CSE435 – Robotics

Lab 3 – RobotBASIC Remote Control

Introduction

- The main goal of robotics is to make robots **autonomous**, allowing them to make decisions based on sensor data.
- Before achieving autonomy, it's important to **manually control the robot** to understand its behavior and motion commands.
- This lab focuses on **remote control programming** using **RobotBASIC**.

Remote Control Styles

There are **three general control modes**:

1. **Momentary Control (Hold-to-Move)**
    - The robot moves **only while a button is pressed**.
    - Releasing the button stops motion.
2. **Toggle Control (Press-to-Start/Stop)**
    - Pressing a button **starts movement**, and pressing again **stops it**.
3. **Command Execution Mode**
    - The robot **executes a full instruction sequence** (like moving to a location) before waiting for the next command.

Programming Constructs in RobotBASIC

Variables

- Store numeric or text values.
- Names must begin with a letter and are **case sensitive**. Example: Distance, distance, and DISTANCE are different.

Keyboard Functions

- GetKey Var: Reads if any key is pressed (returns ASCII code).
- WaitKey "msg", Var: Pauses execution until a key is pressed.
- Ascii("A") → 65, Char(65) → "A"
- Input ExprS, Var: Lets user type input (string or number).

Mouse Function

- ReadMouse Var1,Var2,Var3
    - Gets the **cursor position** and **button status** (without pausing execution).

Random Numbers

- Random(n) → generates number from 0 to n–1.

Loops

Used to continuously check input and control robot motion.

For Loop

```
for I = 1 to 10
  // code
next
```

While Loop

```
while <conditional expression, repeat if true>
  // code
wend
```

repeat .. until

```
repeat
  // code
until <conditional expression, repeat if false>
```

Simple Remote Control Programs

Style 1 – Hold-to-Move Control

Manual control using **keyboard keys**:

```
rLocate 400, 300

while true
    waitKey "h: foward, n:right, v: left, b: back", key

    if key = ascii("h")
      rForward 5
    elseif key = ascii("n")
      rTurn 5
    elseif key = ascii("v")
      rTurn -5
    elseif key = ascii("b")
      rForward -5
    endif

wend
```

With sensors included (for obstacle detection):

- Uses rGpsX(), rGpsY(), and rCompass() for location display.
- Uses rBumper() to avoid collisions before moving.

Style 2 – Toggle Control

Robot **starts/stops** with repeated key presses:

```
rLocate 400, 300


prevKey = 0
lastValidKey = 0
status = 0


while true
    getKey key

    if key <> 0 and key <> prevKey
      if key = ascii("h") or key = ascii("n") or key = ascii("v") or key = ascii("b")
        lastValidKey = key
        status = 1 - status
      endif
    endif

    prevKey = key

    if status
      if lastValidKey = ascii("h")
        rForward 5
      elseif lastValidKey = ascii("n")
        rTurn 5
      elseif lastValidKey = ascii("v")
        rTurn -5
      elseif lastValidKey = ascii("b")
        rForward -5
      endif
    endif


    delay 10

wend
```

Complex Remote Control (Using Mouse)

- Combines **mouse input**, **robot motion**, and **display updates**.
- Allows moving to a **clicked point** or **drawing paths**.

Main Structure

1. **Main Program** calls subroutines:
    - Draw_Obstacles
    - RemoteControl
2. **RemoteControl Subroutine**
    - Uses ReadMouse x,y,b to detect:
        - **Left-click:** Move robot to point (GotoPoint)
        - **Right-click:** Toggle pen up/down (drawing)
    - Displays position and compass data.
3. **GotoPoint Subroutine**

- Calculates direction and distance using:
  - PolarA(dx, dy) for angle
  - PolarR(dx, dy) for distance
- Uses rTurn and rForward commands for movement.
- Stops if a bumper sensor detects a collision.

Key RobotBASIC Functions Used

Function	Description
rForward(n)	Moves robot forward (negative = backward)
rTurn(angle)	Rotates robot
rGpsX(), rGpsY()	Current robot coordinates
rCompass()	Current heading
rBumper()	Detects collisions
rpen up/down	Controls drawing mode
rInvisible	Makes robot body invisible for visual clarity