

Uniwersytet im Adama Mickiewicza
W Poznaniu

Wydział Fizyki

Praca magisterska

„Projekt cyfrowej deski rozdzielczej samochodu, z
wykorzystaniem układu FPGA”

Przemysław Witczak

Kierownik pracy:
Dr Zdzisław Stryła
Zakład Elektroniki Kwantowej, Wydział Fizyki UAM

Poznań 2008

Spis treści

Spis treści.....	2
Spis ilustracji.....	3
Wstęp.....	5
Rozdział I - Charakterystyka elementów.....	7
Układ FPGA – Spartan 3 XC3S200, Xilinx.....	7
Minimoduł MMfpga12, Propox.....	12
Pamięć FLASH M28W640, STM.....	13
Magistrala 1 Wire [®] , Maxim/Dallas Semiconductor.....	14
1. Czujnik temperatury DS18B20.....	16
2. Czterokanałowy przetwornik analogowo cyfrowy DS2450.....	16
Wyświetlacz LCD HY1602F, ze sterownikiem Hitachi HD44780U.....	18
Rozdział II - Opis systemu.....	20
Schemat blokowy.....	20
Opis poszczególnych elementów.....	27
Licznik prędkości i obrotomierz.....	27
Pomiar temperatury i wartości analogowych.....	33
Sterownik wyświetlacza LCD.....	47
Liczniki kilometrów.....	52
Multiplexer magistrali danych.....	62
Zegar.....	63
Sterownik wycieraczek.....	64
Rozdział III - Dokumentacja techniczna.....	67
Wprowadzenie.....	67
Schematy elektryczne.....	67
Zasilanie, przyciski, wyświetlacze siedmiosegmentowe led.....	67
LCD, 1 Wire [®]	69
Wejścia i wyjścia analogowe (transoptory i przekaźniki).....	70
Sterowniki led i przetwornik A/C.....	72
Wyświetlacz prędkości jazdy i obrotów silnika.....	74
Projekty PCB.....	75
Zasilanie, przyciski, wyświetlacze siedmiosegmentowe led.....	75
LCD, 1 Wire [®]	76
Wejścia i wyjścia analogowe (transoptory i przekaźniki).....	76
Sterowniki led i przetwornik A/C.....	77
Wyświetlacz prędkości jazdy i obrotów silnika.....	77
Obwody kontrolki świateł, kierunkowskazów oleju, paliwa, hamulców, świateł ostrzegawczych itp.....	78
Obudowa.....	80
Instalacja w samochodzie.....	84
Podsumowanie.....	85
Literatura.....	87

Spis ilustracji

Rysunek 1. Budowa Schemat blokowy ilustrujący budowę układu FPGA.....	8
Rysunek 2. Budowa slice'a ⁶	8
Rysunek 3. Połączenia bloków CLB ⁶	9
Rysunek 4. Budowa slice'a tworzącego blok CLB ⁶	10
Rysunek 5. Schemat blokowy układu MMfpga12.....	12
Rysunek 6. Pamięć Flash M28W640.....	13
Rysunek 7. Magistrala 1 Wire, przebieg czasowy operacji zapisu	15
Rysunek 8. Magistrala 1 Wire, przebieg czasowy operacji odczytu ¹⁰	15
Rysunek 9. Czujnik DS18B20	16
Rysunek 10. Przetwornik analogowo cyfrowy DS2450.....	16
Rysunek 11. Schemat blokowy wyświetlacza LCD	18
Rysunek 12. Sterownik LED STP16C596.....	19
Rysunek 13. Schemat blokowy układu STP16C596	19
Rysunek 14. Komponent TOP_DIGITAL_DASHBOARD	20
Rysunek 15. Schemat blokowy projektu cz. 1	22
Rysunek 16. Schemat blokowy projektu cz. 2.....	23
Rysunek 17. Komponent top_spd_rpm	27
Rysunek 18. Przebieg czasowy transmisji do układu STP16C596	30
Rysunek 19. Schemat blokowy prędkościomierza i obrotomierza.....	31
Rysunek 20. Komponent top_zamek_drzwi.....	32
Rysunek 21. Komponent top_termometr.....	33
Rysunek 22. Schemat blokowy kontrolera magistrali 1 Wire	34
Rysunek 23. Przebieg czasowy operacji zapisu do kontrolera magistrali 1 Wire	36
Rysunek 24. Przebieg czasowy operacji odczytu z kontrolera magistrali 1 Wire ²⁰	36
Rysunek 25. Schemat blokowy sterownika magistrali 1 Wire, OWM_STER.....	37
Rysunek 26. Schemat blokowy obsługi termometrów cyfrowych	38
Rysunek 27. Schemat blokowy układu dekodującego temperaturę	41
Rysunek 28. Pomiar poziomu paliwa w baku ²¹	42
Rysunek 29. Schemat podłączenia czujników samochodowych do przetwornika.....	42
Rysunek 30. Schemat wyznaczenia wartości mierzonych przez przetwornik.....	45
Rysunek 31. Schemat blokowy układu obsługującego przetwornik a/c.....	46
Rysunek 32. Komponent top_lcd.....	47
Rysunek 33. Schemat blokowy układu obsługującego wyświetlacz LCD.....	49
Rysunek 34. Przebiegi czasowe operacji zapisu do wyświetlacza LCD	51
Rysunek 35. Komponent top_liczniki	52
Rysunek 36. Przebieg czasowy operacji odczytu z pamięci M28W640	54
Rysunek 37. Operacja odblokowania bloku pamięci M28W640	56
Rysunek 38. Operacja kasowania bloku pamięci M28W640	56
Rysunek 39. Operacja zapisu pojedynczego słowa w pamięci M28W640	57
Rysunek 40. Przebieg czasowy operacji zapisu do pamięci M28W640.....	57
Rysunek 41. Schemat blokowy komponentów sterujących pamięcią M28W640.....	59
Rysunek 42. Schemat blokowy liczników kilometrów	61
Rysunek 43. Komponent lcd_flash_mux.....	62
Rysunek 44. Komponent ZEGAR	63
Rysunek 45. Komponent top_wipes	64
Rysunek 46. Schemat blokowy sterownika wycieraczek	66
Rysunek 47. Schemat elektryczny płytki zasilania.....	67
Rysunek 48. Schemat elektryczny płytki podłączenia wyświetlacza LCD	69
Rysunek 49. Schemat płytki wejść analogowych i przekaźników	70

Rysunek 50. Schemat podłączenia przetwornika a/c.....	72
Rysunek 51. Schemat podłączenia sterowników led.....	73
Rysunek 52. Schemat podłączenia wyświetlaczy siedmiosegmentowych.....	74
Rysunek 53. Schemat podłączenia linijek led.....	74
Rysunek 54. Moduł zasilania.....	75
Rysunek 55. Moduł wyświetlacza oraz magistrali 1 Wire®.....	76
Rysunek 56. Moduł wejść transoptorów i przekaźników.....	76
Rysunek 57. Moduł sterowników linijek led i przetwornika A/C.....	77
Rysunek 58. Wyświetlacz prędkości jazdy.....	77
Rysunek 59. Wyświetlacz prędkości obrotowej silnika.....	78
Rysunek 60. Moduł stanu świateł zewnętrznych.....	78
Rysunek 61. Moduł stanu kierunkowskazów, świateł ostrzegawczych oraz świateł stop i hamulca ręcznego.....	78
Rysunek 62. Moduł usterki układu smarowania, wyboru paliwa, rezerwy, ogrzewania i wycieraczek.....	79
Rysunek 63. Wnętrze obudowy, panel frontowy.....	80
Rysunek 64. Ekran, wnętrze wyklejone folią aluminiową.....	81
Rysunek 65. Stelaż z kątownika.....	81
Rysunek 66. Obudowa otwarta.....	82
Rysunek 67. Obudowa zamknięta.....	82
Rysunek 68. Panel frontowy.....	83

Wstęp

Szybko postępujący rozwój technologiczny wiążący się z obniżeniem kosztów produkcji układów elektronicznych ma coraz większy udział w produkcji samochodów. Wyspecjalizowane układy cyfrowe, już nie tylko są odpowiedzialne za sterowanie układami bezpieczeństwa, kontrolę pracy silnika, lecz coraz częściej stają się interfejsem użytkownika.

Producenci urządzeń elektronicznych stosują różne technologie. Obecnie bardzo popularne stały się różne typy mikrokontrolerów, które bardzo ułatwiają budowanie złożonych systemów elektronicznych. Jednak nie wszędzie takie rozwiązania są najlepsze. Często stawiane są bardzo konkretne wymagania, które powinien spełniać wyspecjalizowany mikrokontroler, układ scalony. W przypadku projektowania urządzeń krótkoseryjnych, bądź prototypowych, wyprodukowanie samego układu scalonego może wielokrotnie przewyższyć koszt całego urządzenia. Doskonałym rozwiązaniem tego problemu może być układ FPGA¹. Układy tego typu mają również inne zastosowania: testowanie nowych technologii, algorytmów; praca w systemach podlegających sprzętowej rekonfiguracji; wszędzie tam, gdzie opłaca się zrównoleglenie operacji; wszędzie tam, gdzie nie radzi sobie mikrokontroler.

Poniższa praca zawiera przykład urządzenia zaprojektowanego z wykorzystaniem układu FPGA, firmy Xilinx. Do opisu systemu wykorzystano język VHDL².

Kompletny układ zawiera:

- cyfrowe wskaźniki prędkości jazdy oraz prędkości obrotowej silnika, na wyświetlaczach siedmiosegmentowych;
- „pseudo” analogowe wskaźniki prędkości jazdy oraz prędkości obrotowej silnika, wykonane na liniijkach trzydziestu dwóch diod led³;
- liczniki przebiegu pojazdu ogółem, między tankowaniem zbiorników benzyny oraz gazu (osobno dla każdego) oraz dodatkowy licznik z możliwością kasowania;
- zapis stanu liczników do nieulotnej pamięci typu FLASH⁴;

¹ (ang. Field Programmable Gate Array) to rodzaj programowalnego układu logicznego. Może być wielokrotnie przeprogramowany po tym, jak został już wytworzony.

² (ang. Very High Speed Integrated Circuits Hardware Description Language) jest bardzo popularnym językiem opisu sprzętu używanym w komputerowym projektowaniu układów cyfrowych typu FPGA i ASIC.

³ dioda świecąca, ang. light emitting diode

- pomiar wartości analogowych, poziomu paliwa, ciśnienia oleju, temperatury cieczy chłodzącej, oraz konwersję tych sygnałów na formę cyfrową przy użyciu przetwornika analogowo-cyfrowego;
- pomiar temperatury powietrza na zewnątrz oraz wewnątrz pojazdu;
- zegar, programowalny sterownik pracy wycieraczek oraz automatyczne domykanie drzwi po przekroczeniu pewnej prędkości;
- interfejs użytkownika w postaci monochromatycznego wyświetlacza LCD⁵ oraz kilku przycisków.

Niniejsza praca podzielona jest na trzy główne rozdziały. W pierwszym opisane zostały kluczowe elementy elektroniczne wykorzystane w projekcie, między innymi moduł z układem FPGA, pamięć flash, czujniki temperatury, przetwornik analogowo-cyfrowy. Drugi zawiera opis poszczególnych komponentów składowych projektu, w języku VHDL. Natomiast trzeci rozdział zawiera dokumentację techniczną, schematy elektryczne oraz projekty płytek drukowanych.

⁴ rodzaj pamięci EEPROM (ang. Electrically-Erasable Programmable Read-Only Memory), pozwalającej na zapisywanie lub kasowanie wielu komórek pamięci podczas jednej operacji programowania. Jest to pamięć nieulotna – po odłączeniu zasilania nie traci swej zawartości.

⁵ Wyświetlacz ciekłokrystaliczny, (ang. Liquid Crystal Display) - to urządzenie wyświetlające obraz oparte na mechanizmie zmiany polaryzacji światła na skutek zmian orientacji uporządkowania cząsteczek chemicznych, pozostających w fazie ciekłokrystalicznej, pod wpływem przyłożonego pola elektrycznego.

Rozdział I - Charakterystyka elementów

Układ FPGA – Spartan 3 XC3S200, Xilinx

200 tysięcy bramek, 4320 komórek logicznych;

216 kb pamięci Block RAM;

30 kb pamięci Distributed RAM;

12 sprzętowych układów mnożących;

4 układy Digital Clock Manager;

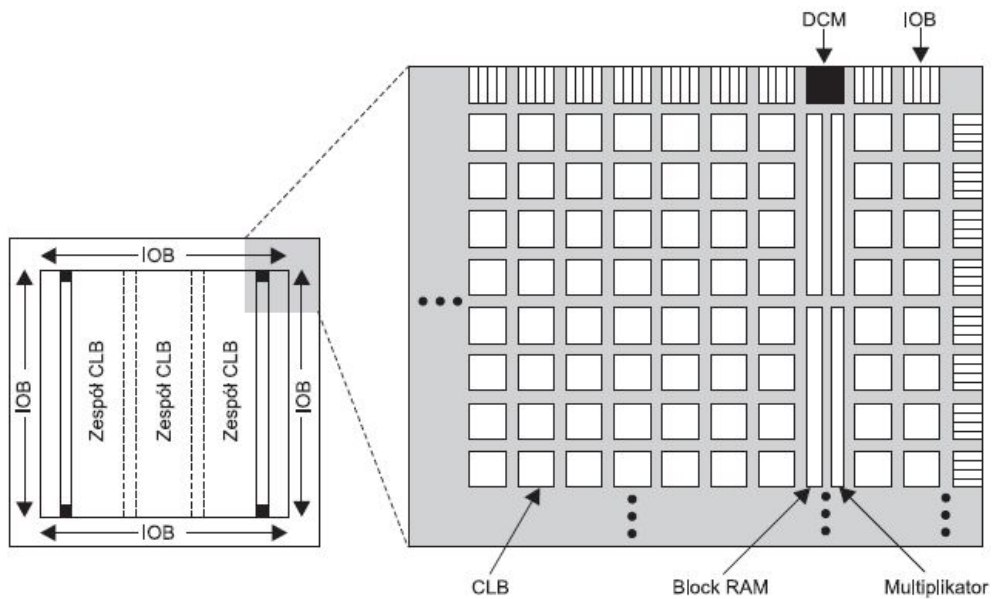
63 piny I/O;

Interfejs JTAG.

Zasoby logiczne odpowiadają ok. 200000 bramek logicznych, zwanych często przeliczeniowymi. Są to zasoby trudne do nadwerężenia w średniozaawansowanych projektach.

Poważniejszym ograniczeniem praktycznym jest stosunkowo niewielka liczba wyprowadzeń dostępnych w obudowie (jest ich 100), ale da się ono odczuć dopiero przy zaawansowanych projektach. XC3S200 jest jednym z najmniejszych układów w rodzinie Spartan 3, a największy spośród obecnie dostępnych układów jest odpowiednikiem 5 milionów bramek przeliczeniowych.

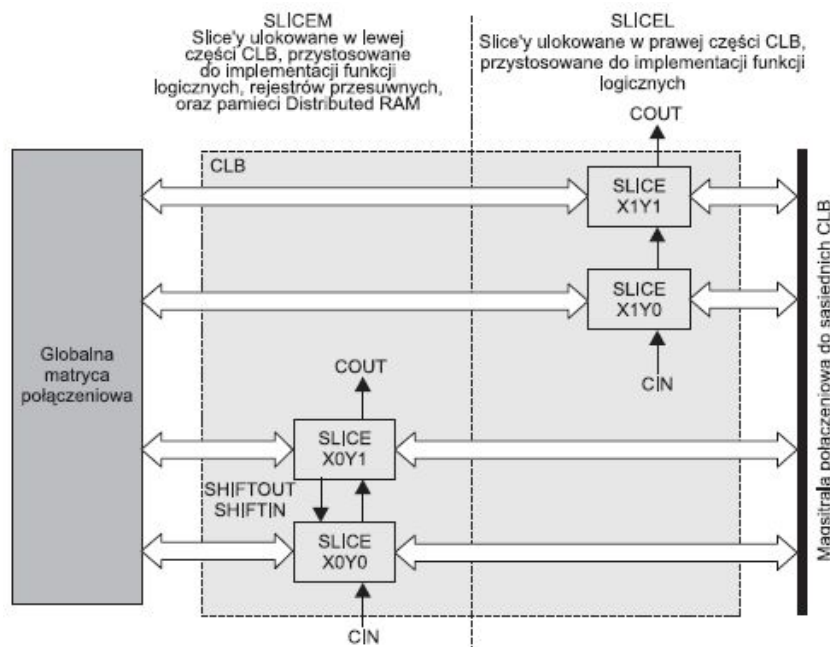
Układy FPGA firmy Xilinx charakteryzują się regularną budową, opartą na zespołach wielu identycznych lub bardzo do siebie podobnych bloków CLB (Configurable Logic Block).



Rysunek 1. Budowa Schemat blokowy ilustrujący budowę układu FPGA⁶

Schemat blokowy układu z rodziny Spartan 3 pokazano na rysunku 1. Na schemacie poza CLB – widać także pamięci BlockRAM, sprzętowe multiplikatory, syntezery przebiegów zegarowych DCM (Digital Clock Manager) oraz komórki I/O o nazwie IOB (Input–Output Block).

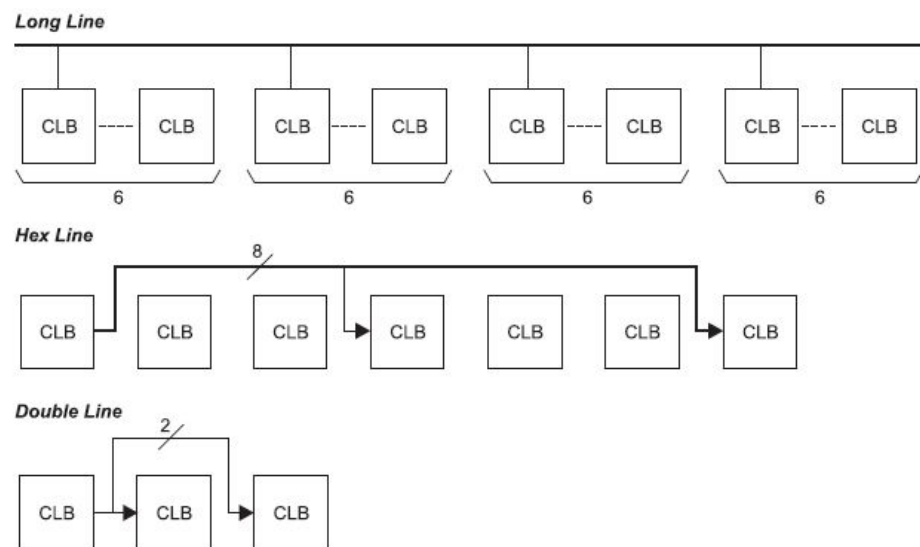
Bloki CLB (w układach Spartan 3 jest ich od 1728 do 74880) są zbudowane z 4 bloków logicznych nazwanych przez firmę Xilinx mianem *slice*.



Rysunek 2. Budowa slice'a⁶

⁶ Elektronika Praktyczna, listopad 2006, „Układy FPGA w przykładach”(4)

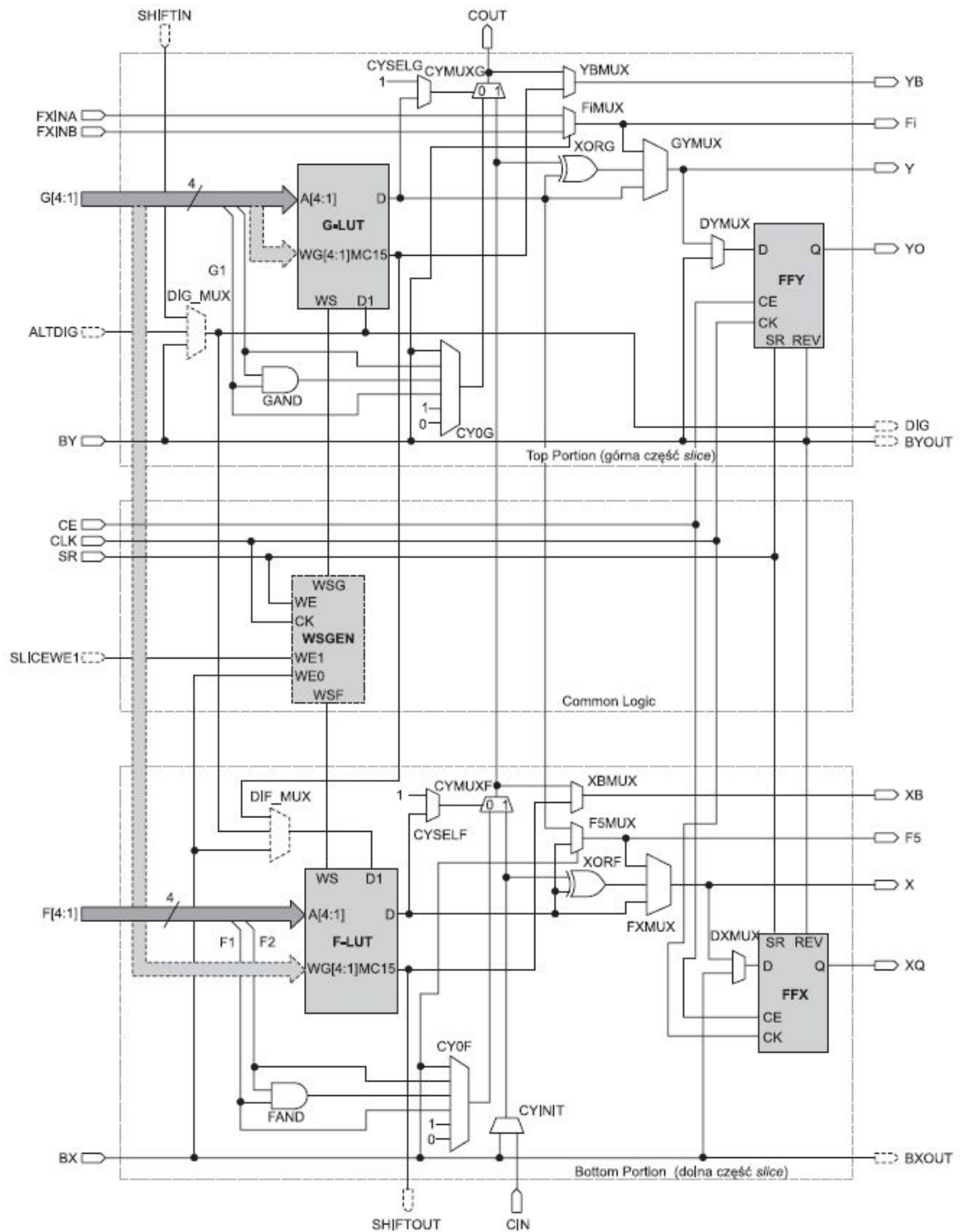
Schemat ilustrujący rozmieszczenie slice'ów w CLB pokazano na rysunku 2. Jak widać, od jednej strony slice'y są dołączone do magistral zapewniających komunikację w obrębie całego układu FPGA (tzw. połączenia globalne o różnym zasięgu), od drugiej strony – do magistral zapewniających komunikację lokalną z sąsiednimi CLB. Slice pogrupowano po dwa (w kolumny) z wydzielonymi szybkimi łańcuchami propagacji sygnału CARRY, dzięki czemu możliwe są implementacje szybko działających bloków logicznych wykorzystujących kaskadowe przeniesienia (liczniki, arytmometry itp.). Jak wspomniano, każdy CLB ulokowany wewnątrz grupy ma możliwość bezpośredniej komunikacji z sąsiadującymi CLB, jest ich zazwyczaj 8.



Rysunek 3. Połączenia bloków CLB⁶

Wymiana danych z dalej położonymi CLB odbywa się za pomocą dodatkowych zasobów połączeniowych (Rysunek 3.):

- linii długich (Long Lines, dołączonych do – kolejno – co szóstego CLB), które są najszybszym traktem komunikacyjnym wewnątrz FPGA, często wykorzystywanym zamiennie z globalnymi liniami zegarowymi,
- linii 8-krotnych (Hex Lines), które rozprawdzają sygnały na mniejsze odległości, oferując większe (niż Long Lines) możliwości połączeniowe i są dołączone do co trzeciego CLB,
- linii podwójnych (Double Lines), które zapewniają bezpośrednią komunikację pomiędzy pozostałymi CLB.



Rysunek 4. Budowa slice'a tworzącego blok CLB⁶

W slice'ach tworzących CLB ulokowano zasoby logiczne, których nawet pobieżna analiza (choćby na schemacie pokazanym na rys. 4.) pokazuje ogrom możliwości i elastyczność tych komórek.

Każdy slice wyposażono w dwie konfigurowalne tablice LUT (F-LUT i G-LUT), na wejścia których są podawane 4 sygnały (zmiennie). Tablice te spełniają rolę konfigurowalnych, kombinacyjnych funkcyjnych logicznych (często są nazywane

generatorami funkcji), które umożliwiają wykonanie dowolnej funkcji logicznej do 4 zmiennych wejściowych. Na wyjściu LUT ulokowano przerzutnik, którego sposób działania (czyli jego typ) można także skonfigurować. Na schemacie pokazanym na rys. 4 zilustrowano budowę slice'y X0Y1 i X0Y0, które wyposażono w sprzętowe rozszerzenia (zaznaczone na rys. 4 linią przerywaną) pozwalające skonfigurować je jako rejestry przesuwne lub zespoły rozproszonej pamięci (tzw. Distributed RAM). Slice'y X1Y0 i X1Y1 mają nieco prostszą budowę (bez fragmentów oznaczonych liniami przerywanymi na rys. 4).

