**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО ”**
Факультет електроніки
Кафедра електронної інженерії

До захисту допущено
**Завідувач кафедри**
_____ В. І. Тимофєєв
“14”_____06_____ 20 21 р.

# Дипломна робота

освітнього рівня «бакалавр»
за напрямом 6.050801 мікро- та наноелектроніка

на тему «Енергозберігаюча система контролю електричних сигналів на основі мікроконтролера з сегнетоелектричною оперативною пам’яттю»

Виконав студент IV курсу, групи ДМ-51
Реутов Сергій Олександрович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник проф. каф. ЕІ, д.т.н., доцент. Ю.В. Прокопенко _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент проф. кафедри елекронних пристроїв та систем, д.т.н., І.В.Мельник
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2021 року

**Національний технічний університет України**
**“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет електроніки
Кафедра електронної інженерії
Освітній рівень «бакалавр»
за напрямом 6.050801 мікро- та наноелектроніка

**ЗАТВЕРДЖУЮ**
**Завідувач кафедри**
_____ В. І. Тимофєєв
“ 14 ” 06 20 21 р.

# З А В Д А Н Н Я
## НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Реутов Сергій Олександрович
(прізвище, ім’я, по батькові)

1. Тема роботи Енергозберігаюча система контролю електричних сигналів на основі мікроконтролера з сегнетоелектричною оперативною пам’яттю

керівник роботи Прокопенко Ю.В., проф. каф. ЕІ, д.т.н., доцент.
( прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від “___” _____20__ року №_____
2. Строк подання студентом роботи 12.06.21
3. Вихідні дані до роботи мікроконтролер з сегнетоелектричною оперативною пам’яттю MSP-EXP430-FR2355, Code Composer Studio v.11.2
4. Зміст дипломної роботи (перелік питань, які потрібно розробити)
Розробка програмного забезпечення до мікроконтроллера з сегнетеелектричною оперативною пам’яттю MSP-EXP430-FR2355: Сенсор освітленності, сенсор газу, сенсор диму, Smart Analog Combo
5. Перелік графічного (ілюстративного) матеріалу (із зазначенням обов’язкових креслень, плакатів, презентацій тощо)
Презентація

6. Консультанти розділів роботи

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

7.Дата видачі завдання___22.04.2021_____

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломної роботи | Строк виконання етапів роботи | Приміт ка |
|---|---|---|---|
| 1 | Збір матеріалів для роботи. Аналіз науково-технічної літератури. | 20.05.2021 -27.05.2021 | |
| 2 | Ознайомлення з методами оперування вивчаємої системи | 27.05.2021 - 1.06.2021 | |
| 3 | Створення, обробка та запис програми до розділу сегнетоелектричної оперативної пам'яті | 01.06.2021 -04.06.2021 | |
| 4 | Написання та оформлення теоретичної частини | 04.06.2021- 08.06.2021 | |
| 5 | Оформлення пояснювальної записки та презентації. | 08.06.2021 -12.06.2021 | |

**Студент** **Реутов Сергій Олександрович**_____ _____
( підпис ) (прізвище та ініціали)

**Керівник роботи** **Прокопенко Юрій Васильович**_____ _____
( підпис ) (прізвище та ініціали)

# РЕФЕРАТ

Дипломна робота: 47 с., 15 рис., 2 дод., 10 джерел.

FRAM, LIGHT-SENSOR, ПРОЕКТУВАННЯ У СЕРЕДОВИЩІ CSS

Мета роботи – огляд компонентів та функціональних можливостей мікроконтролера MSP430FR2355, аналіз його характеристик та застосування, розробка елементів програмного забезпечення для контролю електричних сигналів із забезпеченням низького рівня потужності, що споживається мікроконтролером.

Програмне забезпечення, розроблено у середовищі Code Composer Studio, використовуючи алгоритмічні мови C та Assembler.

У першому розділі наведено порівняння структури та характеристик сегнетоелектричної пам’яті з іншими типами оперативної пам’яті. Другий розділ присвячено огляду модулів мікроконтролера MSP430FR2355, та розглянуто основні відомості щодо середовища розробки та робочого стенду. У третьому розділі розглянуто приклади реалізації схем аналізу електричних сигналів з сенсорів.

Робота виконана згідно вимог нормативних документів КПІ ім. Ігоря Сікорського та чинних державних стандартів, в додатках вказані програмні коди.

# ABSTRACT

Diploma work: 47 p., 15 figures, 2 app., 10 references.

MSP430FR2355 - Mixed-Signal Microcontroller

The purpose of the work is a review of the components and functionality of the microcontroller MSP430FR2355, analysis of its characteristics and applications, development of software elements for the control of electrical signals with a low level of power consumed by the microcontroller.

Software developed in Code Composer Studio using the C and Assembler algorithmic languages.

The first section compares the structure and characteristics of ferroelectric memory with other types of RAM. The second section is devoted to an overview of the MSP430FR2355 microcontroller modules and discusses basic information about the development environment and the stand. The third section discusses examples of the implementation of schemes for the analysis of electrical signals from sensors.

The work was performed in accordance with the requirements of normative documents of KPI. Igor Sikorsky and current state standards, the appendices contain program codes.

MSP430FR2355 - Mixed-Signal Microcontroller

MSP430FR215x and MSP430FR235x microcontrollers (MCUs) are part of the MSP430™ MCU value line portfolio of ultra-low-power low-cost devices for sensing and measurement applications. MSP430FR235x MCUs integrate four configurable signal-chain modules called smart analog combos, each of which can be used as a 12-bit DAC or a configurable programmable-gain Op-Amp to meet the specific needs of a system while reducing the BOM and PCB size. The device also includes a 12-bit SAR

ADC and two comparators. The MSP430FR215x and MSP430FR235x MCUs all support an extended temperature range from –40° up to 105°C, so higher temperature industrial applications can benefit from the devices' FRAM data-logging capabilities. The extended temperature range allows developers to meet requirements of applications such as smoke detectors, sensor transmitters, and circuit breakers. The MSP430FR215x and MSP430FR235x MCUs feature a powerful 16-bit RISC CPU, 16-bit registers, and a constant generator that contribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows the device to wake up from low-power modes to active mode typically in less than 10 µs. The MSP430 ultra-low-power (ULP) FRAM microcontroller platform combines uniquely embedded FRAM and a holistic ultra-low-power system architecture, allowing system designers to increase performance while lowering energy consumption. FRAM technology combines the low-energy fast writes, flexibility, and endurance of RAM with the nonvolatile behavior of flash. MSP430FR215x and MSP430FR235x MCUs are supported by an extensive hardware and software ecosystem with reference designs and code examples to get your design started quickly. Development kits include the MSP-EXP430FR2355 LaunchPad™ development kit and the MSP-TS430PT48 48-pin target development board. TI also provides free MSP430Ware™ software, which is available as a component of Code Composer Studio™ IDE desktop and cloud versions within TI Resource Explorer. The MSP430 MCUs are also supported by extensive online collateral, training, and online support through the E2E™ support forums.

Ferroelectric RAM (FeRAM, F-RAM or FRAM) is a random-access memory similar in construction to DRAM but using a ferroelectric layer instead of a dielectric layer to achieve non-volatility. FeRAM is one of a growing number of alternative non-volatile random-access memory technologies that offer the same functionality as flash memory.

FeRAM's advantages over Flash include: lower power usage, faster write performance and a much greater maximum read/write endurance (about 1010 to 1014

cycles). FeRAMs have data retention times of more than 10 years at +85 °C (up to many decades at lower temperatures). Market disadvantages of FeRAM are much lower storage densities than flash devices, storage capacity limitations and higher cost. Like DRAM, FeRAM's read process is destructive, necessitating a write-after-read architecture

Conventional DRAM consists of a grid of small capacitors and their associated wiring and signaling transistors. Each storage element, a cell, consists of one capacitor and one transistor, a so-called "1T-1C" device. It is typically a type of MOS memory, fabricated using CMOS technology.[11] DRAM cells scale directly with the size of the semiconductor fabrication process being used to make it. For instance, on the 90 nm process used by most memory providers to make DDR2 DRAM, the cell size is 0.22 μm², which includes the capacitor, transistor, wiring, and some amount of "blank space" between the various parts — it appears 35% utilization is typical, leaving 65% of the space empty (for separation).

DRAM data is stored as the presence or lack of an electrical charge in the capacitor, with the lack of charge in general representing "0". Writing is accomplished by activating the associated control transistor, draining the cell to write a "0", or sending current into it from a supply line if the new value should be "1". Reading is similar in nature; the transistor is again activated, draining the charge to a sense amplifier. If a pulse of charge is noticed in the amplifier, the cell held a charge and thus reads "1"; the lack of such a pulse indicates a "0". Note that this process is destructive, once the cell has been read. If it did hold a "1," it must be re-charged to that value again. Since a cell loses its charge after some time due to leak currents, it must be actively refreshed at intervals.

The 1T-1C storage cell design in an FeRAM is similar in construction to the storage cell in widely used DRAM in that both cell types include one capacitor and one access transistor. In a DRAM cell capacitor, a linear dielectric is used, whereas in an

FeRAM cell capacitor the dielectric structure includes ferroelectric material, typically lead zirconate titanate (PZT).

A ferroelectric material has a nonlinear relationship between the applied electric field and the apparent stored charge. Specifically, the ferroelectric characteristic has the form of a hysteresis loop, which is very similar in shape to the hysteresis loop of ferromagnetic materials. The dielectric constant of a ferroelectric is typically much higher than that of a linear dielectric because of the effects of semi-permanent electric dipoles formed in the crystal structure of the ferroelectric material. When an external electric field is applied across a dielectric, the dipoles tend to align themselves with the field direction, produced by small shifts in the positions of atoms and shifts in the distributions of electronic charge in the crystal structure. After the charge is removed, the dipoles retain their polarization state. Binary "0"s and "1"s are stored as one of two possible electric polarizations in each data storage cell. For example, in the figure a "1" is encoded using the negative remnant polarization "-Pr", and a "0" is encoded using the positive remnant polarization "+Pr".

In terms of operation, FeRAM is similar to DRAM. Writing is accomplished by applying a field across the ferroelectric layer by charging the plates on either side of it, forcing the atoms inside into the "up" or "down" orientation (depending on the polarity of the charge), thereby storing a "1" or "0". Reading, however, is somewhat different than in DRAM. The transistor forces the cell into a particular state, say "0". If the cell already held a "0", nothing will happen in the output lines. If the cell held a "1", the re-orientation of the atoms in the film will cause a brief pulse of current in the output as they push electrons out of the metal on the "down" side. The presence of this pulse means the cell held a "1". Since this process overwrites the cell, reading FeRAM is a destructive process, and requires the cell to be re-written.

