**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського»**

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ

КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ АПАРАТУРИ

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_ Лисенко О.М.

(підпис) (ініціали, прізвище)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_р.

**Дипломний проект**

**на здобуття ступеня бакалавра**

зі спеціальності 172 Телекомунікації та радіотехніка

(код та назва напряму підготовки або спеціальності)

на тему Блок керування двигуном по положенню

Виконав: студент ІV курсу, групи ДК-41

Білаш Богдан Олегович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім’я, по батькові) (підпис)

Керівник ст. викл. Антонюк О.І. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Консультант \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(назва розділу) (вчені ступінь та звання, прізвище, ініціали) (підпис)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Київ - 2018 року

**Національний технічний університет України**

**«Київський політехнічний інститут ім. Ігоря Сікорського »**

Факультет електроніки

Кафедра конструювання електронно-обчислювальної апаратури

Освітньо-кваліфікаційний рівень – бакалавр

Спеціальність 172 Телекомунікації та радіотехніка

(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_ Лисенко О.М.

(підпис) (прізвище ініціали)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_р.

**ЗАВДАННЯ**

**на дипломний проект студенту**

Білаша Богдана Олеговича

(прізвище, ім’я, по батькові)

1. Тема проекту Блок керування двигуном по положенню

керівник проекту Антонюк Олександр Ігорович, старший викладач

затверджені наказом по університету від 23.03.2018 року №1008-с

1. Термін подання студентом проекту 12 червня 2018 року
2. Вихідні дані до проекту, пристрій являє собою моноблочну конструкцію, кліматичне виконання УХЛ 4.2 по ГОСТ 15150-69. Пристрій повинен забезпечувати з’єднання з вимірювальним пристроєм, посилання на нього керуючих сигналів, прийняття інформаційних сигналів, їх обробку, та вивід на індикатор.
3. Зміст розрахунково-пояснювальної записки:

* аналіз технічного завдання;
* огляд існуючих засобів керування, патентний пошук;
* розробка схеми електричної принципової;
* обґрунтування вибору елементної бази та друкованої плати;
* проектування у пакеті програм QuartusII;
* розміщення компонентів на друкованій платі блока керування;
* конструкторсько-технологічні розрахунки;
* електричний розрахунок друкованої платиблока керування;
* розрахунок надійності друкованої плати блока керування;
* розрахунок віброміцності друкованої плати блока керування;
* проектування у Altium Designer;
* моделювання проекту;
* підтвердження працездатності проекту;
* висновки.

1. Перелік графічного матеріалу (із зазначенням обов’язкових креслень, плакатів, презентацій тощо):

* електрична принципова схема блока керування;
* друкована плата блока керування;
* складальне креслення друкованої плати блока керування;

1. Консультанти розділів проекту

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
| завдання  видав | завдання  прийняв |
| Розділ |  |  |  |
| Розділ |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання 16.03.18

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів виконання  Дипломного проекту | Термін виконання етапів проекту | Примітка |
| 1 | Аналіз технічного завдання |  | виконано |
| 2 | Схемо-технічне проектування |  | виконано |
| 3 | Вибір елементної бази та друкованої плати |  | виконано |
| 4 | Виконання креслень схем електричних принципових |  | виконано |
| 5 | Конструкторсько-технологічні розрахунки |  | виконано |
| 6 | Електричний розрахунок друкованих плат |  | виконано |
| 7 | Розрахунок надійності, віброміцності |  | виконано |
| 8 | Проектування у Altium Designer |  | виконано |
| 9 | Виконання креслень друкованої плати та складального креслення друкованого вузла |  | виконано |
| 10 | Моделювання |  | виконано |
| 11 | Оформлення пояснювальної записки |  | виконано |

Студент Білаш Б.О.

(підпис) (прізвище та ініціали)

Керівник проекту Антонюк О.І.

(підпис) (прізвище та ініціали)

**Пояснювальна записка**  
**до дипломного проекту**

на тему: **Блок керування двигуном по положенню**

Київ – 2018 року

ЗМІСТ

[Перелік скорочень, умовних позначень, термінів 2](#_Toc514756100)

[ВСТУП 3](#_Toc514756101)

[Розділ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ 4](#_Toc514756102)

[1.1. Аналіз механіки електроприводу. 4](#_Toc514756103)

[1.1.1. Динамічні моделі механічної частини електроприводу. 4](#_Toc514756104)

[1.1.2. 5](#_Toc514756105)

[Розділ 2. СХЕМОТЕХНІЧНЕ ПРОЕКТУВАННЯ 6](#_Toc514756106)

[2.1. Розробка структурної схеми системи 6](#_Toc514756107)

[Розділ 3. ПРОЕКТУВАННЯ ДРУКОВАНОГО ВУЗЛУ 11](#_Toc514756108)

[3.1. Вибір типу, матеріалу друкованих плат 11](#_Toc514756109)

[Розділ 4. РОЗРАХУНКИ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ КОНСТРУКТОРСЬКИХ РІШЕНЬ 12](#_Toc514756110)

[4.1. Конструкторсько-технологічний розрахунок елементів ДП 12](#_Toc514756111)

[4.1.1. Визначення мінімальної ширини друкованого провідника по постійному струму для ланцюгів живлення і 12](#_Toc514756112)

[Розділ 5. СТВОРЕННЯ ВІРТУАЛЬНОГО ПРИСТРОЮ У СЕРЕДОВИЩІ LABVIEW 13](#_Toc514756113)

[Розділ 6. ПРОЕКТУВАННЯ КОРПУСА 14](#_Toc514756114)

[ВИСНОВКИ 15](#_Toc514756115)

[СПИСОК ЛІТЕРАТУРИ 16](#_Toc514756116)

[Додаток А 17](#_Toc514756117)

[Технічне завдання на проектування 17](#_Toc514756118)

[Додаток Б 21](#_Toc514756119)

Перелік скорочень, умовних позначень, термінів

МК – мікроконтролер

ДП – друкована плата

ДМ – друкований монтаж

ДВ – друкований вузол

ОП – операційний підсилювач

КЕ – конструктивні елементи

ПК – персональний комп’ютер

ПМ – посадкове місце

FPGA, ПЛІС – програмована логічна інтегральна схема

ШІМ, PWM – широтно-імпульсна модуляція

АЦП, ADC – аналогово-цифровий перетворювач

ФВЧ – фільтр високих частот

ФНЧ – фільтр низьких частот

ПВЗ – пристрій вибірки зберігання

ПК – персональний комп’ютер

ЖКІ – жидкокристалічний індикатор

EPCS - Electronic prescriptions for controlled substances

PLL – Phaze-locked loop

САПР – система автоматизованого проектування.

УГО – умовно-графічне зображення

ПЗ – програмне забезпечення

ВСТУП

**Розділ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ПАТЕНТНИЙ ПОШУК**

* 1. Аналіз механіки електроприводу.

Електроприводи використовуються для відтворення заданого закону руху виконавчих органів механізму(перша група), або для виконання цими органами корисної механічної роботи («силовий» привід, друга група).

До першої групи відносяться більшість позиційних приводів, приводи випробувальних і перевірочних стендів і іншого обладнання [1].

До другої групи відносяться головні приводи металорізальних станків, прокатних станків, насосів, вентиляторів і усіх інших механізмів і машин, призначених для перетворення механічної енергії в корисну роботу.

Але дуальність двох груп властива всім системам електроприводу. Це змушує одні з них розглядати як системи передачі і перетворення інформації, а інші – як системи електромеханічного перетворення енергії. Класифікацію електроприводів по цим властивостям вперше запропонував В.Г. Каган[2]. Таким чином, в першому випадку при конструюванні електроприводу основна увага приділяється на придання йому властивостей, які забезпечують якісне відтворення переданої інформації. При цьому в сталому режимі роботи встановлена потужність електродвигуна, як правило, не до кінця використовується. В другому випадку домінуюче значення має вибір необхідної потужності електродвигуна і інших елементів силового приводу.

Проте не слід розглядати електроприводи першої групи тільки як системи для відтворення інформації, базуючи всі розрахунки і досліди на положеннях теорії керування. Вочевидь, найдосконаліші керуючі пристрої не зможуть забезпечити заданий рух робочих органів виконавчого механізму, якщо він не забезпечений енергетично силовою частиною приводу.

Так само і головний привід металоріжучого станка при достатньому запасі потужності не зможе виконати своїх енергетичних функцій, якщо система керування не забезпечить заданих законів його руху. Таким чином, в приводі здійснюються складні взаємозв’язки між його інформаційною і енергетичною частинами. Тому, приступаючи до проектування електроприводу, інженер кожен раз неоднозначно вирішує, яку долю уваги слід приділити розробці керування і яку енергетичній частині. Ті ж самі задачі неминуче вирішуються при його виготовленні, наладці та експлуатації.

Для основних різновидів електроприводів були запропоновані і різні системи їх класифікації: для силових приводів – В.К. Поповим, а для приводів відтворення руху – С.А. Ковчіним.

Різновид систем автоматичного керування (САК) містить взаємопов’язані інформаційні і енергетичні канали. Причому інформаційно-замкнута САК формує задані закони руху виконавчих органів. В той же час така САК може реалізувати і задані закони керування перетворенням енергії, якщо будуть належним чином змінюватися її компоненти.

* 1. Вивчення аналогів.

Під час пошуків реалізації блоку керування в першу чергу необхідно було визначити, чи є певний пристрій для керування, який би задовольнив вимоги по керуванню двигуном по положенню.

Першим розглянутим патентом, який пропонує пристрій для керування серводвигуном є «Устройство для управления серводвигателем» 1967 року [3]. Номер патенту в базі СРСР – 191675. Даний патент пропонує керування через перемикаючий діод, який керується імпульсним трансформатором. Даний патент зовсім немає практичного значення в 2018 році, і був вибраний для ознайомлення з методами керування.

Другим розглянутим патентом є «Позиционный серводвигатель». Номер патенту в СРСР 1222907, 1986 року [4]. Порівнюючи його з попереднім патентом, серводвигун має в своєму складі цифрові логічні елементи АБО. Для керування двигуном використовуються дискретні керуючі сигнали Р1-Р3. Повний опис роботи пристрою можна прочитати в описі патенту. Але для даного часу керування застаріло, сучасні можливості електроніки дозволяють більш ширше керувати серводвигуном за допомогою комп’ютерів, мікроконтролерів.

Третім розглянутим варіантом стала «Система автоматического управления сервоприводами» авторів Мірзаєв Р.А., Смірнов Н.А [5]. На жаль – цей виріб не є запатентованим. Даний пристрій запропоновано у 2014 році, що свідчить про його актуальність. Пристрій пропонує керувати серводвигунами за допомогою ШІМ-сигналів, які генеруються контролером через керуючі сигнали, які поступають з ПК до контролера.

Дослідивши третій варіант, було розглянуто його основну ідею, яка може бути застосована при проектуванні власного блока керування.

Перевагами є:

* Сучасність та актуальність методу керування;
* Застосування сучасної елементної бази: а саме ПК, мікроконтролерів;
* Завдяки сучасний компонентам підвищується надійність пристрою;
* Більш спрощений та автоматизований спосіб керування, порівнюючи з попередніми запропонованими методами, який має більш широкий спектр для керування двигуном.

Але при всьому цьому даний пристрій має наступні недоліки, які пропонується усунути пристроєм, який розробляється в дипломному проекті:

* Розглянутий пристрій періодично потребує підключення до ПК. Пристрій, що проектується, цю проблему усуває за рахунок того, що програма записується в пам’ять блоку і зберігається там;
* Розглянутий пристрій використовує зовнішній процесор. Пристрій, що проектується, використовує синтезований процесор, який можна в будь-який момент переконфігурувати.
* Пристрій, що проектується, оброблює не лише значення куту, а ще вимірює значення струму, має ширший діапазон керування.
* Розглянутий пристрій виступає як запропонований варіант, але не є запатентованим.

Дослідивши різні варіанти запатентованих та запропонованих пристроїв, було вирішено створити власний пристрій з нуля, який має свою структуру, будову та реалізацію.

**1.3 Вибір та обґрунтування елементної бази.**

На пристрій поступає напруга живлення номіналом 5 В та 12 В, яка надходить з іншого, вже розробленого універсального блока живлення. Внутрішні компоненти блока керування споживають напругу номіналами 1,2 В, 2,5 В, 3,3 В. Тому необхідно перетворити вхідну напругу. Для цього використовуються перетворювачі напруги. Для перетворення напруги з 5 В в 2,5 В обирається мікросхема TPS79325, у якої вихідне значення напруги фіксоване і становить 2,5 В [6]. Вихідний струм мікросхеми становить 200 мА. Цей струм задовольняє характеристики ПЛІС.

У випадку з 1,2 В та 3,3 В, сімейство мікросхем TPS79325 не може видавати напругу 1,2 В ні у фіксованому ні у режимі підстроювання; також вихідного струму у 200 мА буде недостатньо у випадку з напругою 3,3 В. Тому прийнято рішення використовувати інший перетворювач напруги TPS62000[7]. Він може видавати вихідну напругу в діапазоні від 0,9 В до напруги живлення та струм 600 мА. Було вирішено використати дві мікросхеми для подачі на подальшу схему напругу номіналами 1,2 В та 3,3 В.

Основним компонентом блока є FPGA фірми Altera з вбудованим процесорним ядром Nios II. Для повноцінного вирішення задачі керування двигуном достатньо не дуже потужної мікросхеми, тому було вирішено обрати мікросхему EP3C25E144I7 [8]. Це пристрій сімейства Cyclone III, яка має вбудоване процесорне ядро. На даний момент сімейство Cyclone II вже вийшло з виробництва, а більш потужніші сімейства ставити немає сенсу, адже прогнозується використання дуже малої частини логічних елементів ПЛІС. З цих розмірковувань та доступних варіантів було обрано саме сімейство Cyclone III. Сама мікросхема виготовлена у QFP корпусі, має 144 контакти, з яких універсальних контактів вводу/виводу, доступних користувачу, 83. Цієї кількості буде більше, ніж достатньо для забезпечення роботи блока.

Для конфігурації та застосування флеш-пам’яті було вирішено застосовувати EPCS-мікросхему EPCS16SI16N [9]. У ПЛІС є спеціальні контакти для роботи з даною мікросхемою, то їх потрібно просто з’єднати між собою. Під час пошуків було знайдено лише корпус з 16 контактами, з яких 8 ні до чого не приєднані. В умовах жорстких обмежень по площі, дана мікросхема не підійшла. Але для даного розроблюваного корпусу та габаритів плати можна встановити даний тип корпусу.

На блок керування поступає значення кута від датчика TS2640N691E125 [10]. Даний датчик є синусно-косинусним трансформатором. Необхідно обрати такий драйвер, який зможе генерувати гармонічні сигнали, які поступають до трансформатору, що обертається (резольверу), приймати значення синуса та косинуса цього сигналу, перетворювати їх в цифровий код та передавати дані до керуючого пристрою. Для цих цілей було обрано спеціально розроблений та виготовляємий фірмою Analog Devices перетворювач з опорним генератором AD2S1210 [11]. Дана мікросхема має вбудований програмний маятниковий генератор, який генерує синусоїдальну хвилю спеціально для резольверів. Генератором можна керувати програмно з центрального процесору, що дозволяє змінювати частоту генеруючого гармонійного сигналу. Вихідний сигнал представляється у вигляді 16-розрядного цифрового коду, який виходить з 16 контактів мікросхеми. Інформаційний сигнал, а також керуючі сигнали під’єднуються до універсальних контактів вводу / виводу FPGA.

Для перетворення аналогового значення струму у цифровий код необхідно використовувати АЦП. АЦП повинно мати малошумну широку полосу пропускання, коротку затримку. Для передачі цифрового коду буде достатньо послідовний інтерфейс передачі даних. Виходячи з даних умов в якості АЦП було обрано AD7687BRMZ [12]. Даний АЦП генерує саме послідовний цифровий сигнал на своєму виході. Час перетворення аналогового сигналу в цифровий згідно технічної документації складає до 2,2 мкс, а мінімальний період передачі одного розряду цифрового коду складає 15 нс. Згідно Рисунку 1.1, для передачі цифрового сигналу на вихідний контакт необхідно 17 тактів. Також необхідно 3 нс для встановлення роботи запису з падаючого фронту і час переходу вихідного контакту в низький режим. Для передачі даних з трьох послідовно включених АЦП максимально необхідний час становить 15 нс. В даному пристрої буде застосовуватися один АЦП. Сумарний час для передачі даних з АЦП складає 17\*15+3+15=273 нс = 0,273 мкс. Отже, час передачі даних набагато менший за час перетворення аналогового сигналу в цифровий. Тому для цього достатньо використовувати послідовну передачу сигналу.

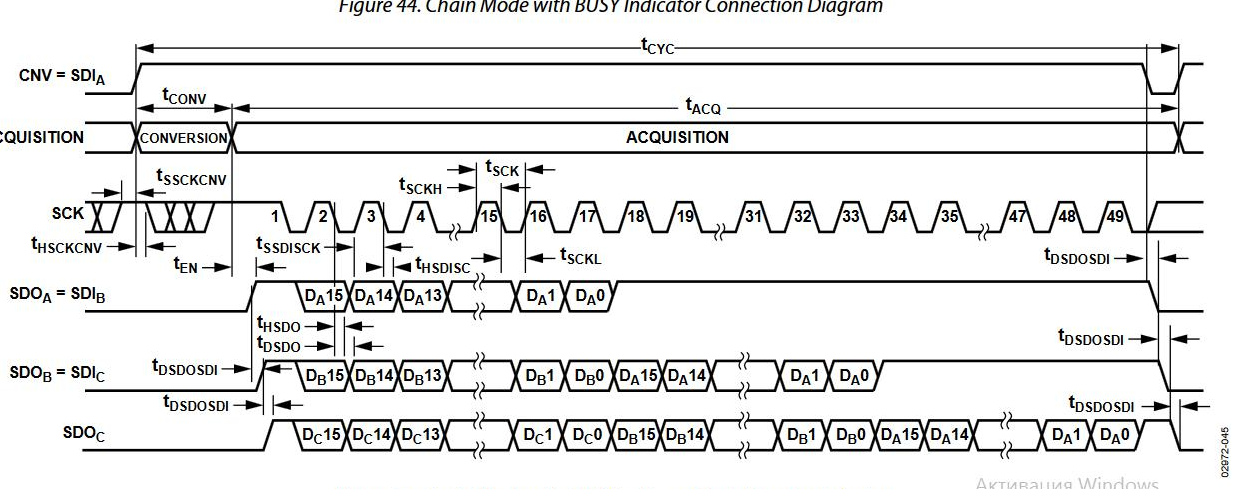


Рисунок 1.1 Часова діаграма АЦП.

Для забезпечення тактування в пристрої необхідно додати тактовий генератор. На вхід ПЛІС необхідно подавати базову тактову частоту, яка всередині завдяки PLL буде перетворюватися на іншу частоту. Проаналізувавши можливості перетворення PLL одних частот в інші було обрано за базову частоту 10 МГц. Тому для генерації цієї частоти було обрано компонент ECS-3963-100 [13]. Він працює на напрузі живлення 3,3 В, і генерує частоту 10 МГц. Сама мікросхема складається з 4 контактів: живлення, вихід, земля, дозвіл роботи. Генератор повинен генерувати тактовий сигнал весь час, незалежно від програми, тому контакт дозволу роботи необхідно під’єднати до живлення.

Для вихідного ШІМ сигналу важливо зберігати його форму при надходженні до двигуна. У зв’язку з тим, що пристрій, для якого розроблюється блок керування, працює в жорстких електромагнітних умовах, необхідно на цей пристрій подавати диференційні сигнали (так як саме вони використовуються в системі, для якої розробляється даний блок керування). Тому використовуються диференційні пари. Щоб з керуючого сигналу створити диференційну пару, вирішено використовувати мікросхему FIN1001 [14]. Мікросхема призначена саме для цієї задачі. Має 5 контактів: живлення 3,3 В, земля, вхідний сигнал, два вихідних диференційних сигнали.

Згідно ГОСТ 12.2.007-75, якщо напруга живлення більше 42 В, то роз’єм, який є джерелом, повинен бути типу «гніздо», а роз’єм, який є приймачем живлення повинен бути типу «вилка» . В даному пристрої напруга живлення менше 42 В, тому немає різниці якого типу встановлювати роз’єми.

Всі типи роз’ємів, розташування сигналів та живлення на контактах узгоджується з приймаючим або передаючим пристроєм заздалегідь.

