```c
/**
 * @file opcontrol.c
 * @brief Controls what happens in operator control
 * 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/robot.h"

#define INTAKE_HOLD 300

extern bool isAuto;

int digital(unsigned char joyNum,
            unsigned char channel,
            unsigned char b1,
            unsigned char b2) {
        return joystickGetDigital(joyNum, channel, b2) * -1 +
               joystickGetDigital(joyNum, channel, b1) * 1;
} /* digital */

void moveDrive();
void moveIntake();
void moveLift();

void operatorControl() {
        #ifdef DEBUG_MODE
                printf("Starting Driver Control...\n");
        #endif /* ifdef DEBUG_MODE */
        reset();
        update();
        isAuto = false;

        // liftSettings.target  = lift.sensor->value;
```

```
        while (true) {
                moveDrive();
                moveIntake();

                moveLift();
                update();

                delay(20);
        }
} /* operatorControl */

void moveDrive() {
        drive[0].power = deadBand(joystickGetAnalog(1, 3), 10) +
                         127 * digital(1, 7, JOY_UP, JOY_DOWN) +
                         127 * digital(1, 7, JOY_RIGHT, JOY_LEFT);
        drive[1].power = deadBand(joystickGetAnalog(1, 2), 10) +
                         127 * digital(1, 8, JOY_UP, JOY_DOWN) +
                         127 * digital(1, 8, JOY_LEFT, JOY_RIGHT);
} /* moveDrive */

void moveIntake() {
        intake[0].power = 127 * digital(1, 5, JOY_UP, JOY_DOWN);
        intake[1].power = 127 * digital(1, 5, JOY_UP, JOY_DOWN);
} /* moveIntake */

void moveLift() {
        lift.power = 127 * digital(1, 6, JOY_UP, JOY_DOWN);
} /* moveLift */
```