

```

/**
 * @file init.c
 * @brief Perform initialization and start handler tasks
 * 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"

void initializeIO() {
    watchdogInit();
}

float zeroRecalc(int x) {
    return 0;
}

float intakeRecalc(int x) {
    return .63f * (float)x;
}

/**
 * Notify both through the terminal and an lcd
 *
 * @param buffer the text to display
 */
void notice(const char *buffer) {
    #ifdef DEBUG_MODE
        print(buffer);
    #endif
    lcdSetText(uart1, 2, buffer);
    delay(5);
} /* notice */

```

```

void init() {
    // LCD initialization
    lcdInit(uart1);
    lcdSetBacklight(uart1, true);

    #ifdef DEBUG_MODE
        print("\nInitializing... ");
    #endif
    lcdSetText(uart1, 1, "Initializing...");

    // Set up the analog sensors
    gyro      = newGyro(1, true, 198);
    gyro.child = new(Sensor);
    *gyro.child = newGyro(2, true, 197);
    notice("gyroscopes, ");
    for (int i = 0; i < 3; i++) {
        line[i] = newAnalog(i + 6, false);
        line[i].inverted = true;
    }
    notice("line sensors");

    // Set up the digital sensors
    Sensor *intakeCoder[2] = { new(Sensor), new(Sensor) };
    *intakeCoder[0]        = newQuad(7, 6, false);
    intakeCoder[0]->recalc = &zeroRecalc;
    *intakeCoder[1]        = newQuad(2, 1, true);
    intakeCoder[1]->recalc = &zeroRecalc;;
    Sensor *driveCoder[2] = { new(Sensor), new(Sensor) };
    *driveCoder[0]        = newQuad(4, 5, true);
    notice("left drive quad, ");
    *driveCoder[1]        = newQuad(8, 9, true);
    notice("right drive quad, ");

    // Initialize and set up all of the motors, servos, etc
    lift      = motorCreate(5, true);
    lift.child = new(Motor);
    *lift.child = motorCreate(6, false);
    notice("lift motors, ");

    intake[0]      = motorCreate(3, true);
    intake[0].recalc = &intakeRecalc;
    intake[0].sensor = intakeCoder[0];
    intake[1]      = motorCreate(8, false);
    intake[1].recalc = &intakeRecalc;
    intake[1].sensor = intakeCoder[1];
    notice("mobile goal motors, ");
}

```

```

drive[0]          = motorCreate(2, true);
drive[0].child    = new(Motor);
*drive[0].child    = motorCreate(4, true);
drive[0].sensor    = driveCoder[0];

drive[1]          = motorCreate(9, false);
drive[1].child    = new(Motor);
*drive[1].child    = motorCreate(7, false);
drive[1].sensor    = driveCoder[1];
notice("drive motors, ");

lcdSetText(uart1, 1, "Ready!");
#ifdef DEBUG_MODE
    print("\n\n");
#endif
setTeamName("709S");
notice("done!");

// Start the LCD task
LCDHandle = GO(lcdTask, NULL);
} /* init */

```