In general, the operation of FeRAM is similar to ferrite core memory, one of the primary forms of computer memory in the 1960s. However, compared to core memory, FeRAM requires far less power to flip the state of the polarity and does so much faster.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ……10
ВСТУП……11
1 ОПЕРАТИВНА ПАМ’ЯТЬ, ВИДИ ПАМ’ЯТІ……12
1.1 Оперативна пам’ять……12
1.2 DRAM……..15
1.3 FRAM або FeRAM……..18
2 ОГЛЯД ХАРАКТЕРИСТИК, КОМПОНЕНТІВ, БАЗОВОГО ВИГЛЯДУ РОБОЧОЇ СТАНЦІЇ ТА ЗАГАЛЬНІ ТЕОРЕТИЧНІ ВІДОМОСТІ……24
2.1 Огляд робочого середовища CCS – Code Composer Studio……....24
2.2 Огляд робочого середовища CCS – Code Composer Studio……...25
3. ПРИКЛАДИ РЕАЛІЗАЦІЇ СХЕМ АНАЛІЗУ ЕЛЕКТРИЧНИХ СИГНАЛІВ З СЕНСОРІВ……30
3.1 Smart Analog Combo……30
3.2 SAC у якості детектору диму……31
3.3 SAC у якості детектору газу……..32
3.4 Сенсор освітленості……33
ВИСНОВКИ……...36

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ……37
Додаток А……...38
Додаток Б……40

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АЦП – Аналогово-цифровий перетворювач
ГІК – Графічний інтерфейс користувача
МК – Мікроконтролер
ADC – Analog-to-digital converter
AES – Advanced Encryption Standard
CCS – Code Composer Studio
DSP – Digital signal processor
GUI – Graphical user interface
LED – Light-emitting diod
RGB – Red, Green, Blue
SAC – Smart Analog Combo
SAR – Successive Approximation ADCs
TI – Texas Instruments

# ВСТУП

Стандартні мікроконтроллери це малогабаритні мікросхеми, що призначені в першу чергу для керування електронними приладами.

Включають в себе мікропроцесор, оперативну пам'ять(ОЗУ) – для виконання програм в режимі реального часу, постійну пам'ять(ПЗУ) – для запису коду та/або результатів його виконання, порти для вводу/виводу данних та інші блоки, що призначені під більш вузькі задачі, такі як: компаратори, датчики, лічильники, модулі ЦАП-АЦП та ін.

У нашому випадку це мікроконтроллер Texas Instruments ™ MSP430FR2355, що має на борту не один десяток вспоміжних блоків, проте основними розглянутими у даній роботі будуть:

- Модуль FRAM
- Модуль SAC
- Light sensor
- Smoke sensor

Практична частина роботи присвячена зверненню до сенгетоелекричного модуля памь'яті з записом на нього робочої програми та перевірки функціоналу.

# 1. ОПЕРАТИВНА ПАМ’ЯТЬ, ВИДИ ПАМ’ЯТІ

## 1.1.Оперативна пам’ять.

Оперативна пам’ять або RAM (ОЗП – оперативний запам’ятовуючий пристрій) – швидкодійна комп’ютерна пам’ять, що призначена для запису, читання та зберігання інформації у процесі її обробки обчислювальною технікою. Фізично, оперативна пам'ять в системі являє собою набір мікросхем або модулів (що містять мікросхеми), які зазвичай підключаються до системної плати.

В процесі роботи пам'ять виступає в якості тимчасового буфера (в ній зберігаються дані і запущені програми) між дисковими накопичувачами і процесором, завдяки значно більшій швидкості читання і запису даних.

Не слід путати ОПЗ із ПЗП(постійний запам’ятовуючий пристрій), оскільки це зовсім різні види пам'яті. Оперативна пам'ять (по типу є динамічною - Dynamic RAM), на відміну від постійної - енергозалежна, тобто для зберігання даних їй необхідна електроенергія, і при її відключенні (виключення комп'ютера) дані видаляються. Приклад незалежній пам'яті ПЗУ - флеш-пам'ять, в якій струм використовується лише для запису і читання, в той час як для самого зберігання даних джерело живлення не потрібен.

За своєю структурою пам'ять нагадує бджолині соти, тобто складається з осередків, кожна з яких призначена для зберігання  певного обсягу даних, як

правило, одного або чотирьох біт. Кожна осередок якої має свою унікальну «домашню» адресу, яка ділиться на два компонента - адреса горизонтального рядка (Row) і вертикального стовпчика (Column).

Осередки представляють собою конденсатори, здатні накопичувати електричний заряд. За допомогою спеціальних підсилювачів аналогові сигнали переводяться в цифрові, які в свою чергу утворюють дані.

Для передачі на мікросхему пам'яті адреси рядка служить сигнал, який називають RAS (Row Address Strobe), а для адреси стовпця - сигнал CAS (Column Address Strobe).

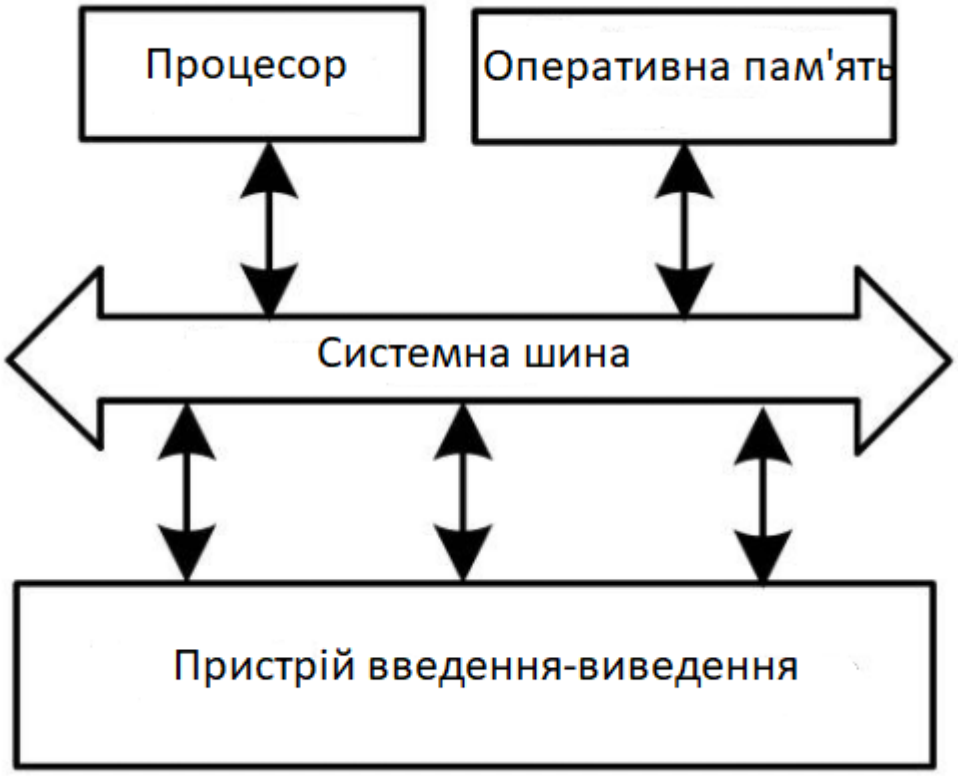


Рисунок 1.1 - Спрощена схема роботи RAM

Обмін даними між процесором і пам'яттю може відбуватися безпосередньо, але частіше все ж буває за участю кеш-пам'яті.

Кеш-пам'ять є місцем тимчасового зберігання найбільш часто запитуваної інформації і являє собою відносно невеликі ділянки швидкої локальної пам'яті. Її використання дозволяє значно зменшити час доставки інформації в регістри процесора, так як швидкодія зовнішніх носіїв (оперативної і дискової підсистем) набагато гірше процесорного. Як наслідок, зменшуються, а часто і повністю усуваються, вимушені простої процесора, що підвищує загальну продуктивність системи.

Оперативною пам'яттю управляє контролер, який знаходиться в чіпсеті материнської плати, а точніше в тій його частині, яка називається North Bridge (північний міст) - він забезпечує підключення CPU (процесора) до вузлів, що використовують високопродуктивні шини: ОЗУ, графічний контролер (див. Рис .1.2).

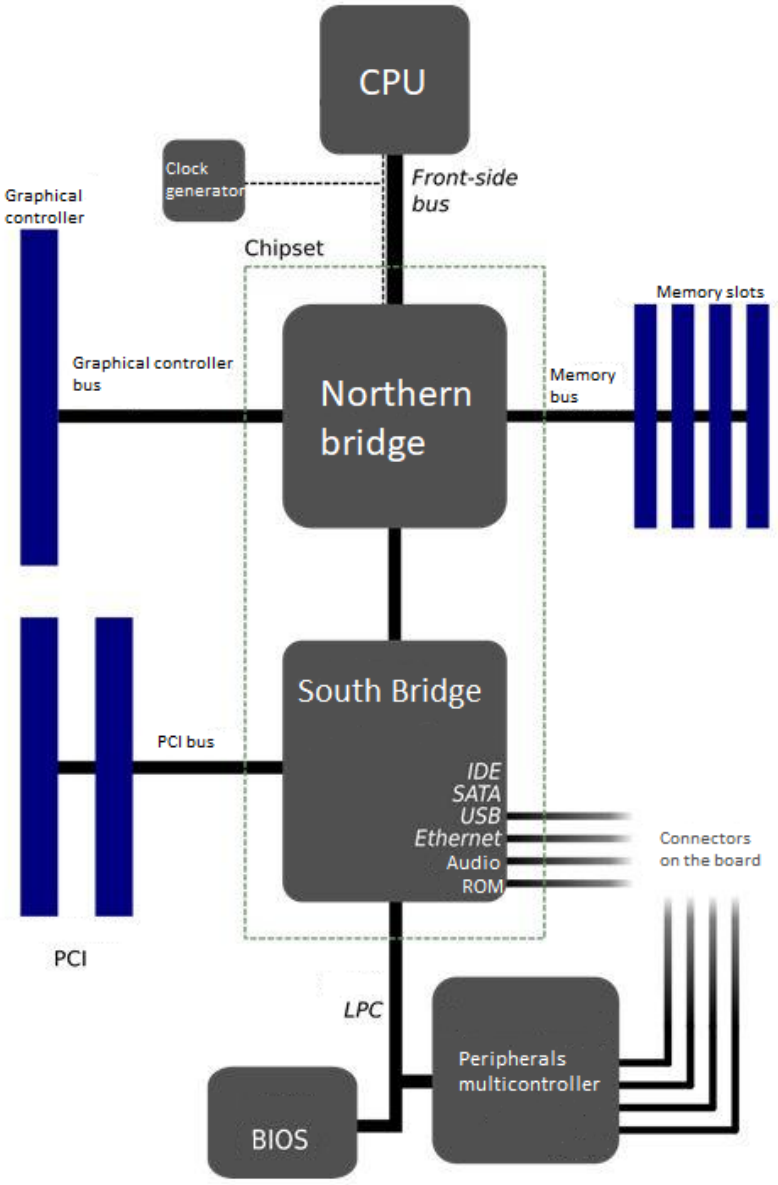


Рисунок 1.2 - Схема керування RAM

(Запозичено з [https://sonikelf.ru/operativnaya-pamyat-chto-k-chemu-i-zachem-podrobno-o-glavnom])

## 1.2.DRAM

DRAM (англ. Dynamic random access memory - динамічна пам'ять з довільним доступом) - тип комп'ютерної пам'яті, що відрізняється використанням

напівпровідникових матеріалів, енергозалежністю і можливістю доступу до даних, що зберігаються в довільних комірках пам'яті. Модулі пам'яті з пам'яттю такого типу широко використовуються в комп'ютерах в якості оперативних запам'ятовуючих пристроїв (ОЗП), також використовуються в якості пристроїв постійного зберігання інформації в системах, вимогливих до затримок.

