```c
/**
 * @file line.h
 * @brief Utilities for the three line sensors
 * Copyright (C) 2017 Ethan Wells
 *
 * This program is free software: you can redistribute it and/or modify it
 * under the terms of the GNU 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 General Public License for more
 * details.
 *
 * You should have received a copy of the GNU General Public License along
 * with this program. If not, see <https://www.gnu.org/licenses/>
 */

#pragma once

#include "robot.h"

#define gline(index) (line[index].value)

/*
 * The distance between line sensors, the first number is inches and everything
 * else converts -> ticks
 */
static const double lineDistance = 80.25791219881197;

typedef enum {
        LEFT  = 0,
        RIGHT = 1,
} Side;

/**
 * @brief Get the current Side. Really simple.
 *
 * @return the current Side, LEFT or RIGHT
 */
Side        getSide();

/**
 * @brief Take and delete a mutex. If it can't be taken, don't delete it.
 *
```

```
 * @param m the Mutex to take (then delete)
 * @param blockTime the maximim time to wait before giving up
 *
 * @return Whether or not the mutex was taken
 */
inline bool mutexTakeDelete(Mutex m, unsigned long blockTime) {
        if (!mutexTake(m, blockTime)) {
                return false;
        }

        mutexDelete(m);
        return true;
} // mutexTakeDelete

/**
 * @brief Calculate the angle of the robot based on the positions of the left
 * and right drive from when each line sensor hit the line
 *
 * @param p an array of the 4 positions, in the order: L1, R1, L2, R2
 * @param o which line sensor hit the line first
 *
 * @return the angle, in degrees clockwise, of the robot relative to the line.
 * If something goes wrong, it will result 0 (I think. Don't quote me on that)
 */
int getAngleFP(int p[4], Side o);

/**
 * @brief Get the angle from the upcoming line when it is hit and store it
 *
 * @param store a pointer to where the angle is to be stored
 * @param m the Mutex to use, if one already exists. If not, use NULL
 * @param maxTime the maximum amount of time the get can take
 *
 * @return A Mutex that will be released when the angle is stored
 */
Mutex angleFromUpcomingLine(int          *store,
                            Mutex         mutex,
                            unsigned long maxTime);
```