Minimoduł MMfpga12, Propox

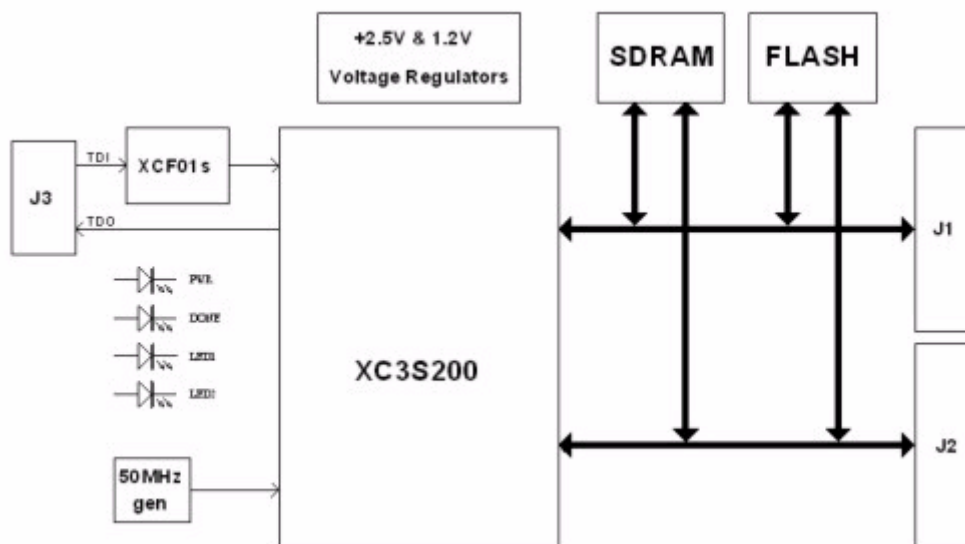
Jest to kompletny, gotowy do użycia system oparty na układzie *FPGA*: XC3S200- układ z rodziny Spartan-3, zawierający 200 tysięcy bramek / 4320 komórek logicznych. Moduł zawiera również pamięć *SDRAM* o pojemności 32MB (16M x 16) oraz pamięć typu *FLASH* o pojemności 8MB (4M x 16).

Wyprowadzenia modułu w postaci dwóch złącz 2 x 40 pinów (rys. 5 złącza J1 i J2) z rastrem 0.1" (2.54 mm), pasujące do wszystkich druków prototypowych zawierają: 43 końcówki we/wy + 8 - bitowa magistrala.

Pamięć konfiguracyjna XCF01S o pojemności 1Mbit umożliwia zapisanie konfiguracji układu *FPGA*. Wbudowane złącze *JTAG* z podłączonym układem *FPGA* i pamięcią konfiguracyjną umożliwia programowanie.

Minimoduł oferowany przez firmę Propox, zawiera wszystkie niezbędne elementy wymagane przez układ *FPGA*: generator kwarcowy 50MHz, stabilizatory 1.2V i 2.5V. Wymaga jedynie podłączenia pojedynczego napięcia zasilania 3.3V.

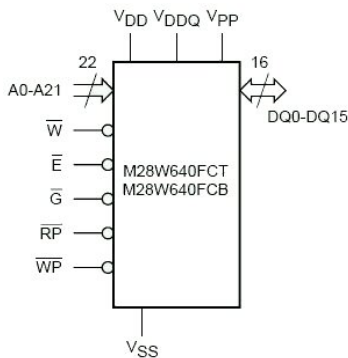
Dodatkowym atutem układu są jego małe wymiary: 40mm x 56mm.



Rysunek 5. Schemat blokowy układu MMfpga12⁷

⁷ Źródło: <http://www.propox.pl/>

Pamięć FLASH M28W640, STM



Rysunek 6. Pamięć Flash M28W640⁸

Układ M28W640 jest 64 megabitową pamięcią nieulotną typu flash⁹. Kasowanie pamięci odbywa się elektrycznie, poszczególnymi blokami. Programowanie odbywać się może pojedynczymi słowami 16 bitowymi. Układ charakteryzuje się niesymetryczną architekturą. Składa się z tablicy 135 bloków, z których 8 (bloki parametrów)składa się z 4000 słów, natomiast pozostałe z 32000. Dostępne są dwie wersje układu z blokami parametrów na początku lub na końcu puli adresów.

Układ umożliwia zabezpieczenie poszczególnych bloków przed usunięciem lub przeprogramowaniem. Blokowanie komórek pamięci odbywa się natychmiastowo. Wszystkie bloki po włączeniu zasilania są zablokowane. Możliwe jest również sprzętowe zabezpieczenie bloków, poprzez podanie odpowiedniego napięcia na wejście WP.

Operacje kasowania oraz zapisu mogą być wstrzymywane np. w celu odczytu lub zapisu danych spod innego adresu, następnie mogą być kontynuowane. Każdy blok może być przeprogramowany ponad 100000 razy.

Operacje zapisu i kasowania sterowane są przez kontroler pamięci, który odpowiada za generowanie odpowiednich sygnałów dla przebiegu powyższych operacji. Zakończenie oraz pomyślne wykonanie zapisu i kasowania jest sygnalizowane poprzez odpowiednie bity rejestru statusu.

⁸ Źródło: M28W640FCT.pdf strona 8(16)

⁹ rodzaj pamięci EEPROM (ang. Electrically-Erasable Programmable Read-Only Memory), pozwalającej na zapisywanie lub kasowanie wielu komórek pamięci podczas jednej operacji programowania. Jest to pamięć nieulotna – po odłączeniu zasilania nie traci swej zawartości

Magistrala 1 Wire[®], Maxim/Dallas Semiconductor

Magistralę 1 Wire[®], opracowano w firmie Dallas Semiconductor, obecnie Maxim. Służy ona do transmisji danych (dwukierunkowo) pomiędzy wieloma urządzeniami typu *slave*, a urządzeniem typu *master*. Jest to konkurencyjne rozwiązanie dla magistrali I²C[®] firmy Philips. Każde urządzenie *slave* ma unikatowy 64 bitowy numer seryjny.

Jak wskazuje nazwa, komunikacja pomiędzy układami może odbywać się przy użyciu tylko jednego przewodu (zasilanie z linii danych) oraz masy. Parametry transmisji umożliwiają stosowanie połączeń na odcinkach do stu metrów. Swoją popularność magistrala zawdzięcza cyfrowym czujnikom temperatury serii DS1820. W chwili obecnej ten sposób transmisji umożliwia wiele układów firmy Maxim; są nimi między innymi czujniki wilgotności, układy identyfikacji, kontrolery stanu baterii, pamięci, zegary rtc, przetworniki analogowo cyfrowe, potencjometry cyfrowe, przełączniki.

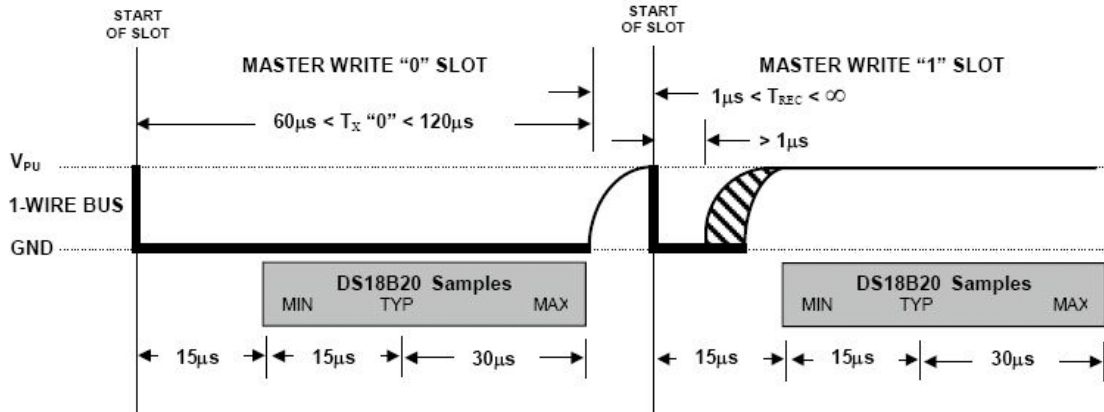
W konfiguracji magistrali przy użyciu tylko jednego przewodu (zasilanie z linii danych). Podczas przerw w transmisji danych, na magistrali utrzymywany jest stan wysoki. Umożliwia to naładowanie wewnętrznych kondensatorów urządzeń podrzędnych. Ładunek w ten sposób zgromadzony, wykorzystany zostaje podczas transmisji do zasilania układów. Niestety takie połączenie magistrali narzuca na projektanta systemu konieczność uwzględnienia opóźnień w transmisji o czas potrzebny do naładowania kondensatorów.

Transmisja jest inicjowana przez urządzenie *master*. Podczas gdy magistrala jest wolna, na linii danych utrzymywany jest stan wysoki. Wówczas urządzenia podrzędne mogą wykonywać swoje wewnętrzne operacje. Transmisja rozpoczyna się od wysłania sygnału reset, co powoduje podanie na linię danych poziomu niskiego na czas 480-960 μ s. Jeżeli do magistrali podłączone są urządzenia podrzędne, odpowiadają stanem aktywnym (poziom niski) na czas 60-240 μ s. Tym sygnałem mikrokontroler jest informowany, że przynajmniej jeden układ jest dołączony do magistrali.

Po takiej inicjalizacji urządzenia *slave* pozostają w nasłuchu. Mikrokontroler następnie decyduje, czy będzie adresował tylko jeden układ podrzędny, czy wszystkie. Odbywa się to poprzez wysyłanie odpowiednich komend.

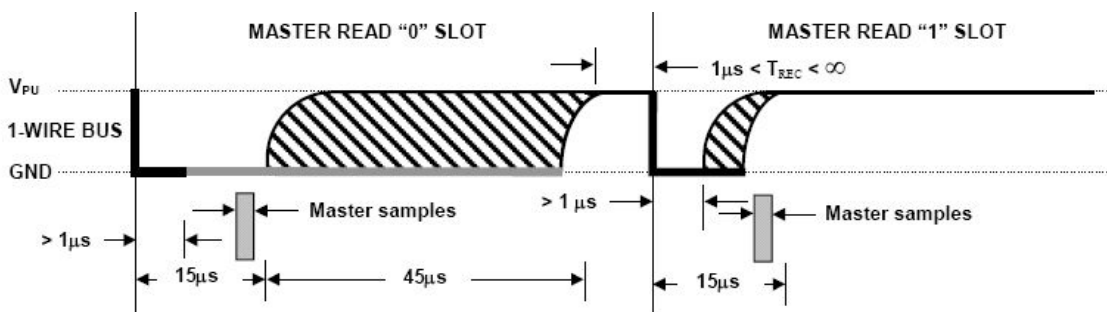
Zapis bitu o wartości "0" polega na wymuszeniu przez układ *master* na linii 1 Wire[®] stanu niskiego przez okres od 60 do 120 μ s, a następnie zwolnieniu linii na

minimum $1\mu\text{s}$. Zapis bitu o wartości "1" do układu slave polega na wymuszeniu przez układ master na linii 1 Wire[®] stanu niskiego przez okres 1-15 μs , a następnie zwolnieniu linii na okres min 60 μs . Przebiegi czasowe podczas zapisu przedstawia rysunek 7:



Rysunek 7. Magistrala 1 Wire, przebieg czasowy operacji zapisu¹⁰

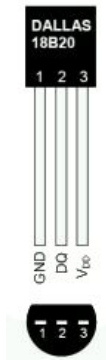
Odbiór bitu z magistrali 1 Wire[®] polega na wygenerowaniu przez układ *master* stanu niskiego przez okres min $1\mu\text{s}$, a następnie zwolnieniu linii 1 Wire[®] i sprawdzeniu jej stanu przed upływem $15\mu\text{s}$. Jeśli urządzenie *slave* transmituje bit o wartości "0", to generowany przez układ *master* impuls ujemny zostanie przedłużony do min $15\mu\text{s}$. Po odczycie stanu linii należy odczekać jeszcze min $46\mu\text{s}$ ($45\mu\text{s}$ na zakończenie transmisji bitu + $1\mu\text{s}$ na zwolnienie magistrali). Jeśli urządzenie *slave* transmituje bit o wartości "1" to impuls generowany przez układ *master* nie zostanie przedłużony i odczyt stanu linii da stan wysoki. Przebiegi czasowe podczas odczytu bitu z układu *slave* są przedstawione na rysunku 8:



Rysunek 8. Magistrala 1 Wire, przebieg czasowy operacji odczytu¹⁰

¹⁰ Źródło: DS18B20.pdf strona 16(11)

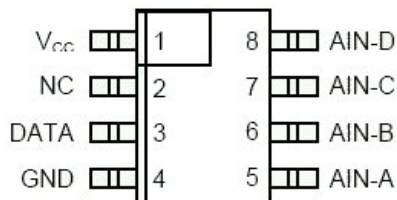
1. Czujnik temperatury DS18B20



Rysunek 9. Czujnik DS18B20¹¹

Jest to scalony, cyfrowy czujnik temperatury, wyposażony w interfejs 1 Wire[®]. Umożliwia pomiar temperatury w zakresie od -55 do 125°C, z maksymalną rozdzielczością 12 bitów. Dokładność w przedziale -10 do 85°C wynosi $\pm 0,5^\circ\text{C}$. Czas konwersji temperatury przy największej rozdzielczości, oraz zasilaniu z linii danych (tzw. zasilanie pasożytnicze) wynosi maksymalnie 750ms. Czujnik ten posiada również możliwość ustawienia alarmów o przekroczeniu temperatury powyżej lub poniżej zaprogramowanych.

2. Czterokanałowy przetwornik analogowo cyfrowy DS2450



Rysunek 10. Przetwornik analogowo cyfrowy DS2450¹²

Jest to przetwornik wykorzystujący popularną metodę przetwarzania sygnału analogowego na cyfrowy, polegającą na doprowadzeniu do (wewnętrznego) przetwornika cyfrowo-analogowego kolejnych kodów przybliżających wynik i porównywaniu sygnału na wyjściu przetwornika cyfrowo-analogowego z analogowym sygnałem wejściowym – przetwarzanie metodą kolejnych przybliżeń.

Umożliwia programowy wybór jednego z dwóch zakresów pomiarowych, 2.56V lub 5.12V. Rozdzielczość pomiarów od 1 do 16 bitów. Zawiera wewnętrzny

¹¹ Źródło: DS18B20.pdf strona 1(11)

¹² Źródło: DS2450.pdf strona 1(12)

czterokanałowy multiplekser sygnału analogowego oraz wejścia o wysokiej impedancji $1M\Omega$.

Jak każde urządzenie zgodne z transmisją 1 Wire[®], DS2450 może być zasilany z linii danych. W przypadku zasilania bezpośrednio z pinu Vcc, wymaga zapisania odpowiedniego parametru w rejestrze konfiguracyjnym, celem właściwego doboru źródła napięcia odniesienia.

Niewykorzystane wejścia przetwornika mogą być użyte jako wyjścia, np. jako źródła przerw dla urządzenia master w przypadku wystąpienia zaprogramowanych alarmów.

Wyświetlacz LCD HY1602F, ze sterownikiem Hitachi HD44780U

Wyświetlacz alfanumeryczny, o organizacji 2x16 znaków. Tło niebieskie, znaki białe (negatyw), z podświetleniem. Pola znakowe w rozmiarze 5x8 pikseli. Przemysłowy, przygotowany do rozszerzonego zakresu temperatur od -20 do 70°C.

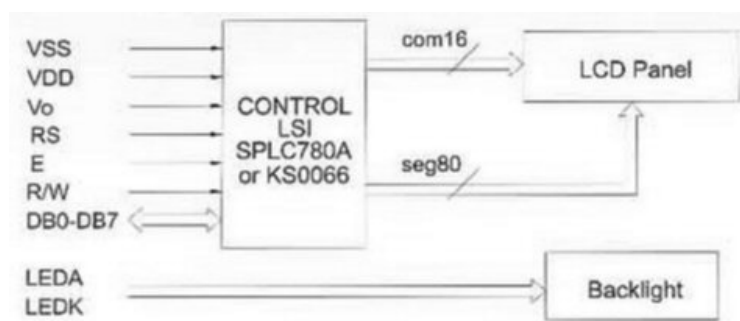
Zintegrowany z wyświetlaczem sterownik oraz kontroler HD44780, zawiera pamięć RAM wyświetlacza oraz generator znaków. Generator znaków zawarty w pamięci ROM sterownika, zawiera 192 znaki. Dodatkowy generator znaków zawarty w pamięci RAM pozwala na zaprojektowanie własnych 8 znaków.

Układ HD44780 umożliwia komunikację z mikrosterownikiem za pomocą czterobitowej lub ośmiobitowej magistrali, oraz sterowanie wyświetlaczem LCD zawierającym maksymalnie 80 znaków.

Inicjalizacja układu odbywa się przy włączeniu napięcia zasilania – napięcie zasilające musi spełniać odpowiednie warunki. Drugim sposobem inicjalizacji jest wysłanie odpowiedniego zestawu instrukcji.

Do sterownika doprowadzona jest szyna danych (D0 – D7) oraz trzy linie sterujące:

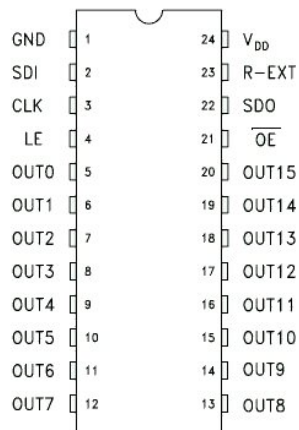
- E – uruchomienie sterownika,
- RS – wybór rejestru wewnętrznego (rejestr instrukcji lub danych)
- RW – zapis lub odczyt pamięci RAM sterownika.



Rysunek 11. Schemat blokowy wyświetlacza LCD¹³

¹³ Źródło: http://www.artronic.pl/o_produkcje.php?id=572

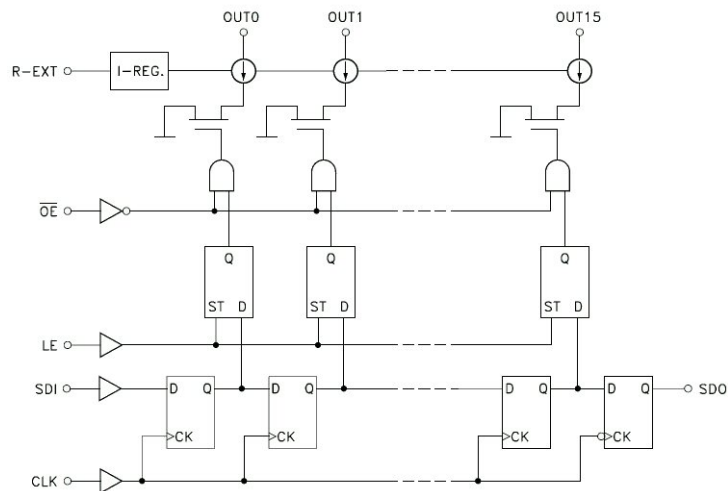
Sterownik LED STP16C596, STM



Rysunek 12. Sterownik LED STP16C596¹⁴

Jest to zintegrowany rejestr przesuwany, z szeregowym wejściem i równoległymi 16 wyjściami, podłączonymi do rejestru składającego się z szesnastu przerzutników D. Zatrzaśnięcie zawartości przerzutników odbywa się po podaniu stanu aktywnego na linię LE. Wyjścia układu umożliwiają płynną regulację natężenia prądu, od 15 do 120 mA dla każdej diody.

Dzięki wyjściu szeregowemu SDO można w prosty sposób łączyć wiele układów tego typu, kaskadowo. Jednak wówczas częstotliwość transmisji szeregowej nie powinna przekraczać 25MHz.



Rysunek 13. Schemat blokowy układu STP16C596¹⁵

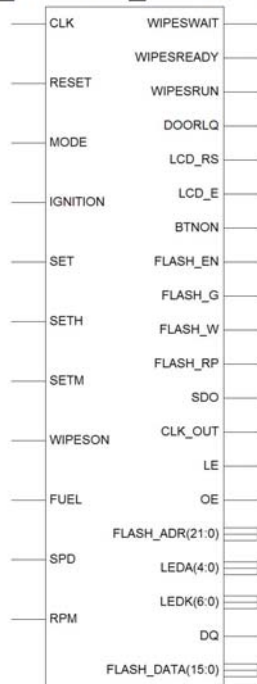
¹⁴ Źródło: stp16c596_rev10.pdf strona 3(18)

¹⁵ Źródło: stp16c596_rev10.pdf strona 5(18)

Rozdział II - Opis systemu

Schemat blokowy

TOP_DIGITAL_DASHBOARD



Rysunek 14. Komponent TOP_DIGITAL_DASHBOARD

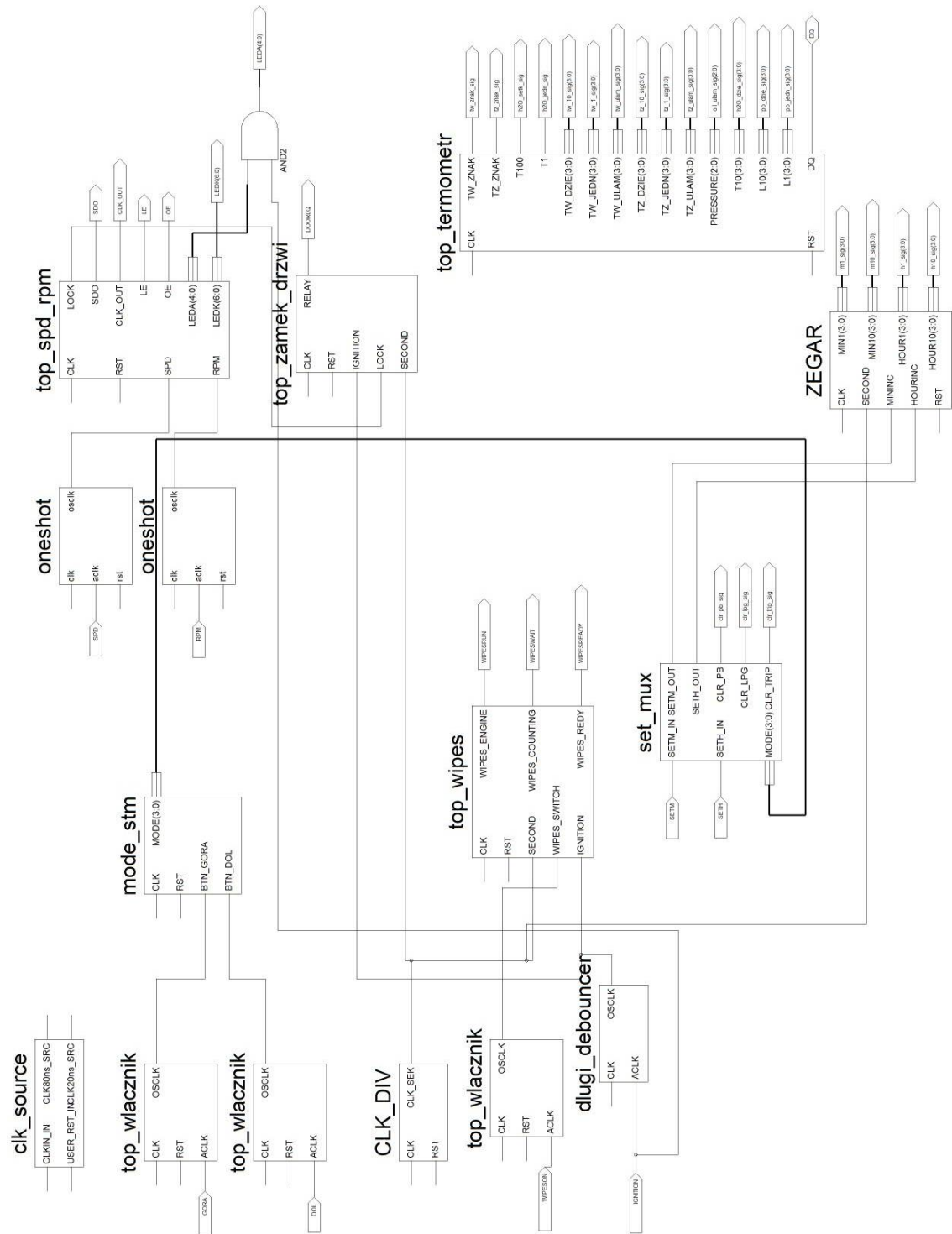
Rysunek 14 przedstawia wszystkie wejścia i wyjścia projektu. Szczegóły połączeń poszczególnych komponentów wewnętrznych znajdują się na kolejnych dwóch rysunkach.