Фізично DRAM складається з осередків, створених в напівпровідниковому матеріалі у вигляді ємності. Заряджена або розряджена ємність зберігає біт даних. Кожна осередок такої пам'яті має властивість розряджатися (через струмів витоку і ін.), тому їх постійно треба заряджати - звідси назва «динамічна» (динамічно заряджати). Сукупність комірок утворює умовний «прямокутник», що складається з певної кількості рядків і стовпців. Один такий «прямокутник» називається сторінкою, а сукупність сторінок називається банком. Весь набір осередків умовно ділиться на кілька областей.

Як пристрій (ЗУ) DRAM є модуль пам'яті будь-якого конструктивного виконання, що складається з друкованої плати, на якій розташовані мікросхеми пам'яті, і роз'єму, необхідного для підключення модуля до материнської плати.

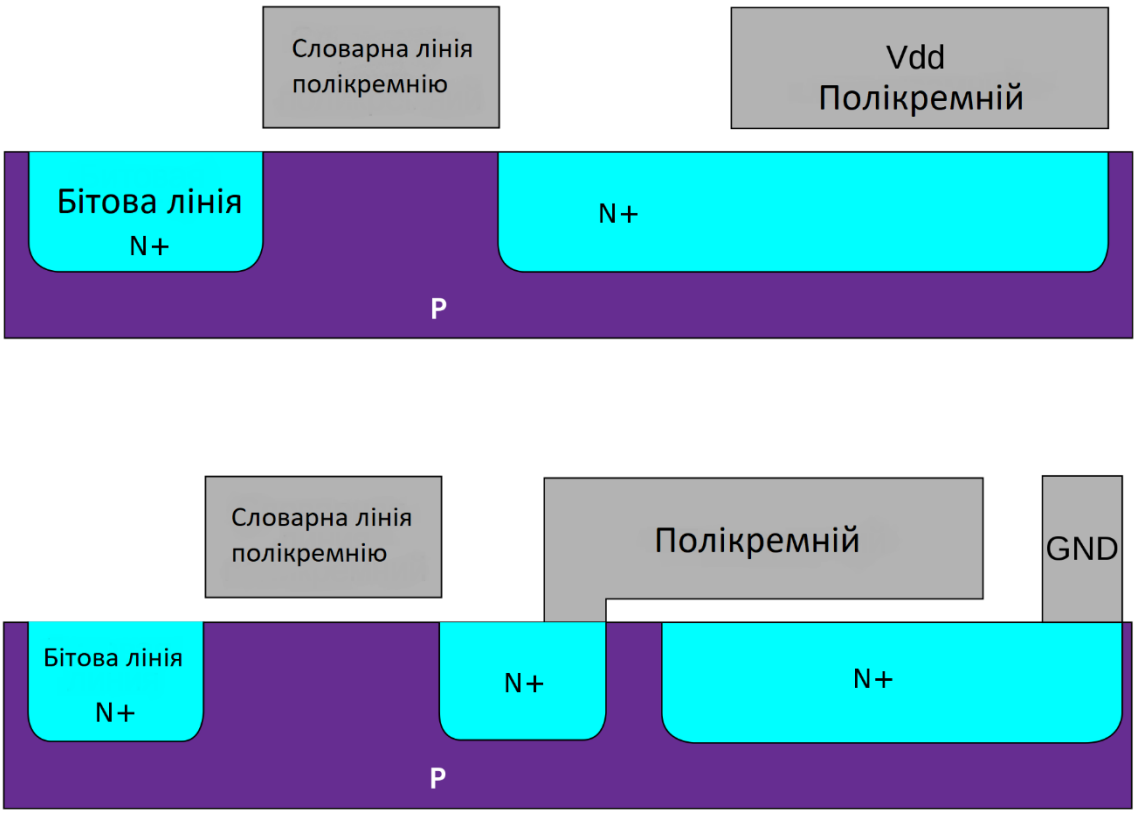

Рисунок 1.3 - Схема поперечного перерізу однобітових осередків пам'яті з одним транзистором і одним конденсатором

(Запозичено з [https://ru.wikipedia.org/wiki/DRAM])

На фізичному рівні пам'ять DRAM являє собою набір осередків, здатних зберігати інформацію. Осередки складаються з конденсаторів і транзисторів, розташованих усередині напівпровідникових мікросхем пам'яті. Конденсатори заряджають під час запису в комірку одиничного біта і розряджають при записі в осередок нульового біта.

При припиненні подачі електроенергії конденсатори розряджаються, і пам'ять обнуляється (спустошується). Для підтримки необхідного напруги на обкладках конденсаторів (для збереження даних) конденсатори необхідно періодично підзаряджати. Підзарядку виконують шляхом подачі на конденсатори напруги через комутуючі транзисторні ключі. Необхідність постійної зарядки конденсаторів (динамічне підтримку заряду конденсаторів) є основоположним принципом роботи пам'яті типу DRAM.

Важливим елементом пам'яті типу DRAM є чутливий підсилювач-компаратор (англ. Sense amplifier), підключений до кожного з стовпців «прямокутника». При читанні даних з пам'яті підсилювач-компаратор реагує на слабкий потік електронів, котрі рвонули через відкриті транзистори з обкладок конденсаторів, і зчитує один рядок цілком. Читання і запис виконуються через підрядник; обмін даними з окремо взятої осередком неможливий.

На відміну від статичної пам'яті (пам'яті типу SRAM (англ. Static random access memory), є конструктивно більш складною, більш дорогою, більш швидкою і застосовуваною в основному в кеш-пам'яті), повільна, але дешева динамічна пам'ять (DRAM) виготовляється на основі конденсаторів невеликої ємності. Такі конденсатори швидко втрачають заряд, тому щоб уникнути втрат даних, що зберігаються конденсатори доводиться заряджати через певні проміжки часу. Цей

процес називається регенерацією пам'яті, здійснюється спеціальним контролером, встановленим або на материнській платі, або на кристалі центрального процесора. Протягом часу, званого кроком регенерації, в DRAM перезаписується ціла рядок комірок, і через 8-64 мс оновлюються всі рядки пам'яті.

Процес регенерації пам'яті в класичному варіанті суттєво гальмує роботу системи, оскільки під час його здійснення обмін даними з пам'яттю неможливий. Регенерація, заснована на звичайному переборі рядків, в сучасних типах DRAM не застосовується.

Серед нових технологій регенерації - PASR (англ. Partial array self refresh), що застосовується деякими компаніями в чіпах пам'яті SDRAM, що відрізняються низьким рівнем енергоспоживання. Регенерація осередків виконується тільки в період очікування в тих банках пам'яті, в яких є дані. Одночасно з цією технологією застосовується технологія TCSR (англ. Temperature compensated self refresh), призначена для регулювання періоду регенерації в залежності від робочої температури.

## 1.3 FRAM або FeRAM

Пам'ять в сучасних мікроконтролерах прийнято розділяти за ознакою залежності від енергопостачання. До енергозалежною пам'яті відносяться технології DRAM і SRAM, до незалежної - EEPROM / Flash. Цей поділ існує за рахунок того, що DRAM / SRAM мають набагато кращу швидкодію в порівнянні з енергонезалежною пам'яттю. Але що було б, якби існувала незалежна пам'ять, яка не поступається енергозалежною пам'яті за швидкістю читання / запису і енергоспоживанню? Виявляється, такі технології існують. Одним з представників цього класу пам'яті є технологія FRAM або FeRAM.

Отже, FeRAM або Ferromagnetic Random Access non-volatile Memory - тип пам'яті, принцип роботи якого грунтується на ефекті гістерезису в сегнетоелектриках. При додатку до осередку електричного поля вона змінює свою поляризацію, переходячи на іншу ділянку петлі гистерезиса. За рахунок цього можна отримати два добре помітних по енергії стану, а це досить для створення пам'яті на основі такого осередку.

Для того, щоб зрозуміти, які переваги це дає перед класичними видами пам'яті, необхідно також згадати основні принципи роботи інших видів пам'яті.

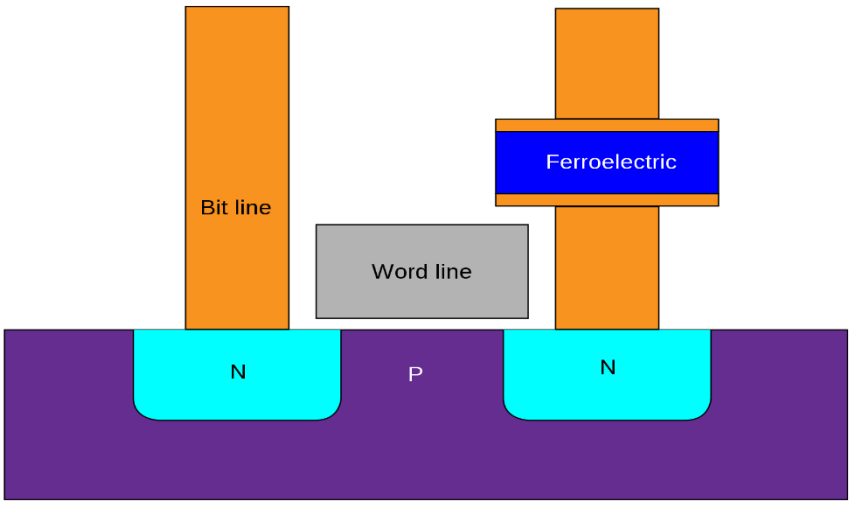


Рисунок 1.4 - Структура FeRAM комірки

(Запозичено з [https://ru.wikipedia.org/wiki/FRAM])

Принцип роботи DRAM (Dynamic RAM) заснований на зчитуванні і зміні заряду конденсатора. Якщо конденсатор заряджений - осередок знаходиться в стані «1», якщо розряджений - в стані «0».

Для збільшення швидкодії в осередках пам'яті застосовуються конденсатори невеликої ємності, заряд з яких відносно швидко витікає. Тому для забезпечення схоронності інформації інформацію доводиться регенерувати. DRAM застосовується в якості оперативної пам'яті на сучасних комп'ютерах через

дешевизну (в порівнянні з SRAM) і високої швидкодії (в порівнянні з дисковими накопичувачами).