**Висновок до розділу:**

Пристрій, що проектується повинен керувати двигуном по положенню. Двигун має працювати за заданим законом руху. Для керування двигуном необхідно знати значення його кута. Для цього було досліджено різну кількість існуючих патентних рішень керування серводвигуном. Зроблено висновок, що дана область керування двигуном по положенню не є досить вивченою, адже більшість знайдених патентних рішень є застарілими. У сучасних запропонованих варіантів керування двигуном є недолік в тому, що вони вимагають постійного контакту з ПК. Двигун, для якого розробляється блок керування, досліджується у лабораторних умовах, в яких знаходження ПК не є можливим. Виходячи з цього прийнято рішення розробити особливий блок керування двигуном в лабораторних умовах. Для керування на резольвері двигуна встановлюється датчик, який передає значення синуса та косинуса обертання кута обертання. Щоб отримувати інформацію з резольвера була обрана спеціальна елементна база. При її виборі враховувалися: спеціальне призначення певних мікросхем для зчитування даних в двигуна, а також масовість та популярність, ціна, постачання та можливість придбання на території України інших стандартного вжитку мікросхем. Вибір елементної бази виявився вдалим, в результаті чого стає можливим подальше проектування пристрою.

**Розділ 2. СХЕМОТЕХНІЧНЕ ПРОЕКТУВАННЯ**

* 1. Розробка структурної схеми та принцип роботи модулю.

Структура блока керування і його взаємозв’язок з установкою моделювання режимів роботи привода зображена на Рисунку 2.1.

|  |
| --- |
|  |
| Рисунок 2.1 Структура блока керування |

Блок керування складається з:

* FPGA, до складу якої входить центральний процесор Nios II, з синтезованим ШІМ-контролером та АЦП-контролером;
* Драйвер датчика кута;
* АЦП;
* Диференційні драйвери;
* Індикатор;
* Клавіатура.

Блок призначений для формування сигналів для випробуваного двигуна, вимірювання кутового положення ротора і струму двигуна.

В схемі вимірювання кута повороту ротора застосовується датчик TS2640N691E125, розташований безпосередньо на осі випробуваного двигуна і мікросхема перетворювача кут / код AD2S1210 в модулі керування. Виміряне значення кута повороту осі ротора двигуна надходить по паралельній 16-ти розрядній шині на цифровий контролер. Це значення відображається на індикаторі і використовується в алгоритмі формування керуючих сигналів на серводвигун (при виборі відповідного режиму роботи). Конструктивно, в двигуні застосовується датчик струму, вихідний сигнал якого надходить на схему вимірювання, що представляє собою 16-ти розрядний аналого-цифровий перетворювач (АЦП). Для управління серводвигуном цифровий контролер формує і видає такі сигнали управління: PWM, DIR, DIS. PWM - це широтноімпульсний модульований сигнал, керуючий швидкістю обертання двигуна. DIR - сигнал вибору напрямку обертання. DIS - сигнал дозволу роботи двигуна. Функціонально, модуль управління двигуном імітує роботу бортового контролера платформи, замикаючи зворотний зв'язок по положенню від датчика кута повороту ротора відповідно до заданого режиму роботи. Інформація про струм і кутове положення двигуна виводиться на індикатор модуля. Додатково, забезпечується режим обмеження струму і кута повороту.

Схема індикації забезпечує візуалізацію заданого режиму роботи і значення наступних параметрів:  
− максимально допустимого кута повороту ротора;  
− максимально допустимого значення величини струму двигуна;  
− номінального значення струму двигуна;  
− поточного значення кута повороту ротора;  
− поточного значення струму двигуна.

2.2 Розробка та розрахунок схеми електричної принципової.

У дипломному проекті розглядається розробка друкованого вузла, який називається «Контролер модуля управління».

Схема контролера модуля управління живиться від 5В постійної напруги живлення. Напруга живлення номіналами 5 В та 12 В поступає з роз’єму XP12.

Для забезпечення напругою номіналами 1,2 В, 2,5 В, 3,3 В, розроблена схема перетворення вхідної напруги. Для формування напруги 1,2 В, 3,3 В використовується мікросхема TPS62000. ЇЇ підключення обрано згідно рекомендаціям технічної документації на дану мікросхему. На вході живлення мікросхеми додається керамічний конденсатор номіналом не менше ніж 10мкФ. Паралельно йому додано ще один конденсатор номіналом 0,1 мкФ. Таке рішення прийнято з загального досвіду, де якомога ближче до контакту мікросхеми ставиться керамічний конденсатор вказаного вище номіналу. Мікросхеми працюють в режимі підстроювання. З виходу контакту FB подається фіксована напруга 0,45 В. Вихідна напруга формується ні дільнику, який складається з двох резисторів R60, R66 для DA5, та R61, R67 для DA7. Розрахунок виконується за формулою (2.1).

(2.1)

Згідно технічної документації сума опорів для резисторів дільника не повинна перевищувати 1 МОм. Для задання вихідної напруги 3,3 В було розраховано резистори R60 = 15 кОм, та R66 = 95,3 кОм. В результаті отримали вихідну напругу:

Для забезпечення надійності згідно технічної документації потрібно додати шунтуючий конденсатор номіналом 220 пФ.

Для задання вихідної напруги 1,2 В було розраховано резистори R61 = 100 кОм, та R67 = 165 кОм. В результаті отримали вихідну напругу:

Для забезпечення надійності згідно технічної документації також додаємо шунтуючий конденсатор.

Для задання вихідної напруги 2,5 В використовується мікросхема TPS79325, яка видає фіксовану напругу номіналом 2,5 В. Схема її підключення та номінали конденсаторів обрано згідно технічної документації.

Для забезпечення послідовного подання живлення на ПЛІС сигнал готовності джерела живлення 1,2 В поступає на вхід дозволу роботи джерел живлення 2,5 В та 3,3 В. Ці рекомендації запропоновані виробником ПЛІС у технічній документації.

Для перевірки роботи напруги на початковому етапі налагодження плати додаються нульові резистори R69-R71, які спочатку не впаюються, а після перевірки правильного перетворення напруг впаюються, тим самим з’єднуючи перетворювачі напруги з усією іншою схемою.

Результуюча схема для перетворення всіх трьох напруг зображена на Рисунку 2.2.

Основною складовою схеми є FPGA EP3C25E144I7 фірми Altera. Вона має 144 контакти. Для даного пристрою це більш ніж достатньо. Різні банки ПЛІС та її ядро живляться напругами 2,5 В, 3,3 В, 1,2 В. Усі сигнали до ПЛІС, окрім тактуючих, подаються на стандартні контакти вводу/виводу мікросхеми. На усі контакти живлення ставиться якомога ближче до корпусу конденсатор 0,1 мкФ, який підтягнутий до землі.

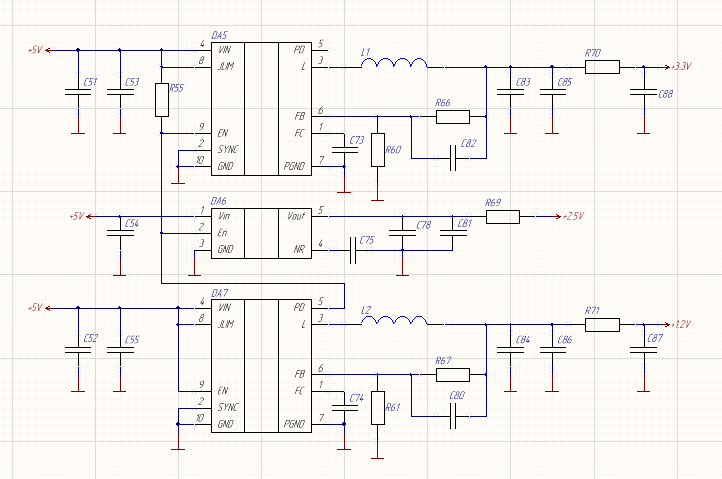


Рисунок 2.2. Схема перетворення вхідної напруги.

Це робиться для попередження імпульсів та шумів в ланцюгах живлення. Коли конденсатор ставиться біля контакту живлення будь-якої мікросхеми, то шлях провідника між контактом мікросхеми та конденсатором має свою паразитну індуктивність та опір, може створитись фільтр, який при високих частотах негативно впливає на роботу мікросхеми. В деяких випадках запобіжні конденсатори необхідно ставити на сам контакт мікросхеми. Це рекомендовано самими виробниками. У випадку з FPGA, Altera має рекомендаційні поради для правильного розташування компонентів на платі. Але для даного пристрою такі вимоги не критичні, тому конденсатори розташовано лише близько до контактів FPGA.

Надалі біля всіх контактів живлення та землі, керуючих контактів також будуть розташовуватися запобіжні конденсатори.

До ПЛІС безпосередньо під’єднується EPCS-мікросхема EPCS16SI16N. Між контактами для передачі даних ставиться невеликий фільтруючий резистор для зменшення крутизни фронту вихідних сигналів, тим самим зменшуючи паразитні шумові перешкоди. Варіант розташування запобіжних конденсаторів береться згідно рекомендацій технічної документації, та розташовується біля контактів живлення.

З ПЛІС з’єднується дра йвер датчика кута AD2S1210. Ця мікросхема приймає диференційні аналогові сигнали *SIN* та *COS*, які поступають з датчику кута. Між цими диференційними сигналами ставиться резистор узгодженого навантаження для поглинання енергії. По провідникам протікає струм, який у мікросхему втікає мізерно малий. Для того, щоб струм не відбивався зворотно у провідник, то ставиться резистор, через який струм стікає до іншого провідника, тобто резистор знищує струм. Резистор ставиться номіналом 10 кОм не для хвильового опору, а як навантаження на резольвер відповідно з рекомендаціями по використанню резольвера.

На тактовий контакт ставиться резистор малого номіналу для відфільтровування перешкод у сигналі.

На виході мікросхема передає 16-бітний двійковий код, який передає до ПЛІС значення кута, а також керуючі сигнали. Також в залежності від команд процесора, драйвер видає гармончні сигнали  та . Ці задають опорну гармоніку для поворотного трансформатора (резольвера). Ці сигнали поступають на диференційний підсилювач THS4130IDGK [15], який також виступає як ФНЧ. Підсилювач живиться від 12 В, і подає сигнал такою амплітудою на резольвер. Так як датчик TS2640N691E125 має коефіцієнт трансформації 0,5±10%, то для того, щоб на вхід мікросхеми приходили сигнали *SIN* та *COS* належної амплітуди, плюс врахувавши втрати, було прийнято рішення використовувати саме 12 В як напругу живлення підсилювача. Мікросхема THS4130IDGK та під’єднані до неї резистори працюють як інвертуючий диференційний підсилювач, так як прямий вихід мікросхеми через зворотній зв’язок з’єднується з інверсним входом. Вихідна напруга інвертуючого диференційного підсилювача знаходиться за формулою (2.2):

(2.2)

У випадку з THS4130IDGK резистром *R2* виступає резистор *R44,* а резистором *R1* виступає резистор *R16.* Отже, вихідною напругою буде:

Так як даний підсилювач виступає як фільтр, то фільтруючу роль в даному каскаді виконує ланцюг *R26, C21*. Частота зрізу знаходиться за формулою (2.3):

(2.3)

Для даного елементу частота зрізу становитиме:

Згідно технічної документації на AD2S1210, дана мікросхема генерує гармонічні сигнали з частотою до 20 кГц.

Дане розташування пасивних елементів справедливо і для протилежного даному каскаду підсилювача.

Розташування елементів в такому порядку рекомендовано в технічній документації THS4130IDGK при використанні мікросхеми в даному режимі роботи.

До ПЛІС під’єднується АЦП. Його схема представлена на Рисунку 2.3.



Рисунок 2.3. Схема підключення АЦП.

Схема складається безпосередньо з АЦП AD7687BRMZ. Даний компонент приймає аналоговий диференційний сигнал, та видає на ПЛІС послідовне 16-розрядне двійкове значення струму двигуна. АЦП певний проміжок часу зчитує значення струму, після чого передає на вихід його цифровий еквівалент. Зчитування значення струму відбувається по принципу ПВЗ: всередині АЦП є конденсатори, які заряджаються до певного значення, з якого потім отримують значення величини напруги, а відповідно і струму. Розмір конденсаторів дуже малий, тому необхідно подавати максимально точну величину струму на АЦП. Для цього застосовується вже згадана вище мікросхема диференційного підсилювача THS4130IDGK. Для цієї схеми на підсилювач подається напруга живлення 5 В, адже необхідно не підсилити, а повторити сигнал. Між підсилювачем та АЦП ставиться фільтр, який складається з резистору та конденсатору, який повинен бути на декілька порядків більше внутрішніх конденсаторів. Поки АЦП не заміряє значення струму, цей конденсатор заряджається. Під час замикання та замірювання АЦП даний конденсатор розряджається, швидко заряджаючи внутрішні конденсатори АЦП. Резистор потрібен для обмеження струму на шляху від підсилювача до АЦП. Зазвичай його значення береться до 1 кОм.

Тактовий генератор з’єднується через резистори малої величини (до 100 Ом) з тактовими входами ПЛІС. До входу живлення додаємо запобіжний конденсатор номіналом 0,1 мкФ.

Мікросхеми FIN1001 мають однаковий інтерфейс в своєму підключенні. Згідно технічної документації рекомендацій щодо підключення пасивних компонентів навколо мікросхеми немає. Тому вирішено поставити біля контакту живлення запобіжний конденсатор номіналом 0,1 мкФ. Між вихідними контактами ставиться резистор узгодженої напруги на 100 Ом.

Роз’єми M20-9770242 з’єднують центральний процесор зі світлодіодами та кнопками керування. У випадку з діодами транзистори [16] ставляться для того, щоб збільшити надійність пристрою, зменшити навантаження на джерело живлення ПЛІС, збільшити струм, яскравість світіння діода. Транзистор працює в ключовому режимі.

У випадку з кнопками використовується RC-ланцюг, який створює навмисні затримки у вузлі для зменшення впливу брязкоту контактів.

**Висновок до розділу:**

На основі поставленого технічного завдання була розроблена структурна схема пристрою. Схема містить в собі усі структурні зв’язки блока керування з пристроєм, на яким виконується контроль, а також з іншими платами та елементами, необхідними для вимірювання та обробки даних з двигуна. Була виконана перевірка, чи задовольняє дана структурна схема усі вимоги до здійснення керування та слідкування за двигуном Обравши елементну базу та опираючись на структурну схему була спроектована схема електрична принципова. При проектуванні схеми електричної принципової окремі каскади будувалися згідно рекомендації технічних документацій до мікросхем для найбільш коректної роботи. В каскадах, де було необхідно виконати розрахунок, були виконані розрахунки вихідної напруги перетворювачів напруги, коефіцієнтів підсилення диференційного підсилювача, частоту зрізу ФНЧ. Результат розрахунків задовольняє вимоги ТЗ. Виходячи з власного досвіду, до всіх контактів живлення додавалися запобіжні конденсатори на 0,1 мкФ. У інформаційних ланцюгах, які виходять з контактів ПЛІС додавалися резистори номіналом не більше 100 Ом для створення спеціальної затримки сигналів. Це необхідно через те, що ПЛІС дуже швидко генерує фронти імпульсів, які, поступають до інших мікросхем, можуть некоректно ними сприйматись. Тому дана затримка дещо тормозе наростання фронтів. На входах диференційних сигналів мікросхем додавались резистори, які поглинали хвильову енергію через стікання по черзі струму один на інший провідник, через що струм не відбивався з мікросхеми знову до сигнальних ланцюгів. Номінал резисторів вибирався згідно хвильовому опору провідників.

**Розділ 3. ПРОЕКТУВАННЯ ДРУКОВАНОГО ВУЗЛУ**

3.1. Вибір типу, матеріалу друкованих плат.

Для виготовлення пристрою було прийнято рішення про виготовлення 4-шарової друкованої плати. Таке рішення прийнято через певну кількість причин:

* Провідники живлення та сигнальні провідники мають знаходитись якомога далі одні від одних для зменшення електромагнітних завад, паразитних ємностей, індуктивностей. Для нейтралізації цих чинників було обрано рішення розмістити якомога більше сигнальних провідників на верхньому шарі ДП, а усі провідники живлення на третьому шарі. Другий шар відведено повністю для землі. Таким чином другий шар виступає «екраном» між різними типами провідників. Усі провідники, які не була змога розмістити на верхньому шарі, розміщувались на нижньому, але все одно окремо від провідників живлення.
* При використанні 4-шарової ДП відбувається оптимізація розміщення друкованих вузлів. Також такий підхід дозволяє створювати на шарі землі критично важливі окремі ділянки землі, які потім з’єднуються з загальному землею.
* Сучасні можливості в відношенні ціна/технологія дозволяють використовувати багатошарові ДП там, де раніше така можливість в порівнянні з двошаровою ДП могла коштувати набагато дорожче.

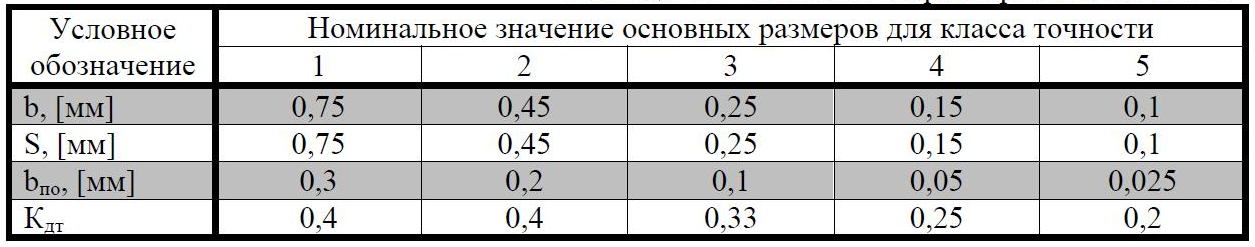
При нарощуванні кількості шарів рекомендується розділяти внутрішні шари землею для екранування. Для даного пристрою 4-шарової ДП буде повністю достатньо.

Виготовлення плати виконується на українському підприємстві «Гальванотехніка». Плата буде виготовлятися з матеріалу FR4-4-35-1,5. Даний матеріал є фольгованим склотекстолітом з підвищеною нагрівостійкістю, товщиною 1,5, мм, облицьований з двох зовнішніх та двох внутрішніх сторін мідною електролітичною фольгою товщиною 35 мкм.

**3.2. Вибір класу точності.**

Згідно ГОСТ 23571-86, існує 5 класів точності ДП (Таблиця 3.1). Клас точності повинен виходити з технології виробництва та застосованих компонентів. Підприємство «Гальванотехніка» має можливість виготовляти ДП до 5 класу точності включно.

*Таблиця 3.1. Номінальне значення основних розмірів для класу точності*



В Таблиці 3.1 параметр b, [мм] - ширина друкованого провідника; S, [мм] – відстань між краями сусідніх елементів; bпо, [мм] - гарантований поясок; Кдт - відношення номінального значення діаметра найменшого з металізованих отворів до товщини друкованої плати.

В умовах даного проекту 5 клас точності критично не потрібен. Даний пристрій має фіксовані габаритні розміри, які визначались виходячи з габаритних розмірів корпусу. Тому аналізуючи їх, надається можливість працювати на великій площині плати. Саме тому, мінімальна товщина провідників була обрана не 0,1 мм, що є мінімальним значенням для 5 класу точності, а 0,25 мм. Товщина плати складає 1,5 мм. Діаметр усіх перехідних отворів складає 0,75 мм. Виходячи з цього, відношення мінімального діаметра металізованого отвору до товщини плати складає 0,5. Товщина гарантованого пояску складає 0,225 мм, що майже в 10 раз більше за мінімальне значення 5 класу точності. Виходячи з усіх названих показників є можливість обрати 3 клас точності. Але відстань між краями сусідніх елементів не дозволяє використовувати цей клас. Мікросхеми, які були обрані, виготовляються в корпусі, для якого розрахована відстань між контактними майданчиками становить 0,2 мм. Тому необхідно використовувати 4 клас точності. Так як параметри, наведені в Таблиці 3.1 є мінімальними, не має необхідності змінювати товщину провідників та діаметр перехідних отворів.

В результаті обрано 4 клас точності. Даний клас передбачає проведення провідників між контактними площадками SMD резисторів. Але дана можливість не використовувалась: простору ДП достатньо для проведення провідників без їх транзитного розміщення під компонентами.

**3.3 Вибір методу виготовлення ДП.**

Підприємство «Гальванотехніка» виготовляє ДП комбінованим позитивним методом.

Комбіновані методи засновані на виготовленні ДП з фольгованих матеріалів. Провідники отримують хімічним методом, а металізацію отворів - хімічним або електрохімічним осадженням. Сутність комбінованих методів полягає в отриманні друкованих провідників шляхом травлення фольгованого діелектрика і металізації отворів електрохімічним способом. У позитивному методі травлення рисунку відбувається після металізації отворів, а для з'єднання металізованих отворів використовується ще не витравлена фольга, спочатку присутня на поверхні заготовки. Експонування рисунку схеми проводиться з фото позитиву. Після експонування проводиться свердління і металізація отворів. Потім рисунок схеми і металевий шар в отворах захищаються шаром гальванічного срібла або іншого металу, стійкого до рідини для травлення міді, після чого проводиться травлення незахищеної міді.

Переваги комбінованого позитивного методу:

* виключається можливість зриву контактних майданчиків під час свердління отворів;
* не потрібно застосування спеціальних контактуючих пристосувань при металізації отворів;
* знижується шкідливий вплив хімічних розчинів на ізоляційну основу і на міцність зчеплення фольги з основою плати.

