driver.py

AUTHORS

JAY CONVERTINO

DATES

2025/03/11

INFORMATION

Brief

Bus Driver for Analog Devices uP

License MIT

upMaster

Drive slave devices over the uP bus

FUNCTIONS

__init__

```
def __init__(
self,
entity,
name,
clock,
resetn,
                                                                    *
args,
                                                                   **
kwargs
)
```

Setup defaults and call base class constructor.

read

```
async def read(
self,
address
)
```

Read from a address and return data

write

```
async def write(
self,
address,
data
)
```

Write to a address some data

_check_type

```
def _check_type(
self,
trans
)
```

Check and make sure we are only sending 2 bytes at a time and that it is a bytes/bytearray

_run_write

```
async def _run_write(
self
)
```

method for write thread

_run_read

```
async def _run_read(
self
)
```

method for read thread

upEchoSlave

Respond to master reads and write by returning data, simple echo core.

FUNCTIONS

__init__

```
def __init__(
self,
entity,
name,
clock,
resetn,
numreg
=
256,
*
args,
**
kwargs
)
```

Setup defaults and call base class constructor.

_check_type

```
def _check_type(
self,
trans
)
```

Check and make sure we are only sending a type of uptrans.

_run_write

```
async def _run_write(
self
)
```

method for write thread

_run_read

```
async def _run_read(
self
)
```

method for read thread