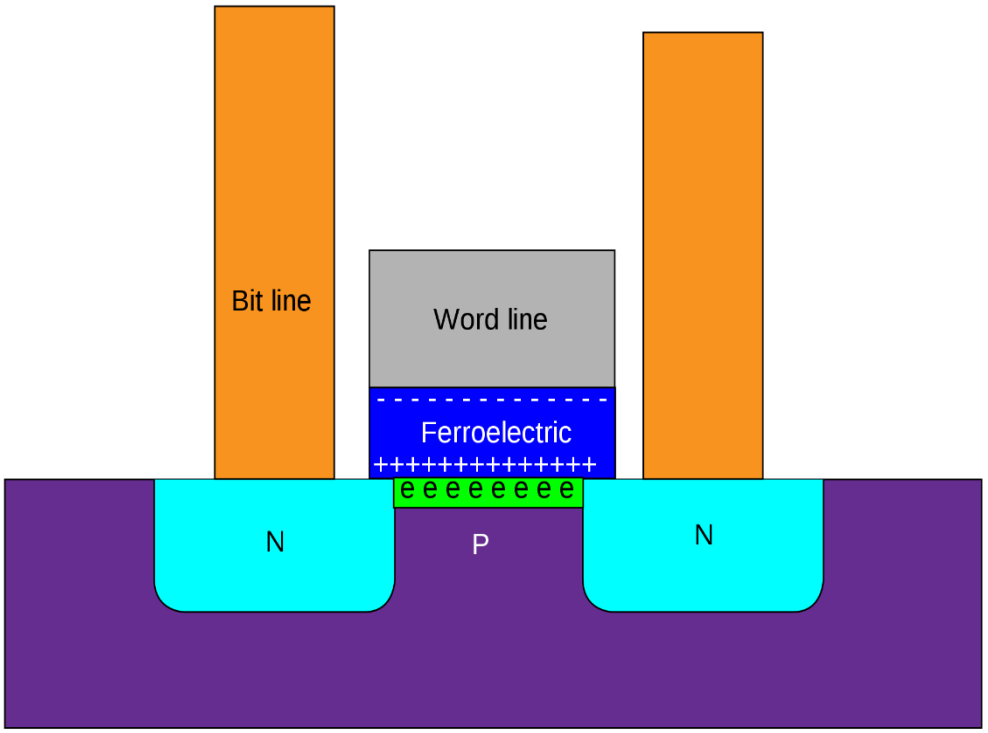


Рисунок 1.5 - Структура однотранзисторної FeRAM комірки та її робочий механізм

(Запозичено з [https://ru.wikipedia.org/wiki/FRAM])

Пам'ять SRAM (Static RAM) набагато складніше DRAM, і тому набагато дорожче. Її принцип дії заснований на застосуванні КМОП-транзисторів. При об'єднанні кількох транзисторів можна отримати тригер - осередок, що зберігає певне логічне стан. Для цього виду пам'яті немає необхідності в регенерації станів, але тим не менше за відсутності харчування дані губляться, тобто пам'ять залишається енергозалежною. Цей вид пам'яті швидше DRAM. Оскільки така пам'ять коштує набагато дорожче DRAM, її застосовують там, де потрібна дуже малий час відгуку - в кеш-пам'яті процесора.

Сучасні Flash і EEPROM засновані на застосуванні транзисторів з так званим плаваючим затвором. Електрони инжектируются в «кишеню» напівпровідникової структури, і їх наявність / відсутність може бути зареєстровано ззовні. Це і є властивість, яке дозволяє застосовувати такі структури

як пам'ять. Заряд з кишені хоч і витікає, але відбувається це досить повільно (~ 10-20 років), що дозволяє застосовувати EEPROM / Flash в якості незалежної пам'яті. Flash застосовується для зберігання коду програм в мікроконтролерних пристроїв, а також в картах пам'яті.

Переваги FRAM у порівнянні з іншими видами пам'яті?

Основна перевага FRAM перед SRAM - це енергонезалежність. При припиненні подачі живлення на мікросхему пам'яті вона зберігає свій попередній стан. У той же час швидкодія цих видів пам'яті є близькою між собою. Так, наприклад, цикл запису на FRAM займає 150 наносекунд проти 55 наносекунд в SRAM, згідно з сайтом Fujitsu. Проте FRAM має обмежене (біля $10^{13}$) число циклів перезапису, тоді як у SRAM такі обмеження відсутні. Разом з тим, DRAM має високе енергоспоживання в порівнянні з FRAM, через необхідність регенерації даних. Тому DRAM не застосовується в пристроях, чутливих до енергоспоживання.

Проте, хоча FRAM за характеристиками можна порівняти з SRAM, основний потенціал застосування зав'язаний на значні переваги перед Flash-пам'яттю. В першу чергу, це величезна швидкодія. З тієї ж посилання на сайт Fujitsu час одного циклу записи на Flash близько 10 мікросекунд. Тут слід згадати особливість застосування flash-пам'яті - запис і стирання в ній виробляється досить великими блоками. Тому перезаписувати один байт у флеш - дуже дороге задоволення як за часом, так і по енергоспоживанню - потрібно кудись зберегти блок даних, змінити в ньому байт, повністю стерти відповідну ділянку блоку і перезаписати в нього оновлені дані. Тут, до речі, ще одна перевага FRAM - це пам'ять з довільним доступом, а значить в ній можна міняти окремі біти, не зачіпаючи сусідні. Але навіть при записі великих блоків даних FRAM на порядок швидше. Так, в контролерах Texas Instrument запис блоку розміру 13 кБ займає 10 мс в FRAM проти 1 секунди в Flash (пруф). Ще один недолік Flash - сильно обмежена кількість циклів перезапису - порядку $10^{5}$.

Чому FRAM досі не витіснив Flash? Дійсно, всі характеристики FRAM на порядки краще характеристик флеш-памя’ті. Ось тут і спливли основні недоліки феромагнітної RAM. В першу чергу це низька щільність розміщення інформації в силу особливостей технології. З цього недоліку випливає інший - ємність накопичувачів типу FRAM не може бути зроблена досить великий. Fujitsu пропонує схеми пам'яті об'ємом до 4 Мбіт, що не порівняти з багатогігабайтними флешками. Ще один недолік - досить висока вартість пам'яті. Сьогодні FRAM займає зовсім мізерну частку ринку напівпровідникових пристроїв.

Для яких же застосувань оптимальна пам'ять типу FRAM? Досить добре FRAM працює в мікроконтролерах в комбінації з невеликим об'ємом SRAM. Власне, це те саме застосування, яке привернуло мене до даного типу пам'яті. Наприклад, компанія Texas Instruments випустила лінійку FRAM-мікроконтролерів з повністю відсутніми Flash / EEPROM. Код в них записується в FRAM сегмент, а до даних в тому ж FRAM можна звертатися також, як до звичайної RAM-пам'яті. Таке застосування зручно там, де є значна кількість даних, яке може часто листуватися. Наприклад, портативний логгер, для якого важливо енергоспоживання. Можна записувати дані в FRAM протягом певного часу, потім аналізувати і, наприклад, відправляти дані про середні величини по бездротовому каналу. Flash пам'ять при такому використанні незручна - вона швидко посадить акумулятор, а через обмеженість циклів запису через якийсь час можуть з'явитися проблеми з пошкодженими елементами пам'яті. Таким чином, FRAM вигідна для low-power додатків з відносно великим об'ємом і високою частотою записи в незалежну пам'ять. Взагалі, TI на своєму сайті вказує, в яких областях на їхню думку така пам'ять найбільш зручна.

Можливості встановленого FRAM:

- Набір утиліт API, який дозволяє легко користуватися режимами низької потужності LPMx.5 та потужним режимом вимкнення, що

дозволяє програмі зберігати та відновлювати важливі компоненти системи при виявленні втрати потужності.

- Стиснення LZ4: вбудована утиліта стиснення на основі формату стиснення LZ4 з відкритим кодом. Оптимізовано для мікроконтролерів MSP, щоб забезпечити кращу реєстрацію даних, передачу даних, оновлення мікропрограми тощо.
- Генератор випадкових чисел: Утиліта генератора випадкових чисел (RNG) реалізує детермінований генератор випадкових байтів (CTR-DRBG) у режимі лічильника відповідно до специфікації NIST SP 800-90A Rev 1
- Енергонезалежне сховище: Бібліотека енергонезалежного сховища (NVS) робить обробку енергонезалежних даних легкою та надійною проти переривчастої втрати потужності або скидання асинхронних пристроїв.

Ще однією перевагою буде те, що FRAM - це дуже надійна технологія пам'яті, навіть при високій температурі. FRAM зберігає свої дані ще більше ніж 10 років при 85 градусах С, або 100 років при 25 ° С. Це далеко перевищує вимоги до більшості програм.

Поширене запитання щодо FRAM – чи є технологія надійною у питанні збереження даних? FRAM використовується в декількох автомобільних програмах і має кваліфікацію щодо витримування надзвичайно суворих умов.

# 2 ОГЛЯД ХАРАКТЕРИСТИК, КОМПОНЕНТІВ, БАЗОВОГО ВИГЛЯДУ РОБОЧОЇ СТАНЦІЇ ТА ЗАГАЛЬНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

## 2.1. Огляд робочого середовища CCS – Code Composer Studio

CCS – Code Composer Studio, що є єдиною сервісною програмою, яка дозволяє нам як користовачу створювати, змінювати та компілювати елементи або програми цілком для мікроконтролерів MSP.

Як і у більшості середовищ розробки, у CCS передбачені:

- Поля вводу коду
- Вікна перегляду значень змінних
- Точки переривання
- Графічний аналіз
- Мови вводу – C, Асемблер, C++
- Компілятор

Сам процес виглядає наступним чином:

Створення проекту – написання робочогу коду – додавання бібліотек із передустановами – компілятор – ассемблер – будування програми. Схематично це виглядає так:

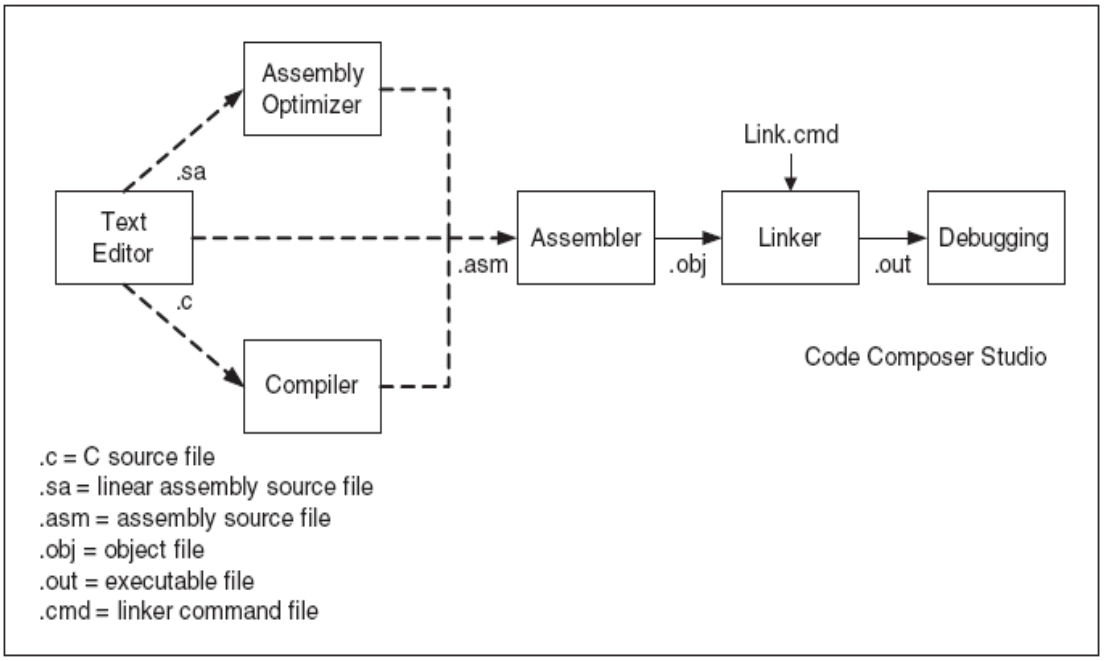


Рисунок 2.1 - Етапи проектування програми

(Запозичено з [https://studfile.net/preview/3991398/page:50/])

Код розробляється апаратно-незалежним, що дозволяє програмісту модуля не думати про розподіл пам'яті, так як ця робота буде зроблена на етапі компонування проекту. При зміні будь-якого з модулів проводиться нова збірка проекту, і питання розподілу пам'яті вирішуються заново, виключаючи ймовірність появи конфліктів.