**3.4. Розміщення компонентів та розводка ДП.**

Розміщення компонентів було розпочато з розміщення на спеціально відведені для цього місця деяких компонентів. Габаритні розміри плати та кріпильні отвори на початок розміщення та створення друкованої плати були вже відомі.

Компонентами, які були першими розміщені на платі, є ХР1 та ХР10. Виходячи з того, що вони є з’єднувальними елементами між пристроєм та зовнішніми блоками, у них заздалегідь було визначено розміщення та контактні майданчики. Також відносно визначеним місцем розміщення є місце для елементу ХР12. На даний роз’єм поступає 5 В та 12 В живлення від плати живлення, яка знаходиться в блоці керування біля плати керування. Тому їх зв’язок необхідно зробити максимально коротким. Біля роз’єму ХР1 розташовуємо три мікросхеми для диференційного перетворення сигналів. Так як в межах самої плати перешкоди не мають значного впливу, було вирішено розташувати диференційні перетворювачі не біля самих виходів з ПЛІС, а біля роз’єму. Це було зроблено через те, що біля самої ПЛІС немає багато вільного місця.

Біля роз’єму ХР10 розташовано мікросхему DA3, яка підсилює і фільтрує опорний гармонічний сигнал, що поступає на резольвер. Дана мікросхема живиться від 12 В, які поступають від заздалегідь визначеного і близько розташованого роз’єму ХР12. Далі в одній площині за ХР10 та DA3 розміщуємо драйвер датчика кута DA1. Розміщення необхідно було виконати в такому положенні, щоб сумарна довжина всіх провідників, які йдуть з драйвера до інших компонентів, була мінімальна.

Основна мікросхема FPGA розташувалась всередині ДП. При виводі провідників з неї була обрана стратегія розташовування компонентів, які з’єднуються цими провідниками з FPGA, на тій стороні, з якої провідники виходять. Також усі інші роз’єми розташувалися з усіх 4 сторін по краям для більшої зручності.

Мікросхеми для перетворення живлення розташувались близько до роз’єму живлення 5 В. Була обрана область на ДП, де немає сигнальних провідників. В тій області відбувається перетворення напруги 5 В на інші напруги, і звідти ті напруги поширюються вже на 3 шари ДП для живлення інших компонентів

Усі інші компоненти розташовувались на вільних місцях ДП.

Усі конденсатори було розташовано якомога ближче до контактів мікросхем. У випадку з ПЛІС використовувався алгоритм створення перехідного отвору, на який з 3 шару поступає напруга живлення. На верхньому шарі цей перехідний отвір з’єднується з контактним майданчиком виводу мікросхеми, а на нижньому шарі перехідній отвір з’єднувався з конденсатором, який надалі під’єднувався до землі. Таким чином виникає ізольована від загального живлення дільниця, на якій конденсатор запобігав проходженню шумів до контакту мікросхеми.

Технології підприємства «Гальванотехніка» не дозволяють створювати перехідні отвори з одного певного шару на інший. Тому усі перехідні отвори йдуть наскрізь через 4 шари. Під час розводки ДП цей фактор теж враховувався.

Уся розводка ДП виконувалась повністю вручну, з урахуванням всіх вище сказаних умов, без застосування автоматичної розводки у САПР.

**3.5 Розробка блока керування у Altium Designer.**

Altium Designer - комплексна система автоматизованого проектування (САПР) електронних модулів на базі друкованих плат, яка дозволяє виконувати повний спектр проектних завдань, від концепції функціонування до випуску повного комплекту конструкторських і виробничих даних для випуску готової продукції електронних модулів.

Програма AD надає великі можливості для розробки електронних пристроїв. Під час написання дипломного проекту були розглянуті такі можливості програми:

* самостійне створення бібліотеки компонентів (додавання УГП і посадкового місця);
* створення принципової електричної схеми;
* трасування друкованої плати (ручне або автотрасування).

PCB Project – це набір документів, необхідний для виготовлення друкованої плати, данні котрої призначені для рішення однієї конструктивно-закінченої задачі. Документи представляють собою файли різних типів, котрі можуть додаватися в проект. Редагування документів виконується редактором, причому назва редактора співпадає з типом документу.

AD має готові бібліотеки компонентів. Але використання цих бібліотек є в загальному випадку неоптимальним у зв’язку з деякими недоліками:

* УГП компонентів не відповідає ГОСТ;
* Дана бібліотека містить в собі обмежену елементну базу відомих виробників.

Тому для оптимальної, правильної роботи в проектуванні друкованого вузла створюється бібліотека компонентів і моделі у відповідності до усіх вимог.

УГП і посадкові місця компонентів формуються у редакторі бібліотек (Library Editor). В середовищі AD є чотири типи бібліотек: символів, моделей, інтегровані бібліотеки, бази даних.

В кожному редакторі AD є свій набір інструментів і панелей для роботи. Основна панель, з якої виконується робота в редакторі схем є панель Libraries. AD ділить об’єкти, котрі є на полі електричної схеми на графічні та електричні.

До графічних відносять:

* Лінія;
* Дуга, еліптична дуга;
* Сплайн-лінія;
* Еліпс, окружність;
* Прямокутник, округлений прямокутник;
* Многокутник;
* Графік;
* Секторна діаграма.

До електричних об’єктів відносять:

* Лінії електричного зв’язку;
* Схемні компоненти;
* Лінії групового зв’язку у виді джгута;
* Лінії групового зв’язку;
* Ідентифікатори електричних кіл.

Процедура формування схеми насправді проста, і в загальному випадку представляє собою послідовне розміщення і з’єднання на листі електричних і графічних об’єктів.

Створення файлу плати може бути виконано вручну, а також за допомогою майстра PCB Board Wizard, котрий по послідовним етапам опитує інформацію про ДП, яка потім виражається у виді конструктивних правил і параметрів проектування. Під розробкою конструктивних параметрів розуміється етап розробки ДП від створення файлу плати до розміщення компонентів, котрі складаються з чотирьох кроків: формування контуру для ДП, описання стека слоїв, установки кріпильних отворів і визначення заборонених зон для трасування.

В AD задача інтерактивного трасування вирішується інструментом автотрасування, який знаходиться в меню AutoRoute. Автоматичне трасування окремих елементів дає не дуже задовільний результат, так як немає можливості налаштування її алгоритму, який може бути вказаний тільки для трасування всієї плати.

**Висновок до розділу:**

В даному розділі обиралась ДП, на якій надалі будуть розміщуватись елементи. Виробництво плати виконується на підприємстві «Гальванотехніка» так як підприємство, для якого розробляється даний блок керування двигуном, замовляє виготовлення ДП саме на «Гальванотехніці». «Гальванотехніка» виготовляє свої плати на матеріалі FR-4, що є надійним та поширеним матеріалом, комбінованим позитивним методом. В цьому випадку опис та аналіз методів та матеріалу відбувався виходячи з можливостей та технологій підприємства а не навпаки. Розміри та форма ДП, розміщення роз’ємів, які з’єднують блок керування з зовнішніми пристроями, були узгоджені з конструкторами, які розробляють корпус, заздалегідь.

Виходячи з початкових умов, розташування компонентів відбувалось послідовно. ПЛІС також першою розташувалась всередині ДП для подальшого розташування зв’язаних компонентів з нею навколо неї. ДП має 4 шари. Основна кількість сигнальних провідників розташована на верхньому шарі, в той час як провідники живлення розташовувались на третьому шарі. Другий шар повністю виступає як загальна земля пристрою. Це виконано для екранування провідників живлення від сигнальних. Мікросхеми перетворення живлення розташувались в стороні від сигнальних провідників.

Усі запобіжні конденсатори було вирішено розміщувати на нижньому шарі через перехідний отвір до контактів живлення. Розміщення самих елементів виконувалось за алгоритмом якомога меншої довжини та максимально прямого розміщення провідників між FPGA та з’єднаних з нею компонентів. Зроблено висновок, що фінальне розміщення компонентів та трасування є вдалим.

**Розділ 4. РОЗРАХУНКИ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ КОНСТРУКТОРСЬКОГО РІШЕННЯ**

**4.1 Виконання конструкторсько-технологічного розрахунку елементів ДМ**

**4.1.1. Визначення мінімальної ширини друкованого провідника по постійному струму для ланцюгів живлення та землі.**

Мінімальна ширина друкованого провідника по постійному струму bmin I (мм) для ланцюгів живлення та «землі» визначається виразом 4.1:

(4.1)

де *Imax*– максимально можливий струм в ланцюгу, А

*j*доп – допустима щільність струму для ДП, яка виготовлена

комбінованим позитивним методом,

*t*пров– товщина друкованого провідника, яка визначається виразом (4.2)

Друкований провідник виготовлюється комбінованим позитивним методом. Згідно методу виготовлення:

(4.2)

де *hф* – товщина фольги, *hф=*0,035 мм

*hгм* – товщина шара гальванічно осадженої міді, hгм = 0,055 мм

*hхм*– товщина шара хімічно осадженої міді, hхм = 0,0065 мм

*tпров*=0,035+0,055+0,0065=0,0965 мм

Параметр *Imax* в формулі (4.1) визначається як сума струмів, які споживають усі активні елементи схеми. Значення струмів, які споживають активні елементи схеми, наведені у Таблиці 4.1.

У результаті:

*Imax =*0,04+0,03+5+250+5+0,3+50+70+50=430 мА

Тоді мінімальна ширина друкованого провідника на постійному струмі для ланцюгів живлення та «землі» визначається наступним чином:

Таблиця 4.3 Струми, які споживають елементи схеми

|  |  |  |
| --- | --- | --- |
| ІС | Кількість ІС | Iспож, мА |
| AD2S1210 | 1 | 0,04 |
| AD7687 | 1 | 0,03 |
| ECS-3963 | 1 | 5 |
| EP3C25E144I7 | 1 | 250 |
| EPCS16SI16N | 1 | 5 |
| FIN1001M5X | 3 | 0,3 |
| THS4130IDGK | 2 | 50 |
| TPS62000DGS | 1 | 70 |
| TPS79325 | 1 | 50 |

Отримане значення мінімальної ширини провідника bminI = 0,09 мм входить в значення обраного 4 класу точності = 0,15 мм). Таким чином, оптимальна ширина провідника на постійному струмі для ланцюгів живлення та «землі» дорівнює розрахованому значенню.

**4.1.2. Визначення мінімальної ширини провідника з урахуванням допустимого падіння напруги на ньому.**

Мінімальна ширина провідника з урахуванням допустимого падіння напруги на ньому, визначається (4.3):

(4.3)

де ρ–питомий опір провідника, виготовленого комбінованим позитивним методом, ρ=0,0175

Lпров – довжина найдовшого друкованого провідника ДП, +3,3 B, Lпров=414 мм.

Uдоп – допустиме падіння напруги на друкованому провіднику, Uдоп=0,05×Eп,

Uдоп =0,05×5=0,25 В

Lпров=0,414 м

**4.1.3. Визначення номінального діаметру монтажного отвору.**

, (4.4)

де dвэ – діаметр виводу елементів, для якого визначається діаметр монтажного отвору,

∆d – нижнє граничне відхилення від номінального діаметру МО, ∆dмо=0,1 мм

r– різниця між мінімальним діаметром МО та максимальним діаметром виводу елемента, r=0,1…0,2 мм

**4.1.4. Визначення діаметра контактної площини.**

, (4.5)

де Dmin1 – мінімальний ефективний діаметр КМ, мм,

hф – товщина фольги, hф = 0,035 мм. Коефіцієнт 1,5hф враховує підтравлювання фольги друкованого провідника у ширину, 0,03 – КМ виготовлюють комбінованим позитивним методом.

, (4.6)

де dmax– максимальний діаметр отвору в ДП, мм,

bпо - ширина пояска КМ, bпо =0,05 мм (Таблиця 3.1),

δо- похибка розташування центру отвору відносно вузла КС, δо=0,07 мм δкм- похибка розташування центру КМ відносно вузла КС, δкм=0,05

Максимальний діаметр отвору ДП:

dmax=d+∆d+(0,1…0,15) (4.7)

де d– номінальний діаметр МО, мм,

∆d- допуск на діаметр отвору, ∆d=0,05 мм

dmax=d+∆d+(0,1…0,15)=0,7+0,05+0,1=0,85 мм

Максимальний діаметр КМ:

D max = Dmin+0,02, (4.8)

D max = 1,27+0,02=1,29 мм

**4.1.5. Визначення мінімальної ширини провідника.**

bmin=, (4.9)

де –мінімальна ширина провідника. Визначаємо з таблиці класів точності (Таблиця 3.1). Для 4-го класу точності ДМ

bmin=

Максимальна ширина провідника:

bmax=bmin+0,02 (4.10)

bmax=0,23+0,02=0,25 мм

**4.1.6. Визначення мінімальної відстані між провідником та контактною площиною.**



(4.11)

де L0– відстань між центрами отворів та друкованим провідником, які кратні кроку КС, L0=1,25 мм (найгірший випадок).

Dmax - максимальний діаметр КП,

bmax - максимальна ширина провідника,

δкм - похибка розташування центра КП відносно вузла КС, δкм=0,05

δсп - похибка, яка враховує зміщення провідника, δсп=0,05 мм

**4.1.7. Визначення мінімальної відстані між двома сусідніми провідниками (між краями провідників)**



(4.12)

**4.1.8. Визначення мінімальної відстані між двох контактних площин.**

(4.13)

де L01- відстань між центрами сусідніх КП, L01=2,5 мм.

Отримане значення задовольняє 4й клас точності.



**Висновок:** В даній роботі в першу чергу був обраний клас точності для проекту. Пристрій не вимагає певних специфічних умов виготовлення плати. Також 5 клас точності потребує додаткових фінансових витрат. Виходячи з цього був обраний 4 клас точності. Тому всі розрахунки ширини, діаметру доріжок, контактних площадок мають бути менші або такого значення, які подані для 4 класу. В розрахунках було отримано дані менші, що є добрим показником. Після того, як розрахунки підтвердили можливість використання 4 класу, було обрано матеріал для друкованої плати, FR4-4-35-1,5.

**4.2. Електричний розрахунок друкованої плати.**

Розрахунок виконується за умов, що плата виготовлена комбінованим методом, згідно ГОСТ Р 50621-93.

**4.2.1 Визначення падіння напруги на найдовшому друкованому провіднику.**

Падіння напруги на друкованому провіднику визначається:

(4.14)

де ρ - питомий об'ємний опір для комбінованого позитивного методу виготовлення ДП,

ρ = 0,0175

lпр – максимальна довжина друкованого провідника, lпр=0,414м.

tпр - товщина провідника, tпр = 0,0965 мм

Imax – максимальний струм у провіднику, Imax = 430 мА

= 0,129 В

Розраховане падіння напруги не перевищує 5% від напруги живлення (Uж = 5 В).

**4.2.2 Визначення потужності втрат двосторонньої друкованої плати.**

Потужність втрат визначається:

(4.15)

де *f*=1, тому що розрахунок виконується на постійному струмі;

*tg*σ – тангенс кута діелектричних втрат для матеріала ДП, *tg*σ =0,002 для

матеріалу FR4;

С – ємність ДП

(4.16)

де ε – діелектрична проникність, ε = 4,5 для FR4

Sm - площа металізації, Sm =552,96 мм2

h - товщина ДП, мм

== 14,9 нФ

Рпот =

**4.2.3 Визначення ємності між двома сусідніми провідниками, які розташовуються на одній стороні ДП та мають однакову ширину.**

(4.17)

де S – відстань між двома паралельними провідниками, S=0,2 мм;

bпр - ширина друкованого провідника, мм;

tпр - товщина друкованого провідника, мм;

lпр - довжина взаємного перекриття двох паралельних провідників, мм;

**4.2.4 Визначення взаємної індуктивності двох паралельних провідників однакової довжини.**

(4.18)

де lпр – довжина перекриття паралельних провідників, lпр=4 см;

Lо-відстань між осьовими лініями двох паралельних провідників, Lо=0,125 см;

нГн

**Висновки:** Отримано значення падіння напруги на найдовшому провіднику 0,129 В. Воно знаходиться дуже далеко від межі, що дорівнює 5% від напруги живлення. Потужність втрат дорівнює 26,95 нВт. Це незначна величина. Паразитна ємність (115 пФ) та індуктивність (0,144 нГн) не впливають на роботу друкованого вузлу.

**4.3 Розрахунок надійності друкованого вузла.**

Найбільш точна кількісна міра надійності кожного конструктивного елементу – його індивідуальне напрацювання до моменту виникнення відмови.

Важлива характеристика надійності - середній час безвідмовної роботи визначається:

Тср = , (4.19)

Інтенсивність відмов ЕРЕ є їх вихідною характеристикою надійності, залежить від режиму роботи та ступеню тяжкості таких зовнішніх впливів, як температура, тепловий удар, вологість, вібрації і т.д.

Тоді можна записати:

λе = λое · K1 · K2 ·…· Kn , (4.20)

де λое - інтенсивність відмов елементу при нормальних умовах роботи (температура навколишнього середовища Тºокр.ср = 20 ± 5ºС, відносна вологість 65 ± 15%);

коефіцієнт електричного навантаження Кn = 1 , К1, К2, Кн - поправочні коефіцієнти, що враховують режими роботи та умови експлуатації.

Для врахування впливу режиму роботи на інтенсивність відмов ЕОА вводять коефіцієнт навантаження, що дорівнює відношенню навантаженню в робочому режимі до навантаження в номінальному режимі:

Кн = , (4.21)

Коефіцієнт навантаження для резисторів

(4.22)

для конденсаторів

Кн.c = , (4.23)

Розраховуємо коефіцієнти навантаження:

Резистори

Таблиця 4.4

|  |  |  |
| --- | --- | --- |
| *R1,R2* | *R-0603 0,125 Вт 10 кОм 1% Yageo* | *0,02* |
| *R3* | *R-0603 0,125 Вт 27 Ом 1% Yageo* | *7,407407* |
| *R4* | *R-0603 0,125 Вт 100 кОм 1% Yageo* | *0,002* |
| *R5* | *R-0603 0,125 Вт 1 кОм 1% Yageo* | *0,2* |
| *R6* | *R-0603 0,125 Вт 100 кОм 1% Yageo* | *0,002* |
| *R7* | *R-0603 0,125 Вт 1 кОм 1% Yageo* | *0,2* |
| Таблиця 4.4 (продовження) | | |
| *R8* | *R-0603 0,125 Вт 100 кОм 1% Yageo* | *0,002* |
| *R9* | *R-0603 0,125 Вт 1 кОм 1% Yageo* | *0,2* |
| *R10,R11* | *R-0603 0,125 Вт 51 кОм 1% Yageo* | *0,003922* |
| *R12,R13* | *R-0603 0,125 Вт 11,3 кОм 1% Yageo* | *0,017699* |
| *R14,R15* | *R-0603 0,125 Вт 24,3 кОм 1% Yageo* | *0,00823* |
| *R16,R17* | *R-0603 0,125 Вт 51 кОм 1% Yageo* | *0,003922* |
| *R18* | *R-0603 0,125 Вт 100 кОм 1% Yageo* | *0,002* |
| *R19* | *R-0603 0,125 Вт 1 кОм 1% Yaganov* | *0,2* |
| *R20* | *R-0603 0,125 Вт 100 кОм 1% Yageo* | *0,002* |
| *R21* | *R-0603 0,125 Вт 1 кОм 1% Yageo* | *0,2* |
| *R22* | *R-0603 0,125 Вт 100 кОм 1% Yageo* | *0,002* |
| *R23* | *R-0603 0,125 Вт 1 кОм 1% Yageo* | *0,2* |
| *R24,R25* | *R-0603 0,125 Вт 51 кОм 1% Yageo* | *0,003922* |
| *R26,R27* | *R-0603 0,125 Вт 11,3 кОм 1% Yageo* | *0,017699* |
| *R28* | *R-0603 0,125 Вт 10k 1% Yaganov* | *0,02* |
| *R29-R36* | *R-0603 0,125 Вт 27 Ом 1% Yageo* | *7,407407* |
| *R37* | *R-1206 0,125 Вт 27 1% Yageo* | *7,407407* |
| *R38* | *R-0603 0,125 Вт 47 Ом 1% Yageo* | *4,255319* |
| *R39-R43* | *R-0603 0,125 Вт 27 Ом 1% Yageo* | *7,407407* |
| *R44,R45* | *R-0603 0,125 Вт 24,3 кОм 1% Yageo* | *0,00823* |
| *R46,R47* | *R-0603 0,125 Вт 47 Ом 1% Yageo* | *4,255319* |
| *R48,R49* | *R-0603 0,125 Вт 27 Ом 1% Yageo* | *7,407407* |
| *R50* | *R-0603 0,125 Вт 47 Ом 1% Yageo* | *4,255319* |
| *R51-R53* | *R-0603 0,125 Вт 100 Ом 1% Yageo* | *2* |
| *R54* | *R-1206 0,125 Вт 27 1% Yaganov* | *7,407407* |
| *R55* | *R-1206 0,125 Вт 47 кОм 1% Yageo* | *4,255319* |
| *R56* | *R-0603 0,125 Вт 10 кОм 1% Yageo* | *0,02* |
| *R57* | *R-1206 0,125 Вт 27 1% Yageo* | *7,407407* |
| *R58,R59* | *R-0603 0,125 Вт 10 кОм 1% Yageo* | *0,02* |
| *R60* | *R-1206 0,125 Вт 15 кОм 1% Yageo* | *0,013333* |
| Таблиця 4.4 (продовження) | | |
| *R61* | *R-1206 0,125 Вт 100 кОм 1% Yageo* | *0,002* |
| *R62-R65* | *R-0603 0,125 Вт 10 кОм 1% Yageo* | *0,02* |
| *R66* | *R-1206 0,125 Вт 95,3 кОм 1% Yageo* | *0,002099* |
| *R67* | *R-1206 0,125 Вт 165 кОм 1% Yageo* | *0,001212* |
| *R68* | *R-0603 0,125 Вт 10 кОм 1% Yageo* | *0,02* |
| *R69-R71* | *R-1206 0,125 Вт 0 Ом 1% Yageo* | *0* |
| *R72-R75* | *R-0603 0,125 Вт 27 Ом 1% Yageo* | *7,407407* |

Конденсатори:

Необхідно визначити результуючу інтенсивність відмов друкованого вузлу блока керування двигуном.

