale nr 5. Nagranie umieszczone w plikach dodatkowych [24] ukazuje referencję mierzonej przez licznik rowerowy prędkości do prędkościomierza samochodowego.

6.4. Bilans energetyczny

Licznik rowerowy (podczas ciągłej pracy) pobiera ok. 90mA z akumulatora litowo-polimerowego. Wartość tą zmierzono wykorzystując multimetr. W tabeli nr 1 zestawiono wartości zmierzone multimetrem oraz wartości pochodzące z not katalogowych [24].

Tabela 1: Bilans energetyczny

Nr.	Nazwa elementu	Pobór zmierzony	Pobór deklarowany
1	GY-NEO6MV2	46.1mA	47mA
2	STM32F303K8	40.7mA	40mA
3	HD44780	794uA	600uA
4	TP4056	370uA	500uA
5	MT3608	-	1uA
6	HC-06 ZS-040	-	40mA
Suma:		87.964mA	128.1mA

Wartość poboru prądu przez przetwornicę MT3608 jest pomijalny.

Moduł Bluetooth został odłączony od zasilania w celach wydłużenia czas pracy urządzenia (Bluetooth nie znalazł zastosowania w projekcie).

Urządzenie zapewnia dobrą wydajność energetyczną przy zastosowanym akumulatorze o pojemności 750mA. Ciągła praca powoduje, że akumulator rozładowuje się po czasie ok. 8 godzin i 20 minut. Czas pracy obliczono za pomocą wzoru nr 2.

$$T = \frac{\text{Pojemność akumulatora}}{\text{Pobór prądu}} = \frac{750mAh}{90mA} = 8.33(3)h \quad (2)$$

W celu nadążenia do rozwiązań rynkowych należałoby zastosować akumulator o większej pojemności (pojemność na poziomie przynajmniej 1200mAh zapewni 13 godzin ciągłej pracy urządzenia).

6.5. Testy funkcjonalności

6.5.1. Warunki

Urządzenie umieszczone było w obudowie, a wszelkie elementy przytwierdzone, usztywnione i wzmocnione wspornikami (tulejkami) wydrukowanymi na drukarce 3D.

Obudowa wraz z urządzeniem została zamontowana na kierownicy rowerowej za pomocą zacisków znajdujących się na tyle obudowy.

Na potrzeby kalibracji podczas testów zdecydowano się na wykonanie pomiarów bez górnej warstwy obudowy. Zapewniało to szybki dostęp do komponentów urządzenia.

Testy odbywały się w kontrolowanych i bezpiecznych warunkach, bez narażania osób trzecich na potencjalny wypadek.

Urządzenie testowano w warunkach zimowych.

6.5.2. Godzina

Godzinę porównywano z godziną wyświetlaną na ekranie telefonu komórkowego oraz ekranie monitora przenośnego komputera osobistego.

6.5.3. Data

Datę porównywano do daty wyświetlanej na ekranie telefonu komórkowego oraz ekranie monitora przenośnego komputera osobistego.

6.5.4. Długość i szerokość geograficzna

Testy wykonywano z pomocą laptopa. Podczas przemieszczania się za pomocą roweru zatrzymywano się w losowych punktach i sprawdzano położenie geograficzne za pomocą Google Maps [15].

Wykonano także okrężenia wokół wyznaczonych punktów, aby sprawdzić czy wracając w to samo miejsce wartości pomiaru będą zbliżone.

6.5.5. Prędkość

Pomiary prędkości dokonywano poprzez nabranie stałej prędkości oraz późniejszym zatrzymaniu się. Przy próbach osiągnięcia wyższych prędkości licznik działał poprawnie. Z przyczyn bezpieczeństwa nie nagrano pracy urządzenia zamontowanego na kierownicy rowerowej, osiągając znaczną stałą prędkość poruszania się.

Dla zweryfikowania poprawności pomiarów użyto licznika samochodowego, wyciągając urządzenie z obudowy i porównując prędkość wyświetlaną na wyświetlaczu LCD z prędkością wskazywaną przez prędkościomierz samochodu. Testy z udziałem samochodu wykonywane były na drodze dojazdowej do prywatnej posesji bez osób znajdujących się na lub nieopodal drogi.

Na nagraniach [24] widoczny jest pomiar prędkości wyświetlany w drugiej linii wyświetlacza, który pokazuje prędkość poruszania się w węzłach. Stanowił on porównanie dla prędkości mierzonej w km/h, gdyż po przejrzeniu nagrań obliczano zbieżność pomiarów.

