Politechnika Poznańska
Wydział Elektroniki i Telekomunikacji

Praca magisterska

Zegar czasu rzeczywistego synchronizowany protokołem NTP

Mirosław Woś

Promotor: dr inż. Krzysztof Lange

Poznań 2007

Spis treści:

1. Wstęp.....3

1.1 Wprowadzenie.....3

1.2 Cel i zakres pracy.....4

2. Odmierzanie czasu.....5

2.1 Własności czasu.....5

2.2 Postulat odmierzalności czasu.....6

2.3 Metody pomiaru czasu.....8

2.3.1 Pomiar czasu metodą astronomiczną.....8

2.3.2 Pomiar czasu oparty o zliczanie drgań mechanicznych-zegary mechaniczne.....10

2.3.3 Pomiar czasu oparty na generatorze kwarcowym.....11

2.3.4 Zegar rubidowy.....15

2.3.5 Zegar cezowy.....19

2.3.6 Pomiar czasu oparty o maser wodorowy.....26

2.3.7 Fontanny cezowe.....29

2.4 Porównanie metod pomiaru czasu.....32

3. Wybrane metody synchronizacji czasu.....35

3.1 Synchronizacja za pomocą protokołu NTP…….....35

3.1.1 Architektura systemu …….....35

3.1.2 Tryby pracy…….....38

3.1.3 Format danych…….....41

3.1.4 Procedury NTP…….....45

3.1.5 Zegar lokalny…….....51

3.1.6 Zastosowanie synchronizacji NTP…….....55

3.2 Synchronizacja czasu poprzez sieć GPS…….....56

3.2.1 Jednodrogowa metoda synchronizacji poprzez GPS…….....57

3.2.2 Dwudrogowa metoda synchronizacja czasu poprzez GPS…….....58

4. Projekt zegara synchronizowanego protokołem NTP…….....…….58

4.1 Zegar mikroprocesorowy…….....59

4.1.1 Blok zasilania…….....59

4.1.2 Blok sterowania……..... .61

4.1.3 Blok wyświetlaczy…….....64
4.1.4 Płyty obwodów drukowanych…….....65
4.2 Opis oprogramowania zegara mikroprocesorowego…….....66
4.3 Opis techniczny oprogramowania komputera PC.....70
4.4 Instrukcja obsługi oprogramowania komputera 77
5. Literatura.....79
6.Załączniki.....81
Załącznik A1. Schemat sterowania zegarem.....81
Załącznik A2. Schemat obwodów bloku wyświetlaczy.....82
Załącznik A3. Schemat obwodów bloku zasilania.....83

1. Wstęp

1.1 Wprowadzenie

Często nie zdajemy sobie sprawy, jak bieg naszego codziennego życia związany jest z biegnącym czasem i zjawiskami z jego upływem związanymi. Rzadko także zastanawiamy się czym jest ów czas pomimo, że oznaki jego upływu widzimy dookoła siebie. Z punktu widzenia filozofii jest niemal tyle spojrzeń na czas ile prądów filozoficznych.

Od zawsze cykle życia ludzkiego upływały zgodnie ze zmianami pór dnia, zmianami pór roku, a te, związane z upływem czasu synchronizowały codzienne czynności z jego biegiem. Dziś tym bardziej cykl życia ludzkiego zależny jest od czasu i tym częściej musi być z nim synchronizowany – poprzez spoglądanie na zegary, słuchanie sygnałów czasu za pośrednictwem odbiorników radiowych i wielu innych źródeł sygnałów biegnącego czasu.

W niniejszej pracy przedstawiono nieco szerszą problematykę związaną z możliwościami pomiaru i synchronizacji czasu, niż wynikało by to z tytułu. Obok projektu i opisu zegara synchronizowanego protokołem NTP, przedstawiono podstawy teoretyczne na temat upływu czasu, tworzenia jego reprezentacji, możliwości i technik jego pomiaru oraz przesyłania sygnałów czasu. Następnie skupiono się na wzorcach czasu i metodach ich tworzenia, a w dalszej kolejności scharakteryzowano różne metody synchronizacji czasu aby w końcu skupić się na jednej, wykorzystanej w projekcie. Projekt zawiera nie tylko praktyczną realizację algorytmu synchronizacji w formie kodu programu, ale tak

czasu nam - użytkownikom komputerów. Taka właśnie synchronizacja czasu jest wymagana w przedstawionym projekcie.

1.2 Cel i zakres pracy

Celem niniejszej pracy jest zaprojektowanie i wykonanie cyfrowego zegara czasu rzeczywistego ze wskaźnikiem opartym o cztery siedmiosegmentowe wyświetlacze LED, synchronizowanego z wybranym wzorcem atomowym poprzez sieć Internet, wykorzystując do tego celu komputer wyposażony w interfejs sieciowy. Komputer ma być połączony z zegarem poprzez port COM. Synchronizacja czasu poprzez sieć ma wykorzystywać wyspecjalizowany protokół NTP.

Celem projektu jest wykonanie zegara który nie będzie wrażliwy nawet na długie zaniki napięcia oraz zawsze będzie zsynchronizowany ze źródłem wzorcowym.

Struktura pracy jest następująca:

- Rozdział drugi obejmuje wstęp teoretyczny dotyczący zagadnień związanych z podstawami i metodami pomiaru czasu, oraz ich praktycznymi realizacjami.
- Rozdział trzeci poświęcony jest najważniejszym metodom synchronizacji czasu, a w szczególności synchronizacji poprzez sieć przy wykorzystaniu protokołu NTP.
- Rozdział czwarty obejmuje praktyczną realizację związanego z niniejszą pracą projektu, a w szczególności opis techniczny zegara oraz oprogramowania PC. Zawiera ponadto szczegółową instrukcję obsługi zegara oraz oprogramowania.
- Rozdział piąty zawiera przegląd wykorzystanej przy pisaniu pracy literatury.

2. Odmierzanie czasu

2.1 Własności czasu

Choć nie znamy natury czasu, jest on jedną z podstawowych wielkości fizycznych określająca kolejność zdarzeń oraz odstępy między zdarzeniami [1]. W fizyce klasycznej jest samodzielną wielkością niezależną od innych wielkości biegnącą w takim samym rytmie w całym wszechświecie. Jest więc absolutny i obiektywnie jednakowy. Izaak Newton twierdził, że „Absolutny, prawdziwy i matematyczny czas sam z siebie, ze swojej własnej natury płynie równomiernie bez żadnej relacji do czegokolwiek zewnętrznego". To spojrzenie na czas obaliła po wielu dyskusjach dopiero szczególna teoria względności A. Einsteina [5]. Zmieniła ona podstawy postrzegania czasu jako wielkości zależnej od przyspieszenia i grawitacji. Czas podlega tu zjawisku dylatacji zgodnie z transformacją Lorenza [5]

$$t \rightarrow t' = \frac{t - \frac{v^2}{c^2}}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{2.1}$$

Gdzie:

c – prędkość światła

v – prędkość poruszania się układu nie będącego w spoczynku

t – czas w spoczywającym układzie odniesienia

t'- czas w poruszającym się układzie odniesienia.

Zgodnie z wzorem (2.1) czas jaki mija pomiędzy dwoma zdarzeniami nie jest jednoznacznie określony, lecz zależy od obserwatora. Różni obserwatorzy obserwują, to samo zdarzenie w różnych momentach czasowych. Czas trwania zjawiska, zachodzącego w punkcie przestrzeni, obserwowany z punktów poruszających się względem danego punktu, jest dłuższy niż czas trwania tego zjawiska w układzie odniesienia, w którym punkt ten spoczywa [5]. Zatem w poruszającym się układzie

odniesienia z punktu widzenia obserwatora będącego w układzie spoczywającym zegar porusza się z prędkością *v*, a czas który upływa między dwoma uderzeniami zegara nie jest równy jednej sekundzie, ale [5]:

$$\frac{1}{\sqrt{1-v^2/c^2}}\,sekund \tag{2.2}$$

Ruch sprawia, że zegar porusza się wolniej niż w stanie spoczynku. Zjawisko to prowadzi między innymi do szeroko dyskutowanego paradoksu bliźniąt.

Kolejnym wynikającym ze szczególnej teorii dla nas wnioskiem jest to, że największą możliwą prędkością jest prędkość światła w próżni, czyli 299792458 m/s, zatem nawet w idealnym przypadku (tzn. kiedy jedyne opóźnienie synchronizacji wnosi czas propagacji sygnału) nie możemy mówić o synchronizacji absolutnej zegarów oddalonych, lecz musimy uwzględnić chociażby owe opóźnienia związane z czasem propagacji sygnału synchronizującego. Właściwości czasu jakie znamy z mechaniki klasycznej obowiązują jednak w nieruchomych układach odniesienia znajdujących się w jednorodnych polach grawitacyjnych i w bardzo dużym przybliżeniu takie właśnie obowiązują w warunkach ziemskich [5].

2.2 Postulat odmierzalności czasu

Skoro nie rozumiemy istoty czasu tym bardziej trudne zdaje się być odmierzanie jego upływu. Czas jest przecież wielkością niematerialną, nie oddziałującą bezpośrednio z materią. Pomiar czasu, tak jak i generowanie wzorcowych sygnałów czasu wiąże się z obserwacją i zliczaniem zjawisk występujących okresowo w czasie, opierając się na postulacie odtwarzalności zjawisk. Postulujemy, że jednakowe zjawiska występują po sobie mają takie same czasy trwania [1]. Zauważmy, że opieramy się jedynie na postulacie, że w danym układzie odniesienia czas upływa jednostajnie dla obserwatora i zjawiska odtwarzalne mają w rzeczywistości identyczne czasy trwania. Oczywiście podczas pomiarów musimy wziąć pod uwagę fakt, że niemożliwe jest występowanie idealnie takich samych zjawisk, a co za tym idzie pomiar czasu na ich podstawie będzie obarczony większym, lub mniejszym błędem. Zatem w praktyce odmierzania czasu dokonujemy wykorzystując zjawiska, które w podobnych warunkach

przebiegają podobnie (nie chaotycznie, gdyż w rzeczywistych układach nie da się wytworzyć ściśle takich samych warunków oddziaływania) [1].

Każde z powtarzalnych zjawisk oznacza chwilę, która jest punktem na osi czasu, a zatem, skoro jest punktem przyjmujemy założenie o nieskończenie krótkim czasie trwania zdarzenia. Dzięki takiej identyfikacji poszczególne chwile można oznaczyć. Tworzony w ten sposób system uporządkowanych zdarzeń nazywamy skalą czasu [1].

Pojęcie skali czasu jest jednak bardzo trudne do jednoznacznego zdefiniowania, dlatego właśnie praktyczne realizacje skal czasu są uznawane jako ich jedyne definicje. Nie można zatem mówić o żadnej idealnej skali czasu, którą można by zrealizować, czy znaleźć w naturze [6].

Wracając jednak do przedstawionego postulatu należy odpowiedzieć na pytanie: Co odmierzają zegary? Jak stwierdzono w postulacie odmierzanie czasu opiera się jedynie na zliczaniu podobnych zjawisk (stanów) występujących okresowo, czyli na badaniu fazy stanów. Zatem zegary odmierzają fazy stanów wzorcowych (takich jak np.: wychylenia wahadła, drgnięcia płytki kwarcu, okresy promieniowania atomów cezu $Ce^{133)}$ które następnie powiązane z przyrostem fazy czasu idealnego dają w wyniku czas T(t) podany w jednostkach czasu idealnego w którym to czasie zegar wygenerowałby fazę Φ(t), gdyby jego rzeczywista szybkość przyrostu równała się szybkości nominalnej. Ponieważ jednak tak nie jest to wskazania danego zegara będą z reguły różne od czasu idealnego. Tak więc sekunda czasu idealnego będzie z reguły różnić się od sekundy wskazywanej przez zegar (przyjętej obecnie jako podstawową jednostkę czasu SI i zdefiniowanej jako czas trwania 9 192 631 770 okresów promieniowania odpowiadającego przejściu między dwoma nadsubtelnymi stanami atomu cezu Ce^{133}). Z przedstawionych powyżej argumentów wynika, że [1]:

$$T(t) = \frac{\Phi(t)}{2\pi\nu_{nom}} \tag{2.3}$$

Gdzie:

T(t) - jest czasem wskazywanym (postrzeganym) przez zegar

Φ(t) - faza całkowita generowana przez zegar

ν_{nom} - częstotliwość postulowana (nominalna)

Zatem sygnał generowany przez zegar może przy właściwym wykorzystaniu (opisie) spełniać funkcję sygnału czasu (jak i synchronizacji) oraz być wykorzystywany do tworzenia skal czasu [1].

2.3 Metody pomiaru czasu

2.3.1 Pomiar czasu metodą astronomiczną

Początki pomiarów czasu były bardzo ściśle związane z rozwojem badań astronomicznych i były oparte jedynie na nich. Już dawno zauważono iż pozorny ruch Słońca i innych ciał niebieskich po sklepieniu niebieskim zdaje się wyznaczać upływ czasu. Pierwszym przyrządem służącym do określania czasu był gnomon, wynaleziony w około 2500 r. p.n.e.[7]. Pomiar czasu za pomocą gnomonu polegał na obserwacji ruchu rzucanego przezeń cienia (pełniącego rolę wskazówki) w ciągu dnia. Nie zdawano sobie jednak sprawy z tego, że zależnie od pory roku temu samemu kierunkowi i tej samej długości cienia odpowiadają różne godziny. Pierwszą zmianą dokonaną w celu polepszenia dokładności pomiarów było umieszczenie pręta rzucającego cień na tarczę godzinową nie pod kątem prostym do odcinków zawierających się w płaszczyźnie tarczy, lecz pod kątem [7]:

$$\Phi = 90^{o} - \beta \tag{2.4}$$

Gdzie:

Φ - kąt nachylenia pręta gnomonu w stosunku do płaszczyzny tarczy

β - szerokość geograficzna umiejscowienia zegara.

Jest to związane z tym, że pozorny ruch słońca po sklepieniu niebieskim nie odbywa się wzdłuż równika, lecz wzdłuż ekliptyki - koła wielkiego nachylonego w stosunku do równika pod pewnym kątem [8]. Wprowadzenie tej modyfikacji zapewniło dużą większą dokładność pomiaru czasu. Skoro więc ekliptyka jest nachylona do równika niebieskiego pod pewnym kątem to badanie ruchu Słońca po niej daje różne wyniki pomiaru nachylenia i długości cienia rzucanego przez gnomon (zależnego od aktualnej odległości kątowej gwiazdy względem równika niebieskiego). Dodatkowo ruch Słońca po ekliptyce, jako skutek niejednostajnego ruchu Ziemi po eliptycznej orbicie, jest także niejednostajny. W rezultacie aby pozorny ruch Słońca prawdziwego

(a więc i cień pręta) mógł określać aktualny czas należy związać ten ruch z ruchem fikcyjnego Słońca „średniego” poruszającego się wzdłuż równika niebieskiego. Równanie wiążące obydwa rodzaje ruchu nazywamy równaniem czasu, a jego wartość można przybliżyć następująco [8],[9]:

$$E = \alpha(\overline{\Theta}) - \alpha(\Theta) = -7{,}7\sin(79^o + l) + 9{,}5\sin(2l) \tag{2.5}$$

Gdzie:

$\alpha(\overline{\Theta})$ - rektascensja Słońca średniego

$\alpha(\Theta)$ - rektascensja Słońca prawdziwego

l - długość ekliptyczna Słońca prawdziwego

Zatem na każdy dzień można wyznaczyć dość dokładnie czas na podstawie pomiarów położenia słońca, znajomości położenia geograficznego oraz równania czasu (na podstawie wzoru (2.5)) bądź też odczytać jego wartości z tablic zamieszczanych w rocznikach astronomicznych. Równanie owo daje w wyniku wartość zerową tylko cztery razy w roku: 16 IV, 14 VI, 1 IX i 24 XII – wtedy słońce średnie i prawdziwe pokrywaj się. Wykreślenie podziałki zegara w tych dniach zagwarantuje nam, że będzie to podziałka „średnia”, najodpowiedniejsza na cały rok. Wartości maksymalne (wynoszące około kwadransa) równanie czasu osiąga dwa razy w roku (12 II i 3 XI) [9].

Pomiar czasu metodą astronomiczną może opierać się na określeniu pozornego ruchu Słońca podczas dnia, jak i na ruchu innych ciał niebieskich w nocy (zegar gwiazdowy). Do dziś czas astronomiczny odgrywa dla nas ważną rolę, bowiem to właśnie pozorny ruch Słońca wyznacza momenty takich zjawisk jak zachody i wschody Słońca (jak i innych ciał niebieskich) oraz wielu innych zjawisk. Należy wspomnieć o lokalności zjawisk na podstawie których wyznaczany jest czas astronomiczny. Wyznaczamy bowiem na ich podstawie *czas słoneczny lokalny* różny dla różnych długości geograficznych.

Pomiar czasu metodą astronomiczną, jest pomimo niskiej rozdzielczości pomiaru i wysokiemu błędowi odczytu metodą wysoce dokładną, szczególnie w dużej skali czasu, ze względu na wyjątkowa regularność zjawisk astronomicznych (wysoce stały czas obrotu Ziemi dookoła własnej osi).

2.3.2 Pomiar czasu oparty o zliczanie drgań mechanicznych-zegary mechaniczne

Znaczącym postępem w dziedzinie pomiaru czasu stało się pojawienie na początku XIV wieku nowych instrumentów do tego służących – zegarów mechanicznych. Pierwsze przyrządy tego typu były stosunkowo prymitywne, nie miały bowiem ani tarcz, ani wskazówek, rolę tę pełnił dzwon, który uruchamiany przez mechanizm wybijał godziny - zegar taki posiadał zatem bardzo mała rozdzielczość, równą jednej godzinie [7]. Początkowo zegary mechaniczne napędzane były ciężarami zawieszonymi na łańcuchach nawijanych na wały. Okres opuszczania się takowych odważników był jedynym źródłem sygnału czasu zegara, a więc jedynie od jednostajności opuszczania się ciężaru zależała dokładność pomiaru czasu przez zegar. Podstawowym czynnikiem wpływającym na pracę zegara było zatem, tarcie jego części, mogące przybierać różne wartość w zależności od temperatury i wilgotności (np.: poranna rosa miała ewidentny wpływ na przyspieszenie pracy mechanizmu). Ze względu zatem na małą stabilność zegary te miały jedynie wskazówkę godzinową. Znaczącym postępem pod względem stabilności pracy było wprowadzenie do budowy zegara wahadła pełniącego rolę regulatora chodu. Wychylenia wahadła dawały zegarowi sygnał taktujący dla odmierzania czasu. Dzięki wykorzystaniu zjawiska rezonansu w pracy wahadła, jego takt stał się stosunkowo regularny i w mniejszym stopniu zależny od tarcia mechanizmu. Okres wahadła można w prosty sposób określić następującym równaniem [7]:

$$T = 2\pi\sqrt{\frac{I}{mgd}} \tag{2.6}$$

Gdzie:

d – odległość od punktu zawieszenia do środka ciężkości

g – przyspieszenie ziemskie

I – moment bezwładności wahadła względem osi obrotu

m – masa wahadła

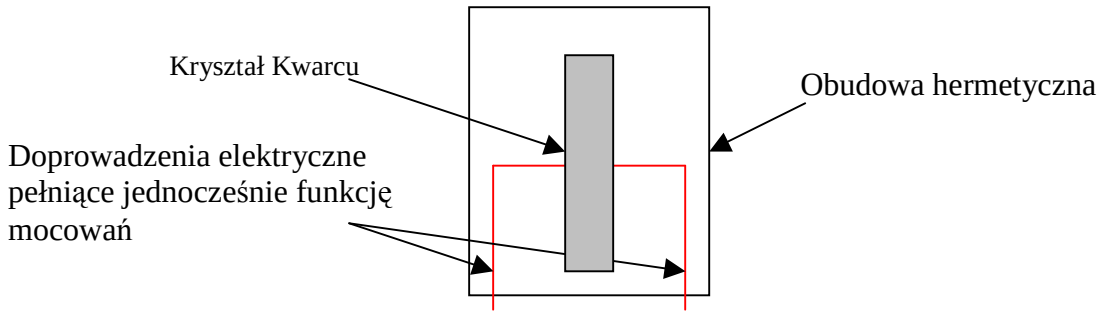
Jak wynika ze wzoru (2.6) okres drgania wahadła jest zależny od czterech czynników, zatem wystarczy regulować jeden z nich w celu dostrojenia pracy zegara. W celu zmiany okresu drgania wahadła stosuje się powszechnie obciążniki o regulowanym

punkcie zawieszenia (zmiana momentu bezwładności). Wynalezienie zegara z wahadłem pozwoliło na tworzenie nowych, innych od astronomicznych skal czasu.

Niestety na pracę wahadła ma wpływ wiele czynników, jak np.: tarcie, nierównomierność zasilania w energię mechaniczną, niedokładność wykonania wychwytu oraz zależność długości wahadła od temperatury. Przez lata udoskonalano zegary minimalizując wpływ poszczególnych czynników na ich pracę, aby w połowie XX wieku osiągnąć stabilność pracy lepszą niż 0.5 s na dobę. Równocześnie z pracami nad udoskonalaniem parametrów pracy zegarów trwały prace nad ulepszeniem ich zasilania w energię mechaniczną. W mniejszych zegarach wahadło i obciążniki zostały zastąpione przez balans ze spiralą (1675) (urządzenia spełniającego rolę wahadła) oraz sprężynę napędową. Pomimo ważnego miejsca zegarów mechanicznych w historii związanej z pomiarem czasu, dziś mechanizmy te znajdują coraz mniejsze zastosowanie, wyparte głównie przez stabilne, dobre, tanie i wytrzymałe zegary kwarcowe [7],[8].

2.3.3 Pomiar czasu oparty o zliczanie drgań elektromechanicznych – rezonator kwarcowy

Głównym elementem rezonatora jest odpowiednio oszlifowany kryształ kwarcu drgający z wysoce stałą częstotliwością. Działanie rezonatora kwarcowego opiera się na zjawisku piezoelektrycznym, polegającym na wzajemnym przetwarzaniu energii elektrycznej w mechaniczną. Częstotliwość drgań jest zależna głównie od jednego czynnika, a mianowicie od rozmiarów drgającej płytki kwarcu. Fakt ten odkrył W. G. Cady na początku lat dwudziestych XX wieku [12].

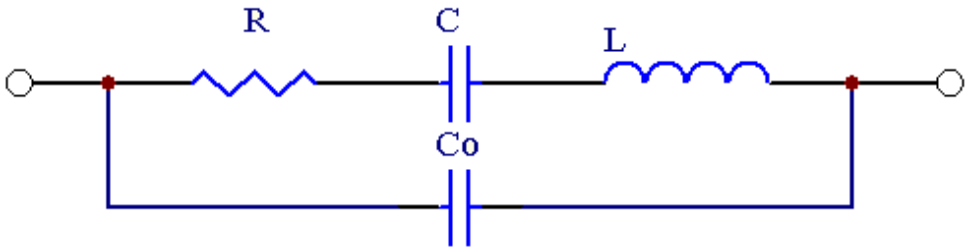


Rys. 2.1 Budowa rezonatora kwarcowego [12]

Już od początku zalety kryształu kwarcu zostały zastosowane do budowy zegarów i to z bardzo dobrym skutkiem – pierwszy zegar zbudowany przez Warrena Marrisona z

Laboratoriów Bella był bardziej precyzyjny od wszystkich mechanicznych zegarów. W owym zegarze (jak i w zegarach produkowanych do dzisiaj) wykorzystywane zostały piezoelektryczne właściwości kwarcu polegające na tym, ze pod wpływem nacisku lub rozciągania wzdłuż jednej osi krystalograficznej powstaje wypadkowy ładunek elektryczny. Oznacza to, że płytka kwarcu (przy odpowiednim ułożeniu sieci krystalograficznej względem jej ścian) poddana mechanicznym odkształceniom, wytwarza na przeciwległych jej ściankach różnicę potencjału. Jeżeli się ją natomiast umieści w zewnętrznym polu elektrycznym to odkształca się pod jego wpływem. Zatem po przyłożeniu do płytki kwarcu odpowiedniego napięcia powoduje się jej odkształcenie. Odkształcenie to z kolei wywołuje powstawanie na końcach płytki różnicy potencjałów - sygnału elektrycznego. Impuls ten z kolei powoduje odkształcenie płytki itd. doprowadzając do rezonansu elektromechanicznego wewnątrz odpowiednio zasilanej płytki piezoelektryka. Typowa budowa rezonatora kwarcowego przedstawiona jest na rysunku 2.1. Jak można na nim zauważyć rezonator jest umieszczony w hermetycznej obudowie (najczęściej metalowej, choć spotyka się także plastikowe o nieco gorszej jakości, stosowane tam gdzie nie jest potrzebna wysoka stabilność częstotliwości (np. we wszelkiego rodzaju pilotach do sprzętu RTV)) w której występuje próżnia celem ograniczenia strat spowodowanych tarciem gazów [12].

Po przyłączeniu takiego rezonatora do odpowiedniego obwodu elektrycznego możliwe jest zliczanie impulsów elektrycznych generowanych przez drgający układ. Drgania te są podtrzymywane za pomocą układów elektronicznych, np. układu generatora samowzbudnego tak skonstruowanego, ze płytka kwarcu stanowi jego główna część i spełnia funkcje stabilizatora drgań samowzbudnych. W porównaniu z oscylatorami mechanicznymi okres drgań w małym stopniu zależy od temperatury i innych czynników zewnętrznych, a w głównym stopniu od obróbki płytki kwarcowej – jej fizycznych rozmiarów. W celu określenia częstotliwościowych właściwości rezonatora za pomocą obwodu elektrycznego stosuje się tzw. ekwiwalent elektryczny:



Rys. 2.2 Ekwiwalent elektryczny rezonatora kwarcowego [13], [14].

Gdzie:

- pojemność C i indukcyjność L stanowiące szeregowy układ LC reprezentują częstotliwościowe właściwości kryształu kwarcu (gdzie częstotliwość rezonansowa przedstawionego układu LC jest równa rezonansowej częstotliwości drgań mechanicznych kryształu kwarcu).
- Pojemność Co reprezentuje pojemność doprowadzeń.
- Rezystancja R reprezentuje straty związane z rozchodzeniem się energii drgań mechanicznych kryształu kwarcu poprzez doprowadzenia oraz związane z niedoskonałością próżni występującej w obudowie rezonatora. Ma ona bardzo małą wartość i dlatego często w opisie matematycznym zachodzących przemian jest pomijana [13], [14].

Jak już wspomniano częstotliwość rezonansowa przedstawionego układu LC jest równa częstotliwości drgań mechanicznych kryształu kwarcu i wynosi [13],[14]:

$$f_s = \frac{1}{2\Pi\sqrt{LC}}\sqrt{1+\frac{C}{C_o}} \tag{2.7}$$

Gdzie f_s oznacza częstotliwość rezonansową generatora. W typowych generatorach kwarcowych pojemność C jest zazwyczaj o co najmniej 3 rzędy niższa od pojemności Co, zatem można pominąć wyrażenie $\frac{C}{C_o}$ upraszczając wzór (2.7) do postaci [13],[14]:

$$f_s = \frac{1}{2\Pi\sqrt{LC}} \tag{2.8}$$

Wykazujemy zatem, że częstotliwość drgań płytki kwarcowej zależy wyłącznie od jej mechanicznych właściwości – sposobu szlifowania [13],[14]. Na tej podstawie można określić w jaki sposób temperatura płytki kwarcowej wpływa na częstotliwość rezonansową. Otóż pod wpływem podwyższonej temperatury płytka kwarcu rozszerza się co ma wpływ na częstotliwość drgań mechanicznych (powoduje jej wzrost) oraz na podstawie wzoru (2.8) powoduje także wzrost częstotliwości drgań elektrycznych.