• Wejścia:

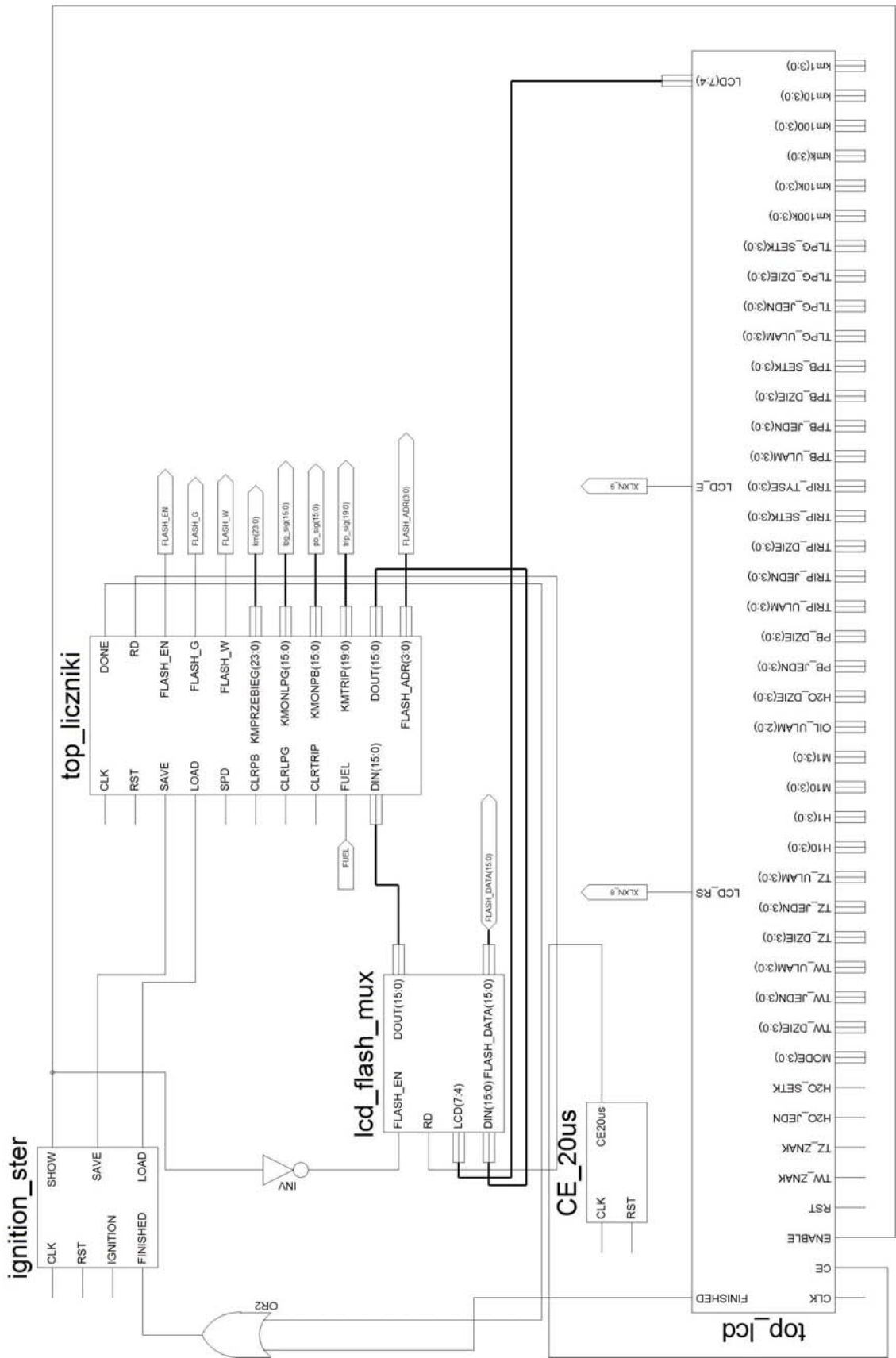
- CLK sygnał generatora kwarcowego o częstotliwości 50Mhz;
- RESET asynchroniczny sygnał resetujący;
- GORA, DOL, SETM, SETH przyciski służący do poruszania się po menu, ustawiania zegara, kasowania liczników, aktywne w zerze;
- IGNITION sygnał ze stacyjki, sterujący pracą automatu zapisującego oraz odczytującego stan liczników do/z pamięci flash. Aktywny w zerze;
- WIPESON sygnał z włącznika wycieraczek, aktywny w zerze. Steruje pracą sterownika wycieraczek;
- FUEL sygnał z elektrozaworu benzyny. Podczas zasilania silnika pojazdu benzyną, sygnał przyjmuje wartość aktywną. Odpowiada za multipleksowanie sygnału SPD pomiędzy licznikami kilometrów na gazie oraz benzynie;
- SPD sygnał z impulsatora linki prędkościomierza. Jeden impuls odpowiada ¼ obrotu linki prędkościomierza;

- RPM sygnał z czujnika położenia wału korbowego silnika. Dwa impulsy odpowiadają jednemu obrotowi silnika.
- **Wyjścia:**
 - WIPESWAIT dioda kontrolna programatora wycieraczek, stan aktywny (dioda świeci) gdy, odbywa się programowanie;
 - WIPESREADY dioda kontrolna programowania wycieraczek, stan aktywny (dioda świeci), gdy wycieraczki zaprogramowane;
 - WIPESRUN wyjście sterujące przekaźnikiem silnika wycieraczek;
 - DOORLQ wyjście sterujące przekaźnikiem zabezpieczenia centralnego zamka;
 - DQ dwukierunkowy port magistrali 1Wire[®];
 - LCD_RS, LCD_E wyjścia sterownika wyświetlacza;
 - SCLK, CKE, CS_N wyjścia sterujące pamięcią RAM. Ponieważ pamięć nie jest wykorzystywana w projekcie, stan na tych wyjściach odpowiada za wyłączenie kontrolera;
 - FLASH_EN, FLASH_G, FLASH_W, FLASH_DATA, FLASH_ADR, FLASH_RP, FLASH_WP są to wyjścia sterujące pamięcią flash. Magistrala FLASH_DATA jest wykorzystywana również do komunikacji z wyświetlaczem LCD;
 - LEDA, LEDK sterują wejściami wyświetlaczy siedmiosegmentowych LED (anody i katody);
 - SDO, CLK_OUT, LE, OE wyjścia te sterują wejściami sterowników linijek LED.

Rysunki 15 i 16 przedstawiają schemat blokowy całego projektu. Ze względu na czytelność diagramów, pominięte zostały linie zegarowe, oraz resetujące.



Rysunek 15. Schemat blokowy projektu cz. 1



Rysunek 16. Schemat blokowy projektu cz. 2

W projekcie wykorzystano jeden z czterech generatorów sygnałów zegarowych DCM, dostępnych w układzie Spartan 3 (komponent *clk_source*). Wynika to z faktu, że elementy logiczne rozmieszczone na powierzchni struktury półprzewodnikowej, pomimo taktowania sygnałem zegarowym pochodzącym z jednego źródła, nie są taktowane jednocześnie. Jest to spowodowane różnym czasem propagacji sygnału w zależności od trasy jaką pokonuje, (różne zasoby połączeniowe wykorzystane do jego transportu), odległości pomiędzy źródłem i celem, a także liczby wejść taktowanych jednocześnie w danym segmencie ścieżki połączeniowej. Aby zminimalizować powyższe problemy, układy Spartan 3 wyposażone są w sprzętowe bloki DCM. Pozwalają one nie tylko skompensować różnice faz (wynikające z opóźnień propagacji) sygnałów zegarowych w (fizycznie) różnych miejscach układu, ale można je również wykorzystać do podziału częstotliwości sygnałów zegarowych.¹⁶ Dzięki temu cała logika projektu taktowana jest sygnałem zegarowym o częstotliwości 12.5 MHz, co odpowiada okresowi 80 ns, oraz sygnały zegarowe wewnątrz struktury układu Spartan prowadzone są głównymi liniami zegarowymi. Częstotliwość, z jaką pracuje układ jest wystarczająca, ponieważ jest mimo wszystko wyższa od częstotliwości zmian sygnałów wejściowych układu. Praca z niższą częstotliwością pozwala również na wykorzystanie „krótszych” liczników wykorzystywanych do odmierzenia odstępów czasu.

Sygnały wejściowe pochodzące z przycisków, włącznika zapłonu oraz czujników prędkości, przechodzą przez komponenty *top_wlacznik*, *dlugi_debouncer* oraz *oneshot*. Są to układy formujące impulsy z zestyków. Jest to spowodowane między innymi zastosowaniem przełączników mechanicznych do sterowania wejść układu cyfrowego. Istnieje niebezpieczeństwo, że przy przełączaniu na wejściu układu może pojawić się zamiast jednego kilka impulsów. Wynika to z drgań kontaktów dociskanych przez sprężyny. Ponieważ cyfrowe układy scalone reagują nawet na bardzo krótkie (~5ns) impulsy, drgania styków grożą poważnymi zakłóceniami. Jednym ze sposobów eliminacji tego zjawiska są obwody RC podłączone na wejściach układów scalonych.¹⁷ W praktyce rozwiązuje się również problem drgań styków przy pomocy przerzutników. Takie rozwiązanie stosuje się na przykład w sytuacji gdy konieczne jest zsynchronizowanie sygnałów zewnętrznych z sygnałem zegarowym.

¹⁶ Elektronika Praktyczna, styczeń 2007, „Układy FPGA w przykładach”(4)

¹⁷ Dieter Nührmann, „Elektronika łatwiejsza niż przypuszczasz”, WKŁ 1986, str. 104(6)

W niniejszej pracy wykorzystano tylko dwa rodzaje „debouncerów”: *oneshot* oraz *dlugi_debouncer*. Komponent *top_wlacznik*, składa się z dwóch komponentów *oneshot*, połączonych kaskadowo. Pierwszy z nich, taktowany jest sygnałem zegarowym o bardzo niskiej częstotliwości (2^{15} razy wolniejszym od głównego sygnału zegarowego). Natomiast podłączony do niego drugi *oneshot*, pozwala uzyskać impuls o długości jednego taktu zegara 12.5 MHz. Taki zabieg pozwolił zastosować w projekcie ogólnie dostępne przełączniki monostabilne o bardzo niskiej jakości.

Komponent *oneshot* służy do generowania pojedynczych impulsów w momencie zwolnienia przycisku. Wykrywa opadające zbocze sygnału wejściowego. Na wyjściu komponentu pojawia się wówczas zsynchronizowany z zegarem pojedynczy impuls.

Do odfiltrowania sygnałów pochodzących z czujnika prędkości oraz położenia wału korbowego silnika, wykorzystany został pojedynczy komponent *oneshot*.

Inaczej zaprojektowany jest komponent *dlugi_debouncer*. Służy on do wykrywania zbocza narastającego oraz opadającego z sygnału na wejściu. Na wyjściu komponentu nie pojawia się pojedynczy impuls, lecz stan aktywny trwający tak długo jak długo podawany jest sygnał na wejście układu FPGA. Jest to rozwiązane poprzez czterobitowy rejestr przesuwany. *Dlugi_debouncer* wykorzystany jest do generowania sygnału sterującego kontrolerem zapisu i odczytu liczników – *ignition_ster*. Komponent *ignition_ster* reaguje na narastające i opadające zbocze sygnału z włącznika zapłonu. Jest to spowodowane faktem, że stan aktywny na wejściu układu FPGA, pojawia się po włączeniu zapłonu (uruchomienie silnika) i trwa dopóty, dopóki silnik nie zostanie zatrzymany. Dlatego konieczne było zastosowanie komponentu zachowującego długość sygnału wejściowego.

CLK_DIV, odpowiada za generowanie impulsów co jedną sekundę. Jest to wykorzystane w komponencie *ZEGAR*, *top_wipes* oraz *top_zamek_drzwi*. W tym ostatnim służy do generowania impulsów powodujących zwarcie styków przekaźnika sterującego centralnym zamkiem. Szczegółowe zastosowanie komponentu *CLK_DIV* opisane jest w rozdziałach poświęconych zegarowi czasu rzeczywistego oraz sterownikowi wycieraczek.

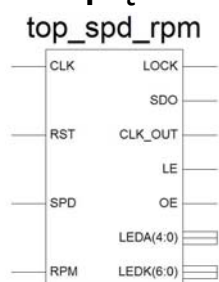
Mode_stm jest licznikiem, którego wartość służy do wyboru wyświetlanych komunikatów. Zmiana wartości odbywa się poprzez podanie stanu aktywnego na wejścia GORA lub DOL. Z wartością sygnału *mode_sig*, ściśle powiązany jest również komponent *set_mux*, który odpowiada za przełączanie sygnałów pochodzących od

przycisków nastawiania minut oraz godzin. Wynika to z faktu, że przycisk nastawiania minut jest wykorzystywany również do kasowania stanu liczników przejechanych kilometrów, odpowiednio dla wyświetlanego licznika. Na przykład, gdy na wyświetlaczu pokazany jest np. stan licznika kilometrów przejechanych na benzynie, naciśnięcie przycisku ustawiania minut powoduje wyzerowanie zawartości tylko tego licznika. Analogicznie naciśnięcie przycisku ustawiania godzin powoduje inkrementację godzin tylko podczas wyświetlania aktualnego czasu.

Sygnaly z wyżej wymienionych przycisków nie są w żaden sposób pozbawiane zakłóceń spowodowanych drganiem styków. Wynika to z faktu, że zmiany zawartości liczników minut i godzin, zezwalane są sygnałem sekund. Dzięki temu możliwe jest również nastawianie wartości zegara poprzez przytrzymanie przycisków, a nie przez pojedyncze „klikanie”. Natomiast podczas wykorzystywania przycisku nastawiania minut do kasowania zawartości liczników kilometrów, nie jest problemem przypadek, gdy kasowanie odbędzie się więcej niż jeden raz.

Opis poszczególnych elementów

Licznik prędkości i obrotomierz



Rysunek 17. Komponent `top_spd_rpm`

Komponent `top_spd_rpm`, służy do pomiaru oraz wyświetlania wartości prędkości jazdy oraz prędkości obrotowej silnika na wyświetlaczach siedmiosegmentowych oraz na 32 diodach led.

Wyjścia `LEDA(4 do 0)` oraz `LEDK(6 do 0)` to odpowiednio pięć anod wyświetlaczy siedmiosegmentowych oraz siedem katod. Trzy cyfry odpowiadają prędkości jazdy, dwie prędkości obrotowej silnika.

Piny `SDO`, `CLK_OUT`, `LE`, `OE` służą do komunikacji szeregowej z układem sterownika diod led, `STP16C596`.

Wyjście `LOCK` służy do sterowania układem odpowiedzialnym za zamknięcie zamka centralnego po przekroczeniu zadanej prędkości jazdy.

Pomiar prędkości jazdy oraz prędkości obrotowej silnika polega na zliczaniu impulsów na wejściach `SPD` oraz `RPM` w pewnych odcinkach czasu. Przedziały czasowe pomiarów dobrane zostały tak, aby liczba zliczonych impulsów odpowiadała prędkości jazdy w kilometrach na godzinę oraz obrotom silnika na minutę.

Do generowania impulsów `SPD` służy impulsator pochodzący z taksometru, założony na linkę prędkościomierza. Impulsator zbudowany jest z wykorzystaniem czujnika Halla `TLE4905`. Generuje cztery impulsy na jeden obrót linki. Sygnał `RPM` pochodzi z czujnika Halla znajdującego się wewnątrz aparatu zapłonowego. Dwa impulsy odpowiadają dokładnie jednemu obrotowi wału korbowego silnika.

Dzięki dobraniu odpowiednich przedziałów czasowych, w których zliczane są impulsy, nie jest konieczne przeliczanie ilości impulsów na wartość obrotów silnika czy prędkości jazdy.

Ponieważ jeden obrót linki prędkościomierza odpowiada przejechaniu dokładnie jednego metra, można przeprowadzić poniższe obliczenia:

$$1 \left[\frac{km}{h} \right] = 1 \cdot \frac{3600}{1000} \left[\frac{m}{s} \right] \approx 0,28 \left[\frac{m}{s} \right].$$

Równanie 1.

Tyle obrotów wykona linka w czasie 1 sekundy. Ponieważ jednemu obrotowi linki towarzyszą cztery impulsy, mnożąc otrzymaną wartość przez cztery otrzymamy $0,28 \cdot 4 = 1,12$ impulsów na sekundę, przy założeniu, że pojazd porusza się z prędkością 1 km/h. Dzieląc jedną sekundę przez otrzymaną ilość impulsów uzyskamy około 0.9 co odpowiada czasowi, w którym otrzymamy całkowitą liczbę impulsów. Podobne obliczenia można przeprowadzić dla obrotomierza. Załóżmy, że wał korbowy silnika wykonuje 1000 obrotów na minutę. Na jeden obrót wału przypadają dwa impulsy.

$$1000 \left[\frac{obr.}{min} \right] = 1000 \cdot \frac{1}{60} \left[\frac{obr}{s} \right] = 16.(6) \left[\frac{obr}{s} \right].$$

Równanie 2.

Z równania 2 wynika, że przy zadanej prędkości obrotowej otrzymamy 33.(3) impulsy na sekundę. Przyjmując, że wyświetlać będziemy obroty z dokładnością stu obrotów na minutę, oraz w formie 1/100, przy założonych obrotach 1000, na wyświetlaczu ukaże się 10.

$$\frac{16.6}{10} = 0,3[s].$$

Równanie 3.

Wartość otrzymana w równaniu 3 odpowiada czasowi, w którym otrzymamy 10 impulsów.

Jak wynika z powyższych obliczeń, impulsy z prędkościomierza zliczane są w przedziałach 0.9 sekundy, natomiast obroty silnika w przedziałach 0.3 sekundy. Dzięki temu nie jest konieczne wykonywanie żadnych obliczeń do określania prędkości jazdy ani prędkości obrotowej silnika. Zliczona liczba impulsów w zadanych odstępach czasu odpowiada wartości prędkości jazdy w kilometrach na godzinę oraz 1/100 obrotów silnika na minutę.

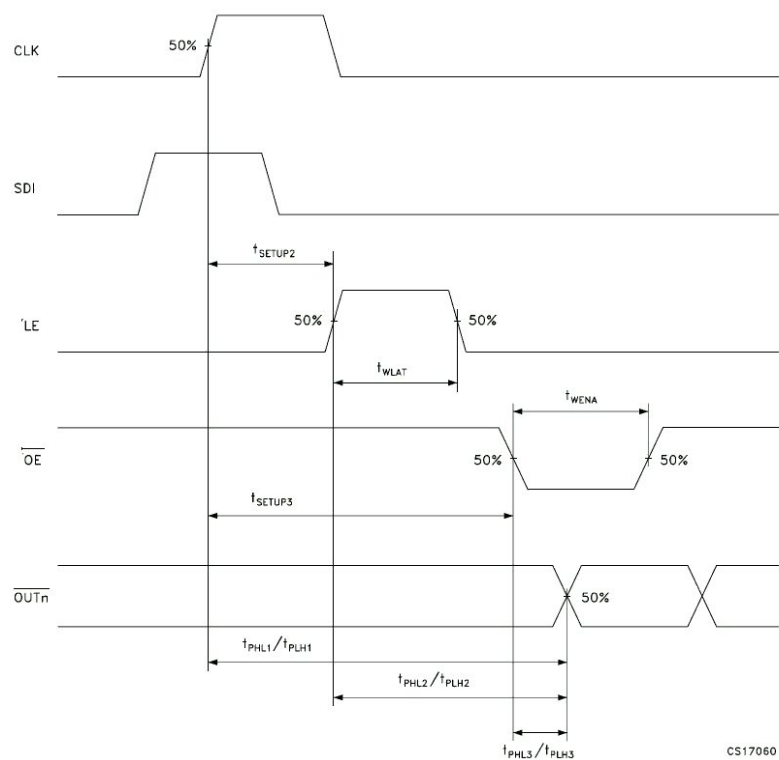
Za generowanie sygnałów zezwalających dla liczników zliczających impulsy odpowiadają komponenty *CLK_COUNTER06* oraz *CLK_COUNTER09*.

Komponenty *RPM_SUM* oraz *SPD_SUM*, składają się z kaskadowo połączonych liczników dziesiętnych, rejestrów przechowujących stan liczników oraz automatów sterujących. Wyjścia JEDN(3 do 0), DZIE(3 do 0) oraz w przypadku

komponentu *SPD_SUM* - SETK odpowiadają wartościom jednostek, dziesiątek oraz setek kilometrów na godzinę oraz wartościom tysięcy i setek prędkości obrotowej silnika. Wyjścia *LED_BAR*(15 do 0), są wyjściami rejestrów przesuwających, odpowiadających szesnastoelementowym linijkom diod LED. Zawartość rejestrów przesuwana jest na starsze pozycje o jeden bit, a na najmłodszy bit wpisywana jest wartość '1'. Dla licznika prędkości przesunięcie odbywa się co każde 10 kilometrów na godzinę, natomiast dla obrotomierza co 400 obrotów.

Komponent *wyświetlacz*, odpowiada za sterowanie anodami wyświetlaczy siedmiosegmentowych oraz za komunikację z układami STP16C596. Anody przełączane są z częstotliwością około 3 kHz. Wyświetlanych cyfr jest pięć dlatego licznik sterujący pracą dekodera anod, oraz multipleksera przełączającego sygnały dla dekodera katod, jest trzybitowy. Jak wiadomo największą liczbą możliwą do zapisania na trzech bitach jest 7. Dla wartości licznika równych: 5, 6 i 7 dekodery anod, wyłączają wszystkie anody. Jest to spowodowane tym, że trzy cyfry wyświetlaczy były by uruchamiane częściej, przez co widoczna była by większa jasność tych segmentów (regulacja PWM).

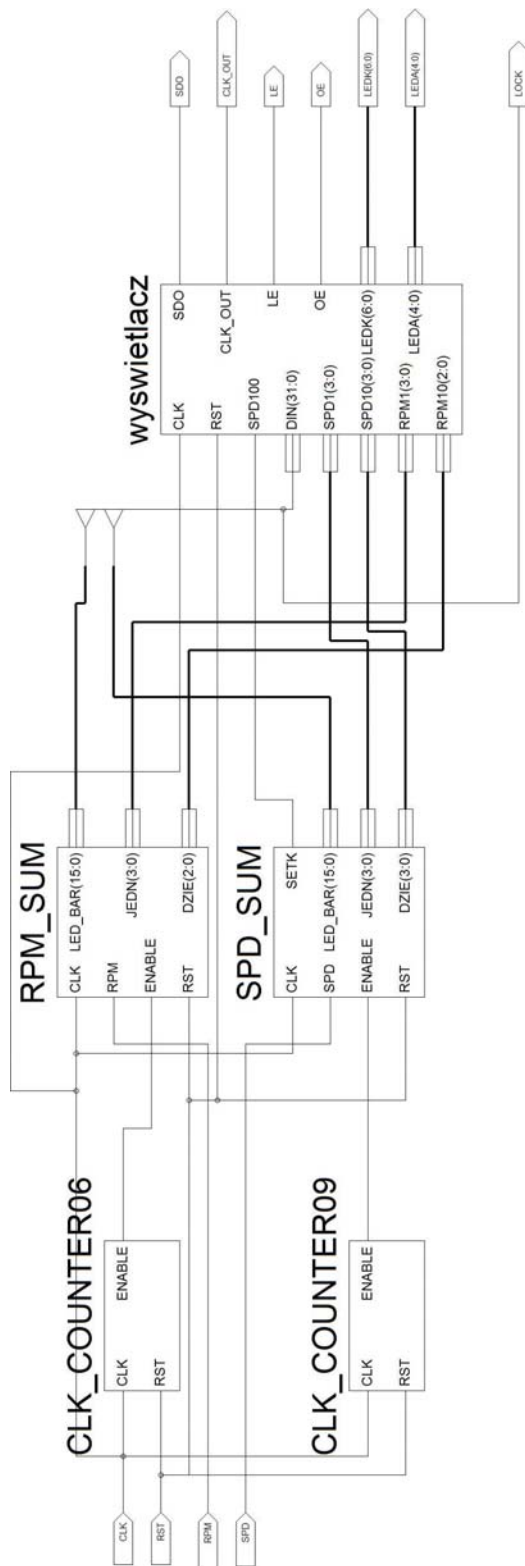
Komunikacja z układami STP16C596 odbywa się z częstotliwością około 4,1 MHz. Przebieg czasowy transmisji przedstawia rysunek 18.



Rysunek 18. Przebieg czasowy transmisji do układu STP16C596¹⁸

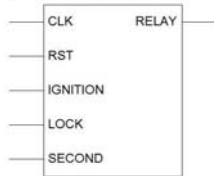
Przez wyjście SDO komponentu *top_spd_rpm*, nadawana jest szeregowo wartość rejestrów *LED_BAR(15 do 0)* komponentów *SPD_SUM* i *RPM_SUM*.

¹⁸ Źródło: stp16c596_rev10.pdf strona 7(18)



Rysunek 19. Schemat blokowy prędkościomierza i obrotomierza

top_zamek_drzwi



Rysunek 20. Komponent top_zamek_drzwi

Komponent *top_zamek_drzwi* odpowiada za sterowanie przekaźnikiem podłączonym do obwodu siłowników zamka centralnego. Zwarcie styków przekaźnika powoduje zabezpieczenie drzwi przed otwarciem z zewnątrz pojazdu.

Jako sygnał nakazu zabezpieczenia drzwi służy wejście LOCK. Jest ono połączone z drugim bitem rejestru LED_BAR(15 do 0) komponentu SPD_SUM. Dzięki temu zamknięcie drzwi następuje po przekroczeniu prędkości 20 km/h.

Automat sterujący opisany wewnątrz komponentu *top_zamek_drzwi*, działa w połączeniu z sygnałem IGNITION, pochodzącym z włącznika zapłonu. Ma to na celu uniknięcie uruchamiania centralnego zamka za każdym razem, gdy prędkość przekroczy 20 km/h, lecz tylko za pierwszym razem po uruchomieniu silnika.

Długość zwarcia styków przekaźnika odpowiada okresowi pomiędzy wystąpieniami sygnału SECOND.

Pomiar temperatury i wartości analogowych



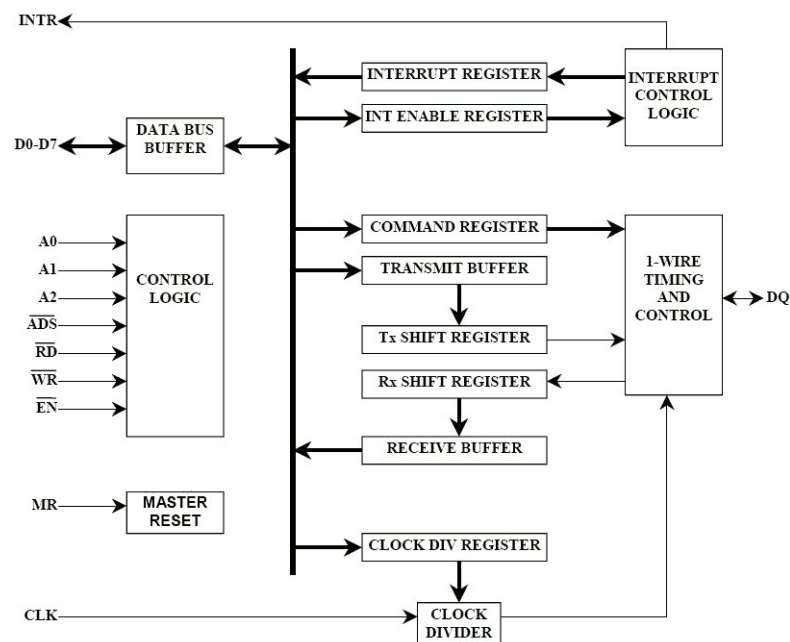
Rysunek 21. Komponent top_termometr

Element odpowiada za komunikację z układami, wykorzystującymi magistralę 1 Wire[®]. Odbywa się to za pośrednictwem dwukierunkowego pinu DQ. Wyjścia TW_ZNAK, TW_DZIE(3 do 0), TW_JEDN(3 do 0), TW_ULAM(3 do 0), TZ_ZNAK, TZ_DZIE(3 do 0), TZ_JEDN(3 do 0), TZ_ULAM(3 do 0), T100, T10(3 do 0), T1, PRESSURE(2 do 0), L10(3 do 0) oraz L1(3 do 0), są wyjściami rejestrów przechowujących wartości temperatury zewnętrznej, wewnętrznej, temperatury cieczy chłodzącej silnik, ciśnienia oleju oraz litrów paliwa w baku. Wyjścia te odpowiadają wejściom komponentu sterującego wyświetlaniem. Warto zwrócić uwagę, że niektóre z sygnałów, np. T1, T100, PRESSURE(2 do 0), nie mają długości 4 bitów. Jest to spowodowane tym, że np. cyfra jednostek temperatury cieczy chłodzącej przyjmuje wyłącznie wartości 0 i 5. Podobnie cyfra setek wartości 0 i 1. Analogicznie dla wartości ciśnienia oleju, 0, 1, 2, 3, 4, 5, 6, 7. Za wyświetlenie odpowiednich wartości odpowiada komponent sterujący wyświetlaczem.