Filmy wykazujące poprawność oraz w przypadku słabego sygnału satelitarnego niepoprawność pomiarów zostały umieszczone w plikach dodatkowych [24].

Film z wykorzystaniem samochodu został wykonany z pomocą osoby trzeciej, siedzącej na miejscu pasażera tak, aby zachować ostrożność oraz bezpieczeństwo podczas dokumentacji.

6.6. Dokładność pomiarów

6.6.1. Warunki

Dane pomiarowe wyświetlane są na wyświetlaczu LCD o wymiarach 2x8. Wyświetlane są niezbędne dane oraz znaki specjalne stanowiące informację o jednostkach oraz opisujące informacje pomiarowe.

6.6.2. Godzina

Wyświetlana godzina zgadza się z aktualną występującą w formacie UTC+1.

6.6.3. Data

Wyświetlana data pomiaru zgadza się z aktualną.

6.6.4. Długość i szerokość geograficzna

Dane dotyczące lokalizacji odbiegają od rzeczywistych w przybliżeniu o ok. 4m. Dokładność oceniono na podstawie lokalizacji pochodzącej z GoogleMaps.

6.6.5. Prędkość

Dane dotyczące aktualnej prędkości nie odbiegają od rzeczywistych. Dokładność oceniono na podstawie testów wykonanych z zastosowaniem referencji w postaci licznika samochodowego [24].

Rozdział 7

Podsumowanie

Pomiary dotyczące aktualnej godziny, daty, danych geograficznych oraz prędkości w zupełności spełniają założenia projektowe i z dużą dokładnością oddają rzeczywiste wartości. W pełni zrealizowano plan pracy oraz wypełniono wymagania projektowe.

Wytrzymałość anteny dołączonej do modułu GPS jest słaba. Z pomiarów środowiskowych wynika, że antena jest podatna na uszkodzenia oraz ma trudności z zalogowaniem się do systemu satelitarnego pomimo sprzyjających warunków pomiarowych (przebywanie na otwartej, niezadaszonej przestrzeni). Na potrzeby pracy zostały zakupione trzy moduły GPS tego samego modelu, dla których dwie anteny uległy uszkodzeniu uniemożliwiając dalszych prac. W dodatkowych plikach dołączonych do pracy [24] zamieszczono zdjęcia obrazujące jedną z uszkodzonych anten oraz proces jej naprawy. W naprawie wykorzystano stację lutowniczą. Regeneracja anten spowodowała jednak brak możliwości stabilnego umieszczenia ich w obudowie urządzenia, gdyż zaprojektowana ona została dla standardowej długości przewodu, a te zostały skrócone.

Podczas optymalizacji kodu programu, urządzenie przestało działać poprawnie. Wstępne oględziny wykazały, że poprawność kodu powinna była zapewnić całkowitą funkcjonalność urządzenia projektowego. Po długim okresie testowania okazało się, że mikroprocesory STM32 serii L mogą mieć problem z definiowaniem oraz odczytywaniem danych z tablic dwuwymiarowych (za pomocą wskaźników). Po wymianie mikroprocesora na procesor z serii F, urządzenie odzyskało pełną sprawność. Przyczyną problemu mogło być również środowisko STM32CubeIDE, które podczas optymalizacji kodu wprowadziło aktualizację swojego oprogramowania. Po tej aktualizacji kompilator nie egzekwował w pełni poleceń kompilacji, powodując powstawanie błędów oraz samoczynnego wyłączania się środowiska lub wyświetlania komunikatu z błędem dotyczącym kompilacji programu.

Z uwagi na ograniczony czas oraz główną część projektu w postaci modułu GPS, moduł Bluetooth, który został zaimplementowany w schemacie licznika rowerowego, ostatecznie nie znalazł zastosowania w końcowym rozwiązaniu.

Podczas testów urządzenia okazało się, że podświetlenie wyświetlacza nie działa poprawnie. Po analizie problemu wykazano, że dioda LED umieszczona na układzie rozwojowym NUCLEO, podłączona bezpośrednio do portu wykorzystywanego jako programator podświetlenia, pobiera prąd przeznaczony do jego zasilenia. Powstał w ten sposób błąd konstrukcyjny, który skutkował zmianą koncepcji działania urządzenia. Rozwiązaniem tego problemu byłoby w tym przypadku zastosowanie tranzystora bipolarnego, zwiększającego moc sygnału docierającego do portu wejściowego anody wyświetlacza ciekłokrystalicznego (LCD). Pomimo obliczenia prawidłowej wartości rezystora R3 jako 80Ω , zastosowano rezystor o wartości 200Ω , co w żaden sposób nie przekłada się na funkcjonowanie podświetlenia z powodu wcześniej omówionego błędu. Stąd prosty wniosek, że szczególną uwagę powinno przykładać się do pierwszych etapów projektowych, w których dobiera się podzespoły oraz przy konstruowaniu schematów elektrycznych wraz z tworzeniem połączeń między modułami.

