```cpp
// gyro.hpp: Gyro utilities that provide ease of access to the robot's rotation
// and manipulation of this rotation
// Copyright (C) 2017 Ethan Wells
//
// This program is free software: you can redistribute it and/or modify
// it under the terms of the GNU Lesser General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public License
// along with this program.  If not, see <http://www.gnu.org/licenses/>.

#pragma once
#include "pid.hpp"

namespace gyro {
  /** A driving that can allow an arc, or keep the robot straight */
  class drive {
  public:
    /** A reference to the gyro which will be used to get values from */
    sensors::gyro_t* gyro;
    /** The ideal heading of the robot (is absolute)*/
    int heading;
    /** The urgency/agressiveness of the arc */
    float urgency;
    /** Turn the arc off */
    void off(void);
    /** Use to initialize and run the task */
    drive(int heading, float urgency = 15.f, bool absolute = false,
          sensors::gyro_t* gyro = &sensors::gyro, unsigned int tolerance = 3);

  private:
    /** The task that runs, keeping the robot straight */
    void task(void* none);
    /** The initial  heading, as opposed to the ideal heading */
    int iHeading;
    /** TaskHandle for the gyro heading task */
    TaskHandle handle;
    /** The internal variable used for changing the pid values */
    float changer;
    /** Whether it is on or not */
```

```
    bool on;
    /** The tolerance for turning */
    int tolerance;
  }; // class drive

} // namespace gyro
```