## 2.2 Мікроконтролер MSP430FR2355

Набір містить все необхідне для початку розробки на платформі мікроконтролера MSP430FR2355 FRAM, включаючи бортовий прилад для відладки, програмування, та вимірювання енергії. На платі також є вбудовані кнопки та світлодіоди для швидкої інтеграції простого користувальницького інтерфейсу, вбудованого роз'єму Grove для зовнішніх датчиків Grove, а також датчик зовнішнього освітлення для демонстрації вбудованої аналогової периферії.

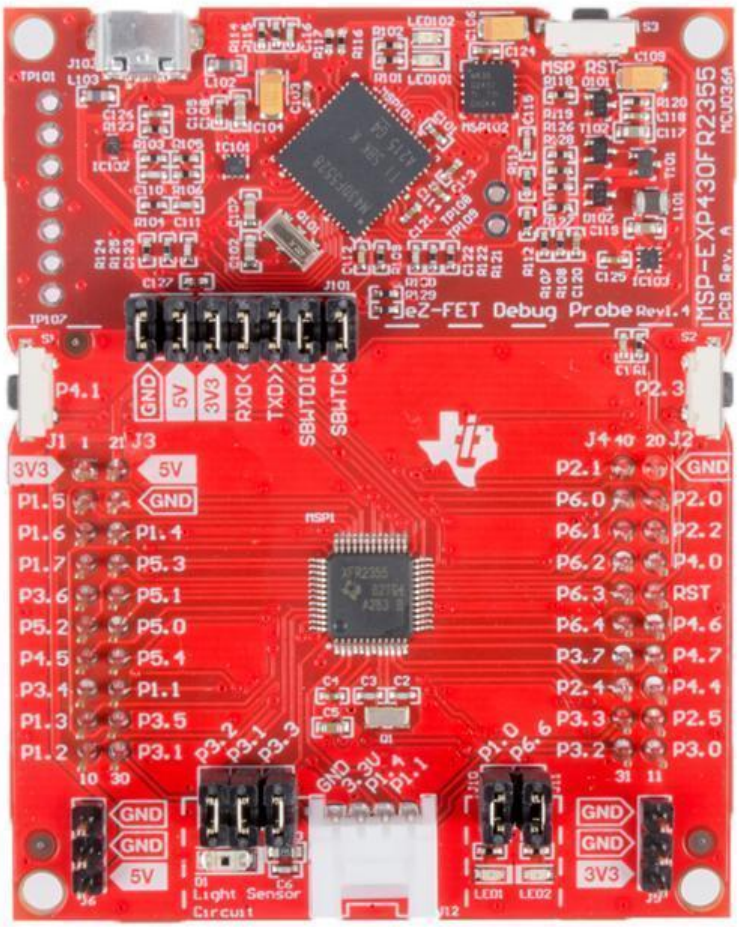


Рисунок 2.2 - Вигляд мікроконтроллера

(Запозичено з [https://www.ti.com/product/MSP430FR2355?keyMatch=MSP430FR2355&ti])

Пристрій MSP430FR2355 з частотою 24 МГц має 32 КБ вбудованого FRAM (сегнетоелектричний довільний доступ пам'яті), енергонезалежна пам'ять, відома своєю наднизькою потужністю, високою витривалістю та високою швидкістю запису доступ. У поєднанні з 4 КБ вбудованої оперативної пам'яті користувачі мають доступ до 32 КБ пам'яті, щоб розділити їх між собою програма та дані за необхідності. Наприклад, програма реєстрації даних може вимагати великої пам'яті даних з відносно невеликою програмною пам'яттю, тому пам'ять може бути розподілена за необхідністю між програмою та пам'яттю даних.

Швидке створення прототипів спрощується за допомогою 40-контактних заголовків модулів BoosterPack ™, які підтримують широкий діапазон доступних модулів BoosterPack. Ви можете швидко додати такі функції, як бездротове підключення, графічні дисплеї, зондування довкілля та багато іншого. Створіть свій власний модуль підключення BoosterPack або вибирайте серед багатьох уже доступних від TI та сторонніх розробників.

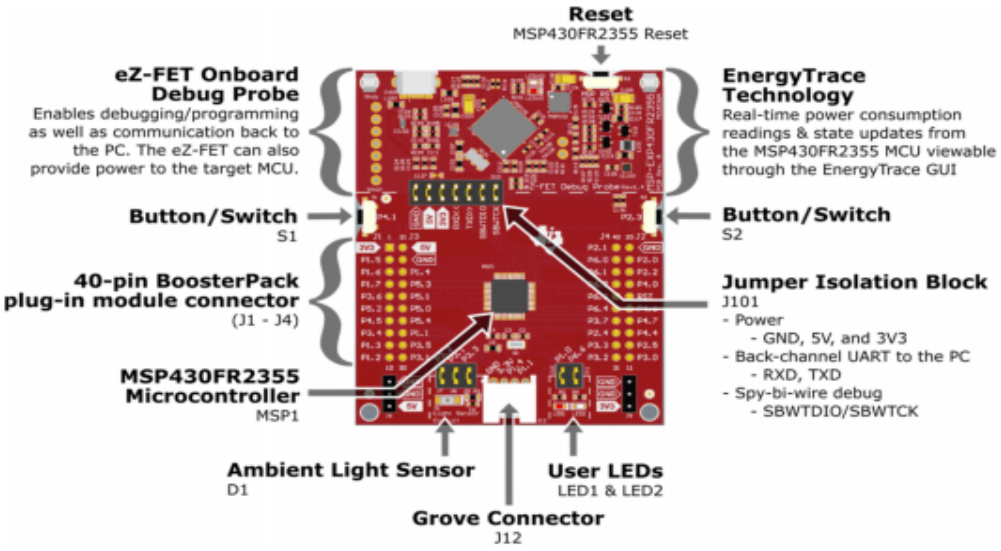


Рисунок 2.3 - Основні елементи плати

(Запозичено з [https://www.ti.com/product/MSP430FR2355?keyMatch=MSP430FR2355&ti])

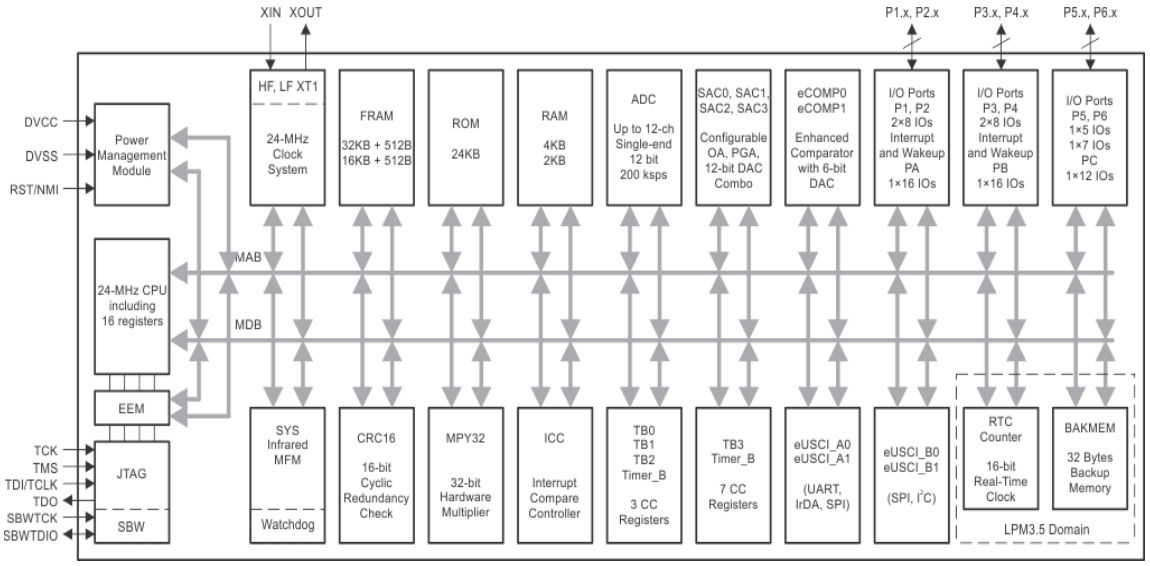


Рисунок 2.4 - Блок-схема MSP430FR2355

(Запозичено з [https://www.ti.com/product/MSP430FR2355?keyMatch=MSP430FR2355&ti])

Основні модулі МК:

CPU (Central Processing Unit) – центральний процесор.

RAM (Random Access Memory) – пам’ять з довільним доступом, чи оперативний запам’ятовуючий пристрій. RAM призначена для 4 зберігання проміжних результатів та інших тимчасових даних протягом виконання програми.

ROM (Read Only Memory) – пам’ять тільки для читання, чи постійно запам’ятовуючий пристрій. ROM зберігає програмні інструкції та таблиці з довідковими даними. У сучасних контролерах вона реалізована у вигляді flesh-пам’яті.

EEPROM (Electrically Erasable and Programmable ROM) – енергонезалежна пам’ять даних. У сучасних МК вона представляє собою flash-пам’ять. Основна відмінність від flash-пам’яті програм (ROM) – це можливість вибіркового програмування окремих байтів, на відміну від поблокового програмування flash-пам’яті програм.

I/O Ports (input/output) – паралельні порти вводу виводу надають інтерфейс між МК та периферійними пристроями вводу/виводу, такими як: клавіатура, дисплей тощо. Ніжки портів можуть бути як двонаправленими, так і однонаправленими, або лише на вхід, або на вихід.

