**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет кібербезпеки, комп’ютерної та програмної інженерії**

Кафедра інженерії програмного забезпечення



*Лабораторна робота №1*

*з дисципліни*

*«Інтернет речі»*

*на тему: “Зовнішнє переривання (external interrupt,*

*GPIO\_EXTI)”*

Виконав: *студент гр. ПІ-328Б*

*Присяжнюк Д.О.*

Прийняв:

*Денис Олександрович Навроцький*

Київ 2023

**Завдання:**

1. Дослідити, що відбувається, коли одночасно спрацьовують декілька зовнішніх

переривань;

2. Дослідити, що відбувається, коли переривання трапляється частіше, ніж

опрацьовується;

3. Розробити схему і написати код для гри «Ковбої», суть якої, що два гравці

одночасно натискають кнопки і має засвітитись світлодіод того гравця, який

швидше натиснув свою кнопку.

**Виконання:**

1. Коли виникає ситуація, коли декілька зовнішніх переривань спрацьовують одночасно, процесор має вирішити, яке переривання обробити першим. Цей вибір може залежати від кількох факторів, включаючи пріоритети переривань та визначений порядок їх обробки.

У деяких системах зовнішні переривання можуть мати призначені пріоритети, які визначають їх важливість. Процесор може використовувати ці пріоритети для вирішення конфліктів між перериваннями. Наприклад, переривання з вищим пріоритетом можуть бути оброблені першими, перш ніж перейти до переривань з нижчим пріоритетом.

Також можливий визначений порядок обробки переривань. Наприклад, система може мати певну чергу обробки, в якій переривання обробляються у визначеній послідовності. Це означає, що першим буде оброблено перше переривання, потім друге і так далі.

Однак важливо зазначити, що обробка зовнішніх переривань зазвичай відбувається досить швидко, і процесор може обробляти їх відносно послідовно, навіть якщо вони спрацьовують майже одночасно. Механізми пріоритетів та визначеного порядку допомагають забезпечити ефективну обробку переривань та зменшити вплив конкуренції на процесорний час.

1. Так, коли переривання трапляються частіше, ніж вони опрацьовуються, можуть виникнути проблеми. Це може призвести до невідпрацьованих переривань, що, в свою чергу, може мати наслідком втрату даних або некоректну роботу системи.

Для запобігання таким проблемам можна вжити кілька заходів. Один з підходів - збільшення швидкості опрацювання переривань. Це може бути досягнуто шляхом оптимізації програмного коду, використання більш швидкодіючого обладнання або вдосконалення алгоритмів обробки переривань. Збільшення швидкості опрацювання переривань допоможе зменшити кількість невідпрацьованих переривань і підвищити загальну продуктивність системи.

Також можна розглянути підвищення пріоритетності важливих переривань. Це означає, що система надасть пріоритет перериванням, які мають найбільший вплив на роботу системи або забезпечують безпеку даних. Це може бути досягнуто шляхом налагодження пріоритетів на рівні операційної системи або в системному програмуванні.

Однак, варто бути обережним з підвищенням пріоритетності переривань, оскільки неправильне налаштування може призвести до ігнорування менш важливих переривань або виникнення проблем з балансом ресурсів системи.

В кінці кінців, ефективне керування перериваннями вимагає уважного аналізу потреб системи, налаштування пріоритетів та оптимізації опрацювання переривань, щоб забезпечити стабільну та надійну роботу системи у ситуаціях з високим навантаженням переривань.

1. Нижче наведений приклад схеми та коду для гри "Ковбої" на мові Arduino:

Схема:

* Кожен гравець має свою кнопку та світлодіод.
* Кнопки гравців підключені до пінів D2 та D3 мікроконтролера Arduino.
* Світлодіоди гравців підключені до пінів D4 та D5 мікроконтролера Arduino.

Код:

const int player1ButtonPin = 2;

const int player2ButtonPin = 3;

const int player1LedPin = 4;

const int player2LedPin = 5;

bool player1Pressed = false;

bool player2Pressed = false;

unsigned long player1ReactionTime = 0;

unsigned long player2ReactionTime = 0;

void player1Interrupt() {

if (!player1Pressed) {

player1Pressed = true;

player1ReactionTime = millis();

}

}

void player2Interrupt() {

if (!player2Pressed) {

player2Pressed = true;

player2ReactionTime = millis();

}

}

void setup() {

pinMode(player1ButtonPin, INPUT\_PULLUP);

pinMode(player2ButtonPin, INPUT\_PULLUP);

pinMode(player1LedPin, OUTPUT);

pinMode(player2LedPin, OUTPUT);

attachInterrupt(digitalPinToInterrupt(player1ButtonPin), player1Interrupt, FALLING);

attachInterrupt(digitalPinToInterrupt(player2ButtonPin), player2Interrupt, FALLING);

}

void loop() {

if (player1Pressed && player2Pressed) {

if (player1ReactionTime < player2ReactionTime) {

digitalWrite(player1LedPin, HIGH);

digitalWrite(player2LedPin, LOW);

} else {

digitalWrite(player1LedPin, LOW);

digitalWrite(player2LedPin, HIGH);

}

player1Pressed = false;

player2Pressed = false;

}

}

Цей код використовує переривання для виявлення натискання кнопок гравців і порівнює час їх реакції. Якщо гравець 1 натискає кнопку раніше, то його світлодіод (пін D4) запалюється, інакше запалюється світлодіод гравця 2 (пін D5). Коли обидва гравці натискають кнопки, вони засвічуються протягом короткого часу, а потім скидаються до початкового стану.

**Висновок:**

Виконуючи дослідження на тему зовнішніх переривань та розробку гри "Ковбої", можна зробити наступні висновки:

1. При одночасному спрацьовуванні декількох зовнішніх переривань відбувається конкуренція за процесорний час. Потрібно враховувати пріоритети та порядок опрацювання переривань, щоб забезпечити правильну роботу системи.
2. Якщо переривання трапляються частіше, ніж вони опрацьовуються, можуть виникнути проблеми. Невідпрацьовані переривання можуть призвести до втрати даних або некоректної роботи системи. Необхідно забезпечити достатню швидкість опрацювання переривань або використовувати пріоритети для обробки найважливіших подій.
3. Гра "Ковбої" є прикладом застосування зовнішніх переривань для визначення реакції гравців. Вона дозволяє двом гравцям конкурувати між собою, натискуючи кнопки якомога швидше. Використання світлодіодів для показу результату додає елемент візуальної зворотного зв'язку.

Зовнішні переривання є потужним інструментом в мікроконтролерах, який дозволяє реагувати на події в реальному часі. Вони особливо корисні в ігрових додатках та ситуаціях, коли точний час реакції має важливе значення.