```
// Task to run catapult
void run_catapult(void* params) {

    double catSpeed = 0;
    aimFire = false;
    bool justFired = false;
    double catPos = 0;
    double catSeek = -1;
    double catNextPos = 0;

    while (true) {

        // Start with catapult not moving
        catSpeed = 0;

        // Calculate current catapult position
        catPos = (cat_1.get_position() + cat_1.get_position())/2;

        double relativeAngle;

        // State machine for fireing
        switch (fireState) {

            case 1:
                // First step, run catapult down full speed
                catSpeed = 127;
                catSeek = -1;
                // If we want to aim
                if (aimFire) {
                    // Get the angle from camera
                    relativeAngle = getRelativeAngle();
                    // And turn that angle
                    turnRelative(relativeAngle,-1);
                }
                // Once we hit the limit switch
                if (cat_Limit.get_value()) {
                    // Move to next step
                    fireState++;
                    // Tare motors
                    cat_1.tare_position();
                    cat_2.tare_position();
                }
                break;

            case 2:
                // Continue to run catapult down
                catSpeed = 127;
                catSeek = -1;
                // Continue to aim
                if (aimFire) {
                    relativeAngle = getRelativeAngle();
                    turnRelative(relativeAngle,-1);
                }
                // Once we are no longer pressing the switch
                // (Catapult has fired)
                if (!cat_Limit.get_value()) {
                    // Move onto next step
                    fireState++;
                    // Stop turning
                    driveStop();
                    aimFire = false;
                    // For autonomous, next command is available
                    nextCommand = true;
                }
                break;

            case 3:
                // We've fired, so draw catapult back to hold position
                catSeek = CAT_HOLD_POS;
                // If we hit the switch, then we're done
                // case 4 doesn't exist, so we will do nothing
                if (cat_Limit.get_value()) {
                    fireState++;
                }
                break;

            case 5:
                // If we're here, just hold catapult at 0°
                catSeek = 0;
                break;

            case 10:
                // This is for when we want to draw back the catapult
                 w/o firing
                // Move catapult back slowly
                catSpeed = 60;
                catSeek = -1;
                // Once we hit the limit switch
                if (cat_Limit.get_value()) {
                    // Move to next step (to do nothing)
                    fireState++;
                    // And hold the catapult at current position
                    catSeek = catPos;
                }
                break;
            case 11:
                // Do nothing
                break;

            case 20:
                // This is for when we want to aim before we fire
```

```
        // Find angle to aim
        relativeAngle = getRelativeAngle();
        // Aim at that angle
        turnRelative(relativeAngle,-1);
        // Once we're aimed, go to step 1
        if (abs(relativeAngle) < 1)
            fireState = 1;
        break;
    default:
        break;
}

// Manual button to move catapult up
if (controller.get_digital(BTN_CAT_UP)) {
    // Clear auto-flags and set speed
    catSpeed = -127;
    fireState = -1;
    catSeek = -1;
}
// Manual button to move catapult down
if (controller.get_digital(BTN_CAT_DOWN)) {
    // Clear auto-flags and set speed
    catSpeed = 127;
    fireState = -1;
    catSeek = -1;
}

// Button to fire w/o aiming
if (controller.get_digital(BTN_FIRE)) {
    // Set flags, go to fireState 1
    aimFire = false;
    fireState = 1;
}
// Button to fire w/ aiming
if (controller.get_digital(BTN_FIRE_AIM)) {
    // Set flags, go to fireState 20
    aimFire = true;
    fireState = 20;
}

// Button to abort auto-funtion
if (controller.get_digital(BTN_ABORT)) {
    // Clear auto-flags
    fireState = -1;
    catSeek = -1;
}

// Button to draw catapult w/o firing
if (controller.get_digital(BTN_TOGGLE)) {
    // Go to fireState 10
    fireState = 10;
}


// If we want to seek a position, run motors proportional to
 distance
if (catSeek >= 0) {
    catSpeed = (catSeek - catPos);
}


// Set motors on catapult
cat_1.move_voltage(catSpeed * 12000 / 127);
cat_2.move_voltage(catSpeed * 12000 / 127);


pros::delay(20);   // don't hog cpu
    }
}
```