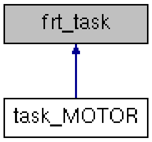
task_MOTOR Class Reference

```
#include <task_MOTOR.h>
```

Inheritance diagram for task_MOTOR:



Public Member Functions

	task_MOTOR (const char *, unsigned portBASE_TYPE, size_t, emstream *)
void	**run** (void)

Protected Types

enum **MOTOR_states** { **FORWARD**, **MOTOR_STOP**, **BACKWARD**, **TURN** }

Protected Attributes

uint8_t **ctr**

Loop counter.

Detailed Description

This task controls the motors based on the input from the various sensors on the sumo robot. The robot only uses the states for moving forward and turning in one direction due to issues with a motor driver failing.

Constructor & Destructor Documentation

◆ task_MOTOR()

```
task_MOTOR::task_MOTOR ( const char *           a_name,
                         unsigned portBASE_TYPE a_priority,
                         size_t                 a_stack_size,
                         emstream *             p_ser_dev
                       )
```

This constructor creates a new motor control task. Its main job is to call the parent class's constructor which does most of the work.

Parameters

a_name	A character string which will be the name of this task
a_priority	The priority at which this task will initially run (default: 0)
a_stack_size	The size of this task's stack in bytes (default: configMINIMAL_STACK_SIZE)
p_ser_dev	Pointer to a serial device (port, radio, SD card, etc.) which can be used by this task to communicate (default: NULL)

Member Function Documentation

◆ run()

```
void task_MOTOR::run ( void )
```

This method is called by the RTOS once to run the task loop for ever and ever.

This task controls the motors based on the input from the various sensors on the sumo robot. The robot only uses the states for moving forward and turning in one direction due to issues with a motor driver failing.

The documentation for this class was generated from the following files:

- Source/**task_MOTOR.h**
- Source/**task_MOTOR.cpp**

Generated by doxygen 1.8.16