Nagrania pokazujące działanie urządzenia należało zmodyfikować w celu zaznaczenia ich autorstwa oraz opisanie wykonywanych czynności. Zrobiono to z wykorzystaniem darmowej aplikacji FilmForth [25] oraz narzędzia Pazera [26] (konwersja wideo).

Antena fabrycznie dołączona do modułu nie zapewnia komfortowej siły sygnału, przez co moduł często traci połączenie podczas użytkowania. Zaleca się użycie zamiennika.

Własnoręcznie zaprojektowana obudowa zapewnia spójność komponentów oraz ciekawy wygląd. Nie jest jednak w pełni szczelna, a demontaż obudowy wraz z wymianą poszczególnych elementów jest czasochłonny i wymaga nieco precyzji oraz delikatności.

W pracach rozwojowych planowane jest: wykorzystanie modułu Bluetooth (zainstalowanego w układzie, ale niewykorzystanego) w celu komunikacji licznika z innym urządzeniem/odbiornikiem, minimalizacja rozmiarów urządzenia, zwiększenie zakresów pomiarowych oraz wymiana wyświetlacza LCD na wyświetlacz OLED.

Do pracy dołączono link do wirtualnej płyty [24], w której skład wchodzi: streszczenie pracy, zdjęcia wykonane podczas prac, filmy zawierające testy urządzenia, kod źródłowy programu, schematy, instrukcję obsługi, wykorzystane biblioteki, kopię „bibliografii internetowej”, pliki oprogramowania, noty katalogowe oraz licencje do oprogramowania oraz bibliotek.

Bibliografia

- [1] https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps (dostęp: 28.12.2021)
- [2] Thuy Mai. https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html (dostęp: 28.12.2021)
- [3] Paweł Kułakowski, <http://www.kt.agh.edu.pl/~brus/satelite/navi.html> (dostęp: 28.12.2021)
- [4] Aleksander Kurczyk. *Mikrokontrolery STM32 dla początkujących*. BTC, Legionowo 2019
- [5] <https://kross.eu/pl/akcesoria/akcesoria-rowerowe/liczniki/licznik-krc-545gps> (dostęp: 28.12.2021)
- [6] <https://kross.eu/pl/akcesoria/akcesoria-rowerowe/liczniki/licznik-krc-211uw> (dostęp: 28.12.2021)
- [7] <https://www.a-hobby.pl/akumulator-litowo-polimerowy-akyga-3-7v-750mah-pcm-1403450-kartapdf-1898.html> (dostęp: 28.12.2021)
- [8] <http://www.tp4056.com/d/tp4056.pdf> (dostęp: 03.01.2022)
- [9] <https://prom-electric.ru/media/MT3608.pdf> (dostęp: 03.01.2022)
- [10] https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf (dostęp: 28.12.2021)
- [11] <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf> (dostęp: 28.12.2021)
- [12] https://www.st.com/resource/en/user_manual/dm00231744-stm32-nucleo32-boards-mb1180-stmicroelectronics.pdf (dostęp: 28.12.2021)
- [13] <https://pl.mouser.com/datasheet/2/306/hc06-1324356.pdf> (dostęp: 28.12.2021)
- [14] <https://www.autodesk.com/products/eagle/free-download> (dostęp: 28.12.2021)
- [15] <https://www.google.pl/maps/@50.2902326,18.6727104,15.25z> (dostęp: 28.12.2021)

-
- [16] <https://www.u-blox.com/en/product/u-center> (dostęp: 28.12.2021)
- [17] https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf
(dostęp: 28.12.2021)
- [18] <https://www.putty.org> (dostęp: 28.12.2021)
- [19] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwim0vKa8az1AhUEv4sKHQ6UB5kQFnoECAYQAQ&url=https%3A%2F%2Fwww.dpm.eu%2Ffile%2F4527%2F_.pdf&usg=AOvVaw06vRICnFKZOba6euI-iqYM
(dostęp: 28.12.2021)
- [20] Glenn Baddeley. <http://aprs.gids.nl/nmea/> (dostęp: 28.12.2021)
- [21] Radosław Kwiecień. <https://radio.dxp.pl/hd44780/> (dostęp: 28.12.2021)
- [22] https://www.keil.com/support/man/docs/armcc/armcc_chr1359124998347.htm
(dostęp: 03.01.2022)
- [23] Mateusz Salamon. https://github.com/lamik/GPS_NEO6_STM32_HAL
(dostęp: 28.12.2021)
- [24] Dodatkowe pliki - https://polslpl-my.sharepoint.com/:f/g/personal/adamsko563_student_polsl_pl/ElyKVTIgL2BFoQ0ZhNAAKDUBV24CngNi9qblMviFILMCdA?e=yeFG7z (dostęp: 28.12.2021)
- [25] <https://www.ioforth.com> (dostęp: 28.12.2021)
- [26] <http://www.pazera-software.pl/products/free-audio-video-pack/> (dostęp: 28.12.2021)

