

```

/**
 * @file auto.h
 * @brief Structures and information pertaining to autonomous that is needed in
 * places other than auto.c
 * 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 "../include/robot.h"

#define MAX_AUTON 4

typedef struct Auton {
    const char *name;
    const char *sensorName;
    Sensor **sensor;
    void (*execute)();
} Auton;

typedef enum Direction {
    dUp = -3,
    dDown = -2,
    dLeft = 1,
    dRight = -1,
    dIn = 2,
    dOut,
} Direction;

typedef struct Triple {
    int a;
    int b;
    int c;
}

```

```

} Triple;

/**
 * A list of the autonomouses/LCD menus
 */
extern Auton autons[MAX_AUTON + 1];
/**
 * The autonomous, as selected by the LCD menu, to run
 */
extern int    selectedAuton;

/**
 * @brief Bring the drive to a specific position
 *
 * @param l the left position
 * @param r the right position
 * @param until the maximum amount of time this can take
 */
void driveToPosition(int          l,
                    int          r,
                    unsigned long until);

/**
 * @brief Bring the drive to a specific position while attempting to maintain an angle
 *
 * @param l the left position
 * @param r the right position
 * @param a the angle to maintain
 * @param until the maximum amount of time this can take
 */
void driveToPositionAngle(int          l,
                        int          r,
                        int          a,
                        unsigned long until);

void driveToPositionAngleT(void *triple);

/**
 * Use PID to turn to a specific angle
 *
 * @param angle the angle to turn to
 * @param until the max amount of time this can take
 */
void turnTo(int          angle,
            unsigned long until);

```