Разработка проекта:
Основы параллелизма и прерываний -

Игра в перетягивание каната

Александр Г. Дин
Факультет Электротехники и Компьютерной Инженерии
Университет штата Северная Каролина
8/31/2015, Версия 1.0



**

# Обзор

В этом проекте вы разработаете игру в перетягивание каната, положение каната указывается одним из четырех светодиодов. Каждое нажатие на BTN1 передвигает виртуальный канат на один шаг влево, каждое нажатие на BTN2 передвигает его на один шаг вправо. Игра заканчивается, если канат вытянут достаточно далеко от центрального стартового положения.

# Начальный код

Начните с проекта Demo\_Basic\_Concurrency\_Interrupts и изменяйте его по необходимости.

# Структура программного обеспечения

Ваш код будет содержать одну задачу и одну процедуру обработки прерывания. Вызов функции Run\_Game\_Tasks из main запускает игру.

void Run\_Game\_Tasks(void) {

 while (1) {

 TASK\_Update\_LEDs();

 }

}

# Общая переменная

Для указания положения каната используется общая переменная G\_Position. Она инициализируется 0.

# Управление игрой

Модифицируйте ISR\_PortA\_Change для изменения G\_Position при каждом нажатии кнопки (т.е. при изменении входного бита порта с 0 в 1).

* Уменьшение на 1 переменной G\_Position при нажатии BTN1.
* Увеличение на 1 переменной G\_Position при нажатии BTN2.
* Удостоверьтесь в том, что ваш код корректно обрабатывает случай одновременного нажатия на BTN1 и BTN2.

# Дисплей

Создайте задачу TASK\_Update\_LEDs, которая будет выводить положение виртуального каната на дисплей (см. далее). В этом проекте вам не нужно использовать конечный автомат.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Положение виртуального каната. | LD1 | LD2 | LD3 | LD4 |
| -3 | Вкл |  |  |  |
| -2 | Вкл | Вкл |  |  |
| -1 |  | Вкл |  |  |
| 0 |  | Вкл | Вкл |  |
| 1 |  |  | Вкл |  |
| 2 |  |  | Вкл | Вкл |
| 3 |  |  |  | Вкл |

Для включения или выключения определенного светодиода вызывается функция Set\_LEDs (см. ниже).

* Передача аргумента 1 включает соответствующий светодиод.
* Передача аргумента 0 выключает соответствующий светодиод.

void Set\_LEDs(int L1, int L2, int L3, int L4)

{

 LD1\_PORT\_BIT = L1;

 LD2\_PORT\_BIT = L2;

 LD3\_PORT\_BIT = L3;

 LD4\_PORT\_BIT = L4;

}

# Завершение игры

Игра заканчивается, если положение виртуального каната становится <= -4 (побеждает левая сторона) или >= 4 (побеждает правая сторона).

Измените функцию Scan\_LEDs: она должна получать целый аргумент dir\_lr.

* Если dir\_lr равно 1, то святящиеся светодиоды должны быстро перемещаться слева направо в течении нескольких секунд, это будет указывать на победу правой стороны.
* Если dir\_lr равно 0, то святящиеся светодиоды должны быстро перемещаться справа налево в течении нескольких секунд, это будет указывать на победу левой стороны.

Попробуйте использовать задержку 1000000 (один миллион) в функции Delay демонстрационного примера.

# Подготовка отчета

Представьте ваш завершенный проект MPLAB X преподавателю и согласуйте с ним проведение демонстрации.