```cpp
// lift.hpp: Header file for utilities relating to the lift
// 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/>.

#ifndef LIFT_HPP
#define LIFT_HPP

#include "drive.hpp"

/** Contains everything relating to the drive */
namespace lift {
  /** Class for a side of the drive */
  struct side_t {
    /** Top motor on the the side */
    motor_t topM;
    /** Middle motor on the side */
    motor_t midM;
    /** Bottom motor on the side */
    motor_t lowM;
    /** Sets all motors on the side to the given power */
    void set(int power);
    /** A pointer to the sensor on the side */
    sensors::pot_t* sensor;
  }; // struct side_t

  /** Positions of the lift */
  typedef enum {
    bottom = 5,
    mobile = 60,
    one    = 100,
    two    = 230,
    three  = 450,
  } position;
```

```
  extern double inch;
  /** The left side of the lift */
  extern side_t left;
  /** The right side of the lift */
  extern side_t right;
  /** Sensor on the lift */
  extern sensors::pot_t* sensor;
  /** The value where the lift will stay up with a standard load */
  static const char lockN = 17;
  /** Any value below this point will result in the lift being set to 0 */
  static const int threshold = 150;

  /** Set the lift at their requested powers */
  void set(int power);

  /** Default value for the lift to be set at when it is no tin use */
  void lock(void);

  /** Initialize the drive subsystem */
  void init(void);

  /** p control for the lift */
  void to(position pos = bottom, int int_pos = -1, int tolerance = 50);

  /** Lift control that should be used in a while loop */
  void control(void);

} // namespace lift

#endif /* end of include guard: LIFT_HPP */
```