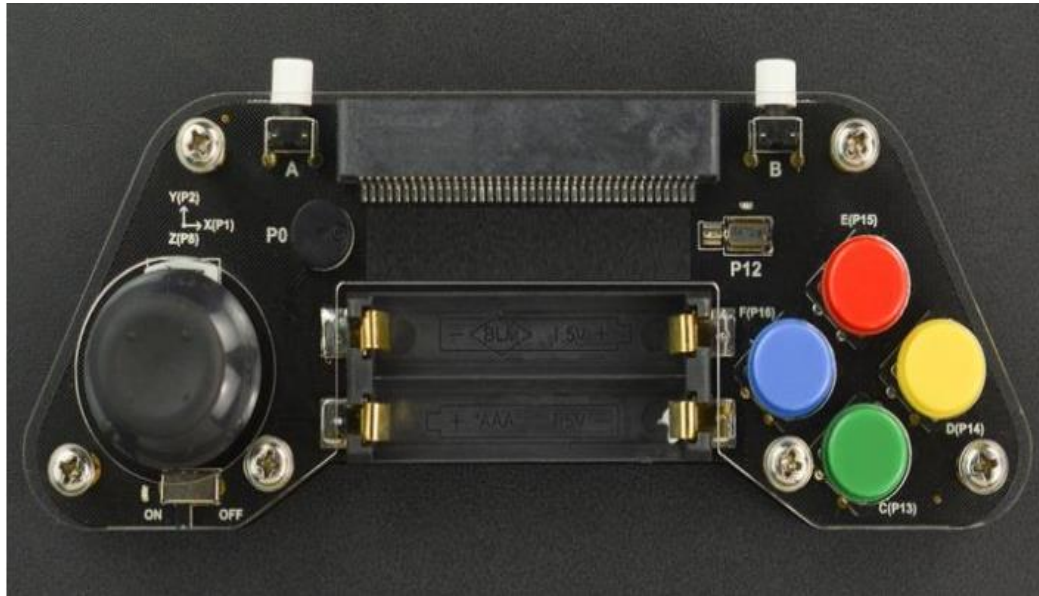


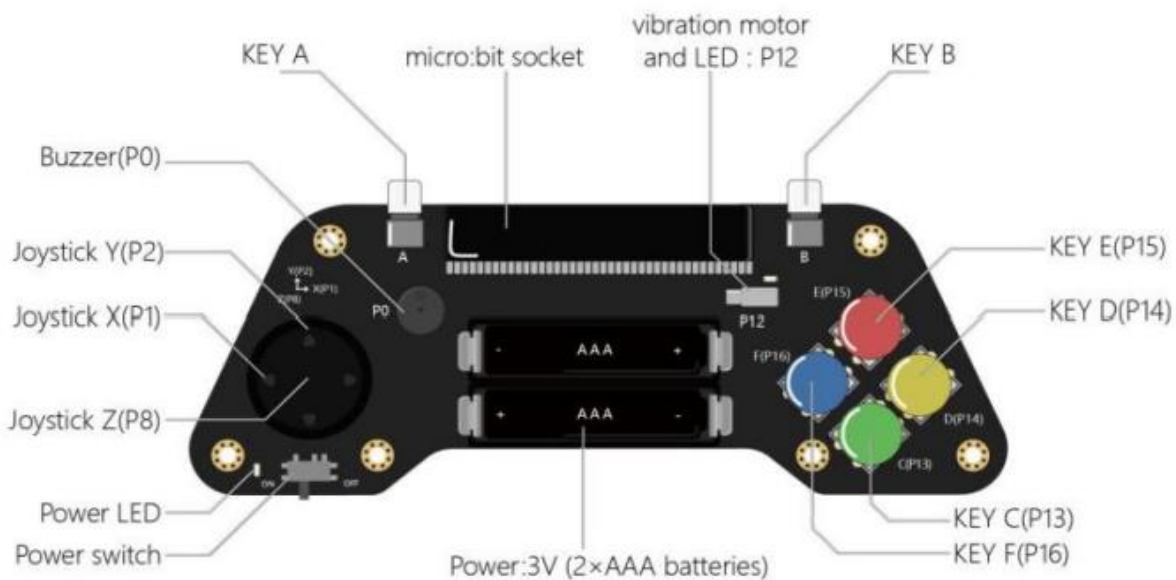
Bluguard GamePad For micro:bit



Chapter 1: Introduction to Bluguard GamePad For micro:bit

Bluguard GamePad for micro:bit is a mobile controller designed with a joystick. It employs a high-precision three-axis analog joystick. The integration of joystick and gamepad help user to perform Bluguard Maqueen's lite direction and speed simultaneously. In addition, there are 7 configurable buttons allowing users to explore more functions and more flexible controls. The onboard PCB has been equipped with external battery box.