Jak już wspomniano rezystancja R jest tak mała, że z wysokim przybliżeniem można ją pominąć w obliczeniach. Dzięki temu impedancja Z rezonatora jest wysoce

stała i zależy jedynie od jego mechanicznych właściwości, a drgania mechaniczne przebiegają z bardzo małymi stratami energii (2.9) [13],[14].

$$Z = \frac{j}{\omega} \frac{\omega^2 LC - 1}{C_0 + C - \omega^2 LCC_0} \tag{2.9}$$

Gdzie:

ω - pulsacja rezonansowa, $\omega = 2\prod f$

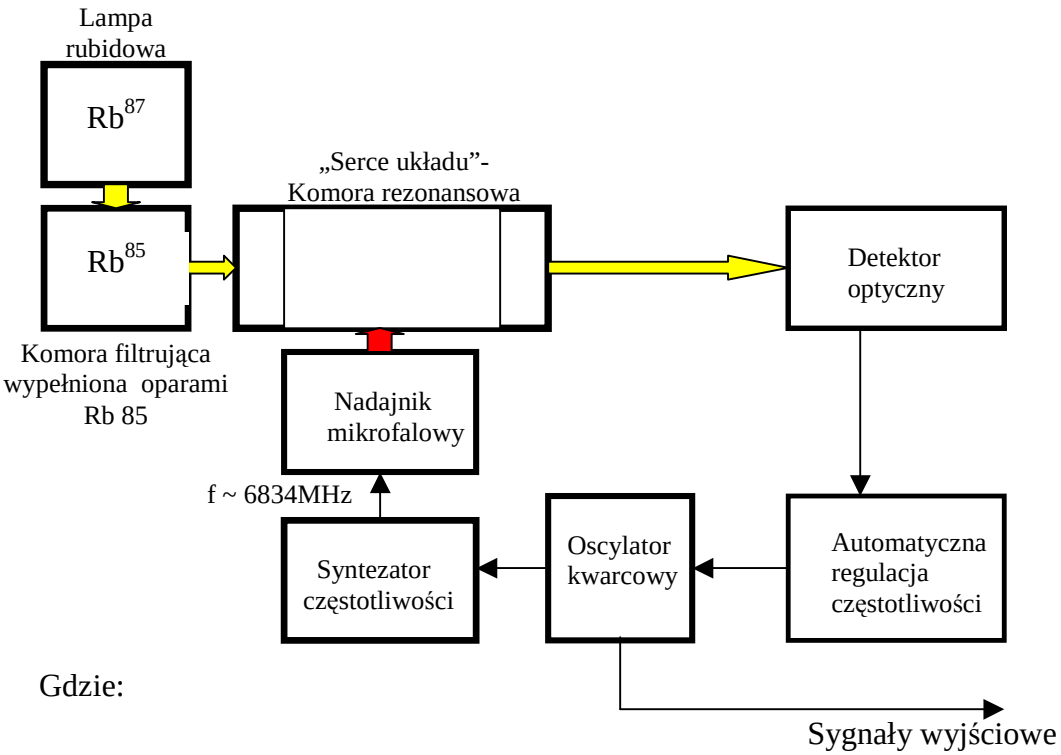
Uzyskaną częstotliwość drgań obniża się za pomocą dzielnika częstotliwości w takim stopniu, aby nadawała się do poruszania zegara synchronicznego [12]. Najczęściej stosowane kryształy kwarcu w zegarach codziennego użytku cechują się częstotliwością drgań równą 32768 Hz= 2^{15} Hz. Dzięki temu po zastosowaniu kaskady piętnastu dzielników częstotliwości otrzymujemy sygnał o okresie 1 s, wykorzystywany dalej jako podstawa czasu do aktualizacji rejestrów godzinowych i daty zegarów. Dzięki niskiej cenie, prostocie wykonania, małej podatności częstotliwości rezonansowej na czynniki zewnętrzne układy kwarcowe zyskały sobie ogromną popularność i są wykorzystywane jako podstawy czasu we właściwie wszystkich urządzeniach elektronicznych począwszy od mikrokontrolerów w sprzęcie AGD, poprzez zegarki, komputery osobiste, aż po układy taktujące w sieciach telekomunikacyjnych. Kryształy kwarcu są bardzo trwałe, jednak podlegają procesowi starzenia, co powoduje wzrost błędu. Należy wspomnieć, że stabilność krótkookresowa rezonatora i jego napędowego obwodu są znacznie lepsze niż jego stabilność długookresowa. Jeśli np.: porównamy czas wskazywany przez zegar atomowy i zegarek kwarcowy na rękę, utrzymywany w stałej temperaturze to różnica miedzy nimi wynosiłaby mniej niż 2 sekundy na miesiąc [12]. Najczęściej jednak tak nie jest, ponieważ pomimo tego, że pojedynczy egzemplarz rezonatora kwarcowego cechuje się wysoką stabilnością, to punkty pracy rezonatorów w całej serii produkcyjnej różnią się nieznacznie między sobą pod względem częstotliwości rezonansowej drgań, a to oznacza, że dla każdego z nich z osobna powinien być dobrany odpowiedni dzielnik częstotliwości, a nie za każdym razem ten sam. Źródłem sygnału taktowania w zegarze będącym tematem projektu w tej pracy także jest kryształ kwarcu. Dzięki niewielkim kosztom produkcji i dość dużej stabilności zegary kwarcowe zrewolucjonizowały przemysł tej dziedziny w dwudziestym wieku. Znakomita większość produkowanych obecnie zegarów opiera swe działanie właśnie na mechanizmie kwarcowym [12].

2.3.4 Zegar rubidowy

Oscylatory atomowe stanowią obecnie najdokładniejsze źródła czasu stosowane i uznane na całym świecie. Popularnym przedstawicielem atomowych wzorców czasu jest zegar rubidowy. Jako wzorce cechujące się najwyższym stosunkiem stabilności do ceny wśród zegarów atomowych zegary rubidowe są bardzo powszechnie stosowane. Wprowadzenie do użytku wzorców atomowych spowodowało powstanie nowej klasy dokładności pomiarów czasu.

W oscylatorze rubidowym jako źródło sygnału czasu wykorzystywana jest rezonansowa częstotliwość zmiany stanu atomów rubidu Rb^{87} (wynosząca 683 468 2614 Hz) względem, której porównywane jest wyjście rezonatora kwarcowego sterowanego napięciem (OCXO). Wyjście rezonatora kwarcowego powielane jest do częstotliwości rezonansowej zmiany stanu atomu rubidu Rb^{87} i używane do sterowania komory mikrofalowej. Wewnątrz komory mikrofalowej za pomocą detektora optycznego (który dalej synchronizuje wejście rezonatora kwarcowego sterowanego napięciem zapewniając jego stabilność średniookresową) wykrywane są zmiany stanu atomów Rb^{87} [19]. Pierwszy zrealizowany wzorzec częstotliwości tego typu powstał w wyniku pracy Carpentera (1960) i Arditi (1960).

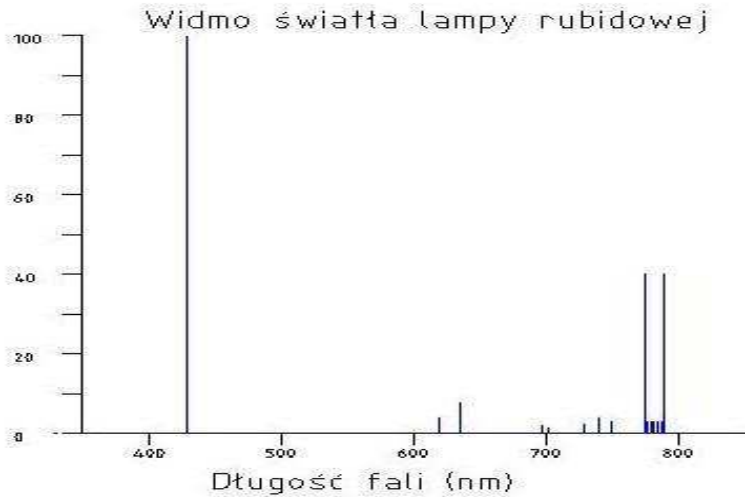
Najbardziej użytecznym okazuje się być pasywny wzorzec częstotliwości tego typu, ze względu na jego małe wymiary dzięki czemu jest bardzo mobilny i może być stosowany jako przenośny wzorzec służący do synchronizacji urządzeń telekomunikacyjnych, jako wzorzec wspomagający nawigację oraz jako wzorzec służący do badań astronomicznych z możliwością jego wyniesienia w przestrzeń kosmiczną jako część sztucznego satelity. Zachowuje przy tym jego wysoką stabilność częstotliwościową. Zegar rubidowy składa się z lampy rubidowej, komory filtrującej, komory rezonansowej, fotodetektora oraz oscylatora kwarcowego kontrolowanego napięciem sprzęgniętego za pomocą układu automatycznej regulacji częstotliwości (w celu dostrojenia pracy oscylatora z częstotliwością rezonansową atomów rubidu Rb^{87}) oraz syntezatora częstotliwości z nadajnikiem mikrofalowym. Blokowa budowa takiego zegara jest przedstawiona na rys. 2.3 [16]



Rys. 2.3 Budowa oscylatora rubidowego [16]

Zasada działania takiego zegara jest dość prosta z punktu widzenia współpracy poszczególnych bloków funkcjonalnych. Lampa zawierająca rubid Rb ^{87} emituje wiązkę optyczną. Widmo lampy rubidowej (w tym przypadku produkowanej przez firmę OSRAM) jest przedstawione na rys. 2.3 [17].

Do komory rezonansowej zegara należy wprowadzić odpowiednio monochromatyczne światło pobudzające przejścia atomów izotopu Rb ^{87} między ściśle określonymi nadsubtelnymi stanami. Dlatego właśnie światło pochodzące z lampy rubidowej musi zostać wprowadzone do komory zawierającej atomy izotopu rubidu Rb^{85} celem filtracji. Wewnątrz niej atomy rubidu w postaci oparów oświetlone światłem pochodzącym z lampy przechodzą na wyższy stan energetyczny pochłaniając część promieniowania (pompowanie optyczne) i odfiltrowując widmo światła lampy rubidowej z niepożądanych częstotliwości 2.4 [17].



Rys. 2.4 Widmo emisyjne lampy rubidowej (OSRAM) [17]

Odfiltrowane światło trafia do rezonansowej wnęki mikrofalowej wypełnionej parami rubidu Rb^{87} oraz mieszaniny lekkich gazów szlachetnych. Przedłużają one czas współdziałania atomów rubidu z promieniowaniem mikrofalowym przez zmniejszenie tempa kolizji atomów ze ściankami komory [18].

Następnie pod wpływem światła następuje depopulacja niższego nadsubtelnego stanu atomów rubidu (pompowanie optyczne) co w dalszej kolejności redukuje absorpcję światła w komorze rezonansowej. Po przejęciu przez komorę rezonansową światło trafia na fotodetektor badający poziom absorpcji światła. Jednocześnie atomy rubidu znajdujące się w komorze rezonansowej są poddawanie działaniu promieniowania mikrofalowego o częstotliwości regulowanej przez syntezator częstotliwości. Sygnał elektryczny z detektora optycznego reguluje częstotliwość oscylatora kwarcowego za pośrednictwem układu automatycznej regulacji częstotliwości w taki sposób aby zminimalizować sygnał pochodzący z detektora, a co za tym idzie aby zminimalizować ilość światła padającego na detektor. Dąży się zatem do maksymalizacji populacji atomów znajdujących się w stanie podstawowym. Stan ten osiąga się poprzez działanie na atomy promieniowaniem mikrofalowym o określonej częstotliwości w przybliżeniu równej $f_P \sim f_{0,}$gdzie f_0=6834682614 Hz. W przypadku odstrojenia od tej częstotliwości populacja stanu podstawowego gwałtownie spada a ilość światła padającego na fotodetektor rośnie. Zatem na postawie sygnału pochodzącego z detektora optycznego można określić odstrojenie oscylatora kwarcowego od częstotliwości przejeść pomiędzy dwoma nadsubtelnymi stanami

atomów rubidu. Otrzymujemy w ten sposób układ rezonansowy o maksymalnym odstrojeniu wynoszącym ok. 500Hz [18].

Obecnie zegary rubidowe są najtańszymi i najbardziej mobilnymi zegarami atomowymi. Znaczna ich liczba (kilka tysięcy rocznie) jest produkowanych i wykorzystywanych w polu rozwiązań telekomunikacyjnych. Wysoce wyspecjalizowane oscylatory rubidowe znalazły także zastosowanie w satelitach systemu GPS. Jest kilka powodów odgrywania przez zegary rubidowe ważnej roli wzorców częstotliwości. Najważniejszym jest jego dokładność i stabilność. Dokładność jest porównywalna ze wzorcem cezowym, przy czym czas pracy jest około 5 razy dłuższy od cezowego. Ponadto stabilność rubidowego wzorca częstotliwości w krótkich przedziałach czasowych, rzędu kilkuset sekund, jest lepsza od cezowego (wzorzec cezowy jest bardziej stabilny w dłuższych okresach obserwacji) [16], [18].

Należy jednak wspomnieć, że zegary rubidowe pomimo licznych zalet posiadają także znaczące wady. Względne różnice pomiędzy częstotliwością f_0=6834682614 Hz niezakłóconego przejścia atomów rubidu pomiędzy dwoma nadsubtelnymi stanami, a częstotliwością postrzeżoną wynoszą ok. 10^{-10} wartości całkowitej. Różnice te są spowodowane głównie przez pole magnetyczne występujące wewnątrz komory rezonansowej, przez kolizje atomów rubidu z gazem buforu oraz przez jednoczesne współdziałanie atomów ze światłem oraz promieniowaniem mikrofalowym. Ponadto zegar podlega procesowi starzenia a w raz z nim zmieniają się parametry jego pracy. Podstawowym elementem podlegającym zużyciu jest lampa rubidowa, która z biegiem czasu zmienia intensywność i widmo emitowanego światła (czas pracy nowoczesnych lamp powyżej 10 lat) jak również kompozycje gazów w komorze filtrującej i rezonansowej ulegające degradacji. Z tego powodu zegary rubidowe wymagają źródła synchronizacji – odniesienia (np.: cezowego) dla zachowania długoterminowej dokładności. Kolejną wadą jest stabilność termiczna wzorca rubidowego - gorsza od wzorca cezowego lub masera wodorowego [16],[18],[19]. Parametry przykładowego wzorca rubidowego oferowanego przez firmę Z.E.A.P. Meratronik S.A. przedstawione są w tabeli 2.1.

Tabela 2.1 Parametry wzorca rubidowego A10 oferowanego przez firmę Meratronik S.A [19]

Model:	• wersja do montażu w stojaku A10-R • wersja telekomunikacyjna A10-T
Wyjścia:	• sygnał sinusoidalny oraz prostokątny o częstotliwościach 1, 5, 10 MHz • 6 dowolnych wyjść o częstotliwościach sygnału 1, 5 lub 10 MHz, 1pps
Stabilność:	• $4*10^{-11}$/dzień
Szum fazowy:	• -155dB/Hz przy 10kHz
Odstrojenie:	• $4*10^{-11}$/ rok

Podsumowując należy zauważyć, że koszt rubidowego wzorca częstotliwości jest dużo niższy niż cezowego, przy znacznie zmniejszonych wymiarach i wadze oraz większej żywotności w porównaniu z droższym choć dokładniejszym zegarem cezowym. Dzięki małym wymiarom i wadze oraz wysokiej tolerancji na warunki środowiskowe rubidowy wzorzec częstotliwości jest najlepszy dla zastosowań mobilnych.

2.3.5 Zegar cezowy

Jak już wcześniej wspomniano wprowadzenie do użytku wzorców atomowych spowodowało powstanie nowej klasy dokładności pomiarów czasu i częstotliwości, a najważniejszymi z nich, odkąd definicja sekundy według SI została ustalona jako 9192631770 okresów rezonansowych atomu cezu Cs^{133} stały się cezowe zegary atomowe. Stało się tak między innymi z powodu wysokiej stabilności ich pracy (przy czym nie brano tutaj pod uwagę warunków środowiskowych, takich jak ruch, wibracje, silne pola magnetyczne oraz wpływu starzenia się które mogą mieć wpływ na warunki pracy oscylatora). Zasada działania rezonatora cezowego opiera się na zjawisku współdziałania między magnetycznym momentem elektronów powłoki elektronowej atomów cezu z polem magnetycznym, przy czym stopień tej interakcji zależny jest od stanu energetycznego atomów oraz pozycji elektronów w powłoce elektronowej. Jak wiadomo pole magnetyczne wytwarzane jest wokół elektronów wirujących na orbitach

wokół jądra atomu, jak również przez wewnętrzny moment magnetyczny elektronów. Na tej podstawie całkowite pole magnetyczne elektronu opisane może zostać wzorem (2.10) [20]:

$$\vec{H}_{el} = \vec{H}_{orbit} + \vec{H}_{spin} \tag{2.10}$$

Gdzie:

$\vec{H}_{el}$ – całkowite pole magnetyczne związane z elektronem

$\vec{H}_{orbit}$ – pole magnetyczne związane ruchem orbitalnym elektronu

$\vec{H}_{spin}$ – pole magnetyczne związane z wewnętrznym momentem magnetycznym elektronu

Dla atomów cezu znajdujących się w podstawowym stanie energetycznym $\vec{H}_{orbit} = 0$, zatem [20]:

$$\vec{H}_{el} = \vec{H}_{spin} \tag{2.11}$$

W zerowym zewnętrznym polu magnetycznym spiny elektronów atomów cezu mogą mieć dwa potencjalne kierunki. Jednak po umieszczeniu atomów w słabym polu magnetycznym poszczególne atomy cezu zaczynają się „przesuwać" na wykresie energetycznym tworząc ostatecznie klika poziomów energetycznych. Wiemy ponadto, że przyłożone pole magnetyczne ma wpływ na częstotliwość rezonansową przejścia między stanami energetycznymi atomów cezu co jest opisane równaniem (2.3.5.12) [20]:

$$f = f_0 + 427H^2 \tag{2.12}$$

Gdzie:

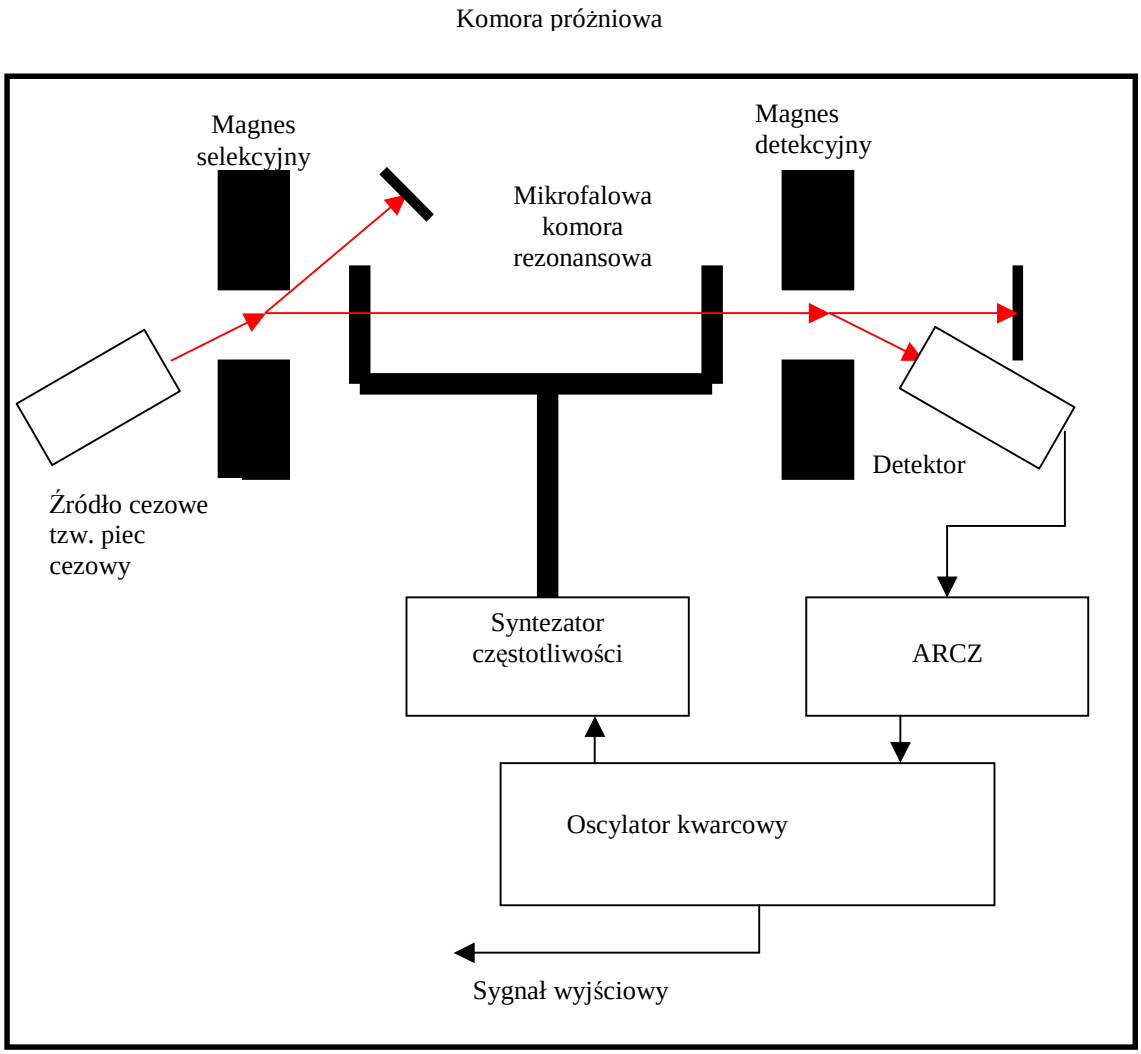
f_0 – częstotliwość rezonansowa przejścia między stanami energetycznym atomów cezu Ce^{133} w przestrzeni wolnej od pola magnetycznego.

f – częstotliwość rezonansowa przejścia między stanami energetycznym atomów cezu Ce^{133} w obecności pola magnetycznego.

W celu uzyskania jak najdokładniejszych pomiarów częstotliwości wzorcowych należy dążyć do minimalizacji wpływu pola magnetycznego, a zatem należy wykorzystać populację atomów cezu znajdującą się w stanie w którym działanie słabego pola magnetycznego oddziałuje z nią w jak najmniejszym stopniu.

Najmniej czułe na przyłożone pole magnetyczne są populacje stanów (F = 3; m_F = 0) oraz (F = 4; m_F = 0). Zatem pomiar częstotliwości przejść między stanami atomów cezu Ce^{133} będzie dokonywany dla stanów (F = 3; m_F = 0) ← →(F = 4; m_F = 0) w obecności słabego pola magnetycznego.

Budowa i działanie oscylatora cezowego



Rys. 2.5 Budowa oscylatora cezowego ze statycznym polem magnetycznym [16]

W oscylatorze cezowym jako źródło sygnału czasu wykorzystywana jest rezonansowa częstotliwość zmiany stanu atomów cezu Cs^{133}, względem której porównywane jest wyjście rezonatora kwarcowego sterowanego napięciem (OCXO). Wyjście OCXO powielane jest do częstotliwości rezonansowej zmiany stanu atomu cezu Cs^{133} i używane do sterowania wnęki mikrofalowej, gdzie wykrywane są zmiany stanu atomów za pomocą detektora optycznego, który dalej synchronizuje wejście

rezonatora kwarcowego sterowanego napięciem. Zegar cezowy składa się ze źródła cezowego, tzw. pieca cezowego, układu magnesów selekcyjnych, mikrofalowej komory rezonansowej, układu magnesów detekcyjnych, detektora oraz oscylatora kwarcowego kontrolowanego napięciem sprzęgniętego z komorą za pomocą układu automatycznej regulacji częstotliwości (w celu dostrojenia pracy oscylatora do częstotliwości rezonansowej atomów cezu) oraz syntezatora częstotliwości wraz z układem mikrofalowym. Blokowa budowa takiego zegara jest przedstawiona na rys. 2.5 [16].

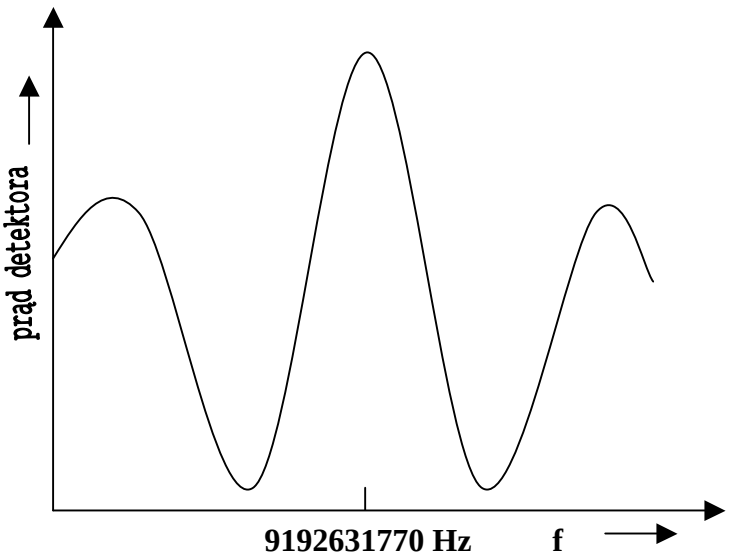
Komercyjnie dostępne oscylatory cezowe wykorzystują technikę strumienia cezowego. Dobroć komercyjnie dostępnych wzorców cezowych wynosi ok. 10^8.

Zasada działania

Aby dokonać pomiaru częstotliwość przejść między stanami energetycznymi (F = 3; m_F = 0) ← → (F = 4; m_F = 0) należy najpierw wyselekcjonować populację znajdującą się w powyższych stanach spośród populacji wszystkich stanów. W tym celu próbki atomów cezu Cs^{133} są rozgrzewane aż o osiągnięcia stanu gazowego, wewnątrz tzw. pieca cezowego. Następnie wiązka atomów w stanie gazowym z wysoką prędkością (powyżej 100 m/s) opuszcza piec cezowy i kierowana jest torem próżniowym w kierunku szczeliny znajdującej się pomiędzy dwoma stałymi magnesami selekcyjnymi (rys. 2.5) Magnesy stałe zastosowane w układzie służą jako bramka, która poprzez odchylanie strumienia atomów kieruje tylko te o określonej energii do mikrofalowej komory rezonansowej. Dobór rozmiarów, rozmieszczenia i rodzaju magnesów jest ściśle określony i związany z ich oddziaływaniem na poszczególne populacje atomów. W tym przypadku magnesy selekcyjne są tak dobrane aby skierować jedynie atomy znajdujące się w stanie (F = 3; m_F = 0) w kierunku komory rezonansowej. Typowa długość takiej komory wynosi ok. 50cm. Reszta atomów jest kierowana z dala od komory rezonansowej. Następnie w komorze rezonansowej atomy poddawane są działaniu promieniowania mikrofalowego o częstotliwości syntezowanej na podstawie częstotliwości drgań oscylatora kwarcowego. Tylko w przypadku, gdy częstotliwość promieniowania mikrofalowego odpowiada częstotliwości rezonansowej atomów cezu Cs^{133} atomy znajdujące się w komorze rezonansowej przejdą miedzy stanami energetycznymi (F = 3; m_F = 0) ← → (F = 4; m_F = 0).