1. Kontroler magistrali 1 Wire®

Ze względu na specyfikację magistrali 1 Wire®, niezbędny jest komponent generujący podstawowe przebiegi czasowe wymagane przez urządzenia podrzędne podłączone do magistrali. Szczegóły opisane zostały w rozdziale poświęconym transmisji 1 Wire®.

Firma Maxim udostępnia na swojej stronie internetowej kod źródłowy w języku VHDL lub Verilog, oraz dokumentację specjalnego komponentu *OWM One Wire Master*, pozwalającego programiście skupić się wyłącznie na wykorzystaniu magistrali w praktyce, a nie na sposobie jej implementacji.



Rysunek 22. Schemat blokowy kontrolera magistrali 1 Wire¹⁹

Jak widać na schemacie (Rysunek 22.) kontroler ten posiada dwukierunkową 8 bitową magistralę danych. Dzięki temu wysyłając poszczególne komendy do urządzeń SLAVE (czujnik temperatury, przetwornik A/C) programista nie musi wnikać w szczegóły transmisji.

Porty A0, A1, A2 służą do wyboru wewnętrznych rejestrów kontrolera. Pozostałe linie ADS, RD, WR, EN, MR służą do sterowania logiką wewnętrzną automatu.

Linia INTR służy jako sygnał przerwania dla nadrzędnego automatu wysyłającego jedno bajtowe komendy do urządzeń SLAVE. Rejestr wewnętrzny sterownika OWM – INT ENABLE REGISTER służy do wyboru źródeł przerw. W projekcie

¹⁹ Źródło: DS1WM.pdf strona 2(10)

wykorzystano jedynie przerwanie od zakończenia komendy reset magistrali 1 Wire[®], oraz od opróżnienia rejestru TRANSMIT BUFFER.

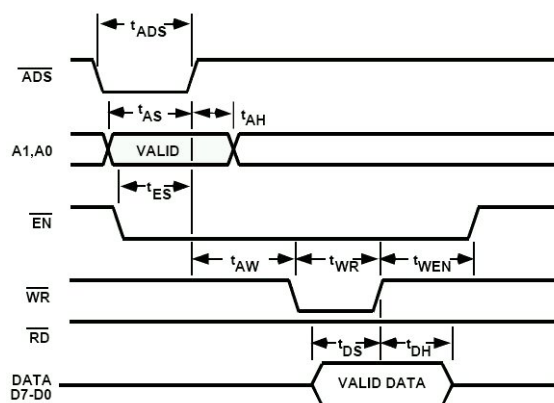
Dzięki temu każda komunikacja z urządzeniami SLAVE odbywa się następująco:

1. Wysłanie komendy reset magistrali, poprzez zapisanie komendy 0x01 do rejestru COMMAND REGISTER;
2. Oczekiwanie na przerwanie;
3. Sprawdzenie czy w rejestrze INTERRUPT REGISTER flaga PRESECE DETECT = '1', gdy urządzenie SLAVE odpowiedziało poprawnie;
4. Wpisanie właściwej komendy do rejestru TRANSMIT BUFFER. Np. 0xCC czyli SkipRom (nadajemy do wszystkich urządzeń);
5. Oczekiwanie na przerwanie od opróżnienia rejestru TRANSMIT BUFFER EMPTY. W tym momencie wewnętrzne automaty OWM'a zajmują się wysyłaniem pojedynczych bitów, poprzez rejestr TRANSMIT SHIFT REGISTER.

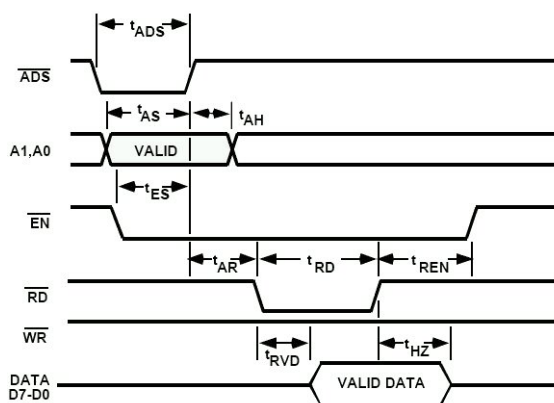
Powtarzając kroki 4 i 5, można nadawać kolejne komendy. Odbiór danych od urządzeń SLAVE, odbywa się poprzez wpisanie bajta 0xFF do rejestru TRANSMIT BUFFER. Po wystąpieniu przerwania TRANSMIT BUFFER EMPTY, można spod adresu rejestru TRANSMIT BUFFER, odczytać dane przysłane przez urządzenie SLAVE.

Istotny jest fakt, że przed rozpoczęciem jakiegokolwiek transmisji poprzez OWM, należy skonfigurować rejestr CLOCK DIV REGISTER, służący do wyboru odpowiedniego dzielnika zegara taktującego kontroler. W celu uproszczenia projektu, ustawienia rejestru CLOCK DIV REGISTER, ustawione zostały statycznie w deklaracji projektu, podobnie ustawienia rejestru przerwania INT ENABLE REGISTER.

Ze względu na wymogi czasowe operacji zapisu i odczytu (rysunki 23 i 24) danych z rejestrów wewnętrznych kontrolera, konieczne było użycie dwóch automatów sterujących tymi procesami.



Rysunek 23. Przebieg czasowy operacji zapisu do kontrolera magistrali 1 Wire²⁰

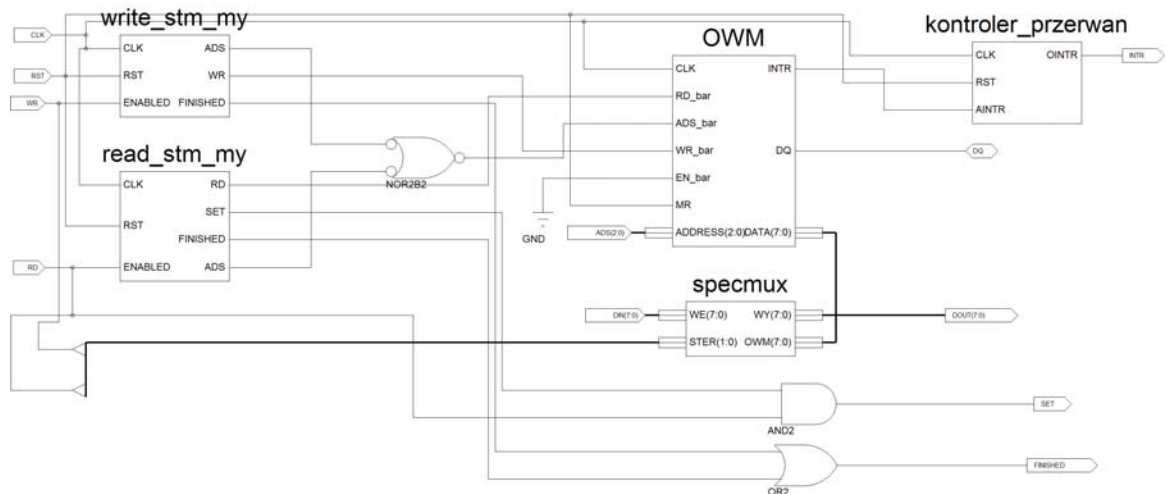


Rysunek 24. Przebieg czasowy operacji odczytu z kontrolera magistrali 1 Wire²⁰

²⁰ Źródło: DS1WM.pdf strona 14(10)

2. OWM_STER

Składa się z kontrolera magistrali 1 Wire®, dwóch maszyn stanów sterujących zapisem (*write_stm_my*), odczytem (*read_stm_my*) rejestrów kontrolera oraz multipleksera (*spec_mux*).



Rysunek 25. Schemat blokowy sterownika magistrali 1 Wire, OWM_STER

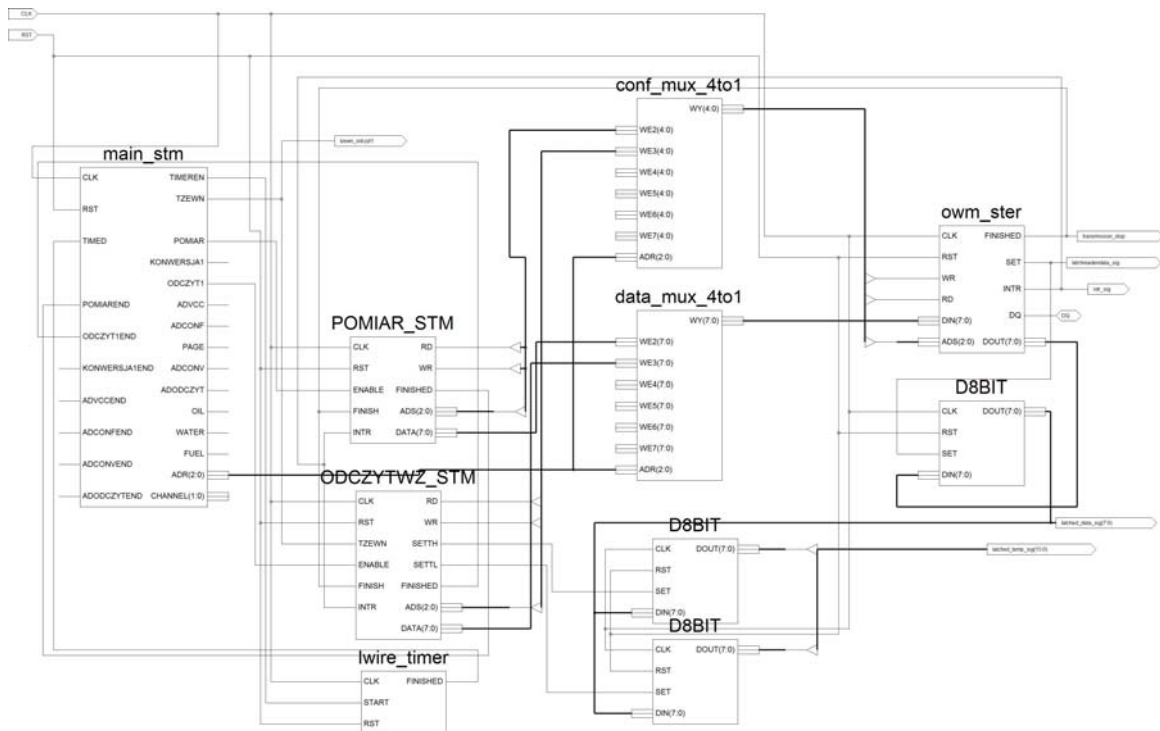
Komponent *kontroler_przerwan*, odpowiada za skrócenie stanu aktywnego sygnału INTR do jednego taktu zegara CLK. Powodem tego jest fakt, iż cała logika wewnętrzna komponentu *OWM* jest taktowana zegarem wewnętrznym o okresie $1\mu\text{s}$ i stan linii INTR zmienia się po upływie $1\mu\text{s}$. Dlatego, aby uniknąć sprawdzania czy linia INTR jest nieaktywna, komponent *kontroler_przerwan* kontroluje stan linii INTR komponentu *OWM*, co w znaczącym stopniu wpływa na uproszczenie pozostałych automatów odpowiedzialnych za konfigurację i komunikację.

Wyjście SET, służy do ładowania rejestru tymczasowego *D8BIT*, połączonego bezpośrednio z *owm_ster*. Stan aktywny ustawiany jest podczas operacji odczytu.

Linia FINISHED przyjmuje stan aktywny, gdy zostanie zakończony zapis lub odczyt.

3. Obsługa termometrów cyfrowych oraz przetwornika A/C

Pomiar temperatury oraz wartości analogowych, sterowany jest poprzez automat Moore'a: *main_stm*. Maszyna stanów steruje wejściami zezwalającymi pozostałych maszyn: *KONFIG_STM*, *POMIAR_STM*, *ODCZYTWZ_STM*, *Iwire_timer*, *dekoder_temperatury*, *AD_vcc_stm*, *AD_conf_stm*, *AD_pomiar_stm*, *AD_odczyt_stm*.



Rysunek 26. Schemat blokowy obsługi termometrów cyfrowych

Wyjścia danych, sygnałów zezwalających na odczyt, zapis oraz adresowych, poszczególnych maszyn, multipleksowane są przez multipleksery *data_mux_4to1* oraz *conf_mux_4to1*.

Podczas pierwszego przejścia przez poszczególne stany, *main_stm* konfiguruje przetwornik analogowo cyfrowy (DS2450). Konfiguracja przetwornika polega na zapisaniu w jego pamięci odpowiednich ustawień dotyczących sposobu zasilania, poprzez automat *AD_vcc_stm*. Następnie maszyna stanów *AD_conf_stm*, zostaje uruchomiona trzykrotnie, w celu konfiguracji poszczególnych wejść przetwornika. Ustawiony zostaje typ wejść (wejścia analogowe), rozdzielczości (8 bitów) oraz zakres napięć mierzonych (5,12 V)

Następnie uruchomiona zostaje maszyna *POMIAR_STM*, wysyłająca do urządzeń SLAVE podłączonych do magistrali komendy 0xCC oraz 0x44. Oba czujniki DS18B20 wykonują konwersję temperatury. Przetwornik analogowy

DS2450 nie wykonuje tej operacji, ponieważ komenda 0x44 nie zostanie rozpoznana. Jest to duże uproszczenie komunikacji, ponieważ nie trzeba adresować pojedynczych urządzeń SLAVE, dzięki komendzie 0xCC.

Podczas gdy czujniki temperatury wykonują pomiar temperatury, uruchomiony zostaje komponent *AD_pomiar_stm*. Wysyła on do wszystkich urządzeń SLAVE komendy 0xCC oraz 0x3C. Komenda 0x3C zostaje rozpoznana wyłącznie przez przetwornik analogowo-cyfrowy. Następnie zostają wysłane dodatkowe informacje niezbędne do konwersji, informujące przetwornik, na których wejściach analogowych ma zmierzyć napięcia. Przetwornik odpowiada dwoma bajtami potwierdzającymi konfigurację.

Następnie *main_stm* uruchamia *Iwire_timer*, który po zliczeniu zadanej ilości taktów zegara, ustawia swoje wyjście FINISHED, w stan aktywny. Ma to na celu wprowadzenie przerwy w transmisji, zakończenia wszystkich operacji konwersji przez urządzenia SLAVE oraz uniknięcia migotania wartości temperatury lub parametrów silnika podczas wyświetlania.

Po odczekaniu 4 sekund transmisja zostaje ponownie rozpoczęta, w celu odczytania zmierzonej temperatury. Odpowiada za to komponent *ODCZYTWZ_STM*. Na tym etapie do komunikacji z urządzeniami SLAVE są używane ich numery seryjne, ponieważ w jednym momencie może nadawać tylko jedno urządzenie. Numery seryjne są na stałe wpisane do kodu źródłowego programu i zadeklarowane jako dwa sygnały 128 bitowe. Niesie to za sobą pewne konsekwencje, ponieważ w przypadku uszkodzenia któregoś z czujników konieczne będzie ponowne zaprogramowanie całego układu FPGA. Jednak nie jest to zbyt duży problem. W tym przypadku takie rozwiązanie jest lepsze, gdyż nie wymaga implementacji skomplikowanej procedury wyszukiwania urządzeń SLAVE.

Wyboru, z którego czujnika temperatury odczytać temperaturę dokonuje się poprzez ustawienie '1' lub '0' na wejściu TZEWN, automatu *ODCZYTWZ_STM*.

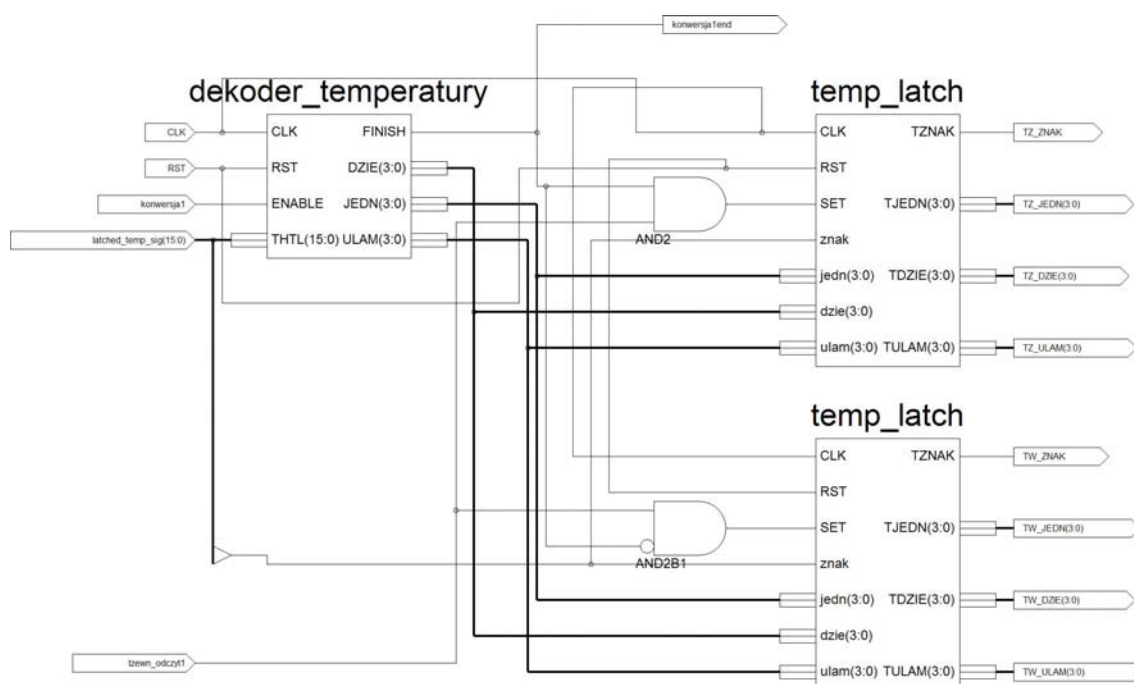
Odczytania zawartość pamięci układu DS18B20 ładowana jest do dwóch rejestrów D8BIT. Wynika to ze sposobu konwersji temperatury przez termometr cyfrowy do postaci dwóch bajtów.

Tabela 1. Organizacja zapisu temperatury w pamięci układów DS18B20

Bardziej znaczący bajt							
7	6	5	4	3	2	1	0
Z	Z	Z	Z	Z	2^6	2^5	2^4
Mniej znaczący bajt							
7	6	5	4	3	2	1	0
2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}

Jak wynika z tabeli 1, pięć bardziej znaczących bitów bardziej znaczącego bajta zawiera informacje, czy mamy do czynienia z temperaturą dodatnią czy ujemną. Kolejne trzy bity oraz cztery bity mniej znaczącego bajta odpowiadają wartości dziesiętnej temperatury. Pozostałe cztery bity odnoszą się do części ułamkowej wartości temperatury. Istotny jest fakt, że w przypadku temperatur ujemnych bity oznaczone 'Z' przyjmują wartość '1' oraz wartość temperatury zostaje zanegowana.

Taki sposób przedstawiania temperatury w pewnym stopniu komplikował wyświetlanie zmierzonej temperatury, ponieważ otrzymujemy całą liczbę a nie poszczególne cyfry. Możliwych było kilka rozwiązań, między innymi stworzenie ogromnego dekodera, który zawierałby instrukcje warunkowe dla każdej możliwej temperatury z przedziału -25 do +125 °C. Najbardziej optymalnym rozwiązaniem było stworzenie dekodera w oparciu o dwa liczniki dziesiętne, które zliczają odpowiednio liczbę dziesiątek oraz jednostek wartości temperatury do wartości zadanej, przechowywanej w rejestrach *D8BIT*. Dodatkowo, przed rozpoczęciem dekodowania temperatury, następuje jej zanegowanie w przypadku temperatur ujemnych. Jest to rozwiązane poprzez funkcję logiczną XOR bitu znaku z kolejnymi bitami wartości dziesiętnej. Część ułamkowa temperatury zostaje zdekodowana, poprzez instrukcję warunkową „select”. Ponieważ konwersja temperatury wykonywana jest z rozdzielczością 12 bitową, wartość ułamkowa zajmuje 4 bity. Dekoder dokonuje zaokrąglenia wartości ułamkowej do jednej cyfry odpowiadającej 10^{-1} . Wartość zdekodowanej temperatury zapamiętywana jest w rejestrach *temp_latch*. Wybór rejestru dokonywany jest poprzez ziloczynowanie sygnału *tzewn_odczyt1* oraz sygnału zakończenia dekodowania temperatury.



Rysunek 27. Schemat blokowy układu dekodującego temperaturę

Po zakończeniu dekodowania temperatury z drugiego czujnika automat *main_stm* przechodzi do odczytywania wartości zmierzonych na wejściach przetwornika analogowo cyfrowego. Ponieważ układ DS2450 jest tylko jeden, w jego przypadku można pominąć nadawanie numeru seryjnego i skorzystać z komendy 0xCC/.

Odczytem steruje automat *AD_odczyt_stm*; uruchamiany jest trzykrotnie dla każdego kanału przetwornika osobno. Wyboru kanału dokonuje się poprzez odpowiednie ustawienie wejścia CHANNEL(1 do 0). Odczytane dane przechodzą za każdym razem przez dekodery wartości ciśnienia oleju, temperatury cieczy chłodzącej i poziomu paliwa, jednak ładowanie rejestrów przechowujących już zdekodowane wartości sterowane jest poprzez *main_stm*.

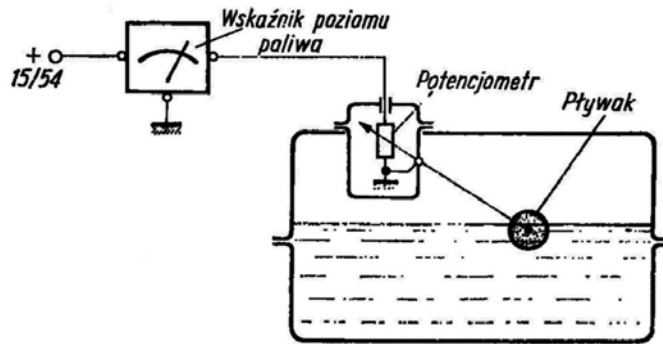
Dekodery wartości analogowych, działają w oparciu o instrukcje warunkowe If... Then.

4. Pomiar ciśnienia oleju, temperatury cieczy chłodzącej oraz poziomu paliwa.

Do pomiaru ciśnienia oleju oraz poziomu paliwa służą oryginalne czujniki samochodu marki FSO Polonez. Wyjątkiem jest czujnik temperatury cieczy chłodzącej pochodzący z samochodu FORD Sierra.

Standardowy sposób pomiaru tych wartości w samochodzie wyjaśniony został na przykładzie pomiaru zapasu paliwa w baku. „Typowym rozwiązaniem,

stosowanym w tym przypadku, jest czujnik potencjometryczny, którego zasadę działania wyjaśnia poniższy rysunek.

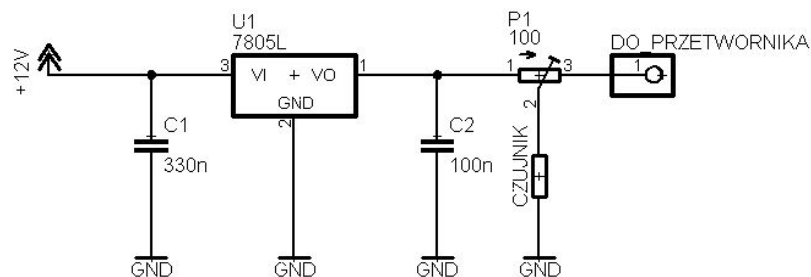


Rysunek 28. Pomiar poziomu paliwa w baku²¹

Na powierzchni cieczy unosi się pływak połączony mechanicznie z potencjometrem. W zależności od poziomu benzyny pływak znajdzie się niżej lub wyżej, co spowoduje, że oporność potencjometru przybierać będzie różne wartości. W szereg z potencjometrem jest włączony bezpośrednio wskaźnik poziomu paliwa reagujący na położenie pływaka.”²¹

Podobnie podłączone są w instalacji elektrycznej samochodu pozostałe czujniki: temperatury cieczy chłodzącej oraz ciśnienia oleju.

Ponieważ przetwornik analogowo cyfrowy umożliwia pomiar napięcia, należało za pośrednictwem czujników samochodowych zbudować dzielniki napięcia. Ze względu na zakres napięć mierzonych przez przetwornik (od 0 do 5,12 V), oraz pojawiające się spadki napięcia w instalacji samochodowej (spowodowane włączeniem np. świateł stop, wentylatora itp.), do budowy dzielników napięcia zastosowany został stabilizator LM7805.