Друкований вузол відноситься до наземної апаратури, експлуатується при Тр = 25º С, інші умови експлуатації нормальні.

Вихідні дані для розрахунку – схема принципова, перелік елементів, часова діаграма та інтенсивність відмов “компонентів надійності” від температурних впливів.

По картам робочих режимів необхідно визначити коефіцієнти навантаження, температурні коефіцієнти ІС та інших ЕРЕ, підрахувати кількість всіх елементів.

Вихідні дані для визначення λр зведені до Таблиці.4.6.

Таблиця 4.5.

|  |  |  |
| --- | --- | --- |
| *C1-C7* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* | *0,1*  *0,1* |
| *C8,C9* | *C-0603 50 В 68 пФ 5% X7R Yageo* |
| *C10* | *C-1206 50 В 10 мкФ 5% X7R Yageo* |
| *C11,C12* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C13* | *C-0603 50 В 68 пФ 5% X7R Yageo* |
| *C14-C16* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C17* | *C-0603 50 В 68 пФ 5% X7R Yageo* |
| *C18,C19* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C20* | *C-1206 50 В 10 мкФ 5% X7R Yageo* |
| *C21,C22* | *C-0603 50 В 68 пФ 5% X7R Yageo* |
| *C23* | *C-1206 50 В 10 мкФ 5% X7R Yageo* |
| *C24-C26* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C27,C28* | *C-0603 50 В 1,8 нФ 5% X7R Yageo* |
| *C29-C50* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C51,C52* | *C-1206 50 В 10 мкФ 5% X7R Yageo* |
| *C53-C77* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C78* | *C-1206 50 В 10 мкФ 5% X7R Yageo* |
| *C79* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C80* | *C-0603 50 В 220 пФ 5% X7R Yageo* |
| *C81* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C82* | *C-0603 50 В 220 пФ 5% X7R Yageo* |
| *C83,C84* | *C-1206 50 В 10 мкФ 5% X7R Yageo* |
| *C85,C86* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |
| *C87-C90* | *C-1206 50 В 10 мкФ 5% X7R Yageo* |
| *C91,C92* | *C-0603 50 В 0,1 мкФ 5% X7R Yageo* |

Таблиця 4.6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Компонент** | **N** | **λ0e·10-8, год-1** | **Кн** | **at** | **ae** | **N ·λ0e· Кн·at·ae·10-8** |
| **Конденсатор** | 92 | 2 | 0,1 | 0,4 | 20 | 147,2 |
| **Резистор** |  |  |  |  |  | 0 |
| R1,R2 | 2 | 4 | 0,02 | 0,15 | 20 | 0,48 |
| R3 | 1 | 4 | 7,407407 | 0,15 | 20 | 88,88889 |
| R4 | 1 | 4 | 0,002 | 0,15 | 20 | 0,024 |
| R5 | 1 | 4 | 0,2 | 0,15 | 20 | 2,4 |
| R6 | 1 | 4 | 0,002 | 0,15 | 20 | 0,024 |
| R7 | 1 | 4 | 0,2 | 0,15 | 20 | 2,4 |
| R8 | 1 | 4 | 0,002 | 0,15 | 20 | 0,024 |
| R9 | 1 | 4 | 0,2 | 0,15 | 20 | 2,4 |
| R10,R11 | 2 | 4 | 0,003922 | 0,15 | 20 | 0,094118 |
| R12,R13 | 2 | 4 | 0,017699 | 0,15 | 20 | 0,424779 |
| R14,R15 | 2 | 4 | 0,00823 | 0,15 | 20 | 0,197531 |
| R16,R17 | 2 | 4 | 0,003922 | 0,15 | 20 | 0,094118 |
| R18 | 1 | 4 | 0,002 | 0,15 | 20 | 0,024 |
| R19 | 1 | 4 | 0,2 | 0,15 | 20 | 2,4 |
| R20 | 1 | 4 | 0,002 | 0,15 | 20 | 0,024 |
| R21 | 1 | 4 | 0,2 | 0,15 | 20 | 2,4 |
| R22 | 1 | 4 | 0,002 | 0,15 | 20 | 0,024 |
| R23 | 1 | 4 | 0,2 | 0,15 | 20 | 2,4 |
| R24,R25 | 2 | 4 | 0,003922 | 0,15 | 20 | 0,094118 |
| R26,R27 | 2 | 4 | 0,017699 | 0,15 | 20 | 0,424779 |
| R28 | 1 | 4 | 0,02 | 0,15 | 20 | 0,24 |
| R29-R36 | 8 | 4 | 7,407407 | 0,15 | 20 | 711,1111 |
| Таблиця 4.6 (продовження) | | | | | | |
| R37 | 1 | 4 | 7,407407 | 0,15 | 20 | 88,88889 |
| R38 | 1 | 4 | 4,255319 | 0,15 | 20 | 51,06383 |
| R39-R43 | 5 | 4 | 7,407407 | 0,15 | 20 | 444,4444 |
| R44,R45 | 2 | 4 | 0,00823 | 0,15 | 20 | 0,197531 |
| R46,R47 | 2 | 4 | 4,255319 | 0,15 | 20 | 102,1277 |
| R48,R49 | 2 | 4 | 7,407407 | 0,15 | 20 | 177,7778 |
| R50 | 1 | 4 | 4,255319 | 0,15 | 20 | 51,06383 |
| R51-R53 | 3 | 4 | 2 | 0,15 | 20 | 72 |
| R54 | 1 | 4 | 7,407407 | 0,15 | 20 | 88,88889 |
| R55 | 1 | 4 | 4,255319 | 0,15 | 20 | 51,06383 |
| R56 | 1 | 4 | 0,02 | 0,15 | 20 | 0,24 |
| R57 | 1 | 4 | 7,407407 | 0,15 | 20 | 88,88889 |
| R58,R59 | 2 | 4 | 0,02 | 0,15 | 20 | 0,48 |
| R60 | 1 | 4 | 0,013333 | 0,15 | 20 | 0,16 |
| R61 | 1 | 4 | 0,002 | 0,15 | 20 | 0,024 |
| R62-R65 | 4 | 4 | 0,02 | 0,15 | 20 | 0,96 |
| R66 | 1 | 4 | 0,002099 | 0,15 | 20 | 0,025184 |
| R67 | 1 | 4 | 0,001212 | 0,15 | 20 | 0,014545 |
| R68 | 1 | 4 | 0,02 | 0,15 | 20 | 0,24 |
| R69-R71 | 3 | 4 | 0 | 0,15 | 20 | 0 |
| R72-R75 | 4 | 4 | 7,407407 | 0,15 | 20 | 355,5556 |
| **Кварцовий генератор** | 1 | 4,6 | 0,8 | 1 | 20 | 73,6 |
| **ІС** | 13 | 1,17 | 1 | 1 | 20 | 304,2 |
| **Друкована плата** | 4 | 10 | 1 | 1 | 20 | 800 |
| **Контакт роз’єма** | 44 | 2 | 1 | 1 | 20 | 1760 |
| Таблиця 4.6 (продовження) | | | | | | |
| **Пайка виводів** | 291 | 0,05 | 1 | 1 | 20 | 291 |
| **Транзистор** | 3 | 16 | 1 | 1 | 20 | 960 |
| **Перехідні отвори** | 153 | 0,0375 | 1 | 1 | 20 | 114,75 |
|  | Сумарна інтенсивність відмов друкованого вузла | | | | | 6841,448 |
|

В Таблиці 4.6:

**ae –** поправочний коефіцієнт на вплив зовнішніх впливів (для переносної апаратури **ae** = 20),

**at** - поправочний температурний коефіцієнт.

Показники інтенсивності відмов, що наведені в таблиці, дещо завищені, що дозволяє виконати розрахунок для «найгіршого випадку».

Результуюча інтенсивність відмов дорівнює сумі інтенсивності відмов компонентів:

(4.24)

Середній час напрацювання до першої відмови:

Тср = .

Ймовірність безвідмовної роботи на протязі року:

*Р=*0,55

Ймовірність відмов на протязі року:

Q(t)=1-0,55=0,45

Графік залежності безвідмовної роботи ДВ та ймовірність відмов ДВ від часу представлені на наступному графіку (Рисунок 4.1)

|  |
| --- |
|  |
| Рисунок 4.1. Графік залежності безвідмовної роботи ДВ і ймовірності відказу ДВ від часу. |

**Висновок:** отримане значення напрацювання на відмову відповідає технічному завданню. З одного боку це за умови безперервної роботи, що на практиці для даного приладу не завжди можливо. З іншої сторони в цих теоретичних розрахунках не враховані такі фактори як старіння приладу, його знос і т.д..

**Висновок до розділу:**

В даному розділі виконались розрахунки, які повинні підтвердити коректність роботи обраних елементів, виготовленої друкованої плати, правильність обраного класу точності.

Пристрій не вимагає певних специфічних умов виготовлення плати. Габарити плати та конструктивне розміщення елементів дозволяють застосовувати 3 клас точності. Але посадокві місця деяких компонентів, таких як FPGA мають відстань між контактними майданчиками 0,2 мм, що не дозволяє використовувати 3 клас точності. Оскільки 3 клас точності надає мінімальну відстань 0,25 мм. З цих міркувань був обраний 4 клас точності. Тому всі розрахунки ширини, діаметру доріжок, контактних площадок мають бути менші або такого значення, які подані для 4 класу. Розрахункові значення отримано менші, що є добрим показником.

Отримано значення падіння напруги на найдовшому провіднику 0,129 В. Воно знаходиться дуже далеко від межі, що дорівнює 5% від напруги живлення. Потужність втрат дорівнює 26,95 нВт. Це незначна величина. Паразитна ємність (115 пФ) та індуктивність (0,144 нГн) не впливають на роботу друкованого вузлу.

Наступним було отримано значення напрацювання на відмову, яке відповідає технічному завданню. В результаті було розраховано, що пристрій може працювати 15000 годин. За умов, що блок керування використовуватиметься тільки в лабораторних умовах, цього значення буде достатньо, адже досягти такої кількості годин можливо за умови безперервної роботи, що на практиці для даного приладу не є можливо. З іншої сторони в цих теоретичних розрахунках не враховані такі фактори як старіння приладу, його знос і т.д.

**Розділ 5. ПРОГРАМУВАННЯ ТА МОДЕЛЮВАННЯ ВІРТУАЛЬНОГО ПРИСТРОЮ**

5.1. Робота драйвера датчика кута.

Мікросхема AD2S1210 отримує дані з зовнішнього датчику кута. Цими даними є аналогові диференційні сигнали SIN, COS. Задача драйвера – представити ці аналогові сигнали у вигляді цифрових, та передати їх далі на FPGA до центрального процесору.

Мікросхема видає на шину даних паралельний цифровий 16 бітний сигнал. Отже, необхідно, щоб контролер видавав інформацію в паралельний порт.

Принцип формування сигналів синуса та косинуса з датчику кута зображений на Рисунку 5.1.



Рисунок 5.1. Принцип формування сигналів синусу та косинусу.

Для того, що правильно приймати та обробляти дані, що поступають на вхід драйверу, необхідно його запрограмувати. Мікросхема працює в двох режимах: режимі конфігурації та нормальному режимі. Режим конфігурації використовується для програмування регістрів, щоб установити частоту збудження, розрядність вихідних даних, порогову напругу, при якій буде працювати драйвер.

Частоту збудження розраховують за формулою (5.1).

(5.1)

Контролер має зчитувати кут з резольвера в розрядності 16 біт. Тому, згідно [11, Таблиця 7] обираємо максимальну частоту збудження 10кГц (Рисунок 5.2).

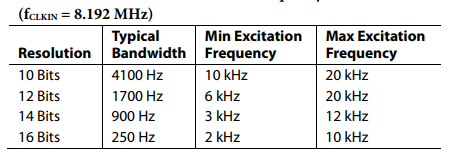


Рисунок 5.2. Максимальна частота збудження

Керуючі сигнали А0, А1 необхідні для зчитування кута представлені на Рисунку 5.3.

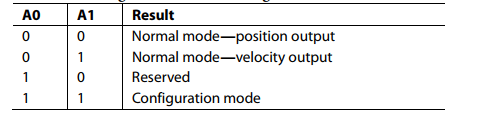


Рисунок 5.3 Керуючі сигнали А0, А1.

Керуючі сигнали *RES0, RES1* задають розрядність вихідного цифрового сигналу. Так як драйвер весь час працює з 16-розрядним числом, на ці входи постійно подається логічна «1» у вигляді напруги живлення 3,3 В.

Згідно карти регістрів необхідно обирати регістри, які необхідно запрограмувати, або з яких необхідно зчитати дані. Карта регістрів представлена на Рисунку 5.4.

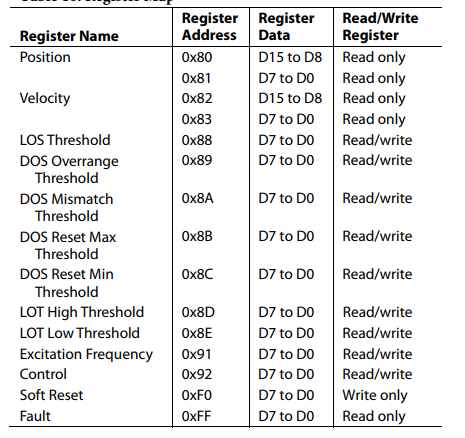


Рисунок 5.4. Карта регістрів драйверу кута.

Часова діаграма запису конфігурації драйверу зображена на Рисунку 5.5. Опис усіх операцій встановлення бітів описано в Додатку Б. Усі часові затримки для конфігурації контактів наведені в [11, Таблиця 2]. Після режиму запису конфігурації наступає режим зчитування драйверу. Він зображений на Рисунку 5.6.

Після запису і зчитування, якщо немає критичних помилок, процесор переходить в нормальний режим зчитування даних з резольверу. До драйвера надходить інформація про позицію та швидкість. Часову діаграму програмування мікросхеми для режиму зчитування даних з резольверу представлено на Рисунку 5.7.

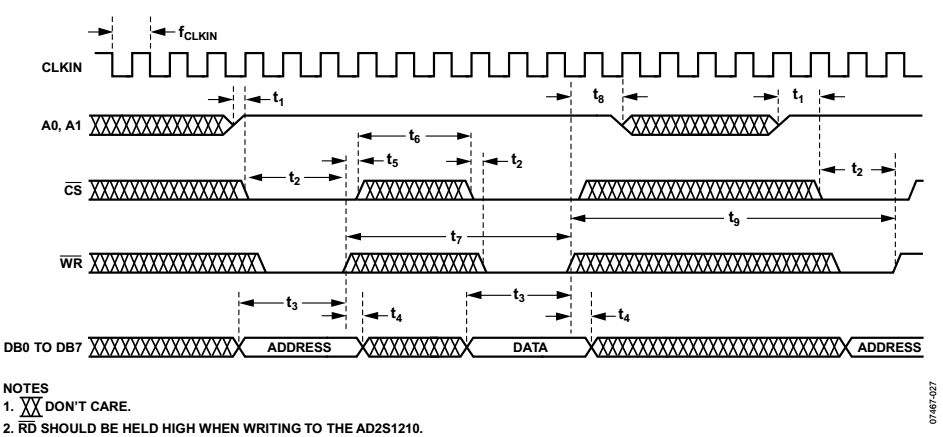


Рисунок 5.5. Конфігурація запису драйверу кута.

Для того, щоб перевірити коректність та правильність роботи драйверу, існує регістр збою. Він дає можливість перевірити вісім окремих умов. Перевіряючи цей регістр, можна визначити, де саме виникла та чи інша помилку. Описання регістру збою представлено на Рисунку 5.8.

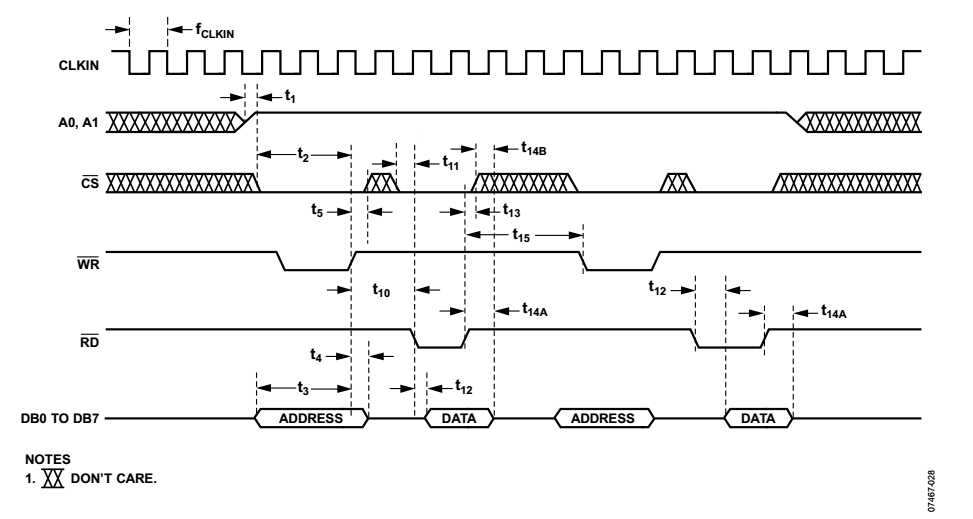


Рисунок 5.6. Конфігурація зчитування драйверу кута.

Дані поступають до FPGA, де надалі будуть оброблятися.

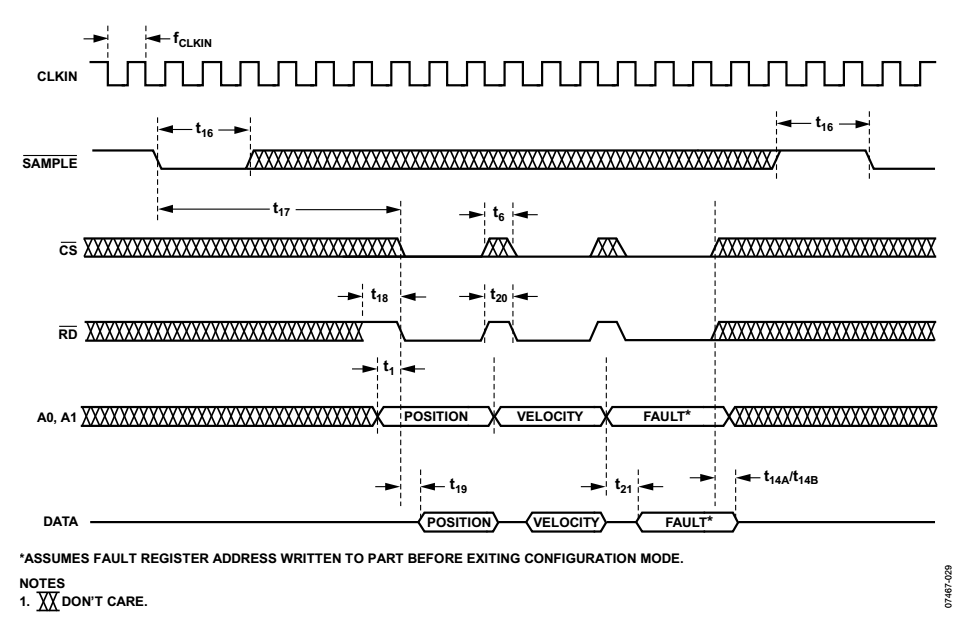


Рисунок 5.7. Зчитування даних з резольверу.

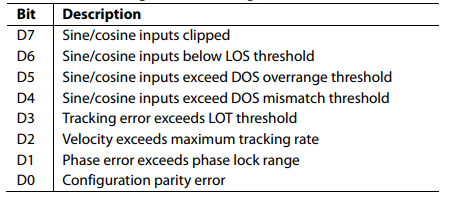


Рисунок 5.8. Регістр збою.

5.2 Робота контролера АЦП.