Następnie atomy (w przypadku braku odstrojenia wszystkie znajdujące się w stanie (F = 4; m_F = 0)) są kierowane wzdłuż komory rezonansowej do szczeliny między

koleją para magnesów stałych – magnesów detekcyjnych. Magnesy te działają jako bramka, która poprzez odchylanie strumienia atomów kieruje tylko te atomy które zmieniły swój stan energetyczny podczas przechodzenia przez rezonansową komorę mikrofalową w kierunku detektora - a wiec przez szczelinę w kierunku detektora może przejść jedynie populacja znajdująca się w stanie ($F = 4$; $m_F = 0$) Atomy które nie zmieniły swojego stanu energetycznego ($F = 3$; $m_F = 0$) są odbijane z dala od detektora. Sygnał sprzężenia zwrotnego z detektora poprzez układ automatycznej regulacji częstotliwości dostraja oscylator kwarcowy w taki sposób aby maksymalizować ilość atomów trafiających do detektora – ilość atomów zmieniających stan energetyczny pod wpływem promieniowania komory. Im częstotliwość promieniowania mikrofalowego w komorze rezonansowej syntezowanego na podstawie sygnałów pochodzących z oscylatora kwarcowego jest bliższa częstotliwości rezonansowej atomów cezu, tym większa część atomów zmienia swój stan energetyczny i trafia do detektora.



Rys. 2.6 Charakterystyka rezonansowa zegara cezowego – wykres prądu detektora w funkcji odstrojenia od częstotliwości rezonansowej [20].

Wykres zależności prądu detektora w funkcji odstrojenia od częstotliwości rezonansowej przedstawiony jest na rys. 2.6. Jak można zauważyć wykres jest symetryczny względem osi przechodzącej przez środkową częstotliwości rezonansowej na osi częstotliwości, co dodatkowo utrudnia dostrojenie, ponieważ zmiana prądu detektora nie dostarcza informacji dotyczącej kierunku odstrojenia (czy częstotliwość mikrofal syntetyzowana na podstawie drgań oscylatora kwarcowego jest za niska, czy za wysoka).

Charakterystyka

Zegary cezowe charakteryzują się dobrą stabilnością długookresową wynoszącą $2 * 10^{-14}$, co oznacza że zegar może spóźniać się lub śpieszyć o 1 sekundę po ok. 1 400 000 lat. Już w latach 60-tych osiągnięto stopę sumaryczną wszystkich odstrojeń na poziomie 1×10^{-11}.

Tabela 2.2 Parametry wzorca cezowy 5071A oferowanego przez firmę SYMMETRICOM, INC [21]

Model:	• Wzorzec cezowy 5071A		
Wyjścia:	• Częstotliwościowe: 100kHz, 1MHz, 5MHz oraz 10MHz Format: sinusoida • Synchronizacyjne: PPS, (2V - 10V), TTL		
Stabilność:		Wydajność standardowa	Wysoka wydajność
	Czas (s)	Dewiacja Allana	Dewiacja Allana
	0.01	$<7.5*10^{-11}$	$<7.5*10^{-11}$
	0.1	$<1.2*10^{-11}$	$<1.2*10^{-11}$
	1	$<1.2*10^{-11}$	$<5.0*10^{-12}$
	10	$<8.5*10^{-12}$	$<3.5*10^{-12}$
	100	$<2.7*10^{-12}$	$<8.5*10^{-13}$
	1,000	$<8.5*10^{-13}$	$<2.7*10^{-13}$
	10,000	$<2.7*10^{-13}$	$<8.5*10^{-14}$
	5 dni	$<5.0*10^{-14}$	$<1.0*10^{-14}$
	30 dni	$<5.0*10^{-14}$	$<1.0*10^{-14}$
	Typowo	$<1.5*10^{-14}$	$<5.0*10^{-15}$
Wejścia:	1PPS, (2V - 10V)		
	Wydajność standardowa		Wysoka wydajność
Dokładność	$\pm 1.0*10^{-12}$		$\pm 2.0*10^{-13}$
Odstrojenie środowiskowe	$\pm 1.0*10^{-14}$		$\pm 8.0*10^{-14}$
Powtarzalność	$\pm 1.0E*10^{-13}$		$\pm 1.0*10^{-13}$

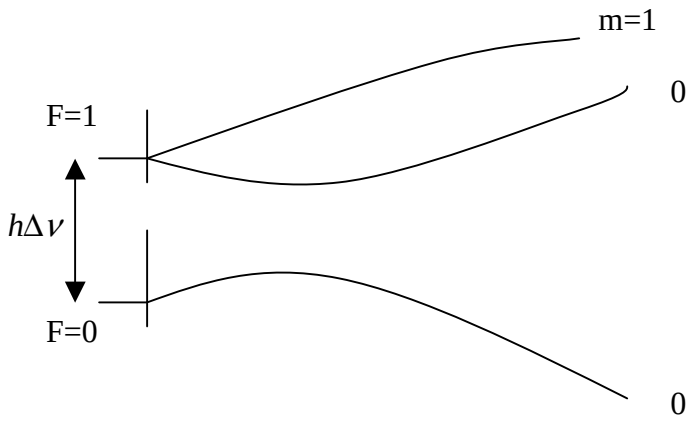
Sygnał pochodzący z wzorca (w tym przypadku cezowego) służy do stabilizacji częstotliwości stabilnego wzorca kwarcowego (np. o częstotliwości rezonansowej równej 5MHz) za pomocą syntezy częstotliwości. Wiele skal czasu jest opartych właśnie na zegarach cezowych. Najpopularniejszym i cechującym się dobrymi parametrami modelem zegara cezowego jest zegar 5071A produkowany aktualnie przez firmę SYMMETRICOM, INC. Oprócz doskonałych parametrów technicznych cechuje się on także rzadko spotykanymi (jak na zegar cezowy) małymi rozmiarami, jednak w przeciwieństwie do oscylatora rubidowego ze względu na wrażliwość na warunki środowiskowe (takie jak działanie pól magnetycznych, wibracje) nie jest to zegar mobilny.

Jak można zauważyć na podstawie danych z tabeli 2.2 stabilność krótkookresowa zegarów cezowych jest gorsza niż zegarów rubidowych. Jednak już w skali dni uwydatnia się przewaga zegarów cezowych, ponieważ ich stabilność staje się 1000 - krotnie wyższa w przypadku oscylatorów cezowych. Zegary cezowe są popularnie wykorzystywane jako źródła synchronizacji dla serwerów czasu NTP. Przykładem może być pobliski serwer vega.cbk.poznan.pl mieszczący się w Borowcu koło Poznania synchronizowany bezpośrednio z zegarem HP5071A (synchronizacja czasu do wyjścia PPS).

2.3.6 Pomiar czasu oparty o maser wodorowy

Maser wodorowy (mikrofalowy wzmacniacz ze stymulowaną emisją promieniowania - *ang. Microwave Amplification by Stimulated Emission of Radiation*) jest najbardziej złożonym, najdroższym komercyjnie dostępnym i najstabilniejszym (krótkookresowo) wzorcem częstotliwości. Działa w oparciu o częstotliwość rezonansową atomów wodoru wynoszącą 1 420 405 752 Hz. Przewyższa oscylator cezowy zarówno pod względem stabilności krótkookresowej (stabilność krótkookresowa aktywnego maser przewyższa stabilność najlepszych oscylatorów cezowych ponad 100 razy) jak i pod względem trwałości, niskich kosztów eksploatacji i wygody obsługi(w zegarach cezowych średnio co pięć lat istnieje konieczność wymiany bardzo kosztownej lampy, natomiast masery wodorowe nie posiadają żadnych wymienialnych elementów i mogą pracować bezawaryjnie przez ponad dziesięć lat). Tak jak w przypadku zegarów cezowych masery wodorowe opierają swoje działanie o zliczanie przejść pomiędzy dwoma stanami energetycznymi atomów wybranego

pierwiastka - w tym przypadku wodoru. Także w tym przypadku podstawowym problemem jest odizolowanie populacji atomów znajdujących się w ściśle wybranych stanach energetycznych spośród całej ich populacji. Atomy wodoru posiadają prostą budowę i przyjmują jedynie 2 poziomy energetyczne, co zostało przedstawione na rys. 2.7 [16], [22].



Rys.2.7 Wykres energetyczny atomów wodoru znajdujących się w stanie podstawowym jako funkcja wartości pola magnetycznego[22]

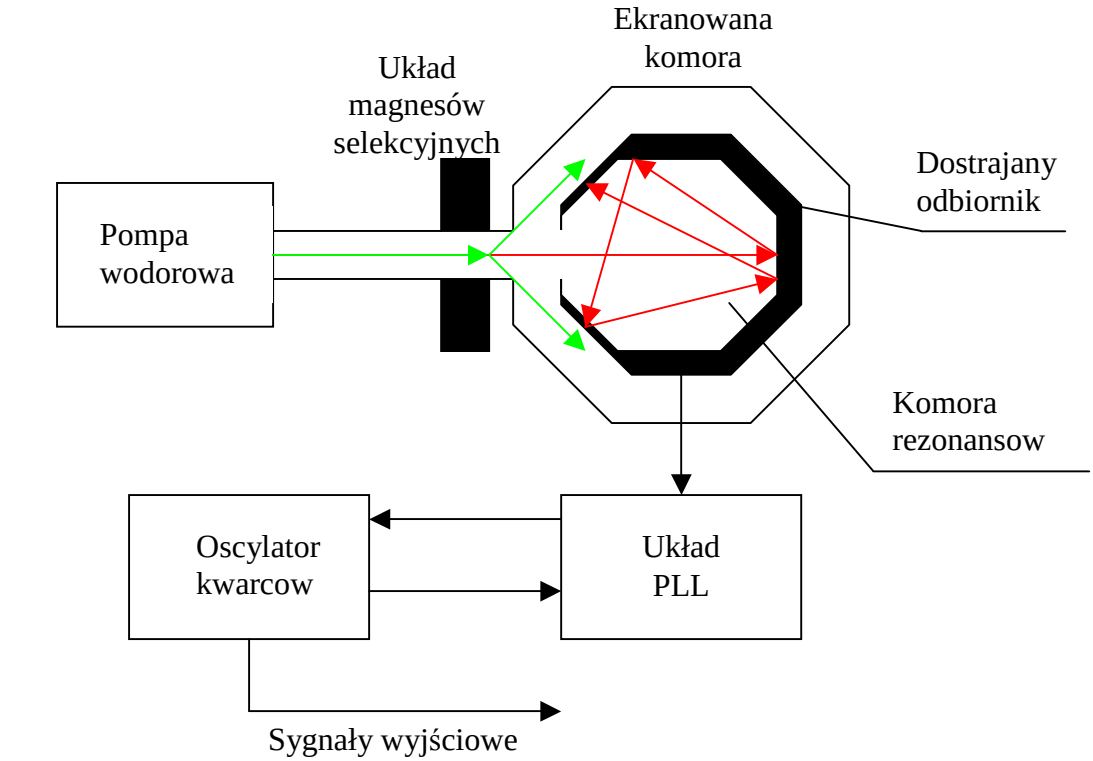
Stan energetyczny (F = 1; $m_F = 0$) widoczny na rys. 2.7 jest charakterystyczny dla zgodnych spinów elektronu i protonu natomiast stan energetyczny (F = 0; $m_F = 0$) odpowiada niezgodnym spinom. Do komory rezonansowej należy wprowadzić atomy w wyższym stanie energetycznym (F = 1; $m_F = 0$) które następnie w wyniku emisji (spontanicznej, lub wymuszonej) zmienią swój stan na (F = 0; $m_F = 0$).

Zatem dobór rozmiarów, rozmieszczenia i rodzaju magnesów jest ściśle określony i związany z ich oddziaływaniem na poszczególne populacje atomów. W tym przypadku magnesy selekcyjne są tak dobrane aby skierować jedynie atomy znajdujące się w stanie (F= 1; $m_F = 0$) do komory rezonansowej. Reszta atomów jest odbijana z dala od komory [16], [22].

Budowa masera wodorowego.

Maser wodorowy składa się ze źródła wodorowego, tzw. pompy wodorowej (wzbudzającej atomy wodoru), układu magnesów selekcyjnych, przechowującej komory rezonansowej (tzw. bulwy), dostrajanego odbiornika rezonansowego

otaczającego bulwę oraz oscylatora kwarcowego kontrolowanego napięciem sprzęgniętego z dostrajalnym odbiornikiem za pomocą pętli fazowej. Blokowa budowa takiego zegara jest przedstawiona na rys. 2.8 [16]; [22]



Gdzie:

Wiązka wzbudzonych atomów wodoru

Wiązka wzbudzonych atomów wodoru znajdujących się w wymaganym, ściśle określonym stanie energetycznym

Rys. 2.8 Schemat blokowy masera wodorowego [16]

Zasada działania

Praca maserów wodorowych opiera się na synchronizacji oscylatora kwarcowego do częstotliwości promieniowania mikrofalowego wytwarzanego przez pobudzone atomy wodoru.

Atomy wzbudzone w pompie wodorowej emitowane są poprzez bramę tworzoną przez 6 magnesów stałych pozwalających przejść bezpośrednio jedynie atomom znajdującym się w określonych stanach energetycznych. Atomy które przejdą przez bramę magnetyczną kierowane są do rezonansowej komory przechowującej (tzw. bulwy) otoczonej dostrajanym odbiornikiem rezonansowym. Atomy są utrzymywane w

bulwie nawet do kilku sekund. W tym czasie dochodzi do ok. 100000 zderzeń atomów z teflonowymi ścinkami bulwy. W stosunku do całkowitej liczby atomów jest to wartość bardzo mała, mająca nieznaczny wpływ na stabilność pracy. Początkowo po wprowadzeniu wzbudzonych atomów do komory rozpoczyna się proces emisji spontanicznej fotonów o częstotliwościach mikrofalowych. Owe fotony pobudzają pozostałe atomy znajdujące się w bulwie, a zatem rozpoczyna się proces emisji wymuszonej, a co za tym idzie następuje gwałtowny wzrost amplitudy samopodtrzymującego się promieniowania mikrofalowego. Dostrajalny odbiornik rezonansowy otaczający komorę posiada właściwości ekranujące odbijając fotony z powrotem do komory w celu podtrzymania emisji wymuszonej. Przy zachowaniu warunków pracy krzywa rezonansowa masera jest niezwykle stroma i możliwy jest samopodtrzymujący się rezonans wewnątrz komory, podtrzymywany fotonami nieustannie wypromieniowywanymi przez atomy wodoru nowo dostarczane ze źródła. Promieniowanie mikrofalowe wewnątrz komory jest ciągle badane przez odbiornik, a jego częstotliwość jest używana do synchronizacji oscylatora kwarcowego [16], [22].

Charakterystyka

Częstotliwość rezonansowa wodoru jest dużo niższa niż częstotliwość rezonansowa cezu co jest jego wadą. Z drugiej jednak strony szerokość rezonansowa w przypadku maserów wodorowych jest dużo niższa i wynosi zazwyczaj jedynie kilka Hz. W wyniku tego maser wodorowy cechuje się bardzo wysoką (najwyższą z komercyjnie dostępnych wzorców) dobrocią sięgającą 10^9. Dzięki temu stabilność krótkookresowa jest lepsza niż w przypadku cezu i wynosi ok. 1×10^{-15} po 1 dniu pracy. Biorąc jednak pod uwagę stabilność długookresowa lepsze są (w większości przypadków) wzorce cezowe. Szczególnie dobrymi właściwościami charakteryzuje się aktywny maser wodorowy. Parametry dwóch przykładowych modeli maserów wodorowych dostępnych w Polsce przedstawione są w tabeli 2.3 [19].

Tabela 2.3 Parametry maserów wodorowych oferowanego przez firmę Z.E.A.P. Meratronik S.A. [19].

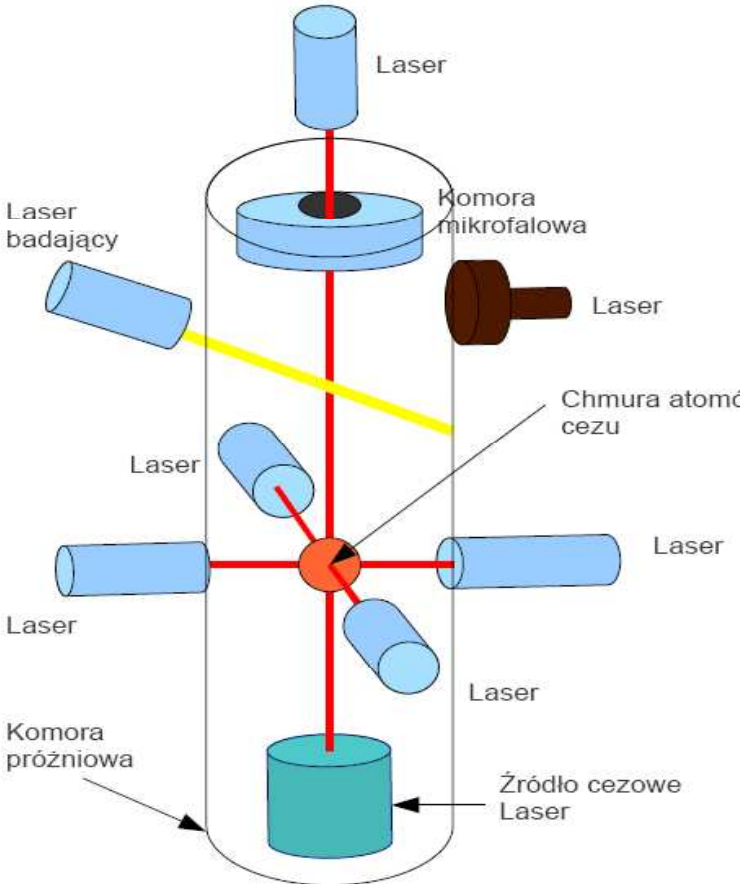
Model:	Aktywny maser wodorowy CH1-75	Pasywny maser wodorowy CH1-76
Wyjścia:	5MHz, PPS (opcje:10 MHz / 10.23 MHz / 2.048 MHz /Mb)	
Stabilność:	$5*10^{-15}$/100s $3*10^{-16}$/dzień	$1*10^{-14}$/dzień, $3*10^{-14}$ / h, $5*10^{-14}$/1000s

2.3.7 Fontanna cezowa

Fontanny cezowe są obecnie najbardziej zaawansowanymi zegarami atomowymi. Urządzenie tego typu (zegar NIST-F1) jest obecnie najważniejszym wzorcem czasu i częstotliwości dla USA. Opiera ono swoje działanie na koncepcji przedłużenia czasu obserwacji atomów w stosunku do zwykłych zegarów cezowych przedstawionej już w roku 1954. Ze względu na ograniczenia techniczne koncepcja doczekała się pierwszej udanej realizacji dopiero w roku 1989 [24]. Fontanna cezowa zawdzięcza swą nazwę podobieństwu w ruchu atomów wewnątrz komory zegara oraz ruchu wody wypływającej z dysz fontanny.

Budowa

Fontanna cezowa składa się ze źródła cezowego – tzw. pieca cezowego, układu sześciu laserów pompujących ustawionych tak, że ich wiązki skierowane są pod kątem prostym jedna do drugiej (po dwa lasery w każdym z trzech wymiarów) w kierunku centrum komory próżniowej, detektora optycznego oraz oscylatora kwarcowego kontrolowanego napięciem sprzęgniętego z detektorem optycznym za pomocą pętli fazowej. Całość zamknięta jest w komorze próżniowej. Blokowa budowa takiego urządzenia jest przedstawiona na rysunku 2.9 [16]; [24].



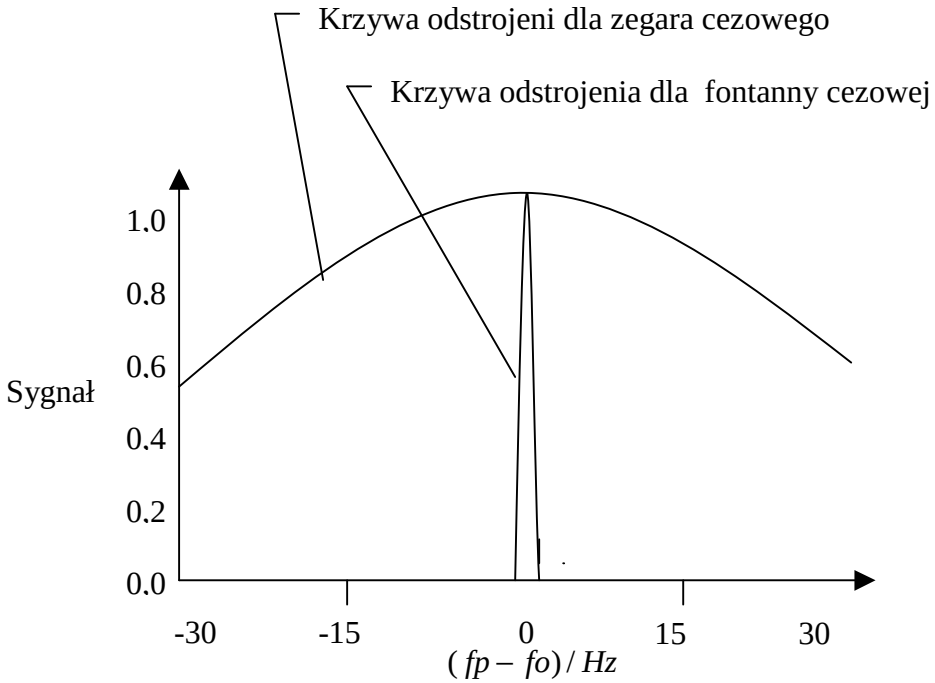
Rys. 2.9 Schemat poglądowy budowy fontanny cezowej [24]

Zasada działania

Zasada pomiaru sygnałów czasu wewnątrz fontanny cezowej jest tak sama jak w przypadku konwencjonalnego zegara cezowego. Główną różnicą między urządzeniami jest szereg rozwiązań zaimplementowanych w budowie fontanny cezowej (i nie występujących w konwencjonalnych zegarach cezowych) mających na celu wydłużenie obserwacji świecenia atomów cezu Ce^{133}. Podobnie jak w przypadku konwencjonalnego zegara cezowego tak i w fontannie cezowej wewnątrz źródła – pieca cezowego wytwarzana jest wiązka wysokoenergetycznych atomów cezu Ce^{133}, która następnie kierowana jest w stronę komory próżniowej. Także i w tym wypadku dla określenia odstrojenia od częstotliwości rezonansowej atomów cezu mierzony jest poziom świecenia ich próbki pod wpływem promieniowania mikrofalowego syntezowanego na podstawie częstotliwości rezonansowej zewnętrznego oscylatora kwarcowego.

Jak już wspomniano istota pracy fontanny cezowej odróżniająca ją od klasycznego zegara cezowego opiera się na znacznym wydłużeniu czasu obserwacji atomów cezu

wewnątrz komory rezonansowej. Po wprowadzeniu porcji atomów do komory próżniowej włączanych jest sześć laserów podczerwieni, których wiązki skierowane są pod kątem prostym jedna do drugiej (po dwa lasery w każdym z trzech wymiarów) w kierunku centrum komory. Wiązki światła laserów oddziałują ze znajdującymi się w centrum komory atomami powodując ich ochłodzenie poprzez zwalnianie ich ruchu (do temperatury tylko 1 μK, co redukuje ich prędkość termiczną do kilku centymetrów na sekundę) oraz ściskają chmurę atomów, koncentrując ją w kształt kuli.



Rys. 2.10 Zależność zmiany wartości sygnału pochodzącego z fotodetektora w funkcji odstrojenia od częstotliwości rezonansowej atomów cezu [25].

Pionowe promienie lasera unoszą chmurę atomów ku górze komory z prędkością ok. 4 m/s, po czym wszystkie lasery są wyłączne. Rozpędzona chmura atomów unosi się na wysokość ok. 1 m od podstawy komory po czym zatrzymuje się i zaczyna opadać jedynie pod wpływem własnej grawitacji. Także w tym wypadku kryterium oceny odstrojenia jest poziom emisji fotonów przez badane atomy. Cała podróż chmury w górę i w dół trwa aż ok. 1 s. Tak długi okres obserwacji atomów umożliwia wielokrotne dostrajanie częstotliwości promieniowania mikrofalowego (oraz oscylatora kwarcowego) do częstotliwości rezonansowej atomów cezu.

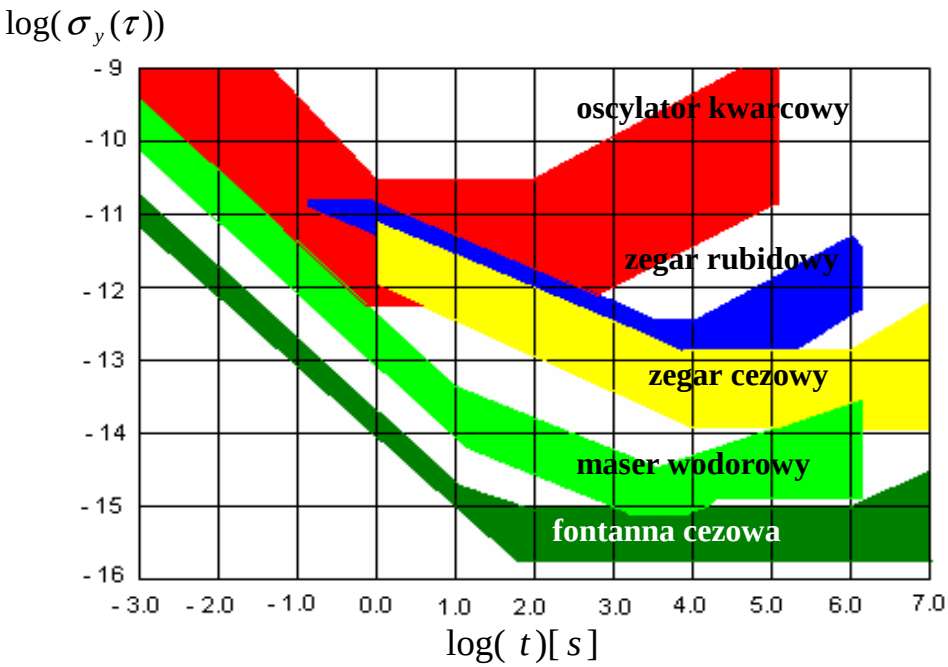
Zalety tej metody, to przede wszystkim:

- dłuższy czas oddziaływania,

- wolniejsze atomy
- mniejsze prędkości poprzeczne,
- jedna wnęka rezonansowa,

Dzięki temu pomimo takiej samej częstotliwości drgań jak w przypadku tradycyjnych zegarów cezowych szerokość rezonansowa jest znacznie niższa (odstrojenie poniżej 1 Hz, co można zaobserwować na rysunku 2.10). Owocuje to niezwykle wysoką dobrocią zegarów tego typu, wynoszącą ok. 10^{10} oraz stabilnością lepszą od 10^{-16}.

