function spdMap(dist:number, v1:number){

    let fastWheel:number = 160 //转向的时候更快的轮子(可以看作匀速档位的基准值)

    let slowWheel:number = 90 //转向的时候更慢的轮子(可以看作匀速档位的基准值)

    let list:Array<number> = [fastWheel, fastWheel]//输出的快慢轮子速度值

    if (dist <= 45 && dist > 5){ //距离小于60的时候，速度档位生效

        let speedLevel = Math.map(dist-45, -45, 0, 0, 4)//dist越小，速度档位越低

        fastWheel = fastWheel+Math.round(Math.map(speedLevel, 0, 4, -60, 60))//根据距离相应地加速or减速

        slowWheel = slowWheel+Math.round(Math.map(speedLevel, 0, 4, -60, 60))//速度值取个整

    } else if (dist <= 5){

        fastWheel = 0 //太近就停下来

        slowWheel = 0

    } else {

        fastWheel = v1 //太远就保持原本的匀速挡速度，没有快慢轮了

        slowWheel = v1

    }

    list = [fastWheel, slowWheel]

    return list

}

function follow\_mode(stdL1:number, stdL2:number){

    L = robotbit.RgbUltrasonic(DigitalPin.P0)//与前车距离

    let v1 = 150 //随便设置一个仅仅在follow\_mode里面使用的匀速档位速度,我这里设置给左轮子

    let speed\_delta = 12 //根据自己的车的情况改这个数，左右轮子速度差

    let light\_delta = 10 //左右灯光亮度差值（容错区间）

    if ((stdL1-stdL2) > light\_delta){ //右转

        robotbit.MotorRunDual(

    robotbit.Motors.M1A,

    spdMap(L,v1)[1],

    robotbit.Motors.M2A,

    spdMap(L,v1)[0]//左边是fastWheel

    )

    } else if ((stdL2-stdL1) > light\_delta){//左转

        robotbit.MotorRunDual(

    robotbit.Motors.M1A,

    spdMap(L,v1)[0],

    robotbit.Motors.M2A,

    spdMap(L,v1)[1]//右边是fastWheel

    )

    } else {

        robotbit.MotorRunDual(//容错区间内不转向,取快慢轮的平均速度作为两个轮子的速度。

    robotbit.Motors.M1A,

    0.5\*(spdMap(L,v1)[0]+spdMap(L,v1)[1]),

    robotbit.Motors.M2A,

    0.5\*(spdMap(L,v1)[0]+spdMap(L,v1)[1])+speed\_delta

    )

    }

}

let L = 10000 //初始化与前车的距离，无限大

let thresh = 100

let baseline1 = 500

let baseline2 = 500

baseline1 = pins.analogReadPin(AnalogPin.P1)

baseline2 = pins.analogReadPin(AnalogPin.P2)

basic.forever(function () {

    let stdL1 = Math.map(pins.analogReadPin(AnalogPin.P1), 0, baseline1, 0, 400)//标准化后的左边light level

    let stdL2 = Math.map(pins.analogReadPin(AnalogPin.P2), 0, baseline1, 0, 400)

    let mode = 0

    if (stdL1 < thresh || stdL2 < thresh){//足够强的光源才能出发follow\_mode()

        mode = 2 //跟踪模式

    } else if (stdL1 >= thresh && stdL2 >= thresh){

        mode = 1 //游走模式

    }

    if (mode == 1){ //这里嵌套一个游走模式的代码block就可

    robotbit.MotorRunDual(

    robotbit.Motors.M1A,

    100,

    robotbit.Motors.M2A,

    100

    )

    } else if (mode == 2){

        follow\_mode(stdL1, stdL2)

    } else if (mode == 0){

        //冬眠模式，暂时没写

    }

})