АЦП передає дані про струм двигуна у вигляді двійкового послідовного сигналу. Контролер перетворює цей сигнал в 16-розрядний паралельний код та передає їх до центрального процесора. Основу контролера складає машина станів, написана на мові Verilog.

Згідно часової діаграми (Рисунок 5.9), коли CNV=0 і SDI=0, то SDO=0. Коли SDI=0, а SCK йде в одиницю, то по передньому фронту CNV ініціалізує вимірювання. Коли почалось вимірювання, CNV повинен триматися в одиниці на протязі всього вимірювання. Тільки після вимірювання можна опустити CNV. Дані записуються у внутрішній зсувний регістр, далі виштовхуються по спадаючому фронту SCK. Для повної передачі даних потрібно 16+1 спадаючих фронтів SCK.

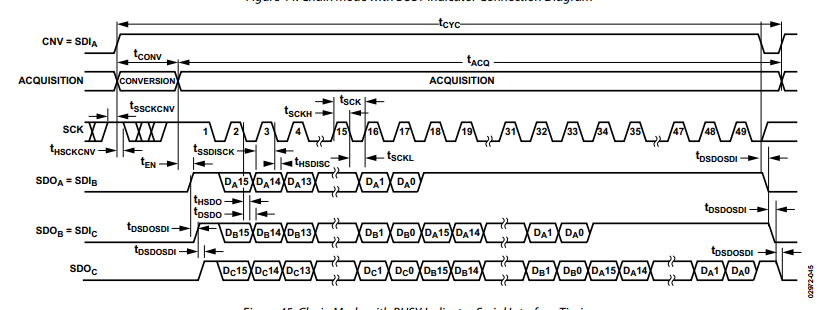


Рисунок 5.9. Векторна діаграма роботи АЦП.

5.3 Робота контролера ШІМ.

ШІМ-сигнал формується драйвером, який синтезується всередині FPGA.

Вхідна частота драйвера складає 60 МГц. Частота ШІМ розраховується за формулою (5.2):

(5.2)

Драйвер керується 12-бітним сигналом, який задає протяжність меандру ШІМ-сигналу.

На вхід драйвера поступають величина струму з АЦП, та максимальна величина струму, задана оператором. Якщо величина струму з АЦП перевищує задану максимальну величину струму, ШІМ блокується, і з контакту OFF поступає сигнал, який вимикає двигун.

Старший розряд вхідного сигналу Pwm\_in задає напрямок руху двигуна. Якщо «1», то вихідний сигнал, який задає напрямок руху двигуна DIR = 1, і двигун рухається за годинниковою стрілкою. Якщо «0», то вихідний сигнал, який задає напрямок руху двигуна DIR = 0, і двигун рухається проти годинникової стрілки.

Молодші 11 розрядів задають скважність ШІМ по модулю.

Значення вихідного сигналу з драйвера формується порівнянням між собою заданої константи для величини ШІМ і лічильника. Програма для реалізації ШІМ надана в Додатку Б.

**5.4. Алгоритм роботи програми.**

Алгоритм роботи основної програми представлений на Рисунку 5.10.

Алгоритм функції «Формування режиму» представлений на Рисунку 5.11.

Алгоритм функції «Налаштування кута повороту» представлений на Рисунку 5.12.

Алгоритм функції «Налаштування струму двигуна» представлений на Рисунку 5.13.

Усі коди програм, які реалізують ці алгоритми, надано у Додатку Б.

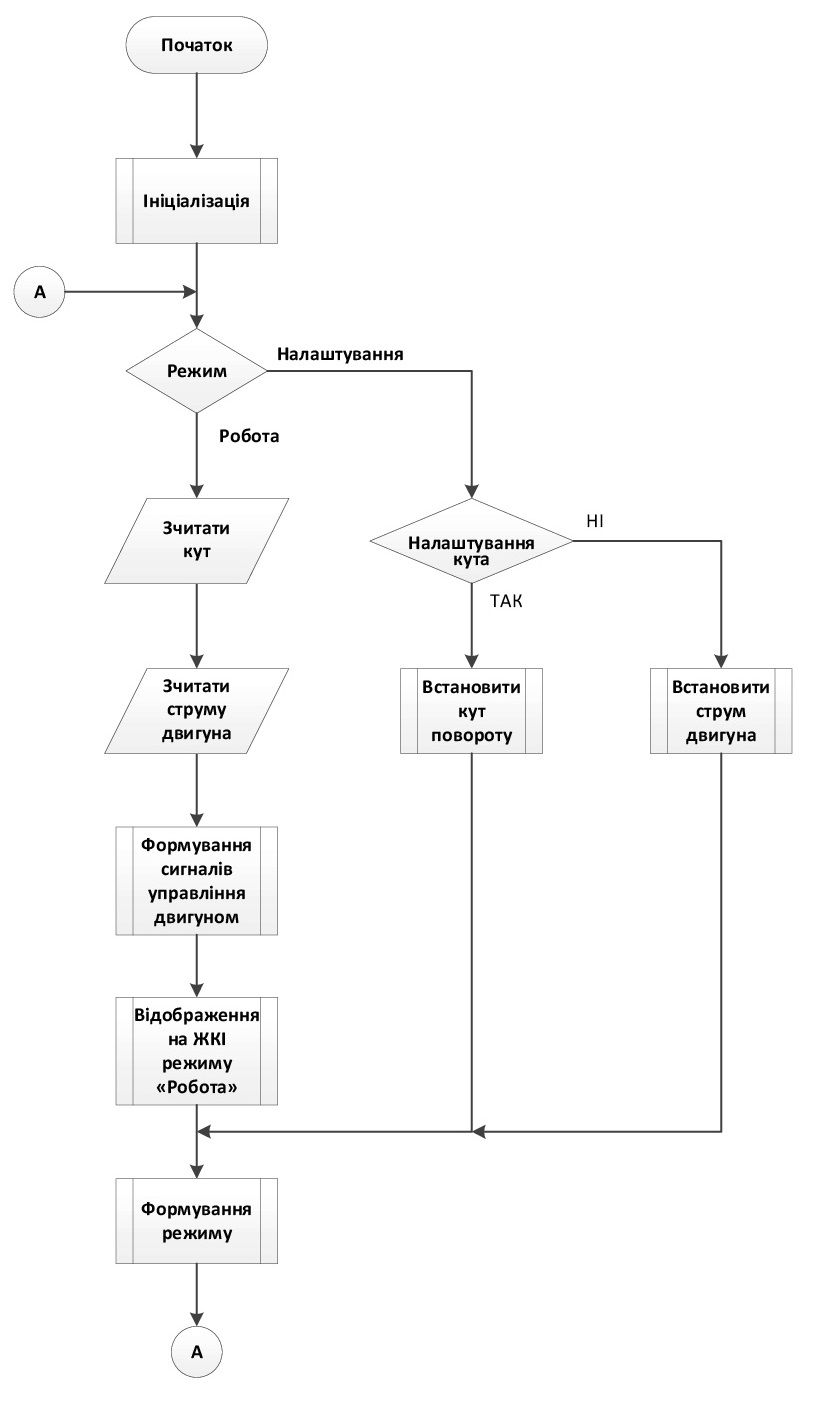


Рисунок 5.10 Алгоритм роботи основної програми.

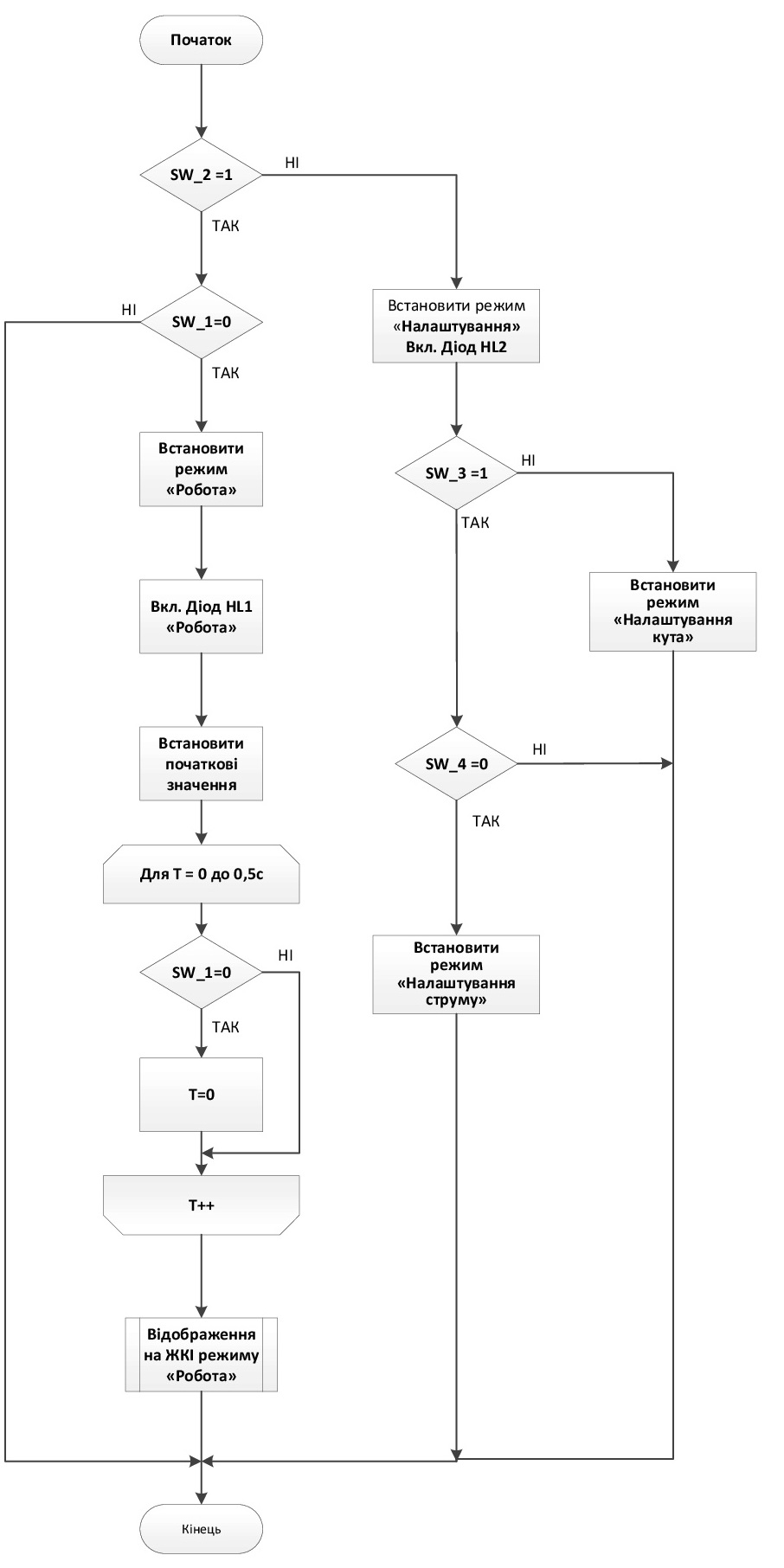


Рисунок 5.11. Алгоритм функції «Формування режиму».

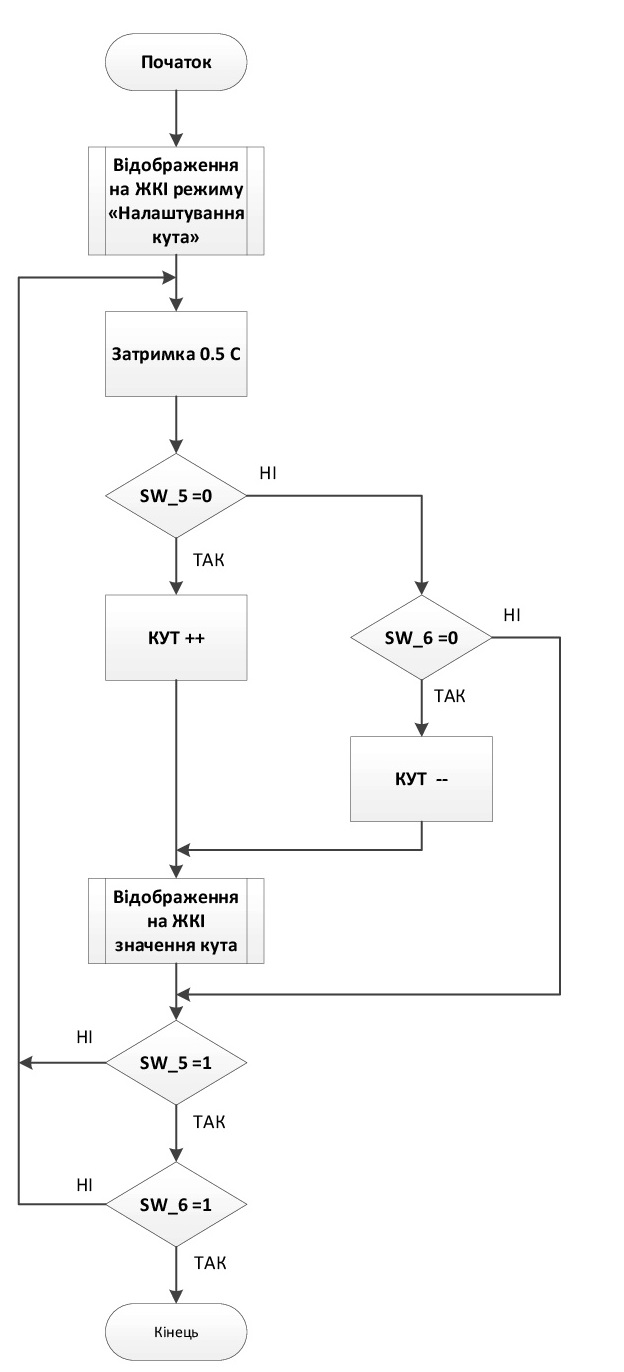


Рисунок 5.12. Алгоритм функції «Налаштування кута повороту»

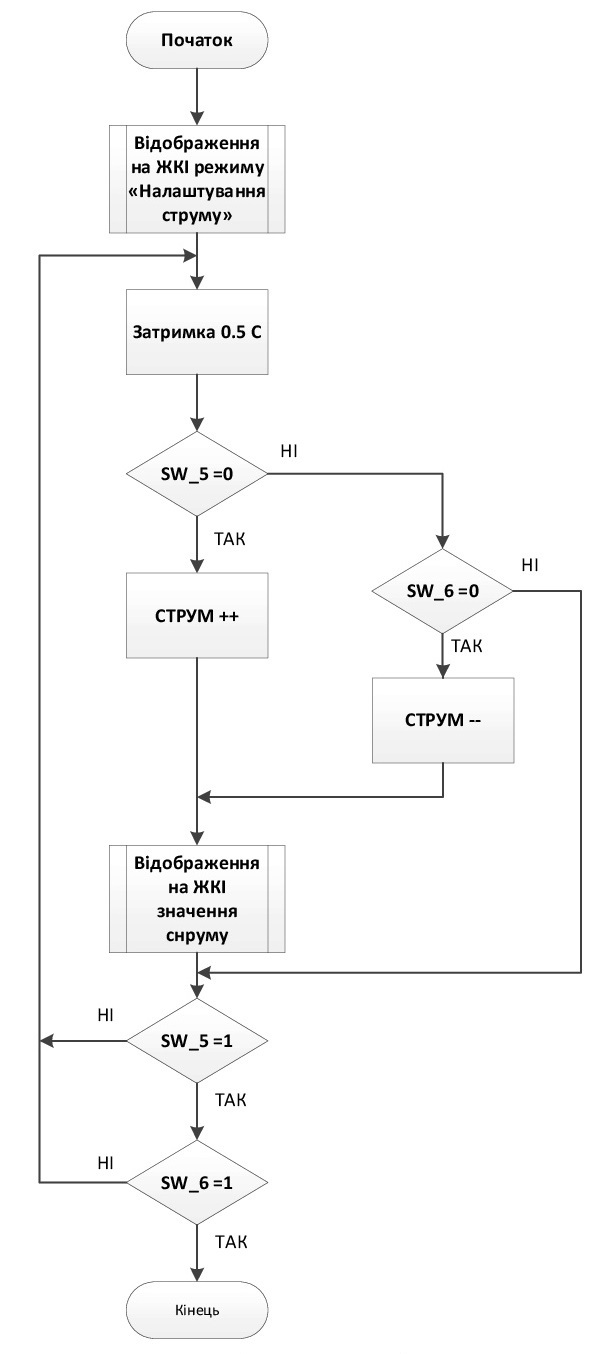


Рисунок 5.13. Алгоритм функції «Налаштування струму двигуна».

**Висновок до розділу:**

Побудова програмного коду для програмування мікросхем відбувалось у середовищі Quartus II. Написання програмного коду складалось з двох частин:

* конфігурація мікросхем через центральний процесор;
* обробка даних з мікросхем та виведення керуючих сигналів.

Конфігурація виконувалась згідно конфігураційним векторним діаграмам, які описані в технічній документації.

Побудова синтезованих компонентів на мові Verilog мала за основу принцип роботи кінцевих автоматів.

Зараз пристрій знаходиться на етапі налагодження на підприємстві.

**ВИСНОВКИ**

В даному дипломному проекті розроблено схемотехнологічне рішення, друкований вузол, програмне забезпечення блока керування двигуном по положенню. Даний пристрій має наступні технічні характеристики, які відповідають вимогам технічного завдання:

|  |  |
| --- | --- |
| Принцип дії |  |
| Інтерфейс зв’язку з системою | Індикація режиму роботи на ЖКІ-дисплеї, задання режиму роботи через клавіатуру |
| Габаритні розміри | 120х100 см. |
| Маса | 63г. |
| Живлення | 5 В, 12 В. |
| Режими роботи | Основний (вимірювання); налаштування. |
| Діапазон виміру кутів | ±60о |

В ході обґрунтування даного рішення виконані наступні етапи:

1. В першому розділі зроблено пошук існуючих патентних рішень керування двигуном. Отримано інформацію про те, що більшість запропонованих рішень є або застарілими, або вимагають постійне підключення до ПК, що не є можливим в лабораторних умовах. Це дозволило сформувати уявлення про конструктивні та структурні рішення, характеристики та параметри розробляємого блока керування двигуном по положенню.

2. Розроблено структурну схему, схему електричну принципову. Розрахунок схеми електричної принципової підтверджує правильність схемотехнічного рішення. Отримана схема є доступною для розуміння роботи пристрою, відповідає ГОСТ, побудована згідно рекомендаціям технічних документацій елементів. Це дає змогу швидко та доступно зрозуміти принцип роботи пристрою, окремих його каскадів.

3. Зроблено аналіз класів точності, типів та матеріалів ДП. Обрано 4-шарову плату з FR-4, за 4 класом точності. Розглянуто розміщення компонентів на ДП та проектування у Altium Designer. Розроблена плата має високу щільність монтажу, підвищену надійність, високу механічну міцність кріплення, та відповідає вимогам ТЗ.

4. Виконано розрахунки, що підтверджують працездатність схеми (Розділ 4), а саме:

- конструкторсько-технологічний розрахунок, який підтвердив правильність вибору 4-го класу точності;

- електричний розрахунок друкованої плати, підтвердив оптимальний вибір трасування провідників. Оскільки, потужність втрат на постійному струмі 26,95 нВт, падіння напруги 0,129 В, паразитна ємність 115 пФ, паразитна індуктивність 0,144 нГн – ці значення не впливають на роботу ДВ;

- розрахунок надійності ДВ. Отримане значення середнього часу напрацювання на відмову задовольняє вимоги

технічного завдання (10000 год);

5. Розроблено програмне забезпечення (ПЗ) для керування двигуном по положенню. Пристрій працює в двох режимах:

- режим конфігурації, в якому центральний процесор програмує інші мікросхеми для режиму роботи;

- режим роботи, в якому центральний процесор отримує дані з двигуна, та генерує керуючі сигнали до двигуна.

Це дає змогу керувати двигуном в лабораторних умовах згідно ТЗ. Дане рішення дозволяє змінювати в майбутньому програмну частину для зміни режимів роботи без зміни апаратної частини.

6. Розроблена конструкторська документація на блок керування двигуном по положенню.

Проект виконано в повному об’ємі, а розрахунки й моделювання підтверджують працездатність пристрою та задовольняють вимогам ТЗ.

