Quick Reference Handbook

- CodyNick Python Quick Reference Handbook

CodyNick Python Quick Reference Handbook

Start Code

```
import CodyNick


cn = CodyNick.CN()
```

`cn` is the connected CodyNick device object. Pass it as the first argument to CodyNick hardware functions.

RGB LED Matrix

Set One LED By Number

```
CodyNick.RGB_Matrix.set(cn, led, color)
```

Parameter	Description
`cn`	Connected CodyNick device
`led`	LED number from `0` to `15`
`color`	Color code `0..7`, RGB hex string, or RGB percentage list

Accepted `color` formats:

```
0
"#00FFFF"
[0, 100, 100]
```

Examples:

```
CodyNick.RGB_Matrix.set(cn, 15, 0)
CodyNick.RGB_Matrix.set(cn, 15, "#00FFFF")
CodyNick.RGB_Matrix.set(cn, 15, [0, 100, 100])
```

Set One LED By Coordinate

```
CodyNick.RGB_Matrix.set_xy(cn, x, y, color)
```

Parameter	Description
`cn`	Connected CodyNick device
`x`	Horizontal coordinate from `0` to `3`
`y`	Vertical coordinate from `0` to `3`
`color`	Color code `0..7`, RGB hex string, or RGB percentage list

Coordinate system:

```
(0,3)  (1,3)  (2,3)  (3,3)
(0,2)  (1,2)  (2,2)  (3,2)
(0,1)  (1,1)  (2,1)  (3,1)
(0,0)  (1,0)  (2,0)  (3,0)
```

Example:

```
CodyNick.RGB_Matrix.set_xy(cn, 0, 0, "#FF0000")
```

Clear All RGB LEDs

```
CodyNick.RGB_Matrix.clear(cn)
```

Turns off all RGB LEDs.

Built-In Color Codes

Code	Name	Hex
0	Red	`#FF0000`
1	Cyan	`#00FFFF`
2	Blue	`#0000FF`
3	Yellow	`#FFFF00`
4	Magenta	`#FF00FF`
5	Green	`#00FF00`
6	Orange	`#FF8000`
7	White	`#FFFFFF`

Joystick

Check One Direction

```
CodyNick.Joystick.position(cn, direction)
```

Parameter	Description
`cn`	Connected CodyNick device
`direction`	`"up"`, `"down"`, `"left"`, or `"right"`

Returns `True` or `False`.

Example:

```
if CodyNick.Joystick.position(cn, "up"):
    print("up")
```

Check Button Click

```
CodyNick.Joystick.click(cn)
```

Returns `True` when the joystick button is pressed.

Read All Current States

```
CodyNick.Joystick.states(cn)
```

Returns a list such as:

```
[]
["UP"]
["LEFT"]
["CLICK"]
["UP", "CLICK"]
```

Example:

```
states = CodyNick.Joystick.states(cn)
```

CJP Sound Maker

Play A Note And Continue

```
CodyNick.CJP_Sound_Maker.play(cn, note, duration_ms)
```

Parameter	Description
`cn`	Connected CodyNick device
`note`	Musical note from `B0` to `D#8`, such as `"C4"`, `"F#2"`, or `"Gb2"`
`duration_ms`	Duration in milliseconds

Example:

```
CodyNick.CJP_Sound_Maker.play(cn, "F#2", 300)
```

Play A Note And Wait Until Done

```
CodyNick.CJP_Sound_Maker.play_until_done(cn, note, duration_ms)
```

Same parameters as `play()`, but Python waits until the note is finished.

Example:

```
CodyNick.CJP_Sound_Maker.play_until_done(cn, "C4", 500)
```

Seven Segment Display

Display A Number

```
CodyNick.Seven_Segment.display(cn, value)
```

Parameter	Description
`cn`	Connected CodyNick device
`value`	Number or numeric string to show

Examples:

```
CodyNick.Seven_Segment.display(cn, 1234)
CodyNick.Seven_Segment.display(cn, -123)
CodyNick.Seven_Segment.display(cn, 1.234)
```

Motion Detection

Detect Motion

```
CodyNick.Motion_Detection.detect(cn)
```

Parameter	Description
`cn`	Connected CodyNick device

Returns:

```
True
False
```

Example:

```
if CodyNick.Motion_Detection.detect(cn):
    print("motion detected")
```

Connection Control

Close The Connection

```
cn.close()
```

Closes the serial connection to the CodyNick device.

> **Quick summary:**
> Use `RGB_Matrix` for RGB LEDs, `Joystick` for joystick input, `CJP_Sound_Maker` for notes, `Seven_Segment` for numeric output, and `Motion_Detection` for PIR motion sensing.