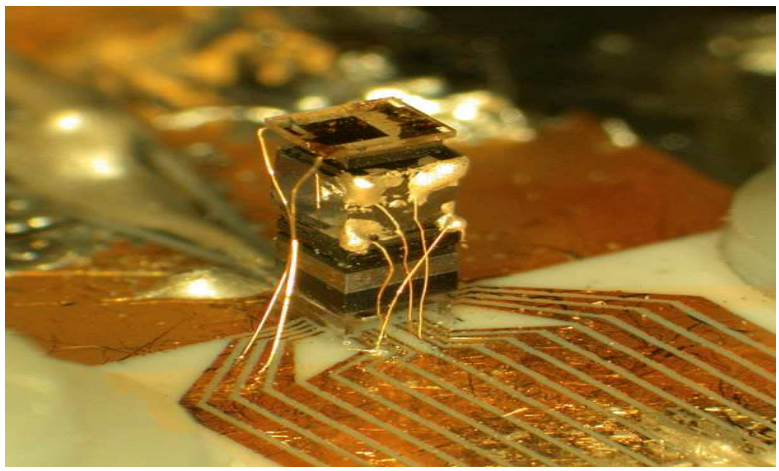
2.3 Porównanie metod pomiaru czasu



Rys 2.11 Wykres porównujący stabilności wzorców czasu i częstotliwości. [25]; [26]

Wykres porównujący stabilności wzorców czasu i częstotliwości przedstawiony jest na rys 2.11. Wśród wymienionych zegarów jedynie oscylator kwarcowy nie jest wzorcem atomowym. Jak można zauważyć wzorce atomowe są o wiele rzędów bardziej stabilne (długookresowo) od oscylatora kwarcowego (a tym bardziej od nie wymienionych tutaj zegarów mechanicznych). Na rys 2.11 dla poszczególnych wzorców czasu przyjęto dość szerokie przedziały stabilności (w szczególności dla oscylatora kwarcowego). Jest to spowodowane tym, że bazujące na danej technologii urządzenia mają różne wymagania na stabilność, co przeważnie przekłada się na cenę urządzenia. Pomimo względnie niskiej stabilności oscylatory kwarcowe znajdują i będą

znajdować najszersze zastosowanie spośród wszystkich źródeł czasu ze względu na prostotę budowy, niską cenę, małe rozmiary, oraz odporność na czynniki środowiskowe. Konkurencją dla oscylatorów kwarcowych stosowanych w małych przenośnych urządzeniach elektronicznych mogą stać się jedynie zegary atomowe w skali chipów - CSAD (Chip Scale Atomic Devices) [16].



Zdjęcie 2.1 Przykład zegara atomowego wykonanego w technologii CSAD [16].

Zegary atomowe oparte na architekturze CSAD budowane są w dobrze znanej technologii MEMS. Aktualnie budowane struktury (przykład na zdjęciu 2.1) są wielkość ziarna ryżu i mogą być zasilane z baterii AA (moc pobierana poniżej 75 mW). Stabilność zegarów budowanych w tej technologii jest porównywalna z zegarami atomowymi konwencjonalnych rozmiarów. W dalszym ciągu jednak wysoka cena zegarów CSAD jest barierą nie pozwalającą na stosowanie ich powszechnie w przenośnych urządzeniach radiokomunikacyjnych, oraz GPS. Wszystkie przedstawione wcześniej typy zegarów są obecnie używane i produkowane, ponieważ każdy z nich posiada odrębne właściwości pozwalające na jego zastosowanie w specyficznych warunkach. Przykładowo najmniej stabilny spośród wzorców atomowych (długookresowo) zegar rubidowy ze względu na małe rozmiary, wysoką niezawodność i odporność na warunki środowiskowe jest szeroko stosowany jako mobilny wzorzec częstotliwości. Masery wodorowe natomiast pomimo wysokiej ceny i małej stabilności długookresowej znajdują zastosowanie jako źródła odniesienia do pomiarów częstotliwości i są stosowane w układach wraz z wzorcami cezowymi w celu zwiększenia ich stabilność krótkookresowej.

3. Wybrane metody synchronizacji czasu

3.1 Synchronizacja za pomocą protokołu NTP

Network Time Protocol jest internetowym protokołem synchronizacji czasu opracowanym przez prof. Davida L. Millsa z Uniwersytetu w Delaware (USA) w 1994 roku. David L. Mills stanął na czele projektu NTP, od kilkunastu lat prowadzi wraz ze swoim zespołem badania nad NTP i nadzoruje kolejne implementacje protokołu. Specyfikacja protokołu jest umieszczona w ogólnodostępnym dokumencie RFC -1305. Na jego podstawie oparta jest treść niniejszego rozdziału. Synchronizacja czasu za pośrednictwem protokołu NTP zyskała ogromną popularność wśród użytkowników komputerów osobistych wraz z rozrostem sieci Internet, ze względu na jej wysoką dostępność. Zyskuje ona także coraz większą popularność w branży telekomunikacyjnej dzięki znakomitym parametrom. NTP pozwala na synchronizację czasu z bardzo dużą precyzją i jest rozwiązaniem bardzo stabilnym i bezpiecznym. Czas jest kalibrowany płynnie bez skoków i konieczności przestawiania zegara (wykorzystuje technikę przyspieszania i spowalniania). Już przy zastosowaniu standardowego sprzętu komputerowego klasy PC, precyzja może wynosić kilka milisekund. Obecnie protokół ma swoje implementacje dla większości współczesnych systemów operacyjnych i urządzeń sieciowych. Strukturę protokołu NTP oparto o założenie, że będzie on synchronizował czas dużej liczby niezależnie pracujących urządzeń względem niewielkiej liczby wzorcowych źródeł czasu. NTP nie obciąża sieci, nie wymaga wydajnych komputerów i jest w pełni zautomatyzowany (nie wymaga ingerencji użytkownika w przypadku wyjątków). Jest implementowany w środowisku TCP/IP. Ogólnie dostępny kod źródłowy napisany w języku C pozwala rozszerzać NTP dla kolejnych nowych systemów operacyjnych i powstających nowych urządzeń.

3.1.1 Architektura systemu

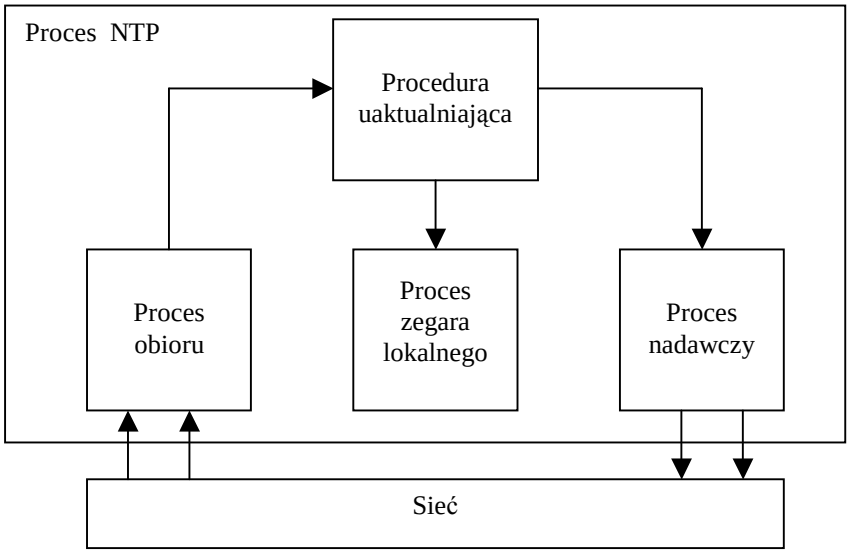
W celu ułatwienia w dalszej części rozdziału urządzenie lokalne określane będzie mianem hosta. Model systemu NTP składa się z ograniczonej liczby urządzeń pełniących role źródeł odniesienia czasu oraz urządzeń końcowych synchronizowanych względem nich [36].

W modelu rozróżniamy warstwy (Stratum). Urządzenia warstwy 1 to te, które są synchronizowane bezpośrednio do wzorców czasu (najczęściej cezowych zegarów atomowych, maserów wodorowych, zegarów wykorzystujących generatory rubidowe lub satelitarnych wzorców radiowych w systemie GPS) Urządzenia te podpięte są do szkieletu sieci Internet i pełnią rolę serwerów czasu. Kolejna warstwa (Stratum 2) składa się z urządzeń synchronizowanych względem serwerów warstwy 1. Podobnie synchronizacja przebiega dla kolejnych warstw. Relacje pomiędzy urządzeniami NTP mają charakter hierarchiczny, co oznacza, że zawsze urządzenia warstwy niższej są synchronizowany względem tych z wyższej warstwy [36]. Zatem komputery warstwy Stratum N mogą być źródłami czasu dla warstwy Stratum N+1, ale nie na odwrót. Serwery warstwy Stratum N są jednocześnie klientami komputerów warstwy Stratum N-1 Często serwery warstw niskich stanowią bramy sieci lokalnych w których komputery synchronizowane są względem nich.

Do wymiany danych stosowane są 72 bajtowe pakiety UDP. Dla celów transmisji protokołu NTP zarezerwowano port 123. Przesyłane dane pozwalają każdemu z klientów obliczyć własne opóźnienie względem źródła pierwotnego. Na tej podstawie każdy z klientów sam kalibruje wskazania własnego, lokalnego zegara systemowego. Kalibracja polega na płynnym przyspieszaniu lub spowalnianiu pracy lokalnego zegara programowego w celu asymptotycznego zbliżeniu się do wskazań wzorcowego źródła czasu. Przy dużych różnicach czasu, większych od 128ms, NTP stosuje jednorazową synchronizację skokową.

NTP jest tak zaprojektowany żeby produkować 3 wyniki: przesunięcie zegara (*clock offset*) opóźnienie transmisji (*roundtrip delay*) oraz dyspersje (*dispersion*), które są względne do zegara odniesienia. Przesunięcie zegara reprezentuje różnice wartości czasu wskazywanego przez zegar lokalny względem zegara odniesienia. Opóźnienie transmisji wyraża zdolność do wysłania do zegara odniesienia i odebrania wiadomości w określonym z góry czasie. Dyspersja reprezentuje maksymalny błąd zegara lokalnego względem zegara odniesienia [36].

W przypadku NTP nie są potrzebne dodatkowe mechanizmy kontroli i korekcji błędów transmisji. Integralność danych jest zapewniona przez sumę kontrolną IP oraz UDP. Nie stosuje się kontroli przepływu ani retransmisji. Detekcja duplikatów jest zawarta w samym algorytmie NTP.



Rys. 3.1 Implementacja modelu NTP [36]

Na rys. 3.1 przedstawiono uproszczoną implementacje modelu hosta na którym uruchomiony jest proces synchronizacji NTP. Rysunek przedstawia trzy procesy współdzielące podzieloną na części bazę danych. Każda część bazy danych jest przydzielona do innego urządzenia zdalnego. Proces nadawczy jest wywoływany przez oddzielne (dla każdego zdalnego hosta) czasomierze, zapisuje wymagane informacje w bazie danych, oraz wysyła wiadomości NTP do zdalnych urządzeń. Każda wysłana wiadomość zawiera lokalne stemple czasowe, oraz informacje konieczne do ustanowienia hierarchicznych połączeń w sieci. Częstość wysyłania wiadomości jest związana z wymaganiami dokładnościowymi zegara lokalnego [36].

Proces odbiorczy odpowiada za odbieranie wiadomości NTP oraz informacji z bezpośrednio dołączonego zegara wzorcowego. Podczas odbioru wiadomości obliczany jest offset pomiędzy lokalnym i zdalnym zegarem, a następnie zapisywany do bazy danych wraz z innymi informacjami niezbędnymi do korekcji błędów i wyboru źródła synchronizacji.

Procedura uaktualniająca wywoływana jest przez proces odbiorczy celem dalszych obliczeń na otrzymanych danych oraz celem wyboru źródła synchronizacji. Proces zegara lokalnego wywoływany jest przez procedurę uaktualniającą. Jest

odpowiedzialny za dostrojenie fazy oraz częstotliwości zegara lokalnego. W praktycznych implementacjach programów służących do synchronizacji poprzez NTP znika często rozgraniczenie między procesami modelu z rys. 3.1

3.1.2 Tryby pracy

W zależności od charakteru pracy urządzeń implementujących NTP mogą one pracować w kilku trybach, tak aby jak najwydajniej wykorzystać ich możliwości obliczeniowe oraz aby jak najmniej obciążać sieć [36]. Skojarzenie NTP jest zawsze ustanawiane między parami urządzeń (we wszystkich trybach oprócz trybu rozsiewczego). Skojarzenie może pracować w pięciu trybach przy czym aktualny tryb jest ustanawiany przez zmienną *mode* zawartą w wiadomości NTP. Zmienna ta jest ustawiana przez klienta w danym skojarzeniu. Dostępne tryby to: symetryczny aktywny, symetryczny pasywny, klient, serwer oraz rozsiewczy. Zdefiniowane są w następujący sposób:

- Tryb symetryczny aktywny – w tym trybie klient wysyła okresowo wiadomości do serwera względem którego się synchronizuje, niezależnie od jego dostępności lub liczby stratum. Pracując w tym trybie host wysyła wiadomości, rozgłaszając jego chęci do synchronizacji względem skojarzonego z nim serwera, oraz do synchronizacji skojarzonych z nim klientów. Tryb symetryczny aktywny jest przeznaczony do użycia dla serwerów występujących bliżej węzłów końcowych (oznaczonych wysoką liczbą Stratum). Zwykle wysokiej jakości synchronizacja może być dokonywana jeśli host jest przyłączony do dwóch źródeł, nawet jeśli połączenie jest słabej jakości i często zwracane są wiadomości błędu
- Tryb symetryczny pasywny – ten typ skojarzenia jest zazwyczaj tworzony podczas nadejścia wiadomości od urządzenia pracującego w trybie symetrycznym aktywnym i jest utrzymywany tak długo jak długo zdalne urządzenie jest dostępne, przy dodatkowym założeniu, że zdalne urządzenie pracuje w warstwie stratum niższej albo równej. W przeciwnym wypadku skojarzenie jest zrywane, jednakże po wysłaniu przynajmniej jednej wiadomości. Podczas działania w tym trybie host zawiadamia o chęci synchronizowania się ze źródłem oraz do synchronizacji skojarzonych z nim odbiorców wiadomości. W trybach symetrycznych znika rozgraniczenie pomiędzy serwerem a klientem. Tryb symetryczny pasywny jest przeznaczony dla serwerów działających w wyższych warstwach drzewa

synchronizacji (posiadają małą liczbę Stratum), posiadających stosunkowo dużą liczbę klientów do zsynchronizowania.

- Tryb klienta – host działający w tym trybie wysyła okresowo wiadomości niezależnie od dostępności albo stratum urządzenia względem którego będzie się synchronizować. Działając w tym trybie host (zazwyczaj stacja robocza w sieci LAN) ogłasza jego gotowość by być synchronizowanym względem serwera ale nie synchronizuje dalej żadnych urządzeń. Klient sporadyczne wysyła wiadomości do serwera. Serwer odpowiada wiadomością w której po prostu wymienia adres oraz port, wypełnia wymagane informacje i odsyła wiadomość do klienta. Serwery nie zachowują żadnych informacji na temat okresów między zgłoszeniami klienta, zatem nie kontrolują okresów synchronizacji. Ten tryb może zostać sprowadzony do prostego wywoływania zdalnych instrukcji bez utraty dokładności albo elastyczności synchronizacji.
- Serwer – ten tryb stowarzyszenia jest zazwyczaj utworzony po nadejściu wiadomości od klienta żądającego synchronizacji i istnieje tylko do momentu odpowiedzi na żądanie klienta po którym skojarzenie jest zrywane. Pracujący w tym trybie host (najczęściej serwer czasu sieci LAN pracujący w medium o wysokiej przepustowości) jest gotów do synchronizacji wszystkich urządzeń w danej sieci ale nie informuje o gotowoości do synchronizacji względem innego urządzenia.
- Tryb rozsiewczy – host działający w tym trybie okresowo rozsiewa wiadomości niezależnie od dostępności stanu lub stratum urządzeń do których one są adresowane. Pracujący w tym trybie host (najczęściej serwer czasu sieci LAN pracujący w medium o wysokiej przepustowości) ogłasza jego gotowość do synchronizacji wszystkich urządzeń w danej sieci ale nie informuje o gotowości do synchronizacji względem innego urządzenia. Różnica między tym trybem a trybem serwera polega na tym, że host pracujący w trybie serwera wysyła widomość do jednego urządzenia (po wcześniejszym otrzymaniu żądania synchronizacji) a w trybie rozsiewczym są one wysyłane do wielu urządzeń jednocześnie. Tryb rozsiewczy jest przeznaczony dla serwerów pracujących blisko węzłów końcowych drzewa synchronizacji posiadających dużą liczbę klientów do zsynchronizowania.

Najczęściej skojarzenie tworzone jest tak że jedno urządzenie działa w trybie aktywnym (symetryczny aktywny, klient albo rozsiewczy) a pozostałe pracują trybie pasywnym (serwer albo symetryczny pasywny), jednakże oba urządzenia mogą być

skonfigurowane do pracy w trybie symetrycznym aktywnym. Jeśli oba urządzenia wymieniające między sobą informacje działają w tym samym rodzaju trybów (aktywnym innym niż symetryczny albo pasywnym) dochodzi do błędu skojarzenia. W takim przypadku urządzenie ignoruje wiadomości od pozostałych urządzeń.

W najbardziej rozpowszechnionym modelu klient – serwer, klient wysyła wiadomość NTP do jednego albo więcej serwerów a następnie przetwarza otrzymaną odpowiedz. Serwer wymienia adres i port, nadpisuje ważne pola wiadomości, przekalkulowuje sumę kontrolną i bezzwłocznie zwraca wiadomość do nadawcy. Informacje zawarte w wiadomości NTP pozwalają klientowi określić czas serwera w odniesieniu do czasu lokalnego i zgodnie z nimi dostroić zegar lokalny.

Jak wspomniano odległość serwera od pierwotnego źródła czasu jest określona przez liczbę tzw. Stratum. Gdy Stratum równe jest 1 to określa główne serwery, a każda kolejna wyższa liczba określa serwery wtórne. Przy zastosowaniu zegarów synchronizowanych radiowo może zostać osiągnięta dokładność synchronizacji względem głównego źródła czasu równa kilku milisekundom. Spadek dokładności wraz ze wzrostem liczby Stratum jest związany ze stanem ścieżek w sieci oraz stabilnością zegarów lokalnych serwerów. Dlatego aby zapobiec poważnym błędom stosuje się estymację błędów w każdej specyficznej konfiguracji.

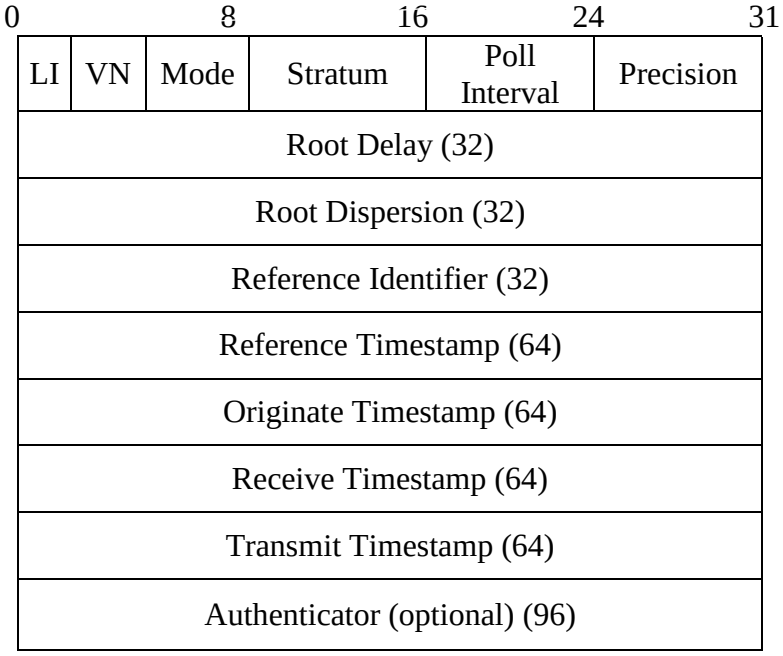
Aby osiągnąć wysoką dokładność niezbędna jest implementacja mechanizmów NTP na bardzo niskim poziomie systemu operacyjnego. Mechanizmy takie zostały zaimplementowane w jądrach rodziny systemów Linux oraz w systemie Unix. Brak jednak tychże mechanizmów w starszych systemach z rodziny *Windows*.

Aby znaleźć jak najkrótsze ścieżki wymiany informacji NTP używa algorytmu Bellmana – Forda. W tym algorytmie jako metryka dystansu używana jest suma liczby stratum oraz dystansu synchronizacji. Wartość dystansu synchronizacji zawiera sumę dyspersji oraz ½ absolutnego opóźnienia. Dzięki temu ścieżka synchronizacji zawsze wybiera najmniejszą ilość serwerów względem serwera pierwotnego przy minimalnym współczynniku błędów. W rezultacie podsieć zawsze rekonfiguruje się automatycznie w strukturze hierarchicznej urządzenie nadrzędne / urządzenie podrzędne, nawet jeśli jeden lub więcej głównych lub wtórnych serwerów zawiedzie. W przypadku wielu serwerów głównych algorytm drzewa rozpinającego zwykle wybierze serwer o minimalnym dystansie synchronizacyjnym.

3.1.3 Format danych

Wszystkie dane w przypadku NTP są wyrażone jako naturalne stałoprzecinkowe liczby. Głównym produktem NTP jest stempel czasowy (timestamp), którego specjalny format został ustanowiony przez projektantów standardu. Stempel czasu NTP jest reprezentowany jako 64-bitowa dodatnia stałoprzecinkowa liczba sekund względem godziny 00:00 pierwszego stycznia 1900 roku [36]. Pierwsze 32 bity reprezentują całkowita liczbę sekund natomiast kolejne 32 bity przedstawiają ułamki sekund. Precyzja tej reprezentacji czasu wynosi około 200 ps. co powinno być wystarczające nawet dla najbardziej wygórowanych wymagań. Stempel czasowy jest wyznaczony przez kopiowanie obecnej wartości czasu wskazywanej przez zegar lokalny do odpowiedniego pola wiadomości. Jak już wspomniano dla osiągnięcia jak najwyższej dokładności ważne jest aby obliczenia były wykonywane jak najbliżej sprzętu (wykonywane na jak najniższym poziomie systemu operacyjnego). Stempel czasowy nie zawsze musi zawierać czas. W niektórych przypadkach stempel czasowy może być nieosiągalny np. wtedy kiedy host jest restartowany albo synchronizacja przebiega pierwszy raz. W takim wypadku, 64 bitowe pole stempla jest wypełnione zerami co oznacza, że jest on nie zdefiniowany.

Format wiadomości NTP



Rys. 3.2 Format wiadomości protokołu NTP [36]

Na rysunku 3.2 przedstawiony jest format wiadomości protokołu NTP. Wiadomość taka jest przesyłana w jednym pakiecie UDP i rozpoczyna się zaraz za jego nagłówkiem:

- Pole LI (Leap Indicator) jest to dwubitowy kod ostrzegawczy dotyczący synchronizacji. Wartości kodu zdefiniowane są w następujący sposób:

- 00 – brak ostrzeżeń
- 01 – ostatnia minuta doby ma 61 sekund
- 10 – ostatnia minuta doby ma 59 sekund
- 11 – alarm warunkowy (zegar nie zsynchronizowany)

- Pole VR (Version Number) jest to trzybitowa liczba całkowita oznaczająca wersję używanego protokołu NTP.
- Pole Mode jest to trzybitowa liczba całkowita wyznaczająca tryb pracy. Jej wartości są zdefiniowane następująco:

- 0 – zarezerwowany dla przyszłych rozwiązań
- 1 – tryb symetryczny aktywny
- 2 – tryb symetryczny pasywny
- 3 – tryb klienta
- 4 – tryb serwera
- 5 – tryb rozsiewczy (broadcast)
- 6 – zarezerwowany dla wiadomości kontrolnych protokołu NTP
- 7 – zarezerwowany dla użytku prywatnego

- Stratum – jest to ośmiobitowa liczba całkowita dodatnia wyznaczająca pozycję zegara w hierarchii synchronizacji drzewa NTP w następujący sposób:

- 0 – niewyspecyfikowany
- 1 – nadrzędne źródło odniesienia
- 2 - 255 – podrzędne źródło odniesienia lub klient

powyższa wartość może zawierać się w zakresie od 0 do Max Stratum włącznie.

- Poll Interval (okres synchronizacji) jest to ośmiobitowa liczba całkowita oznaczająca maksymalny interwał pomiędzy prawidłowymi synchronizacjami, wyrażona w sekundach. Wartość znajdująca się w tym polu może zawierać się pomiędzy stałymi Min Poll Interval i Max Pool Interval włącznie.
- Precision – ośmiobitowa liczba całkowita oznaczająca precyzję zegara lokalnego wyrażona w ilości sekund do najbliższej potęgi liczby 2.

- Root Delay jest to trzydziestodwubitowa liczba całkowita oznaczająca maksymalne opóźnienie synchronizacji względem nadrzędnego zegara w danym drzewie hierarchicznym NTP. Pierwsze 16 bitów oznacza liczbę sekund, kolejne 16 bitów ułamek sekundy. Należy zauważyć, że liczba ta może posiadać dodatnią albo ujemną wartość w zależności od precyzji zegara i przesunięcia.
- Root Dispersion – trzydziestodwubitowa liczba naturalna oznaczająca maksymalny błąd względny do nadrzędnego zegara odniesienia w danym drzewie hierarchicznym NTP. Pierwsze 16 bitów oznacza liczbę sekund, kolejne 16 bitów ułamek sekundy.
- Reference Clock Identifier jest to trzydziestodwubitowy kod identyfikujący poszczególne zegary odniesienia w postaci ciągu ASCII. W przypadku stratum równego 2 lub większego jest to czterooktetowy adres internetowy nadrzędnego źródła odniesienia.
- Reference Timestamp jest to sześćdziesięcioczterobitowa liczba. Pierwsze 32 bity oznaczają liczbe sekund aktualnego czasu wyrażonego w formacie stempla czasowego NTP. Kolejne 32 bity to ułamek sekundy. Jest to stempel czasowy zawierający czas ostatniej poprawnej synchronizacji zegara lokalnego.
- Originate Timestamp – czas lokalny wyrażony w formacie stempla czasowego NTP, w którym żądanie synchronizacji opuściło klienta i zostało wysłane w stronę źródła odniesienia.
- Receive Timestamp jest to czas lokalny wyrażony w formacie stempla czasowego NTP, w którym żądanie synchronizacji dotarło do źródła synchronizacji.
- Transmit Timestamp jest to czas lokalny wyrażony w formacie stempla czasowego NTP, w którym odpowiedź opuściła źródło odniesienia i została wysłana w stroną klienta który wysłał żądanie synchronizacji.
- Authenticator jest polem opcjonalnym. Występuje w przypadku zaimplementowania mechanizmów autentyfikacji NTP.

  Przedstawione pola wiadomości NTP (rys. 3.2) reprezentują jednocześnie zestaw zmiennych (danych) wspólnych dla wszystkich procesów modelu NTP.

