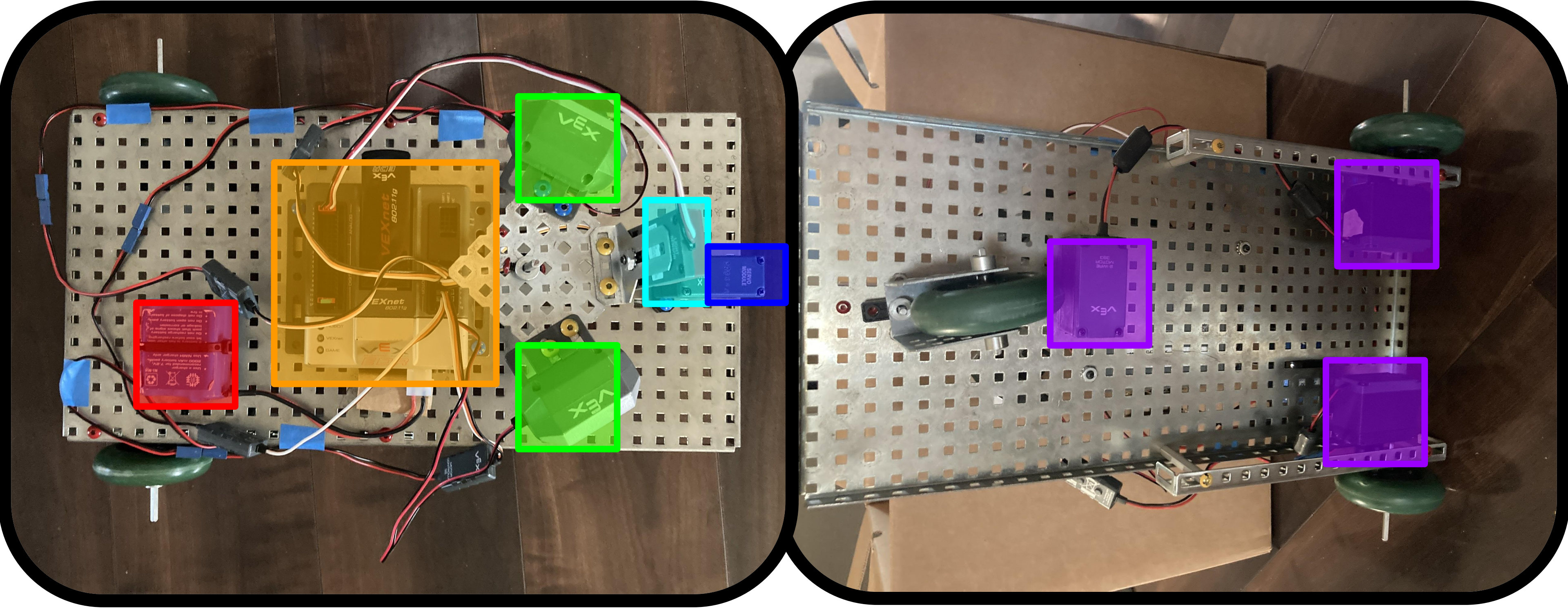
Spotlight Sidekick

Prompt #9 - Searchlight

Problem Statement

1. Design and build a car which can find people via a swiveling searchlight.
2. Searchlights should be able to toggle on and off.
3. Include a system which will detect and report when it finds a person or object.



Diagram

Key:

🟥 Battery

🟧 Vex Cortex

🟩 Flashlights

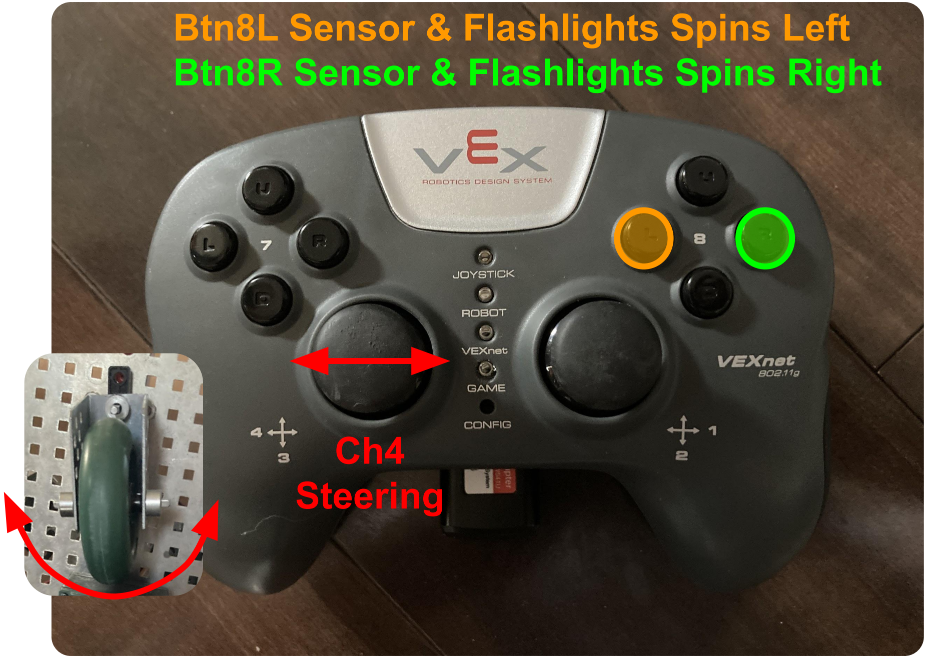
🟦 Light Sensor

🟦 Servo Motor

🟪 3 Wire Motors

Diagram created by Pedro Aguilar

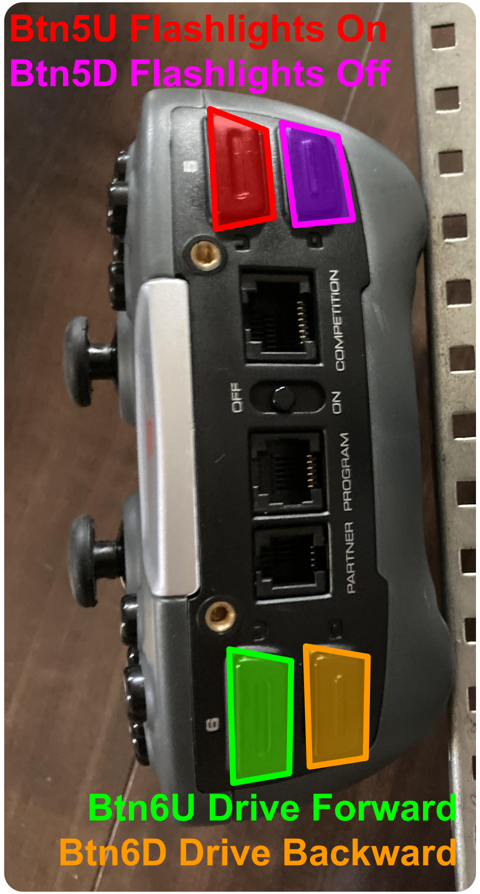
Controls



Btn8L Sensor & Flashlights Spins Left

Btn8R Sensor & Flashlights Spins Right

Ch4 Steering



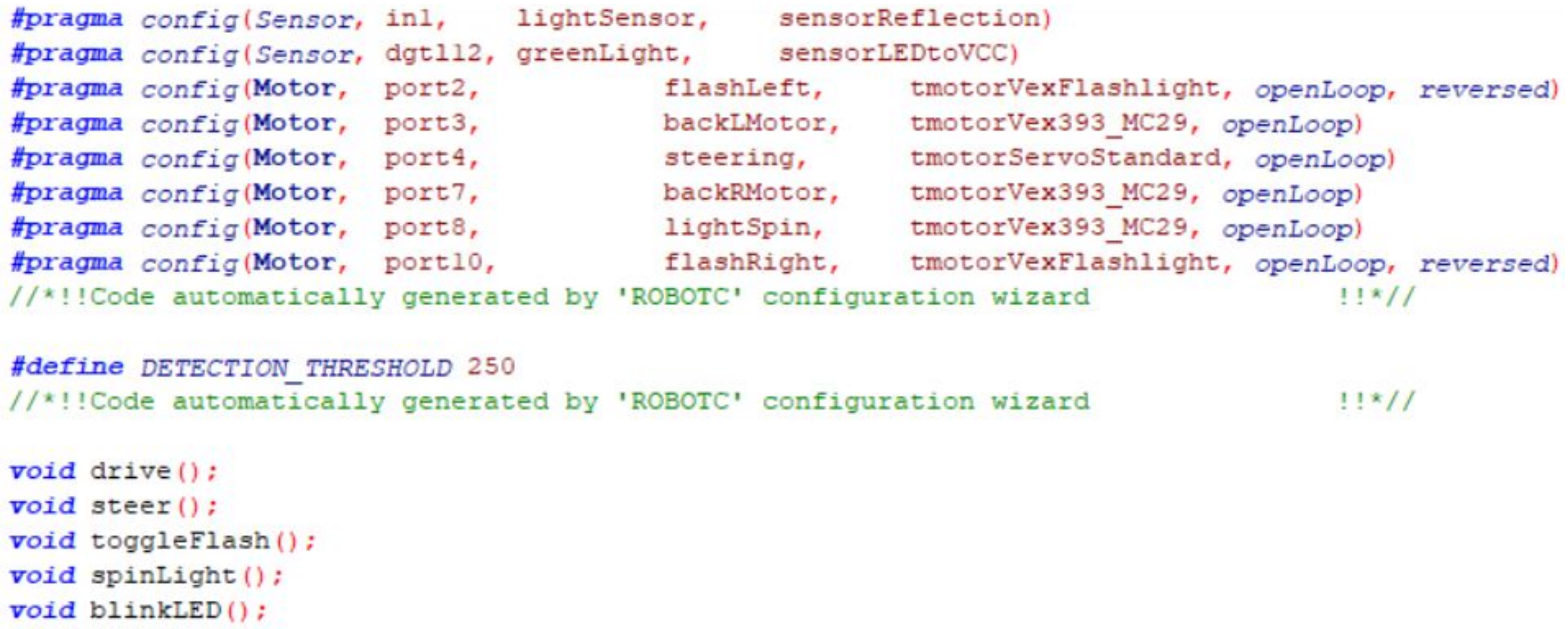
Btn5U Flashlights On

Btn5D Flashlights Off

Btn6U Drive Forward

Btn6D Drive Backward

Configuration

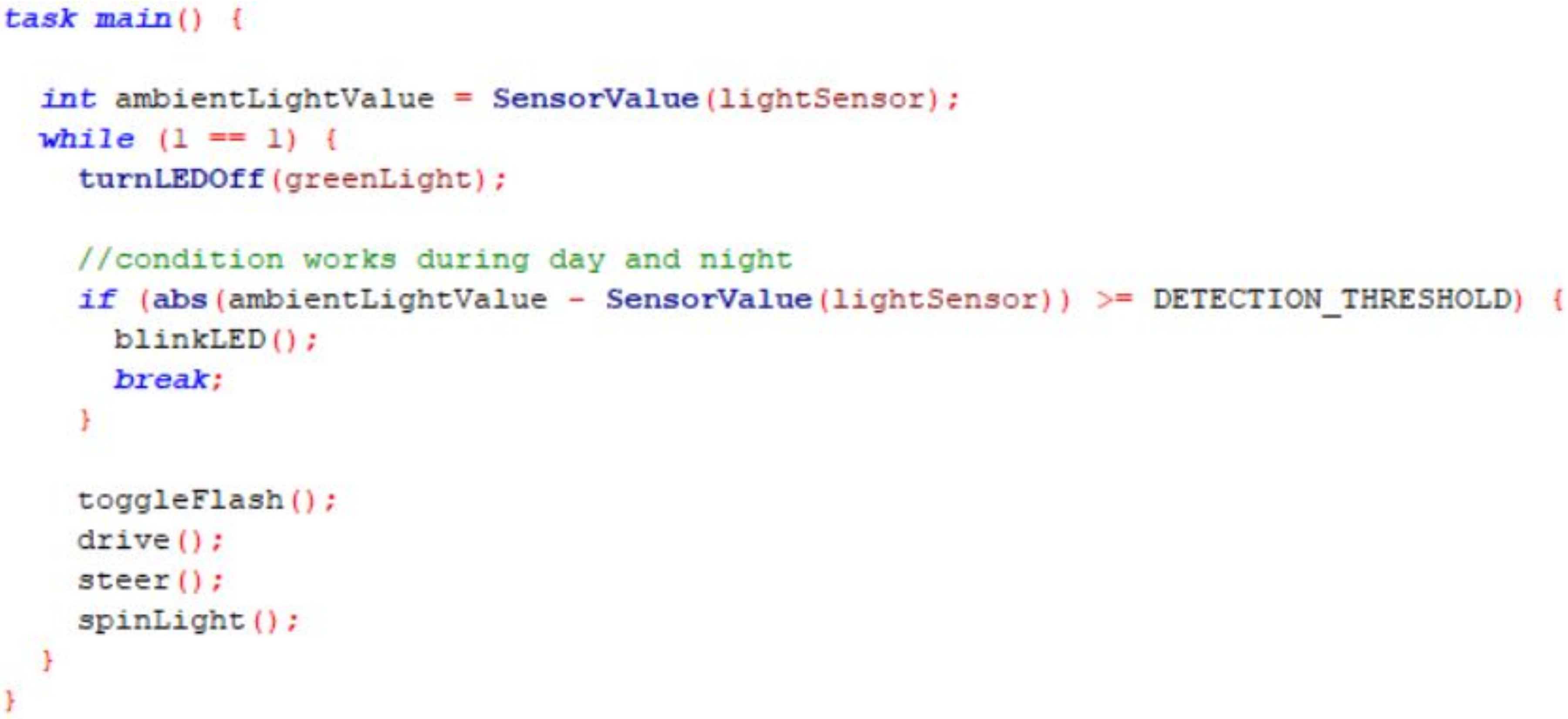


```
#pragma config(Sensor, in1,    lightSensor,    sensorReflection)
#pragma config(Sensor, dgtl12, greenLight,     sensorLEDtoVCC)
#pragma config(Motor,  port2,           flashLeft,     tmotorVexFlashlight, openLoop, reversed)
#pragma config(Motor,  port3,           backLMotor,    tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port4,           steering,      tmotorServoStandard, openLoop)
#pragma config(Motor,  port7,           backRMotor,    tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port8,           lightSpin,     tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port10,          flashRight,    tmotorVexFlashlight, openLoop, reversed)
//*!!Code automatically generated by 'ROBOTC' configuration wizard                !!*//

#define DETECTION_THRESHOLD 250
//*!!Code automatically generated by 'ROBOTC' configuration wizard                !!*//

void drive();
void steer();
void toggleFlash();
void spinLight();
void blinkLED();
```

