```python
from XRPLib.encoded_motor import EncodedMotor
from XRPLib.pid import PID
from XRPLib.timeout import Timeout
import time
import math

import uasyncio as asyncio

class Pump:
    @classmethod
    def get_default_pump(cls):
        motor = EncodedMotor.get_default_encoded_motor(4)
        return Pump(motor, 1.5, 10, -1)

    def __init__(self, motor, turns_to_ml, purge_ml, dispense_direction):
        self.motor = motor
        self.turns_to_ml = turns_to_ml
        self.purge_ml = purge_ml
        self.dispense_direction = dispense_direction

    def stop(self):
        self.motor.set_effort(0)

    async def turn(self, turns):
        start = self.motor.get_position()
        print(f"Start pos: {start}")
        self.motor.set_effort(self.dispense_direction)

        while abs(self.motor.get_position() - start) < turns:
            print(f"Current pos: {self.motor.get_position()}")
            await asyncio.sleep(0.1)

        self.motor.set_effort(0)

    async def water(self, ml):
        turns = ml * self.turns_to_ml
        await self.turn(turns)


```