Serial I/O – послідовні порти для обміну даними, що реалізовують асинхронні (напр., UART) або синхронні (напр., SPI) інтерфейси обміну даними. Асинхронний інтерфейс використовує протокол зі стартовим та стоповим бітами для передачі та прийому. Стартовий та стоповий біти вбудовані у кожен байт даних. Синхронні інтерфейси використовують синхронізуючі імпульси для кожного біта.

Timer/Counter (таймер/лічильник) – використовують для відліку часу меж часових інтервалів між подіями, підрахунку подій, генерації швидкості передачі даних для послідовних портів. Таймери також можуть керувати певними портами у контролері, як приклад, здійснювати підрахунок імпульсів, що поступають на його вхід, або навпаки виводити певні послідовності імпульсів.

ADC (Analog to Digital Converter) – аналогово-цифровий перетворювач – інтерфейс для роботи з аналоговими пристроями.

DAC (Digital to Analog Converter) – цифро-аналоговий перетворювач.

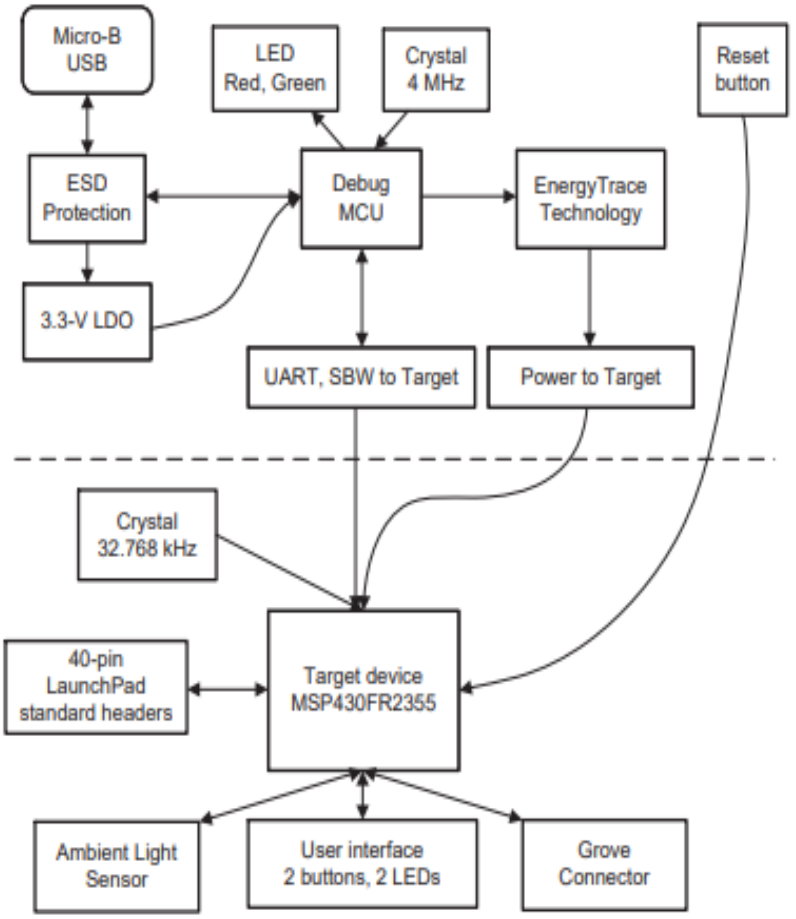


Рисунок 2.5 - Блок-діаграма МК

(Запозичено з [https://www.ti.com/product/MSP430FR2355?keyMatch=MSP430FR2355&ti])

# 3 РОБОТА З ІНШИМИ BUILT-IN МОДУЛЯМИ

## 3.1 Smart Analog Combo

SAC – «Розумне» комбінування сигналів, забезпечує настроюваний ланцюжок циркулюючого сигналу для обробки та об’єднання. Завдяки ньому стає можливим гнучко налаштувати Op-Amp, PGA підсилювача сигналу або 12-бітного ЦАП для внутрішніх IP і зовнішніх замірів напруги.

Переваги:

- Налаштуємі вхідні та вихідні піни
- Адаптована схема аналогового циркулювання
- Можливість програмування змінних
- Зрозумілий «user-friendly» інтерфейс
- Відсутність необхідності у додаткових модулях
- Низьке енергоспоживання

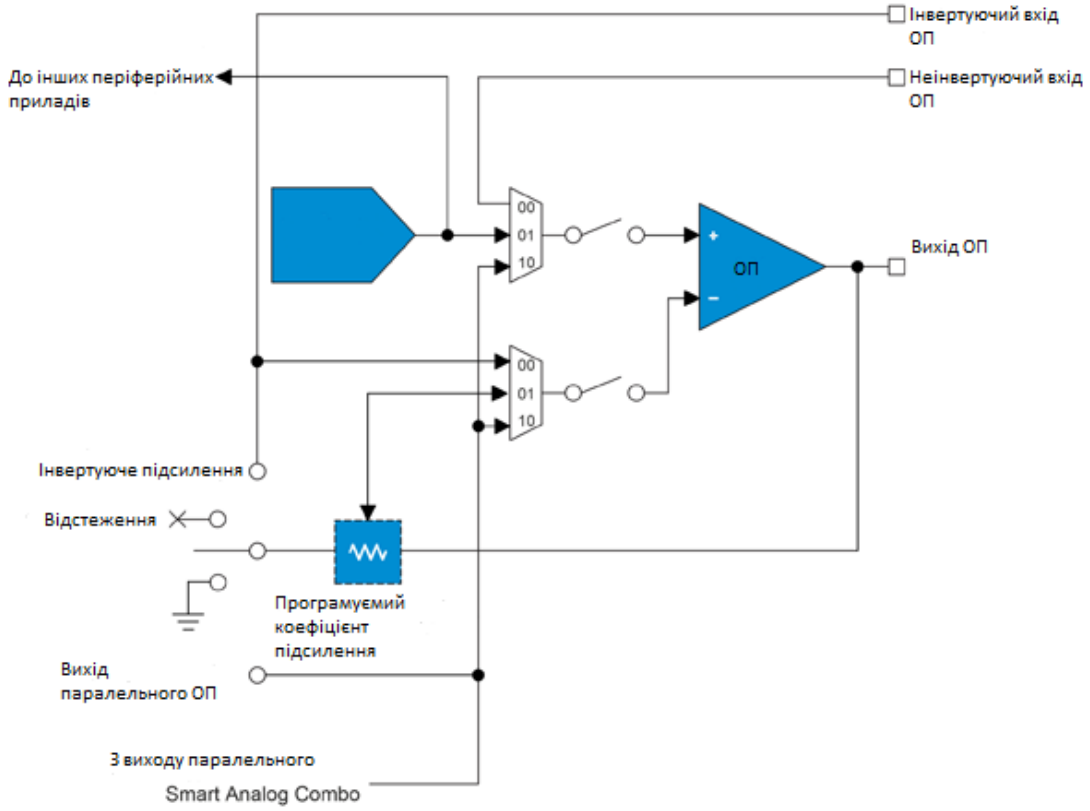


Рисунок 3.1 - Схематичне зображення компонентів SAC
(Запозичено з [https://www.compel.ru/lib/131579])

## 3.2 SAC у якості детектору диму

Існує два методи виявлення диму: за допомогою вимірювання змінного струму або вимірювання постійного струму.

Для вимірювання змінного струму загальний регулятор сигналу напруги вимірює струм фотодіода через блок блокування постійного струму (як правило, великий конденсатор) перед аналого-цифровим перетворенням. На малюнку 2.4 показано змінний струм детектору диму на основі вимірювань, реалізований з використанням двох розумних аналогових комбінованих конфігурацій SAC-L3.

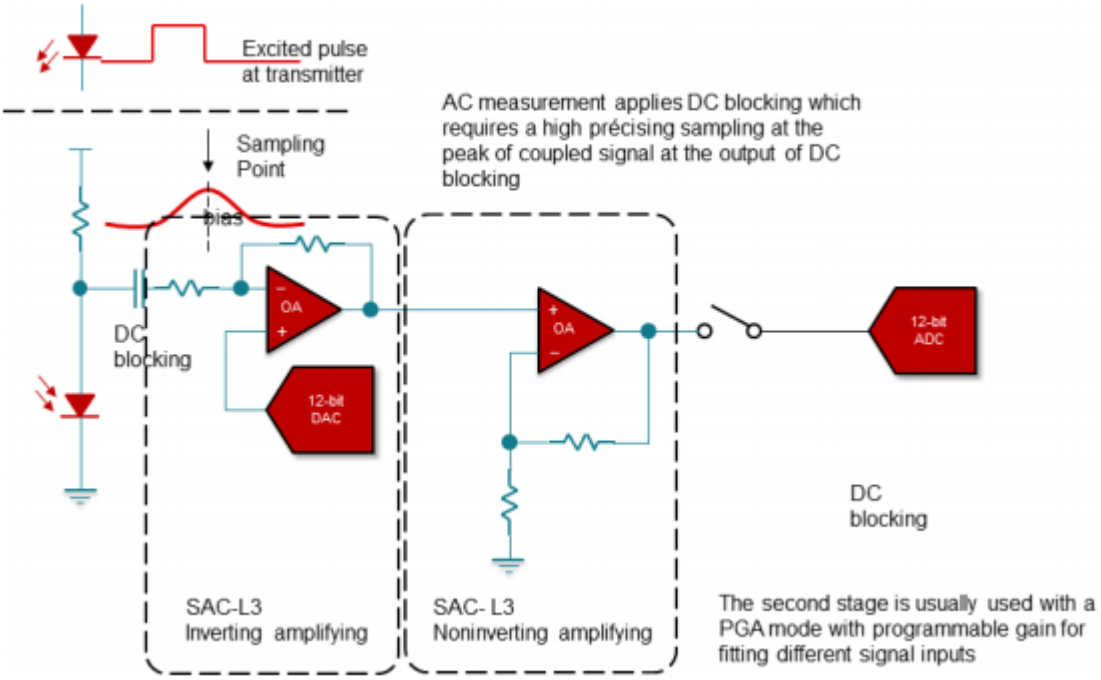


Рисунок 3.2 - Схема детектору диму
(Запозичено з [https://www.ti.com/product/MSP430FR2355?keyMatch=MSP430FR2355&ti])

Для вимірювання постійного струму перетворювач струму в напругу (зазвичай використовується підсилювач трансимпедансу (TIA) з входом із низьким рівнем витоку) вимірює фотодіодний струм за допомогою додаткового циклу сигналу другого ступеня перед аналого-цифровим перетворенням. Замінивши схему зворотного зв'язку на діод або транзистор, в додаток може буии включено логарифмічний підсилювач паралельно, щоб розширити вимірюваний динамічний

діапазон на стискання великих вихідних сигналів. SAC може виконувати функції TIA та внутрішніх з'єднань за допомогою АЦП та спареного SAC може зберегти більше робочої потужності. На малюнку 2.5 показано схему детектора диму на основі вимірювання постійного струму

Детектор, реалізований за допомогою смарт-аналогової комбінованої конфігурації SAC-L3.