Main



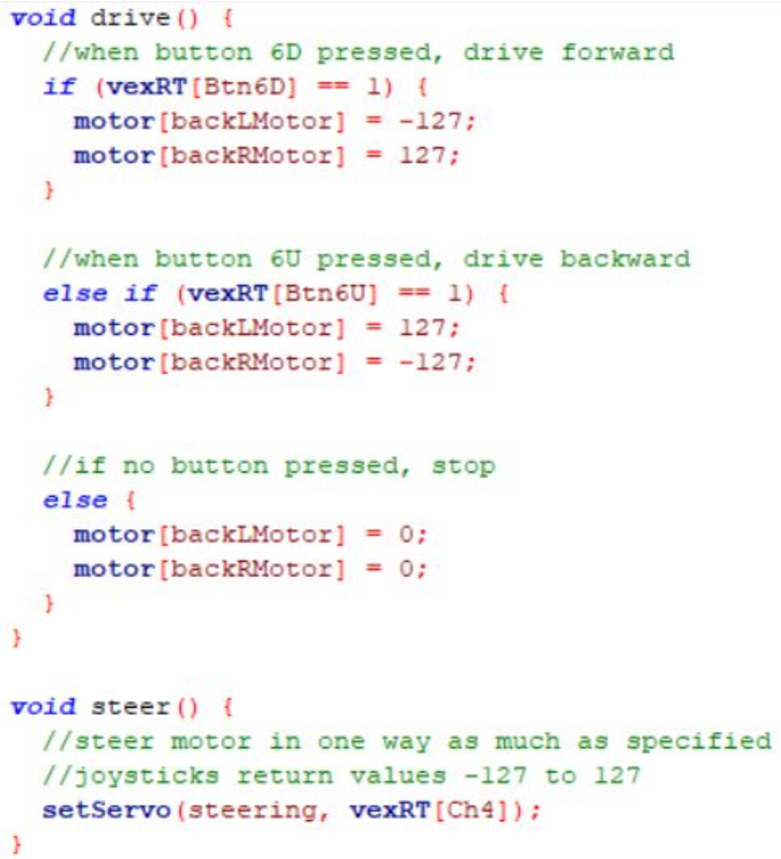
```
task main() {

  int ambientLightValue = SensorValue(lightSensor);
  while (1 == 1) {
    turnLEDOff(greenLight);

    //condition works during day and night
    if (abs(ambientLightValue - SensorValue(lightSensor)) >= DETECTION_THRESHOLD) {
      blinkLED();
      break;
    }

    toggleFlash();
    drive();
    steer();
    spinLight();
  }
}
```

Functions

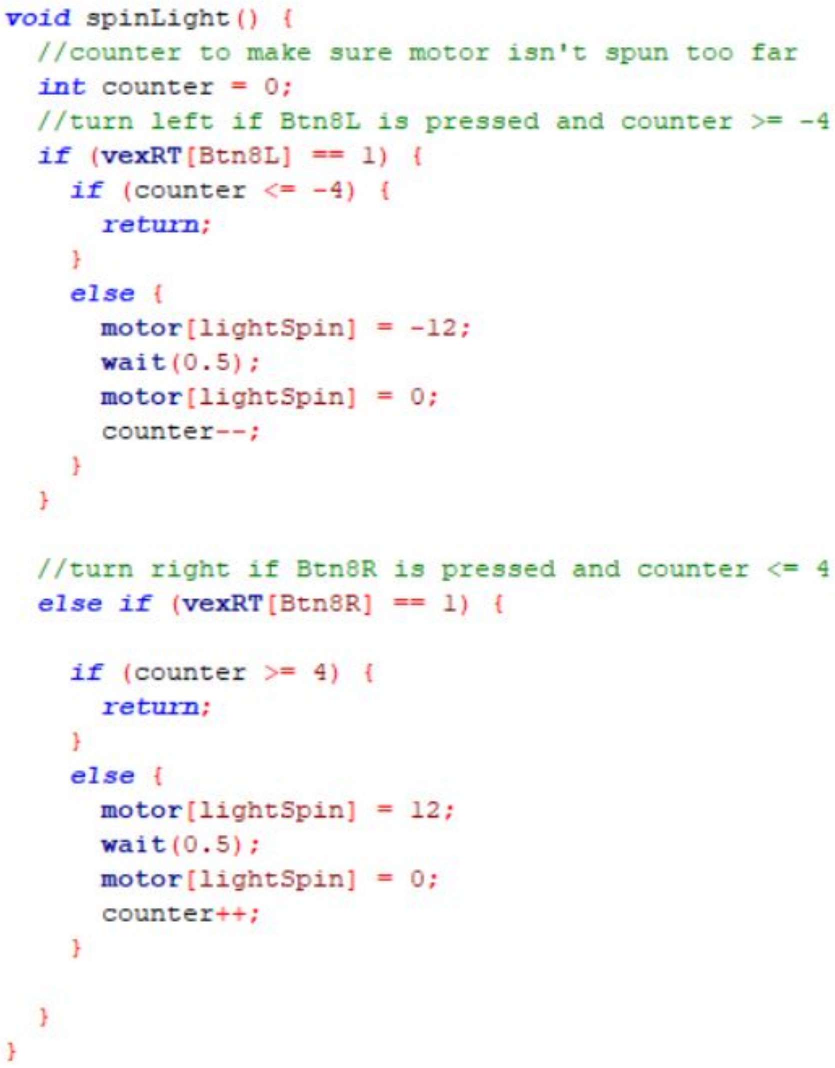


```
void drive() {
  //when button 6D pressed, drive forward
  if (vexRT[Btn6D] == 1) {
    motor[backLMotor] = -127;
    motor[backRMotor] = 127;
  }

  //when button 6U pressed, drive backward
  else if (vexRT[Btn6U] == 1) {
    motor[backLMotor] = 127;
    motor[backRMotor] = -127;
  }

  //if no button pressed, stop
  else {
    motor[backLMotor] = 0;
    motor[backRMotor] = 0;
  }
}

void steer() {
  //steer motor in one way as much as specified
  //joysticks return values -127 to 127
  setServo(steering, vexRT[Ch4]);
}
```