Rysunek 29. Schemat podłączenia czujników samochodowych do przetwornika

²¹ Piotr Słodowy, „Elektronika w moim samochodzie”, WKŁ 1982, str. 81(5)

Rysunek 29 przedstawia sposób podłączenia czujników samochodowych do przetwornika analogowo cyfrowego. Do budowy dzielnika użyty został potencjometr wieloobrotowy P1 o oporze 100 Ohm. Pozwala on na precyzyjne dostrojenie charakterystyki. Kondensatory C1 i C2 są wymagane do pracy stabilizatora.

Charakterystyka czujnika poziomu paliwa zbadana została poprzez mierzenie wartości oporu czujnika podczas dolewania do pustego baku wody, co 1 liter. Wyniki tych pomiarów zawiera tabela 2.

Tabela 2. Przeliczenia litrów paliwa na napięcie

Litr	R[Ω]	U[V]
0	6,3	1,48
1	6,3	1,48
2	6,4	1,50
3	6,4	1,50
4	8,6	1,82
5	12,3	2,25
6	13,9	2,40
7	15,8	2,56
8	17,4	2,69
9	21,1	2,92
10	23,0	3,03
11	24,6	3,11
12	26,5	3,19
13	28,4	3,27
14	30,3	3,34
15	32,1	3,41
16	32,1	3,41
17	33,9	3,47
18	35,3	3,51
19	37,1	3,56
20	38,9	3,61
21	40,7	3,65
22	40,8	3,66
23	42,7	3,70
24	42,9	3,70
25	44,2	3,73
26	44,6	3,74
27	46,3	3,78
28	48,1	3,81
29	54,7	3,92
30	54,8	3,93
31	56,5	3,95
32	56,5	3,95
33	58,2	3,98
34	60,1	4,00
35	67,3	4,09
36	67,4	4,09
37	67,6	4,09
38	69,9	4,12
39	71,5	4,13
40	74,9	4,17
41	76,8	4,18
42	79,2	4,20
43	82,5	4,23
44	87,1	4,27
45	91,5	4,30

Analogicznie obliczone zostały napięcia na pozostałych czujnikach. Charakterystyka czujnika temperatury (tabela 3) cieczy chłodzącej zbadana została podczas wolnego podgrzewania czujnika w oleju, pomiarach temperatury i oporu czujnika. Do pomiaru temperatury użyty został czujnik DS18B20 podłączony do mikrokontrolera AVR Attiny2313.

Charakterystyka czujnika ciśnienia oleju (tabela 4) zbadana została na podstawie pomiarów potencjometru podłączonego do instalacji elektrycznej samochodu. Na

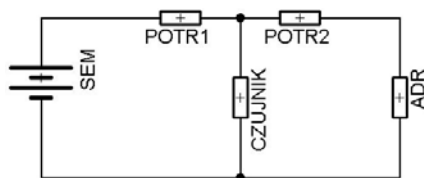
oryginalny wskaźnik na desce rozdzielczej naniesiona została dokładna skala od 0,0 do 0,8 co 0,1. Potencjometr nastawiany był na dane wartości z przedziału, następnie wykonywany był pomiar rezystancji czujnika.

Tabela 3. Przeliczenia temperatury cieczy chłodzącej na napięcie

Temperatura [st. C]	Opór[Ω]	Napięcie[V]
40	470,0	4,52
45	400,0	4,44
50	320,9	4,33
55	268,0	4,21
60	231,2	4,11
65	186,4	3,94
70	158,3	3,80
75	134,4	3,64
80	113,7	3,47
85	97,8	3,31
90	84,0	3,13
95	72,2	2,95
100	62,3	2,77
105	60,1	2,73
110	52,4	2,56
115	46,2	2,40
120	41,1	2,25
125	36,8	2,12
130	33,0	1,99

Tabela 4. Przeliczenia ciśnienia oleju na napięcie

Ciśnienie [Pa]	Opór[Ω]	Napięcie[V]
0.0	245,0	4,71
0.1	186,0	4,63
0.2	149,0	4,54
0.3	123,0	4,46
0.4	91,9	4,30
0.5	71,6	4,13
0.6	52,9	3,90
0.7	26,6	3,20



Rysunek 30. Schemat wyznaczenia wartości mierzonych przez przetwornik

Korzystając ze schematu przedstawionego na rysunku 30, można wyznaczyć napięcie mierzone przez przetwornik analogowo cyfrowy. Rezystor ADR stanowi opór wewnętrzny przetwornika, około 1 MΩ. Rezystory POTR1 oraz POTR2, to potencjometr ze schematu 1. Ponieważ wartość rezystora ADR jest dużo większa od pozostałych, możemy założyć, że natężenie prądu płynącego przez ten rezystor jest bliskie zeru. Takie założenie bardzo upraszcza dalsze obliczenia, dzięki temu szukamy wyłącznie napięcia na rezystorze CZUJNIK. Dlatego możemy skorzystać z prawa Ohma i wyprowadzić następujący wzór (Równanie 4):

$$U_{CZUJNIK} = \frac{R_{CZUJNIK}}{R_{CZUJNIK} + R_{POTR1}} \cdot U_{ZASILANIA}$$

Równanie 4.

Parametry rezystora POTR1 (potencjometr) zostały dobrane doświadczalnie w celu uzyskania najszerszego zakresu zmian napięcia. Obliczone wartości napięcia odpowiadające zapasowi paliwa w baku, temperaturze cieczy chłodzącej oraz ciśnienia oleju zawierają tabele 2, 3 i 4.

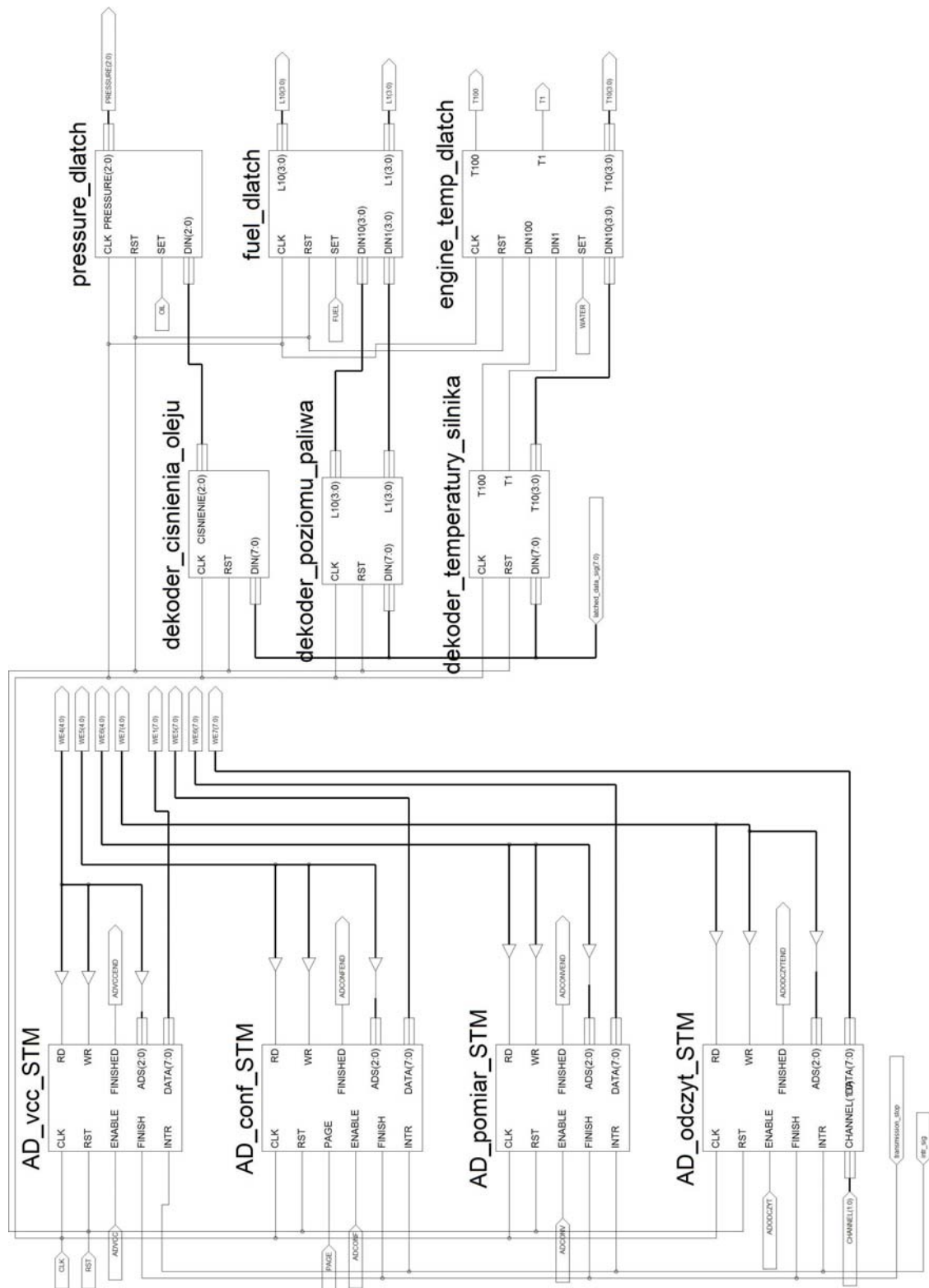
Przetwornik analogowo cyfrowy zamienia mierzone napięcie na liczbę całkowitą z przedziału od 0 do 255 (przy rozdzielczości 8 bitów) określającą jego wartość. Znając krok, z jakim dokonywany jest pomiar napięcia przez przetwornik analogowo cyfrowy (10 mV przy rozdzielczości 8 bitów), oraz wartość odczytaną z pamięci przetwornika można obliczyć wartość mierzonego napięcia. W celu uproszczenia sposobu dekodowania wartości odczytywanych z pamięci przetwornika, na litry paliwa, temperaturę oraz ciśnienie, dekodery:

dekoder_cisnienia_oleju,

dekoder_poziomu_paliwa,

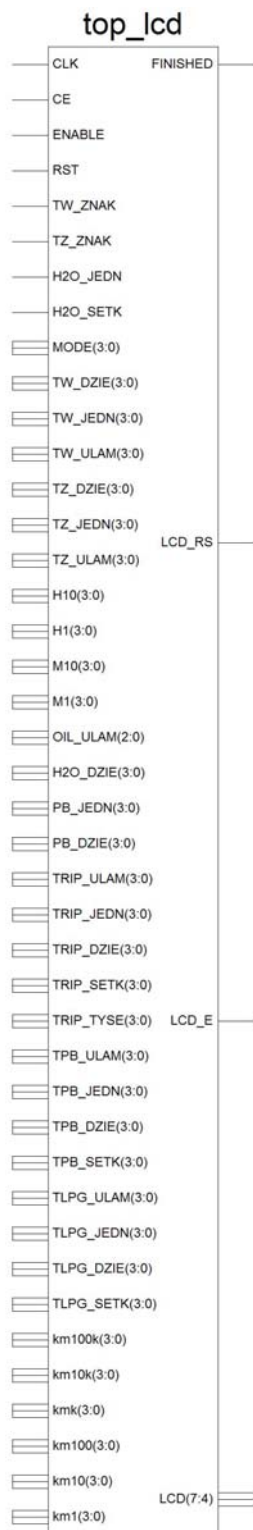
dekoder_temperatury_silnika,

dekodują wartości odebrane z przetwornika na wartości gotowe do wyświetlenia. Odpowiednio: ciśnienie oleju przyjmuje wartości z przedziału 0.0 do 0.7 Pa, z rozdzielczością 0.1 Pa; poziom paliwa w baku od 0 do 45 litrów; temperatura silnika w przedziale od 40°C do 130°C z rozdzielczością 5°C.



Rysunek 31. Schemat blokowy układu obsługującego przetwornik a/c

Sterownik wyświetlacza LCD

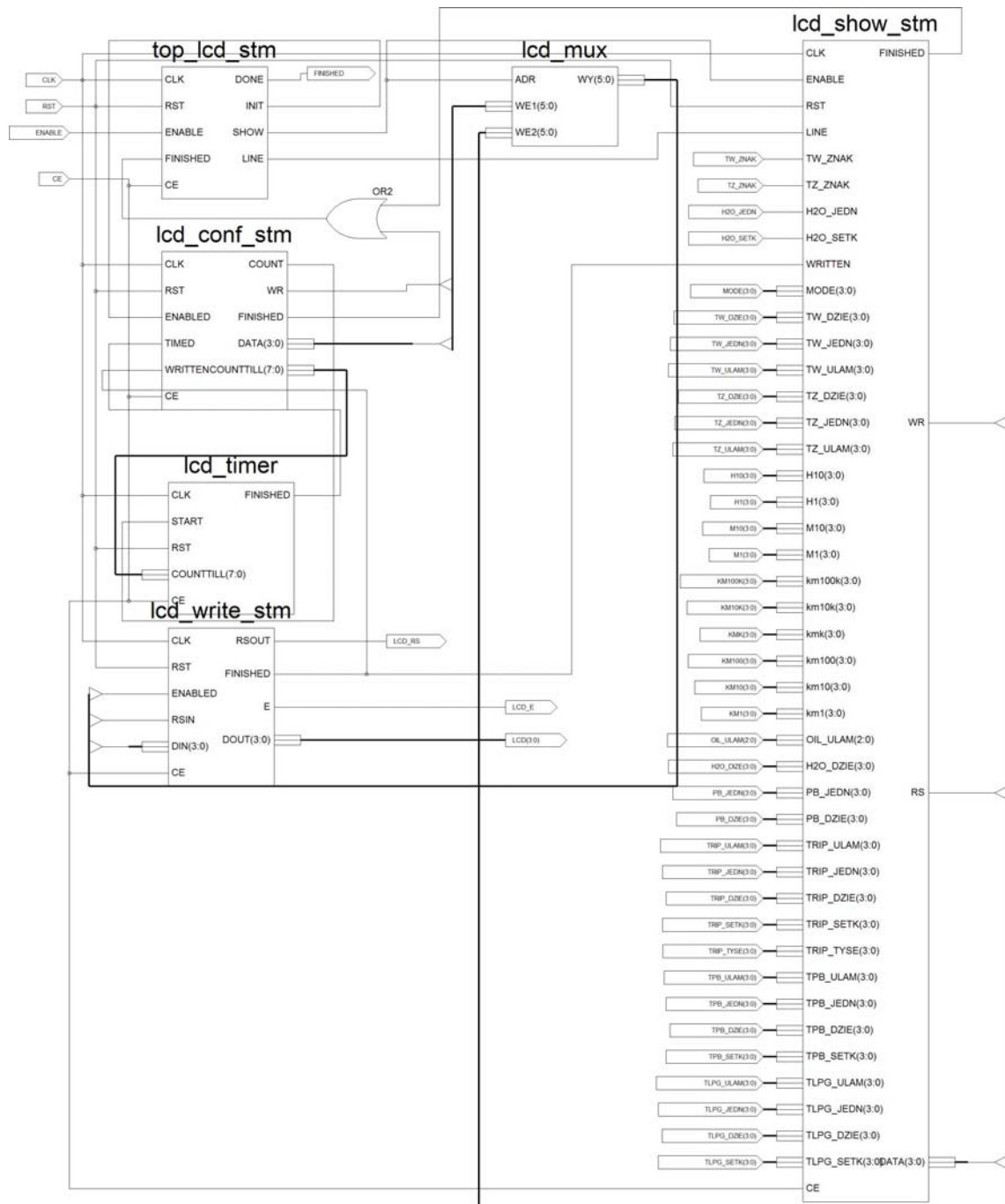


Rysunek 32. Komponent top_lcd

Jest to komponent sterujący wyświetlaczem LCD. Uruchomienie następuje po podaniu stanu aktywnego „1” na linii ENABLE. Działanie sterownika polega na

inicjalizacji kontrolera Hitachi HD44780U, oraz wyświetleniu dwóch linii znaków. Zakończenie komunikacji sygnalizowane jest stanem „1” na linii FINISHED.

Wejścia LCD_E oraz LCD_RS są odpowiadają wejściom wyświetlacza LCD. Magistrala 4 bitowa LCD, służy do komunikacji z wyświetlaczem LCD. Pozostałe wejścia (głównie 4 bitowe) służą do podania wartości różnych danych wyświetlanych na wyświetlacz. Linia MODE, służy do wyboru trybu wyświetlania (powitanie, zegar, temperatura wewnętrzna, temp. zewnętrzna, parametry pracy silnika, liczniki kilometrów).



Rysunek 33. Schemat blokowy układu obsługującego wyświetlacz LCD

Głównym elementem sterującym wyświetlaniem, a za razem pracą pozostałych elementów podrzędnych jest maszyna stanów *top_lcd_stm*. Jest to automat Moore'a. Jego zadaniem jest uruchomienie maszyny stanów odpowiedzialnej za inicjalizację sterownika HITACHI, reset programowy sterownika – *lcd_conf_stm*, ustawienie trybu komunikacji z układem, na 4 bitową magistralę, wybranie rozmiaru znaków 5x8 pikseli, ustawienie parametrów kursora.

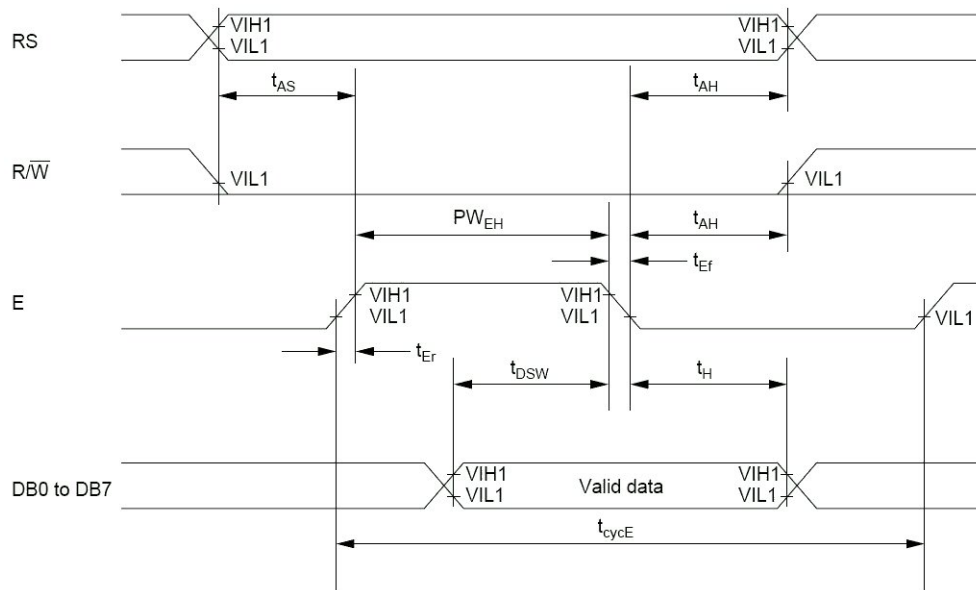
Podczas inicjalizacji, wymagane są odpowiednie odstępy czasowe. Do tego celu służy licznik *lcd_timer*, który odlicza zadaną poprzez wektor COUNTTILL, ilość

taktów zegara CLK. Zakończenie odmierzenia jest sygnalizowane stanem „1” na linii FINISHED. Za uruchamianie licznika odpowiada bezpośrednio komponent *lcd_conf_stm*. Inicjalizacja wyświetlacza, tzw. reset, przeprowadzana jest tylko przy pierwszym przejściu przez automat *top_lcd_stm*. W kolejnych krokach wyświetlane są na przemian dwie linie.

Automat *lcd_show_stm* ma za zadanie wysłanie szesnastu znaków bezpośrednio do kontrolera wyświetlacza LCD. Ponieważ zastosowany wyświetlacz ma dwie linie, układ *lcd_show_stm* uruchomiony zostaje dwukrotnie. Wybór linii do wyświetlenia dokonany zostaje przez wysterowanie sygnału LINE. Rozwiązane jest to wysłaniem komendy ustawiającej kursor na pierwszy znak pierwszej lub drugiej linii.

Wybór trybu wyświetlania dokonywany jest za pomocą sygnału *MODE*(3 do 0). Wewnątrz automatu jest to rozwiązane poprzez multiplekser grupowy. Poszczególne linie trybów wyświetlania są zadeklarowane jako sygnały 128 bitowe (16 znaków, każdy po 8 bitów) Porty wejściowe automatu (*TW_DZIE*; *W_JEDN*; *TW_ULAM*; *TW_ZNAK*; *TZ_DZIE*; *TZ_JEDN*; *TZ_ULAM*; *TZ_ZNAK*; *TRIP_TYSE*; *TRIP_SETK*; *TRIP_DZIE*; *TRIP_JEDN*; *TRIP_ULAM*; *TPB_SETK*; *TPB_DZIE*; *TPB_JEDN*; *TPB_ULAM*; *T LPG_SETK*; *T LPG_DZIE*; *T LPG_JEDN*; *T LPG_ULAM*; *OIL_ULAM*; *H2O_SETK*; *H2O_DZIE*; *H2O_JEDN*; *PB_DZIE*; *PB_JEDN*; *H10*; *H1*; *M10*; *M1*; *km100k*; *km10k*; *kmk*; *km100*; *km10*; *km1*), zawierające pojedyncze wartości poszczególnych komunikatów, są podłączone z odpowiednimi fragmentami 128 bitowych wektorów.

Generowaniem odpowiednich przebiegów czasowych na liniach LCD_RS, oraz LCD_E, zajmuje się automat *lcd_write_stm*. Przykład takiego przebiegu przedstawia rysunek 34.

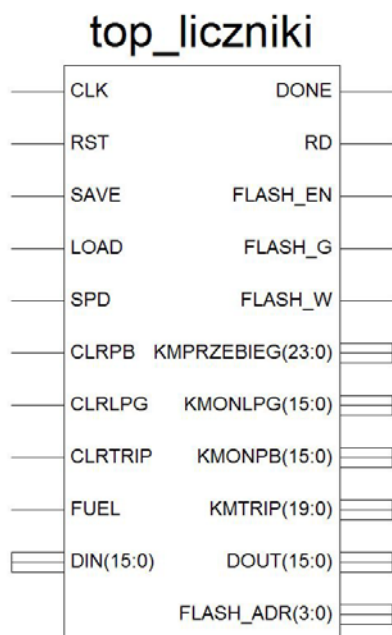


Rysunek 34. Przebiegi czasowe operacji zapisu do wyświetlacza LCD²²

Na schemacie przedstawiono linię R/W, pozwala ona decydować, czy dokonujemy odczytu, czy zapisu. Wykorzystując sterownik wyświetlacza LCD wyłącznie do wyświetlania, możemy tę linię pominąć. Może ona być ustawiona na stałe „0” lub „wisieć”, ponieważ sterownik automatycznie podciąga ją do „0”.

²² Źródło: 44780.pdf strona 58(13)

Liczniki kilometrów



Rysunek 35. Komponent top_liczniki

Element odpowiedzialny za zliczanie przejechanych kilometrów. Składa się z 4 liczników: całkowitego przebiegu pojazdu (6 cyfrowy, maksymalnie 999999 km); licznik przejechanych kilometrów na benzynie (999,9 km), bliźniaczy licznik kilometrów przejechanych na gazie płynnym LPG, oraz dodatkowy licznik umożliwiający zliczenie 9999 kilometrów. Wszystkie liczniki oprócz głównego przebiegu posiadają możliwość kasowania ich zawartości.

Moduł odpowiada również za zapis i ładowanie zawartości liczników do/z pamięci nieulotnej typu *flash* (M28W640). Ze względu na ograniczoną ilość cykli zapisu pojedynczego bloku pamięci flash, wynoszącą 100 tysięcy, ładowanie i zapis odbywa się w chwili włączenia i wyłączenia zapłonu poprzez sygnały *SAVE* i *LOAD*. Zakończenie jest potwierdzone stanem aktywnym na wyjściu *DONE*.

Ponieważ wyświetlacz LCD, wykorzystuje część magistrali podłączonej do pamięci, konieczne było zastosowanie odpowiedniego sterowania modułem liczników i wyświetlania. Dodatkowym utrudnieniem było używanie przez oba moduły różnych sygnałów zegarowych.

Ze względu na dwukierunkowość linii danych, wyjście *RD* odpowiednio steruje multiplexerem przełączającym magistralę. Ponieważ podczas zapisywania stanu

liczników wykonywane są również operacje odczytu z pamięci. Nie możliwe było sterowanie multiplekserem, poprzez sygnały zezwalające na pracę sterownika pamięci.

Wyjścia *KMPRZEBIEG*, *KMONLPG*, *KMONPB*, *KMTRIP* są wektorami, odpowiadającymi zawartości liczników. Długość wektorów odpowiada ilości cyfr w odpowiednim liczniku.

Tabela 5. Organizacja rejestru kilometrów przejechanych

Lp.	<i>KMPRZEBIEG</i> bity	Cyfra dziesiętna
1.	23 do 20	$\times 10^5$
2.	19 do 16	$\times 10^4$
3.	15 do 12	$\times 10^3$
4.	11 do 8	$\times 100$
5.	7 do 4	$\times 10$
6.	3 do 0	$\times 1$

Tabela 6. Organizacja rejestru kilometrów przejechanych

Lp.	<i>KMTRIP</i> bity	Cyfra dziesiętna
1.	19 do 16	$\times 1000$
2.	15 do 12	$\times 100$
3.	11 do 8	$\times 10$
4.	7 do 4	$\times 1$
5.	3 do 0	$\times 0,1$

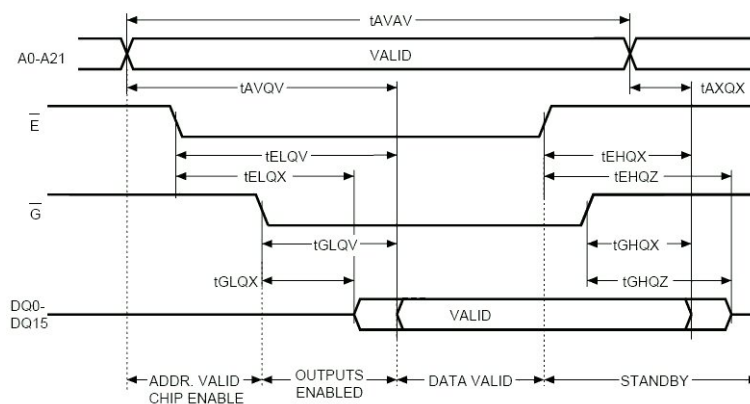
Tabela 7. Organizacja rejestru kilometrów przejechanych

Lp.	<i>KMONPB</i> / <i>KMONLPG</i> bity	Cyfra dziesiętna
1.	15 do 12	$\times 100$
2.	11 do 8	$\times 10$
3.	7 do 4	$\times 1$
4.	3 do 0	$\times 0,1$

1. Sterownik pamięci flash

Aby zapis i odczyt z pamięci był możliwy, konieczny jest pewien ciąg instrukcji. Generowaniem odpowiednich sekwencji sygnałów, zajmuje się element *top_flash*. Posiada on takie same wyjścia jak układ pamięci: *FLASH_EN*, *FLASH_G*, *FLASH_W*. Moduł posiada również flagę zakończenia komunikacji z pamięcią: *FINISH* oraz sygnał *data_valid*, który steruje ładowaniem danych odczytanych z pamięci do rejestrów.

