Peanut Queen Controller

Peanut Queen Controller

Peanut King

Introduction

Peanut Queen Controller is a mobile application that provides Bluetooth-based remote control and real-time data monitoring for Peanut King Solution's products(listed in Specification). The app features a fully customizable dashboard that can be dynamically configured through simple text commands, enabling users to create custom control interfaces without reprogramming the app.

Features

Bluetooth UART connection & auto-reconnect
Dynamic GUI configuration (Button, Toggle, Slider, TextField, Joystick, Variable)
Read app controls in real-time (Getters)
Send micro:bit sensor data back to the app (Setters)
Built-in throttling to prevent UART buffer overflow
Event-driven button press handling with auto-state reset

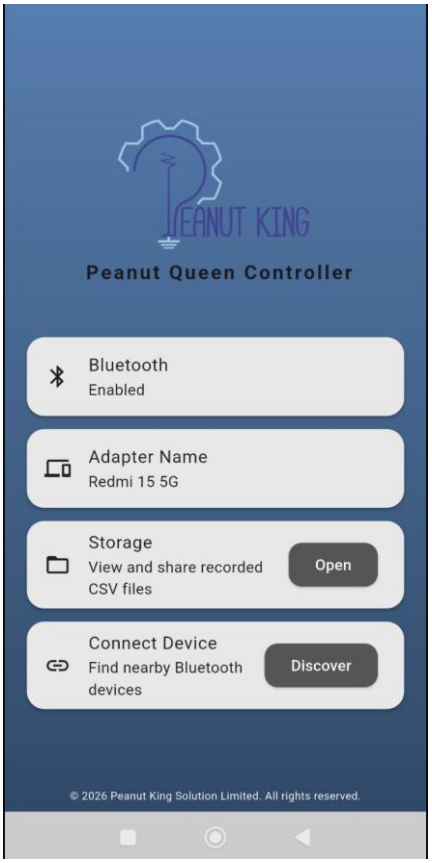
Specification

Hardware: micro:bit V2 with Peanut King micro:bit shield V2
Programming platform: MakeCode for micro:bit
Communication Protocol: Bluetooth UART
Component Name Length ≤ 15 characters
App available on Android and IOS(Progressing)

© 2026 Peanut King Solution Limited. All rights reserved.

Printed on 28 July 2026
This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Interface



Connection

1. Download the app in the link provided below
   https://play.google.com/store/apps/details?id=com.peanut.king.solution
2. Launch the Peanut Queen Controller app
3. Enable Bluetooth on your mobile device
4. Tap the discover button
5. Select your device from the list
6. Once connected, the dashboard will be ready for configuration

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Dashboard Configuration

The app dashboard is configured by sending a configuration message from the connected device (micro:bit/ Arduino) to the app via bluetooth. The app parese the message and renders the corresponding UI components.

Configuration Message Format:
C,I,[no.of_input_components],[Input_components],O,[no.of_output_components],[Output_components],\n

"C"	Configuration command header
I	Input components section marker
[no.of_input_components]	Total number of input controls
[Input_components]	List of input component definitions
O	Output components section marker
[no.of_output_components]	Total number of output plots
[Output_components]	List of output plot definitions
\n	Newline terminator

Supported Input Components:
1. Slider
   Format: S, [min_value], [max_value], [name],
   Example: S,0,255,speed
2. Joystick
   Format: J,[angle_name],[max_strength],[strength_name],
3. Button (Momentary)
   Format: B, [button_name]
   Example: B, fire
4. Toggle Button
   Format: TB, [toggle_button_name]
   Example: TB, mode,
5. Text Field
   Format: TF, [field_name]
   Example: [TF, status]

Output Components (Plots):
Format: [plot_name], [plotable_flag],
[plot-name]: Name of the data series
[plotable-flag]: **true** = enable time graphing, **false** = display as value onlt

Configuration Example:
Here is a complete configuration message that creates a dashboard with buttons, sliders, joysticks, toggle buttons, text fields, and data plots:

C,I,10,B,left,TF,Status,B,right,S,0,255,SliderL,S,0,255,SliderR,J,angle1,255,strength1,Joystick1,J,angle2,255,strength2,Joystick2,TB,ToggleButton,TB,ToggleButtonNum,TF,TextFieldNumber,O,3,ultrasound,true,grayscaleLeft,true,grayscaleRight,false,\n

© 2026 Peanut King Solution Limited. All rights reserved.