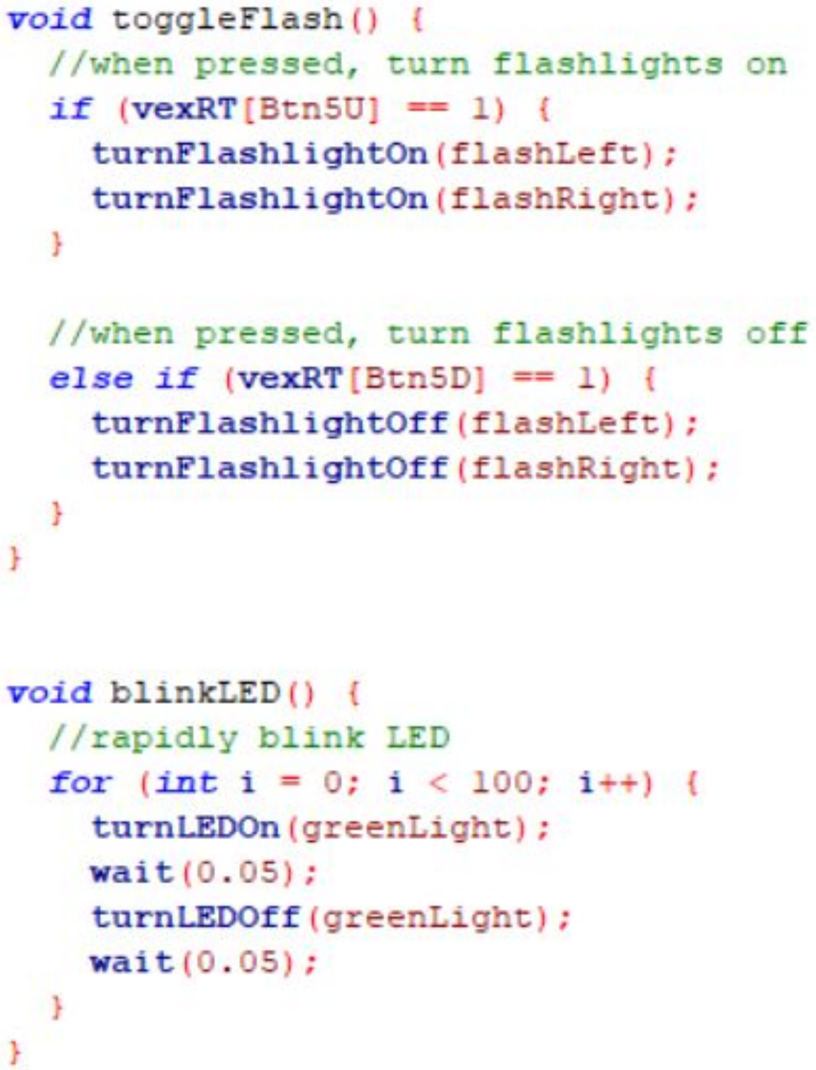
```
void spinLight() {
  //counter to make sure motor isn't spun too far
  int counter = 0;
  //turn left if Btn8L is pressed and counter >= -4
  if (vexRT[Btn8L] == 1) {
    if (counter <= -4) {
      return;
    }
    else {
      motor[lightSpin] = -12;
      wait(0.5);
      motor[lightSpin] = 0;
      counter--;
    }
  }

  //turn right if Btn8R is pressed and counter <= 4
  else if (vexRT[Btn8R] == 1) {

    if (counter >= 4) {
      return;
    }
    else {
      motor[lightSpin] = 12;
      wait(0.5);
      motor[lightSpin] = 0;
      counter++;
    }

  }
}
```