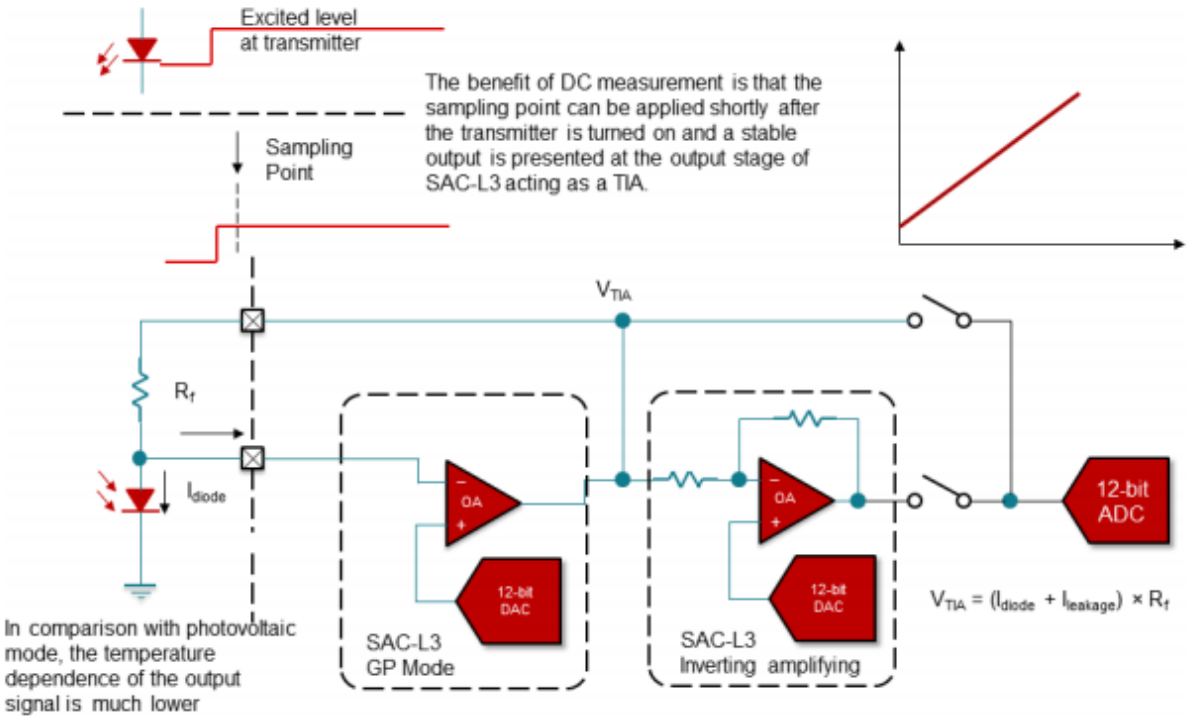


Рисунок 3.3 - Схема на основі вимірювання постійного струму

(Запозичено з [https://www.ti.com/product/MSP430FR2355?keyMatch=MSP430FR2355&ti])

## 3.3 SAC у якості детектору газу

Перетворювач струму в напругу (як правило, підсилювач з повним імпедансом із входом з низьким витоком) вимірює фотодіодний струм, з додатковим регулятором сигналу напруги другого ступеня перед аналого-цифровим перетворення.

У цьому прикладі зазвичай використовується акумулятор; MCU наднизької потужності та пов'язані з ним аналогові модулі найбільше сприяють споживанню енергії. Особливо це стосується вузла газового датчика, який повинен постійно працювати, щоб контролювати концентрацію вуглекислого газу та подавати сигнал тривоги, якщо концентрація перевищує задану межу протягом певного часу. SAC наднизької потужності і використовуються в мікроконтролерах FRAM, перевагою яких є той факт, що вони можуть працювати у режимі постійного контролю сигналу, для таких задач як реалізація газових детекторів. На малюнку 2.6 показано а блок-схема газового детектора з використанням інтелектуальної комбінованої конфігурації SAC-L3.

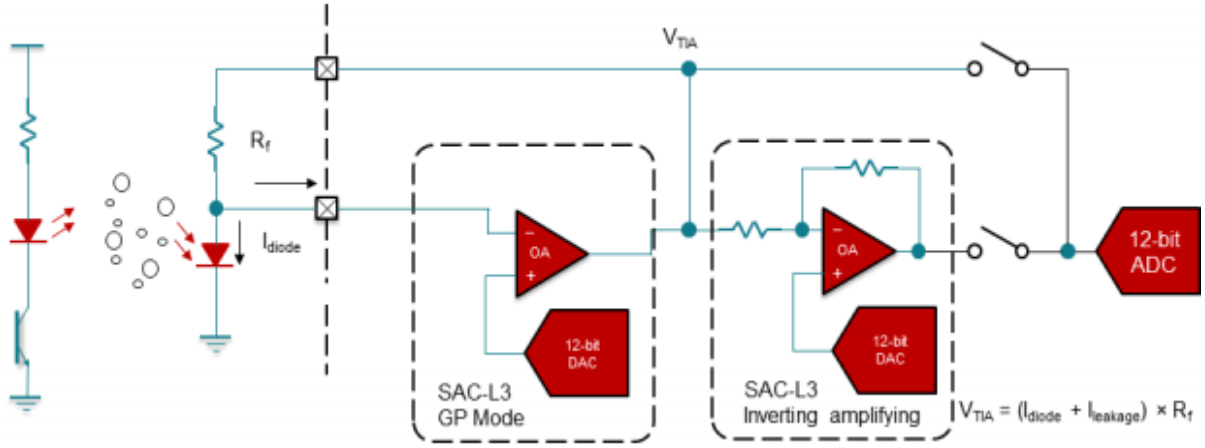


Рисунок 3.4 - Схема для детектора газу

(Запозичено з [https://www.ti.com/product/MSP430FR2355?keyMatch=MSP430FR2355&ti])

## 3.4 Сенсор освітленості

Основою даного сенсору є фоторезистор – напівпровідниковий прилад, що змінює величину свого опору, при освітленні його фоточутливих елементів. Даний принцип грунтується на ефекті фотопровідності – явищі зменшення опору напівпровідника при збудженні світлом його носіїв заряду. Основна характеристика ефекту – одинакова провідність незалежно від напрямку

протікання струму. Сульфід кадмію це найпопулярніший матеріал для виготовлення фоторезисторів.

Фоторезисторами є менш світлочутливими за фотодіоді або фототранзисторами, оскільки два останніх є справжніми напівпровідниковими приладами, в той час як фоторезистор є пасивним компонентом і не має p-n-переходу. Фотоопір (електричний Опір) будь-якого фоторезистора може змінюватися в широких межах в залежності від температури навколишнього середовища, що робить їх непридатними для застосування, що вимагає точного вимірювання або чутливості до світла.

Для фоторезистора також характерна деяка затримка між дією світла і наступною зміною опору, значення якої, як правило, становить близько 10 мс. Час затримки при переході від освітлених до темних серед, є навіть ще більшим, і часто досягає 1 секунди. Ця властивість робить фоторезистори непридатними до вимірювання об'єктів, які швидко блимають, але іноді вони використовуються для згладжування реакції стиснутого аудіосигналу.

Розрахунок потужності джерела світла. Використовуючи значення чутливості при даній довжині хвилі і застосовуючи закон Ома (V = IR), можна розрахувати потужність випромінювання, що пройшов через активну область детектора.

Наприклад, вихідний сигнал фотодетектора має потужність 20 мВ, довжина хвилі лазерного випромінювання 632.8 нм. Навантажувальний опір 50 Ом, потужність вхідного сигналу розраховується як: I = 0.02 В / 50 Ом, звідки I = 0.0004 А.

Спектральна чутливість кремнієвого фотодетектора при довжині хвилі випромінювання 632.8 нм становить 0.4 А / Вт. Тоді 0.0004 А / 0.4 А / Вт = 1 мВт - шукана потужність вхідного випромінювання.

Потрібно відзначити, що розрахунки наводяться тільки для сигналу, яка зачепила робочу область фотодетектора. Розрахунки абсолютної потужності

сигналу не настільки важливі для практичних застосувань, оскільки в реальних установках присутні втрати.

Схема електричного кола фотодетекторів на основі арсеніду галію-індію і кремнію, частотний діапазон менше 2 ГГц:

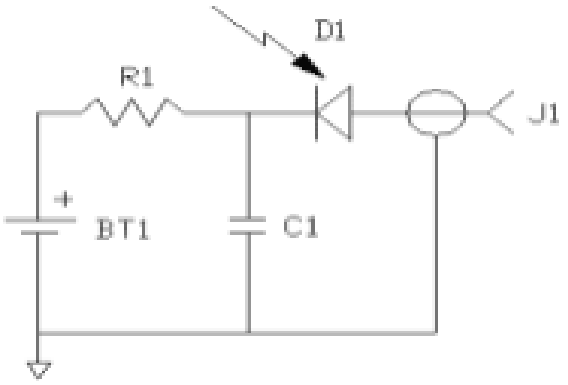


Рисунок 3.5 - Принципова схема фотоприймачів

Режим датчика світла використовує пару SAC0 і SAC2, два модулі Timer_B та АЦП MCU MSP430FR2355 у поєднанні з вбудованим фотодіодами і світлодіодами розробки TI, щоб реалізувати просту програму виміру світла. Він налаштовує SAC2 як загальний підсилювач і разом з R3 і C6(див рис. 1.2), реалізує підсилювач трансимпедансу для перетворення струму фотодіода в напругу. Для вимірювання цієї напруги за допомогою АЦП пристрою, вихідна напруга SAC2 потім подається через SAC0 (налаштований як буфер), вихід якого підключений внутрішньо до АЦП. В АЦП за замовчуванням поріг напруги встановлюється приблизно наполовину повного діапазону виявлення яскравості D1. Якщо вимірювання АЦП нижче цього порогу яскравості, загоряється червоний світлодіод 1, і якщо вимірювання АЦП вище за замовчуванням поріг, зелений світлодіод 2 загоряється відповідно до інтенсивності навколишнього світла. Щоб відкалібрувати поріг до значення АЦП, налаштованого від навколишнього освітлення, натисніть S1. Користувач може впливати на кількість світла, що доходить до фотодіода, використовуючи ліхтарик або покриваючи фотодіод і спостерігаючи за зміною яскравості світлодіодів.

