

# Bluetooth HID Keyboard

SLQ Wiki Fabrication Lab 2025/07/18 10:22

# Bluetooth HID Keyboard

SLQ Wiki Fabrication Lab 2025/07/18 10:22

~~REVEAL~~

# Bluetooth HID Keyboard

## Summary

This prototype was request by Daniel Flood to explore the possibility of make a low cost skooge-like Human Interface Device (HID). Brendan Halliday completed the prototype. The BlueDuino Rev2 is an Arduino compatible microcontroller development board based on the ATmega32U4 IC with Bluetooth 4.0 [Blueduino Documentation from Supplier](#)

## Required Files:

- [Coolterm Serial Console](#)
- [SerialBootTool from TI](#)
- [ZeroKeyboard Firmware](#)
- [Blueduino Library](#)

## Required Hardware:

- Blueduino Board
- FTDI or CP2102 USB-Serial converter.
- Breadboard
- Buttons
- Hookup wires (Male-Male, Male-Female, Female-Female)
- MicroUSB cable.

## Tools

Soldering Iron

## Preparation

**NOTE:** You must upload new firmware to the module if you command it into upgrade mode. Otherwise the module will stay in upgrade mode and seem bricked.

- Wire the USB to UART converter and BlueDuino like so

USB to UART converter	BlueDuino Rev2
RX	RXI (Serial1)
TX	TXO (Serial1)
GND	GND

## Activity Summary

- Plug in the BlueDuino to a PC. Take note of which COM port it shows up as in Device Manager.
- Plug in the USB to UART converter to the same PC. Take note of which COM port it shows up as in Device Manager.
- Open Coolterm, click on **Options**, then set Port to the COM port for the USB to UART converter.
- Set Baudrate to 9600.
- You will get "OK" when you type "AT" and pressing **ENTER**.
- So type "AT+SBLUP", you will get "OK+SBLUP". This means the cc254x module is ready for upgrade. It won't accept any other AT command now.
- Disconnect in Coolterm and then close it.
- Open Arduino and upload an empty sketch to the BlueDuino. An empty sketch will practically be `void setup(){} void loop() {}`
- Open SerialBootTool, it will ask you what device family to use, select BLE.
- Click on **Select File** and select the keyboard-1.0.bin file. Click **Open**.
- Click on **Port Settings** and select the USB to UART serial port and baud rate 115200.
- Click **OK**
- Click **Load Image**
- Wait a while. You will see the words "Download completed successfully"
- You're done!

## Arduino Sample Code

```
#include "AB_BLE.h"

#define BAUD_RATE 57600

AB_BLE ble(&Serial1);

const int buttonPin = 8;    // the number of the pushbutton pin

// variables will change:
int buttonState1 = 0;       // variable for reading the pushbutton status

void setup() {
  Serial1.begin(BAUD_RATE);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState1 = digitalRead(buttonPin);
  // check whether a pushbutton is pressed.
  // if it is, the buttonState is HIGH:

  if (buttonState1 == LOW) {
```

```
ble.print("AT+KEY=");  
ble.println("Hello world");  
}  
delay(500);  
}
```

## Notes and additions

- This kind of bluetooth keyboard has an inherent delay of approximately 100 milliseconds which means it may not be directly usable for bluetooth gamepads but great for simple interactivity.
- The example I've provided above is not properly debounced, which means that it can trigger the keys being sent a few times before you let go. This is why there is a delay(500); as a workaround.
- The key presses are sent by the ble.print("AT+KEY="); and ble.println("Hello world"); lines, change "Hello world" for any phrase or keyboard key you want to be pressed.
- This firmware does not seem to support anything other than the alphanumeric keys.

## Production notes

## Feedback

1. Feedback:

Solution:

2. Feedback:

Solution:

3. Feedback:

Solution:

4. Other observations:

## References

[LED generator](#)

## Files