Functions



```
void toggleFlash() {
  //when pressed, turn flashlights on
  if (vexRT[Btn5U] == 1) {
    turnFlashlightOn(flashLeft);
    turnFlashlightOn(flashRight);
  }

  //when pressed, turn flashlights off
  else if (vexRT[Btn5D] == 1) {
    turnFlashlightOff(flashLeft);
    turnFlashlightOff(flashRight);
  }
}


void blinkLED() {
  //rapidly blink LED
  for (int i = 0; i < 100; i++) {
    turnLEDOn(greenLight);
    wait(0.05);
    turnLEDOff(greenLight);
    wait(0.05);
  }
}
```

Demonstration

Operating the Spotlight Sidekick

- Flashlights and Light Sensor
- Wheels & Steering
- Light Indicator

Development Problems

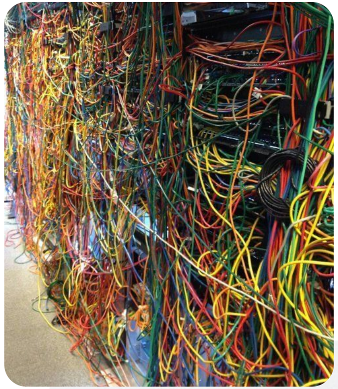
Swiveling Searchlight

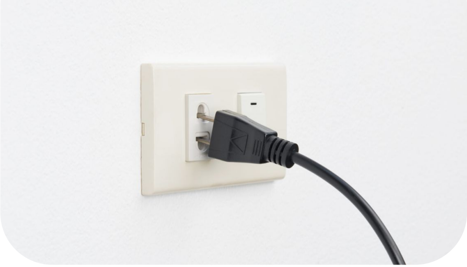
- Original design structurally unstable
- Motors couldn't turn the searchlights enough
- Tangling wires prevented more than a full rotation of the searchlight in either direction

Poorly seated battery plug and wire connections to the cortex caused lots of confusion because they are very hard to spot and misleading

VEXnet RC Keys / Dongles

- Little documentation available
- Connection would frequently drop for apparently no reason, sometimes after power intensive moves.
- Does not work in areas with strong signal or lots of radio interference; very short range.





Possible Improvements or Additions

1. Better Wire and Wire Connections
    a. Pins or prongs making a more solid seating in the ports will allow for cleaner and more flexible cable management
        i. Will decrease chances of disconnection and are more resilient to being bent, pinched, etc.
    b. Wires that will clip together eliminate bad connections and reduce sources of problems during testing and operation.
    c. Battery wires will have less chance to rip and detach from connector when unplugging.
2. Stronger VEXnet keys
    a. Better able to overcome interference / won’t drop connection as often
    b. More operating range and flexibility than a USB cable