Dodatki

Spis skrótów i symboli

<i>GPS</i>	<i>system nawigacji satelitarnej (ang. Global Positioning System)</i>
<i>LCD</i>	<i>wyświetlacz ciekłokrystaliczny (ang. Liquid-Crystal Display)</i>
<i>PCB</i>	<i>obwód drukowany (ang. Printed Circuit Board)</i>
<i>NMEA</i>	<i>protokół komunikacyjny (ang. National Marine Electronics Association)</i>
<i>ASCII</i>	<i>siedmiobitowy system kodowania znaków (ang. American Standard Code for Information Interchange)</i>
<i>UART</i>	<i>uniwersalny asynchroniczny nadajnik-odbiornik (ang. Universal Asynchronous Receiver-Transmitter)</i>
<i>FLASH</i>	<i>rodzaj pamięci stanowiącej rozwinięcie pamięci EEPROM</i>
<i>EEPROM</i>	<i>elektrycznie kasowalna i programowalna pamięć tylko do odczytu (ang. Electrically Erasable Programmable Read-Only Memory)</i>
<i>UTC</i>	<i>uniwersalny czas koordynowany (ang. Coordinated Universal Time)</i>
<i>SPI</i>	<i>szeregowy port urządzeń peryferyjnych (ang. Serial Peripheral Interface)</i>
<i>I2C</i>	<i>szeregowa, dwukierunkowa magistrala (ang. Inter-Integrated Circuit)</i>
<i>USB</i>	<i>uniwersalna magistrala szeregową (ang. Universal Serial Bus)</i>
<i>LED</i>	<i>dioda elektroluminescencyjna (ang. Light-Emitting Diode)</i>
<i>OLED</i>	<i>organiczna dioda elektroluminescencyjna (ang. Organic Light-Emitting Diode)</i>
<i>GPIO</i>	<i>wejście/wyjście ogólnego przeznaczenia (ang. General-Purpose Input/Output)</i>
<i>RX</i>	<i>dana odbierana (ang. Receiving)</i>
<i>TX</i>	<i>dana wysyłana (ang. Transmitting)</i>
<i>BLUETOOTH</i>	<i>standard bezprzewodowej komunikacji krótkiego zasięgu</i>

Lista dodatkowych plików uzupełniających tekst pracy

Do pracy dołączono dodatkowe pliki zawierające:

- streszczenie pracy,
- zdjęcia urządzenia (w obudowie i bez obudowy),
- filmy przedstawiające pracę urządzenia,
- kod źródłowy,
- schemat urządzenia,
- schemat PCB,
- instrukcję obsługi,
- biblioteki (+ licencje),
- oprogramowanie (+ licencje),
- kopię bibliografii internetowej,
- noty katalogowe.

Spis rysunków

Rysunek 1: Schemat blokowy licznika rowerowego.....	9
Rysunek 2: Schemat idowy licznika rowerowego STM	10
Rysunek 3: Projekt układu drukowanego PCB	11
Rysunek 4: Struktura podstawowa programu.....	20
Rysunek 5: Odczyt komunikatów z modułu GPS	21
Rysunek 6: Wywołanie funkcji oraz wybór przechwytywanych protokołów.....	22
Rysunek 7: Algorytm na obliczanie sekund kątowych	23
Rysunek 8: Algorytm dla poprawy formatu wyświetlanej godziny	23
Rysunek 9: Funkcja waitus dla biblioteki opóźnień.....	24
Rysunek 10: Deklaracja przycisku S0	25
Rysunek 11: Deklaracja przycisku S1	25
Rysunek 12: Układ docelowy.....	28
Rysunek 13: Układ drukowany	28
Rysunek 14: Obudowane urządzenie umieszczone na kierownicy rowerowej.....	29
Rysunek 15: Urządzenie bez panelu przedniego obudowy	29

Spis tabel

Tabela 1: Bilans energetyczny	30
-------------------------------------	----