СПИСОК ЛІТЕРАТУРИ

1. Ковчин С.А., Сабинин Ю.А.Теория электропривода - ЭНЕРГОАТОМИЗДАТ, Санкт-Петербург, 2000.
2. Каган В.Г. Электроприводы с предельным быстродействием для систем воспроизведения движений – М.: Энергия, 1975.
3. Устройство для управления серводвигателем [електронний ресурс] – режим доступу: <http://www.findpatent.ru/img_show/3736854.html>
4. Позиционный серводвигатель[електронний ресурс] – режим доступу: <http://patents.su/2-1222907-pozicionnyjj-servodvigatel.html>
5. Система автоматического управления серводвигателями [електронний ресурс] – режим доступу: <https://cyberleninka.ru/article/v/sistema-avtomaticheskogo-upravleniya-servoprivodami>
6. TPS79325 Datasheet [електронний ресурс] – режим доступу: <http://www.ti.com/lit/ds/symlink/tps793.pdf>
7. TPS6200x Datasheet [електронний ресурс] – режим доступу: <http://www.ti.com/lit/ds/symlink/tps62003.pdf>
8. Cyclone III Device Datasheet [електронний ресурс] – режим доступу: <https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyc3/cyc3_ciii52001.pdf>
9. EPCS16SI16N Datasheet [електронний ресурс] – режим доступу: <https://www.altera.com/en_US/pdfs/literature/hb/cfg/cyc_c51014.pdf>
10. TS2640N691E125 Datasheet[електронний ресурс] – режим доступу: <https://www.encoder-technology.com/images/product_specifications/fa-solver.pdf>
11. AD2S1210 Datasheet [електронний ресурс] – режим доступу: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD2S1210.pdf>
12. AD7687BRMZ Datasheet [електронний ресурс] – режим доступу: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD7687.pdf>
13. ECS-3963 Datasheet [електронний ресурс] – режим доступу: <http://www.ecsxtal.com/store/pdf/ecs-3961_3963.pdf>
14. FIN1001 Datasheet [електронний ресурс] – режим доступу: <https://www.fairchildsemi.com/datasheets/FI/FIN1001.pdf>
15. THS4130IDGK Datasheet [електронний ресурс] – режим доступу: <http://www.ti.com/lit/ds/symlink/ths4131.pdf>
16. BC846B Datasheet [електронний ресурс] – режим доступу: <http://www.onsemi.com/pub/Collateral/BC846-D.PDF>

ГОСТ 15150-69 МАШИНЫ, ПРИБОРЫ И ДРУГИЕ ТЕХНИЧЕСКИЕ ИЗДЕЛИЯ

ГОСТ 12.2.007-75 Система стандартов безопасности труда ИЗДЕЛИЯ ЭЛЕКТРОТЕХНИЧЕСКИЕ

ГОСТ 23571-86 ПЛАТЫ ПЕЧАТНЫЕ Основные параметры конструкции

ГОСТ Р 50621-93 ПЛАТЫ ПЕЧАТНЫЕ ОДНО- И ДВУСТОРОННИЕ С НЕМЕТАЛЛИЗИРОВАННЫМИ ОТВЕРСТИЯМИ

|  |  |
| --- | --- |
| ПОГОДЖЕНО | ЗАТВЕРДЖЕНО |
| Начальник сектору | Директор |
| ТОВ «Радіонікс» | ТОВ «Радіонікс» |
| В.Ю. Танигін | С.Б. Зав’ялов |

Додаток А

Технічне завдання на проектування

1. **Найменування та галузь використання**

Блок керування двигуном по положенню. Використовується в стенді вимірювання параметрів прямих приводів (двигунів) гіростабілізованих платформ в лабораторних умовах.

1. **Підстава для розробки**

Підставою для розробки є завдання на дипломний проект згідно наказу по НТУУ «КПІ» №.1008-с від 23.03.2018 р.

1. **Мета і призначення розробки**

Блок керування призначений для використання в складі стенду повної перевірки якості виготовлення і працездатності прямих приводів гіростабілізованих платформ в процесі виробництва.

1. **Джерело розробки**

Пульт призначений для здійснення управління випробуваного двигуна і забезпечення вимірювань кутового положення і струму.  
Схема вимірювання кута повороту побудована на датчику TS2640N691E125, розташованого безпосередньо на осі випробуваного двигуна установки моделювання режимів роботи приводу і мікросхеми перетворювача кут / код пульта управління AD2S1210. Певне значення кута повороту осі датчика (резольвера) надходить по паралельній 16-ти розрядної шині на мікроконтролер, який використовується в алгоритмі роботи контролера блока керування і відображення значення кута на індикатор при виборі відповідного режиму роботи. В конструкції двигуна закладений датчик струму вихідний сигнал якого надходить на схему вимірювання побудовану на базі 16-ти розрядного аналого-цифрового перетворювача (АЦП). Функціонально схема керування випробуваним двигуном імітує роботу бортового контролера платформи, замикаючи зворотний зв'язок по положенню від датчика кута стенду відповідно режиму роботи. Інформація про струм і кут положення двигуна виводитися на індикатор пульта, а також забезпечується режим їх обмеження.

**Технічні характеристики одного з двигунів, що контролюється:**

* Загальник опір обмоток Rд = 9,3 Ом;
* Коефіцієнт моменту та коефіцієнт ЭДС Км = 1,5 Нм/А, Ке = 1,5 Вс/рад;
* Номінальний момент Мн = 1,7 Нм при номінальному струмі Iном = 1,13А;
* Максимальний момент Мmax = 11,0 Нм при максимальному струмі Imax = 7,2А

1. **Технічні вимоги**
   1. **Склад виробу й вимоги до пристрою, що розробляється.**

Блок керування складається з:

* FPGA, до складу якої входить центральний процесор NIOSII, а також синтезуються ШІМ-контролер, та АЦП-контролер;
* Драйвер датчика кута;
* АЦП;
* Диференційні драйвери;
* Індикатор;
* Клавіатура.

**Показники призначення.**

Вимірювальний блок в лабораторних умовах повинен забезпечувати вимірювання і контроль наступних величин і характеристик двигунів:

− миттєві і середні значення струмів, що протікають в обмотках двигунів, їх форму;

− кутове положення ротора двигуна;

− кутову швидкість ротора;

− момент, що розвивається двигуном при обертанні ротора із заданою швидкістю;

− пусковий момент, що розвивається двигунами при загальмованому роторі;

− перехідні процеси розгону / гальмування двигунів без інерційної маси, що імітує масу без корисного навантаження і з ним;

− форма і розмір напруги, що генерується обмотками двигуна при обертанні із заданою швидкістю від зовнішнього двигуна.

* 1. **Вимоги до надійності.**

Середній час напрацювання на відмову повинен бути на менше 10000 год.

* 1. **Вимоги до технологічності.**

Орієнтовані на передові прийоми розробки виготовлення на підприємствах України.

* 1. **Вимоги до рівня уніфікації й стандартизації.**

Для виготовлення пристрою застосувати стандартні, уніфіковані деталі та вироби.

* 1. **Вимоги безпеки обслуговування.**

Керуватися загальними вимогами безпеки до апаратури низької напруги ГОСТ 12.2.007-75.

* 1. **Вимоги до складових частин виробу, сировини, вихідних й експлуатаційних матеріалів.**

Для виробництва пристрою повинні використовуватися матеріали імпортного виробництва, які можливо купити та замінити в Україні, або власна елементна база, яка розробляється в Україні.

* 1. **Умови експлуатації.**

Кліматичне виконання модулю керування УХЛ.4.2 згідно ГОСТ 15150-69. Для експлуатації в лабораторних, капітальних житлових та інших подібного типу приміщеннях.

* 1. **Вимоги до транспортування і зберігання.**

Група умов зберігання Л1 по ГОСТ 15150-69. Зберігати в зачинених, опалювальних та вентильованих приміщеннях, в яких забезпечуються наступні умови: температура повітря +5…+400С, відносна вологість повітря 60% при 200С (середньорічне значення), атмосферний тиск 84…106кПа.

Транспортувати автомобільним, залізничним або авіаційним видами транспорту в спеціальній транспортній тарі.

* 1. **Додаткові технічні вимоги.**

Технічні характеристики:

|  |  |
| --- | --- |
| Принцип дії |  |
| Інтерфейс зв’язку з системою | Індикація режиму роботи на ЖКІ-дисплеї, задання режиму роботи через клавіатуру |
| Габаритні розміри | 120х100 см. |
| Маса | Не більше 200 г. |
| Живлення | 5 В, 12 В. |
| Режими роботи | Основний (вимірювання); налаштування. |
| Діапазон виміру кутів | ±60о |

1. **Результати роботи**
   1. Результати даної роботи можуть бути використані як вихідна документація по створенню прототипу пристрою, його програмування, налагодження;
   2. Дана робота (звітна документація) після виконання надається на кафедру КЕОА для подальшого захисту й зберігання як навчальної документації.
2. **Робота повинна містити в собі документи**

* Пояснювальну записку (формату А4, до 80 аркушів)
* Схеми електричні принципові та переліки елементів (формату А1, А3, А4 відповідно)
* Складальні креслення та специфікації (формату А1, А4 відповідно)
* Креслення друкованих плат (формату А1)
* Додатки (формату А1-А4)

1. **Порядок розгляду й приймання роботи**

Порядок розгляду й приймання роботи на загальних умовах, прийнятих на кафедрі КЕОА. Рецензування й прийняття роботи комісією на загальних умовах. У процесі виконання роботи проміжні звіти надаються комісії не рідше 1 раз у тиждень на загальних умовах.

1. **Економічні показники**

В умовах даного проекту не розглядаються.

1. **Етапи розробки**

|  |  |  |
| --- | --- | --- |
| №  п/п | Назва етапів виконання дипломного  проекту | Час виконання етапів проекту |
| 1 | Аналіз технічного завдання |  |
| 2 | Вибір елементної бази та друкованої плати |  |
| 3 | Схемо-технічне проектування |  |
| 4 | Виконання креслення схеми електричної принципової |  |
| 5 | Конструкторсько-технологічні розрахунки |  |
| 6 | Електричний розрахунок друкованої плати |  |
| 7 | Проектування у Altium Designer |  |
| 8 | Моделювання |  |
| 9 | Виконання креслень друкованої плати та складального креслення друкованого вузла |  |
| 10 | Оформлення пояснювальної записки |  |

Додаток Б

Основна програма:

ACK2\_main.c:

**#include** "sys/alt\_stdio.h"

**#include** "alt\_types.h"

**#include** "altera\_avalon\_pio\_regs.h"

**#include** "system.h"

**#include** "ACK2\_main.h"

**#include** "LCD.h"

**#include** "resolver.h"

**#include** "ANGLE\_SETTING.h"

alt\_16 Angle\_norm;

alt\_16 Current\_norm;

alt\_u8 setting\_mode;

**int** **main**()

{

**#ifdef** ALT\_DEBUG

alt\_putstr("Hello from Nios II!\n");

**#endif**

alt\_u8 fault\_reg;

alt\_16 resolver\_angle;

alt\_u8 k;

alt\_16 Current\_temp;

//инициализация

Resolver\_initialization();

LCD\_INITIALIZATION();

Angle\_norm=0;

Current\_norm=0;

k=0;

**while**(1)

{

fault\_reg= Resolver\_read\_byte\_config\_mode (RES\_ADDR\_FAULT\_REG);

//если регистр не выдает ошибку, то читаем угол с резольвера

**if** (fault\_reg == 0) //читаем угол с резольвера

resolver\_angle=Resolver\_read\_position\_in\_normal\_mode();

**if** (k==0)

{

//устанавливаем режим настройки угла

setting\_mode=3;

Angle\_norm=ANGLE\_SETTING(Angle\_norm);

//устанавливаем режим настройки тока

setting\_mode=2;

Current\_norm=CURRENT\_SETTING(Current\_norm);

k=1;

}

**else**

{

//формирование режима

setting\_mode=MODE\_FORMATION();

**switch** (setting\_mode)

{

//если выбран режим работа

**case** 0:

{

// выставляем выводы SAMPLE в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NSAMPLE\_BASE,0);

//задержка на 2 us (1 такт процессора 20 нс)после сброса SAMPLE до поднятия SAMPLE

Delay\_for\_microsec (2);

// выставляем выводы SAMPLE в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NSAMPLE\_BASE,1);

//задержка на 2 us (1 такт процессора 20 нс)после сброса SAMPLE до поднятия CS

Delay\_for\_microsec (2);

//чтение регистра ошибки

fault\_reg= Resolver\_read\_byte\_config\_mode (RES\_ADDR\_FAULT\_REG);

//если регистр не выдает ошибку, то читаем угол с резольвера

**if** (fault\_reg == 0) //читаем угол с резольвера

resolver\_angle=Resolver\_read\_position\_in\_normal\_mode();

**else**

Out\_ERROR\_RESOLVER();//Печатаем слово ERROR\_RESOLVER

//прочитать ток двигателя с АЦП

Current\_temp=IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_SDO\_ADC\_BASE);

//выдаем на PWM значение померянного тока(из режима настройки тока)

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_PWM\_IN\_BASE,Current\_norm);

//отобразить на ЖКИ режим работа

Out\_WORKING();

}

**break**;

//если выбран режим настройка тока

**case** 2:

CURRENT\_SETTING(Current\_norm);

**break**;

//если выбран режим настройка угл

**case** 3:

Angle\_norm=ANGLE\_SETTING(Angle\_norm);

**break**;

}

k=1;

}

**return** 0;

**getchar**();

}

**void** **Delay\_for\_microsec** (alt\_u32 amount\_us)

{

**int** i;

//дисассемблирование показало,что процессор на выполнение этого цикла тратит 9 тактов

// время одного такта CPU\_CYCLE\_DELAY\_nS =20нс, поэтому время выполнения цикла = 20\*9=190нс

//amount\_us-количество микросекунд задержки.Минимальная задержка 1 микросекунда.

// (1000\*MILI/(DELAY\_KOEFF))-пример задания аргумента функции , если необходима задержка

//в микросекундах, то (число\*MICRO/(DELAY\_KOEFF))

**for** (i=0;i<(amount\_us);i++);

}

ACK2\_main.h:

**#ifndef** ACK1\_MAIN\_H\_

**#define** ACK1\_MAIN\_H\_

**#define** MASK\_D1 0x1E // 11110

**#define** MASK\_D2 0x1D // 11101

**#define** MASK\_D3 0x1B // 11011

**#define** MASK\_D4 0x17 // 10111

**#define** MASK\_D5 0xF // 1111

**#define** E1\_E2MASK 0x3 // 0011

**#define** E3\_E4MASK 0xC // 1100

**extern** alt\_16 Angle\_norm;

**extern** alt\_16 Current\_norm;

**extern** alt\_u8 setting\_mode;

**void** **Delay\_for\_microsec** (alt\_u32 amount\_us);

#endif /\* ACK1\_MAIN\_H\_ \*/

Програма виводу інформації на індикатор:

LCD.h:

**#ifndef** LCD\_H\_

**#define** LCD\_H\_

**#define** LCD\_E\_MASK 4

**#define** LCD\_R\_NOTW\_MASK 2

**#define** LCD\_RS\_MASK 1

**#define** LCD\_BF\_MASK 0x80

**#define** LCD\_ADDRESS\_MASK 0x7f

**#define** LCD\_DIRECTION\_8BIT\_OUTPUT\_MASK 0xFF

**#define** LCD\_DIRECTION\_8BIT\_INTPUT\_MASK 0

**#define** LCD\_MINUS\_NUMBER\_MASK 0x8000

// время выполнения процессором одной команды

//выполняется для рассчета задержки между командами

**#define** CPU\_CYCLE\_DELAY\_nS 20

// коэффициент для определения задержки процедуры Delay\_for\_microsec

// в микросекундах

**#define** DELAY\_KOEFF (CPU\_CYCLE\_DELAY\_nS\*8)

//коэффициент для задания задержки в микросекундах

**#define** MICRO 1000

//коэффициент для задания задержки в мили секундах

**#define** MILI 1000000

//функция для организации задержки не менее 1 микросекунды

//определяем 180 градусов в 16-ричном коде

**#define** GRAD\_180 0x7FFF

//#define pi\_2 1.5707963267948966/0x7FFF

//коэффициент перевода значения угла в градусах в шестнадцатиричный код

// koef=0xFFFF/360=182=B6

**#define** KOEF\_TRANSFORM 0xB6

//Функция передачи инструкции

**void** **FUNCTION\_SET**(alt\_8 instruction);

**void** **Delay\_for\_microsec** (alt\_u32 amount\_us);

**void** **LCD\_INITIALIZATION**();

//---------------------------------------------------------------------------

//Функция вывода на экран символа,заданного в ASCII-16-тиричном коде

**void** **LCD\_print\_symbol**(alt\_u8 code);

//----------------------------------------------------------------------------

//Функция вывода на экран символа,в конкретную позицию соответствующего адресу

//----------------------------------------------------------------------------

**void** **LCD\_print\_symbol\_in\_position** (alt\_u8 code, alt\_u8 address);

//----------------------------------------------------------------------------

//Функция вывода на экран 16-ти разрядного числа, занимает 5 позиций на экране

//----------------------------------------------------------------------------

**void** **LCD\_print\_digital\_number\_on\_screen** (alt\_16 value,alt\_u8 address,alt\_u8 point\_position);

**void** **Out\_SET\_SW1**();

**void** **Out\_SET\_SW2**();

**void** **Out\_SET\_SW3**();

**void** **Out\_EROOR\_CONECT**();

**void** **Out\_EROOR\_RESOLVER**(**void**);

**void** **Out\_WORKING**(**void**);

**void** **Out\_ANGLE\_SETTING**(**void**);

**void** **Out\_CURRENT\_SETTING**(**void**);

**extern** alt\_16 Angle\_norm;

**extern** alt\_16 Current\_norm;

**extern** alt\_u8 setting\_mode;

**#endif** /\*LCD\_H\_\*/

LCD.c:

**#include** "sys/alt\_stdio.h"

**#include** "system.h"

**#include** "alt\_types.h"

**#include** "altera\_avalon\_pio\_regs.h"

**#include** "resolver.h"

**#include** "ACK2\_main.h"

**#include** "LCD.h"

//Функция передачи инструкции

**void** **FUNCTION\_SET**(alt\_8 instruction)

{

alt\_u8 i;

//Объявляем двунаправленный порт в режим вывода (все 8 выводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_LCD\_DATA\_BASE,LCD\_DIRECTION\_8BIT\_OUTPUT\_MASK);

//Записываем в порт данных 0х3с=111100-установка режима 2-линии и дисплей включен

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_DATA\_BASE,instruction);

//Записываем в порт RS и R/NOTW 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 );

//Устанавливаем вывод Е в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 |LCD\_E\_MASK );

//Задержка 200нс (1 "nop" =20ns)

**for** (i=0;i<10;i++) **asm** ("nop");

//обнуляем управляющие выводы RS\_R/NOTW/E

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 );

}

//Функция инициализации LCD индикатора

**void** **LCD\_INITIALIZATION**()

{

//Записываем в порт RS и R/NOTW E 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 );

//задержка на 40 милисекунды (требуется после включения питания)

Delay\_for\_microsec (40\*MILI/(DELAY\_KOEFF));

//инициализация ЖКИ-модуля

//передаем комманду 0х38(8-bit,1,5\*7 dot,2 lines)

FUNCTION\_SET(0x38);

////задержка на 5 милисекунд (busy flag not available)

Delay\_for\_microsec (5\*MILI/(DELAY\_KOEFF));

FUNCTION\_SET(0x38);//(8-bit,1,5\*7 dot,2 lines)

//задержка на 160us (busy flag not available)

Delay\_for\_microsec (2\*MILI/(DELAY\_KOEFF));

//Записываем в порт данных 0х0C=1100 DISPLAY ON, Cursor ON, BLINK OFF

FUNCTION\_SET(0x0C);

//задержка на 160us (busy flag not available)

Delay\_for\_microsec (2\*MILI/(DELAY\_KOEFF));

//Записываем в порт данных 0х01=0001 clear display

FUNCTION\_SET(0x01);

//задержка на 160us (busy flag not available)

Delay\_for\_microsec (2\*MILI/(DELAY\_KOEFF));

//Записываем в порт данных 0х06=0110 entry mode set, cursor moves to right

FUNCTION\_SET(0x06);

//Объявляем двунаправленный порт в режим ввода (все 8 выводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_LCD\_DATA\_BASE,0);

} //Конец функции инициализации индикатора

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Функция вывода на экран 16-ти разрядного числа, занимает 5 позиций на экране

ей передается число (но не больше 65535 ) которое нужно вывести на индикатор,

адрес-соответствующий позиции на экране

0- первый символ первой строчки,64-первый символ второй строчки,16-первый символ третьей строчки,

80-первый символ четвертой строчки).Point\_position-положение запятой (отсчитывается справа).Если

Point\_position=1, то с одним знаком после запятой, если 2- то с 2-мя.