Printed on 28 July 2026
This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Visual Breakdown

Section	Components
Input (I,10)	Button: left, TextField: Status, Button: right, Slider: SliderL (0-255), Slider: SliderR (0-255), Joystick: Joystick1 (angle1/strength1), Joystick: Joystick2 (angle2/strength2), ToggleButton: ToggleButton, ToggleButton: ToggleButtonNum, TextField: TextFieldNumber
Output (O,3)	Plot: ultrasound (graph enabled), Plot: grayscaleLeft (graph enabled), Plot: grayscaleRight (graph disabled)

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Data Communication

Sending Control Values (Device ← App)
When a user interacts with an input component, the app sends the updated value to the connected device in the following format

T,[component_name],[component_value],\n

T	Transmission command header
[component_name]	Name of the UI component
[component_value]	Current value of the component
\n	Newline terminator

Example:

T,speed,188,\n
T,angle1,45,\n
T,fire,1,\n

Sending Data to App (Device → App)

To update the output plots and data displays, the connected device sends data messages to the app:

Single value format:

R,[component_name],[component_value],\n

Multiple values format:

R,[comp1],[val1],[comp2],[val2],...,\n

Example:

R,ultrasound,30,\n
R,grayscaleLeft,148,grayscaleRight,78,\n

Field	Description
R	Report/Update command header
[component_name]	Name of the plot component
[component_value]	Data value to display/plot
\n	Newline terminator

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Command Reference Summary·

Configuration Commands (Device → App)

Command	Format	Description
Configuration Start	C,I,[count],[components],O,[count],[components],\n	Defines the entire dashboard layout

Input Component Commands (Device → App)

Component	Format	Example
Slider	S,[min],[max],[name],	S,0,255,speed,
Joystick	J,[angle],[max],[strength],	J,angle1,255,strength1,
Button	B,[name],	B,fire,
Toggle Button	TB,[name],	TB,power,
Text Field	TF,[name],	TF,status,

Data Commands

Direction	Format	Description
App → Device	T,[name],[value],\n	Control value update
Device → App	R,[name],[value],\n	Single data report
Device → App	R,[n1],[v1],[n2],[v2],...,\n	Multiple data report

© 2026 Peanut King Solution Limited. All rights reserved.

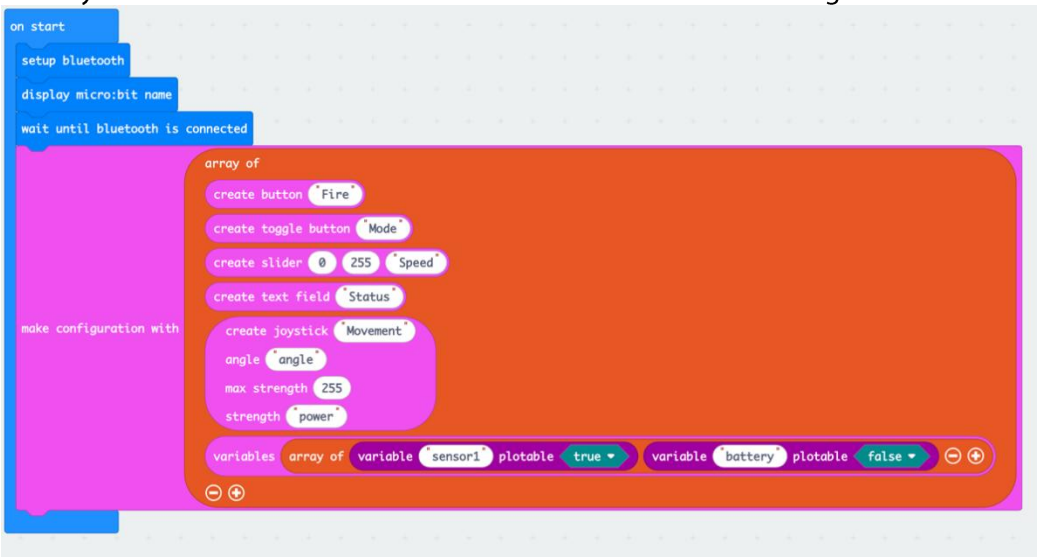
This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Application – micro:bit(MakeCode)

