Code

```
#include <LiquidCrystal.h>

// IR Pins
#define IR_RIGHT A0
#define IR_MIDDLE A1
#define IR_LEFT A2

// Right motor Pins
#define RIGHT_MOTOR_EN 6
#define RIGHT_MOTOR_PIN1 9
#define RIGHT_MOTOR_PIN2 8

// Left motor Pins
#define LEFT_MOTOR_EN 5
#define LEFT_MOTOR_PIN1 7
#define LEFT_MOTOR_PIN2 10

// Motor Speed
#define MOTOR_SPEED 255

// LCD Pins
#define LCD_RS 0
#define LCD_EN 1
#define LCD_D4 2
#define LCD_D5 3
#define LCD_D6 4
#define LCD_D7 11

// IR Reads
#define GROUND HIGH
#define LINE LOW

class Motor {
    private:
    int enable;
    int pin1;
    int pin2;

    public:
    // Motor();
    Motor(int enable, int pin1, int pin2) {
        this->enable = enable;
        this->pin1 = pin1;
        this->pin2 = pin2;
    }
```

```
    void init() {
        pinMode(enable, OUTPUT);
        pinMode(pin1, OUTPUT);
        pinMode(pin2, OUTPUT);
    }

    void forward() {
        digitalWrite(pin1, HIGH);
        digitalWrite(pin2, LOW);
        analogWrite(enable, MOTOR_SPEED);
    }

    void stop() {
        digitalWrite(pin1, LOW);
        digitalWrite(pin2, LOW);
        analogWrite(enable, 0);
    }

    void backward() {
        digitalWrite(pin1, LOW);
        digitalWrite(pin2, HIGH);
        analogWrite(enable, MOTOR_SPEED);
    }
};

class Controller {
    private:
    Motor* rightMotor;
    Motor* leftMotor;
    public:
    Controller(Motor &rightMotor, Motor &leftMotor) {
        this->rightMotor = &rightMotor;
        this->leftMotor = &leftMotor;
    }

    void init() {
        TCCR0B = TCCR0B & B11111000 | B00000010;
    }

    void forward() {
        rightMotor->forward();
        leftMotor->forward();
    }

    void right() {
        rightMotor->stop();
        leftMotor->forward();
    }
```

```
    void left() {
        leftMotor->stop();
        rightMotor->forward();
    }

    void rightCorner() {
        rightMotor->backward();
        leftMotor->forward();
    }

    void leftCorner() {
        leftMotor->backward();
        rightMotor->forward();
    }

    void stop() {
        rightMotor->stop();
        leftMotor->stop();
    }
};

class Screen {
    private:
    LiquidCrystal* lcd;

    public:
    Screen(int rs, int en, int d4, int d5, int d6, int d7) {
        lcd = new LiquidCrystal(rs, en, d4, d5, d6, d7);
    }

    void init() {
        lcd->begin(16, 2);
        displayTeamName();
    }

    void displayTeamName() {
        lcd->setCursor(4, 0);
        lcd->print("G ee k s");
    }

    void displayTimer() {
        lcd->setCursor(0, 1);
        lcd->print(millis() / 1000);
    }

    void displayDirection(const char* direction) {
        lcd->setCursor(4, 1);
        lcd->print(direction);
```

```
    }

    void displayRegion(char region) {
        lcd->setCursor(15, 1);
        lcd->print(region);
    }
};

class Sensor {
    private:
    int pin;
    public:
    Sensor(int pin) {
        this->pin = pin;
    }

    void init() {
        pinMode(pin, INPUT);
    }

    int read() {
        return digitalRead(pin);
    }
};

Motor rightMotor(RIGHT_MOTOR_EN, RIGHT_MOTOR_PIN1, RIGHT_MOTOR_PIN2);
Motor leftMotor(LEFT_MOTOR_EN, LEFT_MOTOR_PIN1, LEFT_MOTOR_PIN2);
Controller ctrl(rightMotor, leftMotor);
Screen screen(LCD_RS, LCD_EN, LCD_D4, LCD_D5, LCD_D6, LCD_D7);
Sensor rightSensor(IR_RIGHT);
Sensor middleSensor(IR_MIDDLE);
Sensor leftSensor(IR_LEFT);

void setup() {
    rightMotor.init();
    leftMotor.init();
    ctrl.init();
    screen.init();
    rightSensor.init();
    middleSensor.init();
    leftSensor.init();
}

void loop() {
    screen.displayTimer();

    int rightIRRead = rightSensor.read();
    int middleIRRead = middleSensor.read();
    int leftIRRead = leftSensor.read();
```

```
    if (rightIRRead == GROUND && leftIRRead == GROUND) {
        ctrl.forward();
        screen.displayDirection(" FORWARD");
    } else if (rightIRRead == GROUND && middleIRRead == GROUND && leftIRRead ==
LINE) {
        ctrl.left();
        screen.displayDirection("  LEFT  ");
    } else if (rightIRRead == GROUND && middleIRRead == LINE && leftIRRead ==
LINE) {
        ctrl.leftCorner();
        screen.displayDirection(" C LEFT ");
    } else if (rightIRRead == LINE && middleIRRead == GROUND && leftIRRead ==
GROUND) {
        ctrl.right();
        screen.displayDirection("  RIGHT ");
    } else if (rightIRRead == LINE && middleIRRead == LINE && leftIRRead ==
GROUND) {
        ctrl.rightCorner();
        screen.displayDirection(" C RIGHT");
    } else if (rightIRRead == LINE && middleIRRead == LINE && leftIRRead == LINE)
{
        ctrl.forward();
        screen.displayDirection(" FORWARD");
    } else {
        ctrl.stop();
    }
}
```