

Объект "mailboxes"

Объект создан для оффлайн-тестирования межагентного взаимодействия (МAB) роботов в среде программирования TrikStudio (на контроллерах Трик), но при необходимости его использование возможно и в других системах. Объект написан на JavaScript (ECMAScript 5.0 в нотации для контроллеров Трик).

Основная идея заключается в том, чтобы была возможность тестировать МAB оффлайн, а при запуске программы на реальных роботах менять только название основного объекта **mailboxes** -> **mailbox**, без изменения всего кода программы (или минимум). В связи с чем названия основных методов MAILBOXES аналогичны методам в mailbox:

`.send(), .receive(), .hasMessages(), .myHullNumber()`

Каждый робот запускается в отдельной программе TrikStudio.

До обращения к объекту надо объявить переменные:

robotHulls = [1, 20, 44] - массив с "бортовыми номерами" роботов, участвующих во взаимодействии
robotNumber = 1 - «бортовой номер» текущего робота (должен меняться в программе у каждого робота)

Описание примера работы МAB (mailboxes_example.js):

1. При необходимости меняются пути:
 - a. К расположению «файлообменника», по умолчанию путь "c:\temp\mailboxes" (свойство 'storagePath')
 - b. При необходимости меняется путь к файлу "mailboxes.js"
 - c. **Все пути указываются через прямой слэш '/' или двойной обратный слэш '\\'**
2. Запускаются отдельные копии TrikStudio (TS) в количестве = количеству роботов.
3. В каждой копии TS открывается скрипт 'mailboxes_example.js', где меняется **ТОЛЬКО** номер текущего робота - переменная *robotNumber*.
4. На роботе с бортовым номер «1» нажимаем кнопку «стрелка вверх»
5. «Робот 1» отправляет сообщение «Роботу 2»
6. «Робот 2» после получения сообщения проезжает вперед 2 клетки и отправляет сообщение «Роботу 3»
7. «Робот 3» после получения сообщения проезжает вперед 2 клетки и отправляет сообщение «Роботу 1»
8. «Робот 1» после получения сообщения проезжает вперед 2 клетки и отправляет сообщение всем роботам
9. Все роботы после получения сообщения одновременно едут вперед 2 клетки

Happy end ☺

Необходимые файлы:

mailboxes.js	объект для оффлайн-«общения» роботов, ссылка
mailboxes_example.js	основная программа для демонстрации примера машинного взаимодействия, ссылка
mav.mov	видео примера работы скриптов выше, ссылка
mailboxes_aio.js	объект mailboxes и демокод в одной программе, ссылка

Свойства объекта *mailboxes*:

Наименование	Тип	Содержание
storagePath	Строка	Локальный / сетевой путь до размещения рабочих файлов
fileExt	Строка	Расширение имен файлов для обмена сообщениями
myFile	Строка	Полное имя файла для текущего робота (путь + название + расширение)

Методы (функции) объекта *mailboxes*:

Наименование	Параметры		Содержание
	Входные	Выходные	
init	-	-	1. Формирует имя файла для текущего робота (на котором запущена программа) и записывает его в свойство myFile 2. Удаляет файлы, соответствующие номерам роботов (номера берутся из глобального массива robotHulls)
myHullNumber	-	<i>number</i>	Возвращает «бортовой номер» текущего робота (по сути значение переменной myHullNumber. Сделано для совместимости с основным объектом <mailbox>.
hasMessages	-	<i>boolean</i>	Проверяет наличие сообщений для текущего робота и возвращает <true> или <false>
receive	-	<i>array</i>	Получает сообщения для текущего робота. Возвращает одномерный массив, где каждый элемент – строка из файла. Если сообщений нет, то будет ждать до тех пор, пока не появится хотя бы одно сообщение. Для обработки сообщений их необходимо «развернуть» из формата JSON командой JSON.parse()
send	robots (<i>array</i>), msg (<i>string/array/object</i>)	-	Формирует и отправляет указанным роботам сообщения. Если не указаны номера роботов, сообщения отправляется всем (из массива номеров robotHulls), примеры: ➤ [] – сообщение отправится всем роботам ➤ 2 или [2] – сообщение отправится только роботу с бортовым номером 2 ➤ [2, 3] – сообщение отправится роботам с номерами 2 и 3 Сообщение отправляется в формате JSON В конце сообщения добавляется перевод строки '\n'
writeFile	file (string), msg(string)	-	Записывает сообщение <msg> в файл <file>.
clearFile	file (string)	-	Удаляет файл <file>. Создает новый файл <file>.
readFile	file (string)	data (array)	Считывает все данные из файла <file>. Обнуляется файл <file>. Возвращает массив с данными, где каждый элемент - строка из файла.