- Za sterowanie odczytem, zapisem, ładowaniem rejestrów i wysterowywaniem wejść multiplekserów, odpowiadają maszyny stanów *LICZNIK_ODCZYT*, *LICZNIK_ZAPIS*. Odczyt z pamięci wymaga jedynie ustawienia adresu (wektor 22 bitowy *FLASH_ADR*), oraz zastosowania odpowiedniej sekwencji czasowej wyjść *FLASH_EN*, oraz *FLASH_G*. Automat *LICZNIK_ODCZYT*, posiada dodatkowo szereg wyjść sterujących odpowiednimi rejestrami. Ponieważ komunikacja z pamięcią odbywa się za pośrednictwem magistrali 16 bitowej, a faktyczna długość wektorów liczników kilometrów jest 33, 26, 22 i 22 bity, odczyt i ładowanie 1 licznika zajmuje maksymalnie 3 operacje odczytu i dodatkowo 1 stan maszyny na załadowanie licznika. Należy tutaj wyjaśnić, dlaczego rzeczywista długość wektorów liczników jest inna od podanej w tabeli. Wynika to z faktu, że jeden obrót linki prędkościomierza (1 impuls), przypada dokładnie jeden metr. Dlatego, aby zachować odpowiednią dokładność liczników, należy zapisywać w pamięci dokładnie całą przebytą drogę, z dokładnością do 1 metra.



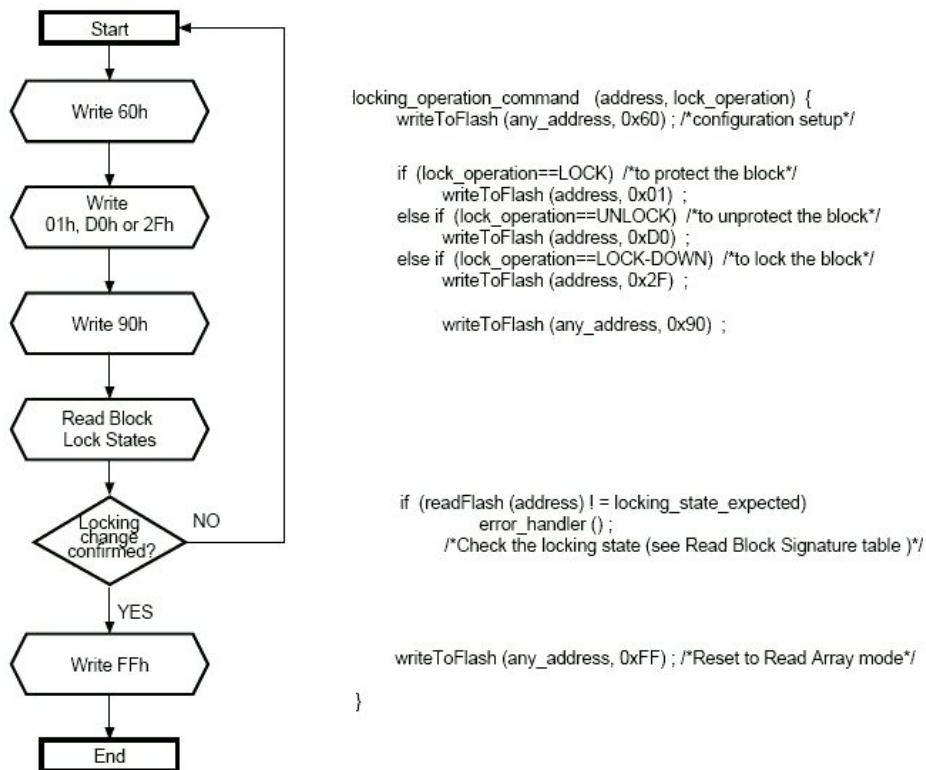
Rysunek 36. Przebieg czasowy operacji odczytu z pamięci M28W640²³

²³ Źródło: M28W640FCT.pdf strona 38(16)

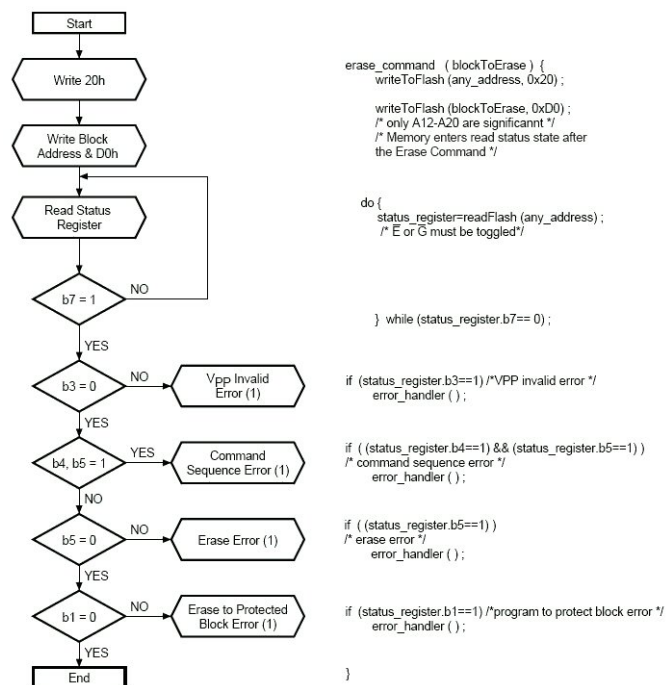
Ponieważ odczyt odbywa się w paczkach 16 bitowych, w układzie zastosowano 3 rejestry tymczasowe TwoByteFlipFlop. Po odczytaniu wszystkich bajtów danego licznika, zawartość rejestrów tymczasowych zostaje załadowana do odpowiedniego licznika. Jeden z rejestrów pełni również funkcję rejestru tymczasowego, podczas operacji zapisu. Do niego zostaje załadowana zawartość rejestru statusu, oraz rejestru zabezpieczeń bloku pamięci.

- Zapis do pamięci jest bardziej skomplikowaną operacją, ponieważ wymaga podania komendy zapisu, danych z adresem, oraz sprawdzenia rejestru statusu. Dodatkowo, przed operacją zapisu należy sprawdzić i odblokować rejestr zabezpieczeń bloku pamięci, oraz wykasować zawartości bloku.

Przykłady operacji odblokowania bloku, kasowania oraz zapisu pojedynczego słowa:



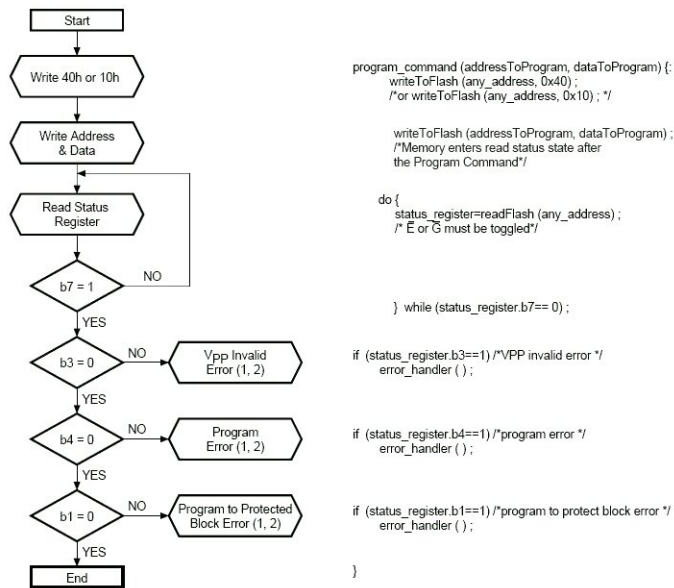
Rysunek 37. Operacja odblokowania bloku pamięci M28W640²⁴



Rysunek 38. Operacja kasowania bloku pamięci M28W640²⁵

²⁴ Źródło: M28W640FCT.pdf strona 70(16)

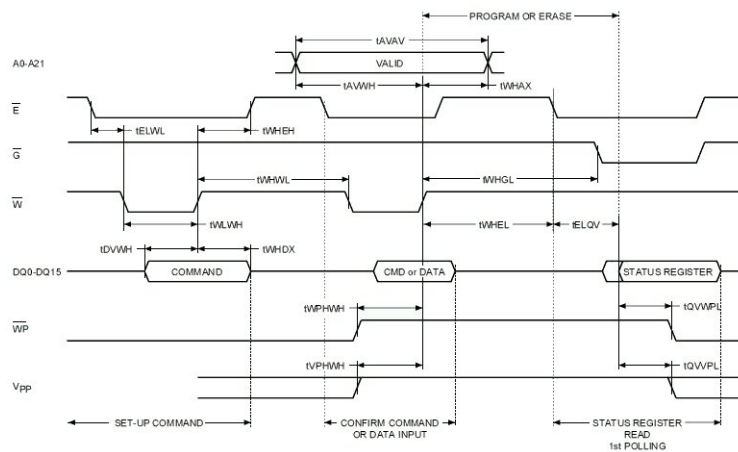
²⁵ Źródło: M28W640FCT.pdf strona 68(16)



Rysunek 39. Operacja zapisu pojedynczego słowa w pamięci M28W640²⁶

Maszyna stanów *LICZNIK_ZAPIS* odpowiada również za wysterowanie (sygnał *DATA_SEL(3 do 0)*) multipleksera (*mux_danych_do_zapisu*), przełączającego wyjścia odpowiednich liczników (co 16 bitów).

Przebiegi czasowe poszczególnych sygnałów podczas operacji zapisu przedstawia poniższy rysunek.



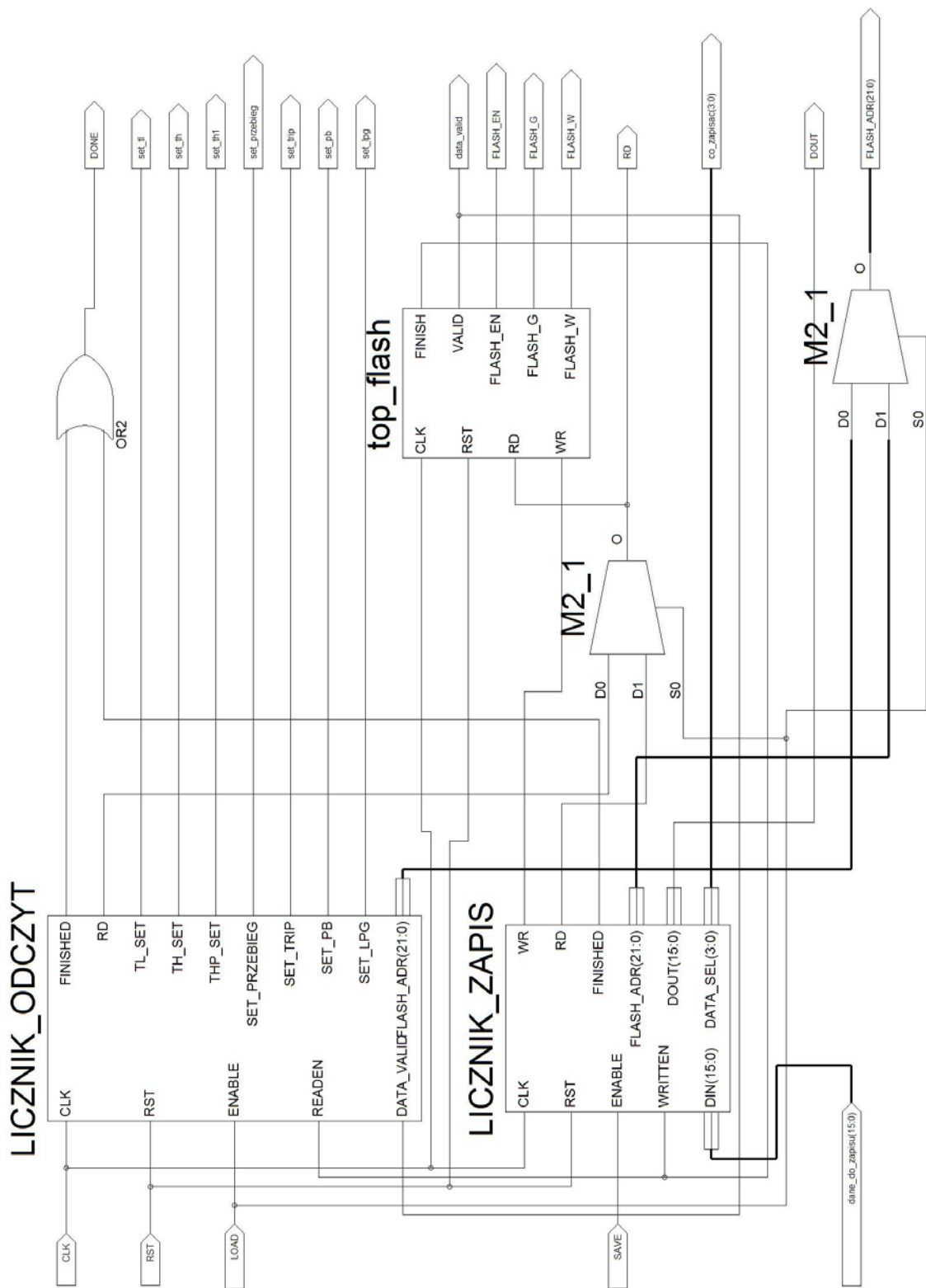
Rysunek 40. Przebieg czasowy operacji zapisu do pamięci M28W640²⁷

Na schemacie z rysunku 40 uwzględniono również linię *WP* (*Write Protect*). Sygnał ten, odpowiedzialny jest za sprzętowe zabezpieczenie bloku pamięci,

²⁶ Źródło: M28W640FCT.pdf strona 64(16)

²⁷ Źródło: M28W640FCT.pdf strona 39(16)

przed wykasowaniem. Ponieważ w projekcie nie jest wykorzystane kodowanie bloków, linia ta jest zawsze w stanie wysokim „1”.



Rysunek 41. Schemat blokowy komponentów sterujących pamięcią M28W640

2. Liczniki kilometrów

Schemat z rysunku 42 przedstawia połączenia sygnałów pomiędzy poszczególnymi licznikami kilometrów, multiplekserem przełączającym dane do zapisu, oraz rejestrami tymczasowymi. Na schemacie widać porty *CLR* trzech liczników. Służą one do kasowania zawartości liczników. Kasowanie odbywa się poprzez naciśnięcie przycisku „SETM” na desce rozdzielczej pojazdu. Ze względu na uproszczenie schematu, pominięto demultiplekser przełączający sygnał *CLR* pomiędzy odpowiednie liczniki. Sygnałem sterującym demultiplekserem jest sygnał *MODE(3 do 0)*, jest to ten sam sygnał, który steruje wyborem trybu wyświetlania.

W celu uniknięcia błędów zapisu, polegających na zwiększeniu zawartości liczników kilometrów podczas zapisywania ich stanu, impulsy zliczane przez liczniki są funkcją sygnałów *SPD*, *LOAD* i *SAVE*. Funkcję tą przedstawia tabela 8.

Tabela 8. Tabela prawdy ładowania oraz zapisania stanu liczników

SPD	LOAD	SAVE	f(SP, LOAD, SAVE)	
0	0	0	0	M ₀
1	0	0	1	m ₀
1	0	1	0	M ₁
1	1	0	0	M ₂

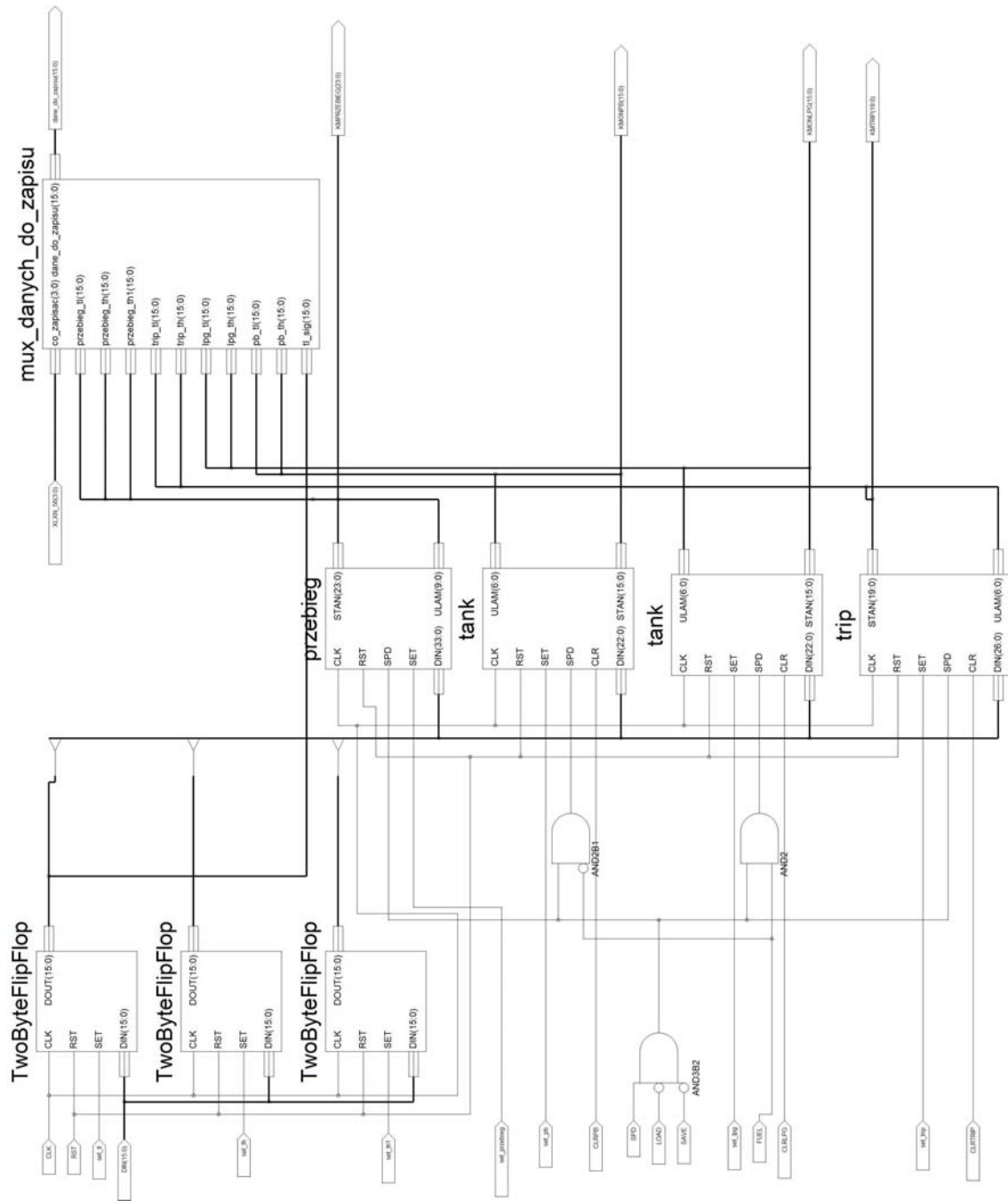
Funkcja ma więc postać iloczynu (Równanie 5):

$$f = SPD * \overline{LOAD} * \overline{SAVE}$$

Równanie 5.

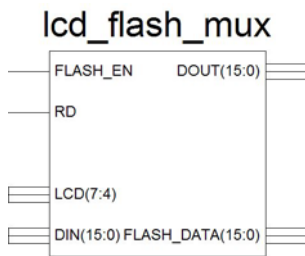
Tabela 8 nie przewiduje przypadku, gdy sygnały *SAVE* i *LOAD* jednocześnie przyjmują wartość jeden, ponieważ taki przypadek nie występuje. Jest to wynik implementacji maszyny stanów sterującej operacjami zapisu i ładowania liczników kilometrów.

Linia *FUEL* służy do odpowiedniego przełączania sygnału z linki prędkościomierza. Odbywa się to poprzez dwie bramki logiczne *AND*. Sygnał *SPD* dociera do licznika kilometrów na benzynie w sytuacji, gdy linia *FUEL* przyjmuje stan wysoki. Natomiast do licznika kilometrów na gazie płynnym sygnał *SPD* jest, ziloczynowany z negacją sygnału *SPD*



Rysunek 42. Schemat blokowy liczników kilometrów

Multiplexer magistrali danych



Rysunek 43. Komponent lcd_flash_mux

Jednostka jest odpowiedzialna za przełączanie sygnałów wejściowych *LCD(7 do 4)* oraz *DIN(15 do 0)* do wyjścia *FLASH_DATA (15 do 0)*. Sterowanie odbywa się poprzez ustawienie odpowiedniego stanu na wejście *FLASH_EN* oraz *RD*. Podczas wykonywania odczytu z portu *DOUT*, nie jest konieczne ustawianie stanu na linii *FLASH_EN* oraz *RD*.

Tabela 9. Tabela prawdy multiplexera przełączającego dane do zapisu

FLASH_EN	RD	FLASH_DATA(15 do 0)	DOUT
0	0	(15 do 8) <= "00000000" (7 do 4) <= LCD(7 do 4) (3 do 0) <= "0000"	FLASH_DATA
1	0	DIN(15 do 0)	
1	1	ZZZZZZZZZZZZZZZZ	

Tabela 9 nie zawiera wszystkich możliwych kombinacji stanów na wejściach *FLASH_EN* oraz *RD*, jedynie występujące w projekcie. Pozostałe kombinacje nie są istotnie. Automat sterujący multiplexserem – *ignition_ster* umożliwia jedynie wystąpienie stanów przedstawionych w tabeli 9.

Zegar



Rysunek 44. Komponent ZEGAR

Komponent składa się z pięciu liczników dziesiętnych, połączonych kaskadowo. Pierwszy licznik zlicza impulsy z linii *SECOND*, generowane co 12500000 taktów generatora kwarcowego 12,5 MHz. Elementem odpowiedzialnym za generowanie tych sygnałów jest komponent *CLK_DIV*, wystawiający stan wysoki w momencie zliczenia zadanej wartości taktów.

Kolejne liczniki liczą odpowiednio jedności minut, dziesiątki minut, jedności godzin oraz dziesiątki godzin.

Zliczenie 59 sekund powoduje inkrementację jednostek minut. Gdy liczba minut jest równa 9, w kolejnym kroku zwiększona zostaje liczba dziesiątek minut.

W przypadku godzin, implementacja licznika zliczającego jedności godzin jest bardziej skomplikowana, ponieważ do godziny 19, cyfra jednostek może przyjmować wartość 9, natomiast po godzinie 20 tylko wartość 3.

Ustawianie zegara polega na podaniu stanu wysokiego, na liniach *MININC* lub *HOURINC*. Ponieważ głównym sygnałem zezwalającym (clock enable) dla wszystkich liczników jest stan wysoki na linii *SECOND*, linie *MININC* oraz *HOURINC* mogą być podpięte bezpośrednio do pinów układu FPGA. Nie jest konieczne użycie dodatkowego komponentu odfiltrowującego drgania styków.

Sterownik wycieraczek



Rysunek 45. Komponent top_wipes

Zaimplementowany w projekcie sterownik wycieraczek odróżnia się od standardowo spotykanych w samochodach możliwością zaprogramowania odstępów czasu pomiędzy ruchami wycieraczek.

Działanie polega na mierzeniu czasu od pierwszego uruchomienia wycieraczek przez kierowcę, do kolejnego. Dzięki temu możemy ustawić okres pracy wycieraczek odpowiedni do warunków atmosferycznych. Zastosowane liczniki 8 bitowe, pozwalają na pracę w odstępach maksymalnie 4,25 min (255 sekund). Praca sterownika jest sygnalizowana za pomocą dwóch diod led. Wyłączenie sterownika następuje po trzecim uruchomieniu wycieraczek ręcznie.

Tabela 10. Algorytm działania sterownika wycieraczek

Lp	Przełącznik wycieraczek	Stan
1.	1 uruchomienie	Programowanie sterownika
2.	2 uruchomienie	Zaprogramowanie
3.	wyłączony	Praca automatyczna
4.	3 uruchomienie	Wyłączenie sterownika

Głównymi elementami sterownika są dwa liczniki binarne (*wipes_counter*, *wipes_ster_counter*) oraz maszyna stanów Moore'a (*wipes_ster_stm*).