  Standard Network Time Protocol w wersji trzeciej definiuje dwie zmienne systemowe. Muszą one występować w każdym oprogramowaniu zgodnym z NTPv3. Zmienne te są zdefiniowane w następujący sposób:

- Zmienna Local Clock – jest to aktualny czas lokalny w formacie stempla czasowego NTP. Lokalny czas jest odczytywany z zegara sprzętowego danego komputera.

- Clock Source – jest to wskaźnik identyfikujący aktualne źródło synchronizacji, jeśli jego wartość to NULL oznacza to, że nie ma obecnie żadnych dostępnych źródeł synchronizacji.

Oprócz wspólnych zmiennych, oraz zmiennych systemowych standard definiuje ich następujące zestawy: Zmienne dotyczące zdalnego hosta, Zmienne dotyczące pakietów, Zmienne procedury filtracji, Zmienne dotyczące autentyfikacji.

Oprócz zmiennych standard definiuje szereg parametrów, które muszą być zdefiniowane dla każdego hosta. Ich wartości są z góry ustalone przez standard i niezmienne niezależnie od trybu pracy, lub systemu operacyjnego.

Tabela 3.1 Parametry systemu [36]

Nazwa parametru	**Przypisana wartość**	**Opis**
Version Number	3	Wersja protokołu NTP. W omawianym przypadku wersja 3.
NTP Port	123	Port używany przez protokół NTP.
Max Stratum	15	Maksymalna dopuszczalna wartość liczby stratum. Jest interpretowana jako nieosiągalność podsieci synchronizacji.
Max Clock Age	86400 s	Maksymalny interwał między uaktualnieniami w którym utrzymana zostanie ważność zegara odniesienia.
Max Skew	1 s	Maksymalne odchylenie zegara względem źródła w okresie synchronizacji.
Max Distance	1 s	Maksymalna odległość synchronizacji względem źródła odniesienia.
Min Polling Interval	64	Minimalna wartość okresu pomiędzy momentami synchronizacji wyrażona w sekundach.
Max Polling Interval	1024	Maksymalna wartość okresu pomiędzy momentami synchronizacji wyrażona w sekundach.
Min Select Clocks	1	Jeśli używany jest algorytm selekcji źródła, jest to minimalna ilość źródeł synchronizacji wybranych do selekcji.
Max Select Clocks	10	Jeśli używany jest algorytm selekcji źródła, jest to maksymalna ilość źródeł synchronizacji wybranych do selekcji.

Tabela 3.1 Parametry systemu [36]

Min Dispersion	0.01s	Jeśli używany jest algorytm filtracyjny, jest to minimalny przyrost dyspersji dla każdej warstwy stratum.
Max Dispersion	16 s	Jeśli używany jest algorytm filtracyjny, jest to maksymalny przyrost dyspersji dla każdej warstwy stratum.
Reach. Reg Size	8 b	Jest to rozmiar rejestru osiągalności zdalnego hosta.
Filter Size	8 b	Jeśli używany jest algorytm filtracyjny, jest to rozmiar filtracyjnego rejestru przesuwnego.
Filter Weight	1/2	Jeśli używany jest algorytm filtracyjny, jest to waga używana do obliczenia dyspersji filtru.
Select Weight	3/4	Jeśli używany jest algorytm selekcyjny, jest to waga używana do obliczenia dyspersji selekcji.

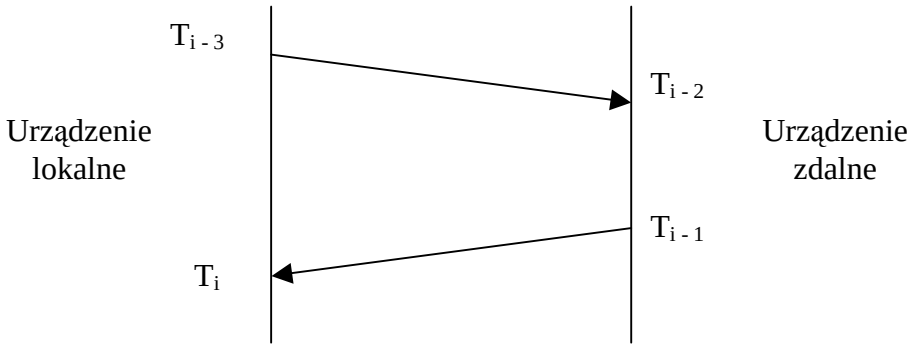
3.1.4 Procedury NTP

W algorytmie NTP zdefiniowano procedury. Według standardu poszczególne procedury odpowiadają za ściśle odkreślone zadania [36]. Nie jest jednak wymagane aby wszystkie procedury występowały w takiej postaci w rzeczywistych implementacjach programów obsługujących synchronizację NTP zgodnych ze standardem. Najczęściej procedury w programach przeplatają się zakresami zadań w stosunku do tych zdefiniowanych przez standard.

- Procedura transmisji – jest wykonywana kiedy czasomierz hosta zdalnego jest dekrementowany do zera dla wszystkich trybów pracy oprócz trybu klienta skojarzonego z serwerem rozgłaszającym. W skojarzeniu klient - serwera wiadomości z serwera są wysyłane tylko w odpowiedzi na otrzymaną wiadomość. Procedura transmisji może być także wywołana przez procedurę odbiorczą w przypadku, gdy odebrana wiadomość NTP jest błędna. Procedura transmisji odpowiada za wypełnienie wszystkich pól wiadomości (łącznie ze stemplami czasowymi). Jeśli zaimplementowana jest procedura autentyfikacji to jest ona wywoływana wewnątrz procedury transmisji. Wywoływane są także procedury selekcji i filtrowania zegarów.
- Procedura odbiorcza – jest wywoływana podczas nadejścia wiadomości NTP. Odpowiada za sprawdzenie poprawności wiadomości, interpretuje zapisane tryby i

wywołuje inne procedury w celu filtrowania danych i wyboru źródła synchronizacji. W razie niezgodności w nagłówku wiadomości (np. inna wersja protokołu niż wykorzystywana przez urządzenie lokalne) wiadomość jest odrzucana. Sprawdzane są: adres IP, suma kontrolna i port w celu dopasowania do konkretnego hosta zdalnego. Jeśli odebrany pakiet okazuje się przydatny, jest przeznaczony do dalszego przetwarzania gdzie poddany jest ośmiu testom.

- Procedura pakietu – sprawdza ważność wiadomości, oblicza opóźnienie, przesunięcie oraz wywołuje inne procedury celem filtracji danych i wyboru źródła synchronizacji. Dane poddawane są trzem testom. Test 1 sprawdza stempel czasowy nadawczy zawarty w wiadomości i porównuje go z ostatnio otrzymanym stemplem z tego samego źródła. Jeśli stemple czasowe są równe oznacza to, że otrzymana wiadomość jest duplikatem. Test 2 sprawdza czas nadania wiadomości przez klienta zawarty w stemplu (orginate timestamp) i porównuje z ostatnio zapisanym stemplem wysłanym z tego samego źródła. Jeśli stemple się różną oznacza to że wiadomość może być otrzymana poza kolejnością lub fałszywa. Test 3 sprawdza stemple czasowe nadania wiadomości i czas odbioru wiadomości. Jeśli są równe zero oznacza to że skojarzenie jest niezsynchronizowane albo stracono połączenie w obu kierunkach.



Rys. 3.3 wykres wymiany wiadomości NTP [36]

Procedura pakietu odpowiada za obliczanie czasu podróży wiadomości i przesunięcia zegara względem zegara zdalnego. Na rysunku 3.3 przedstawiony jest wykres czasowy wymiany wiadomości. Występujące na nim czasy T_{i-3}, T_{i-2}, T_{i-1}, T_i, oznaczają to samo co zawarte w wiadomości stemple czasowe, odpowiednio: czas transmisji przez klienta (orginate timestamp), czas odbioru przez urządzenie zdalne (receive timestamp), czas transmisji przez urządzenie zdalne (transmit

timestamp), oraz czas odbioru przez urządzenie lokalne. Na ich podstawie obliczane jest opóźnienie transmisji δ, przesunięcie zegara θ, oraz dyspersję ε w następujący sposób [36]:

$$\delta = (T_i - T_{i-3}) - (T_{i-1} - T_{i-2}) \tag{3.1}$$

$$\theta = \frac{(T_{i-2} - T_{i-3}) + (T_{i-1} - T_i)}{2} \tag{3.2}$$

$$\varepsilon = (1 << (\text{precyzja zegara lokalnego})) + \varphi\,(T_i - T_{i-3}) \tag{3.3}$$

Gdzie:

φ = Max Skew / Max Clock Age

Otrzymane w ten sposób dane są dalej potrzebne w przypadku procedury filtracji przedtem będą jednak poddane kolejnym 5 testom. Test 4 sprawdza czy opóźnienie transmisji i dyspersja są mniejsze od z góry założonego parametru Max Dispersion. Test 5 sprawdza czy urządzenie zdalne wymaga autentyfikacji. Test 6 sprawdza wartość pola ostrzeżeń (leap). Test 7 sprawdza czy urządzenie zdalne znajduje się wyżej w hierarchii synchronizacji (mniejsza wartość liczby stratum). Test 8 sprawdza czy opóźnienie transmisji i dyspersja są mniejsze od z góry założonego parametru Max Dispersion.

Jeżeli spełnione są testy od 1 – 4 oznacza to, że odebrany pakiet zawiera poprawne dane. Jeśli spełnione są testy 5 – 8 oznacza to, że odebrany pakiet posiada poprawny nagłówek. Pakiety które spełnią wszystkie testy mogą zostać użyte do synchronizacji. Test 1 i 2 nie są używane w trybie rozsiewczym.

- Procedura uaktualnienia zegara – jest wywoływana z procedury odbiorczej oraz procedury pakietu i jeśli otrzymano poprawne wartości przesunięcia zegara, opóźnienia i dyspersji wywołuje ona procedurę uaktualnienia zegara oraz procedurę lokalnego zegara. Następnie wymienia wartości przypisanych do niej zmiennych i na podstawie otrzymanych danych uaktualnia zegar.

- Procedura zegara pierwotnego – jest uruchamiana jedynie w urządzeniach pierwotnych (przyłączonych bezpośrednio lub radiowo do zegara atomowego). Odpowiada ona za pobieranie danych synchronizacyjnych z zegara atomowego w okresach 60 sek. Interpretuje stemple czasowe zegara odniesienia w postaci ciągu ASCII a następnie konwertuje je na postać stempli czasowych NTP celem uaktualnienia zegarów klienckich.
- Procedura inicjalizacyjna – wywoływana po restarcie albo przeładowaniu procesu NTP. Odpowiada za wypełnienie wszystkich pól w wysyłanej wiadomości NTP jako niezdefiniowane lub zerowe. Polu ostrzegawczemu przypisuje się wartość 3 (brak synchronizacji). W przypadku jeśli uruchamiana jest procedura autentyfikacji, procedura inicjalizacyjna odpowiada także za pobranie niezbędnych kluczy.
- Procedura realizacji inicjalizacji – jest wywoływana z procedury inicjalizacyjnej w celu zdefiniowania parametrów skojarzenia. Wypełnia dane adresowe oraz pole trybu na podstawie informacji odczytanych podczas procedury przeładowania albo na podstawie danych wprowadzonych przez operatora.
- Procedura realizacji odbioru – jest wywoływana z procedury odbioru w momencie pojawienia się nowego urządzenia zdalnego. Jeśli wiadomość jest otrzymana z urządzenia zdalnego pracującego w trybie klienta host przechodzi w tryb pracy serwera. w przeciwnym wypadku przechodzi w tryb pracy symetryczny pasywny.
- Procedura realizacji zegara pierwotnego – wywoływana z procedury inicjalizacyjnej w celu ustanowienia odpowiednich wartości zmiennych zegara pierwotnego. Wartość zmiennej Precision jest ustanawiana na podstawie specyfikacji zegara atomowego. Wartość zmiennej Rootdispersion jest ustawiana jako dziesięciokrotność błędu zegara atomowego.
- Procedura kasująca – jest wywoływana w przypadku zdarzeń które mają wpływ na dostępność lub potencjalne uszkodzenia zegara lokalnego. Ustawia wszystkie pola w pakiecie jako niezdefiniowane. Wywołuje procedurę wyboru zegara i procedurę aktualizacji.
- Procedura realizacji aktualizacji – wywoływana gdy zachodzą pewne znaczące zdarzenia w wyniku których może zmienić się okres aktualizacji (poll inteval). Procedura sprawdza wartość okresu aktualizacji (synchronizacji) urządzenia lokalnego i zdalnego a w przypadku błędu nadaje im wartości z ustalonego zakresu.

- Procedura dystansu synchronizacji – oblicza dystans synchronizacyjny względem zdalnych urządzeń w następujący sposób [36]:

$$x = y + z \tag{3.4}$$

Gdzie:

x – opóźnienie względem źródła pierwotnego

y – opóźnienie względem źródła synchronizacji

z – opóźnienie źródła synchronizacji względem źródła pierwotnego

$$e = Pr + Pd + \varphi(\text{czas lokalny} - \text{czas aktualizacji}) \tag{3.5}$$

Gdzie:

e – dyspersja względem źródła pierwotnego

Pr – dyspersja źródła synchronizacji względem źródła pierwotnego

Pd – dyspersja względem źródła synchronizacji

- Procedura filtracyjna – wywoływana po nadejściu wiadomości NTP oraz podczas innych zdarzeń dostarczających nowych danych. Chociaż standard definiuje postać procedury filtracyjnej to jej użycie nie jest konieczne. Algorytm filtracji może zostać dobrany do potrzeb dokładnościowych synchronizacji. W celu ułatwienia zrozumienia tej procedury poniżej przedstawiono jej działanie w postaci pseudokodu o składni podobnej do języka C++. Procedura filtracyjna opisana w standardzie pobiera obliczone argumenty (θ, δ, ε). Rolę filtra w procedurze pełni rejestr przesuwny w którym w każdym polu można zapisać 3 zmienne [θi, δi, εi]. Długość filtru (ilość pól rejestru) określona jest przez parametr Filter Size. Filtr inicjalizowany jest wartościami [0,0, Max Dispersion] we wszystkich stopniach. Nowo obliczane dane są zapisywane do ostatniego pola rejestru po lewej stronie zajmując miejsce starszych danych które są przesuwane w prawą stronę. Najstarsze dane są kasowane. Dodatkowo dane są umieszczane na tymczasowej liście o formacie pola [*distance, index*]. Następnie lista tymczasowa jest sortowana z rosnącym argumentem *distance* zgodnie z treścią poniższego pseudokodu [36]:

```
for (i = Filter Size – 1;i>=1;i--)          // pętla dla całego filtru
{                                           // przesuwanie filtru
    [θi, δi, εi] = [θi-1, δi-1, εi-1];      // obliczanie nowej
```

```
        εi = εi + φ(czas lokalny – czas aktualizacji);    // wartości dyspersji    dodaj
    [λi = εi +|δi|/2, i] do listy tymczasowej;
  }

 [θ0, δ0, ε0] = [θ, δ, ε];// dodaje nowe dane
   dodaj [λ = ε +|δ|/2, 0] do listy tymczasowej;          // resetowanie danych
   czas aktualizacji = czas lokalny;                      // bazowych
   sortuj listę tymczasową względem [distance ||index];
```

Następnie obliczana jest dyspersja filtru εσ. Operację tą przedstawia poniższy pseudokod przy założeniu, że posiadamy już posortowaną listę tymczasową [36]:

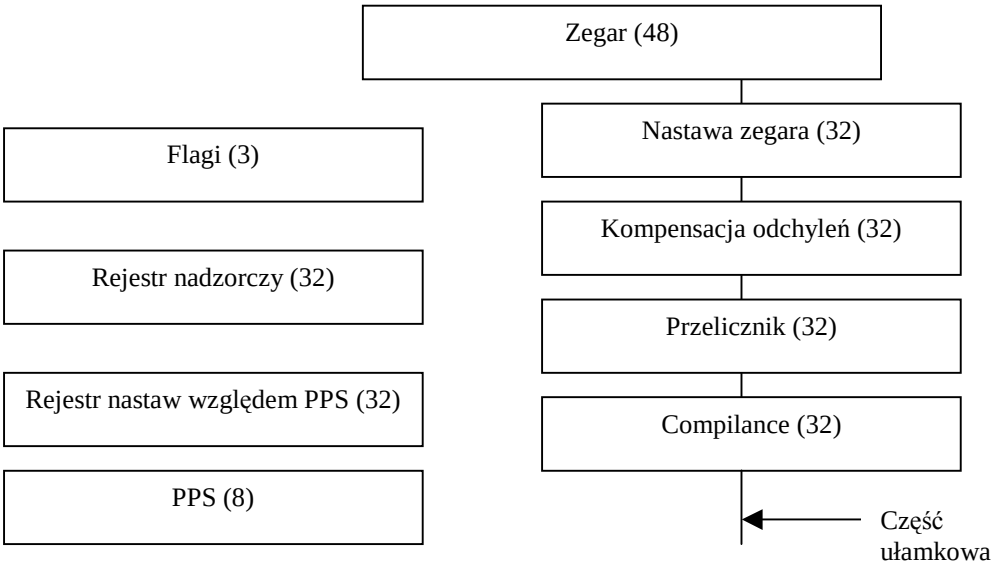
```
εσ = 0;                                          // zerujemy dyspersję filtru
for (i = Filter Size –1; i-- ; i>=0)
{
if (lista tymczasowa.dyspersja[i] ≥ Max Dispersion or
|θi - θ0| > Max Dispersion)
εσ ← (εσ + Max Dispersion) × Filter Weight; // obliczanie dyspersji
else
εσ ← (εσ + |θi - θ0|) x Filter Weight;           // obliczanie dyspersji
}
```

Wartości przesunięcia względem źródła synchronizacji, opóźnienia synchronizacji oraz skojarzonej z nim dyspersja są wybierane jako odnoszące się do próbki o najmniejszym dystansie synchronizacji. Innymi słowy do próbki występującej na pierwszej pozycji listy tymczasowej [36].

```
przesunięcie = θ0;                               // uaktualnienie zmiennych
opóźnienie = δ0;
dyspersja = min (ε0 + εσ, Max Dispersion);
```

- Procedura selekcji zegara – używa zmiennych (θ, δ, ε) oraz τ. Jest wywoływana w przypadku zmiany którejś z tych zmiennych albo zmiany osiągalności zdalnych urządzeń. Chociaż standard definiuje postać procedury selekcyjnej to jej użycie nie jest konieczne. Procedura wykonywana jest w przypadku gdy ilość źródeł synchronizacji jest większa niż jeden.

3.1.5 Zegar lokalny



Rys 3.4 Rejestry zegara [36]

Zegar lokalny składa się ze zbioru sprzętowych i programowych rejestrów razem ze zbiorem algorytmów implementujących zegar logiczny działający jako sterowany oscylator synchronizowany ze źródeł zewnętrznych. Główne komponenty zegara lokalnego są przedstawione na rys. 3.4. Część ułamkowa tyczy się części milisekundy. 48 bitowy rejestr zegara oraz 32 bitowy rejestr nastawczy tworzą sterowany oscylator zawierający ilość milisekund w stosunku do momentu północy. 32 bitowy rejestr nastawy zegara jest używany do nastawy fazy oscylatora w płynny sposób aby uniknąć nieciągłości i skoków we wskazywanym czasie. Rejestr kompensacji odchyleń jest używany do dostosowywania częstotliwości oscylatora przez dostosowywanie jego fazy w okresach nastawień. Kompensuje błędy częstotliwości rzędu 0,01%.

16 bitowy rejestr nadzorczy oraz 32 bitowy rejestr zgodności są używane dla zapewnienia poprawności pracy. 32 bitowy rejestr nastaw względem sygnału PPS używany jest jeśli dostępne jest zewnętrzne źródło tego sygnału, podczas gdy 8 bitowy rejestr PPS bierze udział w wykrywaniu takiego źródła.

Wszystkie rejestry oprócz rejestru przeliczającego są zaimplementowane w pamięci. Rejestr przeliczający jest to 16-bitowy buforowany (razem 32 bity) licznik sterowany przez oscylator kwarcowy i przepełniany z całkowitą wielokrotnością częstotliwości 1000Hz. Przepełnienie licznika wywołuje uruchomienie przerwania którego wynikiem jest inkrementacja rejestru zegara.

Zdefiniowano dodatkowo parametry zegara lokalnego. Standard definiuje ich znaczenie oraz zalecane wartości, jednak ich użycie nie jest konieczne.

Tabela 3.2 Parametry zegara lokalnego [36]

Parametr	Wartość	Opis
Adjustment Interval	4 s	Okres aktualizacji zegara
PPS Timeout	60 s	Czas przepełnienia licznika PPS
Maximum Aperture	± 128 ms	Maksymalne przesuniecie zegara względem źródła przy którym stosuje się jeszcze płynne dostrojenie fazy
Frequency	16	Waga zmiennej oznaczającej częstotliwość
Phase Weight	8	Waga zmiennej oznaczającej fazę
Compliance Maximum	4	Maksymalna wartość rejestru compliance

Podczas inicjalizacji systemu wszystkie liczniki i rejestry są zerowane, a status zegara ustawiana jest jako niezsynchronizowany. Wartość parametru Adjustment Interval jest otrzymywana z licznika PPS tylko jeżeli jego poprzednia wartość jest większa od 0.

W zależności od przesunięcia zegara lokalnego względem źródła synchronizacji możliwe są 2 sposoby dostrojenia czasu lokalnego. W przypadku kiedy wartość przesunięcia przekracza 128ms dokonuje się skokowego dostrojenia czasu. W przeciwnym przypadku dokonuje się płynnego dostrojenia fazy.

Płynne dostrojenie fazy

Zegar lokalny pracuje z częstotliwością i przesunięciem zgodnym z wynikami ostatniej aktualizacji. Zawiera ona 48 bitową dodatnią lub ujemną liczbę milisekund oraz ich część ułamkową, przy czym na część całkowitą poznaczonych jest 16 najstarszych bitów. Aktualizacja dochodzi do skutku po poprawnej selekcji źródła synchronizacji. Normalnie wartość aktualizacji jest określona przez procedurę filtracji zegara. Jeśli jednak wartość licznika PPS jest większa od 0 (co oznacza, że do komputera jest bezpośrednio przyłączone zewnętrzne źródło wzorcowe jak np.: zegar cezowy) do aktualizacji jest używana wartość rejestru nastawy zegara względem PPS.

Niech *u* będzie 32 bitową liczbą z bitami 0-31 ustawionymi jako bity 16-47 wartości aktualizacji. Jeżeli jakieś mniej znaczące bity aktualizacji są nieużywane, są wypełniane wartością znaku liczby. Ta operacja sprawia, że 16 najstarszych bitów liczby *u* reprezentuje wartość całkowitą oraz znak aktualizacji, reszta to część ułamkowa, mająca na celu minimalizację błędów Niech *b* będzie liczbą znaczących zer absolutnej wartości rejestru zgodności natomiast *c* liczbą znaczących zer licznika nadzorczego. Wtedy nową wartość *b* obliczmy w następujący sposób [36]:

```
if ((b – 16 + Compliance Maximum)<0)
b = 0;
else
b = b – 16 + Compliance Maximum;
```

Wynik reprezentuje wartość stałej czasowej pętli. Następnie obliczmy *c*:

```
c = 10 – c ;
if (c < Min Polling Interval)
c = Min Polling Interval;
else if (c > Max Polling Interval)
c = Max Polling Interval;
```

Otrzymana liczba zawierająca się pomiędzy Min Polling Interval, a Max Polling Interval reprezentuje logarytm interwału od ostatniej aktualizacji. W dalszej kolejności obliczana jest nowa wartość rejestru aktualizacji zegara *x* oraz rejestru kompensacji odchyleń *y*[36]:

```
x = u >> b;
y = y + (u >> (b + b - c) );
```

Czasochłonne operacje dzielenia oraz mnożenia są aproksymowane za pomocą operacji przesuwania bitów (w lewo dla mnożenia oraz w prawo dla dzielenia) i oznaczane są symbolami odpowiednio << oraz >>. Operacje mogą być zatem wykonywane w czasie rzeczywistym nawet za pomocą bardzo prostego sprzętu.

Ostatecznie obliczana jest korekcja zegara składająca się z dwóch komponentów. Pierwszy z nich (faza) obliczany jest w następujący sposób:

```
x >> Phase Weight;
```

Drugi (częstotliwość) obliczany jest w następujący sposób:

y >> Frequency Weight;

Suma tych komponentów (z wagami zawartymi w parametrach Frequency oraz Phase Weight) jest ostatecznie dodawana do rejestru zegara. Następnie zerowany jest rejestr nadzorczy. Aktualizowane jest pole Pool Interval w następujący sposób:

Pool Interval = b + Min Polling Interval;

Operacja jest kontynuowana w ten sposób aż do momentu kolejnej korekcji. W przypadku dużych szumów aktualizacji lub wysokich zmian częstotliwości oscylatora sprzętowego rośnie wartość rejestru Compliance, natomiast maleje okres synchronizacji. W przypadku małych szumów oraz przy stabilnym oscylatorze sprzętowym maleje wartość rejestru Compliance, a okres synchronizacji rośnie (maksymalnie do wartości Max Polling Interval). W dobrych warunkach możliwa jest do uzyskania precyzja rzędu milisekundy na dobę.

Skokowe dostrojenie czasu

Jeśli wartość korekcji zegara przekracza wartość parametru Maximum Aperture istnieje możliwość, że zegar jest jak dotąd niezsynchronizowany ze źródłem odniesienia albo nastąpiło jakieś znaczne pogorszenie jakości łącza. W takim przypadku najlepszym rozwiązaniem jest zastąpienie zawartości rejestru zegara zamiast płynnego dostrojenia jego wartości. Jeśli zatem skokowa zmiana jest wskazana, aktualizacja dodawana jest bezpośrednio do rejestru zegara a rejestr aktualizacji zegara oraz licznik nadzorczy są zerowane. Zawartość pozostałych rejestrów pozostanie bez zmian. W dalszej kolejności status zegara ustawiany jest jako niezsynchronizowany (leap = 11). W praktyce potrzeba wykorzystania skokowej aktualizacji jest rzadka i występuje najczęściej po restarcie źródła synchronizacji.

Podstawowy model NTP zakłada, ze zadaniem hosta jest weryfikacja czasu NTP i synchronizacja względem niego [36]. Jednak w niektórych urządzeniach np. zasilanych bateryjnie, elektronicznych zegarach, elektronicznych kalendarzach protokół NTP nie powinien być wykorzystywany jako źródło czasu tylko w celu sprawdzenia poprawności działania zegara oraz sporadycznej synchronizacji w przypadku dużych przesunięć. Do tego typu urządzeń zalicza się zegar będący tematem niniejszej pracy. W niektórych przypadkach kiedy taki zegar lub kalendarz jest bardziej niezawodny lecz mniej stabilny niż sieć synchronizacji NTP jest wskazane aby zawartość rejestru zegara

zawierała czas wynikający z lokalnego kwarcu, a inne rejestry, liczniki, flagi powinny być ustawiane w sposób wcześniej opisany.