Для того,чтобы напечатать с одним знаком после запятой-нужно его предварительно

умножить на 10, если с двумя знаками-то на 100-соответственно.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **LCD\_print\_digital\_number\_on\_screen** (alt\_16 value,alt\_u8 address,alt\_u8 point\_position)

{

alt\_u8 i;

alt\_u8 k;

i=0;

k=0;

//Определяем знак числа

**if** (value&LCD\_MINUS\_NUMBER\_MASK)

{

value=(~value +1);

LCD\_print\_symbol\_in\_position(0x2d,address);

address++;

}

**else**

{

LCD\_print\_symbol\_in\_position(0x20,address);

address++;

}

// если количество знаков,после запятой=5, то печатаем запятую

**if** (point\_position==5)

{

LCD\_print\_symbol\_in\_position(0x2e,address);

address++;

k=1;

}

//Определяем количество 10000 в числе

**while** (value>=10000)

{

k=1;

value = value-10000;

i++;

}

//если число меньше,чем 10000-печатаем пробел

**if** (k==0) LCD\_print\_symbol\_in\_position(0x20,address);

//если число больше,чем 10000-печатаем количество 10000 в этом числе

**else** LCD\_print\_symbol\_in\_position(i+0x30,address);

i=0;

// если количество знаков,после запятой=4, то печатаем запятую

**if** (point\_position==4)

{

LCD\_print\_symbol(0x2e);

k=1;

}

//Определяем количество 1000 в числе

**while** (value>=1000)

{

k=1;

value = value-1000;

i++;

}

//если число меньше,чем 1000 и мы не напечатали ни одного знака перед-печатаем пробел

**if** (k==0) LCD\_print\_symbol (0x20);

//если число больше,чем 1000-печатаем количество 1000 в этом числе

**else** LCD\_print\_symbol(i+0x30);

i=0;

// если число знаков,после запятой=3, то печатаем запятую

**if** (point\_position==3)

{

LCD\_print\_symbol(0x2e);

k=1;

}

//Определяем количество 100 в числе

**while** (value>=100)

{

k=1;

value = value-100;

i++;

}

//если число меньше,чем 100 и мы не напечатали ни одного знака перед-печатаем пробел

**if** (k==0) LCD\_print\_symbol (0x20);

//если число больше,чем 100-печатаем количество 100 в этом числе

**else** LCD\_print\_symbol(i+0x30);

// если количество знаков,после запятой=2, то печатаем запятую

**if** (point\_position==2)

{

LCD\_print\_symbol(0x2e);

k=1;

}

i=0;

//Определяем количество 10 в числе

**while** (value>=10)

{

k=1;

value = value-10;

i++;

}

//если число меньше,чем 10 и мы не напечатали ни одного знака перед-печатаем пробел

**if** (k==0) LCD\_print\_symbol (0x20);

//если число больше,чем 10-печатаем количество 10 в этом числе

**else** LCD\_print\_symbol(i+0x30);

// если количество знаков,после запятой=1, то печатаем запятую

**if** (point\_position==1)

{

LCD\_print\_symbol(0x2e);

k=1;

}

//печатаем количество единиц в этом числе

LCD\_print\_symbol(value+0x30);

} //конец функции

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Функция вывода на экран символа,в конкретную позицию соответствующего адресу

Мы ей передаем число,которое нужно напечатать и адрес-соответствующей позиции,

куда следует вывести это число.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **LCD\_print\_symbol\_in\_position** (alt\_u8 code, alt\_u8 address)

{

**int** A;

alt\_u8 BF;

alt\_u8 k;

**int** i;

//Считывание флажка занятости

//Установка выводов R/nW в значение 1, RS в значение 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0|LCD\_R\_NOTW\_MASK );

//Установка выводов E и R/NOTW в значение 11

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 |LCD\_E\_MASK|LCD\_R\_NOTW\_MASK );

//задержка 140нс

**for** (A=0;A<9;A++) **asm** ("nop");

**do**

{

BF = IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_DATA\_BASE);

k = BF & LCD\_BF\_MASK;

}

**while** (k);

//Записываем в порт RS 0 и R/NOTW 0 и Е 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 );

//Объявляем двунаправленный порт в режим вывода (все 8 выводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_LCD\_DATA\_BASE,LCD\_DIRECTION\_8BIT\_OUTPUT\_MASK);

//Записываем в порт данных aдрес позиции курсора

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_DATA\_BASE,LCD\_BF\_MASK | (address & LCD\_ADDRESS\_MASK));

//Устанавливаем вывод Е в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 |LCD\_E\_MASK );

//Задержка 140нс

**for** (A=0;A<9;A++) **asm** ("nop");

//обнуляем управляющие выводы RS\_R/NOTW/E

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 );

//Задержка 20 наносекунд,требуемая в инструкции

// Delay\_for\_microsec (40\*MICRO/(DELAY\_KOEFF));

**for** (i=0;i<3;i++) **asm** ("nop");

//Установка выводов двунаправленного порта в режим входа

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_LCD\_DATA\_BASE,0x00);

LCD\_print\_symbol(code);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Функция вывода на экран символа-соответствующего коду,передаваемого функции

код следует писать в кодах ASCII в 16-ричном масштабе (0-0x30,1-0x31).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **LCD\_print\_symbol**(alt\_u8 code)

{

**int** A;

alt\_u8 BF;

alt\_u8 k;

**int** i;

//Считывание флажка занятости

//Установка выводов R/nW в значение 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0|LCD\_R\_NOTW\_MASK );

//задержка 40нс

//Delay\_for\_microsec((1\*MICRO/DELAY\_KOEFF));

**for** (i=0;i<21;i++) **asm** ("nop");

//Установка выводов E и R/NOTW в значение 11

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 |LCD\_E\_MASK|LCD\_R\_NOTW\_MASK );

**for** (A=0;A<9;A++) **asm** ("nop");

**do**

{

BF = IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_DATA\_BASE);

k = BF & LCD\_BF\_MASK;

}

**while** (k);

//Задержка на 32 милисекунды (требуется после включения питания)

//Delay\_for\_microsec (1\*MILI/(DELAY\_KOEFF));

//Записываем в порт RS -1 и R/NOTW 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0|LCD\_RS\_MASK );

//Объявляем двунаправленный порт в режим вывода (все 8 выводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_LCD\_DATA\_BASE,LCD\_DIRECTION\_8BIT\_OUTPUT\_MASK);

//Записываем в порт данных ASC код символа,например, 0х30=число 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_DATA\_BASE,code);

//Задержка 100нс

**for** (A=0;A<9;A++) **asm** ("nop");

//Устанавливаем вывод Е в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 |LCD\_E\_MASK | LCD\_RS\_MASK );

//Задержка 140нс

**for** (A=0;A<9;A++) **asm** ("nop");

//обнуляем управляющие выводы RS\_R/NOTW/E

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LCD\_CONTROL\_BASE,0 );

//Задержка 40 микросекунд,требуемая в инструкции

// Delay\_for\_microsec (40\*MICRO/(DELAY\_KOEFF));

**for** (i=0;i<9;i++) **asm** ("nop");

//Установка выводов двунаправленного порта в режим входа

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_LCD\_DATA\_BASE,0x00);

}//конец функции

**void** **Out\_SET\_SW1**(**void**) //Печатаем слово SET\_SW1

{

LCD\_print\_symbol\_in\_position (0x53,0x47);//S

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x5f);//\_

LCD\_print\_symbol(0x53);//S

LCD\_print\_symbol(0x57);//W

LCD\_print\_symbol(0x31);//1

}

**void** **Out\_SET\_SW2**(**void**) //Печатаем слово SET\_SW2

{

LCD\_print\_symbol\_in\_position (0x53,0x47);//S

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x5f);//\_

LCD\_print\_symbol(0x53);//S

LCD\_print\_symbol(0x57);//W

LCD\_print\_symbol(0x32);//2

}

**void** **Out\_SET\_SW3**(**void**) //Печатаем слово SET\_SW3

{

LCD\_print\_symbol\_in\_position (0x53,0x47);//S

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x5f);//\_

LCD\_print\_symbol(0x53);//S

LCD\_print\_symbol(0x57);//W

LCD\_print\_symbol(0x33);//3

}

**void** **Out\_EROOR\_CONECT**(**void**) //Печатаем слово EROOR\_CONNECT

{

LCD\_print\_symbol\_in\_position (0x45,0x44);//E

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x4f);//O

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x5f);//\_

LCD\_print\_symbol(0x43);//C

LCD\_print\_symbol(0x4f);//O

LCD\_print\_symbol(0x4e);//N

LCD\_print\_symbol(0x4e);//N

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x43);//C

LCD\_print\_symbol(0x54);//T

}

**void** **Out\_EROOR\_RESOLVER**(**void**) //Печатаем слово ERROR\_RESOLVER

{

LCD\_print\_symbol\_in\_position (0x45,0x44);//E

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x4f);//O

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x5f);//\_

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x53);//S

LCD\_print\_symbol(0x4f);//O

LCD\_print\_symbol(0x4c);//L

LCD\_print\_symbol(0x56);//V

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x52);//R

}

**void** **Out\_WORKING**(**void**) //Печатаем слово РЕЖИМ РАБОТА

{

LCD\_print\_symbol\_in\_position (0x45,0x57);//W

LCD\_print\_symbol(0x4f);//O

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x4b);//K

LCD\_print\_symbol(0x49);//I

LCD\_print\_symbol(0x4e);//N

LCD\_print\_symbol(0x49);//I

LCD\_print\_symbol(0x47);//G

}

**void** **Out\_ANGLE\_SETTING**(**void**) //Печатаем слово НАСТРОЙКА УГЛА

{

LCD\_print\_symbol\_in\_position (0x45,0x41);//A

LCD\_print\_symbol(0x4e);//N

LCD\_print\_symbol(0x47);//G

LCD\_print\_symbol(0x4c);//L

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x5f);//\_

LCD\_print\_symbol(0x53);//S

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x49);//I

LCD\_print\_symbol(0x4e);//N

LCD\_print\_symbol(0x47);//G

}

**void** **Out\_CURRENT\_SETTING**(**void**) //Печатаем слово НАСТРОЙКА ТОКА

{

LCD\_print\_symbol\_in\_position (0x45,0x43);//C

LCD\_print\_symbol(0x55);//U

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x52);//R

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x4e);//N

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x5f);//\_

LCD\_print\_symbol(0x53);//S

LCD\_print\_symbol(0x45);//E

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x54);//T

LCD\_print\_symbol(0x49);//I

LCD\_print\_symbol(0x4e);//N

LCD\_print\_symbol(0x47);//G

}

Програма керування драйвером кута:

resolver.h:

**#ifndef** RESOLVER\_H\_

**#define** RESOLVER\_H\_

// маска вывода CS0 порта pio

**#define** RESOLVER\_CS0\_MASK 1

// маска вывода CS1 порта pio

**#define** RESOLVER\_CS1\_MASK 2

// маска вывода SAMPLE порта pio

**#define** RESOLVER\_SAMPLE\_MASK 4

// маска выбора первого резольвера порта pio

**#define** RESOLVER\_1RES\_MASK 1

// маска выбора второго резольвера порта pio

**#define** RESOLVER\_2RES\_MASK 2

// маска выбора первого EEPROM порта pio

**#define** RESOLVER\_CS\_EPROM1\_MASK 2

// маска выбора второго EEPROM порта pio

**#define** RESOLVER\_CS\_EPROM2\_MASK 8

// маска не выбирающая ничего из порта pio

**#define** RESOLVER\_CS\_NOTHING 7 //(0|RESOLVER\_CS0\_MASK|RESOLVER\_CS1\_MASK|RESOLVER\_CS2\_MASK|RESOLVER\_CS3\_MASK)

// маска,включающая мультиплексор для передачи первого резольвера RESOLVER0

**#define** RESOLVER\_MUX\_RESOLVER0\_MASK 0

// маска,включающая мультиплексор для передачи второго резольвера RESOLVER1

**#define** RESOLVER\_MUX\_RESOLVER1\_MASK 1

**#define** RESOLVER\_HIGH\_BIT 0x8000

//маска, выставляющая А1А0 в режим конфигурации

**#define** A1\_A0\_MASK\_CONFIGURATION\_MODE 3 //00011

**#define** A1\_A0\_MASK\_NORMAL\_MODE\_POSITION 0

**#define** A1\_A0\_MASK\_NORMAL\_MODE\_VELOCITY 1

**#define** ANGLE\_DIRECTION\_16BIT\_OUTPUT\_MASK 0xFFFF

**#define** ANGLE\_DIRECTION\_16BIT\_INTPUT\_MASK 0

// адреса регистров резольвера ADS1210

**#define** RES\_ADDR\_POSITION\_REG\_H 0x80

**#define** RES\_ADDR\_POSITION\_REG\_L 0x81

**#define** RES\_ADDR\_VELOCITY\_REG\_H 0x82

**#define** RES\_ADDR\_VELOCITY\_REG\_L 0x83

**#define** RES\_ADDR\_LOS\_THRESHOLD\_REG 0x88

**#define** RES\_ADDR\_DOS\_OVERRANGE\_THRESHOLD\_REG 0x89

**#define** RES\_ADDR\_DOS\_MISMATCH\_THRESHOLD\_REG 0x8A

**#define** RES\_ADDR\_DOS\_RESET\_MAX\_THRESHOLD\_REG 0x8B

**#define** RES\_ADDR\_DOS\_RESET\_MIN\_THRESHOLD\_REG 0x8C

**#define** RES\_ADDR\_LOT\_HIGH\_THRESHOLD\_REG 0x8D

**#define** RES\_ADDR\_LOT\_LOW\_THRESHOLD\_REG 0x8E

**#define** RES\_ADDR\_EXCITATION\_FREQUENCY\_REG 0x91

**#define** RES\_ADDR\_CONTROL\_REG 0x92

**#define** RES\_ADDR\_SOFT\_RESET\_REG 0xF0

**#define** RES\_ADDR\_FAULT\_REG 0xFF

//Функция, записывающая в порт resolver байт

// данных-8 бит для режима конфигурации

**void** **Resolver\_write\_byte\_config\_mode** (alt\_u8 address,alt\_u8 data);

//Функция, считывающая из порта spi resolver байт

// данных-8 бит для режима конфигурации

alt\_u8 **Resolver\_read\_byte\_config\_mode** (alt\_u8 address);

//функция, инициализации и конфигурации резольверов

alt\_u8 **Resolver\_initialization**();

alt\_16 **Resolver\_read\_position\_in\_normal\_mode**();

**extern** alt\_16 Angle\_norm;

**extern** alt\_16 Current\_norm;

**extern** alt\_u8 setting\_mode;

#endif /\*RESOLVER\_H\_\*/

resolver.с:

**#include** "sys/alt\_stdio.h"

**#include** "altera\_avalon\_pio\_regs.h"

**#include** "system.h"

**#include** "alt\_types.h"

**#include** "resolver.h"

**#include** "ACK2\_main.h"

//------------------------------------------------------------

//Функция, записывающая байт

// данных-8 бит для режима конфигурации

//------------------------------------------------------------

**void** **Resolver\_write\_byte\_config\_mode** (alt\_u8 address,alt\_u8 data)

{

alt\_u8 i;

// выставляем выводы А1 А0 в режим конфигурации

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_A1\_A0\_BASE,A1\_A0\_MASK\_CONFIGURATION\_MODE);

//Объявляем двунаправленный порт в режим вывода (все 8 выводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_ANGLE\_BASE,ANGLE\_DIRECTION\_16BIT\_OUTPUT\_MASK);

// выставляем адрес регистра 8 бит в порт

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_ANGLE\_BASE,address);

// выставляем вывод CS в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,0);

// выставляем вывод nRD в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NRD\_BASE,1);

// выставляем вывод nWR в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,0);

//задержка на 22 нс (1 такт процессора 20 нс)

**for**(i=1;i<=2;i++) **asm**("nop");

// выставляем вывод nWR в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,1);

// выставляем вывод CS в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,1);

//задержка на 10 нс

**for**(i=1;i<=2;i++) **asm**("nop");

// выставляем вывод CS в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,0);

// выставляем данные 8 бит в порт

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_ANGLE\_BASE,data);

// выставляем вывод nWR в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,0);

// задержка 240 нс между записью адреса и записью данных

Delay\_for\_microsec (10);

// выставляем вывод nWR в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,1);

// выставляем вывод CS в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,1);

//задержка на 620 нс

Delay\_for\_microsec (31);

} // конец Resolver\_write\_byte\_config\_mode

//------------------------------------------------------------

//Функция, считывающая из порта resolver байт

// данных-8 бит для режима конфигурации

//------------------------------------------------------------

alt\_u8 **Resolver\_read\_byte\_config\_mode** (alt\_u8 address)

{

alt\_u8 i;

// Запись адреса в порт

// выставляем выводы А1 А0 в режим конфигурации

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_A1\_A0\_BASE,A1\_A0\_MASK\_CONFIGURATION\_MODE);

//Объявляем двунаправленный порт в режим вывода (все 8 выводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_ANGLE\_BASE,ANGLE\_DIRECTION\_16BIT\_OUTPUT\_MASK);

// выставляем адрес регистра 8 бит в порт

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_ANGLE\_BASE,address);

// выставляем вывод CS в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,0);

// выставляем вывод nRD в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NRD\_BASE,1);

// выставляем вывод nWR в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,0);

//задержка на 22 нс (1 такт процессора 20 нс)

**for**(i=1;i<=2;i++) **asm**("nop");

// выставляем вывод nWR в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,1);

//задержка на 10 нс

**for**(i=1;i<2;i++) **asm**("nop");

// выставляем вывод nRD в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NRD\_BASE,0);

//задержка на 10 нс

**for**(i=1;i<2;i++) **asm**("nop");

//Объявляем двунаправленный порт в режим ввода (все 16 вводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_ANGLE\_BASE,ANGLE\_DIRECTION\_16BIT\_INTPUT\_MASK);

// чтение данных из порта

alt\_u8 rdata = IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_ANGLE\_BASE);

// return rdata;

**return** rdata;

}// конец Resolver\_read\_byte\_config\_mode

//----------------------------------------------------------------

//Функция чтения из регистров преобразователя резольвера

//----------------------------------------------------------------

alt\_16 **Resolver\_read\_position\_in\_normal\_mode** ()

{

alt\_u8 i;

// выставляем выводы SAMPLE в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NSAMPLE\_BASE,1);

// выставляем вывод nRD в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NRD\_BASE,1);

//задержка на 42 нс (1 такт процессора 20 нс)

**for**(i=1;i<2;i++) **asm**("nop");

// выставляем выводы SAMPLE в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NSAMPLE\_BASE,0);

//задержка на 220 нс (1 такт процессора 600 нс)

Delay\_for\_microsec (32);

// выставляем вывод CS в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,0);

// выставляем выводы А1 А0 в режим чтения угла

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_A1\_A0\_BASE,A1\_A0\_MASK\_NORMAL\_MODE\_POSITION);

//задержка на 220 нс (1 такт процессора 20 нс)

Delay\_for\_microsec (2);

// выставляем вывод nWR в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,1);

// выставляем вывод nRD в 0

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NRD\_BASE,0);

//задержка на 42 нс (1 такт процессора 20 нс)

**for**(i=1;i<3;i++) **asm**("nop");

//Объявляем двунаправленный порт в режим ввода (все 16 вводов)

IOWR\_ALTERA\_AVALON\_PIO\_DIRECTION(PIO\_ANGLE\_BASE,ANGLE\_DIRECTION\_16BIT\_INTPUT\_MASK);

//читаем значение угла

alt\_u16 rdata =IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_ANGLE\_BASE);

alt\_16 position= rdata-32760;

//задержка на 100 нс (1 такт процессора 20 нс)

**for**(i=1;i<5;i++) **asm**("nop");

// выставляем вывод CS в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,1);

// выставляем вывод nRD в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NRD\_BASE,1);

//задержка на 220 нс (1 такт процессора 20 нс)

Delay\_for\_microsec (20);

// return(position);

**return** (position);

}

// конец функции Resolver\_read\_position\_in\_normal\_mode чтения из регистров преобразователь резольвера

//--------------------------------------------------------------------

//функция инициализации

alt\_u8 **Resolver\_initialization** ()

{

// выставляем выводы SAMPLE в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NSAMPLE\_BASE,1);

// // выставляем вывод nWR в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NWR\_BASE,1);

// выставляем вывод nRD в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NRD\_BASE,1);

// выставляем вывод CS в 1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NCS\_BASE,1);

//выставляем порт в режим паралельного чтения

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_RES\_NSOE\_BASE,1);

// IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_MUX\_BASE,0 | RESOLVER\_MUX\_RESOLVER0\_MASK);

Resolver\_write\_byte\_config\_mode (RES\_ADDR\_CONTROL\_REG, 0x0);

// инициализация control регистра резольвера 0x7F=111 1111 0x5F= 0101 1111 0xDF 1101 1111

Resolver\_write\_byte\_config\_mode (RES\_ADDR\_CONTROL\_REG, 0xDF);

// инициализация регистра EXCITATION\_FREQUENCY\_REG резольвера 0x21

// fclkin=10kHz(подана с ПЛИС),fexc=33=0x21-- fwc=35=0x23 fexc=10.6kHz3

Resolver\_write\_byte\_config\_mode ( RES\_ADDR\_EXCITATION\_FREQUENCY\_REG, 0x21);

**return** (1);

}

//конец функции инициализации Resolver\_initialization

Програма налаштування кута:

ANGLE\_SETTING.h:

**#ifndef** ANGLE\_SETTING\_H\_

**#define** ANGLE\_SETTING\_H\_

**#define** SW1\_MASK 1

**#define** SW2\_MASK 2 //10

**#define** SW3\_MASK 4 //100

**#define** SW4\_MASK 8 //1000

**#define** SW5\_MASK 16 //10000

**#define** SW6\_MASK 20 //100000

//функция настройки угла

alt\_16 **ANGLE\_SETTING**(alt\_16 k);

//функция настройки тока

alt\_16 **CURRENT\_SETTING**(alt\_16 k);

