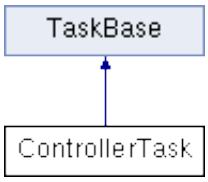
ControllerTask Class Reference

Implements a task to control the system. More...

`#include <ControllerTask.h>`

Inheritance diagram for ControllerTask:



Public Member Functions

	ControllerTask (const char *a_name, unsigned portBASE_TYPE a_priority, size_t a_stack_size, **emstream** *p_ser_dev) Construct a BallPosition task. More...
void	**run** (void) The run method of the Controller task that is repeatedly called by the RTOS scheduler. More...

▸ **Public Member Functions inherited from TaskBase**

Additional Inherited Members

▸ **Static Public Member Functions inherited from TaskBase**

▸ **Protected Member Functions inherited from TaskBase**

▸ **Protected Attributes inherited from TaskBase**

Detailed Description

Implements a task to control the system.

This class is an extension of **TaskBase**. The purpose of the class is to determine the voltage input to the motor in response to state feedback from the ball and beam sensor measurements.

Constructor & Destructor Documentation

◆ ControllerTask()

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

Construct a BallPosition task.

Constructor which creates and initializes a controller task object.

This constructor sets up the task name, priority, stack size, and serial stream.

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)

This constructor creates a FreeRTOS task with the given task run function, name, priority, and stack size. Its purpose is to implement the state space control algorithm with established gains.

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 ControllerTask::run (void) virtual

The run method of the Controller task that is repeatedly called by the RTOS scheduler.

The `run()` function for the Controller task.

This method is called by the RTOS scheduler. The function reads the system system states and establishes the required PWM signal to send to the motor. NOTE: The control algorithm is FULL STATE FEEDBACK

Implements **TaskBase**.

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

- DoxygenFiles/**ControllerTask.h**
- DoxygenFiles/ControllerTask.cpp