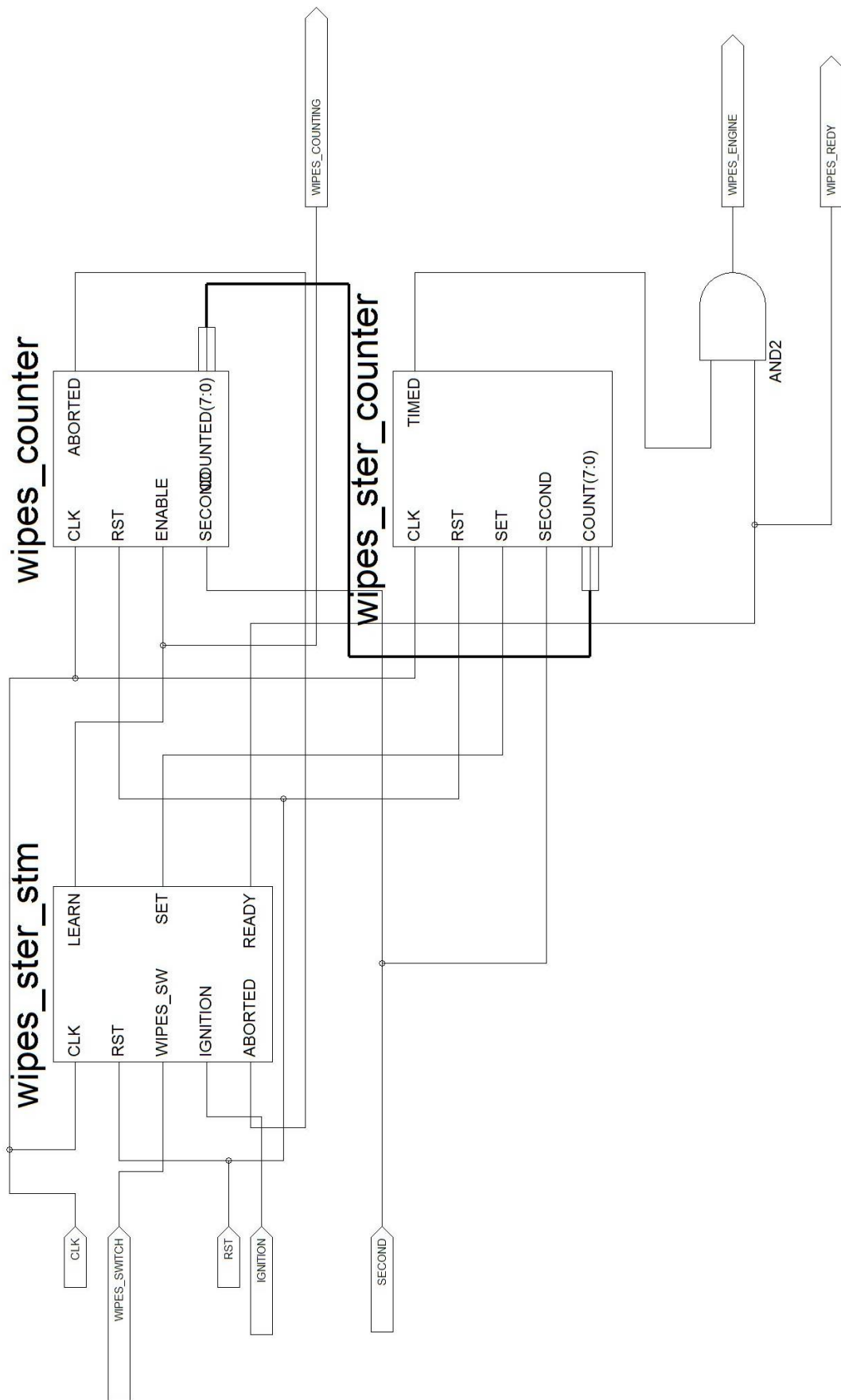
Licznik *wipes_counter*, jest odpowiedzialny za zliczanie impulsów (sekund) podczas programowania sterownika. W przypadku przekroczenia maksymalnego czasu do zaprogramowania (przepełnienie licznika), wystawia na wyjściu *ABORTED* stan '1'. Zawartość licznika przekazywana jest poprzez wektor ośmiobitowy *COUNTED*, do licznika sterującego *wipes_ster_counter*.

Automat stanów zarządzający pracą sterownika, wystawia sygnał zezwolenia dla licznika odpowiedzialnego za programowanie sterownika. Licznik *wipes_ster_counter*, pracuje w trybie ciągłym. Wystawia stan '1' na wyjściu TIMED, gdy wartość zapisana w wewnętrznym rejestrze odpowiada wartości zliczonej. Jednak do uruchomienia wycieraczek, jest wymagany sygnał zezwolenia READY. Oba sygnały są ziloczynowane.

Jako wejścia sterownika, służące do programowania, wykorzystywane są wejścia *WIPES_SWITCH* (sygnał '1' z włącznika wycieraczek); *IGNITION* (włącznik zapłonu) oraz sygnał *SECOND* (ten sam, co w komponencie zegara). Sygnał z instalacji zapłonowej pojazdu, służy do wyłączenia całego sterownika, w przypadku wyłączenia zapłonu (zatrzymanie silnika pojazdu).

Sygnał z włącznika wycieraczek, przechodzi przez „debouncer” *top_wlacznik*, pokazany na głównym schemacie projektu (rysunek 15).

Ciekawym rozwiązaniem problemu sterowania silnikiem wycieraczek jest generowanie przez sterownik wystarczająco długiego sygnału uruchomienia silnika. Mechanizm wycieraczek w samochodzie wymaga jedynie podania impulsu +12 woltów na jedno z wejść silnika, aby silnik wykonał część obrotu. Dalszy obrót silnika dokonywany jest przez „mechanizm powrotny”. Jest to rozwiązane poprzez ścieżki miedzi na trybie przekładni i układ styków. Dzięki takiemu rozwiązaniu, pomimo wyłączenia wycieraczek (ustawienie przełącznika w pozycji neutralnej), wycieraczki zawsze wracają do położenia pierwotnego. Problem rozwiązany został poprzez sumę logiczną sygnału bezpośrednio z włącznika wycieraczek, oraz sygnału ze sterownika. Dlatego użytkownik uruchamiając oraz kończąc programowanie sterownika za każdym razem powoduje, że wycieraczki wykonują pełny cykl ruchu. W samym sterowniku długość sygnału dla uruchomienia wycieraczek wynika z odstępów pomiędzy sygnałem *SECOND*.



Rysunek 46. Schemat blokowy sterownika wycieraczek

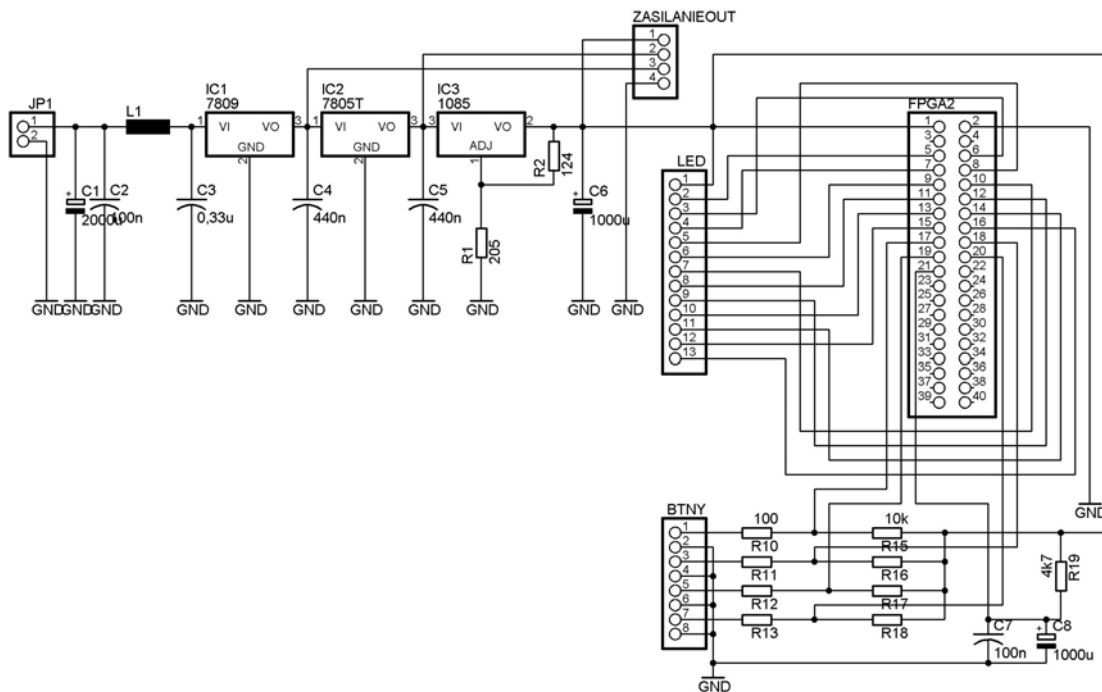
Rozdział III - Dokumentacja techniczna

Wprowadzenie

Projekty schematów elektrycznych układu wykonane zostały w darmowej wersji programu Eagle firmy CadSoft. Producent oprogramowania, w darmowej wersji, ograniczył maksymalny rozmiar projektowanych płytek do szerokości 100 mm oraz wysokości 80 mm.

Schematy elektryczne

Zasilanie, przyciski, wyświetlacze siedmiosegmentowe led



Rysunek 47. Schemat elektryczny płytki zasilania

Rysunek 47 przedstawia schemat płytki zasilającej układ fpga oraz inne peryferia. Wejście JP3 jest punktem podłączenia zasilania +12V z instalacji samochodu. Kondensatory C1, C2 oraz cewka L1 pełnią rolę filtra. Układy U1, U2 oraz U3 to stabilizatory 7809, 7805 oraz 1085. Są źródłami napięć odpowiednio 9V, 5V oraz 3,3V. Ze względu na duży pobór prądu przez cały układ stabilizatory są w wersjach do 3 A. Stabilizator 1085 jest odpowiednikiem 317, ale na 3 ampery.

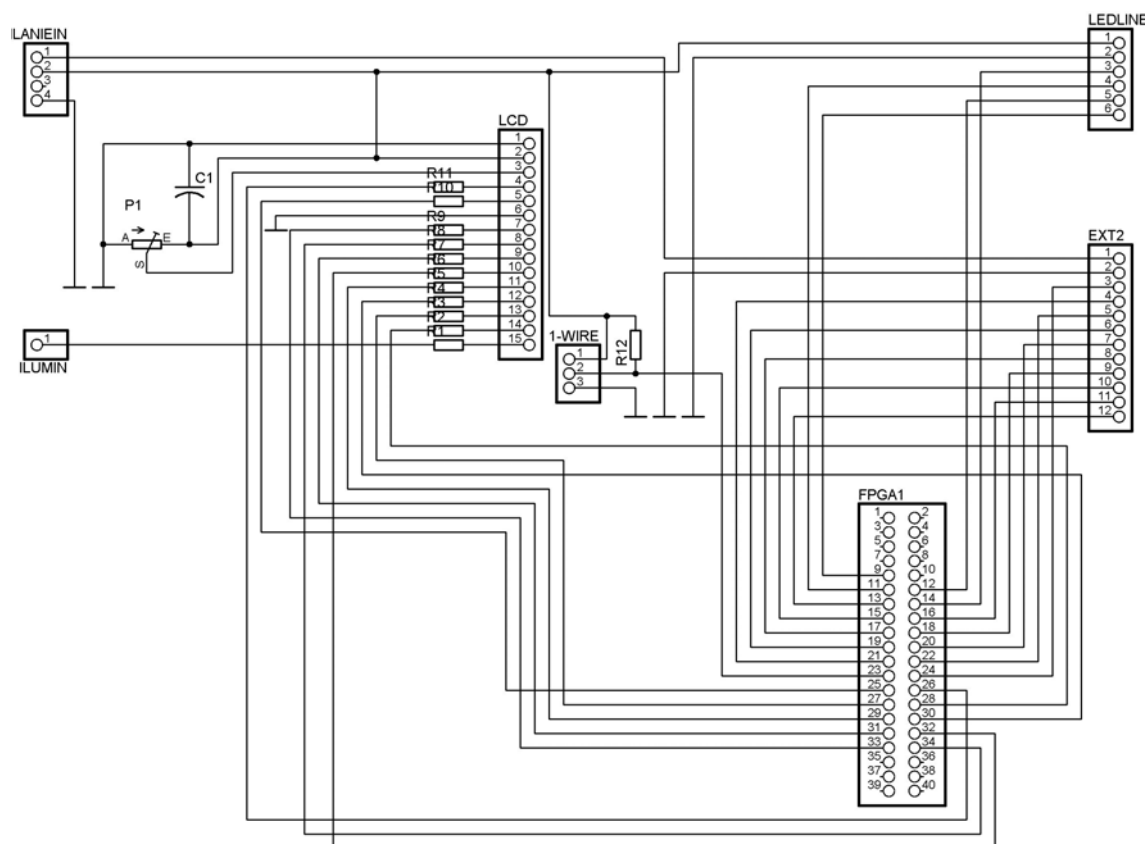
Złącze LED służy do podłączenia anod oraz katod wyświetlaczy siedmiosegmentowych.

Klawisze ustawiania zegara, wyboru trybu wyświetlania oraz kasowania liczników podłączone są do złącza BTNY. Jak widać, wejścia układu fpga podłączone są do przycisków poprzez rezystory 10k podciągnięte do zasilania. Wciśnięcie któregoś przycisku powoduje ściągnięcie linii do masy poprzez rezystor 100 ohm.

Rezystor R19 oraz kondensator C8 pełnią rolę układu resetującego. Po włączeniu zasilania, kondensator ładuje się poprzez rezystor R19. Pojemność kondensatora oraz rezystora dobrane zostały tak, aby napięcie na okładkach kondensatora równe $0,7 \cdot V_{cc}$ (stan wysoki dla standardu LVTTL wejść układu FPGA) osiągnięte było po czasie około jednej sekundy, czyli już po załadowaniu konfiguracji układu.

Gniazdo FPGA2 odpowiada złączu J2 na płycie minimodułu MMfpga12. ZASILANIEOUT służy do podłączenia zasilania pozostałych elementów.

LCD, 1 Wire[®]



Rysunek 48. Schemat elektryczny płytki podłączenia wyświetlacza LCD

Rysunek 48 przedstawia drugą płytkę połączoną z układem MMfpga12 poprzez złącze FPGA1.

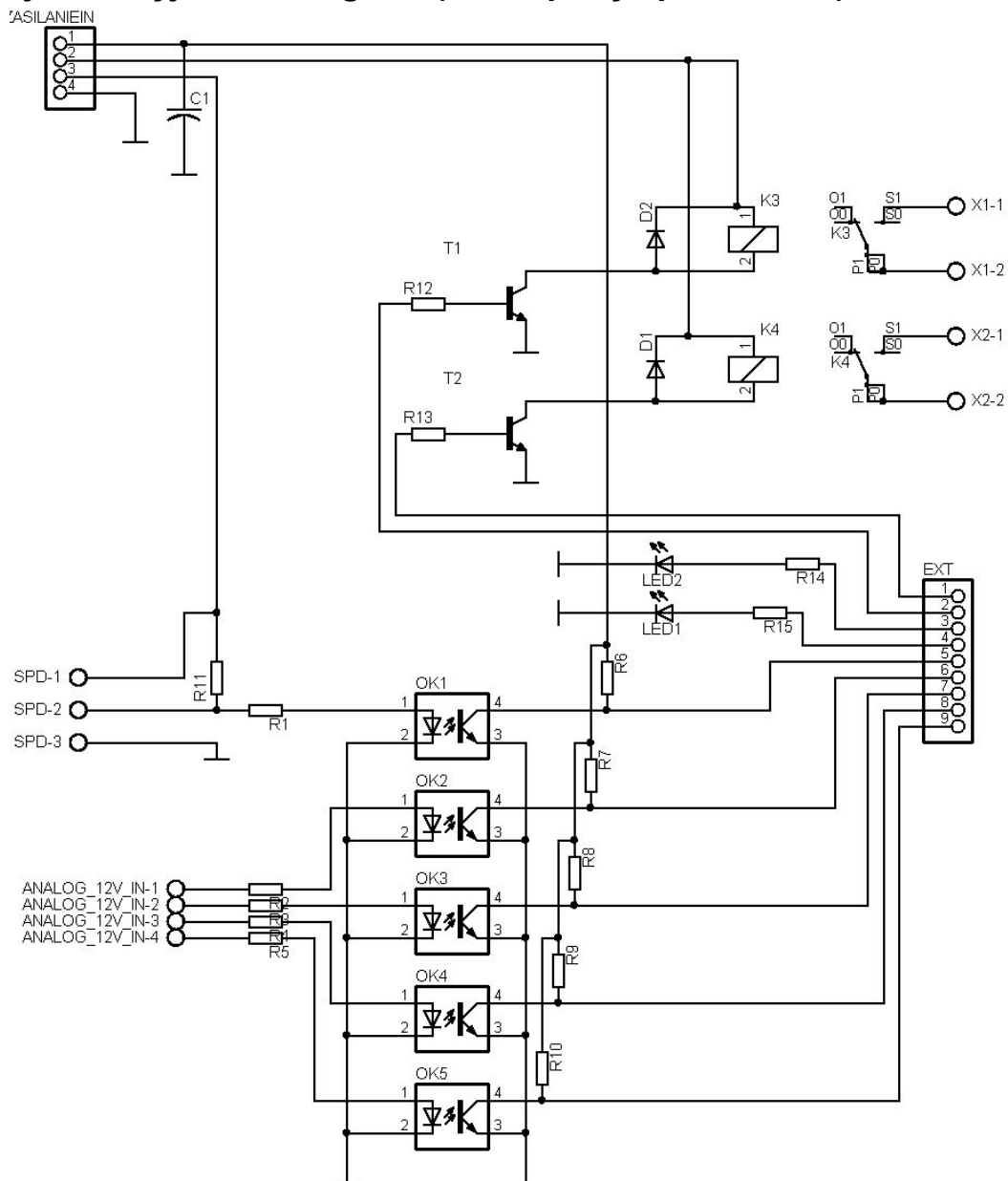
Gniazdo LCD służy do podłączenia wyświetlacza. Ponieważ wyświetlacz korzysta z tych samych wejść układu fpga, do których podłączona jest pamięć flash, wejścia wyświetlacza zabezpieczone zostały rezystorami 100 ohmów. Potencjometr precyzyjny P1 10 k służy do regulacji kontrastu wyświetlacza. Złącze ILUMIN służy do podłączenia zasilania +12V z instalacji samochodu (np. ze świateł pozycyjnych lub z włącznika zapłonu), poprzez rezystor jest połączone z diodą podświetlającą wyświetlacz LCD.

Złącze 1-Wire jest portem magistrali 1 Wire[®], do tego gniazda podłączone jest gniazdo DB-9 na obudowie, służące do podłączenia czujników temperatury DS18B20, oraz przewód łączący z płytką przetwornika analogowo-cyfrowego.

Złącze EXT2 służy do połączenia płytki wejść analogowych oraz wyjść przełączników.

Gniazdo ZASILANIEIN jest wejściem napięć 3.3; 5, oraz GND.

Wejścia i wyjścia analogowe (transoptory i przekaźniki)



Rysunek 49. Schemat płytki wejść analogowych i przekaźników

Rysunek 49 przedstawia płytkę rozszerzeń. Złącze EXT jest odpowiednikiem złącza EXT2 na płytce „LCD, 1Wire”. Gniazdo ZASILANIEIN jest punktem podłączenia zasilania (3.3V, 5V, 9V).

Płytką jest odpowiedzialna za zamianę sygnałów zewnętrznych na sygnały wejściowe bezpieczne dla układu fpga. Rolę izolacji galwanicznej pełnią układy OK1-OK5, transoptory PC817 firmy Sharp. Sygnały wejściowe oraz wyjściowe podłączone są do płytki poprzez listwy ARK.

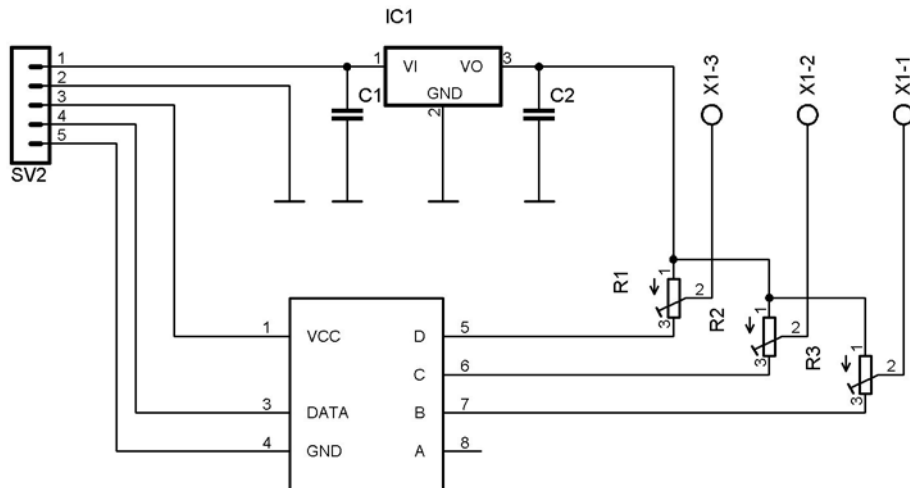
Listwa SPD służy do podłączenia czujnika prędkości jazdy. Złącze ANALOG_IN1, czujnik położenia wału korbowego. ANALOG_IN2, włącznik zapłonu. ANALOG_IN3, elektrozawór benzyny. ANALOG_IN4, włącznik wycieraczek.

Na schemacie (rysunek 49) widoczne są również dwa przekaźniki sterowane 5V ze stykami na 16A; odpowiadają one za zamykanie centralnego zamka oraz uruchamianie silnika wycieraczek.

Diody LED1 oraz LED2 sygnalizują pracę sterownika wycieraczek.

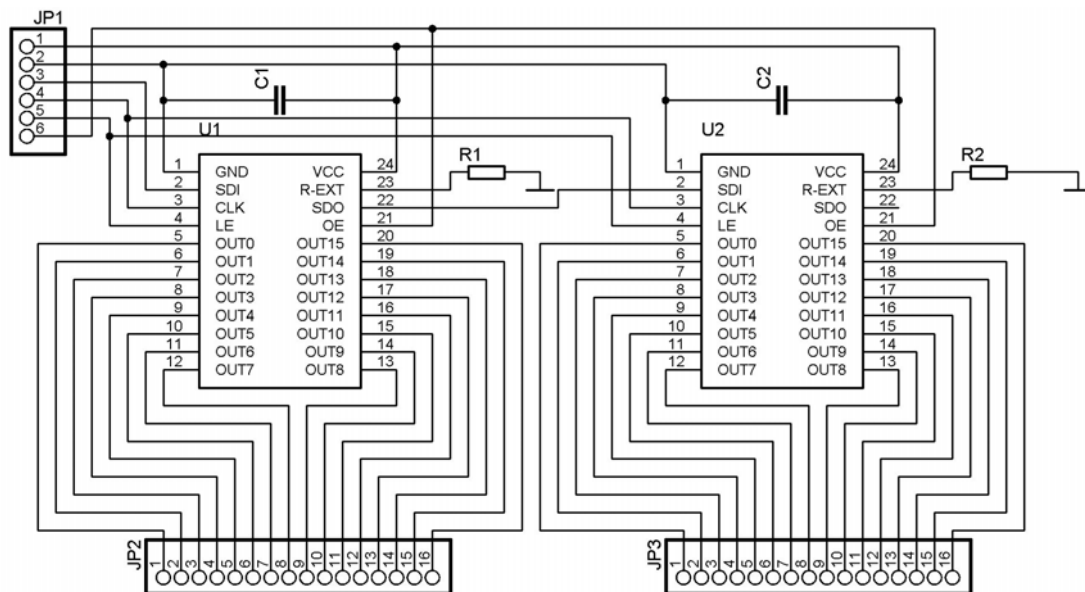
Sterowniki led i przetwornik A/C

Schemat połączeń przetwornika analogowo cyfrowego, przedstawia rysunek 50. Do zasilania czujników służy stabilizator 7805. Czujniki podłączone do złącza ARK X1. Potencjometry precyzyjne 100 ohm, R1, R2, R3 służą do korekty wskazań.



Rysunek 50. Schemat podłączenia przetwornika a/c

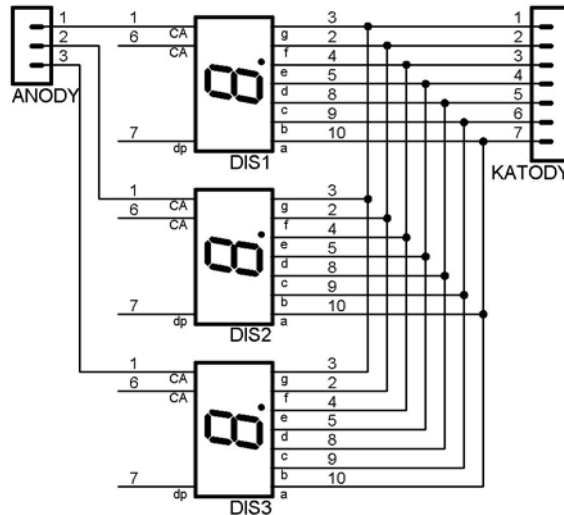
Rysunek 51 przedstawia schemat podłączenia sterowników linii led STP16C596 – układy U1 i U2. Kondensatory C1 i C2 mają wartość 100nF i pełnią rolę zabezpieczającą. Rezystory R1 i R2 mają wartość 1k ohm i służą do regulacji prądu przepływającego przez diody. Ich wartość dobrana została doświadczalnie. Złącze JP1 służy do komunikacji z układem FPGA, połączone jest ze złączem LEDLINE na płycie LCD, 1Wire. Złącza JP2 oraz JP3 służą do podłączenia linii led na płytkach wyświetlaczy prędkości jazdy i obrotów silnika.



