

## « Le programme du char »

```
radio.onReceivedNumber(function (receivedNumber) {
  reception = receivedNumber
  if (reception == 0) {
    basic.showLeds(`
      . # . # .
      . . . . .
      # . . . #
      . # # # .
      . . . . .
      `)
  }
  if (reception == 1) {
    basic.showLeds(`
      . . # . .
      . # . . .
      # # # # #
      . # . . .
      . . # . .
      `)
    Kitronik_Move_Motor.turnRadius(Kitronik_Move_Motor.TurnRadii.Tight)
    Kitronik_Move_Motor.move(Kitronik_Move_Motor.DriveDirections.Right, 20)
    basic.pause(200)
    Kitronik_Move_Motor.move(Kitronik_Move_Motor.DriveDirections.Forward, 1
0)
  }
  if (reception == 2) {
    basic.showLeds(`
      . . # . .
      . . . # .
      # # # # #
      . . . # .
      . . # . .
      `)
    Kitronik_Move_Motor.turnRadius(Kitronik_Move_Motor.TurnRadii.Tight)
    Kitronik_Move_Motor.move(Kitronik_Move_Motor.DriveDirections.Left, 20)
    basic.pause(200)
    Kitronik_Move_Motor.move(Kitronik_Move_Motor.DriveDirections.Forward, 1
0)
  }
})
let reception = 0
let moveMotorZIP = Kitronik_Move_Motor.createMoveMotorZIPLD(4)
Kitronik_Move_Motor.setUltrasonicUnits(Kitronik_Move_Motor.Units.Centimeters)
let distance = 25
basic.showIcon(IconNames.Heart)
```

```

radio.setGroup(66)
reception = 4
moveMotorZIP.setZipLedColor(0, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.White))
moveMotorZIP.setZipLedColor(1, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.White))
moveMotorZIP.setZipLedColor(2, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Red))
moveMotorZIP.setZipLedColor(3, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Red))
basic.forever(function () {
  if (reception == 3) {
    moveMotorZIP.setZipLedColor(0, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.White))
    moveMotorZIP.setZipLedColor(1, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.White))
    moveMotorZIP.setZipLedColor(2, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Red))
    moveMotorZIP.setZipLedColor(3, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Red))
    moveMotorZIP.show()
    Kitronik_Move_Motor.move(Kitronik_Move_Motor.DriveDirections.Forward, 2
0)
    basic.showIcon(IconNames.Triangle)
  }
  if (reception == 2) {
    for (let index = 0; index < 3; index++) {
      moveMotorZIP.setZipLedColor(0, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Orange))
      moveMotorZIP.setZipLedColor(3, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Orange))
      moveMotorZIP.show()
      basic.pause(100)
      moveMotorZIP.clear()
      moveMotorZIP.show()
      basic.pause(100)
    }
    reception = 3
  }
  if (reception == 1) {
    for (let index = 0; index < 3; index++) {
      moveMotorZIP.setZipLedColor(1, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Orange))
      moveMotorZIP.setZipLedColor(2, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Orange))
      moveMotorZIP.show()
    }
  }
}

```

```

        basic.pause(100)
        moveMotorZIP.clear()
        moveMotorZIP.show()
        basic.pause(100)
    }
    reception = 3
}
})
basic.forever(function () {
    if (distance < 20) {
        reception = 4
        Kitronik_Move_Motor.stop()
        Kitronik_Move_Motor.motorBalance(Kitronik_Move_Motor.SpinDirections.Left, 0)
        Kitronik_Move_Motor.motorBalance(Kitronik_Move_Motor.SpinDirections.Right, 0)
        Kitronik_Move_Motor.motorOn(Kitronik_Move_Motor.Motors.MotorRight, Kitronik_Move_Motor.MotorDirection.Reverse, 20)
        Kitronik_Move_Motor.motorOn(Kitronik_Move_Motor.Motors.MotorLeft, Kitronik_Move_Motor.MotorDirection.Reverse, 20)
        moveMotorZIP.setZipLedColor(0, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Blue))
        moveMotorZIP.setZipLedColor(1, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Blue))
        moveMotorZIP.setZipLedColor(2, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Blue))
        moveMotorZIP.setZipLedColor(3, Kitronik_Move_Motor.colors(Kitronik_Move_Motor.ZipLedColors.Blue))
        moveMotorZIP.show()
        basic.showLeds(`
            . . # . .
            . # . # .
            # . . . #
            . # . # .
            . . # . .
            `)
        basic.pause(1500)
        basic.clearScreen()
        Kitronik_Move_Motor.stop()
    }
    distance = Kitronik_Move_Motor.measure()
})

```

---

## Le programme de la télécommande

```
input.onButtonPressed(Button.A, function () {
  radio.sendNumber(2)
  basic.showLeds(`
    . . # . .
    . # . . .
    # # # # #
    . # . . .
    . . # . .
  `)
})
input.onButtonPressed(Button.AB, function () {
  radio.sendNumber(3)
  basic.showLeds(`
    . . # . .
    . # # # .
    # . # . #
    . . # . .
    . . # . .
  `)
})
input.onButtonPressed(Button.B, function () {
  radio.sendNumber(1)
  basic.showLeds(`
    . . # . .
    . . . # .
    # # # # #
    . . . # .
    . . # . .
  `)
})
basic.showIcon(IconNames.Happy)
let message = 0
radio.setGroup(66)
radio.sendNumber(0)
basic.pause(2000)
basic.clearScreen()
```