1. Open MakeCode for micro:bit editor
2. Click **Extensions** and enter the following URL
   https://github.com/peanut-king-solution/pxt-pks-controller
3. Click and search result to add the extension to your project.
4. In MakeCode, select Python and copy the following code to initialize the Bluetooth Service and define the layout in the app's interface.

```
pksController.setupBluetooth()
pksController.showDeviceName()
pksController.waitUntilConnected()
pksController.makeConfiguration([pksController.createButton("Fire"),
    pksController.createToggleButton("Mode"),
    pksController.createSlider(0, 255, "Speed"),
    pksController.createTextField("Status"),
    pksController.createJoystick("angle", 255, "power", "Movement"),
    pksController.formatVariablesList([pksController.createVariable("sensor1", true),
    pksController.createVariable("battery", false)])
])
```

You may use the blocks to have the same result as illustrated in the figure below

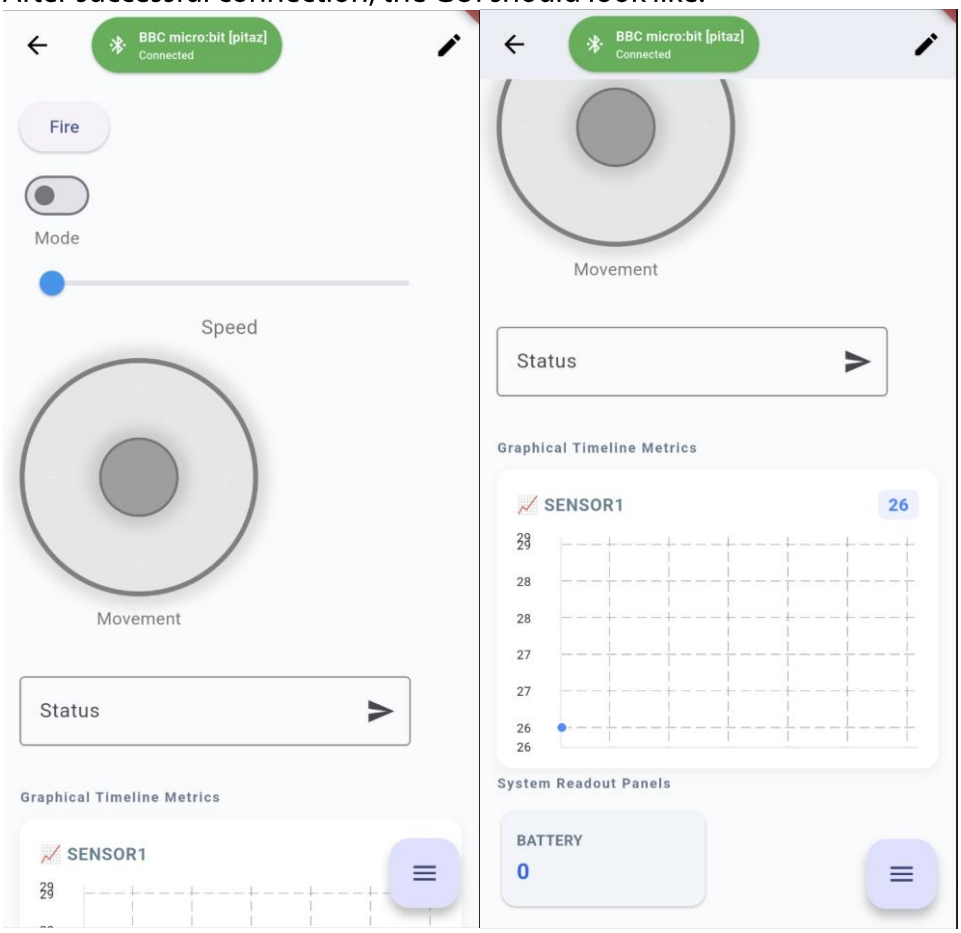


5. The GUI is configured
6. Use the app to connect

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

7. After successful connection, the GUI should look like:



* To build a custom control interface, you must define the layout using the configuration builder. All component names must not exceed 15 characters.

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Application – micro:bit(MakeCode)

Example: Read Values from App (Getters)

Access the current state of the app's GUI elements using the type-safe Getter blocks. These update in real-time as the user interacts with the app.

```
basic.forever(function () {
    // Check if a button is pressed
    if (pksController.buttonToggled("Fire")) {
        basic.showIcon(IconNames.Sword)
    }

    // Get a slider's value
    let speed = pksController.sliderValue("Speed")

    // Get Joystick data
    let angle = pksController.joystickAngle("Movement")
    let power = pksController.joystickStrength("Movement")

    // Get a text field's value
    let statusText = pksController.textFieldValue("Status")

    serial.writeLine("Speed: " + speed + ", Angle: " + angle)
    basic.pause(100)
})
```

Example: Send Data to App (Setters)

Push data from the micro:bit back to the app to update the GUI dynamically. The extension automatically handles UART locking and throttling to prevent serial overflow.

```
basic.forever(function () {
    // Send a number (e.g., from a sensor)
    let temp = input.temperature()
    pksController.sendVariableToApp("sensor1", temp)

    // Send a boolean (true/false)
    let isMoving = pksController.joystickStrength("Movement") > 0
    pksController.sendBooleanToApp("battery", isMoving)