Konwersja z formatu NTP do popularnych formatów daty i czasu używanych przez aplikacje jest znacznie uproszczona jeśli wewnętrzny format lokalnego zegara używa osobno zmiennych daty i czasu. Data jest przepisywana bezpośrednio ze stempla przychodzącego pakietu NTP lub zmieniana po każdej godzinie 24 odliczonej przez wewnętrzny zegar. Zmienna czasu jest tak zaprojektowana aby zerowała się po 24 godzinach wydłużonych lub skróconych o sekundę, co wynika z wartości pola leap znajdującego się w nagłówku wiadomości. Dzień przed usunięciem lub dodaniem sekundy do licznika czasu pole leap jest zmieniane w serwerze pierwotnym. Najczęściej ręcznie. Następnie sekwencyjnie wartość pola jest przekazywana do zegarów o coraz wyższej liczbie stratum. Po dokonaniu wstawienia lub usunięcia sekundy (ostatnia minuta doby może mieć 61 lub 59 sekund) pole leap jest resetowane.

3.1.6 Zastosowanie synchronizacji NTP

Protokół NTP znajduje zastosowanie w coraz szerszym zakresie rozwiązań, zarówno dla potrzeb codziennych jak i w zawansowanych rozwiązaniach telekomunikacyjnych, rachunkowości elektronicznej. Dotychczas NTP znalazł zastosowanie w:

- systemach bilingowych np. telekomunikacja i energetyka
- cyfrowych przekazach dźwięku i obrazów np. telekonferencje
- sterowaniu ruchem naziemnym i lotniczym
- zapewnieniu zgodności czasu pomiędzy routerami
- szyfrowaniu transmisji
- zdalnych systemach ochrony osób i mienia
- ochronie, certyfikacji oraz autoryzacji danych
- automatyce przemysłowej
- systemach czasu rzeczywistego
- środowiskach wieloserwerowych
- macierzach i kastrach komputerowych
- systemach wieloprocesorowych i rozproszonych
- obliczeniach astronomicznych
- jako stempel czasu w podpisie elektronicznym

3.2 Synchronizacja czasu poprzez sieć GPS

Sieć GPS zapewnia dobre możliwości synchronizacyjne, ponieważ każdy z satelitów z którym bezpośrednio synchronizowany jest odbiornik posiada na swoim pokładzie zegar atomowy. Dodatkowo stacje naziemne kontrolują chód zegarów atomowych satelitów (które wymagają źródła synchronizacji (np.: cezowego) dla zachowania długoterminowej stabilności), podając w depeszy poprawki w stosunku do chodu zegarów laboratoryjnych [27]. Dodatkowo struktura wiadomości GPS umożliwia odbiornikowi wyznaczenie czasu, jaki upłynął od momentu wysłania sygnału przez satelitę do momentu odbioru. W ten sposób każdy odbiornik odbierający sygnał GPS ma możliwość synchronizacji własnego zegara z systemowymi zegarami GPS przy dokładności synchronizacji czasu rzędu 60 nanosekund. Jako, że nadajniki satelitów mają stosunkowo małą moc (w stosunku do powierzchni Ziemi na którą promieniują) wymagana jest bezpośrednia widoczność odbiorników z satelitą. Przykładem zaawansowanego urządzenia wykorzystującego do dalszej synchronizacji sieć GPS może być synchronizowane źródło sygnału referencyjnego SYN – Rb opracowane na Politechnice Poznańskiej.

Każdy satelita wysyła zakodowane dwie fale nośne charakterystyczne tylko dla niego: na częstotliwości L1 = 1575.42 MHz, oraz na częstotliwości L2 =1227.60 MHz. Syntezowanie są one z podstawowej częstotliwości L =10.23 MHz generowanej przez zegar atomowy znajdujący się na pokładzie satelity przez pomnożenie podstawowego sygnału przez 154 i 120. Następnie na tak otrzymane sygnały nakłada się kodowane wiadomości. Używane są trzy rodzaje kodów binarnych:

- kod C/A modulujący częstotliwość L1, nadawany w postaci sygnału binarnego. Kod C/A jest sygnałem o bardzo długim okresie (267 dni), (którego segmenty są przypisane poszczególnym satelitom) dlatego przez obserwatora może być postrzegany jako sygnał pseudolosowy (PRN). To właśnie na jego podstawie odbiornik sygnału GPS może rozróżnić satelity.
- kod P (Precise) o częstotliwości 10MHz służący do precyzyjnego wyznaczania pozycji, modulujący obie częstotliwości: L1 i L2. Zawarte w nim poprawki pozwalają na korektę błędu pozycji oraz czasu związanego z występowaniem jonosfery.

Wiadomości nawigacyjne nadawane są na modulowanej przez C/A częstotliwości. Dodatkowo transmitowane są dane o stanie satelitów, aktualne współczynniki do obliczenia opóźnienia jonosferycznego i dane do obliczenia czasu UTC.

3.2.1 Jednodrogowa metoda synchronizacji poprzez GPS

Jednodrogowa metoda synchronizacji jest najprostszą, najczęściej stosowaną i najmniej dokładną metodą synchronizacji. W przypadku tej techniki synchronizacji źródło sygnału odniesienia (satelita systemu GPS) po prostu wysyła aktualny stempel czasu do odbiornika poprzez medium transmisyjne. To właśnie czas jaki potrzebny jest na przesłanie stempla poprzez medium jest głównym źródłem błędów synchronizacyjnych. Zatem w celu zapewnienia dokładnej synchronizacji należy zapewnić jak najkrótszą drogę pomiędzy odbiornikiem i nadajnikiem (co jest trudne do zapewnienia w systemie GPS z racji jego rozległej architektury), tak aby czas propagacji sygnału był pomijalnie mały, lub wysoce dokładnie wyznaczyć opóźnienie transmisyjne w przypadku długiego kanału transmisyjnego w celu korekcji błędów z nim związanych. Opóźnienie transmisyjne wynoszące w najlepszym przypadku 3.3 μs na kilometr, przyjmuje duże wartości w przypadku odległości typowych dla systemu (powyżej 20000 km). Co za tym idzie, w przypadku stosowania tej metody synchronizacji lokalizacja odbiornika musi być dokładnie znana. Oczywiście w przypadku małych wymagań na dokładność transmisji opóźnienia propagacyjne można po prostu pominąć. Sytuacja przedstawia się nieco lepiej w przypadku synchronizacji częstotliwości. W tym przypadku nie musi być znane opóźnienie transmisyjne kanału, lecz jedynie jego zmienność. To właśnie dzięki możliwości określenia parametrów kanału transmisyjnego GPS jest najdokładniejszym ogólnoświatowym jednostronnym system transferu czasu [16].

3.2.2 Dwudrogowa metoda synchronizacja czasu poprzez GPS

Metoda dwudrogowa synchronizacji jest prostą metodą pozwalającą na bezpośrednie porównanie dwóch zegarów w zdalnych miejscach. W tej technice dwa znajdujące się w oddaleniu zegary wymagające synchronizacji między sobą otrzymują jednostronny sygnał równocześnie od wybranego satelity i mierzą różnicę między czasem zawartym w stemplu, otrzymanym z satelity a czasem zegara lokalnego. Następnie dane o obliczonych różnicach czasu są wymieniane między

synchronizowanymi stacjami używając wybranej metody metodę (e-mail, FTP, itp.). Różnica czasu między odległymi zegarami jest obliczana w następujący sposób:

$$D = t_S - t_A - (t_S - t_B) = t_B - t_A \tag{3.6}$$

Gdzie:

t_S - czas zawarty w stemplu otrzymanym z satelity

t_A - czas lokalny zegara A w momencie otrzymania stempla czasu z satelity

t_B - czas lokalny zegara B w momencie otrzymania stempla czasu z satelity

Na podstawie wzoru 3.6 można zauważyć, że różnię czasu pomiędzy zegarami A i B można uniezależnić od czasu satelity. Jeżeli czasy podróży do odbiorników są dokładnie równe, wtedy dwa odbiorniki mogą zsynchronizować swoje zegary z dokładnością, która nie zależy od cech nadajnika albo środka transmisji. Także jeżeli czasy podróży sygnału do odbiorników są różne metoda ta daje bardzo dobre wyniki, ponieważ wahania czasów podróży do obu zegarów są najczęściej wysoce skorelowane (szczególnie kiedy synchronizowane zegary znajdują się blisko siebie, a co za tym idzie warunki atmosferyczne dla dróg zegar- satelita obu z nich są takie same).

4. Projekt zegara czasu rzeczywistego

Integralną częścią niniejszej pracy jest wykonanie projektu i prototypu zegara czasu rzeczywistego synchronizowanego protokołem NTP. Projektowany zegar jest zegarem cyfrowym z wyświetlaczem numerycznym. Na wyświetlaczu przedstawiona jest aktualna godzina wraz z minutami. Zegar synchronizowany jest poprzez sieć z wybranym serwerem NTP za pośrednictwem komputera PC wyposażonego w dedykowane dla układu oprogramowanie. Dalej informacja synchronizacyjna przesyłana jest z komputera do zegara cyfrowego za pośrednictwem interfejsu RS - 232C. Zegar jest projektowany i wykonany w celu umieszczenia go w widocznym miejscu w budynku wydziału Elektroniki i Telekomunikacji Politechniki Poznańskiej. Związany jest z tym faktem szereg problemów technicznych które zostaną omówione w dalszej części niniejszej pracy.

Praca nad projektem składa się z pięciu części:

- Wykonanie oprogramowania dla komputera PC obsługującego przesył danych synchronizacyjnych z sieci Internet do zegara cyfrowego, w którego skład zaimplementowane są mechanizmy NTP. Oprogramowanie napisane jest w środowisku programistycznym Builder C++ 6.0 firmy Borland i dedykowane jest do pracy w środowisku Windows.
- Zaprojektowanie kompletnego schematu układu zegara mikroprocesorowego obejmującego blok zasilania, blok wyświetlaczy, oraz blok sterowania. Projekt zegara został stworzony za pomocą programu Protel 99 SE.
- Zaprojektowanie na podstawie schematu i wykonanie płyt drukowanych zegara.
- Napisanie oprogramowania dla mikrokontrolera AT89C2051 będącego sterownikiem układu zegara i dla tego układu dedykowanego.
- Montaż i uruchomienie zegara.

4.1 Zegar mikroprocesorowy

Zaprojektowanie i wykonanie zegara mikroprocesorowego jest główną częścią niniejszego projektu. Układ zegara składa się z trzech podstawowych bloków funkcjonalnych wykonanych na oddzielnych płytach drukowanych a mianowicie: z bloku zasilania, wyświetlaczy, oraz bloku sterowania.

4.1.1 Blok zasilania

Zegar będący tematem projektu jest wykonany celem umieszczenia go w budynku wydziału Elektroniki i Telekomunikacji Politechniki Poznańskiej. Z tego powodu niemożliwe jest jego usytuowanie ani w pobliżu komputera PC za pomocą którego będzie synchronizowany, ani w pobliżu źródła zasilania sieciowego. W tym przypadku możliwe były dwa rozwiązania:

- umieszczenie bloku sterowania, oraz bloku zasilania we wspólnej obudowie w pobliżu komputera oraz źródła zasilania i transmitowanie gotowych sygnałów zasilających poszczególne segmenty wyświetlaczy LED na odległość kilkudziesięciu metrów. Główną wadą tego rozwiązania jest konieczność użycia aż czternastożyłowego kabla. Kolejną wadą jest możliwość pojawienia się zakłóceń związanych z transmisją sygnału prostokątnego o częstotliwości 500Hz i mocy 1,5W.
- umieszczenie bloku zasilania w pobliżu komputera oraz źródła zasilania oraz przesył sygnałów synchronizacji w postaci cyfrowej (za pomocą kabla podłączonego do złącza COM komputera) oraz zasilania (w postaci stabilizowanego prądu stałego) do bloku sterowania znajdującego się we wspólnej obudowie wyniesionej wraz z wyświetlaczami. Główną wadą tego rozwiązania jest potrzeba przesyłania danych w standardzie RS-232C na dużą odległość wraz z zasilaniem, które może wprowadzać zakłócenia.

W projekcie zostało zastosowane rozwiązanie drugie, tj. umieszczono bloku sterowania wraz z wyświetlaczami we wspólnej obudowie. W tym celu został zaprojektowany specjalny układ zasilający.

W skład układu zasilającego wchodzi transformator sieciowy TS 15/34, oraz płytka z obwodem drukowanym na której znajduje się prostownik wraz ze

stabilizatorem. Układ zasilający (zasilacz) umieszczono w obudowie uniwersalnej Z16. Urządzenie zaopatrzone zostało w przewód zasilania sieciowego o dł. 1,5m. Schemat ideowy zasilacza przedstawiony jest w załączniku A3. Użyty transformator sieciowy TS 15/34 posiada dwa uzwojenia wtórne o napięciu wyjściowym 13,5 V i prądzie znamionowym 0,5 A. Uzwojenia te zostały zwarte aby zsumować ich prądy i umożliwić zasilanie układu prądem do 1 A. Jako prostownik dwupołówkowy napięcia przemiennego pochodzącego z transformatora zastosowano mostek Graetza. Zastosowano cztery diody 1N4001 zgodnie z ich specyfikacją [28]. Zgodnie z załącznikiem A3 wyprostowane napięcie filtrowane jest wstępnie za pomocą układu dwóch kondensorów. Kondensator elektrolityczny ze względu na stosunkowo dużą pojemność odpowiedzialny jest za zgrubne odfiltrowanie zakłóceń o niskich częstotliwościach (głównie rzędu 50Hz). Odfiltrowane wstępnie z zakłóceń napięcie jest podawane na wejście stabilizatora napięcia AN7810 zgodnie ze specyfikacją [35]. Jego zadaniem jest dokładne odfiltrowanie zakłóceń z napięcia zasilającego. Stabilizator napięcia AN7810 został wybrany na potrzeby budowy urządzenia nie tylko ze względu na niską cenę, lecz także dlatego, że jest to wyspecjalizowany element nie wymagających zewnętrznego napięcia odniesienia, posiadający wbudowany układ kompensacji temperatury, wyposażony w zabezpieczenie nadprądowe (ograniczenie prądu większego od znamionowego) oraz zabezpieczenie przed przegrzaniem. Dzięki temu te dodatkowe układy nie muszą być umieszczanie na płycie zasilacza. Obudowę stabilizatora zaopatrzono dodatkowo w radiator odprowadzający ciepło. Jak widać na rysunku w załączniku A3 do wyjścia stabilizatora AN7810 dołączone są dwa kondensatory elektrolityczne znajdujące się po dwóch stronach kabla podłączonego do zegara. Ich zadaniem jest niwelowanie negatywnego wpływu długiego kabla na parametry przesyłanego w nim zasilania. Zasilacz zaopatrzono w diodę LED umieszczoną w obudowie, przyłączoną do wyjścia stabilizatora i sygnalizującą prawidłową jego pracę. Na jego obudowie umieszczono dwa gniazda DB-9. Do jednego z nich (opatrzonego napisem „Do komputera”) dołączony jest przewód biegnący od pobliskiego komputera. Do drugiego (opatrzonego napisem „Do zegara”) dołączony jest kabel biegnący do wyniesionego zegara. Używane piny ze złącza „Do komputera” są połączone bezpośrednio z odpowiadającymi im pinami w złączu „Do zegara”, zatem dane synchronizacyjne są przesyłane przez zasilacz do zegara bez żadnej ingerencji. Ponieważ zegar znajduje się w miejscu ciężko dostępnym (pod sufitem) przycisk „RESET” wprowadzający mikrokontroler AT89C2051 w stan początkowy jest

przeniesiony na obudowę zasilacza, zatem oprócz zasilania i sygnałów synchronizacji w kablu łączącym zasilacz z zegarem dodatkowo dwie żyły przeznaczono do przesłania sygnału „RESET”.

4.1.2 Blok sterowania

Blok sterowania zawiera: mikrokontroler AT89C2051, układ konwersji napięć sterujących, układ stabilizacji i obniżania napięcia, dwanaście wzmacniaczy tranzystorowych, oraz generator kwarcowy. Blok jest umieszczony na osobnej płytce zamkniętej wraz z wyświetlaczami w jednej obudowie. Schemat ideowy bloku sterowania przedstawiony jest. w załączniku A1. Płyta bloku sterowania połączona jest z zasilaczem za pomocą pięciożyłowego kabla, natomiast z płytą wyświetlaczy za pomocą szesnastoprzewodowej taśmy. Dwa kondensatory o pojemnościach 100nF (C20 oraz C222, patrz załącznik A1) zostały umieszczone możliwie najbliżej układów AT89C2051 oraz MAX232 żeby zminimalizować indukcyjność ścieżek między nimi. Kondensatory te dzięki znikomej indukcyjności pasożytniczej, zdolne są do filtracji napięć zasilających owe układy z zakłóceń o częstotliwościach rzędu kilku kHz, po to aby zakłócenia nie miały wpływu na ich pracę. Jak wspomniano wcześniej takie zakłócenia pojawiają się w układzie z powodu przełączania wyświetlaczy, a co za tym idzie szybkozmiennego obciążenia zasilania. Układy AT89C2051 oraz MAX232 wymagają zasilania napięciem 5V. W tym celu w bloku sterowania zastosowano stabilizator napięcia AN7805 zgodnie ze specyfikacją [35] obniżający napięcie z 10V na 5V. Do bloku sterowania doprowadzono napięcie 10V celem zasilania wzmacniaczy sygnałów włączających poszczególne segmenty wyświetlaczy (aby włączyć segment wyświetlacza wymagane jest napięcie minimum 7V, co zostało wykazane doświadczalnie a jest niezgodne ze specyfikacją wyświetlaczy dostarczoną przez producenta [34]). Jak już wspomniano mikrokontroler AT89C2051 będący jedynym elementem cyfrowym stanowiącym o działaniu układu jest jego sercem. Jego podstawowym zadaniem jest sekwencyjne sterowanie wyświetlaczy zgodnie z treścią dedykowanego programu. Kod aktualnie wyświetlanej cyfry przesyłany jest siedmioma najstarszymi wyjściami portu P1. Wyjście P1.0 służy do włączania i wyłączania diod separatora godzinowego z częstotliwością 0.5Hz. Mikrokontroler steruje sekwencyjnym włączaniem i wyłączaniem kolejnych cyfr wyświetlacza za pośrednictwem wyjść 2, 3, 4, 5 portu P3. Układ działa w ten sposób, że kod cyfry doprowadzony jest do

wszystkich wyświetlaczy, natomiast tylko jeden z nich jest w danej chwili zasilany. W kolejnym cyklu ostatnio włączony wyświetlacz jest wygaszany. Na wyjściach P1.1 – P1.7 pojawia się kod nowej cyfry a za pośrednictwem portu P3 włączany jest kolejny wyświetlacz przez który dana cyfra ma zostać wyświetlona. Procedura odbywa się cyklicznie z częstotliwością 500 Hz dając obserwującemu złudzenie, że wszystkie cztery wyświetlacze świecą jednocześnie.

Kolejną funkcją mikrokontrolera jest odmierzanie czasu w przerwach miedzy momentami synchronizacji ze źródłem zewnętrznym. Podstawą czasu jest w tym przypadku długość pojedynczego cyklu maszynowego układu i jest to najkrótszy odcinek czasu z dokładnością do którego można wyregulować pracę zegara. Długość cyklu maszynowego jest równa 12 okresom (taktom) rezonatora kwarcowego przyłączonego do wejść XTAL1 i XTAL2 mikrokontrolera zgodnie ze specyfikacją [32]. Zatem przy zastosowaniu kwarcu o okresie drgań 12,000 MHz. długość cyklu maszynowego jest równa 1μs. Cykle maszynowe zliczane są za pomocą licznika T0, używanego następnie do aktualizacji rejestrów czasu, co zostanie dokładnie opisane w rozdziale 4.2.

Ostatnią z funkcji mikrokontrolera jest współpraca z komputerem PC dołączonym za pośrednictwem portu COM po stronie komputera oraz wyprowadzeń P3.0 (RxD) oraz P3.1 (TxD) po stronie mikrokontrolera. Każdy port mikrokontrolera może zostać wykorzystany zarówno jako wyjście oraz wejście logiczne. W przypadku kiedy port wykorzystywany jest jako wejście przechodzi w stan wysokiej impedancji, a zatem podany na niego poziom logiczny pochodzący z innego układu cyfrowego może zostać bezpośrednio odczytany i zinterpretowany do dalszego działania. W przypadku układu transmisja odbywa się jedynie za pomocą linii RxD i TxD w trybie asynchronicznym. Stany logiczne 1 oraz 0 kodowane są bezpośrednio stanami napięć w linii. Dane synchronizacyjne z komputera przesyłane magistralą szeregową bit po bicie w postaci znaków ASCII są bezpośrednio interpretowane przez mikrokontroler. Nie jest zatem potrzebna konwersja kodu cyfrowego na zrozumiały dla układu. Problemem są za to różnice w poziomach napięć stanów logicznych występujących na złączu w standardzie RS-232 oraz występujących na wyprowadzeniach mikrokontrolera.

W przypadku standardu RS-232 (linie danych):

– napięcie od –3V do –15V interpretowane jest jako logiczna 1

– napięcie od 3V do 15V interpretowane jest jako logiczne 0

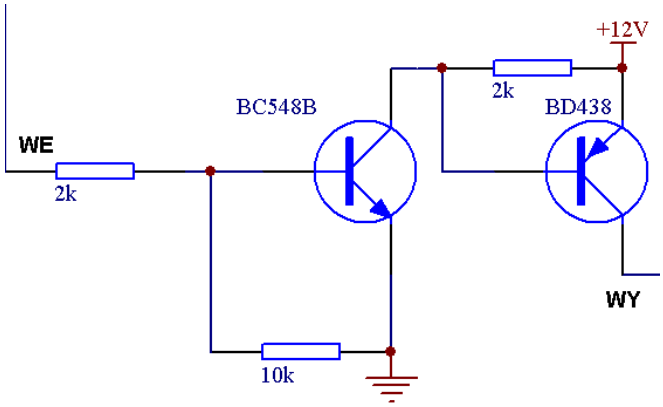
W przypadku mikrokontrolera zasilanego napięciem 5V akceptowalne poziomy napięć to [32]:

– napięcie od –0,5 V do 0,9 V interpretowane jest jako logiczne 0

– napięcie od 1,9V do 5V interpretowane jest jako logiczna 1

Niezbędna jest zatem konwersja napięć umożliwiająca współdziałanie między komputerem a mikrokontrolerem. W tym celu w przypadku niniejszego projektu zastosowano układ MAX232 dokonujący konwersji poziomów sygnałów. Układ ten jest przeznaczony do współpracy ze sygnałami zgodnymi ze standardem RS-232. Zgodnie ze specyfikacją [33] do prawidłowej pracy układ wymaga przyłączenia czterech kondensatorów elektrolitycznych o pojemnościach po 1uF każdy. Układ posiada cztery wejścia i cztery wyjścia sygnałowe, zatem może jednoczenie obsłużyć komunikację między dwoma urządzeniami zgodnymi ze standardem RS232 mikrokontrolerem. W przypadku zegara wykorzystano wyprowadzenia T2OUT oraz R2IN po stronie komputera oraz T2IN i R2OUT po stronie mikrokontrolera.

Ze względu na małą moc sygnałów sterujących pochodzących z mikrokontrolera nie mogą one bezpośrednio zasilać wyświetlaczy LCD, zatem poddawane są wzmocnieniu. Sygnały pochodzące z portu P1 sterują katodami wyświetlaczy. Jako, że użyte wyświetlacze SA40-SRWA pracują w układzie wspólnych anod [44], to większe prądy przepływają przez wyprowadzenia wspólnych anod, niż pojedynczej katody. Dowiedziono doświadczalnie, że przez wyprowadzenie katody przepływa prąd o wartości skutecznej do 25 mA. Zatem jako wzmacniacz sygnału włączania katody pochodzącego z mikrokontrolera zastosowano tranzystor małej mocy BC548 w układzie wspólnego emitera. Do każdego tranzystora dołączone są dwa rezystory jak na rysunku w załączniku A1, pełniące w tym układzie rolę dzielnika napięcia. Wartości rezystorów zostały dobrane tak, aby w przypadku, gdy na odpowiadającym danemu tranzystorowi wyjściu mikrokontrolera występuje stan niski, napięcie przyłożone do bazy tranzystora było niższe od napięcia progowego. Zatem w takim przypadku tranzystor pozostanie zatkany i przez złącze nie popłynie prąd. Jednocześnie gdy wyjście mikrokontrolera zostanie wprowadzone w stan wysoki napięcie przyłożone do bazy musi być wyższe od napięcia progowego otwierając tranzystor. W takim układzie tranzystor pracuje jako przełącznik. Podobna sytuacja występuje w przypadku sygnałów pochodzących z portu P3. W tym jednak wypadku z powodu wyższej mocy przełączanego sygnału należało zastosować tranzystor średniej mocy BD438 zgodnie z rys. 4.1.



Rys. 4.1 Układ wzmacniacza dwustopniowego zastosowanego do przełączania wyświetlaczy

Jak widać na rys. 4.1 do przełączania wyświetlaczy (pracujących w układzie wspólnych anod) wykorzystano wzmacniacze dwustopniowe. Pierwszy stopień oparty na tranzystorze małej mocy BC548 (identyczny jak przedstawiony wcześniej) odwraca fazę sygnały. Podobnie jest w przypadku drugiego stopnia opartego na tranzystorze średniej mocy BD438. Podwójne odwrócenie fazy (o Π każde) skutkuje tym że układ jako całość nie odwraca fazy.

Na płycie układu nie zmontowano przycisku RESET. Jak wspomniano wcześniej trudny dostęp do zegara uniemożliwiałby jego użycie, dlatego właśnie przycisk wyniesiono na obudowę zasilacza, przy czym sygnał RESET transmitowany jest wspólnym kablem wraz z zasilaniem i sygnałami synchronizacji, za pomocą osobnej pary żył.

4.1.3 Blok wyświetlaczy

W projektowanym układzie zastosowano cztery siedmiosegmentowe wyświetlacze LED model: SA-40SRWA firmy KINGBRIGHT o wysokości pojedynczej cyfry równej 101mm [34]. Dzięki zastosowaniu wyświetlaczy o podwyższonej jasności możliwy jest odczyt wskazań nawet jeśli zegar znajduje się w nasłonecznionym miejscu. Dodatkowo ze względu na wysoka jasność w szerokim zakresie wartości napięć zasilających znacznie ułatwiony jest ich dobór.

Ze względu na znaczne rozbieżności między rzeczywistymi parametrami pracy a wyszczególnionymi w specyfikacji elementów wartości napiec zasilania zostały dobrane doświadczanie. W celu utrzymania postrzeganej jasności na stałym poziomie wyświetlacze odświeżane z częstotliwością 500Hz wymagają wyższego napięcia

zasilania niż w przypadku zasilania napięciem stałym (wynoszącym 7,5V). Spadek napięcia na złączu PN stosowanym np.: w diodach prostowniczych spolaryzowanym w kierunku przewodzenia wynosi ok. 0,7V. Dzięki temu w celu dokładnego dostrojenia wartości napięcia zasilającego wyświetlacze na płycie umieszczono zestaw diod 1N4001. Diody są wpięte w szereg z wyprowadzeniami katod wyświetlaczy zgodnie z rysunkiem w załączniku A1. Zaletą tego rozwiązania (wraz ze stosowaniem w zasilaczu stabilizatorów o różnych napięciach wyjściowych) jest możliwość regulacji napięcia zasilającego wyświetlacze w bardzo szerokim zakresie (od 6,9V do 11,4V) bez konieczności modyfikacji pozostałej części układu. Jest to możliwe także dzięki zastosowaniu drugiego stabilizatora napięcia zasilającego pozostałą część układu i uniezależniającego go od napięcia wejściowego.

