Как договаривались ручной цикл только один и находится в pistonTasks[TASK\_NUM - 1] (TASK\_NUM = 3 сейчас).

В ручном цикле максимально STEP\_PER\_TASK\_NUM (сейчас 10) шагов, при этом (STEP\_PER\_TASK\_NUM – 1) шаг всегда должен быть равен NULL (pistonTasks[2]->step[9].poz = NULL)

Настраивается 2мя функциями:

bool PL\_addTaskLine (piston\_task\_t\* task,task\_line\_t tl);

bool PL\_deleteTaskLine (piston\_task\_t\* task, uint8\_t line);

bool PL\_modTaskLine(piston\_task\_t\* task, uint8\_t line, task\_line\_t tl)

1. PL\_addTaskLine

На входе указатель на массив ручного цикла (&pistonTasks[2]) и строчку с самим заданием (положение и время в этом положении).

Возвращает true, если все прошло успешно, и false, если нет (кончилось место в массиве).

1. PL\_deleteTaskLine

На входе указатель на массив ручного цикла (&pistonTasks[2]) и номер строки для удаления.

Возвращает true, если все прошло успешно, и false, если нет (пустой массив или номер строки больше, чем есть элементов).

1. bool PL\_modTaskLine(piston\_task\_t\* task, uint8\_t line, task\_line\_t tl)

На входе указатель на массив ручного цикла (&pistonTasks[2]), номер строки для модификации и строчку с самим заданием (положение и время в этом положении)..

Возвращает true, если все прошло успешно, и false, если нет (номер строки больше, чем есть элементов).

Пример использования:

task\_line\_t tl; //Буфер для хранения строчки задания

tl.poz = &pistonPositions.closedPosition;

tl.secPause = 10;

result = PL\_addTaskLine(&pistonTasks[2], tl); //Запись строчки

tl.poz = &pistonPositions.backwash; //Модификация строчки задания

tl.secPause = 30;

result = PL\_addTaskLine(&pistonTasks[2], tl); //Запись второй строчки задания

result = PL\_deleteTaskLine(&pistonTasks[2], 0); //Удаление первой строчки задания

После окончания заполнения задания нужно вызвать – copyTasksToFlash();

Напомиаю, что последний шаг всегда равен NULL для работы логики. Обход всех элементов в массиве – while(pistonTasks[2]->step[i].poz != NULL){

I++;

……

}