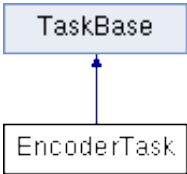


EncoderTask Class Reference

Implements a task to determine the beam angle. [More...](#)

```
#include <EncoderTask.h>
```

Inheritance diagram for EncoderTask:



Public Member Functions

EncoderTask (const char *a_name, unsigned portBASE_TYPE a_priority, size_t a_stack_size, **emstream** *p_ser_dev)
Construct an Encoder task. [More...](#)

void **run** (void)
The run method of the Encoder 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 determine the beam angle.

This class is an extension of **TaskBase**. The purpose of the class is to determine the beam angular position and velocity.

Constructor & Destructor Documentation

◆ EncoderTask()

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

Construct an Encoder task.

Constructor which creates and initializes an Encoder 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 determine the angle of the beam based on the encoder sensor measurements.

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 EncoderTask::run ( void )
```

virtual

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

The **run()** function for the Encoder task.

This method is called by the RTOS scheduler. The function implements a modulo positioning algorithm to account for discontinuity in the absolute position output from the encoder sensor. The encoder counts are converted to angle measurements in radians and angular velocity in rad/s. Shared variables are updated after the calculations are performed.

Implements **TaskBase**.

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

- DoxygenFiles/**EncoderTask.h**
- DoxygenFiles/EncoderTask.cpp