Na wyświetlaczu zegara przewidziano dwukropek oddzielający wyświetlacz godzin i dziesiątek minut. Jest on zbudowany z dwóch diody LED. Diody połączone równolegle względem siebie zasilane są napięciem 3,6V otrzymanym poprzez włącznie w szereg z zasilaniem 5V dwóch diod 1N4001.

4.1.4 Płyty obwodów drukowanych

Jak wspomniano wcześniej elementy zegara osadzone zostały na trzech płytach obwodów drukowanych: płycie zawierającej blok wyświetlaczy, płycie zawierającej blok zasilania oraz płycie zawierającej blok sterowania. Jako, że układ charakteryzuje się mała komplikacją połączeń, w celu obniżenia kosztów ścieżki przewodzące umieszczono tylko po jednej stronie płyty (płyta jednowarstwowa). Nieliczne połączenia ścieżek wymagające przejścia przez dodatkową warstwę realizowane były za pomocą zwór wlutowywanych na płycie. Płyty projektowane były za pomocą programu Protel 99SE w wersji testowej 30-dniowej na podstawnie schematu ideowego układu. Kompletny schemat układu wraz z płytami obwodów drukowanych znajduję się w trzech plikach dołączonych do niniejszej pracy na płycie CD. Każdy z plików obejmuje płytę i blok zgodny z jego nazwą. Projekty płyt tworzone były zgodnie z następującą konwencją:

- schemat ścieżek przewodzących płyt drukowanych umieszczony jest na warstwie „BottomLayer” projektu
- Opisy elementów umieszczone są na warstwie „TopOverlay” projektu

- Komponenty typu „Pad” oraz „Via” umieszczone zostały na warstwie „Multilayer” projektu.

Ścieżki przewodzące na płytach wykonane zostały metodą fotochemiczną na podstawie warstwy „BottomLayer” projektu płyt. Następnie w płytach wykonano otwory na podstawie warstwy „Multilayer”. Na płytach umieszczony został nadruk na podstawie warstwy „TopOverlay” projektu płyt opisujący wlutowywane elementy.

Kształty i rozmiary elementów niezbędne na etapie projektowania i rozmieszczenia ścieżek na płytach (w postaci szablonów „footprint”) tworzone były bądź to na podstawie specyfikacji elementów, bądź na podstawie pomiarów wielkości zakupionych wcześniej elementów. Płyty zawierające bloki sterowania oraz wyświetlaczy zaopatrzono w otwory o średnicy 5mm służące do zamocowania płyt w obudowie. W gotowe płyty wlutowano elementy po uprzednim oczyszczeniu ich wyprowadzeń za pomocą pasty lutowniczej. Mikrokontroler AT89C2051 nie został wlutowany bezpośrednio w płytę bloku sterowania, lecz umieszczony w podstawce ze względu na konieczność wyjmowania go z płyty celem umieszczenia go w programatorze. Układ MAX232 także został umieszczony w podstawce, celem zabezpieczenia go przed przypadkowym zniszczeniem podczas lutowania.

4.2 Opis oprogramowania zegara mikroprocesorowego

Program zapisany w pamięci mikrokontrolera AT89C2051 będącego częścią układu zegara napisany został w języku Asembler. Ze względu na ograniczone możliwości mikrokontrolera program spełnia jedynie proste funkcje:

- Bezzwłocznie pobiera dane synchronizacyjne pojawiające się na porcie COM komputera i zapisuje je do odpowiednich rejestrów mikrokontrolera.
- Odlicza upływający czas na podstawie przerwań pochodzących od wewnętrznego licznika i cyklicznie aktualizuje rejestry czasu.
- Dekoduje cyfry zapisane w rejestrach czasu w postaci binarnej na postać 7 segmentową umożliwiając ich odczyt przez obserwatora na wyświetlaczu.
- Obsługuje prace wyświetlacza.

Program nie implementuje w żaden sposób mechanizmów obliczania opóźnienia synchronizacji. Zatem implementacja protokołu NTP kończy się na komputerze PC.

Rozwiązanie to jest w zupełności wystarczające ponieważ opóźnienia po stronie mikrokontrolera są znane, stałe i bardzo małe (kilka mikrosekund).

Program wykorzystuje 7 rejestrów jednobajtowych (R0 do R6) do zapisu aktualnego czasu. Napisany jest w postaci nieskończonej pętli obsługującej dekodowanie cyfr zawartych w rejestrach na postać przydatną do bezpośredniego podania na wyprowadzenia wyświetlaczy oraz obsługującej pracę wyświetlacza. Dekodowanie zawartości rejestrów polega na tym, że kody symbolizujące cyfry 0 – 9 na wyświetlaczach zapisane są w pamięci programu przy czym wybierany jest jeden z nich zgodnie z zawartością aktualnie wyświetlanego rejestru czasu. Następnie podawany jest na port P1. Wyświetlacze odświeżane są z częstotliwością 500Hz i stosunkiem czasu świecenia do wygaszenia równym 1/8. Stosunek czasu świecenia do wygaszenia można regulować w programie w celu zwiększenia lub zmniejszenia jasności wyświetlaczy. Częstotliwość odświeżania można regulować zmieniając ilość skoków w pętli opóźniającej. W pętli głównej występują dwa źródła przerwań:

– Przerwanie pochodzące od Timera 0 zgodnie z którym aktualizowany jest czas

– Przerwanie związane z pojawieniem się danych w buforze portu szeregowego

Przerwanie pochodzące od portu szeregowego ma wyższy priorytet niż przerwanie pochodzące od Timera 0, ponieważ dane synchronizacyjne pojawiające się na porcie szeregowym muszą być wpisane do rejestru bez zwłoki. W podprogamie obsługi przerwania portu szeregowego poszczególne cyfry czasu przychodzące z komputera dekodowane są z postaci ASCII na naturalne liczby binarne. Ponieważ przerwania z portu szeregowego pojawiają się o pełnych sekundach to Timer 0 i rejestr pomocniczy r0 są zerowane. Timer 0 jest uruchamiany ponownie. Komputer i zegar komunikują się ze sobą z prędkością 1200 bodów na sekundę w trybie asynchronicznym zatem wykorzystywana jest tylko linia RxD (wyprowadzenie p3.0). Prędkość transmisji po stronie mikrokontrolera jest regulowana częstością przepełnień licznika T1 pracującego w trybie 2 i wynosi:

$$V = \frac{f_{XTAL}}{12 * 32 * 2^{SMOD} * (256 - TH1)} \tag{4.1}$$

Gdzie:

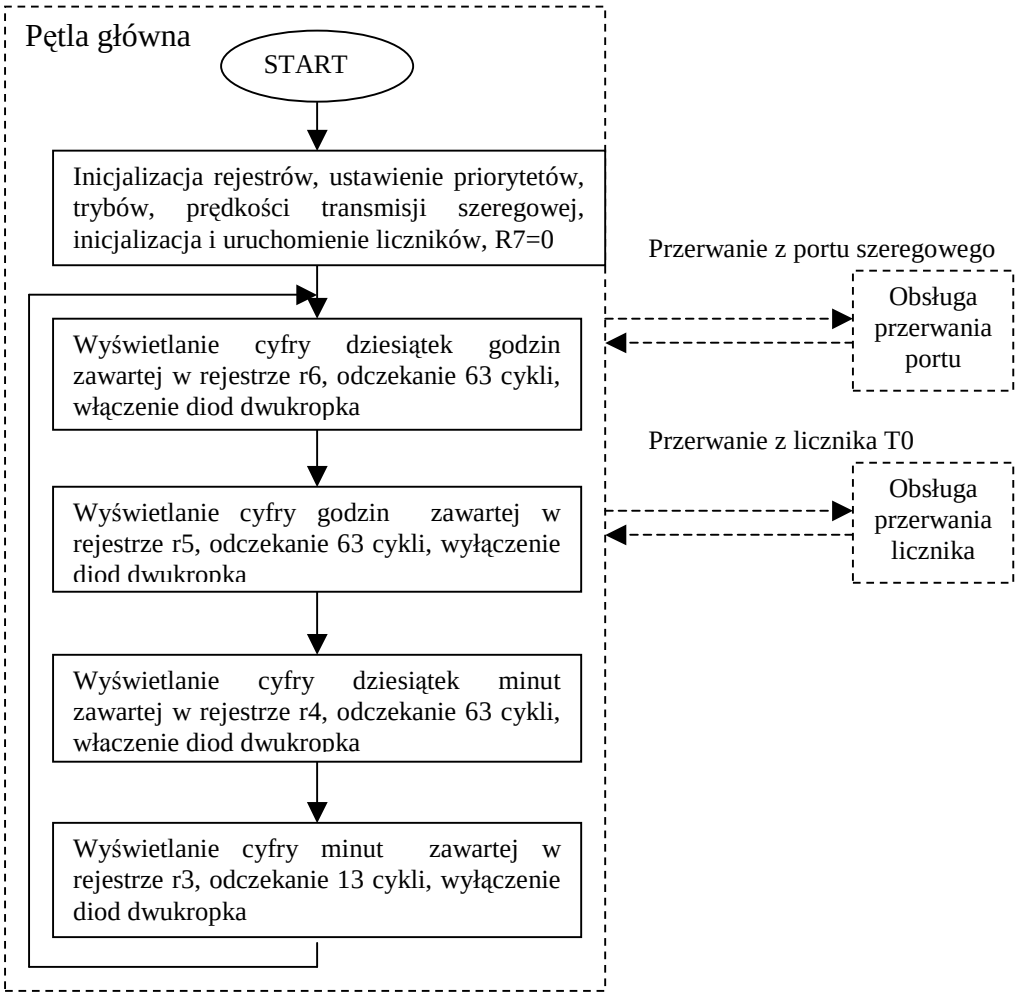
V – prędkość transmisji w bodach na sekundę

f_{XTAL} - częstotliwość kwarcu, w danym przypadku 12Mhz

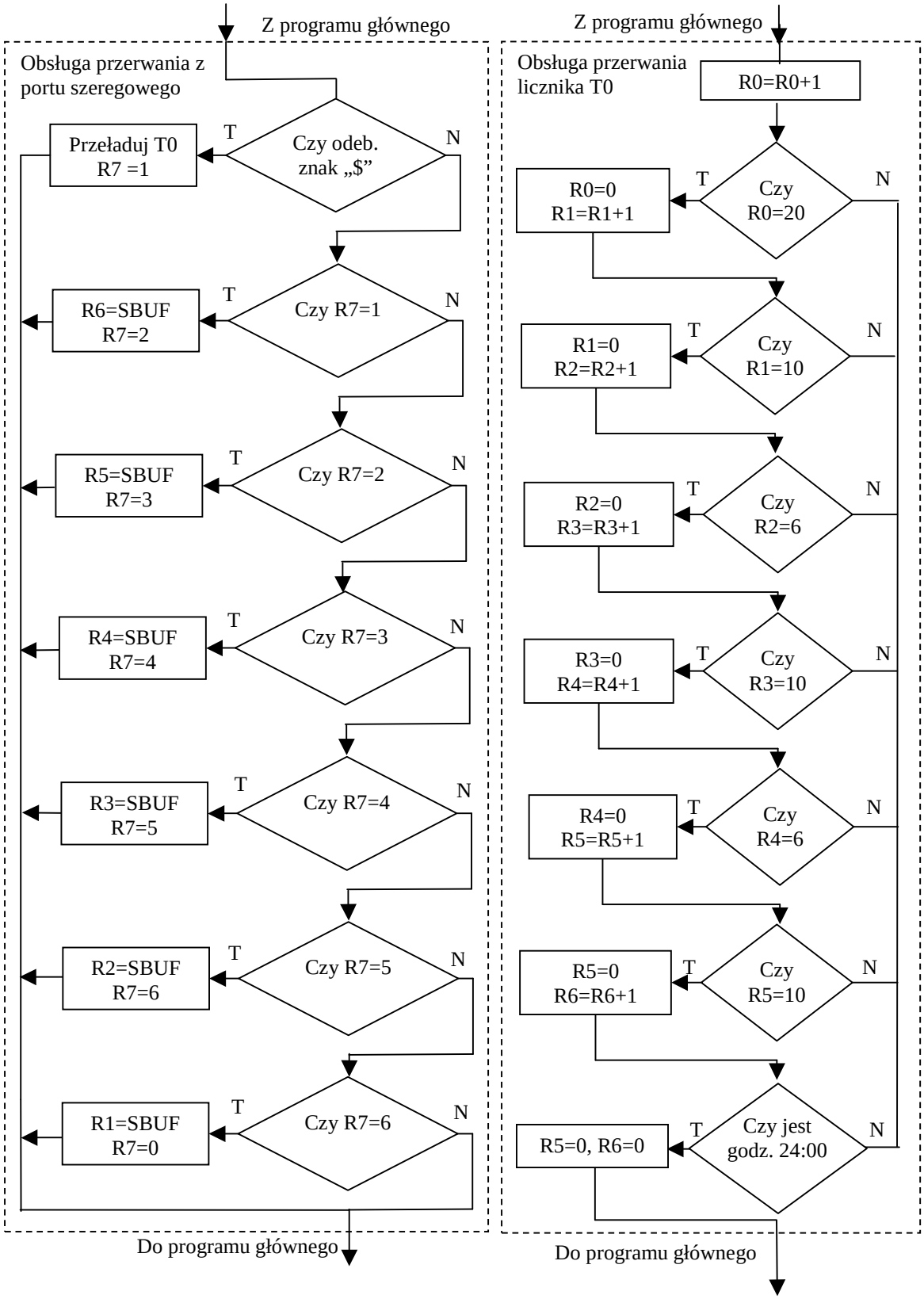
TH1 – starszy bajt licznika T1

W mianowniku występuje liczba 12 ponieważ na tyle okresów drgań kwarcu przypada jeden cykl maszynowy mikrokontrolera.

Przerwanie pochodzące od Timera 0 pracującego w 1 trybie (licznik 16 bitowy bez przeładowania) pojawia się co 50ms. Podczas każdego przerwania przeładowywany jest Timer 0, oraz inkrementowana jest zawartość rejestru pomocniczego r0. Po osiągnięciu przez rejestr pomocniczy wartości 20 jest on zerowany i inkrementowany jest r1 w którym zapisane są sekundy. Analogicznie dzieje się w przypadku osiągnięcia przez rejestr sekund wartości 9. Wtedy to rejestr ten jest zerowany a inkrementowany jest rejestr r2 zawierający ilość dziesiątek sekund, itd. aż do rejestru r6 w którym przechowywana jest liczba dziesiątek godzin. Schemat blokowy programu mikroprocesora został przedstawiony na rysunkach 4.2 oraz 4.3. Na schematach strzałki oznaczają przepływ sterowania, prostokąty oznaczają wykonywanie operacji w nich zawartych natomiast romby oznaczają podejmowanie decyzji na temat wyrażeń w nich zawartych. Na rysunkach rejestry oznaczane są literami R.



Rys. 4.2 Schemat blokowy działania programu zegara mikroprocesorowego - pętla główna.



Rys. 4.3 Schemat blokowy działania programu zegara mikroprocesorowego – obsługa przerwania z portu szeregowego (z lewej), oraz obsługa przerwania z licznika T0 (z prawej)

Przerwania pochodzące z portu szeregowego oraz z licznika T0 mogą być wywołane w dowolnym momencie wykonywania programu, dlatego na schemacie z rys. 4.2 strzałki oznaczające przepływ sterownia do i z podprogramów przerwań nie prowadzą do konkretnych miejsc pętli głównej. Schematy blokowe podprogramów przerwań przedstawione są na rys. 4.3.

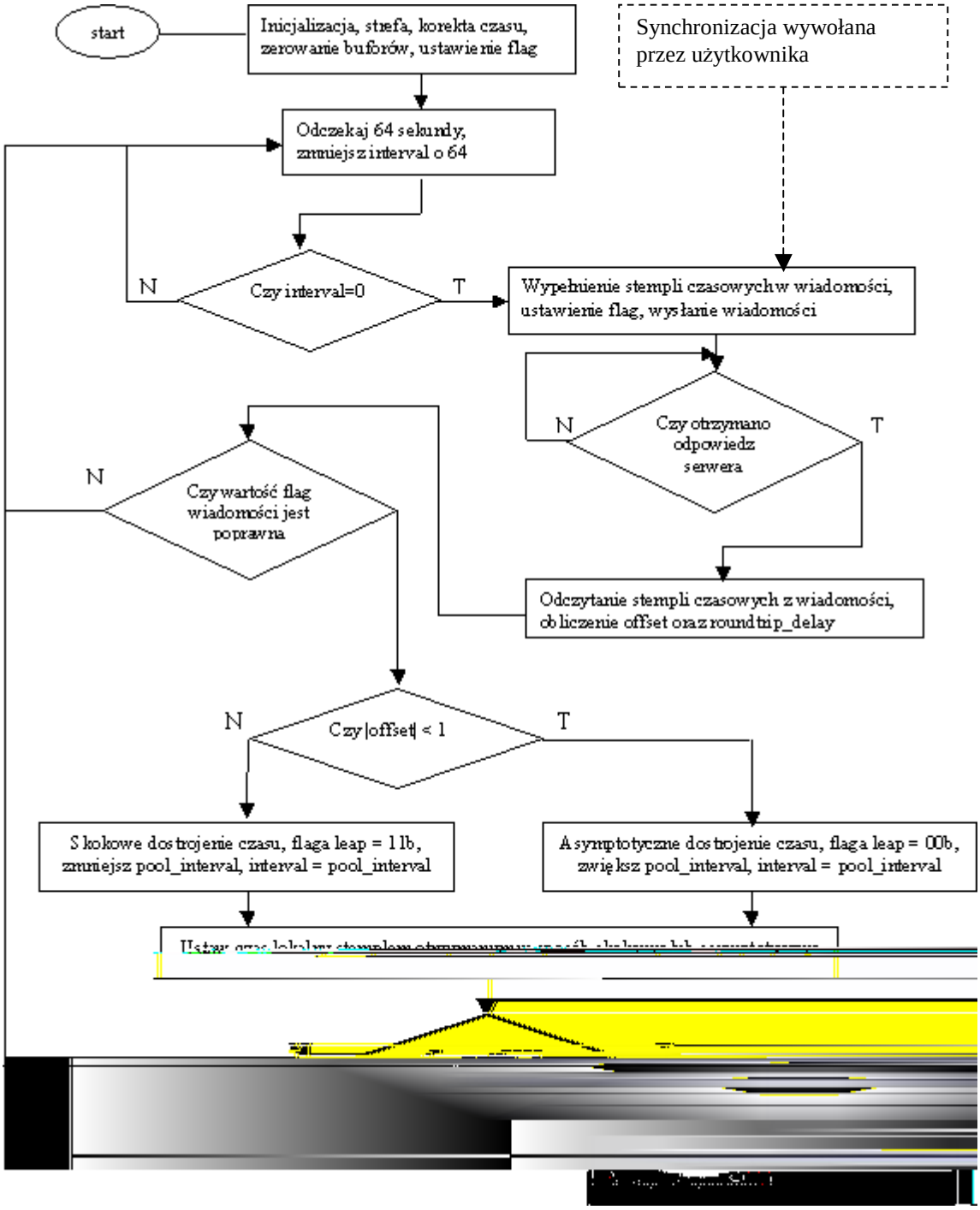
4.3 Opis techniczny oprogramowania komputera PC

Oprogramowanie komputera PC tworzone na potrzeby niniejszego projektu napisane zostało za pomocą zintegrowanego środowiska programistycznego Builder C++ 6.0 firmy Borland. Stworzone zostało dla pracy w systemach operacyjnych: Windows 98, Windows 2000, Windows XP firmy Microsoft. Schemat blokowy programu synchronizującego uruchomionego na komputerze PC został przedstawiony na rysunku 4.4 Na schemacie strzałki oznaczają przepływ sterowania, prostokąty oznaczają wykonywanie operacji w nich zawartych natomiast romby oznaczają podejmowanie decyzji na temat wyrażeń w nich zawartych. Na schemacie przedstawiona została jedynie główna pętla programu. Pozostałe części programu sprowadzają się do prostych, mniej znaczących procedur wywoływanych głownie przez użytkownika. Ich działanie nie jest przedstawione na schemacie blokowym celem ułatwienia zrozumienia istoty działania programu.

Pod względem funkcjonalnym program podzielić można na dwie różne części. Część pierwsza odpowiedzialna jest za obsługę połączenia z wybranym serwerem NTP. W zakres jej zadań wchodzi ustanawianie połączeń, rozłączanie połączeń, oraz wysyłanie i odczytywanie danych otrzymywanych z serwera, pojawiających się na odpowiednim porcie. Kolejnym zadaniem tej części programu jest współpraca z użytkownikiem (za pomocą interfejsu graficznego) celem wyboru odpowiedniego dla jego potrzeb serwera czasu oraz informowanie go o zdarzeniach związanych z połączeniem (czas ostatniej poprawnej synchronizacji oraz status). Należy zwrócić uwagę na fakt, że korekty czasu na podstawie wspomnianego algorytmu dokonuje się jeszcze na komputerze PC i nie uwzględnia ona opóźnienia związanego z wysłaniem danych poprzez port COM do pamięci zegara.

Informacje pomiędzy urządzeniami w sieci wymieniane są za pomocą datagramów UDP o stałej długości. Pole danych datagramu (Wiadomości NTP) ma

także stały rozmiar 48 bajtów . Pozwala to na przechowywanie pochodzących z nich danych w postaci tablicy znaków.



Rys. 4.4 Schemat blokowy programu synchronizującego uruchomionego na komputerze PC

Program napisany został z wykorzystaniem klas wbudowanych w kompilator, dzięki czemu w pewnym stopniu zmniejszyła się ilość kodu napisana przez jego autora. Główna część programu zawarta została w pliku Unit1.cpp w czterech podstawowych

funkcjach. Najważniejsza z nich Button1Click (TObject *Sender) odpowiada za wysłanie i odbieranie wiadomości z serwera, za konwersję danych pomiędzy rożnymi formatami. oraz za obliczanie wartości aktualizacji zegara. Kolejna funkcja - Button2Click(TObject *Sender) wywołuje funkcje Button1Click (TObject *Sender) celem otrzymania danych do aktualizacji zegara, a następnie aktualizuje zegar systemowy otrzymanym stemplem oraz wysyła przez port COM dane do synchronizacji zegara cyfrowego, jeżeli jest on przyłączony. Funkcja Button2Click(TObject *Sender) może być wywołana przez użytkownika po naciśnięciu przycisku „Synchronizuj teraz". Funkcja Button3Click(TObject *Sender) odpowiada za zmianę źródła synchronizacji na wybrane przez użytkownika. Funkcja Timer1Timer(TObject *Sender) odmierza czas pomiędzy kolejnymi synchronizacjami i wywołuje je (wywołuje funkcje Button2Click(TObject *Sender)).

Dane pochodzące bezpośrednio z wiadomości NTP przechowywane są w postaci tablic o rozmiarze 48 natomiast stemple czasowe w postaci liczb całkowitych w następujący sposób:

```
unsigned char buffer_out[48];     // tablica zawierająca treść wiadomości wychodzącej
unsigned char buffer_in[48];      // tablica zawierająca treść wiadomości przychodzącej
unsigned int recive_timestamp=0;        // stempel czasu odbioru
unsigned int transmit_timestamp=0;      // stempel czasu transmisji wiad. przez serwer
unsigned int originate_timestamp=0;     // stempel czasu transmisji zapytania przez
                                        // klienta
unsigned int reference_timestamp=0;  // stempel czasu ostatniej synchronizacji klienta
```

Stemple czasowe zapisane są w postaci zgodnej ze skalą NTP. Aby je wykorzystać do aktualizacji zegara systemowego niezbędna jest ich konwersja. Najpierw zawartość stempli konwertowana jest z postaci zgodnej ze skalą NTP do postaci zgodnej ze skalą Unix. Skale wyrażone są w tych samych jednostkach, tylko ich początki są przesunięte względem siebie – początek skali Unix datuje się na 2208988800s później niż początek skali NTP dlatego od stempla należy odjąć tą wartość. W dalszej kolejności program uwzględnia przesuniecie związane ze strefą czasową oraz z czasem letnim, lub zimowym. Wartości te ustawiane są automatycznie bez ingerencji użytkownika. W tym celu program wykorzystuje zmienne kalendarza systemu Windows i oblicza korektę w następujący sposób:

```
strefa = _timezone;              //odczytanie strefy z kalendarza syst.
czas_letni =_daylight;           //odczytanie strefy z kalendarza syst.
```

```
korekta = czas_letni*3600-strefa;    //obliczanie korekty czasu
```

Korekta wyrażona jest w sekundach i dodawana do stempla czasowego.

Następnie za pomocą wbudowanej funkcji stempel konwertowany jest do postaci akceptowalnej przez system operacyjny:

```
czas=UnixToDateTime(present_time-2208988800+korekta);
```

Parametry oraz zmienne wykorzystywane w programie są zgodne tymi przedstawionymi w treści standardu NTP (posiadają także takie same nazwy anglojęzyczne). Są opisane w rozdziale 4.1 dlatego nie będą omawiane ponownie.

Wiadomości NTP nie zawierają danych w postaci znakowej, dlatego potrzebna jest konwersja stempli czasowych na postać liczbową. Nagłówek wiadomości nie jest konwertowany. W dalszej części pracy w celu ułatwienia liczby binarne są oznaczane literą „b" występująca zaraz za liczbą. W celu ustawienia odpowiednich flag wykuje się na nim operacje bitowe. Np.: Pierwszy bajt wiadomości zawiera: 2 bitowe pole Leap, (wartość 11b przed pierwszą synchronizacją) 3 bitowe pole Version, (dla programu zgodnego z wersją NTPv3 pole ma wartość 011b) oraz 3 bitowe pole Mode (dla programu pracującego w trybie klienta pole ma wartość 011b). Zatem wykorzystując notację szesnastkową flagi ustawiane są w następujący sposób:

```
buffer_out[0]=0xdb;                // ustawienie flag
```

Powyższa operacja wraz z obliczaniem korekty czasu jest wykonywana przy uruchomieniu programu. Jak wspomniano najważniejszą funkcją programu jest Button1Click(TObject *Sender). Po jej uruchomieniu najpierw wysyłana jest wiadomość do serwera NTP. W tym celu do tablicy wysyłanej wiadomości najpierw wpisywane są odpowiednie flagi oraz aktualne stemple czasowe. Stemple czasowe zostają przekonwertowane z postaci liczbowej wykorzystywanej w programie do postaci szesnastkowej występującej w wiadomości poprzez cykliczną operację dzielenia oraz modulo 16 w następujący sposób:

```
for (int i=19; i>=16; i--)        //czas ostatniej aktualizacji
{
do_wpisania = fmod ( reference_timestamp, 0x100 );
reference_timestamp = reference_timestamp /  0x100;
buffer_out[i] = do_wpisania;
}
```

Gdzie: do_wpisania oznacza bajt przekonwertowanych danych kopiowanych do treści wiadomości.