Suggest Age: 8 +



Specification:

- Power: 3V DC (2 x AAA batteries)
- Joystick: 2 axis analog (X: P1 Y: P2) 1 axis digital (Z: P8)
- Onboard LED and vibration motor: P12
- Onboard buzzer: P0
- Keys: A(A), B(B), C(P13), D(P14), E(P15), F(P16), Z(P8)
- Dimension: 5.83x2.24inch / 148 X 57mm

Packing List:

- Bluguard GamePad for micro:bit x 1

Chapter 2: Bluguard GamePad Coding

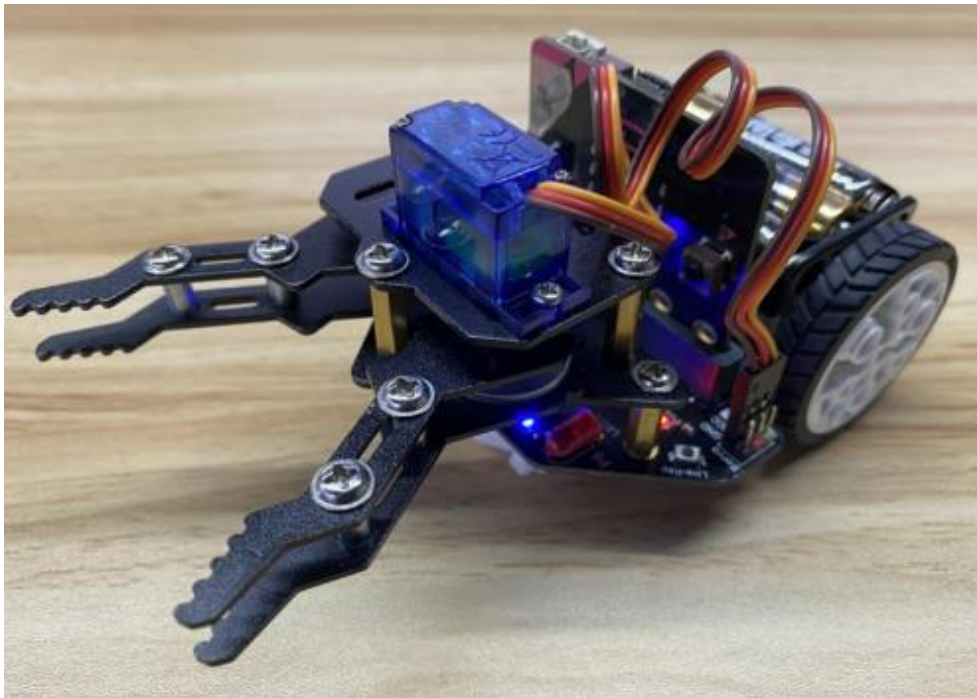
This chapter consists of various programming techniques on Bluguard GamePad on MakeCode. The MakeCode programming platform address: <https://makecode.microbit.org> and you can directly program this product without any third-party libraries.

Project 1 – Control Bluguard Maqueen Lite Via Switch Quantity

This project employs the Bluguard Gamepad's joystick to perform Bluguard Maqueen Lite direction to move forward and backward, turn left and right. You can press the up (red) and down (green) buttons to control the movement of the gripper, the left (blue) and right (yellow) buttons to turn the robot car's LEDs on/off. Install the gripper on Bluguard Maqueen Lite and connect it to port S1.

Servo motor connection as below:

GND = Brown
VCC = Orange
S = Yellow



Program for Maqueen Lite: <https://makecode.microbit.org/Vot7AfUW7g6D>

```
on start
  radio set group 1
  set angle to 90
  servo S1 angle angle
```

```
on radio received receivedString
  if receivedString = "Open" then
    if angle > 0 then
      change angle by -1
      servo S1 angle angle
    +
  else if receivedString = "Close" then -
    if angle < 180 then
      change angle by 1
      servo S1 angle angle
    +
  else if receivedString = "LEDL" then -
    LEDlight left turn ON
  else if receivedString = "LEDR" then -
    LEDlight right turn ON
  else if receivedString = "F" then -
    motor all move Forward at speed 100
  else if receivedString = "B" then -
    motor all move Backward at speed 100
  else if receivedString = "L" then -
    motor left move Forward at speed 20
    motor right move Forward at speed 100
  else if receivedString = "R" then -
    motor left move Forward at speed 100
    motor right move Forward at speed 20
  else -
    LEDlight left turn OFF
    LEDlight right turn OFF
    motor all stop
  +
```