**extern** alt\_16 Angle\_norm;

**extern** alt\_16 Current\_norm;

**extern** alt\_u8 setting\_mode;

#endif /\* ANGLE\_SETTING\_H\_ \*/

ANGLE\_SETTING.c:

**#include** "sys/alt\_stdio.h"

**#include** "alt\_types.h"

**#include** <io.h>

**#include** "altera\_avalon\_pio\_regs.h"

**#include** "system.h"

**#include** "ACK2\_main.h"

**#include** "LCD.h"

**#include** "resolver.h"

**#include** "ANGLE\_SETTING.h"

//функция настройки угла

alt\_16 **ANGLE\_SETTING**(alt\_16 k)

{

alt\_u32 timer;

timer=0;

alt\_16 resolver\_angle;

alt\_u32 perem\_cycla;

perem\_cycla=0;

**if** (perem\_cycla==0)

{

//задержка 0,5с

**while**(timer==500000000)timer++;

//проверка SW\_5" (ждем удержание клавиши)

**if** (SW5\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x0)

{

Angle\_norm++;

//вывести на ЖКИ значение угла

/\* Выводим на ЖКИ-индикатор углы с устройства управления резольверами

\* (переведенный из 16-разрядного кода в угол

\* (код\*100(с двумя знаками после запятой)\*360/65535,в первую строчку с 10-ой позиции

\* с двумя знаками после запятой \*/

//temp = PIO\_2\_ANGLE\_BASE\*100\*360/65535;

LCD\_print\_digital\_number\_on\_screen (Angle\_norm\*100\*360/65535,9,2);

}//проверка SW\_6" (ждем удержание клавиши)

**else** **if** (SW6\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x0)

{

Angle\_norm=Angle\_norm-1;

LCD\_print\_digital\_number\_on\_screen (Angle\_norm\*100\*360/65535,9,2);

}

**if** (((SW5\_MASK||(IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)))==0x01)&&((SW6\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x01)))

perem\_cycla=1;

**else**

perem\_cycla=0;

}

**return** (Angle\_norm);

}

//конец функции alt\_16 ANGLE\_SETTING(alt\_16 k)

//функция настройки тока

alt\_16 **CURRENT\_SETTING**(alt\_16 k)

{

alt\_u32 timer;

timer=0;

alt\_u32 perem\_cycla;

perem\_cycla=0;

**if** (perem\_cycla==0)

{

//задержка 0,5с

**while**(timer==500000000) timer++;

//проверка SW\_5" (ждем удержание клавиши)

**if** (SW5\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x0)

{

Current\_norm++;

//вывести на ЖКИ значение угла

/\* Выводим на ЖКИ-индикатор ток

\* (переведенный из 16-разрядного кода в угол

\* (код\*100(с двумя знаками после запятой)\*360/65535,в первую строчку с 10-ой позиции

\* с двумя знаками после запятой \*/

//temp = PIO\_2\_ANGLE\_BASE\*100\*360/65535;

LCD\_print\_digital\_number\_on\_screen (Current\_norm\*100\*360/65535,9,2);

timer=0;

}//проверка SW\_6" (ждем удержание клавиши)

**else** **if** (SW6\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x0)

{

Current\_norm=Current\_norm-1;

LCD\_print\_digital\_number\_on\_screen (Current\_norm\*100\*360/65535,9,2);

timer=0;

}

**if** (((SW5\_MASK||(IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)))==0x01)&&((SW6\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x01)))

perem\_cycla=0;

**else**

perem\_cycla=1;

}

**return** (Current\_norm);

}

//конец функции alt\_16 CURRENT\_SETTING(alt\_16 k)

//функция настройки режима

alt\_u8 **MODE\_FORMATION**(**void**)

{

alt\_32 timer;

**if** (SW2\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x1)

{

**if**(SW1\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x0)

{

//установить режим работа setting\_mode=1

setting\_mode=1;

//включить диод LED1

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LED2\_LED1\_BASE,0x1);

//установить начальные значения

**while** (timer==25000000) //проверка SW\_1" (ждем удержание клавиши)

{

**if** (SW1\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x0)

timer=0;

**else**

timer++;

}

//отображение режима работа на ЖКИ дисплее

Out\_WORKING();

} }

**else**

{

//установить режим настройка setting\_mode=0

setting\_mode=0;

//выключить диод LED1 и включить диод LED2

IOWR\_ALTERA\_AVALON\_PIO\_DATA(PIO\_LED2\_LED1\_BASE,0x2);

**if** (SW3\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x1)

{

**if** (SW4\_MASK||IORD\_ALTERA\_AVALON\_PIO\_DATA(PIO\_KEYBOARD\_SW6\_SW1\_BASE)==0x0)

//установить режим "настройка тока"

setting\_mode=2;

}

**else**

//установить режим "настройка угла"

setting\_mode=3;

}

**return** (setting\_mode);

}

//конец функции alt\_u8 MODE\_FORMATION(void)

Програма синтезу ШІМ-контролера:

`timescale 1 ns/ 10 ps/\* Модуль формирования общего сброса\*/module REST\_Global( input wire CLK\_IN, input wire PLL\_OK, // Описание выходных сигналов схемы output reg n\_RESET); // объявление переменных reg [7:0] ST\_REST; //начальная установка всех регистров initial begin n\_RESET = 0; ST\_REST = 0; end // Счетчик задержки always @(posedge CLK\_IN or negedge PLL\_OK) begin if (!PLL\_OK) begin ST\_REST <= 8'b0; end else if (!n\_RESET) begin ST\_REST <= ST\_REST; end else begin ST\_REST <= ST\_REST +1 ; end end always @(posedge CLK\_IN or negedge PLL\_OK) begin if (!PLL\_OK) begin n\_RESET <= 0; end else if (ST\_REST > 224) begin n\_RESET <= 1; end else begin n\_RESET <= 0; end end endmodule //REST\_Global

Програма синтезу контролера АЦП:

module AD7687

(

//input wire BUSY,

input wire RESET,

input wire CLK,

input wire DATA\_IN,//выходы АЦП

output reg CNV,//входа АЦП

output wire SCK,

output reg [4:0] ST,

output reg [15:0] SHIFT\_REG\_REC,

output reg [15:0]DATA\_OUT\_ADC

);

reg enable;

//reg [1:0] ST\_1;

reg [2:0] state;

wire HOLD\_CLK,HOLD\_CLK2,HOLD\_CLK3,HOLD\_CLK1;

//ss mashine с состоянием S1,S2,S3,S4,S5,S6

parameter S0=3'b000, S1=3'b001, S2=3'b010, S3=3'b011, S4=3'b100, S5=3'b101, S6=3'b110;

//вводим Lcell , чтобы избавиться от гонок в SCK, который переключается по состоянию S5

LCELL HOLD\_1CLK (.in(CLK), .out(HOLD\_CLK));

LCELL HOLD\_2CLK (.in(HOLD\_CLK),.out(HOLD\_CLK1));

LCELL HOLD\_3CLK (.in(HOLD\_CLK1), .out(HOLD\_CLK2));

LCELL HOLD\_4CLK (.in(HOLD\_CLK2),.out(HOLD\_CLK3));

wire wireCLK;

initial

begin

ST <= 0;//присваиваем начальные значения регистрам

//ST\_1 <= 2'b0;

state<=3'b0;

enable<=0;

CNV<=0;

SHIFT\_REG\_REC<=0;

DATA\_OUT\_ADC<=0;

end

//описание счетчика ST для подсчета принятых SPI бит(16)

//счетчик обнуляется по !RESET

always @(posedge CLK or negedge RESET)

begin

if (!RESET)

ST<=1'b0;

else if (enable ==1'b0 || (state==S0))

ST<=1'b0;

else if (enable ==1'b1)

ST<=ST+1'b1;

end

assign wireCLK = (state==S5) ? (~HOLD\_CLK3) : 1'b1;

assign SCK = wireCLK;

//описание сдвигового регистра получателя

always@(negedge SCK)

begin

if(!RESET)

SHIFT\_REG\_REC<=0;

else if (state==S5)

begin

SHIFT\_REG\_REC[15:0]<= {SHIFT\_REG\_REC[14:0], DATA\_IN};

//SHIFT\_REG\_REC[0]<=DATA\_IN;

end

end

always@(posedge CLK)

begin

if (state==S6)

begin

DATA\_OUT\_ADC=SHIFT\_REG\_REC[15:0];//сохраняем померянное значение

end

end

//описание машины состояния устройства

always@(posedge CLK)

begin

if(!RESET)//исходное состояние -обнуляем счетчик и сдвиговый регистр приемника

state<=S0;

else

case(state)

S0:

state<=S1;

S1://если DATA\_IN=1 переходим в состояние S2, иначе возвращаемся в S0

begin

if(!DATA\_IN)

state<=S0;

else

state<=S2;

end

S2:

state<=S3;

S3://если ложное срабатывание DATA\_IN(0 после 1)- сидим в S3, иначе переход в S4

begin

if(!DATA\_IN)

state<=S3;

else

state<=S4;

end

S4:

state<=S5;

S5:

begin

if(ST<15)

state<=S5;//выдача SCK, считывание D15..D0

else

state<=S6;//возврат в состояние S0

end

S6:

begin

if (ST<30)

state<=S6;

else

state<=S0;

end

default: state<=S0;// значение по умолчанию

endcase

end

always@(posedge CLK) //разрешение на включение шифтового регистра

begin

if(state==S5 || state==S6)

enable<=1;

else

enable<=0;

end

//операции CNV в конечном автомате

always@(posedge CLK or negedge RESET)

begin

if(!RESET)

CNV<=0;

else if(state==S6)

CNV<=0;

else

CNV<=1;

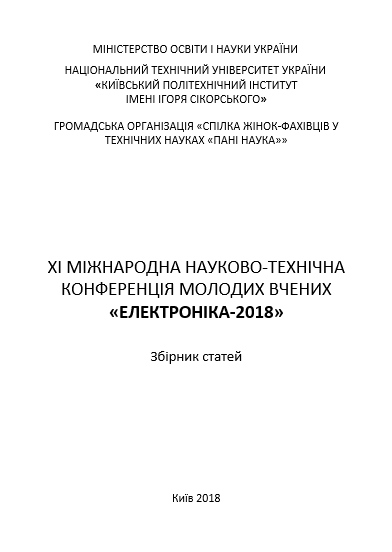
end

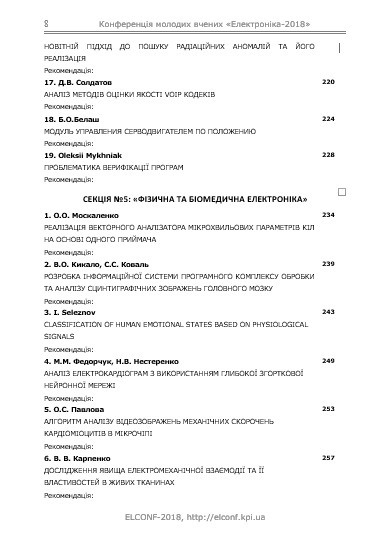
endmodule

**Додаток В.**

XІ МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ МОЛОДИХ ВЧЕНИХ **«ЕЛЕКТРОНІКА-2018»**

**Белаш Б.О. МОДУЛЬ УПРАВЛЕНИЯ СЕРВОДВИГАТЕЛЕМ ПО ПОЛОЖЕНИЮ**

****

****

УДК 621.3

**МОДУЛЬ УПРАВЛЕНИЯ СЕРВОДВИГАТЕЛЕМ ПО ПОЛОЖЕНИЮ**

Б.О.Белаш

КПИ им. Игоря Сикорского, Факультет электроники, Киев, Украина, e-mail: [bogdanbelash35@gmail.com](mailto:bogdanbelash35@gmail.com)

**Рассмотрена структура и принцип работы модуля управления. Модуль используется для лабораторных исследований работы серводвигателя в разных режимах и положениях. Модуль должен принимать все входные аналоговые сигналы с двигателя – обрабатывать их и формировать управляющие сигналы. Возможность переключения режимов осуществляется оператором с помощью клавиатуры. Информация о состоянии работы двигателя, режимах работы отображается на экране индикатора.**

**The structure and operating principle of the control module are considered. The module is used for laboratory studies of servo motor operation in different modes and positions. The module must receive all input analog signals from the motor - process them and generate control signals. The ability to switch modes is performed by the operator using the keyboard. Information about the status of the engine, modes of operation is displayed on the indicator screen.**

**Ключевые слова: пульт, серводвигатель, FPGA, датчик, АЦП, ШИМ**

**Key words: remote controller, servomotor, FPGA, sensor, ADC, PWM**

**Вступление**

Для корректной работы высокотехнологичных узлов и агрегатов каждому механизму требуется электронный блок управления. Для испытания конкретного серводвигателя необходимо разработать модуль, который должен обеспечивать измерение и контроль следующих величин в лабораторных условиях:

− мгновенное и среднее значение тока, протекающего в обмотках двигателя, его форму.

− угловое положения ротора двигателя.

− угловую скорость ротора.

− момент, развиваемый двигателем при вращении ротора с заданной скоростью.

− пусковой момент двигателя.

− переходные процессы разгона / торможения двигателя без инерционной массы и с ней.

− форма и величина генерируемого обмотками двигателя напряжения при вращении с заданной скоростью.

**Структура устройства**

В состав модуля управления входит цифровой контроллер. Он реализован на FPGA с применением встроенного процессорного ядра Nios II. Дополнительно, к процессорному ядру подключаются необходимые для формирования управляющих сигналов контроллер АЦП и ШИМ контроллер. Эти устройства описаны на языке HDL и синтезируются в FPGA архитектуре. Кроме цифрового контроллера, в состав модуля входят драйвер датчика угла, АЦП, индикатор и кнопки управления (клавиатура). Сигналы управления с контроллеров АЦП и ШИМ передаются дифференциальными парами для повышения помехоустойчивости.

Модуль управления взаимодействует с серводвигателем по каналу RS-422.

Испытуемая конструкция состоит из серводвигателя, датчика угла, датчика крутящего момента. Базовым элементом системы измерения крутящего момента является датчик крутящего момента T20WN.

Структура модуля управления представлена на Рис. 1. Блок серводвигателя, модуль

|  |
| --- |
|  |
| Рис. 1 Структура модуля управления |

**Принцип работы устройства**

Модуль предназначен для формирования управляющих сигналов для испытуемого двигателя, измерения углового положения ротора и тока двигателя.

В схеме измерения угла поворота ротора применяется датчик TS2640N691E125, расположенный непосредственно на оси испытуемого двигателя и микросхема преобразователя угол / код AD2S1210 в модуле управления. Измеренное значение угла поворота оси ротора двигателя поступает по параллельной 16-ти разрядной шине на цифровой контроллер. Это значение отображается на индикаторе и используется в алгоритме формирования управляющих сигналов на серводвигатель (при выборе соответствующего режима работы). Конструктивно, в двигателе применяется датчик тока, выходной сигнал которого поступает на схему измерения, представляющую собой 16-ти разрядный аналого-цифровой преобразователь (АЦП).

Схема АЦП изображена на Рис. 2.

Для управления серводвигателем цифровой контроллер формирует и выдает следующие сигналы управления: PWM, DIR, DIS. PWM – это широтноимпульсный модулированный сигнал, управляющий скоростью вращения двигателя. DIR – сигнал выбора направления вращения. DIS – сигнал разрешения работы двигателя.

Функционально, модуль управления двигателем имитирует работу бортового контроллера платформы, замыкая обратную связь по положению от датчика угла поворота ротора в соответствии с заданным режимом работы. Информация о токе и угле положения двигателя выводится на индикатор модуля. Дополнительно, обеспечивается режим ограничения тока и угла поворота.

Рис. 2 Схема аналого-цифрового преобразователя

Схема индикации обеспечивает визуализацию заданного режима работы и значение следующих параметров:

− максимально допустимого угла поворота ротора;

− максимально допустимого значения величины тока двигателя;

− номинального значения тока двигателя;

− текущего значения угла поворота ротора;

− текущего значения тока двигателя.

Алгоритм обработки сигналов обратной связи и формирования сигналов управления двигателем реализован на языке С.

**Выводы**

Целью данной работы было создание модуля управления серводвигателем по положению. В результате было спроектировано устройство, которое принимает и обрабатывает аналоговые сигналы с датчиков двигателя и формирует управляющие сигналы. Основным вычислительным элементом модуля является встроенный процессор Nios II, который обрабатывает только цифровые сигналы. Для обработки аналоговых сигналов в структуру модуля были добавлены драйвер датчика угла и АЦП. Таким образом, в зависимости от текущих значений угла поворота, центральный процессор формирует цифровые управляющие сигналы. ШИМ контроллер, получая эти сигналы, генерирует соответствующие управляющие аналоговые сигналы. Для повышения помехоустойчивости, было принято решение использовать для передачи выходных аналоговых сигналов дифференциальные пары. Проведенная работа и спроектированное устройство подтверждают оптимальность принятых схемотехнических и конструкторских решений, а также перспективность для дальнейших исследований.

**Литература**

1. Гусев Н.В., Букреев В.Г. Системы цифрового управления многокоординатными следящими электроприводами: учебное пособие – Томск: Издательство Томского политехнического университета, 2007. - 213с.
2. Бабаков Н.А., Воронов А.А., Воронова А.А., Теория автоматического управления: Учебник для вузов по спец. «Автоматика и телемеханика». В 2 ч. Ч.1. Теория линейных систем автоматического управления. – М.: Высшая школа, 1986. – 367с.
3. Бабаков Н.А., Воронов А.А., Воронова А.А., Теория автоматического управления: Учебник для вузов по спец. «Автоматика и телемеханика». В 2 ч. Ч.2. Теория нелинейных и специальных систем автоматического управления. – М.: Высшая школа, 1986. – 504с.
4. Управление электродвигателями [электронный ресурс] – Режим доступа: <http://www.symmetron.ua/files/d_3_Motor-Control_ru.pdf>

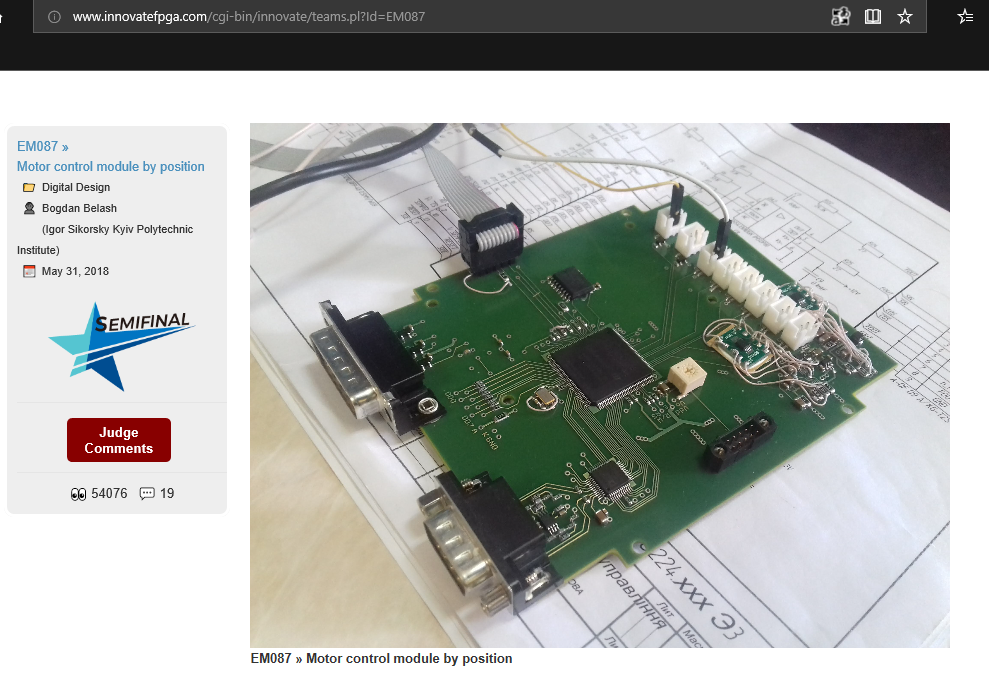
**Рекомендовано к публикации:** ст. преподаватель Антонюк А.И.

КПИ им. Игоря Сикорского, Факультет электроники, кафедра КЭВА, Киев, Украина

**Додаток Г**

**IntelFPGA: Innovate FPGA 2018**

**Bogdan Belash. Motor control module by position.** **<http://www.innovatefpga.com/cgi-bin/innovate/teams.pl?Id=EM087>**

****

Motor control module by position. This module is designed to control the DC motorby position. The field of application is high-precision electric drives. The task is to provide precise control and stopping of the engine at a given position of the motor rotor. The field of application is automotive (in particular electric vehicles), precise positioning systems.

The motor control circuit provides the PWM control signal (speed control of the signal), the signal that determines the direction of rotation of the DID. Additional signals used in the engine control process are the motor winding current, the power voltage, and the temperature of the motor windings.

The purpose of this work is the development of an engine control algorithm implemented on FPGAs in order to provide sufficient accuracy and speed of working out the given engine position.