    basic.pause(500)
})
```

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Application – micro:bit(MakeCode)

Example: Handle Button Events

Use the Event block to run code exactly when a button is tapped.

```
// Register code to run when the "Fire" button is pressed in the app
pksController.onButtonPressed("Fire", function () {
    basic.showLeds(`
        . . # . .
        . # # # .
        # # # # #
        . # # # .
        . . # . .
        `)
    music.play(music.builtinPlayableSoundEffect(soundExpression.happy), musi
c.PlaybackMode.UntilDone)
})
```

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.

Important Information and Disclaimer

The information in this statement reflects Peanut King Solution Limited's knowledge and belief as of the date it is provided. Peanut King Solution Limited relies on information from third parties and does not guarantee its accuracy. Efforts are ongoing to better integrate third-party information. Peanut King Solution Limited has taken and continues to take reasonable steps to provide accurate and representative information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. Certain information is considered proprietary by Peanut King Solution Limited and its suppliers, so CAS numbers and other limited information may not be available for release.

NOTICE

Peanut King Solution Limited reserves the right to make changes to the products in this document to improve performance or for any other reason, or to discontinue them without notice. Customers should contact Peanut King Solution Limited for the latest product information before placing orders or designing Peanut King Solution Limited products into systems. Peanut King Solution Limited assumes no responsibility for the use of any products or circuits described in this document or customer product design, does not convey any license under any patent or other right, and does not guarantee that the circuits are free of patent infringement. Peanut King Solution Limited makes no claim regarding the suitability of its products for any particular purpose and assumes no liability arising from the use of any product or circuit, specifically disclaiming any and all liability, including consequential or incidental damages.

PEANUT KING SOLUTION LIMITED PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN CRITICAL APPLICATIONS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY OR DEATH. THE USE OF PEANUT KING SOLUTION LIMITED PRODUCTS IN LIFE SUPPORT SYSTEMS IS NOT AUTHORIZED.

© 2026 Peanut King Solution Limited. All rights reserved.

This material may not be reproduced or used in any manner whatsoever without the express written permission of the publisher.