Gotowa wiadomość wysyłana jest do serwera za pomocą wbudowanych w kompilator mechanizmów:

```
IdUDPClient1- > SendBuffer ( buffer_out, 48 );
```

Następnie system oczekuje przez zadany czas (w tym przypadku 10000ms) na odpowiedz z serwera:

```
IdUDPClient1- > ReceiveBuffer ( buffer_in,48, 10000 );
```

Po otrzymaniu odpowiedzi z serwera treść wiadomości jest zapisywana w postaci tablicy numerycznej buffer_in[48]. Następnie przetwarzany jest nagłówek i pole wiadomości. Stemple transmit_timestamp, originate_timestamp, recive_timestamp są wyodrębniane z tabeli buffer_in[48]. Aby łatwiej zrozumieć w jakich miejscach tabeli buffer_in[48] odczytywać poszukiwane dane (w tym przypadku stemple czasowe) należy ułożyć kolejne wiersze ramki z rys. 3.2 jeden za drugim, a następnie ponumerować je poczynając od 0. Wtedy numer danego bajtu w ramce odpowiada numerowi pola tabeli. Tym sposobem stemple odczytywane są z tabeli według następującego schematu:

```
transmit_timestamp=0;
for (int i=40; i<=43; i++) {
transmit_timestamp=transmit_timestamp*0x100+buffer_in[i]; }
```

Mnożenie przez 100 (w zapisie o podstawie szesnastkowej) ma na celu przesuniecie bitowe zmiennej transmit_timestamp o bajt w kierunku najbardziej znaczącego bitu.

W dalszej kolejności program oblicza przesuniecie względne zegarów (offset) oraz opóźnienie transmisji (roundtrip_delay) zgodnie z wzorami 3.1 oraz 3.2 w następujący sposób:

```
Offset = ( ( recive_timestamp - originate_timestamp ) + ( transmit_timestamp - ( DateTimeToUnix ( Now () ) + 2208988800 - korekta ) ) ) / 2 ;
roundtrip_delay = ( DateTimeToUnix ( Now() ) + 2208988800 - korekta ) - originate_timestamp - transmit_timestamp + recive_timestamp;
```

W dalszej kolejności sprawdzany jest nagłówek wiadomości celem określenia przydatności otrzymanych danych. Celem wyodrębnienia odpowiednich flag program wykonuje operacje bitowe na odpowiednich komórkach tablicy buffer_in. Np. w celu sprawdzenia zgodności wersji i trybu pracy sprawdzana jest poprawność wartości pierwszego bajtu wiadomości za pomocą porównania:

```
If ( (buffer_in[0]&0x3f ) = = 0x1c )
```

Jeżeli wymieniona zależność nie zostanie spełniona program informuje użytkownika o błędzie.

Poprawność i integralność danych są zapewniane dzięki mechanizmom IP/UDP i nie muszą być sprawdzane w programie. Wartość liczby Stratum nie jest sprawdzana. Przy pierwszej synchronizacji liczba stratum hosta jest niewyspecyfikowana. Po pierwszej synchronizacji program przyjmuje liczbę Stratum (pole buffer_out[1]) większą o jeden od liczby *stratum* źródła synchronizacji (pole buffer_in[1]).

W dalszej kolejności wewnątrz funkcji Button1Click podejmowana jest decyzja co do rodzaju aktualizacji zegara. Jeżeli przesuniecie zegarów jest większe niż 1s następuje skokowe dostrojenie zegara systemowego – zostaje mu przypisana wartość stempla czasowego transmit_timestamp powiększonego o czas transmisji wiadomości przez sieć (roundtrip_delay / 2). W dalszej kolejności polu *leap* przypisuje się wartość 11b (niezsynchronizowany) oraz zmniejszana jest wartość okresu synchronizacji ze źródłem. Okres synchronizacji jest zmieniany w programie ze skokiem 64s między wartościami 64s a 1024s. Wartość okresu synchronizacji zapisana jest w zmiennej pool_interval. Jeżeli przesuniecie zegara lokalnego względem zegara odniesienia jest mniejsze bądź równe 1s następuje płynne dostrojenie zegara systemowego proporcjonalnie do wartości względnego przesunięcia.

W systemie Windows dostęp do dokładnego zegara o rozdzielczości milisekundowej jest utrudniony. Zegary programowe są mało dokładne i przy zwiększonym obciążeniu komputera spóźniają się. Dostęp do zegara sprzętowego wiąże się z częstym programowym sczytywaniem wartości licznika co znacznie obciąża komputer. Zdecydowano się więc na prostszą implementację zegara lokalnego opisanego w standardzie – stanowi go sam zegar systemowy Windows. Dokładność zapewniona przez zegar systemowy jest w zupełności wystarczająca, ponieważ osoba spoglądająca na zegar nie jest w stanie odczytać czasu z wyższą dokładnością.

Po obliczeniu aktualnego czasu (w postaci stempla NTP) jest on konwertowany na postać systemową i zapisywany w zmiennej czas. Po wykonaniu powyższej operacji funkcja Button1Click(TObject *Sender) jest opuszczana.

Funkcja Button2Click(TObject *Sender) wywołuje funcję Button1Click(TObject *Sender). W dalszej kolejności Button2Click(TObject *Sender) pobiera stempel czasowy obliczony w Button1Click(TObject *Sender), za jego pomocą aktualizuje czas systemowy i jeżeli uruchomiona jest transmisja przez port COM, funkcja wysyła stempel czasowy do zegara mikroprocesorowego. W tym celu za pomocą wbudowanej

w kompilator funkcji aktualny czas jest konwertowany na postać znakową i zapisywany w zmiennej napis:

```
String napis = TimeToStr ( Now() );
```

Czas w postaci znakowej zapisany jest w konwencji gg:mm:ss, gdzie: gg oznacza liczbę godzin, mm oznacza liczbę minut, ss oznacza liczbę sekund aktualnego stempla. Dalej program traktuje zmienną napis jak tablicę i odnosząc się do jej pól o numerach 1, 2, 4, 5, 7, 8 konwertuje stempel do postaci ggmmss. Kolejne cyfry zapisywane są w tablicy TableOfTimew następujący sposób:

```
TableOfTime[i] = napis[i];
```

Do pola numer 0 tablicy TableOfTime wpisywany jest znak '$' (jest to znacznik momentu synchronizacji dla zegara mikroprocesorowego) po czym tablica jest wysyłana wprost przez port.

Część druga programu odpowiedzialna jest za współpracę komputera z urządzeniem peryferyjnym (jakim jest zegar będący przedmiotem projektu) poprzez wybrany port COM. Część ta odpowiedzialna jest za otwarcie i wysyłanie danych przez port oraz za współpracę z użytkownikiem (poprzez interfejs graficzny) mającą na celu umożliwienie mu dokonania wyboru odpowiednich parametrów transmisji szeregowej.

Do obsługi portu COM wykorzystane zostały biblioteki Win32 API. Dzięki temu kod programu został znacznie uproszczony. Program obsługi portu COM znajduje się w pliku Unit4.cpp Wykorzystuje klasę Tserial_event zdefiniowaną na potrzeby komunikacji z portem szeregowym. Wewnątrz programu tworzony jest nowy obiekt klasy Tserial_event. Następnie ustawiane są domyślne parametru transmisji:

- Prędkość transmisji: 300b/s (zmienna rate = 300)
- Port: COM1 (zmienna port_name = "COM1")
- Kontrola parzystości: brak. (zmienna partity = 0)

Najważniejszą rolę w komunikacji z portem odgrywają funkcje TForm4::Button1Click(TObject *Sender) oraz SerialEventManager(uint32 object, uint32 event).

Funkcja TForm4::Button1Click(TObject *Sender) odpowiada za zamykanie i otwieranie portu z zadanymi wcześniej parametrami transmisji. W tym celu funkcja najpierw sprawdza czy port został już wcześniej otwarty . Jeżeli tak, (flaga connected = true) niszczony jest obiekt klasy obsługującej to połączenie ponieważ jej parametry nie mogą być zmienione podczas otwarcia portu. Tworzony jest nowy „pusty" obiekt klasy Tserial_event.

```
Form4-> com = new Tserial_event();
```

Na zakończenie tworzony jest nowy obiekt klasy Tserial_event z zadanymi przez użytkownikami parametrami:

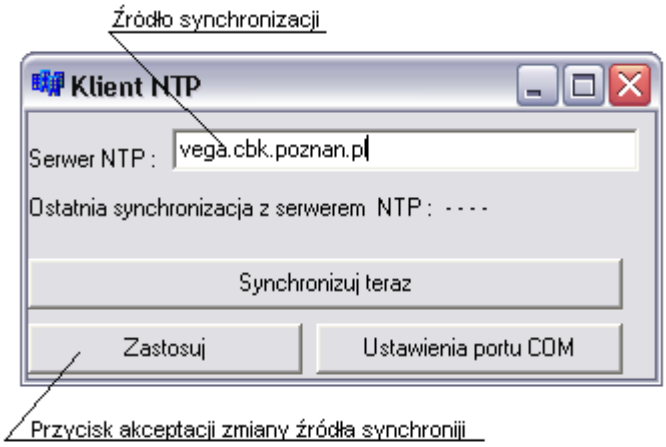
```
erreur = Form4-> com-> connect ( port_name, rate, partity, 8, true );
```

Zmienna erreur używana jest do wykrywania błędów otwarcia portu. Przyjmuje wartość 1 (błąd) jeżeli port nie istnieje, bądź jest wykorzystywany przez inny program. Funkcja SerialEventManager(uint32 object, uint32 event) odpowiada za wykrywanie błędów związanych z połączeniem. Funkcja pobiera zmienną systemową uint32 event. Na podstawie wartości zmiennej event określa stan portu i ustawia flagę connected. Dla transmisji jednokierunkowej bez kontroli przepływu zmienna event może przyjmować wartości: SERIAL_CONNECTED (połaczono), SERIAL_DISCONNECTED (rozłączono).

Za zmianę parametrów transmisji odpowiadają funkcje ComboBox1Change(TObject *Sender), ComboBox2Change(TObject *Sender), ComboBox3Change(TObject *Sender) zawarte w pliku Unit4.cpp. Wewnątrz wymienionych funkcji do zmiennych rate, port_name, partity wpisywane są wartości wybrane przez użytkownika z list rozwijanych.

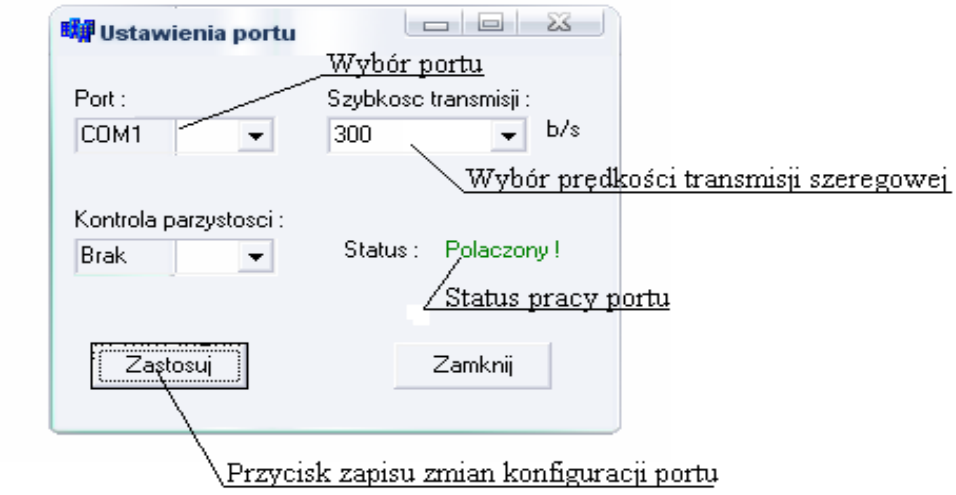
4.4 Instrukcja obsługi oprogramowania komputera

Program „Klient NTP” nie wymaga instalacji w systemie. Wystarczy skopiować go do wybranej lokalizacji i uruchomić. W celu włączenia automatycznego wczytania programu wraz z uruchomieniem systemu operacyjnego najłatwiej jest utworzyć nowy skrót do programu (klikając prawym przyciskiem myszy na ikonie programu i wybierając z menu podręcznego opcję „Utwórz skrót”) a następnie kopiując tak utworzony skrót do katalogu „Autostart ” w Menu „Start” (Start -> Programy -> Autostart). Po uruchomieniu programu ukazuje się główne okno programu pokazane na 4.2. Funkcje elementów okna są opisane na rysunku. Program nie posiada menu ustawień strefy czasowej oraz zmiany czasu z zimowego na letni i odwrotnie, ponieważ dokonywana jest ona wewnątrz programu automatycznie. Po uruchomieniu program automatycznie łączy się z domyślnym serwerem czasu vega.cbk.poznan.pl który został wybrany z powodu najniższych opóźnień komunikacji (szczególnie jeśli klient będzie zlokalizowany w Poznaniu).



Rys.4.2 Okno główne programu

W celu zmiany parametrów transmisji szeregowej należy nacisnąć przycisk „Ustawienia portu COM”. Aby program mógł komunikować się poprawnie z zegarem mikroprocesorowym w oknie „Ustawienia portu” należy wybrać port do którego podłączony jest zegar. Prędkość transmisji należy ustawić na 1200b/s. Po wybraniu odpowiednich parametrów transmisji należy nacisnąć przycisk „Zastosuj” celem uruchomienia transmisji. W przypadku błędu transmisji użytkownik zostanie o nim poinformowany.



Rys.4.3 Okno menu wyboru

5. Literatura

[1] „Sygnał czasu” Andrzej Dobrogowski Wydawnictwo Politechniki Poznańskiej, Poznań 2003

[2] „Czas, przestrzeń, rzeczy ” B.K. Ridley Wiedza powszechna, Warszawa 1987

[3] „Częstotliwość i czas ” Peter Kartaschoff WKŁ, Warszawa 1985

[4] „Synchronizacja zegarów telekomunikacyjnych sieci cyfrowej z warunkowym uzależnieniem dwustronnym sygnałów sterowania”, Andrzej Dobrogowski, Wydawnictwo Politechniki Poznańskiej, Poznań 1982

[5] „Einstein dla początkujących” Joseph Schwartz, Michael McGuinnes, ALFA 1989

[6] ETSI EN 300 462-1-1 Definitions and terminology for synchronization networks

[7] http://www.edu.vernet.pl/kartografia/strona/pc.html

[8] http://eduseek.interklasa.pl/artykuly/artykul/ida/2694/

[9] http://astro.uni.torun.pl/~kb/artykuly/U-PA/Czas2.htm

[10] http://www.meyersusa.com/tato_374/

[11] http://www.wynalazki.mt.com.pl/wyn/zmechan.html

[12] http://wwwnt.if.pwr.wroc.pl/kwazar/jaktopracuje/135500/index.htm

[13] http://www.eres.alpha.pl/elektronika/readarticle.php?article_id=318

[14] http://www.elb.vectranet.pl/~krzysztofg/projekty/generatory_kwarcowe.htm

[15]http://wazniak.mimuw.edu.pl/index.php?title=Uk%C5%82ady_elektroniczne_i_technika_pomiarowa/Modu%C5%82_6 – materiały dydaktyczne polskich uczelni w sieci.

[16] http://tf.nist.gov/general/enc-re.htm - oficjalna strona NIST

[17] http://www.lamptech.co.uk/Spec%20Sheets/Osram%20Spectral%20Rb.htm - strona producenta lamp rubidowych

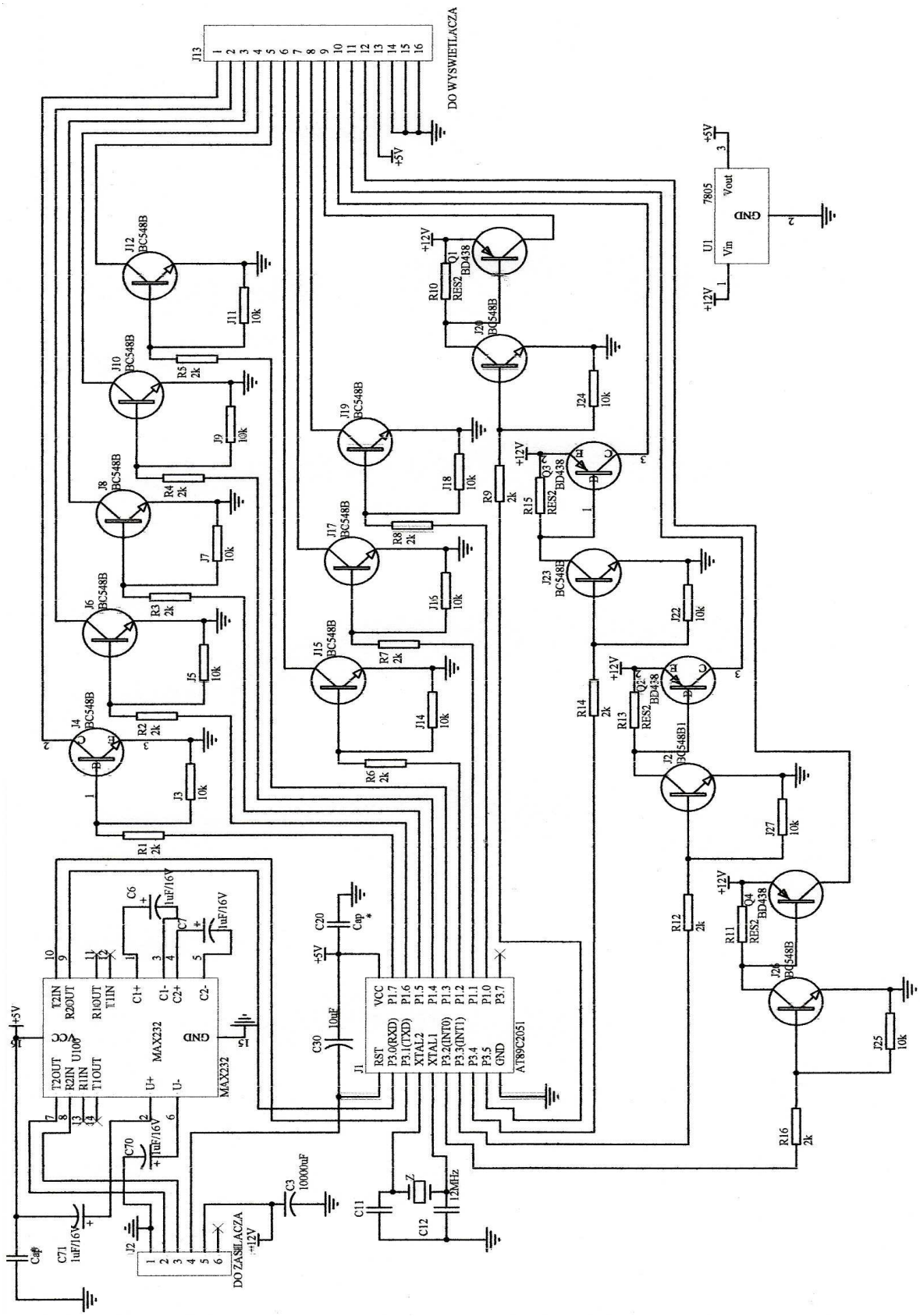
[18] http://www.ptb.de/en/org/4/44/441/_index.htm- oficjalna strona grupy roboczej.

[19] http://www.meratronik.pl/ -oficjalna strona polskiego producenta aparatury i wzorców pomiarowych.

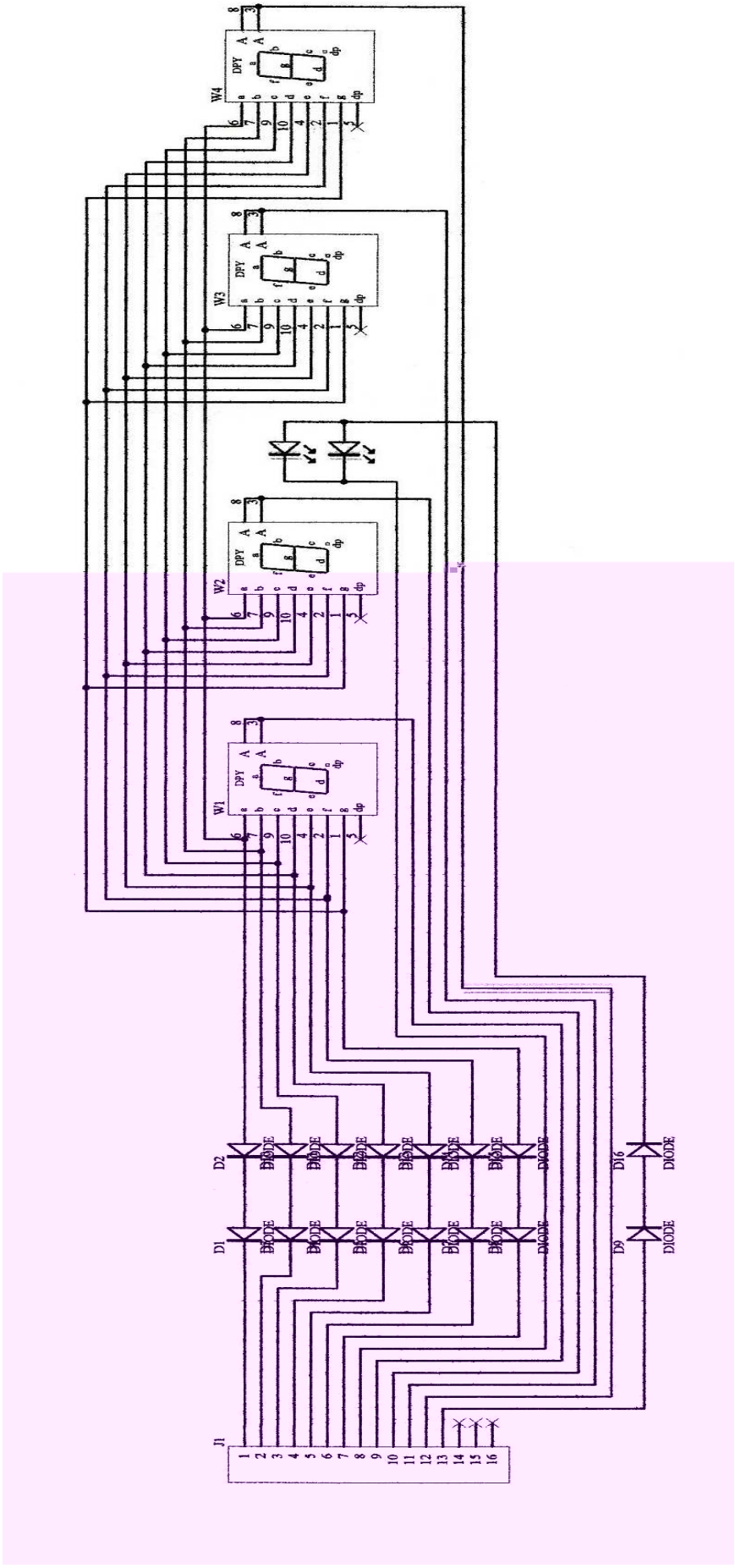
[20] Cesium Beam Atomic Time and Frequency Standards ; National Bureau of Standards, Boulder, Colorado; R. E. BEEHLER, R. C. MOCKLER, and J. M. RICHARDSON

[21] 5071A Primary Frequency Standard ; SYMMETRICOM, INC.
[22] The Atomic Hydrogen Maser, Lyman Physics Laboratory Harvard University, Cambridge, Massachusetts
NORMANF. RAMSEY; October 27,1964)
[23] Cesium Clock and Hydrogen Maser Compared ; Application Note; Oscilloquartz S.A.
[24] Primary Atomic Frequency Standards at NIST; Journal of Research of the National Institute of Standards and TechnologyJanuary–February 2001
[25] Optyczne zegary atomowe, materiały dydaktyczne Uniwersytetu Jagiellońskiego
[26] http://interfacebus.engineer.googlepages.com/atomic_clock_standards
[27] http://www.nauticalissues.com/astronomy6.html - Wyznaczanie współrzędnych na podstawie konstelacji satelitów GPS, materiały dydaktyczne Uniwersytetu Adama Mickiewicza.
[28] www.fairchildsemi.com/ds/**1N**%2F**1N**4004.pdf - specyfikacja elementu 1N4001
[29] www.fairchildsemi.com/ds/BD/BD434.pdf - specyfikacja elementu BD434
[30] www.100y.com.tw/pdf_file/AN7800.pdf - specyfikacja elementów AN7805 / AN7809
[31] www.philohome.com/sensors/gp2d12/gp2d12-datasheets/bc548.pdf - specyfikacja elementu BC548
[32] www.atmel.com/atmel/acrobat/doc0368.pdf - specyfikacja elementu AT89C2051
[33] www.datasheet4u.com/html/M/A/X/MAX232_Maxim.pdf.html - specyfikacja elementu MAX232
[34] SA40-19.pdf – opis elementu SA40-19SRWA
[35] AN7810.pdf, AN7805.pdf
[36] Dokument RFC 1305 – specyfikacja protokołu NTP

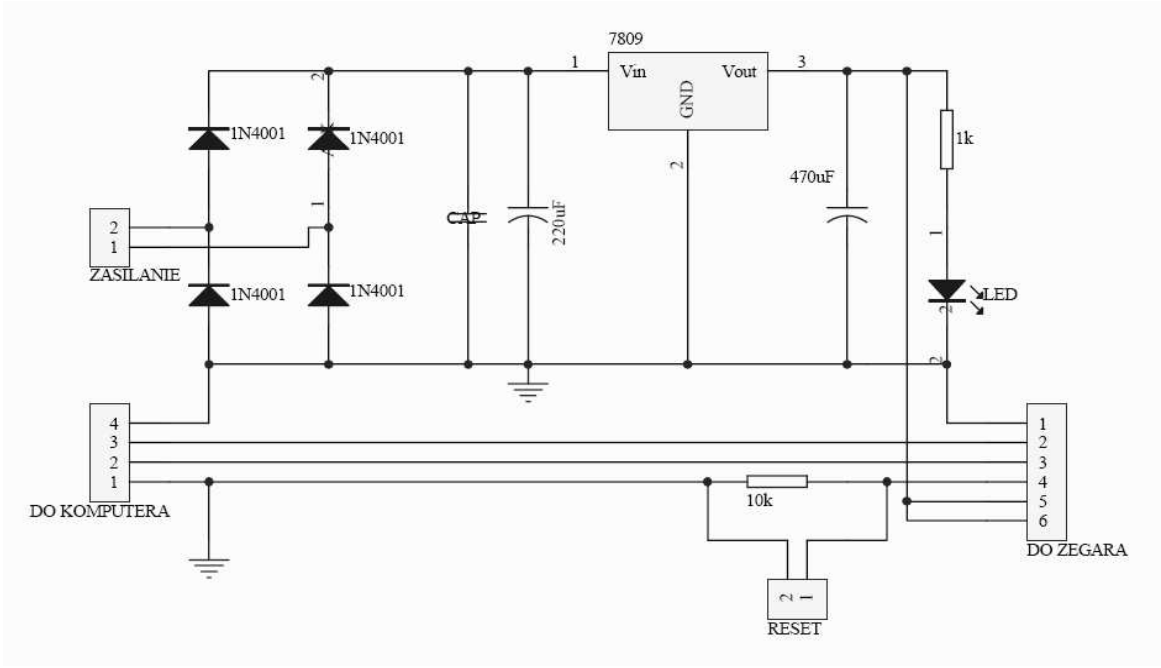
6. Załączniki



Załącznik A1. Schemat sterowania zegarem



Załącznik A2. Schemat obwodów bloku wyświetlaczy



Załącznik A3. Schemat obwodów bloku zasilania