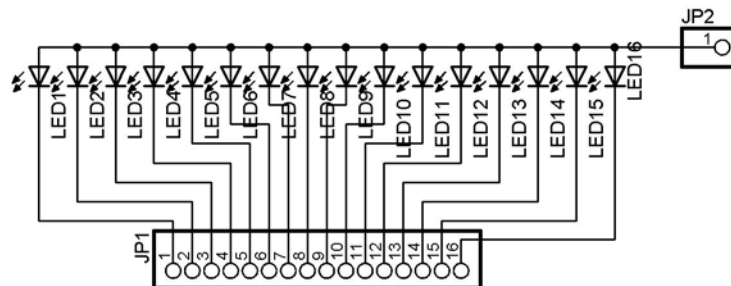
Rysunek 51. Schemat podłączenia sterowników led

Wyświetlacz prędkości jazdy i obrotów silnika

Poniższe schematy (rysunki 52 i 53) dotyczą płytek wyświetlaczy prędkości jazdy oraz prędkości obrotowej silnika. Wyświetlacz prędkości obrotowej jest dwucyfrowy. Segmenty wyświetlaczy mają kolor niebieski. Dlatego nie wymagają rezystorów ograniczających prąd segmentów. Wynika to z faktu, że napięcie progowe struktur świecących na niebiesko wynosi ok. 3.0 - 3.2 V. Elektrody wyświetlaczy (anody i katody) są sterowane bezpośrednio z wyjść FPGA (bez jakichkolwiek buforów), co upraszcza budowę urządzenia i minimalizuje liczbę niezbędnych elementów. Wynika to z faktu, że wydajność prądowa poszczególnych pinów układu Spartan 3 wynosi 24 mA.



Rysunek 52. Schemat podłączenia wyświetlaczy siedmiosegmentowych



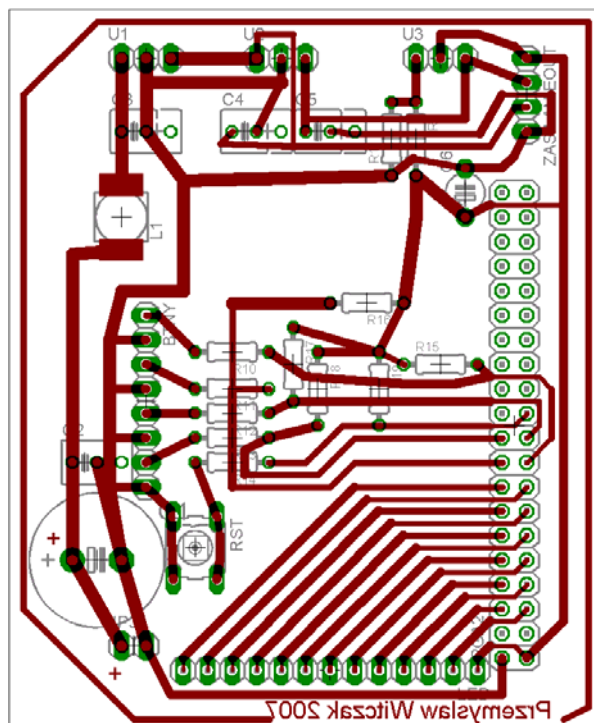
Rysunek 53. Schemat podłączenia linii led

Złącze JP1 odpowiada złączom JP2 i JP3 na płytce sterowników. JP2 jest miejscem podłączenia zasilania 3.3 V.

Projekty PCB

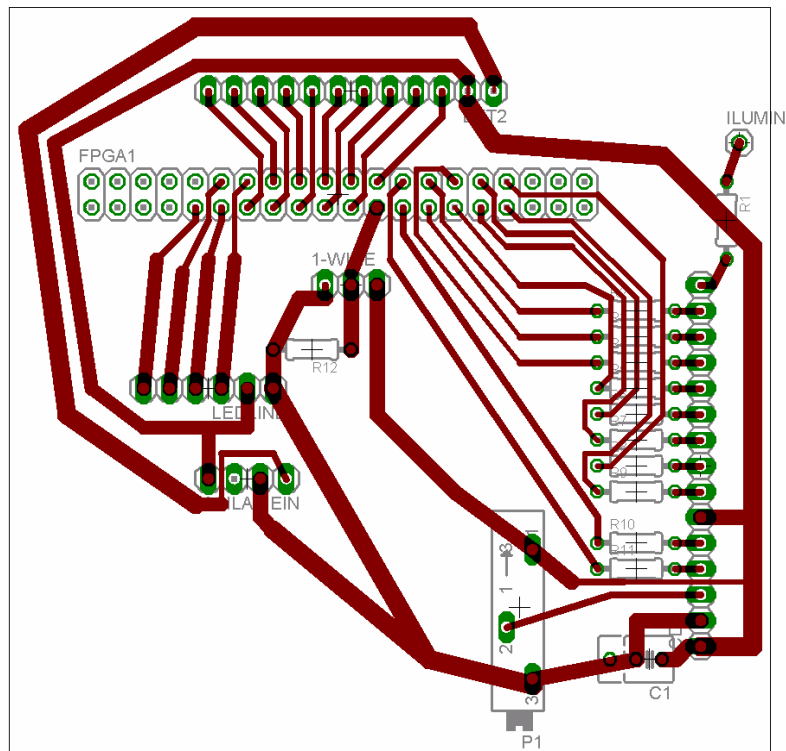
Projekty PCB zaprojektowane zostały również w programie EAGLE. Płytki wykonane na jednostronnym i dwustronnym laminacie szklano-epoksydowym, metodą tzw. termo-transferu. Polega ona na wydrukowaniu projektów płytek na papierze kredowym przy użyciu drukarki laserowej. Następnie za pomocą gorącego żelazka toner z papieru nanosi się na laminat. Kończącym etapem jest wytrawienie niezabezpieczonych tonerem obszarów miedzi. Większość płytek jest dwuwarstwowa, kolorem bordowym oznaczono warstwę spodnią, natomiast turkusowym warstwę wierzchnią. Kolor zielony odpowiada punktom lutowniczym. Rysunki przedstawiają płytki widokiem na wierzchnią warstwę.

Zasilanie, przyciski, wyświetlacze siedmiosegmentowe led



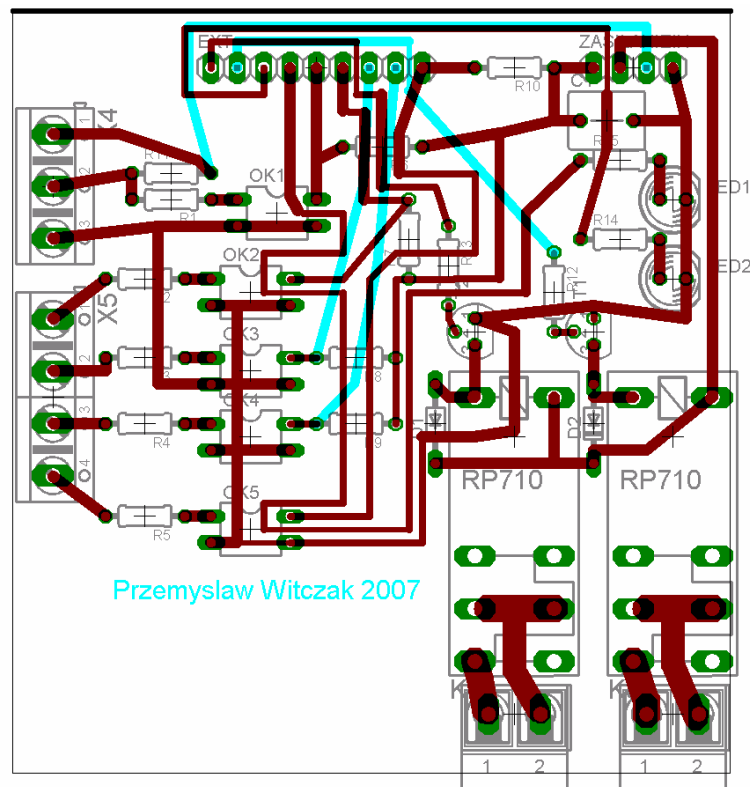
Rysunek 54. Moduł zasilania

LCD, 1 Wire®



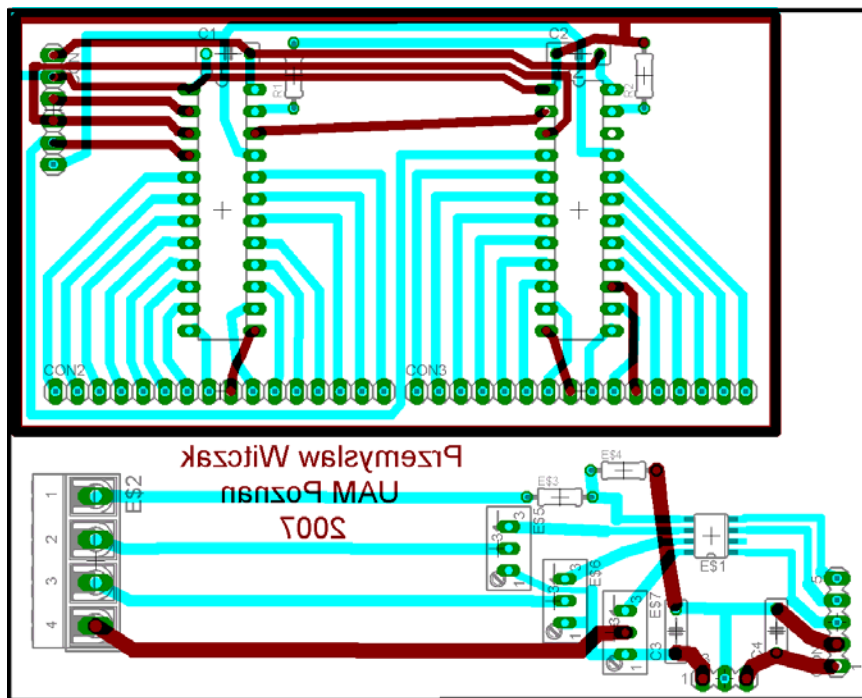
Rysunek 55. Moduł wyświetlacza oraz magistrali 1 Wire®

Wejścia i wyjścia analogowe (transoptory i przekaźniki)



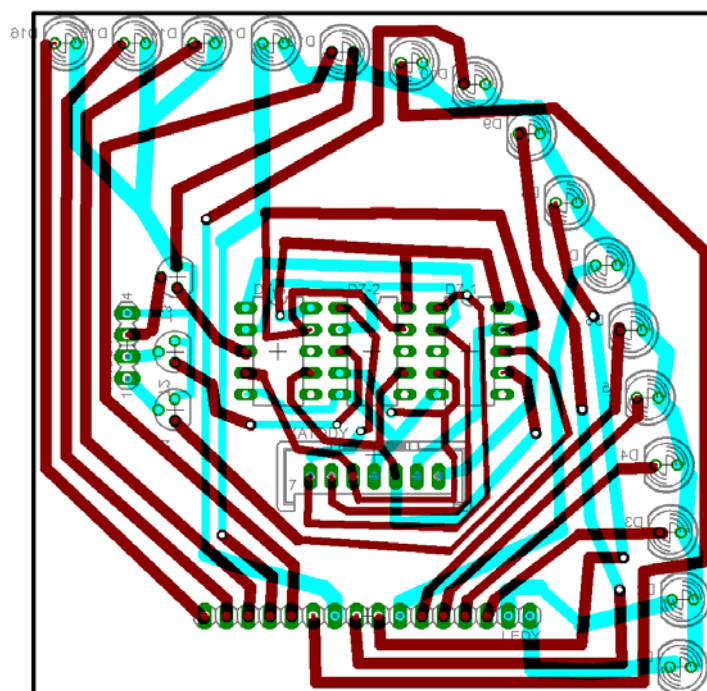
Rysunek 56. Moduł wejść transoptorów i przekaźników

Sterowniki led i przetwornik A/C



Rysunek 57. Moduł sterowników linijek led i przetwornika A/C

Wyświetlacz prędkości jazdy i obrotów silnika

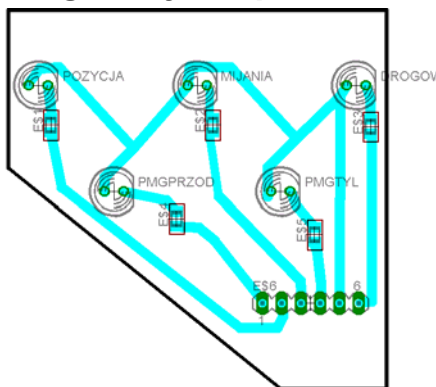


Rysunek 58. Wyświetlacz prędkości jazdy

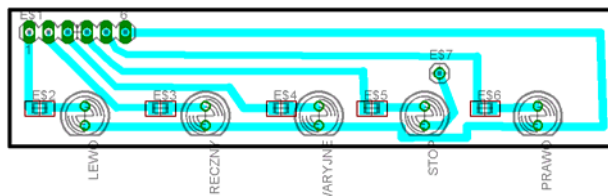


Rysunek 59. Wyświetlacz prędkości obrotowej silnika

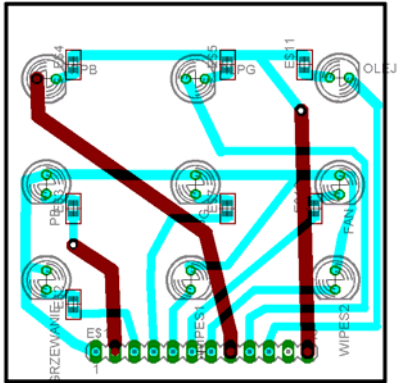
Obwody kontrolek świateł, kierunkowskazów oleju, paliwa, hamulców, świateł ostrzegawczych itp.



Rysunek 60. Moduł stanu świateł zewnętrznych



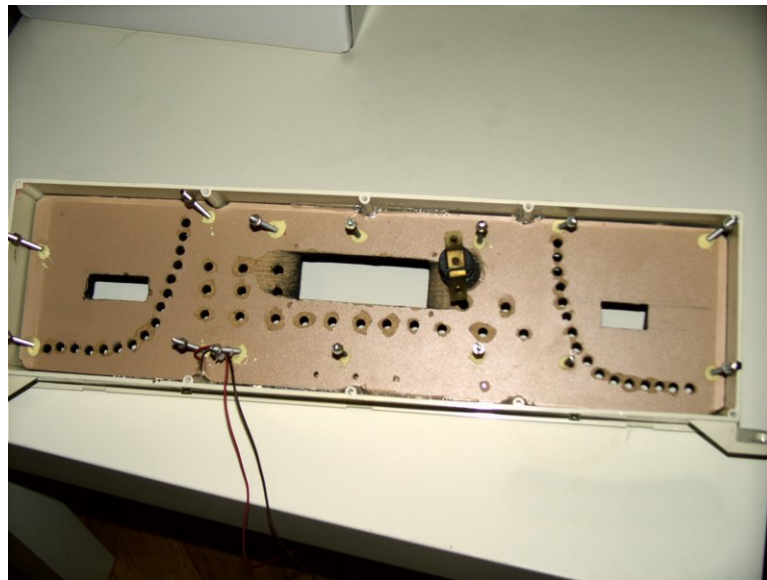
Rysunek 61. Moduł stanu kierunkowskazów, świateł ostrzegawczych oraz świateł stop i hamulca ręcznego



Rysunek 62. Moduł usterki układu smarowania, wyboru paliwa, rezerwy, ogrzewania i wycieraczek

Obudowa

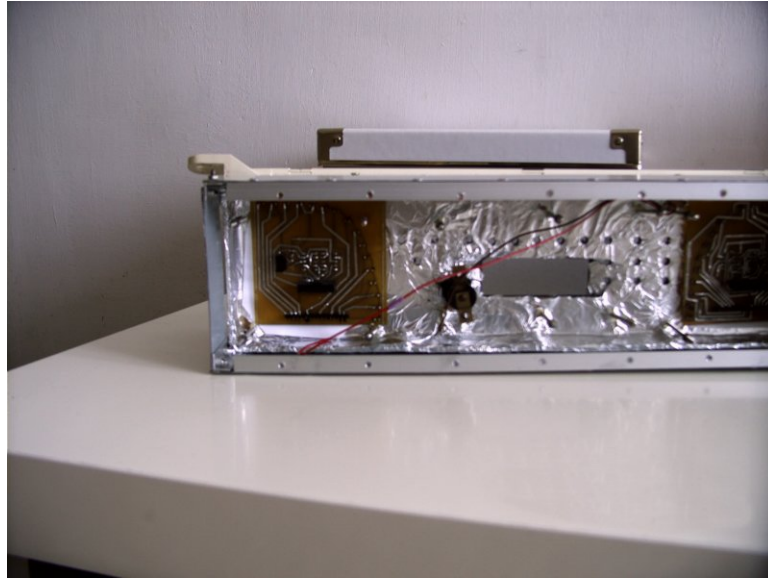
Obudowa wykonana została przy wykorzystaniu fragmentów oryginalnego zestawu wskaźników. Jako panel frontowy posłużyła płyta wiórowa o grubości około 7 mm. Pomalowana została czarnym matowym lakierem w aerozolu. Tylna część obudowy wykonana została z elementów z blachy stalowej, na stelażu z kątowników aluminiowych. Elementy łączone przy użyciu nitów aluminiowych. Kątowniki do obudowy oryginalnych wskaźników przymocowane zostały przy użyciu kleju epoksydowego. Na poniższym zdjęciu (rysunek 63) widać otwory pod wyświetlacze oraz diody kontrolne.



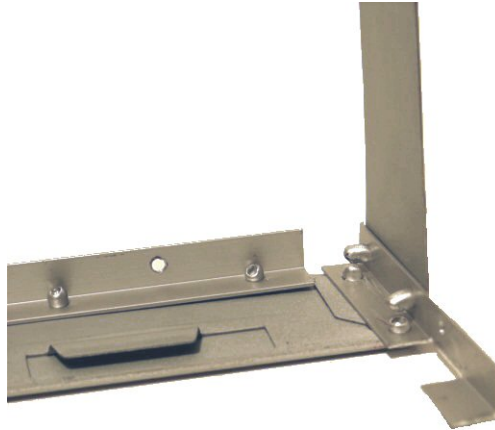
Rysunek 63. Wnętrze obudowy, panel frontowy

Poszczególne moduły znajdujące się w obudowie, zamocowane są dzięki prętom gwintowanym M3. Pręty wklejone są w panel frontowy na klej epoksydowy. Płytki utrzymywane są na swoich miejscach dzięki nakrętkom z obu stron.

W celu zabezpieczenia układów przed skutkami zakłóceń, wewnątrz obudowy od strony panelu frontowego wyklejone zostało folią aluminiową. Dzięki temu cała obudowa tworzy klatkę Faradaya.



Rysunek 64. Ekran, wewnątrz wyklejone folią aluminiową

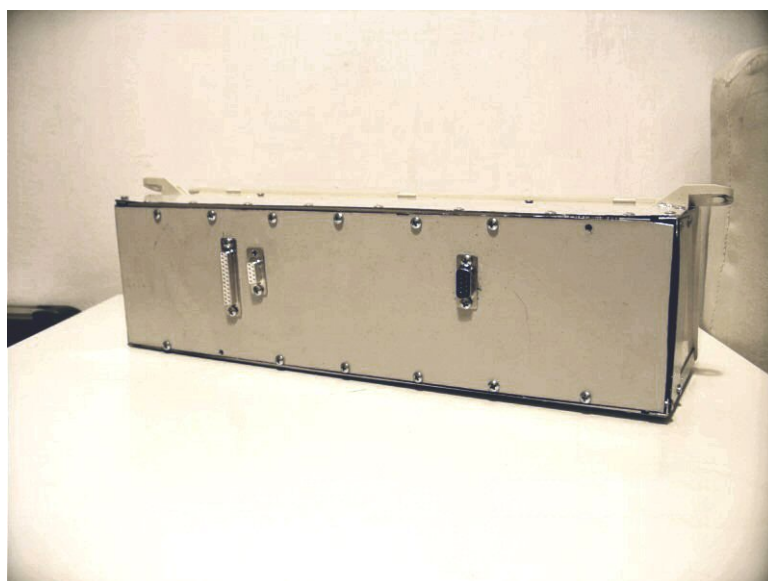


Rysunek 65. Stelaż z kątownika



Rysunek 66. Obudowa otwarta

W tylnej ścianie obudowy znajdują się komputerowe gniazda DB9 oraz DB25, służą one do podłączenia zasilania, magistrali 1 Wire®, przycisków, kontrolki oraz czujników.



Rysunek 67. Obudowa zamknięta

Kontrolki i wyświetlacze zamocowane są bezpośrednio do panelu frontowego. W celu oznaczenia poszczególnych diod posłużyły piktogramy wzorowane na oryginalnych, wykonane na papierze samoprzylepnym.



Rysunek 68. Panel frontowy

Instalacja w samochodzie

Do podłączenia urządzenia w samochodzie posłużyły odpowiednio wykonane przewody. Z jednej strony zakończone zgodnie z wyprowadzeniami instalacji elektrycznej samochodu, drugi koniec odpowiadał gniazdom DB 9 oraz DB 25 układu. Dzięki temu podstawowy montaż nie wymagał ingerencji w instalację samochodu i możliwy jest powrót do standardowej tablicy rozdzielczej.

Konieczne było poprowadzenie dodatkowych przewodów między innymi do:

- czujnika położenia wału(hallotron w aparacie zapłonowym);
- kontrolki zasilania LPG;
- kontrolki zasilania PB;
- kontrolki rezerwy LPG;
- kontrolki przednich świateł przeciwmgielnych;
- kontrolki prawego i lewego kierunkowskazu;
- kontrolki świateł mijania;
- kontrolki świateł stop;
- przekaźnika sterującego centralnym zamkiem;
- przekaźnika wycieraczek;
- włącznika wycieraczek.

Podłączenie sterownika wycieraczek wymagało doprowadzenia dodatkowych przewodów oraz zasilania do silnika. Jednak nie wymagało to ingerencji w instalację samochodu. Do podłączenia wykorzystane zostało złącze, gdzie podłączony był oryginalny sterownik wycieraczek.

Czujnik prędkości jazdy zamontowany został na standardowej lince prędkościomierza, w miejscu oryginalnych zegarów.

Instalacja w samochodzie obejmowała również montaż czujników temperatury zewnętrznej oraz wewnętrznej. Termometry zamontowane zostały w lusterku zewnętrznym oraz pod tapicerką wewnątrz pojazdu. Taki sposób montażu miał na celu wyeliminowanie wpływu czynników zewnętrznych na wskazania czujników.

Podsumowanie

Komputery stają się elementami życia codziennego. Nie dziwi fakt wyposażania prostych urządzeń w elektroniczne układy sterujące, informujące użytkownika o ich stanie bądź ułatwiające ich wykorzystanie.

Coraz powszechniejsze jest wyposażanie samochodów w tak zwane komputery pokładowe. Wynika to z faktu, że dane w ten sposób prezentowane są czytelniejsze dla użytkownika. Wiele informacji może być wyświetlane na jednym wyświetlaczu. Dzięki temu kierowca nie jest zasypywany niepotrzebnymi informacjami. Komputer sam również może informować kierowcę o problemach, lub nawet podejmować odpowiednie działania. Przykładem może być sytuacja, gdy temperatura silnika przekroczy dopuszczalną normę a komputer sterujący wyłączy silnik lub ograniczy jego moc, oraz poinformuje kierowcę o zaistniałym problemie. Rozwiązań tego typu może być wiele.

Wykorzystanie programowalnych układów elektronicznych niesie za sobą również inne korzyści. Można ograniczyć koszty produkcji, ze względu na produkowanie tylko jednego typu układu, natomiast w końcowej fazie produkcji zapisywać w poszczególnych układach różne programy. Takie rozwiązanie stosowane jest na przykład przy produkcji osprzętu sterującego pracą silników. Zmieniając oprogramowanie procesora sterującego wtryskiwaczami paliwa oraz mapą zapłonu można zmieniać dość dowolnie parametry i osiągi silnika.

Dzięki zastosowaniu układów programowalnych możliwe stało się dostosowanie produktu do specyficznych wymagań bez konieczności kosztownych zmian konstrukcyjnych.

Niniejsza praca zawiera przykład urządzenia zaprojektowanego w oparciu o układ FPGA. Projekt zawiera wszystkie funkcje standardowego zestawu wskaźników samochodu oraz dodatkowo umożliwia zaprogramowanie przerywanej pracy wycieraczek, termometr wewnętrzny oraz zewnętrzny.

Projekt dowodzi, że układy FPGA doskonale nadają się do wykonania prototypów lub produktów krótkoseryjnych. Wykorzystana technologia wykonania płytek drukowanych również doskonale sprawdza się w tego typu zastosowaniach. Należy zaznaczyć, że kluczową rolę w pomyślnym ukończeniu pracy odgrywają właśnie płytki drukowane, sposób połączeń poszczególnych elementów systemu. Dzięki

temu możliwe jest rozwijanie projektu, optymalizacja, lub nawet dodanie dodatkowych funkcji bez konieczności przebudowy całego układu.

Literatura

1. Mark Zwoliński; „Projektowanie układów cyfrowych z wykorzystaniem języka VHDL”; WKŁ, Warszawa 2002
2. XST User Guide ver. 8.1i; Xilinx
3. Paul Horowitz, Winfield Hill; „Sztuka Elektroniki”; WKŁ, Warszawa 1995
4. Elektronika Praktyczna, „Układy FPGA w przykładach”, październik, listopad 2006, styczeń 2007
5. Piotr Słodowy, „Elektronika w moim samochodzie”; WKŁ, Warszawa 1982
6. Dieter Nührmann, „Elektronika łatwiejsza niż przypuszczasz, technika cyfrowa”, WKŁ, Warszawa 1986
7. App Note 119 Embedding the 1-Wire Master; Dallas-Maxim
8. App Note 120 1-Wire Communication Through Software; Dallas-Maxim
9. App Note 162 Interfacing The DS18X20; Dallas-Maxim
10. Nota aplikacyjna układu DS1WM; Dallas-Maxim
11. Nota aplikacyjna układu DS18B20; Dallas-Maxim
12. Nota aplikacyjna układu DS2450; Dallas-Maxim
13. Nota aplikacyjna układu HD44780U; Hitachi
14. Nota aplikacyjna układu LM7805; STM
15. Nota aplikacyjna układu LM1085; HTC
16. Nota aplikacyjna układu M28W640FCT; STM
17. Nota aplikacyjna układu PC817; Sharp
18. Nota aplikacyjna układu STP16C596; STM
19. Nota aplikacyjna układu TLE4905; Siemens
20. Schemat mini modułu MMfpga12; Propox