# ВИСНОВКИ

У даній роботі розроблено елементи програмного забезпечення для операцій з сегнетоелектричною пам’ятю енергоефективного мікроконтролера MSP430FR2355, а також реалізація алгоритму для роботи світлочутливого сенсору, детектору газу, диму, що можуть бути використані в системах автоматичного регулювання освітленості, аварійних системах, системах охорони, та інших автоматизованих системах. Використання мікроконтролерів з сегнетоелектричною пам’ятю дозволяє створювати пристрої з низьким рівнем енергоспоживання, які можуть використовуватись в системах Інтернет Речей.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ


1. MSP430 вступ [Електронний ресурс] // Prof Kaikai Liu.– Режим доступу до ресурсу: Http://kaikailiu.cmpe.sjsu.edu/embedded-system/msp430-lab1-introduction/.
1. Texas Instruments [Електронний ресурс] // форум CCSРежим доступу до ресурсу:Https://e2e.ti.com/support/tools/ccs/f/81/t/321694.
2. Сайт офіційного ритейлеру [Електронний ресурс] // Mouser electronics. – Режим доступу до ресурсу: Https://eu.mouser.com/new/Texas-Instruments/ti-msp430-microcontrollers/.
3. Фізичний рівень IEEE [Електронний ресурс] //ЦДПУ wiki – Режим доступу до ресурсу:Https://wiki.cuspu.edu.ua/index.php/Фізичний_рівень_IEEE_802.11.
4. FRAM [Електронний ресурс] //Wikipedia. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/FRAM.
5. Файловий архів: Робота FRAM [Електронний ресурс] // Stupfiles. – Режим доступу до ресурсу: https://www.ti.com/lit/wp/slat151/slat151.pdf?ts=1616673774895&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fdocs%252Funiversalsearch.tsp%253FsearchTerm%253Dfram
6. SAC [Електронний ресурс] //TI.com – Режим доступу до ресурсу: https://www.ti.com/lit/wp/slaa835b/slaa835b.pdf?ts=1616981208312.
7. Фотодетектор [Електронний ресурс] //Wikipedia. – Режим доступу до ресурсу: Https://uk.wikipedia.org/wiki/Фотодетектор.
8. Основиелектроніки / П.Г. Стахів, В.І. Коруд, О.Є. Гамала, 2003. – (УДК: 621.38.075.8). – С77. – С. 90 – 93. (http://dspace.tneu.edu.ua/bitstream/316497/8500/1/fkit_kki_dke_ksm_LEK.pdf)
9. Фотодетектор [Електронний ресурс] //In-science. – Режим доступу до ресурсу: https://in-science.ru/library/article_post/fotodetektory-eot

Додаток А

Переключення світодіоду із частотою 10 Герц

```
void FRAMWrite(void);

unsigned char count = 0;
unsigned long *FRAM_write_ptr;
unsigned long data;

#define FRAM_TEST_START 0x1800



int main(void)
{
   WDTCTL = WDTPW | WDTHOLD;             // Stop watchdog timer

    P1OUT &= ~BIT0;                      // Clear P1.0 output latch for a defined
power-on state
   P1DIR |= BIT0;                       // Set P1.0 to output directionOUT

    PM5CTL0 &= ~LOCKLPM5;                // Disable the GPIO power-on
default high-impedance mode
                                         // to activate previously configured port settings
   data = 0x11111111;                    // Initialize dummy data

   while(1)
   {
      data += 0x00010001;
      FRAM_write_ptr = (unsigned long *)FRAM_TEST_START;
      FRAMWrite();
      count++;
```

```
        if (count > 100)
        {
            P1OUT ^= 0x01;              // Toggle LED to show 512 bytes
            count = 0;                  // have been written
            data = 0x11111111;
        }
    }
}

void FRAMWrite (void)
{
    unsigned int i=0;
    SYSCFG0 = FRWPPW | PFWP;
    for (i = 0; i < 128; i++)
    {
        *FRAM_write_ptr++ = data;
    }
    SYSCFG0 = FRWPPW | DFWP | PFWP;
}
```

Додаток Б

Активація світлодіоду при освітленні датчика

```
#include "driverlib.h"
#include "lightsensor.h"

int led1_dutycycle = 0;
int led2_dutycycle = 0;
int calibratedADC = 500;
int period = 100;
int counter = 0;
int deadzone = 5;
int runningAvg = 500;

/* FRAM Variable that stores lightsensor ADC results*/
#if defined(__TI_COMPILER_VERSION__)
#pragma PERSISTENT(lightsensor_ADC_Result)
#elif defined(__IAR_SYSTEMS_ICC__)
__persistent
#endif
unsigned int lightsensor_ADC_Result = 0;          // ADC conversion result

void lightsensor() {
    /* Initialize peripherals */
    lightsensor_init_GPIO();
    lightsensor_init_LED_PWM();
    lightsensor_init_SACOA();
    lightsensor_init_ADC();

    while(mode == LIGHTSENSOR_MODE){
        __bis_SR_register(LPM0_bits + GIE);
```

```
        runningAvg = (( runningAvg * 9 ) + lightsensor_ADC_Result)/10;
        int diff = (runningAvg - calibratedADC)/4;

        if (diff < deadzone) {
            diff *= -1;
            led1_dutycycle = diff - deadzone;
            led2_dutycycle = 0;
        }
        else if (diff > deadzone) {
            led1_dutycycle = 0;
            led2_dutycycle = diff - deadzone;
        }
        else {
            led1_dutycycle = 0;
            led2_dutycycle = 0;
        }
    }
}

void lightsensor_init_GPIO(void) {
    //Set Px.x to output direction

    P1DIR |= 0xFF;
    P1OUT = 0x00;
    P3DIR |= 0xFF;
    P3OUT = 0x00;
    P6DIR |= 0xFF;
    P6OUT = 0x00;

    //Set P4.1 (S1) to input
```

```
    P4DIR &= (~BIT1);
    P4OUT |= BIT1;
    P4REN |= BIT1;
    P4IES |= BIT1;
    P4IFG &= ~BIT1;
    P4IE |= BIT1;
}

/* Configure TimerB0 to generate PWM on LEDs */
void lightsensor_init_LED_PWM(void) {
    //Start timer
    Timer_B_initUpModeParam param = {0};
    param.clockSource = TIMER_B_CLOCKSOURCE_ACLK;
    param.clockSourceDivider = TIMER_B_CLOCKSOURCE_DIVIDER_1;
    param.timerPeriod = 1;
                                        param.timerInterruptEnable_TBIE          =
TIMER_B_TBIE_INTERRUPT_DISABLE;
    param.captureCompareInterruptEnable_CCR0_CCIE =
        TIMER_B_CCIE_CCR0_INTERRUPT_ENABLE;
    param.startTimer = true;
    Timer_B_initUpMode(TIMER_B0_BASE, &param);
}

void lightsensor_disable_LED_PWM(void) {
    Timer_B_stop(TIMER_B0_BASE);
}

void lightsensor_init_SACOA(void) {
    //Configure Op-Amp functionality
    GPIO_setAsPeripheralModuleFunctionInputPin(GPIO_PORT_P3,
```

```
                                GPIO_PIN1 | GPIO_PIN3 | GPIO_PIN2,
                                GPIO_TERNARY_MODULE_FUNCTION);

    //Select external source for both positive and negative inputs
                                                SAC_OA_init(SAC2_BASE,
SAC_OA_POSITIVE_INPUT_SOURCE_EXTERNAL,
            SAC_OA_NEGATIVE_INPUT_SOURCE_EXTERNAL);

    //Select low speed and low power mode
                                            SAC_OA_selectPowerMode(SAC2_BASE,
SAC_OA_POWER_MODE_LOW_SPEED_LOW_POWER);

    SAC_OA_enable(SAC2_BASE);               // Enable SAC2 OA
    SAC_enable(SAC2_BASE);                  // Enable SAC2

    //Select external source for both positive and negative inputs
                                                SAC_OA_init(SAC0_BASE,
SAC_OA_POSITIVE_INPUT_SOURCE_PAIR_OA,
            SAC_OA_NEGATIVE_INPUT_SOURCE_PGA);

    SAC_OA_enable(SAC0_BASE);               // Enable SAC0 OA
    SAC_enable(SAC0_BASE);                  // Enable SAC0
}

void lightsensor_init_ADC(void) {
    //Initialize the ADC Module
    ADC_init(ADC_BASE,
            ADC_SAMPLEHOLDSOURCE_2,
            ADC_CLOCKSOURCE_ADCOSC,
            ADC_CLOCKDIVIDER_1);
```

```
ADC_enable(ADC_BASE);

ADC_setupSamplingTimer(ADC_BASE,
                       ADC_CYCLEHOLD_16_CYCLES,
                       ADC_MULTIPLESAMPLESDISABLE);

ADC_setResolution(ADC_BASE,
                  ADC_RESOLUTION_12BIT);

//Configure the Memory Buffer
ADC_configureMemory(ADC_BASE,
                    ADC_INPUT_VEREF_P,
                    ADC_VREFPOS_AVCC,
                    ADC_VREFNEG_AVSS);

ADC_clearInterrupt(ADC_BASE,
                   ADC_COMPLETED_INTERRUPT);

//Enable the Memory Buffer Interrupt
ADC_enableInterrupt(ADC_BASE,
                    ADC_COMPLETED_INTERRUPT);

// Generate sample/hold signal to trigger ADC conversion
Timer_B_outputPWMParam param2 = {0};
param2.clockSource = TIMER_B_CLOCKSOURCE_ACLK;
param2.clockSourceDivider = TIMER_B_CLOCKSOURCE_DIVIDER_1;
param2.timerPeriod = TIMER_PERIOD;
param2.compareRegister = TIMER_B_CAPTURECOMPARE_REGISTER_1;
```

```
                                                    param2.compareOutputMode        =
TIMER_B_OUTPUTMODE_TOGGLE_RESET;
   param2.dutyCycle = DUTY_CYCLE;
   Timer_B_outputPWM(TIMER_B1_BASE, &param2);

   //Enable and Start the conversion
   //in Single-Channel, Single Conversion Mode
                                                    ADC_startConversion(ADC_BASE,
ADC_REPEATED_SINGLECHANNEL);
}

#if                    defined(__TI_COMPILER_VERSION__)                ||
defined(__IAR_SYSTEMS_ICC__)
#pragma vector=TIMER0_B0_VECTOR
__interrupt
#elif defined(__GNUC__)
__attribute__((interrupt(TIMER0_B0_VECTOR)))
#endif
void TIMER0_B0_ISR(void)
{
   counter++;
   if (counter > period)
   {
      counter = 0;
      if (led1_dutycycle > 0) {
         GPIO_setOutputHighOnPin(GPIO_PORT_P1, GPIO_PIN0);
      }
      if (led2_dutycycle > 0) {
         GPIO_setOutputHighOnPin(GPIO_PORT_P6, GPIO_PIN6);
      }
```

```
    }

    if (counter == led1_dutycycle)
    {
        GPIO_setOutputLowOnPin(GPIO_PORT_P1, GPIO_PIN0);
    }

    if (counter == led2_dutycycle)
    {
        GPIO_setOutputLowOnPin(GPIO_PORT_P6, GPIO_PIN6);
    }
}

#if                    defined(__TI_COMPILER_VERSION__)                    ||
defined(__IAR_SYSTEMS_ICC__)
#pragma vector=PORT4_VECTOR
__interrupt void PORT4_ISR(void)
#elif defined(__GNUC__)
void __attribute__ ((interrupt(PORT4_VECTOR))) PORT_4_ISR (void)
#else
#error Compiler not supported!
#endif
{
    P4IFG &= ~BIT1;                    // Clear P4.1 IFG
    calibratedADC = lightsensor_ADC_Result;
}
```