Program for GamePad: <https://makecode.microbit.org/PtgDRughPidF>

```
on start
  radio set group 1
  set pull pin P13 to none
  set pull pin P14 to none
  set pull pin P15 to none
  set pull pin P16 to none

forever
  if digital read pin P15 = 0 then
    radio send string "Open"
  else if digital read pin P13 = 0 then
    radio send string "Close"
  else if digital read pin P16 = 0 then
    radio send string "LEOL"
  else if digital read pin P14 = 0 then
    radio send string "LEOR"
  else
    if analog read pin P2 > 550 and analog read pin P1 > 400 and analog read pin P1 < 600 then
      radio send string "F"
    else if analog read pin P2 < 450 and analog read pin P1 > 400 and analog read pin P1 < 600 then
      radio send string "B"
    else if analog read pin P1 < 450 and analog read pin P2 > 400 and analog read pin P2 < 600 then
      radio send string "L"
    else if analog read pin P1 > 550 and analog read pin P2 > 400 and analog read pin P2 < 600 then
      radio send string "R"
    else
      radio send string "S"
```

Project 2 – Control Maqueen Lite Via Analog Quantity

Great! We have learned how to manipulate and control the direction of the robot car in the Project 1. Here, we will use analog quantity of the joystick to control the direction and speed of the robot car at the same time. Press the up (red) and down (green) buttons to control the movement of the loader, the left (blue) and right (yellow) buttons to turn the robot car's LEDs on/off.



Program for Maqueen: <https://makecode.microbit.org/3WRLraYKPM2y>

```
on radio received name value
  if name = "F" then
    motor all move Forward at speed map value from low 550 high 1023 to low 10 high 255
  else if name = "B" then
    motor all move Backward at speed map value from low 1 high 450 to low 255 high 10
  else if name = "L" then
    motor right move Forward at speed map value from low 1 high 450 to low 255 high 40
    motor left move Forward at speed 20
  else if name = "R" then
    motor left move Forward at speed map value from low 550 high 1023 to low 40 high 255
    motor right move Forward at speed 20

on start
  radio set group 1
  set angle to 90
  servo S1 angle angle
```

```
on radio received receivedString
  if receivedString = "Open" then
    if angle > 0 then
      change angle by -1
      servo S1 angle angle
    else if receivedString = "Close" then
      if angle < 180 then
        change angle by 1
        servo S1 angle angle
      else if receivedString = "LEDL" then
        LEDlight left turn ON
      else if receivedString = "LEDR" then
        LEDlight right turn ON
      else
        motor all stop
        LEDlight left turn OFF
        LEDlight right turn OFF
```

Program for GamePad: <https://makecode.microbit.org/1WYL7aKctdav>

```
on start
  radio set group 1
  set pull pin P13 to none
  set pull pin P15 to none
  set pull pin P14 to none
  set pull pin P16 to none

forever
  if digital read pin P15 = 0 then
    radio send string "Open"
  else if digital read pin P13 = 0 then
    radio send string "Close"
  else if digital read pin P16 = 0 then
    radio send string "LEDL"
  else if digital read pin P14 = 0 then
    radio send string "LEDR"
  else
    if analog read pin P2 > 550 and analog read pin P1 > 400 and analog read pin P1 < 600 then
      radio send value 'F' = analog read pin P2
    else if analog read pin P2 < 450 and analog read pin P1 > 400 and analog read pin P1 < 600 then
      radio send value 'B' = analog read pin P2
    else if analog read pin P1 < 450 and analog read pin P2 > 400 and analog read pin P2 < 600 then
      radio send value 'L' = analog read pin P1
    else if analog read pin P1 > 550 and analog read pin P2 > 400 and analog read pin P2 < 600 then
      radio send value 'R' = analog read pin P1
    else
      radio send string "S"
```


Project 3 – Dice Rolling Game

The “Dice Rolling Game” typically use the wireless communication of two micro:bit boards to manipulate the dice to create a random number within 1~6, and then compare the two numbers subsequently, the one who gets the larger number will be become winner. The vibrator motor will vibrate to celebrate it. Download codes into two micro:bit boards and plug them into two GamePads when running this example.

Program Link: https://makecode.microbit.org/_YdWVgV7j7LhA

```
on start
  radio set group 1
  set item to 0
  set value to 0

on button A pressed
  set item to 0
  set value to 0
  digital write pin P12 to 0
  set item to pick random 1 to 6
  repeat item times
    do
      change value by 1
      play tone Middle D for 1 beat
      show number value
  radio send number value

on radio received receivedNumber
  if value < receivedNumber then
    digital write pin P12 to 0
  else
    digital write pin P12 to 1
```