```c
/**
 * @file line.c
 * @brief line sensor stuff to make life easier
 * 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/>
 */

#include "../include/line.h"

typedef struct aful_t {
        int          *result;
        Mutex         mutex;
        unsigned long max;
} Aful;

Side getSide() {
        return line[0].value ? LEFT : RIGHT;
} /* getSide */

int getAngleFP(int p[4], Side s) {
        return atan((double)(s ? p[3] - p[1] : p[4] - p[2]) / lineDistance)
               * 180 / M_PI / 2 / (s ? -1 : 1);
} /* getAngleFP */

void updateLinesDrive() {
        for (int i = 0; i < 3; i++) {
                sensorRefresh(&line[i]);
        }

        for (int i = 0; i < 2; i++) {
                sensorRefresh(drive[i].sensor);
        }
} /* updateLinesDrive */
```

```
Task angleFromUpcomingLineT(void *aful) {
        Aful *a = (Aful *)aful;

        while (!(line[0].value + line[2].value)) {
                updateLinesDrive();
                delay(20);
        }

        int  p[4] = { drivePos(0), drivePos(1) };
        Side s    = getSide();

        if (s == LEFT) {
                while (!line[2].value) {
                        updateLinesDrive();
                        delay(20);
                }
        } else {
                while (!line[0].value) {
                        updateLinesDrive();
                        delay(20);
                }
        }

        p[2] = drivePos(0);
        p[3] = drivePos(1);

        *(a->result) = (millis() <= a->max) ? getAngleFP(p, s) : 0;
        mutexGive(a->mutex);
        free(aful);
        taskDelete(NULL);
} /* angleFromUpcomingLineT */

Mutex angleFromUpcomingLine(int *store, Mutex m, unsigned long maxTime) {
        Aful *a = new(Aful);

        a->result = store;
        a->mutex  = m ? m : mutexCreate();
        a->max    = millis() + maxTime;
        mutexTake(a->mutex, -1);
        GO(angleFromUpcomingLineT, a);
        return a->mutex;
} /* angleFromUpcomingLine */
```